

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

**FACULTAD DE CIENCIAS E INGENIERÍA**



**Sistema de asistencia para personas cuádruplégicas que permite la  
manipulación de un mecanismo robótico mediante la voz**

**TESIS PARA OPTAR EL TÍTULO DE INGENIERO MECATRÓNICO**

**AUTOR**

Joel Eduardo Escalante Zelada

**ASESOR:**

Antonio Lincoln Angulo Salas

**Lima, Marzo, 2020**

## RESUMEN

En la actualidad, la cuadriplejia resulta ser uno de los padecimientos más difíciles de afrontar, al ser una discapacidad irreversible, la cual implica la inmovilización de miembros superiores e inferiores, ocasionando un alto impacto en la vida de un ser humano. El resultado de esta patología influye en la dependencia de una segunda persona, generándole sentimientos de inferioridad, pérdida de su intimidad y sobre todo aislamiento social.

Dada la existencia de esta patología, diversas instituciones han desarrollado soluciones que a la actualidad han demostrado que una persona con padecimiento de cuadriplejia puede tomar el control de diversos dispositivos usando medios alternativos, como el movimiento ocular, señales electroencefalográficas u órdenes verbales resultando atractivo para su desarrollo tecnológico.

El presente proyecto de investigación se centra en la creación de un sistema de asistencia robótica basado en el control por voz que, a diferencia de otros, tiene un óptimo funcionamiento sin la limitación de estar conectado a un ordenador y/o a internet, siendo un sistema fácilmente transportable y adaptable a cualquier entorno en el que se encuentre el discapacitado. Con la ayuda de este sistema, el usuario podrá tener el control de un mecanismo robótico, con el cual ejecutará acciones que normalmente son realizadas por una segunda persona. Adicionalmente, se presentará la simulación del sistema y sus resultados al evaluar su performance y su funcionalidad con respecto al ayudar a personas que padecen de cuadriplejia ya sea en la realización independiente de ciertas actividades de su vida cotidiana, como asearse o alimentarse.

Este proyecto sumará también a la labor de diversos especialistas enfocados en ayudar psicológicamente a las personas con padecimiento de cuadriplejia, cuya esencia se basa en mejorar su autoestima, su inserción en la sociedad y sustancialmente su calidad de vida y de los que lo rodean.



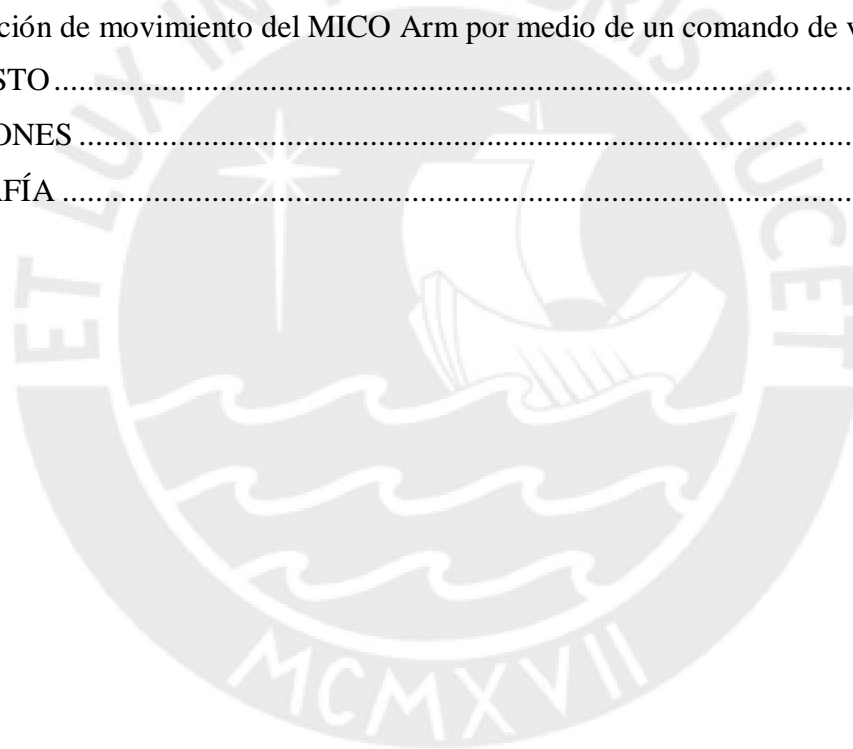


A mis tres amores Elizabeth López Cavero, Dylan Escalante López y Amy Escalante López por ser mi apoyo, mi alegría y motivación diaria. A mis padres Joel Escalante Rengifo y Carito Zelada Bazán, por su apoyo constante y ser mis guías durante toda mi etapa educativa. A mis hermanos Andy y Nicole por su confianza y admiración. A Luchita por ser mi ejemplo de esfuerzo e independencia. A los Sres. Cecilia Cavero y Benito López por su apoyo e inclusión dentro de su hermosa familia.

## TABLA DE CONTENIDO

	Pág.
RESUMEN.....	i
INDICE DE TABLAS.....	v
INDICE DE FIGURAS.....	vi
LISTA DE SÍMBOLOS.....	viii
INTRODUCCIÓN .....	1
ANTECEDENTES.....	2
1.1 Presentación de la problemática .....	2
1.3 Definiciones Conceptuales .....	4
1.3 Estado de la Tecnología .....	5
1.3.1 Publicaciones realizadas. ....	5
1.3.2 Tesis desarrolladas. ....	7
1.3.3 Productos en el mercado .....	8
1.4 Objetivos .....	9
1.4.1 Objetivo general .....	9
1.4.2 Objetivos específicos.....	9
DISEÑO CONCEPTUAL .....	11
2.1 Funciones de asistencia.....	11
2.2 Composición del sistema .....	12
2.2.1 Parámetros de entrada.....	13
2.2.2 Parámetros de salida .....	13
2.3 Requerimientos del sistema.....	15
2.4 Estructura de funciones .....	15
2.5 Matriz morfológica .....	17
2.6 Evaluación de conceptos.....	19
2.6.1 Conceptos de solución .....	19
2.6.2 Evaluación técnico-económica de los conceptos de solución .....	22
2.6.3 Concepto de solución óptimo.....	25
DISEÑO DEL SISTEMA DE ASISTENCIA POR RECONOCIMIENTO DE VOZ .....	27
3.1 Diseño electrónico .....	27
3.1.1 Diagrama de bloques del sistema.....	27
3.1.2 Componentes electrónicos .....	28
3.1.3 Diagramas esquemáticos del sistema de asistencia de control por voz.....	49

3.2 Diseño Mecánico .....	54
3.2.1 Material.....	54
3.2.2 Carcasas .....	55
3.3 Diseño de control.....	61
3.3.1 Control del reconocimiento de voz .....	62
3.3.2 Control del mecanismo robótico .....	65
SIMULACIONES .....	67
4.1 Requisitos.....	67
4.2 Simulación de la etapa de reconocimiento de voz.....	67
4.3 Simulación de la ejecución de movimientos del robot de asistencia .....	70
4.3.1 Configuración inicial del brazo robótico MICO.....	70
4.3.2 Control del robot vía USB .....	72
4.4 Simulación de movimiento del MICO Arm por medio de un comando de voz.....	74
PRESUPUESTO.....	75
CONCLUSIONES .....	77
BIBLIOGRAFÍA .....	78



## ÍNDICE DE TABLAS

Tabla 2.1: Actividades de asistencia .....	12
Tabla 2.2: Requerimientos del Sistema.....	15
Tabla 2.3: Matriz Morfológica del Sistema.....	17
Tabla 3.1: Requerimientos del módulo reconocedor de voz.....	29
Tabla 3.2: Módulos reconocedores de voz.....	30
Tabla 3.3: Características técnicas del módulo reconocedor de voz.....	30
Tabla 3.4: Requerimientos del Controlador para la parte de reconocimiento de voz.....	32
Tabla 3.5: Controladores para la parte de reconocimiento de voz.....	32
Tabla 3.6: Características técnicas del controlador para la parte de reconocimiento de voz..	33
Tabla 3.7: Comparación entre módulos de transmisión y recepción de datos .....	33
Tabla 3.8: Características técnicas de la pantalla LCD 16x02 .....	35
Tabla 3.9: Características técnicas del robot MICO Arm. ....	39
Tabla 3.10: Requerimientos del Controlador del mecanismo robótico.....	41
Tabla 3.11: Controlador del mecanismo robótico.....	41
Tabla 3.12: Características técnicas del PcDuino .....	42
Tabla 3.13: Características técnicas de la batería para la parte de reconocimiento de voz e interfaz de usuario .....	44
Tabla 3.14: Características técnicas del Salva LIPO .....	46
Tabla 3.15: Características técnicas del regulador LM7805 .....	47
Tabla 3.16: Características técnicas la batería de 12 V acido - plomo .....	48
Tabla 3.18: Propiedades del filamento ABS .....	55
Tabla 3.17: Relación de planos.....	61
Tabla 4.1: Comandos para el control del brazo robótico. ....	69
Tabla 5.1: Presupuesto de la parte mecánica .....	75
Tabla 5.2: Presupuesto de la parte electrónica.....	76
Tabla 5.3: Presupuesto del robot de asistencia .....	76

## ÍNDICE DE FIGURAS

Figura 1.1: Clasificación de nervios cervicales en la medula espinal.....	5
Figura 1.2: Tecnología Iris Bond .....	8
Figura 1.3: Tecnología Sip / Puff Switch .....	9
Figura 2.1: Entradas y Salidas del Sistema. ....	13
Figura 2.2: División del Sistema en Subsistemas. ....	14
Figura 2.3: Estructura de Funciones. ....	16
Figura 2.4: Concepto de Solución 1.....	20
Figura 2.5: Concepto de Solución 2.....	21
Figura 2.6: Concepto de Solución 3.....	22
Figura 2.7: Evaluación Técnico–Económica según VDI 2225. ....	25
Figura 2.8: Concepto de solución óptimo del sistema mecatrónico .....	26
Figura 3.1: Diagrama de bloques del sistema.....	28
Figura 3.2: Reconocimiento de voz. ....	29
Figura 3.3: Módulo reconocedor de voz V2 - Elechouse.....	30
Figura 3.4: Ubicación del módulo reconocedor de voz. ....	31
Figura 3.5: Controlador Arduino NANO. ....	32
Figura 3.6: Ubicación del controlador Arduino NANO.....	33
Figura 3.7: Módulo de transmisión inalámbrica NRF24L01. ....	34
Figura 3.8: Ubicación del módulo NRF24L01 .....	34
Figura 3.9: Interfaz de Usuario .....	35
Figura 3.10: Pantalla LCD 16x2. ....	35
Figura 3.11: Módulo de comunicación I2C para la pantalla LCD.....	36
Figura 3.12: Ubicación de la pantalla LCD 16x02 .....	36
Figura 3.13: Ubicación del módulo I2C.....	37
Figura 3.14: Mecanismo Robótico.....	37
Figura 3.15: Robot MICO Arm. ....	38
Figura 3.16: Izquierda: Brazo. Derecha: pinza de 2 dedos y pinza de 3 dedos. ....	38
Figura 3.17: Joystick del Robot MICO Arm. ....	39
Figura 3.18: Movimientos del Robot MICO Arm. ....	40
Figura 3.19: Controlador PcDuino V3. ....	42
Figura 3.20: Ubicación del controlador del mecanismo robótico.....	42
Figura 3.21: Rutina de posiciones para una persona con movilidad limitada. ....	43
Figura 3.22: Batería para la parte de reconocimiento de voz e interfaz de usuario. ....	44

Figura 3.23: Ubicación de la batería LIPO del sistema de reconocimiento de voz.....	45
Figura 3.24: Salva LIPO, medidor de carga para baterías LIPO con alarma.....	45
Figura 3.26: Regulador LM7805 .....	47
Figura 3.27: Ubicación del regulador de voltaje del reconocimiento de voz.....	47
Figura 3.28: Batería 24V - 40Ah .....	48
Figura 3.29: Regulador de voltaje 24V a 5V 3 A de la marca CPT .....	49
Figura 3.30: Ubicación del regulador de voltaje para el mecanismo robótico .....	49
Figura 3.31: Esquemático del controlador del reconocimiento de voz.....	50
Figura 3.32: Esquemático de la interfaz de usuario .....	51
Figura 3.33: Esquemático del módulo reconocedor de voz .....	51
Figura 3.34: Esquemático del circuito de alimentación del sistema de reconocimiento de voz .....	52
Figura 3.35: Esquemático del circuito de alimentación del mecanismo robótico .....	52
Figura 3.36: Esquemático de la etapa de control del mecanismo robótico .....	53
Figura 3.47: Filamento de ABS .....	54
Figura 3.37: Carcasa del sistema de reconocimiento de voz.....	55
Figura 3.38: Elementos que conforman la carcasa del sistema de reconocimiento de voz.....	56
Figura 3.39: Vista frontal del Elemento D .....	57
Figura 3.40: Correas de nylon .....	57
Figura 3.41: Sistema de sujección al espaldar de la silla .....	58
Figura 3.42: Carcasa del sistema de control del robot de asistencia.....	58
Figura 3.43: Elementos que conforman la carcasa del sistema de control del robot .....	59
Figura 3.44: Soporte para la carcasa del sistema de control del robot y para el robot MICO. 59	
Figura 3.45: Soporte para la carcasa del sistema de control del robot.....	60
Figura 3.46: Elementos que conforman la interfaz de usuario .....	60
Figura 3.47: soporte de la interfaz de usuario .....	61
Figura 3.48: Diagrama de flujo del controlador del reconocimiento de voz.....	63
Figura 3.49: Diagrama de flujo de control del módulo reconocedor de voz.....	64
Figura 3.50: Diagrama de flujo de control del mecanismo robótico .....	66
Figura 4.1: Conexiones para la simulación de la parte de reconocimiento de voz.....	68
Figura 4.2: Lógica de reconocimiento de comandos .....	68
Figura 4.3: Conexiones para la simulación del subsistema de reconocimiento de voz .....	70
Figura 4.4: Software JACOSOFT para el control del robot MICO.....	71

Figura 4.5: Ventana “Mapping” para la configuración de los comandos que representaran a los botones del joystick ..... 72

Figura 4.6: Configuraciones del modo “A1” y “A2”..... 72

Figura 4.7: Simulación del movimiento del robot MICO Arm mediante un ordenador ..... 73

Figura 4.8: Simulación del movimiento del robot MICO a través del reconocimiento de voz74

**LISTA DE SÍMBOLOS**

V	: Voltaje	[V]
VCC	: Voltaje Lógico (5V)	[V]
A	: Amperios	[A]
g	: Gramos	[g]
W	: Watts	[W]
m	: metros	[m]
bps	: Bits por segundo	
Pa	: Pascal	[Pa]
GND	: Ground (tierra)	
N	: Newton	[N]



## INTRODUCCIÓN

Hoy en día la discapacidad motora, ya sea por nacimiento, enfermedad, vejez o accidente, es un motivo de preocupación para el mundo entero, debido al impacto negativo que tiene en el desarrollo humano e interacción social de la persona que lo padece, llegando a afectarlo psicológica, física y hasta económicamente dada las limitaciones dentro del ámbito laboral. Este hecho resulta ser más complicado cuando la discapacidad está presente en la mayor parte del cuerpo, como es el caso de la “*cuadriplejia*”, la cual involucra la parálisis total de las cuatro extremidades.

El presente trabajo de investigación se centra en el desarrollo de una herramienta capaz de otorgar a la persona que padece de cuadriplejia, la libertad de poder manipular objetos dentro de su entorno mediante simples comandos de voz y con ello otorgarle cierto grado de independencia en algunas actividades de su vida diaria, como el asearse o alimentarse.

Los capítulos que comprende este documento son: capítulo primero titulado “*Antecedentes*”, en el cual se presenta la situación actual de esta patología en el Perú y el mundo, tratando algunos conceptos que nos ayuden a entender las causas y sus limitaciones, seguido de una exposición breve de las investigaciones que se han elaborado para hacer frente a este padecimiento, y cerrando el capítulo con la definición de los objetivos de este trabajo de investigación. Capítulo segundo titulado “*Diseño conceptual*”, donde se presentará las funciones y composición del sistema mecatrónico, seguido del desarrollo de la matriz morfológica de Zwicky para una óptima elección de componentes y la elección del mejor concepto de solución. Capítulo tercero titulado “*Diseño del sistema de asistencia por reconocimiento de voz*” donde se expondrá la función y características técnicas de cada componente seleccionado para la parte electrónica, mecánica y de control del sistema de asistencia. Capítulo cuarto titulado “*Simulaciones*”, en el cual se expondrá el desarrollo experimental del sistema, validando su performance y funcionalidad en la simulación de una actividad de asistencia. Por último, se comentará el presupuesto para la implementación del sistema y las conclusiones del tema de tesis.

## **CAPÍTULO I**

### **ANTECEDENTES**

En el presente capítulo se expondrá brevemente la problemática que presenta la discapacidad motora en el Perú y el mundo, dando énfasis a la cuadriplejía, tratando conceptos claves para entender sus causas y limitaciones, resumiendo algunas investigaciones para hacer frente a este padecimiento y comentando el objetivo en el que se enfoca el presente proyecto de investigación.

#### **1.1 Presentación de la problemática**

Según la Encuesta Mundial de Salud y el informe de Carga Mundial de Morbilidad (2010), descritas en el Informe Mundial sobre la Discapacidad (Organización Mundial de la Salud, 2011), se estimó que un 15% de la población mundial (1000 millones de personas) padecían de algún tipo de discapacidad para ese entonces. De este número, el 2,2% (190 millones de personas en el mundo) presentaban algún tipo de discapacidad grave, es decir, tenían la mayor parte del cuerpo paralizada o un padecimiento de carácter mortal al corto y mediano plazo.

Estos datos estadísticos dan una proyección estimada de cuantas personas con discapacidad grave existen hoy en el mundo, personas que necesitan de la ayuda de investigadores y especialistas para el desarrollo de tecnologías de asistencia que les permita tener una vida normal dentro de la sociedad.

El mismo informe estableció que en América Latina existían alrededor de 85 millones de personas con algún tipo de discapacidad, siendo las más graves la paraplejía y la cuadriplejía. Debido a esta problemática aparecieron investigaciones de suma importancia, como la

desarrollada por el doctor Gustavo Moviglia del Centro de Investigación en Ingeniería de Tejidos y Terapias Celulares (Doc Salud), de la Universidad Maimónides de Buenos Aires, cuyos resultados fueron la recuperación funcional de algunos miembros a través de las células madre.

El Perú no es ajeno estos padecimientos, según el Informe Nacional “Perfil Sociodemográfico del Perú” (INEI, 2018), elaborado por el Instituto Nacional de Estadística e Informática, revela que el país cuenta con una población aproximada de 31 millones 237 mil habitantes, de los cuales el 10.4% padecen de algún tipo de discapacidad, ya sea de origen genético, lesión cerebral o trauma físico, lo cual complica su inserción en la sociedad. Entrando en más detalle, de la porción de personas que presentan discapacidad, el 15.1% posee dificultad para moverse y/o caminar, cantidad representada por padecimientos como esclerosis múltiple, espina bífida, parálisis cerebral, distrofia muscular, osteogerósia imperfecta (conocida como “huesos de cristal”), paraplejia, hemiplejia y cuadriplejia. Dentro del grupo de padecimientos presentados, la cuadriplejia (también conocida como tetraplejia) resulta ser la más difícil de afrontar, ya que la persona que lo padece presenta inmovilización de los miembros superiores e inferiores, requiriendo la ayuda de una segunda persona para su traslado, alimentación, aseo y otras actividades.

Esta patología es irreversible y su impacto en el discapacitado es alto, porque influye en su calidad de vida, adaptación psicológica, inclusión social e independencia; la mayoría de cuadripléjicos se ven obligados al uso de una silla de ruedas y se ven obligados a la dependencia de otra persona para desarrollar sus actividades cotidianas, generando sentimientos de inferioridad, discriminación, miedo, angustia, depresión, aislamiento social, pérdida de su intimidad y la disminución de expectativas de su futuro.

Si bien hoy en día existen diversos dispositivos robóticos que brindan asistencia a personas discapacitadas con el fin de realizar diversas actividades a pesar de su limitación física, no todas son fácilmente manipulables. El caso se complica aún más cuando se trata de una persona que padece de cuadriplejia, ya que esta está limitada a no tener participación física de alguna parte de sus extremidades.

Por ello, el presente proyecto de investigación propone crear un sistema de control por voz, debido a que el padecimiento de cuadriplejia no afecta al habla en su totalidad a menos de tratarse de una lesión cerebral, y puede ser usado como un medio de control, que ayude a las

personas con cuadriplejia a manipular un robot que lo asista, y esta pueda alimentarse, asearse o realizar otra actividad de una manera independiente, contribuyendo en mejorar sustancialmente su calidad de vida y de las personas que participan en su cuidado, situación que repercutirá en mejorar su inserción en la sociedad.

### 1.3 Definiciones Conceptuales

Integrantes de la Asociación Americana en Lesión Medular (Kirshblum SC, Burns SP, Biering-Sorensen F, Donovan W, Graves DE, Jha A, Johansen M, Jones L, Krassioukov A, Mulcahey MJ, Schmidt-Read M, Waring W, 2011), definen a la cuadriplejia como una discapacidad o pérdida de la función motora y/o sensorial en los segmentos cervicales de la medula espinal por daño a elementos neurales dentro del canal cervical; ésta origina trastorno de la función de los brazos, así como el tronco, piernas y órganos pélvicos, es decir, incluye las cuatro extremidades.

Por otro lado, el artículo titulado “Understanding Spinal Cord Injury” (Shepherd Center, 2011) clasifica las causas de este tipo de lesiones medulares en:

- 1) Lesiones Traumáticas: Contusión total o parcial de la médula espinal por actividades de riesgo ya sea accidentes de auto, futbol americano, gimnasia, caída, intervención quirúrgica, entre otros.
- 2) Lesiones no traumáticas: Causadas por Enfermedades congénita, enfermedades inflamatorias, tumores, artritis, entre otros.

Siendo la lesión medular de causa traumática la de más frecuencia,

Asimismo, menciona que “las vértebras se agrupan en secciones (Ver figura 1.1), cuanto mayor sea la sección en la medula espinal, más disfunción puede ocurrir” (Shepherd Center, 2011), clasificándolos por:

- Nervios cervicales superiores, de las vértebras C1 a la C4.
- Nervios cervicales inferiores, de las vértebras C5 a la C8.
- Nervios torácicos de la T1 a la T12
- Nervios Lumbares de la L1 a la L5
- Nervios sacros de la S1 a la S5

Donde las lesiones ubicadas más arriba de la medula son las más graves y a veces fatales. La cuadriplejía resulta de una lesión en la zona cervical.



Figura 1.1: Clasificación de nervios cervicales en la médula espinal  
Fuente: Understanding Spinal Cord Injury, Shepherd Center.

### 1.3 Estado de la Tecnología

Especialistas en lesiones medulares, han visto a la robótica como un aliado fundamental en la rehabilitación física y asistencial de personas con discapacidad, es por ello que existen investigaciones y creaciones tecnológicas que en la actualidad se desarrollan orientadas a brindar una ayuda funcional en la vida de un cuadripléjico.

#### 1.3.1 Publicaciones realizadas.

Dentro del ámbito de las publicaciones académicas se encuentran investigaciones como:

Las realizadas por S. Udhaya kumar y Vibin Mammen Vinod titulado “EOG Based Wheelchair Control for Quadriplegics” (S. Udhaya kumar and V. M. Vinod, 2015), de la Universidad de Ingeniería de Kongu - Estado de Tamil Nadu - La India y publicado en la 2ª Conferencia Internacional sobre Innovaciones en la Información Incrustada y Sistemas de Comunicación, realizado en la ciudad de Coimbatore - La India. La investigación se basa en el diseño de un sistema de control de una silla de rueda para uso de personas cuadripléjicas, basado en señales eléctricas generadas por el movimiento de los ojos; este proceso es conocido como Electrooculograma (EOG), que consiste en colocar electrodos de superficie en los músculos faciales cercano a cada ojo con el fin de poder medir cambios de potencial entre la córnea y la retina. Se obtuvieron resultados positivos, logrando la simulación del control del movimiento en 3 direcciones (izquierda, derecha y arriba) de un vehículo electrónico.



Así mismo, la publicación titulada “Facilitating Communication and Computer Use with EEG Devices for Non-Vocal Quadriplegics” (G. Williams, Y. S. Lee, S. Ekanayake, P. N. Pathirana and L. Andriske, 2014), de la Escuela de Ingeniería de la Universidad de Deakin – Australia y publicado en la 7ª Conferencia Internacional sobre Información y Automatización para la sostenibilidad, realizado en la ciudad de Colombo - Sri Lanka. La investigación se centra en la creación de un sistema de comunicación para personas con limitaciones en el habla, debido a padecimientos como la cuadriplejía por parálisis cerebral, haciendo uso del dispositivo Emotiv, el cual es dispositivo que mide las señales cerebrales como ordenes mentales, expresiones faciales o métricas de rendimiento cerebral; este fue usado, no con su fin principal el cual es medir señales EEG (electroencefalografía), sino para detectar el ruido producido por señales de los movimientos faciales o movimientos oculares denominados señales EMG (Electromiografía) y EOG (electrooculografía) respectivamente, y por medio de ello controlar un software que reproduce sonidos ayudando en la comunicación del discapacitado. Los resultados fueron positivos respecto a la implementación del proyecto en un paciente con cuadriplejía, el único inconveniente fue la lentitud en el manejo del sistema por parte del paciente al momento de entablar una conversación.

Por otro lado, se encuentra la publicación realizada por Yaya Lu y Ying Chen, titulado “Prototyping potential control systems to assist complete quadriplegics” (Yaya Lu and Ying Chen, 2012), de la Universidad de Tasmania – Australia y publicado en Conferencia Internacional de Ingeniería Biomédica del 2012, realizado en la provincia de Ubon Ratchathani – Tailandia. La investigación se basa en el control de una silla de ruedas a través de movimientos faciales y por reconocimiento de voz haciendo uso de dispositivos de la marca LEGO Mindstorms NXT. El método usado en el reconocimiento de voz se basó en la fusión entre el código Morse y el sistema telegráfico chino creando una serie de comando de voz pro medio de sonidos como “Dits” y “Dahs”. Este prototipo obtuvo resultados exitosos, siendo efectivo todo el control de la silla mediante el movimiento facial y también un eficaz control mediante el reconocimiento de voz al tener un lenguaje universal sin uso de codificadores de lenguaje y/o traductores que impliquen un algoritmo de entrenamiento.

Por último, la publicación realizada por A. Jardón, J. C. González, M. Stoelen, S. Martínez y C. Balaguer, titulado “ASIBOT assistive robot in a domestic environment” (A. Jardón, J. C. González, M. Stoelen, S. Martínez, C. Balaguer, 2009) de la Universidad Carlos III de Madrid – España y publicada en la 2ª Conferencia Internacional sobre Tecnologías

relacionados a la Asistencia en Entornos, realizada en Corfú – Grecia. El artículo relata la creación de un brazo robótico para la asistencia de personas discapacitadas, con funcionalidad en realizar actividades dentro de una cocina, como el abrir cajones, servir agua en una taza desde el caño, sostener utensilios, llevar cosas al fregadero, entre otros. Su control se efectúa a través de un joystick logrando el movimiento a lo largo del ambiente de la cocina por medio de una plataforma móvil.

### **1.3.2 Tesis desarrolladas.**

En indagaciones permanentes que se han realizado en nuestro medio para desarrollar diseños que coadyuven a las personas discapacitadas, especialmente cuadripléjicas, a tener autonomía en sus actividades, se ha podido recoger los siguientes antecedentes:

La tesis titulada “Sistema domótico controlado por Voz para personas con discapacidad en extremidades superiores, utilizando tarjeta Raspberry Pi” (Tomala Cuenca, 2018) de la Universidad Politécnica Salesiana – Guayaquil, quien desarrolló un sistema de control domótico para personas discapacitadas el cual se basa en el control por voz mediante la plataforma de Raspberry Pi. Este controlador conectado inalámbricamente a diversos actuadores dentro de una casa puede asistir a la persona en la apertura de puertas, encendido de luces y activación de equipos electrónicos, todo mediante comandos de voz, hecho realizado gracias al software VoiceAttack y conexión a internet, su implementación fue satisfactoria cumpliendo el papel de otorgar cierta independencia a una persona discapacitada en un ambiente delimitado.

Asimismo, la tesis titulada “Diseño de una silla de ruedas inteligente con integración multisensorial” (Mauricio Villalba, 2008) de la Universidad Nacional de Huancavelica, quien plantea un prototipo de diseño de una silla de ruedas controlada a través de órdenes verbales, gracias a librerías del software Dragon Naturally Speaking, y guiado autónomo mediante sensores infrarrojo y ultrasonido. Esta investigación concluyó en la obtención de una óptima detección de obstáculos en el ambiente y una falla menor al 0.5% en el desplazamiento mediante órdenes verbales, siendo un proyecto viable y de bajos recursos económicos.

Por último, la tesis titulada “Diseño de un sistema de control de desplazamiento de una silla de ruedas basado en el procesador de voz SPCE061A para el Centro de Educación y Rehabilitación Sicomotriz” (Ocaña Velásquez, 2013) de la Universidad de Privada Antenor Orrego – Chimbote; el proyecto se basó en un sistema de reconocimiento de voz para asistir a



personas discapacitadas que necesiten trasladarse a través de una silla de ruedas y lograr su total dependencia. El investigador diseñó un sistema de control de desplazamiento de una silla de ruedas, donde el discapacitado puede comandar el vehículo teniendo como opción dos tipos de operación; el primero a través del uso de un joystick y el segundo controlado por órdenes de voz. Los resultados de las pruebas, dieron un funcionamiento con una efectividad de voz del 89 %, debido a que el prototipo también captó el ruido del ambiente y un tiempo de funcionamiento de aproximadamente 3 horas.

### 1.3.3 Productos en el mercado

Dentro del ámbito de productos comerciales se encuentran dispositivos como:

#### *a) Iris Bond (Jauregui & Iosu, 2013)*

La empresa Iris Bond a través de Vicomtech-IK4, crea un dispositivo cuyo funcionamiento se basa en la videoculografía, el cual consiste en controlar un ordenador con el movimiento de los ojos, orientado para la asistencia de pacientes que sufren de movilidad reducida. Este producto permite al discapacitado poder hacer uso del ordenador con total dependencia, dándole la posibilidad de poder abrir correos hasta poder buscar información en la web, por otro lado, permite a discapacitados con limitaciones en el habla poder comunicarse y expresar sus emociones mediante un software especializado.

El dispositivo proyecta una luz infrarroja inofensiva hacia el ojo humano con el fin de generar destellos en la pupila, estos destellos son registrados por una videocámara y luego mediante un algoritmo avanzado se procesa e identifica la posición de las pupilas, esto permite determinar las coordenadas de la posición del ratón. Así es como el usuario puede controlar del ratón del computador con el movimiento de los ojos.



Figura 1.2: Tecnología Iris Bond  
Fuente: [www.zeroproject.org](http://www.zeroproject.org)

### ***b) Sip/ Puff Switch (Instruments)***

Es un gadget creado por la empresa Origin Instruments, con la finalidad de asistir a personas con discapacidad motora. Su funcionalidad se basa en la generación de soplos por parte del usuario a través de un tubo de diámetro pequeño conectado a la boca, este dispositivo los reconocerá y los convertirá en señales de encendido y apagado. Esta tecnología se puede aplicar a una variedad de propósitos, como el dirigir a una silla de ruedas o navegar en la web.



Figura 1.3: Tecnología Sip / Puff Switch  
Fuente: [www.liberator.co.uk](http://www.liberator.co.uk)

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

Desarrollar un sistema de control por voz que permita a personas cuadripléjicas manipular un robot para asistencia personal.

### **1.4.2 Objetivos específicos**

- Desarrollar un sistema de reconocimiento y procesamiento de señales para detección y clasificación de sonidos vocales.
- Implementar un interfaz humano-máquina controlado por voz. Este debe ser intuitivo y configurable por el usuario.
- Seleccionar, adaptar y programar un mecanismo robótico para asistir a personas con discapacidad.
- Simular el sistema de asistencia para personas cuadripléjicas y evaluar su funcionalidad.

### 1.5 Alcances y limitaciones de la Tesis

El presente documento de tesis desarrolla un sistema de control por voz el cual procesa la señal sonora que emite la voz del discapacitado y la traduce en una acción a ejecutarse, por ello, el presente proyecto será desarrollado en un ambiente controlado, es decir poca interferencia sonora, y con la mínima distancia entre la boca del usuario y el micrófono del sistema, permitiéndole al equipo ser más eficiente en el reconocimiento de comandos de voz. Cabe resaltar que esta eficiencia probablemente se pueda ver afectada si existe una gran interferencia de ruido del exterior del ambiente o que no provenga de la voz del usuario.

Por otro lado, el tema de tesis desarrolla el control de un brazo Robótico, pero este excede del presupuesto para su adquisición, por ello las pruebas de funcionalidad se realizarán a disponibilidad del equipo, el cual se usará en condición de préstamo por la Universidad Pontificia Católica del Perú. Pero gracias a su desarrollo, este sistema deja abierta la posibilidad de ser adaptado como control de cualquier otro actuador por medio de una comunicación inalámbrica, solo que para este caso se centrará en el brazo Robótico “MICO”.

En adición, el sistema de control por voz fue diseñado para ser adaptable a cualquier silla de ruedas que funcione a baterías con alimentación de 24 voltios o disponga de espacio para adaptarlas. Pero también deja abierta la posibilidad de adaptar el equipo a cualquier ambiente que se encuentre el discapacitado.

## CAPÍTULO II

### DISEÑO CONCEPTUAL

El presente capítulo iniciará con una breve descripción de las funciones que deberá realizar el sistema para poder cumplir con el papel de asistir a una persona con padecimiento de cuadriplejía. Seguidamente, se desarrollará la composición del sistema el cual consta de cuatro partes: reconocimiento de voz, interfaz de usuario, alimentación y mecanismo robótico.

Luego, según la metodología de diseño basada en la norma VDI 2225 se determinará el concepto de solución óptimo para poder resolver el problema. Para ello se desarrollará la estructura de funciones que realizará cada parte del sistema, después mediante la matriz morfológica de Zwicky se obtendrán posibles alternativas de solución para la composición del sistema, finalmente las alternativas desarrolladas se someterán a una evaluación técnica-económica cuyo resultado determinará que opción es más viable.

#### **2.1 Funciones de asistencia**

El sistema de control por voz busca brindar a la persona cuadripléjica una ayuda en la ejecución independiente de la mayoría de sus actividades diarias, por ello para poder cumplir con el papel de asistir al discapacitado, éste deberá participar como mínimo en algunas actividades esenciales del día a día, las cuales requieren de la intervención de una segunda persona. Según el “Manual de cuidados en lesión medular” (de Figueiredo Carvalho, Tirado Darder, Mulet Falco, Nuñez Hernandez, & Mulet), la persona que asiste al discapacitado debe ser participe en el cuidado del sistema respiratorio, eliminaciones intestinales, movilización, alimentación e hidratación. Por ello, el presente sistema se enfocará en actividades que no requieran de conocimientos de medicina para su cuidado, como lo son la alimentación, hidratación y adicionalmente la movilización de objetos en su entorno. Algunas de estas

actividades en las que participará el sistema de control por voz se presentan resumidas en la tabla 2.1.

Tabla 2.1: Actividades de asistencia

Funciones	Actividades	
<b>Relacionadas con el usuario</b>	Limpiar la cara	
	Cepillar los dientes	
	Afeitarse la cara con rasuradora eléctrica	
	Comer con cuchara / alimento sólido	
	Beber de un vaso / botella / taza	
<b>Relacionadas con objetos libres del entorno</b>	Coger y poner objetos en:	La mesa
		El piso
		Cajones
	Verter líquido de una jarra en:	Un vaso
	Una taza	
<b>Relacionado con objetos instalados en el entorno</b>	Llenar la cuchara con comida	
	Abrir y cerrar	Puertas
		Cajones
		Ventanas
		Caño del lavadero
	Prender y apagar	Luces
Televisor		

Fuente: Elaboración Propia

Mientras más actividades de asistencia cubra el sistema más beneficioso se vuelve para el discapacitado, en el punto de volverlo más independiente reduciendo por consiguiente la carga que representa para su cuidador o persona que lo asiste.

## 2.2 Composición del sistema

Para poder definir la composición del sistema de asistencia de control por voz, se comenzará a explicar desde un punto de vista global entrando poco a poco a detalle con el transcurso del documento.

Para empezar, se representa el sistema por una “caja negra”, como se puede apreciar en la figura 2.1, en ella se detallan las entradas y salidas requeridas para el buen funcionamiento del sistema y una adecuada interacción con el usuario.



Figura 2.1: Entradas y Salidas del Sistema.  
Fuente: Elaboración Propia

Como es de conocimiento, el sistema ejecutará una acción dada una orden de voz, por ello, como entrada principal está la voz del usuario y como una salida la acción del robot de asistencia. Seguidamente se procede a definir cada parámetro de entrada y de salida que se ha representado en la imagen.

### 2.2.1 Parámetros de entrada

- **Comando de voz:** Sonido vocal que emite el discapacitado para ordenar al sistema que ejecute una acción.
- **Energía:** Representa la fuente de energía eléctrica necesaria para que el sistema esté en funcionamiento.
- **Configuración:** Representa la indicación por parte del usuario para que el sistema de reconocimiento de voz entre en modo de configuración. El modo configuración es cuando el sistema deja de recibir instrucciones para pedir al usuario la grabación de todos los comandos de voz necesarios para el control del sistema, caso contrario el sistema procesa los comandos recibidos y ejecuta una acción mediante el mecanismo robótico.

### 2.2.2 Parámetros de salida

- **Nivel de energía:** El sistema alertará al usuario en caso de escasas de energía. Esta alerta se presenciara en la interfaz de usuario.
- **Ejecución de la acción:** Es la acción resultante que ejecutará el sistema ante una orden de voz.
- **Visualización del comando detectado:** El sistema emitirá un mensaje visual en la interfaz de usuario cada vez que detecte una orden de voz, dando a conocer si



es una orden válida o no, asimismo también dará a conocer el tipo de orden que llegó a reconocer el sistema.

Seguidamente se procede a definir la composición del sistema de asistencia de control por voz que representa la “caja negra”, para ello el sistema se dividirá en cuatro partes que lo componen: reconocimiento de voz, interfaz de usuario, distribución de energía y mecanismo robótico; quienes en su interacción conjunta permitirán ejercer con eficacia la función de asistencia hacia el discapacitado. La figura 2.2 ilustra una representación del sistema y su división, adicionalmente se detalla cómo se efectúa la interacción entre cada parte.

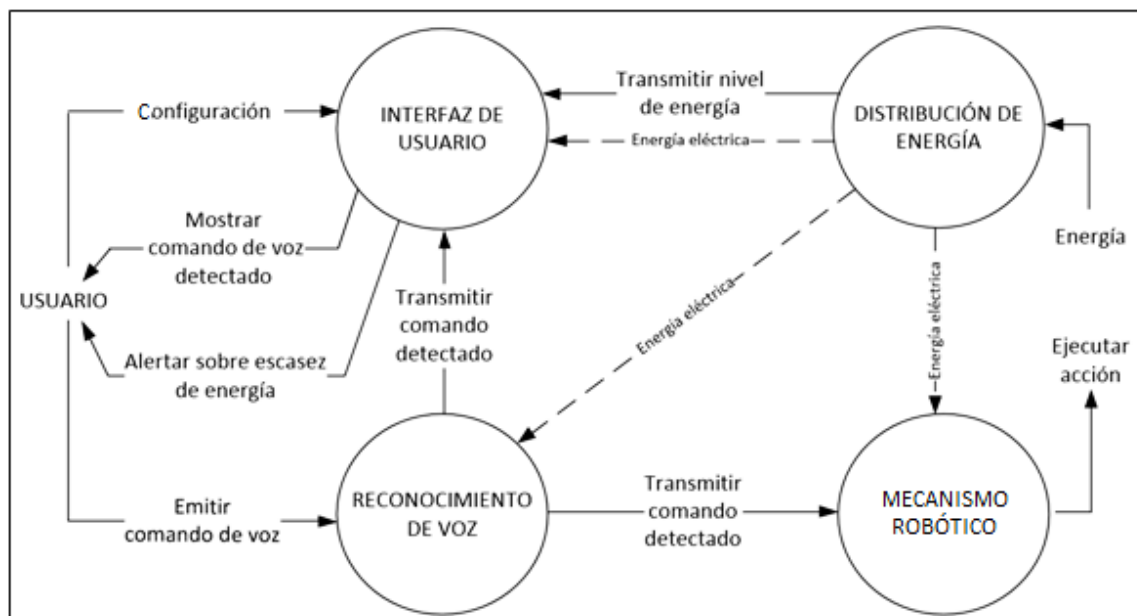


Figura 2.2: División del Sistema en Subsistemas.  
Fuente: Elaboración Propia

La parte denominada “Interfaz de Usuario” será la encargada de interactuar con el discapacitado, dando a conocer el comando detectado y alertar sobre un nivel bajo de energía, asimismo será el medio por el cual se podrá configurar y establecer nuevos comandos de voz.

La parte denominada “Distribución de Energía” será la encargada de abastecer de energía a todo el sistema en general, este estará compuesto por una fuente de alimentación y diversos módulos de regulación. Adicionalmente tendrá un circuito capaz de detectar un bajo nivel de energía en la batería, lo cual se alertará a través de la “Interfaz de Usuario”.

Por otro lado, la parte denominada “Mecanismo Robótico”, será quien tendrá la labor fundamental de ejecutar el trabajo mecánico, asistiendo en algunas de las actividades nombradas en la tabla 2.1, según lo ordene el usuario.



Por último y no menos importante, la parte denominada “Reconocimiento de Voz”, la cual deberá abarcar todo el proceso desde la adquisición de señal de voz hasta la obtención de un comando relacionado con la acción a ejecutar. Tendrá como entrada la voz del usuario y generará como salida el comando detectado, este se visualizará en la “Interfaz de Usuario” y será procesado por el “Mecanismo Robótico” ejecutando la acción correspondiente.

### 2.3 Requerimientos del sistema

En la tabla 2.2 se resume los requerimientos que deberá cumplir el sistema de asistencia de control por voz, siendo necesarios para un funcionamiento eficiente en la modalidad de asistir a personas con padecimiento de cuádruplejía.

Tabla 2.2: Requerimientos del Sistema.

	Interfaz de Usuario	Reconocimiento de voz	Distribución de Energía	Mecanismo Robótico
Requerimientos Mecánicos	<ul style="list-style-type: none"> <li>• Debe poseer una estructura que soporte al conjunto de dispositivos y este ubicado cerca a la vista del discapacitado.</li> <li>• Debe poseer una carcasa de protección a los circuitos.</li> </ul>	<ul style="list-style-type: none"> <li>• Debe poseer una carcasa de protección a los circuitos, ante golpes y partículas de polvo.</li> <li>• El micrófono debe ubicarse cerca a la boca del discapacitado, para evitar ruido externo.</li> <li>• Debe ser flexible en su instalación para comodidad del usuario.</li> </ul>	<ul style="list-style-type: none"> <li>• Debe poseer una carcasa para evitar contacto con líquidos y humedad del ambiente.</li> </ul>	<ul style="list-style-type: none"> <li>• La estructura del robot debe ser portátil y flexible en su instalación.</li> </ul>
Requerimientos Eléctrico-Electrónicos	<ul style="list-style-type: none"> <li>• Debe poseer una pantalla para la visualización y botones para su configuración.</li> </ul>	<ul style="list-style-type: none"> <li>• Debe procesar y almacenar al menos 10 comandos de voz.</li> <li>• Debe mostrar su estado de funcionamiento (modo configuración o modo trabajo).</li> </ul>	<ul style="list-style-type: none"> <li>• Debe tener una etapa de regulación de la energía, con el fin de poder alimentar a cada parte según el consumo de sus componentes.</li> <li>• Debe estar en la capacidad de energizar al sistema de 6 a 10 horas por día.</li> </ul>	<ul style="list-style-type: none"> <li>• Debe poseer un consumo bajo de energía.</li> <li>• Preferiblemente debe poseer un control inalámbrico para evitar enredos con el entorno.</li> </ul>
Requerimientos de control		<ul style="list-style-type: none"> <li>• Debe poseer una alta velocidad de procesamiento en la identificación de comandos de voz</li> </ul>	<ul style="list-style-type: none"> <li>• Debe alertar al usuario ante una escasez de alimentación energética.</li> </ul>	<ul style="list-style-type: none"> <li>• Poseer una velocidad de procesamiento alta para evitar retardos en la ejecución de movimientos del robot.</li> </ul>

Fuente: Elaboración Propia

### 2.4 Estructura de funciones

Dividido el sistema en cuatro subsistemas se procede a ahondar en más a detalle las funciones que realizan cada uno, la figura 2.3 ilustra cada parte estructurado en bloques de funciones y su interacción conjunta dentro del sistema.

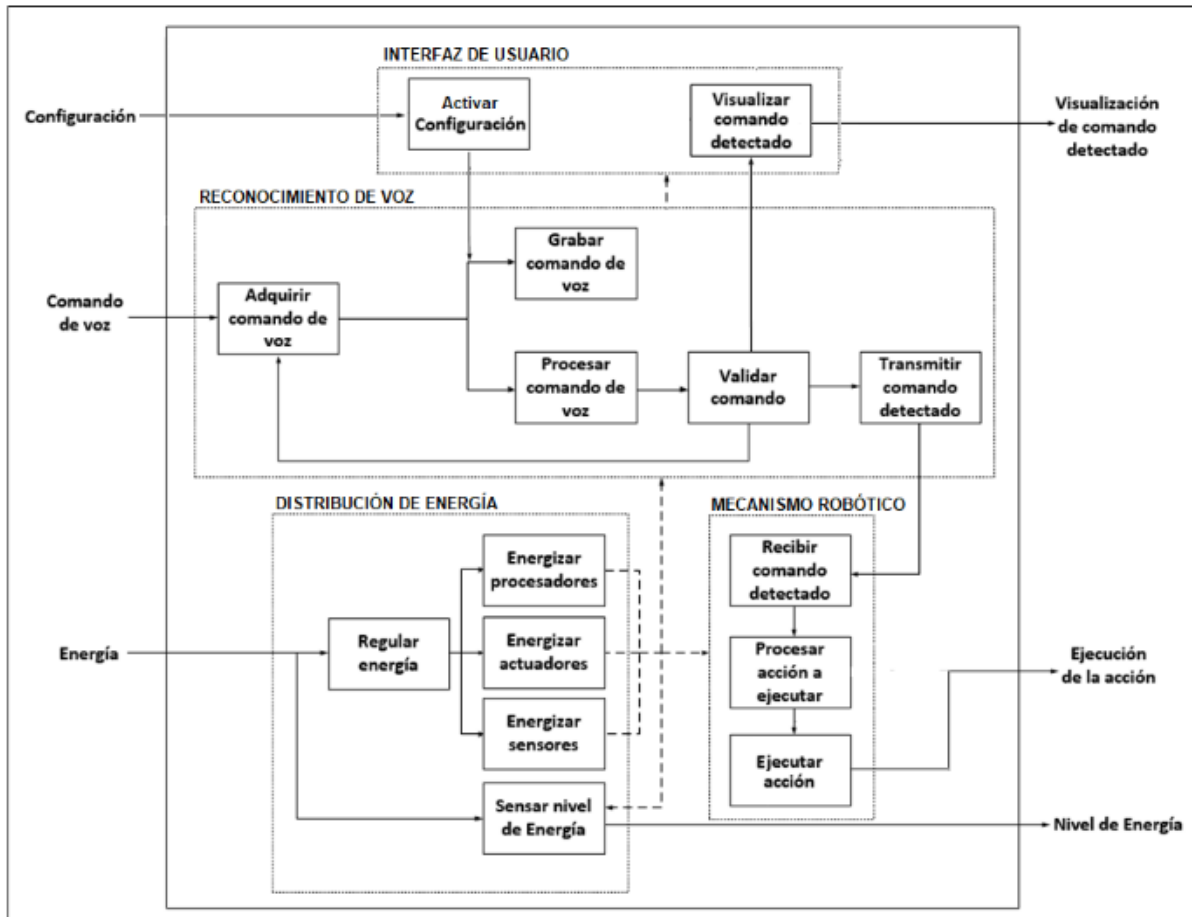


Figura 2.3: Estructura de Funciones.  
Fuente: Elaboración Propia

La parte de “Reconocimiento de Voz” iniciará detectando el modo en el que se encuentra el sistema, ya que, si está en modo de configuración, se procede con la grabación de nuevos comandos de voz que el sistema reconocerá como los controles para comandar el movimiento del “Mecanismo Robótico”. Si no fuese el caso, el sistema permanecerá en reposo hasta detectar un comando de voz, el cual será procesado y verificado. Si el comando detectado fuese válido, se enviará la orden al “Mecanismo Robótico” para que ejecute el movimiento correspondiente, caso contrario se seguirá esperando la detección de otro comando de voz.




















Luego, la parte de “Distribución de Energía” estará compuesto de una etapa de regulación de energía, esencial para poder alimentar a cada dispositivo según el voltaje requerido, y una etapa de censado para alertar en caso exista una escasez de energía en la fuente de alimentación.










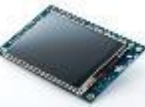














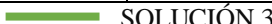
Finalmente, la parte del “Mecanismo Robótico” el cual recepcionará el comando de voz transmitido por la parte de “Reconocimiento de Voz”. Este comando será procesado para determinar la acción que ordena el usuario, y así poder ejecutar el movimiento que corresponda.

## 2.5 Matriz morfológica

Ya definidas las funciones que va a realizar cada parte, se procede a desarrollar cada una eligiendo el componente adecuado para cumplir dicha función, para ello se desarrolla la matriz morfológica de Zwicky, mostrada en la tabla 2.3, donde se obtiene una serie de posibles conceptos que posteriormente se evaluarán eligiéndose la más adecuada para el funcionamiento del sistema de asistencia de control por voz.

Tabla 2.3: Matriz Morfológica del Sistema.

	FUNCIÓN	OPCIÓN 1	OPCIÓN 2	OPCIÓN 3	OPCIÓN 4
RECONOCIMIENTO DE VOZ	Adquirir comando de voz	Micrófono a clip 	Micrófono para pc 	Micrófono de oreja 	
	Procesar comando de voz	Voice Recognition V2 - Elechouse 	ES6928P 	Easy VR 3.0 	Aplicativo de reconocimiento de voz en un móvil 
	Grabar comandos de voz	Arduino UNO 	Arduino Nano 	Arduino Mega ADK 	
	Validar comando de voz				
	Transmitir comando detectado	Radiofrecuencia 	Bluetooth 	Cable 	NRF24L01 
MECANISMO ROBÓTICO	Recibir comando detectado				
	Procesar acción a ejecutar	Laptop 	PcDuino 	Raspberry Pi 	BeagleBone 

					
	Ejecutar acción	Brazo Robótico MICC 	MANUS iArm 		
INTERFAZ DE USUARIO	Visualizar comando detectado	Laptop 	Pantalla LCD 	Móvil 	Pantalla TFT 
	Activar Configuración		Interruptor 		Pulsador 
DISTRIBUCIÓN DE ENERGÍA	Fuente de Energía	Batería 	Doméstica 	Doméstica y batería 	
	Regular Energía	Regulador para protoboard 	LM7805 	Regulador de baterías LIPO 	
	Censar nivel de Energía	Indicador de bajo voltaje 	Diseño propio LM3914N 		
		 SOLUCIÓN 1	 SOLUCIÓN 2	 SOLUCIÓN 3	

Fuente: Elaboración Propia

## 2.6 Evaluación de conceptos.

Con los resultados de la matriz morfológica desarrollada se proponen tres conceptos de solución indicados con líneas de color naranja, azul y verde, los cuales serán comparados eligiéndose el más adecuado para cumplir con los requisitos del sistema.

### 2.6.1 Conceptos de solución

Las tres soluciones desarrolladas apuntan al uso de un mismo dispositivo robótico de asistencia, denominado MICO de la empresa KINOVA (KINOVA, s.f.). Esto se debe a su rescatada ergonomía, eficacia en la sujeción de objetos, fácil instalación y disposición en el mercado.

El control del robot MICO se realizará a través de un hardware específico, el cual dispondrá de un algoritmo capaz de recibir información de la parte de “Reconocimiento de voz” para luego procesarlo y determinar el movimiento del robot. La configuración del robot se realizará mediante un software que la compañía Kinova pone a disposición.

Por otro lado, las tres soluciones difieren en los componentes que conforman la parte de “Reconocimiento de Voz”, “Distribución de Energía” e “Interfaz de Usuario”, es por ello que a continuación se detalla cada integración de los tres conceptos de solución.

#### *Concepto de solución 1:*

En la presente solución ilustrada en la figura 2.4, se considera el uso de una laptop como función de interfaz gráfica para el usuario, este mostrará por medio de una aplicación lo que el sistema desee informar, como mostrar las palabras reconocidas o alertar en caso de batería baja, asimismo también deberá interactuar con el usuario dando a entender si se recibió un comando de voz correcto o erróneo.

La adquisición del sonido se realizará mediante un micrófono de PC, la cual se adaptará para su instalación cerca a la boca del usuario. Este micrófono se conectará a un módulo reconocedor de voz V2 de la marca Elechouse, el cual se encargará de procesar la señal de voz y verificará si es un comando válido. Los comandos de voz para el control del sistema serán previamente configurados según el sonido de voz del usuario.



La respuesta del módulo de reconocimiento de voz sobre algún comando válido será enviada al módulo Arduino Uno, quien se encargará de relacionarlo con el tipo de movimiento que corresponda, para que posteriormente reenvíe la información hacia el dispositivo robótico. Para realizar el envío de información del Arduino hacia el dispositivo robótico, se aprovechará la laptop quien dispone de una conexión USB necesaria para la comunicación del robot.

La alimentación al sistema se basará en una conexión domestica requerida para el funcionamiento de la laptop y una batería quien alimentará los diversos dispositivos electrónicos.

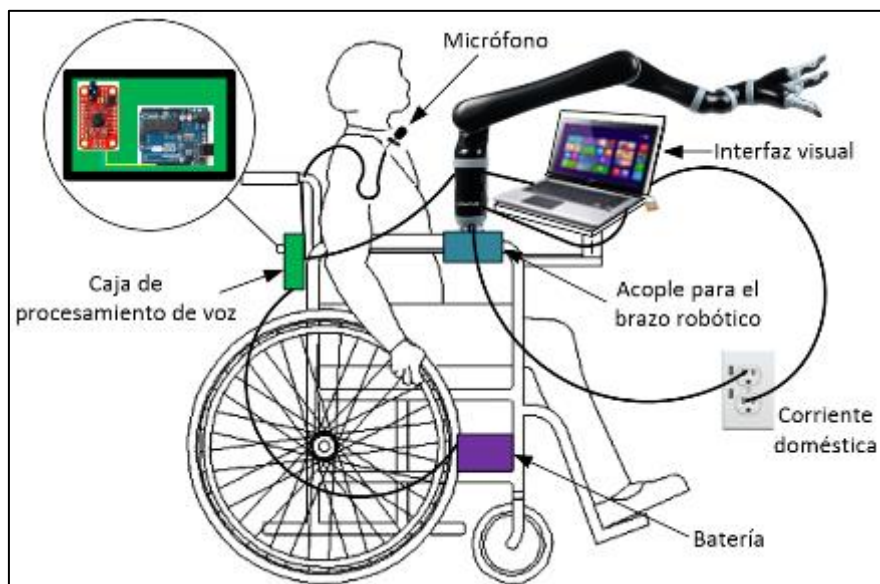


Figura 2.4: Concepto de Solución 1.  
Fuente: Elaboración Propia

### ***Concepto de solución 2:***

Esta solución representada en la figura 2.5, considera el uso de una interfaz visual basada en una pantalla LCD, en donde se podrá visualizar en todo instante el comando de voz que reconoce el sistema, un conjunto de leds indicadores que alertarán al usuario sobre escases de batería y un par de botones para poder entrar en modo de configuración.

Asimismo, esta alternativa propone el uso del módulo reconocedor de voz V2, de la marca Elechouse, quien recibirá órdenes de voz por medio de un micrófono de clip y los procesará con el fin de obtener un comando de voz. Posteriormente la respuesta será enviada al controlador Arduino NANO, el cual se caracteriza por ser pequeño y práctico de usar.

El Arduino NANO se encargará del procesamiento para determinar el movimiento, la información se enviará a un embebido PcDuino a través de una comunicación por radiofrecuencia quien por consiguiente se encargará de efectuar configuraciones básicas del robot y de ordenar movimientos al robot MICO.

La alimentación del sistema se realizará mediante una batería la cual energizará a los circuitos electrónicos y al robot de asistencia, los circuitos electrónicos requieren voltajes de 5V mientras que el robot se energizará con 24V.

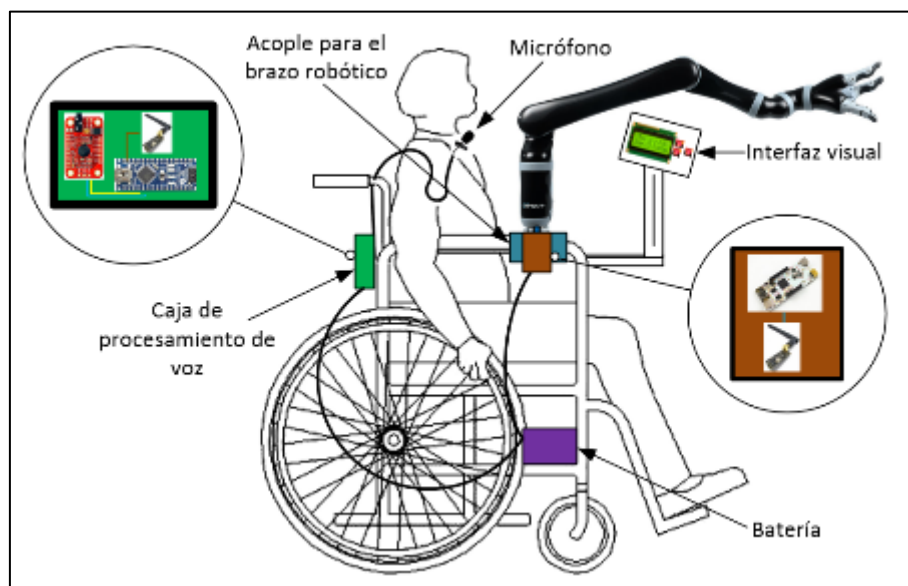


Figura 2.5: Concepto de Solución 2.  
Fuente: Elaboración Propia

### **Concepto de solución 3:**

Esta solución representada por la figura 2.6, propone usar un aplicativo de celular como interfaz de usuario en el sistema, este alertará sobre algún obstáculo y escases de batería, así mismo el aplicativo será encargado de procesar los comandos de voz recibidos. La etapa de adquisición de voz se realizará mediante un micrófono de auricular conectado al celular. Por otro lado, la etapa de procesamiento de voz se realizará por un algoritmo de reconocimiento de voz que brinda el sistema Android, el cual funciona con conexión a internet.

La respuesta de la etapa de reconocimiento se enviará vía bluetooth hacia un módulo PcDuino, quien lo procesará y ordenará la ejecución de movimiento del robot de asistencia. Por último, el sistema se alimentará mediante una batería de 24 V debido a que solo existirán componentes electrónicos y el robot MICO.



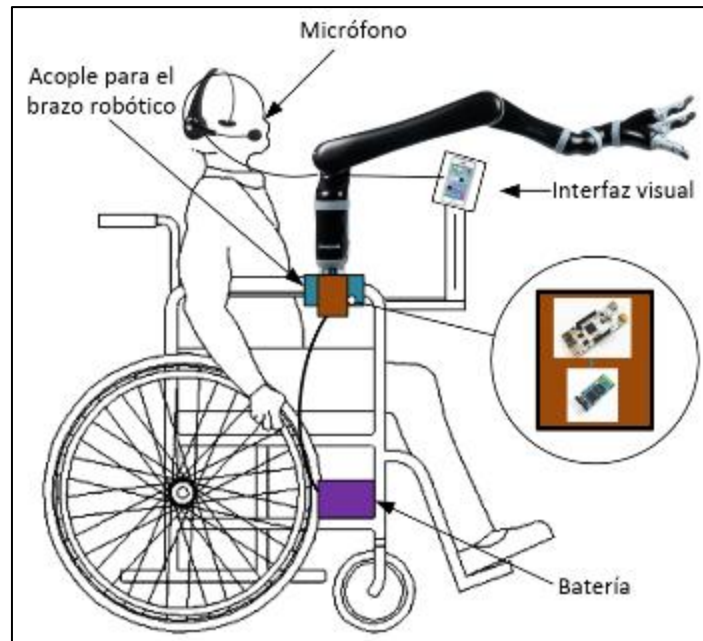


Figura 2.6: Concepto de Solución 3  
Fuente: Elaboración Propia

### 2.6.2 Evaluación técnico-económica de los conceptos de solución

Con los tres conceptos de solución obtenidos por medio de la matriz morfológica, se procede a realizar una evaluación técnico-económica por el método de evaluación de proyectos según la norma VDI 2225, calificando los conceptos según criterios económicos y técnicos con el fin de obtener la mejor alternativa para el desarrollo del sistema. En el Anexo A se muestra a detalle el proceso de evaluación Técnico – Económica aplicada a los tres conceptos de solución.

#### *Evaluación Técnica*

Los criterios técnicos que se analizarán son:

- Seguridad, el usuario requiere de un sistema que no atente contra su integridad física, como evitar conexiones de alto voltaje cerca al usuario y estructuras que puedan colapsar sobre el usuario.
- Rapidez, el sistema debe estar en la capacidad de procesar las órdenes y ejecutarlas con la mayor precisión y rapidez posible, evitando retardos en los movimientos del robot.

- Manipulación, el sistema debe poseer un control sencillo y entendible para el usuario.
- Estabilidad, el sistema debe poseer una estructura segura que evite el colapso futuro de algún componente.
- Ergonomía, el sistema debe ser atractivo para el cliente objetivo debido a que será parte de sus implementos diarios.
- Mantenimiento, el mantenimiento de los componentes debe estar al alcance del usuario, ya que un componente muy complejo de reparar efectuaría un alto costo.
- Montaje, el sistema debe ser de fácil instalación en el entorno que se encuentre el discapacitado, el técnico o tercero que proceda con la instalación deberá hacerlo sin inconvenientes algunos.
- Peso, como el sistema acompañara en todo momento al usuario y en la mayoría de situaciones se empotrará a una silla de ruedas, se espera que el sistema tenga un peso no muy elevado que evite un esfuerzo extra para su traslación y no complique su instalación manual.

**Concepto de solución 1:** Esta solución cumple con el criterio de rapidez, ya que el uso de una computadora personal beneficia con altas velocidades en el procesamiento y envío de datos. Asimismo, cumple con el criterio de manipulación, ya que el usuario no tendrá problemas en interactuar con el sistema. Por otro lado, no cumple con el peso por el uso de la laptop, y montaje, por la conexión eléctrica doméstica que se requiere, lo cual imposibilita la movilización del sistema.

**Concepto de solución 2:** Esta solución cumple con la mayoría de criterios, sobresaliendo por su montaje en cualquier entorno del usuario y seguridad que le ofrece al usuario. A diferencia de la primera solución la rapidez de procesamiento no es muy elevada, pero es eficaz.

**Concepto de solución 3:** Esta solución cumple con el criterio de rapidez, ya que la velocidad de procesamiento que ofrece un móvil es alta. Asimismo, cumple con el criterio del peso ya que el dispositivo es muy liviano. Pero, la manipulación resulta ser muy ineficiente, debido a la dificultad en el usuario para usar aplicativos de celular.

### *Evaluación Económica*

Los criterios económicos que se analizarán son:

- Costos de dispositivos, el sistema debe ser lo más económico posible sin perder la calidad del sistema. Estando a disposición de cualquier persona, con posibilidad de ofrecerse en el mercado.
- Costos de transporte, posiblemente el sistema posea componentes que deberán importarse y su costo debe ser lo más reducido posible, considerando el lugar de donde provenga.
- Vida útil, mientras más larga sea la duración del componente menos se gastará en mantenimiento o compra de repuestos.
- Facilidad de obtención, los componentes del sistema deben estar a disposición del mercado actual.
- Costo de mantenimiento, el sistema debe poseer un bajo costo por reparación.

**Concepto de solución 1:** Es ineficiente con respecto al costo de dispositivos, ya que el costo de una laptop eleva el presupuesto, pero resulta favorable con respecto al criterio de vida útil del sistema.

**Concepto de solución 2:** Es eficiente respecto al costo de dispositivos, debido a poseer componentes atractivos económicamente hablando, y están en disposición del mercado.

**Concepto de solución 3:** Esta solución es menos eficiente que la solución 2 respecto a costos de dispositivos, debido al costo de un móvil con sistema Android, pero resulta favorable con respecto a su disposición en el mercado.

En la figura 2.7 se muestran los resultados obtenidos de la evaluación Técnico-Económica realizada en el “anexo A”, gráfica del valor técnico vs el valor económico, donde se elige el concepto de solución más cercano a la solución ideal. Como es de apreciarse, el concepto de solución 2 es la más cercana a una solución ideal, por ello esta solución será la seleccionada y desarrollada en el transcurso del documento.

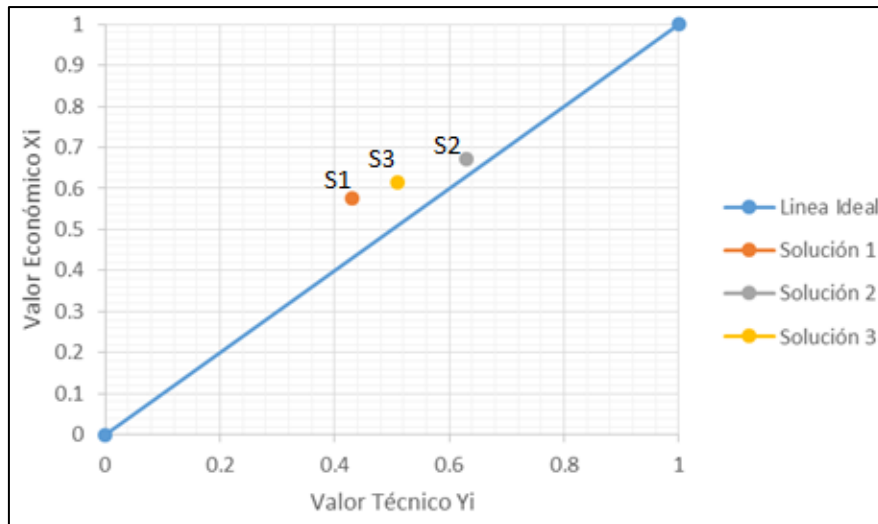


Figura 2.7: Evaluación Técnico-Económica según VDI 2225.  
Fuente: Elaboración Propia

### 2.6.3 Concepto de solución óptimo

La figura 2.8 muestra el diseño del concepto óptimo. El sistema consiste en una interfaz de usuario basada en una pantalla LCD y botones para su configuración (1). Este módulo de interfaz se instalará a la altura del alcance visual del usuario. El soporte de la interfaz debe ser fácilmente maniobrable para su calibración. La comunicación de la interfaz con el reconocimiento de voz se realizará de forma directa mediante una conexión de 8 cables.

La parte de reconocimiento de voz (2) consiste en un módulo reconocedor de voz de la marca Elechouse, quien será el encargado de procesar la señal de voz obtenida por el micrófono cercano a la boca del usuario, esta señal será comparada por una serie de comandos de voz previamente configurados eligiéndose la más idéntica. El control del sistema de reconocimiento de voz lo ejecutará el Arduino NANO encargado de configurar al módulo reconocedor de voz y a la interfaz de usuario, este controlador enviará las respuestas obtenidas del reconocimiento vía radiofrecuencia al módulo de control del mecanismo robótico, el cual consiste de un PcDuino (3). El PcDuino se encargará de controlar al robot por medio de la previa instalación de las librerías que nos brinda el fabricante.

El papel del robot de asistencia lo efectuará el robot MICO de la empresa KINOVA (4), quien ofrece un control mediante una API. El sistema dispondrá de dos baterías para la alimentación, uno energizará el reconocimiento de voz ubicado a la espalda de la silla del discapacitado y el otro será ubicado debajo de la silla el cual alimentará el robot MICO y su dispositivo de control, en caso se instale en una silla eléctrica se usará la batería de esta.

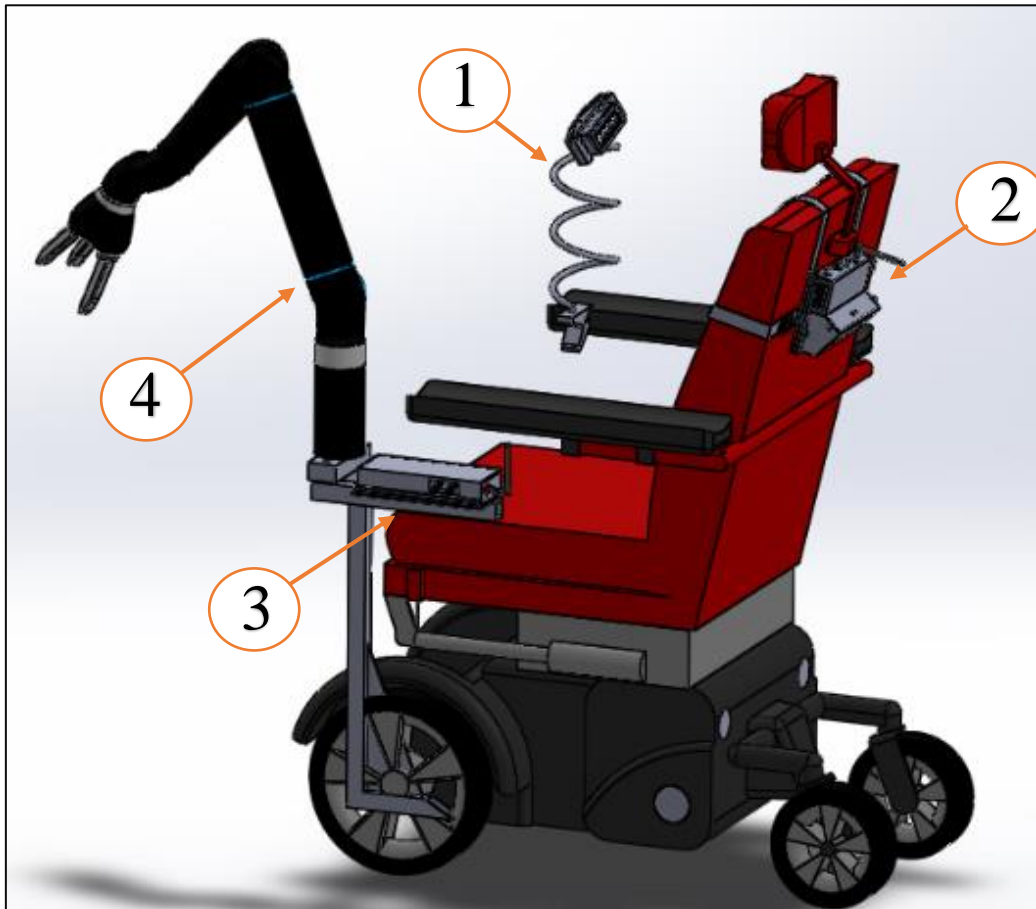
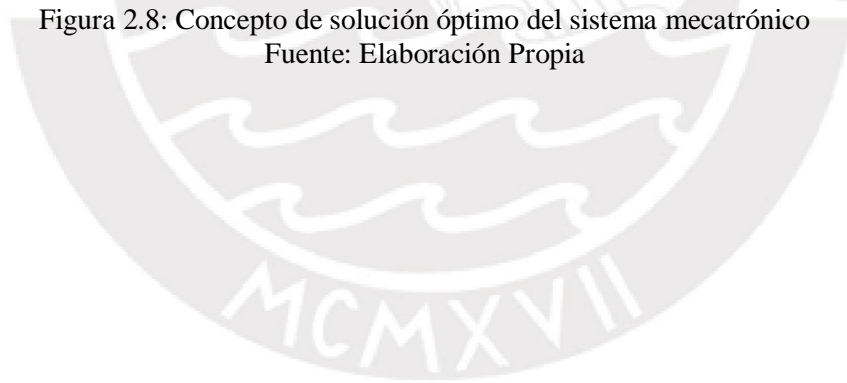


Figura 2.8: Concepto de solución óptimo del sistema mecatrónico  
Fuente: Elaboración Propia



## CAPÍTULO III

### DISEÑO DEL SISTEMA DE ASISTENCIA POR RECONOCIMIENTO DE VOZ

El presente capítulo contempla el desarrollo del sistema de control por voz con respecto al diseño de la parte electrónica, mecánica y de control, enfatizando cada elemento que lo compone y sus características técnicas.

#### 3.1 Diseño electrónico

##### 3.1.1 Diagrama de bloques del sistema

El diagrama de bloques de la figura 3.1 revela las conexiones existentes en el sistema de control por voz. En ello se puede apreciar al sistema compuesto por dos controladores: un Arduino NANO, perteneciente a la parte de reconocimiento de voz, el cual está conectado con el módulo reconocedor de voz, con la pantalla LCD y con el módulo NRF24L01, siendo este último de utilidad para la transmisión de datos inalámbricos. Por otro lado, un controlador PcDuino, perteneciente al mecanismo robótico, quien está en conexión con el módulo NRF24L01 (para la recepción de datos inalámbricos) y con el robot de asistencia MICO Arm vía USB. La comunicación entre ambos controladores es vía inalámbrica.

El módulo reconocedor de voz se comunica con el Arduino NANO a través de los pines TX y RX, reservados para el envío y recepción de datos, estos serán usados en la configuración del módulo y para poder obtener los comandos reconocidos durante el funcionamiento del sistema. La pantalla LCD se comunica usando el protocolo I2C a través de los pines SDA y SCL del Arduino NANO, disminuyendo el uso de pines en el controlador. El Robot de asistencia MICO Arm se controla a través del PcDuino vía USB 2.0, siendo esto posible gracias a las librerías que comparte el fabricante del robot.



En cuanto a la alimentación, se hace el uso de una batería LiPo de 2 celdas proporcionando aproximadamente 7.4V los cuales se regularán a 5V para energizar al controlador del reconocimiento de voz; una batería de 24V para energizar directamente al Robot MICO y un regulador a 5V para energizar el controlador del PcDuino.

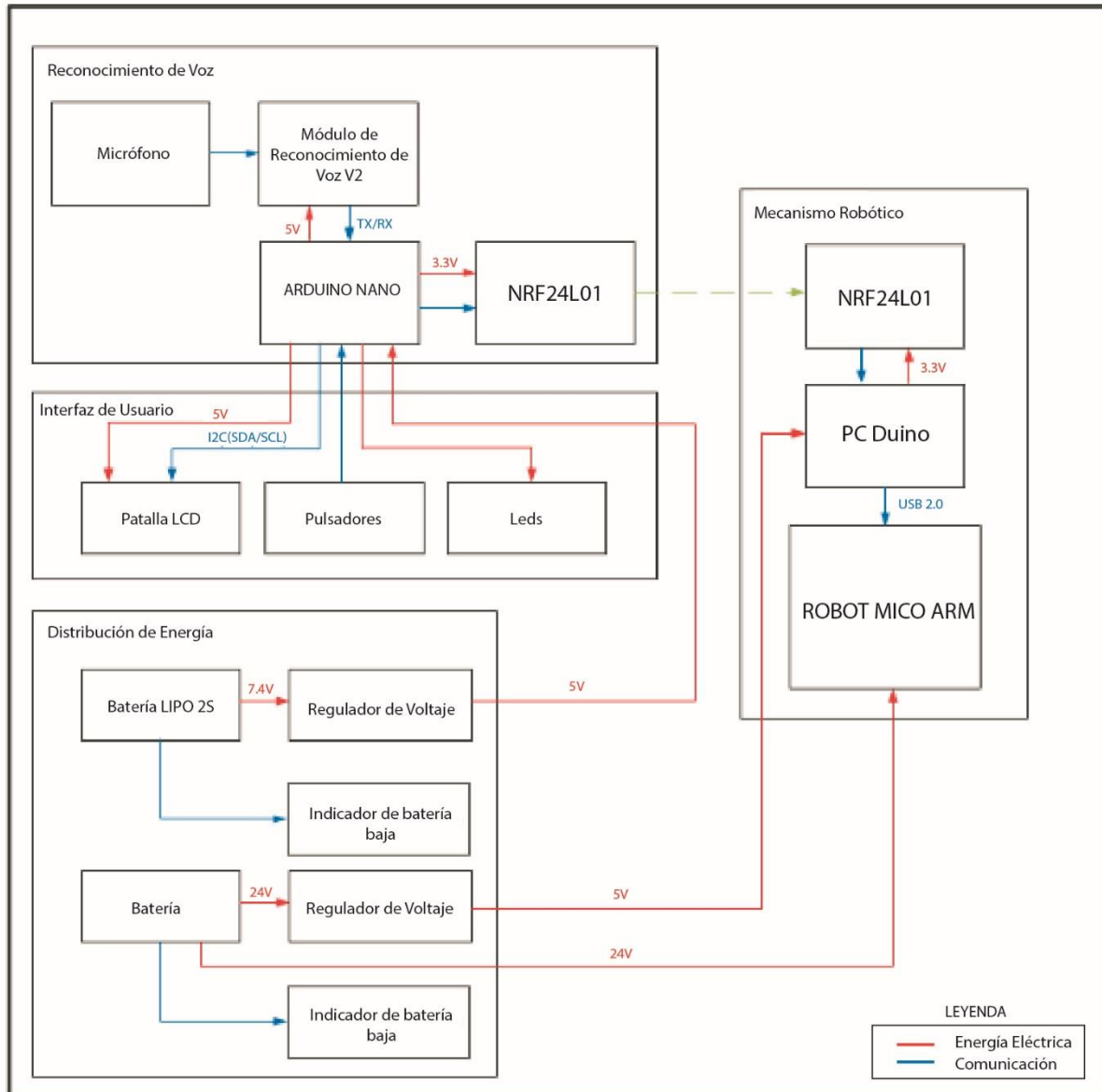


Figura 3.1: Diagrama de bloques del sistema.  
Fuente: Elaboración Propia

### 3.1.2 Componentes electrónicos

En este subcapítulo se describirá cada dispositivo electrónico que forma parte de la composición de cada subsistema, adicionando algunas características técnicas y la ubicación dentro del sistema de asistencia de control por voz.

### 3.1.2.1 Reconocimiento de voz

A continuación, se expondrán los diversos componentes electrónicos que forman parte del reconocimiento de voz (figura 3.2), el cual se ubicará en el espaldar de la silla del discapacitado, ya que requiere estar lo más cercano posible al micrófono para evitar interferencias en la señal por longitud del cable.

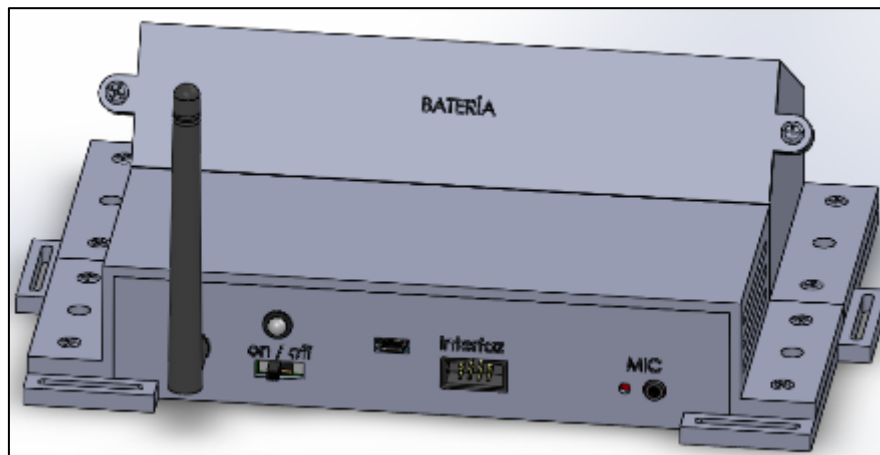


Figura 3.2: Reconocimiento de voz.  
Fuente: Elaboración Propia

#### a) Módulo reconocedor de voz

Este dispositivo será el encargado de procesar las señales de voz generadas por el usuario. Su funcionalidad se basa en comparar las señales de voz previamente grabadas con las palabras pronunciadas por el usuario, este emitirá un comando al identificar una alta similitud entre ellas. En la tabla 3.1 se muestran los principales requerimientos que deberá cumplir el módulo.

Tabla 3.1: Requerimientos del módulo reconocedor de voz

Compatible con Arduino	si
Cantidad de comandos a grabar	>10 comandos
Disponibilidad en el mercado	si

Fuente: Elaboración Propia

Orientado a cumplir con todos los requerimientos, se optó por tres tipos de módulos: módulo de reconocimiento de voz V2 de la marca Elechouse, módulo de reconocimiento de voz EasyVR 3.0 de la marca VeeR y el módulo ES6928P Voice Recognition. En la tabla 3.2 se muestra una breve comparación entre las 3 alternativas escogidas, para una comparación más detallada revisar el Anexo B.

Tabla 3.2: Módulos reconocedores de voz

Modulo	Compatible con Arduino	Cantidad de comandos a grabar	Disponibilidad en el mercado
Reconocimiento de voz V2 - Elechouse	Si	15	Si
Reconocimiento de voz EasyVR 3.0	Si	32	Si (Importación)
ES6928P Voice Recognition	Si	4	No

Fuente: [www.veear.eu](http://www.veear.eu) / [www.elechouse.com](http://www.elechouse.com)

Como resultado el módulo de la marca Elechouse y el módulo EasyVR están en condiciones necesarias para realizar eficazmente la etapa de reconocimiento de voz para el proyecto, pero se elegirá el módulo de reconocimiento de voz de la marca Elechouse (figura 3.3) debido a la flexibilidad en su programación y la facilidad de obtención en el mercado. La tabla 3.3 muestra algunas características de funcionamiento del módulo, adicionalmente el anexo G muestra el manual del dispositivo, el cual contiene información en detalle de su funcionamiento, así como el lenguaje de códigos que se usa para su programación.

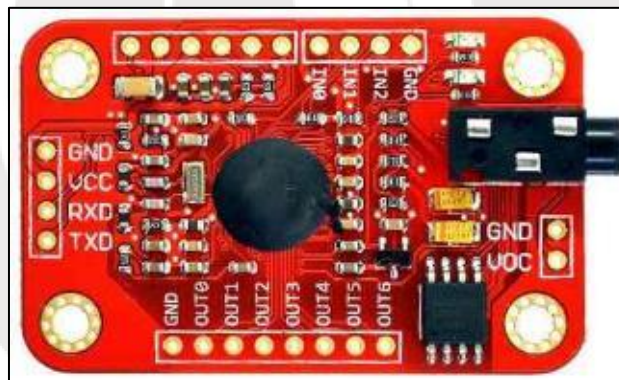


Figura 3.3: Módulo reconocedor de voz V2 - Elechouse.

Fuente: [www.elechouse.com](http://www.elechouse.com)

Tabla 3.3: Características técnicas del módulo reconocedor de voz

Características Técnicas	
Voltaje de alimentación	4.5v - 5.5v
Consumo	<40mA.
Comunicación	UART 5V TTL
Tamaño	30mm x 47.5mm

Fuente: [www.elechouse.com](http://www.elechouse.com)

Este dispositivo es capaz de almacenar 15 comandos de voz separados en 3 grupos de 5 comandos cada uno. Estos grupos de comandos se deben programar por separado y en caso se

desea reconocer un comando, primero se deberá activar el grupo al que pertenece. El módulo posee un 99% de precisión en el reconocimiento dentro de un entorno ideal.

La programación del dispositivo se efectuará mediante una comunicación serial y usando instrucciones en código ASCII detallados en el manual del dispositivo ubicado en el Anexo G, este paso generalmente se realiza a través de una PC, pero gracias a su flexibilidad podemos programarla también a través de un Arduino haciendo uso de los pines VCC, GND, TXD y RXD. La figura 3.4 muestra la ubicación del dispositivo.

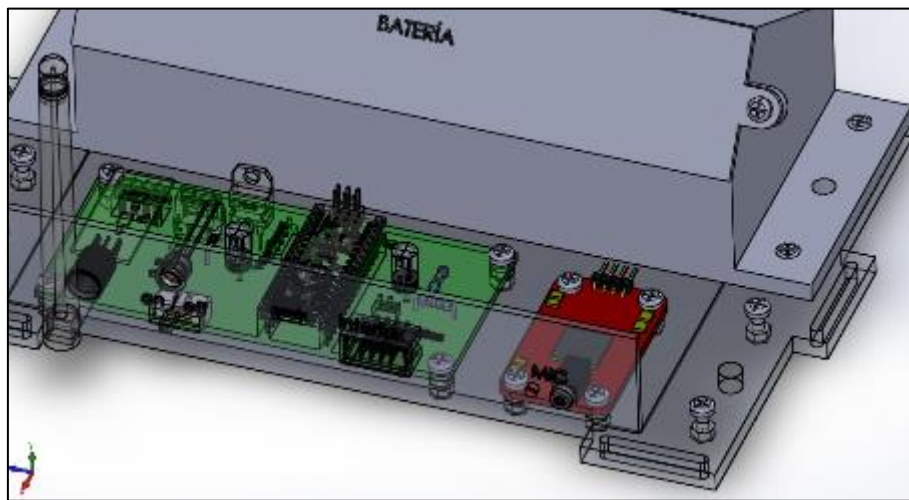


Figura 3.4: Ubicación del módulo reconocedor de voz.

Fuente: Elaboración Propia

### ***b) Controlador***

Este dispositivo será el encargado de controlar la parte del reconocimiento de voz ya que recibirá los comandos obtenidos por el módulo reconocedor de voz y los enviará a la interfaz de usuario para su visualización a través de una pantalla LCD, asimismo también enviará el comando reconocido a la parte del mecanismo robótico para que ejecute la acción correspondiente.

Este dispositivo también tendrá la función de configurar al módulo reconocedor de voz, grabando los comandos de voz a reconocer. En la tabla 3.4 se muestra los requisitos que debe tener este controlador.

Tabla 3.4: Requerimientos del Controlador para la parte de reconocimiento de voz

Numero de pines digitales	>6
Comunicación I2C y UART	Si
Salida de 3.3V	Si
Dimensiones	La menor posible

Fuente: Elaboración Propia

Se optó por tres modelos de controlador: Arduino UNO, Arduino NANO y Arduino Mega ADK. En la tabla 3.5 se muestra una breve comparación entre los 3 módulos.

Tabla 3.5: Controladores para la parte de reconocimiento de voz

Controlador	Numero de pines digitales	Comunicación UART y I2C	Dimensiones	Precio
Arduino NANO	14	Si	45mm x 18mm	S/.50
Arduino MEGA	54	Si	101.52mm x 53,3 mm	S/. 110
Arduino UNO	14	Si	68.6mm x 53.4mm	S/. 80

Fuente: [www.arduino.cc](http://www.arduino.cc)

El módulo a usar será el Arduino NANO (Figura 3.5) debido a poseer la misma funcionalidad que un Arduino UNO, pero con medidas reducidas, y a diferencia del Arduino Mega ADK es más económico. Por otro lado, posee la cantidad de puertos necesarios para controlar una pantalla LCD y comunicarse con el módulo reconocedor de voz. La figura 3.6 muestra la ubicación del dispositivo dentro del sistema y la tabla 3.6 muestra una breve descripción de algunas características técnicas del controlador.



Figura 3.5: Controlador Arduino NANO.

Fuente: [www.arduino.cc](http://www.arduino.cc)



Tabla 3.6: Características técnicas del controlador para la parte de reconocimiento de voz

Características Técnicas	
Voltaje de alimentación	7 - 12V
Microcontrolador	ATmega328
Corriente por I/O Pin	40 mA
Velocidad de reloj	16Mhz
Pines analógicos	8

Fuente: www.elechouse.com

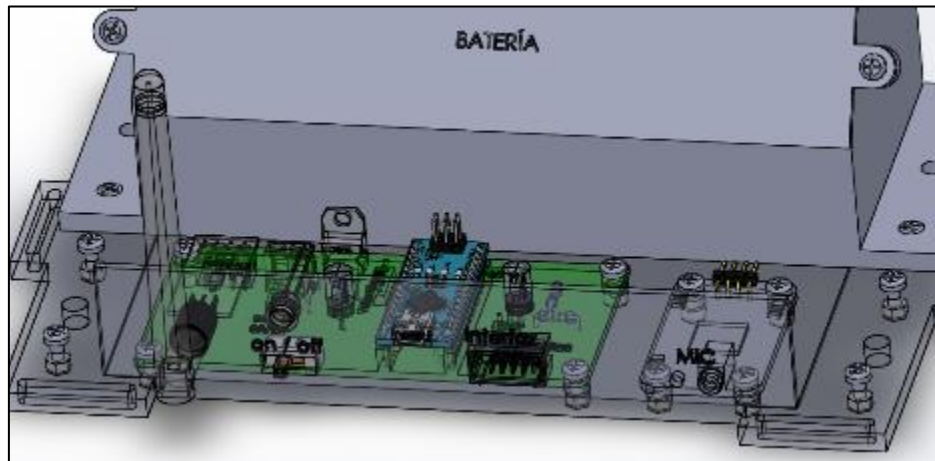


Figura 3.6: Ubicación del controlador Arduino NANO.

Fuente: Elaboración Propia

### c) Transmisión y recepción de datos

Este dispositivo efectuará la comunicación entre la parte de reconocimiento de voz y el mecanismo robótico. La comunicación se realizará inalámbricamente con el fin de evitar accidentes debido a enredos por cables. Adicionalmente se deja abierta la posibilidad de poder controlar otros dispositivos de asistencia que se deseen anexar al sistema. Para esta etapa se optó por dos módulos de comunicación inalámbrica: el módulo NRF24L01 y el Xbee Pro series 1. La tabla 3.7 detalla una comparación entre los dos componentes.

Tabla 3.7: Comparación entre módulos de transmisión y recepción de datos

	Módulo NRF24L01 con antena	Xbee Pro series 1
Alimentación	3.3V	3.3V
Banda de trabajo	2.4Ghz	2.4Ghz
Potencia máxima de salida	+20dbm	+18dbm
Tasa de transmisión	250kbps	250kbps
Distancia	Hasta 1000 m (lugares abiertos)	1500m
Compatible con Arduino	Si	Si
Característica	Emisor ó Receptor (Según configuración)	Emisor y Receptor
Precio Referencial	S/. 35.00	S/. 130.00

Fuente: www.sparkfun.cc



Como se observa, el módulo Xbee Pro es una gran opción debido a que no solo tiene un alto rango de distancia en transmisión, sino también funciona como emisor y receptor a la vez, pero son funciones esenciales que el NRF24L01 las cumple sin ser muy sofisticado, ya que el sistema solo emitirá ordenes mas no habrá retorno de algún dato por parte del mecanismo robótico. Como resultado se descarta el Xbee y se opta por el NRF24L01 (Figura 3.7) cuyo rango de comunicación está dentro de los 10 metros y su costo es más bajo. La figura 3.8 muestra la ubicación del dispositivo en la parte del reconocimiento de voz y en el mecanismo robótico.



Figura 3.7: Módulo de transmisión inalámbrica NRF24L01.

Fuente: [www.sparkfun.cc](http://www.sparkfun.cc)

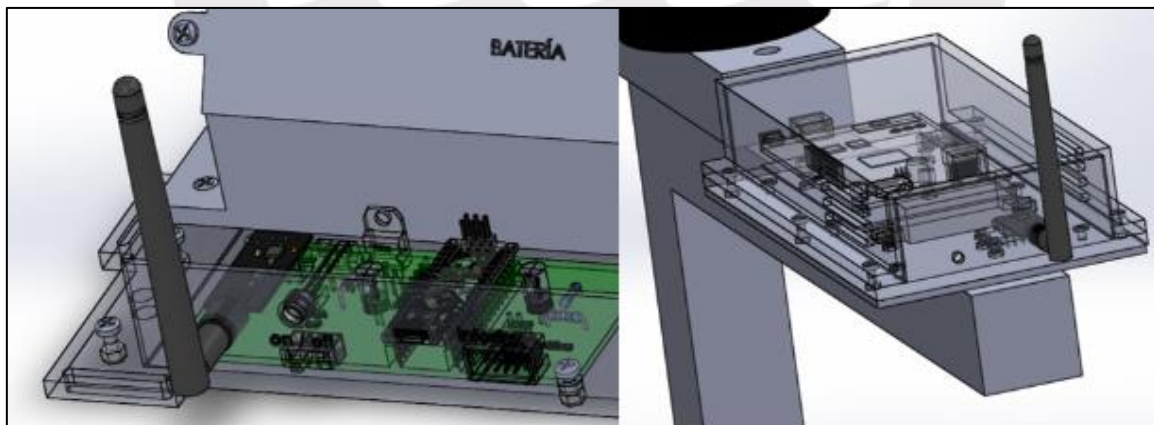


Figura 3.8: Ubicación del módulo NRF24L01

### 3.1.2.2 Interfaz de usuario

A continuación, se expondrán los diversos componentes electrónicos que forman parte de la interfaz de usuario, el cual se muestra en la figura 3.9, este se encontrará ubicado en la dirección visual de la persona discapacitada y se sostendrá por una varilla para su ajuste a una posición adecuada.



Figura 3.9: Interfaz de Usuario  
Fuente: Elaboración Propia

### a) Visualización

Para visualizar el comando detectado se optó por una pantalla LCD (figura 3.10) debido a ser la más económica entre las interfaces visuales, ya que solo se requiere la visualización de una palabra. En este caso se eligió una pantalla LCD de 16x02, es decir puede escribir hasta 16 caracteres horizontalmente y 2 verticalmente. Este dispositivo será controlado por el Arduino NANO encargado de controlar el módulo de reconocimiento de voz. La tabla 3.8 resume las características del dispositivo.

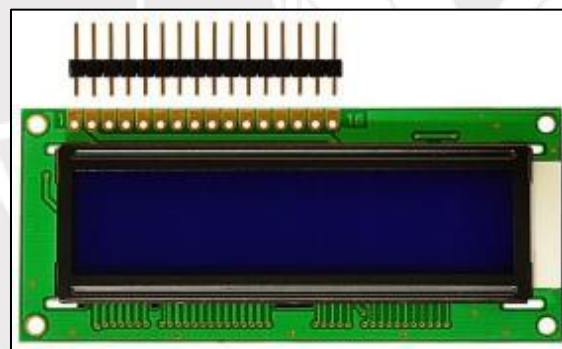


Figura 3.10: Pantalla LCD 16x2.  
Fuente: [www.sparkfun.cc](http://www.sparkfun.cc)

Tabla 3.8: Características técnicas de la pantalla LCD 16x02

Características Técnicas	
Voltaje de alimentación	4.5 - 5.5v,
Cantidad de palabras a mostrar	32 caracteres (2 filas de 16 cada una)
Dimensiones	80mm x 35mm x 11mm
Luz de fondo	Si

Fuente: [www.sparkfun.cc](http://www.sparkfun.cc)

La cantidad de pines que requiere la pantalla LCD puede acaparar todos los pines del módulo Arduino NANO sin espacio para la conexión de la comunicación inalámbrica, por ello

se opta por adicionar a la pantalla un módulo serial I2C 1602 (figura 3.11), el cual permite controlar la pantalla usando solo 4 pines, reduciendo la cantidad de pines a usar en el Arduino NANO.

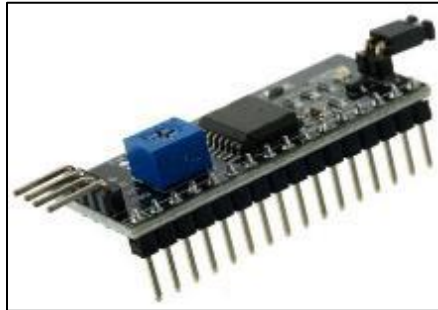


Figura 3.11: Módulo de comunicación I2C para la pantalla LCD.

Fuente: [www.electrocrea.com](http://www.electrocrea.com)

La característica de este módulo es que permite comunicar dispositivos a través de un protocolo de control de dos hilos (I2C), es decir a través de una comunicación por bus, donde el pin SDA le permite transmitir los datos y el pin SCL es el reloj que indica cuando leer los datos. Cada dispositivo conectado en bus tiene su propia dirección, en este caso el LCD es la 0x27. Por otro lado, toda comunicación en bus dispone de un dispositivo que actúa como Maestro y otros como esclavos, para este caso el maestro será el Arduino NANO y el esclavo la pantalla LCD.

Adicionalmente, el dispositivo funciona con 5Vdc y posee un potenciómetro ajustable para la nitidez de la pantalla. Dispone de 4 pines GND, +5 VDC, SCL y SDA, estos dos últimos se conectarán en los pines analógicos del Arduino: A4 (SDA) y A5 (SCL). La figura 3.12 y la figura 3.13 muestran la ubicación de cada uno de los dispositivos.

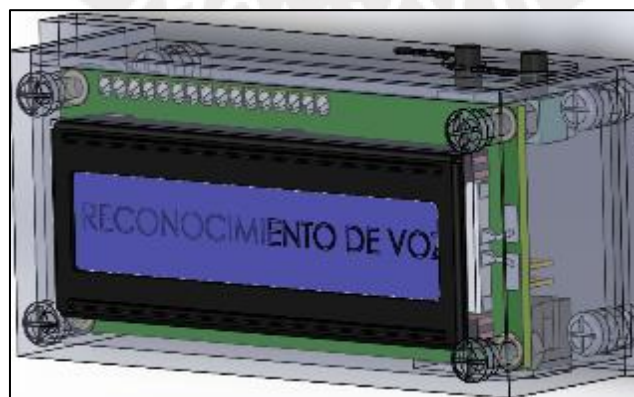


Figura 3.12: Ubicación de la pantalla LCD 16x02

Fuente: Elaboración Propia

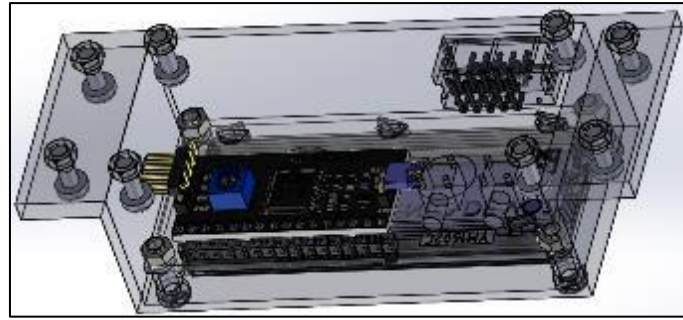


Figura 3.13: Ubicación del módulo I2C  
Fuente: Elaboración Propia

### 3.1.2.3 Mecanismo Robótico

Esta parte estará compuesto por 2 componentes: el robot MICO Arm de la empresa KINOVA quien tendrá el papel del robot de asistencia y la parte de control del robot basado en un controlador PcDuino encargado de la etapa de procesamiento, codificación del comando recibido y emisión de la orden de movimiento al robot MICO. El comando se recibirá a través de un módulo NRF24L01 conectado al PcDuino.

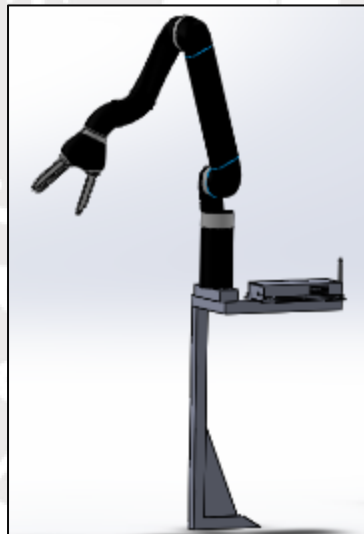


Figura 3.14: Mecanismo Robótico  
Fuente: Elaboración Propia

#### a) Robot MICO ARM

El robot MICO (figura 3.15) es un robot para la asistencia de personas discapacitadas, que poseen limitaciones en sus movimientos corporales ayudándolas a efectuar acciones de sujeción emulando el comportamiento de un brazo humano. Este robot fue diseñado con la finalidad de brindar una herramienta a las personas discapacitadas para que puedan ejercer

actividades sin la necesidad de depender de otra persona mejorando su calidad de vida y confianza en sus capacidades.

Este robot fue elegido a diferencia del Robot MANUS iArm, debido a ser más comercializado en la actualidad y por poseer un mejor desarrollo tecnológico, posee un entorno de desarrollo para que personas que adquieran el producto puedan adaptarlo a sus innovaciones brindando librerías para acceder al control del robot por medio de códigos de programación.



Figura 3.15: Robot MICO Arm.  
Fuente: [www.kinovarobotics.com](http://www.kinovarobotics.com)

El brazo del robot MICO Arm posee 6 grados de libertad al igual que su producto hermano JACO Arm, pero diferenciándolo al ser más liviano y fabricado en base a plástico reforzado en lugar de fibra de carbono. La extensión total del brazo puede alcanzar a los 700mm, adicionalmente se le acopla una pinza, la cual puede estar constituida en base a dos o 3 dedos, cada dedo posee un material de goma para una mejor sujeción y un diseño ligero que se ajusta a la forma del objeto. La forma de estos dos elementos se muestra en la figura 3.16.



Figura 3.16: Izquierda: Brazo. Derecha: pinza de 2 dedos y pinza de 3 dedos.  
Fuente: [www.kinovarobotics.com](http://www.kinovarobotics.com)



La tabla 3.9 muestra las características técnicas fundamentales del robot MICO, para una mayor información a detalle se adjunta las especificaciones técnicas del robot en el Anexo H.

Tabla 3.9: Características técnicas del robot MICO Arm.

Características Técnicas	
Voltaje de alimentación	18VDC – 29VDC
Consumo promedio	25 W
USB 2.0	12Mbps
RS485 interno	11.25Mbps
Peso total	5 Kg
Carga útil	1.25 Kg a mediana extensión 750 g a extensión completa
Estiramiento máximo	70 cm
Tipo de actuadores (en el brazo y dedos)	Motores Brushless DC
API y Software	Compatible con Windows, linux Ubuntu & ROS Conexión mediante el puerto USB 2.0 Programable en lenguajes C# y C++

Fuente: [www.kinovarobotics.com](http://www.kinovarobotics.com)

El control del robot se realiza mediante el uso de un joystick (figura 3.17) o por medio de un ordenador. Si fuese a través de un ordenador, existe un software denominado JACOSOFT, el cual brinda las herramientas para poder configurar la utilización del robot, permitiendo programar la posición de reposo, regular velocidades de cada actuador, establecer lugares donde el robot no debe acceder para la protección ante colisiones e indicar el movimiento que efectuará cada control del joystick.

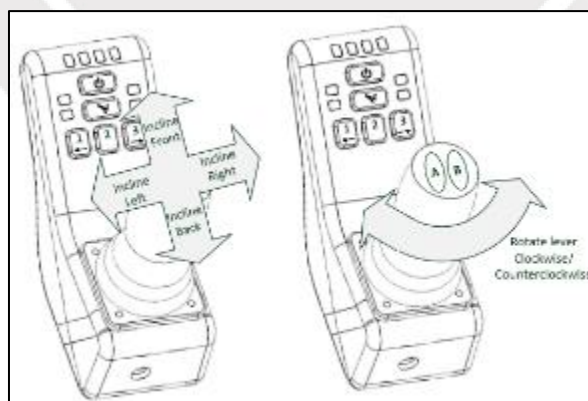


Figura 3.17: Joystick del Robot MICO Arm.

Fuente: [www.kinovarobotics.com](http://www.kinovarobotics.com)

El control del robot se puede efectuar a través de 3 tipos de movimientos: trasladar, rotar y el movimiento del agarre. La figura 3.18 muestra las direcciones de control, estos serán los principales movimientos que ejecutara el sistema de asistencia de control por voz.



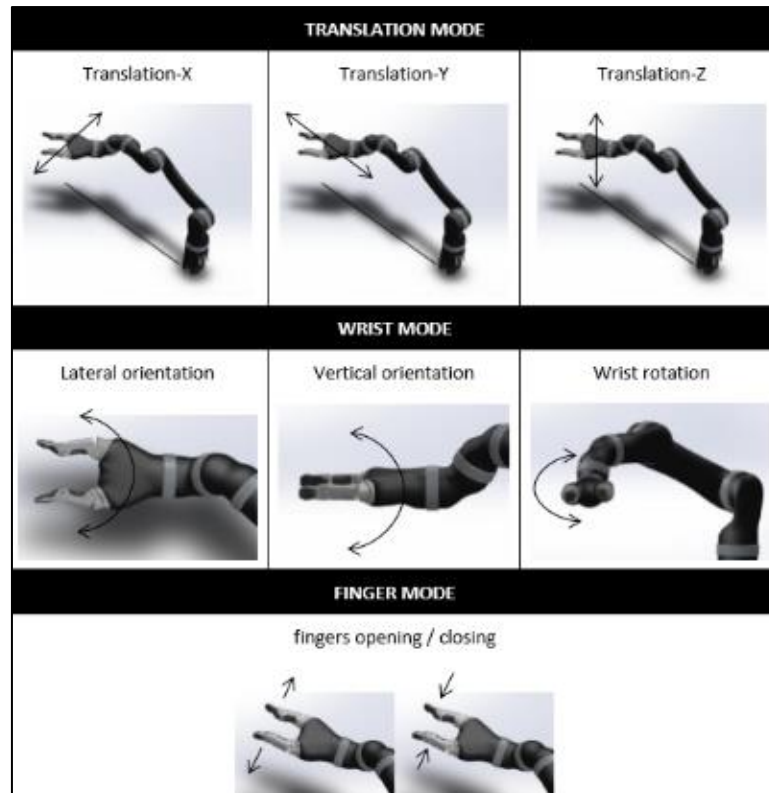


Figura 3.18: Movimientos del Robot MICO Arm.  
Fuente: [www.kinovarobotics.com](http://www.kinovarobotics.com)

Adicionalmente cuenta con una interfaz para aplicaciones avanzadas (API), permitiendo que los usuarios puedan controlar el robot a través del puerto USB 2.0, configurándolo e integrándolo a sus propios proyectos siendo compatible con Windows (7, 8) y Linux Ubuntu (12.04 y 14.04), para efectuar el control solo es necesario la utilización de las librerías "Kinova.API.UsbCommandLayer.h" y "KinovaTypes.h" que brinda el proveedor siendo compatible con el lenguaje de programación C++ y C#.

#### ***b) Controlador del mecanismo robótico***

Este dispositivo será el encargado de recibir la información proveniente del sistema de reconocimiento de voz a través de una comunicación inalámbrica (módulo NRF24L01), esta información será procesada con el fin de determinar el tipo de movimiento que ejecutará el robot de asistencia, así mismo gracias a la capacidad de su sistema operativo se hará uso de las librerías del robot MICO para codificar el comando y enviarlo a través de la conexión USB 2.0 hacia el robot, obteniendo como resultado el movimiento deseado. En la tabla 3.10 se detallan los requerimientos esperados de este controlador.

Tabla 3.10: Requerimientos del Controlador del mecanismo robótico.

Comunicación USB 2.0	Si
Capacidad para instalar librerías del robot	Si
Velocidad de procesamiento y capacidad de almacenamiento	Altas
Lenguaje de programación	C++ o C#

Fuente: Elaboración Propia

Siguiendo los requisitos del controlador se obtuvieron 3 alternativas posibles: un PcDuino V2, un Raspberry Pi 2 modelo B y un BeagleBone Black, la tabla 3.11 muestra una comparación resumida entre estos tres componentes.

Tabla 3.11: Controlador del mecanismo robótico.

Controlador	PcDuino v3	Raspberry Pi 2 modelo B	BeagleBone Black
Comunicación USB 2.0	Si	Si	Si
Admite librerías del robot	Si	Si	Si
Velocidad de procesamiento y capacidad de almacenamiento	CPU 1GHz ARM Cortex A8 1 GB RAM	CPU 1GHz ARM Cortex A8 512 RAM	CPU 900MHz ARM Cortex A7 1GB RAM
Lenguaje de programación que admite	C++, C#, Java y Python	C++, C#, Java y Python	C++, C#, Java y Python
Precio Referencial	S/. 240.00	S/. 300.00	S/. 200.00

Fuente: [www.linksprite.com](http://www.linksprite.com) / [www.raspberrypi.org](http://www.raspberrypi.org) / [www.beagleboard.org](http://www.beagleboard.org)

Los tres dispositivos comparados cumplen con todas las especificaciones requeridas para el control del robot de asistencia, sin embargo, se escogerá al PcDuino V3 (figura 3.19) debido a su mayor velocidad de procesamiento y su alta capacidad de almacenamiento respecto a los otros dispositivos. La tabla 3.12 resume algunas características específicas del controlador seleccionado, y en la figura 3.20 se puede observar la ubicación del dispositivo.

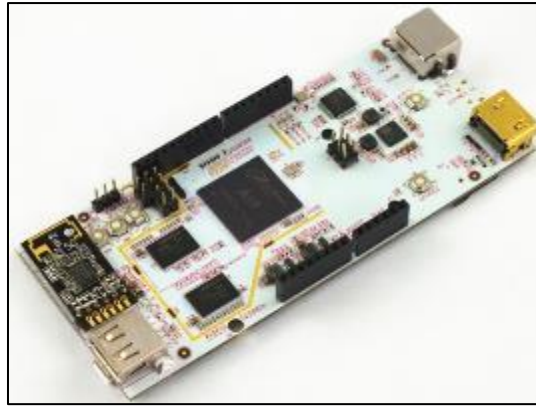


Figura 3.19: Controlador Pcduino V3.

Fuente: [www.linksprite.com](http://www.linksprite.com)

Tabla 3.12: Características técnicas del Pcduino

Características Técnicas	
CPU	1GHz ARM Cortex A8
RAM	1GB
Números de puertos USB	1
OS	Linux3.0 + Ubuntu 12.04, Android ICS 4.0
Voltaje de alimentación	5V (2000mA)
Medidas	125mm X 52mm

Fuente: [www.linksprite.com](http://www.linksprite.com)

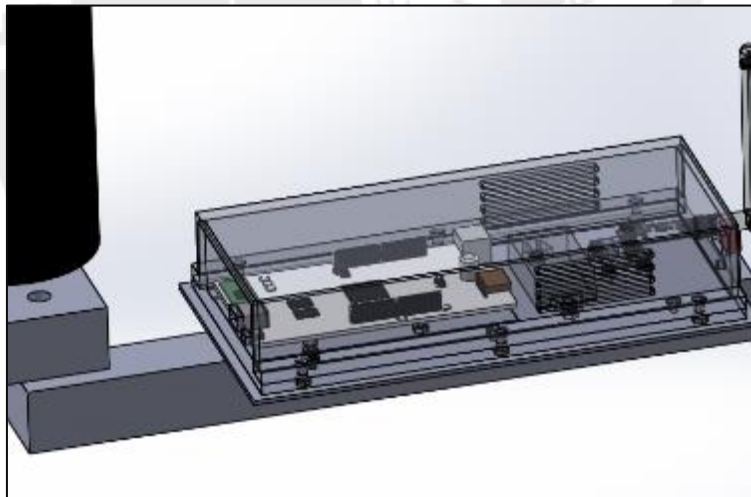


Figura 3.20: Ubicación del controlador del mecanismo robótico.

Fuente: Elaboración Propia

#### 3.1.2.4 Distribución de la energía

Esta parte se divide en dos grupos, el primero se encargará de la distribución de la energía a través de todo el reconocimiento de voz y la interfaz visual, siendo los componentes que lo conforman una batería recargable y un regulador de voltaje a 5V; el segundo se encargará de la distribución a través de todo el mecanismo robótico, energizando al robot y a su módulo de

control, siendo los componentes que lo conforman una batería de 24V recargable y un regulador a 5V, adicionalmente se detallará brevemente el tipo de cable que se usara para cada parte.

El sistema por lo general debe estar en funcionamiento como mínimo 6 horas y un máximo de 10 horas al día, esto se debe ya que según el Grupo Nacional para el Estudio y Asesoramiento en Úlceras por Presión y Heridas Crónicas (GNEAUPP) (SMITH&NEPHEW, 2014), el discapacitado debe de ser cambiado de posición cada 2 horas como máximo para evitar la aparición de úlceras en pacientes que no poseen movilidad, otorgando una rutina ejemplo de cómo sería los cambios posicionales en el paciente durante un día, esta rutina se puede observar en la figura 3.21.

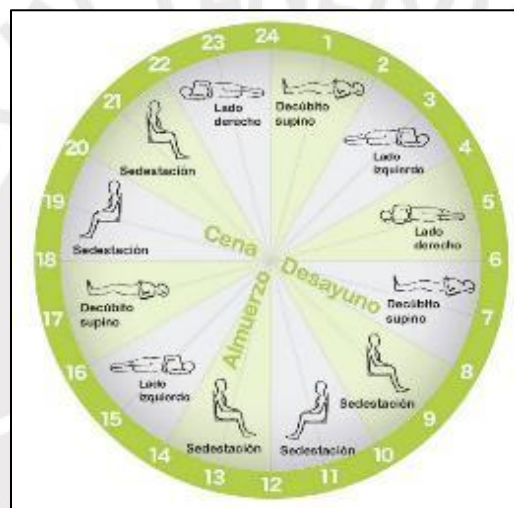


Figura 3.21: Rutina de posiciones para una persona con movilidad limitada.

Fuente: [www.pacientesycuidadores.com](http://www.pacientesycuidadores.com)

En relación a la rutina presentada, el uso del sistema se realizará por lo general cuando el paciente se encuentre en posición de sedestación, lo cual implica estar en esa posición 6 horas continuas entre las 8 am y 2 pm y un máximo de 10 horas por día, el resto del tiempo se usará para recargar la batería.

A continuación, se explicarán los diversos componentes que conforman parte de la distribución de energía, asumiendo que el tiempo de uso del sistema será de un tiempo promedio de 8 horas al día (tiempo promediado entre el uso mínimo y el uso máximo del sistema).

**a) Batería para la parte de reconocimiento de voz y la interfaz visual**

Según cálculos de consumo de corriente para la parte de reconocimiento de voz e interfaz de usuario realizados en el Anexo C, se tiene un consumo máximo de aproximadamente 278.5 mA. Para que la batería pueda alimentar a los componentes durante 8 horas debe tener una capacidad mínima de 2,228 mAh aproximadamente, por consiguiente, se optó por el uso de baterías LiPo (litio polimérico), por ser más comerciales y de tamaños reducidos, existiendo gran variedad en la capacidad de corriente que otorgan.

Como el sistema requiere 5V como mínimo, se elegirá una batería de 2S que proporciona 7.4 V aproximadamente para que luego sea regulada a 5V, la figura 3.22 muestra la batería escogida y la tabla 3.13 muestra algunas de sus características principales.



Figura 3.22: Batería para la parte de reconocimiento de voz e interfaz de usuario.  
Fuente: [www.hobbyking.com](http://www.hobbyking.com)

Tabla 3.13: Características técnicas de la batería para la parte de reconocimiento de voz e interfaz de usuario

Características Técnicas	
Capacidad	3000 mA
Voltaje de salida	7.4V (2S)
Descarga	20C constante
Carga	JST-XH
Peso	146 g
Medidas	88mm x 34mm x 25mm

Fuente: [www.hobbyking.com](http://www.hobbyking.com)

Este componente se ubicará dentro de la carcasa que recubre a la parte de reconocimiento de voz y en un compartimiento que lo separa de la circuitería como se puede apreciar en la figura 3.23. La batería posee un par de cables adicionales con el fin de brindar información sobre la cantidad de carga que posee en ese instante, esta es obtenida gracias a un medidor especial para baterías LiPo el cual se detallará en el siguiente punto.



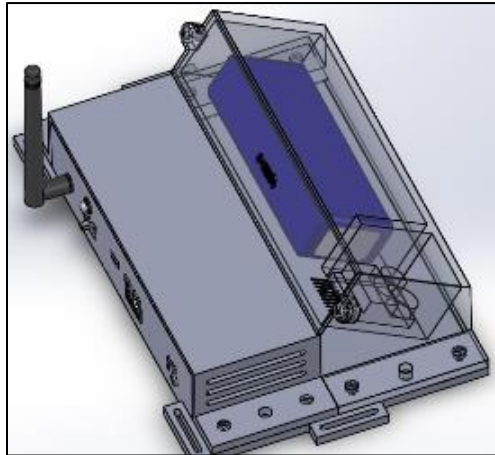


Figura 3.23: Ubicación de la batería LIPO del sistema de reconocimiento de voz.  
Fuente: Elaboración Propia

Los conductores que se usará para suministrar la energía proveniente de la batería LiPo serán de 24 AWG, según el cálculo de conductores realizado en el Anexo C.

***b) Medidor de carga con alarma para la batería***

Este dispositivo es esencial para poder alertar al usuario cuando exista una escasez de carga en la batería, y se proceda con la respectiva recarga. Cumple un papel importante respecto a la seguridad del sistema ya que una excesiva descarga de la batería puede ocasionar su deterioro y malograr la funcionalidad de la misma. El dispositivo elegido es reconocido en el mercado como “salva LIPO” de la marca Ecloud Shop, diseñado especialmente para medir el voltaje de las baterías LIPO, Li-ion, LiMn y Li-Fe, con la capacidad de alertar cuando la batería tenga una carga por debajo de lo configurado, encendiendo una luz roja y emitiendo gran alto sonido zumbador, el modelo se muestra en la figura 3.24, adicionalmente en la tabla 3.14 se muestran algunas características técnicas del dispositivo.

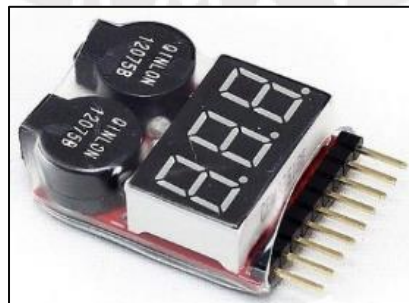


Figura 3.24: Salva LIPO, medidor de carga para baterías LIPO con alarma.  
Fuente: [www.thingnovation.com](http://www.thingnovation.com)



Tabla 3.14: Características técnicas del Salva LIPO

Características Técnicas	
Numero de celdas de la batería a medir	1S – 8S
Precisión de detección	$\pm 0.01$ V
Rango de visualización de voltaje	0.5V – 4.5V
Modelo	YCJ97
Peso	9 g
Medidas	40mm x 25mm x 11mm

Fuente: [www.thingnovation.com](http://www.thingnovation.com)

Como se puede apreciar en la figura 3.25 este dispositivo se ubicará junto al compartimiento de la batería LIPO permitiendo la conexión con los cables auxiliares de la batería y cercano al sentido de audición del usuario.

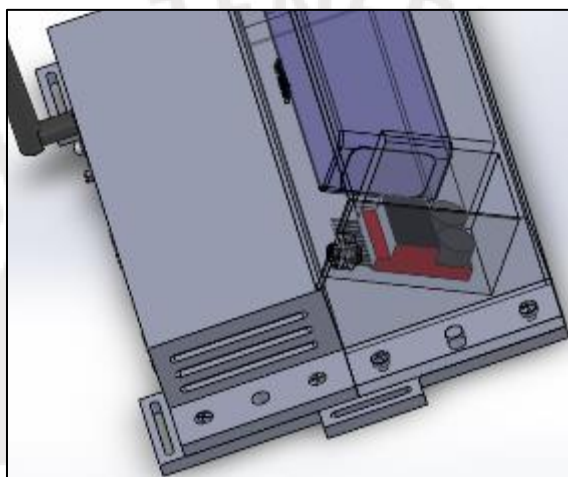


Figura 3.25: Ubicación del medidor con alerta para la batería LIPO

Fuente: Elaboración Propia

### c) *Regulador de voltaje para la parte de reconocimiento de voz*

Los elementos pertenecientes al reconocimiento de voz poseen voltajes de alimentación de 5V en su mayoría, y como se dispone de una batería que nos proporciona 7.4V, necesitamos una etapa de regulación esencial para la conversión a 5V que soporte la cantidad de corriente necesario para el funcionamiento. Como se calculó en el anexo C la cantidad de corriente que consumirá esta parte es de aproximadamente 0.3 A lo que nos indica que el regulador elegido deberá tener la capacidad de soportar alrededor de 1 A para evitar una sobrecarga y aumento de temperatura en el transistor. Se optó por el regulador LM7805 (figura 3.26) el cual regulará la tensión a 5V en corriente continua y con la capacidad de otorgar hasta 1 A, como no se supera la corriente del dispositivo no existe la necesidad de colocar disipador térmico. La tabla 3.15 muestra las características específicas de este dispositivo.

Tabla 3.15: Características técnicas del regulador LM7805

Características Técnicas	
Voltaje de entrada	7V – 25V
Voltaje de salida	5V
Corriente máxima de salida	1 A

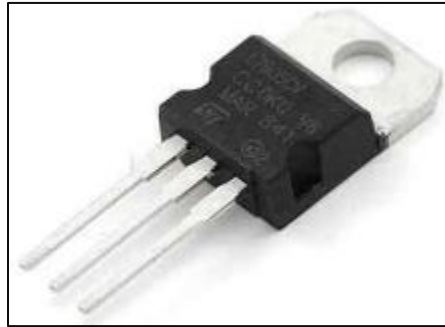
Fuente: [www.sparkfun.com](http://www.sparkfun.com)

Figura 3.26: Regulador LM7805

Fuente: [www.learningaboutelectronics.com](http://www.learningaboutelectronics.com)

La ubicación del dispositivo como se muestra en la figura 3.27, es sobre la placa impresa que forma parte del reconocimiento de voz acompañado por dos capacitores de 100uF y la salida conectado a las líneas de 5V del circuito.

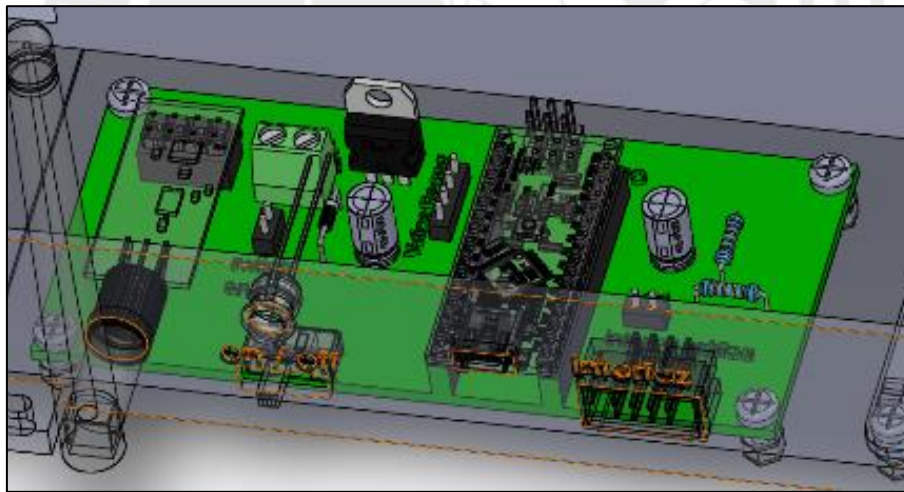


Figura 3.27: Ubicación del regulador de voltaje del reconocimiento de voz

Fuente: Elaboración Propia

#### ***d) Batería para el mecanismo robótico***

Según los cálculos de consumo de energía del mecanismo robótico realizados en el anexo C, se requerirá de una batería el cual produzca un voltaje de 24 V y suministre un mínimo de 3.5 A. Como el sistema debe funcionar por un lapso de 8 horas, la batería debe poseer como mínimo una capacidad de 30 Ah. Se optó por colocar 2 baterías en serie con una capacidad de

40 Ah cada una, la batería seleccionada es una de ácido-plomo de la marca KM la cual se muestra en la figura 3.28, adicionalmente la tabla 3.16 muestra algunas características del producto, toda la información sobre el producto se encuentra en el anexo I. En caso la silla del discapacitado sea eléctrica, se puede hacer uso de la fuente de alimentación de esta.



Figura 3.28: Batería 12V - 40Ah

Fuente: [www.articulo.mercadolibre.com.pe](http://www.articulo.mercadolibre.com.pe)

Tabla 3.16: Características técnicas la batería de 12 V acido - plomo

Características Técnicas	
Voltaje nominal	12 V
Capacidad	40 Ah (10 HR – 4 A)
Máxima descarga	320 A (5s)
Peso	12.6 Kg
Dimensiones	197mm x 165mm x 170mm

Fuente: [www.articulo.mercadolibre.com.pe](http://www.articulo.mercadolibre.com.pe)

#### e) *Regulador de voltaje para el controlador del mecanismo robótico*

Como se explicó en el punto anterior la alimentación del mecanismo robótico se basará en una batería la cual que suministra 24V, pero como el controlador necesita un voltaje de solo 5V para su funcionamiento, se adicionará una etapa de regulación de 24V a 5V. El requerimiento del dispositivo encargado de efectuar la regulación es que debe ser capaz de entregar más de 2 A, debido a que es el consumo mínimo que requiere el controlador PcDuino. Si bien se puede diseñar un circuito basado en reguladores de voltaje se optó por conseguir un componente de fábrica diseñado especialmente para poder soportar cargas mayores a 2 A, debido a que no son muy costosas y la eficiencia respecto a un diseño propio es mejor. Se optó por el regulador de corriente continua de la marca CPT (ver figura 3.29) ya que es capaz de soportar hasta 3 A y posee una salida de alimentación USB ideal para conectarlo al controlador PcDuino de una forma más consistente.



Figura 3.29: Regulador de voltaje 24V a 5V 3 A de la marca CPT  
Fuente: www.dhgate.com

Así mismo, el dispositivo se ubica cerca al controlador (ver figura 3.30) para que pueda alimentarlo y a su vez atornillado al estuche de control del brazo robótico para evitar su movimiento debido a vibraciones del sistema.

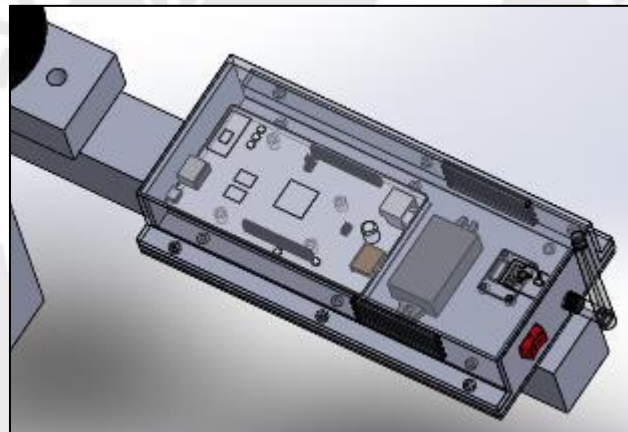


Figura 3.30: Ubicación del regulador de voltaje para el mecanismo robótico  
Fuente: Elaboración Propia

### 3.1.3 Diagramas esquemáticos del sistema de asistencia de control por voz

#### a) *Esquemático del controlador del reconocimiento de voz*

En la figura 3.31 se muestra el controlador Arduino NANO y su conexión con el módulo de transmisión inalámbrica (NRF24L01), el módulo de reconocimiento de voz e interfaz de usuario.

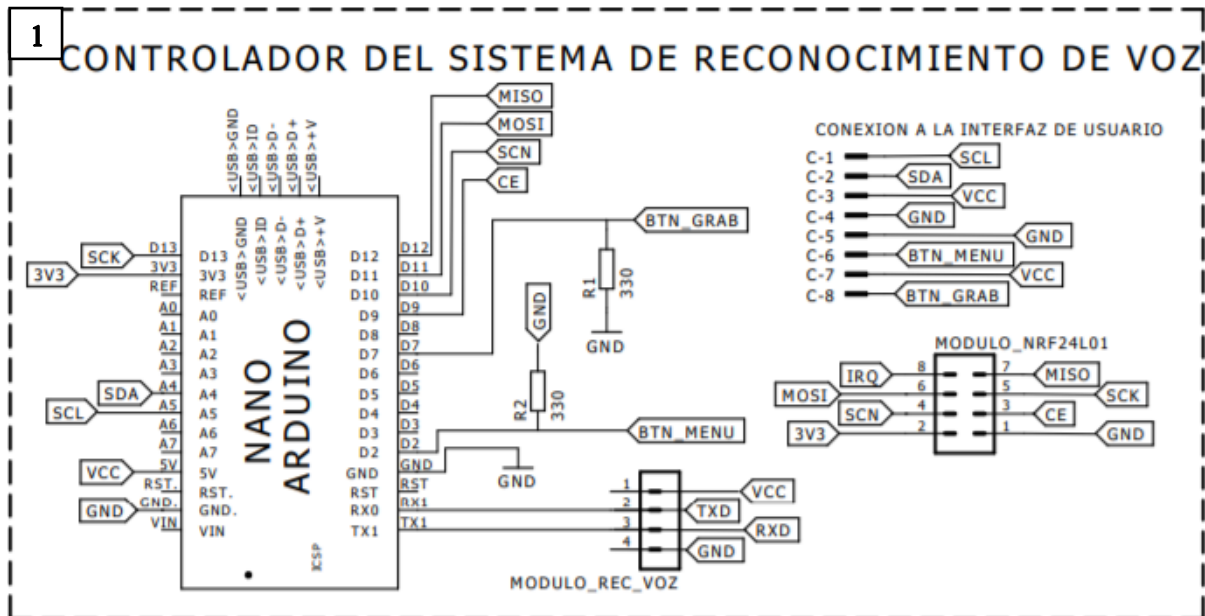


Figura 3.31: Esquemático del controlador del reconocimiento de voz  
Fuente: Elaboración Propia

Se adicionan 2 botones con el objetivo de poder colocar al sistema en modo configuración para la grabación de nuevos comandos de voz o caso contrario, procesar todos los comandos de voz recibidos. El botón denominado “BTN\_MENU” al ser pulsado parará el procesamiento y esperará la confirmación del usuario para entrar en modo configuración. El otro botón “BTN\_GRAB” al ser pulsado confirmará al sistema que se desea entrar en modo configuración, procediendo a grabar los 15 nuevos comandos de voz.

### *b) Esquemático de la interfaz de usuario*

El diagrama esquemático de la interfaz de usuario se muestra en la figura 3.32, este detalla la conexión de los pines del módulo I2C con la pantalla LCD, además estos dos dispositivos juntos a los dos pulsadores serán en conjunto la interfaz que permitirá visualizar los comandos reconocidos y poder configurar el sistema con nuevos comandos de voz.



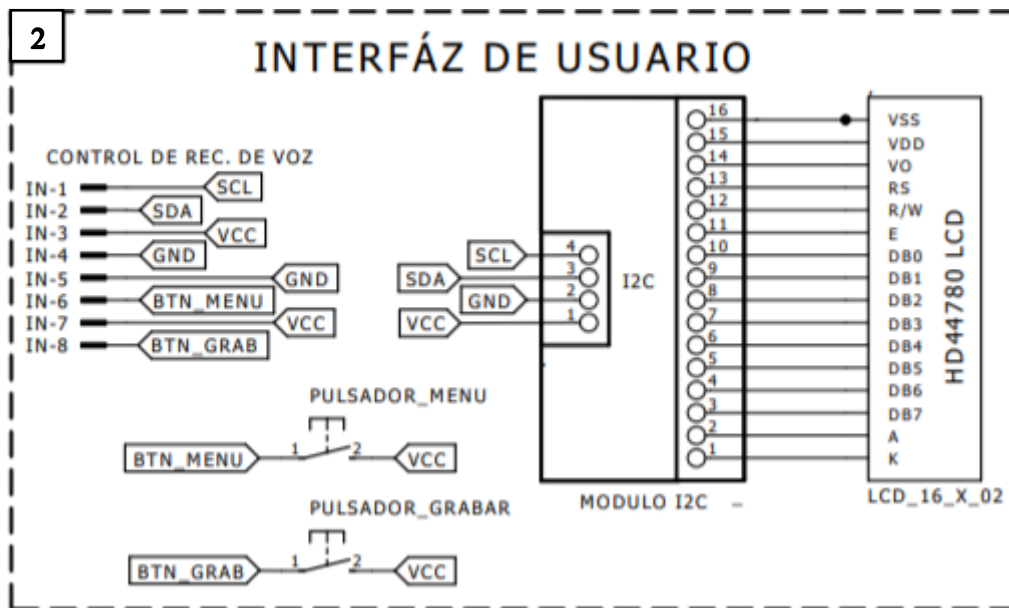


Figura 3.32: Esquemático de la interfaz de usuario  
Fuente: Elaboración Propia

*c) Esquemático del módulo reconocedor de voz*

La comunicación con el módulo reconocedor de voz se efectúa a través de los pines TX/RX del Arduino NANO, adicionalmente como se aprecia en la figura 3.33, posee 8 pines de los cuales 3 pines sirven para seleccionar un grupo de 5 comandos que se desean reconocer y los otros 5 pines restantes se usan como salidas, las cuales se colocan a 5V cuando el comando al que pertenece es reconocido. Por otro lado, posee un conector hembra tipo Jack donde se inserta el micrófono, o en caso no se desee posee como dos puertos para soldar las líneas del cable de un micrófono.

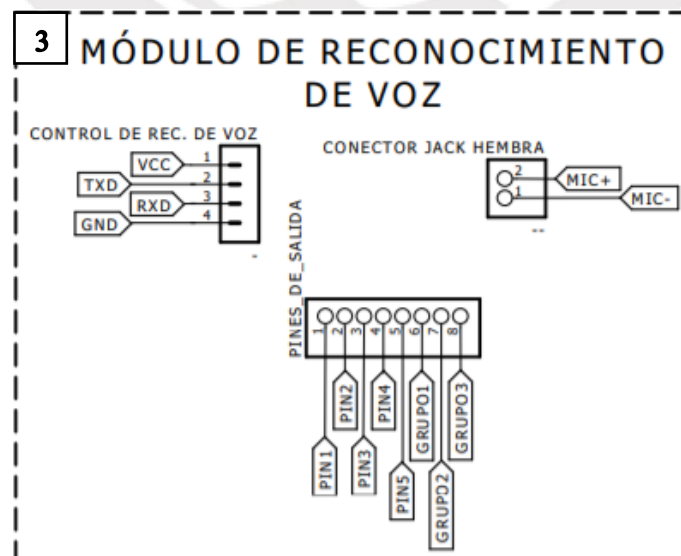


Figura 3.33: Esquemático del módulo reconocedor de voz  
Fuente: Elaboración Propia



*d) Esquemático de la etapa de regulación de voltaje*

La figura 3.34 muestra el circuito de regulación del voltaje para el reconocimiento de voz, como se puede apreciar los dos pines de entrada al circuito se conectarán a los bornes de la batería, quien energizará al sistema cuando se active el switch de encendido. Esta etapa contará en el componente LM7805 quien regulará el voltaje de entrada de 7.4V emitiendo como salida un voltaje de 5V a una corriente máxima de 1.5 A.

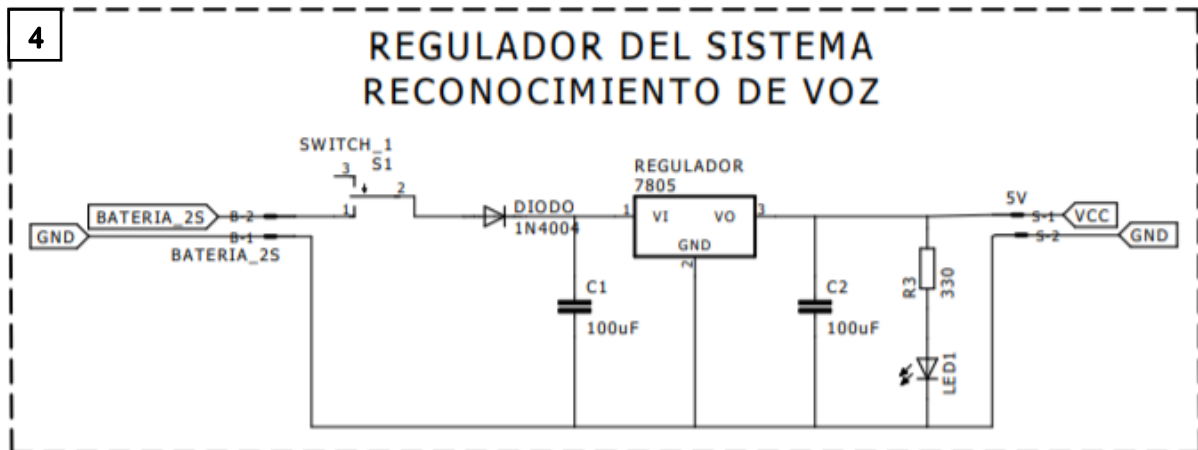


Figura 3.34: Esquemático del circuito de alimentación del sistema de reconocimiento de voz  
Fuente: Elaboración Propia

La etapa de regulación para alimentar al controlador del mecanismo robótico se efectúa mediante un regulador ya fabricado de la marca CPT, este se alimentará con un voltaje de 24V proveniente de una batería externa y producirá un voltaje de salida de 5V para alimentar al PcDuino (ver figura 3.35).

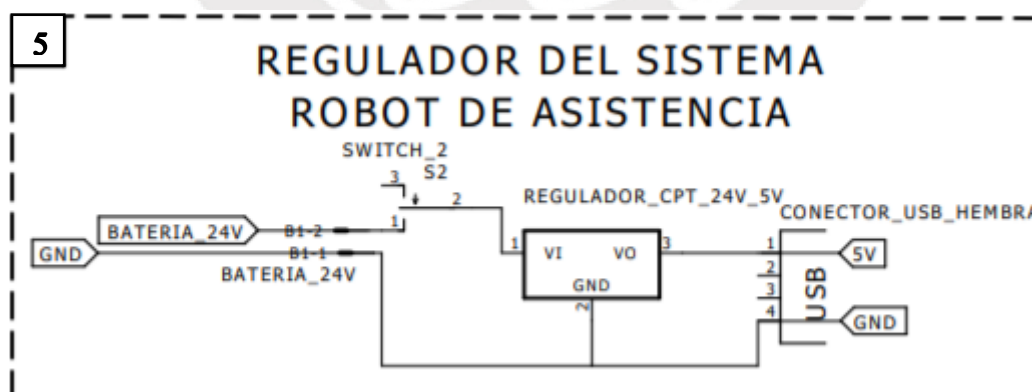


Figura 3.35: Esquemático del circuito de alimentación del mecanismo robótico  
Fuente: Elaboración Propia

e) *Esquemático de la etapa de control del mecanismo robótico*

Como se puede apreciar en la figura 3.36, la parte de control del mecanismo robótico la efectúa el controlador PcDuino quien tendrá el papel de recepcionar comandos de voz provenientes del módulo NRF24L01 (comunicación inalámbrica). Este controlador tendrá la capacidad de procesar el comando recibido y enviar una orden al robot MICO Arm para la ejecución del movimiento según sea el caso. El controlador se alimenta con 5V vía USB, el cual proviene de la salida del regulador.

Por otro lado, el controlador se comunicará con el robot MICO Arm vía USB por donde emitirá los comandos codificados para ejecución del movimiento. El esquemático también detalla las conexiones realizadas en el robot MICO, donde se puede observar que el robot se alimenta con 24V provenientes de una batería externa o de la batería de la silla eléctrica, según sea el caso.

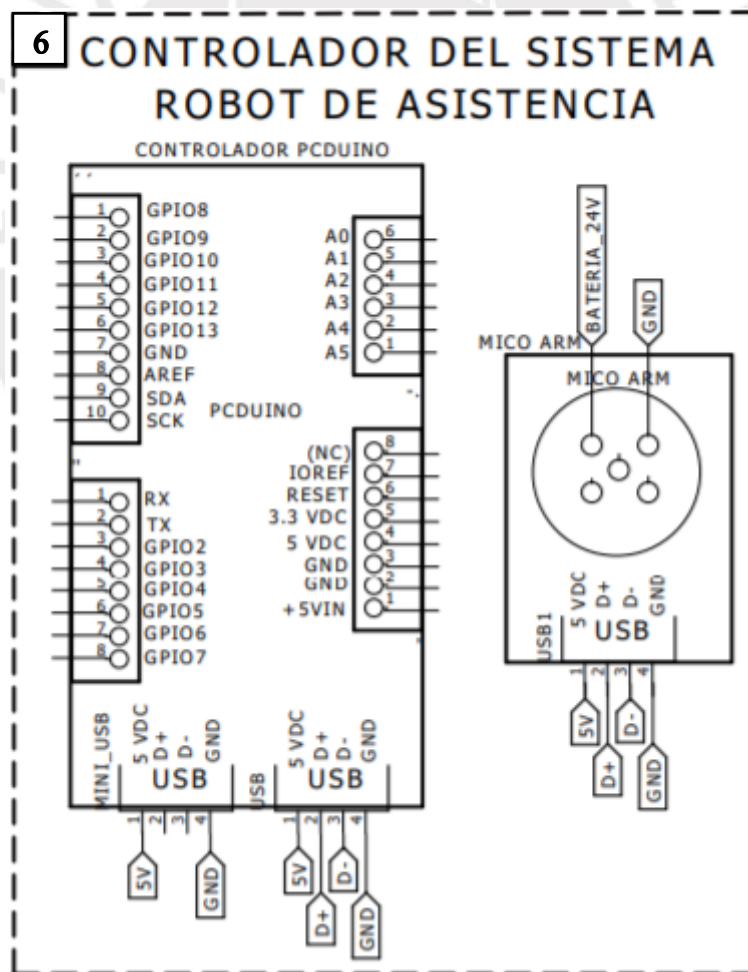


Figura 3.36: Esquemático de la etapa de control del mecanismo robótico  
Fuente: Elaboración Propia

## 3.2 Diseño Mecánico

En este subcapítulo se describirá los elementos mecánicos usados en el desarrollo del sistema robótico, se describirá el diseño de las carcasas y los soportes implementados en cada parte del sistema, quienes tienen el papel de mantener en una posición adecuada los elementos electrónicos y protegerlos contra impactos o del contacto accidental con algún líquido, asimismo se desarrollará la estructura que soporta los elementos del mecanismo robótico.

### 3.2.1 Material

Todas las carcasas que componen el sistema de asistencia de control por voz estarán compuestas de plástico, ya que ese material ofrece un diseño más ergonómico y sobre todo un precio más económico en su producción respecto a una fabricación en masa basado en planchas de aluminio. Por ello se decide la fabricación de la primera pieza prototipo en impresión 3D, con la proyección a futuro de una fabricación en inyección de plástico.

El material base que se usará para las impresiones será el ABS (Acrilonitrilo butadieno estireno) por ser un material de alta resistencia ante impactos a diferencia que el material PLA (Ácido Poliláctico), ofreciendo una larga duración de la pieza además de poseer una baja absorción de agua, lo cual beneficia para la protección de circuitos.

La figura 3.47 muestra el material de impresión ABS, el cual es un ligamento de larga longitud envuelto alrededor de un carrete, la tabla 3.18 muestra las principales propiedades del material.



Figura 3.47: Filamento de ABS  
Fuente: [www.makerbot.com](http://www.makerbot.com)

Tabla 3.18: Propiedades del filamento ABS

Características	
Temperatura de Fusión	100°
Densidad	1.04 g/cm <sup>3</sup>
Módulo de elasticidad (Young)	2.3 GPa
Elongación a la rotura	23% - 25%
Módulo de flexión	2.25 – 2.28 GPa
Resistencia a la flexión	60.6 – 73.1 MPa
Resistencia a la tracción	42.5 – 44.8 MPa

Fuente: [www.teststandard.com/data\\_sheets.htm](http://www.teststandard.com/data_sheets.htm)

### 3.2.2 Carcasas

La figura 3.37 muestra la carcasa que contiene toda la parte del reconocimiento de voz cuya ubicación será en el espaldar de la silla del discapacitado, adecuada para evitar interferencias en la señal por longitud larga del cable que conecta el micrófono con el módulo reconocedor de voz.

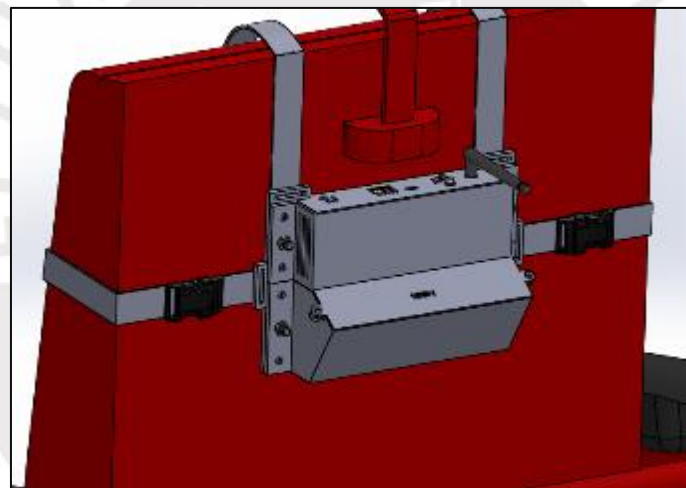


Figura 3.37: Carcasa del sistema de reconocimiento de voz  
Fuente: Elaboración Propia

Esta carcasa está compuesta por 4 elementos los cuales se ensamblan mediante uniones empernadas. Estos elementos se pueden apreciar en la figura 3.38, donde el elemento D corresponde a la base de la carcasa, el elemento C corresponde a la tapa de los componentes electrónicos, el elemento B corresponde al compartimiento donde se alojará la batería y el elemento A corresponde a la tapa del compartimiento de la batería.

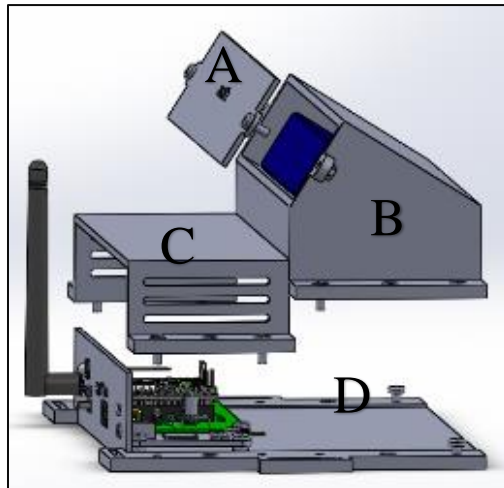


Figura 3.38: Elementos que conforman la carcasa del sistema de reconocimiento de voz

Fuente: Elaboración Propia

El compartimiento de la batería (elemento B) fue diseñado con el objetivo de que se pueda extraer la batería si necesidad de retirar toda la carcasa que protege al conjunto de circuitos, evitando a su vez la caída de la batería gracias a su ángulo de  $58^\circ$  respecto a la horizontal. Por otro lado el elemento C fue diseñado separadamente del elemento B para evitar sustraer toda la carcasa cuando se desee realizar un mantenimiento al conjunto de elementos electrónicos, adicionalmente posee unas aberturas en los costados para permitir la ventilación evitando una concentración de calor interna en los componentes lo cual puede deteriorar la carcasa con el paso del tiempo.

El elemento D fue diseñado para soportar el peso de todos los elementos que componen el subsistema de reconocimiento de voz, posee una pared vertical con aberturas por donde se conectarán todos los dispositivos externos y especialmente posee un diseño en base a 4 aberturas (figura 3.39) con la finalidad de instalar las correas que sostendrán al sistema en el espaldar de la silla del discapacitado. En caso se requiera una instalación sobre una estructura se le añadió 4 agujeros extras para poder empotrar el sistema a una plataforma plana. El análisis de la estructura sometida al peso de los componentes que lo componen se encuentra en el anexo D.

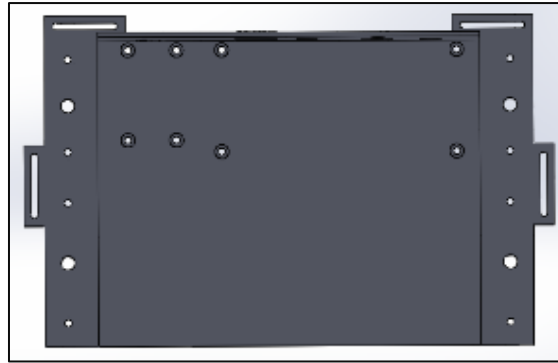


Figura 3.39: Vista frontal del Elemento D  
Fuente: Elaboración Propia

Las correas usadas para soportar la carcasa junto a los componente que lo conforman, son de nylon especiales para confeccion de mochilas con 25mm de ancho (figura 3.40), estas estan unidas al elemento D mediante una costura. Adicionalmente, se usarán hebillas de plastico para unir las correas y sujetadores antideslizantes para su ajuste a la silla del discapacitado.



Figura 3.40: Correas de nylon  
Fuente: [www.es.aliexpress.com](http://www.es.aliexpress.com)

En la figura 3.41 se puede apreciar la forma final que tendrá el elemento D junto al diseño de las correas de sujeción, las hebillas de plástico se ubican a los costados ya que en las dos correas superiores existe una mayor tracción.



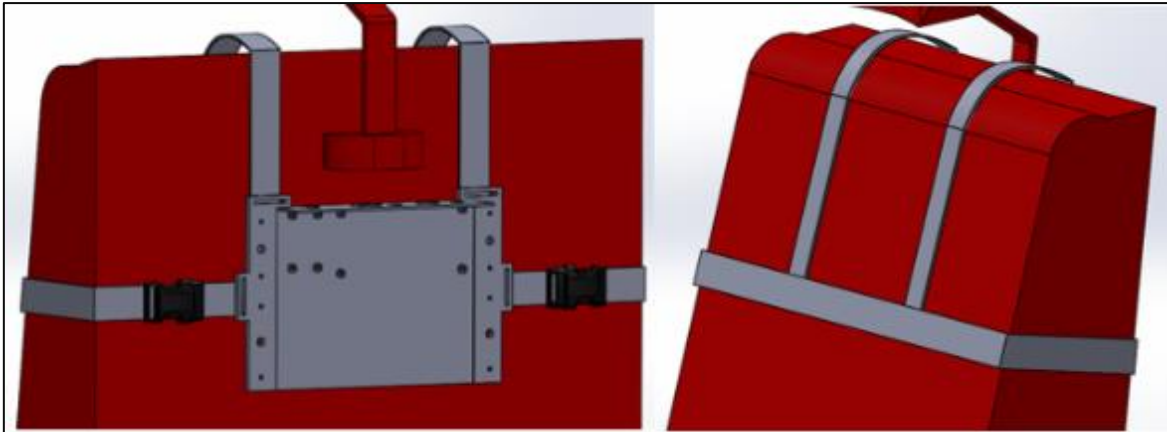


Figura 3.41: Sistema de sujeción al espaldar de la silla  
Fuente: Elaboración Propia

Para la parte del controlador del mecanismo robótico se diseñó una carcasa la cual se ubica cerca al brazo robótico, este se emperna a una estructura metálica que sostiene al sistema como se muestra en la figura 3.42. El objetivo de esta carcasa es proteger al PcDuino, al regulador y al módulo de comunicación inalámbrica ante impactos y contacto con líquidos. La carcasa esta compuesta de dos elementos como se aprecia en la figura 3.43, el elemento superior corresponde a la tapa del componente y el inferior a la base la cual estará empernada a la plataforma de la estructura de soporte. La unión entre la base y la tapa se realiza mediante tornillos facilitando la extracción de la tapa para el mantenimiento de los circuitos electrónicos. Adicionalmente la tapa posee aberturas en los costados para la ventilación en caso de aumento de temperatura en el regulador evitando deterioración del componente o la estructura que lo alberga.

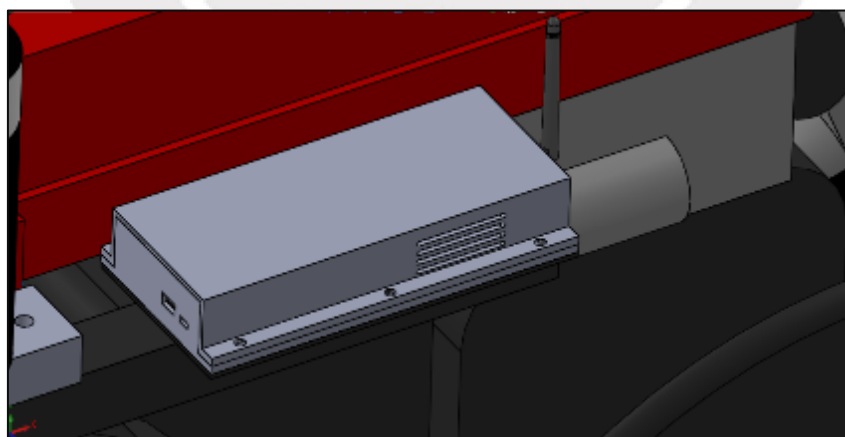


Figura 3.42: Carcasa del sistema de control del robot de asistencia  
Fuente: Elaboración Propia

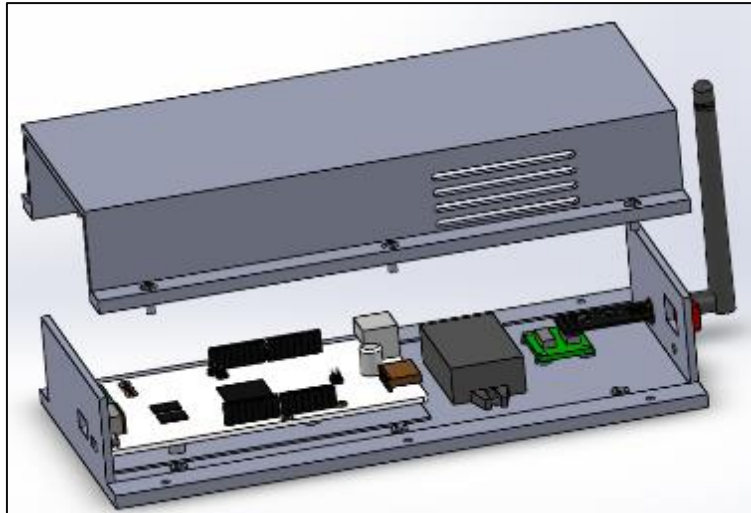


Figura 3.43: Elementos que conforman la carcasa del sistema de control del robot  
Fuente: Elaboración Propia

La carcasa del controlador y el robot MICO Arm son colocados sobre una estructura de tubos cuadrados de 3mm x 3mm que soportarán a los componentes. La estructura ya se encuentra realizada en las inmediaciones del laboratorio “CETAM” de la Pontificia Universidad Católica del Perú brindando soporte al brazo robótico MICO, adicionalmente sólo se plantea agregarle un tubo cuadrado laminado en caliente de 30mm x 30mm x 30.3cm que soportará la carcasa del sistema de control del robot y sus componentes. Este tubo estará soldado a una plancha de metal de 2mm de espesor con la finalidad de poder empernar la carcasa a la estructura (Ver figura 3.45). La figura 3.44 muestra el modelo final del soporte de todo el mecanismo robótico. Adicionalmente en el anexo D se realizó una simulación de los esfuerzos y deformaciones que existirán al añadir esta pieza en la estructura.

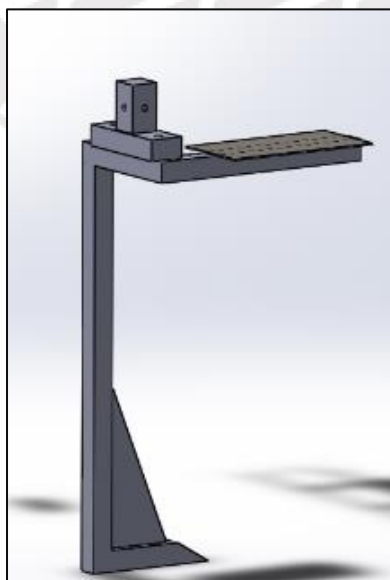


Figura 3.44: Soporte para la carcasa del sistema de control del robot y para el robot MICO  
Fuente: Elaboración Propia

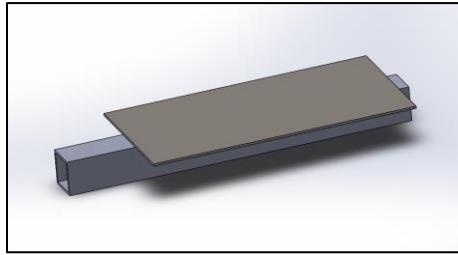


Figura 3.45: Soporte para la carcasa del sistema de control del robot  
Fuente: Elaboración Propia

La parte de interfaz de usuario también posee su propia carcasa la cual protege a los circuitos que lo componen de partículas externas o de algún impacto. Como se aprecia en la figura 3.46 posee dos elementos que lo componen los cuales se unen a través de tornillos y tuercas. En el elemento superior se atornilla la pantalla LCD para evitar movimientos por vibración y en el elemento inferior se atornilla la tarjeta que compone las conexiones y los botones de configuración.

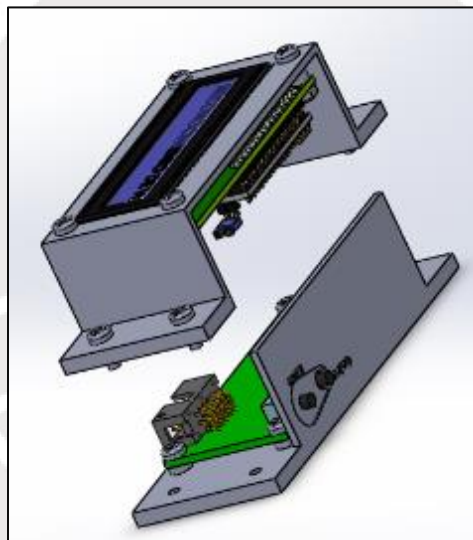


Figura 3.46: Elementos que conforman la interfaz de usuario

Así mismo para la sujeción de la carcasa de la interfaz de usuario se optó por usar un soporte de celular (figura 3.47), el cual posee la característica de ajustarse a la medida del dispositivo y poder acoplarse a la silla del usuario. El peso liviano del componente no generará problema alguno en el soporte.

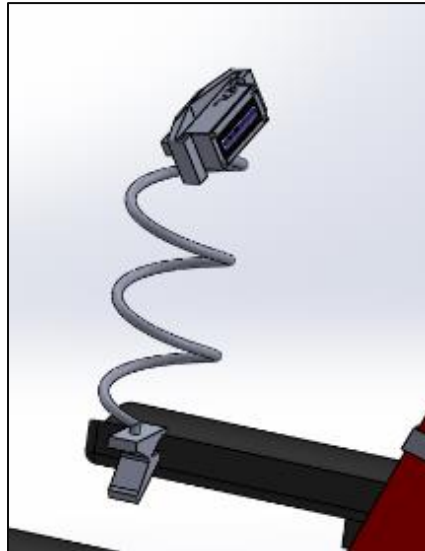


Figura 3.47: soporte de la interfaz de usuario  
Fuente: Elaboración Propia

A continuación, la tabla 3.17 muestra una relación de los planos de las piezas que se fabricarán en base a impresión 3d, estos se encuentran anexados al trabajo de tesis. El código que lleva cada plano indica primero el formato de hoja normalizada seguido del número de plano.

Tabla 3.17: Relación de planos

Numero	Tipo	Descripción	código
1	Despiece	TAPA BATERIA LIPO	A3-01
2	Despiece	TAPA PARA CIRCUITOS	A3-02
3	Despiece	BASE REC. DE VOZ	A3-03
4	Despiece	SOPORTE LIPO	A3-04
5	Despiece	TAPA CONTROL DEL ROBOT	A3-05
6	Despiece	BASE CONTROL DEL ROBOT	A3-06
7	Despiece	TAPA INTERFAZ DE USUARIO	A3-07
8	Despiece	BASE INTERFAZ DE USUARIO	A3-08
9	Sub ensamblaje	ESTUCHE INTERFAZ DE USUARIO	A3-09
10	Sub ensamblaje	ESTUCHE DE REC. DE VOZ	A3-10
11	Sub ensamblaje	ESTUCHE CONTROL DEL ROBOT	A3-11
12	Ensamblaje	SISTEMA DE ASISTENCIA DE CONTROL POR VOZ	A2-12

### 3.3 Diseño de control

Esta etapa se divide en dos partes, el diseño de control del reconocimiento de voz y el diseño de control del mecanismo robótico. A continuación, se explicarán el detalle de su contenido mediante la ayuda visual de diagramas de flujo, esenciales para la generación de su respectivo algoritmo de control.

### 3.3.1 Control del reconocimiento de voz

La parte de control del reconocimiento de voz se basa en la interacción conjunta entre el módulo reconocedor de voz, encargado de procesar las ordenes de voz que emite el usuario a fin de obtener un comando de voz ejecutable, y el controlador Arduino NANO, quien tendrá el papel de configurar al módulo reconocedor de voz y transmitir inalámbricamente el comando reconocido a la interfaz de usuario y al mecanismo robótico.

Cada proceso se explica haciendo uso de diagramas de flujo para poder comprender a detalle las funciones que realiza cada uno de estos componentes.

#### *a) Diagrama de flujo de control del Arduino NANO*

El diagrama se puede observar en la figura 3.48, este inicializa con la declaración de variables que se usarán en el algoritmo, configuración de los pines que se usarán como entradas y salidas, inicialización de la comunicación serial necesario para interactuar con el módulo reconocedor de voz, configuración inicial de la pantalla LCD y la configuración inicial del módulo de comunicación inalámbrica.

Luego de haber culminado la primera parte, se procede a mostrar un mensaje en la pantalla LCD dando a conocer al usuario la activación del reconocimiento de voz. Para que se pueda activar el módulo de reconocimiento de voz el controlador emite un comando serial en código ASCII, el cual es reconocido por el módulo, dando a proceder con la activación e iniciación del procesamiento de sonidos de voz que emite el usuario.

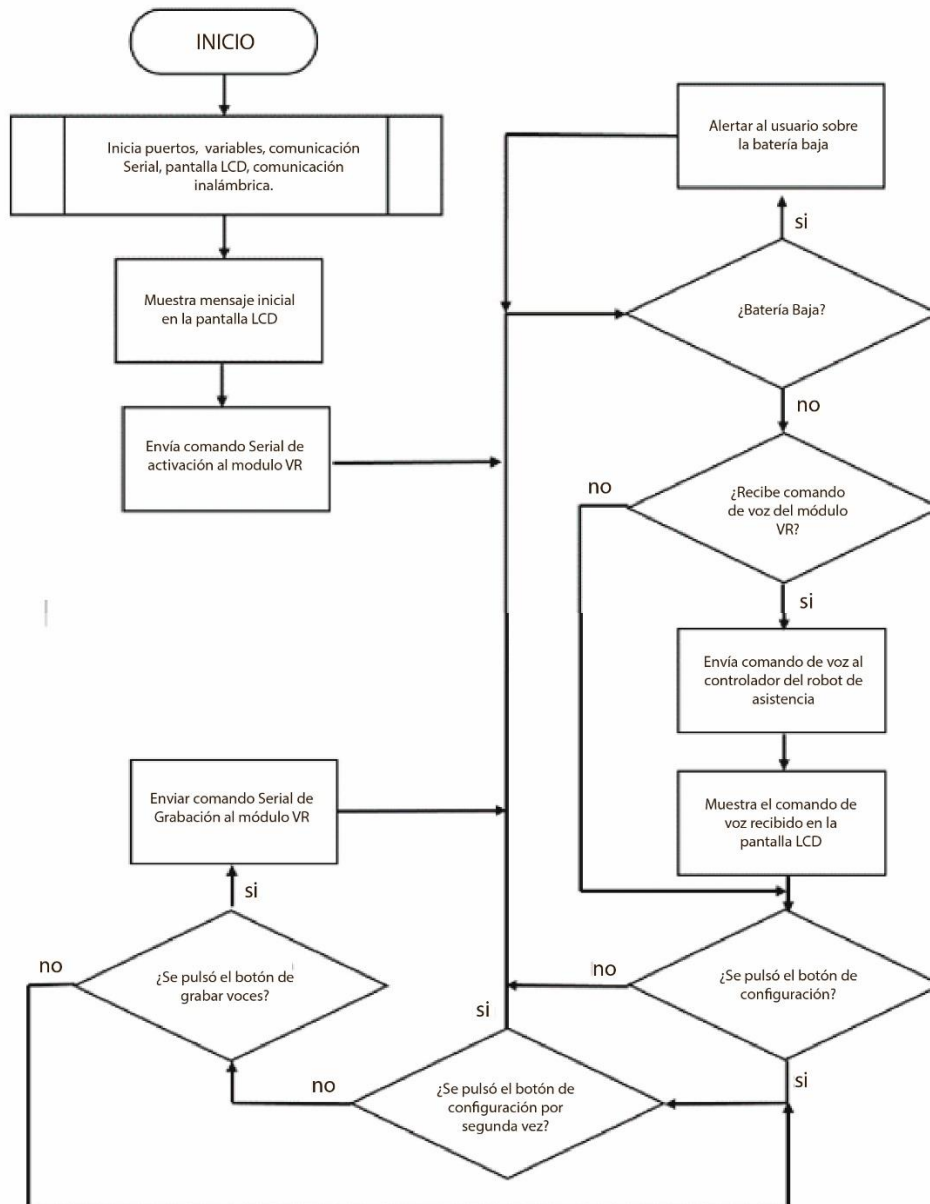


Figura 3.48: Diagrama de flujo del controlador del reconocimiento de voz  
Fuente: Elaboración propia

El primer paso que debe hacer el controlador antes de cualquier acción es preguntar si la batería está en buenas condiciones, si el resultado resulta negativo tendrá que alertar al usuario reclamando la recarga instantánea de la batería. Si el resultado resulta positivo el controlador permanecerá a la espera de algún código ASCII por el puerto serial proveniente del módulo reconocedor de voz, este código se traducirá según esta descrito en el manual (Anexo G) y se mostrará el mensaje respectivo por la pantalla LCD (el mensaje puede ser el comando reconocido o un mensaje de error).



Por otro lado, el controlador también consultará si se ha efectuado la pulsación del botón “MENU” (botón de configuración) ubicado en la interfaz, el cual detendrá el reconocimiento de voz y preguntará si se desea continuar reconociendo o proceder a grabar nuevos comandos de voz, la pulsación del segundo botón “Grabar” efectúa la confirmación de la grabación de nuevos comandos, donde luego pedirá al usuario generar dos veces el sonido de cada comando para evitar errores, configurando uno por uno hasta tener en total 15 comandos grabados exitosamente.

**b) Diagrama de flujo de control del módulo reconocedor de voz**

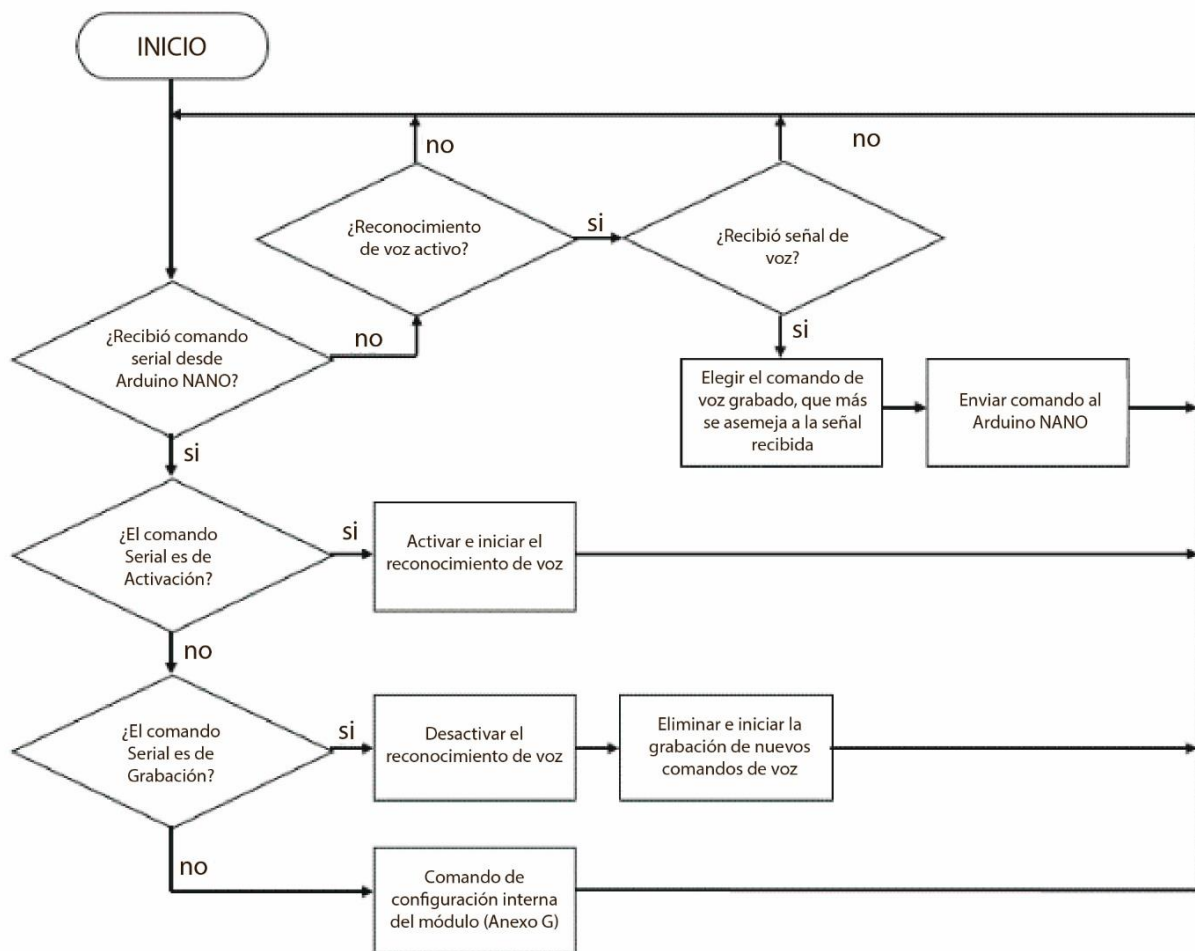


Figura 3.49: Diagrama de flujo de control del módulo reconocedor de voz

Fuente: Elaboración propia

El diagrama se muestra en la figura 3.49 explicando a detalle el funcionamiento del módulo reconocedor de voz. Como el módulo se comunica con al Arduino NANO mediante códigos ASCII por medio de una comunicación serial, esta espera en todo momento el código que le permita colocarse en un estado ya sea de reconocimiento de voz o de grabación de

nuevos comandos de voz. Cuando el módulo entra en un estado de reconocimiento de voz, este procede a captar toda señal de voz por medio del micrófono comparando cada señal con un grupo de 5 comandos previamente grabados (el módulo dispone de 3 grupos de 5 comandos cada uno). De esos 5 comandos elegirá el que posea un alto porcentaje de semejanza y emitirá el código ASCII que le pertenece a través del puerto serial según la ubicación del comando en numeración del 1 al 5.

### **3.3.2 Control del mecanismo robótico**

Esta etapa de control la efectúa el PcDuino quien será el encargado de recibir inalámbricamente el comando reconocido, para que luego proceda a enviarlo codificado hacia el robot MICO Arm vía USB con la finalidad de que este ejecute el movimiento ordenado por el usuario.

#### ***a) Diagrama de flujo de control del mecanismo robótico***

En la figura 3.50 se puede observar el proceso de control del PcDuino, empezando por la inicialización de las variables del algoritmo, configuración inicial de la comunicación serial con el robot MICO, configuración inicial con el modulo inalámbrico y la inicialización de las librerías de control del robot. El PcDuino estará siempre esperando la recepción de algún comando proveniente del sistema de reconocimiento de voz y cuando lo obtenga procederá a determinar qué acción ejecutara el robot de asistencia, una vez seleccionada la acción procederá a enviar la orden por el puerto USB 2.0.

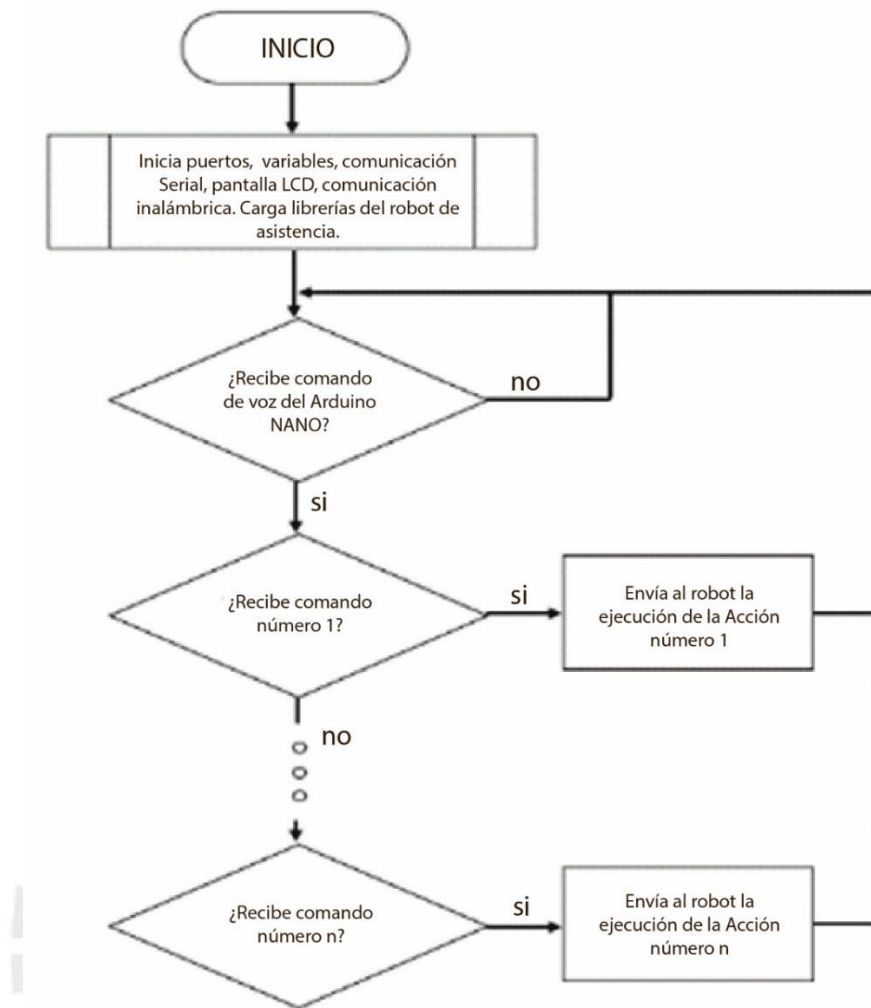


Figura 3.50: Diagrama de flujo de control del mecanismo robótico  
Fuente: Elaboración propia

## CAPÍTULO IV

### SIMULACIONES

El presente capítulo engloba el desarrollo del proceso experimental evaluando el performance y funcionamiento del sistema de asistencia de control por voz. Se analizará la integración total del sistema, simulando las etapas de reconocimiento de voz y control del brazo robótico, determinando la viabilidad en asistir a una persona con padecimiento de cuadriplejía.

#### 4.1 Requisitos

- El brazo robótico debe ser capaz de movilizarse a través del eje X, eje Y y eje Z, tener la capacidad de rotar sobre sus ejes, abrir y cerrar la pinza de sujeción, y todo mediante comandos de voz.
- El robot deberá de ir hacia un punto específico del espacio y realizar alguna acción deseada. Para implementar este punto se le pedirá al robot que agarre un vaso y lo lleve a la boca del usuario para que este pueda beber por sí mismo.

#### 4.2 Simulación de la etapa de reconocimiento de voz

Con el fin de cumplir con los objetivos descritos anteriormente, primero se procederá con implementar la etapa de reconocimiento de voz. Se conectará el Arduino NANO al módulo reconocedor de voz V2 a través del puerto serial, luego, se conectará la pantalla LCD junto al módulo I2C para poder observar las instrucciones que reconoce el sistema, en adición, se conectarán los botones para entrar en modo configuración. La figura 4.1 muestra el circuito implementado en un protoboard, con alimentación vía USB a través de un ordenador.

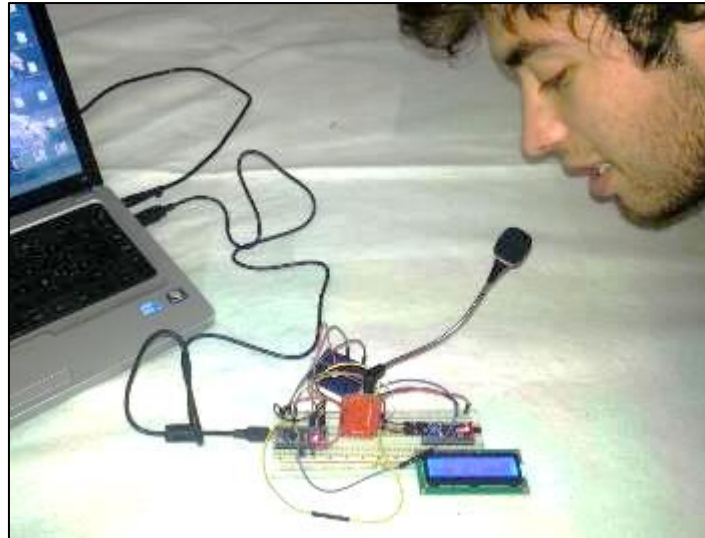


Figura 4.1: Conexiones para la simulación de la parte de reconocimiento de voz  
Fuente: Elaboración propia

El anexo G muestra una tabla de códigos ASCII que representan el lenguaje de comunicación del módulo reconocedor de voz, dispensables para su configuración y lectura de información que emite el dispositivo.

Como el módulo de reconocimiento de voz V2 de la marca Elechouse solo puede reconocer 5 comandos de voz a la vez, es decir, solo puede procesar un grupo de comandos a la vez, usaremos la siguiente lógica mostrada en la figura 4.2 para poder interactuar con la mayor cantidad posible de comandos.

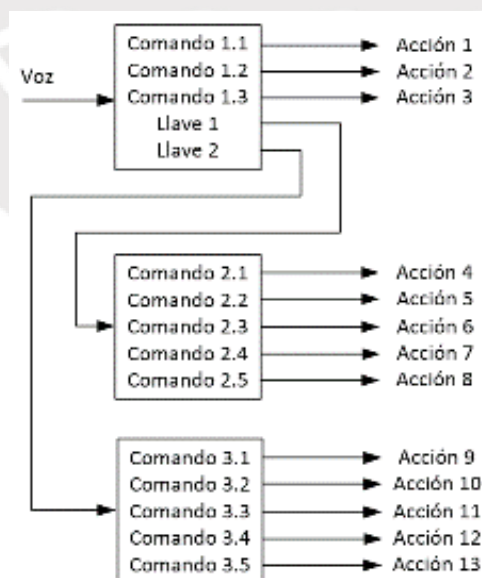


Figura 4.2: Lógica de reconocimiento de comandos  
Fuente: Elaboración propia

La llave 1 y la llave 2 serán comandos de voz que ocasionarán que el módulo de reconocimiento de voz cambie de grupo a procesar, es decir, si necesitamos que se ejecute la acción 10 debemos decir la palabra que activa la llave 2 y luego decir la palabra que activa el comando 10 emitiendo en total 2 sonidos de voz. A diferencia de las acciones 1, 2 y 3 solo requerirán de la emisión de un sonido de voz. Como se observa esta lógica aumenta de 5 a la utilización de 13 comandos de control.

La tabla 4.1 muestra los movimientos que ejecutará el brazo robótico, seguido del comando que se le asignará y el sonido de voz grabado por el usuario:

Tabla 4.1: Comandos para el control del brazo robótico.

<b>Acción</b>	<b>Comando</b>	<b>Comando</b>
Agarrar bebida	1.1	1.1
Tomar la bebida	1.2	1.2
Parar el movimiento	1.3	1.3
Cambiar al grupo 2	Llave 1	Llave 1
Cambiar al grupo 3	Llave 2	Llave 2
Movimiento en el eje z positivo	2.1	“Arriba”
Movimiento en el eje z negativo	2.2	“Abajo”
Movimiento en el eje x positivo	2.3	“Derecha”
Movimiento en el eje x negativo	2.4	“Izquierda”
Gira el brazo (“lateral orientation” de la figura 3.18) oscilando cada 180° de giro y a la espera del comando 1.3	2.5	“Gira”
Movimiento en el eje y positivo	3.1	“Adelante”
Movimiento en el eje y negativo	3.2	“Atrás”
Abrir pinza	3.3	“Abre”
Cerrar pinza	3.4	“Cierra”
Gira la pinza (“wrist rotation” de la figura 3.18) oscilando cada 180° de giro y a la espera del comando 1.3	3.5	“Rota”

Fuente: Elaboración propia

Adicionalmente, el circuito posee la conexión de 2 pulsadores, uno de ellos pausa el sistema y cuestiona si se desea re-grabar nuevos comandos. Para afirmar la consulta se usa el segundo botón, ordenando al sistema para que proceda con la grabación de nuevos sonidos para los comandos de voz.

Ya definido todo lo anterior se procede con la programación del controlador de la parte del reconocimiento de voz, el algoritmo implementado se encuentra en el anexo E, el cual contiene la comunicación con el módulo reconocedor de voz y la visualización en la pantalla LCD de cada comando reconocido.



Como se puede observar en el formato de pruebas de funcionamiento en el anexo K la simulación obtuvo resultados positivos, pudiéndose efectuar la grabación de todos los comandos de voz que el sistema reconocerá, siendo un total de 15 comandos de voz grabados sin problema alguno. Por otro lado, como se puede observar en la figura 4.3, el sistema reconoce los sonidos emitidos y los compara con los comandos previamente grabados, otorgando una efectividad del 100% de comandos reconocidos con éxito, lo cual puede disminuir con la presencia de interferencias sonoras del exterior.

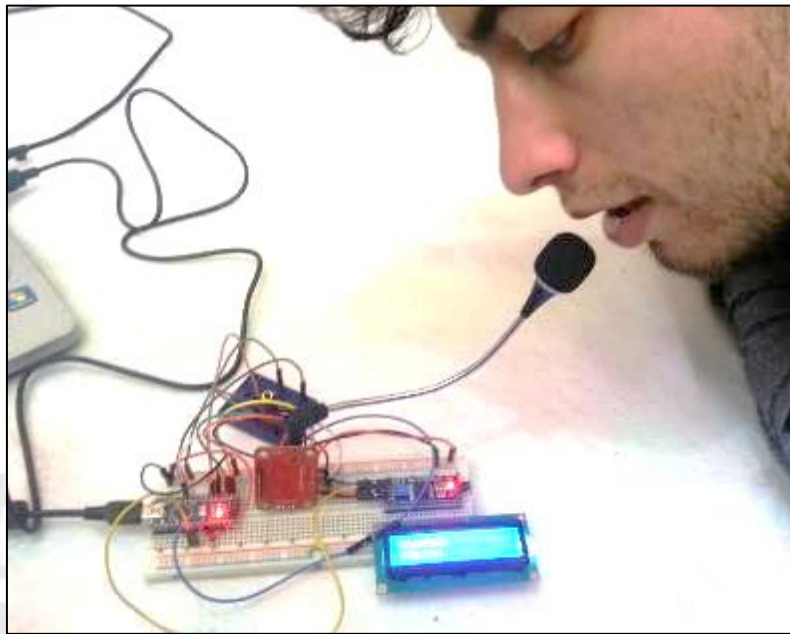


Figura 4.3: Conexiones para la simulación del subsistema de reconocimiento de voz  
Fuente: Elaboración propia

### 4.3 Simulación de la ejecución de movimientos del robot de asistencia

El siguiente paso para demostrar el funcionamiento del sistema es simular el proceso de control del brazo robótico MICO a través del controlador PcDuino y la ejecución de los movimientos requeridos. La explicación de la simulación se dividirá en dos partes: la primera parte se enfoca en la configuración previa que debe tener el brazo robótico MICO haciendo uso de su software “JacoSoft”, configuración necesaria para controlar al robot mediante una conexión USB. La segunda parte se enfoca en el control del robot por medio del controlador usando la conexión USB y el algoritmo implementado para simular un control óptimo del robot.

#### 4.3.1 Configuración inicial del brazo robótico MICO

El control del robot MICO se efectuó emulando los botones del joystick mediante códigos de programación gracias al uso de librerías del robot, en donde se vincula un comando

para cada botón del joystick efectuándose la configuración a través del JacoSoft como se aprecia en la figura 4.4, el cual es un software de ordenador que provee un control a través de una API (Interfaz de programación aplicada), este se usará a beneficio para poder controlar el robot a través de algoritmos en código C++.

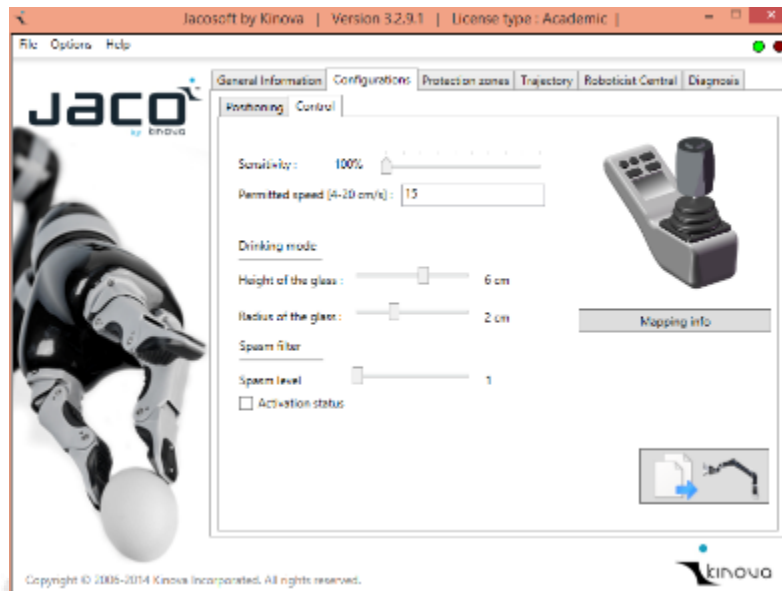


Figura 4.4: Software JACOSOFT para el control del robot MICO  
Fuente: JACOSOFT

Dado que el control del movimiento del robot se realizará de manera similar a la de un joystick primero se deberá configurar todos los movimientos en el software por medio de una API. Para ello en la ventana “Mapping” del software asignaremos a cada botón un respectivo comando, como se muestra en la figura 4.5 existen 6 tipos de movimientos del control, pero se espera poder realizar alrededor de 14 movimientos para ello se usará lo que el software denomina “modo”.

En el modo “A1” (parte superior de la figura 4.6) se configura todos los movimientos dirigidos en el plano X, Y, Z. En el modo “A0” (figura 4.5) se configura los movimientos de rotación respecto a cada eje y por último en el modo “A2” (parte inferior de la figura 4.6) se configura el movimiento de abrir y cerrar la pinza. El cambio de modo se efectúa por medio de los botones de la palanca del joystick, en este caso se efectuará a través del comando que lo representa.



Figura 4.5: Ventana “Mapping” para la configuración de los comandos que representaran a los botones del joystick  
Fuente: JACOSOFT

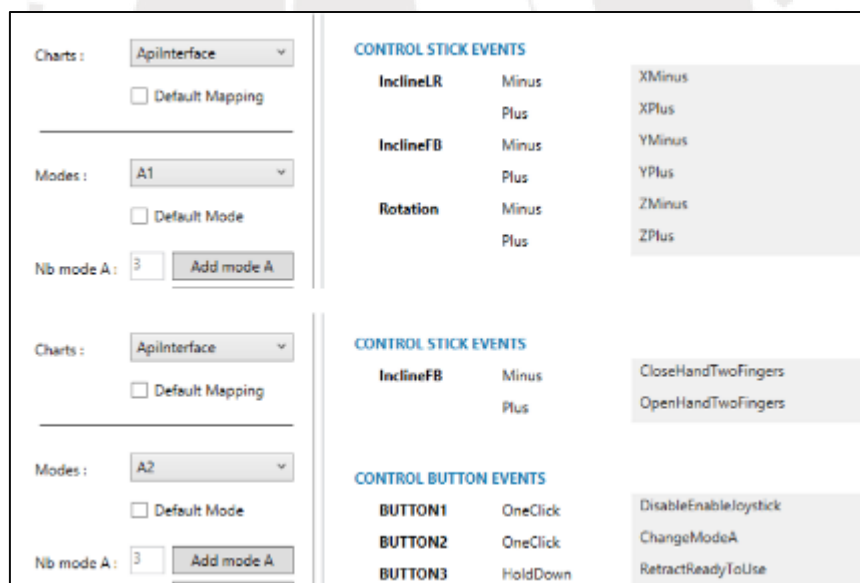


Figura 4.6: Configuraciones del modo “A1” y “A2”.  
Fuente: JACOSOFT

### 4.3.2 Control del robot vía USB

Esta etapa se refiere al control del brazo robótico a través de un controlador PcDuino, este controlador fue simulado a través de un ordenador, es decir, el ordenador hará el papel del

controlador haciendo uso de los mismos códigos y librerías que se implementan, estando también conectado al robot por medio de una conexión USB.

La figura 4.7 muestra la etapa de simulación del control del brazo robótico mediante el ordenador haciendo uso del algoritmo y librerías que se usarán en el controlador PcDuino. Como se puede apreciar en el formato de pruebas de funcionamiento en el anexo K, se obtuvo como resultado un correcto funcionamiento de cada movimiento del robot, sin presentarse inconveniente alguno en la comunicación de ambas partes.



Figura 4.7: Simulación del movimiento del robot MICO Arm mediante un ordenador  
Fuente: Elaboración propia

Orden de movimientos:

1. Movimiento en el eje x+.
2. Movimiento en el eje x-.
3. Movimiento en el eje y+.
4. Movimiento en el eje y-.
5. Movimiento de rotación del brazo.
6. Movimiento de rotación de la muñeca.
7. Abrir pinza.

8. Cerrar pinza.
9. Agarrar una bebida.
10. Llevar la bebida a la boca.

#### 4.4 Simulación de movimiento del MICO Arm por medio de un comando de voz

Finalizada las simulaciones de reconocimiento de voz y del control del brazo robótico, se procedió a simular la integración de todo el sistema con el fin de demostrar el funcionamiento en conjunto, como se puede apreciar en la figura 4.8 se conectó el controlador del sistema de reconocimiento de voz al ordenador, y el ordenador se conectó con el robot MICO, el algoritmo que se implementó en el Arduino NANO se encuentra en el anexo E, el cual tiene como función recibir el comando reconocido, mostrarlo en la pantalla LCD y enviarlo al ordenador. Por otro lado, también se muestra en el anexo F el código implementado para el control del brazo robótico según los comandos reconocidos por el módulo de reconocimiento de voz, este determinará qué tipo de acción efectuará el robot y procederá con enviar la orden del movimiento. La programación se efectuó en el software Visual Studio 2012 en el lenguaje C++, ya que es el lenguaje que acepta el módulo PcDuino, este contiene funciones como la recepción del comando vía Serial proveniente del Arduino NANO y su posterior elección del movimiento a ejecutar.

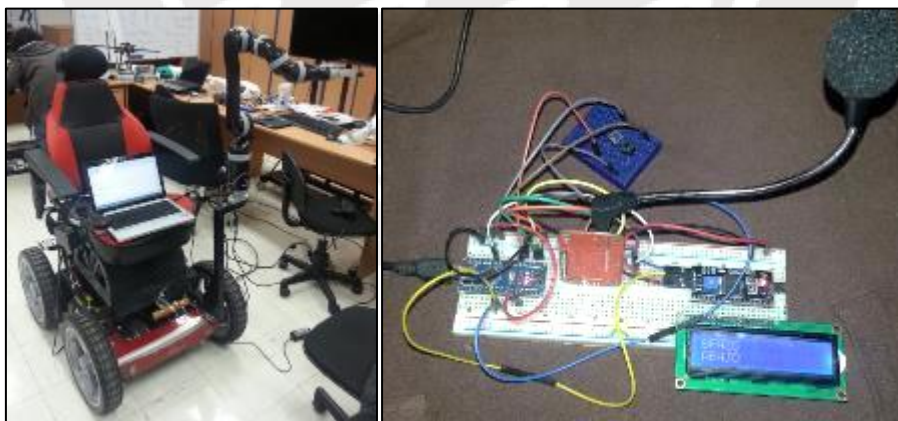


Figura 4.8: Simulación del movimiento del robot MICO a través del reconocimiento de voz

Fuente: Elaboración propia



## CAPÍTULO V

### PRESUPUESTO

En el presente capítulo se muestran los costos de los implementos que conforman el sistema de control por voz. La tabla 5.1 resume el costo de los componentes que conformaran la parte mecánica del sistema, la tabla 5.2 resume el costo de los componentes electrónicos y la tabla 5.3 resume el costo del robot de asistencia. Para obtener un detalle de donde fueron obtenidos los costos se adiciona en el anexo J las proformas obtenidas en las cotizaciones.

Tabla 5.1: Presupuesto de la parte mecánica

<b>Producto</b>	<b>Cantidad</b>	<b>Precio unitario (S/.)</b>	<b>Precio total (S/.)</b>
Impresión 3d	1	390.8	390.8
Tornillos y tuercas M3	100	0.1	10
Correa nylon de 25mm	4	6.6	26.4
Hebillas de plástico	10	2.9	29
Soporte de celular	1	25	25
Tubo cuadrado 30mm x 30mm x 6m	1	31.21	31.21
<b>Subtotal</b>			512.41
<b>IGV 18%</b>			92.23
<b>Total (S/.)</b>			604.64

Fuente: Elaboración propia

Tabla 5.2: Presupuesto de la parte electrónica

Producto	Cantidad	Precio unitario (S/.)	Precio total (S/.)
Módulo de reconocimiento de voz	1	120	120
Arduino NANO	1	30	30
Módulo NRF24L01	2	35	35
Pantalla LCD	1	14	14
Módulo I2C	1	12	12
LM7805	1	1	1
Regulador 24V- 5V CPT	1	112	112
Batería LiPo 2S 3000mah	1	67	67
Batería 12V 40Ah	2	257.4	514.8
Salva LiPo	1	20	20
Pulsadores	2	.5	1
PcDuino	1	240	240
		<b>Subtotal</b>	1,166.8
		<b>IGV 18%</b>	210.02
		<b>Total (S/.)</b>	1,376.8

Fuente: Elaboración propia

Tabla 5.3: Presupuesto del robot de asistencia

Producto	Cantidad	Precio unitario (S/.)	Precio total (S/.)
Robot MICO Arm	1	110385	110,385
		<b>Subtotal</b>	110,385
		<b>IGV 18%</b>	19,869.3
		<b>Total (S/.)</b>	130,254.3

Fuente: Elaboración propia

El costo de materiales del sistema, sin contar la silla de ruedas del discapacitado, sería de **S/. 132,235.74** nuevos soles.

## CONCLUSIONES

En el presente trabajo de investigación se pudo diseñar, adaptar, simular y validar el funcionamiento del sistema de control por voz orientado a personas que padecen de cuadriplejia, otorgando como resultado el cumplimiento de los objetivos planteados al inicio del trabajo de investigación. Asimismo, gracias a tener un diseño adaptable deja abierta la posibilidad de poder integrarse inalámbricamente con otro dispositivo robótico o actuador que complemente en la ejecución de otras tareas.

Con respecto a la capacidad que tiene el sistema de reconocer las ordenes de voz, se realizaron dos simulaciones, uno con la parte de reconocimiento de voz desconectado del brazo robótico y otra conectada para la ejecución del movimiento. El sistema logró reconocer los 15 comandos de voz programados (Ver Tabla 4.1 ) en ambas simulaciones, otorgando una capacidad de reconocimientos al 100% en un ambiente cuya interferencia de ruido externo era casi nulo.

Mediante la comunicación del ordenador con el robot Mico se logró ejecutar los 10 movimientos establecidos, movimiento en el eje “x+”, en el eje “x-”, en el eje “y+”, en el eje “y-”, rotación del brazo, rotación de la muñeca, abrir pinza, cerrar pinza, agarrar una bebida, llevar la bebida a la boca, sin presentar inconveniente alguno en cada primer intento, hecho que evidencia la óptima ejecución del movimiento ante un comando detectado.

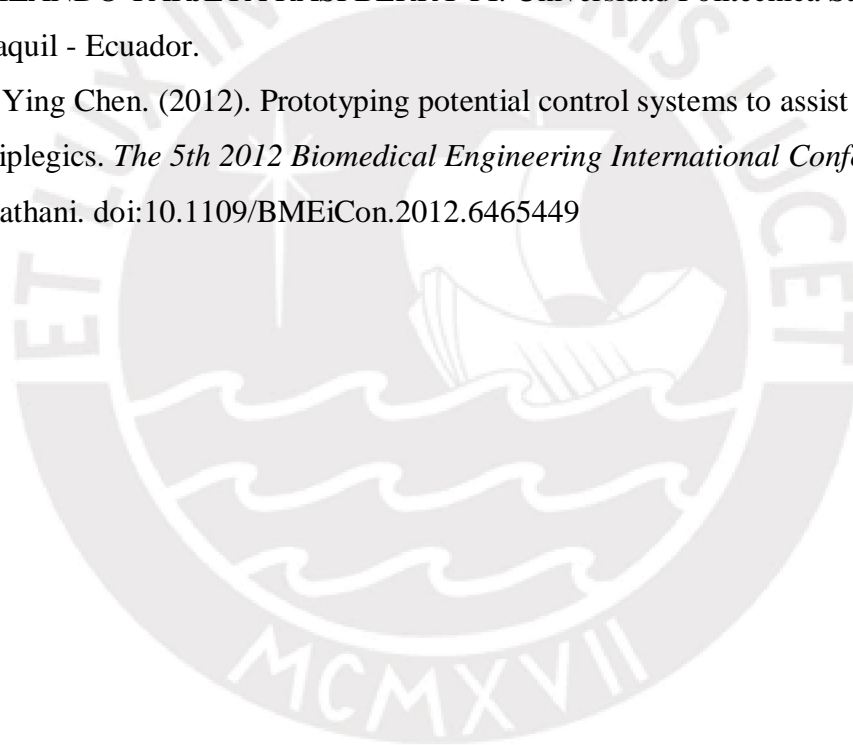
En la evaluación de funcionalidad de todo el sistema integrado (Ver capítulo 4.4 y anexo K), se simuló la acción que emplearía el discapacitado para poder tomar una bebida, este hecho resultó con total efectividad, logrando ejecutarse el movimiento inmediatamente después de mencionar el comando de voz.

El resultado de este trabajo de investigación, no solo otorgará una herramienta robótica a la persona que padece de cuadriplejia, sino también permitirá que esta pueda desarrollar independientemente sus actividades cotidianas, logrará un aumento en la autoestima del discapacitado, facilitará su inserción en la sociedad y mejorará sustancialmente su calidad de vida y la de sus familiares.

## BIBLIOGRAFÍA

- A. Jardón, J. C. Gonzáles, M. Stoelen, S. Martínez, C. Balaguer. (2009). ASIBOT assistive robot in a domestic environment. *2nd International Conference on Pervasive Technologies Related to Assistive Environments*. Corfu, Greece.
- de Figueiredo Carvalho, Z. M., Tirado Darder, J. J., Mulet Falco, F. V., Nuñez Hernandez, A. J., & Mulet, S. C. (s.f.). *Lesion Medular - Manual de Cuidados*. Alicante, España: Consejo de Enfermería de la Comunidad Valenciana.
- Doc Salud. (s.f.). *Inedito avance argentino: lograron que cuadripléjicos pudieran ponerse de pie*. Recuperado el 09 de Abril de 2016, de <http://www.docsalud.com/articulo/6161/in?dito-avance-argentino-lograron-que-cuadripl?jicos-pudieran-ponerse-de-pie>.
- G. Williams, Y. S. Lee, S. Ekanayake, P. N. Pathirana and L. Andriske. (2014). Facilitating communication and computer use with EEG devices for non-vocal quadriplegics. *7th International Conference on Information and Automation for Sustainability*. Colombo. doi:10.1109/ICIAFS.2014.7069604
- INEI. (2018). *Perú: Perfil Sociodemográfico, Informe Nacional*. Lima - Peru.
- Instruments, O. (s.f.). Sip and Puff Switch.
- Jauregui, E., & Iosu, A. (2013). Irisbond.
- KINOVA. (s.f.). *KINOVA*. Recuperado el 15 de Julio de 2019, de <https://www.kinovarobotics.com>
- Kirshblum SC, Burns SP, Biering-Sorensen F, Donovan W, Graves DE, Jha A, Johansen M, Jones L, Krassioukov A, Mulcahey MJ, Schmidt-Read M, Waring W. (2011). International standards for neurological classification of spinal cord injury (revised 2011). *The journal of spinal cord medicine*, 34. doi:10.1179/204577211X13207446293695
- Mauricio Villalba, I. L. (2008). Diseño de una silla de ruedas inteligente con integración multisensorial. Universidad Nacional de Huancavelica, Huancavelica - Peru.
- Ocaña Velásquez, J. D. (2013). Diseño de un sistema de control de desplazamiento de una silla de ruedas basado en el procesador de voz spce061a para el centro de educación y rehabilitación Sicomotriz Cersi – Chimbote. Universidad Privada Antenor Orrego, Trujillo - Peru.

- Organización Mundial de la Salud . (2011). *Informe mundial sobre la discapacidad*. Ginebra, Suiza.
- S. Udhaya kumar and V. M. Vinod. (2015). EOG based wheelchair control for quadriplegics. *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. Coimbatore. doi:10.1109/ICIIECS.2015.7193165
- Shepherd Center. (2011). Understanding Spinal Cord Injury.
- SMITH&NEPHEW. (11 de 05 de 2014). *Pacientes y Cuidadores*. Recuperado el 15 de Junio de 2019, de <http://www.pacientesycuidadores.com/como-prevenir-una-ulcera-por-presion/>
- Tomala Cuenca, D. G. (2018). SISTEMA DOMÓTICO CONTROLADO POR VOZ PARA PERSONAS CON DISCAPACIDAD EN EXTREMIDADES SUPERIORES, UTILIZANDO TARJETA RASPBERRY PI. Universidad Politécnica Salesiana, Guayaquil - Ecuador.
- Yaya Lu and Ying Chen. (2012). Prototyping potential control systems to assist complete quadriplegics. *The 5th 2012 Biomedical Engineering International Conference*. Ubon Ratchathani. doi:10.1109/BMEiCon.2012.6465449



**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

**FACULTAD DE CIENCIAS E INGENIERÍA**



## **Anexos**

**Sistema de asistencia para personas cuatripléjicas que permite la manipulación de un mecanismo robótico mediante la voz**

**TESIS PARA OPTAR EL TÍTULO DE INGENIERO MECATRÓNICO**

**AUTOR**

Joel Eduardo Escalante Zelada

**ASESOR:**

Antonio Lincoln Angulo Salas

**Lima, Marzo, 2020**



## Anexo A

### Evaluación técnico – económica

Cada concepto que ha sido presentado se evalúa con respecto a los demás colocando una serie de puntajes según criterio, de manera que un mayor puntaje representa un concepto más viable con respecto a lo técnico y a lo económico.

Los puntajes se realizarán en una escala de 0 al 4, donde 0 = No satisface, 1 = Aceptable a las justas, 2 = Suficiente, 3 = Bien, 4 = Muy bien.

#### Evaluación técnica:

Tabla A1: Evaluación técnica

Fuente: Propia

Valor Técnico (Xi)								
Proyecto: Sistema de asistencia para personas cuadripléjicas que permite la manipulación de un mecanismo robótico mediante la voz								
p: puntaje de 0 a 4 (Escala de valores según VDI 2225) 0 = No satisface, 1 = Aceptable a las justas, 2 = Suficiente, 3 = Bien, 4 = Muy bien (ideal) g: es el peso ponderado y se da en función de la importancia de los criterios de evaluación.								
Criterio de evaluación para diseños en fase de conceptos o proyectos								
Variantes de Concepto/Proyectos			Solución 1		Solución 2		Solución 3	
Nro	Criterios de evaluación	g	p	gp	p	gp	p	gp
1	Seguridad	4	2	8	3	12	2	8
2	Rapidez	3	3	9	2	6	2	6
3	Manipulación	4	1	4	3	12	1	4
4	Estabilidad	3	2	6	3	9	3	9
5	Ergonomía	2	2	4	3	6	3	6
6	Mantenimiento	2	1	2	2	3	2	4
7	Montaje	3	2	6	2	3	2	6
8	Peso	2	1	2	3	6	3	6
9	Adaptabilidad	2	1	2	3	6	3	6
Puntaje máximo $\Sigma p$ ó $\Sigma gp$		25	17	43	24	63	21	55
Valor Técnico Xi			0.4722	0.43	0.6667	0.63	0.5833	0.55
$X_i = \frac{p_1 + p_2 + p_3 + \dots + p_n}{np_{max}} = \frac{p_1 + p_2 + p_3 + \dots + p_n}{p_{max}} = \frac{\bar{p}}{p_{max}} \leq 1$ $X_i = \frac{g_1 p_1 + g_2 p_2 + g_3 p_3 + \dots + g_n p_n}{(g_1 + g_2 + \dots + g_n) p_{max}} \leq 1$								

Como se puede observar el concepto de solución 2 posee un puntaje de valor técnico más elevado respecto al resto cumpliendo con un 60 % de los criterios establecidos, lo cual lo vuelve la solución más viable en el aspecto técnico.

**Evaluación económica:**

Tabla A2: Evaluación económica  
Fuente: Propia

Valor Económico (Yi)								
Proyecto: Sistema de asistencia para personas cuadripléjicas que permite la manipulación de un mecanismo robótico mediante la voz								
p: puntaje de 0 a 4 (Escala de valores según VDI 2225) 0 = No satisface, 1 = Aceptable a las justas, 2 = Suficiente, 3 = Bien, 4 = Muy bien (ideal) g: es el peso ponderado y se da en función de la importancia de los criterios de evaluación.								
Criterio de evaluación para diseños en fase de conceptos o proyectos								
Variantes de Concepto/Proyectos			Solución 1		Solución 2		Solución 3	
Nro	Criterios de evaluación	g	p	gp	p	gp	p	gp
1	Costo de dispositivos	3	2	6	3	9	2	6
2	Costos de transporte	3	2	6	3	9	3	9
3	Vida útil	2	3	6	2	4	2	4
4	Facilidad de obtención	3	2	6	3	9	3	9
5	Costo de mantenimiento	2	3	6	2	4	2	4
Puntaje máximo $\Sigma p$ ó $\Sigma gp$		13	13	30	13	35	12	32
Valor Económico Yi			0.65	0.5769	0.65	0.67308	0.6	0.61538
$Y_i = \frac{p_1 + p_2 + p_3 + \dots + p_n}{np_{max}} = \frac{p_1 + p_2 + p_3 + \dots + p_n}{n p_{max}} = \frac{\bar{p}}{p_{max}} \leq 1$ $Y_i = \frac{g_1 p_1 + g_2 p_2 + g_3 p_3 + \dots + g_n p_n}{(g_1 + g_2 + \dots + g_n) p_{max}} \leq 1$								

Como se aprecia el concepto de solución 2 posee un puntaje de valor económico más elevado respecto al resto cumpliendo con lo cual se establece también a este concepto de solución como el más viable.

En resumen, la siguiente tabla muestra los puntajes de los 3 conceptos de solución, obtenidos en la evaluación técnico – económico con el fin de graficarlos para evaluar el concepto más óptimo.

Tabla A3: Resultados de la evaluación técnico - económica

Fuente: Propia

	Solución 1	Solución 2	Solución 3
Valor Técnico	0.43	0.63	0.51
Valor Económico	0.57	0.67	0.61



## Anexo B

### Comparación entre los módulos de reconocimiento de voz

El módulo de reconocimiento de voz, cumple una función muy importante dentro del desarrollo del sistema de asistencia, debido a que es el dispositivo que conlleva la parte de reconocer la voz para poder efectuar un control del robot usando la voz. Se investigó sobre la existencia de tres módulos creados con el fin de realizar esta función, la siguiente tabla muestra una comparativa entre los tres módulos con el fin de reconocer cuál de estos es el más viable para el desarrollo del sistema.

Tabla B1: Comparativa entre los módulos de reconocimiento de voz

Fuente: Propia

	Módulo de reconocimiento de voz V2 - Elechouse	Módulo de reconocimiento de voz EasyVR 3.0	Módulo ES6928P Voice Recognition
Ventajas	<ul style="list-style-type: none"><li>• Capaz de reconocer 15 comandos de voz</li><li>• Compatible con la plataforma Arduino</li><li>• 99% de precisión en reconocimiento en un entorno ideal</li><li>• Comunicación serial UART 5v TTL</li><li>• Su programación es sencilla y es flexible ante cambios en el algoritmo.</li><li>• Fácil obtención en el Mercado.</li></ul>	<ul style="list-style-type: none"><li>• Capaz de reconocer 32 comandos de voz</li><li>• Posee salida de altavoz donde se emula la voz humana</li><li>• Compatible con la plataforma Arduino.</li><li>• Posee shield de acoplo a un Arduino.</li><li>• Reconoce 9 tipos de lenguajes.</li><li>• Comunicación UART estándar compatible con los niveles CMOS / TTL</li></ul>	<ul style="list-style-type: none"><li>• Capaz de reconocer 4 comandos de voz</li><li>• Posee salida de altavoz para emular la voz humana</li></ul>
Desventajas	<ul style="list-style-type: none"><li>• No posee salida de altavoz.</li><li>• Solo reconoce a la voz que ha sido grabada previamente</li></ul>	<ul style="list-style-type: none"><li>• Su programación es algo compleja, posee variados comandos en la programación</li><li>• Difícil obtención en el Mercado.</li></ul>	<ul style="list-style-type: none"><li>• Difícil obtención en el Mercado, existen pocos módulos actualmente.</li></ul>

El resultado de esta comparativa se obtuvo al módulo de reconocimiento de voz de la marca Elechouse como el más viable para el sistema de asistencia, debido a que posee una fácil configuración gracias a su compatibilidad con la plataforma de Arduino. Por otro lado, también se optó por su elección debido a su fácil obtención en el mercado, ya que esto favorecerá al desarrollo del sistema durante la etapa de simulación.

## Anexo C

### Cálculos de consumo eléctrico

#### Consumo de corriente para la parte de reconocimiento de voz e interfaz de usuario

Se calcula en conjunto el consumo de corriente de estos dos sistemas debido a que poseerán una fuente de alimentación común.

El Arduino nano tiene un consumo máximo de 200mA, el superar esta carga puede generar daños al microcontrolador.

El módulo de reconocimiento de voz tiene un consumo máximo de 40 mA, por ello se tomará este valor suponiendo un consumo máximo de corriente.

El módulo de transmisión inalámbrica (NRFL24L01) consume como máximo 13.5 mA cuando se encuentra recibiendo a 2Mbps.

El módulo LCD en su trabajo conjunto con el módulo de bus I2C consumen en promedio 25mA, de donde la mayor parte la consume el led de fondo de la pantalla y el resto son consumos menores, tanto del módulo I2C y el integrado del LCD.

Como consumo total del reconocimiento de voz y el interfaz de usuario, se obtiene los siguientes valores:

Tabla C1: Consumo de corriente para la parte de reconocimiento de voz e interfaz de usuario  
Fuente: Propia

Componente	Consumo
Arduino NANO	200 mA
Módulo de reconocimiento de voz	40 mA
Módulo NRFL24L01	13.5 mA
Pantalla LCD y módulo I2C	25 mA
Total	278.5 mA

#### Consumo de corriente para el mecanismo robótico

En este punto se calcula en conjunto el consumo de corriente de todo el sistema del robot de asistencia, el cual engloba al controlador, al módulo de recepción inalámbrica y al robot MICO, quienes compartirán una misma fuente que les suministrara energía.

Para el controlador PcDuino, el consumo máximo que recibe este dispositivo es de aproximadamente 2000 mA. El consumo del módulo NRF24L01 como se explicó en el punto anterior consume alrededor de 13,5 mA.

El robot MICO Arm de la empresa Kinova según su hoja técnica (Anexo D) consume alrededor de 25 W a 24V y algunos picos de 1.5 A.

La siguiente tabla detalla el consumo total del sistema del mecanismo robótico:

Tabla C2: Consumo de corriente del mecanismo robótico  
Fuente: Propia

Componente	Consumo
PcDuino	2 A
NR24L01	0.0135 A
Robot MICO Arm	1.5 A
Total	3.5135 A

### Cálculos de conductores

Se calcula el diámetro del conductor eléctrico haciendo uso de la norma de la American Wire Gauge (AWG), donde se elegirá el conductor según soporte la máxima corriente establecida. La siguiente tabla muestra un extracto de la norma AWG.

Tabla C3: Escala de conductores según AWG  
Fuente: American Wire Gauge

AWG	Diámetro del conductor (mm)	Resistencia (Ohms/Km)	Corriente máxima (Amperios)
12	2.05	5.20	9.3
13	2.82	6.56	7.4
14	2.62	8.2	5.9
15	1.45	10.44	4.7
16	1.29	13.17	3.7
17	1.15	16.60	2.9
18	1.02	20.94	2.3
19	0.91	26.40	1.8
20	0.81	33.29	1.5
21	0.72	41.98	1.2
22	0.64	52.93	0.92
23	0.57	66.78	0.72
24	0.51	84.19	0.57



i) Alimentación del sistema de reconocimiento de voz.

Como se calculó el sistema de reconocimiento de voz se alimenta con una batería LIPO de 2S (7.4V) y un consumo máximo de 0.3 A, el conductor se selecciona mediante un factor de seguridad de 25%, según la tabla se deberá usar un conductor que soporte 0.37A, obteniendo el conductor recomendado de 24 AWG.

ii) Alimentación del sistema del robot de asistencia.

En este punto se alimentara el sistema a través de una batería de 24 voltios y su ingreso se dividirá en dos partes, uno de ellos se dirigirá directamente a la entrada del brazo robótico por el cual pasara un consumo máximo de 1.5 A, el otro se dirigirá al controlador del robot de asistencia por el cual pasara una corriente máxima de 2 A, con lo que se establece un consumo máximo de 3.5 A del sistema y con un factor de seguridad del 25% se estima un consumo de 4.4 A, con lo que se recomienda el uso de un conductor de 15 AWG



## Anexo D

### Análisis de esfuerzos

En este punto se muestra los análisis realizados a las diversas piezas y estructuras que son afectadas de manera crítica por alguna fuerza generada por una carga. Se emplea la simulación de elementos finitos que ofrece el software Autodesk Inventor, el cual también fue empleado en el diseño de las piezas del sistema.

#### **Análisis de esfuerzos en la base de la carcasa de la parte de reconocimiento de voz**

El elemento de la base de la carcasa posee cuatro agujeros en sus extremos, diseñados con la finalidad de instalar las correas que sostendrán los componentes del subsistema de reconocimiento de voz, el cual será colocado en el espaldar de la silla del discapacitado. Según su diseño se alcanza reconocer una parte crítica en la sujeción de las correas, las cuales serán sometidas a tracción, sobre todo en los dos agarres superiores debido a que sostendrán el mayor peso por estar en el eje vertical.

Las características de este componente son las siguientes:

Tabla D1: Características físicas de la base de la parte de reconocimiento de voz  
Fuente: Propia

Material	ABS
Densidad	$1.02 \text{ g/cm}^3$
Masa	134.17 g
Volumen	$131,5 \text{ cm}^3$

Así mismo, el peso de todos los componentes es de aproximadamente 600 gramos, lo que indica que la fuerza que deberá soportar la sujeción de las correas es de aproximadamente 5.8N. Con lo cual se sometió la pieza a un análisis de esfuerzos según Von Mises, generando como resultado un esfuerzo en el punto crítico de 2.16 MPa (ver figura 1), si bien el material tiene una resistencia a la tracción de aproximadamente 45MPa, el valor obtenido está muy por debajo del límite con lo que podemos concluir que la pieza no tendrá alguna falla y podrá efectuar sin problemas la función de sostener a todos los componentes en el espaldar de la silla. Adicionalmente se observa en el análisis de deformaciones (ver figura 2), que el aza de la sujeción de las correas sufrirá una deformación insignificante debido al peso de los componentes.

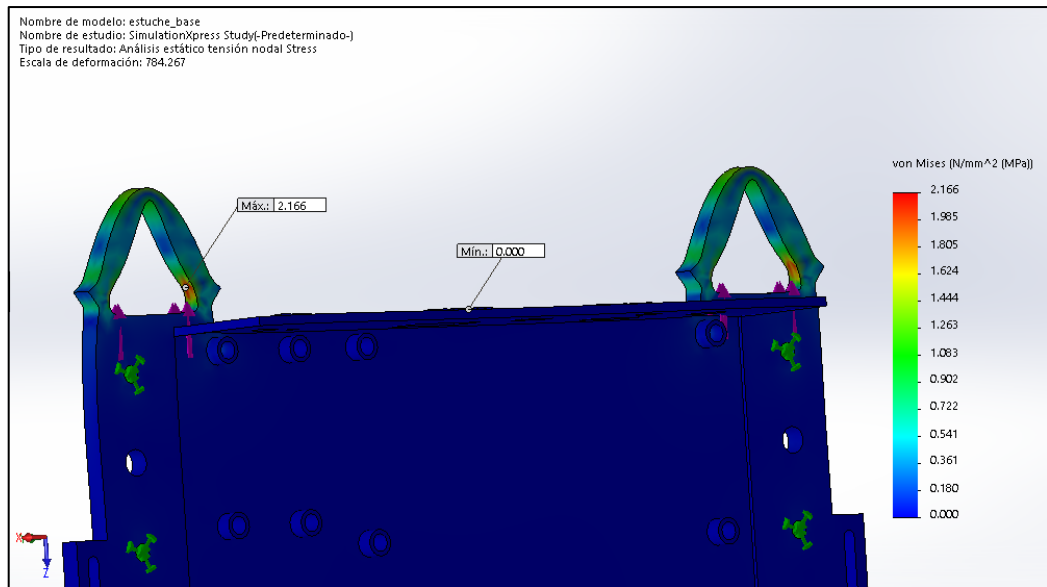


Figura D1: Análisis de esfuerzos según Von Mises de la base de la parte del reconocimiento de voz

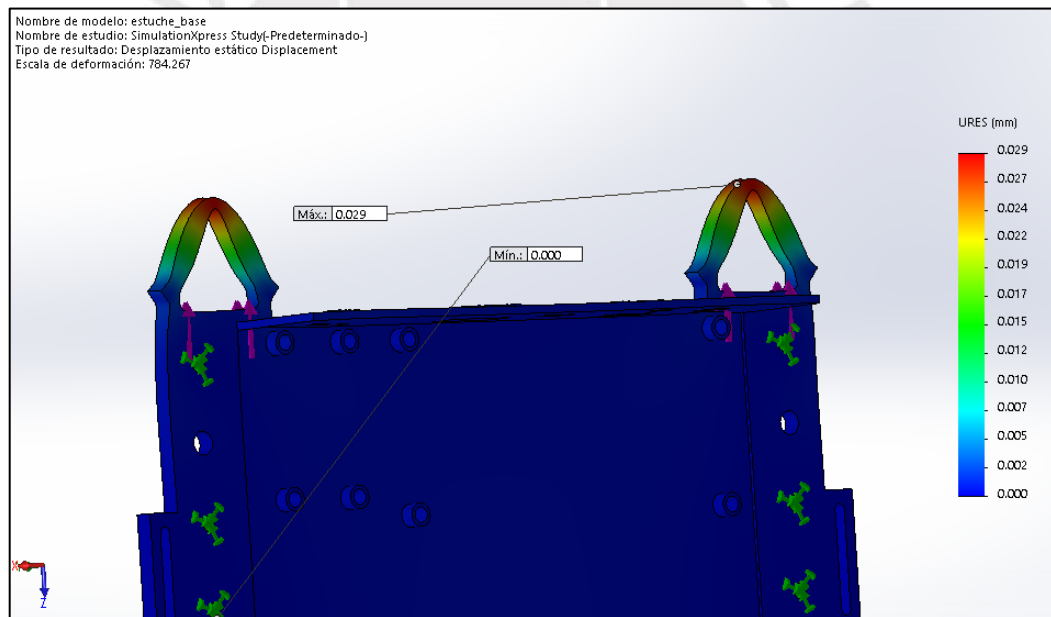


Figura D2: Análisis de desplazamiento en la base de la parte del reconocimiento de voz

### Análisis de esfuerzos en la estructura que soporta a la parte del mecanismo robótico

La estructura que soportará al brazo robótico MICO se encuentra implementada en las inmediaciones de la Pontificia Universidad Católica, pero carece de la plataforma que sostendrá al componente que posee el controlador del robot, por ello se diseñó la estructura con 30 centímetros de tubo cuadrado extra con una placa soldada en su parte superior para poder

empernar el módulo de control. Como la estructura implementada no posee falla alguna, solo se someterá a un análisis el material extra que sostendrá al módulo de control.

Las características de la estructura que se adicionara son los siguientes:

Tabla D2: Características físicas de la estructura que soporta el mecanismo robótico

Fuente: Propia

Material	Tubo cuadrado laminado en caliente
Densidad	7,870 kg/m <sup>3</sup>
dimensiones	30mmx30mmx30.3cm

Los componentes que conforman la parte del control del robot tienen un peso aproximado de 300 gramos, lo cual implica que en el soporte se efectuará una carga de casi 3N. La figura 03 muestra el análisis de esfuerzos según Von Misses que se ha realizado, obteniendo como resultado un esfuerzo máximo de 1.1 MPa en la unión entre la parte que se añadirá con la ya implementada, este valor es muy pequeño para poder efectuar algún tipo de falla en la estructura, con lo que no habrá problema alguno en la adición del material. La figura xx muestra la simulación del desplazamiento en la estructura añadida, con lo cual se obtiene un valor despreciable debido al bajo peso que posee en su totalidad el componente a sostener.

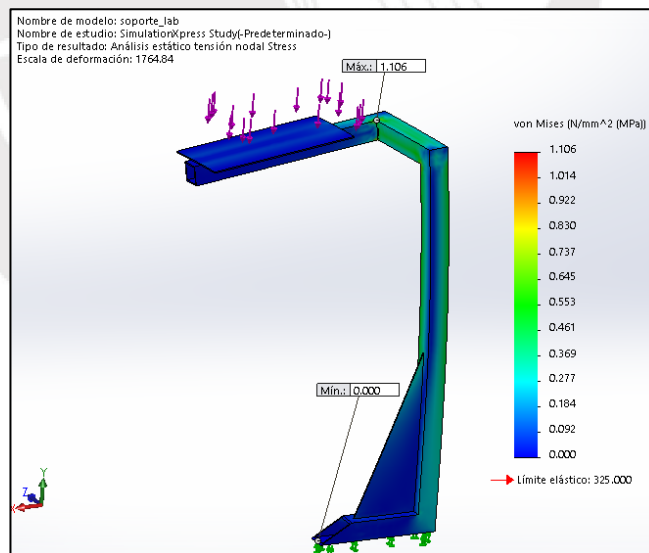


Figura D3: Análisis de esfuerzos según Von Misses del soporte para la parte del mecanismo robótico

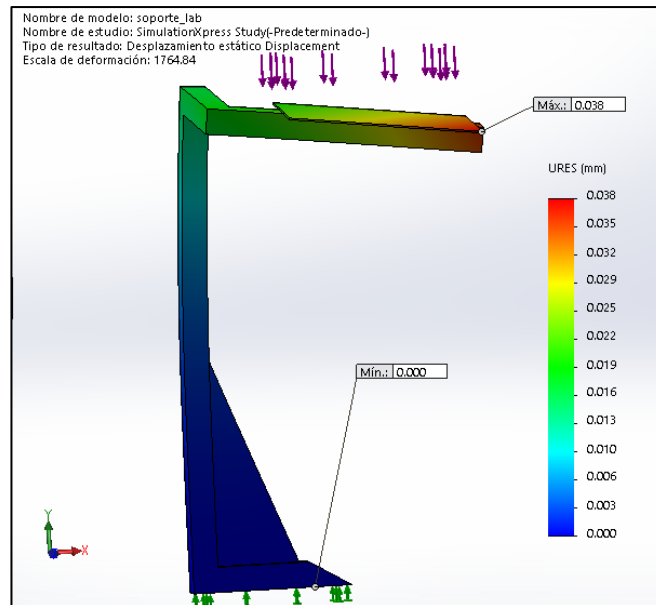
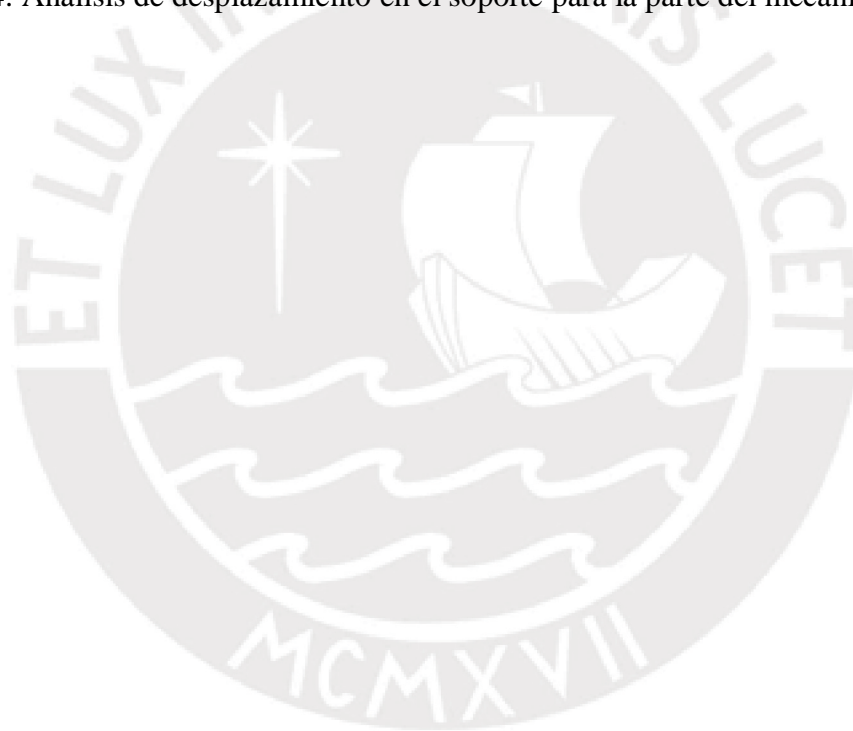


Figura D4: Análisis de desplazamiento en el soporte para la parte del mecanismo robótico



## Anexo E

### Algoritmo implementado en el controlador Arduino NANO para la parte de reconocimiento de voz

El algoritmo se desarrolló en la plataforma de Arduino usando su lenguaje de programación, adicionalmente, se usó librerías para la comunicación serial y para el control de la pantalla LCD. A continuación, se muestra el código implementado en la simulación.

```
#include <SoftwareSerial.h> //librería para la comunicación Serial
#include <Wire.h>           //librería para la comunicación con el módulo I2C y la
#include <LCD.h>           // pantalla LCD
#include <LiquidCrystal_I2C.h>

#define I2C_ADDR 0x27 //dirección de la pantalla LCD = 27

LiquidCrystal_I2C lcd(I2C_ADDR, 2, 1, 0, 4, 5, 6, 7); //configuración de pines I2C

//se inicializa variables
int estado = 0; //estado :: estado=0: reconociendo voz, estado=1:
int grupo = 1; // variable global para decir en qué grupo se encuentra
int boton_menu = 2; //botón de menú conectado al pin 2
int boton_menu_est = 0; //
int boton_grabar = 7; //botón de grabar conectado al pin 7
int boton_grabar_est = 0;

void setup() {

  Serial.begin(9600); // el módulo de reconocimiento de voz posee una comunicación
  // Predeterminada de 9600 baudios
  // el código serial.print se usará para comunicarse con el ordenador
  // via USB
  delay(100); // espera de 100ms

  //configuración inicial del módulo de reconocimiento de voz
  Serial.write(0xAA); // coloca al módulo en modo compacto, es decir, se comunicará
  Serial.write(0x37); // a través de códigos ASCII
  delay(200);
  Serial.write(0xAA); // inicia el procesamiento para comandos de voz del grupo 1
  Serial.write(0x21);

  delay(200); // espera 200ms

  // Configuración inicial del display con 16 caracteres y 2 líneas
  lcd.begin(16, 2); // se especifica el tipo de LCD
  lcd.setBacklightPin(3, POSITIVE); // se prende la luz de fondo
  lcd.setBacklight(HIGH);
  lcd.home (); // inicializamos LCD
  lcd.clear();
  delay(1000);
  lcd.setCursor(0, 0); // cursor al inicio
  lcd.print("RECONOCIMIENTO"); // escribimos mensaje de bienvenida
  lcd.setCursor(0, 1);
  lcd.print(" DE VOZ");
  delay(4000);
  //configuración inicial de los pines del controlador a usar
  pinMode(boton_menu, INPUT); // botón para entrar en modo Configuración
  pinMode(boton_grabar, INPUT); // botón para entrar en modo grabación
}

void loop() {
  boton_menu_est = digitalRead(boton_menu); //leemos estado del botón
```



```

boton_grabar_est = digitalRead(boton_grabar); //leemos estado del botón
byte comando = 0; //variable para la recepción de comandos del modulo
// Se espera a recibir un comando
if (Serial.available()) {
    comando = Serial.read(); //se graba temporalmente el comando recibido
    //se pregunta en que grupo se encuentra el sistema
    if (grupo == 1) { //grupo 1
        //se consulta que código recibió
        switch (comando) {
            case 0x11: //comando 1.1
                lcd.clear(); //se limpia la pantalla
                lcd.setCursor(0, 0); //se coloca cursor al inicio
                Serial.print("bebida"); //se envía la acción reconocida al ordenador
                lcd.print("AGARRAR"); //se muestra el comando reconocido en el LCD
                lcd.setCursor(0, 1);
                lcd.print("BEBIDA");
                break;
            case 0x12: //comando 1.2
                lcd.clear(); //se limpia la pantalla
                lcd.setCursor(0, 0); //se coloca cursor al inicio
                Serial.print("boca"); //se envía la acción reconocida al ordenador
                lcd.print("BEBIENDO"); //se muestra el comando reconocido en el LCD
                break;
            case 0x13: //comando 1.3
                lcd.clear(); //se limpia la pantalla
                lcd.setCursor(0, 0); //se coloca cursor al inicio
                Serial.print("para"); //se envía la acción reconocida al ordenador
                lcd.print("PARANDO"); //se muestra el comando reconocido en el LCD
                break;
            case 0x14: //Llave 1
                lcd.clear(); //se limpia la pantalla
                lcd.setCursor(0, 0); //se coloca cursor al inicio
                lcd.print("BRAZO"); //se muestra el comando reconocido en el LCD
                lcd.setCursor(0, 1);
                Serial.write(0xAA); //se cambia al grupo 2
                Serial.write(0x22);
                grupo = 2;
                break;
            case 0x15: //Llave 2
                lcd.clear(); //se limpia la pantalla
                lcd.setCursor(0, 0); //se coloca cursor al inicio
                lcd.print("MANO"); //se muestra el comando reconocido en el LCD
                lcd.setCursor(0, 1);
                Serial.write(0xAA); //se cambia al grupo 3
                Serial.write(0x23);
                grupo = 3;
                break;
        }
    }
    } else if (grupo == 2) { //grupo 2
        //se consulta que código recibió
        switch (comando) {
            case 0x21: //comando 2.1
                Serial.print("arriba"); //se envía la acción reconocida al ordenador
                lcd.print("ARRIBA"); //se muestra el comando reconocido en el LCD
                Serial.write(0xAA); //se regresa al grupo 1
                Serial.write(0x21);
                grupo = 1;
                break;
            case 0x22: //comando 2.2
                Serial.print("abajo"); //se envía la acción reconocida al ordenador
                lcd.print("ABAJO"); //se muestra el comando reconocido en el LCD
                Serial.write(0xAA); //se regresa al grupo 1
                Serial.write(0x21);
                grupo = 1;
                break;
            case 0x23: //comando 2.3

```

```

Serial.print("derecha"); //se envía la acción reconocida al ordenador
lcd.print("DERECHA"); //se muestra el comando reconocido en el LCD
Serial.write(0xAA); //se regresa al grupo 1
Serial.write(0x21);
grupo = 1;
break;
case 0x24: //comando 2.4
Serial.print("izquierda"); //se envía la acción reconocida al ordenador
lcd.print("IZQUIERDA"); //se muestra el comando reconocido en el LCD
Serial.write(0xAA); //se regresa al grupo 1
Serial.write(0x21);
grupo = 1;
break;
case 0x25: //comando 2.5
Serial.print("gira"); //se envía la acción reconocida al ordenador
lcd.print("GIRANDO"); //se muestra el comando reconocido en el LCD
Serial.write(0xAA); //se regresa al grupo 1
Serial.write(0x21);
grupo = 1;
break;
}

} else if (grupo == 3) { //grupo 3
//se consulta que código recibió
switch (comando) {
case 0x31: //comando 3.1
Serial.print("adelante"); //se envía la acción reconocida al ordenador
lcd.print("ADELANTE"); //se muestra el comando reconocido en el LCD
Serial.write(0xAA); //se regresa al grupo 1
Serial.write(0x21);
grupo = 1;
break;
case 0x32: //comando 3.2
Serial.print("atrás"); //se envía la acción reconocida al ordenador
lcd.print("ATRAS"); //se muestra el comando reconocido en el LCD
Serial.write(0xAA); //se regresa al grupo 1
Serial.write(0x21);
grupo = 1;
break;
case 0x33: //comando 3.3
Serial.print("abriendo"); //se envía la acción reconocida al ordenador
lcd.print("ABRIENDO"); //se muestra el comando reconocido en el LCD
Serial.write(0xAA); //se regresa al grupo 1
Serial.write(0x21);
grupo = 1;
break;
case 0x34: //comando 3.4
Serial.print("cerrando"); //se envía la acción reconocida al ordenador
lcd.print("CERRANDO"); //se muestra el comando reconocido en el LCD
Serial.write(0xAA); //se regresa al grupo 1
Serial.write(0x21);
grupo = 1;
break;
case 0x35: //comando 3.5
Serial.print("gira"); //se envía la acción reconocida al ordenador
lcd.print("GIRANDO"); //se muestra el comando reconocido en el LCD
Serial.write(0xAA); //se regresa al grupo 1
Serial.write(0x21);
grupo = 1;
break;
}
}
} else if (boton_menu_est==HIGH){ //botón menú presionado
while (boton_menu_est==HIGH){ //loop para evitar presión prolongada del
Serial.println("press 1"); //pulsador
boton_menu_est=digitalRead(boton_menu); //leemos estado del botón
}
}
}

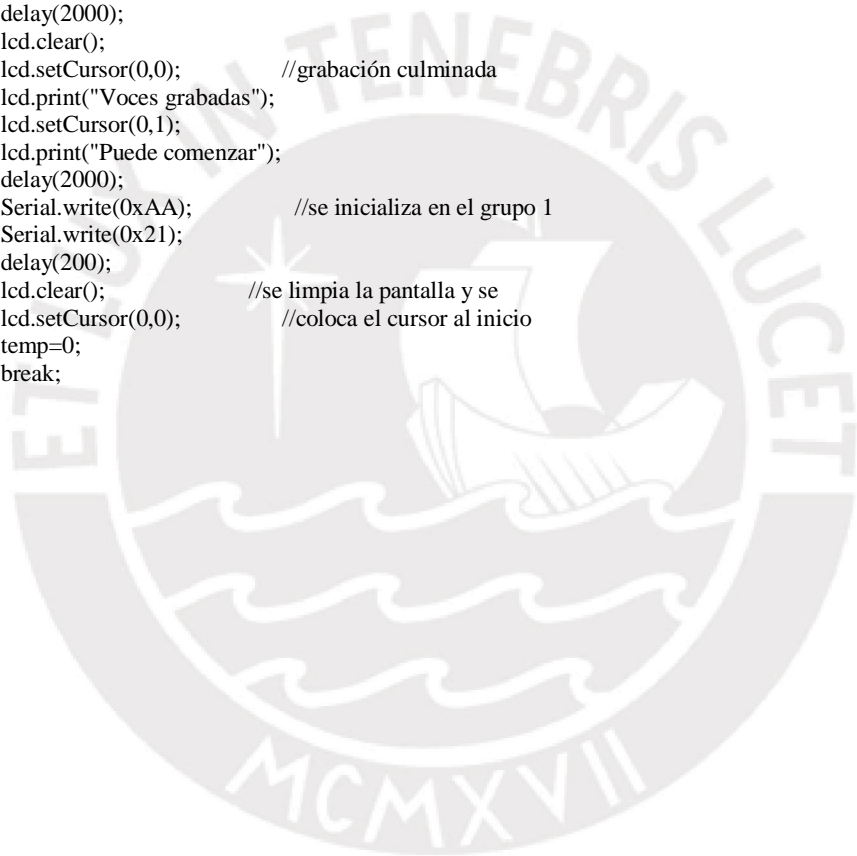
```

```

int temp=1; //variable temporal
lcd.clear();
lcd.setCursor(0,0); //mostramos opciones al usuario
lcd.print("MENU:Regresar");
lcd.setCursor(0,1);
lcd.print("GRABAR:Grabar");
while(temp==1){
  boton_menu_est=digitalRead(boton_menu); //leemos estado del botón menú
  boton_grabar_est=digitalRead(boton_grabar); //leemos estado del botón grabar
  if(boton_menu_est==HIGH){ //botón menú presionado
    while(boton_menu_est==HIGH){ //loop para evitar presión prolongada
      Serial.println("press 2"); //del boton
      boton_menu_est=digitalRead(boton_menu); //leemos estado del botón
    }
    lcd.clear();
    lcd.setCursor(0,0);
    temp=0;
  }else if(boton_grabar_est==HIGH){ //botón grabar presionado
    temp=3; //temporal
    int i=1; //temporal
    int n=1; //contador
    while(temp==3){ //loop para grabación de voces
      if(i==1){
        Serial.write(0xAA); //grabar grupo 1
        Serial.write(0x11);
        i=0;
      }else if(i==2){
        Serial.write(0xAA); //grabar grupo 2
        Serial.write(0x12);
        i=0;
      }else if(i==3){
        Serial.write(0xAA); //grabar grupo 3
        Serial.write(0x13);
        i=0;
      }else if(i==0){
        if(Serial.available()){ //leer datos según código que recibimos
          lcd.clear(); //para luego mostrarlo en el LCD
          lcd.setCursor(0,0);
          comando=Serial.read();
          switch (comando){
            case 0xe0:
              lcd.print("Error de");
              lcd.setCursor(0,1);
              lcd.print("Instruccion");
              break;
            case 0x40:
              lcd.print("Hable ahora!!");
              lcd.setCursor(0,1);
              lcd.print("comando "+String(n)); //n= número de comando
              break;
            case 0x41:
              lcd.print("No se detecto");
              lcd.setCursor(0,1);
              lcd.print("señal de voz");
              break;
            case 0x42:
              lcd.print("repita otra vez");
              lcd.setCursor(0,1);
              lcd.print("por favor");
              break;
            case 0x43:
              lcd.print("Muy ruidoso");
              lcd.setCursor(0,1);
              lcd.print("Para grabar");
              break;
            case 0x44:
              lcd.print("Sonó diferente");

```

```
    lcd.setCursor(0,1);
    lcd.print("Probemos de nuevo");
    break;
case 0x45:
    lcd.print("Comando Nro "+String(n));
    lcd.setCursor(0,1);
    lcd.print("se ha grabado");
    n++;
    break;
case 0x46:
    lcd.print("Grupo 1 Listo");
    i=2;
    break;
case 0x47:
    lcd.print("Grupo 2 Listo");
    i=3;
    break;
case 0x48:
    lcd.print("Grupo 3 Listo");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,0); //grabación culminada
    lcd.print("Voces grabadas");
    lcd.setCursor(0,1);
    lcd.print("Puede comenzar");
    delay(2000);
    Serial.write(0xAA); //se inicializa en el grupo 1
    Serial.write(0x21);
    delay(200);
    lcd.clear(); //se limpia la pantalla y se
    lcd.setCursor(0,0); //coloca el cursor al inicio
    temp=0;
    break;
}
}
}
}
}
}
```



## Anexo F

### Algoritmo implementado en el controlador PcDuino V3 para la parte del mecanismo robótico

```
//////////////////////////////////////  
//  
// Tesis: SISTEMA DE ASISTENCIA PARA PERSONAS CUADRIPLÉJICAS QUE //  
// PERMITE LA MANIPULACIÓN DE UN MECANISMO ROBÓTICO //  
// POR VOZ //  
// Titulo: "Control del brazo MICO Arm con recepción de comandos via //  
// puerto USB" //  
// Autor: Joel Eduardo Escalante Zelada //  
// //  
//////////////////////////////////////  
  
//Anexamos las librerías requeridas para el procesamiento  
#include "SerialClass.h" //librería para la comunicación Serial  
#include <string> //con el arduino NANO  
#include <stdio.h>  
#include <tchar.h>  
#include "Kinova.API.UsbCommandLayer.h" //librerías para la comunicación con  
#include "KinovaTypes.h" //el robot MICO Arm  
#include <conio.h>  
#include <iostream>  
  
using namespace std;  
  
//se inicializa variables globales  
HINSTANCE commandLayer_handle;  
//punteros de funciones relacionadas con el control del robot MICO Arm  
int (*MyInitAPI)();  
int (*MyCloseAPI)();  
int (*MySendJoystickCommand)(JoystickCommand command);  
int (*MyStartControlAPI)();  
int (*MyMoveHome)();  
int (*MySendAdvanceTrajectory)(TrajectoryPoint command);  
int (*MyGetQuickStatus)(QuickStatus &);  
int (*MySetControlMapping)(ControlMappingCharts com);  
int (*MyInitFingers)();  
  
int _tmain(int argc, _TCHAR* argv[])  
{  
    //se inicializa variables locales  
    int result;  
    char incomingData[255] = ""; //arreglo para la data recibida  
    printf("%s\n",incomingData);  
    int dataLength = 255; //tamaño de data  
    int readResult = 0;  
    //variable para verificar la carga de la librería  
    commandLayer_handle = LoadLibrary(L"Kinova.API.UsbCommandLayer.dll");  
    // verificamos si se cargo la librería, si no es asi se para la ejecucion  
    if(commandLayer_handle==NULL)  
    {  
        cout << "Error cuando se abrio la libreria," << endl;  
        return 0;  
    }else{  
        cout << "API cargada satisfactoriamente" << endl;  
    }  
  
    //se verifica si se conectó el arduino por el puerto Serial  
    Serial* SP = new Serial("COM6"); //para este caso el arduino se reconoció //en el puerto  
    "COM6"  
    if (SP->IsConnected()){  
        printf("sistema de reconocimiento de voz conectado...");  
    }else{  
        printf("No se detecto al sistema de reconocimiento de voz...");  
        return 0;  
    }  
  
    //se inicializa las funciones de la librería para el control del robot  
    MyInitAPI = (int (*)( )) GetProcAddress(commandLayer_handle, "InitAPI");
```

```

MyCloseAPI = (int (*)()) GetProcAddress(commandLayer_handle, "CloseAPI");
MySendAdvanceTrajectory = (int (*)(TrajectoryPoint))
GetProcAddress(commandLayer_handle, "SendAdvanceTrajectory");
MyInitFingers = (int (*)()) GetProcAddress(commandLayer_handle, "InitFingers");
MyGetQuickStatus = (int (*)(QuickStatus &))
GetProcAddress(commandLayer_handle, "GetQuickStatus");
MySetControlMapping = (int (*)(ControlMappingCharts))
GetProcAddress(commandLayer_handle, "SetControlMapping");
MySendJoystickCommand = (int (*)(JoystickCommand))
GetProcAddress(commandLayer_handle, "SendJoystickCommand");
MyStartControlAPI = (int (*)()) GetProcAddress(commandLayer_handle, "StartControlAPI");
MyMoveHome = (int (*)()) GetProcAddress(commandLayer_handle, "MoveHome");

//se verifica si las funciones del robot se cargaron correctamente
//si es así se procesa los comandos recibidos enviando las ordenes respectivas
//al robot, caso contrario se termina la ejecución del programa
if((MyInitAPI == NULL) || (MyCloseAPI == NULL) || (MySendAdvanceTrajectory == NULL)
    || (MySendJoystickCommand == NULL) || (MyStartControlAPI == NULL) || (MyMoveHome ==
NULL)
    || (MyGetQuickStatus == NULL) || (MyInitFingers == NULL))
{
    cout << "No se pueden inicializar funciones del robot" << endl;
}
else
{
    //se llaman algunos métodos
    cout << "las funciones se inicializaron correctamente" << endl;
    cout << "llamamos a la funcion InitAPI()" << endl;
    result = (*MyInitAPI)();
    cout << "respuesta del metodo InitAPI() = " << result << endl << endl;

    cout << "se toma control del robot MICO Arm" << endl;
    result = (*MyStartControlAPI)();

    //se inicializa la variable de control como joystick del robot MICO
    JoystickCommand virtualCommand;
    SystemStatus
    //inicializamos el control de los dedos del brazo robótico
    (*MyInitFingers)();
    //inicializamos la variable de posición para el robot
    TrajectoryPoint pointToSend;
    pointToSend.InitStruct();

    //Inicializamos cada comando que se usara para mover el brazo
    //robótico, cada comando copiara una orden del joystick
    // los valores 0 representan botón sin pulsar y 1 botones pulsados
    for(int i = 0; i < JOYSTICK_BUTTON_COUNT; i++)
    {
        virtualCommand.ButtonValue[i] = 0; //botones del joystick sin pulsar
    }

    virtualCommand.InclineForwardBackward = 0;
    virtualCommand.InclineLeftRight = 0;
    virtualCommand.MoveForwardBackward = 0;
    virtualCommand.MoveLeftRight = 0;
    virtualCommand.PushPull = 0;
    virtualCommand.Rotate = 0;
    (*MySendJoystickCommand)(virtualCommand); //se envía la configuración

    (*MyMoveHome)(); //Comando para mover al robot a la posición
    // de inicio de movimiento

    //Proceso de ejecución de movimientos según comando recibido
    while(SP->IsConnected()) //ejecutamos siempre y cuando este conectado el arduino
    {
        readResult = SP->ReadData(incomingData,dataLength); //se lee la data que recibimos
        //printf("Bytes leidos:%i\n",readResult);
        printf("%s",incomingData);

        //se verifica que comando se obtuvo y se ejecuta la acción
        //se comparan las 4 primeras palabras para evitar rellenar con ceros
        //algunas comparaciones
        if(strncmp(incomingData,"bebida",4)==0){ //el comando es bebida?
            printf("detectado\n");
            //char incomingData[0] = '\0' ;

```



```

memset( incomingData , 0 , sizeof ( incomingData ));//se limpia
(*MyMoveHome)(); //el arreglo

pointToSend.Position.Type = ANGULAR_POSITION;
pointToSend.LimitationsActive = 1;

// se procede a mover el brazo al lugar del bazo
// Configurando las posiciones de los actuadores según la posición
//del vaso
pointToSend.Limitations.speedParameter1 = 8;
pointToSend.Limitations.speedParameter2 = 8;
pointToSend.Position.Actuators.Actuator1 = 283.952f;
pointToSend.Position.Actuators.Actuator2 = 256.765f;
pointToSend.Position.Actuators.Actuator3 = 76.5993f;
pointToSend.Position.Actuators.Actuator4 = 162.614f;
pointToSend.Position.Actuators.Actuator5 = 103.841f;
pointToSend.Position.Actuators.Actuator6 = 145.5f;
pointToSend.Position.Fingers.Finger1 = 690.0f;
pointToSend.Position.Fingers.Finger2 = 690.0f;
pointToSend.Position.Fingers.Finger3 = 0.0f;

cout << "Sending trajectory" << endl;
(*MySendAdvanceTrajectory)(pointToSend); //comando para enviar la trayectoria
Sleep(13000);
pointToSend.Position.Type = CARTESIAN_POSITION;
pointToSend.LimitationsActive = 1;

pointToSend.Limitations.speedParameter1 = 8;
pointToSend.Limitations.speedParameter2 = 8;
//(*MySendAdvanceTrajectory)(pointToSend);

pointToSend.Position.CartesianPosition.X = 0.00648323f;
pointToSend.Position.CartesianPosition.Y = -0.515796f;
pointToSend.Position.CartesianPosition.Z = 0.11812f;
pointToSend.Position.CartesianPosition.ThetaX = 1.644413f;
pointToSend.Position.CartesianPosition.ThetaY = 0.0212168f;
pointToSend.Position.CartesianPosition.ThetaZ = 0.033327f;

cout << "Sending trajectory" << endl;
(*MySendAdvanceTrajectory)(pointToSend);

//ya en la posición, el brazo se acercará lentamente hacia el vaso
Sleep(1000);
virtualCommand.InclineForwardBackward = -1;
for(int i = 0; i < 2; i++)
{
    (*MySendJoystickCommand)(virtualCommand);
}
Sleep(1000);

virtualCommand.InclineForwardBackward = 0;
(*MySendJoystickCommand)(virtualCommand);

Sleep(2000);
virtualCommand.ButtonValue[1] = 1;
(*MySendJoystickCommand)(virtualCommand);
Sleep(10);
virtualCommand.ButtonValue[1] = 0;
(*MySendJoystickCommand)(virtualCommand);
// se procederá a cerrar la pinza para sujetar el vaso
virtualCommand.InclineForwardBackward = 1;
for(int i = 0; i < 1; i++)
{
    (*MySendJoystickCommand)(virtualCommand);
}
Sleep(1000);

virtualCommand.InclineForwardBackward = 0;
(*MySendJoystickCommand)(virtualCommand);

virtualCommand.ButtonValue[1] = 1;
(*MySendJoystickCommand)(virtualCommand);
Sleep(10);

```

```

virtualCommand.ButtonValue[1] = 0;
(*MySendJoystickCommand)(virtualCommand);

Sleep(2000);

virtualCommand.ButtonValue[1] = 1;
(*MySendJoystickCommand)(virtualCommand);
Sleep(10);
virtualCommand.ButtonValue[1] = 0;
(*MySendJoystickCommand)(virtualCommand);

}else if(strncmp(incomingData,"boca",4)==0){ //el comando es boca?
printf("detectado\n");
memset( incomingData , 0 , sizeof ( incomingData ));//se limpia
(*MyMoveHome()); //el arreglo

pointToSend.Position.Type = ANGULAR_POSITION;
pointToSend.LimitationsActive = 1;
// se envía el brazo a la posición de la boca del usuario
pointToSend.Limitations.speedParameter1 = 8;
pointToSend.Limitations.speedParameter2 = 8;
pointToSend.Position.Actuators.Actuator1 = 446.085f;
pointToSend.Position.Actuators.Actuator2 = 185.515f;
pointToSend.Position.Actuators.Actuator3 = 91.6544f;
pointToSend.Position.Actuators.Actuator4 = 240.614f;
pointToSend.Position.Actuators.Actuator5 = 83.8636f;
pointToSend.Position.Actuators.Actuator6 = 428.864f;
pointToSend.Position.Fingers.Finger1 = 6000.0f;
pointToSend.Position.Fingers.Finger2 = 6000.0f;
pointToSend.Position.Fingers.Finger3 = 0.0f;
cout << "Sending trajectory" << endl;
(*MySendAdvanceTrajectory)(pointToSend);
Sleep(25000);
pointToSend.Position.Type = CARTESIAN_POSITION;
pointToSend.LimitationsActive = 1;

pointToSend.Limitations.speedParameter1 = 8;
pointToSend.Position.CartesianPosition.X = -0.253109f;
pointToSend.Position.CartesianPosition.Y = 0.285874f;
pointToSend.Position.CartesianPosition.Z = 0.499287f;
pointToSend.Position.CartesianPosition.ThetaX = -1.3244f;
pointToSend.Position.CartesianPosition.ThetaY = -1.28677f;
pointToSend.Position.CartesianPosition.ThetaZ = -2.88717f;

cout << "Sending trajectory" << endl;
(*MySendAdvanceTrajectory)(pointToSend);
Sleep(1000);

virtualCommand.ButtonValue[3] = 1;
(*MySendJoystickCommand)(virtualCommand);
Sleep(10);
virtualCommand.ButtonValue[3] = 0;
(*MySendJoystickCommand)(virtualCommand);

virtualCommand.ButtonValue[1] = 1;
(*MySendJoystickCommand)(virtualCommand);
Sleep(10);
virtualCommand.ButtonValue[1] = 0;
(*MySendJoystickCommand)(virtualCommand);

Sleep(2000);
// se coloca el brazo en modo bebida y se procede a beber del vaso
virtualCommand.ButtonValue[1] = 1;
(*MySendJoystickCommand)(virtualCommand);
Sleep(10);
virtualCommand.ButtonValue[1] = 0;
(*MySendJoystickCommand)(virtualCommand);

virtualCommand.Rotate = 1;
for(int i = 0; i < 4; i++)
{
(*MySendJoystickCommand)(virtualCommand);

```

```

Sleep(1000);
}
    virtualCommand.Rotate = 0;
    (*MySendJoystickCommand)(virtualCommand);

    Sleep(1000);

    virtualCommand.Rotate = -1;
    for(int i = 0; i < 4; i++)
{
        (*MySendJoystickCommand)(virtualCommand);
}
Sleep(1000);
}
    virtualCommand.Rotate = 0;
    (*MySendJoystickCommand)(virtualCommand);

    virtualCommand.ButtonValue[1] = 1;
    (*MySendJoystickCommand)(virtualCommand);
    Sleep(10);
    virtualCommand.ButtonValue[1] = 0;
    (*MySendJoystickCommand)(virtualCommand);
    virtualCommand.ButtonValue[3] = 1;
    (*MySendJoystickCommand)(virtualCommand);
    Sleep(10);
    virtualCommand.ButtonValue[3] = 0;
    (*MySendJoystickCommand)(virtualCommand);

}else if(strncmp(incomingData,"para",4)==0){ //el comando es para?
    printf("detectado\n");

    //char incomingData[0] = '\0' ;
    memset( incomingData , 0 , sizeof ( incomingData ));
    //procederá a deshabilitar cualquier movimiento en el robot
    for(int i = 0; i < JOYSTICK_BUTTON_COUNT; i++)
{
        virtualCommand.ButtonValue[i] = 0;//botones del joystick sin pulsar
}

virtualCommand.InclineForwardBackward = 0;
virtualCommand.InclineLeftRight = 0;
virtualCommand.MoveForwardBackward = 0;
virtualCommand.MoveLeftRight = 0;
virtualCommand.PushPull = 0;
virtualCommand.Rotate = 0;
    (*MySendJoystickCommand)(virtualCommand); //se envía la configuración

}else if(strncmp(incomingData,"arriba",4)==0){ //el comando es arriba?
    printf("detectado\n");

    //char incomingData[0] = '\0' ;
    memset( incomingData , 0 , sizeof ( incomingData ));

    virtualCommand.Rotate= 1;
    (*MySendJoystickCommand)(virtualCommand);

}else if(strncmp(incomingData,"abajo",4)==0){ //el comando es abajo?
    printf("detectado\n");
    memset( incomingData , 0 , sizeof ( incomingData ));

    virtualCommand.Rotate= -1;
    (*MySendJoystickCommand)(virtualCommand);

}else if(strncmp(incomingData,"derecha",4)==0){ //el comando es derecha?
    printf("detectado\n");
    memset( incomingData , 0 , sizeof ( incomingData ));

    virtualCommand.InclineLeftRight= 1;
    (*MySendJoystickCommand)(virtualCommand);

}else if(strncmp(incomingData,"izquierda",4)==0){ //el comando es izquierda?
    printf("detectado\n");
    memset( incomingData , 0 , sizeof ( incomingData ));

```

```

        virtualCommand.InclineLeftRight= -1;
        (*MySendJoystickCommand)(virtualCommand);

        }else if(strncmp(incomingData,"gira",4)==0){ //el comando es gira?
            printf("detectado\n");
            memset( incomingData , 0 , sizeof ( incomingData ));//se limpia
            (*MyMoveHome()); //el arreglo

        }else if(strncmp(incomingData,"adelante",4)==0){ //el comando es adelante?
            printf("detectado\n");
            memset( incomingData , 0 , sizeof ( incomingData ));

            virtualCommand.InclineForwardBackward= -1;
            (*MySendJoystickCommand)(virtualCommand);

            }else if(strncmp(incomingData,"atras",4)==0){ //el comando es atrás?
                printf("detectado\n");
                memset( incomingData , 0 , sizeof ( incomingData ));

                virtualCommand.InclineForwardBackward= 1;
                (*MySendJoystickCommand)(virtualCommand);

                }else if(strncmp(incomingData,"abre",4)==0){ //el comando es abre?
                    printf("detectado\n");
                    memset( incomingData , 0 , sizeof ( incomingData ));//se limpia
                    (*MyMoveHome()); //el arreglo

                    virtualCommand.InclineForwardBackward= 1;
                    (*MySendJoystickCommand)(virtualCommand);
                    }else if(strncmp(incomingData,"cierra",4)==0){ //el comando es cierra?
                        printf("detectado\n");
                        memset( incomingData , 0 , sizeof ( incomingData ));//se limpia
                        (*MyMoveHome()); //el arreglo

                        virtualCommand.InclineForwardBackward= -1;
                        (*MySendJoystickCommand)(virtualCommand);
                        }else if(strncmp(incomingData,"rota",4)==0){
                            printf("detectado\n");
                            memset( incomingData , 0 , sizeof ( incomingData ));//se limpia
                            (*MyMoveHome()); //el arreglo

                            virtualCommand.Rotate= 1;
                            (*MySendJoystickCommand)(virtualCommand);
                        }
                    }
                Sleep(500);
            }
        }
    }
    return 0;
}

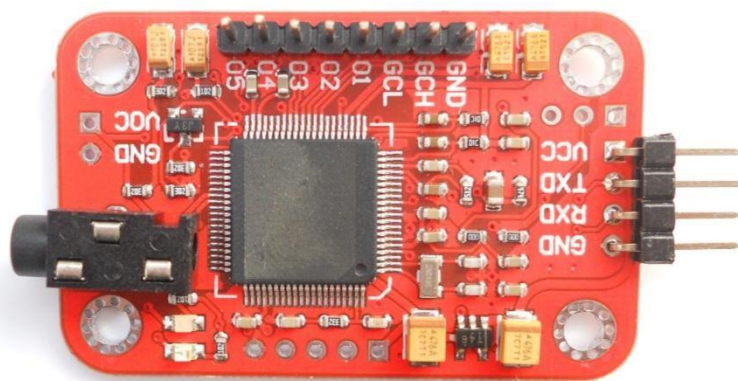
```

## Anexo G

### Manual del módulo de reconocimiento de voz V2 de la marca Elechouse

# Voice Recognition Module V2

Speak to control (Arduino Compatible)



## Introduction

The module could recognize your voice. It receives configuration commands or responds through serial port interface. With this module, we can control the car or other electrical devices by voice. This module can store up to 15 pieces of voice instruction. Those 15 pieces are divided into 3 groups, with 5 in each group. First we should train the module with voice instructions group by group. After that, we should import one group before it could recognize the 5 voice instructions within that group. If we need to implement instructions in other groups, we should import the group first. This module is speaker dependent. If you trained the module, your friend might not be able to make it work.

## What's new in V2

We've updated this module to V2. We made V2 easy to control. Except only serial input or output of V1, V2 has other useful ways to control and output the result.

You could find a new GOPI row on V2. GCH and GCL are used to import the voice group. And O1~O5 are pins which output the result of voice recognition. For example, if the first voice instruction in the working group is recognized, O1 could output HIGH signal. This output sometimes is very useful, such as while controlling the relay.

The O1~O5 output could be set as many type. You could set it by sending command to it through serial interface. Those setting will be recorded in memory. It will not lose even with power off. You could find the commands in later content.

V1 is black and V2 is red.

**Technical**

**Parameters**

- Voltage: 4.5-5.5V
- Current: <40mA
- Digital Interface: 5V TTL level UART interface and GPIO
- Analog Interface: 3.5mm mono-channel microphone connector + microphone pin interface  
 □ Size: 30mm x 47.5mm
- Recognition accuracy: 99% (under ideal environment)

**Serial Command**

This module can be configured by sending commands via serial port. Configuration will be not erased after powered off.

Its interface is 5V TTL. The serial data format: 8 data bits, no parity, 1 stop bit. The default baud rate is 9600 and baud rate can be changed.

Command format is "Head + Key". "Head" is a 0xaa, and "Key" is as follows:

Key (HEX format)	Description	Respond in Common Mode	Respond in Compact Mode
0x00	Enter into "Waiting" state	"Waiting! \n" : successful "ERROR! \n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x01	Delete the instructions of group 1	"Group1 Deleted ! \n" : successful "ERROR! \n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x02	Delete the instructions of group 2	"Group2 Deleted ! \n" : successful "ERROR! \n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x03	Delete the instructions of group 3	"Group3 Deleted ! \n" : successful "ERROR! \n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x04	Delete the instructions of all the 3 groups	" All Groups Deleted !\n " : successful "ERROR! \n" : Instruction error	0xcc : successful 0xe0 : Instruction error
0x11	Begin to record instructions of group 1	"ERROR! \n" : Instruction error "START \n" : Ready for recording, you can speak	0xe0 : Instruction error 0x40 : Ready for recording, you



		<p>now</p> <p>"No voice \ n" : no voice detected</p> <p>"Again \ n" : Speak the voice instruction again.</p> <p>Do not speak until getting the START message</p> <p>"Too loud \ n" : Too loud to record</p> <p>"Different \ n" : voice instruction confirming failed. Voice for the second chance is different with the first one.</p> <p>"Finish one \ n" : recording one voice instruction successfully</p> <p>"Group1 finished! \ n" : finish recording group 1</p>	<p>can speak now</p> <p>0x41 : no voice detected</p> <p>0x42 : Speak the voice instruction again. Do not speak until getting the START message</p> <p>0x43 : Too loud to record</p> <p>0x44 : voice instruction confirming failed. Voice for the second chance is different with the first one.</p> <p>0x45 : recording one voice instruction successfully</p> <p>0x46 : finish recording group 1</p>
<b>0x12</b>	Begin to record instructions of group 2	<p>"ERROR! \ n" : Instruction error</p> <p>"START \ n" : Ready for recording, you can speak now</p> <p>"No voice \ n" : no voice detected</p> <p>"Again \ n" : Speak the voice instruction again.</p> <p>Do not speak until getting the START message</p> <p>"Too loud \ n" : Too loud to record</p> <p>"Different \ n" : voice instruction confirming failed. Voice for the second chance is different with the first one.</p> <p>"Finish one \ n" : recording one voice instruction successfully</p> <p>"Group2 finished! \ n" : finish recording group 2</p>	<p>0xe0 : Instruction error</p> <p>0x40 : Ready for recording, you can speak now</p> <p>0x41 : no voice detected</p> <p>0x42 : Speak the voice instruction again. Do not speak until getting the START message</p> <p>0x43 : Too loud to record</p> <p>0x44 : voice instruction confirming failed. Voice for the second chance is different with the first one.</p> <p>0x45 : recording one voice instruction successfully</p> <p>0x47 : finish recording group 2</p>
<b>0x13</b>	Begin to record instructions of group 3	<p>"ERROR! \ n" : Instruction error</p> <p>"START \ n" : Ready for recording, you can speak now</p> <p>"No voice \ n" : no voice detected</p> <p>"Again \ n" : Speak the voice instruction again.</p> <p>Do not speak until getting the START message</p>	<p>0xe0 : Instruction error</p> <p>0x40 : Ready for recording, you can speak now</p> <p>0x41 : no voice detected</p> <p>0x42 : Speak the voice instruction again. Do not speak</p>

		<p>"Too loud \ n" : Too loud to record</p> <p>"Different \ n" : voice instruction confirming failed. Voice for the second chance is different with the first one.</p> <p>"Finish one \ n" : recording one voice instruction successfully</p> <p>"Group3 finished! \ n" : finish recording group 3</p>	<p>until getting the START message</p> <p>0x43 : Too loud to record</p> <p>0x44 : voice instruction confirming failed. Voice for the second chance is different with the first one.</p> <p>0x45 : recording one voice instruction successfully</p> <p>0x48 : finish recording group 3</p>
<b>0x21</b>	Import group 1 and be ready for voice instruction	<p>"Group1 Imported !\n" : Successful</p> <p>"ERROR! \ n" : Instruction error</p> <p>"Import failed !\n" : Importing voice group failed</p>	<p>0xcc : Successful</p> <p>0xe0 : Instruction error</p> <p>0xe1 : Importing voice group failed</p>
<b>0x22</b>	Import group 2 and be ready for voice instruction	<p>"Group2 Imported !\n" : Successful</p> <p>"ERROR! \ n" : Instruction error</p> <p>"Import failed !\n" : Importing voice group failed</p>	<p>0xcc : Successful</p> <p>0xe0 : Instruction error</p> <p>0xe1 : Importing voice group failed</p>
<b>0x23</b>	Import group 3 and be ready for voice instruction	<p>"Group3 Imported !\n" : Successful</p> <p>"ERROR! \ n" : Instruction error</p> <p>"Import failed !\n" : Importing voice group failed</p>	<p>0xcc : Successful</p> <p>0xe0 : Instruction error</p> <p>0xe1 : Importing voice group failed</p>

<b>0x24</b>	Query the recorded group	<p>"Used group:0\n" : No group is recorded</p> <p>"Used group:1\n" : Group 1 is recorded</p> <p>"Used group:2\n" : Group 2 is recorded</p> <p>"Used group:3\n" : Group 3 is recorded</p> <p>"Used group:12\n" : Group 1 and Group 2 are recorded</p> <p>"Used group:13\n" : Group 1 and Group 3 are recorded</p> <p>"Used group:23\n" : Group 2 and Group 3 are recorded</p> <p>"Used group:123\n" : All the 3 groups are recorded</p> <p>"ERROR! \ n" : Instruction error</p>	<p>0x00 : No group is recorded</p> <p>0x01 : Group 1 is recorded</p> <p>0x02 : Group 2 is recorded</p> <p>0x04 : Group 3 is recorded</p> <p>0x03 : Group 1 and Group 2 are recorded</p> <p>0x05 : Group 1 and Group 3 are recorded</p> <p>0x06 : Group 2 and Group 3 are recorded</p> <p>0x07 : All the 3 groups are recorded</p> <p>0xe0 : Instruction error</p>
<b>0x31</b>	Change the baud rate to 2400bps	<p>"Baud: 2400\n" : Successful</p> <p>"ERROR! \ n" : Instruction error</p>	<p>0xcc : successful</p> <p>0xe0 : Instruction error</p>
<b>0x32</b>	Change the baud rate to 4800bps	<p>"Baud: 4800\n" : Successful</p> <p>"ERROR! \ n" : Instruction error</p>	
<b>0x33</b>	Change the baud rate to 9600bps	<p>"Baud: 9600\n" : Successful</p> <p>"ERROR! \ n" : Instruction error</p>	

<b>0x34</b>	Change the baud rate to 19200bps	"Baud: 19200\n" : Successful "ERROR! \n" : Instruction error	
<b>0x35</b>	Change the baud rate to 38400bps	"Baud: 38400\n" : Successful "ERROR! \n" : Instruction error	
<b>0x36</b>	Switch to Common Mode	"Common Mode\n" : Successful "ERROR! \n" : Instruction error	
<b>0x37</b>	Switch to Compact Mode	"Compact Mode\n" : Successful "ERROR! \n" : Instruction error	
<b>0x41</b>	Reset output of O1	"Ok\n" : Successful "ERROR! \n" : Instruction error	0xcc : successful 0xe0 : Instruction error
<b>0x42</b>	Reset output of O2		
<b>0x43</b>	Reset output of O3		
<b>0x44</b>	Reset output of O4		
<b>0x45</b>	Reset output of O5		
<b>0x46</b>	Reset output of O1~O5		
<b>0x50</b>	Set O1~O5 to Pulse Mode (negative pulse)	"Ok\n" : Successful "ERROR! \n" : Instruction error	0xcc : successful 0xe0 : Instruction error
<b>0x51</b>	Set O1~O5 to Flip Mode		
<b>0x52</b>	Set O1~O5 to Down Mode		
<b>0x53</b>	Set O1~O5 to Up Mode		
<b>0x60</b>	Set the output duty of O1~O5 in Pulse Mode 0x60 -- 10ms 0x61 -- 15ms 0x62 -- 20ms 0x63 -- 25ms 0x64 -- 30ms 0x65 -- 50ms 0x66 -- 60ms 0x67 -- 70ms 0x68 -- 80ms 0x69 -- 90ms 0x6A -- 100ms 0x6B -- 200ms 0x6C -- 300ms	"Ok\n" : Successful "ERROR! \n" : Instruction error	0xcc : successful 0xe0 : Instruction error
<b>0x61</b>			
<b>0x62</b>			
<b>0x63</b>			
<b>0x64</b>			
<b>0x65</b>			
<b>0x66</b>			
<b>0x67</b>			
<b>0x68</b>			
<b>0x69</b>			
<b>0x6A</b>			
<b>0x6B</b>			
<b>0x6C</b>			
<b>0x6D</b>			
<b>0x6E</b>			
<b>0x6F</b>			

<b>0x70</b>	Reset Serial Port to: 9600 baud rate, 8 data bits, no parity, 1 stop bit	"Ok\n" : Successful "ERROR! \n" : Instruction error	0xcc : successful 0xe0 : Instruction error
<b>0xbb</b>	Query version information	Version information	No respond

If you want to modify the serial baud rate to 38400, you need to send command: **0xaa35**. If successful, it will return "Baud: 38400 \n"(in Common Mode) or **0xcc** (in Compact Mode). The baud rate is set to 38400.

The main difference between Compact Mode and Common Mode is the returning message. Common Mode response is long string but Compact Mode response is a byte. For example, after sending **0xaa04** to delete all the contents of the 3 groups, in Common Mode it will return "**All Groups Deleted! \n**", but in Compact Mode it will return a concise bytes such as **0xcc** which means a successful operation.

Here we will introduce more about the output of O1~O5:

**Pulse Mode:** Output is negative pulse. The pulse duration time (pulse duty) could be set by command **0x60 ~ 0x6F**

**Flip Mode:** each time while the module recognizes voice command, it will change the state of the output pin.

**Down Mode:** The output will become LOW from HIGH once it detects voice command. It will never come back to HIGH again until the module receives output reset command **0x41 ~ 0x46**

**Up Mode:** The output will become HIGH from LOW once it detects voice command. It will never come back to LOW again until the module receives output reset command **0x41 ~ 0x46** There will be an example:

A: Starting point while you import the voice group.

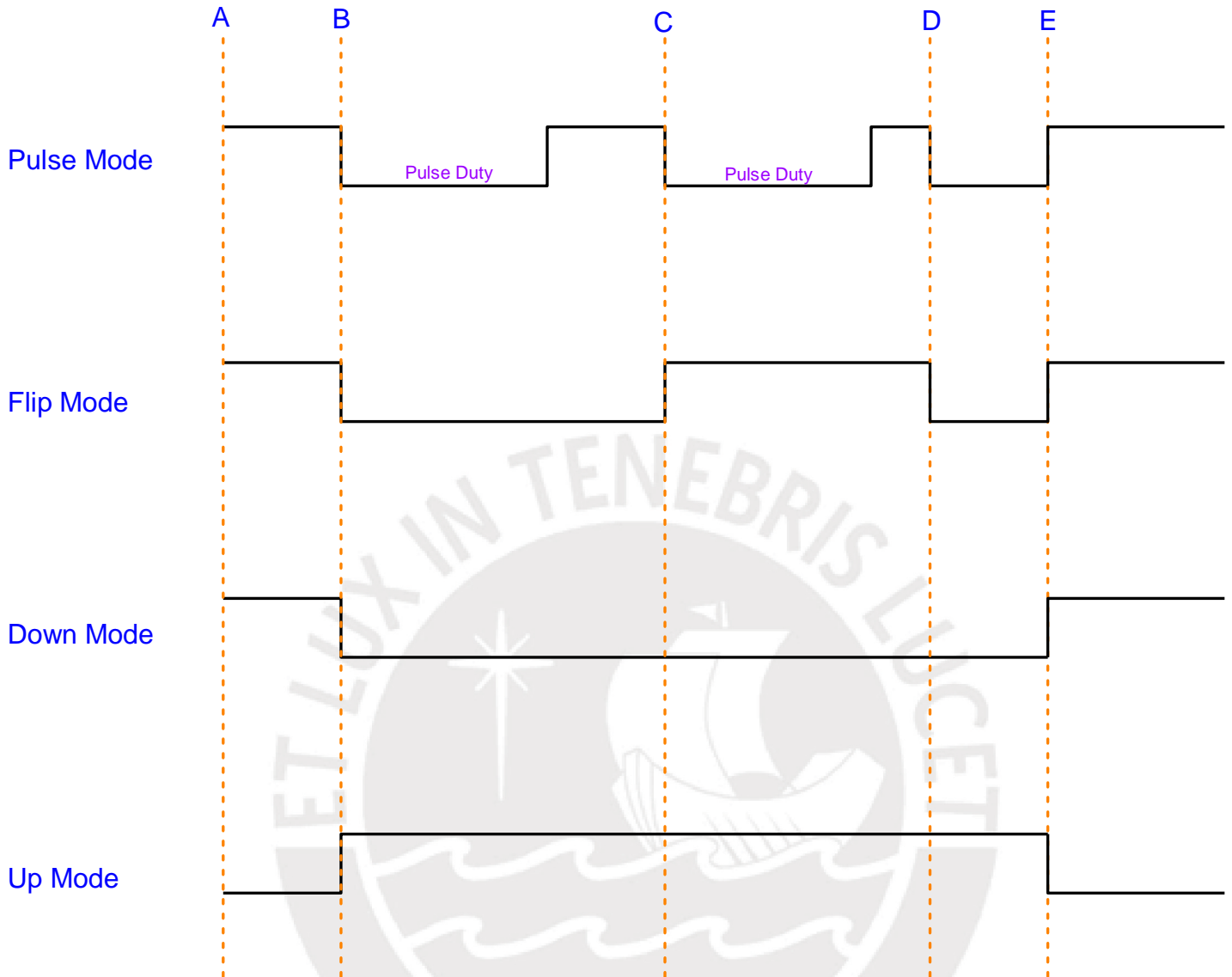
B: The 1<sup>st</sup> time it recognizes voice command.

C: The 2<sup>nd</sup> time it recognizes voice command.

D: The 3<sup>rd</sup> time it recognizes voice command.

E: The time while output reset command is received (**0x41 ~ 0x46**).

The back wire is output wave shape.



For the first-time use, we need to do some configuration:

1. Select the serial baud rate (default 9600)
2. Select the communication mode: Common Mode or Compact Mode
3. Recording five instructions of the first group( or 2nd or 3rd as required)
4. Import the group you need to use (only recognize 5 instructions within one group at the same time)

After all the setting above, you can speak or send voice instruction to it. If identified successfully, result will be returned via serial port in the format: group number + command number. For example, return **Result: 11** (Compact mode returns 0x11) means identified the first command of group 1.

If voice instruction is recorded, each time after you power it on, you need to import the group before letting it identify voice instructions.

## LED

Recording stage:

1. Record indication: D1 (RED) flashes 3 times within the 600ms, then off for 400ms, and then flashes quickly for 4 times within 600ms. Now the recording indication is over.

2. Begin to speak: D1 (RED) is off for 400ms, and then is on. Voice during the time while D1 (RED) is on will be recorded by this module.
3. Recording a voice instruction successfully for the first time: D1 (RED) off, D2 (ORANGE) on for 300ms.
4. Recording a voice instruction successfully for the first time: D1 (RED) off, D2 (ORANGE) on for 700ms.
5. Recording failure: D2 (ORANGE) flashes 4 times within the 600ms. In cases that voice instructions detected twice don't match, or the sound is too large, or there is no sound, recording will fail. You need to start over the recording process for that instruction.

Waiting mode:

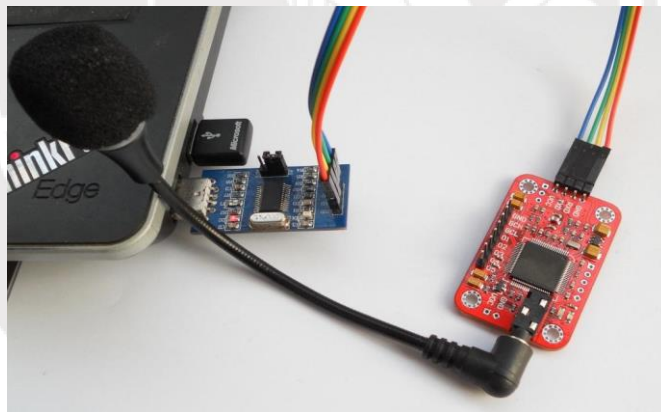
In waiting mode, D2 (ORANGE) is off, and D1 (RED) is on for 80ms every other 200ms, fast flashing. In this mode, it doesn't recognize voice command, only waiting for serial commands.

Recognition stage:

In identification stage, D2 (ORANGE) is off, and D1 (RED) is on for 100ms every other 1500ms, slow flashing. In this stage, this module is processing received voice signal, and if matching, it will send the result immediately via serial port.

## Recording

Before using it, we have train it by recording voice instructions. Each voice instruction has the maximum length of 1300ms, which ensures that most words can be recorded. Once you start recording, you can't stop the recording process until you finish all the 5 voice instructions recording of one group. Also, once you start recording, the previous voice instructions in that group will be erased. In training state, this module doesn't reply to any other serial commands.

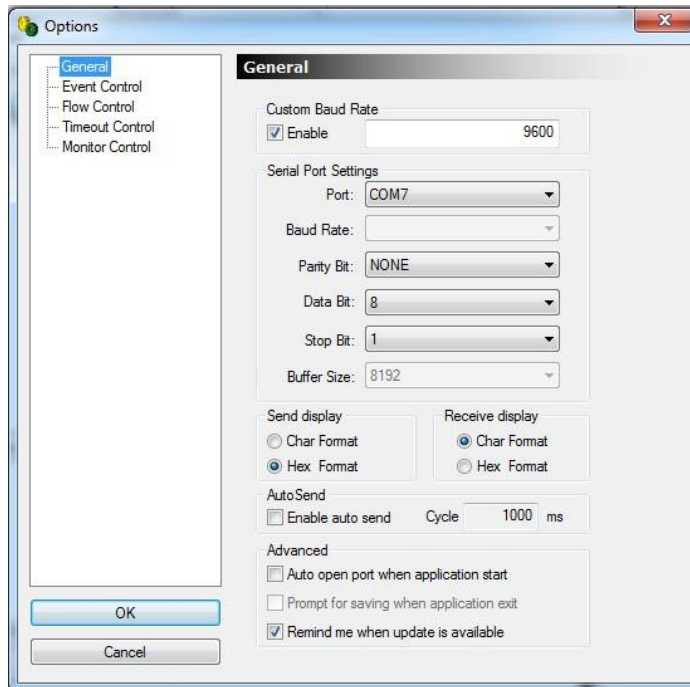


LED will flash to indicate state. Please refer to the LED part.

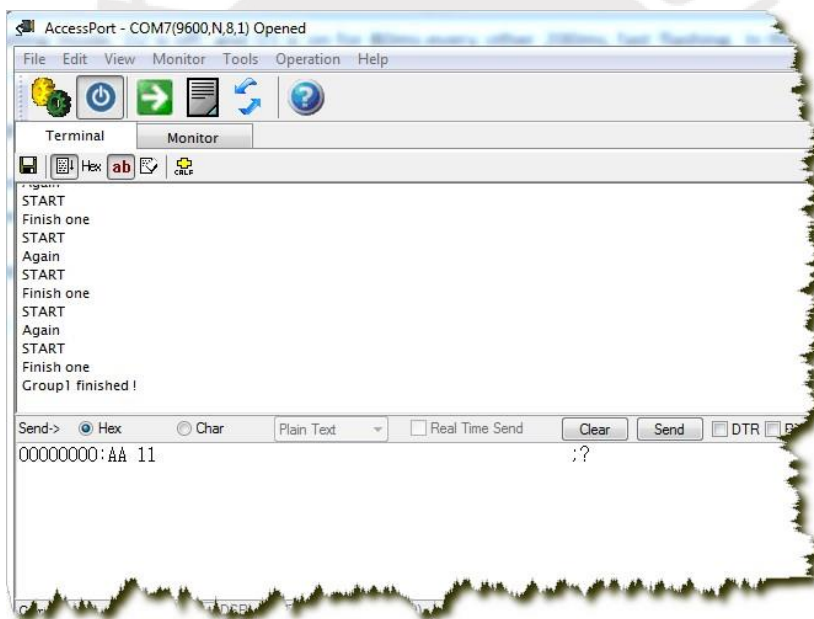
First, you need a serial tool. Here we use AccessPort ([Download page](#)).

1. Serial port setting:
  - Baud rate: 9600
  - Parity bit: None
  - Data bit: 8
  - Stop bit: 1
  - Send format: Hex
  - Receive format: Char





## 2. Send commands



Send:  
**0xaa11**

Receive (in Common Mode):  
**START**

**No voice** // I did not make any sound. So it replied such message

**START**

**Speak now**

**Again**

**START**

Speak again now

**Different** // I spoke another words for the second time. So it replied such message

**START**

Speak now

**Again**

**START**

Speak again now

**Finish one** // recording one instruction successfully

**START**

**Again**

**START**

**Finish one**

**START**

**Again**

**START**

**Finish one**

**START**

**Again**

**START**

**Finish one**

**START**

**Again**

**START**

**Finish one**

**Group1 finished!** // recording group 1 successfully

By now, you've finished instruction recording for group 1.

## **Recognition**

We added another way to import the voice instruction group on V2. For V1, the only way is to send command to it through serial port, for example:



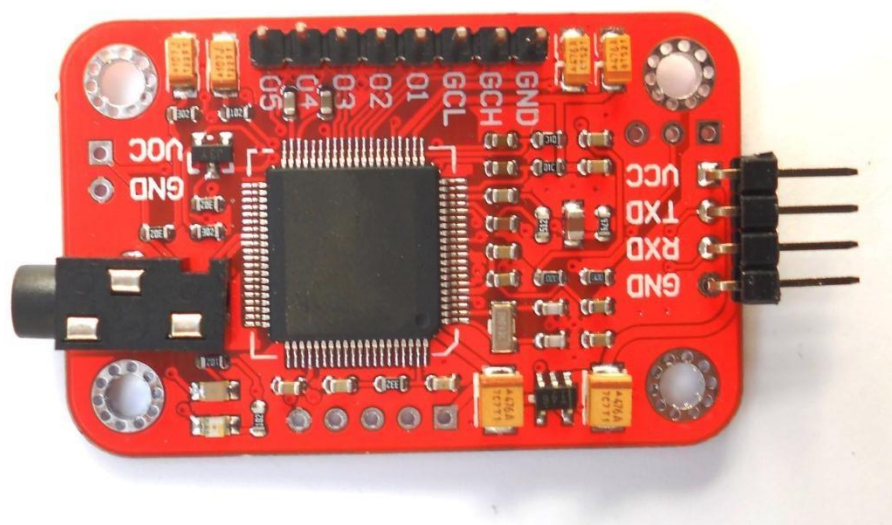
- Send command **0xAA21** to import group 1.
- Send command **0xAA22** to import group 2.
- Send command **0xAA23** to import group 3.

On V2, we added an easy way to import the group:

Look at the following picture, you could control GCH and GCL pins. Those two pins are all 5V TTL.

GCH	GCL	Working State
LOW	LOW	No Change
LOW	HIGH	Import Group 2
HIGH	LOW	Import Group 1
HIGH	HIGH	Import Group 3

Try to connect GND to make sure they have the common ground.



Once the group is important, it will output message through serial port. It could have 15 voice instructions in 3 groups. Each time you need to import the group before it could recognize instructions in that group. That means, this module could recognize 5 voice instructions at the same time. In recognition stage, this module could receive other serial commands. It will not exit the recognition stage until you send **0xAA00**, or delete that group, or begin recording instructions.

Group 1

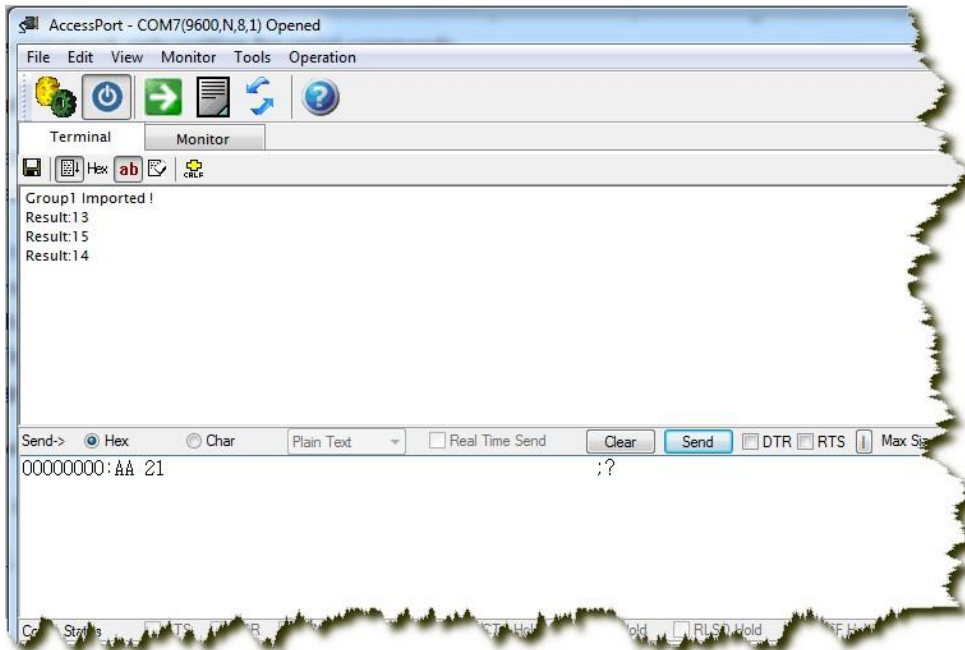
Command A1  
Command A2  
Command A3  
Command A4  
Command A5

Group 2

Command B1  
Command B2  
Command B3  
Command B4  
Command B5

Group 3

Command C1  
Command C2  
Command C3  
Command C4  
Command C5



Output

Video:

<http://www.youtube.com/watch?v=>

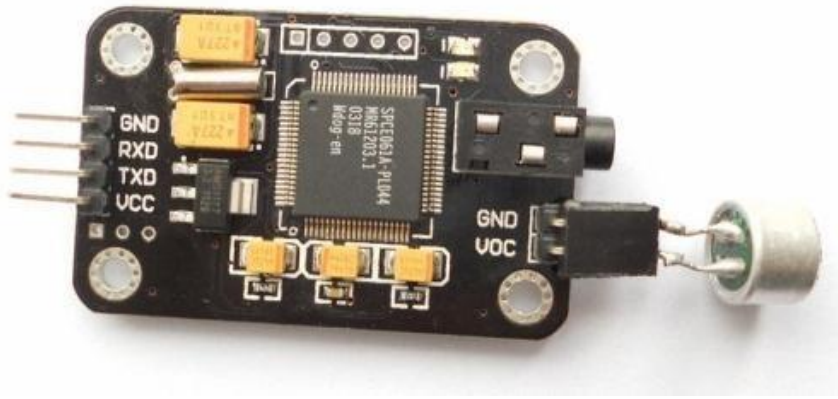
[OZjdu\\_HSr00](#) There are some tricks

to improve the recognition rate.

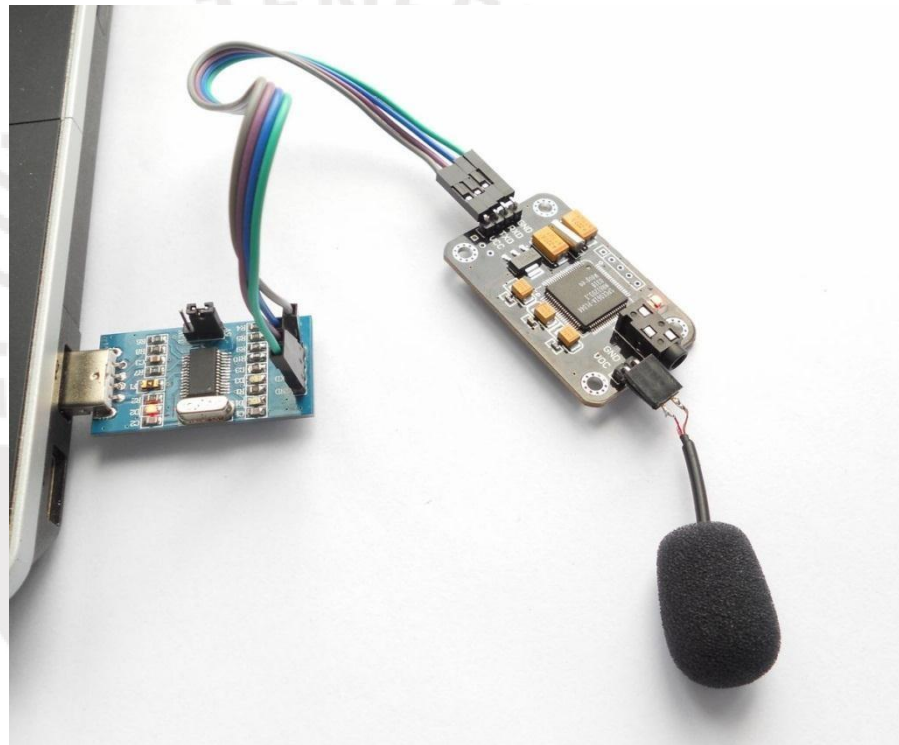
Try to record and use it in the same environment. And noise could greatly reduce the recognition rate. MIC plays very important role. As for the MIC wire connection to this module, the shorter, the better.

In this test, I recorded 5 voice instructions: white, red, green, blue, off.

First time I use a simple MIC as follows:



I found occasionally this module could not distinguish “red” and “green”. Then I changed the MIC as follows:



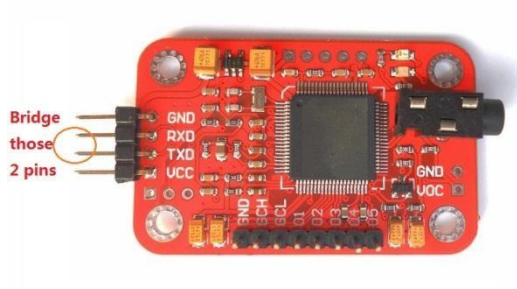
Then the recognition rate is greatly improved.

Also I find that it is a little hard for this module to distinguish “of” and “off”. To distinguish those two similar pronunciations, high-sensitive MIC is required and speaking must be very clear, as well as in quiet environment. All those limitations make low practicality. But there are simple ways to solve it. Make your voice instructions distinctive. Do not use voice instructions of similar pronunciation.

### **Reset the Serial Port**

On V2, we added a way to reset the serial port since many customers would forget the serial parameter some time later after they changed it.

Before powering on the module, bridge the TX and RX pins



Then Power the module on.

### Contact information



Web: [www.elechouse.com](http://www.elechouse.com) Email: [elechouse@elechouse.com](mailto:elechouse@elechouse.com)

### Disclaimer and Revisions

The information in this document may change without notice. Revision History

Rev.	Date	Author	Description
A	Sep. 29 <sup>th</sup> , 2011	Wilson Shen	Initial version
B	Mar. 4 <sup>th</sup> , 2013	Wilson Shen	V2



**Anexo H:**

**Especificaciones técnicas del brazo robótico MICO de la empresa KINOVA**

**MICO**

RESEARCH EDITION

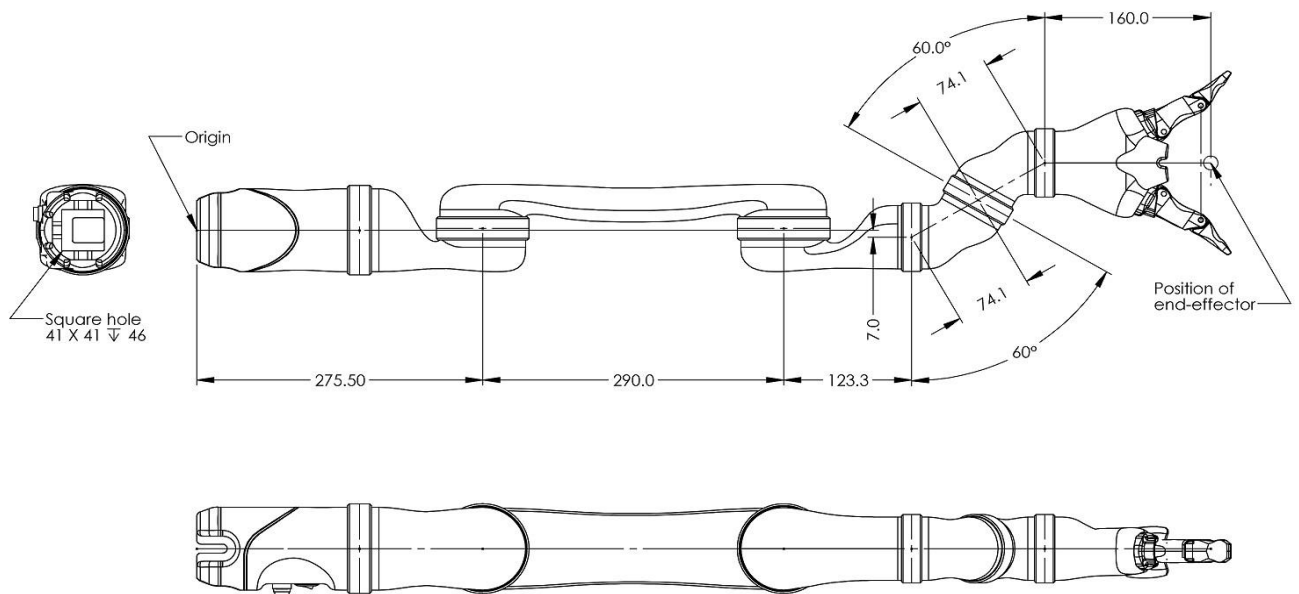


**SPECIFICATIONS**



## MECHANICAL

TOTAL WEIGHT	5.0 Kg
MAXIMUM PAYLOAD	1.25Kg mid-range 750g full extension
MAXIMUM REACH	70 cm
GEAR SYSTEM	monic Drive™
Type	1:136 (large actuator) 1:110 (small actuator)
Gear ratio	When shutdown
Back drivability	
MAXIMUM LINEAR ARM SPEED	20 cm/s
AMBIANT TEMPERATURE	-10 °C to 40 °C expected
WATER RESISTANCE	IPX2 expected



## ELECTRONIC

UTILIZATION VOLTAGE	18 VDC to 29 VDC
AVERAGE POWER	25 W
PEAK POWER	80 W
CONTROL SYSTEM FREQUENCY	500 Hz (Possibility of 1 KHz in open-RS485)
CPU	360 MHz

USB 2.0	12 Mbps
RS485 <b>Internal</b>	11.25 Mbps
CANBUS <b>External</b>	1 Mbps
SLIP RING <b>Location</b>	1 in every actuator
EXPANSION CARDS	
<b>Connectors</b>	2 expansion cards connectors
<b>Protocol</b>	SPI
<b>2 internal / external expansion wires</b>	

## CONTROLLER FEATURES

EACH ACTUATOR CONTROLLED INDEPENDENTLY

REDUNDANT SECURITY ON EACH ACTUATOR/ FINGER

REDUNDANT ERROR CHECK IN ACTUATORS AND CONTROL SYSTEM

POSITION AND ERROR CALCULATION EVERY 0.01 SECOND

SINGULARITY AVOIDANCE SYSTEM

CARTESIAN AND ANGULAR TRAJECTORY CONTROL

## FORCE AND TORQUE CONTROL

## LARGE ACTUATORS - Specifications

QUANTITY IN ARM	3
MOTOR TYPE	Brushless DC motor
GEAR TYPE	Harmonic Drive™ ratio 1:136
DIMENSIONS	
<b>Diameter</b>	74.5 (-0.010 / -0.056) mm
<b>Height</b>	70 mm
<b>Input fixation type</b>	8X M4 screws (10 mm deep)
<b>Output fixation type</b>	8X M4 screws (10 mm deep)
TOTAL WEIGHT	0.6 Kg +/- 3%
NUMBER OF TURNS	+/- 27.7 turns
MOTOR TEMPERATURE	<b>Maximum</b> 80°C
VELOCITY	<b>Maximum</b> 9 RPM
COMMAND/SEC	<b>Maximum</b> 54 ° / sec

UTILIZATION VOLTAGE **18 VDC to 29 VDC**

CURRENT

**Continuous** **0.75 A**

**Peak** **1.5 A**

TORQUE

**Continuous** **9.5 Nm**

**Peak / Momentary peak** **18 Nm / 29 Nm**

ABSOLUTE POSITION SENSOR

**At Start-up** **+/- 1.5°**

**Once indexed** **+/- 0.040°**

*(After 7 degrees movement max in any direction)*

RELATIVE HALL EFFECT POSITION SENSOR **+/- 0.055°**

*(Motor side sensor, equivalent output precision)*

RELATIVE ENCODER POSITION SENSOR **+/- 0.020°**

*(After reducer output encoder)*

TORQUE SENSOR

**Cross-Axis Sensitivity** **0 - 8%**

**Precision** **+/- 0.4 Nm (At room temp)**

**Temperature error** **+/- 0.3 Nm (Over temperature range)**

OTHER SENSORS

**Accelerometer** **+/- 3g in X, Y, Z. 50Hz**

**Actuator current** **Bandwidth +/- 5 A Range.**

**Temperature** **+/- 2°C Max. Accuracy**

Note: Torque, current and velocity values are based on 24V alimentation

## SMALL ACTUATORS - Specifications

QUANTITY IN ARM **3**

**Brushless DC motor**

MOTOR TYPE

GEAR TYPE **Harmonic Drive™ ratio 1:110**

DIMENSIONS

**Diameter** **58 (-0.010 / -0.056) mm**

**Height** **65 mm**

**Input fixation type** **6X M3 screws (4.5 mm deep)**

**Output fixation type** **6X M3 screws (4.5 mm deep)**

TOTAL WEIGHT **0.36 Kg +/- 3%**

NUMBER OF TURNS **+/- 27.7 turns**

TEMPERATURE **Maximum** **80°C**

VELOCITY	<b>Maximum</b>	<b>18 RPM</b>
COMMAND/SEC	<b>Maximum</b>	<b>108 ° / sec</b>
UTILIZATION VOLTAGE		<b>18 VDC to 29 VDC</b>
CURRENT		
<b>Continuous</b>		<b>0.8 A</b>
<b>Peak</b>		<b>1.6 A</b>
TORQUE		
<b>Continuous</b>		<b>3.8 Nm</b>
<b>Peak / Momentary peak</b>		<b>9.8 Nm / 15.7 Nm</b>
ABSOLUTE POSITION SENSOR		
<b>At Start-up</b>		<b>+/- 1.5°</b>
<b>Once indexed</b>		<b>+/- 0.040°</b>
<i>(After 7 degrees movement max in any direction)</i>		
RELATIVE HALL EFFECT POSITION SENSOR		<b>+/- 0.068°</b>
<i>(Motor side sensor, equivalent output precision)</i>		
RELATIVE ENCODER POSITION SENSOR		<b>+/- 0.020°</b>
<i>(After reducer output encoder)</i>		
TORQUE SENSOR		
<b>Cross-Axis Sensitivity</b>		<b>0 - 8%</b>
<b>Precision</b>		<b>+/- 0.4 Nm (At room temp)</b>
<b>Temperature error</b>		<b>+/- 0.3 Nm (Over temperature range)</b>
OTHER SENSORS		
<b>Accelerometer</b>		<b>+/- 3g in X, Y, Z. 50Hz</b>
<b>Actuator current</b>		<b>Bandwidth +/- 5 A Range.</b>
<b>Temperature</b>		<b>+/- 2°C Max. Accuracy</b>

Note: Torque, current and velocity values are based on 24V alimentation

## FINGER ACTUATORS - Specifications

QUANTITY IN ARM		<b>2</b>
MOTOR TYPE		<b>Brushless DC motor</b>
GEAR TYPE		<b>Lead screw – 1 mm pitch</b>
TOTAL WEIGHT		<b>0.13 Kg +/- 3%</b>
MOTOR TEMPERATURE	<b>Maximum</b>	<b>80 °C</b>
VELOCITY	<b>Linear max.</b>	<b>30 mm / sec</b>
TRAVEL		<b>21 mm</b>
UTILIZATION VOLTAGE		<b>18 VDC to 29 VDC</b>

<b>CURRENT</b>	
<b>Continuous</b>	<b>0.4 A</b>
<b>Peak</b>	<b>1.4 A</b>
<b>FORCE</b>	
<b>Continuous</b>	<b>60 N</b>
<b>Peak</b>	<b>120 N</b>
<b>RELATIVE POSITION SENSOR</b>	<b>+/- 0.017 mm</b>
<b>OTHER SENSORS</b>	
<b>Current</b>	<b>+/- 2 A Range.</b>
<b>Temperature</b>	<b>+/- 2 °C Max. Accuracy</b>

Note: Torque, current and velocity values are based on 24V alimentation

## GRIPPER

<b>FINGERS</b>	<b>2</b>
<b>Quantity</b>	<b>Under actuated</b>
<b>Actuation system</b>	<b>One per finger</b>
<b>Actuators</b>	<b>1 current sensor / actuator</b>
<b>Sensors</b>	<b>1 temperature sensor / actuator</b>
	<b>1 optical encoder position sensor / actuator</b>

## API AND SOFTWARE

<b>API</b>	
<b>Compatibility</b>	<b>Windows, Linux Ubuntu &amp; ROS</b>
<b>Port</b>	<b>USB 2.0</b>
<b>Programming languages</b>	<b>C# &amp; C++</b>



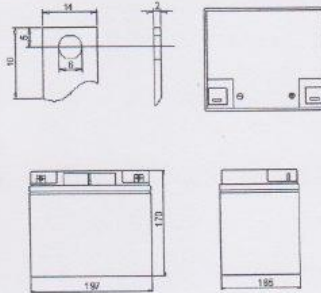
# Anexo I

## Datos técnicos de la batería para alimentación del subsistema del robot asistivo



Sealed Lead-Acid Batteries

BKM(12V-40)

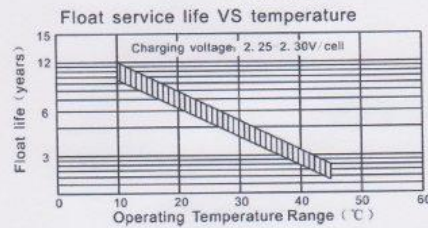
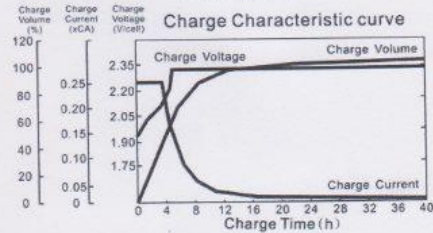
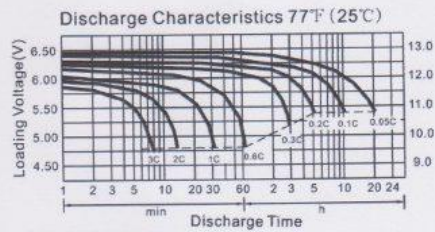


● Specifications

Nominal Voltage		12V
Rated Capacity 77°F(25°C) (10HR)		40.0Ah
Dimensions (mm/inch)	Length	197 (7.76)
	Width	165 (6.50)
	Height	170 (6.69)
	Total Height	170 (6.69)
Approx. Weight (kg/lbs)		12.6 (27.78)
Terminal		T4/T11

● Characteristic

Capacity 77°F(25°C)	20HR (2.12A)	42.4Ah
	10HR (4.00A)	40.0Ah
	5HR (6.80A)	34.0Ah
	1HR (22.0A)	22.0Ah
	15 min rate (70.0A)	17.5Ah
Internal Resistance	Full Charged Battery	Approx. 9.5mΩ
	77°F(25°C)	
Temperature dependence of capacity (10HR)	104°F(40°C)	102%
	77°F(25°C)	100%
	32°F(0°C)	85%
Self-Discharge (Capacity after)	3 months	90%
	6 months	80%
	12 months	60%
Max. Discharge Current, 77°F(25°C)		320A(5s)
Floating design life, 77°F(25°C)		10 years
Constant Voltage Charge, 77°F(25°C)	Cycle	14.4-14.7V(-24mV/°C) max. current: 10 A
	Float	13.6-13.8V(-18mV/°C)



Constant Current Discharge Characteristics (A), 77°F(25°C)

F.V/TIME	5min	10min	15min	30min	60min	2H	3H	5H	8H	10H	20H
1.60V/cell	136.0	88.0	70.0	40.0	23.2	14.41	10.62	7.18	4.80	4.17	2.18
1.70V/cell	129.2	83.6	66.9	38.3	22.4	13.86	10.38	7.20	4.75	4.12	2.16
1.75V/cell	123.2	78.4	65.7	37.5	22.0	13.60	10.18	6.92	4.65	4.06	2.14
1.80V/cell	115.6	72.4	64.5	36.7	21.6	13.36	10.00	6.80	4.60	4.00	2.12

Constant Wattage Discharge Characteristics (Watt), 77°F(25°C)

F.V/TIME	5min	10min	15min	30min	60min	2H	3H	5H	8H	10H	20H
1.60V/cell	241.4	159.8	128.3	73.6	43.0	26.90	20.01	13.65	9.22	8.14	4.27
1.70V/cell	231.5	153.3	123.7	71.1	41.8	26.11	19.73	13.51	9.19	8.11	4.26
1.75V/cell	222.8	145.1	122.5	70.3	41.5	25.84	19.51	13.38	9.11	8.05	4.25
1.80V/cell	211.0	135.2	121.5	69.5	41.1	25.62	19.33	13.26	9.05	8.00	4.24

## Anexo J

### Cotizaciones

#### MODULO ADAPTADOR I2C PARA PANTALLA LCD



The screenshot shows the product page for an I2C adapter module on the electrotec website. The page features the electrotec logo (Electrónica & Robótica) and a search bar. The navigation menu includes: INICIO | ARDUINO | TECNOLOGIA | DRONES | PRODUCTOS | APRENDE | UBICANOS. The product image shows a blue PCB with a microcontroller and various components. The product title is "ADAPTADOR I2C". The description states: "Esta interfaz I2C para LCDs alfanuméricos te permite conectar una pantalla mediante el protocolo de comunicación I2C, usando únicamente 2 pines digitales (SDA y SCL)." The price is listed as "S/. 12.00". A blue button labeled "AGREGAR AL CARRITO" is present. Below the button, there is a "COMPARTIR ESTE PRODUCTO:" section with social media sharing options for Facebook and Twitter.

#### PANTALLA LCD 16 X 02



The screenshot shows the product page for a 16x2 LCD display on the electrotec website. The page features the electrotec logo (Electrónica & Robótica) and a search bar. The navigation menu includes: INICIO | ARDUINO | TECNOLOGIA | DRONES | PRODUCTOS | APRENDE | UBICANOS. The product image shows a green PCB with a blue LCD screen. The product title is "LCD 16X2". The description states: "Pantalla LCD de 16x2 con fondo azul". The price is listed as "S/. 14.00". A blue button labeled "AGREGAR AL CARRITO" is present. Below the button, there is a "COMPARTIR ESTE PRODUCTO:" section with social media sharing options for Facebook and Twitter.

## TESTER PARA BATERIA LIPO , “SALVA LIPO”



electrotec  
Electrónica & Robótica

Buscar...

INICIO | ARDUINO | TECNOLOGIA | DRONES | PRODUCTOS ▾ | APRENDE | UBICANOS

**TESTER DE BATERIA LIPO**

Módulo testeador de baterías LIPO

S/. 20.00

## MÓDULO DE COMUNICACIÓN INALÁMBRICA NRF24L01



electrotec  
Electrónica & Robótica

Buscar...

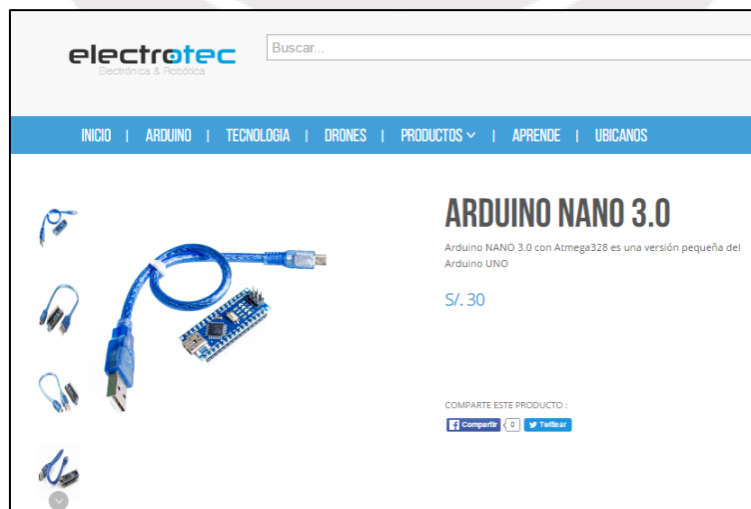
INICIO | ARDUINO | TECNOLOGIA | DRONES | PRODUCTOS ▾ | APRENDE | UBICANOS

**MODULO RF NRF24L01 CON ANTENA**

Módulo NRF24L01 con antena que permite alcanzar hasta 1 Km de distancia.

S/. 35.00

## ARDUINO NANO



electrotec  
Electrónica & Robótica

Buscar...

INICIO | ARDUINO | TECNOLOGIA | DRONES | PRODUCTOS ▾ | APRENDE | UBICANOS

**ARDUINO NANO 3.0**

Arduino NANO 3.0 con Atmega328 es una versión pequeña del Arduino UNO

S/. 30

COMPARTE ESTE PRODUCTO:

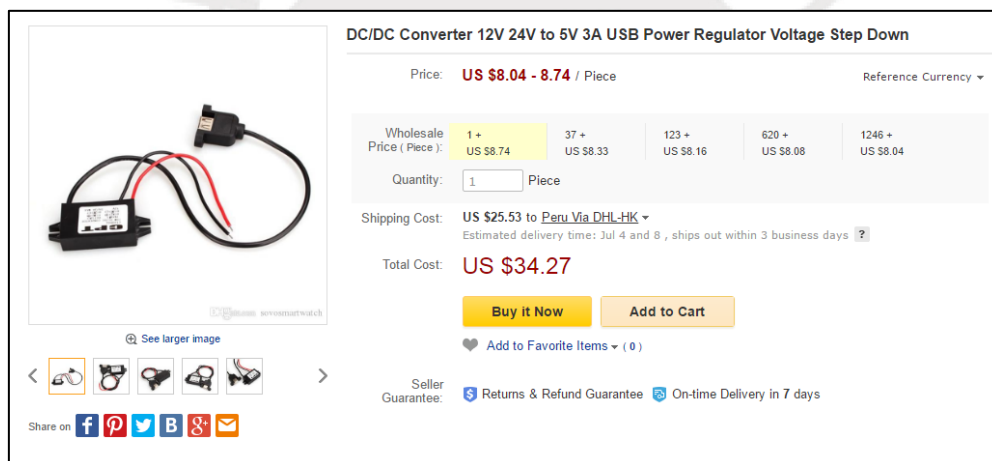
[Compartir](#) [Twitter](#)

## MODULO DE RECONOCIMIENTO DE VOZ



The screenshot shows the Electrotec website interface. At the top, there is a search bar and a navigation menu with links for INICIO, ARDUINO, TECNOLOGIA, DRONES, PRODUCTOS, APRENDE, and UBICANOS. The main content area features a large image of a red voice recognition module with a black microphone. To the right of the image, the product title "MODULO DE RECONOCIMIENTO DE VOZ" is displayed in large, bold letters. Below the title, a short description states: "módulo conectado a un Arduino podrás empezar a controlar diferentes dispositivos eléctricos o electrónicos solamente hablando. Disponible sólo por compra web." The price is listed as "S/. 120.00" and there is a blue button labeled "AGREGAR AL CARRITO". At the bottom, there are social media sharing options for Facebook, Twitter, and YouTube.

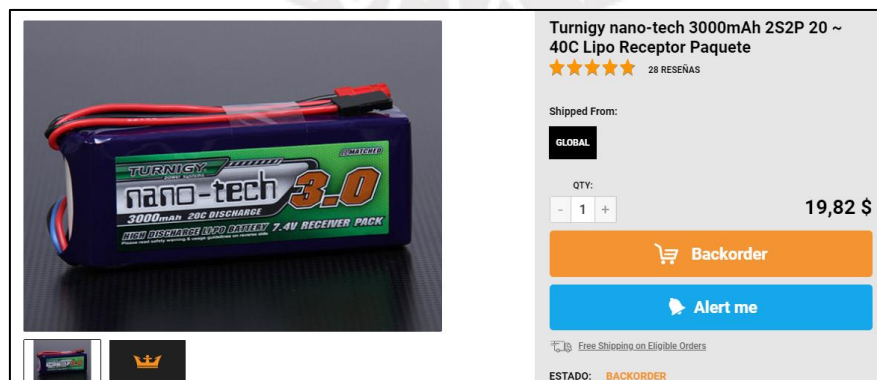
## REGULADO 24V – 5V MARCA CPT CON CONECTOR USB



The screenshot shows a product page for a "DC/DC Converter 12V 24V to 5V 3A USB Power Regulator Voltage Step Down". The product image shows a black rectangular module with a USB connector and several wires. The price is listed as "US \$8.04 - 8.74 / Piece". A table shows wholesale prices for different quantities: 1+ (US \$8.74), 37+ (US \$8.33), 123+ (US \$8.16), 620+ (US \$8.08), and 1246+ (US \$8.04). The quantity is set to 1. The shipping cost is "US \$25.53 to Peru Via DHL-HK" with an estimated delivery time of Jul 4 and 8. The total cost is "US \$34.27". There are buttons for "Buy it Now" and "Add to Cart". The seller guarantee includes "Returns & Refund Guarantee" and "On-time Delivery in 7 days".

Wholesale Price ( Piece )	1 +	37 +	123 +	620 +	1246 +
	US \$8.74	US \$8.33	US \$8.16	US \$8.08	US \$8.04

## BATERIA LiPo 2S 4600mAh



The screenshot shows a product page for a "Turnigy nano-tech 3000mAh 2S2P 20 ~ 40C Lipo Receptor Paquete". The product image shows a blue and red LiPo battery pack. The price is listed as "19,82 \$". There are buttons for "Backorder" and "Alert me". The seller guarantee includes "Free Shipping on Eligible Orders". The status is "ESTADO: BACKORDER".



## BATERÍA 12V 40ah MARCA KM

Bateria Km 12v-40ah Sellada Recargable  

Artículo nuevo



**U\$S 78<sup>84</sup>**

Finaliza en 1 d 13 h

 Pago a acordar con el vendedor.

 Entrega a acordar con el vendedor  
Lima  
[Consultar costos](#)

Cantidad:    [Comprar](#)   

## CORREA NYLON DE 25 mm



*XiWarm*

Venta al por mayor precio 25 mm \* 5 m de Nylon de los rollos de cinta de las correas para hacer bricolaje flejado mochila delantal Bunting ajuste artículos para el hogar

 Ver nombre original del producto en Inglés

★★★★★5.0 (16 votos) | 31 vendidos

Precio: €2.59 / unidad

Oferta: **€ 1,82** / unidad **30% off** 12h:30m:24s

Más ofertas te esperan en la app

Color:

Envío: **€ 0,41 a Peru** vía China Post Registered Air Mail   
Tiempo de entrega: 15-30 días (Se envía en 5 días hábiles)

Cantidad:    unidad (250 unidades available)

Precio total: Depende de las propiedades del producto que selecciones

[Comprar ahora](#) [Añadir a la cesta](#) [Vistos recientemente](#)

## HEBILLAS DE PLASTICO



MagiDeal Liberación Laterales 10pcs Hebillas De Plástico Para Cinchas 25mm

Marca [MagiDeal](#) | [Más Accesorios para Coser MagiDeal](#) ★★★★★ No hay reseñas | Escribe tu reseña

~~S/ 36.00~~ - 19%  
**S/ 29.00**  
Ahorra S/ 7.00

[COMPRAR AHORA](#)

 PROTECCIÓN AL COMPRADOR

 ¿NECESITAS AYUDA? (01) 6400-240

 DESCARGA NUESTRA APP

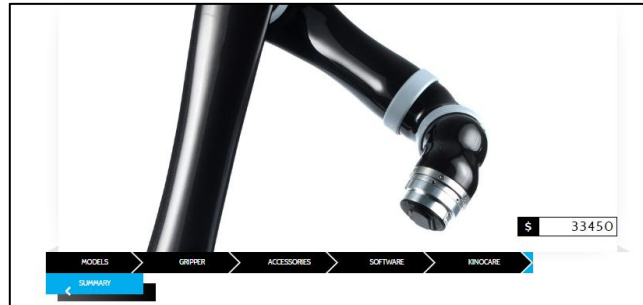
 Envío Internacional

¡Disfruta lo mejor del mundo con Garantía Linió!

- o Garantía Linió: Satisfacción Linió de 14 días naturales.
- o Devoluciones: 14 días con Garantía Linió. Aplican restricciones.
- o Cancelaciones: Antes de confirmar el envío.
- o Garantía adicional: Sujeta a la marca o vendedor.
- o Linió te ayuda con permisos especiales.
- o Compras deben realizarse con DNI o RUC10 (Persona Natural).
- o Es necesario tener RUC en estado "Activo y Habido"

Conoce tu RUC

# ROBOT MICO ARM DE LA EMPRESA KINOVA



SUMMARY	
<b>Model</b>	
6 DOF (qty: 1)	\$ 29950
<b>Gripper</b>	
KG-2 (qty: 1)	\$ 3500
<b>Accessories</b>	
USB Cable (qty: 1)	Included
Power supply unit (PSU) (qty: 1)	Included
Battery power cord (qty: 1)	Included
Power cord (qty: 1)	Included
Joystick (qty: 1)	Included
Mounting plate (qty: 1)	Included
Carrying case (qty: 1)	Included
<b>Software</b>	
Kinova	Included
<b>Kinocare</b>	
Kinocare (base coverage)	Included
<b>TOTAL</b>	<b>33450</b>



## IMPRESIÓN 3D

### COTIZACIÓN SERVICIO DE IMPRESIÓN 3D

24.06.2016

Estimado Joel Escalante a continuación la cotización de tus piezas detallada:

ARCHIVO	SERVICIO	IMPRESORA	MATERIAL	PRECIO	TOTAL
Case_pcdiino_botton	1	Fortus	ABS	S/. 439.7	S/. 1,971.6
case_pcdiino_top	1	Fortus	ABS	S/. 447.5	
estuche_base	1	Fortus	ABS	S/. 432.7	
estuche_tapa	1	Fortus	ABS	S/. 177.7	
estuche_tapa_Tipo2	1	Fortus	ABS	S/. 403.8	
tapa_bateria	1	Fortus	ABS	S/. 70.2	
Case_pcdiino_botton (partida en 2)	1	Bee The First	PLA	S/. 63.1	S/. 338.6
case_pcdiino_top (partida en 2)	1	Bee The First	PLA	S/. 66.0	
estuche_base (partida en 2)	1	Bee The First	PLA	S/. 59.3	
estuche_tapa (partida en 2)	1	Bee The First	PLA	S/. 23.1	
estuche_tapa_Tipo2 (partida en 2)	1	Bee The First	PLA	S/. 48.8	
tapa_bateria (partida en 2)	1	Bee The First	PLA	S/. 18.3	
Horas de pegado de piezas (partida en 2)	3			S/. 60.0	
base_Inte	1	Bee The First	PLA	S/. 27.6	S/. 52.2
tapa_Inter	1	Bee The First	PLA	S/. 24.7	

## Anexo K

### Prueba de Funcionalidad del sistema de asistencia mediante reconocimiento de Voz

#### Registro de prueba de funcionamiento de sistema de asistencia de control por Voz

**Información**

Persona que registra la prueba: Joel Escalante Zelada
Codigo PUCP: 20097025

Fecha de prueba: 25/5/2016

---

**Prueba de funcionamiento del Sub sistema denominado "Reconocimiento de Voz"**

En esta etapa se realizará la prueba de funcionamiento del subsistema enargado de descifrar los comandos de voz emitidos por el usuario y trasladar las ordenes al controlador del sistema de reconocimiento de Voz. Por ello, la persona encargada de realizar las prueba emitira las comandos de voz y validará su detección por el subsistema.

Comando a Reconocer		Se ejecutó	No se ejecutó	Observaciones
Comando	Sonido de voz	✓	X	
1,1	"Bebida"	✓		<i>Los comandos se reconocieron sin problemas</i>
1,2	"Boca"	✓		
1,3	"Para"	✓		
Llave 1	"Brazo"	✓		
Llave 2	"Mano"	✓		
2,1	"Arriba"	✓		
2,2	"Abajo"	✓		
2,3	"Derecha"	✓		
2,4	"Izquierda"	✓		
2,5	"Gira"	✓		
3,1	"Adelante"	✓		
3,2	"Atrás"	✓		
3,3	"Abre"	✓		
3,4	"Cierra"	✓		
3,5	"Rota"	✓		

---

**Prueba de ejecución de movimientos del robot de asistencia mediante el ordenador**

En esta etapa se realizará la prueba de funcionamiento del movimiento del brazo robótico "MICO" a través de un ordenador, el cual será representará el papel de un PCDuino. Por lo cual la persona encargada de realizar la pruebas registrará la ejecución de los movimientos del robot según el comando programado.

Movimientos del brazo Robótico	Se ejecutó	No se ejecutó	Observaciones
	✓	X	
Movimiento en el eje x+	✓		<i>Los movimientos fueron ejecutados sin observaciones</i>
Movimiento en el eje x-	✓		
Movimiento en el eje y+	✓		
Movimiento en el eje y-	✓		
Movimiento de rotación del brazo	✓		
Movimiento de rotación de la muñeca	✓		
Abrir pinza	✓		
Cerrar pinza	✓		
Agarrar una bebida	✓		

Llevar la bebida a la boca

**Prueba de Integración del sistema de reconocimiento de voz: ejecución de movimientos del robot mediante una orden de voz.**

En esta etapa se realizará la prueba de funcionamiento de todo el sistema integrado, donde la persona encargada de realizar la prueba emitirá una orden de voz y este deberá ser ejecutado por el brazo robótico "MICO". Para este caso el controlador del Robot "PCDuino" será representado por un ordenador (Laptop), quien a través del programa Visual Studio y con ayuda de librerías realizará el movimiento del brazo previa comunicación con el subsistema denominado "Reconocimiento de Voz" comandado por un Arduino Nano.

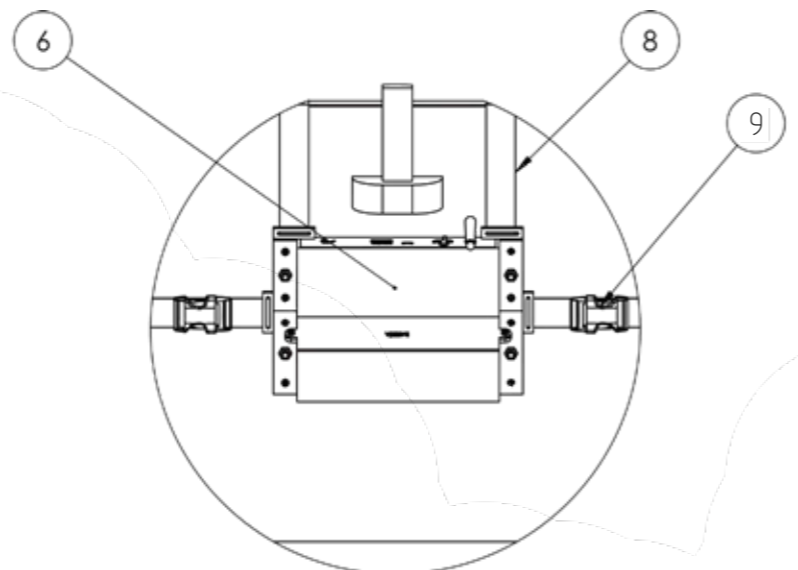
Movimientos del brazo Robótico	✓	X	Observaciones
Abrir pinza	✓		
Cerrar pinza	✓		
Agarrar bebida	✓		
Tomar la bebida	✓		

**Observaciones Finales**

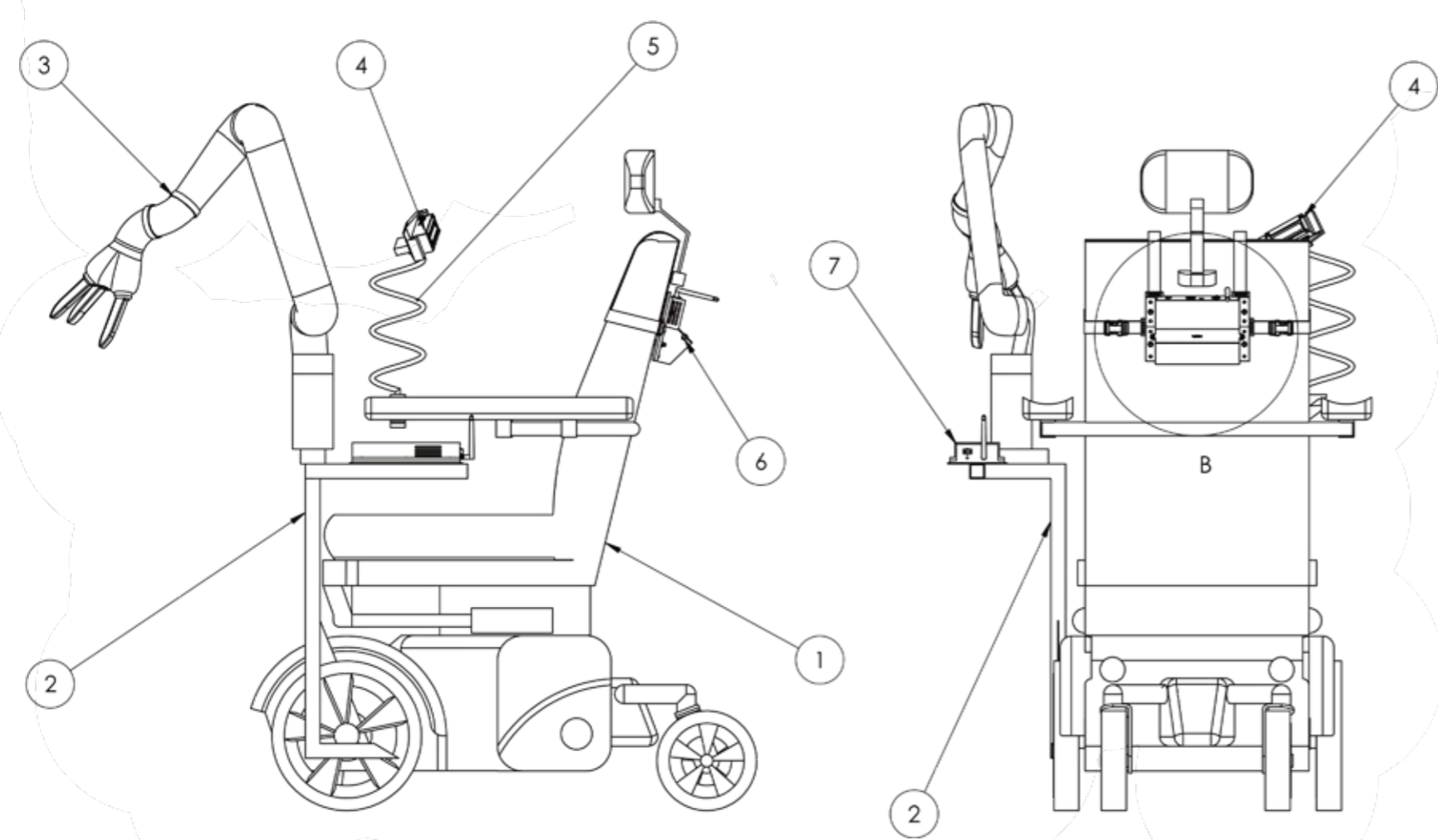
Las pruebas se realizaron sin observación alguna, evidenciando una correcta comunicación entre el subsistema de "Reconocimiento de voz" y el brazo robótico MICO, logrando ejecutarse con efectividad todos los comandos mencionados.

\* Este documento tiene como finalidad sustentar el funcionamiento del sistema Mecatrónico propuesto en el tema de tesis "Sistema de asistencia que permite la manipulación de un mecanismo robótico mediante la Voz"

*[Handwritten Signature]*  
 Joel Escobar E.



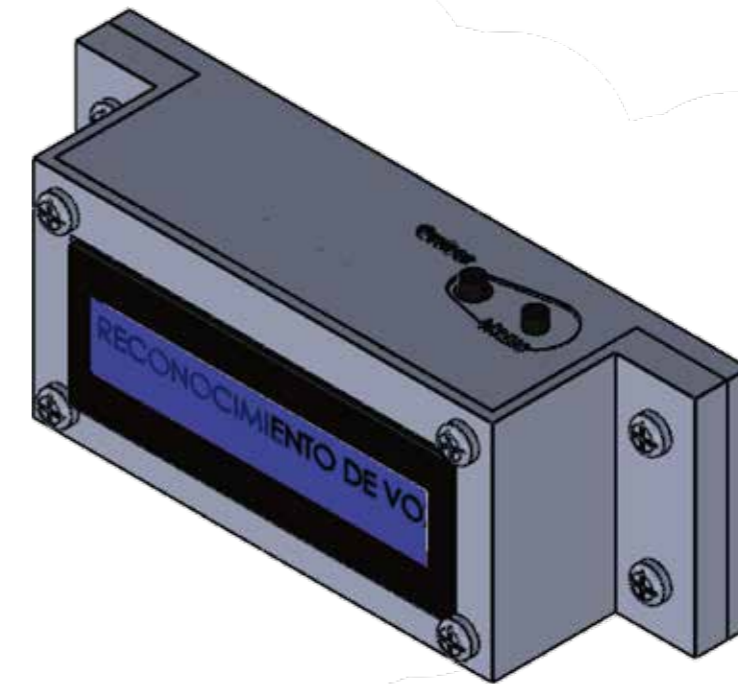
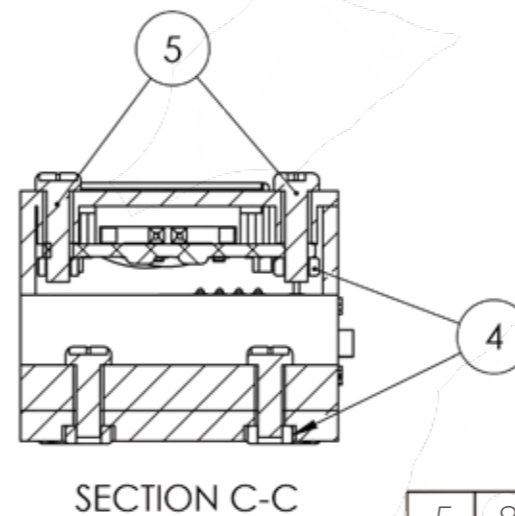
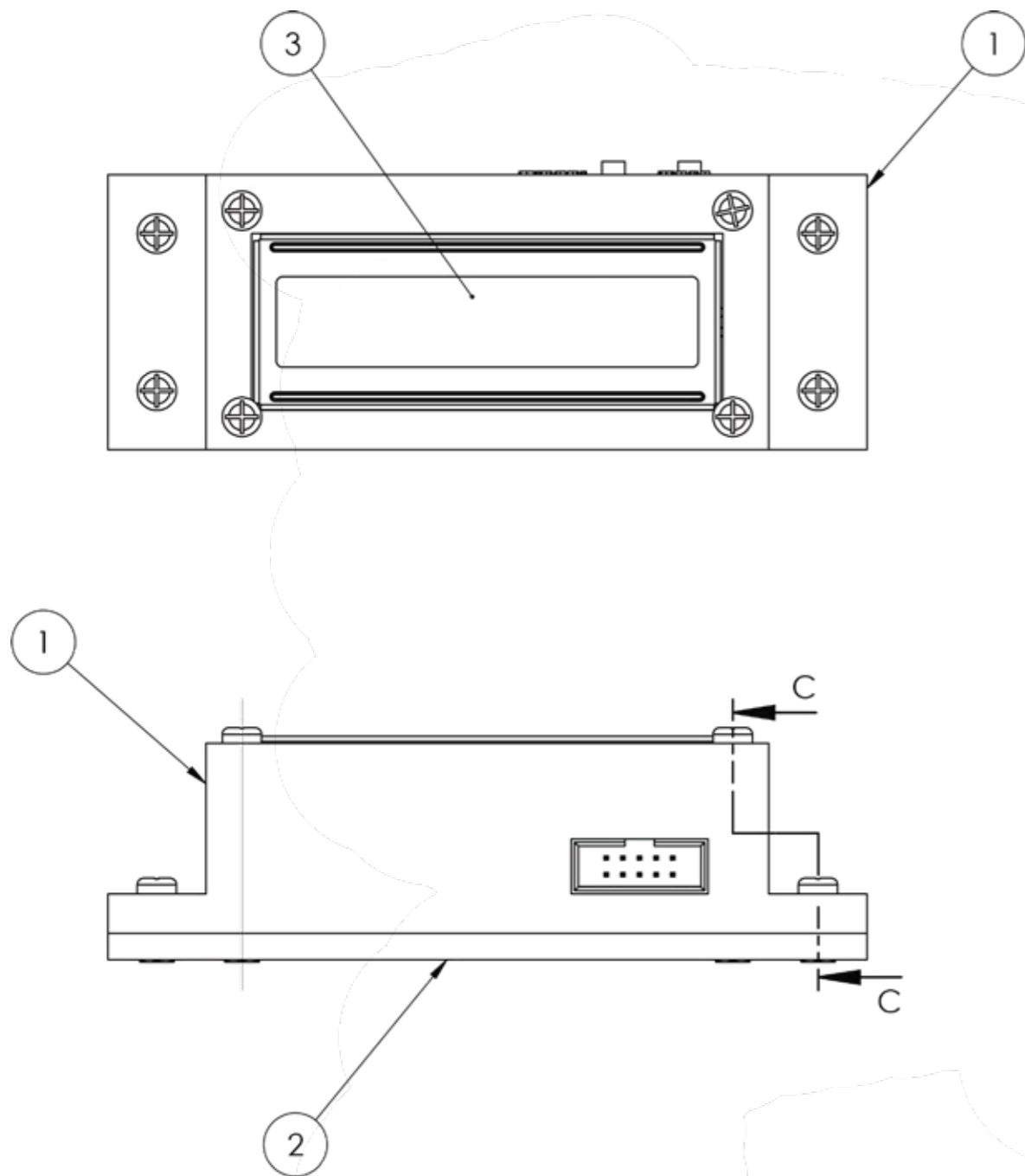
DETAIL B  
SCALE 1 : 5



9	2	HEBILLAS		PLÁSTICO	
8	1	CORREA DE NYLON		NYLON	
7	1	ESTUCHE CONTROL DEL ROBOT			LÁMINA A3-11
6	1	ESTUCHE DE REC. DE VOZ			LÁMINA A3-10
5	1	SOPORTE DE INTERFÁZ DE USUARIO			
4	1	ESTUCHE INTERFÁZ USUARIO			LÁMINA A3-09
3	1	ROBOT MICO ARM			
2	1	SOPORTE DE BRAZO ROBÓTICO			
1	1	SILLA DE RUEDA ELÉCTRICA			
POS.	CANT.	DESCRIPCIÓN	NORMA	MATERIAL	OBSERVACIONES

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA			TRABAJO DE FIN DE CARRERA	
MÉTODO DE PROYECCIÓN 		SISTEMA DE ASISTENCIA DE CONTROL POR VOZ		ESCALA 1:10
20097025		ESCALANTE ZELADA, JOEL EDUARDO		FECHA: 2020.06.15
				LÁMINA: A2-12

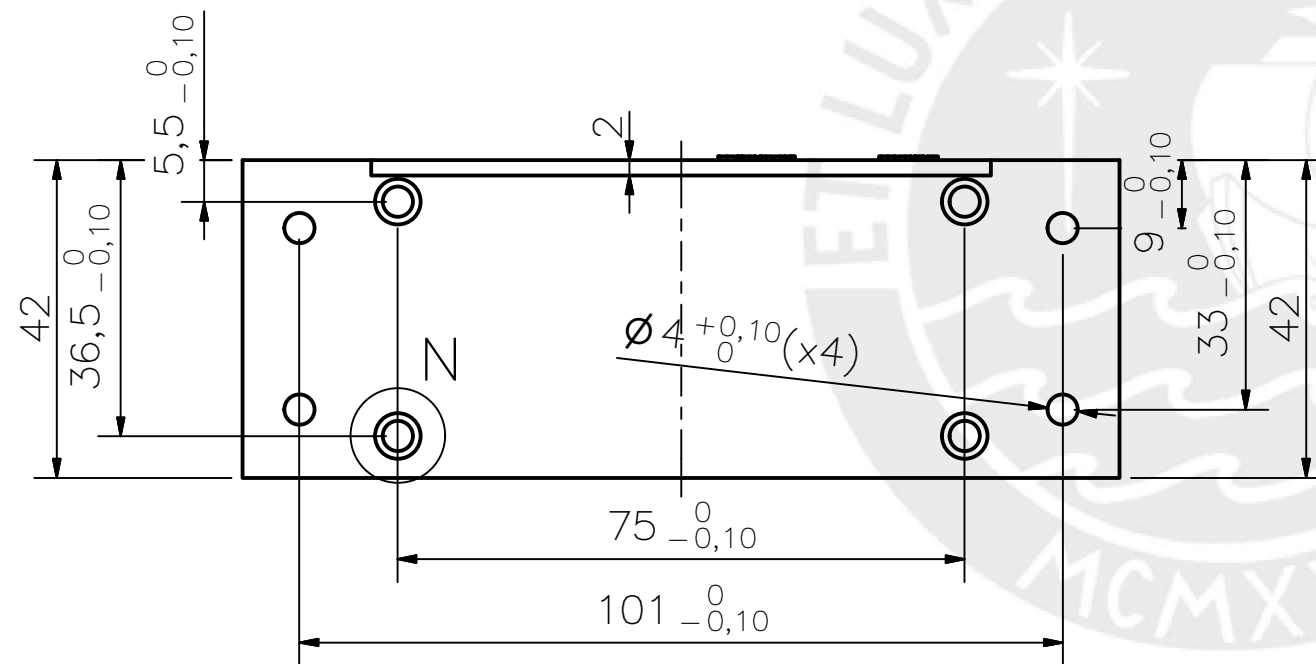
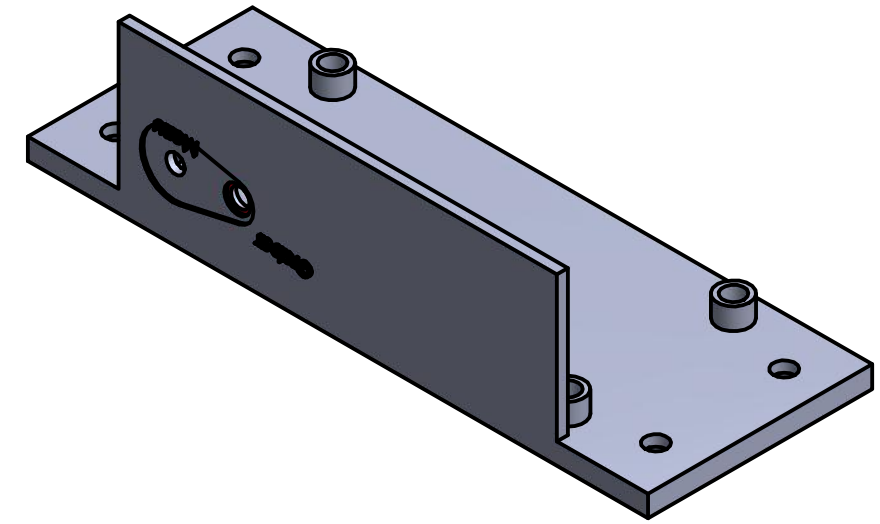
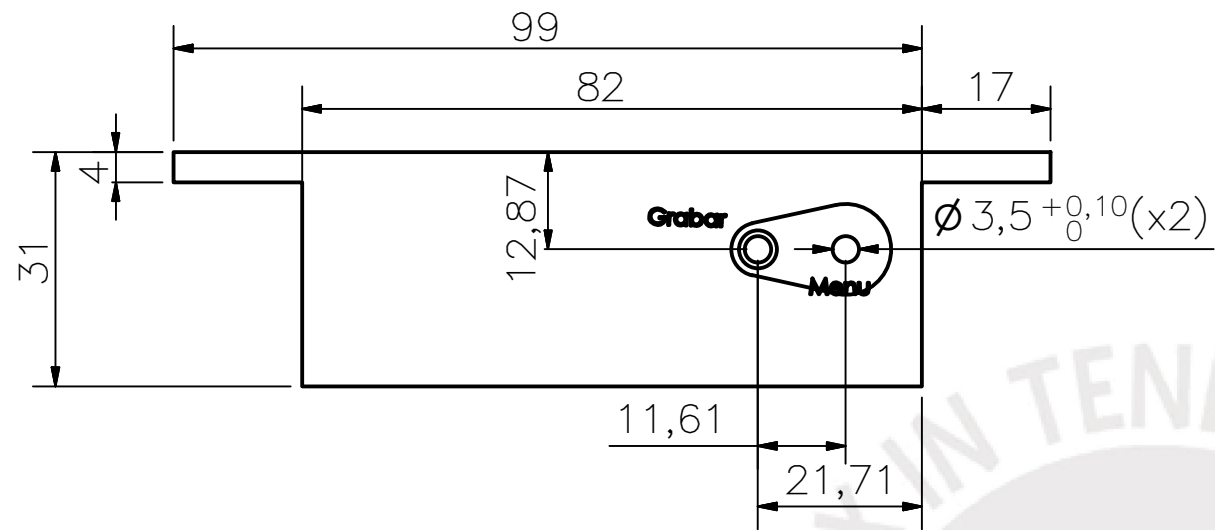




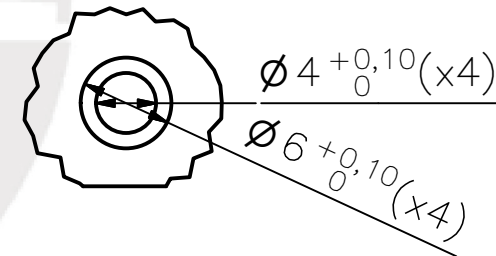
5	8	TORNILLO M3X12	DIN 7985	STEEL	
4	8	TUERCA HEXAGONAL M3	DIN 934	STEEL	
3	1	PANTALLA LCD 16X2			
2	1	BASE INTERFAZ DE USUARIO		ABS	LÁMINA A3-08
1	1	TAPA INTERFAZ DE USUARIO		ABS	LÁMINA A3-07
POS.	CANT.	DESCRIPCIÓN	NORMA	MATERIAL	OBSERVACIONES

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
 FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO DE FIN DE CARRERA	ESCALA
	<b>ESTUCHE INTERFAZ DE USUARIO</b>	1:1
20097025	ESCALANTE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-09



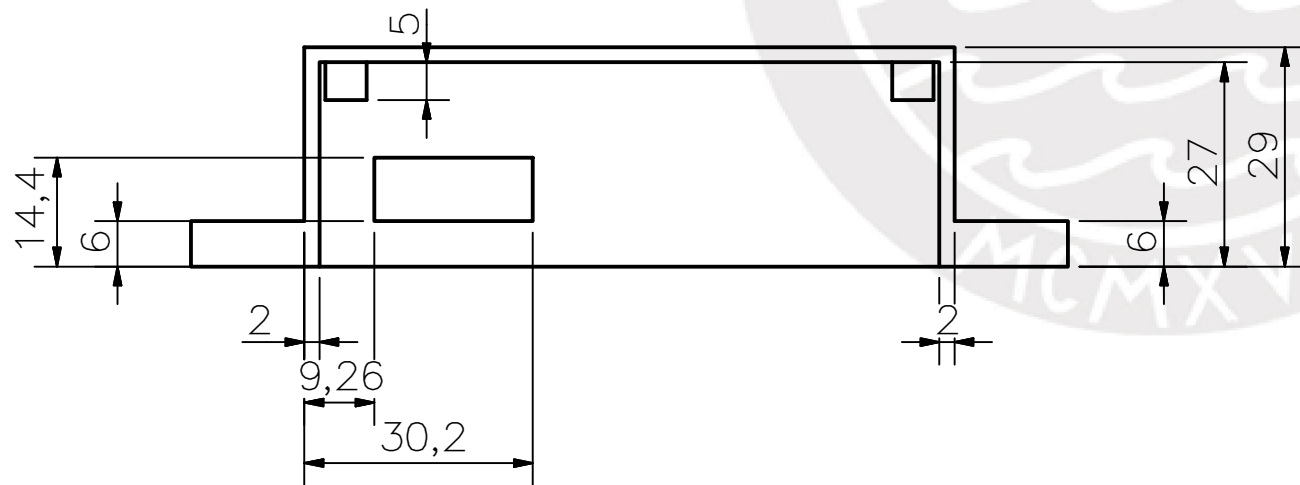
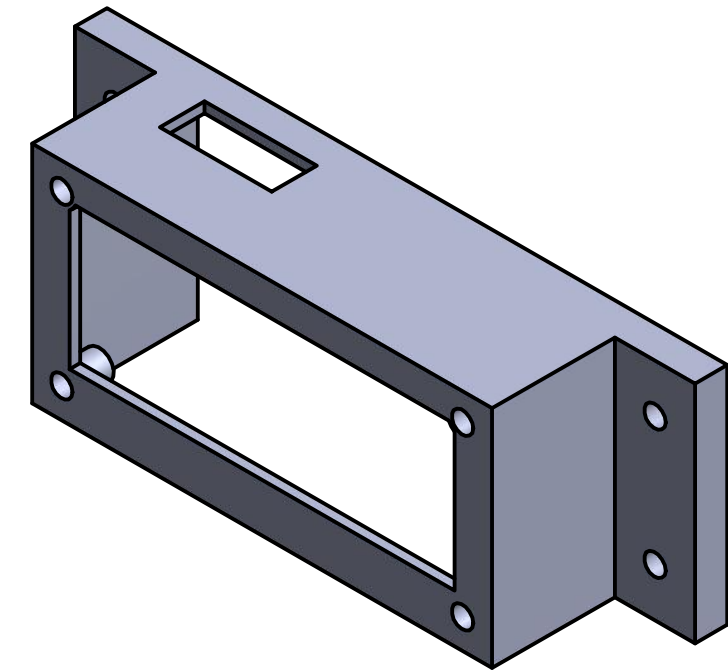
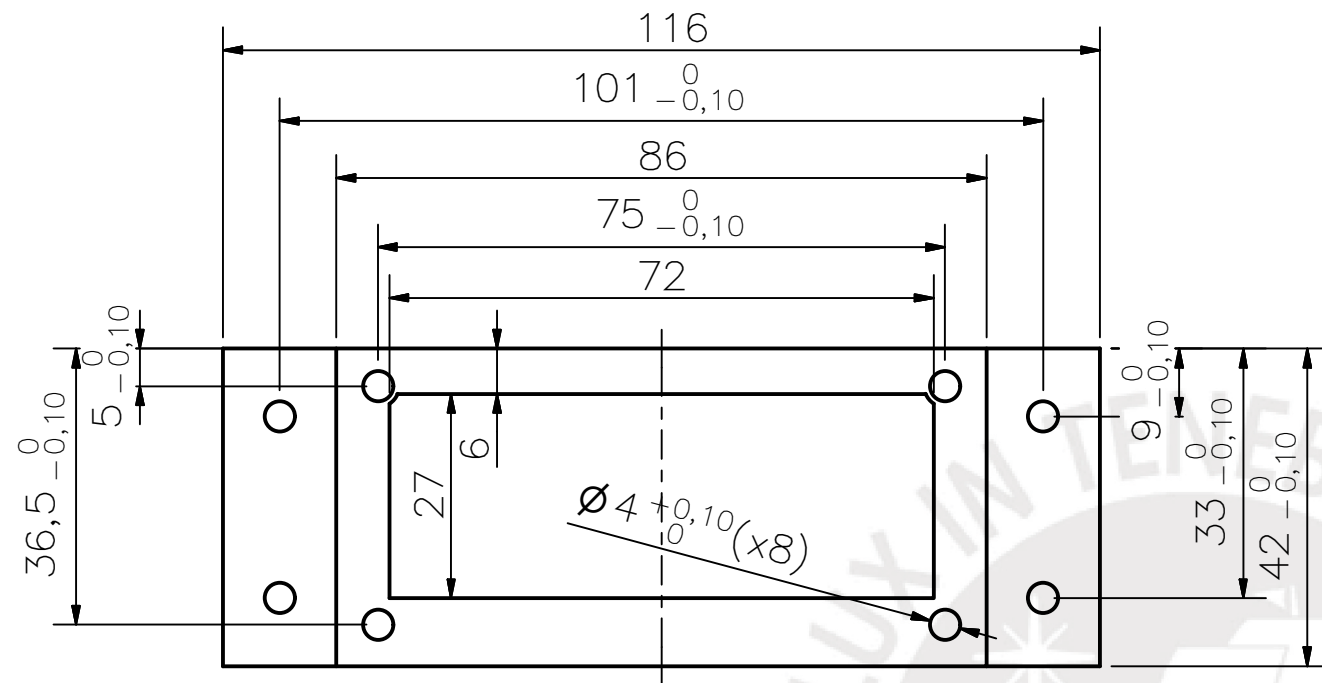
N ( 2 : 1 )



TOLERANCIAS DIMENSIONALES SEGUN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
FINO	±0,05	±0,05	±0,1	±0,15	±0,2

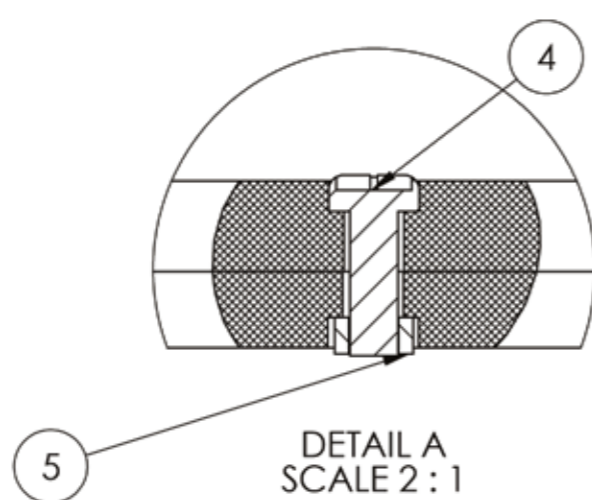
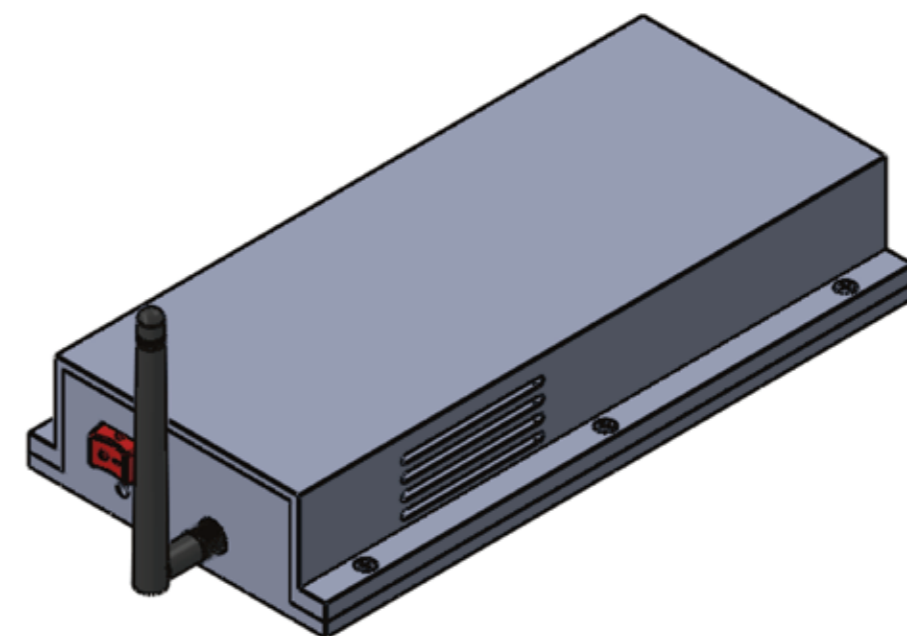
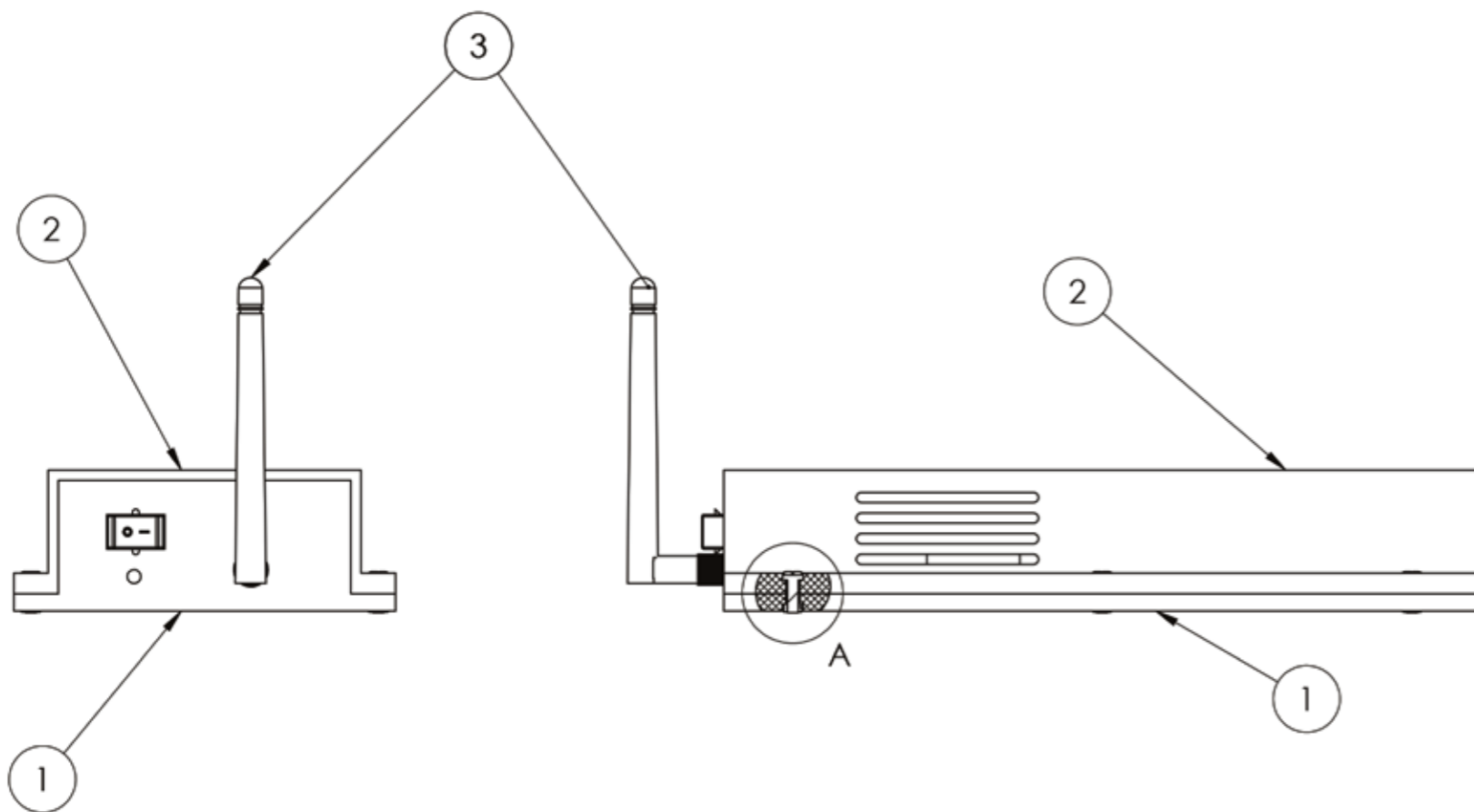
ACABADO SUPERFICIAL ✓	TOLERANCIA GENERAL FINO SEGÚN DIN 7168	MATERIAL ABS
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA		
MÉTODO DE PROYECCIÓN ⊕ ⊞	TRABAJO DE FIN DE CARRERA BASE INTERFAZ DE USUARIO	ESCALA 1:1
20097025	ESCALANTE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-08





TOLERANCIAS DIMENSIONALES SEGUN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
FINO	±0,05	±0,05	±0,1	±0,15	±0,2

ACABADO SUPERFICIAL ✓	TOLERANCIA GENERAL FINO SEGÚN DIN 7168	MATERIAL ABS
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA		
MÉTODO DE PROYECCIÓN ⊕ ⊖	TRABAJO DE FIN DE CARRERA TAPA INTERFAZ DE USUARIO	ESCALA 1:1
20097025	ESCALATE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-07

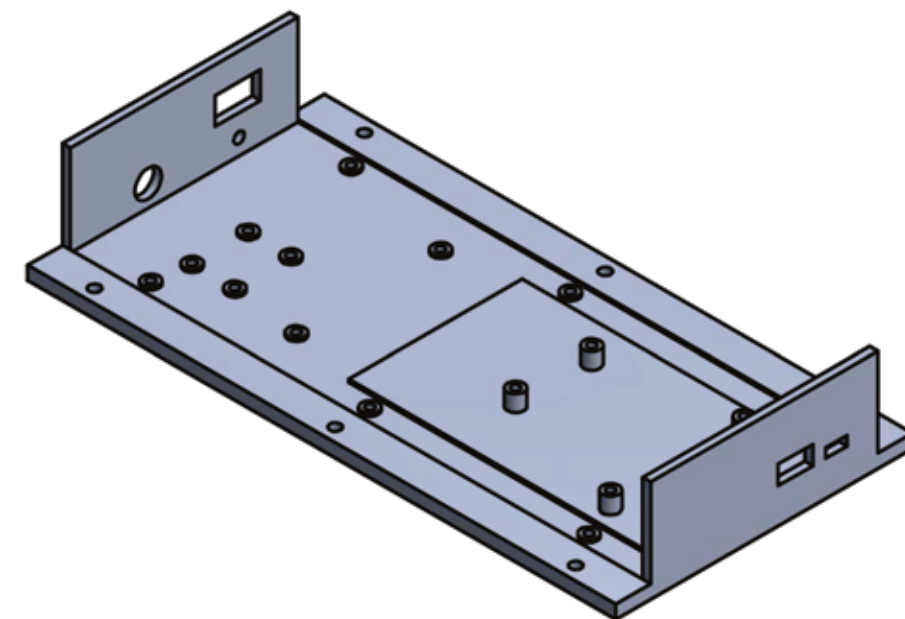
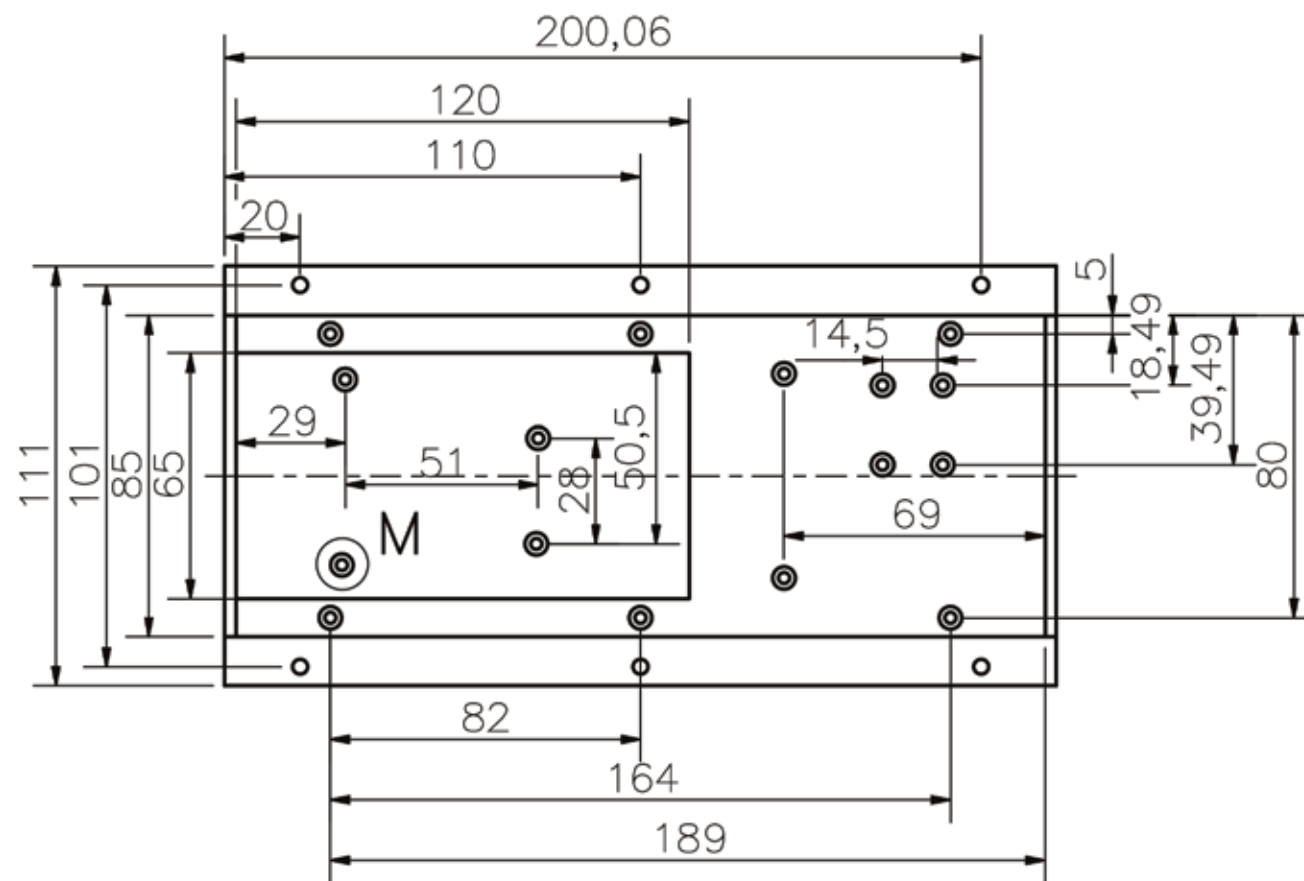
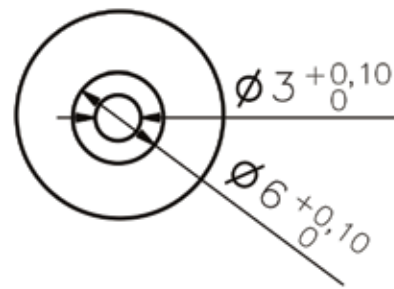


POS.	CANT.	DESCRIPCIÓN	NORMA	MATERIAL	OBSERVACIONES
5	16	TUERCA HEXAGONAL M3	DIN 934	STEEL	
4	16	TORNILLO M3x10	DIN 7985	STEEL	
3	1	NRF24L01			
2	1	TAPA CONTROL DEL ROBOT		ABS	LÁMINA A3-05
1	1	BASE CONTROL DEL ROBOT		ABS	LÁMINA A3-06

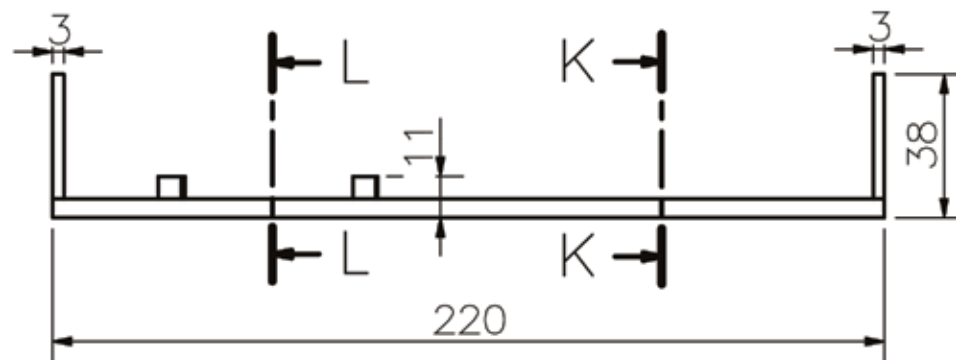
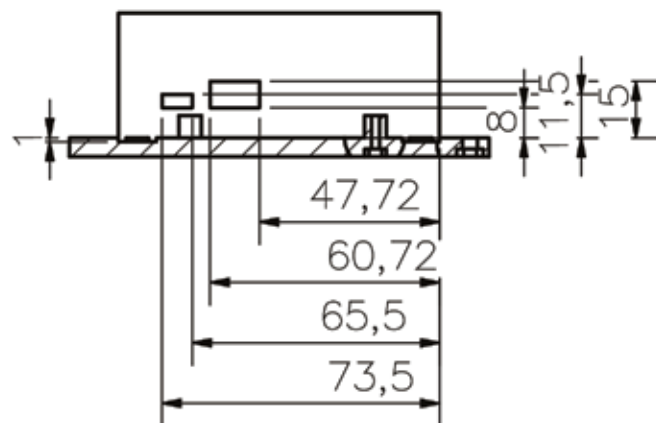
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
 FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN	TRABAJO DE FIN DE CARRERA	ESCALA
	<b>ESTUCHE CONTROL DEL ROBOT</b>	<b>1:2</b>
20097025	ESCALANTE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-11

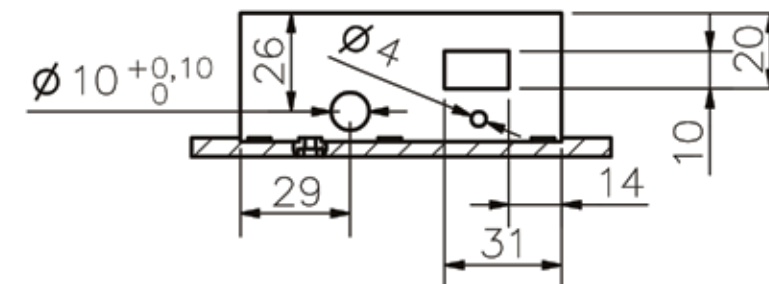
M ( 2 : 1 )



L-L ( 1 : 2 )



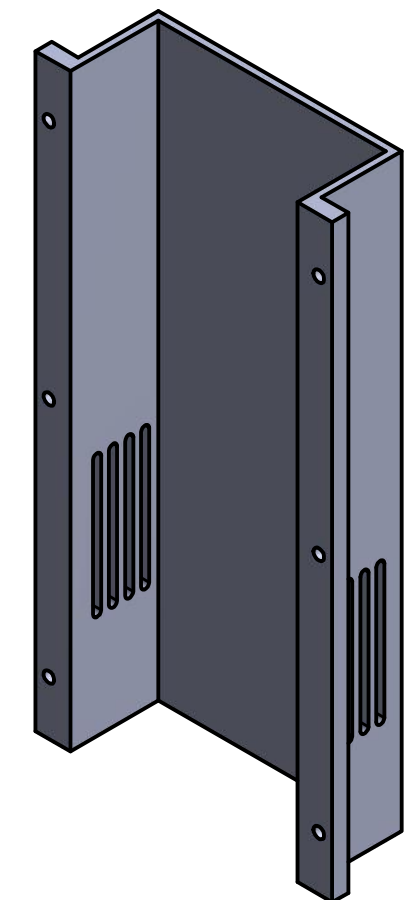
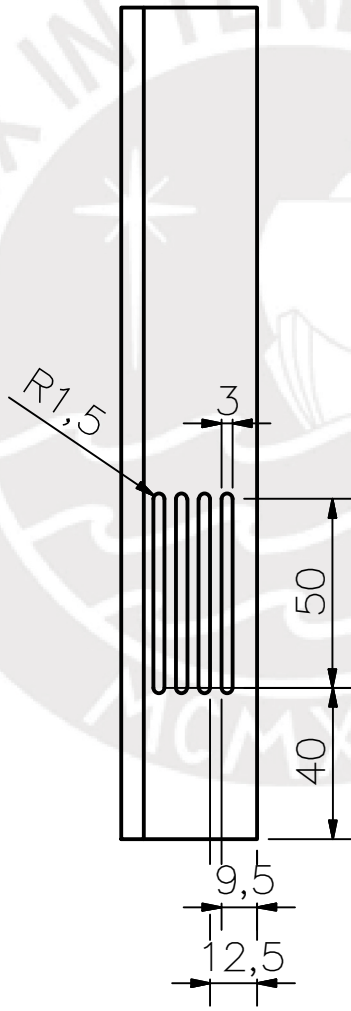
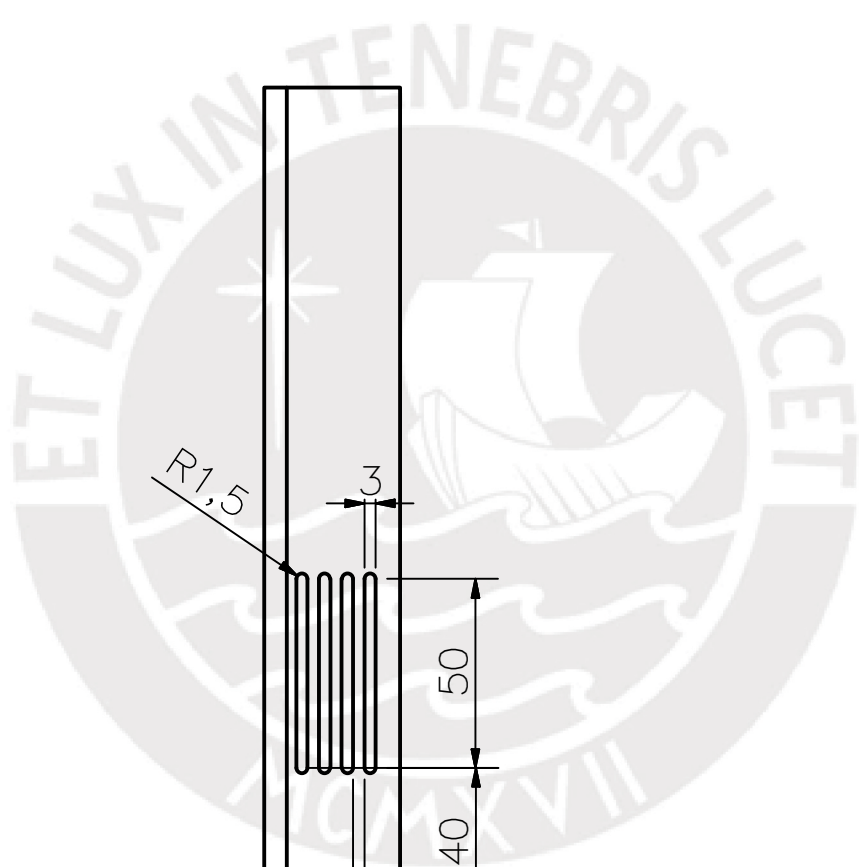
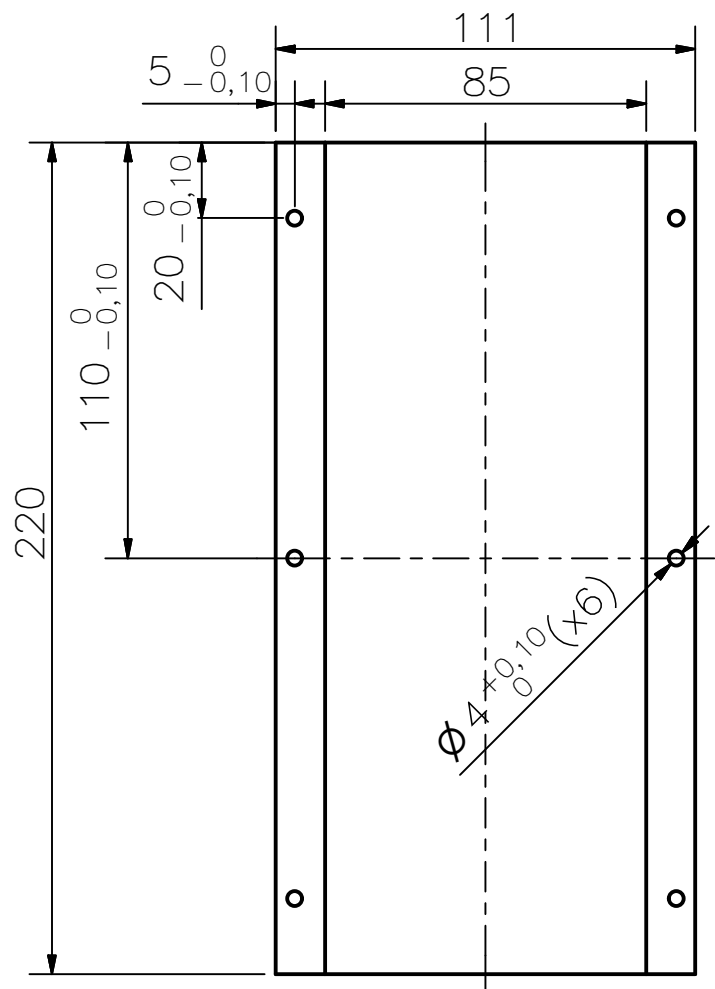
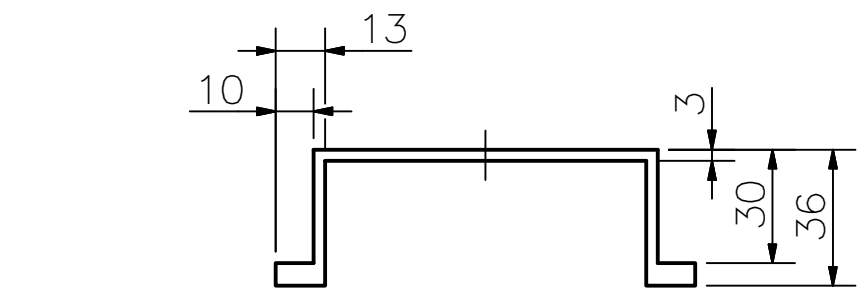
K-K ( 1 : 2 )



TOLERANCIAS DIMENSIONALES  
SEGUN DIN 7168

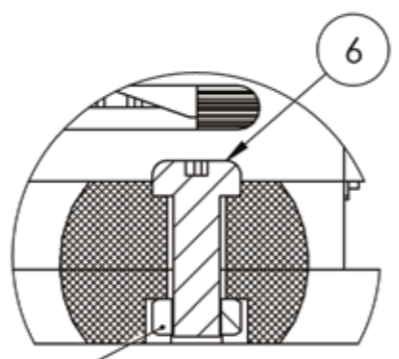
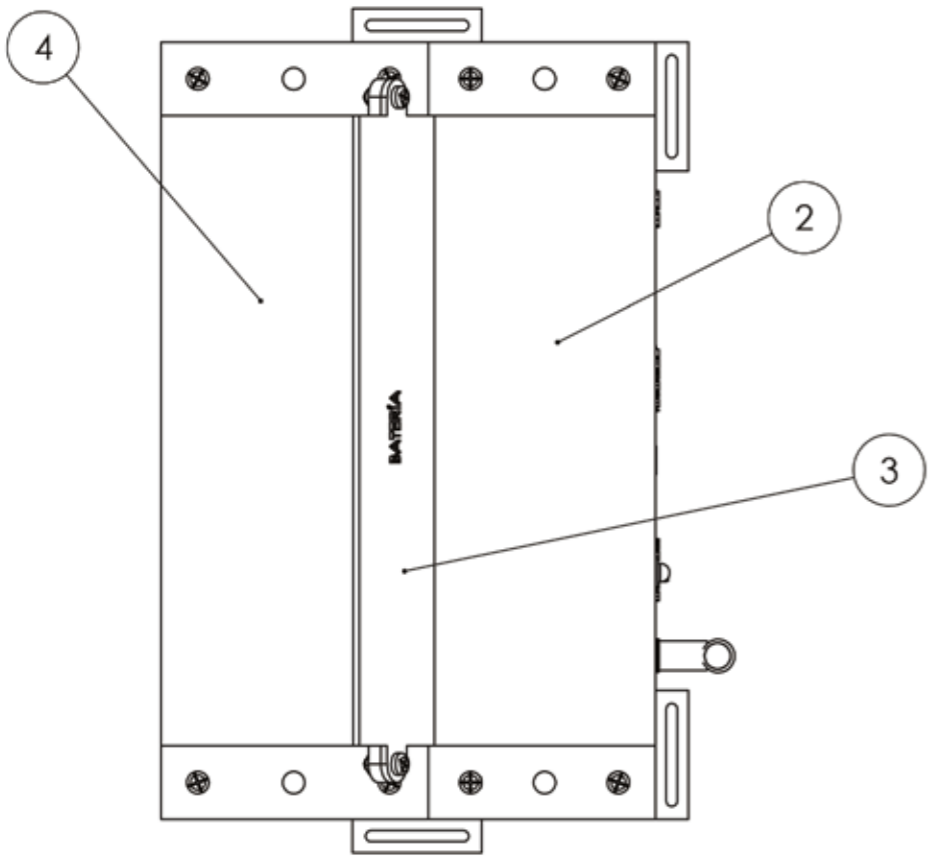
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
FINO	±0,05	±0,05	±0,1	±0,15	±0,2

ACABADO SUPERFICIAL ✓	TOLERANCIA GENERAL FINO SEGÚN DIN 7168	MATERIAL ABS
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA		
MÉTODO DE PROYECCIÓN ⊕	TRABAJO DE FIN DE CARRERA BASE CONTROL DEL ROBOT	ESCALA 1:2
20097025	ESCALANTE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-06

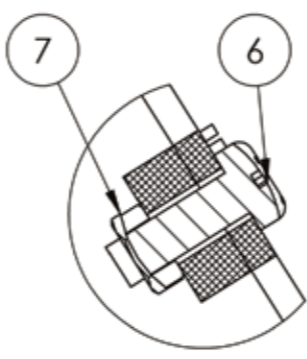


TOLERANCIAS DIMENSIONALES SEGUN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
FINO	±0,05	±0,05	±0,1	±0,15	±0,2

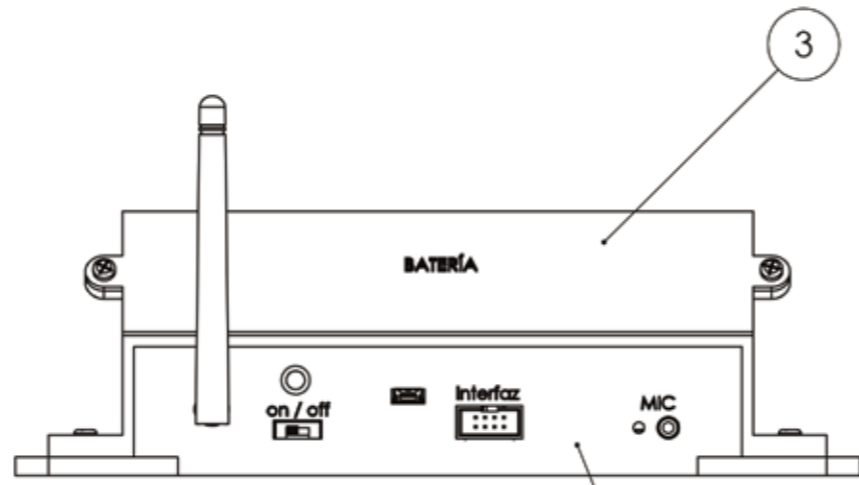
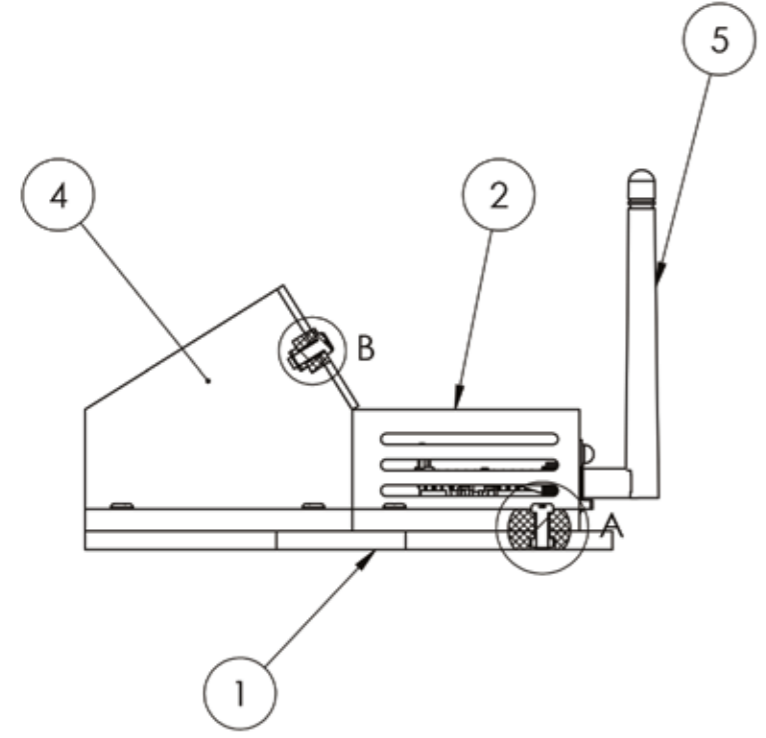
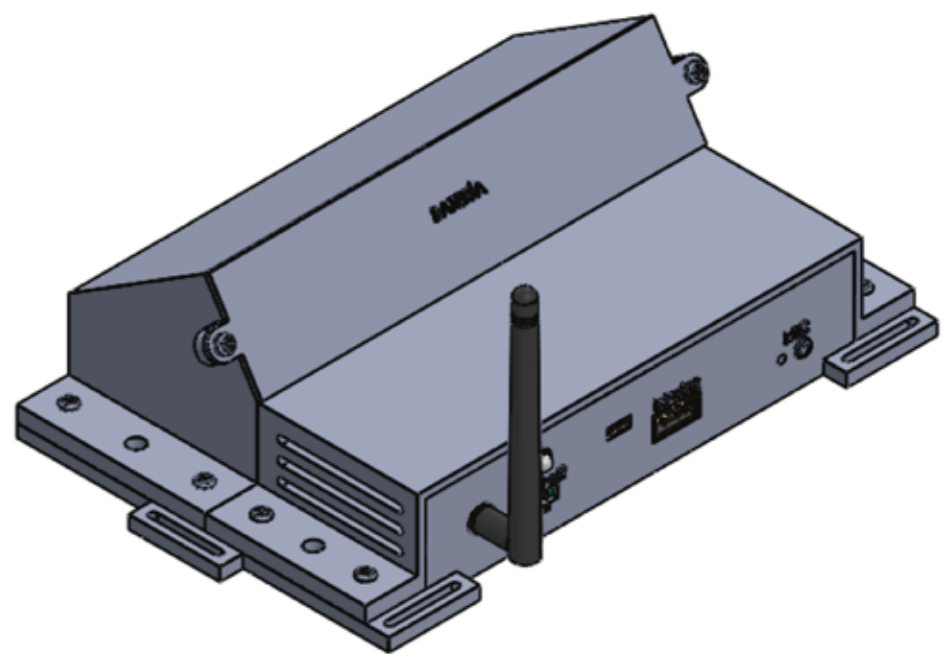
ACABADO SUPERFICIAL ✓	TOLERANCIA GENERAL FINO SEGÚN DIN 7168	MATERIAL ABS
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA		
MÉTODO DE PROYECCIÓN ⊕ ⊖	TRABAJO DE FIN DE CARRERA TAPA CONTROL DEL ROBOT	ESCALA 1:2
20097025	ESCALANTE ZELADA, JEOL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-05



DETAIL A  
SCALE 2:1



DETAIL B  
SCALE 2:1

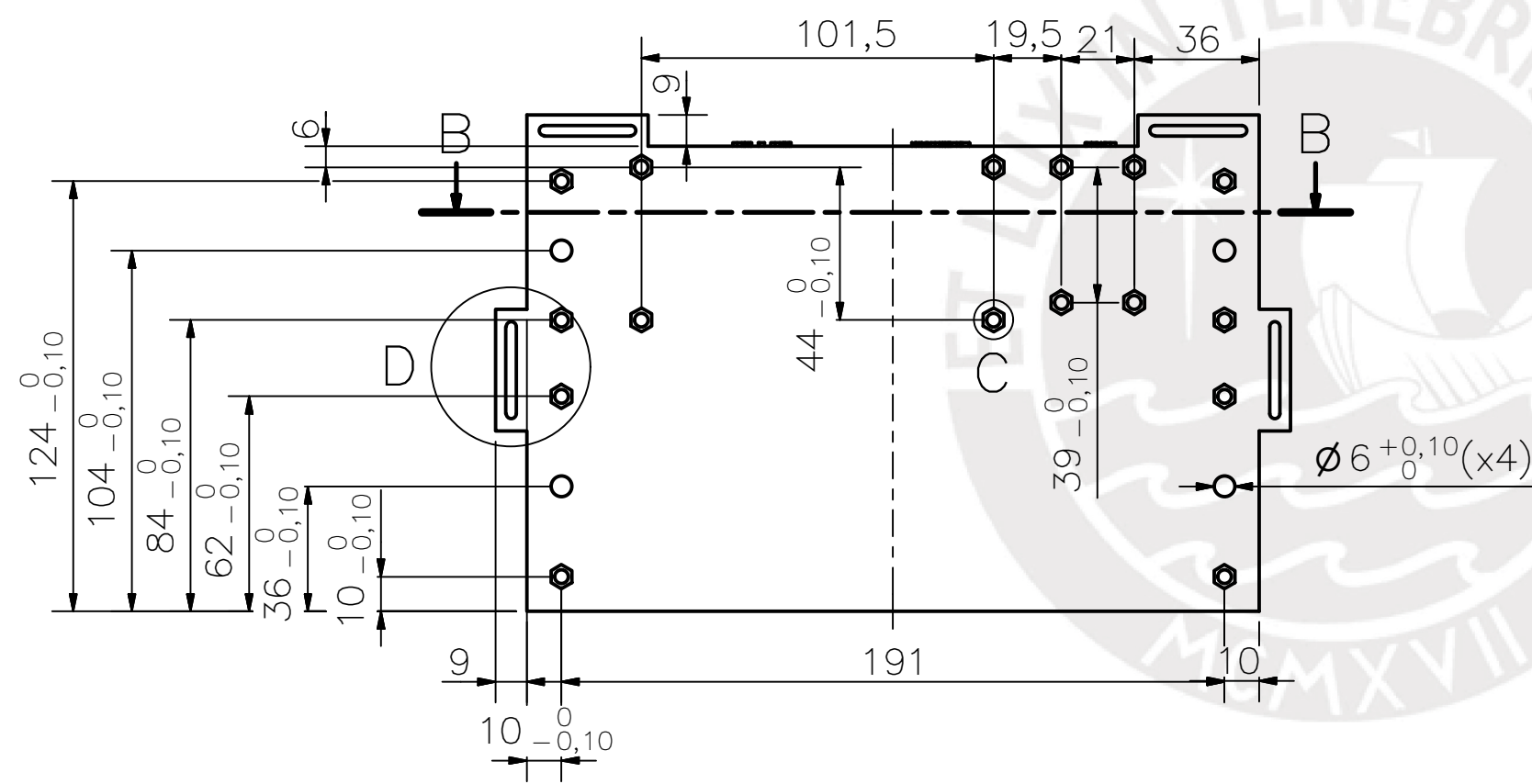
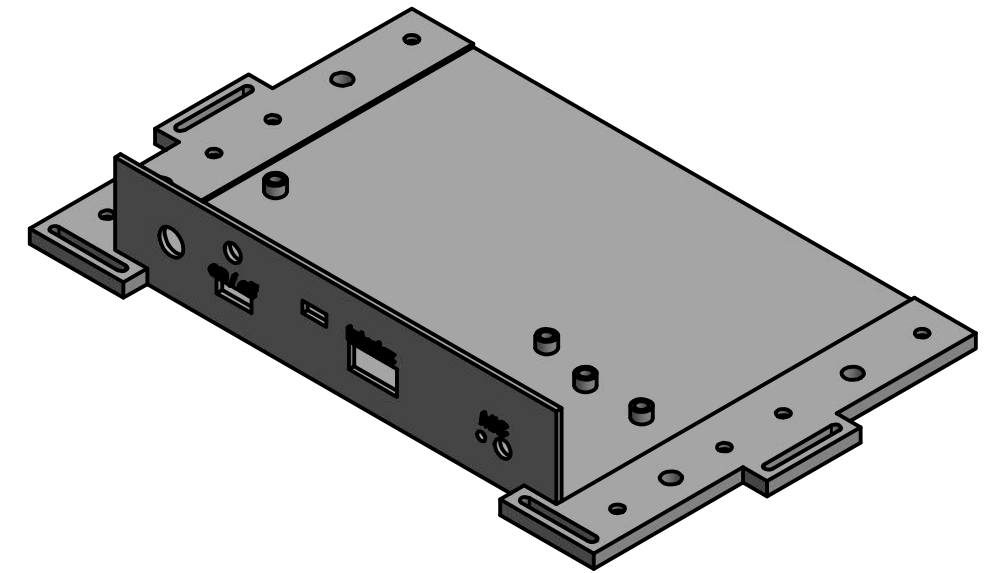
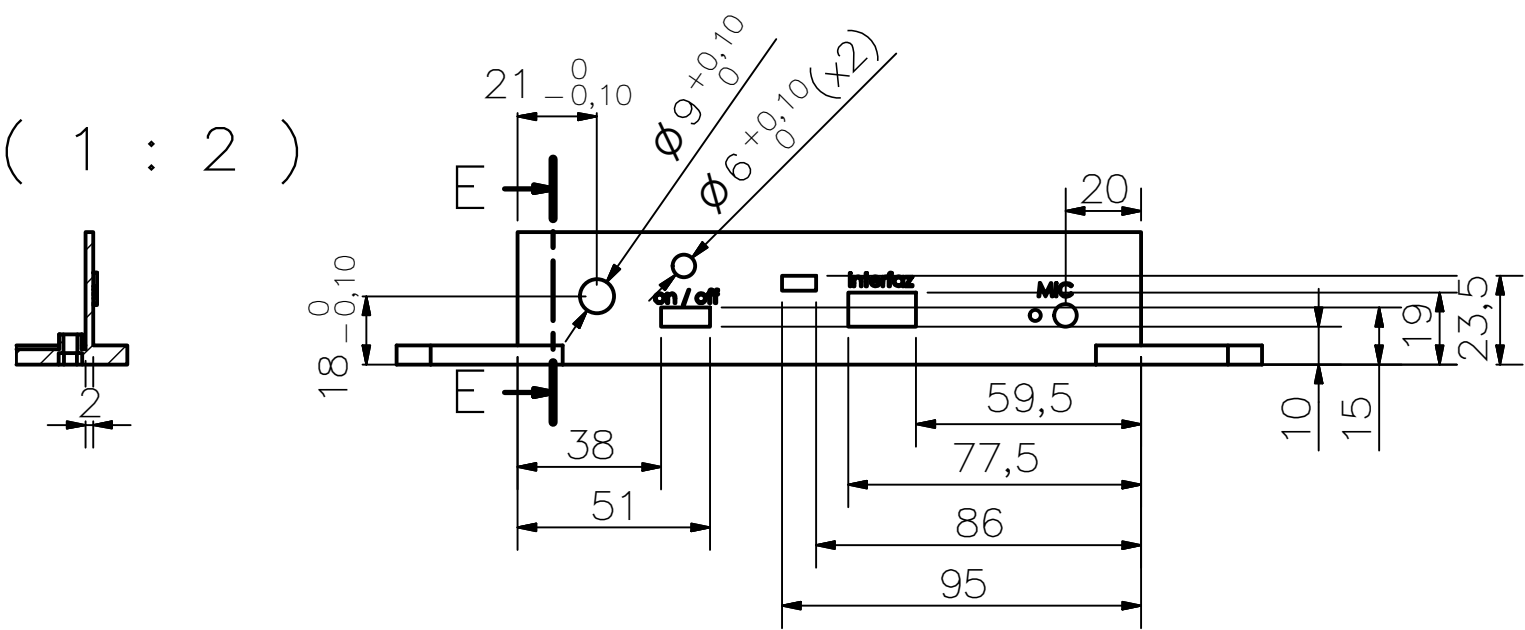


POS.	CANT.	DESCRIPCIÓN	NORMA	MATERIAL	OBSERVACIONES
7	18	TUERCA HEXAGONAL M3	DIN 934	STEEL	
6	18	TORNILLO M3x10	DIN 7985	STEEL	
5	1	NRF24L01			
4	1	SOPORTE LIPO		ABS	LÁMINA A3-04
3	1	TAPA BATERIA LIPO		ABS	LÁMINA A3-01
2	1	TAPA PARA CIRUITOS		ABS	LÁMINA A3-02
1	1	BASE REC. DE VOZ		ABS	LÁMINA A3-03

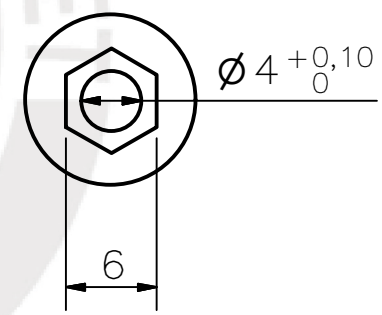
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA		
MÉTODO DE PROYECCIÓN 	TRABAJO DE FIN DE CARRERA <b>ESTUCHE DE REC. DE VOZ</b>	ESCALA <b>1:2</b>
20097025	ESCALANTE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-10



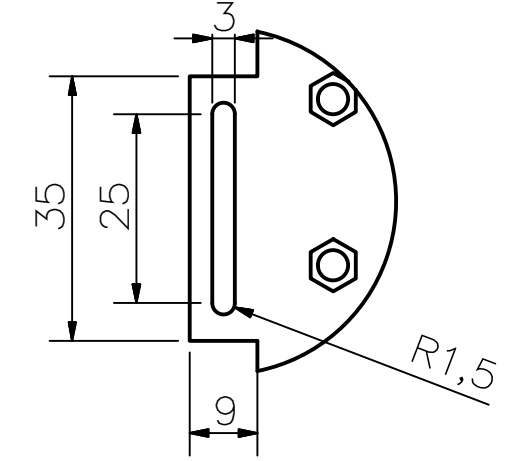
E-E ( 1 : 2 )



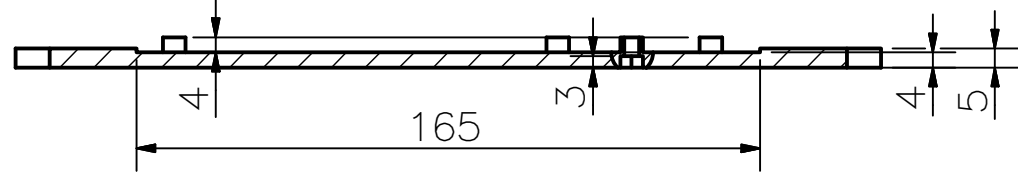
C ( 2 : 1 )



D ( 1 : 1 )



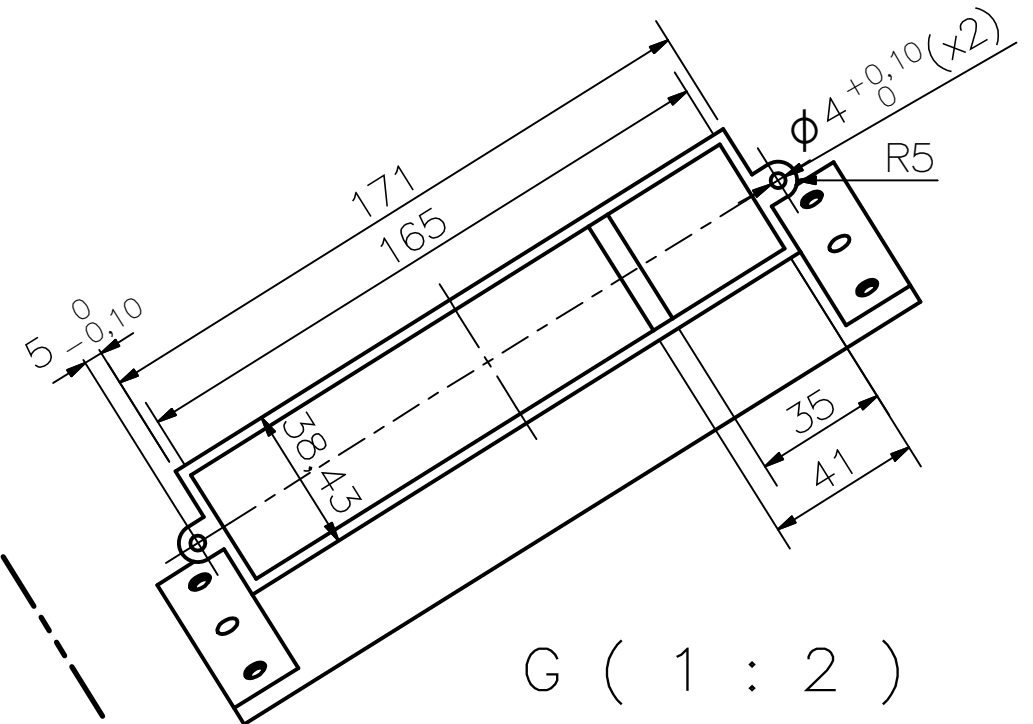
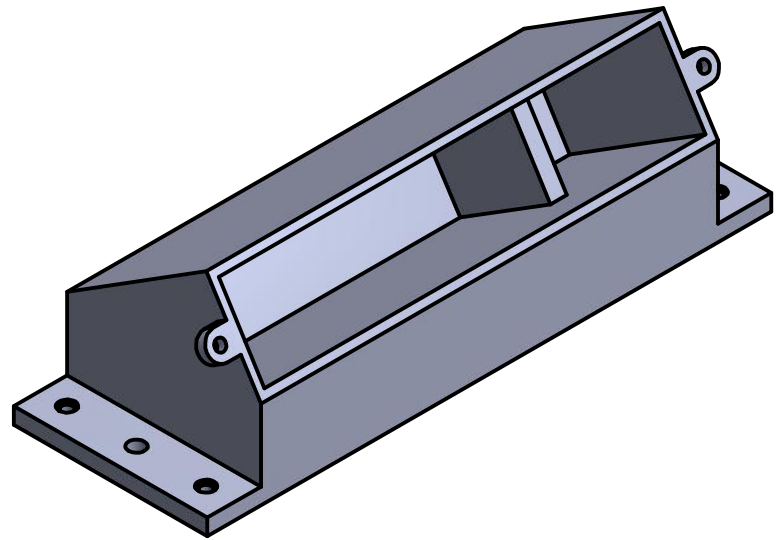
B-B ( 1 : 2 )



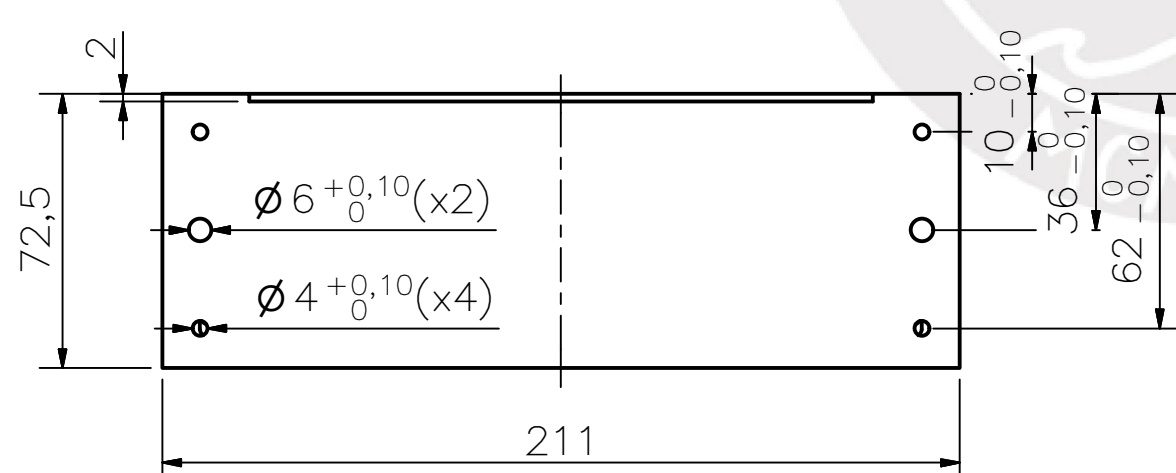
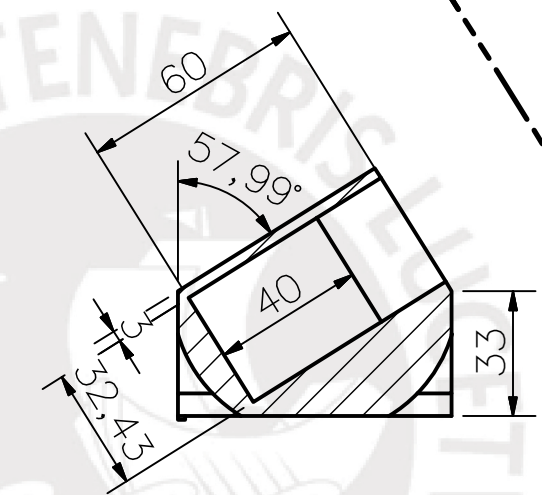
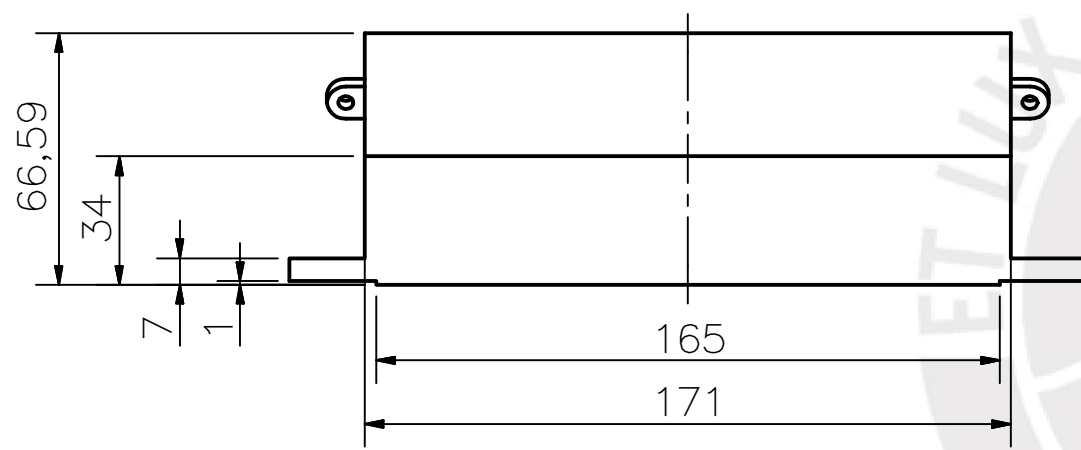
TOLERANCIAS DIMENSIONALES SEGUN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
FINO	±0,05	±0,05	±0,1	±0,15	±0,2

ACABADO SUPERFICIAL ✓	TOLERANCIA GENERAL FINO SEGÚN DIN 7168	MATERIAL ABS
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA		
MÉTODO DE PROYECCIÓN ⊕	TRABAJO DE FIN DE CARRERA	ESCALA
	BASE REC. DE VOZ	1:2
20097025	ESCALANTE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-03





G ( 1 : 2 )



ACABADO SUPERFICIAL ✓	TOLERANCIA GENERAL FINO SEGÚN DIN 168	MATERIAL ABS
--------------------------	---	-----------------

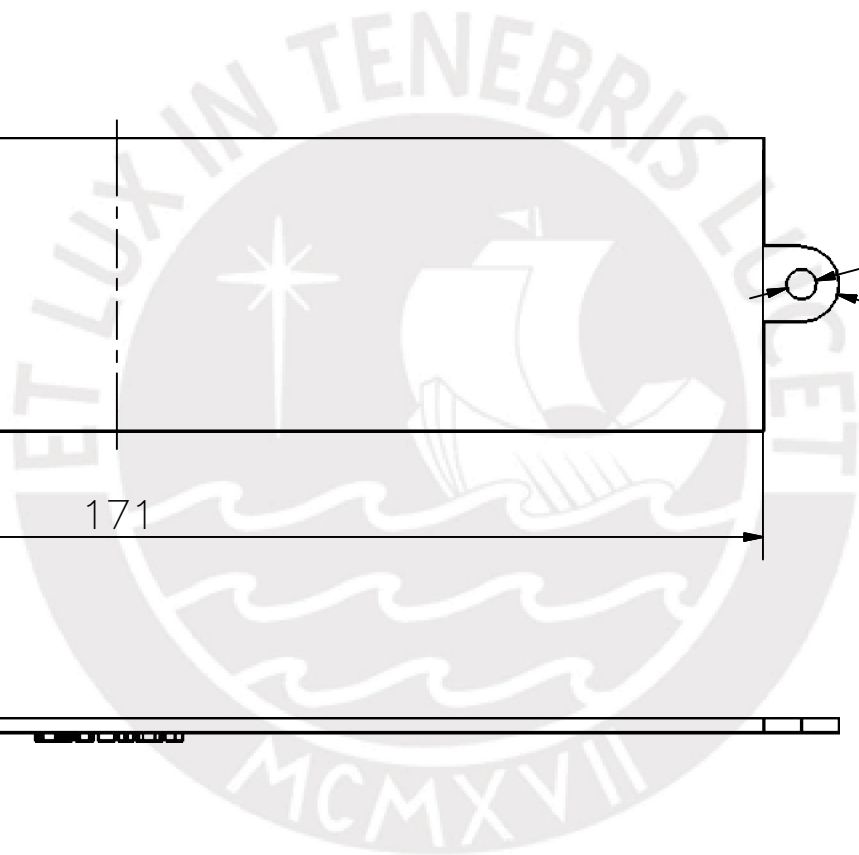
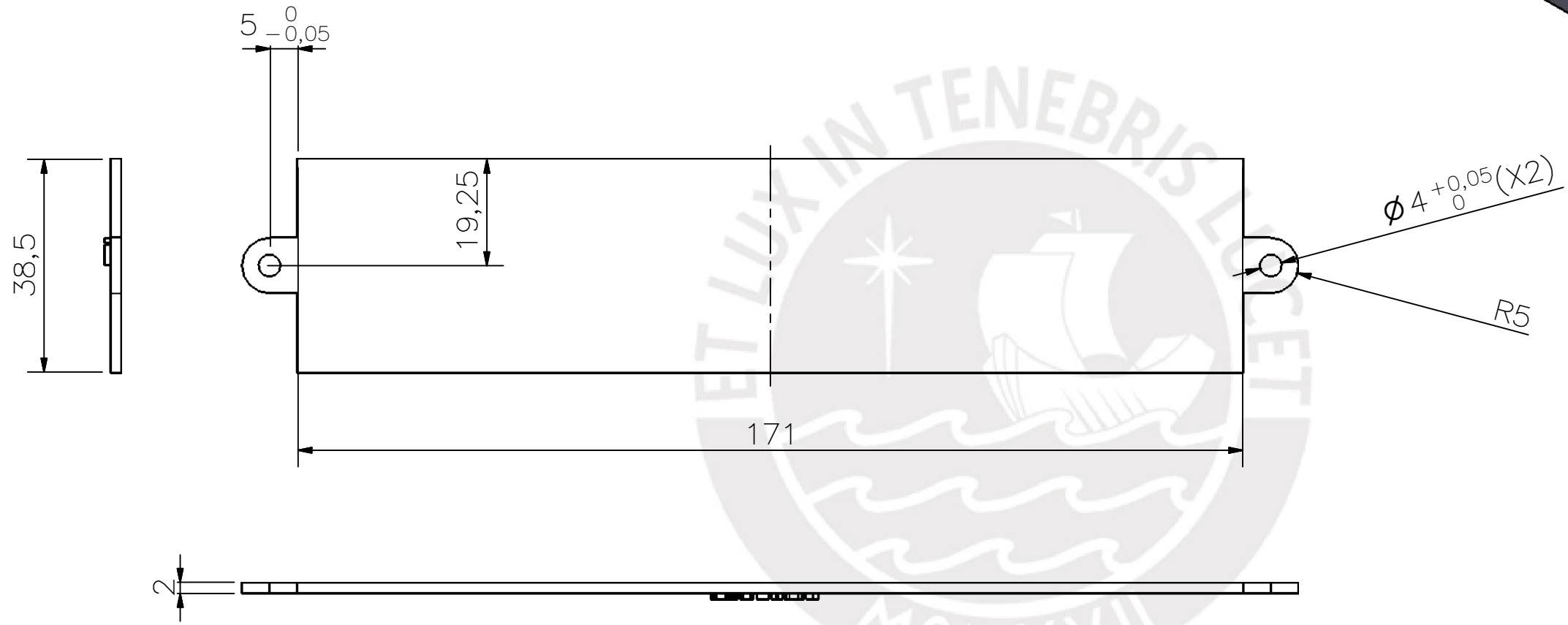
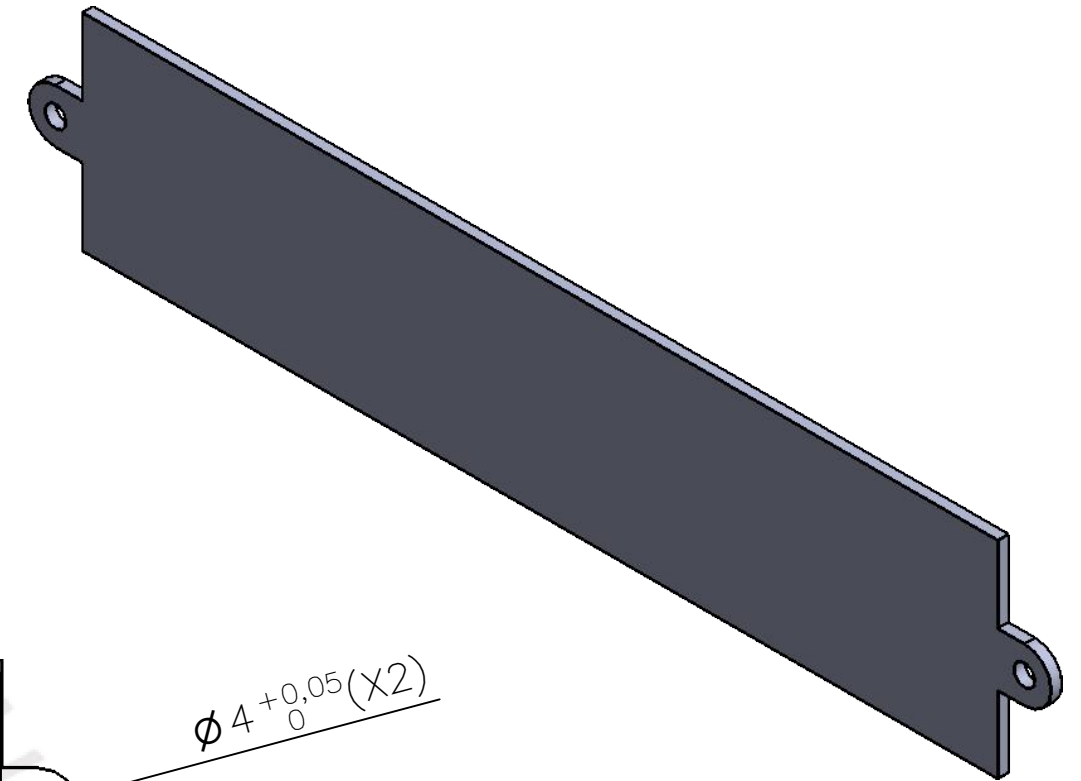
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA

MÉTODO DE PROYECCIÓN ⊕ ⊖	TRABAJO DE FIN DE CARRERA SOPORTE LIPO	ESCALA 1:2
-----------------------------	---	---------------

20097025	ESCALANTE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
----------	--------------------------------	----------------------

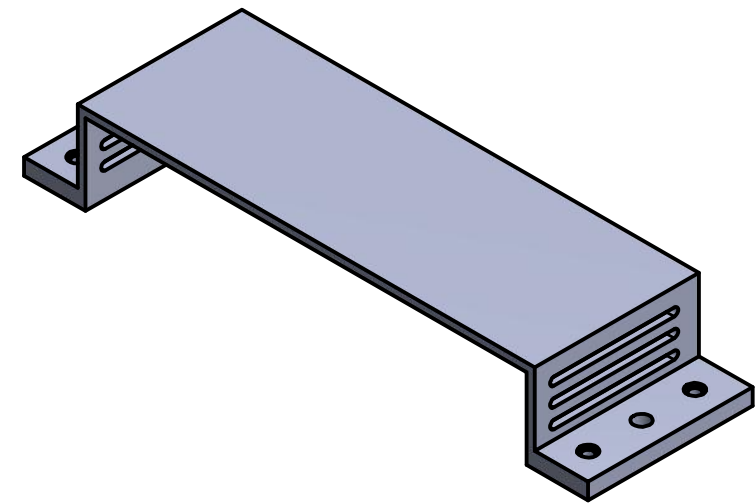
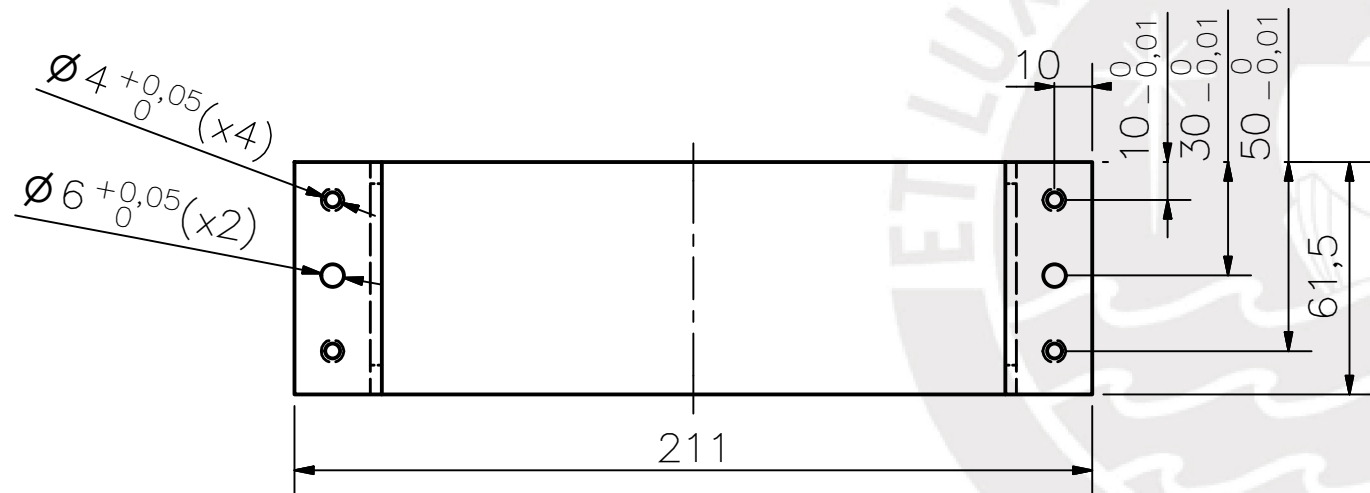
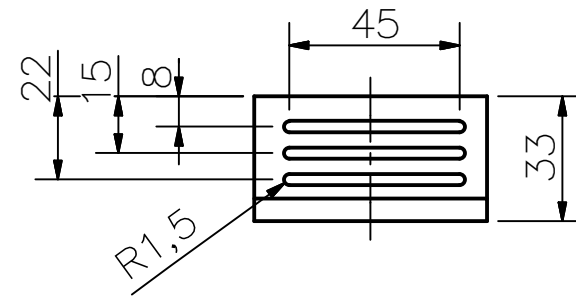
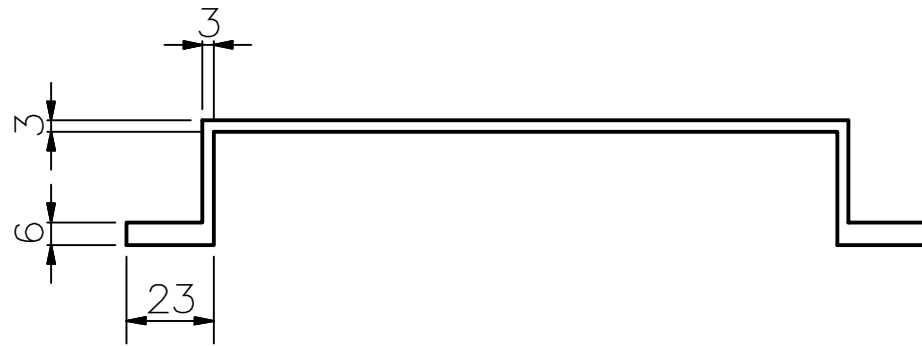
		LÁMINA: A3-04
--	--	------------------

TOLERANCIAS DIMENSIONALES SEGUN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
FINO	±0,05	±0,05	±0,1	±0,15	±0,2



TOLERANCIAS DIMENSIONALES SEGUN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
FINO	±0,05	±0,05	±0,1	±0,15	±0,2

ACABADO SUPERFICIAL ✓	TOLERANCIA GENERAL FINO SEGÚN DIN 7168	MATERIAL ABS
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA		
MÉTODO DE PROYECCIÓN ⊕    ▭	TRABAJO DE FIN DE CARREARA TAPA BATERIA LIPO	ESCALA 1:1
20097025	ESCALANTE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-01



TOLERANCIAS DIMENSIONALES SEGUN DIN 7168					
GRADO DE EXACTITUD	Más de 0,5 hasta 3	Más de 3 hasta 6	Más de 6 hasta 30	Más de 30 hasta 120	Más de 120 hasta 400
FINO	±0,05	±0,05	±0,1	±0,15	±0,2

ACABADO SUPERFICIAL ✓	TOLERANCIA GENERAL FINO SEGÚN DIN 7168	MATERIAL ABS
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ FACULTAD DE CIENCIAS E INGENIERÍA – ESPECIALIDAD INGENIERÍA MECATRÓNICA		
MÉTODO DE PROYECCIÓN ⊕ ⊖	TRABAJO DE FIN DE CARRERA TAPA PARA CIRCUITOS	ESCALA 1:2
20097025	ESCALANTE ZELADA, JOEL EDUARDO	FECHA: 2019.09.22
		LÁMINA: A3-02