UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TEGNOLOGÍAS DE LA INFORMACIÓN

Tema:

APLICACIÓN MÓVIL BASADA EN ANDROID, PARA FACILITAR LA COMUNICACIÓN DE PERSONAS CON DISCAPACIDAD VISUAL

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Tecnologías de la Información.

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Ronny Alejandro Del Pezo Quinde

TUTOR: Ing. Carlos Israel Nuñez Miranda Mg.

Ambato - Ecuador

febrero – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: APLICACIÓN MÓVIL BASADA EN ANDROID, PARA FACILITAR LA COMUNICACIÓN DE PERSONAS CON DISCAPACIDAD VISUAL, desarrollado bajo la modalidad Proyecto de Investigación por el señor Ronny Alejandro Del Pezo Quinde, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024

_____

Ing. Carlos Israel Nuñez Miranda Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: APLICACIÓN MÓVIL BASADA EN ANDROID, PARA FACILITAR LA COMUNICACIÓN DE PERSONAS CON DISCAPACIDAD VISUAL es absolutamente original, autentico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024

Ronny Alejandro Del Pezo Quinde

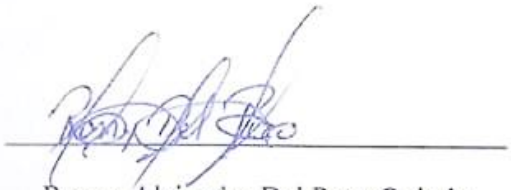
C.C. 0928388214

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de discusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para Titulación de Grado en la universidad Técnica de Ambato.

Ambato, febrero 2024.



Ronny Alejandro Del Pezo Quinde

C.C. 0928388214

AUTOR

APROBACION DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Ronny Alejandro Del Pezo Quinde, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado APLICACIÓN MÓVIL BASADA EN ANDROID, PARA FACILITAR LA COMUNICACIÓN DE PERSONAS CON DISCAPACIDAD VISUAL, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Articulo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos. Conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

Firmado electrónicamente por:
ELSA PILAR URRUTIA URRUTIA

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Firmado electrónicamente por:
PAULO CESAR TORRES ABRIL

Ing. Paulo Cesar Torres Abril, Mg
PROFESOR CALIFICADOR

Firmado electrónicamente por:
SANTIAGO DAVID JARA MOYA

Ing. Santiago David Jara Moya, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

El presente proyecto está dedicado a la mujer más increíble que he tenido el privilegio de llamar madre, Andrea Quinde. Su apoyo constante, sus palabras de aliento y su inquebrantable fe en mí hicieron posible este logro.

A mis hermanos Karelys y Jeremy, quienes han llenado mi vida de alegría y amor incondicional. Este logro no es solo mío, sino también de ustedes, mis mayores motivadores. Gracias por ser mi fuente constante de inspiración. Espero poder ser un ejemplo para ustedes toda la vida.

A mi amada familia, especialmente a mis abuelos María Vera y Pedro Quinde, cuya sabiduría y amor me acompañan siempre a pesar de las distancias.

En memoria de Martha Quinde, quien en vida supo apoyarme en mis pasos universitarios. Este logro académico lleva su huella, aunque no se encuentre físicamente conmigo. En cualquier lugar de la existencia que se encuentre, sé que mira esto con alegría. Gracias por ser mi tía. Descansa en paz.

A mis queridos tíos Sara, Danilo y Pedro. A mis primos Alfredo, Anthony, Nohely, Nayely y Dayana, cómplices de risas y compañeros de aventuras. Y a cada miembro de la familia que siempre creyó en mí, Hoy celebro este logro, que, sin duda, lleva la marca de su aliento y confianza.

AGRADECIMIENTO

Agradezco a cada miembro de mi familia, quienes contribuyeron con su granito de arena tanto económicamente como emocionalmente, durante mi educación. Sin su apoyo incondicional, este logro no habría sido posible.

Mi gratitud se extiende también hacia mi tutor, el Ing. Carlos Núñez, cuya confianza, paciencia y sabiduría fueron fundamentales en el desarrollo de este proyecto.

A mis queridos compañeros de carrera de NSDT1P, les agradezco por todo. No solo fueron colegas, sino amigos que ahora forman parte importante de mi historia.

Quisiera expresar mi gratitud a las personas generosas que me brindaron más que una simple ayuda: una comida, un lugar donde dormir y, sobre todo, su amistad. Agradezco a los amigos en Totoras, quienes me abrieron las puertas de sus hogares y me recibieron como a otro miembro de su familia.

A mis amigos de toda la vida, Lady e Israel, quienes han sido compañeros de aventuras desde los tiempos del colegio y han permanecido como mis mejores amigos, les agradezco por su lealtad, apoyo y por compartir conmigo la maravillosa travesía de la vida.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA ..... **i**

APROBACIÓN DEL TUTOR ..... **ii**

AUTORÍA ..... **iii**

DERECHOS DE AUTOR ..... **iv**

APROBACION DEL TRIBUNAL DE GRADO ..... **v**

DEDICATORIA ..... **vi**

AGRADECIMIENTO ..... **vii**

ÍNDICE GENERAL DE CONTENIDOS ..... **viii**

ÍNDICE DE TABLAS ..... **xi**

ÍNDICE DE FIGURAS ..... **xii**

ÍNDICE DE ANEXOS ..... **xiv**

RESUMEN EJECUTIVO ..... **xv**

ABSTRACT ..... **xvi**

CAPÍTULO I.- MARCO TEÓRICO **1**

1.1. Tema de Investigación 1

1.1.1. Planteamiento del Problema 1

1.2. Antecedentes Investigativos 2

1.3. Fundamentación Teórica 4

1.4. Objetivos 9

1.4.1. Objetivo General 9

1.4.2. Objetivos Específicos 9

CAPÍTULO II.- METODOLOGÍA **10**

2.1. Materiales 10

2.2. Métodos 14

2.2.1. Modalidad de la Investigación 14

2.2.3. Recolección de la Información....16
2.2.4. Procesamiento y Análisis de Datos....32
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN....34
3.1 Análisis y discusión de los resultados....34
3.1.1 Análisis de dificultades al manejar aplicativos para comunicarse. 34
3.1.2 Análisis de necesidades y funcionalidades posibles para aplicativo especializado para personas con discapacidad visual. 36
3.1.3 Análisis de librerías para el uso de comandos de voz. 39
3.1.3 Análisis de metodologías para el desarrollo móvil 43
3.1.4 Fase de la metodología Mobile D 46
3.2 Desarrollo de la propuesta....47
3.2.1 Fase 1 Exploración....47
a. Definición de los Stakeholder 47
b. Definición del alcance....48
c. Establecimiento del proyecto 49
3.2.2 Fase 2 Inicialización....50
a. Configuración del Proyecto....50
Preparación del Ambiente 50
Planificación de las fases....51
Diseño de la arquitectura de la aplicación....52
Base de datos NoSQL 53
Esquema de Navegabilidad....54
Diagramas de casos de uso....55
Prototipo de la aplicación....55
3.2.3 Fase 3 Producción 58

a. Backend 58
b. Archivos de Código Fuente 59
c. Archivos de Recursos 66
d. Archivos de Configuración 67
e. StoryCard 70
3.2.4 Fase 4 Estabilización 82
3.2.5 Fase 5 Pruebas 83
CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES 94
4.1 Conclusiones 94
4.2 Recomendaciones 95
REFERENCIAS BIBLIOGRÁFICAS 96
ANEXOS 98

ÍNDICE DE TABLAS

Tabla 1.Población de estudio 15
Tabla 2. Cuadro comparativo de librerías para uso de comandos de voz. 39
Tabla 3. Cuadro comparativo de Metodologías para el desarrollo móvil 43
Tabla 4. Planificación de las fases. 51
Tabla 5. StoryCard - Conexiones de servicios de Firebase con Android Studio. 70
Tabla 6. StoryCard - Diseño de inicio de Sesión y Registro de usuarios. 71
Tabla 7. StoryCard - Conexiones de autenticación. 73
Tabla 8. StoryCard – Verificación Automática por mensaje OTP. 74
Tabla 9. StoryCard - Almacenar Datos en Realtime Database. 75
Tabla 10. StoryCard - Contactos Celular y Usuarios de Firebase. 76
Tabla 11. StoryCard - Comando Listar Contactos 77
Tabla 12. StoryCard - Comando de Envío de Mensajes. 78
Tabla 13. Comando de reproducción de Mensajes 79
Tabla 14. StoryCard - Interfaz para envío y recepción de mensajes. 80
Tabla 15. StoryCard - Funcionalidades de configuración 81
Tabla 16. StoryCard - Interfaz de configuraciones. 82
Tabla 17. Prueba de aceptación 1 89
Tabla 18. Prueba de aceptación 2 89
Tabla 19. Prueba de aceptación 3 90
Tabla 20. Prueba de aceptación 4 90
Tabla 21. Prueba de aceptación 5 91
Tabla 22.Tiempo de Interacción 91
Tabla 23. Eficiencia de Navegación 92
Tabla 24. Evaluación de Accesibilidad 92
Tabla 25. Errores 93

ÍNDICE DE FIGURAS

Figura 1. Tabulación de resultados pregunta 1. 16
Figura 2. Tabulación de resultados pregunta 2. 17
Figura 3. Tabulación de resultados pregunta 3. 18
Figura 4. Tabulación de resultados pregunta 4. 18
Figura 5. Tabulación de resultados pregunta 5. 19
Figura 6. Tabulación de resultados pregunta 6. 20
Figura 7. Tabulación de resultados pregunta 7. 20
Figura 8. Tabulación de resultados pregunta 8. 21
Figura 9. Tabulación de resultados pregunta 9. 22
Figura 10. Tabulación de resultados pregunta 10. 22
Figura 11. Tabulación de resultados pregunta 11. 23
Figura 12. Tabulación de resultados pregunta 12. 24
Figura 13. Tabulación de resultados pregunta 13. 24
Figura 14. Tabulación de resultados pregunta 14. 25
Figura 15. Tabulación de resultados pregunta 15. 26
Figura 16. Tabulación de resultados pregunta 16. 26
Figura 17. Tabulación de resultados pregunta 17. 27
Figura 18. Tabulación de resultados pregunta 18. 28
Figura 19. Tabulación de resultados pregunta 19. 28
Figura 20. Tabulación de resultados pregunta 20. 29
Figura 21. Fases de la Metodología Mobile D. 46
Figura 22.Arquitectura Orientada a Servicios (SOA). 52
Figura 23. Colecciones en Firebase. 53
Figura 24. Colección Messages 53
Figura 25. Colección Users. 54
Figura 26. Esquema de Navegabilidad. 54
Figura 27. Caso de uso para usuarios con discapacidad visual. 55
Figura 28. Registro e Inicio de Sesión. 56
Figura 29. Verificación de Número Celular 56
Figura 30. Mensajes. 57

Figura 31. Configuraciones. 58
Figura 32. ConfiguracionesFragment 59
Figura 33. EnviarMensajesFragment 60
Figura 34. FCMSend 61
Figura 35.MainActivity 62
Figura 36. Clase Messages 62
Figura 37. Notification_Receiver 63
Figura 38. OTP_Receiver 64
Figura 39. PushNotificationService 65
Figura 40. RegisterActivity 65
Figura 41. TabsAccesorAdapter 66
Figura 42. Permisos en AndroidManifest.xml 67
Figura 43. Ejemplo de código de importaciones. 68
Figura 44. Ejemplo de código de interfaz de Registro. 69
Figura 45. Implementaciones en build Gradle para la utilización de servicios Firebase. 69
Figura 46. Conexión de Android Studio a Servicios de Firebase. 71
Figura 47. Diseño de Registro e Inicio de Sesión. 72
Figura 48. Registro e Inicio de Sesión. 84
Figura 49. Confirmar el número con que se registra el usuario. 84
Figura 50. Verificación Automática. 85
Figura 51. Pantalla Principal (Mensajes/Configuraciones). 86
Figura 52. Comando Contactos en Ejecución 86
Figura 53. Pantalla Confirmación de Mensaje. 87
Figura 54. Reproduciendo Mensajes 88
Figura 55. Pantalla de Configuraciones 88

ÍNDICE DE ANEXOS

Anexo 1. Verificación de un número correcto 98
Anexo 2. Delimitación del código del país. 98
Anexo 3. Verificación y validación de código OTP. 99
Anexo 4. Actualizar Datos del usuario. 99
Anexo 5. Envío de datos a Firebase. 100
Anexo 6. Lista de Usuarios en Firebase. 100
Anexo 7. Lista de Contactos en el dispositivo celular. 101
Anexo 8. Creación de lista comparativa entre usuarios Firebase y contactos del dispositivo. 101
Anexo 9. Código de comando de contactos. 102
Anexo 10. Comando para envío de mensajes. 102
Anexo 11. Confirmación de mensaje con comando. 103
Anexo 12. Configuraciones 103
Anexo 13. Código de opciones desplazables 104
Anexo 14. Manual de usuario 105

RESUMEN EJECUTIVO

El presente proyecto se centra en abordar las necesidades de las personas en cuanto a la comunicación a través de dispositivos móviles. Con este propósito, se llevó a cabo una recopilación de datos para identificar las necesidades y dificultades que enfrentan en su comunicación diaria. El objetivo principal consistió en el desarrollo de una aplicación que no solo facilitara sus habilidades de comunicación, sino que también les brindara la oportunidad de participar activamente en conversaciones de chat con otros usuarios.

La aplicación se construyó utilizando Java en la plataforma Android, y se implementaron comandos de voz con el fin de permitir que los usuarios controlaran la recepción y el envío de mensajes de texto mediante instrucciones orales. Este enfoque no solo buscaba mejorar la accesibilidad, sino también hacer que la aplicación fuera más intuitiva y fácil de usar para aquellos con discapacidades visuales. Para llevar a cabo el desarrollo de manera eficiente y efectiva, se adoptó la metodología Mobile D, garantizando así un proceso ágil que cumplió cada una de sus fases.

El enfoque primordial de esta investigación es proporcionar una solución específica para personas con discapacidades visuales. La aplicación se diseñó con la premisa de que los usuarios pudieran utilizarla de manera autónoma, sin depender de aplicaciones adicionales o asistentes virtuales externos.

Palabras clave: Discapacidad visual, Java, Android, Mobile D, comandos de voz.

ABSTRACT

This project focuses on addressing people's needs in terms of communication via mobile devices. For this purpose, data collection was carried out to identify the needs and difficulties they face in their daily communication. The main objective was to develop an application that would not only facilitate their communication skills, but also give them the opportunity to actively participate in chat conversations with other users.

The application was built using Java on the Android platform, and voice commands were implemented in order to allow users to control the receiving and sending of text messages through spoken instructions. This approach not only sought to improve accessibility, but also to make the application more intuitive and easier to use for those with visual impairments. In order to carry out the development efficiently and effectively, the Mobile D methodology was adopted, ensuring an agile process that fulfilled each of its phases.

The primary focus of this research is to provide a specific solution for people with visual impairments. The application was designed with the premise that users would be able to use it autonomously, without relying on additional applications or external virtual assistants.

Keywords: Visual impairment, Java, Android, Mobile D, voice commands.

CAPÍTULO I.- MARCO TEÓRICO

1.1. Tema de Investigación

APLICACIÓN MÓVIL BASADA EN ANDROID, PARA FACILITAR LA COMUNICACIÓN DE PERSONAS CON DISCAPACIDAD VISUAL

1.1.1. Planteamiento del Problema

A nivel mundial, 2,200 millones de personas están experimentando un deterioro moderado o grave de la visión, entre las cuales se encuentran aquellas que padecen ceguera debido a errores de refracción no corregidos (88.4 millones), cataratas (94 millones), degeneración macular relacionada con la edad (8 millones), glaucoma (7.7 millones), retinopatía diabética (3.9 millones), así como deterioro de la visión cercana causado por presbicia no corregida (826 millones) [1]. Ante la gran cantidad de personas con problemas de visión, es necesario promover la inclusión social, ya que el derecho a vivir de una manera digna corresponde a cada individuo en el mundo.

Las personas con discapacidad visual en Latinoamérica recientemente enfrentaron dificultades durante la pasada pandemia de Covid-19, ya que, debido a sus necesidades, debían estar en constante contacto con quienes los apoyan. Aquellos que vivían solos pudieron experimentar un acceso limitado a la comunicación con centros de información, los cuales son fundamentales para proporcionar pautas sobre cómo proceder en una pandemia y notificar acerca de los cuidados necesarios. Por esta razón, es crucial que cada persona con discapacidad visual tenga un medio para relacionarse con otros y obtener así el beneficio de la interacción social e información.

En Ecuador, según las estadísticas proporcionadas por el Consejo Nacional para la Igualdad de Discapacidades (CONADIS), se destaca la presencia de 54,397 personas con discapacidad visual. De estas, 1,371 están registradas en la provincia de Tungurahua [2]. Las dificultades que enfrentan en su vida diaria abarcan desde problemas de movilidad

hasta dificultades para acceder a servicios de manera eficiente. Además, enfrentan obstáculos al manipular dispositivos tecnológicos de manera sencilla, especialmente para la actividad básica de comunicarse. Estas limitaciones impactan negativamente en la contribución de estas personas a la sociedad y en su participación en el ámbito laboral, lo que a menudo resulta en el desarrollo de personalidades tímidas o depresivas.

En relación con la problemática expuesta, existen aplicativos móviles diseñados para mejorar las competencias básicas de personas con discapacidad visual. Sin embargo, no hay uno específicamente creado para facilitar la comunicación de manera ágil, ajustándose a sus necesidades particulares, ya que estas personas no pueden manejar un dispositivo móvil de la misma manera que un usuario con todas sus capacidades físicas.

Debido a que las personas con discapacidad visual no siempre cuentan con los recursos para adquirir un smartphone actualizado o un asistente de voz como Alexa o Cortana, que les facilite la manera de relacionarse con la tecnología y tener una comunicación más amplia con personas fuera de su entorno familiar, se ven obligadas a depender de otra persona.

En función de lo expuesto, el problema a tratar es la dificultad en el manejo de aplicativos para establecer interacciones entre usuarios con discapacidad visual y la sociedad.

1.2. Antecedentes Investigativos

Una vez analizados varios documentos provenientes de fuentes de investigación de universidades en Ecuador y artículos científicos internacionales, se han recopilado diversas obras en las cuales se ha recolectado información que sirve de respaldo para el presente proyecto.

Cristian Patricio Guachamin Ayala [3] afirma que “Existen tecnologías que se pueden adaptar para el segmento de personas con algunas discapacidad que no pueden utilizarlas de manera normal, como lo son aplicaciones móviles que pueden modificarse siendo

incluyentes para grupos de personas que en ocasiones no son tomadas en cuenta para el lanzamiento de un producto tecnológico". Esto destaca la necesidad de más aportes por parte de desarrolladores para mejorar aplicativos existentes o desarrollar nuevos destinados a personas con discapacidad.

Según Betty Maricel Armijo Moreta [4] indica que "las TIC han producido muchos cambios en el entorno de todas las personas que las utilizan, por lo que se cree que las TIC son herramientas idóneas para conseguir el acceso a la educación, sin dejar de lado a las personas con discapacidad visual, teniendo en cuenta que es un grupo social condicionado por diferentes barreras, resultando uno de los grupos que habitualmente resultan invisibles en las indagaciones sobre comunicación y nuevas tecnologías permitiendo que con las TIC creen un vehículo para superar las distintas barreras de aprendizaje a las que se enfrentan diariamente, esto implica que puedan utilizar de forma eficaz con independencia de sus limitaciones personales o derivadas de su contexto de uso". Se determino que las personas con discapacidad visual se motivan por el uso de herramientas que involucren tecnologías.

En el artículo desarrollado por Suraj Singh Senjam [5] afirma que "los estudiantes con discapacidad visual requieren una variedad de tecnología de asistencia (TA) para sus actividades de la vida diaria. Estas TA ayudan a mejorar la vida independiente, mejoran su productividad en los trabajos de rutina. Las TA para las actividades de la vida diaria están disponibles. Algunos ejemplos son el sensor de nivel de líquido, el detector de color parlante, el reloj o despertador parlante y el identificador de dinero parlante". Cada nueva herramienta desarrollada para facilitar las labores de las personas con discapacidad visual les brinda la oportunidad de sentirse útiles en la sociedad.

La investigación desarrollada por Gioconda Lorena Chicaiza Romero [6] afirma que "la convivencia es importante para los seres humanos, en donde, se convierte en una necesidad de la cual las personas no pueden estar solas en un mundo llena de vida, es decir, deben vivir en compañía y conseguir una manera apropiada y pacífica para

convivir". Esto refleja la necesidad de una interacción constante para que las personas con discapacidad visual no desarrollen personalidades con baja autoestima.

Geesela del Rosario Alban Mollocana [7], en su tesis, utilizó una placa Raspberry Pi y tecnología móvil con el software APP Inventor para Android Studio en el desarrollo del sistema. Al concluir la investigación, se demuestra que el manejo por voz del aplicativo ofrece un elevado porcentaje de efectividad, alcanzando un 91.77% al implementar adicionalmente un micrófono USB EH mini. Se evidencia la efectividad del uso de comandos de voz para la realización de actividades en el hogar, y se sugiere que estas referencias pueden aplicarse al ámbito móvil para llevar a cabo operaciones de envío y recepción de mensajes.

Jennifer Daniela Cadena Oyasa y Johnny Ismael Merino Sangoluisa [8], en el desarrollo del aplicativo, utilizaron el framework Ionic, que posibilita su responsividad y adaptación a cualquier tamaño de pantalla. Además, implementaron una función de traducción de voz a texto. Al culminar este proyecto, se determina que la API VOSK es más efectiva para transcribir voz a texto en comparación con la API Google Cloud.

Según Acrapol Nimmolrat, Pattaraporn Khuwuthyakorn, Purida Wientong y Orawit Thinnukoo [9], afirman que la aplicación móvil desarrollada con base en el uso de tecnología inteligente desempeñará un papel importante en el apoyo a las personas con discapacidad visual en Tailandia, mejorando así la eficacia del autocuidado. Se demuestra que es posible desarrollar aplicativos de acuerdo con las necesidades de personas con discapacidad visual para facilitar las actividades de su vida diaria.

1.3. Fundamentación Teórica

Ingeniería de software

La Ingeniería de Software representa una disciplina que proporciona métodos y técnicas. La razón fundamental para contar con un proceso de desarrollo es mejorar la seguridad

laboral, eliminando riesgos innecesarios y logrando y manteniendo la calidad en el producto a desarrollar. Esto se logra utilizando la esencia fundamental del modelo, que establece que el prototipo debe construirse en poco tiempo y sin utilizar muchos recursos. [10].

Metodologías de desarrollo de software

- **Agile**

Las metodologías ágiles se fundamentan en un conjunto de principios resumidos en el Manifiesto Ágil, que se centran en otorgar prioridad a los elementos que aportan mayor valor durante el desarrollo del proyecto. Estos principios incluyen dar importancia al individuo y la interacción del equipo sobre el proceso y las herramientas; al software funcional sobre la documentación; a la colaboración con y del cliente sobre la negociación del contrato; y a la respuesta a los cambios sobre el seguimiento de un plan [11].

.

- **Scrum**

Es un sistema iterativo que organiza el desarrollo en ciclos de trabajo denominados Sprints. Cada iteración tiene una duración que no supera un mes y se basa en la motivación del trabajo en equipo y en una respuesta ágil al cambio, con miras al logro del resultado final. Es especialmente apropiado para proyectos con requisitos cambiantes de manera rápida y con incertidumbre asociada. Esta metodología, ampliamente adoptada en numerosas empresas y corporaciones dedicadas al desarrollo de software o actividades relacionadas con la tecnología de la información, es actualmente el marco de trabajo más popular en la gestión de proyectos ágiles. Se busca realizar una aproximación a sus características y a las ventajas que representa para las organizaciones, tanto en términos de colaboración entre los miembros del equipo como en la dirección y ejecución efectiva de los planes de trabajo. [11].

• Extreme programming

La metodología de Programación Extrema (XP) se clasifica como ágil, ya que posibilita la priorización en el desarrollo, adaptando los 13 procesos al equipo de desarrollo. Kent Beck introdujo esta metodología por primera vez en el libro "Extreme Programming Explained: Embrace Change," publicado en 1999. [12].

Herramientas de Desarrollo de Software

En una era tan sofisticada, cada día se generan aplicativos y software que facilitan el trabajo del desarrollador, mientras que otros quedan obsoletos. Así, estas son las tres herramientas actualmente comúnmente utilizadas por el programador.

• Slack

Slack Technologies Inc. es una empresa de comunicaciones que se presenta como una aplicación de mensajería, ofreciendo una alternativa al correo electrónico, con el objetivo de facilitar el acceso a la información dentro de las empresas que implementan la aplicación. Este producto de software conecta de manera flexible y abarca a todos los empleados, tanto dentro como fuera de la compañía, a través de los canales de mensajería establecidos, los cuales son accesibles desde cualquier ubicación y resultan útiles, especialmente en situaciones de teletrabajo. En estos canales, los usuarios pueden compartir mensajes y archivos en tiempo real gracias a la integración con otras aplicaciones. Slack ofrece varios planes, como Gratuito, Pro, Businesss+, Enterprise Grid, adaptados a las necesidades de los clientes, con el objetivo de aumentar la productividad en las comunicaciones de los empleados de las empresas que lo utilizan. [13].

• Jira

Su aplicación principal es el rastreo de errores de software; sin embargo, debido a sus funciones avanzadas de personalización, también es apropiado para otros sistemas de emisión de tickets, como órdenes de trabajo, mesas de ayuda, entre otros, así como para la gestión de proyectos. [14].

• Git

Git es un sistema de control de versiones distribuido que habilita a todos los usuarios para trabajar simultáneamente en un proyecto, realizando "capturas" del trabajo individual para luego fusionarlo. Otras opciones de control de versiones distribuidas, como Mercurial o Bazaar, son comparables a Git, pero Git es, con diferencia, el más empleado. [15].

Aplicación móvil

Una aplicación móvil se define como un programa informático que opera en un dispositivo móvil, como teléfonos y tabletas, desempeñando diversas funciones para el usuario. En los últimos años, las aplicaciones móviles han sido uno de los sectores del marketing móvil que ha experimentado un crecimiento significativo [16].

Sistema operativo android

Android representa una plataforma de software de código abierto construida sobre Linux y diseñada para una diversidad de dispositivos y formatos. En la actualidad, se encuentra en su decimotercera versión, la cual incorpora medidas más efectivas de protección de la privacidad para los usuarios. [17].

Inclusión social

Se hace referencia a la ejecución de los derechos, la involucración en la vida social, la entrada a la educación, salud y cuidado, así como a los servicios esenciales de infraestructura y a la vivienda, junto con la obtención de ingresos. Este término alude a un proceso de mejora de las condiciones económicas, sociales, culturales y políticas con el fin de permitir la participación completa de las personas en la sociedad [18].

Habilidades sociales

Las habilidades sociales abarcan aspectos relacionados como la autoestima, la asertividad y la inteligencia emocional. La autoestima se refiere a la satisfacción que una persona experimenta consigo misma y con su vida. La asertividad se caracteriza por una actitud

de afirmación y defensa de los derechos personales, involucrando la expresión adecuada de deseos, opiniones, sentimientos y necesidades, siempre respetando los derechos de los demás. [19].

Tipos de discapacidad

• Discapacidad visual

La discapacidad visual se refiere a la pérdida de la capacidad para ver, ya sea de manera parcial o total. Aquellas personas con afectación parcial o total de su visión no pueden percibir la luz, el color, la forma o el tamaño de los objetos. [20].

• Discapacidad física

La discapacidad física se caracteriza por afectar al sistema locomotor, especialmente a las extremidades, generando una limitación debido al estado físico de la persona, impidiéndole moverse con plena funcionalidad de su sistema motriz de manera permanente e irreversible. La mayoría de las personas con discapacidad física (alrededor del 80%) adquieren esta condición después del nacimiento, ya sea por condiciones comunes o algún accidente, mientras que un porcentaje menor se origina durante el embarazo. Esta discapacidad puede manifestarse a través de la disminución o ausencia de coordinación del movimiento, derivada de trastornos en el tono muscular o en el equilibrio. [21].

• Discapacidad intelectual

La discapacidad intelectual es un término utilizado para describir la situación en la que una persona enfrenta dificultades para aprender al nivel esperado y desempeñarse de manera normal en su vida cotidiana. En niños, la intensidad de la discapacidad intelectual varía significativamente, abarcando desde problemas leves hasta dificultades graves. Los niños con discapacidades del desarrollo pueden encontrar desafíos al comunicar sus deseos o necesidades, además de tener dificultades en el autocuidado. Estas discapacidades pueden resultar en un aprendizaje y desarrollo más lento en comparación con otros niños de la misma edad. Es posible que estos niños requieran más tiempo para

adquirir habilidades como hablar, caminar, vestirse o alimentarse de manera independiente, y también pueden enfrentar obstáculos en su proceso de aprendizaje escolar. [22].

Comunicación de personas con discapacidad visual

Todas las personas, independientemente de su condición social o física, tienen el derecho a la comunicación. En este contexto, han surgido herramientas como asistentes personales con inteligencia artificial que facilitan el acceso a la mensajería y las llamadas para aquellas personas que no pueden utilizar aplicaciones de la misma manera que aquellas sin necesidades especiales. Sin embargo, debido a que constituyen un grupo minoritario, a menudo no se les concede la importancia necesaria para desarrollar y mejorar estas herramientas, lo que limita su capacidad de intercomunicarse con facilidad.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar una aplicación en Android optimizando la usabilidad para facilitar la comunicación de personas con discapacidad visual en el periodo Abril – Septiembre del 2023 en el cantón Ambato.

1.4.2. Objetivos Específicos

- Analizar las necesidades y dificultades que poseen las personas con discapacidad visual para comunicarse en dispositivos móviles.
- Identificar las librerías Android que se deben emplear en el aplicativo móvil que permita cumplir la función de manejo por voz.
- Implementar la aplicación móvil para personas con discapacidad visual para comunicarse y recopilar datos de usabilidad en su ejecución.

CAPÍTULO II.- METODOLOGÍA

2.1. Materiales

En el marco de este proyecto y dentro del entorno social en el que se llevó a cabo, se realizó la elaboración de encuestas dirigidas a personas con discapacidad visual en diversos lugares del cantón Ambato, con la colaboración de familiares que actuaron como terceros para la recopilación de información.

Encuesta aplicada a personas con discapacidad visual en el cantón Ambato

El propósito de la encuesta es identificar las posibles dificultades que una persona con discapacidad visual podría enfrentar al utilizar un dispositivo móvil. También busca determinar si utilizan alguna aplicación especializada para la comunicación con otras personas y conocer qué características existentes manejan con facilidad o les resultan complicadas. El formulario consta de 23 preguntas, que abarcan opciones dicotómicas, de elección múltiple y preguntas abiertas.

CUESTIONARIO:

1. ¿Cómo describiría su discapacidad visual?

Ceguera Parcial ( ) Ceguera Total ( )

2. ¿Utiliza algún dispositivo móvil inteligente?

Si ( ) No( )

3. ¿Cuál es el grado de complejidad al manejar aplicativos móviles?

Muy difícil ( ) Difícil ( ) Neutral ( ) Fácil ( ) Muy fácil ( )

4. ¿Con que frecuencia utiliza aplicativos móviles?

☐ A. Por lo menos una vez al día
☐ B. Por lo menos una vez a la semana
☐ C. Por lo menos una vez al mes
☐ D. Nunca

5. ¿Qué aplicación móvil utiliza con mayor frecuencia?

☐ A. Juegos
☐ B. Redes sociales
☐ C. Productividad
☐ D. Entretenimiento (películas, música, etc.)
☐ E. Ninguno

6. ¿Cómo calificaría su experiencia de uso con los aplicativos móviles que frecuenta?

A. Excelente ( ) B. Buena ( ) C. Regular ( ) D. Mala ( ) E. Muy mala ( )

7. ¿Conoce una aplicación de mensajería especializado para personas con discapacidad visual?

Si ( ) No ( )
En caso de si, ¿Cuál?:_____

8. ¿Estaría dispuesto/a adquirir un aplicativo móvil que cubra sus necesidades y expectativas para la comunicación?

Si ( ) No ( )

9. ¿Qué tipo de aplicativo móvil le gustaría que se desarrollara para personas con discapacidad visual?

☐ A. Juegos
☐ B. Redes sociales y Comunicación
☐ C. Productividad
☐ D. Entretenimiento (películas, música, etc.)

10. ¿Tiene habilitado el asistente de voz en su dispositivo móvil?

Si ( ) No ( )

11. ¿Cree usted que el uso de aplicaciones de mensajería mejoría las capacidades de interactuar con la sociedad?

Si ( ) No ( )

12. ¿Presenta dificultades para comunicarse con sus familiares a través de dispositivo móviles?

Si ( ) No( )
En caso de si, ¿Cuál?:_____

13. ¿Ha utilizado algún aplicativo o software por manejo de voz para comunicarse?

Si ( ) No ( )
En caso de si, ¿Cuál?:_____

14. Necesita ayuda de terceros para manejar un aplicativo móvil o smartphone para comunicarse?

Si ( ) No ( )

15. ¿Tiene activado la función de lector de pantalla Talkback en su dispositivo móvil?

Si ( ) No( )
¿Para qué?:_____

16. ¿Considera necesario una guía de navegación por voz dentro de una aplicación móvil de comunicación?

Si ( ) No ( )

17. ¿Necesita instrucciones por voz para manejar aplicaciones móviles?

Si ( ) No ( )

18. ¿Maneja con agilidad la función de lector de pantalla Talkback del dispositivo móvil?

Si ( ) No ( )

19. ¿Posee constantemente acceso a internet en sus dispositivos?

Nunca ( ) Casi Nunca ( ) Ocasionalmente ( ) Casi todos los días ( ) Todos los días( )

20. ¿Las aplicaciones móviles que utiliza posee características de accesibilidad para personas con discapacidad visual?

Si ( ) No ( )

En caso de si, ¿Cuál?:_____

21. ¿Utiliza o conoce de algún aplicativo para comunicarse con otras personas por mensajes de texto?

_____

22. ¿Qué dificultades a encontrado al utilizar aplicativos comunes para comunicarse con otras personas?

_____

23. ¿Qué funcionalidad o facilidades de opciones le gustaría que se integraran en un nuevo aplicativo de mensajería para comunicarse?

_____

2.2. Métodos

El trabajo de investigación se contextualiza en la modalidad de investigación bibliográfica y de campo.

2.2.1. Modalidad de la Investigación

Investigación de Campo

Tras la investigación de las dificultades experimentadas por individuos con discapacidad visual en la comunicación, se efectuó una visita a sus lugares de residencia con el propósito de comprender directamente la problemática. Posteriormente, se realizaron

ajustes y se desarrollaron características en la aplicación móvil para mejorar su usabilidad por parte de los usuarios.

Investigación Bibliográfica-Documental

El trabajo investigativo se fundamentó en una revisión bibliográfica, a través de la cual se recopiló valiosa información proveniente de diversas fuentes, como investigaciones previas, tesis y revistas especializadas. Esta revisión se centró en la variable de comunicación de personas con discapacidad visual, con el fin de obtener un panorama completo y actualizado sobre el tema.

2.2.2. Población y muestra

El proyecto se centra en el grupo de individuos con discapacidad visual, abarcando una franja de edades de 7 a 64 años en el cantón Ambato. Esta selección se fundamenta en los datos recopilados sobre el número de personas con discapacidad visual disponibles en la página del CONADIS, que señala una población total de 606 personas en esta categoría. Dado que la población supera los 100 individuos, se aplicaron una serie de fórmulas para obtener una muestra significativa.

Tabla 1.Población de estudio

Población	**Número**	**Porcentaje**
Personas con discapacidad visual (7 a 64 años)	61	100%
Total	**61**	**100%**

$$n = \frac{N * Z^2 * p * q}{(N-1)e^2 + Z^2 + p + q}$$

Donde:

n = Tamaño de la muestra

N = Población: 606

Z = Nivel de Confianza 90% Equivalente a 1.65

σ = Desviación estándar constante: 0.5

e = Margen de error: 0.1

$$x = \frac{606 * 0.5^2 * 1.65^2}{605 * 0.1^2 + 0.5^5 * 1.65^2}$$

$$n = 61.28$$

Por lo que la muestra de personas con discapacidad visual se redondeara a 61 encuestados.

2.2.3. Recolección de la Información

Para la recolección de información, se utilizaron formularios físicos, ya que resulta difícil o imposible para las personas con discapacidad visual responder a preguntas en un formulario en línea. Por lo tanto, se llevó a cabo de manera personal, con la asistencia de un tercero.

Resultados de la encuesta realizada a Personas con discapacidad visual

La encuesta fue aplicada a un total de 61 personas con discapacidad visual en el cantón Ambato.

Pregunta 1:

¿Cómo describiría su discapacidad visual?

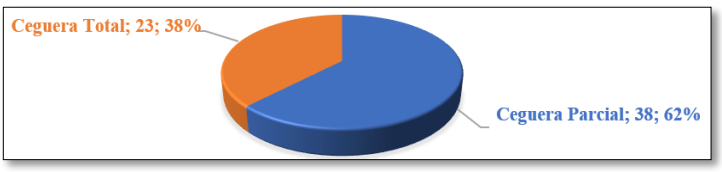


Figura 1. Tabulación de resultados pregunta 1.

Análisis e interpretación de resultados

Según los datos obtenidos y mostrados en la figura 1, el 62% de las personas encuestadas poseen una ceguera parcial, es decir, aún conservan cierta capacidad de visión para poder manejar dispositivos móviles y llevar a cabo actividades con total normalidad, mientras que un 38% de los encuestados presenta una ceguera total, lo que les dificulta el manejo de tecnología o dispositivos no adaptados a sus necesidades, afectando su desenvolvimiento diario. Basándonos en los datos, podemos interpretar que la baja visión es más común en la muestra de personas encuestadas, mientras que la ceguera total afecta a una proporción menor.

Pregunta 2:

¿Utiliza algún dispositivo móvil inteligente?

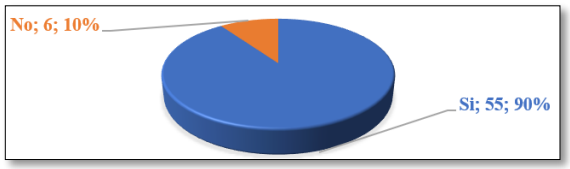


Figura 2. Tabulación de resultados pregunta 2.

Análisis e interpretación de resultados

Según los datos obtenidos y mostrados en la figura 2, el 90% de las personas encuestadas utilizan un dispositivo móvil, lo que les permite desarrollarse en diversos ámbitos con este recurso tecnológico, mientras que el 10% de los encuestados no utiliza dispositivos móviles. Como indican los datos, el uso de dispositivos móviles inteligentes es muy común entre las personas encuestadas, mientras que un pequeño porcentaje opta por no usar un dispositivo móvil inteligente o no tiene acceso a uno.

Pregunta 3:

¿Cuál es el grado de complejidad al manejar aplicativos móviles?

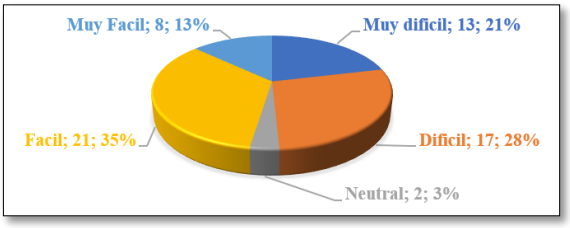


Figura 3. Tabulación de resultados pregunta 3.

Análisis e interpretación de resultados

Según los resultados de la Figura 3, el 21% de los encuestados considera muy difícil el manejo de aplicativos móviles; en este grupo se incluyen personas que no tienen acceso a un dispositivo móvil. El 28% considera que es difícil, el 3% tiene una opinión neutral sobre la complejidad de manejar aplicativos, el 35% de las personas encuestadas encuentra fácil el manejo de aplicativos, mientras que el 13% restante considera muy fácil la complejidad de uso. En relación con las respuestas de los encuestados, la facilidad o dificultad con la que manejan los aplicativos puede estar relacionada con su nivel de ceguera o los años de experiencia en el manejo de dispositivos móviles. Si se manejara con mayor regularidad, sería más fácil para todos y no solo para un cierto porcentaje de personas.

Pregunta 4:

¿Con que frecuencia utiliza aplicativos móviles?

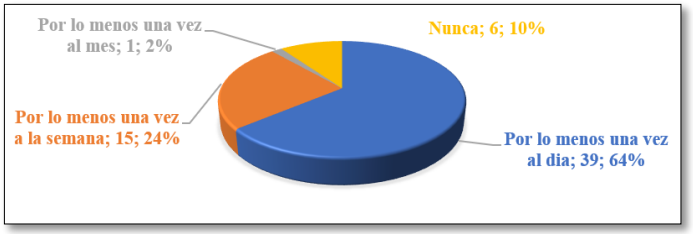


Figura 4. Tabulación de resultados pregunta 4.

Análisis e interpretación de resultados

Según los resultados de la Figura 4, el 64% de los encuestados utilizan aplicativos al menos una vez al día, un 24% los utiliza al menos una vez a la semana, un 10% nunca utiliza aplicativos, mientras que un 2% los usa al menos una vez al mes. Más de la mitad de los encuestados maneja con regularidad sus dispositivos móviles, por lo que es común que los utilicen más de una vez al día. Otros encuestados varían su uso en situaciones limitadas a la semana o al mes, y solo un pequeño porcentaje nunca utiliza un dispositivo móvil, lo que destaca la falta de acceso a estos en ese grupo.

Pregunta 5:

¿Qué aplicación móvil utiliza con mayor frecuencia?

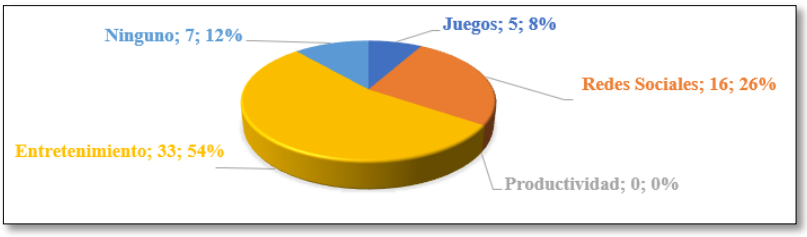


Figura 5. Tabulación de resultados pregunta 5.

Análisis e interpretación de resultados

Según los resultados de la Figura 5, el 54% de los encuestados utiliza aplicativos de entretenimiento, como ver videos, escuchar música o podcasts, el 26% utiliza aplicaciones de redes sociales, lo que indica una adaptación a los servicios disponibles actualmente. Además, el 12% no utiliza ningún tipo de aplicativo, probablemente debido a tener ceguera total o falta de acceso a dispositivos móviles, y un 5% utiliza aplicativos de tipo juegos. Se observa que más de la mitad de los encuestados suele utilizar sus dispositivos inteligentes principalmente para fines de entretenimiento, mientras que un porcentaje significativo también los utiliza para comunicarse, lo que implica que están familiarizados con los servicios de mensajería.

Pregunta 6:

¿Cómo calificaría su experiencia de uso con los aplicativos móviles que frecuenta?

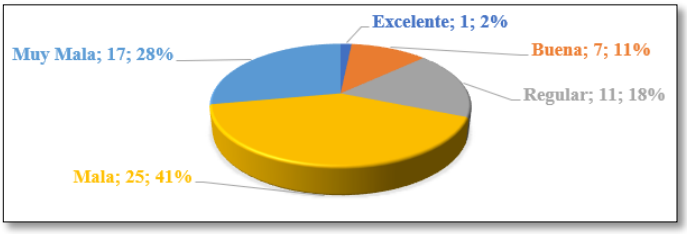


Figura 6. Tabulación de resultados pregunta 6.

Análisis e interpretación de resultados

Según los resultados de la Figura 6, el 41% de las personas encuestadas considera que su experiencia de uso es mala, el 28% la clasifica como muy mala, lo que lleva a que abandonen el uso de aplicativos móviles. Además, el 18% indica una experiencia regular, el 11% la considera buena y, finalmente, el 1% la califica como excelente. Basándonos en los datos, más de la mitad de los encuestados informa que no tiene una experiencia satisfactoria al utilizar aplicaciones móviles, ya que estas no están adaptadas a sus necesidades. Por otro lado, el porcentaje restante tiene una experiencia que va desde regular hasta buena, demostrando una capacidad de adaptación por sí mismos a las aplicaciones.

Pregunta 7:

¿Conoce una aplicación de mensajería especializado para personas con discapacidad visual?

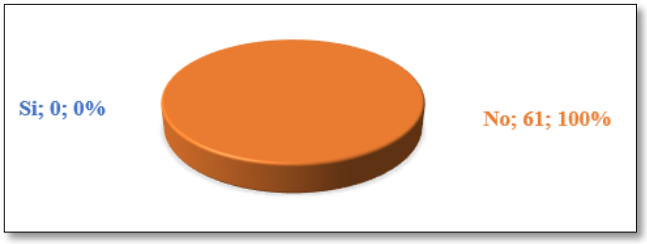


Figura 7. Tabulación de resultados pregunta 7.

Análisis e interpretación de resultados

Según los resultados de la Figura 7, el 100% de las personas encuestadas desconoce la existencia de algún aplicativo especializado para comunicarse que cumpla las necesidades de las personas con discapacidad visual. Esto resalta la poca importancia por parte de los desarrolladores en la creación de aplicativos con accesibilidad adecuada para personas con necesidades especiales.

Pregunta 8:

¿Estaría dispuesto/a adquirir un aplicativo móvil que cubra sus necesidades y expectativas para la comunicación?

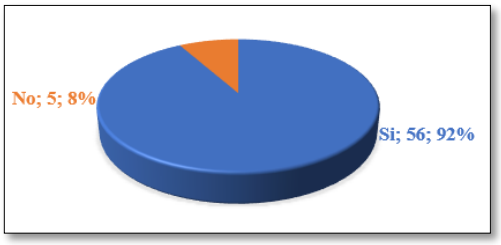


Figura 8. Tabulación de resultados pregunta 8.

Análisis e interpretación de resultados

Según los resultados de la Figura 8, el 92% de los encuestados estaría dispuesto a adquirir un producto que les brinde la facilidad de comunicarse, mientras que el 8% no está dispuesto a hacerlo. Estos resultados indican que la gran mayoría de las personas encuestadas poseen un interés considerable en adquirir un aplicativo que les brinde soluciones de comunicación acordes a sus necesidades y expectativas, mientras que un porcentaje menor no optaría por adquirirlo, ya que no lo ven necesario para satisfacer sus necesidades.

Pregunta 9:

¿Qué tipo de aplicativo móvil le gustaría que se desarrollara para personas con discapacidad visual?

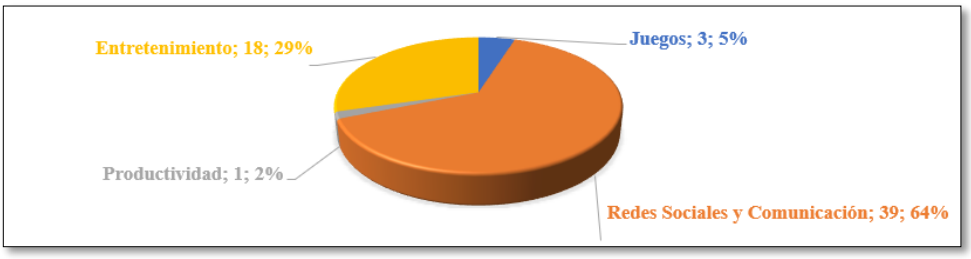


Figura 9. Tabulación de resultados pregunta 9.

Análisis e interpretación de resultados

Según los resultados de la Figura 9, el 64% de los encuestados desearía que se desarrollen más aplicativos que les permitan comunicarse, un 29% se encuentra interesado en que se desarrollen aplicativos para su entretenimiento, como aquellos para reproducir música o escuchar audiolibros, un 5% desea que se desarrollen juegos que de alguna manera ayuden a desarrollar las capacidades de los usuarios, y un 2% desea el desarrollo de aplicativos de productividad. Un porcentaje mayor a la mitad desearía que se desarrollen aplicativos móviles enfocados en redes sociales y comunicación. Esto indica que existe una necesidad y un interés por parte de estas personas en conectarse y comunicarse con otras a través de aplicativos accesibles y adaptados a sus necesidades. También, un porcentaje de personas desean opciones de entretenimiento y juegos con preferencias a este grupo de personas.

Pregunta 10:

¿Tiene habilitado el asistente de voz en su dispositivo móvil?

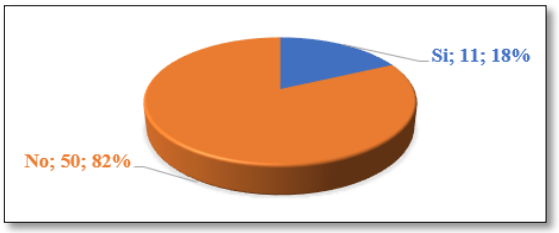


Figura 10. Tabulación de resultados pregunta 10.

Análisis e interpretación de resultados

Según los resultados de la ilustración Figura 10, el 82% de los encuestados no tiene activado el asistente de voz de su celular, por lo que no están acostumbrados a este tipo de ayuda que se puede brindar a través de comandos de voz para ciertas actividades básicas, como realizar una llamada o colocar un recordatorio. Solo el 18% tiene activado el asistente de voz, por lo que son pocas las personas familiarizadas con el manejo de este servicio. De acuerdo con los datos recopilados, la mayoría de los encuestados desconoce los servicios disponibles que pueden facilitar el uso de aplicaciones y realizar funciones para personas con discapacidad visual. Solo un pequeño porcentaje de los encuestados está al tanto de las ventajas que se pueden obtener al activar esta función.

Pregunta 11:

¿Cree usted que el uso de aplicaciones de mensajería mejoraría las capacidades de interactuar con la sociedad?

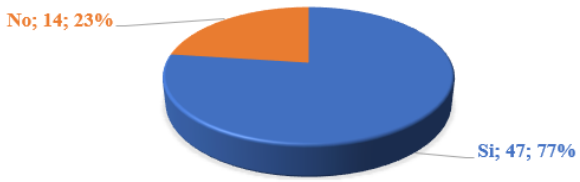


Figura 11. Tabulación de resultados pregunta 11.

Análisis e interpretación de resultados

Según los resultados de la ilustración Figura 11, el 77% de los encuestados considera que tener la oportunidad de utilizar una aplicación de mensajería con otras personas les permitirá mejorar sus capacidades de interacción e integración en la sociedad, mientras que el 23% considera que no. Estos resultados denotan que la mayoría de los encuestados ven el potencial de las aplicaciones de mensajería para facilitar la comunicación y la participación social a pesar de sus limitaciones visuales, y un porcentaje menor de personas cree que no mejorará, esto puede deberse a diferentes razones como la

preferencia de otros medios de comunicación o la percepción de barreras adicionales en el uso de estas aplicaciones.

Pregunta 12:

¿Presenta dificultades para comunicarse con sus familiares a través de dispositivos móviles?

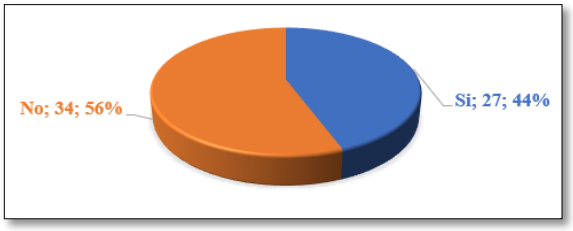


Figura 12. Tabulación de resultados pregunta 12.

Análisis e interpretación de resultados

Según los resultados de la Figura 12, el 56% de los encuestados no presentan dificultades para comunicarse con sus familiares, mientras que el 44% sí presenta dificultades para comunicarse. La mayoría de las personas con discapacidad encuestadas han encontrado formas efectivas de utilizar los dispositivos móviles para mantener la comunicación con sus seres queridos, a pesar de las limitaciones visuales. Por otro lado, un porcentaje menor presenta dificultades de comunicación, lo que puede deberse a varios factores, como la falta de accesibilidad en las aplicaciones utilizadas, dificultades técnicas o falta de conocimiento sobre cómo utilizar adecuadamente los dispositivos y las aplicaciones

Pregunta 13:

¿Ha utilizado algún aplicativo o software por manejo de voz para comunicarse?

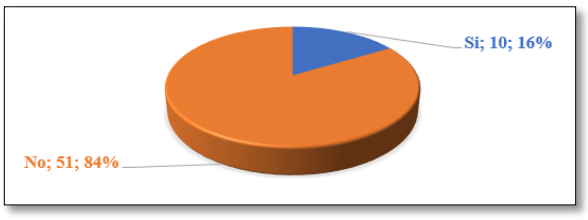


Figura 13. Tabulación de resultados pregunta 13.

Análisis e interpretación de resultados

Según los resultados de la Figura 13, el 84% no ha utilizado algún software o aplicativo que se maneje por comandos de voz, mientras que el 16% sí ha utilizado algún software o aplicativo. La mayoría de las personas encuestadas no han explorado o utilizado estas herramientas tecnológicas específicas para la comunicación, y un porcentaje menor ha encontrado beneficios y ha adoptado el manejo de estas soluciones tecnológicas para mejorar su comunicación, encontrando mayor autonomía con asistentes de voz como Siri, Cortana, Alexa, e incluso el asistente de Google integrado en Android.

Pregunta 14:

¿Necesita ayuda de terceros para manejar un aplicativo móvil o Smathphone para comunicarse?

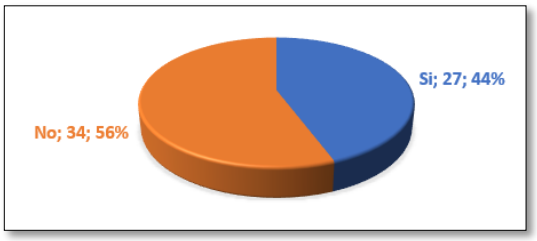


Figura 14. Tabulación de resultados pregunta 14.

Análisis e interpretación de resultados

Según los datos obtenidos y mostrados en la figura 14, el 56% de las personas encuestadas pueden utilizar sus dispositivos para comunicarse sin necesidad de apoyo, mientras que el 44% necesita ayuda de otros para poder manejar un smartphone. Más de la mitad de los encuestados no necesitan ayuda de terceros, lo que indica que tienen la capacidad de utilizar y navegar por las aplicaciones móviles de forma autónoma, a pesar de las limitaciones visuales. El porcentaje restante no posee esta capacidad debido a diferentes factores, como la complejidad de las aplicaciones, falta de accesibilidad o la necesidad de asistencia para configurar o utilizar funciones específicas.

Pregunta 15:

¿Tiene activado la función de lector de pantalla TalkBack en su dispositivo móvil?

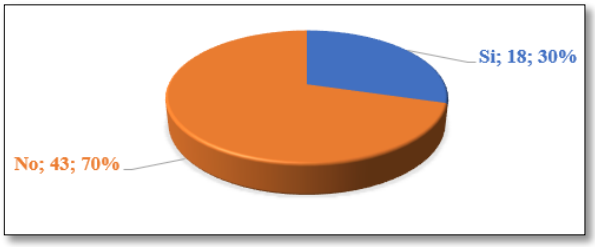


Figura 15. Tabulación de resultados pregunta 15.

Análisis e interpretación de resultados

Según los datos obtenidos y mostrados en la figura 15, el 70% de las personas no tienen activado el sistema TalkBack en sus dispositivos móviles. Esto se debe a que pueden manejarlos sin mucha dificultad al poseer un poco de visión. El 30% sí tiene activado TalkBack para poder desplazarse en la pantalla del smartphone y realizar diferentes actividades, como comunicarse o entretenerse con otras aplicaciones. Aquellas personas que tienen activada esta función la utilizan mayoritariamente para desplazarse por los elementos de la pantalla

Pregunta 16:

¿Considera necesario una guía de navegación por voz dentro de una aplicación móvil de comunicación?

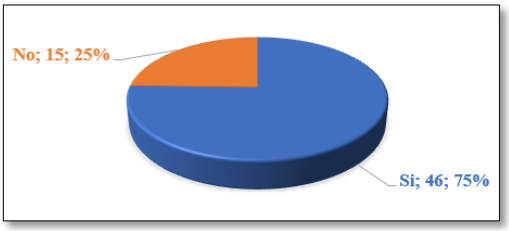


Figura 16. Tabulación de resultados pregunta 16.

Análisis e interpretación de resultados

Según los datos obtenidos y mostrados en la figura 16, el 75% considera necesario que un aplicativo cuente con una guía de navegación como ayuda adicional para personas con

discapacidad visual, mientras que el 25% no considera necesaria esta opción. La mayoría de las personas encuestadas ven beneficios en contar con una función de navegación guiada por voz, lo cual puede facilitar su experiencia de uso y mejorar su accesibilidad. El porcentaje menor de personas no lo considera necesario, ya que pueden preferir otros métodos de navegación o suelen sentirse cómodos utilizando las funciones existentes de las aplicaciones.

Pregunta 17:

¿Necesita de instrucciones por voz para manejar aplicaciones móviles?

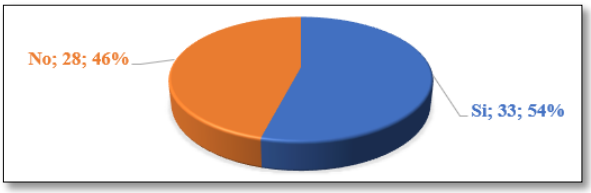


Figura 17. Tabulación de resultados pregunta 17.

Análisis e interpretación de resultados

Según los resultados de la Figura 17, el 54% de las personas encuestadas necesitan ciertas instrucciones por voz, mientras que el 46% no las requieren. El hecho de que el porcentaje mayor de personas necesite estas instrucciones sugiere que valoran y encuentran beneficios en recibir indicaciones verbales que les guíen en el uso de las aplicaciones, mejorando así su autonomía y accesibilidad. El porcentaje menor no las necesita debido a su familiaridad con el uso de aplicaciones o a sus preferencias personales.

Pregunta 18:

¿Maneja con agilidad la función de lector de pantalla TalkBack del dispositivo móvil?

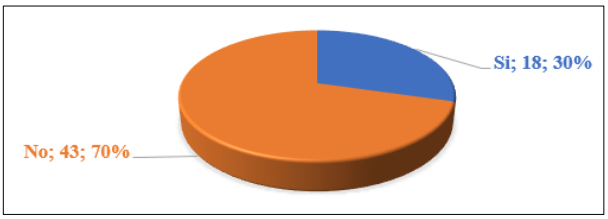


Figura 18. Tabulación de resultados pregunta 18.

Análisis e interpretación de resultados

Según los resultados de la Figura 18, el 70% de las personas encuestadas no manejan el sistema de accesibilidad y asistencia TalkBack, mientras que solo el 30% afirma manejarlo con agilidad para poder realizar diferentes tareas en un celular. La mayoría de las personas no manejan con agilidad esta función debido a que presentan dificultades en su utilización o pueden requerir más práctica y familiarización con su funcionamiento. La cantidad menor de personas encuestadas maneja con agilidad esta función de accesibilidad, lo cual facilita su interacción con el dispositivo móvil y las aplicaciones

Pregunta 19:

¿Posee contantemente acceso a internet en sus dispositivos?

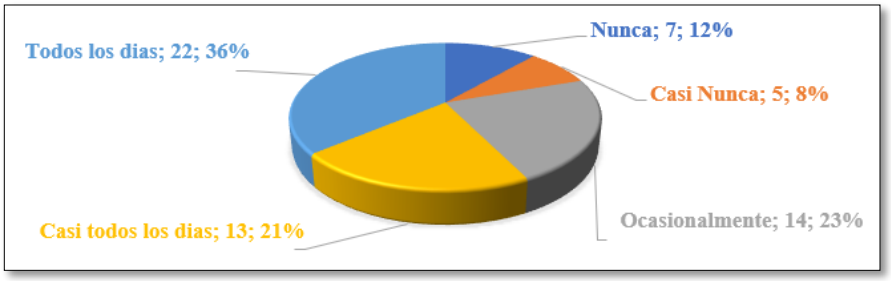


Figura 19. Tabulación de resultados pregunta 19.

Análisis e interpretación de resultados

Según los resultados de la Figura 19, el 36% de las personas encuestadas tienen acceso a Internet en sus dispositivos todos los días, mientras que el 23% lo tiene ocasionalmente.

El 21% de los encuestados posee acceso a Internet casi todos los días, un 12% nunca tiene acceso a Internet y un 8% casi nunca tiene acceso a Internet.

La mayoría de las personas con discapacidad visual tienen una conexión constante a Internet, lo cual les brinda la oportunidad de acceder a información, servicios en línea y comunicación de manera regular. Un porcentaje considerable enfrenta limitaciones en su conectividad y puede depender de la disponibilidad de conexiones Wi-Fi o redes móviles en su entorno. El porcentaje menor experimenta limitaciones significativas en su capacidad para acceder a Internet o simplemente no tiene acceso debido a situaciones socioeconómicas

Pregunta 20:

¿Las aplicaciones móviles que utiliza posee características de accesibilidad para personas con discapacidad visual?

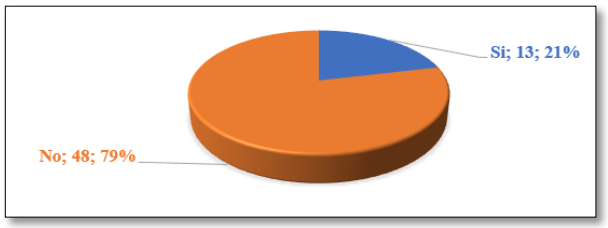


Figura 20. Tabulación de resultados pregunta 20.

Análisis e interpretación de resultados

Según los resultados de la Figura 20, el 79% de los usuarios encuestados afirma que las aplicaciones que utilizan no poseen características de accesibilidad, mientras que el 21% indica que las aplicaciones que utilizan sí tienen características de accesibilidad. El porcentaje mayor sugiere que las personas encuentran dificultades en el acceso y uso de aplicaciones debido a la falta de adaptación a sus necesidades específicas. En contraste, el porcentaje menor se beneficia de cierta manera de las limitadas características que ofrecen algunas aplicaciones, como adaptar el tamaño del texto, el contraste y los colores.

Estas son varias de las opciones que, aunque limitadas, ofrecen aplicaciones como Messenger, WhatsApp o Telegram.

Pregunta 21:

¿Utiliza o conoce de algún aplicativo para comunicarse con otras personas por mensajes de texto?

- WhatsApp
- Facebook Messenger
- Telegram
- Instagram Direct
- Mensajes de Texto (SMS)

Análisis e interpretación de resultados

Con respecto a los resultados de la pregunta 21 la mayoría de las personas que tienen posibilidades de manejar un dispositivo móvil con un poco de visión utilizan o han utilizado aplicativos de uso común para comunicarse con otras personas, se les permite manejarlo, aunque se le haga un poco complicado o exista pérdida de tiempo en su ejecución a través de aplicativos de terceros que le de lectura de pantalla.

Pregunta 22:

¿Qué dificultades a encontrado al utilizar aplicativos comunes para comunicarse con otras personas?

- El texto que se muestra tiene un límite de ajuste en sus dimensiones
- El tamaño de iconos y elementos de la pantalla no son ajustables
- Se necesita ayuda de terceras personas para el manejo de ciertas aplicaciones.
- No existe información sobre la accesibilidad incorporada o inexistente en aplicaciones.
- Complicación de las acciones que se puedan ejecutar.
- No es compatible con lectores de pantalla.
- Sobreesfuerzo físico al intentar encontrar la función apropiada en el aplicativo.

- Gasto de tiempo excesivo hasta encontrar la opción deseada.
- Mala recepción de ordenes de voz
- Comandos limitados no suficiente para una completa comunicación.
- Mala interpretación de léxico del usuario.
- Para persona con incapacidad visual como daltonismo el correcto color de aplicativo dificulta su manejo porque no se trabajan bien los contrastes.
- Las opciones para la navegación son limitadas con el uso de aplicativos terceros.
- No existe o es limitado la descripción de audio de las opciones que posee un aplicativo.
- Los iconos u opciones no tienen una descripción de su función.
- El soporte técnico para personas con discapacidad visual es nulo o limitado.
- Lectores de pantalla no encuentran todos los elementos de la pantalla
- Los gestos incluidos no son suficientes.
- El idioma de la aplicación desarrollada en otro país no se adapta a la semántica ecuatoriana.

Análisis e interpretación de resultados

Basándonos en los resultados de la pregunta 22, se evidencian las complicaciones con las que los usuarios manejan aplicativos de comunicación comunes. Por ejemplo, la necesidad de la ayuda de un tercero no proporciona autonomía al usuario con discapacidad para poder comunicarse, y, en parte, impide mantener privacidad al tener que compartir información para gestionar la comunicación. Asimismo, existen muchas otras maneras en las que un usuario no puede manejar un aplicativo con facilidad.

Pregunta 23:

¿Qué funcionalidad o facilidades de opciones le gustaría que se integraran en un nuevo aplicativo de mensajería para comunicarse?

- Lector de pantalla integrado en la aplicación
- Facilidad de uso

- Accesible según el nivel de discapacidad del usuario.
- Simplicidad
- Que el esfuerzo por encontrar una opción sea sencillo y rápida.
- El acceso sencillo al aplicativo y sus funciones.
- Brindar información necesaria de las funcionalidades del aplicativo y su alcance
- Interfaz sencilla para el usuario.
- Tolerancia al error de comandos de voz con confirmaciones.
- Adaptabilidad al modelo de smartphone.
- No requiera grandes características técnicas para su instalación.
- Comandos suficientes para su manejo.
- Interprete de voz integrados.
- Distribución de espacio entre opciones y textos.
- Tamaños de elementos de aplicativos adaptables al gusto del usuario.
- Tiempos cortos de respuestas entre operaciones.
- Audios explicativos de las funcionalidades del aplicativos.
- Lenguaje comprensible tanto en texto como en voz.
- Comprensión del idioma y léxico del usuario.

Análisis e interpretación de resultados.

Según los resultados de la pregunta 23, los usuarios requieren de múltiples opciones en un nuevo aplicativo, siendo una de las más importantes la integración de un lector de pantalla. Por lo general, los aplicativos que lo permiten necesitan de una aplicación adicional para contar con esta función.

2.2.4. Procesamiento y Análisis de Datos

Con relación a los datos obtenidos de las encuestas a personas con discapacidad visual se determinó lo siguiente.

- Las personas con ceguera total son quienes poseen las mayores dificultades al manejar aplicaciones móviles y Smathphone.

- La mayoría de las personas entrevistadas tienen acceso a un dispositivo inteligente, por lo que están en constante interacción con la tecnología. Sin embargo, siempre existe un pequeño porcentaje que no tiene acceso a estos recursos.
- Las personas con discapacidad visual suelen comunicarse mayoritariamente por llamadas, ya que resulta más factible con las limitadas facilidades que les brinda el sistema operativo.
- Los aplicativos más utilizados son llamadas de voz y aplicativos para entretenimiento, como escuchar música.
- Encuentran frustrante la experiencia con el uso de dispositivos móviles debido a la falta de facilidades en las aplicaciones para adaptarse a sus necesidades.
- No existe o no conocen de un aplicativo de mensajería que posea características de usabilidad para personas con discapacidad visual.
- Mas de la mitad de los encuestados desea se desarrollen más aplicativos basados en sus necesidades, mayoritariamente para la comunicación y entretenimiento.
- La mayoría de los usuarios desconoce las funcionalidades de un asistente de voz y como este le puede facilitar la función de envío de mensajes.
- Las personas con un mayor nivel de ceguera necesitan de terceros para utilizar dispositivos móviles y aplicaciones, lo que no les permite desarrollar autonomía.
- Solo las personas que tienen ceguera total o avanzada utilizan la función Talkback para movilizarse por las diferentes aplicaciones y funcionalidades de un dispositivo móvil.
- El acceso a internet depende de la situación económica del usuario y sus familiares.

CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

Una vez finalizadas las encuestas a las personas con discapacidad visual en el cantón Ambato sobre las necesidades y dificultades que enfrentan al manejar un dispositivo smartphone para comunicarse, procedemos al análisis de sus respuestas para extraer requisitos y requerimientos que orienten el desarrollo de la aplicación destinada a facilitar la experiencia de comunicación con otras personas.

3.1.1 Análisis de dificultades al manejar aplicativos para comunicarse.

a) Limitaciones en la visualización de textos: El texto mostrado en pantalla generalmente se limita a tamaños pequeños, medianos y grandes, lo que dificulta el ajuste a dimensiones específicas y complica la lectura clara para personas con discapacidad visual.

b) Tamaño no ajustable de iconos y elementos de pantalla: La falta de ajuste en el tamaño de los iconos y elementos de pantalla dificulta la identificación de diversas opciones y funcionalidades disponibles en una aplicación.

c) Necesidad de ayuda de terceros: En ciertos casos, las personas con discapacidad visual requieren la ayuda de terceras personas para manejar algunas aplicaciones, limitando su independencia y privacidad.

d) Falta de información sobre accesibilidad: La ausencia de información necesaria sobre aplicaciones con opciones accesibles para personas con discapacidad visual dificulta la identificación de aquellas que se adecuen a sus necesidades.

e) Complicaciones en la ejecución de acciones: Algunas aplicaciones presentan acciones complicadas que requieren esfuerzo físico adicional por parte de usuarios con discapacidad visual, resultando en una pérdida de tiempo adicional.

f) Incompatibilidad de lectores de pantalla: La falta de compatibilidad de algunas aplicaciones con lectores de pantalla limita el acceso a la información, privando al usuario con discapacidad visual de obtenerla.

g) Mala recepción de órdenes de voz: Las órdenes de voz pueden no ser interpretadas correctamente, generar confusiones o malentendidos, dificultando la comunicación con la aplicación.

h) Comandos limitados: La disponibilidad de comandos limitados dificulta la comunicación completa con la aplicación, ofreciendo una experiencia incompleta para usuarios con necesidades adicionales.

i) Mala interpretación del léxico: La aplicación puede malinterpretar las palabras y frases del usuario, complicando la comunicación y llevándola a ser errónea.

j) Limitaciones en la navegación: La mayoría de las aplicaciones carecen de métodos de navegación alternativos al convencional.

k) Falta de descripción de audio: La falta de descripción de audio de las opciones disponibles en una aplicación dificulta el acceso a información para personas con discapacidad visual.

l) Falta de descripción de iconos u opciones: Cuando no hay información suficiente sobre las funciones de iconos u opciones, se dificulta la identificación de características a las que se desea acceder en una aplicación.

m) Soporte técnico limitado: El soporte técnico para personas con discapacidad visual puede ser nulo o limitado, dificultando la solución de problemas.

n) Gestos insuficientes: Aunque el uso de gestos se ha popularizado para agilizar el manejo y navegación en aplicaciones, su inclusión actual es insuficiente para personas con necesidades especiales.

o) Adaptación de idioma: La adaptación del idioma de la aplicación al español ecuatoriano puede ser insuficiente, complicando la comprensión de la información para personas con discapacidad visual.

3.1.2 Análisis de necesidades y funcionalidades posibles para aplicativo especializado para personas con discapacidad visual.

a) Lector de pantalla integrado: Es esencial que la aplicación cuente con un lector de pantalla integrado que permita la navegación y el acceso a las diferentes opciones y funciones mediante la voz.

b) Facilidad de uso: La aplicación debe ser fácil de usar y entender para el usuario; por lo tanto, debe ser simple, intuitiva y fácil de navegar.

c) Accesible según el nivel de discapacidad del usuario: La aplicación debe ser accesible para diferentes niveles de discapacidad visual, adaptándose a las necesidades específicas de cada usuario.

d) Simplicidad: En cuanto a diseño y funciones, para evitar confusiones y dificultades para el usuario.

e) Fácil acceso a las opciones: La aplicación debe permitir un acceso rápido y sencillo a las diferentes opciones y funciones, evitando que el usuario tenga que hacer un gran esfuerzo para encontrar lo que necesita.

f) Accesibilidad a la aplicación y sus funciones: Debe ser fácilmente accesible para el usuario y permitir el acceso a todas sus funciones sin barreras.

g) Información necesaria de las funcionalidades del aplicativo y su alcance: La aplicación debe brindar información clara y concisa sobre sus funcionalidades y alcance, para que el usuario pueda entenderlas y utilizarlas adecuadamente.

h) Interfaz sencilla: Debe ser fácil de entender, con iconos y botones grandes y fáciles de identificar.

i) Tolerancia al error: Debe ser tolerante a los errores de voz y ofrecer confirmaciones para evitar malentendidos y errores.

j) Adaptable al modelo de smartphone: Debe ser compatible y adaptable a diferentes modelos de smartphone, para que el usuario pueda utilizarla en su dispositivo preferido o al que tenga acceso.

k) No requiere grandes características técnicas para su instalación: La aplicación debe ser fácil de instalar y no requerir grandes características técnicas del dispositivo para su funcionamiento.

l) Comandos suficientes: El aplicativo debe contar con suficientes comandos de voz para permitir el control de todas sus funciones sin necesidad de utilizar la pantalla.

m) Intérprete de voz integrado: La aplicación debe contar con un intérprete de voz para facilitar la comunicación entre el usuario y la aplicación.

n) Distribución de espacio: La distribución de espacio entre las opciones y textos debe ser adecuada para facilitar la navegación y los elementos de la interfaz.

o) Tamaños adaptables: Debe permitir la adaptación de los tamaños de los elementos de la interfaz para ajustarse a las preferencias y necesidades del usuario.

p) Tiempos cortos de respuesta: El aplicativo debe ser rápido y ofrecer tiempos cortos de respuesta entre operaciones para evitar la frustración del usuario en su espera.

q) Audios explicativos: Se deben presentar audios explicativos de sus funcionalidades para que el usuario pueda comprender su uso.

r) Lenguaje comprensible: Se debe utilizar un lenguaje comprensible tanto en texto como en voz, para que el usuario pueda entender y utilizar todas sus funciones.

s) Comprensión del idioma y léxico del usuario: Debe ser capaz de entender las variaciones de acento del usuario.

3.1.3 Análisis de librerías para el uso de comandos de voz.

Tabla 2. Cuadro comparativo de librerías para uso de comandos de voz.

Biblioteca	Características	Ventajas	Desventajas	Costos	Desarrollador
Google Cloud Speech API	-Reconocimiento de voz en tiempo real -Adaptación de voz -Modelos de dominio específicos -Compara claramente la calidad -Reconocimiento de voz preciso	- Alta precisión y calidad -Transcribe tu contenido y ofrece subtítulos precisos -Saca partido a la voz para ofrecer mejores experiencias de usuario -Optimiza tu servicio con información valiosa extraída de las interacciones de los clientes -Amplio soporte de idiomas	- Requiere una cuenta de Google Cloud y puede tener costos asociados -Necesita conexión a internet -Funciona con audios de alta calidad -Limitación de idioma y acentos -Costo	Los nuevos clientes reciben 300 USD en crédito gratis para utilizarlo en Speech-to-Text. Además, todos los clientes disponen de 60 minutos gratis al mes para transcribir y analizar audio, y no se les descuentan de su crédito.	Google
PocketSphinx	- Motor de reconocimiento de voz de código abierto -Software Libre -Soporte Múltiple Plataforma	- Gratuito y de código abierto -Soporte Múltiples Idiomas -Modo de Adaptación acústica -Bajo Consumo de Recursos -Reconocimiento de voz sin conexión	- Precisión puede variar según el modelo de idioma y el ruido ambiental -Mayor dependencia del modelo de lenguaje -Mayor necesidad de recursos de almacenamiento y procesamiento.	Gratuito	Carnegie Mellon University (CMU)
CMU Sphinx	-Biblioteca de reconocimiento de voz de código abierto -Diseño Flexible -Soporte Comercial -Reconocimiento de voz sin conexión -Herramientas adicionales como alineamiento automático de voz y texto	-Gratuito y de código abierto -Soporte para varios idiomas como ingles de EEUU, ingles de Reino Unido, francés, mandarín, alemán, holandés, ruso y capacidad de construir modelos para otros -Flexibilidad en el uso de audios -Bajo consumo de recursos	-Menos actualizada que otras soluciones comerciales -No hay soporte en español -Precisión limitada en enfoque más modernos	Gratuito	Carnegie Mellon University (CMU)

Biblioteca	Características	Ventajas	Desventajas	Costos	Desarrollador
			-Mayor necesidad de ajustes y configuraciones manuales -Requiere recursos de procesamiento de memoria		
Android Speech Recognition	- Utiliza el reconocimiento de voz incorporado en Android -Soporte Multilingüe -Reconocimiento en tiempo real -Integración de comando de voz -Interfaz de usuario intuitiva	- Integrado en Android, no requiere configuración adicional -Interfaz de usuario intuitiva -Interacción manos libres -Facilidad de uso -Amplia disponibilidad en dispositivos -Comunidad activa -Amplia documentación	- Precisión puede variar según el dispositivo y la calidad del micrófono - No está destinada a ser utilizada para el reconocimiento continuo -Dependencia de la conexión a internet -Consumo de recursos -Limitaciones en entorno ruidoso - Menos mantenimiento y soporte en comparación con soluciones comerciales -Menor precisión -Limitación en la cantidad de palabras y frases -Menor rendimiento en entornos ruidosos -Limitada cantidad de recursos y documentación	Gratuito	Google
OpenEars	- SDK de reconocimiento de voz de código abierto para iOS y Android	- Gratuito y de código abierto -Integración con sintetizador de voz	-Menos mantenimiento y soporte en	Gratuito	Politepix (Halle Winkler)

Biblioteca	Características	Ventajas	Desventajas	Costos	Desarrollador
	- Compatibilidad con modelos acústicos y gramáticas personalizadas	-Funcionamiento sin conexión -Comunidad Activa	comparación con soluciones comerciales -Menor precisión -Limitación en la cantidad de palabras y frases -Menor rendimiento en entornos ruidosos -Limitada cantidad de recursos y documentación		
Microsoft Cognitive Services Speech SDK	- SDK de reconocimiento de voz de código abierto para iOS y Android - Compatibilidad con modelos acústicos y gramáticas personalizadas - Reconocimiento de voz y síntesis de voz - Síntesis de voz natural -Transcripción de audio en tiempo real -Soporte multiplataforma -Reconocimiento de voz en tiempo real -Amplio soporte de idiomas	- Alta precisión, tecnología avanzada de reconocimiento y síntesis de voz -Potencia el procesamiento en la nube -Facilidad de uso -Escalabilidad -Amplia gama de servicios adicionales como traducción análisis de texto, detección de emociones, etc. -Soporte y documentación extensos	- Requiere una cuenta de Azure -Costos asociados -Dependencia de servicios de la nube -Privacidad y seguridad -Limitaciones de personalización avanzados	Costo varía según du uso	Microsoft
Watson Developer Cloud	- Servicios de voz y lenguaje natural basados en la nube -Amplio conjunto de servicios -APIs y SDK fácil de usar -Personalización y entrenamiento de modelos -Modelos de lenguaje y acústicos avanzados -Integración con otras tecnologías IBM	- Amplio conjunto de servicios para procesamiento de voz y lenguaje natural -Poder y escalabilidad -Servicios adicionales integrados como procesamiento de lenguaje natural y comprensión de texto -Tecnología de vanguardia -Soporte y Documentación	- Requiere una cuenta de IBM Cloud y puede tener costos asociados -Curva de aprendizaje requiere mucho tiempo su entendimiento	Costo varía según su uso	IBM

Biblioteca	Características	Ventajas	Desventajas	Costos	Desarrollador
Kaldi	- Kit de herramientas de reconocimiento de voz de código abierto -Arquitectura modular -Algoritmos avanzados -Flexibilidad en la personalización	- Flexibilidad y personalización en la configuración del sistema de reconocimiento de voz -Código abierto -Alto rendimiento -Precisión -Amplia comunidad y documentación	-Requiere conocimientos técnicos y experiencia en configuración de sistemas de reconocimiento de voz -Curva de aprendizaje y complejidad -Documentación y recursos en ingles -Complejidad de configuración para ajustar modelos -Amplio consumo de requisitos	Gratuito	Universidad Johns Hopkins

Para el desarrollo del presente proyecto, se utilizará Android Speech Recognition, una herramienta que permite el reconocimiento de voz en dispositivos Android. Esta elección se basa en la ventaja de evitar la necesidad de instalar librerías externas, lo que simplificará el proceso de desarrollo. Además, Android Speech Recognition ofrece una interfaz de programación de aplicaciones fácil de usar, lo que facilitará la implementación del reconocimiento de voz en la aplicación para capturar datos del habla del usuario. Además, esta herramienta se integra fácilmente con la API de Text-to-Speech (TTS), una funcionalidad clave que permite generar respuestas de voz en la aplicación. Otro punto por considerar es que Android Speech Recognition es una solución popular, lo que implica que existe una amplia documentación y una comunidad activa que puede brindar apoyo en caso de problemas durante el desarrollo. En resumen, el uso de Android Speech Recognition proporciona una solución nativa, de fácil implementación y con diversas funcionalidades, lo que permitirá desarrollar una aplicación con manejo por voz de manera efectiva y con un buen soporte técnico.

3.1.3 Análisis de metodologías para el desarrollo móvil

Tabla 3. Cuadro comparativo de Metodologías para el desarrollo móvil

	Mobile D	TDD	RUP
Fases:	• Exploración • Inicialización • Producción • Estabilización • Pruebas del Sistema	• Desarrollar una prueba • Escribir Código • Validar Pruebas • Refactorización	• Principio • Elaboración • Construcción • Transición.
Objetivo:	El objetivo final de la metodología de desarrollo "Mobile D" es lograr la creación exitosa de una aplicación móvil que cumpla con los requisitos y expectativas de los usuarios y que brinde una experiencia óptima. La metodología se enfoca en el desarrollo ágil y rápido de aplicaciones móviles, teniendo en cuenta las particularidades y desafíos de este entorno.	El objetivo final del Desarrollo Dirigido por Pruebas (TDD) es desarrollar software de alta calidad que esté bien probado y cumpla con los requisitos del negocio.	El objetivo final del RUP es lograr la entrega exitosa de un sistema de software que cumpla con los requisitos del cliente, se ajuste a las necesidades del negocio y cumpla con los estándares de calidad establecidos.
Equipo	• Product Owner • Desarrollador de software • Diseñador de Interfaz • Arquitecto de software • Tester • Analista de negocios • Gerente de proyecto	• Desarrollador de software • Arquitecto de software • Tester • Analista de negocios • Gerente de proyecto	• Sponsor del proyecto • Desarrollador de software • Diseñador de Interfaz • Arquitecto de software • Tester • Analista de negocios • Gerente de proyecto
Comunicación	Ágil, colaborativa y constante entre los miembros del equipo de desarrollo y otras partes interesadas involucradas en el proyecto. Fomenta la comunicación frecuente, la colaboración cercana, la retroalimentación rápida y el uso de herramientas visuales y de	Se caracteriza por ser estrecha, enfocada y constante entre los miembros del equipo de desarrollo, y se centra en comprender los requisitos, definir pruebas, mantener informados a todos sobre el progreso y colaborar en la resolución de problemas.	Promueve una comunicación integral y colaborativa entre los diferentes miembros del equipo y las partes interesadas involucradas en el proyecto. Con reuniones regulares y la interacción directa con los usuarios finales

	colaboración para mantener a todos los involucrados informados y alineados en el progreso del desarrollo de la aplicación móvil.		
Enfoque de Programación	Desarrollo colaborativo y ágil, enfoque en la implementación de funcionalidades de aplicaciones móviles	Enfoque centrado en escribir pruebas antes de desarrollar funcionalidades, implementación iterativa	Enfoque estructurado y orientado a fases del proyecto, programación basada en requisitos y diseño arquitectónico
Pruebas	Pruebas funcionales y de rendimiento enfocadas en la usabilidad y experiencia del usuario, Se realizan durante todo el proceso de desarrollo, con pruebas continuas y frecuentes	Enfoque en pruebas unitarias, pruebas de integración y pruebas de aceptación automatizadas, Las pruebas unitarias se escriben antes de desarrollar la funcionalidad correspondiente	Pruebas de sistema, pruebas de integración y pruebas de aceptación. Las pruebas se realizan en diferentes fases del proyecto, como parte de la verificación y validación
Documentación	Documento de visión, especificaciones de requisitos, diseños de interfaz, manuales de usuario, informes de pruebas	Documentación de pruebas unitarias, documentación de diseño de funcionalidades	Documento de visión, documento de especificación de requisitos, modelos de casos de uso, modelos de diseño, plan de pruebas, informes de seguimiento y control
Calidad del producto final	Enfoca en la entrega de productos móviles de alta calidad que cumplan con las expectativas del usuario	Promueve la calidad del producto a través de pruebas unitarias exhaustivas y pruebas continuas	Enfoca en la calidad del producto a través de la verificación y validación en cada fase del proyecto
Dificultad de adopción	Moderada	Moderada	Alta
Enfoque principal	Agilidad y entrega rápida de valor al usuario	Calidad del código y pruebas exhaustivas	Gestión del proyecto y desarrollo estructurado
Tiempo de producción	Corto a Moderado	Corto a Moderado	Largo a moderado
Número mínimo de miembros	Pequeños equipos	Puede ser un solo desarrollador o pequeños equipos	Equipos de desarrollo de diferentes tamaños
Gestión de Requisitos	Flexibilidad y adaptabilidad	Requisitos impulsados por pruebas	Análisis y especificación detallada
Extensión de Documentación	Mínima documentación formal	Documentación de pruebas y código	Documentación extensiva y específica
Flexibilidad y Adaptabilidad	Alta	Alta	Moderada
Tiempo estimado de ejecución	3-6 meses	2-4 meses	6-18 meses
Recursos	Bajo a moderado	Moderado	Moderado a Alto

Después de realizar un análisis de diferentes metodologías, se ha decidido trabajar con "Mobile D" debido a su enfoque específico en el desarrollo de aplicaciones móviles. Esta metodología se centra en las particularidades del desarrollo móvil, como el rendimiento de las aplicaciones en dispositivos, y proporciona una estructura y guías más detalladas en comparación con otras metodologías disponibles. La metodología "Mobile D" se destaca por su capacidad de adaptarse fácilmente a los cambios que pueden surgir durante el desarrollo del proyecto, gracias a su base en principios ágiles. Además, se caracteriza por tener ciclos de desarrollo más cortos, lo que permitirá obtener entregables funcionales de manera incremental.

Un aspecto clave de "Mobile D" es su enfoque en la experiencia del usuario y la usabilidad. Se pone un énfasis especial en garantizar una experiencia de interacción óptima para los usuarios con discapacidad visual, ya que el aplicativo está destinado a este grupo. Para lograrlo, se emplearán técnicas y principios de accesibilidad que aseguren la facilidad de uso y brinden una experiencia satisfactoria para las personas con discapacidad visual. En resumen, la elección de trabajar con "Mobile D" como metodología de desarrollo se basa en su enfoque específico en el desarrollo de aplicaciones móviles, su flexibilidad para adaptarse a los cambios, los ciclos de desarrollo más cortos y su enfoque en la experiencia del usuario y la usabilidad, especialmente para las personas con discapacidad visual a las que se dirige el aplicativo.

3.1.4 Fase de la metodología Mobile D

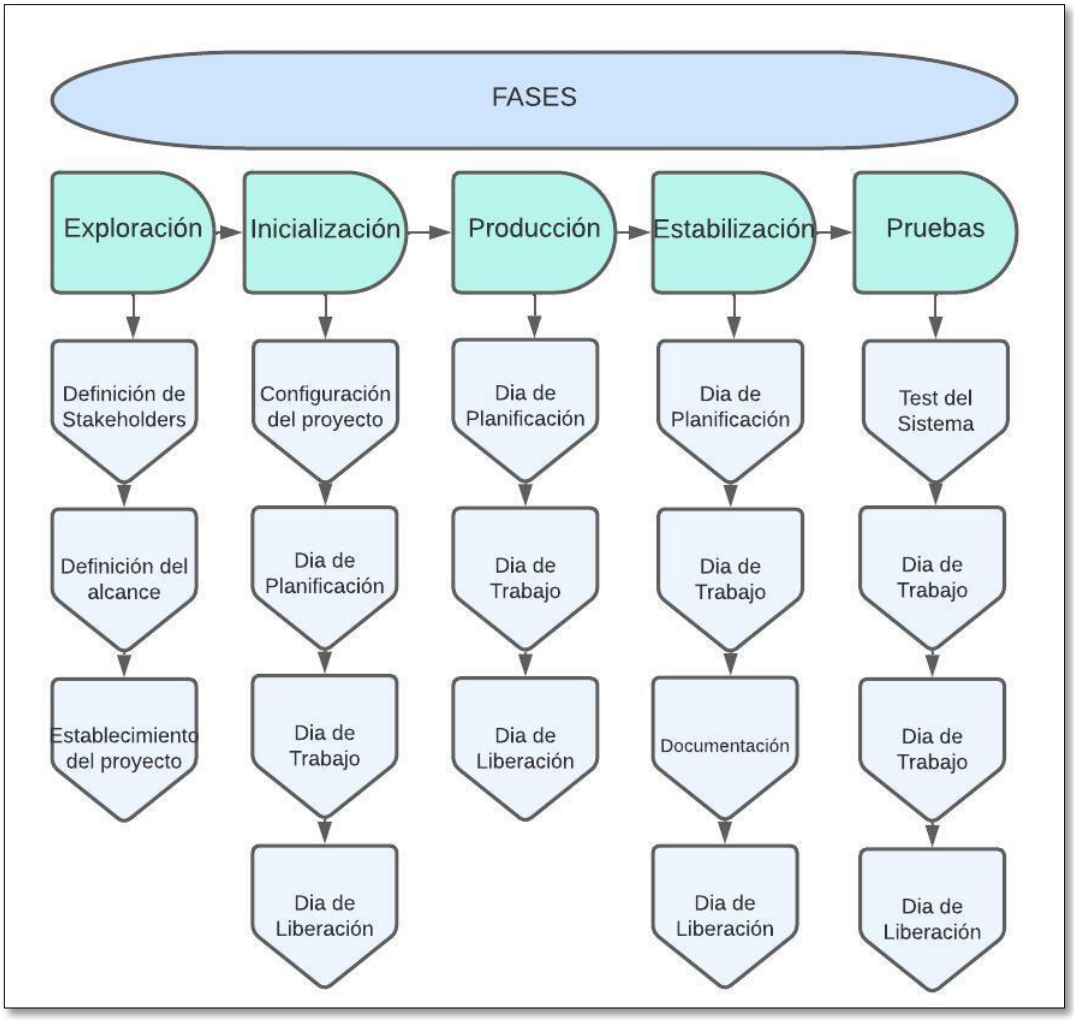


Figura 21. Fases de la Metodología Mobile D.

Fase 1: Exploración

El desarrollador debe crear un plan y determinar las características del proyecto, lo cual se realiza en tres etapas: Definición de Stakeholder, definición de alcance y establecimiento del proyecto.

Fase 2: Inicialización

Esta fase se enfoca en la detección temprana de problemas que surjan en el desarrollo del sistema para que no afecten las fases siguientes. Además, se preparan las herramientas necesarias para la producción del sistema. Se divide en cuatro etapas: configuración del proyecto, día de planificación, día de trabajo y día de liberación.

Fase 3: Producción

Al igual que en la fase 2, se repite la programación de 3 días que irá repitiéndose iterativamente hasta completar las funcionalidades del proyecto. Primero se planifican las tareas a realizar, se preparan pruebas. Las tareas se desarrollarán durante el día de trabajo.

Fase 4: Estabilización

Para asegurar que el producto sea de calidad, se realizan las últimas acciones de integración para que funcione correctamente. Adicionalmente, en esta fase se incluye la documentación.

Fase 5: Pruebas

En esta fase, se verifica una versión estable del producto verificando los requerimientos y se eliminan los defectos encontrados.

3.2 Desarrollo de la propuesta

3.2.1 Fase 1 Exploración

En esta fase, se definen los stakeholders, que son las personas interesadas o grupos de interés que participan en el proyecto de manera directa o indirecta. Se define el alcance y se establece el proyecto.

a. Definición de los Stakeholder

Desarrollador: El investigador del presente proyecto, quien se encarga del desarrollo, diseño y dar soluciones.

Clientes: Enfocado en personas con discapacidad visual en el cantón Ambato.

Usuarios: Personas con acceso a un smartphone con sistema Android y acceso a internet.

b. Definición del alcance

El alcance del aplicativo se alinea con las necesidades y dificultades identificadas durante la recolección de información mediante las encuestas realizadas a personas con discapacidad visual. La aplicación se adapta a las preferencias y necesidades de cada tipo de usuario:

- Usuario con discapacidad visual:
  - Las funcionalidades incluyen el manejo de voz para iniciar sesión y registrar un usuario único.
  - Verificar contactos, enviar mensajes, reproducir mensajes y ajustar configuraciones son opciones disponibles.
  - Cada una de las anteriores opciones proporcionará indicaciones y descripciones de las funciones realizadas. Esto facilitará que el usuario se adapte al uso y manejo ágil mediante la práctica.
- Usuario General:
  - Al poseer capacidades visuales, este usuario tiene acceso a la información mostrada en cada uno de los apartados.
  - Además, cuenta con comandos que agilizarán su interacción con contactos que tienen limitaciones visuales.

Limitaciones

El aplicativo tiene ciertas limitaciones que se detallan a continuación:

- El dispositivo móvil debe contar con acceso a internet para ejecutar las funciones principales de envío y recepción de mensajes. Además, necesita tener acceso a los usuarios registrados en el aplicativo para compararlos con los contactos ya registrados en el dispositivo.

- El sistema operativo requerido es Android, con una versión mínima de 8.0 Oreo para su ejecución, funcionando óptimamente en versiones superiores.

c. Establecimiento del proyecto

Las herramientas manejadas para el desarrollo del proyecto fueron las siguientes:

- **Android Studio**

  Es un entorno de desarrollo utilizado para crear aplicativos Android. A través del editor de código y herramientas desarrolladas por el personal de IntelliJ IDEA, mejora la productividad del desarrollador. Permite el desarrollo de aplicaciones con lenguajes como Java y Kotlin.

- **Java**

  Es un lenguaje de programación desarrollado por Sun Microsystems en 1995. Con el tiempo, ha evolucionado y en la actualidad, la mayoría de los sistemas siguen basándose en Java.

- **Firebase**

  Es una plataforma completa para el desarrollo de aplicativos que permite compilar y desarrollar. Dentro de su entorno, ofrece varios servicios para la integración de funcionalidades según lo requiera el desarrollador.

- **Firebase Authentication**

  Es un servicio que ofrece la posibilidad de almacenar datos de un usuario para la autenticación en un sistema. Proporciona servicios de Backend, SDK fáciles de usar y bibliotecas de IU. Posee una variedad de métodos de autenticación, como contraseñas, números de teléfono, así como mediante las redes más utilizadas como Google o Facebook.

- **Realtime Database**

  Es una base de datos en tiempo real NoSQL que permite la posibilidad de almacenar y sincronizar datos de usuarios al momento. Utiliza una estructura JSON que facilita la manipulación y lectura de datos.

- **Firebase Cloud Messaging**

  Es un sistema de mensajería multiplataforma con el cual se puede notificar a una aplicación cliente que posee nuevos datos.

3.2.2 Fase 2 Inicialización

En esta fase, se lleva a cabo el diseño del aplicativo móvil, tomando en cuenta las necesidades y requerimientos de los usuarios, estableciendo configuraciones y herramientas a utilizar.

a. Configuración del Proyecto

En esta parte del proyecto se define las herramientas que se utilizan para el desarrollo del aplicativo móvil.

- **Tipo de Aplicación:** Nativa
- **Servicios de Autenticación:** Firebase Authentication
- **Servicios de Manejo de Base de Datos:** Realtime Database
- **Servicio de Mensajería:** Cloud Messaging

Preparación del Ambiente

Para el desarrollo se realizó las siguientes instalaciones y configuraciones:

- Android Studio
- SDK versión 33
- Firebase Authentication
- Realtime Database
- Firebase Cloud Messaging

Planificación de las fases

Tabla 4. Planificación de las fases.

Fase	Iteración	Descripción
Exploración	Iteración 0	-Definición de Stakeholders -Definición de alcance -Definición del proyecto
Inicialización	Iteración 1	-Definición de requerimientos de usuarios -Configuración del ambiente de desarrollo
	Iteración 2	-Diseño de la Base de datos -Diseño de interfaz del aplicativo
Producción	Iteración 3	-Implementar las conexiones a Firebase con Android Studio -Crear y actualizar StoryCard
	Iteración 4	-Implementar diseño de inicio y registro de usuarios -Crear y actualizar StoryCard
	Iteración 5	-Implementar conexiones de autenticación con Firebase de logueo y registro de usuarios -Crear y actualizar StoryCard
	Iteración 6	-Implementar funcionalidad de verificación automática leyendo mensajes OTP -Crear y actualizar StoryCard
	Iteración 7	-Implementar almacenar datos de usuario en Realtime Firebase -Crear y actualizar StoryCard
	Iteración 8	-Implementar funciones para comparar contactos de celular y base de datos -Crear y actualizar StoryCard
	Iteración 9	-Implementar comando de listar contactos -Crear y actualizar StoryCard
	Iteración 10	-Implementar comando de envió de mensajes. -Crear y actualizar StoryCard
	Iteración 11	-Implementar comando de reproducción de mensajes.
	Iteración 12	-Implementar interfaz para envío y recepción de mensajes. -Crear y actualizar StoryCard
	Iteración 13	-Implementar funcionalidades de configuraciones. -Crear y actualizar StoryCard
	Iteración 14	-Implementar interfaz de configuraciones. -Crear y actualizar StoryCard
Estabilización	Iteración 15	-Ajustes de diseño de interfaz. -Limpieza de código. -Corrección de errores generales.
Pruebas	Iteración 16	-Pruebas de la aplicación.

Diseño de la arquitectura de la aplicación

La arquitectura del aplicativo es Orientada a Servicios, ya que el usuario accederá a los servicios de Firebase para ejecutar las funciones de mensajería y así brindar soporte a los requisitos de negocios. Los usuarios necesitan acceso a internet para utilizar los servicios, como la autenticación del usuario, el registro de sus datos y mensajes en la base de datos, así como la mensajería que se encarga de las notificaciones. En la Figura 22, se observa el proceso que interviene en el funcionamiento de la aplicación.

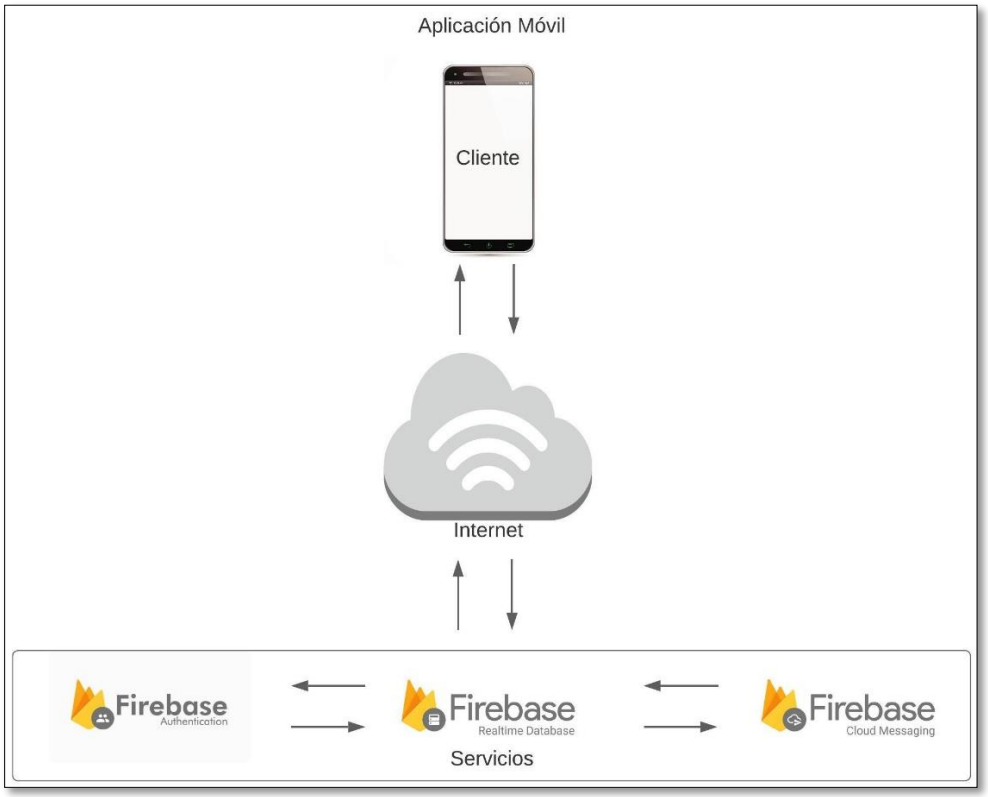


Figura 22.Arquitectura Orientada a Servicios (SOA).

Base de datos NoSQL

Para el desarrollo del aplicativo, se utilizó Firebase, una base de datos NoSQL, debido a los requisitos de acceso a datos en tiempo real. A diferencia de una base de datos de tipo SQL, Firebase no posee tablas ni diagramas; la base de datos en Realtime Database está basada en colecciones.

- Colecciones

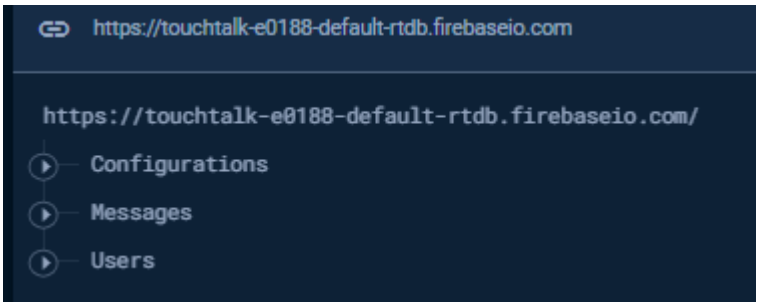


Figura 23. Colecciones en Firebase.

Colección "Messages": Esta colección está compuesta por el ID del usuario que envía el mensaje, el ID del usuario que recibe el mensaje, el ID del mensaje, el contenido del mensaje, el tipo de mensaje y el estado del mensaje.

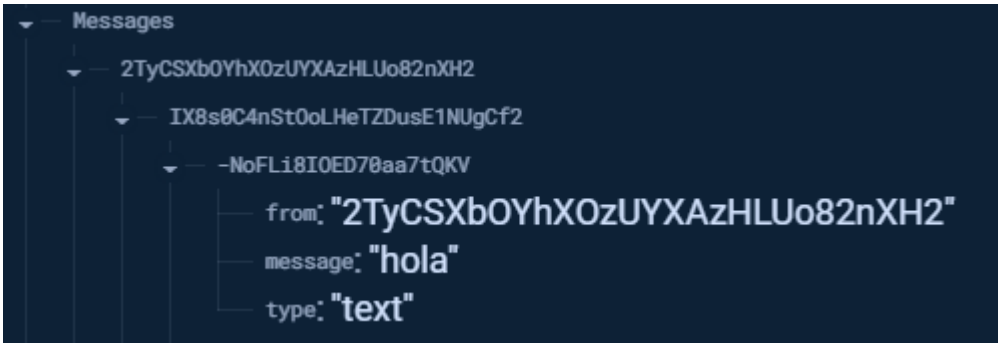


Figura 24. Colección Messages

Colección "Users": Esta colección contiene la información del usuario, como su número de teléfono celular con el añadido de 593, el "device token", que es un código único generado a partir de Cloud Messages para identificar el dispositivo al que se enviará un mensaje, y el ID del usuario registrado.

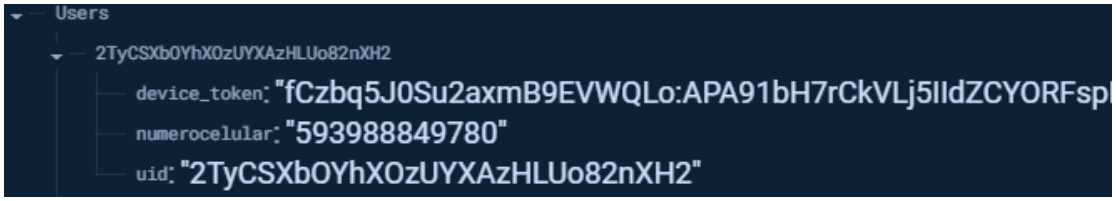


Figura 25. Colección Users.

Esquema de Navegabilidad

En la Figura 26, se representa el flujo de navegación de los usuarios con respecto a la aplicación, lo que permite tener una comprensión de la interacción que se tendrá con el aplicativo.

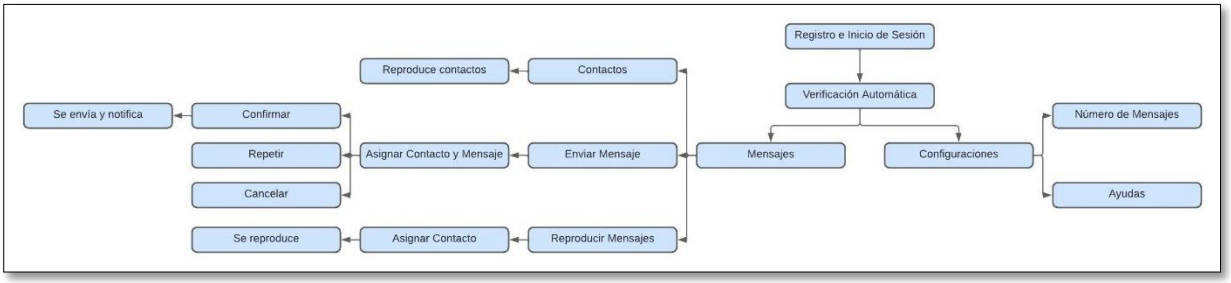


Figura 26. Esquema de Navegabilidad.

Diagramas de casos de uso

Los diagramas de caso de uso permiten establecer las acciones que puede realizar el usuario que haga uso del aplicativo con relación a las funciones integradas. En la Figura 27, se exponen las acciones que puede ejecutar el usuario con discapacidad visual al usar la aplicación móvil.

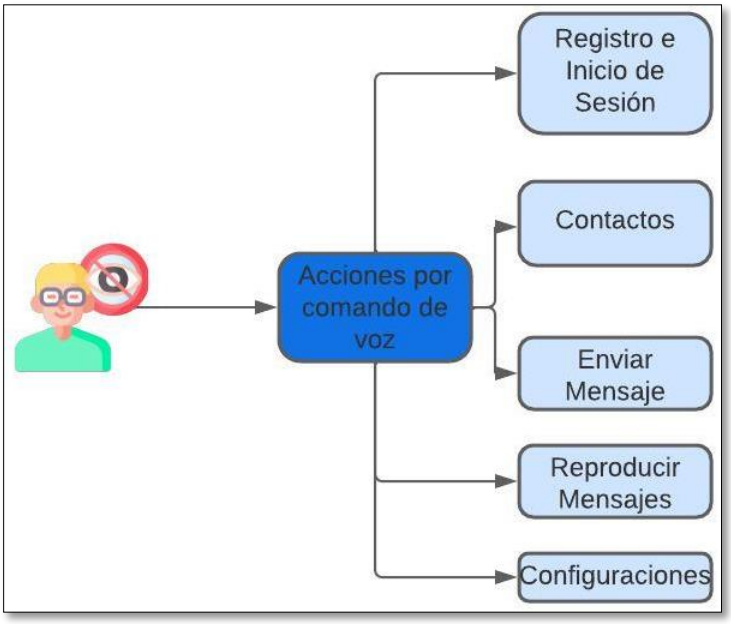


Figura 27. Caso de uso para usuarios con discapacidad visual.

Prototipo de la aplicación

Se utilizó la herramienta JustinMind para crear los prototipos de las interfaces gráficas del aplicativo móvil destinado a usuarios con discapacidad visual, con el fin de tener una idea clara de sus funcionalidades.

Registro e Inicio de Sesión: En la Figura 28, se presenta un campo de texto referente al número celular que debe ser ingresado. Al iniciar el aplicativo, se mostrará un mensaje

indicando al usuario que ingrese su número celular, omitiendo el primer cero. Para proceder con el ingreso, el usuario puede presionar el botón "Registrarse". Se ha ajustado el tamaño del botón con la consideración de facilitar la interacción para usuarios con discapacidad.

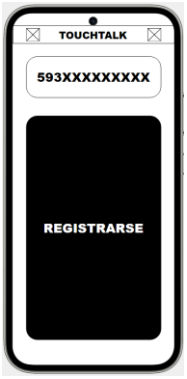


Figura 28. Registro e Inicio de Sesión.

Verificación: En la Figura 29, se ha actualizado la funcionalidad del botón para permitir al usuario confirmar el número ingresado. Después de registrarse, se procede a la verificación. Aunque existe un botón, se reproduce un mensaje de espera para que se verifique automáticamente con un código de 6 dígitos enviado a través de un mensaje OTP.

Figura 29. Verificación de Número Celular

Mensajes: En la Figura 30, se presenta la pantalla principal del aplicativo, la cual consta de dos secciones. La primera es "Mensajes", donde se encuentra un único botón que ocupa casi todo el espacio de la pantalla para facilitar su manipulación por parte del usuario. Además, se dispone de un campo de texto en la parte inferior para proporcionar información adicional a los usuarios que tienen la capacidad de apreciar el contenido de la pantalla, permitiéndoles comunicarse con sus contactos que tienen discapacidad visual.

Figura 30. Mensajes.

Configuraciones: En la Figura 31, se puede observar un único botón que te permitirá modificar opciones sencillas del aplicativo, como el número de mensajes que deseas que se reproduzcan al acceder a un usuario y si se desea eliminar los mensajes largos de ayuda que se reproducen en cada pantalla u opción que se utilice. Además, encontrará un campo de texto inferior que permitirá visualizar las opciones que ya están predeterminadas.

Figura 31. Configuraciones.

3.2.3 Fase 3 Producción

Durante la fase 3 se ejecuta cada una de las iteraciones definidas para si generar los entregables que son detallados en los posteriores StoryCard.

a. Backend

Al utilizar Firebase, se emplea un Backend como servicio (BaaS). Al ser una plataforma de desarrollo en la nube, ofrece una amplia gama de herramientas y servicios para el desarrollo de aplicaciones web y móviles. Al utilizar Android Studio como entorno de desarrollo, desarrollado por los mismos creadores de Firebase, se logra una integración más ágil al importar librerías sin necesidad de descargar herramientas adicionales, lo que permite ahorrar tiempo y recursos en el desarrollo.

La parte lógica del negocio se desarrolla en su mayoría mediante los servicios de Firebase, beneficiándose de las funciones administrativas que brinda la nube. Se incluyen funciones adicionales para complementar los servicios existentes, como la verificación automática de los códigos OTP o el envío de datos a la base de datos en tiempo real a través de comandos de voz. Todo esto facilita un enfoque más específico en la lógica de programación y menos en la infraestructura.

b. Archivos de Código Fuente

Los archivos de código fuente o clases en el caso de desarrollo del proyecto .java son los que contienen el lenguaje para la implementación de la lógica y funcionalidades de la aplicación.

Listado de Clases

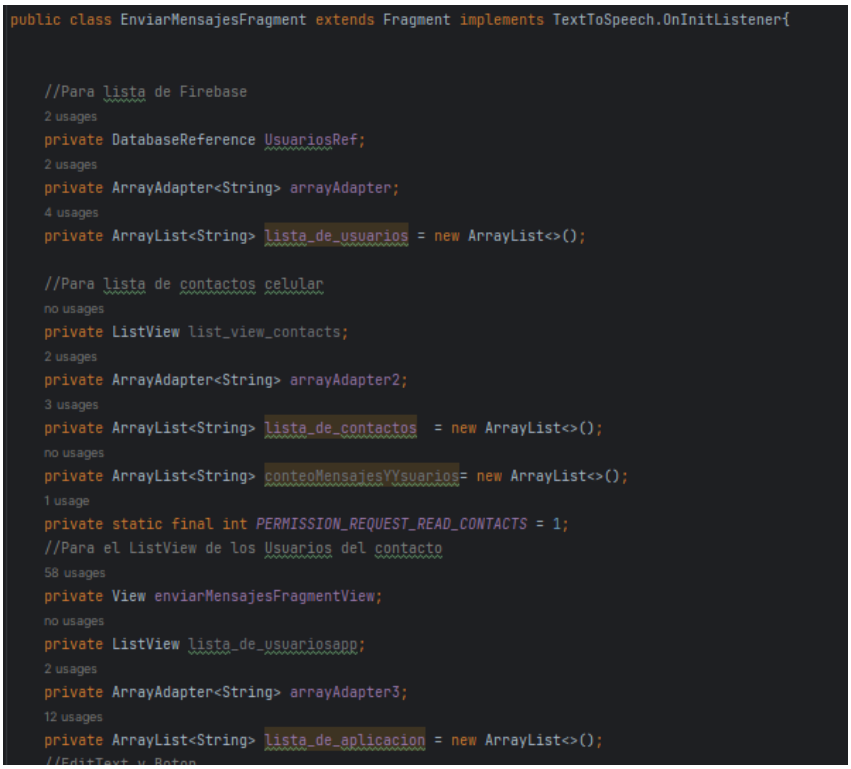
- ConfiguracionesFragment.java

  En la Figura 32 se muestra la clase 'ConfiguracionesFragment', que es el apartado encargado de modificar los parámetros de mensajes de ayuda, como su extensión y la cantidad de mensajes que el usuario desea reproducir.

Figura 32. ConfiguracionesFragment

- EnviarMensajesFragment.java

En la Figura 33 se presenta la clase ‘EnviarMensajesFragment’, la cual se encarga de extraer información de contactos del dispositivo y de la base de datos en Firebase. Esta clase incluye los comandos necesarios para el envío y la reproducción de mensajes.



```
public class EnviarMensajesFragment extends Fragment implements TextToSpeech.OnInitListener{


    //Para lista de Firebase
    2 usages
    private DatabaseReference UsuariosRef;
    2 usages
    private ArrayAdapter<String> arrayAdapter;
    4 usages
    private ArrayList<String> lista_de_usuarios = new ArrayList<>();

    //Para lista de contactos celular
    no usages
    private ListView list_view_contacts;
    2 usages
    private ArrayAdapter<String> arrayAdapter2;
    3 usages
    private ArrayList<String> lista_de_contactos  = new ArrayList<>();
    no usages
    private ArrayList<String> conteoMensajesYYsuarios= new ArrayList<>();
    1 usage
    private static final int PERMISSION_REQUEST_READ_CONTACTS = 1;
    //Para el ListView de los Usuarios del contacto
    58 usages
    private View enviarMensajesFragmentView;
    no usages
    private ListView lista_de_usuariosapp;
    2 usages
    private ArrayAdapter<String> arrayAdapter3;
    12 usages
    private ArrayList<String> lista_de_aplicacion = new ArrayList<>();
    //EditText y Boton
```

Figura 33. EnviarMensajesFragment

- FCMSend.java

En la Figura 34 se muestra la clase ‘FCMSend’, es un componente para el envío de notificaciones a través del servicio Firebase Cloud Messages (FCM). Se incluye el método ‘pushNotification’ que es el encargado de enviar una notificación a un dispositivo en específico identificado por su token.

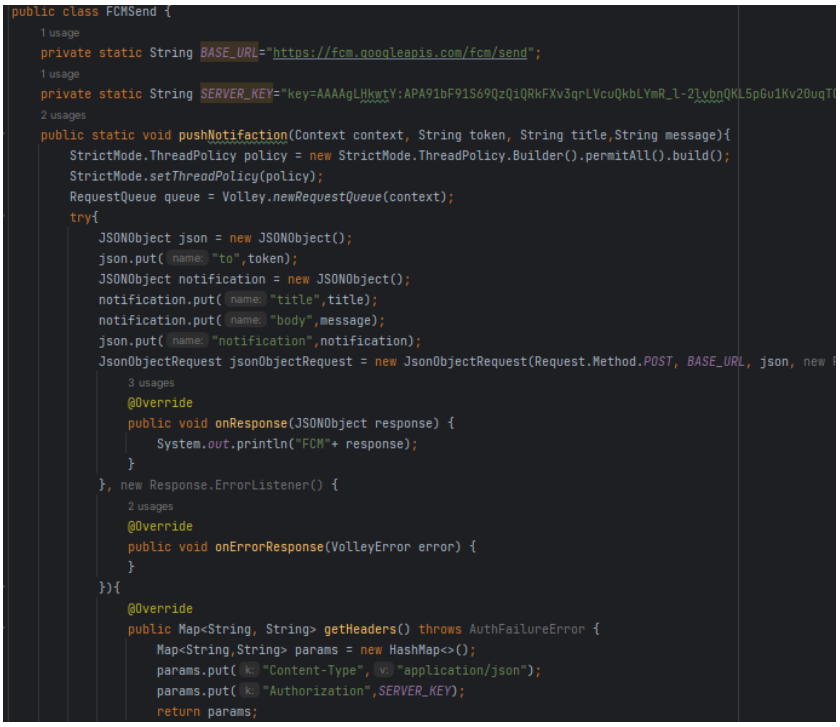


Figura 34. FCMSend

- MainActivity.java

En la figura 35 se muestra la clase principal ‘MainActivity’ quien se encarga de controlar la interfaz principal, que se encargar de la autenticación del usuario y de no estar logueado redirige a la actividad de registro ‘RegisterActivity’.

```java
public class MainActivity extends AppCompatActivity {

    2 usages
    private Toolbar mToolbar;
    4 usages
    private ViewPager2 myViewPager2;
    3 usages
    private TabLayout myTabLayout;
    2 usages
    private TabsAccessorAdapter myTabsAccessorAdapter;

    2 usages
    private FirebaseUser currentUser;
    3 usages
    private FirebaseAuth mAuth;
    2 usages
    private DatabaseReference RootRef;

    @SuppressLint("WrongViewCast")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mToolbar = findViewById(R.id.main_page_toolbar);
        setSupportActionBar(mToolbar);
        getSupportActionBar().setTitle("TouchTalk");

        myTabLayout=findViewById(R.id.tabs_layout);
        myViewPager2 = findViewById(R.id.view_pager);
        myTabsAccessorAdapter = new TabsAccessorAdapter( fragmentActivity: this);
        myViewPager2.setAdapter(myTabsAccessorAdapter);
        mAuth = FirebaseAuth.getInstance();
        currentUser=mAuth.getCurrentUser();
        RootRef= FirebaseDatabase.getInstance().getReference();
```

Figura 35.MainActivity

- Messages.java

La Figura 36 muestra la clase Messages que representa un modelo de datos para gestionar los mensajes en la aplicación. Entre sus atributos se incluye ‘from’ para identificar al remitente, ‘message’ para almacenar el contenido del mensaje y ‘type’ para especificar el tipo de mensaje.

```java
public class Messages {
    3 usages
    private String from, message, type;
    no usages
    public Messages(){
    }
    no usages
    public Messages(String from, String message, String type) {
        this.from = from;
        this.message = message;
        this.type = type;
    }
```

Figura 36. Clase Messages

- Notification_Receiver.java

La figura 37 muestra la clase ‘Notification_Receiver’ que funciona como un receptor de transmisiones (BroadCastReceiver) en la Aplicación. Su propósito es gestionar las notificaciones recibidas a través de FCM y realizar acciones como reproducir la información de la notificación recibida.

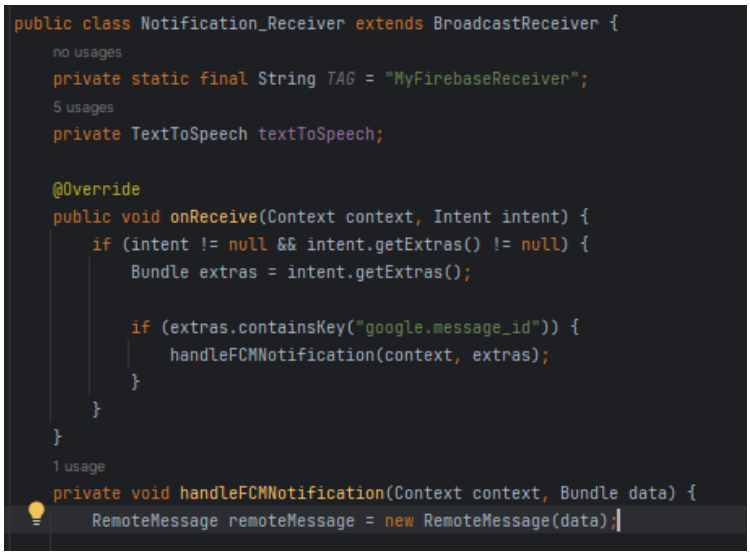


Figura 37. Notification_Receiver

- OTP_Receiver.java

La figura 38 muestra la clase OTP_Receiver que es un receptor de transmisiones, diseñado para interceptar mensajes de texto (SMS) que contiene condigo de verificación OTP. Busca un código de seis dígitos y lo extrae. Facilitando la automatización de la captura de códigos.

```
public class OTP_Receiver extends BroadcastReceiver {

    3 usages
    private static EditText editText;
    2 usages
    public void setEditText(EditText editText) { OTP_Receiver.editText=editText; }
    @Override
    public void onReceive(Context context, Intent intent) {
        SmsMessage[] messages = Telephony.Sms.Intents.getMessagesFromIntent(intent);

        for (SmsMessage sms : messages) {
            String message = sms.getMessageBody();
            // Utiliza expresiones regulares para buscar y extraer el código de verificación
            String otp = extractOTP(message);
            // Configura el EditText con el código OTP
            setEditTextValue(otp);
        }
    }
    1 usage
    private String extractOTP(String message) {
        // Patrón para buscar el código de verificación en diferentes formatos
        Pattern pattern = Pattern.compile( regex: "\\b\\d{6}\\b");
        Matcher matcher = pattern.matcher(message);
        // Encuentra la primera coincidencia
        if (matcher.find()) {
            return matcher.group();
        }
        // Si no se encuentra ninguna coincidencia, devuelve una cadena vacía
        return "";
    }
    1 usage
    private void setEditTextValue(String otp) {
        if (editText != null) {
            editText.setText(otp);
        }
    }
}
```

Figura 38. OTP_Receiver

- PushNotificationService.java

La figura 39 muestra la clase 'PushNotificationService' que se extiende a 'FirebaseMessagingService' lo que se utiliza para manejar las notificaciones Push recibidas a través de FCM.

```
public class PushNotificationService extends FirebaseMessagingService {

    8 usages
    private TextToSpeech textToSpeech;
    3 usages
    private boolean isTtsInitialized = false;
    1 usage
    private DatabaseReference UsuariosRef;
    3 usages
    private ArrayList<String> lista_de_usuarios = new ArrayList<>();
    1 usage
    private ArrayList<String> lista_de_contactos  = new ArrayList<>();
    1 usage
    private String messageSenderID;
    1 usage
    private ArrayAdapter<String> arrayAdapter;
    1 usage
    private ArrayAdapter<String> arrayAdapter3;
    2 usages
    private ArrayList<String> lista_de_aplicacion = new ArrayList<>();
    2 usages
    private List<String> lista_de_usuarios_app_nombres = new ArrayList<>();

    3 usages
    @SuppressLint("MissingPermission")
    @Override
    public void onMessageReceived(@NonNull RemoteMessage message) {
        String title = message.getNotification().getTitle();
        String[] partes = title.split( regex: "\\+");
```

Figura 39. PushNotificationService

- RegisterActivity.java

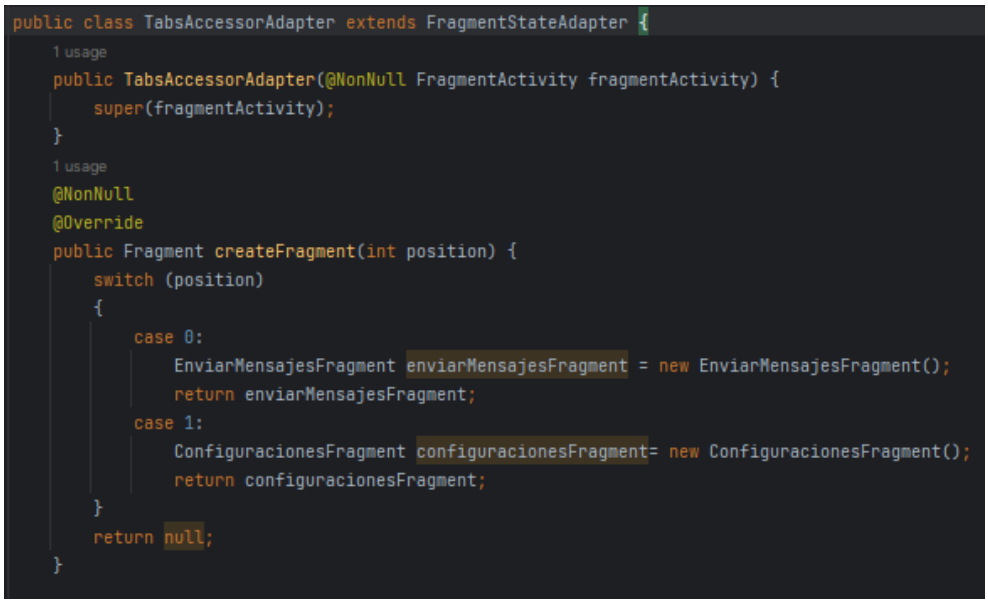
La Figura 40 muestra la clase ‘RegisterActivity’, la cual se encarga del manejo del registro de usuarios en la aplicación, permitiendo el ingreso de un número celular y automatizando el proceso de registro e ingreso de datos en la base de datos de Firebase mediante funciones con OTP_Receiver.

```
public class RegisterActivity extends AppCompatActivity implements OnInitListener{

    2 usages
    private Toolbar mToolbar;
    12 usages
    private Button CreateAccountButton;
    16 usages
    private Button CreateAccountButton2;
    12 usages
    private Button CreateAccountButton3;
    11 usages
    private EditText NumberPhone;
    11 usages
    private EditText NumberVerification;
    20 usages
    private TextToSpeech textToSpeech;
    1 usage
    private static final int REQ_CODE_SPEECH_INPUT=100;
```

Figura 40. RegisterActivity

- TabsAccesorAdapter.java

La figura 41 muestra la clase 'TabsAccesorAdapter' que se extiende en 'FragmentStateAdapter' utilizado para gestionar fragmentos. Se trabaja con 2 fragmentos 'EnviarMensajesFragment' y 'ConfiguracionesFragment'.



```
public class TabsAccessorAdapter extends FragmentStateAdapter {
    1 usage
    public TabsAccessorAdapter(@NonNull FragmentActivity fragmentActivity) {
        super(fragmentActivity);
    }
    1 usage
    @NonNull
    @Override
    public Fragment createFragment(int position) {
        switch (position)
        {
            case 0:
                EnviarMensajesFragment enviarMensajesFragment = new EnviarMensajesFragment();
                return enviarMensajesFragment;
            case 1:
                ConfiguracionesFragment configuracionesFragment= new ConfiguracionesFragment();
                return configuracionesFragment;
        }
        return null;
    }
```

Figura 41. TabsAccesorAdapter

c. Archivos de Recursos

Archivos XML: Estos archivos definen la estructura y diseño de los elementos que intervienen en la interfaz de usuario, así como otros elementos que intervienen en una interfaz como menús, diseños de vistas, estilos entre otros.

Listado de interfaces

- activity_main.xml
- activity_register.xml
- app_bar_layout.xml
- fragment_configuraciones.xml
- fragment_enviar_mensajes.xml

d. Archivos de Configuración

Se encuentran los archivos de manifiesto y archivos Gradle. Los archivos de manifiesto se encargan de incluir los permisos del aplicativo, el nombre del paquete, servicios, actividades principales y las configuraciones generales del aplicativo. Los archivos Gradle se encargan de construir el proyecto. Existen dos tipos, uno a nivel de proyecto y otro a nivel de módulo. En estos se incluyen las dependencias, las versiones de compilación y los servicios externos, como Firebase.

Lista de permisos en AndroidManifest.xml

- INTERNET
- ACCESS_NETWORK_STATE
- WAKE_LOCK
- READ_CONTACTS
- READ_PHONE_STATE
- READ_PHONE_NUMBERS
- RECEIVE_SMS
- WRITE_SMS

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.READ_PHONE_NUMBERS"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"
        tools:ignore="PermissionImpliesUnsupportedChromeOsHardware" />
    <uses-permission android:name="android.permission.WRITE_SMS"
        tools:ignore="PermissionImpliesUnsupportedChromeOsHardware" />
```

Figura 42. Permisos en AndroidManifest.xml

Lista de dependencias

- implementation(platform("com.google.firebase:firebase-bom:32.3.1"))
- implementation 'androidx.appcompat:appcompat:1.6.0'
- implementation 'com.google.android.material:material:1.10.0'
- implementation 'androidx.viewpager2:viewpager2:1.0.0'
- implementation 'androidx.constraintlayout:constraintlayout:2.1.
- implementation 'com.google.firebase:firebase-auth:22.2.
- implementation 'com.google.firebase:firebase-database:20.3.0'
- implementation 'com.android.car.ui:car-ui-lib:2.5.1'
- implementation 'com.google.firebase:firebase-messaging:23.3.1'
- implementation 'com.android.volley:volley:1.2.1'
- testImplementation 'junit:junit:4.13.2'
- androidTestImplementation 'androidx.test.ext:junit:1.1.5'
- androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
- implementation "androidx.lifecycle:lifecycle-viewmodel:2.3.1"

Ejemplo de código de importaciones en clases

```
30  import com.google.android.gms.tasks.OnCompleteListener;
31  import com.google.android.gms.tasks.Task;
32  import com.google.firebase.FirebaseException;
33  import com.google.firebase.auth.AuthResult;
34  import com.google.firebase.auth.FirebaseAuth;
35  import com.google.firebase.auth.PhoneAuthCredential;
36  import com.google.firebase.auth.PhoneAuthOptions;
37  import com.google.firebase.auth.PhoneAuthProvider;
38  import com.google.firebase.database.DatabaseReference;
39  import com.google.firebase.database.FirebaseDatabase;
40  import com.google.firebase.messaging.FirebaseMessaging;
```

Figura 43. Ejemplo de código de importaciones.

La **Figura 43** indica las librerías que intervienen en el registro de un usuario, donde se encuentran las encargadas de permitir el uso de los servicios de Firebase para la autenticación a través de un número telefónico real y agregarlos datos a la base de datos.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RegisterActivity">
    <ImageView
        android:id="@+id/register_image"
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:src="@drawable/signup_photo"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:scaleType="center"/>
    <EditText...>
    <EditText...>
    <Button
        android:id="@+id/register_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/register_password"
        android:padding="4dp"
        android:layout_marginTop="25dp"
        android:layout_marginLeft="25dp"
        android:layout_marginRight="25dp"
        android:text="Crear Cuenta"
        android:textAllCaps="false"
        android:textSize="25sp"
```

Figura 44. Ejemplo de código de interfaz de Registro.

La **Figura 44** muestra la estructura y diseño de interfaz de usuario para el registro de nuevos usuarios, entre sus elementos más comunes encontramos las etiquetas y atributos.

```
//noinspection GradleCompatible
implementation 'com.android.support:appcompat-v7:28.0.0'
implementation 'com.android.support:design:-v7:28.0.0'
implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
implementation 'androidx.appcompat:appcompat:1.6.0'
implementation 'com.google.android.material:material:1.4.0'
implementation 'de.hdodenhof:circleimageview:3.1.0'
implementation platform('com.google.firebase:firebase-bom:32.1.0')
implementation 'com.google.firebase:firebase-analytics'
implementation 'com.google.firebase:firebase-auth-ktx:22.0.0'
implementation 'com.google.firebase:firebase-database-ktx:20.2.2'
implementation 'com.google.firebase:firebase-storage-ktx:20.2.0'
implementation 'com.firebaseui:firebase-ui-database:8.0.2'
testImplementation 'junit:junit:4.13.2'
androidTestImplementation 'com.android.support.test:runner:1.0.2'
androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
```

Figura 45. Implementaciones en build Gradle para la utilización de servicios Firebase.

En la **Figura 45** se muestran las implementaciones necesarias para hacer uso de los diferentes servicios de Firebase.

e. StoryCard

Las StoryCard son herramientas fundamentales en el desarrollo ágil de software, ya que permiten detallar de manera concisa los requisitos y funcionalidades de la aplicación a través de tarjetas documentales. En estas tarjetas, se incluye información crucial sobre la historia, descripción, sprint correspondiente, ubicación de ejecución, implementación y verificación. Esta metodología no solo facilita una planificación más efectiva, sino que también mejora la comunicación y colaboración dentro del equipo de desarrollo. La claridad y la precisión de la información contenida en las StoryCard contribuyen significativamente a la eficiencia y el éxito del proyecto.

StoryCard – Conexiones de Firebase con Android Studio

Tabla 5. StoryCard - Conexiones de servicios de Firebase con Android Studio.

N°	**Tipo**	**Dificultad**		**Esfuerzo**		**Prioridad**
		Antes	**Después**	**Antes**	**Después**	
1	Nuevo	1	1	1	1	Alta
Descripción: Se establece a conexión a los servicios de Firebase con Android Studio						
Fecha		**Estado**		**Comentario**		
01/06/2023		Definido		Sin comentario		
02/06/2023		Implementado		Sin comentario		
05/06/2023		Ejecutado		Sin comentario		
05/06/2023		Verificado		Sin comentario		

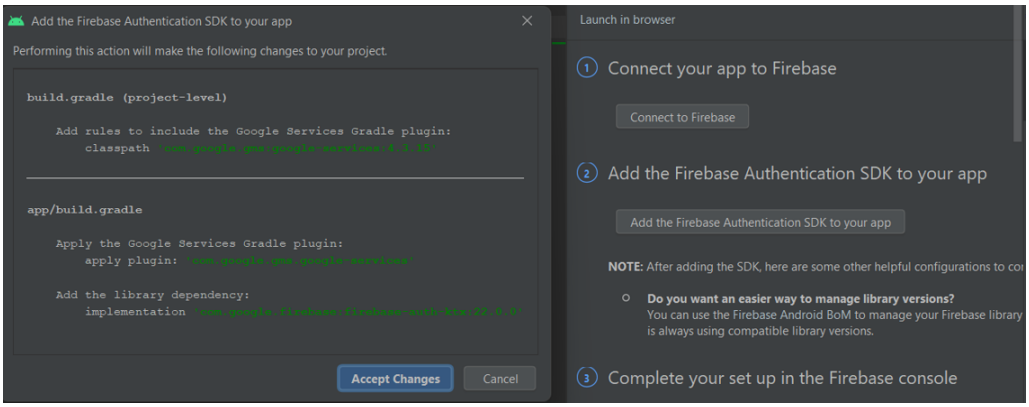


Figura 46. Conexión de Android Studio a Servicios de Firebase.

En la Figura 46 se presenta de manera sencilla cómo establecer la conexión entre los servicios de autenticación y la Realtime Database. Dado que ambos servicios pertenecen a la misma compañía, Google, cuentan con integraciones fluidas que permiten establecer conexiones con un solo clic, utilizando las cuentas compartidas. En Android Studio, la sección "Tools" proporciona un acceso directo a los servicios de Firebase, lo que facilita la configuración de conexiones de manera eficiente.

StoryCard – Diseño interfaz de Inicio y Registro de usuarios

Tabla 6. StoryCard - Diseño de inicio de Sesión y Registro de usuarios.

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
2	Nuevo	2	2	2	2	Alta
Descripción: Se realiza el diseño de Inicio de Sesión y Registros de Usuarios						
Fecha		**Estado**		**Comentario**		
01/06/2023		Definido		Sin comentario		
07/06/2023		Implementado		Sin comentario		
08/06/2023		Ejecutado		Sin comentario		
09/06/2023		Verificado		Sin comentario		

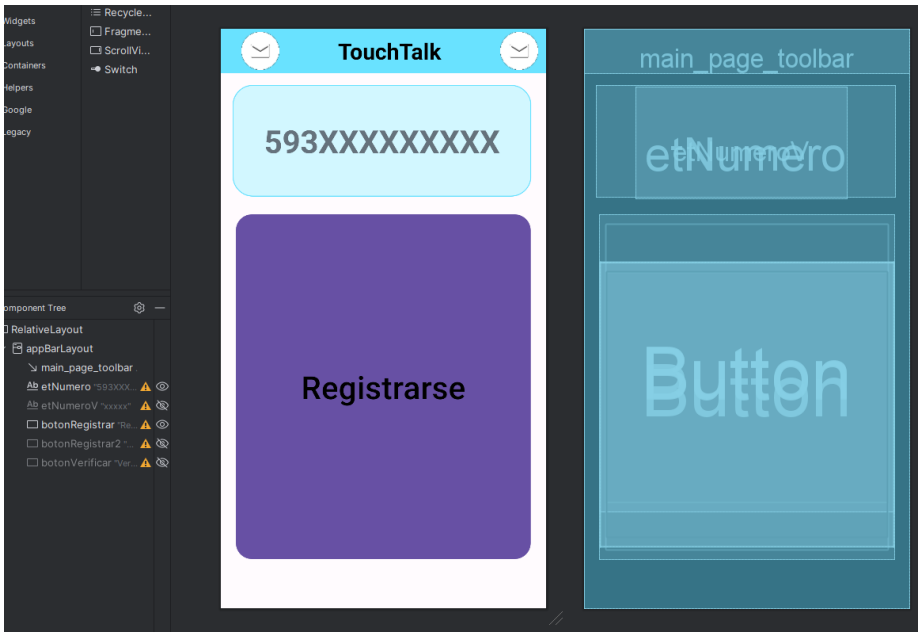


Figura 47. Diseño de Registro e Inicio de Sesión.

En la Figura 47 se presenta un único botón que posibilitará ingresar el número de teléfono mediante comandos de voz. Este número se asignará al campo de texto como referencia para los usuarios. Además, el número ingresado se reproducirá por voz como un medio de verificación o para su corrección, garantizando así la precisión en la entrada de datos. Esta implementación no solo simplifica el proceso de ingreso de información, sino que también mejora la experiencia del usuario al proporcionar una verificación auditiva del número ingresado.

StoryCard – Conexiones de autenticación

Tabla 7. StoryCard - Conexiones de autenticación.

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
3	Nuevo	3	3	3	3	Alta
Descripción: Implementar conexiones de autenticación con Firebase para logueo y registro del usuario						
Fecha		**Estado**		**Comentario**		
01/06/2023		Definido		Sin comentario		
13/06/2023		Implementado		Sin comentario		
14/06/2023		Ejecutado		Sin comentario		
15/06/2023		Verificado		Sin comentario		

La aplicación requirió una verificación a través de la huella digital del certificado para permitir que el método PhoneAuthProvider.verifyPhoneNumber determine el ingreso correcto de un número como se muestra en el código del Anexo 1. Además, se debió delimitar el código de país según el número de teléfono celular y hacer uso de PhoneAuthProvider. Ver Anexo 2.

StoryCard – Verificación Automática leyendo mensaje OTP

Tabla 8. StoryCard – Verificación Automática por mensaje OTP.

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
4	Nuevo	4	4	4	4	Alta
Descripción: Verificar automáticamente un número celular con la ayuda de detectar el código de un mensaje OTP						
Fecha		**Estado**		**Comentario**		
01/06/2023		Definido		Sin comentario		
16/06/2023		Implementado		Sin comentario		
19/06/2023		Ejecutado		Sin comentario		
20/06/2023		Verificado		Sin comentario		

Continuando con el proceso anterior, se procede a llamar a las instancias OnVerificationStateChangedCallbacks, que ayudarán con el manejo de los resultados de la solicitud. Se implementarán tres métodos adicionales: onVerificationCompleted, onVerificationFailed y onCodeSent. Estos métodos se ejecutan en un orden específico; primero, se verifica que el número sea correcto, luego se procede al envío del código de verificación y, si este es válido, se realiza el inicio de sesión del usuario, como se muestra en el Anexo 3.

Antes de la verificación de la corrección del número, se realizó un control para asegurar que solo se ingresen números con la extensión válida de Ecuador, añadiéndoles el prefijo +593.

StoryCard – Almacenar Datos en Realtime Database

Tabla 9. StoryCard - Almacenar Datos en Realtime Database.

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
5	Nuevo	3	3	3	3	Alta
Descripción: Enviar datos del usuario en la aplicación a través del servicio Realtime Database						
Fecha		**Estado**		**Comentario**		
01/06/2023		Definido		Sin comentario		
21/06/2023		Implementado		Sin comentario		
22/06/2023		Ejecutado		Sin comentario		
23/06/2023		Verificado		Sin comentario		

Inmediatamente después de que el usuario inicie sesión y esté a punto de acceder al sistema, los datos del usuario se agregan a una colección llamada "Users", que contiene información como el ID del usuario, el número de celular y el token del dispositivo. Todo esto se implementa en el método "ActualizarDatos". Para acceder al ID del usuario, se utiliza "getCurrentUser"; para extraer el token del dispositivo, es necesario haber implementado previamente el servicio de Cloud Messaging y extraerlo con el método "getToken", como se muestra en el código del Anexo 4.

Una vez obtenida esta información, se procede a crear una estructura de tipo HashMap, que hace referencia a clave-valor, para enviar estos datos a la base de datos en Firebase mediante el método "put". Ver Anexo 5.

StoryCard – Contactos Celular y Usuarios de Firebase

Tabla 10. StoryCard - Contactos Celular y Usuarios de Firebase.

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
6	Nuevo	3	3	3	3	Alta
Descripción: Extraer los usuarios de Firebase y del dispositivo móvil y generar una comparación.						
Fecha		**Estado**		**Comentario**		
01/06/2023		Definido		Sin comentario		
26/06/2023		Implementado		Sin comentario		
27/06/2023		Ejecutado		Sin comentario		
28/06/2023		Verificado		Sin comentario		

Para acceder a los usuarios de Firebase, primero se debe establecer una referencia tipo Database Reference a la cual se le asigna un listener. Esto genera dos métodos: onDataChange y onCancelled. Previamente, se crea una lista y se vacía para posteriormente ingresar nuevos datos dentro de onDataChange. Se realiza una iteración a través de los nodos de la instantánea snapshot utilizando un bucle for para obtener datos como uid, número celular y token del dispositivo. Estos datos se utilizan para generar una lista de usuarios, excluyendo al usuario actualmente logueado. Luego se invoca una función llamada compararUsuariosYContacto y se notifica a un adaptador ArrayAdapter que los datos han cambiado. Ver Anexo 6.

Para obtener una lista de los contactos del celular, se crea un conjunto (set) llamado numeroContactos para almacenar números de celular y evitar duplicados. Se utiliza un cursor para acceder a la lista de contactos del dispositivo, y previamente se deben permitir los permisos que solicite la aplicación. Por cada contacto, se limpia el nombre y el número de teléfono de caracteres no deseados, se ajusta el formato del número para incluir el código del país "593" si es necesario, y se crea una cadena que combina el nombre y el

número. Posteriormente, se notifica al adaptador que los datos han cambiado. Ver Anexo 7.

Finalmente, el método compararUsuariosYContactos compara las listas de contactos del dispositivo y la lista de usuarios almacenados, identificando coincidencias en relación con el número celular para evitar duplicaciones. Se crean dos listas: una que almacena la información combinada, incluyendo nombre, número de teléfono, identificador del usuario y token del dispositivo; y una lista adicional que enumera los nombres de los contactos para su posterior uso. Ver Anexo 8.

StoryCard – Comando Listar Contactos

Tabla 11. StoryCard - Comando Listar Contactos

N°	**Tipo**	**Dificultad**		**Esfuerzo**		**Prioridad**
		Antes	**Después**	**Antes**	**Después**	
7	Nuevo	3	3	3	3	Alta
Descripción: Generar un comando para listar los contactos en la aplicación a través de un comando de voz.						
Fecha		**Estado**		**Comentario**		
29/06/2023		Definido		Sin comentario		
30/06/2023		Implementado		Sin comentario		
03/07/2023		Ejecutado		Sin comentario		
04/07/2023		Verificado		Sin comentario		

Se registra un lanzador de actividad para el manejo de la entrada de voz, normalizando el contenido ingresado para identificar qué parte es comando y qué parte es información adicional. Para listar los contactos, se debe identificar el comando "Contactos" para así ingresar al if correspondiente. Al ingresar, se genera un StringBuilder llamado listaContactoText donde se almacena la información de contactos. Luego, se obtiene la

cantidad de contactos determinando el tamaño de lista_de_usuarios_app_nombres. Se utiliza la función TextToSpeech para informar al usuario sobre la cantidad total de usuarios en la aplicación, seguido de la lectura en voz alta de la lista de nombres de contactos que podrá usar para enviar mensajes o escuchar mensajes. Ver Anexo 9.

StoryCard – Comando envió de Mensajes

Tabla 12. StoryCard - Comando de Envío de Mensajes.

N°	**Tipo**	**Dificultad**		**Esfuerzo**		**Prioridad**
		Antes	**Después**	**Antes**	**Después**	
7	Nuevo	4	4	4	4	Alta
Descripción: Generar un comando para enviar mensajes en la aplicación a través de un comando de voz.						
Fecha		**Estado**		**Comentario**		
18/09/2023		Definido		Sin comentario		
19/09/2023		Implementado		Sin comentario		
20/09/2023		Ejecutado		Sin comentario		
21/09/2023		Verificado		Sin comentario		

Al igual que en el comando anterior, en el mismo lanzador de actividad se registra el comando de voz. Se utiliza el término "mensaje a" para determinar la acción que se debe realizar. Cuando se determina qué comando es, se divide la cadena ingresada para identificar qué parte corresponde a la información de contacto y cuál al mensaje a enviar. A través de un bucle, se compara el contacto ingresado por comando con la lista de usuarios de la aplicación, sin distinguir entre mayúsculas y minúsculas, como se muestra en el código del Anexo 10. Si encuentra el usuario, se continúa con el proceso. Utilizando la función TextToSpeech, se proporcionan instrucciones auditivas al usuario y se habilitan funcionalidades en la interfaz de la aplicación. Además, se manejan casos en los que el mensaje está vacío y se ofrecen ayudas verbales. Ver Anexo 11.

StoryCard – Comando reproducción de Mensajes

Tabla 13. Comando de reproducción de Mensajes

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
7	Nuevo	4	4	4	4	Alta
Descripción: Generar un comando para reproducir mensajes en la aplicación a través de un comando de voz.						
Fecha		**Estado**		**Comentario**		
22/09/2023		Definido		Sin comentario		
25/09/2023		Implementado		Sin comentario		
26/09/2023		Ejecutado		Sin comentario		
27/09/2023		Verificado		Sin comentario		

Todos los comandos se manejan con lanzadores de actividad para determinar qué actividad realizara según el comando de voz, siguiendo la lógica de enviar mensaje se determina la información del comando separándola en partes; la primera parte será la información del comando que en el caso será “Reproducir mensajes” y la otra parte será del contacto que se desee escuchar, con la información de contacto a través de un bucle se verifica que el contacto exista en el aplicativo sin distinguir mayúsculas y minúsculas, si encuentra el contacto activa el método obtenerListaDeMensajes a la cual se le asigna el uid del usuario logueado y el contacto encontrado, con estos datos se accede a la referencia de la base de datos Firebase en la ruta Messages/{yo}/{quienEnvio} donde “yo” es el usuario logueado y quienEnvio es el contacto encontrado de la lista del aplicativo. Luego se establece un escuchador de eventos para recibir actualizaciones en tiempo real de la base de datos, posterior a esto ejecuta el método reproducirMensajes que utiliza la función TextToSpeech para para informar al usuario la existencia de los mensajes.

StoryCard – Interfaz de envío y recepción de mensajes

Tabla 14. StoryCard - Interfaz para envío y recepción de mensajes.

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
7	Nuevo	2	2	2	2	Alta
Descripción: Crear una interfaz para el envío y recepción de mensajes.						
Fecha		**Estado**		**Comentario**		
28/09/2023		Definido		Sin comentario		
29/09/2023		Implementado		Sin comentario		
02/10/2023		Ejecutado		Sin comentario		
03/10/2023		Verificado		Sin comentario		

Dado que la aplicación está diseñada principalmente para personas con discapacidad visual, la interfaz es sencilla, evitando al máximo la manipulación de opciones en pantalla. Todo el proceso se gestiona a través de un único botón, encargado tanto del envío como de la recepción de mensajes. Adicionalmente, para aquellas personas con discapacidad visual que cuentan con familiares capaces de manipular la aplicación de manera convencional, se ha añadido un pequeño campo de texto en la parte inferior del botón. Este campo proporciona información sobre las opciones y datos que se están gestionando. La persona con discapacidad visual solo deberá tocar el centro de la pantalla para manejar las opciones.

StoryCard – Funcionalidades de configuración

Tabla 15. StoryCard - Funcionalidades de configuración

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
7	Nuevo	4	4	4	4	Alta
Descripción: Crear opciones para modificar las ayudas verbales y el número de mensajes a reproducir.						
Fecha		**Estado**		**Comentario**		
04/10/2023		Definido		Sin comentario		
05/10/2023		Implementado		Sin comentario		
06/10/2023		Ejecutado		Sin comentario		
09/10/2023		Verificado		Sin comentario		

Mediante comandos para desactivar ayudas, se logra reducir la longitud de los mensajes de indicaciones, mejorando así el tiempo de ejecución de los procesos al enviar mensajes. Esto se consigue al asignar el comando 'desactivar', el cual activa un booleano que se envía a la actividad de enviar mensajes y determina la longitud de los mensajes de ayuda. La otra configuración es la cantidad de mensajes que se reproducen al listarlos, estando predeterminadamente asignado solo 1. El usuario puede establecer cuántos mensajes desea que se reproduzcan mediante el comando 'Número de mensajes', que puede ir desde 1 hasta 5. Ver Anexo 12.

StoryCard – Interfaz de configuraciones

Tabla 16. StoryCard - Interfaz de configuraciones.

N°	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Antes	Después	
7	Nuevo	3	3	3	3	Alta
Descripción: Crear opciones para modificar las ayudas verbales y el número de mensajes a reproducir.						
Fecha		**Estado**		**Comentario**		
10/10/2023		Definido		Sin comentario		
11/10/2023		Implementado		Sin comentario		
12/10/2023		Ejecutado		Sin comentario		
15/10/2023		Verificado		Sin comentario		

Con el objetivo de facilitar el acceso a las configuraciones, se han creado secciones desplazables hacia la derecha. Estas secciones incluyen una dedicada a las configuraciones, la cual consta de un botón, y otra para el envío de mensajes, que ocupa gran parte de la pantalla a partir de su centro. Asimismo, se ha incorporado un campo de texto que indica las configuraciones para aquellas personas que cuentan con las capacidades visuales necesarias para utilizar la aplicación. Ver Anexo 13.

3.2.4 Fase 4 Estabilización

Esta etapa se enfoca en rectificar fallos, realizar ajustes en el diseño y mejorar la experiencia del usuario con el objetivo de garantizar la confiabilidad de la aplicación.

Modificaciones en la base de datos:

- Incorporación de la colección 'Configuración' para almacenar el estado de las configuraciones por usuario, de manera que estas queden asignadas una vez sean modificadas.

Modificaciones en el código:

- Se han añadido órdenes similares a las de enviar mensajes y reproducción de mensajes para facilitar el manejo de la aplicación.
- Se ha simplificado el lenguaje de las órdenes, eliminando palabras innecesarias.

3.2.5 Fase 5 Pruebas

Esta fase se centra en llevar a cabo pruebas en la aplicación móvil con el objetivo de mejorar su rendimiento, ajustar el diseño y la experiencia del usuario, con el fin de ofrecer una versión estable y plenamente funcional.

a. Pruebas de funcionalidad

La realización de esta prueba reviste una importancia fundamental, ya que su propósito es asegurar que la aplicación móvil satisfaga de manera efectiva los requisitos funcionales previamente establecidos. Este proceso es esencial para garantizar que la aplicación cumpla con las expectativas y funcionalidades requeridas.

- Pantalla inicial o Registro

Directamente, la aplicación abre un registro sencillo que consta de un 'EditText' que muestra el número ingresado y un botón que permite realizar funciones mediante comandos de voz, como se indica en la Figura 48. Se presiona la pantalla o botón y se ingresa el número celular.

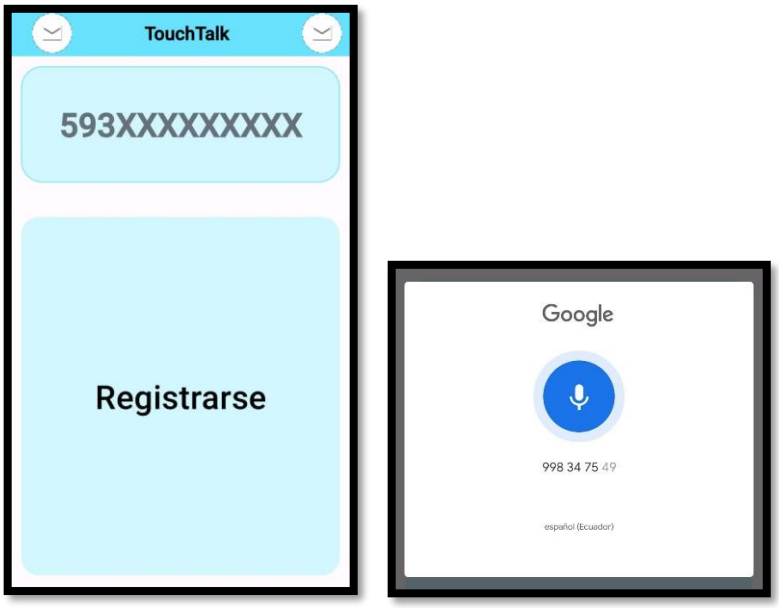


Figura 48. Registro e Inicio de Sesión.

Cuando se ingresa un número de teléfono celular válido que cumple con la cantidad de dígitos requerida, las funciones del botón cambiarán, permitiendo que se confirme o se repita el número, según sea necesario. Este comportamiento se indica en la **Figura 49**..

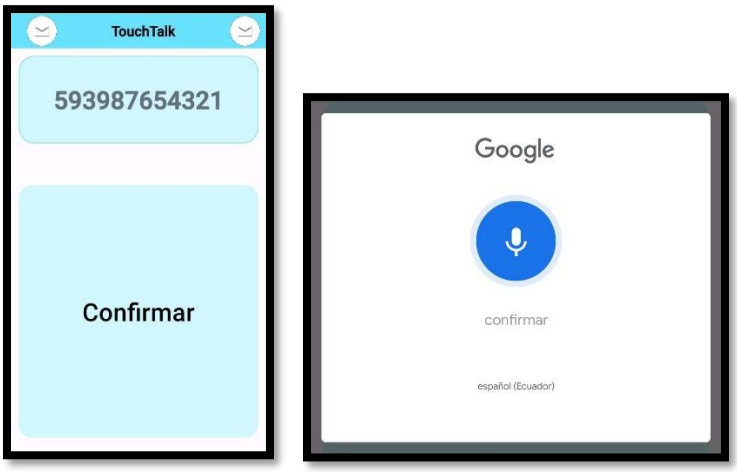


Figura 49. Confirmar el número con que se registra el usuario.

Posteriormente, se verifica si el número es válido o no. En caso de ser correcto, se procede a realizar una verificación automática donde el campo se rellena de forma autónoma. Si

todo es correcto, el usuario es dirigido a la pantalla principal de la aplicación, el proceso se aprecia en la Figura 50.

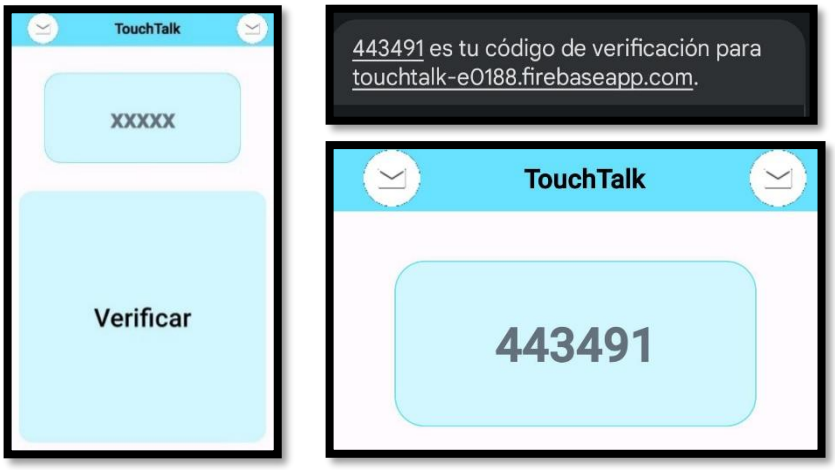


Figura 50. Verificación Automática.

- Pantalla principal Mensajes

Al acceder al aplicativo, se presentará un sistema de desplazamiento entre el sistema de mensajería y las configuraciones. Puede accederse deslizando a la derecha para ver las configuraciones y a la izquierda para acceder a los mensajes. Dentro de este apartado, se encuentra un único botón que se encarga del envío, reproducción de mensajes e información de contactos. En la parte inferior del botón, se encuentran textos que indican información sobre lo que se está ejecutando. Se realizó previamente un intento de envío a un usuario que no existe, por lo que el mensaje informó que no se encontró el usuario en la aplicación, como se puede apreciar en la Figura 51.

Figura 51. Pantalla Principal (Mensajes/Configuraciones).

Para conocer los contactos en el aplicativo se presiona la pantalla y se menciona el comando “CONTACTOS” el aplicativo brindara una información sonora sobre cuantos usuarios tiene disponible y cuales son cada uno. Adicionalmente se publican textos con la información requerida Esto se muestra en la Figura 52.

Figura 52. Comando Contactos en Ejecución

Si se realiza el envío de un mensaje, se habilita otro botón con opciones para confirmar, repetir o cancelar el mensaje, brindando así la oportunidad al usuario de verificar la transmisión de su mensaje. Cada acción se refleja también en textos ubicados en la parte inferior, proporcionando apoyo visual para aquellos usuarios que utilizan la aplicación y desean tener una indicación clara de la acción que están realizando. Este proceso se ilustra en la Figura 53.

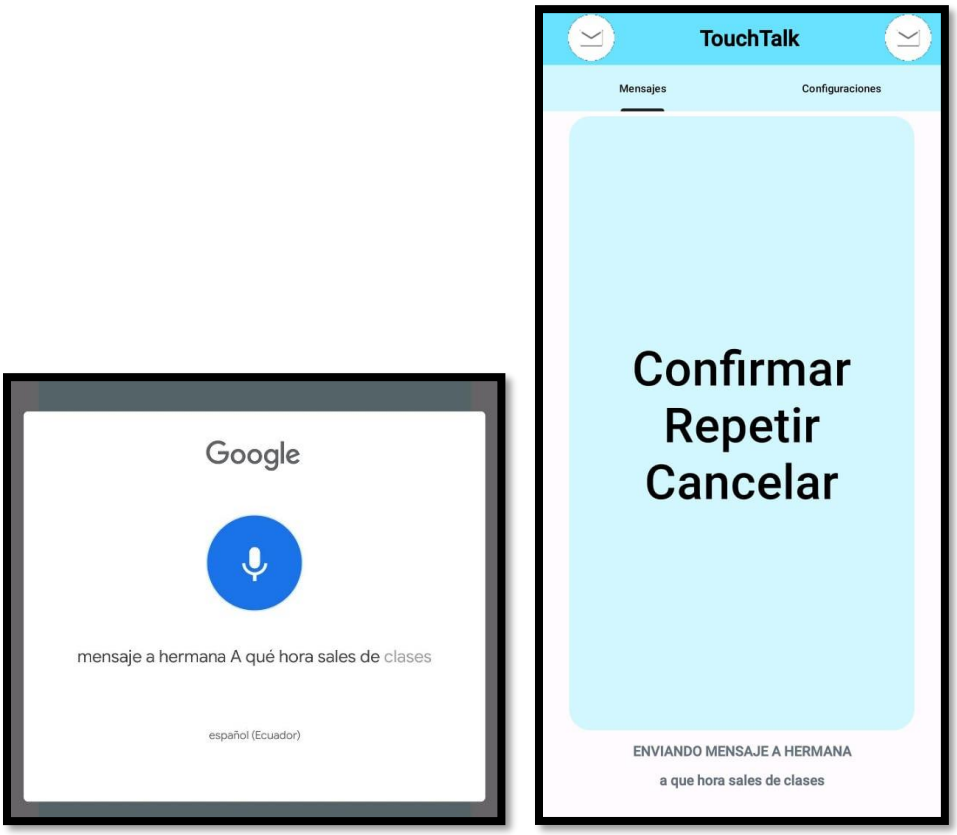


Figura 53. Pantalla Confirmación de Mensaje.

- Reproducción de Mensajes

Para escuchar mensajes de usuarios de quienes se dispone en el aplicativo se puede utilizar los comandos “REPRODUCIR MENSAJES DE”, “ESCUCHAR MENSAJES DE” o simplemente “MENSAJES DE” adicional a estos comandos se menciona algún contacto existente para escuchar los mensajes. Como se muestra en la Figura 54.

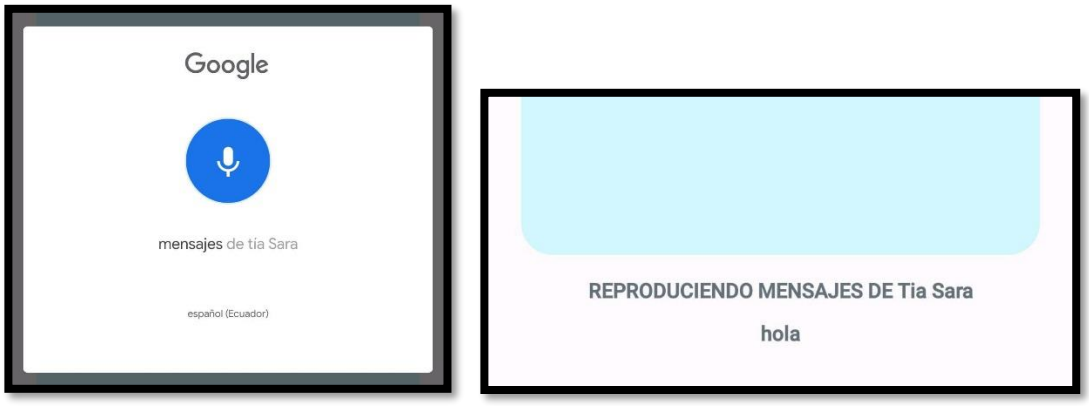


Figura 54. Reproduciendo Mensajes

- Configuraciones

En el apartado de configuraciones, también existe un único botón que permite configurar dos opciones en la aplicación: simplificar las ayudas auditivas y ajustar el número de mensajes por usuario según la preferencia del usuario. Esto se muestra en la Figura 55 Los comandos son “ACTIVAR”,”DESACTIVAR” o un numero entre 1 y 5 para establecer la cantidad de mensajes a escuchar..

Figura 55. Pantalla de Configuraciones

b. Pruebas de Aceptación

Según las pruebas realizadas en el aplicativo móvil se obtuvo como resultado las siguientes pruebas de aceptación.

Tabla 17. Prueba de aceptación 1

Prueba de aceptación	**Numero: 1**
Nombre: Registro de Usuario e Inicio de Sesión	
Descripción: Por medio de comandos de voz y respuestas el usuario puede registrarse y acceder a la aplicación.	
Ejecución: Se visualiza un único botón y un campo que muestra la información ingresada. Al ejecutarse la aplicación se da indicaciones sobre los pasos a realizar, El botón permite ingresar un numero celular, posteriormente con comando se confirma el número, y se verifica que exista automáticamente con un código OTP para acceder al sistema,	
Resultado Esperado: Los datos se validan, numero celular código OTP y se guardan datos del usuario como ID generado, numero celular y token device en Realtime Database de Firebase	
Resultado de la prueba: Prueba Satisfactoria	

Tabla 18. Prueba de aceptación 2

Prueba de aceptación	**Numero: 2**
Nombre: Listar Contactos	
Descripción: Por medio de comandos de voz y respuestas el usuario puede acceder a información de usuarios con los que se puede comunicar	
Ejecución: Al presionar la pantalla o botón que ocupa en su espacio, se menciona el comando "Contactos" y se brinda una respuesta al usuario a través de voz indicando los contactos que posee.	
Resultado Esperado: La aplicación brinda la información requerida al usuario listando en cantidad y detalle el nombre de contactos disponibles.	
Resultado de la prueba: Prueba Satisfactoria	

Tabla 19. Prueba de aceptación 3

Prueba de aceptación	**Número: 3**
Nombre: Envío de Mensajes	
Descripción: Por medio de comandos de voz y respuestas el usuario puede enviar mensajes a usuarios que se encuentren en la lista del aplicativo.	
Ejecución: Al presionar la pantalla o botón que ocupa en su espacio, se menciona el comando "Mensaje a" o "Enviar a", adicional se añade un nombre de usuario e inmediatamente el mensaje a enviar, si todo es correcto el aplicativo permite confirmar, repetir o cancelar el mensaje de igual manera con comandos de voz. Cuando se confirma el mensaje se envía al usuario respectivo. Y se le notifica.	
Resultado Esperado: La aplicación brinda la información requerida al usuario de pasos para enviar un mensaje y la información se guarda en la base de datos de Firebase.	
Resultado de la prueba: Prueba Satisfactoria	

Tabla 20. Prueba de aceptación 4

Prueba de aceptación	**Número: 4**
Nombre: Reproducción de Mensajes	
Descripción: Por medio de comandos de voz y respuestas el usuario puede reproducir mensajes recibidos de usuarios en el aplicativo.	
Ejecución: Al presionar la pantalla o botón que ocupa en su espacio, se menciona el comando "Reproducir Mensajes de", "Escuchar mensajes de" o simplemente "Mensajes de", adicional se menciona un nombre de usuario. Si existen mensajes re reproducirán por voz y si no existen se brindará un mensaje afirmando "No existen mensajes".	
Resultado Esperado: La aplicación brinda la información requerida al usuario de pasos para reproducir un mensaje extrayéndolos de la base da datos.	
Resultado de la prueba: Prueba Satisfactoria	

Tabla 21. Prueba de aceptación 5

Prueba de aceptación	**Número: 5**
Nombre: Configuración de ayudas y número de mensajes a reproducir	
Descripción: Por medio de comandos de voz y respuestas el usuario puede cambiar configuraciones del sistema como extensión de mensajes de ayudas y cantidad de mensajes que reproducirán de los usuarios.	
Ejecución: Al presionar la pantalla o botón que ocupa en su espacio, se menciona el comando "Activar" o "Desactivar" y procederá a cambiar la información que permite reproducir mensajes cortos o extensos de ayuda en el aplicativo. También si se menciona un numero entre 1 y 5 se determinará la cantidad de mensajes que se reproducirán cuando se acceda a mensajes de usuarios.	
Resultado Esperado: La aplicación brinda la información requerida al usuario de pasos para configurar el sistema según sus requerimientos.	
Resultado de la prueba: Prueba Satisfactoria	

Tiempos de Interacción

Como muestra la Tabla 22 se proporciona información sobre el tiempo que tres usuarios diferentes tardan en realizar diversas acciones en la aplicación. Las acciones incluyen Registro, búsqueda de contactos, envío y recepción de mensajes y configuración del sistema.

Tabla 22.Tiempo de Interacción

Tarea Acción	**Usuario 1**	**Usuario 2**	**Usuario 3**
Registro e Inicio de sesión	55 segundos	63 segundos	61 segundos
Búsqueda de Contactos	9 segundos	8 segundos	9 segundos
Envío de Mensaje	14 segundos	16 segundos	21 segundos
Recepción de Mensaje	5 segundos	5 segundos	6 segundos
Configuración de Sistema	5 segundos	5 segundos	4 segundos

Eficiencia de Navegación

La Tabla 18 proporciona información sobre las tareas o secciones de navegación en la aplicación, incluyendo el número de pasos necesario para completar cada tarea y cantidad de toques requeridos.

Tabla 23. Eficiencia de Navegación

Tarea o Sección de Navegación	Pasos	Toques en pantalla
Registro e Inicio de sesión	4	2
Búsqueda de Contactos	1	1
Envío de Mensaje	2	2
Reproducción de Mensajes	1	1
Configuración de Sistema	2	1

Evaluación de Accesibilidad

La Tabla 19 indica que la aplicación presenta un desempeño aceptable en aspectos claves. El uso de lectores de pantalla es un adicional, aunque no completamente necesario ya que el aplicativo incluye mensajes de voz integrados, además que el tamaño de botón esta optimizado para tener toques precisos, mejorando la experiencia del usuario.

Tabla 24. Evaluación de Accesibilidad

Aspecto de Accesibilidad	Respuesta
Uso de lectores de pantalla	Funciona Correctamente
Colores	Aceptable
Imágenes	Sin Imágenes
Tamaño de botones	Optimizado para toques precisos

Errores

Tabla 25. Errores

Tipo de Error o Problema	Versión Previa	Versión Final
Verificación Automática	Se necesitaba ayuda de tercero	El aplicativo detecta el mensaje y verifica al usuario
Mensajes Extensos y Confusos	Presente en algunas opciones	Se pueden modificar a gusto del usuario para agilizar procesos.
FeedBack Táctil	Proporción Tamaño del Botón	Se ocupa la mayoría de la pantalla disponible
Cierres inesperados de la aplicación	Problemas ocasionales en ciertas situaciones	Resuelto en la versión final
Problemas de Conectividad	Se necesita conexión a internet	Se necesita conexión a internet

CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Durante la recopilación y revisión de las limitaciones y necesidades de personas con discapacidad visual, se concluye que enfrentan numerosos desafíos durante la comunicación a través de dispositivos móviles. La mayoría de las aplicaciones de uso diario carecen de soluciones para sus limitaciones, como la inclusión de lectores de pantalla, comandos de voz variados, retroalimentación auditiva e interfaces sencillas. La adaptabilidad es fundamental para garantizar la independencia y privacidad de los usuarios con discapacidad visual**.**

- La herramienta Android Speech Recognition permite implementar el manejo por voz, gracias a su integración nativa que garantiza compatibilidad con una amplia variedad de dispositivos. Esta API ofrece características extensas, como facilidad de uso, eficiencia en el uso de recursos y soporte continuo de Google. Adicionalmente, se implementó TextToSpeech para proporcionar información vocal al usuario**.**

- La aplicación se presenta principalmente como una alternativa para la comunicación de personas con discapacidad visual, quienes han experimentado una mejora significativa en el tiempo de procesamiento para enviar o reproducir mensajes de usuarios específicos**.**

- La aplicación tiene un impacto positivo en la eficiencia de envío de mensajes, ya que reduce el tiempo necesario para realizar esta tarea, especialmente a medida que los usuarios se familiarizan con su uso diario. Además, se destaca la accesibilidad mejorada en la comunicación gracias a la implementación de la funcionalidad Text-to-Speech (TTS) en las notificaciones de mensajes. En conjunto, la aplicación no solo optimiza el proceso de envío de mensajes, sino que también fomenta la inclusión al proporcionar funcionalidades que garantizan una experiencia más equitativa para todos los usuarios.

4.2 Recomendaciones

- Para asegurar un rendimiento óptimo del aplicativo, se recomienda utilizar un dispositivo Android con una versión mínima de 8.0 Oreo.

- El acceso a Internet es fundamental, ya que el aplicativo utiliza los servicios de voz de Google, los cuales requieren conexión a Internet. Además, es necesario extraer información en tiempo real de Firebase, que está disponible en línea.

- Se recomienda simplificar al máximo los nombres de contacto en el dispositivo para facilitar su identificación mediante comandos de voz. La elección de nombres sencillos y fáciles de pronunciar agiliza y optimiza la ejecución de comandos relacionados con contactos. Utilizar nombres claros y concisos promueve una experiencia más intuitiva y sin complicaciones.

- Para maximizar la interacción de usuarios con discapacidad, se recomienda instalar la aplicación en la mayor cantidad de miembros familiares o allegados posibles. Esto contribuirá a fomentar la inclusión en las comunicaciones, promoviendo un entorno participativo y accesible.

REFERENCIAS BIBLIOGRÁFICAS

[1] T. R. Fricke et al., “Prevalencia global de presbicia y deterioro de la visión por presbicia no corregida: revisión sistemática, metaanálisis y modelado”, Ophthalmology, vol. 125, núm. 10, pp. 1492–1499, oct. 2018, doi: 10.1016/j.ophtha.2018.04.013.

[2] Consejo Nacional Para Igualdad de Discapacidades-Conadis, «Consejo Nacional Para Igualdad de Discapacidades,» Ministerio de Salud Pública, Enero 2022. [En línea]. Available: https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/. [Último acceso: 18 Enero 2023].

[3] C. P. Guachamin, “Implementación de un prototipo inalámbrico para la detección de obstáculos y señaléticas dirigido a personas con discapacidad visual”, 2020.

[4] B. M. Armijo, “Uso de las tic para el desarrollo de habilidades sociales en estudiantes con discapacidad visual”, 2018.

[5] S. S. Senjam, “Tecnología de asistencia para estudiantes con discapacidad visual: asuntos de clasificación”, 2019.

[6] R. Chicaiza, “El clima familiar y el autoestima de las personas con discapacidad visual de la unidad educativa especializada para no videntes ‘Julius Doepfner’ del cantón Ambato”, Universidad Técnica de Ambato, Ambato, 2020.

[7] G. del R. Alban, “Sistema domótico de apoyo para personas con discapacidad motriz mediante tecnología móvil y reconocimiento de voz.”, 2018.

[8] J. Cadena y J. Merino, “Desarrollo de una aplicación móvil para personas con discapacidad auditiva”, 2022.

[9] A. Nimmolrat, P. Khuwuthyakorn, P. Wientong, y O. Thinnukool, “Aplicación móvil farmacéutica para personas con discapacidad visual en Tailandia: desarrollo e implementación”, BMC Med Inform Decis Mak, vol. 21, núm. 1, dic. 2021, doi: 10.1186/s12911-021-01573-z.

[10] B. Armenta, I. Rodríguez, L. Medina, y S. González, “Aplicación del modelo de prototipos: Caso de estudio Software RedbotGamesShop”, 2018. [En línea]. Disponible en: www.ecorfan.org/taiwan

[11] F. Saenz-Blanco, F. Gutiérrez-Sierra, y J. C. Ramos-Rivera, “Conformación de equipos ágiles para el desarrollo de software: revisión de literatura”, 2020.

[12] K. Beck, “Extreme Programming Explained”, 1999

[13] E. Gómez del Campillo, “Estrategia internacional en la venta e integración de Slack en Salesforce”, 2022.

[14] R. A. Gómez-Peralta y J. A. Mora-Saltos, “Automatización de proceso para el manejo de ingreso a cuartos tecnológicos utilizando la herramienta jira software, para los edificios principales en guayaquil, quito y cuenca del banco guayaquil”, 2020.

[15] J. Astigarraga y V. Cruz-Alonso, “¡Se puede entender cómo funcionan Git y GitHub!”, Ecosistemas, vol. 31, núm. 1, ene. 2022, doi: 10.7818/ECOS.2332.

[16] Mobile Marketing Association, Libro Blanco de Apps. 2011.

[17] Developers Android, «Arquitectura de Plataforma,» Google Developer, [En línea]. Available: https://developer.android.com/guide/platform. [Último acceso: 19 Enero 2023].

[18] R. Martínez, A. Palma, y A. Velásquez, Revolución tecnológica e inclusión social Reflexiones sobre desafíos y oportunidades para la política social en América Latina 233 POLÍTICAS SOCIALES. 2020.

[19] A. R. Delgado-García, “Habilidades sociales en educación infantil”, 2017.

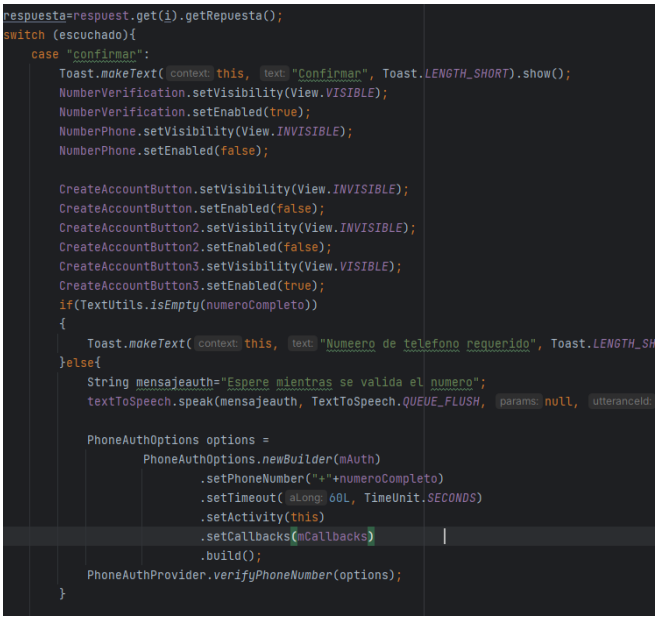
[20] D. S. Sierra, “Funcionalidad del sistema de señalización de calles y la orientación espacial de personas con discapacidad visual en la ciudad de Ambato”, 2022.

[21] M. Bonilla-del-Río, J. C. Figuereo-Benítez, y V. García-Prieto, “Influencers con discapacidad física en Instagram: características, visibilidad y negocio colaboración”, El Profesional de la información, nov. 2022, doi: 10.3145/epi.2022.nov.12.

[22] C. p. e. C. y. l. P. d. E. Centro Nacional de Defectos Congénitos y Discapacidades del Desarrollo de los CDC, «Centro Nacional de Defectos Congénitos y Discapacidades del Desarrollo de los CDC, Centros para el Control y la Prevención de Enfermedades,» 19 Mayo 2022. [En línea]. Available: https://www.cdc.gov/ncbddd/spanish/developmentaldisabilities/hoja-informativa-sobre-discapacidad-intelectual.html. [Último acceso: 19 Enero 2023].

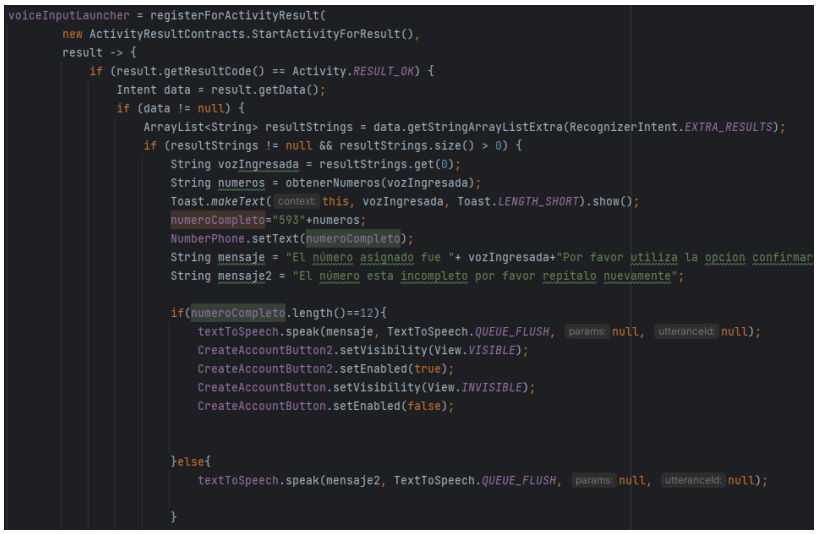
ANEXOS

Anexo 1. Verificación de un número correcto



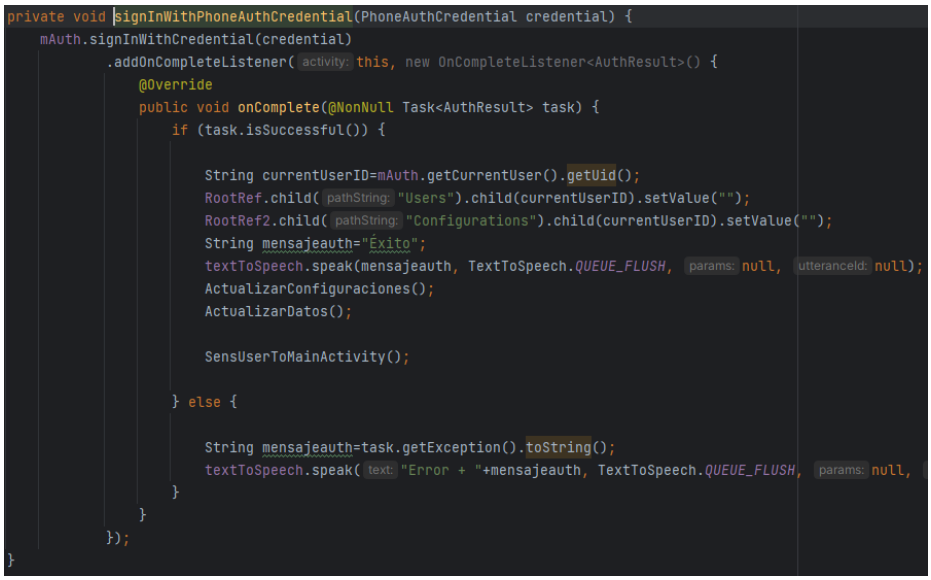
```
respuesta=respuest.get(i).getRepuesta();
switch (escuchado){
    case "confirmar":
        Toast.makeText( context: this,  text: "Confirmar", Toast.LENGTH_SHORT).show();
        NumberVerification.setVisibility(View.VISIBLE);
        NumberVerification.setEnabled(true);
        NumberPhone.setVisibility(View.INVISIBLE);
        NumberPhone.setEnabled(false);

        CreateAccountButton.setVisibility(View.INVISIBLE);
        CreateAccountButton.setEnabled(false);
        CreateAccountButton2.setVisibility(View.INVISIBLE);
        CreateAccountButton2.setEnabled(false);
        CreateAccountButton3.setVisibility(View.VISIBLE);
        CreateAccountButton3.setEnabled(true);
        if(TextUtils.isEmpty(numeroCompleto))
        {
            Toast.makeText( context: this,  text: "Numeero de telefono requerido", Toast.LENGTH_SH
        }else{
            String mensajeauth="Espere mientras se valida el numero";
            textToSpeech.speak(mensajeauth, TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId:

            PhoneAuthOptions options =
                    PhoneAuthOptions.newBuilder(mAuth)
                            .setPhoneNumber("+"+numeroCompleto)
                            .setTimeout( aLong: 60L, TimeUnit.SECONDS)
                            .setActivity(this)
                            .setCallbacks(mCallbacks)
                            .build();
            PhoneAuthProvider.verifyPhoneNumber(options);
        }
```

Anexo 2. Delimitación del código del país.



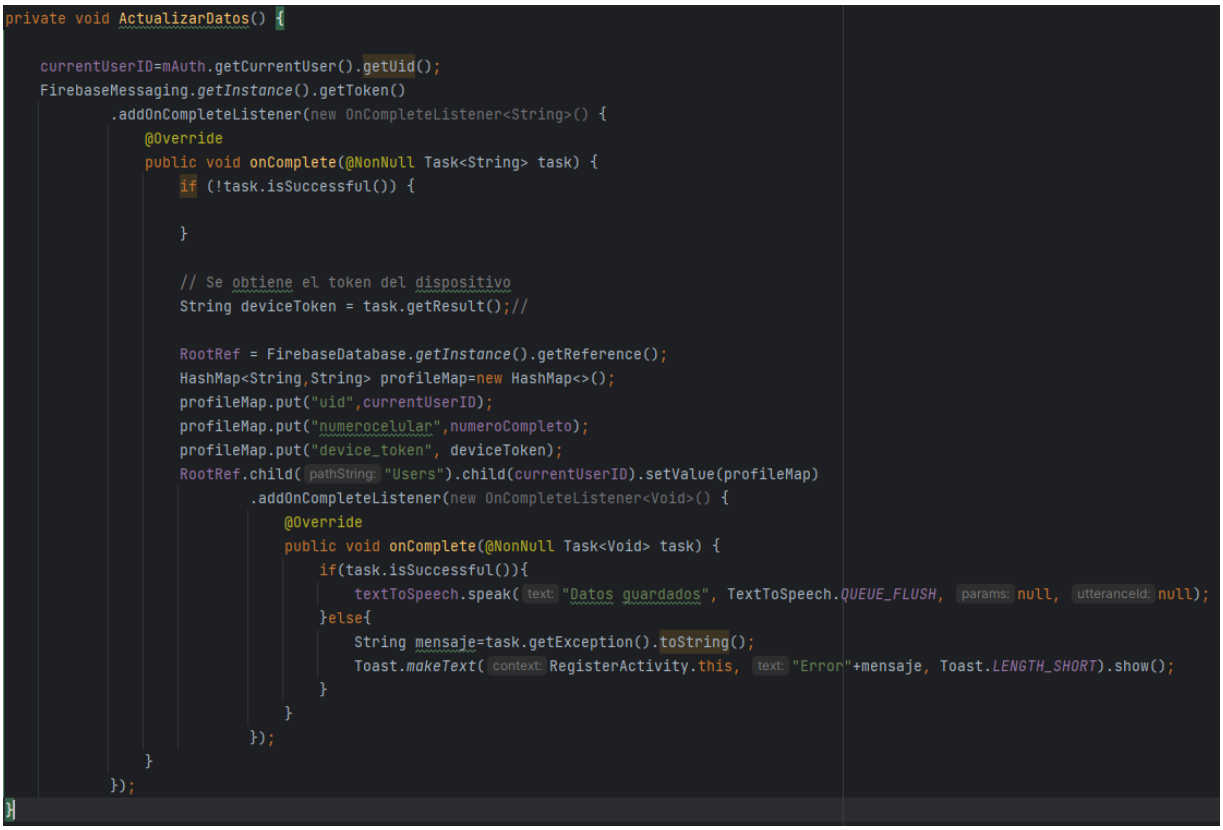
```
voiceInputLauncher = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        result -> {
            if (result.getResultCode() == Activity.RESULT_OK) {
                Intent data = result.getData();
                if (data != null) {
                    ArrayList<String> resultStrings = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    if (resultStrings != null && resultStrings.size() > 0) {
                        String vozIngresada = resultStrings.get(0);
                        String numeros = obtenerNumeros(vozIngresada);
                        Toast.makeText( context: this, vozIngresada, Toast.LENGTH_SHORT).show();
                        numeroCompleto="593"+numeros;
                        NumberPhone.setText(numeroCompleto);
                        String mensaje = "El número asignado fue "+ vozIngresada+"Por favor utiliza la opcion confirmar
                        String mensaje2 = "El número esta incompleto por favor repitalo nuevamente";

                        if(numeroCompleto.length()==12){
                            textToSpeech.speak(mensaje, TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId: null);
                            CreateAccountButton2.setVisibility(View.VISIBLE);
                            CreateAccountButton2.setEnabled(true);
                            CreateAccountButton.setVisibility(View.INVISIBLE);
                            CreateAccountButton.setEnabled(false);


                        }else{
                            textToSpeech.speak(mensaje2, TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId: null);

                        }
```

Anexo 3. Verificación y validación de código OTP.



```
private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {
    mAuth.signInWithCredential(credential)
            .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {

                        String currentUserID=mAuth.getCurrentUser().getUid();
                        RootRef.child( pathString: "Users").child(currentUserID).setValue("");
                        RootRef2.child( pathString: "Configurations").child(currentUserID).setValue("");
                        String mensajeauth="Éxito";
                        textToSpeech.speak(mensajeauth, TextToSpeech.QUEUE_FLUSH, params: null, utteranceId: null);
                        ActualizarConfiguraciones();
                        ActualizarDatos();

                        SensUserToMainActivity();

                    } else {

                        String mensajeauth=task.getException().toString();
                        textToSpeech.speak( text: "Error + "+mensajeauth, TextToSpeech.QUEUE_FLUSH, params: null,
                    }
                }
            });
}
```

Anexo 4. Actualizar Datos del usuario.



```
private void ActualizarDatos() {

    currentUserID=mAuth.getCurrentUser().getUid();
    FirebaseMessaging.getInstance().getToken()
            .addOnCompleteListener(new OnCompleteListener<String>() {
                @Override
                public void onComplete(@NonNull Task<String> task) {
                    if (!task.isSuccessful()) {

                    }

                    // Se obtiene el token del dispositivo
                    String deviceToken = task.getResult();//

                    RootRef = FirebaseDatabase.getInstance().getReference();
                    HashMap<String,String> profileMap=new HashMap<>();
                    profileMap.put("uid",currentUserID);
                    profileMap.put("numerocelular",numeroCompleto);
                    profileMap.put("device_token", deviceToken);
                    RootRef.child( pathString: "Users").child(currentUserID).setValue(profileMap)
                            .addOnCompleteListener(new OnCompleteListener<Void>() {
                                @Override
                                public void onComplete(@NonNull Task<Void> task) {
                                    if(task.isSuccessful()){
                                        textToSpeech.speak( text: "Datos guardados", TextToSpeech.QUEUE_FLUSH, params: null, utteranceId: null);
                                    }else{
                                        String mensaje=task.getException().toString();
                                        Toast.makeText( context: RegisterActivity.this, text: "Error"+mensaje, Toast.LENGTH_SHORT).show();
                                    }
                                }
                            });
                }
            });
}
```

Anexo 5. Envío de datos a Firebase.

```
RootRef.child( pathString: "Users").child(currentUserID).setValue(profileMap)
        .addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if(task.isSuccessful()){
                    textToSpeech.speak( text: "Datos guardados", TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId: null);
                }else{
                    String mensaje=task.getException().toString();
                    Toast.makeText( context: RegisterActivity.this,  text: "Error"+mensaje, Toast.LENGTH_SHORT).show();
                }
            }
        });
```

Anexo 6. Lista de Usuarios en Firebase.

```
private void RecuperarMostrarUsuariosFirebase() {
    UsuariosRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            lista_de_usuarios.clear();
            for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
                // Se accede al campo numero celular dentro del nodo users
                String numerocelular = dataSnapshot.child( path: "numerocelular").getValue(String.class);
                String uid = dataSnapshot.child( path: "uid").getValue(String.class);
                String device_token= dataSnapshot.child( path: "device_token").getValue(String.class);

                if (numerocelular != null && uid != null) {
                    // Verifica si el usuario seleccionado no es el logueado
                    if (!uid.equals(messageSenderID)) {
                        lista_de_usuarios.add(numerocelular + "\n" + uid+"\n"+device_token);
                    }
                }
                compararUsuariosYContactos();
            }

            arrayAdapter.notifyDataSetChanged();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    });
}
```

Anexo 7. Lista de Contactos en el dispositivo celular.

```
private void obtenerListaContactos() {
    Set<String> numerosContactos = new HashSet<>();
    Cursor cursor = requireActivity().getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
            projection: null, selection: null, selectionArgs: null, sortOrder: null);
    if (cursor != null) {
        while (cursor.moveToNext()) {
            @SuppressLint("Range") String name = cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME));
            @SuppressLint("Range") String mobile = cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER));

            name = limpiarNombres(name);
            name = name.replaceAll( regex: "[^a-zA-Z ]", replacement: "");
            mobile = mobile.replaceAll( regex: "[^0-9]", replacement: "");
            if (mobile.startsWith("0")) {
                // Eliminamos el cero y agregamos 593 a los numeros celulares para compararlos con los usuarios.
                mobile = "593" + mobile.substring( beginIndex: 1);
            }

            String contacto = name + "\n" + mobile;

            if (numerosContactos.add(mobile)) {
                lista_de_contactos.add(contacto);
            }
        }
        cursor.close();
    }
    arrayAdapter2.notifyDataSetChanged();
}
```

Anexo 8. Creación de lista comparativa entre usuarios Firebase y contactos del dispositivo.

```
private void compararUsuariosYContactos() {
    lista_de_aplicacion.clear();
    lista_de_usuarios_app_nombres.clear();
    Set<String> numerosAgregados = new HashSet<>();

    int contador = 1;

    for (String contacto : lista_de_contactos) {
        String[] parts = contacto.split( regex: "\n");
        String nombreContacto = parts[0];
        String numeroContacto = parts[1];

        for (String usuario : lista_de_usuarios) {
            String[] usuarioParts = usuario.split( regex: "\n");
            String numeroApp = usuarioParts[0];
            String uid = usuarioParts[1];
            String tokenContacto=usuarioParts[2];
            if (numeroContacto.equals(numeroApp) && !numerosAgregados.contains(numeroApp)) {

                lista_de_aplicacion.add(nombreContacto + "\n" + numeroContacto  + "\n" + uid+"\n"+tokenContacto);
                lista_de_usuarios_app_nombres.add(contador + ". " + nombreContacto);
                numerosAgregados.add(numeroContacto);
                contador++;
            }
        }
    }
    arrayAdapter3.notifyDataSetChanged();

}
```

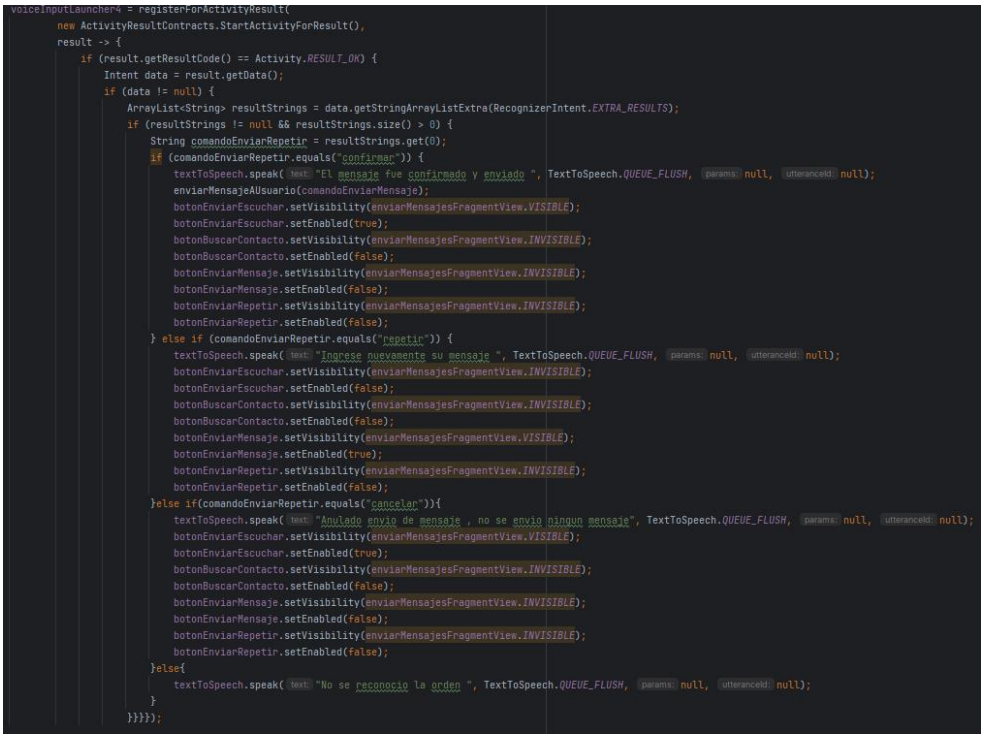
Anexo 9. Código de comando de contactos.

```
} else if(comandoPrincipal.startsWith("contactos")){
    // Realizar la acción correspondiente
    Toast.makeText(requireContext(), text: "buscando contactos", Toast.LENGTH_SHORT).show();
    System.out.println("Buscando contactos");
    StringBuilder listaContactosText = new StringBuilder("\n");
    int cantidadContactos=lista_de_usuarios_app_nombres.size();
    for (String nombreUsuario : lista_de_usuarios_app_nombres) {
        listaContactosText.append(nombreUsuario).append(". ");
    }
    textToSpeech.speak( text: "Usted tiene "+cantidadContactos+"Contactos en la aplicacion" +"Los contactos son: "+listaContactosText.toString(), TextToSpeech.QUEUE
    textoComando.setText("CONTACTOS");
    informacionComando.setText(listaContactosText);
```

Anexo 10. Comando para envío de mensajes.

```
voiceInputLauncher1 = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        result -> {
            if (result.getResultCode() == Activity.RESULT_OK) {
                Intent data = result.getData();
                if (data != null) {
                    ArrayList<String> resultStrings = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    if (resultStrings != null && resultStrings.size() > 0) {
                        String comandoEnviarEscuchar = resultStrings.get(0);
                        comandoEnviarEscuchar = Normalizer.normalize(comandoEnviarEscuchar, Normalizer.Form.NFD)
                                .replaceAll( regex: "\\p{InCombiningDiacriticalMarks}+", replacement: "");
                        String comandoPrincipal=comandoEnviarEscuchar.toLowerCase();
                        if (comandoPrincipal.startsWith("mensaje a")) {

                            String primeraParte = "mensaje a";

                            // Dividiendo comandoPrincipal
                            String[] palabras = comandoPrincipal.split( regex: " ");
                            boolean contactoEncontrado = false;
                            if (palabras.length >= 4) {
                                if (palabras[0].equalsIgnoreCase( anotherString: "mensaje") && palabras[1].equalsIgnoreCase( anotherString: "a")) {
                                    // Obtener la informacion del contacto
                                    String contacto2P = palabras[2] + " " + palabras[3];
                                    // Obtener el mensaje
                                    StringBuilder restoCadena = new StringBuilder();
                                    for (int i = 4; i < palabras.length; i++) {
                                        restoCadena.append(palabras[i]);
                                        if (i < palabras.length - 1) {
                                            restoCadena.append(" ");
                                        }
                                    }
                                    Toast.makeText(requireContext(), contacto2P, Toast.LENGTH_SHORT).show();
                                    Toast.makeText(requireContext(), restoCadena, Toast.LENGTH_SHORT).show();
                                    //Recorrer la lista de contactos
                                    for (String contacto : lista_de_aplicacion) {
                                        String[] parts = contacto.split( regex: "\n");
                                        nombreContactoE = parts[0];
                                        numeroContactoE = parts[1];
                                        uidE = parts[2];
                                        token_ContactoE = parts[3];
                                        if (contacto2P.equalsIgnoreCase(nombreContactoE.toLowerCase())) {
                                            Toast.makeText(requireContext(), text: "Número encontrado: " + numeroContactoE, Toast.LENGTH_SHORT).show();
                                            Toast.makeText(requireContext(), text: "uid Usuario: " + uidE, Toast.LENGTH_SHORT).show();
                                            contactoEncontrado = true;
                                            if (!restoCadena.toString().isEmpty()) {
                                                textToSpeech.speak( text: "El mensaje a enviar es el siguiente " + restoCadena, TextToSpeech.QUEUE_FLUSH, params: null, utteranceId: null);
                                                botonEnviarEscuchar.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                                                botonEnviarEscuchar.setEnabled(false);
                                                botonBuscarContacto.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                                                botonBuscarContacto.setEnabled(false);
                                                botonEnviarMensaje.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                                                botonEnviarMensaje.setEnabled(false);
                                                botonEnviarRepetir.setVisibility(enviarMensajesFragmentView.VISIBLE);
                                                botonEnviarRepetir.setEnabled(true);
                                                te                                                el mensaje con la palabra confirmar, repitalo con la palabra repetir o cancele el proceso con la palab
                                                te                                                SAJE A " + nombreContactoE.toUpperCase());
```

Anexo 11. Confirmación de mensaje con comando.



```
voiceInputLauncher4 = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        result -> {
            if (result.getResultCode() == Activity.RESULT_OK) {
                Intent data = result.getData();
                if (data != null) {
                    ArrayList<String> resultStrings = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    if (resultStrings != null && resultStrings.size() > 0) {
                        String comandoEnviarRepetir = resultStrings.get(0);
                        if (comandoEnviarRepetir.equals("confirmar")) {
                            textToSpeech.speak( text: "El mensaje fue confirmado y enviado ", TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId: null);
                            enviarMensajeAUsuario(comandoEnviarMensaje);
                            botonEnviarEscuchar.setVisibility(enviarMensajesFragmentView.VISIBLE);
                            botonEnviarEscuchar.setEnabled(true);
                            botonBuscarContacto.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                            botonBuscarContacto.setEnabled(false);
                            botonEnviarMensaje.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                            botonEnviarMensaje.setEnabled(false);
                            botonEnviarRepetir.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                            botonEnviarRepetir.setEnabled(false);
                        } else if (comandoEnviarRepetir.equals("repetir")) {
                            textToSpeech.speak( text: "Ingrese nuevamente su mensaje ", TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId: null);
                            botonEnviarEscuchar.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                            botonEnviarEscuchar.setEnabled(false);
                            botonBuscarContacto.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                            botonBuscarContacto.setEnabled(false);
                            botonEnviarMensaje.setVisibility(enviarMensajesFragmentView.VISIBLE);
                            botonEnviarMensaje.setEnabled(true);
                            botonEnviarRepetir.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                            botonEnviarRepetir.setEnabled(false);
                        }else if(comandoEnviarRepetir.equals("cancelar")){
                            textToSpeech.speak( text: "Anulado envio de mensaje , no se envio ningun mensaje", TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId: null);
                            botonEnviarEscuchar.setVisibility(enviarMensajesFragmentView.VISIBLE);
                            botonEnviarEscuchar.setEnabled(true);
                            botonBuscarContacto.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                            botonBuscarContacto.setEnabled(false);
                            botonEnviarMensaje.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                            botonEnviarMensaje.setEnabled(false);
                            botonEnviarRepetir.setVisibility(enviarMensajesFragmentView.INVISIBLE);
                            botonEnviarRepetir.setEnabled(false);
                        }else{
                            textToSpeech.speak( text: "No se reconocio la orden ", TextToSpeech.QUEUE_FLUSH,  params: null,  utteranceId: null);
                        }
                    }}}});
```

Anexo 12. Configuraciones

```
public class ConfiguracionesFragment extends Fragment implements TextToSpeech.OnInitListener {

    6 usages
    private TextToSpeech textToSpeech;

    1 usage
    private Button botonConfigiraciones;
    5 usages
    private View configuracionesFragmentView;
    no usages
    public String comando;
    no usages
    private ActivityResultLauncher<Intent> voiceInputLauncher1;
    1 usage
    private EditText ayuda,cantidad;


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

        configuracionesFragmentView = inflater.inflate(R.layout.fragment_configuraciones, container,
        textToSpeech = new TextToSpeech(requireContext(),  listener: this);

        inicializarCampos();


        return configuracionesFragmentView;
    }
```

Anexo 13. Código de opciones desplazables

```
public class TabsAccessorAdapter extends FragmentStateAdapter {

    1 usage
    public TabsAccessorAdapter(@NonNull FragmentActivity fragmentActivity) {
        super(fragmentActivity);
    }
    1 usage
    @NonNull
    @Override
    public Fragment createFragment(int position) {
        switch (position)
        {
            case 0:
                EnviarMensajesFragment enviarMensajesFragment = new EnviarMensajesFragment();
                return enviarMensajesFragment;
            case 1:
                ConfiguracionesFragment configuracionesFragment= new ConfiguracionesFragment();
                return configuracionesFragment;
        }
        return null;
    }

    @Override
    public int getItemCount() { return 2; }

    no usages
    @Nullable
    public CharSequence getPageTitle(int position)
    {
        switch (position)
        {
            case 0:
                return "Enviar Mensajes";
            case 1:
                return "Configuraciones";
        }
        return null;
    }
}
```

Anexo 14. Manual de usuario

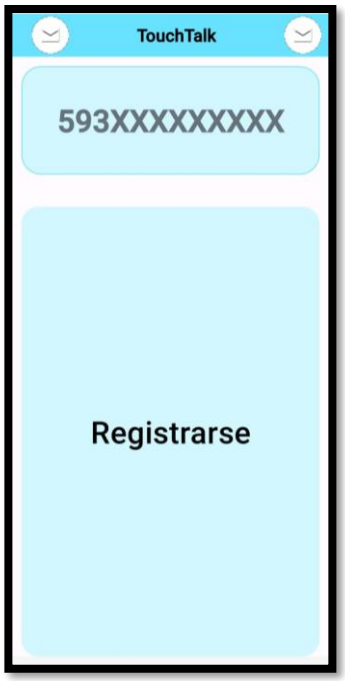
Manual de usuario

“APLICACIÓN MÓVIL BASADA EN ANDROID, PARA FACILITAR LA COMUNICACIÓN DE PERSONAS CON DISCAPACIDAD VISUAL”

Manual de usuario

El presente manual pretende brindar una guía del uso del aplicativo móvil destinado a personas con discapacidad visual quienes quieran probar una nueva alternativa para el envío de mensajes. Este manual tiene el objetivo de facilitar las instrucciones para el uso correcto del aplicativo móvil.

Pantalla Registro e Inicio de Sesión



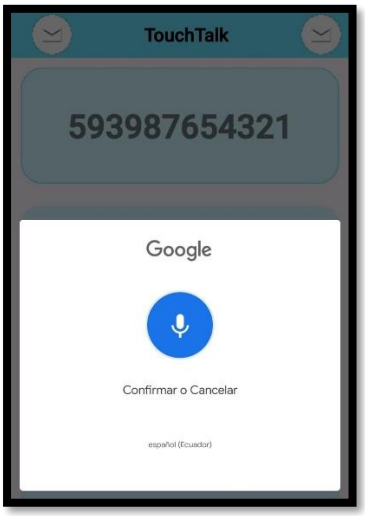
Al ingresar al aplicativo brindara un mensaje de bienvenida posteriormente con el siguiente mensaje por voz.
"Para empezar proporcione su número celular omitiendo el primer cero al presionar la pantalla "

Luego se presiona la pantalla o botón que cubre casi en su totalidad el espacio y se asigna el número por orden de voz por ejemplo "987654321"

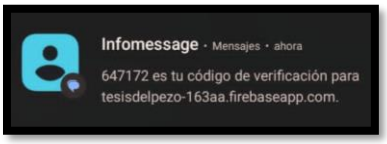
Si el número que se dicta es menor a 9 dígitos el programa indica que el número esta incompleto y menciona que repita nuevamente el número, hasta asignar uno completo y pueda seguir con el proceso

Al digitar un número correcto el programa menciona automáticamente el mensaje "El número asignado fue 593XXXXXXXXX por favor utilice la opción CONFIRMAR o CANCELAR para continuar el proceso"

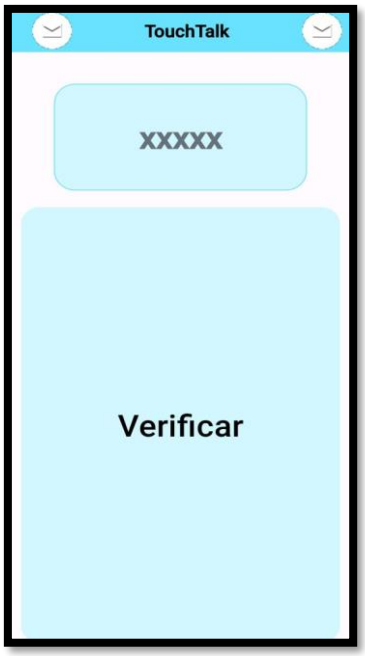
La opción "CANCELAR" permite rectificar el número siendo el caso de que si se ingresó algún digito incorrecto y permite reiniciar el proceso de asignación de número celular.



Se confirma el número por comando de voz, entonces se continua al siguiente paso para que el usuario se registre en la aplicación.



Se inicia un nuevo proceso que se ejecuta de manera automática. El programa proporciona dos nuevos mensajes de voz como guía para el usuario. El primer mensaje informa: "Se procederá con el proceso de registro; por favor, espere un momento". Si el número ingresado es válido y existe, se indicará: "Número verificado, se procederá con la verificación automática". De esta manera, el programa se verifica de forma autónoma, simplemente esperando unos momentos



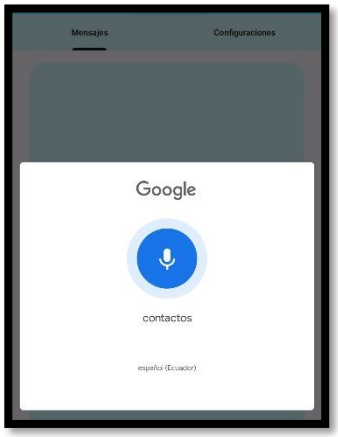
Llega un mensaje del tipo OTP que incluye un código, el código se ingresara automáticamente al aplicativo para una verificación automática y con esto se completa el proceso de registro del usuario en la aplicación. El programa indica con el mensaje por voz de “Éxito” cuando todo el registro culmine.

Nota: El programa solicita permisos al usuario para que funcione con normalidad como acceso a mensajes y contactos.

Pantalla Mensajes/Configuración

Envío de Mensajes

El sistema principal reproduce un mensaje que indica en donde se ubica el usuario, en el primer caso dirá “MENSAJES”, existen varios comandos que funcionan en este apartado al presionar el único botón existente se menciona el comando “CONTACTOS” permitirá que el programa nos indique que cuantos contactos existen en la aplicación y el nombre de cada uno de ellos.

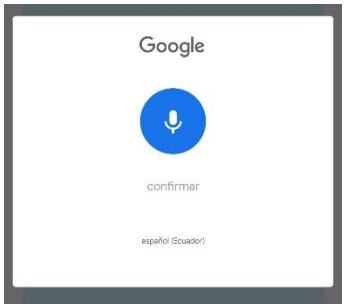


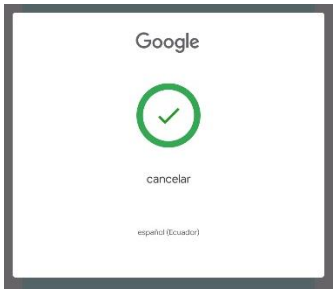
Conociendo los contactos de la aplicación, se puede saber a qué contactos se puede enviar mensajes, para el envío de mensajes existen 2 formas:

Estructura del comando:
“COMANDO”||”CONTACTO”||”MENSAJE”
“MENSAJE A”||”CONTACTO”||”HOLA QUE TAL”
“ENVIAR A”||”CONTACTO”||”A QUE HORA LLEGAS”

Para que el comando funcione es importante que se mencione un contacto existente, y que el comando tenga un mensaje sino la aplicación indicara en un mensaje de voz que el comando esta incompleto o no se encontró el contacto.

Cuando todo es correcto se procede a verificar, rectificar o cancelar. Existen 3 nuevos comandos para el envío de mensaje. “CONFIRMAR” para enviar inmediatamente el mensaje. “REPETIR” en el caso que se quiera cambiar el mensaje y “CANCELAR si se desea anular el envío”





CONFIRMAR: Se envía el mensaje y se notifica al usuario receptor, además de un mensaje de voz para el usuario que realizo el envío confirmando el envío, el botón vuelve a la función de envío, reproducción y listar contactos.

REPETIR: Se reproduce un mensaje indicando que se presione la pantalla para modificar el mensaje que se va a enviar y se repite nuevamente la orden de confirmar, repetir y cancelar.

CANCELAR: Se reproduce un mensaje indicando la nulidad del envío y el botón principal vuelve a la función de envío, reproducción y listar contactos.

Reproducción de Mensajes

Se puede escuchar los mensajes de tres maneras:

Estructura del comando:
"COMANDO"||"CONTACTO"
"MENSAJES DE"||"CONTACTO"||
"REPRODUCIR MENSAJES DE"||"CONTACTO"||
"ESCUCHAR MENSAJES DE"||"CONTACTO"||

Configuraciones

En configuraciones se reproduce un mensaje que indica la respectiva ubicación en el aplicativo, los comandos que se pueden usar son:

"Desactivar" o "Activar", estos comando activa o desactiva los mensajes extensos que se reproducen entre opciones.

"Número de mensajes", una cantidad entre 1 y 5, esto determinara el número de mensajes que se reproducirán cuando un usuario quiera escuchar mensajes de determinado usuario.
Ejemplo: "CINCO","DOS", UNO