

Área de Conocimiento de Tecnología de la
Información y Comunicación

Aplicación móvil para la enseñanza - aprendizaje de habilidades de expresión oral a niños con discapacidad auditiva, mediante técnicas de educación especial

**Trabajo Monográfico para optar al título de Ingeniero
en Computación**

Elaborado por:

Br. Héctor Alberto
Avilés Méndez
Carnet: 2016-0137U

Br. Jairo José
García Salaverry
Carnet: 2016-1208U

Tutor:

M.Sc. Narciso Javier
Aguilera Centeno

Managua, 12 de septiembre de 2024

MSc. Claudia Benavidez Rugama
Directora
Área de Conocimiento de Tecnología de la Información y la Comunicación
UNI
Su Despacho.

Estimada Máster Benavidez:

Reciba cordiales saludos de mi parte, deseándole mucho éxito en sus funciones diarias.

El motivo de la presente es para comunicarle que, en mi calidad de Tutor de monografía presentada por: **Br. Jairo José García Salaverry (Carnet # 2016-1208U)** y **Br. Héctor Alberto Avilés Méndez (Carnet # 2016-0137U)**, para optar al grado de Ingeniero en Computación, cuyo tema es: **“Aplicación móvil para la enseñanza - aprendizaje de habilidades de expresión oral a niños con discapacidad auditiva, mediante técnicas de educación especial”**; considero que dicho trabajo reúne los requisitos según la normativa de culminación de estudios de la UNI, para ser sometido a presentación, defensa y evaluación ante el tribunal examinador que designe.

Sin nada más que agregar, le saludo.

Atentamente,

Narciso Javier Aguilera Centeno
Docente Tutor

Agradecimientos

Queremos expresar nuestro profundo agradecimiento al Lic. Jimmy Francisco Baltodano y su intérprete, Lic. Larry Acuña, por su invaluable apoyo y colaboración durante el desarrollo de este trabajo monográfico. Brindaron su experiencia y conocimientos sin esperar nada a cambio.

Agradecemos sinceramente al colegio Melania Lacayo por su continuo respaldo al brindarnos la información necesaria y el entorno adecuado para realizar nuestras investigaciones y pruebas. Su compromiso con la educación especial ha sido una fuente constante de inspiración.

Queremos expresar nuestro más sincero agradecimiento a la Universidad Nacional de Ingeniería por brindarnos la oportunidad de formarnos académica y profesionalmente. Agradecemos a todos los profesores y personal administrativo por su dedicación y compromiso, que han hecho posible nuestro desarrollo integral durante estos años de estudio.

Dedicatoria

Este trabajo monográfico está dedicado a mi madre, cuyo amor incondicional y sacrificios constantes han sido la base de mi educación. Gracias por enseñarme la importancia de la perseverancia y la dedicación. Este logro es tanto de mi madre como mío, sin su apoyo nunca habría alcanzado este importante hito en mi vida.

A mi abuelo, quien con su sabiduría y ejemplo de vida me ha inspirado a ser una persona de principios y valores. La plena confianza que has depositado en mí me ha dado la fortaleza para enfrentar cada desafío. Este título universitario es un reflejo de tu legado y una promesa de seguir adelante con la misma determinación que tú me has inculcado.

Por último, a mi abuela, que en paz descansa. Fue una persona con un gran corazón. ella siempre espero verme graduarme, siendo uno de sus más grandes deseos. Aunque no estes físicamente presente, siempre estarás en mi corazón.

Héctor Alberto Avilés Méndez

Dedicatoria

A mi querida madre, quiero agradecerte profundamente por tu apoyo constante hacia mí. Tu amor incondicional, tus palabras de aliento en los momentos más difíciles me han ayudado a seguir adelante. También tu comprensión en este largo y complicado camino.

A mi padre, ejemplo de firmeza y dedicación. Gracias por enseñarme con tu exigencia el valor del esfuerzo y la disciplina. Este trabajo es un tributo a tu constante guía y amor incondicional. Al igual que mi madre siempre has estado ahí para mí.

Jairo José García Salaverry

RESUMEN

En esta monografía se describe el desarrollo de una aplicación móvil que sirve como una herramienta de apoyo al trabajo de docentes de educación especial en el proceso de enseñanza - aprendizaje de habilidades de expresión oral a niños con discapacidad auditiva, en los niveles elemental y básico de un centro de educación especial. Para su implementación, se aplicó la metodología ágil y de desarrollo para dispositivos móviles Mobile-D, por considerar que se adecuaba muy convenientemente a las necesidades de este trabajo.

La aplicación fue creada utilizando Android Studio, PHP y MySQL, cumpliendo con los requerimientos técnicos y funcionales establecidos. En general, se considera que la misma alcanza las metas planteadas en este trabajo monográfico y que es una herramienta de utilidad práctica para los docentes de educación especial, ya que facilita el proceso de enseñanza y mejora la calidad de la educación para estudiantes con discapacidad auditiva.

Contenido

1. INTRODUCCIÓN	1
2. ANTECEDENTES	3
3. JUSTIFICACIÓN	5
4. OBJETIVOS	7
5. MARCO TEÓRICO	8
5.1. Investigación enfocada en sistemas	8
5.2. Discapacidad auditiva	9
5.3. El sonido y sus características.....	9
5.4. Espectro gráfico del sonido.....	10
5.5. Discapacidad auditiva según grado de pérdida auditiva	11
5.6. Fonética en discapacidad auditiva	12
5.6.1. Fonemas.....	12
5.6.2. Método de enseñanza de comunicación total de Benson Schaeffer ..	13
5.7. Reconocimiento de voz.....	14
5.7.1. Reconocimiento automático del habla (ASR)	14
5.8. Sistema informático	16
5.8.1. Aplicación móvil.....	16
5.8.2. Base de datos de modelo relacional.....	16
5.8.3. Modelo cliente servidor.....	16
5.9. Herramientas de desarrollo.....	17
5.9.1. Plataforma de desarrollo Java	17
5.9.2. El lenguaje PHP	17
5.9.3. Suite de herramientas XAMPP	18
5.9.4. Herramienta de gestión phpMyAdmin y Base de datos MySQL	18
5.9.6. Entornos de desarrollo integrados (IDE)	18
5.9.7. Github.....	19
5.9.8. API de reconocimiento de voz Vosk	19
5.9.9. AmplitudeGraphView	20
5.9.10. Cliente HTTP Retrofit	20
5.9.11. Cliente HTTP Volley	20

5.9.12. Editor de video DaVinci Resolve	21
6. METODOLOGÍA Y DESARROLLO.....	22
6.1. Investigación tecnológica aplicada	22
6.2. Mobile-D	23
6.3. Ciclo de vida	23
6.3.1. Exploración.....	24
6.3.1.1. Perfiles del público de interés.....	26
6.3.1.2. Definición de alcance	26
6.3.1.3. Requerimientos funcionales	26
6.3.1.4. Requerimientos no funcionales	28
6.3.1.5. Limitaciones	29
6.3.2. Iniciación.....	29
6.3.2.1. Configuración de proyecto.....	29
6.3.2.2. Preparación del ambiente.....	30
6.3.2.3. Planificación de fases por iteraciones	30
6.3.2.4. Patrón de diseño MVC (Modelo-Vista-Controlador)	34
6.3.2.5. Arquitectura modelo CS (Cliente-Servidor)	35
6.3.2.6. Diseño	35
6.3.2.7. Material Design	36
6.3.3. Producción.....	37
6.3.3.1. Historias de usuario y StoryCard.....	37
6.3.3.2. Inicio de sesión.....	40
6.3.3.2.1. Implementaciones en Inicio de sesión	41
Autenticación de usuario mediante Retrofit.....	41
Inició sesión como invitado.	41
6.3.3.3. Registro de usuario	42
6.3.3.3.1. Implementaciones del registro de usuario	43
Encriptación de contraseña.....	43
6.3.3.4. Lista de palabras	43
6.3.3.4.1. Implementaciones en lista de palabras.....	45
Listado de contenido por medio de RecyclerView.....	45

Navegación entre fragments	45
6.3.3.5. Reconocimiento de voz	46
6.3.3.6. Reproductor de media	47
6.3.3.6.1. Implementaciones del reproductor de media	49
Reproductor de video	49
Grabadora y reproductor de audio	50
Barra indicadora de porcentaje de acierto	50
6.3.3.7. Cálculo de resultados y códigos de colores	50
6.3.3.8. Evaluación de la pronunciación	52
6.3.3.9. Guardado de progreso	53
6.3.3.10. Diccionario.....	54
6.3.3.11. Estadísticas	56
6.3.3.11.1. Implementaciones de estadísticas	57
Gráfico de barras.	57
Tabla de puntajes.....	58
6.3.3.12. Graficación del sonido	58
6.3.3.12.1. Implementaciones de graficación del sonido	59
Grafica de intensidad con respecto al tiempo	59
6.3.3.13. Base de datos.	60
6.3.3.14. Adición y tratamiento de contenido	60
6.3.4. Estabilización.....	63
6.3.4.1. Refactorización.....	63
6.3.5. Pruebas del sistema	64
6.3.5.1. Esquema de navegación	64
6.3.5.2. Verificación de vistas.....	66
6.3.5.3. Pruebas funcionales.....	67
6.3.5.3.1. Casos de pruebas.....	67
6.4. Estudio de factibilidad.....	71
6.4.1. Factibilidad técnica	71
6.4.2. Factibilidad operativa.....	73
6.4.3. Factibilidad económica	75

6.4.4. Cálculo del esfuerzo mediante Planning Poker .	77
6.4.4.1. Procedimiento	77
7. CONCLUSIONES	81
7.1. Recomendaciones	82
8. BIBLIOGRAFÍA	83
9. ANEXOS	i
A. Diagrama de clases	i
B. Diagramas de actividad	ii
C. Diagrama de base de datos	iv
D. Diagrama de componentes	iv

Índice de Gráficos

Fig. 1. Espectro gráfico del sonido. [Autoría propia]	10
Fig. 2. Ejemplo de descomposición de fonemas. [20].....	13
Fig. 3. Esquema del reconocimiento automático del habla. [25].....	14
Fig. 4. Esquema de reconocimiento de palabras aisladas. [25].....	15
Fig. 5. Esquema de modelo de habla continua. [25].....	15
Fig. 6. Modelo cliente/servidor. [30]	17
Fig. 7. Ciclo de vida de Mobile-D. [27]	24
Fig. 8. Modelo Vista Controlador. [48].....	34
Fig. 9. Modelo cliente servidor. [30]	35
Fig. 10. Inicio de sesión. [autoría propia]	40
Fig. 11. Formulario de ingreso.	42
Fig. 12. Lista de palabras. [autoría propia].....	44
Fig. 13. Reproductor de media. [autoría propia].....	48
Fig. 14. Pantalla de lectura del diccionario. [autoría propia].....	55
Fig. 15. Pantalla de estadísticas.	56
Fig. 16. Esquema de navegación. [autoría propia].....	64
Fig. 17. Estadísticas de uso por versión en Nicaragua. [53]	74
Fig. 18. Diagrama de clases. [autoría propia]	i
Fig. 19. Diagrama de actividad de formulario de ingreso. [autoría propia]	ii
Fig. 20. Diagrama de actividad de inicio de sesión. [autoría propia]	ii
Fig. 21. Diagrama de actividad pantalla principal. [autoría propia].....	iii
Fig. 22. Diagrama de actividad reproductor de media. [autoría propia]	iii
Fig. 23. Diagrama de base de datos. [autoría propia]	iv
Fig. 24. Diagrama de componentes, función externa. [autoría propia].....	iv
Fig. 25. Diagrama de componentes, función interna. [autoría propia].....	v

Índice de tablas

TABLA 1. REQUERIMIENTOS FUNCIONALES. [autoría propia].....	27
TABLA 2. REQUERIMIENTOS NO FUNCIONALES. [autoría propia]	28
TABLA 3. PLANIFICACIÓN, FASE DE EXPLORACIÓN. [autoría propia]	30
TABLA 4. PLANIFICACIÓN, FASE DE INICIACIÓN. [autoría propia]	31
TABLA 5. PLANIFICACIÓN, FASE DE PRODUCCIÓN. [autoría propia].....	31
TABLA 6. PLANIFICACIÓN, FASE DE ESTABILIZACIÓN. [autoría propia].....	32
TABLA 7. PLANIFICACIÓN, FASE DE PRUEBAS DEL SISTEMA. [autoría propia]	33
TABLA 8. DISEÑO DE UNA STORYCARD. [autoría propia].....	38
TABLA 9. STORYCARD INICIO DE SESIÓN. [autoría propia].....	40
TABLA 10. STORYCARD REGISTRO DE USUARIO. [autoría propia]	42
TABLA 11. STORYCARD LISTA DE PALABRAS. [autoría propia]	44
TABLA 12. STORYCARD RECONOCIMIENTO DE VOZ. [autoría propia].....	46
TABLA 13. STORYCARD REPRODUCTOR DE MEDIA. [autoría propia]	48
TABLA 14. STORYCARD CÁLCULO DE RESULTADOS. [autoría propia]	51
TABLA 15. STORYCARD DE EVALUACIÓN DE LA PRONUNCIACIÓN.....	52
TABLA 16. STORYCARD GUARDADO DE PROGRESO. [autoría propia]	54
TABLA 17. STORYCARD DICCIONARIO. [autoría propia]	55
TABLA 18. STORYCARD DE ESTADÍSTICAS. [autoría propia]	57
TABLA 19. STORYCARD DE GRAFICACIÓN DEL SONIDO. [autoría propia].....	58
TABLA 20. VERIFICACIÓN DE VISTAS. [autoría propia]	66
TABLA 21. CASO DE PRUEBA, GRABACIÓN Y REPRODUCCIÓN. [autoría propia] ...	68
TABLA 22. CASO DE PRUEBA, GUARDADO DEL PROGRESO. [autoría propia]	69
TABLA 23. VERIFICAR EVALUACIÓN DE PRONUNCIACIÓN. [autoría propia]	69
TABLA 24. CASO DE PRUEBA DE VERIFICACIÓN DE LAS ESTADÍSTICAS. [autoría propia].....	70
TABLA 25. PROPUESTA 1 DE FACTIBILIDAD TÉCNICA. [autoría propia].....	71
TABLA 26. PROPUESTA 2 DE FACTIBILIDAD TÉCNICA. [autoría propia].....	72
TABLA 27. PROPUESTA 1 DE FACTIBILIDAD ECONÓMICA. [autoría propia].....	75
TABLA 28. PROPUESTA 2 DE FACTIBILIDAD ECONÓMICA. [autoría propia].....	76
TABLA 29. ESTIMACIÓN DEL ESFUERZO CON PLANNING POKER. [autoría propia]	78

1. INTRODUCCIÓN

Como contribución al desarrollo de habilidades de expresión oral en niños con discapacidades auditivas, en el presente trabajo, se desarrolló una **aplicación móvil de reconocimiento de voz**, que contribuye a mejorar las habilidades de expresión oral en niños con hipoacusia.

Dicha aplicación provee ejemplos y actividades que los usuarios puedan seguir y practicar de manera autónoma, con poca intervención de especialistas. Utilizando la comparación entre sonidos de la vocalización introducidas por el usuario y las de la aplicación, se otorgará una evaluación cualitativa con visualización gráfica y de código de colores.

En la sociedad actual, las personas con discapacidad auditiva sufren de exclusión, por no poder expresarse verbalmente, consecuencia de la pérdida del sentido de la audición [1]. Estos niños que nacen sordos o quedan sordos a temprana edad (antes de aprender a hablar), enfrentan múltiples dificultades en la comunicación oral, aunque su laringe y cuerdas vocales estén sanas. No oír les impide saber cómo expresar correctamente los fonemas verbales (se les dificulta articular palabras). Este es un problema que requiere mucha ayuda profesional de especialistas en logopedia, y un sin número de horas dedicadas a la práctica de ejercicios de articulación de palabras [2].

En base a las necesidades anteriormente planteadas y a las de portabilidad, se eligió el smartphone como la opción más económica - portable, siendo un recurso más adecuado, al tener las comodidades de traer integrados los periféricos necesarios para una aplicación de reconocimiento de voz, y al hecho de poder ser utilizado tanto por docentes y estudiantes en cualquier momento. Por lo cual se decantó por una aplicación móvil, para el sistema operativo Android, que satisfaga lo antes descrito [3].

Este software es soportado por la plataforma de desarrollo Android Studio, programada en lenguaje JAVA + PHP y utilizará una base de datos MySQL. Esta aplicación como recurso didáctico les permite practicar la pronunciación de letras

y palabras, de forma independiente, o con la ayuda de un instructor. Se prevé de utilidad para estudiantes de los primeros años de educación especial (preescolar / primaria).

Teniendo en cuenta el nivel educativo, el software evalúa la habilidad fonética del alumno, otorgándole una calificación cualitativa de: Excelente, Muy Bueno, Bueno, Regular, Deficiente. Cada una se representará con un patrón de colores de: verde, verde claro, verde lima, amarillo, rojo (respectivamente). La calificación otorgada dependerá del grado en que la pronunciación lograda por el estudiante se aproxime a lo correcto. Para ello se aplicarán técnicas lúdicas en los procesos de enseñanza - aprendizaje.

En el desarrollo de este trabajo monográfico, se contó con la asesoría y colaboración de intérpretes y maestros especialistas en lenguaje de señas, del Centro de Educación Especial Melania Lacayo con experiencia en el trabajo con niños con discapacidades auditivas.

2. ANTECEDENTES

En Nicaragua actualmente, el proceso de enseñanza - aprendizaje en personas de discapacidad auditiva, solo se da por el medio gesticular el lenguaje de señas nicaragüense. Por lo tanto, cualquier tipo de enseñanza verbal, que pueda ser provista de manera ordenada y con un plan de por medio, es inexistente por parte de las instituciones.

Varias tecnologías en el ámbito internacional, tanto hardware, como software, han sido creadas para ayudar a personas con discapacidad auditiva (parcial o completa) [4]. Estas promueven el aprendizaje verbal y la comprensión del sonido, aumentando la capacidad de comunicación de la persona, más allá de los lenguajes por señas o de escritura. Desafortunadamente en Nicaragua, solo se encuentran aplicaciones centradas en la parte gesticular, no verbal.

Un problema general de estas aplicaciones es la gran necesidad de intervención de un tutor a la hora de ocupar alguna de estas herramientas. También la limitada cantidad de soluciones que ofrecen. Otros problemas que presentan son: el costo, las limitaciones del software, contenido obsoleto o no completo y diseños poco amigables para la interacción con el usuario [5].

Speech viewer III [6] es un programa en desuso creado por IBM en 1997 para Windows 98. Está centrado en ejercicios interactivos para ayudar en todos los apartados en la característica del habla (tono, intensidad, duración y sonoridad), pero tratando cada uno de manera individual y solo de soporte para lo que el tutor pueda enseñar.

Un sintetizador para el habla [1] fue desarrollado en Venezuela en el año 2008, como dispositivo para ingresar por teclado palabras que luego se reproducirán, por medio de un sintetizador que simula la voz con las reglas de la fonética aplicadas.

Globus 3 [7] es una aplicación de escritorio lanzada en el 2009, pensada para que las personas con limitaciones en su capacidad de audición puedan ver en la pantalla del monitor, que han emitido un sonido y el volumen con el que lo han emitido. Está enfocada en ejercitar las articulaciones del habla y no puede

utilizarse más que como un soporte limitado por la necesidad de supervisión de un tutor, para que compare los sonidos producidos.

El dispositivo interactivo para el desarrollo de la comunicación oral [8] fue creado en Guatemala en el año 2018 con la capacidad de realizar ritmos en base a un compás (2/4, 3/4 y 4/4), de manera visual y medir los decibeles producidos por la voz; todo mediante leds, sensores, y una placa de Arduino. Es usado de soporte como parte de la enseñanza de lo que es el ritmo y para aprender el control de la intensidad del volumen de la voz.

La aplicación móvil Señas Nicas [9], fue desarrollada en el 2018 con el propósito de ser usado como material didáctico y diccionario de lenguaje de señas. Tiene un catálogo de las señas más comunes, siendo ordenados por letras y categorías de palabras. Al ser gesticulaciones, no se entra en ningún momento en el apartado verbal del aprendizaje

La aplicación móvil "Señas y Sonrisas – Nicaragua" [10] fue desarrollada en el año 2019. Es el diccionario de lenguaje de señas de Nicaragua más completo hasta la fecha, con el propósito de servir como un recurso didáctico para familias con niños sordos. Además, incorpora una función de prueba de conocimiento de señas; sin embargo, no se enfoca en la práctica de la vocalización.

Aplicación móvil y dispositivos centrados en el Internet de las cosas para niños con discapacidad auditiva [11] fue creado en el año 2020 para emplear una variedad de actividades como ejercicio para la memoria visual, la atención selectiva, la percepción auditiva y la orientación espacial. La parte de interacción física y digital se trabajó en actividades de memoria visual y percepción auditiva.

Teniendo en consideración los casos anteriores, se puede apreciar que estas aplicaciones, enfocadas en ayudar a personas con discapacidad auditiva, no tienen como objetivo la enseñanza del habla, sino ser un complemento del aprendizaje y a veces una alternativa a la habilidad de vocalizar. No identificamos una aplicación centrada en el habla, como se propone en este trabajo monográfico.

3. JUSTIFICACIÓN

Las personas que carecen del sentido auditivo desde su nacimiento, o que lo pierden a temprana edad, por lo general, ven limitadas sus posibilidades de aprender a expresarse oralmente. Superar esa limitación requiere el apoyo de especialistas en el tratamiento y corrección de trastornos que afectan a la voz, la pronunciación y el lenguaje oral [12]. También necesitan muchas horas de práctica supervisada. Disponer de una aplicación móvil como recurso didáctico, les permitiría practicar con mínima supervisión. Además, potenciaría la capacidad de atención por parte del maestro, tanto en calidad como en cantidad de discentes atendidos [5].

Muchas de estas personas, pueden emitir sonidos, pero no desarrollan esa habilidad. Esto suele provocar su exclusión, interfiriendo con su desarrollo e integración social [1].

Por tanto, se justifica el desarrollo de una aplicación móvil, enfocada a niños y niñas con discapacidad auditiva, con la que puedan realizar, con cierto nivel de autonomía, ejercicios que les permitan aprender a pronunciar vocablos, pasando por los niveles educativos, elemental (preescolar), hasta los grados de la educación básica (primaria) teniendo en cuenta las limitaciones que representa la discapacidad [12]. Por tal motivo, en este trabajo, solo se abarcará el abecedario, y palabras simples. Se realiza la introducción por voz de las palabras que se muestren en la aplicación, ayudando al usuario en su pronunciación, como una alternativa al cuadernillo de primeras letras.

Teniendo en cuenta que el uso de la aplicación debe estar disponible en cualquier lugar, se optó, por los smartphones, como el dispositivo para la ejecución del software, el cual integra todos los periféricos necesarios para el uso del reconocimiento de voz, sin necesidad de otros externos. Además, los smartphones son el dispositivo con acceso a Internet más adquirido por la población, en comparación con dispositivos, como laptops o computadoras de sobremesa [13].

El espectro gráfico del sonido, medidor de tiempo, imágenes y videos de la fonética en las palabras, serán las herramientas a las que el usuario accederá para comprender mejor la forma en cómo debe hacer las articulaciones en la boca y cuanta fuerza debe poner en el volumen - duración de su voz [8], [14].

Con esta aplicación, se reducirá la presión sobre los docentes, que deben atender simultáneamente a muchos niños con problemas de audición. Disponer de ella, reducirá el tiempo que los instructores necesitan para tratar con cada alumno. También se aliviarán los costes en la adquisición de los recursos didácticos. Muchas escuelas especiales y padres de familia no tienen la capacidad económica para adquirir las herramientas de aprendizaje que, usualmente, se necesitan. Además, con esta aplicación, se tendrá, una alternativa a la adquisición de otras herramientas (analógicas) que suelen utilizarse.

4. OBJETIVOS

4.1. Objetivo General

- Desarrollar una aplicación móvil que permita la mejora de habilidades de expresión oral, en niños con discapacidad auditiva, en los niveles de enseñanza elemental, básica de un centro de educación especial.

4.2. Objetivo Especifico

- Explorar las técnicas de enseñanza para personas con discapacidad auditiva.
- Inicializar la planificación de la aplicación según las técnicas de enseñanza identificadas y los requerimientos propuestos por educadores especialistas en el desarrollo de habilidades de expresión oral en la niñez.
- Producir la aplicación móvil a través de la metodología Mobile-D, para brindar una organización en el proceso de elaboración del software, utilizando las herramientas de desarrollo Android studio, php y la base de datos MySQL.
- Estabilizar todas las funciones integradas y aprobadas anteriormente por el cliente
- Testear la versión estable de la aplicación móvil, utilizando los criterios de usabilidad del software, que valida su correcto funcionamiento.

5. MARCO TEÓRICO

El contenido presente en este marco teórico fue elaborado teniendo como base la línea de pensamiento sistemática, producto de la **metodología de investigación (tecnológica aplicada)** seleccionada [15]. Como parte de la naturaleza del trabajo monográfico, se definen conceptos de lingüística, sonidos, salud, enseñanza-aprendizaje especial e ingeniería de software necesarias para la aplicación.

5.1. Investigación enfocada en sistemas

Una **investigación** es un proceso formal, intencionado que se lleva a cabo mediante el método científico del análisis, y es la herramienta principal para llevar a cabo cualquier estudio formal.

Un **sistema** es un conjunto de elementos que se interrelacionan para funcionar como un todo y contribuir a un determinado objetivo. Esto se ajustaría a la parte de ingeniería de software en la cual se requiere enfocar la investigación.

Para cumplir con el requisito anterior del enfoque. Se decidió seleccionar un tipo, de las diferentes formas de metodologías para llevar una investigación. Las cuales se divide en tipo y en nivel. Así el método seleccionado, **investigación tecnológica aplicada** es compuesto por las partes [15], [16]:

Investigación de “tipo” tecnológica: Es el tipo de investigación que resuelve problemas en los procesos técnicos, buscando eficiencia. Aprovechando el conocimiento científico proveniente de su contraparte la **investigación básica**; que solo tiene como propósito ampliar el conocimiento científico. Por lo cual está orientada al descubrimiento, despreocupándose de su aplicación práctica.

Investigación de “nivel” aplicada: esta parte aplica los conocimientos teóricos a determinadas situaciones. La filosofía de este nivel, es buscar conocer para hacer, actuar, construir y modificar, preocupándose por la aplicación inmediata. El conocimiento utilizado en este nivel, viene de su contraparte la investigación experimental que se centra en crear conocimientos nuevos para mejorar los existentes.

5.2. Discapacidad auditiva

Una persona con discapacidad auditiva se define como: “aquella que no puede escuchar normalmente debido a algún tipo de anormalidad en el órgano de la audición: el oído” [17]. Sus causas pueden ser accidental, congénita, hereditaria o genética, siendo ésta última, la más importante y poco previsible.

5.3. El sonido y sus características

El sonido se define como “la vibración mecánica que se transmite con pequeñas variaciones de presión a través de un medio elástico” [14]. Este sonido se transmite a través de ondas en el aire (el medio elástico gaseoso), aunque también puede ser en un líquido o un sólido.

El sonido este compuesto por cuatro características: tono, duración, intensidad y timbre; aunque solo los tres primeros deben ser tomados en cuenta para personas con ausencia o pérdida de audición [6], [14].

1. **Tono:** distingue entre la agudeza o gravedad de un sonido (sonido alto o bajo). Esta es determinada por su frecuencia, siendo más aguda si su frecuencia es alta y más grave si es baja.

Frecuencia sonora (F): es la cantidad de ciclos por segundos que completan en su oscilación las partículas del aire. Su unidad de medida es el Hertzio (Hz).

2. **Duración:** es la característica del sonido que nos dice la diferencia entre un sonido corto y un sonido largo. Depende de la persistencia de la onda: cuánto dura la onda hasta que se desvanece.
3. **Intensidad (I) sonora:** indica la diferencia entre un sonido fuerte y un sonido suave. Se define como “el valor promedio de energía que cruza un área en un intervalo de tiempo”. Se mide en unidades llamadas decibelios (dB).
4. **Timbre:** “Sensación auditiva compleja que permite percibir la estructura interna de sonidos compuestos” [14] Por medio de esta característica se

permite diferenciar cuando una misma nota es producida por diferentes medios, como instrumentos de madera, metal, o en este caso la voz humana.

5.4. Espectro gráfico del sonido

El espectro del sonido es la representación gráfica de las ondas sonoras a determinada frecuencia e intensidad. Está diseñado para hacer un registro de estos parámetros en un instante de tiempo, por lo que esta herramienta es esencial para que una persona con discapacidad auditiva pueda guiarse al momento de reproducir un sonido.

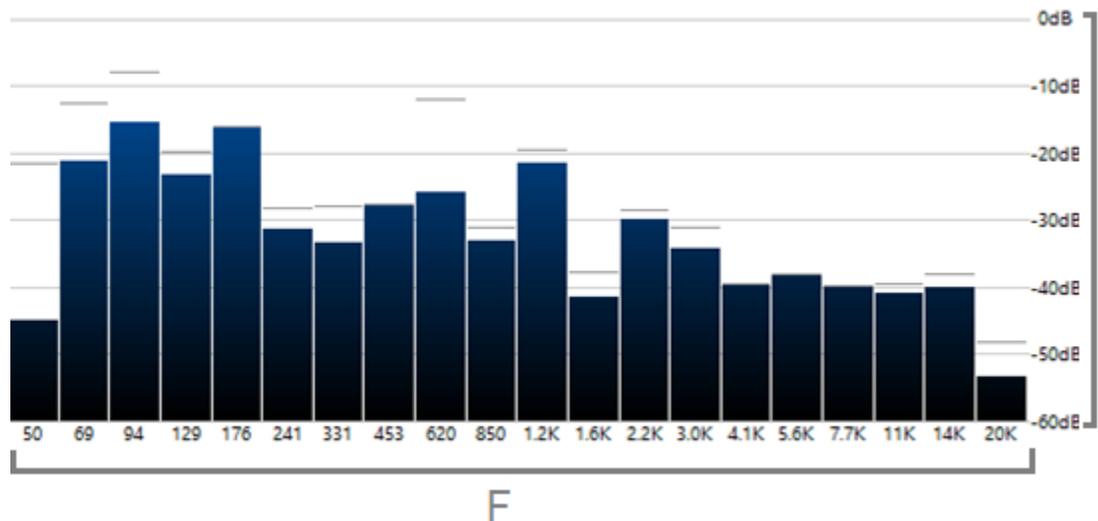


Fig. 1. Espectro gráfico del sonido. [Autoría propia]

Entre los parámetros que lo componen:

Frecuencia (F): El rango de frecuencia representada en una gráfica es entre 20 Hz y 20000 Hz, el cual es el rango normal de audición de un oído humano.

Intensidad (I): Representada en el estándar de escala relativa de decibeles utilizada en los equipos de audio. dada por la formula [18]:

$$20 * \log (X/Y) \quad (1)$$

Donde:

- $\text{Log}(X/Y)$ es la diferencia entre dos presiones de sonido X e Y en un valor entre 0 al 1, siendo 1 el máximo, representando el 100% del volumen que puede proveer cualquier dispositivo de entrada de sonido.
- 20: Resulta de elevar al cuadrado un valor logarítmico teniendo en cuenta la presión de un área, las cuales se miden en unidades cuadradas.

5.5. Discapacidad auditiva según grado de pérdida auditiva

La escala del grado de pérdida de audición es la severidad de dicha discapacidad, la cual propone una medición en decibelio (dB) de la sensibilidad del oído hacia el sonido. Una persona normal pierde de entre 0 a 20 dB en su audición durante todo su periodo de vida, más que esto se considera una discapacidad.

Todo dispositivo utiliza una clasificación para la configuración de sus parámetros o la forma en que funcionara la herramienta. Por norma general los softwares con el objetivo de tratar esta discapacidad usan la escala del grado de pérdida de audición, esto debido a la detección de síntomas que propone la escala y su compatibilidad con la variable decibel (dB) en dispositivos como micrófonos, y las posibles representaciones visuales que se puede proveer por pantallas [12].

A continuación, se presenta los síntomas de una perdida mayor a 20 dB, objetivos de la aplicación [2], [4], [12]:

1. Discapacidad auditiva Leve (20 – 40 dB)

- Los pacientes con discapacidad auditiva leve pueden presentar alteraciones fonéticas.
- Presentan dificultades para escuchar la voz de baja intensidad.

2. Discapacidad auditiva moderada (40 – 70 dB)

- Dentro de este grado de pérdida auditiva el lenguaje está caracterizado por problemas articulatorios, de nasalización, de prosodia, presentan una intensidad de voz inestable y dificultades sintácticas.
- El paciente con hipoacusia moderada presenta dificultad para escuchar una conversación normal.
- También son frecuentes las dificultades de comunicación y en el proceso de enseñanza-aprendizaje, como en la lectoescritura.

5.6. Fonética en discapacidad auditiva

La fonética estudia, desde distintos puntos de vista, los sonidos del habla en general (fonos), de cualquier lengua, en su carácter físico. Esta a su vez se divide en otras dos ramas.

- La fonética acústica, la cual estudia las propiedades físicas, las ondas sonoras generadas por los órganos de fonación y propagadas por el aire.
- La fonética articulatoria, que clasifica los sonidos del habla a partir del modo como son producidos por el aparato fonador [19].

5.6.1. Fonemas

Los fonemas son la unidad sonora mínima por la que se puede distinguir una palabra de otra, esta dependerá del entorno y el idioma en el que se hable. Representan los sonidos del habla y la articulación que conlleva el sonido. Está representada por una vocal o consonante entre dos plecas, como puede ser: /a/, /d/.

Un fonema puede tener distintos sonidos variantes llamadas alófonos, los cuales se representan en un corchete tales como: [s], [b].

Fonema	Alófono	Ejemplos
//	[l]	loro ['lo-ro] alba ['al-ba]
	[ll]	alto ['a _l -to] aldea [a _l -'de-a]
/s/	[s]	sol ['sol]
	[z]	mismo ['miz-mo] desde [ðez-'ðe]

Fig. 2. Ejemplo de descomposición de fonemas. [20]

En el ejemplo se puede observar cómo loro y alto tienen el fonema // la cual al ser pronunciadas producen sonidos diferentes en las dos palabras lo que se traducen en dos alófonos [l] distintos.

Esta teoría debe ser explicada y aplicada por un experto de manera breve por un medio audiovisual, ya que debe hacer énfasis en el sonido que debe producir y como articular la boca, no en el significado [19].

5.6.2. Método de enseñanza de comunicación total de Benson Schaeffer

El material audiovisual debe estar basado en un método de enseñanza enfocado en personas discapacitadas, teniendo en cuenta el rango de pérdida de audición. Se seleccionó un método oral combinado con gestual para ser aplicado en los ejercicios de la aplicación.

Este método se conoce como el **sistema de Comunicación Total, de Benson Schaeffer**. Es uno de los más utilizados para tratar con trastornos que afectan la comunicación en general. En este método, se aprovecha la audición residual para desarrollar el lenguaje oral, pudiendo ajustarse a métodos alternativos, si se necesita. En este sistema, se utilizan de forma simultánea lenguaje oral y signado, repitiendo los signos tantas veces como el número de sílabas de la palabra [4].

5.7. Reconocimiento de voz

Por definición, es una tarea de reconocimiento de patrones y de los sistemas biométricos. En este caso, la señal de voz, que es analizada para extraer la información relevante una vez que ha sido digitalizada [21].

5.7.1. Reconocimiento automático del habla (ASR)

Es el proceso mediante el cual los sonidos provenientes del aparato productor del habla (fonación y articulación), son transcritos a un conjunto de símbolos ortográficos compatibles con las reglas gramaticales de la lengua objetivo de que se trate. En nuestro caso, el idioma español [22], [23], [24]. Se compone de tres partes:

- **El diccionario fonético**, es el vocabulario de palabras junto con la división fonética de cada una de ellas, con el cual va a operar el sistema del ASR
- **El modelo del lenguaje**, representa la distribución de probabilidad de la palabra y el orden de la misma en el sistema
- **El modelo acústico**, representa la distribución de probabilidad de los fonemas en la señal de audio

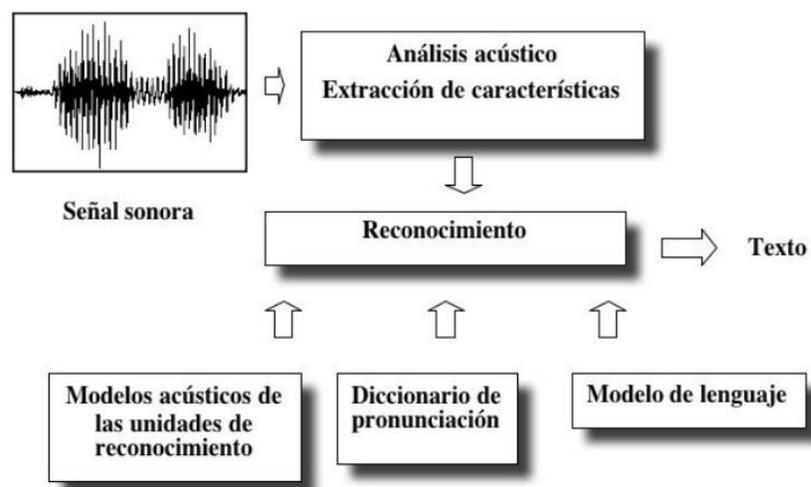


Fig. 3. Esquema del reconocimiento automático del habla. [25]

El ASR también tiene varias formas de realizar este mismo proceso, pero se han identificado dos aplicables a este trabajo monográfico:

Los reconocedores de palabras aisladas (IWR, Isolated Word Recognition): el reconocimiento se realiza sobre palabras completas emitidas de forma aislada entre sí, donde se compara la señal introducida con el modelo almacenado

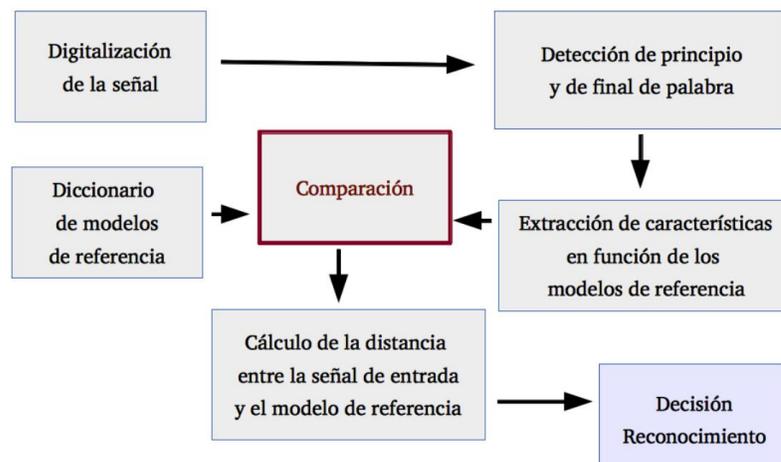


Fig. 4. Esquema de reconocimiento de palabras aisladas. [25]

Los reconocedores de voz o discurso continuo (CSR, Continuous Speech Recognition): para un reconocimiento complejo. Realizan la tarea atendiendo a unidades inferiores a la palabra (fonemas, dífonos, etc.) y sobre frases completas, sin necesidad de establecer silencios entre las palabras que las constituyen.

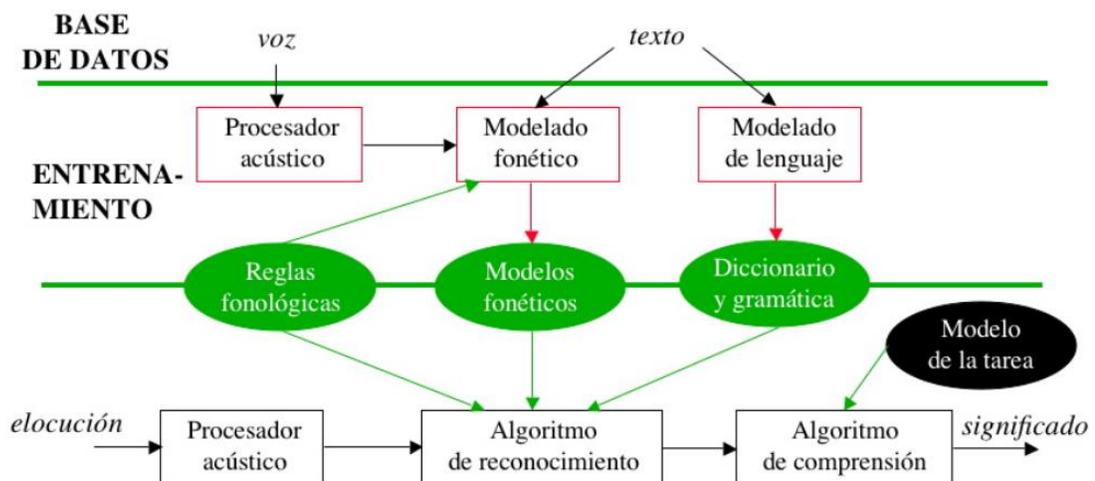


Fig. 5. Esquema de modelo de habla continua. [25]

5.8. Sistema informático

Un sistema informático es definido como un conjunto de partes relacionadas entre sí; en el cual se emplea un ordenador (dispositivo móvil en este caso), en el que se programa dispositivos para la captura, el almacenaje y el procesamiento de datos. También son partes de este, los usuarios que lo usan y los periféricos que dan soporte [26].

5.8.1. Aplicación móvil

Se entiende por aplicación móvil, aquellas aplicaciones que estén desarrolladas para ser ejecutadas en el ecosistema móvil. Las características de los ecosistemas están dadas por el sistema operativo, los dispositivos de entrada de información que tiene el hardware, el rendimiento del dispositivo y los propios usuarios a los que están enfocados [27]. En este caso el entorno móvil cumple con los requisitos necesarios de portabilidad, bajo costo de hardware, dispositivos de entrada y acceso a internet, todo sin necesidad de algún hardware externo de terceros.

5.8.2. Base de datos de modelo relacional

En una base de datos de modelo relacional los datos están esparcidos en varias partes de almacenamiento separadas (tablas) en vez de poner todo en un solo espacio de almacenamiento [28].

Este tipo de base de datos evitan problemas como datos duplicados, y el tener que lidiar con una desorganización al tener toda la información junta. La parte relacional está dada con el uso de las “claves primarias y foráneas” que se enlazan los datos que están divididos en tablas separadas. Con las claves primarias se distinguen los datos irrepetibles de cada tabla, mientras que con la foránea se identifica la relación que tienen con la información de otras tablas.

5.8.3. Modelo cliente servidor

El modelo cliente servidor hace referencia a dos procesos involucrados: el de la maquina cliente y el de la maquina servidor. Esta comunicación ocurre cuando el

proceso cliente envía un mensaje a través de la red al proceso servidor esperando su respuesta. Cuando el servidor recibe la solicitud, lleva a cabo la tarea o busca la información en la base de datos del servidor y los devuelve al cliente [29].

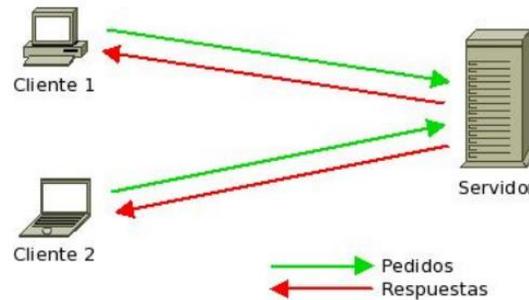


Fig. 6. Modelo cliente/servidor. [30]

Este modelo permite diversificar el trabajo que realiza cada aplicación, así el cliente no se sobrecarga, cosa que ocurriría si ellos intentaran desempeñar todas las funciones, también ayuda en el aspecto organizativo debido a la centralización de la información y la repartición de las responsabilidades.

5.9. Herramientas de desarrollo

5.9.1. Plataforma de desarrollo Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Es un lenguaje orientado a objetos utilizando la tecnología de C++. Es el lenguaje de programación principal, soportado por el (IDE) Android Studio [31], [32].

Además, es distribuido y hace uso de librerías de rutinas para usar protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales. Esta característica permite usarlo en aplicaciones con servicio de web y con el modelo cliente/servidor [33].

5.9.2. El lenguaje PHP

PHP Group creó en 1995 un lenguaje de hipertexto al cual dieron su nombre: PHP (acrónimo de Hypertext Pre-processor). Es un lenguaje de script que se interpreta

en el lado del servidor. Se usa en páginas web dinámicas, embebidas en HTML y ejecutadas en el servidor. Para su funcionamiento necesita tener instalado un servidor de Apache. Utiliza la combinación de sintaxis de C, Java y Perl con algunas características específicas de estas [34].

5.9.3. Suite de herramientas XAMPP

El recurso XAMPP es una distribución de Apache Servidor web HTTP de código abierto, completamente gratuita, que contiene: PHP, Perl y gestores de base de datos ya integrados MySQL, SQLite y MariaDB. Con este programa se realizará todo lo relacionado al servicio web de la aplicación. Además, incluye la opción de servicio FTP para almacenar audios y videos. Fue elegido por su popularidad, el paquete de herramientas del que dispone y por el hecho de ser de código abierto [35].

5.9.4. Herramienta de gestión phpMyAdmin y Base de datos MySQL

La herramienta phpMyAdmin es un recurso con licencia de software libre, destinada a manejar la administración de MySQL en la Web. Las operaciones de uso frecuente (administración de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc.) se pueden realizar a través de la interfaz de usuario, mientras aún tiene la capacidad de ejecutar directamente cualquier instrucción SQL [36]. El objetivo de esta base de datos en esta aplicación es el servicio web, por lo que solo se tendrá acceso a esta por medio de conexión a internet.

5.9.6. Entornos de desarrollo integrados (IDE)

- **Android Studio**

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA, el cual es otra plataforma para la creación de aplicaciones. Al ser el entorno de desarrollo móvil de Google tiene una gran cantidad de documentación actualizada, además de tener soporte

de la mayoría de las herramientas de desarrollo de esta empresa. Entre los lenguajes más importantes soportados por este se encuentran Java, Kotlin, C++ [31].

- **Visual Studio Code**

Visual Studio Code es un editor de código optimizado con soporte para operaciones de desarrollo como depuración, ejecución de tareas y control de versiones desarrollado por Microsoft". Su objetivo es proporcionar las herramientas para realizar los ciclos de compilación y depuración de código y deja las operaciones de trabajo más complejas para los IDE con funciones más completas, como Visual Studio IDE [37]. Este dispone de todo tipo de extensiones que proveen de una mayor cantidad de opciones para adaptarse a cualquier necesidad.

5.9.7. Github

Herramienta para el control de versiones y repositorio de código en la nube. Está destinado para que varios desarrolladores puedan trabajar en un mismo proyecto y ver las ediciones en tiempo real, también cuenta con el establecimiento de permisos y roles, además de la opción de poder hacer una recuperación de versiones anteriores. Tiene soporte en la mayoría de los IDE y trabaja en conjunto con Git, aplicación para control de versiones con la diferencia de que es un repositorio local.

5.9.8. API de reconocimiento de voz Vosk

Vosk es una biblioteca práctica de reconocimiento de voz, recién desarrollada en 2020, que viene con modelos preparados, scripts, prácticas y proporciona reconocimiento de voz listo para usar para diferentes plataformas como aplicaciones móviles o Raspberry Pi, además de que cuenta con más de 20 lenguajes (incluido español) con sus respectivos modelos. Trabaja con el modelo de "Reconocimiento automático del habla (RHA)" [38] y tiene la ventaja de estar listo para usar después de la instalación

Esta API funciona bajo la licencia de Apache-2.0 por lo que se es libre de modificar y hacer uso comercial. Puede utilizarse de manera local en Android, iOS, Raspberry Pi y en servidores con Python, Java, C# y Node.

5.9.9. AmplitudeGraphView

Biblioteca gráfica para trazar cualquier tipo de valor con respecto al tiempo. Específicamente fue diseñado para captar señales de audio en tiempo real y graficarlas de acuerdo al espectro grafico del sonido de “amplitud con respecto al tiempo”.

Esta herramienta es utilizada para que el usuario pueda obtener una interacción visual del sonido que producen en la aplicación, debido a que solo podrán utilizar partes de sus restos auditivos por la naturaleza de su discapacidad. Esta librería provee de una guía visual para producir el volumen correcto de la voz, graficando la onda de sonido en tiempo real y al mismo tiempo confirmando que se han producido sonidos.

5.9.10. Cliente HTTP Retrofit

Es una librería para Android y Java, para hacer peticiones de llamadas de red y obtener datos de forma automática, lo que facilita el realizar peticiones a una API HTTP y obtener su respuesta, todo en lenguaje Java. Se va utilizar para la carga masiva de datos. Es open source, trabajando bajo la licencia de Apache-2.0 [39].

5.9.11. Cliente HTTP Volley

Es una librería desarrollada por google para la optimización de envíos de peticiones HTTP desde la aplicación a servidores externos. **Volley** es sencillo de implementar, debido al uso de menos código [40]. No es muy efectivo para la carga masiva de datos, por lo que se usara el otro cliente HTTP antes mencionado **Retrofit** para esta tarea.

Todo lo anterior propuesto en el marco teórico, se ha tomado en consideración, de acuerdo a la metodología de desarrollo, y el hecho de ser una aplicación móvil.

Hay que tomar en cuenta que, aunque son los conceptos y herramientas esenciales, estos pueden expandirse, para adaptarse a las necesidades del cliente o del desarrollo mismo.

5.9.12. Editor de video DaVinci Resolve

DaVinci Resolve es una herramienta profesional para editar videos, desarrollada por Blackmagic Design. Ofrece edición de video, corrección de color, efectos visuales y postproducción de audio. Con una interfaz fácil de usar y funciones avanzadas, permite crear proyectos de alta calidad. Está disponible en una versión gratuita y una versión de pago, siendo accesible tanto para principiantes como para profesionales [41].

Como conclusión, este marco teórico se basa en una línea de pensamiento sistemática, siguiendo la metodología de investigación tecnológica aplicada. Se definen conceptos necesarios para la aplicación, como lingüística, sonidos y enseñanza especial. Se seleccionaron herramientas para el desarrollo de este trabajo monográfico a través de esta investigación, que se compone en dos partes: investigación tecnológica y aplicada. La primera resuelve problemas técnicos y de procesos, mientras que la segunda aplica conocimientos teóricos a situaciones prácticas para construir y mejorar.

6. METODOLOGÍA Y DESARROLLO

Se valoraron varias metodologías y enfoques que se consideraron aplicables en este trabajo monográfico. Tomando en cuenta que se quiere desarrollar una aplicación móvil con fines didácticos en educación especial, se eligió el tipo de investigación a seguir, resultando la metodología de investigación tecnológica aplicada y de desarrollo Mobile-D, las que mejor se adecuan a las proyecciones a futuro de nuestro trabajo [15].

6.1. Investigación tecnológica aplicada

se refiere al uso de conocimientos científicos y tecnológicos para resolver problemas prácticos y mejorar procesos o productos existentes. A diferencia de la investigación básica, que busca expandir el conocimiento sin un objetivo práctico inmediato, la investigación aplicada tiene un enfoque directo en la solución de problemas específicos y la implementación de innovaciones tecnológicas [15].

En este caso, se tomó un **enfoque sistemático** que afronta los problemas en su completa extensión. A través de la forma de pensamiento basada en su totalidad, se estudia la relación de todas sus partes y de las propiedades resultantes [15], [16].

Este enfoque fue seleccionado debido a los distintos tópicos que presenta este trabajo monográfico, como son la salud, lingüística, sonido y la ingeniería de software, todas las cuales se relacionan entre sí, partes del mismo problema, siendo demasiado grande para resolverlo mediante otro esquema como el reduccionista, el cual es aplicado en la mayoría de las investigaciones.

En el tipo de investigación utilizado en este trabajo, no es obligatorio formular una hipótesis o realizar experimentación, ya que no se están desarrollando nuevas tecnologías. Este trabajo monográfico tiene las siguientes características [42]:

Se implementan tecnologías existentes:

En esta investigación se trata de adaptar o mejorar una tecnología que ya ha sido validada previamente, la experimentación es limitada o innecesaria en términos de validación técnica.

integración de sistemas:

Se integran diferentes sistemas o tecnologías ya probadas, el foco está en garantizar la interoperabilidad entre sistemas más que en realizar pruebas experimentales.

6.2. Mobile-D

Mobile-D es una metodología ágil de desarrollo para dispositivos móviles en él, se basa en otras metodologías como lo puede ser: Extreme Programming (XP), Crystal Methodologies y Rational Unified Process (RUP), pero aplicada de una forma más estricta. Tomando de XP las prácticas de desarrollo, Crystal la escala de los métodos y RUP como base en el diseño del ciclo de vida. Pudiendo así obtener distintas fases en el desarrollo de la misma. Como lo pueden ser: Exploración, iniciación, producción, estabilización y pruebas [27].

6.3. Ciclo de vida

El objetivo de Mobile-D es conseguir ciclos de desarrollos muy rápidos en equipos muy pequeños (de no más de diez desarrolladores) trabajando en un mismo espacio físico. Según este método, trabajando de esa manera se deben conseguir productos totalmente funcionales en un corto periodo de tiempo.



Fig. 7. Ciclo de vida de Mobile-D. [27]

Cada una de estas fases (excepto la inicial) tiene un día de planificación y una fecha de entrega [27], [43].

6.3.1. Exploración

Esta Primera fase se centra en la planificación y los fundamentos del trabajo monográfico. Aquí, define el alcance del trabajo monográfico y su estructura junto con la funcionalidad que desea alcanzar [43], [44].

Para este trabajo monográfico se requirió de diferentes reuniones con el público de interés en el colegio Melania Lacayo de Masaya esto con el fin de identificar las necesidades, los problemas y el público específico al que va dirigida la aplicación. Según lo dicho en las reuniones, a los niños con hipoacusia no se les prioriza la ayuda en el desarrollo de las habilidades de vocalización y solo se les enseña el lenguaje de señas nicaragüense. Por lo tanto, para el desarrollo de la aplicación fue necesaria de la ayuda del centro educativo y los maestros que proveyeron de la información necesaria sobre los diferentes puntos que se debieron abordar entre los cuales tenemos:

- **El tipo de aplicación que necesitan y los servicios que tendrán disponibles:**

En el caso de la aplicación se les propuso una de tipo móvil debido al fácil acceso y a la disponibilidad del hardware. Aparte de la funcionalidad principal de reconocimiento de voz también se concluyó que debería incluir guardado en la nube, presentación del progreso por medio de estadísticas, y una opción para entrar como invitado por si el usuario no tuviera acceso a internet, aunque esto implica que no podrá guardar progreso

- **El público objetivo al que va dirigido la app:**

Se decidió que la app estaría dirigida para personas con hipoacusia leve y moderada, debido a la ineficacia del aprendizaje de la vocalización en personas con un nivel grave de la enfermedad [12]. Aunque la aplicación servirá para personas de cualquier edad, se priorizó un diseño y contenido enfocados a niños con este problema.

- **El conjunto de palabras que serán introducidas:**

De acuerdo a los maestros expertos en el tema, después de incluir el alfabeto, se priorizaron las palabras más comúnmente utilizadas en la vida cotidiana, además de las que pueden ser usadas en caso de emergencia.

- **Los maestros expertos en lenguaje de señas que ayudaran con la creación del contenido:**

El contenido a incluir en la aplicación como las palabras elegidas y los videos de lenguaje de señas deben ser realizados por el maestro que servirá como interprete y con un método de enseñanza, en este caso el método de enseñanza de comunicación total de Benson Schaeffer explicado en la sección(5.6.2) del marco teórico.

En los siguientes puntos de la fase de exploración se verán reflejado los puntos mencionados y como resultado los requisitos requeridos para la aplicación.

6.3.1.1. Perfiles del público de interés.

- **Desarrolladores:**

Su trabajo incluye una variedad de tareas, desde el diseño y desarrollo de software hasta las pruebas y el mantenimiento de software, con el objetivo de satisfacer las necesidades técnicas de los usuarios y las organizaciones.

- **Estudiantes con discapacidad auditiva:**

Lo podemos describir como los usuarios finales en la aplicación para el aprendizaje progresivo de la vocalización.

- **Directora del centro estudiantil:**

Directora del centro del colegio público de Masaya quien nos brindó los primeros requisitos iniciales de la aplicación para su desarrollo inicial.

- **Profesores del centro educativo especial:**

Profesores que desempeñan roles específicos adaptados a las necesidades de los estudiantes con discapacidades o necesidades educativas especiales. En el caso del trabajo monográfico, que tenga habilidades en el lenguaje de señas.

6.3.1.2. Definición de alcance

Se identifica el propósito y los objetivos de la aplicación móvil. Esto implica comprender las necesidades de los usuarios, los requisitos, resultados esperados y las limitaciones [27].

6.3.1.3. Requerimientos funcionales

Los requerimientos funcionales describen qué debe hacer el sistema. Son especificaciones de las funciones, características o tareas que el sistema debe realizar, en las interacciones entre el software y el usuario [45].

Se desarrollo una aplicación móvil con el IDE de Android Studio que permite brindar un apoyo a los profesores con sus alumnos hipoacúsicos. En base a toda la información recolectada por los entes del colegio Melania Lacayo de Masaya, se definieron los siguientes requisitos.

TABLA 1. REQUERIMIENTOS FUNCIONALES. [autoría propia]

ID	Requerimiento	Descripción	Prioridad
RF1	Inicio de sesión	El sistema debe permitir acceder a los usuarios a la aplicación por medio de una cuenta registrada.	Alta
RF2	Registro de Usuario	El usuario tendrá la facilidad de crear su propia cuenta para acceder a la aplicación y todas sus funcionalidades.	Alta
RF3	Listado de palabras	La aplicación debe mostrar un listado categorizado de las palabras elegidas por los expertos en educación especial.	Alta
RF4	Reconocimiento de voz	El sistema debe permitir capturar la voz del usuario mediante reconocimiento de voz.	Alta
RF5	Reproductor de media	La aplicación deberá de reproducir videos específicos con referencia al lenguaje de señas y la logopedia. También deberá grabar audio y reproducir el resultado.	Alta
RF6	Cálculo de resultados	En la aplicación se tendrá que mostrar indicadores del acierto que el usuario obtuvo una vez realice un ejercicio.	Alta
RF7	Evaluación de la Pronunciación	La aplicación debe evaluar la pronunciación del estudiante y presentar visualmente el resultado	Alta
RF8	Guardar Progreso	La aplicación debe tener la opción de guardar el progreso del estudiante.	Alta
RF9	Diccionario	La app debe tener una opción en la cual se muestre un diccionario con definiciones imágenes y videos de lenguaje de señas que referencien las palabras.	Media
RF10	Estadísticas	En la app se podrá consultar el progreso del estudiante por medio de las estadísticas.	Media

6.3.1.4. Requerimientos no funcionales

TABLA 2. REQUERIMIENTOS NO FUNCIONALES. [autoría propia]

ID	Requerimiento	Descripción	Prioridad
RNF1	Lenguaje de Programación	La aplicación está hecha en Java con el IDE de Android Studio	Alta
RNF2	WebServices	El lenguaje de programación del webservice fue desarrollado en PHP	Alta
RNF3	Usabilidad	La aplicación debe de ser intuitiva para el usuario y adaptable para dispositivos móviles	Media
RNF4	Manejo de acciones y errores	Se deben proporcionar mensajes de error claros y guías de ayuda contextual para facilitar la experiencia del usuario.	Alta
RNF5	Compatibilidad	El sistema debe de ejecutarse en sistemas operativos compatible con la plataforma de Android 9.0 Pie en adelante.	Alta
RNF6	Seguridad	La aplicación debe de contar con verificación e inicio de sesión y códigos de encriptamiento.	Alta
RNF7	Disponibilidad	La aplicación debe de estar disponible en todo momento, con todas las funciones, exceptuando la de guardado si el usuario no tiene acceso a internet y no está registrado.	Alta
RNF8	Rendimiento	La aplicación debe de tener tiempos de respuesta cortos	Alta

6.3.1.5. Limitaciones

Las limitaciones de la aplicación móvil son las siguientes:

- La aplicación móvil solo guardara información por medio de conexión a internet y verificando que sea un usuario registrado.
- La versión mínima del SDK, que puede ejecutarse en dispositivos móviles con el sistema operativo Android, es la API 28 (Android 9.0 Pie).

6.3.2. Iniciación

Se configura el proyecto identificando y preparando todos los recursos y requisitos iniciales del cliente. Durante esta fase, dedicaremos un día a la planificación y el resto del tiempo se empleará en preparar al equipo de trabajo. El objetivo es que los desarrolladores se enfoquen en establecer tanto un patrón de diseño, como la arquitectura del software [43], [46].

6.3.2.1. Configuración de proyecto

En este apartado, el objetivo del desarrollador es establecer un entorno propicio y cómodo para llevar a cabo el desarrollo de la aplicación móvil, lo cual incluye la instalación de herramientas y la configuración del versionado, entre otros aspectos. Durante esta fase, el desarrollador también tiene la capacidad de realizar pruebas de los requisitos sin necesidad de llevar a cabo una implementación real.

- **Tipo de proyecto:** Aplicación móvil.
- **Framework de red:** Retrofit, Volley.
- **Entorno de desarrollo:** Android Studio, Visual Studio Code.
- **Herramienta de control de versionado:** Git, Github.
- **Bibliotecas y modelos de terceros:** Vosk, AmplitudeGraphView.

6.3.2.2. Preparación del ambiente

En esta fase se realizan las configuraciones necesarias para establecer un entorno de trabajo coherente y adecuado para el equipo de desarrollo. Este proceso incluye la instalación y configuración de herramientas y software, la configuración de sistemas de control de versiones, la instalación de bases de datos, la configuración de servidores y servicios necesarios. En este trabajo monográfico el objetivo es asegurar que los desarrolladores trabajen en un entorno uniforme y eficiente, que facilite el desarrollo, las pruebas y el despliegue del software.

Instalación de las siguientes herramientas: Visual Studio Code, phpMyadmin, Git, Vosk, Xampp, Postman, MySQL, AmplitudeGraphView.

6.3.2.3. Planificación de fases por iteraciones

una iteración es un ciclo de trabajo dentro de un proceso de desarrollo ágil. Durante cada iteración, el equipo de desarrollo realiza un conjunto completo de actividades que incluye planificación, diseño, codificación, revisión, para producir una función del software. El objetivo de las iteraciones es permitir una retroalimentación continua y ajustes rápidos, mejorando gradualmente el producto final a través de ciclos repetidos de desarrollo.

Para llevar un registro adecuado de las iteraciones, Mobile-D las representa a través de las diferentes vistas y funcionalidades desarrolladas en la aplicación, adaptando la duración de las iteraciones según sea necesario.

TABLA 3. PLANIFICACIÓN, FASE DE EXPLORACIÓN. [autoría propia]

Fase de Exploración	
Iteración	Descripción
Iteración 00	<ul style="list-style-type: none">➤ Establecimiento del grupo de interesados y de los actores del trabajo monográfico.➤ Se define el alcance que tendrá el trabajo monográfico, los requerimientos iniciales y las limitaciones del mismo.

TABLA 4. PLANIFICACIÓN, FASE DE INICIACIÓN. [autoría propia]

Fase de Iniciación	
Iteración	Descripción
Iteración 00	<ul style="list-style-type: none"> ➤ Se analizan los requerimientos iniciales. ➤ Se establece la arquitectura y se configuran las herramientas del entorno de desarrollo. ➤ Se realiza el prototipado y el esquema de navegabilidad de la aplicación.

TABLA 5. PLANIFICACIÓN, FASE DE PRODUCCIÓN. [autoría propia]

Fase de Producción	
Iteración	Descripción
Iteración 01	<ul style="list-style-type: none"> ➤ Se implementa diseño y se crean las diferentes ventanas de la aplicación según el esquema de navegabilidad y el diseño del prototipado.
Iteración 02	<ul style="list-style-type: none"> ➤ Implementación de la funcionalidad de inicio de sesión. ➤ Mejora de la interfaz. ➤ Creación de botones de acceso para inicio de sesión de usuarios registrados e invitados. ➤ Actualización del estado de la funcionalidad en el storycard.
Iteración 03	<ul style="list-style-type: none"> ➤ Implementación de registro de usuarios. ➤ Mejora de la interfaz. ➤ Actualización del estado de la funcionalidad en el storycard.
Iteración 04	<ul style="list-style-type: none"> ➤ Implementación de listas de palabras para categorizar contenido por medio de “RecyclerView” y “TableRow”. ➤ Uso de adaptadores para recibir, ordenar y enviar información. ➤ Mejora de la interfaz y actualización del estado del storycard de la funcionalidad.
Iteración 05	<ul style="list-style-type: none"> ➤ Implementación de la funcionalidad de reconocimiento de voz y tratamiento del audio mediante la librería Vosk. ➤ Instalación y configuración de las librerías de reconocimiento de voz y de graficación de ondas sonoras.

Fase de Producción	
Iteración	Descripción
Iteración 06	<ul style="list-style-type: none"> ➤ Implementación de la vista de reproductor de media ➤ Permite grabar audio y reproducir lo grabado. ➤ reproductor de video con contenido en referencia al lenguaje de señas ➤ Mejora de interfaz y actualización del estado en el storycard.
Iteración 07	<ul style="list-style-type: none"> ➤ Implementación de medición de aciertos. ➤ Por medio de código de colores se muestran las partes correctas, incorrectas e innecesarias de la palabra a pronunciar.
Iteración 08	<ul style="list-style-type: none"> ➤ Implementación de funcionalidad de diccionario de señas. ➤ Incorporación del contenido de videos e imágenes para cada palabra. ➤ Mejora de la interfaz. Actualización de estado del storycard.
Iteración 09	<ul style="list-style-type: none"> ➤ Implementación de la vista de “Estadísticas”. ➤ Gráfico de barra y tabla dinámica con los puntajes del usuario.
Iteración 10	<ul style="list-style-type: none"> ➤ Implementación de la graficación de sonidos. ➤ Por medio de un espectro de sonido se debe visualizar la intensidad de la voz del estudiante.
Iteración 11	Implementación de la base de datos y el servicio web (Web Service).

TABLA 6. PLANIFICACIÓN, FASE DE ESTABILIZACIÓN. [autoría propia]

Fase de Estabilización	
Iteración	Descripción
Iteración 12	<ul style="list-style-type: none"> ➤ Refactorización de la navegabilidad y Establecimiento de las interfaces finales para las ventanas.
Iteración 13	<ul style="list-style-type: none"> ➤ Refactorización y validación de las funcionalidades de Inicio de sesión tanto de usuarios registrados, como de invitados. ➤ Revisión definitiva de la interfaz.
Iteración 14	<ul style="list-style-type: none"> ➤ Refactorización de la funcionalidad de registro de usuarios. ➤ Validación de los campos requeridos por parte del usuario. ➤ Reajuste de la selección de los campos del registro.

Fase de Estabilización	
Iteración	Descripción
Iteración 15	<ul style="list-style-type: none"> ➤ Refactorización de la funcionalidad de “Listado de palabras”. ➤ Revisión definitiva de la interfaces. ➤ Reordenamiento por prioridad de las categorías de las palabras.
Iteración 16	<ul style="list-style-type: none"> ➤ Refactorización de funcionalidades del reproductor de media. ➤ Revisión definitiva de la interfaz.
Iteración 17	<ul style="list-style-type: none"> ➤ Refactorizar la funcionalidad de medición de aciertos. ➤ Mejora del cálculo de puntaje.
Iteración 18	<ul style="list-style-type: none"> ➤ Refactorización de la funcionalidad de Diccionario de señas. ➤ Revisión de la interfaz y mejora de los botones de navegación.
Iteración 19	<ul style="list-style-type: none"> ➤ Refactorización, corrección de funcionalidades y validaciones de la vista, “Estadísticas”. ➤ Revisión definitiva de la interfaz.
Iteración 20	<ul style="list-style-type: none"> ➤ Refactorización y corrección de errores de la base de datos y el servicio Web.

TABLA 7. PLANIFICACIÓN, FASE DE PRUEBAS DEL SISTEMA. [autoría propia]

Fase de Pruebas del sistema	
Iteración	Descripción
Iteración 21	<ul style="list-style-type: none"> ➤ Se prueba y evalúa cada vista y sus funcionalidades. Se Presenta y analiza el estado final de cada vista según lo registrado mediante los StoryCard.

6.3.2.4. Patrón de diseño MVC (Modelo-Vista-Controlador)

MVC es un patrón de diseño arquitectónico que se utiliza para organizar el código en aplicaciones, separando las preocupaciones en tres componentes [47]:

- **Modelo:** Siendo el encargado de recibir datos y notificar por cualquier actualización de datos .
- **Vista:** Se encarga de la presentación y la interfaz de usuario.
- **Controlador:** Actúa como intermediario entre el Modelo y la Vista, gestionando la entrada del usuario y actualizando la Vista en consecuencia.

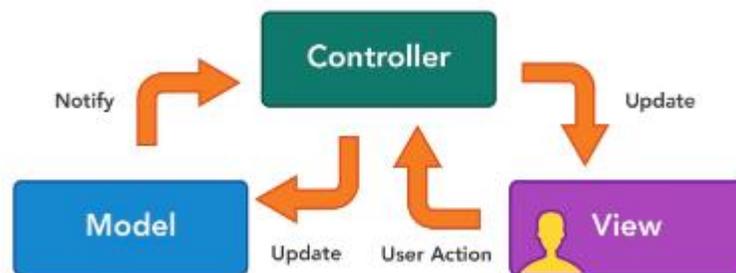


Fig. 8. Modelo Vista Controlador. [48]

En el presente trabajo monográfico, se utiliza este enfoque que destaca por la simplicidad en la conexión de los distintos componentes. La parte de la **vista** está formada por los diferentes layouts que proporcionan la interfaz gráfica al usuario. El **controlador**, que gestiona las actualizaciones en las actividades de los datos introducidos por el usuario y notificados por el modelo. Por último, el **modelo** que consiste en el backend escrito en Java, se encarga de la lógica de las funciones de la aplicación, así como de la comunicación mediante clientes HTTP con el servicio web que aloja la base de datos.

6.3.2.5. Arquitectura modelo CS (Cliente-Servidor)

Es una arquitectura de red que organiza la interacción entre dos entidades principales: el cliente y el servidor. El cliente se refiere a los diferentes dispositivos móviles que solicitarán el servicio de la base de datos alojada en un servidor, el cual proporcionará una respuesta con los datos correspondientes [30].



Fig. 9. Modelo cliente servidor. [30]

Debido a que la función de guardado y estadísticas requieren de una comunicación externa para enviar y recibir datos, se optó por utilizar este modelo de arquitectura para el presente trabajo monográfico.

6.3.2.6. Diseño

En esta etapa, se definen las interfaces principales del sistema, sus interacciones y relaciones. Se decide la estructura general del software, como los patrones de diseño, la distribución de los módulos y la lógica general de cómo funcionará el sistema [43].

Para la implementación de este apartado de la app, se realizó un primer diseño utilizando el IDE de Android Studio y los componentes de Material Design para cada vista, el cual fue adaptado con el tiempo según las necesidades del cliente y del desarrollador. Se tomaron en cuenta parámetros como la organización de la estructura de navegación, la información de alertas y el público objetivo al que va dirigido.

6.3.2.7. Material Design

Los componentes de Material Design son bloques de construcción interactivos que ayudan a crear una interfaz de usuario consistente y accesible. Estos incluyen distintas propiedades que afectan su interacción con el usuario y su comunicación. Ejemplos de estas propiedades son: la selección, la activación, el manejo de errores, el desplazamiento, la pulsación, el arrastre y los estados deshabilitados [49].

Esta biblioteca se ha utilizado en los siguientes apartados de la aplicación, según los principios de diseño de Material Design para implementar su estilo:

- En la **visualización** Hemos organizado las interfaces utilizando componentes como **CardView**, listas y formas (shapes). Por ejemplo, en la pantalla principal de la aplicación, utilizamos CardViews para presentar la información de manera clara y organizada, respetando la jerarquía visual a través de la propiedad de elevación. Esto permite que los usuarios distingan fácilmente entre diferentes secciones y niveles de importancia dentro de la interfaz.
- En **navegación** se usó la herramienta de pestañas de navegación (tabs) y fragments, para un movimiento más fluido del usuario por las distintas ventanas de la app. Esto con el objetivo de que el usuario no se pierda entre las distintas partes de la app pues se requiere que la navegación se fácil para los niños.
- En la **comunicación** se alerta al usuario de las distintas acciones que realiza mediante cuadros de dialogos y snackbars. Esto retroalimenta al usuario de forma inmediata advirtiéndole de los diferentes estados en el que se encuentran sus interacciones.

Al aplicar estos componentes y principios de Material Design, logramos crear una experiencia de usuario más intuitiva, consistente y visualmente atractiva, mejorando así la funcionalidad y la accesibilidad de nuestra aplicación.

6.3.3. Producción

El objetivo de la fase de producción es realizar la funcionalidad deseada del producto mediante el uso de un ciclo de desarrollo iterativo e incremental. El desarrollo basado en pruebas se utiliza para implementar la funcionalidad del sistema. Planifica los requerimientos y las iteraciones de tareas pendientes separando las iteraciones de prueba [43], [44].

6.3.3.1. Historias de usuario y StoryCard

Las historias de usuario son una técnica utilizada en el desarrollo de software ágil para capturar y describir requisitos del sistema desde la perspectiva del usuario final. Son descripciones breves y sencillas que explican qué necesita hacer un usuario con el sistema y por qué [50]. También son usados como base para hacer los casos de prueba.

La tarjeta de historia (StoryCard) sirve como una unidad de trabajo clara y comprensible que facilita la colaboración y la comunicación dentro del equipo de desarrollo sobre el estado del desarrollo de las funcionalidades. Contiene información esencial sobre los requisitos, el propósito y los criterios de aceptación. Suele usarse durante la planificación y ejecución del trabajo en ciclos iterativos, como en el marco Scrum o en el presente trabajo con la metodología Mobile-D [51].

Esta herramienta se aplicó en el trabajo monográfico para mantener un registro detallado de las vistas de la aplicación y de las funciones implementadas. Estas vistas y funciones reflejan las diversas iteraciones y etapas de desarrollo por las que ha pasado la aplicación.

TABLA 8. DISEÑO DE UNA STORYCARD. [autoría propia]

Título					
Numero	Dificultad		Esfuerzo		Prioridad
00	Antes	Después	Estimado	Gastado	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Fácil Moderado Difícil	Fácil Moderado Difícil	En días	En días	
Descripción					
Criterios de aceptación					
Estado			Comentario		
<ul style="list-style-type: none"> ▪ Planificado 					
<ul style="list-style-type: none"> ▪ Implementado 					
<ul style="list-style-type: none"> ▪ Completado 					

- **Título** es el nombre designado del requerimiento al que hace referencia la storycard
- En **Numero** se asigna el identificador de la historia de usuario
- El campo **Dificultad** representa el nivel de complejidad para implementar las funcionalidades, basado en la experiencia del equipo de desarrollo. Se asignan valores antes y después de haber implementado la funcionalidad, siendo recomendado medirlo con los términos Fácil, Moderado y Difícil.
- En el campo de **Esfuerzo** se registra el tiempo necesario para implementar la funcionalidad, medido en días. Se realiza una estimación inicial del tiempo en el campo Estimado, y al final se presenta el tiempo real que se tardó en completarse en el campo Gastado.

- En **Prioridad** se valora que tan importante es la funcionalidad implementada, contiene los valores de Baja, Media, Alta.
- La **Descripción** indica la manera en que deberían comportarse las funcionalidades de ciertas vistas o requisitos descritos en la StoryCard. Como buena práctica, se recomienda que la descripción sea lo más general posible, evitando detalles técnicos.
- **El criterio de aceptación** es el conjunto de condiciones se considere completa y lista para ser entregada.
- El **Estado** indica en qué medida las funcionalidades descritas en el story card han sido implementadas. Los valores son los siguientes: **Planificado**, si la función está planeada para el futuro; **Implementado**, si la funcionalidad ha sido codificada incluso a un nivel básico; y **Completado**, si ha habido una revisión y pulimiento de las funcionalidades.
- En el campo **Comentarios** se menciona cualquier limitación, falla o información relevante según el estado en el que se encuentre.

6.3.3.2. Inicio de sesión

El inicio de sesión es el proceso por el cual tanto los estudiantes como los docentes accederán a la aplicación. Solo existe un tipo de usuario que incluyen los estudiantes y docentes. Se tiene como objetivo que los estudiantes estén registrados para poder guardar su progreso por medio de internet y tener retroalimentación del maestro que lo esté apoyando. También se tiene la opción de acceder de modo offline como un usuario invitado, pero este no tiene posibilidad de guardar su progreso.



Fig. 10. Inicio de sesión. [autoría propia]

TABLA 9. STORYCARD INICIO DE SESIÓN. [autoría propia]

Inicio de sesión					
Numero	Dificultad		Esfuerzo		Prioridad
	Antes	Después	Estimado	Gastado	
01	Fácil	Fácil	5d	6d	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Moderado Duro	Moderado Duro			
Descripción					
<p>Como Estudiante registrado, quiero poder iniciar sesión en la aplicación para acceder a mi cuenta y ver mi progreso personal.</p>					

Criterios de aceptación	
<ul style="list-style-type: none"> • La pantalla de inicio de sesión debe contener campos para ingresar el correo electrónico y la contraseña. • Debe estar disponible la opción de entrar como usuario registrado, o usuario invitado. • Si las credenciales son incorrectas, debe mostrarse un mensaje de error indicando "Credenciales incompletas o incorrectas". 	
Estado	Comentario
▪ Planificado	Sin Comentario
▪ Implementado	Sin Comentario
▪ Completado	Sin Comentario

6.3.3.2.1. Implementaciones en Inicio de sesión

Autenticación de usuario mediante Retrofit.

Se verifica la identidad del estudiante o docente mediante su correo y contraseña, y se capturan los datos para guardarlos una vez que sus credenciales son válidas. Este proceso utiliza la librería Retrofit para validar que el correo sea el correcto y que la contraseña cumpla con los requisitos. Una vez verificados, se realiza la petición al servidor, la cual devolverá una respuesta indicando si la operación fue exitosa o si ocurrió algún error.

Inició sesión como invitado.

En caso de que el usuario, ya sea estudiante o docente, no quiera crear una cuenta o no tenga acceso a internet y solo desee explorar la aplicación, existirá la opción de iniciar sesión como invitado. Este modo brindará al usuario ciertas limitaciones al interactuar con la aplicación como no poder guardar el progreso. Esto es posible gracias a una función que adjunta datos adicionales. En este objeto, pasamos una identificación y un valor que indican que el usuario es un invitado.

6.3.3.3. Registro de usuario

Con el formulario de ingreso, los estudiantes y maestros tendrán la posibilidad de crear un perfil de usuario, cuyo objetivo es guardar el progreso en la nube y controlar el tiempo de uso del estudiante. Para crear el perfil, se requiere incluir un correo electrónico, que puede ser del estudiante o de su tutor. Sin embargo, se prefiere que sea el correo del tutor si el estudiante es de primaria.

Fig. 11. Formulario de ingreso.

TABLA 10. STORYCARD REGISTRO DE USUARIO. [autoría propia]

Registro de usuario					
Numero	Dificultad		Esfuerzo		Prioridad
02	Antes	Después	Estimado	Gastado	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Fácil Moderado Duro	Fácil Moderado Duro	10d	8d	
Descripción					
<p>Como nuevo usuario o estudiante, quiero poder registrarme en la aplicación, para crear una cuenta y acceder a todas las funciones de la aplicación.</p>					

Criterios de aceptación	
<ul style="list-style-type: none"> • La pantalla de registro debe contener los siguientes campos obligatorios: nombre completo, correo electrónico contraseña, confirmación de contraseña. • Al hacer clic en "Registrarse", si los datos son correctos, se debe crear una nueva cuenta en el sistema. • Si el correo electrónico ya está registrado, debe mostrarse un mensaje de error indicando que el correo ya está en uso. 	
Estado	Comentario
▪ Planificado	Sin Comentario
▪ Implementado	Sin Comentario
▪ Completado	

6.3.3.3.1. Implementaciones del registro de usuario

Encriptación de contraseña

Cuando el usuario envíe el formulario de ingreso en la aplicación, su contraseña se almacenará utilizando el algoritmo de encriptación RSA (Rivest-Shamir-Adleman) y se recuperará utilizando el mismo método. RSA, un algoritmo de criptografía asimétrica, emplea un par de claves: una pública y otra privada. Durante el envío de datos del formulario de registro, se cifran mediante la clave pública y se descifran mediante la clave privada correspondiente. La clave privada se almacena en el servidor de forma encriptada.

6.3.3.4. Lista de palabras

Esta listas de categorías contiene distintas palabras de una misma temática que se presentan como método de organización del contenido disponible, en el que seleccionando alguno de los ítems se puede acceder a los ejercicios o al diccionario con su respectivo significado. Cada categoría y palabra fue consultado con un intérprete y maestro en lenguaje de señas que dieron su aprobación a dicho contenido.



Fig. 12. Lista de palabras. [autoría propia]

TABLA 11. STORYCARD LISTA DE PALABRAS. [autoría propia]

Lista de palabras					
Numero	Dificultad		Esfuerzo		Prioridad
	Antes	Después	Estimado	Gastado	
03	Fácil Moderado Duro	Fácil Moderado Duro	12d	15d	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
Descripción					
<p>Como usuario,</p> <p>quiero poder ver un listado de palabras categorizadas en la aplicación,</p> <p>para poder elegir y aprender según la categoría que me interese.</p>					
Criterios de aceptación					
<ul style="list-style-type: none"> • Al seleccionar una categoría, el usuario debe ver un listado de palabras correspondientes a esa categoría. • Las categorías y palabras seleccionadas como contenido deben ser aprobadas por los maestros en educación especial. 					
Estado			Comentario		
<ul style="list-style-type: none"> ▪ Planificado 			Sin Comentario		
<ul style="list-style-type: none"> ▪ Implementado 			Sin Comentario		
<ul style="list-style-type: none"> ▪ Completado 			Sin Comentario		

6.3.3.4.1. Implementaciones en lista de palabras

Listado de contenido por medio de RecyclerView

El RecyclerView es una herramienta que facilita la creación de listados de elementos y la reutilización de un diseño común; se implementa en el presente trabajo debido a la masiva cantidad de ítems a mostrarse. Después de que el usuario selecciona una categoría, la aplicación muestra palabras para elegir mediante la función de listado (RecyclerView) y el adaptador (Adapter).

En la configuración de esta función, se establece una lista de arreglos para el modelo de objeto, la vista de listado y la lista de adaptadores. Se valida que la actividad de categorías haya enviado los elementos en formato JSON y que no estén vacíos. Una vez confirmada la información, se asigna la vista de listado en el gestor de diseños, se instancia la lista de palabras con la información proporcionada y se asigna al adaptador correspondiente. El adaptador, que interactúa con los valores y la interfaz gráfica, se crea como una clase que se extiende a la vista de listado (RecyclerView). Se establece el constructor del adaptador y se implementan los elementos necesarios, como onCreateViewHolder (para crear y devolver una nueva instancia de la vista), onBindViewHolder (para vincular los datos a la vista en la posición dada) e getItemCount (para devolver el número total de elementos en los datos).

Navegación entre fragments

Un fragment es un componente modular de una interfaz de usuario que representa una porción o una sub-sección de una actividad. Los fragments permiten dividir una actividad en partes modulares, cada una con su propio ciclo de vida, interfaz de usuario y lógica. Esto facilita la reutilización de componentes en diferentes actividades.

En la parte inferior, se encuentra un menú que facilita la navegación entre fragments siendo estos los menús que se encuentran después del inicio de sesión. Para implementar la navegabilidad, se crea un recurso de diseño (layout) para el menú de navegación con ítems. Posteriormente, en el directorio de recursos (res),

se establece un directorio denominado nav_graph, donde se agregan y unen los fragmentos. Una vez que se ha completado, se accede a la actividad principal la cual contiene todos los fragments.

6.3.3.5. Reconocimiento de voz

El reconocimiento de voz constituye el método mediante el cual se obtiene información sobre la pronunciación del audio grabado por el por el estudiante, siendo esta una tarea realizada por el modelo de la herramienta Vosk. La API de Vosk dispone de diversos modelos acústicos y de lenguaje previamente entrenados, diseñados para convertir voz en texto, lo que la convierte en una herramienta versátil para aplicaciones de reconocimiento de voz en tiempo real.

Considerando el hardware específico para el cual se requiere implementar la herramienta, se optó por utilizar la versión "**Vosk-model-small-es**", la cual está entrenada específicamente para su empleo en dispositivos portátiles y en el idioma español. Este modelo asume la responsabilidad de extraer y convertir el audio a texto para su tratamiento posterior. Para lograrlo, Vosk emplea el proceso de reconocimiento automático del habla (ASR, por sus siglas en inglés).

El audio de entrada se segmenta en fragmentos cortos y se procesa de manera secuencial, asignando probabilidades a diferentes palabras en cada fragmento. Posteriormente, se realiza una búsqueda para encontrar la secuencia de palabras más probable.

TABLA 12. STORYCARD RECONOCIMIENTO DE VOZ. [autoría propia]

Reconocimiento de voz					
Numero	Dificultad		Esfuerzo		Prioridad
	Antes	Después	Estimado	Gastado	
05	Fácil Moderado Duro	Fácil Moderado Duro	25d	30d	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta

Descripción	
<p>Como estudiante, quiero utilizar el reconocimiento de voz dentro de la aplicación, para practicar la pronunciación de palabras, de manera que pueda recibir retroalimentación sobre mi desempeño.</p>	
Criterios de aceptación	
<ul style="list-style-type: none"> • El usuario debe poder activar el reconocimiento de voz desde una vista específica, que le permita iniciar la grabación de su pronunciación. • La aplicación debe permitir al usuario repetir el ejercicio tantas veces como lo desee. • El reconocimiento debe activarse solo durante una cantidad determinada de tiempo por intento. En este caso 20 segundos. 	
Estado	Comentario
<ul style="list-style-type: none"> ▪ Planificado 	
<ul style="list-style-type: none"> ▪ Implementado 	
<ul style="list-style-type: none"> ▪ Completado 	

6.3.3.6. Reproductor de media

El Reproductor de media funciona como la interfaz para realizar las actividades prácticas de vocalización. Esta aglomera un conjunto de funciones relacionadas con el reconocimiento de voz, reproductor de video, audio y puntaje del ejercicio.

- Cada video presentado funciona como guía y ejemplo del lenguaje de señas y la pronunciación de dicha palabra. Estos videos requirieron de un maestro experto en lenguaje de señas y un intérprete para crear este contenido.
- Una gráfica del espectro de sonido que se activa al grabar la voz del estudiante. Esta hecho con el propósito de tener otra forma de guía visual para estudiantes con discapacidad auditiva.
- Una grabadora de voz, que es el encargado de captar la pronunciación de la palabra por un periodo de 20 segundos para que el estudiante no abuse del tiempo de grabación.

- Por último, un evaluador de pronunciación por medio de reconocimiento de voz el cual se utiliza para comparar lo pronunciado por el estudiante con el resultado que se espera del que se obtendrá un puntaje.



Fig. 13. Reproductor de media. [autoría propia]

TABLA 13. STORYCARD REPRODUCTOR DE MEDIA. [autoría propia]

Reproductor de media					
Numero	Dificultad		Esfuerzo		Prioridad
04	Antes	Después	Estimado	Gastado	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Fácil Moderado Duro	Fácil Moderado Duro	20d	27d	
Descripción					
<p>Como estudiante, quiero poder reproducir videos, grabar mi voz y reproducirla, para visualizar contenido educativo relacionado con la pronunciación y lenguaje de señas. También para practicar y mejorar mi pronunciación y habilidades de comunicación.</p>					

Criterios de aceptación	
<p>Funcionalidad de Reproductor de video.</p> <ul style="list-style-type: none"> • Se debe mostrar un video en lenguaje de señas y con vocalización siendo referente al ejercicio en cuestión. <p>Funcionalidad de Grabación.</p> <ul style="list-style-type: none"> • El estudiante puede iniciar la grabación de su voz durante 20 segundos presionando el botón "Grabar". • Durante la grabación, el botón "Grabar" cambia a "Detener" para finalizar la grabación. • La grabación de la voz debe estar vinculada al reconocimiento de voz. <p>Funcionalidad de Reproducción</p> <ul style="list-style-type: none"> • El Estudiante puede reproducir el archivo de audio grabado presionando un botón "Reproducir". • Durante la reproducción, el botón "reproducir" cambia a "Detener" para finalizar la reproducción 	
Estado	Comentario
▪ Planificado	Usuarios invitados no enviaran datos
▪ Implementado	Validación si es usuario invitado
▪ Completado	Se presenta un retraso de 1 segundo entre el presionar el botón de grabación y el inicio del reconocimiento de voz.

6.3.3.6.1. Implementaciones del reproductor de media

Reproductor de video

A través del reproductor, los usuarios tienen la capacidad de reproducir el video que hayan seleccionado, ya sea una palabra, letra o número previamente elegido en las vistas de señas o en la de práctica (pantalla principal). Dado que es necesario acceder a los videos en cualquier momento, estos se almacenan localmente. Se ha establecido una ruta de archivo de video mediante una variable que contiene la ubicación del archivo correspondiente. Dada la naturaleza breve

de los videos, que constan de unos pocos segundos de reproducción, se ha implementado un único botón de reproducción y se ha desactivado la función de reproducción en bucle.

Grabadora y reproductor de audio

La implementación de la grabadora de audio tiene dos objetivos: obtener una muestra de audio de la palabra requerida para que el profesor o logopeda pueda corregir la pronunciación, y la segunda es obtener una cadena de texto para mostrar como resultado del audio captado mediante el reconocimiento de voz. Al establecer un contador cada vez que se presiona el botón de grabación, la herramienta tiene la capacidad de grabar durante un período de 20 segundos, que cuenta como un intento de realizar el ejercicio por parte del estudiante. Esto ayuda a evitar el abuso en el uso y también previene problemas al reproducir una extensión de audio considerable que el usuario pudo haber grabado. Debido a la corta duración de la grabación, solo hay tres estados: grabando audio, reproduciendo audio o detenido.

Barra indicadora de porcentaje de acierto

La barra se llenará con el último intento del usuario en un ejercicio de pronunciación, mostrando el porcentaje de lo pronunciado correctamente en una evaluación. Se llenará en función del número de letras pronunciadas correctamente. Por ejemplo, si el usuario pronuncia correctamente el 70% de una palabra, la barra alcanzará el 70% de su capacidad, y un valor numérico o textual acompañará visualmente la barra.

6.3.3.7. Cálculo de resultados y códigos de colores

Esta función proporciona una representación visual rápida y comprensible del rendimiento del usuario en el ejercicio, ayudando a contextualizar el puntaje en términos de calidad de ejecución. Se calcula contando los caracteres acertados y desacertados del total de la palabra, para luego mostrar la media en porcentaje, del acierto en el pronunciación. Un código de color se añade de acuerdo al porcentaje de acierto: verde para “excelente” de un 90% en acierto o más, verde

tierno para “muy bueno” de 80%, color amarillo para “bueno” un 70% de aciertos, naranja para regular un 60%, y rojo si es deficiente con un acierto de 50% o menor.

TABLA 14. STORYCARD CÁLCULO DE RESULTADOS. [autoría propia]

Cálculo de resultados					
Numero	Dificultad		Esfuerzo		Prioridad
06	Antes	Después	Estimado	Gastado	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Fácil Moderado Duro	Fácil Moderado Duro	7d	7d	
Descripción					
<p>Como usuario, quiero que la aplicación calcule mis resultados y verlo reflejado visualmente al completar un ejercicio, para que pueda conocer mi desempeño y mejorar.</p>					
Criterios de aceptación					
<ul style="list-style-type: none"> • En tiempo real, la aplicación debe calcular automáticamente el puntaje del estudiante en función de su precisión en la pronunciación. • El cálculo debe reflejar el porcentaje de aciertos en base al total de ítems de la palabra. • La aplicación debe mostrar los resultados con el porcentaje y una evaluación general de forma visual. • Los resultados deben presentarse de forma clara y visualmente atractiva, utilizando gráficos de barras o indicadores de color. 					
Estado			Comentario		
▪ Planificado					
▪ Implementado					
▪ Completado					

6.3.3.8. Evaluación de la pronunciación

En la vista de Reproductor de media se ha implementado un proceso de comparación visual y cuantitativa entre dos cadenas de texto, la pronunciación extraída por medio de la api de reconocimiento de voz y la pronunciación con el resultado esperado (la cadena de referencia).

Este proceso se lleva a cabo mediante la utilización de un bucle que itera a través de los caracteres de ambas cadenas, identificando coincidencias y desviaciones. Durante esta comparación, se contabilizan tanto las coincidencias como las discrepancias, proporcionando así una evaluación cuantitativa del rendimiento.

TABLA 15.STORYCARD DE EVALUACIÓN DE LA PRONUNCIACIÓN.

Evaluación de la pronunciación					
Numero	Dificultad		Esfuerzo		Prioridad
07	Antes	Después	Estimado	Gastado	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Fácil Moderado Duro	Fácil Moderado Duro	20d	21d	
Descripción					
<p>Como estudiante con discapacidad auditiva, Quiero que mi pronunciación sea evaluada visualmente, para poder identificar las partes de la palabra que pronuncie correctamente, incorrectamente o de forma innecesaria.</p>					
Criterios de aceptación					
<ul style="list-style-type: none"> • Durante la grabación el modelo de reconocimiento de voz analiza la pronunciación del usuario • La aplicación debe resaltar las partes de la palabra de la siguiente manera: <ol style="list-style-type: none"> 1. En verde: Partes pronunciadas correctamente. 2. En rojo: Partes pronunciadas incorrectamente. 3. En amarillo: Partes innecesarias (sonidos adicionales que no deberían estar). 					

Estado	Comentario
▪ Planificado	
▪ Implementado	
▪ Completado	

La visualización de los resultados se realiza en tiempo real mediante la manipulación del color de fondo de un TextView, destacando de manera gráfica los caracteres acertados, erróneos y aquellos que no están presentes en la cadena de referencia.

Este enfoque busca facilitar una retroalimentación inmediata y efectiva al usuario, contribuyendo así a la mejora de la pronunciación y comprensión auditiva. Además, se incorpora una lógica especial para el manejo de espacios en blanco, asegurando una representación precisa del análisis comparativo.

Hay tres códigos de colores que representan los estados en cuanto a la presentación de resultados en el acierto de los caracteres:

- Carácter resaltado en **verde**. Representa que la pronunciación del usuario fue acertada en ese determinado carácter.
- Carácter resaltado en **rojo**. Representa que la pronunciación de ese carácter fue errada.
- Carácter resaltado en **amarillo**. Estas partes son caracteres innecesarios que han sido pronunciados por el usuario

6.3.3.9. Guardado de progreso

Esta funcionalidad permite que los resultados, y progresos realizados por el Estudiante previamente registrado sean almacenados en la nube, brindando continuidad en el uso de la aplicación.

Esto es realizado por medio de la función de envío y recibimiento de datos gracias a los clientes HTTP en conexión con el servicio web y la base de datos

TABLA 16. STORYCARD GUARDADO DE PROGRESO. [autoría propia]

Guardado de progreso					
Numero	Dificultad		Esfuerzo		Prioridad
08	Antes	Después	Estimado	Gastado	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Fácil Moderado Duro	Fácil Moderado Duro	11d	13d	
Descripción					
<p>Como estudiante registrado, quiero poder guardar mi progreso, para para poder continuar mi aprendizaje desde donde lo dejé en cualquier momento.</p>					
Criterios de aceptación					
<ul style="list-style-type: none"> • El estudiante puede guardar su progreso al presionar el botón de guardar si tiene conexión a internet. • Debe aparecer un mensaje para confirmar si el estudiante quiere reemplazar los datos si hay información previamente guardada. 					
Estado			Comentario		
<ul style="list-style-type: none"> ▪ Planificado 					
<ul style="list-style-type: none"> ▪ Implementado 					
<ul style="list-style-type: none"> ▪ Completado 					

6.3.3.10. Diccionario

El diccionario en la aplicación móvil está diseñado para proporcionar definiciones claras y concisas de las palabras, junto con ejemplos y recursos visuales adicionales. Cada entrada del diccionario incluye la palabra, su definición, un video en lenguaje de señas y una imagen relacionada (si corresponde), para mejorar la comprensión. Entre los componentes de esta pantalla se encuentra:

- El video correspondiente a la palabra seleccionada que presenta su respectiva seña.

- La definición de la palabra, necesaria para que el estudiante relacione el lenguaje de señas y la pronunciación con el contexto del uso de la palabra. Esto incluye un ejemplo de su uso.
- Una imagen que ilustra el significado y proporciona información visual para ayudar a los estudiantes con hipoacusia a relacionar la imagen con la palabra.



Fig. 14. Pantalla de lectura del diccionario. [autoría propia]

TABLA 17. STORYCARD DICCIONARIO. [autoría propia]

Diccionario					
Numero	Dificultad		Esfuerzo		Prioridad
	Antes	Después	Estimado	Gastado	
09	Fácil	Fácil	12d	10d	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Moderado	Moderado			
	Duro	Duro			
Descripción					
<p>Como Estudiante,</p> <p>quiero acceder a un diccionario de palabras,</p> <p>para mejorar mi comprensión y aprendizaje sobre las palabras de los ejercicios.</p>					

Criterios de aceptación	
<ul style="list-style-type: none"> Al seleccionar una palabra, el usuario puede visualizar su definición, una imagen relacionada y un video con audio de la palabra en lenguaje de señas. El diccionario se puede acceder tanto desde la lista categorizada de palabras como desde cualquier ejercicio en curso 	
Estado	Comentario
<ul style="list-style-type: none"> Planificado 	
<ul style="list-style-type: none"> Implementado 	
<ul style="list-style-type: none"> Completado 	

6.3.3.11. Estadísticas

La pantalla de estadísticas de la aplicación ofrece a los usuarios información de la cantidad de intentos en cada ejercicio y su puntaje. Esta pantalla permite a los estudiantes y docentes monitorear el progreso y el desempeño a lo largo del tiempo.

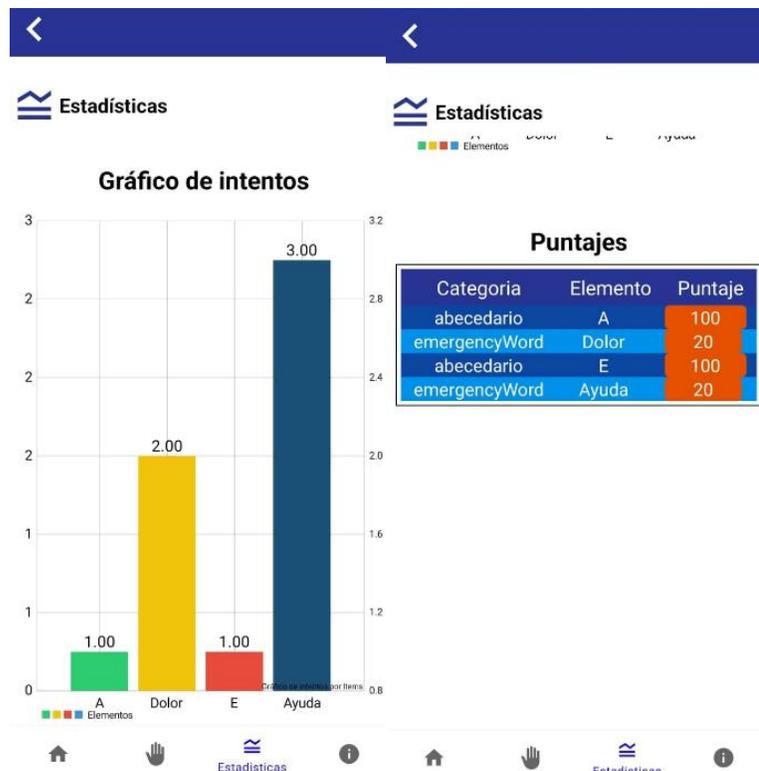


Fig. 15. Pantalla de estadísticas.

TABLA 18. STORYCARD DE ESTADÍSTICAS. [autoría propia]

Estadísticas					
Numero	Dificultad		Esfuerzo		Prioridad
10	Antes	Después	Estimado	Gastado	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Fácil Moderado Duro	Fácil Moderado Duro	17d	20d	
Descripción					
<p>Como estudiante registrado, quiero ver mi progreso medido, para entender mejor mis avances y áreas de mejora.</p>					
Criterios de aceptación					
<ul style="list-style-type: none"> • Debe ser accesible desde la pantalla principal • Debe mostrar un listado de todos los ejercicios completados • Debe mostrar puntaje como forma de progreso 					
Estado			Comentario		
<ul style="list-style-type: none"> ▪ Planificado 			Sin Comentarios		
<ul style="list-style-type: none"> ▪ Implementado 			Sin Comentarios		
<ul style="list-style-type: none"> ▪ Completado 			Sin Comentarios		

6.3.3.11.1. Implementaciones de estadísticas

Gráfico de barras.

Para una medición del progreso y la consistencia se implementó el grafico de barras en el cual se observa el número de usos en la aplicación. Las operaciones se ejecutan a través del backend alojado en el servidor. Se obtiene un conjunto de elementos según el identificador (ID) del usuario que está autenticado en la aplicación. Luego, se crean las clases correspondientes y las funciones de etiquetado de los ejes X y Y para personalizar la presentación del gráfico.

Tabla de puntajes

La puntuación es obtenida del Reproductor de media cuando el usuario guarda los resultados de cualquier ejercicio. Para crear dinámicamente la tabla de puntajes, no es necesario utilizar bibliotecas adicionales. Basta con realizar una solicitud al servidor para obtener la información de los puntajes obtenidos y guardados por el usuario logueado. Posteriormente, se puede utilizar la clase `TableLayout` para construir dinámicamente el elemento gráfico de la tabla utilizando los datos obtenidos del servidor.

6.3.3.12. Graficación del sonido

La graficación del sonido es una herramienta esencial para visualizar de manera clara y comprensible los elementos sonoros de la voz. Su propósito es proporcionar una guía visual que permita al estudiante con discapacidad auditiva controlar su voz de manera más efectiva.

TABLA 19. STORYCARD DE GRAFICACIÓN DEL SONIDO. [autoría propia]

Graficación del sonido					
Numero	Dificultad		Esfuerzo		Prioridad
11	Antes	Después	Estimado	Gastado	<ul style="list-style-type: none"> ▪ Baja ▪ Media ▪ Alta
	Fácil Moderado Duro	Fácil Moderado Duro	16d	20d	
Descripción					
<p>Como estudiante con discapacidad auditiva, quiero visualizar el sonido de mi voz mientras realizo los ejercicios de pronunciación, para poder controlar mejor mi voz</p>					

Criterios de aceptación	
<ul style="list-style-type: none"> • El sistema debe generar una gráfica en tiempo real de los sonidos emitidos por el usuario. • La gráfica debe ser fácil de interpretar, mostrando variaciones del volumen por medio de colores. Siendo verde la intensidad correcta y rojo la intensidad incorrecta • El usuario debe recibir retroalimentación visual inmediata al emitir un sonido. 	
Estado	Comentario
▪ Planificado	Sin Comentarios
▪ Implementado	Sin Comentarios
▪ Completado	Sin Comentarios

6.3.3.12.1. Implementaciones de graficación del sonido

Grafica de intensidad con respecto al tiempo

Mediante la herramienta de **graphviewlibrary** descrita en la sección del marco teórico se representará de forma visual la intensidad del sonido que el usuario este introduciendo mediante su voz al dispositivo móvil. Esto con el objetivo guiar al usuario para que controle la intensidad de su voz.

En este caso, **graphviewlibrary** provee de una gráfica del espectro del sonido personalizable por lo que se configuro con una escala de 0db como máximo según la escala relativa mencionada en la sección y con un periodo de actualización de 100 milisegundos entre cada onda senoidal dibujada. También se agregaron códigos de colores siendo un color cercano a verde si la intensidad de la voz es la correcta, y un color cercano a rojo si la intensidad es baja o muy alta.

6.3.3.13. Base de datos.

Para el desarrollo de la aplicación era necesario guardar registros del avance del usuario. Para ello, se utilizó phpMyAdmin junto con el gestor de base de datos MySQL. MySQL utiliza el lenguaje SQL, que es muy sencillo y cuenta con mucha documentación disponible [36]. Para implementar la base de datos, se debe activar la opción de MySQL y Apache en el panel de control de XAMPP. Una vez activos, en el navegador se escribe la siguiente dirección: <http://localhost/phpmyadmin/>.

PhpMyAdmin da la opción al usuario de crear una base de datos utilizando el lenguaje SQL o bien utilizando los componentes de la interfaz gráfica; de la misma forma se utiliza para crear las tablas de la base de datos. Una vez creada la base de datos, se crea una clase en el web service que recibe el nombre del servidor (localhost en este caso), el nombre de la base de datos, el usuario y la contraseña (estos últimos campos pueden dejarse vacíos si se prefiere, ya que se ejecuta en un entorno local para empezar).

Es importante destacar que, durante el desarrollo, la aplicación utiliza una base de datos local para facilitar las pruebas a través de un servicio web local. Sin embargo, en el despliegue final de la aplicación, la base de datos será alojada en la nube. Se tomó la decisión de realizar el almacenamiento en la nube debido a la necesidad de que el usuario tenga acceso a sus datos de progreso en cualquier momento, siempre que disponga de conexión a internet. De esta manera, se aprovecha al máximo la portabilidad del dispositivo móvil.

6.3.3.14. Adición y tratamiento de contenido

En esta parte del trabajo monográfico se requería añadir dos tipos de contenidos videos e imágenes con el fin de guiar al usuario en el apartado de señas, audio, y la representación visual de palabras que tengan un significado como expresiones u objetos. A continuación, se describirán los pasos que se necesitaron seguir para añadir el contenido:

Obtención de imágenes: Las imágenes utilizadas son recursos sacados de internet, que presentan una conexión entre el significado de la palabra y su representación visual.

Grabación de video: Para la grabación de los videos el único requisito es tener un dispositivo que pueda grabar en una resolución Full HD que equivale a 1920X1080. Esto es debido a que se requiere que el video tenga la suficiente calidad como para que las señas realizadas sean claras para seguirlas como guía.

Se requirió de un espacio bien iluminado y un fondo con buen contraste para que las señas del interprete puedan observarse de manera correcta. En total se grabaron 140 videos, los cuales en promedio tienen una duración de entre 2 a 5 segundos cada uno. Esto fue grabado en un periodo de 4 horas con una lista ya preparada de las palabras a ser grabadas.

Tratamiento de video: el tratamiento se realizó por medio de un editor de video, para este trabajo monográfico se utilizó la herramienta DaVinci Resolve. En el Tratamiento se incluye corrección de color de la imagen, ajuste de la resolución de video y tratamiento del ruido de imagen y audio. Como resultado se obtiene una imagen más nítida del video y con la resolución correcta.

La edición de video requirió que el contenido se organizara por categoría para luego ser editado por la herramienta. Cada categoría necesitaba de un tiempo de edición de entre media a 1 hora de acuerdo a la cantidad de contenido de la categoría. Luego cada categoría de videos necesitaba renderizarse lo que sumaba de entre 30 minutos a 2 horas en el que el programa terminaba de exportar los videos.

Introducir contenido: Todo el contenido es introducido en la carpeta RAW donde se guarda todo el material audiovisual de manera local. Esta decisión fue tomada de acuerdo a uno de los requisitos planteado por las partes interesadas del trabajo monográfico, con el objetivo de tener el contenido siempre disponible.

Después de introducir los videos se necesitan cargar dinámicamente los recursos, cada video tiene su respectiva información relacionada con las pronunciaciones, nombres de las señas, descripciones, recursos de audio y recursos gráficos.

A continuación, se dará una descripción más detallada de la información introducida para cada palabra y letra:

Nombre de la palabra: Funciona como el identificador de la palabra o letra, siendo el nombre escrito de ella. Ej: (Z: nombre de la letra), (Zeta: nombre moderno de la letra en español).

Pronunciación: Es la forma en que se emiten los sonidos del habla al articular palabras o frases en un idioma determinado; y en el caso de la aplicación es el resultado esperado de cómo se pronuncia una palabra. En la app la mayoría de las veces la pronunciación se representa de la misma forma que la palabra escrita a consideración del usuario. No obstante, según el modelo de reconocimiento las palabras y letras son tratadas según su notación fonética AFI (Alfabeto Fonético Internacional). Ej: (Q: nombre de la letra), (Ku: nombre en el alfabeto fonético).

Significado: la palabra y letra tiene una descripción de lo que representa como concepto básico, requerido para la función de diccionario.

Video: la ruta de ubicación de los videos de señas para cada palabra o letra es guardada en este espacio

Imagen: la ruta de ubicación de las imágenes con alusión a cada palabra o letra es guardada en este espacio

6.3.4. Estabilización

Se realizan las operaciones finales de integración para garantizar que todo el sistema funcione correctamente. El objetivo de esta fase es asegurar la calidad de la ejecución del trabajo monográfico.

6.3.4.1. Refactorización

La refactorización es una práctica que implica la reorganización y mejora del código existente sin afectar su comportamiento externo. Su objetivo es aumentar la claridad, eficiencia y mantenibilidad del código, entre otros aspectos, sin introducir nuevos errores. Este proceso puede abarcar ajustes en la estructura del código, eliminación de partes redundantes, mejora en la denominación de variables y funciones, y otras modificaciones destinadas a lograr un código más comprensible y manejable. A continuación, se describirá los estándares y las mejoras finales en las implementaciones.

- Revisión de aplicación de los principios de diseño de Material Design.
- Reestructuración y optimiza el código existente sin cambiar su comportamiento externo. El objetivo es mejorar la legibilidad, la eficiencia y la mantenibilidad.
- limpieza de código no utilizado en cada vista, modulo y función implementada.
- Corrección de las validaciones de la aplicación en relación con las funciones de "feedback del usuario". Estas funciones implican la comunicación de la aplicación con el usuario para informar sobre posibles errores o proporcionar información relevante durante el uso.
- Adición de comentarios de código para facilitar el desarrollo continuo a futuro.

6.3.5. Pruebas del sistema

Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema según los requisitos del cliente. se suelen realizar pruebas para validar que la aplicación móvil cumple con los requisitos establecidos y que funciona de manera esperada en el entorno real de los dispositivos móviles.

Debido a que este trabajo consiste en aplicar herramientas tecnológicas, se requiere de la verificación de las vistas y requerimientos funcionales con las que se comprobaran el cumplimiento de los requisitos del cliente. Estas pruebas se realizaron de manera incremental y registrándolo mediante los storycards y verificando mediante casos de prueba los requerimientos más importantes.

6.3.5.1. Esquema de navegación

Los esquemas de navegación se utilizan para planificar y organizar cómo los usuarios se moverán a través de una aplicación. Estos esquemas son esenciales para diseñar una experiencia de usuario intuitiva y eficiente [44], [46].

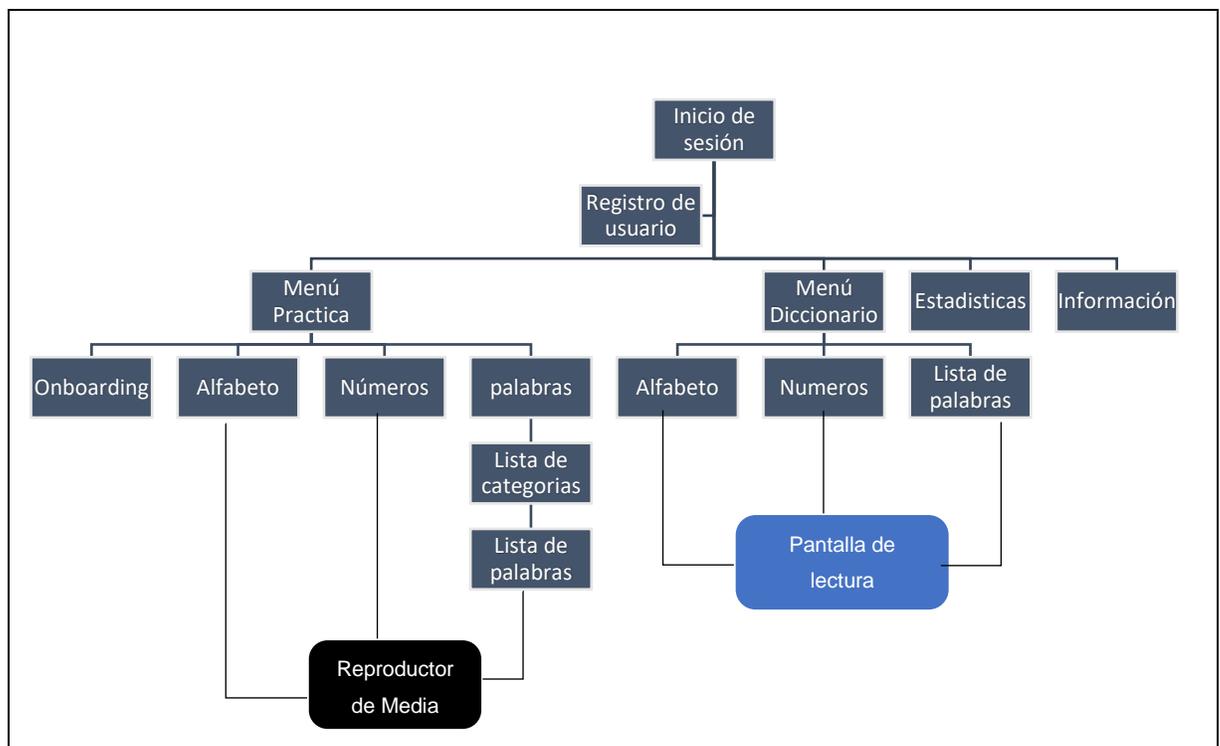


Fig. 16. Esquema de navegación. [autoría propia]

A continuación, se describirá cada campo del esquema:

Inicio de sesión: La pantalla de inicio de sesión permite a los usuarios existentes acceder a la aplicación ingresando sus credenciales, como nombre de usuario y contraseña. Este punto de entrada asegura que solo usuarios autorizados puedan acceder a la información personalizada.

Registro de usuario: La pantalla de registro de usuario es donde los nuevos usuarios pueden crear una cuenta proporcionando información básica como nombre, correo electrónico y una contraseña.

Onboarding: El proceso de Onboarding introduce a los nuevos usuarios a la aplicación, destacando sus características principales y guiando al usuario en sus funciones.

Menú de practica: El menú de práctica ofrece a los usuarios opciones para acceder a las diferentes listas de ejercicios.

Menú de diccionario: Este menú tiene las mismas opciones que el menú de práctica, pero desde aquí solo se puede acceder al menú de diccionario

Estadísticas: La sección de estadísticas muestra el progreso y rendimiento del usuario a lo largo del tiempo. Incluye métricas como tiempo dedicado a la práctica, puntuaciones y otros indicadores de progreso.

Información: Esta sección proporciona detalles sobre la aplicación como la información de contacto del centro educativo, los autores y colaboradores del trabajo monográfico.

Alfabeto: Presenta a los usuarios las letras del alfabeto desde las cuales puedes acceder a los ejercicios o al diccionario, dependiendo de si el usuario viene desde el menú de practica o del diccionario.

Números: Al igual que el alfabeto, esta presenta los números dígitos los cuales al seleccionarlos llevan al ejercicio en cuestión o al diccionario.

Lista de categorías: En esta vista se presentan las clasificaciones de las palabras como un método para organizarlas. Cada categoría agrupa palabras relacionadas con un mismo tema. Un ejemplo de esto es la categoría “Familia”, que incluye palabras como: mamá, papá, hermano, etc.

Lista de palabras: Esta vista presenta un listado de las palabras contenidas en una categoría, y desde aquí se puede acceder al ejercicio correspondiente a cada palabra.

Reproductor de medios: Aquí es donde se realizan los ejercicios relacionados con el reconocimiento de voz. También incluye un reproductor de video y un reproductor de audio.

Pantalla de lectura: En esta vista se muestra el significado de la palabra seleccionada desde el listado, e incluye un video de lenguaje de señas, un ejemplo de uso y una imagen que ilustra el significado.

6.3.5.2. Verificación de vistas

En este apartado se considera el estado de finalización de cada vista, en lo que respecta a usabilidad y navegación y diseño. Estas vistas fueron actualizadas conforme al ciclo de vida del trabajo monográfico avanzaba.

TABLA 20. VERIFICACIÓN DE VISTAS. [autoría propia]

Vista de la Aplicación	Estado de cumplimiento
Inicio de sesión	completado
formulario de ingreso	completado
Menú de practica	completado
Abecedario y números	completado
palabras simples	completado
Reproductor de media	completado

Vista de la Aplicación	Estado de cumplimiento
Menú de diccionario	completado
Diccionario de palabras, letras y números.	completado
Estadísticas	completado
Información	completado
Onboarding	completado

6.3.5.3. Pruebas funcionales

Las pruebas funcionales son un tipo de evaluación en el desarrollo de software que se centra en verificar si un sistema o una aplicación cumple con los requisitos y las especificaciones funcionales para los cuales fue diseñado. Esto se hace por medio de un análisis de los requerimientos funcionales y desde el punto de vista tanto del desarrollador como del usuario.

6.3.5.3.1. Casos de pruebas

Los casos de prueba son documentos detallados que describen las condiciones y los pasos necesarios para verificar que una funcionalidad o característica específica de una aplicación de software funciona correctamente. Son esenciales para la validación del software y aseguran que el producto final cumple con los requisitos especificados y funciona como se espera [52]. Con los casos de prueba se espera encontrar cualquier error que la aplicación pueda producir. En este caso se usaron en conjunto con las historias de usuario para realizar las verificaciones de los requerimientos más importantes.

A continuación, se describe cada campo de los casos de pruebas:

- **ID del Caso de Prueba:** Un identificador único para el caso de prueba.
- **Título:** Un nombre descriptivo para el caso de prueba.
- **Descripción:** Una breve explicación del propósito del caso de prueba.

- **Precondiciones:** Las condiciones que deben cumplirse antes de ejecutar el caso de prueba.
- **Pasos a Seguir:** Una lista de pasos para realizar la prueba.
- **Datos de Entrada:** Los datos necesarios para ejecutar el caso de prueba.
- **Resultado Esperado:** El resultado que se espera obtener si el software funciona correctamente.
- **Resultado Real:** El resultado obtenido después de ejecutar el caso de prueba (completado durante la prueba).
- **Estado:** Indica si el caso de prueba ha pasado o fallado, basado en la comparación entre el resultado real y el esperado.

TABLA 21. CASO DE PRUEBA, GRABACIÓN Y REPRODUCCIÓN. [autoría propia]

ID	CP001
Título	Verificar Funcionalidad de Grabación y Reproducción de Voz
Descripción	Este caso de prueba verifica que el usuario puede grabar su voz y reproducirla correctamente.
Precondiciones	La aplicación debe tener permisos de acceso al micrófono.
Pasos a seguir	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Navegar a la pantalla de grabación de voz. 3. Presionar el botón "Grabar" y pronunciar una palabra o frase. 4. Presionar el botón "Detener" para finalizar la grabación. 5. Presionar el botón "Reproducir" para escuchar la grabación.
Datos de entrada	Se grabo la frase "Hola, esto es una prueba"
Resultado esperado	<ul style="list-style-type: none"> • El sistema guarda la grabación correctamente. • El sistema reproduce la grabación sin errores.
Resultado real	La aplicación manda el resultado esperado
Estado	Aprobado

TABLA 22. CASO DE PRUEBA, GUARDADO DEL PROGRESO. [autoría propia]

ID	CP002
Título	Verificar Guardado del Progreso del Usuario
Descripción	Este caso de prueba verifica que la función de guardado funcione correctamente.
Precondiciones	El estudiante debe estar registrado previamente y conectado a internet.
Pasos a seguir	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Completar un ejercicio. 3. Presionar el botón de guardar 4. Confirmar el mensaje de guardado 5. Cerrar la aplicación. 6. Abrir la aplicación nuevamente. 7. Verificar que el progreso del ejercicio anterior se haya guardado.
Datos de entrada	Ejercicio de la palabra “mamá” completado.
Resultado esperado	El resultado del último intento es mostrado.
Resultado real	La aplicación manda el resultado esperado
Estado	Aprobado

TABLA 23. VERIFICAR EVALUACIÓN DE PRONUNCIACIÓN. [autoría propia]

ID	CP003
Título	Verificar Evaluación de Pronunciación y Retroalimentación Visual
Descripción	Este caso de prueba verifica que la aplicación evalúa correctamente la pronunciación del usuario y proporciona retroalimentación visual por medio del reconocimiento de voz y graficación del espectro de sonido.
Precondiciones	La aplicación debe tener permisos de acceso al micrófono y el estudiante debe haber iniciado sesión como un usuario registrado o invitado.
Pasos a seguir	<ol style="list-style-type: none"> 1. Iniciar sesión en la aplicación. 2. Navegar al menú de palabras. 3. Seleccionar una palabra o frase para practicar

	<ol style="list-style-type: none"> 4. Presionar el botón "Grabar" y pronunciar la palabra o frase 5. Presionar el botón "Detener" para finalizar la grabación. 6. Verificar la retroalimentación visual de la pronunciación por medio de códigos de colores y la gráfica del sonido.
Datos de entrada	Ejercicio de la palabra "Familia" completado.
Resultado esperado	<ul style="list-style-type: none"> • La palabra "familia" se muestra con colores: Verde en partes pronunciadas correctamente. Rojo para partes pronunciadas incorrectamente. Amarillo para sonidos adicionales innecesarios. • Se ha graficado correctamente el sonido de la voz del estudiante
Resultado real	La aplicación manda el resultado esperado
Estado	Aprobado

TABLA 24. CASO DE PRUEBA DE VERIFICACIÓN DE LAS ESTADÍSTICAS. [autoría propia]

ID	CP004
Título	Verificar la visualización de las estadísticas del estudiante
Descripción	Este caso de prueba verifica que el estudiantes pueda medir su progreso mediante las estadísticas.
Precondiciones	El estudiante debió haber completado uno o varios ejercicios.
Pasos a seguir	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Navegar al apartado de estadísticas con la barra de navegación. 3. Verificar que en lista de puntuaciones estén los últimos ejercicios completados
Datos de entrada	Se completo el ejercicio de las palabras: Dia, Ayuda Se completo ejercicio de la letra: Z/zeta.
Resultado esperado	El puntaje de todos los ejercicios completados, y el número de intentos se ve reflejado en las estadísticas.
Resultado real	La aplicación manda el resultado esperado
Estado	Aprobado

6.4. Estudio de factibilidad

El estudio de factibilidad es un proceso de evaluación y análisis que se realiza antes de comenzar un proyecto para determinar si es práctico, viable y beneficioso llevar a cabo el desarrollo de un sistema de software. Este estudio aborda diferentes aspectos del trabajo monográfico y proporciona información clave para la toma de decisiones antes de invertir recursos significativos [46].

6.4.1. Factibilidad técnica

Se refiere a la capacidad y viabilidad técnica para llevar a cabo la implementación exitosa del sistema propuesto, teniendo en cuenta si los desarrolladores tienen las herramientas y tecnología necesaria para llevar adelante el trabajo monográfico. Además, se definen los equipos que el cliente necesita para utilizar de manera efectiva el software final.

A continuación, se presentarán los recursos necesarios para el desarrollo del trabajo monográfico.

Propuesta numero 1:

TABLA 25. PROPUESTA 1 DE FACTIBILIDAD TÉCNICA. [autoría propia]

FACTOR HUMANO		
Nombre del recurso	Descripción	Cantidad
Área de desarrollo	Programador Full Stack	2
Intérprete	Logopeda o Profesor de personas con discapacidades auditivas	1
FACTOR DE HARDWARE		
Nombre del equipo	Descripción	Cantidad
Desktop/laptop	<ul style="list-style-type: none">• Procesador Intel Core I5• Memoria 8GB DDR4• Espacio de disco 250GB SSD	2

FACTOR DE SOFTWARE		
Nombre del Software	Descripción	Cantidad
Android Studio	Entorno de desarrollo (IDE)	2
Visual studio code	Entorno de desarrollo	2
Xampp	Gestor de base de datos	2
DaVinci Resolve	Editor de video	2
Servidor Web	Almacenamiento en la nube	1

Propuesta numero 2:

TABLA 26. PROPUESTA 2 DE FACTIBILIDAD TÉCNICA. [autoría propia]

FACTOR HUMANO		
Nombre del recurso	Descripción	Cantidad
Área de desarrollo	Programador Full Stack	2
Intérprete	Logopeda o Profesor de personas con discapacidades auditivas	1
FACTOR DE HARDWARE		
Nombre del equipo	Descripción	Cantidad
Desktop/laptop	<ul style="list-style-type: none"> • Procesador Intel Core I3 • Memoria 6GB DDR3 • Espacio de disco 250 GB SSD 	2
Smartphone	<ul style="list-style-type: none"> • Versión Android 9.0 Pie • Procesador Snapdragon 665 2GHz • Memoria RAM 4GB • Almacenamiento 32gb 	2
FACTOR DE SOFTWARE		
Nombre del Software	Descripción	Cantidad
Android Studio	Entorno de desarrollo (IDE)	2
Visual studio code	Entorno de desarrollo	2

Nombre del Software	Descripción	Cantidad
Xampp	Gestor de base de datos	2
DaVinci Resolve	Editor de video	2
Servidor Web	Almacenamiento en la nube	1

6.4.2. Factibilidad operativa

La factibilidad operativa es la evaluación de si el trabajo monográfico puede implementarse de manera práctica y eficiente en el entorno operativo existente. Implica analizar aspectos organizativos, de recursos humanos, procedimientos y procesos para determinar si la aplicación o sistema puede integrarse de manera efectiva en la estructura operativa de la organización.

La finalidad de AudiNoir es ser usado tanto por los profesores como punto de apoyo y los estudiantes como una herramienta de practica para la vocalización, por lo que se requiere de una herramienta amigable e intuitiva con la que los usuarios puedan adaptarse rápidamente. Se tiene conocimiento de que los profesores tienen acceso a smartphones y aunque los estudiantes no tendrán acceso en clase, podrán hacer uso de la aplicación en sus hogares.

Gracias al diseño de navegación, los usuarios solo requieren saber del uso básico de cualquier smartphone que tenga sistema operativo Android, aunque la app incluye una guía de las funciones básicas, las cuales son abordadas en la vista de recorrido de la aplicación (Onboarding). Gracias a esto, no se necesita una gran cantidad de capacitación, aunque es necesaria como introducción a la aplicación por parte de los desarrolladores.

Para utilizar AudiNoir, los usuarios solo necesitan un dispositivo móvil con sistema operativo Android y acceso a internet para utilizar todas las funciones de la aplicación. Además, es importante destacar que la aplicación está diseñada para ser compatible con las versiones de **Android 9.0 Pie** en adelante, siendo 9.0 Pie la versión mínima requerida.

Esta elección se basa en que cada dispositivo con esta versión cumple con los requisitos mínimos de hardware y compatibilidad del SDK (kit de desarrollo de software) con las herramientas utilizadas para el desarrollo.

Por último, se evalúa la cantidad de usuarios que tendrán acceso a AudiNoir según la versión específica de la aplicación, considerando aspectos de distribución y disponibilidad para diferentes usuarios. En las siguiente tabla de la web statcounter se observa las estadísticas del uso por versión de cada smartphone y Tablet actualmente activos en Nicaragua.

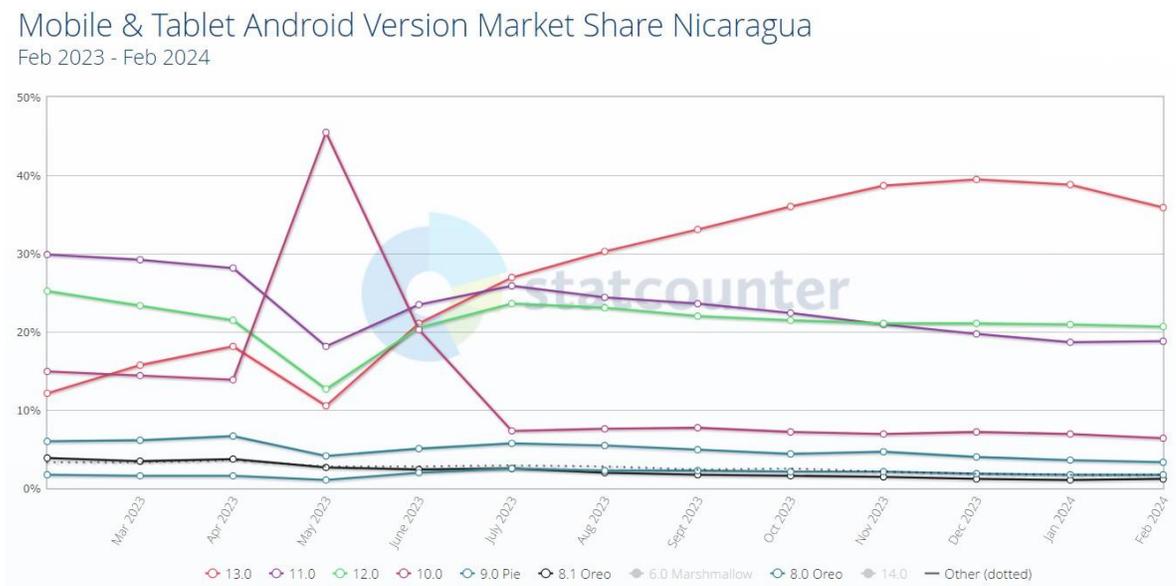


Fig. 17. Estadísticas de uso por versión en Nicaragua. [53]

Como resultado los smartphones con versión anterior al **Android 9.0 pie** actualmente activos es del 6.0%, smartphones que activamente utilizan la versión 9.0 es del 4.0% y los smartphones que utilizan versiones superiores a la 9.0 es del 90%.

6.4.3. Factibilidad económica

La factibilidad económica es un análisis que evalúa la viabilidad financiera de un trabajo monográfico o iniciativa. En este estudio se verá el valor económico reflejado anteriormente en los recursos mencionados para el desarrollo de la aplicación.

Los software que tienen un costo de \$0 en las siguientes tablas se encuentran bajo la licencia apache 2.0.

Factibilidad para propuesta numero 1:

TABLA 27. PROPUESTA 1 DE FACTIBILIDAD ECONÓMICA. [autoría propia]

Recurso	Cantidades	Precio unitario	Total
Humano			
Full Stack Developer	2	\$400	\$4000 / 5 meses
Interprete	2	\$100	\$200 / 1 día
Hardware			
<ul style="list-style-type: none"> ➤ Lenovo ThinkPad • Procesador Intel Core I5 • Memoria 8GB DDR4 • Espacio de disco 250GB SSD 	2	\$400	\$800
software			
Servidor Web	1 plan	\$80	\$400 / 5 meses
Android Studio	2	\$0	\$0
Visual studio code	2	\$0	\$0
DaVinci Resolve	2	\$0	\$0
Xampp	2	\$0	\$0

servicios			
Energía Eléctrica	1 plan	\$0,20	\$60 / 5 meses
Internet	1 plan	\$25,00	\$125/ 5 meses
Transporte	2 personas	\$0,10	\$15 / 5 meses
Total			\$5600

Factibilidad para propuesta numero 2:

TABLA 28. PROPUESTA 2 DE FACTIBILIDAD ECONÓMICA. [autoría propia]

Recurso	Cantidades	Precio unitario	Total
Humano			
Full Stack Developer	2	\$400	\$4000/5 meses
Interprete	2	\$100	\$200 / 1 día
Hardware			
➤ Dell Inspiron <ul style="list-style-type: none"> • Procesador Intel Core I3 • Memoria 6GB DDR3 • Espacio de disco 250 GB SSD 	2	\$200	\$ 400
➤ Redmi note 8 <ul style="list-style-type: none"> • Versión Android 9.0 Pie • Procesador Snapdragon 665 2GHz • Memoria RAM 4GB • Almacenamiento 32gb 	2	\$150	\$300
software			
Servidor Web	1 plan	\$80	\$400 / 5 meses
Android Studio	2	\$0	\$0
Visual studio code	2	\$0	\$0
Xampp	2	\$0	\$0

servicios			
Energía Eléctrica	1 plan	\$0,20	\$60 / 5 meses
Internet	1 plan	\$25,00	\$125/ 5 meses
Transporte	2 personas	\$0,10	\$15 / 5 meses
Total			\$5500

6.4.4. Cálculo del esfuerzo mediante Planning Poker .

Planning Poker es una técnica de estimación del esfuerzo colaborativa y basada en consenso, utilizada principalmente en metodologías ágiles como Scrum para estimar el esfuerzo necesario para completar una tarea o historia de usuario [54].

El objetivo de este proceso es principalmente llevar a cabo una conversación entre los miembros del equipo; en dicha conversación se explorarán los problemas a tratar y su posible solución [55]. El uso de Planning Poker fue aplicado en este trabajo monográfico para estimar el esfuerzo relativo de las tareas.

6.4.4.1. Procedimiento

En la reunión cada participante recibe un conjunto de tarjetas. Y el procedimiento que se lleva a cabo es el siguiente:

1. Las cartas se enumeran de manera que cuanto mayor sea el número de la carta, mayor es la incertidumbre.
2. A mayor incertidumbre, mayor esfuerzo. De esta forma, el Planning Poker facilita la estimación y reconsideración de las decisiones tomadas.

Para poder determinar el esfuerzo realizado para completar esta aplicación se considerarán los siguientes campos.

StoryCard: presenta el título de las historias de usuario que el equipo va a trabajar.

Puntos: Representan una estimación del esfuerzo necesario para completarla.

Días Invertidos: Detalla la cantidad de días laborales estimados que el equipo necesitará para completar cada historia de usuario, basándose en la equivalencia de puntos a tiempo calculada previamente.

Sprint o iteración: Esta es una fracción de tiempo comúnmente medido en 2 semanas laborales en metodologías de desarrollo ágil. Se usa para saber una estimación de iteraciones presentes en un proyecto.

TABLA 29. ESTIMACIÓN DEL ESFUERZO CON PLANNING POKER. [autoría propia]

StoryCard	Puntos	Días invertidos	Sprint
Inicio de sesión	10 puntos	5 Días	0.5 Sprint
Registro de usuario	21 puntos	10 Días	1.0 Sprint
Listado de palabras	22 puntos	11 Días	1.1 Sprint
Reconocimiento de voz	55 Puntos	27 Días	2.7 Sprint
Reproductor de media	44 puntos	22 Días	2.2 Sprint
Cálculo de resultados	17 puntos	8 Días	0.8 Sprint
Evaluación de la pronunciación	44 puntos	22 Días	2.2 Sprint
Guardar progreso	34 Puntos	17 Días	1.7 Sprint
Diccionario	21 puntos	10 Días	1.0 Sprint
Estadísticas	27 Puntos	14 Días	1.4 Sprint
Graficación del sonido	27 puntos	14 Días	1.4 Sprint
Correcciones	20 puntos	10 Días	1.0 Sprint
Total	422 puntos	170 Días	16 Sprint

Para poder realizar los cálculos correspondientes para obtener los puntos, días trabajados y el total de Sprint se realizó de la siguiente manera.

Tiempo promedio de duración de un sprint.

En promedio se requiere de total de 10 laborales o bien 2 semanas laborales de lunes a viernes. Pero también necesitamos convertir esos días en horas laborales.

$$\text{Total de horas por sprint} = 10 \text{ días} \times 8 \text{ horas por día} = 80 \text{ horas} \quad (2)$$

La velocidad promedio de trabajo.

El cálculo es relativo a las estimaciones subjetivas del esfuerzo en puntos, realizadas por el equipo de trabajo para las tareas durante un sprint.

$$\text{Velocidad promedio} = \frac{\sum_{i=1}^n \text{Estimación}_i}{n} \quad (3)$$

$$\text{Velocidad promedio} = \frac{22 + 18}{2} = 20 \text{ puntos}$$

Una vez obtenida la velocidad promedio por medio de la ecuación (3) se puede determinar la cantidad de horas que se le asignaría a cada punto

$$\frac{\text{total horas por sprint}}{\text{velocidad promedio}} = \text{velocidad de trabajo} \quad (4)$$

$$\frac{80 \text{ horas}}{20 \text{ puntos}} = 4 \text{ horas por puntos}$$

Según la ecuación (4) el equivalente a 4 horas laborales o medio día de trabajo es igual a **1 punto**.

Estimación del esfuerzo en tiempo

Para calcularlo en horas se multiplica los puntos de determinada tarea por la velocidad de trabajo estimada.

$$\begin{aligned} \text{Horas} &= \text{Puntos estimados} \times \text{velocidad de trabajo} \\ 5 \text{ puntos} \times 4 \text{ horas por puntos} &= 20 \text{ horas} \end{aligned} \quad (5)$$

Si la jornada laboral es de 8 horas, esto equivale a:

$$\begin{aligned} \text{Días} &= \frac{\text{horas}}{\text{horas por día}} \\ \text{Días} &= \frac{20 \text{ horas}}{8 \text{ horas por día}} = 2.5 \text{ días laborales} \end{aligned} \quad (6)$$

Calculadas las estimaciones se puede controlar de una manera más sencilla el tiempo o duración que tiene el presente trabajo monográfico y así también tener una mejor organización sobre los Sprint y tareas que se necesitaran ejecutar para el correcto funcionamiento del app.

En definitiva, las cualidades y el desarrollo del ciclo de vida de la metodología Mobile-D se han evidenciado a lo largo de cada fase de la elaboración de la aplicación, logrando un proceso ágil, cómodo y factible. Esto ha permitido obtener los mejores resultados para la aplicación final.

7. CONCLUSIONES

Durante el transcurso de este trabajo monográfico se descubrió la necesidad de los niños con hipoacusia. Aunque no pueden escuchar, tienen la posibilidad de desarrollar y entrenar el habla mediante técnicas y herramientas especializadas. Debido a la falta de estas herramientas se desarrolló una aplicación móvil que mediante el reconocimiento de voz y técnicas de educación especial mejorara las expresiones orales de estos niños.

Para encontrar una solución, se dedicó gran parte del tiempo a la **exploración** de herramientas y técnicas adecuadas para los métodos de enseñanza dirigidos a personas con discapacidades auditivas. Esto logro establecer una base sólida para orientar el diseño pedagógico de la aplicación móvil, asegurando que el software se adapte a las necesidades de estudiantes y maestros.

Se logro realizar una planificación de acuerdo a los requerimientos proporcionados por educadores especialistas y al ciclo de vida del trabajo monográfico.

Se logro desarrollar la aplicación utilizando la metodología Mobile-D, mediante las herramientas de desarrollo Android Studio, PHP y MySQL cumpliendo con los estándares técnicos establecidos.

Todas las funciones integradas en la aplicación fueron revisadas y estabilizadas conforme a las expectativas del cliente. Los maestros en educación especial del colegio Melania Lacayo participaron activamente en el trabajo monográfico, dando seguimiento y verificando el cumplimiento de los objetivos y la calidad esperada de la aplicación.

7.1. Recomendaciones

Gracias a la experiencia en el desarrollo de este trabajo se pueden tomar en consideración estos puntos como recomendación:

- Para dar continuidad a este trabajo monográfico, es necesario elaborar contenido continuamente mediante la colaboración entre expertos en discapacidad auditiva (logopedas, intérpretes, expertos en lenguaje de señas) y los desarrolladores de la aplicación móvil.
- Aunque la aplicación promueve la autonomía del estudiante en el aprendizaje, no puede reemplazar por completo el apoyo de un experto en la enseñanza a personas con necesidades especiales, como la discapacidad auditiva.
- Se recomienda que la universidad brinde soporte con un hosting. Esto facilitaría el desarrollo y mantenimiento de la aplicación y fomentaría la participación activa de la universidad en trabajo monográficos innovadores y de carácter social, ayudando a personas con discapacidades.
- Para el presente trabajo monográfico se utilizó una librería de IA de reconocimiento de voz preentrenada y de código abierto, lo que permitió reducir costos y tiempos de desarrollo, aunque con una menor precisión en los resultados. Para mejorar la precisión del reconocimiento y aumentar la cantidad y calidad de los datos obtenidos, se recomienda realizar un entrenamiento personalizado de la IA o utilizar una API de pago diseñada específicamente para la evaluación de la pronunciación.

8. BIBLIOGRAFÍA

- [1] E. Mora Gallardo, “Discapacidad y comunicación: una experiencia de fonética aplicada”, *Estudios de fonética experimental*, vol. 17, núm. 1, pp. 318–329, ene. 2008, [En línea]. Disponible en: <https://raco.cat/index.php/EFE/article/view/140075>
- [2] S. Serrato, “La discapacidad auditiva, ¿Cómo es el niño sordo?”, *Revista digital innovación y experiencias educativas*, vol. 1, núm. 16, pp. 1–10, mar. 2009, [En línea]. Disponible en: https://archivos.csif.es/archivos/andalucia/ensenanza/revistas/csicsif/revista/pdf/Numero_16/SABINA_PABON_2.pdf
- [3] Kinsta, “Móvil vs. Escritorio Cuota de Mercado en 2022”. Consultado: el 17 de agosto de 2022. [En línea]. Disponible en: <https://kinsta.com/es/movil-vs-escritorio-cuota-mercado/>
- [4] J. C. García, “La discapacidad auditiva. Principales modelos y ayudas técnicas para la intervención”, *Revista Internacional de apoyo a la inclusión, logopedia, sociedad y multiculturalidad*, vol. 1, núm. 1, Art. núm. 1, ene. 2015, Consultado: el 2 de febrero de 2022. [En línea]. Disponible en: <https://revistaselectronicas.ujaen.es/index.php/riai/article/view/4141>
- [5] D. Salwerowicz, “Design Proposal for a Software Tool for Speech Therapy - Modern Application Structure of Visual Speech Therapy App for Children”, M.S. Thesis, UiT - The Arctic University of Norway, Narvik, Nordland, 2019. doi: 10.13140/RG.2.2.16038.98885.
- [6] M. Puyuelo Sanclemente, “El visualizador fonético speech viewer. características”, *Revista de Logopedia, Foniatría y Audiología*, vol. 12, núm. 1, pp. 45–47, ene. 1992, doi: 10.1016/S0214-4603(92)75533-8.
- [7] J. L. Roset, “Project Fressa : globus 3”. 2017. Consultado: el 1 de febrero de 2022. [En línea]. Disponible en: <http://projectefressa.blogspot.com/2017/11/globus-3.html>
- [8] C. E. L. Rodas Sánchez, C. R. Lopez Santay, D. A. Gonzalez, y O. E. Sierra Pac, “Dispositivo interactivo para apoyar el desarrollo de la comunicación oral en personas con discapacidad auditiva en el Benemérito Comité Pro Ciegos y Sordos”, tesis doctoral, Universidad de San Carlos de Guatemala, Quetzaltenango, Guatemala, 2018. [En línea]. Disponible en: <http://ingenieria.cunoc.usac.edu.gt/portal/articulos/284e31dc31deaa392356e19b26603b032bdf0870.pdf>

- [9] M. Miranda, *Señas Nicas*. (el 18 de noviembre de 2018). Java. Nicaragua, Managua. Consultado: el 18 de mayo de 2022. [Android]. Disponible en: <https://github.com/mimg2805/SeniasNicas>
- [10] “Señas y Sonrisas - Nicaragua - Apps en Google Play”. Consultado: el 3 de septiembre de 2024. [En línea]. Disponible en: https://play.google.com/store/apps/details?id=com.manosunidas.senasdesentido&hl=es_SV
- [11] S. Cano, V. Peñeñory, C. A. Collazos, y S. Albiol-Pérez, “Designing Internet of Tangible Things for Children with Hearing Impairment”, *Information*, vol. 11, núm. 2, Art. núm. 2, feb. 2020, doi: 10.3390/info11020070.
- [12] T. Pérez Garzón, “Recursos Técnicos Para El Alumnado Con Discapacidad Auditiva”. 2010. [En línea]. Disponible en: https://archivos.csif.es/archivos/andalucia/ensenanza/revistas/csicsif/revista/pdf/Numero_29/TERESA_GARZON_2.pdf
- [13] G. Vega, “Acceso a internet desde el móvil”, *El País*. Consultado: el 16 de junio de 2022. [En línea]. Disponible en: <https://elpais.com/tecnologia/2020-03-05/el-915-de-los-internautas-ya-accede-a-internet-a-traves-del-movil.html>
- [14] A. Birlis, *Sonido para audiovisuales: manual de sonido*. Ugerman Editor, 2010. Consultado: el 26 de febrero de 2022. [En línea]. Disponible en: https://elibro.net/es/lc/unicaragua/titulos/78904?as_all=el__sonido&as_all_op=unaccent__icontains&prev=as
- [15] C. Espinoza Montes, *Metodología de Investigación Tecnológica, Pensando en Sistemas*. Universidad Nacional del Centro del Perú, 2014. Consultado: el 22 de febrero de 2022. [En línea]. Disponible en: <http://repositorio.uncp.edu.pe/handle/20.500.12894/1148>
- [16] N. Esteban Nieto, “Tipos de investigación”, *USDG*, p. 4, jun. 2018, [En línea]. Disponible en: <https://core.ac.uk/download/250080756.pdf>
- [17] H. F. M. Pérez, “Comunicación, Desempeño Laboral Y Discapacidad Auditiva”, *Orbis. Revista Científica Ciencias Humanas*, vol. 11, núm. 32, pp. 23–43, 2015, Consultado: el 22 de marzo de 2022. [En línea]. Disponible en: <https://www.redalyc.org/articulo.oa?id=70945572003>
- [18] Imagine Line, “La escala de medición de dB”. Consultado: el 28 de febrero de 2022. [En línea]. Disponible en: https://www.image-line.com/fl-studio-learning/fl-studio-online-manual/html/mixer_dB.htm
- [19] M. Bigot, *Apuntes de lingüística antropológica*. Rosario, Argentina: Centro Interdisciplinario de Ciencias Etnolingüísticas y Antropológico-sociales,

2010. Consultado: el 3 de marzo de 2022. [En línea]. Disponible en: <http://rephip.unr.edu.ar/xmlui/handle/2133/1367>
- [20] A. Ganeshan, “Los alófonos y la transcripción fonética”, *Universidad de Ohio*, núm. 1, jul. 2019, Consultado: el 28 de marzo de 2022. [En línea]. Disponible en: <https://linghispintro.pressbooks.com/chapter/los-alofonos/>
- [21] A. Montoto, E. Alvarez, G. Chavira, y J. Orozco, “El Reconocimiento de Voz como alternativa de inclusión para discapacitados auditivos en un entorno educativo”, *RelbCi*, vol. 5, núm. 3, p. 10, jun. 2018, Consultado: el 3 de marzo de 2022. [En línea]. Disponible en: <http://www.reibci.org/publicados/2018/jun/2900777.pdf>
- [22] H. D. Barrobés y M. R. Costa-jussà, “Reconocimiento automático del habla”, *FUOC*, vol. 9, núm. 1, p. 32, feb. 2021, Consultado: el 2 de febrero de 2022. [En línea]. Disponible en: [https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_\(Modulo_7\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_(Modulo_7).pdf)
- [23] J. C. Núñez, R. L. Castro, B. M. Delgado, S. S. Mora, y S. C. Casadiego*, “Modelo Acústico y de Lenguaje del Idioma Español para el dialecto Cucuteño, Orientado al Reconocimiento Automático del Habla”, *Ingeniería*, vol. 22, núm. 3, pp. 362–376, 2017, Consultado: el 8 de marzo de 2022. [En línea]. Disponible en: <https://www.redalyc.org/journal/4988/498853957005/html/>
- [24] H. L. Rufiner y D. H. Milone, “Sistema de reconocimiento automático del habla”, *Ciencia, Docencia y Tecnología*, vol. 15, núm. 28, pp. 151–177, 2004, Consultado: el 23 de marzo de 2022. [En línea]. Disponible en: <https://www.redalyc.org/articulo.oa?id=14502806>
- [25] J. Llisterri, “El reconocimiento automático del habla”. Consultado: el 16 de marzo de 2022. [En línea]. Disponible en: http://liceu.uab.es/~joaquim/speech_technology/tecnol_parla/recognition/speech_recognition/reconocimiento.html
- [26] J. L. R. Cabrera y L. R. González, *Sistemas informáticos*. Madrid, España: RA-MA Editorial, 2015. Consultado: el 28 de marzo de 2022. [En línea]. Disponible en: <https://elibro.net/es/lc/unicaragua/titulos/62481nicaragua/titulos/62481>
- [27] R. R. Vique, “Métodos para el desarrollo de aplicaciones móviles”, *FUOC*, vol. 1, núm. 1, p. 66, sep. 2011, Consultado: el 8 de marzo de 2022. [En línea]. Disponible en: https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarroll

o_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf

- [28] E. M. Suárez, “¿Que es una base de datos relacional?”, *Universidad de murcia, Murcia, España*, vol. 1, ago. 2008, Consultado: el 6 de marzo de 2022. [En línea]. Disponible en:
<http://www.uprh.edu/adem/Base%20de%20datos%20relacional.pdf>
- [29] A. S. Tanenbaum, *Redes de computadoras*, 5a ed., vol. 1. México: Pearson educación, 2012.
- [30] I. E. Marini, “El Modelo Cliente/Servidor”. 2012. [En línea]. Disponible en:
<https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>
- [31] Android Developers, “Introducción a Android Studio”. Consultado: el 4 de marzo de 2022. [En línea]. Disponible en:
<https://developer.android.com/studio/intro?hl=es-419>
- [32] Oracle, “¿Qué es Java y para qué es necesario?” Consultado: el 6 de marzo de 2022. [En línea]. Disponible en:
https://www.java.com/es/download/help/whatis_java.html
- [33] F. J. C. Sierra, *Java 2: lenguaje y aplicaciones*. Madrid, España: RA-MA Editorial, 2015. Consultado: el 7 de marzo de 2022. [En línea]. Disponible en:
https://elibro.net/es/lc/unificaragua/titulos/62458?as_all=java&as_all_op=unaccent__icontains&prev=as
- [34] O. A. Salazar, F. A. M. Aguirre, y J. A. C. Osorio, “Herramientas para el desarrollo rápido de aplicaciones web”, *Scientia et Technica*, vol. 1, núm. 47, pp. 254–258, 2011, Consultado: el 4 de marzo de 2022. [En línea]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=4525952>
- [35] D. D. Dvorski, “Installing, configuring, and developing with Xampp”, *Skills Canada*, p. 10, mar. 2007, Consultado: el 10 de marzo de 2022. [En línea]. Disponible en:
<http://dalibor.dvorski.net/downloads/docs/installingconfiguringdevelopingwithxampp.pdf>
- [36] phpMyAdmin contributors, “phpMyAdmin”. Consultado: el 4 de marzo de 2022. [En línea]. Disponible en: <https://www.phpmyadmin.net/>
- [37] Microsoft, “Documentation for Visual Studio Code”. Consultado: el 5 de marzo de 2022. [En línea]. Disponible en: <https://code.visualstudio.com/docs>

- [38] Alpha Cephei, "VOSK language model adaptation", VOSK Offline Speech Recognition API. Consultado: el 8 de marzo de 2022. [En línea]. Disponible en: <https://alphacephei.com/vosk/lm>
- [39] Square, "Retrofit". Consultado: el 16 de junio de 2022. [En línea]. Disponible en: <https://square.github.io/retrofit/>
- [40] google, "Volley, android developers", Android Developers. Consultado: el 16 de junio de 2022. [En línea]. Disponible en: <https://developer.android.com/training/volley?hl=es-419>
- [41] "DaVinci Resolve 19 | Blackmagic Design". Consultado: el 27 de mayo de 2024. [En línea]. Disponible en: <https://www.blackmagicdesign.com/products/davinciresolve>
- [42] C. Cruz del Castillo, S. Olivares Orozco, y M. González García, *Metodología de la investigación*, 1a ed. México: Grupo Editorial Patria, 2014. Consultado: el 22 de febrero de 2022. [En línea]. Disponible en: <https://elibro.net/es/lc/unicaragua/titulos/39410>
- [43] A. C. Spataru, "Agile development methods for mobile applications", M.S. Thesis, University of Edinburgh, Edinburgh, Scotland, 2010.
- [44] David Hernandez, "Mobile D (programacion dispositivos moviles)", 22:31:49 UTC. Consultado: el 8 de marzo de 2022. [En línea]. Disponible en: https://es.slideshare.net/pipehernandez1020/mobile-d-programacion-dispositivos-moviles?from_action=save
- [45] R. S. Pressman y B. Maxim, *Software Engineering: A Practitioner's Approach*. New York, NY, 2014.
- [46] A. M. B. Goyeneche, D. Briceño, A. C. N. Lemus, y O. S. Morales, "Desarrollo ágil de una aplicación para dispositivos móviles. Caso de estudio: Taxímetro móvil", *Ingeniería*, vol. 21, núm. 3, p. 1, 2016, Consultado: el 6 de marzo de 2022. [En línea]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=5677874>
- [47] E. B. Pantoja, "El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing", vol. 2, 2004.
- [48] T. Salami, "The MVC Framework", Medium. Consultado: el 21 de septiembre de 2024. [En línea]. Disponible en: <https://tsalami44.medium.com/the-mvc-framework-cfa43889da7>
- [49] "Material Design". Consultado: el 27 de mayo de 2024. [En línea]. Disponible en: <https://m3.material.io/>

- [50] C. O'hEocha y K. Conboy, "The Role of the User Story Agile Practice in Innovation", en *Lean Enterprise Software and Systems*, vol. 65, P. Abrahamsson y N. Oza, Eds., en *Lecture Notes in Business Information Processing*, vol. 65. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 20–30. doi: 10.1007/978-3-642-16416-3_3.
- [51] H. Sharp, H. Robinson, J. Segal, y D. Furniss, "The role of story cards and the wall in XP teams: a distributed cognition perspective", en *AGILE 2006 (AGILE'06)*, jul. 2006, p. 11 pp. – 75. doi: 10.1109/AGILE.2006.56.
- [52] J. L. Aristegui O., "Los casos de prueba en la prueba del software", *Lámpsakos*, núm. 3, pp. 27–34, 2010, Consultado: el 27 de mayo de 2024. [En línea]. Disponible en:
<https://dialnet.unirioja.es/servlet/articulo?codigo=3399441>
- [53] "Mobile & Tablet Android Version Market Share Nicaragua", StatCounter Global Stats. Consultado: el 27 de mayo de 2024. [En línea]. Disponible en:
<https://gs.statcounter.com/android-version-market-share/mobile-tablet/nicaragua>
- [54] M. Cohn, *Agile Estimating and Planning*. Pearson Education, 2005.
- [55] J. Grenning, "Planning Poker or How to avoid analysis paralysis while release planning", 2002.

9. ANEXOS

A. Diagrama de clases

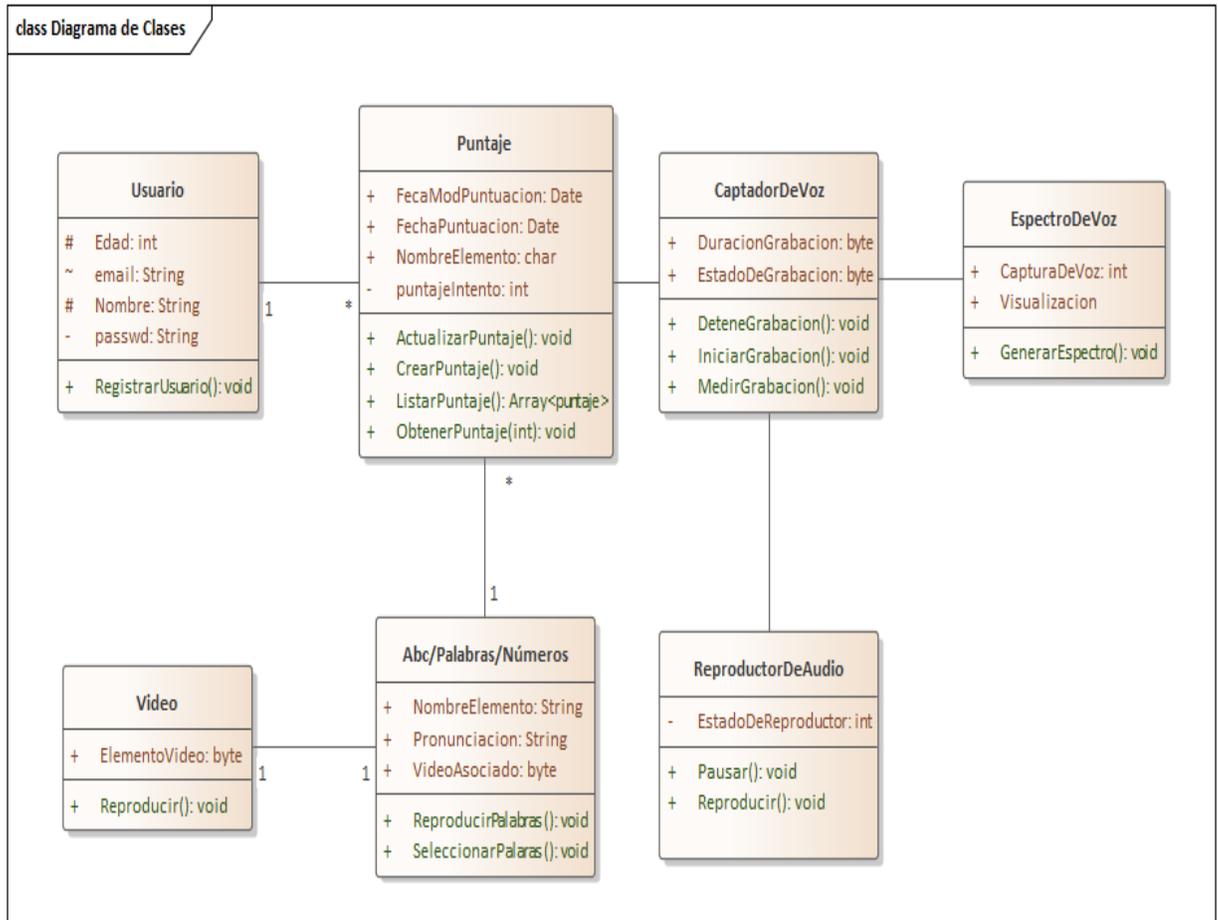


Fig. 18. Diagrama de clases. [autoría propia]

B. Diagramas de actividad

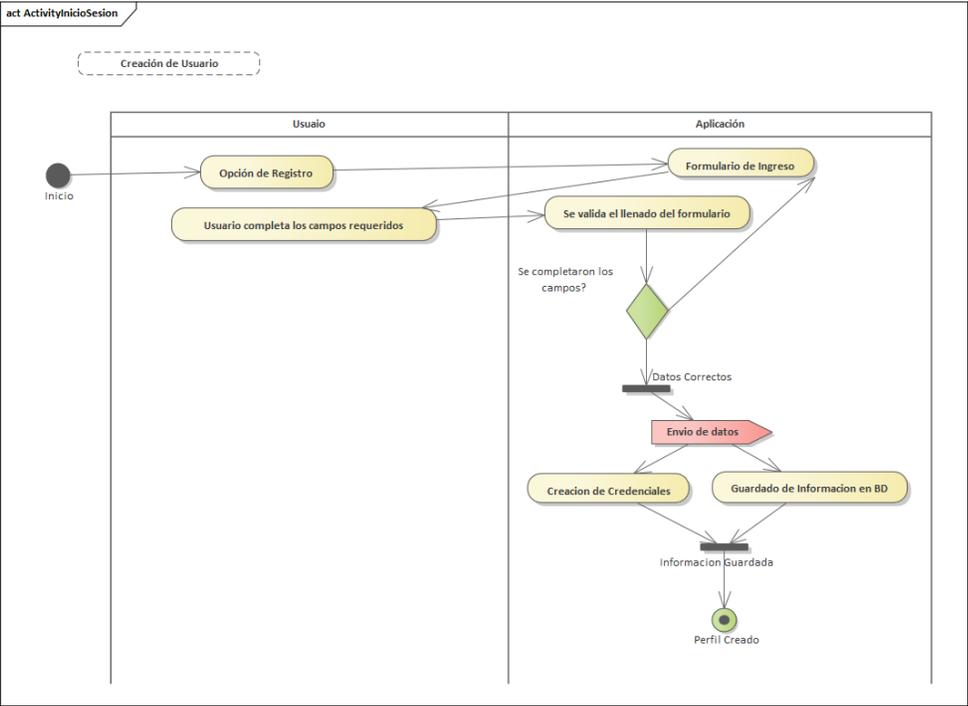


Fig. 19. Diagrama de actividad de formulario de ingreso. [autoría propia]

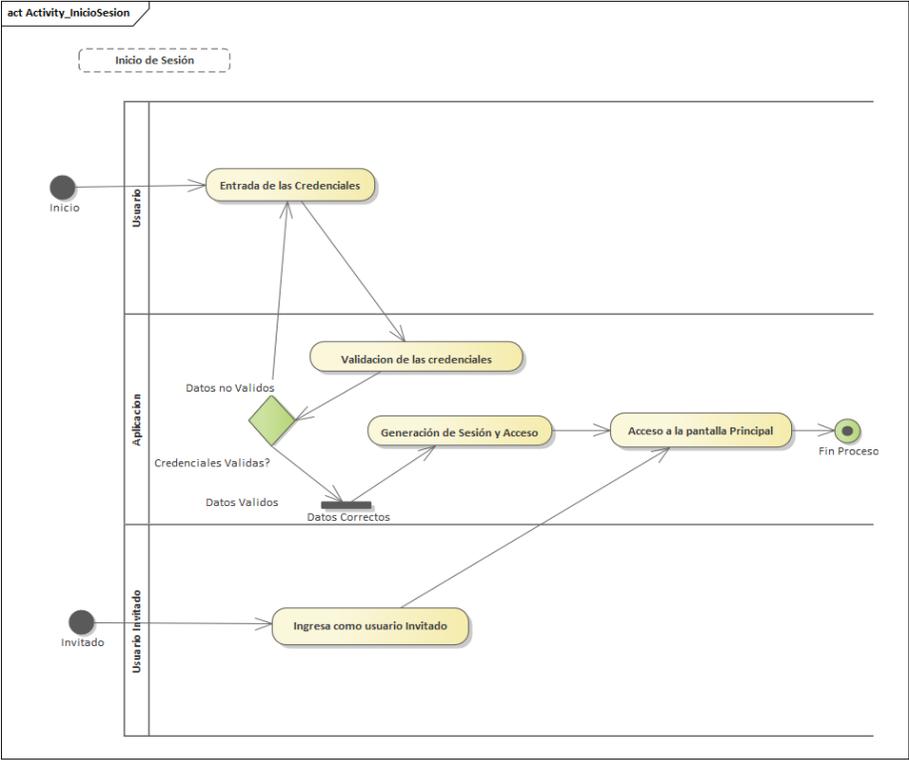


Fig. 20. Diagrama de actividad de inicio de sesión. [autoría propia]

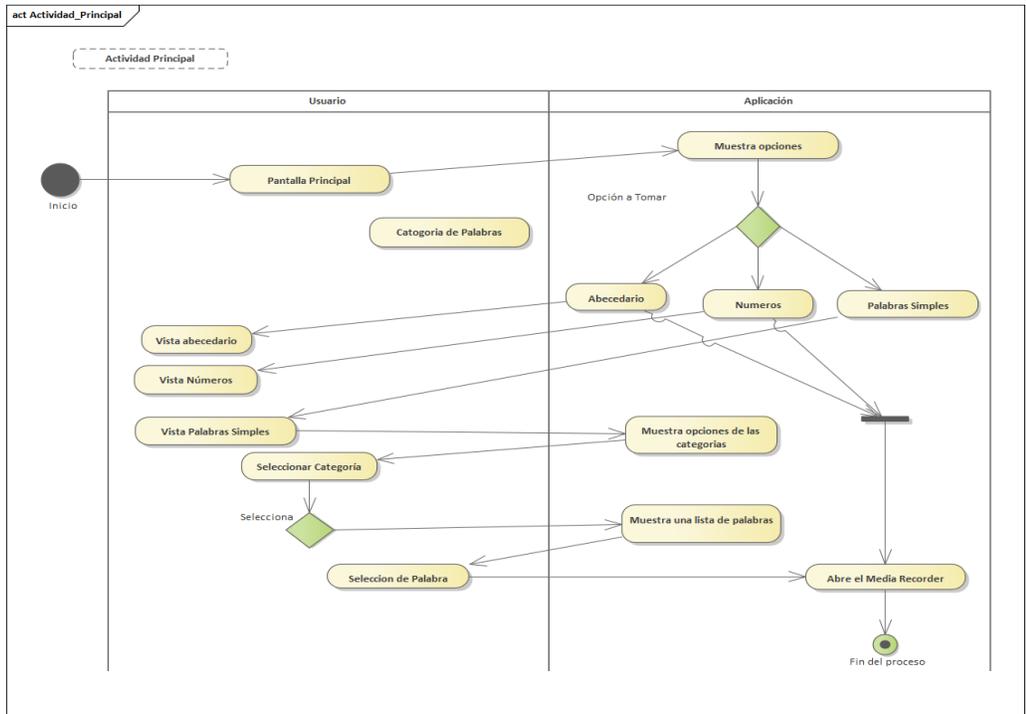


Fig. 21. Diagrama de actividad pantalla principal. [autoría propia]

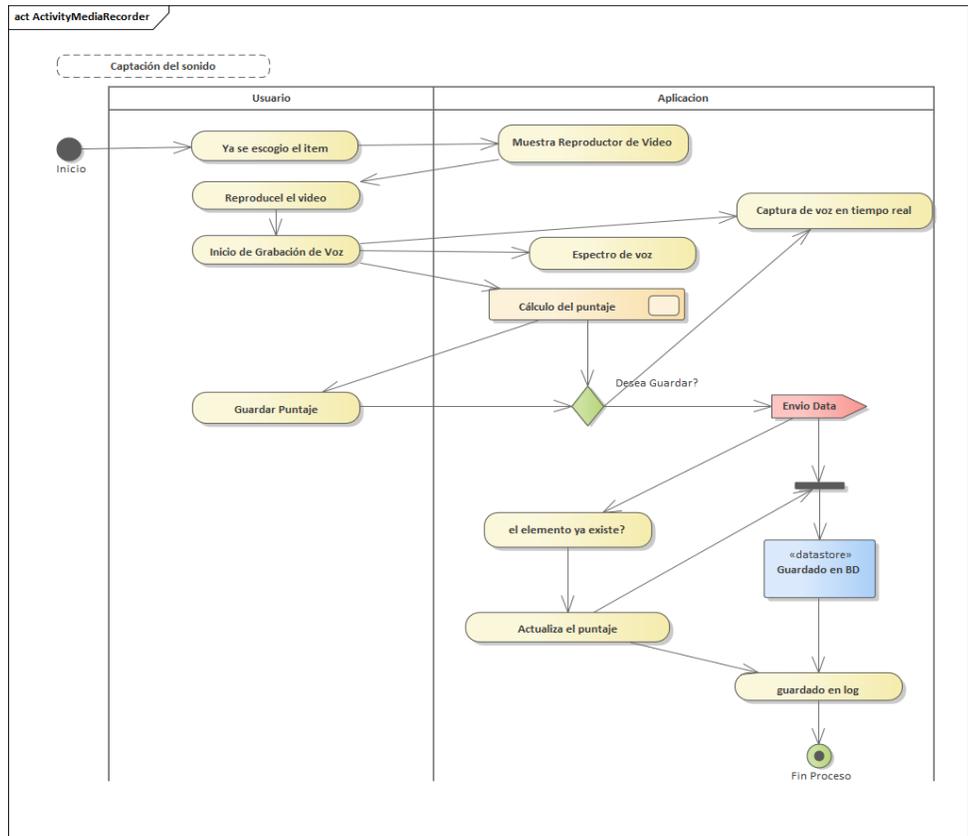


Fig. 22. Diagrama de actividad reproductor de media. [autoría propia]

C. Diagrama de base de datos

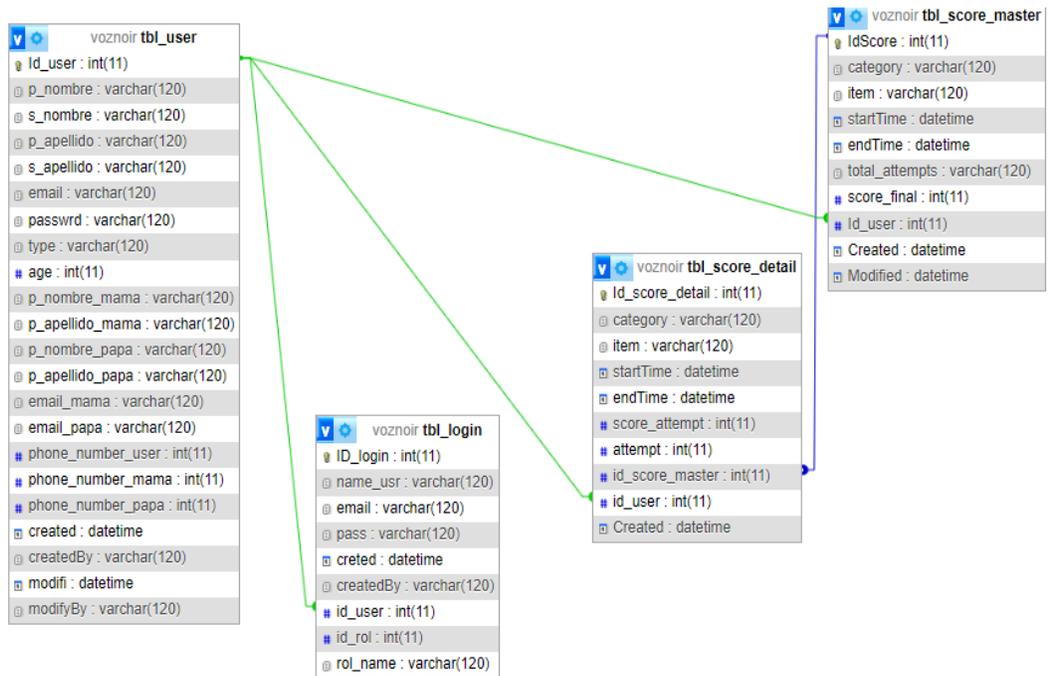


Fig. 23. Diagrama de base de datos. [autoría propia]

D. Diagrama de componentes

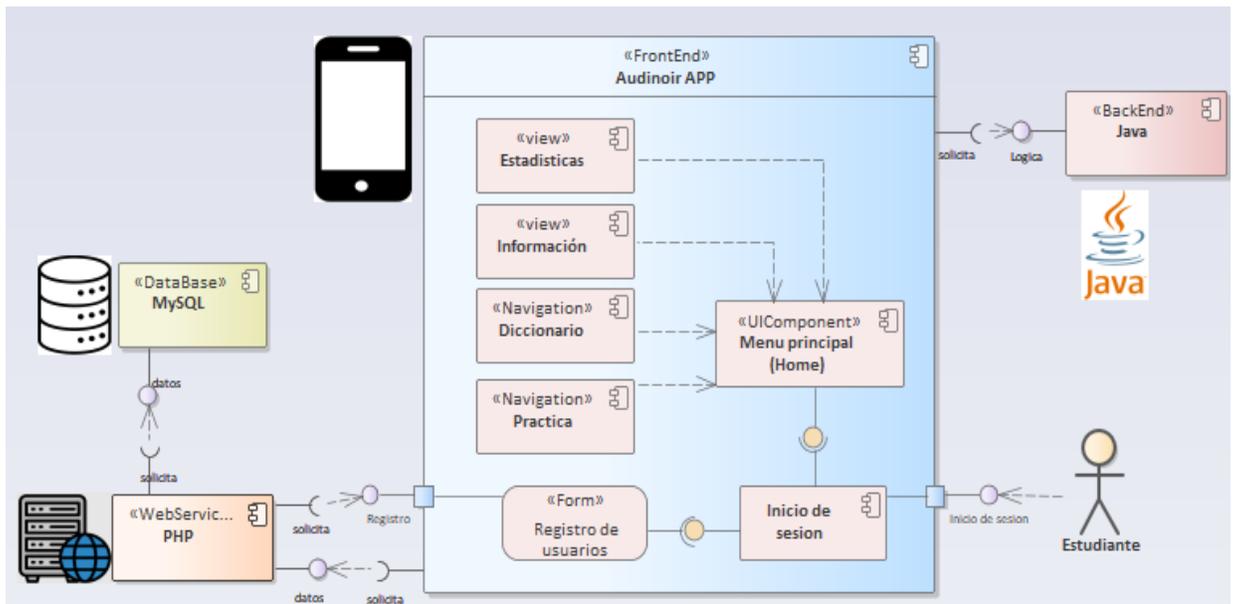


Fig. 24. Diagrama de componentes, función externa. [autoría propia]

