

UNIVERSIDAD NACIONAL DE CAAGUAZÚ
FACULTAD DE CIENCIAS Y TECNOLOGÍAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA



**PROPUESTA DE DISEÑO DE UN RESPIRADOR
ARTIFICIAL, MONITOREADO VÍA WEB, PARA
PACIENTES DEL ÁREA DE EMERGENCIA DEL
HOSPITAL DISTRITAL DE LA CIUDAD DE CAAGUAZÚ**

Osmar Javier Ramoa Duarte

Liz Mabel Aranda González

Tutor: Prof. Ing. Federico Rodolfo Fernández Gómez

CORONEL OVIEDO, MARZO DE 2025



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.



Usted es libre de:

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato
- **Adaptar** — remezclar, transformar y construir a partir del material

Bajo los siguientes términos:

- **Atribución** — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.
- **NoComercial** — Usted no puede hacer uso del material con [propósitos comerciales](#).



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

DERECHO DE AUTOR

Quienes suscriben, **Liz Aranda y Osmar Ramoa**, autores del trabajo de investigación titulado **“Propuesta de diseño de un respirador artificial, monitoreado vía web, para pacientes del área de emergencia del Hospital Distrital de la ciudad de Caaguazú”**, declaran que voluntariamente ceden a título gratuito en forma pura y simple ilimitada e irrevocablemente a favor de la Facultad de Ciencias y Tecnologías – UNCA, el derecho de autor de contenido patrimonial, que le corresponde sobre el trabajo de referencia. Conforme a lo anteriormente expresado, esta sesión le otorga a la FCyT la Facultad de comunicar la obra divulgarla, publicarla y reproducirla en soportes analógicos o digitales en la oportunidad que así lo estime conveniente. La FCyT deberá indicar qué autoría o creación del trabajo corresponde a nuestra persona y hará referencia a los autores y a las personas que hayan colaborado en la realización del presente trabajo de investigación.

En la ciudad de Coronel Oviedo, a los del mes dedel 2025

.....

Firmas



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

PÁGINA DE APROBACIÓN

Trabajo de fin de grado para la obtención del Título de Ingeniero en Electrónica, aprobado en representación de la Facultad Ciencias y Tecnología de la Universidad Nacional de Caaguazú, por el Tribunal Examinador constituido por los siguientes profesores y con la siguiente nota final:

Prof. Ing.

Prof. Ing.

Prof. Ing.

Calificación Final: _____

Acta N°: _____

Fecha: _____



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Dedicatoria

A Dios, por ser mi fuente de fortaleza y guía en cada paso de este camino, por darme la paciencia en los momentos difíciles y la esperanza para seguir adelante.

A mis padres, Alberto y Francisca, cuyo esfuerzo, amor y sacrificio me permitieron llegar hasta aquí. Su apoyo constante ha sido mi mayor motivación para seguir adelante y superar cada obstáculo.

A mis hermanos, Javier, Sandra y Luján, quienes han sido mi compañía inigualable en este viaje. Su presencia, consejos y palabras de aliento me han impulsado en los momentos de incertidumbre.

A mi sobrina Anahí, por ser una fuente constante de alegría y motivación. Su confianza en mí y su cariño incondicional han sido un refugio en los días más desafiantes. Este logro también se lo dedico a ella.

A mis abuelos, por su amor infinito y la sabiduría que han compartido conmigo. Su ejemplo me ha enseñado el valor del esfuerzo, la perseverancia y la humildad.

A mis amigos y compañeros de estudio, por acompañarme en este proceso, por hacer cada reto más llevadero y cada logro más significativo. Sus palabras de aliento y su compañía han sido fundamentales en este camino.

Liz Mabel Aranda González



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Dedicatoria

Dedico este trabajo a mi querida Madre, Mary, quien me ha apoyado en este largo camino, en los altos y bajos, y más aún en aquellos momentos en los que parecía que iba a desistir.

A mis hermanos, quienes fungieron como padres en cada paso que he dado y me ayudaron a crecer como persona. Sin ellos, no estaría donde me encuentro ahora.

A mi novia, Larissa, quien ha estado ahí desde el comienzo de este capítulo universitario de mi vida y cuyo aliento, apoyo y compañía me ayudaron a seguir adelante.

A mis queridos compañeros de la carrera, en especial a David Irrazábal y Gabriel Gauto, por haberme enseñado mucho sobre este maravilloso mundo de la electrónica.

Osmar Javier Ramoa Duarte



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Agradecimientos

En primer lugar, queremos agradecer a Dios por su guía omnipresente, que siempre nos ha acompañado, especialmente en esta etapa universitaria nuestra.

Expresamos nuestro agradecimiento al Ing. Federico Fernández por sus valiosas observaciones y su apoyo en el desarrollo de esta obra, las cuales nos permitieron orientar adecuadamente nuestro trabajo.

A todos nuestros compañeros de carrera, amigos y futuros colegas, por su colaboración en aspectos clave del proyecto.

Finalmente, a la Facultad de Ciencias y Tecnologías, por brindarnos un espacio para aprender y formarnos como profesionales.



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Resumen

El desarrollo de respiradores artificiales es fundamental para mejorar la respuesta hospitalaria en situaciones críticas. En el Hospital Distrital de Caaguazú, la escasez de estos dispositivos en el área de emergencias ha generado la necesidad de un diseño optimizado y ser capaz de monitorear las variables biomédicas en tiempo real. Este estudio tiene como objetivo proponer un respirador artificial con un sistema de control basado en tecnologías embebidas, asegurando su eficiencia operativa y la supervisión remota de sus parámetros de ventilación. La metodología empleada incluyó la recopilación de datos sobre la disponibilidad de equipos en el hospital, el análisis de normativas internacionales y la selección de componentes electrónicos y mecánicos adecuados para la implementación del sistema. Se diseñó un modelo 3D del dispositivo, integrando en el mismo controladores, sensores de presión y flujo y el actuador del sistema, además del diseño de una interfaz gráfica que permite la configuración de variables y el monitoreo en tiempo real a través de una plataforma web. Los resultados muestran que el diseño propuesto cumple con las normativas ISO 80601-2-12 y IEC 60601-1, encuadrándose en los estándares de seguridad y funcionalidad exigidos para ventiladores mecánicos. Además, el costo estimado del equipo es considerablemente menor en comparación con los modelos comerciales, sin comprometer su rendimiento técnico. Se resalta la integración de un sistema de alarmas y alertas visuales, lo que permite notificar al personal médico ante cambios críticos en los parámetros del paciente, dando así una respuesta inmediata. La implementación de este respirador artificial contribuiría a mejorar la capacidad del Hospital Distrital de Caaguazú para atender emergencias respiratorias, optimizando la disponibilidad de los dispositivos y facilitando la supervisión del paciente de manera eficiente.

Palabras clave: *Sistemas Mecatrónicos, Robótica, ODS 3 - Salud y bienestar, ODS, ODS 9 - Industria, Innovación e Infraestructura.*



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Abstract

The development of artificial respirators is essential to improve hospital response in critical situations. At the Caaguazú District Hospital, the shortage of these devices in the emergency area has created an urgent need for an optimized and real-time monitored design. This study aims to propose an artificial respirator with a control system based on embedded technologies, ensuring its operational efficiency and remote supervision of its ventilatory parameters. The methodology included data collection on the availability of equipment in the hospital, analysis of international regulations, and selection of appropriate electronic and mechanical components for system implementation. A computational model of the device was designed, integrating controllers, pressure and flow sensors, motors, and a graphical interface that allows variable configuration and real-time monitoring via a web platform. The results show that the proposed design complies with ISO 80601-2-12 and IEC 60601-1 standards, ensuring the safety and functionality required for mechanical ventilators. Additionally, the estimated cost of the device is considerably lower than commercial models, without compromising its technical performance. The integration of a visual and audible alarm system stands out, allowing medical personnel to be notified of critical changes in patient parameters, ensuring an immediate response. In conclusion, the implementation of this artificial respirator would help improve the Caaguazú District Hospital's capacity to handle respiratory emergencies, optimizing device availability and facilitating efficient patient monitoring.

Keywords: *Mechatronic Systems, SDG 3 - Good health and well-being, SDG 9 - Industry, Innovation and Infrastructure.*



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Índice

Introducción	1
1. Objetivos de investigación	4
1.1. Objetivo General.....	4
1.2. Objetivos Específicos	4
2. Metodología	5
2.1. Área de emergencias y respirador en uso	5
2.2. Tipos principales de ventilación asistida	6
2.2.1. Parámetros fundamentales de la ventilación asistida.....	7
2.2.2. Modelo óptimo y propuesta	8
2.3. Normativas aplicadas al diseño	10
2.3.1. Normativas paraguayas.....	10
2.3.2. Normativas internacionales.....	10
2.4. Selección de componentes.....	11
2.4.1. Sensores para el monitoreo de parámetros respiratorios	11
2.4.2. Controladores.....	12
2.4.3. Actuadores	13
2.4.4. Interfaz y alarmas.....	15
2.4.5. Fuente de alimentación y respaldo.....	15
2.4.6. Filtros de seguridad Biológica	17
2.4.7. Criterios de selección para el mecanismo	17
2.4.7.1. Mecanismo del sistema	19
2.5. Programación del sistema.....	20
2.6. Diagrama de conexiones y esquema.....	23
2.6.1. Circuito electrónico.....	23



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

2.6.2. Esquema general del respirador artificial	24
2.7. Programas utilizados	24
2.8. Análisis económico.....	25
3. Resultados y análisis	27
4. Conclusiones y recomendaciones.....	29
4.1. Recomendaciones	29
Referencias.....	30
Anexos.....	32
Anexo A. Diagramas y planos técnicos	32
Anexo A.1. Vista isométrica del respirador artificial propuesto.....	32
Anexo A.2. Vista explosionada del respirador con los principales componentes.....	32
Anexo A.3. Medidas generales del respirador	33
Anexo A.4. Mecanismo dual-finger (pinza)	34
Anexo A.5. Sujetador AMBU	35
Anexo A.6. Barra de soporte.....	36
Anexo A.7. Perfil de soporte.....	37
Anexo A.8. Mecanismo y polea.....	38
Anexo B. Diseño electrónico.....	39
Anexo B.1. Diagrama esquemático del respirador artificial.....	39
Anexo B.2. Representación electrónica completa del respirador artificial.....	40
Anexo B.3. Representación electrónica del sistema de conmutación de emergencia	40
Anexo B.4. Configuración de baterías en 4S4P con módulo BMS-4S-30A	41
Anexo B.5. Diagrama esquemático de conexión del motor pasos Nema 17, A4988 y Arduino Mega	41
Anexo C. Programación del sistema.....	42



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Anexo C.1. Programación en Arduino IDE	42
Anexo C.2. Programación de la Raspberry Pi 3 en Python	45
Anexo D. Programación de la interfaz local y web	48
Anexo D.1. Página de autenticación	48
Anexo D.2. Página de inicio	51
Anexo D.3. Página de configuración de variables	54
Anexo D.4. Página de gráficos y parámetros.....	57
Anexo E. Desglose presupuestario	62
Anexo E.1. Costo total de componentes electrónicos.....	62
Anexo E.2. Costos tentativos de materiales.....	63
Anexo E.3. Costo total de programación y mano de obra	63
Anexo E.4. Costo del certificado de autorización de la DINAVISA	63



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Índice de Tablas

Tabla 1. Características del motor a pasos Nema 17	14
Tabla 2. Consumo de corriente aproximado del sistema.....	16
Tabla 3. Ponderación de criterios y elección de mecanismo ideal para el sistema	18
Tabla 4. Costos de operación por mecanismo	18
Tabla 5. Resultados obtenidos para la selección del mecanismo	19
Tabla 6. Costos totales estimados para la implementación del respirador artificial.....	25
Tabla 7. Costo total de componentes electrónicos	62
Tabla 8. Costos tentativos de materiales	63
Tabla 9. Costo total de programación y mano de obra.....	63
Tabla 10. Costo del certificado de autorización de la DINAVISA.....	63



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Índice de Figuras

Figura 1. Curvas de operaciones esenciales [2]	9
Figura 2. Propuesta teórica para el diseño del respirador artificial	9
Figura 3. Esquema de filtro eléctrico tipo PS00XDH3A [6].....	11
Figura 4. Esquemático completo del MPX5010DP	12
Figura 5. Diagrama típico de conexionado entre la A4988 y un motor bipolar [13]	14
Figura 6. Diagrama de conexión del sistema de conmutación de emergencia.....	16
Figura 7. Tipos de accionamientos mecánicos	17
Figura 8. Mecanismo de tipo doble-dedo con accionador	19
Figura 9. Configuración del mecanismo y motor a pasos Nema 17, en un sistema de poleas para el accionamiento.....	20
Figura 10. Pantalla de autenticación	21
Figura 11. Pantalla de inicio.....	21
Figura 12. Pantalla parámetros a seleccionar	22
Figura 13. Pantalla gráficos en tiempo real.....	22
Figura 14. Esquema de conexión	23
Figura 15. Esquema interno del respirador y componentes	24
Figura 16. Vista isométrica del respirador artificial propuesto (en mm).....	32
Figura 17. Vista explosionada del respirador con los principales componentes.....	32
Figura 18. Medidas generales del respirador	33
Figura 19. Mecanismo tipo dual-finger (pinza)	34
Figura 20. Sujetador de AMBU.....	35
Figura 21. Barra de soporte	36
Figura 22. Perfil de soporte	37
Figura 23. Mecanismo y polea	38



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentido crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

Figura 24. Diagrama esquemático del respirador artificial	39
Figura 25. Representación electrónica completa del respirador artificial	40
Figura 26. Representación electrónica del sistema de conmutación de emergencia.....	40
Figura 27. Configuración de baterías en 4S4P con módulo BMS-4S-30A.....	41
Figura 28. Diagrama esquemático de conexión del motor pasos Nema 17, A4988 y Arduino Mega	41

Introducción

Los respiradores artificiales son dispositivos esenciales en hospitales y cualquier centro de salud públicos y privados, especialmente en unidades de emergencia y cuidados intensivos, donde pueden marcar la diferencia entre la vida y la muerte en pacientes críticos. Aunque la pandemia de COVID-19 ya no es una crisis sanitaria global, persisten enfermedades respiratorias derivadas de ella, además de afecciones crónicas y estacionales que requieren soporte de ventilación. También hay un aumento de pacientes accidentados que necesitan ventilación asistida, especialmente en épocas de inicio de clases, fin de año y fines de semana.

Este hospital, además de atender a la población local, recibe pacientes de distritos aledaños como Repatriación, Yhú, J. Eulogio Estigarribia y Vaquería. Sin embargo, la cantidad de respiradores artificiales disponibles es limitada. Según el Dr. Cristhian Ramón Vera, director de la V Región Sanitaria, la falta de estos dispositivos en el área de emergencia ha obligado a trasladar pacientes críticos al Hospital Regional de Coronel Oviedo, lo que implica un riesgo debido a la distancia y el tiempo de traslado.

Actualmente, el hospital solo dispone de cuatro cicladores modelo Puritan Bennet 560 y un quinto equipo que está en una ambulancia. En emergencias que requieren ventilación mecánica inmediata, el personal debe recurrir a dispositivos AMBU, que requieren la operación manual, lo que reduce la disponibilidad del personal para otros casos urgentes. La falta de mantenimiento en los equipos existentes agrava aún más la situación, ya que la asistencia técnica del Ministerio de Salud Pública y Bienestar Social no es rápida ni eficiente.

Para abordar esta problemática, se propone el diseño de un respirador artificial automatizado que se ajuste a las necesidades del hospital, y que incorpore un sistema de monitoreo remoto que optimice la supervisión de los parámetros de ventilación. Este desarrollo no solo busca mejorar la calidad de la atención, sino también reducir la carga de trabajo del personal médico y optimizar la respuesta en situaciones de emergencia

A nivel mundial se han realizado múltiples investigaciones para abordar la optimización de ventiladores mecánicos y así mejorar su accesibilidad y funcionalidad. En Paraguay, el Consejo Nacional de Ciencia y Tecnología (CONACYT) impulsó un equipo multidisciplinario para desarrollar ventiladores mecánicos nacionales, evaluados en la Agencia Espacial del Paraguay con ajustes regulatorios y económicos. Posteriormente, [1], [2] introdujeron el PytuTester, un dispositivo de código abierto basado en Raspberry Pi, diseñado para medir flujo, presión y oxígeno, lo que ha resultado útil en la validación de nuevos ventiladores.

Otras iniciativas han surgido en distintos países. Por ejemplo, se encuentra el diseño de [3], el cual realizó un banco de pruebas para dispositivos CPAP en Ecuador, utilizando ingeniería inversa para evaluar su desempeño. Por otra parte, [4] propuso un prototipo de respirador y oxímetro automático para uso domiciliario, mientras que [5] desarrollaron un ventilador portátil de estructura abierta, con control basado en tasa de flujo. Uno de los trabajos que sirvieron de principal pilar para el desarrollo del trabajo fue el de [6], el cual presentó un sistema híbrido que combina Arduino y Raspberry Pi 4 para integrar mediciones fisiológicas. Otras investigaciones han explorado nuevas tecnologías, como la aplicación IoT e inteligencia artificial para mejorar la telemonitorización de pacientes ventilados [7]. Finalmente, [8] diseñaron un ventilador de bajo costo utilizando componentes accesibles, destacando la importancia de sistemas asequibles en contextos hospitalarios con recursos limitados.

Como bien se ha mencionado anteriormente, este trabajo se justifica por la necesidad de ampliar la disponibilidad de respiradores en el Hospital Distrital de Caaguazú, mejorando así la capacidad de respuesta en situaciones de emergencia respiratoria. La propuesta de diseño incorpora funciones que optimizan su operatividad y facilitan la supervisión en tiempo real que, además de eso, la implementación de sistemas embebidos permitirá un monitoreo remoto, asegurando un ajuste dinámico de los parámetros de ventilación en relación a las condiciones del paciente.

Desde el punto de vista académico, este proyecto se enmarca en la línea investigativa de sistemas mecatrónicos, debido a que integra principios de la mecánica, electrónica y software para desarrollar un sistema de ventilación automatizado; la implementación de sensores, actuadores y algoritmos de control permite una administración correcta del flujo de aire, lo que asegura un ajuste ideal para cada paciente. La incorporación de una interfaz web permitirá una gestión eficiente del dispositivo, minimizando la necesidad de supervisión manual y aumentando dando una mejora en la eficiencia del personal médico.

El trabajo tiene como objetivo proponer un diseño de respirador artificial que pueda responder a las necesidades del centro médico, asegurando su operatividad mediante tecnologías modernas. Para ello, se han definido objetivos específicos orientados a identificar variables clínicas esenciales, selección de componentes adecuados, implementación de estrategias de control eficientes y el desarrollo de una plataforma digital para monitoreo remoto.

En términos generales, la ventilación mecánica es un procedimiento para el soporte respiratorio en pacientes con insuficiencia pulmonar. Los ventiladores mecánicos regulan parámetros como el volumen tidal, la presión inspiratoria pico (PIP) y la frecuencia respiratoria, permitiendo una oxigenación controlada. Existen dos tipos principales de ventilación asistida: controlada por volumen (VCV) y controlada por presión (PCV). En la VCV, el ventilador administra un volumen

de aire constante en cada respiración, mientras que en la PCV, se establece una presión límite para evitar lesiones pulmonares.

Para garantizar la seguridad del paciente, el trabajo contempla las normativas que debe cumplir, como la ISO 80601-2-12, que regula los estándares de seguridad en ventiladores mecánicos. De la misma forma, la implementación de filtros HME es importante para prevenir infecciones y así asegurar un sistema de ventilación higiénico.

El desarrollo del respirador artificial representa una solución innovadora para fortalecer la capacidad del Hospital Distrital de Caaguazú en la atención de emergencias respiratorias. La combinación de automatización, optimización de recursos y tecnologías accesibles permitirá mejorar la eficiencia del hospital y garantizar una atención médica eficiente.

1. Objetivos de investigación

1.1. Objetivo General

Proponer un diseño de respirador artificial que se ajuste a las necesidades específicas del área de emergencia del Hospital Distrital de Caaguazú, atendiendo a las especificaciones técnicas y configuraciones óptimas para su monitoreo mediante remoto.

1.2. Objetivos Específicos

- Identificar variables clínicas a monitorear de acuerdo a las normas de referencias seleccionadas.
- Definir los criterios que se aplicarán para el diseño del respirador artificial.
- Identificar las funcionalidades mínimas y especificaciones técnicas que deben considerarse para el diseño del respirador artificial.
- Evaluar las mejores alternativas tecnológicas en cuanto a materiales, dispositivos de potencia, sensores y control para el diseño del respirador artificial.
- Definir la configuración y programación adecuada para el correcto funcionamiento del respirador artificial.
- Explorar la viabilidad de desarrollar una aplicación web que permita el monitoreo remoto del respirador artificial.
- Realizar el presupuesto del diseño final del respirador artificial.

2. Metodología

La presente investigación se enmarcó dentro de un enfoque técnico y aplicado, orientado a la propuesta de diseño de un respirador artificial con monitoreo remoto. Se planteó como una investigación aplicada ya que se utilizó conocimientos previos en ingeniería para definir un sistema factible dentro del contexto hospitalario. Además de eso se complementó con un análisis bibliográfico, lo que permitió establecer fundamentos teóricos y tecnológicos basados en estudios previos (hechos antes, durante y después de la pandemia del COVID-19). Al detallar también parámetros esenciales del diseño, componentes y condiciones técnicas para la correcta implementación, el trabajo también se considera descriptiva.

El proceso metodológico inició con la recopilación de información sobre la disponibilidad de respiradores en el Hospital Distrital de Caaguazú, en donde se identificó, al realizar una visita técnica en la institución, la necesidad de mejorar y optimizar la atención médica en emergencias respiratorias. Para ello, se realizaron entrevistas con profesionales de la salud encargados del área de emergencia y una revisión documental sobre las tecnologías de ventilación mecánica utilizadas, enfocándose en dispositivos desarrollados entre 2020 y 2024.

Posteriormente, se hizo el análisis de requisitos y la definición de los aspectos técnicos fundamentales que debía cumplir la propuesta, como los parámetros de ventilación, los criterios de seguridad y la compatibilidad con normativas nacionales e internacionales. Seguidamente, se llevó a cabo la revisión de normativas y regulaciones aplicables, considerando los estándares internacionales ISO 80601-2-12 y normativas nacionales de dispositivos médicos, específicamente de los equipos de respiradores artificiales.

Con base en estos criterios, se establecieron los lineamientos técnicos para el diseño del respirador, priorizando aspectos como seguridad, usabilidad y viabilidad económica. Posteriormente, se realizó la búsqueda y selección de componentes adecuados para el diseño teórico del sistema, incluyendo sensores, microcontroladores y fuentes de alimentación. También, se definieron los protocolos de comunicación para la correcta interacción entre los distintos módulos del sistema y la plataforma de monitoreo remoto.

Por último, se elaboró un presupuesto estimado, detallando los costos asociados con los materiales y la posible implementación del dispositivo en el contexto hospitalario.

2.1. Área de emergencias y respirador en uso

En el área de emergencias del Hospital Distrital de Caaguazú se utilizan, actualmente, dos cicladores Puritan Bennett pb-560, los cuales hasta la última visita hecha, se verificó la realización

del correspondiente mantenimiento preventivo a uno de los dispositivos, estando disponible solo uno. Esto es especialmente alarmante cuando se presentan picos de saturación de pacientes en el área, lo que aumenta el trabajo manual de los médicos a la hora de brindar soporte a los pacientes.

El Puritan Bennet 560 es un ventilador mecánico portátil diseñado para proporcionar soporte respiratorio tanto a pacientes adultos como pediátricos. Entre sus características se encuentran su diseño compacto y ligero, lo que facilita su transporte y uso, además de que cuenta con una batería interna recargable que ofrece hasta 11 horas de funcionamiento continuo.

Presenta diversos modos de ventilación, incluyendo ventilación asistida/controlada por volumen (VAC), ventilación asistida/controlada por presión (VP A/C), Ventilación mandatoria intermitente sincronizada por volumen (V SIMV), ventilación mandatoria intermitente sincronizada por presión (P SIMV), presión positiva continua en la vía aérea (CPAP) y ventilación con soporte de presión con frecuencia de respaldo (PSV/ST).

Un punto a mencionar es que, según la Agencia Española de Medicamentos y Productos Sanitarios (AEMPS), estos dispositivos tienen una vida útil de 10 años, y su uso acumulativo más allá de 14 años puede ser potencialmente perjudicial para la salud del paciente

2.2. Tipos principales de ventilación asistida

La ventilación mecánica es un procedimiento esencial en el soporte respiratorio de pacientes con insuficiencia respiratoria, el cual asegura un intercambio adecuado de gases en los pulmones. Existen diversos tipos de ventilación que se diferencian de acuerdo a los parámetros que controlan y de la participación del paciente en el proceso respiratorio.

Uno de los tipos más utilizados es la ventilación mandatoria continua (CMV, por sus siglas en inglés), en la cual todas las respiraciones son generadas por el ventilador, ya sean iniciadas por el dispositivo o por el paciente. Como lo indica [9], este tipo de ventilación se emplea en pacientes que no pueden respirar de manera espontánea o que requieren una asistencia total en su función respiratoria.

Otra modalidad importante es la ventilación controlada por volumen (VC), donde el ventilador administra un volumen de aire preestablecido en cada ciclo respiratorio, el cual es independientemente de la presión generada en las vías respiratorias. Esta modalidad garantiza una ventilación constante y es especialmente útil en pacientes con insuficiencia respiratoria grave [10]. Sin embargo, su principal desventaja es que puede generar presiones excesivas en los pulmones si la mecánica ventilatoria del paciente cambia.

Por otro lado, tenemos la ventilación controlada por presión (PC), el cual proporciona un nivel de presión inspiratoria constante en cada respiración, lo que permite un flujo de aire variable y adaptado a la demanda del paciente. La principal ventaja de este tipo de ventilación radica en que radica en la reducción del riesgo de lesión pulmonar inducida por presión excesiva.

Una variante avanzada de estas modalidades es la ventilación con control de volumen regulado por presión (PRVC), que combina las ventajas de ambas estrategias. En este caso, el ventilador ajusta automáticamente la presión inspiratoria y así garantiza la entrega de un volumen tidal específico, minimizando la posibilidad de barotrauma [10].

Además de estas modalidades básicas, existen otras estrategias como la ventilación con presión positiva al final de la espiración (PEEP), que se utiliza para evitar el colapso alveolar, y la ventilación asistida, donde el ventilador solo proporciona soporte cuando el paciente inicia un esfuerzo inspiratorio.

Para el diseño de un respirador artificial, la selección del tipo de ventilación es una parte primordial, debido a que debe adaptarse a las necesidades clínicas del paciente y a las limitaciones del equipo.

Se priorizan modos de ventilación que brinden estabilidad respiratoria, minimicen los riesgos de complicaciones pulmonares, los cuales pueden darse en caso de que el sistema no cumpla con las debidas indicaciones y/o normativas pertinentes, y sean compatibles con sistemas de monitoreo remoto.

2.2.1. Parámetros fundamentales de la ventilación asistida

La ventilación mecánica asistida se basa en una serie de parámetros que garantizan su correcto funcionamiento y permiten su adaptación a las necesidades del paciente. Existen ventiladores mecánicos configurados con alta tecnología, pero todos comparten funciones básicas tales como el control de inicio, que regula cuándo y cómo comienza la respiración asistida, la limitación del flujo de aire, que define la cantidad de gas suministrado a los pulmones en cada ciclo, y el ciclado, que, según [3], determina la duración del ciclo respiratorio.

Uno de los parámetros más relevantes es el volumen tidal (VT), que corresponde a la cantidad de aire entregado en cada inspiración, expresado en mililitros por kilogramo de peso corporal. El volumen minuto (VE), por su parte, se refiere al total de aire movilizado por minuto, producto del volumen tidal y la frecuencia respiratoria. Otro aspecto crítico es la presión inspiratoria máxima (PIP), que define el límite de presión generado en la vía aérea durante la inspiración y debe mantenerse en rangos seguros para evitar lesiones pulmonares.

La presión positiva al final de la espiración (PEEP) es otro parámetro esencial, ya que evita el colapso alveolar al mantener una presión residual en las vías respiratorias al final de la espiración. La PEEP se ajusta según la necesidad del paciente, con valores típicos entre 5 y 20 cmH₂O. De todas formas, la relación inspiración-espiración (I:E) define el tiempo relativo de cada fase del ciclo respiratorio y varía según el tipo de ventilación [2].

El monitoreo de la fracción inspirada de oxígeno (FiO₂) es importante para asegurar una oxigenación adecuada sin inducir toxicidad por oxígeno. En entornos clínicos, los ventiladores mecánicos permiten ajustar este parámetro entre el 21 %, el cual corresponde al oxígeno ambiental, y el 100 % en situaciones críticas. El empleo de sensores de flujo y presión para detectar alteraciones en la mecánica respiratoria y ajustar la ventilación de manera óptima es también un factor importante en cualquier sistema de ventilación.

Por otra parte, en dispositivos de ventilación basados en sistemas Ambu motorizados, como es el caso de este trabajo, se requieren elementos específicos para su correcto ensamblaje; según [3], estos corresponden a un mezclador de aire y oxígeno, reguladora de presión de precisión, válvula reguladora de presión, motor de pasos NEMA 17, manifold de inhalación, sensores de presión y de oxígeno, electroválvulas, filtros EPA, fuente de poder de 12VDC, placa controladora y relés. Estos permiten un control eficiente de la ventilación y aseguran la estabilidad del paciente durante su uso.

2.2.2. Modelo óptimo y propuesta

Desde el punto de vista funcional, el respirador debe ser de uso intuitivo y simple, con montaje seguro de mangueras, imposibilidad de conexiones incorrectas y facilidad de limpieza. Además, debe ser liviano y transportable, permitiendo su movilidad dentro de la unidad de emergencias o terapia intensiva [4].

En términos técnicos, el modelo debe integrar una pantalla táctil LCD que permita el monitoreo y control de parámetros respiratorios como la frecuencia respiratoria, volumen tidal, relación inspiración-espiración (I:E), presión inspiratoria máxima (PIP), presión positiva al final de la espiración (PEEP), flujo inspiratorio y fracción de oxígeno inspirada (FiO₂). También debe soportar modos de ventilación controlados por volumen y presión, ajustables según la condición clínica del paciente. En la Fig. 1 se muestran las principales curvas que debe proporcionar el ventilador mecánico, según [2].

El sistema debe contar con un sistema de alarmas integrado, capaz de detectar desconexiones, baja PEEP, presión excesiva y fallas energéticas. Para ello, se estableció la necesidad de una fuente de alimentación dual, funcionando tanto con corriente alterna (220V, 50Hz) como con baterías

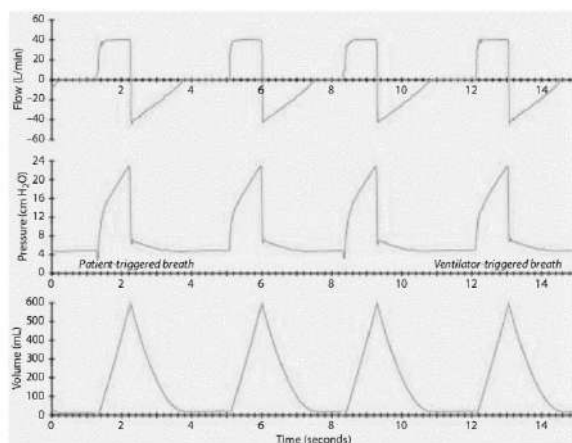


Figura 1. Curvas de operaciones esenciales [2]

recargables; para el trabajo propuesto, la fuente de alimentación primaria consiste en una fuente de voltaje de 12 V a 5 A, y la fuente de respaldo está constituido por baterías 9 baterías de litio 18650 (3.7V hasta 4.2V cada una, con una autonomía de 2000 mAh), en configuración 4S4P, lo que garantiza una autonomía mínima de 2.7 horas ante cortes de energía. Este apartado será explicado más adelante, junto con el circuito de conmutación.

De igual forma, el modelo debe especificar la compatibilidad con filtros bacterianos y virales de alta eficiencia, válvulas y mangueras libres de látex y esterilizables, de acuerdo a la norma ISO 10993.

Además de los estándares propuestos y para mejorar la supervisión, se propone la integración de una plataforma web adaptable a diferentes dispositivos móviles, lo que permite monitorear los parámetros respiratorios y alertas del sistema en tiempo real. En la Fig. 2 se presenta la propuesta teórica del respirador artificial a ser diseñado, presentando todos los componentes de manera general.

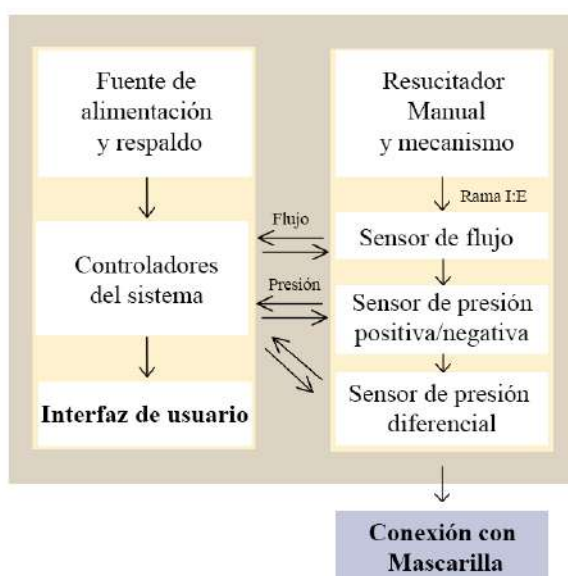


Figura 2. Propuesta teórica para el diseño del respirador artificial

2.3. Normativas aplicadas al diseño

El diseño del respirador artificial debe cumplir con estándares normativos tanto nacionales como internacionales, enfocándose en la seguridad del paciente, la compatibilidad electromagnética (siendo este un problema que pueda acarrear diversos problemas si no se maneja adecuadamente) y la funcionalidad del equipo en entornos hospitalarios. Para esto, se investigó las regulaciones que establecen requisitos de bioseguridad, alarmas, alimentación energética y fiabilidad operativa.

2.3.1. Normativas paraguayas

A nivel nacional, la Ley N° 1119/1997 regula la fabricación e importación de dispositivos médicos, en donde se exige que estos cumplan con estándares de calidad certificados por el Ministerio de Salud Pública y Bienestar Social (MSPBS). Complementando esta regulación se encuentra la Resolución S.G. N° 669/2016, el cual establece los requisitos para la obtención del registro sanitario, asegurando la trazabilidad y certificación de los equipos médicos antes de su comercialización y uso en hospitales.

Igualmente, se encuentra la Ley N° 4659/12, en su Artículo 13, que prohíbe expresamente la adquisición y uso de dispositivos médicos no certificados en hospitales públicos y privados, protegiendo tanto a pacientes como al personal de salud. Esta ley es de vital importancia a tener en cuenta para la fabricación e implementación del diseño de un respirador, debido a que es obligatorio el cumplimiento de la misma y a aprobación correspondiente del MSPBS para el uso hospitalario de estos dispositivos médicos

Para asegurar la seguridad eléctrica del respirador, la fuente de alimentación recomendada debe ser de grado médico y permitir la conversión de 110–240 V AC a 12 V DC, cumpliendo con los medios de protección (MOP) establecidos por la norma IEC 60321-1 C14.

2.3.2. Normativas internacionales

La ISO 80601-2-12 establece los requisitos de seguridad específicos para ventiladores médicos, regulando aspectos como la protección contra fallos en el suministro de oxígeno, ajustes accidentales y compatibilidad con filtros bacterianos y virales. También menciona la integración de alarmas críticas para eventos como desconexión, baja presión PEEP y fallo energético [5]. Todos estos puntos mencionados en la norma serán cubiertos más adelante en el trabajo, mencionando explicando cada componente a utilizar para el respirador artificial.

La IEC 60601-1-2 se enfoca en la compatibilidad electromagnética (EMC), asegurando que el dispositivo no interfiera ni sea afectado por otros equipos eléctricos del entorno hospitalario. Para cumplir con esta normativa, la ventilación mecánica requiere filtros de interferencia

electromagnética (EMI) que impidan fluctuaciones de energía y minimicen la corriente de fuga, reduciendo el riesgo de choques eléctricos. Para el desarrollo del trabajo, se optó por la recomendación brindada por [6] para la implementación de un filtro EMI (Fig. 3).

Otro punto importante a mencionar es que para el uso del filtro EMI es debido a la baja calidad de la energía eléctrica en hospitales, lo cual hace indispensable su uso en la entrada del equipo médico, cumpliendo con la norma IEC 60321-1 C14, de grado médico.

Por otro lado, el Reglamento (UE) 2017/745 establece criterios para la gestión de riesgos en dispositivos médicos, enfatizando la seguridad durante todo el ciclo de vida del producto [7]. Asimismo, la IEC 60601-1-8 especifica los requisitos de alarmas sonoras y visuales para alertar sobre eventos críticos, asegurando una respuesta rápida del personal de salud [8].

2.4. Selección de componentes

2.4.1. Sensores para el monitoreo de parámetros respiratorios

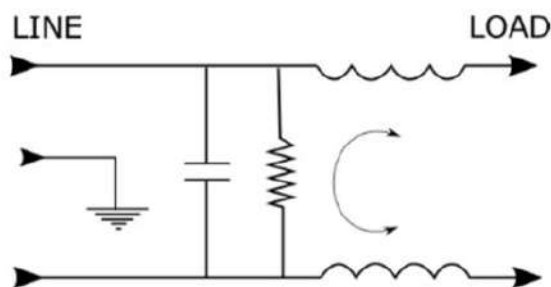


Figura 3. Esquema de filtro eléctrico tipo PS00XDH3A [6]

El monitoreo de los parámetros respiratorios del ventilador mecánico es esencial para asegurar el suministro adecuado de oxígeno y prevenir complicaciones en el paciente. Por lo anterior, se eligieron sensores específicos que permiten medir la presión, el flujo de aire y las variaciones en la ventilación asistida.

Se utilizó el sensor de presión diferencial MPX5010DP (Fig. 4) para medir la presión en las líneas inspiratoria y espiratoria del sistema. Este sensor es común en dispositivos médicos debido a su alta sensibilidad y precisión, lo que permite detectar variaciones en la presión en un rango de 0 a 10 kPa. A diferencia de otros sensores de presión de tipo MPX, este permite procesar las señales enviadas de manera adecuada, ya que su tensión de salida se encuentra en el rango de 0.2V a 4.7V [9]. Su integración en el circuito de control del respirador facilita la calibración del volumen tidal

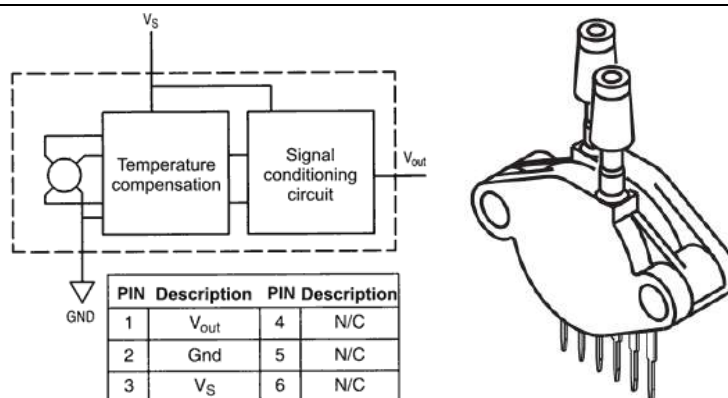


Figura 4. Esquemático completo del MPX5010DP

y la presión positiva al final de la espiración (PEEP), que son parámetros críticos para la ventilación asistida.

Para complementar el monitoreo de la presión, se incorporó el sensor de presión absoluta XGZP6847, que mide la presión total en el circuito respiratorio. Este sensor es especialmente utilizado en el campo médico, así como en equipos de terapia, monitoreo, entre otros. Así como el MPX5010P, su tensión de salida es del rango de 0.5 - 4.5V [10]. Esto es crucial para ajustar la presión inspiratoria máxima (PIP) y asegurar que el flujo de aire suministrado sea seguro y efectivo para el paciente.

En cuanto a la medición del flujo de aire, se eligió el sensor YFS201, que permite medir la cantidad de gas rico en oxígeno que ingresa al paciente [6]. Este sensor es adecuado para aplicaciones estándar de ventilación mecánica, proporcionando una lectura confiable del volumen administrado en cada ciclo respiratorio. Sin embargo, para entornos clínicos que requieran mayor precisión y control (terapias intensivas, por ejemplo), se recomienda el uso de sensores de grado médico, como el FS6122 o el PMF4101V, que ofrecen mayor exactitud y estabilidad en la medición del flujo. No obstante y como indica [6], su implementación requiere un acondicionamiento de señal adicional, amplificando y calibrando la salida del sensor (8–40 mV) para que sea detectada correctamente por los sistemas embebidos.

2.4.2. Controladores

Una de las partes más importantes de todo sistema es el procesamiento de los datos y, para una ventilación asistida eficiente y adaptable a las necesidades del paciente, se integraron dos tipos de controladores, los cuales son la Raspberry Pi 3 y el Arduino Mega, cada uno con funciones específicas dentro del ventilador.

La Raspberry Pi 3 fue seleccionado como el controlador principal debido a su capacidad de procesamiento y manejo de interfaces gráficas en tiempo real [6], [11]. Este dispositivo, basado en un procesador Broadcom BCM2837 SoC, Cortex - A53 de 64 bits (ARMv8) a 1.4 GHz con 1

GB de LPDDR2 SDRAM, posibilita la visualización y monitoreo de los parámetros respiratorios a través de una interfaz gráfica de usuario (GUI). Además, su capacidad para gestionar múltiples procesos y comunicaciones mediante conexión USB y Ethernet la hace ideal para el control remoto y la supervisión del sistema vía web.

El Arduino Mega, por su parte, se empleó como controlador secundario, encargado de la adquisición de datos de sensores y la ejecución de las acciones en tiempo real dentro del sistema [11]. Debido a que la Raspberry Pi está optimizada para procesamiento gráfico y gestión de datos, pero no para la lectura eficiente de sensores y accionamiento de actuadores, el Arduino Mega se encarga de la lectura y procesamiento de los sensores de presión, flujo y oxígeno, asegurando tiempos de respuesta rápidos en la activación de válvulas y motores. Además de lo anterior, el Arduino se encarga de controlar el actuador principal del sistema (Nema 17).

Para una comunicación eficiente entre ambos controladores, la Raspberry Pi se conecta al Arduino Mega utilizando el protocolo UART (a través de USB), el cual, como indica [12], “se basa en la comunicación de datos en serie de manera asíncrona”, es decir, los datos son enviados a través de una sola línea que conecta dos dispositivos, sin necesidad de un reloj compartido por ambos para sincronizar la recepción y envío de datos. El Arduino Mega se comunica con otros módulos a través de los pines Serial Clock (SCL) y Serial Data (SDA), además de los pines digitales y analógicos. Esta arquitectura mejora la estabilidad del sistema evitando retrasos en la generación de señales de control y garantizando la precisión en la entrega de aire al paciente.

En proyectos avanzados, se recomienda el uso de sistemas embebidos dedicados para tareas críticas, como ATMEGA328P o ATTINY85 [11], que pueden complementar la Raspberry Pi al mejorar la precisión del control de señales PWM para las válvulas proporcionales.

La combinación de estos dos sistemas embebidos permite un equilibrio entre el procesamiento de datos y la gestión de la interfaz gráfica, lo cual asegura una respuesta rápida ante cualquier variación en los parámetros respiratorios y así facilitar el monitoreo remoto del paciente.

2.4.3. Actuadores

Para el diseño del respirador artificial, se tuvo especial atención en componentes mecánicos robustos y confiables, que pueda garantizar un funcionamiento eficiente, además de seguro. En principio, se consideró el uso de electroválvulas para controlar el flujo de aire en los ciclos de ventilación, pero finalmente, debido al costo extra que conllevaría adquirir estos dispositivos, se optó por emplear las válvulas estándar de inspiración y espiración incluidas en el sistema Ambu, reduciendo así los costos y sin comprometer la funcionalidad del equipo.

Para el actuador principal, se optó por motor paso a paso NEMA 17, puesto que ofrece precisión y control sobre los volúmenes de aire administrados en cada ciclo respiratorio. Este motor cuenta con 200 pasos por revolución y una precisión del $\pm 5\%$, permitiendo un control preciso de la compresión del AMBU. Además, su resistencia térmica y mecánica lo hacen ideal para aplicaciones de uso prolongado que, para este caso, es un entorno médico. En la Tabla 1 se presentan las principales características del NEMA 17, donde se destaca su par de retención de 0.83 Nm, lo que asegura estabilidad en el posicionamiento del motor.

Característica	Valor
Pasos por revolución	200
Precisión	$\pm 5\%$
Corriente nominal	2:00 a. m.
Torque de retención	0.83 Nm
Voltaje de operación	12-24 V
Temperatura de operación	-20°C a +40°C

Tabla 1. Características del motor a pasos Nema 17

Para controlar el NEMA 17 se utilizó el driver A4988, el cual es un controlador de micropasos muy utilizado para este tipo de motores y que permite operar al actuador en modos de full-step, 1/2, 1/4, 1/8 y 1/16 de paso [13]. Su arquitectura optimiza el consumo energético y mejora la precisión del movimiento, lo que reduce el ruido y la vibración. En la Fig. 5 se muestra el diagrama típico de conexión del A4988, mismo esquema que se utiliza para este trabajo.

Utilizar el NEMA 17 junto con el A4988 permite un control óptimo y por más preciso del flujo de aire, haciendo que el volumen administrado sea constante y ajustable, según las necesidades del paciente.

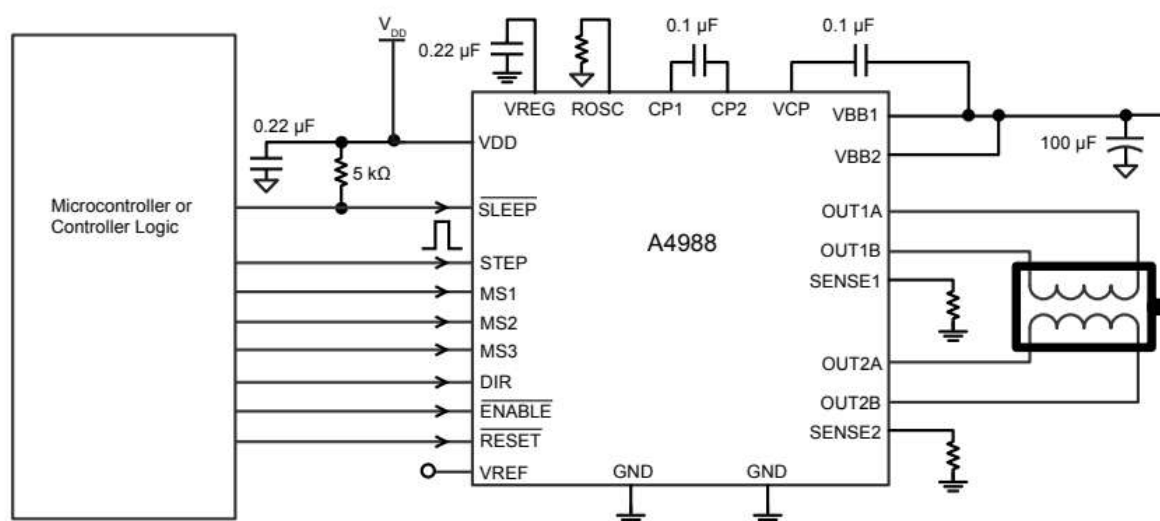


Figura 5. Diagrama típico de conexionado entre la A4988 y un motor bipolar [13]

2.4.4. Interfaz y alarmas

El sistema incorpora una interfaz gráfica y un sistema de alarma para la operación intuitiva y segura por parte de los personales médicos. La visualización de los parámetros respiratorios se realiza a través de la Raspberry Pi Touch Display, el cual es una pantalla LCD táctil de 7 pulgadas con resolución 720×1280 píxeles, desarrollado también por la Raspberry Pi Foundation, y que permite la interacción mediante controles táctiles sin la necesidad de periféricos adicionales. Este dispositivo facilita el ajuste de los modos de ventilación y la supervisión en tiempo real del estado del paciente.

En cuanto a la seguridad del sistema, se integró un buzzer piezoeléctrico que emite señales acústicas en caso de eventos críticos, como lo son los fallos en la alimentación, desconexión del circuito respiratorio o valores anómalos de presión y flujo de aire. La incorporación de alarmas sonoras responde a los requisitos de la norma IEC 60601-1-8, el cual establece la necesidad de alertas sonoras en equipos médicos para asegurar una respuesta rápida del personal sanitario ante emergencias [8].

Cabe mencionar también que el respirador cuenta con un pulsador de emergencia de 22 mm que permite al operador detener el sistema de inmediato en caso de anomalías. Con esto, se mejora la seguridad operativa del sistema y brinda mayor control sobre el dispositivo en situaciones de riesgo.

2.4.5. Fuente de alimentación y respaldo

El diseño del sistema de alimentación del respirador artificial se centró en una fuente de energía estable, segura y con respaldo en caso de fallas en la red eléctrica del hospital, como suele ocurrir con frecuencia en nuestro país. Para ello se implementó una fuente conmutada de 12V y 10A, capaz de proporcionar la potencia necesaria para el funcionamiento del dispositivo médico, incluyendo el consumo de los controladores, sensores y actuadores anteriormente mencionados. Según las especificaciones técnicas de todos los componentes activos, el consumo total del sistema es de aproximadamente 3.8 A en operación estándar, como se puede observar en la Tabla 2, el cual asegura que la fuente seleccionada tenga un margen de seguridad adecuado.

Modelo/ Especificación	Consumo de corriente mA
Raspberry Pi 3	1100 mA
Arduino Mega	700 mA
MPX5050DP	10 mA
YFS 201	15 mA
XGZP6847	15 mA
Buzzer KY-006	25 mA
Pulsador de Emergencia 22mm c/ Traba 1NC	20 mA
Potenciómetro 10k Ω	10 mA

NEMA 17	2000 mA
Total de corriente aproximada	3800 mA

Tabla 2. Consumo de corriente aproximado del sistema

Para proteger la integridad del sistema ante interferencias en la red eléctrica, se incorporó un filtro EMI cumpliendo con la norma IEC 60321-1 C14, el cual evita que fluctuaciones de energía afecten la estabilidad del equipo.

En caso de ocurran interrupciones en la alimentación principal, como cortes de energía o desconexiones accidentales, el sistema cuenta con un respaldo basado en baterías de ion-litio 18650, configuradas en formato 4S4P (4 en serie y 4 en paralelo), los cuales proporcionan un voltaje total de 14.8V y una capacidad de 10.4 Ah, permitiendo una autonomía mínima de 2.7 horas, aproximadamente. Para la gestión segura de las baterías, se implementó un sistema de protección BMS-4S-30A, encargado de evitar sobrecargas, sobrecorrientes y descargas profundas, lo que asegura la vida útil del banco de baterías.

Un punto importante a mencionar en esta sección es el cambio entre la fuente principal y las baterías; este se realiza mediante un circuito conmutador YX850, diseñado para realizar conmutaciones de emergencia entre una fuente de energía primaria a otra. En la Fig. 6, se presenta el diagrama de conexión del conmutador y la integración del sistema de baterías, utilizando el software de diseño Fritzing.

Como puede observarse en el diagrama de conexiones, luego de pasar por el la fuente conmutada AC/DC, se conecta a dos entradas, DC+ y DC-, del módulo YX850, mientras que la fuente las baterías de litio, en configuración 4S4P, se conectan al módulo BMS-4S-30A (de acuerdo a la

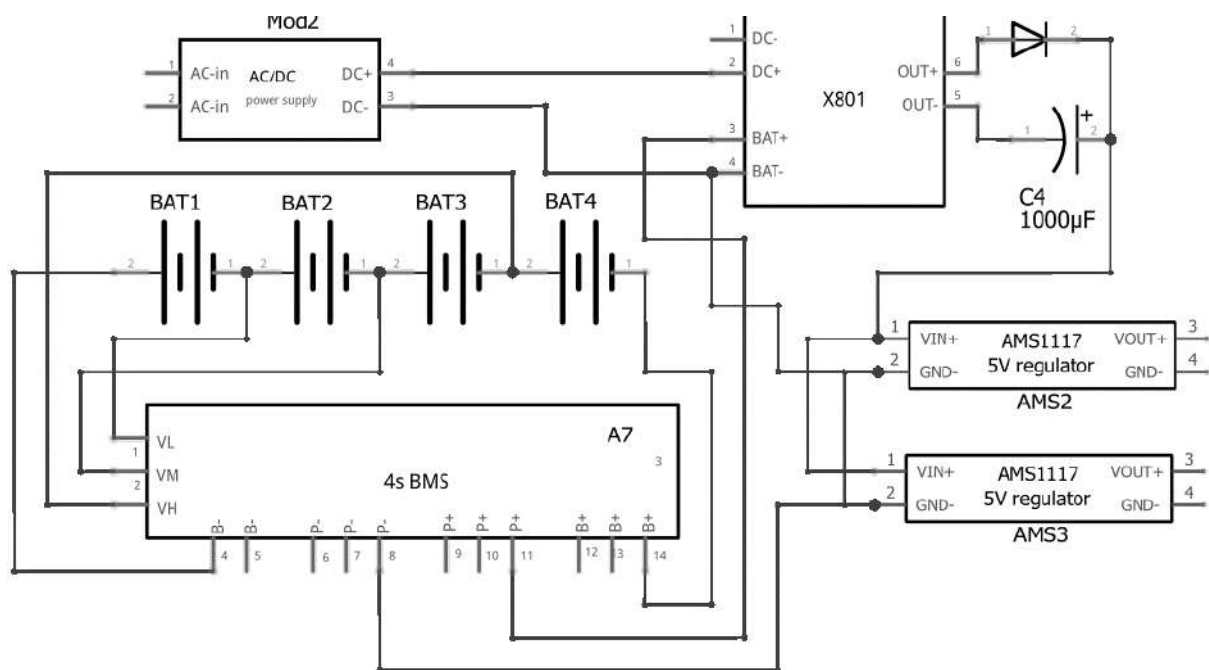


Figura 6. Diagrama de conexión del sistema de conmutación de emergencia

configuración indicada de 0V, 4.2V, 8.4V y 12.6 V), y la salida de esta se conecta a los otros BAT+ y BAT- del conmutador de emergencia; mientras no se realice conmutación alguna, el módulo YX850 pasará corriente de al sistema, al mismo tiempo que carga las baterías a través del módulo BMS. Esta arquitectura permite una alimentación constante y adaptable a condiciones de emergencia, lo que asegura que el respirador artificial continúe operando de manera segura ante fallos eléctricos inesperados.

2.4.6. Filtros de seguridad Biológica

El diseño del respirador artificial incorpora filtros y medidas de seguridad biológica para asegurar la protección del paciente y del equipo. La normativa ISO 80601-2-12 establece la necesidad de utilizar filtros bacterianos y virales en los circuitos respiratorios para evitar la contaminación cruzada y proteger el sistema de ventilación. Los filtros empleados son de tipo HME (filtros intercambiadores de calor y humedad, por sus siglas en inglés), muy utilizados en el entorno médico.

El sistema de ventilación se basa en un resucitador manual reutilizable (AMBU) modelo Masc. Reusable N.º 5 con reservorio de 2.5 L ARG 3896, el cual tiene una capacidad de 1600 ml y está diseñado específicamente para respiración artificial manual y resucitación cardiopulmonar. Su integración en el diseño del respirador permite un flujo de aire controlado y una ventilación eficiente en situaciones de emergencia.

2.4.7. Criterios de selección para el mecanismo

Una parte vital del respirador artificial es la manera en la que debe ser accionado el resucitador manual (AMBU). Para esto, se evaluaron distintos mecanismos propuestos con el objeto de

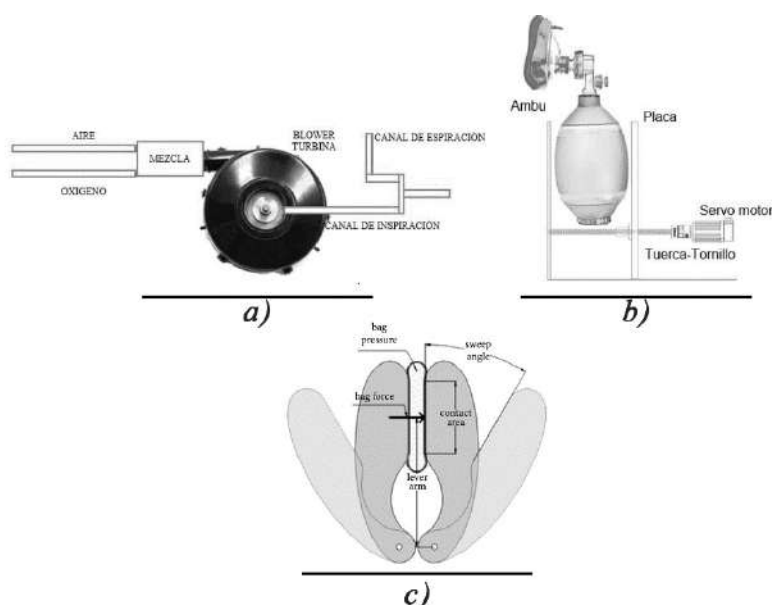


Figura 7. Tipos de accionamientos mecánicos

seleccionar la mejor opción en términos de precisión, costo y compatibilidad con el sistema. Entre las alternativas consideradas se incluyeron: (1) el uso de un sistema de turbina centrífuga (Fig. 7a), (2) un mecanismo de motor-tornillo-placa (Fig. 7b), y (3) un sistema de compresión con actuadores mecánicos tipo "dedos" (Fig. 7c),

Cada mecanismo fue analizado de acuerdo a criterios como la estabilidad en la compresión, la seguridad, el costo de fabricación y la facilidad de integración con el sistema de control. Para la elección del mecanismo, se utilizó el método ordinal corregido de criterios ponderados; se aplican las prioridades asignadas a cada criterio y se compara entre alternativas. En este caso, los criterios a tener en cuenta son los siguientes:

- Diseño Compacto (DC): 0.10
- Funcionamiento Sencillo (FS): 0.10
- Alta Durabilidad (AD): 0.15
- Costo de Operación (CO): 0.10
- Costo de Manufactura (CM): 0.15
- Seguridad (S): 0.15
- Compatibilidad Electromagnética (EMC): 0.10
- Energía y Respaldo (EE): 0.10
- Cumplimiento Normativo (NO): 0.05

Cada mecanismos será evaluado con una puntuación comprendida entre el 0 al 1 (por criterio), el cual se multiplica por la ponderación asignada. En la Tabla 3 se observan las ponderaciones hechas y los puntajes asignados.

Criterios	Ponderación	Turbina Centrifuga	Actuador dual-finger	Tornillo y Placa
DC	0.10	0.09	0.09	0.07
FS	0.10	0.10	0.07	0.06
AD	0.15	0.10	0.15	0.15

Tabla 3. Ponderación de criterios y elección de mecanismo ideal para el sistema

Criterios	Ponderación	Turbina Centrifuga	Actuador dual-finger	Tornillo y Placa
Costo de Operación (CO)	0.10	0.08	0.07	0.06
CM	0.15	0.14	0.13	0.10
S	0.15	0.12	0.15	0.15
EMC	0.10	0.09	0.09	0.07
EE	0.10	0.10	0.09	0.06
NO	0.05	0.04	0.05	0.03

Tabla 4. Costos de operación por mecanismo

Al realizar el cálculo de los puntajes totales correspondientes a los diseños propuestos, se obtuvieron los resultados mostrados en Tabla 5.

Mecanismo	Puntaje Total
Turbina centrífuga	0.86
Actuador dual-finger (pinza)	0.89
Tornillo y Placa	0.75

Tabla 5. Resultados obtenidos para la selección del mecanismo

El mecanismo de actuador dual-finger (pinza) obtuvo el puntaje más alto, con 0.89, destacándose en durabilidad, seguridad, y diseño eficiente. Además de lo anterior y un análisis hecho por [14], este tipo de mecanismo permite una compresión más natural y controlada del AMBU, ya que replica la acción manual de un operador. Otro punto a destacar es que este mecanismo reduce el estrés mecánico sobre el resucitador, prolongando la vida útil del mismo y asegurando una ventilación estable.

2.4.7.1. Mecanismo del sistema

De acuerdo a lo anteriormente mencionado, el mecanismo ideal para el sistema es el actuador dual-finger (pinza) (Fig. 8); la estructura está conformada por dos brazos articulados sincronizados mediante engranajes de dientes rectos, lo cual asegura un movimiento coordinado y estable. Las medidas de este mecanismo están detalladas en el Anexo A.

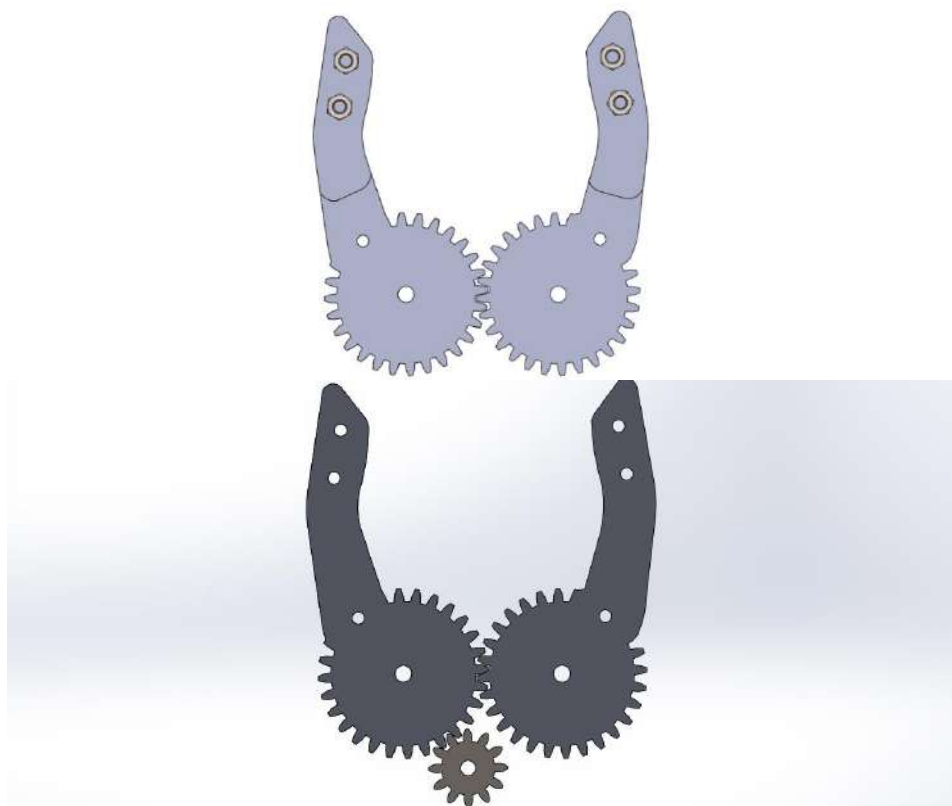


Figura 8. Mecanismo de tipo doble-dedo con accionador

Según [15], el ángulo de compresión de las pinzas influye directamente en el volumen de aire suministrado al paciente. Para la optimización de este proceso, se justifica el uso del mecanismo de engranajes interconectados para la compresión del AMBU, evitando deformaciones irregulares que podrían afectar el volumen tidal.

Para la implementación del motor a paso Nema, se utilizó una rueda dentada de 12 dientes y de radio igual a 14.24 mm. Este se conecta con un eje que, a su vez, se interconecta con una polea de 60 dientes, el cual es accionado por una correa conectada al cabezal del nema 17, lo que asegura un correcto movimiento del sistema de engranajes. En la Fig. 9 se pueden observar la configuración descrita anteriormente.

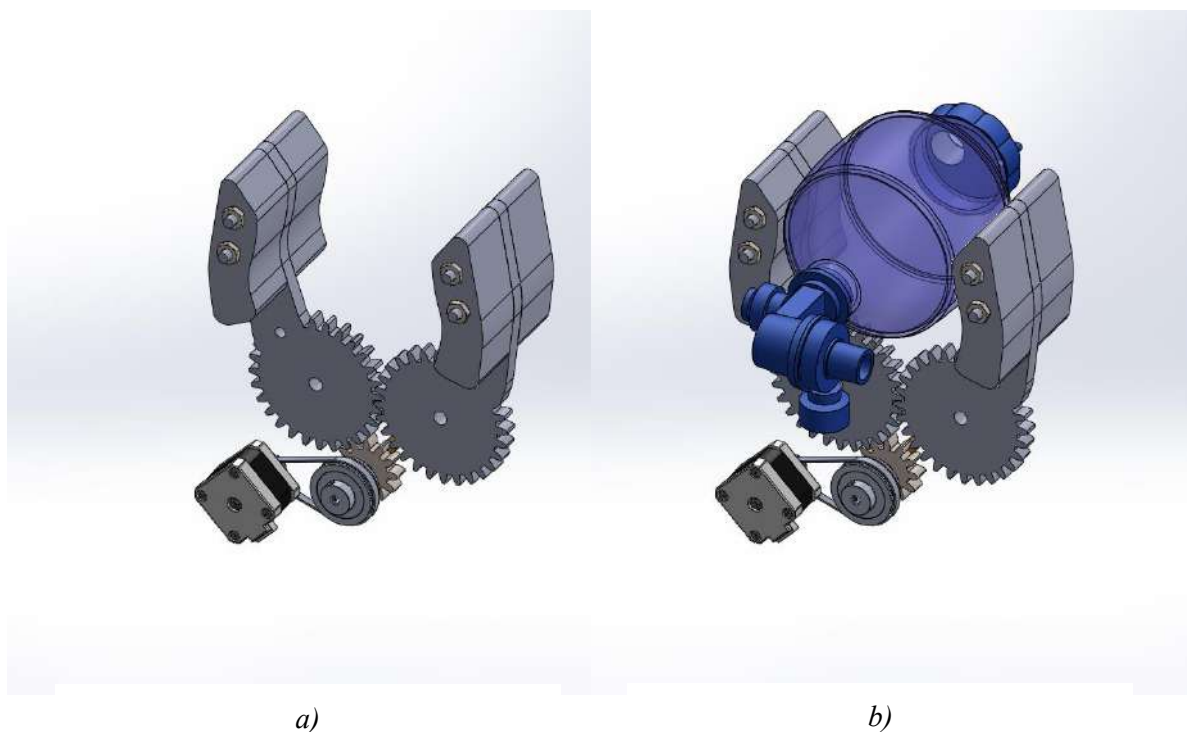


Figura 9. Configuración del mecanismo y motor a pasos Nema 17, en un sistema de poleas para el accionamiento

2.5. Programación del sistema

El software del respirador artificial ha sido diseñado para ofrecer una gestión eficiente de la ventilación asistida, donde se integró el Arduino Mega y Raspberry Pi 3 en un sistema que asegura el monitoreo en tiempo real, el control preciso, además de la seguridad operativa. Como bien se indicó en el apartado 2.3.2, la Raspberry Pi se encarga de la interfaz gráfica y la comunicación con el usuario, mientras que el Arduino Mega ejecuta las funciones de control y procesamiento de los sensores y actuadores. Para empezar, el sistema inicia con una pantalla de autenticación, esto para asegurar que solo personal autorizado pueda modificar los parámetros del ventilador (Fig. 10).



Figura 10. Pantalla de autenticación

Una vez dentro, se muestra una interfaz de inicio para ingresar datos del pacientes, fecha y algunas descripciones (Fig. 11).



Figura 11. Pantalla de inicio

A continuación se tiene la interfaz principal para la definición de parámetros, lo que permite configurar variables específicas, como modo de ventilación (VCV o PCV), volumen tidal (VT), frecuencia respiratoria (FR) y presión inspiratoria máxima (PIP) (Fig. 12). Estos parámetros son enviados al Arduino Mega, que ajusta el funcionamiento del actuador principal en base a los valores configurados.

Definición de Variables Respiratorias

MODO DE VENTILACIÓN: VCP

FR (Frecuencia Respiratoria) [12 - 20 rpm]: 15

VT (Volumen Tidal) [300 - 800 mL]: 785

TI (Tiempo de Inspiración) [0.1 - 10 s]: 7

Te (Tiempo de expiración) [0.1 - 10 s]: 4

Relación I:E [1:X] (X de 1 a 5): 3

Presión Inspiratoria Máxima (PIP) [20 - 80 cmH₂O]: 10

REGRESAR INICIAR RESPIRACIÓN

Figura 12. Pantalla parámetros a seleccionar

Para el monitoreo en tiempo real, la interfaz gráfica está configurada para mostrar gráficos de presión, flujo y volumen tidal (Fig. 13), los cuales se actualizan cada dos segundos mediante peticiones AJAX con Fetch.



Figura 13. Pantalla gráficos en tiempo real

El sistema también incluye una interfaz de monitoreo remoto, accesible dentro de la misma red local, lo que facilita a los profesionales de la salud supervisar el estado del ventilador sin necesidad de estar físicamente presentes. Cuando el sistema no está siendo utilizado, en la interfaz remoto aparecerá el mensaje de “Cargando...” y “Esperando datos...”.

La Raspberry Pi y el Arduino Mega incluyen un Watchdog Timer que reinicia el sistema en caso de bloqueos, además de un sistema de monitoreo de batería que advierte al usuario si el ventilador está operando con la fuente principal o con la batería de respaldo. En el Anexo se encuentra el código completo del sistema.

Los códigos de programación para estas interfaces se encuentran adjuntadas en el Anexo D.

2.6. Diagrama de conexiones y esquema

2.6.1. Circuito electrónico

En la Fig. 14 se observa el circuito que representa la conexión general del sistema, sin el circuito de conmutación de la Fig. 4. Siendo el software utilizado para el diseño del esquema electrónico el Fritzing, por su versatilidad y facilidad de uso.

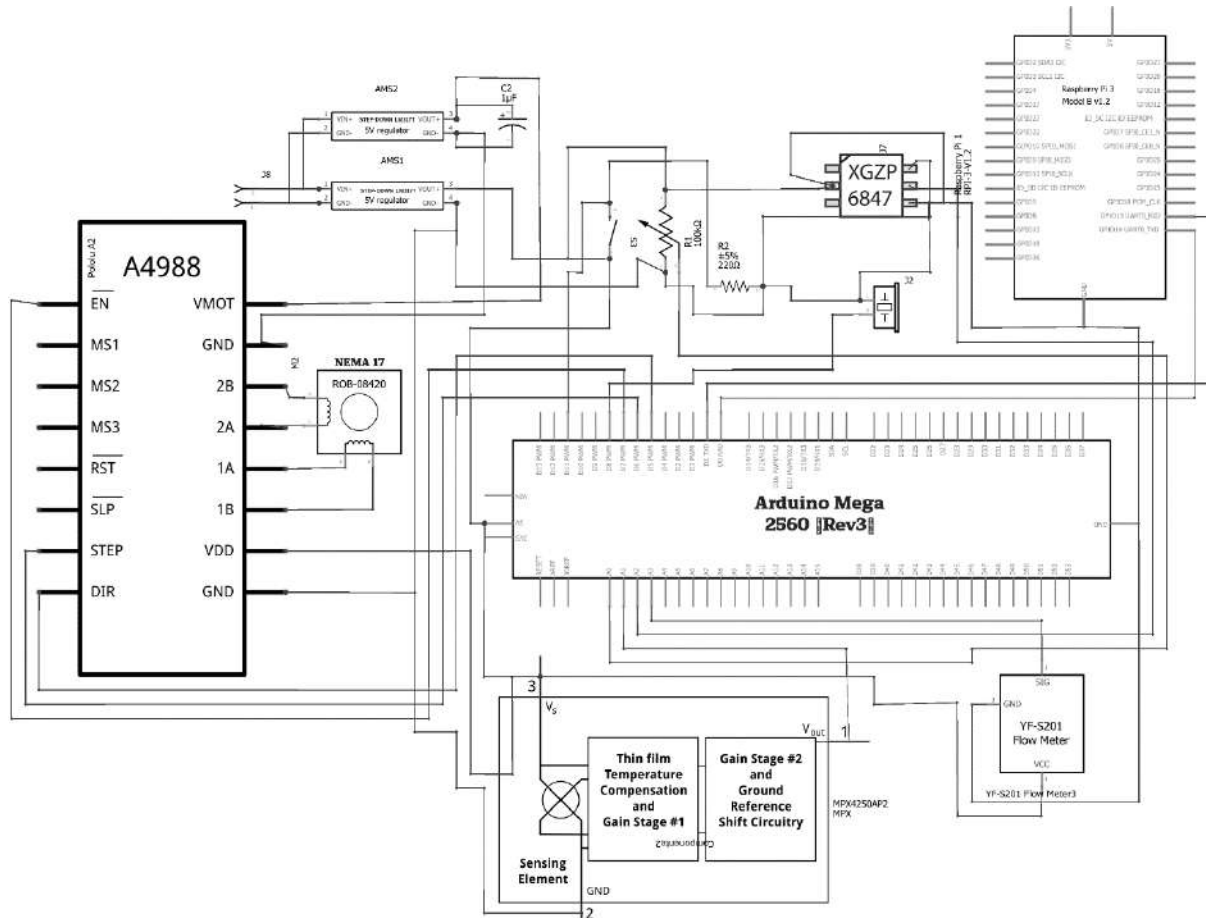


Figura 14. Esquema de conexión

Como se puede observar, el Arduino Mega se encarga de la adquisición de datos de los sensores y del control del Nema 17 para el accionamiento de los mecanismos del respirador artificial, además de comunicarse con la Raspberry Pi 3 mediante UART a través de los pines TX (1) y RX (0) (cuando se lleve a la construcción del prototipo, se puede reemplazar esas conexiones simplemente con un cable USB). El sensor de presión diferencial MPX5050DP está conectado a los pines A1 del Arduino para medir la presión en la vía aérea, mientras que el sensor de presión XGZP6847 se conecta al pin A2 para detectar presión negativa y positiva, ambos utilizando el ADC interno del microcontrolador. El sensor de flujo YF-S201 se conecta al pin A3, el cual utiliza una interrupción externa para contar los pulsos generados por la turbina y calcular el flujo de aire.

El motor paso a paso NEMA 17, controlado por el driver A4988, se conecta a los pines D3 (STEP) y D4 (DIR) del Arduino para definir la velocidad y dirección del giro. El buzzer conectado al pin

D8 emite alarmas sonoras en caso de fallos o valores críticos. Además, un pulsador de emergencia, ubicado en el pin D11, permite apagar el sistema en caso de un fallo grave. Todo este sistema recibe alimentación a través del reductor los reductores de voltaje Step-Down 20W 2A LM317T; se utilizaron dos reductores VCC – VCC, uno para la alimentación exclusiva del motor a pasos (por su consumo de corriente elevado) y otro para el resto de sensores y las controladores principales.

2.6.2. Esquema general del respirador artificial

El esquema general del respirador puede observarse en la Fig. 15, en donde se detallan los principales componentes del mismo.

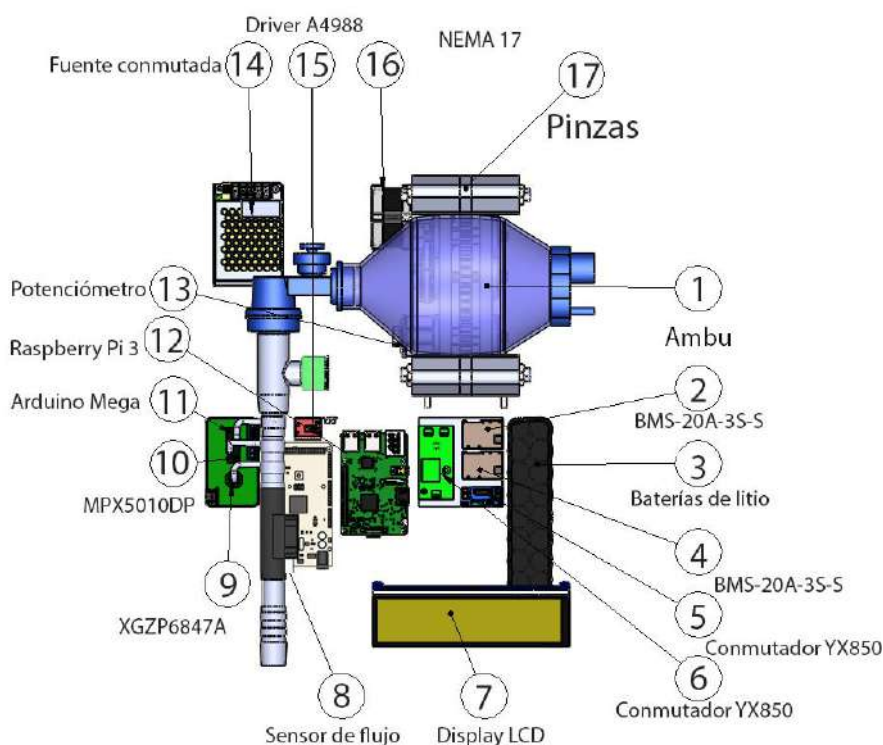


Figura 15. Esquema interno del respirador y componentes

Como puede observarse, el esquema interno del respirador inicia desde el AMBU, con el accionamiento del sistema por medio de las pinzas, los cuales dirigen el aire por el conducto. En el mismo, se encuentran instalados los tres principales sensores del sistema (8, 9, 10). Todo este sistema, como ya se ha indicado anteriormente, es procesado por los controladores (11 y 12), y representados en el display (7). Las dimensiones del respirador son 440x320x280 mm; en el Anexo A se pueden observar las medidas detalladas y vistas completas del proyecto.

2.7. Programas utilizados

Para el desarrollo del proyecto, se utilizaron dos principales programas, los cuales facilitaron el modelado 3D, el diseño electrónico y la programación del sistema embebido. En la parte de diseño

mecánico, se utilizó SolidWorks fue para la modelación 3D de la estructura del mecanismo de compresión del AMBU y así como todos los otros componentes mencionados en las secciones anteriores. Para la representación de los circuitos electrónicos y la interconexión de los sensores, controladores y actuadores, se empleó Fritzing, una herramienta intuitiva que permitió diseñar los esquemáticos eléctricos de manera clara.

En la programación del sistema, se utilizó Python, ejecutado en la Raspberry Pi 3, utilizando Thonny Python IDE y bibliotecas para la adquisición y procesamiento de datos. Para la programación del Arduino Mega, se utilizó el Arduino IDE, permitiendo la configuración de los módulos mediante lenguaje C/C++. Además, la interfaz web de monitoreo remoto fue desarrollada con HTML, CSS y JavaScript, dando especial énfasis en la accesibilidad y la interactividad. Para la gestión y comunicación con la interfaz gráfica, se empleó Flask, un framework ligero de Python, ayudando a una integración correcta con el sistema de monitoreo en tiempo real.

2.8. Análisis económico

Para el análisis económico del diseño del respirador artificial, se debe tener en cuenta varios puntos importantes, los cuales son: costos materiales, desarrollo del software e implementación, costos de ensamblaje y costos de certificación.

Aspecto a considerar	Costos parciales en dólares (\$)
Costos de materiales	647,33
Desarrollo del software e implementación	450
Costos de ensamblaje y mano de obra	361.5
Certificados y patentes	62.08
Costos totales	\$1.520,91

Tabla 6. Costos totales estimados para la implementación del respirador artificial

La Tabla 6 muestra, de manera general, el costo para el desarrollo del proyecto, equivalente a unos \$1.520,91, lo que son unos Gs. 12.022.357,69 en moneda nacional (cambio a Gs. 7.904,7, según la última actualización). Comparando esto con otros modelos de respiradores en el mercado, se encuentra muy por debajo de los respiradores mecánicos comerciales como el PUCP MASI (\$5000) y el Puritan Bennet 560 (\$2990), siendo este último el utilizado en el Hospital Distrital de Caaguazú. Esto quiere decir que el diseño es aproximadamente 2 veces más barato que el equipo usado en el centro hospitalario y más de 3 veces más barato que el PUCP MASI. Además, así como los respiradores mencionados, ofrece personalización de parámetros respiratorios a través de una interfaz táctil y web, permitiendo ajustes específicos para cada paciente. A esto se suma la funcionalidad de monitoreo remoto, el cual es un punto que da más ventaja en comparación a los otros modelos y que posibilita la supervisión del ventilador desde cualquier dispositivo conectado

a la misma red local, algo que no todos los modelos comerciales ofrecen. En el Anexo E se muestra el desglose general del costo de desarrollo del equipo.

El diseño del respirador artificial propuesto representa una opción económicamente accesible para hospitales, centros de salud públicos y privados, ofreciendo características similares a las de modelos comerciales a una fracción del costo. No obstante, es importante considerar los costos adicionales relacionados con certificaciones y pruebas clínicas para su implementación en entornos hospitalarios a nivel país.

3. Resultados y análisis

El prototipo del ventilador artificial fue diseñado con el propósito de proporcionar un equipo eficiente, accesible y adaptable al entorno del área de emergencia del Hospital Distrital de Caaguazú. Mediante un minucioso análisis de literatura especializada y las normativas, se establecieron los parámetros esenciales para garantizar el correcto desempeño del dispositivo. Durante el proceso de diseño, se identificaron las variables clínicas a monitorear, se seleccionaron los componentes electrónicos y mecánicos adecuados, y se implementó un sistema de monitoreo remoto basado en una plataforma web, lo que permitió alcanzar los objetivos propuestos en este estudio.

Primero, se establecieron los estándares fundamentales para la implementación del sistema de ventilación asistida, tomando en cuenta normativas como la ISO 80601-2-12, que regula los requisitos de seguridad y desempeño de los ventiladores mecánicos. Se definieron parámetros clínicos específicos, entre ellos la frecuencia respiratoria, la presión inspiratoria pico (PIP), el volumen tidal (VT), la relación inspiración-espriación (I:E) y la presión positiva al final de la espiración (PEEP). Estos fueron seleccionados con base en recomendaciones médicas y los datos recabados en los distintos trabajos ya hechos, además de la funcionalidad de los sistemas de ventilación asistida empleados en situaciones de emergencia.

La arquitectura del ventilador se fundamenta en un sistema de control basado en Raspberry Pi 3 y Arduino Mega 2560, donde la Raspberry Pi se encarga de la interfaz gráfica y la gestión de comunicaciones, mientras que el Arduino Mega actúa como el controlador de sensores y el actuador del sistema (motor a pasos Nema 17) en tiempo real. La selección de estos microcontroladores se hizo considerando la necesidad de un procesamiento eficiente de datos y una comunicación estable, implementando el protocolo UART para la transmisión de información entre ambos dispositivos. Además, la interfaz de usuario se desarrolló con Flask, HTML, CSS y JavaScript, permitiendo la visualización y configuración de los parámetros del ventilador mediante una aplicación web accesible desde dispositivos móviles y computadoras conectadas a la misma red local.

Para el mecanismo de ventilación, se optó por un sistema basado en un motor paso a paso NEMA 17, que permite la compresión precisa del Ambu. La elección de este motor se fundamentó en su capacidad de ofrecer un control preciso del volumen de aire administrado, lo que asegura una ventilación asistida adecuada a las necesidades del paciente. Para el correcto control de pasos del Nema, se utilizó un driver A4988, proporcionando así la precisión necesaria en la administración del flujo de aire en cada ciclo respiratorio.

Desde el punto de vista de la seguridad biológica, se incorporaron filtros HME para evitar la contaminación cruzada, en conformidad con la norma ISO 80601-2-12, y un sistema de alarmas basado en un buzzer y alertas visuales en la interfaz gráfica. Estos elementos garantizan que el personal médico sea notificado ante cualquier anomalía en el funcionamiento del ventilador, como la desconexión del circuito, la caída de presión o el agotamiento de la batería de respaldo.

El sistema de alimentación y respaldo energético fue diseñado para asegurar un funcionamiento continuo en caso de fallos eléctricos. Se utilizó una fuente conmutada de 12V y 10A, capaz de suministrar la energía requerida por todos los componentes, y un banco de baterías de ion-litio 18650 en configuración 4S4P, proporcionando una autonomía aproximada de 2.7 horas en caso de cortes de energía. La gestión de carga y protección de las baterías se realiza mediante un BMS-4S-30A, y el cambio automático entre la fuente principal y el respaldo es gestionado por el módulo conmutador YX850, asegurando una transición sin interrupciones.

Desde el punto de vista económico, la propuesta de diseño presenta un costo significativamente inferior al de los ventiladores comerciales. Mientras que dispositivos como el Puritan Bennet 560 tienen un costo aproximado de \$2990 y el PUCP MASI alcanza los \$5000, el presente diseño tiene un costo proyectado de \$1.520,91 (aproximadamente Gs. 12.022.357,69), lo que representa una alternativa viable para instituciones con recursos limitados.

En cuanto al monitoreo remoto, la implementación de una aplicación web basada en Flask permite la visualización en tiempo real de los parámetros del ventilador, facilitando la supervisión del equipo por parte del personal médico sin necesidad de estar físicamente presente junto al paciente. Sin embargo, se identificaron oportunidades de mejora en este aspecto, como la posibilidad de migrar el sistema a una infraestructura en la nube, lo que permitiría almacenar historiales clínicos y acceder a la información desde cualquier ubicación con conexión a internet. Además, el desarrollo de una aplicación móvil nativa (desarrollado específicamente en sistemas operativos Android o iOS) optimizaría la usabilidad y funcionalidad del sistema en dispositivos portátiles.

Este estudio demuestra la viabilidad de un ventilador artificial de bajo costo con monitoreo remoto, donde se resalta su potencial como una alternativa accesible para los centros asistenciales. Además de su futura implementación en el Hospital Distrital de Caaguazú, también puede contemplarse la idea de llevar este equipo en hospitales de regiones con escasez de equipos de ventilación, lo que podría representar una solución efectiva para la atención de pacientes en situaciones críticas, mejorando la capacidad de respuesta en emergencias médicas y reduciendo la dependencia de dispositivos importados de alto costo.

4. Conclusiones y recomendaciones

- Se identificaron los parámetros clínicos esenciales y las normativas aplicables, lo que permitió el correcto diseño del sistema de ventilación asistida funcional y alineado con los estándares médicos.
- Se establecieron los criterios técnicos de selección de componentes, garantizando un balance entre eficiencia, costo y operatividad del respirador artificial.
- El sistema de alimentación y respaldo energético, basado en baterías de ion-litio y un conmutador de emergencia, asegura un funcionamiento continuo en situaciones de cortes eléctricos.
- El análisis económico reflejó que el costo de implementación del diseño es significativamente inferior al de los ventiladores comerciales, manteniendo funcionalidades clave.
- Se identificaron áreas de mejora en la programación y optimización del software, lo que permitiría una mayor estabilidad y usabilidad del sistema.

4.1. Recomendaciones

- Se recomienda mejorar el software del sistema, implementando una interfaz más precisa e intuitiva, junto con algoritmos avanzados de control para una mejor respuesta del ventilador.
- Se sugiere evaluar la integración de sensores de grado médico con mayor precisión, aunque su incorporación aumentaría los costos del dispositivo.
- Se debe considerar la escalabilidad del sistema a una infraestructura en la nube, lo que permitiría almacenar historiales clínicos y mejorar el acceso remoto desde diferentes ubicaciones.
- Es recomendable realizar simulaciones avanzadas del flujo de aire en el sistema de ventilación, asegurando la eficiencia del diseño mecánico y la correcta distribución del volumen tidal.

Referencias

- [1] F. Morales, L. Bernal, G. Pereira, S. Pérez Buitrago, M. Kammer y D. Stalder, «PytuTester: RaspberryPi open-source ventilator tester,» *HardwareX*, vol. 12, n° e00334, 2022.
- [2] D. D. Stalder, «Informe del Equipo de Respiradores Artificiales,» Asunción, 2020.
- [3] M. P. F. Ortiz, *Diseño y construcción de un banco de pruebas para controlar y monitorizar los parámetros de funcionamiento del respirador artificial cpap ucl – ventura*, Ambato, 2022.
- [4] S. D. M. Borja, *Implementación de un prototipo de respirador artificial automático y un oxímetro para el tratamiento de pacientes en recuperación de COVID19 en el hogar*, Latacunda, 2020.
- [5] A. S. Tran, H. Q. Thinh Ngo, V. K. Dong y A. H. Vo, «Design, Control, Modeling, and Simulation of Mechanical Ventilator for Respiratory Support,» *Mathematical Problems in Engineering*, vol. 1, p. 15, 2021.
- [6] E. A. K. Malaekah, H. Farouk, E. Abuabid, V. V. Mukhanov, A. Alahmari y H. Alshagag, «Designing Hybrid Mechanical Ventilator System Based on Arduino and Raspberry Pi 4,» *Journal of Medical Devices*, vol. 16, n° 1, p. 014501, 2022.
- [7] A. Selmani, B. Benhala, A. Soufiane, M. Guerbaoui, A. Ed-Dahhak y A. Lachhab, «A portable artificial respirator with remote managing features.,» *E3S Web of Conf.*, vol. 469, n° 00100, pp. 1-5, 2023.
- [8] O. Flor, M. Fuentes, H. Carvajal , J. Quiroga, V. Luzuriaga , J. Tapia y P. Acosta Vargas, «Emergency Mechanical Ventilator Design: Low-Cost and Accessible Components,» *Electronics*, p. 18, 26 Noviembre 2022.
- [9] Á. E. Araujo Herrera y G. A. Campos Guato, Artists, *Diseño de Ventilador Mecánico Inteligente para pacientes con COVID-19 en UCI de hospitales*. [Art]. Escuela Superior Politécnica del Litoral, 2021.
- [10] F. M. Pasquevich, C. C. F. L. M., R. D. Hamann, G. N. Garaventa y M. D. Actis, «Diseño, construcción y ensayos de un respirador mecánico de emergencia,» *VII Congreso Argentino*

de Ingeniería Mecánica (CAIM) y II Congreso Argentino de Ingeniería Ferroviaria (CAIFE), pp. 1680-1691, 2021.

- [11] K. G. Naula Mediavilla y M. A. Pupiales Potosí, Artists, *Diseño y construcción de un prototipo de Ambu motorizado ensamblado con elementos automotrices para pacientes con SARS*. [Art]. Universidad Politécnica Salesiana, 2022.
- [12] Comité Superior de Normalización (CSN), *Norma Def San 1203, RES. MD N° 593/17*, 2017.
- [13] International Standard (ISO), *ISO 80601-2-12:2023. Particular requirements for basic safety and essential performance of critical care ventilators*, Second edition ed., 2020.
- [14] Diario Oficial de la Unión Europea, *Reglamento (UE) 2017/745 del Parlamento Europeo y del Consejo*, 2017.
- [15] International Standard, *IEC60601-1-8. Part 1-8: General requirements for basic safety and essential performance – Collateral standard: General requirements, tests and guidance for alarm systems in medical electrical equipment and medical electrical systems*, 2020.
- [16] Freescale Semiconductor, *MPX5010, MPXV5010, MPVZ5010 Series Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated*, 2009.
- [17] CFSensor, *XGZP6847A Pressure Sensor Module*.
- [18] M. Dávila Ascanio, Artist, *Estudio y propuesta de control de un sistema de ventilación asistida..* [Art]. Universidad de Alicante. Departamento de Tecnología Informática y Computación, 2024.
- [19] Allegro MicroSystems, LLC, *A4988. DMOS Microstepping Driver with Translator And Overcurrent Protection*, U.S.A.
- [20] C. T. Truong, K. H. Huynh, V. T. Duong, H. H. Nguyen, L. A. Pham y T. T. Nguyen, «Characteristic of Paddle Squeezing Angle and AMBU Bag Air Volume in Bag Valve Mask Ventilator,» *arXiv:2109.08019*, 2021.

Anexos

Anexo A. Diagramas y planos técnicos

Anexo A.1. Vista isométrica del respirador artificial propuesto

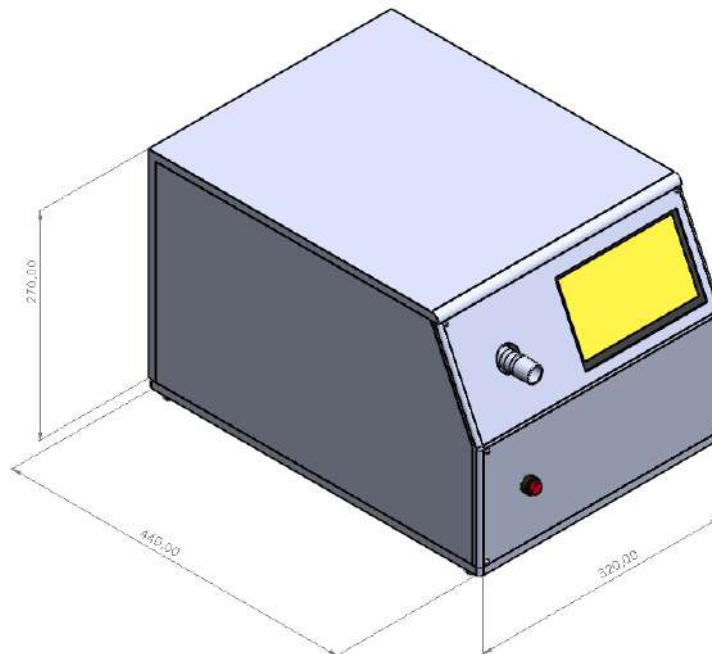


Figura 16. Vista isométrica del respirador artificial propuesto (en mm)

Anexo A.2. Vista explosionada del respirador con los principales componentes

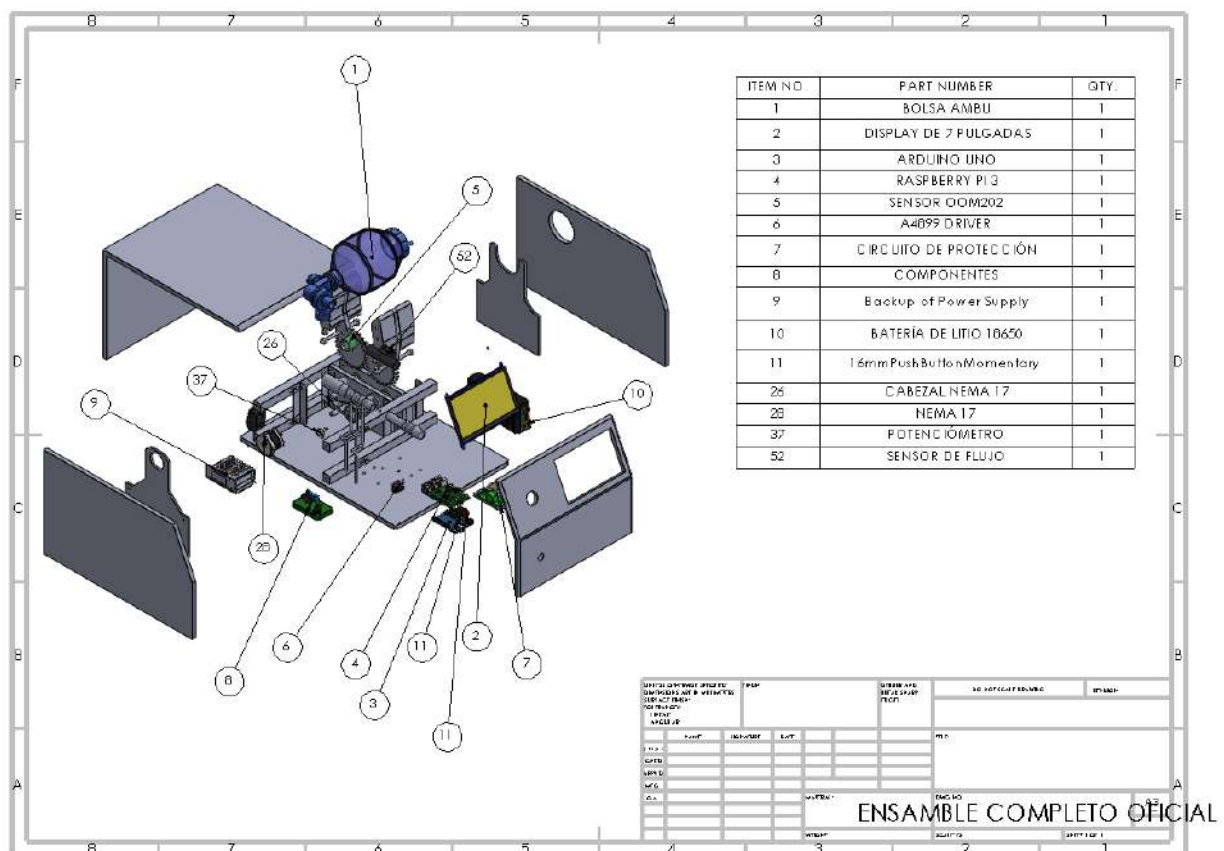


Figura 17. Vista explosionada del respirador con los principales componentes

Anexo A.3. Medidas generales del respirador

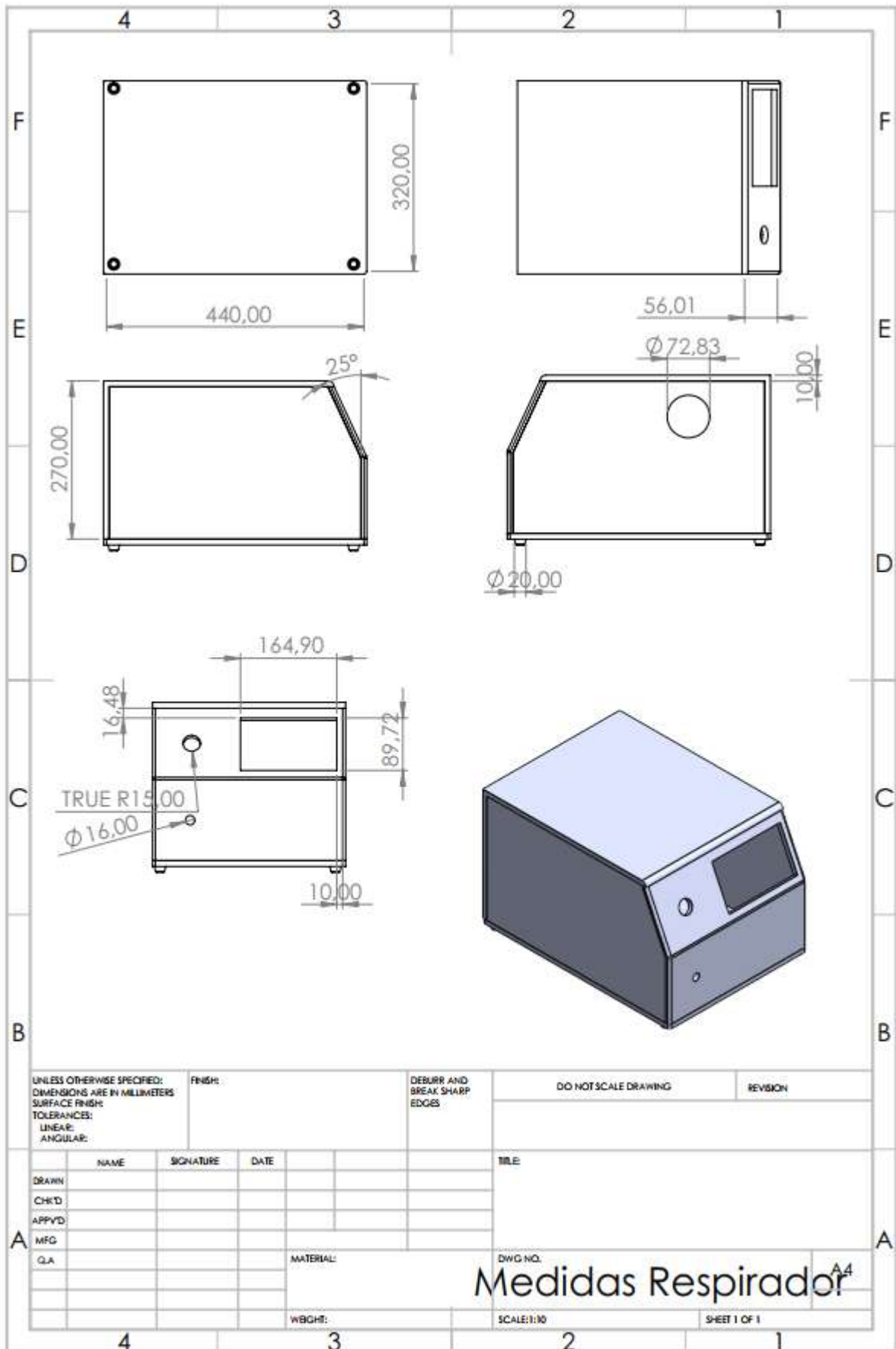


Figura 18. Medidas generales del respirador

Anexo A.4. Mecanismo dual-finger (pinza)

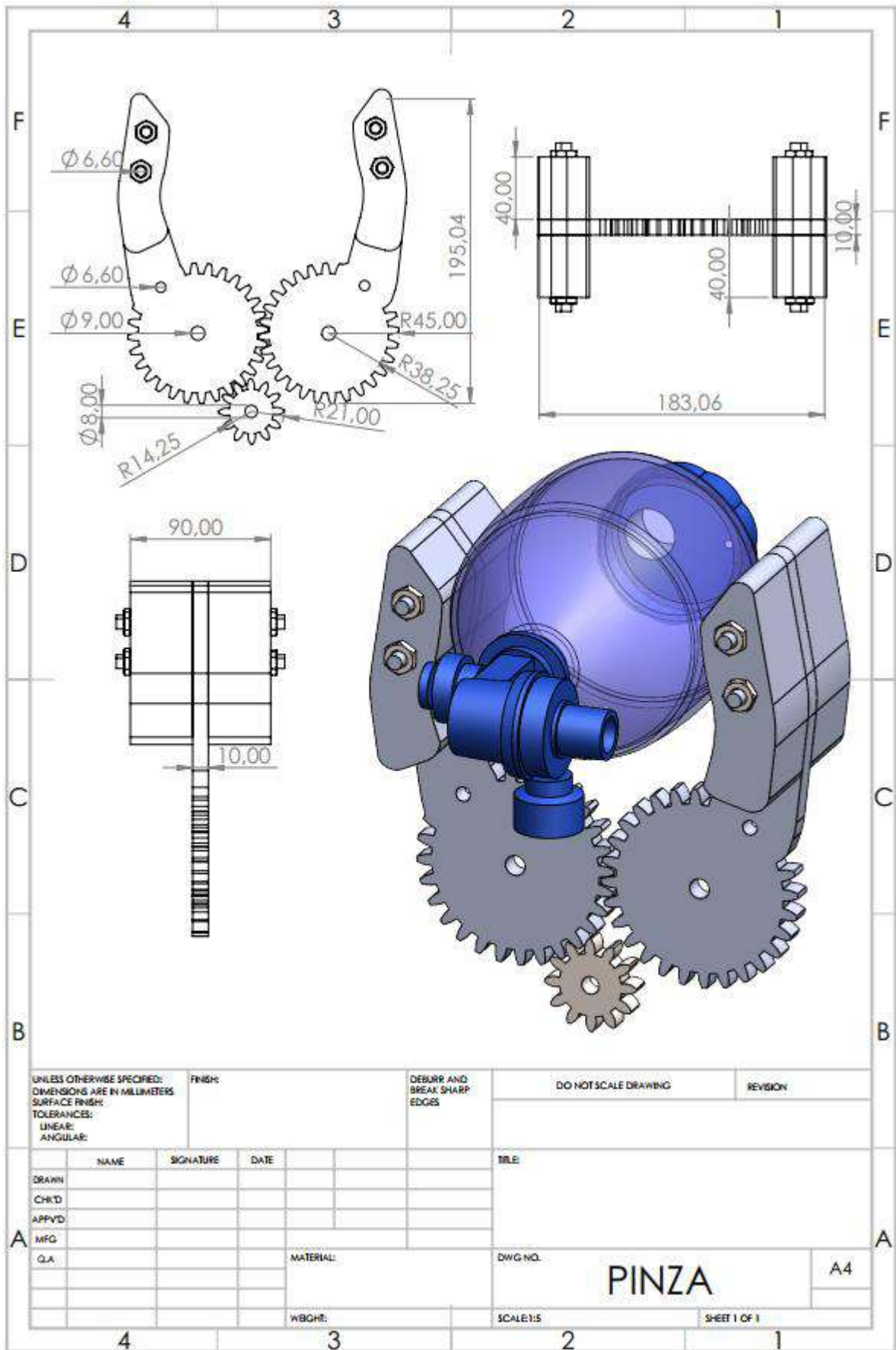


Figura 19. Mecanismo tipo dual-finger (pinza)

Anexo A.5. Sujetador AMBU

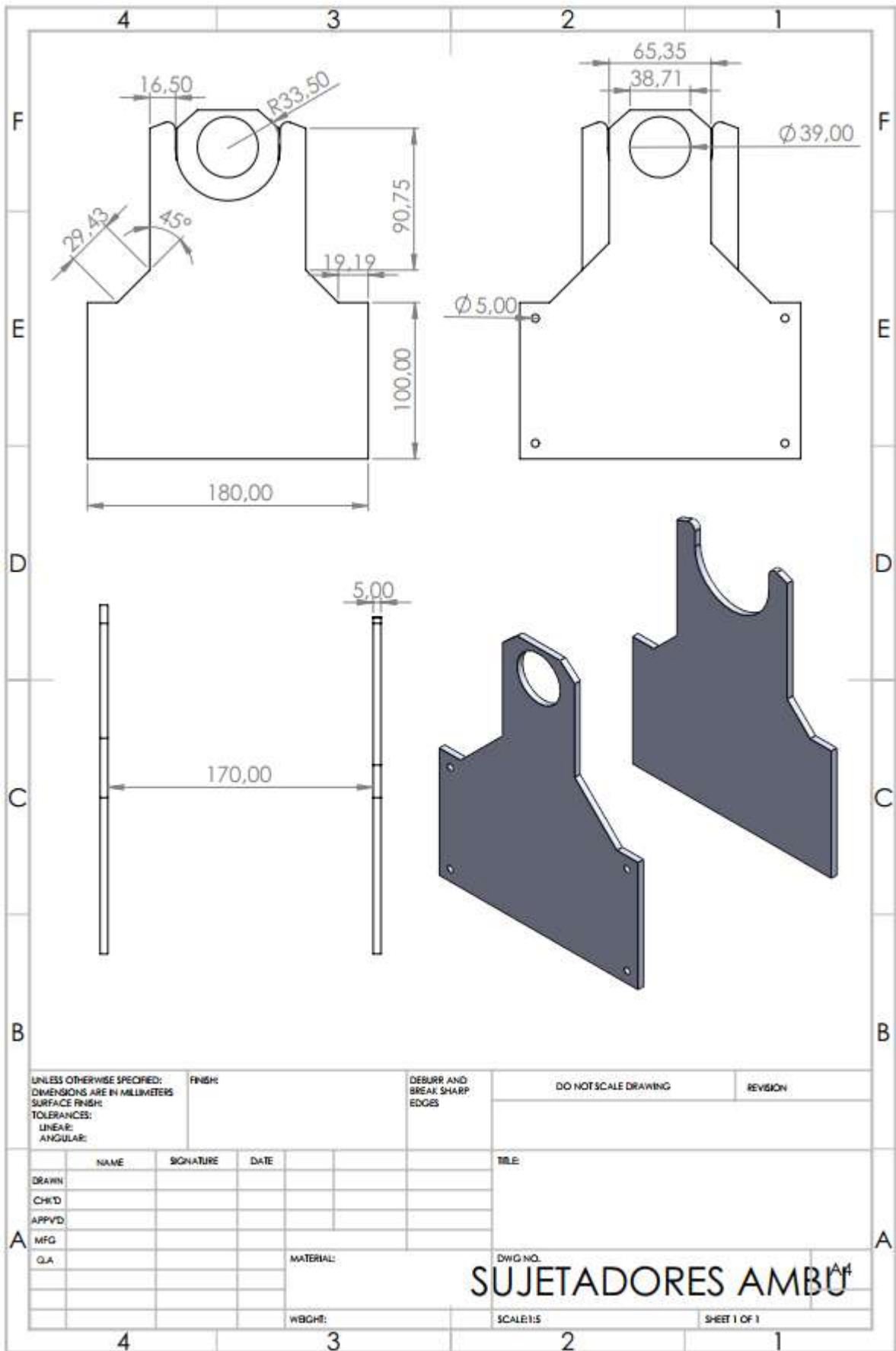


Figura 20. Sujetador de AMBU

Anexo A.6. Barra de soporte

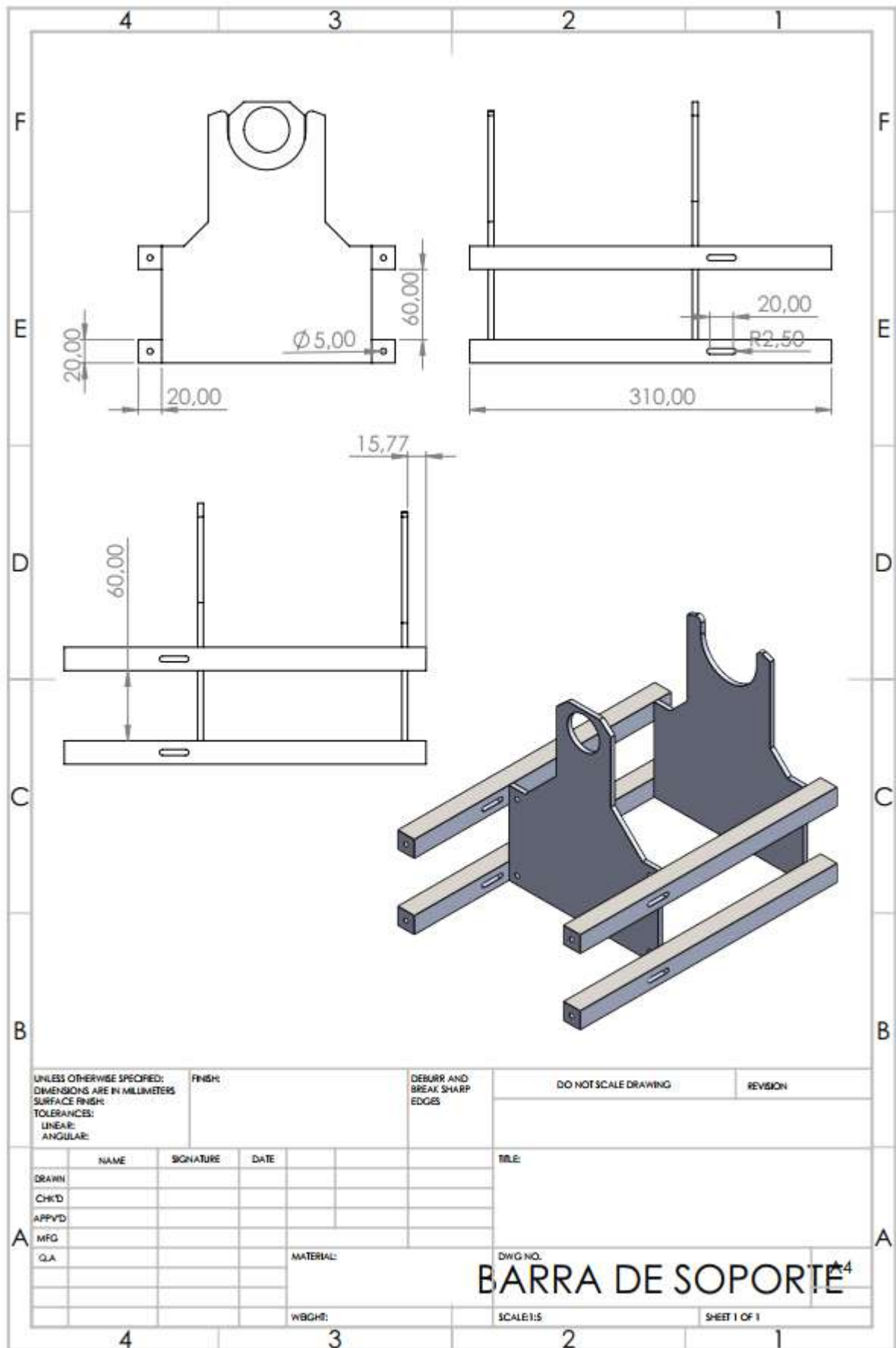


Figura 21. Barra de soporte

Anexo A.7. Perfil de soporte

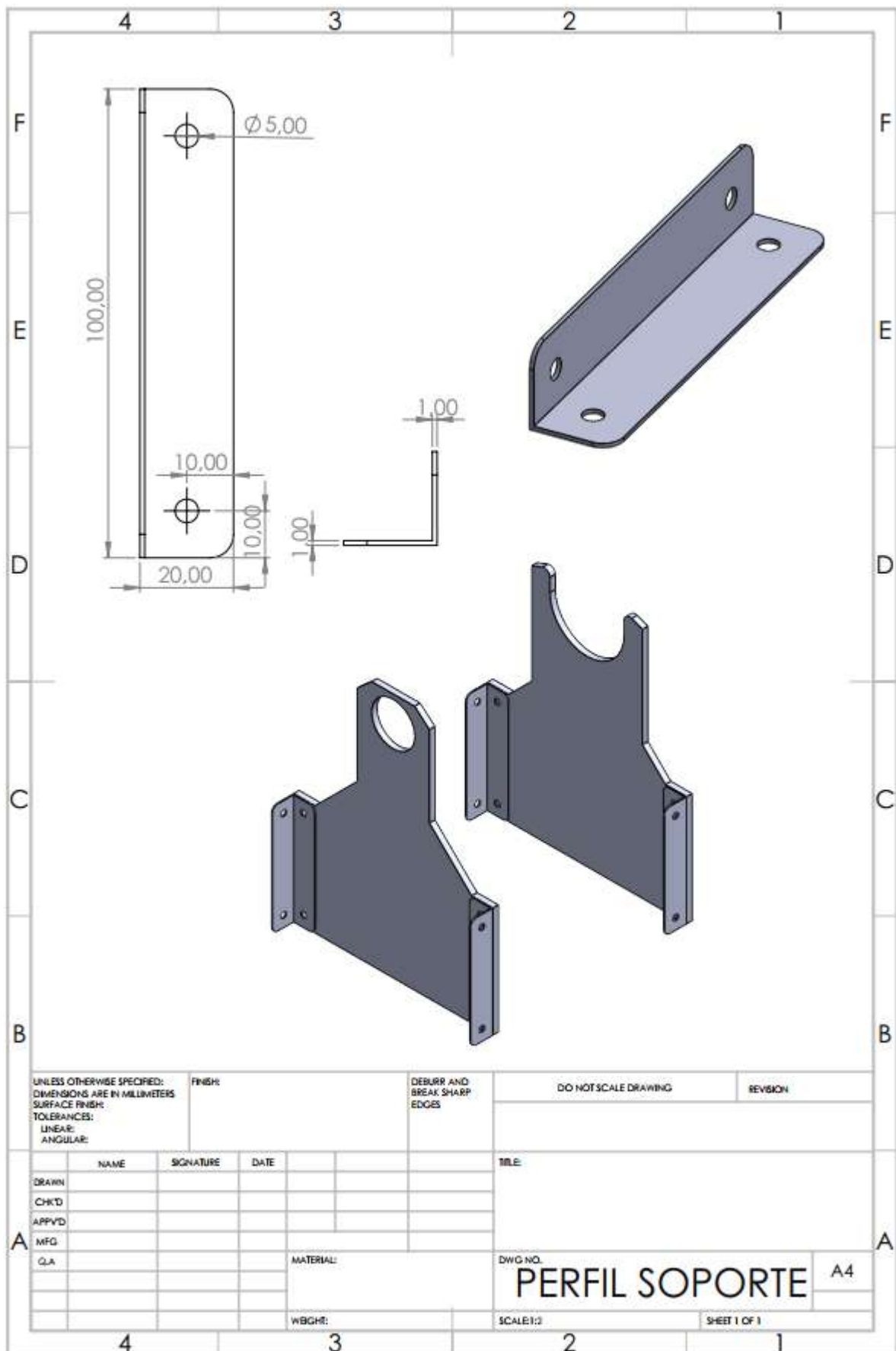


Figura 22. Perfil de soporte

Anexo A.8. Mecanismo y polea

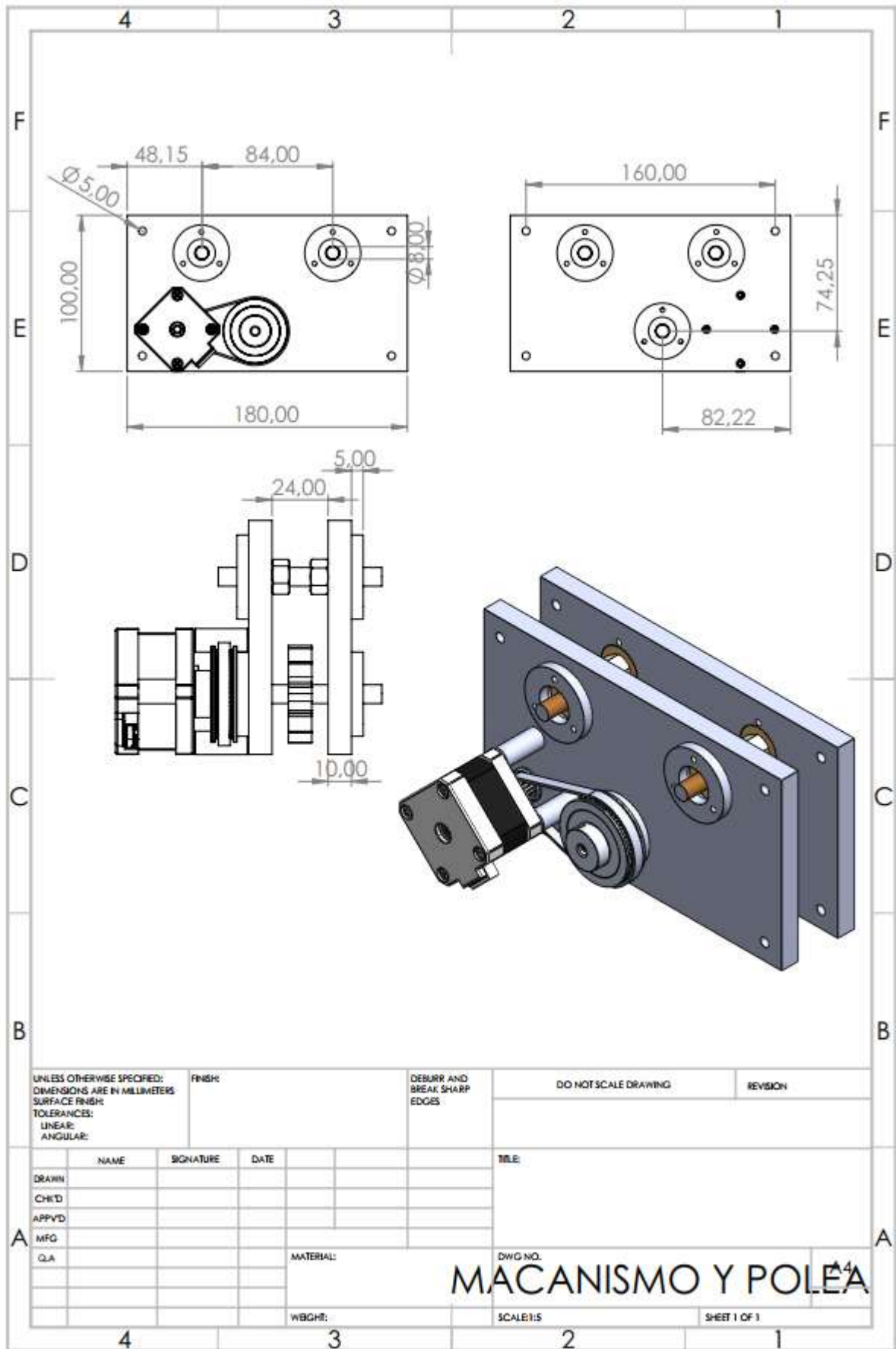


Figura 23. Mecanismo y polea

Anexo B. Diseño electrónico

Anexo B.1. Diagrama esquemático del respirador artificial

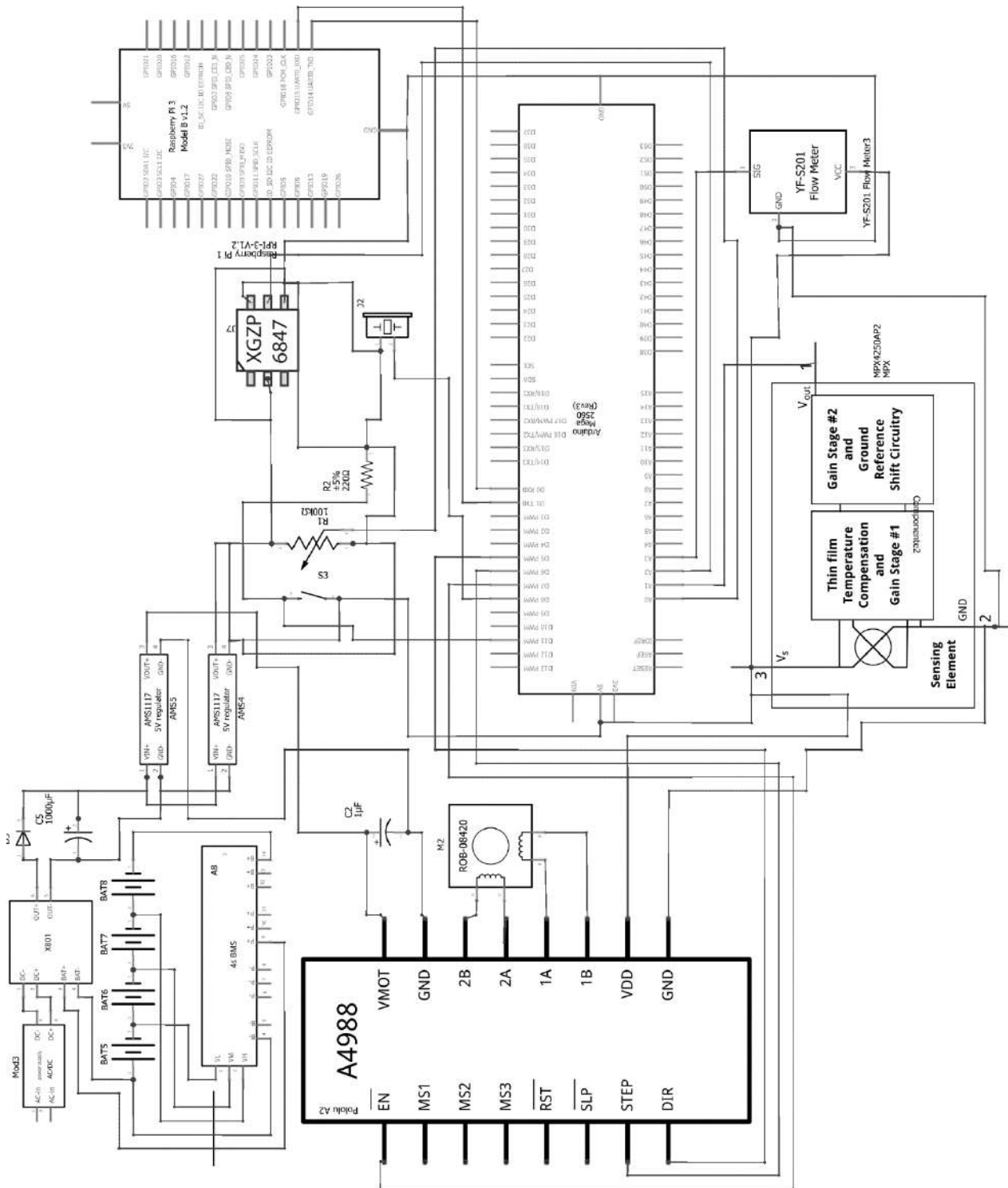


Figura 24. Diagrama esquemático del respirador artificial

Anexo B.2. Representación electrónica completa del respirador artificial

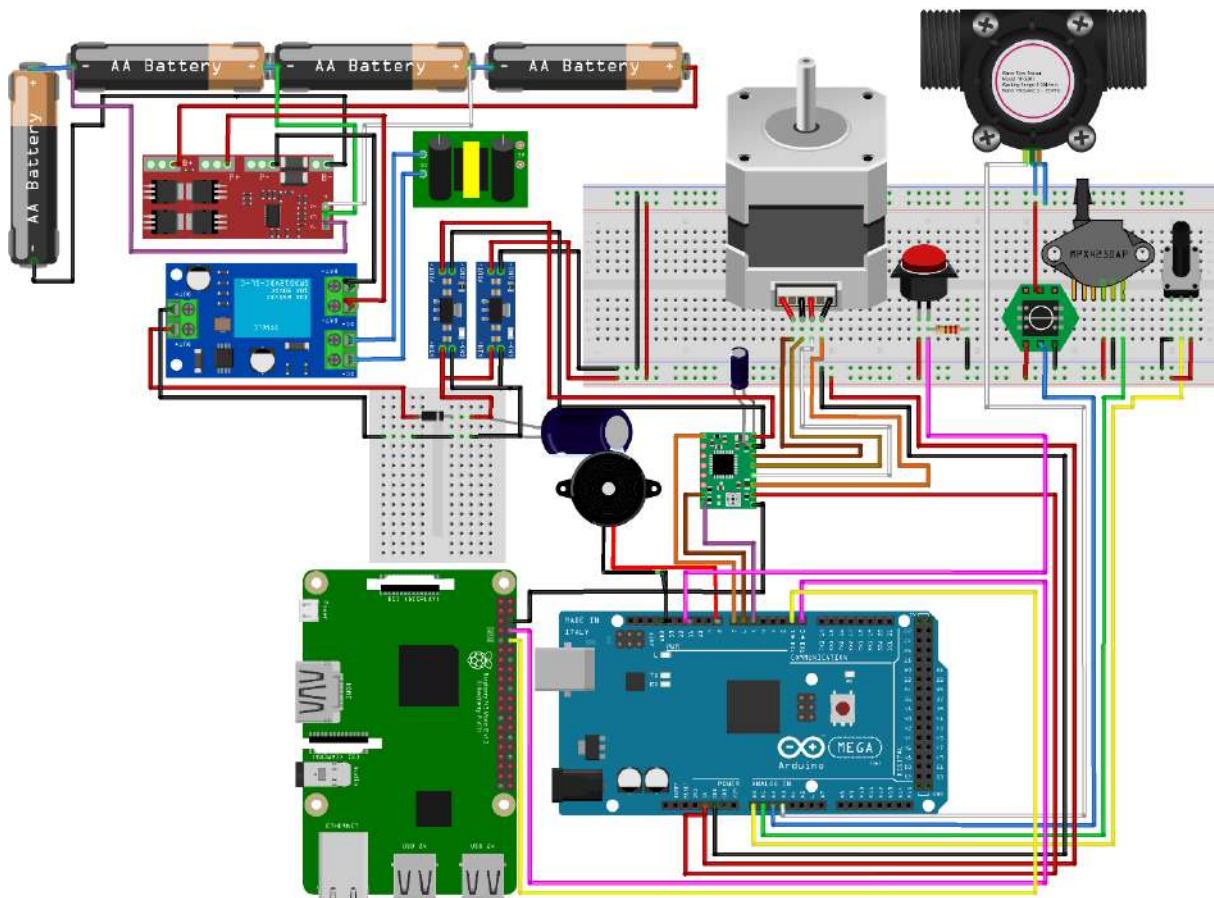


Figura 25. Representación electrónica completa del respirador artificial

Anexo B.3. Representación electrónica del sistema de conmutación de emergencia

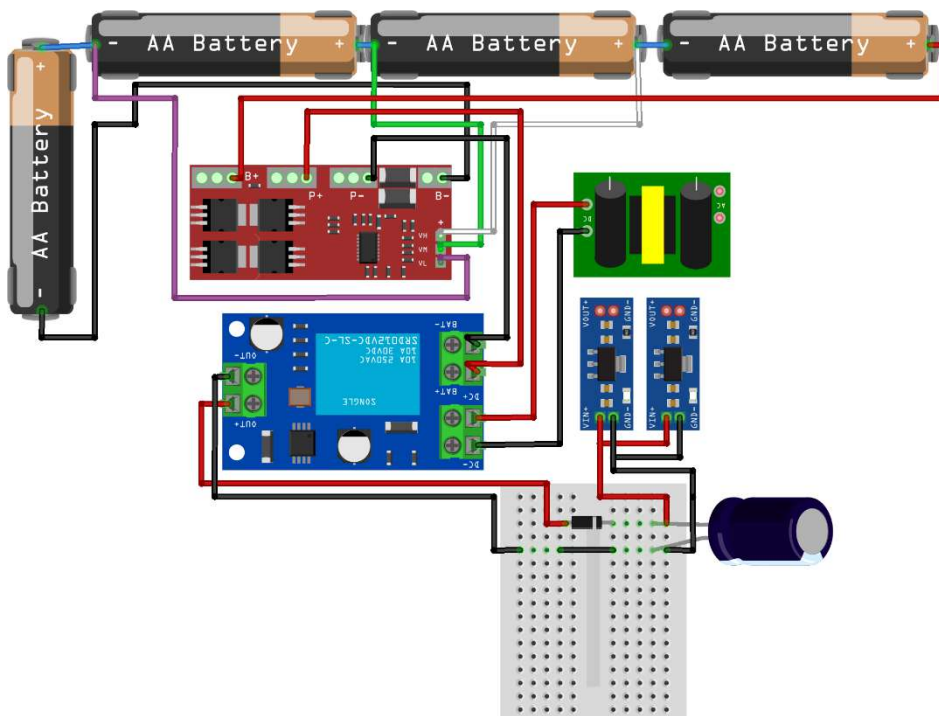


Figura 26. Representación electrónica del sistema de conmutación de emergencia

Anexo B.4. Configuración de baterías en 4S4P con módulo BMS-4S-30A

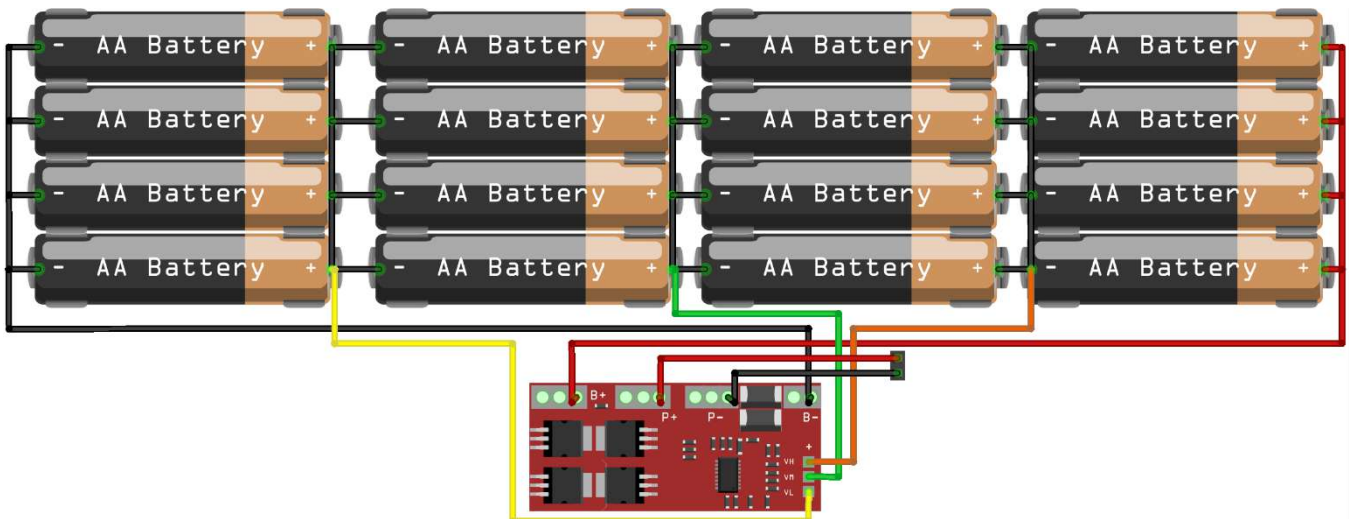


Figura 27. Configuración de baterías en 4S4P con módulo BMS-4S-30A

Anexo B.5. Diagrama esquemático de conexión del motor pasos Nema 17, A4988 y Arduino Mega

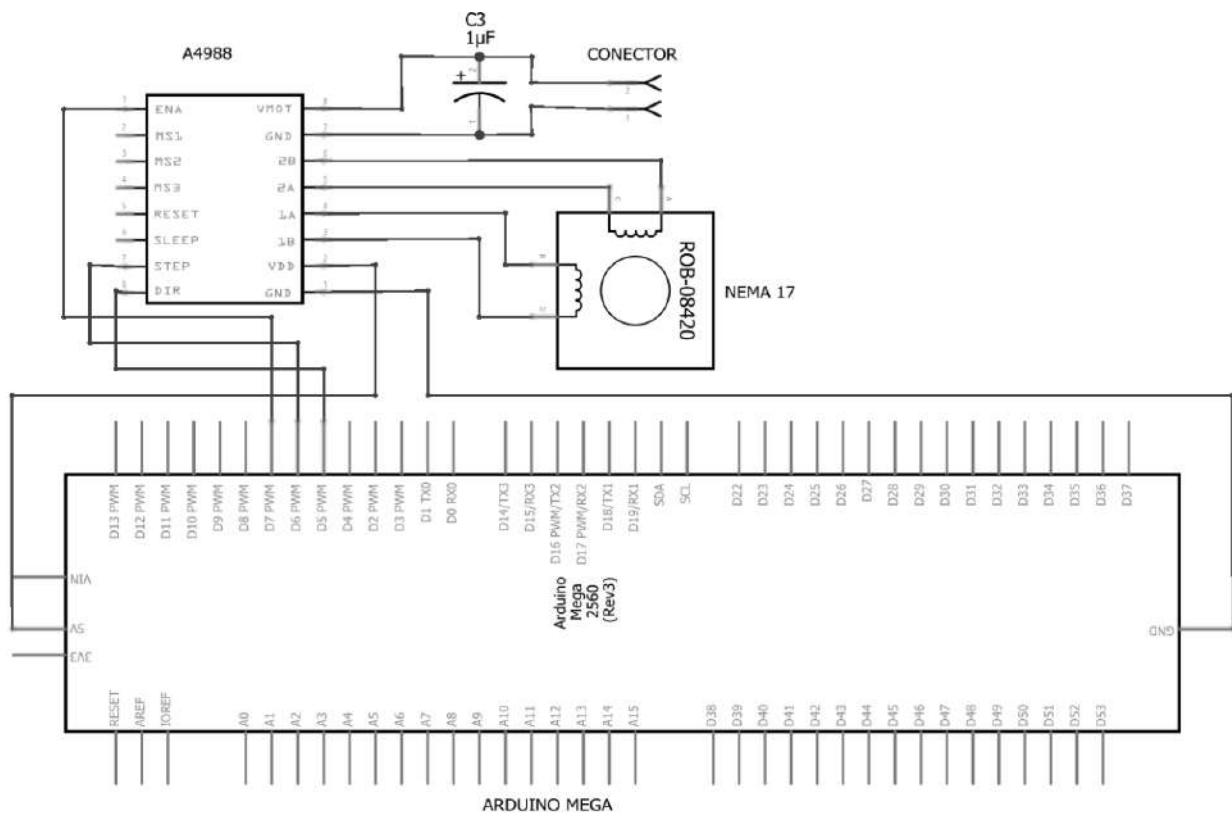


Figura 28. Diagrama esquemático de conexión del motor pasos Nema 17, A4988 y Arduino Mega

Anexo C. Programación del sistema

Anexo C.1. Programación en Arduino IDE

```
1
2 /*Respirador Mecánico*/
3
4 #include <Arduino.h>
5 #include <ArduinoJson.h>
6 #include <avr/wdt.h> // Librería para Watchdog Timer
7
8 /*Definición de pines*/
9 const int PIN_FLOW_SENSOR = A3;
10 const int PIN_PRESSURE_SENSOR = A2;
11 const int PIN_DIFF_PRESSURE_SENSOR = A1;
12 const int PIN_POTENTIOMETER = A0;
13
14 const int STEPPER_DIR_PIN = 5;
15 const int STEPPER_STEP_PIN = 6;
16 const int STEPPER_EN_PIN = 7;
17 const int BUZZER_PIN = 8;
18 const int EMERGENCY_BUTTON = 11;
19
20 /*Constantes*/
21 const unsigned long SEND_INTERVAL = 200;
22 const unsigned long SERIAL_BAUDRATE = 115200;
23 const float SCALE_FLOW = 1.0;
24 const float SCALE_PRESSURE = 0.5;
25 const float SCALE_DIFF_PRESSURE = 0.4;
26 const int POTENTIOMETER_MIN_ANGLE = 0;
27 const int POTENTIOMETER_MAX_ANGLE = 40; // Máximo rango permitido para la pinza
28
29 /*Variables Globales*/
30 unsigned long lastSendTime = 0;
31 unsigned long lastResponseTime = 0;
32 bool systemActive = false;
33 String inputCommand = "";
34
35 /*Setup*/
36 void setup() {
37     Serial.begin(SERIAL_BAUDRATE);
38
39     pinMode(PIN_FLOW_SENSOR, INPUT);
40     pinMode(PIN_PRESSURE_SENSOR, INPUT);
41     pinMode(PIN_DIFF_PRESSURE_SENSOR, INPUT);
42     pinMode(PIN_POTENTIOMETER, INPUT);
43
44     pinMode(STEPPER_DIR_PIN, OUTPUT);
45     pinMode(STEPPER_STEP_PIN, OUTPUT);
46     pinMode(STEPPER_EN_PIN, OUTPUT);
47     pinMode(BUZZER_PIN, OUTPUT);
48     pinMode(EMERGENCY_BUTTON, INPUT_PULLUP);
49
50     digitalWrite(STEPPER_EN_PIN, HIGH);
51     tone(BUZZER_PIN, 800, 300);
```

```
52     delay(300);
53     noTone(BUZZER_PIN);
54
55     Serial.println("Arduino Mega listo - Respirador en REPOSO");
56     wdt_enable(WDTO_8S); // Activar Watchdog Timer con reinicio en 8 segundos
57 }
58
59 /* Loop principal */
60 void loop() {
61     wdt_reset(); // Resetear Watchdog Timer
62     unsigned long currentMillis = millis();
63
64     if (currentMillis - lastSendTime >= SEND_INTERVAL) {
65         lastSendTime = currentMillis;
66         readAndSendData();
67     }
68     receiveCommands();
69     checkEmergencyStop();
70     checkMotorStatus();
71     checkPotentiometer();
72
73     if (millis() - lastResponseTime > 10000) {
74         Serial.println("{\"error\":\"NO RESPONSE FROM RASPBERRY\"}");
75         tone(BUZZER_PIN, 2000, 1000);
76     }
77 }
78
79 /* Lectura y envío de datos */
80 void readAndSendData() {
81     StaticJsonDocument<200> doc;
82     doc["status"] = systemActive ? "ACTIVO" : "REPOSO";
83     doc["flow"] = analogRead(PIN_FLOW_SENSOR) * SCALE_FLOW;
84     doc["pressure"] = analogRead(PIN_PRESSURE_SENSOR) * SCALE_PRESSURE;
85     doc["diff_pressure"] = analogRead(PIN_DIFF_PRESSURE_SENSOR) * SCALE_DIFF_PRESSURE;
86     doc["potentiometer"] = analogRead(PIN_POTENTIOMETER);
87
88     String output;
89     serializeJson(doc, output);
90     Serial.println(output);
91 }
92
93 /*Verificación del potenciómetro*/
94 void checkPotentiometer() {
95     int potValue = analogRead(PIN_POTENTIOMETER);
96     int angle = map(potValue, 0, 1023, 0, 90); // Convierte el valor a ángulo
97
98     if (angle > POTENTIOMETER_MAX_ANGLE) {
99         Serial.println("{\"error\":\"PINZA FUERA DE RANGO\"}");
100         tone(BUZZER_PIN, 1500, 500);
101     }
102 }
103
104 /* Recepción de Comandos */
105 void receiveCommands() {
```

```
106     while (Serial.available() > 0) {
107         char c = (char)Serial.read();
108         if (c == '\n') {
109             parseCommand(inputCommand);
110             inputCommand = "";
111             lastResponseTime = millis();
112         } else {
113             inputCommand += c;
114         }
115     }
116 }
117
118 /* Parse de Comandos */
119 void parseCommand(String command) {
120     command.trim();
121     if (command.startsWith("start")) {
122         systemActive = true;
123         Serial.println("{\"info\":\"Ventilación INICIADA\"}");
124     } else if (command.startsWith("stop")) {
125         systemActive = false;
126         Serial.println("{\"info\":\"Ventilación DETENIDA\"}");
127     } else if (command.indexOf("buzzer=on") >= 0) {
128         tone(BUZZER_PIN, 1000, 500);
129     } else if (command.indexOf("buzzer=off") >= 0) {
130         noTone(BUZZER_PIN);
131     }
132 }
133
134 /* Control del Motor */
135 void checkMotorStatus() {
136     if (systemActive) {
137         digitalWrite(STEPPER_EN_PIN, LOW);
138     } else {
139         digitalWrite(STEPPER_EN_PIN, HIGH);
140     }
141 }
142
143 /* Emergencia */
144 void checkEmergencyStop() {
145     if (digitalRead(EMERGENCY_BUTTON) == LOW) {
146         systemActive = false;
147         noTone(BUZZER_PIN);
148         digitalWrite(STEPPER_EN_PIN, HIGH);
149         Serial.println("{\"error\":\"EMERGENCY_STOP\"}");
150         while (1) {
151             delay(1000);
152         }
153     }
154 }
155 }
```

Anexo C.2. Programación de la Raspberry Pi 3 en Python

```
1 import threading
2 import time
3 import json
4 import os
5 from flask import Flask, request, jsonify, render_template, redirect, url_for, session
6 import serial
7
8 app = Flask(__name__)
9 app.secret_key = "super_secret_key" # Para gestionar sesiones
10
11 # ----- Configuración Serial -----
12 SERIAL_PORT = "/dev/ttyACM0"
13 BAUD_RATE = 115200
14 RECONNECT_DELAY = 5
15
16 ser = None
17 serial_lock = threading.Lock()
18
19 # Almacenar los datos más recientes enviados por el Arduino
20 current_data = {}
21 data_lock = threading.Lock()
22
23 # Código de seguridad
24 SECURITY_CODE = "1234" # Código de acceso para el personal médico
25
26 # ----- Configuración del Watchdog Timer -----
27 def keep_alive():
28     while True:
29         os.system("echo '1' > /dev/watchdog")
30         time.sleep(10) # Resetear watchdog cada 10 segundos
31
32 watchdog_thread = threading.Thread(target=keep_alive, daemon=True)
33 watchdog_thread.start()
34
35 # ----- Hilo de lectura Serial -----
36 def serial_reader_thread():
37     global ser
38     while True:
39         try:
40             if ser is None or not ser.is_open:
41                 print(f"Intentando conectar a {SERIAL_PORT} ...")
42                 ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
43                 print(f"Conectado a {SERIAL_PORT} a {BAUD_RATE} baudios.")
44
45             line = ser.readline().decode('utf-8').strip()
46             if line:
47                 try:
48                     data_json = json.loads(line)
49                     with data_lock:
50                         current_data.update(data_json)
51
```

```
52         # Verificar si hay error en el potenciómetro
53         if "error" in data_json and data_json["error"] == "PINZA FUERA DE
RANGO":
54             print("⚠ ALERTA: PINZA FUERA DE RANGO DETECTADA")
55         except json.JSONDecodeError:
56             print(f"No se pudo decodificar JSON: {line}")
57
58     except serial.SerialException as e:
59         print(f"[ERROR] Serial: {e}")
60         if ser:
61             try:
62                 ser.close()
63             except:
64                 pass
65             ser = None
66             time.sleep(RECONNECT_DELAY)
67     except Exception as ex:
68         print(f"[ERROR] Hilo serial: {ex}")
69         time.sleep(1)
70
71 # ----- Función para enviar comandos al Arduino -----
72 def send_command_to_arduino(cmd_str):
73     global ser
74     with serial_lock:
75         if ser and ser.is_open:
76             command_line = cmd_str.strip() + "\n"
77             ser.write(command_line.encode('utf-8'))
78         else:
79             print("No hay conexión serial disponible para enviar comandos.")
80
81 # ===== Rutas Flask =====
82 @app.route("/")
83 def password():
84     return render_template("password.html")
85
86 @app.route("/verify_code", methods=["POST"])
87 def verify_code():
88     code = request.form.get("password")
89     if code == SECURITY_CODE:
90         session['authenticated'] = True
91         return redirect(url_for("inicio"))
92     else:
93         return render_template("password.html", error="Código incorrecto. Inténtelo de
nuevo.")
94
95 @app.route("/inicio", methods=["GET", "POST"])
96 def inicio():
97     if not session.get('authenticated'):
98         return redirect(url_for("password"))
99     if request.method == "POST":
100         session['patient_data'] = request.form.to_dict()
101         return redirect(url_for("variables"))
102     return render_template("Inicio.html")
103
```

```
104 @app.route("/variables", methods=["GET", "POST"])
105 def variables():
106     if not session.get('authenticated'):
107         return redirect(url_for("password"))
108     if request.method == "POST":
109         session['ventilation_params'] = request.form.to_dict()
110         return redirect(url_for("graficos"))
111     return render_template("variables.html")
112
113 @app.route("/graficos")
114 def graficos():
115     if not session.get('authenticated'):
116         return redirect(url_for("password"))
117     return render_template("gráficos.html")
118
119 @app.route("/api/data", methods=["GET"])
120 def api_data():
121     with data_lock:
122         data_copy = dict(current_data)
123         return jsonify(data_copy)
124
125 @app.route("/api/set_params", methods=["POST"])
126 def api_set_params():
127     params = request.form.to_dict()
128     valid_params = {key: params[key] for key in ["freq", "vt", "pip", "peep", "fio2",
129 "mode", "ti", "te"] if key in params}
130
131     if valid_params:
132         cmd = "set:" + " ".join(f"{k}={v}" for k, v in valid_params.items())
133         send_command_to_arduino(cmd)
134
135     return redirect(url_for("variables"))
136
137 @app.route("/api/control", methods=["POST"])
138 def api_control():
139     action = request.form.get("action", "")
140     if action in ["start", "stop", "buzzer_on", "buzzer_off"]:
141         send_command_to_arduino(action)
142     elif action == "resume":
143         send_command_to_arduino("resume")
144     return redirect(url_for("graficos"))
145
146 # ===== Main =====
147 if __name__ == "__main__":
148     reader_t = threading.Thread(target=serial_reader_thread, daemon=True)
149     reader_t.start()
150     app.run(host="0.0.0.0", port=5000, debug=False)
```

Anexo D. Programación de la interfaz local y web

Anexo D.1. Página de autenticación

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Acceso Seguro</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      text-align: center;
11      background-color: #001f3f;
12      color: white;
13      display: flex;
14      justify-content: center;
15      align-items: center;
16      height: 100vh;
17      margin: 0;
18    }
19
20    .container {
21      max-width: 400px;
22      padding: 20px;
23      background-color: #003366;
24      border-radius: 10px;
25      box-shadow: 0px 0px 10px rgba(255, 255, 255, 0.3);
26    }
27
28    h1 {
29      margin-bottom: 10px;
30    }
31
32    p {
33      margin-bottom: 20px;
34    }
35
36    .input-container {
37      position: relative;
38      margin-bottom: 15px;
39    }
40
41    input {
42      padding: 10px;
43      font-size: 18px;
44      width: 100%;
45      border-radius: 5px;
46      border: 1px solid #ccc;
47      box-sizing: border-box;
48    }
49
50    .toggle-password {
51      position: absolute;
```

```
52     right: 10px;
53     top: 50%;
54     transform: translateY(-50%);
55     background: none;
56     border: none;
57     font-size: 16px;
58     cursor: pointer;
59     color: white;
60 }
61
62 .button {
63     padding: 10px;
64     font-size: 18px;
65     width: 100%;
66     border-radius: 5px;
67     background-color: #007bff;
68     color: white;
69     border: none;
70     cursor: pointer;
71     margin-top: 10px;
72 }
73
74 .button:hover {
75     background-color: #0056b3;
76 }
77
78 </style>
79 </head>
80 <body>
81
82     <div class="container">
83         <h1>Acceso Seguro</h1>
84         <p>Ingrese la contraseña para acceder al sistema:</p>
85
86         <form onsubmit="event.preventDefault(); verificarPassword();">
87             <div class="input-container">
88                 <input type="password" id="password" placeholder="Ingrese su contraseña"
required>
89                 <button type="button" class="toggle-password" onclick="togglePassword()">⊞
</button>
90             </div>
91             <button type="submit" class="button">Ingresar</button>
92         </form>
93     </div>
94
95     <script>
96         function togglePassword() {
97             const passwordField = document.getElementById("password");
98             passwordField.type = passwordField.type === "password" ? "text" : "password";
99         }
100
101         function verificarPassword() {
102             const passwordIngresada = document.getElementById("password").value;
103             const passwordCorrecta = "1234"; // Puedes cambiar la contraseña aquí
```

```
104
105     if (passwordIngresada === passwordCorrecta) {
106         localStorage.setItem("accesoPermitido", "true");
107         window.location.href = "Inicio.html";
108     } else {
109         alert("Contraseña incorrecta. Intente de nuevo.");
110     }
111 }
112 </script>
113 <script>
114 document.getElementById("passwordForm").addEventListener("submit", async function(event)
115 {
116     event.preventDefault(); // Evita la recarga completa
117
118     let password = document.getElementById("password").value;
119     let response = await fetch("/verify_code", {
120         method: "POST",
121         headers: { "Content-Type": "application/x-www-form-urlencoded" },
122         body: `password=${password}`
123     });
124
125     if (response.redirected) {
126         window.location.href = response.url; // Redirige si el código es correcto
127     } else {
128         alert("⚠ Código incorrecto. Inténtelo de nuevo.");
129     }
130 });
131 </script>
132 </body>
133 </html>
```

Anexo D.2. Página de inicio

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Pantalla de Inicio</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      background-color: black;
11      color: white;
12      display: flex;
13      justify-content: center;
14      align-items: center;
15      height: 100vh;
16      margin: 0;
17      padding: 10px;
18    }
19
20    .container {
21      background-color: #111;
22      padding: 20px;
23      border-radius: 10px;
24      box-shadow: 0 0 10px rgba(255, 255, 255, 0.3);
25      width: 90%;
26      max-width: 500px;
27      text-align: center;
28    }
29
30    .icon {
31      width: 70px;
32      height: 70px;
33      background-color: white;
34      border-radius: 50%;
35      display: flex;
36      justify-content: center;
37      align-items: center;
38      margin: 0 auto 20px;
39    }
40
41    .icon img {
42      width: 50px;
43      height: 50px;
44    }
45
46    label {
47      display: block;
48      text-align: left;
49      margin-top: 10px;
50      font-weight: bold;
51    }
```

```
52
53     input, textarea {
54         width: 100%;
55         padding: 10px;
56         margin-top: 5px;
57         border-radius: 5px;
58         border: none;
59         font-size: 16px;
60     }
61
62     input {
63         background-color: white;
64         color: black;
65     }
66
67     textarea {
68         background-color: #ccc;
69         resize: none;
70         height: 80px;
71     }
72
73     .row {
74         display: flex;
75         gap: 10px;
76         flex-wrap: wrap;
77     }
78
79     .row div {
80         flex: 1;
81         min-width: 120px;
82     }
83
84     .buttons {
85         display: flex;
86         justify-content: space-between;
87         margin-top: 20px;
88         flex-wrap: wrap;
89         gap: 10px;
90     }
91
92     .button {
93         background: white;
94         color: black;
95         border: none;
96         padding: 10px 20px;
97         border-radius: 20px;
98         font-size: 16px;
99         cursor: pointer;
100        box-shadow: 2px 2px 5px rgba(255, 255, 255, 0.2);
101        flex: 1;
102        min-width: 130px;
103    }
104
105    .button:hover {
```

```
106     background: #ddd;
107   }
108
109   @media (max-width: 480px) {
110     .row {
111       flex-direction: column;
112     }
113
114     .buttons {
115       flex-direction: column;
116     }
117   }
118
119 </style>
120 </head>
121 <body>
122
123   <div class="container">
124     <!-- Ícono de usuario -->
125     <div class="icon">
126       
127     </div>
128
129     <!-- Formulario -->
130     <form>
131       <label>Nombre del paciente:</label>
132       <input type="text" placeholder="Ingrese el nombre">
133
134       <div class="row">
135         <div>
136           <label>N° C.I.:</label>
137           <input type="text" placeholder="Ingrese C.I.">
138         </div>
139         <div>
140           <label>Sexo:</label>
141           <input type="text" placeholder="M/F">
142         </div>
143       </div>
144
145       <div class="row">
146         <div>
147           <label>Edad:</label>
148           <input type="number" placeholder="Edad">
149         </div>
150         <div>
151           <label>Peso (Kg):</label>
152           <input type="number" placeholder="Peso">
153         </div>
154       </div>
155
156       <label>Descripción:</label>
157       <textarea placeholder="Ingrese una descripción"></textarea>
158
159     <!-- Botones -->
```

```
160     <div class="buttons">
161         <button type="button" class="button">FECHA/HORA</button>
162         <button type="submit" class="button">ACEPTAR</button>
163         <button type="button" class="button"
onclick="window.location.href='variables.html'">SIGUIENTE</button>
164     </div>
165 </form>
166 </div>
167 <script>
168     document.getElementById("formPaciente").addEventListener("submit", function(event) {
169         let nombre = document.getElementById("nombre").value;
170         let ci = document.getElementById("ci").value;
171
172         if (nombre.trim() === "" || ci.trim() === "") {
173             alert("⚠ Todos los campos deben estar completos.");
174             event.preventDefault(); // Detiene el envío del formulario
175         }
176     });
177 </script>
178 </body>
179 </html>
180
```

Anexo D.3. Página de configuración de variables

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Definición de Variables Respiratorias</title>
7     <style>
8         body {
9             font-family: Arial, sans-serif;
10            background-color: black;
11            color: white;
12            display: flex;
13            justify-content: center;
14            align-items: center;
15            min-height: 100vh;
16            margin: 0;
17            padding: 20px;
18        }
19
20        .container {
21            background: white;
22            color: black;
23            padding: 20px;
24            border-radius: 10px;
25            box-shadow: 0px 0px 10px rgba(255, 255, 255, 0.3);
26            width: 90%;
27            max-width: 500px;
28            text-align: center;
29        }

```

```
30
31     h2 {
32         margin-bottom: 20px;
33     }
34
35     .form-group {
36         text-align: left;
37         margin-bottom: 15px;
38     }
39
40     .form-group label {
41         display: block;
42         font-weight: bold;
43         margin-bottom: 5px;
44     }
45
46     .form-group input,
47     .form-group select {
48         width: 100%;
49         padding: 8px;
50         font-size: 16px;
51         border: 2px solid black;
52
53         border-radius: 5px;
54         box-sizing: border-box;
55     }
56
57     .buttons {
58         display: flex;
59         justify-content: space-between;
60         margin-top: 20px;
61     }
62
63     .button {
64         background: black;
65         color: white;
66         border: 2px solid white;
67         padding: 10px 15px;
68         font-size: 16px;
69         font-weight: bold;
70         text-transform: uppercase;
71         cursor: pointer;
72         border-radius: 5px;
73         width: 48%;
74     }
75
76     .button:hover {
77         background: white;
78         color: black;
79         border: 2px solid black;
80     }
81
82     @media (max-width: 500px) {
83         .buttons {
84             flex-direction: column;
85             gap: 10px;
86         }
87     }
```

```
86
87     .button {
88         width: 100%;
89     }
90 }
91
92 </style>
93 </head>
94 <body>
95
96     <div class="container">
97         <h2>Definición de Variables Respiratorias</h2>
98
99         <div class="form-group">
100             <label>MODO DE VENTILACIÓN:</label>
101             <select id="modo">
102                 <option>VCV</option>
103                 <option>VCP</option>
104             </select>
105         </div>
106
107         <div class="form-group">
108             <label>VT (Volumen Tidal) [300 - 800 mL]:</label>
109             <input type="number" id="vt" min="300" max="800" value="500">
110         </div>
111
112         <div class="form-group">
113             <label>FR (Frecuencia Respiratoria) [12 - 20 rpm]:</label>
114             <input type="number" id="fr" min="12" max="20" value="15">
115         </div>
116
117         <div class="form-group">
118             <label>Ti (Tiempo de inspiración) [0.1 - 10 s]:</label>
119             <input type="number" id="ti" min="0.1" max="10" step="0.1" value="1">
120         </div>
121
122         <div class="form-group">
123             <label>Te (Tiempo de expiración) [0.1 - 10 s]:</label>
124             <input type="number" id="te" min="0.1" max="10" step="0.1" value="1">
125         </div>
126
127         <div class="form-group">
128             <label>Relación I:E [1:X] (X de 1 a 5):</label>
129             <input type="number" id="ie" min="1" max="5" value="2">
130         </div>
131
132         <div class="form-group">
133             <label>Presión Inspiratoria Máxima (PIP) [20 - 80 cmH2O]:</label>
134             <input type="number" id="pip" min="20" max="80" value="30">
135         </div>
136
137         <div class="buttons">
138             <button class="button"
139                 onclick="window.location.href='Inicio.html'">Regresar</button>
140             <button class="button" onclick="iniciarRespiracion()">Iniciar
141                 Respiración</button>
142         </div>
143     </div>
144 </body>
145 </html>
```

```
143 <script>
144     function iniciarRespiracion() {
145         const parametros = {
146             modo: document.getElementById("modo").value,
147             vt: document.getElementById("vt").value,
148             fr: document.getElementById("fr").value,
149             ti: document.getElementById("ti").value,
150             te: document.getElementById("te").value,
151             ie: document.getElementById("ie").value,
152             pip: document.getElementById("pip").value
153         };
154
155         localStorage.setItem("parametros", JSON.stringify(parametros));
156         window.location.href = 'gráficos.html';
157     }
158
159     // Cargar los valores guardados si existen
160     window.onload = function() {
161         const parametrosGuardados = JSON.parse(localStorage.getItem("parametros"));
162         if (parametrosGuardados) {
163             document.getElementById("modo").value = parametrosGuardados.modo;
164             document.getElementById("vt").value = parametrosGuardados.vt;
165             document.getElementById("fr").value = parametrosGuardados.fr;
166             document.getElementById("ti").value = parametrosGuardados.ti;
167             document.getElementById("te").value = parametrosGuardados.te;
168             document.getElementById("ie").value = parametrosGuardados.ie;
169             document.getElementById("pip").value = parametrosGuardados.pip;
170         }
171     };
172 </script>
173 <script>
174     document.querySelectorAll(".button").forEach(button => {
175         button.addEventListener("mouseenter", () => button.style.backgroundColor =
176 "#007bff");
177         button.addEventListener("mouseleave", () => button.style.backgroundColor = "white");
178     });
179 </script>
180 </body>
181 </html>
```

Anexo D.4. Página de gráficos y parámetros

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Monitoreo de Ventilación</title>
7     <style>
8         /* Estilos para los gráficos */
9         .charts-container {
10             flex: 3;
11             display: flex;
12             flex-direction: column;
13             gap: 20px;
14             min-width: 300px;
15         }
16         .chart {
17             background-color: #fff;
```

```
18     border: 1px solid #ccc;
19   }
20   .chart-title {
21     background-color: #000;
22     color: #fff;
23     padding: 10px;
24     font-weight: bold;
25     text-transform: uppercase;
26   }
27   .chart-content {
28     display: flex;
29     align-items: stretch;
30     height: 150px;
31   }
32   .chart-y-axis {
33     background-color: #00bfff;
34     color: #fff;
35     display: flex;
36     flex-direction: column;
37     justify-content: space-between;
38     padding: 10px;
39     font-weight: bold;
40     width: 60px;
41     text-align: right;
42   }
43   .chart-y-axis div {
44     line-height: 1;
45   }
46   .chart-graph {
47     flex: 1;
48     position: relative;
49     background-color: #fff;
50     border-left: 1px solid #ccc;
51   }
52
53   /* Estilos generales */
54   body {
55     font-family: Arial, sans-serif;
56     background-color: black;
57     color: white;
58     display: flex;
59     justify-content: center;
60     align-items: center;
61     min-height: 100vh;
62     margin: 0;
63     padding: 20px;
64   }
65   .container {
66     width: 90%;
67     max-width: 800px;
68     text-align: center;
69   }
70
71   /* Contenedor que agrupa gráficos y parámetros en columnas */
72   .monitoring-wrapper {
73     display: flex;
74     gap: 20px;
75     align-items: flex-start;
76     justify-content: center;
77     flex-wrap: wrap;
78   }
```

```
79
80     /* Parámetros (columna de la derecha) */
81     .parametros {
82         flex: 1;
83         display: flex;
84         flex-direction: column;
85         gap: 10px;
86         min-width: 150px;
87         margin-top: 20px;
88     }
89     .param-box {
90         background: white;
91         color: black;
92         padding: 10px;
93         border-radius: 5px;
94         text-align: center;
95         font-size: 18px;
96         font-weight: bold;
97         width: 120px;
98         border: 2px solid black;
99         margin: 0 auto;
100    }
101
102    /* Botones */
103    .buttons {
104        display: flex;
105        justify-content: space-between;
106
107        margin-top: 20px;
108        gap: 10px;
109        flex-wrap: wrap;
110    }
111    .button {
112        background: black;
113        color: white;
114        border: 2px solid white;
115        padding: 10px 15px;
116        font-size: 16px;
117        font-weight: bold;
118        text-transform: uppercase;
119        cursor: pointer;
120        border-radius: 5px;
121        flex: 1;
122        min-width: 130px;
123    }
124    .button:hover {
125        background: white;
126        color: black;
127        border: 2px solid black;
128    }
129    @media (max-width: 768px) {
130        .monitoring-wrapper {
131            flex-direction: column;
132            align-items: center;
133        }
134    }
135    @media (max-width: 500px) {
136        .buttons {
137            flex-direction: column;
```

```
138         .button {
139             width: 100%;
140         }
141     }
142 </style>
143 </head>
144 <body>
145
146     <div class="container">
147         <h2>Monitoreo de Ventilación</h2>
148         <div class="monitoring-wrapper">
149             <!-- Columna izquierda: Gráficos -->
150             <div class="charts-container">
151                 <!-- Gráfico de Presión -->
152                 <div class="chart">
153                     <div class="chart-title">PRESIÓN (cmH2O)</div>
154                     <div class="chart-content">
155                         <div class="chart-y-axis">
156                             <div>25</div>
157                             <div>12.5</div>
158                             <div>0</div>
159                         </div>
160                         <div class="chart-graph"></div>
161                     </div>
162                 </div>
163                 <!-- Gráfico de Flujo -->
164                 <div class="chart">
165                     <div class="chart-title">FLUJO (L/min)</div>
166                     <div class="chart-content">
167                         <div class="chart-y-axis">
168                             <div>30</div>
169                             <div>15</div>
170                             <div>0</div>
171                         </div>
172                         <div class="chart-graph"></div>
173                     </div>
174                 </div>
175                 <!-- Gráfico de Volumen -->
176                 <div class="chart">
177                     <div class="chart-title">VOLUMEN (mL)</div>
178                     <div class="chart-content">
179                         <div class="chart-y-axis">
180                             <div>800</div>
181                             <div>400</div>
182                             <div>0</div>
183                         </div>
184                         <div class="chart-graph"></div>
185                     </div>
186                 </div>
187             </div>
188             <!-- Columna derecha: Parámetros -->
189             <div class="parametros">
190                 <div class="param-box"><span>Modo</span><br><span id="modo">--</span>
191
192                 <div class="param-box"><span>VT (mL)</span><br><span id="vt">--</span>
193
194                 <div class="param-box"><span>FR (rpm)</span><br><span id="fr">--</span>
195
196                 <div class="param-box"><span>Ti (s)</span><br><span id="ti">--</span>
197
198                 <div class="param-box"><span>Te (s)</span><br><span id="te">--</span>
199             </div>
200         </div>
201     </div>
202 </body>
203 </html>
```

```
195         <div class="param-box"><span>I:E</span><br><span id="ie">--</span></div>
196         <div class="param-box"><span>PIP (cmH2O)</span><br><span id="pip">--
</span></div>
197     </div>
198 </div>
199 <!-- Botones -->
200 <div class="buttons">
201     <button class="button"
onclick="window.location.href='variables.html'">Retroceder</button>
202     <button class="button" onclick="alert('Ventilación en
curso...')">Detener</button>
203 </div>
204 </div>
205
206 <script>
207     function cargarParametros() {
208         const parametros = JSON.parse(localStorage.getItem("parametros"));
209         if (parametros) {
210             document.getElementById("modo").textContent = parametros.modo;
211             document.getElementById("vt").textContent = parametros.vt;
212             document.getElementById("fr").textContent = parametros.fr;
213             document.getElementById("ti").textContent = parametros.ti;
214             document.getElementById("te").textContent = parametros.te;
215             document.getElementById("ie").textContent = "1:" + parametros.ie;
216             document.getElementById("pip").textContent = parametros.pip;
217         }
218     }
219     window.onload = cargarParametros;
220 </script>
221 <!-- Aquí va el script al final del body -->
222 <script>
223     async function obtenerDatos() {
224         try {
225             const respuesta = await fetch('/api/data'); // Petición a Flask
226             const datos = await respuesta.json();
227
228             // Actualizar valores en la interfaz
229             document.getElementById("pip").innerText = datos.pip + " cmH2O";
230             document.getElementById("peep").innerText = datos.peep + " cmH2O";
231             document.getElementById("flow").innerText = datos.flow + " L/min";
232             document.getElementById("fio2").innerText = datos.fio2 + "%";
233             document.getElementById("vt_meas").innerText = datos.vt_meas + " ml";
234         } catch (error) {
235             console.error("Error al obtener datos:", error);
236         }
237     }
238
239     // Actualizar datos cada 2 segundos
240     setInterval(obtenerDatos, 2000);
241 </script>
242 </body>
243 </html>
```

Anexo E. Desglose presupuestario

Anexo E.1. Costo total de componentes electrónicos

N°	Componente	Modelo/	Cantidad	Precio unitario	SUBTOTAL
1	Controlador principal	Raspberry Pi 3	1	38,5	38,5
2	Microcontrolador secundario	Arduino Mega	1	27,7	27,7
3	Pantalla LCD táctil	Raspberry Pi Touch Display	1	75	75
4	Sensor de presión	MPX5050DP	1	25,5	25,5
5	Sensor de flujo	YFS 201	1	36,8	36,8
6	Sensor de presión negativa/positiva	XGZP6847	1	4,99	4,99
7	Motor paso a paso	NEMA 17	1	24,95	24,95
8	Driver	A4988	1	4,42	4,42
9	Buzzer	Buzzer KY-006	1	1,5	1,5
10	Pulsador	Pulsador de Emergencia 22mm c/ Traba 1NC	1	3,15	3,15
11	Toma de corriente de CA con interruptor y fusible		1	7,9	7,9
12	Regulador de voltaje DC-DC	STEP-DOWN 20W 2A LM317T	2	4,42	8,84
13	Módulo Conmutador de Emergencia entre Batería y Alimentación DC	YX850	1	6,5	6,5
14	Módulo de carga de Baterías de litio 18650	BMS-4S-30A	1	7,03	7,03
15	Potenciómetro	10kΩ	1	1,26	1,26
16	Capacitor electrolítico	1000μF	1	0,5	0,5
17	Diodo rectificador	300V a 3A	2	0,05	0,1
18	Batería recargable	Batería de litio 18650	12	3,5	42
19	Resucitador Adulto (AMBU)	Masc. Reusable. N.º 5 Y reservorio des 2,5 L. ARG 3896	1	30	30
20	Filtros antibacterianos	ISO 80601-2-12	1	2,99	2,99
SUBTOTAL					\$ 349,63

Tabla 7. Costo total de componentes electrónicos

Anexo E.2. Costos tentativos de materiales

Material	Cantidad Aproximada	Precio Unitario (PYG)	Subtotal (PYG)
Perfiles de aluminio 1 metro	2	12,52	25,0
Tornillos M2,5 x 0,45 mm (acero inoxidable)	20	0,50	10,0
Tornillos M4 x 20 mm (acero inoxidable)	20	0,50	10,0
Tuercas M8x1.25 mm	20	0,50	10,0
varilla roscada 1.25	2	3,00	6,0
Arandelas M2,5	30	0,90	27,0
Soportes de fijación en L	1	12,50	12,5
Cable dupont jumper hembra-macho (20 unidades)	4	2,53	10,1
Cable dupont jumper macho-macho	4	2,53	10,1
Láminas de aluminio 1.25x0.05 m	2	50,00	100,0
Filamento PLA (1 kg)	2	25,30	50,6
Polea de rueda Nema 17	1	6,95	7,0
Polea dentada 60 dientes para correa GT2 5mm Nema 17	1	7,99	8,0
Rulemán 8 mm	6	1,9	11,4
SUBTOTAL			\$297,7

Tabla 8. Costos tentativos de materiales

Anexo E.3. Costo total de programación y mano de obra

Concepto	Detalles / Justificación	Costo (USD)
Desarrollo del Software	Programación en Arduino C++, Python y Flask para la interfaz web	200
Integración de Sensores	Configuración de sensores de presión, flujo y oxígeno	100
Programación de la Interfaz Web	Diseño responsivo y optimización de la interfaz gráfica	150
Mano de Obra - Electrónica	Ensamblaje de componentes, soldadura y cableado	180
Mano de Obra - Mecánica	Montaje del sistema de compresión AMBU y ajustes estructurales	120
Mantenimiento y Soporte (3 meses)	Ajustes de firmware, calibración de sensores y pruebas continuas	61,5
SUBTOTAL		\$811,5

Tabla 9. Costo total de programación y mano de obra

Anexo E.4. Costo del certificado de autorización de la DINAVISA

Certificado	Costo en guaraníes	Equivalente en dólares
Expedición de Certificados: libre venta del producto, habilitación de empresas, certificados de productos	Gs. 490.445	\$62,082

Tabla 10. Costo del certificado de autorización de la DINAVISA