



**UNIVERSIDAD UTE**

**FACULTAD DE CIENCIAS DE LA INGENIERÍA E  
INDUSTRIAS**

**CARRERA DE INGENIERÍA MECATRÓNICA**

**DISEÑO ELECTRÓNICO Y CONTROL DE UN VENTILADOR  
ARTIFICIAL PARA CUIDADOS INTENSIVOS.**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN MECATRÓNICA**

**JOHN ANDRÉS GAVILÁNEZ MARTÍNEZ**

**DIRECTOR: ING. FAUSTO FREIRE, Ph.D.**

**Quito, marzo 2022**

© Universidad UTE. 2021

Reservados todos los derechos de reproducción

# FORMULARIO DE REGISTRO BIBLIOGRÁFICO

## TRABAJO DE TITULACIÓN

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1723954150
APELLIDO Y NOMBRES:	Gavilánez Martínez John Andrés
DIRECCIÓN:	Calle Libertad y Calle Sucre, Quito, Pichincha, Ecuador.
EMAIL:	ironrockjohn@gmail.com
TELÉFONO FIJO:	02-2879-023
TELÉFONO MOVIL:	0993914071

DATOS DE LA OBRA	
TÍTULO:	Diseño electrónico y control de un ventilador artificial para cuidados intensivos.
AUTOR O AUTORES:	Gavilánez Martínez John Andrés
FECHA DE ENTREGA DEL PROYECTO DE TITULACIÓN:	18 de diciembre de 2021
DIRECTOR DEL PROYECTO DE TITULACIÓN:	Ing. Freire Carrera Fausto Rodrigo, PhD.
PROGRAMA	<b>PREGRADO</b> <input checked="" type="checkbox"/> <b>POSGRADO</b> <input type="checkbox"/>
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
RESUMEN: Mínimo 250 palabras	Ante la crisis sanitaria causada por la enfermedad (COVID-19), se evidencio que los hospitales no se prepararon con los insumos necesarios. La falta de equipos médicos en centros hospitalarios fue uno de los principales problemas que se han presentado durante la pandemia. El objetivo del presente trabajo fue realizar el diseño electrónico y de control para un ventilador artificial, que permita suplir las funciones respiratorias de personas con

graves lesiones pulmonares causadas por afecciones relacionadas con el coronavirus tipo 2 (SARS-Cov-2). Se desarrolló un algoritmo de funcionamiento de la respiración, que obtiene y monitorea información en tiempo real del paciente; se diseñó un circuito eléctrico que permite alimentar tanto actuadores y sensores que necesitan un nivel de corriente y voltaje óptimo para su desempeño. Para la interacción entre usuario y dispositivo se implementó una interfaz, que puede ser accedida a través de una pantalla, y que fue desarrollada en JAVA, la cual permitirá al personal a cargo del equipo controlar y monitorear el estado del enfermo. Se obtuvo como resultados de la parte electrónica una fuente múltiple capaz de suministrar energía de 5 y 12 [V], cumpliendo con los voltajes de operación idóneos de los elementos electrónicos considerados. En lo correspondiente a la UI se estructuró un sistema controlado por volumen mismo que puede ser variado en rangos de 0 a 1023 [ml], de la misma manera cuenta con controles que pueden ser modificados en cualquier momento, dependiendo de la afección pulmonar. Al modificar sus parámetros se obtuvieron cambios en las curvas de las variables controladas, estos datos se validaron mediante la visualización de las gráficas, y los valores reflejados en el monitor.

<b>PALABRAS CLAVES:</b>	Ventilador artificial, Control por volumen, Simulación, Diseño electrónico.
<b>ABSTRACT:</b>	<p>In the face of the health crisis caused by the disease (COVID-19), it was evident that hospitals did not prepare with the necessary supplies. The lack of medical equipment in hospitals was one of the main problems that have arisen during the pandemic. The objective of this work was to carry out the electronic and control design for an artificial ventilator, which allows to replace the respiratory functions of people with serious lung injuries caused by conditions related to the coronavirus type 2 (SARS-Cov-2). A breathing operation algorithm was developed, which obtains and monitors real-time information from the patient; an electrical circuit was designed that allows to power both actuators and sensors that need an optimal level of current and voltage for their performance. For the interaction between user and device, an interface was implemented, which can be accessed through a screen, and which was developed in JAVA, which will allow the personnel in charge of the team to control and monitor the status of the patient. The results of the electronic part were obtained as a multiple source capable of supplying energy of 5 and 12 [V], complying with the ideal operating voltages of the electronic elements considered. Regarding the UI, a volume-controlled system was structured that could be varied in ranges from 0 to 1023</p>

	[ml], in the same way it has controls that can be modified at any time, depending on the lung condition. By modifying their parameters, changes were obtained in the curves of the controlled variables, these data were validated by viewing the graphs, and the values reflected in the monitor.
<b>KEYWORDS</b>	Artificial fan, Volume control, Simulation, Electronic design.

Se autoriza la publicación de este Proyecto de Titulación en el Repositorio Digital de la Institución.



f: \_\_\_\_\_

GAVILÁNEZ MARTÍNEZ JOHN ANDRÉS

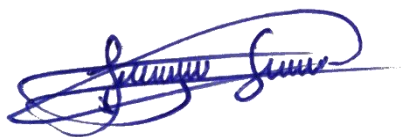
1723954150

## DECLARACIÓN Y AUTORIZACIÓN

Yo, **GAVILÁNEZ MARTÍNEZ JOHN ANDRÉS**, CI 1723954150 autor/a del trabajo de titulación: **DISEÑO ELECTRÓNICO Y CONTROL DE UN VENTILADOR ARTIFICIAL PARA CUIDADOS INTENSIVOS** previo a la obtención del título de **INGENIERO EN MECATRÓNICA** en la Universidad UTE.

1. Declaro tener pleno conocimiento de la obligación que tienen las Instituciones de Educación Superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación de grado para que sea integrado al Sistema Nacional de información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la BIBLIOTECA de la Universidad UTE a tener una copia del referido trabajo de titulación de grado con el propósito de generar un Repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, 20 de diciembre de 2021.

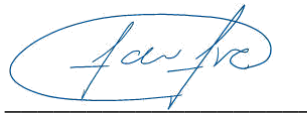


f: \_\_\_\_\_

**GAVILÁNEZ MARTÍNEZ JOHN ANDRÉS**  
1723954150

## CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor de tesis de grado, certifico que el presente trabajo que lleva por título **DISEÑO ELECTRÓNICO Y CONTROL DE UN VENTILADOR ARTIFICIAL PARA CUIDADOS INTENSIVOS** para aspirar al título de **INGENIERO EN MECATRÓNICA** fue desarrollado por **GAVILÁNEZ MARTÍNEZ JOHN ANDRÉS**, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería e Industrias; y que dicho trabajo cumple con las condiciones requeridas para ser sometido a la presentación pública y evaluación por parte del Jurado examinador que se designe.

A handwritten signature in blue ink, appearing to read 'fausto freire', is written over a horizontal line.

(Ing. Fausto Freire, PhD.)

**DIRECTOR DEL TRABAJO**

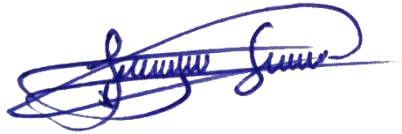
C.I. 1802424737



## DECLARACION JURAMENTADA DEL AUTOR

Yo, John Andrés Gavilánez Martínez, portador(a) de la cédula de identidad N.º 1723954150, declaro que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en ese documento.

La Universidad UTE puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.



f: \_\_\_\_\_

GAVILÁNEZ MARTÍNEZ JOHN ANDRÉS

1723954150

## DEDICATORIA

Quiero dedicar a Dios en primer lugar por permitirme tener un día más de vida, por cumplir una meta más, por la fortaleza, y paciencia para concluir mi carrera de manera exitosa, a mis padres por darme la vida, y que son el pilar fundamental en mi vida, quienes siempre me han apoyado en todo momento.

A mi madre que me dio la vida, quien, con su dedicación y compromiso con su hogar me convirtió en un hombre de bien inculcándome valores, formando mi carácter y guiando mi camino desde pequeño, quien me enseñó el valor de la vida, el esfuerzo, la perseverancia, quien siempre cree en mí, a ella mi principal inspiración a quien siempre le dedicare mis éxitos, gracias a ti por ser mi más preciado tesoro.

A mi padre quien me enseñó desde pequeño a resolver los problemas con sabiduría, quien despertó en mí, uno de mis principales talentos el ingenio, gracias por ser mi ejemplo, eres un hombre trabajador dedicado enteramente con tu hogar me convertiste en un hombre íntegro enseñándome valores como la responsabilidad, puntualidad, el respeto, y la humildad.

A mi hermano quien es mi mejor amigo, quien siempre está en los momentos más complicados, con quien he compartido momentos únicos gracias a ti, a pesar de que eres mi hermano menor, siempre apreciare tus consejos y aprenderé de ti.

A ti que día a día me apoyaste, te preocupaste por mí, estuviste en los momentos más difíciles en mi vida, quien jamás dejó de creer en mí, quien con su sonrisa me inspiro a seguir adelante, eres el motivo de mi felicidad, gracias por ser parte de mi vida y por llegar a ella.

Y en general quiero agradecer a todas las personas que fueron trascendentales en mi vida, a quienes aún están, y las que ya no están siempre las llevare en el corazón, jamás les he fallado y jamás les fallare.

## **AGRADECIMIENTOS**

Principalmente mis agradecimientos a mi tutor de tesis por su colaboración y orientación en el desarrollo de este proyecto, quien me guio, y compartió sus amplios conocimientos en el campo de la tecnología.

Quiero agradecer a cada uno de los Ingenieros que me supieron aconsejar, quiénes fueron guías durante mi vida de estudiante, y aquellos quienes compartieron sus conocimientos.

# INDICE DE CONTENIDOS

	PÁGINA
<b>RESUMEN</b> .....	1
<b>ABSTRACT</b> .....	2
<b>1. INTRODUCCIÓN</b> .....	3
<b>2. METODOLOGÍA</b> .....	10
2.1 REQUERIMIENTOS .....	10
2.2 CARACTERÍSTICAS FISIOLÓGICAS .....	11
2.2.1 GENERACION DE FLUJO .....	11
2.2.2 RESISTENCIA DE LAS VIAS AEREAS .....	11
2.2.3 COMPLIANCIA Y ELASTANCIA .....	12
2.2.4 VENTILACION ARTIFICIAL .....	12
2.2.4.1 Parámetros programados control por volumen .....	12
2.3 DISEÑO CONCEPTUAL .....	14
2.4 DISEÑO ELECTRONICO .....	14
2.4.1 DISEÑO DEL CIRCUITO .....	15
2.4.2 CALCULO DE CONSUMO DE LOS ACTUADORES Y SENSORES .....	17
2.4.3 DISEÑO DE LA FUENTE DE ALIMENTACION .....	18
2.4.3.1 Cálculo de la fuente con salida a 12 y 5 [v] .....	19
2.4.4 SIMULACION DEL CIRCUITO EN PROTEUS .....	28
2.4.5 RASPBERRY PI .....	29
2.4.6 MICROCONTROLADOR .....	30
2.4.7 CIRCUITO DE CONTROL DE VENTILADOR CENTRIFUGO ..	31
2.4.8 CONTROL DE ELECTROVALVULAS SELENOIDES .....	32
2.5 INTEGRACION DEL DISEÑO ELECTRONICO .....	33
2.6 ACTUADORES .....	34
2.7 SENSORES .....	39
2.8 CONEXIÓN DEL VENTILADOR ARTIFICIAL .....	40
2.9 DISEÑO DEL SISTEMA DE CONTROL .....	41
2.10 SELECCIÓN DEL MODO DE VENTILACION Y PARAMETROS ...	41
2.11 DIAGRAMA DE FLUJO DEL SISTEMA .....	42
2.12 FUNCIONALIDAD DEL SISTEMA .....	43
2.13 INTERFAZ HOMBRE-MAQUINA .....	43
2.13.1 ELEMENTOS DEL SISTEMA .....	44
2.13.2 CONTROLES .....	45
2.13.3 LECTURA DEL PUERTO Y CONEXIÓN DEL ARDUINO CON EL ORDENADOR .....	45
2.13.4 DATOS ADICIONALES .....	46
2.13.5 PANEL DE GRAFICAS .....	46

2.14	CODIGO DE LA INTERFAZ INTELLIJ IDEA.....	47
2.15	CODIGO EN ARDUINO .....	50
2.16	FUNCIONAMIENTO DE LA INTERFAZ.....	52
2.16.1	CALCULO DEL PESO IDEAL PREDICHO .....	53
2.16.2	PROGRAMACIÓN DEL SISTEMA .....	53
2.16.3	GRAFICAS DEL SISTEMA.....	54
<b>3.</b>	<b>RESULTADO Y DISCUSIÓN .....</b>	<b>56</b>
3.1	CARACTERISTICAS DEL COMPUTADOR.....	56
3.2	RESULTADOS DEL DISEÑO ELECTRONICO .....	56
3.3	RESULTADOS DE LA SIMULACION DE LA FUENTE .....	56
3.4	RESULTADOS DEL ALGORITMO DE CONTROL .....	60
<b>4.</b>	<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>63</b>
4.1	CONCLUSIONES.....	63
4.2	RECOMENDACIONES .....	64
	<b>BIBLIOGRAFIA .....</b>	<b>65</b>
	<b>ANEXOS.....</b>	<b>69</b>

# INDICE DE TABLAS

## PÁGINA

<b>Tabla 1.</b> Análisis del estado ventilatorio en función del pH y la PaCO <sub>2</sub> . .....	12
<b>Tabla 2.</b> Características eléctricas de los actuadores. ....	16
<b>Tabla 3.</b> Consumo de energía de los sensores. ....	16
<b>Tabla 4.</b> Especificaciones del transformador toroidal. ....	18
<b>Tabla 5.</b> Fórmulas para el cálculo del voltaje de rizado. ....	21
<b>Tabla 6.</b> Componentes de la fuente diseñada. ....	29
<b>Tabla 7.</b> Consumo de corriente de los modelos Raspberry Pi. ....	29
<b>Tabla 8.</b> Especificaciones técnicas de Raspberry pi 4 modelo B. ....	30
<b>Tabla 9.</b> Especificación de fuente de alimentación de Raspberry Pi 4. ....	30
<b>Tabla 10.</b> Especificaciones técnicas del Arduino Uno. ....	30
<b>Tabla 11.</b> Componentes del circuito puente H. ....	31
<b>Tabla 12.</b> Componentes del circuito de relés. ....	33
<b>Tabla 13.</b> Especificaciones del ventilador centrifugo. ....	35
<b>Tabla 14.</b> Valores estimados del motor centrifugo. ....	37
<b>Tabla 15.</b> Especificaciones de la electroválvula solenoide. ....	39
<b>Tabla 16.</b> Especificaciones del sensor de flujo. ....	39
<b>Tabla 17.</b> Especificaciones del sensor de oxígeno. ....	40
<b>Tabla 18.</b> Cálculo del peso ideal. ....	53
<b>Tabla 19.</b> Programación del dispositivo controlado por volumen. ....	54

# INDICE DE FIGURAS

## PÁGINA

<b>Figura 1.</b> Curva Flujo vs Tiempo. ....	7
<b>Figura 2.</b> Patrones de flujo. ....	8
<b>Figura 3.</b> Curva Volumen vs Tiempo. ....	8
<b>Figura 4.</b> Diagrama de Requerimientos. ....	10
<b>Figura 5.</b> Presión de resistencia y de elastancia dentro del pulmón. ....	11
<b>Figura 6.</b> Morfologías de patrones de flujo. ....	14
<b>Figura 7.</b> Diagrama del sistema electrónico y de control. ....	14
<b>Figura 8.</b> Diagrama de Definición de Bloques del ventilador artificial. ....	15
<b>Figura 9.</b> Estructura básica de una fuente de alimentación ....	16
<b>Figura 10.</b> Transformador toroidal tap central. ....	18
<b>Figura 11.</b> Diodo BY251. ....	20
<b>Figura 12.</b> Características típicas del diodo. ....	20
<b>Figura 13.</b> Condensador electrolítico 3300 [uf]. ....	22
<b>Figura 14.</b> Regulador LM7812. ....	23
<b>Figura 15.</b> Transistor PNP MJE15033G. ....	23
<b>Figura 16.</b> Conexión del transistor. ....	23
<b>Figura 17.</b> Circuito térmico del transistor. ....	25
<b>Figura 18.</b> Especificación del disipador del transistor. ....	26
<b>Figura 19.</b> Condensador electrolítico 1000 [uf]. ....	26
<b>Figura 20.</b> Regulador de voltaje LM7805. ....	27
<b>Figura 21.</b> Especificación del disipador del regulador. ....	28
<b>Figura 22.</b> Esquema del circuito diseñado. ....	28
<b>Figura 23.</b> Raspberry Pi 4. ....	29
<b>Figura 24.</b> Placa Arduino uno. ....	30
<b>Figura 25.</b> Módulo LN298 puente H. ....	31
<b>Figura 26.</b> Esquema del circuito puente H. ....	31
<b>Figura 27.</b> Módulo réle 5 [V] a 4 canales. ....	32
<b>Figura 28.</b> Circuito de relés. ....	32
<b>Figura 29.</b> Circuito electrónico completo. ....	33
<b>Figura 30.</b> Diseño de PCB del circuito. ....	34
<b>Figura 31.</b> Representación 3D del circuito. ....	34
<b>Figura 32.</b> Ventilador Centrifugo de alta presión. ....	35
<b>Figura 33.</b> Modelo de un motor cd. ....	35
<b>Figura 34.</b> Función de transferencia del motor centrifugo. ....	37
<b>Figura 35.</b> Respuesta escalón de la función de transferencia. ....	37
<b>Figura 36.</b> Grafica que relaciona la velocidad angular con el caudal. ....	38
<b>Figura 37.</b> Electroválvula solenoide. ....	39
<b>Figura 38.</b> Sensor de Flujo YF-SR401. ....	39
<b>Figura 39.</b> Sensor de Oxigeno (Grove - Gas Sensor). ....	40
<b>Figura 40.</b> Esquema de conexión del circuito del sistema. ....	40

<b>Figura 41.</b> Flujograma del sistema. ....	42
<b>Figura 42.</b> Caso de uso del ventilador artificial.....	43
<b>Figura 43.</b> Ventana de registro de información. ....	44
<b>Figura 44.</b> Ventana de configuración .....	44
<b>Figura 45.</b> Controles de la aplicación.....	45
<b>Figura 46.</b> Control para la comunicación entre Arduino y ordenador. ....	45
<b>Figura 47.</b> Panel de datos Calculados. ....	46
<b>Figura 48.</b> Panel de graficas volumen y flujo con respecto al tiempo.....	46
<b>Figura 49.</b> Comunicación serial, envío y recepción de datos. ....	47
<b>Figura 50.</b> Programación búsqueda de puertos. ....	47
<b>Figura 51.</b> Uso de hilos para el envío y funcionamiento del sistema.....	48
<b>Figura 52.</b> Diseño de la Ventana registro.....	48
<b>Figura 53.</b> Diseño de la ventana de configuración. ....	49
<b>Figura 54.</b> Cálculos necesarios para el algoritmo de control.....	49
<b>Figura 55.</b> Algoritmo de funcionamiento, lectura de datos, y graficas. ....	50
<b>Figura 56.</b> Envío de datos al microcontrolador. ....	50
<b>Figura 57.</b> Recepción de datos en Arduino. ....	51
<b>Figura 58.</b> Control de electroválvulas.....	51
<b>Figura 59.</b> Ficha de registro del paciente .....	52
<b>Figura 60.</b> Datos iniciales previo a la configuración. ....	53
<b>Figura 61.</b> Diagrama Volumen vs Tiempo.....	55
<b>Figura 62.</b> Grafica de Flujo vs Tiempo. ....	55
<b>Figura 63.</b> Diseño electrónico del ventilador artificial. ....	56
<b>Figura 64.</b> Esquema de simulación del circuito. ....	57
<b>Figura 65.</b> Señales generadas por la fuente a 12 [V]. ....	57
<b>Figura 66.</b> Señal generada con carga elevada.....	58
<b>Figura 67.</b> Señal generada con carga disminuida. ....	59
<b>Figura 68.</b> Señales generadas por la fuente a 5 [V]. ....	59
<b>Figura 69.</b> Señales obtenidas del circuito a 5 [V]. ....	60
<b>Figura 70.</b> Simulación con relación 1:1. ....	60
<b>Figura 71.</b> Simulación con relación 1:2. ....	61
<b>Figura 72.</b> Simulación con aumento del VT. ....	61
<b>Figura 73.</b> Simulación con aumento del VT, Fr y tiempo espiratorio. ....	62



# INDICE DE ANEXOS

	<b>PÁGINA</b>
<b>Anexo 1:</b> Diseño Electrónico .....	69
<b>Anexo 2:</b> Diseño Informático .....	79

## RESUMEN

Ante la crisis sanitaria causada por la enfermedad (COVID-19), se evidenció que los hospitales no se prepararon con los insumos necesarios. La falta de equipos médicos en centros hospitalarios fue uno de los principales problemas que se han presentado durante la pandemia. El objetivo del presente trabajo fue realizar el diseño electrónico y de control para un ventilador artificial, que permita suplir las funciones respiratorias de personas con graves lesiones pulmonares causadas por afecciones relacionadas con el coronavirus tipo 2 (SARS-Cov-2). Se desarrolló un algoritmo de funcionamiento de la respiración, que obtiene y monitorea información en tiempo real del paciente; se diseñó un circuito eléctrico que permite alimentar tanto actuadores y sensores que necesitan un nivel de corriente y voltaje óptimo para su desempeño. Para la interacción entre usuario y dispositivo se implementó una interfaz, que puede ser accedida a través de una pantalla, y que fue desarrollada en JAVA, la cual permitirá al personal a cargo del equipo controlar y monitorear el estado del enfermo. Se obtuvo como resultados de la parte electrónica una fuente múltiple capaz de suministrar energía de 5 y 12 [V], cumpliendo con los voltajes de operación idóneos de los elementos electrónicos considerados. En lo correspondiente a la UI se estructuró un sistema controlado por volumen mismo que puede ser variado en rangos de 0 a 1023 [ml], de la misma manera cuenta con controles que pueden ser modificados en cualquier momento, dependiendo de la afección pulmonar. Al modificar sus parámetros se obtuvieron cambios en las curvas de las variables controladas, estos datos se validaron mediante la visualización de las gráficas, y los valores reflejados en el monitor.

**Palabras clave:** Ventilador artificial, Control por volumen, Simulación, Diseño electrónico.

## ABSTRACT

In the face of the health crisis caused by the disease (COVID-19), it was evident that hospitals did not prepare with the necessary supplies. The lack of medical equipment in hospitals was one of the main problems that have arisen during the pandemic. The objective of this work was to carry out the electronic and control design for an artificial ventilator, which allows to replace the respiratory functions of people with serious lung injuries caused by conditions related to the coronavirus type 2 (SARS-Cov-2). A breathing operation algorithm was developed, which obtains and monitors real-time information from the patient; an electrical circuit was designed that allows to power both actuators and sensors that need an optimal level of current and voltage for their performance. For the interaction between user and device, an interface was implemented, which can be accessed through a screen, and which was developed in JAVA, which will allow the personnel in charge of the team to control and monitor the status of the patient. The results of the electronic part were obtained as a multiple source capable of supplying energy of 5 and 12 [V], complying with the ideal operating voltages of the electronic elements considered. Regarding the UI, a volume-controlled system was structured that could be varied in ranges from 0 to 1023 [ml], in the same way it has controls that can be modified at any time, depending on the lung condition. By modifying their parameters, changes were obtained in the curves of the controlled variables, these data were validated by viewing the graphs, and the values reflected in the monitor.

**Keywords:** Artificial fan, Volume control, Simulation, Electronic design.

## **1. INTRODUCCIÓN**

A finales del año 2019 en la ciudad China Wuhan, provincia de Hubei la comisión municipal de salud notifica la existencia de un conglomerado con casos de neumonía. Según estudios realizados en el sitio se determina que todos están causados por un nuevo coronavirus (SARS-CoV-2) causante de la enfermedad COVID-19. A inicios del 2020 ante el aumento de contagios en el lugar la OMS declara a la epidemia como una emergencia de salud pública, y de preocupación internacional (O.M.S, 2020).

Durante los siguientes meses se constata que ningún país a nivel mundial está preparado para una emergencia sanitaria a gran escala, debido a la falta de insumos en los hospitales. Pese al aviso por parte de la OMS de restringir los vuelos y el ingreso de personas provenientes de la región afectada inicialmente, la enfermedad no dio tregua y se extiende en varios países, elevando el número de contagiados de manera exponencial.

El coronavirus es un virus respiratorio por esta razón comienza infectando la garganta, su medio de entrada puede ser por los ojos, boca o la nariz, “se sujeta a las células de la mucosa” expresa William Schaffner, profesor de Medicina Preventiva y Enfermedades Infecciosas del Centro Médico de la Universidad Valderbilt, en Estados Unidos (Plitt, 2020).

Desde tos seca, elevación de temperatura, dolor de cabeza y de garganta, en personas asintomáticas el virus no muestra un grado de letalidad alto que conlleve a una siguiente fase, de manera que en ciertos casos la enfermedad puede ser confundida con gripes y resfriados lo que representa un 80%. Sin embargo, un 15% de los pacientes muestran síntomas graves que requieren hospitalización y en un 5% deben tratarse en unidades de cuidado intensivo (Paules, 2020).

El tiempo de incubación del virus oscila de 1 a 14 días, en la fase viral este se propaga por las vías respiratorias causando dificultades para respirar. Ya para las etapas de gravedad se produce una inflamación de las membranas de los pulmones lo cual provoca un mal funcionamiento del sistema respiratorio, en donde es muy probable que el paciente tenga neumonía (Rivera, 2020).

De manera que se han identificado tres fases en la cual la enfermedad actúa: la primera correspondiente a la infecciosa en la que la mayoría son personas asintomáticas. Una intermedia, en la que se presenta insuficiencia pulmonar, dada por infiltrados en vidrio esmerilado e hipoxemia y una tercera, hiperinflamatoria, en la que podemos ubicar a los pacientes más graves, con disfunción multiorgánica, trastornos de la coagulación y síndrome de dificultad respiratoria aguda (SDRA) (Torres, Botero, & Celis, 2020).

En el caso de que el paciente sufra de SDRA el procedimiento a seguir es el tratamiento mediante ventilación artificial esto con el fin de mejorar los niveles de oxígeno. Cuando los alveolos y los vasos capilares del pulmón resultan

afectados, la sangre y los líquidos se filtran al interior, lo cual impide la entrada de O<sub>2</sub> y a su vez la salida de CO<sub>2</sub> (Patel, 2020).

Durante esta pandemia unos de los principales problemas que se presenta es la falta de insumos en las salas de cuidado intensivo, siendo una de las causas de muerte, ya que no todos los infectados pueden tener acceso a los equipos para su tratamiento, esto debido a que varios centros hospitalarios no cuentan con el espacio para internar a más personas en las UCI. Ya que las cifras varían constantemente lo que provoca que se supere la capacidad máxima de los hospitales, a ello se suma el alto costo para obtener dispositivos de ventilación artificial.

En lo referente a equipos médicos los ventiladores artificiales cuentan con una variedad de diseños con más o con menos funciones. Además de ello existen métodos de ventilación, donde el no invasivos se ha generalizado con un uso extendido en pacientes con fallo respiratorio hipoxémico, en sí se lo considera como un soporte para personas con infección por COVID-19 (Hernandez, 2020).

El uso del método invasivo se considera una medida de soporte que busca reemplazar la función respiratoria del paciente. Para ello, se maneja una serie de elementos físicos que incluyen fuerza (presión), desplazamiento (volumen) y velocidad de cambio en relación con el tiempo (flujo) (Abarca, Vargas, & García, 2020).

Desde los primeros pulmones de acero hasta los ventiladores de última generación la evolución tecnológica día a día promueve grandes avances tanto en los modos de ventilación como en la monitorización de las variables respiratorias (Ernesto & Galdames, 2016). Estos cambios son gracias a la creación de circuitos electrónicos más reducidos, y al desarrollo de softwares que permiten que los sistemas sean menos dependientes de un ser humano.

A partir de la fisiología la función respiratoria es una de las más importantes del cuerpo humano, permite dotar de oxígeno y eliminar dióxido de carbono, en este proceso se encuentra involucrado el torrente sanguíneo que se encarga de distribuir y remover los gases antes mencionados de manera que se realiza el intercambio gaseoso, el cual es de vital importancia, tanto que un sujeto vive días sin alimentarse, pero sin respirar solo en cuestión de minutos bastaría para que fallezca. En reposo se inspira y espira alrededor de 6 litros de aire por minuto (West, 2016).

Para el desarrollo de equipos de ventilación se analiza el funcionamiento y las partes que integran el dispositivo. Previo al diseño y posterior construcción de estos dispositivos, se consideran los parámetros bajo los cuales debe funcionar un ventilador artificial.

Como parte de la programación y manipulación de los controles se utilizan los siguientes valores:

- Presión máxima: 40 [cmH<sub>2</sub>O].
- Presión nominal de funcionamiento: 30 [cmH<sub>2</sub>O].
- Tiempo inspiratorio: 0,6 a 0,8 [s].
- PEEP: 5 a 15 [cmH<sub>2</sub>O].
- Frecuencia de respiración: 7 a 30 por [min].
- Relación entre Inhalación y Exhalación I/E 1:2.
- Volumen de inhalación (Volumen Tidal): 150 cm<sup>3</sup> a 800 cm<sup>3</sup>.
- Regulación de FiO<sub>2</sub>: 60% -100%.

Considerando el uso de alarmas se programan en base a las siguientes necesidades:

- Desconexión del paciente.
- Presión superior a la establecida.
- Volumen de inspiración menor al establecido.
- Desconexión del equipo de la fuente de energía.

Entre las variables más comunes a monitorear están:

- Presión de inspiración.
- Presión de expiración.
- Fracción inspirada de FiO<sub>2</sub>.
- Control visual del funcionamiento general del equipo.

El sistema eléctrico de soporte comprende:

- Batería para funcionamiento por 24 horas.
- Redundancia en alarmas.

En lo referente al circuito ventilatorio el dispositivo contiene filtros tales como:

- Filtro HMEF antes de la entrada/salida de los tubos de respiración de la boca del paciente.
- Filtros HEPA en la salida del aire expirado.

El sistema integra una gran cantidad de componentes y sistemas para su funcionamiento entre los cuales se puede destacar los siguientes:

El panel de programación permite tener conocimiento sobre el estado del individuo, mediante el uso de sensores se puede realizar un testeo en tiempo real de las variables tales como presión, volumen, temperatura, cantidad de oxígeno, flujo de aire mismas que se visualizan en una pantalla. A su vez los parámetros son programados de acuerdo con las condiciones en las que se encuentre el paciente, y son guardadas en un microprocesador el cual desarrollara su algoritmo de funcionamiento.

El sistema electrónico contiene un circuito de alimentación, el mismo que dota de energía a los actuadores, sensores microcontroladores, pantalla entre

otros aspectos. La fuente de poder dentro de un dispositivo es fundamental para su funcionamiento, y más cuando se trata de equipos médicos.

La manipulación de actuadores es otro punto de consideración por tal motivo el sistema electrónico cuenta con un circuito de control. En esta parte es capaz de manipular elementos electrónicos que requieren voltajes operativos altos, al igual que corrientes significativas para la activación de los mismo.

Además de lo mencionado anteriormente estos dispositivos cuentan con un mezclador de aire y oxígeno. En el proceso de inspiración y espiración, el flujo se produce por un generador que es conducido por mangueras o tubos los cuales para dar paso hacia los pulmones o al exterior, son controlados por electroválvulas o válvulas distribuidoras, siendo complementadas por el circuito de ventilación.

En la parte de control se debe tomar en cuenta el aislamiento de señales de tipo analógicas o digitales, para ello es de vital importancia evitar interferencias como consecuencia de la activación de elementos electromecánicos, esto con la finalidad de tener un sistema más estable. Cabe recalcar que no solo se depende de la alimentación para prevenir contratiempos, es conveniente el uso de métodos matemáticos, y componentes electrónicos para filtrar variaciones producidas por corrientes parásitas.

En la actualidad se desarrollan dispositivos mucho más sofisticados, de modo que de a poco se deja de presidir de un sujeto para el manejo de equipos médicos. Mediante el desarrollo de algoritmos de control se brinda soporte al personal médico a cargo de manipular estos aparatos, como se mencionó anteriormente lo que se pretende hoy en día es que obtengan sistemas completamente autónomos.

Para el desarrollo de estos equipos es fundamental tener conocimiento de las variables que se pretende controlar, ya sean estáticas o dinámicas. Al igual que la lectura de las gráficas que se obtienen como resultado de la interacción entre el dispositivo y el paciente, esto con la finalidad de obtener información de su estado.

Las curvas de función respiratoria son la representación gráfica de los cambios que presenta un parámetro fisiológico determinado durante un ciclo respiratorio. Estas variaciones pueden mostrarse en relación con el tiempo o de una variable respecto a la otra, comúnmente son utilizadas para lo siguiente:

- Analizar la fisiopatología de un proceso determinado.
- Detectar cambios en el estado clínico.
- Optimizar una estrategia ventilatoria.



- Valorar la respuesta a un determinado tratamiento o cambio de modo ventilatorio.
- Evitar complicaciones e iatrogenia.
- Evaluar el curso del weaning.
- Establecer un pronóstico en la evolución del paciente.

La ventilación a través de presión positiva se obtiene cuando existe un intercambio entre el dispositivo y los pulmones, mismos que generan un gradiente mismo que producen un flujo. De manera que se puede tener flujos muy altos a volúmenes muy bajos.

Ecuación de movimiento respiratorio expresado en presión:

$$\text{Trabajo VM} = \text{Carga Elástica} + \text{Carga resistiva} \quad [1]$$

Donde:

Carga resistiva: Paso del aire por vía la aérea.

Carga elástica: Elongación de tejido pulmonar.

$$\text{Trabajo VM} = \frac{\text{Volumen}}{\text{distensibilidad}} + \text{Resistencia} \cdot \text{Flujo} \quad [2]$$

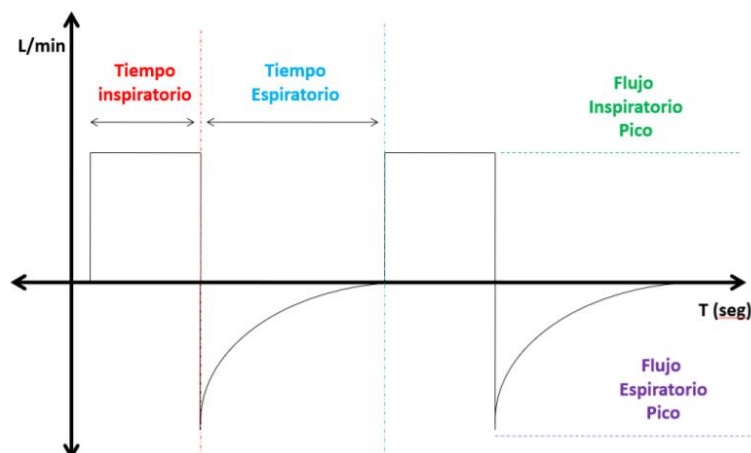
Donde:

Distensibilidad: Es la facilidad que tiene un cuerpo para deformarse.

Elastancia: La capacidad de un cuerpo de volver a su estado inicial.

De manera grafica se observa el comportamiento de las variables controladas (Figura 1). El flujo es considerado como la velocidad de ingreso y salida del aire, por consiguiente, se los conoce como:

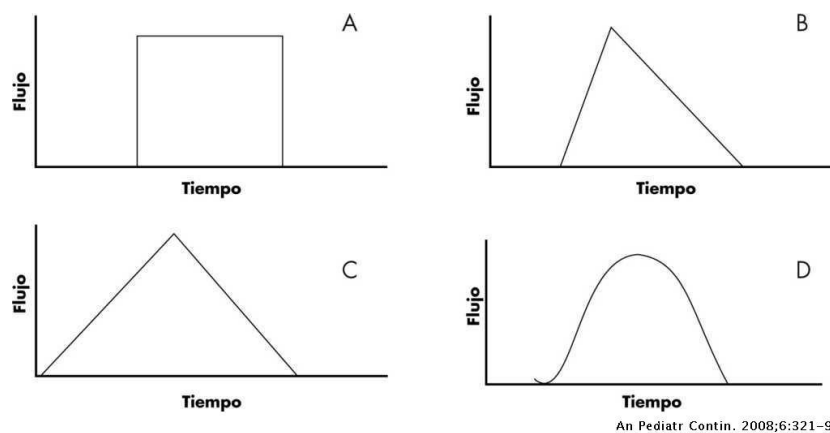
- Flujo inspiratorio: dirección del alveolo.
- Flujo espiratorio: dirección de la boca o saliendo del alveolo.



**Figura 1.** Curva Flujo vs Tiempo.  
(Cediel & David, 2020)

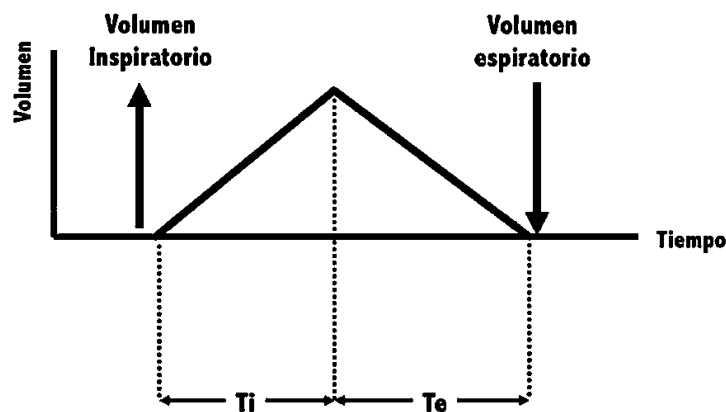
El área bajo la curva del flujo corresponde al volumen corriente movilizado VM. Se debe considerar que en ciertas modalidades quien se encarga de la programación es el operador, en otros casos depende de la mecánica pulmonar y en modos controlados por presión la ventilación es pasiva; por lo tanto, la velocidad espiratoria dependerá de la distensibilidad y la resistencia de la vía aérea.

Los patrones de flujos son una forma de visualizar el modo en el cual esta interactuando el dispositivo con el paciente. En la Figura 2 se observan los que son utilizados usualmente, en donde el acelerante (C) y sinusoidal (D), son programados en casos especiales. Mientras que en modalidades controladas por volumen se programan formas como la constante (A) y desacelerante (B) siendo estos dos los más usados.



**Figura 2.** Patrones de flujo.  
(Ortiz & Dueñas, 2013)

La curva de volumen que se visualiza en la Figura 3, en un tiempo determinado parte desde 0 y llega a un máximo; el cual indica la acción inspiratoria. De la misma manera partiendo del punto más alto existe un descenso hasta 0; en este aspecto se describe la espiración, todo esto sucede constantemente en los dos casos y en tiempos establecidos tanto por el paciente como por el dispositivo.



**Figura 3.** Curva Volumen vs Tiempo.  
(Medina, 2015)

Cabe agregar que es necesario considerar los efectos secundarios que provoca el uso de este tipo de dispositivos, específicamente en la ventilación invasiva. Su uso prolongado causa severos daños en las vías respiratorias donde es introducido el tubo endotraqueal (Facciola, Martinez, Morgade, & Briva, 2017).

Con lo analizado anteriormente surgió la necesidad de diseñar un sistema electrónico y control de un ventilador artificial para cuidados intensivos.

Que tuvo como objetivos específicos los siguientes:

- Determinar los requerimientos del sistema.
- Diseñar el subsistema de alimentación y aislamiento del equipo.
- Diseñar el subsistema electrónico y control.
- Desarrollar la interfaz entre el equipo y el usuario.
- Evaluar el sistema mediante simulaciones.

Además, que el sistema debe cumplir con normas y protocolos médicos, así como también brinde facilidades en su manipulación.

## **2. METODOLOGÍA**

En el presente capítulo se muestra los procedimientos empleados que fueron usados para el desarrollo del sistema en su parte electrónica y de control, así como también la ejecución de cada una de sus etapas. Para el efecto se usó la metodología en V de diseño Mecatrónico. Cabe mencionar que, en lo concerniente a las pruebas de funcionamiento y la evaluación, se verificó el cumplimiento de los objetivos y requerimientos planteados, se utilizó software y hardware.

## 2.1 REQUERIMIENTOS

Previo al diseño electrónico y de control del dispositivo, se determinó los requerimientos (Figura 4) para su posterior desarrollo. Se consideró algunas recomendaciones, relacionadas con la construcción de dispositivos ideales de ventilación artificial, planteados por médicos especialista e instituciones tales como la OMS y OPS.

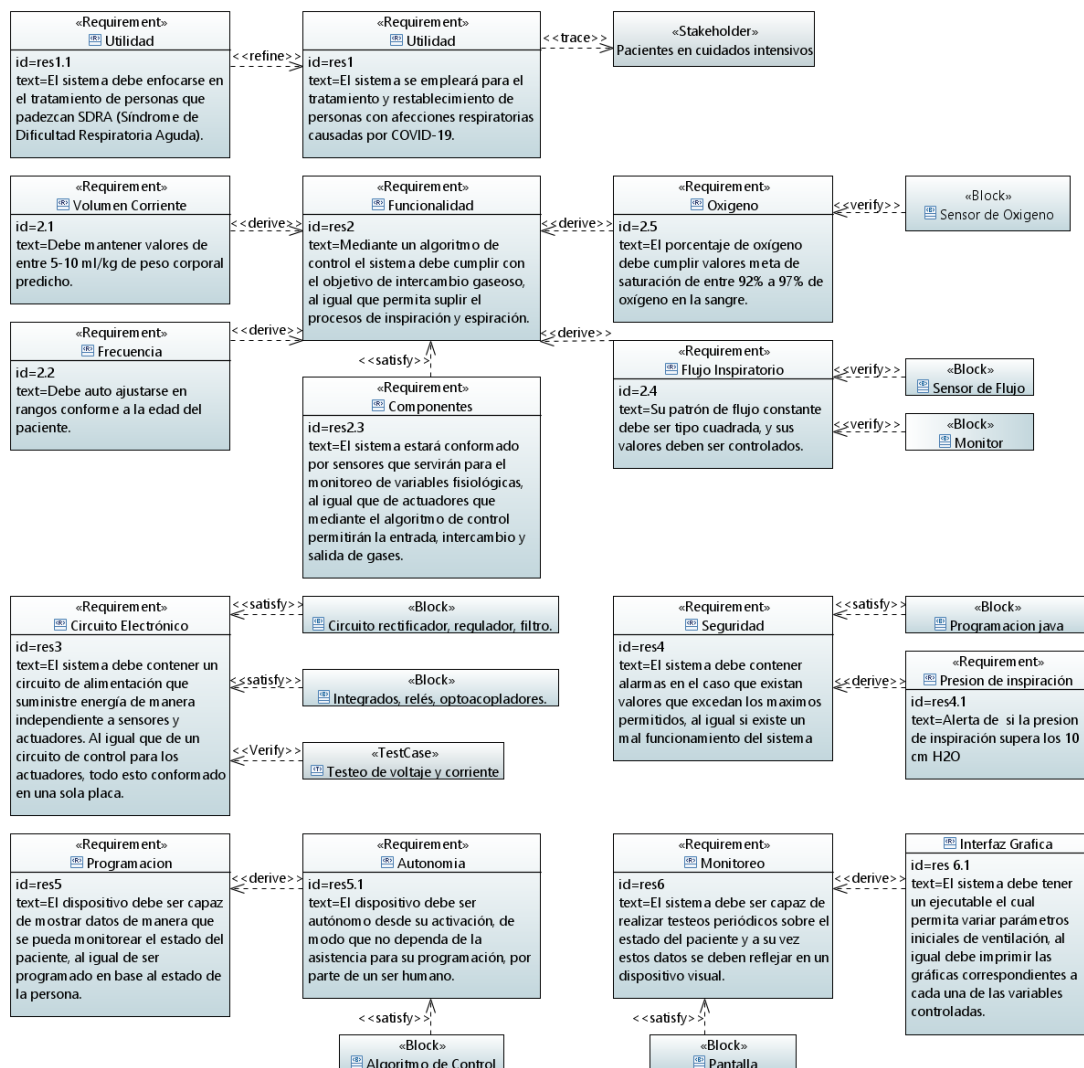


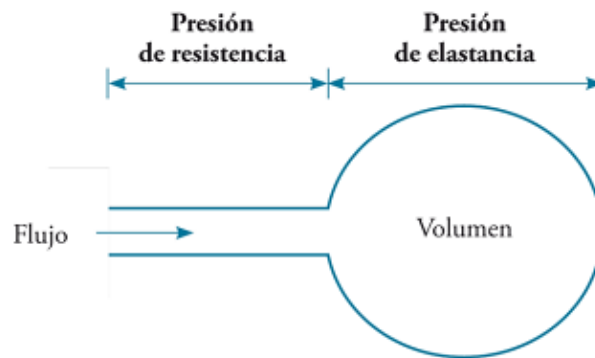
Figura 4. Diagrama de Requerimientos.

## 2.2 CARACTERÍSTICAS FISIOLÓGICAS

Previo al desarrollo del sistema electrónico y de control se realizó un análisis y un estudio sobre las variables fisiológicas que intervienen dentro del funcionamiento del aparato respiratorio. Al igual que se abordó temas relacionados con la programación de ventiladores artificiales convencionales, específicamente en el manejo de sus parámetros.

### 2.2.1 GENERACION DE FLUJO

En lo que corresponde al funcionamiento fisiológico del cuerpo humano esta se genera a través de la contracción y/o expansión de la caja torácica, en la Figura 5, se observa una representación de la entrada de flujo y las variables que se hacen presentes en la misma, como tal el volumen del pulmón aumenta por el descenso del diafragma y se produce una presión atmosférica dentro del tórax (Gómez & Salvador, 2012).



**Figura 5.** Presión de resistencia y de elastancia dentro del pulmón.  
(Gómez & Salvador, 2012)

En la Ecuación 3 se representa el comportamiento de la fisiología respiratoria donde las presiones musculares  $P_{mus}$  generada menos que la elástica de retroceso  $P_{ela}$  es igual al resto de presión por el flujo  $\dot{V}$ .

$$P_{mus} - P_{ela} = P \cdot \dot{V} \quad [3]$$

Durante la espiración e inspiración se genera una presión positiva en el interior de las vías aéreas, este efecto está gobernado por la Ecuación 4, la cual depende de las propiedades resistivas PR y elásticas PE del sistema respiratorio.

$$P_T = P_R + P_E \quad [4]$$

### 2.2.2 RESISTENCIA DE LAS VIAS AEREAS

En lo que corresponde a la entrada y salida de aire que entra a los pulmones, esta tiene que vencer fuerzas opuestas, una de ellas es la resistencia de las vías aéreas la cual se define como la relación entre la variación de presión aplicada sobre el flujo aéreo (Gómez & Salvador, 2012).

Las complicaciones respiratorias se ven relacionadas con la obstrucción e inflamación de las vías aéreas, la resistencia inspiratoria y espiratoria pueden ser diferentes hasta un 15%, siendo este un porcentaje considerable en pacientes con dificultades para respirar. Según estudios clínicos el tubo endotraqueal es la vía aérea que genera más oposición de aire dentro del sistema respiratorio.

### 2.2.3 COMPLIANCIA Y ELASTANCIA

Otro aspecto que se opone al paso del aire es la elastancia y compliancia, que corresponde a las resistencias elásticas del aparato respiratorio, son quienes se oponen a la deformación de las estructuras anatómicas. Se puede definir también como la variación del volumen sobre la presión.

### 2.2.4 VENTILACION ARTIFICIAL

Para el tratamiento de enfermedades respiratorias dependiendo de la patología es necesario el uso de dispositivos como los respiradores artificiales, los cuales permiten suplir funciones que realiza el ser humano al respirar. Previo a la implementación de este dispositivo es pertinente realizar pruebas de gasometría de modo que se conozca sí el paciente tiene complicaciones, mediante el análisis de porcentaje de oxígeno existente en la sangre, se puede determinar si debe o no tener asistencia respiratoria. En la Tabla 1 se observa cómo se identifican situaciones clínicas concretas.

**Tabla 1.** Análisis del estado ventilatorio en función del pH y la PaCO<sub>2</sub>.

	PaCO <sub>2</sub> < 30 [mmHg]	PaCO <sub>2</sub> > 50 [mmHg]
pH > 7.5	Alcalosis respiratoria Hiperventilación	Alcalosis metabólica
pH < 7.3	Acidosis metabólica	Acidosis respiratoria Hipoventilación

(Gómez & Salvador, 2012)

#### 2.2.4.1 Parámetros programados control por volumen

Para la programación del respirador artificial existen varios parámetros a programar, al igual que modos de ventilación. Para el proyecto se consideró el modo más conveniente para el tratamiento de enfermos con patologías relacionadas con COVID-19. Se optó por el desarrollo de un sistema controlado por volumen ya que actualmente en los centros hospitalarios se usa con mayor frecuencia en pacientes con SDRA.

#### Volumen tidal o corriente

Dentro del funcionamiento del sistema corresponde al volumen que entra y sale de los pulmones. El uso de este parámetro netamente sirve para la protección pulmonar el cual depende de la altura del paciente; de otro modo es posible estimar la cantidad máxima de aire que se puede enviar. En el caso

donde se dificulte el uso de instrumentos de medición como una cinta métrica para conocer la estatura del enfermo, se podrá usar valores que mantengan niveles promedio de entre 6 a 7 [mL/kg], que representan los siguientes volúmenes corrientes dependiendo del género:

Hombres: 450 [mL]

Mujeres: 350 [mL]

$$VT = IBW \cdot 0.6 \text{ ml/Kg} \quad [5]$$

Donde:

VT : Volumen Tidal

IBW : Peso predicho

### **Volumen minuto**

El volumen minuto (Ecuación 6) es la cantidad de aire que ingresará a los pulmones en su proceso de inspiración o espiración. Este parámetro se relaciona con la frecuencia respiratoria y generalmente se lo puede medir usando un espirómetro.

$$V_{\text{min}} = Fr * VT \quad [6]$$

Donde:

V<sub>min</sub> : Volumen minuto

Fr : Frecuencia respiratoria

VT : Volumen Tidal

### **Relación espiración, inspiración**

La relación inspiración, espiración, es otro de los controles importantes dentro de la programación del ventilador, puesto que este indica el tiempo en el cual se le insufla aire al paciente, hasta el intercambio gaseoso (Oxígeno-CO<sub>2</sub>) así como también en lo que corresponde a la salida de CO<sub>2</sub> de los pulmones. Lo recomendable en pacientes con SDRA es una I/E de 1/2 a 1/3 esto dará paso al cálculo de los tiempos, para ello se considera la Ecuación 7 (Gómez & Salvador, 2012).

$$Ti = \frac{\left(\frac{60}{Fr}\right)}{Ti + Te} \quad [7]$$

Donde:

Ti : Relación de tiempo inspiratorio.

Te : Tiempo espiratorio.

Fr : Frecuencia respiratoria.

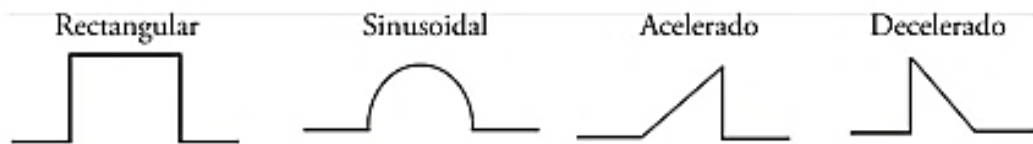


## Tiempo total de ciclo

Es el total de ciclos por minuto que se realizará en la inspiración y la espiración, resulta de dividir 60 segundos para la frecuencia respiratoria.

## Patrón de flujo inspiratorio

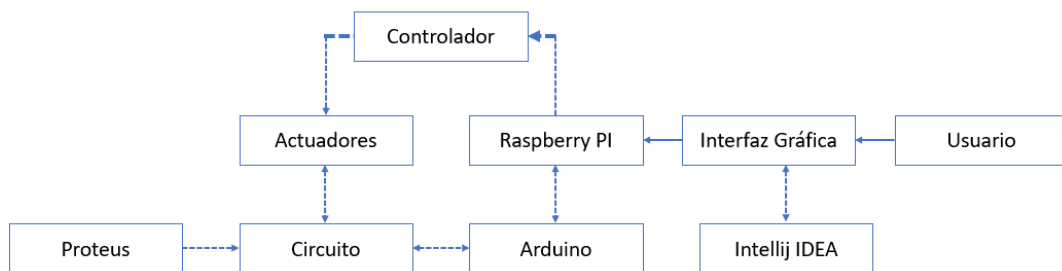
Es necesario saber el de patrón a implementar, ya que esto dependerá del modo de ventilación y la enfermedad que padezca el paciente. Entre los más conocidos están los de tipo cuadrada, ascendente, descendente y sinusoidal como se puede observar en la Figura 6.



**Figura 6.** Morfologías de patrones de flujo.  
(Gómez & Salvador, 2012)

## 2.3 DISEÑO CONCEPTUAL

En la Figura 7 se observa el diseño del sistema y como están distribuidos los componentes que se utilizaron para su desarrollo y simulación.

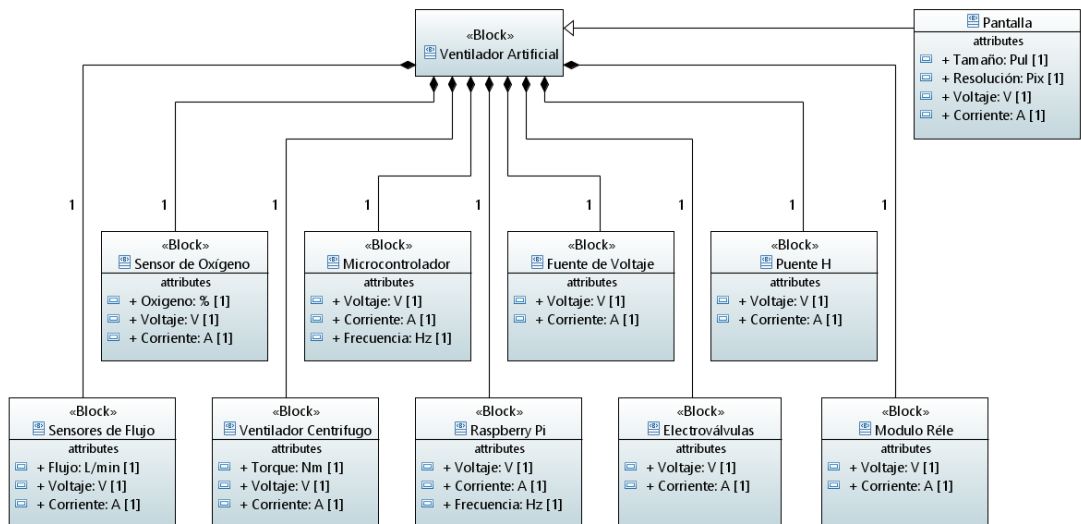


**Figura 7.** Diagrama del sistema electrónico y de control.

Como se observó en el diagrama, el uso de Raspberry y Arduino fue la base del sistema. El primero en el cual se desarrolló la interfaz y el algoritmo de control, y el segundo usado para recibir y enviar información.

## 2.4 DISEÑO ELECTRONICO

Los elementos que se consideraron para el sistema (Figura 8), en su mayor parte son componentes electrónicos tales como sensores, actuadores y microcontrolador, adicional a ello un ordenador de placa reducida, (Raspberry Pi). Donde además se integró una pantalla, misma que no interfiere directamente con el funcionamiento del dispositivo.



**Figura 8.** Diagrama de Definición de Bloques del ventilador artificial.

El diagrama anterior aporta para el desarrollo del sistema de manera significativa, ya que se profundizó en la composición del dispositivo en su parte electrónica. El objetivo principal fue obtener un circuito que sea eficiente, estable, libre de variaciones eléctricas. Además, se logró integrar todos los componentes en una sola placa.

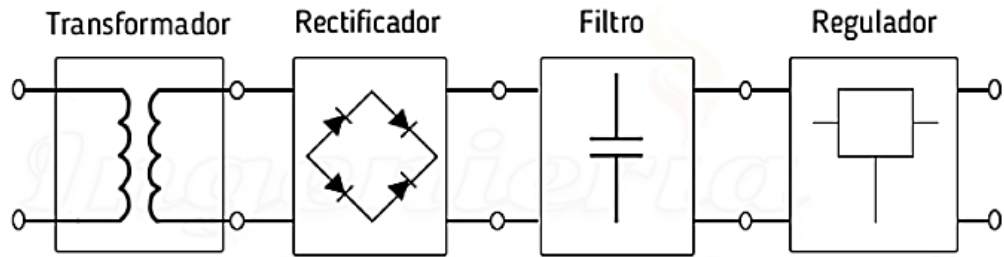
### 2.4.1 DISEÑO DEL CIRCUITO

Para el funcionamiento de un dispositivo es primordial el diseño de un circuito eléctrico, electrónico y de control, de manera que el mismo pueda desempeñar tareas con base en instrucciones o acciones que se realicen al interactuar con este. Uno de los puntos a tratar fue analizar si adquirir o diseñar una fuente de alimentación eléctrica, así como también la forma en la que se manipularían los actuadores mediante la obtención de señales provenientes del microcontrolador.

Al tratarse de un dispositivo que integra microcontroladores, elementos electrónicos, y ordenadores de placa reducida (Raspberry Pi), se realizó la integración de estos dispositivos en un circuito común, con la finalidad de evitar el exceso de cables. Previo al desarrollo de los circuitos se efectuó un estudio sobre el consumo de cada uno de los componentes considerados. Aclarando lo mencionado los actuadores necesitan más corriente a ciertos voltaje operativos debido a que en su funcionamiento se basa netamente en el manejo de flujo y campo electromagnético para ejecutar una acción. En lo que corresponde a sensores usualmente se manejan corrientes muy pequeñas, motivo por el cual se optó por el diseño de una fuente.

En el apartado de requerimientos se especificó que tanto sensores como actuadores serán alimentados de manera independiente, para evitar fallos en el funcionamiento, entre ellos están sobre voltajes, ruido en las señales, daños

en microcontroladores a causa de corrientes provenientes de elementos electromecánicos. A continuación, en la Figura 9 se puede visualizar la estructura de una fuente lineal, la misma que fue la base para su diseño.



**Figura 9.** Estructura básica de una fuente de alimentación  
(Mecafenix, 2018)

La información sobre los parámetros de consumo, fue sustraída de las hojas de datos facilitadas por los fabricantes. El primer circuito de alimentación fue diseñado para suministrar voltajes y corrientes ideales para el funcionamiento de componentes electromecánicos. El segundo se desarrolló con la finalidad de dotar de energía a sensores. A continuación, en la Tabla 2 se describe las características de cada uno de los elementos electrónicos los cuales corresponden a los actuadores del sistema.

**Tabla 2.** Características eléctricas de los actuadores.

Componente	Características
Relé	Voltaje de operación: 250 [VAC] a 30[VAC]
	Voltaje de bobina relé: 5 [V]
	Corriente máxima de operación: 10 [A]
	Consumo de la bobina: 90 [mA]
	Corriente de activación por relé: 10 [mA] a 20 [mA]
Electroválvula Solenoide	Voltaje de operación: 12 [VDC]
	Consumo de la bobina: 170 [mA]
	Potencia Consumo: 2 [W]
Ventilador Centrifugo	Voltaje de operación: 12 [VDC]
	Corriente de operación: 1.97 [A]
	Potencia de consumo: 23.64 [W]

En la Tabla 3, se observa los valores de consumo por parte de los sensores.

**Tabla 3.** Consumo de energía de los sensores.

Componente	Característica
Sensor de Flujo YF-S201	Tipo de sensor: Efecto Hall
	Voltaje de funcionamiento: 5 a 18 [V]
	Consumo máximo de corriente: 15 [mA]
Sensor de Oxígeno (Grove - Gas Sensor)	Tipo de sensor: Análogo
	Voltaje de funcionamiento: 3.3 a 5 [V]
	Consumo máximo de corriente: 86 [uA]

## 2.4.2 CALCULO DE CONSUMO DE LOS ACTUADORES Y SENSORES

Existen componentes que deben ser alimentados acorde al voltaje de operación ideales para su activación; el primer paso fue analizar los elementos que poseen características eléctricas similares. Como se observó en las especificaciones, a los actuadores se les debe suministrar 12 [V] para su accionamiento independientemente de la corriente que necesitan. A continuación, se detalla el cálculo realizado:

### Relés

**Consumo de bobina:** 90 [mA]

$$I_{\text{relé}} = 90 \text{ [mA]} \cdot N_{\text{relé}}$$

$$I_{\text{relé}} = 90 \text{ [mA]} \cdot 3$$

$$I_{\text{relé}} = 270 \text{ [mA]}$$

### Electroválvulas

**Consumo de bobina:** 170 [mA]

$$I_{\text{elect}} = 170 \text{ [mA]} \cdot N_{\text{elect}}$$

$$I_{\text{elect}} = 170 \text{ [mA]} \cdot 3$$

$$I_{\text{elect}} = 510 \text{ [mA]}$$

### Turbina

$$I_{\text{turb}} = 1.97 \text{ [A]}$$

### Corriente total de los actuadores

$$I_{\text{act}} = I_{\text{relé}} + I_{\text{elect}} + I_{\text{turb}} \quad [8]$$

$$I_{\text{act}} = (0.27 + 0.51 + 1.97) \text{ [A]}$$

$$I_{\text{act}} = 2.75 \text{ [A]}$$

Una vez determinado el consumo de los actuadores, se calculó la corriente necesaria para los sensores; para estos componentes es necesario 5 [V] para su correcto funcionamiento. A continuación, se detalla el cálculo realizado:

### Sensor de flujo

**Consumo del sensor:** 15 [mA]

$$I_{\text{sf}} = 15 \text{ [mA]} \cdot N_{\text{sens}}$$

$$I_{\text{sf}} = 15 \text{ [mA]} \cdot 3$$

$$I_{\text{sf}} = 45 \text{ [mA]}$$

### Sensor de oxígeno

$$I_{\text{so}} = 86 \text{ [uA]}$$

### Corriente total de los sensores

$$I_{t_{sen}} = I_{sf} + I_{so} \quad [9]$$
$$I_{t_{sen}} = (45 + 0.086) \text{ [mA]}$$
$$I_{t_{sen}} = 45.086 \text{ [mA]}$$

### 2.4.3 DISEÑO DE LA FUENTE DE ALIMENTACION

Una vez que se obtuvo el consumo de corriente total se realizó el análisis del transformador a utilizar, este componente mediante el efecto de inducción electromagnética y al aplicar una fuerza electromotriz en el devanado primario proveniente de una fuente alterna de voltaje origina un flujo magnético en el núcleo de hierro, mismo que viajara hasta un secundario permitiendo obtener una tensión en un nivel alto o bajo (Chapman, 2012).

Se consideró el uso de un transformador con tap central que permite la entrada de 120 [V] a 60 [Hz], y entrega una salida de 15 [V] en sus secundarios, con una corriente de 4 [A]. Se utilizó los dos devanados del secundario, el primero permitirá la alimentación de los actuadores y el segundo para los sensores. En cuanto al tipo se optó por el uso de un toroidal (Figura 10) ya que frente a uno convencional su rendimiento es superior, posee menos dispersión electromagnética, bajo nivel de ruido, menor calentamiento.



**Figura 10.** Transformador toroidal tap central.  
(multicom, 2018)

A continuación, en la Tabla 4 se observa las especificaciones del transformador toroidal.

**Tabla 4.** Especificaciones del transformador toroidal.

Parámetros	Características
Voltaje de entrada	115 [VAC]
Voltaje de salida	Tap central 15 [V]
Frecuencia	60 [Hz]
Potencia	120 [VA]

### 2.4.3.1 Cálculo de la fuente con salida a 12 y 5 [V]

#### Fuente de alimentación de los actuadores

Se inicio con el cálculo de la fuente que alimentara a los actuadores, cabe mencionar que en esta parte del circuito el consumo total es un dato considerable. Como se analizó anteriormente cada uno de ellos se ven en la necesidad de exigir más corriente para su funcionamiento, la Ecuación 10 fue utilizada para referenciar el voltaje máximo que se requiere del secundario.

#### Voltaje pico

$$V_p = V_{ef} \cdot \sqrt{2} \quad [10]$$

Donde:

$V_p$  : Voltaje pico

$V_{ef}$  : Voltaje eficaz

#### Voltaje de entrada al regulador

Se realizó este cálculo para conocer el voltaje idóneo que debe tener el regulador a su entrada  $V_{i_{reg1}}$ , misma que deber ser superior a su salida  $V_{o_{reg1}}$ . Para ello se sumó 3 [V], esto como recomendación por parte del fabricante para obtener el valor deseado.

$$V_{i_{reg1}} = V_{o_{reg1}} + 3 \text{ [V]}$$

$$V_{i_{reg1}} = 12 + 3 = 15 \text{ [V]}$$

#### Voltaje pico máximo del secundario del transformador

Mediante el uso de la Ecuación 10, se calculó el voltaje pico máximo, para ello el valor eficaz utilizado fue el de entrada al regulador  $V_{i_{reg1}}$ .

$$V_{ps_{max1}} = V_{i_{reg1}} \cdot \sqrt{2}$$

$$V_{ps_{max1}} = 15 \text{ [V]} \cdot \sqrt{2}$$

$$V_{sp_{max1}} = 21.213 \text{ [V]}$$

Se obtuvo  $V_{sp_{max1}}$ , el cual sirve como referencia para saber si el voltaje que suministra el transformador es ideal para la fuente a diseñar. En las especificaciones la salida del transformador alcanza los 15 [V] hablando de su tensión eficaz, su valor pico  $V_{pt}$  es el siguiente:

$$V_{pt} = V_{ef} \cdot \sqrt{2}$$

$$V_{pt} = 15 \text{ [V]} \cdot \sqrt{2}$$

$$V_{pt} = 21.213 \text{ [V]}$$

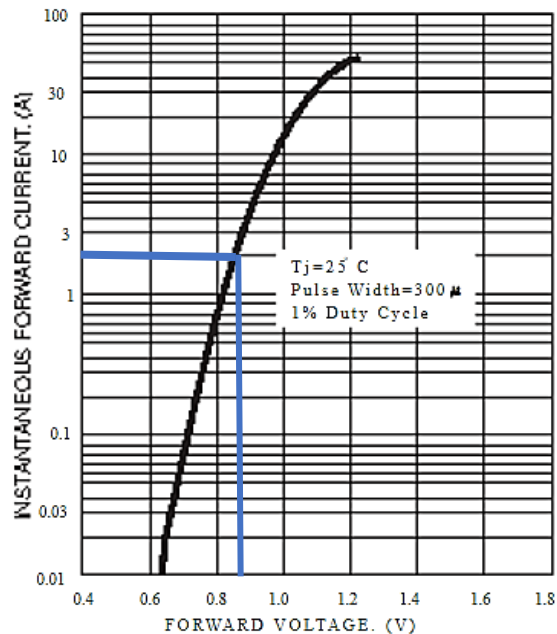
### Voltaje de diodos

Para el diseño se consideró el uso del BY251 (Figura 11), puesto que se diseñó un puente de diodos para la rectificación de la señal, proveniente de uno de los secundarios del transformador. Este componente soporta valores de hasta 1300 [V], y corrientes no más de 3 [A].



**Figura 11.** Diodo BY251.  
(multicomp, 2008)

Continuando con el cálculo se consideró el valor de 1.7 [V] de caída de tensión en los dos diodos que conducen en el puente rectificador, debido a que por los mismos pasa una corriente muy alta y esto demanda un poco más de consumo, en la (Figura 12) se observa el comportamiento del diodo.



**Figura 12.** Características típicas del diodo.  
(multicomp, 2008)

$$V_{\text{diodos}} = 1.7 \text{ [V]}$$

## Voltaje máximo del condensador

$$V_{C_{\max}} = V_{pt} - V_{\text{diodos}}$$

$$V_{C_{\max}} = (21.213 - 1.7) \text{ [V]}$$

$$V_{C_{\max}} = 19.513 \text{ [V]}$$

Con los datos calculados se obtuvo el valor del condensador que servirá para filtrar la señal saliente del puente de diodos. Se utilizó la fórmula que corresponde al cálculo del voltaje de rizado  $V_r$  de onda completa que se observa en la Tabla 5.

**Tabla 5.** Fórmulas para el cálculo del voltaje de rizado.

Tipo	Formulas
Con rectificador de media onda	$V_{r_{pp}} = \frac{I}{f \cdot C}$
Con rectificador de onda completa	$V_{r_{pp}} = \frac{I}{2 \cdot f \cdot C}$

(Boylestad & Nashelsky, 2015)

Donde:

$V_{r_{pp}}$ : Voltaje rizado pico-pico.

$f$ : Frecuencia.

$C$ : Capacitor.

## Voltaje de rizado pico-pico

Para el cálculo se utilizó el voltaje máximo del condensador  $V_{C_{\max}}$  y el valor de entrada al rectificador  $V_{i_{reg1}}$ .

$$V_{r_{pp1}} = V_{C_{\max}} - V_{i_{reg1}}$$

$$V_{r_{pp1}} = (19.513 - 15) \text{ [V]}$$

$$V_{r_{pp1}} = 4.51 \text{ [V]}$$

Seguidamente se realizó el cálculo de la tensión que ingresará al condensador  $V_{Con1}$  el cual equivale a un valor promedio. Se utilizó la Ecuación 11, donde se consideró al voltaje máximo  $V_{C_{\max}}$ , al igual que el dato obtenido anteriormente  $V_{r_{pp1}}$ .

$$V_{Con1} = V_{C_{\max}} - \frac{V_{r_{pp1}}}{2} \quad [11]$$

$$V_{Con1} = 19.513 - \frac{4.51}{2}$$

$$V_{Con1} = 17.26 \text{ [V]}$$



## Capacitor

Para el cálculo se utilizó la Ecuación 12, donde el valor del corriente usado fue de 3 [A], un valor aproximado a la salida del transformador.

$$V_{r_{pp1}} = \frac{I}{2 \cdot f \cdot C} \quad [12]$$
$$4.51 = \frac{3 \text{ [A]}}{2 \cdot 60 \text{ [Hz]} \cdot C1}$$
$$C1 = 5543.23 \text{ [uf]}$$

Ya que no se encontró un capacitor similar al dato calculado, se consideró el uso de un componente con un valor superior y que sea comercial. Cabe mencionar que la formula aplicada proporciona un aproximado, adicional a ello es recomendable el uso de dos condensadores en la etapa de filtrado. Los elementos que se consideraron fueron de 3300 [uf] a 35 [V], (Figura 13) mismo que puede ser ajustado en la simulación analizando el rizado que este nos proporcione.



**Figura 13.** Condensador electrolítico 3300 [uf].  
(Panasonic, RS-online, 2015)

Una vez determinado el filtro del condensador, es de suma importancia comprobar de manera visual si una parte del circuito se encuentra activado y funcionando, para ello se colocó un led indicador a la entrada de los circuitos de alimentación. Junto al led se le ubico una resistencia en serie que servirá como protección, misma que se calculó de la siguiente manera:

### Resistencia para led indicador

Se asumió un valor de corriente de  $I_1 = 10 \text{ [mA]}$ .

$$R_1 = \frac{V_{Con1} - V_{led}}{I_1}$$
$$R_1 = \frac{(17026 - 2) \text{ [V]}}{10 \text{ [mA]}}$$
$$R_1 = 1.5 \text{ [K}\Omega\text{]}$$

Por el momento se cumplió 2 etapas para obtener una fuente regulada; la rectificación y el filtro, ahora se procede a regular. En este punto se aclara que el regulador no proporciona la corriente necesaria para alimentar a los actuadores puesto a que aporta cantidades muy bajas; para satisfacer esta necesidad se utilizó un transistor.

El regulador que se consideró es el LM7812 (Figura 14), el cual proporciona un voltaje de salida regulada a 12 [V] a partir de una entrada de voltajes en un rango de 14.5 [V] a 30 [V].



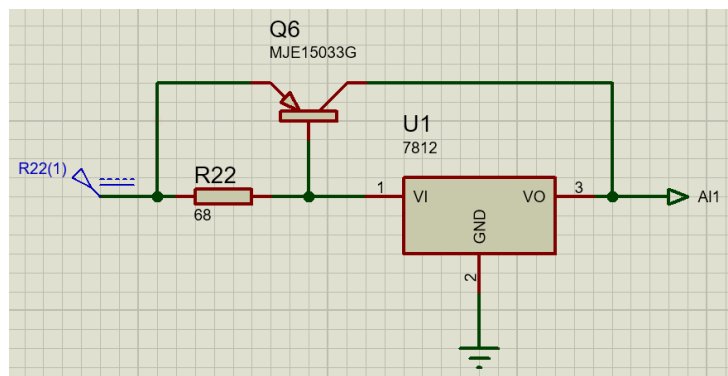
**Figura 14.** Regulador LM7812.  
(FAIRCHILD, 2014)

El transistor MJE15033G de tipo PNP (Figura 15) fue seleccionado con la finalidad de regular la corriente. Como mínimo se necesita un valor de 2.81 [A] mismo que fue obtenido del consumo total de los actuadores; este componente soporta en el colector un máximo de 8 [A].



**Figura 15.** Transistor PNP MJE15033G.  
(Semiconductor, 2011)

A continuación, se muestra la ubicación y el modo de conexión del transistor (Figura 16).



**Figura 16.** Conexión del transistor.

Como se aprecia en la imagen los dos componentes tienen dos puntos de conexión común. La primera es en la base del transistor con la entrada del regulador, acompañados de una resistencia de 68  $[\Omega]$  este valor es asumible ya que en la simulación puede ser ajustado; por el circulará una corriente menor a la que ingresa por el emisor.

El valor de la resistencia es inversamente proporcional a la corriente del colector. El otro punto común de conexión es la salida de los dos componentes. Para proteger al transistor de temperaturas muy altas se colocó un disipador.

### **Cálculo del disipador del transistor**

Como datos principales se obtuvo la resistencia térmica entre la unión y el ambiente  $R_{th_{j-a}}$ , de este modo se determinó si el transistor necesita un disipador. Además de ello se necesitó conocer  $R_{th_{j-c}}$  que corresponde a la capacidad de oponerse al flujo de temperatura en la parte interna con la carcasa.

El valor  $R_{th_{j-a}}$  equivale a 62.5  $[\text{°C/W}]$  y  $R_{th_{j-c}}$  a 2.5  $[\text{°C/W}]$  estos valores se obtuvieron de la hoja de datos del fabricante. Se calculó la potencia promedio disipada por el transistor PT para ello se utilizó las siguientes formulas:

$$PT = V_{EC} \cdot I_c \quad [13]$$

Donde:

PT : Potencia disipada por el transistor.

$V_{EC}$  : Voltaje de emisor colector.

$I_c$  : Corriente de colector.

$$V_{EC} = V_{Con} - V_{o_{reg}} \quad [14]$$

Donde:

$V_{Con}$ : Voltaje del condensador.

$V_{o_{reg}}$ : Voltaje de salida del regulador

Para el cálculo del voltaje que ingresa al emisor colector se utilizó la Ecuación 14 donde la salida del regulador  $V_{o_{reg1}}$  corresponde a 12 [V].

$$V_{EC} = V_{Con1} - V_{o_{reg1}}$$

$$V_{EC} = (17.26 - 12)[V]$$

$$V_{EC} = 5.26 [V]$$

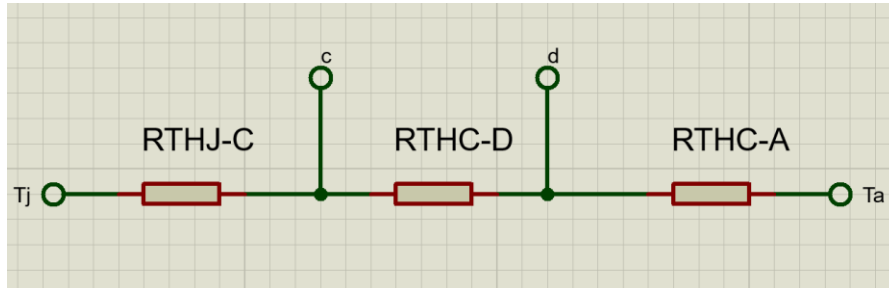
Mediante el uso de la Ecuación 13 se procedió al cálculo de la potencia promedio que debe disipar el transistor a una corriente de 3[A] como resultado se obtuvo:

$$PT = V_{EC} \cdot I_c$$

$$PT = 5.26 [V] \cdot 3[A]$$

$$PT = 15.78 [W]$$

Se analizó el circuito térmico del transistor (Figura 17), de las tres resistencias dos de ellas son conocidas. Se realizó el cálculo de la resistencia térmica del disipador y el ambiente  $R_{th_{d-a}}$ , para ello se asumió un  $R_{th_{c-d}}$  con un valor de  $1 [^{\circ}C/W]$ .



**Figura 17.** Circuito térmico del transistor.

Se utilizó la Ecuación [15, donde se consideró las temperaturas de la unión  $T_j$ , y ambiente  $T_a$ , adicionalmente se tomó en cuenta la recomendación del fabricante de no superar los  $150 [^{\circ}C]$ . Cabe indicar que se trabajó con un factor de seguridad de 0.7, al igual que un  $T_a$  con un valor de  $25 [^{\circ}C]$ .

$$T_j - T_a = PT \cdot (R_{th_{j-c}} + R_{th_{c-d}} + R_{th_{d-a}}) \quad [15]$$

Donde:

- $T_j$  : Temperatura de la unión interna del transistor.
- $T_a$  : Temperatura ambiente.
- $PT$  : Potencia disipada.
- $R_{th_{j-c}}$  : Resistencia térmica de la unión a la carcasa.
- $R_{th_{c-d}}$  : Resistencia térmica de la carcasa al disipador.
- $R_{th_{d-a}}$  : Resistencia térmica del disipador al ambiente.

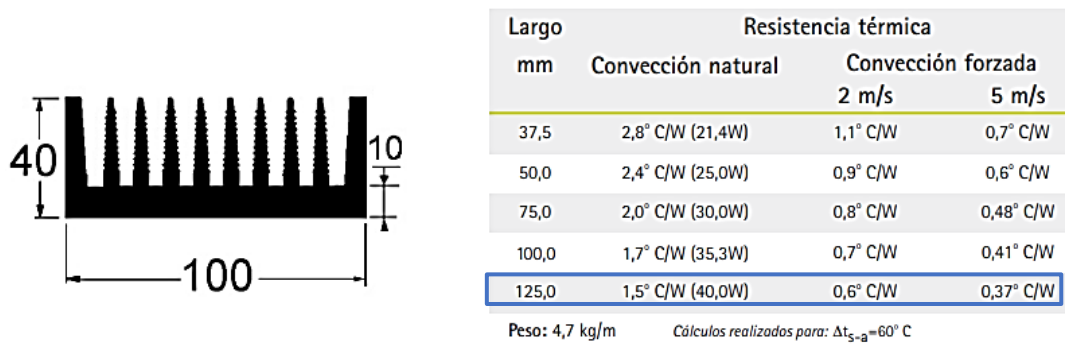
Se usó la Ecuación 15 para obtener el valor de la resistencia térmica del disipador al ambiente  $R_{th_{d-a}}$ .

$$R_{th_{d-a}} = \frac{T_j - T_a}{PT} - (R_{th_{j-c}} + R_{th_{c-d}})$$

$$R_{th_{d-a}} = \frac{((0.7 \cdot 150) - 25) [^{\circ}C]}{15.78 [W]} - (2.5 + 1) [^{\circ}C/W]$$

$$R_{th_{d-a}} = 1.569 [^{\circ}C/W]$$

Con el dato calculado se buscó un valor igual o menor, y se eligió el disipador ideal entre los que se muestran en la Figura 18 (DISIPA, 2019).



**Figura 18.** Especificación del disipador del transistor.  
(DISIPA, 2019)

Como se observó en la imagen se tiene el dato de la longitud del disipador que corresponde a 125 [mm], entre otros como la velocidad para agregar un ventilador y mejorar la refrigeración.

### Fuente de alimentación de los sensores

En este punto se procedió con el diseño de la fuente de alimentación para los sensores, cuyo valor de consumo es 5 [V], para ello se hizo uso de la otra salida del transformador toroidal, cabe mencionar que se consideró los valores calculados anteriormente.

### Capacitor

Para el cálculo se utilizó la Ecuación [12] y el dato de la corriente de 0.5 [A], un valor aproximado a la salida del secundario.

$$V_{r_{pp2}} = \frac{I}{2 \cdot f \cdot C}$$

$$C2 = \frac{0.5 \text{ [A]}}{2 \cdot 60 \text{ [Hz]} \cdot 4.51 \text{ [V]}}$$

$$C2 = 923.87 \text{ [uf]}$$

El condensador que se consideró fue de un valor de 1000 [uf] a 35 [V], (Figura 19) puesto a que son comerciales en comparación al obtenido en el cálculo.



**Figura 19.** Condensador electrolítico 1000 [uf].  
(Panasonic, 2014)

El regulador de voltaje que se consideró en esta parte fue el LM7805 (Figura 20) el cual provee un valor de 5 [V], con una corriente máxima de 1 [A]. Ya que su consumo en comparación con el circuito anteriormente diseñado es menor, se descartó la posibilidad elegir un transistor.



**Figura 20.** Regulador de voltaje LM7805.  
(life.augmented, 2014)

### Cálculo del disipador del regulador

Como en el anterior caso se procedió a obtener los valores de las resistencias térmicas,  $R_{thj-a}$  equivale a 65 [°C/W] y  $R_{thj-c}$  a 5 [°C/W] estos valores se obtuvieron de la hoja de datos del fabricante.

Se calculo la potencia promedio disipada por el regulador PTR.

$$\begin{aligned} V_{pr} &= (V_{i_{reg2}} - V_{o_{reg1}}) + V_{r_{pp2}} \\ V_{pr} &= (15 - 5) + 4.51 \\ V_{pr} &= 14.51 [V] \end{aligned}$$

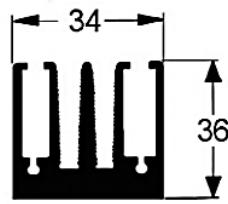
Mediante el uso de la Ecuación [13] se procedió al cálculo de la potencia promedio que debe disipar el regulador a una corriente de 0.5[A], como resultado se obtuvo:

$$\begin{aligned} PTR &= V_{pr} \cdot I_c \\ PTR &= 14.51 [V] \cdot 0.5[A] \\ PTR &= 7.26 [W] \end{aligned}$$

A continuación, se realizó el cálculo de la resistencia térmica del disipador al ambiente  $R_{th_{d-a}}$ , para ello se usó la Ecuación [15].

$$\begin{aligned} R_{th_{d-a}} &= \frac{((0.7 \cdot 150) - 25) [^{\circ}C]}{7.26 [W]} - (5 + 1)[^{\circ}C/W] \\ R_{th_{d-a}} &= 5.019 [^{\circ}C/W] \end{aligned}$$

Con el dato que se obtuvo nos dirigimos a la página del fabricante de manera que elegiremos el disipador ideal; como se puede observar en la Figura 21.



Largo mm	Resistencia térmica		
	Convección natural	Convección forzada	
		2 m/s	5 m/s
30,0	6,6° C/W (9,1W)	2,8° C/W	1,5° C/W
37,5	5,8° C/W (10,3W)	2,3° C/W	1,3° C/W
50,0	5,0° C/W (12,0W)	1,9° C/W	1,2° C/W
75,0	4,1° C/W (14,6W)	1,6° C/W	1,0° C/W
100,0	3,5° C/W (17,1W)	1,4° C/W	0,9° C/W

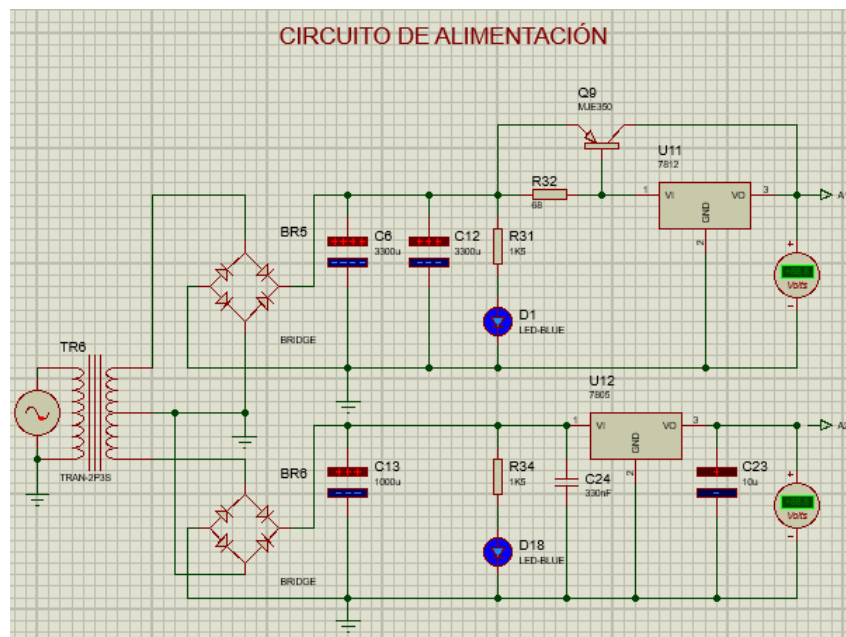
Peso: 1,56 kg/m      Cálculos realizados para:  $\Delta t_{s-a} = 60^\circ \text{C}$

**Figura 21.** Especificación del dissipador del regulador.  
(DISIPA, 2019)

#### 2.4.4 SIMULACION DEL CIRCUITO EN PROTEUS

Para la simulación se utilizó la aplicación Proteus, la cual permite desarrollar, construir proyectos electrónicos en varias etapas tales como su diseño, esquema y programación al igual que posibilita la creación de placas (Proteus, 2021). Se diseñó en el software mencionado considerando los datos calculados anteriormente, de manera que se pueda analizar el comportamiento del circuito desde la entrada de la fuente de energía en corriente alterna hasta sus valores regulados.

Cabe mencionar que dentro de la aplicación se hizo uso de las herramientas que este programa nos proporciona, tales como el osciloscopio para visualizar el comportamiento en cada parte del circuito, al igual que el voltímetro y amperímetro para verificar que los valores cumplan con lo requerido. A continuación, en la Figura 22 se presenta el esquema.



**Figura 22.** Esquema del circuito diseñado.

En la Tabla 6 se detalla los componentes electrónicos que se encuentran en el esquema de la fuente múltiple diseñada.

**Tabla 6.** Componentes de la fuente diseñada.

Componente	Cantidad	Tipo
Diodos	8	BY251
Capacitores Electrolíticos	2	3300 [ <i>uf</i> ]
Capacitores Electrolíticos	1	1000 [ <i>uf</i> ]
Capacitores Electrolíticos	1	10 [ <i>uf</i> ]
Capacitores Cerámico	1	330 [ <i>nf</i> ]
Resistencia	2	1.5 [ <i>KΩ</i> ]
Resistencia	1	68 [ <i>Ω</i> ]
Leds	2	Azul
Transistor	1	MJE15033G
Regulador	1	LM7812
Regulador	1	LM7805

### 2.4.5 RASPBERRY PI

Este ordenador controlara el funcionamiento del dispositivo, cabe añadir que fue donde se instaló la aplicación que contiene el algoritmo de control. Además, como parte del análisis en la Tabla 7 se detallan los parámetros requeridos para la alimentación de los distintos tipos de Raspberry existentes.

**Tabla 7.** Consumo de corriente de los modelos Raspberry Pi.

Modelos de Raspberry Pi	Alimentación necesaria
Raspberry Pi modelo A	700 [mA]
Raspberry Pi Modelo B	1.2 [A]
Raspberry Pi Modelo A+	700 [mA]
Raspberry Pi Modelo B+	1.8 [A]
Raspberry Pi 2 Modelo B	1.8 [A]
Raspberry Pi 3 Modelo B	2.5 [A]
Raspberry Pi 3 Modelo A+	2.5 [A]
Raspberry Pi 3 Modelo B+	2.5 [A]
Raspberry Pi 4 Modelo B	3 [A]
Raspberry Pi Zero W/WH	1.2 [A]
Raspberry Pi Zero	1.2 [A]

(Sergio, 2019)

En la tabla se evidencio que el consumo de corriente varía dependiendo del tipo. Para el proyecto se consideró utilizar el Raspberry Pi 4 modelo B (Figura 23), el cual es el más utilizado actualmente.



**Figura 23.** Raspberry Pi 4.  
(Rus, 2019)



En la Tabla 8 se observa las características del modelo.

**Tabla 8.** Especificaciones técnicas de Raspberry pi 4 modelo B.

<b>Procesador</b>	ARM Cortex-A72
<b>Frecuencia</b>	1.5 [GHz]
<b>GPU</b>	Video Core VI (con soporte para OpenGL ES 3.x)
<b>Memoria</b>	1 [GB] / 2 [GB] / 3 [GB] LPDDR4 SDRAM
<b>Conectividad</b>	Bluetooth 5.0, Wi-Fi 802.11ac, Gigabit Ethernet
<b>Puertos</b>	GPIO 40 pines, 2 x micro HDMI, 2 x USB 2.0, 2 x USB 3.0, CSI (cámara) DSI (pantalla táctil) Micro SD Conector de audio jack USB-C (alimentación)

(Rus, 2019)

La fuente de alimentación del modelo considerado tiene las siguientes especificaciones, a continuación, se detalla en la Tabla 9.

**Tabla 9.** Especificación de fuente de alimentación de Raspberry Pi 4.

<b>Parámetros</b>	<b>Características</b>
Voltaje de entrada	100 a 240 [VAC]
Voltaje de salida	5.1 [V]
Corriente	3 [A]
Potencia Nominal.	5.3 [W]

(Raspberry, 2019)

## 2.4.6 MICROCONTROLADOR

Para el sistema se consideró el uso de Arduino (Figura 24), no fue incluida dentro del circuito a pesar de que el esquema para su fabricación es de libre uso. Por temas de facilidad de alimentaciones eléctrica, conexión y comunicación se decidió optar por la placa electrónica.



**Figura 24.** Placa Arduino uno.  
(Guerrero, 2014)

En la Tabla 10 se describe las principales características del Arduino UNO.

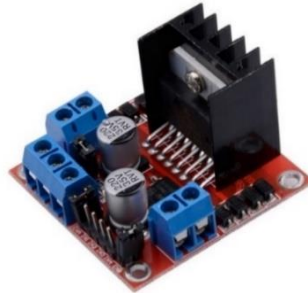
**Tabla 10.** Especificaciones técnicas del Arduino Uno.

<b>Microcontrolador</b>	ATmega328
<b>Voltaje Operativo</b>	5 [V]
<b>Voltaje de Entrada</b>	7 [V] a 12[V]
<b>Entradas/Salidas Digital</b>	14 (De las cuales 6 son salidas PWM)
<b>Entradas Análogas</b>	Pines de Entradas Análogas: 6
<b>Memoria Flash</b>	Memoria Flash: 32 [KB]

(Guerrero, 2014)

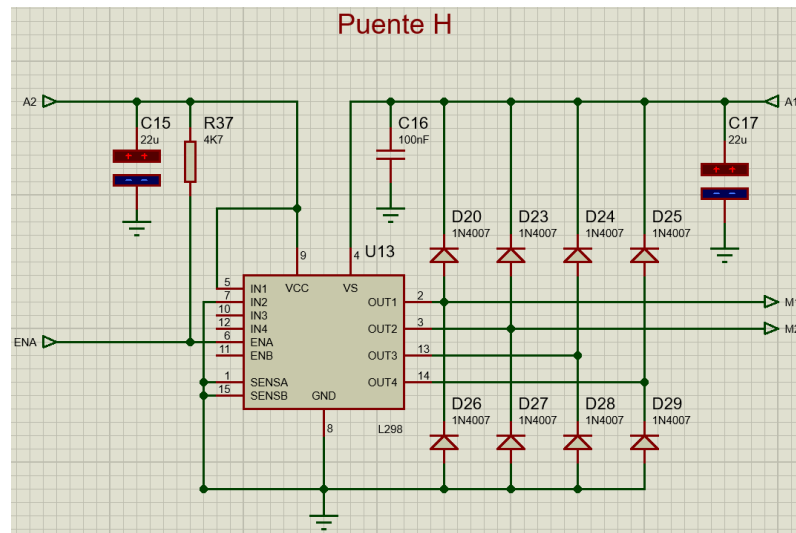
## 2.4.7 CIRCUITO DE CONTROL DE VENTILADOR CENTRIFUGO

Es imprescindible contar con un circuito de control cuando se manipula un motor o en este caso un ventilador centrífugo, entre las opciones se puede considerar obtener un módulo puente H (Figura 25), o diseñar el mismo. Como tal permite proteger al microcontrolador de corrientes que regresan al activar un actuador, evitando daños en los puertos de la Raspberry Pi.



**Figura 25.** Módulo LN298 puente H.  
(ELECTRONILAB, 2016)

Para el proyecto se optó por el diseño, ya que se pretende conformar un solo circuito, entre los elementos se consideró el integrado LN298, además de los diodos para el puente H. A continuación, se observa el esquema realizado en Proteus (Figura 26).



**Figura 26.** Esquema del circuito puente H.

En la Tabla 11 se observa los componentes usados para el circuito.

**Tabla 11.** Componentes del circuito puente H.

Componente	Cantidad	Tipo
Diodos	8	1N4007
Capacitores Electrolíticos	2	22 [uf]
Capacitores Cerámicos	1	100 [nf]
Integrado	1	LM298
Resistencias	1	4.7 [KΩ]

## 2.4.8 CONTROL DE ELECTROVALVULAS SELENOIDES

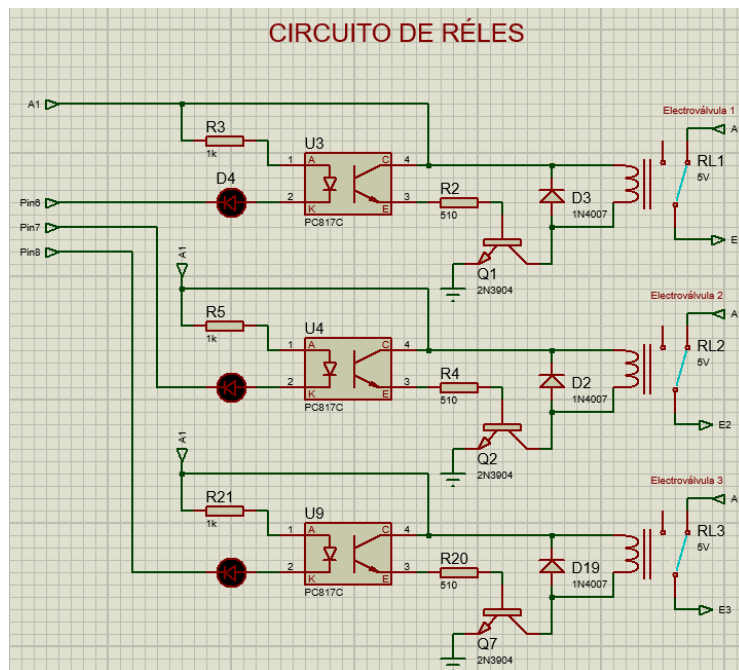
Para controlar la acción de las electroválvulas del mismo modo que el anterior circuito se puede optar por el uso de un módulo réle (Figura 27). De la misma manera se optó por diseñar, al tratarse de relés, estos manejan un porcentaje de corriente considerable, valores que al igual fueron tomados en cuenta en el diseño de la fuente de alimentación.

Como en todo circuito este posee elementos útiles para su activación, siendo el réle, el elemento principal mismo que puede ser activado con 5 [V], y que a través de sus entradas pueden circular voltajes desde los 12 [VCA] a 250 [VCA]. Su función será la de realizar acciones de encendido y apagado de electroválvulas al igual que evitará el regreso de corrientes a las salidas digitales del microcontrolador.



**Figura 27.** Módulo réle 5 [V] a 4 canales.  
(Tolocka, 2015)

A continuación, se muestra en la Figura 28 el esquema del circuito diseñado.



**Figura 28.** Circuito de relés.

En la Tabla 12 se presenta los componentes usados para el diseño del circuito.

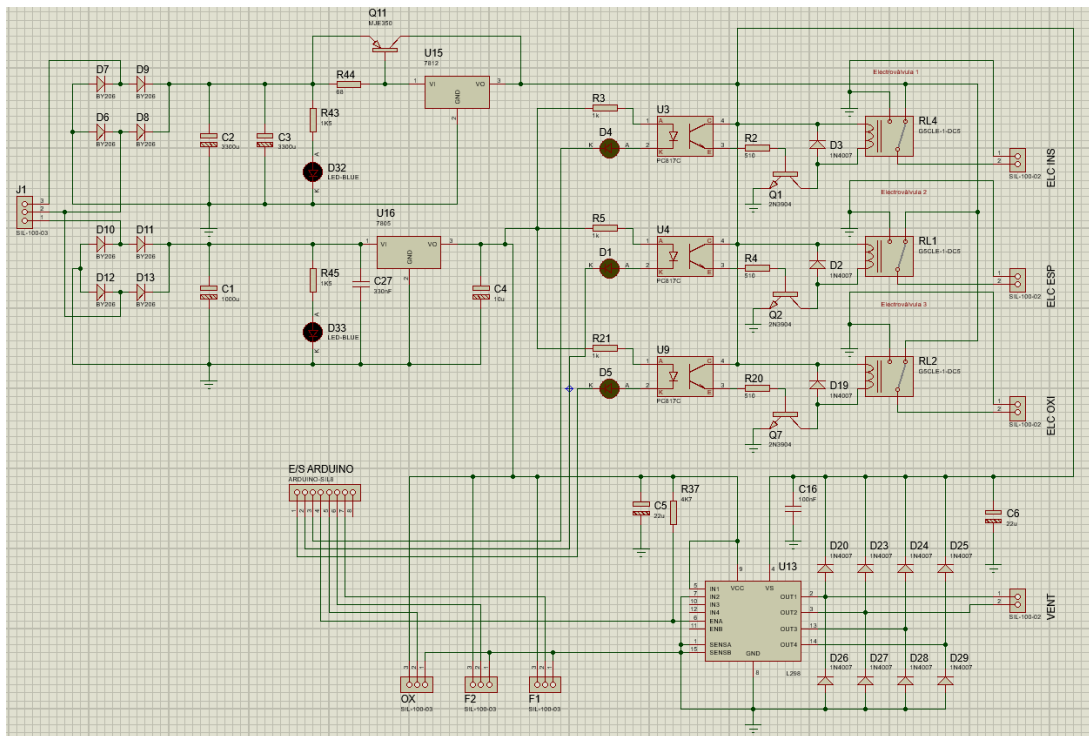
**Tabla 12.** Componentes del circuito de relés.

Componente	Cantidad	Tipo
Diodos	3	1N4007
Resistencias	3	510 [ $\Omega$ ]
Resistencias	3	1 [K $\Omega$ ]
Transistores	1	2N3904A
Relés	3	SRD-5 [V]

## 2.5 INTEGRACION DEL DISEÑO ELECTRONICO

Como se mencionó en apartados anteriores, el diseño electrónico del sistema debe integrarse en un solo circuito, evitando el uso de módulos en la parte de control, además del uso excesivo de cables en la conexión de componentes electrónicos. Del mismo modo nos permitirá eliminar interferencias en señales provenientes de sensores, causadas por corrientes parásitas producidas por los elementos electromecánicos al ser activados.

En la Figura 29 se presenta el esquema del circuito que integra los diseños anteriormente realizados.



**Figura 29.** Circuito electrónico completo.

Una vez que se obtuvo el esquema general del circuito se procedió a generar el PCB (Figura 30), con más detalle se podrá observar en el apartado de anexos del documento.

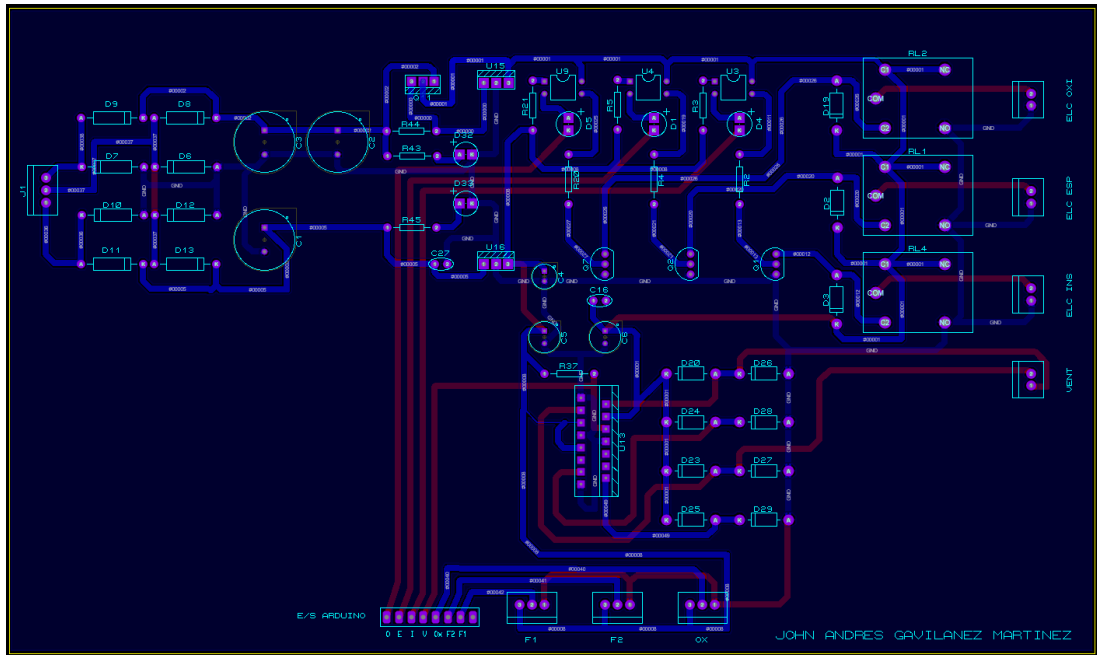


Figura 30. Diseño de PCB del circuito.

En la Figura 31 se observa el circuito diseñado.

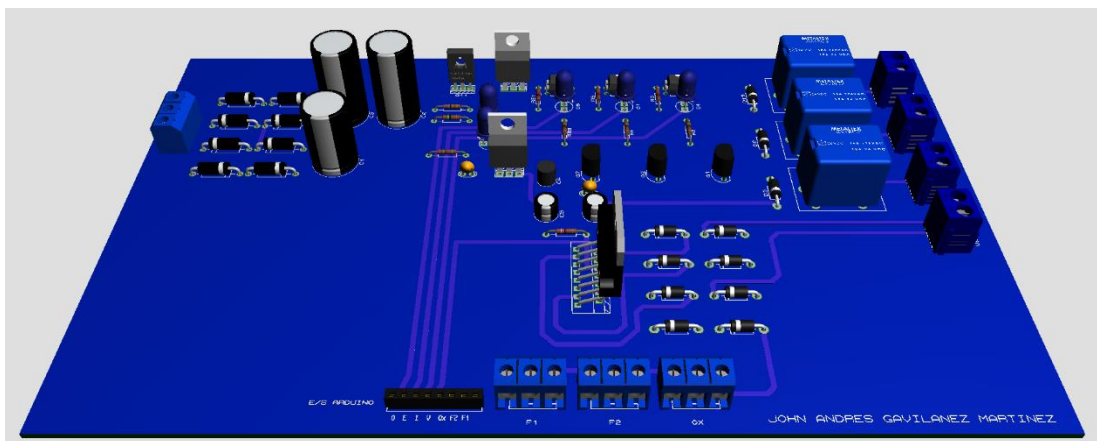


Figura 31. Representación 3D del circuito.

## 2.6 ACTUADORES

### Ventilador Centrifugo

El ventilador centrifugo es un componente por el cual ingresa el aire paralelamente al rotor, mismo que contiene una cantidad de paletas o alabes de diferentes formas y curvaturas, los cuales mediante su rotación produce un flujo que viaja hacia un conducto de manera tangencial.

En si transforma la energía cinética del flujo, en presión estática. Estos ventiladores pueden llegar a producir presiones altas a su salida (Figura 32), como tal este componente será el responsable de generar aire, mismo que se mezclará con el oxígeno y se enviará a los pulmones.



**Figura 32.** Ventilador Centrifugo de alta presión.  
(Delta, 2020)

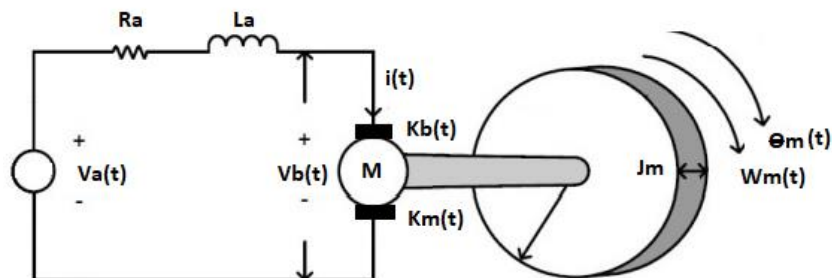
En la Tabla 13 se puede observar las características y especificaciones del actuador.

**Tabla 13.** Especificaciones del ventilador centrifugo.

Parámetro	Característica
Voltaje nominal	12 [V]
Flujo de aire	$1.27 \left[ \frac{\text{m}^3}{\text{min}} \right]$
Presión estática	3694 [H <sub>2</sub> O]
RPM	5800 [rpm]
Corriente nominal	1.97 [A]

### Modelo dinámico del ventilador centrifugo

El ventilador centrifugo es similar a uno de cd convencional, tomando en cuenta que se trata de un motor sin escobillas. En la Figura 33, se observa la representación del sistema electromecánico en cd.



**Figura 33.** Modelo de un motor cd.  
(Becerra & Moreno, 2014)

Se comprobó si el ventilador elegido es el indicado, para ellos se obtuvo el modelo matemático. De modo que se realizó el análisis de la parte mecánica y eléctrica del motor, que corresponde a la armadura y rotor:

$$V_a(t) = V_{Ra} + V_{La} + V_b(t) \quad [16]$$

$$V_b(t) = K_b \cdot W_m(t) \quad [17]$$

$$\tau(t) = \tau_{Ja} + \tau_{fa} \quad [18]$$

$$\tau(t) = K_m \cdot i(t) \quad [19]$$

Con las fórmulas obtenidas del sistema se realizó los respectivos remplazos para obtener la función de transferencia. En la Ecuación 16 se remplazó los

voltajes de la resistencia e inductor, al igual que el equivalente de la fuerza contraelectromotriz (Ecuación 17). Como resultado se obtuvo una ecuación diferencial de primer orden (Ecuación [20]), la cual tiene coeficientes constantes, es ordinaria al tener una sola variable independiente que en este caso es el tiempo y es no homogénea en  $V_a(t)$ .

$$V_a(t) = R_a \cdot i(t) + L_a \cdot \frac{di(t)}{dt} + K_b \cdot W_m(t)$$

$$L_a \cdot \frac{di(t)}{dt} = V_a(t) - R_a \cdot i(t) - K_b \cdot W_m(t) \quad [20]$$

En la Ecuación [18 se realizó el remplazo del par inercial de armadura y el par de fricción, de tal manera que se obtiene una ecuación de primer orden (Ecuación 21), al ser una expresión con coeficientes constante es ordinaria y no homogénea en  $\tau(t)$ .

$$\tau(t) = J_a \cdot \frac{dW_m(t)}{dt} + F_a \cdot W_m(t)$$

$$J_m \cdot \frac{dW_m(t)}{dt} = \tau(t) - F_a \cdot W_m(t) \quad [21]$$

Se obtuvo las transformadas de Laplace de las Ecuaciones [19, [20 y [21:

$$\tau(s) = K_m \cdot i(s) \quad [22]$$

$$L_a \cdot s \cdot i(s) = V_a(s) - R_a \cdot i(s) - K_b \cdot W_m(s) \quad [23]$$

$$J_m \cdot s \cdot W_m(s) = \tau(s) - F_a W_m(s) \quad [24]$$

Se despejo el término de corriente  $i(s)$  de la Ecuación [22 y se remplazó en [23, como resultado se obtuvo la siguiente expresión (Ecuación 25).

$$i(s) = \frac{\tau(s)}{K_m}$$

$$L_a \cdot s \cdot \frac{\tau(s)}{K_m} = V_a(s) - R_a \cdot \frac{\tau(s)}{K_m} - K_b \cdot W_m(s) \quad [25]$$

De la Ecuación [24 se despejo el término de torque  $\tau(s)$  y se remplazó en la expresión antes obtenida (Ecuación [25).

$$\tau(s) = W_m(s) \cdot (J_m \cdot s + F_a) \quad [26]$$

$$V_a(s) = L_a \cdot s \cdot \frac{W_m(s) \cdot (J_m \cdot s + F_a)}{K_m} + R_a \cdot \frac{W_m(s) \cdot (J_m \cdot s + F_a)}{K_m} + K_b \cdot W_m(s)$$

$$V_a(s) = W_m(s) \cdot \left[ \frac{L_a \cdot J_m \cdot s^2 + (L_a \cdot F_a + R_a \cdot J_m) \cdot s + R_a \cdot F_a + K_m \cdot K_b}{K_m} \right]$$

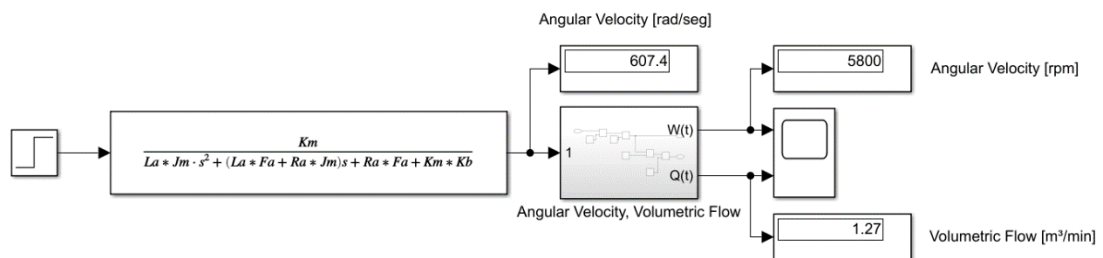
$$\frac{W_m(s)}{V_a(s)} = \frac{K_m}{L_a \cdot J_m \cdot s^2 + (L_a \cdot F_a + R_a \cdot J_m) \cdot s + R_a \cdot F_a + K_m \cdot K_b} \quad [27]$$

Como resultado de las operaciones realizadas se obtuvo la Ecuación 27. De manera que para las simulaciones se realizó estimaciones con las características ventiladores centrífugos comerciales a continuación se puede visualizar en la Tabla 14 los datos que se utilizó para el propósito.

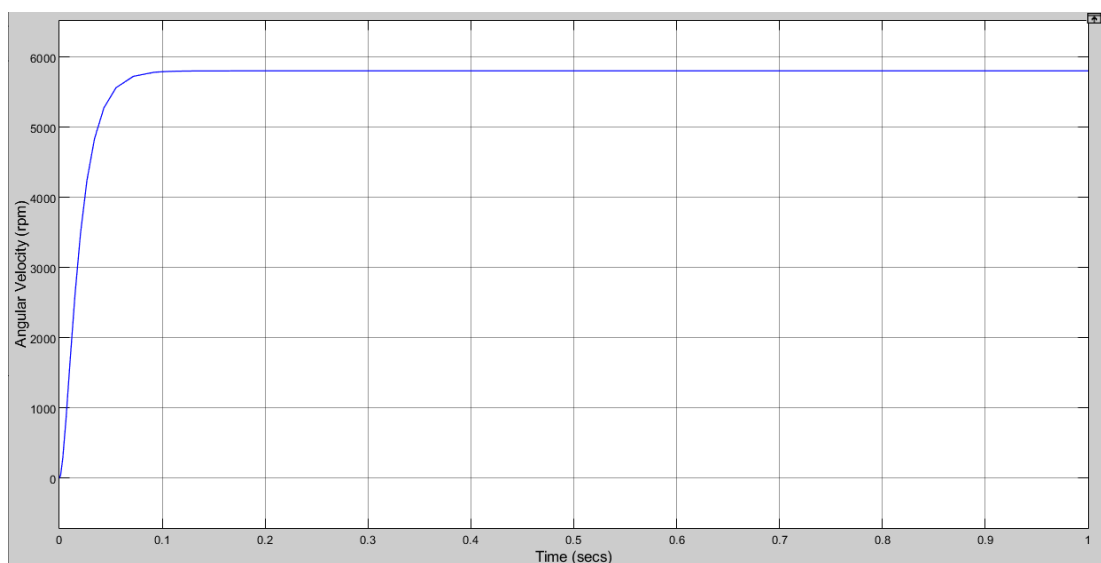
**Tabla 14.** Valores estimados del motor centrifugo.

Parámetro	Valor	Unidades
$V_a$	12	V
$W$	5800	rpm
$R_a$	5	$\Omega$
$L_a$	0.1	H
$J_m$	$6.2652 \times 10^{-8}$	$\text{Kg} \cdot \text{m}^2$
$F_a$	$1.1481 \times 10^{-8}$	$\text{N} \cdot \text{m} \cdot \text{s}$
$K_B$	$3.53982 \times 10^{-3}$	$\text{V}/\text{rad}$
$K_m$	$3.53982 \times 10^{-3}$	$\text{Nm}/\text{A}$

Las simulaciones se las realizo en el programa Matlab en su herramienta Simulink, como resultado se obtuvo la siguiente representación en diagrama de bloques (Figura 34), y la respuesta a una entrada escalón (Figura 35).



**Figura 34.** Función de transferencia del motor centrifugo.



**Figura 35.** Respuesta escalón de la función de transferencia.



## Relación de la velocidad y el flujo

Como se observó anteriormente la función de transferencia del motor, la salida corresponde a la velocidad angular en revoluciones por minuto [rpm]. Para relacionar el parámetro mencionado con el caudal se utilizó la siguiente formula:

$$Q_V = Q_{v0} \cdot \frac{n}{n_0} \quad [28]$$

Donde:

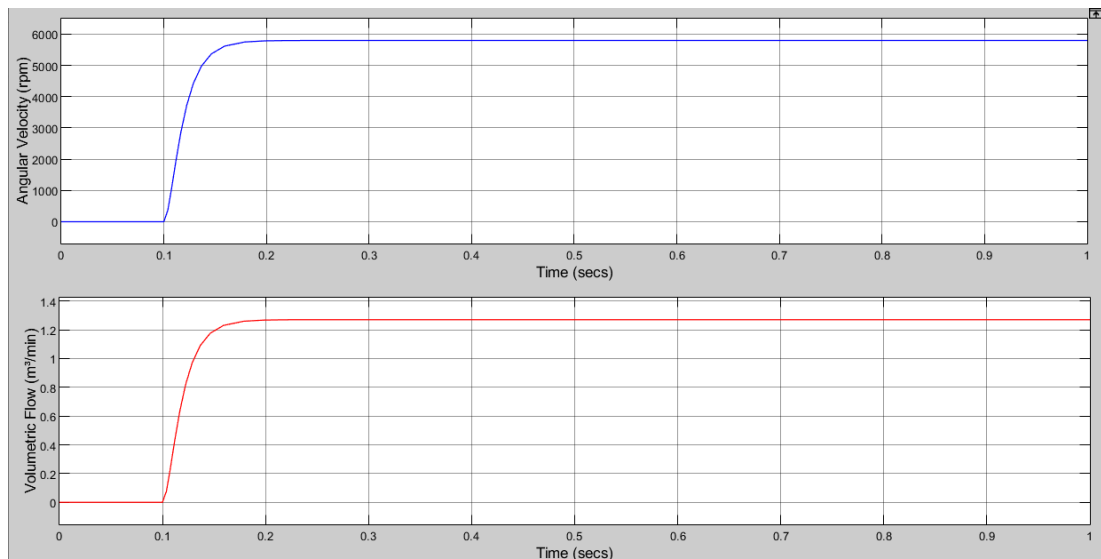
$Q_V$  : Caudal final.

$Q_{v0}$  : Caudal inicial.

$n$  : Velocidad rotacional final.

$n_0$  : Velocidad rotacional inicial.

Como se puede observar en la Figura 36, a una velocidad de rotación se genera un flujo en un tiempo determinado.



**Figura 36.** Grafica que relaciona la velocidad angular con el caudal.

Como se observó en las gráficas, con un voltaje de entrada de 12 [V] a su salida se generó una velocidad angular máxima del ventilador centrífugo de 5800[rpm]. Con la formula aplicada se obtuvo un flujo volumétrico aproximado a 1,27 [m<sup>3</sup>/min].

## Electroválvula

Entre otros de los actuadores considerados para la simulación están las electroválvulas (Figura 37) las cuales permiten, la entrada de oxígeno y el paso de flujo de aire en la inspiración hacia los pulmones y la espiración al exterior del sistema, este tipo de componente al poseer un solenoide demanda una corriente significativa para su activación.



**Figura 37.** Electroválvula solenoide.  
(Velleman, 2018)

La Tabla 15 se presentan las especificaciones de la electroválvula solenoide.

**Tabla 15.** Especificaciones de la electroválvula solenoide.

Parámetro	Característica
Voltaje nominal	12 V
Temperatura	100 °C
Potencia	8 W
Presión	0.02 a 0.8 MPa
Corriente nominal	0.6 A

## 2.7 SENSORES

Dentro de un sistema electrónico es muy necesario el uso de sensores de modo que nos permitan tener información sobre cualquier fenómeno físico. El sensor de flujo considerado fue el modelo YF-SR201 (Figura 38).



**Figura 38.** Sensor de Flujo YF-SR401.  
(Arrieta, 2019)

En la Tabla 16 se detalla las características del sensor.

**Tabla 16.** Especificaciones del sensor de flujo.

Parámetro	Característica
Voltaje de operación	5V a 18V
Rango de Flujo	1 30 lmin
Pulsos por litro	450
Factor de conversión	7.5
Presión	1.75 MPa
Corriente de activación	15 mA

El sensor (Figura 39), proporciona información sobre el porcentaje de oxígeno en el aire, este tipo de sensor posee una alta sensibilidad, así como una fuerte capacidad anti-interferencia.



**Figura 39.** Sensor de Oxígeno (Grove - Gas Sensor).  
(Latorre, 2015)

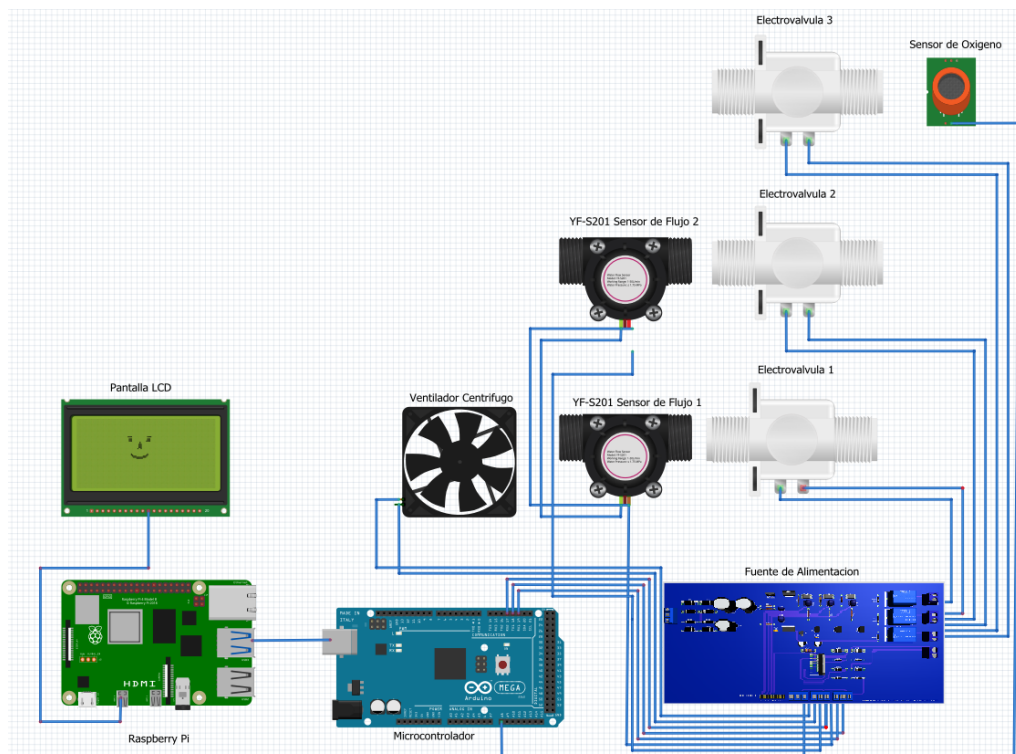
En la Tabla 17 se presenta las especificaciones del sensor de oxígeno.

**Tabla 17.** Especificaciones del sensor de oxígeno.

Parámetro	Característica
Rango de medición	5 a 25 %V
Vida útil	2 años
Sensibilidad	0.05 a 0.015 mA (Aire)
Rango de temperatura	20°C a 50°C
Tiempo de respuesta	≤ 15 segundos
Estabilidad	< 2%

## 2.8 CONEXIÓN DEL VENDILADOR ARTIFICIAL

Una vez que se determinó los elementos principales en el sistema se procedió a realizar el diagrama de conexión (Figura 40). Donde se incluyó a los actuadores, sensores, microcontrolador (Arduino), ordenador de placa reducida (Raspberry Pi), así como también la fuente diseñada, y la pantalla.



**Figura 40.** Esquema de conexión del circuito del sistema.

## **2.9 DISEÑO DEL SISTEMA DE CONTROL**

En la parte de control es imprescindible el uso de una interfaz que permita la interacción del usuario con el dispositivo, previo al desarrollo de la aplicación se realizó un análisis del software y lenguaje de programación; el sistema operativo bajo el cual se ejecutara el programa, así como las herramientas que se utilizarían para simular su funcionamiento.

## **2.10 SELECCIÓN DEL MODO DE VENTILACION Y PARAMETROS**

En el siguiente apartado se muestra el análisis que se realizó para la selección del modo de ventilación idóneo para el tratamiento de enfermedades provocadas por COVID-19. El SDRA un síndrome que a partir de la mencionada neumonía provoca que se filtre fluido desde los vasos sanguíneos, lo cual agrava la condición respiratoria del paciente, es aquí donde se necesita de hospitalización urgente, en este estado lo más probable es que se tome la decisión de entubar y asistir mediante un ventilador artificial. Sepsis es otra complicación causada por el virus, ocurre cuando una infección alcanza el flujo sanguíneo, provocando el colapso de órganos como los pulmones, corazón entre otros (Internacional, 2021).

En lo correspondiente a sistemas ventilatorios existe varios modos utilizados en la actualidad para el tratamiento de pacientes con enfermedades pulmonares graves, se han considerado dos los cuales son muy comunes en equipos de ventilación artificial. En dispositivos controlados por volumen (VCV) todas las respiraciones son controladas, no depende del estímulo inicial del enfermo, por lo cual es especialmente para aquellos que no tienen esfuerzo espiratorio espontáneo o están paralizados. En aparatos programados por presión (VCP) consiste en el establecimiento de un valor determinado de esta variable, así como el tiempo inspiratorio, Fracción respiratoria y relación I/E, el flujo entregado varía de acuerdo con la demanda del paciente, puede reclutar alveolos colapsados y congestivos (Castillo, 2014).

El modo de control por volumen es el más utilizado para el combatir el SDRA, cabe recalcar que no es el único que se utiliza; en este tema se difiere, en cuanto a la preferencia de ventilación. Se puede concluir que no existe un método eficaz, pero por la facilidad de programación, ya que cuenta con menos variables a configurar, es amigable con el personal a cargo de estos equipos. Esta aplicación se diseñó para el tratamiento de personas mayores de edad ya que ocho de cada diez muertes reportadas están entre los 35 a 65 años, dato que va cambiando a medida que se presentan nuevas variantes es por eso se consideró que el dispositivo trate a enfermos de 15 a 65 años (choices, 2021).

## 2.11 DIAGRAMA DE FLUJO DEL SISTEMA

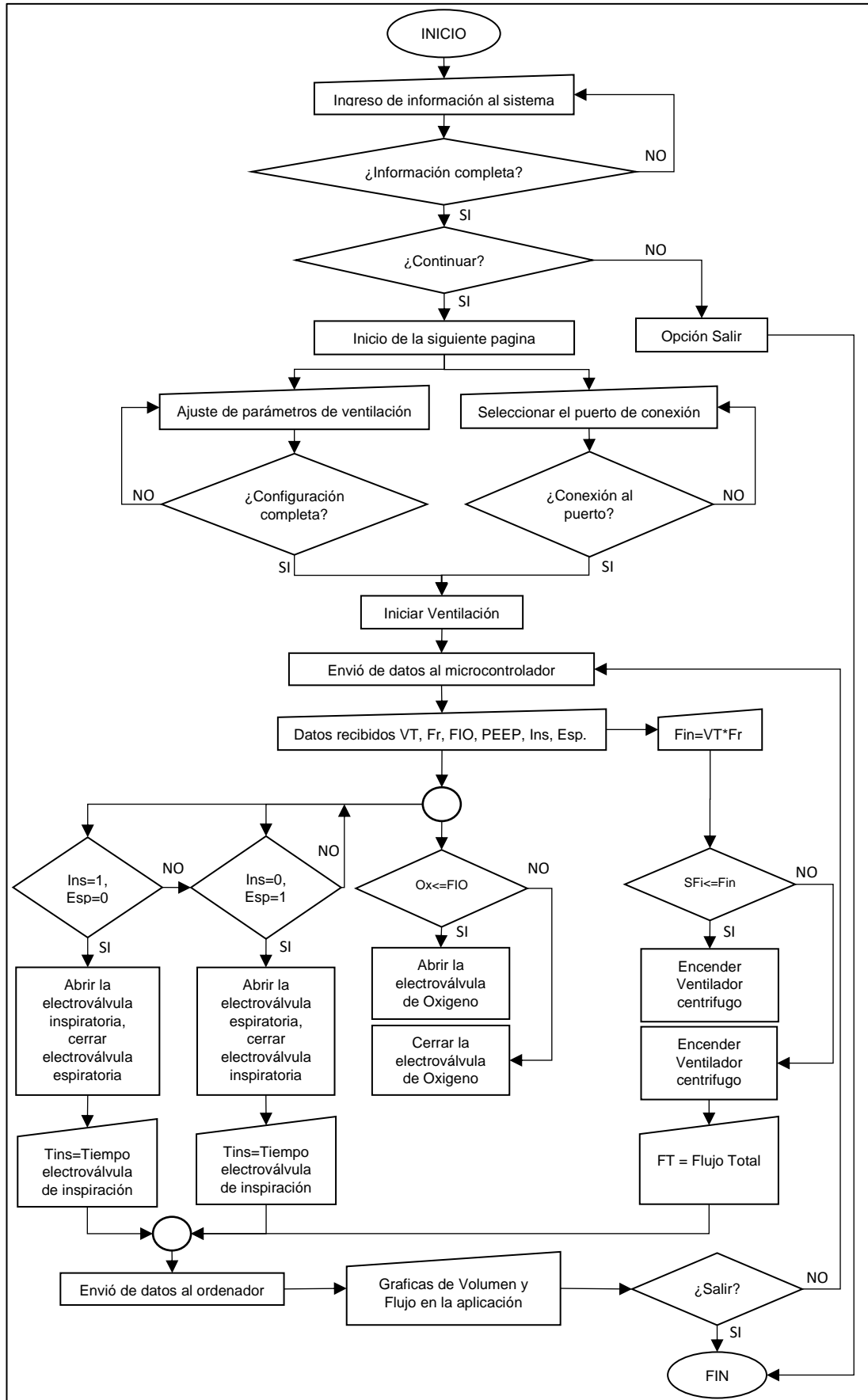


Figura 41. Flujograma del sistema.

Como se observó en la Figura 41 el diagrama de flujo brinda una información previa de cómo se estructuró el programa, para posteriormente iniciar con la codificación. Para ello el paso a seguir fue el diseño y la implementación del algoritmo de funcionamiento de la aplicación.

## 2.12 FUNCIONALIDAD DEL SISTEMA

En el siguiente diagrama de casos de uso representado en la Figura 42 se puede observar el funcionamiento del sistema, el modo en el cual el usuario interactuara con el mismo, al igual que las funciones que brinda.

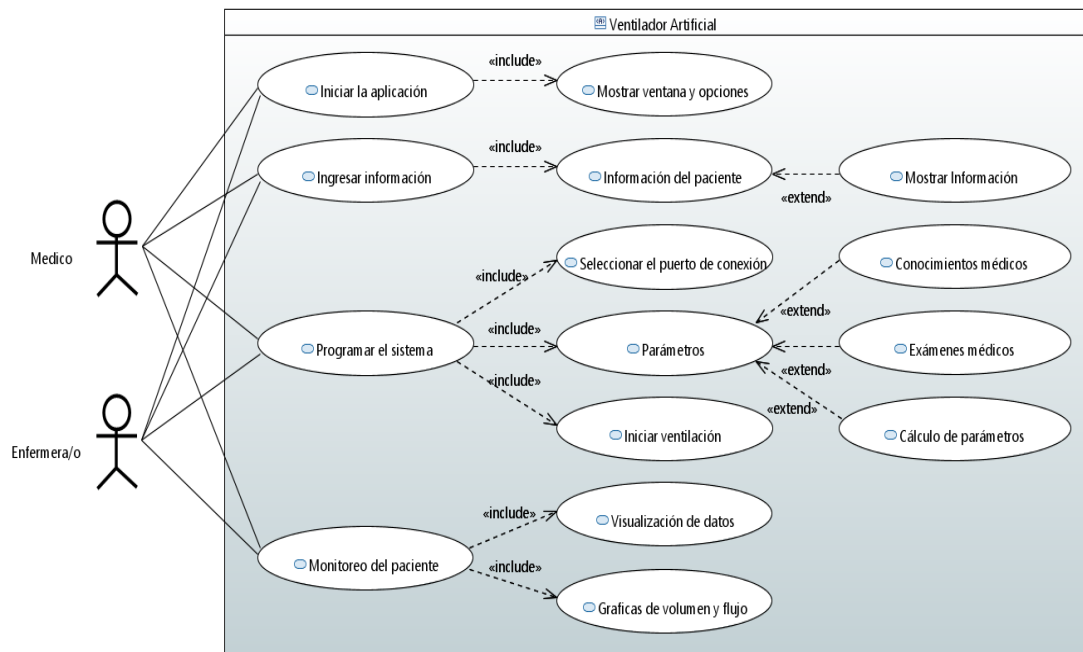


Figura 42. Caso de uso del ventilador artificial.

## 2.13 INTERFAZ HOMBRE-MAQUINA

Para el desarrollo de la interfaz gráfica se hizo uso de un software de programación, para ello entre dos programas conocidos como lo son NetBeans y IntelliJ IDE. Se optó por el segundo ya que es una herramienta de código abierto en su versión comunitaria, y al ser multiplataforma ofreció la posibilidad de programar tanto en Windows y Linux independientemente de cualquier tipo de arquitectura.

IntelliJ IDEA es un entorno de desarrollo integrado (IDE), es sensible al contexto para trabajar con Java y otros lenguajes JVM como Kotlin, Scala y Groovy en todo tipo de aplicaciones. Una característica es que verifica el estado y la validez de su código con inspecciones sobre la marcha. Esto ayuda a programar más rápido, cumpliendo con altos estándares de calidad y mantener la confianza (Jetbrains, 2021). Para la estructuración de la interfaz gráfica se utilizó este software en su versión 2021.2.1.

### 2.13.1 ELEMENTOS DEL SISTEMA

El sistema diseñado posee una interfaz gráfica dividida en dos ventanas por un lado se realiza el registro del paciente y por otro la respectiva configuración, para su posterior funcionamiento. En la página del formulario consta de cajas de texto donde se coloca la información del enfermo como se puede visualizar en la Figura 43.

Datos Monitor

Nombres

Apellidos

Estatura cm

Fecha de Nacimiento 1 Enero 1900

Genero  Hombre  Mujer

Continuar Salir

Figura 43. Ventana de registro de información.

En la ventana de configuración (Figura 44), se visualiza los controles con base en normas de programación de equipos médicos y valores recomendados por la OMS fueron considerados. Además, consta de un panel donde se ubicaron las gráficas de volumen y flujo.

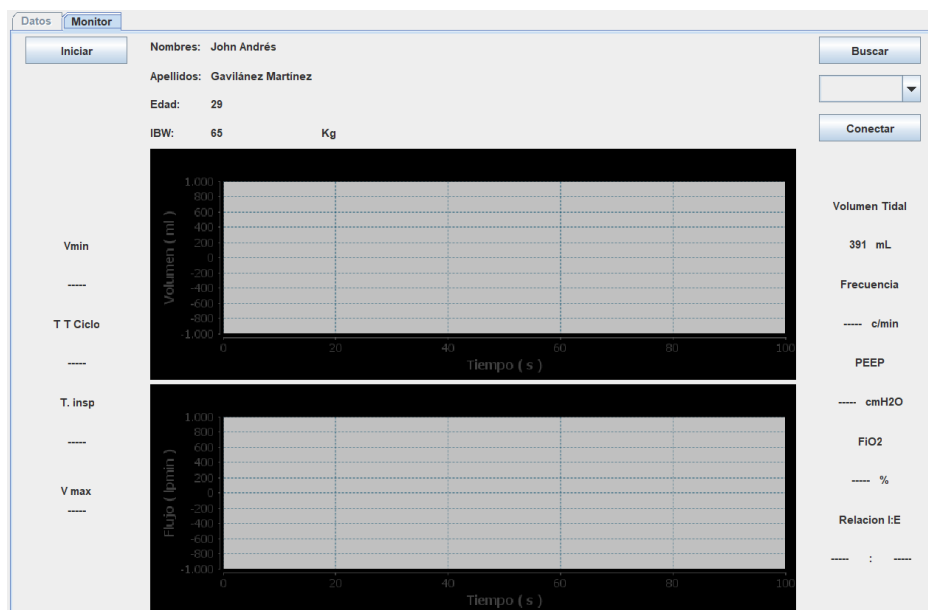


Figura 44. Ventana de configuración

### 2.13.2 CONTROLES

En la Figura 45 se observa los controles principales de la aplicación, los cuales corresponden al volumen corriente (VT), que podrá ser ajustado conforme al estado del paciente; ya que esta variable puede ser modificada siempre que se detecte una posible obstrucción en las vías aéreas o alguna otra complicación. Además de ello se encuentra el control de frecuencia respiratoria el mismo que debe ser modificado basándose en tiempos que permitan realizar un adecuado intercambio gaseoso, el PEEP que corresponde a la presión al final de la espiración. El  $FiO_2$  determina el porcentaje de oxígeno que ingresara a los pulmones, y la relación I:E que determinará el tiempo de entrada y salida de aire.

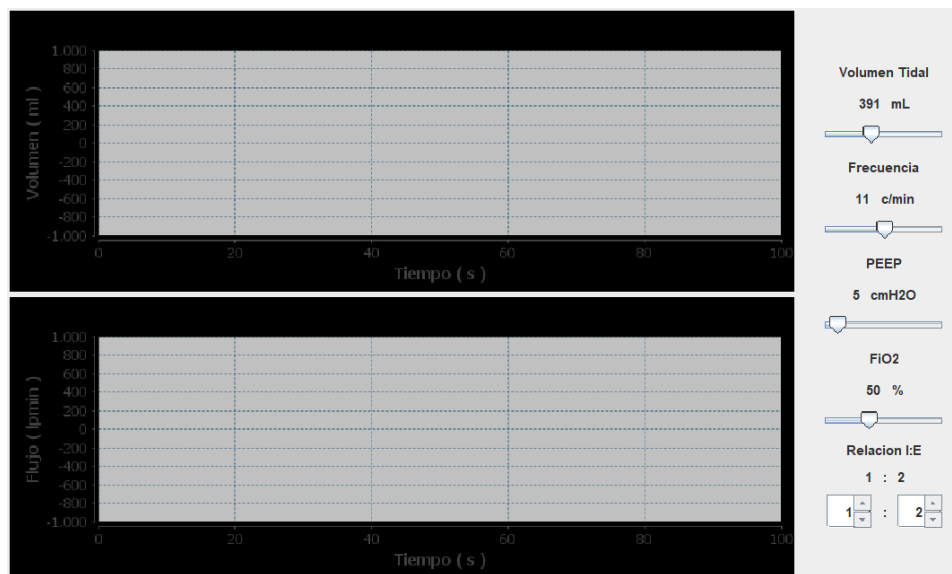


Figura 45. Controles de la aplicación.

### 2.13.3 LECTURA DEL PUERTO Y CONEXIÓN DEL ARDUINO CON EL ORDENADOR

En la parte superior derecha se ubicaron los controles que servirán para el funcionamiento del sistema (Figura 46). Se selecciona el botón buscar para obtener los dispositivos disponibles, posteriormente se realiza la selección del puerto para empezar la comunicación e iniciar el proceso.



Figura 46. Control para la comunicación entre Arduino y ordenador.



### 2.13.4 DATOS ADICIONALES

En la Figura 47 se visualiza otros parámetros que serán generados al iniciar la aplicación, los cuales corresponden al volumen minuto ( $V_{min}$ ) y máximo ( $V_{max}$ ), además del tiempo total de ciclo (TTCiclo), e inspiratorio ( $T_{insp}$ ).

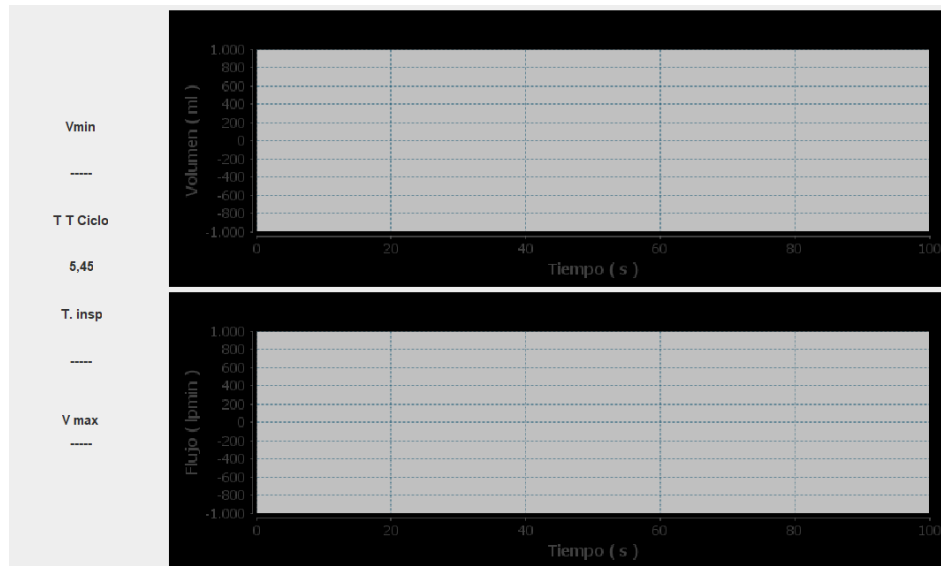


Figura 47. Panel de datos Calculados.

### 2.13.5 PANEL DE GRAFICAS

En la Figura 48 se visualiza las dos graficas que corresponden al volumen, flujo/tiempo. Donde se podrá observar en el panel el comportamiento de estas variables, así como el patrón programado para su funcionamiento.

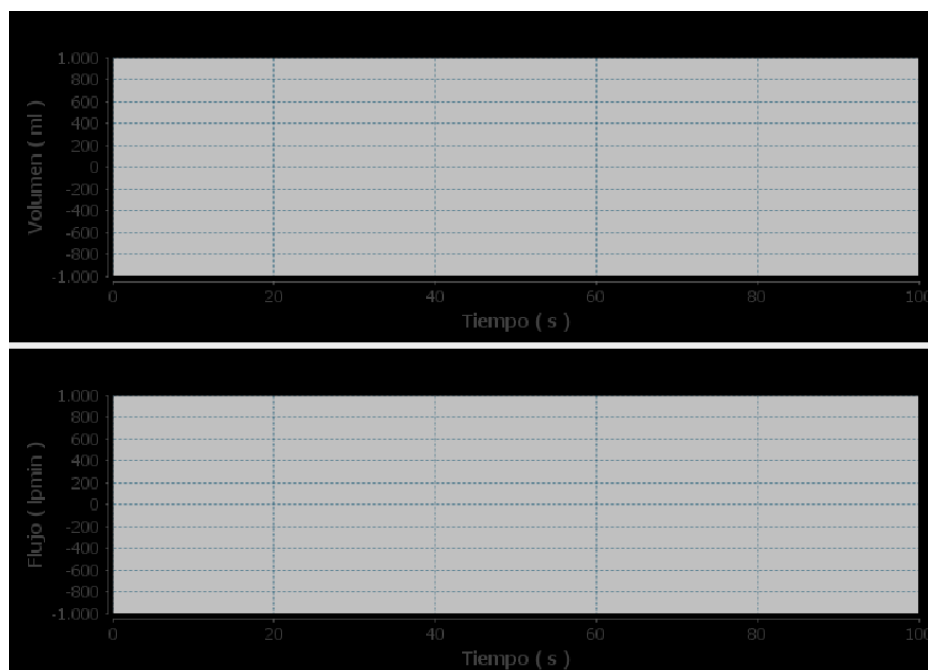


Figura 48. Panel de graficas volumen y flujo con respecto al tiempo.

## 2.14 CODIGO DE LA INTERFAZ INTELLIJ IDEA

Para la programación de la comunicación y gráficas se usó la librería JSSC (java serial simple connector), y JfreeChart respectivamente, las cuales son estables con IntelliJ IDEA en plataforma Linux, y son compatibles con Raspberry. Se menciona esto debido a que existen librerías usadas para este propósito las mismas que originan problemas a la hora de la conexión, velocidad de envío y recepción (Finocchiaro, 2021).

Para establecer de manera automática la comunicación se estableció parámetros como la velocidad, arquitectura, paridad (Figura 49). En cuanto a la búsqueda de puertos se guardó la lista de los datos leídos en un vector, parte del código se puede visualizar en la Figura 50.

```
sp.openPort();
sp.setParams( baudRate: 9600, dataBits: 8, stopBits: 1, parity: 0);
sp.addEventListener(new LecturaSerial(sp), SerialPort.MASK_RXCHAR);
```

**Figura 49.** Comunicación serial, envío y recepción de datos.

```
try {
    String puertos[] = SerialPortList.getPortNames();
    Boolean l = Boolean.parseBoolean(puertos[0]);
    cbxPuertos.addItem("-Seleccione-");

    if (l == false) {
        for (String n : puertos) {
            cbxPuertos.addItem(n);
        }
    }
} catch (Exception e1) {
    JOptionPane.showMessageDialog(parentComponent: null, message: "No se detectaron puertos");
    cbxPuertos.removeAllItems();
}
}
```

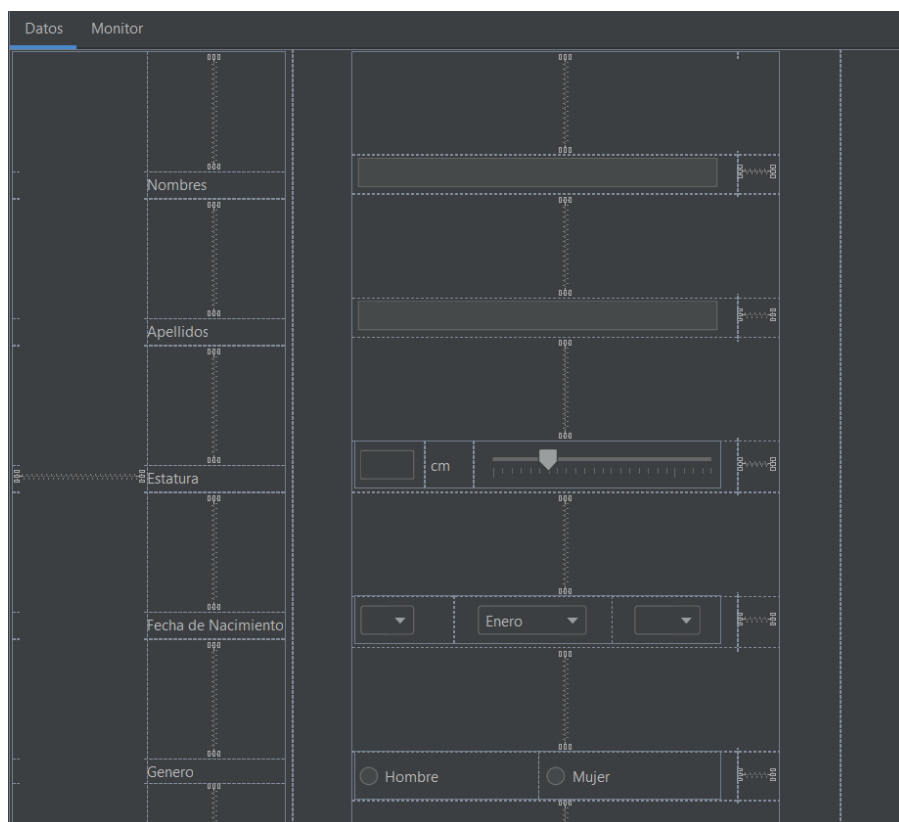
**Figura 50.** Programación búsqueda de puertos.

Se hizo uso de múltiples hilos empleando la clase Thread, los mismo que de manera sincrónica se ejecutarán al interior de la aplicación, en la Figura 51, se visualiza parte de la codificación. Todo el arreglo se incluyó en el método serialEvent, el cual se ejecuta constantemente. Lo que se logró con esta estructura de programación es que se desarrollen varias tareas a la vez como el envío y recepción de datos, al igual que el cálculo dentro del programa. De este modo se evitará tener procesos secuenciales que desarrollan un solo proceso a la vez, o dependen de la finalización de una tarea para iniciar otra, esto limita significativamente sus funciones.

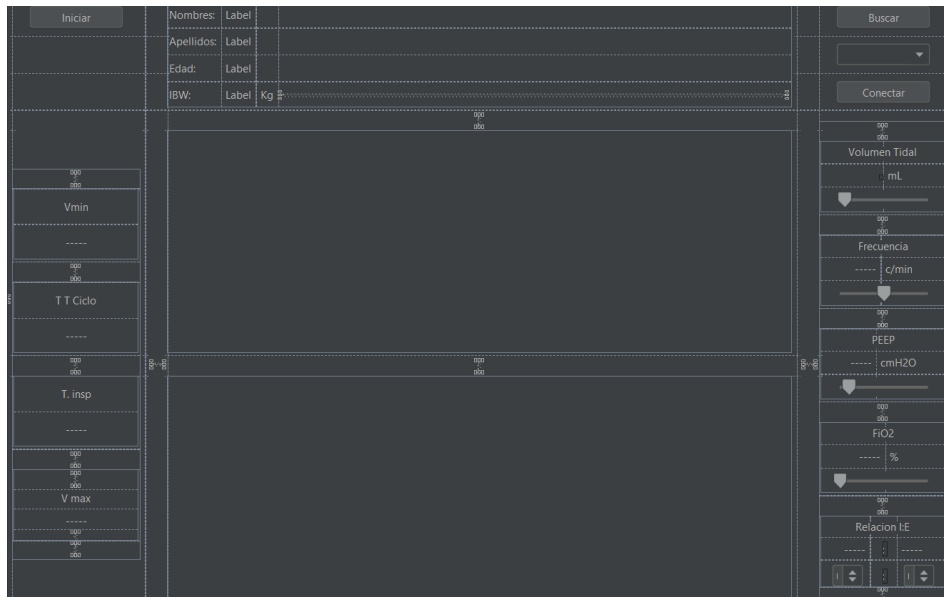
```
public void serialEvent(SerialPortEvent serialPortEvent) {  
  
    if (serialPortEvent.isRXCHAR() {  
  
        if (sp.isOpened()){  
  
            Grafica h = new Grafica();  
            Tins tins=new Tins();  
            envio1 En1=new envio1();  
            envio2 En2=new envio2();  
            envio3 En3=new envio3();  
  
            maxmin mm=new maxmin();  
  
            //inicio de hilos  
            h.start();  
            tins.start();  
            En1.start();  
            En2.start();  
            En3.start();  
            mm.start();  
            //inicio de hilos  
  
            lblVmin.setText(String.valueOf(Integer.parseInt(lblVtd.getText()+Integer.parseInt(lblFr.getText()))));  
  
        }  
    }  
}
```

**Figura 51.** Uso de hilos para el envío y funcionamiento del sistema.

Una vez que se estructuró la comunicación y los procesos se inició con el diseño de la ventana; se colocaron, botones, cajas de texto, paneles para la ubicación de cada elemento. Fue diseñada en dos partes, la primera para colocar los datos del paciente, y segunda para la configuración. En la Figura 52 y Figura 53 se puede visualizar el diseño de la aplicación.



**Figura 52.** Diseño de la Ventana registro.



**Figura 53.** Diseño de la ventana de configuración.

La programación realizada de botones implementación de métodos, eventos, clases anidadas, se podrá visualizar en el apartado de anexos con más detalle.

### Estructura del algoritmo de control

La aplicación se estructuro de manera sincrónica mediante el uso de hilos (Theads), y están ubicados en diferentes clases; al dar la orden de iniciar el programa ejecutará las tareas programadas simultáneamente. Los datos se actualizan constantemente. En la Figura 54 se puede visualizar los cálculos de tiempo de espiración e inspiración condicionada por la relación I/E.

```

public void run(){
    if (Integer.parseInt(lblIns.getText()) == 1 && Integer.parseInt(lblExp.getText())>=1){
        lblTinsp.setText(String.format("%.2f", (60/Double.parseDouble(lblFr.getText()))/(Double.parseDouble(lblIns.getText())
        +Double.parseDouble(lblExp.getText())));
        Tins= (long) (((60/Double.parseDouble(lblFr.getText()))/(Double.parseDouble(lblIns.getText())+Double.parseDouble(lblExp.getText())))*1000);
        Texp= (long) (((60/Double.parseDouble(lblFr.getText()))-((60/Double.parseDouble(lblFr.getText()))/(Double.parseDouble(lblIns.getText())
        +Double.parseDouble(lblExp.getText())))*1000);
    }else if (Integer.parseInt(lblExp.getText()) == 1 && Integer.parseInt(lblIns.getText())>=1){
        lblTinsp.setText(String.format("%.2f", (60/Double.parseDouble(lblFr.getText()))-
        ((60/Double.parseDouble(lblFr.getText()))/(Double.parseDouble(lblIns.getText())+Double.parseDouble(lblExp.getText()))));
        Tins= (long) (((60/Double.parseDouble(lblFr.getText()))-((60/Double.parseDouble(lblFr.getText()))/(Double.parseDouble(lblIns.getText())
        +Double.parseDouble(lblExp.getText())))*1000);
        Texp= (long) (((60/Double.parseDouble(lblFr.getText()))/(Double.parseDouble(lblIns.getText())+Double.parseDouble(lblExp.getText())))*1000);
    }
}

```

**Figura 54.** Cálculos necesarios para el algoritmo de control.

Para obtener las gráficas y designar el patrón de flujo se utilizó la siguiente estructura, los datos se obtuvieron del microcontrolador y se guardaron en un arreglo de bytes llamado buffer1, estos valores fueron transformados a enteros para posteriormente enviarlos a un método que los interpolará. La

cadena de bytes varía entre 0 a 254 en java, mientras que en Arduino va de 0 a 255, esto se concluyó al realizar la ejecución de la aplicación y someterlo a pruebas, en la (Figura 55) se observa cómo se estructuró el programa.

```

if (cg<=100){
    byte[] buffer1 = sp.readBytes(); //cadena en byte
    valIn = buffer1 [0] & 0xff; //valores enteros
    mapeo=mapVtd(valIn, inmin: 0, inmax: 255, outmin: 0, outmax: 1);
    Ti=Tins;
    operacion1=mapeo*sldVTd.getValue();
    operacion2=operacion1/(Ti/10);
    operacion3=(operacion2*100);

    if (operacion1>0){
        //volumen ins
        grafVolIn=operacion1*(0.067+(0.1*(contInVol/1.5))); //3.21
        //volumen ins

        //flujo ins
        grafFluIn=operacion3;
        //flujo ins

        contInVol++;
        contEsVol=14;
        contEsFlu=1;

        series.add(cg,grafVolIn);
        series1.add(cg,grafFluIn);
    }else{
        //volumen exp
        grafVolEs=sldVTd.getValue()*(0.067+(0.1*(contEsVol/1.5)));
        //volumen exp

        //flujo exp
        grafFluEs=sldVTd.getValue()*-Math.exp(-contEsFlu);
        //flujo exp

        contInVol=0;
        contEsVol--;
        contEsFlu++;

        series.add(cg,grafVolEs);
        series1.add(cg,grafFluEs);
    }

    cg++;
} else{
    cg=0;
    series.clear();
    series1.clear();
}

```

**Figura 55.** Algoritmo de funcionamiento, lectura de datos, y graficas.

Para él envío de datos al microcontrolador, se realizó un arreglo de bytes, para evitar retardos y perdidas de información en la Figura 56 se visualiza como se codifico.

```

class envio1 extends Thread{
    public void run(){
        try {
            a=mapVtd(sldVTd.getValue(),sldVTd.getMinimum(),sldVTd.getMaximum(), outmin: 0, outmax: 255);
            int [] buffer2=new int[]{1,a,sldFr.getValue()};
            sp.writeIntArray(buffer2);
        } catch (SerialPortException e) {
            //e.printStackTrace();
        }
    }

    int mapVtd(int x,int inmin,int inmax,int outmin,int outmax){
        return ((x-inmin)*(outmax-outmin))/((inmax-inmin)+outmin);
    }
}

```

**Figura 56.** Envío de datos al microcontrolador.

## 2.15 CODIGO EN ARDUINO

Una vez que se diseñó la aplicación, se procedió a la programación en el microcontrolador, dónde este se encargará de obtener las instrucciones programadas, para la activación y desactivación de los actuadores. En la Figura 57 se puede observar cómo se estructuró el programa para recibir y enviar los datos.

```

if(Serial.available()>3){

    int lectura = Serial.readBytes(bufferIn,3);
    proceso(lectura);

}else{

    Vins=HIGH;
    Vesp=HIGH;
    digitalWrite(Vc,LOW);
    digitalWrite(VOx,LOW);

    digitalWrite(Vol,LOW);
    digitalWrite(LED7,LOW);

    DatoIn=HIGH;

```

**Figura 57.** Recepción de datos en Arduino.

En la Figura 58 se detalla la codificación para activación y desactivación de los elementos electromagnéticos, cuando la aplicación es ejecutada y configurada, realiza él envío de datos al Arduino y este cumple con los procesos requeridos. Dentro del programa se usan los valores de tiempo inspiratorio y espiratorio, de esa manera los actuadores cumplirán las funciones deseadas.

```

void frecuenciaRes(){

    if(Ins==1&&Exp>=1){

        Tins=((60/Fr)/(Exp+Ins))*1000;
        Texp=((60/Fr)-((60/Fr)/(Exp+Ins)))*1000;
        interval1=Texp; //close
        interval2=Tins; //open
        currentTime=millis();

        if(Vins){
            interval=interval2;
        }else{
            interval=interval1;
        }
        if((currentTime-previousTime)>interval){
            previousTime=currentTime;
            Vins=!Vins;
            Vesp=!Vesp;

            DatoIn=!DatoIn;

            digitalWrite(4,Vins);
            digitalWrite(5,Vesp);

            digitalWrite(10,DatoIn);
        }
    }

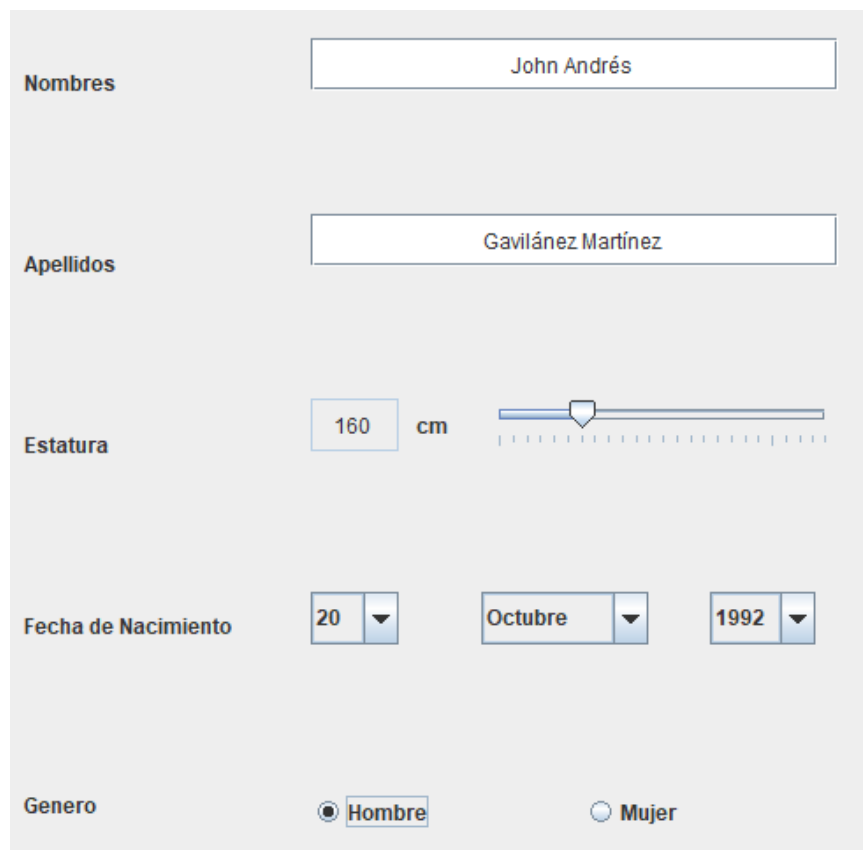
```

**Figura 58.** Control de electroválvulas.

## 2.16 FUNCIONAMIENTO DE LA INTERFAZ

La aplicación desarrollada en java deberá tener soporte para la distribución de Linux de Raspberry Pi, además debe contener la versión jdk 11, ya que una inferior provocaría conflictos en el programa.

Como se puede observar en la Figura 59 se colocó los nombres, apellidos, la altura en [cm] este es un dato que se considerará más adelante, también se agregó la fecha de nacimiento el cual ayudará al sistema a identificar en qué etapa de vida se encuentra el paciente. Una vez que se incluyó la información, permitirá acceder a otra página donde se realizará la programación.



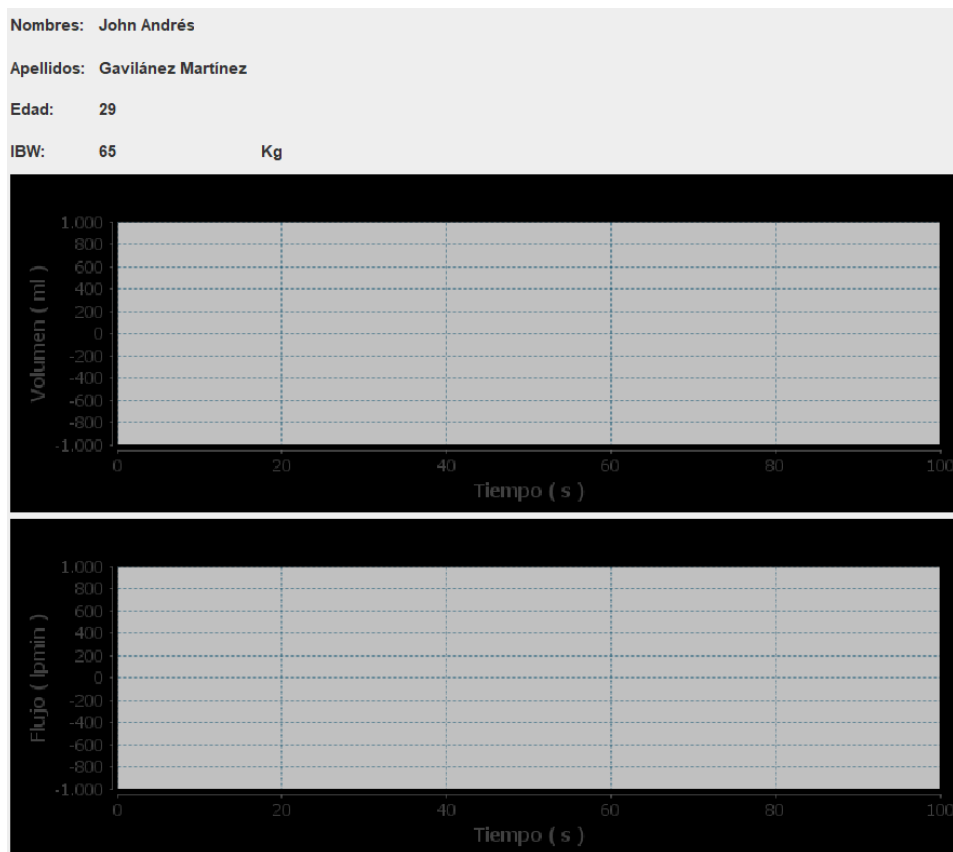
Formulario de registro del paciente con los siguientes campos:

- Nombres:** John Andrés
- Apellidos:** Gavilánez Martínez
- Estatura:** 160 cm (con control deslizante)
- Fecha de Nacimiento:** 20 de Octubre de 1992
- Genero:**  Hombre  Mujer

**Figura 59.** Ficha de registro del paciente

En la ventana que se generó se procedió a configurar el ventilador, el cual consta de controles para variar valores mismos que deben ser ajustados basándose en requerimientos y recomendaciones médicas dependiendo de la patología presentada como consecuencia de la enfermedad.

En la parte superior se puede observar (Figura 60) los datos anteriormente ingresados, a excepción de la altura, este dato fue usado para el cálculo del peso predicho (IBW).



**Figura 60.** Datos iniciales previo a la configuración.

El peso real y predicho son datos diferentes, el primero es usado en el caso de tener otro tipo de patología, diferente a las producidas por la COVID-19, mientras que el segundo se utiliza como parte del cálculo del volumen tidal.

### 2.16.1 CALCULO DEL PESO IDEAL PREDICHO

Las fórmulas que se observan en la Tabla 18 fueron utilizadas para el cálculo del peso ideal en base a talla y género. Esta información es importante previo a la programación del ventilador.

**Tabla 18.** Cálculo del peso ideal.

Parámetro	Característica
Hombre:	$50 + 0,9 [\text{Altura (cm)} - 152.4]$
Mujer:	$45,5 + 0.9 [\text{Altura (cm)} - 152.4]$

En base al peso predicho dependiendo del género del paciente se calcula el volumen corriente (VT). El cual es un valor usado para la protección, ya que corresponde a la cantidad de aire que ingresara a los pulmones.

### 2.16.2 PROGRAMACIÓN DEL SISTEMA

El volumen tidal se calcula automáticamente según la estatura del enfermo y se podrá ajustar sobre la base de otro requerimiento o alguna consideración clínica. La frecuencia respiratoria se programará en función de la edad y la



enfermedad del paciente, principalmente en quienes padezcan de SDRA se colocan valores de 10 c/min que es lo ideal.

Pará el ajuste del oxígeno ( $F_{iO_2}$ ) es necesario iniciar con una cantidad alta, se puede empezar con un 60% puesto a que dependiendo de la gasometría realizada se verificará el grado de saturación de  $O_2$  en la sangre ( $SO_2$ ). Si es menor al 92 % se deberá ajustar hasta lograr un valor meta que se encuentra entre los 92% a 96%, no es recomendable tampoco que el paciente posea el 100% del  $O_2$ .

En lo referente a la frecuencia respiratoria se usan valores entre 12 a 16 [c/min] y de 12 hasta 20 sobre el nivel del mar. En la Tabla 19 se resume los parámetros adicionales que se utilizan en ventiladores convencionales para su programación.

**Tabla 19.** Programación del dispositivo controlado por volumen.

Parámetros	Valor
FIO <sub>2</sub>	21 – 60%
VT	6 – 8 [ml/kg] de peso idea (sin SDRA) 4 – 8 [ml/kg] de peso idea (con SDRA, iniciar con 6)
Disparo	1 a 3 [L/ min] o 0.5 a 2 [cmH <sub>2</sub> O]
PEEP	5 a 8 [cmH <sub>2</sub> O]
Fr	La necesaria por un CO <sub>2</sub> normal o deseado acorde a la patología
PI	0.3 [s] si se requiere monitorizar presión meseta o la necesaria para llevar relación I: E de 1:2 a 1:3
Flujo	30 a 60 [L/min] ASMA o EPOC agudos hasta 100 [L/min] Ajustar para la relación I: E de 1:2 a 1:3

### 2.16.3 GRAFICAS DEL SISTEMA

Los objetivos de la monitorización de las gráficas son optimizar la interacción entre el paciente y el ventilador, con la finalidad de evitar asincronía, lo que provocaría:

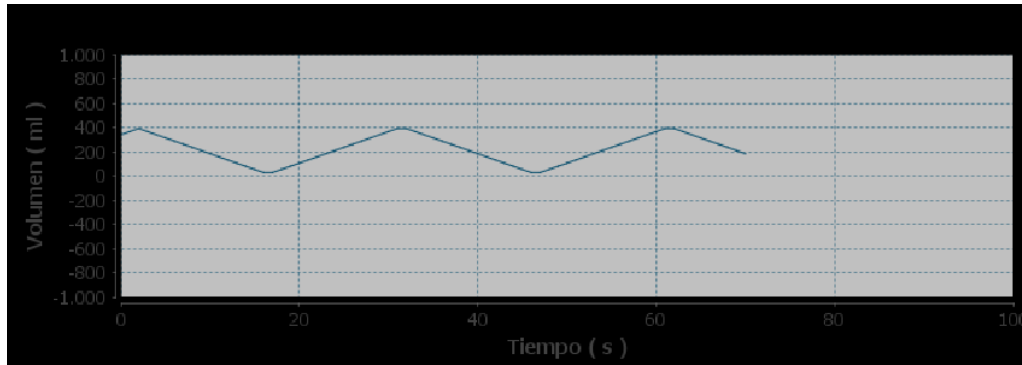
- Aumento del disconforme del enfermo.
- Necesidad de mayor sedación.
- Aumento del tiempo de ventilación.
- Posiblemente aumento de mortalidad

#### Relación paciente

En la actualidad los respiradores permiten contemplar en tiempo real la morfología de 3 curvas las cuales corresponden a presión flujo y volumen. En lo que corresponde a la aplicación diseñada. Al ejecutar el sistema visualiza los datos al lado izquierdo del programa y a la derecha los controles, así como también las gráficas que se generan al programar. Se podrá observar dos tipos de graficas.

## Diagrama volumen tiempo

En la Figura 61 se describen los cambios graduales en la entrada del volumen durante la inspiración y la expiración, este se representa en [ml] y el tiempo en [s] en la fase de flujo inspiratorio se nota un aumento. La monitorización de interacción paciente ventilador, es parte del cuidado integral del sujeto ventilado, como tal la observación del enfermo y de las curvas generadas por el sistema permite valorar la sincronía P-V.

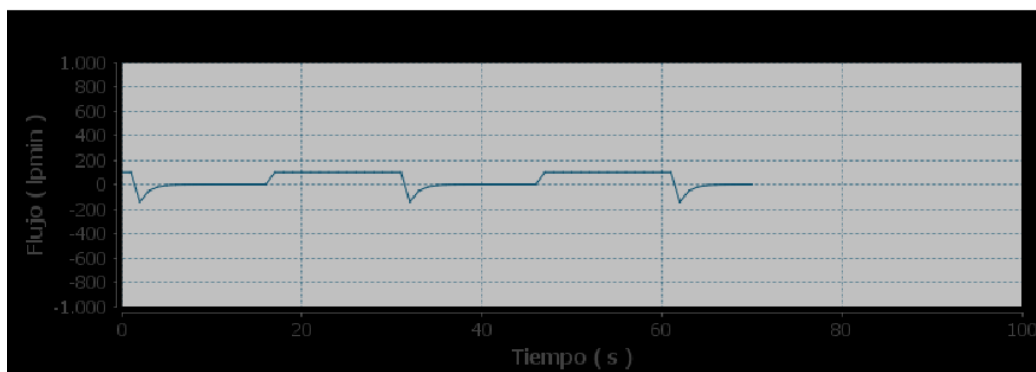


**Figura 61.** Diagrama Volumen vs Tiempo.

La capacidad residual funcional (FRC) considera, ya que durante la expiración el volumen transferido disminuye como resultado de una exhalación pasiva.

## Diagrama de flujo tiempo

En la Figura 62 se observa el diagrama de flujo, en donde muestra los cambios graduales producidos en el proceso inspiratorio y espiratorio. Esta variable se representa en [lpm] y el tiempo en [s] durante el periodo de la inspiración la curva se ve muy influenciada por el tipo de ventilación seleccionado, sólo en la fase espiratoria se obtiene información como la resistencia general y compliancia (distensibilidad) de los pulmones y del sistema. Existen cuatro tipos de onda: cuadrada, desacelerada, acelerada y sinusoidal.



**Figura 62.** Grafica de Flujo vs Tiempo.

### **3. RESULTADO Y DISCUSIÓN**

### 3.1 CARACTERISTICAS DEL COMPUTADOR

En cuanto al diseño y simulación se utilizaron los siguientes paquetes: la parte electrónica en Proteus 8.12; el control se realizó en Matlab 2020, en lo que corresponde al desarrollo de la aplicación se usó IntelliJ IDE 2021, IDE de Arduino.

La simulación y pruebas de funcionamiento se realizó en Raspbian, en una máquina virtual (VirtualBox). Para enlazar las aplicaciones se utilizó un simulador de puertos (Virtual Serial Ports Emulator). La computadora utilizada fue una laptop con: Windows 10, procesador Intel Core i7 de 2.2 [GHz], 16 G en RAM, tarjeta de video Nvidia Geforce.

### 3.2 RESULTADOS DEL DISEÑO ELECTRONICO

Como resultado de la parte electrónica se obtuvo el siguiente esquema del diseño electrónico, donde se detalla cada parte del circuito (Figura 63).

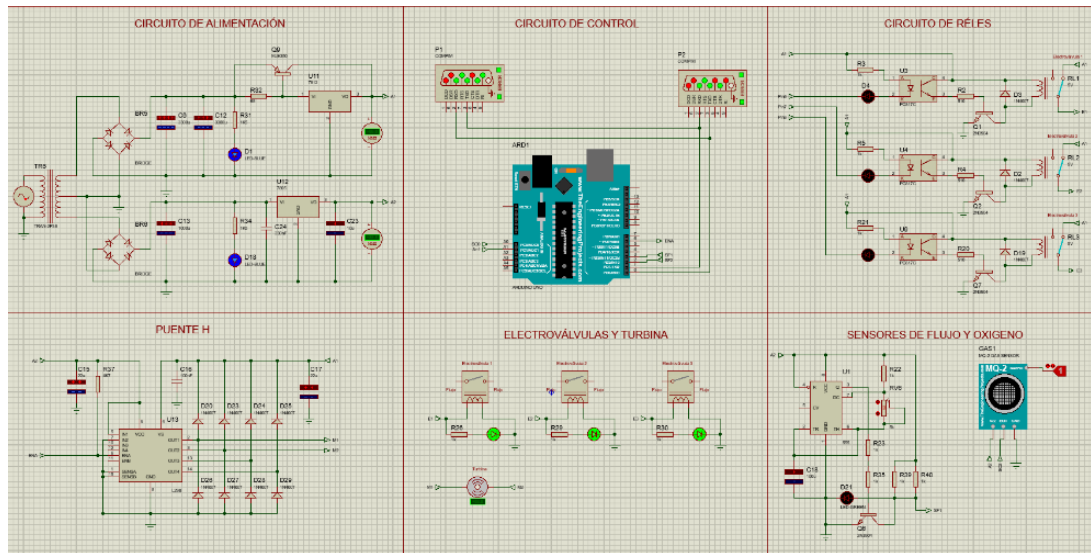
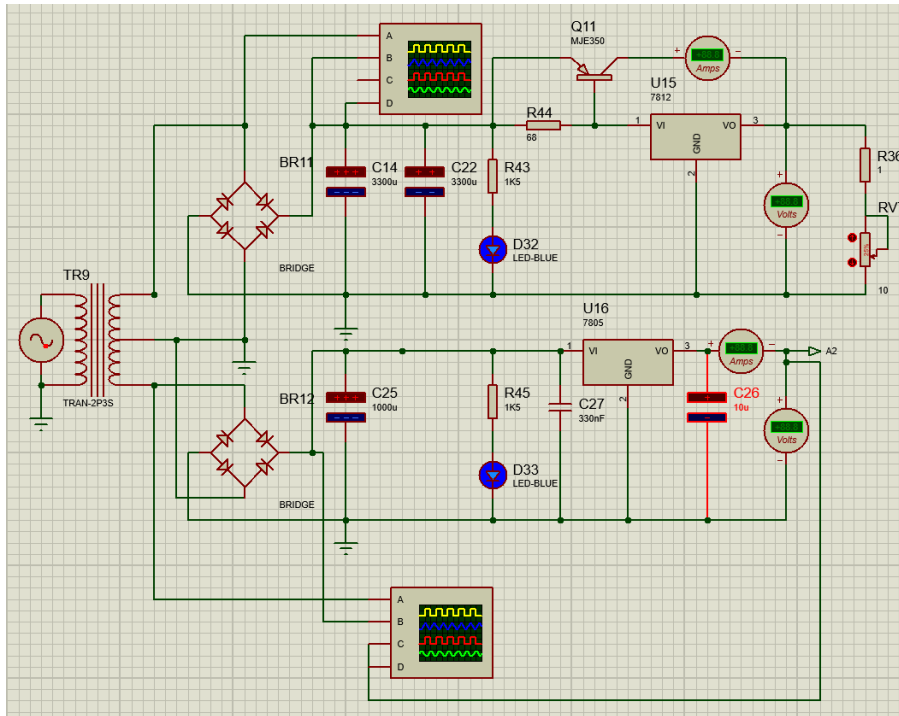


Figura 63. Diseño electrónico del ventilador artificial.

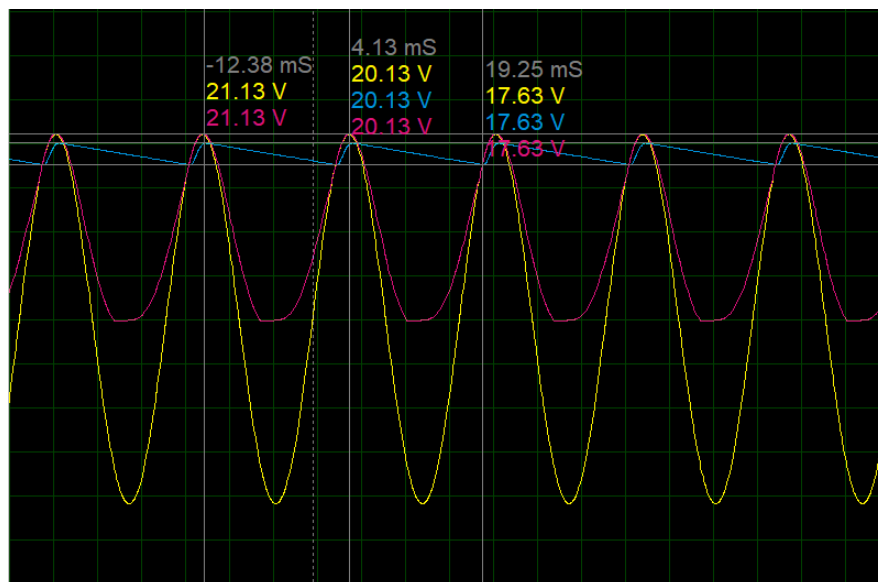
### 3.3 RESULTADOS DE LA SIMULACION DE LA FUENTE

Para verificar el funcionamiento del equipo se inicia el análisis desde la fuente de poder para ello se realiza pruebas con una resistencia variable alimentada a 12 [V], de manera que se compruebe que el transistor seleccionado provee la corriente que se necesita para los actuadores. En lo que corresponde a la parte regulada de 5 [V], se verifica el voltaje de salida agregando una carga, que en este caso serán los sensores. A continuación, se presenta el esquema del circuito de alimentación (Figura 64).



**Figura 64.** Esquema de simulación del circuito.

En la Figura 65, se observa las señales generadas por la fuente, estas son graficadas por el monitor del osciloscopio, cada una se la puede diferenciar con un color diferente. La señal proveniente del secundario es amarilla, la onda recortada es roja, y el resultado del filtro conocido como rizado es azul. El  $V_{pt}$ , representa la tensión máxima que se entrega al condensador. El promedio del voltaje corresponde al valor de entrada al regulador  $V_{i_{reg1}}$ .



**Figura 65.** Señales generadas por la fuente a 12 [V].

Por tanto, al comparar los valores obtenidos en la simulación y los calculados anteriormente tenemos los siguientes resultados:

Valores calculados:

$$V_{pt} = 21.213 \text{ [V]}$$

$$V_{C_{max}} = 19.513 \text{ [V]}$$

$$V_{i_{reg1}} = 15 \text{ [V]}$$

Valores simulados:

$$V_{pt} = 21.13 \text{ [V]}$$

$$V_{C_{max}} = 20.13 \text{ [V]}$$

$$V_{i_{reg1}} = 17.63 \text{ [V]}$$

En los cálculos se puede visualizar que los valores no tienen mucha variación, con excepción del voltaje de entrada al regulador  $V_{i_{reg1}}$  el cual se analizará más adelante. Como se indicó con anterioridad; a la fuente de 12 [V] se le aplicará una carga variable de manera que se podrá observar el comportamiento en distintas situaciones.

### Con carga aumentada

La carga está representado por una resistencia de 1 [ $\Omega$ ] constante y otra de 10 [ $\Omega$ ] variable, las mismas que fueron conectadas en serie. En la Figura 66 se visualiza que la corriente disminuye llegando a un valor de 1 [A], además se observa que la señal correspondiente a la carga y descarga del condensador conocida como voltaje de rizo, no varía significativamente.

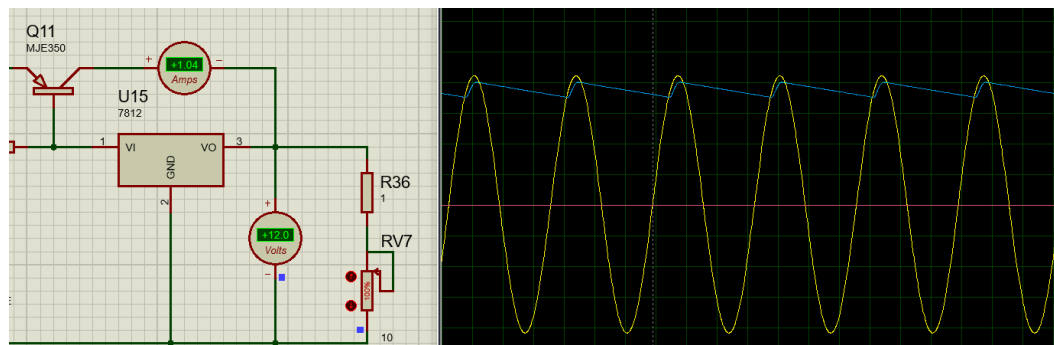
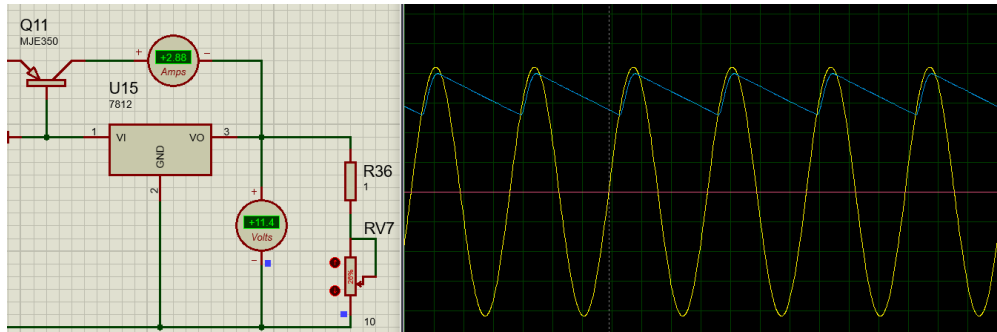


Figura 66. Señal generada con carga elevada.

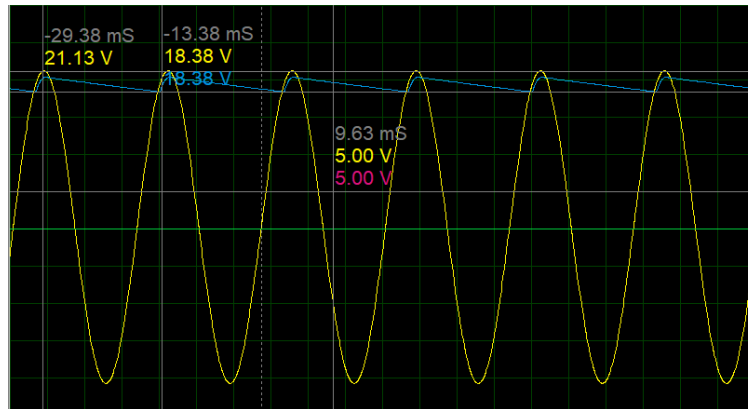
### Con carga disminuida

Una vez que se visualizó el comportamiento de la fuente con carga aumentada, se procedió al análisis con un valor menor (Figura 67), evidenciando un aumento de la corriente. Estos valores superan los 3 [A], al igual que se observa que el voltaje de rizo aumenta conforme se disminuye la resistencia.



**Figura 67.** Señal generada con carga disminuida.

En esta parte se procede con el análisis de la fuente de 5 [V], donde se observa las señales generadas al alimentar a los sensores. En el monitor del osciloscopio (Figura 68) se puede observar el comportamiento del valor pico del secundario, el voltaje de rizado, y señal de salida que proporciona el regulador el cual está representado de color azul.



**Figura 68.** Señales generadas por la fuente a 5 [V].

Valores calculados

$$V_{pt} = 21.213 \text{ [V]}$$

$$V_{C_{max}} = 19.513 \text{ [V]}$$

$$V_{o_{reg2}} = 5 \text{ [V]}$$

Valores simulados

$$V_{pt} = 21.13 \text{ [V]}$$

$$V_{C_{max}} = 18.38 \text{ [V]}$$

$$V_{i_{reg1}} = 5 \text{ [V]}$$

En lo que corresponde a los datos calculados existe una diferencia de 1 [V] aproximadamente. En cuanto a los voltajes y la corriente no existen variaciones, en la Figura 69 se observa el comportamiento de la señal de color rojo correspondiente a la parte regulada del circuito, al dotar energía a los sensores no presentan fluctuaciones de ningún tipo.

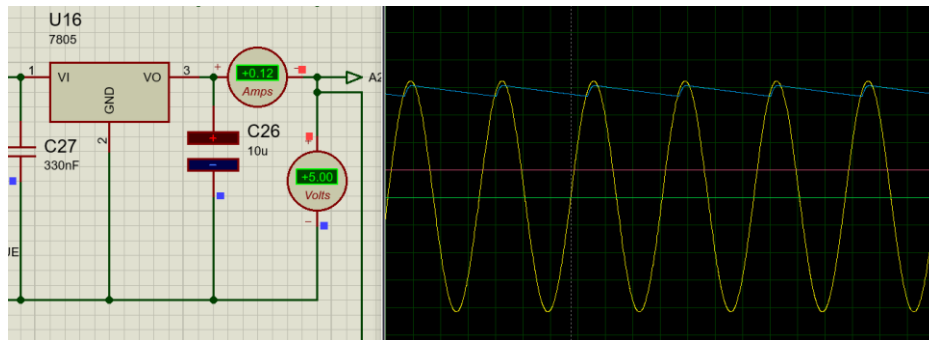


Figura 69. Señales obtenidas del circuito a 5 [V].

### 3.4 RESULTADOS DEL ALGORITMO DE CONTROL

#### Gráfica de volumen y flujo con incremento de la relación I/E

En la Figura 70, se puede observar que existe una ventilación normal por parte del paciente, ya que no existe ningún tipo de variación y se mantiene constante; los valores usados fueron:

$$V_T = 341 \text{ [ml]}$$

$$F_r = 10 \text{ [c/min]}$$

$$I: E = 1: 1$$

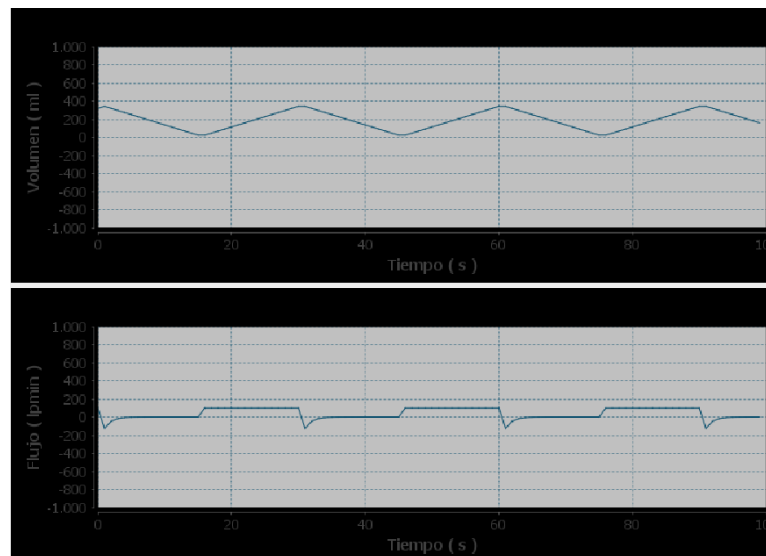


Figura 70. Simulación con relación 1:1.

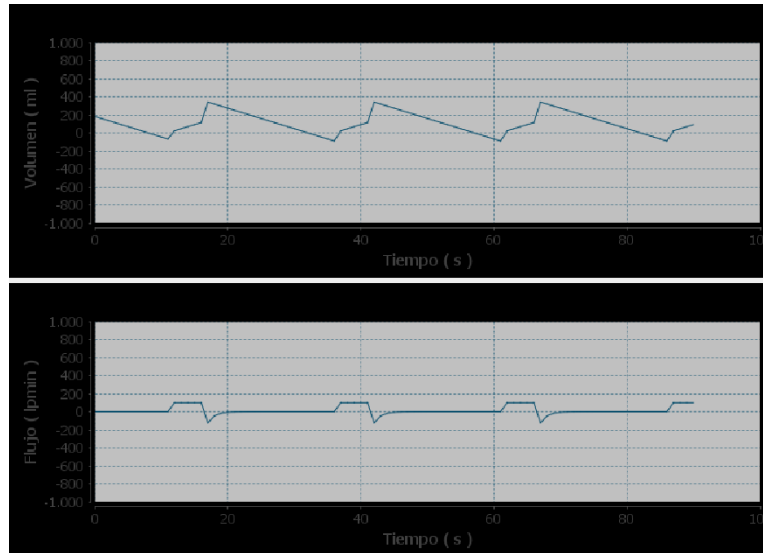
Con una relación de inspiración diferente se observa en la Figura 71 que existe una variación en la gráfica, constatando que hay un aumento en el tiempo espiratorio. Con un valor 1:2 se puede realizar un intercambio gaseoso adecuado.

$$V_T = 341 \text{ [ml]}$$

$$F_r = 11 \text{ [c/min]}$$

$$I: E = 1: 2$$





**Figura 71.** Simulación con relación 1:2.

La variación que se puede observar en la gráfica de volumen y de flujo corresponde a los tiempos, en el caso de la espiración existe un aumento. Mientras que en la inspiración se produce una disminución.

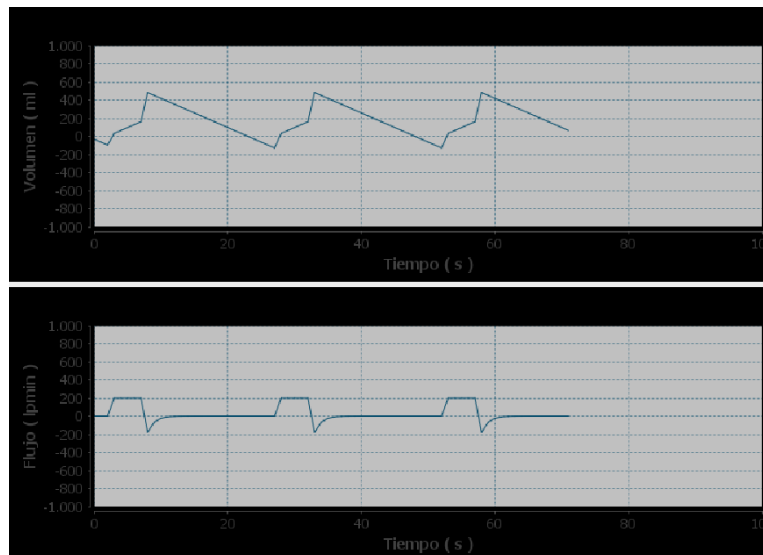
### **Gráfica de volumen y flujo con aumento del volumen tidal**

Al realizar un aumento del volumen tidal, se obtiene un volumen minuto de 5313 [ml], además de coloco los siguientes valores:

$$VT = 483 \text{ [ml]}$$

$$Fr = 11 \text{ [c/min]}$$

$$I: E = 1: 2$$



**Figura 72.** Simulación con aumento del VT.

Como se observó en la Figura 72, al realizar un aumento en el volumen tidal, se nota un aumento en el valor de la gráfica de volumen y flujo.

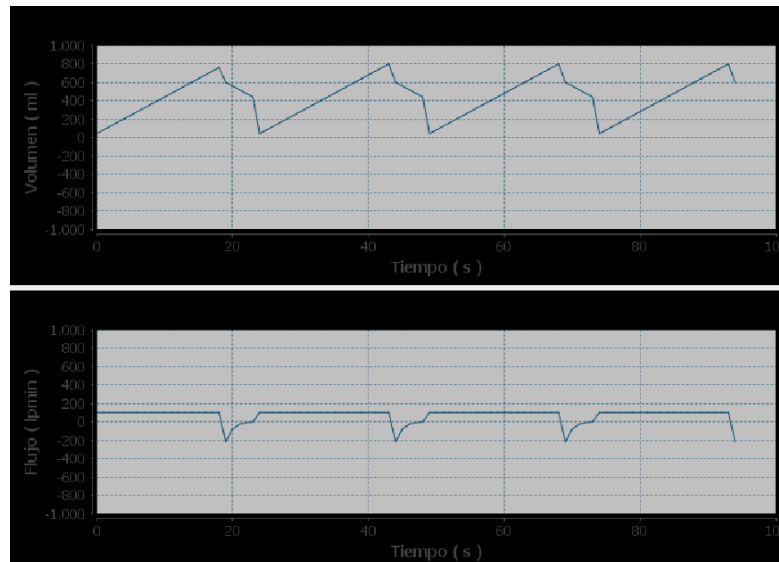
## Gráfica de volumen y flujo con incremento del tiempo inspiratorio

En la Figura 73 se observa un aumento en el tiempo de inspiración, como resultado de incluir los siguientes valores:

$$V_T = 598 \text{ [ml]}$$

$$F_r = 12 \text{ [c/min]}$$

$$I:E = 3:1$$



**Figura 73.** Simulación con aumento del  $V_T$ ,  $F_r$  y tiempo espiratorio.

Como se visualizó el valor del volumen no tiene cambios y se mantiene constante, el cambio de forma se produce por el aumento del tiempo espiratorio. En la gráfica del flujo se puede observar una variación en los tiempos, lo que supondría que no existe problema. Cuando la curva se ve ligeramente alargada, se debe a que el paciente está espirando más aire de lo normal.

En el caso que se produzca una reducción en la curva, se puede concluir que existe una gran oposición por parte de las vías aéreas, y que no permite salir todo el volumen de aire que ingresó al pulmón. Lo cual supondría que se trata de una obstrucción.

## **4. CONCLUSIONES Y RECOMENDACIONES**

## 4.1 CONCLUSIONES

- Los paquetes de software actuales permitieron obtener simulaciones con resultados que se asemejan a la realidad, esto creo mejores oportunidades de desarrollo en el sistema diseñado, al igual que facilito realizar mejoras en el mismo.
- Las recomendaciones que se consideró por parte de médicos especialistas y organizaciones internacionales permitieron cumplir objetivos puntuales para el desarrollo de este tipo de sistemas, a futuro se puede considerar realizar nuevas invenciones o modelos de utilidad que complementen aún más estos dispositivos, por tal motivo es conveniente desarrollar nuevos estudios en ciertos campos de la medicina para obtener mejores resultados, así como incluir ramas de biomedicina en carreras técnicas.
- Con el cumplimiento de los requerimientos planteados al inicio, previo al desarrollo del proyecto, se trabajó de manera más organizada para obtener mejores resultados, al igual que la metodología utilizada nos permitió detectar problemas y algunas falencias, que fueron solucionadas.
- Durante el desarrollo del proyecto en la parte del funcionamiento del sistema se analizó el uso de otros tipos de algoritmos de control, que pueden ser considerados para realizar mejoras a dispositivos médicos, especialmente si se desea crear equipo que sean completamente autónomos.
- Mediante el uso de simulaciones en el caso del diseño electrónico se comprobó los efectos producidos al conectar una carga variable a una fuente regulada, estos resultados pueden ser significativos para comprobar el funcionamiento y verificar si se cumplen con los requerimientos planteados.
- Con este proyecto se constató que los avances de la tecnológica en el campo de la medicina son cruciales, para el desarrollo de equipos que permitan realizar tratamientos eficaces de modo que se puedan salvar más vidas.

## 4.2 RECOMENDACIONES

- Para el desarrollo de este tipo de equipos debería considerarse el trabajo mancomunado entre médicos e ingenieros para implementar nuevos prototipos, al igual que incluir materias relacionadas con medicina en carreras técnicas.
- En lo que corresponde a construcción y diseño de circuitos para dispositivos médicos se debe profundizar en temas de electrónica de potencia, de manera que se pueda obtener equipos estables, además de considerar el manejo de energías renovables.
- Considerar el uso del algoritmo super twisting para controlador tipo deslizante es una opción viable para mejorar el funcionamiento de estos dispositivos, de manera que se puede obtener más funciones de ventilación y conseguir mejores resultados en los tratamientos médicos.
- Como futuros proyectos para la asistencia médica se debería implementar algoritmos relacionados con machine learning, así como en cuestiones de soporte medico el manejo de IOT es otra de las opciones que se podrían desarrollar.
- Se debe considerar el desarrollo de un dispositivo que permita la ventilación múltiple de pacientes, de este modo se podrían salvar más vidas, en la actualidad se ha manejado esta idea, pero se ve limitada por la falta de investigación.
- Es de vital importancia agregar materias en la universidad que se encarguen del estudio de variables fisiológicas del cuerpo humano, al igual que el análisis de modelos matemáticos, o a su vez realizar proyectos de investigación y desarrollo de dispositivos con los compañeros de carrera de medicina.

## **BIBLIOGRAFIA**

- Abarca, B., Vargas, J., & García, J. (30 de Mayo de 2020). Características de la ventilación mecánica invasiva en COVID-19 para médicos no especialistas. *Revista Chilena de Anestesia*. doi:<https://doi.org/10.25237/revchilanestv49n04-06>
- Arrieta, V. (25 de Enero de 2019). *Arca Electronica*. Obtenido de <https://www.arcaelectronica.com/blogs/tutoriales/sensor-de-flujo-de-liquido-yf-s201-arduino>
- BBC News Mundo. (21 de Junio de 2021). *BBC*. Obtenido de <https://www.bbc.com/mundo/noticias-57527964>
- Becerra, J., & Moreno, F. (Diciembre de 2014). *scielo*. Obtenido de <http://www.scielo.org.co/pdf/teclo/v17n33/v17n33a06.pdf>
- Boylestad, R., & Nashelsky, L. (2015). *Electrónica : Teoría de Circuitos y Dispositivos Electrónicos*. Mexico: Pearson Educación,.
- Castillo, G. (2014). Ventilación mecánica no invasiva e invasiva. *MEDICINE*, 3761-3763. doi:10.1016/S0304-5412(14)70840-6
- Cediel, X., & David, R. (2020). *Enfoque del paciente crítico y ventilación mecánica para no expertos*. Colombia: Búhos Editores Ltda.
- Chapman, S. J. (2012). *Maquinas Electricas*. Mexico: Mc Graw Hill.
- choices, c. &. (2021). COVID-19 conociendo Sus Opciones. *compassion & choices* , 2-5. Obtenido de <https://compassionandchoices.org/resource/covid-19-conociendo-sus-opciones/>
- Delta. (25 de Mayo de 2020). *Digi-Key*. Obtenido de <https://www.delta-fan.com/Download/Spec/BFN0724SS-01.pdf>
- DISIPA. (2019). Catalogo de disipadores. Barcelona, España.
- El Comercio. (29 de Febrero de 2020). Ministerio de Salud confirma primer caso de coronavirus en Ecuador. Obtenido de <https://www.elcomercio.com/tendencias/sociedad/salud-confirma-primer-caso-coronavirus.html>
- ELECTRONILAB. (23 de Octubre de 2016). *candy-ho*. Obtenido de <https://candy-ho.com/Drivers/Tutorial%20Uso%20Driver%20L298N%20para%20motores%20DC%20y%20paso%20a%20paso%20con%20Arduino.pdf>
- Ernesto, H., & Galdames, T. (2016). Respiratorio. *Ventilación Mecánica*, 1-11.
- Facciola, N., Martinez, M., Morgade, P., & Briva, A. (2017). Prueba de ventilación espontánea en pacientes. *Revista Medica Uruguay*, 187-194.

- FAIRCHILD. (Septiembre de 2014). *DatasheetCafe*. Obtenido de <http://j5d2v7d7.stackpathcdn.com/wp-content/uploads/2015/09/LM7812.pdf>
- Finocchiaro, T. (18 de Agosto de 2021). *github*. Obtenido de <https://github.com/java-native/jssc/releases>
- Gomez, L., & Salvador, B. (2012). *Fundamentos de ventilacion mecanica*. Barcelona: Marge Medica Books.
- Guerrero, J. (21 de Septiembre de 2014). *PLUSELECTRIC*. Obtenido de <https://pluselectric.wordpress.com/2014/09/21/arduino-uno-especificaciones-y-caracteristicas/>
- Hernandez, H. (Noviembre de 2020). *Medicina intensiva*. doi:10.1016/j.medin.2020.03.015
- Internacional, C. (19 de Febrero de 2021). *Clinica Internacional*. Obtenido de <https://www.clinicainternacional.com.pe/blog/danos-pulmones-covid/>
- Jetbrains. (23 de Agosto de 2021). *Intellij IDEA*. Obtenido de <https://www.jetbrains.com/es-es/idea/features/>
- Latorre, S. (20 de Noviembre de 2015). *Seba Electronic Lab*. Obtenido de <http://sebalabs.blogspot.com/2015/11/arduino-14-arduino-sensor-de-oxigeno.html>
- Lázaro, Á. (13 de Marzo de 2020). *redacción médica*. Obtenido de <https://www.redaccionmedica.com/la-revista/noticias/coronavirus-historia-virus-familiar-conocido-en-el-3-300-a-c--9292>
- life.augmented. (Marzo de 2014). *altronics*. Obtenido de [https://altronics.cl/uploads/L7800CV\\_datasheet.pdf](https://altronics.cl/uploads/L7800CV_datasheet.pdf)
- López, C. (21 de Julio de 2020). *Gaceta Médica*. Obtenido de <https://gacetamedica.com/investigacion/en-que-se-diferencian-los-seis-tipos-de-covid-19/>
- Mecafenix, I. (12 de junio de 2018). *La enciclopedia de la ingeniería*. Obtenido de <https://www.ingmecafenix.com/electronica/fuente-de-alimentacion/>
- Medina, J. (16 de Noviembre de 2015). *Enfermería en Urgencias y UCI*. Obtenido de <http://enfermeroenurgencias.blogspot.com/2015/11/parametros-del-respirador-programar.html>
- multicom. (15 de Diciembre de 2018). *Farnell*. Obtenido de <https://www.farnell.com/datasheets/2712881.pdf>
- multicomp. (12 de Julio de 2008). *Farnell*. Obtenido de <https://www.farnell.com/datasheets/64389.pdf>



- Naveira, P. (16 de Julio de 2020). *as*. Obtenido de [https://as.com/diarioas/2020/07/16/actualidad/1594901498\\_130532.html](https://as.com/diarioas/2020/07/16/actualidad/1594901498_130532.html)
- O.M.S. (27 de Abril de 2020). *Organizacion Mundial de la Salud*. Obtenido de <https://www.who.int/es/news/item/27-04-2020-who-timeline---covid-19>
- OPS. (s.f). *Organizacion Panamericana de la Salud*. Obtenido de <https://www.paho.org/es/temas/coronavirus>
- Ortiz, G., & Dueñas, C. (23 de Abril de 2013). *Bases de ventilación mecánica*. Obtenido de <https://enfermerapediatrica.com/wp-content/uploads/2014/01/BASES-VENTILACION-MECANICA.pdf>
- Panasonic. (2 de Mayo de 2014). Obtenido de <https://docs.rs-online.com/714f/0900766b80f8045d.pdf>
- Panasonic. (7 de Junio de 2015). *RS-online*. Obtenido de <https://docs.rs-online.com/a195/0900766b8154d042.pdf>
- Patel, B. (Abril de 2020). *Manual MSD*. Obtenido de <https://www.msmanuals.com/es-ec/hogar/trastornos-del-pulm%C3%B3n-y-las-v%C3%ADas-respiratorias/insuficiencia-respiratoria-y-s%C3%ADndrome-de-dificultad-respiratoria-aguda/s%C3%ADndrome-de-dificultad-respiratoria-aguda-sdra>
- Paules, C. (23 de Enero de 2020). *JAMA Network*. Obtenido de <https://jamanetwork.com/journals/jama/fullarticle/2759815>
- Plitt, L. (14 de Marzo de 2020). *BBC Extra*. Obtenido de <https://www.bbc.com/mundo/noticias-51858185>
- Proteus. (2021). *Proteus*. Obtenido de <https://www.labcenter.com/>
- Raspberry. (Junio de 2019). *UNIT ELECTRONIC*. Obtenido de <https://static.raspberrypi.org/files/product-briefs/USB-C-Product-Brief.pdf>
- Rivera, S. (30 de Marzo de 2020). *Los Angeles Times*. Obtenido de <https://www.latimes.com/espanol/california/articulo/2020-03-30/linea-de-tiempo-del-coronavirus-dentro-del-cuerpo-de-infeccion-a-recuperacion-o-muerte>
- RTVE.es. (6 de Septiembre de 2021). *rtve*. Obtenido de <https://www.rtve.es/noticias/20210906/variantes-del-coronavirus-mas-preocupan-cuantas-hay-donde-se-encuentran/2070949.shtml>

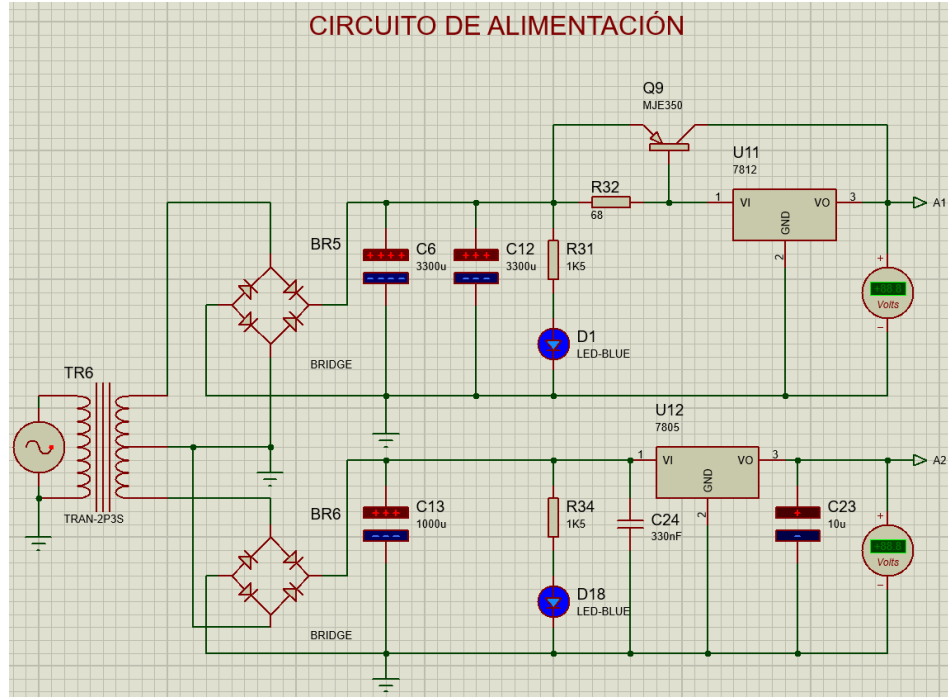
- Rus, C. (25 de Julio de 2019). *xataka*. Obtenido de <https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica>
- Semiconductor, O. (Octubre de 2011). *alltransistors*. Obtenido de [https://alltransistors.com/adv/pdfdatasheet\\_onsemi/mje15033g.pdf](https://alltransistors.com/adv/pdfdatasheet_onsemi/mje15033g.pdf)
- Sergio. (20 de Abril de 2019). *Raspberry para novatos*. Obtenido de <https://raspberryparanovatos.com/articulos/la-importancia-de-la-alimentacion-en-raspberry-pi/>
- Tolocka, E. (9 de Mayo de 2015). *profe Tolocka*. Obtenido de <https://www.profetolocka.com.ar/2015/05/09/modulo-de-4-reles-para-arduino/>
- Torres, J., Botero, J., & Celis, C. (14 de Septiembre de 2020). *Scielo*. doi:<https://doi.org/10.11144/Javeriana.umed61-4.fibr>
- Velleman. (16 de Abril de 2018). *robotlandia*. Obtenido de <https://robotlandia.es/motores/261-valvula-de-agua-electrovalvula-de-12v.html>
- West, J. (2016). *Fisiología Pulmonar*. Wolters Kluwer.

**ANEXOS**

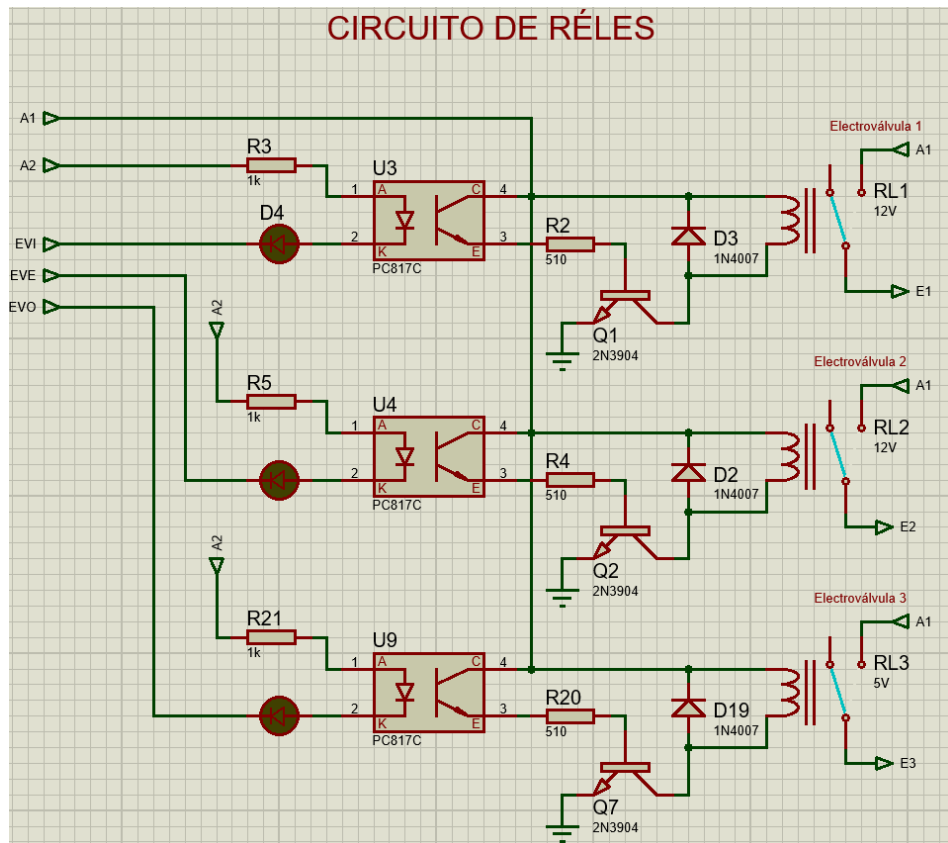
# Anexo 1: Diseño Electrónico

## Esquema de los circuitos del sistema

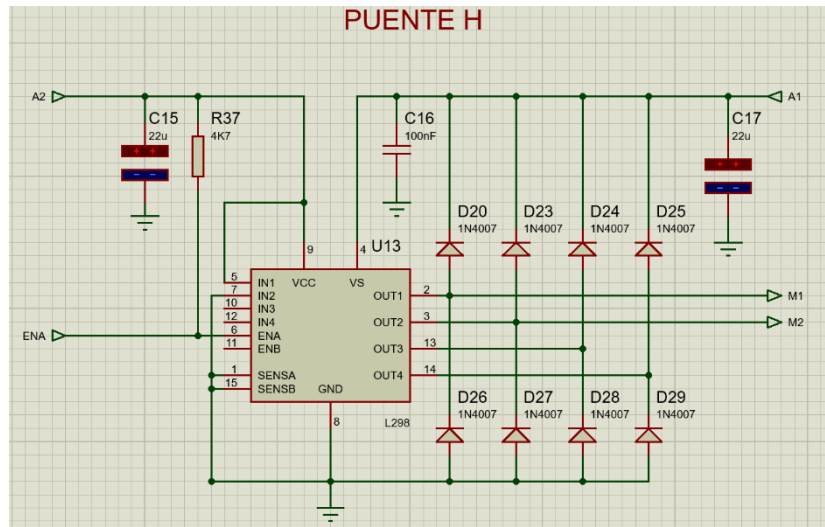
Circuito de Alimentación 12 [V] y 5 [V].



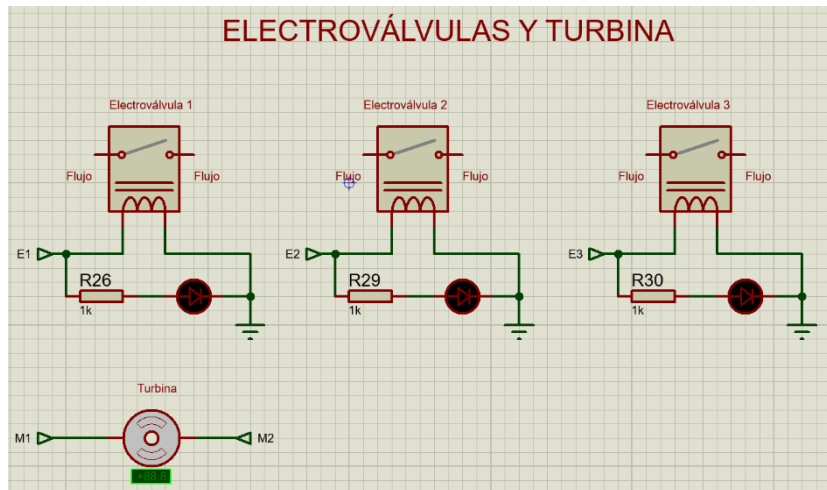
Esquema del circuito de control de electroválvulas.



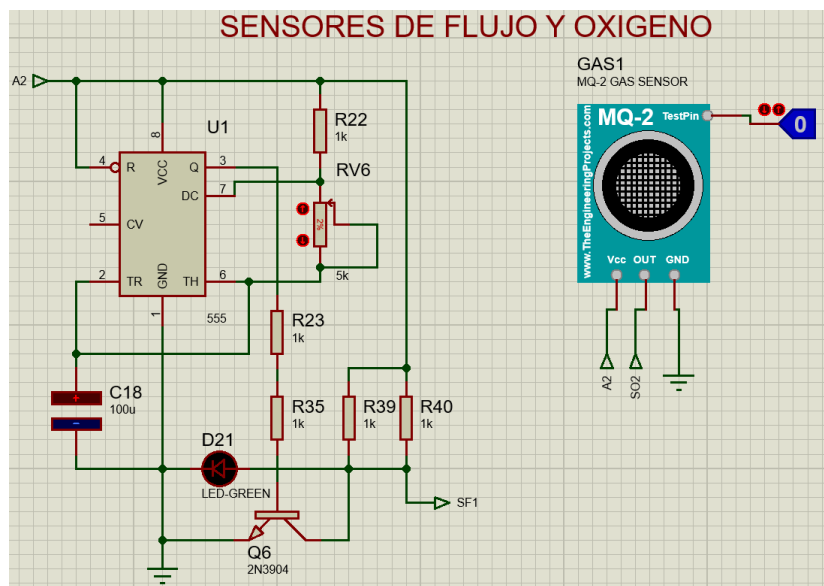
Esquema del circuito para el control del ventilador centrífugo.



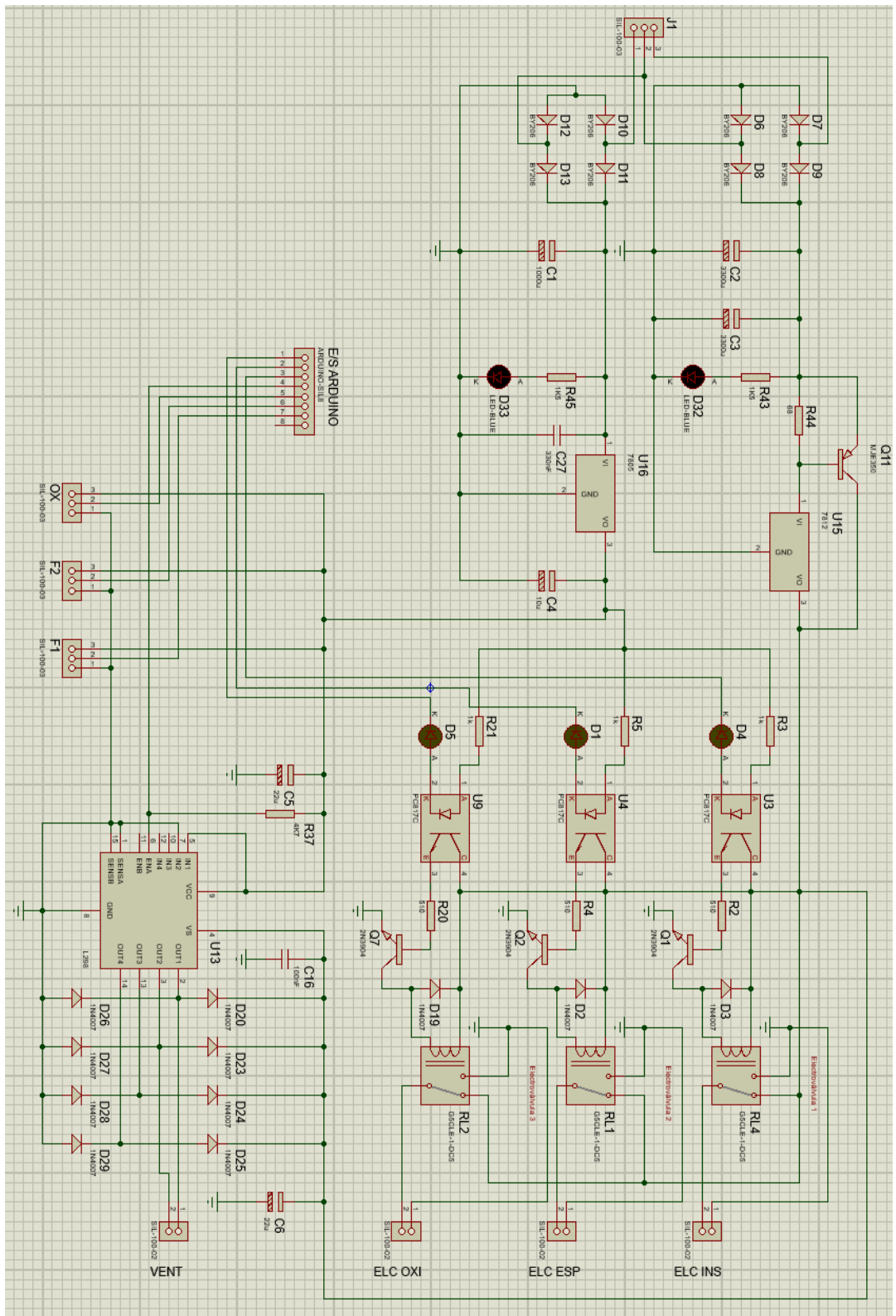
Esquema del circuito de las electroválvulas y motor centrífugo.



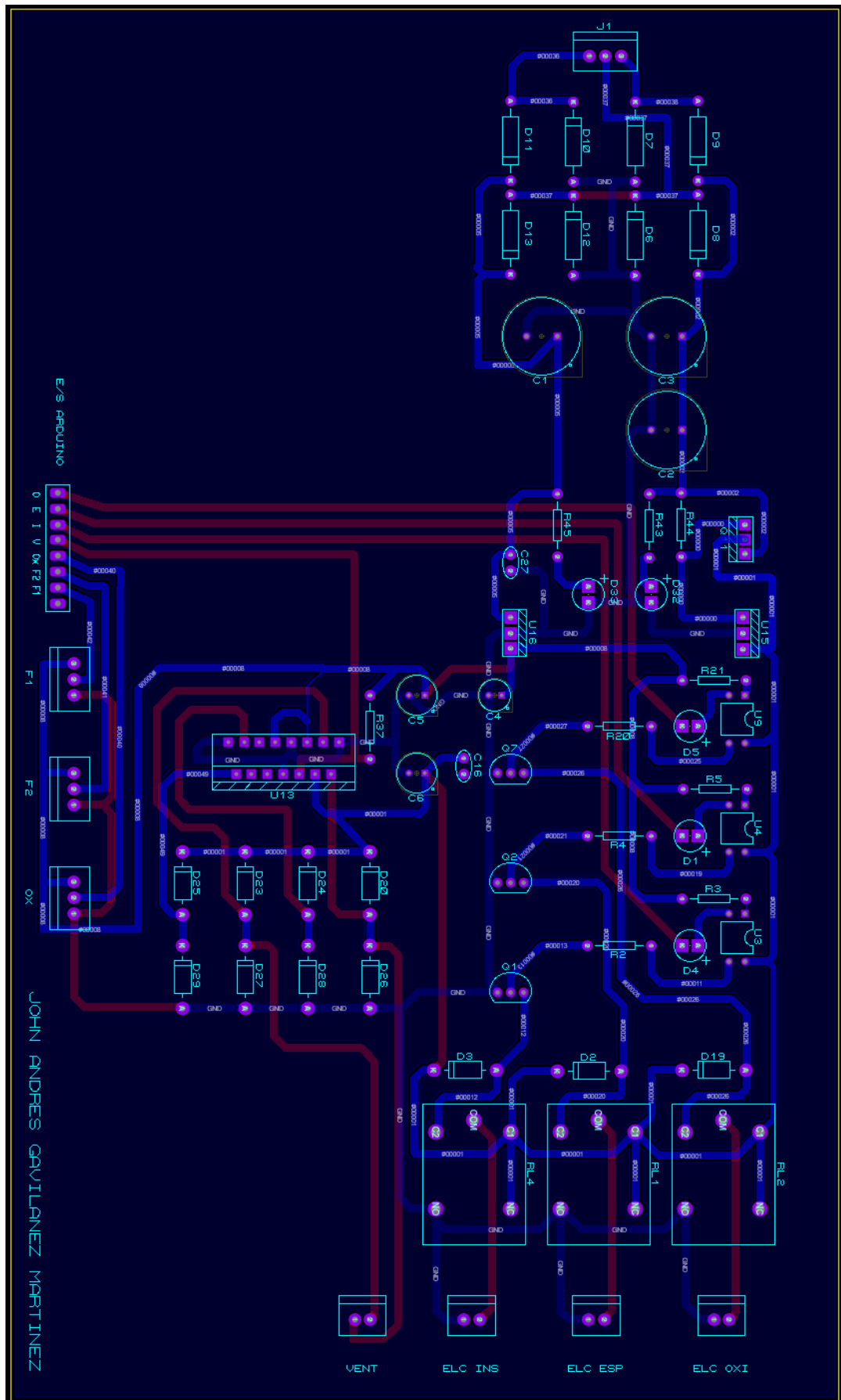
Representación de un sensor de flujo y oxígeno.



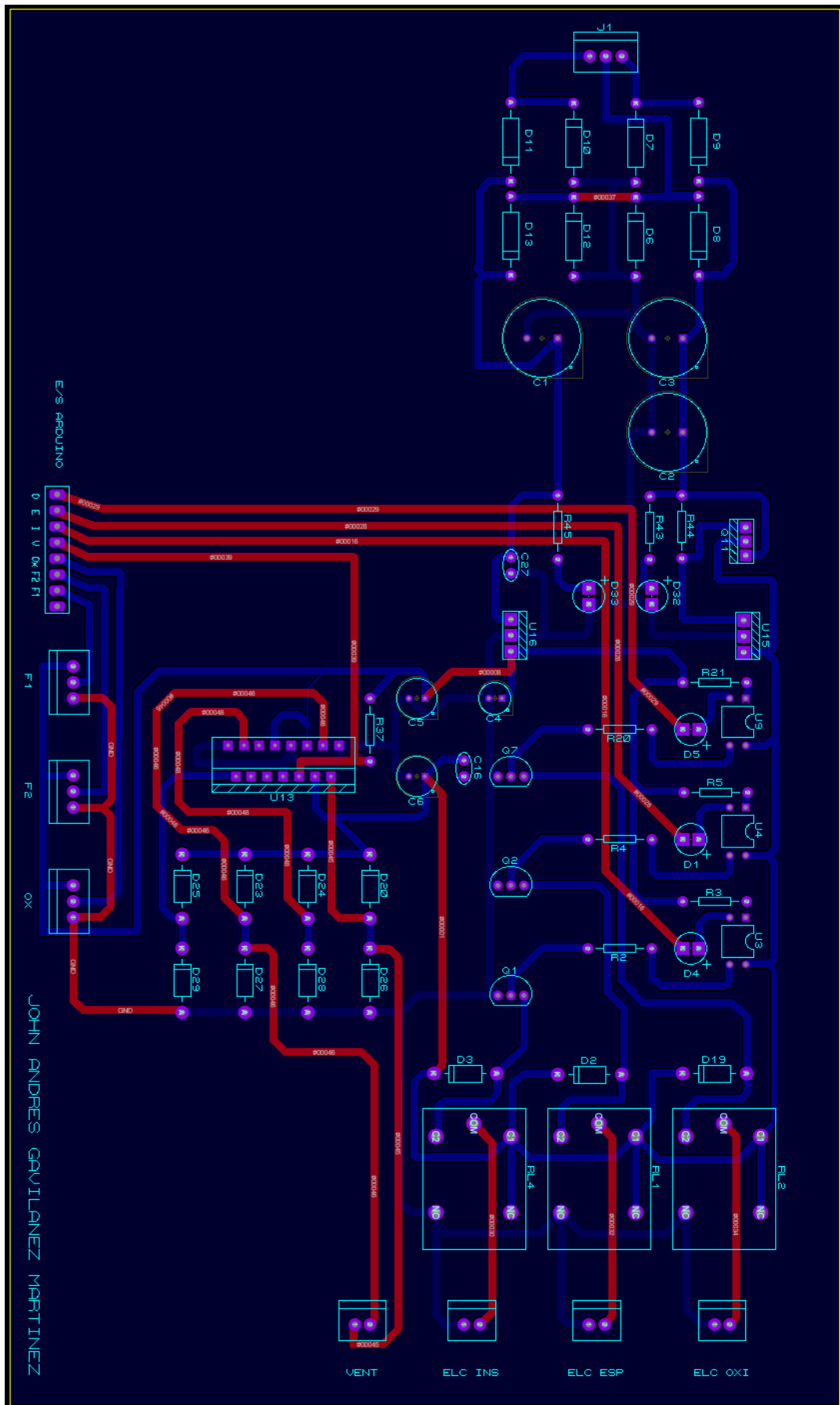
# Integración de los circuitos previo al diseño de la placa.



Diseño de la placa del circuito vista inferior.



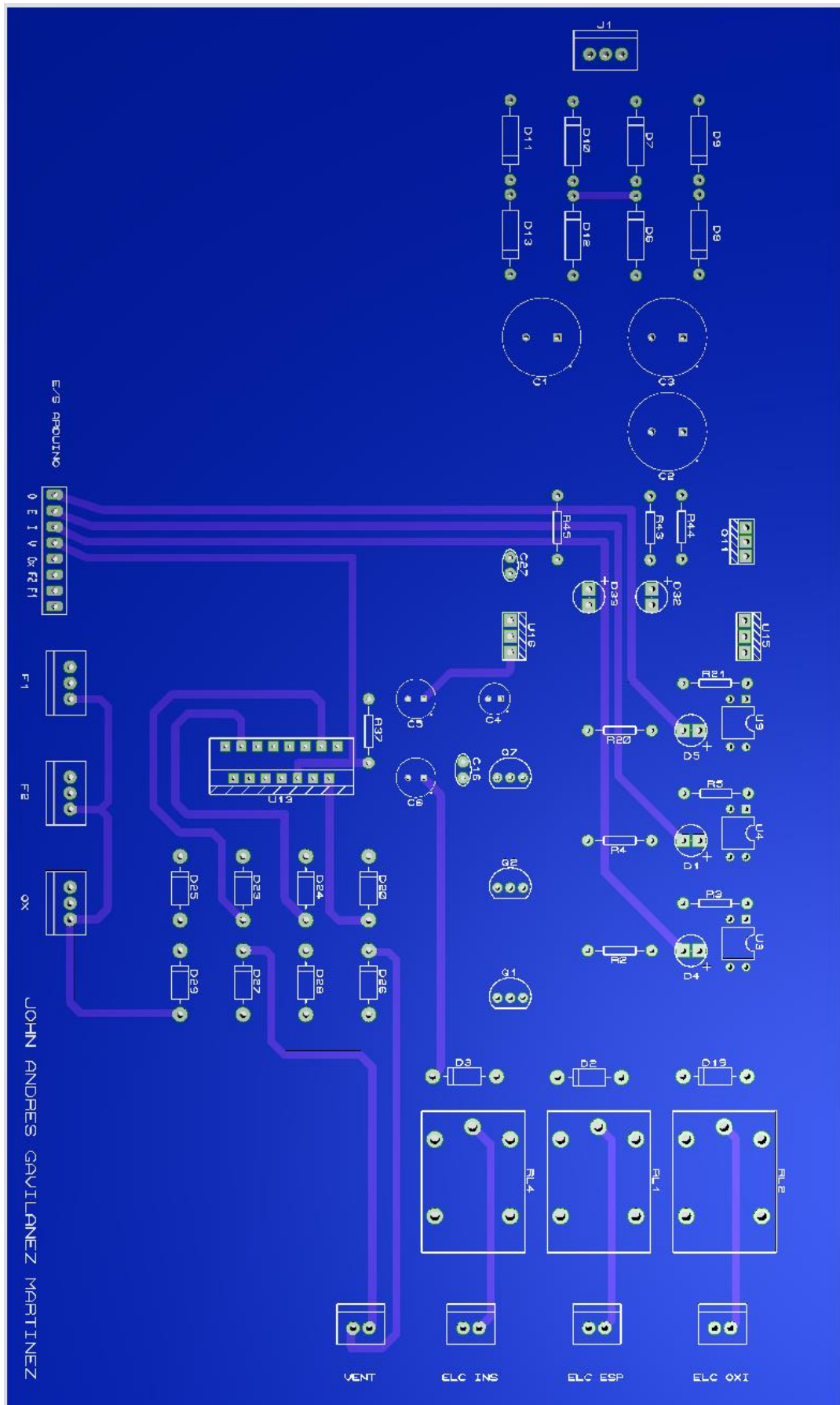
# Diseño de la placa del circuito vista superior



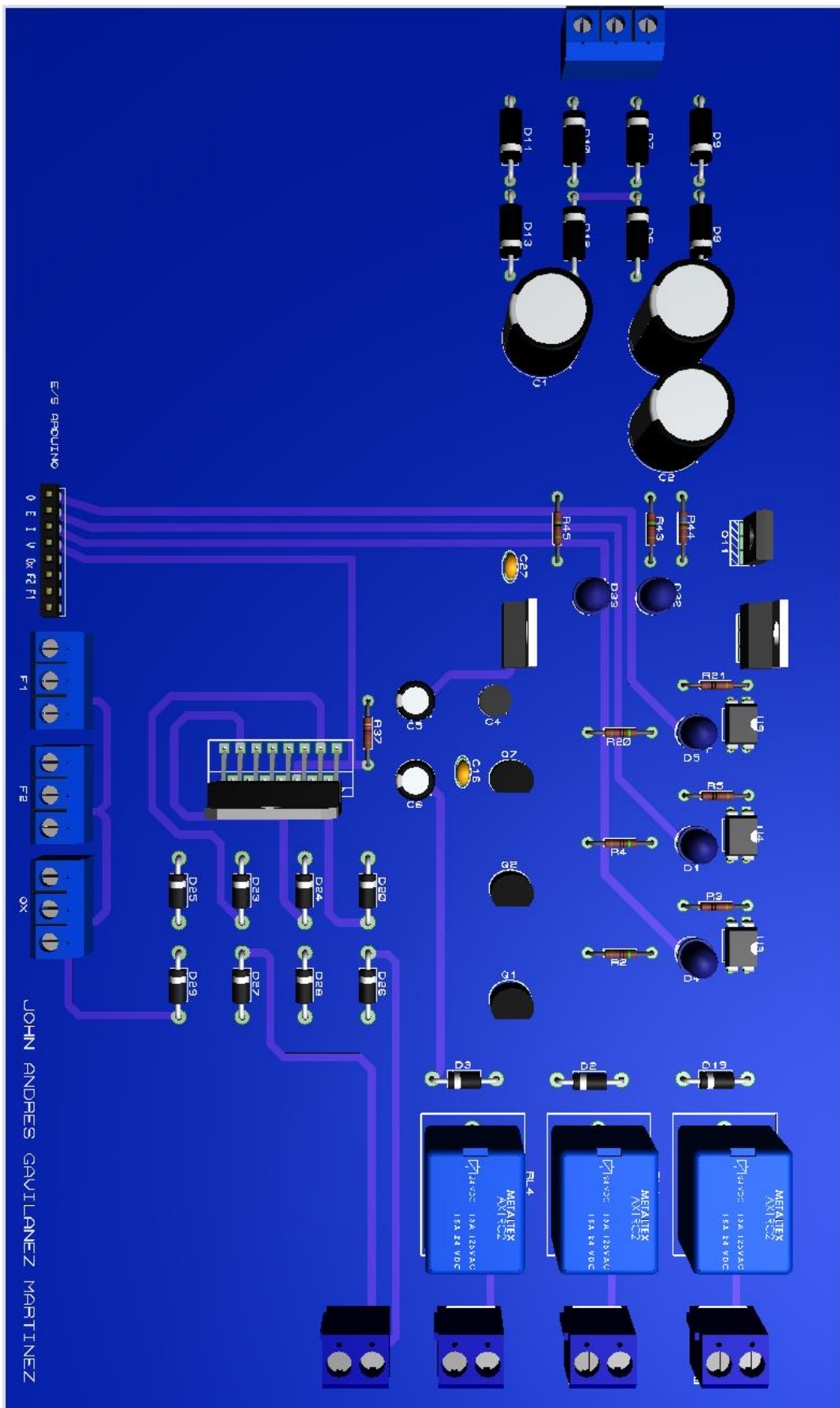
JOHN ANDRES GAVILANEZ MARTINEZ



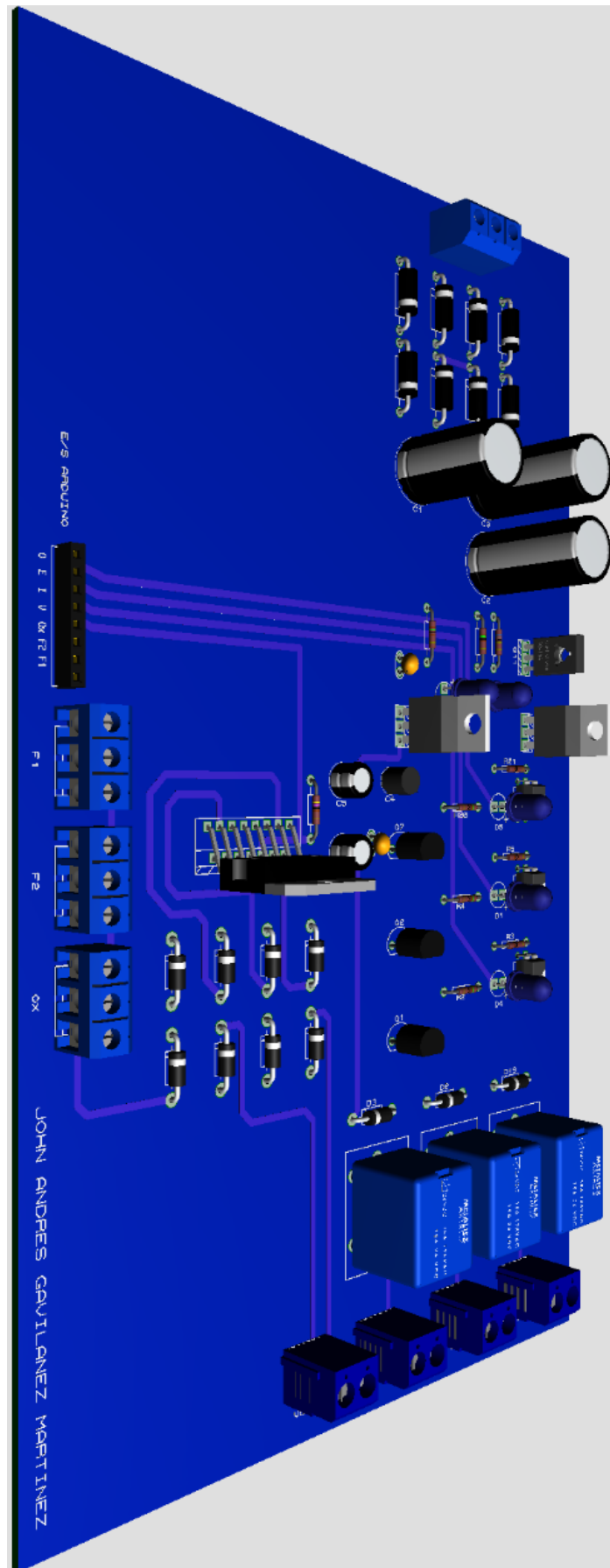
# Simulación de la placa diseñada del circuito vista superior



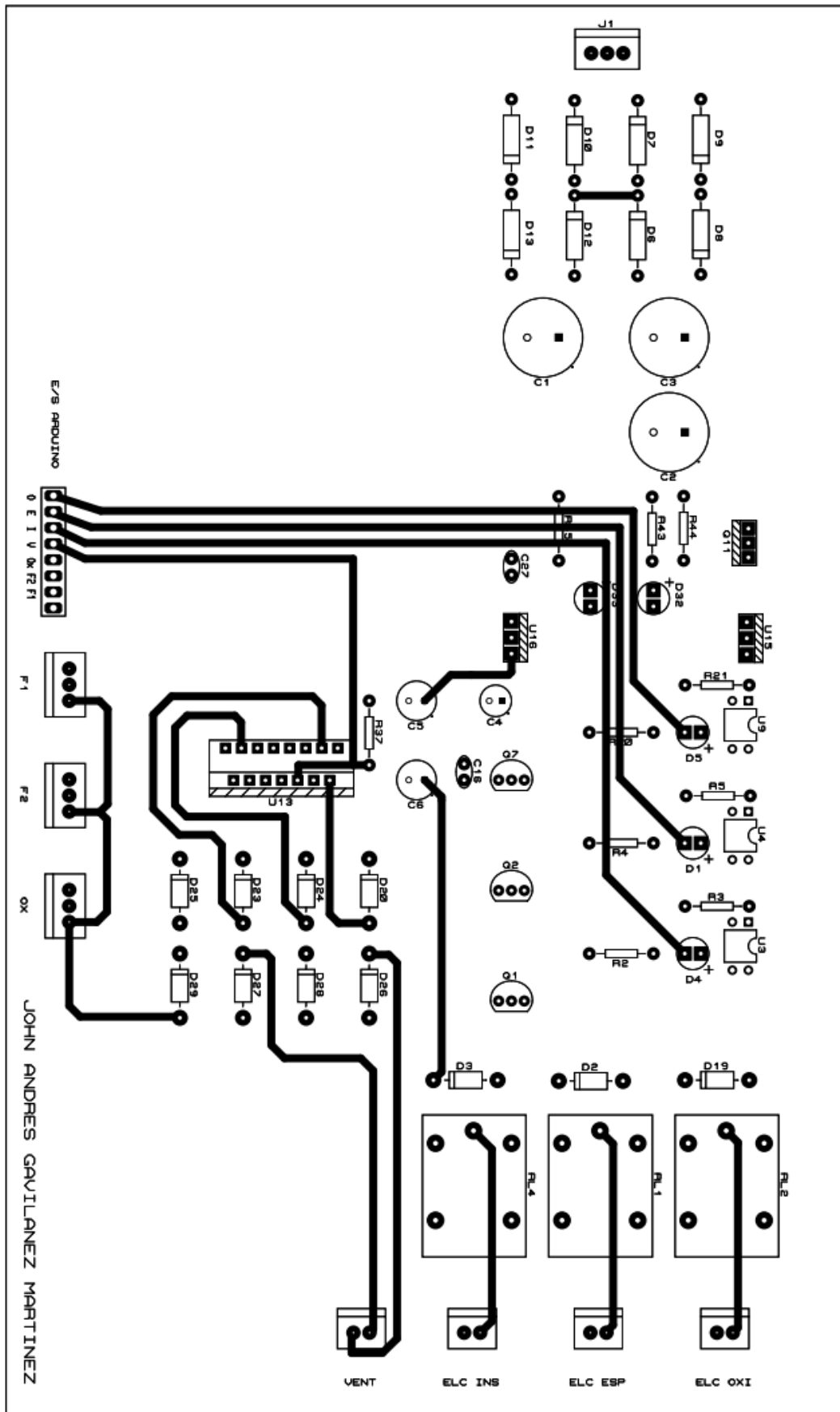
Simulación de la placa diseñada del circuito con los componentes.



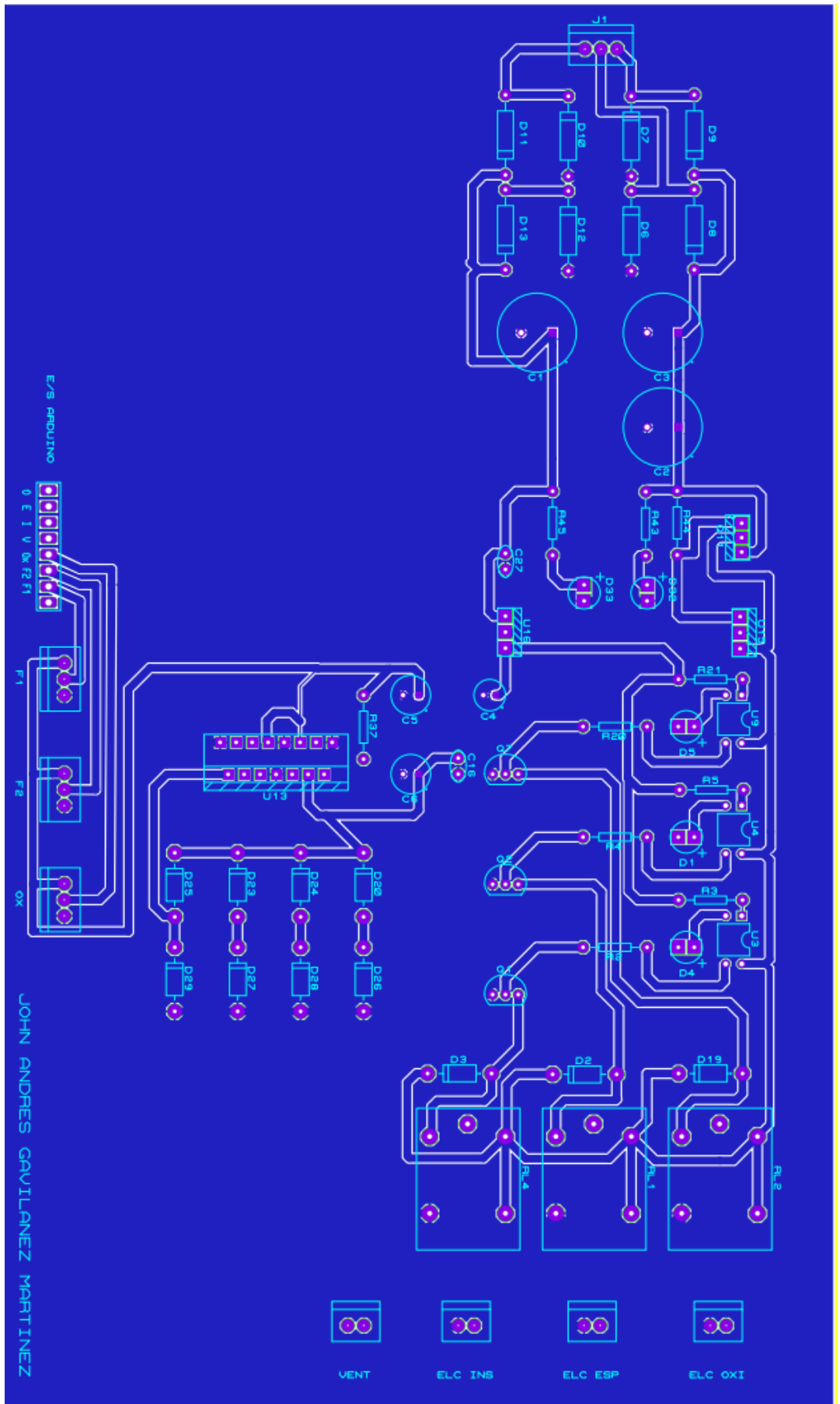
Vista en perspectiva del circuito



PCB generado por el programa para su fabricación parte superior.



PCB generado por el programa para su fabricación parte inferior.



VENT



ELC INS



ELC ESP



ELC OXI

## Anexo 2: Diseño Informático

### Código en IntelliJ IDEA

#### Asignación de variables

```
private SerialPort sp; //Comunicacion serial
final XYSeries series=new XYSeries( key: "Datos"); //creacion de la variable series
final XYSeriesCollection coleccion=new XYSeriesCollection(); //creacion de la variable coleccion

final XYSeries series1=new XYSeries( key: "Datos"); //creacion de la variable series1
final XYSeriesCollection coleccion1=new XYSeriesCollection(); //creacion de la variable coleccion1

private static Color ColorRecuadros = new Color( r: 11, g: 80, b: 110);
private static Color ColorFondo = Color.BLACK;
private static Color ColorSerie1 = new Color( r: 11, g: 80, b: 110);

JFreeChart Grafica; //cracion de la variable grafica
JFreeChart Grafica1;

int cg=0; //valor para el contador de la grafica
int valIn; //variable donde se guarda el valor entero del bufer
int E; //variable Edad
double IBW; //variable IBW peso ideal
double VTd; //variable VTd volumen tidal
boolean estadoVTd; //variable booleano Vtd visibilidad
boolean estadoFr; //variable booleano Fr visibilidad
boolean estadoRelc; //variable booleano Relc I:E visibilidad
boolean estadoSep; //variable booleano separacion : visibilidad
boolean estadoFiO; //variable booleano fraccion inspirada de oxigeno

boolean estadoPEEP; //?

double Shmax=-10000; //Valor maximo
int Shmin=10000; //?
int ValorPico=0; //?
int VpicoAnt=0; //?

int mapeo;
int operacion1;
long operacion2;
long operacion3;
long Ti=1;

double grafVolIn;
double grafVolEs;
int contInVol=0;
int contEsVol=14;

double grafFluIn;
double grafFluEs;
int contInFlu=0;
int contEsFlu=0;

//agregado

int a;
int b;
```

```

int c;
double d;
int e=0;
int f=15;

int cins=0;
int cexp=0;

long Tins;
long Texp;

int valor2=0;

//atributos

//constructor
public Primero (){

    this.setUndecorated(true); //icono y opciones de maximizar, cerrar, minimizar
    this.setExtendedState(MAXIMIZED_BOTH); //maximizar la pantalla
    this.setContentPane(pnlPrimero); //llamado de contenedor al pnlPrimero
    tblpContenido.setEnabledAt(index 1, enabled: false); //Desabilitar uno de los indices de tblContenido

    sldVTd.setVisible(false); //booleano sldVtd visibilidad
    sldFr.setVisible(false); //booleano sldFr visibilidad

```

```

sldPEEP.setVisible(false);
sldFI0.setVisible(false);

spIns.setVisible(false); //booleano spIns visibilidad
spExp.setVisible(false); //booleano spExp visibilidad
lblSep.setVisible(false); //booleano lblSep visibilidad

estadoVTd=true; //booleano estadoVtd click
estadoFr=true; //booleano estadoFr click
estadoRelc=true; //booleano estadoRelc click
estadoSep=true; //booleano estadoSep click
estadoFI0=true; //booleano estadoFI0 click

estadoPEEP=true; //?

//metodos creados
cbox(); // llamado del metodo Combox fechas
spinnerIns(); // llamado del metodo spinner insp
spinnerExp(); // llamado del metodo spinner exp
//metodos creados

```

```

//Graficas 1
series.add(x: 0, y: 0); //Inicializacion serie en la posicion 0,0
coleccion.addSeries(series); //Agregar serie a coleccion
Grafica=ChartFactory.createXYLineChart( title: " ",
    xAxisLabel: "Tiempo", yAxisLabel: "Volumen",
    coleccion, PlotOrientation.VERTICAL,
    legend: false, tooltips: false, urls: false); //Caracteristicas de la Grafica

```

```

Grafica.setBackgroundPaint(ColorFondo); //Color de Fondo
final XYPlot plot = (XYPlot) Grafica.getPlot(); //variable plot
configurarPlot(plot);
final XYLineAndShapeRenderer renderer = (XYLineAndShapeRenderer)plot.getRenderer();
configurarRendered(renderer);
final NumberAxis domainAxis = (NumberAxis)plot.getDomainAxis(); //crecionde variables domain para la grafica
configurarDomainAxis(domainAxis); //llamado de metodo configurarDomainAxis
final NumberAxis rangeAxis = (NumberAxis)plot.getRangeAxis(); //crecion de la variable rangeAxis
configurarRangeAxis(rangeAxis); //llamado del metodo configurarRangeAxis
//Graficas 1

//Graficas 2
series1.add(x: 0, y: 0); //Inicializacion serie en la posicion 0,0
coleccion1.addSeries(series1); //Agregar serie a coleccion
Grafica1=ChartFactory.createXYLineChart( title: " ",
    xAxisLabel: "Tiempo", yAxisLabel: "Flujo",
    coleccion1, PlotOrientation.VERTICAL,
    legend: false, tooltips: false, urls: false); //Caracteristicas de la Grafica

Grafica1.setBackgroundPaint(ColorFondo); //Color de Fondo
final XYPlot plot1 = (XYPlot) Grafica1.getPlot(); //variable plot
configurarPlot(plot1);
final XYLineAndShapeRenderer renderer1 = (XYLineAndShapeRenderer)plot1.getRenderer();
configurarRendered(renderer1);

```

```

final NumberAxis domainAxis1 = (NumberAxis)plot1.getDomainAxis(); //creacion de variables domain para la g
configurarDomainAxis(domainAxis1); //llamado de metodo configurarDomainAxis
final NumberAxis rangeAxis1 = (NumberAxis)plot1.getRangeAxis(); //creacion de la variable rangeAxis
configurarRangeAxis(rangeAxis1); //llamado del metodo configurarRangeAxis
//Graficas 2

//Configuracion Graficas
ChartPanel panel=new ChartPanel(Grafica); //creacion del objeto panel
pnlGrafica.setLayout(new BorderLayout()); //asignar nuevos bordes al pnlGrafica
pnlGrafica.add(panel,BorderLayout.CENTER); //agregar el objeto creado dentro del panel en el centro
pnlGrafica.repaint(); //imprimir el cuadro de la grafica
pack(); //llamdo del contenedor ChartPanel
//Configuracion Graficas

//Configuracion Graficas1
ChartPanel panel1=new ChartPanel(Grafica1); //creacion del objeto panel
pnlGrafica1.setLayout(new BorderLayout()); //asignar nuevos bordes al pnlGrafica
pnlGrafica1.add(panel1,BorderLayout.CENTER); //agregar el objeto creado dentro del panel en el centro
pnlGrafica1.repaint(); //imprimir el cuadro de la grafica
pack(); //llamdo del contenedor ChartPanel
//Configuracion Graficas1

this.pack(); //llamado de contenedor al pnlPrimero

```

## Programación de Botones

```

btmBuscar.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        try {

            String puertos[]= SerialPortList.getPortNames();
            Boolean l=Boolean.parseBoolean(puertos[0]);
            cbxPuertos.addItem ("-Seleccione-");

            if (l==false) {
                for (String n : puertos) {
                    cbxPuertos.addItem(n);
                }
            }
        }catch (Exception e1){

            JOptionPane.showMessageDialog( parentComponent: null, message: "No se detectaron puertos");
            cbxPuertos.removeAllItems();
        }

    }
});

```

```

btmConectar.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        if (lblFr.getText().equals("----")||lblIns.getText().equals("----")||
            lblExp.getText().equals("----")||lblPEEP.getText().equals("----")){

            JOptionPane.showMessageDialog(rootPane, message: "Por favor, agregue los parametros requeridos para el inicio");
        }else{

            sp = new SerialPort(cbxPuertos.getSelectedItem().toString());

            try{

                sp.openPort();
                sp.setParams( baudRate: 9600, dataBits: 8, stopBits: 1, parity: 0);
                sp.addEventListener(new LecturaSerial(sp),SerialPort.MASK_RXCHAR);

                //agregado
                lblTInsp.setText(String.format("%.2f", (60/Double.parseDouble(lblFr.getText())/
                    (Double.parseDouble(lblIns.getText())+Double.parseDouble(lblExp.getText()))));
                //agregado

                //Thread.sleep(1000)
            }
        }
    }
});

```



```

btmSalir.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        try{
            //sp.closePort(); //revisar activa y desactivado

            System.exit( status: 0);

        }catch (Exception e1){

        }

    }
});
//Salir

//Continuar
btmContinuar.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        try{

            if (txtNombres.getText().isEmpty()||txtApellidos.getText().isEmpty()){

```

```

            }
            else{

                if(rbHombre.isSelected()){

                    int respuesta = JOptionPane.showConfirmDialog (rootPane, message: "La informacion sera ingresada, dese

                    if (respuesta==JOptionPane.YES_OPTION){

                        lblNombres.setText(txtNombres.getText()); //envio de caja de texto al label
                        lblApellidos.setText(txtApellidos.getText()); //envio de caja de texto al label
                        Fecha();

                        //Calculo de peso corporal IBW y Volumen tidal
                        IBW=(Double.parseDouble(txtEstatura.getText())-152.4)*0.91+50;
                        VTd=IBW*6;
                        lblPeso.setText(String.valueOf((int) Math.round (IBW)));
                        sldVTd.setValue((int) Math.round(VTd));
                        //Calculo de peso corporal IBW y Volumen tidal

                        tblpContenido.setSelectedIndex(1); //seleccionar indice del tbl
                        tblpContenido.setEnabledAt( index: 1, enabled: true); //habilitar opcion dos indice 1
                        tblpContenido.setEnabledAt( index: 0, enabled: false); //deshabilitar opcion uno indice 0

                    }

                }
            }
        }
    }
});

```

```

        }
    }
});

int respuesta = JOptionPane.showConfirmDialog (rootPane, message: "La informacion sera ingresada, dese

if (respuesta==JOptionPane.YES_OPTION){

    lblNombres.setText(txtNombres.getText()); //envio de caja de texto al label
    lblApellidos.setText(txtApellidos.getText()); //envio de caja de texto al label
    Fecha();

    //Calculo de peso corporal IBW y Volumen tidal
    IBW= (Double.parseDouble(txtEstatura.getText())-152.4)*0.91+45.5;
    VTd=IBW*6;
    lblPeso.setText(String.valueOf((int) Math.round (IBW)));
    sldVTd.setValue((int) Math.round(VTd));
    //Calculo de peso corporal IBW y Volumen tidal

    tblpContenido.setSelectedIndex(1); //seleccionar indice del tbl
    tblpContenido.setEnabledAt( index: 1, enabled: true); //habilitar opcion dos indice 1
    tblpContenido.setEnabledAt( index: 0, enabled: false); //deshabilitar opcion uno indice 0

}

}
}

JOptionPane.showMessageDialog(rootPane, message: "Faltan campos por completar");

```

## Cajas de texto

```
//Cajas de texto
txtNombres.addKeyListener(new KeyAdapter() {
    @Override
    public void keyTyped(KeyEvent e) {
        super.keyTyped(e);

        char validar=e.getKeyChar();

        if (Character.isDigit(validar)){
            getToolkit().beep();
            e.consume();
            JOptionPane.showMessageDialog(rootPane, message: "En este campo no se admiten numeros");
        }
    }
});

txtApellidos.addKeyListener(new KeyAdapter() {
    @Override
    public void keyTyped(KeyEvent e) {
        super.keyTyped(e);

        char validar=e.getKeyChar();

        if (Character.isDigit(validar)){
            getToolkit().beep();
            e.consume();
        }
    }
});

txtEstatura.addKeyListener(new KeyAdapter() {
    @Override
    public void keyTyped(KeyEvent e) {
        super.keyTyped(e);{
            char validar=e.getKeyChar();

            if (txtEstatura.getText().length()==4||Character.isLetter(validar)){

                getToolkit().beep();
                e.consume();
                JOptionPane.showMessageDialog(rootPane, message: "En este campo no se admiten letras, ni valores superiores");
            }
        }
    }
});
//Cajas de texto
```

## Slider de parámetros

```
sldAltura.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) { txtEstatura.setText(sldAltura.getValue() + ""); }
});

sldVTD.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {
        lblVTD.setText(sldVTD.getValue()+"");
    }
});

sldFr.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {
        if (E>=5&&E<=14) {
            sldFr.setMaximum(30);
            sldFr.setMinimum(20);
            lblFr.setText(sldFr.getValue()+"");
            lblTTC.setText(String.format("%.2f",60/Double.parseDouble(lblFr.getText())));
        }else if (E>=15&&E<=18){

```

```

        sldFr.setMaximum(15);
        sldFr.setMinimum(12);
        lblFr.setText(sldFr.getValue()+"");
        lblTTC.setText(String.format("%.2f",60/Double.parseDouble(lblFr.getText())));

    }else if (E>=19&&E<=65){

        sldFr.setMaximum(14);
        sldFr.setMinimum(8);
        lblFr.setText(sldFr.getValue()+"");
        lblTTC.setText(String.format("%.2f",60/Double.parseDouble(lblFr.getText())));

    }

});

sldPEEP.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {

        lblPEEP.setText(sldPEEP.getValue()+"");

    }
});

sldFIO.addChangeListener(new ChangeListener() {
    sldFIO.addChangeListener(new ChangeListener() {
        @Override
        public void stateChanged(ChangeEvent e) {

            lblFIO.setText(sldFIO.getValue()+"");

        }
    });
//slider

//spiner
spIns.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {

        lblIns.setText(spIns.getValue().toString());

    }
});
spExp.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {

        lblExp.setText(spExp.getValue().toString());
    }
});

```

## Conexión de con Arduino

```

class LecturaSerial implements SerialPortEventListener{

    private SerialPort sp;
    //constructor
    public LecturaSerial(SerialPort sp) {

        this.sp = sp;

    }
    //constructor

    //metodo
    @Override
    public void serialEvent(SerialPortEvent serialPortEvent) {

        if (serialPortEvent.isRXCHAR()) {

            if (sp.isOpened()){

                Tins tins=new Tins();
                Grafica h = new Grafica();
                envio1 En1=new envio1();
                envio2 En2=new envio2();
                envio3 En3=new envio3();
            }
        }
    }
}

```

```

class Tins extends Thread{

    public void run(){

        if (Integer.parseInt(lblIns.getText()) == 1 && Integer.parseInt(lblExp.getText())>=1){

            lblTinsp.setText(String.format("%.2f", (60/Double.parseDouble(lblFr.getText()))/
                (Double.parseDouble(lblIns.getText())
                +Double.parseDouble(lblExp.getText()))));

            Tins= (Long) (((60/Double.parseDouble(lblFr.getText()))/(Double.parseDouble(lblIns.getText())+
                Double.parseDouble(lblExp.getText())))*1000);
            Texp= (Long) (((60/Double.parseDouble(lblFr.getText()))-(60/Double.parseDouble(lblFr.getText()))/
                (Double.parseDouble(lblIns.getText())
                +Double.parseDouble(lblExp.getText())))*1000);

        }else if (Integer.parseInt(lblExp.getText()) == 1 && Integer.parseInt(lblIns.getText())>=1){

            lblTinsp.setText(String.format("%.2f", (60/Double.parseDouble(lblFr.getText()))-
                ((60/Double.parseDouble(lblFr.getText()))/(Double.parseDouble(lblIns.getText())+
                Double.parseDouble(lblExp.getText()))));

            Tins= (Long) (((60/Double.parseDouble(lblFr.getText()))-(60/Double.parseDouble(lblFr.getText()))/
                (Double.parseDouble(lblIns.getText())
                +Double.parseDouble(lblExp.getText())))*1000);
            Texp= (Long) (((60/Double.parseDouble(lblFr.getText()))/(Double.parseDouble(lblIns.getText())+
                Double.parseDouble(lblExp.getText())))*1000);

        }

    }

}

class Grafica extends Thread{

    public void run(){

        //lectura del arduino

        try {

            if (cg<=100){

                byte[] buffer1 = sp.readBytes(); //cadena en byte
                valIn = buffer1 [0] & 0xff; //valores enteros
                mapeo=mapVtd(valIn, intmin: 0, intmax: 254, outtmin: 0, outtmax: 1);
                Ti=Tins;
                operacion1=mapeo*sldVTd.getValue();
                operacion2=operacion1/(Ti/10);
                operacion3=(operacion2*100);

                if (operacion1>0){

                    //volumen ins
                    grafVolIn=operacion1*(0.067+(0.1*(contInVol/1.5))); //3.21
                    //volumen ins

                    //flujo ins
                    grafFLUIn=operacion3;
                    //flujo ins

                }

            }else{

                //volumen exp
                grafVolEs=sldVTd.getValue()*(0.067+(0.1*(contEsVol/1.5)));
                //volumen exp

                //flujo exp
                grafFLUEs=sldVTd.getValue()*-Math.exp(-contEsFlu);
                //flujo exp

                contInVol=0;
                contEsVol--;
                contEsFlu++;

                series.add(cg, grafVolEs);
                series1.add(cg, grafFLUEs);

            }

            cg++;

        } else{

            cg=0;
            series.clear();

        }

    }

}

```

## Envió de Datos

```
class envio2 extends Thread{

    public void run(){

        //funcional 3 envio de datos enteros

        try {

            int [] buffer3=new int[]{2,sldPEEP.getValue(),sldFI0.getValue()};

            sp.writeIntArray(buffer3);

        } catch (SerialPortException e) {
            //e.printStackTrace();
        }

    }

}
```

```
class envio1 extends Thread{

    public void run(){

        try {

            a=mapVtd(sldVtd.getValue(),sldVtd.getMinimum(),sldVtd.getMaximum(), outmin: 0, outmax: 255);
            int [] buffer2=new int[]{1,a,sldFr.getValue()};
            sp.writeIntArray(buffer2);

        } catch (SerialPortException e) {
            //e.printStackTrace();
        }

    }

    int mapVtd(int x,int inmin,int inmax,int outmin,int outmax){
        return ((x-inmin)*(outmax-outmin))/((inmax-inmin)+outmin);
    }

}
```

```
class envio3 extends Thread{

    public void run(){

        //funcional 3 envio de datos enteros

        try {

            int [] buffer3=new int[]{3,Integer.parseInt(lblIns.getText()),Integer.parseInt(lblExp.getText())};

            sp.writeIntArray(buffer3);

        } catch (SerialPortException e) {
            //e.printStackTrace();
        }

    }

}
```

# Código Arduino

## Asignación de variables

```
int sensorOx = A0; //sensor de Oxigeno
int sensorFlu1 = 2; //Sensor de flujo 1
int SensorFlu2 = 3; //Sensor de flujo 2
//sensores

//variables sensor de flujo
volatile int pulsos; //variable que acumula los pulsos recibidos
double lmin; //litros por minuto
float fconver=7.5; //factor de conversion puede ser 6.67
unsigned long tiempoAnterior;
unsigned long pulsosAcumulados;
float litros;
//variables sensor de flujo

//valores analog prueba
int analogPin = A1;
int analogValuel = 0;
//valores analog prueba

//actuadores
//const int LED1=4;
//const int LED2=5;
const int Vc=6; //motor
int flujoIns;

const int VOx=7; //electrovalvulaO2
const int Vol=9;

//variables adicionales
float valorSensorO2;
float voltajeSensorO2;
float valorO2;

int a=0;

int Op=0;
int Fr=0;
int PEEP=0;
int O2=0;
int Ins=0;
int Exp=0;

double Tins;
double Texp;

//frecuencia respiratoria
unsigned long currentTime=0;
unsigned long previousTime=0;
unsigned long interval,intervall, interval2;
bool Vins=LOW; //Electrovalvulal
bool Vesp=HIGH; //Electrovalvulal
bool DatoIn=LOW;

void setup() {

//inicio de comunicacion serial
Serial.begin(9600);
//inicio de comunicacion serial

//sensor de flujo
pinMode(sensorFlu1, INPUT); //Inicializar como entrada
attachInterrupt(0,flujo,RISING); // inicializamos las interrupciones ,metodo,flanco_de_subida

//prueba
interrupts();
tiempoAnterior=millis();
//prueba
//sensor de flujo

//definir estado de la salida
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(Vc,OUTPUT);
pinMode(VOx,OUTPUT); //electrovalvulaO2

pinMode(Vol,OUTPUT);
pinMode(10,OUTPUT);
//definir estado de la salida

}
```

```

void loop() {

  Sensores();
  Input=lmin;
  Setpoint=a;
  Output=computePID(Input);
  delay(100);
  analogWrite(Vc,Output);

  if(Serial.available()>3){

    int lectura = Serial.readBytes(bufferIn,3);           //instruccion que solo da entrada l
    proceso(lectura);                                     //llamar a metodo proceso

  }else{

    Vins=HIGH;
    Vesp=HIGH;
    digitalWrite(Vc,LOW);
    digitalWrite(VOx,LOW);

    digitalWrite(Vol,LOW);
    digitalWrite(LED7,LOW);

    DatoIn=HIGH;

  }
}

```

## Métodos para sensores

```

void Sensores(){

  SensorFlul();
  SensorOx();
  lecturaA();

}

//sensor de flujo ins
void flujo ()
{
  pulsos++;
}

void SensorFlul(){

  if( millis() - tiempoAnterior > 1000){

    tiempoAnterior = millis(); // Updates cloopTime
    pulsosAcumulados += pulsos;
    lmin = (pulsos/ 6.67); // (Pulse frequency x 60 min) / 7.5Q = flow rate in L/hour
    pulsos = 0; // Reset Counter
    //Serial.println(lmin);

  }

}

void SensorOx(){

  valorSensorO2 = analogRead(sensorOx);
  voltajeSensorO2 = (valorSensorO2/1023)*5.0;
  valorO2 = (voltajeSensorO2*20.94)/2.47;
  //Serial.println(valorO2);

}

//sensor de Oxigeno

//Lectura analoga prueba
void lecturaA(){

  analogValue1 = analogRead(analogPin);
  analogValue1 = map(analogValue1, 0, 1023, 0, 255);
  Serial.write(analogValue1);|
  delay(100); //tiempo necesario para enviar valores

}

```

## Lectura de datos

```
void proceso(int entrada){

    Op=bufferIn[0];

    //primero
    if(Op==1){

        a=map(bufferIn[1],0,255,0,1023);
        Fr=bufferIn[2];
        flujoIns=a/Ins;
        analogWrite(Vol,flujoIns);

    //primero

    //segundo
    }else if (Op==2){

        PEEP=bufferIn[1];
        O2=bufferIn[2];

        if(valorO2<=O2){

            digitalWrite(VOx,HIGH);

        }else{

            digitalWrite(VOx,LOW);

        }

    }

}

void frecuenciaRes(){

    if(Ins==1&&Exp>=1){

        Tins=((60/Fr)/(Exp+Ins))*1000;
        Texp=((60/Fr)-((60/Fr)/(Exp+Ins)))*1000;
        interval1=Texp; //close
        interval2=Tins; //open
        currentTime=millis();

        if(Vins){
            interval=interval2;
        }else{
            interval=interval1;
        }
        if((currentTime-previousTime)>interval){
            previousTime=currentTime;
            Vins=!Vins;
            Vesp=!Vesp;

            DatoIn=!DatoIn;

            digitalWrite(4,Vins);
            digitalWrite(5,Vesp);

            digitalWrite(10,DatoIn);

        }

    }

}
```