



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E**  
**INDUSTRIAL**  
**CARRERA DE TELECOMUNICACIONES**

**Tema:**

---

**SISTEMA AUTOMÁTICO DE DECODIFICACIÓN DE LENGUA DE SEÑAS**  
**ECUATORIANO A COMUNICACIÓN VERBAL.**

---

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniera en Telecomunicaciones.

**ÁREA:** Comunicaciones

**LÍNEA DE INVESTIGACIÓN:** Programación y Redes

**AUTOR:** Wendy Patricia Gavilanes Proaño

**TUTOR:** Ing. Geovanni Danilo Brito Moncayo

**Ambato – Ecuador**

**agosto – 2023**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del trabajo de titulación con el tema: SISTEMA AUTOMÁTICO DE DECODIFICACIÓN DE LENGUA DE SEÑAS ECUATORIANO A COMUNICACIÓN VERBAL, desarrollado bajo la modalidad Proyecto de Investigación por la señorita Wendy Patricia Gavilanes Proaño, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2023.

-----  
Ing. Geovanni Danilo Brito Moncayo Mg.

TUTOR

## AUTORÍA

El presente trabajo de titulación titulado: SISTEMA AUTOMÁTICO DE DECODIFICACIÓN DE LENGUA DE SEÑAS ECUATORIANO A COMUNICACIÓN VERBAL es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2023.



---

Wendy Patricia Gavilanes Proaño

C.C. 1803897980

AUTORA

## DERECHOS DE AUTOR

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

Ambato, agosto 2023.



---

Wendy Patricia Gavilanes Proaño

C.C. 1803897980

AUTORA

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del informe final del trabajo de titulación presentado por la señorita Wendy Patricia Gavilanes Proaño, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA AUTOMÁTICO DE DECODIFICACIÓN DE LENGUA DE SEÑAS ECUATORIANO A COMUNICACIÓN VERBAL, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2023.

-----  
Ing. Elsa Pilar Urrutia Urrutia, Mg.

**PRESIDENTE DEL TRIBUNAL**

-----  
Ing. Franklin Wilfrido Salazar Logroño, Mg.

**PROFESOR CALIFICADOR**

-----  
Ing. Hernán Fabricio Naranjo Avalos, Mg.

**PROFESOR CALIFICADOR**

## **DEDICATORIA**

*Quiero dedicar este proyecto a mi familia que ha sido mi apoyo incondicional durante todo el proceso. Su amor, paciencia y apoyo constante han sido fundamentales para alcanzar este logro.*

*A mi mamita Mónica por su ejemplo de perseverancia y sacrificio, y a mi hermano Andrés por su aliento y motivación constante. También quiero dedicar este trabajo a mi amiga del alma Evelyn, quien me han brindado su amistad y aliento en cada paso del camino.*

*A mi abuelita Eloísa, que a pesar de que ya no este con nosotros, su influencia y recuerdo fue la inspiración principal para este proyecto.*

**Wendy Patricia Gavilanes Proaño**

## **AGRADECIMIENTO**

*Quiero agradecer principalmente a mi mamá y hermano, por su apoyo y aliento en todo este proceso, agradecerles infinitamente por la paciencia que me han tenido y por estar siempre a mi lado en el desarrollo de este proyecto.*

*A mi tutor, el Ing. Geovanni Brito, por su guía, paciencia y apoyo brindado en el desarrollo de mi proyecto.*

*De manera especial agradezco a mi mejor amiga Evelyn por siempre aconsejarme y apoyarme en cada etapa de mi vida. Y de manera general a todos mis amigos que formaron parte de mi vida universitaria les agradezco infinitamente, por el apoyo brindado en cada semestre, junto con las risas y tristezas, que me permitieron vivir la mejor etapa de mi vida y a crecer como persona.*

**Wendy Patricia Gavilanes Proaño**

## ÍNDICE GENERAL DE CONTENIDOS

|   |             |
|---|-------------|
| <b>PORTADA</b> .....  | <b>i</b>    |
| <b>APROBACIÓN DEL TUTOR</b> .....   | <b>ii</b>   |
| <b>AUTORÍA</b> .....  | <b>iii</b>  |
| <b>DERECHOS DE AUTOR</b> .....  | <b>iv</b>   |
| <b>APROBACIÓN TRIBUNAL DE GRADO</b> .....   | <b>v</b>    |
| <b>DEDICATORIA</b> .....  | <b>vi</b>   |
| <b>AGRADECIMIENTO</b> .....   | <b>vii</b>  |
| <b>ÍNDICE GENERAL DE CONTENIDOS</b> .....   | <b>viii</b> |
| <b>ÍNDICE DE TABLAS</b> .....   | <b>xii</b>  |
| <b>ÍNDICE DE FIGURAS</b> .....  | <b>xiii</b> |
| <b>RESUMEN EJECUTIVO</b> .....  | <b>xv</b>   |
| <b>ABSTRACT</b> .....   | <b>xvi</b>  |
| <b>CAPÍTULO I</b> .....   | <b>1</b>    |
| <b>MARCO TEÓRICO</b> .....  | <b>1</b>    |
| 1.1 Tema de investigación.....  | 1           |
| 1.2 Antecedentes Investigativos .....   | 1           |
| 1.3 Contextualización del problema.....   | 3           |
| 1.4 Fundamentación teórica .....  | 4           |
| 1.4.1 La audición y el habla .....  | 4           |
| 1.4.1.1 Funcionamiento de un oído sano .....  | 5           |
| 1.4.1.2 Funcionamiento de un oído afectado por pérdida auditiva.....  | 5           |
| 1.4.1.3 Consecuencias de la pérdida auditiva .....  | 6           |
| 1.4.2 Discapacidad auditiva y del habla.....  | 7           |
| 1.4.2.1 Grados de pérdida auditiva .....  | 8           |
| 1.4.3 Herramientas de comunicación para asistencia de personas que viven<br>en situación de discapacidad auditiva y del habla ..... | 9           |
| 1.4.4 Lengua de señas ecuatoriana.....  | 9           |
| 1.4.4.1 Diccionario de lengua de señas ecuatoriano “Gabriel Román” ....   | 10          |
| 1.4.5 Placas de desarrollo electrónico.....   | 12          |
| 1.4.6 Sensores flexibles.....   | 13          |



|                                      |  |           |
|--------------------------------------|--|-----------|
| 1.4.7                                | Unidad de Medición Inercial (IMU) .....                          | 14        |
| 1.4.8                                | Inteligencia Artificial (IA) .....                               | 15        |
| 1.4.9                                | Aprendizaje automático .....                                     | 16        |
| 1.4.9.1                              | Aprendizaje Supervisado .....                                    | 16        |
| 1.4.9.2                              | Aprendizaje no supervisado.....                                  | 17        |
| 1.4.9.3                              | Aprendizaje reforzado.....                                       | 18        |
| 1.4.9.4                              | Aprendizaje profundo .....                                       | 18        |
| 1.4.10                               | Funcionamiento del modelo k-Nearest Neighbors (KNN) .....        | 18        |
| 1.4.11                               | Funcionamiento del modelo Random Forest .....                    | 20        |
| 1.4.12                               | Lenguajes de programación .....                                  | 20        |
| 1.5                                  | Objetivos .....  | 21        |
| 1.5.1                                | Objetivo general .....   | 21        |
| 1.5.2                                | Objetivos específicos .....                                      | 21        |
| <b>CAPÍTULO II .....</b>             |  | <b>22</b> |
| <b>METODOLOGÍA .....</b>             |  | <b>22</b> |
| 2.1                                  | Materiales .....   | 22        |
| 2.2                                  | Métodos .....  | 22        |
| 2.2.1                                | Modalidad de la Investigación .....                              | 22        |
| 2.2.2                                | Recolección de Información .....                                 | 23        |
| 2.2.3                                | Procesamiento y Análisis de Datos .....                          | 23        |
| 2.2.4                                | Propuesta de Solución .....                                      | 23        |
| 2.2.5                                | Desarrollo del Proyecto.....                                     | 24        |
| <b>CAPITULO III.....</b>             |  | <b>26</b> |
| <b>RESULTADOS Y DISCUSIONES.....</b> |  | <b>26</b> |
| 3.1                                  | Análisis y discusión de los resultados .....                     | 26        |
| 3.2                                  | Desarrollo de la propuesta .....                                 | 26        |
| 3.2.1                                | Consideraciones previas para el desarrollo del sistema .....     | 26        |
| 3.2.1.1                              | Usuarios del sistema automático.....                             | 27        |
| 3.2.1.2                              | Palabras, símbolos y signos más empleados por la comunidad sorda | 28        |
| 3.2.1.3                              | Gestos estáticos .....   | 28        |
| 3.2.1.4                              | Gestos dinámicos .....   | 29        |
| 3.2.1.5                              | Análisis de los movimientos de las manos y los dedos.....        | 31        |
| 3.2.2                                | Esquema general del proyecto .....                               | 33        |

|  |  |           |
|--|--|-----------|
| 3.2.3                                      | Análisis y selección de los componentes electrónicos.....              | 34        |
| 3.2.3.1                                    | Selección del sensor flexible.....                                     | 34        |
| 3.2.3.2                                    | Selección del sensor inercial.....                                     | 35        |
| 3.2.3.3                                    | Selección de la placa de desarrollo.....                               | 36        |
| 3.2.4                                      | Análisis y selección del software del sistema.....                     | 37        |
| 3.2.5                                      | Consideraciones previas para el desarrollo del guante electrónico..... | 38        |
| 3.2.5.1                                    | Funcionamiento del sensor Flex 2,2”.....                               | 38        |
| 3.2.5.2                                    | Funcionamiento del Sensor MPU 6050.....                                | 42        |
| 3.2.6                                      | Diagrama del sistema electrónico del guante.....                       | 46        |
| 3.2.6.1                                    | Construcción del guante electrónico.....                               | 47        |
| 3.2.7                                      | Desarrollo del sistema.....  | 48        |
| 3.2.8                                      | Configuración inicial del sistema.....                                 | 48        |
| 3.2.8.1                                    | Selección del algoritmo de aprendizaje automático.....                 | 49        |
| 3.2.9                                      | Preparación de datos.....  | 51        |
| 3.2.9.1                                    | Adquisición de señales.....  | 51        |
| 3.2.9.2                                    | Limpieza de datos.....   | 53        |
| 3.2.10                                     | Modelado.....  | 55        |
| 3.2.10.1                                   | Análisis y selección del ajuste de hiperparámetro.....                 | 55        |
| 3.2.10.2                                   | Entrenamiento de los modelos.....                                      | 57        |
| 3.2.10.3                                   | Entrenamiento del modelo k-Nearest Neighbors (KNN).....                | 58        |
| 3.2.10.4                                   | Entrenamiento del modelo Random Forest.....                            | 60        |
| 3.2.11                                     | Despliegue de la interfaz de usuario.....                              | 63        |
| 3.2.11.1                                   | Creación página de inicio – Menú.....                                  | 64        |
| 3.2.11.2                                   | Creación de la página Hoja de datos.....                               | 65        |
| 3.2.11.3                                   | Creación de la página K-Nearest Neighbors.....                         | 67        |
| 3.2.11.4                                   | Creación de la página Random Forest.....                               | 68        |
| 3.2.11.5                                   | Diagrama de flujo del proceso de la interfaz web.....                  | 70        |
| 3.2.12                                     | Manual de usuario.....   | 73        |
| 3.2.13                                     | Pruebas de funcionamiento y análisis de resultados.....                | 73        |
| 3.2.14                                     | Presupuesto.....   | 81        |
| <b>CAPÍTULO IV.....</b>                    |  | <b>82</b> |
| <b>CONCLUSIONES Y RECOMENDACIONES.....</b> |  | <b>82</b> |
| 4.1  | Conclusiones.....  | 82        |

|   |                       |           |
|---|-----------------------|-----------|
| 4.2   | Recomendaciones ..... | 83        |
| <b>BIBLIOGRAFÍA</b> .....   |                       | <b>85</b> |
| <b>ANEXOS</b> .....   |                       | <b>89</b> |
| Anexo 1: Certificado de funcionamiento .....                            |                       | 90        |
| Anexo 2: Hoja de datos técnicos sensor MPU-6050 .....                   |                       | 91        |
| Anexo 3: Hoja de datos técnicos del sensor Flex 2.2” .....              |                       | 96        |
| Anexo 4: Hoja de datos técnicos Placa Arduino Nano.....                 |                       | 98        |
| Anexo 5: Código Arduino NANO.....                                       |                       | 101       |
| Anexo 6: Código Python – Main de Streamlit .....                        |                       | 102       |
| Anexo 7: Código de Python – Hoja de datos de Streamlit.....             |                       | 103       |
| Anexo 8: Código Python – K-NN de Streamlit.....                         |                       | 109       |
| Anexo 9: Código Python – Random Forest de Streamlit .....               |                       | 113       |
| Anexo 10: Código Python – Lengua de Señas Ecuatoriana de Streamlit..... |                       | 116       |
| Anexo 11: Análisis similitudes físicas .....                            |                       | 120       |
| Anexo 12: Pruebas de funcionamiento.....                                |                       | 124       |
| Anexo 13: Manual de usuario.....  |                       | 125       |

## ÍNDICE DE TABLAS

|   |    |
|---|----|
| <b>Tabla 1</b> Consecuencias de la pérdida auditiva.....                              | 7  |
| <b>Tabla 2</b> Grados de pérdida auditiva.....  | 8  |
| <b>Tabla 3</b> Algoritmos de aprendizaje supervisado .....                            | 17 |
| <b>Tabla 4</b> Actividades para el Desarrollo del Proyecto (1/2) .....                | 24 |
| <b>Tabla 5</b> Actividades para el Desarrollo del Proyecto (2/2) .....                | 25 |
| <b>Tabla 6</b> Gestos estáticos alfabeto dactilológico .....                          | 28 |
| <b>Tabla 7</b> Gestos dinámicos para el sistema 1/2 .....                             | 29 |
| <b>Tabla 8</b> Gestos dinámicos para el sistema 2/2 .....                             | 30 |
| <b>Tabla 9</b> Movimientos de la mano.....  | 32 |
| <b>Tabla 10</b> Comparación sensores de flexión.....                                  | 34 |
| <b>Tabla 11</b> Comparación sensores gestuales .....                                  | 35 |
| <b>Tabla 12</b> Comparación placas de desarrollo .....                                | 36 |
| <b>Tabla 13</b> Selección del Framework para la interfaz .....                        | 37 |
| <b>Tabla 14</b> Mediciones de mínima y máxima flexión.....                            | 42 |
| <b>Tabla 15</b> Ángulos mínimos y máximos del sensor MPU-6050 .....                   | 46 |
| <b>Tabla 16</b> Asignación de pines del sistema electrónico del guante .....          | 46 |
| <b>Tabla 17</b> Selección del algoritmo de aprendizaje automático [20] .....          | 49 |
| <b>Tabla 18</b> Librerías empleadas para el sistema.....                              | 50 |
| <b>Tabla 19</b> Conjunto de datos de los gestos estáticos .....                       | 51 |
| <b>Tabla 20</b> Conjunto de datos de los gestos dinámicos.....                        | 52 |
| <b>Tabla 21</b> Conjunto de datos de los números.....                                 | 53 |
| <b>Tabla 22</b> Valores mínimos y máximos de los sensores Flex y MPU6050 .....        | 54 |
| <b>Tabla 23</b> Análisis letras CH, K, P .....  | 54 |
| <b>Tabla 24</b> Características específicas de los gestos estáticos y dinámicos ..... | 55 |
| <b>Tabla 25</b> Técnicas para el ajuste de hiperparámetros [20] .....                 | 56 |
| <b>Tabla 26</b> Evaluación de los algoritmos K-NN y Random Forest .....               | 62 |
| <b>Tabla 27</b> Porcentajes de aciertos para los gestos estáticos .....               | 74 |
| <b>Tabla 28</b> Porcentajes de aciertos para los gestos dinámicos.....                | 76 |
| <b>Tabla 29</b> Porcentajes de aciertos para los números .....                        | 78 |
| <b>Tabla 30</b> Resultado general del sistema .....                                   | 80 |
| <b>Tabla 31</b> Presupuesto de construcción.....                                      | 81 |

## ÍNDICE DE FIGURAS

|   |    |
|---|----|
| <i>Figura 1 Partes del oído [10]</i> .....  | 5  |
| <i>Figura 2 Funcionamiento de un oído sano [10]</i> .....                                       | 5  |
| <i>Figura 3 Funcionamiento del oído causada por una pérdida de audición [10]</i> .....          | 6  |
| <i>Figura 4 Alfabeto Dactilológico Ecuatoriano [13]</i> .....                                   | 10 |
| <i>Figura 5 Puntos de referencia de las manos para la articulación de la seña [14]</i> ..       | 11 |
| <i>Figura 6 Trayectoria y dirección de los movimientos de las manos[1/2] [14]</i> .....         | 11 |
| <i>Figura 7 Trayectoria y dirección de los movimientos de las manos [2/2] [14]</i> .....        | 12 |
| <i>Figura 8 Orientación de ejes de sensibilidad [18]</i> .....                                  | 14 |
| <i>Figura 9 Sistema de coordenadas [18]</i> .....   | 15 |
| <i>Figura 10 Técnicas de la Inteligencia Artificial [19]</i> .....                              | 15 |
| <i>Figura 11 Ejemplo del clasificador K-NN [21]</i> .....                                       | 19 |
| <i>Figura 12 Composición del modelo Random Forest [22]</i> .....                                | 20 |
| <i>Figura 13 No. Registros de personas con discapacidad auditiva en Tungurahua [8]</i><br>..... | 27 |
| <i>Figura 14 Números en lengua de señas ecuatoriano [13]</i> .....                              | 28 |
| <i>Figura 15 Gestos dinámicos del alfabeto dactilológico ecuatoriano [13]</i> .....             | 29 |
| <i>Figura 16 Anatomía de la mano [24]</i> .....   | 31 |
| <i>Figura 17 Esquema general del proyecto</i> .....   | 33 |
| <i>Figura 18 Funcionamiento del sensor Flex [17]</i> .....                                      | 39 |
| <i>Figura 19 Circuito divisor de voltaje [17]</i> .....   | 39 |
| <i>Figura 20 Diagrama de conexión de los sensores Flex al Arduino Nano</i> .....                | 40 |
| <i>Figura 21 Diagrama de flujo de los sensores Flex</i> .....                                   | 41 |
| <i>Figura 22 Movimiento de los ejes del Sensor MPU-6050 [32]</i> .....                          | 43 |
| <i>Figura 23 Diagrama de conexión del Sensor MPU-6050 al Arduino Nano</i> .....                 | 44 |
| <i>Figura 24 Diagrama de flujo del sensor MPU-6050</i> .....                                    | 44 |
| <i>Figura 25 Diagrama de conexiones del guante electrónico</i> .....                            | 46 |
| <i>Figura 26 Diagrama esquemático del guante electrónico</i> .....                              | 47 |
| <i>Figura 27 Prototipo guante electrónico</i> .....   | 48 |
| <i>Figura 28 Proceso de entrenamiento del aprendizaje automático</i> .....                      | 48 |
| <i>Figura 29 Librerías empleadas para el sistema</i> .....                                      | 50 |
| <i>Figura 30 División de los datos de entrenamiento y prueba</i> .....                          | 57 |

|                  |  |    |
|------------------|--|----|
| <b>Figura 31</b> | <i>Modelo del clasificador K-Nearest Neighbors</i> .....                       | 58 |
| <b>Figura 32</b> | <i>Predicción del modelo KNN</i> .....   | 58 |
| <b>Figura 33</b> | <i>Métricas de precisión del modelo K-NN</i> .....                             | 59 |
| <b>Figura 34</b> | <i>Modelo del clasificador Random Forest</i> .....                             | 60 |
| <b>Figura 35</b> | <i>Predicciones del modelo Random Forest</i> .....                             | 60 |
| <b>Figura 36</b> | <i>Evaluación de rendimiento del modelo Random Forest</i> .....                | 61 |
| <b>Figura 37</b> | <i>Métricas de precisión del modelo Random Forest</i> .....                    | 61 |
| <b>Figura 38</b> | <i>Instalación del Framework Streamlit</i> .....                               | 63 |
| <b>Figura 39</b> | <i>Ejecución de Streamlit</i> .....  | 63 |
| <b>Figura 40</b> | <i>Inicio de Streamlit</i> .....   | 64 |
| <b>Figura 41</b> | <i>Creación de las páginas del aplicativo web Streamlit</i> .....              | 64 |
| <b>Figura 42</b> | <i>Interfaz web - Menú Principal</i> .....                                     | 65 |
| <b>Figura 43</b> | <i>Lista de nombre de los archivos de imágenes</i> .....                       | 65 |
| <b>Figura 44</b> | <i>Selección para el puerto serie</i> .....                                    | 65 |
| <b>Figura 45</b> | <i>Guardar nuevos datos</i> .....  | 66 |
| <b>Figura 46</b> | <i>Interfaz web - Hoja de datos</i> .....                                      | 67 |
| <b>Figura 47</b> | <i>Interfaz web - Página K-Nearest Neighbors</i> .....                         | 67 |
| <b>Figura 48</b> | <i>Interfaz web - Página KNN - Predicción del modelo</i> .....                 | 68 |
| <b>Figura 49</b> | <i>Interfaz web - Página Random Forest</i> .....                               | 68 |
| <b>Figura 50</b> | <i>Interfaz web - Página Random Forest - Predicción del modelo</i> .....       | 69 |
| <b>Figura 51</b> | <i>Funcionamiento del modelo entrenado</i> .....                               | 69 |
| <b>Figura 52</b> | <i>Predicción del modelo</i> .....   | 70 |
| <b>Figura 53</b> | <i>Interfaz web - Página LSEC</i> .....  | 70 |
| <b>Figura 54</b> | <i>Diagrama de flujo de la Interfaz web 1/2</i> .....                          | 71 |
| <b>Figura 55</b> | <i>Diagrama de flujo de la interfaz web 2/2</i> .....                          | 72 |
| <b>Figura 56</b> | <i>Porcentaje de aciertos para la traducción de los gestos estáticos</i> ..... | 75 |
| <b>Figura 57</b> | <i>Porcentaje de aciertos para la traducción de los gestos dinámicos</i> ..... | 77 |
| <b>Figura 58</b> | <i>Porcentaje de aciertos para la traducción de los números</i> .....          | 79 |

## RESUMEN EJECUTIVO

Las personas con discapacidad auditiva y del habla dependen de la lengua de señas para expresar sus ideas, emociones y necesidades. Esta forma de comunicación gestual emplea movimientos de las manos y expresiones faciales para transmitir mensajes de manera efectiva. Sin embargo, la falta de conocimiento de este lenguaje por parte de la población oyente limita las oportunidades de interacción y comprensión mutua. Por este motivo, se desarrolló un sistema automático capaz de traducir la lengua de señas ecuatoriana a comunicación verbal empleando inteligencia artificial.

El proyecto se desarrolló en base al diccionario de lengua de señas ecuatoriano “Gabriel Román” para establecer el vocabulario básico del sistema, el cual consta del alfabeto dactilológico ecuatoriano, números y palabras básicas. Se construyó un guante electrónico para la mano derecha, con cinco sensores Flex, un sensor inercial MPU6050 y una placa Arduino Nano, los cuales se encargaron de registrar los movimientos de la mano y los dedos, como lo son flexión, extensión e inclinación, estos datos son procesados mediante Python para crear la base de datos, con la finalidad de aplicar algoritmos de aprendizaje automático. Los algoritmos estudiados son k-Nearest Neighbors (KNN) y Random Forest, con precisiones de entrenamiento del 94,62% y 96,20% respectivamente. Finalmente, aplicando el modelo Random Forest se creó una interfaz gráfica en Streamlit, donde el usuario puede visualizar la traducción de la lengua de señas mediante texto y una salida de audio.

**Palabras clave:** Guante electrónico, aprendizaje automático, discapacidad auditiva, lengua de señas ecuatoriana.

## ABSTRACT

People with hearing and speech disabilities rely on sign language to express their ideas, emotions, and needs. This form of gestural communication employs hand movements and facial expressions to effectively convey messages. However, the lack of knowledge of this language on the part of the hearing population limits the opportunities for interaction and mutual understanding. For this reason, an automatic system capable of translating and learning the Ecuadorian sign language into verbal communication using artificial intelligence was developed.

The project was developed based on the "Gabriel Román" Ecuadorian Sign Language Dictionary " to establish the basic vocabulary of the system, which consists of the Ecuadorian dactylological alphabet, numbers and basic words. An electronic glove for the right hand was built, with five Flex sensors, an MPU6050 inertial sensor and an Arduino Nano board, which were responsible for recording the movements of the hand and fingers, such as flexion, extension and inclination, these Data is processed using Python to create the database, in order to apply machine learning algorithms. The algorithms studied are k-Nearest Neighbors (KNN) and Random Forest, with training accuracies of 94,62% and 96,20% respectively. Finally, applying the Random Forest model, a graphical interface was created in Streamlit, where the user can visualize the translation of the sign language through text and an audio output.

**Keywords:** Electronic glove, machine learning, hearing impairment, ecuadorian sign language.



# CAPÍTULO I

## MARCO TEÓRICO

### 1.1 Tema de investigación

Sistema automático de decodificación de lengua de señas ecuatoriano a comunicación verbal.

### 1.2 Antecedentes Investigativos

Al finalizar la investigación bibliográfica basados en artículos científicos y en los principales repositorios de las universidades se pueden encontrar los siguientes antecedentes:

Diego Alejandro Guzmán Arellano en el año 2017 desarrollo la investigación con el tema “Guante Electrónico para Traducir de Lenguaje de Señas a Caracteres con Voz Artificial y Conexión Inalámbrica a Dispositivos Móviles para Personas con Discapacidad Auditiva y de Lenguaje en la Universidad Técnica de Ambato”, se enfoca en el desarrollo de un guante electrónico empleado para personas con discapacidades auditivas, mediante el uso de sensores flexibles y un Acelerómetro MMA7361 que juntamente con una tarjeta Arduino Mega 2560 y un módulo Text to Speech “EMIC2” permiten capturar los movimientos de mano y a través de un módulo Bluetooth HC-05 se transmitirá la información mostrándose en una pantalla Shield TFT de 2.8 pulgadas o a través de un Smartphone, como resultado la programación en Arduino del guante electrónico presenta una precisión del 90% del alfabeto dactilológico. [1]

Mateo Alejandro Parreño Alvarez en el año 2018 desarrollo la investigación titulada “Desarrollo de un juego didáctico mediante sensores Leap Motion para estimular el aprendizaje del lenguaje básico de señas ecuatoriano”. Se basa en el desarrollo de un juego didáctico para la enseñanza-aprendizaje del lenguaje básico de señas ecuatoriano, mediante un sensor Leap Motion y el software Unity 3D, los cuales estarán vinculados de manera bidireccional con el software MatLab realizando el algoritmo Dynamic Time Warping (DTW), para el reconocimiento de los gestos de las manos y capturar los diferentes movimientos de los dedos y las articulaciones, como resultado el sistema tienen una precisión de clasificación con señas estáticas del 86% y en señas de números del 89%. [2]

Jacob Schioppo, Zachary Meyer, Diego Fabiano y Shaun Canavan en el año 2019 desarrollaron la investigación titulada “Learning Sign Language in a Virtual Environment”. Se enfoca en crear un entorno virtualizado para el aprendizaje de lengua de señas empleando un controlador Leap Motion para capturar los movimientos de las manos y entrenando los datos a través de Machine Learning, aplicando dos métodos, Bosques Aleatorios (Random Forest) y Redes Neuronales de Retroalimentación Profunda (Deep Feedforward Network - DFNN) a través de visión en primera persona para reconocer una amplia gama de gestos, como resultado se obtuvo un entorno virtual VR para el alfabeto de 26 letras del lenguaje de señas estadounidense donde los algoritmos Random Forest y DFNN presentaron precisiones del 96.6% y 97.1% respectivamente. [3]

Byron Fernando Garzón Tipán en el año 2020 desarrollo la investigación con el tema “Sistema de traducción de lenguaje de señas a lenguaje oral y escrito mediante un guante electrónico”, se enfoca en la creación de un guante capaz de traducir el abecedario dactilológico y los números a lenguaje oral y escrito mediante sensores Flex y micro pulsadores los cuales estarán conectados a una tarjeta Arduino nano para el procesamiento de los datos, y a través del módulo Bluetooth HC-05 se enviara a una aplicación móvil desarrollada en APP Inventor 2. Como resultado se obtuvo la traducción treinta y tres caracteres entre números y letras del abecedario dactilológico ecuatoriano con una precisión bastante alta. [4]

Diego Andrés López Albán y Manuel Alejandro López Barrera en el año 2021 desarrollo la investigación con el tema “Sistema de reconocimiento automático de lengua de señas colombiana mediante dispositivo óptico de captura”, el cual, describe el desarrollo de un sistema que detecta los gestos de las manos realizado por el usuario mediante un sensor óptico Leap Motion Controller junto con un entorno virtual creado en la plataforma Unity 3D, para procesar la información en base a un Dataset de Lengua de Señas Colombiano (LSC), donde posteriormente se aplicara aprendizaje automático de tres algoritmos: LGBM, Ova Perceptron y SDCA para finalmente mostrar las respectivas traducciones; como resultado se obtuvo que el algoritmo LGBM fue el modelo con mayor precisión con un porcentaje del 92.86% al reconocer sesenta gestos del LSC. [5]

### **1.3 Contextualización del problema**

La Ley define la discapacidad auditiva como cualquier disminución de la audición, ya sea fluctuante o permanente, que interfiere con la capacidad de la persona para desempeñarse en su entorno, debido a que tienen problemas para entablar una comunicación oral con otros individuos, además de existir varios tipos de sordera, como puede ser total, parcial o congénita, este último se presenta al momento del nacimiento, impidiendo también el desarrollo del habla.

A nivel mundial según la Federación Mundial de Sordos (DMA) [6], existen aproximadamente 70 millones de personas sordas en todo el mundo. Más del 80% vive en países en desarrollo y como colectivo, utilizan más de 300 diferentes lenguas de señas. Por otra parte, según la Organización Mundial de la Salud (OMS) [7] estima que América Latina y el Caribe representan el 9% de la carga de la pérdida de la agudeza auditiva en el mundo, lo que lleva a que puedan comunicarse únicamente a través de lengua de señas.

En el Ecuador en el año 2022 existen 66.538 personas con discapacidad auditiva, según datos del Consejo Nacional para la Igualdad de Discapacidades (CONADIS), el cual representa el 37,99% de la población, siendo las provincias con mayor discapacidad auditiva el Guayas con 13.933 personas y Pichincha con 12.396 personas en el rango de todas las edades [8]. Teniendo actualmente un crecimiento de 15958 personas comparado al último censo realizado en el 2015, el cual registro 50580 personas con discapacidad auditiva y 5562 con discapacidad de lenguaje. [9]

En la provincia de Tungurahua se registran 3.418 personas con discapacidad auditiva y del habla según la CONADIS, los cuales cuentan con el apoyo por parte de la Asociación de Sordos de Tungurahua (ASTU), el cual ofrece cursos de Lengua de Señas Ecuatoriana, sin embargo existe falta de personal y herramientas tecnológicas que faciliten la enseñanza de la lengua de señas, presentándose un mayor problemas en jóvenes debido a que podrían sentirse excluidos de interacciones sociales en escuelas convencionales o rezagarse en el aprendizaje del lenguaje de señas local.

Este proyecto evidencia la necesidad de crear un sistema automático de decodificación de lengua de señas ecuatoriano a comunicación escrita u oral, debido a que la forma tradicional de aprenderlo y traducirlo es con la ayuda de intérpretes o personas que

conozcan del tema, sin embargo no siempre está al alcance de todos, por lo tanto la creación del sistema permite incentivar a las personas sordas y oyentes a involucrarse en el aprendizaje de esta lengua, reduciendo barreras de comunicación, mejorando la inclusión de las personas con discapacidad auditiva en ámbitos educativos, laborales y sociales.

## **1.4 Fundamentación teórica**

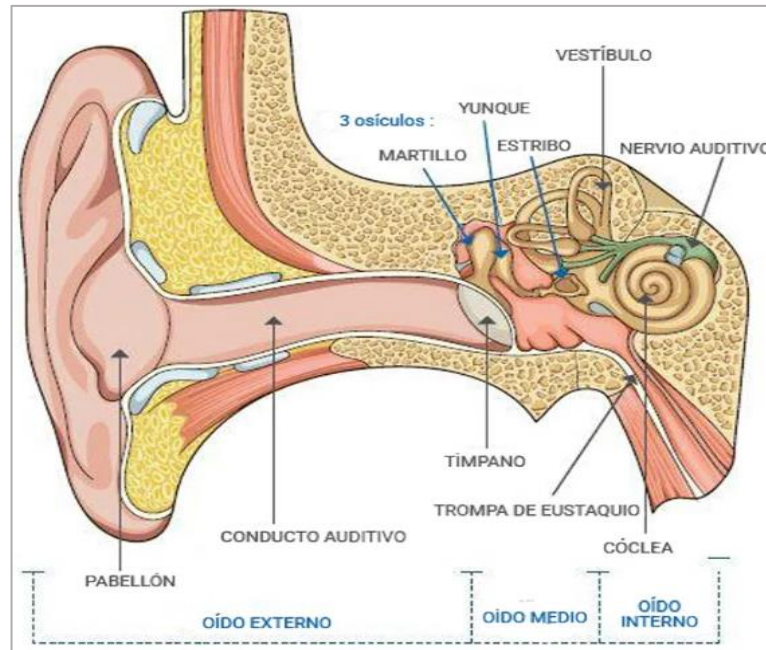
### **1.4.1 La audición y el habla**

El habla se define como la capacidad de expresar pensamientos y sentimientos mediante sonidos articulados, los cuales son captados por el oído, es así como el habla está íntimamente relacionado y coordinado con el sistema auditivo. [10]

El oído humano posee un campo auditivo de 130 dB, es un aparato muy pequeño que capta ondas de sonido y las convierte en señales eléctricas que viajan a través de los nervios hasta el cerebro. Para ello, el oído funciona como amplificador, filtro, atenuador, frecuencímetro y a la vez como sistema de comunicación multicanal del sistema sonoro, de esta forma el cerebro interpreta las señales y las conecta con otras impresiones y experiencias, como sonidos altos o bajos. El oído adulto puede reconocer y distinguir unos 400.000 sonidos diferentes relacionados con el habla, la música, las personas y los sonidos naturales, de igual modo tiene una enorme selectividad para percibir y determinar la ubicación de la fuente sonora con una precisión de aproximadamente 1 grado. [10]

El oído se compone de 3 partes:

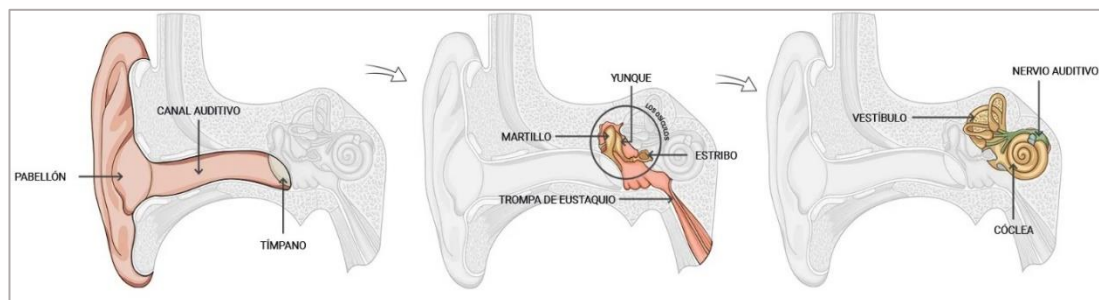
- **El oído externo:** Pertenece a la parte visible de la oreja, donde se encuentran el pabellón auditivo, conducto auditivo y al tímpano. Permite recibir los sonidos.
- **El oído medio:** Se compone del tímpano y la cavidad timpánica la cual posee 3 pequeños huesos (huesecillos) llamados martillo, estribo y yunque, los cuales transmiten las vibraciones al oído interno.
- **El oído interno:** también llamado sistema vestibular, se compone de la cóclea y el órgano del equilibrio, transforma los sonidos en señales nerviosas los cuales viajan al cerebro.



*Figura 1 Partes del oído [10]*

#### 1.4.1.1 Funcionamiento de un oído sano

El oído convierte las ondas de sonido en señales eléctricas las cuales son enviadas al cerebro. Esto ocurre cuando las ondas de sonido entran al oído y hacen vibrar la membrana timpánica (tímpano), que luego es transmitida a los huesecillos (martillo, yunque y estribo) del oído medio. Los huesecillos amplifican y transmiten las vibraciones al oído interno, donde las células ciliadas llamadas estereocilios las convierten en señales eléctricas y las envían al cerebro a través de las fibras nerviosas. [10]



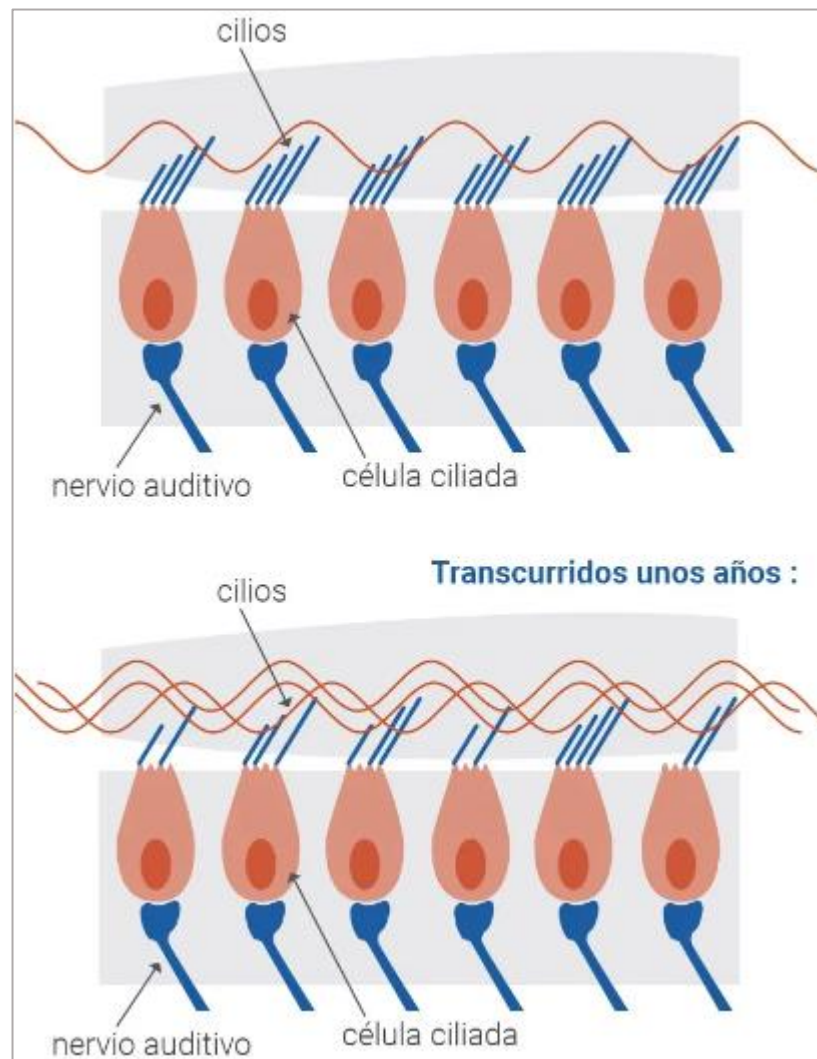
*Figura 2 Funcionamiento de un oído sano [10]*

#### 1.4.1.2 Funcionamiento de un oído afectado por pérdida auditiva

Las células ciliadas marcan la diferencia entre un oído sano y uno con pérdida auditiva, estas células son muy frágiles y pueden destruirse repentinamente o gradualmente cuando se

exponen a niveles de sonido significativos, lo cual es un problema debido a que no son renovables y no se pueden reparar ni reemplazar. [10]

Cuando la célula ciliada se daña, la transmisión de la señal al cerebro se vuelve imprecisa y la audición se vuelve difícil. Por otro lado, cuando la célula ciliada se destruye, no puede transmitir una señal al cerebro, por lo tanto, ya no se escucha. Esta pérdida de audición es irreversible.



**Figura 3** Funcionamiento del oído causada por una pérdida de audición [10]

### 1.4.1.3 Consecuencias de la pérdida auditiva

Existen diversos tipos de trastornos relacionados con una insuficiencia auditiva, entre los cuales se encuentran detallados en la siguiente tabla:

*Tabla 1 Consecuencias de la pérdida auditiva*

| <b>Tipos</b>                  | <b>Consecuencias</b>  |
|-------------------------------|---|
| <b>Problemas sociales</b>     | <b>Dificultades de comunicación:</b> la persona no entiende o malinterpreta al interlocutor y le pide que repita o hable más alto para entender.<br><b>Aislamiento:</b> una persona con discapacidad auditiva tiende a retraerse y, por lo tanto, a aislarse socialmente. Poco a poco, esta persona se autoexcluye. |
| <b>Trastornos físicos</b>     | Dolores de cabeza, hipertensión, pérdida del equilibrio, acúfenos o hiperacusia   |
| <b>Problemas psicológicos</b> | Pueden experimentar estrés al no entender el mundo que les rodea y en consecuencia se sienten inseguros, tienen baja moral debido a la falta de confianza en sí mismos, baja autoestima, vergüenza, etc.  |

**Fuente:** Investigador basado en [10]

#### **1.4.2 Discapacidad auditiva y del habla**

La discapacidad auditiva es una afección que afecta la capacidad de una persona para escuchar y entender los sonidos del entorno. Puede ser causada por problemas en el oído o por trastornos en el sistema nervioso auditivo y puede afectar una o ambas orejas. De igual manera puede ser congénita o adquirida causada por diversos factores, como enfermedades, lesiones o el envejecimiento. Además de afectar la capacidad de una persona para comunicarse y participar plenamente en la sociedad. [11]

La discapacidad del habla, también conocida como afasia, es una afección que afecta la capacidad de una persona para hablar, escribir y comprender el lenguaje hablado o escrito. Puede ser causada por lesiones en el cerebro, enfermedades o trastornos neurológicos y puede afectar la capacidad de una persona para comunicarse con los demás. [11]

Ambas discapacidades no necesariamente están relacionadas, una persona puede ser sorda pero puede desarrollar el habla, o por el contrario, las personas que pueden escuchar no pueden hablar, sin embargo se debe tener en cuenta que dichas

discapacidades pueden afectar la vida diaria de las personas que las padecen y pueden requerir el uso de dispositivos de asistencia, como audífonos o dispositivos de comunicación alternativos, o terapias especializadas para mejorar la comunicación y la capacidad de escucha o de habla. [11]

#### 1.4.2.1 Grados de pérdida auditiva

Existen diferentes grados de pérdida auditiva, en donde se pueden distinguir los diferentes niveles de pérdida medidos en decibeles, por lo tanto, se clasifica de la siguiente manera: [11]

*Tabla 2 Grados de pérdida auditiva*

| <b>Clasificación</b>                 | <b>Nivel de pérdida</b>   | <b>Definición</b>   |
|--------------------------------------|---|---|
| <b>Audición Normal o Normoacusia</b> | De 0dB a 20dB   | No presentan pérdida auditiva.  |
| <b>Templando</b>                     | De 21dB a 40dB  | Pérdida auditiva leve, es posible que escuche algunos sonidos del habla, pero tendrá dificultades con los sonidos suaves.               |
| <b>Moderado</b>                      | <b>Primer grado:</b><br>41 dB a 55dB<br><b>Segundo grado:</b><br>56dB a 70dB.   | Presenta dificultades para escuchar/comprender el habla cuando alguien está hablando a un nivel normal.                                 |
| <b>Severo</b>                        | <b>Primer grado:</b><br>71dB a 80dB<br><b>Segundo grado:</b><br>81dB a 90dB   | Escuchará poco o nada cuando se les hable a niveles normales, y solo escuchará algunos sonidos fuertes.                                 |
| <b>Profundo</b>                      | <b>Primer grado:</b><br>91dB a 100dB<br><b>Segundo grado:</b><br>101dB a 110dB<br><b>Tercer grado:</b><br>111dB a 119dB | Es posible que solo escuche sonidos muy fuertes y nada del habla, en este caso solo se usa lengua de señas y/o comunicaciones escritas. |
| <b>Cofosis</b>                       | La pérdida tonal media es de 120 dB   | Deficiencia auditiva total  |

**Fuente:** Investigador basado en [11]



### **1.4.3 Herramientas de comunicación para asistencia de personas que viven en situación de discapacidad auditiva y del habla**

Existen diversas herramientas de asistencia disponibles capaces de ofrecer diversas funciones, con el objetivo de facilitar la vida de las personas con discapacidad auditiva y del habla. Estas tecnologías incluyen audífonos, implantes cocleares, lengua de señas, dispositivos de lectura labial y dispositivos de síntesis de voz. [12]

Los audífonos son dispositivos que se colocan en los oídos para amplificar los sonidos de esta manera para mejorar la audición de las personas con discapacidad auditiva. El implante coclear, por otro lado, es un dispositivo que se implanta quirúrgicamente en el oído y convierte los sonidos en impulsos eléctricos que se envían al cerebro, lo que permite a las personas con discapacidad auditiva escuchar y entender el habla. [12]

El lenguaje de señas es un sistema de comunicación visual que utiliza gestos y movimientos de las manos y el cuerpo para expresar palabras y conceptos. Las personas con discapacidad auditiva y del habla pueden utilizar el lenguaje de señas para comunicarse con los demás. [12]

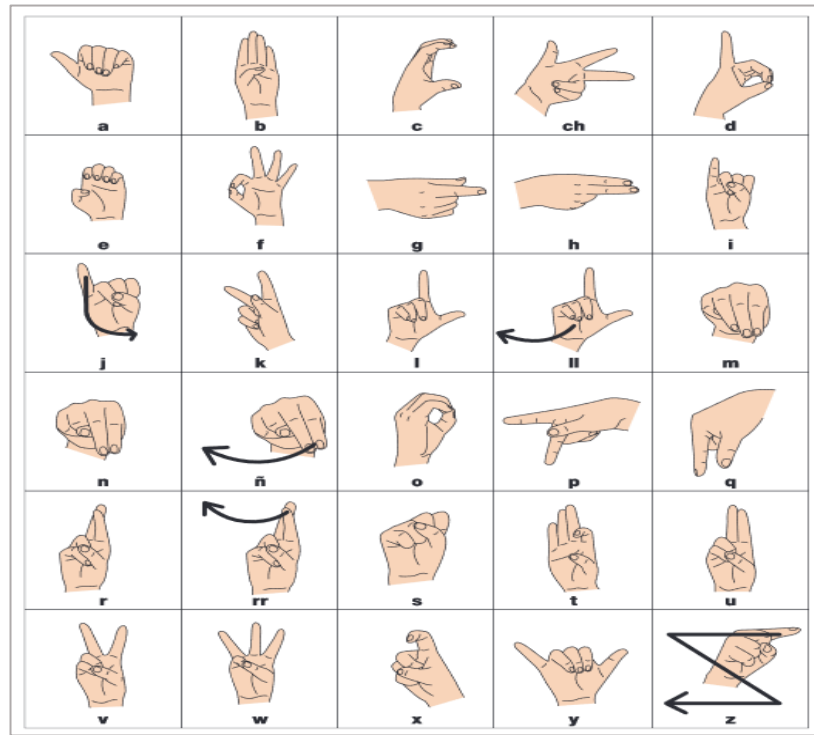
Los dispositivos de lectura labial utilizan cámaras y software para detectar y traducir los movimientos de los labios cuando alguien está hablando, permitiendo a las personas con discapacidad auditiva entender el habla. Por otro lado, los dispositivos de síntesis de voz utilizan software para sintetizar la voz y permiten a las personas con discapacidad del habla puedan hablar usando teclado o un control de voz. [12]

### **1.4.4 Lengua de señas ecuatoriana**

La Lengua de Señas Ecuatoriana es una herramienta que facilita la interacción e inclusión entre personas con o sin discapacidad auditiva, es un idioma con vocabulario, gramática y estructura propios, la cual está en constante evolución, con variaciones regionales y dialectales. Para su aprendizaje existe un diccionario creado por la CONADIS, Fenasec y la Universidad Tecnológica Indoamérica, llamado “Diccionario de lengua de señas ecuatoriano Gabriel Román” el cual hizo crecer de gran manera la inclusión a la sociedad de las personas con discapacidad auditiva. [13]

Las personas con este tipo de discapacidad emplean un alfabeto dactilológico, el cual consiste en grupos de señas realizados por las manos, capaz de representar letras del

alfabeto con el que se escribe la lengua oral del país y además de incluir la simbolización de los números tanto para una o ambas manos. Es decir, es un sistema que permite simbolizar las letras del alfabeto como se puede apreciar en la Figura 4.



*Figura 4 Alfabeto Dactilológico Ecuatoriano [13]*

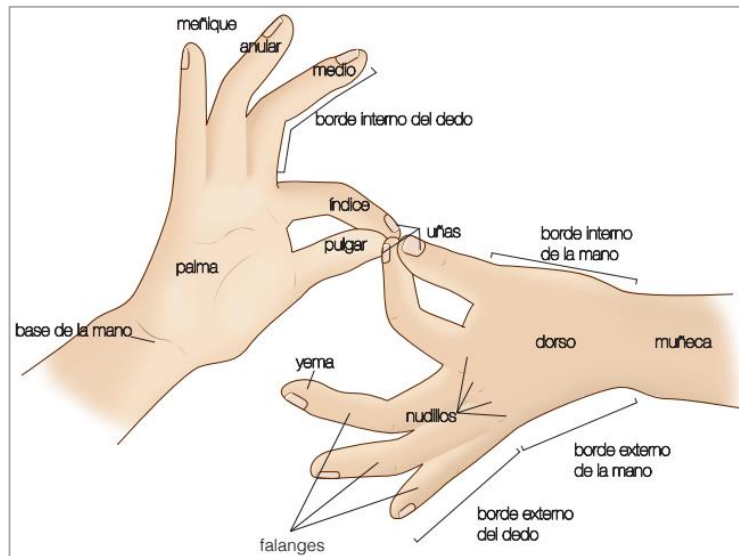
En Ecuador, se estima que hay alrededor de 66.538 personas con discapacidad auditiva, de las cuales solo el 13,54% tiene trabajo, según la CONADIS en el año 2022. Sin embargo, es probable que exista un número mayor con esta discapacidad debido a barreras comunicativas lo cual no permita registrar a todas las personas. [8]

#### **1.4.4.1 Diccionario de lengua de señas ecuatoriano “Gabriel Román”**

Es una herramienta útil para aprender la lengua de señas ecuatoriano, contiene aproximadamente 5.000 palabras tanto en su libro físico como en su plataforma web. Contiene 9 conjuntos de configuraciones de señales con 82 variaciones, donde cada una representa una pose de la mano, y su plataforma web cuenta con videos de cómo realizar cada letra, tanto con el alfabeto como en frases comunes. [14]

Por otra parte, el manual de uso digitalizado posee las indicaciones necesarias respecto a los diferentes movimientos que requieren las manos, como son: repeticiones,


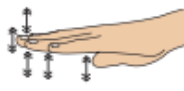
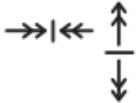
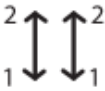
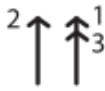










velocidad de movimiento, puntos de referencia de las manos respecto a las articulaciones de las señas, trayectoria, dirección, entre otras. Todas ellas son representadas en símbolos de movimiento como se muestran en las Figuras 5, 6 y 7. La lengua de señas ecuatoriano además de utilizar movimientos de las manos, emplea expresiones faciales y corporales para comunicar significados. [14]



**Figura 5** Puntos de referencia de las manos para la articulación de la seña [14]

|                      |                 |  |                                       |                                     |
|----------------------|-----------------|--|---------------------------------------|-------------------------------------|
| Trayectoria lineal   | <br>Línea Recta | <br>De izquierda a derecha<br>De derecha a izquierda | <br>Desde abajo hacia arriba          | <br>Desde arriba hacia abajo        |
|                      |                 | <br>Diagonal desde abajo hacia arriba                | <br>Diagonal desde arriba hacia abajo | <br>Realizar trayectoria en zigzag  |
| Trayectoria curva    | <br>Línea Curva | <br>De izquierda a derecha                           | <br>De derecha a izquierda            | <br>Desde abajo hacia arriba        |
|                      |                 | <br>Desde arriba hacia abajo                         | <br>Realizar trayectoria ondulada     | <br>Desplazamiento en curvas        |
| Trayectoria circular | <br>Círculo     | <br>De derecha a izquierda                           | <br>De izquierda a derecha            | <br>Realizar trayectoria en espiral |

**Figura 6** Trayectoria y dirección de los movimientos de las manos [1/2] [14]

|   |  |   |   |
|---|--|---|---|
| <b>MOVIMIENTO</b><br>Flechas simples que indican el movimiento de mano, dedos y boca.   | <br>Giro en su propio eje.              | <br>Movimiento alternado de los dedos.  | <br>Representa la unión de manos o dedos y abrir o cerrar la boca. |
|   | <br>Movimiento simultáneo de las manos. | <br>Movimiento alterno de las manos.    | <br>Representa el abrir o cerrar los dedos de una mano.            |
|   | <br>Expulsar aire con la boca abierta.  | <br>Expulsar aire con la boca cerrada. |   |
| <b>REPETICIÓN</b><br>Flechas que indican la cantidad de veces que se realiza un movimiento.                                       | <br>Realizar el movimiento una vez.     | <br>Realizar el movimiento dos veces.   | <br>Realizar el movimiento varias veces.                           |
| <b>VELOCIDAD</b><br>Flechas que indican la velocidad del movimiento.  | <br>Velocidad normal.                 | <br>Menor velocidad.                  |   |
| <b>Términos utilizados:</b><br><br><b>DESPLAZAR</b><br>Movimiento de manos y dedos en el espacio sin tocar otra parte del cuerpo. |                                       | <b>DESLIZAR</b><br>Movimiento de manos y dedos sobre otra parte del cuerpo.   |    |

*Figura 7 Trayectoria y dirección de los movimientos de las manos [2/2] [14]*

### 1.4.5 Placas de desarrollo electrónico

Las placas de desarrollo electrónico son dispositivos utilizados para la creación de prototipos y el desarrollo de proyectos electrónicos. Estas placas suelen incluir un microcontrolador, el cual se considera una pequeña computadora que se usa para controlar varios componentes del circuito electrónico y una serie de pines de entrada/salida (I/O) para conectarse a otros componentes. Para facilitar su trabajo, las placas de desarrollo a menudo también contienen componentes básicos como resistencias y condensadores.

Algunos ejemplos populares de placas de desarrollo son los siguientes:

- **Arduino:** Mayormente aplicado a proyectos de robótica, IoT y control de procesos, es versátil y posee una gran cantidad de documentación y recursos disponibles en línea. Además, existe una amplia gama de shields (expansores) disponibles para Arduino, lo que permite a los usuarios añadir diferentes funcionalidades a sus proyectos de manera sencilla. [15]
- **Raspberry Pi:** Comúnmente utilizada en proyectos informáticos y multimedia. Es más potente y compleja de utilizar a comparación de Arduino. Raspberry Pi incluye un procesador de alto rendimiento y una serie de pines de entrada/salida (I/O) que se pueden utilizar para conectar diferentes componentes. [15]
- **NodeMCU:** es una placa de desarrollo basada en el microcontrolador ESP8266, aplicado al mundo de la robótica y del IoT. Algunas de las ventajas de esta placa son su bajo costo y su integración de WiFi, lo que la hace ideal para proyectos de conectividad. Además, NodeMCU es compatible con una amplia variedad de lenguajes de programación y cuenta con una gran cantidad de documentación y recursos en línea. [16]
- **BeagleBone:** Utilizada en proyectos de IoT y computación embebida. Posee una alta capacidad de procesamiento y una amplia variedad de pines de entrada/salida (I/O). [16]
- **Particle photon de Adafruit:** una plataforma de desarrollo de IoT que incluye varias placas de desarrollo y herramientas de programación. Posee una amplia gama de funcionalidades de conectividad, como Wifi, Bluetooth y LTE. [16]

#### 1.4.6 Sensores flexibles

Los sensores flexibles son resistencias variables, funcionan midiendo la resistencia eléctrica de una película delgada y flexible de un material conductor, que cambia su resistencia cuando se flexiona o se dobla. Esta película se coloca en una base elástica o flexible, que permite que se doble o se curve cuando se aplica una fuerza. La fuerza aplicada a la película cambia su forma y, por lo tanto, su resistencia eléctrica. La resistencia de la película se puede medir mediante la conexión de los extremos de la película a un circuito eléctrico, donde la resistencia del sensor es más baja cuando la

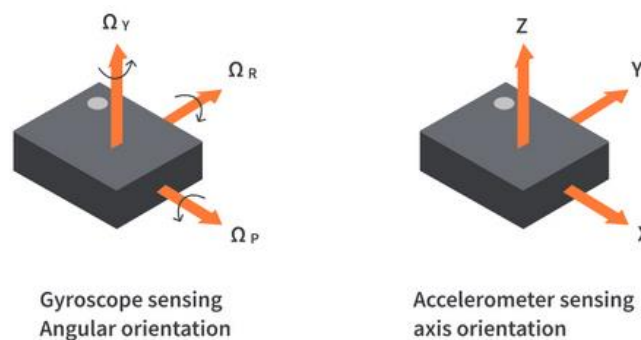
superficie es plana y aumenta cuando se dobla lentamente, alcanzando un máximo cuando está en un ángulo de 90 grados. [17]

Poseen una amplia gama de aplicaciones, desde la monitorización del movimiento corporal en deportes y fitness, hasta la detección de la curvatura en estructuras o materiales. También se utilizan en robótica y en la industria de la fabricación para medir la deformación en piezas o componentes durante el proceso de producción. [17]

#### 1.4.7 Unidad de Medición Inercial (IMU)

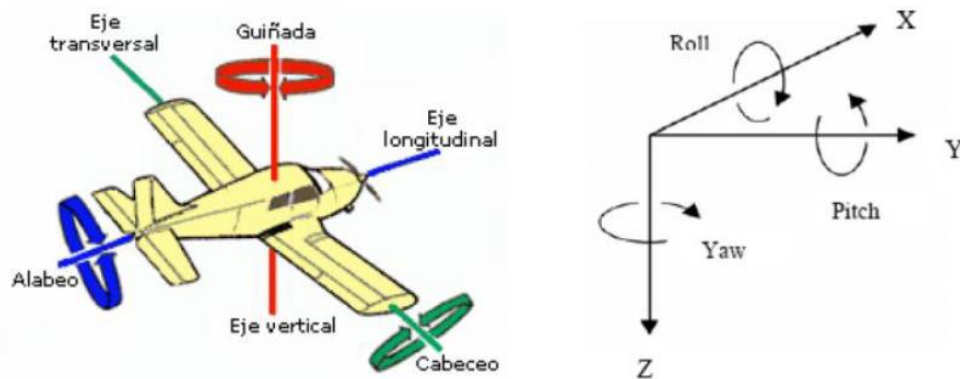
Tanto los acelerómetros como los giroscopios generalmente se encuentran dentro de una Unidad de Medición Inercial o IMU. Es un paquete que contiene acelerómetros, giroscopios, magnetómetros, sensores de temperatura, etc.

Los acelerómetros detectan la aceleración lineal de los dispositivos, es decir, la aceleración a lo largo de un eje. Los giroscopios, por otro lado, funcionan con el efecto Coriolis detectando la velocidad angular, es decir, qué tan rápido gira el cuerpo. Sin embargo, los giroscopios no informan el ángulo actual, informan la velocidad a la que gira el dispositivo. Para obtener la posición del ángulo, debe integrarla con el tiempo. La figura 8 muestra el tipo de movimiento detectado por acelerómetros y giroscopios. [18]



*Figura 8 Orientación de ejes de sensibilidad [18]*

Por otra parte, los acelerómetros poseen un conjunto de ángulos necesarios para medir sus movimientos en un momento dado, estos son: ángulo de alabeo (Roll), gira en torno al eje horizontal (eje X), ángulo de cabeceo (Pitch), gira en torno al eje latitudinal (eje Y). Y ángulo de guiñado (Yaw), es el ángulo Y creado por el movimiento cuando gira en torno al eje vertical (eje Z). [18]

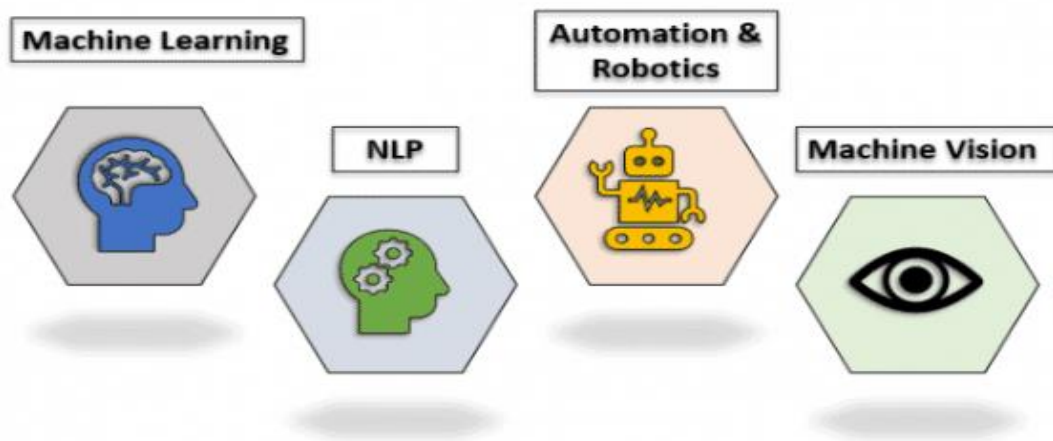


*Figura 9 Sistema de coordenadas [18]*

### 1.4.8 Inteligencia Artificial (IA)

La inteligencia artificial es una rama de amplia de las ciencias de la computación que se ocupa de construir máquinas inteligentes que pueden realizar tareas que normalmente requieren inteligencia humana, como pensar y tomar decisiones, cosas que solo los seres vivos son capaces de hacer. A continuación, se muestran las técnicas principales de la Inteligencia Artificial. [19]

Las principales técnicas de la inteligencia artificial son: Aprendizaje Automático, NLP (Procesamiento del Lenguaje Natural), Automatización y Robótica, y Visión artificial. Cada técnica posee varias ramificaciones de acuerdo con la implementación, área y problemas que requieran resolver. [19]



*Figura 10 Técnicas de la Inteligencia Artificial [19]*

### 1.4.9 Aprendizaje automático

Aprendizaje automático o Machine Learning (ML), es una de las aplicaciones de la IA donde las máquinas no están programadas explícitamente para realizar ciertas tareas; más bien, aprenden y mejoran de la experiencia automáticamente. [19]

Se basa en algoritmos que analizan datos y patrones para identificar relaciones y hacer predicciones o tomar decisiones sin la intervención humana. Ha sido utilizado ampliamente en la vida cotidiana como los motores de búsqueda, los sistemas de recomendación, la visión artificial y el reconocimiento de voz, entre otros. Y ha obtenido un gran impacto en la investigación científica. El aprendizaje automático se divide en aprendizaje supervisado, no supervisado, reforzado y profundo. [19]

#### 1.4.9.1 Aprendizaje Supervisado

Deduce una función de los datos de entrenamiento, que consiste en un conjunto de un objeto de entrada (conocidos como datos de entrenamiento) y la salida deseada. Es decir, permite predecir un resultado determinado a partir de una entrada dada. A menudo el aprendizaje supervisado requiere esfuerzo humano para construir el conjunto de entrenamiento, pero luego automatiza y, a menudo, acelera una tarea que de otro modo sería laboriosa o inviable. Hay dos tipos principales de problemas de aprendizaje automático supervisado, llamados clasificación y regresión. [20]

- **Clasificación:** Utilizado para predecir a que categorías pertenece un elemento. Por ejemplo, clasificar si un correo electrónico es spam, o para identificar en una imagen si contiene un perro o un gato. La clasificación tiene como finalidad predecir una o varias etiquetas de clase, las cuales están en una lista predefinida de posibilidades. [20]
- **Regresión:** Es utilizado para predecir valores numéricos. Por ejemplo, es posible emplear este enfoque para realizar una estimación del valor de una vivienda tomando en consideración aspectos como su tamaño, ubicación y cantidad de habitaciones. [20]

Estos son algunos de los algoritmos de aprendizaje supervisado más importantes:



*Tabla 3 Algoritmos de aprendizaje supervisado*

| <b>Algoritmo</b>                         | <b>Descripción</b>  | <b>Tipo</b>               | <b>Ejemplo</b>   |
|--|---|---------------------------|--|
| <i>k-Nearest Neighbors</i>               | Clasificador basado en la cercanía, busca los k puntos de entrenamiento más cercanos a un punto de datos no etiquetado.                         | Clasificación             | Clasificación de correos electrónicos como spam o no spam.   |
| <i>Linear Regression</i>                 | Modelo para predecir valores numéricos continuos.<br>Modela la relación lineal entre una variable de entrada y una variable de salida continua. | Regresión                 | Predicción del precio de una casa en función de su tamaño.   |
| <i>Logistic Regression</i>               | modela la relación entre variables de entrada y una variable de salida binaria.   | Clasificación             | Predicción de si un cliente comprará o no un producto.       |
| <i>Support Vector Machines (SVMs)</i>    | encuentra un hiperplano que separa los puntos de datos en diferentes clases. encuentra el límite de decisión óptimo entre dos clases.           | Clasificación y regresión | Detección de fraude en transacciones financieras.            |
| <i>Decision Trees and Random Forests</i> | Modelo que divide los datos en ramas para hacer predicciones.   | Clasificación y Regresión | Predicción de si un paciente sobrevivirá o no a una cirugía. |
| <i>Neural networks</i>                   | Modelo basado en el funcionamiento del cerebro humano.  | Clasificación y Regresión | Identificación de objetos en imágenes.                       |

**Fuente:** Investigador basado en [20]

### 1.4.9.2 Aprendizaje no supervisado

El aprendizaje no supervisado se encarga de encontrar patrones, grupos o relaciones en los datos sin la orientación de un supervisor, el algoritmo no requiere etiquetas, categorías predefinidas o información previa sobre los datos de entrada, al contrario, al algoritmo se le pide que descubra patrones inherentes en los datos. Es decir que el

algoritmo debe poder encontrar patrones y relaciones por sí mismo. Una técnica común en el aprendizaje no supervisado es el clustering (agrupación), donde los datos se agrupan en subconjuntos basados en sus características comunes, es útil para descubrir patrones subyacentes en los datos y puede ser utilizado en una variedad de aplicaciones, como segmentación de mercado, análisis de redes sociales y recomendación de contenido. [20]

#### **1.4.9.3 Aprendizaje reforzado**

El aprendizaje por refuerzo se basa en el concepto de un agente, el cual puede observar el entorno, seleccionar y realizar acciones y obtener recompensas a cambio (o sanciones en forma de recompensas negativas de ser el caso). El agente toma una acción y el entorno responde con una recompensa y una nueva observación del estado. El objetivo del agente es aprender una política óptima que maximice la recompensa a largo plazo. [20]

#### **1.4.9.4 Aprendizaje profundo**

El aprendizaje profundo, conocido como Deep Learning, es una técnica que utiliza redes neuronales artificiales, para aprender relaciones y patrones complejos de datos, posee una amplia gama de aplicaciones en áreas como el procesamiento de imágenes, el procesamiento de lenguaje natural y el reconocimiento de voz. Estas redes neuronales son modelos inspirados en la estructura y funcionamiento del cerebro humano, se componen de múltiples capas de neuronas interconectadas, y pueden utilizar una gran cantidad de datos para aprender patrones y relaciones complejas en los datos de entrada. [20]

#### **1.4.10 Funcionamiento del modelo k-Nearest Neighbors (KNN)**

El modelo k-Nearest Neighbors (KNN) o en español “K-Vecino más cercano”, es un algoritmo de aprendizaje supervisado utilizado en la minería de datos y el aprendizaje automático. Su funcionamiento se basa en la clasificación de nuevos datos según la proximidad que tengan con otros datos conocidos. Cada nuevo registro se clasifica en un grupo específico por su proximidad a los vecinos k más cercanos. Calcula la distancia entre el elemento nuevo y los existentes, ordena estas distancias de menor a mayor y elige el grupo con la mayor frecuencia entre los k vecinos más cercanos. Las

fases del algoritmo KNN que realizan todo el proceso de clasificación se describen a continuación. [21]

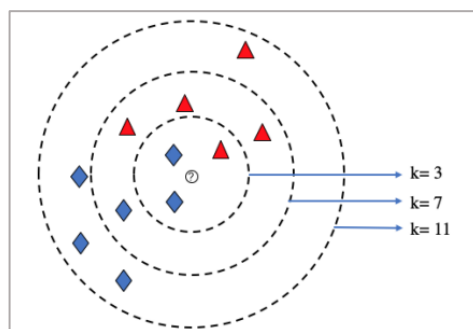
### 1. Fase de entrenamiento

Se lleva a cabo la identificación, aprendizaje y almacenamiento de las características y clases de los datos de entrenamiento. Estos datos, representados como vectores multidimensionales, contienen etiquetas (o tipos) de clase que serán utilizadas posteriormente para la clasificación. Estas etiquetas agrupan los datos en diferentes categorías, permitiendo al algoritmo aprender y reconocer patrones relevantes para futuras clasificaciones. [21]

### 2. Fase de clasificación

Para la fase de clasificación se llevan a cabo varios pasos. En primer lugar, se determina el parámetro  $k$ , que representa el número de vecinos a considerar al evaluar la distancia y determinar la clasificación de un dato en relación con los demás. La selección de este parámetro depende en gran medida de los datos disponibles, donde valores más altos de  $k$  tienden a reducir el impacto del ruido en la clasificación, pero también pueden difuminar los límites entre las clases. [21]

En segundo lugar, se calcula la distancia entre el punto de prueba y los  $k$  vecinos más cercanos, identificando así los puntos más cercanos y a qué clase pertenecen. Utilizando esta información, se asigna al punto de prueba la misma clase que el vecino más cercano, es decir, aquel que presenta la menor distancia. En el algoritmo KNN, la métrica más comúnmente utilizada para calcular las distancias entre puntos es la distancia euclidiana. Sin embargo, también se pueden utilizar otras métricas como la distancia de Manhattan, la distancia de Mahalanobis, coeficientes de correlación, entre otras, dependiendo de las características de los datos y del problema en cuestión. [21]



**Figura 11** Ejemplo del clasificador K-NN [21]

### 1.4.11 Funcionamiento del modelo Random Forest

El modelo Random Forest, o en español Bosques Aleatorios, se basa en un conjunto de árboles de decisión individuales, denominado ensemble. Este modelo se utiliza para abordar el problema de equilibrio entre sesgo y varianza en el aprendizaje estadístico y el Machine Learning. El sesgo se refiere a qué tan cerca están las predicciones promedio del modelo de los valores reales, lo cual indica su capacidad para capturar la relación real entre los predictores y la variable objetivo. Por otro lado, la varianza se refiere a cómo varía el modelo en función de los datos de entrenamiento utilizados. [22]

Una característica distintiva de Random Forest es que en cada árbol solo se utilizan un número  $m$  de predictores, siendo  $m < p$ , donde  $p$  es el total de predictores disponibles. Además, la selección de  $m$  en cada árbol se realiza de forma aleatoria. Sin embargo, puede presentar problemas de sobreajuste, por lo que es necesario ajustar los hiperparámetros adecuadamente. En resumen, Random Forest combina múltiples árboles de decisión utilizando el método de *bagging* para mejorar las predicciones y controlar el equilibrio entre sesgo y varianza. [22]

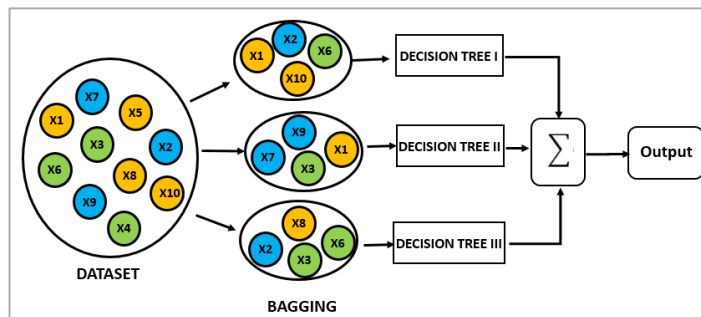


Figura 12 Composición del modelo Random Forest [22]

### 1.4.12 Lenguajes de programación

Un lenguaje de programación es un lenguaje informático donde los programadores crean programas de software, scripts y otros conjuntos de instrucciones para que las computadoras sean capaz de ejecutarlas. A pesar de los puntos en común entre numerosos lenguajes, cada uno tiene su propia sintaxis. Un programador escribe el código fuente en un editor de texto o IDE después de familiarizarse con las reglas, y la estructura del lenguaje. Luego, el programador convierte con frecuencia el código

en un lenguaje de máquina comprensible para la computadora. Los lenguajes de secuencias de comandos necesitan un intérprete para ejecutar la secuencia de comandos porque no necesitan un compilador. [23]

Pueden categorizarse según el enfoque del programador, como puede ser para el desarrollo de aplicaciones, programas, desarrollo de inteligencia artificial, base de datos o incluso juegos. Entre los lenguajes de programación más utilizados en la actualidad están:

- **Python, C/C++ y C#:** Permiten desarrollo de inteligencia artificial o campos relacionados, donde se involucra la creación de interacciones de personajes en juegos de computadora, partes de programas que toman decisiones, chatbots y más. [23]
- **Java, PHP y JavaScript:** Permiten desarrollo de páginas web y aplicaciones de Internet, permiten desarrollos robustos y rápidos. Java por su parte posee una gran portabilidad y puede funcionar en casi cualquier sistema. [23]

## 1.5 Objetivos

### 1.5.1 Objetivo general

Construir un sistema automático de decodificación de lengua de señas ecuatoriano a comunicación verbal.

### 1.5.2 Objetivos específicos

- Establecer el vocabulario básico del lenguaje de señas ecuatoriano basado en el “Diccionario Oficial de Lengua de Señas Ecuatoriana Gabriel Román”, según la realidad actual de las personas con discapacidad.
- Definir técnicas, tecnologías y herramientas a utilizarse en el desarrollo del sistema para aplicar en el algoritmo de aprendizaje automático.
- Implementar el sistema automático de decodificación de lengua de señas ecuatoriano a comunicación verbal para el alfabeto y palabras básicas.
- Validar el funcionamiento del sistema por medio de pruebas técnicas a personas con conocimiento sobre el lenguaje de señas ecuatoriano.

## **CAPÍTULO II**

### **METODOLOGÍA**

#### **2.1 Materiales**

En el presente proyecto, para el desarrollo del guante electrónico se emplearon cinco sensores Flex, para ser colocados en cada dedo de la mano, junto con un sensor MPU 6050, los cuales permiten capturar los movimientos de la mano y dedos necesarios para crear el sistema decodificador, estos datos son enviados a una tarjeta de desarrollo Arduino NANO, la cual se encarga de la adquisición de los datos para posteriormente ser procesados mediante el lenguaje de programación Python y ser visualizados mediante Streamlit, de esta manera se crea la base de datos según el alfabeto, números y palabras básicas.

En la etapa de procesamiento, se emplearon algoritmos de aprendizaje automático supervisado del tipo clasificación. Los algoritmos utilizados son K-Nearest Neighbors (KNN) y Random Forest, por lo tanto, se utilizó las librerías Pandas y NumPy para manipular y modificar las tablas de datos, y la librería Scikit-Learn, para la división de los conjuntos de datos en pruebas y entrenamiento. Por último, se incorporó la biblioteca “gTTS” (Google Text-to-Speech) en la interfaz para traducir los resultados a una comunicación verbal.

#### **2.2 Métodos**

##### **2.2.1 Modalidad de la Investigación**

La modalidad del proyecto para el desarrollo de un sistema automatizado de decodificación de lengua de señas ecuatoriano a comunicación verbal es de Investigación Aplicada, la cual se realizó mediante las siguientes técnicas:

Investigación aplicada, ya que se puso en práctica los conocimientos científicos adquiridos como también la información sobre la tecnología actual relacionada al tema, que sirven para dar solución al problema planteado.

Investigación bibliográfica debido a que el sustento científico del tema planteado se obtuvo de libros, publicaciones, artículos científicos y repositorios disponibles en el internet.

Investigación de campo porque se trata de un estudio de los hechos, en el lugar que se produce para ello se utilizaron técnicas como la observación, la entrevista que permitió estar en contacto directo al investigador con la finalidad de recolectar y registrar información.

### **2.2.2 Recolección de Información**

Para la recolección de información se realizó un análisis y procesamiento de fuentes bibliográficas tales como libros, artículos científicos, revistas indexadas, fuentes online y proyectos de investigación presentes en los repositorios universitarios que estén relacionados con sistemas de traducción de lengua de señas.

### **2.2.3 Procesamiento y Análisis de Datos**

Para el procesamiento y análisis de datos se plantearon las siguientes actividades:

- Revisión, Análisis y filtración de las fuentes de recopiladas.
- Determinación de los métodos de traducción automática de lengua de señas a través de diferentes medios como sensores o prototipos electrónicos.
- Interpretación de la información con mayor relevancia para dar soluciones y resultados para determinar un procedimiento de resolución de la propuesta.
- Planteamiento de la propuesta de solución.

### **2.2.4 Propuesta de Solución**

El desarrollo de un sistema para traducir de manera automática la lengua de señas ecuatoriano a comunicación verbal, permitió descubrir alternativas para aprender y traducir la lengua de señas, debido a que involucra algoritmos de aprendizaje automático lo cual permite garantizar una mejor predicción de las señas, además que permite una gran escalabilidad en cuanto a las palabras y símbolos que puede traducir, como resultado las personas con discapacidad auditiva poseen mayores oportunidades de inclusión debido a que gracias a sistemas como estos las personas oyentes se involucran en mayor medida a aprender lengua de señas y a innovar en este tipo de tecnologías asistivas.

El sistema está orientado a jóvenes y adultos sin necesidad de presentar discapacidad auditiva que deseen traducir o aprender la lengua de señas ecuatoriana mediante el uso de un guante electrónico. El sistema cuenta con un guante electrónico empleado en la mano derecha, por ser la mano dominante, en el cual cuenta con 5 sensores flexibles y un acelerómetro/giroscopio para la captura de los datos, por lo tanto, la fiabilidad del sistema es alto debido a que se tiene un mejor control del equipo electrónico y de la

programación. Además, se busca cumplir con un prototipo de bajo costo para dar mayor accesibilidad a los usuarios y a futuras investigaciones.

### 2.2.5 Desarrollo del Proyecto

Para el desarrollo del sistema se planteó las siguientes actividades:

*Tabla 4 Actividades para el Desarrollo del Proyecto (1/2)*

|  |   |
|--|---|
| <b>Tema:</b> Sistema automático de decodificación de lengua de señas ecuatoriano a comunicación verbal.  |   |
| <b>Objetivo General:</b> Construir un sistema automático de decodificación de lengua de señas ecuatoriano a comunicación verbal.   |   |
| <b>Objetivo Específico 1</b>   |   |
| Establecer el vocabulario básico del lenguaje de señas ecuatoriano basado en el “Diccionario Oficial de Lengua de Señas Ecuatoriana Gabriel Román”, según la realidad actual de las personas con discapacidad. |   |
| <b>1</b>   | Categorización de los diferentes tipos de discapacidad auditiva total o parcial.  |
| <b>2</b>   | Análisis del alfabeto dactilológico ecuatoriano otorgado por el “Diccionario de lengua de señas ecuatoriano Gabriel Román”.   |
| <b>3</b>   | Determinación de las palabras, símbolos y signos más empleados por la comunidad sorda.  |
| <b>Objetivo Específico 2</b>   |   |
| Definir técnicas, tecnologías y herramientas a utilizarse en el desarrollo del sistema para aplicar en el algoritmo de aprendizaje automático.   |   |
| <b>4</b>   | Análisis de las diferentes herramientas de comunicación para asistencia de personas que viven en situación de discapacidad auditiva y de habla.   |
| <b>5</b>   | Recolección de información de las diferentes alternativas sobre las tecnologías, herramientas, software y hardware para el desarrollo del sistema de decodificación de lengua de señas. |
| <b>6</b>   | Selección de las técnicas de aprendizaje automático para el reconocimiento del lenguaje de señas.   |
| <b>Objetivo Específico 3</b>   |   |
| Implementar el sistema automático de decodificación de lengua de señas ecuatoriano a comunicación verbal para el alfabeto y palabras básicas.  |   |
| <b>7</b>   | Calibración del funcionamiento del dispositivo para la captura de los gestos de la mano.  |
| <b>8</b>   | Desarrollo de la programación y la interfaz del sistema para la decodificación del lenguaje de señas.   |
| <b>9</b>   | Recopilación de las señales para crear una base de datos de los diferentes movimientos de la mano.  |
| <b>10</b>  | Entrenamiento aplicado a las técnicas de aprendizaje automático para el reconocimiento del lenguaje de señas a comunicación verbal.   |

**Fuente:** Investigador



*Tabla 5 Actividades para el Desarrollo del Proyecto (2/2)*

| <b>Objetivo Específico 4</b>  |   |
|---|---|
| Validar el funcionamiento del sistema por medio de pruebas técnicas a personas con conocimiento sobre el lenguaje de señas ecuatoriano. |   |
| <b>11</b>   | Obtención de pruebas de funcionamiento y parámetros del sistema.            |
| <b>12</b>   | Desarrollo del reporte con los datos obtenidos.                             |
| <b>13</b>   | Análisis de resultados para la redacción de conclusiones y recomendaciones. |
| <b>14</b>   | Elaboración del informe final de la propuesta.                              |

**Fuente:** Investigador

## **CAPITULO III**

### **RESULTADOS Y DISCUSIONES**

#### **3.1 Análisis y discusión de los resultados**

El desarrollo y construcción del sistema automático de decodificación de lengua de señas ecuatoriano a comunicación verbal, constituye como una nueva herramienta de apoyo para las personas con y sin discapacidad auditiva, permite al usuario aprender y traducir un vocabulario básico conformado por el alfabeto dactilológico, los primeros diez números y una selección de palabras básicas, mediante el uso aprendizaje automático. Este proyecto permite incentivar a las personas a aprender la lengua de señas e involucrarse con la comunidad sorda, reduciendo barreras de comunicación mediante un sistema interactivo y fácil de utilizar.

#### **3.2 Desarrollo de la propuesta**

Para el desarrollo del sistema se realizó una investigación de acuerdo con el “Diccionario de lengua de señas ecuatoriano Gabriel Román”, con el objetivo de determinar cuáles son las palabras, símbolos y signos más empleados por la comunidad sorda.

En la investigación se toma en cuenta que necesidades presentan las personas sordas al momento de tratar de comunicarse con personas oyentes, o viceversa, y que estas no conozcan o tengan acceso a un intérprete, por lo tanto, este proyecto se desarrolló para aprender el alfabeto dactilológico ecuatoriano como base para poder comunicarse de manera rápida y precisa, además de ciertas palabras definidas para facilitar dicha comunicación. Se utilizó un guante electrónico para la captura de los movimientos de la mano, el cual es capaz de conectarse al ordenador, registrar, guardar y procesar los datos, el sistema permite replicar los símbolos mediante una interfaz gráfica donde se muestran las posiciones de la mano y el significado de cada una. Finalmente, se aplicó técnicas de aprendizaje automático, con el objetivo de validar el porcentaje de confiabilidad del sistema.

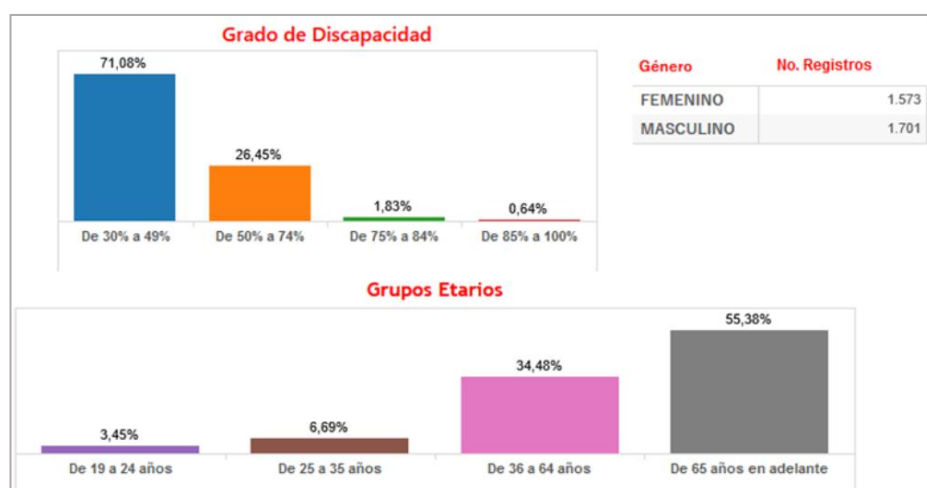
##### **3.2.1 Consideraciones previas para el desarrollo del sistema**

En el desarrollo del sistema y fabricación del prototipo del guante electrónico para traducir la lengua de señas ecuatoriano, fue fundamental determinar criterios de diseño como:

- Definir a los usuarios del sistema automático, junto con el glosario básico, en base a las palabras, símbolos y signos mayormente empleados por la comunidad sorda, con el objetivo de ajustar el diseño del guante a los gestos seleccionados para el sistema.
- Los componentes electrónicos son de pequeñas dimensiones y resistentes, con el objetivo de que se adapten a los diferentes movimientos de la mano, junto con los movimientos de flexión de cada dedo.
- La interfaz gráfica debe tener una apariencia amigable e intuitiva, para que sea fácil de manejar por el usuario.

### 3.2.1.1 Usuarios del sistema automático

El diseño del prototipo del guante electrónico y del sistema automático de decodificación de lengua de señas ecuatoriano está orientado a jóvenes y adultos de la provincia de Tungurahua, quienes representan un grupo demográfico con un mayor número de personas sordas, de acuerdo con datos registrado por la CONADIS. [8] En la figura 13 se presenta la cantidad de 3274 personas registradas que presentan discapacidad auditiva.



**Figura 13** No. Registros de personas con discapacidad auditiva en Tungurahua [8]

Este enfoque busca proporcionar una herramienta accesible para aquellas personas que deseen aprender o traducir un vocabulario básico de la lengua de señas ecuatoriano,

independientemente de que presenten o no discapacidad auditiva y del habla. Por lo tanto, el desarrollo del sistema automático busca fomentar la inclusión social y motivar a las personas oyentes a involucrarse con este tipo de lenguaje.

### 3.2.1.2 Palabras, símbolos y signos más empleados por la comunidad sorda

Según el diccionario de lengua de señas ecuatoriano “Gabriel Román”, la forma de comunicación de las personas sordas se basa en el alfabeto dactilológico ecuatoriano, donde consta de 30 letras. De esta manera pueden comunicarse a través de deletrear cada letra, sin embargo, el diccionario también ofrece palabras ya definidas para agilizar la comunicación.

Para finalidades del proyecto, se analizó palabras comunes que requieren las personas en una comunicación básica, donde se clasifican en gestos estáticos (sin movimiento) y gestos dinámicos (implican movimiento), es así como se decidió clasificar los gestos que se emplearon en el proyecto de la siguiente manera.

### 3.2.1.3 Gestos estáticos

En esta parte se incluyen las 25 letras del alfabeto dactilológico ecuatoriano, junto con los primeros 10 números, presentados en la tabla 6 y figura 14, respectivamente.

*Tabla 6 Gestos estáticos alfabeto dactilológico*

| Gestos Estáticos |   |   |    |   |   |   |
|------------------|---|---|----|---|---|---|
| A                | B | C | CH | D | E | F |
| G                | H | I | K  | L | M | N |
| O                | P | Q | R  | S | T | U |
| V                | W | X | Y  |   |   |   |

Fuente: Investigador basado en [13]



*Figura 14 Números en lengua de señas ecuatoriano [13]*

### 3.2.1.4 Gestos dinámicos

Incluyen las 5 letras del alfabeto que contienen movimiento los cuales son J, LL, Ñ, RR y Z, como se aprecian en la figura 15.

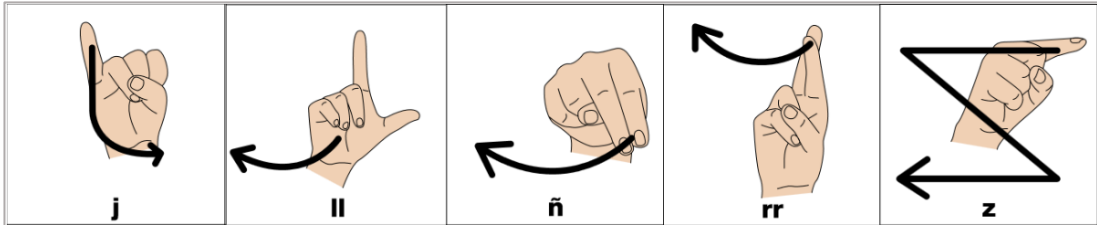






Figura 15 Gestos dinámicos del alfabeto dactilológico ecuatoriano [13]

De igual manera se analizó las necesidades que tienen las personas sordas al momento de entablar conversaciones básicas, por lo tanto, se tiene las tablas 7 y 8 como la lista de las palabras básicas del sistema.

Tabla 7 Gestos dinámicos para el sistema 1/2

| COLORES   |  |
|---|--|
| <p><b>Amarillo</b></p>  <p>La mano se desliza de adelante hacia atrás de la cabeza en forma ondulante.</p> | <p><b>Azul</b></p>  <p>La mano se desliza por la mejilla.</p> |
| <p><b>Blanco</b></p>  <p>La mano con palma extendida se mueve de abajo hacia arriba.</p>                   | <p><b>Negro</b></p>  <p>La mano se abre y se cierra.</p>      |

Fuente: Investigador basado en [13]

*Tabla 8 Gestos dinámicos para el sistema 2/2*

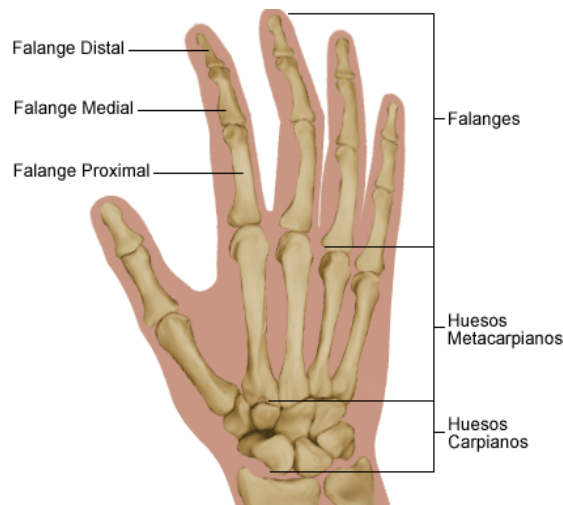
| <b>PALABRAS BÁSICAS</b>   |   |
|---|---|
| <p><b>Buenos días</b></p>  <p>La mano desde la frente se desplaza hacia abajo y después hacia arriba mientras se abre simultáneamente.</p> | <p><b>Hola</b></p>  <p>La mano con los dedos juntos se abre y se cierra repetidas veces.</p> |
| <p><b>Chao</b></p>  <p>La mano se mueve hacia afuera y hacia adentro repetidamente.</p>   | <p><b>Gracias</b></p>  <p>La mano inicia desde el mentón y se mueve hacia abajo.</p>        |
| <p><b>Bien</b></p>  <p>La mano se mueve levemente hacia adelante.</p>  | <p><b>Mal</b></p>  <p>La mano se desliza hacia abajo.</p>                                  |

**Fuente:** Investigador basado en [13]

Es importante mencionar que las palabras deben poder realizarse con una única mano, debido a que se busca crear un prototipo de bajo costo, de esta manera se implementó únicamente un guante para la mano derecha, al ser la que mayormente predomina en la población.

### 3.2.1.5 Análisis de los movimientos de las manos y los dedos

Para extraer los datos de cada letra, palabra y número de la lengua de señas de manera precisa, fue fundamental realizar un análisis completo de los movimientos que una mano puede realizar desde una perspectiva anatómica. Esto permitió determinar los ángulos y direcciones que se registraron para la captura de cada movimiento. Además, se tomaron en cuenta los niveles de flexión de los dedos para lograr una captura completa de los gestos. Por lo tanto, en la figura 16 se puede apreciar la anatomía de la mano de acuerdo con sus huesos.



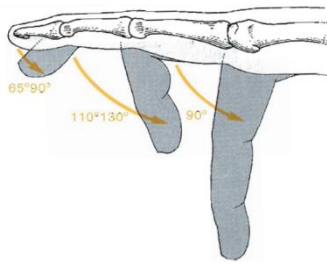
*Figura 16 Anatomía de la mano [24]*

- **Metacarpiano:** el hueso que conecta el dedo con la muñeca, excepto el pulgar.
- **Falange proximal:** el hueso que se encuentra en la base del dedo, conectado a la palma de la mano.
- **Falange medial:** el hueso que está en el medio del dedo, entre la punta y la base.
- **Falange distal:** el hueso que está al final del dedo.

La tabla 9 muestra los diferentes movimientos que puede realizar una mano anatómicamente hablando, cada movimiento se analizó en base a los requerimientos del proyecto. La flexión de los dedos permite representar letras y palabras específicas, mientras que la abducción y aducción de los dedos añaden mayor riqueza y expresividad a los gestos. La flexión y extensión de la mano, junto con la inclinación radial y cubital de la muñeca, contribuyen a la formación de patrones de gestos más complejos y significativos.

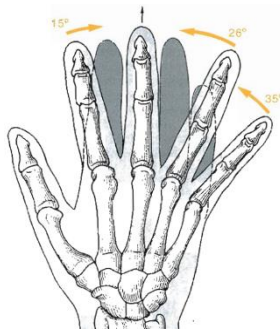
**Tabla 9** Movimientos de la mano

**Movimiento de flexión de los dedos**



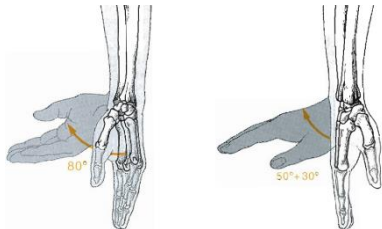
Corresponde a los ángulos de flexión de los dedos, normalmente este movimiento alcanza ángulos entre 30° a 90°, dependiendo de la posición de cada dedo y la movilidad de cada articulación. [24]

**Abducción y Aducción de los dedos**



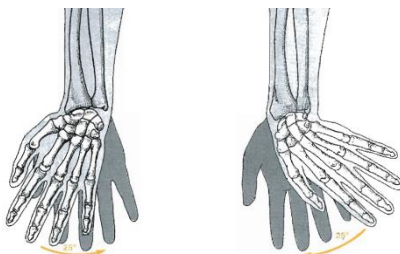
Son los movimientos de separación (abducción) y acercamiento (aducción) de los dedos entre sí, tiene como punto de referencia el dedo medio y el plano de la palma de la mano. El movimiento de separación alcanza 60° y 45° para los dedos índice, anular y meñique respectivamente. [24]

**Flexión y extensión de la mano**



Para la captura de los gestos dinámicos, se debe considerar los movimientos de la mano en el espacio. Considera como punto de referencia el sentido de la palma y dorso de la mano junto con la muñeca, de esta manera se miden los ángulos de flexión, los cuales alcanzan valores de 80° y 90°. [24]

**Inclinación radial y cubital de la muñeca**



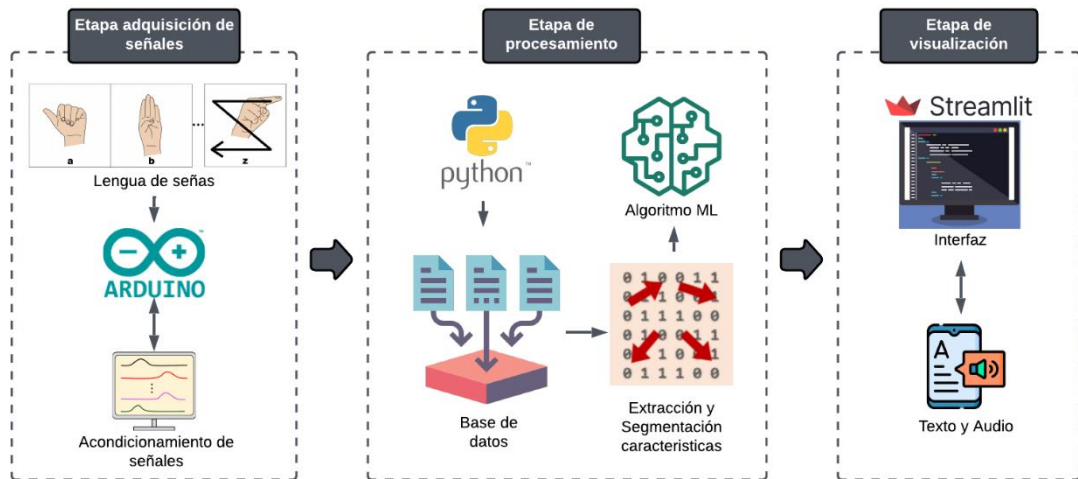
Considera el giro de la muñeca en direcciones laterales, donde llega a obtener ángulos de 25°. [24]

**Fuente:** Investigador basado en [24]



### 3.2.2 Esquema general del proyecto

La siguiente figura 17 representa una descripción metodológica del desarrollo del sistema traductor de lengua de señas ecuatoriano.



*Figura 17 Esquema general del proyecto*

*Fuente: Investigador*

- a) **Etapa de adquisición de señales:** Se adquiere las señales de la lengua de señas ecuatoriano a través de un guante electrónico. Se realiza un proceso de acondicionamiento de los sensores requeridos por el sistema mediante la placa de desarrollo Arduino.
- b) **Etapa de procesamiento:** Una vez recopilado los datos, se procede a crear una base de datos mediante el lenguaje de programación Python. En esta etapa se preparan los datos para la extracción y segmentación de características necesarias para aplicar los algoritmos de Aprendizaje Automático.
- c) **Etapa de visualización:** Por último, se crea una interfaz gráfica mediante Streamlit, un Framework de Python que permite crear un entorno web local. En esta interfaz, se muestra el alfabeto dactilológico ecuatoriano, los números y las palabras que conforman el sistema, además al momento de la traducción se presenta una salida tanto de texto como de audio.

### 3.2.3 Análisis y selección de los componentes electrónicos

#### 3.2.3.1 Selección del sensor flexible

El movimiento de flexión de cada dedo puede ser capturado por sensores flexibles, sin embargo, se debe tomar en cuenta los ángulos y curvatura que pueden registrar, por lo cual se compara dos alternativas las cuales se muestran en la tabla 10:

*Tabla 10 Comparación sensores de flexión*

| Parámetros                        | Hoja conductiva tipo Velostat  | Sensor flexible   |
|-----------------------------------|--|---|
| <b>Dimensiones</b>                | Existen en tres medidas:<br>10cmx10cm<br>30cmx30cm<br>50cmx50cm  | Existen en dos medidas<br>2.2" (5.58cm)<br>4.5" (11.43cm)   |
| <b>Voltaje de operación</b>       | 5 a 12 V   | 5 a 12 V  |
| <b>Espesor</b>                    | 0.1mm  | 0.48mm  |
| <b>Peso</b>                       | 18.66gr aprox. la hoja   | 0.27gr  |
| <b>Temperatura límite</b>         | -45°C a 65°C   | -35°C a +80°C   |
| <b>Resistencia plana</b>          | Depende de la elaboración  | 25K Ohms  |
| <b>Tolerancia de resistencia</b>  | Depende de la elaboración  | ±30%  |
| <b>Características especiales</b> | Resistividad Volumétrica:<br><500 ohm-cm<br>Resistividad Superficial: <<br>31,000 ohms/cm <sup>2</sup> | Rango de resistencia a la flexión: 45 K a 125 K ohmios (dependiendo del radio de curvatura)<br>Potencia nominal: 0,50 vatios continuos. pico de 1 vatio |
| <b>Costo</b>                      | Bajo   | Medio   |

Fuente: Investigador basado en [25], [26]


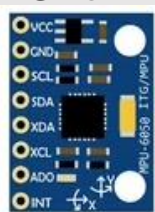

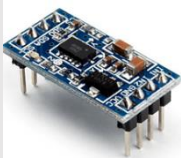
Para el desarrollo del guante electrónico se consideró fabricar los sensores con hojas conductivas Velostat, sin embargo, presentan una significativa variación de los valores censados, al ser elaborados de manera casera, el acondicionamiento de las señales no se logró estandarizar a las mediciones requeridas, por lo que conlleva a un riesgo en los registros de los datos, por lo tanto, se seleccionó los sensores Flex comerciales. De esta manera se puede registrar todos los movimientos necesarios de los dedos de la mano para cada gesto del sistema traductor, los cuales requieren diversos ángulos de

flexión de al menos 90° a 130° necesarios para diferenciar cada gesto, por lo tanto, se emplearon 5 sensores Flex de 2.2” colocados en cada dedo.

### 3.2.3.2 Selección del sensor inercial

Para capturar los movimientos de flexión y extensión de la mano junto con la inclinación de la muñeca, se realizó una comparación de los diferentes módulos de acelerómetros disponibles capaces de capturar dichos datos, teniendo a continuación la siguiente tabla:

*Tabla 11 Comparación sensores gestuales*

| Parámetros                        | MMA7361<br> | MPU-6050<br>GY-521<br> | ADXL345<br> | MMA7455<br> |
|-----------------------------------|--|---|--|--|
| <b>Dimensiones</b>                | 28mm x 17mm x 1 mm   | 22 mm x 15 mm x 1 mm  | 3mm x 5 mm x 1 mm  | 3mm x 5mm x 1mm  |
| <b>Rango</b>                      | ±1.5 g y ±6g   | ± 2, 4,8,16g  | ±16 g  | ±2g, ±4g, ±8g  |
| <b>Interfaz</b>                   | Analógica  | Digital   | Digital SPI /I2C   | Digital SPI/I2C  |
| <b>Ejes</b>                       | 3  | 3   | 3  | 3  |
| <b>Requerimientos de energía</b>  | 2.2 a 3.6 V, 400µA   | 2.4 a 3.5[V], 500 µA  | 2.0 V to 3.6 V40 µA  | 2.4 V – 3.6 V400 µA  |
| <b>Características especiales</b> | Compensación de temperatura. Sensibilidad seleccionable. Detección 0g.                       | Acelerómetro y giroscopio en los 3 eje. Posee ADC internos de 16Bit y se maneja por I2C                 | Resolución fija de 10 bits. Detección de doble golpe. Detección caída libre.                   | Dos pines de interrupción. Detección golpe, vibración y caída libre.                           |
| <b>Costo</b>                      | \$8,02   | \$ 4,20   | \$ 5,10  | \$8,32   |

Fuente: Investigador basado en [27]




Los movimientos de la mano analizados previamente, como lo son flexión, extensión e inclinación, son representativos de los gestos dinámicos seleccionados para el sistema. Con el objetivo de capturar estos movimientos, se eligió el sensor MPU-6050. Este sensor incorpora la capacidad de medir la aceleración y giros en los tres ejes (X, Y, Z). Esto permite capturar la orientación angular y los movimientos de abducción

radial y cubital de la mano en los intervalos de tiempo necesarios para el sistema. Además, el costo accesible y las dimensiones del sensor lo hace adecuado para la construcción del guante. En el Anexo 2 constan las características y especificaciones técnicas del sensor MPU-6050.

### 3.2.3.3 Selección de la placa de desarrollo

Es necesario una placa de desarrollo capaz de procesar una amplia cantidad de información junto con una buena cantidad de entradas analógicas debido a que los sensores Flex y el sensor MPU-6050 presentan entradas análogas. De igual manera sus dimensiones deben ser pequeñas para dar comodidad en el diseño del guante electrónico, por tal motivo, se realizó una tabla comparativa de las distintas placas que se ajustan a los requerimientos.

*Tabla 12 Comparación placas de desarrollo*

| Parámetros                     | <br><b>Arduino MEGA</b> | <br><b>Arduino UNO</b> | <br><b>Arduino NANO</b> |
|--------------------------------|---|---|---|
| <b>Dimensiones</b>             | 101.52 mm x 53.3 mm   | 6.8 cm x 5.3 cm   | 45 x 18 Mm  |
| <b>Peso</b>                    | 35 gr   | 25 gr   | 5 gr  |
| <b>Voltaje de Operación</b>    | 5 [V]   | 5 [V]   | 7V Input, 5V Operating  |
| <b>Procesador</b>              | ATMega 2560   | ATmega 328  | ATmega328   |
| <b>Frecuencia de Operación</b> | 16 Mhz  | 16 MHz  | 16Mhz   |
| <b>Memoria Flash</b>           | 256 KB  | 32 KB   | 32 KB   |
| <b>SRAM</b>                    | 8 KB  | 2 KB  | 2 KB  |
| <b>Pines</b>                   | 54 digitales<br>16 analógicos   | 14 digitales<br>6 analógicos  | 14 digitales, 6 Pwm,<br>8 analógicos  |
| <b>Costo</b>                   | Medio   | Bajo  | Bajo  |

Fuente: Investigador basado en [28], [29]

Para la implementación del prototipo del guante electrónico se eligió la placa Arduino Nano debido a su cantidad de pines analógicos aptos para conectar los sensores Flex y el sensor MPU-6050, se requiere aproximadamente 8 entradas analógicas y finalmente su pequeño tamaño favorece a una construcción sencilla y ergonómica para el usuario.

En el apartado de anexos constan las características y especificaciones técnicas de la tarjeta electrónica seleccionada.

### 3.2.4 Análisis y selección del software del sistema

Para la selección del lenguaje de programación se utilizó Python, dado que ofrece una sintaxis simple y posee una amplia disponibilidad de bibliotecas para el desarrollo de Aprendizaje Automático, lo cual permite cumplir con los objetivos del proyecto. En cuanto a la interfaz gráfica se optó por analizar las diferentes herramientas compatibles con Python para la creación de aplicativos webs.

*Tabla 13 Selección del Framework para la interfaz*

| <b>Característica</b>            | <b>Streamlit</b>   | <b>Gradio</b>  | <b>Dash</b>   |
|----------------------------------|--|--|---|
| <b>Soporte de entrada/salida</b> | Apoyo completo a los widgets de entrada/salida y los widgets de visualización de datos   | Gran soporte a los widgets de entrada/salida y los widgets de visualización de datos   | Soporte limitado a través de componentes de entrada/salida y visualizaciones personalizadas   |
| <b>Documentación</b>             | completa y detallada   | clara y detallada  | limitada  |
| <b>Facilidad de uso</b>          | Fácil de aprender y usar   | Fácil de usar, pero con una curva de aprendizaje ligeramente más empinada debido a su enfoque en modelos pre-entrenados  | Curva de aprendizaje más empinada, requiere un conocimiento sólido de HTML, CSS, y JavaScript   |
| <b>Ecosistema</b>                | En pleno auge, con una gran cantidad de bibliotecas y herramientas de apoyo  | Emergente, con menos bibliotecas y herramientas de apoyo   | Robusto, con una gran cantidad de bibliotecas y herramientas de apoyo   |
| <b>Ventajas</b>                  | Cuenta con una comunidad activa con documentos de soporte y ejemplos prácticos. Permite actualizar la visualización de los datos en tiempo real. Integra modelos de ML fácilmente. | Permite probar rápidamente modelos de aprendizaje automático sin escribir todo el código desde cero. Posee una amplia gama de componentes interactivos predefinidos para el usuario. | Proporciona flexibilidad en la personalización del diseño y apariencia de la aplicación web. Permite manejar conjuntos de datos de gran tamaño sin exceder los requerimientos de almacenamiento de memoria. |

|                    |   |   |   |
|--------------------|---|---|---|
| <b>Desventajas</b> | La personalización del diseño no es muy amplia. Puede tener problemas cuando se manipulan conjuntos de datos muy grandes. | La personalización del diseño puede ser limitada. Puede no ser adecuado para proyectos más avanzados o complejos. | La curva de aprendizaje es un poco más alta que otras bibliotecas, lo que hace que pueda ser difícil de aprender por principiantes. No hay tantas plantillas y ejemplos de uso disponibles. |
|--------------------|---|---|---|

**Fuente:** Investigador basado en [30]

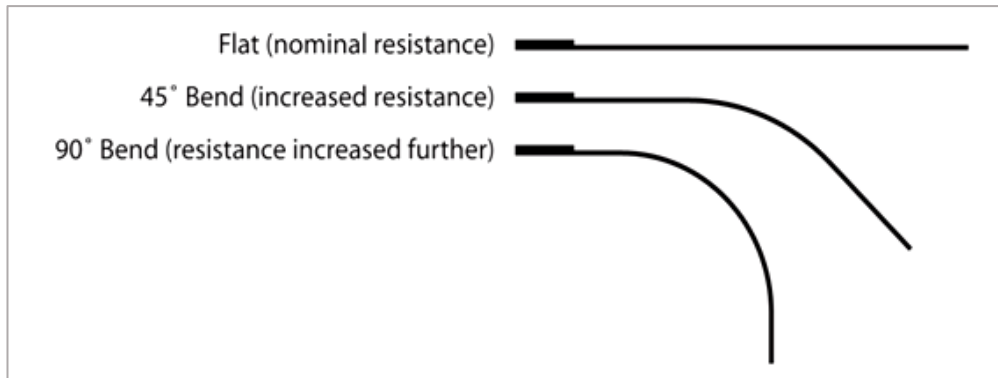
Para la creación de la interfaz gráfica se seleccionó Streamlit, debido a que su curva de aprendizaje es la más fácil de usar y más completa para construir aplicaciones web de aprendizaje automático, además que posee mayor documentación detallada y un enfoque en la facilidad de uso debido a que está enfocada en crear aplicaciones web personalizadas para aprendizaje automático en poco tiempo y es compatible con las principales bibliotecas de Python, como scikit-learn, Keras, PyTorch, SymPy (latex), NumPy, pandas, Matplotlib, etc. [31]

### **3.2.5 Consideraciones previas para el desarrollo del guante electrónico**

Se procedió a realizar un análisis de los sensores seleccionados, con el fin de calibrar su funcionamiento a los requerimientos del sistema, de esta manera se detalla a continuación el acondicionamiento de los sensores Flex y la calibración del sensor MPU 6050 a la placa de desarrollo Arduino Nano, antes de realizar la montura del guante.

#### **3.2.5.1 Funcionamiento del sensor Flex 2,2”**

Como se mencionó anteriormente, los sensores flexibles básicamente son resistencias variables cuya resistencia nominal aumenta cuando el sensor está doblado. Dependiendo de esta manera el aumento de la resistencia de acuerdo con la linealidad de la superficie. [17]

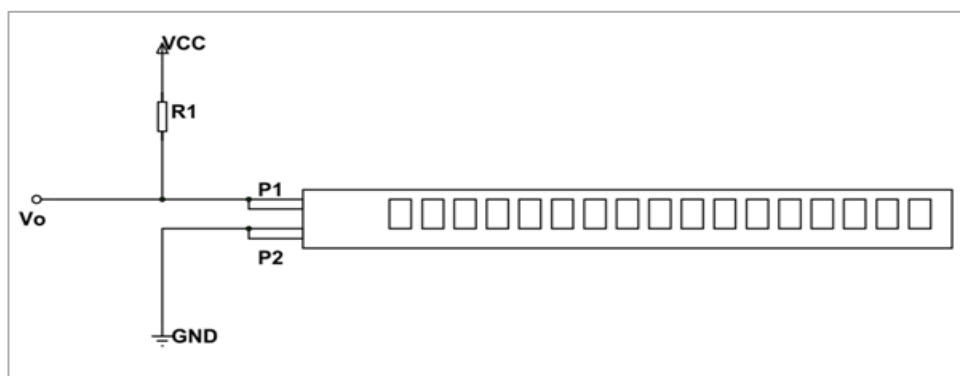


**Figura 18** Funcionamiento del sensor Flex [17]

Como se muestra en la figura 18, cuando la superficie del sensor Flex es completamente lineal, tiene su resistencia nominal. Cuando está doblado en un ángulo de 45 grados, la resistencia aumenta al doble que antes. Y cuando la inclinación es de 90 grados, la resistencia podría llegar hasta cuatro veces la resistencia nominal. Entonces, la resistencia a través de los terminales se eleva linealmente respecto al ángulo doblado, de esta manera convirtiendo el ángulo flexible en parámetro de resistencia. [17]

### Acondicionamiento del sensor Flex

Para la lectura del sensor Flex es necesario convertir el parámetro de resistencia a voltaje, para eso se usó un circuito divisor de voltaje presentado en la figura 19, el cual emplea la ecuación 1.



**Figura 19** Circuito divisor de voltaje [17]

Por lo tanto, se tiene la siguiente ecuación:

$$V_o = V_{CC} \left( \frac{R_x}{R_1 + R_x} \right) \quad \text{Ecuación 1}$$

Donde:

$R_x$  = resistencia del sensor Flex

$R_1$  = Resistencia constante

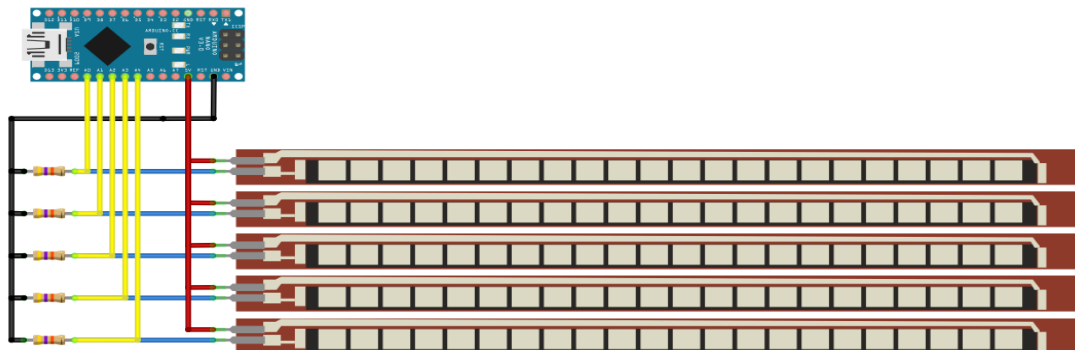
$V_o$  = Voltaje de salida.

De acuerdo con el fabricante los valores de  $R_1$  pueden variar entre  $10\text{ K}\Omega$  a  $100\text{ K}\Omega$ , el cual recomienda utilizar una resistencia de  $47\text{ K}\Omega$ , al mostrar un amplio rango de valores mínimos y máximos de acuerdo con cada ángulo de flexión, y, por lo tanto, cumple con los objetivos del diseño del guante electrónico.

Para facilitar la lectura de los datos de acuerdo con la flexión de los dedos se realizó un reajuste al circuito divisor de voltaje, se invirtió la polaridad, para de esta manera tener un valor nominal alto, y cada vez que se flexione este valor disminuya, con la finalidad de facilitar la lectura de los datos al momento de realizar las señas del sistema, de esta manera la lectura de los datos respecto a los valores mínimos y máximos de flexión son directamente proporcionales.

### Diagrama de conexión de los sensores Flex

El diagrama de conexión del circuito junto con el divisor de voltaje se muestra en la figura 20. Como se observa en la figura el pin P1 del sensor Flex se conecta a 5V, mientras el pin P2 está conectado a la resistencia de  $47\text{ K}\Omega$ , para posteriormente ser conectado a tierra, de esta manera se logra el reajuste del circuito para facilitar la lectura de los datos de los 5 sensores Flex.

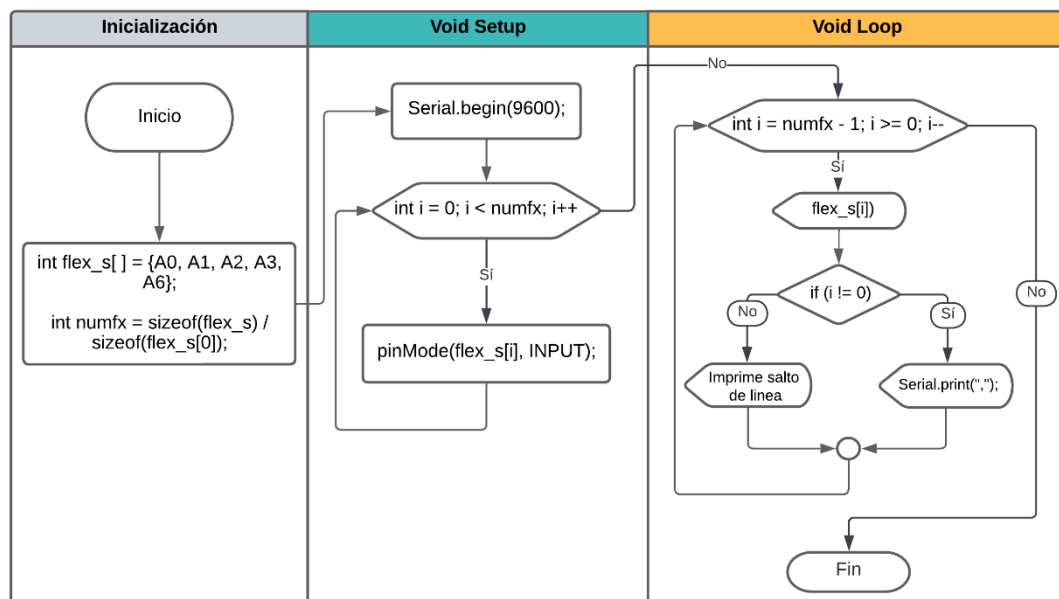


*Figura 20 Diagrama de conexión de los sensores Flex al Arduino Nano*

**Fuente:** Investigador



La programación en Arduino se observa en la figura 21. Se inició con un arreglo flex\_s para leer los valores analógicos de los pines correspondientes a los sensores flexibles. Después se inició el void setup donde se coloca en un ciclo for la lectura de los sensores, para de igual manera en el void loop imprimir los datos a través de otro ciclo for. Estos valores se envían a través de la comunicación serial utilizando Serial.print(), luego, se espera un breve intervalo de tiempo antes de repetir el proceso. Obtenido los datos, se procede a analizar individualmente el valor nominal y grados de flexión con el fin de determinar qué tan viable es la lectura para capturar movimientos de los dedos.



**Figura 21** Diagrama de flujo de los sensores Flex

**Fuente:** Investigador

Como se puede observar en la tabla 14, se registró cinco niveles de flexión, con la finalidad de obtener un amplio rango de valores, de esta manera cada porcentaje corresponde a la flexión de las falanges Distal, Medial y Proximal de cada dedo, con los valores de 25%, 50% y 75% respectivamente. Lo que permite concluir que la ubicación óptima de los sensores Flex es a partir de la unión de los nudillos del guante.

*Tabla 14 Mediciones de mínima y máxima flexión*

| <b>Mediciones de mínima y máxima flexión de cada sensor</b> |               |               |              |               |                |
|---|---------------|---------------|--------------|---------------|----------------|
|   | <i>Pulgar</i> | <i>Índice</i> | <i>Medio</i> | <i>Anular</i> | <i>Meñique</i> |
| <i>Min</i>  | 334           | 233           | 313          | 384           | 326            |
| <i>25%</i>  | 448           | 313           | 420          | 432           | 390            |
| <i>50%</i>  | 560           | 434           | 527          | 466           | 415            |
| <i>75%</i>  | 654           | 465           | 608          | 644           | 593            |
| <i>Max</i>  | 682           | 522           | 651          | 671           | 638            |

Fuente: Investigador

### **3.2.5.2 Funcionamiento del Sensor MPU 6050**

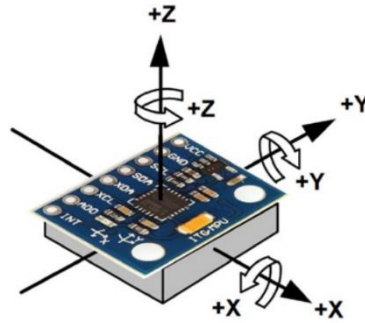
El MPU6050 es una unidad de medición inercial IMU de 6 grados de libertad (DoF) que combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes diseñado para Arduino. Los acelerómetros y giroscopios internamente tienen un MEMS (MicroElectroMechanical Systems) para su funcionamiento, lo que les permite medir la aceleración, la velocidad, la orientación, el desplazamiento y muchos otros parámetros relacionados con el movimiento de un sistema u objeto. [32]

El sensor MPU6050 admite un protocolo de comunicación I2C de dos direcciones separadas: 0x68HEX y 0x69HEX. Esto permite usar dos MPU6050 de ser el caso en un mismo bus, o evitar conflictos de direcciones con otro dispositivo. [32]

- **Aceleración y acelerómetro:** Los acelerómetros funcionan de manera similar a un sistema de masa y resorte, permitiendo la medición de la aceleración. Son usados para medir el movimiento lineal ya sea movimiento, choque o vibración, pero sin una referencia fija. [32]
- **Velocidad angular y giroscopio:** el giroscopio mide la velocidad angular, y si se integra con respecto al tiempo se obtiene el desplazamiento angular (posición angular si se sabe dónde se inició el giro). Se basa en el principio del efecto Coriolis, se produce cuando una masa en movimiento experimenta un movimiento angular externo, generando una fuerza que provoca un desplazamiento perpendicular de la masa. Cuando se aplica un movimiento angular, el efecto Coriolis provoca un cambio en la capacitancia entre las masas dependiendo del eje

del movimiento angular. Este cambio en la capacitancia se detecta y luego se convierte en una lectura. [32]

Es importante tener en cuenta la orientación de los ejes indicada en el módulo MPU6050 para evitar errores en la interpretación del signo de las aceleraciones y giros.



*Figura 22 Movimiento de los ejes del Sensor MPU-6050 [32]*

### **Calibración del sensor MPU 6050**

Es necesario calibrar el sensor de acuerdo con los requerimientos del sistema, es necesario que los ejes X e Y estén alineados con el dorso de la mano, es decir, debe dibujar un eje de coordenadas. De esta manera el eje X se alinea a los dedos pulgar e índice, el eje Y al dedo medio, y el eje Z corresponde al movimiento de ambos ejes. Por lo tanto, la orientación del sensor será colocado verticalmente con el eje X, paralelo al dedo medio, y el eje Y paralelo al pulgar e índice, la posición del sensor se observa en la figura 27.

Para el proceso de lectura se realizaron cálculos utilizando valores específicos de acuerdo con la hoja de datos técnicos. El valor de 16384 se utilizó para normalizar el rango de lecturas de la aceleración en los ejes X, Y y Z. Esto se debe a que el rango completo de lecturas de aceleración del sensor MPU6050 es de  $\pm 16384$  unidades [33], y al dividir por este valor, se obtienen valores normalizados en el rango de -1.0 y 1.0, que representan la aceleración relativa en cada eje.

Para las lecturas del giroscopio en los ejes X, Y y Z, se dividió para 131. Debido a que el rango completo de lecturas del giroscopio del sensor MPU6050 es de  $\pm 131$  unidades por segundo [33], y al dividir por este valor, se obtienen valores normalizados en grados por segundo, que representan la velocidad angular relativa en cada eje. Estos valores normalizados se utilizaron para el cálculo del ángulo de Yaw. Los cálculos de

los ángulos de pitch, roll y Yaw se basaron en relaciones trigonométricas y en la relación entre las componentes de aceleración y velocidad angular. Las fórmulas emplearon la función atan2() para obtener los ángulos resultantes a partir de las componentes de aceleración en los ejes X, Y y Z, respectivamente en las ecuaciones 2, 3, y 4.

$$\text{Pitch} \rightarrow \theta = \arctan2(axg, \sqrt{ayg^2 + azg^2}) * \frac{180.0}{\pi} \quad \text{Ecuación 2}$$

$$\text{Roll} \rightarrow \theta = \arctan2(ayg, \sqrt{axg^2 + azg^2}) * \frac{180.0}{\pi} \quad \text{Ecuación 3}$$

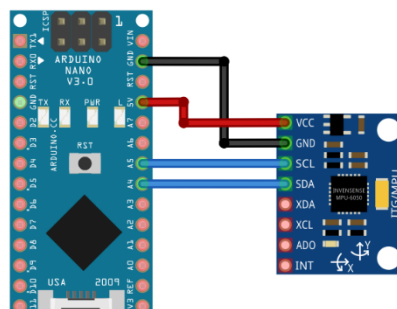
$$\text{Yaw} \rightarrow \theta = \arctan2(-gyds, gzds) * \frac{180.0}{\pi} \quad \text{Ecuación 4}$$

Donde  $axg$ ,  $gyds$  y  $gzds$  representaron la aceleración normalizada en el eje x y las velocidades angulares normalizadas en el eje Y y Z, respectivamente.

Estos cálculos permitieron obtener los ángulos necesarios para describir la orientación espacial del sensor MPU6050. Y, por lo tanto, cumple con los requerimientos del sistema al registrar los valores de los gestos dinámicos, dado que los movimientos de extensión, abducción cubital y radial pueden ser capturados únicamente con la aceleración y giros del sensor.

### Diagrama de conexión del sensor MPU-6050

El diagrama de conexión del circuito del sensor MPU-6050 se muestra en la figura 23.

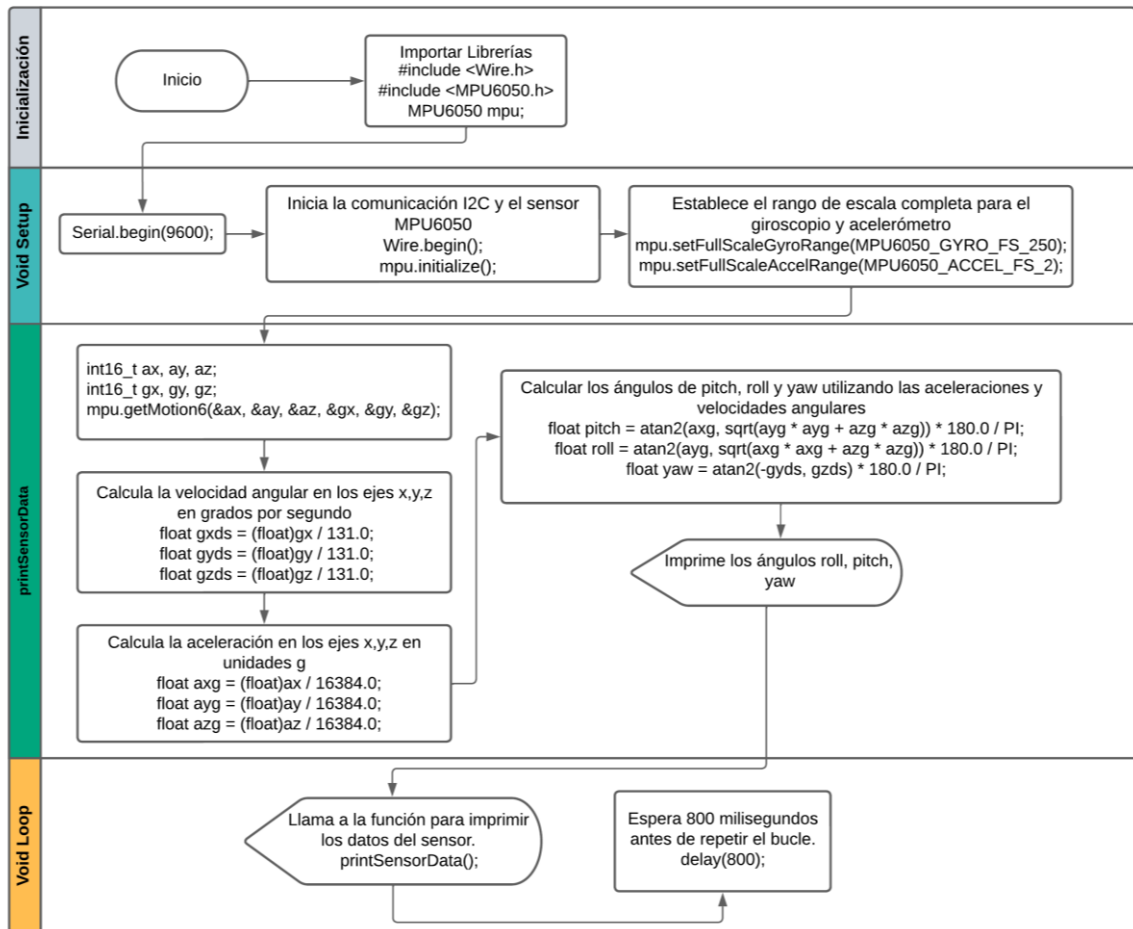


*Figura 23 Diagrama de conexión del Sensor MPU-6050 al Arduino Nano*

*Fuente: Investigador*

La programación en Arduino se observa en la figura 24. Se inició importando las librerías Wire.h y MPU6050, dado que utiliza comunicación I2C. A continuación, en

el Void Setup se inicializo la comunicación serial y se estableció el rango de escala completa del giroscopio y acelerómetro. Se creo una función “printSensorData”, para el cálculo de la aceleración y velocidad angular en los ejes x, y, z, los cuales permiten calcular los ángulos pitch, roll y yaw a través de la función atan2. Finalmente, en el Void Loop se llama a esta función con un delay de 800, permitiendo de esta manera una mejor sincronización y visualización de los datos.



**Figura 24** Diagrama de flujo del sensor MPU-6050

*Fuente: Investigador*

Dada la orientación vertical del sensor, como se observa en la figura 27. Los valores mínimos y máximos de movimiento censados respecto a los gestos seleccionadas del sistema se muestran en la tabla 15. Al igual que los sensores Flex, se consideró una escala de 5 lecturas.

**Tabla 15** Ángulos mínimos y máximos del sensor MPU-6050

| Sensor MPU-6050 |          |           |          |
|-----------------|----------|-----------|----------|
|                 | roll (X) | pitch (Y) | yaw (Z)  |
| Min             | -67,4    | -86,26    | -179,64  |
| 25%             | 6,3875   | -16,5725  | 10,695   |
| 50%             | 11,715   | -7,27     | 77,985   |
| 75%             | 18,96    | 7,5375    | 101,2625 |
| Max             | 86,92    | 86,21     | 179,62   |

Fuente: Investigador

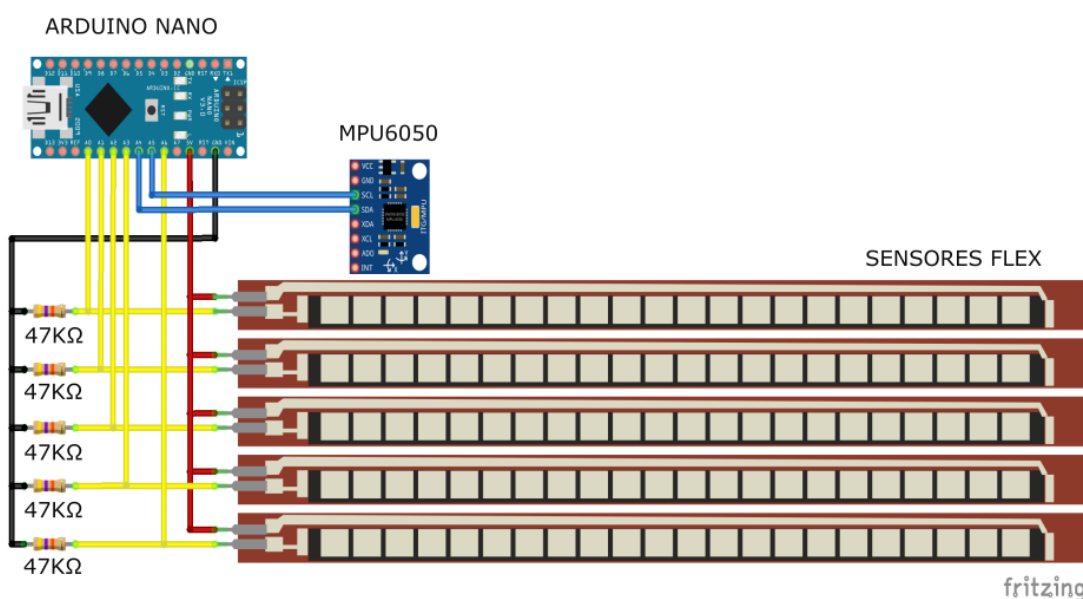
### 3.2.6 Diagrama del sistema electrónico del guante

El diagrama de conexión del circuito electrónico que contiene la placa de desarrollo Arduino NANO, los divisores de tensión y el sensor MPU-6050 con sus respectivas conexiones se muestra en la figura 25, junto con la asignación de pines en la tabla 16.

**Tabla 16** Asignación de pines del sistema electrónico del guante

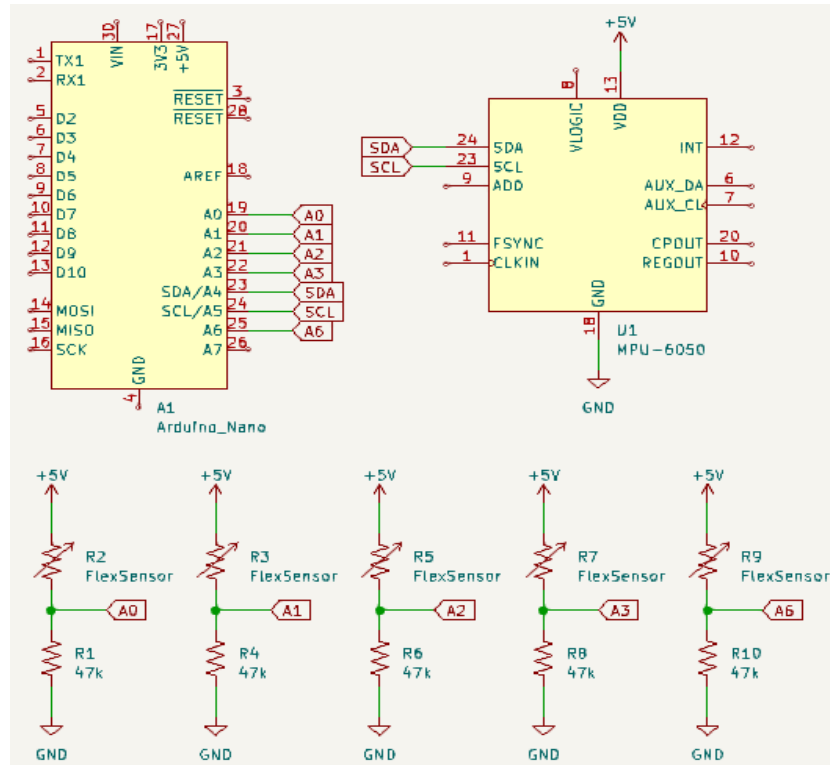
| Componente            | Pin sensor | Pin Arduino NANO   | Comunicación | Dirección I2C           |
|-----------------------|------------|--------------------|--------------|-------------------------|
| <b>Sensores Flex</b>  | P2         | A0, A1, A2, A3, A6 | Analógica    | -                       |
|                       | P1         | 5V                 |              |                         |
| <b>Sensor MPU6050</b> | VCC        | 5V                 | I2C          | Predeterminad a 0x68HEX |
|                       | GND        | GND                |              |                         |
|                       | SCL        | A5                 |              |                         |
|                       | SDA        | A4                 |              |                         |

Fuente: Investigador



**Figura 25** Diagrama de conexiones del guante electrónico

Fuente: Investigador



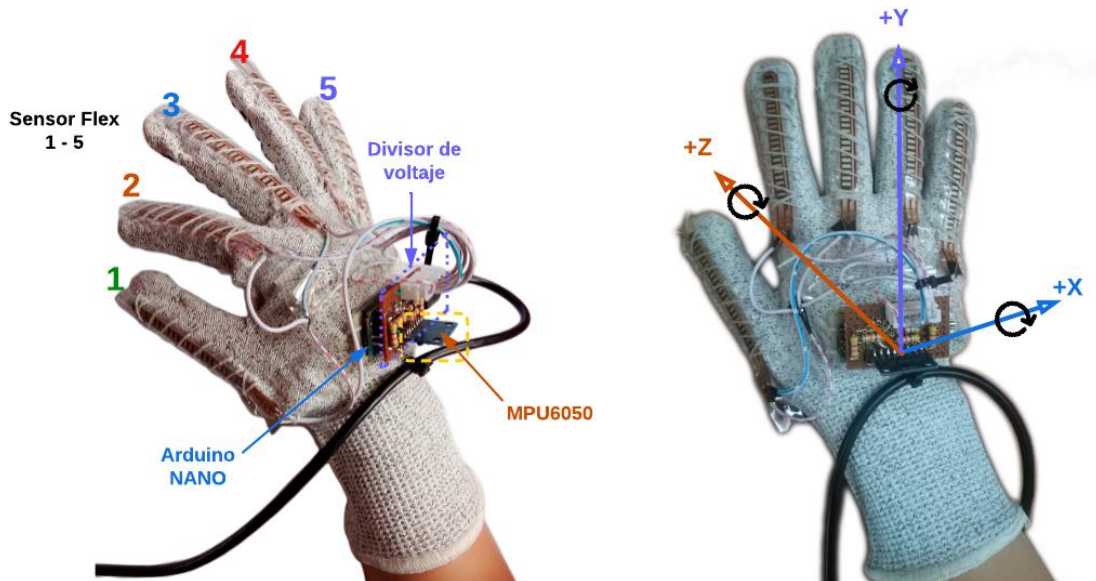
**Figura 26** Diagrama esquemático del guante electrónico

*Fuente: Investigador*

### 3.2.6.1 Construcción del guante electrónico

De acuerdo con el análisis establecido en la sección 3.2.1.1, se estableció trabajar con un guante talla M, con las dimensiones estándar de 20cm de largo x 21,5 cm de circunferencia de la mano, permitiendo ser usado por jóvenes y adultos, sin ningún inconveniente en cuanto a posición de los sensores.

El material seleccionado es Nylon, al ser un material dieléctrico permite evitar cortocircuitos. Los sensores Flex fueron colocados mediante tiras plásticas transparentes las cuales fueron cocidas al guante, de esta manera al flexionar los dedos el sensor este acoplado al movimiento de estos, de igual manera para sujetar la placa Arduino NANO se procedió a coser una baquelita perforada al dorso del guante, y encima de la placa Arduino se colocó el circuito electrónico con los divisores de voltaje y la conexión del sensor MPU-6050, de esta manera el circuito se mueve a la par de la mano. El diseño final del prototipo del guante electrónico se observa en la figura 27.

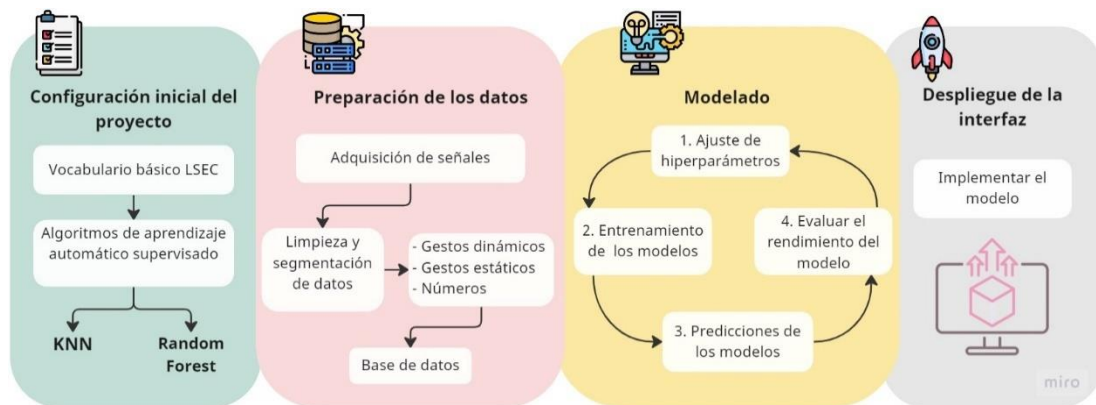


*Figura 27 Prototipo guante electrónico*

*Fuente: Investigador*

### 3.2.7 Desarrollo del sistema

El desarrollo del sistema automático para traducir la lengua de señas ecuatoriana se realizó en base a la figura 28, la cual detalla los pasos para seleccionar y trabajar con los algoritmos de aprendizaje automático del proyecto.



*Figura 28 Proceso de entrenamiento del aprendizaje automático*

*Fuente: Investigador basado en [20]*

### 3.2.8 Configuración inicial del sistema

Para comprender los objetivos del proyecto y su solución se llevó a cabo la investigación previa detallada en la sección 3.2.1. De acuerdo con la investigación se seleccionó traducir el alfabeto dactilológico ecuatoriano, los primeros 10 números y 10 palabras básicas con la finalidad de establecer un vocabulario básico. De esta



manera el sistema cumple con la traducción y aprendizaje tanto para personas sordas como oyentes al presentar en una interfaz gráfica las señas realizadas tanto en imágenes, texto y audio.

### 3.2.8.1 Selección del algoritmo de aprendizaje automático

Dada la naturaleza de los datos, el aprendizaje automático del proyecto es supervisado del tipo clasificación, este tipo de aprendizaje permite predecir un resultado determinado a partir de una entrada dada, lo que permite cumplir con las traducciones de la lengua de señas. De esta manera los algoritmos seleccionados son k-Nearest Neighbors y Random Forests, las razones se encuentran enlistadas en la tabla 17 de acuerdo con las secciones 1.4.10 y 1.4.11.

*Tabla 17 Selección del algoritmo de aprendizaje automático [20]*

---

| <b>k-Nearest Neighbors</b>  |
|---|
| <ul style="list-style-type: none"><li>• Adaptable a nuevos datos sin necesidad de entrenamiento previo, es capaz de realizar predicciones sin requerir un entrenamiento previo exhaustivo.</li><li>• KNN tiene la capacidad de detectar e identificar patrones en los datos de entrada.</li><li>• Permite ajustar características individuales de los datos según el usuario. Esto significa que el modelo puede adaptarse a las particularidades y preferencias de cada usuario</li><li>• Aprende a partir de ejemplos existentes, lo que le permite capturar la variabilidad de las señas en función de los datos de entrenamiento.</li></ul> |
| <b>Random Forests</b>   |
| <ul style="list-style-type: none"><li>• Es capaz de manejar conjuntos de datos grandes y complejos.</li><li>• Es versátil y puede manejar problemas de clasificación con múltiples clases. De acuerdo con el proyecto donde existen diferentes señas, números y palabras, esta capacidad es crucial para una traducción precisa.</li><li>• Puede adaptarse tanto a tareas de clasificación como de regresión. Lo que permite trabajar en tiempo real.</li><li>• Random Forest es robusto frente a datos ruidosos y puede evitar el sobreajuste.</li></ul>   |

---

Establecido los algoritmos se empezó con la importación de los paquetes y funciones necesarias para la programación en Python. En la tabla 18 se describe las funciones de las librerías empleadas.

**Tabla 18** Librerías empleadas para el sistema

| Librerías   | Descripción   |
|---|---|
| Pandas y numpy  | Se utilizan para el procesamiento y la manipulación de datos.   |
| RandomForestClassifier  | Es un algoritmo de aprendizaje automático utilizado para tareas de clasificación y se importa desde el módulo "sklearn.ensemble"        |
| precision_score, confusion_matrix, replace_score y ConfusionMatrixDisplay | Son métricas que se utilizan para evaluar el rendimiento de un modelo de clasificación y se importan desde el módulo "sklearn.metrics". |
| RandomizedSearchCV y train_test_split                                     | Se utilizan para ajustar hiperparámetros y dividir los datos en conjuntos de entrenamiento y prueba, respectivamente.                   |
| randint es una función del módulo "scipy.stats"                           | Genera números enteros aleatorios dentro de un rango específico.  |
| export_graphviz   | Es una función del módulo "sklearn.tree". Exporta un árbol de decisiones en formato Graphviz.   |
| Image y graphviz  | Se utilizan para visualizar el árbol de decisiones en Jupyter Notebook.   |

**Fuente:** Investigador basado en [20]

```
# Data Processing
import pandas as pd
import numpy as np

# Modelling
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, ConfusionMatrixDisplay
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from scipy.stats import randint

# Tree Visualisation
from sklearn.tree import export_graphviz
from IPython.display import Image
import graphviz
```

**Figura 29** Librerías empleadas para el sistema

**Fuente:** Investigador

### 3.2.9 Preparación de datos

#### 3.2.9.1 Adquisición de señales

Para la adquisición de señales se creó una Script en Python, el cual con la ayuda de la librería “PySerial” se envió la información de la placa Arduino Nano a través del puerto serial. De esta manera se empezó a registrar y guardar las señales en un archivo .csv, el cual es el formato utilizado para la base de datos. El sistema consta de 50 gestos, 30 del alfabeto, 10 de los números y 10 de las palabras básicas. Por lo tanto, para tener una buena base de datos se empezó a registrar 25 muestras por seña, teniendo al final 1250 datos de un usuario, los cuales son suficientes para preparar datos de entrenamiento y prueba al momento de desarrollar los algoritmos de aprendizaje automático.

Dado que algunas señas poseen ciertas similitudes físicas en cuanto a posición de la mano y de los sensores, se optó por categorizar los registros de la base de datos de acuerdo con los gestos estáticos, gestos dinámicos y los números. Este último tuvo su propia categoría debido a que los números 2, 4, 6 y 9 son exactamente las letras V, B, W y F respectivamente, y la única manera para traducirlos es en base a un contexto en la conversación, por lo tanto, la solución fue darle su propia categoría. Cabe mencionar que, al realizar un gesto exacto en múltiples ocasiones, es poco probable que arroje los mismos valores, dada la naturaleza de los sensores, por este motivo, se estableció una tolerancia de error del  $\pm 15\%$  para compensar estas variaciones. A continuación, las tablas 19,20 y 21 muestran cómo está estructurado la base de datos para cada categoría.

Los datos registrados están organizados en 10 columnas, 5 pertenecen a los sensores flexibles con los nombres de cada dedo, 3 pertenecen a los ángulos X, Y y Z del sensor MPU6050, y por último se tiene la salida con la seña correspondiente.

#### Conjunto de datos de los gestos estáticos

*Tabla 19* Conjunto de datos de los gestos estáticos

| <b>Pulgar</b> | <b>Índice</b> | <b>Medio</b> | <b>Anular</b> | <b>Meñique</b> | <b>Roll</b> | <b>Pitch</b> | <b>Yaw</b> | <b>Salida</b> |
|---------------|---------------|--------------|---------------|----------------|-------------|--------------|------------|---------------|
| 625           | 308           | 399          | 457           | 392            | 4.10        | -1.33        | 87.40      | A             |
| 418           | 458           | 597          | 652           | 599            | 6.25        | -22.21       | 75.76      | B             |
| 462           | 302           | 432          | 476           | 495            | 10.10       | -2.83        | 87.87      | C             |

|     |     |     |     |     |       |        |        |    |
|-----|-----|-----|-----|-----|-------|--------|--------|----|
| 664 | 464 | 613 | 469 | 401 | 79.03 | -10.73 | 63.98  | CH |
| 412 | 467 | 446 | 455 | 435 | 9.22  | -7.92  | -74.81 | D  |
| 350 | 277 | 378 | 442 | 405 | 9.43  | -7.54  | 96.87  | E  |
| 393 | 299 | 624 | 635 | 603 | 6.24  | -10.33 | 64.44  | F  |
| 606 | 477 | 396 | 432 | 385 | 84.54 | -2.33  | 82.87  | G  |
| 578 | 452 | 613 | 470 | 405 | 81.47 | -7.58  | 82.61  | H  |
| 474 | 348 | 509 | 468 | 610 | 4.74  | -25.75 | 92.86  | I  |
| 636 | 474 | 603 | 461 | 386 | 6.39  | -22.88 | 80.98  | K  |
| 672 | 480 | 383 | 426 | 380 | 6.45  | 0.93   | 74.31  | L  |
| 447 | 387 | 562 | 580 | 408 | 2.64  | 72.54  | 81.39  | M  |
| 454 | 395 | 528 | 476 | 417 | 11.92 | 77.71  | 96.88  | N  |
| 410 | 313 | 437 | 430 | 424 | 9.51  | 2.91   | 65.13  | O  |
| 607 | 493 | 547 | 489 | 429 | 12.15 | 71.46  | 41.89  | P  |
| 660 | 424 | 425 | 439 | 447 | 10.87 | 73.01  | 83.40  | Q  |
| 453 | 490 | 584 | 419 | 400 | 7.44  | -17.80 | 96.19  | R  |
| 440 | 406 | 476 | 430 | 414 | 15.25 | -10.06 | -30.38 | S  |
| 416 | 268 | 599 | 658 | 619 | 3.32  | -1.88  | 78.58  | T  |
| 443 | 462 | 606 | 435 | 403 | 11.77 | -21.95 | 71.08  | U  |
| 458 | 446 | 594 | 411 | 411 | 10.67 | -14.00 | 84.43  | V  |
| 480 | 462 | 611 | 668 | 431 | 14.72 | -22.14 | 91.23  | W  |
| 434 | 306 | 402 | 449 | 393 | 15.34 | 0.63   | 101.71 | X  |
| 644 | 331 | 503 | 488 | 601 | 12.84 | -7.61  | 69.44  | Y  |

Fuente: Investigador

### Conjunto de datos de los gestos dinámicos

Tabla 20 Conjunto de datos de los gestos dinámicos

| Pulgar | Índice | Medio | Anular | Meñique | Roll   | Pitch  | Yaw     | Salida      |
|--------|--------|-------|--------|---------|--------|--------|---------|-------------|
| 402    | 312    | 457   | 408    | 623     | 45.62  | -42.97 | 78.04   | J           |
| 654    | 479    | 392   | 421    | 398     | 4.79   | 5.25   | -165.76 | LL          |
| 460    | 400    | 528   | 484    | 421     | 24.57  | 64.23  | -177.84 | Ñ           |
| 441    | 505    | 591   | 423    | 375     | 11.10  | -19.35 | 151.76  | RR          |
| 458    | 450    | 396   | 429    | 389     | 14.51  | 62.72  | 49.20   | Z           |
| 643    | 450    | 623   | 653    | 609     | -34.70 | 17.62  | 67.51   | AMARILLO    |
| 656    | 284    | 383   | 449    | 420     | 21.50  | 16.37  | -22.92  | AZUL        |
| 671    | 459    | 611   | 658    | 606     | 26.63  | -46.57 | -46.55  | BUENOS DÍAS |
| 636    | 390    | 491   | 501    | 525     | 2.23   | -78.60 | 93.18   | BLANCO      |
| 639    | 432    | 605   | 646    | 598     | 26.89  | -30.44 | -92.89  | CHAO        |
| 652    | 480    | 636   | 670    | 612     | 12.94  | -73.53 | -149.64 | GRACIAS     |
| 635    | 265    | 351   | 390    | 339     | -13.81 | -9.89  | 92.75   | HOLA        |
| 665    | 319    | 410   | 437    | 393     | -49.15 | 32.89  | -164.72 | MAL         |
| 672    | 446    | 612   | 649    | 605     | 7.75   | 0.20   | 67.35   | NEGRO       |
| 660    | 289    | 389   | 429    | 376     | 76.09  | -2.41  | 134.67  | BIEN        |

Fuente: Investigador

## Conjunto de datos de los Números

*Tabla 21* Conjunto de datos de los números

| <b>Pulgar</b> | <b>Índice</b> | <b>Medio</b> | <b>Anular</b> | <b>Meñique</b> | <b>Roll</b> | <b>Pitch</b> | <b>Yaw</b> | <b>salida</b> |
|---------------|---------------|--------------|---------------|----------------|-------------|--------------|------------|---------------|
| 449           | 450           | 395          | 455           | 374            | 14.48       | -13.52       | 93.92      | 1             |
| 441           | 456           | 613          | 426           | 408            | 19.42       | -12.58       | 82.87      | 2             |
| 659           | 480           | 608          | 463           | 396            | 12.76       | -14.27       | 38.23      | 3             |
| 393           | 459           | 595          | 637           | 601            | 13.53       | -9.26        | 90.00      | 4             |
| 653           | 466           | 615          | 654           | 607            | 13.37       | -9.38        | 90.00      | 5             |
| 468           | 464           | 613          | 659           | 333            | 15.87       | -32.52       | 72.69      | 6             |
| 448           | 446           | 580          | 400           | 571            | 16.40       | -22.51       | 110.16     | 7             |
| 451           | 429           | 394          | 625           | 598            | 16.47       | -14.53       | 74.08      | 8             |
| 440           | 276           | 621          | 632           | 595            | 10.13       | -8.18        | 83.66      | 9             |
| 660           | 299           | 398          | 477           | 410            | 60.70       | -25.59       | -158.62    | 10            |

Fuente: Investigador

### Posición de reposo

Para que el sistema no realice una traducción continua se estableció entrenar una posición como “reposo”, donde el usuario podrá descansar la mano y el sistema detendrá la lectura hasta que el usuario realice el siguiente gesto. La posición seleccionada es la palma abierta hacia abajo, de esta manera el usuario podrá apoyar la palma en una superficie, evitando errores de traducción y generando una traducción continua sin que exista lecturas intermedias.

Para que el sistema reconozca esta posición se capturo las muestras en cada categoría, dado que son 50 gestos, se aumenta 1 gesto en cada uno, por lo que de manera general se tendrá 1325 datos registrados, y de acuerdo con las pruebas de funcionamiento se recopiló muestras para 5 usuarios. De esta manera el sistema cuenta con 1375 datos para los números, 2000 datos para los gestos dinámicos y 3250 para los gestos estáticos, logrando al final tener una base de datos de 6625 registros.

### 3.2.9.2 Limpieza de datos

Creada la base de datos se procedió a identificar el rango de los valores mínimos y máximos de los sensores mediante la siguiente tabla 22, en la cual muestra si los datos están dentro de los rangos establecidos en la sección 3.2.7. De esta manera se asegura que no exista datos fuera de rango implicando algún sensor desconectado o dañado, además de asegurar que el registro sea de datos numéricos.

**Tabla 22** Valores mínimos y máximos de los sensores Flex y MPU6050

| Valores mínimos y máximos de los sensores Flex y MPU6050 |               |               |              |               |                |                 |                  |                |
|--|---------------|---------------|--------------|---------------|----------------|-----------------|------------------|----------------|
|  | <i>Pulgar</i> | <i>Índice</i> | <i>Medio</i> | <i>Anular</i> | <i>Meñique</i> | <i>roll (X)</i> | <i>pitch (Y)</i> | <i>yaw (Z)</i> |
| <b>Min</b>   | 334           | 233           | 313          | 384           | 326            | -67,4           | -86,26           | -179,64        |
| <b>25%</b>   | 448           | 313           | 420          | 432           | 390            | 6,3875          | -16,5725         | 10,695         |
| <b>50%</b>   | 560           | 434           | 527          | 466           | 415            | 11,715          | -7,27            | 77,985         |
| <b>75%</b>   | 654           | 465           | 608          | 644           | 593            | 18,96           | 7,5375           | 101,2625       |
| <b>Max</b>   | 682           | 522           | 651          | 681           | 638            | 86,92           | 86,21            | 179,62         |

Fuente: Investigador

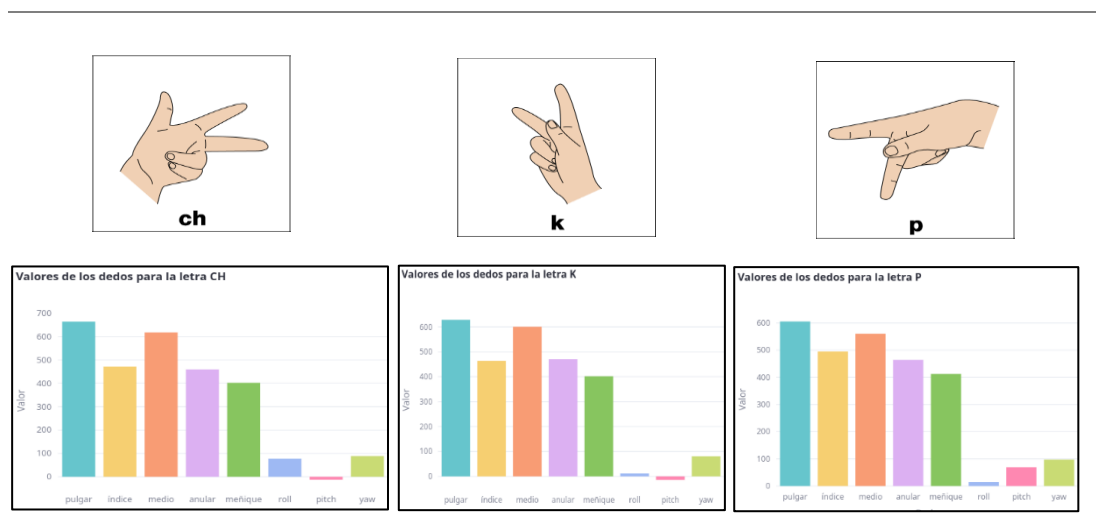
### Análisis de similitudes físicas del alfabeto dactilológico ecuatoriano

A pesar de que la base de datos este categorizado para evitar conflictos, es necesario analizar ciertas similitudes físicas en cuanto al alfabeto, debido o que es importante determinar ciertas características en los datos, para posteriormente facilitar el entrenamiento de los modelos. A continuación, se muestra como ejemplo el análisis de las letras CH, K y P, el análisis completo de las demás letras del alfabeto se detalla en la sección de anexos.

- **Letras CH, K y P**

El factor que permite diferenciar la letra CH de la K y P es la inclinación de la muñeca, como se aprecia en la tabla 23, los ángulos roll (X) y yaw (Z) presentan una inclinación hacia el eje horizontal mayor de 80°. Por otra parte, la letra P presenta una inclinación mayor en el eje Y (Pitch), de esta manera cada letra puede ser diferenciada sin dificultad gracias al sensor MPU-6050.

**Tabla 23** Análisis letras CH, K, P



Fuente: Investigador

En la tabla 24 se presenta las características específicas de las diferentes categorías, con la finalidad de evitar conflictos al momento de aplicar el entrenamiento de los modelos.

*Tabla 24 Características específicas de los gestos estáticos y dinámicos*

| <b>Gestos Estáticos</b>    |  |
|----------------------------|--|
| <i>Letras</i>              | <i>Características específicas</i>   |
| <i>C y O</i>               | Nivel de curvatura de los dedos índice, medio, anular y meñique es menor para la letra O |
| <i>CH, K, P</i>            | Orientación angular e inclinación de la muñeca   |
| <i>F y T</i>               | Grado de flexión del dedo índice y pulgar  |
| <i>G, L, Q</i>             | Orientación angular e inclinación de la muñeca   |
| <i>H, U, V</i>             | Orientación angular e inclinación de la muñeca   |
| <b>Gestos Dinámicos</b>    |  |
| <i>Hola, Chao</i>          | Movimiento de abducción radial y cubital (Inclinación de la muñeca)                      |
| <i>Bien, Mal</i>           | Orientación angular  |
| <i>Buenos días, blanco</i> | Orientación angular  |

Fuente: Investigador

### 3.2.10 Modelado

#### 3.2.10.1 Análisis y selección del ajuste de hiperparámetro

Los hiperparámetros son configuraciones que no se aprenden directamente del conjunto de datos, sino que se establecen antes del proceso de entrenamiento del modelo. Cada algoritmo de Machine Learning tiene sus propios conjuntos de hiperparámetros que controlan aspectos específicos del modelo, como la complejidad, la regularización, la velocidad de aprendizaje, el número de vecinos en un algoritmo KNN, el número de árboles en un bosque aleatorio, entre otros. [20] Estos hiperparámetros influyen en el rendimiento, la capacidad de generalización y la eficiencia del modelo. Por lo tanto, para seleccionar la mejor técnica se analizó la siguiente tabla 25.

**Tabla 25** Técnicas para el ajuste de hiperparámetros [20]

| <b>Técnica</b>  | <b>Ventajas</b>  | <b>Desventajas</b>  |
|---|--|---|
| <i>Grid Search</i>  | Explora todo el espacio de búsqueda.   | Puede ser computacionalmente costoso si el espacio de búsqueda es grande.   |
| <i>Random Search</i>  | Eficiente cuando el espacio de búsqueda es grande.                                     | No garantiza explorar todo el espacio de búsqueda.  |
| <i>Búsqueda Bayesiana</i>   | Búsqueda inteligente y eficiente.  | Requiere más tiempo de configuración y ajuste para definir la distribución inicial y el método de actualización de la distribución. |
| <i>Optimización basada en algoritmos genéticos</i>                          | Explora el espacio de búsqueda de manera no determinista y adaptable.                  | Puede requerir más tiempo y recursos computacionales para converger hacia los mejores hiperparámetros.                              |
| <i>Ajuste manual</i>  | Permite un control total sobre los hiperparámetros y una comprensión intuitiva.        | Requiere experiencia y conocimiento específico sobre los hiperparámetros y su impacto en el modelo.                                 |
| <i>Preprocesamiento de datos</i>  | Mejora la calidad de los datos de entrada y, por lo tanto, el rendimiento del modelo.  | Puede requerir conocimiento específico del dominio y tiempo adicional de preprocesamiento antes de ajustar los hiperparámetros.     |
| <i>Selección de características</i>   | Reduce la dimensionalidad y el ruido en los datos.                                     | Puede requerir algoritmos y métodos adicionales para seleccionar características óptimas.   |
| <i>Ajuste automático con bibliotecas (por ejemplo, Keras-Tuner, Optuna)</i> | Ofrece una implementación eficiente y fácil de usar para el ajuste de hiperparámetros. | Puede requerir una curva de aprendizaje para familiarizarse con la biblioteca específica y su configuración.                        |

**Fuente:** Investigador

La técnica seleccionada es el ajuste manual, dado que el proyecto utiliza los algoritmos de KNN y Random Forest, el ajuste manual ofrece un mejor control al momento de optimizar el rendimiento. Permitiendo una mejor adaptabilidad y flexibilidad, lo que es beneficioso en situaciones específicas donde se requiere un enfoque personalizado y un análisis detallado de los hiperparámetros. Además, el ajuste manual es ligero en



términos de carga computacional, lo que lo hace más adecuado cuando se cuenta con recursos limitados.

### 3.2.10.2 Entrenamiento de los modelos

Al momento de entrenar los modelos para el sistema automático de traducción de lengua de señas, fue necesario dividir los datos de entrenamiento y prueba. En la cual se estableció usar el 80% de los datos para entrenamiento y el 20% para prueba.

Los datos de entrenamiento son usados para aprender la relación entre las características y el objetivo. Por otra parte, los datos de prueba se utilizan para evaluar el rendimiento del modelo. En la figura 30 se observa la división de los datos en variables separadas para las características y el objetivo, para ser divididos en datos de entrenamiento y prueba.

```
# Divide los datos en características (X) y objetivo (y)
X = bank_data.drop('y', axis=1)
y = bank_data['y']
# Divide los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)
```

*Figura 30* División de los datos de entrenamiento y prueba

*Fuente:* Investigador

Se toma cuatro argumentos, las características (X), la variable objetivo (y), el tamaño de la prueba de 0.2 (significa que se usa el 20% de los datos) y un estado aleatorio. De esta manera la función devuelve cuatro variables, X\_train es el conjunto de funciones de entrenamiento, X\_test es el conjunto de funciones de prueba, y\_train es el conjunto de entrenamiento de la variable objetivo y y\_test el conjunto de prueba de la variable objetivo.

Establecidos los conjuntos de datos de entrenamiento y prueba, se realizó la evaluación de los modelos KNN y Random Forest.

### 3.2.10.3 Entrenamiento del modelo k-Nearest Neighbors (KNN)

Se creó una instancia del modelo KNN, luego se ajustó a los datos de entrenamiento, para enviar tanto las características como la variable de destino, para que el modelo pueda aprender, como se observa en la figura 31.

```
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
```

*Figura 31 Modelo del clasificador K-Nearest Neighbors*

*Fuente: Investigador*

Se empleó el modelo `KNeighborsClassifier()` de la librería `Scikit-learn` para entrenar el algoritmo KNN, como prueba inicial se estableció el parámetro de `n_neighbors` en 3 vecinos. Esto significa que el clasificador considera los 3 vecinos más cercanos al hacer predicciones. La línea `knn.fit(X_train, y_train)` ajusta el clasificador a los datos de entrenamiento “X\_train” y “y\_train”. De esta manera el clasificador KNN aprende de los datos de entrenamiento y puede hacer predicciones sobre nuevos datos basados en lo que ha aprendido.

### Predicciones del modelo KNN

En este punto, se tiene entrenado el modelo KNN, a continuación, se evaluó si está haciendo predicciones precisas.

```
# K-Nearest Neighbors
y_pred = knn.predict(X_test)
```

*Figura 32 Predicción del modelo KNN*

*Fuente: Investigador*

Se hace uso de los métodos de predicción del objeto `KNeighborsClassifier (knn)` para hacer predicciones sobre un conjunto de datos de prueba (`X_test`). Luego los valores pronosticados se almacenan en la variable `y_pred`. En este paso es donde el modelo se entrena en un conjunto de datos de entrenamiento para posteriormente ser usado para predicciones sobre datos nuevos.

### Evaluación del rendimiento de KNN

Para la evaluación del rendimiento se empleó el ajuste manual, para el modelo de KNN se evaluó el parámetro “n\_neighbors” el cual determina la cantidad de vecinos más

cercanos que se consideran al realizar una predicción. Cuanto mayor sea el valor para "n\_neighbors", más vecinos se toman en cuenta, por lo tanto, se debe considerar lo siguiente:

- **Overfitting (sobreajuste):** Si se elige un valor demasiado bajo, el modelo podría ser muy sensible a los puntos atípicos o ruido en los datos de entrenamiento. Esto puede resultar en un sobreajuste, donde el modelo se adapta demasiado a los datos de entrenamiento específicos y tiene dificultades para generalizar a nuevos datos.
- **Underfitting (subajuste):** Por otro lado, si se selecciona un valor demasiado alto, el modelo podría perder detalles y características importantes de los datos. Resultando en un subajuste, donde el modelo no captura de manera adecuada las relaciones y patrones presentes en los datos de entrenamiento.

Por lo tanto, usando el ajuste manual se realizaron pruebas para determinar el valor óptimo de "n\_neighbors". A través del parámetro "random\_state" para variar según la distribución de los datos y el tamaño del conjunto de entrenamiento, en la figura 33 se puede observar cómo cambian las métricas de precisión al variar los valores de "n\_neighbors" y "random\_state".

|             | Random_State |      |      |      |      |
|-------------|--------------|------|------|------|------|
| n_neighbors | 10           | 8    | 6    | 4    | 2    |
| 1           | 0.94         | 0.93 | 0.88 | 0.91 | 0.93 |
| 2           | 0.93         | 0.91 | 0.87 | 0.81 | 0.90 |
| 4           | 0.89         | 0.88 | 0.88 | 0.88 | 0.88 |
| 6           | 0.87         | 0.84 | 0.88 | 0.86 | 0.86 |
| 8           | 0.83         | 0.83 | 0.84 | 0.83 | 0.83 |
| 10          | 0.82         | 0.81 | 0.83 | 0.81 | 0.83 |

*Figura 33 Métricas de precisión del modelo K-NN*

*Fuente: Investigador*

Al aumentar el valor de "n\_neighbors" de 1 a 10, se puede observar una tendencia general de que la precisión disminuye gradualmente. Esto puede deberse a que, al considerar un mayor número de vecinos, el modelo se vuelve más conservador y tiende a realizar predicciones más generales. Por otro lado el término de "random\_state", los resultados no parecen seguir un patrón claro en relación con la precisión. Por lo que se

puede concluir que no tiene un impacto lineal y predecible en la precisión del modelo. En general, los valores de precisión obtenidos son bastante buenos, oscilando entre 0.81 y 0.94. Donde un accuracy del 0.94 (94%) es un resultado prometedor y generalmente se considera bastante alto. Significa que el 94% de las predicciones realizadas por el modelo de K-Nearest Neighbors (KNN) son correctas en el conjunto de datos utilizado para evaluar el modelo.

#### 3.2.10.4 Entrenamiento del modelo Random Forest

Así mismo, primero se creó una instancia con los parámetros predeterminados, para ser ajustados a los datos de entrenamiento, como se observa en la figura 34. RandomForestClassifier es un algoritmo de aprendizaje automático que crea un bosque de árboles de decisión y combina sus predicciones para generar una predicción final.

```
rf = RandomForestClassifier()  
rf.fit(X_train, y_train)
```

*Figura 34 Modelo del clasificador Random Forest*

*Fuente: Investigador*

Se creó una instancia de la clase RandomForestClassifier () y se la asignó a la variable rf. Luego, se ajustó el modelo a los datos de entrenamiento X\_train y y\_train utilizando el método fit(). El método fit() entrenó el modelo en los datos de entrada ajustando los parámetros de los árboles de decisión para minimizar el error entre los valores previstos y reales.

#### Predicciones del modelo RF

En la figura 35 se hace uso del método de predicción del objeto RandomForestClassifier (rf) sobre un conjunto de datos de prueba (X\_test). Luego los valores pronosticados se almacenan en la variable y\_pred.

```
# Random Forest  
y_pred = rf.predict(X_test)
```

*Figura 35 Predicciones del modelo Random Forest*

*Fuente: Investigador*

#### Evaluación del rendimiento de Random Forest

De igual manera al modelo de KNN se utilizó el ajuste manual, el código mostrado en la figura 36 calcula la precisión de las predicciones del modelo comparando los valores

predichos (`y_pred`) con los valores reales (`y_test`) y luego usando la función `precision_score` de la biblioteca `scikit-learn` para calcular la precisión.

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

*Figura 36 Evaluación de rendimiento del modelo Random Forest*

*Fuente: Investigador*

Para el cálculo se utilizó las siguientes métricas.

- **accuracy\_score:** Calcula la precisión global del modelo, es decir, la proporción de predicciones correctas en relación con el total de predicciones. Se calcula dividiendo el número de predicciones correctas entre el número total de predicciones.
- **precision\_score:** Calcula la proporción de verdaderos positivos (predicciones positivas correctas) en relación con el total de predicciones positivas. Es decir, mide la precisión del modelo al predecir una clase específica. Un valor alto de precisión indica que el modelo tiene una baja tasa de falsos positivos.
- **recall\_score:** Calcula la proporción de verdaderos positivos en relación con el total de ejemplos positivos en el conjunto de datos. Mide la exhaustividad o sensibilidad del modelo al capturar instancias de una clase específica. Un valor alto de recall indica que el modelo tiene una baja tasa de falsos negativos.

Todas estas métricas oscilan entre 0 y 1, donde un valor de 1 representa un rendimiento perfecto en la métrica correspondiente. En la figura 37 se observa la precisión resultante del algoritmo Random Forest.

|              | Random_State |      |      |      |      |
|--------------|--------------|------|------|------|------|
| n_estimators | 10           | 8    | 6    | 4    | 2    |
| 1000         | 0.96         | 0.96 | 0.95 | 0.95 | 0.95 |
| 500          | 0.95         | 0.95 | 0.95 | 0.94 | 0.95 |
| 250          | 0.96         | 0.96 | 0.94 | 0.95 | 0.95 |
| 100          | 0.96         | 0.95 | 0.93 | 0.95 | 0.94 |
| 50           | 0.95         | 0.95 | 0.94 | 0.92 | 0.95 |

*Figura 37 Métricas de precisión del modelo Random Forest*

*Fuente: Investigador*

En general, los valores de precisión se mantienen relativamente altos en un rango estrecho, alrededor de 0.92-0.96. manteniendo una precisión consistente, esto sugiere que el modelo está obteniendo un buen rendimiento, independientemente de los valores específicos de “n\_estimators” y “random\_state”.

Se puede observar cierta variabilidad en cuanto al valor de “random\_state”, debido a las muestras aleatorias durante el entrenamiento, sin embargo, esta variabilidad no parece tener un impacto significativo en la precisión general. Por otra parte, el número de estimadores “n\_estimators” no presenta un cambio drástico en la precisión, los valores más altos de “n\_estimators” (como 1000) tienden a tener una precisión ligeramente mayor, pero la diferencia es relativamente pequeña en comparación con los valores más bajos (como 10 o 50). Por ende, la precisión se mantiene estable y en un nivel alto en todo el rango de parámetros evaluados, lo que indica que los datos son bastante discernibles y que el modelo es capaz de aprender y generalizar. De modo que no es necesario realizar ajustes significativos en los hiperparámetros para obtener una precisión alta.

### Evaluación de los algoritmos de Aprendizaje Automático

*Tabla 26 Evaluación de los algoritmos K-NN y Random Forest*

| <b>Algoritmos</b> | <b>K-Nearest Neighbors</b>         | <b>Random Forest</b>                   |
|-------------------|------------------------------------|--|
| <i>Parámetros</i> | n_neighbors=1,<br>random_state =10 | n_estimators =100,<br>random_state =10 |
| <i>Accuracy:</i>  | 0.9321266968325792                 | 0.9565217391304348                     |
| <i>Precision:</i> | 0.9462992889463478                 | 0.9620271033314512                     |
| <i>Recall:</i>    | 0.9321266968325792                 | 0.9565217391304348                     |

**Fuente:** Investigador

En la tabla 26 se indican los valores de precisión calculados de los algoritmos de KNN y Random Forest de acuerdo con los datos de entrenamiento de las figuras 36 y 37 respectivamente. Determinando que el mejor algoritmo para traducir la lengua de señas ecuatoriano es el modelo Random Forest (Bosques Aleatorios), ya que obtuvo un Accuracy (Exactitud) de 0.956 y una precisión de 0.9620, superando al algoritmo KNN. Por lo tanto, este modelo es aplicado en el sistema.

### 3.2.11 Despliegue de la interfaz de usuario

Antes de comenzar con el despliegue es necesario crear un entorno virtual, para la instalación de todos los paquetes necesarios, dado que permite gestionar de mejor manera las dependencias del proyecto. Al manejar algoritmos de aprendizaje automático, los proyectos suelen requerir de numerosas bibliotecas y dependencias, cada una con sus propias versiones y requisitos, por lo tanto, la creación de un entorno virtual facilita, aísla y gestiona estas dependencias de manera eficiente, evitando conflictos entre ellas.

El sistema operativo en el que se implementa el sistema es GNU/Linux, para crear un entorno virtual se siguen los siguientes comandos presentados en la figura 38, junto con la instalación de Streamlit, los cuales deben ejecutarse en la terminal.

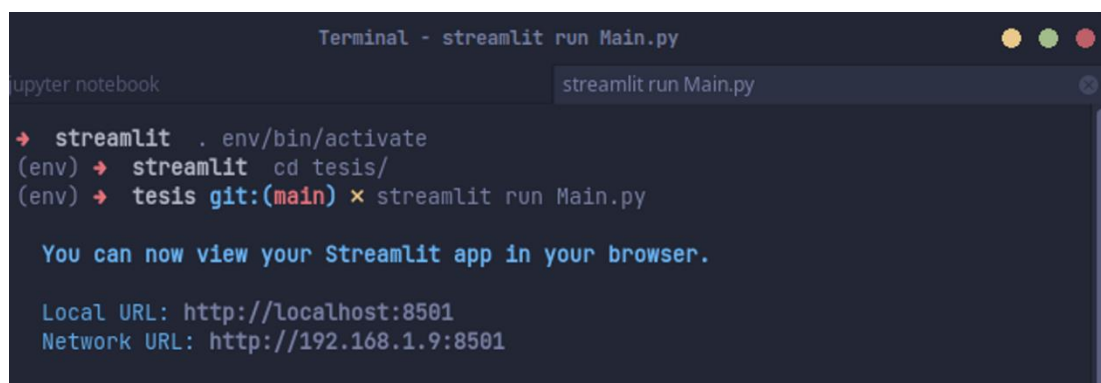
```
$ python3 -m pip install virtualenv
$ python3 -m venv nombre_del_entorno_virtual
$ source nombre_del_entorno_virtual/bin/actíivate

$ pip install streamlit
$ streamlit hello
```

**Figura 38** Instalación del Framework Streamlit

*Fuente: Investigador*

Para iniciar Streamlit se creó una script en Python llamada Main.py, dentro del entorno virtual. De esta manera se abre en el navegador el servicio de Streamlit de manera local, el cual se abre en el puerto 8501 de manera predeterminada, como se observa en la figura 39, la cual presenta una plantilla lista para usar.



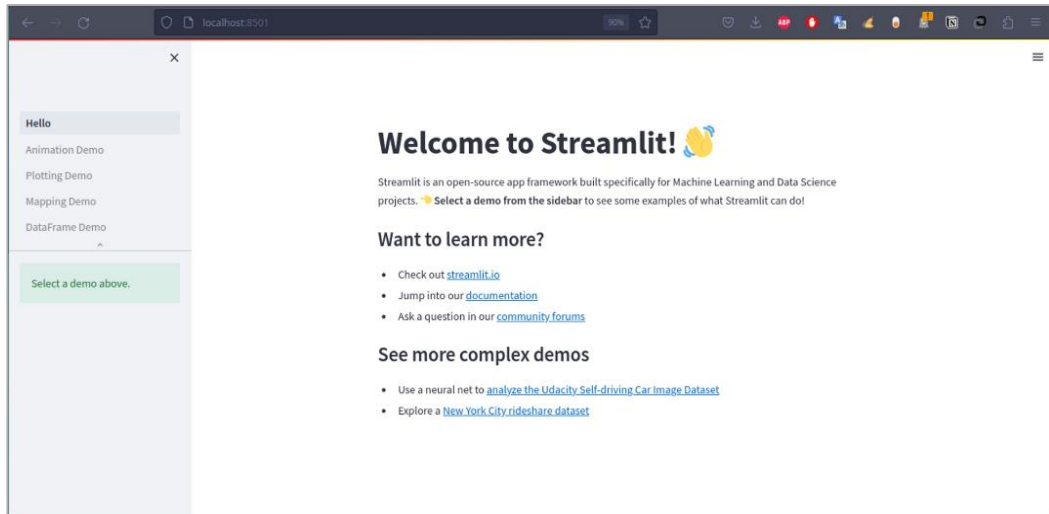
```
Terminal - streamlit run Main.py
upyter notebook
streamlit run Main.py
→ streamlit . env/bin/activate
(env) → streamlit cd tesis/
(env) → tesis git:(main) × streamlit run Main.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.9:8501
```

**Figura 39** Ejecución de Streamlit

*Fuente: Investigador*

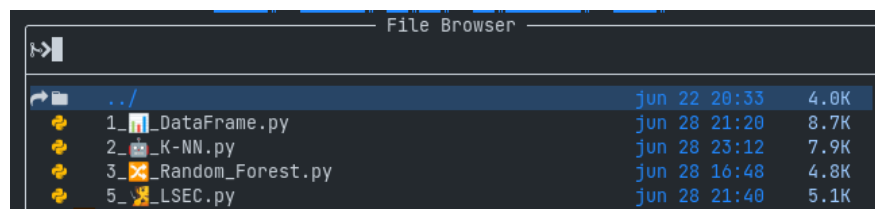


**Figura 40** Inicio de Streamlit

*Fuente: Investigador*

### 3.2.11.1 Creación página de inicio – Menú

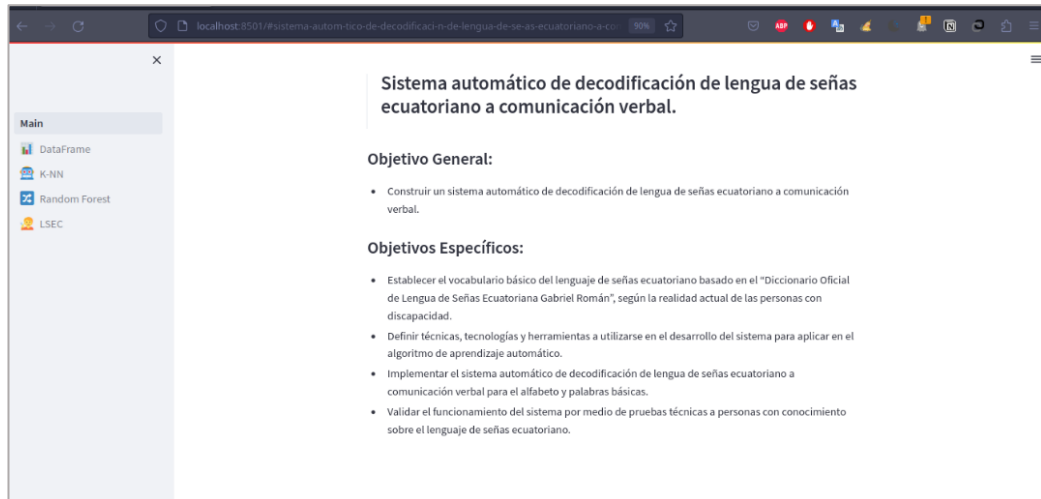
Para navegar en Streamlit es necesario establecer un archivo de punto de entrada, la primera página que la aplicación muestra al usuario, se creó el archivo “Main.py” el cual cumple como punto de entrada o también llamado menú. En la figura 42 se presenta la página principal, la cual detalla información del proyecto como el tema, objetivos y el vocabulario básico seleccionado. En la parte izquierda de la interfaz se crearon páginas los cuales describen el desarrollo del proyecto junto con la aplicación final donde el usuario puede traducir la lengua de señas ecuatoriano. Para agregar páginas se crean archivos .py en el directorio /pages del proyecto, como se muestra en la figura 41.



**Figura 41** Creación de las páginas del aplicativo web Streamlit

*Fuente: Investigador*





**Figura 42** Interfaz web - Menú Principal

*Fuente: Investigador*

### 3.2.11.2 Creación de la página Hoja de datos

Se creó la carpeta en el siguiente directorio pages/DataFrame.py. Permite visualizar el registro de los gestos del sistema para del Alfabeto, Números y Palabras, donde cada uno puede ser visualizado mediante imágenes que presenta la posición de la mano. Se crearon listas de los nombres en formato .PNG, de esta manera mediante un slider se puede navegar entre ellas, como se observa en la figura 43.

```
# Lista con los nombres de los archivos de imagen
img_alph = ['A', 'B', 'C', 'CH', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
'LL', 'M', 'N', 'Ñ', 'O', 'P', 'Q', 'R', 'RR', 'S', 'T', 'U', 'V', 'W', 'X', 'Y',
'Z']
img_num = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
img_words = ['Hola', 'Buenosdias', 'Bien', 'Mal', 'Gracias',
'Amarillo', 'Azul', 'Blanco', 'Negro', 'Chao']
```

**Figura 43** Lista de nombre de los archivos de imágenes



*Fuente: Investigador*

En la figura 44 se puede apreciar el código para seleccionar y conectar el puerto serie, donde se conecta el prototipo del guante electrónico.




```
port = st.sidebar.selectbox("Selecciona el puerto serie",
port_options) # Widget de selección para el puerto serie
baudio = st.sidebar.selectbox(
"Select Baudio ▾", (4800, 9600, 19200, 38400, 57600, 115200),
index=1 )
# Conectarse al puerto serie seleccionado
ser = serial.Serial(port, baudio)
```

**Figura 44** Selección para el puerto serie

*Fuente: Investigador*

Para guardar los nuevos datos se abre el archivo dataf.csv, como se aprecia en la figura 45, y de acuerdo con el formato se registra los valores para los dedos Pulgar, Índice, Medio, Anular y Meñique, y los ángulos Roll, Pitch y Yaw del sensor MPU6050, el check con el icono “” inicia la lectura de los datos y el botón “ Guardar datos” abre el archivo .csv y guarda el nuevo registro de acuerdo con el número de muestras que el usuario desee colocar.

```

btn = st.sidebar.button('  Guardar datos')
sensor_data = [] # store data
if pSerial == " " and port:
    c_serial.write("Run")
    data = c_serial.empty()
    while True:
        with data.container():
            line = ser.readline().strip().decode("utf-8")
            values = line.split(",")
            if len(values) == 8:
                a, b, c, d, e, f, g, h = values
                st.json({
                    "Pulgar": a,
                    "Indice": b,
                    "Medio": c,
                    "Anular": d,
                    "Meñique": e,
                    "Roll": f,
                    "Pitch": g,
                    "Yaw": h,
                    "Letra": s_img
                })
            else:
                st.write("  Guardando datos espere")
# abre el archivo dataf.csv
fileName = "dataf.csv"
file = open(fileName, "a")
# guarda los datos
with open(fileName, 'a', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerows(sensor_data)
file.close()

```

**Figura 45** Guardar nuevos datos

*Fuente: Investigador*

Finalmente, en la figura 46 se aprecia la interfaz web para el registro de nuevos datos, en la parte inferior izquierda se observa el puerto serie junto con los baudios, y en la parte superior derecha se observa el inicio de lectura del guante electrónico representado con el check verde.

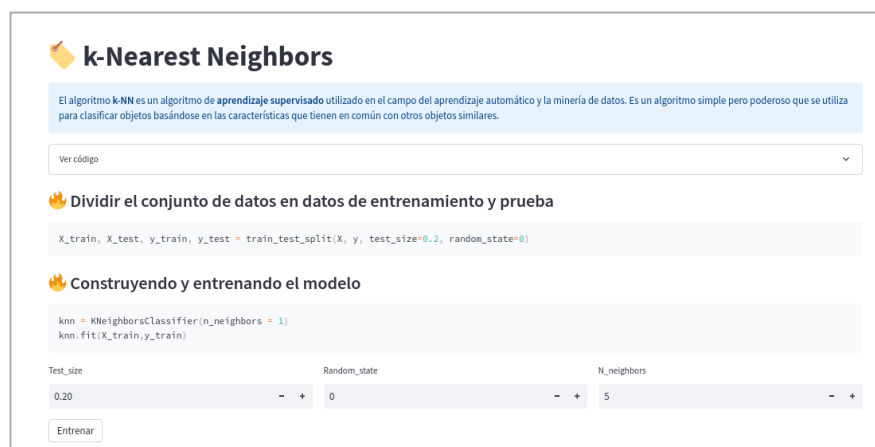


**Figura 46** Interfaz web - Hoja de datos

*Fuente: Investigador*

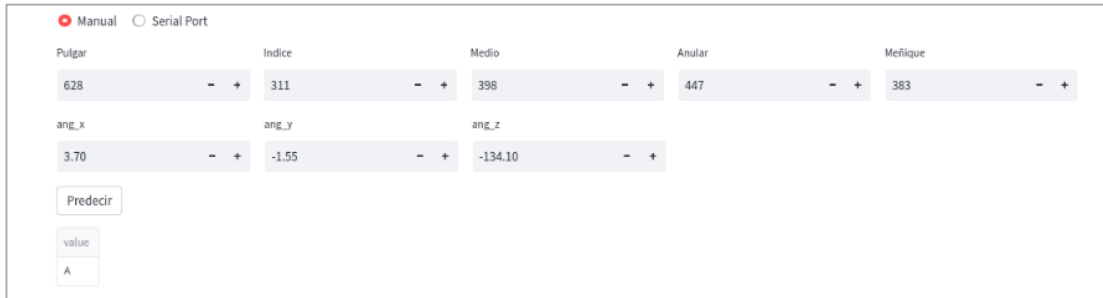
### 3.2.11.3 Creación de la página K-Nearest Neighbors

Se creó la carpeta en la dirección *pages/K-NN.py*. En esta página se presenta información del modelo y el proceso de entrenamiento, con la finalidad de presentar el proceso que se llevó a cabo para predecir la lengua de señas. Presenta el código explicado en la sección 3.2.10. Las figuras 47 y 48 presentan una predicción del modelo de manera manual, el cual se ingresan los valores en las casillas correspondientes de acuerdo con la tabla de datos, en este ejemplo se ingresó la letra A. La opción Serial Port permite una predicción con el guante electrónico, sin embargo, su predicción es algo lenta dado que realiza todo el proceso de entrenamiento.



**Figura 47** Interfaz web - Página K-Nearest Neighbors

*Fuente: Investigador*



**Figura 48** Interfaz web - Página KNN - Predicción del modelo

*Fuente: Investigador*

### 3.2.11.4 Creación de la página Random Forest

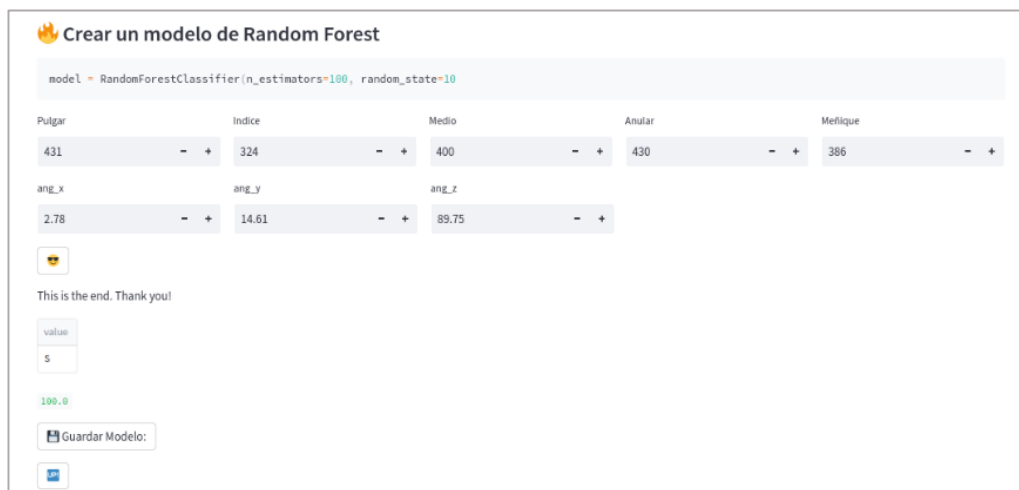
Se creó la carpeta en la dirección *pages/Random\_Forest.py*. De igual manera en esta página se detalla información del modelo junto con el código y proceso de entrenamiento descritos en la sección 3.2.10. Se presenta la opción de cargar el archivo .csv para iniciar el proceso, y en las figuras 49 y 50 se puede apreciar una predicción manual. El algoritmo de Random Forest al presentar mejor precisión fue el seleccionado para el sistema final, por ende, en esta sección se agregó el botón “guardar”, permite crear el archivo .pkl del modelo para poder usarlo en el sistema. sin embargo, se debe tener en cuenta que el modelo entrenado se dividió en tres archivos, para evitar conflictos en las predicciones. Los archivos entrenados son los siguientes: “rf\_wndf.pkl” para los gestos estáticos, “rf\_num.pkl” para los números y “rf\_wd.pkl” para los gestos dinámicos.

Para los gestos dinámicos se categorizo en colores y palabras, respectivamente con los archivos “rf\_data\_colores” y “rf\_data\_palabras”, con la finalidad de organizar de mejor manera las palabras básicas que conforman el sistema y evitar conflicto en las predicciones dado a los movimientos que requieren dichos gestos.



**Figura 49** Interfaz web - Página Random Forest

*Fuente: Investigador*



**Figura 50** Interfaz web - Página Random Forest - Predicción del modelo

*Fuente: Investigador*

### Creación de la página Lengua de Señas Ecuatoriana (LSEC)

Así mismo, se creó la página en la dirección pages/Game.py. En las figuras 51, 52 y 53 se presentan el funcionamiento del sistema automático para traducir la lengua de señas empleando el modelo de aprendizaje automático entrenado. Se carga el modelo “rb.pkl”, a continuación, se presentan dos opciones de acuerdo con los iconos “🗣️” y “🔊” para iniciar el sistema el cual permite visualizar la predicción tanto en una salida de texto como de audio respectivamente.

```
# Cargar el modelo desde el archivo
with open (archivo, 'rb') as f:
    rf = pickle.load(f)
# Predecir con salida de texto
check = st.checkbox('🗣️ 🗒️')
if check and port:
    st.write('# 🗣️ 🗒️')
    data = st.empty()
    while True:
        with data.container():
            line = ser.readline().strip().decode("utf-8")
            valores = [float(valor) for valor in line.split(',')]
            X_new = [valores]
            st.text(X_new)
            letra_predicha = rf.predict(X_new)
            st.metric(label="🤖", value=str(letra_predicha),
            delta="🚀 ")
```

**Figura 51** Funcionamiento del modelo entrenado

*Fuente: Investigador*

Para la predicción con salida de audio se utilizó la biblioteca gTTS (Google Text-to-Speech) para convertir el texto en un archivo de audio en español, luego lo guarda como "example.mp3" para ser reproducida con la función play().

```
# predecir con salida de audio
def generar_y_reproducir_audio(texto):
    audio = gTTS(text=texto, lang="es", slow=False)
    audio.save("example.mp3")
    audio_file = open('example.mp3', 'rb')
    audio_f = "example.mp3"
    sound = AudioSegment.from_file(audio_f, format="mp3")
    play(sound)

chk = st.checkbox('🔊')
letra_predicha = rf.predict(X_new)
st.write(letra_predicha)

# Generar y reproducir el audio en un hilo separado
audio_thread = threading.Thread(target=generar_y_reproducir_audio,
args=(txt,))
audio_thread.start()
st.cache_resource.clear()
```

**Figura 52** Predicción del modelo

*Fuente: Investigador*



**Figura 53** Interfaz web - Página LSEC

*Fuente: Investigador*

### 3.2.11.5 Diagrama de flujo del proceso de la interfaz web

A continuación, se presentan los diagramas de flujo de los procesos de la interfaz web del sistema automático de decodificación de lengua de señas descritas anteriormente.

La figura 54 muestra el diagrama de flujo correspondiente al proceso para el ingreso de la página de hoja de datos, la página KNN y la página Random Forest.

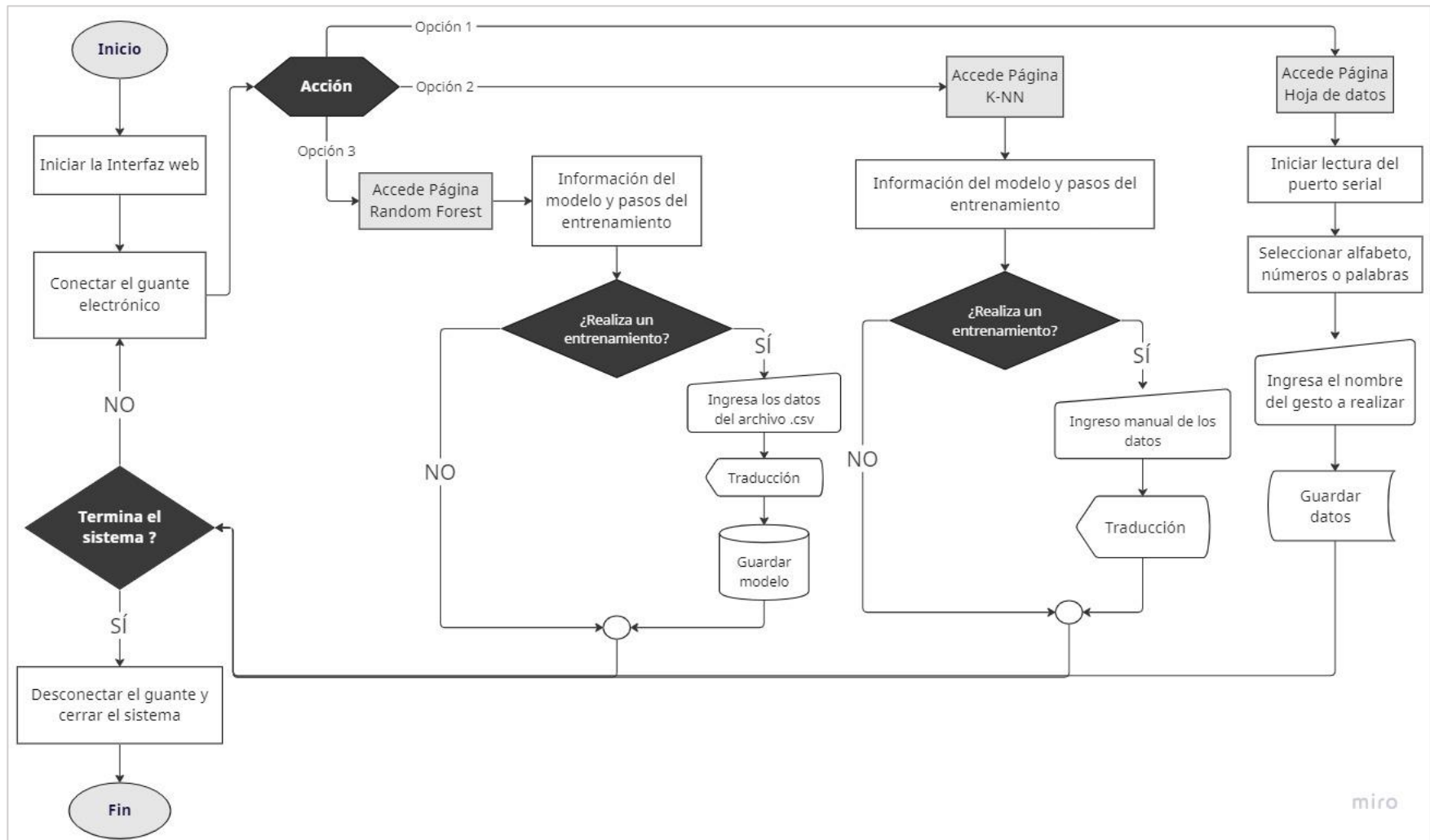
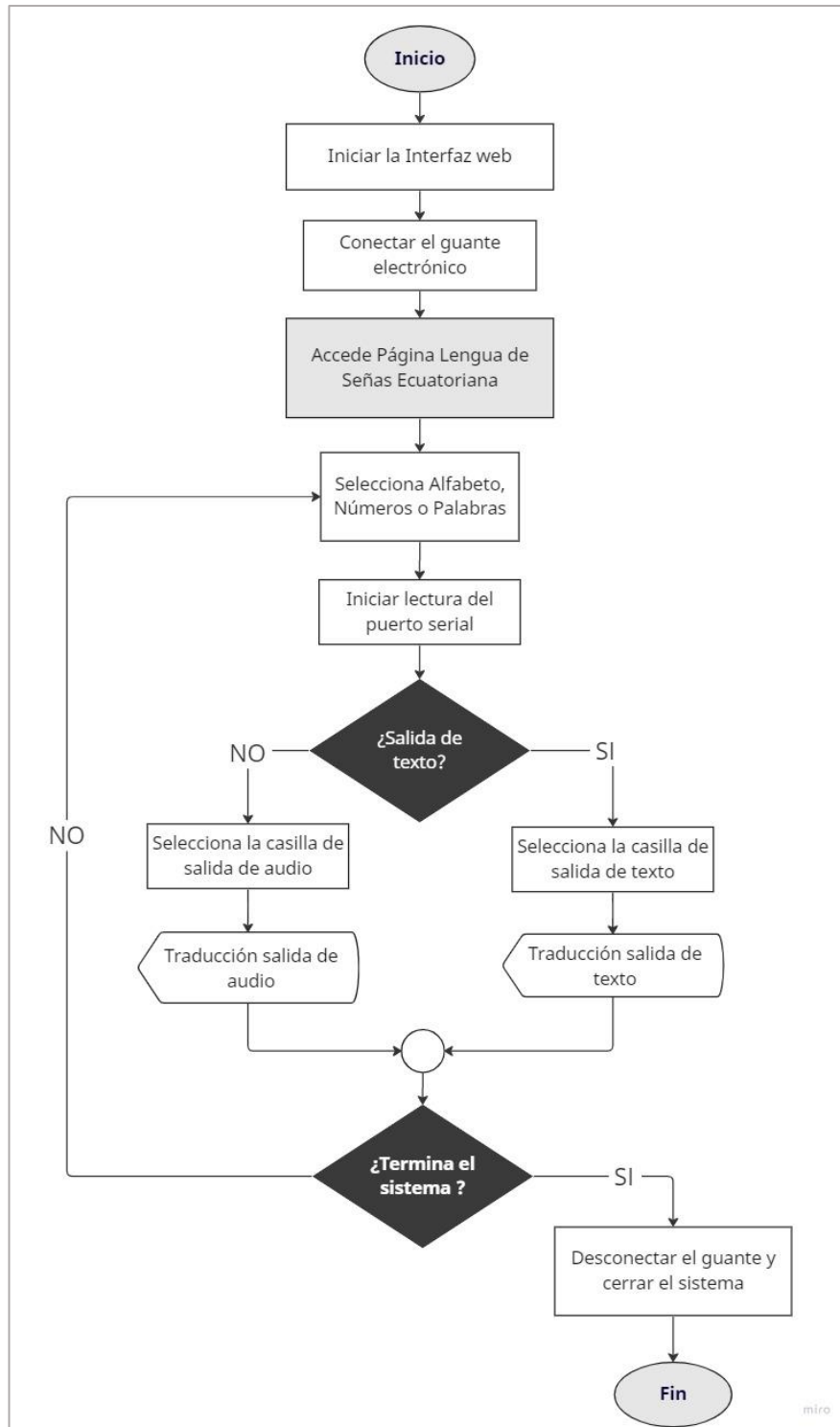


Figura 54 Diagrama de flujo de la Interfaz web 1/2

Fuente: Investigador

La figura 55 muestra el diagrama de flujo correspondiente al proceso para el ingreso de la página Lengua de Señas Ecuatoriana.



**Figura 55** Diagrama de flujo de la interfaz web 2/2

**Fuente:** Investigador



### **3.2.12 Manual de usuario**

Se presento un manual de uso con las instrucciones necesarias respecto a la utilización del prototipo del guante electrónico junto con la interfaz web. El manual presenta el glosario básico del sistema junto con una descripción básica del manejo del guante y de las páginas de la interfaz web; el manual se encuentra en la sección de anexos.

### **3.2.13 Pruebas de funcionamiento y análisis de resultados**

#### **Pruebas de funcionamiento**

Una vez finalizado el sistema automático de decodificación de lengua de señas, es necesario verificar la precisión de la traducción en usuarios. Se realizaron pruebas de funcionamiento a cinco personas con la finalidad de determinar el porcentaje de aciertos para las tres categorías del glosario básico.

Para determinar el porcentaje de aciertos, se realizaron 5 pruebas llamadas números de aciertos, de esta manera para cada usuario se calculó el porcentaje individual de cada gesto, para posteriormente realizar un promedio por usuario, generando así porcentajes de aciertos para cada categoría, y finalmente obtener un resultado general de la precisión del sistema.

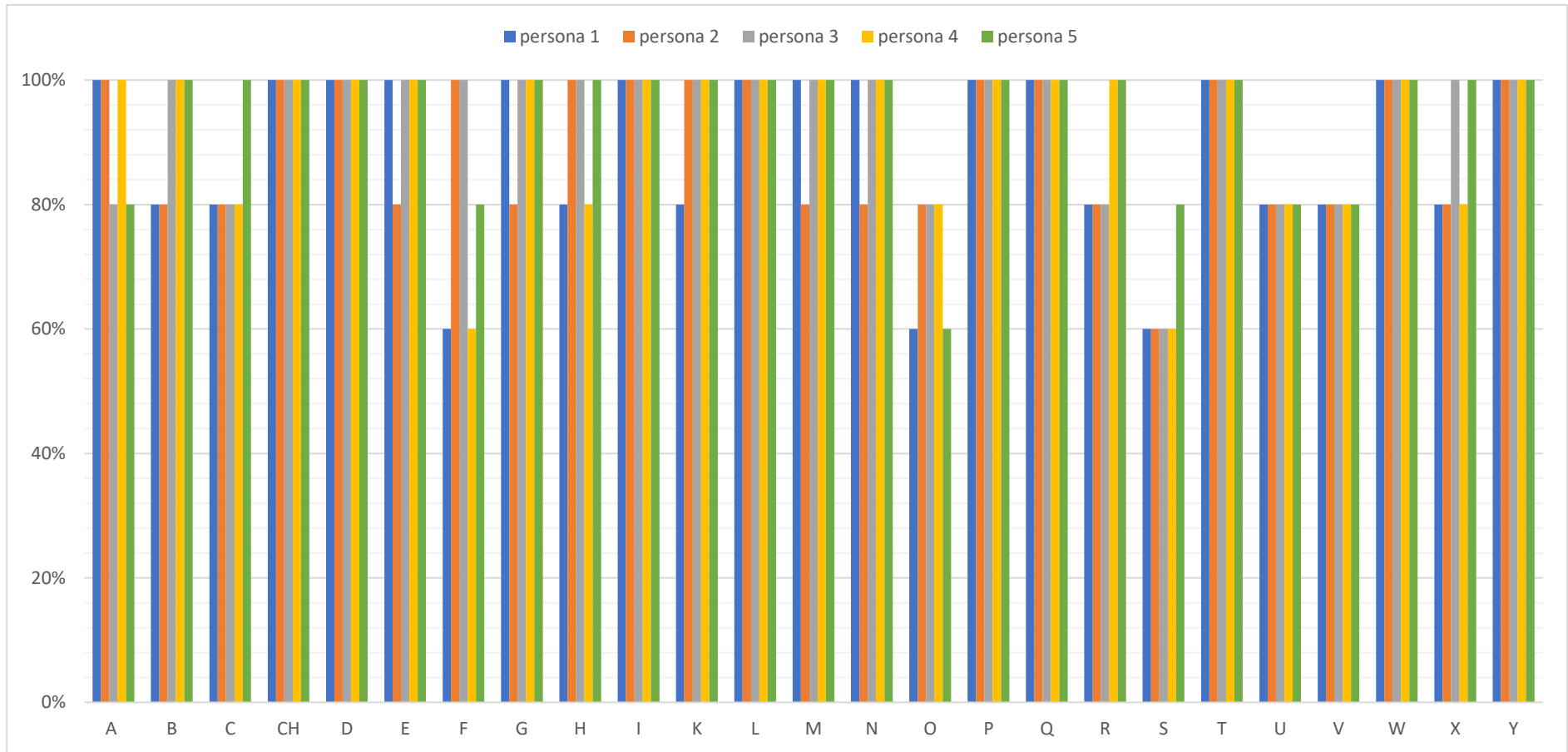
- **Gestos Estáticos**

Considerando los datos registrados en la tabla 27 e ilustrados en la figura 56, el porcentaje de números de aciertos promedio de los cinco usuarios es del 91,68%. Las letras que presentaron menor número de aciertos son la “F, O, S” dando aciertos de 3 a 5, dada sus similitudes físicas con las letras “T, S y E” respectivamente. Las letras “U y V” presentan una precisión del 80% en todos los usuarios dado que su posición es similar, logrando identificarlas únicamente porque se parte de la posición de reposo. Por otra parte, las demás letras presentan altos porcentajes incluso varios del 100%, por lo que se concluye para esta categoría que el modelo Random Forest entrenado presenta una buena predicción.

*Tabla 27 Porcentajes de aciertos para los gestos estáticos*

| Letras       | Persona 1 |      |      | Persona 2 |      |       | Persona 3 |      |      | Persona 4 |      |       | Persona 5 |      |       |
|--------------|-----------|------|------|-----------|------|-------|-----------|------|------|-----------|------|-------|-----------|------|-------|
|              | N° A      | %    | L.S. | N° A      | %    | L. S. | N° A      | %    | L.S. | N° A      | %    | L. S. | N° A      | %    | L. S. |
| <b>A</b>     | 5         | 100% |      | 5         | 100% |       | 4         | 80%  | E    | 5         | 100% |       | 4         | 80%  | E     |
| <b>B</b>     | 4         | 80%  | F    | 4         | 80%  | F     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>C</b>     | 4         | 80%  | E    | 4         | 80%  | E     | 4         | 80%  | E    | 4         | 80%  | E     | 5         | 100% |       |
| <b>CH</b>    | 5         | 100% |      | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>D</b>     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>E</b>     | 5         | 100% |      | 4         | 80%  |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>F</b>     | 3         | 60%  | T    | 5         | 100% |       | 5         | 100% |      | 3         | 60%  | T     | 4         | 80%  | T     |
| <b>G</b>     | 5         | 100% |      | 4         | 80%  | Q     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>H</b>     | 4         | 80%  | CH   | 5         | 100% |       | 5         | 100% |      | 4         | 80%  | CH    | 5         | 100% |       |
| <b>I</b>     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>K</b>     | 4         | 80%  | V    | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>L</b>     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>M</b>     | 5         | 100% |      | 4         | 80%  | Q     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>N</b>     | 5         | 100% |      | 4         | 80%  | O     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>O</b>     | 3         | 60%  | S,E  | 4         | 80%  | C     | 4         | 80%  | E    | 4         | 80%  | C,S   | 3         | 60%  | S,E   |
| <b>P</b>     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>Q</b>     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>R</b>     | 4         | 80%  |      | 4         | 80%  | D     | 4         | 80%  | D    | 5         | 100% |       | 5         | 100% |       |
| <b>S</b>     | 3         | 60%  | E,O  | 3         | 60%  | O     | 3         | 60%  | E,O  | 3         | 60%  | E,O   | 4         | 80%  | O     |
| <b>T</b>     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>U</b>     | 4         | 80%  | V    | 4         | 80%  | V     | 4         | 80%  | V    | 4         | 80%  | V     | 4         | 80%  | V     |
| <b>V</b>     | 4         | 80%  | U    | 4         | 80%  | U     | 4         | 80%  | U    | 4         | 80%  | U     | 4         | 80%  | U     |
| <b>W</b>     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>X</b>     | 4         | 80%  | S    | 4         | 80%  | C     | 5         | 100% |      | 4         | 80%  | S     | 5         | 100% |       |
| <b>Y</b>     | 5         | 100% |      | 5         | 100% |       | 5         | 100% |      | 5         | 100% |       | 5         | 100% |       |
| <b>Total</b> | 88,80%    |      |      | 89,60%    |      |       | 93,60%    |      |      | 92,00%    |      |       | 94,40%    |      |       |

Fuente: Investigador



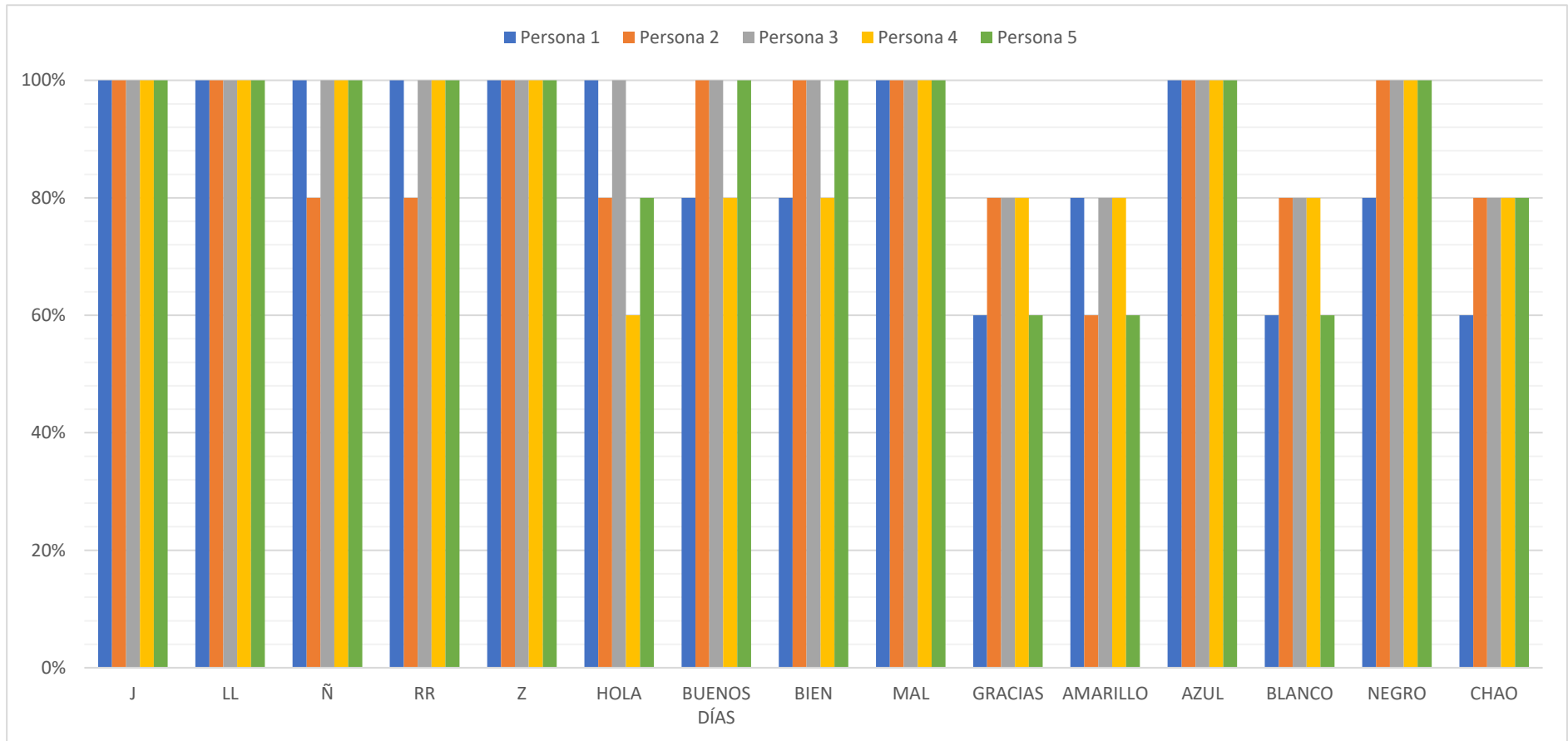
**Figura 56** Porcentaje de aciertos para la traducción de los gestos estáticos

**Fuente:** Investigador

*Tabla 28 Porcentajes de aciertos para los gestos dinámicos*

| Salida             | Persona 1 |      |                   | Persona 2 |      |             | Persona 3 |      |             | Persona 4 |      |                  | Persona 5 |      |             |
|--------------------|-----------|------|-------------------|-----------|------|-------------|-----------|------|-------------|-----------|------|------------------|-----------|------|-------------|
|                    | N° A      | %    | P.S.              | N° A      | %    | P.S.        | N° A      | %    | P.S.        | N° A      | %    | P.S.             | N° A      | %    | P.S.        |
| <b>J</b>           | 5         | 100% |                   | 5         | 100% |             | 5         | 100% |             | 5         | 100% |                  | 5         | 100% |             |
| <b>LL</b>          | 5         | 100% |                   | 5         | 100% |             | 5         | 100% |             | 5         | 100% |                  | 5         | 100% |             |
| <b>Ñ</b>           | 5         | 100% |                   | 4         | 80%  | Z           | 5         | 100% |             | 5         | 100% |                  | 5         | 100% |             |
| <b>RR</b>          | 5         | 100% |                   | 4         | 80%  | Z           | 5         | 100% |             | 5         | 100% |                  | 5         | 100% |             |
| <b>Z</b>           | 5         | 100% |                   | 5         | 100% |             | 5         | 100% |             | 5         | 100% |                  | 5         | 100% |             |
| <b>Hola</b>        | 5         | 100% |                   | 4         | 80%  | Buenos días | 5         | 100% |             | 3         | 60%  | Buenos días, mal | 4         | 80%  | Buenos días |
| <b>Buenos días</b> | 4         | 80%  | Chao              | 5         | 100% |             | 5         | 100% |             | 4         | 80%  | Chao             | 5         | 100% |             |
| <b>Bien</b>        | 4         | 80%  | Hola              | 5         | 100% |             | 5         | 100% |             | 4         | 80%  | Hola             | 5         | 100% |             |
| <b>Mal</b>         | 5         | 100% |                   | 5         | 100% |             | 5         | 100% |             | 5         | 100% |                  | 5         | 100% |             |
| <b>Gracias</b>     | 3         | 60%  | Amarillo, negro   | 4         | 80%  | Amarillo    | 4         | 80%  | Amarillo    | 4         | 80%  | Amarillo         | 3         | 60%  | Amarillo    |
| <b>Amarillo</b>    | 4         | 80%  | Gracias           | 3         | 60%  | Gracias     | 4         | 80%  | Gracias     | 4         | 80%  | Gracias          | 3         | 60%  | Gracias     |
| <b>Azul</b>        | 5         | 100% |                   | 5         | 100% |             | 5         | 100% |             | 5         | 100% |                  | 5         | 100% |             |
| <b>Blanco</b>      | 3         | 60%  | Negro             | 4         | 80%  | Amarillo    | 4         | 80%  | Negro       | 4         | 80%  | Amarillo         | 3         | 60%  | Negro       |
| <b>Negro</b>       | 4         | 80%  | Amarillo          | 5         | 100% |             | 5         | 100% |             | 5         | 100% |                  | 5         | 100% |             |
| <b>Chao</b>        | 3         | 60%  | Hola, Buenos días | 4         | 80%  | Hola        | 4         | 80%  | Buenos días | 4         | 80%  | Buenos días      | 4         | 80%  | Buenos días |
| <b>Total</b>       | 86,67%    |      |                   | 89,33%    |      |             | 94,67%    |      |             | 89,33%    |      |                  | 89,33%    |      |             |

Fuente: Investigador



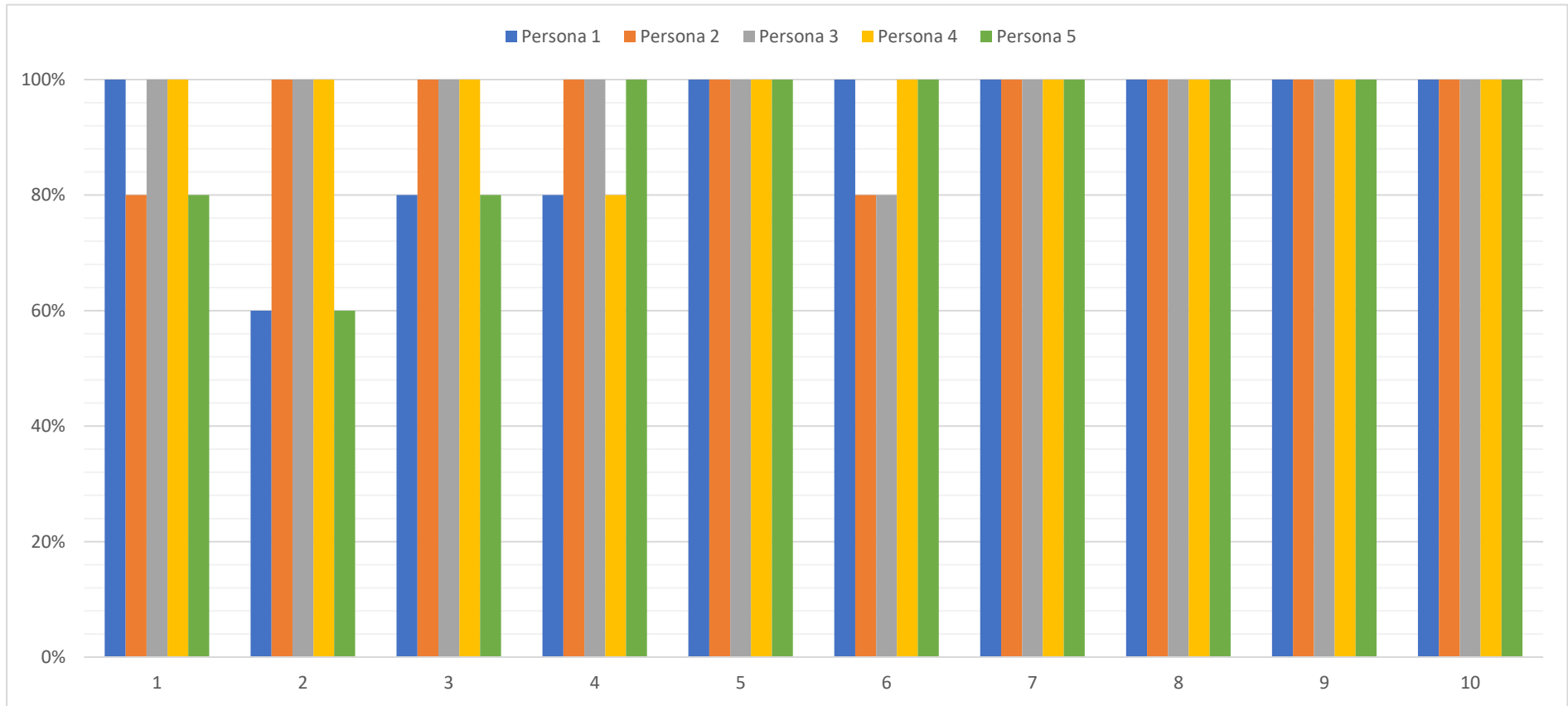
**Figura 57** Porcentaje de aciertos para la traducción de los gestos dinámicos

**Fuente:** Investigador

**Tabla 29** Porcentajes de aciertos para los números

| Números      | Persona 1   |      |           | Persona 2   |      |           | Persona 3   |      |           | Persona 4   |      |           | Persona 5   |      |           |
|--------------|-------------|------|-----------|-------------|------|-----------|-------------|------|-----------|-------------|------|-----------|-------------|------|-----------|
|              | N° Aciertos | %    | Num. Sem. | N° Aciertos | %    | Num. Sem. | N° Aciertos | %    | Num. Sem. | N° Aciertos | %    | Num. Sem. | N° Aciertos | %    | Num. Sem. |
| <b>1</b>     | 5           | 100% |           | 4           | 80%  | 10        | 5           | 100% |           | 5           | 100% |           | 4           | 80%  | 10        |
| <b>2</b>     | 3           | 60%  | 7         | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 3           | 60%  | 7         |
| <b>3</b>     | 4           | 80%  |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 4           | 80%  |           |
| <b>4</b>     | 4           | 80%  | 8         | 5           | 100% |           | 5           | 100% |           | 4           | 80%  | 5         | 5           | 100% |           |
| <b>5</b>     | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           |
| <b>6</b>     | 5           | 100% |           | 4           | 80%  | 2         | 4           | 80%  | 5         | 5           | 100% |           | 5           | 100% |           |
| <b>7</b>     | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           |
| <b>8</b>     | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           |
| <b>9</b>     | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           |
| <b>10</b>    | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           | 5           | 100% |           |
| <b>Total</b> | 92%         |      |           | 96%         |      |           | 98%         |      |           | 98%         |      |           | 92%         |      |           |

Fuente: Investigador



**Figura 58** Porcentaje de aciertos para la traducción de los números

**Fuente:** Investigador

- **Gestos Dinámicos**

Considerando los datos registrados en la tabla 28 e ilustrados en la figura 57, para esta categoría se tiene un porcentaje promedio del 89,87 %. Las palabras que presentan un bajo porcentaje de aciertos son “gracias, amarillo, blanco y chao”, debido a las similitudes iniciales de movimiento que presentan con la palma abierta y la separación de los dedos. por lo que llegan a confundirse entre sí y con las palabras “hola y buenos dias”, provocando porcentajes de aciertos hasta en un 60%. Para las palabras “mal, azul y negro” se conserva una precisión del 80% y 100%, resultando las palabras básicas con mejor predicción del sistema.

- **Números**

Considerando los datos registrados en la tabla 29 e ilustrados en la figura 58, para esta categoría se tiene un porcentaje promedio del 95,20%, siendo el porcentaje más alto de todas las categorías, dado que los números por si solos no presentan gran complejidad al momento de traducir mediante el modelo entrenado. Entre los números que presentan similitudes se encuentran el 1, 2, 4 y 6, respecto a los números 10, 7, 8 y 5, sin embargo, el porcentaje de aciertos no baja del 80% (4 aciertos de 5), por lo que el promedio del usuario más bajo es del 92%, resultando ser la categoría con la precisión más alta.

### **Análisis general de los resultados**

De acuerdo con las tablas 27,28 y 29 se resumen los resultados en la tabla 30. Por lo que se concluye que el porcentaje general del sistema es del 92,25%.

*Tabla 30 Resultado general del sistema*

| <b>Grupos</b>           | <b>Número de gestos</b> | <b>Porcentaje de aciertos</b> |
|-------------------------|-------------------------|-------------------------------|
| <i>Gestos estáticos</i> | 25                      | 91,68%                        |
| <i>Gestos dinámicos</i> | 15                      | 89,87%                        |
| <i>Números</i>          | 10                      | 95,20%                        |
| <b>Total</b>            | 50                      | 92,25%                        |

**Fuente:** Investigador



### 3.2.14 Presupuesto

En la tabla 31 se presenta el costo de los materiales utilizados en el desarrollo del sistema y la construcción del guante electrónico, dando un total de \$176,96. Cabe mencionar que el costo total del proyecto puede incrementar de acuerdo con la mano de obra, sin embargo en este caso no se coloca porque se realizó una investigación de más de seis meses en donde se probaron varios prototipos y se experimentó con distintos componentes electrónicos hasta llegar a determinar el más eficiente y con el que se obtuvieron mejores resultados para el trabajo de titulación propuesto.

*Tabla 31 Presupuesto de construcción*

| <i>ítem</i>      | <i>Descripción</i>  | <i>Unidad</i> | <i>Cantidad</i> | <i>Costo Unitario</i> | <i>Costo total</i> |
|------------------|---------------------|---------------|-----------------|-----------------------|--------------------|
| <b>1</b>         | Sensores Flex 2.2"  | c/u           | 5               | \$ 25,00              | \$ 125,00          |
| <b>2</b>         | Sensor MPU-6050     | c/u           | 1               | \$ 4,50               | \$ 4,50            |
| <b>3</b>         | Arduino NANO        | c/u           | 1               | \$ 8,50               | \$ 8,50            |
| <b>4</b>         | Resistencias        | c/u           | 5               | \$ 0,10               | \$ 0,50            |
| <b>5</b>         | Guante              | c/u           | 1               | \$ 8,00               | \$ 8,00            |
| <b>6</b>         | Cables              | metro         | 1               | \$ 0,50               | \$ 0,50            |
| <b>7</b>         | Plástico            | metro         | 1               | \$ 1,50               | \$ 1,50            |
| <b>8</b>         | Costura             | tiras         | 1               | \$ 1,50               | \$ 1,50            |
| <b>9</b>         | Baquelita perforada | c/u           | 1               | \$ 1,50               | \$ 1,50            |
| <b>10</b>        | Conector para PCB   | c/u           | 1               | \$ 1,50               | \$ 1,50            |
| <b>11</b>        | Cable USB           | c/u           | 1               | \$ 3,50               | \$ 3,50            |
| <b>12</b>        | Estaño              | c/u           | 1               | \$ 1,50               | \$ 1,50            |
| <b>Subtotal</b>  |                     |               |                 |                       | \$ 158,00          |
| <b>IVA (12%)</b> |                     |               |                 |                       | \$ 18,96           |
| <b>Total</b>     |                     |               |                 |                       | \$ 176,96          |

Fuente: Investigador

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

- Se realizó una investigación en base al diccionario de lengua de señas ecuatoriano “Gabriel Román” para establecer el vocabulario básico del sistema. Determinando que la lengua de señas se divide en dos grandes grupos, los gestos estáticos y los gestos dinámicos, por lo tanto, para llevar a cabo la implementación del sistema se seleccionó gestos básicos a parte del alfabeto dactilológico, como lo son saludos, despedidas, colores y números que pueden realizarse con una única mano, de esta manera se logró establecer un glosario básico de 50 gestos fácil de interpretar y realizar para los usuarios.
- El diseño y construcción del guante electrónico cumple con los requerimientos necesarios para una traducción fiable y precisa de la lengua de señas. La implementación de componentes electrónicos como los sensores Flex y el sensor MPU6050 permitieron capturar las señas con mayor precisión. Además, el uso del software como Python con Streamlit para el entrenamiento automático brindó un enfoque eficiente en el desarrollo del sistema. La combinación de estos componentes electrónicos y herramientas de software resultó en un prototipo de bajo costo que promueve la accesibilidad para los usuarios y la viabilidad para futuras investigaciones.
- La base de datos recopilada fue generada en base a cinco usuarios entre jóvenes y adultos. Este conjunto de datos permitió obtener un total de 6625 registros, contribuyendo a los modelos K-NN y Random Forest a mejorar la precisión del sistema, gracias a esta amplia variedad de información. Cabe mencionar que al utilizar el Framework Streamlit de Python, permitió crear una interfaz web en la cual se visualiza el código de los modelos evaluados junto con su porcentaje de precisión, de esta manera los usuarios pueden familiarizarse con el funcionamiento del aprendizaje automático de una manera fácil e intuitiva.

- Los modelos K-NN y Random Forest arrojaron una precisión del 94,62% y 96,20% respectivamente, por tanto, el funcionamiento del sistema fue evaluado de acuerdo con el modelo Random Forest. Se realizaron pruebas de funcionamiento a 5 personas entre jóvenes y adultos para determinar el porcentaje de aciertos, de esta manera se evaluó el rendimiento del sistema, concluyendo con un porcentaje general del 92,25%, haciendo al sistema óptimo para traducir la lengua de señas ecuatoriano en base a un glosario básico.

#### **4.2 Recomendaciones**

- Se recomienda considerar la incorporación de algoritmos de reconocimiento más avanzados, como las redes neuronales artificiales (RNA), Support Vector Machines (SVM) o el alineamiento temporal dinámico (DTW), con el propósito de mejorar significativamente la precisión del sistema en la traducción de lengua de señas ecuatoriana. Sin embargo, se debe tener en cuenta la entrada y salida de los datos, debido a que la LSEC presenta una salida de texto, por lo que involucraría la utilización de encoders aumentando de esta manera la complejidad, el tamaño y el preprocesamiento de los datos.
- Para futuros trabajos se podría diseñar un guante electrónico inalámbrico, de esta manera los gestos dinámicos realizados por los usuarios tendrán mayor libertad de movimiento, sin embargo, se debe tener en cuenta consideraciones de diseño, dado que aumentaría el circuito electrónico, por ende, el peso y tamaño del guante incrementaría. Además, cabe mencionar en el caso que se utilice tecnología Bluetooth existiría una disminución de la velocidad de transmisión de los datos.
- Con un mayor presupuesto dependiendo del costo de los nuevos sensores se puede lograr que la predicción de los gestos tenga una mayor precisión al utilizar sensores de mayor calidad y capacidad, debido a que el sensor MPU-6050 depende de gran medida de la colocación en el dorso de la mano, resultando en que, si el guante no está colocado de manera correcta existirá errores en la predicción de los gestos dinámicos, de igual manera los sensores Flex 2.2” al ser de una longitud corta puede fallar en censar los valores de flexión de cada dedo dado a sus diferentes longitudes.

- Para ampliar el alcance del proyecto, se recomienda diseñar dos guantes electrónicos para ambas manos, de esta manera el glosario básico del sistema puede incrementar y abarcar palabras más compuestas, sin embargo, se debe considerar un mayor presupuesto, dado que la lengua de señas requiere movimientos muy amplios por lo que los componentes electrónicos deben cubrir varias necesidades de diseño.

## BIBLIOGRAFÍA

- [1] D. A. Guzmán Arellano , «Guante Electrónico para Traducir de Lengua de Señas a Caracteres con Voz Artificial y Conexión Inalámbrica a Dispositivos Móviles para Personas con Discapacidad Auditiva y de Lengua en la Universidad Técnica de Ambato.,» Universidad Técnica de Ambato, Ambato, 2017.
- [2] M. A. Parreño Alvarez , «Desarrollo de un juego didáctico mediante sensores Leap Motion para estimular el aprendizaje del lenguaje básico de señas ecuatoriano.,» Universidad Politécnica Salesiana, Quito, 2018.
- [3] J. Schioppo, Z. Meyer, D. Fabiano y S. Canavan, «Learning Sign Language in Virtual Environment,» University of South Florida, Tampa, 2019.
- [4] B. F. Garzón Tipán , «Sistema de traducción de lenguaje de señas a lenguaje oral y escrito mediante un guante electrónico.,» Universidad Tecnológica Israel, Quito, 2020.
- [5] D. A. López Albán y M. A. López Barrera, «Sistema de reconocimiento automático de lengua de señas colombiana mediante dispositivo óptico de captura.,» Universidad Mariana, Pasto, 2021.
- [6] Naciones Unidas, «un.org,» Alianza Internacional de la Discapacidad, 23 Septiembre 2021. [En línea]. Available: <https://www.un.org/es/observances/sign-languages-day>. [Último acceso: 22 Junio 2022].
- [7] T. Izquierdo, «Infomed,» Red de Salud Cuba, 3 Marzo 2014. [En línea]. Available: <https://articulos.sld.cu/editorhome/2014/03/03/3-de-marzo-dia-internacional-del-cuidado-del-oido-y-la-audicion/>. [Último acceso: 22 Junio 2022].
- [8] CONADIS, «Consejo Nacional para la Igualdad de Discapacidades,» Gobierno del Encuentro, Enero 2022. [En línea]. Available:

<https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>.  
[Último acceso: 26 Junio 2022].

- [9] CONADIS, «Ministerio de Salud Pública,» Agosto 2015. [En línea]. Available: [https://www.consejodiscapacidades.gob.ec/wp-content/uploads/downloads/2015/09/estadistica\\_conadis.pdf](https://www.consejodiscapacidades.gob.ec/wp-content/uploads/downloads/2015/09/estadistica_conadis.pdf). [Último acceso: 26 Junio 2022].
- [10] A. Álvarez, «cotral.es,» Contral Lab, 14 Enero 2019. [En línea]. Available: <https://www.cotral.es/blog/prevencion-riesgos-auditivos/el-funcionamiento-del-oido-humano.html#:~:text=El%20oído%20se%20compone%20de,las%20vibraciones%20al%20oído%20interno..> [Último acceso: 4 Diciembre 2022].
- [11] National Institutes of Health, «Starkey,» 2007. [En línea]. Available: <https://www.starkey.com/hearing-loss/types-and-causes>. [Último acceso: 30 Junio 2022].
- [12] A. M. Ferrer Manchón , «uv.es,» 6 Noviembre 2002. [En línea]. Available: [https://www.uv.es/ferreram/publicaciones/Tecnoneet\\_Antonio\\_Ferrer.pdf](https://www.uv.es/ferreram/publicaciones/Tecnoneet_Antonio_Ferrer.pdf). [Último acceso: 20 Diciembre 2022].
- [13] A. Monroy, «El telégrafo,» 23 Septiembre 2018. [En línea]. Available: <https://www.eltelegrafo.com.ec/noticias/sociedad/6/lengua-senas-inclusion>. [Último acceso: 30 Junio 2022].
- [14] CONADIS, «plataformaconadis.gob.ec,» 15 Julio 2014. [En línea]. Available: <http://www.plataformaconadis.gob.ec/~platafor/diccionario/>. [Último acceso: 4 Diciembre 2022].
- [15] D. Aranda, ELECTRÓNICA - Plataformas Arduino y Raspberry Pi, Buenos Aires : Andina, 2014.
- [16] D. Thalmann, Intelligent Embedded Systems: Select Proceedings of ICNETS2, Volume II, Singapore: Springer Nature, 2018.

- [17] D. Das, «components101.com,» 18 Marzo 2018. [En línea]. Available: <https://components101.com/sensors/flex-sensor-working-circuit-datasheet>. [Último acceso: 15 Enero 2023].
- [18] J. Upadhyay, «www.circuitbread.com,» 2 Febrero 2022. [En línea]. Available: <https://www.circuitbread.com/ee-faq/how-do-accelerometers-and-gyroscopes-work>. [Último acceso: 3 Enero 2023].
- [19] P. Pedamkar, «EDUCBA,» 2022. [En línea]. Available: <https://www.educba.com/artificial-intelligence-techniques/>. [Último acceso: 30 Junio 2022].
- [20] A. Müller y S. Guido, Introduction to Machine Learning with Python, Boston: O'Reilly Media, Inc., 2017.
- [21] M. I. Arango Palacio, M. J. Bernal Vélez y I. Montoya Henao, «Predicción de enfermedades del corazón usando el algoritmo K-Nearest Neighbors,» Universidad EAFIT, Medellín, 2022.
- [22] L. Hernández Cubo, «oa.upm.es,» 10 Febrero 2022. [En línea]. Available: [https://oa.upm.es/72650/1/TFG\\_LAURA\\_HERNANDEZ\\_CUBO.pdf](https://oa.upm.es/72650/1/TFG_LAURA_HERNANDEZ_CUBO.pdf). [Último acceso: 29 Marzo 2023].
- [23] Computer Hope, «www.computerhope.com,» 13 Marzo 2021. [En línea]. Available: <https://www.computerhope.com/jargon/p/programming-language.htm>. [Último acceso: 30 Junio 2022].
- [24] V. Delmas, Anatomía Humana, Barcelona: Masson, 2005.
- [25] Steren, «www.steren.com,» 23 abril 2022. [En línea]. Available: <https://www.steren.com.ec/hoja-conductiva-tipo-velostat.html>. [Último acceso: 24 diciembre 2022].
- [26] AVElectronics , «avelectronics.cc,» 12 octubre 2022. [En línea]. Available: <https://avelectronics.cc/producto/flex-sensor-2-2/>. [Último acceso: 4 Enero 2023].

- [27] 5Hertz Electrónica, «5hertz.com,» 8 Agosto 2020. [En línea]. Available: [https://www.5hertz.com/index.php?route=tutoriales/tutorial&category\\_id=1&utorial\\_id=2#33](https://www.5hertz.com/index.php?route=tutoriales/tutorial&category_id=1&utorial_id=2#33). [Último acceso: 2 Enero 2023].
- [28] A. García González, «/panamahitek.com,» 23 Enero 2013. [En línea]. Available: <https://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>. [Último acceso: 15 Enero 2023].
- [29] Components101, «components101.com,» 22 Abril 2020. [En línea]. Available: <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>. [Último acceso: 11 Enero 2023].
- [30] Brooke, «anvil,» 2023. [En línea]. Available: <https://anvil.works/articles/4-alternatives-streamlit>. [Último acceso: 13 Marzo 2023].
- [31] S. Kuppudurai y V. Jothi, «www.latentview.com,» 17 Mayo 2021. [En línea]. Available: <https://www.latentview.com/data-engineering-lp/introduction-to-streamlit/>. [Último acceso: 2 Enero 2023].
- [32] J. Joseph, «Circuit Digest,» 16 Mayo 2022. [En línea]. Available: <https://circuitdigest.com/microcontroller-projects/interfacing-mpu6050-module-with-arduino>. [Último acceso: 16 Abril 2023].
- [33] InvenSense, «product.tdk.com,» 19 Agosto 2013. [En línea]. Available: [https://product.tdk.com/system/files/dam/doc/product/sensor/motion-inertial/imu/data\\_sheet/mpu-6000-datasheet1.pdf](https://product.tdk.com/system/files/dam/doc/product/sensor/motion-inertial/imu/data_sheet/mpu-6000-datasheet1.pdf). [Último acceso: 2 Enero 2023].



# **ANEXOS**

## Anexo 1: Certificado de funcionamiento

### CERTIFICACIÓN

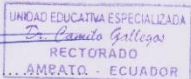
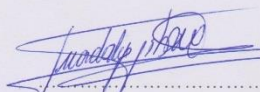
Ambato, 16 de junio de 2023

Ingeniera, Mg.  
Pilar Urrutia  
DECANA  
Facultad de Ingeniería en Sistemas, Electrónica e Industrial  
Presente

Señora Decana:

Por medio de la presente, yo Mg. GUADALUPE ELIZABETH TIBAN VIVAR con cédula de identidad N° 1802024958, Rectora de la Unidad Especializada Dr. Camilo Gallegos Dominguez, certifico que he revisado el proyecto técnico "SISTEMA AUTOMÁTICO DE DECODIFICACIÓN DE LENGUA DE SEÑAS ECUATORIANO A COMUNICACIÓN VERBAL", el mismo que fue diseñado en su total autoría por la Sra. WENDY PATRICIA GAVILANES PROAÑO, con cédula de identidad N° 1803897980, estudiante de la carrera de Telecomunicaciones de la Universidad Técnica de Ambato. Después de haber realizado las pruebas del sistema me permito verificar la funcionalidad del equipo y puedo validar que los resultados obtenidos hasta la fecha son aceptables y de gran ayuda en el área de traducción y aprendizaje de la lengua de señas ecuatoriana.

Saludos cordiales,  
Atentamente,



GUADALUPE ELIZABETH TIBAN VIVAR  
Rectora de la Unidad Especializada Dr. Camilo Gallegos Dominguez  
C.I: 1802024958

## Anexo 2: Hoja de datos técnicos sensor MPU-6050

|   |  |   |
|---|--|---|
|  | <b>MPU-6000/MPU-6050 Product Specification</b> | Document Number: PS-MPU-6000A-00<br>Revision: 3.4<br>Release Date: 08/19/2013 |
|---|--|---|

### 6 Electrical Characteristics

#### 6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

| PARAMETER   | CONDITIONS                                   | MIN | TYP                            | MAX | UNITS  | NOTES |
|---|--|-----|--------------------------------|-----|--|-------|
| <b>GYROSCOPE SENSITIVITY</b>                        |  |     |                                |     |  |       |
| Full-Scale Range                                    | FS_SEL=0<br>FS_SEL=1<br>FS_SEL=2<br>FS_SEL=3 |     | ±250<br>±500<br>±1000<br>±2000 |     | °/s<br>°/s<br>°/s<br>°/s                         |       |
| Gyroscope ADC Word Length                           |  |     | 16                             |     | bits   |       |
| Sensitivity Scale Factor                            | FS_SEL=0<br>FS_SEL=1<br>FS_SEL=2<br>FS_SEL=3 |     | 131<br>65.5<br>32.8<br>16.4    |     | LSB/(°/s)<br>LSB/(°/s)<br>LSB/(°/s)<br>LSB/(°/s) |       |
| Sensitivity Scale Factor Tolerance                  | 25°C   | -3  |                                | +3  | %  |       |
| Sensitivity Scale Factor Variation Over Temperature |  |     | ±2                             |     | %  |       |
| Nonlinearity  | Best fit straight line; 25°C                 |     | 0.2                            |     | %  |       |
| Cross-Axis Sensitivity                              |  |     | ±2                             |     | %  |       |
| <b>GYROSCOPE ZERO-RATE OUTPUT (ZRO)</b>             |  |     |                                |     |  |       |
| Initial ZRO Tolerance                               | 25°C   |     | ±20                            |     | °/s  |       |
| ZRO Variation Over Temperature                      | -40°C to +85°C                               |     | ±20                            |     | °/s  |       |
| Power-Supply Sensitivity (1-10Hz)                   | Sine wave, 100mVpp; VDD=2.5V                 |     | 0.2                            |     | °/s  |       |
| Power-Supply Sensitivity (10 - 250Hz)               | Sine wave, 100mVpp; VDD=2.5V                 |     | 0.2                            |     | °/s  |       |
| Power-Supply Sensitivity (250Hz - 100kHz)           | Sine wave, 100mVpp; VDD=2.5V                 |     | 4                              |     | °/s  |       |
| Linear Acceleration Sensitivity                     | Static                                       |     | 0.1                            |     | °/s/g  |       |
| <b>SELF-TEST RESPONSE</b>                           |  |     |                                |     |  |       |
| Relative  | Change from factory trim                     | -14 |                                | 14  | %  | 1     |
| <b>GYROSCOPE NOISE PERFORMANCE</b>                  |  |     |                                |     |  |       |
| Total RMS Noise                                     | FS_SEL=0<br>DLPCFG=2 (100Hz)                 |     | 0.05                           |     | °/s-rms  |       |
| Low-frequency RMS noise                             | Bandwidth 1Hz to 10Hz                        |     | 0.033                          |     | °/s-rms  |       |
| Rate Noise Spectral Density                         | At 10Hz                                      |     | 0.005                          |     | °/s/√Hz  |       |
| <b>GYROSCOPE MECHANICAL FREQUENCIES</b>             |  |     |                                |     |  |       |
| X-Axis  |  | 30  | 33                             | 36  | kHz  |       |
| Y-Axis  |  | 27  | 30                             | 33  | kHz  |       |
| Z-Axis  |  | 24  | 27                             | 30  | kHz  |       |
| <b>LOW PASS FILTER RESPONSE</b>                     |  |     |                                |     |  |       |
|   | Programmable Range                           |     | 5                              |     | 256  | Hz    |
| <b>OUTPUT DATA RATE</b>                             |  |     |                                |     |  |       |
|   | Programmable                                 |     | 4                              |     | 8,000  | Hz    |
| <b>GYROSCOPE START-UP TIME</b>                      |  |     |                                |     |  |       |
| ZRO Settling (from power-on)                        | DLPCFG=0<br>to ±1% of Final                  |     | 30                             |     | ms   |       |



**6.2 Accelerometer Specifications**

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

| PARAMETER                              | CONDITIONS                        | MIN | TYP    | MAX   | UNITS  | NOTES |
|--|-----------------------------------|-----|--------|-------|--------|-------|
| <b>ACCELEROMETER SENSITIVITY</b>       |                                   |     |        |       |        |       |
| Full-Scale Range                       | AFS_SEL=0                         |     | ±2     |       | g      |       |
|  | AFS_SEL=1                         |     | ±4     |       | g      |       |
|  | AFS_SEL=2                         |     | ±8     |       | g      |       |
|  | AFS_SEL=3                         |     | ±16    |       | g      |       |
| ADC Word Length                        | Output in two's complement format |     | 16     |       | bits   |       |
| Sensitivity Scale Factor               | AFS_SEL=0                         |     | 16,384 |       | LSB/g  |       |
|  | AFS_SEL=1                         |     | 8,192  |       | LSB/g  |       |
|  | AFS_SEL=2                         |     | 4,096  |       | LSB/g  |       |
|  | AFS_SEL=3                         |     | 2,048  |       | LSB/g  |       |
| Initial Calibration Tolerance          |                                   |     | ±3     |       | %      |       |
| Sensitivity Change vs. Temperature     | AFS_SEL=0, -40°C to +85°C         |     | ±0.02  |       | %/°C   |       |
| Nonlinearity                           | Best Fit Straight Line            |     | 0.5    |       | %      |       |
| Cross-Axis Sensitivity                 |                                   |     | ±2     |       | %      |       |
| <b>ZERO-G OUTPUT</b>                   |                                   |     |        |       |        |       |
| Initial Calibration Tolerance          | X and Y axes                      |     | ±50    |       | mg     | 1     |
|  | Z axis                            |     | ±80    |       | mg     |       |
| Zero-G Level Change vs. Temperature    | X and Y axes, 0°C to +70°C        |     | ±35    |       | mg     |       |
|  | Z axis, 0°C to +70°C              |     | ±80    |       | mg     |       |
| <b>SELF TEST RESPONSE</b>              |                                   |     |        |       |        |       |
| Relative                               | Change from factory trim          | -14 |        | 14    | %      | 2     |
| <b>NOISE PERFORMANCE</b>               |                                   |     |        |       |        |       |
| Power Spectral Density                 | @10Hz, AFS_SEL=0 & ODR=1kHz       |     | 400    |       | μg/√Hz |       |
| <b>LOW PASS FILTER RESPONSE</b>        |                                   |     |        |       |        |       |
|  | Programmable Range                | 5   |        | 260   | Hz     |       |
| <b>OUTPUT DATA RATE</b>                |                                   |     |        |       |        |       |
|  | Programmable Range                | 4   |        | 1,000 | Hz     |       |
| <b>INTELLIGENCE FUNCTION INCREMENT</b> |                                   |     |        |       |        |       |
|  |                                   |     | 32     |       | mg/LSB |       |

1. Typical zero-g initial calibration tolerance value after MSL3 preconditioning
2. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*



**6.3 Electrical and Other Common Specifications**

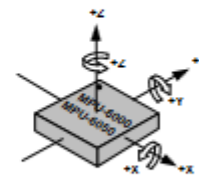
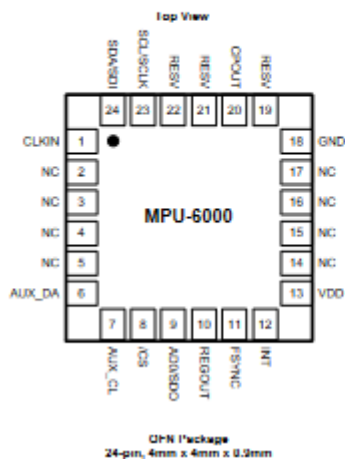
VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

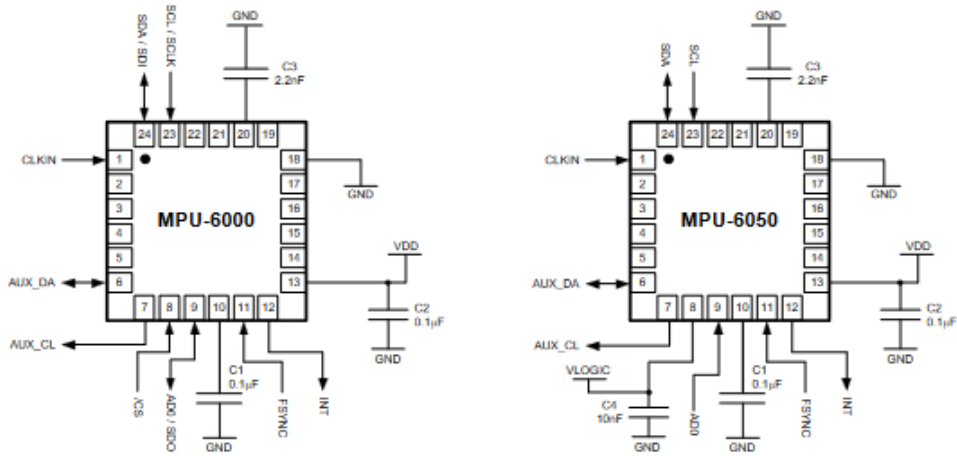
| PARAMETER                            | CONDITIONS   | MIN   | TYP        | MAX  | Units  | Notes |
|--------------------------------------|--|-------|------------|------|--------|-------|
| <b>TEMPERATURE SENSOR</b>            |  |       |            |      |        |       |
| Range                                |  |       | -40 to +85 |      | °C     |       |
| Sensitivity                          | Untrimmed  |       | 340        |      | LSB/°C |       |
| Temperature Offset                   | 35°C   |       | -521       |      | LSB    |       |
| Linearity                            | Best fit straight line (-40°C to +85°C)  |       | ±1         |      | °C     |       |
| <b>VDD POWER SUPPLY</b>              |  |       |            |      |        |       |
| Operating Voltages                   |  | 2.375 |            | 3.46 | V      |       |
| Normal Operating Current             | Gyroscope + Accelerometer + DMP  |       | 3.9        |      | mA     |       |
|                                      | Gyroscope + Accelerometer (DMP disabled)   |       | 3.8        |      | mA     |       |
|                                      | Gyroscope + DMP (Accelerometer disabled)   |       | 3.7        |      | mA     |       |
|                                      | Gyroscope only (DMP & Accelerometer disabled)  |       | 3.6        |      | mA     |       |
|                                      | Accelerometer only (DMP & Gyroscope disabled)  |       | 500        |      | µA     |       |
| Accelerometer Low Power Mode Current | 1.25 Hz update rate  |       | 10         |      | µA     |       |
|                                      | 5 Hz update rate   |       | 20         |      | µA     |       |
|                                      | 20 Hz update rate  |       | 70         |      | µA     |       |
|                                      | 40 Hz update rate  |       | 140        |      | µA     |       |
| Full-Chip Idle Mode Supply Current   |  |       | 5          |      | µA     |       |
| Power Supply Ramp Rate               | Monotonic ramp. Ramp rate is 10% to 90% of the final value                                     |       |            | 100  | ms     |       |
| <b>VLOGIC REFERENCE VOLTAGE</b>      |  |       |            |      |        |       |
| Voltage Range                        | MPU-6050 only  | 1.71  |            | VDD  | V      |       |
| Power Supply Ramp Rate               | VLOGIC must be ≤VDD at all times<br>Monotonic ramp. Ramp rate is 10% to 90% of the final value |       |            | 3    | ms     |       |
| Normal Operating Current             |  |       | 100        |      | µA     |       |
| <b>TEMPERATURE RANGE</b>             |  |       |            |      |        |       |
| Specified Temperature Range          | Performance parameters are not applicable beyond Specified Temperature Range                   | -40   |            | +85  | °C     |       |

## 7 Applications Information

### 7.1 Pin Out and Signal Description

| Pin Number                 | MPU-6000 | MPU-6050 | Pin Name   | Pin Description  |
|----------------------------|----------|----------|------------|--|
| 1                          | Y        | Y        | CLKIN      | Optional external reference clock input. Connect to GND if unused.       |
| 6                          | Y        | Y        | AUX_DA     | I <sup>2</sup> C master serial data, for connecting to external sensors  |
| 7                          | Y        | Y        | AUX_CL     | I <sup>2</sup> C Master serial clock, for connecting to external sensors |
| 8                          | Y        |          | /CS        | SPI chip select (0=SPI mode)   |
| 8                          |          | Y        | VLOGIC     | Digital I/O supply voltage   |
| 9                          | Y        |          | AD0 / SDO  | I <sup>2</sup> C Slave Address LSB (AD0); SPI serial data output (SDO)   |
| 9                          |          | Y        | AD0        | I <sup>2</sup> C Slave Address LSB (AD0)                                 |
| 10                         | Y        | Y        | REGOUT     | Regulator filter capacitor connection                                    |
| 11                         | Y        | Y        | FSYNC      | Frame synchronization digital input. Connect to GND if unused.           |
| 12                         | Y        | Y        | INT        | Interrupt digital output (totem pole or open-drain)                      |
| 13                         | Y        | Y        | VDD        | Power supply voltage and Digital I/O supply voltage                      |
| 18                         | Y        | Y        | GND        | Power supply ground  |
| 19, 21                     | Y        | Y        | RESV       | Reserved. Do not connect.  |
| 20                         | Y        | Y        | CPOUT      | Charge pump capacitor connection   |
| 22                         | Y        | Y        | RESV       | Reserved. Do not connect.  |
| 23                         | Y        |          | SCL / SCLK | I <sup>2</sup> C serial clock (SCL); SPI serial clock (SCLK)             |
| 23                         |          | Y        | SCL        | I <sup>2</sup> C serial clock (SCL)                                      |
| 24                         | Y        |          | SDA / SDI  | I <sup>2</sup> C serial data (SDA); SPI serial data input (SDI)          |
| 24                         |          | Y        | SDA        | I <sup>2</sup> C serial data (SDA)                                       |
| 2, 3, 4, 5, 14, 15, 16, 17 | Y        | Y        | NC         | Not internally connected. May be used for PCB trace routing.             |



**7.2 Typical Operating Circuit**

**Typical Operating Circuits**
**7.3 Bill of Materials for External Components**

| Component                           | Label | Specification                 | Quantity |
|-------------------------------------|-------|-------------------------------|----------|
| Regulator Filter Capacitor (Pin 10) | C1    | Ceramic, X7R, 0.1µF ±10%, 2V  | 1        |
| VDD Bypass Capacitor (Pin 13)       | C2    | Ceramic, X7R, 0.1µF ±10%, 4V  | 1        |
| Charge Pump Capacitor (Pin 20)      | C3    | Ceramic, X7R, 2.2nF ±10%, 50V | 1        |
| VLOGIC Bypass Capacitor (Pin 8)     | C4*   | Ceramic, X7R, 10nF ±10%, 4V   | 1        |

\* MPU-6050 Only.

### Anexo 3: Hoja de datos técnicos del sensor Flex 2.2”




## FLEX SENSOR FS

*Special Edition Length*

#### Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
  - Robotics
  - Gaming (Virtual Motion)
  - Medical Devices
  - Computer Peripherals
  - Musical Instruments
  - Physical Therapy
- Simple Construction
- Low Profile

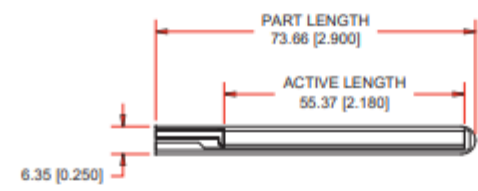
#### Mechanical Specifications

- Life Cycle: >1 million
- Height:  $\leq 0.43\text{mm}$  (0.017")
- Temperature Range: -35°C to +80°C

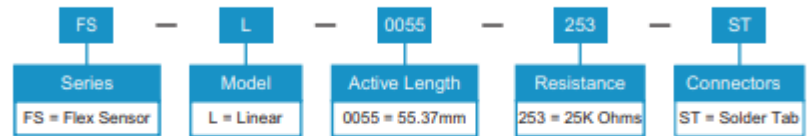
#### Electrical Specifications

- Flat Resistance: 25K Ohms
- Resistance Tolerance:  $\pm 30\%$
- Bend Resistance Range: 45K to 125K Ohms (depending on bend radius)
- Power Rating : 0.50 Watts continuous. 1 Watt Peak

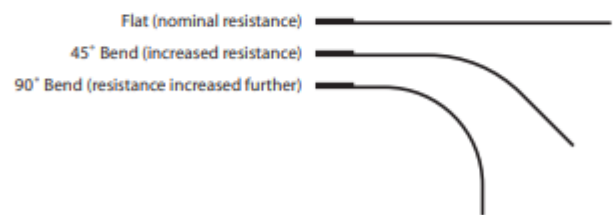
#### Dimensional Diagram - Stock Flex Sensor



#### How to Order - Stock Flex Sensor

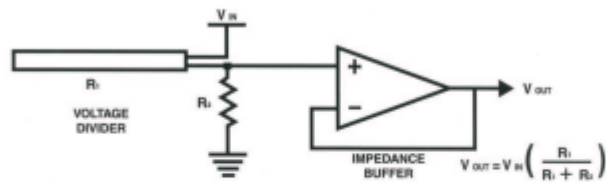


#### How It Works





## BASIC FLEX SENSOR CIRCUIT:



Following are notes from the ITP Flex Sensor Workshop

"The impedance buffer in the [Basic Flex Sensor Circuit] (above) is a single sided operational amplifier, used with these sensors because the low bias current of the op amp reduces error due to source impedance of the flex sensor as voltage divider. Suggested op amps are the LM358 or LM324."

"You can also test your flex sensor using the simplest circuit, and skip the op amp."

"Adjustable Buffer - a potentiometer can be added to the circuit to adjust the sensitivity range."

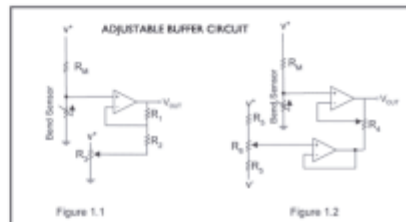
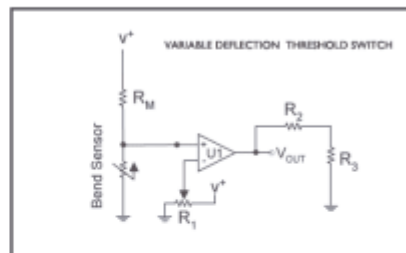


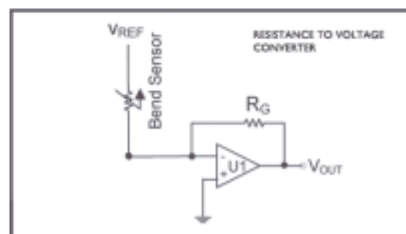
Figure 1.1

Figure 1.2

"Variable Deflection Threshold Switch - an op amp is used and outputs either high or low depending on the voltage of the inverting input. In this way you can use the flex sensor as a switch without going through a microcontroller."

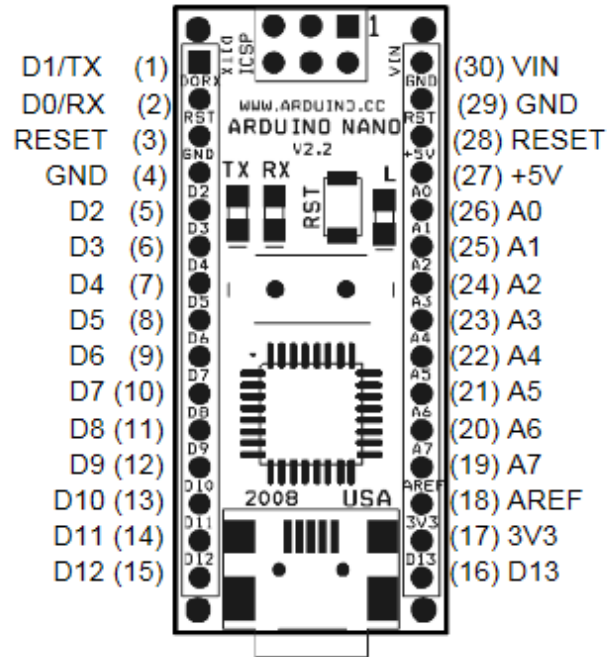


"Resistance to Voltage Converter - use the sensor as the input of a resistance to voltage converter using a dual sided supply op-amp. A negative reference voltage will give a positive output. Should be used in situations when you want output at a low degree of bending."



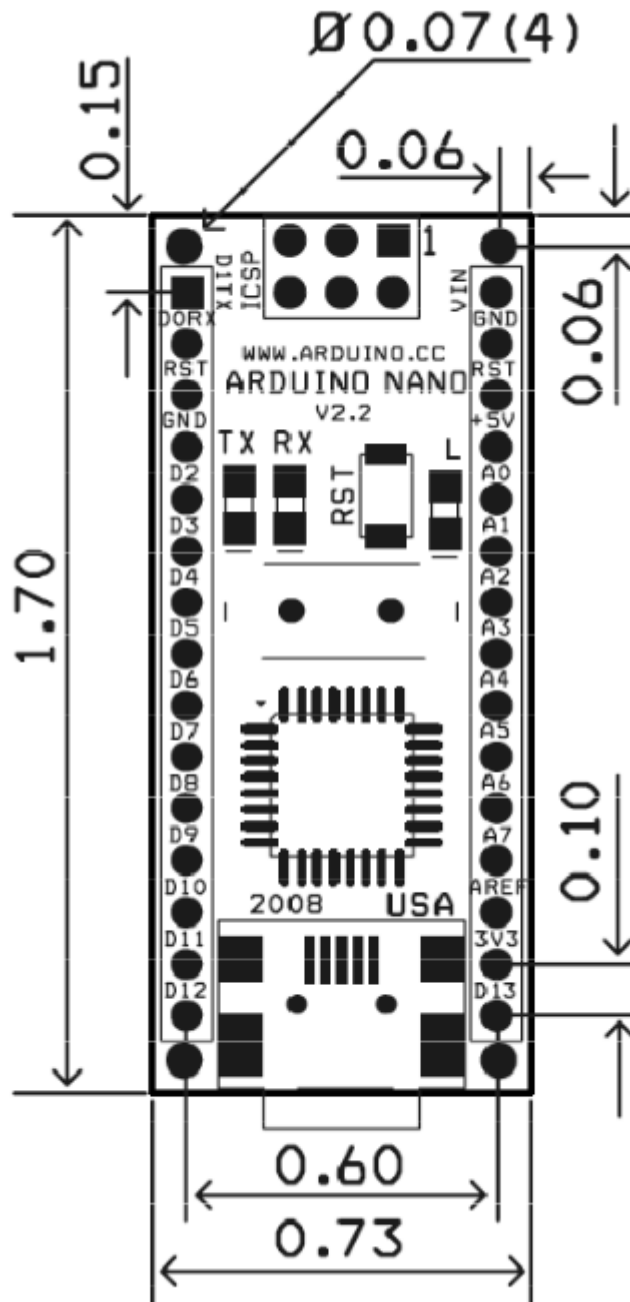
Anexo 4: Hoja de datos técnicos Placa Arduino Nano

**Arduino Nano Pin Layout**



| Pin No.   | Name   | Type            | Description  |
|-----------|--------|-----------------|--|
| 1-2, 5-16 | D0-D13 | I/O             | Digital input/output port 0 to 13  |
| 3, 28     | RESET  | Input           | Reset (active low)   |
| 4, 29     | GND    | PWR             | Supply ground  |
| 17        | 3V3    | Output          | +3.3V output (from FTDI)   |
| 18        | AREF   | Input           | ADC reference  |
| 19-26     | A7-A0  | Input           | Analog input channel 0 to 7  |
| 27        | +5V    | Output or Input | +5V output (from on-board regulator) or +5V (input from external power supply) |
| 30        | VIN    | PWR             | Supply voltage   |

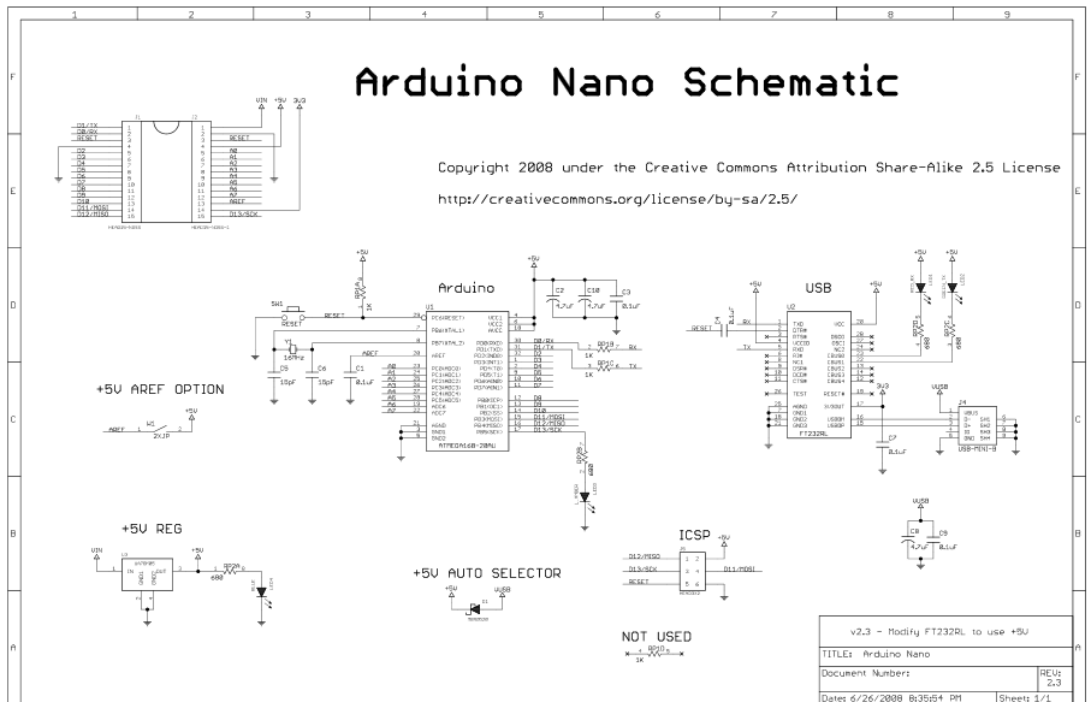
**Arduino Nano Mechanical Drawing**



ALL DIMENSIONS ARE IN INCHES

**Arduino Nano Bill of Material**

| Item Number | Qty. | Ref. Dest.     | Description  | Mfg. P/N          | MFG         | Vendor P/N         | Vendor  |
|-------------|------|----------------|--|-------------------|-------------|--------------------|---------|
| 1           | 5    | C1,C3,C4,C7,C9 | Capacitor, 0.1uF 50V 10% Ceramic X7R 0805                      | C0805C104K5RACTU  | Kemet       | 80-C0805C104K5R    | Mouser  |
| 2           | 3    | C2,C8,C10      | Capacitor, 4.7uF 10V 10% Tantalum Case A                       | T491A475K010AT    | Kemet       | 80-T491A475K010    | Mouser  |
| 3           | 2    | C5,C6          | Capacitor, 18pF 50V 5% Ceramic NOP/COG 0805                    | C0805C180J5GACTU  | Kemet       | 80-C0805C180J5G    | Mouser  |
| 4           | 1    | D1             | Diode, Schottky 0.5A 20V                                       | MBR0520LT1G       | ONsemi      | 863-MBR0520LT1G    | Mouser  |
| 5           | 1    | J1,J2          | Headers, 36PS 1 Row  | 68000-136HLF      | FCI         | 649-68000-136HLF   | Mouser  |
| 6           | 1    | J4             | Connector, Mini-B Recept Rt. Angle                             | 67503-1020        | Molex       | 538-67503-1020     | Mouser  |
| 7           | 1    | J5             | Headers, 72PS 2 Rows   | 67996-272HLF      | FCI         | 649-67996-272HLF   | Mouser  |
| 8           | 1    | LD1            | LED, Super Bright RED 100mcd 640nm 120degree 0805              | APT2012SRCPRV     | Kingbright  | 604-APT2012SRCPRV  | Mouser  |
| 9           | 1    | LD2            | LED, Super Bright GREEN 50mcd 570nm 110degree 0805             | APHCM2012CGCK-F01 | Kingbright  | 604-APHCM2012CGCK  | Mouser  |
| 10          | 1    | LD3            | LED, Super Bright ORANGE 160mcd 601nm 110degree 0805           | APHCM2012SECK-F01 | Kingbright  | 04-APHCM2012SECK   | Mouser  |
| 11          | 1    | LD4            | LED, Super Bright BLUE 80mcd 470nm 110degree 0805              | LTST-C170TBKT     | Lite-On Inc | 160-1579-1-ND      | Digikey |
| 12          | 1    | R1             | Resistor Pack, 1K +/-5% 62.5mW 4RES SMD                        | YC164-JR-071KL    | Yageo       | YC164J-1.0KCT-ND   | Digikey |
| 13          | 1    | R2             | Resistor Pack, 680 +/-5% 62.5mW 4RES SMD                       | YC164-JR-0768ORL  | Yageo       | YC164J-680CT-ND    | Digikey |
| 14          | 1    | SW1            | Switch, Momentary Tact SPST 150gf 3.0x2.5mm                    | B3U-1000P         | Omron       | SW1020CT-ND        | Digikey |
| 15          | 1    | U1             | IC, Microcontroller RISC 16kB Flash, 0.5kB EEPROM, 23 I/O Pins | ATmega168-20AU    | Atmel       | 556-ATMEGA168-20AU | Mouser  |
| 16          | 1    | U2             | IC, USB to SERIAL UART 28 Pins SSOP                            | FT232RL           | FTDI        | 895-FT232RL        | Mouser  |
| 17          | 1    | U3             | IC, Voltage regulator 5V, 500mA SOT-223                        | UA78M05CDCYRG3    | TI          | 595-UA78M05CDCYRG3 | Mouser  |
| 18          | 1    | Y1             | Crystal, 16MHz +/-20ppm HC-49/US Low Profile                   | ABL-16.000MHZ-B2  | Abracon     | 815-ABL-16-B2      | Mouser  |



## Anexo 5: Código Arduino NANO

```
#include <Wire.h>
#include <MPU6050.h>
MPU6050 mpu;
int flex_s[] = {A0, A1, A2, A3, A6};
int numfx = sizeof(flex_s) / sizeof(flex_s[0]);

void setup(){
  Serial.begin(9600);

  Wire.begin();
  mpu.initialize();
  mpu.setFullScaleGyroRange(MPU6050_GYRO_FS_250);
  mpu.setFullScaleAccelRange(MPU6050_ACCEL_FS_2);

  for (int i = 0; i < numfx; i++) {
    pinMode(flex_s[i], INPUT);
  }
}

void printSensorData() {
  for (int i = numfx - 1; i >= 0; i--) {
    Serial.print(analogRead(flex_s[i]));
    if (i != 0) {
      Serial.print(",");
    }
  }
  int16_t ax, ay, az;
  int16_t gx, gy, gz;
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  float axg = (float)ax / 16384.0;
  float ayg = (float)ay / 16384.0;
  float azg = (float)az / 16384.0;
  float gxds = (float)gx / 131.0;
  float gyds = (float)gy / 131.0;
  float gzds = (float)gz / 131.0;

  // Calcular los ángulos de pitch, roll y yaw
  float pitch = atan2(axg, sqrt(ayg * ayg + azg * azg)) * 180.0/PI;
  float roll = atan2(ayg, sqrt(axg * axg + azg * azg)) * 180.0/PI;
  float yaw = atan2(-gyds, gzds) * 180.0 / PI;

  Serial.print(",");
  Serial.print(roll);
  Serial.print(",");
  Serial.print(pitch);
  Serial.print(",");
  Serial.print(yaw);
  Serial.println();
}

void loop() {
  printSensorData();
  delay(800);
}
```

## Anexo 6: Código Python – Main de Streamlit

```
import streamlit as st

st.set_page_config(
    page_title="Hello",
    page_icon="👋",
)

# st.sidebar.success("Hola ")

st.markdown(
    """
    > ### Sistema automático de decodificación de lengua de señas
    ecuatoriano a comunicación verbal.
    ##### Objetivo General:
    - Construir un sistema automático de decodificación de lengua
    de señas ecuatoriano a comunicación
    verbal.
    ##### Objetivos Específicos:
    * Establecer el vocabulario básico del lenguaje de señas
    ecuatoriano basado en el "Diccionario
    Oficial de Lengua de Señas Ecuatoriana Gabriel Román",
    según la realidad actual de las
    personas con discapacidad.
    * Definir técnicas, tecnologías y herramientas a utilizarse en
    el desarrollo del sistema para
    aplicar en el algoritmo de aprendizaje automático.
    * Implementar el sistema automático de decodificación de
    lengua de señas ecuatoriano a
    comunicación verbal para el alfabeto y palabras básicas.
    * Validar el funcionamiento del sistema por medio de pruebas
    técnicas a personas con
    conocimiento sobre el lenguaje de señas ecuatoriano.
    """
)
```

## Anexo 7: Código de Python – Hoja de datos de Streamlit

```
import streamlit as st
from PIL import Image
from serial.tools.list_ports import comports
import plotly.graph_objs as go
import plotly.express as px
import pandas as pd
import serial
import csv
from gtts import gTTS
import numpy as np
from pydub import AudioSegment
from pydub.playback import play

st.set_page_config(
    page_title="DataFrame",
    page_icon="📊",
    layout="wide",
)

# Lista con los nombres de los archivos de imagen
img_alph = ['A', 'B', 'C', 'CH', 'D', 'E', 'F', 'G', 'H', 'I',
            'J', 'K', 'L',
            'LL', 'M', 'N', 'Ñ', 'O', 'P', 'Q', 'R', 'RR', 'S',
            'T', 'U', 'V',
            'W', 'X', 'Y', 'Z']

img_num = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']

img_words = ['Hola', 'Buenosdias', 'Bien', 'Mal', 'Gracias',
             'Amarillo',
             'Azul', 'Blanco', 'Negro', 'Chao']

# Lista vacía para almacenar objetos Image
img_list_alph = []
img_list_num = []
img_list_words = []

# Cargar cada imagen y añadirla a la lista
for name in img_alph:
    img = Image.open(f'img/alfabeto/{name}.PNG')
    img_list_alph.append(img)

for name in img_num:
    img = Image.open(f'img/numeros/{name}.PNG')
    img_list_num.append(img)

for name in img_words:
    img = Image.open(f'img/palabras/{name}.PNG')
    img_list_words.append(img)

col1, col2 = st.columns(2)

with col1:
```

```

st.title('📊 Data-Frame')
tab1, tab2, tab3 = st.tabs(["Alfabeto", "Números",
"Palabras"])

with tab1:
    c = st.container()

with tab2:
    c1 = st.container()

with tab3:
    c2 = st.container()

s_img = c.select_slider(
    '⬇️',
    options=img_alph)

s_img_num = c1.select_slider(
    '🔢',
    options=img_num)

s_img_words = c2.select_slider(
    '🗨️',
    options=img_words)

# Definir un diccionario que asocia cada letra con su imagen
correspondiente
img_dict = {name: img_list_alph[i] for i, name in
enumerate(img_alph)}
img_dict_num = {name: img_list_num[i] for i, name in
enumerate(img_num)}
img_dict_words = {name: img_list_words[i] for i, name in
enumerate(img_words)}

# Mostrar la imagen correspondiente a la letra ingresada por el
usuario
if s_img in img_dict:
    c.image(img_dict[s_img], caption=" ")
else:
    c.write("La letra ingresada no tiene una imagen asociada")

if s_img_num in img_dict_num:
    c1.image(img_dict_num[s_img_num], caption=" ", width=250)
else:
    c1.write("El numero ingresado no tiene una imagen asociada")

if s_img_words in img_dict_words:
    c2.image(img_dict_words[s_img_words], caption=" ")
else:
    c2.write("La palabra ingresada no tiene una imagen asociada")

with col2:
    st.title('🔌 Arduino')
    c_serial = st.container()

```



```

# Obtener una lista de los nombres de los puertos serie
disponibles
port_list = comports()
port_options = [port.device for port in port_list]

# Widget de selección para el puerto serie
port = st.sidebar.selectbox("Selecciona el puerto serie",
port_options)

baudio = st.sidebar.selectbox(
    "Select Baudio ", (4800, 9600, 19200, 38400, 57600, 115200),
index=1
)
# Conectarse al puerto serie seleccionado
ser = serial.Serial(port, baudio)

pSerial = c_serial.radio("PySerial", ("", ""),
horizontal=True)

tx = st.sidebar.text_input('abc',
max_chars=12,
placeholder="abc",
label_visibility="collapsed")

samples = st.sidebar.number_input('# Muestras',
min_value=1,
max_value=10,
value=5,
step=1)

btn = st.sidebar.button('Guardar datos')

sensor_data = [] # store data

if pSerial == "

```

```

values = line.split(",")
if len(values) == 8:
    a, b, c, d, e, f, g, h = values
    st.json({
        "Pulgar": a,
        "Indice": b,
        "Medio": c,
        "Anular": d,
        "Meñique": e,
        "Roll": f,
        "Pitch": g,
        "Yaw": h,
        "Letra": s_img
    })
else:
    st.write("📁 Guardando datos espere")

if btn:
    line = 0
    while line <= samples:
        getData = ser.readline()
        dataString = getData.decode('utf-8')
        data = dataString[0:][::-2]

        readings = data.split(",")
        readings_with_txt = readings + [str(tx)]

        sensor_data.append(readings_with_txt)
        # c_serial.json(sensor_data)
        line = line+1
    st.balloons()
    break
if pSerial == "Run" and not port:
    c_serial.info('conectar', icon="🔌")

fileName = "dataf.csv"
file = open(fileName, "a")

# create the CSV
with open(fileName, 'a', encoding='UTF8', newline='') as f:
    writer = csv.writer(f)
    writer.writerows(sensor_data)

file.close()

st.markdown('---')

df = pd.read_csv("dataf.csv")

with st.expander('📦 CSV'):

    sc = st.radio("📁", ("dataframe", "describe", "grafica"),
horizontal=True)
    st.write(df.shape)

```

```

if sc == "dataframe":
    search_term = st.text_input('🔍 Buscar')
    if search_term:
        df = df[df['salida'].str.contains(search_term)]
        data = df.to_dict(orient='records')
        st.dataframe(data, use_container_width=True)
    else:
        st.dataframe(df, use_container_width=True)
if sc == "describe":
    st.dataframe(df.describe(), use_container_width=True)
if sc == "grafica":

    colmn1, colmn2 = st.columns(2)

    with colmn1:

        # Agrupamos los datos por letra
        df_grouped = df.groupby('salida').mean()

        # Creamos un widget de selección para la letra
        letra_seleccionada = st.select_slider('Selecciona una
letra', df['salida'].unique())

        # Obtenemos los valores de los dedos para la letra
seleccionada
        valores_dedos = df_grouped.loc[letra_seleccionada]

        colores =
px.colors.qualitative.Pastel[:len(valores_dedos)]
        # Creamos un objeto de tipo 'bar' para cada dedo

        data = [go.Bar(x=valores_dedos.index,
                        y=valores_dedos.values,
                        marker_color=colores)]

        # Creamos la figura
        fig = go.Figure(data=data)

        # Configuramos el título y los nombres de los ejes
        fig.update_layout(title=f'Valores de los dedos para la
letra {letra_seleccionada}',
                           xaxis_title='Dedo',
                           yaxis_title='Valor')

        # Mostramos la figura en Streamlit
        st.plotly_chart(fig, use_container_width=True)
        st.text(valores_dedos)
    with colmn2:

        # Agrupamos los datos por letra
        df_grouped2 = df.groupby('salida').mean()

        # Creamos un widget de selección para la letra
        letra_seleccionada2 = st.select_slider('Selecciona una
', df['salida'].unique())

```

```

        # Obtenemos los valores de los dedos para la letra
seleccionada
        valores_dedos2 = df_grouped2.loc[letra_seleccionada2]

        colores2 =
px.colors.qualitative.Pastel[:len(valores_dedos2)]
        # Creamos un objeto de tipo 'bar' para cada dedo

        data2 = [go.Bar(x=valores_dedos2.index,
                        y=valores_dedos2.values,
                        marker_color=colores2)]

        # Creamos la figura
        fig2 = go.Figure(data=data2)

        # Configuramos el título y los nombres de los ejes
        fig2.update_layout(title=f'Valores de los dedos para
la letra {letra_seleccionada2}',
                        xaxis_title='Dedo',
                        yaxis_title='Valor')
        st.plotly_chart(fig2, use_container_width=True)
        st.text(valores_dedos2)

with st.expander('gTTS (Google Text-to-Speech)'):
    txt = st.text_input('Text:', 'Hola')
    if name != '':
        audio = gTTS(text=txt, lang="es", slow=False)

        audio.save("example.mp3")
        # os.system("start example.mp3")
        audio_file = open('example.mp3', 'rb')

        audio_bytes = audio_file.read()

        st.audio(audio_bytes, format='audio/ogg', start_time=0,)

```

## Anexo 8: Código Python – K-NN de Streamlit

```
import streamlit as st
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, f1_score
import serial
from serial.tools.list_ports import comports
import pandas as pd
import matplotlib.pyplot as plt
import json
import numpy as np

st.set_page_config(
    page_title="K-NN",
    page_icon="📄",
    layout="wide",
)

st.title('📄 k-Nearest Neighbors')
st.info(
    """
    El algoritmo k-NN es un algoritmo de aprendizaje supervisado utilizado en el campo del aprendizaje automático y la minería de datos. Es un algoritmo simple pero poderoso que se utiliza para clasificar objetos basándose en las características que tienen en común con otros objetos similares.
    """
)

with open("pages/knn.md", "r") as f:
    contenido_md = f.read()

with st.expander("Ver código"):
    st.markdown(contenido_md)
    st.markdown("""
    ### 💧 dividir el conjunto de datos en datos de entrenamiento y prueba
    """)
    code = """
    X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.2, random_state=0)
    """
    st.code(code, language='python')
    st.markdown("""
    ### 💧 Construyendo y entrenando el modelo.
    """)
    st.code("""
    knn = KNeighborsClassifier(n_neighbors = 1)
    knn.fit(X_train,y_train)
    """, language='python')
    col1, col2, col3 = st.columns(3)
    with col1:
        t_s = st.number_input('Test_size', value=0.2, step=0.1)
```

```

with col2:
    r_s = st.number_input('Random_state', value=0)

with col3:
    n_n = st.number_input('N_neighbors', value=5)
# -----
df = pd.read_csv("dataf.csv")

X = df.drop(['salida'], axis=1).values
y = df['salida'].values
# -----
if 'clicked' not in st.session_state:
    st.session_state.clicked = {1: False, 2: False}
def clicked(button):
    st.session_state.clicked[button] = True
st.button('Entrenar', on_click=clicked, args=[1])
if st.session_state.clicked[1]:
# -----
    X_train, X_test, y_train, y_test = train_test_split(X, y,

test_size=t_s,
random_state=r_s
                                                    )

    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

    # st.write(X_train)
    # st.write(X_test)
    classifier = KNeighborsClassifier(n_neighbors=n_n)

    classifier.fit(X_train, y_train)
    # evaluar el modelo utilizando la validación cruzada
    scores = cross_val_score(classifier, X, y, cv=5)
    st.write("📊 Cross-validation scores:")
    st.text(scores)
    st.write("📊 Average cross-validation score:")
    st.write(scores.mean())

    accuracy = classifier.score(X_test, y_test)

    ac = str(round(accuracy*100, 2))
    training_accuracy = []
    test_accuracy = []
    k_values = range(1, 11)
# -----
    cv_scores = []
    for k in k_values:
        knn = KNeighborsClassifier(n_neighbors=n_n)
        knn.fit(X_train, y_train)
        # Calcular la precisión de entrenamiento y prueba
        training_accuracy.append(knn.score(X_train, y_train))
        test_accuracy.append(knn.score(X_test, y_test))
        scores = cross_val_score(knn, X_train, y_train, cv=10,
scoring='accuracy')

```

```

        cv_scores.append(scores.mean())
    training_accuracy_json = json.dumps(training_accuracy)
    test_accuracy_json = json.dumps(test_accuracy)
    st.metric(label="^\ Accuracy", value=ac, delta="%")
    search_term = st.text_input('Buscar')

    if search_term:
        df = df[df['salida'].str.contains(search_term)]
        data = df.to_dict(orient='records')
        st.dataframe(data, use_container_width=True)

    r = st.radio("ssss", ('Manual', 'Serial Port'),
horizontal=True)
    if r == 'Manual':
        c1, c2, c3, c4, c5 = st.columns(5)

        with c1:
            pulgar = st.number_input('Pulgar',
                                     min_value=250,
                                     max_value=700,
                                     value=634,
                                     step=1)

        with c2:
            indice = st.number_input('Indice',
                                     min_value=250,
                                     max_value=700,
                                     value=664,
                                     step=1)

        with c3:
            corazon = st.number_input('Medio',
                                     min_value=250,
                                     max_value=700,
                                     value=603, step=1)

        with c4:
            anular = st.number_input('Anular',
                                     min_value=250,
                                     max_value=700,
                                     value=476,
                                     step=1)

        with c5:
            menique = st.number_input('Meñique',
                                     min_value=250,
                                     max_value=700,
                                     value=445,
                                     step=1)

    cc1, cc2, cc3, cc4 = st.columns([1, 1, 1, 2])
    nx = cc1.number_input('ang_x')
    ny = cc2.number_input('ang_y')
    nz = cc3.number_input('ang_z')

    X_new = scaler.transform([[pulgar, indice, corazon,
anular, menique, nx, ny, nz]])

    st.button('Predecir', on_click=clicked, args=[2])
    if st.session_state.clicked[2]:

```

```

        classifier.fit(X_train, y_train)
        letra_predicha = classifier.predict(X_new)
        st.write(letra_predicha)
    else:
        st.write("Arduino")

        port_list = comports()
        port_options = [port.device for port in port_list]

        # Widget de selección para el puerto serie
        port = st.sidebar.selectbox("Selecciona el puerto serie",
port_options)

        baudio = st.sidebar.selectbox(
            "Select Baudio ▾", (4800, 9600, 19200, 38400, 57600,
115200), index=1
        )
        ser = serial.Serial(port, baudio)
        check = st.checkbox('Read')
        if check and port:
            st.write('RUN')
            data = st.empty()
            while True:
                with data.container():
                    line = ser.readline().strip().decode("utf-8")
                    a, b, c, d, e, f, g, h = line.split(",")
                    st.json({
                        "Pulgar": a,
                        "Indice": b,
                        "Medio": c,
                        "Anular": d,
                        "Meñique": e,
                        "ang_x": f,
                        "ang_y": g,
                        "ang_z": h
                    })
                valores = [float(valor) for valor in
line.split(',')]

                X_new = [valores]
                #st.text(X_new)
                letra_predicha = classifier.predict(X_new)
                st.write(letra_predicha)
        else:
            st.info('💡 Conect to arduino')

```



## Anexo 9: Código Python – Random Forest de Streamlit

```
import streamlit as st
import time
import numpy as np
import altair as alt
import plotly.graph_objs as go
import plotly.express as px
import pickle
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

st.set_page_config(
    page_title="Random Forest",
    page_icon="📁",
    layout="wide",)

st.title('📁 Random Forest Classifier')
st.info(
    """
    Consiste en una gran cantidad de árboles de decisión
    individuales que
    operan como un conjunto. Cada árbol individual en el
    bosque aleatorio
    escupe una predicción de clase y la clase con más votos se
    convierte
    en la predicción de nuestro modelo
    """
)

# Create a key in session state to track the stage
if 'stage' not in st.session_state:
    st.session_state.stage = 0

# Stage function to update the stage saved in session state
def set_stage(stage):
    st.session_state.stage = stage

#st.write('Welcome! Click to begin.')

file = st.file_uploader("Choose a file")
# Each button runs the Stage function, passing the stage number as
an argument
st.button('😊', on_click=set_stage, args=[1])

# Content for each stage within the body of an if statement
if st.session_state.stage > 0 and file:

    df = pd.read_csv(file)
```

```

with st.expander("📄 DataFrame"):
    st.dataframe(df, use_container_width=True)

    st.markdown('### 💧 Dividir los datos en conjunto de
entrenamiento y conjunto de prueba')
    X = df.drop('salida', axis=1).values
    y = df['salida'].values

    st.code('''
        X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.2, random_state=10)
        ''', language='python')
    c11, c12, c13 = st.columns([1,1,3])
    with c11:
        ts = st.number_input('Test Size',
                             min_value=0.0,
                             max_value=20.0,
                             value=0.2,
                             step=0.1)

    with c12:
        rs = st.number_input('Random_state', value=10, step=1)

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=ts, random_state=rs)

    st.write('Datos de entrenamiento: ', X_train.shape,)
    with st.expander("👁 Ver datos"):
        st.dataframe(X_train, use_container_width=True)
    st.write('Datos de prueba: ', X_test.shape)
    with st.expander("👁 Ver datos "):
        st.dataframe(X_test, use_container_width=True)

    st.button('▶▶', on_click=set_stage, args=[2])
if st.session_state.stage > 1:
    st.markdown('### 💧 Crear un modelo de Random Forest')
    st.code('''
        model = RandomForestClassifier(n_estimators=100,
random_state=10)
        ''', language='python')
    model = RandomForestClassifier(n_estimators=100,
random_state=10)
    model.fit(X_train, y_train)

    c1, c2, c3, c4, c5 = st.columns(5)

    with c1:
        pulgar = st.number_input('Pulgar',
                                 min_value=250,
                                 max_value=700,
                                 value=626,
                                 step=1)

    with c2:
        indice = st.number_input('Indice',
                                 min_value=250,
                                 max_value=700,

```

```

        value=271,
        step=1)

with c3:
    corazon = st.number_input('Medio',
                               min_value=250,
                               max_value=700,
                               value=359,
                               step=1)

with c4:
    anular = st.number_input('Anular',
                              min_value=250,
                              max_value=700,
                              value=397,
                              step=1)

with c5:
    menique = st.number_input('Meñique',
                               min_value=250,
                               max_value=700,
                               value=336,
                               step=1)

cc1, cc2, cc3, cc4 = st.columns([1, 1, 1, 2])
nx = cc1.number_input('ang_x')
ny = cc2.number_input('ang_y')
nz = cc3.number_input('ang_z')

st.button('😊', on_click=set_stage, args=[3])
if st.session_state.stage > 2:
    st.write('This is the end. Thank you!')

X_new = [[pulgar, indice, corazon, anular, menique, nx, ny,
nz]]
predecir = model.predict(X_new)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)*100
st.write(predecir)
st.write(accuracy)
if st.button('📁 Guardar Modelo: '):
    st.write('Modelo guardado')
    # Guardar el modelo en un archivo
    with open('modelo_rf.pkl', 'wb') as archivo:
        pickle.dump(model, archivo)
    st.balloons()

st.button('👍', on_click=set_stage, args=[0])

```

## Anexo 10: Código Python – Lengua de Señas Ecuatoriana de Streamlit

```
import random
import time
import streamlit as st
from PIL import Image
import pickle
from serial.tools.list_ports import comports
import serial
import pytttsx3
import numpy as np
from gtts import gTTS
from pydub import AudioSegment
from pydub.playback import play
import threading

st.set_page_config(
    page_title="Ex-stream-ly Cool App",
    page_icon="📺",
    layout="wide",
    initial_sidebar_state="expanded",
)

#
img_static = ['A', 'B', 'C', 'CH', 'D', 'E', 'F', 'G', 'H', 'I',
              'K', 'L', 'M',
              'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X',
              'Y']

img_numbers = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']

img_dynamic = ['Hola', 'Buenosdias', 'Bien', 'Mal', 'Gracias',
               'Amarillo',
               'Azul', 'Blanco', 'Negro', 'Chao', 'J', 'LL', 'RR',
               'Z', 'Ñ']

img_list_static = [Image.open(f'img/alfabeto/{name}.PNG') for name
in img_static]
img_list_numbers = [Image.open(f'img/numeros/{name}.PNG') for name
in img_numbers]
img_list_dynamic = [Image.open(f'img/palabras/{name}.PNG') for
name in img_dynamic]

st.title('📺 Lengua de Señas Ecuatoriana)

tb1, tb2, tb3 = st.tabs(["^ Gestos Estáticos", "☑ Gestos
Dinamicos", "# Numbers"])

with tb1:
    st.code(img_static)
    with st.expander("🧐"):
        cn1 = st.container()

with tb2:
```

```

st.code(img_dynamic)
with st.expander("See explanation"):
    cn2 = st.container()

with tb3:
    st.code(img_numbers)
    with st.expander("See explanation"):
        cn3 = st.container()

s_img_s = cn1.select_slider(
    '',
    options=img_static)

s_img_d = cn2.select_slider(
    '',
    options=img_dynamic)

s_img_n = cn3.select_slider(
    '',
    options=img_numbers)

#
img_dict_s = {name: img_list_static[i] for i, name in
enumerate(img_static)}
img_dict_num = {name: img_list_numbers[i] for i, name in
enumerate(img_numbers)}
img_dict_d = {name: img_list_dynamic[i] for i, name in
enumerate(img_dynamic)}

if s_img_s in img_dict_s:
    cn1.image(img_dict_s[s_img_s], caption=" ")
else:
    cn1.write("La letra ingresada no tiene una imagen asociada")

if s_img_d in img_dict_d:
    cn2.image(img_dict_d[s_img_d], caption=" ")
else:
    cn2.write("La palabra ingresada no tiene una imagen asociada")

if s_img_n in img_dict_num:
    cn3.image(img_dict_num[s_img_n], caption=" ")
else:
    cn3.write("La palabra ingresada no tiene una imagen asociada")


r = st.sidebar.text("■■■■ 0 0 0 0 0 0 ■■■■")

port_list = comports()
port_options = [port.device for port in port_list]

# Widget de selección para el puerto serie
port = st.sidebar.selectbox("Selecciona el puerto serie",
port_options)

```

```

baudio = st.sidebar.selectbox(
    "Select Baudio ", (4800, 9600, 19200, 38400, 57600,
115200), index=1
)

ser = serial.Serial(port, baudio)








# Definir el diccionario de opciones
opciones = {
    'Static': {'archivo': 'rf_wndf.pkl', 'mensaje': 'Gestos
Estaticos'},
    'Dynamic': {'archivo': 'rf_wd.pkl', 'mensaje': 'Gestos
Dinámicos'},
    'Numbers': {'archivo': 'rf_num.pkl', 'mensaje': 'Numeros'}
}

# Mostrar el radio button para seleccionar la opción
option = st.sidebar.radio("Seleccione una opción",
list(opciones.keys()))

# Verificar la opción seleccionada
if option in opciones:
    # Obtener el nombre de archivo y el mensaje correspondientes
    archivo = opciones[option]['archivo']
    mensaje = opciones[option]['mensaje']

    # Cargar el modelo desde el archivo
    with open(archivo, 'rb') as f:
        rf = pickle.load(f)

    # Mostrar el mensaje en la barra lateral
    st.sidebar.info(mensaje)
else:
    st.sidebar.error('Opción inválida')

check = st.checkbox('   ')
if check and port:
    st.write('#   ')
    data = st.empty()
    while True:
        with data.container():
            line = ser.readline().strip().decode("utf-8")
            valores = [float(valor) for valor in line.split(',')]
            X_new = [valores]
            st.text(X_new)
            letra_predicha = rf.predict(X_new)
            st.metric(label="  ", value=str(letra_predicha),
delta="  ")
            #st.text(letra_predicha)
    else:
        st.sidebar.info('  Conect to arduino')

@st.cache_resource
def generar_y_reproducir_audio(texto):

```

```

audio = gTTS(text=texto, lang="es", slow=False)
audio.save("example.mp3")

audio_file = open('example.mp3', 'rb')
audio_f = "example.mp3"
sound = AudioSegment.from_file(audio_f, format="mp3")
play(sound)

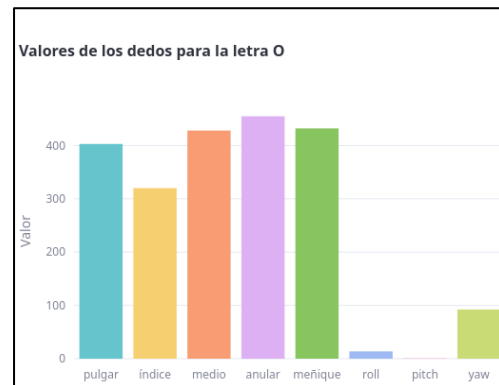
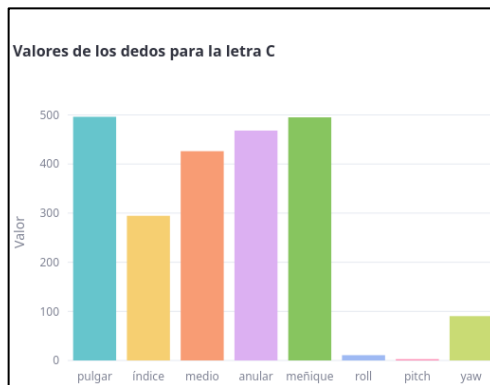
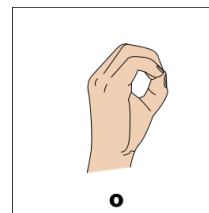
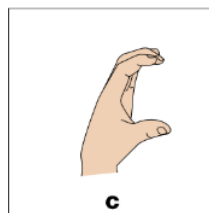
chk = st.checkbox('🔊')
if chk and port:
    st.write('# 📄')
    data = st.empty()
    button = st.button('# 🗣️')
    #while True:
    if button:
        with data.container():
            line = ser.readline().strip().decode("utf-8")
            valores = [float(valor) for valor in line.split(',')]
            X_new = [valores]
            st.text(X_new)
            letra_predicha = rf.predict(X_new)
            st.write(letra_predicha)
            txt = letra_predicha[0]
            # Generar y reproducir el audio en un hilo separado
            audio_thread =
            threading.Thread(target=generar_y_reproducir_audio, args=(txt,))
            audio_thread.start()
            st.cache_resource.clear()

```

## Anexo 11: Análisis similitudes físicas

### Letras C y O

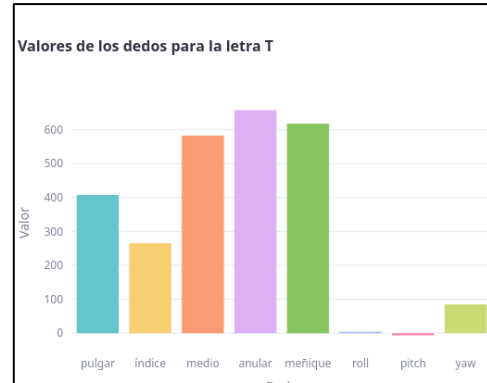
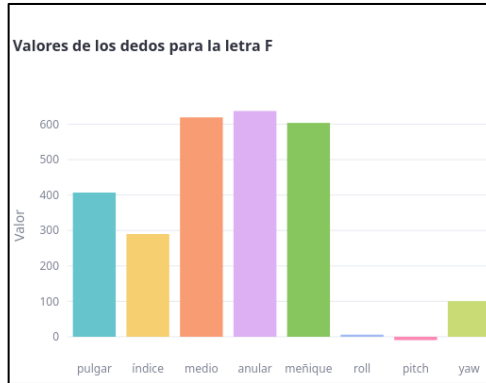
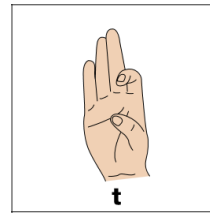
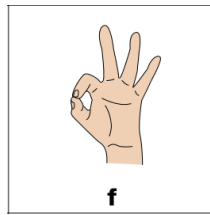
No hay manera de determinar con los sensores Flex cuando el pulgar toca la punta de los demás dedos, por eso un factor clave para diferenciar la letra C de la O. Los sensores de los dedos pulgar y meñique, como se muestra en el grafico presentan un valor de diferencia entre el  $\pm 93.11$  y  $\pm 62.84$  respectivamente, por lo tanto, su identificación se puede realizar sin ningún conflicto, únicamente es necesario colocar los dedos de manera correcta.



### Letras F y T

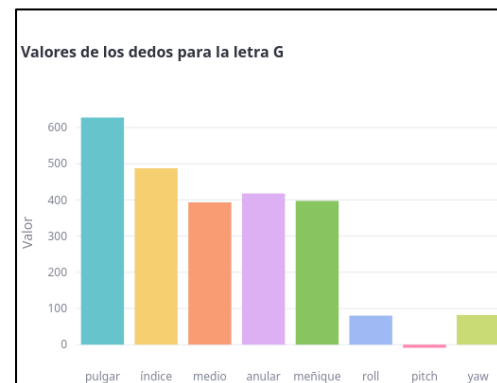
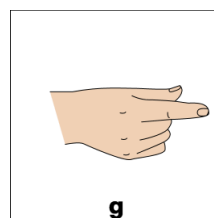
Los sensores Flex no pueden detectar cuando los dedos están juntos o separados respecto al dedo medio. Por lo tanto, para diferenciar la letra F de la T, el valor del dedo índice es el factor clave. Para la letra F, la lectura del sensor es aproximadamente 300, por el contrario, la letra T al no requerir una mayor flexión, el valor ronda por 250. Esto permite diferenciar estas letras sin ningún problema.

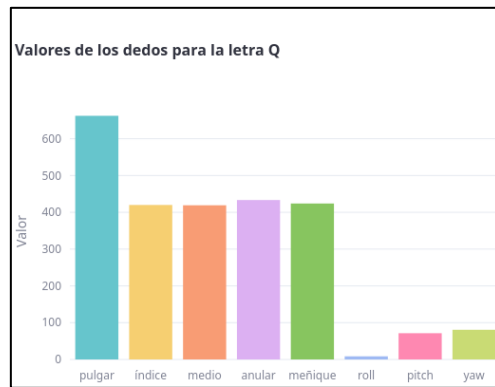
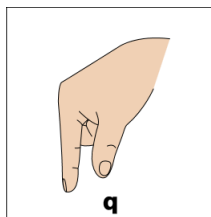
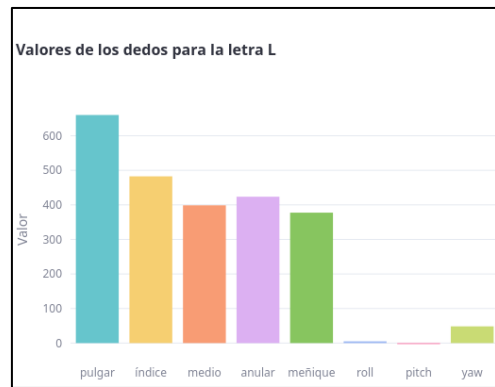
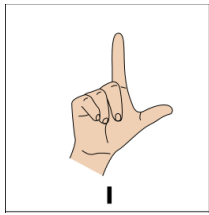




### Letras G, L, Q

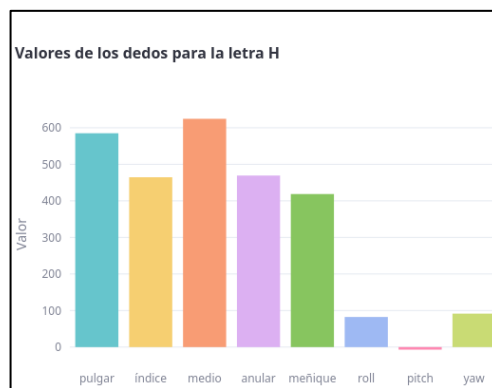
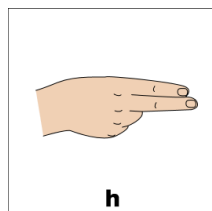
Los sensores colocados en los dedos pulgar e índice son los que están extendidos en su totalidad en cada letra, visualmente se pueden diferenciar, sin embargo, los sensores Flex no detectan si los dedos están juntos o separados, por lo tanto, el sensor MPU6050 permite identificar cada letra debido a su orientación angular, los ángulos roll y pitch son la clave para diferenciar las letras G y Q de la letra L.

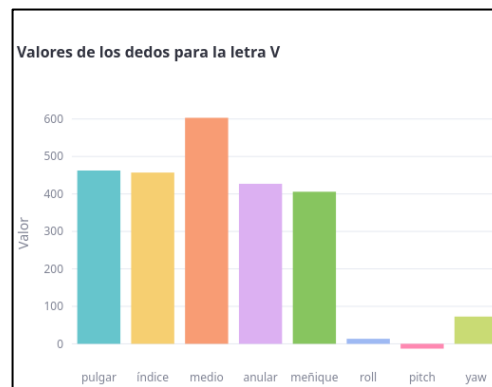
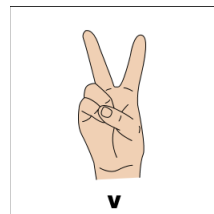
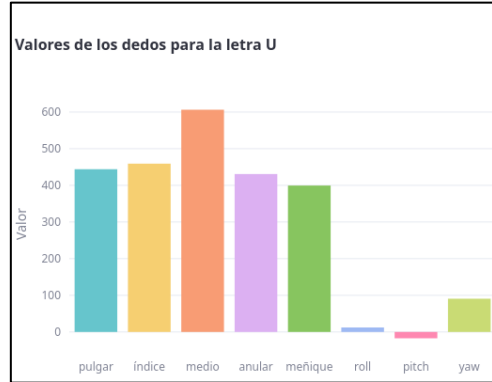
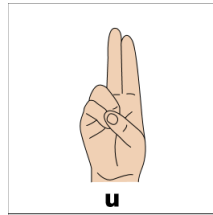




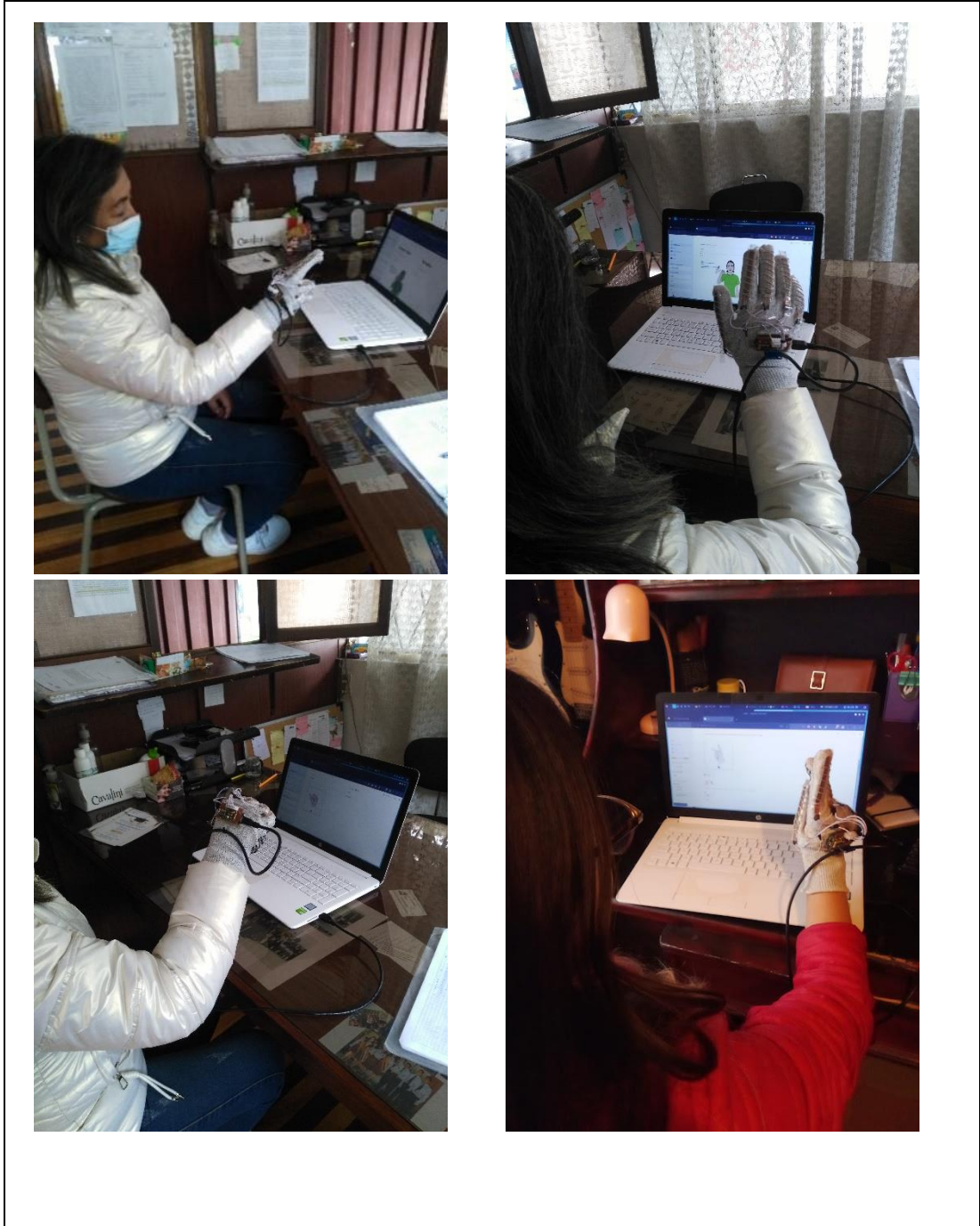
### Letras H, U, V

Estas letras presentan un conflicto, debido a que su posición de dedos es igual, la única diferencia es la separación de los dedos índice y medio, sin embargo, gracias a la ubicación de los sensores Flex desde la parte de la falange media de los dedos, se presenta una mínima variación como se muestra en los gráficos, permitiendo identificar las letras U y V. Para la letra H se identifica con el ángulo de inclinación en roll.





Anexo 12: Pruebas de funcionamiento







## Vocabulario básico del sistema

En este apartado se ilustra los gestos que conforman el sistema automático de decodificación de lengua de señas ecuatoriano.

### Alfabeto Dactilológico Ecuatoriano:

|          |           |          |           |          |
|----------|-----------|----------|-----------|----------|
|          |           |          |           |          |
| <b>a</b> | <b>b</b>  | <b>c</b> | <b>ch</b> | <b>d</b> |
|          |           |          |           |          |
| <b>e</b> | <b>f</b>  | <b>g</b> | <b>h</b>  | <b>i</b> |
|          |           |          |           |          |
| <b>j</b> | <b>k</b>  | <b>l</b> | <b>ll</b> | <b>m</b> |
|          |           |          |           |          |
| <b>n</b> | <b>ñ</b>  | <b>o</b> | <b>p</b>  | <b>q</b> |
|          |           |          |           |          |
| <b>r</b> | <b>rr</b> | <b>s</b> | <b>t</b>  | <b>u</b> |
|          |           |          |           |          |
| <b>v</b> | <b>w</b>  | <b>x</b> | <b>y</b>  | <b>z</b> |

## Palabras dinámicas:



**Buenos Días**  
Expresión de saludo usado para desear a alguien que tenga una día agradable. La mano derecha o izquierda se levanta hacia arriba y después hacia abajo mientras se dice o escuchamos.



**Pelo**  
Expresión de confusión o sorpresa al hablar. La mano con los dedos juntos se alza y se abren repetidas veces.



**Gracias**  
Expresión usada para demostrar agradecimiento. La mano se levanta del nivel bajo al pecho cuando el usuario habla.



**Chao**  
Compañía usada como despedida de alguien. La mano se mueve hacia arriba y hacia abajo repetidamente.



**Bien**  
De manera correcta. Expresión usada para indicar que alguien se encuentra satisfecho con una situación. La mano se mueve lentamente hacia adelante con los dedos.



**Moi**  
Refiriendo a una cosa incómoda. Expresión usada con un objeto o a alguien no sea cómodo con una situación. La mano se levanta hacia abajo. El dedo se mueve y se sacude un poco con los dedos hacia arriba.



**amarrillo, rojo, az**  
Del color amarillo. **rojo, az**  
Persona que tiene cabello claro o amarillo. La mano se levanta de adelante hacia atrás de la cabeza en forma ondulatoria.



**azul**  
Del color del cielo. La mano se levanta por la magia.



**Mancha**  
Del color de la mancha. La mano sobre nosotros en alto. Se hace un giro.



**negro, a**  
Del color de la oscuridad absoluta. La mano se alza y se mueve.

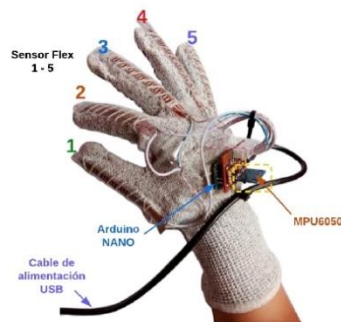
MIDO

## Números:



## Prototipo guante electrónico

En este apartado se explica las instrucciones de uso del prototipo del guante electrónico para traducir la lengua de señas



### Descripción:

El guante electrónico está conformado por 5 sensores flexibles, una placa Arduino NANO, un circuito electrónico, un sensor inercial MPU6050, y un cable USB para conectar al ordenador.

### Modo de empleo:

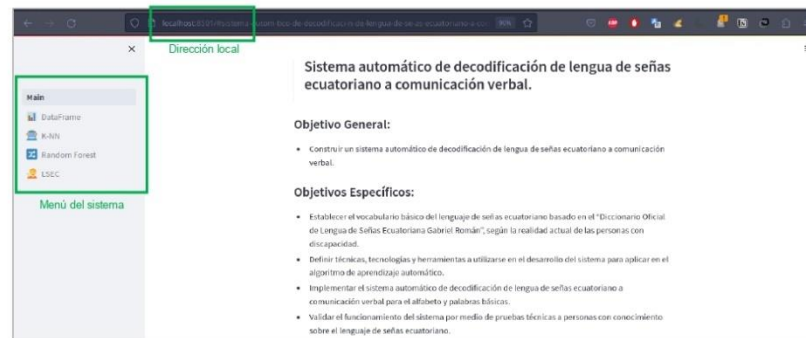
El usuario debe colocarse el prototipo en la mano derecha asegurándose que pueda mover sin dificultad los dedos, y que el cable de alimentación este extendido para poder conectarlo al ordenador.

## Aplicación web local

En este apartado se explica las instrucciones de uso de la interfaz gráfica basada en web.

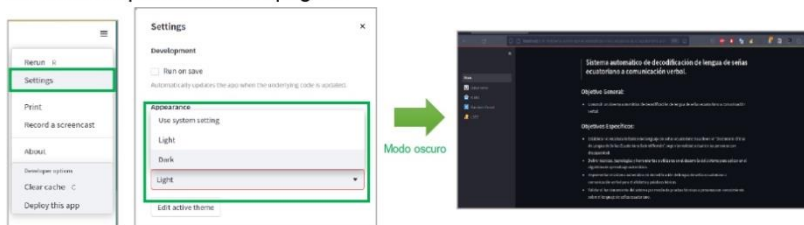
### Página menú principal:

Para acceder a la aplicación es necesario iniciar el servicio de Streamlit mediante el comando Streamlit run Main.py, y mediante el uso de un navegador se abrirá el servicio de manera local con la ruta: <http://localhost:8501>. La página inicia con el menú principal, donde se encuentra información sobre el proyecto, en la parte izquierda se enlista las demás páginas de la aplicación, cada uno representado con un icono.



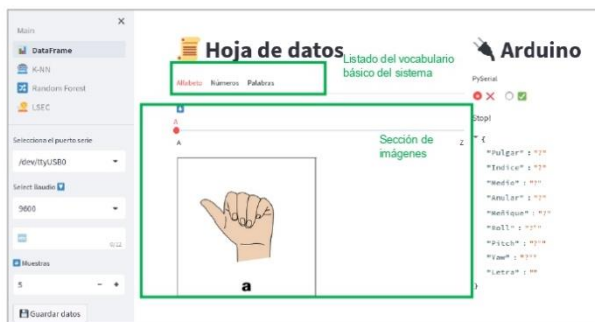


En la parte superior derecha se encuentra los ajustes, donde el usuario puede cambiar la apariencia de la página.



### Página Hoja de datos:

En este apartado, se presenta la hoja de datos, donde el usuario puede registrar nuevos gestos del vocabulario básico del sistema. En la parte superior se observa las categorías a la que el usuario puede acceder, como es el Alfabeto, Números y Palabras.



Para iniciar una captura de un gesto, el usuario debe seleccionar que seña desea guardar de acuerdo con el listado presentado, seguidamente se verifica la conexión del guante electrónico al ordenador en la parte izquierda, seguido del nombre del gesto y el número de muestras.



Verificado la conexión, el gesto y el número de muestras, se inicia la lectura seleccionando la casilla con el visto verde "✓", Y finalmente para guardar el nuevo registro se presiona el botón Guardar datos. Para detener el sistema se selecciona el icono "✗"

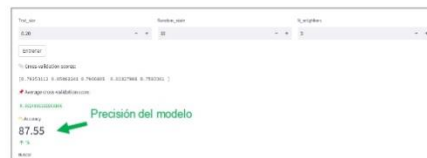


### Página KNN:

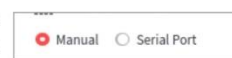
En este apartado se encuentra información del modelo K-Nearest Neighbors, se puede ver el código con el que trabaja y los pasos para entrenar el modelo. Permite al usuario familiarizarse con el funcionamiento de técnicas de clasificación del aprendizaje automático.



Para entrenar el modelo se especifica el número de vecinos y el estado aleatorio, el usuario puede ir probando valores, los mismos que mostrarán que tan preciso está el modelo al presionar el botón "Entrenar".



De acuerdo con el modelo entrenado, el usuario puede probar la precisión de manera manual o por el puerto serie el cual corresponde a la lectura del guante electrónico.



**Forma manual:** Se presiona el botón "Buscar", se desglosa una tabla que contiene los datos de los gestos, en la parte inferior se colocan cada valor correspondiente a las casillas de los nombres de los dedos. Finalmente, al presionar el botón "Predecir" se mostrará el gesto seleccionado.



**Forma automática:** Se inicia la lectura del puerto serie donde se mostrará los valores de cada dedo y los ángulos de movimiento, mostrando en la parte final la predicción de acuerdo con el modelo KNN entrenado.



### Página Random Forest:

En este apartado se encuentra información del modelo Random Forest, se puede ver el código con el que trabaja y los pasos para entrenar el modelo. Permite al usuario familiarizarse con el funcionamiento de técnicas de clasificación del aprendizaje automático.

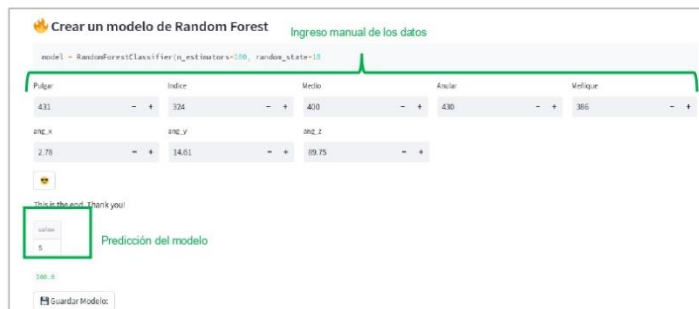


El modelo está dividido con un 80% de entrenamiento y un 20% de prueba. Para visualizar los valores se debe presionar la barra con el mismo nombre "Ver datos".

Para iniciar el modelo, el usuario puede colocar de manera manual los valores de acuerdo con las tablas.



Para iniciar la predicción del modelo se debe ingresar de manera manual los valores de posición de los dedos y los ángulos de acuerdo con la tabla de prueba, y para iniciar se debe presionar el botón con el emoji "👉", de esta manera se presentará la predicción. Por último, para guardar el modelo se debe dar clic en el botón con el mismo nombre "📄 Guardar Modelo".



### Página Lengua de Señas Ecuatoriana:

En este apartado se encuentra el sistema final, donde se puede traducir o aprender la lengua de señas de acuerdo con el vocabulario establecido. Se debe seleccionar en la parte superior la categoría que se desee aprender/traducir, en este apartado se cuenta con los títulos: Gestos Estáticos, que corresponden al alfabeto; los Gestos Dinámicos, que corresponde a las Palabras básicas; y los números. a continuación, se verifica que este seleccionada la casilla correspondiente al tipo de gesto que se realizara como se observa en la parte inferior izquierda.



Para visualizar los gestos, se debe hacer clic en la barra con el emoji “👋”, se desplegará la lista de imágenes de acuerdo con los gestos.



Para predecir se tiene dos opciones, salida de texto representado con el icono “🗨️” y salida de audio, representado con el icono “🔊”. Para iniciar el sistema se debe dar clic en cualquiera de ellas.

Para la salida de audio se debe tener en cuenta el botón con el icono “🔊” para que se pueda escuchar la traducción. A continuación, se presentan las salidas correspondientes a cada categoría del vocabulario básico del sistema.

- Gestos Estáticos



- Gestos Dinámicos

Salida de texto

The screenshot shows a software interface with a title bar containing 'Gestos Estáticos', 'Gestos Dinámicos', and 'Numeros'. Below the title bar is a list of gestures: ['Hola', 'Buenos dias', 'Bien', 'Mal', 'Gracias', 'Amarillo', 'Az...']. A 'See explanation' button is visible. The main area contains a 'Casilla salida de texto' (text output box) with a green arrow pointing to it, a 'Respuesta' (response) field containing the text ['HOLA!'], and a 'Numeros' field with the value '[[625.0, 294.0, 417.0, 483.0, 415.0, -4.07, -32.6, 69.69]]'.

Salida de audio

The screenshot shows a software interface with a title bar containing 'Gestos Estáticos', 'Gestos Dinámicos', and 'Numeros'. Below the title bar is a list of gestures: ['Hola', 'Buenos dias', 'Bien', 'Mal', 'Gracias', 'Amarillo', 'Az...']. A 'See explanation' button is visible. The main area contains a 'Casilla salida de audio' (audio output box) with a green arrow pointing to it, a 'Respuesta' (response) field containing the text 'HOLA', and a 'Numeros' field with the value '[[108.0, 267.0, 355.0, 436.0, 384.0, 7.7, 7.34, 69.83]]'.

- Números

Salida de texto

The screenshot shows a software interface with a title bar containing 'Gestos Estáticos', 'Gestos Dinámicos', and 'Numeros'. Below the title bar is a list of gestures: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']. A 'See explanation' button is visible. The main area contains a 'Casilla salida de texto' (text output box) with a green arrow pointing to it, a 'Respuesta' (response) field containing the text '[1]', and a 'Numeros' field with the value '[[440.0, 459.6, 375.0, 376.0, 402.4, 4.41, -11.51, 35.90]]'.

Salida de audio

The screenshot shows a software interface with a title bar containing 'Gestos Estáticos', 'Gestos Dinámicos', and 'Numeros'. Below the title bar is a list of gestures: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']. A 'See explanation' button is visible. The main area contains a 'Casilla salida de audio' (audio output box) with a green arrow pointing to it, a 'Respuesta' (response) field containing the text '1', and a 'Numeros' field with the value '[[249.0, 451.0, 455.0, 464.0, 429.0, 5.5, -15.44, 25.91]]'.