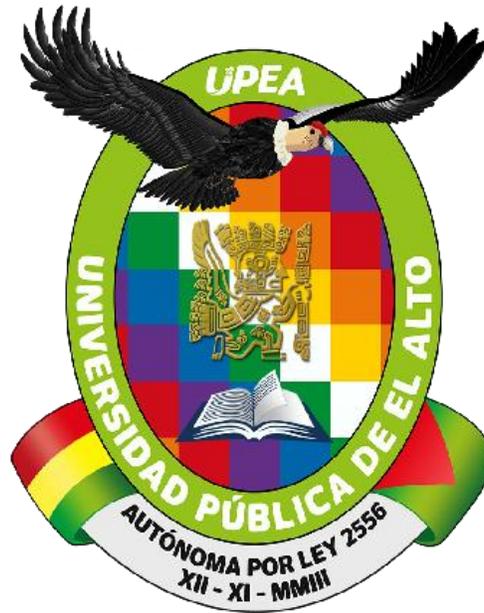


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



TESIS DE GRADO

“GUANTE CON SENSORES Y APLICACIÓN MÓVIL PARA LA INTERPRETACIÓN BIDIRECCIONAL DEL ALFABETO DACTILOLÓGICO”

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante : Univ. Alfredo Huanca Chui
Tutor Metodológico : Lic. Ing. Dionicio Henry Pacheco Ríos
Tutor Revisor : M. Sc. Lic Elías Ali Alvarez
Tutor Especialista : Ing. Rolando Alarcón Choquehuanca

EL ALTO - BOLIVIA

2023

DEDICATORIA

Dedico esta tesis de grado a: A Dios, por permitirme llegar a uno más, de los tantos momentos de dicha y felicidad en mi vida.

A mi mama, María por darme todo su amor, apoyo, motivación, paciencia, confianza y además depositar su fe, en mí.

A mis Hermanos Cosme y Julia por brindarme siempre su ayuda y cariño en estos años.

AGRADECIMIENTOS

Agradecer a Dios por haberme guiado y protegido durante estos años. Agradecer a todos los integrantes de mi Familia por el apoyo incondicional que me dieron en estos años. Agradecer a la Universidad pública de el alto y a la carrera de ingeniería de sistemas por brindarme sus predios y así desarrollarme profesionalmente.

También agradecer a mis distinguidos tutores:

A mi tutor metodológico Ing. Dionicio Henry Pacheco Ríos, por el conocimiento, apoyo, paciencia y motivación brindada a mi persona para culminar esta meta.

A mi tutor especialista Ing. Rolando Alarcón Choquehuanca, por el apoyo y el conocimiento que compartió hacia mi persona durante el desarrollo de la presente tesis.

A mi tutor revisor M.Sc. Elías Ali Alvarez, por el apoyo y orientación brindadas durante la culminación de la presente tesis.

RESUMEN

En este trabajo se presenta la elaboración de un guante traductor del lenguaje de señas y una aplicación que facilitara la comunicación de las personas sordomudas con las personas que se encuentran en su entorno.

La función del guante es traducir el lenguaje de señas por medio de una aplicación y el uso de varios sensores de flexión conectados a los dedos, mediante el empleo de un guante, dichos sensores transformarán la flexión en señales eléctricas para su posterior procesamiento y control mediante una tarjeta Arduino Uno, para reproducir en sonidos el contenido del mensaje que se enviará a la aplicación se usará un celular Android. Mediante la aplicación móvil el transmisor podrá mandar un mensaje en forma de texto a el guante para que se pueda visualizar en el LCD.

ÍNDICE

1. MARCO PRELIMINAR	1
1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES	1
1.2.1 Antecedentes Internacionales	1
1.2.2 Antecedentes Nacionales	2
1.3 PLANTEAMIENTO DEL PROBLEMA.....	3
1.3.1 Problema Principal	3
1.3.2 Problemas Secundarios	3
1.3.3 Formulación del problema	3
1.4. OBJETIVOS	3
1.4.1 Objetivo General	3
1.4.2 Objetivos Específicos	3
1.5. HIPÓTESIS	4
1.6 JUSTIFICACIÓN	5
1.6.1 Justificación técnica.....	5
1.6.2 Justificación económica.....	5
1.6.3 Justificación social.....	5
1.7 METODOLOGÍA.....	6
1.7.1 Método científico	6
1.7.2 Metodología de desarrollo	6
1.8 HERRAMIENTAS	6
1.9 LÍMITES Y ALCANCES.....	8

1.9.1 Límites.....	8
1.9.2 Alcances.....	8
1.10 APORTES	9
2. MARCO TEÓRICO.....	11
2.1 INTRODUCCIÓN	11
2.2 APLICACIÓN MÓVIL.....	11
2.3 GUANTE SENSORIZADO.....	11
2.4 APRENDIZAJE.....	12
2.5 LENGUAJE DE SEÑAS	12
2.6 METODOLOGÍA MOBILE-D.....	12
2.7 METODOLOGÍA DE DISEÑO	14
2.7.1 Etapa de Adquisición de señales.....	15
2.7.2 Procesamiento de Datos	16
2.7.3 Despliegue de Datos	16
2.8 DACTILOLOGÍA O ALFABETO GESTUAL	16
2.9 DISCAPACIDAD AUDITIVA	17
2.10 CAUSAS Y CLASIFICACIÓN DE LA PÉRDIDA AUDITIVA.....	17
2.11 SENSORES	18
2.12 SENSOR DE SEÑALES	18
2.13 TRANSDUCTOR DE SEÑALES.....	18
2.14 CLASIFICACIÓN DE LOS SENSORES	18
2.14.1 Según el Tipo de Señal de Entrada	18
2.14.2 Según el Tipo de Señal Entregada por el Sensor	19
2.15 TECNOLOGÍAS DE COMUNICACIÓN INALÁMBRICA.....	20

2.15.1 Modulo Bluetooth.....	20
2.16 ARDUINO.....	20
2.16.1 Lenguaje de desarrollo y entorno de programación	21
2.16.2 Ide de Arduino	22
2.16.3 Módulos Arduino.....	24
2.17 ELEMENTOS ELECTRÓNICOS	26
2.17.1 Resistencias	27
2.17.2 Diodo led	28
2.17.3 Sensor flexible resistivo	29
2.17.4 Cable conductor	31
2.18 APLICACIONES MÓVILES	31
2.19 APP INVENTOR.....	32
2.19.1 Funcionamiento	32
2.19.2 Diseñador	33
2.19.3 Editor de Bloques	34
2.19.4 Emulador	34
3. DISEÑO METODOLÓGICO	41
3.1 TIPO DE INVESTIGACIÓN	41
3.1.1 Enfoque.....	41
3.1.2 Nivel de investigación.....	41
3.2 DISEÑO DE INVESTIGACIÓN.....	41
3.3 VARIABLES DE LA INVESTIGACIÓN	41
3.4 AMBIENTE DE LA INVESTIGACIÓN	41
3.4.1 Población y muestra	42

3.5 DESCRIPCIÓN DE LA METODOLOGÍA	42
3.5.1 Desarrollo del sistema en base a la metodología Mobile-D	42
3.6 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS.....	57
3.7 COSTO DEL PROTOTIPO.....	58
3.7.1 Costo De Desarrollo De Software.....	58
4. PRUEBAS Y RESULTADOS.....	61
4.1 DISEÑO Y DESARROLLO DE PRODUCTO	61
4.1.1 Diseño de producto	61
4.1.2 Fase de producción	63
4.2 DEMOSTRACIÓN DEL PROTOTIPO	69
4.2.1 Pruebas del sistema electrónico	69
4.2.2 Rangos de entradas analógicas	70
4.2.3 Pruebas de funcionamiento	70
4.2.4 Prueba de aprendizaje	72
4.3 PRUEBA DE FUNCIONAMIENTO DE SOFTWARE	73
4.4 PRUEBA DE HIPÓTESIS.....	74
4.4.1 Estado de la Hipótesis.....	74
5. CONCLUSIONES Y RECOMENDACIONES.....	79
5.1 CONCLUSIONES.....	79
5.2 RECOMENDACIONES	79
BIBLIOGRAFÍA	80
ANEXOS	81

CAPÍTULO I

1. MARCO PRELIMINAR

1.1 INTRODUCCIÓN

Con el paso del tiempo la tecnología forma parte de la vida diaria, en gran parte de los casos facilita el desempeño de ciertas actividades realizadas por la población. En este trabajo se presenta la elaboración de un guante traductor y una aplicación que facilitara la comunicación de las personas sordomudas con las personas que se encuentran en su entorno.

La función del guante es traducir el lenguaje de señas por medio de una aplicación y el uso de varios sensores de flexión conectados a los dedos, mediante el empleo de un guante, dichos sensores transformarán la flexión en señales eléctricas para su posterior procesamiento y control mediante una tarjeta Arduino Uno, para reproducir en sonidos el contenido del mensaje que se enviará a la aplicación se usará un celular Android. Mediante la aplicación móvil el transmisor podrá mandar un mensaje en forma de texto a el guante para que se pueda visualizar en el LCD.

El guante tendrá la función de detectar las señas del abecedario para formar una palabra letra a letra, así también algunas palabras básicas.

1.2 ANTECEDENTES

1.2.1 Antecedentes Internacionales

- Según Chuan y Guardino, (2016) con el tema “Designing SmartSignPlay”, describe el diseño de una app interactiva llamada SmartSignPlay que permite el uso de la interactividad y la popularidad de los teléfonos inteligentes como una gran ventaja para el aprendizaje de los niños sordos y sus familias del vocabulario y las frases de lenguaje sordomudo, la misma que empleo la metodología para la calibración a través de la aplicación de regresión lineal a una población.
- Según Jones Hamilton y Petmecky (2015) con el tema “Mobile Phone Access to a Sign Language Dictionary”, se desarrolla como modelo una aplicación móvil que ayuda a los niños sordos y a sus padres en la búsqueda de anuncios definidos en la lengua de señas estadounidense a través de la cámara del smartphone el Proyecto es de Investigación Aplicada.

- Según Rocha Francisco y Morgado (2015). Con el tema “Aplicación Android para diccionario, voz–texto, texto-voz”, enfocada en el lenguaje de señas para atender a personas con capacidades especiales auditivas como ayuda en la comunicación con la población oyentes. La aplicación, se enfoca en la conversión de texto a voz y voz a texto, la misma que fue desarrollada en el lenguaje de programación Java y XML, los cuales son necesarios para diseños de app en Android Studio. La metodología involucrada en la investigación es de tipo bibliográfica. Los resultados están dirigidos al tiempo, como parte importante en el dialogo entre una persona sorda y una oyente, que ayude experimentar el lapso de utilización, al escribir, repetir y aclarar la conversación con la app y cuál es la diferencia de interacción sin ella.

1.2.2 Antecedentes Nacionales

- Según Ciro Alejandro Zeballos Claire (2020), Universidad Católica Boliviana, regional de Tarija. Título del proyecto Dispositivo traduce el lenguaje de señas con inteligencia artificial la metodología empleada es la descriptiva. Este invento usa un dispositivo electrónico que permite capturar la motricidad fina del ser humano, que define los movimientos que realiza con precisión y exactitud, y los impulsos eléctricos producidos por los músculos durante la elaboración del movimiento, que luego son enviados a un parlante como si fuera la voz de quien lo use.
- Reynaldo Antonio Aquino (2018) tesis de grado reconocimiento e interpretación del alfabeto dactilológico de la lengua de señas mediante tecnología móvil y redes neuronales Universidad Mayor de San Andrés. se realizó un sistema basado en la captura de imágenes de una señal del alfabeto dactilológico a través de la cámara de un dispositivo móvil con sistema operativo Android. Las imágenes son procesadas por el sistema el cual realizará el reconocimiento de la señal mediante redes neuronales artificiales. El procesamiento de las imágenes se realiza utilizando los recursos físicos del dispositivo móvil la metodología usada en el proyecto es la exploratoria.

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 Problema Principal

Las personas sordomudas, tienen problemas para entablar una comunicación vía oral con otros individuos y por lo tanto se dificulta la forma en que se relacionan con su entorno obstaculizando su inclusión educativa, social y laboral.

1.3.2 Problemas Secundarios

- Dificultad en el aprendizaje del Lengua de Señas en niños
- Falta de aplicaciones orientadas a las personas sordomudas
- Aplicaciones orientadas a sordomudos con costos elevados.

1.3.3 Formulación del problema

¿Cómo se puede ayudar a romper la barrera de comunicación que existe en las personas con discapacidad auditiva empleando tecnología de comunicación móvil y un guante sensorizado?

1.4. OBJETIVOS

1.4.1 Objetivo General

Desarrollar un guante con sensores y una aplicación móvil para la interpretación bidireccional del alfabeto dactilológico entre personas sordomudas y personas que desconozcan del lenguaje de señas.

1.4.2 Objetivos Específicos

- Investigar el lenguaje dactilológico para su interpretación dentro del sistema en una comunicación básica de reproducción letra a letra.
- Evaluación y elección de las herramientas para el desarrollo del sistema.
- Implementar un guante con un sistema de sensores que permita la adquisición de los movimientos de los dedos para reproducirlos en una aplicación móvil de manera textual y voz.
- Crear comunicación entre el microcontrolador y el dispositivo móvil por medio de un módulo bluetooth para el envío y recepción de los datos.

- Diseñar una aplicación móvil por la cual se podrá visualizar, reproducir a voz las letras del abecedario o alguna palabra básica.
- Realizar las pruebas del sistema para la comprobación del funcionamiento bidireccional de comunicación y el análisis de los resultados obtenidos.

1.5. HIPÓTESIS

El uso del guante y la aplicación móvil beneficiara la comunicación entre personas sordomudas y personas que desconozcan del lenguaje de señas.

1.5.1 Identificación de variables

Variable independiente: Aplicación móvil y guante sensorizado.

Variable dependiente: Aprendizaje del lenguaje de señas.

1.5.2 Operacionalización de Variables

Tabla 1

Variables independientes y dependientes

Variable	Definición conceptual	Dimensión	Recursos
Aplicación móvil y guante sensorizado	<p>-Aplicación móvil: Es aquel software que utiliza en un dispositivo móvil como herramienta de comunicación, gestión, venta de servicios-productos orientados a proporcionar al usuario las necesidades que demande de forma automática e interactiva. (Lazaro Florido Benitez,2011)</p>	<p>-Software de dispositivo móvil</p> <p>-Lenguaje de programación.</p> <p>- Alfabeto dactilológico</p>	<ul style="list-style-type: none"> - Aplicación Android - Placa Arduino y ordenador - Uso de guante electrónico

Variable independiente	Definición conceptual	Dimensión	Indicador
Aprendizaje del lenguaje de señas	- La lengua de signos es una lengua natural de expresión y configuración gesto espacial y percepción visual (Belen Perez de Arado, 2011)	Lectura y escritura	-reconoce el alfabeto dactilológico . -Imágenes asociadas a palabras. - reconoce las palabras y su significado

Nota: variables dependientes e independientes

1.6 JUSTIFICACIÓN

1.6.1 Justificación técnica

Se elige elaborar una aplicación móvil debido al uso masivo que tienen los dispositivos móviles por su reducido tamaño y portabilidad. Dentro de los sistemas operativos móviles se elige Android por demostrar mayor presencia a nivel nacional.

1.6.2 Justificación económica

La presente tesis de grado se justifica económicamente porque permitirá eliminar el costo que existiría pagar a un profesional traductor de la lengua de señas al lenguaje oral. Además, el presente trabajo se desarrollará bajo plataforma de software libre y tecnologías open source, lo cual permitirá desarrollar el proyecto para que esté disponible de forma gratuita para los usuarios finales, de tal forma que el uso del sistema no debe generar gastos significativos adicionales para el usuario.

1.6.3 Justificación social

La presente tesis de grado es un trabajo de carácter social, ya que permitirá mejorar

la integración y la calidad de vida de las personas con deficiencia auditiva, como también promover y facilitar el aprendizaje de la lengua de señas a personas oyentes.

1.7 METODOLOGÍA

1.7.1 Método científico

El método científico es un conjunto de pasos ordenados que se emplean para adquirir nuevos conocimientos. Para poder ser calificado como científico debe basarse en el empirismo, en la medición y, además, debe estar sujeto a la razón. (Gargantilla, 2020)

Estos son los cinco pasos del método científico:

- Observación: hace referencia a lo que queremos estudiar o comprender.
- Hipótesis: se formula una idea que pueda explicar lo observado.
- Experimentación: se llevan a cabo diferentes experimentos para comprobar o refutar una hipótesis.
- Teoría: permite explicar la hipótesis más probable.
- Conclusiones: se extraen de la teoría formulada.

1.7.2 Metodología de desarrollo

Se emplea en este trabajo una metodología de ingeniería de software ágil para desarrollo de aplicaciones, denominada Mobile-D. Según la metodología de desarrollo de software propuesta, se contempla distintas etapas de un proceso: exploración, inicialización, producción, estabilización y pruebas.

1.8 HERRAMIENTAS

Arduino:

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. Por otro lado Arduino nos proporciona un software consistente en un entorno de desarrollo (IDE) que implementa el lenguaje de programación de arduino y el bootloader ejecutado en la placa. La principal característica del software de programación y del lenguaje de programación es su

sencillez y facilidad de uso. Se puede utilizar para desarrollar y controlar elementos autónomos, conectándose a dispositivos e interactuar tanto con el hardware como con el software.

Módulo Inalámbrico Bluetooth Hc-06

El módulo Bluetooth HC-06 nos permite conectar nuestros proyectos con Arduino a un smartphone, celular o PC de forma inalámbrica (Bluetooth), con la facilidad de operación de un puerto serial. La transmisión se realiza totalmente en forma transparente al programador, por lo que se conecta en forma directa a los pines seriales de nuestro microcontrolador preferido (respetando los niveles de voltaje, ya que el módulo se alimenta con 3.3V). Todos los parámetros del módulo se pueden configurar mediante comandos AT. La placa también incluye un regulador de 3.3V, que permite alimentar el módulo con un voltaje entre 3.6V - 6V.

El módulo Bluetooth HC-06 viene configurado de fábrica para trabajar como Slave.

Especificaciones técnicas

- Voltaje de operación: 3.3V - 5VDC
- Corriente de operación: < 40mA
- Corriente modo sleep: < 1mA
- Chip: BC417143
- Bluetooth: V2.0+EDR
- Frecuencia: Banda ISM de 2,4 GHz
- Modulación: GFSK (Gaussian Frequency Shift Keying)
- Potencia de emisión: 4 dBm, clase 2
- Sensibilidad: -84dBm a 0.1% VER
- Alcance 10 metros
- Interfaz de comunicación: Serial UART TTL
- Velocidad de transmisión: 1200bps hasta 1.3Mbps
- Velocidad asíncrona: 2.1Mbps (máx.) / 160 kbps.
- Velocidad síncrona: 1Mbps/1Mbps

- Seguridad: Autenticación y encriptación
- Compatible con Android
- Dimensiones: 37*16 mm
- Peso: 3.2 gramos

Parámetros de fábrica

- Nombre: HC-06 o Linvor
- Código Pin: 1234

Sensor de Flexión

Este sensor tiene como variable principal la resistencia, la misma que cambia su valor cuando se lo dobla, es ideal para trabajar en ambientes flexibles en donde se requiera medir la deformación de algún objeto.

App Inventor

App Inventor es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web.

1.9 LÍMITES Y ALCANCES

1.9.1 Límites

- El sistema solo realizara el reconocimiento e interpretación del alfabeto dactilológico.
- La aplicación móvil solo estará disponible para dispositivos móviles con sistema operativo Android
- El sistema reconocerá e interpretará una seña a la vez a una distancia adecuada.

1.9.2 Alcances

- Se realizará la traducción al lenguaje oral mediante el altavoz del dispositivo móvil.

- Para garantizar que la comunidad sorda tenga otros medios de comunicación la presente tesis plantea un guante electrónico que, por medio de una aplicación reproduzca a voz las señas que se realizaron con el guante.

1.10 APORTES

- Una herramienta útil para la traducción de la lengua de señas para personas que no conocen dicho lenguaje.
- Una herramienta útil para difundir el alfabeto dactilológico.
- Reducción de costos en la contratación de un traductor profesional.
- Inclusión de personas con deficiencia auditiva en la comunidad.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 INTRODUCCIÓN

En el presente capítulo se desarrollará definiciones, teoría, conceptos, metodologías y herramientas que se utilizarán para la implementación y elección de sensores que tendrá el guante.

Finalmente, en base a la teoría y el objetivo que se pretende conseguir con esta tesis utilizando herramientas de Software y Hardware como ser App Inventor, lenguaje de programación C ++ y los módulos de Arduino nos ayudara a conseguir los resultados que deseamos alcanzar.

2.2 APLICACIÓN MÓVIL

En la actualidad, el crecimiento de dispositivos inteligentes en el mundo ha hecho que el desarrollo de aplicaciones móviles como juegos, entretenimiento, localizaciones, noticias, libros, datos del tiempo, deportes, entre otros; aumente en gran medida, con la finalidad de cubrir las necesidades de los usuarios.

Una aplicación móvil, o app (en inglés) es una aplicación informática diseñada para ser ejecutada en dispositivos móviles. (Delgado, 2009, párr. 1).

2.3 GUANTE SENSORIZADO

Es un instrumento para el reconocimiento de gestos realizados con la mano. Existen distintos tipos de guantes electrónicos. Los más comunes son los que tienen sensores de presión mediante resistencia eléctrica a lo largo del guante. Por lo general estos guantes se programan para juegos donde cada dedo controla algún aspecto del juego. Otros poseen sensores de flexión en cada dedo y la palma de la mano que permite conocer el gesto que realiza la mano en base a una diferencia de resistencia eléctrica generada por flexión del sensor. Los guantes más avanzados y precisos utilizan giroscopios y acelerómetros para conocer la posición y el movimiento que se realiza cada dedo y la mano en general, siendo este tipo de guantes utilizados para investigaciones médicas en el uso de terapias y telemedicina (Valverde, 2010, párr.2).

2.4 APRENDIZAJE

Se denomina aprendizaje al proceso de adquisición de conocimientos, habilidades, valores y actitudes, posibilitado mediante el estudio, la enseñanza o la experiencia. Dicho proceso puede ser entendido a partir de diversas posturas, lo que implica que existen diferentes teorías vinculadas al hecho de aprender. La psicología conductista, por ejemplo, describe el aprendizaje de acuerdo a los cambios que pueden observarse en la conducta de un sujeto (Porto, 2008).

2.5 LENGUAJE DE SEÑAS

Es una lengua natural que se basa en expresiones gesto-espaciales, movimientos corporales y de percepción visual. Tiene un código y estructura propia, conformada por seis parámetros: configuración, localización, movimiento, orientación, dirección y expresión facial. Mediante ellos, una persona se puede comunicar fácilmente con otra (Geovanni y Diego, 2017).

2.6 METODOLOGÍA MOBILE-D

La metodología Mobile-D tiene 5 fases de las cuales lo detallan de la siguiente manera: El objetivo es conseguir ciclos de desarrollos muy rápidos en equipos muy pequeños (de no más de diez desarrolladores) trabajando en un mismo espacio físico. Según este método, trabajando de esa manera se deben conseguir productos totalmente funcionales en menos de diez semanas.

Figura 2.6

Ciclo de desarrollo de Mobile-D



Nota: Etapas de la metodología para el desarrollo móvil (Ramírez Vique, 2015)

Cada fase (excepto la inicial) tiene siempre un día de planificación y otro de entrega.

Las fases son:

Exploración. Se dedica a la planificación y a los conceptos básicos del proyecto. Es diferente del resto de fases.

Inicialización. Se preparan e identifican todos los recursos necesarios.

Se establece el entorno técnico.

Fase de producto. Se repiten iterativamente las sub fases, con un día de planificación, uno de trabajo y uno de entrega. Aquí se intentan utilizar técnicas como la del test driven development para conseguir la mayor calidad.

Fase de estabilización. Se llevan a cabo las acciones de integración para asegurar que el sistema completo funciona correctamente.

Fase de pruebas y reparación. Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema según los requisitos del cliente. (Ramirez Vique, 2015, par.43)

Otro de los autores menciona que el ciclo de vida de la metodología Móvil-D se divide en cinco fases: Exploración, inicialización, producción, estabilización y La fase de exploración, siendo ligeramente diferente del resto del proceso de producción, se dedica al establecimiento de un plan de proyecto y los conceptos básicos. Por lo tanto, se puede separar del ciclo principal de desarrollo (aunque no debería obviarse). Los autores de la metodología ponen además especial atención a la participación de los clientes en esta fase.

Fase de inicialización, los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico (incluyendo el entrenamiento del equipo de desarrollo). Los autores de Mobile-D afirman que su contribución al desarrollo ágil se centra fundamentalmente en esta fase, en la investigación de la línea arquitectónica. Esta acción se lleva a cabo durante el día de planificación. Los desarrolladores analizan el conocimiento y los patrones arquitectónicos utilizados en la empresa (extraídos de proyectos anteriores) y los relacionan con el proyecto actual. Se agregan las observaciones, se identifican similitudes y se extraen soluciones viables para su aplicación en el proyecto.

Finalmente, la metodología también contempla algunas funcionalidades nucleares que se desarrollan en esta fase, durante el día de trabajo.

En la fase de producción, se repite la programación de tres días (planificación trabajo-liberación) se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano (de ahí el nombre de esta técnica de TestDriven Development, TDD). Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.

Fase de estabilización, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyectos multi-equipo con diferentes subsistemas desarrollados por equipos distintos.

Fase de prueba, la última fase (prueba y reparación del sistema) tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados. (Blanco P, 2009)

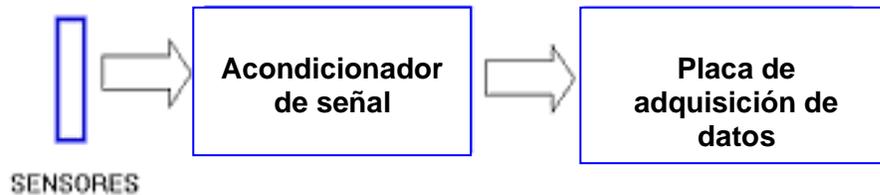
Al conocer las fases de la metodología Mobile- D nos indica que es primordial primero conocer el trabajo que se realizara como las temáticas y que se necesita para poder aprender una segunda lengua, en la inicialización se reconoce todos los recursos que se necesitara para luego producir la aplicación de RA y finalmente revisando el trabajo y realizando las respectivas pruebas para que cumpla los objetivos planteados.

2.7 METODOLOGÍA DE DISEÑO

La metodología se divide en tres etapas, tal como se muestra en la figura 2.7. Estas etapas son: adquisición de las señales enviadas por el guante, procesamiento de señales y finalmente el despliegue del mensaje.

Figura: 2.7

Diagrama en bloques de la adquisición de datos.



Nota : Esquema del sistema de adquisición de datos (Roger Aroa, 2020)

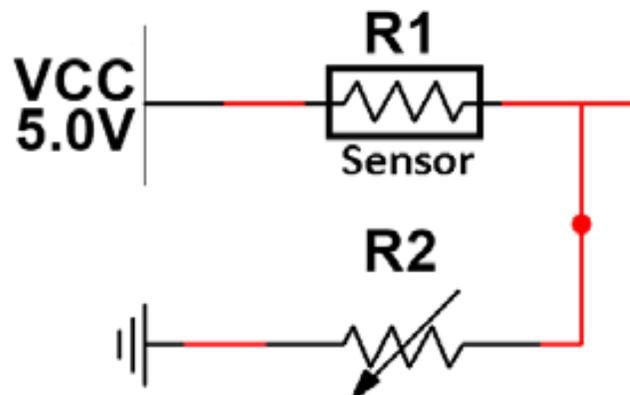
En la parte de la adquisición, se utilizaron 5 sensores de flexión resistivos, los cuales se montaron en la parte posterior de los dedos de la mano sobre un guante, estos envían una señal eléctrica al microcontrolador y este último se encarga de traducir la señal eléctrica a una señal binaria, la cual lo realiza mediante su convertidor A/D de 10 bits.

2.7.1 Etapa de Adquisición de señales

Para el acondicionamiento de señales eléctricas que entrega el guante, se utilizó un divisor de voltaje, utilizando como R1 al sensor de flexión y R2 un potenciómetro para la calibración del divisor de voltaje el diagrama electrónico para esta etapa puede observarse en la figura 2.7.1.

Figura 2.7.1

Diagrama electrónico de la etapa de adquisición.



Nota : Circuito divisor de voltaje (Roger Aroa, 2020)

2.7.2 Procesamiento de Datos

Para el procesamiento de datos se utilizó un Arduino Uno, el cual es adecuado para esta aplicación debido a sus características principales. En el microcontrolador de Arduino se programará el código para el envío y visualización de datos.

Se adquiere la señal eléctrica proveniente de la etapa de adquisición, dicha señal se obtiene y se convierte a un dato binario con la ayuda del convertidor A/D de 10 bits integrado en el microcontrolador. La señal digitalizada proporciona datos binarios los cuales se comparan con los datos almacenados en la memoria del microcontrolador. Cuando una serie de datos provenientes de cada uno de los cinco dedos concuerda con los datos relacionados previamente a un mensaje en específico se muestra el mensaje de texto en la aplicación Android. Los niveles de voltaje previamente calibrados y almacenados que resultan de cada movimiento de los dedos se compararon con los valores correspondientes a cada una de las letras. El sistema se calibró de tal manera que existiera la menor cantidad de errores posibles, los cuales se lograron utilizando sensores de alto desempeño y de bajo costo.

2.7.3 Despliegue de Datos

El resultado de la comparación entre los niveles de voltaje obtenidos con el guante y los almacenados en el microcontrolador hacen corresponder los movimientos de la mano con una frase que será desplegada en la aplicación.

2.8 DACTILOLOGÍA O ALFABETO GESTUAL

El Grupo de señas para representar las letras del alfabeto con el que se escribe la lengua oral del país. Las personas instruidas que tienen discapacidad auditiva y de lenguaje lo emplean para lograr una comunicación efectiva (ONSIPD, 2011).

Tiene 30 configuraciones manuales para simbolizar las letras, las cuales se muestran en la Figura 2.8.

Figura 2.8

Alfabeto Dactilológico Universal



Nota: Deletreo del lenguaje de señas (Estrada, 2016)

2.9 DISCAPACIDAD AUDITIVA

La discapacidad auditiva se refiere a la falta o disminución en la capacidad para oír debido a un problema en algún lugar del aparato auditivo (INP, 2006).

2.10 CAUSAS Y CLASIFICACIÓN DE LA PÉRDIDA AUDITIVA

Las causas que originan la discapacidad auditiva pueden ser:

Genéticas: Se pueden transmitir de padres a hijos. En el caso del área auditiva, este aspecto es el más frecuente y poco previsible.

Congénitas: se refiere a las características o rasgos con los que nace un individuo y que no depende sólo de factores hereditarios, sino que son adquiridos durante la gestación, por ejemplo, las infecciones virales del embrión, destacando la rubéola materna. Destacan también la ingesta de medicamentos durante el embarazo y la asfixia al nacer.

Adquiridas: Son las ocasionadas por algún accidente o enfermedad después del nacimiento. Destaca la meningitis meningocócica, la otitis crónica, la hiperbilirrubinemia y las enfermedades de tipo infeccioso. Otra causa adquirida es la que afecta especialmente a los trabajadores expuestos a ruidos de tipo industrial y la

pérdida auditiva que se relaciona con el envejecimiento.

2.11 SENSORES

Los sensores son herramientas que detectan y responden a algún tipo de información del entorno físico.

Existe una amplia gama de sensores utilizados en la vida diaria, que se clasifican según las cantidades y características que detectan.

Algunos ejemplos incluyen sensores de corriente eléctrica, magnéticos o de radio, sensores de humedad, sensores de velocidad o flujo de fluidos, sensores de presión, sensores térmicos o de temperatura, sensores ópticos, sensores de posición, sensores ambientales y sensores químicos (Pérez Porto, 2010).

2.12 SENSOR DE SEÑALES

Son dispositivos que funcionan como captadores o detectores de información, estos se encuentran en contacto directo con la magnitud que se va a evaluar. En general, los sensores reciben y transforman una magnitud física o química en una señal eléctrica de baja potencia o en una señal óptica, para luego ser entregada a un transductor.

2.13 TRANSDUCTOR DE SEÑALES

Es un elemento o dispositivo que tiene la misión de adaptar un tipo de energía en otro más adecuado para el sistema, es decir “convierte las variaciones de una magnitud física en variaciones de una magnitud eléctrica” (Valverde, 2010, parr 1).

2.14 CLASIFICACIÓN DE LOS SENSORES

A pesar de que pueden hallarse un sin número de clasificaciones para los sensores, a continuación, se tomará como una guía, dependiendo de su tipo de señal ya sea de entrada o salida, así como también de su señal eléctrica generada:

2.14.1 Según el Tipo de Señal de Entrada

Los sensores pueden ser clasificados dependiendo del tipo de señal al cual responden como pueden ser:

2.14.1.1 Mecánica y Térmica

“Permite la medición de magnitudes como: longitud, área, volumen, masa, flujo, fuerza, torque, presión, velocidad, aceleración, posición, acústica, longitud de onda e intensidad acústica. Mientras que la térmica permite la medición de temperatura, flujo de calor y conductividad” (Juan González, 2010, párr. 2).

2.14.1.2 Eléctrica y Magnética

Reconoce la medición de variables como el “voltaje, corriente, carga, resistencia, inductancia, capacitancia, constante dieléctrica, polarización, campo eléctrico y frecuencia. Mientras que la magnética registra la medición de intensidad de campo, densidad de flujo, momento magnético y permeabilidad magnética” (Juan González, 2010, párr. 3).

2.14.1.3 Química y Radiación

“Permite la medición de una composición, concentración, oxidación, potencial de reducción, porcentaje de reacción y PH. Mientras que la radiación reconoce mediciones tanto de una intensidad, longitud de onda, polarización, fase, reflectancia, transmitancia e índice de reflectancia” (Juan González, 2010, párr. 4).

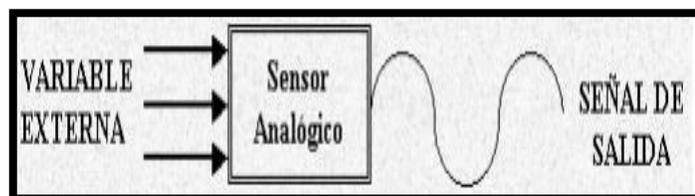
2.14.2 Según el Tipo de Señal Entregada por el Sensor

2.14.2.1 Sensores Analógicos

“La gran mayoría de sensores entregan su señal de manera continua en el tiempo. Son ejemplo de ellos los sensores generadores de señal y los sensores de parámetros variables, fotoceldas o fotoresistores” (Juan González, 2010, párr. 8).

Figura 2.14

Diagrama en bloques de un sensor analógico



Nota: Entrada y salida analógica en un sensor (Juan González, 2010)

2.14.2.2 Sensores Digitales

Son aquellos que frente a un estímulo pueden cambiar de estado ya sea de cero a uno o de uno a cero, en este caso no existen estados intermedios y los valores de tensión que se obtienen son únicamente dos, 5 voltios y 0 voltios; o valores muy próximos. Son ejemplos de este tipo de sensores: switch e interruptores (Juan González, 2010, párr. 9).

2.15 TECNOLOGÍAS DE COMUNICACIÓN INALÁMBRICA

Las comunicaciones inalámbricas se refieren a establecer un tipo de comunicación sin cables, es decir, que se necesitan el uso de las frecuencias de radio u ondas infrarrojas para la transferencia de información desde un punto a otro. Las principales ventajas de utilizar estas tecnologías inalámbricas es que permiten generar una amplia libertad de movimientos por parte del usuario, también facilita la reubicación de las estaciones de trabajo, evitando la necesidad de establecer cableado y la rapidez en la instalación; otra de las ventajas es la reducción de costos al momento de la implementación. Existen varias tecnologías de transmisión inalámbrica haciendo cada una de ellas adecuadas a determinados usos o aplicaciones como pueden ser: Bluetooth, Wi-Fi, Infrarrojos, Zigbee, GPS; entre otros.

2.15.1 Modulo Bluetooth

El término Bluetooth describe una tecnología de red desarrollada por el grupo de trabajo IEEE 802.15.1 del Institute of Electrical and Electronics Engineers estadounidense como estándar industrial para conexiones inalámbricas. La tecnología Bluetooth sirve para la transferencia de voz y datos punto a punto sin conexión u orientada a la conexión entre dos dispositivos digitales diferentes. El objetivo principal de esta tecnología es reemplazar las conexiones por cable, es decir, dejarlas obsoletas, lo cual supone una ventaja, sobre todo, para dispositivos móviles como smartphones o tabletas. (Yúbal Fernández,2020).

2.16 ARDUINO

“Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar” (Letrán, 2012, párr. 1).

En la actualidad en el mundo del desarrollo de plataformas electrónicas ha hecho que

personas de toda edad se motiven a desarrollar e investigar todo tipo de proyectos o prototipos, que permitan solucionar algún problema planteado; especialmente en el campo de la “educación donde empieza a considerarse de gran importancia dotar a las nuevas generaciones de conocimientos básicos de programación de aplicaciones o de desarrollo electrónico” (Arenas, 2014, párr. 1).

El desarrollo de proyectos interactivos ha permitido que, “Arduino pueda tomar información del entorno a través de sus pines de entrada de toda una gama de interruptores o sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores” (Letrán, 2012, párr. 1).

Además, se necesita un ordenador donde instalar el entorno de desarrollo para escribir los programas que harán que el sistema Arduino funcione de una determinada manera.

2.16.1 Lenguaje de desarrollo y entorno de programación

El microcontrolador en la placa Arduino contiene un lenguaje de programación Arduino basado en Wiring, y el entorno de desarrollo Arduino basado en Processing; es decir que los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software.

2.16.1.1 Wiring

“Es una plataforma abierta de prototipos electrónicos compuestos de un entorno de programación, en donde se tiene documentación creada por expertos, ingenieros, estudiantes, y personas con experiencia”, donde todos ellos comparten sus ideas, conocimiento y experiencias adquiridas conforme al desarrollo e investigación de cualquier tipo de proyecto (Letrán, 2012, párr. 4).

2.16.1.2 Processing

“Es un lenguaje de programación de código abierto, enfocado para las personas que quieran crear imágenes, animaciones e interacciones. Inicialmente desarrollado para enseñar los fundamentos de la programación de computadora en un contexto visual” (Letrán, 2012, párr. 3).

2.16.2 Ide de Arduino

Arduino incluye su propio IDE conocido como el entorno interactivo de desarrollo basado en la aplicación escrita en Java, el cual permite que la programación de la placa electrónica sea sencilla debido a que se basa en el código abierto de Processing y Wiring. Fue desarrollada para ser lo más amigable con el usuario.

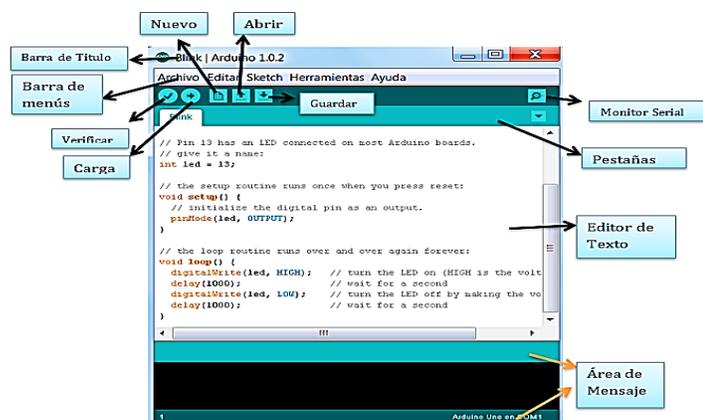
El IDE de Arduino trae una biblioteca escrita en código de programación de alto nivel C/C++ llamada "Wiring" que hacen más fácil escribir la programación. "Una biblioteca es un conjunto de rutinas previamente programadas con las instrucciones habituales de entrada/salida, cuyo funcionamiento interno es transparente para el usuario" (Maocho, 2012, párr.6).

El proceso de descarga del IDE de Arduino se puede realizar desde su página oficial mediante el siguiente link <http://arduino.cc/es/Main/Software> donde se tendrán opciones de descarga dependiendo del tipo de ordenador ya sea Windows, Mac, o Linux. Una vez ejecutado e instalado el archivo Arduino.exe en modo administrador, prácticamente se procede al reconocimiento del modelo de tarjeta Arduino con su puerto serie asignado al controlador Arduino.

Se puede observar el entorno interactivo de desarrollo de Arduino con sus respectivas partes mediante la siguiente imagen.

Figura 2.16

IDE de Arduino



Nota: Principales partes del entorno de Arduino (Abdullahi,2014)

Existen 5 etiquetas en donde se muestra el menú principal y las demás partes fundamentales del IDE de Arduino, los cuales ofrecen funciones específicas para cada uno como:

Archivo.- Muestra opciones de crear, abrir, guardar, cerrar, imprimir, cargar, configurar y salir con respecto a los programas realizados en el editor de textos.

Editar.- Muestra opciones aplicadas para el editor de textos donde se podrá copiar, pegar, seleccionar, comentar, buscar con respecto al código de programación realizado.

Sketch.- Muestra opciones para verificar y complicar el código de programación así como también para añadir las diferentes librerías que utiliza Arduino.

Herramientas.- Muestra las dos opciones más importantes como son la elección del tipo de tarjeta electrónica Arduino que se use, así como también del puerto serial del controlador Arduino al cual esté conectado.

Ayuda.- Muestra opciones de ayuda en donde se tiene como referencia la página web de Arduino con las preguntas más frecuentes, con el fin de ayudar al programador a utilizar de mejor manera el IDE de Arduino.

Botones de Acceso.- Muestra botones para verificar el código para encontrar errores en la programación, también para compilar el código y cargarlo en la placa electrónica Arduino, así como crear, cargar y guardar nuevos archivos.

Editor de Texto.- “Es un procesador de texto especializado para escribir los programas con características específicas como el resaltado de nombres de comandos, sangrías, espacios, entre otros” (Maocho, 2012, párr. 3).

Área de mensajes. - Muestra el resultado de la compilación del código de programación ya sea esta que falle o sea exitosa sin errores.

Líneas de Consola.- Muestra resultados como el tamaño del código de programación compilado, así como también de mensajes de error al momento de realizar la compilación del código de programación.

2.16.3 Módulos Arduino

Existen varios boards o placas electrónicas que presenta Arduino, entre las cuales se pueden destacar las siguientes:

2.16.3.1 Arduino Uno

El Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 pines digitales de entrada/salida de los cuales 6 pueden utilizarse para salidas PWM, 6 entradas analógicas, una conexión USB, un conector de alimentación, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA o la batería a DC para empezar (Maocho, 2012,párr. 1).

Figura 2.16.1

Placa Arduino Uno



Nota: Estructura física de la placa Arduino UNO (Abdullahi,2014)

2.16.3.2 Arduino Mega 2560

Es una placa electrónica basada en el microcontrolador Atmega1280. Posee 54 pines digitales de entrada/salida los cuales 15 pines proporcionan salida PWM, 16 entradas analógicas, 4 puertos seriales, un oscilador de 16MHz, una conexión USB, un conector de alimentación y un botón de reinicio (Abdullahi ,párr. 1).

Figura 2.16.2

Placa electrónica Arduino Mega 2560



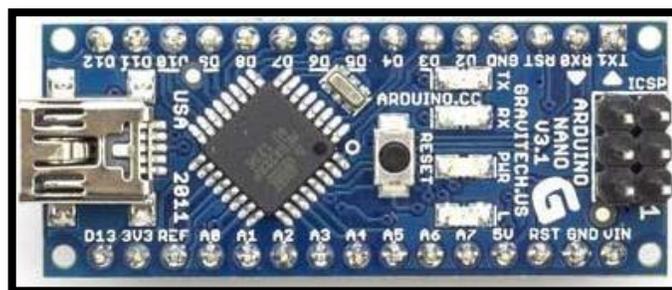
Nota: Estructura física de la placa Arduino Mega (Abdullahi,2014)

2.16.3.3 Arduino Nano

“Su principal ventaja es que puede ser conectado directamente al protoboard, sus dimensiones van desde los 1.9cm de ancho hasta los 4.5cm de largo”, es muy práctico y muy utilizable, pero con la desventaja de que posee menos memoria de almacenamiento con respecto a las demás placas Arduino (Martínez, 2012, parr. 4).

Figura 2.16.3

Placa electrónica Arduino Nano



Nota: Estructura física de la placa Arduino Nano (Abdullahi,2014)

2.16.3.4 Arduino Leonardo

“Difiere de todas las placas anteriores en que el ATmega32u4 ha incorporado en la comunicación USB, eliminando la necesidad de un procesador secundario. Esto permite que el Leonardo aparezca a un ordenador conectado como un ratón y el teclado” (Arduino, 2014, párr. 1)

Figura 2.16.4

Placa electrónica Arduino Leonardo



Nota: Estructura física de la placa Arduino LEONARDO (Abdullahi,2014)

2.16.3.5 Arduino Ethernet

Se diferencia de otras placas en que no tiene un chip integrado controlador de USB a serie, pero tiene una interfaz Wiznet Ethernet. “Es la misma interfaz que se encuentra en el escudo Ethernet. Un lector de tarjetas microSD a bordo, que se puede utilizar para almacenar archivos para servir a través de la red, es accesible a través de la Biblioteca SD” (Arduino, 2014, párr. 1).

Figura 2.16.5

Placa electrónica Arduino Ethernet



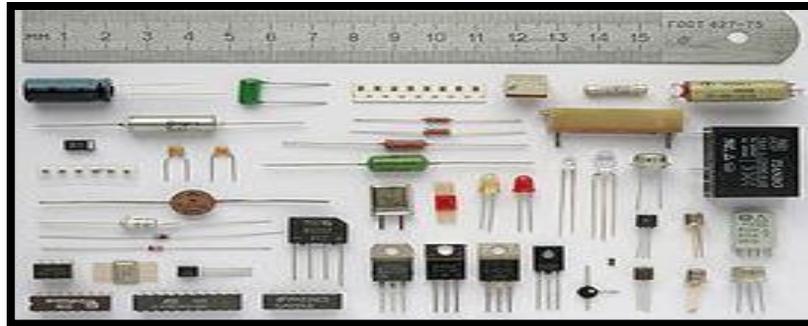
Nota: Estructura física de la placa Arduino Ethernet (Abdullahi,2014)

2.17 ELEMENTOS ELECTRÓNICOS

Son dispositivos que forma parte de un circuito electrónico. Se suele encapsular, generalmente en un material cerámico, metálico o plástico, y terminar en dos o más terminales o patillas metálicas. Se diseñan para ser conectados entre ellos, normalmente mediante soldadura, a un circuito impreso, para formar el mencionado circuito (Andrea Aristizabal ,párr. 1).

Figura 2.17

Componentes electrónicos



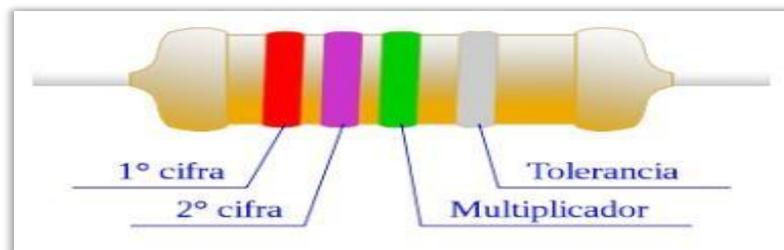
Nota: Componentes electrónicos más comunes en circuitos (Rodríguez,2014)

2.17.1 Resistencias

Componente electrónico que cumple la función de oponerse al paso de corriente eléctrica, permiten controlar el flujo de corriente y tensión dentro de un circuito, esta magnitud se mide en Ohmios (Ω); además se componen de una película de carbón enrollada sobre un soporte cilíndrico cerámico (José García (2012), párr. 1).

Figura 2.17.1

Código de colores en resistencias



Nota: Distribución de bandas de una resistencia (Comunidad Electronica,2014)

El valor de una resistencia viene determinado según la tabla del código de colores, en donde la primera y segunda banda son las dos primeras cifras, la tercera banda nos indica por cuanto tenemos que multiplicar las dos primeras cifras o el número de ceros que hay que añadir a los dos primeros valores obtenidos; mientras que la última banda indica la tolerancia que es el porcentaje más o menos que podrá tener esa resistencia.

Tabla 2*Distribución del código de colores para resistencias*

COLOR	PRIMERA BANDA	SEGUNDA BANDA	TERCERA BANDA	TOLERANCIA
Negro	0	0	x 1	
Café	1	1	x 10	2%
Rojo	2	2	x 100	
Naranja	3	3	x 1000	
Amarillo	4	4	x 10000	
Verde	5	5	x 100000	
Azul	6	6	x 1000000	
Violeta	7	7	x 10000000	
Gris	8	8	x 100000000	
Blanco	9	9	x 1000000000	
Dorado				5%
Plata				10

Nota: Bandas de colores y tolerancia para resistencias (Electrónica Joan, 2000)

2.17.2 Diodo led

“Componente electrónico cuya función principal es convertir la energía eléctrica en una fuente luminosa al momento de ser polarizado de forma directa” (García, 2012, párr. 4).

Soportan tensiones menores a los 2 voltios por lo que es necesario colocar una resistencia en serie con la finalidad de que no llegue a quemarse este componente. Posee dos patillas el ánodo (+) identifica por ser la patilla más larga y el cátodo (-) la patilla más corta.

Son útiles para aplicaciones como indicación de estado de circuitos, señalizaciones, indicadores de frecuencias de reloj; entre otros.

Figura: 2.17.2

Diodo Emisor de Luz



Nota: Diodos led de diferentes colores (Electrónica Joan,2014)

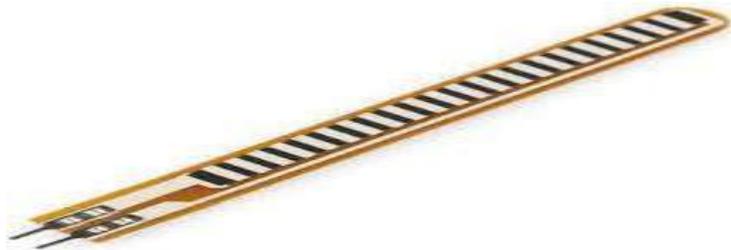
2.17.3 Sensor flexible resistivo

Este sensor es un “elemento electrónico de tipo pasivo, ya que necesita de alguna polarización para actuar como un transductor, es decir convertir una magnitud o tipo de energía en otra” (Espinosa, Pozo, 2013, p. 26).

Se basa en la impresión de una tinta de polímero en una cara del sensor, el cual almacena partículas conductoras, que dependiendo de la posición del sensor mostrara distintos valores resistivos.

Figura: 2.17.3

Sensor Flexible Resistivo

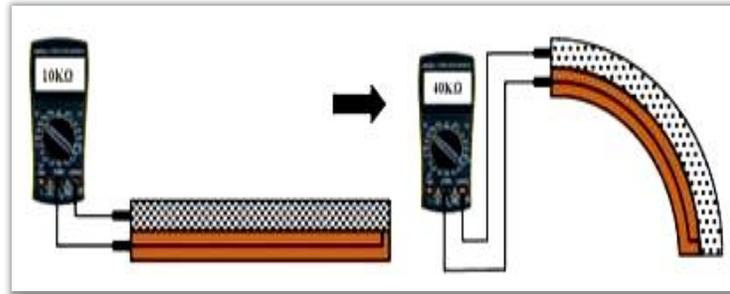


Nota : Sensor flex de resistencia 1M Ohm(Electrónica Joan, 2000)

El funcionamiento de este sensor se basa en un conductor de tipo resistivo llamados hilos de resistencia finos, el cual modifica su valor óhmico cuando es sometido a alguna deformación, desplazamiento o flexión de sus contactos, en la forma de estrechar o alargar el sensor; según se muestra en la figura 2.17.4.

Figura: 2.17.4

Medición del valor óhmico del sensor flexible con el tester



Nota: Escala en Ohmio para medir la resistencia (Espinosa, 2013)

Para medir el valor resistivo del sensor flexible es necesario poner en contacto sus dos terminales de cualquier manera ya que no poseen polaridad sobre los terminales de un potenciómetro; “si el sensor es flexionado su valor óhmico aumentara dependiendo del rango de funcionamiento que posea en sensor flexible” (Espinosa, Pozo, 2013, parr. 27).

Sus características principales de este sensor flexible son:

Dimensiones. - Se pueden encontrar sensores con una longitud de 5.58 centímetros y de 11.43 centímetros, presentan un ancho de 0.64 centímetros y un espesor de 0.020 centímetros.

Rango de Variación. - Si el sensor se encuentra en su “posición lineal normal su valor óhmico será mínimo, pero si es flexionado a 90 grados tiende a su máximo valor óhmico” (Espinosa, Pozo, 2013, parr. 28).

Construcción. - Membrana flexible y resistente, con tinta de polímero con partículas conductoras. En el interior del sensor existen elementos resistivos de carbono.

Fabricantes. - “Existen tres fabricantes de estos sensores flexibles que son, Spectra Symbol Flex Sensors, Gnetile Abrams Sensor y Gizmo Music” (Espinosa, Pozo, 2013, parr. 29).

Tiempo de Vida Útil.- Tiene aproximadamente más de un millón de flexiones, dependiendo en las condiciones expuestas a su funcionamiento.

Rango de Temperatura.- Opera en los rangos de -35°C hasta los $+80^{\circ}\text{C}$.

Voltaje.- Opera en el rango de voltajes de los 5 a 12 voltios.

2.17.4 Cable conductor

Normalmente son “fabricados de un material de cobre, envueltos con un material tipo aislante o protector, el cual permite conducir el flujo de corriente eléctrica desde un punto hacia otro, estos cables son utilizados para las prácticas de laboratorio de electrónica” (Marquina, Gutiérrez, Fonseca, 2012, pag. 2).

Un cable conductor básicamente se compone de las siguientes partes:

Conductor: “Elemento que conduce la corriente eléctrica y puede ser de diversos materiales metálicos como el cobre o aluminio. Puede estar formado por uno o varios hilos” (Marquina, Gutiérrez, Fonseca, 2012, parr. 2).

Aislamiento: Recubrimiento que envuelve al conductor, para evitar la circulación de corriente eléctrica fuera del mismo.

Capa de relleno: Material aislante que envuelve a los conductores para mantener la sección circular del conjunto.

Cubierta: Está hecha de “materiales que protejan mecánicamente al cable. Tiene como función proteger el aislamiento de los conductores de la acción de la temperatura, sol, lluvia, polvo” (Marquina, Gutiérrez, Fonseca, 2012, parr. 2).

2.18 APLICACIONES MÓVILES

“Gracias a los avances tecnológicos de los últimos años y la aparición de sistemas operativos móviles mucho más eficientes y dinámicos, el desarrollo de las aplicaciones móviles ha evolucionado de manera impresionante” (Espinosa, 2012).

En la actualidad se cuentan con dispositivos de última tecnología, los cuales son utilizados por el usuario de tal manera que puedan interactuar con el equipo móvil de una forma sencilla.

El crecimiento del mercado de las aplicaciones móviles es muy alto en cuanto a la aparición de sistemas operativos iOS de Apple, Android y Windows Phone.

“Los desarrolladores de aplicaciones a nivel mundial se encuentran enfocados exclusivamente a la programación de aplicaciones para Smartphone” e introducen una implementación para dispositivos con pantallas más amplias como las Tablet. La función principal de un Smartphone, es de servir como plataforma para aplicaciones

que hagan provecho de las características del teléfono mismo (Espinosa, 2012, párr. 9).

2.19 APP INVENTOR

José Rederjo (2013) señala que “usar los recursos informáticos como instrumentos de resolución de problemas específicos, ha permitido integrar la información textual, numérica y grafica obtenido de cualquier fuente para elaborar contenidos propios y publicarlos en la Internet” (párr. 1).

App Inventor fue desarrollado por Google Labs en agosto del 2011, y su desarrollo fue trasladado al Instituto Tecnológico de Massachusetts (MIT). Fue publicado como software libre bajo la licencia Apache 2.0.

“Es una herramienta de diseño y entorno de desarrollo de aplicaciones para Smartphone y tablets, bajo el sistema operativo Android” (Rederjo, 2013, párr. 2).

Puede desarrollarse mediante un navegador web, un teléfono o emulador para PC, los servidores App Inventor almacenan la información y realizan un seguimiento de los proyectos realizados.

Puede ser instalado en ordenadores con sistemas operativos Windows, GNU/LINUX y para Mac OS X; además para Smartphone o tablets compatibles con varias marcas entre ellas tenemos las más conocidas como Dell, Hp, Lenovo, Huawei, HTC, Kyocera, Motorola, Panasonic, Pioneer, Samsung, TCT Alcatel, Sony, Sony Ericsson, Toshiba; entre otras más.

2.19.1 Funcionamiento

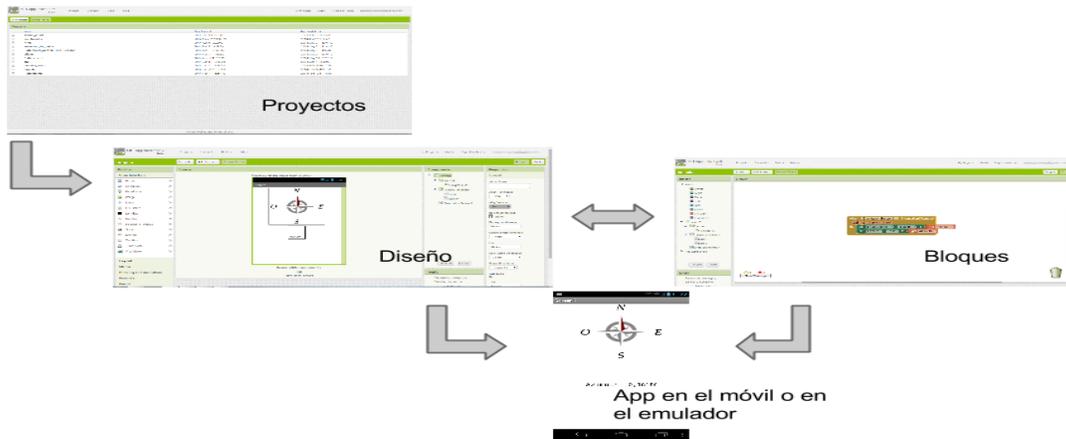
“App Inventor se basa en un lenguaje de programación por bloques” , esto permite de una u otra manera facilitar el desarrollo de aplicaciones móviles, es decir no se necesita escribir ninguna línea de programación (Sánchez, 2014, párr. 2).

“Se colocan bloques para construir bucles, condiciones, variables, eventos, entre otros que permiten pensar lógicamente y solucionar los problemas de forma ordenada” (Rederjo, 2013, párr. 7).

El diagrama de funcionamiento de App Inventor se divide en tres componentes básicos que son la parte del diseñador, el editor de bloques y el emulador o dispositivo Android; los cuales se observan en la siguiente figura 2.19.

Figura: 2.19

Diagrama de Funcionamiento de App Inventor



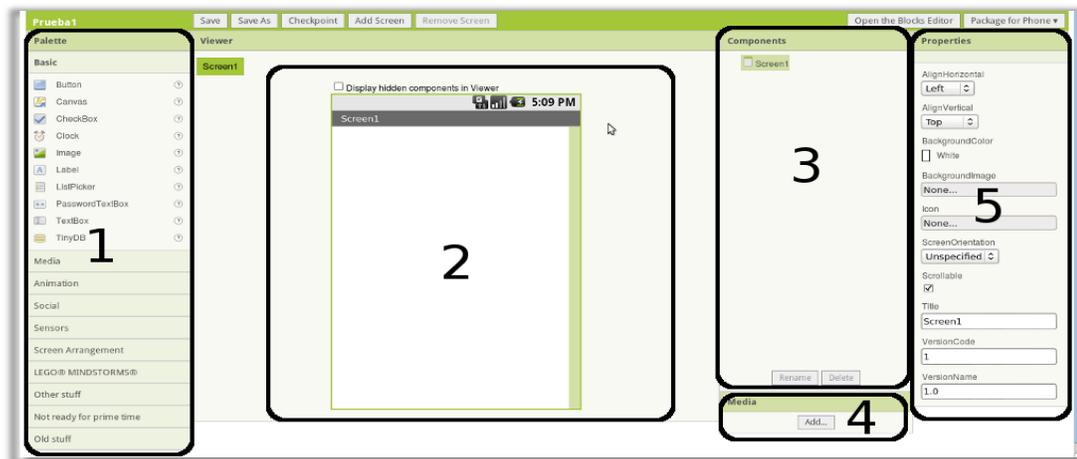
Nota : Diagrama de componentes App Inventor (Sánchez, 2014)

2.19.2 Diseñador

Es la pantalla principal del entorno de desarrollo App Inventor en modo gráfico, aquí el usuario podrá hacer uso de todas las opciones que a continuación se detallan en la figura 2.19.2:

Figura: 2.19.2

Pantalla de Diseño de la aplicación en App Inventor



Nota : Entorno de trabajo en App Inventor (Sánchez, 2014)

Paleta.- Contiene todos los elementos que pueden ser arrastrados o insertados en nuestra aplicación, existen elementos gráficos como cuadros de texto, botones,

dibujos; también elementos no visibles en la pantalla principal como acelerómetro, cámara de video, y bases de datos.

Visor de Pantalla.- Permite simular la apariencia visual que tendrá la aplicación móvil desarrollada por el usuario. Los elementos deben ser insertados desde la paleta hacia el visor de pantalla.

Componentes.- Muestra una lista de todos los elementos colocados desde la paleta hacia el visor de pantalla.

Media.- Muestra las imágenes y sonidos que el usuario cargue y estarán disponibles para usarlas en la aplicación.

Propiedades.- Primero hay que seleccionar los elementos arrastrados al visor de pantalla para lograr observar las propiedades de los mismos, como longitud, ancho, fuente, color, entre otros, dependiendo del tipo de elemento seleccionado.

2.19.3 Editor de Bloques

“Es el entorno en donde se ensamblan los bloques del programa que especifican cómo deben comportarse los componentes elegidos en el diseñador” (Sánchez, 2014, parr. 3).

2.19.4 Emulador

“Es una aplicación que requiere de herramientas de programación Android y son instaladas en un ordenador” (Sánchez, 2014, parr. 3).

También se puede utilizar una conexión a internet por medio del dispositivo móvil y descargar la aplicación MIT Acompañan desde la tienda App Store para poder simular desde un dispositivo móvil la aplicación desarrollada en App Inventor.

2.19.5 Programación en Bloques

El editor de bloques de la aplicación App Inventor utiliza la librería Open Blocks de Java para crear un lenguaje visual a partir de bloques. Estas librerías están distribuidas por el Instituto Tecnológico de Massachusetts (MIT) bajo su licencia libre (MIT License).

El compilador que traduce el lenguaje visual de los bloques para la aplicación en Android utiliza Kawa como lenguaje de programación, distribuido como parte del

sistema operativo GNU de la FSF.

El Editor de bloques tiene dos fichas en la esquina superior izquierda: Built-In y My Blocks. Los botones debajo de cada ficha se amplían y se muestran en bloques cuando se activan mediante un clic. Los bloques Built-In son el conjunto estándar de bloques que están disponibles para cualquier aplicación que se desarrolle. Mientras que, My Blocks contienen bloques específicos que están vinculados al conjunto de componentes que se ha elegido para la aplicación. Antonio Riego (2011).

El Editor de bloques se ejecuta en una ventana independiente, la programación por bloques permite al usuario armar un rompecabezas con piezas que estén en función de la parte del diseño de la aplicación, donde cada pieza cumplirá funciones diferentes.

2.19.5.1 Eventos

Aplicaciones App Inventor están orientadas a eventos”, es decir que toda actividad producida por el usuario como tocar un botón o arrastrar el dedo sobre la pantalla, de un dispositivo móvil produce una respuesta a estos eventos (Riego, 2014, párr. 4).

Los eventos se identifican en la parte de My Blocks del editor de bloques, con su color verde como se muestra en la figura 2.19.5.

Figura: 2.19.5

Bloque de eventos



Nota : Botón 1 de color verde (Riego, 2014)

2.19.5.2 Expresiones y Acciones

Se sitúan dentro de un bloque de eventos, es decir, “son bloques que indican acciones o expresiones como por ejemplo la llamada a una función o un procedimiento” (Villarroel, 2013, párr. 3).

Estos bloques se caracterizan por ser de color azul o morado como se muestra en

la figura 2.19.5.2, según la función a ser asignado y siempre se colocan apilados uno debajo del otro indicando un orden lógico de ejecución de programación en bloques.

Figura: 2.19 .5.2

Bloques de expresiones y funciones



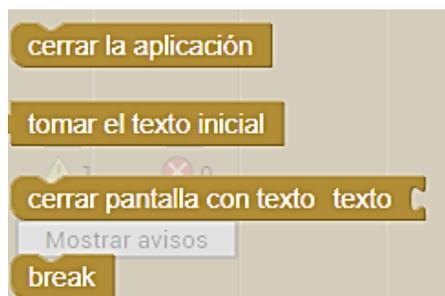
Nota : Llamada de función (Villarroel, 2013)

2.19.5.3 Funciones de Texto

“Permite utilizar funciones para manipular cadenas de texto” , estos bloques son de color marrón como se muestra en la figura 2.19.5.3 y siempre van unidos a los parámetros o bloques indicados dependiendo de la lógica de programación en bloques realizada por el usuario (Villarroel, 2013, párr. 4).

Figura 2.19.5.3

Bloques de funciones de texto



Nota: Bloques para cuadro de texto

2.19.5.4 Funciones de Números

“Permite utilizar bloques en el caso del tratamiento de números y el uso de funciones matemáticas como operadores de comparación” (Villarroel, 2013, párr. 5).

Son bloques de color verde claro como se muestra en la figura 2.19.5.4, y siempre están unidas a los parámetros o bloques. indicados dependiendo de la lógica de programación en bloques realizada por el usuario.

Figura: 2.19.5.4

Bloques numéricos



Nota : Condicionales y comparadores

2.19.5.5 Estructura de Control

“Estos bloques permiten evaluar condiciones de la aplicación para mostrar resultados y tomar acciones correspondientes a la secuencia de bloques desarrollada” (Villarroel, 2013, párr. 5).

Son funciones básicas del sistema como el abrir y cerrar la aplicación. Son de color amarillo y pueden englobar el resto de funciones.

Figura 2.19.5.5

Bloques de control



Nota: Bloques condicionales

2.19.5.6 Conectividad App Inventor

Una de las ventajas que App Inventor ofrece al usuario es el bloque de conectividad que se compone de elementos como:

Ciente Bluetooth

Es un componente que define propiedades tanto del nombre y direcciones MAC de los dispositivos Bluetooth que pueden ser emparejadas dentro de un área o rango de cobertura disponible. Si el dispositivo Bluetooth se encuentra disponible, existen solicitudes de conexión con un módulo externo bluetooth utilizado en servidor de modulo interno del dispositivo móvil, para esto se requiere almacenar la dirección MAC de forma manual en los bloques de programación y es única del módulo bluetooth cliente. La dirección MAC de la aplicación es sincronizada e “implícitamente se deben cumplir dos condiciones para que la conexión se lleve a cabo y con éxito; el modulo bluetooth externo debe estar dentro del rango de alcance y su dirección MAC debe coincidir con la solicitada” (Pérez y Flores, 2013, parr. 4).

Figura: 2.19.5.6

Componente Cliente Bluetooth



Nota: bloque de conexión con bluetooth

Servidor Bluetooth

Pérez y Flores señalaron que “bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN), que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM a 2.4 GHz” (Espinoza, parr. 1).

Figura: 2.19.5.7

Componente Servidor Bluetooth



Al conectar dos dispositivos bluetooth, uno debe actuar como un servidor. El

propósito de la toma de servidor es para escuchar las solicitudes de conexión entrantes por medio de dispositivos bluetooth configurados en modo cliente, cuando una solicitud es aceptada, se proporciona un estado de conexión exitosa. Cuando un cliente bluetooth se ha vinculado a un servidor bluetooth, la conexión puede ser cerrada, a menos que se desee aceptar más conexiones.

Web

Una alternativa para acceder a la información en internet de una manera rápida hoy en día es el desarrollo de aplicaciones personalizadas que usan especialmente botones específicos para acceder a páginas Web deseadas por el usuario.

La conectividad Web permitirá de alguna manera recibir algún servicio como el envío de correos, acceso a basas de datos, entretenimiento, compra-venta de productos, entre otros.

Figura: 2.19.5.8

Componente Web



El componente Web de App Inventor “facilita una aplicación para hablar con un servicio web utilizando el protocolo de transferencia de hipertexto estándar HTTP” (Adam Tothfejel, 2014, párr. 9); es decir que permite mostrar una página Web dentro de una aplicación.

Este componente también da la funcionalidad de enviar y obtener los datos de un servidor o un sitio web a través de GET y POST de solicitudes. Este componente puede decodificar datos JSON y HTML.

Normalmente al usar el componente Web en la programación por bloques de App Inventor “se llama a una URL⁵⁵ que permitirá especificar a qué servicio web se comunicará y se llama a uno de los métodos HTTP para solicitar algún tipo de acción” (Adam Tothfejel, 2014, párr. 10).

CAPÍTULO III

3. DISEÑO METODOLÓGICO

3.1 TIPO DE INVESTIGACIÓN

Se realizó una investigación descriptiva por que se seleccionó una serie de cuestiones y se midió cada una de ellas independientemente para así describirlo.

3.1.1 Enfoque

El enfoque en el que se desarrolla la presente tesis es el cuantitativo, puesto que este es el que mejor se adapta a las características y necesidades de la investigación.

3.1.2 Nivel de investigación

La presente investigación es de tipo causal, según. Los estudios experimentales van más allá de la descripción de conceptos y fenómenos o del establecimiento de relaciones entre conceptos; están dirigidos a encontrar las causas de los eventos, sucesos y fenómenos físicos y sociales (Hernández y Fernández 2007, p: 66).

3.2 DISEÑO DE INVESTIGACIÓN

Investigación experimental.

3.3 VARIABLES DE LA INVESTIGACIÓN

Variable independiente: Aplicación móvil y guante sensorizado.

Variable dependiente: Aprendizaje del lenguaje de señas.

3.4 AMBIENTE DE LA INVESTIGACIÓN

En esta tesis el guante será usado por un grupo de sordomudos (entre 6 y 8 años) a los cuales se les presentará una aplicación móvil para apoyar en su aprendizaje del lenguaje de señas y afirmar o negar la hipótesis planteada anteriormente. En este caso la variable a manipular será el aprendizaje del lenguaje de señas por parte de los sordos y personas comunes, la que se verá apoyada o no según la influencia de la aplicación móvil como variable independiente.

3.4.1 Población y muestra

3.4.1.1 Población

La población está conformada por sordomudos del CEE Huascar Cajias. Estos pacientes son en su mayoría niños, que se encuentran aprendiendo el lenguaje de señas.

3.4.1.2 Muestra

La muestra está conformada por 4 estudiantes nivel primaria del CEE Huascar Cajias.

3.5 DESCRIPCIÓN DE LA METODOLOGÍA

3.5.1 Desarrollo del sistema en base a la metodología Mobile-D

Para el desarrollo de la aplicación móvil se utiliza la metodología Mobile- D siendo que es una metodología ágil de software.

3.5.1.1 Fase de Exploración

Esta fase se dedica a establecer el proyecto y marcar la planificación inicial de cómo será la aplicación para la enseñanza del lenguaje de señas.

La aplicación tiene como finalidad incentivar a los usuarios al aprendizaje de la lengua de señas y motivar a seguir aprendiendo en su casa, siendo que podrán reproducir en video y ver las imágenes con su dispositivo móvil cuando el usuario lo desee.

3.5.1.1.1 Análisis de Requerimiento

La presente tesis se basa en el aprendizaje del lenguaje de señas de forma bidireccional mediante un guante sensorizado y aplicación Android.

Se realizó el prototipo con la finalidad de tener éxito en los objetivos mencionados.

3.5.1.1.2 Requerimientos funcionales

Los requerimientos funcionales son aquellos requerimientos que se espera que debe hacer la aplicación móvil de lenguaje sordomudo.

Tabla 3

Requerimientos funcionales

R1	El sistema debe de ser diseñado con un entorno amigable y de fácil uso.
R2	Los usuarios no tienen la necesidad de loguear.
R3	El usuario debe de ingresar una letra o palabra para su consulta.
R4	El usuario puede visualizar la letra o palabra en el lenguaje sordomudo.
R5	El usuario puede reproducir el texto recibido por el guante a voz
R6	El usuario puede enviar texto a el guante

3.5.1.1.3 Requerimientos no funcionales

Dentro de los requerimientos no funcionales se tiene:

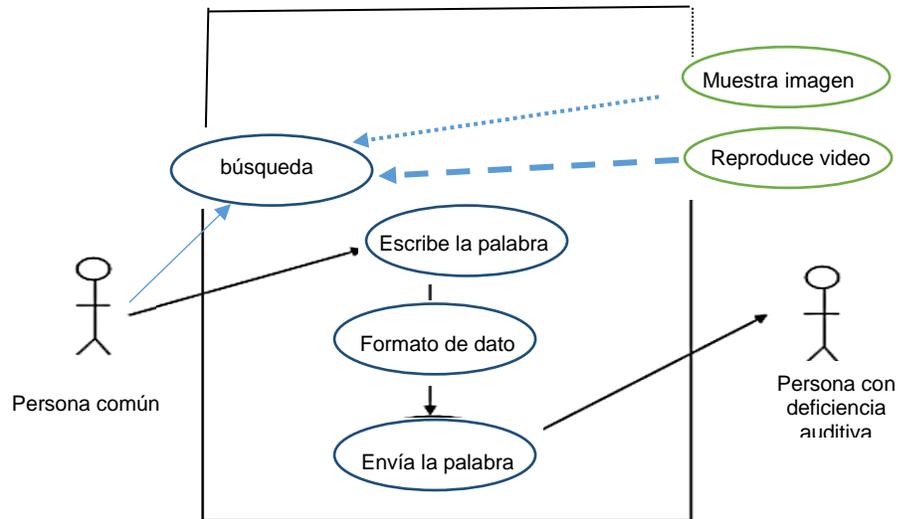
R1	Como circuito base el guante lleva un Arduino Uno
R2	Lleva una placa para la conectividad entre módulo y sensores.
R3	Para la comunicación entre el celular y guante lleva un módulo bluetooth
R4	Sensores flex para identificar valores por el monitor serial
R5	LCD para la visualización de texto
R6	Batería de 9 voltios para alimentar el guante sensorizado.
R7	Requerimiento de dispositivos móviles Smartphone o Tablet con sistema operativo Android
R8	Velocidad de acceso.

3.5.1.1.4 Diagrama de caso de uso

Se realiza el esquema mostrando la relación que tendrá la aplicación de lenguaje de señas con el usuario.

Figura 3.5

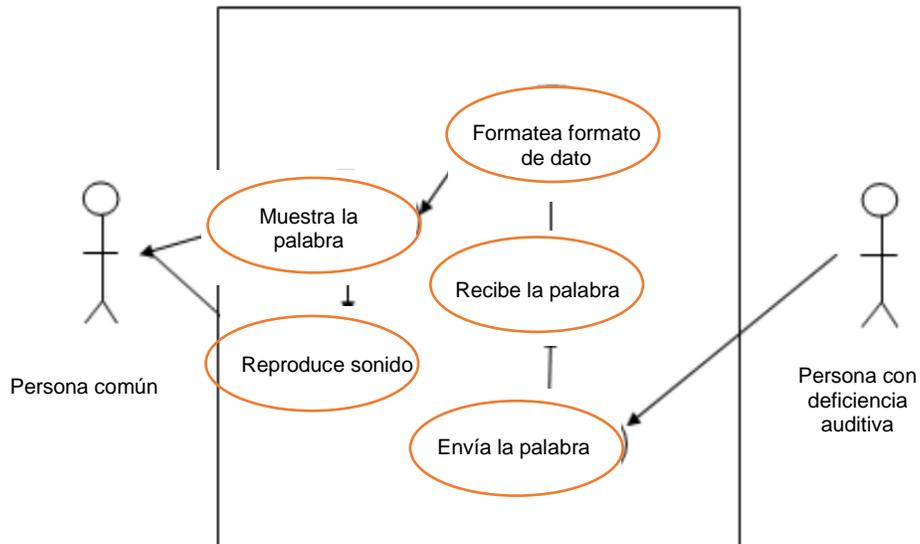
Diagrama de caso de uso de envió de palabra



Nota: Comunicación de celular a guante

Figura 3.5.1

Diagrama de caso de uso de recepción de palabra.



Nota: Comunicación de guante a celular

3.5.1.2 Fase de Inicialización

En esta fase se realizó la identificación, análisis y determinación de los recursos primordiales necesarios para realizar el proyecto.

3.5.1.2.1 Soporte de Software y Hardware

Para el proceso de la aplicación móvil de lenguaje sordomudo en el hardware serán necesarias las siguientes características:

- Memoria RAM mínima 512 Mb
- Memoria libre mínima de 1140 Mb

El sistema operativo requerido será:

- Android v 4.2.2 o superior

3.5.1.2.2 Herramientas necesarias para la aplicación móvil

Para la aplicación móvil se utilizó las siguientes herramientas:

- AppInventor.- Es una herramienta que ayuda a crear apps gratuitas y basada en un lenguaje de bloques.
- Adobe Photoshop.- Para el diseño de la imágenes.

3.5.1.2.3 Diseño de Inicio de la aplicación móvil

Pantalla de inicio

Tiene un botón que nos permite conectar con el módulo bluetooth del guante sensorizado.

Figura .3.5.1

Pantalla para conectar bluetooth



Nota: Botón de listado de bluetooth

Pantalla de envío y recepción de datos

Cuenta con los siguientes botones:

Botón de envío de texto así el guante para que se visualice en un LCD.

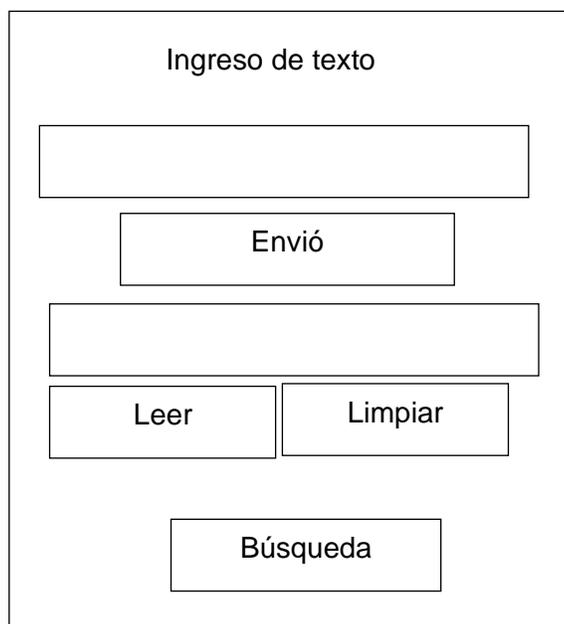
Botón de lectura de texto nos permite leer el texto que se recibe del guante.

Botón de limpieza nos permite limpiar el texto que encuentra en los cuadros de texto.

Botón de búsqueda nos envía a la pantalla de búsqueda en la aplicación.

Figura 3.5.2

Pantalla de envío y recepción de datos



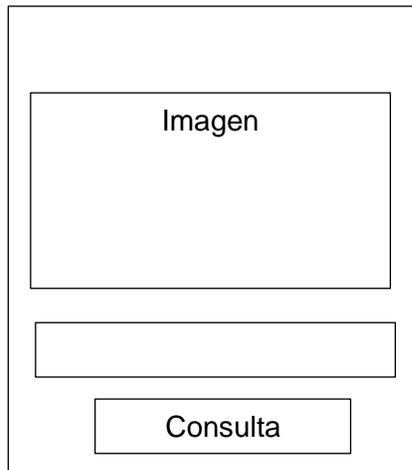
Nota: Boton de busqueda

Pantalla de busqueda

Cuenta con un cuadro de texto en el cual se ingresara el texto a buscar con el boton busqueda según sea letra o palabra nos mostrara una imagen si la consulta es una letra y si es una palabra nos mandara a una pantalla de reproduccion de video.

Figura 3.5.3

Cuadro de imágenes de el absedario en el lenguaje de señas



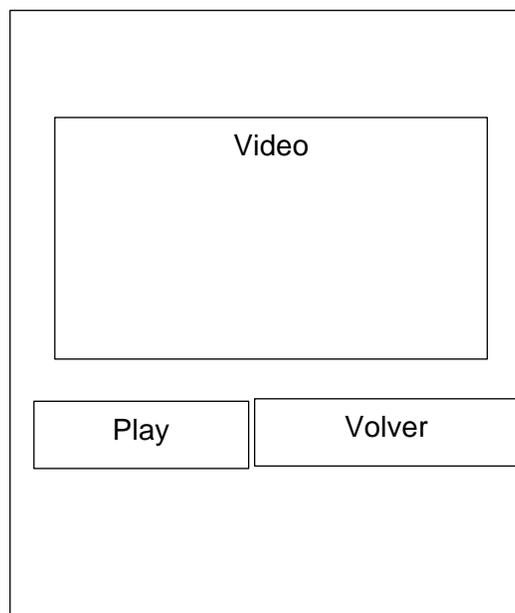
Nota : Botón de consulta de letras o palabras para la visualización en imagen o video

Pantalla de reproducción de video

Esta pantalla cuenta con un botón de reproducción para reproducir el video y también con un botón de volver para regresar a la pantalla de búsqueda.

Figura 3.5.4

Video de palabras en lenguaje sordomudo



Nota: Botones de reproducción y volver a la pantalla de busqueda

3.5.1.3 Fase de Producción

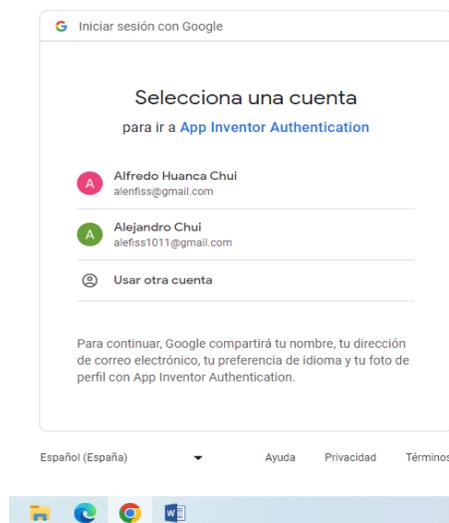
La fase de Producción especifica el proceso de desarrollo de la aplicación, ejecutando a través de iteraciones hasta llegar a satisfacer todas las funcionalidades.

3.5.1.3.1 Diseño de la Aplicación Móvil en App Inventor

Para acceder a la parte del diseñador de la aplicación móvil primeramente accedemos a nuestra cuenta de GMAIL ingresando al siguiente link <http://ai2.appinventor.mit.edu> como se observa en la siguiente figura.

Figura 3.5.4.1

Cuenta de Gmail para ingresar a AppInventor



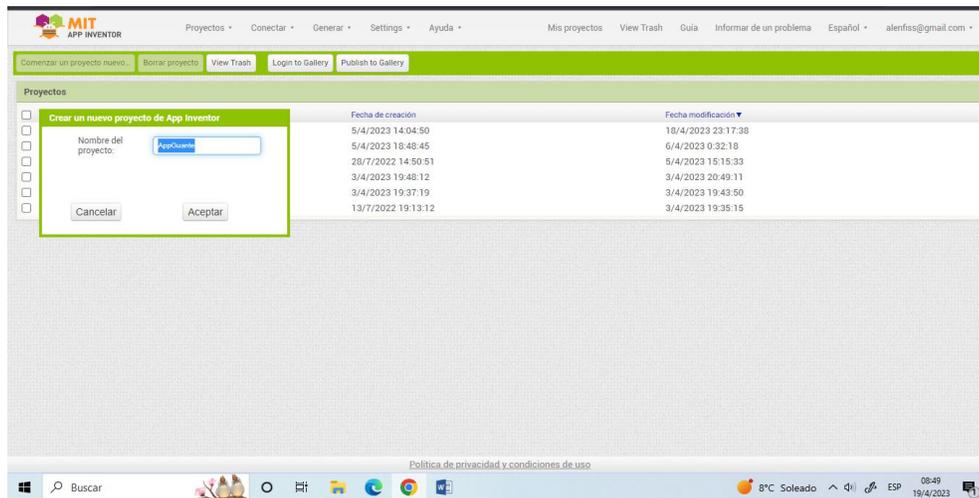
Nota: Elección de una cuenta Gmail

Es importante saber que Google compartirá la dirección de correo electrónico con el MIT AppInventor Versión 2, pero no la contraseña o cualquier otra información personal.

Iniciamos con un nuevo proyecto el cual lo llamamos AppGuante, como se observa en la siguiente figura 3.5.5.

Figura 3.5.5

Nombre del proyecto AppGuante



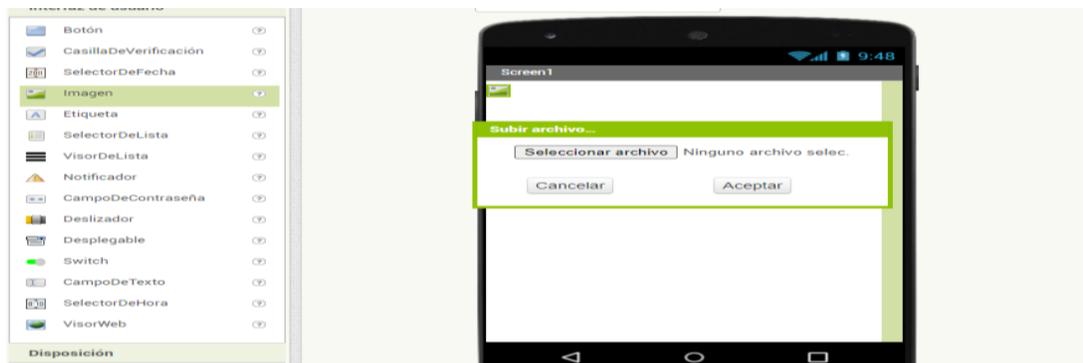
Nota: Pantalla del nombre del proyecto AppInventor

Una vez ingresado ya al entorno de desarrollo de App Inventor podemos observar la pantalla del diseñador en donde se realizó la selección y distribución de cada componente de la paleta que contiene todos los elementos que fueron usados en la aplicación.

En la parte de propiedades en la opción de BackgroundImage se procedió a subir la imagen quedando de la siguiente manera como se observa en la siguiente figura.

Figura 3.5.6

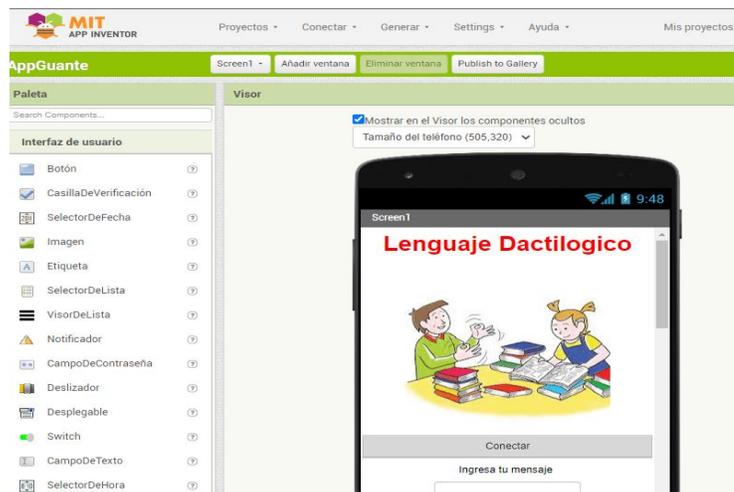
Eleccion de imagen según ubicacion.



Nota: Interfas de usuario imagen

Figura 3.5.7

Imagen de pantalla de conexión



Nota: Cuadro de texto con la imagen de inicio

Se agregó un botón de la Paleta en la parte de Interface de usuario, el cual será programado para conectar y desconectar el dispositivo móvil con el guante traductor de señas básicas por medio de su dispositivo Bluetooth, como se observa en la figura 3.5.8.

Figura 3.5.8

Pantalla final de inicio de aplicación



Nota: Botón de conexión a bluetooth

Se añadieron además funciones que permiten ser usadas en la programación por bloques, como se observa en la figura; estos son un texto de voz, dos relojes que es un componente no visible que permite disparar un temporizador en intervalos de tiempo en milisegundos para dar cierta acción programada, también se añadió notificador para alguna acción incorrecta que se presente, finalmente se insertó componente de conectividad cliente Bluetooth que define las propiedades tanto del nombre y direcciones MAC de los dispositivos Bluetooth que pueden ser emparejadas dentro de un área o rango de cobertura disponible. Si el dispositivo Bluetooth se encuentra disponible, existen solicitudes de conexión con un módulo externo bluetooth que en este caso será proveniente del módulo Bluetooth HC-06 del guante traductor de señas básicas.

Figura 3.5.9

Implementación del Bluetoothcliente1



Nota: Bluetoothcliente permite mostrar el listado de dispositivos encontrados

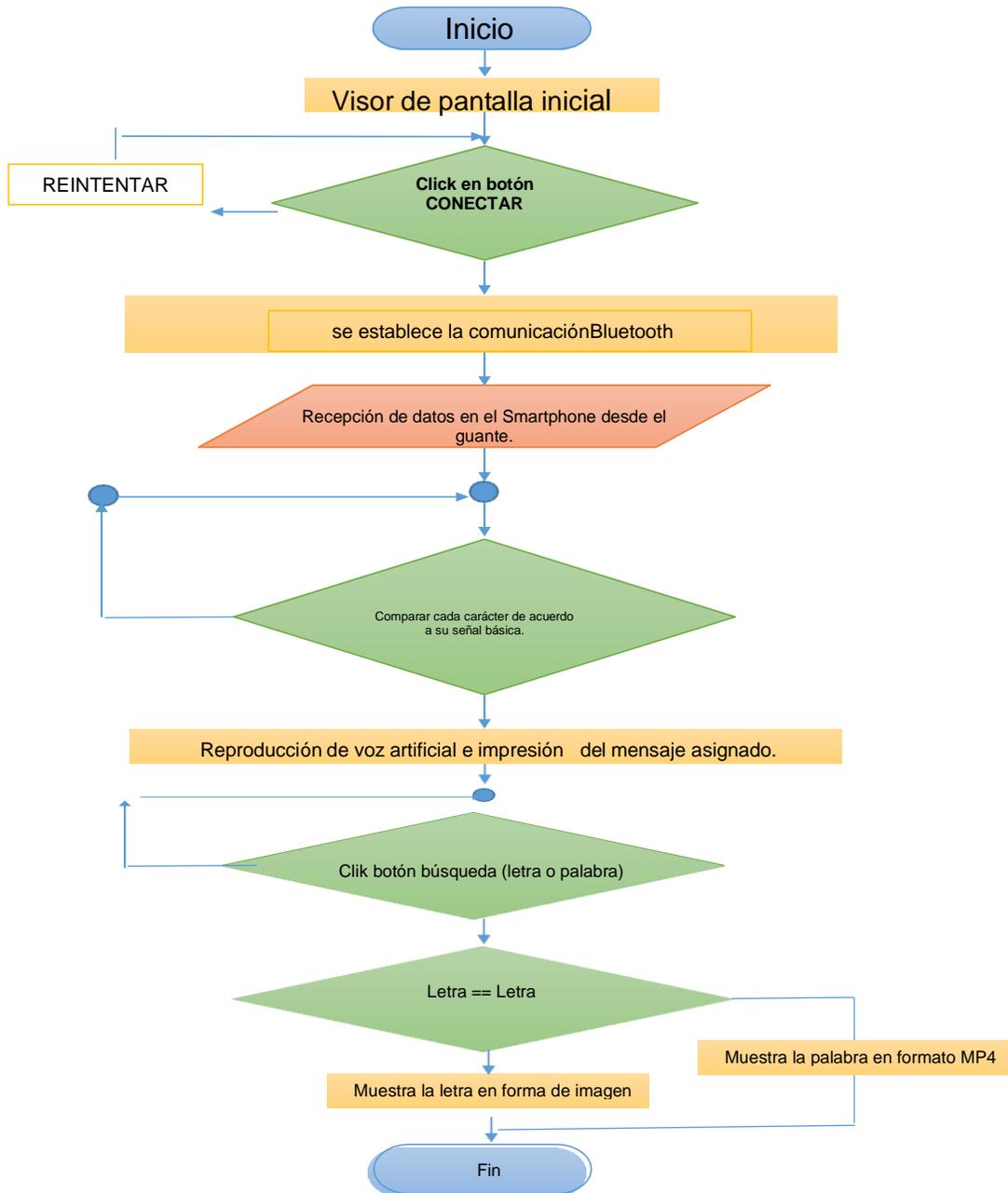
Programación en bloques de la aplicación móvil en App Inventor

El editor de bloques de App Inventor permite facilitar la programación de los componentes que fueron vinculados en el visor de pantalla en la parte del diseño de la aplicación. Cada pieza o bloque que se inserte cumple funciones diferentes ya que trabajan en función de la parte del diseño de la aplicación. Para ello se realizó un

diagrama de flujo que indica el funcionamiento de la aplicación móvil desarrollada. Como se puede observar en el diagrama de flujo se tienen la función del botón CONECTAR que permitirá establecer una conexión Bluetooth por parte del guante y el dispositivo inteligente.

Figura 3.5.10

Algoritmo de aplicación Android



Nota: Condicionales para los diferentes casos según botones

Primero se realizó la construcción del bloque de conectividad del elemento BluetoothClient1, donde la programación realizada corresponde al botón CONECTAR; se utilizó el bloque de eventos que consiste en que toda actividad que realice el usuario, en este caso, tocar el botón CONECTAR, producirá una respuesta de establecer conectividad del dispositivo bluetooth del Smartphone con el módulo Bluetooth HC-06 ubicado en el guante traductor de señas básicas.

Figura 3.5.11

Bloques de conectividad Bluetoothcliente



Nota: ListPicker1 para la lista de dispositivos

A continuación, se realiza el bloque de texto de envío de datos hacia el guante por medio de un botón de nombre envió.

Figura 3.5.12

Bloques del botón enviar

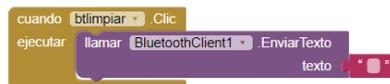


Nota: Bluetoothcliente1 para el envío de texto

Ubicamos el bloque que permite limpiar los datos enviados y recibidos en los cuadros de texto.

Figura 3.5.13

Bloques del botón limpiar



Nota: Bluetoothcliente1 para limpiar el texto en LCD

El bloque leer nos permite leer los datos recibidos por el guante los cuales se visualizan en un cuadro de texto.

Figura 3.5.14

Bloques del botón leer

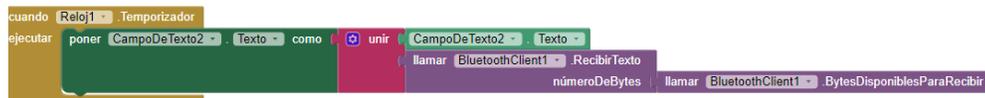


Nota: llamar la función leer texto en el campo de texto2

El componente no visible Clock1 que ha sido elegido en el diseño de esta aplicación, permite ofrecer un temporizador que se sincroniza con el reloj interno del Smartphone, y establece regularmente un cálculo de tiempo para recibir datos provenientes del módulo Bluetooth HC_06 del guante traductor de señas básicas mediante la conexión Bluetooth.

Figura 3.5.15

Bloques del reloj1

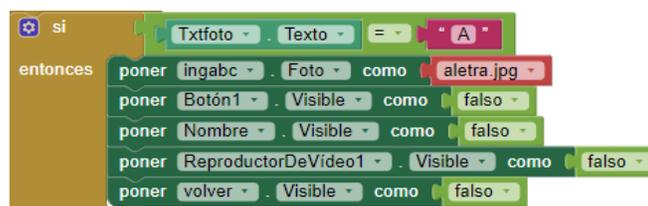


Nota : Reloj temporizado con 2 segundos

A continuación, se realizó los bloques que corresponden a cada letra del abecedario básico en el lenguaje de señas, donde se asignan letras en los bloques de texto de comparación, conjuntamente con los bloques del componente Txtfoto y su respetiva imagen.

Figura 3.5.16

Bloque condicional de la letra A



Nota: Ruta y nombre de la imagen correspondiente a la letra “A”

Se eligió el bloque de control que plantea una condición IF en donde la comparación con respecto a el texto ingresado se ha asignado con la letra B en el bloque de comparación de texto; cuando compruebe la condición permitirá visualizar la imagen asociada a dicha letra.

Figura 3.5.17

Bloque condicional de la letra “B”

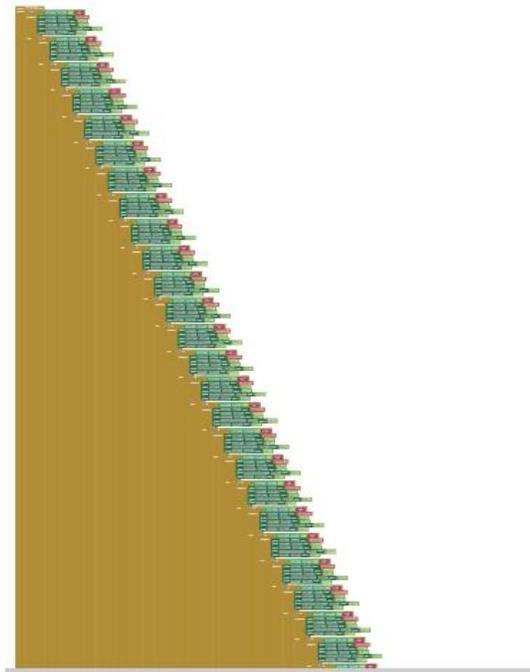


Nota: Nombre” bletra” correspondiente a la letra “B”

Finalmente, los bloques de comparación correspondientes a cada letra del lenguaje de señas, se unen formando un solo bloque como se observa en la siguiente figura.

Figura 3.5.18

Bloques de todo el abecedario



Nota: Rutas de imágenes según cada letra del abecedario

Para las palabras se realizó las ocho comparaciones que corresponden a las necesidades básicas vistas en las siguiente tabla 4 , donde se asignan las palabras en

los bloques de comparación conjuntamente con los bloques del componente reproductor de video.

Tabla 4

Palabras en formato MP4

Texto ingresado	Seña básica en formato de video
HOLA	Hola.mp4
BIEN	Bien.mp4
SI	Si.mp4
COMO ESTAS	Comoestas.mp4
DE NADA	De_nada.mp4
GRACIAS	Gracias.mp4
MAL	Mal.mp4
NO	No.mp4
PERDON	Perdón.mp4
Permiso	Permiso.mp4

Nota: palabras básicas en lenguaje sordomudo

Se eligió el bloque de control que plantea una condición IF en donde la comparación con respecto a la primera palabra que se ha asignado con la palabra HOLA en el bloque de comparación de texto; cuando la condición se cumple el bloque ReproducciondeVideo permitirá asignar el primer video correspondiente a dicha palabra, como se observa en la figura 3.5.19.

Figura 3.5.19

Bloques para la palabra “Hola”



Nota: Bloque if comparativa palabra “Hola”

Se eligió el bloque de control que plantea una condición IF en donde la comparación con respecto a la palabra ingresada se comparará con la palabra “BIEN” en el bloque de comparación de texto; el bloque de reproducción de video permitirá asignar el video correspondiente a dicha palabra, como se observa en la figura 3.5.20.

Figura 3.5.20

Bloques para la palabra “Bien”

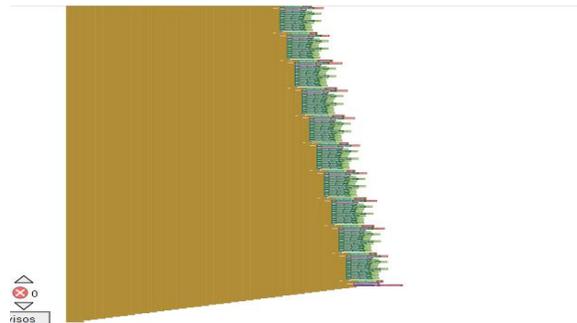


Nota: llamada a la función reproducción según su nombre

Finalmente, los bloques de comparación correspondientes a cada palabra, se unen a los bloques de letra formando un solo bloque como se observa en la siguiente figura 3.5.21.

Figura 3.5.21

Bloques de las 10 palabras básicas del lenguaje de señas



Nota: condicionales comparativos según palabra

3.6 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

El primer método para emplear la recolección de datos, es el Hipotético – Deductivo donde se le describe como un procedimiento que parte de unas afirmaciones en calidad de hipótesis y busca afirmar o refutar dicha hipótesis confrontando dichas

aseveraciones con los resultados en base a hechos.

El segundo método que se pondrá en uso es el método de la implementación. Este método de implementación supone su uso cuando se desarrollan nuevas aplicaciones de software para un uso determinado ya que esta nueva herramienta se deberá poner a prueba en el campo a fin de demostrar si realmente posee las ventajas propuestas y se obtienen resultados como los esperados (Berndtsoon,2001)

3.7 COSTO DEL PROTOTIPO

En este apartado se detallan los costos de fabricación del prototipo y desarrollo del software, basado en los resultados obtenidos tras las pruebas de funcionamiento. Para el cálculo del costo de fabricación; en vista que el prototipo es un producto tecnológico, se considera materia prima a todos los componentes electrónicos principales, los cuales se adquieren de proveedores de mercado y son la base para el desarrollo del producto final. Como costos indirectos, se tuvo en cuenta los elementos consumibles, la mano de obra de fabricación del circuito y el desarrollo de software.

3.7.1 Costo De Desarrollo De Software

El salario de un programador es de 3500 Bs según Federico Sarmiento (Sarmiento, 2021).

Tabla 5

Costo de ingeniería

Descripción	Horas	Precio / hora	Precio total
Diseño de software	30	30 Bs	900Bs
Diseño de hardware	18	30 Bs	540Bs
Montaje de hardware	120	30 Bs	3600Bs
Pruebas funcionales	40	30 Bs	1200Bs
Confección de documentación	120	30 Bs	3600Bs
Total			9840Bs

Costos de materiales

En este apartado se detallan todos los componentes utilizados en la realización del proyecto, junto con la descripción de las características.

Tabla 5.1

Costos de materiales

Referencia	Unidad	Precio unidad	Total
Cables	7 metros	3 Bs	21Bs
Resistencias	10	0.3Bs	3Bs
Guante	1	5Bs	5Bs
Placa	1	5Bs	5Bs
Batería	1	15Bs	15Bs
Modulo Bluetooth	1	45Bs	45Bs
Arduino Uno	1	100Bs	100Bs
pulsadores	5	1,50Bs	7,5Bs
Pantalla LCD 16*2	1	45Bs	45Bs
Sensores flex	5	180Bs	900Bs
Total			1146,5Bs

Costos totales

Se suman todos los costos del proyecto.

Tabla 5.2

Costo total del proyecto

Descripción	Precio total
Costos de ingeniería	9840Bs
Costos de materiales	1146,5Bs
Total, bolivianos	10986,5Bs
Total, en dólares	1569,5\$

Costos Beneficio

El prototipo de guante sensorizado y aplicación Android tiene un beneficio bastante accesible en cuestión de costos ya que su producción a mayor escala no lograría rebasar el costo de un salario a un intérprete que traduzca las señas.

CAPÍTULO IV

4. PRUEBAS Y RESULTADOS

En el presente capítulo se detallará las pruebas realizadas para la ejecución correcta de los intérpretes, se verificará el funcionamiento del sistema electrónico y la aplicación, la interpretación correcta de señas, la verificación de la hipótesis del proyecto.

4.1 DISEÑO Y DESARROLLO DE PRODUCTO

Se aborda la etapa de diseño del modelo, donde se siguió una metodología basada en el libro “Diseño y desarrollo de productos” de Karl T, Ulrich. El mismo abarca 3 funciones principales en el desarrollo de un producto: mercadotecnia, diseño y manufactura. Sin embargo, la orientación de un prototipo de guante sensorizado con aplicación móvil no contempla una orientación comercial, por lo cual sólo nos guiaremos para la etapa de diseño.

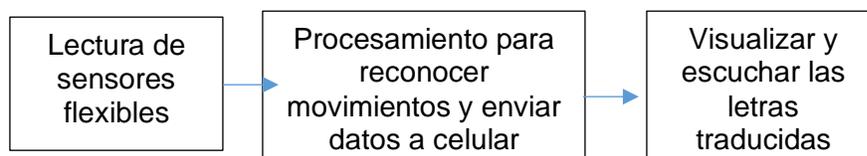
4.1.1 Diseño de producto

4.1.1.1 Fase inicial

El prototipo desarrollado permite establecer una comunicación bluetooth, enviando la información a una aplicación para dispositivos móviles, en donde se realiza la traducción de las señas a texto y escucharlas, de la misma forma, el receptor puede comunicarse con la persona discapacitada a través de mensajes que son enviados mediante una conexión bluetooth y visualizados en una pantalla LCD.

Figura 4.1

Esquema del proyecto



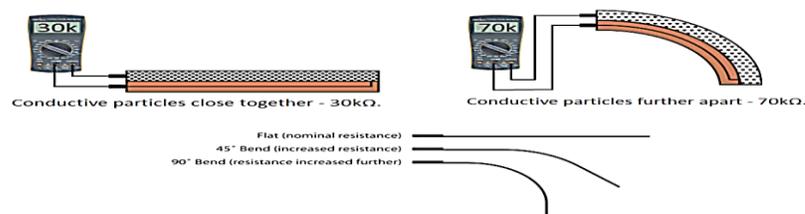
Nota: bloques del proyecto

Elección del sensor

Al momento de implementar el prototipo electrónico, la elección del tipo de sensor a utilizar depende mucho de los requerimientos al cual va a ser sujeto dicho sensor; ya que, dependiendo de las necesidades del prototipo, el sensor debe adaptarse de la mejor manera para que pueda cumplir con la funcionalidad del guante traductor de señas básicas. A pesar de la gran variedad de sensores existentes en el mercado y de la gran cantidad de funcionalidades y aplicaciones a los que son sujetos, se procedió a optar por utilizar el sensor flexible, ya que este reúne todas las características específicas que permiten adaptarse de la mejor manera en el guante.

Figura 4.1.1

Medición de resistencia de sensores flex



Nota: Angulo de flexión (automatización robótica)

La tabla que se presenta a continuación (Tabla 6), indica los valores de cada uno de los sensores adquiridos para la realización del guante.

Tabla 6

Valores resistivos de los sensores flex

Sensores(S)	S1	S2	S3	S4	S5
$K\Omega$					
Valor Mínimo ($K\Omega$)	21	19.7	22.3	22.3	24.6
Valor Máximo ($K\Omega$)	38	30	40	32.7	35
Diferencia ($K\Omega$)	17	10.3	17.7	10.4	10.4

Nota: Máximos y mínimos valores de los sensores flex

Se puede apreciar en esta tabla que en realidad si existen diferentes valores de resistencia en los sensores medidos a pesar de ser de las mismas

características, luego de hacer esto se realiza la ubicación de los sensores en cada dedo como más convenga.

4.1.2 Fase de producción

Estructura del guante electrónico

El prototipo está dividido en dos partes; la primera es el guante en donde se encuentran los sensores flexibles que permiten detectar la posición de los dedos al formar las letras del alfabeto dactilológico y la segunda un brazalete el cual se coloca alrededor de la muñeca en donde se encuentra la pantalla LCD, la tarjeta electrónica con los respectivos módulos que permiten realizar el procesamiento y presentación de los datos.

Elección del guante y colocación de sensores

El guante para el desarrollo del proyecto debe ser robusto y ergonómico por el constante uso al cual será sometido por lo que se optó por uno de buena calidad. Una vez teniendo el guante a usar procedemos a realizar una simulación visual respecto a la ubicación de los sensores lo que da una mejor idea de diseño al momento de implementar en el proyecto.

Figura 4.2

Ubicación de los sensores flex en el guante



Los cinco sensores se encuentran conectados con resistencias de 47Kohm cada una. Hay que cerciorarse que cada elemento este asegurado al momento de ubicarlos ya que así nos evitamos que se desconecten o rompan y tengamos problemas posteriores en su funcionamiento.

Figura 4.2.1

Guante sensorizado.



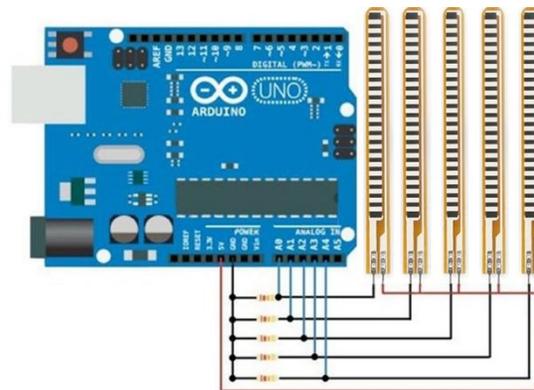
Nota: Sensores flex en el guante con sus conexiones.

Diagrama de conexión de los sensores con la placa Arduino

Una vez teniendo todos los valores de los elementos electrónicos se procede a realizar las conexiones, las cuales van de los pines A0 al A4 del controlador que obedecerán al movimiento de cada dedo como se observa en la figura 4.2.2.

Figura 4.2.2

Simulación en Proteus



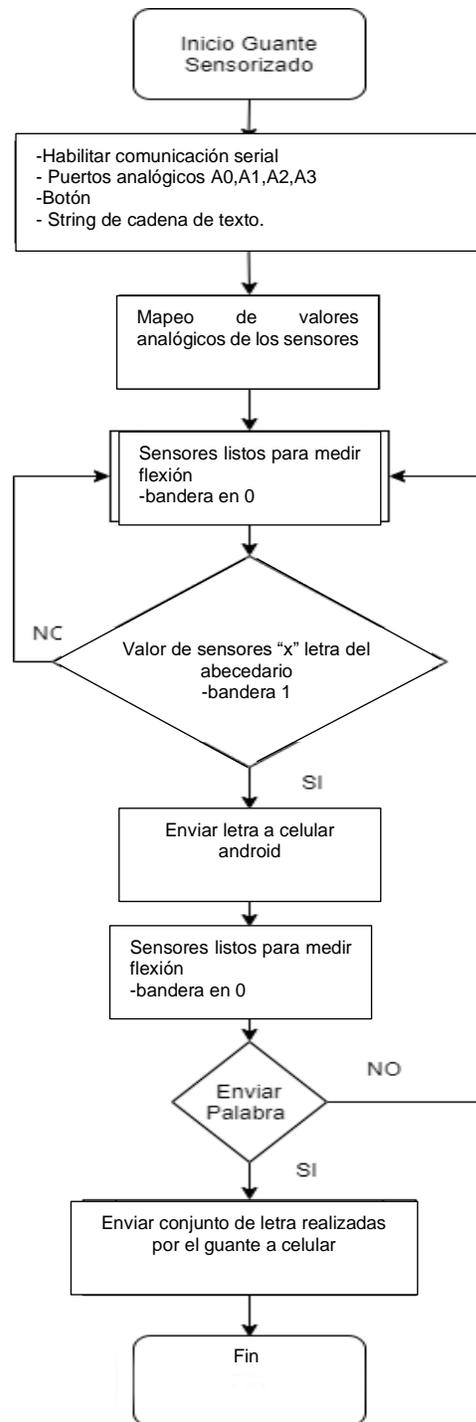
Nota: Esquema de las conexiones con los sensores y sus pines

Diagrama de flujo del guante sensorizado

En la Figura 4.2.3 se detalla el diagrama de flujo que se obtiene del guante sensorizado.

Figura 4.2.3

Diagrama de flujo del programa en Arduino



Nota: Estructura detallada del diagrama de flujo

Conexión inalámbrica del guante

Se usará un módulo bluetooth que estará conectado a los pines (pines10 (RX) y 11 (TX)) del Arduino Uno para enviar los datos del guante hacia el teléfono celular como se muestra en la figura 4.2.4.

Figura 4.2.4

Conexión de modulo Bluetooth a Arduino Uno



Nota: Montaje en protoboard para pruebas

Pantalla LCD del guante

Se uso una pantalla LCD de 16x2 para la visualización de las palabras que serán enviadas desde el celular.

Figura 4.2.5

LCD para visualización de texto



Nota: pines del LCD para el conexionado en Arduino

Conexión de todos los elementos en el guante

Lo siguiente es realizar la conexión de todos los componentes, se procede a soldar todos los elementos en la placa y a realizar las conexiones de alimentación en las borneras que unen a los elementos con la placa. Luego se realiza las pruebas necesarias para corroborar el correcto funcionamiento de todos los componentes que forman el prototipo del guante sensorizado. En la figura 4.2.6, se muestra el resultado final del guante junto a la conexión de todos sus elementos.

Figura 4.2.6

Guante sensorizado con pantalla LCD



Nota: Guante con sus sensores, placa, controlador y batería.

Programación de Arduino para el funcionamiento del guante

A continuación, se explicará brevemente el código de programación que se cargó en el Arduino Uno.

Figura 4.2.7

Código de mapeo para los sensores flex.

```
// Menique
const float Resistencia_SinFlex = 44000.0;
const float Resistencia_Flexion = 98000.0;
int flexADC0 = analogRead(menique);
float flexV0 = flexADC0 * 4.98 / 1023.0;
float Vsensor= 0.0118*flexV0+1.814
float lecturaMenique = map(Vsensor, Resistencia_SinFlex, Resistencia_Flexion,
                          0, 90.0);

delay(10);
```

Nota: Código para calcular el ángulo de deformación de cada sensor.

Para identificar la letra que se forma con el guante se tiene que cumplir con los parámetros requeridos por cada una de ellas, tomaremos como ejemplo a la letra A, en la Tabla 7. se muestran los ángulos que se midieron en cada dedo para ser reconocida.

Tabla 7

Parámetros de la letra A.

Ubicación de sensores flex	Grados (°)
Meñique	90
Anular	90
Medio	90
Índice	90
Pulgar	90
muñeca	70

Nota: Valores medidos en grados para identificar a la letra A.

Una vez teniendo los valores generados de cada sensor al deformarse y los parámetros que cada letra necesita se procede a realizar una comparación entre ellos y así se sabrá la letra que se formó en el guante. En la figura 4.2.9. se muestra como ejemplo el código que se realizó para la obtención de la letra A.

Figura 4.2.9

Código de programación para obtener la letra A.

```
if (lecturaMenique > 55 && lecturaMenique < 95 && lecturaAnular > 50 && lecturaAnular < 90 && lecturaMedio > 60 &&
lecturaMedio < 100 && lecturaIndice > 50 && lecturaIndice < 90 && lecturaPulgar > -5 && lecturaPulgar < 10 &&
lecturaMuneca > -5 && lecturaMuneca < 5 && a == 0) {
  cadena += "A";
  Serial2.print("  ");
  Serial2.print('A'); //envia dato letra A a modulo bluetooth
  Serial2.print("#");
  Serial.println('A');//envia dato letra A al monitor serie
  a = 1;
  delay(100);//tiempo de envio de cada dato 500ms
}
```

Nota: Código para identificar la letra A.

Una vez generada las letras, estas son enviadas a la aplicación móvil, letra por letra para formar una palabra, en la Figura 4.2.10. se muestra el código de programación que se usó para realizar el envío a la aplicación; Esto es todo con respecto a la construcción y calibración del guante sensorizado.

Figura 4.2.10.

Código de programación para él envío de datos.

```
if (datBot == 1) {
  Serial.println(" ");
  Serial.print("TEXTO: ");
  Serial.print(cadena);
  Serial2.print(" ");
  Serial2.print(cadena);
  Serial2.print("#");
  delay(100);
  cadena = "\0";
  Serial.println(" ");
  delay(100);
}
```

Nota: Código para enviar letras o palabras a los dispositivos

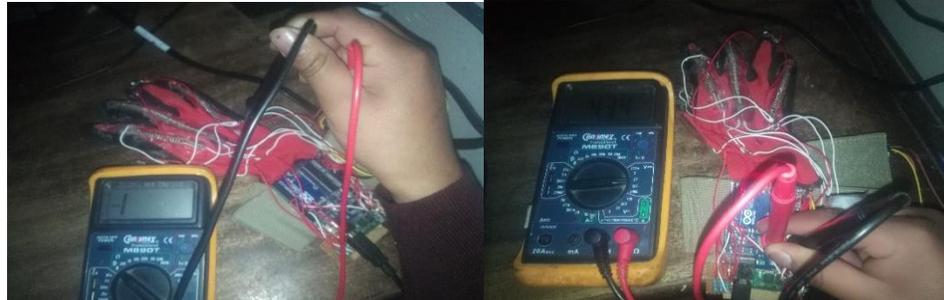
4.2 DEMOSTRACIÓN DEL PROTOTIPO

4.2.1 Pruebas del sistema electrónico

Para comprobar el buen funcionamiento del sistema electrónico primero se realizó la medición y se verificó que el voltaje de entrada se aproxime a 5V en plena carga, para que los elementos electrónicos trabajen correctamente en el guante como muestra en la figura 4.2.11. Posteriormente se revisa la continuidad en los cables de cada conexión.

Figura 4.2.11

Comprobación de alimentación del guante y conexiones



Nota: prueba del conexionado de los sensores

4.2.2 Rangos de entradas analógicas

Se ha realizado pruebas para obtener los rangos de los sensores del sistema, las cuales consisten en ejecutar el movimiento de extender y flexionar (abrir y cerrar) los dedos varias veces para obtener los valores de los sensores. La Tabla 8 muestra los valores de cada sensor.

Tabla 8

Datos de los sensores.

Posición	Dedo pulgar	Dedo índice	Dedo medio	Dedo anular	Dedo meñique
Extendido	430	440	640	430	470
Flexionado	120	140	480	150	145

De la tabla se puede observar las medidas de los sensores donde al extender los valores tienden a 600 y al flexionar al 100.

4.2.3 Pruebas de funcionamiento

Para evaluar el funcionamiento del guante intérprete se realizó 10 repeticiones para cada letra del abecedario como se observa en la tabla 9.

Tabla 9*Letras del abecedario con su respectivo error para 10 repeticiones.*

Nº	señas	Numero de repeticiones	Seña correcta	Seña incorrecta	Error(%)
1	Letra A	10	10	0	0
2	Letra B	10	10	0	0
3	Letra C	10	8	2	20
4	Letra D	10	9	1	10
5	Letra E	10	9	1	10
6	Letra F	10	7	3	30
7	Letra G	10	9	1	10
8	Letra H	10	10	0	0
9	Letra I	10	8	2	10
10	Letra J	10	7	3	30
11	Letra K	10	8	2	20
12	Letra L	10	9	1	10
13	Letra M	10	8	2	20
14	Letra N	10	8	2	20
15	Letra Ñ	10	7	3	30
16	Letra O	10	8	2	20
17	Letra P	10	8	20	20
18	Letra Q	10	7	3	30
19	Letra R	10	7	3	30
20	Letra S	10	8	2	20
21	Letra T	10	9	10	10
22	Letra U	10	9	1	10
23	Letra V	10	9	1	10
24	Letra W	10	7	3	30
25	Letra X	10	7	3	30
26	Letra Y	10	10	0	0
27	Letra Z	10	6	4	40
Error promedio					17.40

El error promedio es de 17.40 % en la ejecución de las señas, es un error aceptable.

Tabla 10

Palabras básicas del lenguaje de señas para 10 repeticiones con su porcentaje de error.

PONDERACIÓN				
Señal	Numero de repeticiones	Seña correcta	Seña incorrecta	Error(%)
Hola	10	10	0	0
Bien	10	9	1	10
Gracias	10	8	2	20
Perdon	10	8	2	20
Si	10	8	2	20
No	10	9	1	10
Como estas	10	8	2	20
De nada	10	8	2	20
Mal	10	10	0	0
Permiso	10	8	2	20
Error promedio				60

El error promedio es de 14 % en la ejecución de las palabras, es un error aceptable.

4.2.4 Prueba de aprendizaje

Para realizar las pruebas de aprendizaje se realizó 4 preguntas en el test, se avaluó dos grupos estudiantes y familiares.

Personas participantes en el test cuatro estudiantes y cuatro familiares total ocho, preguntas realizadas en el cuestionario cuatro.

Tabla 11

Ponderación de las preguntas

PONDERACIÓN				
Calificación	Escribe el dictado dactilológico "Gracias"	Escribe el nombre de la seña "perdón"	Marca la seña de "Como estas"	Encuentra la letra "J"
Bien	6	5	5	3
Mal	2	3	3	5
Total	8	8	8	8

Nota: Cuatro de respuestas correctas e incorrectas en el cuestionario

Se obtuvo como análisis que el aprendizaje depende mucho de los niños debido a que su grado de discapacidad no es el mismo, sin embargo, tenemos un aprendizaje alto.

4.3 PRUEBA DE FUNCIONAMIENTO DE SOFTWARE

En esta prueba se realizó el funcionamiento de la aplicación Android la conexión con el guante y la recepción y envió de datos se realizaron 10 pruebas de funcionamiento.

Tabla 12

Aspectos considerados en el funcionamiento

Aspectos a evaluar	valoración		Observaciones
	Si	No	
Conexión Bluetooth	x		La conexión fue exitosa. Pero baja la calidad de la señal a un estado intermedio a medida que aumenta la distancia entre guante y celular.
Búsqueda de imágenes relacionadas a letras	X		Tamaño de imágenes variadas

Aspectos a evaluar	valoración		Observaciones
	Si	No	
Búsqueda de palabras	X		Reproducción de video de diez palabras básicas .
Envío de palabras a el guante	X		Visualización de los datos enviados por medio de pantalla LCD del guante
Envío de letras del guante al celular	X		Reproducción de texto y visualización en la aplicación.

4.4 PRUEBA DE HIPÓTESIS

Para demostrar la validez de la hipótesis se realiza la prueba de hipótesis para comprobar de forma estadística si se acepta o se rechaza.

4.4.1 Estado de la Hipótesis

Para realizar la verificación de la hipótesis planteada de esta tesis se utilizó la siguiente formulación de hipótesis:

- Hipótesis nula = H_0
- Hipótesis alternativa = H_1

Hipótesis: La presente tesis, guante con sensores y aplicación móvil para la interpretación bidireccional del alfabeto dactilológico tiene un nivel de confianza del 95% de funcionalidad.

Hipótesis Nula: La presente tesis, guante con sensores y aplicación móvil para la interpretación bidireccional del alfabeto dactilológico no beneficia la comunicación entre personas sordomudas y personas que desconozcan del lenguaje de señas.

Hipótesis Alternativa: La presente tesis, guante con sensores y aplicación móvil para la interpretación bidireccional del alfabeto dactilológico beneficia la comunicación entre personas sordomudas y personas que desconozcan del lenguaje de señas”.

Modelo matemático

$H_0 > H_1$

$H_0 < H_1$

Modelo estadístico

$$x^2 = \sum \frac{(O - E)^2}{E}$$

x^2 = Chi cuadrado

Σ = Sumatoria

O = Frecuencia observada

E = Frecuencia esperada

Se trabajó con cuatro niños y cuatro familiares, se realizaron cuatro preguntas como se observa en la siguiente tabla 13.

Tabla 13

Tabla de frecuencias observadas

Calificación	PONDERACIÓN				Total
	Escribe el dictado dactilológico "Gracias"	Escribe el nombre de la seña "perdón"	Marca la seña de "Como estas"	Encuentra la letra "J"	
Bien	6	5	5	3	19
Mal	2	3	3	5	13
Total	8	8	8	8	32

Se evaluó la capacidad de los estudiantes para aprender las señas con el guante y la aplicación Android. Posteriormente se calcula la frecuencia esperada con la ecuación 1 . La Tabla 14 indica los valores de frecuencia esperada.

$$E = \frac{O_i * O_j}{o} \dots\dots\dots \text{ecuación 1}$$

En donde:

O_i : Frecuencia observada total fila

O_j : Frecuencia observada total columnas

o : Frecuencia observada total

E : Frecuencia esperada

Tabla 14

Valores de frecuencias esperadas

PONDERACIÓN				
Calificación	Escribe el dictado dactilológico "Gracias"	Escribe el nombre de la seña "perdón"	Marca la seña de "Como estas"	Encuentra la letra "J"
Bien	4,75	4,75	4,75	4,75
Mal	3,25	3,25	3,25	3,25

Por último se obtiene los valores de chi cuadrado con la ecuación 2 , como se observa en la Tabla 15.

$$x^2 = \sum \frac{(O-E)^2}{E} \text{ ----- ecuación 2}$$

En donde:

x^2 : Chi cuadrado calculado

I : Número de filas

J : Número de columnas

Tabla 15

Valores calculados de Chi-Cuadrado.

PONDERACIÓN				
Calificación	Escribe el dictado dactilológico "Gracias"	Escribe el nombre de la seña "perdón"	Marca la seña de "Como estas"	Encuentra la letra "J"
Bien	0,33	0,013	0,013	0,64
Mal	0,48	0,019	0,019	0,95
Chi-Cuadrado				2,464

A continuación, se calcula los grados de libertad, que es un estimador de número de categorías independientes en una prueba, se obtuvo mediante la ecuación 3.

$$G_D = (I - 1) * (J - 1) \dots\dots\dots \text{ecuación 3}$$

$$G_D = (2 - 1) * (4 - 1)$$

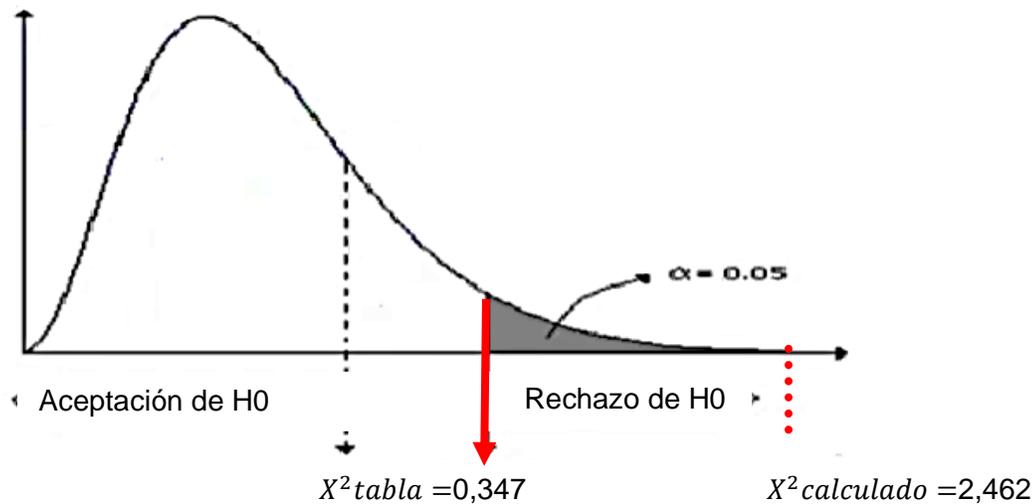
$$G_D = 3$$

El nivel de confianza o nivel de significación alfa para el proyecto fue de 0.05, el grado de libertad G_D calculado es 3. Con estos datos se observó el chi-cuadrado tabulado (x^2) en la tabla del Anexo B (Valores Críticos de la Distribución Chi Cuadrado) que fue de 0.357. El valor calculado del chi-cuadrado (x^2 prueba) fue de 2.464, entonces como (x^2 prueba) es mayor que (x^2 tabla) la hipótesis nula se rechazó y la hipótesis de trabajo fue aceptada.

Se verificó el cumplimiento de la hipótesis planteada, después de realizar las diferentes pruebas y obtener los resultados positivos, el guante y la aplicación facilitaron el aprendizaje del lenguaje en los alumnos con discapacidad auditiva.

Figura 4.2

Gráfico de la región crítica donde se rechaza la hipótesis nula.



CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- La aplicación desarrollada conjuntamente con el guante electrónico contó con una interfaz de fácil manejo facilitando la comunicación bidireccional entre emisor y receptor.
- Para el desarrollo del proyecto se utilizó un módulo bluetooth HC-O6 , sensores flex de 1M Ohm , pantalla LCD, resistencias y placa Arduino Uno.
- Se logró calibrar los sensores flex del guante y visualizar las letras dependiendo del rango para su reproducción letra a letra en la aplicación.
- Se logró establecer la conexión bluetooth entre el guante y el celular.
- Se logró desarrollar la aplicación Android para la visualización, reproducción a voz y búsqueda de palabras del lenguaje sordomudo.
- Se realizó las pruebas pertinentes como los rangos adecuados para los sensores y la conectividad del guante con el celular para comprobar el buen funcionamiento del sistema guante y aplicación, esto se evidencia a través del gran número de respuestas como nivel alto de satisfacción en la utilización y facilidad de manipulación de los recursos presentados.

5.2 RECOMENDACIONES

- Se sugiere que el guante sensorizado se use para enviar letra a letra alguna palabra al celular para posteriormente visualizar la palabra en la aplicación.
- Se sugiere que la aplicación sea instalada en un celular con sistema operativo Android.
- Se sugiere que el estudiante comience su aprendizaje con señas básicas.
- Se sugiere que la conexión inalámbrica para el guante sea a través del módulo bluetooth logrando así un alcance de 10m máximo de conexión.
- Para tener una mayor capacidad y variedad en palabras traducidas se sugiere utilizar una placa Arduino Mega o Esp 32.
- Para lograr mayor precisión en las señales se sugiere trabajar con sensores digitales o sensor Kinect.

BIBLIOGRAFÍA

- Collazos, C., y Bustos, V. (2015). Aplicación móvil para el aprendizaje de la lectoescritura con FitzGerald para Niños con Discapacidad Auditiva.
- Chuan, C.-H., y Guardino, C. A. (2016). Designing SmartSignPlay. Companion Publication of the 21st International Conference on Intelligent User Interfaces - IUI '16 Companion
- CONADIS. (2019). Estadísticas de Discapacidad – Consejo Nacional para la Igualdad de Discapacidades. Retrieved June 18, 2019.
- Gavilanes, W., y Silva, A. (2018). Aprendizaje del lenguaje de señas mediado por las TIC Use of ICT for sign language learning. In Pág (Vol. 39).
- Ortega, H. (2014). Aplicación de apoyo visual para el aprendizaje del alfabeto dactilológico con sensores microcontrolador.
- Gámez, I. (2015). Los Modelos Tecno-Educativos, revolucionando el aprendizaje del siglo XXI.
- Lazalde Alan, Torres Jenny, Vila-Viñas David, “Recomendaciones para el fomento de la innovación ciudadana con Arduino”
- Guzmán, D. (2017). Guante Electrónico para Traducir de Lenguaje de Señas a Caracteres con Voz Artificial y Conexión Inalámbrica a Dispositivos. http://repo.uta.edu.ec/bitstream/123456789/25193/1/Tesis_t1222ec.pdf
- Hernández, C., Márquez, H., & Martínez, F. (2015). Propuesta Tecnológica para el Mejoramiento de la Educación y la Inclusión Social en los Niños Sordos. <https://doi.org/10.4067/S0718-50062015000600013>

ANEXOS

ANEXO A

MANUAL DE USUARIO

Conexión bluetooth con guante y celular

Para empezar, debemos de alimentar el guante con una batería

Figura 8

Conexión de guante sensorizado con batería



Luego encendemos el celular activamos el bluetooth del mismo.

Figura 9

Activado del bluetooth



A continuación, se deberá ingresar en la aplicación e iniciar la conexión

Figura 10

Entorno de conexión de la aplicación Android

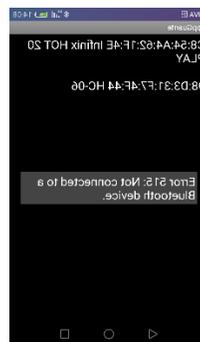


Enlace vía bluetooth

Se deberá vincular el celular con el módulo bluetooth del guante seleccionando el módulo de una lista de dispositivos se seleccionará el módulo HC-06

Figura 11

Selección de dispositivo



En la pantalla de inicio el botón enviar que se ve en la figura 10 da la función de enviar las palabras o mensaje hacia el guante el cual se visualizara en el LCD del mismo.

Figura 12

Envío de texto hacia el guante



El botón leer nos permite leer el texto que se envía del guante a el celular

Figura 13

Recepción de datos en el celular visualizado en cuadro de texto



El botón limpiar nos permite limpiar el texto que se encuentra en el cuadro de texto.

Figura 14

Botón limpiar de la aplicación Android



Botón búsqueda nos mandara a una nueva pantalla en la cual podemos visualizar un cuadro de texto y un botón búsqueda.

Figura 15

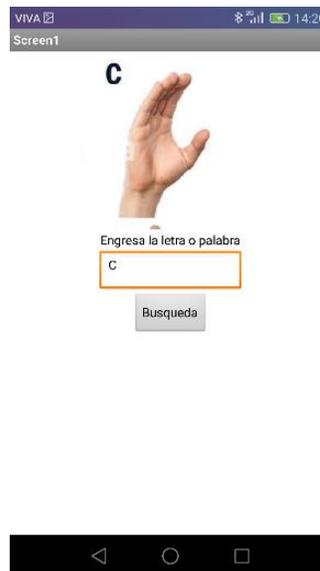
Pantalla de búsqueda



Con el botón buscar podemos buscar letras o palabras como por ejemplo “C” el cual al buscar nos mostrara dicha letra en formato de imagen.

Figura 16

Imagen de la letra C en lenguaje dactilológico



Al introducir una palabra y dar en el botón búsqueda nos mandara a otra pantalla en la cual podremos reproducir la palabra buscada con el botón play de la nueva pantalla.

Figura 17

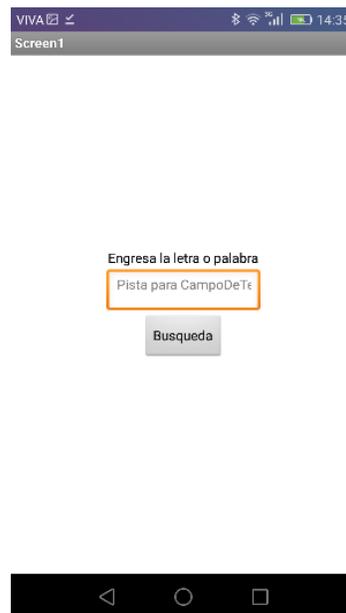
Pantalla de reproducción de video



Con el botón retroceder volveremos a la pantalla de búsqueda de la aplicación.

Figura 18

Botón de retroceder



Para el guante sensorizado

Nos colocaremos el guante en la mano derecha

Figura 19

Uso del guante con la mano derecha extendida



Para poder enviar alguna palabra se realizara de letra a letra como por ejemplo la palabra “HOLA” , primero debemos de enviar la letra “H” y guardar el dato con el botón que se encuentra en la parte de alimentación del Arduino

Figura 20

Posicionado de los dedos para la letra H



Figura 21

Botón de envío de palabras



Para enviar la letra “O” ubicaremos nuestros dedos en forma de “O” como se muestra en la imagen según el alfabeto dactilológico, luego presionaremos la el botón que se encuentra en la parte de alimentación del Arduino.

Figura 22

Posicionado de dedos para la letra O



Para enviar la letra “L” ubicaremos nuestros dedos como se muestra en la imagen según el alfabeto dactilológico, luego presionaremos el botón que se encuentra en la parte de alimentación del Arduino.

Figura 23

Posicionado de dedos para la letra L.



Para enviar la letra “A” ubicaremos nuestros dedos como se muestra en la imagen según el alfabeto dactilológico, luego presionaremos el botón que se encuentra en la

parte de alimentación del Arduino.

Figura 24

Posicionado de dedos para la letra A.



Para enviar la palabra a la aplicación lo primero es posicionar los dedos de forma extendida como se observa en la imagen luego presionar el botón que se encuentra en la parte de alimentación del Arduino.

Figura 25

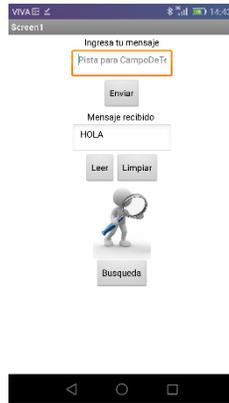
Posición de inicio para el guante



Visualización de la palabra “HOLA” en la aplicación y para su lectura presionaremos el botón leer como se observa en la imagen.

Figura 26

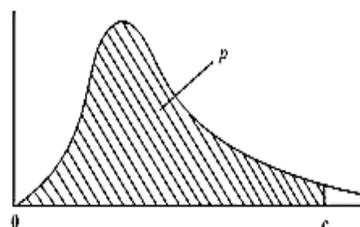
Lectura de texto recibido en la aplicación



Anexo B

Tabla de Distribución

$$p = P(X \leq c)$$



p	0,005	0,01	0,025	0,05	0,1	0,9	0,95	0,975	0,99	0,995
$\nu=1$	0,00004	0,0002	0,001	0,004	0,016	2,706	3,841	5,024	6,635	7,879
2	0,010	0,020	0,051	0,103	0,211	4,605	5,991	7,378	9,210	10,597
3	0,072	0,115	0,216	0,352	0,584	6,251	7,815	9,348	11,345	12,838
4	0,207	0,297	0,484	0,711	1,064	7,779	9,488	11,143	13,277	14,860
5	0,412	0,554	0,831	1,145	1,610	9,236	11,070	12,833	15,086	16,750
6	0,676	0,872	1,237	1,635	2,204	10,645	12,592	14,449	16,812	18,548
7	0,989	1,239	1,690	2,167	2,833	12,017	14,067	16,013	18,475	20,278
8	1,344	1,646	2,180	2,733	3,490	13,362	15,507	17,535	20,090	21,955
9	1,735	2,088	2,700	3,325	4,168	14,684	16,919	19,023	21,666	23,589
10	2,156	2,558	3,247	3,940	4,865	15,987	18,307	20,483	23,209	25,188
11	2,603	3,053	3,816	4,575	5,578	17,275	19,675	21,920	24,725	26,757
12	3,074	3,571	4,404	5,226	6,304	18,549	21,026	23,337	26,217	28,300
13	3,565	4,107	5,009	5,892	7,042	19,812	22,362	24,736	27,688	29,819
14	4,075	4,660	5,629	6,571	7,790	21,064	23,685	26,119	29,141	31,319

DOCUMENTACIÓN