

# Carpeta Técnica

Proyecto SignaLink 2025

**Escuela de Educación Secundaria Técnica  
Nº7 "Taller Regional Quilmes" IMPA**

Prácticas Profesionalizantes: Especialidad Aviónica





# Índice

## Preámbulo

- 1.1. ¿Quiénes somos?
- 1.2. Datos de contacto
- 1.3. Docentes responsables a cargo
- 1.4. Información adicional del proyecto
  - 1.4.1. Tiempo total invertido en el desarrollo
  - 1.4.2. Herramientas y programas utilizados
- 1.5. Agradecimientos y reconocimientos

## Introducción

- 2.1. Descripción general del proyecto
- 2.2. Origen y motivación del proyecto
- 2.3. Problemática social a resolver
- 2.4. Solución tecnológica propuesta
- 2.5. Metas y expectativas de logro

## Especificaciones Técnicas

- 3.1. Principio de funcionamiento del sistema
  - 3.1.1 Transporte y sincronización de datos
- 3.2. Diagrama de bloques funcional
- 3.3. Sistema de alimentación y consumo energético

## Módulos

- 4.1. Módulo de la muñeca
  - 4.1.1. Componentes electrónicos empleados
  - 4.1.2. Diseño de la placa de circuito impreso (PCB)
  - 4.1.3. Modelado y diseño 3D del módulo
- 4.2. Módulo del pecho
  - 4.2.1. Componentes electrónicos empleados
  - 4.2.2. Diseño de la placa de circuito impreso (PCB)
  - 4.2.3. Modelado y diseño 3D del módulo



## Tecnología

- 5.1. Lenguajes de programación implementados
  - 5.1.1. Lenguaje de programación en C
  - 5.1.2. Lenguaje de programación en Python
  - 5.1.3. Desarrollo web
- 5.2. Uso del framework ESP-IDF
- 5.3. Configuración del entorno virtual de desarrollo
  - 5.3.1. Proceso de creación
  - 5.3.2. Activación del entorno virtual
  - 5.3.3. Instalación de dependencias
  - 5.3.4. Ventajas y beneficios
- 5.4. Aplicaciones de inteligencia artificial en el proyecto
  - 5.4.1. Proceso de Entrenamiento y Recopilación de Datos
  - 5.4.2. Adaptación del modelo
- 5.5. Comunicación mediante Bluetooth Low Energy (BLE)
- 5.6. Comunicación mediante MQTT(Message Queuing Telemetry Transport)
- 5.7. Inteligencia Artificial

## Página Web

- 6.1. Introducción al sitio web
- 6.2. Arquitectura y diseño de la página
- 6.3. Secciones principales y funcionalidades
- 6.4. Tecnología utilizada y metodología de desarrollo
- 6.5. Sponsors, impacto y proyección

## Imágenes

- 7.1. Componentes utilizados
- 7.2. Prototipo
- 7.3. Placas

## Mejoras a futuro

- 8.1 Raspberry Pi Compute Module
- 8.2. Placa PCB de Módulo
- 8.3. Aplicación



## Preámbulo

### 1.1. ¿Quiénes somos?

Somos un grupo de estudiantes de séptimo año de la especialidad de Aviónica en la Escuela de Educación Secundaria Técnica N.º 7 “Taller Regional Quilmes” (IMPA). Nos caracteriza la responsabilidad, el compromiso y el trabajo en equipo, valores que nos han permitido afrontar distintos desafíos académicos con éxito. Creemos firmemente que la colaboración, la constancia y la creatividad son claves para alcanzar grandes resultados. Cada integrante del equipo aporta sus conocimientos, su esfuerzo y su mirada personal, lo que en conjunto nos permite formar un grupo equilibrado y con capacidad de innovación.

#### Integrantes:



**ALBORNOZ THIAGO AGUSTIN**

**Desarrollo de Software**  
**Gestión de Redes Sociales y Marketing**

**DNI: 48.058.741**



**ERBINO SEBASTIAN JOSÉ**

**Diseño del Prototipo**  
**Integración de Hardware**

**DNI: 47.643.304**



**FRANCO VALENTIN**

**Armado de Hardware**  
**Contacto de Sponsors**

**DNI: 47.643.304**



### **LESME GARCIA FRANCO AGUSTIN**

**Armado de Software**

**Armado de Hardware**

**DNI: 47.561.583**



### **POGGI JANIN LORENZO**

**Registro de Seguimiento**

**Programación del Sistema**

**DNI: 47.862.040**



### **SARNIGUETTE VALENTINO**

**Prueba Electrónica**

**Armado de Circuitos**

**DNI: 47.230.092**

#### **Correos Electrónicos:**

[albornozthiagoagustin@gmail.com](mailto:albornozthiagoagustin@gmail.com)  
[serbino25@gmail.com](mailto:serbino25@gmail.com)  
[valentinfranco2506@gmail.com](mailto:valentinfranco2506@gmail.com)  
[franco.lesme206@gmail.com](mailto:franco.lesme206@gmail.com)  
[lorenzo.poggijanin@gmail.com](mailto:lorenzo.poggijanin@gmail.com)  
[valentinosarniguette@gmail.com](mailto:valentinosarniguette@gmail.com)



**Figura 1**

Equipo que conforma el proyecto SignaLink



**Figura 2**

Equipo SignaLink trabajando en el hangar



## 1.2. Datos de contacto

Puedes encontrar a SignaLink en:

- Mail: [signalink2025@gmail.com](mailto:signalink2025@gmail.com)
- Página web: [signalink2025.com](http://signalink2025.com)
- Instagram: [instagram.com/Signalink\\_ /](https://www.instagram.com/Signalink_/)
- GitHub: [github.com/impatrq/SignaLink](https://github.com/impatrq/SignaLink)

## 1.3. Docentes a cargo



**CARLASSARA FABRIZIO**

**Profesor de Prácticas  
Profesionalizantes**



**PALMIERI DIEGO**

**Profesor de Prácticas  
Profesionalizantes**



**MEDINA SERGIO**

**Profesor de Prácticas  
Profesionalizantes**



**SOLOMIEWICZ FEDERICO**

**Profesor de Prácticas  
Profesionalizantes**



**ARGÜELO GABRIEL**

**Profesor de Prácticas  
Profesionalizantes**



**BIANCO CARLOS**

**Profesor de Prácticas  
Profesionalizantes**

#### **1.4. Información adicional del proyecto**

##### **1.4.1. Tiempo total invertido en el desarrollo**

- Fecha de inicio: 07 de abril de 2025
- Duración: 27 semanas de trabajo (hasta ONIET)
- Esfuerzo individual: 14 horas de trabajo semanales
- Esfuerzo grupal: 378 horas de trabajo totales.



#### 1.4.2. Herramientas y programas utilizados

- **Visual Studio Code:** Editor de código que utilizamos para programar, depurar y organizar el software del proyecto de manera eficiente.
- **PlatformIO:** Entorno de desarrollo integrado que facilita la programación de los microcontroladores (ESP32 en nuestro caso), la gestión de librerías y la compilación del firmware de forma automatizada.
- **ESP-IDF:** Framework oficial de Espressif para programar microcontroladores ESP32 en C, con soporte para conectividad Wi-Fi, Bluetooth, periféricos y drivers de hardware.
- **Git/GitHub Desktop:** Sistema de control de versiones que usamos desde la consola para llevar un registro detallado de las modificaciones y coordinar el desarrollo de manera organizada.
- **PowerShell:** Medio de interacción directa con el sistema para ejecutar comandos, compilar programas y realizar configuraciones técnicas dentro de Raspberry Pi.
- **Python:** Lenguaje de programación que utilizamos en la Raspberry Pi para procesar datos, comunicarnos con los microcontroladores y ejecutar el motor de reconocimiento de voz.
- **Vosk:** Motor de reconocimiento de voz que integramos en el proyecto para convertir el habla en texto en tiempo real, optimizando la interacción con el sistema.
- **KiCad:** Software de diseño electrónico que empleamos para crear y simular circuitos, así como para elaborar las placas de circuito impreso (PCB).
- **AutoCAD:** Herramienta de diseño asistido que usamos para realizar planos, esquemas técnicos y piezas necesarias en el desarrollo del prototipo.
- **Canva:** Plataforma de diseño gráfico que utilizamos para elaborar material visual y presentaciones del proyecto de forma clara y atractiva.
- **Cura:** Software de laminado para impresión 3D que utilizamos para preparar los modelos en un formato compatible con la impresora, ajustando parámetros de temperatura, velocidad y calidad de impresión para obtener piezas precisas.

#### 1.5. Agradecimientos y reconocimientos:

En este apartado se mencionan a todas las **personas, sponsors, instituciones y empresas** que hicieron posible el desarrollo del Proyecto SignaLink:

**Queremos comenzar reconociendo el apoyo técnico y humano de:**

***Fabrizio Carlassara, Sergio Medina, Diego Palmieri, Gabriel Arguello, Federico Solomiewicz, Agustín Palmieri, Belén Avinceta, Ramiro Escalante Leiva, Tomas Bond.***

Su acompañamiento, tiempo y conocimientos fueron esenciales para resolver problemas técnicos, orientar nuestras decisiones y motivarnos en cada etapa de desarrollo.



**Sponsors, Instituciones y Empresas que impulsaron a SignaLink:**

- ***Universidad Argentina De la Empresa (UADE)***
- ***Newton Microscopios***
- ***Baires Divan***
- ***Asociación Cooperadora IMPA***
- ***Urban Belts***
- ***Ciampone & Franco Abogados***
- ***Micro Automación Argentina***
- ***ELECTRÓNICA ELEMON S.A.***

Queremos expresar nuestro **agradecimiento a cada uno**. Más allá de que no todos pertenezcan directamente al rubro tecnológico o educativo, cada aporte (ya sea en forma de materiales, asesoramiento o apoyo económico) fue fundamental para que este proyecto pudiera avanzar. Gracias a todos por su confianza y compromiso, hoy SignaLink es una realidad en construcción. Este acompañamiento no solo nos brinda recursos, sino que también nos motiva a seguir trabajando con dedicación, demostrando que cuando la comunidad se une en torno a una causa, es posible generar un impacto positivo en la sociedad.

## **Introducción**

### **2.1. Descripción general del proyecto**

El proyecto SignaLink tiene como objetivo principal el diseño y desarrollo de un dispositivo portátil e innovador de traducción de lengua de señas a audio y de voz a texto en tiempo real. Este sistema busca eliminar las barreras de comunicación que enfrentan las personas sordas o hipoacúsicas en situaciones cotidianas.

El prototipo está compuesto por dos módulos principales: uno en la mano/muñeca y otro en el pecho. Estos módulos trabajan de manera conjunta para registrar los gestos del usuario, procesar la información y permitir la comunicación en ambos sentidos. Gracias a esta



integración, el sistema facilita que una persona con discapacidad auditiva pueda expresarse a través de gestos y recibir, a su vez, mensajes claros de su interlocutor.

SignaLink no se limita a ser un prototipo escolar: constituye una plataforma tecnológica con potencial de crecimiento, que combina conocimientos en programación, electrónica y diseño. De este modo, se presenta como un proyecto de carácter inclusivo, interdisciplinario y escalable.

## **2.2. Origen y motivación del proyecto**

La iniciativa nace en la Escuela Técnica N.º 7 “Taller Regional Quilmes” (IMPA), en el marco de la especialidad de Aviónica, donde un grupo de seis estudiantes decidió aplicar sus aprendizajes a un problema social de gran relevancia.

La motivación del equipo se centra en **generar un impacto positivo y duradero en la sociedad**. Además, el proyecto representa un desafío académico equiparable al de nivel universitario, lo que les permite poner en práctica conocimientos técnicos avanzados en un contexto real y con un propósito inclusivo.

Un aspecto clave en este recorrido ha sido el acompañamiento de **Belén, nuestra bibliotecaria**, quien habitualmente enseña la lengua de señas en la institución. Su compromiso y ejemplo han servido de inspiración y guía para comprender la importancia de la comunicación inclusiva y reforzar la relevancia del proyecto.

## **2.3. Problemática social a resolver**

En la actualidad, más de un millón de personas en Argentina conviven con hipoacusia o sordomudez, una cifra que se multiplica si se toma en cuenta a toda Latinoamérica. A pesar de que la lengua de señas constituye la herramienta principal de comunicación para esta comunidad, la mayoría de la sociedad no la maneja, lo que genera una brecha estructural que limita la integración plena de estas personas.

Esta falta de entendimiento se traduce en obstáculos significativos en la vida cotidiana. Muchas veces deben enfrentar situaciones de aislamiento social, al no poder interactuar de



manera fluida en espacios comunes. En el ámbito educativo, las dificultades se reflejan en la ausencia de intérpretes y recursos inclusivos suficientes, lo que restringe sus posibilidades de aprendizaje.

En el plano laboral, la comunicación deficiente reduce las oportunidades de acceder a empleos de calidad o de participar en entrevistas en igualdad de condiciones. Incluso en actividades básicas, como realizar trámites, pedir ayuda en la vía pública o establecer vínculos sociales, la comunicación limitada obliga en muchos casos a depender de familiares o terceros para desenvolverse.

Más allá de la magnitud del problema, existe también una subestimación social: las dificultades de comunicación suelen percibirse como un tema secundario, cuando en realidad impactan de forma directa en la calidad de vida, en el acceso a derechos y en la construcción de oportunidades. Esta situación evidencia la necesidad urgente de generar soluciones inclusivas, innovadoras y accesibles, que contribuyan a reducir la exclusión y promuevan una sociedad más equitativa.

#### **2.4. Solución tecnológica propuesta**

La propuesta se centra en el desarrollo de un dispositivo portátil que permita establecer una comunicación fluida y en tiempo real entre personas sordas y oyentes. El enfoque del sistema es bidireccional, ya que contempla tanto la traducción de señas hacia voz como de voz hacia texto.

En primer lugar, los módulos ubicados en las manos incorporan varios tipos de sensores como los flex, giróscopos y acelerómetros, capaces de registrar el movimiento de los dedos y la orientación de la mano. La información recolectada es enviada a un microcontrolador, que procesa los datos y los convierte en un mensaje de voz emitido a través de un parlante. De manera complementaria, el dispositivo cuenta con un módulo de entrada de voz, compuesto por un micrófono integrado que capta las palabras del interlocutor oyente. Estas son transformadas en texto y proyectadas en un display colocado en la muñeca de la persona sorda o hipoacúsica, lo que posibilita que la comunicación se realice de manera natural y directa.

El sistema presenta diversas ventajas que lo convierten en una propuesta tecnológica viable. Su diseño portátil facilita el uso cotidiano y garantiza comodidad al usuario. Los



componentes empleados son de bajo consumo energético y poseen alta disponibilidad en el mercado, lo que reduce costos y asegura accesibilidad. A su vez, elimina la necesidad de contar con intérpretes en situaciones cotidianas, favoreciendo la independencia del usuario. Finalmente, el prototipo constituye una base sólida para la incorporación futura de inteligencia artificial, lo que permitirá aumentar la precisión y el alcance en la interpretación de gestos.

## 2.5. Metas y expectativas de logro

El proyecto contempla un conjunto de objetivos que se estructuran en tres etapas: inmediata, a mediano y a largo plazo. Estas metas reflejan tanto la factibilidad técnica del desarrollo como el impacto social esperado.

### Objetivos inmediatos

1. **Construcción del prototipo funcional:** Desarrollar una primera versión operativa del dispositivo que permita comprobar la viabilidad de la propuesta.
2. **Participación en las ONIET:** Presentar el prototipo en las Olimpiadas Nacionales de Innovación Educativa y Tecnología (Córdoba), validando el trabajo en un ámbito académico de prestigio.
3. **Validación técnica inicial:** Realizar pruebas de funcionamiento que confirmen la factibilidad técnica del sistema y su capacidad para cumplir con los requerimientos planteados.
4. **Aplicación de conocimientos adquiridos:** Demostrar que los estudiantes pueden integrar y aplicar competencias complejas de programación, electrónica y diseño en un contexto real y con impacto social.



## Metas a mediano y largo plazo

1. **Optimización del prototipo:** perfeccionar la velocidad, precisión y confiabilidad del sistema de traducción de señas, garantizando una experiencia de uso más fluida.
2. **Escalabilidad del proyecto:** evolucionar hacia un producto compacto, ergonómico y económicamente accesible para usuarios finales en Argentina y otros países.
3. **Integración de inteligencia artificial:** incorporar algoritmos de IA que permitan ampliar el repertorio de gestos reconocidos, adaptándose a variaciones individuales.
4. **Impacto social sostenido:** promover la inclusión de personas con discapacidad auditiva en diferentes ámbitos, tales como educación, empleo, servicios y vida cotidiana, reduciendo las barreras de comunicación.
5. **Proyección como modelo de innovación:** consolidar a *SignaLink* como un ejemplo de cómo la tecnología puede ponerse al servicio de causas sociales de alto impacto, inspirando futuras iniciativas inclusivas.

## Especificaciones Técnicas

### 3.1. Principio de funcionamiento del sistema

El sistema SignaLink se basa en la captura, procesamiento y conversión de datos provenientes de movimientos y señales acústicas para habilitar una comunicación bidireccional entre personas sordas y oyentes. Los gestos realizados con las manos se detectan mediante sensores de flexión, giroscopios y acelerómetros que convierten el movimiento en señales eléctricas; estas se digitalizan y se procesan en el módulo central. En sentido inverso, la voz del interlocutor se capta con micrófono, se procesa y se entrega como texto en el display del usuario.

Arquitectura por módulos. El prototipo opera con tres módulos: uno en cada mano (captura de gestos) y un módulo central en el pecho (procesamiento, E/S y gestión de energía).



### 3.1.1 Transporte y sincronización de datos.

- Entre los ESP32 de las manos y el ESP32 del pecho, la telemetría de sensores, en un principio se transmitía por Bluetooth Low Energy (BLE) usando perfil GATT. Los ESP32 de las manos funcionan como periféricos que publican características de sensores y envían notificaciones; el ESP32 del pecho actúa como central/cliente GATT, suscribiéndose a dichas características, pero analizando e investigando las posibilidades de su uso, se llegó a la conclusión de que para la conexión entre ESP32 de la muñeca al módulo de pecho, era de mayor conveniencia utilizar mqtt, el cual trabaja con suscripciones, y de ese modo nos ahorraron el uso del ESP32 del módulo de pecho, ya que con este nuevo método de comunicación se puede conectar de manera rápida y sencilla el ESP32 a la raspberry
- Dentro del módulo del pecho, la coordinación entre la Raspberry Pi (procesamiento de alto nivel: por ejemplo, TTS/STT, administración de UI) y el ESP32 se realizaba mediante UART (nivel TTL 3.3 V), donde la Raspberry Pi operaba como maestro/host y el ESP32 como dispositivo esclavo. El framing del enlace UART contemplaba encabezado, longitud, carga útil y checksum/CRC, además de comandos de control (STREAM ON, STREAM OFF, CONFIG, STATUS, ACK/NACK) para asegurar integridad y control de flujo. Al haber cambiado de comunicación el ESP32 se conecta a la raspberry directamente sin la necesidad de un intermediario, como sucedía anteriormente.

Esta separación de roles permitía baja latencia en la ruta gesto→voz (BLE con notificaciones) y determinismo en la coordinación local (UART), manteniendo el consumo energético bajo y la robustez del enlace.



### 3.2. Diagrama de bloques funcional

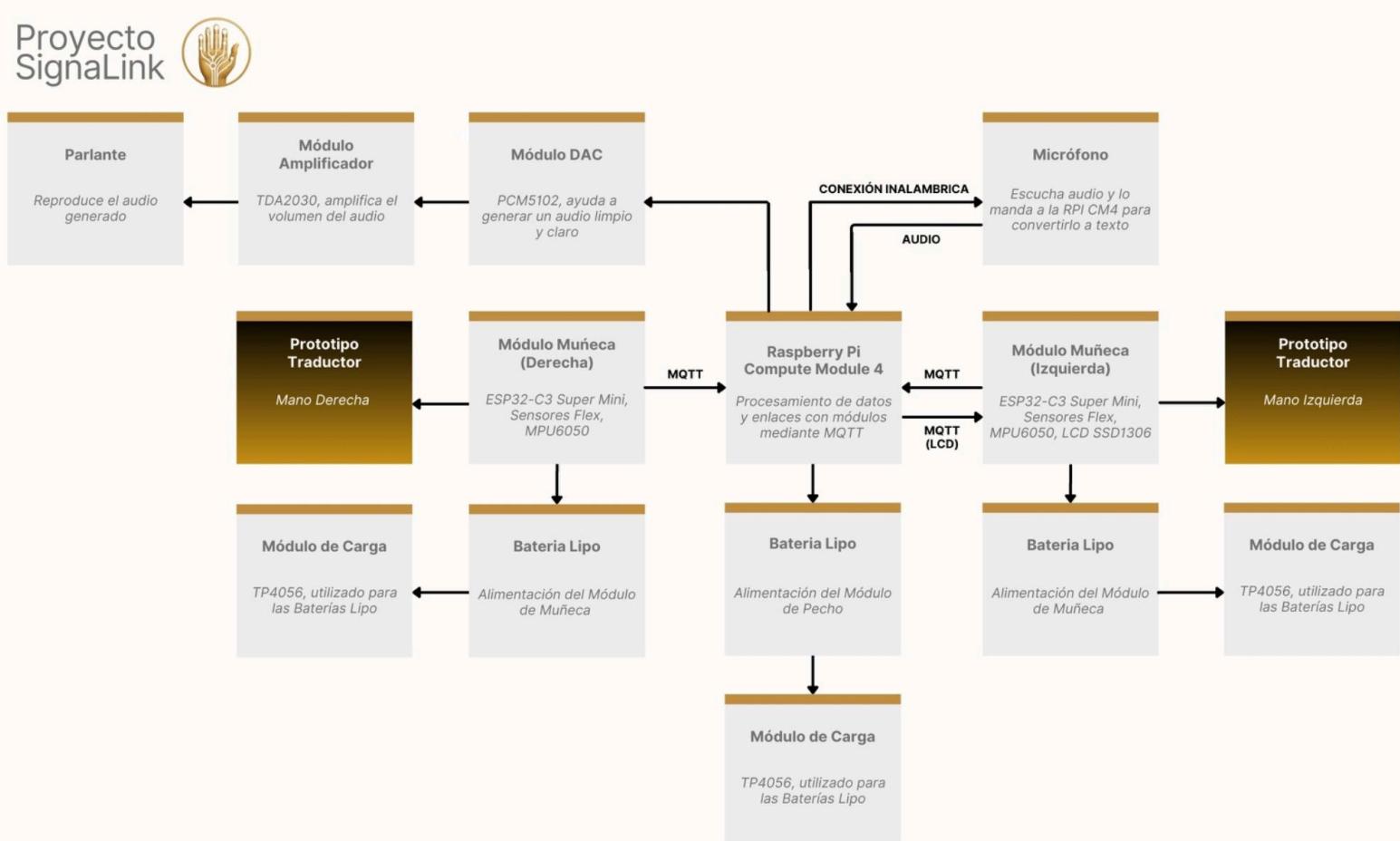


Figura 4 - Diagrama de bloques de la última versión sobre el funcionamiento

### 3.3. Sistema de alimentación y consumo energético

Fuentes y regulación. Cada módulo incorpora baterías LIPO recargables con protección (sobre/descarga y térmica) y reguladores DC-DC dedicados para sensores, MCU, radio y periféricos. La distribución contempla rieles separados para minimizar ruido en audio y mediciones.



### Gestión de consumo.

- **BLE:** se emplean intervalos de conexión y notificaciones calibradas para equilibrar latencia y autonomía; modos light-sleep en los ESP32 de mano cuando no hay movimiento.
- **UART/local:** la comunicación Raspberry-ESP evita radio innecesaria y reduce el costo energético de tareas de control.
- **Periféricos:** políticas de apagado parcial (backlight del display, mute de amplificador de audio, mic en espera) y duty-cycling del muestreo cuando el sistema está inactivo.

**Autonomía y seguridad.** La arquitectura prioriza bajo consumo en las manos y concentra picos en el pecho (audio/TTS). Se prevé autonomía operativa de jornada con recarga diaria, indicadores de estado de batería y protecciones para maximizar vida útil.

## Módulos

### 4.1. Módulo de la muñeca

#### 4.1.1. Componentes electrónicos empleados

- ESP32-C3 Super Mini
- OLED 0.91'
- Sensores Flex
- Microprocesador del MPU
- Diodos
- Resistencias
- Módulo de Carga
- Lipo 3.7 V 1200 mA

El sistema se basa en un **ESP32-C3 Super Mini**, que actúa como microcontrolador principal, encargado de procesar datos y transmitirlos de manera inalámbrica. La **pantalla OLED de 0.91"** permite visualizar información en tiempo real de forma clara y compacta.

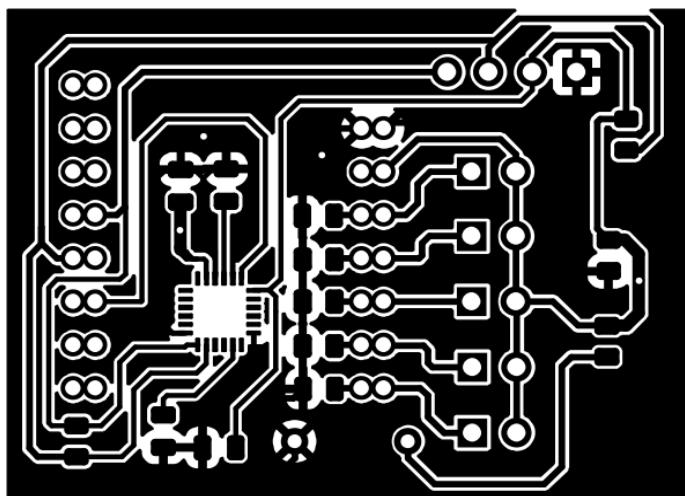
Para la detección de movimiento se utiliza el **MPU6050**, que integra acelerómetro y giroscopio, mientras que los **sensores flex** colocados en los dedos permiten captar la flexión de la mano para identificar gestos. Todos estos datos son recopilados por el ESP32 para su posterior envío.



En el circuito se incorporan **diodos** para protección de polaridad y **resistencias** para limitar la corriente en los puntos necesarios.

La alimentación se realiza mediante una **batería LiPo de 3.7 V**, que otorga portabilidad al sistema, y un **módulo de carga por USB**, que permite recargar la batería de forma segura sin desmontar el dispositivo.

#### 4.1.2. Diseño de la placa de circuito impreso (PCB)



#### 4.1.3. Modelado y diseño 3D del módulo

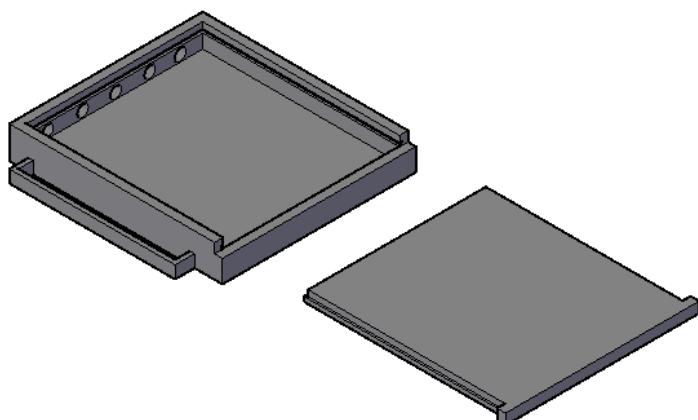




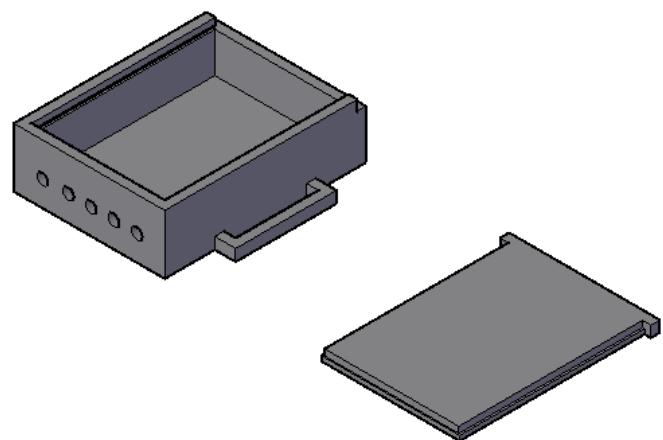
Diseño 1.0 en AutoCad



Diseño 1.1 en AutoCad

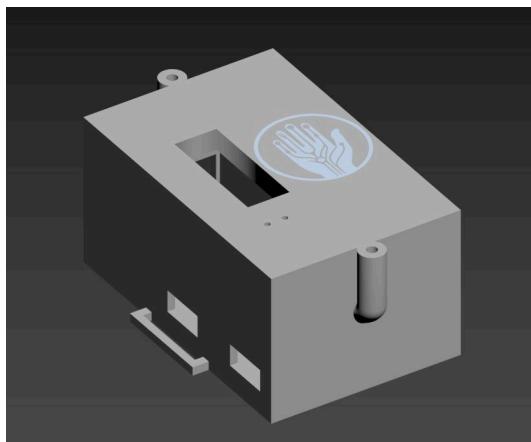


Diseño 2.0 en AutoCad





Diseño terminado en AutoCad

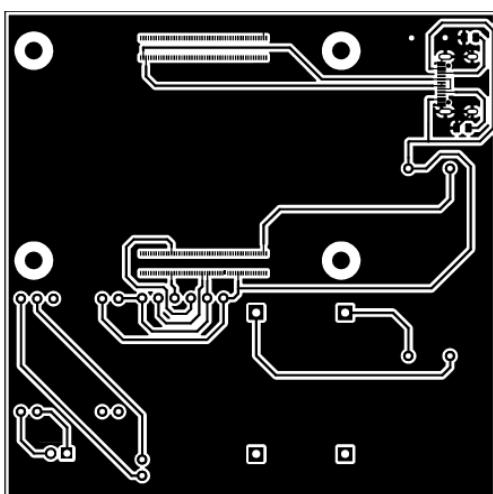


#### 4.2. Módulo del Pecho

##### 4.2.1. Componentes electrónicos empleados

- Raspberry Pi Computer Module 4
- Parlante 0,5W
- Rx y Tx del micrófono
- Elevador de voltaje
- Bateria Lipo 3.7 V 1200mA

##### 4.2.2. Diseño de la placa de circuito impreso (PCB)





Actualmente, esta sección del prototipo se encuentra en fase de desarrollo. Recientemente incorporamos la **Raspberry Pi Compute Module 4 (CM4)** al sistema, tras realizar la migración desde la **Raspberry Pi Zero 2 W**, con el objetivo de obtener un mayor rendimiento de procesamiento y una mejor capacidad de expansión.

Debido a limitaciones de tiempo, aún no ha sido posible integrar la CM4 dentro de una **placa PCB personalizada**, la cual se encuentra en etapa de diseño. No obstante, se logró poner en funcionamiento el módulo utilizando la **IO Board**, que permite el acceso a sus puertos **USB, GPIO, HDMI y de alimentación**, posibilitando así las pruebas preliminares de software y comunicación con los demás componentes del sistema.

Finalmente, tenemos como meta **completar el diseño e implementación de la placa definitiva hacia fines de noviembre**, con el objetivo de presentarla plenamente funcional durante la **exposición anual de nuestra institución**.

#### 4.2.3. Modelado y diseño 3D del módulo



**Figura:** Primer Diseño conceptual del Módulo de Pecho



## Tecnologías

### 5.1. Lenguajes de programación implementados

El desarrollo de este proyecto requirió un enfoque meticuloso en la arquitectura de software, aprovechando varios lenguajes de programación para abordar requisitos funcionales distintos. La selección de cada lenguaje se basó en sus fortalezas específicas y su idoneidad para la tarea en cuestión.

#### 5.1.1. Lenguaje de programación en C

Este tipo de lenguaje fue la piedra angular de nuestro desarrollo de sistemas embebidos. Fue elegido específicamente por su control de bajo nivel y eficiencia, que son primordiales al trabajar con microcontroladores e interfaces de sensores.

#### Aplicaciones claves:

**-ESP32-C3 Super Mini:** Se utilizó C para programar, un microcontrolador potente y compacto. Esta elección fue estratégica, ya que C proporciona acceso directo a los periféricos de hardware del dispositivo, permitiendo un control granular y una utilización óptima de los recursos.

**-Integración de Sensores:** El lenguaje facilitó la integración y adquisición de datos sin fisuras de varios sensores, incluidos los sensores flex y un acelerómetro-giróscopo. Su naturaleza orientada al rendimiento aseguró el procesamiento de datos en tiempo real, lo cual es fundamental para mediciones precisas y un control receptivo.

**-Interfaz LCD:** C también se empleó para gestionar la comunicación con la pantalla LCD, lo que permitió la presentación clara y eficiente de datos en tiempo real al usuario.

#### Justificación de la Selección del Lenguaje en C:

La decisión de usar C estuvo fuertemente influenciada por su compatibilidad con el ESP-IDF (Espressif IoT Development Framework). Este framework proporciona un conjunto completo de librerías, controladores y ejemplos que agilizan significativamente el proceso de desarrollo para dispositivos ESP32. Al utilizar C dentro del entorno ESP-IDF, pudimos aprovechar bases de código y ejemplos existentes y bien documentados, lo que aceleró el desarrollo y aseguró una base robusta y estable para el firmware del proyecto.



### 5.1.2. Lenguaje de programación Python

Python sirvió como el lenguaje principal para toda la lógica de alto nivel y el control ejecutado en la Raspberry Pi Zero 2W. Su legibilidad, su extenso ecosistema de librerías y sus capacidades de prototipado rápido lo convirtieron en la opción ideal para este rol.

#### Aplicaciones claves:

- Orquestación del Sistema:** Los scripts de Python fueron responsables de orquestar el sistema en general, gestionando los flujos de datos y manejando la lógica que conecta el sistema embebido con otros componentes.
- Procesamiento de Datos:** Sus potentes librerías de manipulación de datos se utilizaron para procesar los datos de los sensores recibidos del ESP32, transformando las mediciones sin procesar en información significativa.
- Protocolos de Comunicación:** Python fue fundamental para establecer y gestionar los protocolos de comunicación, asegurando una transferencia de datos confiable entre las diferentes partes del sistema.

#### Justificación de la selección de lenguaje Python:

Se eligió Python para la Raspberry Pi Zero 2W debido a su versatilidad y su facilidad de uso, lo cual es crucial para el desarrollo en un entorno de mini-computadora. Su vasta colección de librerías, como las dedicadas a la comunicación serial, el procesamiento de datos y la interacción con APIs, permitió implementar funcionalidades complejas de manera eficiente. Además, la naturaleza interpretada de Python facilitó un ciclo de desarrollo rápido, permitiendo realizar pruebas y ajustes en tiempo real sin necesidad de compilación, lo que fue una ventaja significativa para la fase de prototipado.

### 5.1.3. Desarrollo Web

La interfaz de usuario y la accesibilidad del proyecto fueron un enfoque clave, lo que llevó al desarrollo de un portal web dedicado. Esto requirió el uso sinérgico de tres lenguajes web fundamentales: HTML, CSS y JavaScript.

#### Aplicaciones claves:

- HTML (HyperText Markup Language):** Este lenguaje proporcionó la estructura fundamental para el sitio web, definiendo el diseño y el contenido de cada sección.
- CSS (Cascading Style Sheets):** Se utilizó CSS para aplicar un diseño cohesivo y visualmente atractivo, garantizando una experiencia intuitiva y fácil de usar. Controló aspectos como colores, fuentes y diseño, haciendo que la información fuera fácilmente digerible.



**-JavaScript:** Javascript introdujo elementos dinámicos e interactivos en el sitio web. Fue responsable de manejar las interacciones del usuario, recuperar datos y actualizar el contenido en tiempo real. Esto hizo que el sitio web fuera no sólo un recurso informativo, sino también un panel de control en vivo para monitorear el estado del prototipo.

### Justificación de la selección de lenguajes

Esta combinación de tecnologías fue elegida por su universalidad, compatibilidad entre plataformas y capacidad para crear una buena experiencia de usuario, receptiva y accesible. El sitio web sirve como un recurso completo que detalla los requisitos, la metodología y el progreso del proyecto, proporcionando una visión general clara y accesible para las partes interesadas y los usuarios por igual.

### 5.2. Uso del framework ESP-IDF

El framework ESP-IDF (Espressif IoT Development Framework) fue un componente fundamental en el desarrollo del firmware para el microcontrolador ESP32-C3 Super Mini. Su selección no fue casual, sino que respondió a la necesidad de contar con un entorno de desarrollo robusto, completo y optimizado para las capacidades de la familia de chips ESP32.

Este framework, desarrollado y mantenido por Espressif Systems, se utilizó principalmente para las siguientes funciones:

**-Configuración y Gestión de Periféricos:** El ESP-IDF proporcionó las bibliotecas y herramientas necesarias para la configuración y control preciso de los periféricos del ESP32-C3. Esto incluyó la inicialización de los sensores de flexión, el acelerómetro-giróscopo y la pantalla LCD. El framework nos permitió interactuar directamente con el hardware, habilitando los pines de entrada/salida (GPIO), configurando los protocolos de comunicación (como I2C o SPI para los sensores) y gestionando la comunicación con la pantalla.

**-Manejo de Tareas y Conurrencia (RTOS):** Una de las mayores ventajas del ESP-IDF es su integración con FreeRTOS, un sistema operativo en tiempo real (RTOS). Esto fue crucial para la programación del proyecto, ya que permitió dividir el código en múltiples tareas concurrentes. Por ejemplo, una tarea se encargaba de leer datos de los sensores, otra de procesar la información y una tercera de actualizar la pantalla LCD. Esta arquitectura de tareas no solo mejoró el rendimiento general del sistema, sino que también hizo el código más modular, legible y fácil de mantener.

**-Acceso a la Conectividad Inalámbrica:** El ESP-IDF facilitó la implementación de funcionalidades de conectividad. A través de sus bibliotecas, pudimos configurar y gestionar las conexiones Wi-Fi y Bluetooth del ESP32-C3, abriendo la puerta a futuras capacidades de



comunicación y telemetría.

**-Aprovechamiento de Ejemplos y Componentes Preexistentes:** La rica colección de ejemplos y componentes del ESP-IDF fue de gran valor durante el proceso de desarrollo. En lugar de escribir código desde cero para cada funcionalidad, pudimos adaptar y reutilizar ejemplos bien documentados para tareas comunes, como la lectura de sensores o la inicialización de la pantalla. Esta aproximación no solo aceleró el desarrollo, sino que también nos permitió aprender las mejores prácticas de programación para el hardware de Espressif y asegurar la fiabilidad del software.

### 5.3. Configuración del entorno virtual de desarrollo

Para el desarrollo del proyecto en la Raspberry Pi Zero 2W, se implementó un entorno virtual utilizando el módulo **venv** de Python. Esta práctica es fundamental y se considera un estándar en el desarrollo de software, ya que garantiza el aislamiento de dependencias y la reproducibilidad del proyecto.

Un entorno virtual es un directorio que contiene una instalación de Python y una serie de librerías específicas para un proyecto, completamente aisladas de la instalación global de Python del sistema operativo. Esto evita conflictos entre las diferentes versiones de librerías que pueden requerir proyectos distintos.

#### 5.3.1. Proceso de creación

La creación del entorno virtual se realizó mediante la siguiente sintaxis en la línea de comandos:

**En la terminal:** `python3 -m venv`

Este comando crea una carpeta que alberga la estructura del entorno virtual. Dentro de esta carpeta, se genera una copia del intérprete de Python y subdirectorios como bin, que contiene los ejecutables, y lib, donde se almacenarán las librerías instaladas.

#### 5.3.2. Activación del entorno virtual

Una vez creado, el entorno virtual debe ser activado para que la Raspberry Pi comience a usar el intérprete y las librerías de ese entorno en lugar de los del sistema. La activación se logra ejecutando el script activate que se encuentra en el directorio bin del entorno:

**En la terminal:** `source venv/bin/activate`

Al ejecutar esto, el nombre del entorno virtual cambia a activo.



### 5.3.3. Instalación de dependencias

**En la terminal:** [pip install](#)

La principal ventaja de este método es que las librerías se instalan únicamente en el directorio del entorno virtual, sin modificar la configuración global del sistema

### 5.3.4. Ventajas y beneficios

**-Aislamiento:** Se evita la sobrecarga o la contaminación del sistema operativo con librerías específicas de un proyecto. Esto mantiene la estabilidad y el orden del sistema.

**-Replicabilidad:** El entorno virtual facilita la replicación exacta de un entorno de desarrollo. Con un archivo de requisitos (requirements.txt), se pueden instalar todas las dependencias del proyecto en cualquier lado de forma automática.

**-Control de versiones:** Permite el uso de versiones de librerías diferentes, resolviendo posibles conflictos de dependencias. Por ejemplo, un proyecto puede requerir la versión 1.0 de una librería mientras que otro necesita la versión 2.0.

## 5.4. Aplicaciones de inteligencia artificial en el proyecto

La implementación de inteligencia artificial (IA) es un pilar fundamental en el desarrollo de este prototipo, cuyo objetivo es superar las barreras inherentes a la variabilidad en la ejecución del lenguaje de señas. A diferencia de las señas estandarizadas, la comunicación se caracteriza por la singularidad individual, donde cada persona ejecuta los movimientos a una velocidad y con una forma particular, lo que puede generar ambigüedad en su interpretación.

Para abordar este desafío, se ha diseñado un enfoque basado en el aprendizaje automático supervisado, que permite al sistema adaptarse a las características únicas de cada usuario. La metodología se centra en el entrenamiento de un modelo de IA utilizando el framework de código abierto TensorFlow, reconocido por su robustez en el desarrollo de modelos de aprendizaje profundo.



#### 5.4.1. Proceso de Entrenamiento y Recopilación de Datos

**-Captura de Movimientos:** El usuario realiza una señal específica de forma repetida, con un rango de 20 a 30 veces por cada señal a programar. Esta repetición es crucial para que el modelo aprenda a reconocer la variabilidad natural del movimiento del usuario.

**-Registro de Datos:** Los datos capturados de los sensores (como el giroscopio-acelerómetro y los sensores flex) durante cada repetición se registran y se almacenan como una serie de puntos de datos. Esta información, que representa la posición, aceleración y curvatura de la mano, se organiza de manera estructurada en un archivo en formato CSV.

**-Almacenamiento Local:** Los archivos CSV se guardan directamente en el dispositivo principal del prototipo, la Raspberry Pi Zero 2W. Esta elección permite que todo el proceso de entrenamiento y el archivo de datos permanezcan en el sistema, asegurando la autonomía y privacidad del usuario sin necesidad de conexión a servicios en la nube.

#### 5.4.2. Adaptación del modelo

Una vez que se han recopilado los datos para todas las señales programadas, el modelo de IA se somete a un proceso de entrenamiento iterativo. El modelo de TensorFlow utiliza los datos del archivo CSV para ajustar sus parámetros, aprendiendo a asociar el patrón de movimiento de cada usuario con la señal correspondiente. Este enfoque garantiza que el sistema no dependa de una base de datos de señales genéricas, sino que se personalice para el usuario que lo va a utilizar.

El objetivo de este proceso es crear un modelo de IA que minimice los errores de interpretación y la confusión entre señales similares. Esto permite que el usuario se comunique de manera fluida y efectiva, eliminando la frustración que podría surgir de la falta de reconocimiento. La implementación de IA, por lo tanto, no es solo una característica técnica, sino una solución fundamental para garantizar la usabilidad y la eficacia del prototipo en un entorno real.

#### 5.5. Comunicación mediante Bluetooth Low Energy (BLE)

En principio cabe aclarar que la comunicación BLE fue reemplazada por la comunicación MQTT, ya que este contenía muchas problemáticas a la hora de las conexiones y en el procesamiento de los códigos, en principal había fallas al ensamblar los códigos del MPU6050 (giroscopio-acelerómetro) y el LCD oled(ssd1306), debido a que ambos comparten la comunicación I2C. Sin consecuencia de esto se encontrará explicado todo el procedimiento.



### Arquitectura de comunicación:

La conectividad en nuestro prototipo se concibió bajo un enfoque modular, distribuyendo las responsabilidades entre dos microcontroladores para optimizar el rendimiento y la eficiencia energética. La arquitectura principal se basa en la comunicación inalámbrica a través de Bluetooth Low Energy (BLE), complementada con un enlace de datos por cable a la unidad central de procesamiento.

#### 5.5.1. Roles de dispositivos

El sistema está compuesto por dos módulos principales, cada uno equipado con un microcontrolador ESP32-C3 Super Mini. La asignación de sus roles, inspirada en los conceptos de un GATT-Server y GATT-Client.

**-Módulo de muñeca(GATT-CLIENT):**Este ESP32-C3 Super mini , ubicado en la muñeca del usuario, actúa como cliente en la red BLE. Su función principal es la adquisición de datos en la fuente. Es el punto donde se recopila la información crítica del movimiento a través de varios sensores:

**-Sensores flex:** Varían su resistencia, dependiendo de la curvatura de los dedos

**-Sensor Giroscopo-acelerómetro:** Registra la orientación y movimiento de la mano en el espacio

**-Display Lcd:**Muestra información en tiempo real al usuario.Este módulo es el encargado de empaquetar los datos, para enviarlos de forma inalámbrica

**-Módulo de pecho(GATT-SERVER):** El segundo Esp32 c3 super mini, posicionado en una ubicación central en el torso del usuario, funciona como el servidor. Su rol es ser un punto de agregación de datos. Se encuentra esperando recibir la información transmitida por el módulo de la muñeca. Al recibir los datos, su única misión es reenviarlos, de manera eficiente, a la unidad de procesamiento principal.

#### 5.5.2. Flujo de Datos y Justificación Técnica

**-Enlace Inalámbrico (BLE):** La conexión entre el módulo de la muñeca (cliente) y el módulo del pecho (servidor) se realiza mediante Bluetooth Low Energy. Esta tecnología fue seleccionada por su bajo consumo de energía, una característica vital para dispositivos portátiles que operan con batería. El cliente envía los datos de los sensores al servidor de manera regular, estableciendo un enlace de comunicación robusto y de baja latencia.



**-Enlace por Cable (UART):** Una vez que el ESP32-C3 Super Mini del pecho (servidor) ha recibido los datos, estos se transmiten a la Raspberry Pi Zero 2W a través de una conexión UART (Universal Asynchronous Receiver/Transmitter). La elección de una conexión por cable para esta etapa final fue el asegurar una transmisión de datos fiable y de alta velocidad, eliminando cualquier posible interferencia inalámbrica en el último tramo antes del procesamiento.

## 5.6. Comunicación mediante MQTT (Message Queuing Telemetry Transport)

Una vez que nos dimos cuenta que la mejor idea era utilizar MQTT, nos centramos profundamente en esto. El diseño inicial de la conectividad fue revisado para adoptar un paradigma de comunicación más eficiente y escalable. En lugar de una red de comunicación punto a punto con BLE, la arquitectura actual se centra en el protocolo MQTT. Este cambio no solo simplifica el sistema al reducir el número de módulos Esp32 necesarios, ya que se puede realizar la conectividad del esp32 ubicado en la muñeca a la raspberry, sino que también establece un marco de comunicación más flexible y confiable.

### 5.6.1. Publicación y suscripción

El corazón de este nuevo diseño es el modelo de publicación/suscripción inherente a MQTT. En lugar de que un dispositivo se conecte directamente a otro (como en BLE), los dispositivos se comunican a través de un intermediario central, conocido como bróker. Este enfoque distribuye el flujo de datos de una manera mucho más limpia:

- Un dispositivo publica un mensaje en un tópico específico.
- Cualquier dispositivo que esté "suscripto" a ese mismo tópico recibe el mensaje.  
De esta manera nos ahorraremos las conexiones directas, las cuales podrían traer fallas de conexión en algún futuro

### 5.6.2. Componentes y Flujo de Datos

**-El Cliente:** Este módulo, ubicado en la muñeca del usuario, es el "publicador" de los datos. Su única tarea es recopilar la información de los sensores(el giróscopo y acelerómetro, los sensores flex) y enviarla al broker. Además, continúa gestionando la pantalla LCD para proporcionar retroalimentación visual al usuario. Al operar como un cliente MQTT, su código se simplifica enormemente, ya que no necesita preocuparse por la conexión directa con otro ESP32. Solo necesita conectarse a la red y publicar los datos en un tópico predefinido.

**-El Broker:** La Raspberry Pi Zero 2W asume el rol del broker MQTT, actuando como el servidor central y el cerebro de toda la operación. Es el punto de encuentro donde se reciben todos los datos de los sensores. La Raspberry Pi está constantemente "suscrita" a los tópicos



relevantes y, al recibir la información, la procesa para la interpretación del lenguaje de señas mediante el modelo de inteligencia artificial. Esta consolidación de funciones en un solo dispositivo reduce la complejidad del hardware y simplifica la gestión del sistema.

#### Tópicos:

Contamos con dos tópicos, “sensors/mpu\_flex” y “display/lcd”. En el primer tópico nombrado es por donde se van a enviar los datos que se generan a medida que se van haciendo las señas, y el segundo tópico va a ser utilizado para una vez inicializado el programa y el lcd se encienda, va a enviar un ‘ON’; gracias a este mensaje el micrófono empieza a funcionar y tomar las frases, estas mismas van a ser enviadas al mismo tópico, las cuales recibirá el lcd y las imprimira en la pantalla.

Para la implementación del bróker, se utiliza Mosquitto, un ligero bróker MQTT. Su eficiencia lo hace ideal para el entorno de recursos limitados de la Raspberry Pi Zero 2W. La configuración del punto de acceso local, utilizando la herramienta hostapd, fue crucial para establecer una red segura y privada.

#### 5.6.3. Red local

Uno de los aspectos más importantes de este nuevo método de comunicación es la configuración de una red local, eliminando la dependencia de redes Wi-Fi públicas. Esta decisión se tomó para mitigar riesgos de seguridad y asegurar la estabilidad de la conexión. La red, nombrada **SignaLink\_AP**, opera como un punto de acceso privado al que sólo los dispositivos que cuenten con la contraseña pueden conectarse. Esto se logró configurando **HostAPD**, el servicio encargado de que la Raspberry Pi funcione como un punto de acceso Wi-Fi. En su configuración se definieron parámetros como la interfaz de red utilizada, el nombre de la red, la contraseña, el canal y el tipo de señal.

Para que los dispositivos se conecten automáticamente y reciban una dirección IP válida, se configuró **dnsmasq** como servidor DHCP y DNS básico. Este servicio se encarga de asignar direcciones dentro del rango **192.168.4.x** a todos los clientes que se unan a la red, mientras que la Raspberry Pi mantiene la dirección estática **192.168.4.1**, funcionando como puerta de enlace.

También se configuraron los servicios para que se inicien automáticamente al arrancar la Raspberry Pi. De esta forma, la interfaz de red se levanta con la IP correcta, luego HostAPD crea la red, dnsmasq reparte las direcciones IP y finalmente **Mosquitto** se ejecuta junto con el script de Python del proyecto. Esto asegura que el sistema siempre esté operativo al encenderlo, sin necesidad de intervención manual.

#### 5.7. Inteligencia Artificial

El componente de inteligencia artificial (IA) constituye el núcleo del procesamiento cognitivo del sistema SignaLink. Su propósito principal es interpretar los datos recibidos desde el



módulo de la muñeca —provenientes de los sensores flex y del giroscopio-acelerómetro (MPU6050)— y traducirlos en texto en tiempo real.

Para esta etapa se emplea la Raspberry Pi Compute Module 4, que ejecuta los modelos de aprendizaje automático desarrollados y entrenados mediante TensorFlow. En el repositorio del proyecto, dentro de inteligencia-artificial/TensorFlow, se encuentran los scripts y datasets utilizados para el entrenamiento del modelo, así como los algoritmos de inferencia optimizados para dispositivos de bajo consumo.

### **5.7.1. Entrenamiento del modelo**

El modelo fue entrenado a partir de un conjunto de datos recopilado manualmente durante la fase experimental del proyecto, donde se registraron lecturas simultáneas de los sensores flex y del MPU6050 correspondientes a diferentes señas básicas.

Cada señal fue etiquetada. Luego, utilizando TensorFlow, se entrenó una red neuronal ligera capaz de reconocer patrones en las combinaciones de flexión y orientación de la mano.

### **5.7.2. Inferencia en la Raspberry Pi**

El modelo entrenado se exporta en formato TensorFlow Lite (TFLite), lo que permite su ejecución eficiente en la Raspberry Pi Compute Module 4.

Durante la operación, el flujo de datos se da de la siguiente manera:

1. El ESP32 de la muñeca envía los datos de los sensores a través de MQTT al bróker Mosquitto ejecutado en la Raspberry Pi.
2. El script principal en Python recibe estos datos mediante una suscripción al tópico sensors/mpu\_flex.
3. Los valores son normalizados y enviados al modelo TFLite para su clasificación.
4. El resultado del modelo (la señal reconocida) se transforma en audio y se muestra en el parlante del módulo de pecho.

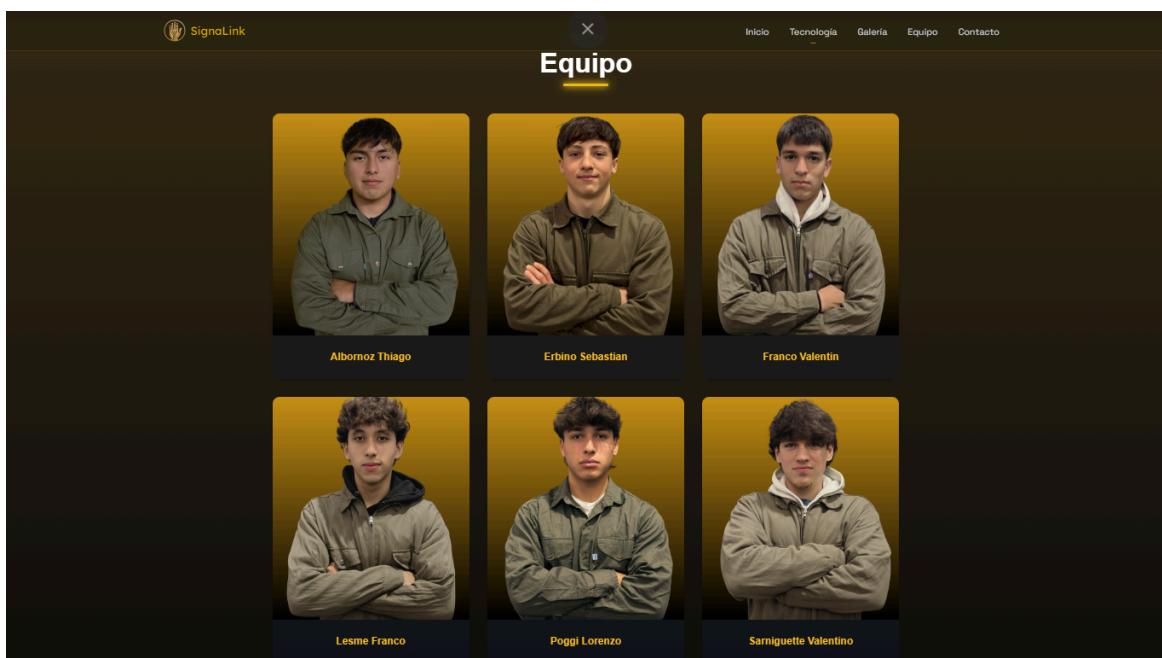
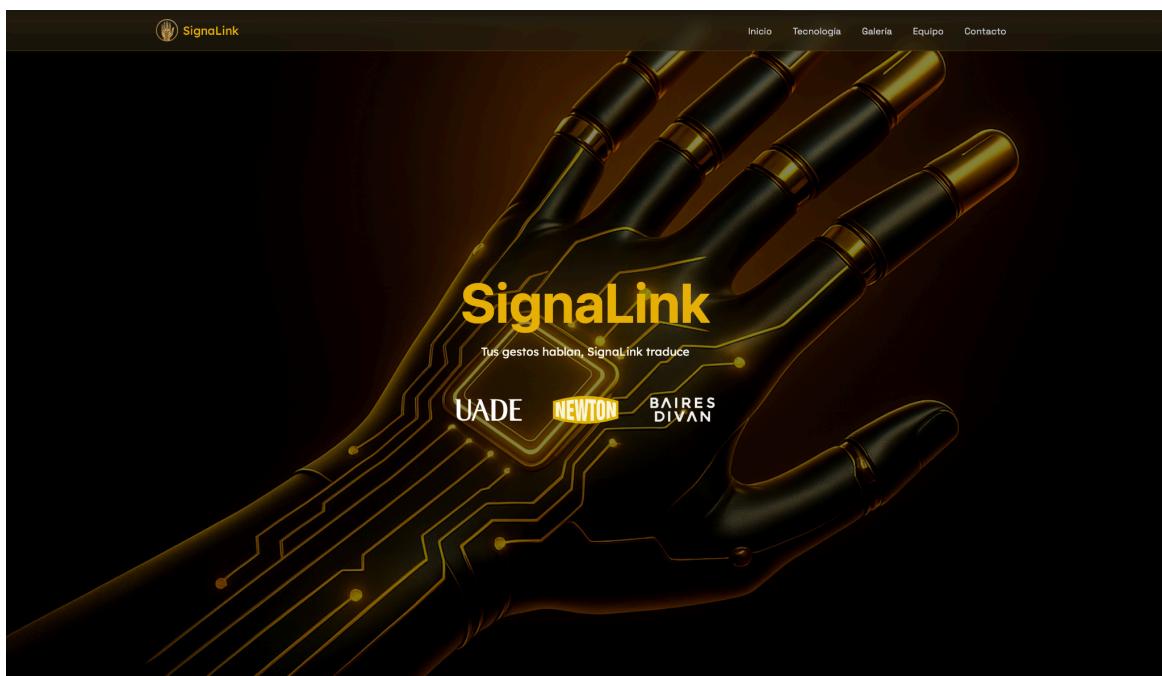
### **5.7.3. Ventajas del enfoque**

El uso de TensorFlow Lite y la arquitectura MQTT permite un flujo de datos liviano y modular, adaptable a futuros modelos más complejos sin necesidad de rediseñar el hardware.

Además, al ejecutarse de manera local en la Raspberry Pi, se evita depender de servicios en la nube, garantizando privacidad, baja latencia y funcionamiento completamente offline.



## Página Web





## 6.1. Introducción al sitio web

El sitio web oficial de *Signalink 2025* (<https://signalink2025.com>) constituye la **plataforma central de comunicación y difusión del proyecto**. Fue desarrollado con un enfoque moderno, ágil y adaptable a múltiples dispositivos, lo que permite transmitir de manera clara tanto el propósito general como los aspectos técnicos de la iniciativa.

Desde su concepción, el objetivo principal fue disponer de un espacio digital que integrara en un solo lugar la **historia del proyecto, los avances logrados, las tecnologías implementadas, la presentación del equipo de trabajo y la visibilidad de los sponsors** que nos acompañan. De este modo, la página funciona como una tarjeta de presentación completa, confiable y profesional ante instituciones, potenciales aliados y el público interesado en nuestro desarrollo.

## 6.2. Arquitectura y diseño de la página

El diseño de la web se pensó siguiendo un estilo minimalista pero altamente tecnológico, priorizando la experiencia de usuario. Se adoptaron lineamientos visuales modernos: tipografías limpias, uso de contrastes en la paleta de colores y animaciones que refuerzan la interacción dinámica.

La arquitectura está organizada en apartados principales —Inicio, Tecnología, Galería, Equipo y Contacto— de manera que la navegación sea intuitiva, rápida y ordenada. A nivel técnico, se aplicaron criterios de usabilidad y accesibilidad, garantizando que cualquier usuario pueda interactuar con la web sin dificultades. Además, el diseño es responsive, adaptándose a distintos dispositivos (computadoras, tablets y smartphones) sin perder calidad visual ni funcionalidad.

## 6.3. Secciones principales y funcionalidades

El sitio web está estructurado en apartados clave que resumen todo el alcance del proyecto:

- **Inicio:** Introducción al proyecto, presentación de la propuesta y visibilidad inmediata de los sponsors que colaboran.
- **Tecnología:** Explicación detallada de las herramientas empleadas, incluyendo software especializado, modelado 3D, integración de hardware y simuladores digitales que dieron soporte al desarrollo.



- **Galería:** Espacio visual con imágenes de los avances logrados, prototipos desarrollados y momentos clave del equipo en acción.
- **Equipo:** Presentación de cada integrante, destacando sus funciones, áreas de especialización y el rol que cumple dentro del proyecto.
- **Contacto:** Módulo que facilita la comunicación directa con la organización, ya sea para consultas técnicas, propuestas de colaboración o interés institucional.

Cada sección cumple un rol específico y en conjunto brindan una visión integral, mostrando tanto la base técnica como la dimensión humana del proyecto.

#### **6.4. Tecnología utilizada y metodología de desarrollo**

La web también sirve como escaparate para detallar los recursos tecnológicos empleados en Signalink 2025. Se destacan:

- Software de modelado 3D para el diseño y la simulación de prototipos.
- Hardware especializado en pruebas y validaciones técnicas.
- Plataformas de integración digital que permitieron vincular procesos de software con los desarrollos físicos.
- Lenguajes de programación y entornos de desarrollo web que garantizan una página veloz, estable y con animaciones fluidas.

La metodología aplicada combina principios de diseño iterativo e incremental, lo que permitió validar avances de forma temprana y optimizar la experiencia del usuario en cada fase del desarrollo.

#### **6.5. Sponsors, impacto y proyección**

Los sponsors que acompañan a Signalink 2025 son presentados en el sitio web como colaboradores clave en el proceso. Su apoyo permitió no solo financiar etapas críticas, sino también dar legitimidad y credibilidad al proyecto frente a nuevas instituciones interesadas.

El impacto de la página trasciende la simple difusión: se convierte en un punto de encuentro entre innovación, equipo y comunidad. Al narrar nuestra historia, detallar a quiénes ayudamos y mostrar cómo trabajamos, proyectamos un mensaje claro de profesionalismo y compromiso social.



De cara al futuro, la web se mantendrá como plataforma viva, actualizándose con los avances del proyecto y consolidándose como canal de comunicación técnico y estratégico.

## Imágenes

### 7.1. Componentes utilizados

En esta sección se presenta una descripción detallada de la función de cada componente empleado, incluyendo aquellos que fueron considerados como alternativas durante el proceso de selección, junto con una explicación de su funcionamiento.

**ESP32-C3 SUPER MINI**



El ESP32-C3 Super Mini es una placa de desarrollo compacta basada en el chip ESP32-C3 de Espressif, perteneciente a la familia ESP32. Incorpora un procesador RISC-V de 32 bits de un solo núcleo con una frecuencia de hasta 160 MHz, 400 KB de SRAM, 384 KB de ROM y una memoria flash integrada de 4 MB.

La placa es compatible con los protocolos Wi-Fi 802.11b/g/n en la banda de 2.4 GHz, y admite los modos Station, SoftAP, SoftAP+Station y modo mixto, además de contar con conectividad Bluetooth 5.0.

Dispone de 16 pines, de los cuales 11 son GPIO programables, que permiten el uso de funciones ADC, PWM, UART, I2C y SPI. Incluye los botones RST (reset) y BOOT, este último utilizado para poner la placa en modo bootloader al momento de cargar el código.

También cuenta con una interfaz USB-C, que permite alimentar la placa, cargar programas o realizar comunicación serial.



### XIAO-NRF52840



El Seeed Studio XIAO nRF52840 es una placa de desarrollo compacta y potente, basada en el SoC nRF52840 de Nordic Semiconductor. Está diseñada para dispositivos portátiles, aplicaciones de Internet de las Cosas (IoT) y proyectos de bajo consumo energético. La placa incorpora conectividad Bluetooth 5.0, comunicación NFC y una interfaz USB integrada que permite tanto la programación como la alimentación eléctrica. Además, es compatible con múltiples interfaces de comunicación serial, como UART, I2C (IIC) y SPI, y cuenta con una memoria flash integrada de 2 MB. La placa también ofrece soporte para Near Field Communication (NFC), lo que facilita la vinculación rápida de dispositivos y la implementación de soluciones de pago sin contacto. Este microcontrolador fue sustituido por el ESP32, ya que solamente se conseguía en el exterior.

### RASPBERRY PI ZERO 2W



La Raspberry Pi Zero 2 W es una computadora de placa única (single-board computer) compacta y potente, compatible con la mayoría de los gabinetes existentes de la serie



Raspberry Pi Zero. Incorpora un procesador Arm Cortex-A53 de 64 bits y cuatro núcleos, con una frecuencia de 1 GHz, e integra un chip Broadcom BCM2710A1 junto con 512 MB de memoria LPDDR2 SDRAM.

La placa también cuenta con conectividad inalámbrica Wi-Fi 802.11 b/g/n a 2.4 GHz y Bluetooth 4.2, con soporte para Bluetooth Low Energy (BLE). Incluye una ranura para tarjeta microSD, un conector CSI-2 para cámara, un puerto USB On-The-Go (OTG) y un espacio sin soldar para un encabezado GPIO de 40 pines compatible con HAT.

La Raspberry Pi Zero 2 W se alimenta mediante un conector micro USB, y su salida de video se realiza a través de un puerto mini HDMI

Este microcontrolador ya no se encuentra más en uso ya que migramos a la Raspberry pi compute module 4

#### RASPBERRY PI COMPUTE MODULE 4



La Raspberry Pi Compute Module 4 (CM4) es un potente Sistema en Módulo (SoM) que integra los componentes internos de la Raspberry Pi 4, incluyendo un procesador ARM Cortex-A72 de cuatro núcleos y opciones de memoria RAM de 1 GB, 2 GB, 4 GB u 8 GB.

Ofrece diversas configuraciones de almacenamiento, que incluyen versiones con 0 GB (Lite), 8 GB, 16 GB o 32 GB de memoria flash eMMC. El CM4 está diseñado para aplicaciones embebidas, permitiendo a los desarrolladores aprovechar la arquitectura de hardware y el ecosistema de software de Raspberry Pi en sistemas personalizados.

Supera a nuestra anterior Raspberry Pi, en prácticamente todos los aspectos de rendimiento, lo que lo convierte en una opción adecuada para una amplia variedad de aplicaciones.



### IO BOARD



La placa de expansión IO del Raspberry Pi Compute Module 4 es una interfaz versátil que permite conectar el Compute Module 4 con una amplia variedad de periféricos. Proporciona acceso a múltiples puertos de entrada/salida, incluyendo HDMI, USB, GPIO y PCIe, lo que la hace adecuada para prototipado, desarrollo e integración en aplicaciones personalizadas. La placa permite realizar simulaciones en tiempo real y es compatible con diversos proyectos, como la conexión con sensores y dispositivos de salida.

### LCD DISPLAY OLED SSD1306, 0.91”



La pantalla OLED I2C de 0,91 pulgadas y resolución 128 x 32 píxeles funciona mediante un controlador integrado SSD1306, encargado de gestionar el funcionamiento del display. Se comunica con el microcontrolador a través del protocolo I2C, lo que permite un bajo consumo de energía y una visualización de alta resolución.

La pantalla se alimenta mediante el pin VCC, que normalmente se conecta a la fuente de



alimentación del microcontrolador, y establece la comunicación con la placa a través de los pines SDA y SCL del bus I2C. Esta configuración permite mostrar texto, gráficos e información diversa en sistemas embebidos y proyectos con microcontroladores

#### LCD DISPLAY ST7735



La pantalla LCD TFT de 1.8 pulgadas con controlador ST7735 es un módulo de visualización a color de resolución 128 × 160 píxeles, diseñado para aplicaciones embebidas que requieren mostrar texto, gráficos o imágenes con buena calidad y bajo consumo energético.

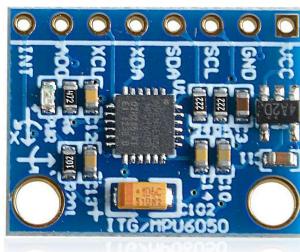
El controlador ST7735 integrado se encarga de gestionar la comunicación entre el microcontrolador y la pantalla mediante una interfaz SPI (Serial Peripheral Interface), lo que permite una transmisión rápida de datos.

El módulo cuenta con varios pines de conexión, entre ellos: **GND**(tierra); **VCC**: alimentación (generalmente 3.3V o 5V); **SCL** (reloj del bus SPI); **SDA** (datos del bus SPI); **RES** (reinicio del display); **DC** (selección entre datos y comandos); **CS** (selección del chip); **BLK** (control de la retroiluminación)

Este display fue reemplazado por LCD DISPLAY OLED SSD1306, ya que contiene una programación más sencilla y nos proporciona un tamaño ideal.

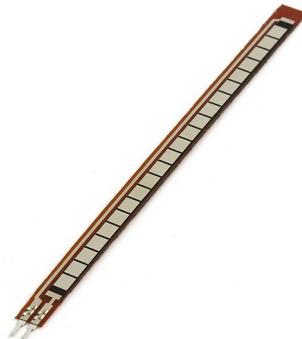


### MPU-6050



El MPU6050 funciona midiendo el movimiento de un objeto en tres dimensiones: X, Y y Z. Utiliza un acelerómetro de tres ejes para detectar la aceleración lineal y un giroscopio de tres ejes para medir la rotación angular. El sensor integra estas mediciones para ofrecer una unidad completa de procesamiento de movimiento. Su interfaz I2C permite una comunicación sencilla con microcontroladores, facilitando el procesamiento de los datos de movimiento. Además, su tecnología Motion Fusion™ proporciona una salida detallada de 9 ejes, ideal para aplicaciones que requieren seguimiento y control de movimiento de alta precisión.

### SENSORES FLEX





Los sensores flex, también conocidos como sensores de flexión, son sensores resistivos que modifican su resistencia eléctrica cuando se doblan o flexionan.

El principio de funcionamiento de un sensor flex se basa en el efecto piezorresistivo, que consiste en la variación de la resistencia eléctrica de un material al ser sometido a un esfuerzo mecánico. Cuando el sensor se dobla, la capa de tinta conductora experimenta compresión en el interior de la curva y tensión en el exterior, provocando que las partículas conductoras se acerquen o se separen entre sí. Esto genera una variación en la resistencia eléctrica del material, la cual es proporcional al grado de flexión, permitiendo así medir el ángulo de curvatura.

#### **MICRÓFONO INALÁMBRICO CORBATERO-C11**

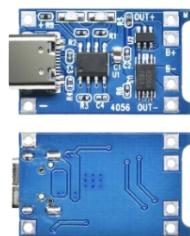


El Corbatero C11 es un dispositivo de captura de audio de alta calidad, diseñado para ofrecer una solución práctica, eficiente y discreta. Su formato compacto y su conectividad directa lo convierten en una herramienta indispensable para creadores que requieren movilidad sin comprometer la fidelidad sonora.

**Especificaciones del micrófono:** Posee un transmisor de audio portátil; contiene una entrada de audio integrado con rejilla superior para la captación vocal; cuenta con una luz led verde, indicadora de que el conector USB-C inalámbrico está funcionando de manera correcta.

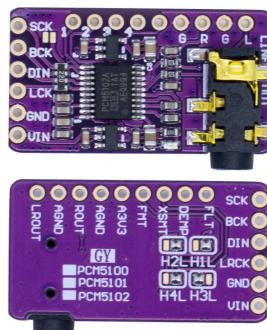


## MÓDULO DE CARGA TP-4056



El TP4056 funciona gestionando el proceso de carga de baterías de ion de litio en dos modos principales: corriente constante (CC) y voltaje constante (CV). Esto permite un proceso de carga seguro y eficiente. El circuito integrado incorpora varias funciones para garantizar la seguridad, incluyendo protección contra sobrevoltaje, bloqueo por bajo voltaje y regulación térmica. Estas características ayudan a prevenir daños tanto en la batería como en el cargador, asegurando la durabilidad y fiabilidad del sistema de batería.

## PCM5102

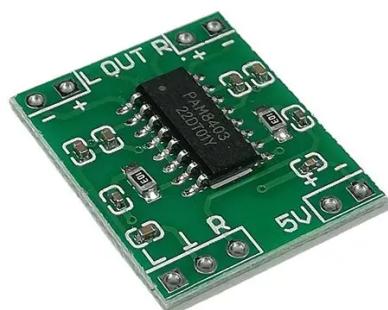


El DAC PCM5102 funciona convirtiendo señales de audio digital en una salida de audio analógica de alta calidad. Es compatible con el protocolo I2S para la entrada de datos de



audio y puede procesar audio de 32 bits con tasas de muestreo de hasta 384 kHz. Este DAC está diseñado para aplicaciones de audio de alta fidelidad, ofreciendo baja distorsión, bajo nivel de ruido y un alto rango dinámico.

#### **AMPLIFICADOR PAM 8403**



El PAM8403 es un chip amplificador digital que ofrece amplificación estéreo de audio de alta calidad. Es ampliamente utilizado en dispositivos portátiles debido a su bajo consumo de energía y mínima generación de calor. El PAM8403 es capaz de entregar una salida de 3W + 3W con excelente calidad de sonido y eficiencia, lo que lo convierte en una opción ideal para aplicaciones alimentadas por batería, es utilizado para amplificar señales de audio desde un conector de 3.5 mm para alimentar un altavoz.

#### **PARLANTE 0,5W 8Ω**





Este parlante funciona transformando señales eléctricas en sonido. Internamente, tiene una bobina móvil que está conectada a los terminales del parlante. Cuando esa bobina recibe corriente alterna (la señal de audio), se convierte en un electroimán. Al estar ubicada dentro del campo magnético de un imán permanente, la bobina vibra hacia adelante y hacia atrás según las variaciones de la señal.

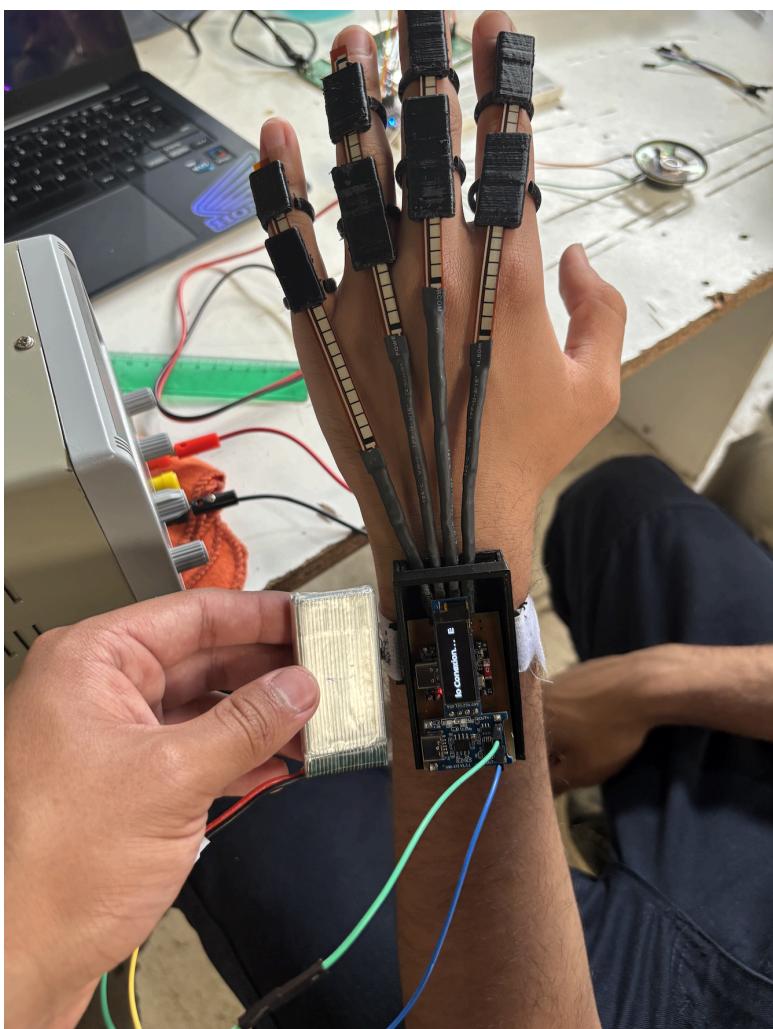
### REGULADOR DE VOLTAJE MT-3608



El MT 3608 funciona utilizando un inductor, un interruptor (como un MOSFET), un diodo y capacidores para aumentar un voltaje de entrada bajo a un voltaje de salida más alto. Cuando el interruptor está encendido, el inductor almacena energía en forma de campo magnético. Al apagarse, el campo magnético colapsa y libera esa energía, que se transfiere al capacitor de salida y a la carga. Este proceso se repite a alta velocidad, elevando efectivamente el voltaje de corriente continua (DC) de entrada a uno más alto. El circuito de retroalimentación monitorea el voltaje de salida y ajusta automáticamente la relación para mantener el voltaje establecido, asegurando una salida de voltaje DC estable y con mínimas fluctuaciones. El potenciómetro del módulo modifica la resistencia de retroalimentación, permitiendo ajustar el voltaje de salida, que puede alcanzar hasta 28V dependiendo del voltaje de entrada y de las condiciones de carga.

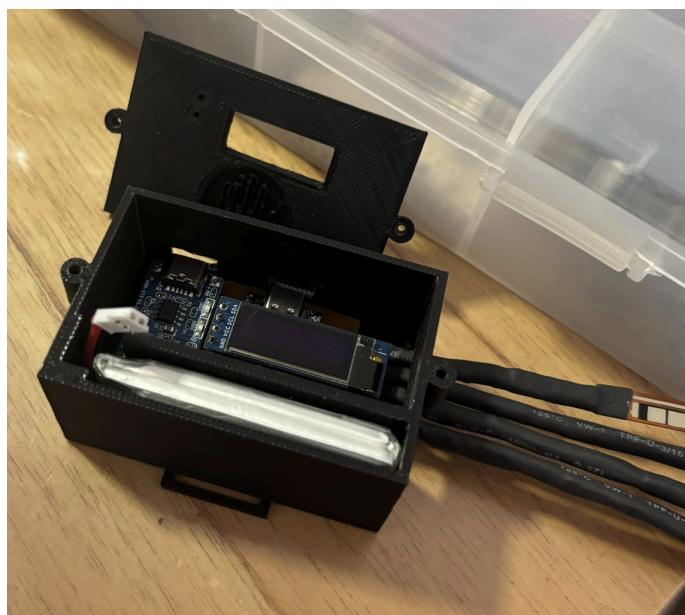


## 7.2. Prototipo



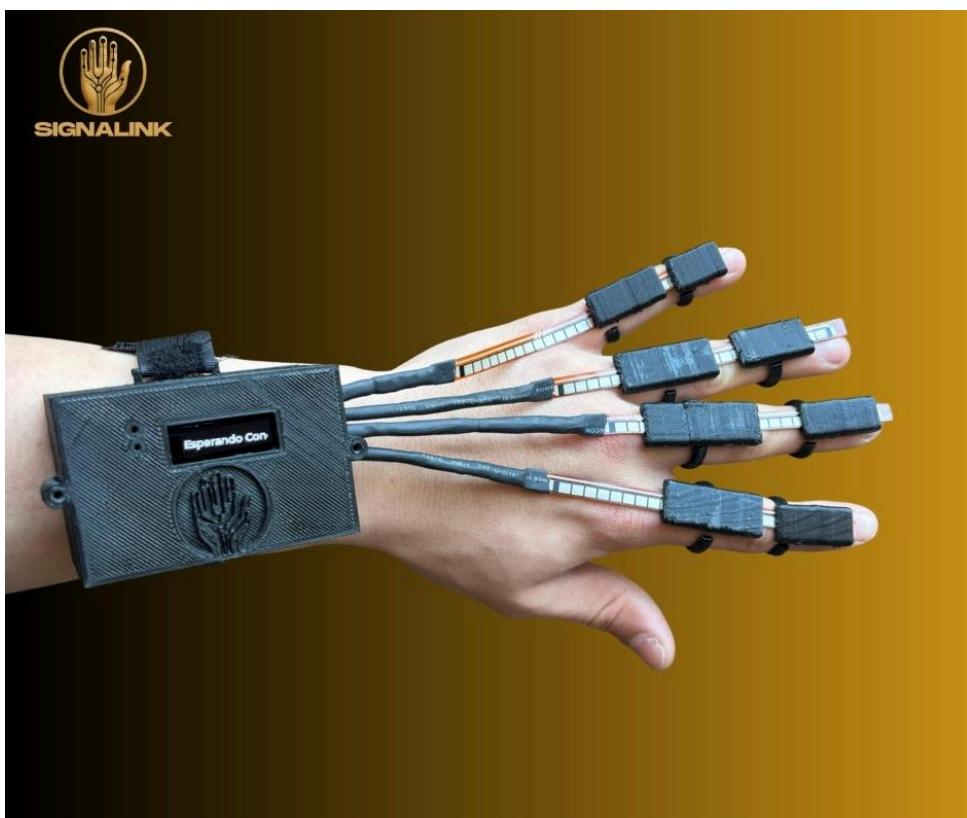
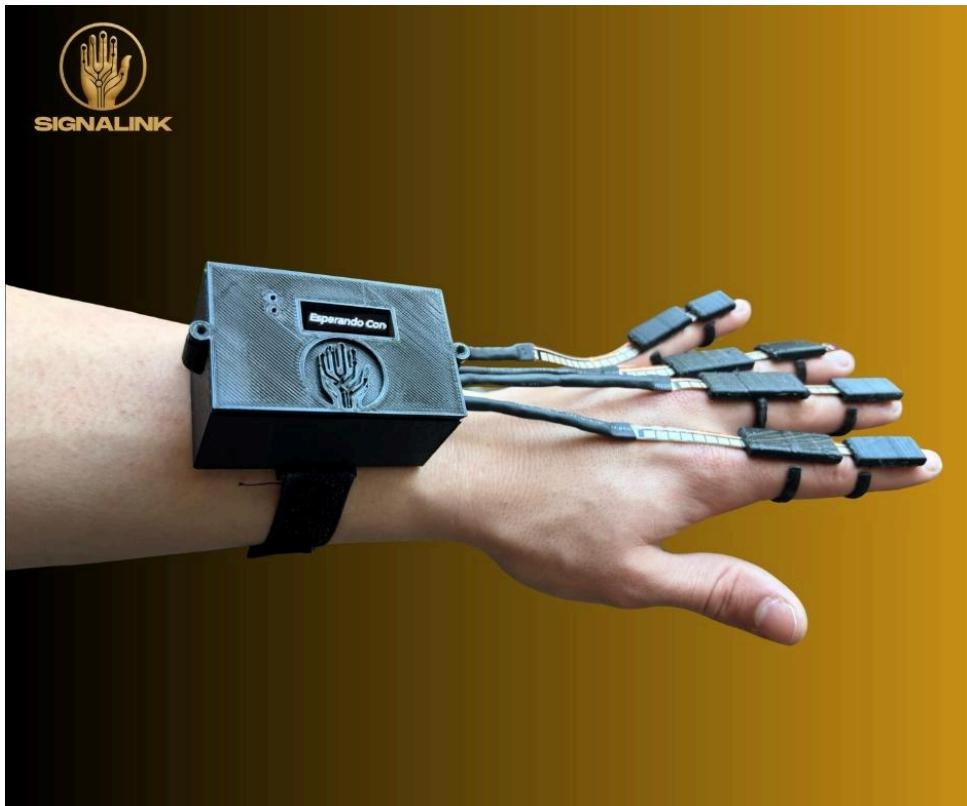


PROYECTO  
**SIGNALINK**



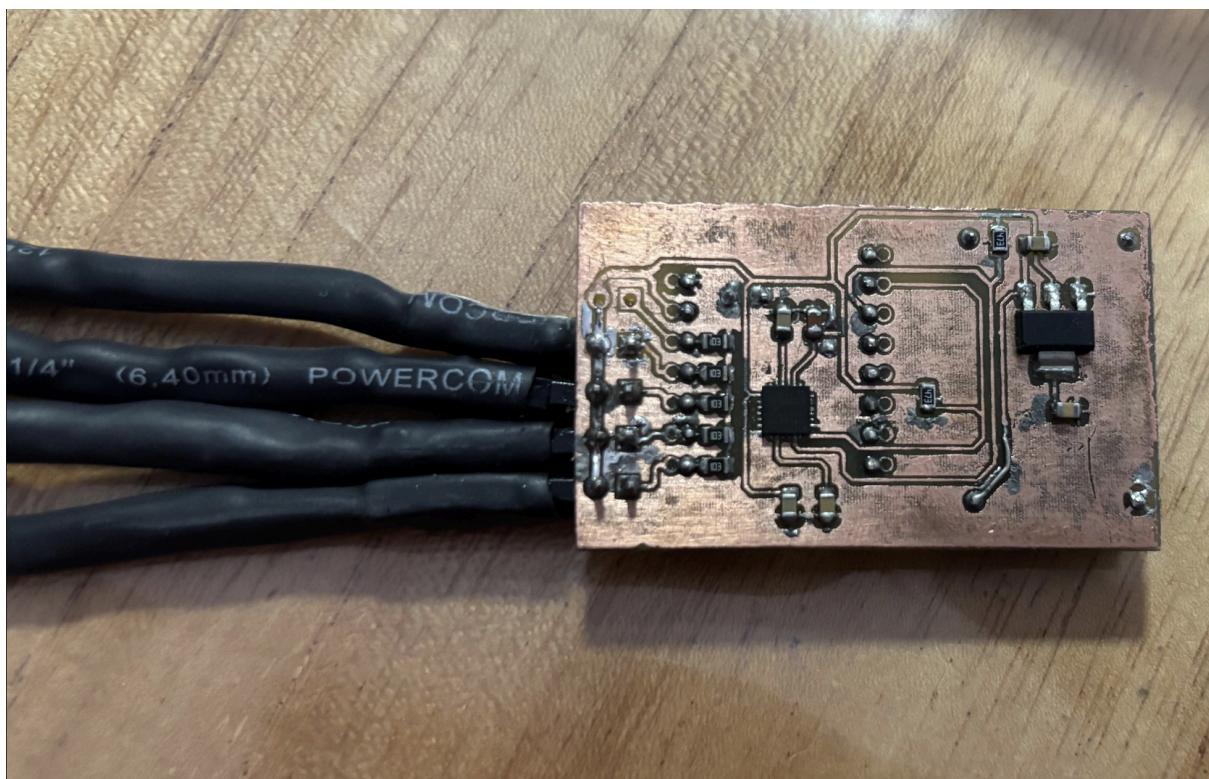
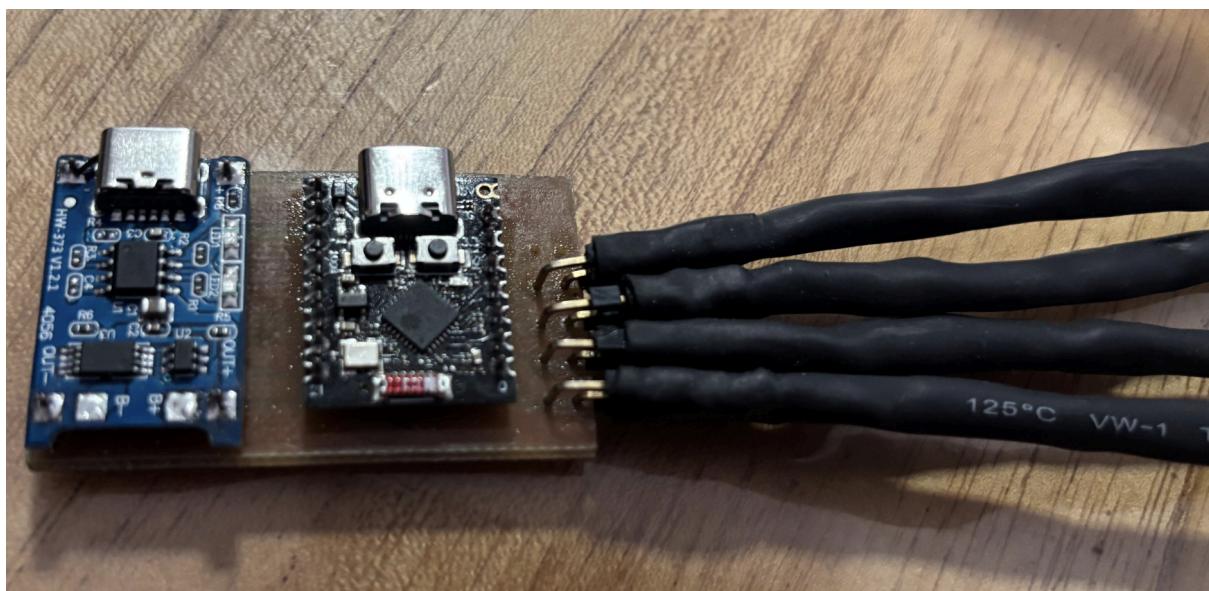


PROYECTO  
**SIGNALINK**





### 7.3. Placas





## Mejoras a futuro

### 8.1 Raspberry Pi Compute Module 4

Durante las últimas semanas, hemos estado trabajando en la migración e implementación de nuestra nueva adquisición, Raspberry Pi Compute Module 4 (CM4), con el objetivo de mejorar el rendimiento general y la capacidad de procesamiento de nuestro sistema. Este cambio fue motivado por las limitaciones que presentaba nuestra anterior unidad, la Raspberry Pi Zero 2W, la cual, si bien cumplió su función durante un período considerable, comenzó a mostrar restricciones significativas en cuanto a memoria RAM, capacidad de almacenamiento y velocidad de respuesta, especialmente al ejecutar tareas concurrentes o procesos más exigentes.

Como parte del proceso de transición, se realizó una reconstrucción completa de la estructura de carpetas y archivos que utilizamos en la RPi Zero 2 W, replicando y reorganizando los directorios en la nueva CM4 para asegurar compatibilidad y continuidad operativa. Esta tarea incluyó la migración de scripts, configuraciones de servicios, bases de datos locales y módulos de comunicación, todo adaptado a las nuevas capacidades del hardware.

La Raspberry Pi Compute Module 4 nos ofrece ventajas sustanciales, entre ellas:

- Mayor capacidad de RAM, lo que permite ejecutar procesos más pesados y mejorar la multitarea.
- Procesador más potente, que incrementa la velocidad de ejecución.

Además, se han realizado ajustes en el sistema operativo, optimizando el entorno para aprovechar al máximo los recursos disponibles, y se incorporaron medidas de monitoreo para evaluar el rendimiento en tiempo real.

Este avance representa un paso importante en la evolución de nuestra infraestructura, permitiéndonos escalar el proyecto con mayor estabilidad y eficiencia.

### 8.2 Placa PCB de Módulo Pecho

Otra de las tareas que aún tenemos en desarrollo, con una meta estimada para mediados de noviembre, es integrar la Raspberry Pi Compute Module 4 en una placa PCB junto con los demás componentes y puertos necesarios. De esta forma, buscaremos eliminar la dependencia de la IO Board y de fuentes de alimentación externas.



### 8.3 Aplicación Móvil

Como complemento del prototipo físico, se desarrollará una **aplicación móvil** destinada a la **configuración, calibración y personalización** del sistema. Esta herramienta permitirá que cada usuario pueda **ajustar los parámetros de detección de movimiento y flexión** de los sensores del guante de acuerdo con sus propios gestos.

El motivo principal de esta funcionalidad radica en que **no todas las personas realizan las señas de la misma manera**: existen variaciones naturales en la **amplitud de los movimientos, la velocidad de ejecución, la flexión de los dedos y la orientación de la mano**. Por ello, el sistema debe adaptarse a cada usuario para lograr un reconocimiento preciso y confiable.

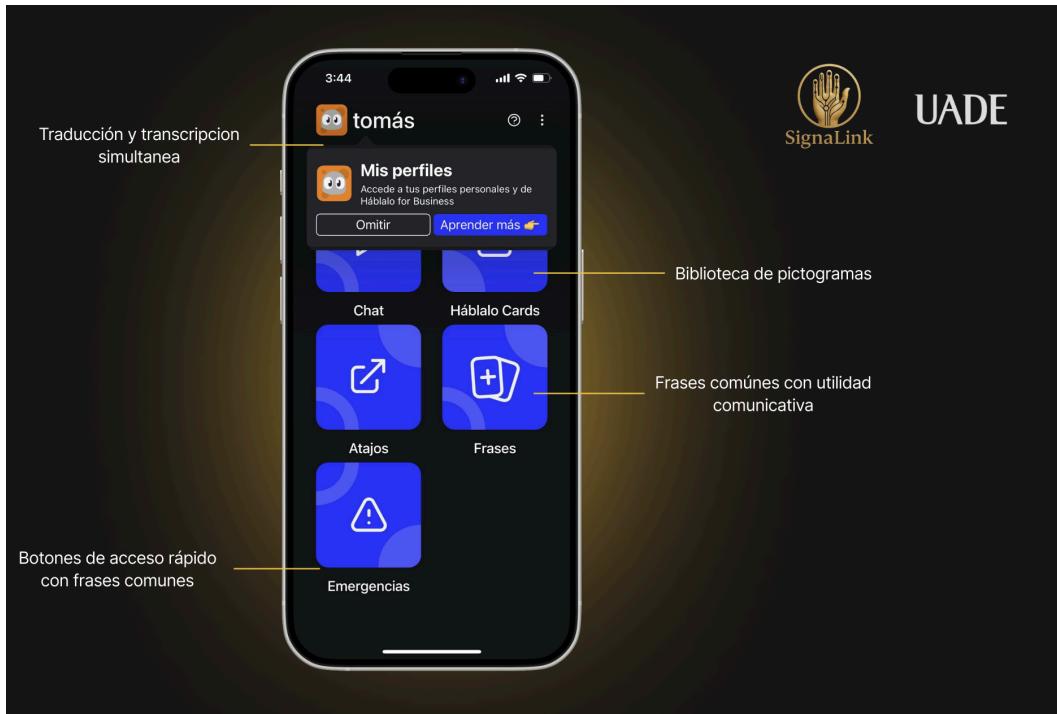
Durante la **etapa de calibración**, la aplicación guiará al usuario a realizar una serie de señas predefinidas mientras el prototipo registra los valores de referencia de los sensores **flex** y del **acelerómetro-giroscopio**. Estos datos se almacenarán en un perfil personalizado que permitirá al sistema interpretar correctamente los movimientos del usuario específico.

Además de la calibración, la aplicación podrá ofrecer diversas funcionalidades complementarias, tales como:

- **Gestión de perfiles de usuario**, permitiendo guardar configuraciones individuales.
- **Actualización de firmware o parámetros** del dispositivo mediante conexión **Bluetooth Low Energy (BLE)** o **Wi-Fi**.
- **Historial de uso y estadísticas** sobre las señas detectadas y su frecuencia.
- **Modo de entrenamiento**, que permitirá practicar señas y recibir retroalimentación visual o sonora en tiempo real.
- **Sincronización con la Raspberry Pi Compute Module 4**, para el envío de datos o el ajuste de modelos de reconocimiento.

De esta forma, la aplicación se integra como una **herramienta clave para la personalización, el mantenimiento y la mejora continua** del sistema, garantizando que el dispositivo SignaLink pueda adaptarse a las particularidades de cada usuario y entorno de uso.

Esta parte de SignaLink será desarrollada junto a alumnos pertenecientes a la facultad de ingeniería de UADE, quienes actualmente están cursando la carrera Licenciatura en gestión de tecnología de la información (Sistemas).



**Figura:** Diseño conceptual de aplicación *Háblalo*.



Desarrollado con **orgullo**  
por estudiantes  
comprometidos con la  
**innovación** y la **inclusión**

Gracias por tu **tiempo** y  
por ser **parte** del  
crecimiento de  
**SignaLink**