



Facultad de Ciencias
de la Computación

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Tablero de comunicación: aplicación móvil para el apoyo de comunicación de niños con mutismo.

TESIS

Para obtener el título de
Licenciado en Ciencias de la Computación.

Presenta:

Luis Roberto Caba Flores.

Asesores:

Dr. Juan Manuel González Calleros.

Dra. Josefina Guerrero García.

Dedicatorias

A Dios

Por haberme permitido llegar hasta este punto de mi vida y haberme dado salud para lograr mis objetivos en la vida.

A mis padres

Por haberme apoyado en todo momento, el cariño dado, sus sabios consejos, la disciplina y perseverancia que me ha permitido ser una persona de bien, pero más importante su amor.

A mis hermanos

Por haberme permitido ser el ejemplo, ser un maestro, ser un alumno, ser un amigo, pero, sobre todo el gran apoyo incondicional que recibo de ustedes.

A mis maestros

Por ser una gran guía en esta etapa de mi vida, sus conocimientos, su apoyo incondicional, su confianza hacia mi, por haber compartido su tiempo, sin pedir nada a cambio.

A mis amigos

Por su apoyo, sus consejos, el tiempo compartido, las risas, pero sobre todo su amistad.

Agradecimientos

Deseo expresar mi más sincero agradecimiento a los profesores Dr. Juan Manuel González Calleros y a la Dra. Josefina Guerrero García, quienes son mis asesores, sin ellos la elaboración de esta tesis no hubiera sido posible, quienes además de transmitirme sus conocimientos, me orientaron e impulsaron en mi formación académica, siempre estaré agradecido por los momentos que compartimos, las experiencias inolvidables, así como la dedicación y la atención recibida de su parte.

A CERENE le agradezco el tiempo y asesoría de los psicólogos, sin su apoyo no hubiera sido posible la elaboración de esta tesis, la atención recibida, así como la colaboración mutua para mejorar la calidad de vida de los niños.

Índice.

1. Resumen	1
2. Abstract.....	2
3. Introducción	3
3.1 Motivación	3
3.2 Antecedentes del Problema	3
3.3 Objetivo	4
4. Estado del Campo o del Arte	6
4.1.1 Mutismo.....	7
4.1.2 Mutismo Selectivo	7
4.1.3 Espectro Autista.....	8
4.1.4 Deficiencia Auditiva.....	8
4.1.5 Parálisis Cerebral	8
4.2 Tecnología de apoyo.....	9
4.2.1.1 Sistema Operativo Android.....	9
4.2.1.2 Breve Historia	9
4.2.1.3 Versiones	10
4.2.1.4 Distribución de Aplicaciones.....	11
4.2.1.5 Arquitectura de Android	11
4.2.1.6 Ciclo de vida de una aplicación en Android.....	13
4.2.1.7 Android Studio.....	15
4.2.1.8 Servicios Web en Android.....	15
4.2.2 Modelo Vista Controlador (MVC)	16
4.2.2.1 Modelo	16
4.2.2.2 Vista.....	16
4.2.2.3 Controlador	16
4.2.2.4 MVC Php Codeigniter	17
4.2.3 Google Analytics.....	18
4.2.3.1 ¿Qué proporciona Google Analytics?	18
4.2.4 Aplicaciones Similares	20
4.2.4.1 Tabla Comparativa	21
4.2.5 Tabla 2: Tabla comparativa de aplicaciones similares Justificación de Tecnología .	22
5. Metodología.....	23

5.1	Fase de Inicio.....	24
5.1.1	Identificación de lugar a observar, acudir y documentar la visita	24
5.1.2	Diseñar entrevistas con base a lo observado	26
5.1.3	Diario de Actividades.	27
5.1.4	Diseñando la interfaz de usuario.....	31
5.1.5	Diseño de Estereotipos de Usuario PERSONAS	32
5.1.6	Identificación de Tareas.....	34
5.1.7	Historias de Usuario.....	37
5.1.8	Pruebas para las historias de usuario (DONE)	40
5.1.9	Casos de Uso	41
5.2	Fase de Planeación.....	42
5.3	Fase de ejecución	44
5.3.1	Estadísticas de uso	57
5.4	Fase Control y Cierre.....	58
6.	Implementación de la Aplicación Móvil.....	60
7.	Pruebas	84
7.1	Pruebas de Black Box Test.....	85
7.2	Pruebas de Unidad.....	90
7.3	Pruebas de Usuario.....	91
8.	Impacto Socioeconómico.....	93
9.	Conclusiones.	94
10.	Referencias.....	95



1. Resumen

Cuando hablamos de Educación Especial nos referimos a personas con Necesidades Educativas Especiales (NEE), tales como sobredotación intelectual, discapacidades psíquicas, sensoriales o físicas.

En México medio millón de personas tienen discapacidad auditiva [40]. Una cifra que puede aumentar en un millón si tenemos en cuenta el grado y el origen de la pérdida de audición. Es por ello que se requiere buscar formas más lógicas, accesibles y adaptar el proceso de enseñanza-aprendizaje para personas con algún tipo de discapacidad.

Este trabajo de tesis, propone el desarrollo de una aplicación móvil que tiene un sintetizador de voz para que un niño(a) pueda expresar sus necesidades, acciones o emociones, así mismo, hacerse escuchar en su entorno social. Además, cuenta con un reconocimiento de voz formando una herramienta de apoyo para problemas de lenguaje, mediante actividades que refuerzan y estimulan el habla del niño (a).

Un reto que se presenta es la definición de las actividades, que debe considerar aspectos de usabilidad para niños(as), padres de familia y docentes de educación especial, así como cuidar la parte pedagógica del entrenamiento que se apoya con la aplicación.

De igual manera, las características de nuestros usuarios serán tomadas en cuenta para el diseño de un sistema con prevención de errores, por ejemplo: desde la validación de datos de entrada hasta simplificar la interacción mediante campos fáciles de entender y de llenar. La experiencia del usuario es prioridad en este proyecto.

El proyecto será asesorado por especialistas del Centro de Rehabilitación Neuropsicológica Infantil (CERENE) de la ciudad de Puebla y será probada por usuarios reales. De esta forma validaremos nuestro trabajo.

La aplicación es de gran utilidad y está disponible en la tienda oficial de aplicaciones de Android "Google Play".



Figura 1: Aplicación móvil



2. Abstract

This project develops a mobile Android application to help communicate to children with autism spectrum, selective mutism, cerebral palsy and hearing impairment.

The project was supported by psychologists and neuropsychologists of CERENE group, this is in order to define the main characteristics required to develop the application. For the development of the application was necessary to use the IDE official Google "Android Studio".

The application is available in the official Android app store "Google Play".



3. Introducción

3.1 Motivación

Después de realizar una investigación sobre la educación especial en niños, se observa que existen diferentes áreas que requieren del apoyo de sistemas informáticos que apoyen en el desarrollo de habilidades y destrezas de estos niños. Este trabajo considera la Discapacidad auditiva debido a que puede generar problemas de aprendizaje y de comunicación. Dependiendo de la gravedad puede ser corregida con aparatos auditivos o será necesario recurrir a otros métodos de comunicación (lenguaje de señas). La motivación para la realización de la aplicación móvil del que se hablará a lo largo de este documento, es ayudar a personas con, mutismo selectivo, deficiencia o pérdida de audición para mejorar y/o tener una comunicación de manera autónoma.

Otro factor que ha motivado la realización de este proyecto es la aceptación y expansión que tienen hoy en día los dispositivos móviles inteligentes (Smartphone), y más aún, en el ámbito escolar.

La gran capacidad de procesamiento, almacenamiento y personalización con la que cuentan estos grandes dispositivos, hace posible que hoy en día cualquier niño pueda aprender de una manera más ágil, divertida, segura y sana.

3.2 Antecedentes del Problema

Los niños con discapacidad auditiva enfrentan dificultad para adquirir el lenguaje, esto le dificultara adquirir conocimientos y comprender los eventos a su alrededor. Acorde con el INEGI (<http://www.inegi.org.mx>), en México existe una población de 498,640 personas con limitaciones para escuchar.

Los trastornos de la comunicación tienen el potencial de aislar a los individuos de sus alrededores sociales y educacionales, por ello es esencial encontrar una intervención justa y apropiada. Cuando los niños tienen trastornos musculares, problemas en la audición, o atrasos del desarrollo, su adquisición del habla, lenguaje, y destrezas relacionadas pueden ser afectadas.

Hoy en día vivimos en un mundo que está regido por la tecnología, a la cual los niños tienen accesos desde muy pequeños, en base a ello, podemos hacer que el interés de aprendizaje sea mayor, reforzando habilidades auditivas, de lenguaje y comunicación, todas ellas tienen un diferente enfoque dependiendo de los objetivos específicos que se deseen lograr para que puedan ser usados tanto en un entorno escolar, familiar, como social. La tecnología puede ayudar aquellos niños cuyas condiciones físicas hacen la comunicación difícil.



El uso de sistemas de comunicación electrónicos permite que la gente que no habla y personas con diferentes discapacidades físicas aumenten su participación en la discusión del pensamiento.

3.3 Objetivo

El objetivo del proyecto es ayudar a niños con mutismo selectivo y deficiencia auditiva para que puedan comunicarse en un entorno social, familiar y escolar, además de favorecer el proceso de enseñanza-aprendizaje. Mediante el uso de las nuevas tecnologías a niños con problemas de audición, lenguaje y aprendizaje, la aplicación móvil, puede hacer uso de ella para potenciar sus habilidades de comunicación en el entorno social donde se encuentre. Así mismo, para mejora de la aplicación se enviarán estadísticas de uso para identificar que sección utilizan más los usuarios.

Refiriendo a todo lo anterior se han definido los siguientes objetivos específicos:

- Implementar actividades de enseñanza:
 - Generar una experiencia agradable al interactuar con las actividades de enseñanzas implementadas.
 - Implementar un sintetizador de voz en las actividades de enseñanza para tratar problemas de lenguaje.
- Implementar la comunicación entre el usuario y el entorno que lo rodea.
 - Generar una experiencia agradable al interactuar con temas de comunicación implementadas.
 - Personalizar la comunicación con la selección de género.
 - Ampliar un sintetizador de voz en todos los temas de comunicación ya propuestos.
- Implementar el lenguaje de señas para el aprendizaje de los padres o del propio usuario.
 - Generar una serie de imágenes que representen el abecedario con el lenguaje de señas, cada imagen reproducirá el sonido de la letra que corresponda, esto servirá como aprendizaje de los interesados.
- Implementar el reconocimiento de Voz para reforzar el habla de los usuarios.
 - Implementar el reconocimiento de voz para reforzar el habla de los usuarios mediante la repetición de letras, monosílabas y de forma libre.
- Realizar estadísticas de uso.



- Para mejora de la aplicación, el usuario debe permitir enviar estadísticas de uso de forma periódica, las cuales pueden incluir preferencias en secciones de la aplicación.

La aplicación móvil puede facilitar el trabajo en sesiones entre el psicólogo y el niño, así como dar autonomía a un niño con problemas de lenguaje y/o auditivo para comunicarse mediante un sintetizador de voz con las personas que están en su mismo entorno, además de reforzar el habla de los usuarios mediante el reconocimiento de voz.



4. Estado del Campo o del Arte

La investigación del estado del campo, se divide en dos secciones, por un lado, está la parte correspondiente a los trastornos del habla y sus posibles causas, y por otra parte el aspecto tecnológico que puede ser empleado para apoyar en la comunicación de los pacientes con este trastorno del habla.

4.1 Trastornos del habla

Los trastornos del habla se refieren a las dificultades en la producción de los sonidos requeridos para hablar o problemas con la calidad de la voz. Estos se pueden caracterizar por una interrupción en el flujo o ritmo del habla como, por ejemplo, el tartamudeo o falta de fluencia. Los trastornos del habla pueden constituir problemas con la formación de sonidos, los cuales se llaman trastornos de la articulación o fonológicos, o pueden incluir dificultades con el tono, volumen, o calidad de la voz.

Las intervenciones para la rehabilitación son dirigidas de acuerdo a las metas específicas que tiene cada paciente. Existen diferentes enfoques para lograr estas metas, los cuales están agrupados de la siguiente manera:

- **Enfoque de compensación:** El objetivo es enseñar o entrenar a las personas a usar otros medios, objetos o herramientas sin ayuda alguna, por ejemplo; agendas, listas, alarmas, dispositivos electrónicos como teléfonos celulares, etc. de esta manera se le brinda confianza de sus capacidades al paciente [3]. Es importante destacar que este enfoque solo se utiliza cuando las funciones cognitivas no pueden ser restablecidas.
- **Enfoque de restauración:** Es una estrategia usada cuando se intenta mejorar, restablecer la discapacidad cognitiva del usuario, mejorando la comunicación. Esto se realiza directamente con rutinas repetitivas que simulan habilidades o capacidades cognitivas específicas.
- **Enfoque meta-cognitivo:** Es una técnica enfocada en los pacientes con pérdida de la audición, problemas de lenguaje o de comunicación. Un ejemplo de este enfoque es la práctica constante en la aplicación para mejorar su comunicación y la posibilidad de mejorar los problemas de lenguaje [2].
- **Cambio ambiental asistido:** Es una técnica que altera el ambiente actual del usuario, por ejemplo, con actividades pegadas en la pared, el paciente



recordará fácilmente el uso de la aplicación y podrá ubicar sin dificultad las secciones de su interés.

- **Reemplazo y optimización:** Esta técnica tiene la intención de brindar recursos al paciente de reemplazar otros. Un claro ejemplo es cuando un paciente no puede comunicarse, puede usar la aplicación como un sustituto de comunicación.

Después de tener pláticas constantes con psicólogos y terapeutas, se encontraron posibles enfoques que van dirigidos a este tipo de rehabilitación, pero desafortunadamente no se encontró algún enfoque aplicado a la tecnología.

Otro aspecto importante de este trabajo es acerca del aspecto tecnológico de la propuesta ya mencionada, diseñar e innovar nuevas herramientas intuitivas para esta área, además de tener un enfoque centrado en usuarios con problemas de lenguaje y de comunicación con una interfaz gráfica muy fácil de usar.

4.1.1 Mutismo

El concepto de mutismo hace referencia a la mudez, es decir a la suspensión del habla por parte de un individuo. Las personas que lo presentan se mantienen en silencio ante determinadas situaciones; las causas de este silencio pueden ser varias y activarse de forma voluntaria o como consecuencia de una condición que escapa a su control. Las principales causas del mutismo son físicas y suelen estar vinculadas a las cuerdas vocales, la lengua, la boca, la garganta o los pulmones. Muchas veces el mutismo se define a través del concepto de afonía, que es el concepto médico que se emplea respecto a la falta de capacidad para el habla [1].

4.1.2 Mutismo Selectivo

El mutismo selectivo es más común en niños menores de cinco años que en niños de otras edades y sus causas se desconocen. La mayoría de los expertos cree que los niños con esta afección heredan una tendencia a ser ansiosos e hiperactivos. La mayoría de los niños con esta afección tienen cierta forma de miedo (fobia) extrema, además de que el individuo puede dejar de hablar generalmente en la escuela o en contextos sociales.

Los padres con frecuencia piensan que el niño ha optado por no hablar, pero generalmente el niño en realidad es incapaz de hablar en ciertos contextos.

Algunos niños afectados tienen una historia familiar de mutismo selectivo, timidez extrema o trastornos de ansiedad, lo cual puede aumentar el riesgo de sufrir problemas similares.



Mutismo Selectivo no es lo mismo que mutismo, en el cual los niños nunca hablan. En el caso del mutismo selectivo, el niño tiene la capacidad tanto de entender como de hablar, pero es incapaz de hablar en ciertos contextos o ambientes [4].

4.1.3 Espectro Autista

El trastorno del espectro autista (TEA) es una condición neurológica y de desarrollo que comienza en la niñez y dura toda la vida. Afecta en el comportamiento de una persona, además de afectar en la comunicación, aprendizaje y en el entorno social.

Estos pueden tener problemas para hablar con los demás y es posible que no miren a los ojos cuando una persona les habla. Además, pueden tener intereses limitados y comportamientos repetitivos. Es posible que pasen mucho tiempo ordenando cosas o repitiendo una frase una y otra vez, parecieran estar en su "propio mundo" [5].

4.1.4 Deficiencia Auditiva.

La sordera infantil profunda es un fenómeno cultural en que se unen inseparables pautas y problemas sociales, emotivos, lingüísticos e intelectuales.

La deficiencia auditiva supone un grave impedimento para adquirir y comprender el lenguaje de los oyentes, con una larga tradición cultural, faculta a quienes la padecen al desarrollar un lenguaje propio de naturaleza visual, el lenguaje de signos, de tan fácil y rápida adquisición como el habla para los oyentes, y les hace participes de una cultura perpetuada por este genuino lenguaje [6].

4.1.5 Parálisis Cerebral

La parálisis cerebral es un grupo de trastornos que afecta la capacidad de una persona para moverse, mantener el equilibrio, la postura y comunicarse. Los trastornos aparecen durante los primeros años de vida. Por lo general, no empeoran con el tiempo.

Las personas con parálisis cerebral pueden tener dificultades para caminar, moverse, hacer alguna actividad, comunicarse, tener problemas realizando tareas como escribir o usar tijeras. Algunas tienen otras afecciones médicas, como trastornos convulsivos, o discapacidad mental.

La parálisis cerebral ocurre cuando las áreas del cerebro que controlan el movimiento y la postura no se desarrollan correctamente o se lesionan. Los signos prematuros de parálisis cerebral suelen aparecer antes de los 3 años de edad [7]. Algunos niños nacen con parálisis cerebral; otros la adquieren después del nacimiento.



4.2 Tecnología de apoyo

4.2.1.1 Sistema Operativo Android

Android es una plataforma de software y un sistema operativo para dispositivos móviles basada en un Kernel Linux, desarrollada por Google y más tarde por la Open Handset Alliance [8]. Esta plataforma permite a los desarrolladores escribir código en Java que se ejecuten en móviles mediante las librerías JAVA desarrolladas por Google. También se pueden escribir aplicaciones en otros lenguajes como por ejemplo C, para posteriormente ser compiladas en código nativo ARM y ejecutarlas, aunque este proceso de desarrollo no está soportado oficialmente por Google.

La mayor parte de la plataforma de Android está disponible bajo licencia de software libre de Apache y otras licencias de código abierto. La estructura del sistema operativo, compuesto por más de 12 millones de líneas de código escritas en C, C++, Java y XML, está formada por un conjunto de aplicaciones que se ejecutan en un Framework Java [9] de aplicaciones orientadas a objetos sobre una máquina virtual conocida como Dalvik Virtual Machine y que se compilan en tiempo de ejecución.

Por un lado, el sistema es libre, y las compañías pueden usarlo en sus teléfonos sin pagar licencias a Google y cuentan con el derecho a realizar todas las modificaciones que quieran, lo que les permite personalizarlo a su antojo.

Por otro lado, es un sistema atractivo para los usuarios, ya que cuenta con cientos de miles de aplicaciones disponibles, en su mayoría gratuitas.

Una de las ventajas de Android, que a su vez puede suponer un problema para los desarrolladores, es la diversidad de los terminales que utilizan su sistema operativo.

Al contar con soporte para hardware tan diferente, la compatibilidad de las aplicaciones en algunos casos se ve reducida, y es necesario diseñar interfaces de diferentes tamaños.

4.2.1.2 Breve Historia

En Julio de 2005, Google adquirió Android, Inc., una pequeña startup de California. En esos momentos, la compañía se dedicaba a la creación de software para teléfonos móviles. Una vez en Google, el equipo desarrolló un S.O. basado en Linux para dispositivos móviles. Más adelante, Google adaptó su buscador y sus aplicaciones para su uso en móviles. En septiembre del 2007, Google tenía varias patentes de aplicaciones sobre el área de la telefonía móvil [8].



El 5 de noviembre del mismo año, se anunció la fundación de la Open Handset Alliance al mismo tiempo que la creación de la plataforma Android.

4.2.1.3 Versiones

A continuación, se muestran las versiones de Android desde sus inicios hasta la actualidad:

No. de Versión	Nombre de la Versión	Fecha
1.0	Apple Pie	23 de septiembre de 2008
1.1	Banana Bread	9 de febrero de 2009
1.5	Cupcake	27 de abril de 2009
1.6	Donut	15 de septiembre de 2009
2.0–2.1	Eclair	26 de octubre de 2009
2.2–2.2.3	Froyo	20 de mayo de 2010
2.3–2.3.7	Gingerbread	6 de diciembre de 2010
3.0–3.2.6	Honeycomb1	22 de febrero de 2011
4.0–4.0.4	Ice Cream Sandwich	18 de octubre de 2011
4.1–4.3.1	Jelly Bean	9 de julio de 2012
4.4–4.4.4	KitKat	31 de octubre de 2013



5.0–5.1.1	Lollipop	12 de noviembre de 2014
6.0–6.0.1	Marshmallow	5 de octubre de 2015
7.0	Nougat	15 de junio de 2016

Tabla 1: Lista de distintas Versiones del Sistema Operativo Android.

4.2.1.4 Distribución de Aplicaciones

Las aplicaciones para estos dispositivos se pueden obtener desde Google Play, la tienda de aplicaciones oficial de Google para Android.

Esta tienda es una de las ventajas del desarrollador, le permite una vez pagada la licencia de desarrollo (cuesta 25 dólares americanos), publicar su trabajo y que de esta forma sea descargable desde cualquier dispositivo móvil que cuente con dicho sistema operativo.

4.2.1.5 Arquitectura de Android

Android se encuentra estructurado en varias capas, las cuales se describen a continuación:

- **Aplicaciones** Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones** Los desarrolladores tienen acceso completo a las mismas Apis del framework usadas por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes. Cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean remplazados por el usuario.
- **Bibliotecas.** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android.



- **Runtime de Android.** Incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik 2 Es un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. Página 20 ejecuta archivos en el formato DalvikExecutable (. dex), el cual está optimizad o para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que se transformaron al formato. dex por la herramienta incluida "dx".
- **Núcleo Linux.** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

En la siguiente figura se muestra la arquitectura de Android

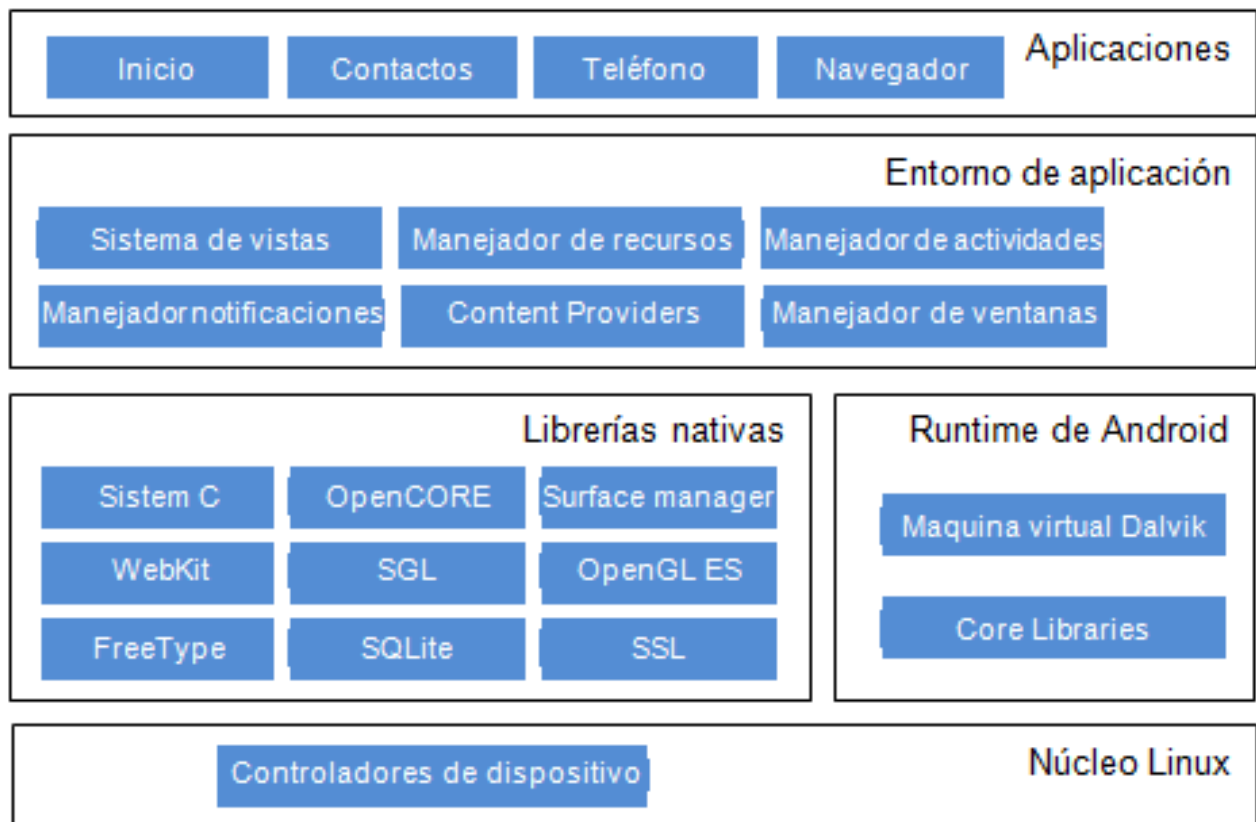


Figura 2: Arquitectura de Android



4.2.1.6 Ciclo de vida de una aplicación en Android

Cada aplicación de Android corre en su propio proceso, el cual es creado por la aplicación cuando se ejecuta, y permanece hasta que la aplicación deja de trabajar o el sistema necesita memoria para otras aplicaciones que se estén ejecutando.

Una característica fundamental de Android es que el ciclo de vida de una aplicación no está controlado por la misma aplicación, sino que lo determina el sistema a partir de una combinación de estados como pueden ser qué aplicaciones están funcionando, qué prioridad tienen para el usuario y cuánta memoria queda disponible en el sistema.

De esta manera, Android sitúa cada proceso en una jerarquía de “importancia” basada en los procesos que actualmente se estén ejecutando o en primer plano [10].

- Un proceso en primer plano, es un proceso que se requiere para lo que el usuario esté actualmente haciendo. Se considera en primer plano si está ejecutándose una Actividad perteneciente a la pantalla con la que el usuario está interactuando.
- Un proceso visible es aquel que contiene una Actividad que es visible al usuario mediante la pantalla, pero no en primer plano (está pausada). Este proceso solo se eliminará en caso de que sea necesario mantener ejecutando los procesos en primer plano.
- Un proceso de servicio es aquel que contiene un servicio que ha sido inicializado.
- Un proceso en Background es aquel que acoge una actividad que no está actualmente visible al usuario. Mientras que dichos procesos implementan su propio ciclo de vida, el sistema puede eliminarlos para dar memoria a cualquiera de los procesos anteriores.
- Un proceso vacío es aquel que no contiene ningún componente activo de ninguna aplicación. La única razón para mantener dicho proceso es para mejorar sus inicializaciones posteriores a modo de caché.

Para comprender mejor el ciclo de vida de una aplicación de Android, en la siguiente figura se muestra el diagrama de flujo de dicho ciclo, mencionando también los métodos que se llaman durante el transcurso del mismo.

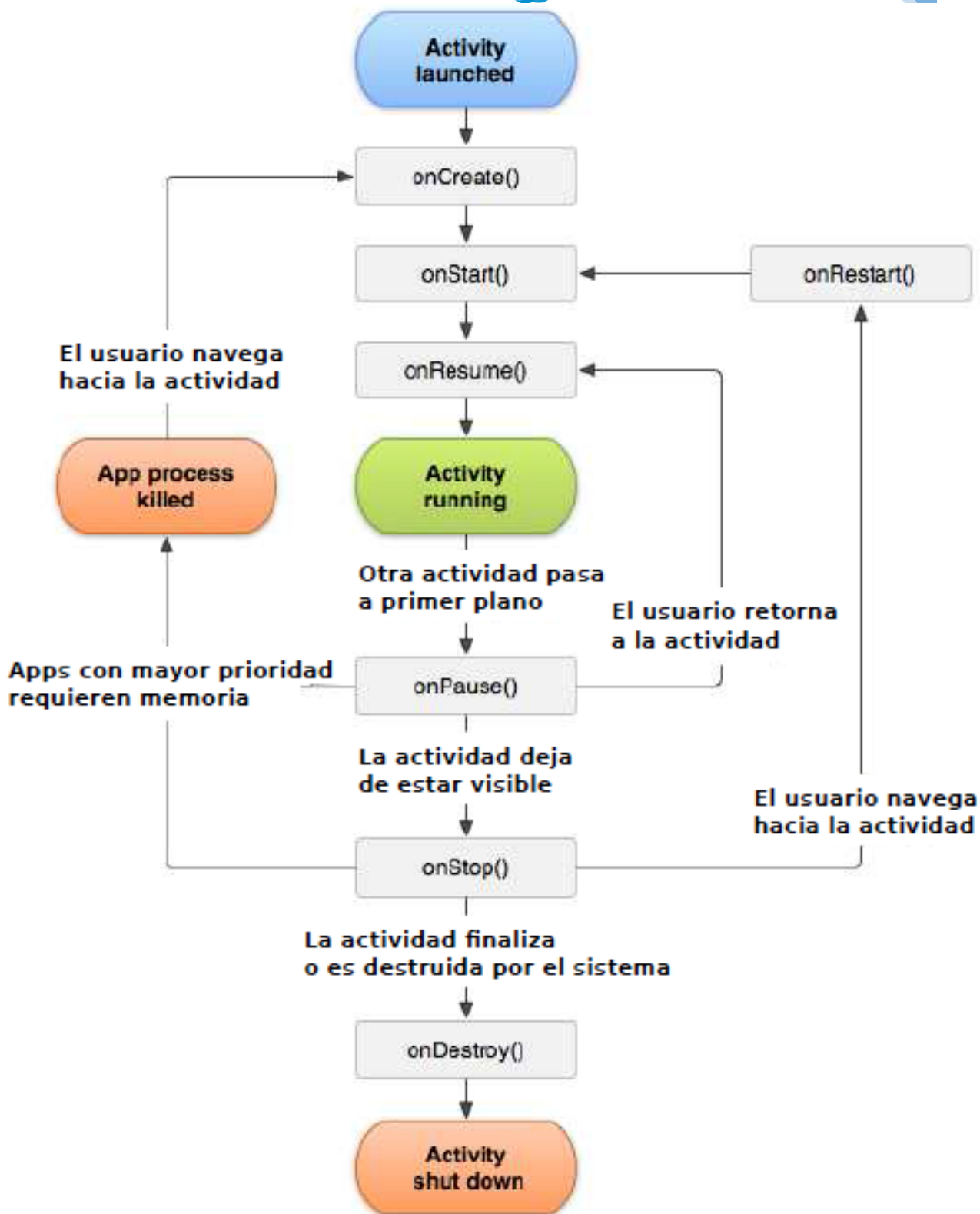


Figura 3: Ciclo de vida de una aplicación en Android



De la Figura 3 pueden sacarse las siguientes conclusiones:

- onCreate (), onDestroy (): abarcan todo el ciclo de vida. Cada uno de estos métodos representan el principio y el fin de la actividad.
- onStart (), onStop (): representan la parte visible del ciclo de vida. Desde onStart () hasta onStop (), la actividad será visible para el usuario, aunque es posible que no tenga el foco de acción por existir otras actividades superpuestas con las que el usuario está interactuando. Pueden ser llamados múltiples veces.
- onResume (), onPause (): delimitan la parte útil del ciclo de vida. Desde onResume () hasta onPause (), la actividad no sólo es visible, sino que además tiene el foco de la acción y el usuario puede interactuar con ella.

4.2.1.7 Android Studio

Android Studio es un entorno de desarrollo integrado para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android.

Al crear un nuevo proyecto en Android Studio, la estructura del proyecto aparece con casi todos los archivos dentro del directorio SRC (fuente), un cambio a un sistema de generación basado Gradle que proporcionará una mayor flexibilidad para el proceso de construcción.

Además, gracias a su sistema de emulación integrado, Android Studio permite ver los cambios que realizamos en nuestra aplicación en tiempo real, pudiendo además comprobar cómo se visualiza en diferentes dispositivos Android con distintas configuraciones y resoluciones de forma simultánea [11].

4.2.1.8 Servicios Web en Android

Para el envío de estadísticas de uso se implementarán servicios web, el término “servicios web” designa una tecnología que permite a las aplicaciones comunicarse en una forma que no depende de la plataforma ni del lenguaje de programación. Un servicio web es una interfaz de software que describe un conjunto de operaciones a las cuales se puede acceder por la red. Usa protocolos basados en el lenguaje XML con el objetivo de describir una operación para ejecutar o datos para intercambiar con otro servicio web. Un grupo de servicios web que interactúa de esa forma define la aplicación de un servicio web específico en una arquitectura orientada a servicios [12].

Para crear una conexión con una base de datos y recopilación de datos de forma remota se utilizará un servidor externo y se creará código php para consumir



datos, debido al gran crecimiento de las aplicaciones web en MVC el proyecto será basado en MVC.

4.2.2 Modelo Vista Controlador (MVC)

Para entender que es un MVC (Model-View-Controller o Modelo-Vista-Controlador) primero debemos comprender que es un patrón de diseño, un patrón de diseño se considera una solución a problemas con un contexto común en diseño de software.

MVC es un patrón de diseño que separa los datos, la lógica de negocios y las interfaces de usuario. Como su nombre lo dice, está separado en tres componentes: modelo, vista y controlador. Al separar componentes se evita código espagueti, además permite reutilizar código, regresar vistas totales o parciales, evitando duplicar estilos o contenido en las vistas. [13]

4.2.2.1 Modelo

Es la capa encargada de los datos, es decir, tiene mecanismos para acceder a la información y también para actualizar su estado. Comúnmente se encarga de comunicarse con la base de datos mediante funciones que accederán a las tablas y realizarán las funciones habituales de datos.

4.2.2.2 Vista

Se trata del código que nos permitirá presentar los datos que el modelo nos proporciona, nos permite mostrar la salida de los datos procesados mediante vistas a los usuarios.

4.2.2.3 Controlador

Es la capa que sirve de enlace entre la vista y el modelo. Envía al modelo comandos para actualizar su estado, y a la vista correspondiente para cambiar su presentación, pero no es el encargado de manipular los datos ni de generar una salida.

Para comprender como funciona un MVC se muestra una representación gráfica del mismo:

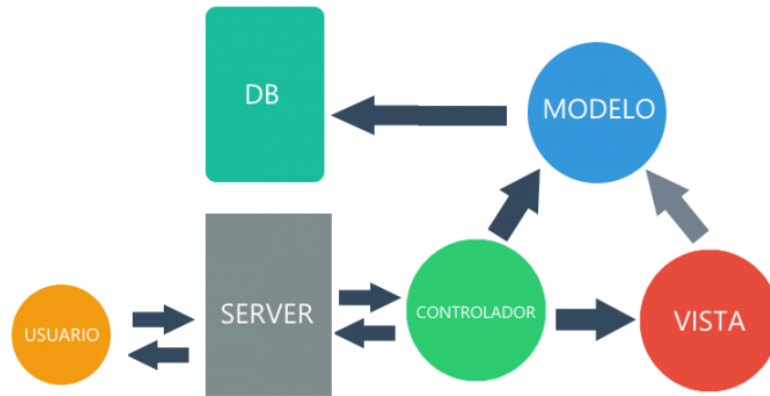


Figura 4: Representación Gráfica de MVC.

- El usuario solicita una acción al servidor
- El servidor atiende la petición y manda a llamar al controlador
- El controlador llama al modelo necesario
 - El modelo atiende la petición y realiza las operaciones de datos correspondientes
 - El modelo regresa el resultado
- El controlador llama a la vista, enviándole los datos procesados del modelo
 - La vista presenta los datos
- El controlador devuelve la vista al servidor
- El servidor presenta el resultado al cliente.

4.2.2.4 MVC Php Codeigniter

PHP Codeigniter es un framework PHP para el desarrollo rápido de aplicaciones web. Codeigniter es un framework para el desarrollo de aplicaciones en php que utiliza el MVC. Permite a los programadores Web mejorar la forma de trabajar y hacerlo a mayor velocidad. [14]

Este framework será usado en la aplicación web que consumirá los datos de estadísticas de uso que enviará la aplicación.



Figura 5: Php Codeigniter



4.2.3 Google Analytics

Como alternativa a servicios web para la recolección de datos de una manera más fácil y ligera se optó por utilizar Google Analytics, al ser multiplataforma tiene soporte y compatibilidad para Android Studio; se implementará Google Analytics a la aplicación para ser monitorizada.

Google Analytics es una herramienta multiplataforma gratuita que permite monitorizar todo el tráfico de cualquier aplicación, esto es con el propósito de conocer el comportamiento de los usuarios, conocer sus hábitos, orientar nuestra aplicación hacia aquello que funciona o mejorar en aquello que no funciona bien. [39]

4.2.3.1 ¿Qué proporciona Google Analytics?

En la página principal de Google Analytics lo primero que llama la atención es el gráfico de gran tamaño que muestra las visitas en un determinado periodo de tiempo, por defecto es en el último mes.

Este gráfico puede ser personalizado para mostrar aquellos datos más interesantes, e igualmente para periodos personalizados. Por defecto vienen visitas de hoy, ayer, la última semana y el último mes, pero se puede seleccionar de un calendario visual aquellas fechas que nos interesan ver en el gráfico.

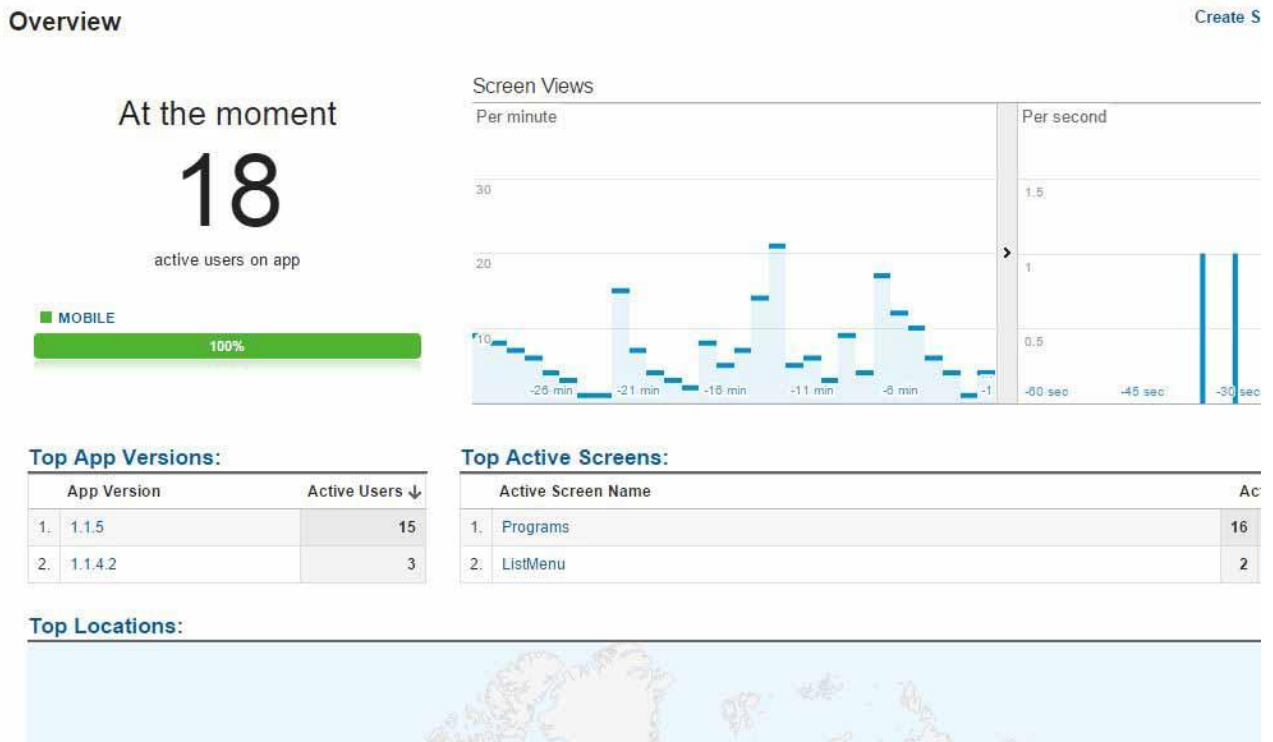


Figura 6: Aspecto de la pantalla de informes estándar de Google Analytics.



Los datos a mostrar más relevantes son:

- **Número de visitantes:** Quizás es el valor más importante, ya que nos indica la visibilidad de nuestra aplicación. Este valor no es un indicador directo de éxito, ya que es también importante ofrecer a los usuarios aquello que realmente buscaban. De nada sirve tener tráfico si la visita no sale satisfecha o el resultado no coincide con lo que esperaba.
- **Vistas más populares:** Nos aporta un valor representativo de los usuarios que entran a la aplicación móvil. En este caso las vistas más visitadas tendrán mayor popularidad frente a otras
- **Tiempo de estancia en la aplicación:** Da una media del tiempo que permanecen los usuarios dentro de la aplicación. A mayor tiempo de estancia, más posibilidades de que el contenido sea de interés y considerado de calidad por los usuarios.
- **Porcentaje de rebote:** Nos dice la cantidad de usuarios que entran a la aplicación y salen de la misma sin realizar interacción alguna. Junto al tiempo de estancia, nos puede dar una idea de la aceptación y utilidad de la aplicación para los usuarios.
- **Datos geográficos:** Saber desde que país los usuarios visitan la aplicación resulta bastante útil, como traducir la aplicación a diferentes idiomas en las que se empieza a notar un incremento de visitas y gracias a ellos nos puede reportar aún más visitas.
- **Tiempo real:** Nos indica el número de usuarios que están visitando la aplicación en tiempo real. Podemos ver varios datos dentro del tiempo real de Google Analytics, entre ellos, las vistas exactas dentro de la aplicación que están siendo visitadas, o desde que países los usuarios están visitando la aplicación.



4.2.4 Aplicaciones Similares

Existen otras aplicaciones similares al proyecto establecido, a continuación, citamos algunos ejemplos:

- **Scout.** El comunicador Sc@ut es un sistema de comunicación sencillo y adaptable al usuario, desarrollado por el grupo de Especificación, Desarrollo y Evolución del Software (GEDES) del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada. Su finalidad es favorecer la comunicación de personas carentes de lenguaje funcional o que lo tienen bastante limitado, entre ellos autistas, personas con síndrome de Down o afectados de parálisis cerebral.

El comunicador Sc@ut puede ser implementado a través del ordenador, Tablet pc, PDA o Nintendo.

Dispone de tres herramientas básicas que permiten definir el perfil del usuario, su calendario y/o agenda de actividades y los escenarios de comunicación [15].

- **E- Mintza.** Pensado inicialmente para facilitar la comunicación de las personas con autismo, puede ser útil también para niños con sordera que aún no han adquirido un lenguaje oral, personas con discapacidad intelectual grave, pacientes con daño cerebral adquirido por ictus o accidentes, personas mayores con enfermedades neurodegenerativas, personas que no pueden expresarse por requerir intubación o ventilación asistida durante su estancia hospitalaria, etc. [16].
- **Board Maker.** Este software de la empresa Mayer-Johnson es básicamente un editor de tableros de comunicación de amplio uso en nuestro contexto, pues permite diseñar con gran facilidad tableros de comunicación para diferentes usuarios y objetivos [17].
- **in TIC PC.** El software In-TIC para PC consiste en una aplicación gratuita para entorno Windows que posibilita que las personas con diversidad funcional mejoren su autonomía personal adaptando las funcionalidades del ordenador a sus propias necesidades. Por un lado, permite facilitar las habilidades de comunicación a personas con dificultades en el habla, utilizando los recursos de voz sintetizada incorporados en los ordenadores y pictogramas de ARASAAC. Por otro lado, permite acceder y utilizar las aplicaciones y opciones básicas del ordenador a través de la creación de un entorno virtual simplificado y personalizable [18].



4.2.4.1 Tabla Comparativa

A continuación, se mostrará una tabla comparativa de las aplicaciones vistas anteriormente con el proyecto definido.

Nombre	Descripción	Ventajas	Desventajas	Costo
Tablero de Comunicación Móvil	Es una aplicación móvil para ayudar a niños que sufren problemas auditivos o personas con mutismo, así como la posibilidad de potenciar sus habilidades de comunicación en el entorno social donde se encuentre.	<ul style="list-style-type: none"> • Portabilidad. • Fácil uso. • Aplicación intuitiva. • Sección de Aprendizaje. • Implementación de reconocimiento de voz. • Funciona de manera Offline para el sintetizador de Voz. 	<ul style="list-style-type: none"> • Limitado a ciertos Usuarios con cierta experiencia en el uso de dispositivos Android. • Limitado a usuarios con lectura y escritura básica. 	Gratis
Scout	Favorecer la comunicación de personas carentes de lenguaje funcional o que lo tienen bastante limitado, entre ellos autistas, personas con síndrome de Down o afectados de parálisis cerebral.	<ul style="list-style-type: none"> • Facilita la comunicación. • Uso en personas con Síndrome de Down • Uso en personas autistas. 	<ul style="list-style-type: none"> • Actualmente no se ha actualizado a nuevas versiones de Sistema Operativo en este caso, Windows. • No es portable. 	Gratis
E-Mintza	Facilitar la comunicación de las personas con autismo, puede ser útil también para niños con sordera que aún no han adquirido un lenguaje oral, personas con discapacidad intelectual grave, pacientes con daño cerebral adquirido.	<ul style="list-style-type: none"> • Facilita la comunicación. • Uso en personas con escasez de lenguaje • Uso en personas autistas. 	<ul style="list-style-type: none"> • No es portable. 	Gratis
Board Maker	Es básicamente un editor de tableros de comunicación de amplio uso en nuestro contexto, pues permite diseñar	<ul style="list-style-type: none"> • Edición de tableros de comunicación 	<ul style="list-style-type: none"> • No puede ser usado en un 	



	con gran facilidad de tableros de comunicación para diferentes usuarios y objetivos	<ul style="list-style-type: none">• Personalización	dispositivo móvil.	Gratis
In-TIC PC	El software In-TIC para PC consiste en una aplicación gratuita para entorno Windows que posibilita que las personas con diversidad funcional mejoren su autonomía personal adaptando las funcionalidades del ordenador a sus propias necesidades.	<ul style="list-style-type: none">• Personalización del ordenador.• Mejorar autonomía de usuarios.	<ul style="list-style-type: none">• No puede ser usado en un dispositivo móvil.	Gratis

4.2.5 Tabla 2: Tabla comparativa de aplicaciones similares Justificación de Tecnología

1) Android

Alternativas: iOS, Windows Phone.

Se ha optado por utilizar Android debido a la gran aceptación de este sistema en los Smartphones de todo el mundo, siendo a día de hoy, el sistema operativo para móviles de mayor crecimiento, además de su cada vez mayor utilización en el mundo de las tabletas. Además, otro motivo suficientemente importante como para emplear Android ha sido la facilidad para el desarrollo, pues se programa en Java, lenguaje que se ha utilizado de manera mayoritaria durante toda la estadía universitaria.

2) Android Studio

Se ha optado por la utilización de Android Studio como IDE para el proyecto debido al hecho de que es el Entorno de desarrollo oficial de Google para Android, además de poseer un conocimiento lo suficientemente amplio para un correcto desarrollo de la aplicación.



5. Metodología

El desarrollo de productos de calidad requiere de la adopción de una metodología de manera robusta y acorde a cada problema. En este caso, optamos por una estrategia de desarrollo ágil, Scrum para ser más precisos, que ha sido resultado de varios años de experiencia [31] y que hemos tropicalizado a nuestro contexto académico para la gestión de proyectos. Cuando hablamos de desarrollar un producto de calidad, no sólo nos referimos a la producción de aspectos funcionales, sino que además cuidamos el producto como un todo. Sin lugar a dudas, uno de los grandes retos es la producción de documentación de calidad, parte fundamental de todo producto software.

A lo largo del desarrollo del proyecto, se visualizaron problemas de diseño [31], identificamos y coincidimos con otros autores como el origen de errores con la calidad de la documentación, como son: uso incorrecto de semántica para nombrar las cosas en todos los niveles del diseño del producto [19][21][23][26][27][28], incorrecta comprensión de los niveles de abstracción durante el modelado [19], identificación incorrecta cardinalidad entre modelos [19], sintaxis y herramientas incorrectamente usadas [20][24], ausencia de coherencia entre las capas de los modelos por una incorrecta adopción de un diseño evolutivo [19], carga cognitiva para lidiar con tantas cosas a la vez [21][24], limitadas capacidades para procesar información [22].

La metodología de trabajo considera el diseño centrado en el usuario que consiste de las siguientes etapas:

1. Identificación del contexto de uso. Para que el proyecto tenga éxito se deberá tener una relación con personas especialistas que se dediquen a tratar a personas de educación especial, esto es la parte más importante del proyecto, los psicólogos nos pueden proporcionar datos cruciales de cómo realizar este proyecto. Este traslado de información se hará mediante entrevistas a psicólogos y visitas a centros de educación especial. de forma que se ira logrando mediante diferentes técnicas como: la observación participativa, las entrevistas, el diario de campo, modelado de tareas. Las entrevistas serán realizadas por el desarrollador de la aplicación, puede hacer tanto entrevista individual como a un grupo de psicólogos que para ello es necesario que las preguntas sean escritas, de tipo cuestionario o preguntas abiertas, de manera que los psicólogos tengan la libertad para responder, no obstante, las preguntas deben ser cuidadosamente diseñadas para que todos las interpreten de igual manera.
2. Diseño centrado en el usuario. Enfoque de diseño cuyo proceso está dirigido por información sobre las personas que van a hacer uso del



producto. Es muy importante tener una observación de una sesión entre el psicólogo y el niño; el seguimiento de los avances en cada uno de los casos para conocer que colores usar y que tipo de imágenes utilizar para llamar la atención del usuario final. Esto se logra mediante el uso de patrones interactivos, selección de técnica de interacción adecuada, prototipo de mediano nivel.

3. Validación y Pruebas. La usabilidad de la aplicación es muy importante ya que es necesario una aplicación móvil fácil de usar, entender, útil para que el aprendizaje o la comunicación se vuelva más amena. Para lograr este objetivo, la interacción con los psicólogos y los niños serán vitales para este proyecto. La evaluación de la aplicación móvil será realizada por psicólogos y los mismos usuarios, los cuales expresaran su opinión para tener una aplicación estable y completa. Cuando los usuarios o psicólogos no estén en total acuerdo en partes del contenido, se establecerá una comunicación para cambiar ciertos aspectos de forma que sea agradable para todos.

5.1 Fase de Inicio

El proceso de identificación de requerimientos se basa en la elicitación de escenarios textuales. El proceso de observación participativa es muy significativo ya que permite adentrarse al quehacer de las personas mediante la observación durante la ejecución de su trabajo. Con esto amoldamos la solución a la forma en que trabajan las personas y en consecuencia el sistema será una extensión a su quehacer diario.

5.1.1 Identificación de lugar a observar, acudir y documentar la visita

Cuando hablamos de personas que presentan de una discapacidad auditiva o con problemas de mutismo, no hay suficientes lugares en donde se puedan atender estos problemas pues no se les brinda demasiada importancia, por lo tanto, no es fácil encontrar lugares donde atiendan a este tipo de personas y nos permitan hacer observación del trabajo que se realiza.

Al encontrarse con pocas alternativas que resuelven este problema, se tuvo un acercamiento a un Centro de Rehabilitación Neuropsicológica Infantil (CERENE), el cual se encuentra ubicado en la 4 norte 401 Col. Centro. Puebla, Puebla.



CERENE es un centro que cuenta con un equipo de profesionales trabajando con pacientes que presentan diversos problemas en diversos grados en el normal desempeño de sus vidas, así como también en la de sus familias.

Los déficits que se producen a nivel emocional, conductual y cognitivo afecta a la independencia del individuo, condicionado a su entorno, afectando notablemente la calidad de vida de todos los afectados.

En el centro de rehabilitación se atienden diferentes tipos de problemáticas con respecto a discapacidades y problemas psicológicos, de esta manera tuvimos acceso a las terapias con distintos niños.

Los programas para cada niño son personalizados de acuerdo a la condición que presenta, integrando el aporte específico de diversas disciplinas. Se trabajó en forma conjunta con el terapeuta para visualizar varios pacientes y así lograr comprender el tratamiento que reciben este tipo de niños.

El objetivo que se espera al realizar las observaciones de las terapias es la calidad y la permanencia de superación, autocrítica y deseo constante de mejorar.

Las visitas fueron constantes donde el terapeuta estuvo trabajando con un niño de 3 años, que presenta discapacidad auditiva profunda de nacimiento, de esta manera, le impedía comunicarse con su entorno social, por lo tanto, se utilizó la aplicación móvil “Tablero de Comunicación” que se desarrolló para que el niño tuviera facilidades, así mismo, el niño podría realizar una mayor posibilidad de actividades y mejorar la comunicación con los demás, de esta manera el niño puede ser independiente.

Posteriormente tuvimos la oportunidad de trabajar con una niña con mutismo severo de 17 años de edad, el diagnóstico que se presentó fue que no tenía problemas psicológicos o traumáticos, solo tenía un impedimento en su lenguaje el cual no tenía la forma de expresión y así mismo la comunicación con los demás, se tuvo la oportunidad de poder trabajar cooperando de manera correcta con las actividades propuestas e incluso hubo intentos de repetir frases que reproducía la aplicación.

Una parte vital de la observación es buscar la manera de llamar, atraer el interés y la atención del usuario, esto es crucial para la aplicación pues al tener el niño un problema auditivo, de mutismo o algún problema de discapacidad, su principal comunicación con el exterior es el sentido de la vista, la interfaz debe ser llamativa y de fácil uso, esto implica una correcta selección de colores, imágenes y navegación de usuario. Se obtuvo apoyo de psicólogos de CERENE para poder



realizar la interfaz de usuario, teniendo la experiencia adecuada con niños y saber que es lo ideal para ellos.

5.1.2 Diseñar entrevistas con base a lo observado

Al realizar la visita, se diseñaron preguntas tanto a los psicólogos como padres. Las entrevistas fueron las siguientes:

Preguntas para los psicólogos

- 1.- ¿Cree que algún niño ha hecho movimientos extraños durante las actividades realizadas? *Por el momento los comportamientos son estables.*
- 2.- ¿Cómo nota que los niños prestan atención y durante qué tiempo lo hacen? *Se muestran participativos, responden preguntas concretas, durante 10 o 15 minutos dependiendo de la actividad a realizar.*
- 3 - ¿Cómo hace para llamar la atención de los niños? *A través de colores e imágenes llamativas*
- 4 ¿Qué hace cuando el niño no quiere prestar atención? *Cambiar de actividad y hacerla más dinámica.*
- 5.- ¿Cuáles son los métodos que utiliza? *Dependiendo de qué es lo que se va a trabajar con el niño ya sea de manera cognitiva o motriz.*
- 6- ¿Qué tipo de material usa con los niños? *Pelotas, tarjetas, fichas, pinturas, cuentos y dibujos.*
- 7.- ¿Cómo es que usted se dirige con ellos? *Señas, hablarle fuerte al niño en caso de que tenga problemas auditivos.*
8. ¿Cómo responden a nuevos juegos? *Se hacen adecuaciones a las actividades a realizar ya sea subir o bajar el nivel de aprendizaje*
9. ¿Los niños tienen alguna clase de computación o saben manejar algún dispositivo electrónico? *Si, Tablet, celular, computadora.*
- 10.- ¿Qué tanto usted usa la tecnología con ellos? *En algunos casos.*
- 11.- ¿Los niños ponen atención a los aparatos electrónicos? *Ponen demasiada atención*
- 12.- ¿Qué tantos intereses tienen los padres con sus hijos? *Si están al pendiente*



13.- ¿Usted con su experiencia como haría para mejorar su situación? **Mejor tratamiento, más material para trabajar con los niños.**

Entrevista a madre/padre de familia

1.- ¿Cuántas horas al día pasa haciendo la tarea con su hijo? **Depende de cómo él se sienta, porque podríamos tardar 5 horas o 15 minutos si él está con ganas de hacer la tarea.**

2.- ¿Qué colores son los preferidos de su hijo? **Los colores como el Tornasol, más que nada los colores brillantes, y en particular también el rojo.**

3.- ¿Qué actividades típicas realiza? **Le gusta la equitación, no sé si le gusten las computadoras pues no tenemos una en la casa (ríe).**

4.- ¿Qué es lo más difícil de convivir con él (puntos de conflicto)? **Mi carácter no ayuda pues me desespero con facilidad (Ríe).**

5.- ¿Conoce sus metas? **Él ha expresado su deseo de llegar a ser independiente, quiere ser policía cuando sea grande, aunque también ha dicho que quiere ser entrenador de perros o biólogo marino, a veces me da respuestas que no entiendo de dónde saca, a veces me dice que él es líder, tiene aptitudes de líder.**

6.- Cuando usted no puede cuidar de su hijo ¿Quién cuida de él? **Cuando no puedo cuidarlo yo, lo cuida mi mamá, pero si ella tampoco puede entonces se lo dejo a una vecina el problema ahí es que ella no sabe el lenguaje de señas y a veces le desespera porque no lo entiende.**

5.1.3 Diario de Actividades.

Para poder desarrollar correctamente la aplicación móvil se realizó un diario de actividades, el cual consta de 10 semanas de observación, que a continuación se describe:

Número de Prioridad de Riesgo.

NPR: Productividad	
1	No se aprendió nada
2	Se aprendió poco
3	Se aprendió lo suficiente
4	Fue de mucha ayuda



Semana número: 1

Fecha	Hora inicio- fin	Actividad	Descripción	NPR
Sábado, 13 de febrero	3:00 pm - 4: 00 pm	Platica con Lic. Félix.	Platica de familiarización y problemáticas que se tienen en el centro CERENE acerca de los niños discapacitados del habla y cómo podemos apoyarlos con las TICS.	4

Tabla 3: Semana 1

Semana número: 2

Fecha	Hora inicio- fin	Actividad	Descripción	NPR
Sábado, 20 de febrero	3:00 pm - 4: 00 pm	Observación de niño con discapacidad del habla	Esta fue la primera semana en donde se tuvo la oportunidad de trabajar con el niño y su psicóloga. Pero desafortunadamente el niño no llego, pues se le presentaron algunos inconvenientes a su familia, lo cual orilló que no pudiera asistir a su terapia.	1

Tabla 4: Semana 2

Semana número: 3

Fecha	Hora inicio- fin	Actividad	Descripción	NPR
Sábado, 27 de febrero	3:00 pm - 4: 00 pm	Observación de niño con discapacidad del habla	Observación constante del niño que presenta discapacidad del habla. Se observó su comportamiento: Cómo interactúa con objetos que se le dan, La forma de comunicación entre terapeuta-niño. Es muy expresivo a la hora de querer hacer algo. Es muy activo con las distintas actividades. Le gustan las cosas coloridas e interactivas. Se desespera fácilmente con una actividad después de mucho tiempo de interacción.	4

Tabla 5: Semana 3



Semana número: 4

Fecha	Hora inicio- fin	Actividad	Descripción	NPR
Sábado, 2 de abril	3:00 pm - 4: 00 pm	Pruebas de usuario de la aplicación con niño con discapacidad del habla	<p>En esta ocasión tuvimos la oportunidad de que el niño por fin interactuara con nuestra aplicación, la cual al principio le resulto un tanto extraña, pues, aunque si había tenido relación con artefactos tecnológicos, nunca había interactuado de manera más formal con una aplicación.</p> <p>Es por ello que la primera vez que la utilizó no hizo mucho y solo le llamo la atención, pues él esperaba un juego o algo más relacionado con su entretenimiento.</p> <p>Después con la ayuda de la psicóloga le pudimos explicar el funcionamiento de nuestra aplicación y lo “divertido que podía ser utilizarla”.</p> <p>También se tomó la opinión de sus padres, los cuales de principio mostraron una cara de sorpresa, pues les parecía un poco extraño ver que un celular “hablara”. Aunque después les gustó mucho y les pareció que podría ser de gran ayuda. Lo que la psicóloga nos recomendó fue mejorar los colores ya que para ese prototipo se utilizaron colores un tanto oscuros.</p>	4

Tabla 6: Semana 4



Semana número: 5

Fecha	Hora inicio- fin	Actividad	Descripción	NPR
Sábado, 9 de abril	10:00 am - 11: 00 am	Pruebas de usuario de la aplicación con otro niño discapacitado de mayor edad	La psicóloga evaluó la app con la niña y resultó muy buena en cuanto al uso. La psicóloga propuso nuevas ideas para la aplicación como el teclado de lenguajes de señas para una mejor comunicación, cambiar las oraciones para que sean menos formales y hacer una sección de aprendizaje que contenga distintas actividades.	4

Tabla 7: Semana 5

Semana número: 6

Fecha	Hora inicio- fin	Tipo de Actividad	Descripción	NPR
Sábado, 23 de abril	11:00 am - 12: 00 pm	Reasignación de psicóloga	La psicóloga no pudo continuar con el proyecto debido a cambios de horario, una nueva psicóloga se ofreció para trabajar con el proyecto.	2

Tabla 8: Semana 6

Semana número: 7

Fecha	Hora inicio-fin	Actividad	Descripción	NPR
Sábado, 6 de mayo	11:00 am - 12: 00 pm	Trabajar en conjunto con una niña.	Con una nueva psicóloga asignada además de un cambio de horario se asignó para el proyecto una niña con mutismo, la primera sesión fue para conocer a la niña e identificar posibles situaciones en las que pueda ser útil la aplicación, además de recibir consejos en la aplicación como para comunicación y aprendizaje.	3

Tabla 9: Semana 7



Semana número: 8

Fecha	Hora inicio- fin	Actividad	Descripción	NPR
Sábado, 20 de mayo	12:00 pm - 01: 00 pm	Pruebas de usuario con la niña con mutismo.	Se implementaron los consejos dados por la psicóloga, se hicieron pruebas de usuario con la niña las cuales fueron satisfactorios, sin embargo, la interfaz no era muy clara para la niña.	4

Tabla 10: Semana 8

Semana número: 9

Fecha	Hora inicio- fin	Actividad	Descripción	NPR
Sábado, 3 de junio	12:00 pm - 01: 00 pm	Pruebas de usuario con la niña con mutismo.	La interfaz se rediseño para que la experiencia de usuario fuera agradable. Se corrigieron varios errores de programación, así como implementación de imágenes agradables.	2

Tabla 11: Semana 9

Semana número: 10

Fecha	Hora inicio- fin	Actividad	Descripción	NPR
Lunes, 6 de junio	05:00 pm – 10:00 pm	Presentación de aplicación a personal directivo.	La aplicación fue presentada a directivos de USAER y diferentes CAM para conocer la aplicación y así poder validar la aplicación.	5

Tabla 12: Semana 10

5.1.4 Diseñando la interfaz de usuario.

En el diseño de la aplicación se utilizaron diferentes herramientas y colores para poder hacer la aplicación más atractiva para el usuario. Como la aplicación móvil va dirigida principalmente a personas con discapacidad auditiva o mutismo, se decidió implementar la técnica interactiva de manipulación directa.

“La técnica interactiva de manipulación directa es un estilo de interacción hombre-máquina que conlleva la representación de los objetos, acciones y retroalimentación rápida. Los usuarios se sienten más identificados al ver objetos como iconos los cuales funcionan como normalmente lo harían en el mundo real”.

Los colores de la aplicación fueron propuestos por psicólogos. Se determinó usar diferentes tipos de colores en cada una de las diferentes opciones del menú para que la interfaz fuera más llamativa para el usuario. A continuación, se muestra un ejemplo de la técnica interactiva de manipulación directa con iconos.



Figura 7: Aplicación de la técnica de interacción directa con iconos.

5.1.5 Diseño de Estereotipos de Usuario PERSONAS

En esta sección se presenta una de las problemáticas más comunes y menos atendidas acerca de las personas que utilizarán la aplicación, la cual es la interacción de estas personas que se encuentran en un entorno sin conocimiento de lenguaje de señas.

En el proyecto “Tablero de Comunicación” se centra en personas con sordera o mutismo, ya sea total o parcial de nacimiento, parálisis cerebral, deficiencia auditiva o espectro autista, el propósito de la aplicación es ayudar a este tipo de personas para que puedan interactuar con mayor facilidad en cualquier entorno en el que se encuentren, así mismo se diseñó una interfaz interactiva, intuitiva y fácil de usar para los usuarios.

A continuación, se describe el prototipo de usuarios que posiblemente usaran la aplicación en una tabla:



Tipo de usuario	Características
Usuario novato	<ul style="list-style-type: none">• No tienen ningún conocimiento sintáctico. Cuentan con un conocimiento semántico pobre sobre el uso de sistemas interactivos.• Medianamente tienen un entendimiento de la aplicación• Habilidades de Aprendizaje Bajas• Bajo conocimiento de Tecnologías
Usuario Intermedio	<ul style="list-style-type: none">• Tienen buen conocimiento semántico tanto del uso de sistemas de software como del dominio de tareas• Presentan inconvenientes a nivel de cómo se llevan a cabo las tareas en el sistema, con problemas en recordar detalles de conocimiento sintáctico.• Habilidades de Aprendizaje Medias• Mediano conocimiento de las tecnologías
Usuario Experto	<ul style="list-style-type: none">• Están muy familiarizados con todos los aspectos sintácticos y semánticos del sistema.• Habilidades de Aprendizaje Altas• Alto conocimiento de las tecnologías.

Tabla 13: Usuarios

Los principales usuarios de la aplicación serían personas con sordera y/o mutismo, parálisis cerebral, espectro autista, discapacidades o necesidades educativas especiales, como usuarios de la aplicación, se ha desarrollado para ayudarlos a reforzar habilidades, lenguaje y de comunicarse con otras personas, mejorando su calidad de vida.

Su uso es muy práctico para poder establecer un diálogo con las personas que se encuentren en su entorno.

Tiene algunas restricciones, claro está, por el hecho de que la persona que utilice la aplicación deberá ya saber escribir y leer de forma correcta, así que el tipo de usuario ideal sería el intermedio.

La finalidad de hacer prototipos de usuario es para mejorar la aplicación para cuando usen los usuarios la aplicación se sientan satisfechos y cómodos.



5.1.6 Identificación de Tareas.

Para el proyecto “Tablero de Comunicación” es necesario identificar las tareas que se van a realizar y así poder planificarlas.

A continuación, se describe las tareas identificadas por la aplicación móvil para usuarios de nivel intermedio, la tabla se divide en:

- Tarea.
- Justificación.
- Naturaleza.

ID	Tarea	Pred.	Justificación	Naturaleza
1	Entrar a la aplicación “Tablero de Comunicación”	/	El usuario abrirá la aplicación móvil.	Iterativa
2	Proporcionar datos.	1	Al ingresar a la aplicación por primera vez el usuario debe proporcionar su género y edad.	Iterativa
3	El usuario navegara por la interfaz de la aplicación.	2	El usuario podrá visualizar todas las secciones que la aplicación ofrece.	Iterativa
4	Entrar a la sección de Aprendizaje	3	El usuario podrá visualizar todas las secciones que ofrece la sección de Aprendizaje	Iterativa
5	Entrar a la sección de Reconocimiento de Voz	4	El usuario podrá visualizar todas las secciones que ofrece la sección de Reconocimiento de Voz.	Iterativa
6	Entrar a la sección de Letras del Reconocimiento de Voz.	5	El usuario elegirá una letra y activará el reconocimiento de voz para práctica del habla.	Iterativa
7	Entrar a la sección de Monosílabas	6	El usuario elegirá una letra, posteriormente elegirá una monosílaba de la letra elegida.	Iterativa
			El usuario podrá practicar de	



8	Entrar a la sección de practica libre del Reconocimiento de Voz.	7	forma libre con el reconocimiento de voz para práctica del habla.	Iterativa
9	Entrar a la sección de Figuras Geométricas	8	El usuario conocerá las figuras geométricas básicas y podrá escucharlas mediante el Sintetizador de Voz.	Iterativa.
10	Entrar a la sección de Colores	9	El usuario conocerá los colores y podrá escucharlos mediante el Sintetizador de Voz.	Iterativa
11	Entrar a la sección de Números	10	El usuario conocerá los números y podrá escucharlos mediante el Sintetizador de Voz.	Iterativa
12	Entrar a la sección de Fonemas	11	El usuario podrá identificar las letras más rápidamente mediante usando imágenes.	Iterativa
13	Entrar a la sección de Animales	12	El usuario podrá conocer los animales, además de poder escuchar los sonidos de los mismos.	Iterativa
14	Entrar a la sección de Operaciones	13	El usuario podrá realizar las 4 operaciones básicas con distintos objetos de la vida diaria	Manual
15	Entrar a la sección de Partes del cuerpo	14	El usuario conocerá las partes del cuerpo y podrá escucharlos mediante el Sintetizador de Voz.	Iterativa
16	Entrar a la sección de Comunicación	15	El usuario podrá visualizar todas las subsecciones que ofrece la sección de Comunicación.	Iterativa
17	Entrar a la sección de Saludos	16	El usuario podrá visualizar los saludos que ofrece la aplicación, el usuario decidirá si escucharlos o agregarlos a favoritos.	Iterativa



18	Entrar a la sección de Alimentos	15	El usuario podrá visualizar los alimentos que ofrece la aplicación, el usuario decidirá si escucharlos o agregarlos a favoritos.	Iterativa
19	Entrar a la sección de Emociones	18	El usuario podrá visualizar las emociones que ofrece la aplicación, el usuario decidirá si escucharlas o agregarlas a favoritos.	Iterativa
20	Entrar a la sección de Ejercicio	19	El usuario podrá visualizar los ejercicios que ofrece la aplicación, el usuario decidirá si escucharlos o agregarlos a favoritos.	Iterativa
21	Entrar a la sección de Salud	20	El usuario podrá visualizar los estados de salud que ofrece la aplicación, el usuario decidirá si escucharlos o agregarlos a favoritos.	Iterativa
22	Entrar a la sección de Necesidades	21	El usuario podrá visualizar las necesidades que ofrece la aplicación, el usuario decidirá si escucharlas o agregarlas a favoritos.	Iterativa
23	Entrar a la sección de actividades Extras.	22	El usuario podrá visualizar las actividades extra que ofrece la aplicación, el usuario decidirá si escucharlas o agregarlas a favoritos.	Iterativa
24	Entrar a la sección de Respuestas	23	El usuario podrá visualizar las respuestas disponibles que ofrece la aplicación, el usuario decidirá si escucharlas o agregarlas a favoritos.	Iterativa
25	Entrar a la sección de Preguntas	24	El usuario podrá visualizar las preguntas disponibles que ofrece la aplicación, el usuario decidirá si escucharlas o agregarlas a favoritos.	Iterativa



26	Entrar a la sección de Agregar	25	El usuario podrá ingresar texto de su interés para bien escucharlo o agregarlo a la sección de favoritos.	Iterativa
27	Entrar a la sección de Favoritos	26	El usuario podrá visualizar los favoritos agregados, podrá escucharlos, editarlos o bien, eliminarlos.	Iterativa
28	Entrar a la sección de Lenguaje de señas	27	El usuario podrá visualizar el lenguaje de señas.	Iterativa
29	Entrar a la sección de Configuración	28	El lenguaje podrá elegir el color de fondo, además de proporcionar el género y la edad.	Manual
30	Entrar a la sección de Contacto	29	El usuario visualizar el contacto de la aplicación.	Iterativa

Tabla 14: Identificación de Tareas.

En total fueron 30 tareas que fueron identificadas para verificar que exista una buena interacción humano computadora.

5.1.7 Historias de Usuario.

Las historias de usuario son descripciones cortas y simples de una funcionalidad, escritas desde la perspectiva de la persona que necesita una nueva capacidad de un sistema, por lo general el usuario, área de negocio o cliente.

A continuación, se muestra las historias de usuario para la aplicación móvil.

Historia de usuario: Niño.

Historia 1)

Como: usuario de la aplicación (niño)

Quiero: comunicarme con las personas que me rodean

Para: expresar lo que siento.

Criterios de Aceptación



- La aplicación debe ser amigable para los niños
- Debe tener un atractivo visual para ellos.

Historia 2)

Como: usuario de la aplicación (niño)

Quiero: expresar mis necesidades básicas con la ayuda de la aplicación

Para: que las personas que me rodean me atiendan.

Criterios de Aceptación

- La aplicación debe ser amigable para los niños
- La aplicación debe tener imágenes relacionadas con lo que se quiere expresar.

Historia 3)

Como: usuario de la aplicación (niño).

Quiero: agregar frases personalizadas que no estén en las frases predeterminadas

Para: expresar más cosas.

Criterios de Aceptación

- Para formar frases personalizadas la aplicación debe ser amigable y fácil de usar.

Historia de usuario: Psicóloga.

Historia 1)

Como: psicóloga de CERENE

Quiero: ayudar a los usuarios con discapacidades del habla y auditiva

Para: que puedan mejorar en su aprendizaje y desarrollo a través de una aplicación móvil que les servirá de herramienta.

Criterios de Aceptación:

- Debo ser capaz de ubicar los temas y contenido de la aplicación.
- Poder identificar qué hace cada acción

Historia 2)

Como: psicóloga de CERENE

Quiero: enseñarles a los usuarios el uso de la aplicación y sus funciones

Para: que puedan ser aprender el uso y manejo de esta.



Criterios de aceptación

- Debo ser capaz de poder enseñar a los niños el uso de la app y sus funciones que tiene
- Identificar problemas que puedan surgir en el aprendizaje

Historia 3)

Como: psicóloga de CERENE

Quiero: agregar mensajes predeterminados a la aplicación para que puedan estar disponibles

Para: que puedan ser usadas por el usuario y aprenda el uso de esta opción.

Criterios de aceptación

- Debo poder identificar cómo funcionan las opciones de agregar mensajes predeterminados
- Saber qué hace cada botón del sistema para tener mayor dominio de esta app.

Historia de usuario: Padres.

Historia 1)

Como: usuario de la aplicación (padres)

Quiero: aprender a usar la aplicación y entender las funciones de la aplicación

Para: ayudar a mi hijo a comprender su situación.

Criterios de Aceptación

- Debe ser capaz de ubicar los temas y contenidos de la aplicación
- Poder identificar que hace cada acción

Historia 2)

Como: usuario de la aplicación (padres)

Quiero: enseñar a mi hijo a usar la aplicación y sus diferentes acciones

Para: estar al pendiente de su desarrollo

Criterios de Aceptación

- Debe ser capaz de ubicar los temas y contenidos de la aplicación
- Dominar la aplicación para enseñar a usarla.



5.1.8 Pruebas para las historias de usuario (DONE)

Historia 1)

Como: María

Quiero: expresar mi enojo

Para: que se den cuenta de mi desagrado.

Historia 2)

Como: Norma

Quiero: expresar mi necesidad de que me cambien la playera

Para: que me bañen.

Historia 3)

Como: Papa de Jorge

Quiero: aprender a usar la aplicación

Para: comprender a mi hijo en sus emociones

Historia 4)

Como: Mama de maría

Quiero: ayudar a mi hijo a aprender

Para: mi hijo aprenda la aplicación

Historia 5)

Como: Luis

Quiero: expresar mi antojo

Para: que se me den de comer



5.1.9 Casos de Uso

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. Para mostrar gráficamente los casos de uso, se utilizan diagramas que representa la forma en como un Usuario (Actor) opera con el sistema en desarrollo.

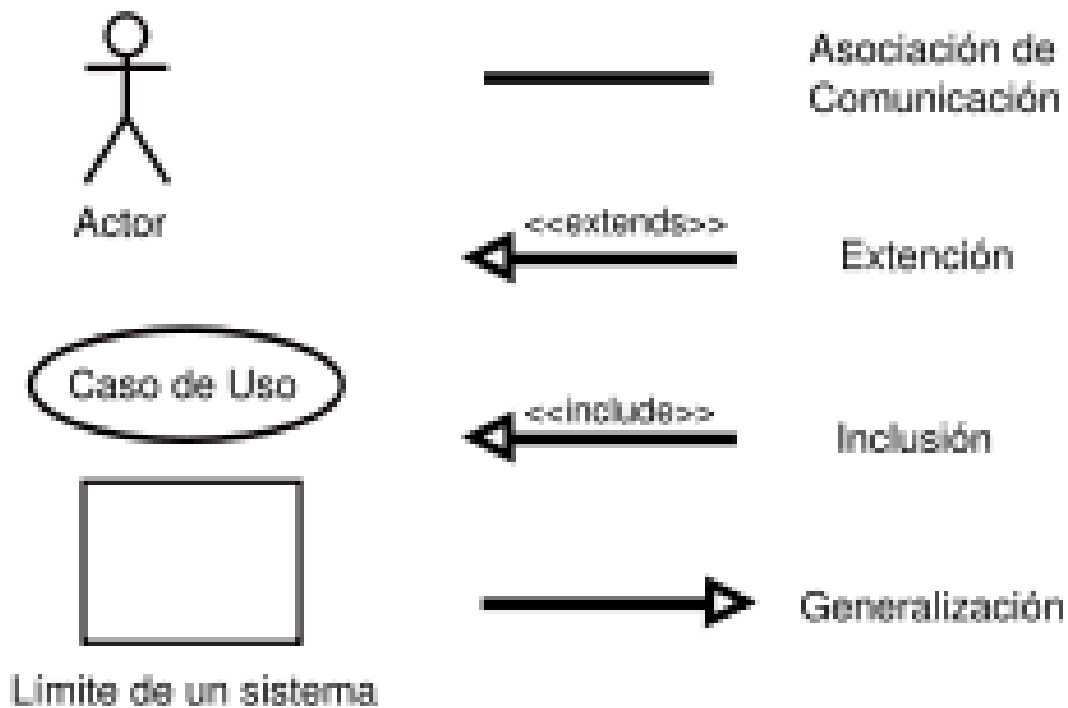


Figura 8: Elementos de un diagrama de Casos de Uso.



A continuación, se muestra un diagrama de casos de uso para la aplicación móvil.

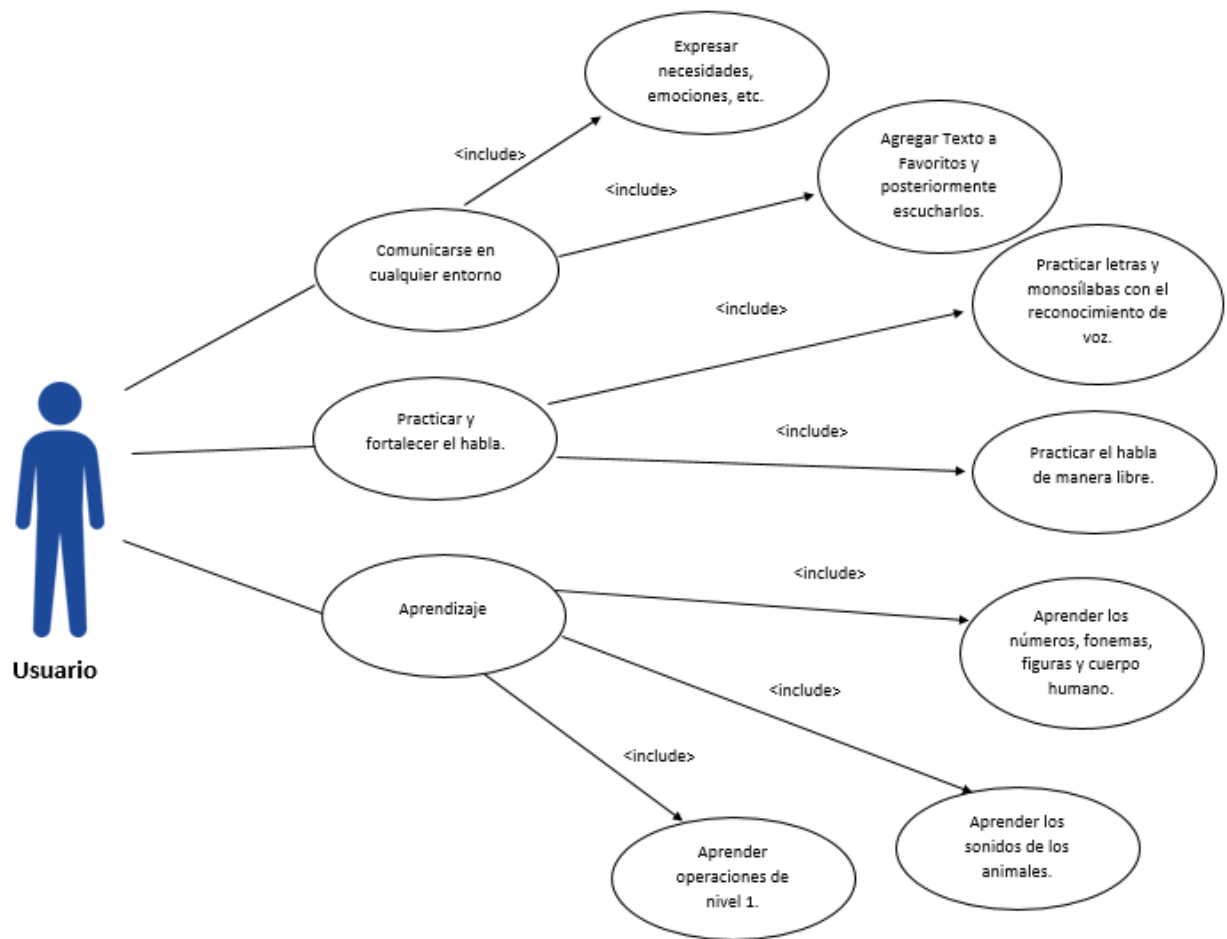


Figura 9: Diagrama de Casos de Uso.

5.2 Fase de Planeación

La técnica de visión del producto define la meta a largo plazo del proyecto que vamos a realizar [34]. Esta técnica fue creada por [35] y tiene como fundamento la importancia de trabajar con un rumbo claro. Normalmente cuando trabajamos con una idea clara de lo que queremos hacer mantendremos a nuestro equipo de trabajo motivado y la calidad de su trabajo será superior [36]. Y para lograrlo vamos a reusar todo lo que hemos documentado en etapas previas. Para ello tenemos que definir cuatro elementos [35]:

- **Visión:** A dónde queremos llegar al desarrollar esta solución. Esta frase o párrafo se redacta en función de los siguientes cuatro elementos: grupo objetivo, necesidades, producto, valor.
- **Grupo Objetivo:** Son los clientes o usuarios para los que se está desarrollando la solución.



- **Necesidades:** En esa búsqueda de soluciones útiles y de la mano de lo que las PERSONAS necesitan, tenemos que tener una lista muy clara y concreta de los problemas que queremos resolver.
- **Producto:** Aquí se describen las funciones que debería tener el sistema a desarrollar.
- **Valor:** De la mano de las necesidades y posibles soluciones, tenemos que identificar los beneficios o el valor que aportamos a nuestros usuarios si resolvemos de manera adecuada su problemática.

Es importante definir estos elementos ya que representa un factor de motivación [36] para el equipo de desarrollo. La empatía con los problemas de nuestro cliente junto con el valor que le aportamos a su vida al resolverlo de forma correcta se ha demostrado generan cambio positivo en la forma en la que resolvemos el trabajo. Y un producto que cubre con lo requerido simplemente dejará a nuestra usuaria sin ningún beneficio.

De igual forma que las personas, nuestra visión del producto debe plasmarse en una ficha de trabajo de tal forma que este a la vista siempre del equipo de trabajo y así mantengamos un rumbo fijo durante el desarrollo del producto. Para cada usuario nuestra recomendación es crear una visión del producto, de esta manera dejamos muy claro el objetivo de cada módulo del proyecto.

A continuación, se hace énfasis al proyecto mediante la herramienta “Vision Board”.






 VISION STATEMENT			
 TARGET GROUP	 NEEDS	 PRODUCT	 VALUE
Niños discapacitados tienen con problemas para comunicarse con otras personas.	Los niños tienen la necesidad de comunicarse mejor y más rápido con las personas sin las discapacidades que ellos tienen.	Una aplicación para plataforma móvil que funciona como un sintetizador de voz para suplantar la voz de la persona discapacitada, además de tener una	Se podrá beneficiar a los niños con discapacidad por medio de la comunicación y obtener una mejor calidad de vida

Figura 10: Vision Board



Durante las etapas anteriores se han definido los elementos relevantes para la elaboración del proyecto. Sin embargo, hasta ahora estos procesos son transcritos manteniendo el uso del lenguaje natural. Para poder hacer la elicitación de requerimientos, tomando como base los datos recolectados en la observación participativa, las entrevistas, necesitamos transcribir esta información a un formato más estable.

La elicitación (un término aceptado en la comunidad de ingeniería de software) tiene como objetivo identificar dentro de los escenarios de trabajo los actores, encargados de ejecutar el trabajo. Además, capturan de forma concreta la manera en que se realiza el trabajo [37]. Esto es representativo y evocativo para un usuario final ya que tiene elementos para validar los requerimientos del sistema educativo a crear.

Esta estrategia, a diferencia de los métodos tradicionalmente usados por los ingenieros en software (por ejemplo, modelos del dominio, modelos del usuario) maximizan las propiedades deseables, con la exhaustividad, coherencia y corrección del usuario final, el contexto educativo. Con base en la descripción textual que se obtiene del proceso educativo, el diseñador de flujos de procesos identifica las tareas, los recursos a su cargo, su desarrollo, la unidad donde se ejecutan y así sucesivamente.

5.3 Fase de ejecución

Para los problemas de comunicación que se pretenden resolver, se propone una interfaz sencilla y fácil de usar, además se han propuesto varias actividades que serán realizadas en un dispositivo móvil usando el sistema operativo Android.

La vista principal cuenta con un menú de actividades a elegir por el usuario, tomando como referencia que la aplicación sea para niños se utilizan colores llamativos así, la experiencia de los usuarios es agradable debido a que la interfaz es visualmente atractiva para ellos. En caso de que sea la primera vez que el usuario entre a la aplicación se le pedirán datos de edad y sexo, esto es para fines de personalización dentro de la aplicación. En la figura 11 podemos ver el menú de actividades a elegir.

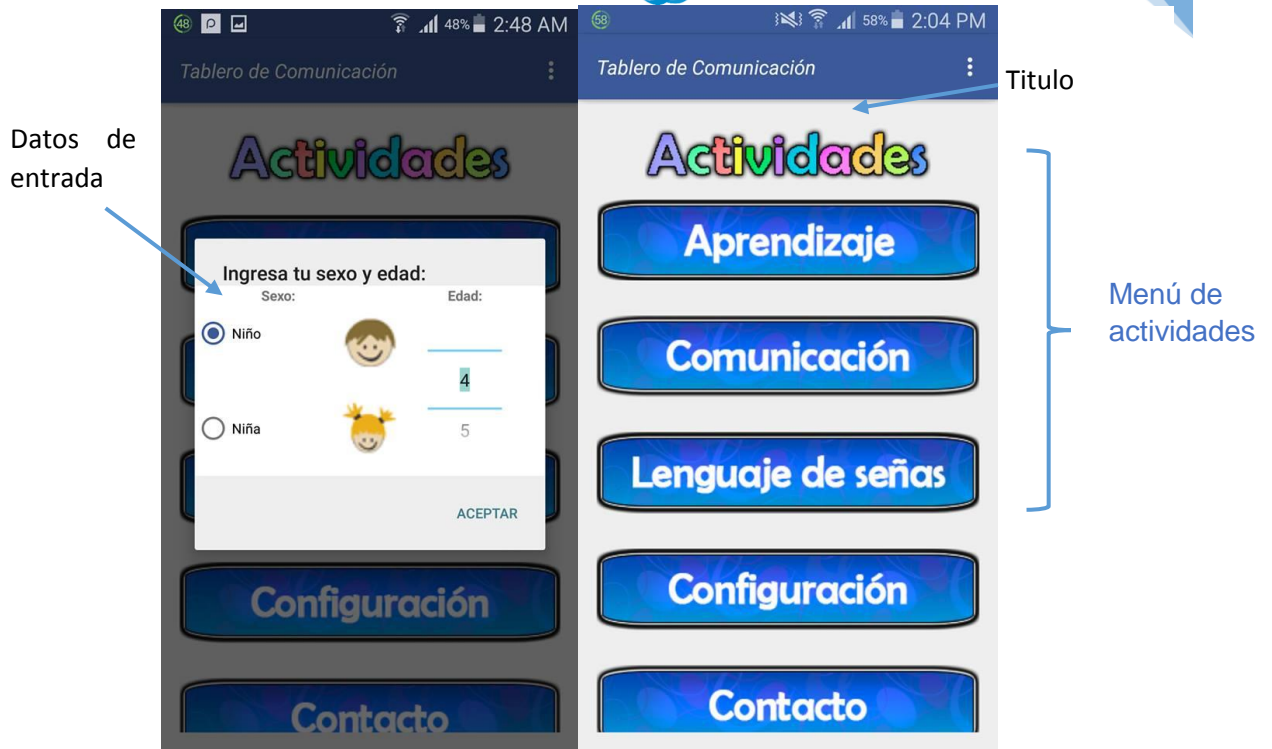


Figura 11: Menú de actividades a elegir por el usuario

Al ingresar por primera vez también se le pedirá al usuario la autorización de envío de estadísticas de uso. Si no acepta, la aplicación no enviara datos de forma remota.

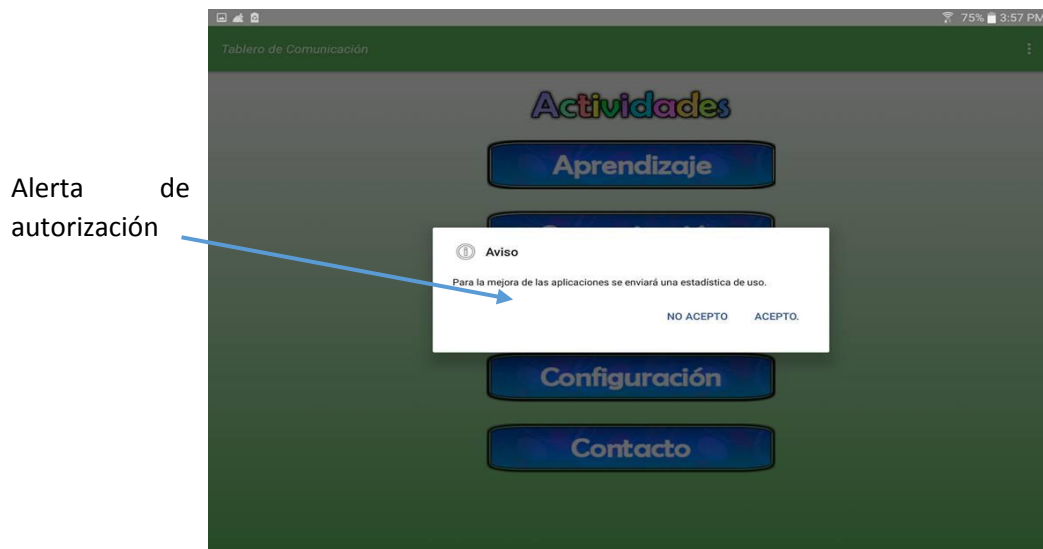


Figura 12: Autorización de envío de estadísticas de uso.



En la sección de aprendizaje muestra un menú de actividades de nivel 1 por tanto se podrá reforzar el conocimiento del usuario como son: reconocimiento de voz, aprendizaje de figuras geométricas, colores, números, identificación de fonemas mediante imagen-palabra, identificación de animales con sonidos, operaciones básicas y partes del cuerpo humano. En la figura 13 podemos ver el menú de aprendizaje a elegir.



Figura 13: Menú de aprendizaje

En la sección de Reconocimiento de Voz trae actividades que reforzaran el habla, que incluye práctica de letras, monosílabas y una sección libre. En la figura 14 podemos ver el menú de aprendizaje a elegir.



Figura 14: Menú de reconocimiento de Voz

En la sección de letras, el usuario visualizará una letra, posteriormente podrá elegir una de su preferencia, así podrá practicar y reforzar el habla, cuando el usuario repita la letra elegida la aplicación regresará una respuesta positiva en caso de que el resultado sea correcto, en caso opuesto regresará una respuesta negativa, en cualquier momento puede elegir otra letra, así mismo el usuario podrá controlar el volumen del dispositivo. En la figura 15 observamos una letra a elegir.

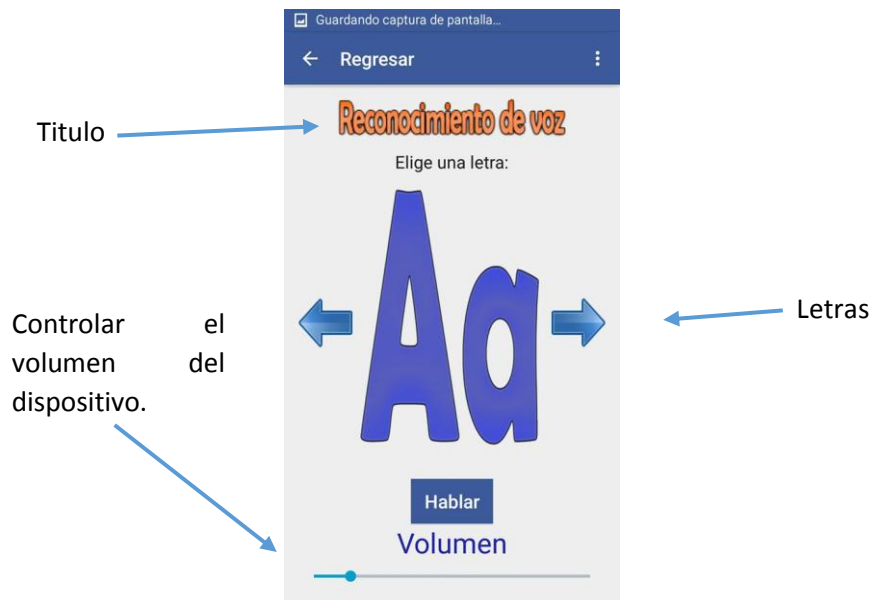


Figura 15: Letras a elegir.

En la sección de monosílabas, el usuario visualizará una letra y las monosílabas correspondientes, posteriormente podrá elegir una letra de su preferencia y se podrá visualizar, cuando el usuario repita la monosílaba elegida, la aplicación



regresará una respuesta positiva en caso de que el resultado sea correcto, en caso opuesto regresará una respuesta negativa, en cualquier momento puede elegir otra letra. En la figura 16 observamos una letra a elegir.

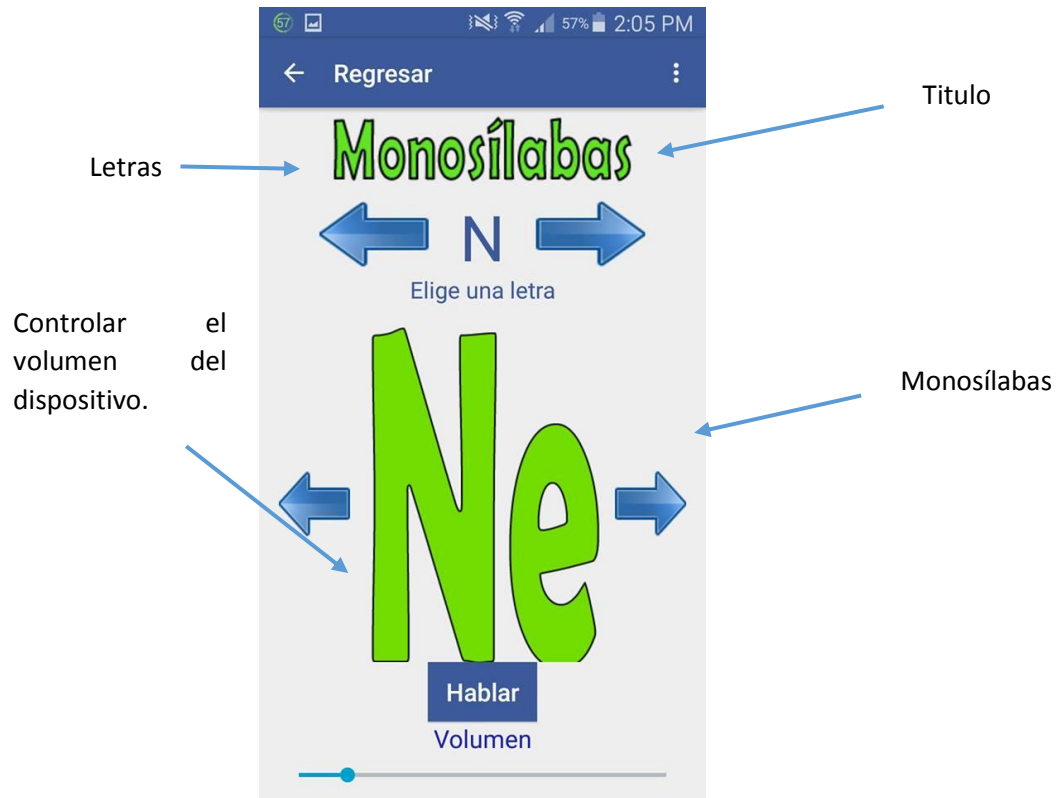


Figura 16: Letras a elegir con las monosílabas

En la sección de practica libre el usuario practicará de forma libre, podrá decir tanto palabras sencillas como complejas, si el usuario decide escuchar lo que dijo podrá hacerlo mediante un botón de audio. En la figura 17 observamos la interfaz de practica libre.



Figura 17: Practica libre.

En la sección de figuras geométricas, el usuario conocerá las figuras básicas así mismo podrá escucharlas para un aprendizaje visual como auditivo. Para que el usuario visualice todas las figuras geométricas hará uso de un Scrolling. En la figura 14 observamos la interfaz de Figuras Geométricas.

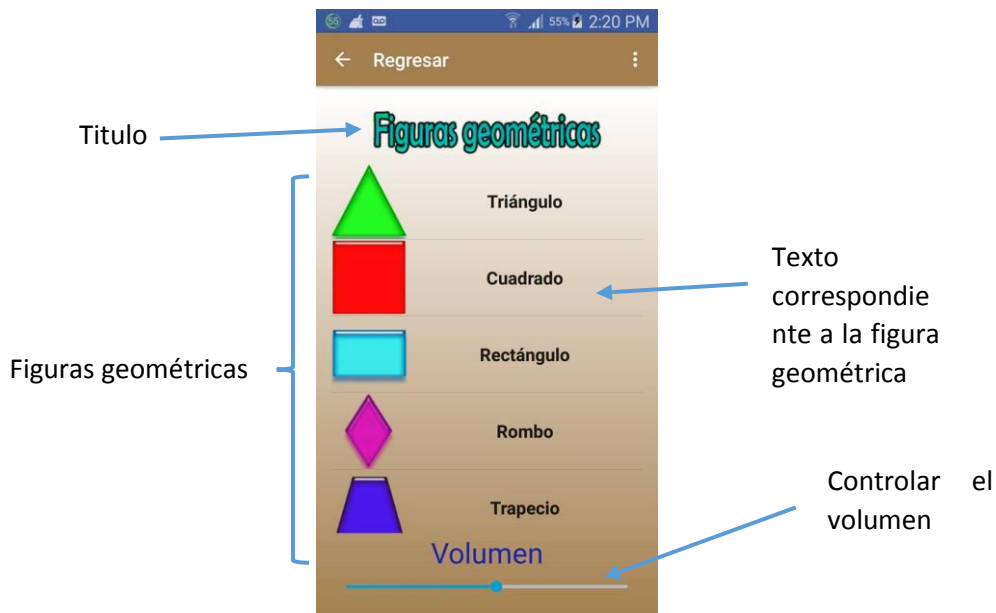


Figura 18: Interfaz de Figuras Geométricas.

En la sección de colores, el usuario conocerá colores y podrá escucharlos con el Sintetizador de Voz para un aprendizaje visual y auditivo. Para que el usuario



visualice todos los colores hará uso de un Scrolling. En la figura 19 observamos la interfaz de los colores.

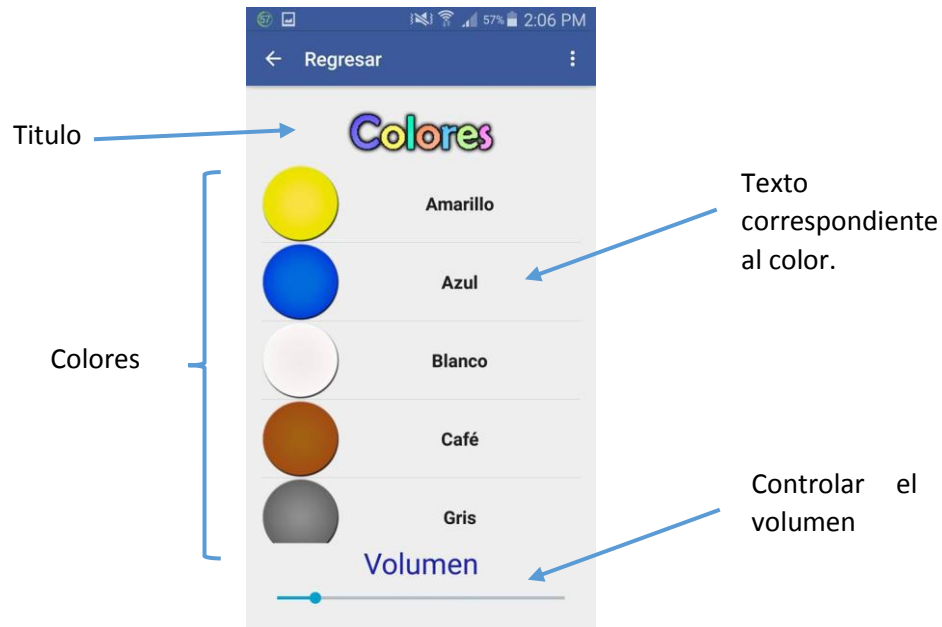


Figura 19: Interfaz para el aprendizaje de colores.

En la sección de números, el usuario conocerá los números de forma fácil, además podrá identificarlos mediante colores. En la figura 20 observamos la interfaz de los colores.

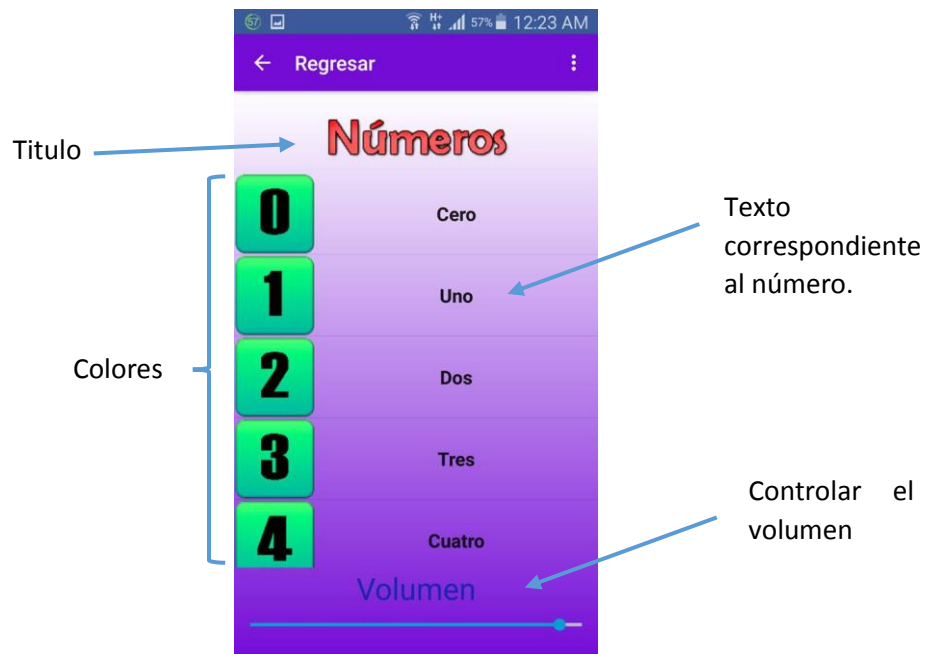


Figura 20: Interfaz para el aprendizaje de números.



En la sección de fonemas, el usuario podrá identificar las letras mediante imágenes, donde podrá escuchar los sonidos e identificar letras mediante las imágenes relacionadas. Para que el usuario visualice todos los fonemas hará uso de un Scrolling. En la figura 21 observamos la interfaz de los fonemas.

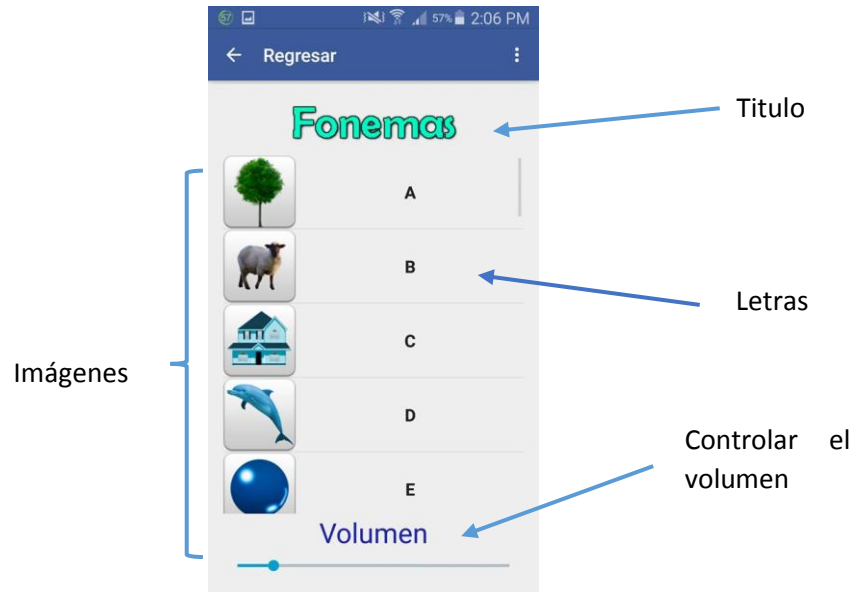


Figura 21: Interfaz para el aprendizaje de los fonemas.

En la sección de animales el usuario aprenderá a identificar los animales mediante los sonidos que emiten los mismos, para escuchar algún animal tan solo basta con seleccionar alguno. Para que el usuario visualice todos los animales hará uso de un Scrolling, además podrá controlar el volumen del dispositivo. En la figura 22 observamos la interfaz de los animales.

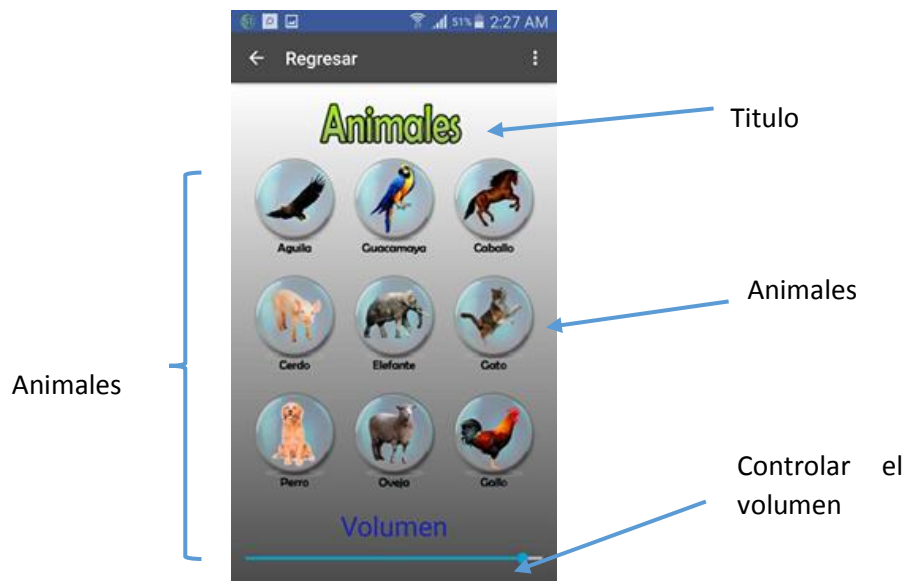


Figura 22: Interfaz para la identificación de animales.



En la sección de Operaciones, el usuario podrá realizar operaciones básicas de matemáticas de Nivel 1 las cuales son: suma, resta, multiplicación y división, la complejidad se basa en la edad del usuario. El usuario podrá reconocer los números mediante imágenes aleatorias, así reforzará el aprendizaje de los números. En el caso de la resta se procura que el resultado sea positivo o igual a cero, para el caso de la división se procura que el resultado sea un número entero y positivo. En caso de que la casilla del resultado sea incorrecta o se encuentre vacía se le notificara al usuario con el sintetizador de Voz que vuelva a intentarlo, en caso contrario la aplicación felicitará al usuario por haber hecho correctamente la operación.

Si el usuario decide aumentar o disminuir la dificultad de las operaciones podrá hacerlo dando click en la esquina superior derecha, saldrán distintos niveles de dificultad, según la opción del usuario será la dificultad de la operación. En la figura 23 observamos la interfaz de las operaciones.



Figura 23: Interfaz para realizar operaciones.

En la sección de partes del cuerpo humano, el usuario conocerá las partes del cuerpo, así mismo podrá escucharlos con el Sintetizador de Voz para un aprendizaje visual como auditivo. Para que el usuario visualice todas las partes del cuerpo hará uso de un Scrolling. En la figura 24 observamos la interfaz de las partes del cuerpo.



Figura 24: Interfaz para aprender partes del cuerpo humano.

La sección de Comunicación contiene diferentes temas que permite a los usuarios con distintas discapacidades poder expresar necesidades para la vida diaria. Desde la solicitud de alimentos hasta la expresión de dolor. En la figura 25 podemos ver los diferentes temas que ofrece la aplicación.



Figura 25: Interfaz que muestra los diferentes temas de comunicación.



Una de los temas que ofrece la sección de comunicación es la expresión de emociones, el usuario podrá expresar sus estados de ánimo tales como alegría, tristeza, enojo, sorpresa y susto, utilizando la técnica interactiva de Manipulación directa el usuario identificará las emociones según las expresiones de la cara, esto es debido a que el usuario se siente más identificado con estas imágenes, las cuales funcionan como normalmente la gente expresa sus sentimientos en el mundo real., además podrá agregar las emociones a una sección de favoritos dando click en la estrella que representa los favoritos.

Según el género del usuario la aplicación mostrará de forma personalizada el estado de ánimo, de igual manera los demás temas que ofrece la sección de comunicación son similares al tema de emociones. En la figura 26 podemos ver los diferentes estados de ánimo que ofrece la aplicación.

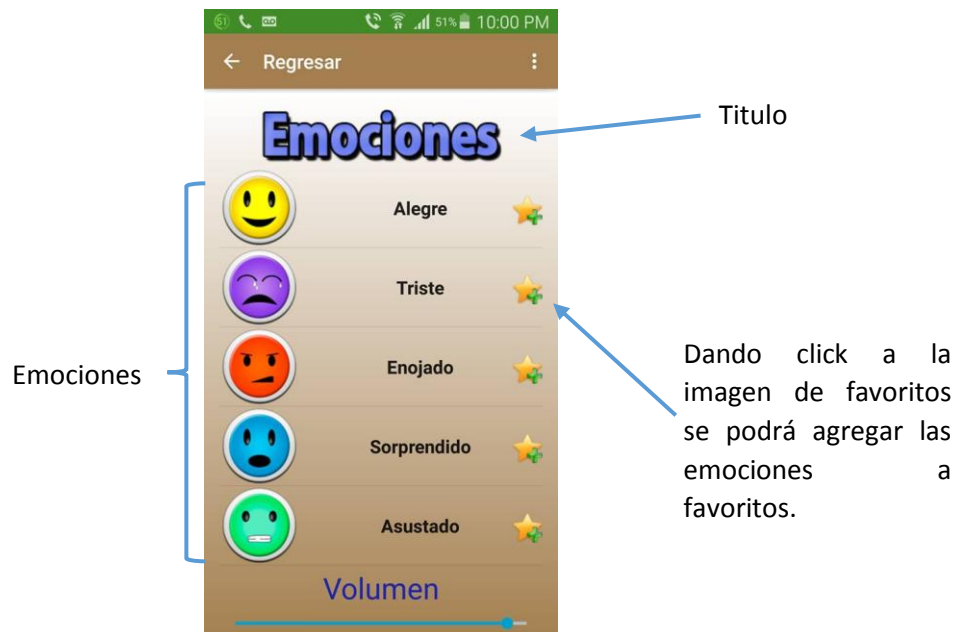
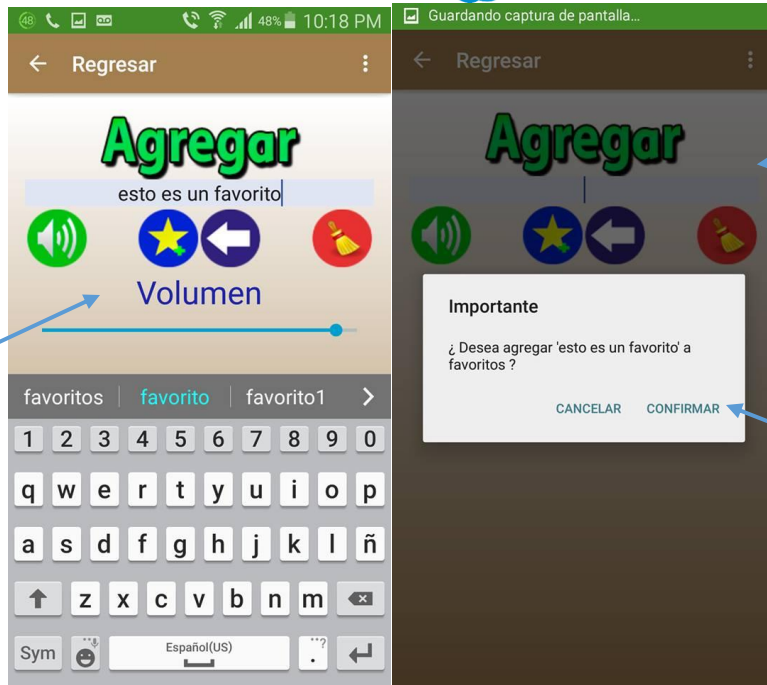


Figura 26: Interfaz que muestra los diferentes estados de ánimo que ofrece la aplicación.

Hay una sección que ofrece a los usuarios la posibilidad de escuchar y/o agregar texto personalizado a la sección de Favoritos, para que el usuario pueda agregar texto, la aplicación valida que el campo de entrada no sea vacío o nulo, además el usuario solo podrá ingresar un texto de máximo 150 caracteres, enseguida saldrá una alerta pidiendo confirmación por parte del usuario para agregar lo ingresado a favoritos. En la figura 27 podemos ver la interfaz de agregar texto a favoritos.

Dando click en el botón, el usuario podrá escuchar el texto personalizado.

Dando click en el botón, el usuario agrega el texto personalizado a favoritos.

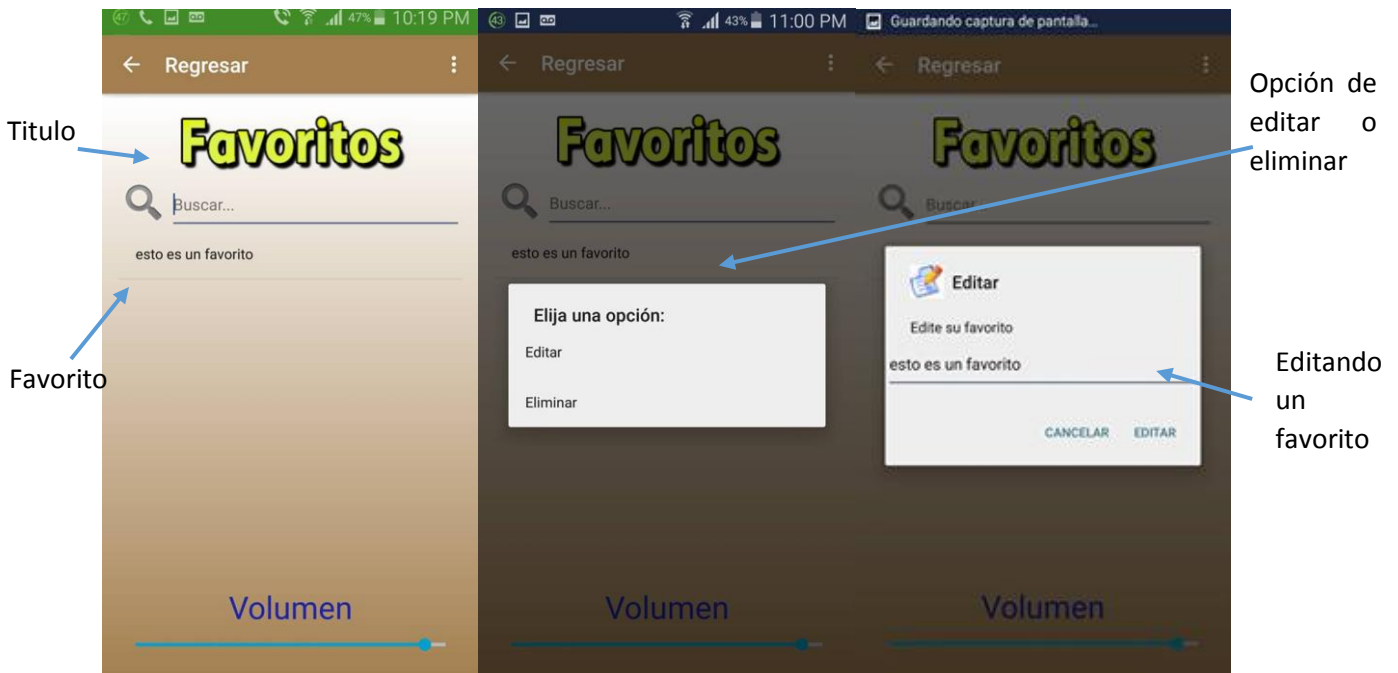


Título

Alerta de confirmación para agregar texto a favoritos.

Figura 27: Interfaz donde el usuario puede agregar texto personalizado a la sección de favoritos.

En la sección de favoritos el usuario visualizará sus favoritos donde los podrá buscar y escuchar, si desea editar o eliminar algún favorito tan solo basta con dar un largo click en el texto favorito, dependiendo el gusto del usuario elegirá la opción de editar o bien eliminar. En la figura 28 vemos la interfaz de los favoritos.



Título

Favorito

Opción de editar o eliminar

Editando un favorito

Figura 28: Interfaz donde el usuario puede ver, editar e incluso eliminar sus favoritos.

En la sección de lenguaje de señas el usuario conocerá el sonido de las letras del lenguaje de señas forma fácil. En la figura 29 observamos la interfaz del lenguaje de señas.

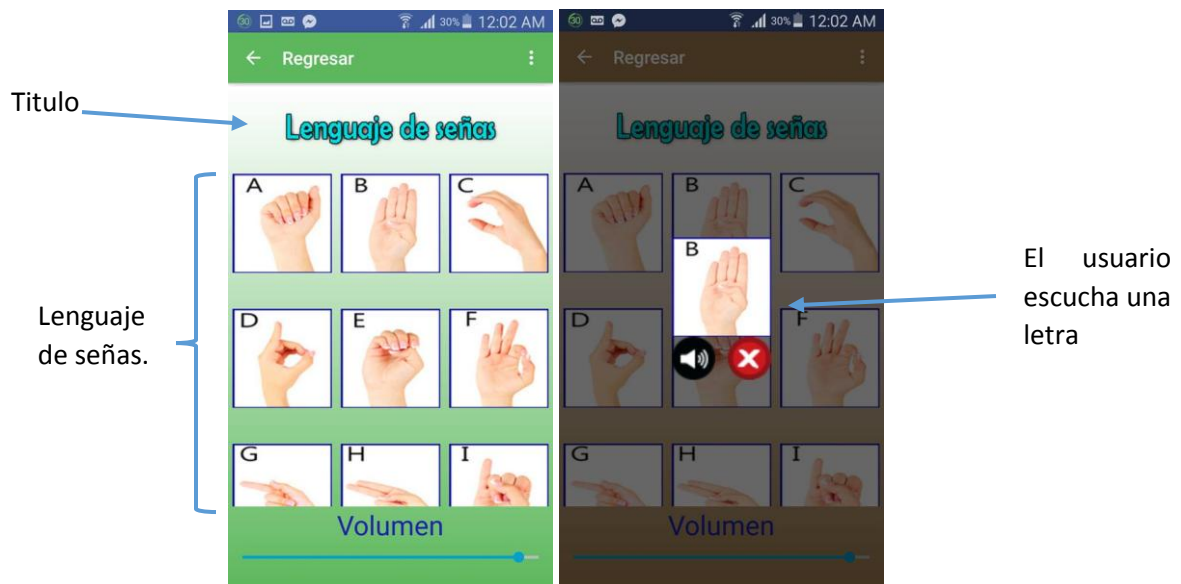


Figura 29: Interfaz del lenguaje de señas.

En la sección de configuración el usuario podrá editar su edad, el sexo, y personalizar el fondo de toda la aplicación con colores llamativos. En esta sección la aplicación puede ser también restaurada, es decir, se eliminan todos los datos del usuario y volverá a iniciar desde cero. Para que la experiencia del usuario sea agradable los cambios son guardados automáticamente. En la figura 30 observamos la interfaz de la configuración cambiando el color de fondo, además de editar la edad y el sexo.



Figura 30: Interfaz de la configuración.



5.3.1 Estadísticas de uso

Para mejorar la aplicación es necesario que los usuarios permitan el envío de datos, estos pueden incluir preferencias de los usuarios en las secciones que ofrece la aplicación como: comunicación, aprendizaje y lenguaje de señas.

Es necesario conocer como los usuarios utilizan la aplicación, saber el tiempo y que secciones son más utilizadas, para mejora de la aplicación.

La aplicación contara los segundos cuando el usuario este en una sección y enviara datos de forma semanal a un servidor.

Los resultados de uso fueron los siguientes:

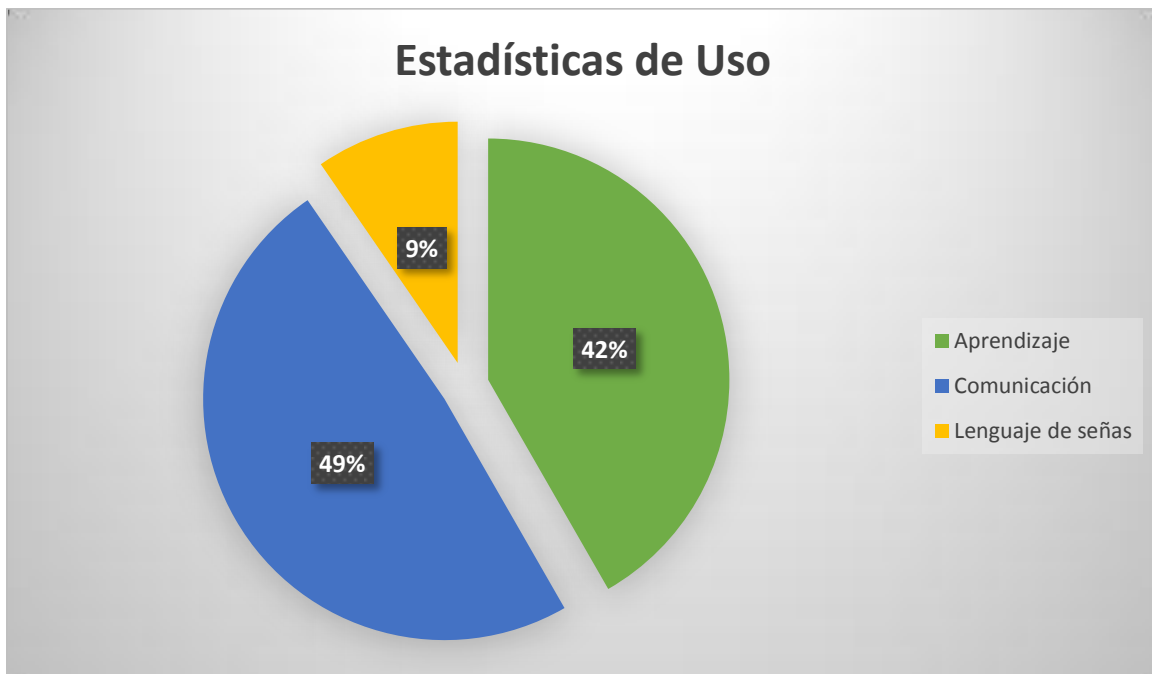
Sección de Aprendizaje: 10114 segundos que equivalen a 2.8094444 horas.

Sección de Comunicación: 11809 segundos que equivalen a 3.2802778 horas.

Sección de Lenguaje de Señas: 2328 segundos que equivalen a 0.6466667 horas.

A continuación, se muestra gráficamente que secciones son más utilizadas por los usuarios.

Grafica de Estadísticas de uso





De la gráfica podemos observar como la aplicación es usada por los usuarios, la sección de comunicación es del 49% mientras que la de aprendizaje es del 42% por último, la sección de lenguaje de señas es del 9%.

Otra forma de enviar estadísticas de uso es mediante Google Analytics, Google Analytics envía de manera remota datos que pueden ser: preferencias de los usuarios, tiempo que los usuarios visitan las vistas de la aplicación, países donde los usuarios visitan la aplicación, así como la visualización de usuarios en tiempo real.

Los datos recopilados son necesarios para conocer como los usuarios utilizan la aplicación, saber el tiempo y que secciones son más utilizadas, todo esto es para mejora de la aplicación.

Para monitorear cualquier aplicación en Android es necesario tener una cuenta en Google Analytics y registrar cualquier aplicación multiplataforma, posteriormente se asocia un ID con la aplicación.

5.4 Fase Control y Cierre

Las pruebas unitarias que se utilizaron en la aplicación “Tablero de Comunicación” es para definir particiones de equivalencia los cuales son:

- grupos de datos de entrada con un comportamiento similar
- escenarios válidos
- entradas no válidas.

En el caso de la aplicación, “Tablero de Comunicación” cuando el usuario entra por primera vez, se hace una entrada válida para elegir género (sexo) y edad en un rango de 4 a 18 años, en caso de que el usuario no ingrese algún dato la entrada no será válida. Se tiene que tomar en cuenta que el análisis de pruebas es un trabajo que debe ser exhaustivo y completo, sin ellos el usuario al presentarse errores en la aplicación puede confundirse con facilidad y la aplicación no será interactiva y de bajo interés para los usuarios.

Para las estadísticas de uso saldrá una alerta que pedirá la confirmación de datos de envío, si el usuario omite esta alerta, la aplicación bloquea el envío de datos por la red.

Es por ello que debemos hacer énfasis en entradas que puedan alterar alguna funcionalidad de la aplicación móvil pues normalmente es en donde una mayor cantidad de errores se encuentran.

Si el usuario omite los datos de entrada puede causar errores, pues la aplicación depende totalmente de los datos para sea personalizable, es por ello, si el usuario no ingresa datos por cualquier motivo la aplicación móvil inicializa por default una edad y sexo esto es con el propósito de que la aplicación sea estable y no



contenga errores, sin embargo, deben realizarse bastantes pruebas exhaustivas para detectar el mayor número de errores y corregirlos.



6. Implementación de la Aplicación Móvil

En esta sección se mostrarán los detalles y partes de código más sobresalientes de cómo fue desarrollado la aplicación móvil.

Se creó un nuevo Proyecto en Android Studio el cual se nombró “Tablero de Comunicación” con nombre de paquete “cerenetablero.mx”, enseguida se edita el archivo Manifest.xml, es necesario incluir varios permisos para el correcto funcionamiento de la aplicación, tal es el caso de acceso a internet para el “Reconocimiento de Voz” y envió de datos de manera remota, además de incluir un soporte para distintos tamaños de pantalla.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="mx.cerenetablero">
    <supports-screens
        android:anyDensity="true"
        android:largeScreens="true"
        android:normalScreens="true"
        android:smallScreens="true"
        android:xlargeScreens="true" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.VIBRATE" />
```

Para introducir texto mediante voz se ha usado la API de Google TextToSpeech y que requiere que el dispositivo móvil tenga instalada la aplicación de sintetizador de voz de Google. esta clase fue utilizada en todas las Activity, a continuación, se muestra el código.

```
public class TextToSpeech implements
    TextToSpeech.OnInitListener
{
    private TextToSpeech textToSpeech = null;
    public Sintetizador(Context context)
    {
        textToSpeech = new TextToSpeech(context, this);
    }
    public void onInit(int status) {
        if (status == TextToSpeech.SUCCESS) {
            Locale loc = new Locale("spa", "MX");
            int result = textToSpeech.setLanguage(loc);
            if (result == TextToSpeech.LANG_MISSING_DATA
                || result == TextToSpeech.LANG_NOT_SUPPORTED) {
                Log.e("error", "This Language is not supported");
            } else;
        } else {
            Log.e("error", "Initilization Failed!");
        }
    }
    public void shutDown() {
        textToSpeech.shutdown();
    }
    public void hablar(String text) {
        //en esta parte, el texto es convertido a voz
        textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, null);
    }
}
```



Para que la aplicación funcione de manera personalizada es necesario tener una Tabla Hash que funcione como una base de datos local, para que no pueda ser eliminada cada vez que la aplicación cierre o finalice, será guardado como un objeto en un archivo local.

Una Tabla Hash es una estructura de datos que asocia llaves o claves con valores.

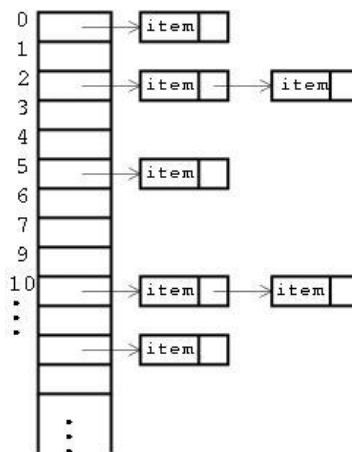


Figura 31: Representación gráfica de una Tabla Hash

Para personalizar datos como: colores, edad y genero el usuario debe entrar a la sección de “Configuración”, la tabla hash es utilizada para guardar las preferencias del usuario, a continuación, se muestra la clase Configuración del proyecto.

```
public class Configuracion extends AppCompatActivity implements
NumberPicker.OnValueChangeListener{
    RelativeLayout r;

    //definimos una variable de tipo relative para el color de fondo
    int edad;
    //definimos radiobutton de genero
    private RadioButton r1,r2;

    Definimos una tabla hash de forma global para guardar datos
    Hashtable<String ,Object> seleccion=new Hashtable<String,Object>();
    int switch_b;
    Toolbar toolbar;
    int color; //= Color.parseColor("#438cca");
    static Dialog d ;
    Button tv1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_configuracion);
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        //leemos la tabla hash que está guardado en el archivo.
```



```
Hashtable<String ,Object> sel=new Hashtable<String,Object>();
    sel=lee_hash();
    setTitle("Regresar");
tv1=(Button) findViewById(R.id.boton_edad);
    r=(RelativeLayout) findViewById(R.id.configuracion);
//verificamos en la tabla hash el color que está guardado
    if(sel.get("color")==null);else {
        toolbar.setBackgroundColor((int) sel.get("color"));
        if((int)sel.get("color") == Color.parseColor("#438cca"))
            r.setBackgroundResource(R.drawable.degradado_azul);
        else if((int)sel.get("color") == Color.parseColor("#e888c4"))
            r.setBackgroundResource(R.drawable.degradado_rosa);
        else if((int)sel.get("color") == Color.parseColor("#5db85d"))
            r.setBackgroundResource(R.drawable.degradado_verde);
        else if((int)sel.get("color") == Color.parseColor("#F50909"))
            r.setBackgroundResource(R.drawable.degradado_rojo);
        else if((int)sel.get("color") == Color.parseColor("#a37f4f"))
            r.setBackgroundResource(R.drawable.degradado_cafe);
        else if((int)sel.get("color") == Color.parseColor("#740cd6"))
            r.setBackgroundResource(R.drawable.degradado_morado);
        else if((int)sel.get("color") == Color.parseColor("#ecce20"))
            r.setBackgroundResource(R.drawable.degradado_amarillo);
        else if((int)sel.get("color") == Color.parseColor("#494949"))
            r.setBackgroundResource(R.drawable.degradado_negro);
        else if((int)sel.get("color") == Color.parseColor("#FFA420"))
            r.setBackgroundResource(R.drawable.degradado_naranja);
    }
    RadioGroup contenedor = (RadioGroup)
findViewById(R.id.radioGroup2);

//verificamos de la tabla hash cual es el género
    if((int)sel.get("genero") == 0){
        //niño
        RadioButton opcionI2 = (RadioButton) contenedor.getChildAt(0);
        opcionI2.setChecked(true);
    }
    else if((int)sel.get("genero") == 1){
        //niña
        RadioButton opcionI2 = (RadioButton) contenedor.getChildAt(1);
        opcionI2.setChecked(true);
    }
    if(sel.get("edad")==null)
    {
    }else{
        edad = (int) sel.get("edad");
        tv1.setText(edad+"");
    }
    r1=(RadioButton) findViewById(R.id.r1);
    r2=(RadioButton) findViewById(R.id.r2);
}
@Override

public void save2(int gen)
{

//en esta función guardamos el color, género y edad que el usuario eligió.
if(color==Color.parseColor("#ffffff")){
    seleccion.put("genero", gen);

    seleccion.put("edad", edad);
    guarda_seleccion();
}else
{
```



```
seleccion.put("genero", gen);
seleccion.put("color", color);
seleccion.put("edad", edad);
guarda_seleccion();}}
public void restaurar2()
{

//esta función borra todos los datos de la aplicación
AlertDialog.Builder adb = new AlertDialog.Builder(this);
adb.setTitle("¿Desea restaurar la aplicación?");
adb.setMessage("¡Al restaurar la aplicación se borrarán todos los
datos!");
adb.setIcon(R.drawable.alert);
adb.setPositiveButton("Si", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
        clearApplicationData();
        Intent in = new
Intent(getApplicationContext(),MainActivity.class).addFlags(Intent.FLAG_ACT
IVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP|Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(in);
        finish();
    } });
    adb.setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
        } });adb.show();
}
public void guarda_seleccion() {

//esta función guarda la tabla hash como un objeto
try {
    FileOutputStream fileOut = openFileOutput("hash.obj",
Context.MODE_PRIVATE);
    ObjectOutputStream out = new ObjectOutputStream(fileOut);
    out.writeObject(seleccion);
    out.close();
    fileOut.close();
} catch (FileNotFoundException e) {
    Toast.makeText(getApplicationContext(), "error2",
Toast.LENGTH_LONG).show();
} catch (IOException e) {
    Toast.makeText(getApplicationContext(), "error3",
Toast.LENGTH_LONG).show();
}}
public Hashtable lee_hash()
{

//esta función es de tipo tabla hash, lee la tabla hash desde archivo
Hashtable<String,Object> seleccion=null;
try{
    //FileInputStream fileIn = new FileInputStream("open.pref");
    FileInputStream fileIn = openFileInput("hash.obj");
    ObjectInputStream in = new ObjectInputStream(fileIn);

    seleccion = (Hashtable<String,Object>) in.readObject();
    in.close();
    fileIn.close();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} catch (FileNotFoundException e) {
} catch (IOException e) {
    Toast.makeText(getApplicationContext(), "error3.2",
```



```
Toast.LENGTH_LONG).show();
    }
    return seleccion;
}
//las siguientes funciones guardan el color de la preferencia del usuario.
public void azul(View v) {
    r.setBackgroundResource(R.drawable.degradado_azul);
    color= Color.parseColor("#438cca"); toolbar.setBackgroundColor(
Color.parseColor("#438cca"));
}
public void rojo(View v) {
    r.setBackgroundResource(R.drawable.degradado_rojo);
    color=
Color.parseColor("#F50909"); toolbar.setBackgroundColor(Color.parseColor("#F
50909"));
}
public void verde(View v) {
    r.setBackgroundResource(R.drawable.degradado_verde);
    color=
Color.parseColor("#5db85d"); toolbar.setBackgroundColor(Color.parseColor("#5
db85d"));
}
public void cafe(View v) {
    r.setBackgroundResource(R.drawable.degradado_cafe);
    color= Color.parseColor("#a37f4f"); toolbar.setBackgroundColor(
Color.parseColor("#a37f4f"));
}
public void negro(View v) {
    r.setBackgroundResource(R.drawable.degradado_negro);
    color= Color.parseColor("#494949"); toolbar.setBackgroundColor(
Color.parseColor("#494949"));
}
public void morado(View v) {
    r.setBackgroundResource(R.drawable.degradado_morado);
    color= Color.parseColor("#740cd6"); toolbar.setBackgroundColor(
Color.parseColor("#740cd6"));
}
public void naranja(View v) {
    r.setBackgroundResource(R.drawable.degradado_naranja);
    color= Color.parseColor("#FFA420"); toolbar.setBackgroundColor(
Color.parseColor("#FFA420"));
}
public void rosa(View v) {
    r.setBackgroundResource(
        R.drawable.degradado_rosa);
    color= Color.parseColor("#e888c4"); toolbar.setBackgroundColor(
Color.parseColor("#e888c4"));
}
public void amarillo(View v) {

    r.setBackgroundResource(R.drawable.degradado_amarillo);
    color= Color.parseColor("#ecce20"); toolbar.setBackgroundColor(
Color.parseColor("#ecce20"));
}
public void default1(View v) {
    r.setBackgroundColor(Color.WHITE);
    color= Color.parseColor("#ffffff"); toolbar.setBackgroundColor(
Color.parseColor("#3c5a99"));
}
//función para regresar con la tecla de atrás.
@Override
public void onBackPressed() {
    if (r1.isChecked() == true) {
```



```
        save2(0); // gen = 0; //
    } else if (r2.isChecked() == true) {
        save2(1); // gen = 1; //
        Intent in = new
Intent(getApplicationContext(), MainActivity.class).addFlags(Intent.FLAG_ACT
IVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_SINGLE_TOP|Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(in);
        finish();
    }
    public void onValueChange(NumberPicker picker, int oldVal, int newVal)
{
    }
    public void show(View v) {

//Esta función muestra un number picker para elección de edad del usuario
        final Dialog d = new Dialog(Configuracion.this);
        d setContentView(R.layout.dialog);
        Button b1 = (Button) d.findViewById(R.id.button1);
        Button b2 = (Button) d.findViewById(R.id.button2);
        final NumberPicker np = (NumberPicker)
d.findViewById(R.id.numberPicker1);
        np.setMaxValue(18); // max value 18
        np.setMinValue(4); // min value 4
        np.setWrapSelectorWheel(false);
        np.setOnValueChangedListener(this);
        b1.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v) {
                tv1.setText(String.valueOf(np.getValue()));
                edad=Integer.parseInt(String.valueOf(np.getValue()));
                d.dismiss();
            }
        });
        b2.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v) {
                d.dismiss(); // dismiss the dialog
            }
        });
        d.show();
    public void clearApplicationData() {
        File cache = getCacheDir();
        File appDir = new File(cache.getParent());
        if(appDir.exists()){
            String[] children = appDir.list();
            for(String s : children){
                if(!s.equals("lib")){
                    deleteDir(new File(appDir, s));
                }
            }
        }
    }

    public static boolean deleteDir(File dir) {
        if (dir != null && dir.isDirectory()) {
            String[] children = dir.list();
            for (int i = 0; i < children.length; i++) {
                boolean success = deleteDir(new File(dir, children[i]));
                if (!success) {
                    return false;
                }
            }
            return dir.delete();
        }
    }
}
```



En el código de la clase Operaciones al generar operaciones para los usuarios es necesario que los resultados sean números enteros, divisiones exactas y evitando números negativos.

Para calcular la dificultad de las operaciones, se toma como referencia la edad del usuario, es por ello que se crea la función “edadfunction” para calcular la dificultad, si posteriormente el usuario desea cambiar de nivel podrá hacerlo dando click en la esquina superior derecha, el usuario podrá elegir distintos niveles de dificultad, los cuales son: “fácil”, “medio fácil”, medio,” medio difícil”, “difícil”, según la preferencia del usuario será la dificultad de la misma.

Iniciando con un arreglo de imágenes e intercambiando las imágenes de forma aleatoria.

```
int[] manzanas= {R.drawable.man1,R.drawable.man2, R.drawable.man3,
R.drawable.man4, R.drawable.man5, R.drawable.man6
    , R.drawable.man7 ,R.drawable.man8 ,R.drawable.man9 , R.drawable.ma10};

int[] peras= {R.drawable.p1, R.drawable.p2, R.drawable.p3, R.drawable.p4,
R.drawable.p5, R.drawable.p6
    , R.drawable.p7 ,R.drawable.p8 , R.drawable.p9 , R.drawable.p10};

int[] estrellas= {R.drawable.e1, R.drawable.e2, R.drawable.e3, R.drawable.e4,
R.drawable.e5, R.drawable.e6
    ,R.drawable.e7 ,R.drawable.e8 ,R.drawable.e9 ,R.drawable.e10};

int[] balones= {R.drawable.b1, R.drawable.b2, R.drawable.b3, R.drawable.b4,
R.drawable.b5, R.drawable.b6
    , R.drawable.b7 , R.drawable.b8 , R.drawable.b9 , R.drawable.b10};

int[] arboles= {R.drawable.a1, R.drawable.a2,R.drawable.a3,
R.drawable.a4,R.drawable.a5, R.drawable.a6
    , R.drawable.a7 , R.drawable.a8 ,R.drawable.a9 , R.drawable.a10};

public void suma () {
    identifica=1;
    b1.setBackgroundResource (R.drawable.suma2);
    b2.setBackgroundResource (R.drawable.resta3);
    b3.setBackgroundResource (R.drawable.multiplica3);
    b4.setBackgroundResource (R.drawable.division3);
    simbolo.setBackgroundResource (R.drawable.suma);
    tipo.setText ("Suma");
    n1=new Random ();
    n2=new Random ();
    obj=new Random ();
    num1=n1.nextInt (edadfunction ());
    num2=n2.nextInt (edadfunction ());
    int objeto=obj.nextInt (5);
    int temp=0;
    if (objeto==0) {
        obj1.setBackgroundResource (manzanas [num1]);
        obj2.setBackgroundResource (manzanas [num2]);
    }
    else if (objeto==1) {
        obj1.setBackgroundResource (peras [num1]);
        obj2.setBackgroundResource (peras [num2]);
    }
    else if (objeto==2) {
        obj1.setBackgroundResource (estrellas [num1]);
    }
}
```



```
        obj2.setBackgroundResource(estrellas[num2]);
    }
    else if (objeto==3){
        obj1.setBackgroundResource(balones[num1]);
        obj2.setBackgroundResource(balones[num2]);
    }
    else if (objeto==4){
        obj1.setBackgroundResource(arboles[num1]);
        obj2.setBackgroundResource(arboles[num2]);}
}
public void resta(View v){
    simbolo.setBackgroundResource(R.drawable.resta);
    tipo.setText("Resta");
    identifica=2;
    n1=new Random();
    n2=new Random();
    obj=new Random();
    b1.setBackgroundResource(R.drawable.suma3);
    b2.setBackgroundResource(R.drawable.resta2);
    b3.setBackgroundResource(R.drawable.multiplica3);
    b4.setBackgroundResource(R.drawable.division3);
    num1=1;num2=2;
    while (true){
        num1=n1.nextInt(edadfunction());
        num2=n2.nextInt(edadfunction());
        if (num1>num2||num1==num2){
            break;}
    }
    int objeto=obj.nextInt(5);
    int temp=0;
    if (objeto==0){
        obj1.setBackgroundResource(manzanas[num1]);
        obj2.setBackgroundResource(manzanas[num2]);
    }
    else if (objeto==1){
        obj1.setBackgroundResource(peras[num1]);
        obj2.setBackgroundResource(peras[num2] (num1,num2,"peras","resta"));
    }
    else if (objeto==2){
        obj1.setBackgroundResource(estrellas[num1]);
        obj2.setBackgroundResource(estrellas[num2]);
    }
    else if (objeto==3){
        obj1.setBackgroundResource(balones[num1]);
        obj2.setBackgroundResource(balones[num2]);
    }
    else if (objeto==4){
        obj1.setBackgroundResource(arboles[num1]);
        obj2.setBackgroundResource(arboles[num2]);
        sintetiza(num1,num2,"balones","suma");
    }
}
public void multiplica(View v){
    identifica=3;
    b1.setBackgroundResource(R.drawable.suma3);
    b2.setBackgroundResource(R.drawable.resta3);
    b3.setBackgroundResource(R.drawable.multiplica2);
    b4.setBackgroundResource(R.drawable.division3);
    n1=new Random();
    n2=new Random();
    obj=new Random();
    simbolo.setBackgroundResource(R.drawable.multiplica);
    tipo.setText("Multiplica");
```



```
num1=n1.nextInt(edadfunction());
num2=n2.nextInt(edadfunction());
int objeto=obj.nextInt(5);
int temp=0;
if (objeto==0){
    obj1.setBackgroundResource(manzanas[num1]);
    obj2.setBackgroundResource(manzanas[num2]);
}
else if (objeto==1){
    obj1.setBackgroundResource(peras[num1]);
    obj2.setBackgroundResource(peras[num2]);
}
else if (objeto==2){
    obj1.setBackgroundResource(estrellas[num1]);
    obj2.setBackgroundResource(estrellas[num2]);
}
else if (objeto==3){
    obj1.setBackgroundResource(balones[num1]);
    obj2.setBackgroundResource(balones[num2]);
}
else if (objeto==4){
    obj1.setBackgroundResource(arboles[num1]);
    obj2.setBackgroundResource(arboles[num2]);
}
}
}
public void suma2(View v){
    suma();
}
public void divide(View v){
    identifica=4;
    n1=new Random(); n2=new Random(); obj=new Random();
    b1.setBackgroundResource(R.drawable.suma3);
    b2.setBackgroundResource(R.drawable.resta3);
    b3.setBackgroundResource(R.drawable.multiplifica3);
    b4.setBackgroundResource(R.drawable.division2);
    simbolo.setBackgroundResource(R.drawable.division);
    tipo.setText("Divide");
    int objeto=obj.nextInt(5);
    int temp=0;
    if (objeto==0){
        while (true){
            num1=n1.nextInt(edadfunction());
            num2=divisor(num1);
            if (num1>num2 || num1==num2)
                break;
            obj1.setBackgroundResource(manzanas[num1]);
            obj2.setBackgroundResource(manzanas[num2]);
        }
    }
    else if (objeto==1){
        while (true){
            num1=n1.nextInt(10);
            num2=divisor(num1);
            if (num1>num2 || num1==num2)
                break;
        }
        obj1.setBackgroundResource(peras[num1]);
        obj2.setBackgroundResource(peras[num2]);
    }
    else if (objeto==2){
        while (true){
            num1=n1.nextInt(10);

```



```
        num2=divisor(num1);
        if (num1>num2 || num1==num2) {
            break;
        }
    }
    obj1.setBackgroundResource(estrellas[num1]);
    obj2.setBackgroundResource(estrellas[num2]);
}
else if (objeto==3) {
    while (true) {
        num1=n1.nextInt(10);
        num2=divisor(num1);
        if (num1>num2 || num1==num2) {
            break;
        }
    }
    obj1.setBackgroundResource(balones[num1]);
    obj2.setBackgroundResource(balones[num2]);
}
else if (objeto==4) {
    while (true) {
        num1=n1.nextInt(10);
        num2=divisor(num1);
        if (num1>num2 || num1==num2) {
            break;
        }
    }
    obj1.setBackgroundResource(arboles[num1]);
    obj2.setBackgroundResource(arboles[num2]);
}
}
public void verifica(View v) {
    int temp=0;
    if
(resultadoFinal.getText().toString().matches("") || resultadoFinal.getText().equals
("")) | resultadoFinal.getText().toString() == null || resultadoFinal.getText().toString
().length() == 0 || resultadoFinal.getText().toString().isEmpty()) {
        temp=0;
        Snackbar.make(v, "Ingresa un número", Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
        textToSpeech.speak("Ingresa un número", TextToSpeech.QUEUE_FLUSH, null);
    } else {
        temp = Integer.parseInt(resultadoFinal.getText().toString());
        if (identifica == 1) {
            if (temp == num1 + num2 + 2) {
                Snackbar.make(v, "Respuesta Correcta, el resultado es " + temp,
                Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
                textToSpeech.speak("Respuesta Correcta, el resultado es " + temp,
                TextToSpeech.QUEUE_FLUSH, null);
            } else {
                Snackbar.make(v, "Respuesta incorrecta, vuelve a intentarlo",
                Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
                textToSpeech.speak("Respuesta incorrecta, vuelve a intentarlo",
                TextToSpeech.QUEUE_FLUSH, null);
            }
        }
        if (identifica==2) {
            if (temp == (num1 + 1) - (num2 + 1)) {
                Snackbar.make(v, "Respuesta Correcta, el resultado es " + temp,
                Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
                textToSpeech.speak("Respuesta Correcta, el resultado es " + temp,
```



```
TextToSpeech.QUEUE_FLUSH, null);
    } else {
        Snackbar.make(v, "Respuesta incorrecta, vuelve a intentarlo ",
Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
        textToSpeech.speak("Respuesta incorrecta, vuelve a intentarlo ",
TextToSpeech.QUEUE_FLUSH, null);
    }
}
if (identifica == 3) {
    if (temp == (num1 + 1) * (num2 + 1)) {
        Snackbar.make(v, "Respuesta Correcta, el resultado es " + temp,
Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
        textToSpeech.speak("Respuesta Correcta, el resultado es " + temp,
TextToSpeech.QUEUE_FLUSH, null);
    } else {
        Snackbar.make(v, "Respuesta incorrecta, vuelve a intentarlo ",
Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
        textToSpeech.speak("Respuesta incorrecta, vuelve a intentarlo",
TextToSpeech.QUEUE_FLUSH, null);
    }
}
if (identifica==4) {
    if (temp == (num1 + 1) / (num2 + 1)) {
        Snackbar.make(v, "Respuesta Correcta, el resultado es " + temp,
Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
        textToSpeech.speak("Respuesta Correcta, el resultado es " + temp,
TextToSpeech.QUEUE_FLUSH, null);
    } else {
        Snackbar.make(v, "Respuesta incorrecta, vuelve a intentarlo ",
Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
        textToSpeech.speak("Respuesta incorrecta, vuelve a intentarlo ",
TextToSpeech.QUEUE_FLUSH, null);
    }
}
}}}
```

Se muestra el código de la función “edadfunction”, además del código para cambiar de nivel.

```
public int edadfunction()
{
    if(dificultad==0) {
        if (edad == 4) {
            return 2;
        } else if (edad > 4 && edad < 8) {
            return 4;
        } else if (edad >= 8 && edad < 12) {
            return 6;
        } else if (edad >= 12 && edad < 15) {
            return 8;
        } else {
            return 10;
        }
    }
    else
    {
        return dificultad;
    }
}
```



```
public void nivel()
{
    final CharSequence[] items = { "Fácil", "Medio Fácil", "Medio", "Medio
Difícil", "Difícil" };
    AlertDialog.Builder builder = new AlertDialog.Builder(this); //ERROR
ShowDialog cannot be resolved to a type
    builder.setTitle("Elige el nivel de dificultad:");

    builder.setSingleChoiceItems(items, -1,
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int item) {
                aux=item+1;
            }
        });
    builder.setPositiveButton("Aceptar", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            dificultad=aux;
            if (dificultad==1) {
                dificultad=2;
            } else if (dificultad==2) {
                dificultad=4;
            } else if (dificultad==3) {
                dificultad=6;
            } else if (dificultad==4) {
                dificultad=8;
            } else if (dificultad==5) {
                dificultad=10;
            }
        }
    });
    builder.setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            dificultad=0;
        }
    });
    AlertDialog alert = builder.create();
    alert.show();
}
```

Para guardar elementos favoritos se implementó otra tabla Hash, para editar y eliminar los elementos, se implementó un menú contextual que se activa cuando el usuario da un largo click en cualquier elemento de los favoritos.

```
public Hashtable lee_fav()
{
    Hashtable<String,String> seleccion=null;
    try{//FileInputStream fileIn = new FileInputStream("open.pref");
        FileInputStream fileIn = openFileInput("hash_fav.obj");
        ObjectInputStream in = new ObjectInputStream(fileIn);

        seleccion = (Hashtable<String,String>) in.readObject();
        in.close();
        fileIn.close();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (FileNotFoundException e) {

    } catch (IOException e) {
        Toast.makeText(getApplicationContext(),
"error3.2",Toast.LENGTH_LONG).show();
    }
    return seleccion;
}
```



```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                               ContextMenu.ContextMenuInfo menuInfo)
{
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    if(v.getId() == R.id.listView)
    {
        AdapterView.AdapterContextMenuInfo info =
            (AdapterView.AdapterContextMenuInfo)menuInfo;
        palabra=itemlist.getAdapter().getItem(info.position).toString();
        menu.setHeaderTitle("Elija una opción:");
        inflater.inflate(R.menu.menu_fav2, menu);
    }
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    AdapterView.AdapterContextMenuInfo info =
        (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
    switch (item.getItemId()) {
        case R.id.mnuEditar:
            edita(palabra);
            return true;
        case R.id.mnuEliminar:
            elimina(palabra);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

public void elimina(String favorito)
{
    final String text=favorito;
    AlertDialog.Builder dialog01 = new AlertDialog.Builder(this);
    dialog01.setTitle("¡¡Cuidado!!");
    dialog01.setIcon(R.drawable.alert);
    dialog01.setMessage("¿ Desea realmente eliminar '"+text+"' de
    favoritos ?");
    dialog01.setCancelable(false);
    dialog01.setPositiveButton("Confirmar", new
    DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog01, int id) {
            Enumeration e = seleccion.keys();
            String llave="";
            while( e.hasMoreElements() ){
                llave=e.nextElement().toString();
                if(llave.contains(text)){
                    seleccion.remove(llave);
                    guarda_fav();
                    break;}}
                finish(); startActivity(getIntent());
            }
        });
    dialog01.setNegativeButton("Cancelar", new
    DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog01, int id) {
            }
        });
    dialog01.show();
}
```



```
public void edita(String texto)
{
    final String texto1=texto;
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    alert.setTitle("Editar");
    alert.setIcon(R.drawable.editar);
    alert.setMessage("Edite su favorito");
    final EditText input = new EditText(this);
    int maxLength = 70;
    InputFilter[] fArray = new InputFilter[1];
    fArray[0] = new InputFilter.LengthFilter(maxLength);
    input.setFilters(fArray);
    alert.setView(input);
    input.setText(texto);
    alert.setPositiveButton("Editar", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int whichButton) {
            final String editada;
            editada = input.getText().toString();
            final String aux;
            Enumeration e = seleccion.keys();
            String llave="";
            while( e.hasMoreElements() ){
                llave=e.nextElement().toString();
                if(llave.contains(texto1)){
                    seleccion.remove(llave);
                    aux=editada+"-";
                    seleccion.put(aux, aux);
                    guarda_fav();
                    break;
                }
            }
            finish(); startActivity(getIntent());
        }
    });
    alert.setNegativeButton("Cancelar", new DialogInterface.OnClickListener()
    {
        public void onClick(DialogInterface dialog, int whichButton) {}
    });
    alert.show();
}
```

Para que la clase principal funcionara como una pantalla principal, se utilizó la clase Thread para dormir durante 2.5 segundos a la clase Principal y así entrar a la Activity del menú Principal.

```
public class Inicio extends AppCompatActivity {

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inicio);
        getSupportActionBar().setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
        getSupportActionBar().setCustomView(R.layout.abs_main);
        setTitle("Inicio");
        Thread timer = new Thread(){
            public void run(){
                try{
                    sleep(2500);
                }catch(InterruptedException e){e.printStackTrace();}
                finally{
                    main();
                }
            }
        };
        timer.start();}
}
```



```
public void main()
{
    Intent intent = new Intent(this, MainActivity.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent); finish();
}
}
```

Para el código de sonidos de Animales se utilizó la variable Media Player, además del Sintetizador de Voz para identificar al animal y un control de volumen, a continuación, se muestra el código:

```
public void sonido(String animal, MediaPlayer m)
{
    if (mp.isPlaying() == true) {
    } else {
        mp.pause();
        textToSpeech.speak(animal + " ", TextToSpeech.QUEUE_FLUSH, null);
        mp = m;
        mp.start();
    }
}

private void Volumen() {
    try {
        sk_volumen = (SeekBar) findViewById(R.id.seekBar);
        audioManager = (AudioManager)
            getSystemService(Context.AUDIO_SERVICE);

        sk_volumen.setMax(audioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC));

        sk_volumen.setProgress(audioManager.getStreamVolume(AudioManager.STREAM_MUSIC));

        sk_volumen.setOnSeekBarChangeListener(new
        SeekBar.OnSeekBarChangeListener() {
            @Override public void
            onStopTrackingTouch(SeekBar arg0) {
            }

            @Override public void
            onStartTrackingTouch(SeekBar arg0) {
            }

            @Override public void
            onProgressChanged(SeekBar arg0, int progress, boolean arg2) {
                audioManager.setStreamVolume(AudioManager.STREAM_MUSIC, progress,
                0);
            }
        });
    } catch (Exception e)
    {
        e.printStackTrace();
    }
}
```



Para mostrar la sección de “Acerca”, se hace uso de la herramienta Alert Dialog.

```
public Acerca(Context context)
{ AlertDialog alertDialog = new AlertDialog.Builder(context).create();
  alertDialog.setTitle("Acerca de");
  alertDialog.setMessage("Aplicación desarrollada por el equipo de Cerene
Interactivo.");
  alertDialog.setButton("Aceptar", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
    }
  });
  alertDialog.show();
}
```

Se muestra a continuación el código de la Api de Google de reconocimiento de voz que se usó en la aplicación.

```
public void click(View view) {
  Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
  intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "es-MX");
  intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
  try {
    startActivityForResult(intent, 1);
  } catch (ActivityNotFoundException a) {
    Toast.makeText(getApplicationContext(), "Tu dispositivo no soporta el
reconocimiento de voz", Toast.LENGTH_LONG).show();
  }
}
public void onActivityResult(int requestcode, int resultcode, Intent datos)
{
  if (resultcode== Activity.RESULT_OK && datos!=null)
  {
    ArrayList<String>
text=datos.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
    voz2=text.get(0);
    Snackbar.make(boton, "Dijiste "+voz2, Snackbar.LENGTH_LONG)
      .setAction("Action", null).show();
  }
}}
```

Para mostrar letras en el reconocimiento de voz, se implementó un arreglo de imágenes además de un arreglo de String para cada letra del abecedario, es necesario tener un índice para controlar el flujo de las imágenes inicializado en “0” pues es el indicador de la letra “A”, cuando el usuario elija una letra, el índice indicará la posición en el arreglo de String y podrá verificar si lo que dijo concuerda con la letra elegida. A continuación, se muestra el código de los arreglos implementados y el índice.

```
String[] letras2={"A", "BE", "SE", "DE", "E", "F", "G", "H", "I", "J", "K", "L", "M",
, "N", "O", "P", "Q", "R", "ESE", "TE", "U", "V", "W", "X", "Y", "Z"};
int indice=0;
int[] letras= {R.drawable.ar, R.drawable.br, R.drawable.cr,
R.drawable.dr, R.drawable.er, R.drawable.fr
, R.drawable.gr, R.drawable.hr, R.drawable.ir, R.drawable.jr,
R.drawable.kr, R.drawable.lr,
R.drawable.mr, R.drawable.nr, R.drawable.or, R.drawable.pr,
R.drawable.qr, R.drawable.rr, R.drawable.sr,
R.drawable.tr, R.drawable.ur, R.drawable.vr, R.drawable.wr,
R.drawable.xr, R.drawable.yr, R.drawable.zr};
```



Para controlar el índice de las letras fue necesario la implementación de 2 funciones, una función servirá para incrementar el índice y la otra será para disminuir el índice.

```
public void cambia(View view) {
    vibrar(50);
    indice++;
    if(indice==letras.length) {
        indice=0;
    }
    letra_abc.setBackgroundResource(letras[indice]);
    try {
        trimCache(this);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
public void cambia0(View view){
    vibrar(50);
    indice--;
    if(indice<0)
    {
        indice=letras.length-1;
        letra_abc.setBackgroundResource(letras[indice]);
    }
    else
    {
        letra_abc.setBackgroundResource(letras[indice]);
    }
    try {
        trimCache(this);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Además, se implementó una función para verificar lo que dijo el usuario con la letra seleccionada.

```
public void verifica( String texto,String voz)
{
    String aux;
    if(texto.equals(voz)) {
        aux="¡Bien hecho! Dijiste la letra correctamente.";
        hablar(aux);
        Toast.makeText(getApplicationContext(),aux, Toast.LENGTH_LONG).show();
    }
    else{
        aux="Te equivocaste, no dijiste la letra correcta.";
        hablar(aux);
        Toast.makeText(getApplicationContext(),aux, Toast.LENGTH_LONG).show();
    }
}
```

Para la sección de monosílabas, la implementación es similar a la sección de letras, sin embargo, no se implementa un arreglo de imágenes, se implementa una matriz de imágenes, al ser una matriz es necesario tener 2 índices para controlarla, un índice será para controlar las filas que serán las letras y otro índice



será para controlar las monosílabas de cada letra, además de llevar una matriz de imágenes, se implementa una matriz de String de monosílabas de cada letra omitiendo las vocales y un arreglo de letras.

A continuación, se muestra el código de las matrices, arreglos e índices.

```
int indice=1;

int currentIndex=1;
String strings[] =
{"B","C","D","F","G","H","J","K","L","M","N","P","Q","R","S","T","V","W","X","Y",
"Z"}; //21 elementos 0-20
String [][]matriz2={
    {"B","BA","BE","BI","BO","BU"}, //0
    {"C","CA","CE","CI","CO","CU"}, //1
    {"D","DA","DE","DI","DO","DU"}, //2
    {"F","FA","FE","FI","FO","FU"}, //3
    {"G","GA","GE","GI","GO","GU"}, //4
    {"H","HA","HE","HI","HO","HU"}, //5
    {"J","JA","JE","JI","JO","JU"}, //6
    {"K","KA","KE","KI","KO","KU"}, //7
    {"L","LA","LE","LI","LO","LU"}, //8
    {"M","MA","ME","MI","MO","MU"}, //9
    {"N","NA","NE","NI","NO","NU"}, //10
    {"P","PA","PE","PI","PO","PU"}, //11
    {"Q","QA","QE","QI","QO","QU"}, //12
    {"R","RA","RE","RI","RO","RU"}, //13
    {"S","SA","SE","SI","SO","SU"}, //14
    {"T","TA","TE","TI","TO","TU"}, //15
    {"V","VA","VE","VI","VO","VU"}, //16
    {"W","WA","WE","WI","WO","WU"}, //17
    {"X","XA","XE","XI","XO","XU"}, //18
    {"Y","YA","YE","YI","YO","YU"}, //19
    {"Z","ZA","ZE","ZI","ZO","ZU"} //20 };
Object [][] matriz = {
{"B",R.drawable.ba,R.drawable.be,R.drawable.bi,R.drawable.bo,R.drawable.bu}, //0
{"C",R.drawable.ca,R.drawable.ce,R.drawable.ci,R.drawable.co,R.drawable.cu}, //1
{"D",R.drawable.da,R.drawable.de,R.drawable.di,R.drawable.don,R.drawable.du}, //2
{"F",R.drawable.fa,R.drawable.fe,R.drawable.fi,R.drawable.fo,R.drawable.fu}, //3
{"G",R.drawable.ga,R.drawable.ge,R.drawable.gi,R.drawable.go,R.drawable.gu}, //4
{"H",R.drawable.ha,R.drawable.he,R.drawable.hi,R.drawable.ho,R.drawable.hu}, //5
{"J",R.drawable.ja,R.drawable.je,R.drawable.ji,R.drawable.jo,R.drawable.ju}, //6
{"K",R.drawable.ka,R.drawable.ke,R.drawable.ki,R.drawable.ko,R.drawable.ku}, //7
{"L",R.drawable.la,R.drawable.le,R.drawable.li,R.drawable.lo,R.drawable.lu}, //8
{"M",R.drawable.ma,R.drawable.me,R.drawable.mi,R.drawable.mo,R.drawable.mu}, //9
{"N",R.drawable.na,R.drawable.ne,R.drawable.ni,R.drawable.no,R.drawable.nu}, //10
{"P",R.drawable.pa,R.drawable.pe,R.drawable.pi,R.drawable.po,R.drawable.pu}, //11
{"Q",R.drawable.qa,R.drawable.qe,R.drawable.qi,R.drawable.qo,R.drawable.qu}, //12
{"R",R.drawable.ra,R.drawable.re,R.drawable.ri,R.drawable.ro,R.drawable.ru}, //13
{"S",R.drawable.sa,R.drawable.se,R.drawable.sir,R.drawable.so,R.drawable.su}, //14
{"T",R.drawable.ta,R.drawable.te,R.drawable.ti,R.drawable.to,R.drawable.tu}, //15
{"V",R.drawable.va,R.drawable.ve,R.drawable.vi,R.drawable.vo,R.drawable.vu}, //16
{"W",R.drawable.wa,R.drawable.we,R.drawable.wi,R.drawable.wo,R.drawable.wu}, //17
{"X",R.drawable.xa,R.drawable.xe,R.drawable.xi,R.drawable.xo,R.drawable.xu}, //18
{"Y",R.drawable.ya,R.drawable.ye,R.drawable.yi,R.drawable.yo,R.drawable.yu}, //19
{"Z",R.drawable.za,R.drawable.ze,R.drawable.zi,R.drawable.zo,R.drawable.zu} };
```

Para obtener una mejor experiencia de usuario fue necesaria la implementación de un TextSwitcher para controlar las letras elegidas, así como el color de las



mismas y el tamaño, además de hacer uso de la clase Animation para la animación de las letras elegidas. A continuación, se muestra la implementación de las mismas.

```
private TextSwitcher simpleTextSwitcher;
simpleTextSwitcher = (TextSwitcher) findViewById(R.id.textSwitcher);
simpleTextSwitcher.setFactory(new ViewSwitcher.ViewFactory() {
    public View makeView() {
        t = new TextView(Monosilabas.this);
        t.setGravity(Gravity.TOP | Gravity.CENTER_HORIZONTAL);
        t.setTextSize(60); t.setTextColor(Color.parseColor("#3c5a99"));
        return t;});
Animation in = AnimationUtils.loadAnimation(this, android.R.anim.slide_in_left);
Animation out = AnimationUtils.loadAnimation(this,
android.R.anim.slide_out_right);
simpleTextSwitcher.setInAnimation(in); simpleTextSwitcher.setOutAnimation(out);
simpleTextSwitcher.setCurrentText("B");
```

Para el control de las letras con las monosílabas se implementaron 2 funciones, una función sirve para incrementar el índice de las letras y la otra para disminuir el índice, ambas funciones cuando son llamadas inicializan el índice de las monosílabas además pueden disminuir o incrementar el índice de las monosílabas.

Dependiendo del valor de los indicadores, se mostrará la imagen de los arreglos. A continuación, se muestran las dos funciones implementadas.

```
//función 1
public void derecha(View view)
{
    vibrar(50);
    currentIndex++;
    if (currentIndex == messageCount)
        currentIndex = 0;
    simpleTextSwitcher.setText(strings[currentIndex]);

    indice=1;
    letra_principal.setBackgroundResource((int)matriz[currentIndex][indice]);
}

//función 2
public void izquierda(View view) {
    vibrar(50);
    currentIndex--;
    if (currentIndex < 0) {
        currentIndex = messageCount-1;
        simpleTextSwitcher.setText(strings[currentIndex]);    }
    else {
        simpleTextSwitcher.setText(strings[currentIndex]);
    }
    indice=1;
    letra_principal.setBackgroundResource((int)matriz[currentIndex][indice]);
}
```

Para verificar que el resultado sea correcto, una variable guardara lo que el usuario dijo, dependiendo de los apuntadores se verificara que la variable sea igual a la actual posición de la matriz de String de monosílabas.



A continuación, se muestra la función asignada para verificar que los datos sean correctos.

```
public void onActivityResult(int requestCode, int resultCode, Intent datos)
{
    if (resultCode== Activity.RESULT_OK && datos!=null)
    {
        ArrayList<String>
text=datos.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
        String voz2;
        voz2=text.get(0);
        voz2= voz2.toUpperCase();
        if(voz2.equals(matriz2[currentIndex][indice])){
            hablar("Bien hecho dijiste la monosílaba correcta");
        }else
        {
            hablar("Te equivocaste");
        }
        Toast.makeText(getApplicationContext(),"Dijiste: '"+ voz2+"'",
Toast.LENGTH_LONG).show();}
}
```

Para implementar estadísticas de uso por secciones se contará el tiempo en cada una de ellas, por ello es necesario utilizar la clase Time en todas las secciones, a continuación, se muestra la clase Time.

```
public Timer timer = new Timer();
public int segundos=0;
class Time extends TimerTask {
    public void run() {
        segundos++;
        Log.d("MyApp", segundos+"");
    }
}
public void Contar()
{
    this.segundos=0;
    timer = new Timer();
    timer.schedule(new Contador(), 0, 1000);
}
public void Detener() {
    //this.segundos=0;
    timer.cancel();
    timer.purge();
}

public int getSegundos()
{
    return this.segundos;
}

public void reanudar()
{
    timer.cancel();
    timer.purge();
    timer = new Timer();
    timer.schedule(new Contador(), 0, 1000);
}
}
```



Para consumir datos en un servicio web, se utilizó el siguiente código en Java.

```
public void datos(String valor1, String valor2, String valor3){
    new CargarDatos().execute("http://mexilab.com/cerene-
interactivo/App/recibir_datos/"+valor1+"/"+valor2+"/"+valor3);
}

private class CargarDatos extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may be invalid.";
        }
    }
    @Override
    protected void onPostExecute(String result) {
    }
}

private class ConsultarDatos extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may be invalid.";
        }
    }
}

private String downloadUrl(String myurl) throws IOException {
    Log.i("URL", ""+myurl);
    myurl = myurl.replace(" ", "%20");
    InputStream is = null;
    // Only display the first 500 characters of the retrieved
    // web page content.
    int len = 500;
    try {
        URL url = new URL(myurl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setReadTimeout(10000 /* milliseconds */);
        conn.setConnectTimeout(15000 /* milliseconds */);
        conn.setRequestMethod("GET");
        conn.setDoInput(true);
        // Starts the query
        conn.connect();
        int response = conn.getResponseCode();
        Log.d("respuesta", "The response is: " + response);
        is = conn.getInputStream();
        String contentAsString = readIt(is, len);
        return contentAsString;
    } finally {
        if (is != null) {
            is.close();
        }
    }
}

public String readIt(InputStream stream, int len) throws IOException,
UnsupportedEncodingException {
    Reader reader = null;
    reader = new InputStreamReader(stream, "UTF-8");
    char[] buffer = new char[len];
    reader.read(buffer);
    return new String(buffer);
}
```



Usando el MVC php Codeigniter en la aplicación web para recibir datos de la aplicación se implementó un modelo y controlador.

El controlador recibe datos de tiempo de la aplicación, posteriormente serán enviados al modelo para ser procesados e insertados en la base de datos.

```
public function recibir_datos(){
    $id = $this->uri->segment(3);
    $id = str_replace ( "%20"," ", $id);
    $id2 = $this->uri->segment(4);
    $id2 = str_replace ( "%20"," ", $id2);
    $id3 = $this->uri->segment(5);
    $id3 = str_replace ( "%20"," ", $id3);
    echo "Entro al controlador";
    $this->load->model('Modelo_prueba');

    $aplicacion = array(
        'aprendizaje' => $id,
        'comunicacion' => $id2,
        'lenguaje' => $id3);
    $this->Modelo_prueba->insertar($aplicacion);
}
```

Se implementó en el modelo código php para insertar, sumar tiempo y actualizar tiempo en la base de datos, además se creó una función para convertir segundos a horas.

```
function insertar($aplicacion){
    //Ingresa primero los datos del usuario.
    $this->load->database();

    $aprendizaje_segundos=$aplicacion['aprendizaje'];
    $comunicacion_segundos=$aplicacion['comunicacion'];
    $lenguaje_segundos=$aplicacion['lenguaje'];
    $tiempo = $this->db->query("SELECT * FROM aplicacion");
    if($tiempo->num_rows() == 0){

        $aprendizaje_tiempo=self::conversorSegundosHoras($aprendizaje_segundos);

        $comunicacion_tiempo=self::conversorSegundosHoras($comunicacion_segundos);
        $lenguaje_tiempo=self::conversorSegundosHoras($lenguaje_segundos);

        $data1 = array(
            'aprendizaje_segundos' => $aprendizaje_segundos,
            'aprendizaje_tiempo' => $aprendizaje_tiempo,
            'comunicacion_segundos' => $comunicacion_segundos,
            'comunicacion_tiempo' => $comunicacion_tiempo,
            'lenguaje_segundos' => $lenguaje_segundos,
            'lenguaje_tiempo' => $lenguaje_tiempo
        );

        $this->db->insert('aplicacion',$data1);
    }else{
        foreach ($tiempo->result() as $row){
            $aprendizaje_segundos=$aprendizaje_segundos+$row-
            >aprendizaje_segundos;

            $aprendizaje_tiempo=self::conversorSegundosHoras($aprendizaje_segundos);
            $comunicacion_segundos=$comunicacion_segundos+$row-
            >comunicacion_segundos;

            $comunicacion_tiempo=self::conversorSegundosHoras($comunicacion_segundos);
        }
    }
}
```



```

    $lenguaje_segundos=$lenguaje_segundos+$row->lenguaje_segundos;
    $lenguaje_tiempo=self::conversorSegundosHoras($lenguaje_segundos);

    echo "aprendizaje: ".$aprendizaje_segundos." segundos.";
    echo "comunicación: ".$comunicacion_segundos." segundos.";
    echo "lenguaje: ".$lenguaje_segundos." segundos.";

    }
    $this->db->query("UPDATE aplicacion SET
aprendizaje_segundos=".$aprendizaje_segundos."");
    $this->db->query("UPDATE aplicacion SET
aprendizaje_tiempo='".$aprendizaje_tiempo."'");
    $this->db->query("UPDATE aplicacion SET
comunicacion_segundos=".$comunicacion_segundos."");
    $this->db->query("UPDATE aplicacion SET
comunicacion_tiempo='".$comunicacion_tiempo."'");
    $this->db->query("UPDATE aplicacion SET
lenguaje_segundos=".$lenguaje_segundos."");
    $this->db->query("UPDATE aplicacion SET
lenguaje_tiempo='".$lenguaje_tiempo."'");
    }
}
function conversorSegundosHoras($tiempo_en_segundos) {
    $horas = floor($tiempo_en_segundos / 3600);
    $minutos = floor(($tiempo_en_segundos - ($horas * 3600)) / 60);
    $segundos = $tiempo_en_segundos - ($horas * 3600) - ($minutos * 60);

    if($horas<10){
        $horas="0".$horas;
    }if($minutos<10){
        $minutos="0".$minutos;
    }if($segundos<10){
        $segundos="0".$segundos;
    }
    return $horas . ':' . $minutos . ':' . $segundos;
}
}

```

La tabla que almacenara el tiempo de cada sección se llama “aplicación” y contiene campos como:

- Aprendizaje_tiempo: este campo es de tipo Time.
- Aprendizaje_segundos: este campo es de tipo int.
- Comunicación_tiempo: este campo es de tipo Time.
- Comunicación_segundos: este campo es de tipo int.
- Lenguaje_tiempo: este campo es de tipo Time.
- Lenguaje_segundos: este campo es de tipo int.

Tabla aplicación:

Aplicación

Aprendizaje_tiempo	Aprendizaje_segundos	Comunicación_tiempo	Comunicación_segundos	Lenguaje_tiempo	Lenguaje_segundos

No se fue necesaria una vista pues los datos serán mostrados únicamente en la base de datos.



Como alternativa a servicios Web fue Google Analytics, un ID debe estar asociado a una cuenta de Google Analytics, el ID asociado para la aplicación es: UA-87965006-1.

A continuación, se muestra el código que se utilizó para implementar Google Analytics.

```
GoogleAnalytics analytics;  
  
Tracker tracker;  
analytics = GoogleAnalytics.getInstance(this);  
analytics.setLocalDispatchPeriod(1800);  
tracker = analytics.newTracker("UA-87965006-1");  
tracker.enableExceptionReporting(true);  
tracker.enableAdvertisingIdCollection(true);  
tracker.enableAutoActivityTracking(true);
```

7. Pruebas

Uno de los procesos fundamentales del proceso de fabricación es probar el proceso de ejecución del sistema con el fin de encontrar defectos y errores. El proceso de software no termina con la implementación del sistema, se debe ejecutar una fase de pruebas para asegurarse que el sistema no contenga errores y también para comprobar que los requerimientos han sido cubiertos en su totalidad por el sistema desarrollado.



Figura 32: Pruebas de Software

Los documentos de pruebas deben diseñarse en conjunto con el diseño principal del sistema, de esta forma será posible elaborar documentos que relacionen la funcionalidad programada con los requerimientos solicitados en el sistema. Así mismo, los casos de pruebas deben elaborarse teniendo en cuenta que se deben incluir datos de entrada válidos e inválidos, así como elaborar pruebas para casos inesperados.

Existen diversos enfoques para generar pruebas de software, Pressman [38] asegura que el propósito fundamental de una metodología de pruebas se puede resumir en tres reglas:

- El proceso de pruebas consiste en ejecutar las funciones de un programa con la intención de encontrar errores.
- Un buen caso de prueba es aquel que tiene la mayor posibilidad de encontrar un error no detectado todavía.
- Un caso de prueba exitoso es aquel que descubre un error no detectado con anterioridad.



7.1 Pruebas de Black Box Test

Las pruebas de Caja Negra o Black Box Test también llamadas pruebas funcionales se enfocan en crear casos de prueba que cubran todos los requerimientos funcionales del sistema.

Los casos de prueba que cubran todos los requerimientos funcionales del sistema pueden ser:

1. Funcionamiento faltante o incorrecto.
2. Errores de interfaz.
3. Errores en las estructuras internas de datos o en la base de datos.
4. Errores de funcionalidad o de performance.
5. Errores de inicialización y terminación del sistema.

Estas pruebas pueden ejecutarse en las últimas etapas de pruebas, ya que se enfocan en las entradas y salidas de información del sistema.

En la aplicación “Tablero de Comunicación” se realizará una serie de pruebas donde se mostrarán los posibles escenarios, donde las pruebas pueden fallar o ser exitosas.

Prueba de ingreso de datos del usuario.

Ingreso de datos del usuario.		
No.	Ingresar edad y sexo	Validación
1	Edad valida y sexo valido.	Ingreso correcto.
2	Omisión de ingreso de datos	Lanza una excepción la aplicación.

Escenario de ingreso de datos del usuario.

Escenario Ingreso de datos del usuario.		
Sexo	Edad	No. de prueba que satisface
Niño	13	1
-	-	2



Prueba de Configuración.

Configuración.					
No.	Cambiar Edad	Cambiar sexo	Cambiar color	Restaurar Aplicación	Validación
1	Cambio de edad.	--	--	--	Cambios guardados correctamente.
2	--	Cambio de sexo.	--	--	Cambios guardados correctamente.
3	--	--	Cambio de color.	--	Cambios guardados correctamente.
4	--	--	--	Restaurar aplicación	Aplicación restaurada correctamente

Escenario de Configuración.

Escenario Configuración				
Edad	Sexo	Color	Restaurar aplicación.	No. de prueba que satisface
5	--	--	--	1
--	Niña	--	--	2
--	--	Morado	--	3
--	--	--	Si	4

Prueba de autorización de envío de estadísticas de uso.

Autorización de estadísticas de uso.		
No.	Aceptación de estadísticas de uso	Validación
1	El usuario acepta las condiciones de uso.	Ingreso correcto
2	El usuario no acepta las condiciones de uso.	Ingreso correcto

Escenario de autorización de estadísticas de uso.

Escenario de estadísticas de uso.	
Aceptación de estadísticas de uso	No. de prueba que satisface
El usuario acepta las condiciones de uso.	1
El usuario no acepta las condiciones de uso.	2



Prueba de Reconocimiento de Voz.

Reconocimiento de Voz.				
No.	Letras	Monosílabas	Practica Libre	Validación
1	El usuario dice correctamente la letra.	El usuario dice correctamente la monosílaba.	El usuario dice algo de su preferencia	El usuario escucha correctamente lo que dijo y la aplicación valida el resultado.
2	El usuario no dice la letra correctamente.	El usuario no dice la monosílaba correctamente.	El usuario no dice correctamente algo de su preferencia.	El usuario no escucha correctamente lo que dijo y no valida su resultado.

Escenario de reconocimiento de voz.

Escenario de estadísticas de uso.	
El usuario dice algo.	No. de prueba que satisface
El usuario dice la letra "A".	1
El usuario dice la monosílaba "ma"	1
El usuario dice "agua"	1
El usuario intenta decir "TA" y dice "TRA"	2

Pruebas de Agregar Favoritos

Agregar Favoritos.			
No.	El usuario ingresa un favorito valido.	El usuario agrega un favorito de los elementos ya existentes.	Validación
1	El usuario agrega un favorito valido.	--	La palabra ingresada se agrega a la sección de favoritos.
2	--	El usuario elije un elemento ya existente como favorito.	El elemento es agregado a la sección de favoritos.
3	El usuario agrega un favorito vacío.	--	El elemento no es agregado a la sección de favoritos pues es nulo.
4	El usuario agrega un favorito	--	Si hay más de 100 favoritos, el favorito no es ingresado.



Escenario de agregar favoritos.

Escenario de Agregar Favoritos.		
Ingresar un favorito válido.	Ingresar un elemento ya existente.	No. de prueba que satisface
El usuario ingresa el favorito "Hola".	--	1
El usuario ingresa el favorito "".	--	3
--	El usuario está en la sección de "Emociones" y elige la emoción de "estoy feliz".	2
El usuario ingresa el favorito "figura" y la palabra no es ingresada.	--	4

Pruebas de eliminar favoritos

Eliminar Favoritos.		
No.	El usuario elimina un favorito.	Validación
1	El usuario elimina un favorito.	La palabra escogida se elimina de la sección de favoritos.

Escenario de eliminar favoritos.

Escenario de Eliminar Favoritos.	
Elimina un favorito.	No. de prueba que satisface
El usuario elimina el favorito "Hola".	1

Pruebas de Editar Favoritos.

Editar Favoritos.		
No.	El usuario edita un favorito.	Validación
1	El usuario edita un favorito.	La palabra escogida se edita de la sección de favoritos.
2	El usuario edita un favorito, borrándolo de manera total.	Al borrar todo el favorito, sale una alerta para Eliminar el favorito.



Escenario de Editar favoritos.

Escenario de Editar Favoritos.	
Edita un favorito.	No. de prueba que satisface
El usuario edita el favorito “Hola” a “Hola a todos”.	1
El usuario edita el favorito “Hola a todos” a “”.	2

Pruebas de Operaciones

Operaciones.		
No.	El usuario hace una operación.	Validación
1	El usuario hace una operación e ingresa como resultado una letra.	La aplicación pedirá como resultado un número entero.
2	El usuario hace una operación e ingresa como resultado un resultado vacío.	La aplicación pedirá al usuario un número entero como resultado.
3	El usuario ingresa un número entero como resultado, pero sin ser la respuesta correcta.	El usuario deberá volver ingresar un número para validar la operación.
4	El usuario ingresa un número entero como resultado y es la respuesta correcta.	Resultado exitoso.

Escenario de operaciones.

Escenario de Operaciones.	
Ingresar un resultado.	No. de prueba que satisface
El usuario ingresa como resultado la letra “L”.	1
El usuario ingresa no ingresa algún carácter.	2
El usuario ingresa un resultado negativo en una operación de suma.	3
El usuario ingresa como resultado “4” en la operación de suma “2+2”	4



7.2 Pruebas de Unidad

Las pruebas unitarias se enfocan en verificar la unidad más pequeña del diseño de software. En el caso del presente trabajo se efectuaron pruebas unitarias por cada clase del proyecto. Usando el diseño de clases como guía, se ejecutaron pruebas unitarias independientes para cada clase; el resultado esperado en cada clase era la salida esperada para su interacción con todos los demás componentes del sistema.

Las pruebas unitarias que se utilizaron en la aplicación “Tablero de Comunicación” es para definir particiones de equivalencia los cuales son:

- grupos de datos de entrada con un comportamiento similar
- escenarios válidos
- entradas no válidas.

En el caso de la aplicación, “Tablero de Comunicación” cuando el usuario entra por primera vez, se hace una entrada válida para elegir género (sexo) y edad en un rango de 4 a 18 años, en caso de que el usuario no ingrese algún dato la entrada no será válida. Se tiene que tomar en cuenta que el análisis de pruebas es un trabajo que debe ser exhaustivo y completo, sin ellos el usuario al presentarse errores en la aplicación puede confundirse con facilidad y la aplicación no será interactiva y de bajo interés para los usuarios.

Para las estadísticas de uso saldrá una alerta que pedirá la confirmación de datos de envío, si el usuario omite esta alerta, la aplicación bloquea el envío de datos por la red.

Es por ello que debemos hacer énfasis en entradas que puedan alterar alguna funcionalidad de la aplicación móvil pues normalmente es en donde una mayor cantidad de errores se encuentran.

Si el usuario omite los datos de entrada puede causar errores, pues la aplicación depende totalmente de los datos para sea personalizable, es por ello, si el usuario no ingresa datos por cualquier motivo la aplicación móvil inicializa por default una edad y sexo esto es con el propósito de que la aplicación sea estable y no contenga errores, sin embargo, deben realizarse bastantes pruebas exhaustivas para detectar el mayor número de errores y corregirlos.



7.3 Pruebas de Usuario

La aplicación móvil “Tablero de Comunicación” fue probada por usuarios reales, tal es el caso de Carmen una joven adolescente de 17 años que presenta un diagnóstico de pérdida de audición en un 70% aunque tiene restos auditivos, así mismo un grado moderado de mutismo y retraso en el desarrollo psicomotor.

Su nivel socio económico es bajo y no tenía las posibilidades de ir a una escuela regular y mucho menos tener comunicación con sus compañeros, maestros y familiares cercanos (padres, hermano).

Así mismo, en las primeras semanas se trabajó la aplicación móvil “Tablero de comunicación”, para que conociera colores, ubicaciones, diferencias entre aprender desde figuras geométricas, colores, números, animales hasta conocer y practicar la comunicación como; saludos, alimentos, expresar emociones, necesidades, entre otras opciones.

Al iniciar el usuario no emitía sonidos y palabras, se ponía nerviosa al grado de angustiarse, dándole la confianza suficiente se le dieron instrucciones cortas y claras de manera que pudiera entender.

Se empezó a trabajar la parte de comunicación iniciando con el saludo como fue: Hola, Buenos días, Buenas tardes y Buenas noches, en cada término de la palabra se le explicaba de manera breve en qué momento se tenía que decir e intentaba realizar repeticiones de cada palabra.

Cada semana se trabajaba la aplicación “Tablero de comunicación” con el fin de mejorar en base a las necesidades y su lenguaje del usuario desde una acción como es: “quiero cereal”, “estoy alegre”, “me siento cansado” y “quiero correr” entre otros.

En las siguientes semanas se le mostraron a través de la aplicación, imágenes de alimentos como fueron: bebidas, frutas y postres, sin embargo, el usuario seleccionaba lo que le gustaba y empezaba a identificar las imágenes con las letras a través del sonido, mostrando un avance favorable debido a que le llamo la atención, teniendo el interés por saber de la aplicación y de cuáles eran los contenidos que podía repetir.

Además de mostrar cómo se siente y expresar necesidades básicas como: “quiero ir al baño” “quiero dormir” entre otras opciones de higiene personal.

Sin embargo, las posibilidades de comunicación con los demás fue mejorando, con ayuda de la parte de “agregar” escribiendo “hola soy Carmen”, agregándolo a favoritos y utilizando el sintetizador de voz para que los demás escucharan lo que el usuario quiere decir o expresar.

Agregando palabras y frases en favoritos para la facilidad de ir al botón y evitar escribirlo una y otra vez. Por otra parte, en el apartado de lenguaje de señas, se



muestra una serie de imágenes que van en orden alfabético en la que el usuario reforzaba repitiendo la letra.

El usuario empezó a emitir fonemas, palabras y frases con el apoyo de la aplicación móvil “tablero de comunicación”. Durante las últimas pruebas, pudo utilizar de manera autónoma la aplicación expresando lo que pensaba, sentía o de requerir ayuda.



8. Impacto Socioeconómico

El impacto socioeconómico va dirigido principalmente a psicólogos, terapeutas que trabajan en sesiones con niños con diferentes discapacidades, que si bien es cierto existen múltiples herramientas para ayudar a los niños en sus terapias o para ayudarlos a comunicarse, no hay una herramienta móvil que ayude a comunicar a los usuarios que tengan problemas de lenguaje o autistas, con esto se pretende beneficiar a los usuarios para poder concluir la terapia en un tiempo más corto, además de la posibilidad de realizar actividades en casa para así fortalecer sus problemas de lenguaje y comunicación en su vida cotidiana.



9. Conclusiones.

La aplicación móvil, tiene como fin dar la estimulación necesaria para niños con mutismo, diferentes tipos de trastornos, problemas de audición, retraso en el desarrollo psicomotor o alguna otra discapacidad con Necesidades Educativas Especiales.

A través de la aplicación móvil “Tablero de comunicación” se obtuvo resultados favorables sobre el aprendizaje de los diferentes niños y se logró dar un avance positivo de los diferentes problemas relacionados con la comunicación, además de implementar configuraciones personalizadas con diferentes colores, género, edad, la aplicación es apta para todo público y podrá ser usada en diferentes contextos, así mismo puede ser usada de manera agradable por psicólogos y padres de familia, dando diferentes estrategias para estimular el lenguaje y facilitar la comunicación del usuario en el medio que lo rodea.

Con el apoyo de diferentes especialistas del Centro de Rehabilitación Neurológica Infantil (CERENE), se centró en las necesidades de las personas, con información hacia el entorno social y docentes de educación especial, en cuanto a las observaciones y realizando diferentes pruebas para obteniendo resultados favorables.

Es recomendable el uso de la aplicación móvil “Tablero de comunicación” para todas aquellas personas que tengan problemas de lenguaje o alguna discapacidad para mejorar el habla con el entorno social. Empezando desde fonemas, palabras, frases ilustrados con imágenes y tono de voz para la comunicación, hasta lenguaje de señas y diferentes tipos y sonidos de animales.

Como trabajo a futuro, se espera enriquecer el vocabulario del tablero de control para permitir una comunicación más continua de los niños con los demás. Se ha considerado también el uso de algunos videojuegos serios para estimular el habla de los niños.



10. Referencias

- [1] Molina García, S. (1995). Bases psicopedagógicas de la Educación Especial. Marfil, Alcoy.
- [2] King, D. (1990). La aplicación del software en la Educación Especial. *Comunicación, lenguaje y educación*, 2(5), 31-46.
- [3] Lion, C. (2006). Imaginar con tecnologías: relaciones entre tecnologías y conocimiento. Editorial Stella.
- [4] Olivares, J., Méndez, F. X., & Maciá, D. (1996). Mutismo selectivo. Un modelo explicativo. *Psicología conductual*, 4(2), 169-192.
- [5] Hernández, J. M., Artigas, J., Martos, J., Palacios, S., Fuentes, J., Belinchón, M., & Posada, M. A. (2005). Guía de buena práctica para la detección temprana de los trastornos del espectro autista. *Rev Neurol*, 41(4), 237-45.
- [6] <http://www.psicopedagogia.com/articulos/?articulo=458>
- [7] <https://medlineplus.gov/spanish/cerebralpalsy.html>
- [8] Gironés, J. T. (2012). El gran libro de Android. Marcombo.
- [9] Lafosse, J. (2010). *Struts 2: El framework de desarrollo de aplicaciones Java EE*. Ediciones Eni.
- [10] PÉROCHON, S. H. S. (2014). *Android: Guía de desarrollo de aplicaciones para Smartphone y Tablet* (2a edición). Ediciones ENI.
- [11] Ducrohet, X., Norbye, T., & Chou, K. (2013). Android Studio: An IDE built for Android. Internet: <http://android-developers.blogspot.in/2013/05/android-studio-ide-built-for-android.html>.
- [12] <http://www.ibm.com/developerworks/ssa/webservices/newto/service.html>.
- [13] <https://possibleapp.com/blog/2016/01/que-es-mvc-y-por-que-es-tan-usado-en-el-desarrollo-web/>
- [14] <http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>
- [15] <http://scout.ugr.es/scout/>
- [16] <http://www.fundacionorange.es/emintza.html>
- [17] <http://www.mayer-johnson.com/boardmaker-software>
- [18] <https://www.proyectosfundacionorange.es/intic>
- [19] Bolloju, N., & Leung, F. S. (2006). Assisting novice analysts in developing quality conceptual models with UML. *Communications of the ACM*, 49(7), 108-112.
- [20] Davis, C. J., & Hevner, A. R. (2015). *Neurophysiological Analysis of Visual Syntax in Design*. In *Information Systems and Neuroscience* (pp. 99-105). Springer International Publishing.
- [21] Kouhen, A., Gherbi, A., Dumoulin, C., & Khendek, F. (2015). On the Semantic Transparency of Visual Notations: Experiments with UML. In *SDL 2015: Model-Driven Engineering for Smart Cities* (pp. 122-137). Springer International Publishing.
- [22] Figl, K., Mendling, J., & Strembeck, M. (2013). The influence of notational deficiencies on process model comprehension. *Journal of the Association for Information Systems*, 14(6), 312.
- [23] Genon, N., Heymans, P., & Amyot, D. (2010). Analyzing the cognitive effectiveness of the BPMN 2.0 visual notation. In *Software Language Engineering* (pp. 377-396). Springer Berlin Heidelberg.
- [24] Jacques, B.: *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, Madison, Wisconsin (1983).
- [25] Kim, J., Hahn, J., & Hahn, H. (2000). How do we understand a system with (so) many diagrams? Cognitive integration processes in diagrammatic reasoning. *Information Systems Research*, 11(3), 284-303.
- [26] Moody, D., & van Hillegerberg, J. (2008). Evaluating the visual syntax of UML: An analysis of the cognitive effectiveness of the UML family of diagrams. In *Software Language Engineering* (pp. 16-34). Springer Berlin Heidelberg.
- [27] Moody, D. L., Heymans, P., & Matulevicius, R. (2009, August). Improving the effectiveness of visual representations in requirements engineering: An evaluation of i* visual syntax. In *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International* (pp. 171-180).
- [28] Moody, D. L. (2009). The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering. *Software Engineering, IEEE Transactions on*, 35(6), 756-779.
- [29] Guerrero-García, J. (2014). Evolutionary design of user interfaces for workflow information systems. *Science of Computer Programming*, 86, 89-102.
- [30] ISO/IEC: 24744: *Software Engineering - Metamodel for Development Methodologies*. International Organization for Standardization/International Electrotechnical Commission, Geneva, 2007.
- [31] Guerrero-García, J., González-Calleros, J., Muñoz-Arteaga, J., Morales, A., & Monarca, I. (2016, April). Getting Research Findings into Practice: Guidelines to Produce Quality Software Engineering Diagrams



- to Assist Novice Engineers. In 2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT) (pp. 149-158). IEEE.
- [32] Pruitt, J., Adlin, T., The Persona Lifecycle: Keeping People in Mind Throughout Product Design. Morgan Kaufmann, 2006.
- [33] Aliño, J. J. L. I., & Miyar, M. V. (2008). DSM-IV-TR: Manual diagnóstico y estadístico de los trastornos mentales. American Psychiatric Pub.
- [34] Wibas (2013). Scrum Compact. Version 5.0. ISBN 978-3-981-5837-1-7.
- [35] Pichler R. (2016). Strategize: Product Strategy and Product Roadmap Practices for the Digital Age. Pichler Consulting, 1 edition.
- [36] Sutherland, J. (2015). Scrum: El arte de hacer el doble de trabajo en la mitad de tiempo. Grupo Planeta (GBS).
- [37] Guerrero García, Josefina, Jean Vanderdonckt, and Christophe Lemaigre. Identification criteria in task modeling. Human-Computer Interaction Symposium. Springer US, 2008.
- [38] Pressman, Roger S. (2009). Software Engineering, A practitioner's approach; 7th Edition. Estados Unidos: McGraw Hill.
- [39] <https://www.ciudadano2cero.com/que-es-google-analytics/>
- [40] <http://www.inegi.org.mx>