

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA**

SISTEMA DE APOYO PARA LA RESPIRACION NO INVASIVO

Trabajo de Titulación para optar al Título
de Ingeniero de Ejecución en CONTROL E
INSTRUMENTACIÓN INDUSTRIAL

Alumno:
Sebastián Padilla

Profesor Guía:
Guelis Montenegro Zamora

RESUMEN

KEYWORDS: VENTILACION MECANICA, MICROCONTROLADORES, DISEÑO MECÁNICO, INVENTOR AUTODESK, HARDWARE LIBRE, PROGRAMACION.

El trabajo de título aborda la solución de la problemática de la falta de equipamiento médico necesario para sobre llevar las repercusiones provocadas por la enfermedad llamada Covid19 o cualquier otra de índole respiratorio la cual necesite el apoyo respiratorio no invasivo en su grado inicial.

También aborda la fabricación, elaboración, programación de todo el hardware y software necesario, en un concepto de código abierto y libre para cualquiera.

Todo esto para poder trasmitir el conocimiento para construir la maquinaria necesaria en caso de tener nuevamente la carencia de equipamiento médico. También puede ser un elemento de ayuda donde el equipamiento médico sea un recurso costos o bien destinado a familias donde el apoyo respiratorio sea para tratamiento de alguna enfermedad o padecimiento.

ÍNDICE

INTRODUCCIÓN	1
CAPITULO 1: PROBLEMÁTICA	2
1. PROBLEMÁTICA	3
1.1 <u>PROPIUESTA</u>	3
1.2 <u>ALTERNATIVAS DE SOLUCIÓN</u>	4
1.2.1. <u>Alternativa 1: "Actuador de Volumen Variable para Ventilación Mecánica</u>	4
1.2.2. <u>Alternativa 2: "Aire comprimido"</u>	6
1.2.3. <u>Alternativa 3: "Bolsa Manual (AMBU)"</u>	7
1.3 <u>SOLUCION SELECCIONADA</u>	8
1.4 <u>ALTERNATIVAS PARA EL ACTUADOR</u>	11
1.4.1. <u>Motor paso a paso</u>	12
1.4.2. <u>Servo motor</u>	12
1.4.3. <u>Motor DC</u>	13
1.5 <u>SOLUCIÓN PARA ESCOGER EL ACTUADOR</u>	13
CAPITULO 2: DISEÑO MECÁNICO PARA LA MANIPULACIÓN DEL AMBU	14
1. DISEÑO MECÁNICO PARA LA MANIPULACIÓN DEL AMBU	15
1.1 <u>DISEÑO MECANICO Y ELECCIÓN DE ELEMENTOS</u>	15
1.1.1. <u>Conjunto de rodamiento lineales</u>	16
1.1.2. <u>Motor paso a paso</u>	17
1.1.3. <u>Driver DM556</u>	19
1.1.4. <u>Polea GT2 y correa dentada</u>	20
1.1.5. <u>Sujetador de correa dentada</u>	22
1.1.6. <u>Celda de carga</u>	22
1.1.7. <u>Modulo HX711</u>	23
1.1.8. <u>Pantalla LCD 20x4</u>	25
1.1.9. <u>Botoneras tipo membrana</u>	27
1.1.10. <u>Botoneras PARTIR/PARAR</u>	28

1.1.11.	<u>Luces piloto o indicadores de funcionamiento</u>	28
1.1.12.	<u>Fuente de poder</u>	29
1.2	DISEÑO DE COMPONENTES Y SISTEMAS	30
2.2.1.	<u>Sistema de riel con carro</u>	31
2.2.2.	<u>Soporte principal</u>	32
2.2.3.	<u>Soporte de motor</u>	34
2.2.4.	<u>Conjunto tensor</u>	36
2.2.5.	<u>Sistema de medición de fuerza</u>	39
2.2.6.	<u>Soporte para acrílico</u>	42
2.2.7.	<u>El microcontrolador</u>	45
2.2.8.	<u>Circuito de alimentación</u>	48
2.2.9.	<u>Circuito de Botón Start, Stop y sensor magnético</u>	49
2.2.10.	<u>Circuito de botones de membrana</u>	51
2.2.11.	<u>Circuito de driver para motor paso a paso</u>	54
2.2.12.	<u>Circuito de celdas de carga</u>	56
2.2.13.	<u>Circuito de LCD</u>	58
2.2.14.	<u>Circuito de luces de proceso</u>	59
2.2.15.	<u>Disposición de pines y función</u>	60
CAPITULO 3: DESARROLLO	62
2.	DESARROLLO	63
2.1	IMPRESIÓN 3D	63
1.1.1.	<u>Laminador o Slicer</u>	64
2.2	CONSTRUCCIÓN	67
1.2.1.	<u>Elección de materiales</u>	68
3.2.2.	<u>Elección de componentes electrónicos</u>	70
2.3	CONSTRUCCIÓN ELECTRÓNICA.	73
3.3.1.	<u>Confección de esquemático y distribución PCB</u>	74
3.3.2.	<u>Fabricación de PCB</u>	76
3.3.3.	<u>Ensamble de componentes</u>	79
3.4	PROGRAMACIÓN.	81

3.4.1. <u>Instalación de Visual Studio Code</u>	82
3.4.2. <u>Configuración de entorno</u>	82
3.4.3. <u>Cargar proyecto al entorno</u>	82
3.4.4. <u>Preparación y verificación de proyecto</u>	84
3.4.5. <u>Grabado de proyecto al microcontrolador</u>	85
3.5 DESCRIPCIÓN DE INTERFAZ Y FUNCIONALIDADES DEL SISTEMA	87
3.5.1. <u>Pantalla</u>	87
3.5.2. <u>Botones de Configuración</u>	88
3.5.3. <u>Botones de operación</u>	88
3.5.4. <u>Luces Indicadoras</u>	89
3.5.5. <u>Interacción General</u>	89
3.5.6. <u>Configuración</u>	90
3.5.7. <u>Modificar parámetros</u>	90
3.5.8. <u>Descripción de parámetros</u>	92
3.5.9. <u>Funciones Importantes</u>	93
3.6 VALORES Y COSTOS	94
3.6.1. <u>Costos de materiales</u>	94
3.6.2. <u>Costos de componentes electrónicos</u>	94
3.6.3. <u>Carta Gantt de trabajos</u>	95
3.6.4. <u>Valores de HH Especialistas</u>	95
3.6.5. <u>Costo total</u>	96
CONCLUSIÓN	97
AGRADECIMIENTOS	98
APÉNDICE 1: MENÚ DE CONFIGURACIÓN	100
APÉNDICE 2: CARRO DE COMPRA	107
APÉNDICE 3: CODIGO	117

ÍNDICE DE FIGURAS

Figura 1-1. Gráfico en porcentaje de evaluación de alternativas	10
Figura 1-2. Fotografía de AMBU	10
Figura 2-1. Diseño de funcionamiento de AMBU.....	16
Figura 2-2. Conjunto de rodamientos lineales	17
Figura 2-3. Motor pasó a paso Bipolar	18
Figura 2-4. Driver DM556	20
Figura 2-5. Correa dentada.....	21
Figura 2-6. Polea dentada.....	21
Figura 2-7. Diseño de sujetador de correa	22
Figura 2-8. Celda de carga	23
Figura 2-9. Modulo HX711.....	23
Figura 2-10. Pantalla LCD 20x4	25
Figura 2-11. Pines de la pantalla LCD 20x4.....	26
Figura 2-12. Botonera tipo membrana.....	27
Figura 2-13. Botón de parada	28
Figura 2-14. Luz indicador de partida	28
Figura 2-15. Fuente de poder	29
Figura 2-16. Software Autodesk Inventor	30
Figura 2-17. Diseño de sistema de carro	31
Figura 2-18. Diseño de soporte principal del carro	32
Figura 2-19. Diseño de soporte principal del carro (3D)	32
Figura 2-20. Diseño de soporte principal del carro (3D vista dinámica del proceso)	33
Figura 2-21. Diseño de soporte del motor paso a paso	34
Figura 2-22. Diseño de soporte del motor paso a paso (simulación dinámica)	35
Figura 2-23. Diseño de sistema tensor de correa.....	36
Figura 2-24. Diseño de sistema tensor de correa.....	37
Figura 2-25. Diseño de sistema tensor de correa (Soporte)	37

Figura 2-26. Diseño de sistema tensor de correa (Deslizante).....	38
Figura 2-27. Diseño de sistema de sensor de fuerza.....	39
Figura 2-28. Diseño de sistema de sensor de fuerza (celda de carga)	40
Figura 2-29. Diseño de sistema de sensor de fuerza (Simulación dinámica)	41
Figura 2-30. Diseño de soporte para acrílico.....	42
Figura 2-31. Diseño de medidas de acrílico.....	43
Figura 2-32. Acrílico incorporado	44
Figura 2-33. Microcontrolador	45
Figura 2-34. PinIN/OUT del Microcontrolador	46
Figura 2-35. Ensamble de elementos	47
Figura 2-36. Diagrama de alimentación	48
Figura 2-37. Diagrama de alimentación lado PCB	48
Figura 2-38. Diagrama de botones y sensor	49
Figura 2-39. Diagrama de conexión y funcionamiento opto acoplado	50
Figura 2-40. Diagrama de conexionado opto acoplado a microcontrolador	51
Figura 2-41. Diagrama de divisor resistivo para entrada ADC	52
Figura 2-42. Simulación de esquema de conexión para uso del ADC	53
Figura 2-43. Esquema de conexión de drive para motor paso a paso	54
Figura 2-44. Salida opto acoplada hacia entrada del driver.....	55
Figura 2-45. Conexionado de celdas de carga.....	56
Figura 2-46.Fotografía de modulo HX-711	57
Figura 2-47. Esquema de conexión para uso de PCF8574 y pantalla LCD.....	58
Figura 2-48. Esquema de conexión para luces de estado y buzzer	59
Figura 2-49. Disposición de pines y función, en microcontrolador.....	60
Figura 2-50. Lógica de funcionamiento de maquinaria.....	61
Figura 3-1. Página principal para descargar el software.....	65
Figura 3-2 Elemento de compresión móvil en laminador, posición lista para impresión 3D.	66
Figura 3-3 Disposiciones de elementos de tensor de correa listos para impresión en 3D.....	67
Figura 3-4 captura de pantalla de circuito de electrónico PCB conexionado lado superior.....	74
Figura 3-5 captura de pantalla de circuito de electrónico PCB conexionado lado posterior.	75

Figura 3-6 Captura de pantalla de esquemático circuito.	76
Figura 3-7 Ejemplo de distribución de procesos de confección de PCB.	77
Figura 3-8 Captura de pantalla de inicio dentro de la página.	77
Figura 3-9 Fotografía N°1 de PCB ensamblada, lista para montaje y pruebas.....	79
Figura 3-10 Fotografía N°1 de PCB ensamblada, lista para montaje y pruebas.....	80
Figura 3-11 Fotografía de PCB montada y conectada.	81
Figura 3-12 Captura de pantalla de proyecto en Visual Studio Code.....	83
Figura 3-13 Captura de pantalla de proyecto, en carpeta de platformio.ini en Visual Studio Code.....	83
Figura 3-14 Captura de pantalla de proyecto, accediendo a la fuente en Visual Studio Code.	84
Figura 3-15 Captura de pantalla de proyecto, ícono de verificación en Visual Studio Code.	84
Figura 3-16 Captura de pantalla de proyecto, estatus de verificación correcta en Visual Studio Code.....	85
Figura 3-17 Captura de pantalla de proyecto, ícono de Upload hacia el micro controlador en Visual Studio Code.....	85
Figura 3-18 Captura de pantalla de aviso en la terminal acción grafica de usuario.	86
Figura 3-19 Captura de pantalla de aviso en la terminal, donde se realizo correctamente el Upload.	86
Figura 3-20 Representación grafica de pantalla principal en LCD.....	87
Figura 3-21 Imagen de muestra de teclado para LCD.	88
Figura 3-22 Imagen de muestra de botones de acción.	88
Figura 3-23 Imagen de muestra de luces de estado de funcionamiento.....	89
Figura 3-24 Representación grafica de interacción entre menú.....	90
Figura 3-25 Representación grafica de elementos en el menú de configuración.....	91
Figura 3-26 Representación grafica de desplazamientos y elementos de la pantalla principal y de configuración.	92
Figura 3-27 Captura de pantalla de tabla de carta Gantt.....	95
Figura 3-28 Captura de pantalla de tabla de gastos valores de HH y especialistas.	96
Figura 3-29 Captura de pantalla de tabla de gastos de los materiales del proyecto.....	96
Figura apéndice 2-1 captura de pantalla N°1 de carrito de compras de Aliexpress.	108
Figura apéndice 2-2 captura de pantalla N°2 de carrito de compras de Aliexpress.	109

Figura apéndice 2-3 captura de pantalla N°3 de carrito de compras de Aliexpress.....	110
Figura 3-75 captura de pantalla N°4 de carrito de compras de Aliexpress.....	111
Figura apéndice 2-4 captura de pantalla de total de carrito de compras de Aliexpress.	111
Figura apéndice 2-5 Captura de pantalla de listado de compra de componentes de LCSC como proyecto.....	112
Figura apéndice 2-6 Captura de pantalla N°1 de carro de compra a componentes de LCSC..	113
Figura apéndice 2-7 Captura de pantalla N°2 de carro de compra a componentes de LCSC..	113
Figura apéndice 2-8 Captura de pantalla N°3 de carro de compra a componentes de LCSC..	114
Figura apéndice 2-9 Captura de pantalla de total de carro de compra en componentes de LCSC.	115
Figura apéndice 2-10 Captura pantalla N°4 de configuración de proyecto dentro de la página de JLCPCB, asignación de envío.....	115
Figura apéndice 2-11 Captura pantalla de valor total para la elaboración de PCB dentro de la página de JLCPCB.	116

ÍNDICE DE TABLAS

Tabla1.1. Escala de calificación de alternativas.....	9
Tabla1.2. Evaluación De Alternativas Según Criterio	9
Tabla2.1. Disposición de pines en microcontrolador	60
Tabla 3.1. Materiales de estructura.	68
Tabla 3.2. Componentes electrónicos para PCB.....	71

SIGLAS Y SIMBOLOGÍA

A. SIGLAS:

OMS: Organización Mundial De La Salud.

AMBU: Unidad De Bolsa y Mascara Para La Vía Airea (Airway Mask Bag Unit).

NEMA:	Asociación de Fabricación Estadounidense de Equipos Eléctricos.
CNC:	Control Numérico Por Computadora.
LCD:	Pantalla De Cristal Liquida.
CAD:	Diseño Asistido Por Computadora.
IAM:	Ensamble en ambiente Inventor.
IPT:	Pieza en ambiente Inventor.
STL/OBJ:	Tipos de archivo 3D.
G-CODE:	Código de control para maquinas CNC.
EEPROM:	Memoria solo de lectura programable eléctricamente.
IDE:	Entorno de desarrollo integrado.
SMD:	Surface Mount Device, dispositivo de montaje superficial.
ADC:	Conversor análogo-digital, convierte señales analógicas a datos computables.
I2C:	Interfaz de comunicación inter-integrada, protocolo asincrónico.
VS Code:	Visual Studio Code, entorno de programación integrado.
PCB:	Placa de circuito impreso.
DT:	Tipo de comunicación serial de modulo HX711 (Data).
SCK:	Tipo de comunicación serial de modulo HX711 (Clock).
SCL:	Tipo de comunicación serial del protocolo I2C (Serial Clock).
SDA:	Tipo de comunicación serial del protocolo I2C (Serial Data).

B. SIMBOLOGIA

- N*cm: Unidad de fuerza por longitud
- A: Unidad de corriente eléctrica, Amper.
- V: Unidad de voltaje o tensión eléctrica, Volt.
- MHz: Unidad de ciclos por segundos.
- mm: Unidad de longitud (10^{-3}).
- mV: Unidad de tensión (10^{-3}).
- kg: Kilogramo, unidad de masa.
- ml: Mililitro, unidad de volumen.
- cmH₂O: Centímetro de agua, unidad de presión.
- mPa : milipascal, unidad de presión.

OBJETIVOS GENERALES

Crear un sistema de código y hardware abierto a la comunidad que permita dar apoyo a la respiración de forma no invasiva.

OBJETIVOS ESPECIFICOS

- Contextualizar la problemática a solucionar.
- Estudiar asesorías con médicos internalistas.
- Seleccionar los dispositivos que formaran parte del proyecto.
- Realizar la programación del microcontrolador.
- Obtener resultados prácticos de implementación del proyecto.

INTRODUCCIÓN

A finales de 2019, el mundo enfrentó un desafío sin precedentes: la aparición del COVID-19. Originado en Wuhan, China, y declarada pandemia por la OMS el 11 de marzo de 2020, este nuevo coronavirus se propagó rápidamente a nivel mundial. Con una transmisión principalmente a través de las vías respiratorias, afectó a todas las edades, saturando los sistemas de salud y desbordando las capacidades hospitalarias. Los síntomas, que variaban desde fiebre y tos hasta severas dificultades respiratorias, llevaron a medidas de aislamiento y distanciamiento social en un intento por contener su rápida expansión.

Este texto busca explorar las repercusiones de la pandemia de COVID-19 en la salud, la sociedad y la economía global, resaltando la fragilidad humana ante tales crisis sanitarias y soluciones prácticas de bajo recurso económico.

De acuerdo con esto se plantea la hipótesis de conformar los estudios necesarios para la realización de una maquinaria la cual pueda contribuir a la falta de equipamiento médico para sobre llevar esta y cualquier crisis sanitaria de índole respiratoria a futuro. También para brindar una solución a lugares donde el equipamiento hospitalario sea difícil de adquirir o bien la utilización en hogares para de forma asistida para mejorar las condiciones o sobrellevar enfermedades de índole respiratoria.

CAPITULO 1: PROBLEMÁTICA

1. PROBLEMÁTICA

La carencia de equipamiento médico durante la crisis llevó a muchos inventores, a menudo llamados "MAKERS", a desarrollar respiradores artificiales de forma casera. A pesar de la falta de sofisticación técnica, su compromiso y determinación desempeñaron un papel crucial en la respuesta a la escasez de equipos médicos durante tiempos críticos.

Si bien estos respiradores caseros no sustituyeron a los dispositivos médicos de alta calidad, ejemplificaron la creatividad y la solidaridad de la comunidad de "MAKERS" y recordaron la importancia de la innovación en situaciones de emergencia

La problemática central se identifica en cómo la carencia de elementos técnicos como también la estandarización de componentes, uso de diagramas para elaboración de maquinaria necesaria para suplir la carencia de indumentaria médica.

1.1 PROPUESTA

Para suplir la problemática central, se realizará la creación de una maquinaria para el apoyo a la respiración de forma no invasiva, esta maquinaria cubre el vacío en la etapa donde el paciente adquiere el virus y en donde hay impedimento para realizar el ejercicio de respirar, por lo tanto, esta maquinaria será de ayuda para sobre llevar las consecuencias de la enfermedad y no avanzar o reducir los pacientes en etapa de entubación, con ello poder reducir la contingencia de la utilización de respiradores artificiales como también la mortalidad de dicha enfermedad.

Por otro lado, se dispondrá del material e información necesaria para la elaboración, fabricación y programación, dispuesta de forma libre y abierto para todo el mundo, a través de una base de datos.

La utilización de elementos de fácil acceso como también las tecnologías de impresión en 3D

Todo esto para estandarizar el proceso, como materiales y elaborar plan de acción en caso de caso de presentar nuevamente la necesidad.

Para ellos se conforma el siguiente informe técnico donde se abordará los siguientes puntos:

- ✓ Función y utilización.
- ✓ Componentes
- ✓ Fabricación
- ✓ Desarrollo
- ✓ Pruebas de funcionamiento
- ✓ Expansión
- ✓ Base de datos.

1.2 ALTERNATIVAS DE SOLUCIÓN

En esta etapa se presenta la función y utilización de elementos, empezar por lo básico, se necesita suministrar aire a un paciente de forma controlada para poder ayudar en el ejercicio de respirar.

1.2.1. Alternativa 1: "Actuador de Volumen Variable para Ventilación Mecánica

Esta alternativa implica la implementación de un actuador que comparte similitudes con un cilindro en su diseño y funcionamiento. Un émbolo dentro del elemento el cual se puede ajustar la capacidad volumétrica del cilindro, controlando de una masa de aire hacia el paciente durante un proceso de ventilación mecánica.

➤ **Ventajas:**

- ✓ Rapidez de suministro de aire:

El dispositivo permite una entrega rápida y precisa de aire al paciente, lo que es esencial en situaciones críticas.

- ✓ Operación visualmente transparente:

Su funcionamiento es fácilmente observable y comprensible para los operadores médicos, lo que facilita su monitoreo y ajuste.

- ✓ Nivel de ruido moderado:

Aunque genera un nivel de ruido apreciable, se encuentra dentro de rangos aceptables en entornos médicos.

➤ Desventajas:

- ✓ Costos de fabricación elevados:

Los materiales de alta calidad necesarios para la fabricación hacen que este dispositivo sea costoso de producir.

- ✓ Complejidad de ensamblaje:

La gran cantidad de componentes y la forma de adquirirlos hacen que sea complejo ensamblaje, lo que puede requerir habilidades más artesanales.

- ✓ Tamaño aparatoso:

La maquinaria asociada a este sistema puede ocupar un espacio considerable en el entorno clínico, lo que puede resultar incómodo y requerir una distribución adecuada.

- ✓ Sellado poco confiable y alto roce de las paredes:

El sistema de sellado presenta ciertas limitaciones en cuanto a su confiabilidad, y el roce elevado de las paredes puede influir en la eficiencia y durabilidad del dispositivo.

- ✓ Desgaste acelerado y alta necesidad de mantenimiento:

Debido a la naturaleza de su funcionamiento y el roce mencionado, este dispositivo tiende a experimentar un desgaste más rápido, lo que implica una necesidad de mantenimiento frecuente.

- ✓ Dificultad en la esterilización de componentes:

Algunos elementos del sistema pueden ser complicados de esterilizar adecuadamente, lo que es crítico en entornos médicos donde la higiene es fundamental.

1.2.2. Alternativa 2: "Aire comprimido"

Esta alternativa se basa en la utilización de un compresor de aire sumado a un sistema de control, el cual regula el suministro hacia el paciente, todo esto para que el paciente reciba el aire de forma controlada.

- Ventajas:

- ✓ Rapidez de suministro de aire:

El dispositivo permite una entrega rápida y precisa de aire al paciente, lo que es esencial en situaciones críticas.

- ✓ Pocos materiales de implementación:

Esto permite obtener materiales de forma rápido, lo cual se traduce a menor tiempo de fabricación y puesta en servicio.

- Desventajas:

- ✓ Generación notable de ruido:

Uno de los principales inconvenientes radica en el ruido producido por el compresor de aire lo cual no permite un reposo del paciente.

- ✓ Generación de vibración:

La vibración es una consecuencia del uso de compresor de aire el cual puede generar problemas de contacto y piezas sueltas, lo cual, se traduce en funcionamiento erróneos

- ✓ Altos costo de implementación
- ✓ Difícil esterilización:

Esta etapa es fundamental para el tratamiento de indumentaria médica, lo cual tiene protocolos específicos de esterilización y sanitización.

1.2.3. Alternativa 3: "Bolsa Manual (AMBU)"

Esta alternativa se trata de la utilización de una bolsa de respiración (AMBU), el cual es empleado en situaciones en las que el paciente carece parcial o completamente de la capacidad de respirar por sí mismo. Este dispositivo se compone de una unidad que, de manera automática y controlada, comprime y expande una bolsa de aire, permitiendo así la administración de oxígeno y la eliminación de dióxido de carbono en el sistema respiratorio del paciente.

➤ Ventajas:

- ✓ Rápido funcionamiento de suministro de aire:

El AMBU permite una administración rápida y efectiva de oxígeno, lo que es crucial en situaciones de emergencia.

- ✓ Pocos materiales de implementación:

Se requieren pocos componentes para utilizar la bolsa de respiración, lo que facilita su disponibilidad y transporte.

- ✓ Maquinaria de poco espacio:

La AMBU es compacta y fácil de transportar, lo que la hace ideal para situaciones móviles o en entornos con espacio limitado.

- ✓ Muchas funciones operativas:

Este dispositivo no se limita solo a la administración de oxígeno; también puede utilizarse para ventilar al paciente, lo que amplía su utilidad en diferentes escenarios clínicos.

- ✓ Poco ruidosa:

A diferencia de algunas alternativas más ruidosas, la AMBU funciona de manera silenciosa, lo que puede ser beneficioso para el confort del paciente y el entorno.

➤ Desventajas:

- ✓ Fabricación compleja:

La AMBU es un dispositivo médico complejo de fabricar, lo que puede generar costos adicionales y limitar su disponibilidad en ciertas regiones o situaciones.

- ✓ Desafío de ingeniería:

Este desarrollo implica la competencia en varios campos de desarrollo de forma inicial, hasta terminar el prototipo y su visión completa.

1.3 SOLUCION SELECCIONADA

A partir de las opciones de solución presentadas, se realizará el análisis de las ventajas y desventajas de cada una de ellas, para determinar de forma óptima la más ventajosa para su implementación. Cada uno de estos criterios cualitativos será evaluado en ponderación de 1 a 7 donde, 1 se comprende a “Muy deficiente” y 7 “Excelente”, luego se realizará un promedio para determinar el elemento a utilizar.

Tabla1.1. Escala de calificación de alternativas

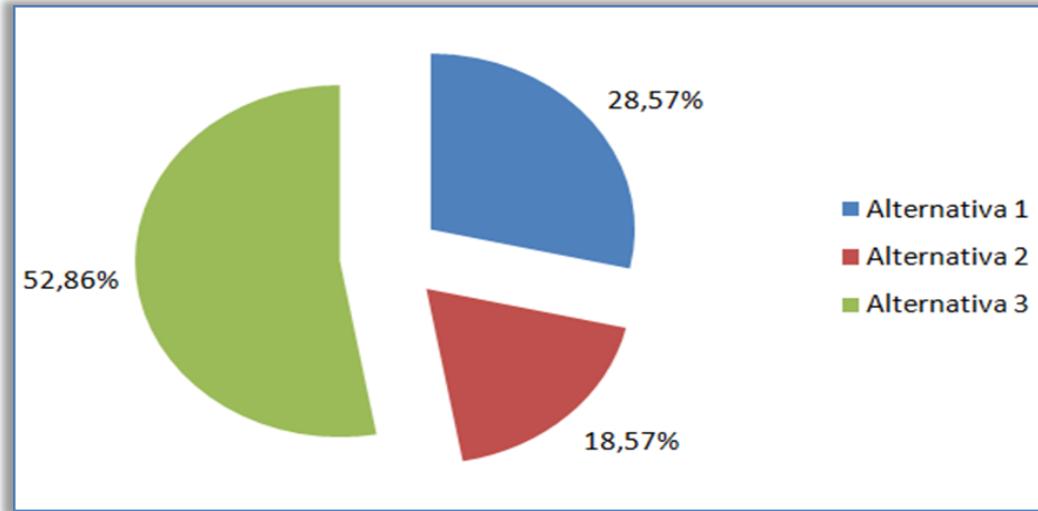
Tabla de puntajes							
Muy deficiente	Deficiente	Poco satisfactorio	Satisfactorio	Bueno	Muy bueno	Excelente	
1	2	3	4	5	6	7	

Fuente: Tabla de evaluación de alternativa

Tabla1.2. Evaluación De Alternativas Según Criterio

Características	Alternativa 1	Alternativa 2	Alternativa 3
Ruido	2	1	5
Vibración	3	1	5
Costo de fabricación	5	5	5
Visualizar el movimiento	6	1	5
Desafío de Ingeniería	2	2	3
Espacio de postura	1	2	7
Limpieza	1	1	7
Totales	20	13	37

Fuente: Tabla de evaluación de criterio



Fuente: Grafico de porcentaje de evaluación de alternativas, propio.

Figura 1-1. Gráfico en porcentaje de evaluación de alternativas

Como se puede observar en la figura 1-1, le porcentaje mayor del grafico de torta, fue para la alternativa N°3, donde tras un exhaustivo análisis y consideración de diferentes criterios se concluyó que la solución más adecuada para necesidades, es la bolsa de reanimación conocida comúnmente como AMBU, como se puede observar en la figura 1-2.



Fuente: <https://tactical-medicine.com/blogs/news/bolsa-de-reanimacion-manual-AMBU-guia-de-oxigenoterapia-y-nebulizaciones-by-dr-ramon-reyes-md>

Figura 1-2. Fotografía de AMBU

Esta herramienta no solo debe ser utilizada manualmente, sino que además buscamos automatizar su funcionamiento mediante un sistema mecanizado.

La elección de la bolsa AMBU se basa en varios factores determinantes. Uno de los más significativos es su mantenimiento, ya que se puede realizar limpiezas con equipamiento hospitalario de autoclave, el cual a través de vapor y temperatura realiza la esterilización.

La durabilidad y facilidad con la que se puede mantener en óptimas condiciones, así como su esterilización, la hacen resaltar frente a otras opciones.

Es vital garantizar la higiene y funcionalidad del dispositivo, especialmente en contextos médicos donde la salud del paciente es primordial.

El diseño del AMBU es ingeniosamente simple y altamente eficiente. Compuesto por una bolsa auto expandible de plástico, se conecta a dos válvulas unidireccionales en sus extremos, mientras una válvula facilita la entrada de aire fresco al interior del dispositivo, la otra asegura que el aire exhalado sea dirigido al exterior, evitando así la re-ingestión y garantizando que el paciente reciba aire limpio en cada ciclo.

Una ventaja adicional del AMBU es su versatilidad. Está disponible en diversos tamaños, lo que permite adaptarse a distintas morfologías faciales, garantizando un sellado adecuado y eficiente.

También posee sistema de válvula de seguridad en caso de presentar alguna obstrucción, se activa cuando superamos el umbral de presión, en este caso la mayoría de AMBU posee una válvula de 40 cmH₂O, superado esta presión la válvula por medio de un resorte, alivia la presión hacia el exterior, también de forma manual se puede girar para utilizarla o no, ya que su funcionamiento mecánico queda bloqueado para que no actué.

1.4 ALTERNATIVAS PARA EL ACTUADOR

Ya que se escogió el elemento en cuestión el cual será el responsable de suministrar el aire, debemos escoger los mecanismos que realizan la acción de contraer o apretar dicha bolsa.

En la evaluación anterior se pudo determinar que el aire suministrado por elemento compresor o aire comprimido, no era factible por la cantidad de rudo involucrado, no se podrá evaluar un sistema neumático o hidráulico para este fin por lo tanto, para ello hay tres candidatos para dicho movimiento, estos son:

- Motor paso a paso.
- Servo motor.
- Motor DC.

1.4.1. Motor paso a paso

El motor paso a paso posee un gran desventaja es un elemento ampliamente utilizado en ambientes industriales donde se necesita un par de torque elevado y un control de sus movimientos.

Ventajas:

- Torque elevado.
- Robustez.
- Simplicidad para la programación y control del elemento.
- Bajo costo.

Desventajas:

- Es un mecanismo de lazo abierto.

1.4.2. Servo motor

El servo motor es muy similar al motor paso a paso en sus características donde posee un alto torque, control de sus movimientos, etc. La diferencia es que el servo motor posee un encoder que va guiando su actuación, por lo tanto es de lazo cerrado, lo cual beneficia la pérdida de pasos.

Ventajas:

- Torque elevado.
- Robustez.

- Simplicidad para la programación.
- Posee un sistema de lazo cerrado.

Desventaja:

- Coste elevado.

1.4.3. Motor DC.

El motor DC es un motor es ampliamente utilizado en industria, pero en maquinas de movimiento rotatorio o continuo, este posee un alto par de torque.

Ventajas:

- Alto par de torque.

Desventaja:

- Difícil mantenimiento.
- No posee control de sus movimientos.
- Difícil adquisición.
- De lazo abierto.

1.5 SOLUCIÓN PARA ESCOGER EL ACTUADOR

Todos los candidatos son aptos para a función de contraer la bolsa AMBU, pero la competición real es entre el motor paso a paso y el servomotor. Este último pareciera que es mejor ya que no posee perdida de pasos, pero en realidad la perdida de pasos es un parámetro que se debe considerar para yaqué puede anunciar que la maquinaria posee un deficiencia o mal uso de la maquinaria, para esto se debe determinar el inicio y termino apoyado de la lógica de control, también su coste elevado perjudica el fin que lleva esta maquinaria. Por estas razones se elige que el motor paso a paso es el mejor elemento de movimiento de este proyecto.

CAPITULO 2: DISEÑO MECÁNICO PARA LA MANIPULACIÓN DEL AMBU

1. DISEÑO MECÁNICO PARA LA MANIPULACIÓN DEL AMBU

En esta etapa de la construcción de la maquinaria dispondrá de una serie de etapas las cuales reformarán y dirigirán los modelos actuales, logrando una retroalimentación creaciones según utilización y factibilidad de cada elemento estas etapas son:

- Diseño mecánico y elección de elementos.
- Diseño de componentes 3D y sistemas.
- Diagrama electrónico.

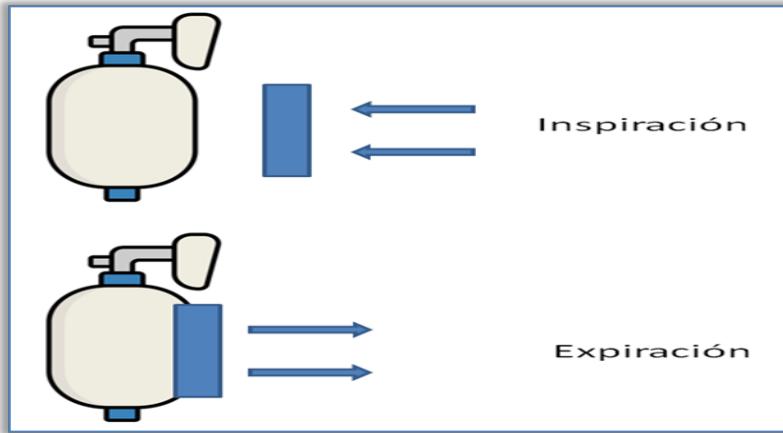
1.1 DISEÑO MECANICO Y ELECCIÓN DE ELEMENTOS

Para este diseño, se ha introducido varios procesos que se centrarán en guiar los diseños ya sea de forma mecánica como también sus beneficios según sea la utilización .

Estos procesos no son estáticos. Son dinámicos y evolucionan en función de la información de cada elemento. Esta retroalimentación es esencial, ya que ayuda a entender cómo se utiliza el equipo en el mundo real, también los ajustes necesarios para mejorar su rendimiento y eficiencia.

El sistema de transmisión ha sido diseñado para ser robusto y duradero. Sin embargo, a medida que se revisa el feedback de diseño, se percata de que hay áreas donde puede ser más ligero sin comprometer su durabilidad. Al hacerlo, no solo se reduce el peso total del equipo, sino que también se mejora su eficiencia energética.

Para esta fase del proyecto, se ha ideado una solución que integra un elemento móvil, cuya principal función es ejercer acciones mecánicas específicas sobre el AMBU: apretar y soltar tal como muestra la figura 2-1.



Fuente: creación Propia

Figura 2-1. Diseño de funcionamiento de AMBU

Esta acción, aunque parece simple en teoría, pero en la práctica requiere de una precisión y fiabilidad absolutas, dada la importancia del dispositivo en cuestión.

La base del diseño gira en torno a un sistema lineal. Este sistema se compone de un carril de guía sobre el cual se desplaza un carro. Este carro, será el encargado de transmitir la fuerza necesaria al AMBU para llevar a cabo las acciones de apretar y soltar.

Junto con la acción a ejecutar se dispone de un elemento tipo paleta en el extremo final donde finalizara su recorrido.

Se planteo la idea de la utilización de 4 dispositivos de rieles para que no ocurra la acción de tijera y perdiera posición (quedando cruzado).

1.1.1. Conjunto de rodamiento lineales

Se dispondrá 2 ejes de acero inoxidable de 8mm y cuatro SCS8UU, para evitar el descuadre y garantizar la simetría de distribución de peso del elemento. Estos ejes se montarán en cuatro soportes SK8, dispuestos en los extremos de cada barra de acero, tal como se observa en la figura 2-2 estos soportes tienen forma de "T" invertida, estos elementos proporcionan la altura necesaria para el conjunto.

De sus características, denota la carcasa de aluminio + LM. UU los rodamientos lineales de bolas, donde su aplicación principal es el trabajo con eje lineales, los rodamientos de bolas

son universales lo cual implica una gran opción de variedad para ser reemplazados en caso de falla, estos dispositivos tienen una gran vida útil, mejorando su eficiencia de trabajo.

Este tipo de dispositivos son ampliamente utilizados en electrodomésticos, industrias, por su bajo ruido, trabajos a alta velocidad como también en maquinarias de construcción.

Resumiendo, este elemento proporciona un movimiento suave, baja fricción, alta rigidez, larga vida útil, es económico y fácil de mantener.



Fuente: <https://www.rmpartes.cl/sku-259/kit-riel-ejes-lineal-8x550-mm>

Figura 2-2. Conjunto de rodamientos lineales

1.1.2. Motor paso a paso

Para dar movimiento a este sistema de carro, se usara el motor paso a paso de acuerdo a las prestaciones y características esenciales de la utilización de este proyecto, junto a la innovación y diseño, esta elección no es casualidad, ya que los motores paso a paso se caracterizan por su capacidad de controlar de manera precisa el movimiento, permitiendo determinar con exactitud tanto la distancia que recorrerá el carro (conocida técnicamente como la carrera) como la velocidad a la que se desplazará.

Estas dos variables son críticas para garantizar que el AMBU sea manipulado de manera adecuada, evitando daños y asegurando su correcto funcionamiento.

Los motores paso a paso son elementos que intercambian la energía eléctrica a energía por movimiento, a diferencia de los motores de inducción simple, estos poseen más bobinas así estas se van energizando secuencialmente, posicionándose el eje del motor según el campo magnético.

Los motores paso a paso se pueden encontrar en distintos formatos como tamaño, peso, corriente de operación etc. Inclusive el tipo de cableado.



Fuente: <https://rhona.cl/producto/682/motor-paso-a-paso.html>

Figura 2-3. Motor pasó a paso Bipolar

Para este requerimiento se determinó el uso de un motor nema 23, modelo 23HS5628 similar a la figura 2-3, el cual tiene las siguientes características:

Posee un ángulo de paso 1,8 grados lo cual, garantiza el control fino del movimiento del rotor, junto con una corriente nominal de 2 Amperes el cual garantiza un torque de 126Ncm o 1.26 Nm, también no menos importante es su inercia de rotor de 300 g*cm², el cual debemos tener en cuenta en las aceleraciones como desaceleraciones.

Para ello se entenderá el inicio y termino de movimiento con rampas de aceleración o desaceleración según sea el caso.

1.1.3. Driver DM556

Para mover el motor paso a paso se utilizo un dispositivo driver DM556 para motor paso a paso como se observa en la figura 2-4.

Este driver posee varios beneficios para este proyecto, uno de ellos es la dinámica de uso del motor paso a paso bipolares, con tan solo 3 entradas se pueden controlar, enable (ENA) el cual habilitará o no el drive, Dirección (DIR) el cual según su estado lógico podemos determinar su dirección de funcionamiento y Pulso (PULL) que en su traducción es tira o jala, este permite atreves de un pulso o flanco, cambiar alimentación las bobinas con ello dar movimiento al motor.

De sus características, posee alimentación entre 9 a 40 V lo cual puede manejar corrientes hasta 4 amperes también posee la capacidad de segmentar la cantidad de pasos llamados micro pasos y subdividir el movimiento.

De la subdivisión o micro step, se puede decir que el motor empleado posee 200 paso, se comprende que debe tener 200 pasos para completar una vuelta, pero el driver empleado hace una subdivisión a eso paso logrando un máximo de 6400, por lo tanto, este drive puede realizar una subdivisión de 32 por cada paso dando un total de 6400 pasos para que el motor complete un vuelta, todo esto es configurable atreves de Dipswitch integrados, estos son pequeños interruptores ordenados de tal manera que tenga una interpretación lógica según la tabla demarcada en el costado del drive, con ello podemos realizar subdivisiones de pasos, también podemos configurar la capacidad de corriente de trabajo.

Del feedback se puede destacar que la utilización de un servo motor sería la solución ya que este no perdería paso, ya que rápidamente ajustaría las pérdidas de paso de forma automática, pero este dispositivo tiene un costo más elevado y también no se puede controlar la perdida de pasos, de acuerdo a esto sigue siendo mejor opción el motor paso a paso.



Fuente: [Figura 2-4. Driver DM556](https://www.cimech3d.cl/producto/dm556-driver-motor-paso-a-paso-5-6a-con-caja-de-proteccion/#:~:text=%C3%97%204%20cm-,Descripci%C3%B3n,de%20entrada%20de%20hasta%202000KHz. (fotografía N°5).</p>
</div>
<div data-bbox=)

Este driver ofrece una limitación de corriente desde 1.4 a 5.6 A, utilizando una protección de sobre corriente, lo cual es perfecto con el uso de una fuente de poder de 24v, también soporta niveles lógicos de 3.3 a 24 v.

Diferencia notable en este driver es el paso atenuado en la conmutación de bobinas, esto se traduce en menos vibración en la bobina del motor por lo consecuente menos ruido, este driver se caracteriza por ser silencioso.

1.1.4. Polea GT2 y correa dentada

Para poder transferir el movimiento desde el eje del motor y desplazar el carro por los rieles, se utilizará un sistema de polea y correa dentada similar a la figura N° 2-11, la cual permite la sincronización del movimiento.

La cinta dentada utilizada en las industrias como maquinarias CNC, esta cinta tiene una nomenclatura de GT2.



Fuente: Correa Dentada Gt2-6 Refuerzo Fibra De Vidrio 5m - Cimech 3d | Cuotas sin interés (mercadolibre.cl)

Figura 2-5. Correa dentada

La imagen 2-5 corresponde a una correa GT2 y esta tiene varios beneficios, ya que es una cinta dentada con un paso entre dientes de 2mm, altura de diente de 0.75 mm, también posee un refuerzo interior de fibra de vidrio el cual permite la duración y robustez necesaria para el movimiento.

También se usará una polea dentada como se muestra en la imagen 2-6, esta permitirá una sujeción al eje del motor el cual tiene que ser de 6.25 mm, se utilizará una polea de 20 dientes. Y del otro extremo de la correa una polea con rodamiento la cual girará libre y será utilizada como tensor de la correa en cuestión.

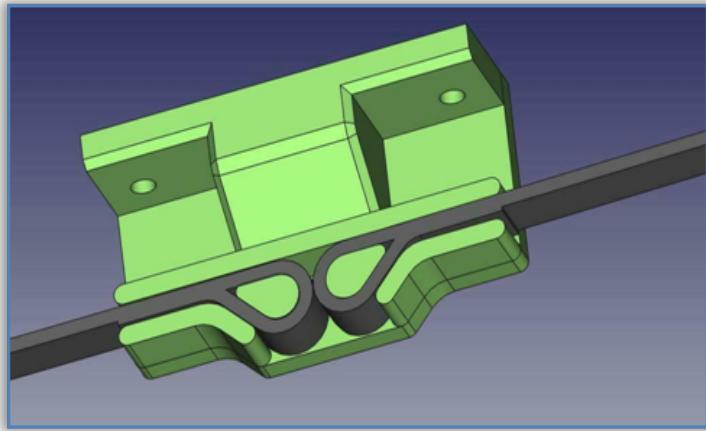


Fuente: <https://mtlab.pe/producto/polea-gt2-10-20t/>

Figura 2-6. Polea dentada

1.1.5. Sujetador de correa dentada

Para el sistema de sujeción de la correa al carro de movilidad se dispone de una pieza de fabricación en 3D la cual brindará sujeción utilizando los mismos dientes de la correa para quedar sujetada de ambos extremos como se muestra en la figura 2-7.



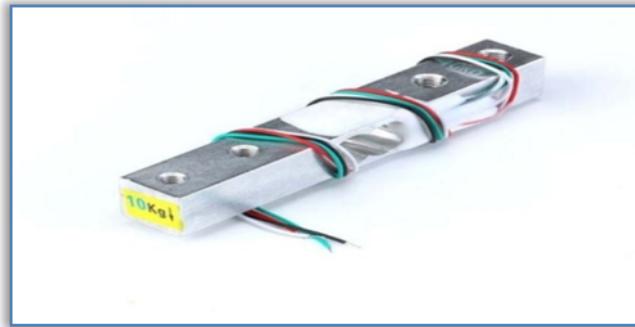
Fuente: <https://cults3d.com/es/modelo-3d/herramientas/y-belt-holder-prusa-i3-rework-1-5-no-tie> (Imagen N°2).

Figura 2-7. Diseño de sujetador de correa

De las pruebas de funcionamiento en el laboratorio se determina que esta disposición, es la más adecuada para el movimiento del carro el cual tendrá como misión apretar el AMBU, ya que no posee mayor intervención y la simpleza del diseño lo hacen realmente funcional.

1.1.6. Celda de carga

Para determinar la fuerza ejercida con el AMBU, se dispondrán de celdas de cargas de 5kg tal como muestra la figura 2-8, estas estarán dispuesta de forma paralela sujetas en una base y su acción será de forma perpendicular al movimiento del carro, con ello lograra realizar ion de medir la fuerza atreves de la presión ejercida al presionar el AMBU.



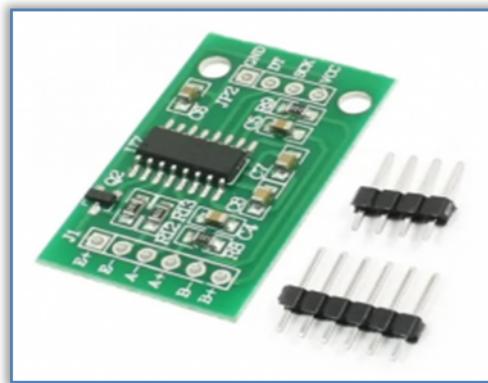
Fuente: <https://www.mechatronicstore.cl/celda-de-carga-10kg-para-pesaje/>

Figura 2-8. Celda de carga

Este elemento es un sensor de fuerza que mide a través de la torsión, lo conforma una barra de aluminio calibrada y fabricada para flexionar en su zona elástica, en esta zona se dispone de una resistencia la cual afecta su resistencia según la deformación del material, esta celda se conforma de un puente de Wheatstone.

1.1.7. Modulo HX711

Para el correcto funcionamiento de la celda de carga se especifica el uso del módulo amplificador HX711 como se muestra en la figura 2-9.



Fuente: <https://altronics.cl/modulo-hx711>

Figura 2-9. Modulo HX711

El módulo HX711 es un conversor entre las celdas de carga y un microcontrolador, permitiendo leer el valor en la celda de manera sencilla. Es compatible con las celdas de carga de 1, 5, 10, 20 y 50 kg. Utilizado en sistemas de medición automatizada, procesos industriales, industria médica.

El chip HX711 posee internamente la electrónica para la lectura del puente de Wheatstone formado por la celda de carga y también un conversor ADC de 24 bits. Se comunica con el microcontrolador por medio de un protocolo de tipo serial mediante 2 pines (Clock y Data).

Las celdas de carga están formadas por galgas extensiométricas en configuración de puente Wheatstone. Para conectar la celda al módulo HX711 son necesarios 4 cables, los colores utilizados habitualmente son Rojo, Negro, Blanco y Verde. Cada color corresponde a una señal como se muestra a continuación:

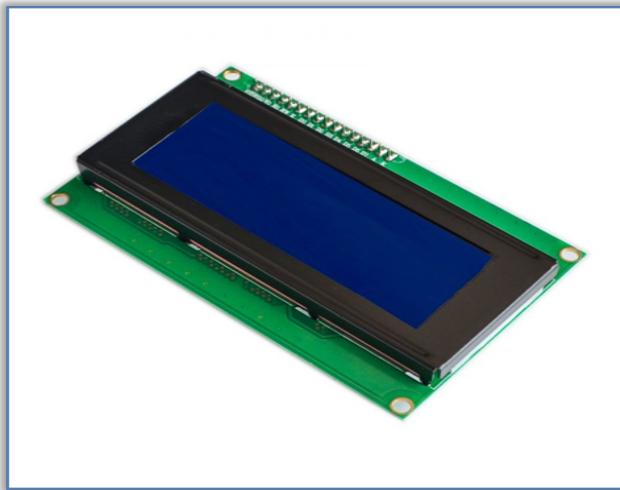
- Rojo: Voltaje de excitación +, E+, VCC
- Negro: Voltaje de excitación -, E-, GND
- Verde: Amplificador -, Señal -, A-
- Blanco: Amplificador +, Señal +, A+.

Es importante recalcar que estos cables utilizan tenciones bajas por lo cual son susceptibles a variaciones de forma inductiva esto quiere decir que pueden variar si se les complementan con equipos o sistema alternos.

El tamaño de la celda de carga varía de acuerdo a su utilización, en este proyecto se utilizó celdas de 5k, por lo cual otras celdas no servirán según el diseño.

1.1.8. Pantalla LCD 20x4

Para visualizar los datos y cambiar parámetros se dispondrá de una pantalla LCD 20x4 esto tendrá mejor interpretación hacia el usuario lo cual se podrá ver cada elemento dentro del funcionamiento de la maquinaria, todo esto para realizar un sistema HMI, esta pantalla posee 20 caracteres por línea y 4 líneas como muestra la figura 2-10.

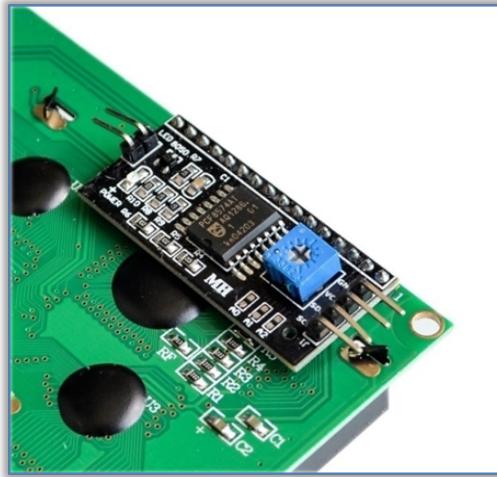


Fuente: <https://www.mechatronicstore.cl/pantalla-lcd2004-con-i2c/>

Figura 2-10. Pantalla LCD 20x4

El LCD 2004 posee 4 filas y 20 columnas de dígitos alfanuméricicos, funciona con el controlador interno HD44780, que es un integrado muy utilizado y para el cual existe amplia documentación. Para conectar la pantalla LCD a nuestro Arduino/PIC se necesitan 6 pines: 2 de control y 4 de datos.

Para poder conectar la pantalla LCD al microcontrolador se utiliza un módulo de conversión de paralelo a serial I2C y el cual se instala en los pines de conexión en la parte trasera de la pantalla como indica la figura 2-11, esto reduce la cantidad de entradas y salidas del microcontrolador que se pueden emplear en algún otro elemento.



Fuente: <https://www.mechatronicstore.cl/pantalla-lcd2004-con-i2c/> (Imagen N°4).

Figura 2-11. Pines de la pantalla LCD 20x4

El módulo incorporado otorga un notable disminuido de cables, debido al protocolo I2C que se basa en un bus de comunicación serial que permite utilizar 2 cables de transmisión de información y otros 2 cables de alimentación.

Por lo tanto, el módulo incorporado I2C, permitiendo conectar la pantalla LCD utilizando solo cuatro cables.

El bus I2C, un estándar que facilita la comunicación entre microcontrolador, memorias y otros dispositivos con cierto nivel de «inteligencia», sólo requiere de dos líneas de señal y un común o masa.

Este integrado fue diseñado por Philips y permite el intercambio de información entre muchos dispositivos a una velocidad aceptable, de unos 100 Kbits por segundo, aunque hay casos especiales en los que el reloj llega hasta los 3,4 MHz.

Del feedback se puede determinar que los cables son susceptibles a variaciones de inducción esto puede afectar a los datos que se entregan, por lo tanto, deben ir protegidos o apantallados.

1.1.9. Botoneras tipo membrana

Se dispondrá la utilización de botonera tipo membrana la cual tendrá la función de cambiar parámetro dispuesto en el menú de operaciones tal como muestra la figura 2-12.



Fuente: https://www.pcfactory.cl/producto/30557-pcf-oem-teclado-de-membrana-1x4--extras?gad_source=1&gclid=EAIalQobChMlp8re1OazggMVZkVIAB1RtgEnEAQYASABEgJstPD_BwE (Imagen N°4).

Figura 2-12. Botonera tipo membrana

El teclado Matricial 4×1 de membrana, incorpora un cable tipo flexor con conector hembra 1x5 con un espacio entre pines de 2.54mm, para una mayor facilidad a la hora de realizar prácticas, bien sea en protoboard, tarjetas de desarrollo, board Arduino, MCU, PIC, AVR, entre otros. Este teclado se puede fijar fácilmente en cualquier superficie ya que incluye película auto adherible.

Su utilización en ambientes húmedos de 90% de humedad, es propicio para la función a ejercer, ya que este dispositivo se limpiará y sanitizará a menudo.

Se proyecta que el teclado pueda estar cerca de la pantalla así poder realizar los cambios según el operador desee, esto facilita mucho la utilización del equipo.

1.1.10. Botoneras PARTIR/PARAR

Para que la maquina pueda realizar la función de incio o bien de detención de su funcionamiento se dispondrá de dos botones uno rojo y el otro verde como se ilustra en la figura 2-13.



Fuente: <https://edelco.co/producto/boton-pulsador-rasante-rojo-parada-nc/>

Figura 2-13. Botón de parada

1.1.11. Luces piloto o indicadores de funcionamiento

Para la visualización de funciones se incorporará luces piloto, de color verde, roja y amarilla como se muestra en la imagen 2-14, los cuales funcionaran a 24 VDC

La luz verde dará cuenta del funcionamiento de la maquinaria, la roja será de detención y la amarilla de alarma.



Fuente: https://eocol.cl/luces-piloto/1116-piloto-verde-22mm-led-24v-ac-dc.html?gclid=EA1alQobChMI073Fz-mzggMV-FRIAB3WGQftEAQYASABEgLiffD_BwE

Figura 2-14. Luz indicador de partida

1.1.12. Fuente de poder

Para la utilización de todos los componentes eléctricos como electrónicos, se dispondrá de una fuente de poder de 24 VDC con una capacidad de 3 Amperes tal como muestra en la figura 2-15.

Esta tensión resulta que es conveniente ya que se utilizara en las luces piloto, en el driver DM556, en el motor paso a paso, al igual que la electrónica la cual comandara todos estos elementos.

También es un acierto la utilización de 24 VDC, ya que, en ambientes húmedos las tensiones no deben exceder este voltaje.



Fuente: <https://triacs.cl/fuentes-de-poder/505-fuente-switching-24v-60w-.html>

Figura 2-15. Fuente de poder

Las fuentes Switching o conmutadas transforman la energía eléctrica mediante transistores en conmutación. Las fuentes de este tipo utilizan transistores polarizados que se conmutan activamente a altas frecuencias (20-100 kilociclos típicamente) entre corte (abiertos) y saturación (cerrados).

Las ventajas de este método incluyen menor tamaño y peso del núcleo, mayor eficiencia y por lo tanto menor calentamiento.

Se escogió este modelo ya que es de fácil adquisición y económica, lo cual es una ventaja si se pretende replicar este modelo en cualquier parte del mundo.

1.2 DISEÑO DE COMPONENTES Y SISTEMAS

Para este desarrollo se utilizo el diseño 3D para evaluar y simular los entornos necesarios para el funcionamiento de la maquinaria, para ello se escogió distintos software de diseño entre ellos INVENTOR de la empresa AUTODESK en la figura 2-16 podemos ver la caratula del software en cuestión.



Fuente: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fimage-grafix.com%2Fproduct%2Fautodesk-inventor-professional%2F&psig=AOvVaw0f0uYRR4rw05vAAfdT3IQs&ust=1699583705687000&source=images&cd=vfe&opi=89978449&ved=0CBMQjhxqFwoTCMjYyK3xtYIDFQAAAAAdAAAABAD>

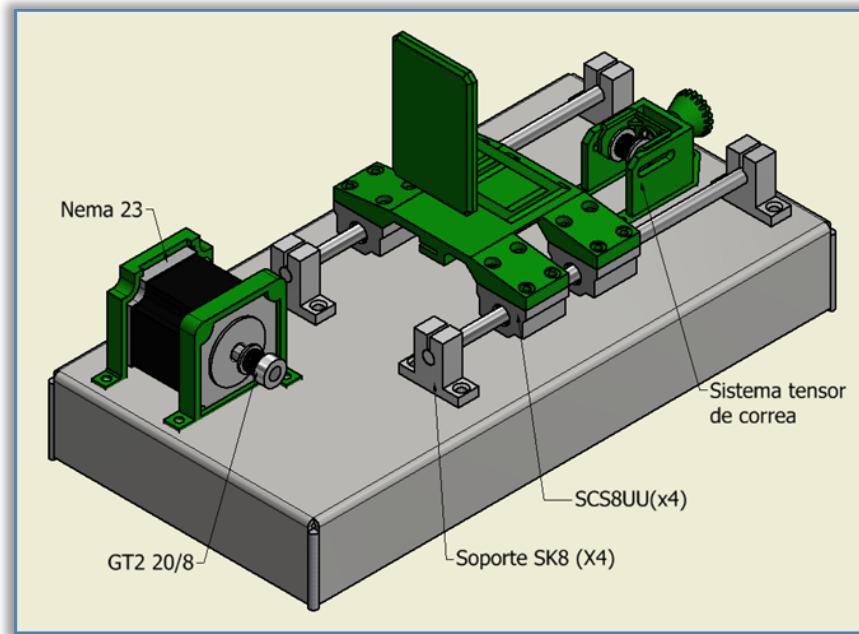
Figura 2-16. Software Autodesk Inventor

El software Autodesk Inventor proporciona a ingenieros y diseñadores una solución de ingeniería y diseño de nivel profesional para el diseño mecánico en 3D paramétrico, la simulación, la visualización y la documentación. Con este software se puede integrar datos en 2D y 3D en un único entorno de diseño, y así crear una representación virtual del producto final que permite validar la forma, el ajuste y la función del elemento antes de crearlo.

Este software incluye funciones dinámicas de uso, edición directa como la representación gráfica mediante diferentes vistas.

2.2.1. Sistema de riel con carro

Para este sistema se establece un conjunto de elementos donde se utilizaran de forma paralela los dos rieles lineales para ser utilizados para dar desplazamiento al carro tal como muestra en la figura 2-17.



Fuente: Creación propia

Figura 2-17. Diseño de sistema de carro

Todos estos elementos estarán afianzados sobre una mesa o superficie, en este caso mesa será de acero inoxidable, lo cual tiene varios beneficios uno de ellos es la limpieza y la protección a la corrosión.

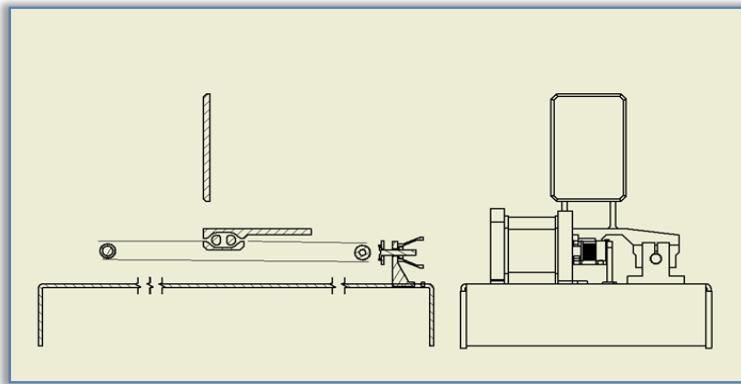
Se puede observar la utilización de 4 rieles SCS8UU, estos se deslizarán por una barra de acero 8mm y 4 soportes SK8 para barra.

También se puede ver el motor paso a paso sujetado con un soporte fabricado en 3D y en su eje la polea dentada GT2 de 10mm.

Esta distribución de elementos será la base que se empleará este prototipo.

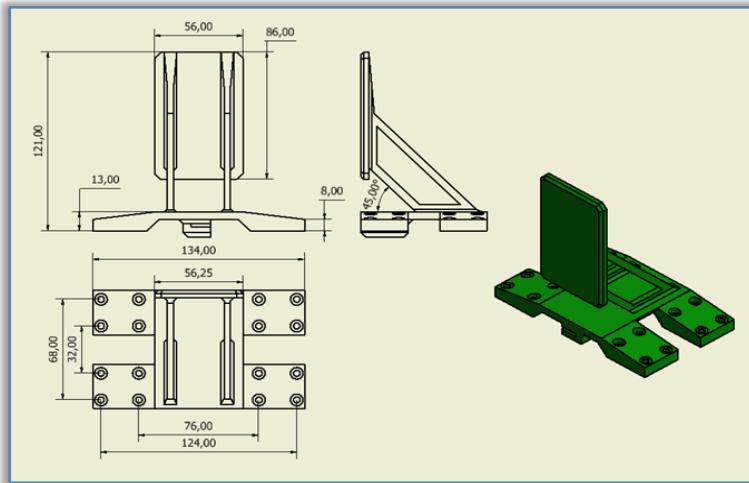
2.2.2. Soporte principal

En la figura 2-18 se puede observar el sistema con un corte transversal el cual muestra la sujeción de la cinta al carro móvil. La correa de este, se fijará por medio de la misma correa utilizando los dientes para lograr la fijación.



Fuente: Creación propia

Figura 2-18. Diseño de soporte principal del carro

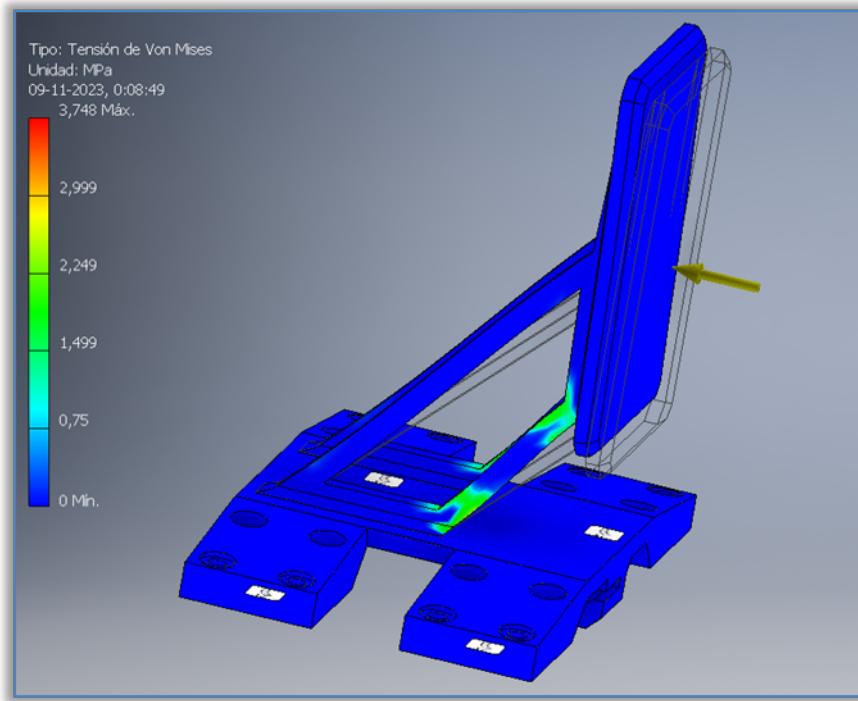


Fuente: Creación propia

Figura 2-19. Diseño de soporte principal del carro (3D)

Sumado a informe técnico se ha integrado herramientas de análisis de estrés y deformación o análisis de tensión-deformación la cual es una prueba que proporciona datos poderosos para medir los cambios de un material bajo los efectos de diversas condiciones de carga. Esta técnica se puede realizar en ambos entornos de trabajo mediante métodos de prueba tanto físicos como también simulado atreves del software.

Este análisis simulado muestra el cambio de color denota el grado de deformación de la pieza mostrando el mismo color y a medida de calcular la deformación esta área cambiará de color tendiendo hacia el color rojo donde esta deformación pasar del área elástica la cual nos brinda una recuperación de la pieza sometida al estrés, por el área plástica, la cual la pieza sufre una deformación importante, por lo que no volverá a recuperar su forma original.



Fuente: Creación propia

Figura 2-20. Diseño de soporte principal del carro (3D vista dinámica del proceso)

La figura 2-20, muestra de forma gráfica y simulada los puntos de inflación como también las áreas criticas sujetas al esfuerzo dinámico de componentes dando estrés a la

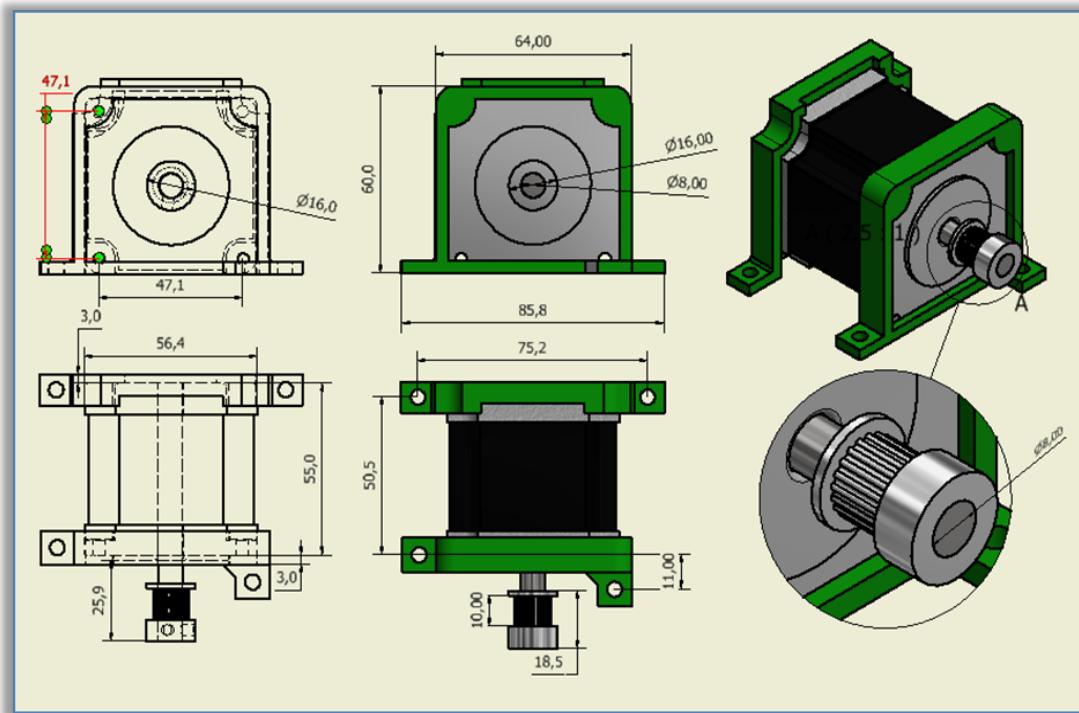
pieza, podemos observar que la unidad de medida es Mpa (Mega Pascal), para traducir a una escala comprensible se debe multiplicar este valor por 10,197 para obtener Kg/cm².

Para este caso el grado de deformación máxima esta cercano a los 3,748 Mpa= 38 kg/cm² esta medición es simulada, pero puede servir en efecto práctico.

Su deformación de trabajo se estable cercano a los 1,874=19,1 kg/cm².

2.2.3. Soporte de motor

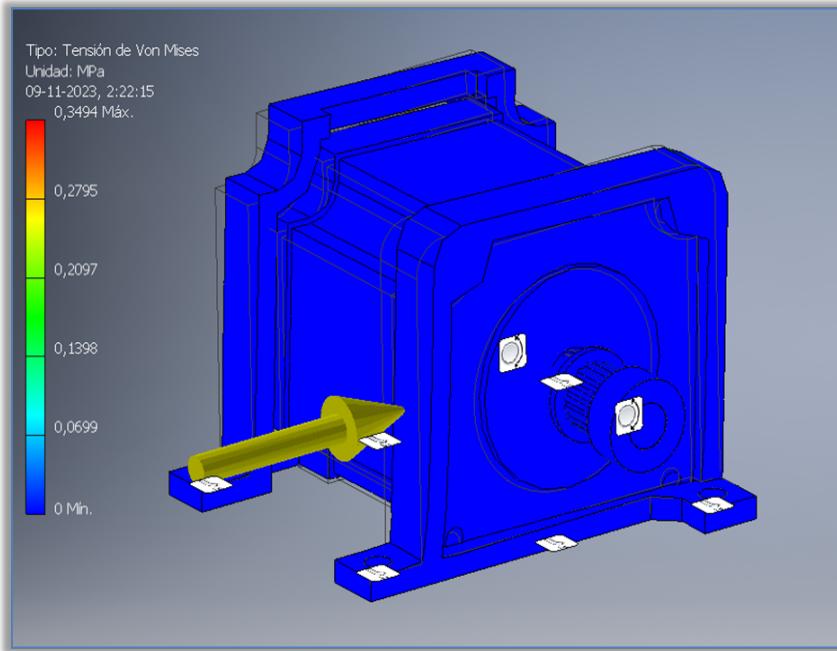
Este diseño implica la creación de dos piezas una en la parte posterior y otra de frente como se muestra en la figura 2-21, los cuales también proporcionarán la sujeción a la placa de metal donde lo sujetan.



Fuente: Creación propia

Figura 2-21. Diseño de soporte del motor paso a paso

Estas piezas se encargarán de sujetar el motor y soportar las cargas dinámicas que ejercerá el motor, de acuerdo al funcionamiento como también la carga de trabajo.



Fuente: Creación propia

Figura 2-22. Diseño de soporte del motor paso a paso (simulación dinámica)

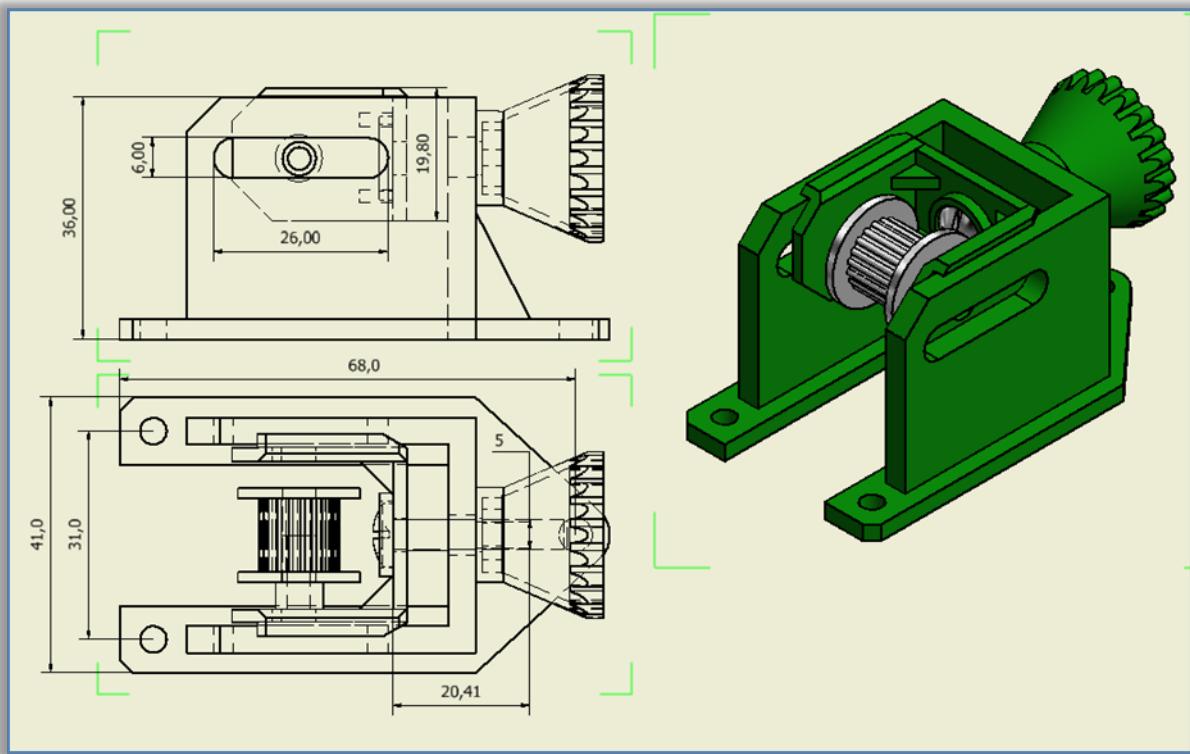
A este conjunto de pieza se aplicó un análisis dinámico del conjunto completo (ver figura 2-22) y se determinó que el análisis simulado estable las zonas plásticas en 0,3494 Mpa=3,562 kg/cm².

Si bien no se observa zonas afectadas, esta deformación radica en la sujeción como también en el área conjunta de contactos entre pieza.

Esta pieza es apropiada para el uso y cargas dinámicas de funcionamiento, logrando una desviación mínima de movimiento al eje del motor con respecto al tensor de la correa.

2.2.4. Conjunto tensor

La figura 2-23 muestra el conjunto de varias piezas las cuales tendrán la función específica de tensar la corre por medio de atracción y cambio de longitud la correa logrando que esta se tense.



Fuente: Creación propia

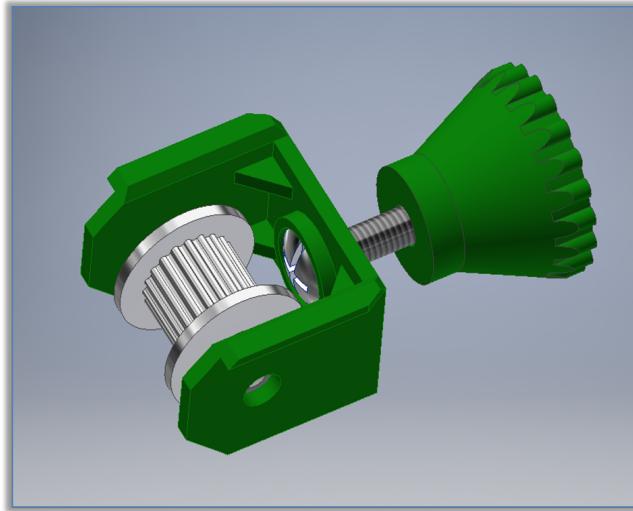
Figura 2-23. Diseño de sistema tensor de correa

La pieza de móvil está ubicada en la parte central del tensor, su función es soportar en su interior la polea dentada, esta tendrá un rodamiento, para evitar el roce cuando esta gire.

Esta pieza funciona como carro deslizante por medio de la cara superior.

Para este efecto tendrá un parilla la cual giraremos para lograr el desplazamiento del carro.

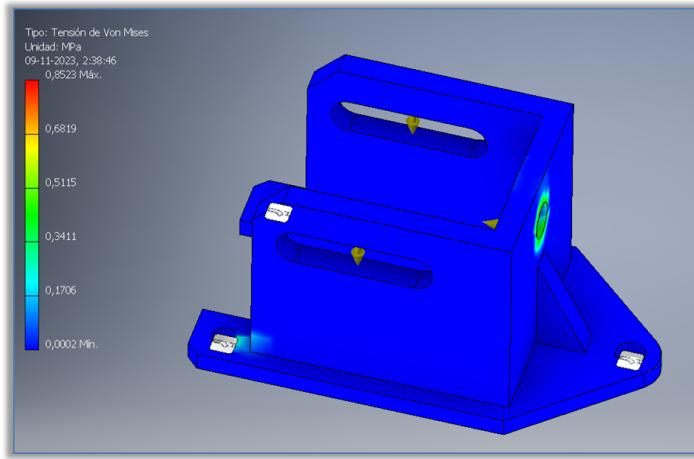
En el interior de esta pieza con forma de "C" se encuentra un perno con tuerca, esto son los encargados de unir de forma mecánica, tal como se muestra en la imagen 2-24.



Fuente: Creación propia

Figura 2-24. Diseño de sistema tensor de correa

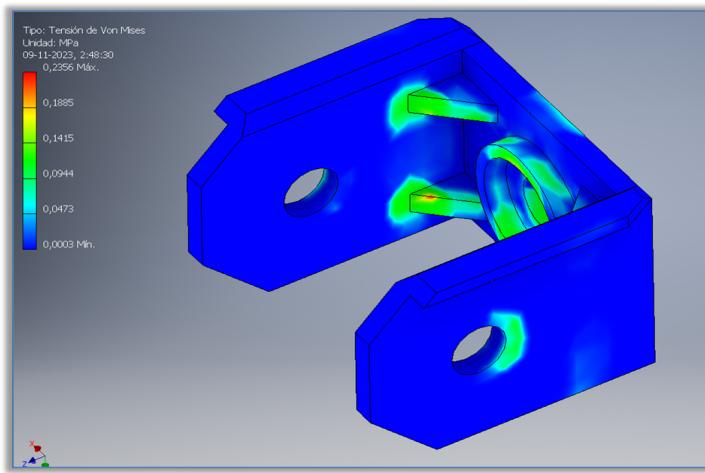
Como se puede observar al girar la perilla se logra reducir la longitud de la polea con respecto a la pared del soporte, con esto realiza la acción de tensar la correa ya que su longitud cambia en uno de sus extremos, se debe recordar que al mantener la correa unida en sus extremos, se obtiene un lazo cerrado.



Fuente: Creación propia

Figura 2-25. Diseño de sistema tensor de correa (Soporte)

En la figura 2-25, se puede observar el análisis simulado de la deformación de la pieza, enfocando la atención desde la perilla. Este análisis nos muestra de forma gráfica, los cambios dinámicos a través de colores, lo cual facilita la interpretación de la deformación, así como la identificación de las zonas críticas. En este caso, la pieza no presenta zonas críticas, lo que permite afirmar que es apropiada para el uso de tensionar la correa, así como para los ciclos de trabajo para los cuales fue diseñada.



Fuente: Creación propia

Figura 2-26. Diseño de sistema tensor de correa (Deslizante)

En la figura 2-26, se muestra la pieza que se desliza para proporcionar la atención necesaria. Dicha pieza debe incluir la polea dentada, la cual se sitúa en su interior y es jalada desde la parte central hacia atrás. Se pueden observar las zonas críticas donde la pieza experimenta una transición de la zona elástica sin pasar a su zona plástica.

Podemos determinar que esta pieza es adecuada y está específicamente diseñada para el trabajo de mantener tensada la correa y además, para soportar la polea dentada.

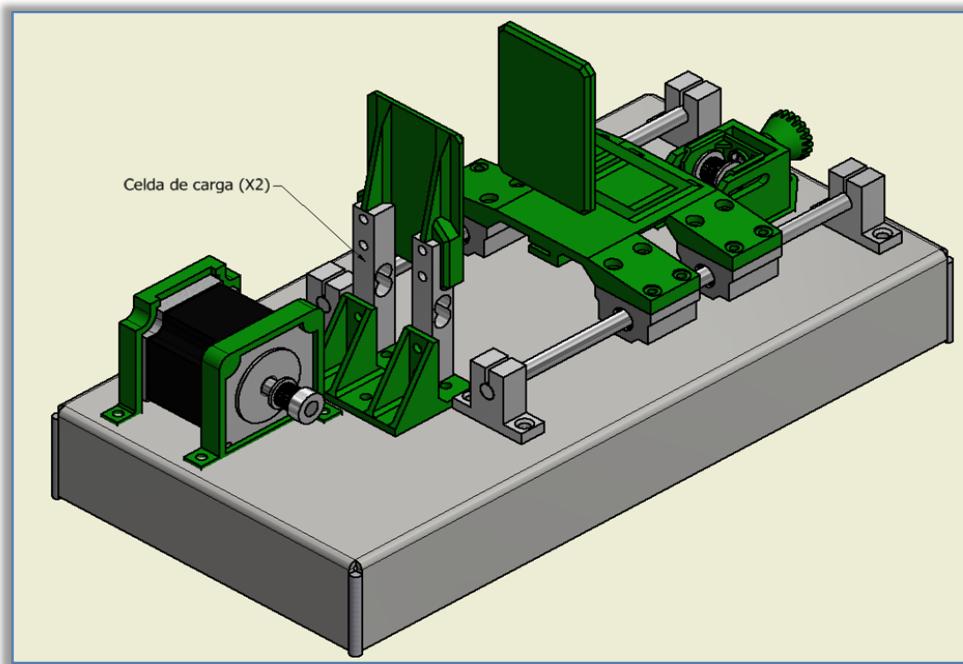
Del feedback del análisis dinámico se incorporo nervios en los ángulos del soporte de tracción o soporte de polea dentada, esto es de beneficio, ya que, mejora estructuralmente la tracción sin comprometer su integridad.

2.2.5. Sistema de medición de fuerza

Unos de los grandes desafíos fue implementar un sistema capaz de medir la fuerza ejercida por el carro hacia el AMBU, también se estableciendo así una alarma de seguridad que se active al superar los niveles máximos permitidos.

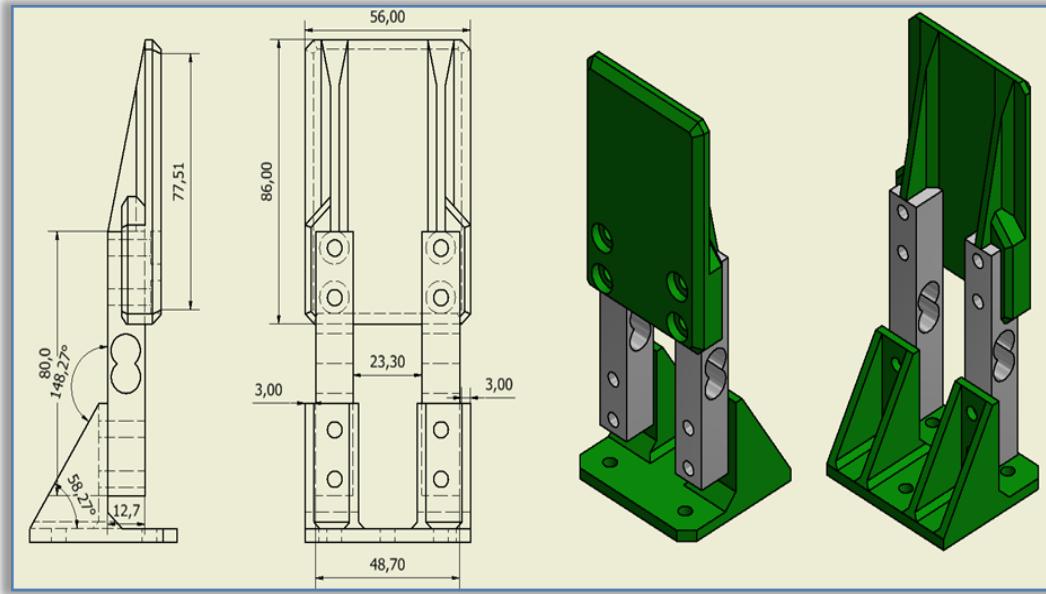
La solución propuesta fue la utilización de celdas de carga, que mediante deformación, pueden determinar la magnitud de la fuerza aplicada. Por lo tanto, se optó por emplear dos celdas de carga de 5 kg cada una.

Se realizó un diseño específico para soportar estas celdas y distribuirlas de manera simétrica como se muestra en la figura 2-27. Además, el sistema incluye una pala que recibirá la carga ejercida por el AMBU. Este sistema de medición de fuerza se instalará en el área donde el carro ejerce la máxima presión que también es el final del sistema de carro.



Fuente: Creación propia

Figura 2-27. Diseño de sistema de sensor de fuerza



Fuente: Creación propia

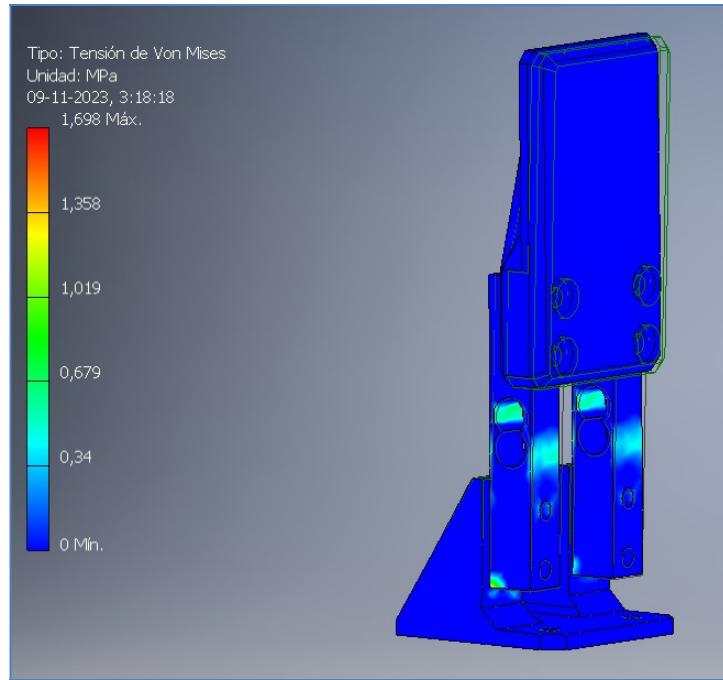
Figura 2-28. Diseño de sistema de sensor de fuerza (celda de carga)

En la figura 2-28 se destaca el diseño gráfico, así como las perspectivas de las vistas donde se ensamblarán y fabricarán las piezas requeridas.

Estas piezas son esenciales para posicionar dos galgas extensiométricas o celdas de carga de manera simétrica, se observa que el diseño se compone de tres partes principales:

La parte inferior, que sostiene las dos celdas de carga, la parte superior con forma de paleta, a la cual se pueden añadir tornillos para su sujeción y una parte media, intencionadamente libre, para permitir la deformación adecuada de las galgas extensiométricas, facilitando así la medición precisa de la fuerza. Además, se aprecian las perforaciones necesarias en la base para su fijación en una placa metálica.

Este diseño ofrece la mejor distribución de fuerzas de acuerdo a sus espesores de 4mm, también los soportes en ángulo para garantizar su firmeza, ya que esta pieza recibirá la fuerza ejercida por el carro.



Fuente: Creación propia

Figura 2-29. Diseño de sistema de sensor de fuerza (Simulación dinámica)

En La figura 2-29, se presenta un análisis de estrés simulado. Donde se observa que las zonas de inflexión se localizan en la parte central de la celda de carga, donde ocurre la deformación necesaria para la medición de fuerzas por medio de las galgas extensiométricas.

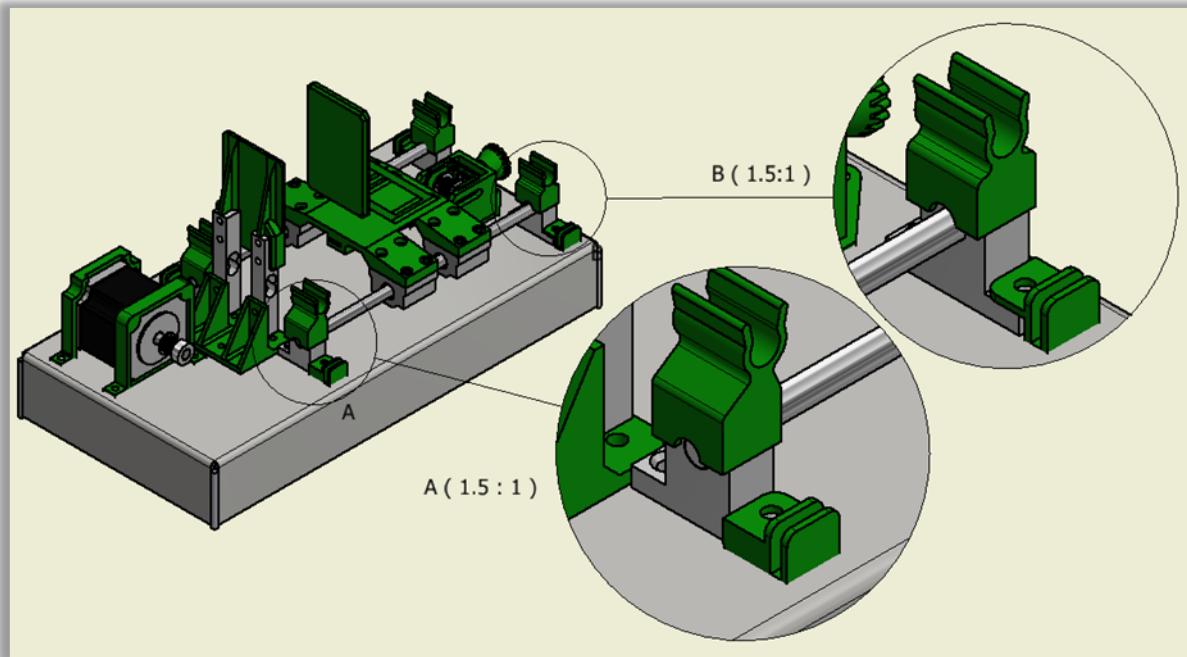
Estas galgas proporcionan datos esenciales sobre las fuerzas ejercidas por el carro hacia el AMBU. Este análisis demuestra que las piezas diseñadas en 3D funcionan adecuadamente, disipando la torsión de manera simétrica hacia los puntos críticos de la celda de carga. Esto confirma la eficacia del sistema de detección de fuerza, asegurando su rigidez en la parte bajo y transfiriendo toda la carga de torsión a la parte central de la celda de carga.

Si bien estas celdas son de 5 kilos, se distribuye de forma pareja la carga, por lo tanto, se puede ofrecer un total de 10 kilos antes de pasar a si zona plástica.

2.2.6. Soporte para acrílico

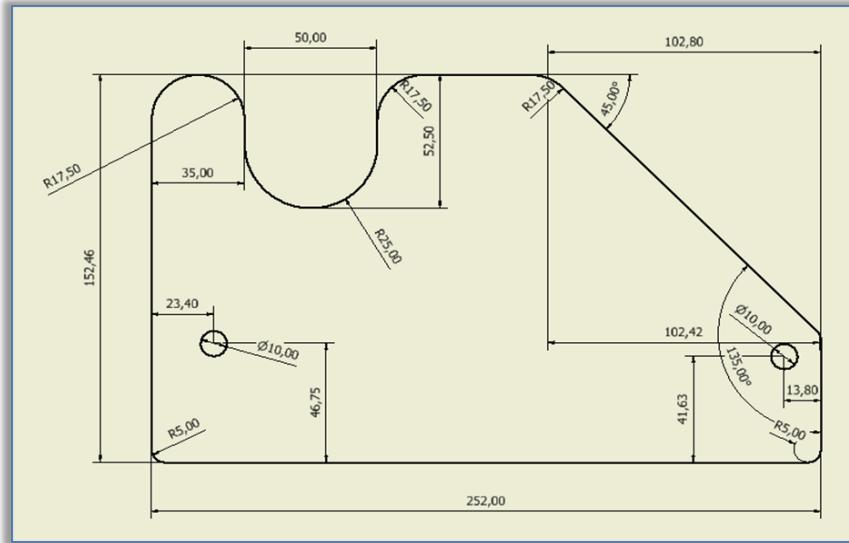
Para poder sujetar el acrílico y dar firmeza al el AMBU, se ha propuesto la utilización de los contornos de pizas existente tal como se muestra en la figura 2-30.

Este sistema de sujeción utilizará las piezas que sujetan los rieles, se ideó la creación de cuatro piezas de soporte, adecuadas para un hilo de tres octavos de pulgada o 10 mm. Estas piezas se utilizarán para asegurar los acrílicos laterales y se colocarán sobre los soportes que sostienen la barra de acero inoxidable. Adicionalmente, en la parte inferior del acrílico, se agregarán otras dos piezas para proporcionar mayor estabilidad a la estructura, sujetándola por debajo esto proporcionará la firmeza necesaria para sostener el acrílico dando estabilidad al sistema.



Fuente: Creación propia

Figura 2-30. Diseño de soporte para acrílico



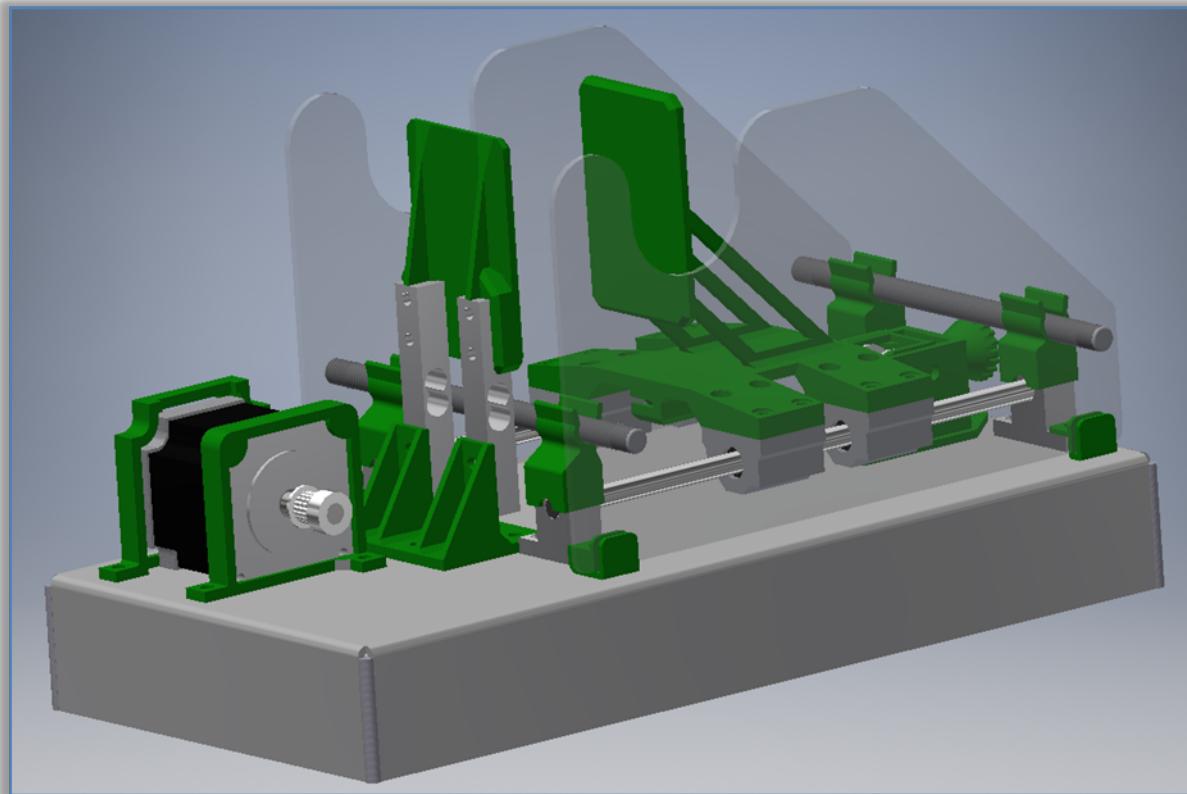
Fuente: Creación propia

Figura 2-31. Diseño de medidas de acrílico

La figura 2-31 muestra el diseño gráfico en 2D, detallando la creación y las dimensiones del acrílico a utilizar, estas piezas serán conformadas por dos elementos de acrílico, por lo tanto, este diseño se deberá cortar de forma espejada o pareada para las dos piezas laterales.

Estas piezas brindarán soporte a la bolsa AMBU. El acrílico, con un espesor de 5 mm, asegura el soporte necesario y permite visualizar claramente lo que está sucediendo, mostrando en qué etapa se encuentra el respirador, el carro y el sensor de fuerza. Además, el diseño incluye las dimensiones precisas y especifica el material requerido para su fabricación, proporcionando una guía completa para su elaboración.

Este diseño se realiza por las ventajas para ser retirado en cualquier momento y poder dar mantenimiento a la unidad.



Fuente: Creación propia

Figura 2-32. Acrílico incorporado

En la figura 2-32, se aprecia claramente la incorporación de los acrílicos laterales. A través de ella, podemos identificar tanto la forma como el tamaño de estas piezas y comprender su funcionamiento en conjunto con el ambo. Es evidente que estos dos acrílicos desempeñan un papel crucial al soportar tanto la parte de la boquilla como la parte posterior de la bolsa de respiración. De esta manera, la bolsa queda situada entre las dos plaquetas, formando parte integral del carro y el sistema de censado de fuerza.

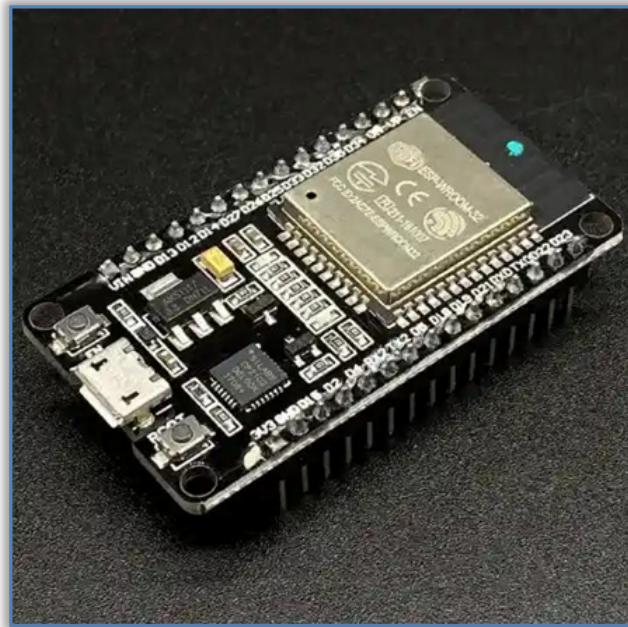
Como se puede observar, el acrílico no tan solo sirve para poder ver lo que sucede al interior de la maquina también ofrece soporte al AMBU y protección al usuario.

2.2.7. El microcontrolador

Uno de los desafíos más grandes es el hecho de juntar todos los dispositivos necesarios y poder controlarlos de forma eficiente, logrando una interacción a través de algoritmos para el correcto funcionamiento, sumado a la capacidad de cálculo es compleja y difícil de procesar lo cual se requiere de un elemento de control para todas las tareas y de bajo costo.

Para esta tarea se escogió el microcontrolador ESP32 WROOM el cual es una excelente elección.

También llamado DEVKIT V1 WROOM 32 (ver figura 2-33), es un potente microcontrolador dual Core basado en un chip Wi-Fi que incorpora Bluetooth, el cual puede ser programado desde el IDE de Arduino.



Fuente:<https://digilog.pk/products/esp32-nodemcu>

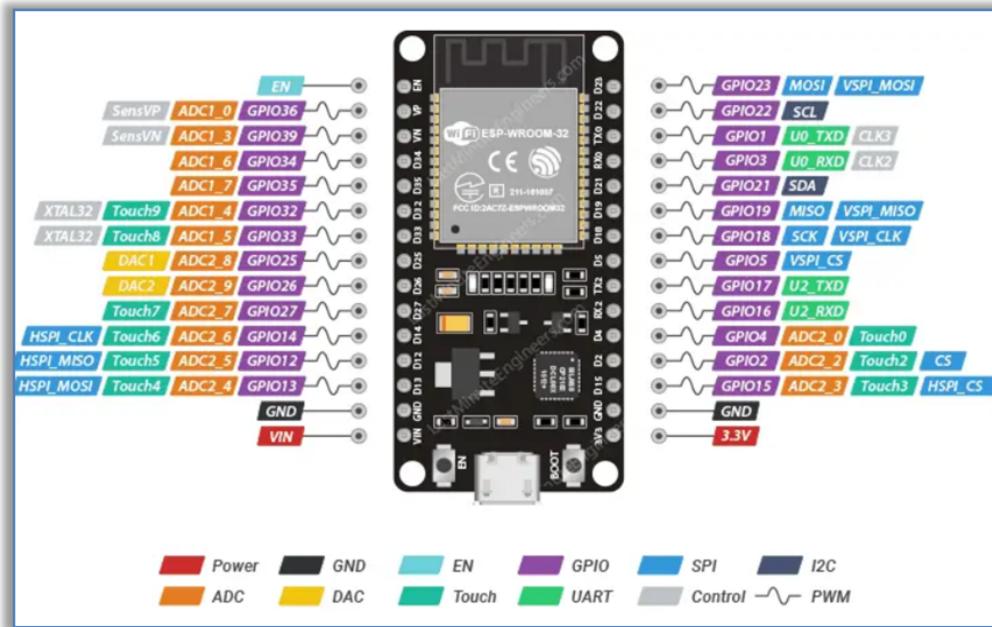
Figura 2-33. Microcontrolador

Como descripción general este módulo es muy usado para prototipo rápido de proyectos con IoT (Internet de las cosas). Integra en una placa el SOM ESP-WROOM-32 que

tiene como base al SOC ESP32, el conversor USB-serial CP2102 necesario para programar por USB el microcontrolador, reguladores de voltaje y leds indicadores.

Por su potencia y pequeño tamaño es muy usado en aplicaciones IoT, redes de sensores de baja potencia hasta las tareas más exigentes, como codificación de voz, transmisión de música y descodificación de MP3.

En la figura 2-34, podemos ver su PinOut de entradas y salidas (I/O).



Fuente: https://lastminuteengineers.com/esp32-pinout-reference/#google_vignette

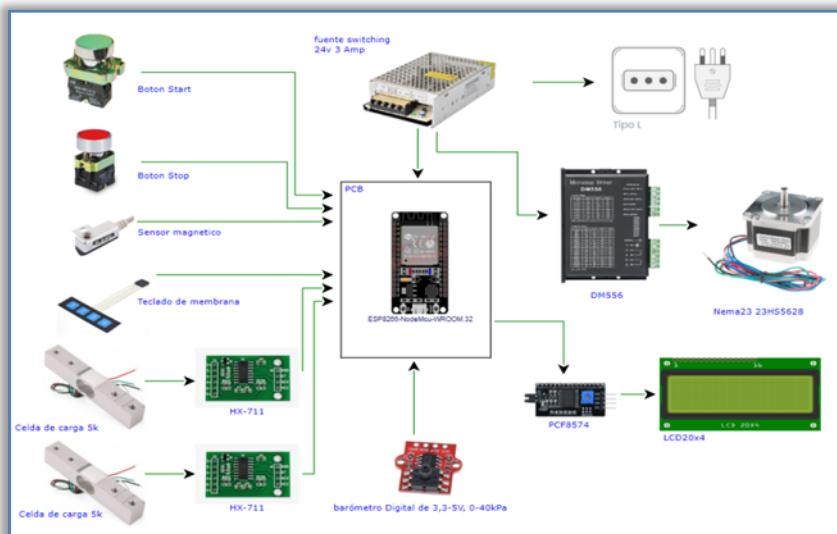
Figura 2-34. PinIN/OUT del Microcontrolador

En su núcleo se encuentra el SoM ESP-WROOM-32 que tiene como base al SoC ESP 32. Hay dos núcleos de CPU que pueden controlarse individualmente, y la frecuencia de reloj de la CPU es ajustable de 80 a 240 MHz.

Dentro de sus especificaciones técnicas se encuentra su memoria ROM de 448 Kbyte, SRAM de 520 Kbyte, 24 pines GPIO de los cuales 16 son PWM, también 18 ADC (Conversor Analógico Digital) 3.3v de 12 bit sumado a 2 canales UART, también lleva embebido una antena PCB.

Para que este microcontrolador pueda interactuar con todas las señales de entradas como salidas las cuales son de nivel de tensión diferentes a lo admitido, se debe acondicionar para que esto no dañe el microcontrolador y a su vez que pueda operar con tensiones superiores como también con señales de datos, se debe realizar una placa electrónica lo cual permita acondicionar estas señales, haciendo que el sistema sea robusto.

En la figura 2-35 podemos ver de forma gráfica todos los elementos de interacción y control que el microcontrolador tendrá que interactuar, ya sea de entrada o de salida, al igual que la alimentación.



Fuente: Creación propia.

Figura 2-35. Ensamble de elementos

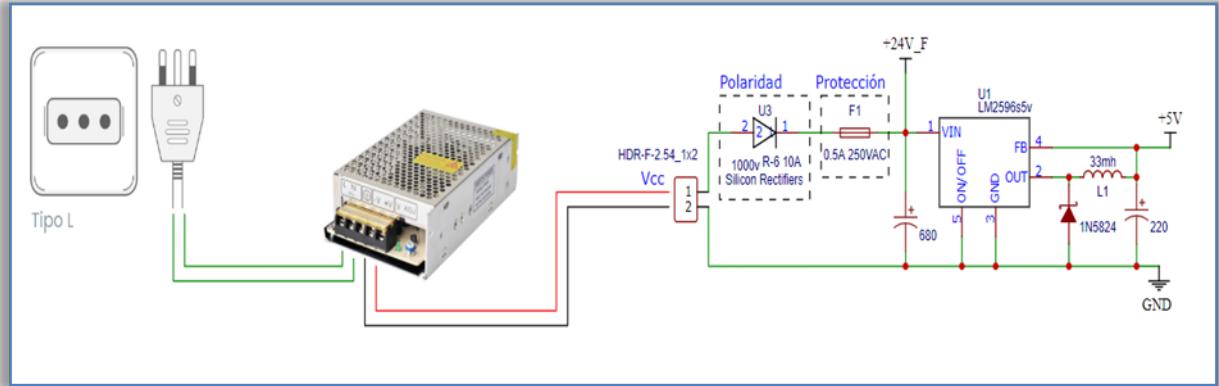
Se puede observar que los elementos de entrada son los botones, sensores, y sus salidas encontramos al motor paso a paso como también la pantalla LCD.

También se observa los acondicionadores de entrada, donde se encuentra HX-711 un amplificador de señal de la celda de carga y controladores de salida como el PCF8574 conversor a LCD con protocolo I2C como también DM556 controlador de motor paso a paso.

Adicionalmente se encuentra la fuente de poder. Esta se encargará de alimentar e interactuar con los dispositivos como también de sus estados lógicos.

2.2.8. Circuito de alimentación

En la figura 2-36, muestra el diagrama donde se puede observar que la fuente switching se conecta a la red eléctrica por medio de un enchufe, si bien esta fuente es universal la cual puede ser utilizada en tensiones de 110 y 220 según el país donde se encuentre.

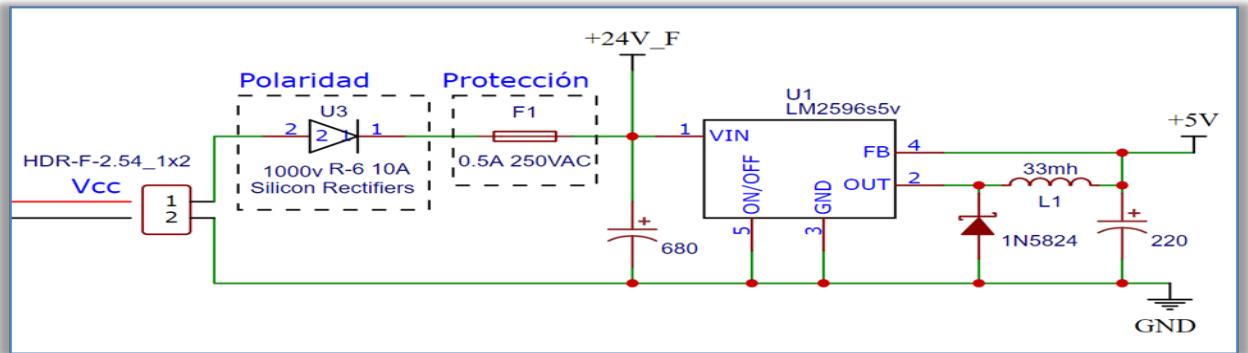


Fuente: Creación propia

Figura 2-36. Diagrama de alimentación

Su salida será de 24V, donde será conectada en los terminales de entrada a la placa electrónica.

Esta placa será estabilizara y regulara el voltaje a 5VDC.



Fuente: Creación propia.

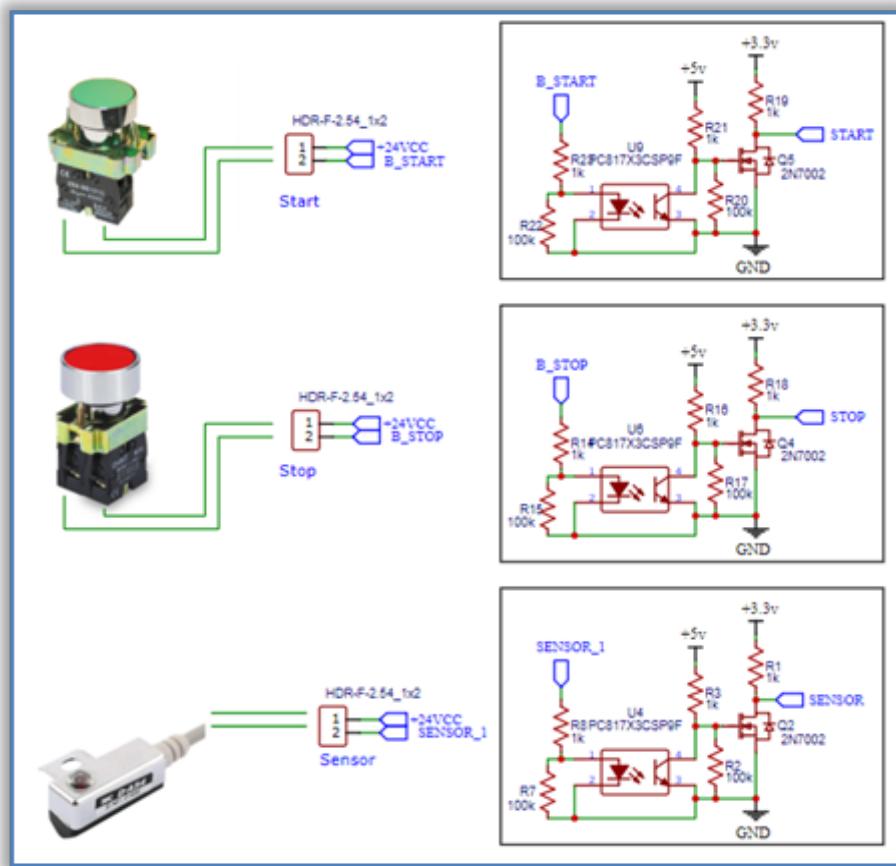
Figura 2-37. Diagrama de alimentación lado PCB

En la figura 2-37 se puede ver el esquemático electrónico, donde su entrada es de 24 VDC, se encuentra un diodo rectificador el cual funciona como protección en caso de conexión de forma inversa, luego un fusible que opera en caso de tener sobre corriente.

En este punto se encuentra con una fuente stepdown la cual posee un regulador LM2596. Con una configuración para trabajar con 5v entregando 3 A de carga.

Es este esquemático tiene como objetivo regular a 5VDC, pero también ofrecer 24 VDC después del fusible de sobre corriente, para tener mejor control en caso de corto circuito.

2.2.9. Circuito de Botón Start, Stop y sensor magnético

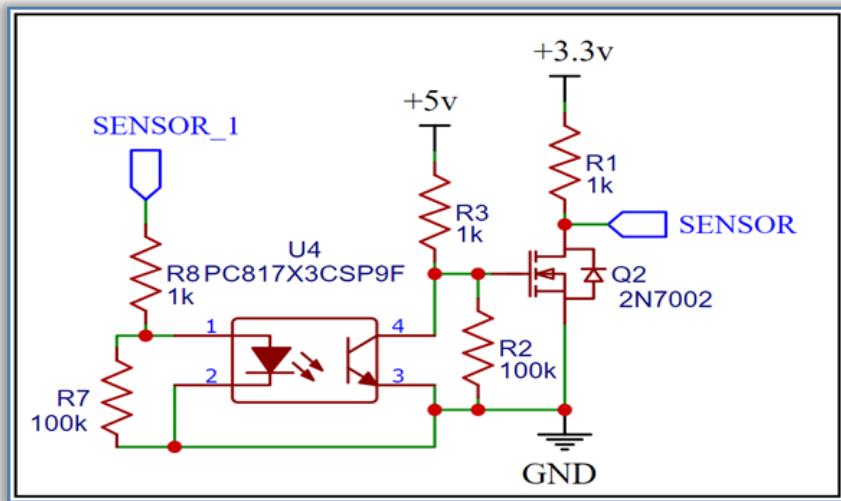


Fuente: Creación propia.

Figura 2-38. Diagrama de botones y sensor

En la figura 2-38 se puede ver las integraciones de botones Start como Stop sumado a al sensor magnético a la placa electrónica. Esta placa permitirá acondicionar los estados lógicos de cada elemento.

Por lo tanto, esta PCB tendrá borneras de conexión hacia el exterior lo cual permite un mejor rendimiento y una fácil instalación.



Fuente: Creación propia.

Figura 2-39. Diagrama de conexión y funcionamiento opto acoplado

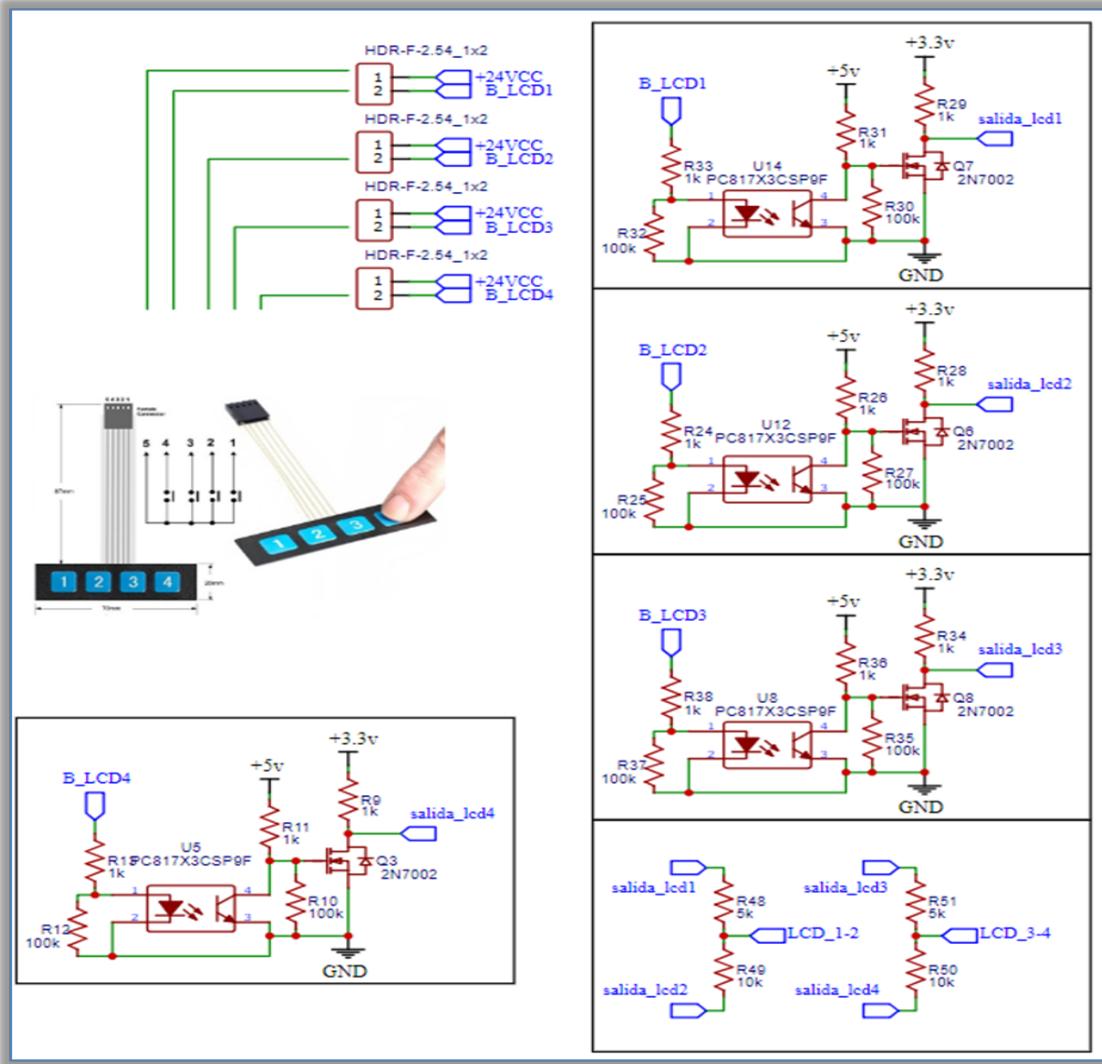
En la figura 3-39 se puede ver el uso de PC817 que es un opto transistor, esto permite separar de forma óptica las señales, para esto fue imprescindible para poder usar un mosfet para normalizar la señal de salida y tomar medianos estados de 3.3V de forma lógica y estable.

Este esquemático reduce el riesgo de inducción entre las líneas.

Se puede observar los pines de entrada los cuales se utilizará en el microcontrolador, donde START será el terminal de entrada número D12, STOP será el número D13 y el sensor magnético será el D32.

Después de la salida de PC817, se observa que se utiliza una resistencia pulldown antes de entrar al mosfet, esto garantiza que la entrada no quede inestable o al aire.

2.2.10. Circuito de botones de membrana

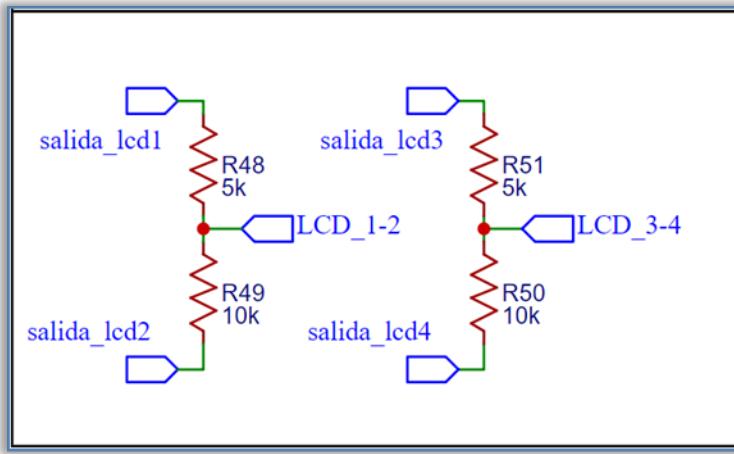


Fuente: Creación propia.

Figura 2-40. Diagrama de conexionado opto acoplado a microcontrolador

La figura 2-40 describe el conexionado y diagrama esquemático para el uso de panel táctil compuesto por botoneras de membrana por lo cual al presionarlo se comporta como botones, para ello se ha conformado de un sistema de conexión el cual nos permite trabajar de forma opto acoplada, lo cual beneficia al microcontrolador por su robustez.

También se puede ver la conexión como la utilización del opto acoplado PC817.



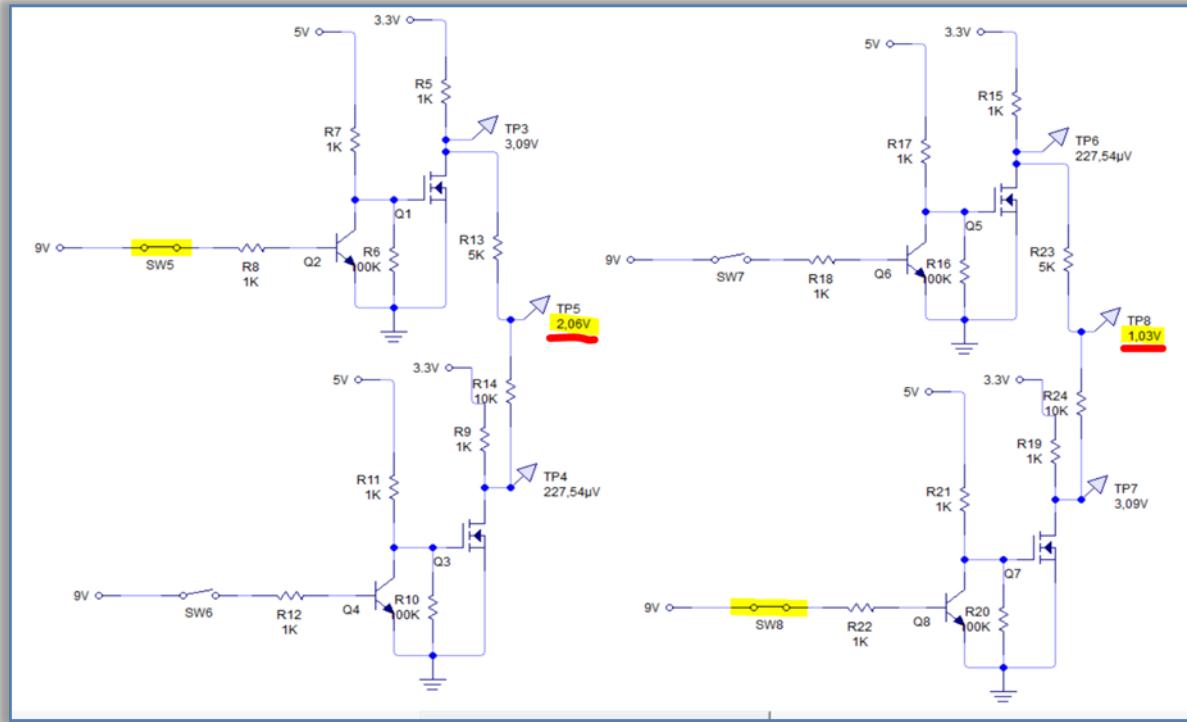
Fuente: Creación propia.

Figura 2-41. Diagrama de divisor resistivo para entrada ADC

La figura 2-41 se puede observar el divisor resistivo, que utiliza dos resistencias en serie con una salida hacia la entrada analógica. Este diseño tiene como objetivo combinar dos entradas digitales en una sola señal analógica. En otras palabras, cuando se presiona un botón, se generará un voltaje específico, y al presionar el otro botón, se obtendrá un voltaje diferente.

La razón de utilización de un divisor resistivo y no una entrada digital es la falta de pines digitales, con ello se utilizará la entrada analógica para este efecto, donde a través de una entrada podemos tener el dato de operación de dos o más botones, en este caso se utilizará dos botones por entrada lógica.

En la figura 2-42, se puede observar la simulación de estados lógicos en conjunto con la salida analógica, lo que permite demostrar el funcionamiento del divisor resistivo al combinar dos entradas digitales.



Fuente: Creación propia.

Figura 2-42. Simulación de esquema de conexión para uso del ADC

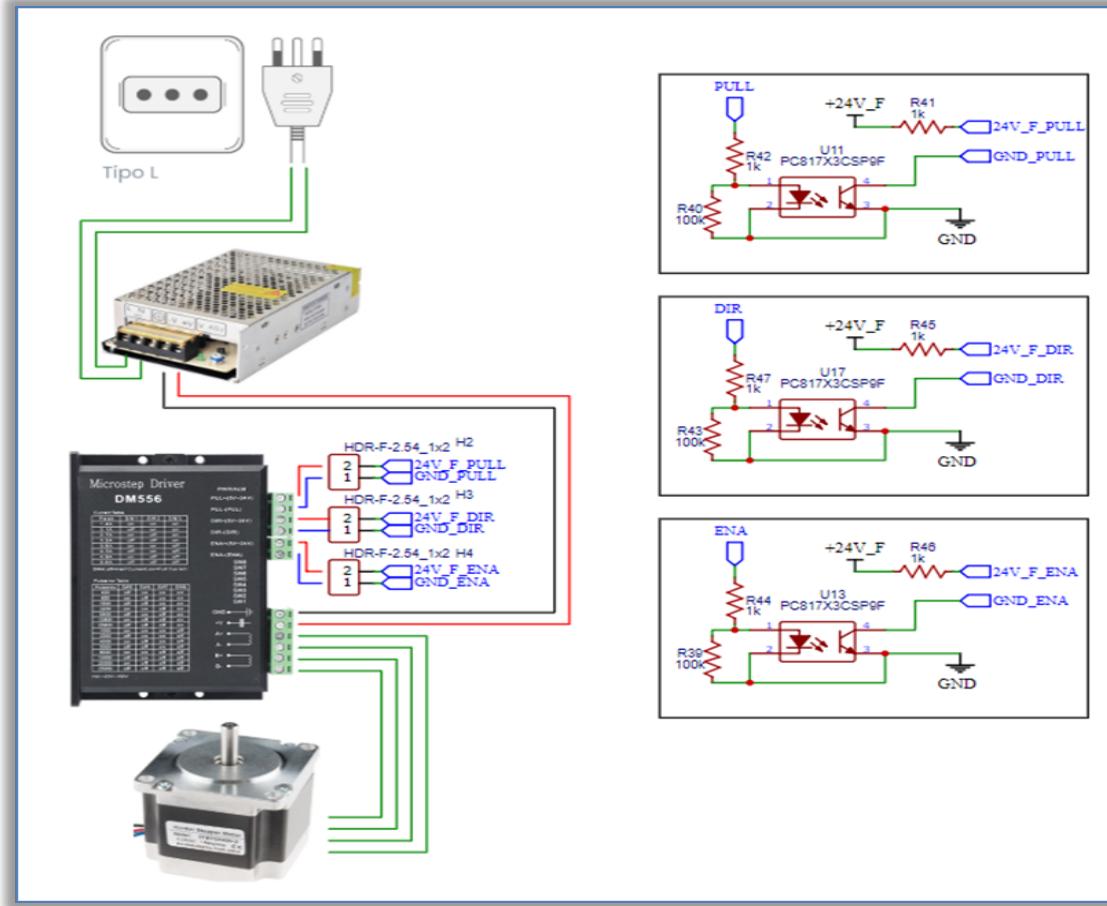
Cuando se presiona el primer botón, la salida del divisor muestra una tensión de 2.06 V. Luego, al soltar el primer botón y presionar el segundo, se puede observar que la tensión se reduce a 1.03 V. Además, al presionar ambos botones simultáneamente, se logra obtener una tensión de 3.3 V.

En la figura 2-59, podemos observar la conexión y las entradas analógicas utilizadas en el microcontrolador. Dos botones, denominados LCD_1 y LCD_2, hacen uso de la entrada analógica D34, como también LCD_3 y LCD_4, que hacen uso de la entrada analógica D35.

Es importante resaltar que el convertidor analógico a digital (ADC) de este microcontrolador tiene una resolución de 12 bits y opera en un rango de 0 a 3.3 voltios. Esto permite dividir este rango en 4095 pasos, cada uno con una diferencia de 0.8 mV. Este alto nivel de resolución nos brinda una gran precisión en la conversión de señales analógicas a digitales.

2.2.11. Circuito de driver para motor paso a paso

En la figura 2-43, se puede examinar el diagrama de conexión y la disposición del driver, que actúa como intermediario en el control del motor paso a paso.



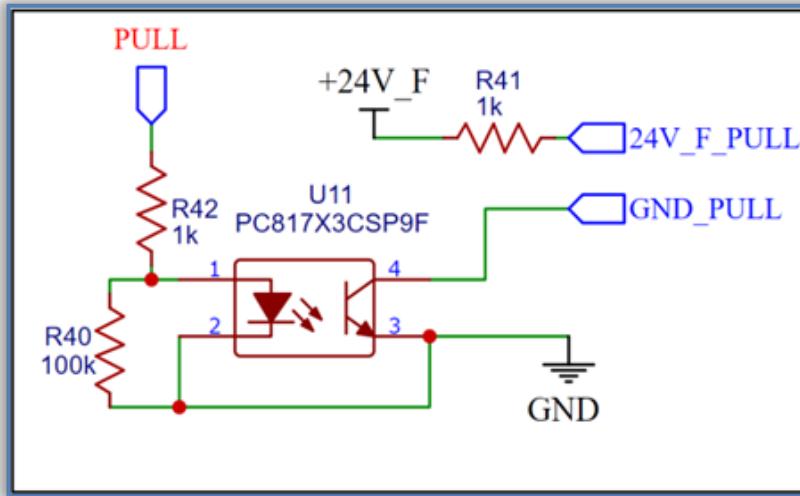
Fuente: Creación propia.

Figura 2-43. Esquema de conexión de drive para motor paso a paso

Este diagrama muestra claramente las entradas del controlador, así como los pines específicos utilizados por el microcontrolador. En esta instancia, se utiliza la entrada ENA del controlador, que actúa como un habilitador para el funcionamiento del motor.

El pin PULL, a través de pulsos en su entrada, proporciona los pasos necesarios para impulsar el motor, alternando la activación y desactivación de sus bobinas. Por último, el pin DIR determina la dirección de movimiento del motor en función de su estado.

Además, en el diagrama, se puede observar que se utiliza la misma fuente de alimentación para el conjunto, lo que asegura una alimentación coherente y estable para el motor paso a paso y el controlador.



Fuente: Creación propia.

Figura 2-44. Salida opto acoplada hacia entrada del driver

En la figura 2-61, se puede ver la utilización del circuito opto acoplado PC817 como también su diagrama de conexión, podemos observar que el sistema utilizará el positivo común lo cual será un sistema PNP, donde se conectarán los mismos negativos.

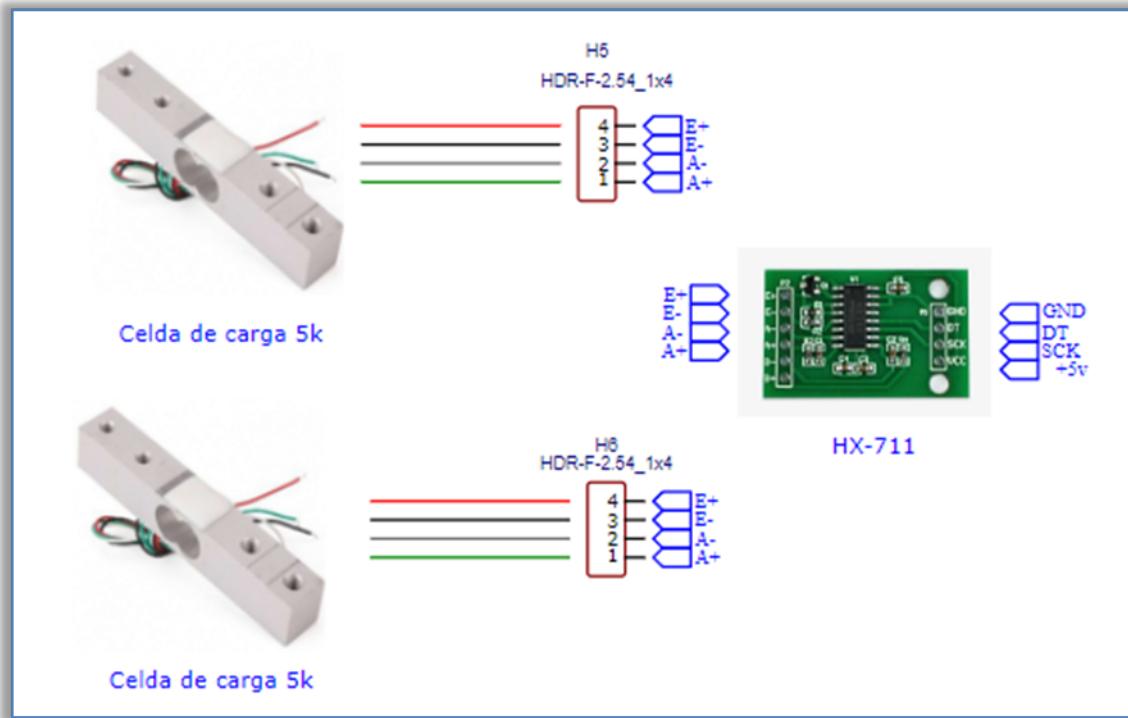
También se observa que el driver se debe alimentar de forma lógica con una resistencia pullUp, ya que se utilizará el optotransistor para poder completar el circuito lógico.

Según su fabricante especifica el funcionamiento del método PNP, ya que se obtiene mejor rendimiento sin pérdida de pasos, ya que los cantes de subida deben ser más cortos.

También podemos observar la utilización de la salida de alimentación 24_F, esto quiere decir que es una salida de voltaje protegida por el fusible, este fusible se encuentra en la PCB y se encarga de todas las señales de 24v hacia el exterior.

2.2.12. Círcuito de celdas de carga

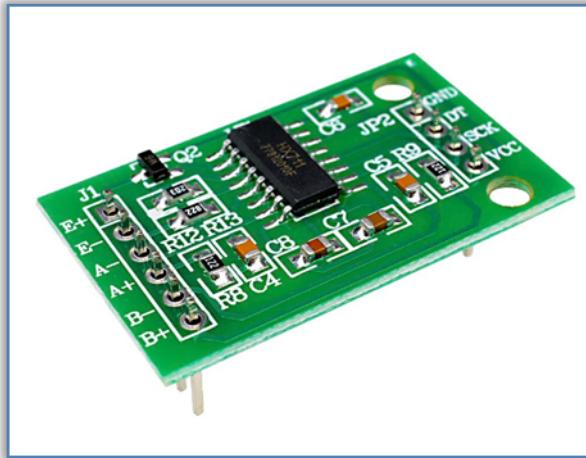
Para esta etapa se utilizará celda de carga que son sensores que se utilizan para medir fuerzas o pesos. Están diseñadas para deformarse cuando se aplica una carga sobre ellas y generan una señal eléctrica proporcional a la fuerza aplicada.



Fuente: Creación propia.

Figura 2-45. Conexionado de celdas de carga

La figura 2-45 muestra la conexión de dos celdas de carga de 5kilos. A un amplificador HX-711, que a su vez están conectados a un microcontrolador. Esta configuración se utiliza para medir con precisión el peso o la carga de dos objetos diferentes simultáneamente. Cada celda de carga está conectada a su propio amplificador HX-711 para garantizar una medición precisa y se utilizan para aplicaciones que requieren un control preciso del peso o la fuerza en tiempo real.



Fuente: <https://altronics.cl/modulo-hx711>

Figura 2-46.Fotografía de modulo HX-711

Para esta etapa se utilizará un amplificador de señal HX-711 son circuitos diseñados específicamente para trabajar con celdas de carga. Estos amplificadores amplifican la señal generada por las celdas de carga y la convierten en una señal digital que puede ser leída por un microcontrolador (ver figura 2-46).

Este módulo tiene una resolución de 24 bit en su ADC donde trasmite hacia el microcontrolador la señal digital por medio de dos cables (Clock y data).

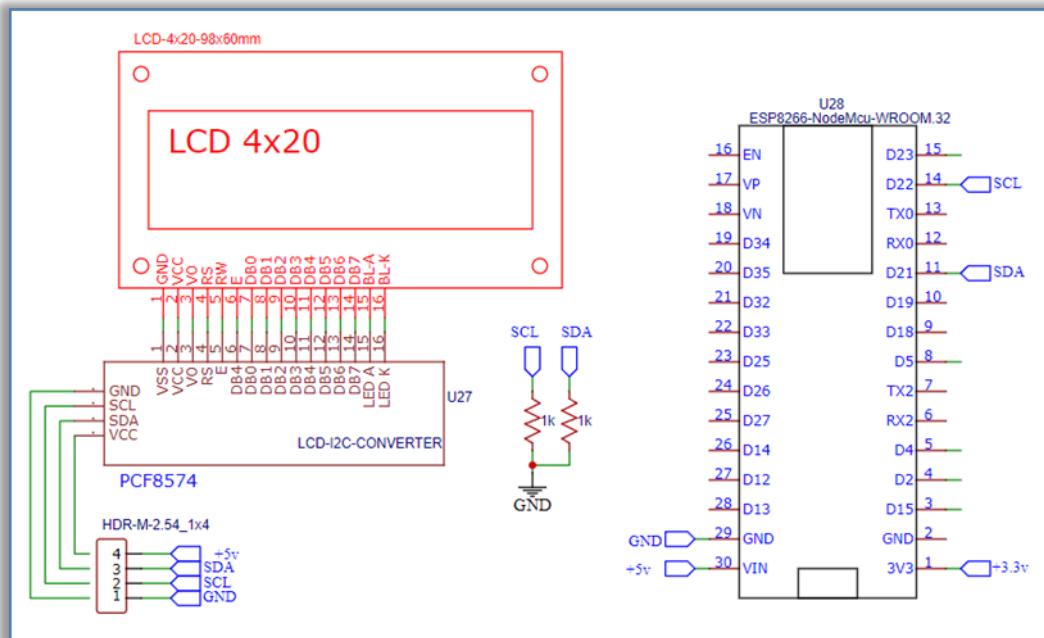
Los pines utilizados en la conexión hacia el microcontrolador serán de la siguiente manera, los pines de entrada serán D27 y D14, mientras los del modulo HX711 serán DT y SCK correspondientemente de celda de carga.

Como se puede observar las dos celdas de carga se conectarán en paralelo, pero de igual manera se compensa para que hayan variaciones, este modelo es muy utilizado en balanzas donde hay más de 4 celdas de carga.

2.2.13. Circuito de LCD

Para visualizar los datos del proceso y también poder modificar cualquier parámetro mientras está en funcionamiento, se ha incluido una pantalla LCD de 20 caracteres y 4 filas (20*4), la cual se conectará a un módulo PCF8574 el cual permite pasar los datos de forma paralelo a serial reduciendo la cantidad de pines utilizados para su funcionamiento.

Es un gran desafío poder utilizar el protocolo I2C en este proyecto, ya que, el sistema pasa de ser de comunicación paralela a una señal serie, lo cual reduce la rapidez de activación de diferentes elementos , como por ejemplo el motor paso a paso.



Fuente: Creación propia.

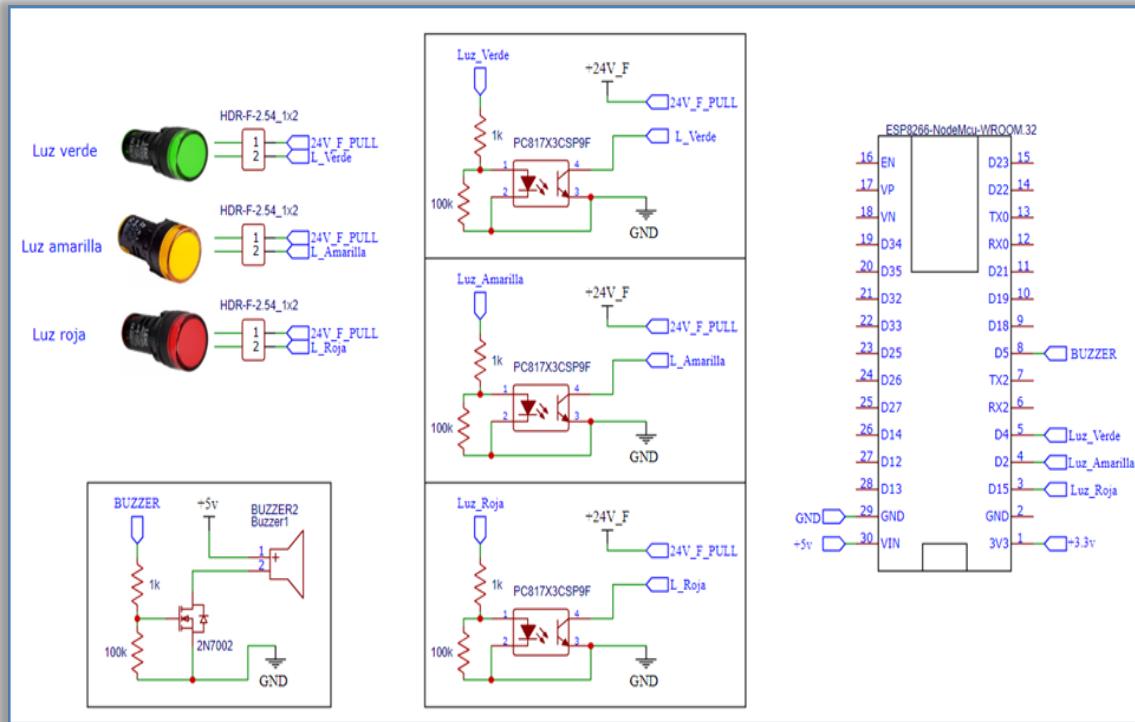
Figura 2-47. Esquema de conexión para uso de PCF8574 y pantalla LCD

En la figura 2-47 podemos ver el diagrama de conexiones para la interconexión de elementos, donde utilizaremos el protocolo I2C, mediante dos pines SCL y SDA, por lo que se utilizará las salidas digitales D22 y D21 correspondientemente.

Estos pines de conexión están establecidos para el protocolo de comunicación I2C, en la tarjeta de desarrollo.

2.2.14. Circuito de luces de proceso

La figura 2-67, nos muestra gráficamente la conexión de las luces las cuales usarán una tensión de 24 volt corriente continua, la misma de la fuente, pero de igual forma se proyectó el uso de la tensión de fusible, que, en caso de corto circuito, operaría el fusible de la placa base.



Fuente: Creación propia.

Figura 2-48. Esquema de conexión para luces de estado y buzzer

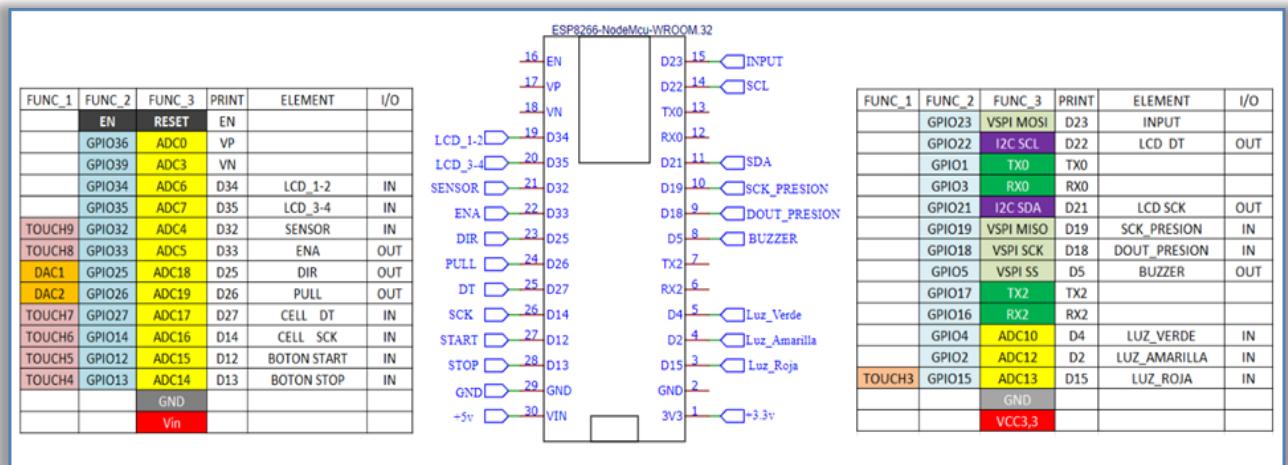
Las entradas para Las luces de estado serán asociadas a las entradas D4, D2 y D15 las cuales se identificarán, luz de color verdes, amarilla y roja correspondientemente, La conmutación será de forma opto acopladas.

También es este circuito se proyecta el uso de un dispositivo sonoro el cual se registrara como buzzer, este dispositivo nos alertara según la configuración.

El dispositivo de buzzer utilizara la salida D5.

2.2.15. Disposición de pines y función

En la figura2-49 podemos observar la disposición de pines y la función de cada uno, si entrada o salida como también la alimentación de entrada 5v como también salida de 3.3v. Todo esto para garantizar una programación específica de cada elemento según la disposición de pines previamente seleccionados, al igual que en la tabla 2.1.



Fuente: Creación propia.

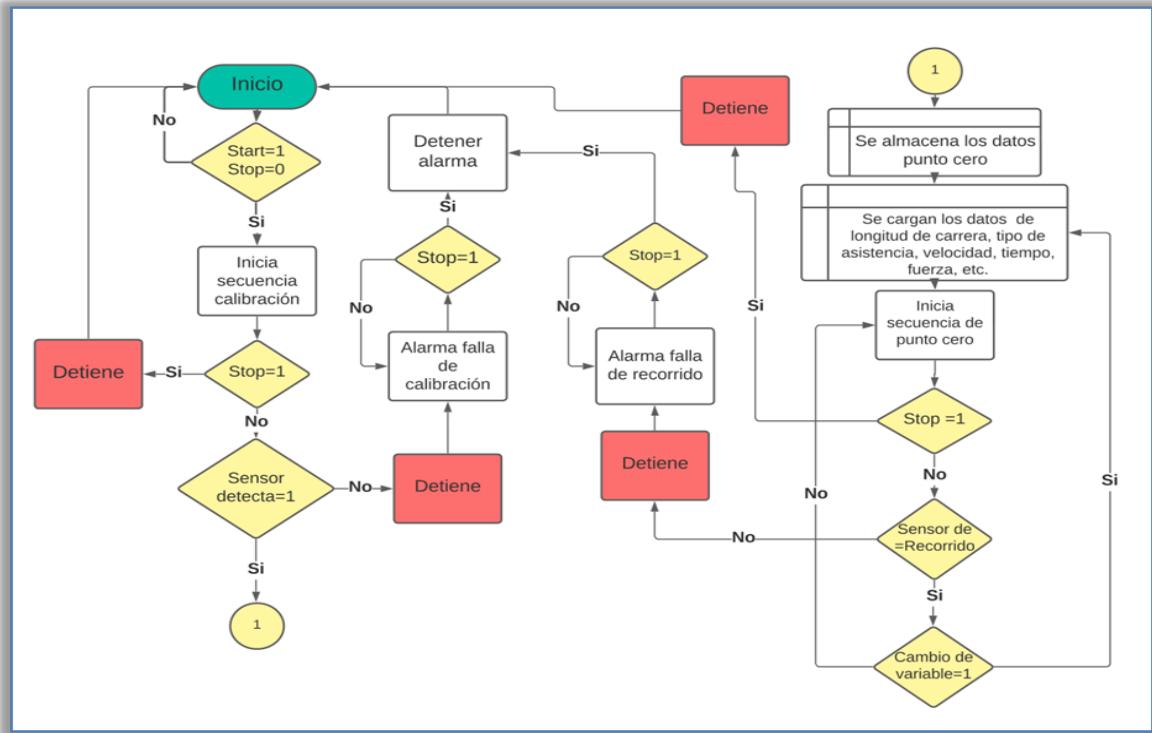
Figura 2-49. Disposición de pines y función, en microcontrolador

Tabla2.1. Disposición de pines en microcontrolador

PRINT	ELEMENT	I/O
EN		
VP		
VN		
D34	B_1-2	IN
D35	B_3-4	IN
D32	SENSOR MAGNETICO	IN
D33	ENA	OUT
D25	DIR	OUT
D26	PULL	OUT
D27	CELDA_DT	IN
D14	CELDA_SCK	IN
D12	BOTON_START	IN
D13	BOTON_STOP	IN

PRINT	ELEMENT	I/O
D23	INPUT AUX	IN
D22	LCD SCL	OUT
TX0		
RX0		
D21	LCD SCK	OUT
D19	PRESION SCK	IN
D18	PRESION SCK	IN
D5	BUZZER	OUT
TX2		
RX2		
D4	LUZ VERDE	OUT
D2	LUZ AMARILLA	OUT
D15	LUZ ROJA	OUT

Fuente: Tabla de disposición de pines.



Fuente: Creación propia.

Figura 2-50. Lógica de funcionamiento de maquinaria

En la figura 2-50, podemos observar el diagrama de funcionamiento básico del equipo, con un enfoque especial en el proceso cíclico esencial.

Para iniciar el proceso, primero debemos asegurarnos de que el botón de "Stop" no esté presionado y luego pulsar el botón de "Start". Esto iniciará la secuencia de calibración, durante la cual el carro se moverá hacia una posición inicial específica determinada por un sensor. Si el sensor no se detecta y el carro no logra encontrarlo, la etapa de calibración se detendrá y se mostrará una alarma de "Falla de calibración". Sin embargo, si el sensor es detectado, la secuencia continuará, cargando todos los datos necesarios para llevar a cabo la operación.

Es importante destacar que en cualquier momento en que haya movimiento involucrado, podemos detener la maquinaria de inmediato presionando el botón de "Stop"

CAPITULO 3: DESARROLLO

2. DESARROLLO

Para este punto donde se debe desarrollar la maquinaria necesaria, se dispondrá de varias tecnologías al igual que de varias etapas para lograr la confección total del funcionamiento como también de la lógica de funcionamiento, todo esto embebido para lograr la robustez necesaria para su óptimo funcionamiento.

Para este paso se han tomado en consideración las siguientes etapas:

- Impresión 3D.
- Adquisición de materiales.
- Construcción electrónica.
- Programación
- Descripción de interfaz y funcionamiento.
- Valores y costos.

2.1 IMPRESIÓN 3D

En esta etapa se realizará la configuración diseño como también los alcances de poder materializar de forma práctica en ambiente 3D como también la fabricación por medio de técnica de impresión 3D.

Para esto necesitaremos entender algunos conceptos básicos para poder familiarizarnos con lo que es el desarrollo al igual que la configuración y ajustes de diferentes ambientes de manufactura.

Para lograr entender desde lo básico hasta lo más complejo debemos comprender el concepto CAD, el cual se describe como el Diseño asistido por computadora (Computer Assisted Design), el cual es un software que nos permite crear, modificar, analizar y optimizar planos o modelos en dos o tres dimensiones, que nos permite manipular de forma sencilla elementos geométricos.

1.1.1. Laminador o Slicer

La tecnología de impresión 3D es el proceso por el cual, se hace un objeto tridimensional, casi de cualquier forma, a partir de un modelo 3D, básicamente bajo un proceso de adición de material. Es por eso que también se denomina manufactura aditiva. Esto es, El objeto es formado por la superposición de capas de material que se agregan una y una.

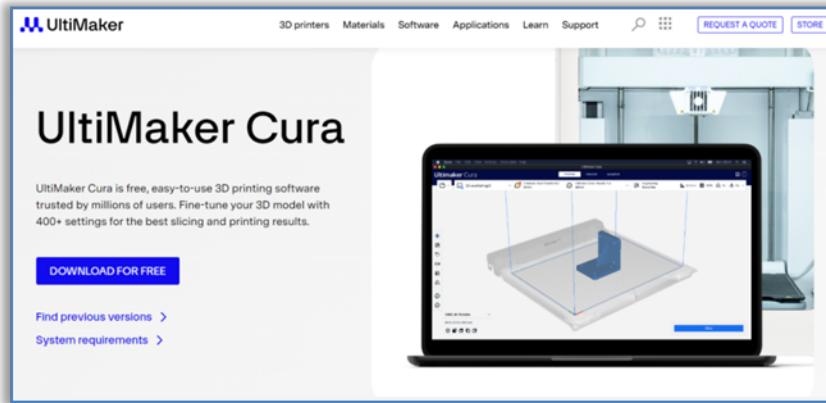
Para lograr esto, en el mercado hay una gran variedad de laminadores cuya función es poder brindar solución al momento de interactuar con la impresora 3D, este laminador nos permite ubicar nuestro modelo 3D en un espacio tridimensional similar al espacio de impresión de nuestra impresora 3D, para luego dividir en una serie de capas nuestro modelo como también generar la mejor ruta de deposición de material y conjunto a esto lograr unir las capas entre sí, también ofrece múltiples elementos de apoyo para la impresión, ya sea, aéreas con voladizos, sectores vacío, o elementos donde es necesario extremas las función de impresión dando mayor área de relleno.

Para nuestro proyecto ocuparemos el laminador Ultimaker Cura, como se muestra en la figura 3-1, el cual es uno de los laminadores mayormente conocidos, posee una gran comunidad de ayuda, como también tutoriales en todas las categorías.

En la instalación del software de laminación debemos descargarlo en la página de ultimaker, en el siguiente enlace se encontrará la versión 5.6.0 a la fecha:

<https://ultimaker.com/software/ultimaker-cura/>

Este software se puede descargar cual sea su distribución de sistema operativo ya sea de 32 o 64 bit como también en sistema OS o Linux.



Fuente: <https://ultimaker.com/es/software/ultimaker-cura/>.

Figura 3-1. Página principal para descargar el software.

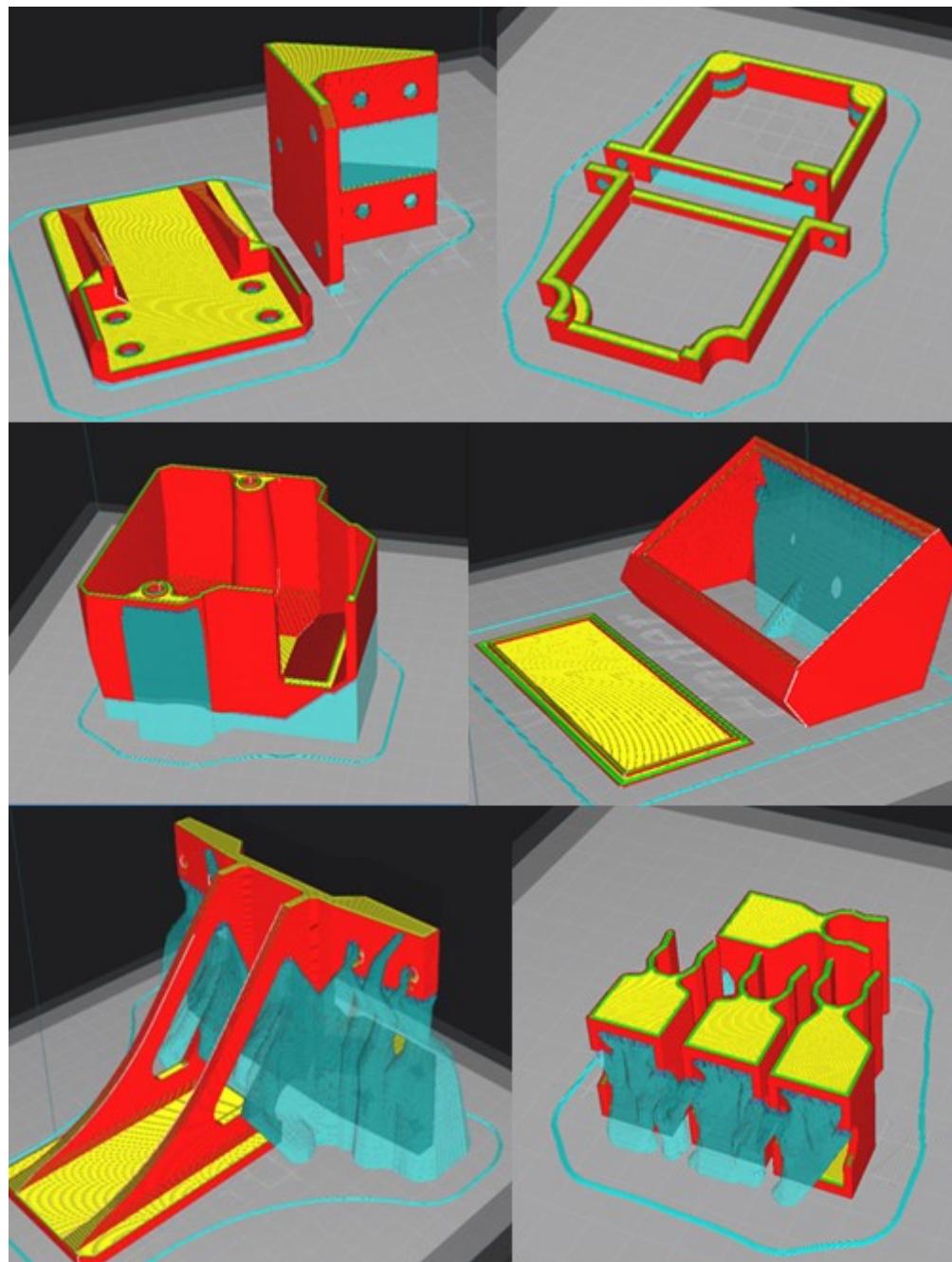
Después de instalar nuestro laminador, nos concentraremos en conocer los tipos de archivos los cual admite como también los archivos de salida.

En nuestro impreso hay configuraciones necesarias las cuales puede servir para muchas piezas, ya que, son transversales mientras en otras piezas a imprimir por su forma tendrá que ser ajustes distintos.

Para nuestra impresora configuraremos los siguientes parámetros:

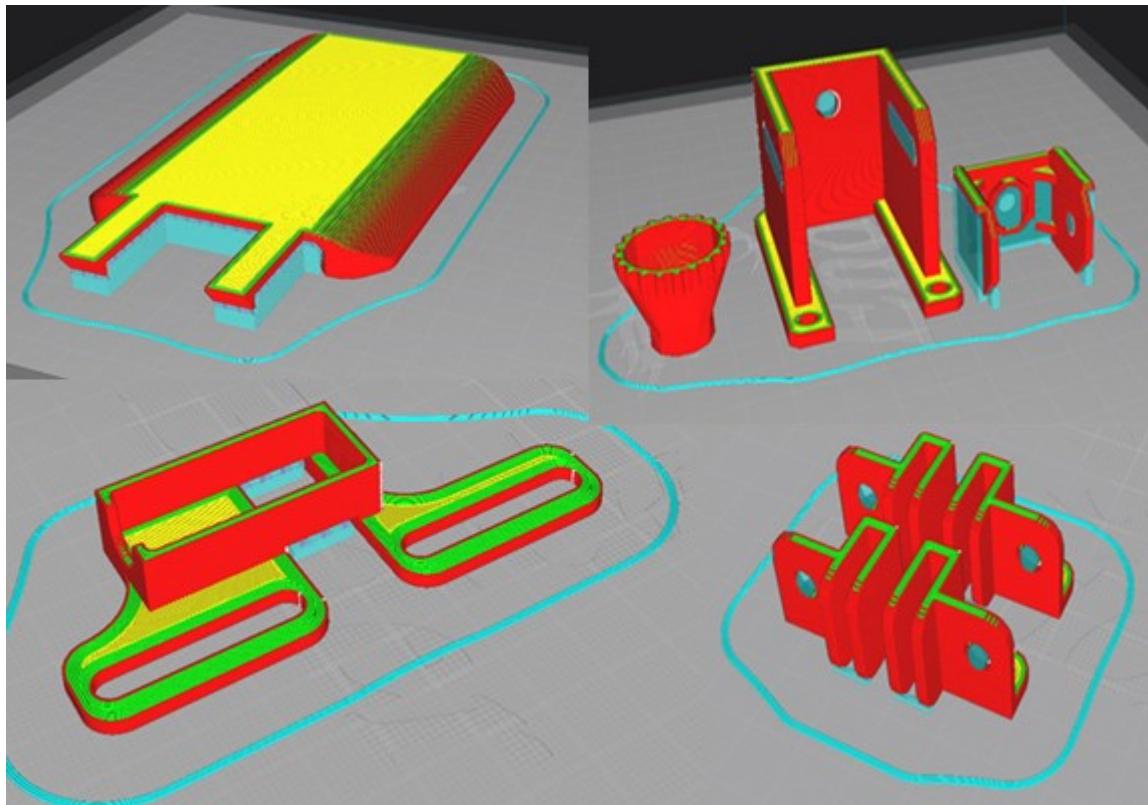
Calidad:	Altura de capa: = 0.2mm
Paredes:	Recuento de líneas de pared: =3
Relleno:	Densidad de relleno: = 60%
Material:	Temperatura de impresión: = 200°C (depende del filamento)
Velocidad:	Velocidad de impresión: = 50mm/s
Soporte:	Generar soporte: = Habilitado
Tipo de soporte:	Normal

las figuras 3-2 y 3-3 mostrarán la forma del despiece como también la forma y posición cual se dispuso para poder realizar la impresión 3D.



Fuente: Creación Propia

Figura 3-2 Elemento de compresión móvil en laminador, posición lista para impresión 3D.



Fuente: Creación Propia

Figura 3-3 Disposiciones de elementos de tensor de correa listos para impresión en 3D.

2.2 CONSTRUCCIÓN

En esta etapa se realizará el proceso de compra o adquisición de materiales necesarios para la construcción y desarrollo de este equipo.

Para ello se ha utilizado la plataforma de AliExpress o cualquiera de su misma similitud, ya que, este proyecto está pensado para poder ser elaborado en cualquier parte del mundo de bajo costo y globalizado, también se puede observar que la mayoría de los componentes es propósito general lo cual permite la modificación e integración según sea necesario para el usuario.

1.2.1. Elección de materiales.

Para la compra de materiales se escogió la plataforma de AliExpress entre otras, ya que posee, una gran variedad de vendedores y productos los cuales podemos cambiar de nuestra lista de compras.

Se ha dispuesto el siguiente link de la página, el cual se podrá adjunto en el carrito de compres los elementos necesarios para poder adquirirlos.

Esta dirección cargara el listado de forma automática, donde los precios serán referenciales a Agosto2024.

<https://www.aliexpress.com/p/shoppingcart/index.html?spm=a2g0o.detail.0.0.4d39fTHMfTHMxg>

De este link podemos observar la lista de materiales, en este caso se ha conformado la lista con fotografías y descripción como se observa en tabla 3.1.

Tabla 3.1. Materiales de estructura.

Ítem	Descripción	Detalle	Cant.	Fotografía
1	SC8UU	Rodamiento Lineal 8mm	4	
2	SK8	Soporte Guía Eje Lineal SK8	4	
3	Barra Redonda Acero inox 8 mm	Eje 8mm L 400mm Lineal Cromo Eje WCS Barra Redonda Barra De Acero Cilindro Linear Rail 3d impresora varilla	2	
4	LC-5KG-HX711	Celda De Carga 5 Kg con módulo HX711	2	

5	GT2/10/20 POLEA	Polea de sincronización GT2-10mm 20 dientes	1	
6	GT2/10/20-8mm	Polea fija GT2-10mm 20 dientes	1	
7	Nema 23HS5630A4	Motor paso a paso Nema 23 DE 2 fases, 57HS5630A4, 1,8 grados, 1,2nm/171oz.in, 3A, 56mm, para máquina CNC	1	
8	DM556	Controlador de Motor paso a paso de 2 fases, controlador de Motor paso a paso DM556S, 18-50VDC, 1,4-5.6A, para Nema17, Nema23, Nema34	1	
9	Correa Síncrona GT2-10mm	Correa de distribución abierta GT2 para impresora 3D, banda de goma de fibra de Armida	1	
10	IEC320 C14	LED IEC320 C14, interruptor basculante de latón de 250V, 10A C6, Conector de enchufe de entrada de fusible, montaje en ordenador	1	
11	S-60-24-3	Fuente switching de 24v 3 Amperes	1	
12	AD16-22DS Verde	Indicador de potencia LED montado en Panel, AD16-22DS, 22mm, 24V.	1	
13	AD16-22DS Rojo	Indicador de potencia LED montado en Panel, AD16-22DS, 22mm, 24V.	1	

14	AD16-22DS Amarillo	Indicador de potencia LED montado en Panel, AD16-22DS, 22mm, 24V.	1	
15	XB2-BA42 22mm Rojo	Interruptor de botón de autoreinicio, 1 piezas, cabeza de seta pequeña, parada de emergencia, perilla de 22mm	1	
16	XB2-BA31 22mm Verde	Interruptor de botón de autoreinicio, 1 piezas, cabeza de seta pequeña, parada de emergencia, perilla de 22mm	1	
17	Teclado membrana 1x4	Teclado de interruptor de membrana para coche inteligente arduino, matriz de 1x4	1	
18	Pantalla LCD 20x4 Vede	Módulo adaptador de interfaz Serial para Arduino, LCD2004	1	
19	Modulo PCF8574	Placa de interfaz serie IIC I2C TWI SPI, Puerto 1602, 2004,	1	
20	Perno acero inox 4mm*20mm	Tornillo redondo de acero inoxidable Perno Allen de cabeza hexagonal de M4 (10 Uni)	1	

3.2.2. Elección de componentes electrónicos.

Dentro de las elecciones de materiales realizadas se incluye la confección de una PCB la cual se utilizará para integrar todos los periféricos necesarios para funcionar, esta tarjeta debe ser robusta, para ello se ha realizado un lista de materiales en la página de

LCSC Electronics en el siguiente link, enviara al carrito de compra con los materiales necesarios para la confección de este proyecto.

<https://www.lcsc.com/personalCenter/myProject/11B6443E646EA5948EE7286FFF892>

BB4

LSCS Electronics es una fábrica ubicada en China, en la localidad Shenzhen, se especializa en el almacenaje de todos los componentes electrónicos actuales y es uno de los mayores abastecedores de suministro electrónico en el mundo.

En la tabla 3.2 identifica todos los materiales necesario para la confección de este proyecto.

Tabla 3.2. Componentes electrónicos para PCB.

Código LSCS	Descripción	General	Detalle	Cantidad
C2461	Diodo rectificador	6A10	6A 950mV@6A 1kV R-6 Diodes - General Purpose ROHS	1
C8545	Mosfet	2N7002	60V 115mA 200mW 7.50@10V,500mA 2.5V@250uA N Channel SOT-23 MOSFETs ROHS	15
C8678	Diodo Schottky	SS34	40V Single 3A 0.55V@3A SMA(DO-214AC) Schottky Barrier Diodes (SBD)	1
C96006	Foto transistor	PC817X3NSZ9F	80V 5000V 50mA 0.1V@1mA,20mA 1 6V 1.2V DC DIP-4 Optocouplers -	10
C181844	Bobina fuente Switching	SLH1207S330MTT	3A 33uH ±20% SMD Power Inductors ROHS	1

C75781	Res 1K 0805 1%	TC0525F1001T5E	250mW Thin Film Resistor 150V ±25ppm/? ±1% 10000 0805 Chip Resistor	24
C723424	Res 100k 0805 1%	RC0805FR-7W100KL	250mW Thick Film Resistors ±100ppm/? ±1% 10kΩ 0805 Chip Resistor - Surface Mount ROHS	19
C855098	Res 10k 0805 1%	AC0805FR-7W10KL	250mW Thick Film Resistors ±100ppm/? ±1% 10kΩ 0805 Chip Resistor - Surface Mount ROHS	2
C2073635	Res 5k 0805 1%	PLT0805Z5001AST5	250mW Thin Film Resistor 100V ±5ppm/□ ±0.05% 5kΩ 0805 Chip Resistor - Surface Mount	2
C125977	Capacitor Aluminio 220uf 50v	UWT1H221MNL1GS	220uF 50V 220mA@120Hz ±20% SMD,D10xL10mm Aluminum Electrolytic Capacitors - SMD ROHS	1
C7461236	Capacitor Aluminio 680uf 50v	126RV0105	680uF 35V 380mA ±20% SMD,D10xL10.5mm Aluminum Electrolytic Capacitors - SMD ROHS	1
C2936128	fusible 5x20	0218010.HXP	Fuse ROHS	1
C377342	Holder Fusible	HF-004/J	Fuse Holders ROHS	2
C2897378	Heder 15P	PM254-1-15-Z-8.5	2.54mm Plugin 3A 15P - 40□~+105□ Straight 1x15P 8.5mm Top	2

			Square Holes Plugin,P=2.54mm Female Headers ROHS	
C40877	Heder 6P	Plugin,P=2.54mm	2.54mm Plugin 6P Straight 1x6P Plugin,P=2.54mm Female Headers ROHS	2
C2718488	Heder 4P	Plugin,P=2.54mm	2.54mm Plugin 3A 4P bronze -40°C~+105°C Straight 1x4P Top Square Holes Plugin,P=2.54mm Female Headers ROHS	2
C557646	Terminal Block 2P	XY126V-5.0-2P	1x2P 10A 300V Green 14~26 Straight 5mm 2.5 1 2 Plugin,P=5mm Screw terminal ROHS	15
C5188441	Terminal Block 3P	MX127-5.0-03P-GN01-Cu-Y-A	Copper Alloy 1x3P Tin 5mm 1.5 1 3 10A - 40°C~+105°C 300V Green 14~26 Straight Plugin,P=5mm Screw terminal ROHS	1

2.3 CONSTRUCCIÓN ELECTRÓNICA.

En esta etapa se realiza el diseño electrónico de una PCB como también, la petición de fabricación de PCB como también el envío y posteriormente el ensamblaje de componentes, todo esto para lograr una placa que pueda utilizar la lógica de las funciones ya programadas.

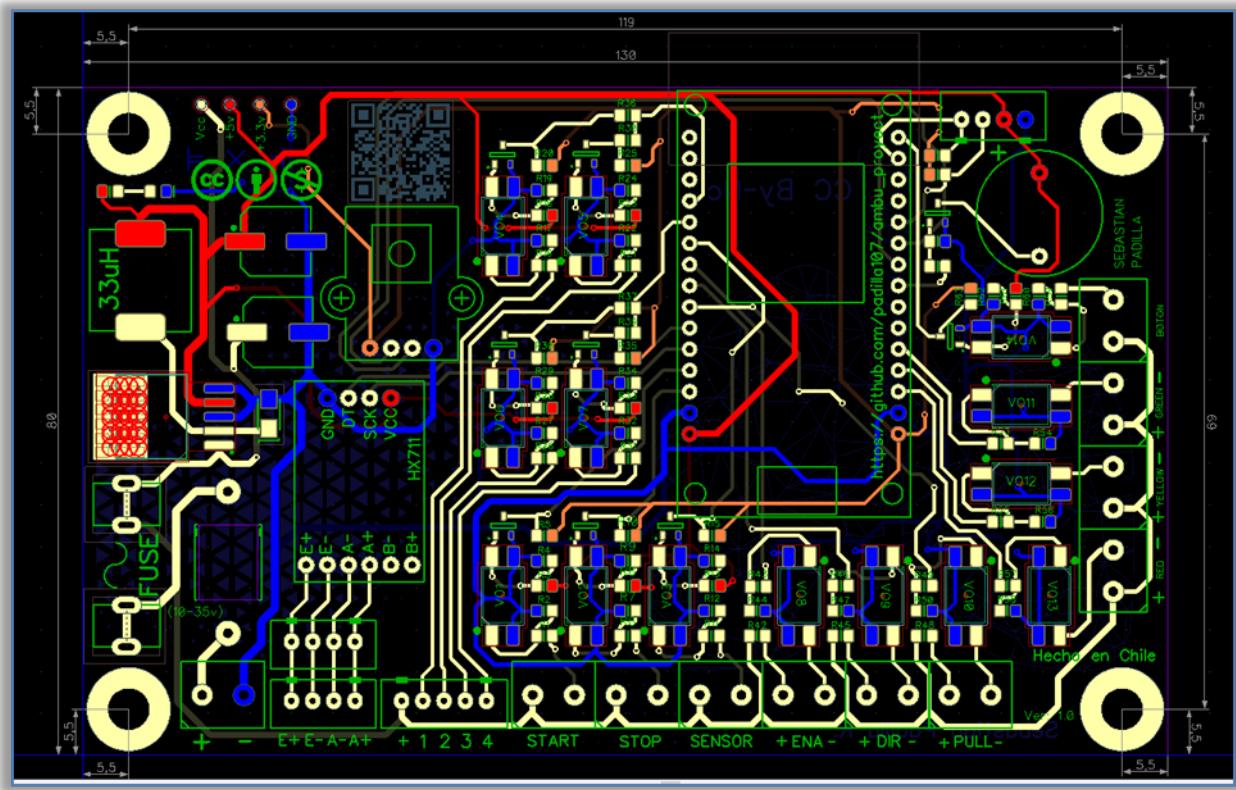
Para el diseño se utilizó DIPTRACE como entorno de desarrollo, y esquemático de forma asistida.

3.3.1. Confección de esquemático y distribución PCB

DIPTRACE es un software especializado en el diseño de circuitos impresos y esquemas electrónicos. Su interfaz intuitiva y sus potentes características permiten diseñar placas complejas con precisión. En esta fase, se emplean tanto el editor de esquemas como el editor de PCB de DIPTRACE para garantizar que el diseño sea funcional y cumpla con las especificaciones requeridas.

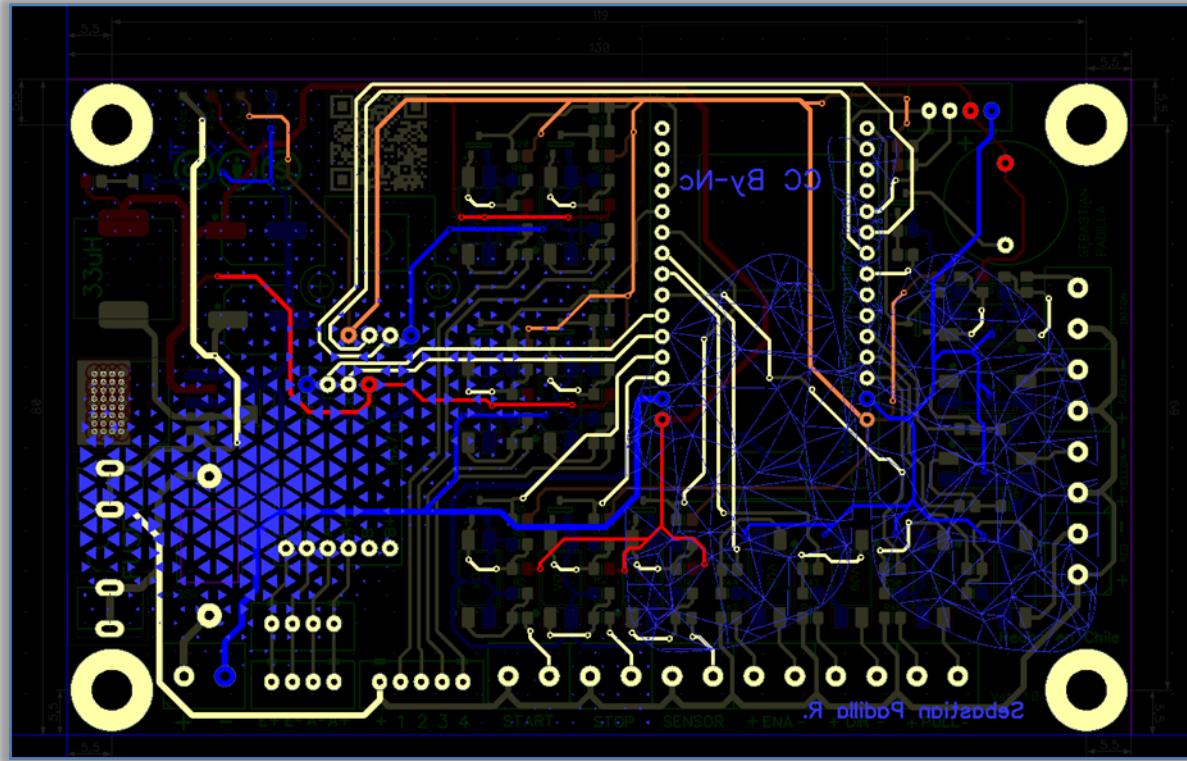
Una vez completado el diseño, se genera el archivo Gerber necesario para la fabricación de la PCB. Este archivo contiene toda la información necesaria para que el fabricante pueda producir la placa. La solicitud de fabricación se realiza en una empresa especializada que se encargará de producir las placas según el diseño enviado.

En las figuras 3-4 y 3-5, muestra el diseño y conexiones por componentes, en la vista superior o inferior.



Fuente: Creación Propia, en DipTrace.

Figura 3-4 captura de pantalla de circuito de electrónico PCB conexionado lado superior.



Fuente: Creación Propia, en DipTrace.

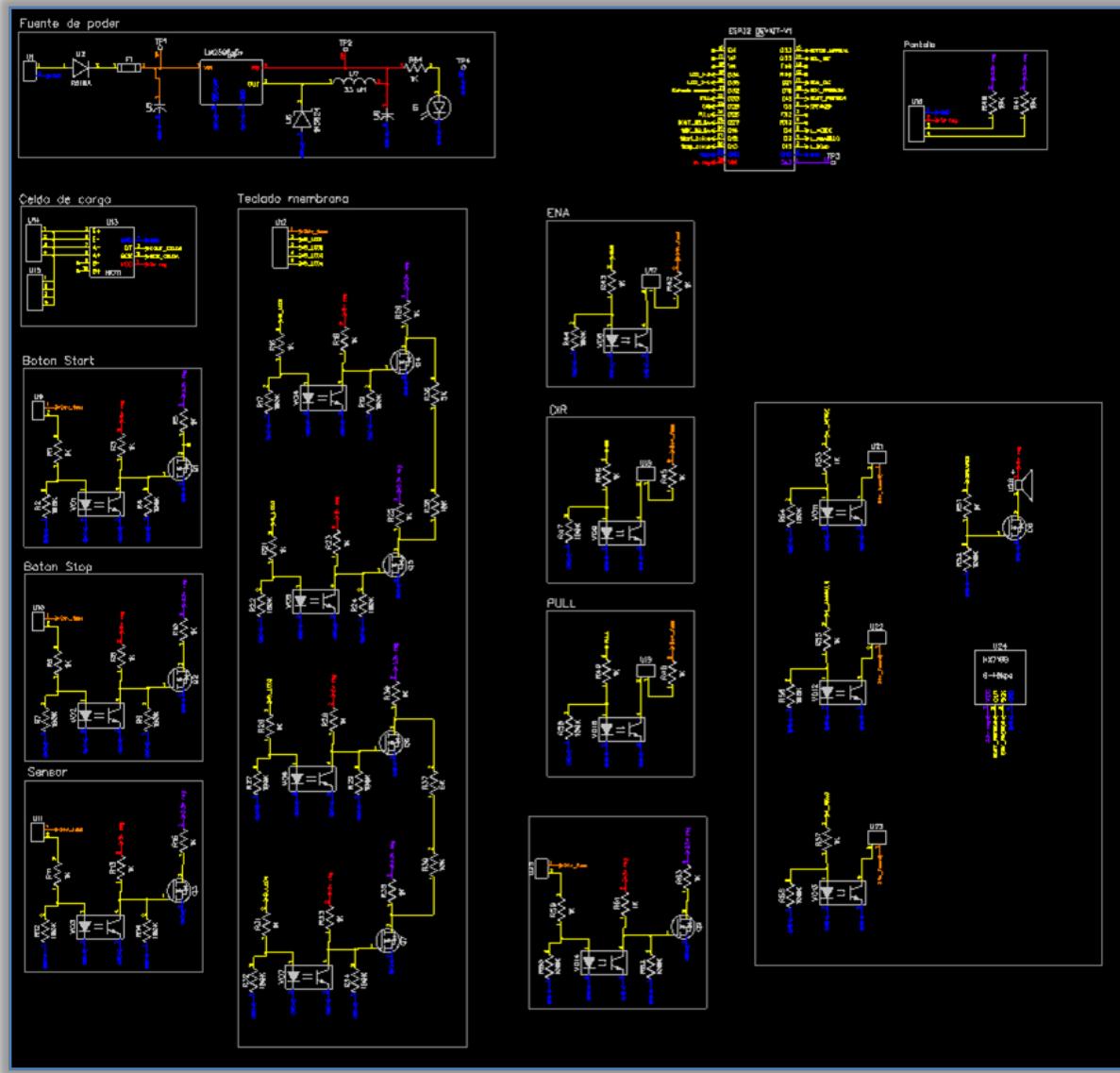
Figura 3-5 captura de pantalla de circuito de electrónico PCB conexionado lado posterior.

En la imagen 3-6 podemos ver el diagrama esquemático el cual nos servirá para guiarnos en la conexión entre componentes.

Este diagrama se observa que sus textos están escrito de forma incorrecta, ya que están volteados, esto es porque la imagen esta en vista atravesando la PCB, por esta razón los textos se encuentran volteados.

El diagrama esquemático es el mismo que se utilizo en los capítulos anteriores, solo que en este paso se destina a otro software donde se analiza las conexiones y se interconecta entre si, todo en una PCB.

Todo esto brinda una conexión entre el esquema técnico y los componentes, ya que, los componentes pueden variar según su encapsulado.



Fuente: Creación Propia, en DipTrace.

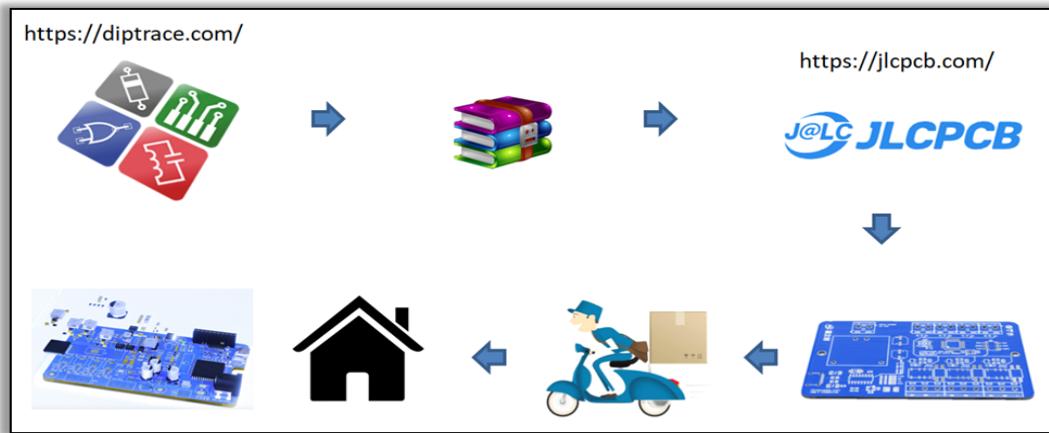
Figura 3-6 Captura de pantalla de esquemático circuito.

3.3.2. Fabricación de PCB.

Para la fabricación de la placa electrónica, se ha optado por confiar en la producción profesional de PCBs. Existen numerosas opciones en el mercado que ofrecen alta calidad, y en este caso, se ha elegido a JLCPCB debido a su excelente relación entre precio y calidad.

Esta empresa es conocida por su fiabilidad y por ofrecer un servicio de fabricación de placas de circuito impreso que cumple con los estándares profesionales a precios competitivos.

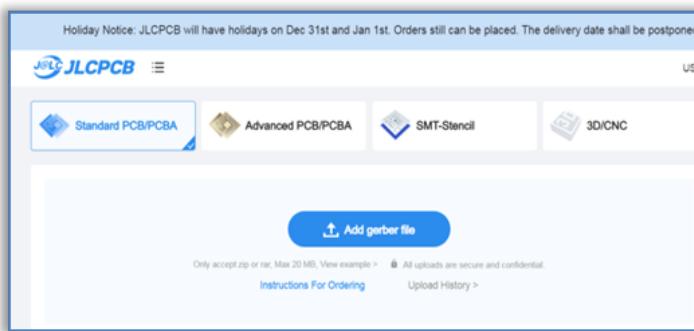
La figura 3-7 ejemplifica el flujo de procesos para enviar a fabricar la PCB en cuestión, este proceso es válido para cualquier fabrica de PCB, en nuestro caso JLCPCB.



Fuente: Creación Propia.

Figura 3-7 Ejemplo de distribución de procesos de confección de PCB.

Dentro de la página de JLCPCB desde el siguiente link: www.jlpcb.com encontraremos un apartado donde cargaremos nuestro gerber de la pcb que se realizara, realizamos la búsqueda del archivo en nuestro equipo tal como muestra en la figura 3-8.



Fuente: www.jlpcb.com.

Figura 3-8 Captura de pantalla de inicio dentro de la página.

Una vez que el archivo se haya cargado, aparecerán los parámetros de configuración necesarios para la confección básica del PCB (Placa de Circuito Impreso). Estos parámetros estarán ajustados de acuerdo con los requisitos estándar iniciales, pero pueden modificarse según las especificaciones particulares del usuario. La interfaz permite adaptar los parámetros a las necesidades específicas del diseño, asegurando una configuración personalizada y adecuada para cada aplicación.

Los parámetros utilizados en estos proyectos son los siguientes:

Base Material=FR4

Product Type=Industrial/Consumer electronics.

PCB Thickness= 1,6.

Silkscreen=White.

Surface Finish=HASL.

Outer Copper Weight=1 oz.

Confirm Production file>No.

Mark on PCB=Order Number.

Electrical Test=Flying Probe Full Test.

Gold Fingers>No.

Castellated Holes>No.

Edge Plating>No.

Al termino de la configuración daremos siguiente, esto no lleva la ventana de envío y proyecto, se selecciona nuestro proyecto y volvemos a dar siguiente, esto nos lleva a la pantalla de costo de fabricación y envío.

Ingresamos la dirección junto con el importe, luego especificamos el operador para el envío y lo declaramos en aduana o en correos de chile.

3.3.3. Ensamble de componentes.

Una vez que la PCB ha llegado desde JLCPCB, se deberán seguir los siguientes pasos para soldar los componentes SMD de manera correcta y eficiente:

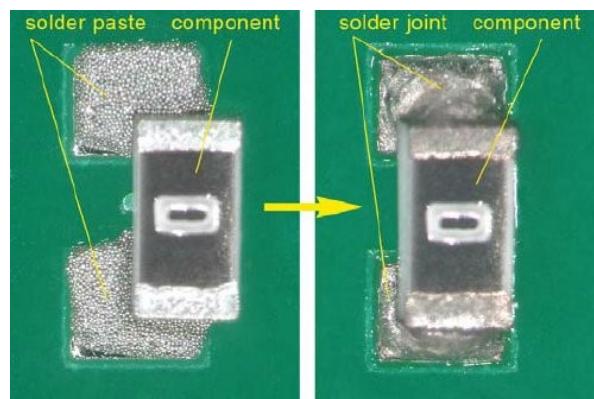
Se debe utilizar una jeringa de pasta de soldadura para aplicar una pequeña cantidad de pasta en las almohadillas donde se montarán los componentes SMD. La pasta debe aplicarse de manera uniforme y únicamente en las áreas correspondientes para evitar cortocircuitos.

La estación de soldadura debe configurarse a la temperatura adecuada (generalmente entre 250°C y 350°C). Se recomienda usar una punta fina para una mayor precisión.

Se recomienda usar una soldadura al 60% con un diámetro de 0.5mm, también para se debe utilizar un extractor de aire o en su defecto soldadura libre de plomo Pb Free.

Se debe iniciar con los compontes más pequeños y finalizar en los grandes, esta práctica es para la movilidad de uso con cautín y espacio al soldar.

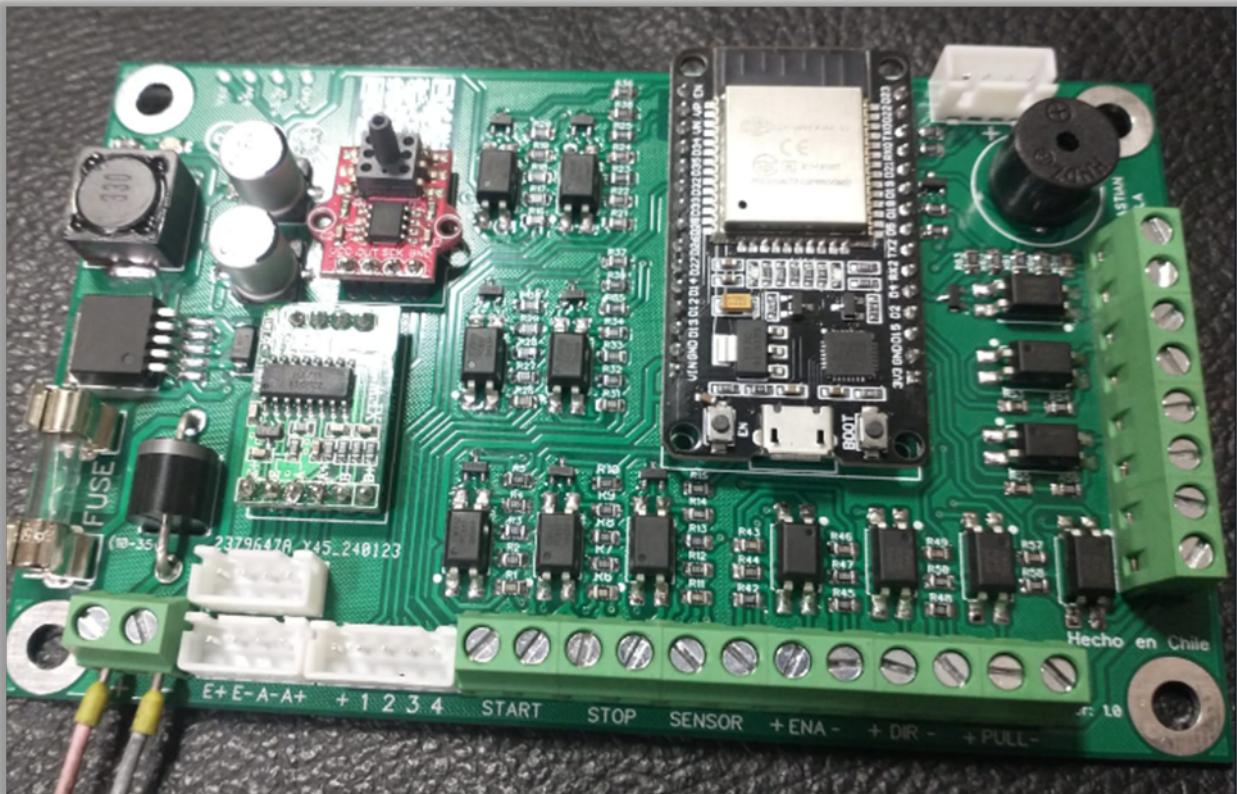
Si se utiliza soldadura en pasta se pueden acomodar y cuando se realiza la exposición por aire caliente el componente se acomoda solo, tal como muestra en la figura 3-9.



Fuente: <https://resources.altium.com/es/p/common-smt-process-defects-to-avoid-during-soldering>.

Figura 3-9 Fotografía N°1 de PCB ensamblada, lista para montaje y pruebas.

Una vez que todas las conexiones estén verificadas, se deberá encender el circuito y realizar pruebas funcionales para asegurarse de que todos los componentes SMD estén correctamente soldados y funcionando según lo previsto.



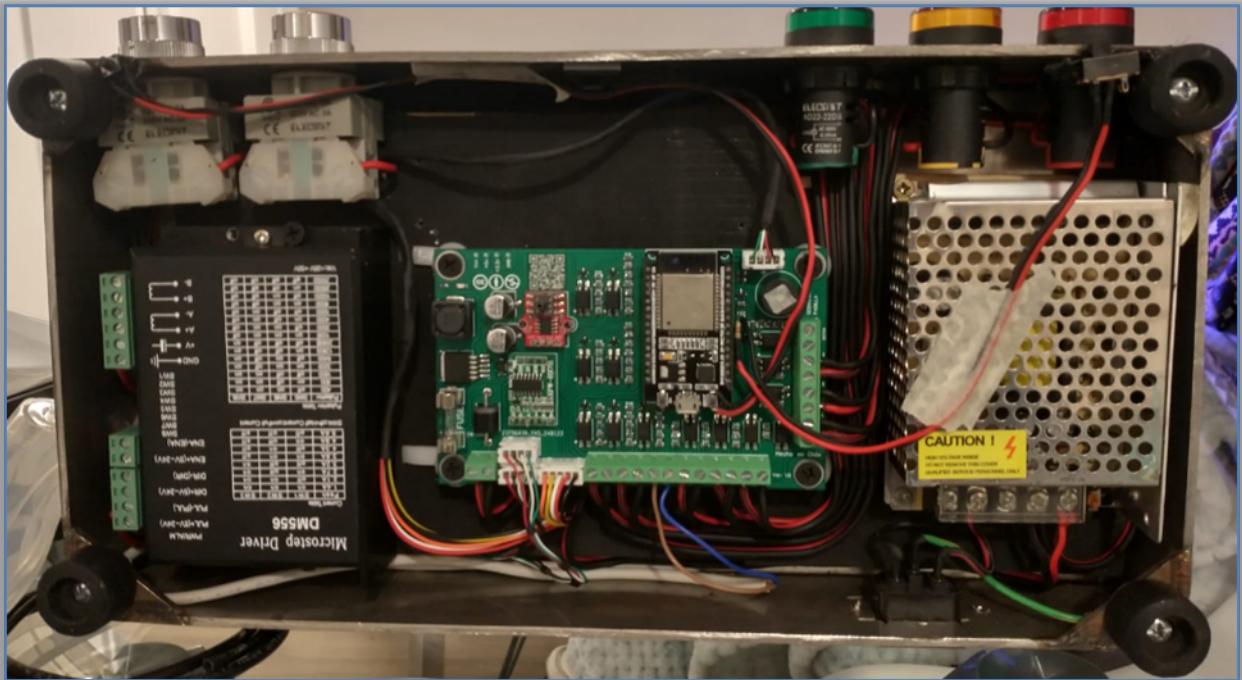
Fuente: Creación Propia.

Figura 3-10 Fotografía N°1 de PCB ensamblada, lista para montaje y pruebas.

La figura 3-10, muestra el ensamble terminado donde veremos el microcontrolador junto con todos los componentes fijados adecuadamente.

De realiza pruebas de encendido donde no tenemos recalentamientos ni corto circuitos.

Después de realizarla pruebas, debemos montar de elemento, podemos ver en la figura 3-50, donde veremos la instalación de la PCB junto con la fuente switching al igual que el controlador de potencia del motor paso a paso



Fuente: Creación Propia.

Figura 3-11 Fotografía de PCB montada y conectada.

3.4 PROGRAMACIÓN.

La programación de micro controlador es una habilidad fundamental en el diseño y desarrollo de sistemas embebidos y dispositivos electrónicos. Un microcontrolador es un pequeño ordenador en un solo chip que puede ser programado para realizar tareas específicas, desde controlar dispositivos simples hasta gestionar sistemas complejos, para ello se ha especificado el IDE de VS CODE para esta tarea esencial.

3.4.1. Instalación de Visual Studio Code.

Para instalar Visual Studio Code se debe descargar el instalador desde el sitio web oficial de VS code (<https://code.visualstudio.com/>), se debe ejecutar el instalador esto cambiara según el equipo.

3.4.2. Configuración de entorno.

Configurar entorno Vs code: al Inicia el editor después de la instalación tendremos que cargar distintas extensiones lo cual resultara amigable los paso siguientes de verificación y cargar de nuestro proyecto hacia el micro controlador, las extensiones serán las siguientes: C/C++ , Spanish language pack for visual studio code y platformIO.

3.4.3. Cargar proyecto al entorno.

Para cargar el proyecto debemos bajar la de repositorio del siguiente link:

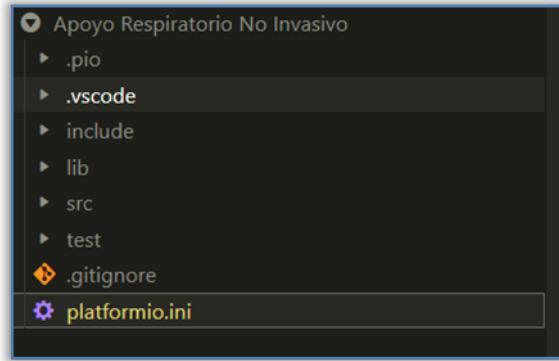
https://github.com/padilla107/ambu_proyect.git

Desde el menú se debe dirigir a la pestaña superior e ir a Archivo > Abrir Carpeta y selecciona la carpeta que contiene el proyecto que deseas cargar.

Luego debe seleccionar el proyecto competo, esto permitirá grabar las dependencias y librerías necesarias para la programación

Se debe destacar que el proyecto ya posee las librerías del proyecto con lo cual no se tendrá problema a la hora de cargar el código al nuestro microcontrolador.

Dentro de la plataforma y previamente cargada todas las extensiones necesarias, debemos dirigirnos al costado izquierdo que encontraremos el explorador, seleccionamos nuestro proyecto y dentro del encontraremos una carpeta llamada “platformIO.ini” como indica en la figura 3-56.



Fuente: Visual Studio Code

Figura 3-12 Captura de pantalla de proyecto en Visual Studio Code.

Al seleccionar el archivo “platformIO” se desplegara una ventana al costado donde veremos las configuraciones como también las librerías de dependencias, donde encontraremos las librerías que se esta utilizando, como indica en la siguiente imagen.

```

1 ; PlatformIO Project Configuration File
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
15 monitor_speed = 115200
16 lib_deps =
17     marcoschwartz/LiquidCrystal_I2C@^1.1.4
18     bogde/HX711@^0.7.5
19     thomasfredericks/Chrono@^1.2.0
20     Wire
21     waspinator/AccelStepper@^1.64

```

Fuente: Visual Studio Code

Figura 3-13 Captura de pantalla de proyecto, en carpeta de platformio.ini en Visual Studio Code.

En la figura 3-13, especifica el tipo de placa que estamos utilizando al igual que plataforma y velocidad de trasmision en monitoreo.

Si quiere saber las librerías que se utilizaron podemos obtenerlas de forma extena en los siguientes enlaces.

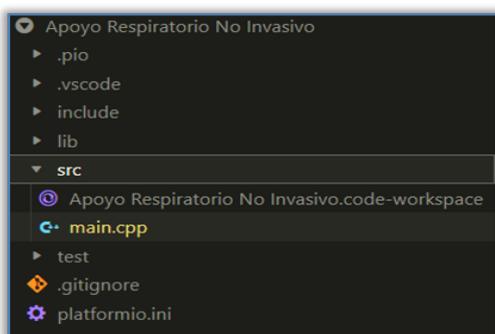
https://github.com/marcoschwartz/LiquidCrystal_I2C.git

Incluye la librería "WIRE".

<https://github.com/bogde/HX711.git>

<https://github.com/Thomasfredicks/Chrono.git>

<https://github.com/waspinator/AccelStepper.git>



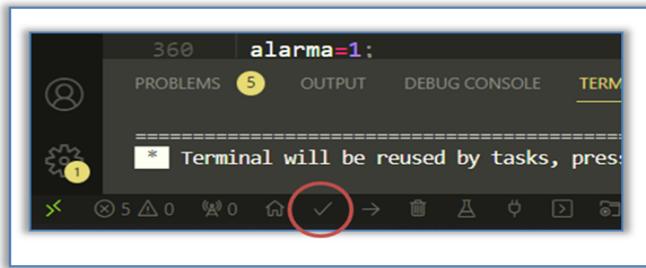
Fuente: Visual Studio Code

Figura 3-14 Captura de pantalla de proyecto, accediendo a la fuente en Visual Studio Code.

La figura 3-14 muestra el archivo raíz de programación de la carpeta.

3.4.4. Preparación y verificación de proyecto.

Una vez cargado el proyecto, podremos ver un menú en la parte inferior izquierda de la ventana de la aplicación, en este sector encontraremos un ícono con signo de visto al cual debemos dar inicio, como muestra en la figura 3-60.



Fuente: Visual Studio Code

Figura 3-15 Captura de pantalla de proyecto, ícono de verificación en Visual Studio Code.

Al llevar a cabo esta acción, el sistema realizará una verificación del estado y el funcionamiento del dispositivo antes de proceder con la grabación en el micro controlador. Durante este proceso, se comprobarán todos los parámetros necesarios para asegurar que todo esté en orden. Si todas las verificaciones se completan con éxito y no se detectan problemas, en la terminal se mostrará un mensaje de “SUCCESS”, esto confirmara que el procedimiento se ha realizado correctamente. Como se muestra en la figura 3-16.

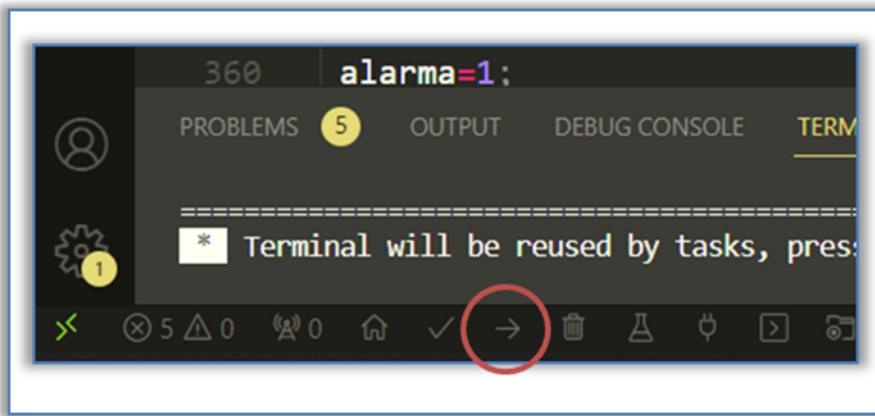
```
RAM: [=           ] 7.0% (used 23088 bytes from 327680 bytes)
Flash: [==          ] 22.7% (used 298105 bytes from 1310720 bytes)
===== [SUCCESS] Took 3.54 seconds =====
* Terminal will be reused by tasks, press any key to close it.
```

Fuente: Visual Studio Code

Figura 3-16 Captura de pantalla de proyecto, estatus de verificación correcta en Visual Studio Code.

3.4.5. Grabado de proyecto al microcontrolador.

Una vez completada la verificación exitosa del código, el siguiente paso es cargarlo en el micro controlador. Para ello, primero conectaremos el micro controlador utilizando un cable USB tipo A a micro USB tipo B. A continuación, nos dirigiremos a la parte inferior izquierda de la pantalla, donde encontraremos un ícono con una flecha apuntando hacia la derecha. Este ícono representa la opción para realizar el proceso de carga, denominado [UPLOAD], hacia el micro controlador. Como representa en la figura 3-17.



Fuente: Visual Studio Code

Figura 3-17 Captura de pantalla de proyecto, ícono de Upload hacia el micro controlador en Visual Studio Code.

En la terminal, observaremos que los datos comienzan a compilarse para su grabación. Cuando aparezca el mensaje “CONNECTING...”, debemos presionar el botón de BOOT durante 3 segundos en nuestro micro controlador. Esto iniciará el proceso de carga del código hacia el micro controlador. Como indica la figura 3-18.



Fuente: Visual Studio Code

Figura 3-18 Captura de pantalla de aviso en la terminal acción grafica de usuario.

Después de realizar la acción de BOOT en el micro controlador, la terminal iniciará el proceso de carga del archivo. Al finalizar este proceso, se mostrará un mensaje de “SUCCESS”, como se ilustra la figura 3-19.

```
Hash of data verified.
Compressed 298464 bytes to 167455...
Writing at 0x00010000... (9 %)
Writing at 0x0001b9d5... (18 %)
Writing at 0x0002465b... (27 %)
Writing at 0x00029db0... (36 %)
Writing at 0x0002f16c... (45 %)
Writing at 0x00034673... (54 %)
Writing at 0x00039e13... (63 %)
Writing at 0x00043143... (72 %)
Writing at 0x0004cb0e... (81 %)
Writing at 0x000520d8... (90 %)
Writing at 0x0005799b... (100 %)
Wrote 298464 bytes (167455 compressed) at 0x00010000 in 4.2 seconds
61.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 12.18 seconds =====
* Terminal will be reused by tasks, press any key to close it.
```

Fuente: Visual Studio Code

Figura 3-19 Captura de pantalla de aviso en la terminal, donde se realizó correctamente el Upload.

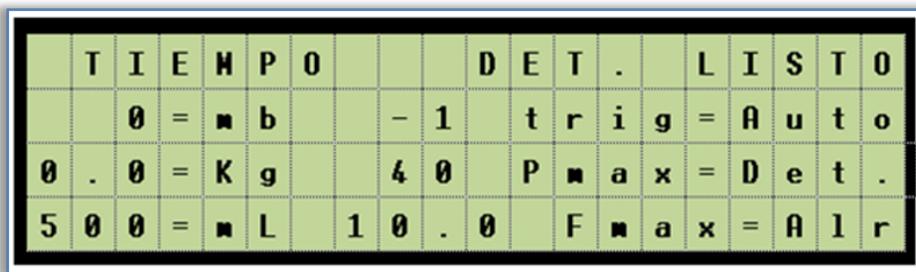
Una vez completado este paso, el programa estará grabado en el micro controlador tal, esto permitirá utilizarlo junto con la lógica que fue programada, asegurando que el sistema funcione según lo previsto.

3.5 DESCRIPCIÓN DE INTERFAZ Y FUNCIONALIDADES DEL SISTEMA.

En el diseño de dispositivos médicos complejos, la configuración e interfaz entre el usuario y la máquina son cruciales para garantizar un funcionamiento seguro y efectivo. En este contexto, la interfaz humano-máquina (IHM) se diseña para facilitar una interacción intuitiva y eficiente mediante una pantalla LCD de 20x4 caracteres, cuatro botones de configuración, dos botones de control y tres luces indicadoras.

3.5.1. Pantalla.

La pantalla LCD de 20x4 caracteres sirve como el principal medio visual para la interacción entre el usuario y el equipo. Esta pantalla permite la visualización clara y detallada de la información crítica del dispositivo, como los parámetros de configuración, el estado actual del respirador y mensajes de alerta como indica la figura 3-20. La capacidad de mostrar hasta 20 caracteres por línea en 4 líneas facilita la presentación de datos extensos y menús de configuración, haciendo que la información sea accesible y fácil de interpretar para el usuario.



Fuente: Creación Propia.

Figura 3-20 Representación grafica de pantalla principal en LCD.

3.5.2. Botones de Configuración.

Cuatro botones pequeños están integrados en el diseño para permitir la modificación y ajuste de los parámetros del respirador. Estos botones como se muestra en la figura 3-21, permiten al usuario navegar por los menús en la pantalla LCD, ajustar los valores de configuración como la frecuencia respiratoria, el volumen de aire y otros parámetros esenciales. La disposición de los botones ha sido cuidadosamente diseñada para ofrecer una respuesta táctil precisa y una navegación intuitiva, lo que facilita la configuración rápida y sencilla del dispositivo.



Fuente:<https://www.google.com/search?q=teclado+de+membrana+arduino&rlz=>.

Figura 3-21 Imagen de muestra de teclado para LCD.

3.5.3. Botones de operación.

El respirador cuenta con dos botones de control, uno para iniciar el funcionamiento del dispositivo y otro para detenerlo, tal como se muestra en la figura 3-22. Estos botones proporcionan una manera clara y directa de controlar el estado operativo del respirador, permitiendo al usuario activar o desactivar el dispositivo según sea necesario. La implementación de estos botones asegura que el usuario pueda manejar el respirador con facilidad durante situaciones críticas.



Fuente:<https://www.google.com/search?q=botones+rojo+verde&rlz=>.

Figura 3-22 Imagen de muestra de botones de acción.

3.5.4. Luces Indicadoras.

Tres luces indicadoras están presentes para proporcionar señales visuales rápidas sobre el estado de funcionamiento del equipo. Estas luces sirven para informar al usuario sobre el estado, la figura 3-23 muestra la distribución del elemento en cuestión.

Luz Verde: Indica que el respirador está funcionando correctamente.

Luz Amarilla: Señala que el dispositivo está en modo alarma.

Luz Roja: Alerta sobre un posible error o mal funcionamiento o detenido.



Fuente:<https://www.google.com/search?q=luces+verde+amarilla+roja&rlz=>.

Figura 3-23 Imagen de muestra de luces de estado de funcionamiento.

3.5.5. Interacción General.

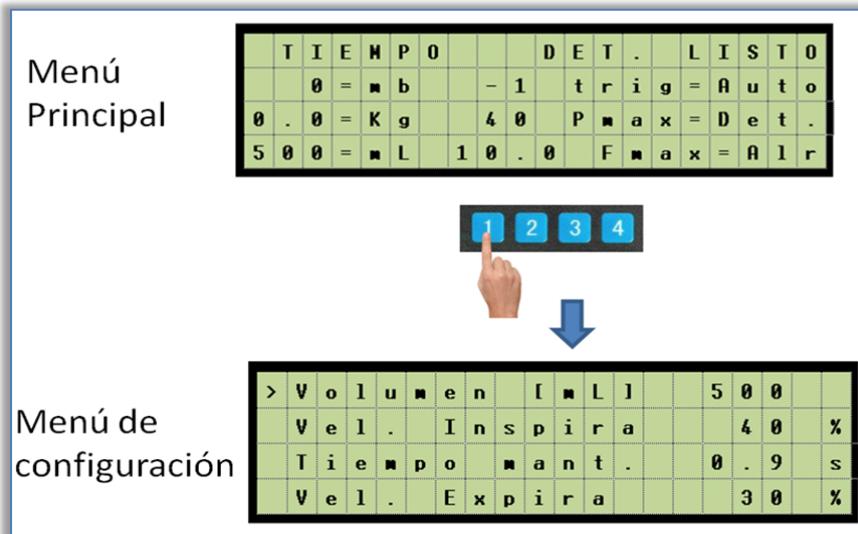
La combinación de la pantalla LCD, los botones de configuración, los botones de control y las luces indicadoras conforma una interfaz humano-máquina efectiva y eficiente. La pantalla LCD proporciona la información necesaria de manera clara y detallada mientras los botones permiten una configuración precisa y el control del respirador, por otro lado, las luces indicadoras ofrecen señales visuales rápidas sobre el estado del dispositivo.

Este diseño integral asegura que el usuario pueda operar el respirador artificial con confianza y precisión, facilitando la configuración adecuada y la monitorización del estado del dispositivo en situaciones críticas. La interfaz ha sido optimizada para proporcionar una experiencia de usuario intuitiva y accesible, fundamental para la seguridad y eficacia en el uso de equipos médicos.

3.5.6. Configuración

Al energizar el equipo, la pantalla LCD de 20x4 se encenderá y mostrará el menú principal, en este menú presenta de: Variables de Condición de Funcionamiento, Estado Actual, Ajustes de Presión y Fuerza, Trigger Y Variables Físicas Actuales.

Luego al presionar cualquier botón de membrana los cuales identificaremos como botón 1-2-3-4, entraremos al menú de configuración como indica en la figura 3-24.



Fuente: Creación Propia.

Figura 3-24 Representación grafica de interacción entre menú.

3.5.7. Modificar parámetros

Dentro del menú de configuración, veremos un ícono en forma de flecha que indica la línea actual en la que nos encontramos. Utilizando los botones 1 y 2, podemos desplazarnos hacia arriba o hacia abajo a través de los 18 parámetros disponibles en el menú como muestra en la figura 3-26.

Para ajustar los valores de los parámetros seleccionados, emplearemos los botones de membrana 3 y 4, que permiten aumentar o reducir los valores según sea necesario.

Si no se realiza ninguna interacción en el menú durante 5 segundos, la pantalla cambiará automáticamente de vuelta al menú principal.

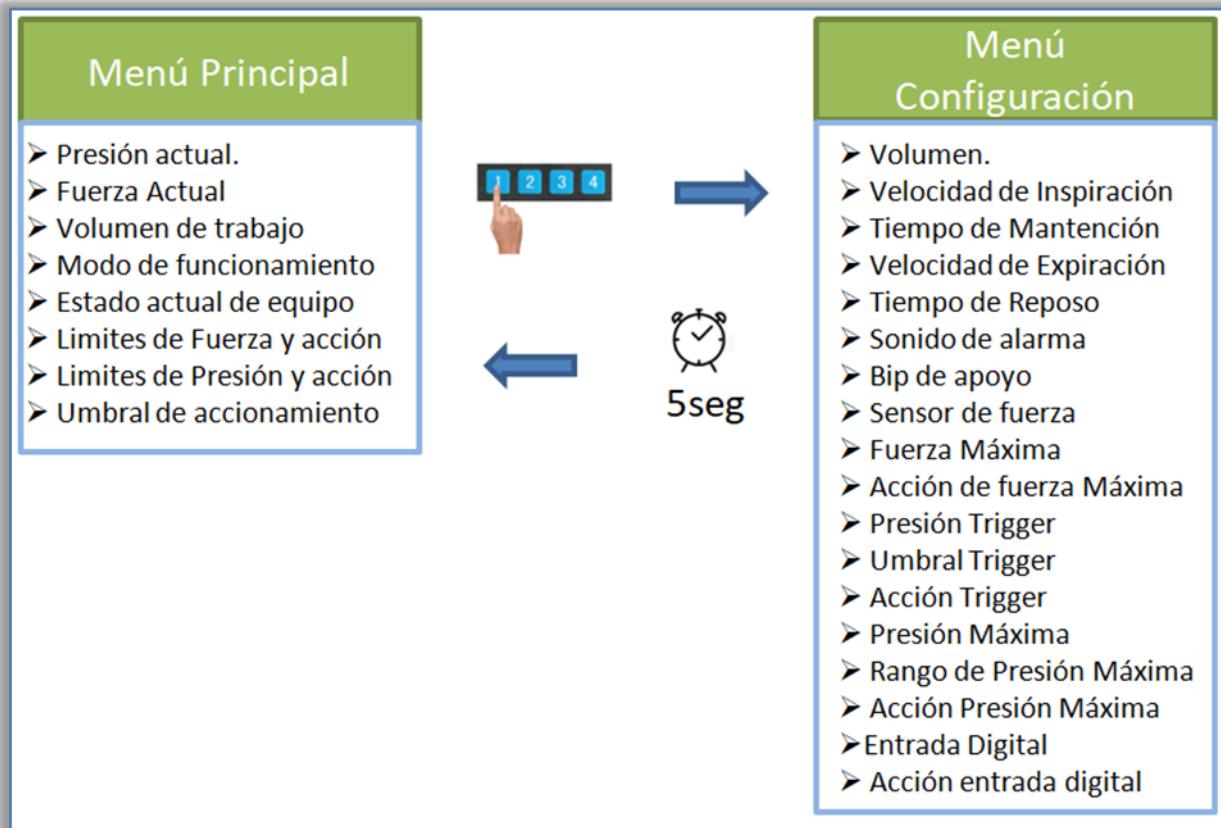
La figura 3-25, selecciona 3 elementos, donde la imagen superior es el elemento indicador de la línea que nos encontramos. El cuadro de el medio especifica la descripción del o titulo del parámetro que queremos modificar y el cuadro de abajo especifica el valor del parámetro que queremos cambiar, los cuales cambiaremos con los botones 3-4.



Fuente: Creación Propia.

Figura 3-25 Representación grafica de elementos en el menú de configuración.

La figura 3-26, grafica claramente la interacción entre los 18 parámetros disponibles y muestra cómo se navega entre la pantalla de configuración y la pantalla principal. Esta representación gráfica ayuda a comprender cómo se visualizan y ajustan los diferentes parámetros en el menú de configuración, así como la transición y el acceso a la pantalla principal del dispositivo.



Fuente: Creación Propia.

Figura 3-26 Representación grafica de desplazamientos y elementos de la pantalla principal y de configuración.

3.5.8. Descripción de parámetros.

En el apéndice N°1 de menú de configuración, presenta un desglose detallado de los 18 parámetros disponibles en el menú de configuración. Esta tabla ilustra el uso y la interacción de cada parámetro, proporcionando una visión clara de cómo se ajustan y configuran las distintas opciones. La información presentada facilita la comprensión de las funciones específicas de cada parámetro y su impacto en la operación del dispositivo.

3.5.9. Funciones Importantes.

Dentro de las configuraciones del equipo, se han incorporado varias funciones importantes que aumentan su versatilidad y seguridad:

Alarma de Fallo: El equipo está equipado con una alarma que se activa ante cualquier fallo del sensor o corte de correa, notificando inmediatamente sobre cambios o interrupciones en el funcionamiento del motor.

Interrupción de Funcionamiento: Si se supera el umbral de 12 kg de fuerza, el equipo interrumpe automáticamente su funcionamiento para prevenir daños o riesgos.

Memoria EEPROM: Todos los datos de configuración se almacenan en la memoria EEPROM cada vez que se sale del menú de configuración y se regresa al menú principal. Esto garantiza que los ajustes se conserven incluso después de apagarse el equipo.

Auto-inicio: En caso de corte de energía, el equipo realizará un auto inicio al reiniciarse, ajustándose a su estado anterior para asegurar un funcionamiento continuo, todo esto en caso de uso de equipos de generación auxiliar.

Ajustes Rápidos: En el menú de configuración, si se mantiene presionado el botón para aumentar o reducir el valor de un parámetro, el ajuste se realizará rápidamente, evitando la necesidad de pulsar repetidamente.

Atenuación de la Luz de Fondo: La luz de fondo de la pantalla LCD se apagará automáticamente después de un minuto de inactividad, ayudando a conservar energía y prolongar la vida útil de la pantalla.

3.6 VALORES Y COSTOS.

Los valores y costos son importantes para el desarrollo de este proyecto ya que podemos visualizar los elementos críticos como la viabilidad para la realización de tareas y adquisición de materiales para ello se han separado entre costos y valores.

Para este proyecto no se considera herramientas básicas, tampoco cables e impresora 3D con sus suministros.

3.6.1. Costos de materiales.

Para realizar el pedido o compra se ha dejado un enlace lo cual resultara más fácil la adquisición de cada ítem.

<https://www.aliexpress.com/p/shoppingcart/index.html?spm=a2g0o.detail.0.0.4d39fTHMfTHMxg>

El valor de compra es de \$155.579 correspondiente a los materiales de construcción de nuestro respirador, de un valor referencial de U\$165 dls. Correspondiente a Agosto 2024.

Estos valores corresponden a la pagina aliexpress donde especifica precios a la fecha antes mencionada.

3.6.2. Costos de componentes electrónicos.

El siguiente enlace muestra el pedido completo de los componentes electrónicos para el ensamblado de la placa electrónica PCB.

<https://www.lcsc.com/personalCenter/myProject/11B6443E646EA5948EE7286FFF892BB4>

El valor de los componentes electrónicos es de US85.43 dls correspondiente a \$75.154 peso chileno.

El valor de importe y fabricación de la PCB es de U\$66.46 dls correspondiente a \$58.466 peso chileno.

3.6.3. Carta Gantt de trabajos.

La carta Gantt de este proyecto contempla 13 semanas donde se utilizara recursos humanos, podemos ver la utilización de personal Mecánico, Electrónico como también la línea de supervisión.

Esta herramienta nos presenta un cronograma de manera visual, de la función de cada trabajador, como se muestra en la figura 3-27.

Se puede observar que el mayor trabajo se realizara por parte del ingeniero a cargo y el electrónico.

Item	Actividades	Responsables	Desarrollo de sistema de ventilacion No invasiva												
			Semanas												
			S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
1	Planteamiento de problemas y generar soluciones	Ingeniero Supervisor	X												
2	Realizar diseño de proyecto	Ingeniero Supervisor		X	X										
3	Contactar a proveedores de materiales y equipos	Ingeniero Supervisor				X									
4	Realizar estructura de respirador	Mecanico Soldador					X								
5	Realizar Impresión 3D de elementos	Ingeniero Supervisor					X								
6	Realizar montaje de materiales en estructura	Electronico						X							
7	Realizar montaje de componentes electronicos en PCB	Electronico						X							
8	Realizar conexionado	Electronico							X						
9	Realizar programación	Ingeniero Supervisor							X	X					
10	Realizar pruebas de funcionamiento	Ingeniero Supervisor								X	X				
11	Realizar backup de archivos en nube	Ingeniero Supervisor									x				
12	Realizar capacitacion al personal	Ingeniero Supervisor										x			
13	Entrega de proyecto	Ingeniero Supervisor											x		

Fuente: Creación Propia.

Figura 3-27 Captura de pantalla de tabla de carta Gantt.

3.6.4. Valores de HH Especialistas.

La figura 3-85, especifica los valores de horas hombre, según el criterio de tiempos establecidos en la carta Gantt, estos valores fueron extraido en la bolsa del trabajo donde podemos obtener mayor un valor promediado según especialista en la materia.

Item	Profesional	Dias	HH Por dia	Valor Unit x HH (UF)	Valor (UF) Unit. X Dia	Costo por profesional	
						Valor Total (UF)	Valor en pesos (\$)
1	Ingeniero Supervisor	55	8	0,4	3,2	176	\$6.472.015
2	Tecnico Electronico e instrumentista	15	8	0,3	2,4	36	\$1.323.821
3	Tecnico Mecanico Soldador	5	8	0,3	2,4	12	\$441.274
4	Ayudante Tecnico Mecanico Solador	5	8	0,15	1,2	6	\$220.637
						230	\$8.457.746
				UF	PESOS		
				1	\$36.773		

Fuente: Creación Propia.

Figura 3-28 Captura de pantalla de tabla de gastos valores de HH y especialistas.

Evento	Peso chileno	Dolares	Monto
Compra de materiales	\$155.579		\$155.579
Fabricacion PCB+Envio		\$66	\$58.466
Componentes PCB		\$85	\$75.154
Fabricacion Base	\$30.000		\$30.000
U\$=879,71		Total	\$319.198

Fuente: Creación Propia.

Figura 3-29 Captura de pantalla de tabla de gastos de los materiales del proyecto.

3.6.5. Costo total

El costo total se compone de la combinación de diferentes costos y valores como son los materiales, componentes, fabricación.

Entre los valores podemos ver el valor de especialista según el tiempo de duración del la confección del proyecto, en la figura 3-86 podemos ver el costo total del proyecto.

CONCLUSIÓN

El desarrollo del Sistema de Apoyo para la Respiración No Invasivo representa un avance significativo en el campo de la asistencia respiratoria, abordando una necesidad tangible en el ámbito médico. A lo largo de este proyecto, se ha logrado crear una solución integral que abarca desde el diseño 3D hasta la programación de microcontroladores, alcanzando la confección de la maquinaria funcional y la creación de tarjeta PCB. Este dispositivo, que emula las funciones de un respirador artificial pero de manera no invasiva, demuestra un potencial considerable para mejorar la calidad de vida de los pacientes con dificultades respiratorias.

La integración de estas diversas etapas del proceso de desarrollo no solo ha permitido la creación de una herramienta funcional y efectiva, sino que también ha establecido una base sólida para futuras innovaciones en el área. La capacidad de personalizar y ajustar el dispositivo según las necesidades específicas de los pacientes es un aspecto fundamental que aporta gran valor a su aplicación clínica junto a su configuración intuitiva.

El repositorio creado para este proyecto facilita el acceso a la información técnica y los recursos necesarios para replicar y adaptar la tecnología. Este esfuerzo por compartir el conocimiento y las herramientas disponibles contribuye al avance en el diseño de dispositivos médicos y fomenta la colaboración entre profesionales e investigadores en el campo.

A pesar de los logros alcanzados, es esencial reconocer las limitaciones y áreas de mejora. La adaptación y optimización continua del dispositivo. Además, la exploración de nuevas tecnologías y materiales puede llevar a mejoras en el rendimiento y la funcionalidad del equipo.

En resumen, este proyecto no solo ha cumplido con los objetivos iniciales, sino que también ha abierto nuevas vías para la investigación y el desarrollo en la asistencia respiratoria. La combinación de diseño innovador, ingeniería precisa y un enfoque orientado al usuario establece un precedente para futuras iniciativas en el campo, ofreciendo una solución prometedora para mejorar la vida de aquellos que enfrentan desafíos respiratorios.

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas que han sido fundamentales en la realización de este proyecto de tesis llamado “Sistema de apoyo para la respiración no invasiva”.

Como ya saben, este trabajo no ha sido fácil, he estado lleno de altos y bajos. El cambio de casa, el cambio de trabajo y el constante estrés han sido desafíos significativos que me han puesto a prueba durante todo este tiempo. Sin embargo, aquí estoy. Recuerdo cuando la idea de convertirme en ingeniero parecía un sueño lejano, no solo por obtener un título, sino por ser verdaderamente un profesional. Desde que dejé mi hogar a una edad temprana hasta el día de hoy, cada paso ha sido una cuesta arriba. Por eso, agradezco profundamente a todos los que me han acompañado en este viaje.

En primer lugar, agradezco enormemente a mis amigos de la universidad. Su apoyo constante y sus valiosas sugerencias han sido cruciales en cada etapa del desarrollo del proyecto. Sus conocimientos y perspectivas han enriquecido significativamente mi trabajo.

A mi polola, le estoy profundamente agradecido por su paciencia y apoyo incondicional. Sus preguntas simples que me ayudaron a simplificar y mejorar las soluciones técnicas, permitiéndome ver el proyecto desde un ángulo más claro, dándome ideas geniales.

Mi agradecimiento se extiende a los paramédicos del trabajo de la mutual, quienes me proporcionaron el dispositivo AMBU. Este recurso fue esencial para las pruebas y el desarrollo del equipo y su disposición para colaborar ha sido de gran importancia.

Agradezco también a todas las personas que me ofrecieron herramientas y apoyo a través de diversas plataformas y foros en línea. Los comentarios y recomendaciones que recibí me ayudaron a ajustar y enfocar el proyecto de manera más efectiva, enriqueciendo la calidad del trabajo final.

Un agradecimiento especial va para mi amigo Iván Rojas, quien revisó y ayudó a mejorar el código del proyecto. Su experiencia y conocimientos fueron invaluables para asegurar el correcto funcionamiento del sistema, dándome un enfoque más simple a ideas difíciles con eso pude iniciar desde cero.

También agradezco a mis profesores especialmente a profesor Guelis Montenegro. El cual me ayudo dándome ánimos en momentos difíciles los cuales valoro muchísimo.

Finalmente, a mi amigo Iliass Hannaoui, le agradezco por su apoyo constante y por animarme durante los momentos más desafiantes. Su aliento y asistencia fueron cruciales para mantener mi motivación y avance en el proyecto.

A todos ustedes, mi más sincero agradecimiento por su colaboración, apoyo y dedicación. Cada uno de ustedes ha jugado un papel importante en la culminación exitosa de esta tesis.

APÉNDICE 1: MENÚ DE CONFIGURACIÓN.

LCD	Elemento	Descripción
Volumen [ml]	Volumen.	<p>La configuración de volumen corresponde al ajuste para determinar la capacidad del volumen que queremos proyectar hacia la persona, muchas veces se confunde por capacidad pulmonar.</p> <p>Esta unidad aparecerá en la pantalla principal mostrando el valor fijado con la sigla XXX=mL.</p> <p>Por ejemplo: si ajustamos el valor a 450ml en la pantalla principal mostrara “450=mL” en el apartado de variables.</p> <p>Rango: 10 [ml] a 500 [ml].</p> <p>Unidad: mililitro.</p> <p>Intervalo de cambio: 10 ml.</p>
Vel. Inspira	Velocidad de Inspiración	<p>La configuración de velocidad corresponde al ajuste para determinar la velocidad de movimiento que realiza el equipo cuando realiza el ejercicio de inspiración.</p> <p>Rango: 10% a 100%. Correspondiente a 800mm/s</p> <p>Unidad: Porcentaje</p> <p>Intervalo de cambio: 1%.</p>
Tiempo mant.	Tiempo de Mantención	<p>La configuración de tiempo de Mantención corresponde al ajuste para determinar el tiempo que el equipo se detiene después de una inspiración.</p> <p>Rango: 0.5 [s] a 5[s].</p> <p>Unidad de medida: Segundos.</p> <p>Intervalo de cambio: 0.1seg.</p>
Vel. Expira	Velocidad de Expiración	<p>La configuración de velocidad corresponde al ajuste para determinar la velocidad de movimiento que realiza el equipo cuando realiza el ejercicio de</p>

		<p>expiración.</p> <p>Rango: 10% a 100%. Correspondiente a 800mm/s</p> <p>Unidad: Porcentaje</p> <p>Intervalo de cambio: 1%.</p>
T.Reposo	Tiempo de Reposo	<p>La configuración de tiempo de reposo corresponde al ajuste para determinar el tiempo que el equipo se detiene después de una expiración, también se conoce como tiempo de descanso.</p> <p>Rango: 0.5 [s] a 5[s].</p> <p>Unidad de medida: Segundos.</p> <p>Intervalo de cambio: 0.1seg.</p>
Sonido Alar.	Sonido de alarma	<p>La configuración del sonido de la alarma permite activar el sonido de bip que se emite en caso de alarma, acompañado por una alerta intermitente de una luz amarilla.</p> <p>Rango: 2 (SI-NO).</p> <p>Unidad de medida: no posee</p> <p>Intervalo de cambio: 1.</p>
Bip apoyo	Bip de apoyo	<p>La configuración del bip de apoyo permite activar el sonido de bip que se emite en caso de inicial el proceso de inspiración, expiración o ambas.</p> <p>Rango: 3 (Insp , Exip, I+D).</p> <p>Unidad de medida: no posee</p> <p>Intervalo de cambio: 1.</p>
Sen. Fuerza	Sensor de fuerza	<p>La configuración de sensor de fuerza, corresponde al ajuste para determinar el uso o no de la variable de la fuerza ejercida por el equipo, donde el sensor son dos celdas de carga.</p>

		<p>Rango: 2 (SI – NO).</p> <p>Unidad de medida: No posee.</p> <p>Intervalo de cambio: 1.</p>
Fuerza Max.	Fuerza Máxima	<p>La configuración de fuerza máxima, corresponde al ajuste para determinar el límite de fuerza ejercida por el mecanismo hacia la paleta fija que contiene la celda de carga.</p> <p>Rango: 0.5 kg - 10 kg.</p> <p>Unidad de medida: kilogramo.</p> <p>Intervalo de cambio: 0.1 kg.</p>
Accion F.Max.	Acción de fuerza Máxima	<p>La configuración de acción fuerza máxima, corresponde al ajuste para determinar el actuar del equipo en caso de sobre pasar el valor de fuerza máxima, también de estar activo el valor de Sensor de fuerza.</p> <p>Donde podemos determinar alarma o detención.</p> <p>Rango: 2 (alar– det.)</p> <p>Unidad de medida: No posee</p> <p>Intervalo de cambio: 1</p>
Presion Trig.	Presión Trigger	<p>La configuración de presión trigger, corresponde al ajuste para determinar el uso o no de la presión de vacío, con este dato se puede determinar uso del equipo</p> <p>Rango: 2 (SI – NO).</p>

		Unidad de medida: No posee. Intervalo de cambio: 1.
Umbral Trig.	Umbral Trigger	<p>La configuración de umbral trigger, corresponde al ajuste para determinar el límite de succión ejercida por la persona al momento de inspirara el cual iniciara el movimiento.</p> <p>Rango:-1mb a- 15mb</p> <p>Unidad de medida: milibar</p> <p>Intervalo de cambio:1</p>
Accion Trig.	Acción Trigger	<p>La configuración de acción de trigger, corresponde al ajuste para determinar el actuar del equipo en caso de sobre pasar el valor de presión de vacío, se debe tener en cuenta que debe estar activo el valor de presión trigger.</p> <p>Donde podemos determinar:</p> <p><u>Único</u>:realiza la inspiración solo cuando se sobrepasa el límite de umbral trigger.</p> <p><u>Automático</u>: realiza la inspiración cuando se sobrepasa el límite de umbral trigger o por tiempo.</p> <p>Rango: 2 (Solo - Auto.)</p> <p>Unidad de medida: No posee</p> <p>Intervalo de cambio: 1</p>
Presion Max.	Presión Máxima	La configuración de presión máxima, corresponde al ajuste para determinar el uso o no de la variable de

		<p>presión para realizar cambio de función en el equipo.</p> <p>Rango: 5 mb - 50mb</p> <p>Unidad de medida: milibar</p> <p>Intervalo de cambio: 1</p>
Rango Pmax	Rango de Presión Máxima	<p>La configuración de rango de presión máxima, corresponde al valor de umbral del límite de presión.</p> <p>Rango: 0.5 kg - 10 kg.</p> <p>Unidad de medida: kilogramo.</p> <p>Intervalo de cambio: 0.1 kg.</p>
Accion Pmax.	Acción Presión Máxima	<p>La configuración de acción de presión máxima, corresponde al ajuste para determinar el actuar del equipo en caso de sobre pasar el valor de presión máxima, también de estar activo el valor de Sensor de presión.</p> <p>Donde podemos determinar alarma o detención.</p> <p>Rango: 2 (alar– det.)</p> <p>Unidad de medida: No posee</p> <p>Intervalo de cambio: 1</p>
Entrada Digi.	Entrada Digital	<p>La configuración de entrada digital corresponde al uso de entrada digital, lo cual puede ser: interruptor, botón sensor, o cualquier elemento que sea externo al el equipo.</p> <p>Rango: 2 (Si - No)</p>

		Unidad de medida: No posee Intervalo de cambio: 1
Accion Digi.	Acción entrada digital	<p>La configuración de Acción para entrada digital permite ajustar cómo reaccionará el equipo cuando se active la entrada. Esta acción se puede configurar en cuatro modos diferentes: Detención, Alarma, inspiración, Automático</p> <p>Rango: 2 (Det. - alar. – Insp - auto)</p> <p>Unidad de medida: No posee</p> <p>Intervalo de cambio: 1</p>

APÉNDICE 2: CARRO DE COMPRA

Cesta de la compra (20)

Seleccionar todos los artículos | [Borrar artículos seleccionados](#)

Entrega mejorada por vendedor global ⓘ

 Has mejorado a entrega en 14 días >

	Tornillo redondo de acero inoxidable o negro, Perno Allen de cabeza hexagonal... Stainless Steel 304/M4 20pcs/20... >	<input type="button" value="−"/> <input checked="" type="button" value="1"/> <input type="button" value="+"/>  
	Placa de interfaz serie IIC I2C TWI SPI, Puerto 1602, 2004, LCD1602, adaptad... CLP 347 >	<input type="button" value="−"/> <input checked="" type="button" value="1"/> <input type="button" value="+"/>  
	Módulo adaptador de interfaz Serial para Arduino, LCD2004 + I2C 2004 20x4... Yellow green >	<input type="button" value="−"/> <input checked="" type="button" value="1"/> <input type="button" value="+"/>  
	Teclado de interruptor de membrana para coche inteligente arduino, matriz... 1x4 >	<input type="button" value="−"/> <input checked="" type="button" value="1"/> <input type="button" value="+"/>  
	Interruptor de botón de autoreinicio, 1 piezas, cabeza de seta pequeña, par... XB2-BA31 >	<input type="button" value="−"/> <input checked="" type="button" value="1"/> <input type="button" value="+"/>  
	Interruptor de botón de autoreinicio, 1 piezas, cabeza de seta pequeña, par... XB2-BA42 >	<input type="button" value="−"/> <input checked="" type="button" value="1"/> <input type="button" value="+"/>  

Fuente: Creación Propia

Figura apéndice 2-1 captura de pantalla N°1 de carrito de compras de Aliexpress.

The screenshot shows a shopping cart on the AliExpress platform with the following items:

- CLP 5.612**: 8mm 7.5mm 6.5mm Varilla de acero inoxidable 304 eje lineal red... (2 units)
- CLP 2.842**: 1 Uds 4 Uds SK8 SK10 SK12 SK13 SK16 SK20 SK25 SK30 de alta calidad sopor... (4 units)
- CLP 930**: Buje de bloque deslizante de rodamiento de bolas de movimiento lineal, pie... (4 units)
- CLP 1.886**: Células de carga de báscula humana, módulo AD HX711, portátil, electrónic... (2 units)
- CLP 3.084**: Polea tensora de sincronización para impresora 3D, rueda síncrona tensora ... (1 unit)
- Artículos de vendedores internacionales** (link)
- Button Switch Electrical Store** (link)
 - CLP 765**: Indicador de potencia LED montado en Panel, luz piloto, lámpara de señal, A... (1 unit)
 - CLP 765**: Indicador de potencia LED montado en Panel, luz piloto, lámpara de señal, A... (1 unit)

Fuente: Creación Propia

Figura apéndice 2-2 captura de pantalla N°2 de carrito de compras de Aliexpress.

 Indicador de potencia LED montado en Panel, luz piloto, lámpara de señal, A... [Red/22mm/24V >](#) - 1 +

CLP 765
+ Envío:CLP802

 W-SOURCE Store Toma de corriente eléctrica LED IEC320 C14, interruptor basculante de latón... [1PCS AC-06A >](#) - 1 +

CLP 4.885
Envío gratis

 AIRSKY Automation Store Correa de distribución abierta GT2 para impresora 3D, banda de goma de fi... [1Meter/GT2-10mm >](#) - 1 +

CLP 3.954
Envío gratis

 Carol Store Polea de sincronización 2GT, 12T, 14T, 14T, 15T, 16T, 17T, 18T, 19T, 20 dientes, 2... [20Teeth/Belt Width 10mm/Bore 8... >](#) - 1 +

CLP 2.030
Envío gratis

 Cludray Motor Store Controlador de Motor paso a paso de 2 fases, controlador de Motor paso a ... [CHINA >](#) - 1 +

CLP 28.745
+ Envío:CLP155

 HLTNC Automation Store Motor paso a paso Nema 23 DE 2 fases, 57HS5630A4, 1,8 grados, 1,2nm/171... [8mm >](#) - 1 +

CLP 7.981
+ Envío:CLP20.126

Fuente: Creación Propia

Figura apéndice 2-3 captura de pantalla N°3 de carrito de compras de Aliexpress.



Fuente: Creación Propia

Figura 3-75 captura de pantalla N°4 de carrito de compras de Aliexpress.



Fuente: Creación Propia

Figura apéndice 2-4 captura de pantalla de total de carrito de compras de Aliexpress.

My Project

Project Name	Date Created All	Last Update All	Search	Create a New Project
Apoyo para la respiracion No invasivo	12/29/2023	12/29/2023	17	Copy Project

Fuente: <https://www.lcsc.com/personalCenter/myProject/11B6443E646EA5948EE7286FFF892BB4>

Figura apéndice 2-5 Captura de pantalla de listado de compra de componentes de LCSC como proyecto.

Product Detail	Quantity	Available	Unit Price	Ext. Price	Notes ⓘ	Operate
 125mW Thin Film Resistor 150V ±0.5% ±25ppm/°C 5kΩ 0805 Chip Resistor - Surface Mount ROHS LCSC#: C2828783 Mfr#: ARG050TC5001 Mfr: Viking Tech Customer #:	50 Min:50, Mult:50 Full Reel: 5000	43350 In Stock 6700 ship in 1-2 days 36650 ship in 3-4 days	US\$0.0104	US\$0.52	Edit	Delete
 250mW Thick Film Resistors 150V ±100ppm/°C ±1% 1kΩ 0805 Chip Resistor - Surface Mount ROHS LCSC#: C727989 Mfr#: AC0805FR-7W1KL Mfr: YAGEO Customer #:	100 Min:100, Mult:100 Full Reel: 5000	4300 In Stock ship in 3-4 days	US\$0.0067	US\$0.67	Edit	Delete
 250mW Thick Film Resistors ±100ppm/°C ±1% 10kΩ 0805 Chip Resistor - Surface Mount ROHS LCSC#: C855098 Mfr#: AC0805FR-7W10KL Mfr: YAGEO Customer #:	100 Min:100, Mult:100 Full Reel: 5000	17500 In Stock 12600 ship in 1-2 days 4900 ship in 3-4 days	US\$0.0075	US\$0.75	Edit	Delete
 220μF 50V 220mA@120Hz ±20% SMD,D10xL10mm Aluminum Electrolytic Capacitors - SMD ROHS LCSC#: C125977 Mfr#: UWT1H221MNL1GS Mfr: Nichicon Customer #:	1 Min:1, Mult:1 Full Reel: 500	1524494 In Stock 576577 ship in 1-2 days 947917 ship in 3-4 days	US\$0.1891	US\$0.19	Edit	Delete
 680μF 35V 380mA ±20% SMD,D10xL10.5mm Aluminum Electrolytic Capacitors - SMD ROHS LCSC#: C7461236 Mfr#: 126RV0105 Mfr: KNSCHA Customer #:	5 Min:5, Mult:5 Full Reel: 500	11940 In Stock 5275 ship in 1-2 days 6665 ship in 3-4 days	US\$0.1433 US\$0.1500	US\$0.72	Edit	Delete
 - Fuse ROHS LCSC#: C2936128 Mfr#: 0218010.HXP Mfr: Litefuse Customer #:	5 Min:5, Mult:5	470 In Stock ship in 3-4 days	US\$0.3309	US\$1.65	Edit	Delete

Fuente: <https://www.lcsc.com/personalCenter/myProject/11B6443E646EA5948EE7286FFF892BB4>

Figura apéndice 2-6 Captura de pantalla N°1 de carro de compra a componentes de LCSC.

<input checked="" type="checkbox"/>	 <p>2.54mm Plugin 3A 4P bronze -40°C~+105°C Straight 1x4P Top Square Holes Plugin,P=2.54mm Female Headers ROHS LCSC#: C2718488 Mfr#: C2718488 Mfr: BOOMELE(Boom Precision Elec) Customer #:</p>	<input type="text" value="10"/> Min:10, Mult:10	22130 In Stock 7380 ship in 1-2 days 14750 ship in 3-4 days	US\$0.043	US\$0.43		
<input checked="" type="checkbox"/>	 <p>2.54mm Plugin 6P Straight 1x6P Plugin,P=2.54mm Female Headers ROHS LCSC#: C40877 Mfr#: C40877 Mfr: BOOMELE(Boom Precision Elec) Customer #:</p>	<input type="text" value="10"/> Min:10, Mult:10	8580 In Stock 4240 ship in 1-2 days 4340 ship in 3-4 days	US\$0.071	US\$0.71		
<input checked="" type="checkbox"/>	 <p>2.54mm Plugin 3A 15P -40°C~+105°C Straight 1x15P 8.5mm Top Square Holes Plugin,P=2.54mm Female Headers ROHS LCSC#: C2897378 Mfr#: PM254-1-15-Z-8.5 Mfr: HCTL Customer #:</p>	<input type="text" value="6"/> Min:6, Mult:6	5610 In Stock 2286 ship in 1-2 days 3324 ship in 3-4 days	US\$0.179 US\$0.2237	US\$1.07		
<input checked="" type="checkbox"/>	 <p>- Fuse Holders ROHS LCSC#: C377342 Mfr#: HF-004U Mfr: HOLLY Customer #:</p>	<input type="text" value="10"/> Min:10, Mult:10	3280 In Stock ship in 1-2 days	US\$0.0551	US\$0.55		
<input checked="" type="checkbox"/>	 <p>3A 33uH ±20% SMD Power Inductors ROHS LCSC#: C181844 Mfr#: SLH1207S330MTT Mfr: Suntek Tech Customer #:</p>	<input type="text" value="1"/> Min:5, Mult:5 Full Reel: 500	20510 In Stock 10275 ship in 1-2 days 10235 ship in 3-4 days	US\$0.1917	US\$0.19		
<input checked="" type="checkbox"/>	 <p>1x2P 10A 300V Green 14~26 Straight 5mm 2.5 1.2 Plugin,P=5mm Screw terminal ROHS LCSC#: C557646 Mfr#: XY126V-5.0-2P Mfr: Ningbo Xinlaiya Elec. Customer #:</p>	<input type="text" value="5"/> Min:5, Mult:5	41400 In Stock 31655 ship in 1-2 days 9745 ship in 3-4 days	US\$0.0743	US\$0.37		

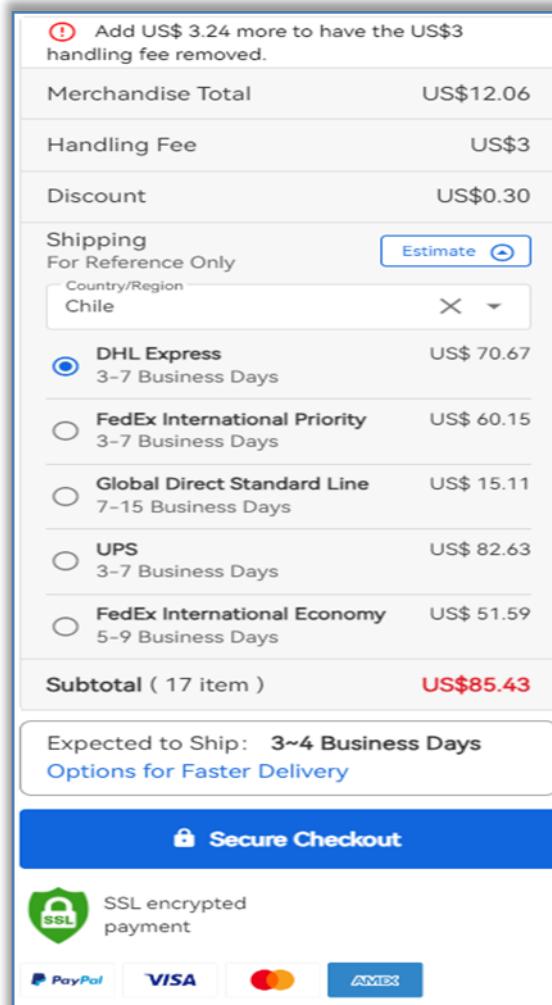
Fuente: <https://www.lcsc.com/personalCenter/myProject/11B6443E646EA5948EE7286FFF892BB4>

Figura apéndice 2-7 Captura de pantalla N°2 de carro de compra a componentes de LCSC.

<input checked="" type="checkbox"/>	 <p>1x3P 12A 300V Green Straight 3.5mm 0.2-1.5 1.3 Plugin,P=3.5mm Screw terminal ROHS LCSC#: C2683167 Mfr#: Y00301500000G Mfr: Amphenol ICC Customer #: <input type="text"/></p>	<input type="text" value="1"/> Min:1, Mult:1	4251 In Stock 3689 ship in 1-2 days 562 ship in 3-4 days	US\$0.6046	US\$0.60		
<input checked="" type="checkbox"/>	 <p>80V 5000V 50mA 0.1V@1mA,20mA 1 6V 1.2V DC DIP-4 Optocouplers - Phototransistor Output ROHS LCSC#: C96006 Mfr#: PC817X3NSZ9F Mfr: Sharp Microelectronics Customer #: <input type="text"/></p>	<input type="text" value="10"/> Min:5, Mult:5	17800 In Stock 9925 ship in 1-2 days 7875 ship in 3-4 days	US\$0.0865	US\$0.87		
<input checked="" type="checkbox"/>	 <p>40V Single 3A 0.55V@3A SMA(DO-214AC) Schottky Barrier Diodes (SBD) ROHS LCSC#: C8678 Mfr#: SS34 Mfr: MDD (Microdiode Electronics) Customer #: <input type="text"/></p>	<input type="text" value="20"/> Min:20, Mult:20 Full Reel: 5000	813600 In Stock 396160 ship in 1-2 days 417440 ship in 3-4 days	US\$0.0233	US\$0.47		
<input checked="" type="checkbox"/>	 <p>60V 115mA 200mW 7.5Q@10V,500mA 2.5V@250uA N Channel SOT-23 MOSFETs ROHS LCSC#: C8545 Mfr#: 2N7002 Mfr: Jiangsu Changling Electronics Technology Co., Ltd. Customer #: <input type="text"/></p>	<input type="text" value="50"/> Min:50, Mult:50 Full Reel: 3000	1134650 In Stock 612050 ship in 1-2 days 522600 ship in 3-4 days	US\$0.015	US\$0.75		
<input checked="" type="checkbox"/>	 <p>6A 950mV@6A 1kV R-6 Diodes - General Purpose ROHS LCSC#: C2461 Mfr#: 6A10 Mfr: MDD (Microdiode Electronics) Customer #: <input type="text"/></p>	<input type="text" value="5"/> Min:5, Mult:5	32750 In Stock 19140 ship in 1-2 days 13610 ship in 3-4 days	US\$0.0951	US\$0.48		

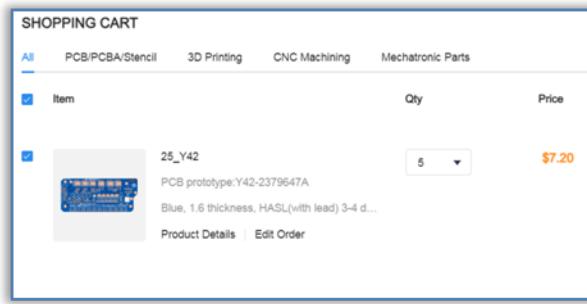
Fuente: <https://www.lcsc.com/personalCenter/myProject/11B6443E646EA5948EE7286FFF892BB4>

Figura apéndice 2-8 Captura de pantalla N°3 de carro de compra a componentes de LCSC.



Fuente: <https://www.lcsc.com/personalCenter/myProject/11B6443E646EA5948EE7286FFF892BB4>

Figura apéndice 2-9 Captura de pantalla de total de carro de compra en componentes de LCSC.



Fuente: www.jlpcb.com.

Figura apéndice 2-10 Captura pantalla N°4 de configuración de proyecto dentro de la página de JLPCB, asignación de envío.

SUMMARY	
Merchandise Total	\$7.20
Shipping	\$59.26
Grand Total:	\$66.46

Fuente: www.jlcpcb.com.

Figura apéndice 2-11 Captura pantalla de valor total para la elaboración de PCB dentro de la página de JLCPCB.

APÉNDICE 3: CODIGO

Código de Sistema de apoyo para la respiración no invasiva

```

/*
*****
*
* Proyecto:          Apoyo Respiratorio No Invasivo
* Autor:            Sebastian Padilla
* Area:             Medicina, Automatización, Electronica.
* Descripcion:      Equipo desarrollado para realizar apoyo respiratorio
*                   con bolsa Ambu, automatizado.
* Version Software: 2.0.1
* Actualizacion:    20-08-24
* Version Hardware: 1.0
* Platform:         espressif32
* Board:            esp32doit-devkit-v1
* Framework:        arduino
* Documentación:   https://github.com/padilla107/ambu_proyect.git
*
* PinOut board
*
* INPUT
*     Botones
*         start:      D12
*         stop:       D13
*         Sensor magnetico: D32
*         Boton membrana 1-2: D34 analogo      Lv1=1.03v  Lv2=2.06v  Lv3-2=3.3v
*         Boton membrana 2-4: D35 analogo      Lv1=1.03v  Lv2=2.06v  Lv3-2=3.3v
*         Celda de carga HX711
*             SCK:        D14
*             DT:         D27
*         Modulo presion MPS20N004D
*             SCK:        D19
*             OUT:        D18
*             Input:      D23
*
* OUTPUT
*     Motor Stepper
*         ENA:        D33
*         PULL:       D26
*         DIR:        D25
*         Luces piloto
*             Verde:    D4
*             Amarillo: D2
*             Rojo:     D15
*             Buzzer:   D5
*             LCD 16*2 I2C
*                 SDA:      D21

```

```

*      SCL:      D22
*
*****
*/
//----- librerias
#include<Arduino.h>//19-08-24 002
#include"EEPROM.h"
#include<Wire.h>
#include<LiquidCrystal_I2C.h>
#include<Chrono.h>
#include<HX711.h>
#include<AccelStepper.h>

LiquidCrystal_I2C lcd(0x27,20,4);
HX711 O_PRESION;
HX711 O_CELDA;
AccelStepper stepper (1,26,25); // (tipo driver=1 , pull, direccion) pin
constint PRESION_DOUT_PIN =18;//18
constint PRESION_SCK_PIN =19;//19
constint CELDA_DOUT_PIN =27;//27
constint CELDA_SCK_PIN =14;//14
///////////////////////////////
bool enciende=0;
bool b_inicio=0;
int pin_entrada=23;
bool estado_input=0;

int Pin_lcd12=34;
int valor_lcd12=0;
int pin_lcd34=35;
int valor_lcd34=0;

float PRESION_CERO=0;
//fLoat SENSOR_PRESION=0;
bool b_presion_cero=0;
int PRESION=0;
bool trigg=0;
int P_SUMA=0;
float CELDA_CERO=0;
float CELDA=0;
float pos_celda=0;    // dato positivo de celda
int C_SUMA=0;

```

```

bool light_on=0;
int n_inspira=1;
bool insp_boton=0;
bool insp_presion=0;
bool insp_tiempo=0;
bool estado_boton_start;
bool estado_boton_stop;
int boton_start=12;           //pin de entrada
int boton_stop=13;            //pin de entrda
int luz_amarillo=2;//8        //pin salida
int luz_rojo=15;              //pin salida
int luz_verde=4;              //pin salida
int buzzer=5;                 //pin salida
int PinSensor_Mag=32;//11      /sin usar
bool Sensor_Mag=0;             //sin usar
//int boton_input=0;
//bool estado_input=0;          //sin usar

bool actual_1,anterior_1,boton_1; //estado de boton abajo
bool actual_2,anterior_2,boton_2; //estado de boton arriba
bool actual_3,anterior_3,boton_3; //estado de boton derecha
bool actual_4,anterior_4,boton_4; //estado de boton izquierda
int pos_f=0;
int pag_menu=0;
bool entra_menu=0;               // Entrada menu Lcd
bool entra_MenuAnterior=0;
/////////////////////////////// String para concatenar
char conca_d1[30], conca_d2[30], conca_d3[30], conca_d4[30], conca_d5[30],
conca_d6[30];
char conca_d7[30], conca_d8[30], conca_d9[30], conca_d10[30], conca_d11[30],
conca_d12[30];
char conca_d13[30], conca_d14[30], conca_d15[30], conca_d16[30],
conca_d17[30], conca_d18[30];
char
conca_A1[30],conca_A2[30],conca_A3[30],conca_A4[30],conca_A1_1[30],conca_A1_2[
30];
int conca_linea_A1_1=0;
int conca_linea_A1_2=1;
int ant_dato_0;
int dato_0=0;                   // pulsacion de boton start=1 o stop=0
(envia)
int dato_1=10;                  //recorrido 0-100 /
(10-100)
int dato_2=10;                  //velocidad inspiracion 0-100
(envia)

```

```

float dato_3=0; //tiempo mantencion
(0.0 - 5.0)
int dato3_1=0; //tiempo mantencion*10(entero)
(envia)
int dato_4=10; //velocidad expiracion
(envia)
float dato_5=0; //tiempo reposo
(0.0 - 5.0)
int dato5_1=0; // tiempo reposo (entero)
int dato_6=0; // sonido alarma off/on
int dato_7=0; // bip apoyo off/insp/exp/e+i
int dato_8=0; // fuerza si/no
float dato_9=0.5; // Rango de fuerza 0.5-10 kg (0.5 -
10.0)
int dato_10=0; // Accion de F (det- alarm)
int dato_11=0; // Presion inspira (si - no)
int dato_12=-1; // Umbral detc. (10 - 40)
int dato_13=0; // Accion Umb. (comp-solo)
int dato_14=0; // Presion Max (si - no)
int dato_15=5; // Rango Pmax. (5-40)
int dato_16=0; // Accion Pmax (det-alarma)
int dato_17=0; // Boton (si - no)
int dato_18=0; // accion boton (detencion/
alarma/insp/auto)
bool alarma=0; /// dato de alarma 0 apagada 1 encendido
bool EstadoAmarillo=0;
bool AntAlarm=0;
int menu_preceso=0;
int linea=0;
int linea_cambia=0;
bool mantenido4=0;
bool mantenido3=0;
bool pulso3=0;
bool pulso4=0;
bool beep_Act=0;
bool beep_Alarm=0;
int Status=0;
int linea_menu=0;
int linea_alarma=0;
int linea_presion=0;
////////////////////////////// variables de motor
int TRANS_dato_1=0; //recorrido 0-100 /
(10-100)
int TRANS_dato_2=0; //velocidad inspiracion 0-100
(envia)

```

```

int TRANS_dato3_1=0;                                //tiempo mantencion*10(entero)
    (envia)
int TRANS_dato_4=0;                                //velocidad expiracion
    (envia)
int TRANS_dato5_1=0;                                // tiempo reposo (entero)

int home1=0;
int bandera_home=0;
int Calibra1=0;
int etapa=0;
int estado_sensor;
int detecta_atras=0;
int home_paso=0;
int funciona=0;
int Longitud=0;                                     //map()10-100:200-1000
int Velocidad_Inspira=0;
int Aceleracion_Inspira=0;
int Aceleracion_Expira=0;      //
int Velocidad_Expira=0;
int acel_fin_adelante=0;
int acel_inicio_delante=0;
int Rampa_adelante=0;
int pasos_margen=300;
int total_pasos=1200;
int Tiempo_inspira=0;
int Tiempo_reposo=0;
int acel_fin_atras=0;
int acel_inicio_atras=0;
int Rampa_atras=0;
int pulsoUP=10;        /// 10 us
int cambio_menu=0;
//-----                                         Temporizadores
Chrono luz;           // backlight lcd
Chrono timeOut;        // salir del menu
Chrono Tiempo_Beep;     // tiempo de sonido beep
Chrono tiempo_manB4;    // tiempo
Chrono tiempo1_manB4;
Chrono tiempo_manB3;
Chrono tiempo1_manB3;
Chrono time_alarma;      // tiempo de alarma
Chrono Refresh_menu;     // tiempo de refresh LCD
Chrono tiempo_inspira;
Chrono tiempo_1(Chrono::MICROS);
Chrono tiempo_2(Chrono::MICROS);
Chrono Tiempo_espera;

```

```

/////////////////////////////inicio

void alerta(){
    if(alarma!=AntAlarm){           // inicial de alarma
        EstadoAmarillo=1;
        digitalWrite(luz_amarillo,EstadoAmarillo);
        AntAlarm=alarma;
    }
    switch(alarma){   //funcion de luz y sonido alternado
        case0:
            EstadoAmarillo=0;
            digitalWrite(luz_amarillo,EstadoAmarillo);      //apagado de luz
            time_alarma.restart();                         //reset tiempo de Luz
            time_alarma.stop();                           //stop tiempo de Luz
            break;
        case1:
            time_alarma.resume();
            if(time_alarma.hasPassed(250)){
                EstadoAmarillo=!EstadoAmarillo;
                if((EstadoAmarillo)&&(dato_6)){
                    beep_Alarm=1;
                }                               // si es distinto
                digitalWrite(luz_amarillo,EstadoAmarillo);  //
                time_alarma.restart();          // reset tiempo alarma
            }
            break;
    }
}

void beep_respiracion(){
    if((beep_Act)|| (beep_Alarm)){
        Tiempo_Beep.resume();
        digitalWrite(buzzer,HIGH);
    }
    if(Tiempo_Beep.hasPassed(50)){
        beep_Act=0;
        beep_Alarm=0;
        digitalWrite(buzzer,LOW);
        Tiempo_Beep.restart();
        Tiempo_Beep.stop();
    }
}

} //fin de void beep

```

```

void inicio(){
    if(enciende==0){
        dato_0=EEPROM.readInt(0);
        dato_1=EEPROM.readInt(10);
        dato_2=EEPROM.readInt(20);
        dato_3=EEPROM.readFloat(30);
        dato_4=EEPROM.readInt(40);
        dato_5=EEPROM.readFloat(50);
        dato_6=EEPROM.readInt(60);
        dato_7=EEPROM.readInt(70);
        dato_8=EEPROM.readInt(80);
        dato_9=EEPROM.readFloat(90);
        dato_10=EEPROM.readInt(100);
        dato_11=EEPROM.readInt(110);
        dato_12=EEPROM.readInt(120);
        dato_13=EEPROM.readInt(130);
        dato_14=EEPROM.readInt(140);
        dato_15=EEPROM.readInt(150);
        dato_16=EEPROM.readInt(160);
        dato_17=EEPROM.readInt(170);
        dato_18=EEPROM.readInt(180);

        dato3_1=int(((dato_3)*10)+0.4);
        dato5_1=int(((dato_5)*10)+0.4);

        if(dato_0==1){
            b_inicio=1;
        }
        enciende=1;
    }

}

void Input(){
    estado_boton_start=digitalRead(boton_start);
    estado_boton_stop=digitalRead(boton_stop);
    estado_sensor=digitalRead(PinSensor_Mag);
}

void sensor_celda(){

    CELDA=((O_CELDA.read()-(CELDA_CERO))/97250); // PRESION 115600

    if(dato_8){ // activacion sensor fuerza
        if((CELDA>=dato_9)&&(dato_0)){
}

```

```

switch(datos_10) // detencion o aviso
{
    case0:
        alarma=1;
        etapa=0;
        linea_alarma=3;
        stepper.setCurrentPosition(0);
        break;

    case1:
        alarma=1;
        linea_alarma=3;

        break;
    }
}
}

void sensor_presion(){
    if(etapa!=18){
        PRESION=((O_PRESION.read())-(PRESION_CERO))/64285); // (cmH2O o mbar)
64285
    }
    if(etapa==18){
        PRESION=((O_PRESION.read())-(PRESION_CERO))/12350); // (cmH2O o mbar)
64285
    }
}

if(datos_14){
    if(PRESION>=datos_15){
        switch (datos_16)
        {
            case0:
                alarma=1;
                etapa=0;
                linea_alarma=4;
                stepper.setCurrentPosition(0);
                break;
            case1:
                alarma=1;
                linea_alarma=4;
                break;
        }
    }
}

```

```

        }

if(datos_11){
    if((tiempo_inspira.hasPassed(1000))&&(etapa==18)){
        if(b_presion_cero==0){
            PRESION_CERO=0_PRESION.read();
            b_presion_cero=1;
        }
        if(PRESION<=dato_12){

            trigg=1;
        }
    }
    else{
        trigg=0;
    }
}
}

void pin_input(){
    estado_input=digitalRead(pin_entrada);
    if((dato_17)&&(estado_input)){
        switch (dato_18)
        {
            case0:
                alarma=1;
                etapa=0;
                linea_alarma=5;
                stepper.setCurrentPosition(0);
                break;
            case1:
                alarma=1;
                linea_alarma=5;
                break;
        }
    }
}

void proceso(){

if(((estado_boton_start)|| (b_inicio==1))&&(etapa==0)){
    etapa=1;
    b_inicio=0;
    //Tiempo_espera.start();
}
}

```

```

    }

    if((estado_boton_start)&&((linea_alarma>=3))){    /// reset de alarmas con
start
        alarma=0;
        linea_alarma=0;
    }
    if(estado_boton_stop){
        etapa=0;
        alarma=0;
        linea_alarma=0;
        Tiempo_espera.restart();
        Tiempo_espera.stop();
        stepper.setCurrentPosition(0);
        stepper.disableOutputs();
    }
    if(etapa==0){
        digitalWrite(luz_rojo,1);
        digitalWrite(luz_verde,0);
    }
    if(etapa!=0){
        digitalWrite(luz_rojo,0);
        digitalWrite(luz_verde,1);
    }
    if((home1==12) || (home1==32)){

        TRANS_dato_1=dato_1;                      //recorrido 10-100 /
        TRANS_dato_2= dato_2;                      //velocidad inspiracion 0-100
        TRANS_dato3_1=dato3_1;                     //tiempo mantencion*10(entero)
        TRANS_dato_4= dato_4;                      //velocidad expiracion
        TRANS_dato5_1=dato5_1;                     // tiempo reposo (entero)
        Longitud=map(TRANS_dato_1,10,100,200,1000); //pasos Longitud
        Tiempo_inspira=map(TRANS_dato3_1,0,50,1,5000); // TIEMPO INSPIRA (ms)
        Tiempo_reposo=map(TRANS_dato5_1,0,50,1,5000);
    }

}

void teclado(){
    valor_lcd12=analogRead(Pin_lcd12); // analogico para boton 1,2
    if((valor_lcd12 >850)&&(valor_lcd12 <1500)){actual_2=1;} else
{actual_2=0;}
    if((valor_lcd12 >1500)&&(valor_lcd12 <3500)){actual_1=1;} else
{actual_1=0;}
    valor_lcd34=analogRead(pin_lcd34);
    if((valor_lcd34 >850)&&(valor_lcd34 <1500)){ actual_4=1;} else {actual_4=0;}
}

```

```

if((valor_lcd34 >1500)&&(valor_lcd34 <3500)){ actual_3=1;}else
{actual_3=0;}
//boton 3 mantenido
if(actual_3==1){ //codigo de mantencion de boton
    tiempo_manB3.resume();
    tiempo1_manB3.resume();
}
if(tiempo_manB3.elapsed()>500){
    mantenido3=1;
}
if(actual_3==0){
    tiempo_manB3.restart();
    tiempo_manB3.stop();
    tiempo1_manB3.restart();
    tiempo1_manB3.stop();
    pulso3=0;
    mantenido3=0;
}
// boton 3 pulsos
if(mantenido3==1){
    if(tiempo1_manB3.hasPassed(50)){
        pulso3=1;
        cambio_menu=1;
        tiempo1_manB3.restart();
    }
}
//boton 4 mantenido
if(actual_4==1){
    tiempo_manB4.resume();
    tiempo1_manB4.resume();
}
if(tiempo_manB4.elapsed()>500){
    mantenido4=1;
}
if(actual_4==0){
    tiempo_manB4.restart();
    tiempo_manB4.stop();
    tiempo1_manB4.restart();
    tiempo1_manB4.stop();
    pulso4=0;
    mantenido4=0;
}
if(mantenido4==1){
    if(tiempo1_manB4.hasPassed(50)){
        pulso4=1;
    }
}

```

```

cambio_menu=1;
tiempo1_manB4.restart();
}
}
switch(actual_1){ // Lach boton 1
case0:
switch(anterior_1)
{
case0: boton_1=LOW; anterior_1=LOW;break;
case1: boton_1=LOW; anterior_1=LOW;break;
}
break;
case1:
switch(anterior_1)
{
case0: boton_1=HIGH; cambio_menu=1; anterior_1=HIGH; break;
case1: boton_1=LOW; anterior_1=HIGH; break;
}
break;
}
switch(actual_2){ // Lach boton 2
case0:
switch(anterior_2)
{
case0: boton_2=LOW; anterior_2=LOW;break;
case1: boton_2=LOW; anterior_2=LOW;break;
}
break;
case1:
switch(anterior_2)
{
case0: boton_2=HIGH; cambio_menu=1; anterior_2=HIGH; break;
case1: boton_2=LOW; anterior_2=HIGH; break;
}
break;
}
switch(actual_3){ // Lach boton 3
case0:
switch(anterior_3)
{
case0: boton_3=LOW; anterior_3=LOW;break;
case1: boton_3=LOW; anterior_3=LOW;break;
}
break;
case1:

```

```

switch(anterior_3)
{
    case0: boton_3=HIGH; cambio_menu=1; anterior_3=HIGH; break;
    case1: boton_3=LOW; anterior_3=HIGH; break;
}
break;

switch(actual_4){ // Lach boton 3
case0:
    switch(anterior_4)
    {
        case0: boton_4=LOW; anterior_4=LOW;break;
        case1: boton_4=LOW; anterior_4=LOW;break;
    }
break;
case1:
    switch(anterior_4)
    {
        case0: boton_4=HIGH; cambio_menu=1; anterior_4=HIGH; break;
        case1: boton_4=LOW; anterior_4=HIGH; break;
    }
break;
}

voidlight(){// tempo Luz fondo

if(actual_1||actual_2||actual_3||actual_4||mantenido3||mantenido4||(linea_alarm
a!=0)){
    luz.restart();
    timeOut.restart();
    //lcd.setBacklight(1);
    if(light_on==0){                      // una vez
        light_on=1;
        lcd.setBacklight(1);                //ENCENDER LUZ}
    }
}
if(luz.elapsed()>30000){ //15
    luz.restart();
    luz.stop();
    pos_f=0;
    pag_menu=0;
    lcd.setBacklight(0);
}
}

```

```

}

void time_Out(){//tempo para salir al proceso solo
  if(timeOut.elapsed()>5000){
    timeOut.restart();
    timeOut.stop();
    light_on=0;
    entra_menu=0;
    //B_menu_dato=0;
    //EEPROM.writeInt(0,dato_0);
    EEPROM.writeInt(10,dato_1);
    EEPROM.writeInt(20,dato_2);
    EEPROM.writeFloat(30,dato_3);
    EEPROM.writeInt(40,dato_4);
    EEPROM.writeFloat(50,dato_5);
    EEPROM.writeInt(60,dato_6);
    EEPROM.writeInt(70,dato_7);
    EEPROM.writeInt(80,dato_8);
    EEPROM.writeFloat(90,dato_9);
    EEPROM.writeInt(100,dato_10);
    EEPROM.writeInt(110,dato_11);
    EEPROM.writeInt(120,dato_12);
    EEPROM.writeInt(130,dato_13);
    EEPROM.writeInt(140,dato_14);
    EEPROM.writeInt(150,dato_15);
    EEPROM.writeInt(160,dato_16);
    EEPROM.writeInt(170,dato_17);
    EEPROM.writeInt(180,dato_18);
    EEPROM.commit();
  }
}

void menu_logica(){// desplazamiento flecha abajo
  if(boton_1){
    switch(pos_f){
      case0:pos_f=1;boton_1 =LOW;break;
      case1:pos_f=2;boton_1 =LOW;break;
      case2:pos_f=3;boton_1 =LOW;break;
      case3:
        switch(pag_menu) // desplazamiento de menu abajo
        {
          case0: pag_menu =1; boton_1 =LOW;break;
          case1: pag_menu =2; boton_1 =LOW;break;
          case2: pag_menu =3; boton_1 =LOW;break;
        }
    }
  }
}

```

```

    case3: pag_menu =4; boton_1 =LOW;break;
    case4: pag_menu =5; boton_1 =LOW;break;
    case5: pag_menu =6; boton_1 =LOW;break;
    case6: pag_menu =7; boton_1 =LOW;break;
    case7: pag_menu =8; boton_1 =LOW;break;
    case8: pag_menu =9; boton_1 =LOW;break;
    case9: pag_menu =10; boton_1 =LOW;break;
    case10: pag_menu =11; boton_1 =LOW;break;
    case11: pag_menu =12; boton_1 =LOW;break;
    case12: pag_menu =13; boton_1 =LOW;break;
    case13: pag_menu =14; boton_1 =LOW;break;
}
break;
}
}

if(boton_2){ // desplazamiento flecha arriba
switch(pos_f)
{
    case3: pos_f=2; boton_2 =LOW;break;
    case2: pos_f=1; boton_2 =LOW;break;
    case1: pos_f=0; boton_2 =LOW;break;
    case0: // desplazamiento menu arriba
        switch(pag_menu)
    {
        case1: pag_menu =0; boton_2 =LOW;break;
        case2: pag_menu =1; boton_2 =LOW;break;
        case3: pag_menu =2; boton_2 =LOW;break;
        case4: pag_menu =3; boton_2 =LOW;break;
        case5: pag_menu =4; boton_2 =LOW;break;
        case6: pag_menu =5; boton_2 =LOW;break;
        case7: pag_menu =6; boton_2 =LOW;break;
        case8: pag_menu =7; boton_2 =LOW;break;
        case9: pag_menu =8; boton_2 =LOW;break;
        case10: pag_menu =9; boton_2 =LOW;break;
        case11: pag_menu =10; boton_2 =LOW;break;
        case12: pag_menu =11; boton_2 =LOW;break;
        case13: pag_menu =12; boton_2 =LOW;break;
        case14: pag_menu =13; boton_2 =LOW;break;
    }
    break;
}
}

linea=pag_menu+pos_f;
}

```

```

switch(linea)           // Logica linea del menu
{
    case0:           ///////////////////////////////// Linea 0   Volumen
    (dato_1)
        if(((boton_4 ==HIGH)|| (pulso4==1))&&(dato_1<500)){
            dato_1=dato_1+10;
            pulso4=0;
            boton_4=LOW;
        }
        if(((boton_3 ==HIGH)|| (pulso3==1))&&(dato_1>10)){
            dato_1=dato_1-10;
            pulso3=0;
            boton_3=LOW;
        }
    break;
    ///////////////////////////////// Linea 1   Velocidad
inspira (dato_2)
    case1:
        if(((boton_4 ==HIGH)|| (pulso4==1))&&(dato_2<100)){
            dato_2=dato_2+5;
            pulso4=0;
            boton_4=LOW;
        }
        if(((boton_3 ==HIGH)|| (pulso3==1))&&(dato_2>10)){
            dato_2=dato_2-5;
            pulso3=0;
            boton_3=LOW;
        }
    break;
    ///////////////////////////////// Linea 2   tiempo de
mant (dato_3)
    case2:
        if(((boton_4 ==HIGH)|| (pulso4==1))&&(dato_3<4.9)){
            dato_3=dato_3+0.1;
            pulso4=0;
            boton_4=LOW;
        }
        if(((boton_3 ==HIGH)|| (pulso3==1))&&(dato_3>0.1)){
            dato_3=dato_3-0.1;
            pulso3=0;
            boton_3=LOW;
        }
    dato3_1=int(((dato_3)*10)+0.4); // pasa dato flotante a entero para ser
enviado

```

```

break;
/////////////////////////////// linea 3 velocidad
Expira (dato 4)
case3:
if(((boton_4 ==HIGH) || (pulso4==1))&&(dato_4<100)){
    dato_4=dato_4+5;
    pulso4=0;
    boton_4=LOW;
}
if(((boton_3 ==HIGH) || (pulso3==1))&&(dato_4>10)){
    dato_4=dato_4-5;
    pulso3=0;
    boton_3=LOW;
}

break;
/////////////////////////////// linea 4 tiempo reposo
(dato 5)
case4:
if(((boton_4 ==HIGH) || (pulso4==1))&&(dato_5<=4.9)){
    dato_5=dato_5+0.1;
    pulso4=0;
    boton_4=LOW;
}
if(((boton_3 ==HIGH) || (pulso3==1))&&(dato_5>0.1)){
    dato_5=dato_5-0.1;
    pulso3=0;
    boton_3=LOW;
}
dato5_1=int(((dato_5)*10)+0.4);
break;
/////////////////////////////// linea 5 Sonido alarma
(dato_6)
case5:
if((boton_4 ==HIGH)){
    dato_6=!dato_6;
    pulso4=0;
    boton_4=LOW;
}
if((boton_3 ==HIGH)){
    dato_6=!dato_6;
    pulso3=0;
    boton_3=LOW;
}
}// fin dato7
break;

```

```

////////// (dato_7)                                Linea 6    bip de apoyo
case6:
  if((boton_4 ==HIGH)&&(dato_7<3)){
    dato_7++;
    pulso4=0;
    boton_4=LOW;
  }
  if((boton_3 ==HIGH)&&(dato_7>0)){
    dato_7--;
    pulso3=0;
    boton_3=LOW;
  } //fin dato 8
break;
////////// (dato_8)      (si-no)                  Linea 7    Sensor de fuerza
case7:
  if((boton_4 ==HIGH)){
    dato_8=!dato_8;
    pulso4=0;
    boton_4=LOW;
  }
  if((boton_3 ==HIGH)){
    dato_8=!dato_8;
    pulso3=0;
    boton_3=LOW;
  } // fin dato7
break;
////////// (dato_9)      (0.5-10 kg)             Linea 8    Fuerza Max.
case8:
  if(((boton_4 ==HIGH)|| (pulso4==1))&&(dato_9<=10.0)){
    dato_9=dato_9+0.1;
    pulso4=0;
    boton_4=LOW;
  }
  if(((boton_3 ==HIGH)|| (pulso3==1))&&(dato_9>=0.6)){
    dato_9=dato_9-0.1;
    pulso3=0;
    boton_3=LOW;
  }
break;
////////// (dato_10)     (det-aviso)            Linea 9    Accion fuerza max.
case9:

```

```

if((boton_4 ==HIGH)){
    dato_10=!dato_10;
    pulso4=0;
    boton_4=LOW;
}
if((boton_3 ==HIGH)){
    dato_10=!dato_10;
    pulso3=0;
    boton_3=LOW;
}// fin dato7
break;
/////////////////////////////// Linea 10 presion trigger
(dato_11) (No-Si)
case10:
if((boton_4 ==HIGH)){
    dato_11=!dato_11;
    pulso4=0;
    boton_4=LOW;
}
if((boton_3 ==HIGH)){
    dato_11=!dato_11;
    pulso3=0;
    boton_3=LOW;
}// fin dato7
break;
/////////////////////////////// Linea 11 Umbral Trigger
(dato_12) (-0.5 -15)
case11:
if(((boton_4 ==HIGH) || (pulso4==1))&&(dato_12>-20)){
    dato_12--;
    pulso4=0;
    boton_4=LOW;
}
if(((boton_3 ==HIGH) || (pulso3==1))&&(dato_12<-1)){
    dato_12++;
    pulso3=0;
    boton_3=LOW;
}
break;
/////////////////////////////// Linea 12 Accion Trigger
(dato_13) ( solo - auto)
case12:
if((boton_4 ==HIGH)){
    dato_13=!dato_13;
    pulso4=0;
}

```

```

        boton_4=LOW;
    }
    if((boton_3 ==HIGH)){
        dato_13=!dato_13;
        pulso3=0;
        boton_3=LOW;
    }// fin dato7
    break;
    //////////////////////////////// Linea 13 Presion Max. (dato_14)
(No - Si)
case13:
    if((boton_4 ==HIGH)){
        dato_14=!dato_14;
        pulso4=0;
        boton_4=LOW;
    }
    if((boton_3 ==HIGH)){
        dato_14=!dato_14;
        pulso3=0;
        boton_3=LOW;
    }// fin dato7
    break;
    //////////////////////////////// Linea 14 Rango Presion max
(dato_15) (5 - 50 [mb] )
case14:
    if(((boton_4 ==HIGH)|| (pulso4==1))&&(dato_15<50)){
        dato_15++;
        pulso4=0;
        boton_4=LOW;
    }
    if(((boton_3 ==HIGH)|| (pulso3==1))&&(dato_15>5)){
        dato_15--;
        pulso3=0;
        boton_3=LOW;
    }
    break;
    //////////////////////////////// Linea 15 Accion Presion max
(dato_16) (Det - Alarma)
case15:
    if((boton_4 ==HIGH)){
        dato_16=!dato_16;
        pulso4=0;
        boton_4=LOW;
    }
    if((boton_3 ==HIGH)){

```

```

        dato_16=!dato_16;
        pulso3=0;
        boton_3=LOW;
    }// fin dato7
    break;
////////////////////////////// Linea 16 Entrada digital
(dato_17) (Si - No)
case16:
    if((boton_4 ==HIGH)){
        dato_17=!dato_17;
        pulso4=0;
        boton_4=LOW;
    }
    if((boton_3 ==HIGH)){
        dato_17=!dato_17;
        pulso3=0;
        boton_3=LOW;
    }// fin dato7
    break;
////////////////////////////// Linea 17 [Accion digital]
(dato_18) (Det - Alarma - insp - Auto)
case17:
    if((boton_4 ==HIGH)&&(dato_18<3)){
        dato_18++;
        pulso4=0;
        boton_4=LOW;
    }
    if((boton_3 ==HIGH)&&(dato_18>0)){
        dato_18--;
        pulso3=0;
        boton_3=LOW;
    }//fin dato 8
    break;
}
/////////// fin switch (linea)

voidconcatena_lcd(){ // variables de union para enviar lcd
    sprintf(conca_d1," Volumen [mL] %3d ",dato_1); // linea
0
    sprintf(conca_d2," Vel. Inspira %3d%s",dato_2,"%"); // linea 1
    sprintf(conca_d3," Tiempo mant. %1.1f%s",dato_3,"s"); // linea 2
    sprintf(conca_d4," Vel. Expira %3d%s",dato_4,"%"); // linea 3
}

```

```
sprintf(conca_d5," T.Reposo           %1.1f%s",dato_5,"s");          //  
Linea 4  
    switch(dato_6){                                //  
Linea 5  
        case0:printf(conca_d6," Sonido Alar.      %2s","No");  break;  
                           //  
        case1:printf(conca_d6," Sonido Alar.      %2s","Si");   break;  
                           //  
    }  
    switch(dato_7){                                //  
Linea 6  
        case0:printf(conca_d7," Bip apoyo       %2s","No");  break;  
        case1:printf(conca_d7," Bip apoyo       %4s","Insp");break;  
        case2:printf(conca_d7," Bip apoyo       %4s","Expi");break;  
        case3:printf(conca_d7," Bip apoyo       %3s","E+I");break;  
    }  
    switch(dato_8){                                //  
Linea 7  
        case0:printf(conca_d8," Sen. Fuerza     %2s","No");  break;  
                           //  
        case1:printf(conca_d8," Sen. Fuerza     %2s","Si");   break;  
                           //  
    }  
    if(dato_9<10){printf(conca_d9," Fuerza Max.   %1.1f%2s",dato_9,"kg");}  
//Linea 8  
    else{printf(conca_d9," Fuerza Max.   %2.1f%2s",dato_9,"kg");}  
  
    switch(dato_10){                                //  
Linea 9  
        case0:printf(conca_d10," Accion F.Max.  %4s","Det.");  break;  
                           //  
        case1:printf(conca_d10," Accion F.Max. %5s","Alarm");  break;  
                           //  
    }  
    switch(dato_11){                                //  
Linea 10  
        case0:printf(conca_d11," Presion Trig.  %2s","No");  break;  
                           //  
        case1:printf(conca_d11," Presion Trig.  %2s","Si");   break;  
                           //  
    }  
    printf(conca_d12," Umbral Trig. %3d%2s",dato_12,"mb");  
//Linea 11  
    switch(dato_13){                                //  
Linea 12
```

```

case0:sprintf(conca_d13," Accion Trig.    %4s","Solo"); break;
case1:sprintf(conca_d13," Accion Trig.    %4s","Auto"); break;
}

switch(dato_14){                                         // linea 13
    case0:sprintf(conca_d14," Presion Max.    %2s","No"); break;
    case1:sprintf(conca_d14," Presion Max.    %2s","Si"); break;
}

sprintf(conca_d15," Rango Pmax.    %2d%2s",dato_15,"mb"); // linea 14
switch(dato_16){                                         // linea 15
    case0:sprintf(conca_d16," Accion Pmax.    %4s","Det."); break;
    case1:sprintf(conca_d16," Accion Pmax.    %5s","Alarm"); break;
}

switch(dato_17){                                         // linea 16
    case0:sprintf(conca_d17," Entrada Digi.    %2s","No"); break;
    case1:sprintf(conca_d17," Entrada Digi.    %2s","Si"); break;
}

switch(dato_18){                                         // linea 17
    case0:sprintf(conca_d18," Accion Digi.    %4s","Det."); break;
    case1:sprintf(conca_d18," Accion Digi.    %5s","Alar."); break;
    case2:sprintf(conca_d18," Accion Digi.    %5s","Insp."); break;
    case3:sprintf(conca_d18," Accion Digi.    %4s","Auto"); break;
}

switch(linea)                                         // posicion de flecha sobre escribe
{
    case0:
        //sprintf(conca_d1, "%sLongitud      %3d %s", ">", dato_1, "%");
        sprintf(conca_d1,"%sVolumen [mL]  %3d  ", ">", dato_1);
// Linea 0
}

```

```

//sprintf(conca_d1, "%sLongitud      %3d %s", ">", dato_1, "%");
break;
case1:
    sprintf(conca_d2,"%sVel. Inspira  %3d%s", ">", dato_2, "%");
break;
case2:
    sprintf(conca_d3,"%sTiempo mant.  %1.1f%s", ">", dato_3, "s");
break;
case3:
    sprintf(conca_d4,"%sVel. Expira   %3d%s", ">", dato_4, "%");
break;
case4:
    sprintf(conca_d5,"%sT.Reposo      %1.1f%s", ">", dato_5, "s");
break;
case5:
    switch(dato_6){                                //
// Linea 5
        case0: sprintf(conca_d6,"%sSonido Alar.    %2s", ">","No"); break;
                  //
        case1: sprintf(conca_d6,"%sSonido Alar.    %2s", ">","Si"); break;
                  //
    }
break;
case6:
    switch(dato_7){
// Linea 6
        case0: sprintf(conca_d7,"%sBip apoyo      %2s", ">","No"); break;
        case1: sprintf(conca_d7,"%sBip apoyo      %4s", ">","Insp"); break;
        case2: sprintf(conca_d7,"%sBip apoyo      %4s", ">","Expi"); break;
        case3: sprintf(conca_d7,"%sBip apoyo      %3s", ">","E+I"); break;
    }
break;

case7:
    switch(dato_8){                                //
// Linea 7
        case0: sprintf(conca_d8,"%sSen. Fuerza    %2s", ">","No"); break;
        case1: sprintf(conca_d8,"%sSen. Fuerza    %2s", ">","Si"); break;
    }
break;
case8:
    if(dato_9<10){sprintf(conca_d9,"%sFuerza Max.
%1.1f%2s", ">", dato_9, "kg");} //Linea 8
}

```

```

else{sprintf(conca_d9,"%sFuerza Max. %2.1f%2s",>,dato_9,"kg");}
break;
case9:
switch(dato_10){                                //
// Linea 9
    case0: sprintf(conca_d10,"%sAccion F.Max. %4s",>,"Det."); break;
    case1: sprintf(conca_d10,"%sAccion F.Max. %5s",>,"Alarm"); break;
}
break;
case10:
switch(dato_11){                                //
    case0: sprintf(conca_d11,"%sPresion Trig. %2s",>,"No"); break;
    case1: sprintf(conca_d11,"%sPresion Trig. %2s",>,"Si"); break;
}
break;
case11:
sprintf(conca_d12,"%sUmbral Trig. %3d%2s",>,dato_12,"mb");
break;
case12:
switch(dato_13){                                //
    case0: sprintf(conca_d13,"%sAccion Trig. %4s",>,"Solo"); break;
    case1: sprintf(conca_d13,"%sAccion Trig. %4s",>,"Auto"); break;
}
break;
case13:
switch(dato_14){                                //
    case0: sprintf(conca_d14,"%sPresion Max. %2s",>,"No"); break;
    case1: sprintf(conca_d14,"%sPresion Max. %2s",>,"Si"); break;
}
break;
case14:
sprintf(conca_d15,"%sRango Pmax. %2d%2s",>,dato_15,"mb");
break;
case15:
switch(dato_16){                                //
    case0: sprintf(conca_d16,"%sAccion Pmax. %4s",>,"Det."); break;
}

```

```

    case1: sprintf(conca_d16,"%sAccion Pmax. %5s",>,"Alarm"); break;

}
break;
case16:
switch(dato_17){ // 
    case0: sprintf(conca_d17,"%sEntrada Digi. %2s",>,"No"); break;

    case1: sprintf(conca_d17,"%sEntrada Digi. %2s",>,"Si"); break;

}
break;
case17:
switch(dato_18){ // 
    case0: sprintf(conca_d18,"%sAccion Digi. %4s",>,"Det."); break;

    case1: sprintf(conca_d18,"%sAccion Digi. %5s",>,"Alar."); break;
    case2: sprintf(conca_d18,"%sAccion Digi. %5s",>,"Insp."); break;

    case3: sprintf(conca_d18,"%sAccion Digi. %4s",>,"Auto"); break;

}
break;
}

}

voidconca_pricipal(){
/////////////////////////////Linea 0
switch(n_inspira)
{
    case1:sprintf(conca_A1_1," TIEMPO "); break;
    case2:sprintf(conca_A1_1," TRIGGER "); break;
    case3:sprintf(conca_A1_1," T+TRIG. "); break;
    case4:sprintf(conca_A1_1," INPUT "); break;
    case5:sprintf(conca_A1_1," T+INPUT "); break;
    case6:sprintf(conca_A1_1," TRIG.+IN "); break;
    case7:sprintf(conca_A1_1,"TRIG+T+IN"); break;
}

if((Status==0)&&(linea_alarma==0)){
    conca_linea_A1_2=1;
    dato_0=0;
}
if((Status==1)&&(linea_alarma==0)){
    conca_linea_A1_2=2;
}

```

```

}
if((Status==2)&&(linea_alarma==0)){
    conca_linea_A1_2=3;
    dato_0=1;
}
if((linea_alarma==1)|| (linea_alarma==2)){
    conca_linea_A1_2=4;
    dato_0=0;
}
if(Status==2){
    if(linea_alarma==3){
        conca_linea_A1_2=5;
    }
    if(linea_alarma==4){
        conca_linea_A1_2=6;
    }
    if(linea_alarma==5){
        conca_linea_A1_2=7;
    }
}
if(Status==0){
    if(linea_alarma==3){
        conca_linea_A1_2=8;
        dato_0=0;
    }
    if(linea_alarma==4){
        conca_linea_A1_2=9;
        dato_0=0;
    }
    if(linea_alarma==5){
        conca_linea_A1_2=10;
        dato_0=0;
    }
}
if(dato_0!=ant_dato_0){
    ant_dato_0=dato_0;
    EEPROM.writeInt(0,dato_0);
    EEPROM.commit();
}
switch(conca_linea_A1_2)
{
    // Status= 0 detenido; 1 calibrando; 2 funcionando
    // Linea_alarma= 0 sistema ok;1 sensor no det; 2 sen no acciona;3lim
    fuerza:4 Limitepresion; 5 input
    case1:   sprintf(conca_A1_2,"DET. LISTO");  break;
}

```

```

case2: sprintf(conca_A1_2,"CALIBRANDO"); break;
case3: sprintf(conca_A1_2," FUNCIONA "); break;
case4: sprintf(conca_A1_2,"ERR.SENSOR"); break;
case5: sprintf(conca_A1_2,"ALM. F.MAX"); break;
case6: sprintf(conca_A1_2,"ALM. P.MAX"); break;
case7: sprintf(conca_A1_2,"ALM. INPUT"); break;
case8: sprintf(conca_A1_2,"DET. F.MAX"); break;
case9: sprintf(conca_A1_2,"DET. P.MAX"); break;
case10: sprintf(conca_A1_2,"DET. INPUT"); break;
}

sprintf(conca_A1,"%9s%10s",conca_A1_1,conca_A1_2);
/////////////////////////////Linea 1
switch(dato_11)
{
    case0:
        sprintf(conca_A2,"%3d=mb %3d Trig= Off",PRESION,dato_12);
        break;
    case1:
        switch(dato_13)
        {
            case0:
                sprintf(conca_A2,"%3d=mb %3d Trig=Insp",PRESION,dato_12);
                break;
            case1:
                sprintf(conca_A2,"%3d=mb %3d Trig=Auto",PRESION,dato_12);
                break;
        }
        break;
}
/////////////////////////////Linea 2
if(CELDA>=0){          // variables para positivos
    pos_celda=CELDA;
}
switch(dato_14)
{
    case0:
        sprintf(conca_A3,"%1.1f=Kg  %2d Pmax= Off",pos_celda,dato_15);
        break;
    case1:
        switch(dato_16)
        {
            case0:
                sprintf(conca_A3,"%1.1f=Kg  %2d Pmax= Det",pos_celda,dato_15);
                break;
            case1:

```

```

        sprintf(conca_A3,"%1.1f=Kg  %2d Pmax=Alar",pos_celda,dato_15);
        break;
    }
    break;
}
//////////////////////////////Linea 3
switch(dato_8)
{
    case0:
        sprintf(conca_A4,"%3d=mL %1.1f Fmax= Off",dato_1,dato_9);
        break;
    case1:
        switch(dato_10)
        {
            case0:
                sprintf(conca_A4,"%3d=mL %1.1f Fmax=Det.",dato_1,dato_9);
                break;
            case1:
                sprintf(conca_A4,"%3d=mL %1.1f Fmax=Alar",dato_1,dato_9);
                break;
        }
        break;
}

void pantallaPrincipal(){

    lcd.setCursor(0,0); lcd.print(conca_A1);
    lcd.setCursor(0,1); lcd.print(conca_A2);
    lcd.setCursor(0,2); lcd.print(conca_A3);
    lcd.setCursor(0,3); lcd.print(conca_A4);
}
void menu_texto(){//LCD print

switch(pag_menu){    ///////////////////////// Pagina del menu
    case0: //menu 1
        lcd.setCursor(0,0); lcd.print(conca_d1);
        lcd.setCursor(0,1); lcd.print(conca_d2);
        lcd.setCursor(0,2); lcd.print(conca_d3);
        lcd.setCursor(0,3); lcd.print(conca_d4);
    break;
    case1: //menu2
        lcd.setCursor(0,0); lcd.print(conca_d2);
        lcd.setCursor(0,1); lcd.print(conca_d3);
        lcd.setCursor(0,2); lcd.print(conca_d4);
}
}

```

```
lcd.setCursor(0,3); lcd.print(conca_d5);
break;
case2:
lcd.setCursor(0,0); lcd.print(conca_d3);
lcd.setCursor(0,1); lcd.print(conca_d4);
lcd.setCursor(0,2); lcd.print(conca_d5);
lcd.setCursor(0,3); lcd.print(conca_d6);
break;
case3:
lcd.setCursor(0,0); lcd.print(conca_d4);
lcd.setCursor(0,1); lcd.print(conca_d5);
lcd.setCursor(0,2); lcd.print(conca_d6);
lcd.setCursor(0,3); lcd.print(conca_d7);
break;
case4:
lcd.setCursor(0,0); lcd.print(conca_d5);
lcd.setCursor(0,1); lcd.print(conca_d6);
lcd.setCursor(0,2); lcd.print(conca_d7);
lcd.setCursor(0,3); lcd.print(conca_d8);
break;
case5:
lcd.setCursor(0,0); lcd.print(conca_d6);
lcd.setCursor(0,1); lcd.print(conca_d7);
lcd.setCursor(0,2); lcd.print(conca_d8);
lcd.setCursor(0,3); lcd.print(conca_d9);
break;
case6:
lcd.setCursor(0,0); lcd.print(conca_d7);
lcd.setCursor(0,1); lcd.print(conca_d8);
lcd.setCursor(0,2); lcd.print(conca_d9);
lcd.setCursor(0,3); lcd.print(conca_d10);
break;
case7:
lcd.setCursor(0,0); lcd.print(conca_d8);
lcd.setCursor(0,1); lcd.print(conca_d9);
lcd.setCursor(0,2); lcd.print(conca_d10);
lcd.setCursor(0,3); lcd.print(conca_d11);
break;
case8:
lcd.setCursor(0,0); lcd.print(conca_d9);
lcd.setCursor(0,1); lcd.print(conca_d10);
lcd.setCursor(0,2); lcd.print(conca_d11);
lcd.setCursor(0,3); lcd.print(conca_d12);
break;
case9:
```

```

lcd.setCursor(0,0); lcd.print(conca_d10);
lcd.setCursor(0,1); lcd.print(conca_d11);
lcd.setCursor(0,2); lcd.print(conca_d12);
lcd.setCursor(0,3); lcd.print(conca_d13);
break;
case10:
lcd.setCursor(0,0); lcd.print(conca_d11);
lcd.setCursor(0,1); lcd.print(conca_d12);
lcd.setCursor(0,2); lcd.print(conca_d13);
lcd.setCursor(0,3); lcd.print(conca_d14);
break;
case11:
lcd.setCursor(0,0); lcd.print(conca_d12);
lcd.setCursor(0,1); lcd.print(conca_d13);
lcd.setCursor(0,2); lcd.print(conca_d14);
lcd.setCursor(0,3); lcd.print(conca_d15);
break;
case12:
lcd.setCursor(0,0); lcd.print(conca_d13);
lcd.setCursor(0,1); lcd.print(conca_d14);
lcd.setCursor(0,2); lcd.print(conca_d15);
lcd.setCursor(0,3); lcd.print(conca_d16);
break;
case13:
lcd.setCursor(0,0); lcd.print(conca_d14);
lcd.setCursor(0,1); lcd.print(conca_d15);
lcd.setCursor(0,2); lcd.print(conca_d16);
lcd.setCursor(0,3); lcd.print(conca_d17);
break;
case14:
lcd.setCursor(0,0); lcd.print(conca_d15);
lcd.setCursor(0,1); lcd.print(conca_d16);
lcd.setCursor(0,2); lcd.print(conca_d17);
lcd.setCursor(0,3); lcd.print(conca_d18);
break;
}
cambio_menu=0;
}
TaskHandle_t task2;
void loop2(void*parameter){
while(1){
//teclado();
time_Out();
light();
alerta();
}

```

```

beep_respiracion();
sensor_celda();
sensor_presion();
pin_input();
if(entra_menu){
    //concatena_Lcd();
}
//concatena_Lcd();
if(actual_1||actual_2||actual_3||actual_4){
    entra_menu=1;
}
if(entra_menu!=entra_MenuAnterior){      // clear al ser distinto
    lcd.clear();
    pos_f=0;
    entra_MenuAnterior=entra_menu;
}
if(Refresh_menu.hasPassed(10)){
    switch(entra_menu)
    {
        case0:
            conca_principal();
            pantallaPrincipal();
            break;

        case1: //LCD.setBacklight(1); //ENCENDER LUZ
            concatena_lcd();
            menu_texto();
            break;
    }
    Refresh_menu.restart();
    //delay(200); // resfresco de
}
vTaskDelay(1000); //advertencia watchdog
}

void setup(){
if(!EEPROM.begin(1000)){
    delay(1000);
    ESP.restart();
}
dato_0=EEPROM.readInt(0);
//Serial.begin(115200);
stepper.setEnablePin(33); // definir pin enable stepper
}

```

```

stepper.setPinsInverted(0,0,1); // invertir dirección, paso, habilitar
stepper.setMaxSpeed(4200); //steps/second 3320 a 500 rpm trabajo / 6640
=1000rpm max
stepper.setAcceleration(2000); // step/(second)^2 14000
stepper.disableOutputs();

delay(10);
O_PRESION.begin(PRESION_DOUT_PIN, PRESION_SCK_PIN);
O_CELDA.begin(CELDA_DOUT_PIN, CELDA_SCK_PIN);

for(int i =1; i <=5; i++){
    P_SUMA=P_SUMA+(O_PRESION.read());
    delay(100);
}
for(int i =1; i <=5; i++){
    C_SUMA=C_SUMA+(O_CELDA.read());
    delay(100);
}
PRESION_CERO=P_SUMA/5;
CELDA_CERO=C_SUMA/5;

Wire.begin();

lcd.begin(20,4);
lcd.clear();

pinMode(boton_start,INPUT);
pinMode(boton_stop,INPUT);
pinMode(luz_rojo,OUTPUT);
pinMode(luz_verde,OUTPUT);
pinMode(luz_amarillo,OUTPUT);
pinMode(buzzer,OUTPUT);
pinMode(PinSensor_Mag,INPUT);
pinMode(pin_entrada,INPUT);

luz.restart();
lcd.setBacklight(1);
timeOut.restart();
timeOut.stop();
tiempo_manB4.restart();
tiempo_manB4.stop();
tiempo1_manB4.restart();
tiempo1_manB4.stop();
tiempo_manB3.restart();
tiempo_manB3.stop();

```

```

tiempo1_manB3.restart();
tiempo1_manB3.stop();
time_alarma.restart();
time_alarma.stop();
Tiempo_espera.restart();
Tiempo_espera.stop();
Tiempo_Beep.restart();
Tiempo_Beep.stop();
Refresh_menu.restart();
tiempo_inspira.restart();
tiempo_inspira.stop();
xTaskCreatePinnedToCore(
    loop2, /*funcion de la tarea*/
    "task_2", /*Nombre de la tarea*/
    10000, /*Tamaño de la pila*/
    NULL, /*Parametros de entrada*/
    5, /*Prioridad de tarea*/
    &task2, /*objeto TaskHandle_t*/
    0); /*Nucleo donde se c*/
}

delay(500);
}

voidloop(){
    inicio();
    Input();
    proceso();
    teclado();
    menu_logica();

    switch(etapa){
        case0:

            stepper.setCurrentPosition(0);
            stepper.disableOutputs();
            Status=0;
        break;
        case1:
            alarma=1;
            Status=1;
            Tiempo_espera.restart();
            etapa=2;
        break;
        case2: // tiempo de espera para iniciar calibracion
            if(Tiempo_espera.hasPassed(1000)){

```

```

    Tiempo_espera.restart();
    Tiempo_espera.stop();
    etapa=3;
}
break;

case3:
stepper.enableOutputs();
stepper.setAcceleration(2000);
stepper.setMaxSpeed(4200);
stepper.moveTo(100);
//
etapa=4;
break;

case4:
if(stepper.currentPosition()>=100){ //no encontro sensor ida
stepper.setCurrentPosition(0);
Tiempo_espera.start();
etapa=5;
}
break;

case5:   /// tiempo de espera para buscar home hacia atras
if(Tiempo_espera.hasPassed(200)){
Tiempo_espera.restart();
Tiempo_espera.stop();
etapa=6;
}
break;

case6:

//stepper.enableOutputs();
stepper.setMaxSpeed(1000);    // velocidad hacia atras
stepper.moveTo(-1000);
etapa=7;
break;

case7:
if(stepper.currentPosition()<=-1000){ // no encontro sensor vuelta
stepper.setCurrentPosition(0);
etapa=0;
linea_alarma=1;
}
break;

```

```

case8:
    //stepper.stop();
    stepper.setMaxSpeed(200);
    stepper.moveTo(-150);
    etapa=9;
    break;

case9:
    if(stepper.currentPosition() == (-150)){ // sensor no acciona
        stepper.setCurrentPosition(0);
        etapa=0;
        linea_alarma=2;
    }
    if(estado_sensor==0){                         //posicion home
        stepper.setCurrentPosition(0);
        Tiempo_espera.restart();
        etapa=10;
    }
    break;

case10:
    if(Tiempo_espera.hasPassed(1000)){           // tiempo de espera despues de
calibracion
        Tiempo_espera.restart();
        Tiempo_espera.stop();
        alarma=0;
        //stepper.disableOutputs();
        etapa=11;
        Status=2;
    }
    break;
case11:                                //// calculos iniciales

TRANS_dato_1=dato_1;                      //recorrido 10-100 /
TRANS_dato_2= dato_2;                      //velocidad inspiracion 0-100
TRANS_dato3_1=dato3_1;                     //tiempo mantencion*10(entero)
TRANS_dato_4= dato_4;                      //velocidad expiracion
TRANS_dato5_1=dato5_1;                     // tiempo reposo (entero)

Longitud=map(TRANS_dato_1,10,500,200,918);

Velocidad_Inspira=map(TRANS_dato_2,10,100,1500,4000);
Aceleracion_Inspira=map(Velocidad_Inspira,1500,4000,1500,18000);

```

```

Tiempo_inspira=map(TRANS_dato3_1,0,50,1,5000);           // TIEMPO INSPIRA (ms)

Velocidad_Expira=map(TRANS_dato_4,10,100,1500,4000);
Aceleracion_Expira=map(Velocidad_Expira,1500,4000,1500,18000);

Tiempo_reposo=map(TRANS_dato5_1,0,50,1,5000);
//CELDA_CERO=(O_CELDA.read());
etapa=12;
break;
case12:                                // Movimiento INSPIRACION
stepper.setMaxSpeed(Velocidad_Inspira);
stepper.moveTo(Longitud);
stepper.setAcceleration(Aceleracion_Inspira);
etapa=13;
break;
case13:
if(stepper.currentPosition()== Longitud ){
    stepper.setCurrentPosition(0);
    Tiempo_espera.start();
    etapa=14;
} //no encontro sensor ida
break;
case14:
if(Tiempo_espera.hasPassed(Tiempo_inspira)){           // Tiempo INSPIRACION
    Tiempo_espera.restart();
    Tiempo_espera.stop();
    // stepper.disableOutputs ();
    etapa=15;
}
break;
case15:

stepper.setMaxSpeed(Velocidad_Expira);
stepper.moveTo(-Longitud +30);                      // margen sensor
stepper.setAcceleration(Aceleracion_Expira);
etapa=16;

break;
case16:
if(stepper.currentPosition()==(-Longitud +30)){ // sensor no encontrado
    stepper.setCurrentPosition(0);
    etapa=0;
    alarma=1;
    linea_alarma=1;
}

```

```

    }

    if(estado_sensor){
        //stepper.setCurrentPosition(0);
        stepper.stop();
        stepper.setMaxSpeed(500);
        stepper.moveTo(-Longitud-50);
        etapa=17;

    }

    break;
case17:
    if(stepper.currentPosition()==(-Longitud-50)){ // sensor no acciona
        stepper.setCurrentPosition(0);
        etapa=0;
        alarma=1;

        linea_alarma=2;
    }

    if(estado_sensor==0){                                //posicion home Expiracion
        stepper.setCurrentPosition(0);
        Tiempo_espera.start();
        tiempo_inspira.restart();
        b_presion_cero=0;
        etapa=18;
    }

    break;
case18:
    //tiempo_inspira.restart();
    switch(n_inspira)
    {
        case1:
            if(Tiempo_espera.hasPassed(Tiempo_reposo)){           // Tiempo
INSPIRACION
                Tiempo_espera.restart();
                Tiempo_espera.stop();
                etapa=11;
            }
        break;
        case2:
            if(trigg==HIGH){
                Tiempo_espera.restart();
                Tiempo_espera.stop();
                //trigg=0;
                etapa=11;
            }
    }
}

```

```

break;
case3:
    if((Tiempo_espera.hasPassed(Tiempo_reposo))||(trigg)){           // 
// Tiempo INSPIRACION
    Tiempo_espera.restart();
    Tiempo_espera.stop();
    //trigg=0;
    etapa=11;
}

break;
case4:
    if(estado_input){
        Tiempo_espera.restart();
        Tiempo_espera.stop();
        //trigg=0;
        etapa=11;
    }
break;
case5:
    if((Tiempo_espera.hasPassed(Tiempo_reposo))||(estado_input)){      //
// Tiempo INSPIRACION
    Tiempo_espera.restart();
    Tiempo_espera.stop();
    //trigg=0;
    etapa=11;
}
break;
case6:
    if((trigg)|| (estado_input)){
        Tiempo_espera.restart();
        Tiempo_espera.stop();
        //trigg=0;
        etapa=11;
    }
break;
case7:

if((Tiempo_espera.hasPassed(Tiempo_reposo))||(estado_input)|| (trigg)){   //
// Tiempo INSPIRACION
    Tiempo_espera.restart();
    Tiempo_espera.stop();
    //trigg=0;
    etapa=11;
}

```

```

        break;
    }
    break;
}

///////////  boton
if(dato_17){
    if(dato_18>=2){
        insp_boton=1;
    }
    else{insp_boton=0;}
}
else{insp_boton=0;}
/////////// presion
if(dato_11){
    if(dato_13>=0){
        insp_presion=1;
    }
    else{insp_presion=0;}
    //insp_presion=0;
}
else{insp_presion=0;}
/////////// tiempo
if((insp_presion==0)&&(insp_boton==0)){
    insp_tiempo=1;
}
elseif((insp_presion)&&(dato_13)){
    insp_tiempo=1;
}
elseif((insp_boton)&&(dato_18==3)){
    insp_tiempo=1;
}
else{
    insp_tiempo=0;
}
n_inspira=insp_boton*4+insp_presion*2+insp_tiempo;

if(estado_sensor &&(etapa==4||etapa==7)){
    stepper.setCurrentPosition(0);
    etapa=8;
}
if((etapa==12)&&((dato_7==1)|| (dato_7==3))){
    beep_Act=1;
}
if((etapa==15)&&((dato_7==2)|| (dato_7==3))){
```

```
    beep_Act=1;
}
stepper.run(); // funcion de motor
}
```