DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“Sistema inteligente de asistencia a la comunicación oral de personas con disartria utilizando reconocimiento de patrones de voz”

T E S I S

PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE

MAESTRA EN CIENCIAS DE LA COMPUTACIÓN

Paloma Valeria Contreras González

Director:
Dra. María Trinidad Serna Encinas

Codirectora:
Dra. Rosalía del Carmen Gutiérrez Urquídez

Hermosillo, Sonora, México 22 de junio de 2022

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Hermosillo
División de Estudios de Posgrado e Investigación

Hermosillo, Sonora a 22 de junio DE 2022
SECCIÓN: DIV. EST. POS. E INV.
No. OFICIO: DEPI/133/22.
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS.

C. PALOMA VALERIA CONTRERAS GONZÁLEZ
P R E S E N T E

Por este conducto, y en virtud de haber concluido la revisión del trabajo de tesis que lleva por nombre **"SISTEMA INTELIGENTE DE ASISTENCIA A LA COMUNICACIÓN ORAL DE PERSONAS CON DISARTRIA UTILIZANDO RECONOCIMIENTO DE PATRONES DE VOZ"**, quien fue dirigida por la Dra. María Trinidad Serna Encinas y la Dra. Rosalía del Carmen Gutiérrez Urquídez, que presenta para el examen de grado de la MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN, y habiéndola encontrado satisfactoria, nos permitimos comunicarle que se autoriza la impresión del mismo a efecto de que proceda el trámite de obtención de grado.

Deseándole éxito en su vida profesional, quedo de usted.

A T E N T A M E N T E

DRA. MARÍA TRINIDAD SERNA ENCINAS
DIRECTORA DE TESIS

M.C. CÉSAR ENRIQUE ROSE GÓMEZ
SECRETARIO

M.C. RAFAEL ARMANDO GALAZ BUSTAMANTE
VOCAL

DR. GERMÁN ALONSO RUÍZ DOMÍNGUEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

GARD/momv*

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE HERMOSILLO
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

100% LIBRE DE PLÁSTICO
ISO 9001:2015

2022 Año de Ricardo Flores Magón
PRECURSOR DE LA REVOLUCIÓN MEXICANA

CARTA CESIÓN DE DERECHOS

En la ciudad de Hermosillo Sonora a el día 22 de junio del año 2022 la que suscribe C. Paloma Valeria Contreras González, alumna de la maestría en Ciencias de la Computación adscrito a la División de Estudios de Posgrado e Investigación, manifiesta que es autora intelectual del presente trabajo de Tesis titulado “Sistema inteligente de asistencia a la comunicación oral de personas con disartria utilizando reconocimiento de patrones de voz” bajo la dirección de Dra. María Trinidad Serna Encinas y ceden los derechos del mismo al Tecnológico Nacional de México/Instituto Tecnológico de Hermosillo, para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben de reproducir el contenido textual, graficas, tablas o datos contenidos sin el permiso expreso del autor y del director del trabajo. Este puede ser obtenido a la dirección de correo electrónico siguiente: m15331156@hermosillo.tecnm.mx. Una vez otorgado el permiso se deberá expresar el agradecimiento correspondiente y citar la fuente del mismo.

A T E N T A M E N T E

Paloma Valeria Contreras González

Agradecimientos

Esta tesis no sólo refleja mi esfuerzo durante estos años de maestría, también refleja el apoyo de las personas que me acompañaron a lo largo de este camino. Muchas gracias a mi mamá y a mi papá, por todo el cariño y apoyo que me han brindado toda la vida. Gracias a mis hermanos: Andrea, César y José Miguel. Porque siempre han sido personas ejemplares para mí, de quienes he aprendido que es posible lograr cualquier cosa cuando te lo propones. También me han enseñado que la sencillez y humanidad son clave para ser un buen profesionista.

Gracias a Milthon, por acompañarme tanto en los momentos de logro y felicidad como en aquellos en los que más abrumada me sentía. Te agradezco tu presencia, tu apoyo y tu interés por aprender junto conmigo a lo largo de la maestría.

Muchas gracias al Centro de Rehabilitación e Inclusión Infantil Teletón (CRIT) de Sonora, por abrirme las puertas y darme la oportunidad de aportar mi "granito de arena". No me queda duda de la gran labor que se realiza día con día en el Centro. Muchas gracias a Ernesto Ontiveros, a los niños y respectivos tutores que me apoyaron para el desarrollo de esta investigación. Fueron una pieza fundamental. Muchas gracias por su disposición y amabilidad en todo momento.

Gracias a mis compañeros de maestría, con quienes tuve la oportunidad de coincidir y aprender muchas cosas. Valoro mucho el apoyo que siempre existió entre nosotros. Muchas gracias a mis maestros porque aprendí muchísimo de ellos; no sólo en temas que me forjaron en lo profesional, sino también en lo personal. Muchas gracias a la Dra. María Trinidad Serna Encinas por su apoyo durante todo este tiempo; agradezco su paciencia, su amabilidad, disposición y su gran corazón. Sin duda alguna es usted para mí una gran profesora, directora y amiga.

Resumen

La disartria es un trastorno que frecuentemente limita a la persona que la padece en su interacción con otras personas. Esto debido a la reducción en la inteligibilidad de su habla a causa del trastorno; teniendo como resultado la exclusión social.

Con la finalidad de brindar a las personas con disartria una herramienta que la apoye en su interacción con otros, se han desarrollado sistemas enfocados al reconocimiento de patrones de voz para personas con este trastorno. Sin embargo, debido a que son enfocados a un idioma en específico, como el inglés, o por sus altos costos, estos sistemas tienden a ser inaccesibles.

En el presente trabajo de investigación se propone un sistema de reconocimiento de patrones de voz enfocado a personas con disartria leve y moderada. Para su desarrollo se parte desde la investigación de los conceptos más relevantes, la identificación y aplicación de técnicas de inteligencia artificial hasta el análisis de los resultados obtenidos.

Debido a que el objetivo de la investigación es desarrollar un sistema dependiente del usuario, se crearon distintos conjuntos de datos para cada una de las personas que apoyaron en el desarrollo de la investigación. Los conjuntos de datos obtenidos, los cuales contienen entre 195 y 300 datos, fueron utilizados para el entrenamiento y evaluación del modelo propuesto. Debido a la escasez de datos, se considera la extracción de características del modelo YAMNet, ofrecido por TensorFlow Hub, para utilizar los datos obtenidos como entrada a una red neuronal multicapa. Para la disartria leve, se obtuvo un reconocimiento de 77% en la clasificación de 5 palabras; mientras que, para la disartria moderada, se obtuvo un 62% para tres palabras y un 75% para dos palabras. Con ello se tiene como resultado dos modelos que pueden ser de apoyo para los casos de estudio; además de que se aportan las bases para su aplicación en más casos de disartria.

Índice general

Agradecimientos iii
Resumen iv
Lista de figuras vii
Capítulo I: Introducción 1
1.1 Introducción 1
1.2 Antecedentes 2
1.3 Planteamiento del problema 3
1.4 Objetivos 4
1.5 Justificación 5
1.6 Alcances y delimitaciones 5
1.7 Metodología 6
Capítulo II: Estado del arte 7
2.1 Introducción 7
2.2 Discapacidad 7
2.2.1 Clasificación de la discapacidad 8
2.2.2 Disartria 11
2.3 Inteligencia Artificial 14
2.3.1 Machine learning 15
2.3.2 Redes neuronales 16
2.3.3 Transfer learning 23
2.4 Reconocimiento automático del habla (RAH) 25
2.4.1 Clasificación de los sistemas para el RAH 26
2.5 Extracción de características en señales de audio para su uso en el aprendizaje profundo 28
2.5.1 Espectrograma 28
2.5.2 Espectrograma de Mel 30
2.6 TensorFlow 32
2.7 Trabajos relacionados 33
Capítulo III: Análisis y diseño del sistema 36
3.1 Introducción 36
3.2 Metodología 36
3.3 Diagramas del análisis del sistema 38

3.4 Diagramas del diseño del sistema 41
3.5 Modelo de datos 46
3.6 Arquitectura del sistema 47
Capítulo IV: Implementación de sistema 49
4.1 Conjuntos de datos (*datasets*) 49
4.2. Modelo basado en Transfer Learning 51
4.3. Aplicación móvil 58
Capítulo V: Resultados y discusión 64
5.1 Clasificación para caso I-DL 64
5.2 Entrenamiento para el caso Y-DM 68
5.3 Pruebas de usabilidad 72
5.4 Discusión 75
6.1 Conclusiones 77
6.2 Trabajo a futuro 78
Bibliografía 79

Lista de figuras

Figura 1 Modelo integral del funcionamiento y la discapacidad [13]. 8
Figura 2. Clasificación de la disartria. Elaboración propia. 13
Figura 3. Arquitectura de red neuronal. Adaptación de [26]. 16
Figura 4. Arquitectura de una red neuronal convolucional. Adaptado de [29]. 18
Figura 5. Función de activación ReLU [21]. 19
Figura 6. Conexión recurrente de una RNN [25]. 20
Figura 7. Versión "desplegada" de una RNN [32]. 21
Figura 8. Topologías de una RNN. Adaptado de [33]. 22
Figura 9. Diagrama de un sistema para el reconocimiento automático del habla [40]. 27
Figura 10. Señal en el dominio del tiempo vs en el dominio de la frecuencia [42]. 28
Figura 11. Espectrograma [43]. 30
Figura 12. Escala de Mel [48]. 31
Figura 13. Espectrograma de Mel [43]. 31
Figura 14. Pantalla de entrenamiento de la aplicación móvil Voiceitt [53]. 34
Figura 15. Interfaz gráfica de aplicación móvil VocaTempo [54]. 35
Figura 16. Metodología del proyecto. 37
Figura 17. Diagrama de contexto: nivel 0. 38
Figura 18. Diagrama de nivel superior: nivel 1. 39
Figura 19. Casos de uso. 40
Figura 20. Diagrama de clases. 42
Figura 21. Diagrama de actividades para añadir registro. 43
Figura 22. Diagrama de actividades para eliminar registro. 44
Figura 23. Diagrama de actividades para modificar registro. 44
Figura 24. Diagrama de actividades para entrenamiento del modelo. 45
Figura 25. Diagrama de actividades para reconocimiento de voz. 46
Figura 26. Modelo de datos. 47
Figura 27. Arquitectura del sistema. 48
Figura 28. Arquitectura MobileNetV1 [56]. 53
Figura 29. Proceso para la inferencia utilizando el modelo YAMNet [57]. 54
Figura 30. Modelo personalizado. 55
Figura 31. Proceso de inferencia utilizando modelo YAMNet y modelo personalizado [57]. 56

Figura 32. Gráfico de navegación del proyecto. 59
Figura 33. Pantalla principal. 60
Figura 34. Lista de palabras. 61
Figura 35. Interfaz para agregar nuevos registros. 62
Figura 36. Interfaz para la modificación, entrenamiento o eliminación de registro. 63
Figura 37. Matriz de confusión 2 segundos, 5 palabras y exactitud de 77%. 67
Figura 38. Matriz de confusión 2 segundos, 2 palabras y exactitud de 75%. 70
Figura 39. Matriz de confusión 2 segundos, 3 palabras y exactitud de 65%. 71

Lista de tablas

Tabla 1. Identificación de los casos y su respectiva disartria. 50
Tabla 2. Conjuntos de datos para el caso I-DL..... 51
Tabla 3. Entrenamiento con dataset I-DL-3 (duración de 3 segundos por audio), considerando 30, 35 y 40 repeticiones por palabra y un batch de 32. 64
Tabla 4. Entrenamiento con dataset I-DL-3, considerando 30, 35 y 40 repeticiones por palabra y un batch de 16. 65
Tabla 5. Entrenamiento con dataset I-DL-2 (duración de 2 segundos por audio), considerando 30, 35 y 40 repeticiones por palabra y un batch de 32. 66
Tabla 6. Entrenamiento con dataset I-DL-2, considerando 30, 35 y 40 repeticiones por palabra y un batch de 16. 66
Tabla 7. Comparación en la clasificación de 2, 3, 4 o 5 palabras, considerando 35 repeticiones y un batch de 16. 68
Tabla 8. Entrenamiento con dataset Y-DM para el reconocimiento de dos palabras, considerando 35, 40 y 45 repeticiones por palabra y un batch de 16. 69
Tabla 9. Entrenamiento con dataset Y-DM para el reconocimiento de dos palabras, considerando 35, 40 y 45 repeticiones por palabra y un batch de 32. 69
Tabla 10 Entrenamiento con dataset Y-DM para el reconocimiento de tres palabras, considerando 35 y 45 repeticiones por palabra y un batch de 16..... 70
Tabla 11. Entrenamiento con dataset Y-DM para el reconocimiento de tres palabras, considerando 35 y 45 repeticiones por palabra y un batch de 32..... 71
Tabla 12. Resultados obtenidos de pruebas de usabilidad aplicadas a tutor de caso I-DL. 73
Tabla 13. Resultados obtenidos en las pruebas de usabilidad aplicadas al caso I-DL. 74

Capítulo I: Introducción

1.1 Introducción

Según el Instituto Nacional de Estadística y Geografía (INEGI), en el año 2018, de 115.7 millones de personas mayores de 5 años que habitaban en México, 7.7 millones presentaban algún tipo de discapacidad, es decir, el 6.7% de la población. Tomando como base el total de la población con discapacidad, 9.7% presentaban una dificultad para hablar o comunicarse, es decir, más de 700,000 personas en el país [1]. Específicamente en Sonora, en el año 2014, aproximadamente 159,085 personas padecían algún tipo de discapacidad y, de ellas, más de 23 mil presentaban dificultad para hablar o comunicarse [2].

Dentro de las discapacidades para comunicarse se encuentra la disartria, la cual puede ser provocada por esclerosis lateral amiotrófica, parálisis cerebral, accidentes cerebro vasculares, traumatismos encéfalo craneanos, enfermedad de Parkinson, entre otros. La disartria se trata de un trastorno de origen neurológico, en el que existen problemas para el control de la musculatura utilizada para hablar, lo que puede derivar en síntomas como, por ejemplo, lentitud, incoordinación o presencia de movimientos involuntarios en el habla; por lo que la comunicación oral de las personas que la padecen se ve afectada en su calidad e inteligibilidad [3], [4].

Actualmente, la aplicación de sistemas basados en el reconocimiento automático del habla (RAH), para asistir la comunicación oral de personas con disartria, se ha visto prometedora. Sin embargo, existen escasos sistemas de este tipo para su aplicación en personas cuya lengua materna sea el español; por lo que el presente trabajo de investigación consiste en la implementación de un sistema móvil para dicho idioma que, por medio de una red neuronal, reconozca el habla de una persona con disartria para su posterior traducción por medio de un generador de voz. De esta manera se busca facilitar la comunicación de la persona con disartria al interactuar con otros.

1.2 Antecedentes

La disartria se trata de un trastorno de origen neurológico, en donde uno o más procesos motores básicos implicados en el habla, tales como la respiración, fonación, resonancia, articulación y prosodia, se ven afectados en grados variables; teniendo como resultado una reducción en la inteligibilidad del habla [3], [5]. Según el grado de severidad, la disartria se clasifica en leve, moderada y severa. En las personas con disartria leve se presentan problemas en la articulación de ciertos fonemas o una alteración en la velocidad y/o intensidad del habla; sin embargo, en este grado de severidad el habla es generalmente inteligible. En el segundo grado de severidad se presentan problemas más graves de articulación en una mayor cantidad de fonemas, teniendo como resultado un habla ininteligible en ciertas ocasiones; mientras que en la disartria severa, la movilidad de los músculos que repercuten en el habla se ve más afectada, por lo que se vuelve aún más complicado el entender a la personas con este grado de severidad [6].

Generalmente, el manejo de la disartria se enfoca en facilitar la recuperación funcional del habla del paciente por medio de terapias; sin embargo, en aquellos en los que no es posible una recuperación funcional, se recomienda el uso de medios aumentativos y/o alternativos de comunicación [3].

La aplicación de sistemas basados en el reconocimiento automático del habla (RAH), como medio de comunicación para personas con disartria, se ha visto prometedora, ya que en la mayoría de los casos, la disartria es acompañada por movimientos corporales limitados o con poca coordinación, haciendo complicado el uso de herramientas de comunicación basadas en teclados o palancas de mando [7]. No obstante, la inconsistencia en el habla de las personas con dicho trastorno impide su reconocimiento en sistemas desarrollados para el habla sin afectaciones [8], por lo que es necesario el desarrollo de sistemas de reconocimiento para su aplicación específica en la disartria.

Para el desarrollo de los sistemas basados en el RAH para la disartria, se han utilizados técnicas de Inteligencia Artificial como las redes neuronales. Éstas se inspiran en la estructura de neuronas interconectadas del cerebro al consistir en un conjunto de nodos conectados entre sí de una manera concreta, siendo organizados en grupos denominados capas. El comportamiento de las redes neuronales depende de la forma en la que los nodos se encuentran conectados, así como de la ponderación de las conexiones existentes [9],[10].

Debido a que las redes neuronales son capaces de aprender de los datos, generalizar y abstraer características principales de un conjunto de datos, es posible entrenarlas con la finalidad de reconocer patrones, clasificar datos o pronosticar eventos futuros [9].

En la aplicación de redes neuronales para el desarrollo de sistemas para el RAH enfocados a la disartria, se han desarrollado sistemas dependientes e independientes del hablante, los cuales en su mayoría se tratan de sistemas discretos. Por ejemplo, en [11] se desarrolló un sistema para el RAH en el idioma italiano de tipo dependiente del hablante, el sistema se basó en una red neuronal convolucional para el reconocimiento de 12 palabras, obteniendo un porcentaje de reconocimiento de 57.5%.

En [7] se desarrolló una red neuronal recurrente-convolucional para el idioma inglés de tipo independiente del hablante para el reconocimiento de 16 palabras. Para su desarrollo se utilizaron audios provenientes del *dataset* TORGO. En la evaluación, se obtuvo un 40.6% de reconocimiento frente a un 31.4% de una red neuronal convolucional desarrollada para su comparación bajo las mismas condiciones.

1.3 Planteamiento del problema

La disartria es un trastorno que puede repercutir negativamente en la calidad de vida de las personas que la padecen, ya que un trastorno de la comunicación no sólo afecta en la expresión de opiniones, necesidades y deseos, sino también repercute

en la autonomía de la persona y en su proceso de interacción con otras personas [12].

Debido a lo anterior, surge el reto de desarrollar una herramienta de comunicación alternativa para las personas que padecen este trastorno, la cual sea capaz de reconocer sus patrones de voz y de traducir su habla a un discurso claro. De ello se plantean las siguientes preguntas de investigación:

- ¿Qué es un patrón de voz?
- ¿Cuál es el proceso a seguir para lograr el reconocimiento de patrones de voz?
- ¿Qué características debe tener el sistema inteligente para ser utilizado por personas con disartria?
- ¿Qué algoritmo de inteligencia artificial es el más adecuado para el reconocimiento de patrones de voz en una persona con disartria?

Lo anterior nos lleva al siguiente planteamiento del problema:

¿Qué funcionalidades deben considerarse en la implementación de un sistema inteligente móvil, que asista la comunicación oral de personas con disartria?

1.4 Objetivos

A continuación, se describen los objetivos planteados para el desarrollo del trabajo de investigación propuesto.

1.4.1 Objetivo general

Implementar un sistema inteligente móvil para asistir la comunicación oral de personas con disartria en el idioma español, mediante el reconocimiento de patrones de voz, utilizando redes neuronales.

1.4.2 Objetivos específicos

- Conocer a profundidad los temas principales que inciden en el proyecto de investigación.
- Determinar las características y requerimientos funcionales a cumplir por el sistema móvil.
- Analizar y determinar el algoritmo de inteligencia artificial a utilizar.
- Implementar el sistema inteligente móvil.
- Analizar y validar las pruebas funcionales y de usabilidad del sistema.

1.5 Justificación

Los problemas de comunicación causados por la disartria pueden convertir a la expresión de ideas, deseos y necesidades en un reto; por lo que sería beneficioso el proporcionar a las personas con disartria, una herramienta que les ayude a traducir a un discurso claro lo expresado oralmente. Sin embargo, en la actualidad la mayoría de las investigaciones relacionadas al reconocimiento del habla de personas con disartria para su traducción, se enfocan en el idioma inglés.

Debido a lo anterior, el desafío del presente trabajo de investigación es la implementación de un sistema inteligente para el reconocimiento del habla enfocado al idioma español. De esta manera, no solo se pretende apoyar a personas hispanohablantes con disartria, sino que también se busca aportar conocimiento en el campo de estudio enfocado al desarrollo tecnológico para las discapacidades.

1.6 Alcances y delimitaciones

El presente trabajo se enfoca en la implementación de un sistema inteligente, el cual:

- Sea dependiente del usuario.
- Reconocerá palabras previamente entrenadas y utilizará un generador de voz para reproducir la salida.

- Considera el caso de una persona con disartria leve o moderada cuya lengua materna sea el español.

Además, el trabajo de investigación se llevará a cabo en cooperación académica entre el Centro de Rehabilitación e Inclusión Infantil Teletón (CRIT Sonora) ubicado en Hermosillo, Sonora, y el Instituto Tecnológico de Hermosillo, con el fin de identificar en colaboración nuestro caso de estudio.

1.7 Metodología

La metodología por seguir consiste de tres etapas:

La primera etapa consiste en realizar una investigación bibliográfica sobre las diferentes temáticas relacionadas con el trabajo de investigación. También contempla el análisis de las diferentes propuestas de solución existentes; así como el identificar los alcances y delimitaciones del trabajo.

La segunda etapa se relaciona con la identificación de los requerimientos funcionales y de usabilidad del sistema, además de su análisis y diseño.

La tercera etapa se compone de la implementación del sistema y el análisis de los resultados con la finalidad de verificar el funcionamiento del sistema.

Capítulo II: Estado del arte

2.1 Introducción

En este capítulo se realiza un análisis sobre los temas centrales del presente trabajo de investigación. Dentro de la primera sección (2.2), se hace un análisis de la discapacidad y sus tipos, para después comprender la disartria y sus características. En la siguiente sección (2.3), se abordan los temas relacionados con la inteligencia artificial. Posteriormente, se realiza un análisis sobre los sistemas para el reconocimiento automático del habla (sección 2.4), la extracción de características en señales de audio para su uso en el aprendizaje profundo (sección 2.5) y, finalmente, se explica qué es TensorFlow y sus aplicaciones (Sección 2.6). Por último, se describen diferentes trabajos desarrollados, los cuales se relacionan al presente proyecto (sección 2.6).

2.2 Discapacidad

La Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud (CIF), publicada por la Organización Mundial de la Salud (OMS) en el 2001, se considera como un lenguaje estándar y universal de los componentes de la salud; siendo un marco conceptual para la comprensión de la discapacidad, el funcionamiento y la salud [13].

La CIF indica que el funcionamiento hace referencia a las funciones corporales, las actividades y la participación de una persona [13]. El funcionamiento indica los aspectos positivos entre la interacción de una persona con cierta condición de salud y sus condiciones contextuales (condiciones ambientales y personales) [14]. Esto puede observarse en el modelo integral del funcionamiento y la discapacidad, el cual se encuentra en la Figura 1.

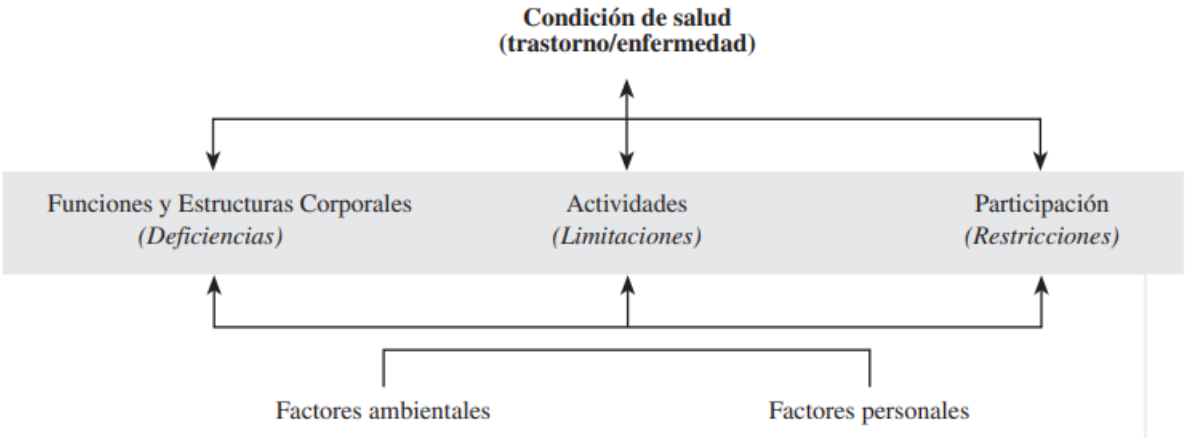

Figura 1 Modelo integral del funcionamiento y la discapacidad [13].

De manera específica, las funciones y estructuras corporales se refieren a las funciones fisiológicas y a los elementos anatómicos del cuerpo. Las actividades se entienden como la ejecución de tareas o acciones; mientras que la participación hace referencia al desenvolvimiento de una persona en situaciones vitales [14].

Por otro lado, la discapacidad es un término que abarca las deficiencias, las limitaciones de la actividad y las restricciones en la participación [13]. El primer componente, se refiere a la ausencia o alteración en las funciones y estructuras corporales; las limitaciones de la actividad incluyen las dificultades que una persona puede presentar para ejecutar tareas o acciones, y las restricciones de la participación hacen mención a aquellos problemas que una persona puede enfrentar para implicarse en situaciones consideradas como vitales. Mientras que el funcionamiento hace referencia a los aspectos positivos, la discapacidad se centra en los aspectos negativos de la interacción de una persona con cierta condición de salud y sus condiciones contextuales [14].

2.2.1 Clasificación de la discapacidad

La Clasificación de Tipo de Discapacidad publicada por el INEGI incluye distintas deficiencias y discapacidades, y muestra una clasificación de acuerdo con el órgano,

función o área del cuerpo que se encuentra afectada o presenta limitación. El clasificador se conforma por cuatro grupos principales, los cuales se dividen en subgrupos, como se muestra a continuación [15]:

- Discapacidades motrices: Se incluyen aquellas discapacidades para caminar, manipular objetos y coordinar movimientos. Se dividen en discapacidades de las extremidades inferiores, tronco, cuello y cabeza y discapacidades de las extremidades superiores.
  - Discapacidades de las extremidades inferiores, tronco, cuello y cabeza: incluyen limitaciones para moverse o caminar a causa de la falta total o parcial de las piernas, o debido a restricciones o ausencia de movimiento en ellas; la clasificación también incluye limitaciones para doblarse, estirarse, agacharse, además de deficiencias que afectan a la postura y el equilibrio del cuerpo. En este subgrupo se considera a la atetosis, atrofia de piernas, poliomielitis, parálisis de piernas, entre otros.
  - Discapacidades de las extremidades superiores: consideran aquellas limitaciones para utilizar brazos y manos debido a su falta de movimiento o a causa de su pérdida total o parcial. Ejemplos de discapacidades de las extremidades superiores son la ausencia de manos y la atrofia muscular en brazos o manos.
- Discapacidades mentales: abarcan las discapacidades para aprender y para comportarse, se divide en discapacidades intelectuales y discapacidades conductuales y otras mentales.
  - Discapacidades intelectuales: incluye a las discapacidades que se manifiestan como deficiencia mental y pérdida de la memoria. Dentro de las discapacidades intelectuales se considera, por ejemplo, a la enfermedad de Alzheimer, atrofia cerebral y amnesia.
  - Discapacidades conductuales y otras mentales: se trata de discapacidades en el rango de moderadas a severas, las cuales repercuten en el comportamiento de la persona. En este tipo de discapacidades se pueden presentar situaciones como, por ejemplo, que las personas puedan tener una respuesta inadecuada

ante situaciones externas, o que pueden presentar dificultad para identificar objetos y personas o las dimensiones de tiempo y espacio. Dentro de este subgrupo se considera la esquizofrenia, autismo, agorafobia, entre otros.

- Discapacidades sensoriales y de la comunicación: se tratan de aquellas relacionadas con la vista, escucha y habla. Estas discapacidades se dividen en los siguientes 4 subgrupos:
  - Discapacidades para ver: se incluye la pérdida total de la vista, debilidad visual y otras limitaciones como, por ejemplo, cataratas, acorea, dictioma y leucoma; en este caso se considera discapacidad si la condición se da en uno o los dos ojos.
  - Discapacidades para oír: abarcan la pérdida total o parcial severa de la audición en uno o ambos oídos.
  - Discapacidades para hablar: se considera la pérdida total del habla, es decir, la mudez.
  - Discapacidades de la comunicación y comprensión del lenguaje: se consideran incapacidades para producir, emitir y comprender mensajes del habla, además de limitaciones severas en el lenguaje que impiden el poder realizar mensajes claros; como ejemplos de este último subgrupo se tiene a la afasia, alexia, alofasia, disfasia, extirpación de la laringe, labio y paladar hendido y la disartria.
- Discapacidades múltiples y otras: comprende discapacidades múltiples y otras discapacidades no correspondientes a los grupos anteriormente mencionados.
  - Discapacidades múltiples: este subgrupo incluye a la hemiplejía, cuadriplejía, apoplejía, parálisis cerebral, parálisis agitante, entre otros.
  - Otro tipo de discapacidades: incluye malformaciones en partes del cuerpo, deficiencias de los órganos internos, así como enfermedades crónicas, degenerativas y progresivas.

2.2.2 Disartria

La disartria es un trastorno, resultado de lesiones en el sistema nervioso central o periférico, en el que se presenta dificultad para el control de los músculos y los movimientos que afectan en la producción del habla [16], [3]. Este trastorno se trata de una alteración del habla y no del lenguaje; es decir, la persona con disartria no presenta dificultad para entender o emplear el lenguaje hablado, sino que presenta problemas para articular el habla.

Esta discapacidad puede ser causada por enfermedades neuromusculares como la parálisis cerebral y la esclerosis múltiple; por daño cerebral causado por tumores, accidentes cerebrovasculares o lesiones traumáticas, o por daño a los nervios que actúan en los músculos faciales [17].

Las personas con disartria pueden presentar incoordinación en los movimientos para hablar, además de dificultades para realizar movimientos articulatorios imprecisos, complicados y lentos [18]. En ocasiones, la disartria puede ocurrir acompañada de otros problemas del lenguaje o el habla, tales como la apraxia o la afasia.

2.2.2.1 Clasificación de la disartria

La disartria puede clasificarse considerando distintos criterios, como los que se enlistan a continuación [3]:

- Según el curso que la disartria presente, es posible clasificarla como:
  - Regresiva, es decir, que puede disminuir la severidad del trastorno. Una disartria regresiva puede darse cuando la causa es un accidente cerebrovascular (ACV).
  - Estable, se da en casos de parálisis cerebral en adultos.
  - Progresiva, es decir, que en la mayoría de los casos el trastorno empeora. Este tipo de disartria se da en casos de esclerosis lateral amiotrófica (ELA), enfermedad de Parkinson, entre otros.
  - Fluctuante, se da en algunos casos de esclerosis múltiple.

- Según la severidad del trastorno, la disartria se puede clasificar en leve, moderada o severa.
- Por las características sintomatológicas, la disartria se divide en:
  - Disartria flácida: este tipo de disartria se puede dar debido a un ACV, ELA, tumores del sistema nervioso central, entre otros. Los pacientes con este tipo de disartria pueden tener características en su habla tales como voz soplada, hipernasalidad y falta de precisión en sonidos consonánticos.
  - Disartria espástica: dentro de sus causas están los ACV, traumatismos encéfalocraneanos (TEC) y enfermedades degenerativas. La voz de pacientes con disartria espástica puede ser forzada, lenta, hipernasal y con falta de precisión en sonidos consonánticos.
  - Disartria atáxica: sus causas más comunes son los ACV, TEC, cerebilitis, entre otros. Se pueden presentar características en el habla como: falta de precisión en sonidos consonánticos y acentuación marcada e igual para cada sílaba
  - Disartria hipocinética: se da comúnmente por enfermedad de Parkinson. Se presentan características como la falta de acentuación, tonalidad uniforme (monotonalidad), falta de variaciones normales en la intensidad (monointensidad) y disminución en la intensidad de la voz (hipofonía).
  - Disartria hipercinética: se presentan movimientos involuntarios y, según su velocidad, se clasifica en:
    - Disartria hipercinética predominantemente rápida: en este tipo de disartria existen movimientos involuntarios rápidos. Dentro de sus causas se tiene el síndorme de Gilles de la Tourette. En este caso, el habla presenta una velocidad variable y falta de precisión en sonidos consonánticos.
    - Disartria hipercinética predominantemente lenta: en esta disartria existen movimientos involuntarios lentos. Se presenta en casos de atetosis, discinesia tardía, entre otros. Dentro de las características en el habla se tiene: falta de precisión en sonidos consonánticos, voz ronca y con sonido forzado, además de monointensidad y monotonalidad.
  - Disartrias mixtas: se pueden presentar casos que son una combinación de los tipos de disartria mencionados anteriormente como, por ejemplo: disartria

mixta espástica-fláccida, espástica-atáxica-fláccida y espástica-atáxica-hipocinética.

En la Figura 2 se muestra de manera general las distintas clasificaciones de la disartria mencionadas anteriormente.

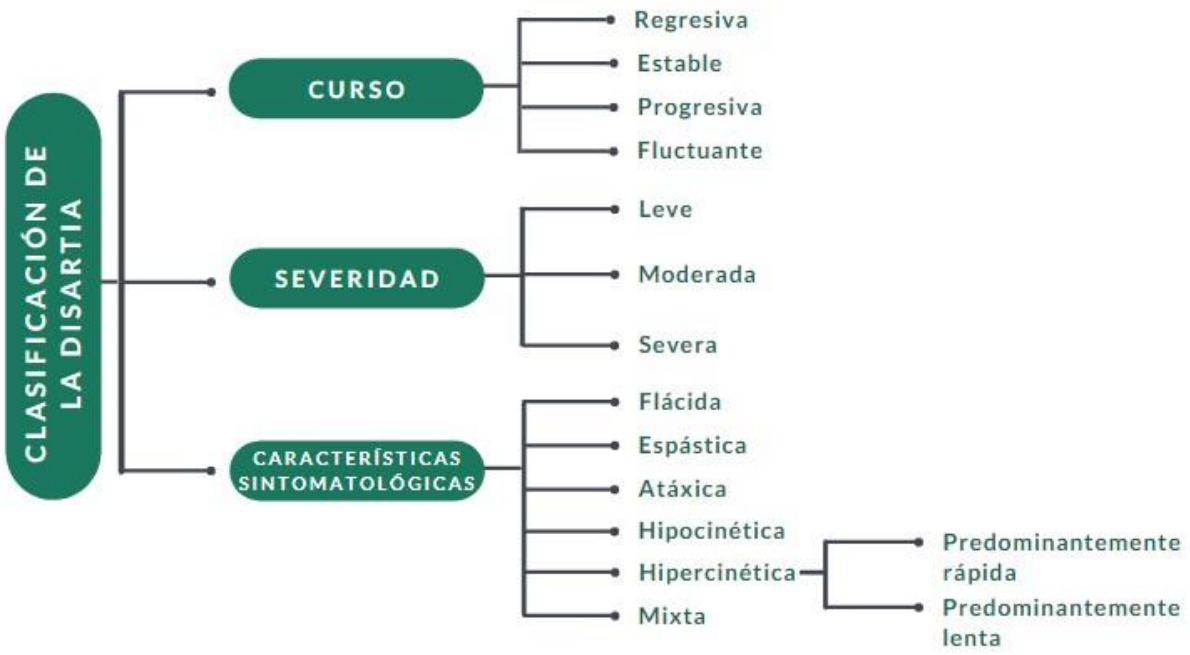

Figura 2. Clasificación de la disartria. Elaboración propia.

2.2.2.2 Características en el habla de personas con disartria

El habla de las personas con disartria se caracteriza por los siguientes síntomas en la pronunciación de fonemas [19]:

- Sustitución: cuando un fonema es reemplazado por otro.
- Omisión: cuando un fonema es omitido, por ejemplo, "uez" es pronunciado en lugar de "nuez".

- Inserción: un fonema que no forma parte de la palabra es insertado como apoyo para la pronunciación de un fonema que es difícil de pronunciar, por ejemplo, "Enerique" es pronunciado en lugar de "Enrique".
- Distorsión: se pronuncia un sonido que no coincide con el fonema a pronunciar, pero se acerca al sonido correcto.

2.3 Inteligencia Artificial

De manera general, la Inteligencia Artificial (IA) busca sintetizar y automatizar tareas intelectuales, por lo que es una herramienta relevante para diversas áreas en donde es necesario el intelecto humano [20]. A pesar de que la IA es una de las ciencias más recientes, en la actualidad es utilizada para distintas tareas como: el desarrollo de asistentes de voz, el análisis de imágenes médicas y la automatización del servicio al cliente, entre otras.

Para definir la IA, ciertos autores se enfocan en la similitud de los sistemas con los humanos, mientras que otros se enfocan en su capacidad de hacer lo correcto tomando en cuenta su conocimiento; es decir, se centran en el aspecto racional de los sistemas. De la misma manera, cada uno de los enfoques mencionados se centra en procesos mentales o en la conducta [20], [21]. Tomando en consideración el enfoque centrado en el humano, la IA puede ser definida como la habilidad de imitar las capacidades de la mente humana, como el aprender de los ejemplos y la experiencia, reconocer objetos, tomar decisiones y resolver problemas [22]. También centrándose en el humano, Elaine Rich define la IA como "el estudio de cómo hacer que los ordenadores hagan cosas que, por ahora, los humanos hacemos mejor" [21]. Por otro lado, considerando el enfoque de la racionalidad, la IA puede ser vista como "el estudio de los cálculos que hacen posible percibir, razonar y actuar" [20].

2.3.1 Machine learning

De manera tradicional, si se desea resolver un problema utilizando una computadora, es necesario implementar un algoritmo el cual especifique, de manera detallada, todas las acciones que se tienen que llevar a cabo ante distintas situaciones. Sin embargo, en problemas con una gran cantidad de casos particulares imposibles de prever, esta tarea se vuelve complicada [21].

Con el aprendizaje automático (en inglés, *machine learning*) los sistemas son capaces de aprender por sí mismos a resolver un problema. Para ello, aprenden de los datos a los que tienen acceso, en lugar de ser explícitamente programados para llevar a cabo la resolución del problema [23].

En la actualidad, el aprendizaje automático tiene diversas aplicaciones en la vida diaria, como la clasificación automática de correos electrónicos, separando los correos basura de aquellos que no lo son, así como la identificación de canciones utilizando pequeños fragmentos de ellas. Algunas otras aplicaciones para las que el aprendizaje automático es útil, es el reconocimiento de voz o la identificación de objetos en imágenes, ya que, para realizar este tipo de tareas, es necesario tener en cuenta diversos factores que dan lugar a distintas situaciones imposibles de especificar de antemano [21].

Existen dos subconjuntos de aprendizaje automático: el aprendizaje supervisado y no supervisado. En el primero, los algoritmos utilizan datos etiquetados previamente, para conocer cómo tendrá que ser categorizada la información nueva. Los algoritmos no supervisados, de manera contraria, no utilizan datos etiquetados, ya que su objetivo es el identificar maneras de clasificarlos para, por ejemplo, tener mejor entendimiento de las correlaciones que existen en los datos [24], [25].

2.3.2 Redes neuronales

Una red neuronal es un modelo con estructura semejante a la estructura de neuronas interconectadas en el cerebro [9]. La red neuronal consiste en un conjunto de unidades elementales, también denominadas neuronas, las cuales se encuentran conectadas entre sí de una manera concreta y se organizan en grupos denominados capas. En las redes neuronales existe una capa de entrada, en la cual se presentan los datos a la red neuronal, y una capa de salida, que brinda una respuesta ante una entrada. Las capas intermedias se denominan capas ocultas [10]. Esto se observa en la Figura 3.

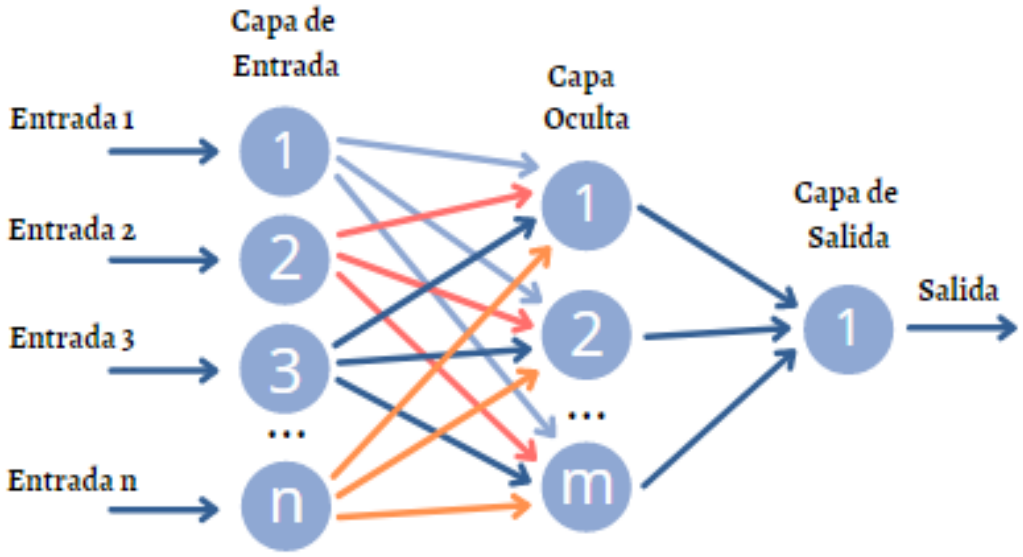

Figura 3. Arquitectura de red neuronal. Adaptación de [26].

Las redes neuronales se dividen en superficiales y profundas, las primeras consisten en dos o tres capas de neuronas conectadas, mientras que las redes profundas consisten de varias capas, incluso cientos de ellas [9].

El comportamiento de una red neuronal depende de la manera en la que se encuentran conectadas sus neuronas, así como de la ponderación de sus conexiones. Durante el entrenamiento de la red neuronal, las ponderaciones se ajustan de

manera automática de acuerdo con una regla de aprendizaje, hasta que el modelo sea capaz de realizar la tarea de una manera correcta [9].

2.3.2.1 Redes neuronales convolucionales

Una red neuronal convolucional (CNN, por sus siglas en inglés de *Convolutional Neural Network*) es una arquitectura de red capaz de aprender directamente de los datos; por lo que, al utilizarlas, no es necesario extraer de manera manual las características de los datos.

Las CNN, son capaces de procesar datos que tienen una topología en forma de cuadrícula. Por ejemplo, los datos de series en el tiempo, pueden ser considerados como una cuadrícula de una dimensión que toma muestras a intervalos de tiempo regulares; otro caso son los datos de imágenes, que se pueden considerar como una cuadrícula de pixeles en dos dimensiones [27]. El nombre de "redes neuronales convolucionales" indica que la red utiliza una operación matemática denominada convolución en al menos una de sus capas [27], [28].

Este tipo de arquitectura consta de múltiples capas; en donde se encuentran operaciones como la convolución, la no-linealidad y el *pooling* [28]. En una CNN es posible tener varias capas con estas operaciones en serie. Al final de la red neuronal, se tiene una capa completamente conectada (en inglés, *fully connected layer*) que genera la salida [29]. La arquitectura de una CNN puede observarse en la Figura 4.

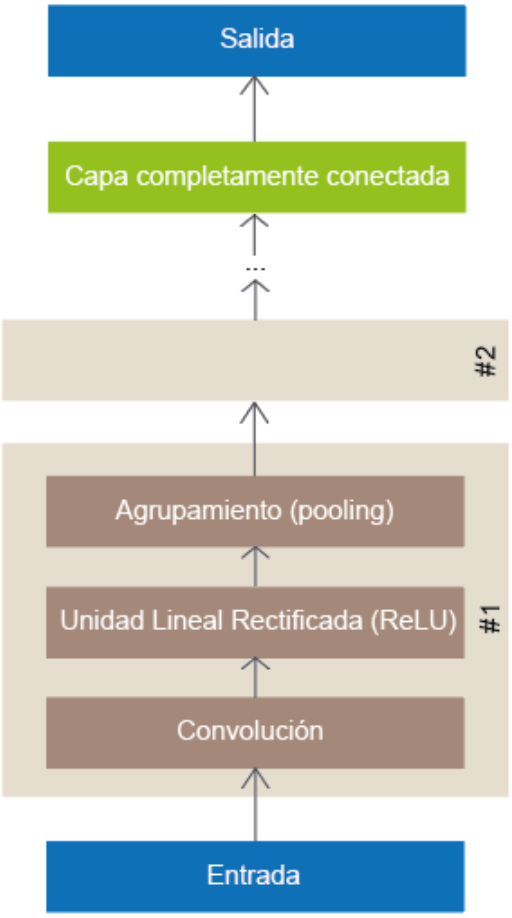

Figura 4. Arquitectura de una red neuronal convolucional. Adaptado de [29].

Convolución

La convolución es un tipo de operación lineal que, de manera general, se trata de una operación de dos funciones, $x(t)$ y $w(t)$, esto se muestra en (1). Típicamente, la operación de convolución se denota con un asterisco [27].

$$s(t) = (x * w)(t) \tag{1}$$

En las CNN, $x(t)$ indicaría la entrada y $w(t)$ el kernel. A la salida, $s(t)$ se le denomina mapa de características [27].

No-linealidad

La capa posterior a la de convolución es la de no-linealidad, la cual puede ser utilizada para ajustar o cortar la salida generada. En la actualidad, la función de activación más utilizada es la Unidad Lineal Rectificada (ReLU, por sus siglas en inglés de *Rectified Linear Unit*) [27]. Ésta se encarga de rectificar la salida de la capa de convolución para convertir todos los valores negativos a cero [29]. En la Figura 5 se muestra graficada la función ReLU.

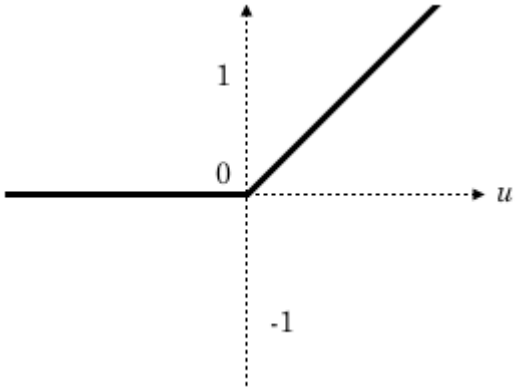

Figura 5. Función de activación ReLU [21].

Pooling

El *pooling* se encarga de realizar una reducción del volumen de datos, al reemplazar bloques de mayor tamaño con un solo valor. El método de agrupamiento más comúnmente utilizado es el llamado Max Pooling; en este método, el valor máximo del bloque se usa para reemplazar el bloque completo. El *pooling* reduce drásticamente la dimensionalidad de los datos que fluyen en la red mientras mantiene la información importante [29].

Dentro de las aplicaciones de las redes neuronales convolucionales se tiene la detección de células cancerosas en imágenes médicas; la conducción autónoma, con

la finalidad de detectar señales u otros objetos; y el procesamiento de audio, al ser capaces de detectar palabras claves independientemente del entorno [30].

2.3.2.2 Redes neuronales recurrentes

La característica de las redes neuronales tradicionales es que cada una de las entradas es procesada de manera independiente, sin que se mantenga ningún estado entre ellas. De manera contraria, nuestra inteligencia biológica nos permite procesar información de manera incremental, manteniendo un modelo interno de lo que está procesando, el cual se construye a partir de la información pasada y se actualiza al obtener nueva información [25].

Para lograr lo anterior, las redes neuronales recurrentes (RNN, por sus siglas en inglés de *Recurrent Neural Network*) son un tipo de red neuronal que contienen bucles, los cuales les permiten tener un tipo de "memoria" [31]. Esto se muestra en la Figura 6.

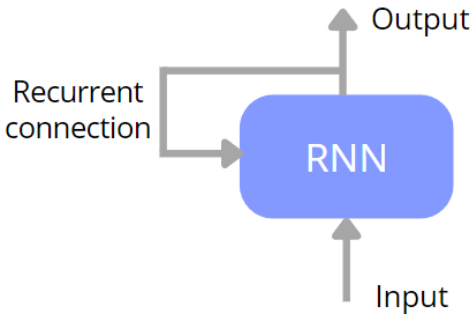

Figura 6. Conexión recurrente de una RNN [25].

Las RNNs procesan un elemento de una secuencia a la vez, manteniendo una especie de vector que contiene información sobre todos los elementos pasados de la secuencia [32]. Las RNNs pueden ser utilizadas para textos, el habla, series en el tiempo, o cualquier aplicación en la cual la ocurrencia de un elemento dependa de los elementos que aparecieron previamente [33]. La Figura 7 muestra la versión "desplegada" de una RNN.

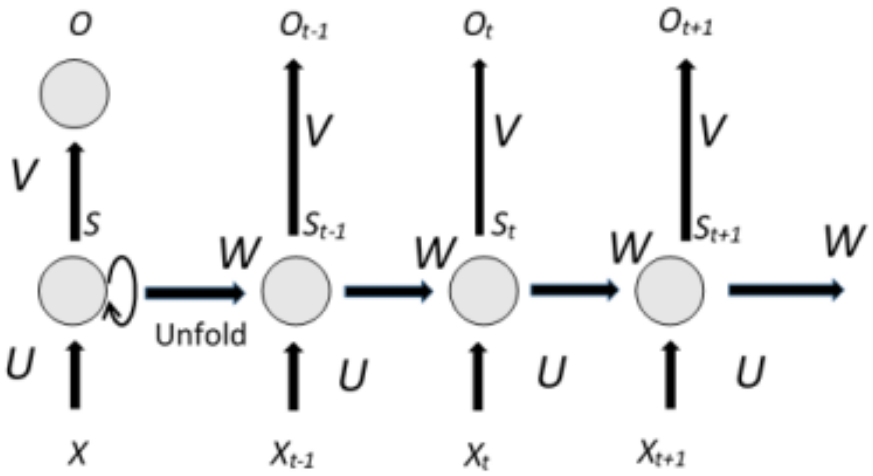

Figura 7. Versión "desplegada" de una RNN [32].

La Figura 7 muestra el aspecto de una RNN en su versión "desplegada", el cual explica la estructura de la red para toda una secuencia de entradas. En ese caso, *Xt* es la entrada de la red neuronal en el instante *t*, la cual puede ser, por ejemplo, un vector que representa la palabra de una oración. *St* es un vector que contiene información sobre todos los elementos anteriores de la secuencia. Para obtener *St*, se relacionan la entrada actual (tiempo *t*) y el parámetro *U*, y el estado evaluado en el instante anterior (tiempo $t - 1$) con el parámetro *W*. *Ot* es la salida en el instante *t*, la cual es calculada utilizando el parámetro *V* [32]. En algunos casos, la salida *Ot* puede ser una predicción de $Xt+1$; en otras ocasiones, *Ot* no es generada para cada momento, sino solo al final de la secuencia [34].

La Figura 8 presenta las distintas topologías de una RNN.

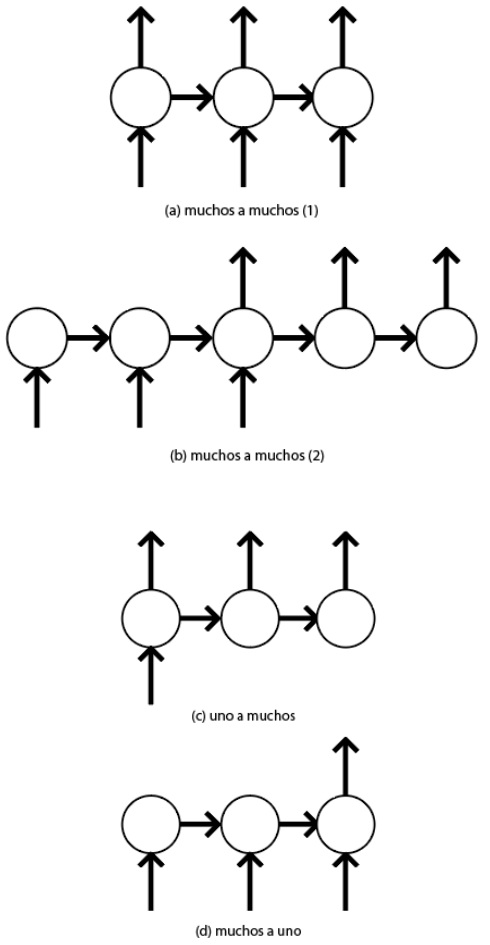

Figura 8. Topologías de una RNN. Adaptado de [33].

Una RNN puede presentar distintas topologías. Estas topologías se derivan de la estructura anteriormente mencionada de la Figura 7, la cual es considerada una topología de "muchos a muchos". Otro ejemplo de una RNN de "muchos a muchos" se observa en (b), la cual toma una secuencia como entrada y produce otra secuencia como salida; por ejemplo, la entrada puede ser una secuencia de palabras en inglés, las cuales conforman una oración, y la salida puede tratarse de palabras de la sentencia traducida al español [33].

Otra variante son la RNN de "una a muchas", mostrada en (c); ésta puede ser utilizada para, por ejemplo, subtitular imágenes. Siendo la entrada una imagen y la salida una

secuencia de palabras. En (d) se presenta la topología "muchas a una", utilizada para hacer análisis de sentimientos en las oraciones; donde la entrada es una secuencia de palabras, y la salida es un sentimiento positivo o negativo [33].

Como se mencionó anteriormente, las RNNs son efectivas para el procesamiento de secuencias, siendo especialmente efectivas para dependencias a corto plazo, ya que si la secuencia es larga, puede perderse información a largo plazo [35].

2.3.3 Transfer learning

En diversas situaciones, es útil adaptar un sistema efectivamente capacitado, con la finalidad de aplicarlo en un dominio similar en el que no se dispone de suficientes datos para el entrenamiento [29]. El aprendizaje por transferencia (*Transfer learning* por sus siglas en inglés) es un enfoque de aprendizaje profundo, en donde un modelo que ha sido entrenado para llevar a cabo una tarea, se utiliza como punto de partida para desarrollar un modelo que realice una tarea similar [36]. Su finalidad es la de transferir el conocimiento que una red neuronal ha adquirido en una tarea, para su aplicación en una nueva tarea. Dentro del contexto de las redes neuronales, por conocimiento se hace referencia a las características extraídas [37]. Esta técnica es comúnmente utilizada para aplicaciones de detección de objetos, reconocimiento de voz y reconocimiento de imágenes [36].

En diversos problemas del mundo real, no es posible obtener millones de datos para entrenar modelos complejos [37]; debido a lo anterior, el aprendizaje por transferencia es una técnica muy popular, ya que permite entrenar modelos con menos datos etiquetados al reutilizar modelos que ya han sido entrenados con grandes conjuntos de datos. Otras de las ventajas que ofrece es la posible reducción del tiempo de entrenamiento y de los recursos informáticos utilizados, ya que los pesos de la red neuronal no se aprenden desde cero [36].

Para la aplicación del aprendizaje por transferencia, es común el siguiente flujo de trabajo [38]:

1. Tomar capas de un modelo previamente entrenado.
2. Congelar las capas para evitar perder la información contenida durante los futuros entrenamientos.
3. Añadir capas nuevas frente a las capas congeladas, las cuales aprenderán a convertir las características antiguas en predicciones para el nuevo conjunto de datos.
4. Entrenar las nuevas capas con el nuevo conjunto de datos.
5. De manera opcional, se lleva a cabo un ajuste fino (*fine-tuning* en inglés). Esto consiste en descongelar una parte o todo el modelo obtenido, y reentrenarlo con los nuevos datos a una tasa baja de aprendizaje. Tiene como finalidad adaptar lo preentrando a los nuevos datos.

Además de aprovechar los pesos de una red preentrenada para inicializar los de nuestra red neuronal y hacer un ajuste parcial o total de estos, como en el caso previamente explicado; también es posible utilizar un modelo como extractor de características. En este caso, se obtienen las salidas de las capas de la red entrenada y éstas son tomadas como entrada para nuestra red. También es posible usar estas características extraídas para entrenar otro método de aprendizaje automático como, por ejemplo, las máquinas de vectores de soporte (SVM por sus siglas en inglés) [21].

Andrej Karpathy, director de inteligencia artificial de la empresa Tesla, explica que, el método de aprendizaje por transferencia a seleccionar, depende del tamaño del conjunto de datos del que se dispone, además de la similitud de éste con el conjunto de datos que fue utilizado para entrenar el modelo, del que se desea aprovechar sus pesos [21]. Según Karpathy [21]:

- Si el conjunto de datos es pequeño y similar al utilizado para entrenar al modelo, se recomienda extraer características del modelo previamente entrenado y, con ello, entrenar a nuestro modelo.

- Si el conjunto de datos es pequeño y diferente al utilizado para el entrenamiento, no se recomienda utilizar transfer learning. En este caso, la única opción, es aprovechar la salida de las primeras capas del modelo previamente entrenado.
- Si el conjunto es grande y similar al utilizado, es posible ajustar los parámetros del modelo preentrenado.
- Si el conjunto es grande y diferente al empleado, no es necesario utilizar transfer learning. Se recomienda entrenar una red desde cero. Sin embargo, es posible aprovechar los pesos del modelo previamente entrenado para no inicializar los pesos de nuestra desde cero.

Existen diversos modelos preentrenados disponibles para su uso, al seleccionar alguno de ellos, es importante considerar factores como el tamaño y la velocidad de predicción. La importancia del tamaño del modelo varía dependiendo de dónde y cómo se desee implementar; el tamaño de la red es relevante principalmente cuando se busca utilizar el modelo en un sistema con poca memoria. La velocidad de una predicción puede variar por el hardware utilizado, así como de la arquitectura del modelo elegido y su tamaño [36].

2.4 Reconocimiento automático del habla (RAH)

Existe una diferencia entre el reconocimiento del habla y la comprensión del habla: mientras que el último se refiere a la habilidad de entender el significado de lo que se dice más que su transcripción, el reconocimiento del habla se refiere a la habilidad de un sistema para reconocer las palabras que se dicen [39].

El reconocimiento automático del habla (ASR por sus siglas en inglés de *automatic speech recognition*), consiste en transcribir el contenido de una señal de voz sin que una persona intervenga [40]. Es decir, el ASR es un proceso independiente en el que se decodifica y transcribe el habla oral. Típicamente, un sistema de este tipo recibe una señal acústica como entrada, la analiza utilizando algún patrón, modelo o algoritmo, y produce una señal de salida generalmente en forma de texto [39].

2.4.1 Clasificación de los sistemas para el RAH

Es posible clasificar a los sistemas para el reconocimiento automático del habla al considerar características como el tipo de habla que son capaces de reconocer y la dependencia del hablante. Según el tipo de habla a reconocer, los sistemas se dividen en [39]:

- Sistemas para el reconocimiento de palabras aisladas: se encargan de reconocer palabras pronunciadas de forma aislada. En este tipo de sistemas se requiere que el usuario inserte pausas entre las palabras con la finalidad de que sea posible identificar su comienzo y final.
- Sistemas para el reconocimiento de palabras conectadas: son capaces de reconocer palabras aisladas que son pronunciadas sin pausas entre ellas.
- Sistemas para el reconocimiento de habla continua: reconocen frases completas sin la necesidad de insertar pausas entre las palabras. Sin embargo, en este tipo de sistemas es necesario insertar pausas entre frases para identificar su inicio y término, además de procesar la secuencia de palabras contenidas [41].
- Sistemas de detección de palabras: extrae palabras o frases de un habla continua.

Considerando la dependencia del sistema al hablante, los sistemas se clasifican en [39]:

- Dependientes del hablante: el sistema debe estar entrenado para cada usuario.
- Independientes del hablante: se utilizan diversos ejemplos de habla de diferentes personas para que el sistema pueda reconocer a cualquier usuario. En este caso, el sistema no requiere de su entrenamiento o adaptación para el reconocimiento del habla de un usuario en particular.
- Adaptativo: el sistema comienza como independiente del hablante y, conforme se utiliza, se adapta a un usuario en particular [39]. Este tipo de sistemas están diseñados para ajustarse al usuario sin la necesidad de entrenar cada una de las palabras que conforman el vocabulario a reconocer [41].

2.4.2 Funcionamiento básico de los sistemas reconocedores

La mayoría de los sistemas para el reconocimiento automático del habla se realiza un proceso en donde se lleva a cabo el procesamiento de la señal, la extracción de características y la descodificación [40]. Esto se muestra en la Figura 9.

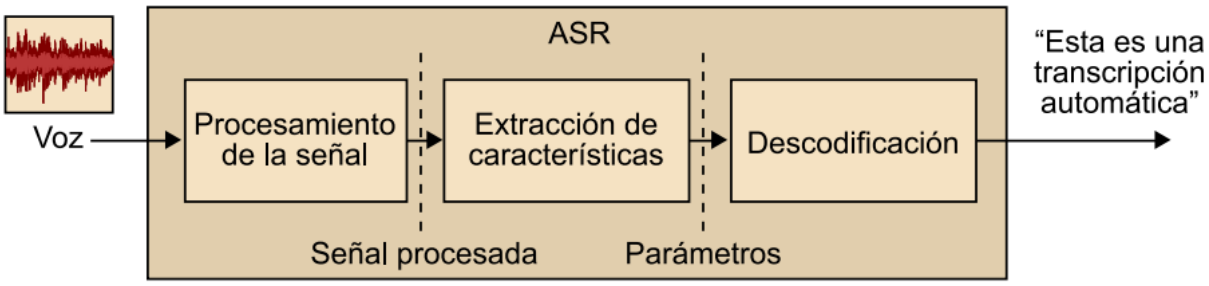

Figura 9. Diagrama de un sistema para el reconocimiento automático del habla [40].

Durante el procesamiento de la señal, el audio de entrada se transforma para su reconocimiento. Generalmente, en esta etapa, se realizan técnicas para la detección de la voz y del ruido para su posterior reducción. Los segmentos de audio obtenidos que corresponden a la voz son utilizados en las etapas posteriores [40].

En la etapa de extracción de características, se obtiene un vector de características de cada uno de los segmentos de voz. La salida de esta etapa es una secuencia de vectores de características que contienen la información necesaria para realizar el reconocimiento. Finalmente, en la etapa de descodificación, se obtiene la transcripción que con más probabilidad se deduce que se ha pronunciado a partir de los vectores de características [40].

2.5 Extracción de características en señales de audio para su uso en el aprendizaje profundo

Al trabajar con señales de audio, es común extraer características de ellas. El tipo de características a obtener dependerá de su aplicación; ya que éstas pueden capturar diferentes aspectos del sonido. Por ejemplo, para aplicaciones relacionadas al reconocimiento del habla, es común el procesamiento de la señal para obtener el Espectrograma, el Espectrograma de Mel o los Coeficientes Cepstrales en la Frecuencia de Mel.

2.5.1 Espectrograma

El descomponer una señal de audio en las diferentes frecuencias que la componen, es una práctica útil para extraer información relevante. Para ello, se utiliza la Transformada de Fourier, en donde pasamos una señal del dominio del tiempo al dominio de la frecuencia [42]. Se puede observar una imagen representativa en la Figura 10.

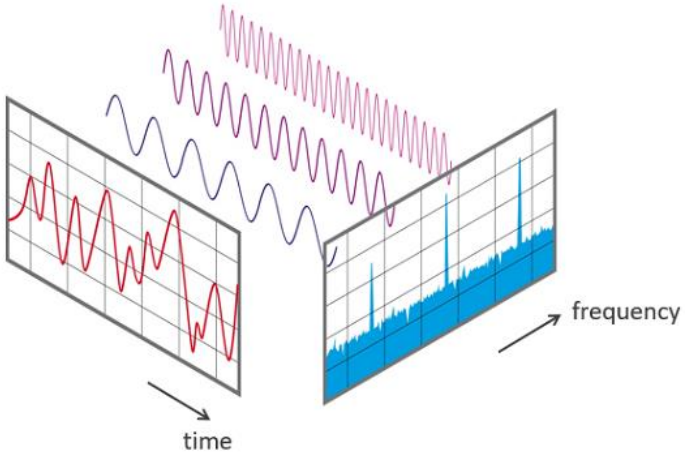

Figura 10. Señal en el dominio del tiempo vs en el dominio de la frecuencia [42].

Sin embargo, en señales no periódicas como el habla, resulta útil conocer la manera en la que las frecuencias varían en el tiempo; a diferencia del ejemplo pasado, en donde se obtienen las frecuencias que componen a nuestra señal de una manera general. Para ello, se utiliza la Transformada de Fourier de Tiempo Reducido (*Short-time Fourier transform o* STFT, por sus siglas en inglés). En este caso, la Transformada Rápida de Fourier es calculada de segmento a segmento en la señal, utilizando ventanas que se superponen entre sí. Esto da como resultado el espectrograma de la señal [43].

El espectrograma es una representación gráfica de un espectro de frecuencias de una señal variable en el tiempo. Es utilizado para caracterizar el contenido de una señal del habla o una musical; también se utiliza para el análisis de señales del cuerpo, por ejemplo, el electrocardiograma o el electroencefalograma [44].

Los espectrogramas se componen de tres dimensiones. En el eje *y* se representan, partiendo de abajo hacia arriba, las frecuencias de menor a mayor. En el eje *x* se muestra el tiempo de menor a mayor, de izquierda a derecha. La tercera dimensión es representada por medio de colores en el gráfico, y muestran la amplitud de una frecuencia particular en un momento dado. Por ejemplo, los colores oscuros corresponden a amplitudes bajas, mientras que colores más brillantes indican la presencia de amplitudes más altas [45], [46]. En la Figura 11 se muestra como ejemplo un espectrograma con las características mencionadas.

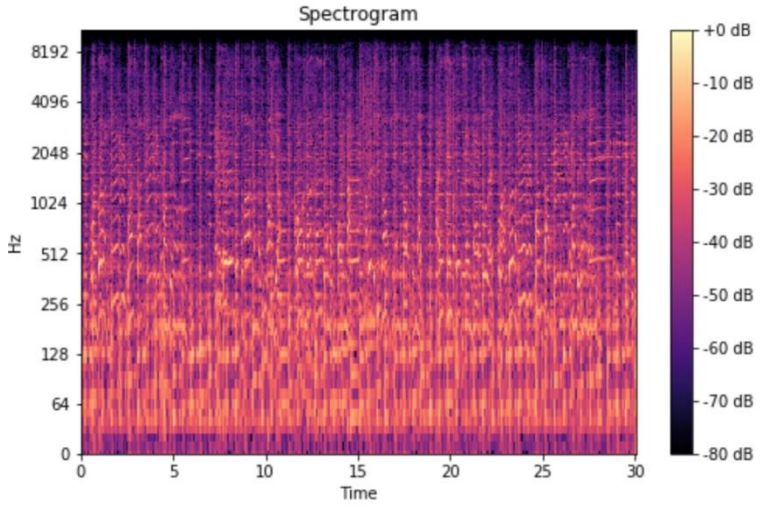

Figura 11. Espectrograma [43].

2.5.2 Espectrograma de Mel

Los humanos no percibimos las frecuencias de manera lineal, sino de manera logarítmica. Para obtener una representación de la manera en la que percibimos las frecuencias, se desarrolló la escala de Mel. Ésta se realizó mediante experimentaciones con diversas personas [47]. La escala se presenta en la Figura 12, en donde se visualiza la relación de Hertz con la escala de Mel. La gráfica presentada en la Figura nos indica que somos más sensibles a las diferencias entre frecuencias bajas que a las diferencias entre frecuencias más altas.

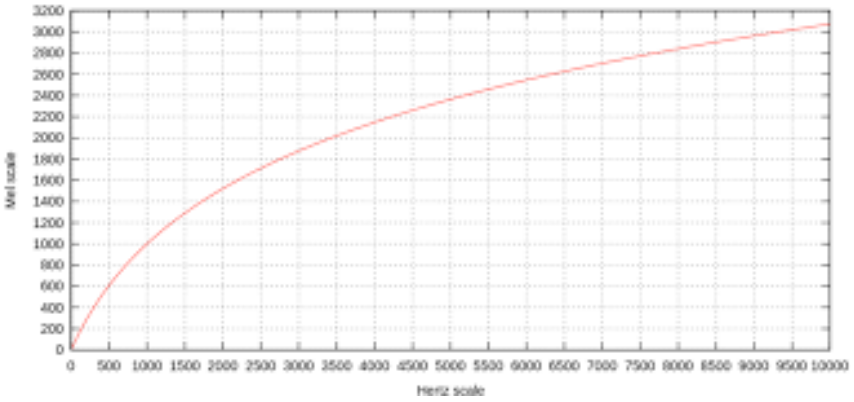

Figura 12. Escala de Mel [48].

A diferencia del Espectrograma, en donde se representan las frecuencias en Hertz, el Espectrograma de Mel, como su nombre lo indica, presenta las frecuencias utilizando la escala de Mel; encontrándose, típicamente, en el eje *y* de la gráfica. Mientras que en el eje *x* se representa el tiempo. Al igual que en el Espectrograma, se representan las amplitudes en decibeles y por medio de colores [43]. En La Figura 13 se muestra un ejemplo de un Espectrograma de Mel.

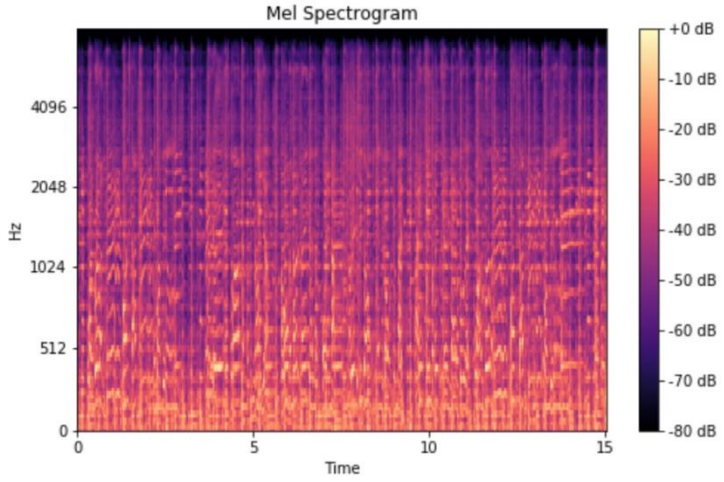

Figura 13. Espectrograma de Mel [43].

2.6 TensorFlow

TensorFlow comenzó en 2011 como DisBelief, el cual era un proyecto interno, closed-source de Google, usado para trabajar con redes neuronales de aprendizaje profundo. Posteriormente, este proyecto se transformó en TensorFlow y fue lanzado para su libre uso en noviembre de 2015 [49].

TensorFlow es un librería de código abierto, desarrollada por el equipo de investigación Google Brain, la cual es utilizada para implementar sistemas de aprendizaje automático y aprendizaje profundo [32]. La Librería está implementada en C++ [50]. Dispone de APIs en distintos lenguajes para la construcción y ejecución de gráficos de TensorFlow; sin embargo, la API de Python es actualmente la más completa y fácil de utilizar [51]. Dentro de las aplicaciones de TensorFlow se tiene el reconocimiento de voz y sonido; aplicaciones basadas en texto, tales como traductores de idiomas; el reconocimiento de imágenes, como la búsqueda de exoplanetas y la detección de cáncer; y aplicaciones de series en el tiempo, como los sistemas de recomendación [49].

Generalmente, un programa de TensorFlow consiste en dos secciones: la construcción de un grafo computacional y la ejecución del grafo computacional. El gráfico se compone de aristas y nodos [52]:

- Las aristas representan los datos en forma de tensor (por ejemplo, un vector, una matriz o una matriz de datos de mayor dimensión), el cual fluye a través del grafo. De lo anterior surge el nombre de la librería, TensorFlow.
- Los nodos son denominados operaciones que representan cálculos en los tensores; por ejemplo, suma y multiplicación. Una operación toma cero o más tensores como entrada y produce cero o más tensores como salida.

Dentro de las principales características de TensorFlow, se encuentra su capacidad para optimizar y calcular de manera eficiente expresiones matemáticas que involucran matrices multidimensionales. Otra de sus características es el brindar la

capacidad de escribir un código y poder ejecutarlo ya sea en el CPU o en la GPU, TensorFlow es capaz de determinar las partes del código que se ejecutarán en la GPU [32].

2.7 Trabajos relacionados

En 2018, se publicó un artículo por parte de la Universidad de Pisa, en el cual se explica el desarrollo de un modelo enfocado al reconocimiento automático del habla, para su aplicación en personas con disartria espástica de leve a severa. El objetivo de la investigación se limitaba al reconocimiento de 12 palabras cortas en italiano como "*volume*", "*uscita*" y "*tappa*" y, para ello, se desarrolló una red neuronal convolucional. Para el entrenamiento del modelo, se utilizaron 1000 contribuciones de 3 hombres con disartria y diferentes inteligibilidades. Para la evaluación de la red neuronal, se utilizaron 100 archivos de sonido con una duración de 2500 milisegundos cada uno. El porcentaje de reconocimiento obtenido por el modelo desarrollado fue de un 57.5% [11].

En otra investigación publicada en 2020, se desarrolló una red neuronal recurrente-convolucional, para su aplicación en el reconocimiento del habla para personas con disartria. El modelo desarrollado se enfocó en el reconocimiento de 16 palabras en inglés, tales como "*sigh*", "*air*", "*beat*" y "*spark*". Para el entrenamiento y evaluación del modelo se utilizaron archivos de audio provenientes del *dataset* TORGO, el cual contiene audios de personas con disartria y sin disartria, mencionando distintas frases o palabras. Las palabras seleccionadas para la investigación contaban con 30 a 50 clips cada una. Para el entrenamiento del modelo, el 80% de los datos fue utilizado, mientras que el porcentaje restante fue utilizado para la evaluación. Los resultados de la investigación indican que la red neuronal recurrente-convolucional obtuvo un 40.6% de reconocimiento frente a un 31.4% de una red neuronal convolucional desarrollada para su comparación bajo las mismas condiciones [7].

En la actualidad, se han desarrollado aplicaciones móviles para asistir la comunicación oral de personas con disartria, al reconocer su habla y traducir su discurso por medio de generadores de voz. Un ejemplo de ello es la aplicación Voiceitt, la cual requiere de un entrenamiento previo al reconocimiento de frases. El entrenamiento consiste en la repetición de cada una de las frases que se desea que se reconozcan, de aproximadamente 30 veces; la interfaz de entrenamiento se muestra en la Figura 14. Actualmente, la aplicación no se encuentra disponible para ser descargada. Voiceitt tendrá un costo de 200 dólares por año [53]. La Figura 14 muestra la pantalla de entrenamiento de la aplicación móvil Voiceitt

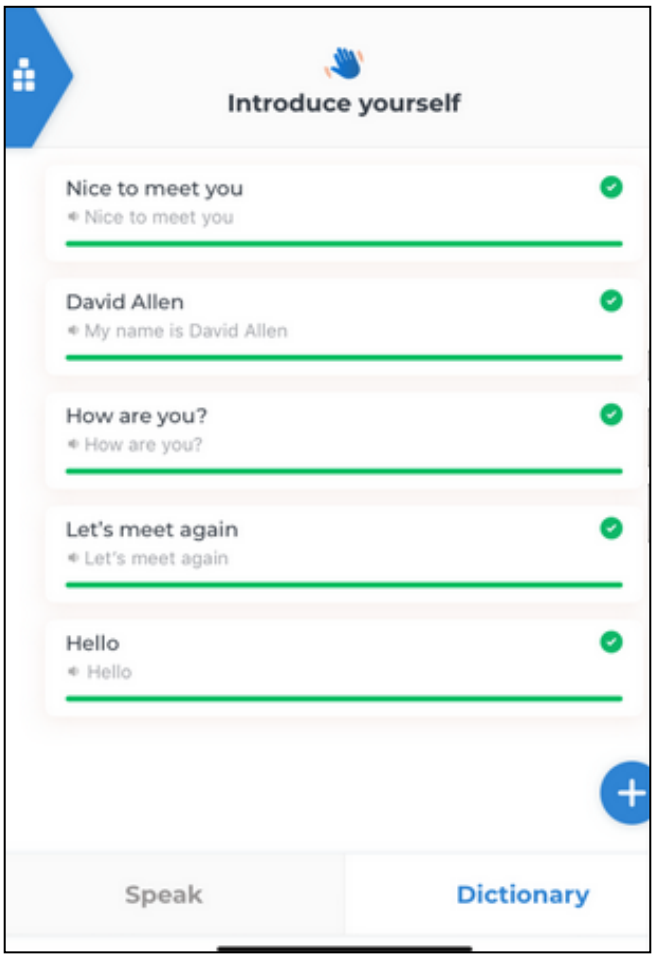

Figura 14. Pantalla de entrenamiento de la aplicación móvil Voiceitt [53].

Otra aplicación móvil desarrollada para asistir la comunicación oral de personas con este trastorno, es VocaTempo. La interfaz gráfica de VocaTempo, mostrada en la Figura 15, consta de distintas celdas que contienen frases e imágenes, las cuales es posible seleccionar de manera manual o por medio de la voz. Al seleccionar una celda, la aplicación emite una salida de voz con lo que la persona desea comunicar. Para que VocaTempo reconozca los patrones de voz del usuario, es necesario grabar 8 veces cada una de las frases que se desea que reconozca. Actualmente la aplicación se encuentra disponible para iOs y Android a un precio actual de 174 dólares [54].

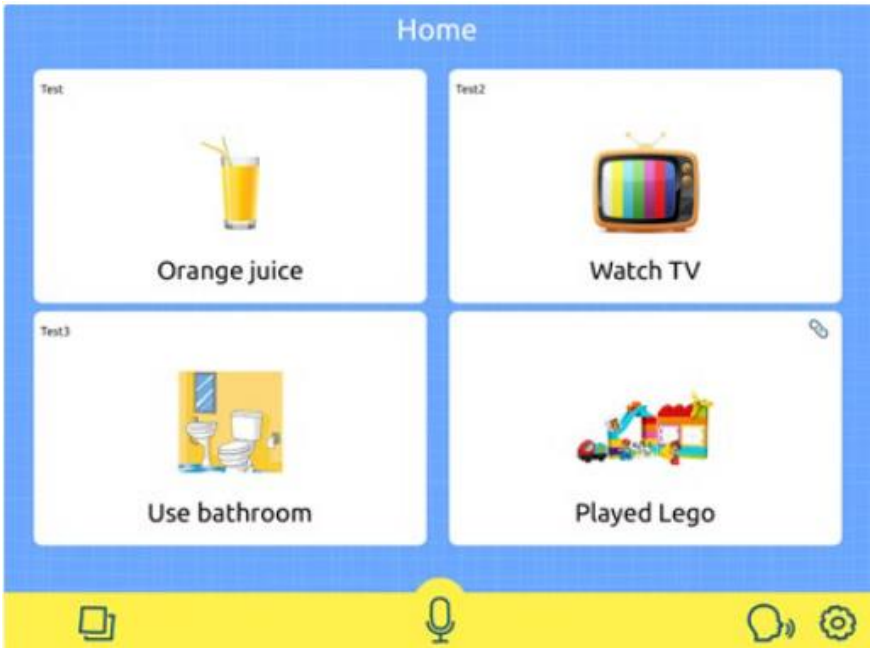

Figura 15. Interfaz gráfica de aplicación móvil VocaTempo [54].

Capítulo III: Análisis y diseño del sistema

3.1 Introducción

En este capítulo se realiza una descripción de la metodología a implementar para el desarrollo del sistema (sección 3.1); también se presentan los diagramas de análisis y diseño del sistema (sección 3.2 y 3.3, respectivamente), el modelo de datos (sección 3.4) y, por último, la arquitectura del sistema propuesto (sección 3.5).

3.2 Metodología

El proceso para el desarrollo del sistema considera desde el estudio de los conceptos relevantes para el proyecto y de los trabajos relacionados, hasta la implementación del sistema, las pruebas de funcionalidad y usabilidad y el análisis de resultados. La metodología consiste en tres etapas, las cuales se pueden visualizar en la Figura 16.

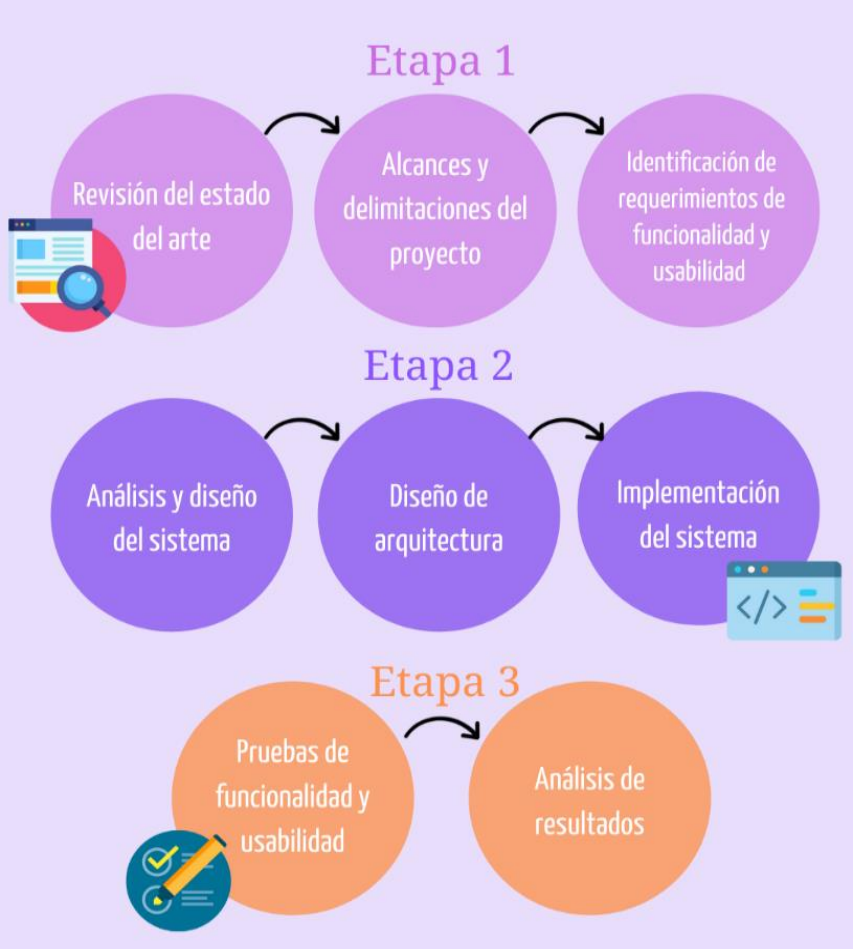

Figura 16. Metodología del proyecto.

El primer apartado de la etapa 1 (etapa 1.1) se compone de la revisión de los temas fundamentales del proyecto. Por ejemplo, la disartria y sus tipos, la inteligencia artificial y las diferentes técnicas para su aplicación; también, esta etapa considera el análisis de las propuestas existentes. Al tener un amplio conocimiento del tema, de las distintas técnicas disponibles a implementar y de lo desarrollado anteriormente por otros investigadores, es posible determinar los alcances y delimitaciones del proyecto (etapa 1.2). Al tener identificado el enfoque del proyecto, es posible identificar los requerimientos de usabilidad y funcionalidad del sistema (etapa 1.3).

Una vez definido el sistema de manera general, es posible realizar un análisis más específico que ayude a establecer pautas para el desarrollo del sistema; para ello, se realizan los diagramas de análisis diseño del sistema (etapa 2.1). Con ello, se da inicio a la segunda etapa de la metodología. En la etapa 2.2, se considera el diseño de la arquitectura, la cual es desarrollada con ayuda de los diagramas realizados en el punto anterior. Al definir la arquitectura del sistema, es posible comenzar la implementación del sistema (etapa 2.3). Finalmente, se tiene la tercera etapa, la cual consiste en las pruebas de funcionalidad y usabilidad del sistema (etapa 3.1) y, posteriormente, el análisis de los resultados obtenidos a partir de ellas (etapa 3.2).

3.3 Diagramas del análisis del sistema

3.3.1 Diagrama de contexto: nivel 0

El diagrama de contexto de nivel 0 proporciona una vista general del sistema y las entidades externas con las que se relaciona. Esto se presenta en la Figura 17.

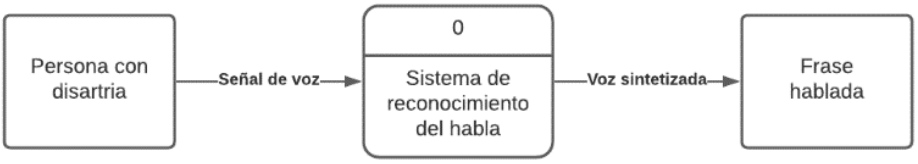

Figura 17. Diagrama de contexto: nivel 0.

En este caso, el sistema recibe una señal de audio que es proporcionada por la persona con disartria. El sistema procesa la señal y, como salida, se tiene la frase o palabra relacionada con la entrada. Dicha frase o palabra será emitida por medio de un sintetizador de voz.

3.3.2 Diagrama de nivel superior: nivel 1

En el diagrama de nivel superior, nivel 1, se describe de manera más específica el sistema; este puede observarse en la Figura 18. Como en la Figura se indica, primero,

el sistema recibe la señal de audio de la persona con disartria. Posterior a ello, se realiza un preprocesamiento de audio en donde se considera, por ejemplo, la extracción de características que serán utilizadas para el entrenamiento de la red neuronal. Las características serán almacenadas en una base de datos, de donde serán obtenidas para el entrenamiento del modelo. Posterior a ello, se llevará a cabo una validación del entrenamiento y con ello, el modelo será capaz de realizar inferencias.

La salida de la red neuronal está ligada a una frase o palabra que se encuentra almacenada en la base de datos. Una vez que se obtenga la inferencia de la red neuronal, se identificará la frase o palabra a la cual se encuentra ligada para que ésta sea desplegada.

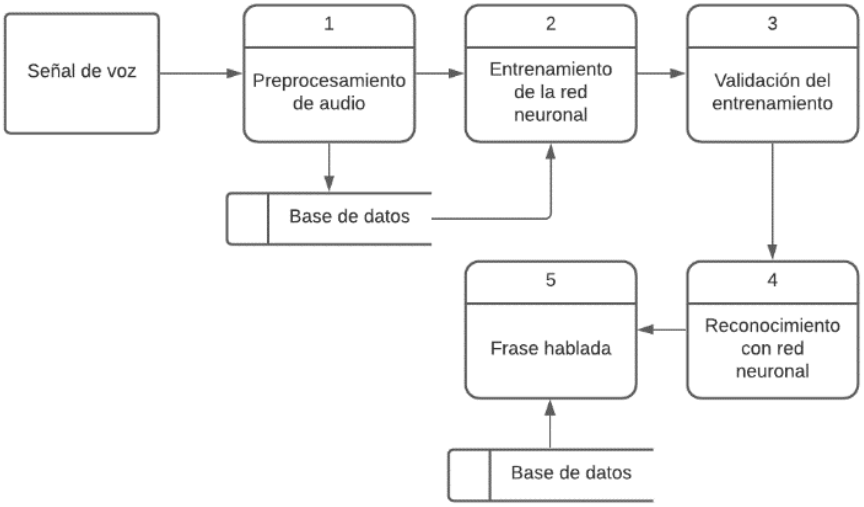

Figura 18. Diagrama de nivel superior: nivel 1.

3.3.3 Casos de uso

Un diagrama de casos de uso permite representar la manera en la que los actores interactúan con el sistema. El actor puede ser, por ejemplo, una persona u otro sistema. En este caso, se consideran dos actores: el usuario final, que es la persona con disartria, y el tutor. El usuario final es capaz de entrenar la red neuronal y de utilizarla. Para ello, el sistema debe permitir al usuario grabar los audios que serán la entrada a la red neuronal. También se considera la administración del catálogo de palabras como caso de uso. Por ejemplo: modificar, agregar o eliminar una palabra. Esta tarea puede ser realizada por el usuario final o por un tutor. Esto debido a que, si bien, el usuario final puede tener la capacidad de administrar el catálogo, en otros casos puede ser una tarea compleja para ellos. Ya sea porque el usuario final aún no adquiere la habilidad para leer y escribir o porque se le complica realizar tareas finas en un dispositivo móvil, debido a algún tipo de discapacidad motriz. El diagrama de casos de uso se muestra en la Figura 19.

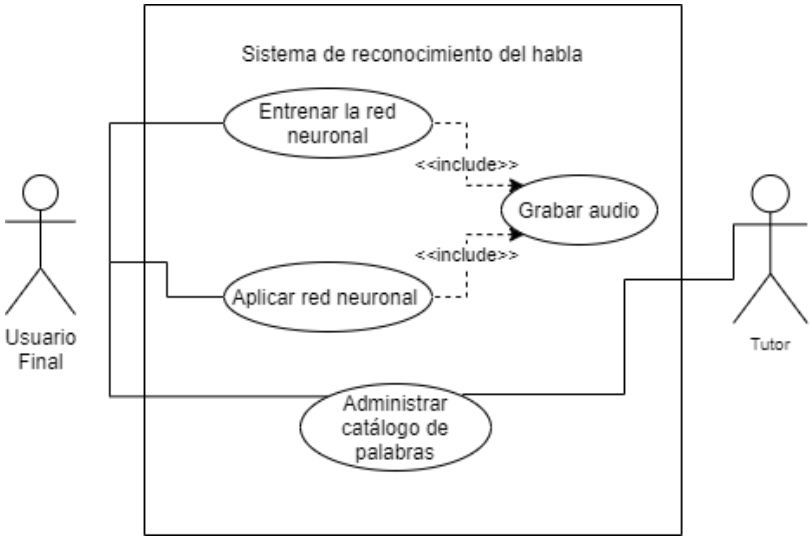

Figura 19. Casos de uso.

3.4 Diagramas del diseño del sistema

3.4.1 Diagrama de clases

Es posible visualizar las distintas clases que componen a un sistema por medio del diagrama de clases UML. En este diagrama, las clases se representan por medio de rectángulos que se dividen en tres partes de manera horizontal. La parte superior del rectángulo indica el nombre de la clase, la central muestra los diferentes atributos y, por último, la parte inferior indica los distintos métodos que la clase posee [55]. En la Figura 20 se muestra el diagrama de clases del sistema. El diagrama se compone de cuatro clases en total:

- **Principal:** muestra al usuario la pantalla principal
- **Catálogo:** con esta clase es posible insertar, modificar, eliminar y obtener un registro. En este caso, una instancia contendrá como atributos: su ID (idPalabra), la palabra que el usuario pronunciará (palabraIn) y la palabra o frase que el sistema emitirá (palabraOut).
- **Reconocimiento:** el propósito de esta clase es realizar el procesamiento de la señal, realizar la inferencia y reproducir el resultado obtenido.
- **Entrenamiento:** clase encargada de obtener las señales de audio con las que el modelo será entrenado. En esta clase se consideran actividades como el procesamiento de la señal, la reproducción del audio, eliminación del audio y el entrenamiento del modelo.

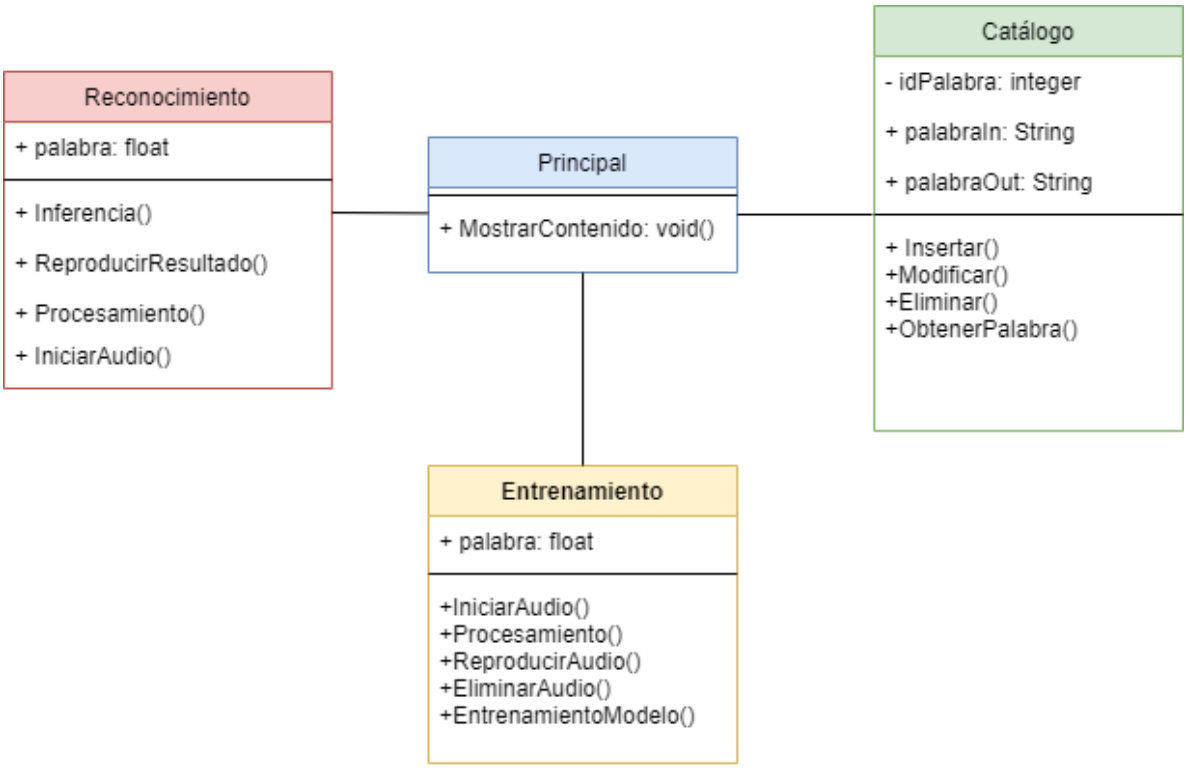

Figura 20. Diagrama de clases.

3.4.2 Diagrama de actividades

Para el sistema se considera un total de 5 actividades que muestran la interacción entre el sistema y el usuario. Dentro de las actividades se considera: dar de alta, modificar o eliminar registros, realizar el entrenamiento de palabras y su reconocimiento.

Añadir registro

La actividad comienza cuando el usuario selecciona la opción para añadir registro. Una vez seleccionado, el sistema muestra una pantalla con los datos requeridos para añadir un registro. El usuario llena el formulario. Posteriormente, el sistema verifica si los datos brindados por el usuario son correctos; por ejemplo, que no existan campos vacíos o que la palabra a reconocer no haya sido almacenada anteriormente.

En caso de que los datos sean incorrectos, el sistema le informa al usuario y le muestra nuevamente la pantalla para añadir un registro; en caso contrario, el sistema almacena los datos en la base de datos y le notifica al usuario cuando el proceso se haya terminado y se haya realizado correctamente. Esto se muestra en la Figura 21.

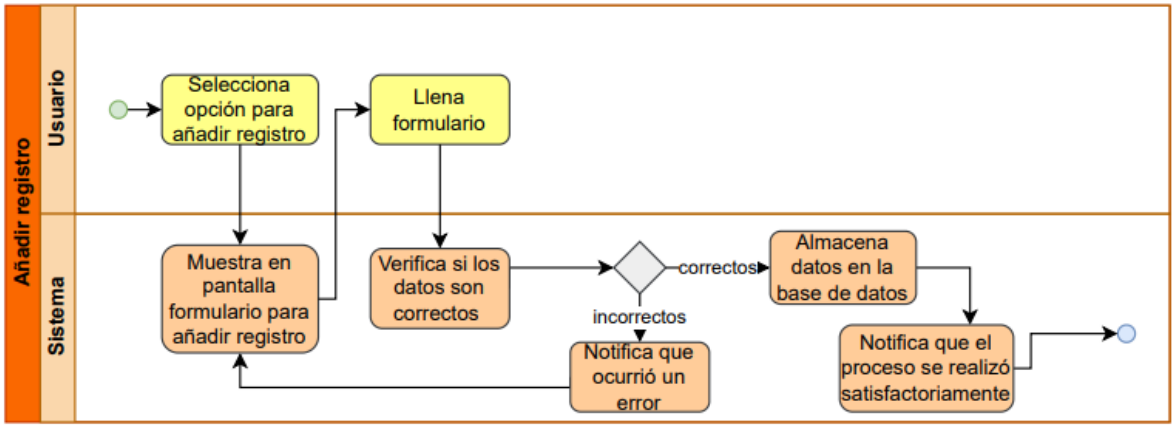

Figura 21. Diagrama de actividades para añadir registro.

Eliminar registro

Para eliminar un registro de la base de datos, primero el sistema muestra el catálogo de registros almacenados. El usuario selecciona el registro que desea eliminar. Como respuesta a ello, el sistema muestra el contenido del registro y también muestra la opción para eliminarlo. Cuando la persona selecciona la opción de eliminar, el sistema le solicita su confirmación; esto con el objetivo de asegurar que la persona desea eliminar el registro y no haya sido seleccionada la opción de manera accidental. En caso de que la persona confirme la eliminación, el sistema elimina el registro de la base de datos y le notifica al usuario cuando el proceso se haya realizado correctamente. En caso de que el usuario cancele la acción, el sistema muestra nuevamente el contenido del registro. Esto se muestra en la Figura 22.

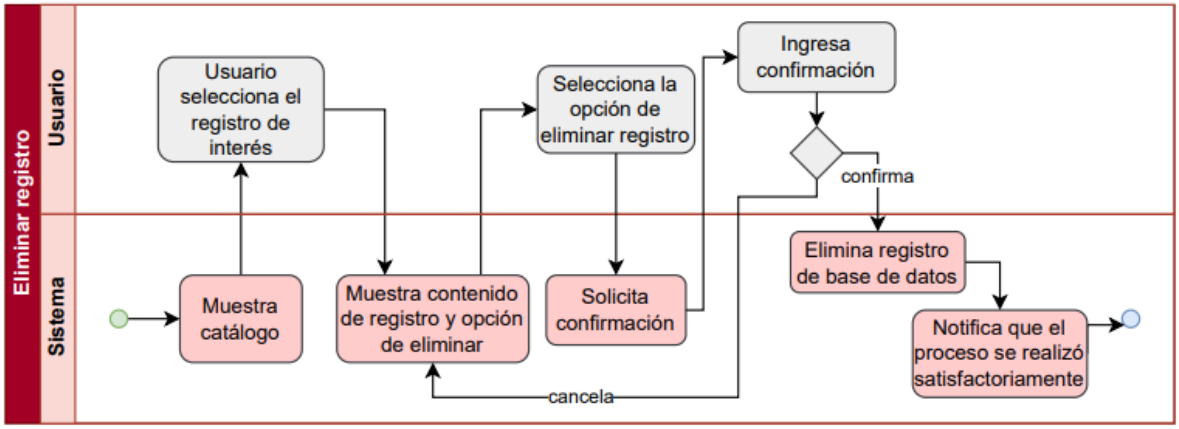

Figura 22. Diagrama de actividades para eliminar registro.

Modificar registro

Para editar un registro, el sistema muestra el catálogo de registros. El usuario selecciona el registro que desea modificar, con la finalidad de que el sistema le muestre su contenido. Posteriormente, el usuario realiza las modificaciones deseadas y selecciona la opción "guardar". El sistema valida los datos ingresados; en caso de que exista un error que impida guardar los cambios, el sistema le notifica al usuario y le muestra nuevamente la pantalla con el contenido del registro. En caso contrario, el sistema realiza las modificaciones en la base de datos y le notifica al usuario que el proceso se realizó correctamente. Esto se puede observar en la Figura 23.

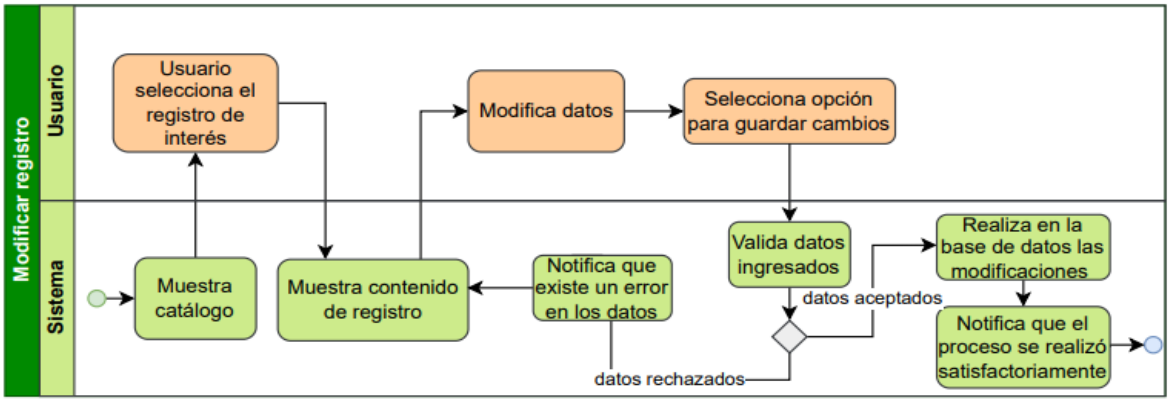

Figura 23. Diagrama de actividades para modificar registro.

Entrenar modelo

Para entrenar el modelo, es necesario obtener audios del usuario pronunciando la palabra a reconocer. Para ello, el sistema le muestra el catálogo de registros al usuario. El usuario selecciona el registro a entrenar y, como respuesta, el sistema muestra el contenido del registro y distintas opciones, en las que se encuentra la opción de entrenar el modelo. El usuario selecciona la opción de entrenamiento y el sistema muestra la pantalla con la opción para iniciar grabación; el usuario selecciona la opción y pronuncia la palabra. Una vez terminada la grabación, el sistema le permite al usuario reproducir el audio y le brinda opciones para almacenar el audio o eliminarlo. En caso de que la persona haya seleccionado la opción de reproducción del audio, el sistema lo reproducirá y, una vez realizado, le mostrará nuevamente las opciones anteriormente mencionadas. En caso de que el usuario seleccione la opción de eliminar audio, el sistema solicitará una confirmación; si el usuario confirma, el sistema regresa a la pantalla con la opción inicial de grabar audio, si el usuario cancela la eliminación, se muestran las opciones de reproducción, almacenamiento o eliminación del audio. Como última opción, si el usuario selecciona la opción de aceptar, el sistema almacena el audio y le notifica a la persona que la actividad fue realizada correctamente. Esto se muestra en la Figura 24.

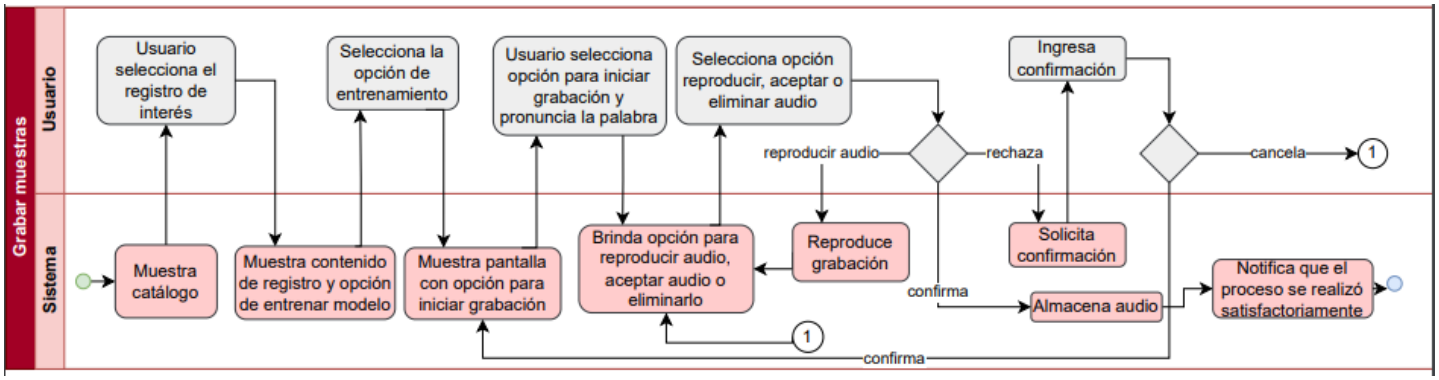

Figura 24. Diagrama de actividades para entrenamiento del modelo.

Emplear modelo para reconocimiento

Con la finalidad de utilizar el modelo para el reconocimiento de palabras, primero el usuario selecciona la opción de reconocimiento. De esta manera, el sistema muestra la pantalla correspondiente, la cual comenzará una grabación. El usuario pronuncia la palabra deseada y, posteriormente, el sistema aplica el modelo de inteligencia artificial. Una vez obtenido el resultado del modelo, se reproduce la salida que corresponde con el resultado. La Figura 25 muestra el diagrama de actividades para el reconocimiento.

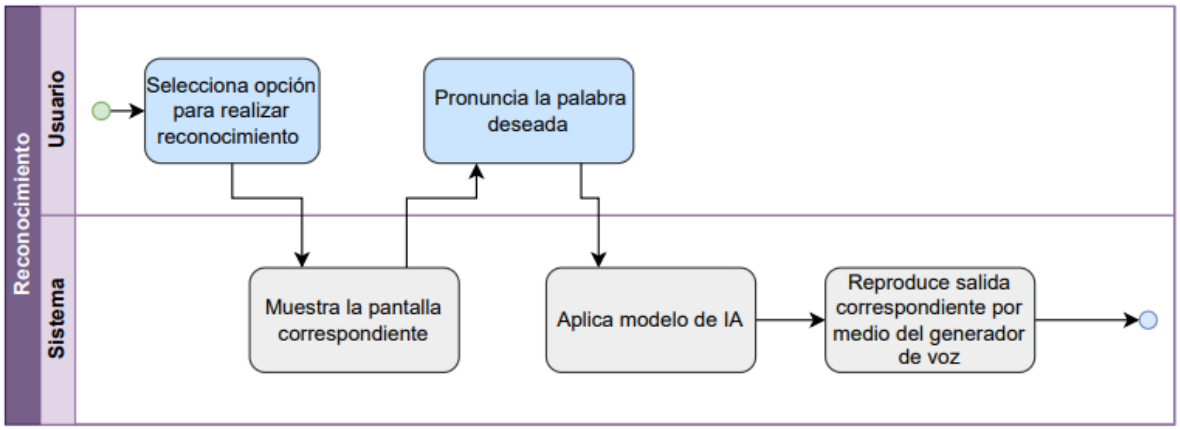

Figura 25. Diagrama de actividades para reconocimiento de voz.

3.5 Modelo de datos

El modelo de datos del presente proyecto se compone de dos entidades: Palabra y Audio. La entidad Palabra tiene como llave primaria id_palabra, y considera la palabra a pronunciar por el usuario (palabra_in) y el resultado a emitir por el dispositivo (palabra_out). También se considera el almacenamiento de los datos que serán utilizados para el entrenamiento del modelo; es por ello que, la entidad Audio considera su almacenamiento en conjunto con su id para su identificación (id_audio). Debido a que una palabra puede estar relacionada con varios audios, es necesario

que Audio tenga como llave foránea la llave primaria de Palabra (id_palabra). El modelo de datos se presenta en la Figura 26.

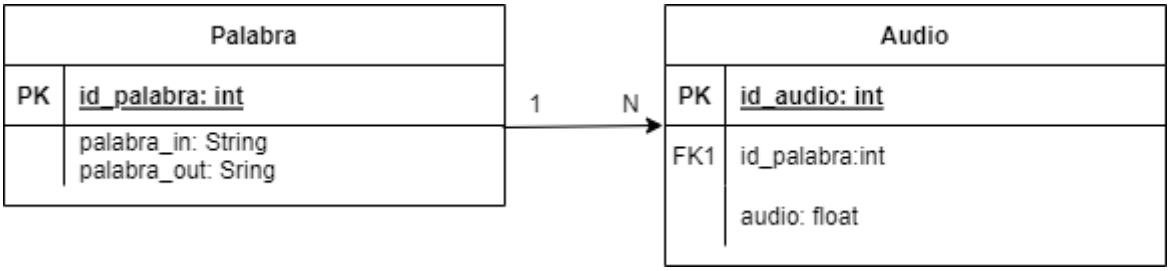

Figura 26. Modelo de datos.

3.6 Arquitectura del sistema

La arquitectura considerada para el sistema se compone de 4 módulos: el módulo de entrada, de entrenamiento, de reconocimiento y de resultados. El módulo de entrada obtiene la señal de audio, la cual contiene la palabra que el usuario desea que se reconozca. Posterior a ello, se encuentra el módulo de entrenamiento. En él se considera el preprocesamiento del audio y el etiquetado de datos. Como parte del preprocesamiento se contempla la configuración del audio según las características deseadas por el modelo a utilizar y la extracción de características del audio. Tanto las características extraídas como el etiquetado de los datos serán almacenados en la base de datos y serán utilizados para el entrenamiento de la red neuronal. En el módulo de reconocimiento también se lleva a cabo la extracción de características como parte del preprocesamiento; además de la aplicación del modelo entrenado. Finalmente, en el módulo de resultados se obtiene la frase o palabra que se encuentra vinculada con la salida de la red neuronal. La frase es conseguida desde la base de datos y es reproducida por medio de un generador de voz. La arquitectura del sistema se puede visualizar en la Figura 27.

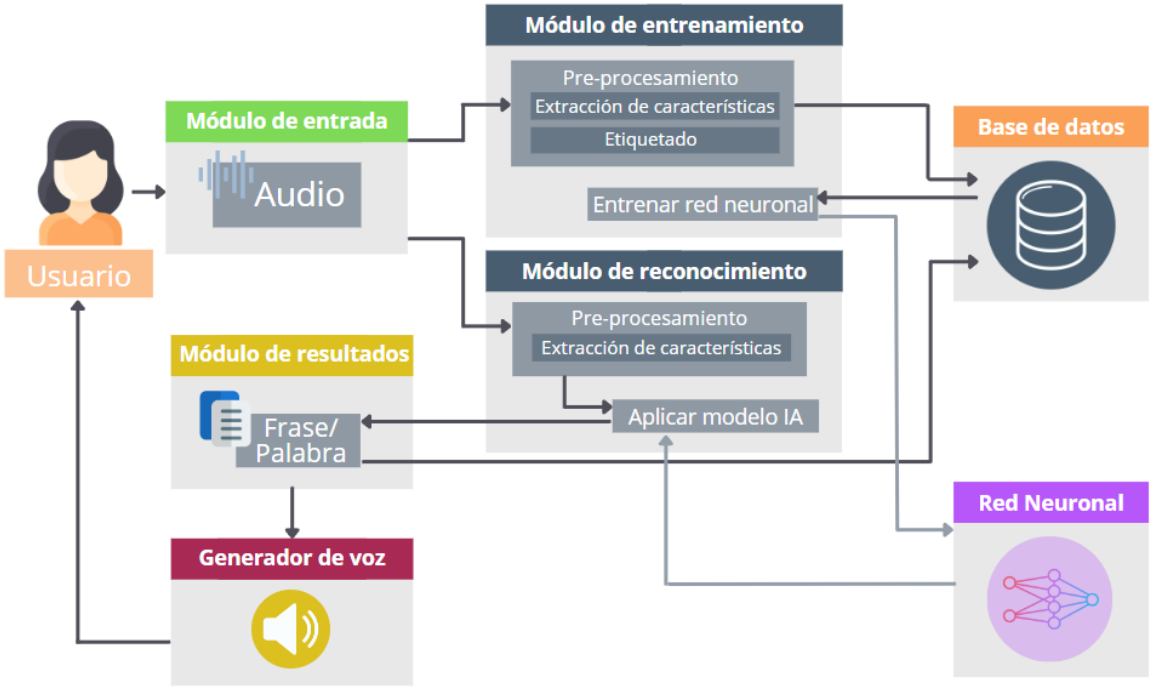

Figura 27. Arquitectura del sistema.

Capítulo IV: Implementación de sistema

4.1 Conjuntos de datos (*datasets*)

4.1.1. Identificación de casos

Para entrenar el modelo propuesto, es necesaria la obtención de un conjunto de datos que sea utilizado para su entrenamiento y validación. En este caso, se consideraron dos métodos para la recolección de datos: un conjunto de datos realizado por terceros y se encuentre disponible para su uso (1) o el desarrollo de un conjunto de datos propio (2). En el primer método se plantea el uso de un *dataset* ofrecido por terceros, en el cual se encuentren audios que contengan palabras pronunciadas por personas con disartria leve a moderada en español. Para el segundo, se considera la generación de un *dataset* partiendo de cero; llevando a cabo la recopilación de audios de usuarios potenciales para el sistema. Sin embargo, hasta el momento, no se ha hallado un conjunto de datos de terceros que cumpla con las características mencionadas y se encuentre disponible para su uso en la presente investigación. Debido a lo anterior, se optó por el segundo método; es decir, se generó un *dataset* con apoyo de usuarios potenciales.

Los audios obtenidos para la investigación se obtuvieron de dos pacientes del Centro de Rehabilitación e Inclusión Infantil Teletón (CRIT) ubicado en Hermosillo, Sonora. Las personas consideradas, quienes son menores de edad, presentan disartria leve y moderada. La selección de casos se realizó con el apoyo del terapista de lenguaje del mismo centro de rehabilitación, Ernesto Ontiveros Medina. Dentro de la investigación, se mencionará a cada uno de los casos por medio de un ID generado con la inicial del nombre de la persona y su tipo de disartria. Para la persona con disartria leve, se tiene el ID I-DL; mientras que, para la persona con disartria moderada, se considera el ID Y-DM. Esto se puede observar en la Tabla 1.

Tabla 1. *Identificación de los casos y su respectiva disartria.*

Identificación del paciente	Tipo de disartria
I-DL	Disartria leve
Y-DM	Disartria moderada

Fuente: elaboración propia.

4.1.2. Características del dispositivo y condiciones del lugar para obtención de audios

Para la recolección de audios se utilizó una tableta Samsung Galaxy Tab A7 Lite. La cual tiene sistema operativo Android 11, una capacidad de almacenamiento de 32 GB y RAM de 3 GB. Los audios fueron grabados en las instalaciones del CRIT, se grabaron en distintas zonas del lugar con la finalidad de obtener audios con ruido de fondo variable. Estos audios fueron recolectados en distintos días en un período de 6 meses.

4.1.3. Generación de *datasets*

Debido a que uno de los propósitos del sistema es que sea dependiente del usuario, se desarrollaron 2 conjuntos de datos; uno para cada caso. Las palabras que conforman cada conjunto de datos fueron seleccionadas con ayuda del terapista del lenguaje y el padre o tutor del paciente. Se consideraron aquellas palabras de uso común para la persona con disartria. El conjunto de datos para el caso *I-DL* se compone de 5 palabras: caricatura, PawPatrol, sándwich, teléfono y dinosaurio. Mientras que, para el caso Y-DM, se consideraron las palabras Kenia, agua y silla.

Al momento de realizar las grabaciones, varias repeticiones de una palabra fueron grabadas en un mismo audio; debido a lo anterior, fue necesario cortarlos para obtener solo una repetición por audio. Para ello, se importaron los audios de la Tablet a una Laptop y se utilizó el software Audacity que permite la manipulación de audios. Con ello, se obtuvieron los conjuntos de datos a utilizar. Para el caso de I-DL, se obtuvieron 60 repeticiones por cada una de las palabras; es decir, se obtuvieron 300 audios en total. Para el caso de Y-DM se obtuvieron 65 repeticiones de las 3 palabras, siendo un total de 195 audios.

Como parte de las experimentaciones, se generaron dos conjuntos de datos para el caso I-DL (observar Tabla 2), con la finalidad de tener un conjunto en donde la duración de cada repetición fuera de 2 segundos y otro con repeticiones de 3 segundos de duración. De esta manera, se busca analizar si la duración de los audios repercute en el reconocimiento del modelo y, si es el caso, optar por aquella que brinda mejores resultados de reconocimiento.

Tabla 2. *Conjuntos de datos para el caso I-DL.*

Nombre del dataset	Duración por palabra (seg)
I-DL-2	2
I-DL-3	3

Fuente: elaboración propia.

4.2. Modelo basado en Transfer Learning

Una de las técnicas para trabajar con pocos datos y obtener buenos resultados, es la denominada *transfer learning.* En nuestro caso, el conjunto de datos es pequeño, por lo que se determinó conveniente aplicar un método de *transfer learning*.

Tomando en consideración la propuesta de Andrej Karphaty, mencionada en el capítulo 2, para la identificación del método a aplicar; se consideró la extracción de características a partir del modelo YAMNet para su empleo en el entrenamiento de una red neuronal superficial.

4.2.1 Modelo YAMNet

YAMNet (*Yet Another Mobile Network*, en inglés) es un clasificador de audios ofrecido por TensorFlow Hub; el cual es un repositorio que provee diversos modelos preentrenados. Para el entrenamiento de YAMNet se empleó AudioSet; un *dataset* conformado por 2.1 millones de audios etiquetados, los cuales fueron obtenidos a partir de videos de la plataforma YouTube. Siendo en total 5 mil 800 horas de audio y 521 clases. El modelo clasifica audios en categorías como: música, habla, vehículos, animales, viento, vehículo de emergencias, risas, helicópteros, entre otros. El modelo YAMNet emplea la arquitectura MobileNetV1, la cual está optimizada para su uso en dispositivos móviles, al requerir menos memoria del dispositivo que otras arquitecturas. MobileNetV1, arquitectura propuesta por Google, reduce su tamaño en comparación de otras arquitecturas al utilizar las convoluciones separables en profundidad (*Depthwise separable convolution* en inglés), en donde la cantidad de parámetros es menor a los de una convolución estándar. En este tipo de convoluciones, se realizan dos tipos de operaciones: convoluciones en profundidad y las puntuales. En la Figura 28, se presenta la arquitectura de MobileNetV1 [56].

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figura 28. Arquitectura MobileNetV1 [56].

Para llevar a cabo la inferencia, YAMNet debe recibir como entrada un tensor unidimensional de tipo flotante, el cual contenga una señal de audio con las siguientes características: señal de un solo canal (mono) de 16kHz y con un rango de -1 a 1 de amplitud. El modelo extrae características de la señal de audio al generar el espectrograma de Mel. La señal es dividida en ventanas de 0.96 segundos de largo con un salto de 0.48 segundos. El arreglo obtenido a partir del espectrograma de Mel es utilizado como entrada al modelo y, como resultado, se obtienen las predicciones. De manera general, este proceso se observa en la Figura 29.

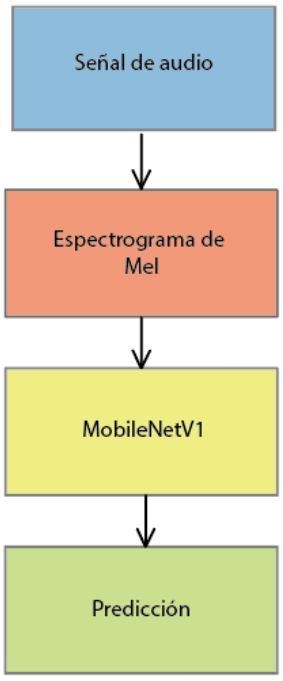

Figura 29. Proceso para la inferencia utilizando el modelo YAMNet [57].

4.2.2 Modelo personalizado

Para la presente investigación, se consideró un modelo personalizado constituido por una capa de entrada, capas de dilución (*Dropout* en inglés), capa *Flatten* y una capa completamente conectada (*Dense layers* en inglés). En total, 535,045 parámetros son entrenados. La composición de la red neuronal se puede observar en la Figura 30.

```
Model: "my_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dropout_42 (Dropout)        (None, 4, 1024)           0

 dense_42 (Dense)            (None, 4, 512)            524800

 dropout_43 (Dropout)        (None, 4, 512)            0

 flatten_23 (Flatten)        (None, 2048)              0

 dense_43 (Dense)            (None, 5)                 10245

=================================================================
Total params: 535,045
Trainable params: 535,045
Non-trainable params: 0
_________________________________________________________________
```

Figura 30. Modelo personalizado.

Las capas del modelo tienen las siguientes características y funciones:

- Tf.keras.layers.Dense: se trata de una capa completamente conectada en la red neuronal.
- Tf.keras.layers.Flatten: tiene como finalidad cambiar las dimensiones de la entrada, de multidimensional a unidimensional. Esta capa es utilizada frecuentemente en la transición de una capa convolucional a una capa completamente conectada.
- Tf.keras.layers.Dropout: se utiliza con el objetivo de reducir el sobreaprendizaje de la red neuronal; es decir, se trata de una técnica de regularización. En este caso, se utilizan *dropouts* de 0.5, es decir del 50%. Este porcentaje nos indica la probabilidad de que una neurona se mantenga activa durante el entrenamiento. La eliminación de neuronas es de manera aleatoria y temporal; ya que, después de actualizar los pesos, se recuperan las neuronas eliminadas y, siguiendo con el entrenamiento, se seleccionan nuevamente, de manera aleatoria, las siguientes neuronas que serán desactivadas.

La capa de entrada del modelo personalizado recibe las características extraídas del modelo YAMNet. El tamaño de este tensor de entrada depende de la duración del

audio recibido por YAMNet. En el caso de los audios con duración de 2 segundos, el tamaño del tensor es de 4 x 1024; mientras que, en los audios de 3 segundos, es de 6 x 1024. Posterior a la capa de entrada, se tiene un *dropout* del 50%. La arquitectura del modelo continúa con una capa completamente conectada que se compone de 512 neuronas, donde se utiliza a ReLU como función de activación; seguido de otro *dropout* de un 50%. Finalmente, se tiene una capa completamente conectada, en donde, el número de neuronas depende de la cantidad de clases que se consideren. Es decir, si el objetivo es clasificar 5 palabras diferentes, se tendrán 5 neuronas en esta capa. El proceso para la inferencia se visualiza en la Figura 31.

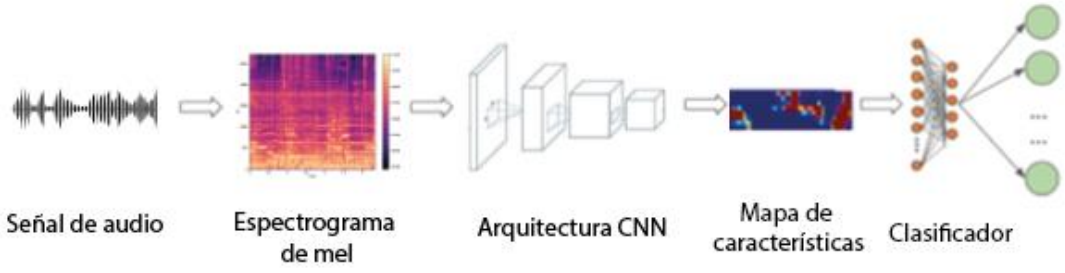

Figura 31. Proceso de inferencia utilizando modelo YAMNet y modelo personalizado [57].

4.2.3. Procesamiento de la señal

Para mejorar el rendimiento de una red neuronal, en ocasiones se utilizan técnicas de normalización de datos. Un ejemplo de ello es el normalizar las entradas sobre el intervalo [0,1] o utilizar z-scores. En este caso, ya que se utilizará un modelo existente como base, es necesario adaptarse a los requerimientos establecidos por dicha red. Para poder utilizar YAMNet, es necesario que los datos estén normalizados en un rango de [-1, 1]. También, es necesario que la frecuencia de muestreo sea de 16,000 Hz y la señal sea de un solo canal (canal mono).

Cuando se trabaja con señales de audio, frecuentemente ésta es convertida a un espectrograma, donde se obtienen las frecuencias en el tiempo; también es común

convertir la señal a un espectrograma de Mel (*Mel spectrograms* en inglés), en donde se representa la señal en una escala logarítmica, similar a como realmente los humanos percibimos el sonido. Los resultados obtenidos de la conversión del audio son utilizados como entrada a la red neuronal. En este caso, YAMNet convierte la señal de audio de entrada en espectrograma de Mel, que posteriormente se utiliza como entrada para el modelo.

4.2.4. Entrenamiento del clasificador

Para el entrenamiento del modelo, se utilizaron los distintos conjuntos de datos generados. Para poder observar el rendimiento del modelo ante distintas situaciones, se varió la cantidad de repeticiones por palabra en el entrenamiento, validación y/o en la evaluación del modelo. De la misma manera, se varió la cantidad de palabras a reconocer, con la finalidad de analizar la respuesta del modelo ante diferentes cantidades de clases (palabras a reconocer).

Durante el entrenamiento, se consideró *Sparse categorical crossentropy* como función de pérdida, mientras que, para optimizar a la red neuronal, se utilizó el optimizador *Adam*. Se consideraron 60 épocas como límite. También, se utilizó el método *EarlyStopping* con paciencia de 3 que monitoreaba el error en el conjunto de entrenamiento. De manera que, si el error durante el entrenamiento no disminuía en el transcurso de 3 épocas, el entrenamiento se detenía en ese momento, antes de las 60 épocas.

Es importante mencionar que los audios provenientes de un mismo audio (lo comentado anteriormente), fueron colocados en una misma división, ya sea en el entrenamiento, validación o evaluación; puesto que, de otra manera, si los audios obtenidos de una misma fuente son distribuidos en distintas divisiones, se considera que los resultados del modelo son poco confiables.

4.3. Aplicación móvil

Para la interacción de la persona con el modelo, se desarrolló una aplicación móvil para su uso en dispositivos con sistema operativo Android. Se utilizó el entorno de desarrollo integrado oficial de Android, Android Studio. La versión del IDE es Arctic Fox 2020.3.1.

La aplicación fue programada con el lenguaje de programación Kotlin y el lenguaje de marcado XML. Fue utilizado el componente de navegación de Android Jetpack. Este último, se trata de un conjunto de librerías que busca facilitar el desarrollo de aplicaciones móviles, al ayudar a reducir el código estándar y el seguir prácticas apropiadas. El componente de navegación facilita la programación de la navegación que ocurre entre las diferentes pantallas de la aplicación. En la Figura 32 se muestra el gráfico de navegación del presente proyecto, en donde se observan los distintos destinos y las posibles rutas.

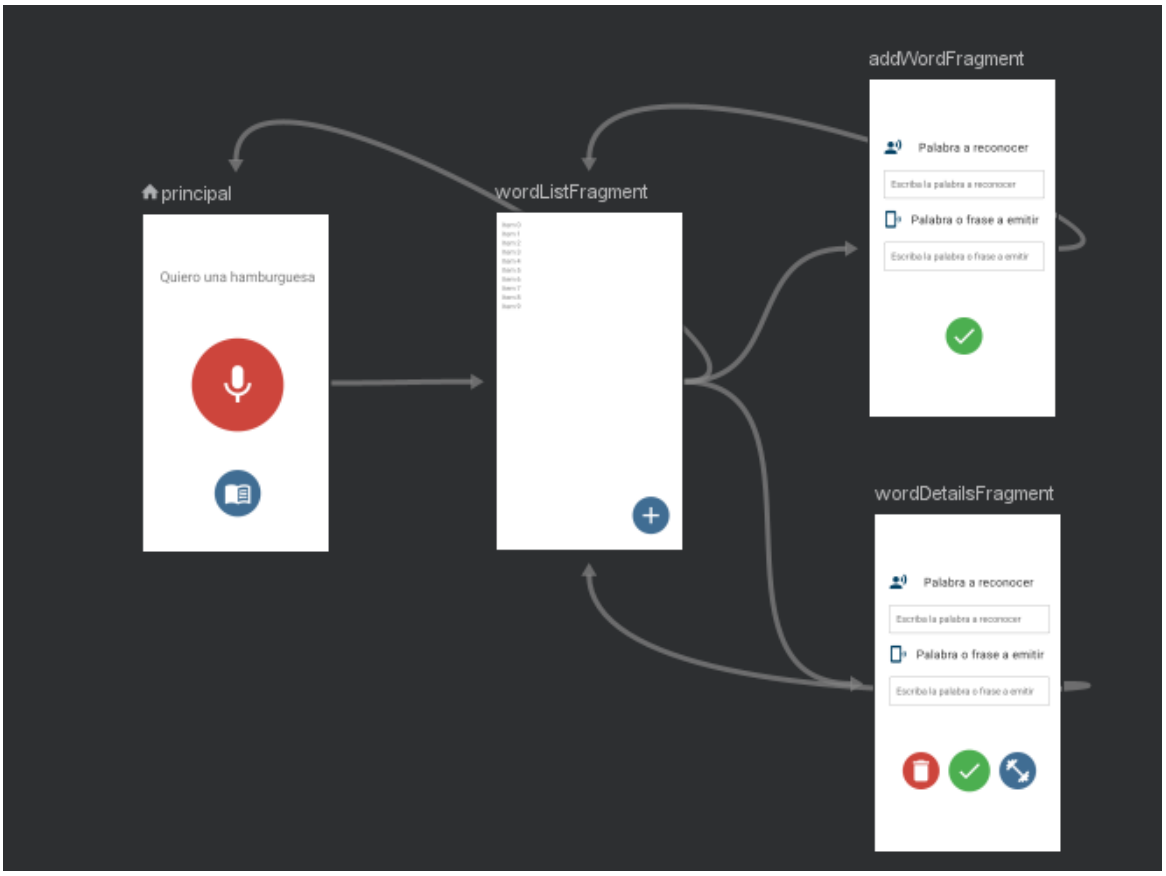

Figura 32. Gráfico de navegación del proyecto.

En la pantalla principal, mostrada en la Figura 33, se muestran dos botones. Al presionar el botón rojo, la aplicación comienza a grabar audio, con la finalidad de realizar un reconocimiento del habla. El botón azul, al dar *click*, lleva a la lista de palabras que el usuario ha agregado para su reconocimiento. Esta pantalla se observa en la Figura 34. Cada uno de los registros de la lista contiene la palabra a reconocer por el sistema (mostrado en la parte superior del cuadro azul) y la palabra o frase a emitir una vez realizado el reconocimiento (ubicada en la parte inferior del cuadro azul). Dentro de la misma pantalla, es posible dar *click* a alguno de los registros para poderlo editar, entrenar o eliminar; también es posible agregar un nuevo registro por medio del botón ubicado en la esquina inferior derecha de la pantalla.

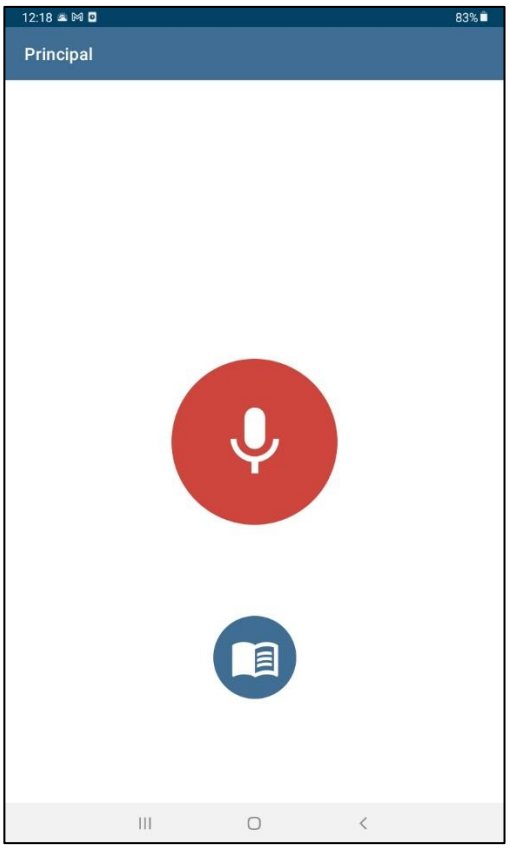

Figura 33. Pantalla principal.

Figura 34. Lista de palabras.

En la Figura 35 se muestra la pantalla en donde es posible agregar un nuevo registro a la lista de palabras. En ella, es necesario indicar la palabra a reconocer y la palabra o frase a emitir. Para el almacenamiento y manipulación de los registros, se empleó la librería Room, ofrecida por Android; la cual, facilita el manejo de base de datos SQLite.

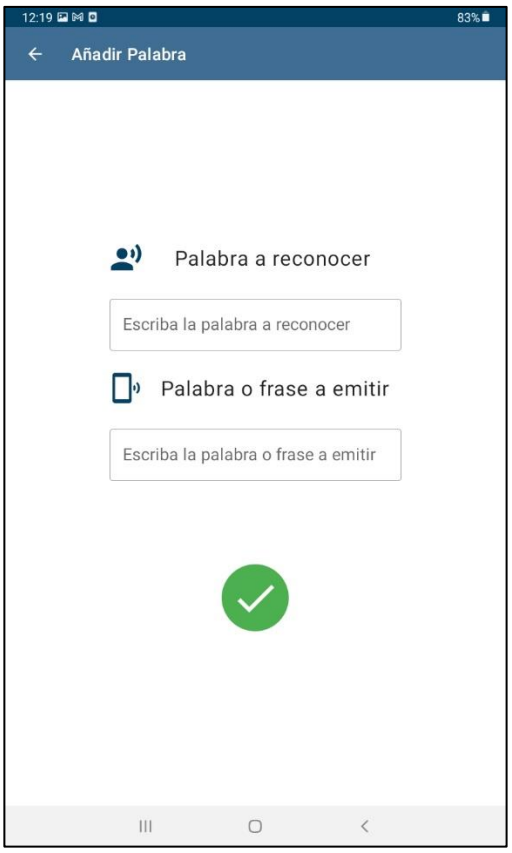

Figura 35. Interfaz para agregar nuevos registros.

Para modificar, entrenar o eliminar registros, se emplea la interfaz mostrada en la Figura 36. En este caso, para modificar la palabra o frase a emitir, es necesario hacer la modificación dentro del TextView correspondiente y, posteriormente, guardar la modificación al presionar el botón verde. Para grabar repeticiones de la palabra a reconocer para su posterior uso en el entrenamiento, es necesario presionar el botón azul ubicado en la parte inferior derecha. En caso de que el usuario deseé eliminar el registro, la acción se realiza al presionar el botón rojo.

Figura 36. Interfaz para la modificación, entrenamiento o eliminación de registro.

Capítulo V: Resultados y discusión

5.1 Clasificación para caso I-DL

Como se mencionó en el capítulo 3, se generaron 2 *datasets* para el caso I-DL.: I-DL-2 e I-DL-3. En el primer conjunto de datos mencionado, las grabaciones tienen una duración de 2 segundos cada una; mientras que, para el dataset I-DL-3, la duración es de 3 segundos. Para cada uno de los *datasets* del caso I-DL, se realizaron experimentaciones considerando 30, 35 o 40 repeticiones por palabra para llevar a cabo el entrenamiento; el resto de los datos fueron utilizados para le evaluación del modelo. En la Tabla 3 es posible observar los resultados de las experimentaciones al utilizar el *dataset* de tres segundos por grabación. Para el entrenamiento, se utilizó un *batch* de 32. El mayor porcentaje de exactitud obtenido en la evaluación es de un 67%, utilizando 30 y 35 repeticiones.

Tabla 3. *Entrenamiento con dataset I-DL-3 (duración de 3 segundos por audio), considerando 30, 35 y 40 repeticiones por palabra y un batch de 32.*

	Entrenamiento		Evaluación	
Cantidad de datos para entrenamiento por palabra	Error	Exactitud	Error	Exactitud
30	0.2379	0.9128	1.24	0.67
35	0.666	0.7803	1.409	0.67
40	0.827	0.7716	2.085	0.61

Fuente: Elaboración propia.

En la Tabla 4, se muestran los resultados obtenidos con el *dataset* I-DL-3 considerando 30, 35 y 40 repeticiones. Sin embargo, a diferencia del anterior, el *batch* es modificado de 32 a 16. Al comparar los resultados de la Tabla 4 y 5, es posible

observar que, en los tres casos (30, 35 y 40) la exactitud obtenida en el entrenamiento considerando un *batch* de 16, aumenta considerablemente con respecto a los resultados obtenidos con un valor de 32. No obstante, la exactitud obtenida en la evaluación es similar presenta solamente un aumento de 9% y 4% para el caso de las 35 y 40 repeticiones, respectivamente.

Tabla 4. *Entrenamiento con dataset I-DL-3, considerando 30, 35 y 40 repeticiones por palabra y un batch de 16.*

	Entrenamiento		Evaluación	
Cantidad de datos para entrenamiento por palabra	Error	Exactitud	Error	Exactitud
30	0.1637	0.9530	1.1613	0.67
35	0.0955	0.9653	0.8485	0.75
40	0.2444	0.9137	0.9633	0.65

Fuente: Elaboración propia.

En la Tabla 5 se visualizan los resultados de las experimentaciones realizadas al utilizar el *dataset* I-DL-2. Es posible observar que el mayor porcentaje de exactitud obtenido es de un 73% con el *dataset* de 35 repeticiones.

Tabla 5. *Entrenamiento con dataset I-DL-2 (duración de 2 segundos por audio), considerando 30, 35 y 40 repeticiones por palabra y un batch de 32.*

	Entrenamiento		Evaluación	
Cantidad de datos para entrenamiento por palabra	Error	Exactitud	Error	Exactitud
30	0.1292	0.9597	4.4536	0.63
35	0.0558	0.9885	2.8314	0.73
40	0.1414	0.9397	3.3789	0.68

Fuente: elaboración propia.

Posteriormente, se continuaron las experimentaciones con el *dataset* I-DL-2, en el cual los audios tienen una duración de 2 segundos. Se realizaron tres experimentaciones, considerando: 30, 35 y 40 repeticiones por palabras. Los resultados obtenidos se encuentran en la Tabla 6.

Tabla 6. *Entrenamiento con dataset I-DL-2, considerando 30, 35 y 40 repeticiones por palabra y un batch de 16.*

	Entrenamiento		Evaluación	
Cantidad de datos para entrenamiento por palabra	Error	Exactitud	Error	Exactitud
30	0.0354	0.9866	5.1268	0.64
35	0.0553	0.9828	4.1892	0.77
40	0.1213	0.9648	3.2883	0.69

Fuente: elaboración propia.

El mayor porcentaje obtenido en las experimentaciones es de un 77%; en la experimentación de 35 repeticiones con duración de 2 segundos y un *batch* de 16.

En la Figura 37 se encuentra su matriz de confusión. De las 20 repeticiones por palabra utilizadas para la evaluación; en el caso de PawPatrol, caricatura y dinosaurio, 15 de ellas fueron clasificadas correctamente. Mientras que en las palabras sándwich y teléfono, 16 fueron clasificadas correctamente y 4 de manera errónea.

	PawPatrol	Caricatura	Sándwich	Teléfono	Dinosaurio
PawPatrol	15	1	0	4	0
Caricatura	0	15	4	0	1
Sándwich	0	2	16	1	1
Teléfono	2	2	0	16	0
Dinosaurio	1	4	0	0	15

Figura 37. Matriz de confusión 2 segundos, 5 palabras y exactitud de 77%.

Con la finalidad de analizar la exactitud del modelo ante diferente número de clases a reconocer, se realizaron pruebas en donde se tienen 2, 3, 4 y 5 clases. Los resultados se observan en la siguiente tabla.

Tabla 7. *Comparación en la clasificación de 2, 3, 4 o 5 palabras, considerando 35 repeticiones y un batch de 16.*

	Entrenamiento		Evaluación	
Número de palabras	Error	Exactitud	Error	Exactitud
2	0.0151	1	0.6529	0.95
3	0.0306	1	0.7973	0.8667
4	0.0353	0.9929	1.1667	0.7626
5	0.0553	0.9828	4.1892	0.77

Fuente: elaboración propia.

La exactitud, considerando dos palabras, es de 95%, este porcentaje disminuye aproximadamente 9% al agregar una clase más, siendo un total de tres clases. Para la clasificación de 4 y 5 palabras, la exactitud se mantiene entre un 76 y 77%.

5.2 Entrenamiento para el caso Y-DM

Para el caso Y-DM, se consideró la clasificación de dos y tres palabras con una duración de dos segundos por audio. A continuación, se muestran los resultados obtenidos.

Clasificación considerando dos palabras

En este caso, se consideró el entrenamiento del modelo para la clasificación de dos palabras: agua y Kenia; utilizando 35, 40 y 45 repeticiones por palabra. La cantidad de datos restantes fue empleada para la evaluación del modelo. Dentro de las pruebas, se realizó una variación al *batch*, considerando un *batch* de 16 y 32, los resultados se observan en la Tabla 8 y 9, respectivamente.

Tabla 8. *Entrenamiento con dataset Y-DM para el reconocimiento de dos palabras, considerando 35, 40 y 45 repeticiones por palabra y un batch de 16.*

	Entrenamiento		Evaluación	
Cantidad de datos para entrenamiento por palabra	Error	Exactitud	Error	Exactitud
35	0.1089	0.9714	0.7534	0.75
40	0.0805	0.975	0.8336	0.66
45	0.1001	0.9667	0.8057	0.7

Fuente: elaboración propia.

Tabla 9. *Entrenamiento con dataset Y-DM para el reconocimiento de dos palabras, considerando 35, 40 y 45 repeticiones por palabra y un batch de 32.*

	Entrenamiento		Evaluación	
Cantidad de datos para entrenamiento por palabra	Error	Exactitud	Error	Exactitud
35	0.1299	1	0.7836	0.625
40	0.0745	0.975	0.9031	0.7
45	0.1151	0.9778	0.7761	0.65

Fuente: elaboración propia.

El porcentaje de exactitud varía dentro de un 62% y un 75%. En este caso, el mayor porcentaje de exactitud fue obtenido utilizando un *batch* de 16 con 35 repeticiones por palabra. En la Figura 38 se muestra la matriz de confusión obtenida. En el caso de la palabra agua, 12 palabras fueron correctamente clasificadas de las 20 repeticiones en total. Mientras que en la palabra Kenia, se clasificaron correctamente 18 de 20.

	Agua	Kenia
Agua	12	8
Kenia	2	18

Figura 38. Matriz de confusión 2 segundos, 2 palabras y exactitud de 75%.

Clasificación considerando tres palabras

Además de dos palabras, también se realizaron experimentaciones para la clasificación de tres palabras: agua, Kenia y silla. Se consideraron 35 y 45 repeticiones por palabra para el entrenamiento. Al igual que en el caso de dos palabras, se realizó una variación en el *batch* de 16 a 32. Los resultados se pueden observar en la Tabla 10 y 11, respectivamente.

Tabla 10. *Entrenamiento con dataset Y-DM para el reconocimiento de tres palabras, considerando 35 y 45 repeticiones por palabra y un batch de 16.*

	Entrenamiento		Evaluación	
Cantidad de datos para entrenamiento por palabra	Error	Exactitud	Error	Exactitud
35	0.1408	0.981	1.0104	0.65
45	0.2305	0.9185	0.9515	0.5167

Fuente: elaboración propia.

Tabla 11. *Entrenamiento con dataset Y-DM para el reconocimiento de tres palabras, considerando 35 y 45 repeticiones por palabra y un batch de 32.*

	Entrenamiento		Evaluación	
Cantidad de datos para entrenamiento por palabra	Error	Exactitud	Error	Exactitud
35	0.2472	0.9429	0.9209	0.5833
45	0.147	0.9778	1.002	0.6

Fuente: elaboración propia.

Los porcentajes de exactitud alcanzados, a comparación de la clasificación de 2 palabras, se redujeron a un rango de 51% - 65%. De la misma manera que en la experimentación con 2 clases, se obtuvieron los mejores resultados al considerar un *batch* de 16 y 35 repeticiones por palabra, teniendo un 65% de exactitud. En la Figura 39 se observa su matriz de confusión. Es posible observar que, la mitad de los datos pertenecientes a la clase "agua", fueron clasificados correctamente. En el caso de la palabra "Kenia", se obtuvieron 17 clasificaciones correctas de un total de 20 datos. En la clase "silla", más de la mitad fueron clasificados correctamente, siendo un total de 12 clasificaciones correctas de un total de 20.

	Agua	Kenia	Silla
Agua	10	8	2
Kenia	2	17	1
Silla	3	5	12

Figura 39. Matriz de confusión 2 segundos, 3 palabras y exactitud de 65%.

5.3 Pruebas de usabilidad

Se realizaron pruebas con apoyo del caso I-DL y su tutor. Las pruebas consistieron en la navegación entre las diferentes pantallas de la aplicación móvil, con la finalidad de conocer qué tan intuitiva es para su uso. Las personas se evaluaron de manera individual. Para realizar las pruebas, se les indicaron a las personas una serie de instrucciones a seguir. Cuando las personas interactuaban con la aplicación para completar la instrucción dada, se evaluó si finalmente pudo realizar la tarea, la seguridad con la que la realizó (en un rango del 0 al 5), la cantidad de errores que tuvo la persona, el tiempo que le tomó y si presentó signos de frustración (en un rango del 0 a 5).

Para la evaluación del tutor, se consideraron las siguientes instrucciones:

1. Dirígete a la lista de registros,
2. Al terminar la instrucción anterior, agrega un nuevo registro.
3. Partiendo de la página de inicio, busca la manera de eliminar un registro o de modificarlo.
4. Después de la instrucción anterior, regresa a la página principal.

El tutor fue capaz de realizar todas las instrucciones correctamente. La persona comentó que los iconos utilizados para ir a la lista de registros (instrucción 1) y para regresar a la página principal (instrucción 4) son útiles y entendibles. Durante la ejecución de la instrucción 2, comentó que sería adecuado modificar el color del botón para añadir registros a un color más contrastante, con la finalidad de que éste resulte más llamativo y sea fácil ubicarlo. Al estar en la interfaz para añadir un nuevo registro (véase la Figura 35), el tutor mostró signos de frustración, ya que no comprendía la instrucción mostrada en la pantalla que solicita escribir la "palabra o frase a emitir". Indicó que una frase como "frase que quieres que se escuche" es más entendible. En la ejecución de la instrucción 3, la persona se mostró dudosa, ya que no comprendía si era necesario presionar o deslizar el registro para eliminar o modificar el registro. Los resultados obtenidos se resumen en la Tabla 12.

Tabla 12. *Resultados obtenidos de pruebas de usabilidad aplicadas a tutor de caso I-DL.*

Instrucción	Ejecución de tarea		Esfuerzo		Satisfacción
	¿Pudo realizar la tarea? (Sí/No)	Seguridad (0-5)	Cantidad de errores	Tiempo total para ejecutar tarea	Frustración al realizar la tarea (0-5)
1.Ve a la lista de registros.	Sí	5	0	3 seg	0
2. Al terminar la instrucción 1, agrega un nuevo registro.	Sí	4	0	1min 22 seg	3
3.Desde la página de inicio, busca la manera de eliminar un registro o modificarlo.	Sí	1	1	1 min 18 seg	4
4.Regresa a la página principal.	Sí	5	0	4 seg	0

Fuente: elaboración propia.

Para el caso I-DL, se consideraron las mismas instrucciones que para el tutor, además de otras dos instrucciones: una para evaluar la capacidad de la persona para ubicar la interfaz de la aplicación en donde posible realizar grabaciones para entrenar una palabra; y otra instrucción para evaluar que tan fácil es para la persona ubicar el botón que tiene como objetivo el iniciar una grabación para hacer el reconocimiento del habla. El orden de las instrucciones y los resultados obtenidos se muestran en la Tabla 13.

Tabla 13. *Resultados obtenidos en las pruebas de usabilidad aplicadas al caso I-DL.*

Instrucción	Ejecución de tarea		Esfuerzo		Satisfacción
	¿Pudo realizar la tarea? (Sí/No)	Seguridad (0-5)	Cantidad de errores	Tiempo total para ejecutar tarea (seg)	Frustración al realizar la tarea (0-5)
1. Utiliza la aplicación para empezar una grabación y así, la aplicación reconozca lo que quieres decir.	Sí	5	0	1	0
2.Ve a la lista de registros	Sí	5	0	5	0
3. Después, agrega un nuevo registro.	No	0	0	3	5
4.Desde la página de inicio, busca la manera de eliminar un registro o modificarlo.	Sí	1	1	40	4
5.Desde la página de inicio, busca la manera de entrenar un registro.	Sí	1	1	40	4
6. Regresa al inicio	Sí	5	0	2	0

Fuente: elaboración propia.

El caso I-DL fue capaz de realizar 5 de un total de 6 instrucciones. Mostrando facilidad para iniciar una grabación (instrucción 1), identificar la lista de registros (instrucción 2) y para regresar al inicio (instrucción 6). La persona fue incapaz de realizar la instrucción 3, debido a que aún no sabe leer y escribir. Se identificaron signos de frustración y falta de seguridad para encontrar la manera de eliminar un registro (instrucción 4) y para identificar la manera de entrenar una palabra (instrucción 5).

5.4 Discusión

Para el desarrollo de la investigación, fue necesario generar un conjunto de datos para cada uno de los casos de estudio y, para ello, se tomaron, durante meses, diversas grabaciones de los casos de estudio pronunciando las distintas palabras a reconocer. Es importante mencionar que el obtener los conjuntos de datos fue una tarea difícil; ya que dependía de los ánimos de los participantes para colaborar en su desarrollo. Las grabaciones se comenzaban una vez los participantes finalizaban sus terapias en el centro de rehabilitación. Es por ello que, en ocasiones, las personas se sentían cansadas para grabar. En otras ocasiones, incluso en las terapias mostraban poca participación y, para las grabaciones, se mostraban con el mismo carácter. Con el transcurso del tiempo, se determinó que el incentivar a los participantes por medio de recompensas fue un método exitoso para trabajar con ellos y, así, obtener grabaciones para el conjunto de datos. También es importante mencionar que, para el caso de disartria leve, era posible obtener varias repeticiones en un solo día de grabación, puesto que la persona mostraba capacidad suficiente para realizar la actividad repetidas veces. Sin embargo, no fue lo mismo para el caso con disartria moderada; ya que resultaba evidente el esfuerzo que debía realizar la persona para cada una de las repeticiones; es por ello que, por sesión, la cantidad de repeticiones obtenidas fue menor a las del caso de disartria leve.

Al realizar las experimentaciones, se obtuvieron distintos porcentajes de exactitud, dependiendo del número de palabras a reconocer y el tipo de disartria. En el caso de 5 palabras, siendo el mayor número de clases con el que se realizaron

experimentaciones para el caso de disartria leve, se obtuvo como resultado un 77% de exactitud, considerando 35 repeticiones por palabra con una duración de 2 segundos. Para el caso de la disartria moderada, la mayor cantidad de clases con las que se experimentó fue de 2, obteniendo un 75% como el porcentaje de exactitud más alto. Esto fue obtenido con 35 repeticiones por palabra.

En el caso de las experimentaciones de disartria leve, se observa que, al aumentar la cantidad de clases, se reduce el porcentaje de reconocimiento por parte del modelo. Sin embargo, para la clasificación en 4 y 5 clases, los porcentajes se asemejan considerablemente a comparación de los anteriores. Por lo que, se deduce que después de 4 clases, el porcentaje de reconocimiento varía ligeramente cuando la cantidad de clases aumenta. Sin embargo, para su comprobación, es necesario realizar experimentaciones con una mayor cantidad de palabras a reconocer.

Trabajos similares fueron desarrollados para el idioma italiano [11] e inglés[7]; para los modelos se utilizó una arquitectura CNN y fueron desarrollados el reconocimiento de 12 a 16 palabras. La cantidad de datos utilizados para el idioma italiano fue de 1,000 contribuciones de 3 personas; en el caso del idioma inglés, se utilizaron de 30 a 59 repeticiones por cada palabra. Teniendo como resultado para el idioma italiano, un 57.5% de reconocimiento; mientras que, para el idioma inglés, se obtuvo un 31.4% de exactitud.

Los resultados obtenidos en esta investigación demuestran que, al tener un conjunto de datos reducidos, el utilizar técnicas de aprendizaje por transferencia es una opción prometedora para obtener un mayor porcentaje de reconocimiento. Sin embargo, es necesario el realizar experimentaciones con una mayor cantidad de palabras para su verificación.

Capítulo VI: Conclusiones

6.1 Conclusiones

Durante el presente trabajo de investigación, se desarrollaron distintos conjuntos de datos, con el apoyo de dos personas con disartria y distintos niveles de inteligibilidad: leve y moderada. Estos conjuntos de datos fueron utilizados para el entrenamiento del modelo, con el objetivo de obtener un modelo dependiente del usuario. Debido a la escasez de datos, se aplicó la extracción de características a un modelo preentrenado, YAMNet, con la finalidad de utilizar los datos obtenidos como entrada a un modelo multicapa personalizado.

De las experimentaciones realizadas, se observó que existe una tendencia a disminuir el porcentaje de reconocimiento al aumentar la cantidad de palabras a reconocer. Para el caso de disartria leve, el porcentaje de exactitud disminuyó un 18%, considerando los resultados obtenidos con 2 y 5 palabras a clasificar. Mientras que, en el caso de disartria moderada, el porcentaje disminuyó de 75 a 62% al agregar una clase.

Los mejores resultados obtenidos para la clasificación de 5 palabras en el caso de disartria leve, fue al emplear 35 repeticiones por palabra para el entrenamiento, con una duración de dos segundos por audio y empleando un *batch* de 16; teniendo como resultado un 77% de exactitud. Para la clasificación de dos y tres palabras en el caso de disartria moderada, los mejores resultados fueron obtenidos con los mismos parámetros que el caso anterior, siendo de 75 y 65%, respectivamente.

6.2 Trabajo a futuro

Para poder conocer el comportamiento del modelo a mayor profundidad, es necesario realizar más pruebas con diversas personas con disartria leve y moderada. Y que, para cada caso, se aumente la cantidad de palabras a clasificar. De igual manera, se propone continuar con la generación de un *dataset* de disartria en idioma español. Tomando como punto de partida lo obtenido durante la presente investigación. Al continuar con la creación del conjunto de datos, será posible emplearlo para continuar con más experimentaciones y, así, analizar y determinar las mejores características para el funcionamiento del modelo.

Por otro lado, es necesario realizar modificaciones a la aplicación móvil considerando los resultados obtenidos en las pruebas de usabilidad y la retroalimentación obtenida por parte de los participantes de las pruebas. También se considera importante el realizar pruebas de usabilidad considerando a personas con cierta restricción motora. Con la finalidad de realizar modificaciones a la aplicación en caso de que presenten dificultad para manipular la aplicación.

Bibliografía

[1] Instituto Nacional de Estadística y Geografía (INEGI), "Estadísticas a propósito del día internacional de las personas con discapacidad," México, 2019. [Online]. Available: https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2019/Discapacidad2019_Nal.pdf.

[2] Instituto Nacional de Estadística y Geografía (INEGI), "La discapacidad en México, datos al 2014," México, 2016. [Online]. Available: http://coespo.qroo.gob.mx/Descargas/doc/DISCAPACITADOS/ENADID 2014.pdf.

[3] R. A. González and J. A. Bevilacqua, "Las disartrias," *Rev. Hosp. Clínico Univ. Chile*, vol. 23, pp. 299–309, 2012.

[4] Mayo Clinic, "Síntomas y causas," *Disartria*, 2019. https://www.mayoclinic.org/es-es/diseasesconditions/dysarthria/symptoms-causes/syc-20371994 (accessed Sep. 27, 2020).

[5] S. R. Shahamiri and S. S. B. Salim, "A multi-views multi-learners approach towards dysarthric speech recognition using multi-nets artificial neural networks," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 5, pp. 1053–1063, 2014, doi: 10.1109/TNSRE.2014.2309336.

[6] INECO, "Disartria," 2018. https://www.ineco.org.ar/patologias/disartria/#:~:text=En los pacientes con disartria,en palabras de mayor longitud. (accessed Dec. 10, 2020).

[7] H. Albaqshi and A. Sagheer, "Dysarthric Speech Recognition using Convolutional Recurrent Neural Networks," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 6, pp. 384–392, 2020, doi: 10.22266/ijies2020.1231.34.

[8] G. Jayaram and K. Abdelhamied, "Experiments in dysarthric speech recognition using artificial neural networks," *J. Rehabil. Res. Dev.*, vol. 32, no. 2, pp. 162–169, 1995.

[9] MathWorks, "Redes neuronales," 2015. https://la.mathworks.com/discovery/neural-network.html (accessed Dec. 18, 2020).

[10] X. Basogain, "Redes neuronales artificiales y sus aplicaciones," *Publicaciones la Esc. Ing.*, 1998.

[11] D. Mulfari, G. Meoni, and L. Fanucci, "Machine learning in assistive technology: A solution for people with dysarthria," *ACM Int. Conf. Proceeding Ser.*, pp. 308–309, 2018, doi: 10.1145/3284869.3284928.

[12] S. Dickson, R. S. Barbour, M. Brady, A. M. Clark, and G. Paton, "Patients' experiences of disruptions associated with post-stroke dysarthria," *Int. J. Lang. Commun. Disord.*, vol. 43, no. 2, pp. 135–153, 2008, doi: 10.1080/13682820701862228.

[13] J. A. Fernández-López, M. Fernández-Fidalgo, R. Geoffrey, G. Stucki, and A. Cieza, "Funcionamiento y discapacidad: la clasificación internacional del funcionamiento (CIF)," *Rev. Esp. Salud Pública*, vol. 83, no. 6, pp. 775–783, 2009, doi: 10.1590/s1135-57272009000600002.

[14] OMS, *Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud (CIF-10)*, vol. 8, no. 1. 1999.

[15] Instituto Nacional de Estadística y Geografía (INEGI), "Clasificación de Tipo de Discapacidad - Histórica," *Inegi*, pp. 1–55, [Online]. Available: http://www.inegi.org.mx/est/contenidos/proyectos/aspectosmetodologicos/clasificadoresycatalogos/doc/clasificacion_de_tipo_de_discapacidad.pdf.

[16] F. Maritza, "Desarrollo de un sistema de información con soporte inteligente para brindar apoyo en el estudio de casos clínicos para estudiantes de fonoaudiología," Universidad de Cuenca, 2017.

[17] G. Bonilla, "Interfaz de voz para personas con disartria," Universidad Tecnológica de la Mixteca, 2012.

[18] C. Clares and F. Zamorano, "Trastornos de la comunicación y el lenguaje." Murcia, 2006.

[19] G. Bonilla and S. Caballero, "Communication Interface for Mexican Spanish Dysarthtic Speakers," *Acta Universitaria*, vol. 22, Guanajuato, México, pp. 98–105, Mar. 2012.

[20] S. Russell and P. Norvig, *Artificial Intelligence. A modern Approach*, Third Edit. Prentice Hall, 2010.

[21] F. Berzal, *Redes Neuronales & Deep Learning*. Independently published, 2019.

[22] IBM, "Artificial Intelligence," 2020. https://www.ibm.com/cloud/learn/what-is-artificial-intelligence (accessed Feb. 18, 2021).

[23] IBM, "Machine Learning." https://www.ibm.com/analytics/machine-learning (accessed Feb. 19, 2021).

[24] L. Rouhiainen, *Inteligencia artificial*. Barcelona: Planeta, 2018.

[25] F. Chollet, *Deep Learning with Python*. New York: Manning, 2018.

[26] A. Innovation, "Qué son las redes neuronales y sus funciones," 2019. https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/ (accessed Feb. 22, 2021).

[27] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," *MIT Press*, 2016. http://www.deeplearningbook.org/.

[28] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *Int. Conf. Eng. Technol. ICET 2017*, pp. 1–6, 2017, doi: 10.1109/ICEngTechnol.2017.8308186.

[29] A. V Joshi, *Machine Learning and Artificial Intelligence*. Springer, 2020.

[30] MathWorks, "Redes neuronales convolucionales." https://la.mathworks.com/discovery/convolutional-neural-network-matlab.html (accessed Feb. 25, 2021).

[31] IBM, "Recurrent Neural Networks," 2020. https://www.ibm.com/cloud/learn/recurrent-neural-networks (accessed Feb. 27, 2021).

[32] G. Zaccone, *Getting started with TensorFlow 2.0*. Bimingham: Packt publishing, 2016.

[33] A. Gulli and S. Pal, *Deep Learning with Keras*. Birmingham: Packt publishing, 2017.

[34] C. Aggarwal, *Neural Networks and Deep Learning*. Cham: Springer, 2018.

[35] K. Huang, A. Hussain, Q.-F. Wang, and R. Zhang, *Deep Learning: Fundamentals, Theory and Applications*. Cham: Springer, 2019.

[36] MathWorks, "Transfer Learing for Training Deep Learning Models." https://la.mathworks.com/discovery/transfer-learning.html (accessed Apr. 05, 2021).

[37] M. Elgendy, *Deep Learning for Vision Systems*. Shelter Island: Manning Publications, 2020.

[38] F. Chollet, "Transfer learning and fine-tuning," 2020. https://keras.io/guides/transfer_learning/ (accessed Apr. 03, 2021).

[39] J. Levis and R. Suvorov, "Automatic speech recognition," *The Encyclopedia of Applied Linguistics*. Blackwell Publishing, pp. 1–8, 2013.

[40] H. D. Barrobés and M. R. Costa-jussa, *Reconocimiento automático del habla*. Universitat Oberta de Catalunya, 2020.

[41] K. Rosen and S. Yampolsky, "Automatic speech recognition and a review of its functioning with dysarthric speech," *Augment. Altern. Commun.*, vol. 16, pp. 48–60, 2000, doi: 10.1080/07434610012331278904.

[42] K. Doshi, "Audio Deep Learning Made Simple: State-of-the-Art Techniques," *Towards Data Science*, 2021.

[43] R. Leland, "Understanding Mel Spectrogram," *Analytics Vidhya*, 2020.

[44] T. Holton, *Digital Signal Processing: Principles and Applications*. Cambridge University Press, 2021.

[45] P. N. S. Network, "What is a Spectrogram?" https://pnsn.org/spectrograms/what-is-a-spectrogram (accessed Jun. 20, 2022).

[46] M. Jeramy and B. Smus, "Chrome Music Lab: Spectrogram." https://musiclab.chromeexperiments.com/spectrogram/ (accessed Jun. 15, 2022).

[47] K. Doshi, "Audio Deep Learning Made Simple: Why Mel Spectrograms perform better," *Towars Data Science*, 2021.

[48] H. Victor, "Emociones en Señales de Voz: Reconocimiento con Redes Neuronales Profundas," Universidad Politécnica de Cataluña, 2021.

[49] T. Holdroyd, *TensorFlow 2.0 Quick Start Guide*. Bimingham: Packt Publishing, 2019.

[50] N. Shukla and K. Fricklas, *Machine Learning With TensorFlow*. Shelter Island: Manning Publications, 2018.

[51] TensorFlow, "API Documentation," 2021. https://www.tensorflow.org/api_docs (accessed Apr. 15, 2021).

[52] B. Pang, E. Nijkamp, and Y. N. Wu, "Deep Learning With TensorFlow: A Review," *J. Educ. Behav. Stat.*, vol. 20, no. 10, pp. 1–22, 2019, doi: 10.3102/1076998619872761.

[53] "Voiceitt." https://www.voiceitt.com/why-voiceitt.html (accessed Sep. 24, 2020).

[54] Therapy Box, "VocaTempo," 2020. https://therapy-box.co.uk/vocatempo (accessed Sep. 22, 2020).

[55] Lucidchart, "Tutorial de diagrama de clases UML," 2021. https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml/#section_0 (accessed Jul. 12, 2021).

[56] A. G. Howard and W. Wang, "MobileNets: Efficient Convolutional Neural

Networks for Mobile Visio Applications," 2017.

[57] C. Malmberg, "Real-time Audio Classification on an Edge Device - Using YAMNet and TensorFlow Lite," 2021.