**UNIVERSIDAD DE TARAPACÁ**



**FACULTAD DE INGENIERÍA**

Departamento de Ingeniería en Computación e Informática



**Guante Traductor**

**de**

**Lenguaje de Señas (G.T.S.)**

**Autor(es): Ivan Callasaya**

**Jorge Gutierrez**

**Fabian Flores**

**Asignatura: Proyecto 2**

**Profesor(es): Diego Aracena**

ARICA, 12 de Diciembre 2022

**Historial de Cambios**

| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| --- | --- | --- | --- |
| 23/08/2022 | 1.0 | Versión preliminar del proyecto. | Ivan Callasaya  Jorge Gutierrez  Fabián Flores |
| 13/09/2022 | 1.1 | Desarrollo del panorama general y referencias.  Modificación del informe del plan de proyecto. | Ivan Callasaya  Jorge Gutierrez  Fabián Flores |
| 15/09/22 | 1.2 | Revisión y entrega de la versión final del informe de avance. | Ivan Callasaya  Jorge Gutierrez  Fabián Flores |
| 11/10/22 | 2.0 | Desarrollo del análisis del proyecto. | Ivan Callasaya  Jorge Gutierrez  Fabián Flores |
| 12/10/22 | 2.1 | Desarrollo de los modelos de diseño y requerimientos funcionales y no funcionales. | Ivan Callasaya  Jorge Gutierrez  Fabián Flores |
| 13/10/22 | 2.2 | Descripción de la arquitectura, y desarrollo de la IGU. | Ivan Callasaya  Jorge Gutierrez  Fabián Flores |
| 13/10/22 | 2.3 | Corrección de los errores del primer informe. | Fabián Flores |
| 29/11/22 | 3.0 | Inicio desarrollo informe final  y actualización de la Carta Gantt. | Jorge Guitierrez  Fabián Flores |
| 30/11/22 | 3.1 | Desarrollo de planificación de procesos de soporte e implementación del proyecto. | Ivan Callasaya  Jorge Gutierrez  Fabián Flores |
| 10/12/22 | 3.2 | Desarrollo Pruebas de aplicación, Conclusión actualizada | Ivan Callasaya  Jorge Gutierrez  Fabián Flores |
| 12/12/22 | 4.0 | Informe Final | Ivan Callasaya  Jorge Gutierrez  Fabián Flores |

**Índice**

[**1. Panorama General**](#_heading=h.gjdgxs) **5**

[1.1 Objetivos](#_heading=h.30j0zll) 5

[1.2 Resumen del proyecto](#_heading=h.1fob9te) 5

[2. Referencias](#_heading=h.3znysh7) **7**

[3. Organización del Proyecto](#_heading=h.2et92p0) **8**

[3.1. Personal y entidades internas](#_heading=h.tyjcwt) 8

[3.2. Roles y responsabilidades](#_heading=h.3dy6vkm) 8

[3.3. Mecanismos de Comunicación](#_heading=h.1t3h5sf) 8

[4. Planificación de los procesos de gestión](#_heading=h.4d34og8) **9**

[4.1. Planificación inicial del proyecto](#_heading=h.2s8eyo1) 9

[4.2. Lista de actividades (carta Gantt)](#_heading=h.m9ha2xe2g63d) 12

[4.3. Planificación de la gestión de riesgos](#_heading=h.18fn8ouoltzn) 14

[5. Planificación de los procesos técnicos](#_heading=h.gr0mh11aumbs) **15**

[5.1. Herramientas y técnicas](#_heading=h.26in1rg) 15

[**6. Especificación de Proyecto**](#_heading=h.bjbk6irn4m49) **15**

[7. Análisis](#_heading=h.lnxbz9) **16**

[7.1. Modelo de proceso](#_heading=h.35nkun2) 16

[7.1.1. Diagrama de casos de uso](#_heading=h.1ksv4uv) 16

[7.1.2. Descripción de casos de uso de sistema](#_heading=h.44sinio) 17

[7.1.3. Diagrama de secuencia.](#_heading=h.hcosf11p6nfy) 20

[7.2. Descripción de Arquitectura](#_heading=h.7tclyjwboyel) 22

[**8. Requerimientos**](#_heading=h.fni5amxqk0jn) **23**

[**8.1. Funcionales**](#_heading=h.26hax5nm82lz) **23**

[**8.2. No funcionales**](#_heading=h.m2jmsh5vjjj8) **23**

[**9. Interfaz Gráfica**](#_heading=h.5ciw2ggw62fb) **24**

[9.1. Cliente](#_heading=h.sy3ypa71c8tp) 24

[9.2. Administrador](#_heading=h.625gljqphr3m) 25

[**10. Implementación**](#_heading=h.tjqa4c95du3u) **27**

[**11. Pruebas de la aplicación**](#_heading=h.np376g1uzsl) **29**

[**12. Conclusión**](#_heading=h.v8g2xt8im1vh) **32**

# Panorama General

## 1.1 Objetivos

* **Objetivo General:**
* Desarrollar un software que logre traducir el lenguaje de señas, permitiendo que el usuario pueda comunicarse con mayor facilidad, mediante un guante con sensores flex.
* **Objetivo Específico:**
* Investigar y recopilar información necesaria o llamativa para la elaboración del proyecto.
* Desarrollo de la documentación mediante informes que muestren en detalle los avances realizados a lo largo del proyecto (informe, bitácora, presentaciones, etc)
* Estudiar aspectos fundamentales con respecto al lenguaje de programación y las herramientas que se utilizarán para el desarrollo del software.
* Diseño de la interfaz adecuada para el usuario de la aplicación.
* Desarrollar el código del aplicativo en Arduino.
* Realizar pruebas o testeos del funcionamiento de la aplicación, y efectuar modificaciones si es necesario.
* Elaborar un manual de uso para instruir al usuario en el manejo correcto de cada mecanismo del aplicativo.
* Presentar el proyecto finalizado.

## 1.2 Resumen del proyecto

* **Propósito, alcance, objetivos**

**Propósito:** El proyecto permitirá la mayor facilidad entre la comunicación de una persona sordo muda con aquellas personas que desconocen el lenguaje de señas de forma que les resulte didáctico y entretenido aprender este lenguaje dándole un enfoque más educativo.

**Alcance:** El software contará con módulos para el ingreso, modificación y eliminación de datos de clientes. Así mismo, será desarrollado usando orientación a objetos y en el lenguaje C. También se usará el modelo clásico.

**Objetivos:** Mejorar la comunicación de las personas sordomudas facilitándoles el entendimiento de su lenguaje de señas para las demás personas.

* **Escenario del problema**

Las personas que presentan discapacidad de sordomudo se enfrentan diariamente a dificultades para comunicarse con el resto de personas, ya que, la gran mayoría de la población no suele manejarse con distintos tipos de lenguajes, más aún si se trata de lenguaje de señas. Entre las dudas que surgen a esta disyuntiva:

¿Cómo puede comunicarse o hacerse entender un sordomudo en situaciones cotidianas si su entorno no le entiende, o peor aún, en situaciones de riesgo no puedan entenderle?

Por lo general las soluciones que reciben las personas con este tipo de discapacidad son muy inexactas y confusas, tales como:

El simple hecho de dar instrucciones básicas es dificultoso para las personas que no entienden el lenguaje de señas.

Las personas suelen hacer mímicas sin sentido que confunden más que ayudan a las personas con este tipo de discapacidad.

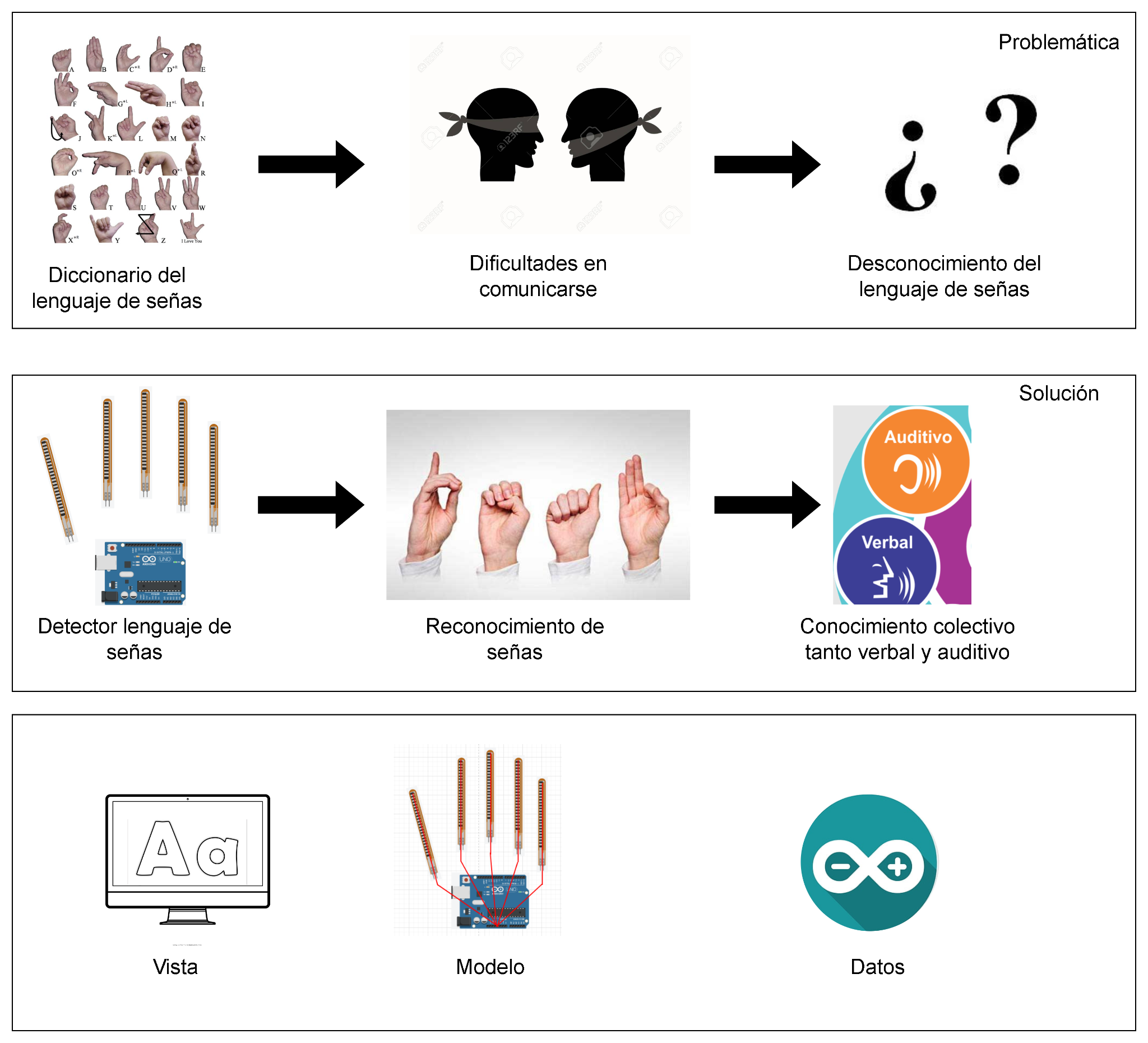
* **Esquema solución**

Figura 1: Esquema problema y solución

**Solución:**

Lamentablemente el lenguaje de señas no es considerado o incluido en la enseñanza básica, por lo cual, surge la necesidad de elaborar un traductor de lenguaje de señas a través de texto por medio de un guante compuesto con sensores flex para tener más precisión a la hora de hacer las señas.

* **Suposiciones y restricciones**

Suposiciones:

* Se pueda mantener activo de forma autónoma por baterías

Restricciones:

* Dependencia de electricidad constante
* Falta de capital
* Contemplación en un lapso de tiempo de entrenamiento para que el usuario domine el aparato
* **Entregables del Proyecto**
  + Bitacoras
  + Informes
  + Wiki del Proyecto.

# Referencias

* Documentación del sensor flex: <https://www.sparkfun.com/datasheets/Sensors/Flex/flex22.pdf>
* Manual de usuario de arduino uno:

<https://www.arduino.cc/reference/es/>

* Costo de luz: <https://www.enel.cl/es/clientes/tarifas-y-regulacion/consumo-artefactos-electricos.html>

# **Organización del Proyecto**

## 3.1. Personal y entidades internas

* Líder,
* Programador
* Diseño
* Documentador

## 3.2. Roles y responsabilidades

| Roles | Responsables |
| --- | --- |
| Líder | Ivan Callasaya |
| Programador | Ivan Callasaya  Jorge Gutierrez  Fabian Flores |
| Diseño | Jorge Gutierrez |
| Documentador | Fabian Flores |

## 3.3. Mecanismos de Comunicación

Para realizar los informes y bitácoras se utilizará Google Docs, que permite a los miembros del grupo trabajar simultáneamente en los documentos, además se usará Redmine como herramienta web para la gestión del proyecto.

**Cuentas en redes sociales:**   
El grupo se comunicará principalmente por:

* WhatsApp del proyecto.
* Discord.

# **Planificación de los procesos de gestión**

## 4.1. Planificación inicial del proyecto

* **Planificación de estimaciones**

Se describirán los diferentes recursos tanto por hardware-software requeridos para el desarrollo del proyecto:

| **Recurso** | **Producto** |
| --- | --- |
| **Hardware** | **Computadores**  **Arduino**  **Sensores flex**  **cable puente (macho-hembra)** |
| **Sofware** | **Github**  **Arduino IDE**  **Google Drive**  **Discord**  **Tinkercad** |

| **Elemento** | **Cantidad** | **Costo** |
| --- | --- | --- |
| Computadores | 3 unidades | $800.000 |
| Sensores flex | 5 unidades | $8.000 |
| Arduino uno | 1 unidad | $17.000 |
| Cable puente | 8 unidad | $2.000 |
| Resistencias | 6 unidades | $600 |
| Costo CGE | 3 boletas | $4.600 |
| Github | 3 cuentas | $0 (software libre) |
| Google Drive | 3 cuentas | $0 (software libre) |
| Arduino IDE | 3 cuentas | $0 (software libre) |
| Discord | 3 cuentas | $0 (software libre) |
| Tinkercad | 3 cuentas | $0 (software libre) |
| **Total** | | **$2.490.400** |

* **Planificación de Recursos Humanos**

Programador: 2, Documentador: 1, Diseñador:1, Jefe de Proyecto: 1.

La siguiente tabla muestra el precio para cada rol y también se muestra el número de responsables en cada rol.

| **Rol** | **Número de personas** | **Precio hora** | **Horas por semana** |
| --- | --- | --- | --- |
| **Programador** | 3 | $10.000 | 25 hrs |
| **Diseño** | 1 | $6.000 | 10 hrs |
| **Documentador** | 1 | $6.000 | 12 hrs |
| **Jefe de Proyecto** | 1 | $20.000 | 15 hrs |
| **Total:** | | $62.000 | 62 hrs |

A continuación, se hará la estimación de costo semanal.

| **Rol** | **Número de personas** | **Horas totales por semana** | **Costo Total por semana** |
| --- | --- | --- | --- |
| **Programador** | 3 | 25 hrs | $750.000 |
| **Diseño** | 1 | 10 hrs | $60.000 |
| **Documentador** | 1 | 12 hrs | $72.000 |
| **Jefe de Proyecto** | 1 | 15 hrs | $300.000 |
| **Total, por equipo de trabajo:** | | 62 hrs | $1.182.000 |

Luego tenemos una estimación mensual considerando los meses de desarrollo del proyecto (4 meses)

| **Total por mes** | $1.182.000 \* (4 semanas) |
| --- | --- |
| **Total** | **$4.728.000 \* (4 meses) = $18.912.000** |

Finalmente, se calculará el precio total del proyecto considerando los recursos de hardware-software más la estimación final de los recursos humanos.

| **Recursos** | **Costos** |
| --- | --- |
| **Hardware y Software** | $2.490.400 |
| **Recursos Humanos** | $18.912.000 |
| **Total Proyecto** | **$21.852.400** |

## 4.2. Lista de actividades (carta Gantt)

* **Actividades de trabajo**

1. Organización de trabajo

2. Planificación proyecto

3. Cotización de materiales

4. Bocetos del proyecto

5. Aprendizaje de las herramientas que se utilizaran

6. Montaje del proyecto

7. Testear y calibrar sensores

8. Corrección de errores con el código

* **Asignación de tiempo**

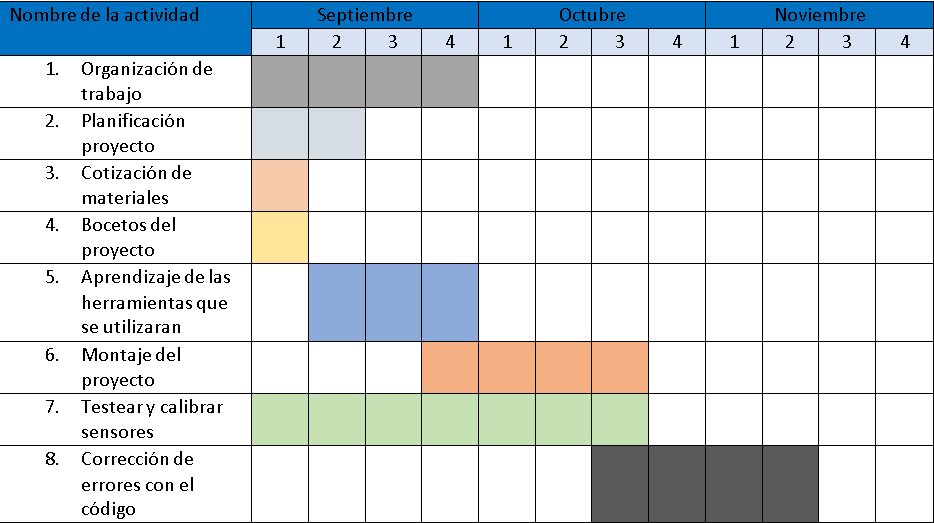
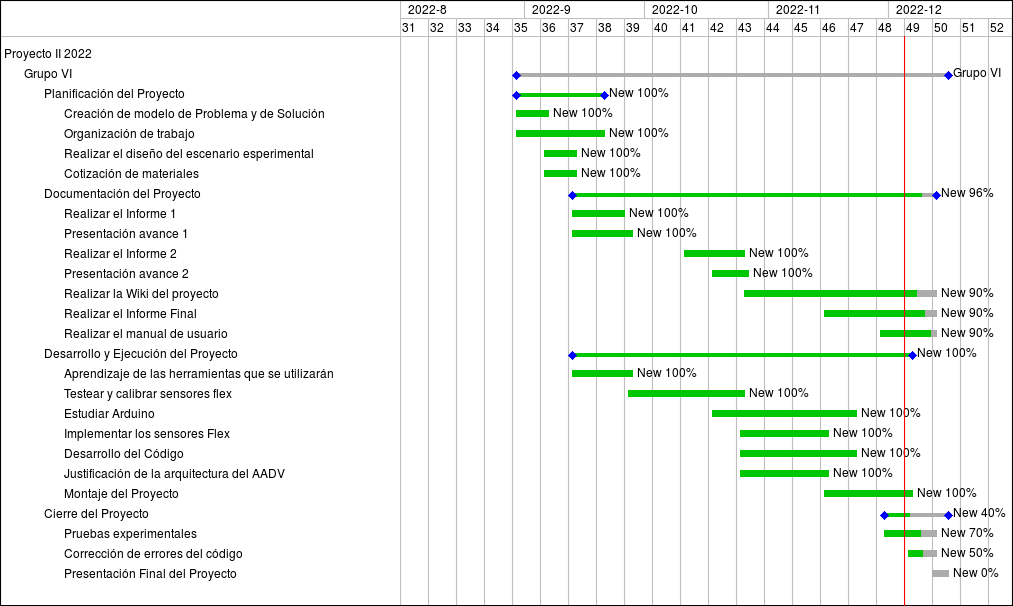


Ilustración 1: Carta Gantt original del proyecto.

Ilustración 2: Carta Gantt actualizada del proyecto.

## 4.3. Planificación de la gestión de riesgos

Durante el desarrollo del proyecto pueden presentarse una serie de riesgos o problemas que pueden interferir con el fluido avance del mismo, a continuación, se describirán los posibles riesgos que se pueden presentar.

| **Niveles** | **Categoría de Riesgos** |
| --- | --- |
| **1** | Catastrófico |
| **2** | Crítico |
| **3** | Marginal |
| **4** | Despreciable |

| **RIESGOS** | **PROBABILIDAD DE OCURRENCIA** | **NIVEL DE IMPACTO** | **ACCIÓN REMEDIAL** |
| --- | --- | --- | --- |
| **Defectos con las herramientas** | 30% | 3 | Tener una cotización de respaldo por posibles |
| **Falta de los conocimientos para el desarrollo del proyecto** | 30% | 4 | Repasar o buscar información relevante para cumplir con la tarea que presente el conflicto. |
| **Enfermedad de algún compañero de trabajo** | 25% | 1 | Trabajo remoto para evitar retrasos. |
| **Daño hacia el equipo donde se respalda la información** | 45% | 2 | Tener respaldada la información en la nube siempre que se realice algún avance importante. |
| **Error en la ejecución del código** | 45% | 3 | Revisión total del código para identificar los errores que causan el conflicto o los posibles errores a futuro. |
| **La ausencia o salida de algún de los integrantes del equipo** | 33% | 1 | Se tendrán que reasignar las actividades programadas entre los integrantes del equipo. |

# **Planificación de los procesos técnicos**

## 5.1. Herramientas y técnicas

**Herramientas**:

Redmine,

* Microsoft Office,
* Arduino IDE

**Técnicas:**

* Calibración de los sensores
* Organización de cada gesto
* Metodología scrum

# **Especificación de Proyecto**

El proyecto consiste en el desarrollo de un detector de señas, en el cual el usuario por medio de un guante compuesto por sensores flex permitirá el reconocimiento de las señas que realizará la persona con el fin de poder entregar el mejor entendimiento hacia las personas sordomudas, en los ambientes donde se utilizará el software será tanto por Windows y el entorno Arduino.

Todos los tipos de seguimientos del avance del proyecto serán por medio de Redmine.

Los siguientes puntos se explicará de mejor manera los puntos principales sobre el alcance de proyecto:

* El software a desarrollar (será por el entorno Arduino) llamado G.T.S. (Guante Traductor de Señas)
* Todo usuario que quiera utilizar el software tendrá que utilizar el guante compuesto por sensores flex en cada dedo con el fin de tener mayor precisión (ya que manejar el hardware se tendrá mejor detección de cada dedo) cada dedo tendrá sus respectivas longitudes con el fin de poder designar cada seña sus longitudes por dedo.
* Cada vez que por accidente se realice una mala compostura de dedos muy difícilmente será reconocido las malas señas con el fin de agilizar el aprendizaje de las personas al utilizar este software.

Las metas primordiales del proyecto es facilitar la comunicación y el aprendizaje para todo usuario que quiera adquirir con mayor facilidad el lenguaje de señas con el fin de mejorar la comunicación de las personas

# **Análisis**

A continuación, veremos de forma detallada el análisis del proyecto, mediante diagramas de casos de uso dependiendo de la interacción que se desea obtener, junto con su descripción, la interfaz gráfica que se implementará y las especificaciones de requerimientos funcionales y no funcionales.

## 7.1. Modelo de proceso

### 7.1.1. Diagrama de casos de uso

En la siguiente figura se puede observar los casos de uso del proyecto.

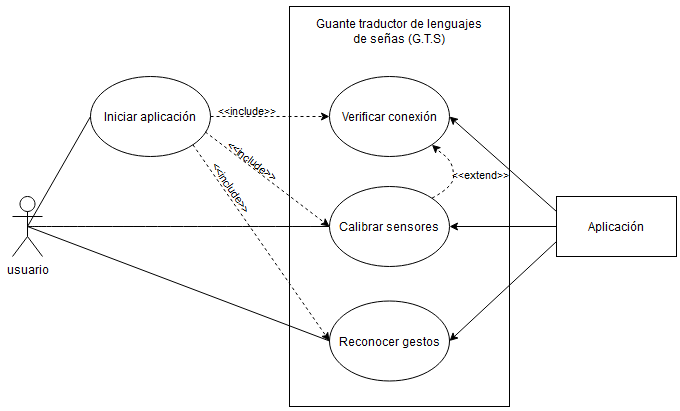


Figura 2: Diagrama de casos de uso.

### 7.1.2. Descripción de casos de uso de sistema

Caso de uso de sistema **“Iniciar aplicación”**.

| **Nombre CUS: Iniciar aplicación** | |
| --- | --- |
| **Descripción:** Permite al usuario iniciar la aplicación | |
| **Actor:** Usuario | |
| **Precondición:** La aplicación debe estar instalada en el ordenador del usuario, además debe tener a disposición el dispositivo físico (guante con los sensores flex). | |
| **Flujo Principal:** Usuario  1. El usuario pulsa el icono de la aplicación en su ordenador. | **Flujo Principal:** Sistema  2. La aplicación se inicializa.  3.  <<Include>> el caso de uso **“Verificar conexión”** |
| **Flujo Alternativo:** | 3.1 <<Include>> el caso de uso **“Calibrar sensores”.** |
| **Flujo Alternativo:** | 3.2 <<Include>> el caso de uso **“Reconocer gestos**”. |
| **Postcondiciones:** La aplicación se encuentra inicializada y lista para realizar sus funcionalidades. | |

Caso de uso del sistema **“Verificar conexión”**.

| **Nombre CUS: Verificar conexión** | |
| --- | --- |
| **Descripción:** Permite al aplicativo que todas las conexiones de los dispositivos de E/S funcionen correctamente. | |
| **Actor:** Aplicación | |
| **Precondición:** La aplicación debe estar inicializada. | |
| **Flujo Principal:** Aplicación | **Flujo Principal:** Sistema  1. El sistema verifica la existencia de los sensores guardados previamente.  2. El sistema verifica el estado de cada uno de los dispositivos que funcionen correctamente.  3.  <<extend>> al caso de uso **“Calibrar sensores”.** |
| **Flujo Alternativo:** | 1.1. El sistema no reconoce los sensores registrados previamente. |
| **Flujo Alternativo:** | 2.1. El sistema indica que el dispositivo no funciona correctamente. |
| **Postcondiciones:** Se validan los dispositivos de entrada para poder hacer uso del aplicativo. | |

Caso de uso del sistema **“Calibrar sensores”**.

| **Nombre CUS:** **Calibrar sensores** | |
| --- | --- |
| **Descripción:** El sistema calibra los sensores del flex para que se pueda detectar sin problemas cada gesto realizado por el usuario. | |
| **Actor:** Aplicación y Usuario | |
| **Precondición:** La aplicación debe estar inicializada, además de que las conexiones de los dispositivos de E/S fusionen correctamente. | |
| **Flujo Principal:** Aplicación, Usuario  3. El usuario realiza los movimientos solicitados. | **Flujo Principal**: Sistema  1. El sistema identifica cada uno de los sensores flex.  2. Solicita al usuario que realice movimientos para identificar que los sensores funcionan correctamente.  4. El sistema registra las mediciones obtenidas.  5. El sistema verifica que las mediciones de la calibración son óptimas para el funcionamiento. |
| **Flujo Alternativo:** | 5.1. Las mediciones obtenidas no son óptimas para el funcionamiento. |
| **Postcondiciones:** El nuevo proveedor es registrado en el sistema para generar informes de compra de insumos. | |

Caso de uso del sistema **“Reconocer gestos”**.

| **Nombre CUS: Reconocer gestos** | |
| --- | --- |
| **Descripción:** Reconocimiento de gestos por el software | |
| **Actor:** Aplicación, Usuario | |
| **Precondición:** La aplicación debe estar inicializada, además de que las conexiones de los dispositivos de E/S fusionen correctamente. Tener montado y programado el guante sensor flex. | |
| **Flujo Principal:** Usuario    3. El usuario realiza gestos del lenguaje de señas. | **Flujo Principal:** Sistema  1. El sistema tiene almacenado todo tipo de gestos pre programados.  2. Solicita al usuario que realice gestos para identificarlos.  4. La aplicación empieza a capturar o reconocer los gestos realizados por el usuario.  5. La aplicación indica mediante texto los gestos obtenidos. |
| **Flujo Alternativo:** |  |
| **Postcondiciones:** Ninguna | |

### 7.1.3. Diagrama de secuencia.

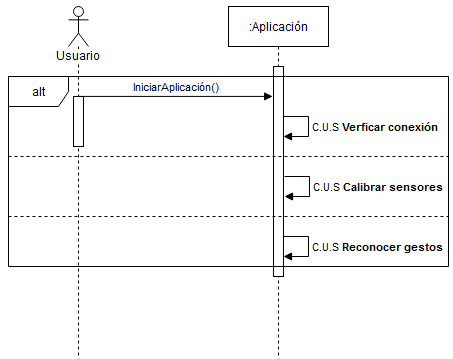
Diagrama de Secuencia **“Iniciar aplicación”**.

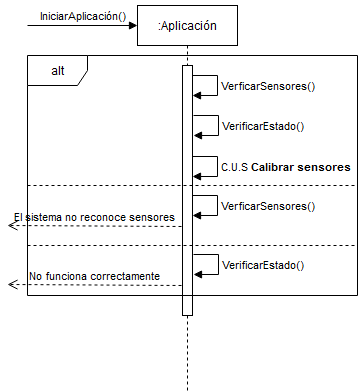
Diagrama de Secuencia **“Verificar conexión”**.

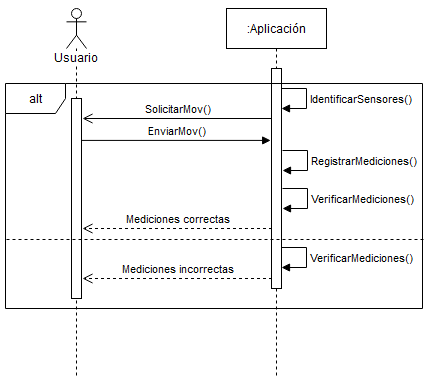
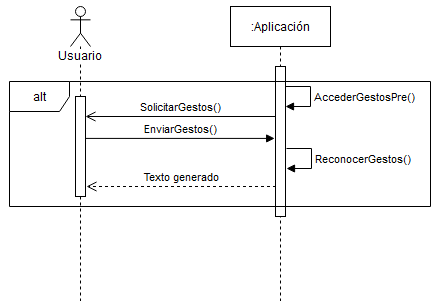
Diagrama de Secuencia **“Calibrar sensores”**.

Diagrama de Secuencia **“Reconocer gestos”**.

## 7.2. Descripción de Arquitectura

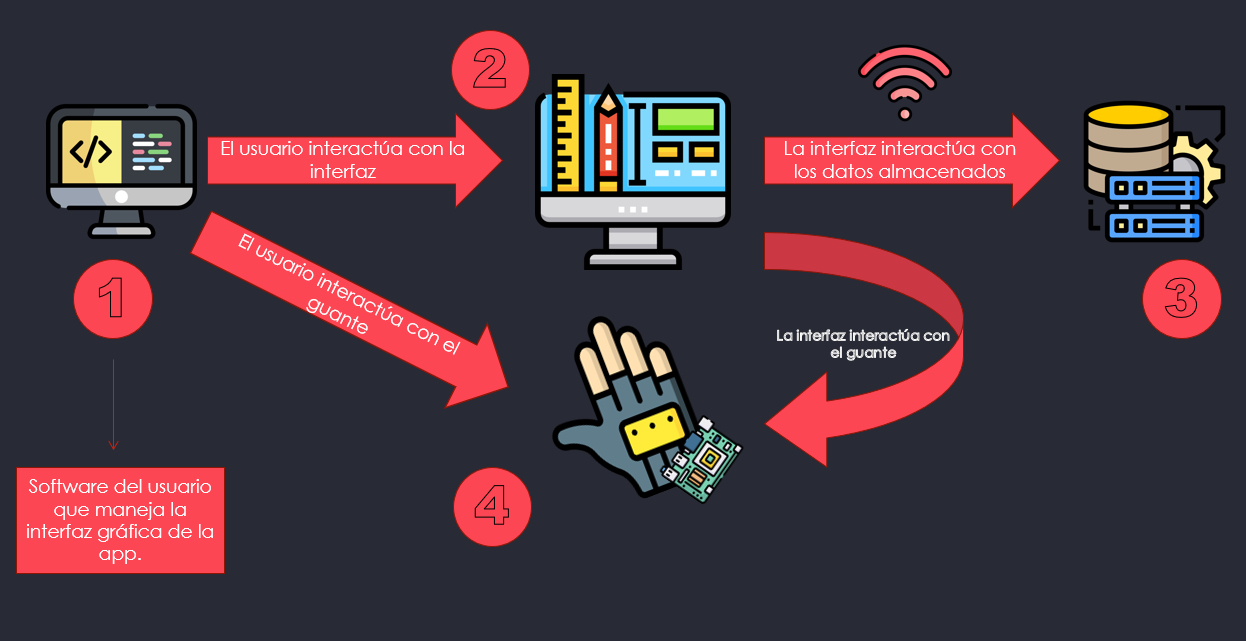


Figura 3: Arquitectura Propuesta

1. Dispositivos que utilizan el software con el que el usuario puede interactuar con el aplicativo.
2. Interfaz gráfica del aplicativo con la que el usuario debe interactuar, esta mostrará en pantalla el texto generado por los gestos realizados por el usuario mediante el guante con los sensores flex.
3. El registro de los datos que será necesario para el almacenamiento de los gestos previamente programados en el software.
4. El usuario interactúa con la aplicación mediante los gestos realizados por los sensores flex, es así como se puede identificar las mediciones registradas para posteriormente identificarlas y ser emitidas como texto en la interfaz del usuario.

# **Requerimientos**

## 8.1. Funcionales

| **Referencia** | **Descripción** |
| --- | --- |
| RF01 | Permitirá el reconocimiento del cliente mediante los sensores en cada dedo |
| RF02 | Se establecerá las pre programación de cada gesto |
| RF03 | Se puede modificar y establecer tiempo de espera en cada gesto |
| RF04 | El sistema visualizará cada gesto realizado por el cliente |
| RF05 | Adaptación de los sensores de las longitudes de dedos más comunes del cliente |
| RF06 | Permitirá agregar cualquier tipo de gesto nuevo que requiera el cliente |

## 8.2. No funcionales

| **Referencia** | **Descripción** |
| --- | --- |
| RNF01 | Contará con 2 modos de uso:   * Modo creador: Permitirá agregar todo tipo de gestos nuevos dentro del lenguaje de señas que requiera el cliente * Modo usuario: Solo permitirá el reconocimiento de gestos dentro del software |
| RNF02 | El sistema reproducirá audio por cada gesto realizado |

# **Interfaz Gráfica**

## 9.1. Cliente

Se despliega para aquellas personas que quieren usar el software

Figura 4: Menu de inicio cliente

Una vez seleccionada la única opción de la Figura 4. se despliega otra ventana reconociendo los gestos y lo que significa cada seña como se ve en la Figura 5.

Figura 5: Reconocimiento por señas

## 9.2. Administrador

Este es el menú cuando un desarrollador está trabajando dentro del software

Figura 6: Menu de inicio cliente

Dependiendo de cual opción elijas dentro de la Figura 6. Se desplegará una nueva página donde se podrá editar tanto el tipo de gesto junto con su significado.

Figura 7: Menu de inicio cliente

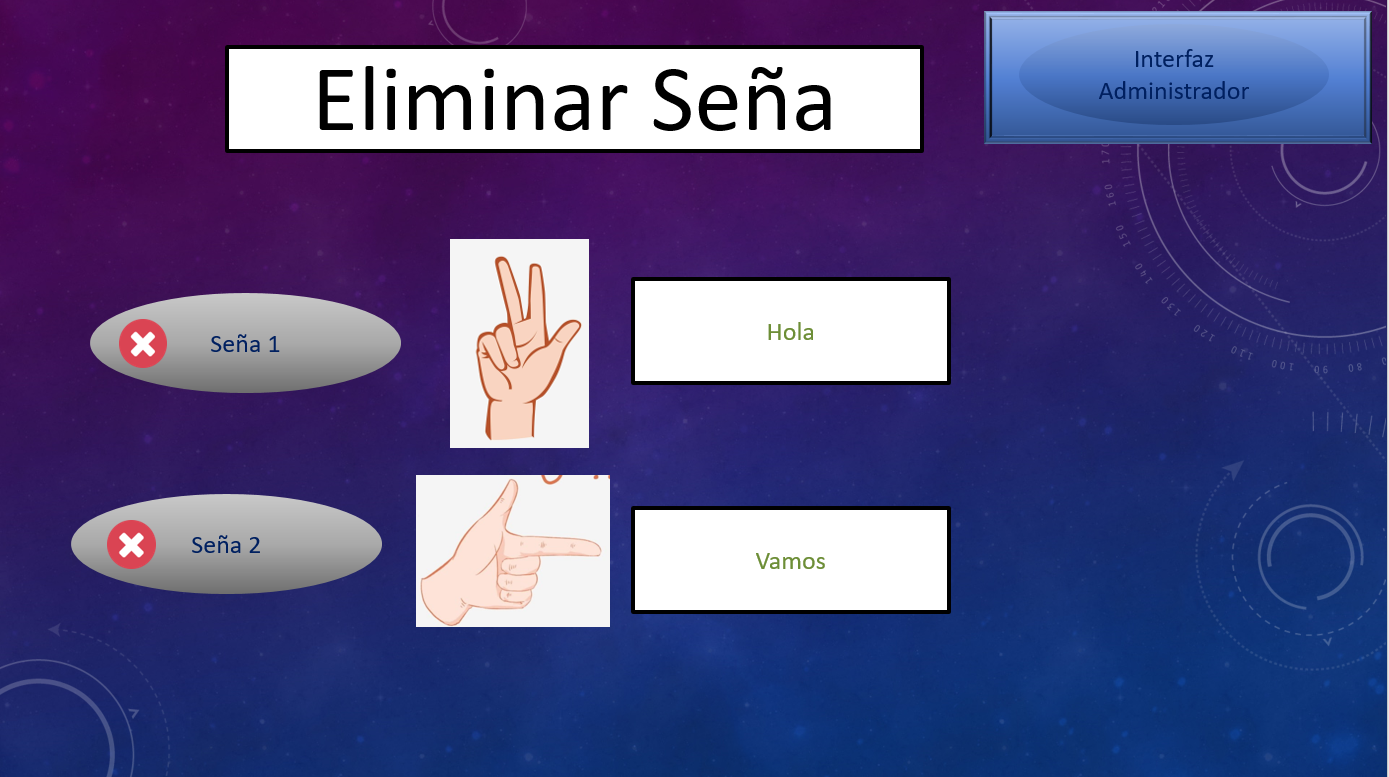
En caso contrario de que se haya asignado una seña y después querer editar su significado o eliminar alguna en particular como se ve en la Figura 8.

Figura 8: Menu de inicio cliente

# Implementación

**10.1. Modelos de Implementación**

* **Relación hardware con software:**

Reconocimiento de los sensores flex, la forma para hacer la intersección entre el sensor con el arduino será a través de la parte analógica del mismo, para que este lo detecte como una única entrada analógica, así cada sensor flex depende de él de forma individual, y con ello cada sensor tendrá su misma coordenada.

* **Lectura de pantalla Lcd:**

Para poder hacer uso de las funcionalidad de la biblioteca digital LiquidCrytal, primero debemos exportar la biblioteca en sí añadiendo en el archivo el objeto de la clase LiquidCrystal\_I2C, con dirección, columnas y filas indicadas.



Ilustración 3: Implementación de la librería LiquidCrystal.

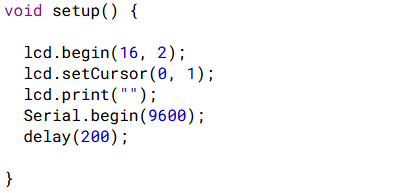
Luego creamos una función void setup() para inicializar la comunicación serial. Es primordial, ya que, en nuestro programa tenemos que seleccionar el bloque en el que se almacenarán estos comandos. Tomando en cuenta que inicializamos la interfaz de la pantalla LCD, especificando las dimensiones (ancho y alto) de la pantalla con el comando lcd.begin (). Luego le damos una nueva posición a la posición (fila, columna) del cursor LCD, es decir, establecemos la ubicación en la que se mostrará el texto subsiguiente escrito en la pantalla LCD.

Ilustración 4: Módulo inicial creado en Arduino.

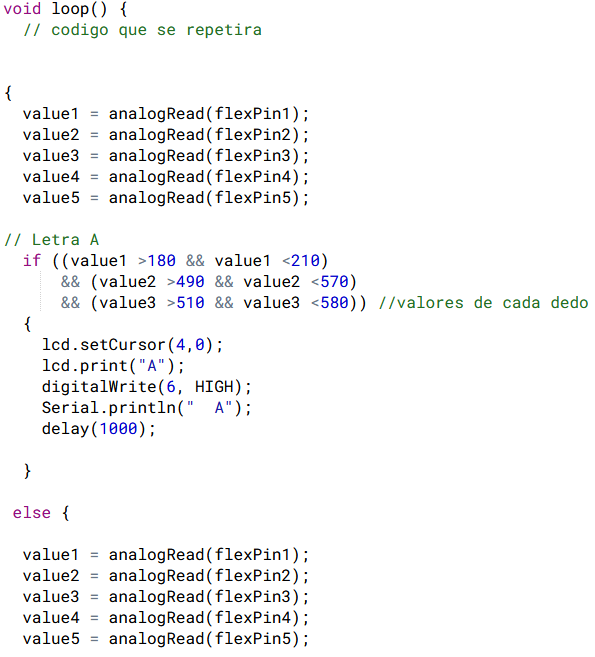
Continuamos, llamando a una función void loop( ), y declarando variables a cada lector analógico del Arduino (por cada sensor flex), luego ejecuta la condición que llama a cumplir los valores asignados para formar cada letra, en la siguiente ilustración se puede apreciar el código para la letra “A”.

Ilustración 5: Código en ejecución.

# Pruebas de la aplicación

En un principio, se fue probando el proyecto por partes, es decir, una parte enfocada en la maquetación y confección del guante, y otra directamente en el funcionamiento del código.

* **Pruebas de calibración de sensores flex:** En las primeras pruebas se implementó un pequeño código para verificar que las conexiones de los sensores flex con las entradas analógicas de Arduino estuvieran correctamente conectadas. En cuanto a la maquetación del guante como primer intento se unieron los sensores flex con cinta adhesiva al guante, provocando una rigidez a la hora de hacer las señas, y afectando las mediciones en la calibración. 

Ilustración 6: Calibración de sensores flex.

Debido a los problemas surgidos por la rigidez del guante se decidió, coser comparaciones en el guante para guardar los sensores flex, y evitar la rigidez presentada en nuestras primeras pruebas.

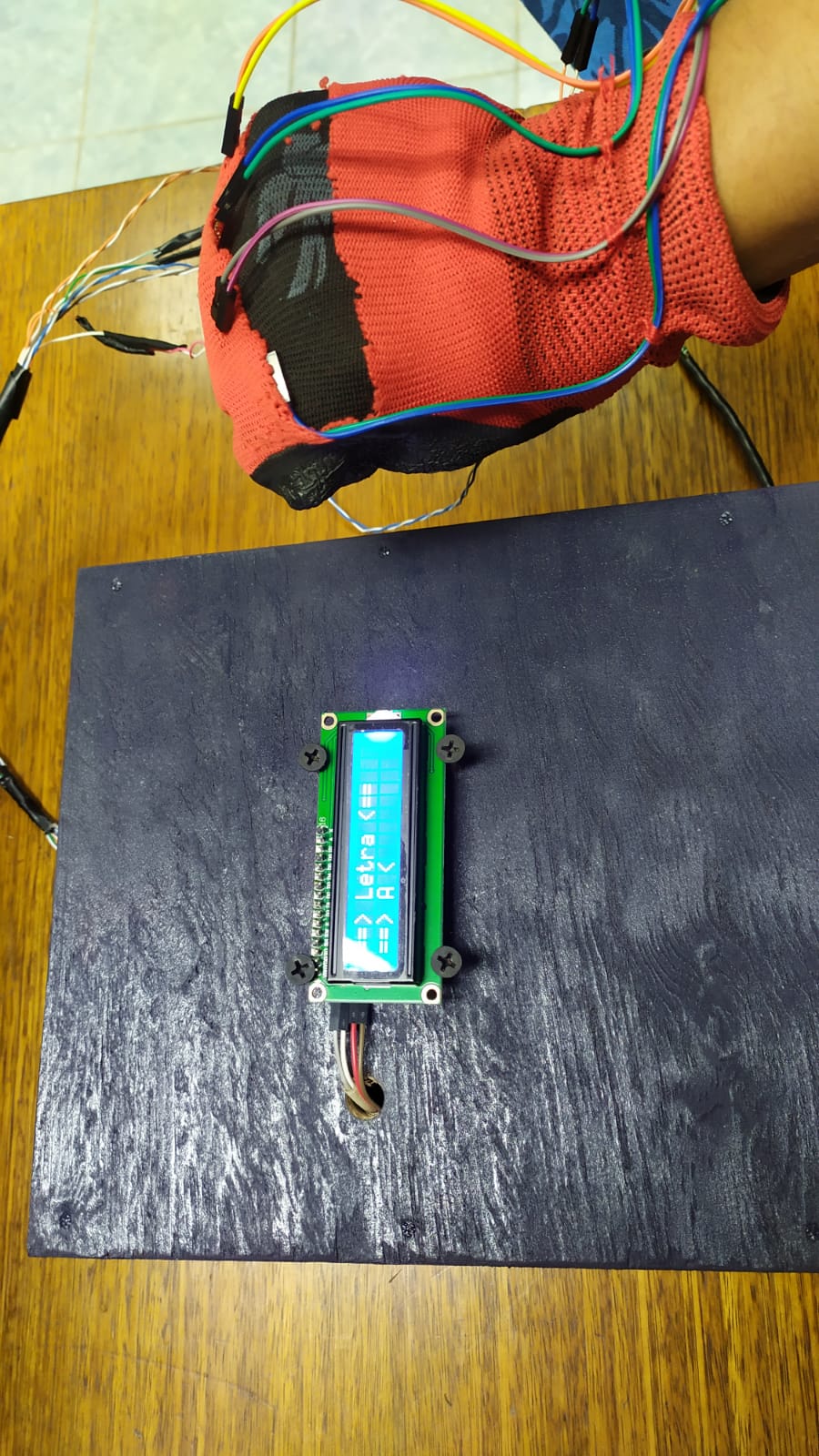
Para las últimas pruebas presenta mayor fluidez en cuanto al movimiento de los dedos, los sensores responden correctamente a los gestos o señas, en cuanto a la maquetación se pulió el tema de cableado de los sensores, dándoles mayor extensión en el largo del cable.

Ilustración 7: Calibración de sensores flex.

* **Pruebas ejecución de código:** En las primeras pruebas surgieron problemas a la hora de ejecutar el código debido a la rigidez que presentaba el guante, si bien, era posible ejecutarse correctamente resultaba incómodo para el usuario gesticular con el guante.

Una vez que le quitamos la rigidez del guante los sensores pudieron mandar sus coordenadas de forma correcta para que el código los detectara, así mismo una vez se hace la lectura de las coordenadas el led imprime la letra detectada en la pantalla cómo se puede apreciar en la *ilustración 8.*

.

Ilustración 8: Ejecución código.

En la *Ilustración 8* se puede ver una de las pruebas realizadas para la lectura de las coordenadas de los sensores flex, e imprimiendo el resultado correspondiente en formato texto mediante el dispositivo LCD en tiempo real.

# **Conclusión**

Para concluir este informe, se debe mencionar la importancia de optimizar la productividad de las actividades y procesos que transcurren durante el proyecto, ya que contamos con muchos inconvenientes al momento de realizar el cableado y maquetación del guante, es necesario priorizar los puntos a esenciales en el desarrollo del proyecto, de modo que la investigación se centre en los aspectos relevantes a conocer en el proyecto y sea fructífera para el conocimiento del equipo del proyecto.

En el transcurso del proyecto, se documentó el avance realizado del mismo en diversas formas, ya sean bitácoras que describen lo realizado en las distintas reuniones de equipo de trabajo, con una Carta Gantt que sirva de ayuda organizacional de las tareas a realizar, o por medio del presente informe que detalle todo lo realizado durante todo un periodo de trabajo que abarcó un aproximado de cuatro meses.

Entres otros aspectos, cabe destacar que, debido al objetivo del proyecto, resultó interesante darle un enfoque distinto al desarrollo de la aplicación, pasando a un proyecto mucho más entretenido e interactivo, pues se decidió no considerar una interfaz de usuario como tal, ya que, el usuario ahora puede interactuar de forma más directa con el guante y el aplicativo como tal.

Llegado a esta etapa de finalización , se debe destacar que a la hora de planificar el proyecto y realizar el apartado de análisis y reevaluar el diseño de la aplicación se pudo realizar con algunos inconvenientes presentados sobre todo a la hora de la maquetación del aplicativo, debido a que no llevamos un orden con los cableados, causando que se enredaran muchas veces unos con otros, o que el forcejeo de los cables provocara daño en los pines de los sensores flex, sin embargo se logró sobreponerse ante las dificultades presentadas, llegando a tener un proyecto consistente.