



Acta de Correcciones al Proyecto de Grado Ingeniería Electrónica

Fecha: abril 10 de 2024

Autores: Juan Camilo Restrepo Zuleta

Nombre del Proyecto de Grado: Diseño e implementación de un sistema tecnológico para el proceso de validación de equipos de apoyo a las terapias respiratorias.

Director: Manuel Vicente Valencia Diaz

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Manuel V. Valencia

Firma de Director(a) del Proyecto de Grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero Electrónico.

Camilo Rocha

Dr. HERNAN CAMILO ROCHA NIÑO
Decano de la Facultad de Ingeniería

[Handwritten signature]

Dr. LUIS EDUARDO TOBON LLANO
Director Carrera Ingeniería Electrónica.

Manuel V. Valencia

M. Sc. MANUEL VICENTE VALENCIA DIAZ
Director(a) Trabajo

[Handwritten signature]

ING. Valentina Corchuelo
Jurado 1

[Handwritten signature]

ING. Hernán Darío Vargas
Jurado 2



Pontificia Universidad
JAVERIANA
Cali

Diseño e implementación de un sistema tecnológico para el
proceso de validación de equipos de apoyo a las terapias
respiratorias.

Juan Camilo Restrepo Zuleta

Director: M. Sc. Manuel Vicente Valencia Diaz

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias.
Ingeniería Electrónica.
Proyecto de Grado.

Santiago de Cali, .

Señores

Pontificia Universidad Javeriana Cali.

Dr. Luis Eduardo Tobón

Director Carrera de Ingeniería Electronica

Cali.

Cordial Saludo.

Por medio de la presente me permito informarle que el estudiante de Ingeniería electrónica Juan Camilo Restrepo Zuleta (cod: 6022362) trabaja bajo mi dirección en el proyecto de grado titulado “Diseño e implementación de un sistema tecnológico para el proceso de validación de equipos de apoyo a las terapias respiratorias.” y autorizar el envío del documento al comite de evaluación.

Atentamente,

Manuel V. Valencia

M. Sc. Manuel Vicente Valencia Diaz

Santiago de Cali, .

Señores

Pontificia Universidad Javeriana Cali.

Dr. Luis Eduardo Tobón

Director Carrera de Ingeniería Electronica.

Cali.

Cordial Saludo.

Me permito presentar a su consideración el proyecto de grado titulado “Diseño e implementación de un sistema tecnológico para el proceso de validación de equipos de apoyo a las terapias respiratorias.” con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el proyecto de grado y posteriormente optar al título de Ingeniero Electronico.

Al firmar aquí, doy fe que entiendo y conozco las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería y Ciencias aprobadas el 26 de Noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del anteproyecto y del trabajo de grado.

Atentamente,



Juan Camilo Restrepo Zuleta

Código: 6022362

Resumen

Resumen—Este documento presenta el diseño e implementación de un sistema mecatrónico cuyo propósito es, emular el proceso de respiración en la fase de inspiración y espiración de una persona para que de esta forma se pueda verificar y validar la operación de dispositivos de apoyo a las terapias de respiración por medio de diferentes tipos de rutinas lo cual permitirá probar los diferentes tipos de terapias en estos dispositivos para verificar su precisión y el comportamiento frente a estas, de esta forma se facilitará el acceso de las personas a estas tecnologías en búsqueda de mejorar sus procesos de recuperación de la capacidad pulmonar.

Palabras Clave: Respiración, pandemia, terapias, actuador y controlador.

Abstract— This document presents the design and implementation of a mechatronic system whose purpose is to emulate the breathing process during the inspiration and expiration phases of a person. This allows the verification and validation of the operation of devices supporting breathing therapies through various types of routines. It will enable testing different types of therapies on these devices to verify their accuracy and performance in response to them. Thus, it will facilitate people's access to these technologies in order to improve their pulmonary capacity recovery processes.

Keywords: Breathing, pandemic, therapies, actuator, and controller.

Índice

1. Descripción del Proyecto	9
1.1. Planteamiento del Problema	9
1.2. Formulación	9
1.2.1. Sistematización	10
1.3. Objetivos	10
1.3.1. Objetivo General	10
1.3.2. Objetivos Específicos	10
1.4. Justificación	10
1.5. Marco Teórico	12
1.5.1. Trabajos Relacionados	12
1.5.2. Base Teórica	15
2. Diseño del Software y Hardware	22
2.1. Diseño del Sistema Hardware	22
2.1.1. Descripción General	22
2.1.2. Diagrama de flujo del funcionamiento del HMI.	24
2.1.3. Diseño mecánico.	26
2.1.4. Selección de los componentes electrónicos	28
2.1.5. Componentes seleccionados	29
2.2. Diseño del sistema software	34
2.2.1. Descripción General	34
3. Implementación del Software y Hardware	38
3.1. Implementación del Sistema Hardware	38
3.1.1. Sistema mecánico	38
3.1.2. Calibración del Sensor	39
3.1.3. Conexiones	40
3.2. Implementación del Sistema Software	43
3.2.1. Control Arduino	43
3.2.2. Movimiento del Motor	44
3.2.3. Interfaz	47
4. Pruebas y Análisis de Resultados	53
4.1. Pruebas	53
4.1.1. Pruebas Mecánicas	53
4.1.2. Pruebas de Software	54
4.2. Resultados	55

4.2.1. Resultados iniciales	55
4.2.2. Resultados finales	56
5. Conclusiones y Recomendaciones de Trabajo Futuro	61
5.1. Recomendaciones de Trabajo Futuro	61
5.2. Conclusiones	61
Bibliografía	63
6. Anexos	67
6.1. Anexos	67
6.1.1. Clasificación de los dispositivos médicos	67
6.1.2. Alternativas mecánicas	70
6.1.3. Alternativas para los componentes	72
6.1.4. Manual de usuario	76
6.1.5. Protocolo de pruebas	82

Introducción

A finales del año 2019 se dio un brote de neumonía causado por un virus sin identificar en Wuhan, la capital de la provincia de Hubei, en China. Al virus causante de esta neumonía se lo identificó como síndrome respiratorio agudo severo – coronavirus 2 (SARS – CoV – 2). En muy poco tiempo esta enfermedad se expandió por todo el mundo, llegando a contagiar a más de 70.000 personas en tan solo 2 meses [1]. Las principales secuelas de este virus en los casos graves son las alteraciones pulmonares intersticiales que producen disnea o sensación de ahogo.

Cuatro años después de esto y gracias a las vacunas se ha logrado casi erradicar las afectaciones graves causadas por este virus, sin embargo, la neumonía todavía puede ser causada por otras bacterias u hongos. Esta enfermedad, al igual que la COVID-19 y diversos procedimientos quirúrgicos, compromete las funciones pulmonares; cuando esto pasa, se ha identificado una pérdida de volumen y capacidad pulmonar. Debido a esto, es necesario que se usen técnicas de re expansión o terapias respiratorias con la ayuda de dispositivos, como el incentivo respiratorio que permite mejorar la función pulmonar, la excursión del diafragma y el desempeño en las actividades del día a día [2]. Estos dispositivos de incentivo respiratorio deben ser calibrados para poder ser usados y actualmente este proceso de calibración se realiza manualmente [3].

Por esto, es necesario diseñar e implementar un sistema tecnológico, el cual permita validar en varios inspirómetros la precisión y exactitud de las variables críticas como el flujo en una terapia respiratoria. El objetivo de este proyecto es, pues, la propuesta de un sistema actuador automático que emule la función respiratoria, y que además tenga las ventajas de ser automático, versátil y portable.

De tal forma, se presenta en el Capítulo 1 de ese documento la definición del problema y sistematización de la problemática junto con la justificación del porqué es importante desarrollar este proyecto, el cual debe cumplir con unos objetivos que se identifican como las metas a alcanzar hasta culminar el proceso, teniendo en cuenta los alcances y limitaciones. También, se mencionan trabajos relacionados que sirven como base para el trabajo a desarrollar. Posteriormente, se expone el marco teórico, que se centra en abrir el panorama sobre los conceptos básicos del tema. En el capítulo 2 se abordará el proceso de diseño del sistema, donde se hará una descripción general del sistema y de los componentes y materiales seleccionados para la implementación. En el capítulo 3 se hablará acerca de la implementación del sistema y se dará una explicación de su funcionamiento. En el capítulo 4 se expondrán las pruebas hechas luego de la implementación y los resultados obtenidos de estas pruebas. Por último, en el capítulo 5 se mostrarán las conclusiones y recomendaciones de trabajo futuro.

Descripción del Proyecto

1.1. Planteamiento del Problema

¿Actualmente es necesaria la implementación de dispositivos que ayuden a la calibración automática de equipos usados para las terapias respiratorias? Con base a esta pregunta, para este proyecto se buscó trabajar en un sistema actuador que emule el proceso de respiración de una persona que consiste en inspirar y espirar, porque se requiere de un dispositivo que pueda validar las medidas de flujo que esté recibiendo el inspirómetro y en el cual se puedan probar diferentes tipos de rutinas de respiración o tipos de flujo inspiratorio como el flujo constante, el flujo desacelerado o el flujo sinusoidal. [5].

Una vez validado y probado el inspirómetro, este va a poder estar listo para el uso en personas para así ayudar a los sistemas de salud que los requieran, esto debido a que durante la pandemia de la COVID-19 se pudo evidenciar que los casos más graves producen problemas respiratorios que durante y después de superada la enfermedad tienen que ser tratados con terapia respiratoria [2]. Aunque ya no se den tantos casos de esta enfermedad, siguen existiendo otro tipo de enfermedades y procedimientos quirúrgicos que producen afecciones respiratorias a las personas, las cuales eventualmente deben ser tratadas con terapias en algunos casos.

Debido al alto costo y la escasez de este tipo de dispositivos de apoyo a las terapias respiratorias que se vivió durante la pandemia, a la imposibilidad de hacer pruebas en personas por la alta tasa de contagio de muchos de los virus, y a los trámites que se deben hacer con los entes que aprueban o rechazan esta clase de dispositivos, es necesario que se produzca un sistema que pueda emular estos comportamientos respiratorios.

1.2. Formulación

¿Cómo brindar una herramienta de fácil uso, que sea automática, versátil y portable que permita simular y medir los flujos de aire por medio de rutinas para así validar los equipos usados en las terapias respiratorias?

1.2.1. Sistematización

- ¿Cuáles son las principales características de los sistemas tecnológicos empleados para la validación de inspirómetros en terapias respiratorias?
- ¿Qué características técnicas y requerimientos funcionales debe tener un actuador para emular la función respiratoria?
- ¿Cómo definir un sistema que permita emular la función respiratoria?
- ¿Cómo desarrollar un sistema que permita emular la función respiratoria?
- ¿Cómo validar que la implementación del sistema actuador automático emula correctamente la función respiratoria?

1.3. Objetivos

1.3.1. Objetivo General

Implementar un sistema actuador automático capaz de emular la función respiratoria en la fase de inspiración y espiración bajo distintas secuencias según los requerimientos de las terapias respiratorias.

1.3.2. Objetivos Específicos

- Identificar las principales características de los sistemas tecnológicos empleados para la validación de los inspirómetros incentivos usados en las fisioterapias respiratorias.
- Definir las características principales y requerimientos funcionales de un actuador para emular la función respiratoria bajo diferentes secuencias de inspiración y espiración.
- Diseñar y especificar los elementos mecánicos y la instrumentación de un sistema emulador de función respiratoria según estándares vigentes.
- Implementar un sistema tipo actuador automático que emule la función respiratoria para la calibración y validación de sistemas de incentivo respiratorio.
- Mostrar el flujo generado por las rutinas de respiración mediante una interfaz que le permita al usuario visualizar estos datos en un formato tipo gráfica y tabla para su posterior análisis.

1.4. Justificación

Hace cuatro años el mundo atravesó la pandemia de la COVID-19, enfermedad que es causada por el coronavirus conocido como SARS-CoV-2. Según la OMS, alrededor del 15% de las personas que contraen este virus desarrollan una enfermedad grave y requieren oxígeno, mientras que el 5%

llegan a un estado crítico y precisan de cuidados intensivos [7]. Estas personas que llegan a la unidad de cuidados intensivos, en su mayoría, presentan insuficiencia respiratoria hipoxémica, sin embargo, una serie de tomografías computarizadas de tórax tomadas en pacientes al inicio de la pandemia revelaron que existen diferentes etapas de afectación respiratoria en estas personas [8].

Algunas personas que logran superar la enfermedad siguen experimentando síntomas que van desde fatiga hasta diversos síntomas respiratorios y neurológicos, por ende, deben someterse muchas veces a una fisioterapia respiratoria para, de esta forma, poder recuperarse por completo. No obstante, la fisioterapia de rehabilitación también se hace en muchos casos mientras la persona está enferma. Al haber diferentes tipos de afectaciones, se requieren diversos tipos de fisioterapias respiratorias, que incluyen ventilación mecánica, para las cuales se necesitan equipos como los inspirómetros de incentivo [2].

Entre las razones por las que se decidió hacer este proyecto está la escasez de equipos que ayuden a validar los inspirómetros usados para las terapias respiratorias, lo que se pudo evidenciar en la pandemia, donde debido a la gran cantidad de casos que había, estos equipos empezaron a ser más escasos y los nuevos que se estaban fabricando se demoraban en estar disponibles por la cantidad de normas y regulaciones que deben cumplir, como lo son la Directiva 93/42 CEE, la cual se aplica para todos los dispositivos de uso médico y la estandarización de espirometría de 2019 para los inspirómetros usados en las terapias de respiración [9].

Igualmente, dentro de la investigación que se realizó, se encontró que, según datos de la OPS, realizar terapias de respiración va a beneficiar a los sistemas de salud y a las personas, ya que pueden reducir las complicaciones relacionadas con la internación en una UCI, como el síndrome post-UCI. Asimismo, facilita el alta temprana de pacientes que se encontraban en la unidad de cuidados intensivos [10], lo cual supone un gran beneficio ya que reduce la cantidad de camas ocupadas en el caso que exista una gran demanda de estas camas. Por último, reduce el riesgo de reingreso, ya que estas rehabilitaciones ayudan a que los pacientes no se deterioren o sufran una recaída luego de haber salido de cuidados intensivos [10].

Con base en lo mencionado, el realizar un dispositivo que sea capaz de emular el proceso de respiración de una persona para validar los equipos usados en las terapias resulta muy beneficioso ya que es un dispositivo que puede entrar a ayudar al sistema de salud. Esto debido a que se puede verificar la precisión de los inspirómetros con ayuda de dispositivos de referencia como por ejemplo el CITREX H3 y comprobar nuevas rutinas de respiración que sean más eficientes, las cuales sirven para recuperar pacientes que se encuentran graves o pacientes que han sufrido secuelas de enfermedades respiratorias, lo cual, a su vez, ayuda a que la disponibilidad de camas en la UCI sea mayor.

1.5. Marco Teórico

1.5.1. Trabajos Relacionados

Para dar respuesta a los dos primeros objetivos específicos del proyecto, se investigó acerca de diferentes sistemas creados durante la pandemia del Covid-19 o después de esta, para suplir la necesidad de ventiladores mecánicos que existía en ese momento, estos sistemas eran capaces de emular la respiración de una persona o asistirle durante este proceso por lo que iba a ayudar a definir las principales características o requerimientos funcionales necesarios para diseñar un sistema de este tipo. También se investigó acerca de los tipos de calibración que existen actualmente para identificar las principales características de los sistemas y métodos usados actualmente en la calibración de dispositivos usados en terapias respiratorias. Dentro de los diseños encontrados, en un caso se halló un dispositivo de prueba que simula al pulmón, para esto, inicialmente se usa una ecuación que describe al pulmón como un sistema mecánico, luego se obtuvieron los parámetros clave del dispositivo por medio de simulaciones en donde el dispositivo fue calibrado usando sensores de presión y desplazamiento. Aparte de esto, está equipado con un manómetro y una aguja que indica el volumen para que se pueda usar independientemente de la adquisición de datos el sistema [11]. Como se puede apreciar mejor en la figura 1.1.



Figura 1.1: Pulmón mecánico de prueba [11].

En otro caso, físicos e ingenieros de las universidades de Medea y Naama en Argelia y Malasia desarrollaron un ventilador de bajo costo que se hizo con la intención de suplir la falta de ventiladores causada por la pandemia. Los parámetros los encontraron por medio de análisis numéricos y, según varios ventiladores fabricados, decidieron investigar dos en concreto. El primero es un ventilador mecánico basado en un resucitador BVM como se puede ver en la figura 1.2, el cual presenta dificultades para controlar las variables de ventilación, lo que en un tiempo provocaría más problemas de salud como el barotrauma. Por otro lado, el ventilador mecánico basado en pistón y cilindro resulta más exitoso en lo que respecta a los parámetros asignados al paciente. Sin embargo, la salida del ventilador se ve muy afectada por el flujo de entrada [12]. Otra propuesta hecha para suplir esta necesidad de ventiladores mecánicos fue la de crear un ventilador mecánico basado en una inteligencia artificial (IA) como se puede ver en la figura 1.3. Lo que proponen los

autores en [13] es usar el ventilador propuesto por los autores en [12] y agregarle una inteligencia artificial que sea capaz de tomar decisiones en cuanto a cómo se debe manejar la terapia del paciente.

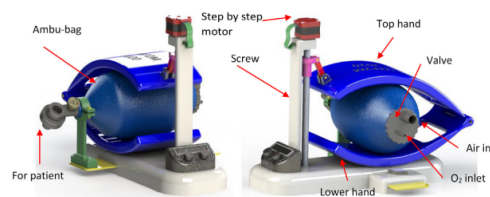


Figura 1.2: Ventilador mecánico basado en la bolsa ambu [12].

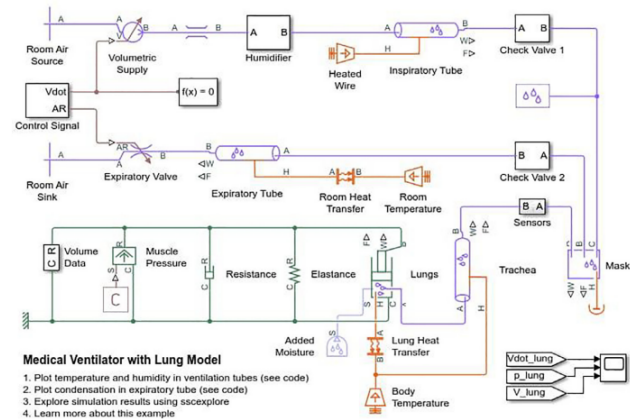


Figura 1.3: Modelo de ventilador basado en inteligencia artificial [13].

Siguiendo la idea de un ventilador con inteligencia artificial, se tiene otro dispositivo basado en un resucitador manual llamado “Masi”. Este dispositivo está dividido en varios bloques funcionales, el primero es el circuito del ventilador, en el cual la entrada del oxígeno se establece en un nanómetro de bajo flujo con los valores recomendados por un experto de la salud, a través de un sensor de FiO_2 hasta el resucitador reutilizable como se puede ver en la figura 1.4.

Además de lo anterior, el diseño del sistema comprende tres bloques principales. En primer lugar, el circuito eléctrico establece que el sistema debe estar alimentado por un tomacorriente AC con conexión a tierra. La potencia será administrada por una fuente de alimentación ininterrumpida AC/DC con una potencia nominal de 96 W y una entrada de voltaje automático de 90 VAC a 264 VAC. En segundo lugar, el sistema mecánico incluye un mecanismo de accionamiento del reanimador manual. Este mecanismo se basa en un sistema de paletas unidas a un eje que sirve como pivote. El extremo superior de las paletas recibe la fuerza a través de una transmisión de cadena. Finalmente, el tercer bloque es la interfaz remota. Esta interfaz permite a los usuarios conectarse a un servidor web alojado en el ventilador. Desde esta interfaz remota, los usuarios pueden visualizar la presión, el flujo y los gráficos de volumen. Por otro lado, tienen la capacidad de seleccionar los parámetros programables de forma remota. [14]

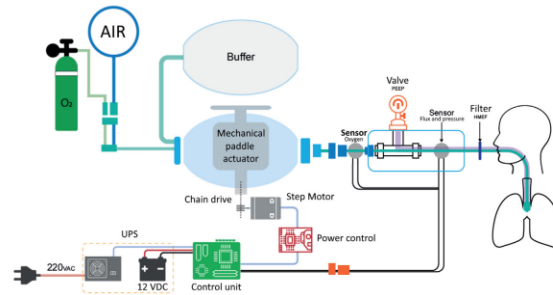


Figura 1.4: Diagrama de bloques del resucitador manual "Masi"[14].

Para pasar a un ámbito más local, en el año 2020 la Universidad del Valle se encontraba desarrollando un sistema de ventilación mecánica, llamado “VentyNet” para atender a personas afectadas por el COVID-19. Este prototipo, como se puede ver en la figura 1.5, está pensado para que incluya un sistema de filtrado de aire de entrada y salida, un módulo que permita visualizar los históricos del comportamiento del paciente y un protocolo de rehabilitación respiratoria con manejo local o remoto [15].

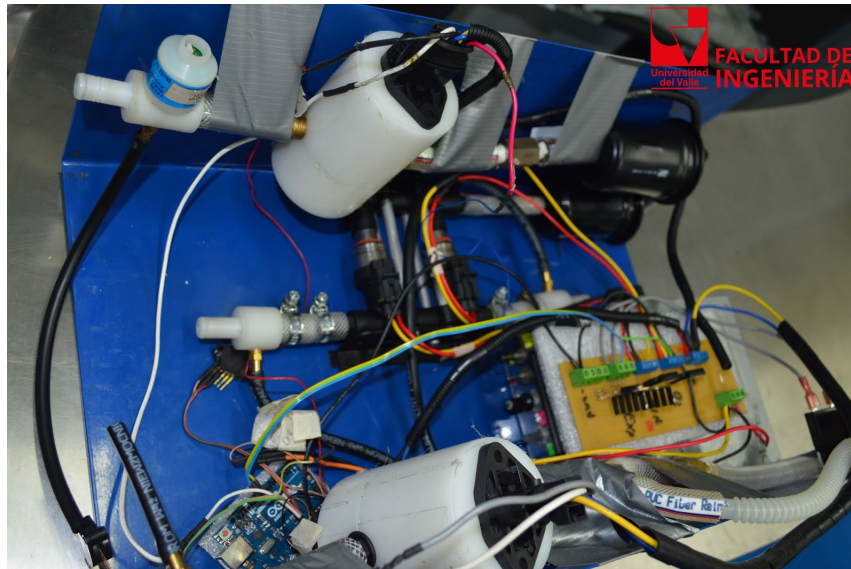


Figura 1.5: Prototipo "VentyNet"[15].

Luego de conocer las diferentes alternativas a respiradores mecánicos, se debe hablar de los tipos de métodos y alternativas que hay para realizar calibraciones de los equipos de apoyo a terapias respiratorias. Los tres tipos de calibración más usados son: la calibración estática, la cual se recomienda realizar diariamente, método que se realiza con ayuda de una jeringa de calibración y consiste en conectar el dispositivo a la jeringa y realizar un mínimo de cinco emboladas, las cuales simulan

el proceso de inspiración y espiración; este movimiento debe realizarse de manera controlada, sin movimientos bruscos o interrupciones y garantizando de que la velocidad del registro sea de 2 cm/s para espirómetros de campana y secos [16].

En segundo lugar, la calibración dinámica, que se recomienda realizar semanalmente. Esta se realiza con ayuda de un descompresor explosivo tipo Petusevsky, equipo compuesto por un cilindro estanco de 4 litros de volumen, un manómetro de presión, una electroválvula y un juego de cuatro resistencias al paso del flujo de aire con un diámetro calibrado. Para este método, se llena el cilindro con aire a una atmósfera de presión, se colocan las resistencias según su enumeración de menor a mayor, se conecta el descompresor al espirómetro y se realiza la expulsión de aire del cilindro luego de presionar el interruptor. Es importante tener en cuenta que el interruptor debe quedar presionado hasta que la aguja del manómetro marque 0 de presión, este paso se debe repetir con cada resistencia [16].

Finalmente, otra de las alternativas es la calibración de la linealidad de flujo. Para este método se necesita de un dispositivo más sofisticado que esté compuesto por un generador de flujos, un regulador de presión para incrementar o disminuir el flujo y un caudalímetro. Esta calibración se realiza conectando el calibrador al espirómetro y realizar un mínimo de tres emboladas para cada punto de presión, se debe iniciar con los valores bajos de flujo para ir aumentando hasta los valores máximos. Las dos escalas de flujo usadas para esta calibración son de 0 a 12 litros/segundo y de 0 a 3 litros/segundo [16].

1.5.2. Base Teórica

1.5.2.1. Conceptos claves

1. *Mediciones de flujo espiratorio (FEP)*: las mediciones de flujo espiratorio son fundamentales ya que por medio de estas se puede determinar el grado de obstrucción bronquial; además, contribuyen a diagnosticar en enfermedades respiratoria agudas y permiten el seguimiento de enfermedades crónicas, aunque no existen valores de referencias en sujetos sanos [18].
2. *Inspirómetro incentivo (IS)*: otro concepto clave en este proyecto es el del inspirómetro incentivo (IS, por sus siglas en inglés), este es un dispositivo que estimula el logro de inspiraciones máximas sostenidas a través de retroalimentación visual o auditiva. Estos dispositivos son ampliamente usados en las terapias respiratorias para incrementar la ventilación alveolar y la capacidad residual funcional, la cual es la capacidad de aire que mantienen los pulmones oxigenados luego de una exhalación normal, lo que ocurre en condiciones donde la ventilación del pulmón está deteriorada o en peligro [19].
3. *Terapia respiratoria*: cómo se mencionaba anteriormente, estas terapias son el proceso que se le realiza a una persona cuando presenta dificultades para respirar ya sea causadas por el COVID-19 o por cualquier otra enfermedad. Realizar estos tipos de terapias es muy beneficioso ya que se ha demostrado en estudios que las personas sometidas a estas terapias tienden

a recuperarse más rápido, por lo que se les puede dar de alta rápido para que, de esta forma otra persona que se encuentre en un estado delicado pueda ocupar la cama de la UCI [10].

1.5.2.2. Tipos de terapia respiratoria

Puesto que existen varios tipos de afecciones respiratorias, se requieren de diferentes tipos de terapia para así poder abordar cada caso de la mejor manera posible. Dentro de estos casos se encontró que, para las personas más graves, se usan procesos convencionales que incluyen técnicas de limpieza de las vías respiratorias, expansión pulmonar a través de maniobras que buscan abrir los alvéolos colapsados, ventiladores, terapias inhalatorias, humidificación y traqueotomía y bronco aspiración [2].

Un tipo de terapia es la fisioterapia de tórax durante la ventilación mecánica. La fisioterapia temprana mientras se usa un ventilador mecánico con el paciente ha demostrado ser viable y segura para evitar de esta forma que el paciente vuelva a tener problemas en el futuro una vez superada la enfermedad [2].

Existen técnicas de reexpansión pulmonar, las cuales son modalidades terapéuticas usadas generalmente en pacientes hospitalizados que han presentado complicaciones respiratorias luego de algunas cirugías como la cardiaca. El objetivo de estas técnicas es lograr la reexpansión pulmonar por medio de inspiraciones máximas, para de esta forma elevar la presión transpulmonar a niveles mayores que los alcanzados durante la ventilación en reposo [21].

Una vez el paciente sale de hospitalización, debe seguir realizando otro tipo de reexpansiones pulmonares, entre estos se encuentran: ventilación tranquila o patrón diafragmático, ventilación a nivel de capacidad inspiratoria media, ventilación a nivel de capacidad inspiratoria máxima o apnea máxima post inspiratoria, patrón ventilatorio con inspiración fraccionada en tiempos, patrón con suspiros inspiratorios, patrón ventilatorio a nivel de capacidad residual funcional, inspiración profunda sostenida y presión positiva intermitente [20].

Para la ventilación tranquila o patrón diafragmático se debe respirar de una manera suave y tranquila, este tipo de respiración se enfoca en el musculo del diafragma y se deben realizar de 12 a 16 respiraciones por minuto dejando un segundo de intervalo entre cada inspiración. Por otro lado, para la ventilación a nivel de capacidad máxima o apnea máxima post inspiratoria se debe realizar un tipo de respiración profunda de manera nasal, pausada y evitando el excesivo aumento del trabajo respiratorio. En esta respiración se realiza una pausa inspiratoria o apnea de 3 a 10 segundos, lo que tiene como objetivo lograr que el gas inspirado se reparta por todo el pulmón [20].

Por ultimo, para la inspiración profunda sostenida se debe realizar una respiración lenta y profunda de manera nasal y sosteniendo el aire de 2 a 5 segundos. Uno de los principales objetivos de esta técnica es mantener y mejorar los volúmenes y capacidades pulmonares luego de lograr la

apertura de los alveolos afectados [20].

1.5.2.3. Tipos de flujo inspiratorio

Muchos de los ventiladores mecánicos que se usan y que podrían ser usados para calibrar inspirómetros permiten modificar la modalidad en la que entra el aire al paciente [22]. Como se puede ver en la figura 1.6, existen cuatro tipos de flujo inspiratorio, según las características de cada paciente y su enfermedad se pueden usar diferentes tipos de flujo.

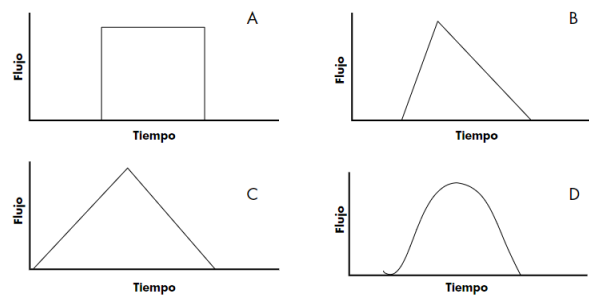


Figura 1.6: Tipos de flujo inspiratorio [23].

El flujo constante o de onda cuadrada (A) es muy común en la modalidad de volumen y su característica principal es que el flujo se mantiene constante durante la etapa de entrada de aire al paciente. Por otro lado, el flujo decelerado (B) es aquel en el cual el aire entra rápidamente al inicio de la inspiración y luego va disminuyendo su velocidad. En el caso contrario, se tiene el flujo acelerado (C) en el cual la entrada de aire es lenta al principio, pero luego va aumentando su velocidad. Finalmente, el flujo sinusoidal (D) es aquel en el cual se inicia la toma de aire de forma lenta, acelerando hasta alcanzar un máximo y luego desacelerando de forma progresiva [23].

1.5.2.4. Medición de capacidad pulmonar

Estos equipos usados para las terapias respiratorias poseen una serie de volúmenes y capacidades que posteriormente son revisadas por profesionales de la salud, los cuales son:

1. Volumen corriente (VT o *Tidal volume*). El cuál es el volumen de gas que entra y sale de los pulmones en una respiración basal [24].
2. Volumen de reserva inspiratorio (IRV o *Inspiratory reserve volume*). Representa el volumen adicional de gas que puede introducirse en los pulmones al realizar una inspiración máxima desde volumen corriente [24].

3. Volumen de reserva espiratorio (ERV o *Expiratory reserve volume*). Es el volumen de gas adicional que puede exhalarse del pulmón tras espirar a volumen corriente [24].

4. Volumen residual (RV o *Residual volume*). Corresponde al volumen de gas que permanece dentro del pulmón tras una espiración forzada máxima [24].

5. Capacidad vital (VC o *Vital capacity*). Es el máximo volumen de gas pulmonar movilizable, es decir, es la suma del volumen corriente y los volúmenes de reserva espiratoria y espiratoria. Dependiendo de la forma en que se considere, se diferencia capacidad vital inspiratoria (VC o *inspiratory vital capacity*), que es el volumen de gas que puede ser introducido en el pulmón con un esfuerzo inspiratorio máximo, tras una espiración máxima lenta. La capacidad vital lenta (SVC o *slow vital capacity*) es el volumen de gas que puede ser exhalado del pulmón con un esfuerzo espiratorio máximo lento, tras un esfuerzo inspiratorio máximo. Y la más utilizada, capacidad vital forzada (FVC o *forced vital capacity*), que supone el volumen de gas exhalado con un esfuerzo espiratorio máximo tras una inspiración máxima. En condiciones de normalidad, apenas existen diferencias entre las distintas modalidades de capacidad vital, pero sí puede haberlas cuando existe patología [24].

6. Capacidad inspiratoria (IC o *Inspiratory capacity*). Es la suma del volumen corriente y del volumen de reserva inspiratorio. Representa el máximo volumen inspirado tras una espiración tranquila [24].

7. Capacidad residual funcional (FRC o *Functional residual capacity*). Suma del volumen de reserva espiratorio y del volumen residual. Es el volumen de gas que hay dentro de los pulmones al final de una espiración tranquila y, como se mencionará después, corresponde al punto de equilibrio entre la retracción elástica del pulmón y de la caja torácica [25].

8. Capacidad pulmonar total (TLC o *Total lung capacity*). Abarca el volumen corriente, el volumen de reserva inspiratorio, el volumen de reserva espiratorio y el volumen residual. Es el máximo volumen de gas que pueden contener los pulmones [25].

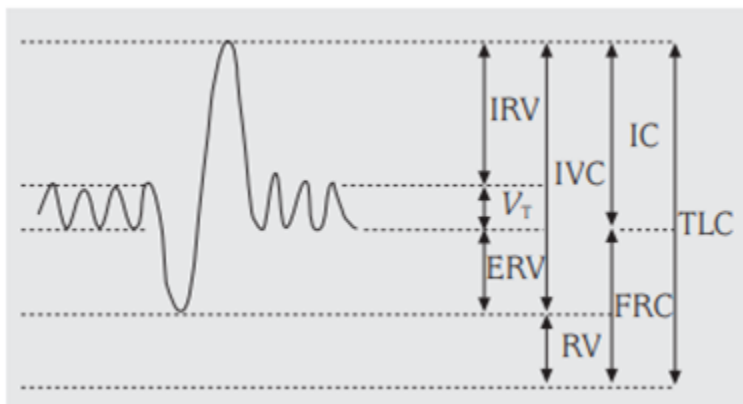


Figura 1.7: Volúmenes y capacidades sobre una curva espirométrica volumen/tiempo. VT: Volumen corriente; RV: Volumen residual; IC: Capacidad inspiratoria; FRC: Capacidad residual funcional; TLC: Capacidad pulmonar total [24].

1.5.2.5. Valores estándar de espirometría

El espirómetro debe acumular volúmenes de 8 L durante un mínimo de 15 s con una precisión de al menos 3% de lectura o 0.05 L, lo que sea mayor, con flujos entre 0 y 14 L. La resistencia total al flujo de aire a 14 L debe ser menor a 1.5 cmH₂O*L y debe medirse en cualquier tubo, válvula, prefiltro, etc. Incluidos los que se pueden insertar entre el sujeto y el espirómetro. Algunos dispositivos pueden presentar cambios en la resistencia debido al agua generada por la condensación de vapor, y se deben cumplir los requisitos de precisión bajo condiciones de temperatura corporal y presión ambiental saturado (BTPS por sus siglas en inglés) de hasta ocho maniobras FVC sucesivas realizadas en un período de 10 min sin inspiración del instrumento [25].

1.5.2.6. Normatividad comercial

El INVIMA (Instituto Nacional de Vigilancia de Medicamentos y Alimentos) es la entidad encargada de evaluar los equipos médicos que circulan en el comercio colombiano. Para cumplir las condiciones que aseguran la calidad del equipo y la seguridad de los pacientes, dicha entidad establece una guía para determinar la clasificación del dispositivo y así mismo su nivel de riesgo [26]. Tanto el riesgo como los requisitos que deben cumplir los fabricantes de dispositivos médicos se rigen por la Directiva 93/42 CEE, la cual cuenta con doce anexos técnicos que se describen a continuación [26]:

- **Anexo I Requisitos esenciales:** Establece requisitos de diseño y fabricación del dispositivo médico, las propiedades físicas, químicas y biológicas, los requisitos para eliminar y disminuir la infección y contaminación microbiana; así como las propiedades concernientes a la fabricación y el medio ambiente. Hace referencia a los equipos con función de medición, cuyos requisitos exigen constancia y precisión; y los requisitos para la protección contra la radiación eléctrica e ionizante.

- **Anexo II Declaración CE de conformidad:** Sistema completo de garantía de calidad. Es la aplicación del sistema de calidad aprobado para el diseño, fabricación y el control final de los productos sanitarios.
- **Anexo III Examen CE de tipo:** El organismo notificador comprueba y certifica una muestra representativa del producto, la cual se considera cumple con las disposiciones de la directiva.
- **Anexo IV Verificación CE:** El organismo notificador verifica si los productos cumplen con los requisitos certificados en el examen.
- **Anexo V Declaración CE de Conformidad:** Establece la garantía de la calidad en la producción. Anexo VI Declaración CE de Conformidad. Establece la garantía de la calidad del producto.
- **Anexo VII Declaración CE de Conformidad:** Establece el cumplimiento de los requisitos y obligaciones de productos estériles.
- **Anexo VIII Declaración relativa a los productos que tengan una finalidad especial:** Establece el cumplimiento de los requisitos y obligaciones de productos con finalidad especial, por ejemplo, los utilizados para investigación.
- **Anexo IX Criterios de la clasificación por riesgo:** Este apartado será descrito específicamente en la sección de anexos.
- **Anexo X Evaluación clínica:** Cuenta con disposiciones generales para el cumplimiento de los requisitos relacionados a las características y desempeño contemplados en condiciones normales de utilización de un producto, así como la evaluación de los efectos secundarios y de la aceptabilidad de la relación beneficio riesgo, basada en datos clínicos.
- **Anexo XI criterios mínimos que deben observarse para la designación de los organismos notificados:** Es la evaluación y verificación que realizan los organismos notificadores y la agencia sanitaria sobre el diseño, fabricación, comercialización y mantenimiento de los productos.
- **Anexo XII Marchamo CE de Conformidad:** Es la marca CE.

Debido a que en este caso la estación de calibración no entra en contacto directo con algún paciente, si no que se usa para realizar la validación de los inspirómetros, debe de estar regulada y estandarizada por medio de la dirección ISO/IEC 17025.

El estándar ISO/IEC 17025 le permite a los laboratorios que estén operados por un gobierno, industria, organización o universidad, demostrar que operan competentemente y generan resultados validos y confiables en sus pruebas o calibraciones [27]. Dentro de los principales objetivos de este estándar se encuentran:

- Establecer una aceptación internacional de resultados para ensayos y/o calibraciones, incluyendo muestreo. Esto permitirá más cooperación entre agencias e instituciones internacionales y reduce la necesidad de hacer realizar las mismas pruebas para ahorrar tiempo y recursos. [27]
- Facilitar la interpretación y la aplicación de los requisitos, evitando, al máximo posible, opiniones contrarias y conflictivas. Esto ayuda a reducir la necesidad de documentos explicativos adicionales para las pruebas. [27]

Este estándar cuenta con 5 bloques principales:

- **Requerimientos generales:** esta sección cubre la imparcialidad y la confidencialidad, los cuales son dos requerimientos vitales para mantener la confianza entre los usuarios y los laboratorios que realizan las calibraciones. La imparcialidad que implica que el laboratorio no va a permitir presiones comerciales, financieras o de cualquier tipo para comprometer la calidad de los resultados y la confidencialidad que requiere que el laboratorio mantenga los resultados de las pruebas privadas. [28]
- **Requerimientos estructurales:** esta sección define los componentes organizativos básicos de un laboratorio. Establece que un laboratorio acreditado debe ser una entidad legal o parte de una entidad legal, que es responsable por sus actividades de prueba y calibración. Además de esto, define los requerimientos básicos para el personal, la autoridad dada a estos y los recursos necesarios para que estos lleven a cabo sus responsabilidades. [28]
- **Requerimientos de recursos:** hay seis cláusulas que abordan el requisito de que el laboratorio tenga disponible el personal, las instalaciones, los equipos, los sistemas y los servicios de apoyo necesarios para realizar sus actividades de laboratorio. [28]
- **Requerimientos del proceso:** esta sección cubre 11 procesos fundamentales para mejorar la eficiencia. Dentro de los más destacados se encuentran: revisión de solicitudes, licitaciones y contratos, la selección, verificación y validación de métodos, presentación de informes y resultados, entre otros. [28]
- **Requerimientos de los sistemas de gestión:** esta sección cubre ocho actividades, incluida la documentación del SGC, como políticas y objetivos, control de documentación y registros, tratamiento de riesgos y oportunidades, mejoras y acciones correctivas y finaliza con la auditoría interna y la revisión de la gestión. Una vez más, esta norma sólo brinda los requisitos generales, para unos más detallados se debe consultar la norma ISO 19011. [28]

Diseño del Software y Hardware

2.1. Diseño del Sistema Hardware

2.1.1. Descripción General

En este capítulo se hará un enfoque en el aspecto del diseño mecánico del sistema tecnológico planteado, los distintos componentes seleccionados para cumplir con los requerimientos en todos los ámbitos y cómo es la interacción directa entre el usuario y el sistema por medio de una interfaz creada en un computador para cumplir con el tercer objetivo específico planteado en el proyecto.

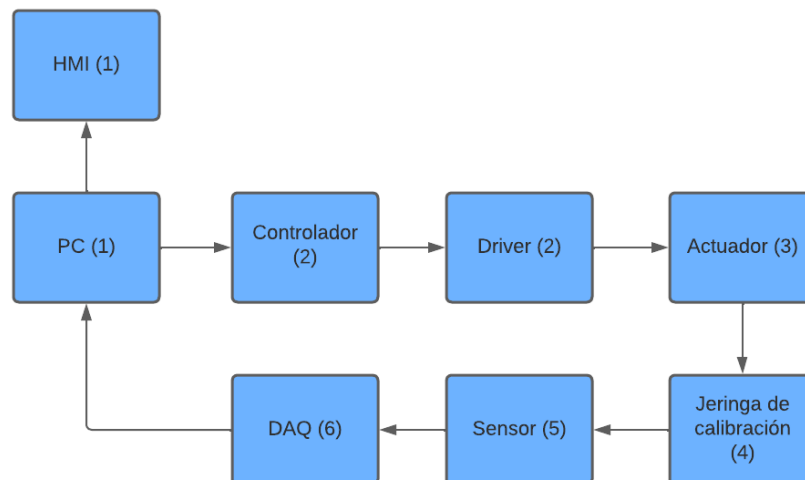


Figura 2.1: Diagrama de bloques para funcionamiento del hardware.

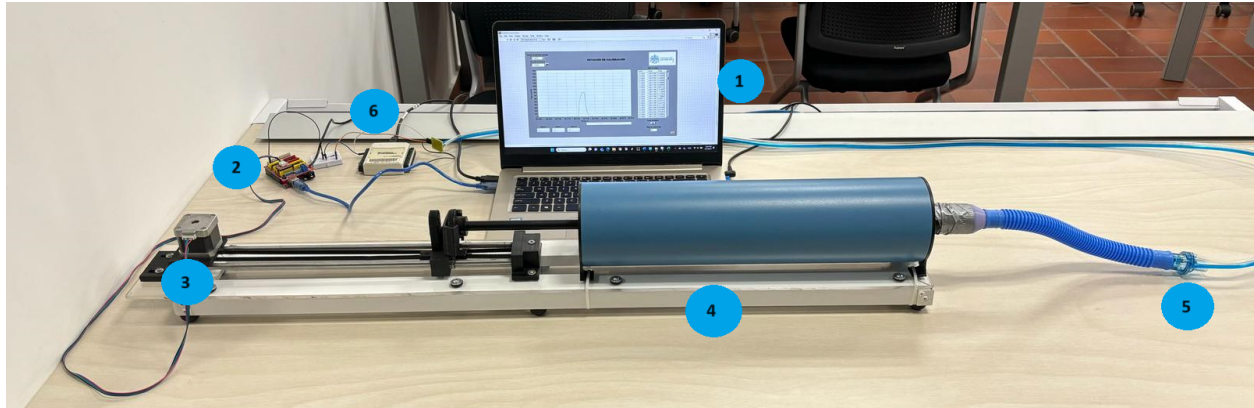


Figura 2.2: Montaje del sistema.

Durante la metodología, se definieron una serie de parámetros de diseño, entre estos se encontraban que:

- El sistema debe de alcanzar un flujo mínimo de $300 \text{ cm}^3/\text{s}$ y uno máximo de $1000 \text{ cm}^3/\text{s}$.
- Debe de mostrar los datos medidos en una gráfica de flujo contra tiempo.
- Debe de generar un formulario con una tabla donde se puedan visualizar los datos tomados.
- Debe tener una interfaz donde se pueda controlar el sistema y ver los datos tomados por el sensor.
- Debe de contar con tres rutinas diferentes que generen diferentes tipos de gráficas; debe ser económico, portable y fácil de usar.
- El motor debe alcanzar una velocidad mínima de 2000 pasos/segundo y una máxima de 18000 pasos/segundos.
- El actuador debe desplazarse una distancia no mayor a 29 cm para evitar daños en la jeringa o el mecanismo.
- Al sistema se le debe realizar un mantenimiento cada 100 horas de uso. Para este mantenimiento se debe lubricar los ejes de acero con aceite lubricante 3 en 1.

Con base en estos requerimientos, la estación de calibración se diseña para que, por medio de una jeringa de calibración (4) suministrada por la Pontificia Universidad Javeriana Cali, entregue un suministro de aire, el cual puede ser usado para el proceso de inspiración y espiración. El embolo de esta jeringa debe desplazarse de manera automática con ayuda de un actuador lineal (3) que se moverá con ayuda de un motor, un controlador (2) y su respectivo driver (2), además, a la salida de la jeringa debe de haber un sensor (5) que tome como medida final el flujo de aire que va a pasar

por el sistema validador y el dispositivo a validar. Esto se puede observar mejor en las figuras 2.1 y 2.2.

La estación de calibración tiene como objetivo medir el flujo que esté pasando a la salida de la jeringa, la cual se va a mover según la rutina seleccionada. Este flujo va a ser medido por un sensor como se mencionó anteriormente y esta información del sensor va a ser tomada por un adquisidor de datos (6) que enviará la información recolectada a la interfaz (1) para que sea graficada. Una vez se obtenga la gráfica, esta podrá ser analizada por los expertos para determinar diferentes factores.

2.1.2. Diagrama de flujo del funcionamiento del HMI.

A continuación, se presenta el diagrama de flujo que representa de una manera dinámica el funcionamiento general de la interfaz para posteriormente mostrar una breve explicación de cada bloque:

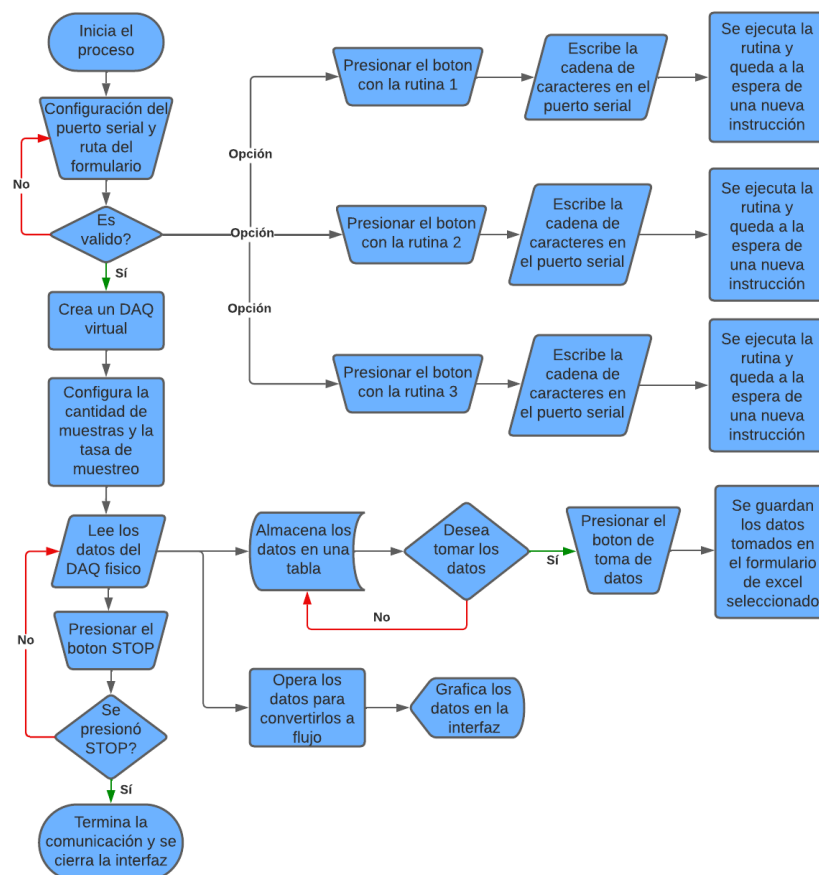


Figura 2.3: Diagrama de flujo para funcionamiento de LabView.

- **Inicia el proceso:** este bloque consiste en el encendido de la interfaz. En este momento se muestra la gráfica y las opciones para seleccionar y un botón de STOP para detener la comunicación serial y por lo tanto la interfaz.
- **Configuración del puerto serial y ruta del formulario:** por medio de una caja desplegable se selecciona el puerto serial con el que se vaya a trabajar, en este caso sería el puerto donde esté conectado el Arduino. En otra caja desplegable, se debe seleccionar la ruta de la plantilla del formulario que se usará para almacenar los datos tomados por el sensor.
- **Decisión de validez:** este bloque tiene varias opciones, en caso de que el puerto seleccionado no sea válido, la interfaz no va a realizar ninguna acción y se quedará esperando a que se seleccione un puerto válido. Si por el contrario es válido, el programa internamente se prepara para la lectura y escritura de datos.
- **Creación de DAQ virtual:** si el puerto es válido el programa creará una DAQ virtual la cual se configura con un número de muestras y una tasa de muestreo definida.
- **Lectura de datos:** el programa lee y opera los datos que recibe del sensor por medio de la DAQ física.
- **Gráfica de datos:** este bloque representa la gráfica que se encuentra en la interfaz para mostrar los datos que ha leído la DAQ y que el programa ha operado anteriormente, esta gráfica posee los datos de flujo [cm^3/s] con respecto al tiempo transcurrido de lectura [s].
- **Almacenamiento de datos:** este bloque representa el almacenamiento temporal de los datos en una tabla ubicada en la interfaz para que se pueda observar con exactitud el dato tomado.
- **Botón "Toma de datos":** este botón permite almacenar los datos guardados en la tabla en una plantilla de Excel definida previamente por el usuario.
- **Botón "STOP":** al presionar este botón la comunicación serial se cerrará y la interfaz se apagará. Si no se presiona el DAQ seguirá leyendo datos.
- **Botón "Rutina 1":** si se presiona este botón el sistema enviará una cadena de caracteres a Arduino lo que ayudará a identificar que se seleccionó la rutina 1. Luego de esto se ejecuta la rutina y una vez terminada quedará a espera de otra selección.
- **Botón "Rutina 2":** si se presiona este botón el sistema enviará una cadena de caracteres a Arduino, lo que ayudará a identificar que se seleccionó la rutina 2. Luego de esto se ejecuta la rutina y una vez terminada quedará a espera de otra selección.
- **Botón "Rutina 3":** si se presiona este botón el sistema enviará una cadena de caracteres a Arduino lo que ayudará a identificar que se seleccionó la rutina 3. Luego de esto se ejecuta la rutina y una vez terminada quedará a espera de otra selección.

2.1.3. Diseño mecánico.

En esta sección se hablará de los bloques del actuador y la jeringa, teniendo en cuenta el diagrama de bloques de la figura 2.1. Uno de los componentes principales para este proyecto es lograr emular la respiración de una persona. Debido a que el proyecto tenía que considerar limitaciones de presupuesto, se decidió hacer uso de una jeringa de calibración de 3L que proporcionó la mencionada Universidad.

Las jeringas de calibración son usadas para comprobar y calibrar la medición de flujo del dispositivo de espirometría de manera manual. Como el objetivo del proyecto es realizar esta acción de manera automática, se necesitó de un sistema que mueva el embolo de la jeringa sin necesidad de tener a una persona moviéndola. Además de esto, se usó una jeringa de 3L debido a que es el estándar que se usa para las calibraciones a pesar de que la capacidad pulmonar total sea de aproximadamente 6L [29].

Durante la etapa de diseño se plantearon varias alternativas y sistemas para lograr este movimiento automático. Para la selección final se usó un método de 4 pasos: el primero se centra en definir el problema, en el segundo se realizó una lluvia de ideas, en el tercero se descartan aquellas que no cumplan los requerimientos y en el cuarto se da la solución al problema teniendo en cuenta las ideas no descartadas y las descartadas [30].

Siguiendo estos pasos, se planteó el principal problema del proyecto, el cual era lograr mover de manera automática la jeringa de calibración de 3L o 29 cm en mínimo 3 segundos. Este tiempo se definió debido a que en las espiraciones, por lo general, se alcanza el volumen máximo en un tiempo de aproximadamente 3 o 4 segundos [31]. Con esto definido, en el segundo paso se encontró que para lograr esta velocidad en este tiempo se necesitaban de sistemas que tuvieran la fuerza suficiente para mover el embolo de la jeringa, pocas perdidas mecánicas para lograr un movimiento fluido y motores que fueran capaces de alcanzar estas especificaciones. Estos requerimientos hicieron que la mayoría de ideas se pasaran del presupuesto disponible para el proyecto.

Para el paso 3 se contempló la idea de usar un actuador lineal que ayudara a mover el embolo de la jeringa. Estos dispositivos convierten el movimiento rotatorio de un motor en desplazamiento lineal y son usados mayormente para mover pequeñas cargas de manera continua. Otra opción que se consideró fue la de crear un sistema basado en un tornillo sin fin conectado a un motor eléctrico que fuera capaz de mover el embolo de la jeringa mediante un acople que permitiera este movimiento lineal.

Se planteó también usar un motor paso a paso con un sistema tipo tijera el cual se acoplará a la jeringa para convertir este movimiento rotatorio del motor en un movimiento lineal. Por último, se planteó no usar la jeringa y reemplazarla por un sistema con una válvula proporcional conectado a un suministro de oxígeno junto con un regulador de presión y mediante la retroalimentación del sensor la válvula se abriría en un determinado porcentaje, dependiendo del flujo que se necesite.

Finalmente, en el paso 4 se descartaron todas las opciones debido a que en la mayoría, los motores que se necesitaban para recorrer esa distancia en ese tiempo eran muy costosos o los sistemas mecánicos no permitían tener un control de la distancia que se había recorrido. Sin embargo, se tomó la idea del motor paso a paso porque tenía la fuerza y la velocidad que se necesitaba para mover el embolo y porque el costo era menor, también se podía variar la velocidad y tener un control de los pasos que había dado el motor para así saber cuánto había recorrido el sistema.

Como se puede apreciar en la figura 2.7, respecto al sistema se escogió un mecanismo tipo cremallera o correa dentada (8) debido a su precio y la facilidad que conlleva implementar este tipo de correas, esta a su vez va conectada a dos poleas dentadas, como se puede ver en la figura 2.6 que se moverán cada que el motor dé un paso, lo cual convertirá el movimiento rotatorio del motor a movimiento lineal.

Estas poleas estarán dentro de un bloque (3) al inicio y a final del recorrido, estos bloques a su vez están conectados con dos ejes de acero (4), sobre los que se moverá el acople (6) que se puede ver en la figura 2.5 el cual a su vez irá conectado con la correa dentada. Este acople posee un agarre (2) de teflón como se ve en la figura 2.4 el cual servirá para conectar el embolo de la jeringa con el sistema. Este mecanismo conectado con el motor paso a paso (7) y la jeringa dará como resultado el sistema mecánico como se puede ver en la figura 2.8. En la sección de anexos se podrán ver los otros diseños que se plantearon y se mencionaron anteriormente.

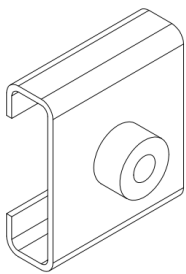


Figura 2.4: Agarre para el émbolo de la jeringa.

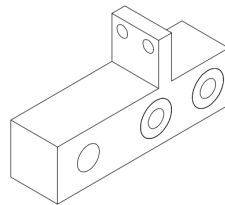


Figura 2.5: Acople con el sistema.

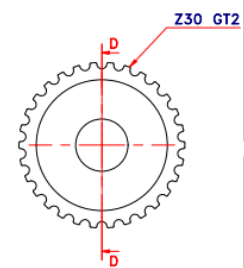


Figura 2.6: Polea dentada.

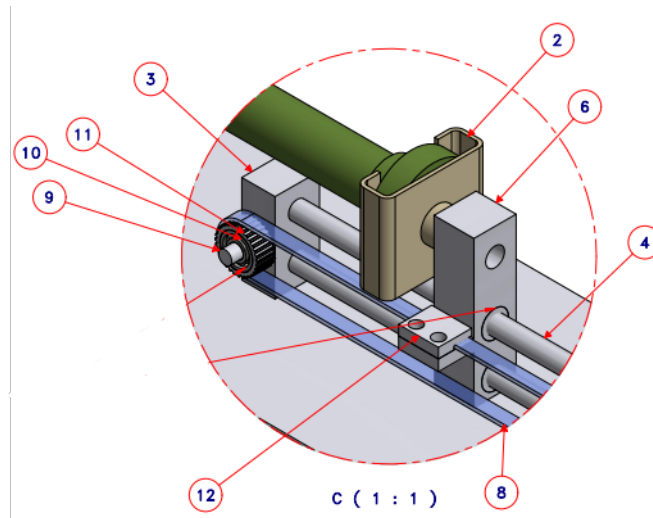


Figura 2.7: Diseño del agarre.

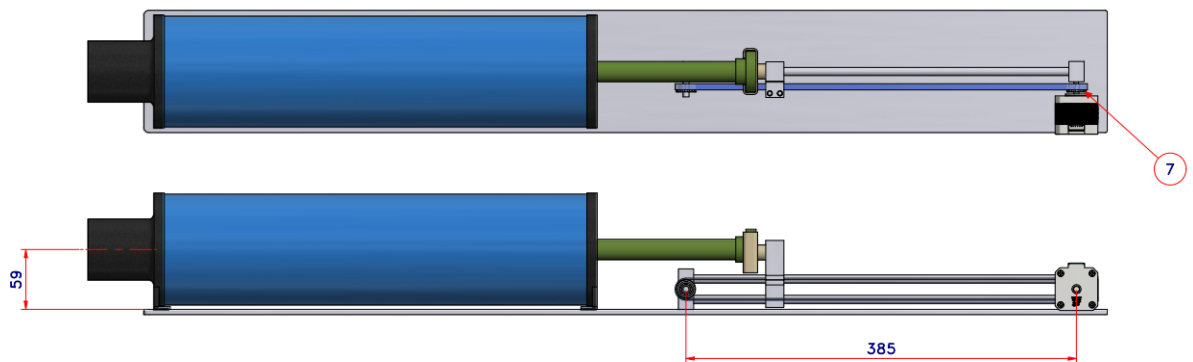


Figura 2.8: Diseño de mecanismo con correa dentada.

2.1.4. Selección de los componentes electrónicos

Para la implementación del sistema y siguiendo la metodología planteada, se seleccionaron componentes de referencias específicas los cuales pasaron por unos criterios que fueron útiles a la hora de hacer mejor selección de estos componentes. Inicialmente se hizo una investigación acerca de trabajos relacionados para posteriormente seleccionar tres referencias, las cuales son evaluadas con los criterios definidos de la siguiente manera:

2.1.4.1. Criterios de selección

Se investigan trabajos relacionados y se seleccionan varios tipos de componentes para la implementación de la estación, donde posteriormente se hará una selección de cada uno de los componentes

del sistema y se explicará el porqué de su selección basados en los siguientes criterios:

- **Costos de implementación:** una de las principales razones de la implementación de este criterio es debido al presupuesto que se manejó para el prototipo lo cual limita la selección de componentes más caros, siempre y cuando tengan una alternativa más económica que cumpla con los demás requerimientos.
- **Facilidad de acople:** uno de los parámetros más importantes a la hora de realizar un proyecto de ingeniería es que los componentes seleccionados sean compatibles entre sí, o que por lo menos exista una forma viable de acoplarlos entre sí.
- **Rangos de operación:** las variables a medir y los requerimientos definidos cuentan con unos rangos específicos que necesitan ser cumplidos para que de esta manera se llegué a un resultado óptimo.
- **Precisión:** se busca implementar un sistema que pueda entregar información confiable y precisa al usuario acerca de las variables a medir, ya que los equipos médicos deben de cumplir con esta condición.
- **Facilidad de comunicación:** debido a que el dispositivo central del sistema es un computador, los componentes deben de tener una comunicación sencilla entre ellos y el computador.

2.1.5. Componentes seleccionados

Teniendo en cuenta las características de los componentes investigados, se hace la selección de los componentes siguiendo la ponderación de los criterios, en la cual se comparan los criterios frente a los otros para así encontrar el peso de cada criterio, lo que ayudará a hacer una mejor selección de los componentes para la etapa de implementación.

Criterios	1 vs 2	1 vs 3	1 vs 4	1 vs 5	2 vs 3	2 vs 4	2 vs 5	3 vs 4	3 vs 5	4 vs 5	Porcentaje
1. Costos de implementación.	9	6	4	7							26%
2. Facilidad de acople.	1				4	3	4				12%
3. Rangos de operación.		4			6			6	6		22%
4. Precisión.			6			7		4		6	23%
5. Facilidad de comunicación.				3			6		4	4	17%
Total	10	10	10	10	10	10	10	10	10	10	100%

Figura 2.9: Ponderación de criterios.

Como se puede ver en la figura 2.9, el criterio que más peso tiene al momento de elegir un componente es el del costo de implementación con un porcentaje del 26% siguiendo el método mencionado anteriormente. Sin embargo, el segundo criterio que más peso tiene es el de la precisión

con un peso del 23%. Esto quiere decir que para la selección se debía escoger la opción más viable económicamente sin dejar a un lado la precisión que tenga el componente.

2.1.5.1. Actuador

Actuador		Costos de implementación	Facilidad de acople	Rangos de operación	Precisión	Facilidad de comunicación	Total
		<i>Pesos de los criterios:</i>					
		26%	12%	22%	23%	17%	
Actuador lineal	Nota	0.5	4.5	3	4	5	3.1
	Puntaje	0.13	0.54	0.66	0.92	0.85	
Motor eléctrico	Nota	4	2	4	3	5	3.7
	Puntaje	1.04	0.24	0.88	0.69	0.85	
Motor paso a paso	Nota	4.5	4	4	4	5	4.3
	Puntaje	1.17	0.48	0.88	0.92	0.85	

Figura 2.10: Actuador seleccionado

El actuador seleccionado para la implementación, como se mencionó anteriormente es el motor paso a paso. Esto debido a que este tipo de motores son más económicos, tienen una capacidad de carga considerable para lo que se requiere en este proyecto, y se puede tener un control de la cantidad de pasos que han recorrido, para de esta forma delimitar el recorrido del mecanismo y que de esta forma no dañe los bloques en los finales del recorrido, el acople o la jeringa.

2.1.5.2. Motor paso a paso

Motor paso a paso		Costos de implementación	Facilidad de acople	Rangos de operación	Precisión	Facilidad de comunicación	Total
		<i>Pesos de los criterios:</i>					
		26%	12%	22%	23%	17%	
28BYJ-48	Nota	4.5	4	1.5	4	5	3.75
	Puntaje	1.17	0.48	0.33	0.92	0.85	
NEMA 17	Nota	4	4	4	4	5	4.17
	Puntaje	1.04	0.48	0.88	0.92	0.85	
NEMA 23	Nota	1.5	4	4.5	4	5	3.63
	Puntaje	0.39	0.48	0.99	0.92	0.85	

Figura 2.11: Motor paso a paso seleccionado.

El motor seleccionado es el NEMA 17, ya que aparte de tener una puntuación muy por encima de los otros motores, este motor tiene un torque de 4 kg/cm lo que hace que sea bueno para aplicaciones que necesitan mover cargas considerables con buena exactitud, además de esto tiene una velocidad máxima recomendada de 1000 RPM y por lo tanto una potencia máxima de 40 Watts lo que hace que sea usado para otras aplicaciones como impresoras 3D [32].

2.1.5.3. Driver

Driver		Costos de implementación	Facilidad de acople	Rangos de operación	Precisión	Facilidad de comunicación	Total
		<i>Pesos de los criterios:</i>					
		26%	12%	22%	23%	17%	
DRV8825	Nota	4	5	4.8	5	5	4.696
	Puntaje	1.04	0.6	1.056	1.15	0.85	
A4988	Nota	5	5	4.6	5	5	4.912
	Puntaje	1.3	0.6	1.012	1.15	0.85	
L298N	Nota	4.5	5	5	5	5	4.87
	Puntaje	1.17	0.6	1.1	1.15	0.85	

Figura 2.12: Driver seleccionado.

A pesar de que las tres opciones son muy similares y hacen casi lo mismo, se seleccionó el driver A4988 debido a que, a diferencia de los otros, este driver requiere una duración mínima del pulso de 1 us, lo que hace que se puedan enviar pulsos más rápidos, lo que significa una respuesta más rápida del motor. Además de esto, el driver puede bajar hasta los 1/16 micro-pasos, entre más altos sean los micro-pasos, se tendrá una operación más suave y silenciosa [33]. Debido a que el motor que se seleccionó puede recibir una corriente máxima de 1.5 A, se necesita un driver que sea capaz de entregarle esta cantidad, en este caso, este driver es capaz de entregar hasta 2 A de corriente, lo que lo convierte en una muy buena opción.

2.1.5.4. Microcontrolador

Microcontrolador		Costos de implementación	Facilidad de acople	Rangos de operación	Precisión	Facilidad de comunicación	Total
		<i>Pesos de los criterios:</i>					
		26%	12%	22%	23%	17%	
Arduino UNO	Nota	4.5	4	4.5	4	5	4.41
	Puntaje	1.17	0.48	0.99	0.92	0.85	
ESP 32	Nota	4	4	4	4	5	4.17
	Puntaje	1.04	0.48	0.88	0.92	0.85	
STM32F103C8T6	Nota	5	4	3.5	4	5	4.32
	Puntaje	1.3	0.48	0.77	0.92	0.85	

Figura 2.13: Microcontrolador seleccionado.

El microcontrolador seleccionado es el Arduino UNO. Esta tarjeta está basada en la ATmega328P, cuenta con 14 pines de entradas y salidas digitales de las cuales 6 pueden ser usadas como salidas PWM y 6 entradas análogas. Tiene una SRAM de 2 KB y alcanza una velocidad de reloj de 16 MHz [34]. Debido a que solo se controlará el motor desde el microcontrolador, estas especificaciones son suficientes para un correcto funcionamiento del sistema, además de esto, es el que más facilidad tiene para acoplar el driver seleccionado, aparte de esto puede dar una alimentación de 5V o 3.3V, lo cual es útil en caso de que se quiera alimentar el sensor u otra clase de circuito en alguna mejora futura. Además, esta placa de Arduino se puede comunicar por puerto serial con labview, aparte de que existen librerías y documentación acerca de la comunicación entre estos dos programas.

2.1.5.5. Sensor

Sensor		Costos de implementación	Facilidad de acople	Rangos de operación	Precisión	Facilidad de comunicación	Total
		<i>Pesos de los criterios:</i>					
		26%	12%	22%	23%	17%	
XGZP6897A	Nota	5	4	4	4	5	4.43
	Puntaje	1.3	0.48	0.88	0.92	0.85	
HAFUHM0100L4AXT	Nota	3.8	4	4.5	4.5	4	4.173
	Puntaje	0.988	0.48	0.99	1.035	0.68	
SFM3200-AW	Nota	4	5	4.5	3.8	5	4.354
	Puntaje	1.04	0.6	0.99	0.874	0.85	

Figura 2.14: Sensor seleccionado.

El sensor seleccionado es el XGZP6897A de la marca CFSensor, es un sensor de presión diferencial usado para medir valores positivos o al vacío, además de esto es especial para medir la presión de gases no corrosivos o aire como se requiere en este caso. El sensor cuenta con una salida amplificada de 0 a 5 V y una alimentación de 5V, por lo que puede ser alimentado con la tarjeta de Arduino y, por lo tanto, no hay necesidad de fuentes externas a esta. Posee un rango de medición de -1000 kPa a -0.5 kPa o de 0.5 kPa a 200 kPa. Este sensor es ampliamente usado en dispositivos médicos como ventiladores, equipos para tratar la apnea de sueño, analizadores de orina y dispositivos de masajes médicos. El sensor está totalmente calibrado y compensado en temperatura para el desplazamiento, la sensibilidad, la temperatura y la no linealidad [35].

2.1.5.6. Módulo de adquisición de datos

DAQ		Costos de implementación	Facilidad de acople	Rangos de operación	Precisión	Facilidad de comunicación	Total
		<i>Pesos de los criterios:</i>					
		26%	12%	22%	23%	17%	
MyDAQ	Nota	4	5	1.5	4	5	3.74
	Puntaje	1.04	0.6	0.33	0.92	0.85	
USB-6009	Nota	3.5	5	4.5	3.5	5	4.155
	Puntaje	0.91	0.6	0.99	0.805	0.85	
USB-6210	Nota	2	5	4.5	4.5	5	3.995
	Puntaje	0.52	0.6	0.99	1.035	0.85	

Figura 2.15: Tarjeta DAQ seleccionada.

Por último, la tarjeta de adquisición de datos seleccionada es la DAQ USB-6009. Este módulo posee 7 pines de conexión a tierra, 8 pines de entradas analógicas, 2 pines de salidas analógicas, un pin con 2.5 V de referencia, un pin con 5 V que se puede usar como fuente y 12 pines para configurarlos como entradas o salidas digitales. Este módulo DAQ cuenta también con una velocidad de muestreo de 48KS/s y una resolución de 12 bits [36] y se seleccionó debido a estos criterios. Además, aparte de ser de un precio intermedio, esta tarjeta cuenta con terminales con configuración RSE, lo que quiere decir que la terminal mide la diferencia potencial entre la entrada analógica y la tierra analógica, algo importante ya que la salida del circuito del sensor es de este tipo.

2.2. Diseño del sistema software

2.2.1. Descripción General

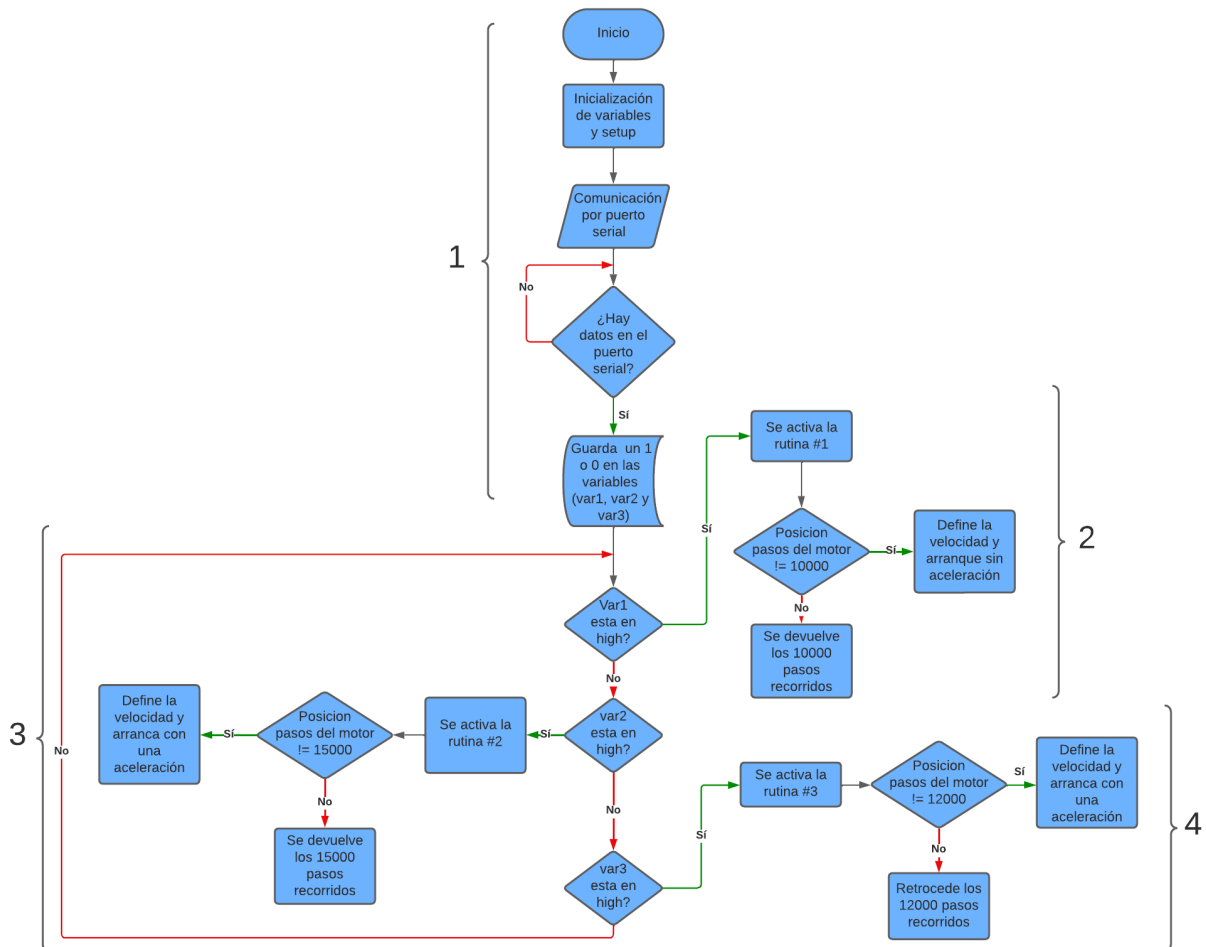


Figura 2.16: Diagrama de Flujo para funcionamiento del control.

Durante la etapa de diseño de la estación de calibración, se definió que se necesitaría de un sistema de control de lazo abierto, desde el cual se podrá seleccionar la rutina lo que permitirá que el motor se mueva de acuerdo a la rutina que se haya seleccionado. Por otro lado, se espera que el entorno de programación sea fácil de manejar, tenga comunicación con otro software para la interfaz y que sea económicamente accesible para poder trabajar con todas las opciones que cuente el controlador y su entorno de diseño. Lo mencionado anteriormente se puede observar de una forma más detallada en la figura 2.16 que muestra el diagrama de flujo del funcionamiento del software

para el controlador.

En la parte 1 del diagrama se encuentra el proceso de inicialización de las variables y la configuración de estas. En este caso son 3 variables: var1, var2 y var3, las cuales se encargan de leer el estado de los botones en la interfaz, es decir, si reciben un 1 significa que se presionó ese botón en la interfaz y por lo tanto debe correr la instrucción seleccionada. Por otro lado, si recibe un 0 significa que no debe correr la rutina que esté siendo controlada por la variable. Luego de esta configuración de las variables, el programa hace una lectura del puerto serial, si no hay datos en el puerto, el programa seguirá leyendo hasta que le llegue un dato, una vez llegue este dato, se almacenará en su respectiva variable.

Para la parte 2 del diagrama, se tiene el caso en el que la variable var1 está en HIGH o tiene un 1 almacenado, esto significa que en la interfaz se seleccionó la rutina 1 por lo que el programa ejecuta la configuración de la rutina 1. Para esta rutina el programa revisa que la cantidad de pasos que ha dado el motor sea diferente a la requerida, en caso de que sea diferente, el programa seguirá revisando hasta que sea igual. Una vez se compruebe que el valor es igual, el motor recorre la misma cantidad de pasos en el sentido contrario siguiendo la misma lógica anteriormente mencionada.

En la parte 3 del diagrama se toma el caso en el que la variable var2 está en HIGH, lo que significa que se seleccionó la rutina 2, para esta rutina se sigue la misma lógica de la primera con la diferencia de que en esta el recorrido y la velocidad es diferente. Finalmente, en la parte 4 se tiene el caso en el que la variable var3 está en HIGH, lo que quiere decir que se seleccionó la rutina 3, para esta rutina el programa revisa si la cantidad de pasos del motor es diferente a la determinada, mientras no sean iguales el motor se moverá a una determinada velocidad con una aceleración configurada previamente, una vez sean iguales se devolverá la misma cantidad de pasos con la misma configuración de velocidad y aceleración.

Otro de los puntos esenciales en el diseño fue el de la interfaz. Por medio de esta, la persona encargada del uso de la estación de calibración debe seleccionar la rutina que desea probar en el inspirómetro por medio de botones que le permita a esta persona reconocer la rutina que ha seleccionado. Esta interfaz también se debe de comunicar con el control para mover el motor según la secuencia seleccionada, también debe recibir los datos que el sensor esté enviando y además de esto debe graficar estos datos para que se puedan visualizar en la interfaz.

Para el desarrollo tecnológico de la estación de calibración, se definieron los siguientes aspectos para la implementación, con base a los requerimientos, así:

- Control.
- Interfaz.
- Comunicación.

- Adquisición de datos.

Control: Se seleccionó el microcontrolador Arduino UNO con el IDE de Arduino ya que cuenta con comunicación serial, por la facilidad que brinda este software para el control del motor paso a paso según los requerimientos estipulados en la etapa de diseño y a que se puede programar en diferentes lenguajes de programación como Espressif IDE, JavaScript o LUA; para este proyecto se seleccionó el IDE de Arduino debido a las librerías que permiten la comunicación con LabView y un mejor control del motor paso a paso.

Dentro del IDE de Arduino se encontrarán las rutinas del motor las cuales moverán a diferentes velocidades el motor y por lo tanto la jeringa. Para lograr esto, se usará la librería de AccelStepper.h, esta librería permite a la tarjeta controlar diferentes tipos de motores paso a paso y, más importante aún, permite cambiarles su velocidad en pasos por segundo, permite tener un control de los pasos que ha recorrido que en este caso puede ayudar con la distancia que ha recorrido el embolo de la jeringa y la aceleración del motor.

Interfaz: El sistema de calibración tiene como componente central un computador, en este se visualizará la interfaz humano-maquina (HMI) la cual se va a realizar en el software LabView ya que gracias a las librerías que posee, se puede hacer lectura y escritura a Arduino por medio del puerto serial del computador. Además de esto, el software de LabView tiene la facilidad de comunicarse con las tarjetas de adquisición de datos por medio de puerto USB y cuenta con librerías y bloques predefinidos para el funcionamiento de estas. Desde esta interfaz se debe poder seleccionar las rutinas que estén disponibles y también se debe de poder visualizar por medio de gráficas desde esta interfaz los datos que este entregando el sensor a la salida de la jeringa, para que los expertos puedan revisar estas gráficas y determinar si el inspirómetro cumple con las condiciones para su uso o no.

Comunicación: : La comunicación se va a realizar por medio del puerto serial del computador ya que una de las opciones que tienen las placas de Arduino para comunicarse con un computador es por medio de este puerto, a través de esta comunicación se pueden cargar los programas realizados en el IDE de Arduino y se pueden leer o escribir los datos que estén disponibles en el puerto. Además de esto, existen varios métodos de comunicación por puerto serial entre LabView y Arduino, entre los que se encuentran las extensiones de VISA o LIFA de LabView, que permiten escribir o leer datos por medio del puerto serial. La principal diferencia entre estas dos extensiones es que VISA es una API que proporciona una interfaz de programación para controlar instrumentos Ethernet/LXI, GPIB, seriales, USB, PXI y VXI [37], esto permite que LabView lea o escriba datos de un programa independiente de Arduino por medio del puerto.

Agregando a lo anterior, LIFA es un toolkit que permite adquirir o enviar datos directamente de las entradas o salidas de las placas de Arduino y procesarlos en LabView, pero, con este fin, se debe de descargar en la tarjeta de Arduino una configuración predeterminada para lograr esta comunicación. Esto quiere decir que el programa de control se debe de hacer totalmente en LabView, lo que hace que la programación del motor por rutinas sea más complicada y no permite las opciones

que se pueden implementar en el IDE de Arduino, por lo que se decidió usar el API de VISA para poder separar el control en Arduino y la interfaz en LabView.

Adquisición de datos: Para la adquisición de datos se contemplaron dos opciones, la primera fue que el Arduino recibiera los datos, los operara dentro del programa y los enviara a LabView para que fueran graficados en la interfaz. La otra opción era por medio de un sistema DAQ que recibiera los datos y los enviara directamente a LabView. En este caso se eligió la segunda opción ya que por medio de la primera el puerto serial se encontrará ocupado escribiendo los estados de los botones, por lo que mientras se esté realizando la secuencia, no va a poder leer los datos del sensor y, por consiguiente, no va a poder graficar en tiempo real los datos que esté recibiendo el sensor.

Implementación del Software y Hardware

3.1. Implementación del Sistema Hardware

Teniendo en cuenta el diseño de hardware planteado anteriormente, la implementación se puede dividir en tres componentes esenciales. El primero es la implementación del sistema mecánico el cual es el encargado de mover el embolo de la jeringa, el segundo es la calibración del sensor de presión diferencial para que verifique que puede ser usado para aplicaciones de flujo y por ultimo las conexiones de todos los componentes seleccionados.

3.1.1. Sistema mecánico

Para la implementación del sistema mecánico, se imprimieron en 3D y de teflón, las piezas para el agarre del embolo, el acople del agarre con los ejes y los bloques de inicio y final de carrera como se puede ver en las figuras 2.4, 2.5 y 2.7 respectivamente. Posteriormente, se instaló el motor encima de uno de los bloques y debajo de este se acopló el eje del motor con la polea dentada. Por otro lado, la otra polea dentada se instaló dentro del otro bloque el cual se posicionó a una distancia de 32 cm del bloque del motor. Para fijar esta distancia se instalaron los ejes de acero a los cuales se había acoplado previamente el bloque central, el cual cuenta también con el agarre para el embolo de la jeringa.

Con estas distancias fijadas y todo unido se dio paso a instalar la correa alrededor de los ejes y a lo largo de todo el mecanismo. Todo esto se fijó con ayuda de tornillos a una placa de acrílico la cual a su vez se fijó a una base de aluminio. Luego de tener el mecanismo fijo se instaló la jeringa usando el agarre para el embolo, para sostener la jeringa se usaron unas amarras alrededor de las patas de la jeringa y la base de aluminio como se puede ver en la figura 3.1.

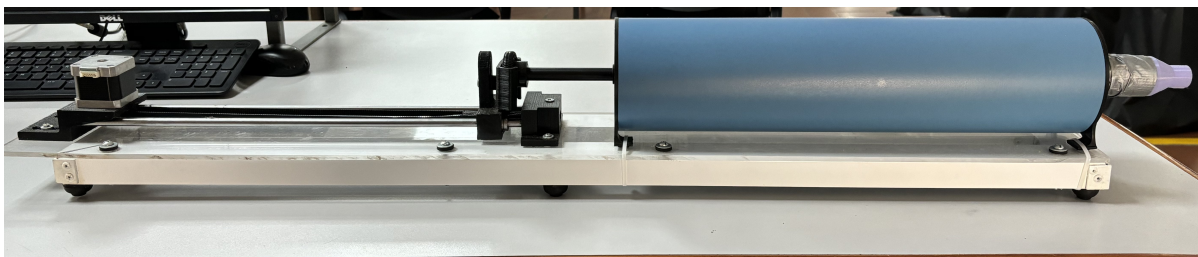


Figura 3.1: Sistema mecánico.

3.1.2. Calibración del Sensor

La calibración del sensor fue llevada a cabo de manera experimental en la Universidad Javeriana Cali, para esta calibración se hizo uso de la jeringa de calibración con las rutinas implementadas, un medidor de presión diferencial, el citrex H3 (especial para medir flujo y presión) y el sensor Hamilton. Se conecto el sistema de manera que los tubos del sensor Hamilton se conectaran al manómetro para medir la presión diferencial y el citrex se conectó justo después del Hamilton. Se realizó un barrido por todos los valores de flujo que podían arrojar las rutinas lo que dio como resultado los siguientes valores de flujo y presión:

Citrex [cm ³ /s]	Manometro [Pa]
307	40
340	45
389	51
423	56
456	60
496	65
534	70
585	79
590	79
648	89
656	90
723	100
726	101
820	116
826	117
844	120
1014	140
1025	140
1040	144
1286	178

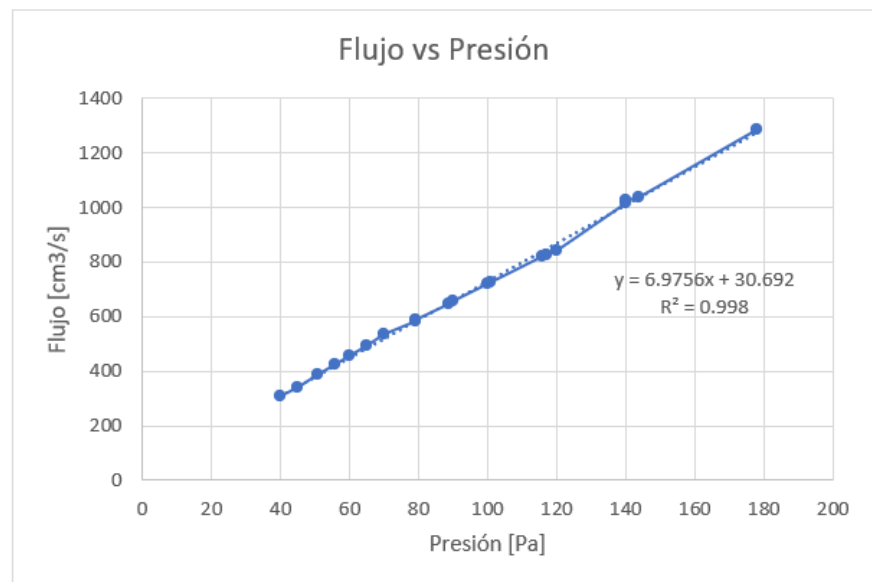


Figura 3.2: Datos Flujo vs Presión.

Figura 3.3: Gráfica de Flujo vs Presión.

Como se puede apreciar en las figuras 3.2 y 3.3, el sensor tiene un comportamiento mayormente lineal en cuanto a la relación del flujo y la presión. Para los valores de flujo bajos tiene un incremento lineal y a medida que este valor de flujo va aumentando, el incremento varía y se comporta parcialmente diferente. Por medio de una regresión lineal fue posible hallar la ecuación que relaciona estas dos variables como se puede ver en la ecuación (3.1).

$$Flujo = 6,9756 * (Presion) + 30,692 \quad (3.1)$$

3.1.3. Conexiones

Como se mencionó en las secciones anteriores, el computador es el dispositivo central del sistema debido a que, aparte de que se controla y se visualiza todo desde este, el microcontrolador y el módulo DAQ deben ir conectados a este por medio de los cables USB con los que cuentan los dos dispositivos. Para el microcontrolador se usó un shield cnc el cual es una extensión de la placa de Arduino UNO que permite una mayor facilidad al momento de conectar el driver A4988. Este shield se coloca sobre el Arduino y el driver se coloca sobre uno de los módulos para los ejes de los motores, luego de esto en los terminales de energía del shield (2 y 3) se le conecta un adaptador AC DC el cual puede ir conectado a una toma de corriente convencional para energizar el motor. Por último, el Arduino se conecta con el computador mediante el adaptador USB (1) que posee la tarjeta de Arduino, como se puede apreciar en la figura 3.4.

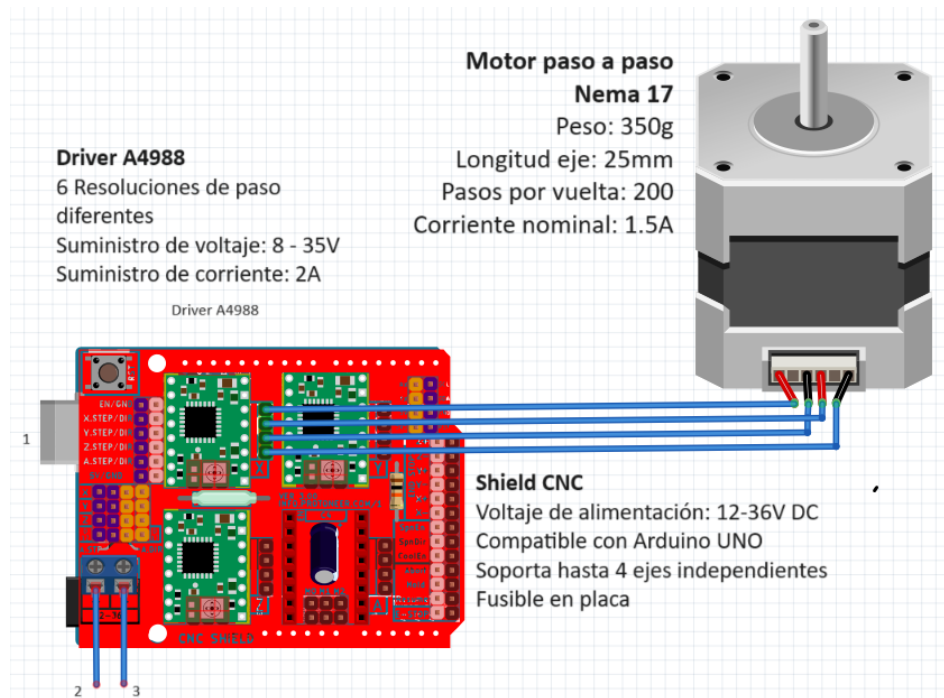


Figura 3.4: Conexión del motor paso a paso.

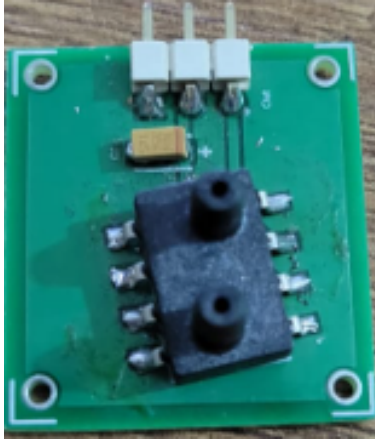


Figura 3.5: Placa del sensor de presión.

NI USB-6009

1	17	P00
2	18	P01
3	19	P02
4	20	P03
5	21	P04
6	22	P05
7	23	P06
8	24	P07
9	25	P10
10	26	P11
11	27	P12
12	28	P13
13	29	PF10
14	30	+2.5V
15	31	+5V
16	32	GND

GND	AI0/AI 0+
AI0/AI 0-	GND
AI1/AI 1+	AI5/AI 1-
GND	AI2/AI 2+
AI6/AI 2-	GND
AI3/AI 3+	AI7/AI 3-
GND	AO0
AO1	GND

Figura 3.6: Pines del DAQ NI USB-6009.

Una vez conectado el Arduino, se pasa a conectar el módulo DAQ. Para esta conexión se debe de tener en cuenta la placa del sensor la cual consta de el sensor de presión diferencial XGZP6891A junto con un condensador, esta placa tiene 3 pines como se puede observar en la figura 3.5. Estos pines de izquierda a derecha son los pines de tierra, alimentación y salida. Estos tres pines se conectan al módulo DAQ de la siguiente manera: El pin de tierra se conecta a la terminal 1 del módulo, el pin de alimentación se conecta al terminal 31 del módulo y el pin de salida se conecta al pin 2 del módulo, como se ve en la figura 3.6.

La placa con el sensor XGZP6891A necesita tomar la medida de la presión diferencial en el punto deseado que, en este caso, se encuentra a la salida de la jeringa. Para lograr esto se usó un sensor de flujo proximal de la marca Hamilton como se puede ver en la figura 3.7. Para acoplar este sensor al sistema se debe conectar la parte de mayor diámetro a la jeringa y los dos tubos que salen del sensor se deben conectar a la placa del sensor XGZP6891A, para esto se debe conectar el tubo azul en la entrada mas cercana a los pines de la placa y el tubo transparente en la siguiente entrada.



Figura 3.7: Sensor de flujo proximal [38].

Para lograr una gráfica mas precisa y sin tanto ruido producido por ser una señal de voltaje, se decidió implementar un filtro pasa baja con un condensador de 10nF y una resistencia de 1K Ω como se puede ver en la figura 3.8. La decisión de usar un condensador en vez de un inductor se tomó debido a que los condensadores son especiales para evitar cambios bruscos en el voltaje, lo cual es ideal para esta aplicación y así poder generar una gráfica más confiable. Debido a que se va a usar el filtro, la conexión entre los componentes varia. En este caso, el pin de output del sensor no iría directamente conectado al módulo DAQ si no que iría conectado al pin 1 de la resistencia y la salida de este circuito sería la que estaría conectada al pin 2 del módulo.

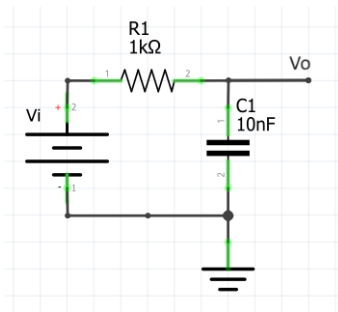


Figura 3.8: Filtro pasa bajas RC.

Para encontrar la frecuencia de corte del filtro, se usa la siguiente ecuación:

$$F = \frac{1}{2 * \pi * RC} \quad (3.2)$$

Reemplazando los datos de la ecuación (3.2) con una resistencia $R = 1K\Omega$ y una capacitancia $C = 10nF$ nos da como resultado una frecuencia de corte $F = 15,9KHz$. Lo que significa que todas las señales que estén por debajo de esta frecuencia pasarán atenuadas a la salida del sistema y por lo tanto a la entrada del módulo DAQ.

3.2. Implementación del Sistema Software

3.2.1. Control Arduino

3.2.1.1. Comunicación con LabView

Luego de realizar el diseño del sistema de control mencionada anteriormente, se dio paso a la implementación. Como se mencionó anteriormente, el control del motor se realizó con el IDE de Arduino y con ayuda de la extensión de VISA por el lado de labview para la comunicación serial. En este primer apartado se buscaba controlar el sistema desde la interfaz, es decir, desde la interfaz se buscaba seleccionar la rutina que movería el motor de determinada forma.

Para lograr esto, inicialmente se definen los pines digitales del Arduino los cuales enviaran un pulso para darle arranque al motor según la rutina que se seleccione, aparte de esto se crean las variables que van a recibir la información de los estados de estos botones, como en este caso solo se tienen tres rutinas, se crean solo tres variables. Luego, en la configuración del código se debe definir la tasa de comunicación del puerto serial. Como en este caso no se necesita de una comunicación crítica o que requiera una velocidad muy alta, se dejó en el estándar que se maneja que es de 9600 bits por segundo. Además de esto, se define cómo trabajarán los pines digitales seleccionados anteriormente, así que, para enviar el estado de las variables al motor mediante estos pines, se configuran como pines de salida, como se puede ver en la figura 3.9.

```
#include <AccelStepper.h>
#define x_paso 2
#define x_dire 5
#define x_habi 8
#define boton1 6
#define boton2 3
#define boton3 7
#define motorInterfaceType 1

int var1;
int var2;
int var3;
int contador = 0;

AccelStepper Motor1 = AccelStepper(motorInterfaceType, x_paso, x_dire);
void setup() {
  Serial.begin(9600);
  pinMode(x_paso, OUTPUT); pinMode(x_dire, OUTPUT); pinMode(x_habi, OUTPUT);
  pinMode(boton1, OUTPUT); pinMode(boton2, OUTPUT);
  pinMode(A0, INPUT);
  Motor1.setPinsInverted(true, true, true);
  Motor1.setMaxSpeed(18000);
}
```

Figura 3.9: Definición de variables y configuración.

Luego de realizada la configuración de las variables y los pines se realiza la lectura de los datos desde labview. Inicialmente con ayuda de la función `Serial.available()` se revisa si hay datos en el puerto serial, esto quiere decir que mientras haya datos, se realizará la lectura de lo que haya en el puerto serial. Si se cumple esta condición, se restringen las variables creadas anteriormente a que lean solo enteros con ayuda de la función `parseInt()`, esta función lo que hace es devolver el primer número entero valido en el buffer e ignorar lo demás en caso de que llegue algún símbolo o algo que no sea un número entero, si este es el caso, la función terminará.

Posteriormente a que se restringen las variables de esta forma se hace una lectura del puerto serial con la función `Serial.read()`, mientras la lectura que se hace del puerto serial sea una cadena de caracteres tipo `char` con longitud de 13, se escribirán digitalmente estos valores a las variables y a su vez a los pines definidos anteriormente, como se puede apreciar en la figura 3.10.

```
if (Serial.available()){
  var1 = Serial.parseInt();
  var2 = Serial.parseInt();
  var3 = Serial.parseInt();
  if(Serial.read() == char(13)){
    digitalWrite(boton1, var1);
    digitalWrite(boton2, var2);
    digitalWrite(boton3, var3);
  }
}
```

Figura 3.10: Función de lectura por puerto serial.

3.2.2. Movimiento del Motor

Para el movimiento del motor se usaron dos alternativas, una primera para pruebas del motor y del sistema mecánico y otra para implementar las rutinas deseadas. La primera forma se realizó sin librerías y con unos pulsadores físicos, interactuando solamente con los pines de paso, dirección y habilitación y con la frecuencia de los pulsos mediante retrasos.

Para lograr mover el motor de esta forma, inicialmente se debe de poner en LOW el pin de habilitación del driver para que de esta forma se pueda manejar el motor, luego de esto el programa revisa si el pulsador 1 está en HIGH. De ser verdadero, el pin de dirección se pone en LOW lo que representa la dirección de movimiento en sentido horario que en este caso representaría una exhalación. Luego de esto se ejecuta un bucle que se repetirá determinado número de veces, este número de veces está dado por las revoluciones que se quieran ejecutar teniendo en cuenta que dadas las características del motor. Una revolución equivale a 200 pasos del motor.

Mientras se ejecuta ese bucle, se envían pulsos al motor con retrasos, entre menor tiempo tengan estos retrasos, mayor será la frecuencia con la que se envían los pulsos para el movimiento del motor.

Esto mismo se repite para el movimiento contrario o de inhalación con la excepción de que el pin de dirección se debe poner en HIGH para que se pueda mover en sentido antihorario. Esto se puede evidenciar mejor en la figura 3.11.

```
digitalwrite(x_habi, LOW); // Habilita el Driver
if (contador == 0){
if (digitalRead(boton1) == HIGH ){
digitalwrite(x_dire, LOW); // direccion de giro 1
contador++;
for(int i=0;i<tiempo;i++){ // da pasos por un tiempo
digitalwrite(x_paso, HIGH);
delayMicroseconds(retardo);
digitalwrite(x_paso, LOW);
delayMicroseconds(retardo);
}
}
}
```

Figura 3.11: Movimiento del motor sin librería.

3.2.2.1. Rutinas

Para la implementación de las rutinas se hizo uso de la librería AccelStepper.h de Mike McCauley. Se usó esta librería ya que aparte de permitir controlar mejor a velocidad de los pasos del motor, también se le podía agregar una aceleración o desaceleración a este movimiento y tener también un control de la posición del sistema que, en este caso, se controla con la cantidad de pasos que ha dado el motor, para así poder limitar el sistema a que se mueva hasta un límite de pasos que se fijó en aproximadamente 25800, para evitar que el embolo de la jeringa colisione con el final del sistema y así evitar daños al sistema y a la jeringa.

Para programar las rutinas con esta librería se inició configurando los pines de la misma forma como se ve en la figura 3.9, aparte de esto se debe configurar también el tipo de interfaz del motor que en este caso sería 1 ya que se está usando un controlador de pasos y dirección. Luego de esto con ayuda de la clase Accelstepper() se define un nombre para el motor y se configura con la interfaz y los pines que se van a usar para los pasos y la dirección.

Una vez realizada esta primera configuración, se pasa a definir una velocidad máxima de pasos por segundos del motor con ayuda de la función setMaxSpeed(), en este caso se define una velocidad máxima de 18000 pasos por segundo. Luego de definir la velocidad deseada se pasa a crear la rutina, para empezar se debe especificar la posición de los pasos del motor como 0 con ayuda de la función setCurrentPosition(), esto hace que donde sea que este el motor sea tomado como la posición 0. Una vez configurado esto, se le pone la condición de que mientras la posición del motor sea diferente a la deseada. En el momento en que llega a la posición establecida, se define de nuevo esa posición como 0 y se devuelve la cantidad de pasos deseados, teniendo en cuenta de que se le debe poner el signo menos antes del número de pasos que se requieran. Luego de esto, siguiendo el mismo procedimiento pasado, mientras la posición sea diferente a la definida el motor se moverá a una velocidad establecida, con la condición de que la entrada de la función en este caso debe ser también

negativa.

```

Motor1.setCurrentPosition(0);
if (var1 == HIGH){
  while(Motor1.currentPosition() != -10000){
    Motor1.setSpeed(-18000);
    Motor1.runSpeed();
  }
  delay(2000);
  Motor1.setCurrentPosition(0);
  while(Motor1.currentPosition() != 9800){
    Motor1.setSpeed(3000);
    Motor1.runSpeed();
  }
}

```

Figura 3.12: Código para la rutina 1.

Como se puede apreciar en la figura 3.12, para la primera rutina se configura una distancia de 10000 a una velocidad de 18000 pasos/segundo. Debido a que la lógica del sistema implica que para una inspiración el mecanismo se debe de mover de derecha a izquierda y para una espiración debe de moverse de izquierda a derecha y que para una respiración se inicia inspirando aire y luego espirando este aire, la posición inicial del embolo debe ser a la derecha del mecanismo. Para esta rutina se buscaba generar una gráfica de un pico, lo que en la práctica significa una inspiración de aire rápida, para lo cual se configuró la velocidad máxima del sistema y una distancia corta. Una vez alcanzada esta distancia, el programa espera 2 segundos y se devuelve la misma distancia para volver a su posición inicial.

```

Motor1.setCurrentPosition(0);
if (var2 == HIGH){
  while(Motor1.currentPosition() != -15000){
    Motor1.setSpeed(-6000);
    Motor1.runSpeed();
  }
  delay(2000);
  Motor1.setCurrentPosition(0);
  while(Motor1.currentPosition() != 14000){
    Motor1.setSpeed(3000);
    Motor1.runSpeed();
  }
}

```

Figura 3.13: Código para la rutina 2.

En la figura 3.13 se muestra la rutina 2. Para esta rutina se necesitaba generar una gráfica de tipo escalón, lo que quiere decir una inspiración de flujo constante por un tiempo más largo. Esto se

logró con la configuración de una distancia de 15000 pasos a una velocidad de 6000 pasos/segundo, al agregar más distancia y disminuir la velocidad se espera que se consiga una toma de flujo constante ya que el sistema se demorará un poco más en completar los pasos requeridos.

```
Motor1.setCurrentPosition(0);
if (var3 == HIGH){
  Motor1.setMaxSpeed(-12000);
  Motor1.setAcceleration(-8000);

  Motor1.moveTo(-6000);
  Motor1.runToPosition();

  delay(2000);

  Motor1.moveTo(0);
  Motor1.runToPosition();
}
```

Figura 3.14: Código para la rutina 3.

Finalmente, en la figura 3.14 se puede observar la rutina 3. Para esta se buscaba generar una gráfica tipo sinusoidal lo que significa una respiración rápida y luego lenta. Así, se hizo uso de una de las herramientas de la librería que permite agregarle una aceleración al motor. En este caso se determinó una distancia de 12000 pasos a una velocidad de 8000 pasos/segundos con una aceleración de 6000 pasos/segundo.segundo.

3.2.3. Interfaz

3.2.3.1. Comunicación con Arduino

En este apartado se hará énfasis en la comunicación entre LabView y Arduino tomando un enfoque desde LabView, la cual, como ya se ha mencionado anteriormente, se logró con la extensión de VISA de National Instruments. LabView es un entorno de programación gráfica usado ampliamente para desarrollar sistemas, pruebas automatizadas de investigación, validación y producción. La programación de estos sistemas se puede hacer de varias maneras, pero la más usada y común en este programa es mediante bloques.

Para la comunicación, lo primero que se debe hacer es crear un bloque de VISA de configuración del puerto serial. A este bloque se le pueden crear hasta diez constantes o controles, y en este trabajo se usaron dos: el nombre del recurso y la velocidad de transmisión. Al configurar este bloque se puede hacer de hasta tres formas, la primera es crear un control, lo que quiere decir que el usuario desde la interfaz puede controlar esta entrada de este bloque, la segunda es una constante que el usuario solo podrá modificar en el diagrama de bloques, no en la interfaz y la tercera es un indicador que solamente muestra en la interfaz lo que se ha seleccionado. En este primer bloque, se empleó un controlador para asignar el nombre del recurso. Dicho controlador permite seleccionar el puerto serial por el cual el programa se comunicará. Como esto puede cambiar según las conexiones que

haga el usuario, se consideró dejarlo como un control desde la interfaz. Para la tasa de transferencia se usó una constante, ya que esta tasa está definida desde Arduino y es muy poco probable que se cambie.

Una vez configurado este bloque, el programa estará listo para comunicarse por el puerto serial seleccionado. Para esta aplicación se necesitaba que desde la interfaz se seleccionara la rutina deseada por el usuario, para esto se crearon 3 botones en la interfaz los cuales generan 3 bloques de los botones en el diagrama de bloques. Estos bloques tienen como salida un TRUE o un FALSE, debido a que la función en Arduino espera un número, se usa un bloque para convertir esta salida en un 1 o un 0, luego a la salida de este bloque se inserta un convertidor de número a una cadena de decimales. Una vez los tres botones estén configurados de esta forma, se pasa a concatenar estas salidas para que queden juntos en una cadena de decimales de los cuales cada valor estará separado por una coma. Luego de que se organizan de esta forma, esta información entra al bloque de escritura de VISA y gracias a este bloque se escriben estos datos en el puerto serial. Esto se puede apreciar mejor en la figura 3.15.

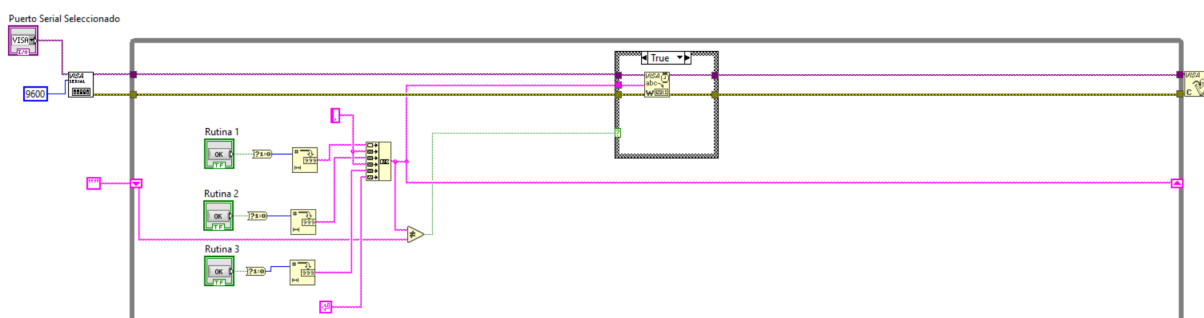


Figura 3.15: Comunicación entre LabView y Arduino.

3.2.3.2. Adquisición de Datos

Para la adquisición de datos del sensor, como se está usando un módulo DAQ de National Instruments, la comunicación y programación es mucho más sencilla. En este caso se tienen dos opciones, debido a que el mencionado anteriormente LabView cuenta con una extensión llamada DAQmx la cual es exclusiva para trabajar con módulos DAQ. Entre estas opciones se encuentra el bloque DAQ Assistant desde el cual se define si se quiere adquirir o generar señales, qué tipo de señales, la terminal de entrada o de salida y muchas otras opciones. Pero, con el fin de tener un mejor control de todo el proceso, se decidió hacer la comunicación con el módulo paso a paso.

El proceso se inicia creando una DAQ virtual con el bloque de DAQmx Create Virtual, en este bloque se define la configuración del módulo y en este caso se realizó esta configuración con constantes para evitar que el usuario mueva algo que pueda dañar esta comunicación. En este bloque se define que se va a adquirir una señal de voltaje entre 0 y 7.54V y las terminales se usarán en modo RSE. Luego de configurar este bloque, se conecta a uno para configurar el tiempo de muestreo,

en este caso se necesita que el sistema este midiendo constantemente por lo que se selecciona un modo de muestreo constante, tomando 10 muestras con una frecuencia de 1000 Hz. Luego de esto se conecta a un bloque que inicia la tarea con esta configuración, que a su vez está conectado a otro bloque que lee los datos cada 100 ms y los libera como un solo dato por un canal (1). Esto se puede evidenciar mejor en la figura 3.16.

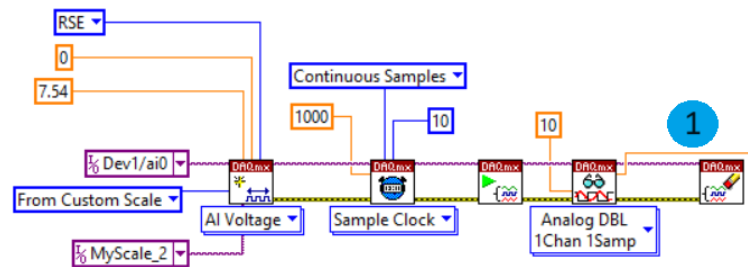


Figura 3.16: Comunicación entre módulo DAQ y LabView.

3.2.3.3. Gráfica

Debido a que el sensor es un sensor de presión diferencial y los datos que se necesitan son de flujo, se debe operar la salida del bloque de lectura para convertir estos datos de voltaje a presión y de presión a flujo. Para esto se usa nodo de formula donde se pueden configurar varias entradas y salidas y operarlas dentro de este nodo de diferentes formas. Este nodo recibe el dato de la salida del canal de la DAQ virtual (1) y se opera con las ecuaciones y filtros usados por otro grupo de trabajo de la Universidad, los cuales llegaron a estos datos mediante las pruebas mencionadas en el apartado de calibración del sensor. Estos datos salen del nodo de formula y van a una gráfica que mostrará en la interfaz los datos con respecto al tiempo que ha transcurrido desde que se inicia la interfaz hasta que se apaga.

3.2.3.4. Tabla

Generalmente cuando se realizan inspirometrías, se hacen usando como referencia las curvas de ventilación mecánica. Las más comunes son las de flujo constante u onda cuadrada, flujo acelerado y desacelerado y flujo sinusoidal [22] [23]. Gracias a la gráfica se puede comprobar que se está realizando la rutina de manera adecuada, pero esta no muestra los valores exactos de flujo, por esto, se necesita de una tabla que pueda mostrar los valores tomados por el sensor. Para esta tabla se toma el valor luego de ser operado (1) y se transforma a un string, luego de esto se guarda en un array. Posteriormente, se usa el bloque de hora y fecha para generar ambos datos, estos se separan y se guardan individualmente en un array, estos arrays temporales van conectados a un shift register (3) y (4) para permitir que se tome un dato cada que se repita el ciclo while de todo el proceso. Luego de esto se construye el array principal, el cual genera un bloque de los datos, se toma este bloque de datos y se envía al bloque de tabla para que se visualicen en la interfaz, como se puede observar en la figura 3.17. Este dato del array (2) se usará posteriormente para el formulario.

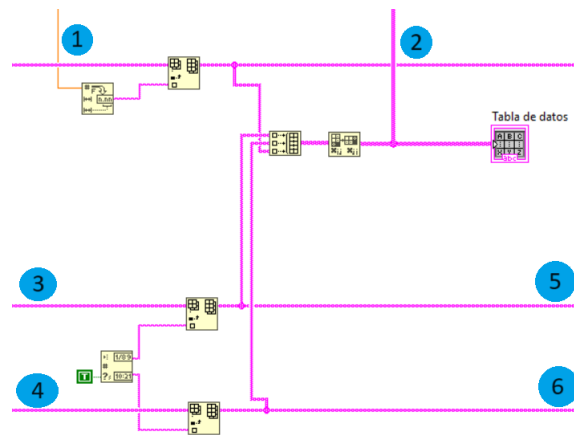


Figura 3.17: Creación de la tabla.

3.2.3.5. Formato de calibración

Como se mencionaba anteriormente, los datos son almacenados en la tabla de manera temporal, esto quiere decir que una vez se apague la interfaz estos serán borrados de la tabla. LabView ofrece muchas alternativas para este problema y una de ellas es excel, para lograr esto se debe crear inicialmente una plantilla en excel que contenga una tabla y los espacios donde se desee almacenar la información que enviará LabView.

Formato de calibración			
	Flujo objetivo:	783.33	Flujo Maximo: 734.07
			Error: 6.29%
Fecha	Hora	Flujo	
2/1/2024	2:39:57 PM	0.00	
2/1/2024	2:39:58 PM	0.00	
2/1/2024	2:39:59 PM	0.00	
2/1/2024	2:39:59 PM	0.00	
2/1/2024	2:39:59 PM	0.00	
2/1/2024	2:39:59 PM	0.00	
2/1/2024	2:39:59 PM	0.00	
2/1/2024	2:39:59 PM	0.00	
2/1/2024	2:39:59 PM	0.00	
2/1/2024	2:39:59 PM	0.00	
2/1/2024	2:39:59 PM	268.75	
2/1/2024	2:39:59 PM	469.28	
2/1/2024	2:39:59 PM	574.92	
2/1/2024	2:40:00 PM	672.78	
2/1/2024	2:40:00 PM	734.07	
2/1/2024	2:40:00 PM	716.10	
2/1/2024	2:40:00 PM	696.55	
2/1/2024	2:40:00 PM	348.28	
2/1/2024	2:40:00 PM	174.14	

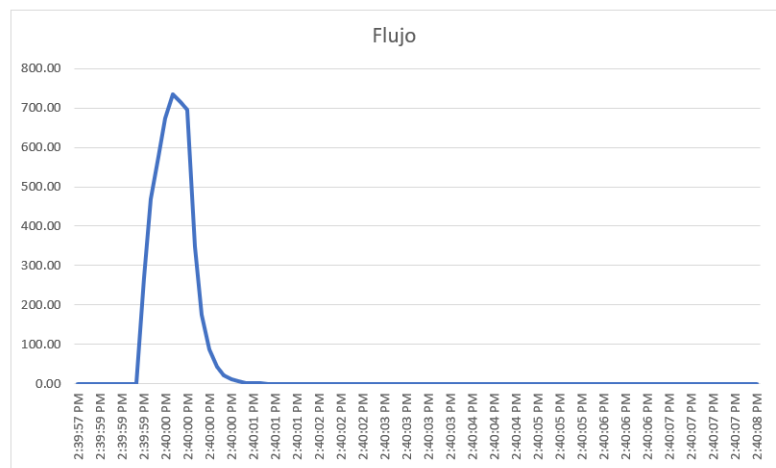


Figura 3.18: Formato de calibración.

Como se puede ver en la figura 3.18, esta plantilla cuenta con un apartado para la fecha, hora

y el valor de flujo que este midiendo el sensor, además de esto, con una celda para el flujo objetivo, el cual se ingresa manualmente desde la interfaz y que es el flujo máximo de referencia que debería coincidir con el valor de flujo máximo tomado por el sensor, mostrado en la celda de "Flujo Máximo". Una vez se tengan estos dos valores, se hace la operación automática para encontrar el error entre estas dos mediciones y se muestra en la celda de Error. Este formato cuenta también con una gráfica la cual mostrará todos los datos que se hayan tomado y permitirá también guardar esta gráfica para su posterior análisis.

Luego de tener listo el formato de calibración, se debe de configurar el envío de los datos entre LabView y excel. Inicialmente se crea un botón en la interfaz, el cual va a permitir hacer el envío de estos datos. Además de esto se configura la entrada del dato de referencia que se va a usar para la secuencia, esto se logra con un indicador el cual multiplica el valor ingresado por 1000 y divide por 60 debido a que el dispositivo de referencia muestra los datos en Litros/segundos y para este proyecto se necesitan valores en cm^3/s . Dentro de un bloque if, se usará inicialmente el bloque para crear un reporte. Ya que necesitamos que este reporte se cree en un archivo existente, se le crea un control en el cual se introduce manualmente la dirección de la plantilla y se especifica en que programa va a crear el reporte, luego de esto se especifica en qué filas y columnas se quieren guardar los datos tomados (2) e introducidos por el usuario y finalmente se elimina el formulario internamente para que el programa quede disponible para crear uno nuevo, como se puede ver en la figura 3.19.

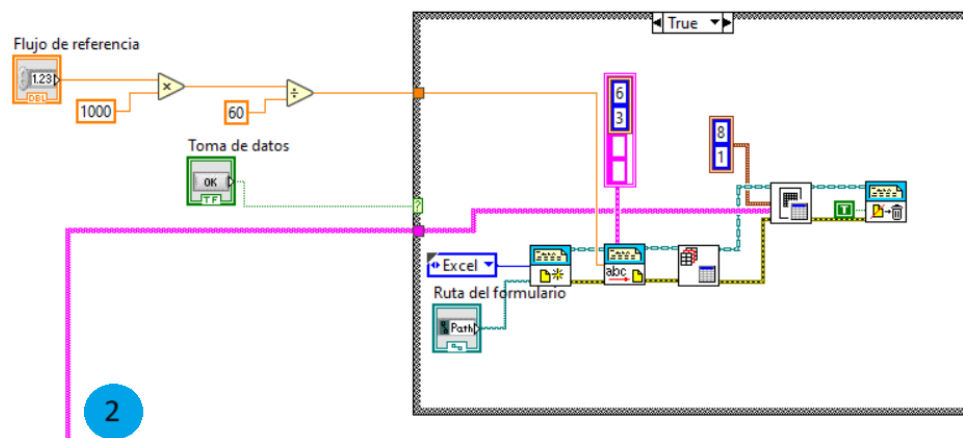


Figura 3.19: Envío de datos de LabView a Excel.

Todas estas configuraciones para comunicar LabView, Arduino y el modulo de adquisición de datos son esenciales para la interfaz, como se mencionaba anteriormente y como se puede observar en la figura 3.20, desde esta se podrá seleccionar la rutina deseada y se podrán revisar los datos que este enviando el sistema para determinar si el flujo que sale del sistema es el mismo que está midiendo el inspirómetro y de esta forma validar si se puede usar o no.



Figura 3.20: Interfaz de usuario.

A lo largo de este capítulo se puede evidenciar como se dio el cumplimiento del cuarto y quinto objetivo específico planteado en el proyecto, luego de que se logrará la implementación del sistema mecánico que fuera capaz de emular el proceso de respiración de una persona por medio de rutinas las cuales pueden ser seleccionadas desde una interfaz en donde se puede observar las gráficas de flujo en tiempo real y desde la cual se pueden generar reportes con los datos tomados y una gráfica de estos datos para su posterior análisis.

Pruebas y Análisis de Resultados

4.1. Pruebas

Luego de la implementación tanto del hardware como del software, se pasó a la etapa de pruebas. Para este proyecto, durante toda la implementación se probaron los diferentes bloques del sistema por separado para posteriormente probarlos juntos a medida que se realizaban las conexiones y comunicaciones respectivas. Además de esto las pruebas se dividieron en dos bloques principales, las pruebas mecánicas las cuales consisten en las pruebas iniciales realizadas al motor implementado en el sistema mecánico sin la jeringa y luego con la jeringa y las pruebas de software en las cuales se probó la comunicación entre la interfaz y Arduino.

4.1.1. Pruebas Mecánicas

Una vez instalado el motor paso a paso en el sistema mecánico, se necesitaba probar como era el comportamiento del motor con el sistema en cuanto a velocidad y tiempo, como aún no se realizaba la comunicación entre LabView y Arduino estas pruebas se hicieron con la ayuda de dos pulsadores, un pulsador se encargaba de dar la señal para que el motor se moviera en una dirección y el otro para que se moviera en la dirección opuesta, para esta prueba se usó otro código de Arduino como se mencionó en el capítulo anterior, este código sin librerías mueve el motor según la frecuencia que se defina en el programa, por esto no se puede tener un control más exacto de la velocidad a la que se desea mover la jeringa y por esta razón se decidió dejar este método únicamente como prueba del funcionamiento del motor y posteriormente del mecanismo.

Luego de probar el motor y el mecanismo solos, se decidió acoplar la jeringa para verificar que el mecanismo se movía sin problemas y sin interrupciones. Se repitió cada rutina cinco veces para garantizar que en estas 5 pruebas el comportamiento del sistema fuera el mismo y que el embolo de la jeringa llegara siempre hasta el mismo lugar. Una vez verificado esto, se acopló el sensor, para validar su funcionamiento se usó la terminal del Arduino IDE. En esta terminal, se comprobó que el sensor estaba tomando datos acordes al movimiento y la velocidad de la jeringa y el motor.

Seguido a esto, se dieron paso a las pruebas con la librería `Accelstepper()`. Una vez programadas las tres rutinas, se probaron individualmente cinco veces para comprobar que el movimiento del motor y de la jeringa fuera el mismo en cada una de las pruebas. A diferencia de las pruebas anteriores, con esta librería si se podía tener un control de la posición en la que debería estar el embolo de la jeringa al momento de finalizar la rutina, por lo que se realizaron pruebas y ajustes

para determinar cuánto era la cantidad de pasos recomendada para que el sistema llegará al final de la carrera y no golpear la protección de la polea final, para así evitar daños en la jeringa y en el mecanismo.

Finalmente, se realizó una prueba con el mecanismo, la jeringa, un manómetro y el citrex para validar la respuesta del motor a las rutinas implementadas. Esto con el fin de comprobar que las rutinas daban como resultado las gráficas que se usan en la ventilación mecánica las cuales son usadas para las terapias de respiración [23]. Como se puede ver en la figura 1.6.

4.1.2. Pruebas de Software

Luego de comprobar el funcionamiento del mecanismo solo y con la jeringa, se necesitaba comunicar esto con la interfaz para remover los pulsadores físicos y que todo quedará controlado desde la interfaz. Para lograr esto se creó una interfaz básica que constaba de dos botones que simulaban los pulsadores y dos indicadores que representaban el estado de los pulsadores físicos, esto con el fin de comprobar que se estuviera haciendo la lectura y escritura desde LabView. Inicialmente se quería comprobar que el motor se moviera con el comando de LabView por lo que solo se usó una rutina y por lo tanto un botón, una vez validado esto, se dio paso a probar las tres rutinas desde la interfaz y con el programa nuevo en el que se usaba la librería mencionada anteriormente.

Después de validar la comunicación entre LabView y Arduino, se iniciaron las pruebas con el sensor y la tarjeta de adquisición de datos. Para estas pruebas se corrieron las rutinas y por medio de la gráfica implementada en la interfaz se comprobó que la tarjeta estuviera recibiendo los datos y los estuviera enviando a LabView. Durante estas pruebas se encontró que la tarjeta generaba un ruido muy alto debido a la resistencia que maneja la DAQ por dentro. Para solucionar esto se planteó usar un seguidor de voltaje, pero para no agregar más componentes externos, se decidió usar la configuración de la DAQ desde LabView y agregarle una escala de 2.46 lo cual removía el ruido generado por la resistencia interna. Después de verificar el sensor y la DAQ, se realizó el acople del filtro para lograr una mejor gráfica.

Seguido a esto, se realizaron pruebas para verificar la creación de los formatos de calibración para validar que se estuviera enviando la información correctamente al archivo, que se estuviera guardando cada vez que se accionaba el botón de toma de datos y no cada vez que tomara un dato y para verificar que la gráfica estuviera graficando únicamente los datos que se habían tomado para ese archivo.

Finalmente se realizaron las pruebas finales las cuales consistían en probar todo el sistema junto funcionando correctamente, es decir, que las rutinas se corrieran de acuerdo a la solicitud del usuario, que la DAQ tomara los datos del sensor y los graficara correctamente y que el formulario se llenara con los datos deseados. Para esta última prueba se usó el citrex, el cual fue proporcionado por la Pontificia Universidad Javeriana Cali, para verificar el porcentaje de error entre las medidas tomadas

por el sensor y la de referencia que en este caso era proporcionada por el citrex.

4.2. Resultados

4.2.1. Resultados iniciales

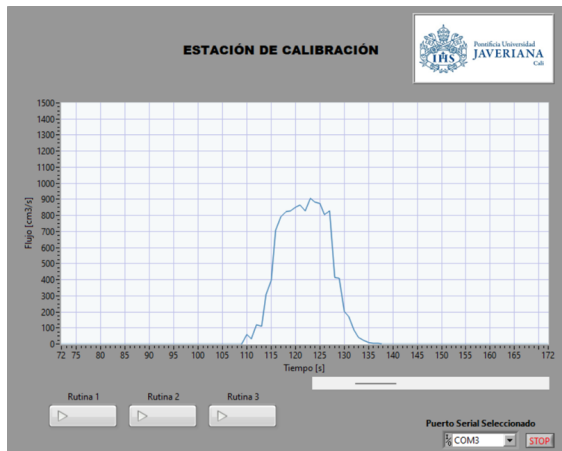


Figura 4.1: Prueba de la Rutina 1 (Solo jeringa y sensor).

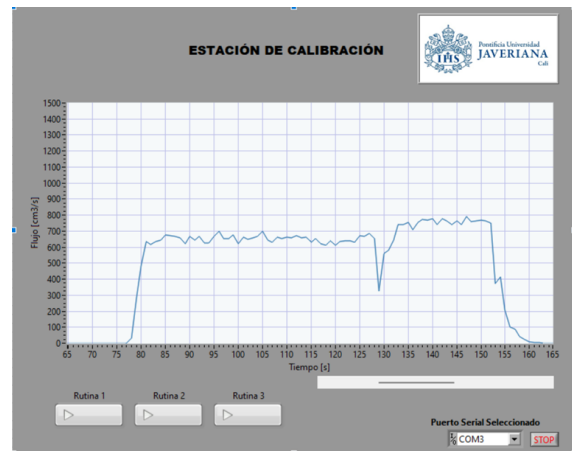


Figura 4.2: Prueba de la Rutina 2 (Solo jeringa y sensor).

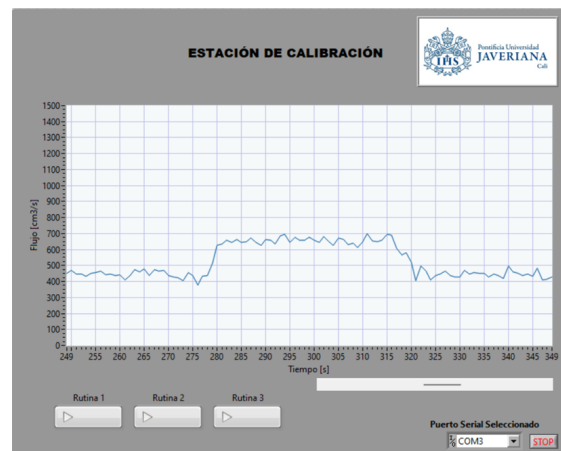


Figura 4.3: Prueba de la Rutina 3 (Solo jeringa y sensor).

Para las primeras pruebas de todo el sistema, se utilizó la jeringa, el sensor y la DAQ. En este caso, se estaba explorando la respuesta de la estación a los tipos de rutina que se querían implementar y a los cambios que podía tener la gráfica si se variaba la velocidad o aceleración en algún

tramo del recorrido.

En este caso, la figura 4.1 corresponde a la rutina 1 la cual se programó para que el motor se moviera 5000 pasos a 14500 pasos/segundo. La figura 4.2 corresponde a la rutina 2 la cual se programó para que el motor se moviera 10000 pasos con una aceleración de 100 pasos/segundo.segundo y una velocidad de 10000 pasos/segundo, luego 5000 pasos sin aceleración con una velocidad de 13000 pasos/segundos. Por último, la figura 4.3 corresponde a la rutina 3 en la cual se programó el motor para que se moviera 7000 pasos a 5000 pasos/segundos con una aceleración de 1000 pasos/segundo.segundo, luego 7000 pasos sin aceleración a 10000 pasos/segundo y por último 7000 pasos a 5000 pasos/segundo con una desaceleración de 1000 pasos/s².

Como se puede ver en la figura de la rutina 1, se buscaba la forma de esta gráfica al mover el embolo una distancia corta a una velocidad alta lo que significaría una toma de aire rápida por una persona. Para la rutina 2 se quería observar cómo se comportaba la gráfica si el sistema empezaba con una aceleración y después se movía sin esta aceleración, como se puede observar en la figura 4.2 la gráfica toma la forma de un escalón y en el momento en el que se mueve sin la aceleración a una velocidad mayor baja abruptamente el flujo y luego sube a un valor mayor de flujo debido al aumento de la velocidad por lo que la rutina de esta forma se descartó. Finalmente, para la tercera rutina se buscaba obtener una gráfica tipo sinusoidal acelerando y desacelerando el sistema para lograr esto, como se puede observar en la figura 4.3 el resultado estuvo alejado de lo que se esperaba.

4.2.2. Resultados finales

Para las pruebas de la respuesta del motor a las rutinas finales, se obtuvieron los siguientes resultados:

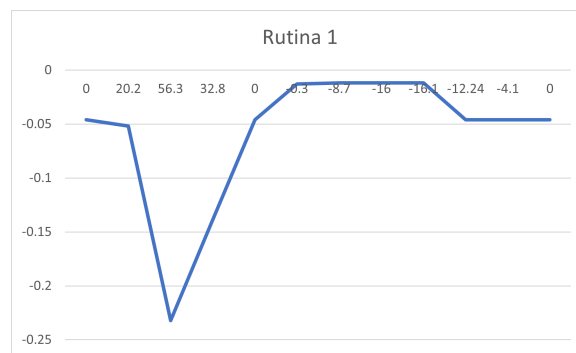


Figura 4.4: Prueba de la Rutina 1 (Jeringa, Manómetro y Citrex).

Como se puede observar debido a las conexiones y a la lógica que se usó en las pruebas de que el dispositivo se moviera de izquierda a derecha, las gráficas en la que se centró la prueba se encuentran

en los valores negativos. En la figura 4.4 se puede ver una curva de tipo flujo desacelerado como la que se encuentra en el apartado B de la figura 1.6, este tipo de curva se genera cuando el aire entra rápidamente al sistema y luego la velocidad de este disminuye, para lograr este tipo de gráfica se programó el sistema para que recorriera 10000 pasos a una velocidad de 18000 pasos/segundo.

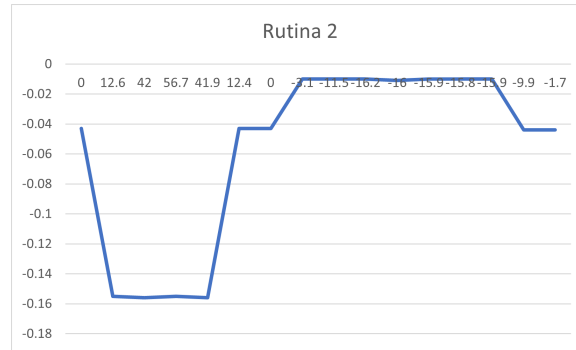


Figura 4.5: Prueba de la Rutina 2 (Jeringa, Manómetro y Citrex).

En la figura 4.5 se muestra el resultado de la prueba para la rutina 2. En este caso se puede ver una curva de flujo constante o de onda cuadrada, para lograr este tipo de gráfica se le realizó un cambio con respecto a la primera rutina por lo que para esta se definió que se moviera 15000 pasos a una velocidad de 6000 pasos/segundo.

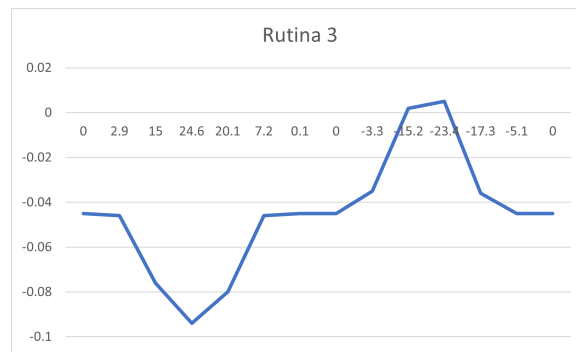


Figura 4.6: Prueba de la Rutina 3 (Jeringa, Manómetro y Citrex).

Finalmente, en la figura 4.6 se puede apreciar una gráfica de flujo tipo sinusoidal, este tipo de gráfica ocurre cuando el flujo ingresa de una forma lenta y va acelerando su ingreso a la persona hasta alcanzar su punto máximo para luego ir desacelerando [23], para lograr esta gráfica se programó el sistema para que se moviera 6000 pasos a una velocidad de 12000 pasos/segundos con una aceleración de 8000 pasos/s^2 . La diferencia de esta programación con respecto a la de la prueba anterior fueron las velocidades y distancias y el uso de la herramienta de la librería `moveto()` la cual permite configurar una velocidad y una aceleración para el tramo que se desee aplicar.

Una vez comprobado que el motor estaba respondiendo correctamente a las rutinas y estas estaban arrojando las gráficas que se necesitaban, se pasó a probar el sistema final con todos sus componentes y siguiendo la lógica de una respiración, es decir, iniciando a la derecha del recorrido y moviéndose de derecha a izquierda para una inspiración y de izquierda a derecha para una espiración. A su vez se probaron los formularios para comprobar que el formulario estaba arrojando la misma gráfica de la interfaz, estaba guardando el dato del flujo de referencia y si estaba calculando el error entre este valor y el valor máximo alcanzado por el sensor.

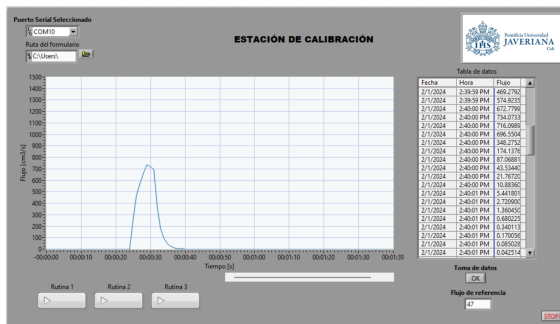


Figura 4.7: Prueba Final - Rutina 1 - HMI.

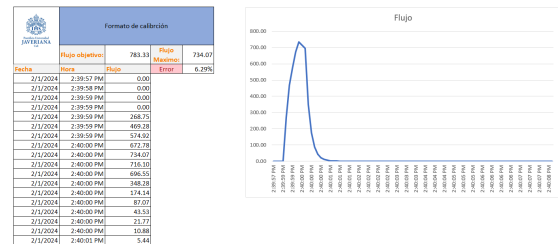


Figura 4.8: Prueba Final - Rutina 1 - Formulario.

Prueba	Flujo (Sensor) [cm3/s]	Flujo (Citrex) [cm3/s]	Error
1	734.06	783.33	6.29%
2	787.08	860	8.48%
3	729.49	818.33	10.86%
4	770.52	888.33	13.26%
5	768.55	871.67	11.83%
Promedio	757.94	844.332	10.14%

Figura 4.9: Resultados de la Rutina 1.

Como se puede observar en las figuras 4.7 y 4.8 las gráficas tanto en la interfaz como en el formulario son iguales para la rutina 1. Esta prueba se repitió 5 veces para comprobar que tanto variaba el error en cada intento y como se puede ver en la figura 4.9 en promedio se obtuvo un error del 10.14% para esta primera rutina, en cuanto a las gráficas, estas se mantuvieron similares en todos los casos.

Para la rutina 2 se realizaron las mismas pruebas a excepción de que para las dos últimas se decidió programar un recorrido más largo y una velocidad más larga para comprobar el error para distancias más largas. Como se puede ver en las figuras 4.10 y 4.11 las gráficas nuevamente arrojaron una figura tipo escalón la cual es igual tanto en la interfaz como en el formulario para todas las

pruebas que se realizaron. En promedio se obtuvo un error del 6.87% para esta rutina donde el flujo máximo alcanzado por el sensor fue de 466.63 cm³/s y por el citrex fue de 500 cm³/s. En el momento en el que se realizó las pruebas para una distancia más larga se encontró que producía un error mucho más grande de aproximadamente 21.66% en promedio.



Figura 4.10: Prueba Final - Rutina 2 - HMI.

Formato de calibración			
Flujo objetivo	Flujo	Flujo Medido	Error
498.33	466.63	466.63	6.36%
Fecha	Hora	Flujo	
2/2/2024	3:27:12 PM	0.00	
2/2/2024	3:27:14 PM	0.00	
2/2/2024	3:27:14 PM	262.656	
2/2/2024	3:27:14 PM	378.68	
2/2/2024	3:27:14 PM	262.66	
2/2/2024	3:27:14 PM	314.95	
2/2/2024	3:27:14 PM	392.29	
2/2/2024	3:27:15 PM	418.02	
2/2/2024	3:27:15 PM	425.36	
2/2/2024	3:27:15 PM	444.874	
2/2/2024	3:27:15 PM	458.954	
2/2/2024	3:27:15 PM	410.471	
2/2/2024	3:27:15 PM	410.471	
2/2/2024	3:27:15 PM	408.93	
2/2/2024	3:27:15 PM	444.68	
2/2/2024	3:27:15 PM	405.36	
2/2/2024	3:27:15 PM	410.47	
2/2/2024	3:27:15 PM	418.27	
2/2/2024	3:27:15 PM	466.63	
2/2/2024	3:27:16 PM	493.11	
2/2/2024	3:27:16 PM	414.09	

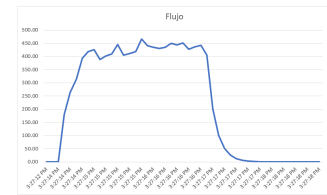


Figura 4.11: Prueba Final - Rutina 2 - Formulario.

Finalmente, para la rutina 3 se realizaron 5 pruebas con la misma configuración. En las figuras 4.12 y 4.13 se puede ver que se obtuvo una gráfica tipo sinusoidal tanto en la interfaz como en el formulario. Como se puede evidenciar en la figura 4.14 en promedio se obtuvo un error del 2.22% siendo la rutina que mejor se desempeñó en las pruebas frente al Citrex.



Figura 4.12: Prueba Final - Rutina 3 - HMI.

Formato de calibración			
Flujo objetivo	Flujo	Flujo Medido	Error
355.00	334.47	334.47	5.78%
Fecha	Hora	Flujo	
2/2/2024	4:20:41 PM	0.00	
2/2/2024	4:20:44 PM	0.00	
2/2/2024	4:20:44 PM	82.89	
2/2/2024	4:20:44 PM	137.83	
2/2/2024	4:20:44 PM	355.30	
2/2/2024	4:20:44 PM	281.46	
2/2/2024	4:20:44 PM	291.578	
2/2/2024	4:20:44 PM	201.024	
2/2/2024	4:20:44 PM	277.8541	
2/2/2024	4:20:45 PM	307.9621	
2/2/2024	4:20:45 PM	330.3842	
2/2/2024	4:20:45 PM	334.4958	
2/2/2024	4:20:45 PM	280.9077	
2/2/2024	4:20:45 PM	260.8141	
2/2/2024	4:20:45 PM	154.4075	
2/2/2024	4:20:45 PM	63.20354	
2/2/2024	4:20:45 PM	12.60719	

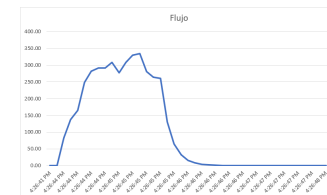


Figura 4.13: Prueba Final - Rutina 3 - Formulario.

Prueba	Flujo (Sensor) [cm ³ /s]	Flujo (Citrex) [cm ³ /s]	Error
1	346.73	348.33	0.46%
2	343.33	333.33	0.30%
3	334.47	355	5.78%
4	340.98	353.33	3.50%
5	330.2	326.67	1.08%
Promedio	339.142	343.332	2.22%

Figura 4.14: Resultados de la Rutina 3.

Conclusiones y Recomendaciones de Trabajo Futuro

5.1. Recomendaciones de Trabajo Futuro

Teniendo en cuenta los resultados obtenidos se procede a presentar las siguientes recomendaciones a futuro para mejorar la estación de calibración en cuanto a diseño y funcionamiento.

1. Sustitución de la jeringa de calibración por un suministro de oxígeno limpio para poder manejar valores mayores a 3L de aire y de capacidad que tiene la jeringa. Este tanque de oxígeno conectado a un regulador de presión y a una válvula proporcional para convertir el control del sistema a un control de lazo cerrado.
2. Para mejorar la resolución de las gráficas se recomienda usar el sensor de flujo de la marca Sensiron SFM3200-AW. Este sensor es reutilizable y es creado especialmente para aplicaciones médicas, lo cual permitirá tener una medida mucho más exacta del flujo que está pasando a la salida de la jeringa y, por ello, dará como resultado una gráfica mucho más exacta.
3. Cambiar el motor NEMA 17 por el motor NEMA 23 en caso de que se requiera implementar más dispositivos o sensores al sistema. Esto debido a que a medida que se le acoplan más dispositivos al sistema, cada uno generará una mayor resistencia al movimiento del motor, por lo que se necesitará un motor que tenga la potencia necesaria para mover esa carga.

5.2. Conclusiones

- Para la validación de los inspirómetros es importante contar con un dispositivo que, a la vez que pueda probar qué tan exacto es, también pueda simular diferentes tipos de respiraciones que se van a hacer en el inspirómetro de manera exacta y precisa, ya que actualmente estos procesos se hacen de manera manual, lo que puede llevar a imprecisiones.
- Para simular el proceso de respiración de una persona se necesita de un suministro de aire sobre el cual se pueda tener un control, un actuador que permita el paso de este aire o que provoque el flujo necesario, un controlador que le de instrucciones al sistema y un sensor que permita verificar como se está haciendo la respiración.

- Se logró el diseño y la implementación de un sistema capaz de emular la respiración de una persona en su fase de inspiración y espiración, además de esto es capaz de simular diferentes tipos de respiración o flujos inspiratorios y a su vez es capaz de graficar y generar un reporte de los datos tomados hasta el momento de la toma.
- El proyecto está abierto a ser escalable en un futuro, es decir, se le pueden implementar otros sensores si se necesitan medir otras variables y se le pueden agregar más instrucciones de ser requerido, sin necesidad de cambiar lo que está implementado actualmente en la estación de calibración.
- El sistema posee un error aceptable y funciona bien para velocidades altas y bajas y para distancias cortas. Sin embargo, para rutinas que impliquen que el actuador se mueva distancias mayores a 25 cm, el error producido por el sensor aumentará y, por lo tanto, no se garantiza una medida confiable del flujo.
- Las rutinas actualmente implementadas pueden ser usadas para calibrar inspirómetros, sin embargo, si se desea tener una gráfica más exacta y precisa del flujo, se debe realizar un gasto mayor en el sensor para lograr estos resultados.

Bibliografía

- [1] Dong, E., Du, H. y Gardner, L. (2020)." An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases*", 20(5), 533–534. [Online]. Available: [https://doi.org/10.1016/s1473-3099\(20\)30120-1](https://doi.org/10.1016/s1473-3099(20)30120-1). [Accessed: 09- May- 2022].
- [2] " Denise Battaglini, Chiara Robba, Salvatore Caiffa, Lorenzo Ball, Iole Brunetti, Maurizio Loconte, Daniele Roberto Giacobbe, Antonio Vena, Nicolò Patroniti, Matteo Bassetti, Antoni Torres, Patricia RM Rocco, Paolo Pelosi, Chest physiotherapy: An important adjuvant in critically ill mechanically ventilated patients with COVID-19, *Respiratory Physiology & Neurobiology*, Volume 282, 2020, 103529, ISSN 1569-9048"[Online]. Available: <https://doi.org/10.1016/j.resp.2020.103529>. [Accessed: 09- May- 2022].
- [3] P. Consultores, “Calibración de Espirometros,” 03-Nov-2018. [Online]. Available: https://youtu.be/QyBIwuTOpr8?si=_wC7U16st2fL8eq2.
- [4] " Ricardo Duarte, Michel Mesnard, Jean-Pierre Nadeau. An innovative design approach to develop external articular medical devices.(2017).*International Journal on Interactive Design and Manufacturing*, Vo.11,pp.375–383."
- [5] J. Canet “Fisiología Respiratoria”, sf. [Online]. Available: <https://www.studocu.com/co/document/universidad-santiago-de-cali/cirugia/fisiologia-respiratoria/11795161>. [Accessed: 09- May- 2022].
- [6] Setipa, J., 2020. Producción local: la solución a la escasez de equipos médicos esenciales. [online] Available at:https://elpais.com/elpais/2020/05/14/planeta_futuro/1589467257_970330.html. [Accessed 12 October 2021].
- [7] " Información básica sobre la COVID-19. WHO | World Health Organization."[Online]. Available: <https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19>. [Accessed: 09- May- 2022].
- [8] Lazzeri, M., Lanza, A., Bellini, R., Bellofiore, A., Cecchetto, S., Colombo, A., D’Abrosca, F., Del Monaco, C., Gaudellio, G., Paneroni, M., Privitera, E., Retucci, M., Rossi, V., Santambrogio, M., Sommariva, M., Frigerio, P., 2020. " Respiratory physiotherapy in patients with COVID-19 infection in acute setting: a Position Paper of the Italian Association of Respiratory Physiotherapists (ARIR). *Monaldi Arch. Chest Dis.* 90."
- [9] American Thoracic Society, “Standardization of Spirometry 2019 Update. An Official American Thoracic Society and European Respiratory Society Technical Statement,” *American Journal of Respiratory and Critical Care Medicine*, vol. 200, no. 8, pp. 70–88, Oct. 2019.

- [10] " Consideraciones relativas a la rehabilitación durante el brote de COVID-19. IRIS PAHO Home." [Online]. Available: https://iris.paho.org/bitstream/handle/10665.2/52104/OPSNMHHMCOVID-19200010_spa.pdf?sequence=5. [Accessed: 09- May- 2022].
- [11] I. C. Coropetchi, A. I. Indreş, F. Moţa and A. Vasile, "Mechanical Test Lung Simulation Device," 2020 International Conference on e-Health and Bioengineering (EHB), 2020, pp. 1-4, doi: 10.1109/EHB50910.2020.9280171.
- [12] Abdellah El-Hadj, Mohamed Kezrane, Hijaz Ahmad, Houari Ameer, S. Zamree Bin Abd Rahim, Abdelhakime Younsi, Hanaa Abu-Zinadah, "Design and simulation of mechanical ventilators, Chaos, Solitons & Fractals, Volume 150, 2021, 111169, ISSN 0960-0779."
- [13] Jayant Giri, Niraj Kshirsagar, Aishwary Wanjari, "Design and simulation of AI-based low-cost mechanical ventilator: An approach, Materials Today: Proceedings, 2021, ISSN 2214-7853."
- [14] Javier Chang, Augusto Acosta, Jorge Benavides-Aspiazu, Jaime Reategui, Christiam Rojas, Jordi Cook, Richard Nole, Luigi Giampietri, Sandra Pérez-Buitrago, Fanny L. Casado, Benjamin Castaneda, "Masi: A mechanical ventilator based on a manual resuscitator with telemedicine capabilities for patients with ARDS during the COVID-19 crisis, HardwareX, Volume 9, 2021, e00187, ISSN 2468-0672."
- [15] " VentyNet, respiradores artificiales modulares, teleoperados - Universidad del Valle / Cali, Colombia." [Online]. Available: <https://www.univalle.edu.co/ciencia-y-tecnologia/respira-casa-alternativa-ventilador-artificial>. [Accessed: 09- May- 2022].
- [16] Instituto Nacional de Seguridad e Higiene en el Trabajo, "NTP 217: Validación de un espirómetro", 1988.
- [17] " INVIMA. (2013). ABC De Dispositivos Médicos" [online] Available at: <https://www.studocu.com/co/document/universidad-ces/biologia-molecular/abc-dispositivos-medicos-invima-invimapdf/18544550> [Accessed: 09- May- 2022].
- [18] Maya E. Kotas, B. Taylor Thompson, "Toward Optimal Acute Respiratory Distress Syndrome Outcomes: Recognizing the Syndrome and Identifying Its Causes, Critical Care Clinics, Volume 37, Issue 4, 2021, Pages 733-748, ISSN 0749-0704, ISBN 9780323794640."
- [19] Liliane Patrícia de Souza Mendes, Larissa Faria Borges, Letícia Mendonça, Dayane Montezzo, Giane Amorim Ribeiro-Samora, Veronica Franco Parreira, "Incentive spirometer: Aspects of the clinical practice of physical therapists from Minas Gerais working with patients with respiratory dysfunction, Brazilian Journal of Physical Therapy, 2021, ISSN 1413-3555."
- [20] S. L. C. Muñoz, J. A. L. Durán, and A. S. Ordóñez, "Técnicas de reexpansión pulmonar," in Técnicas de fisioterapia respiratoria: Perspectivas de práctica basada en la evidencia, Cali, Colombia: Universidad Santiago de Cali, 2022, pp. 157–185.

- [21] A. Segura, " Efecto de dos técnicas de reexpansión pulmonar en la ventilación de pacientes sometidos a cirugía cardiovascular. ensayo clínico controlado fase ii b", Trabajo de grado, Universidad del Valle, Santiago de Cali, 2018.
- [22] M. Perez y J. Mancebo, "Monitorización de la mecánica ventilatoria", Medicina intensiva, vol. 30, núm. 9, pp. 440–448, dic. 2006.
- [23] J. López-Herce y Á. Carrillo, "Ventilación mecánica: indicaciones, modalidades y programación y controles", Anales de Pediatría Continuada, vol. 6, núm. 6, pp. 321–329, dic. 2008.
- [24] Cienfuegos, I. and de la Torre, S., 2021. VOLÚMENES PULMONARES
- [25] M.R. Miller, J. Hankinson, V. Brusasco, F. Burgos, R. Casaburi, A. Coates, R. Crapo, P. Enright, C.P.M. van der Grinten, P. Gustafsson, R. Jensen, D.C. Johnson, N. MacIntyre, R. McKay, D. Navajas, O.F. Pedersen, R. Pellegrino, G. Viegi and J. Wanger.,2005. "Standardisation of spirometry"
- [26] Ministerio de Salud y Protección Social, 2014. ESTÁNDARES SEMÁNTICOS PARA LOS DISPOSITIVOS MÉDICOS. Bogotá.
- [27] International Organization for Standardization, "ISO 17025 Accreditation", 2017.
- [28] T. Evans, "What is ISO 17025?", Advisera. [Online]. Available: <https://advisera.com/17025academy/what-is-iso-17025/>. [Accessed: 09- May- 2022].
- [29] Departamento de Fisiología, Facultad de Medicina, UNAM, "Mecánica de la ventilación pulmonar. Espirometría", 2007.
- [30] Ricardo Duarte, Michel Mesnard, Jean-Pierre Nadeau. An innovative design approach to develop external articular medical devices.(2017).International Journal on Interactive Design and Manufacturing, Vo.11,pp.375–383."
- [31] X. Pardell, "Ventilación Mecánica", Apuntes de Electromedicina. [Online]. Available: <https://www.pardell.es/ventilacion-mecanica.html>. [Accessed: 22-oct-2023].
- [32] "Motor Paso A Paso Nema 17 – 36.8 oz.in – 200 Pasos/vuelta", Electronilab. [Online]. Available: <https://electronilab.co/tienda/motor-paso-a-paso-nema-17-25-oz-in-200-pasosvuelta/>. [Accessed: 03-nov-2023].
- [33] V. Garcia, "DESCRIPCIÓN DEL DRIVER A4988," Electrónica Práctica Aplicada, 05-Mar-2020. [Online]. Available: <https://www.diarioelectronicohoy.com/blog/descripcion-del-driver-a4988>. [Accessed: 03-Nov-2023].
- [34] "UNO R3," Arduino Documentation. [Online]. Available: <https://docs.arduino.cc/hardware/uno-rev3/>. [Accessed: 03-Nov-2023].

- [35] “Sensor de Presión Diferencial XGZP6897A,” CFSensor. [Online]. Available: <https://cfsensor.com/es/product/pressure-differential-sensor-xgzp6897a/>. [Accessed: 02-Feb-2024].
- [36] “USB-6009,” National Instruments. [Online]. Available: <https://www.ni.com/es-co/support/model.usb-6009.html>. [Accessed: 03-Nov-2023].
- [37] “Información general sobre ni-visa,” NI, <https://www.ni.com/es/support/documentation/supplemental/06/ni-visa-overview.html> [Accessed: 03-Feb-2024].
- [38] “Flow sensors. Proximal flow and pressure measurement,” Hamilton Medical. [Online]. Available: https://www.hamilton-medical.com/en_US/Products/Consumables/Flow-Sensors.html. [Accessed: 03-Nov-2023].
- [39] “Motor Paso A Paso Nema 23 – 178.5 oz.in – 200 Pasos/vuelta,” Electronilab. [Online]. Available: <https://electronilab.co/tienda/motor-paso-a-paso-nema-23-125-oz-in-200-pasos-vuelta/>. [Accessed: 03-Nov-2023].
- [40] “Driver Dual L298N para motores – Puente H,” Electronilab. [Online]. Available: <https://electronilab.co/tienda/driver-dual-para-motores-full-bridge-l298n/>. [Accessed: 03-Nov-2023].
- [41] “ESP-32,” SIGMA Electrónica. [Online]. Available: <https://www.sigmaelectronica.net/producto/esp-32/>. [Accessed: 03-Nov-2023].
- [42] “STM32F103C8,” STMicroelectronics. [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html>. [Accessed: 03-Nov-2023].
- [43] “HAFUHM0100L4AXT,” Mouser Electronics. [Online]. Available: <https://co.mouser.com/ProductDetail/Honeywell/HAFUHM0100L4AXT?qs=xch9k0p861cdFIwbsyrpag%3D%3D>. [Accessed: 03-Nov-2023].
- [44] “SFM3200-AW,” Sensirion. [Online]. Available: <https://www.sensirion.com/products/catalog/SFM3200-AW/>. [Accessed: 03-Nov-2023].
- [45] “USB-6210,” National Instruments. [Online]. Available: <https://www.ni.com/es-co/shop/model/usb-6210.html>. [Accessed: 03-Nov-2023].
- [46] G. Santos, “Cómo calibrar tu espirómetro con una jeringa certificada,” Serbinter. [Online]. Available: <https://serbinter.com/como-calibrar-tu-espirometro/>. [Accessed: 03-Nov-2023].
- [47] M. A. Veloz-Montenegro, J. G. Benalcázar-Game, y E. I. Domínguez-Bernita, “Algunas consideraciones sobre el examen de Pico Flujo y su medición”, *Dominio de las Ciencias*, vol. 3, pp. 177–187, mar. 2017.

6.1. Anexos

6.1.1. Clasificación de los dispositivos médicos

Para productos no invasivos:

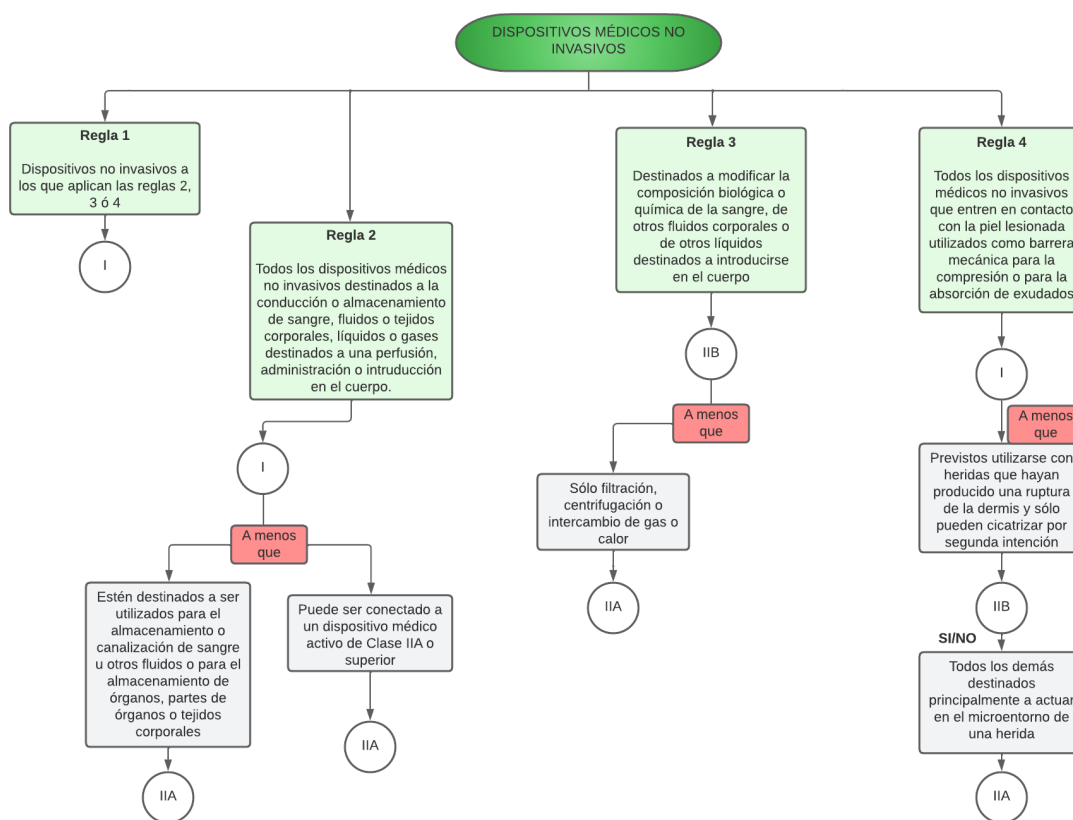


Figura 6.1: Dispositivos médicos no invasivos. Tomado de Estándares semánticos para los dispositivos médicos, I Fase de Análisis [26]

Como se puede observar en la figura 6.1, los equipos médicos que entran en esta categoría de no invasivos entran en su mayoría en la clasificación de riesgo I o máximo II seguido de una letra, las

cuales representan un riesgo menor con A hasta un riesgo mayor con D. Dentro de estos dispositivos, se encuentran aquellos que ayudan a la conducción o almacenamiento de fluidos hacia el cuerpo, aquellos destinados a modificar la composición de la sangre u otros líquidos destinados a introducirse en el cuerpo y aquellos que entren en contacto con la piel lesionada.

Para productos invasivos:

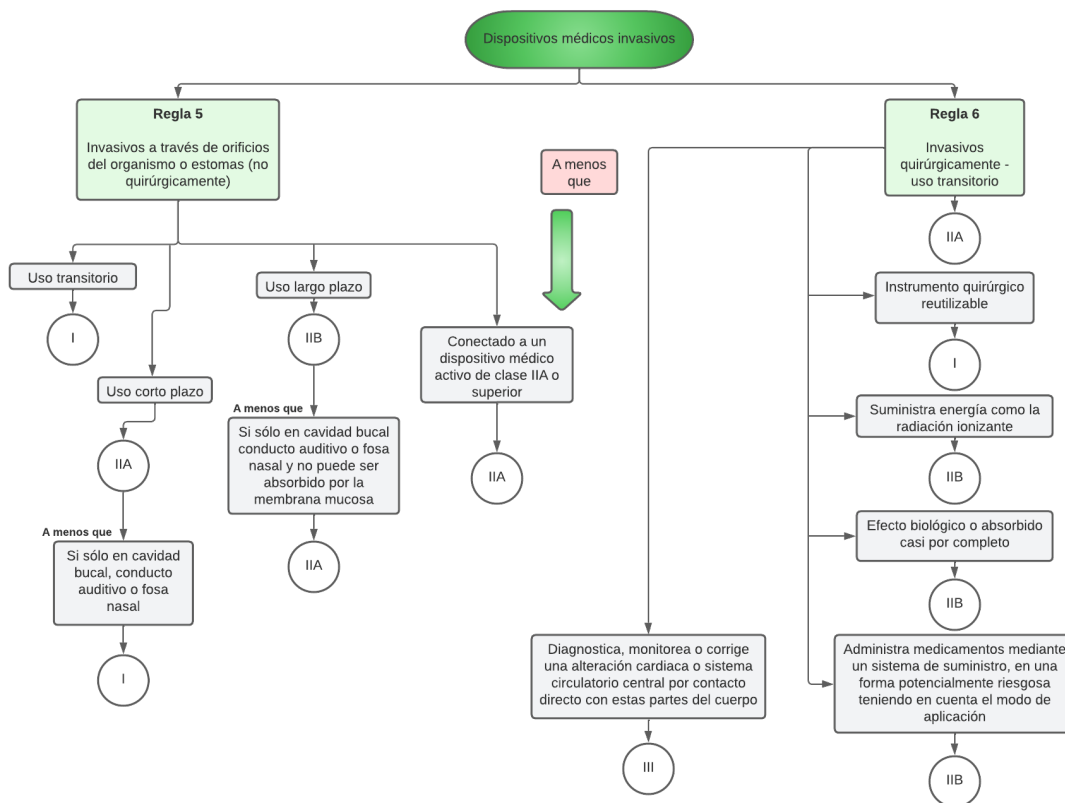


Figura 6.2: Dispositivos médicos invasivos. Tomado de Estándares semánticos para los dispositivos médicos, I Fase de Análisis [26]

En la figura 6.2 se mencionan los equipos invasivos. Para la mayoría de estos su clasificación de riesgo es mayor a la de los no invasivos, pero existen algunos casos en los que estos tienen una clasificación de riesgo bajo. Los dispositivos que se consideran como invasivos son aquellos que se usan a través de algún orificio del cuerpo o estoma y aquellos que se usan quirúrgicamente. [26]

Reglas adicionales aplicables a los productos activos:

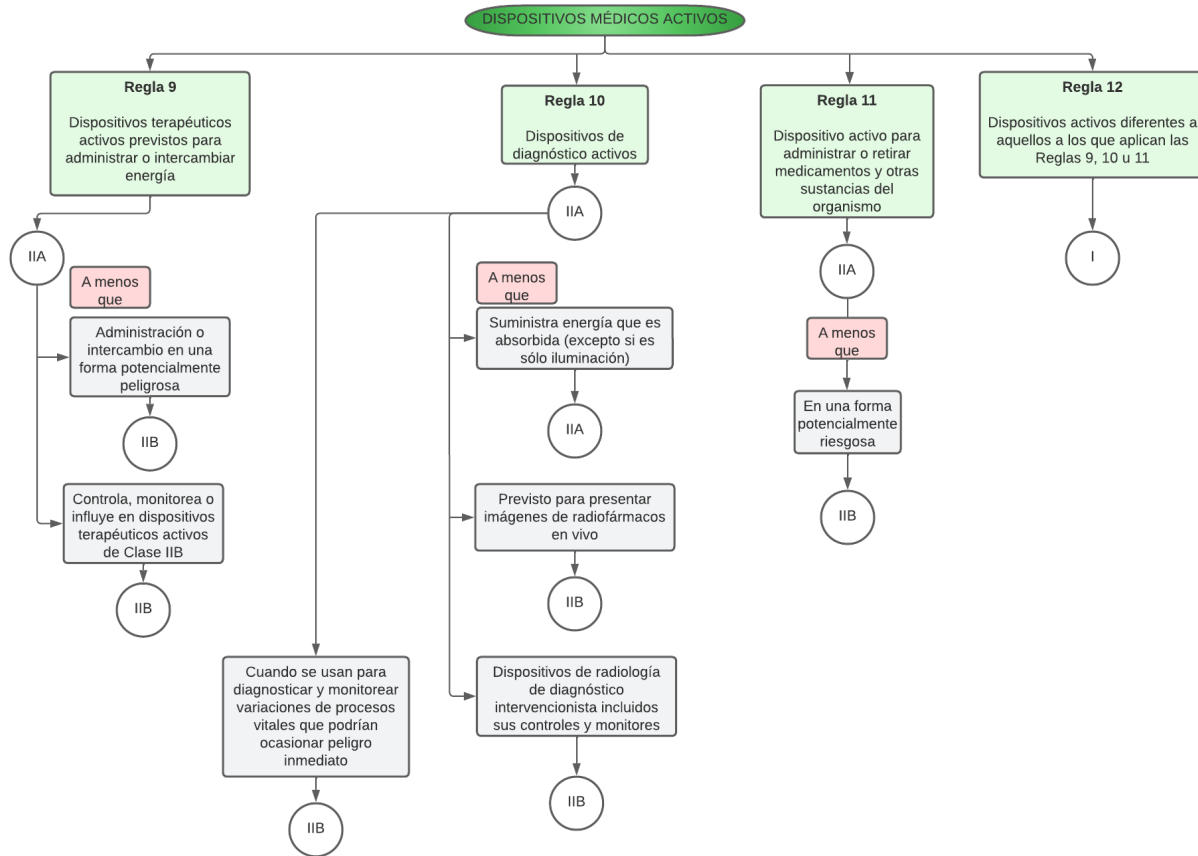


Figura 6.3: Dispositivos médicos activos. Tomado de Estándares semánticos para los dispositivos médicos, I Fase de Análisis [26]

En la figura 6.3 se pueden observar la clasificación de los dispositivos activos, estos son aquellos dispositivos cuyo funcionamiento dependa de una fuente de energía eléctrica o cualquier otra diferente a la producida por el cuerpo o la gravedad. Dentro de estos, ninguno cuenta con una clasificación de riesgo bajo y son aquellos usados para terapias, diagnósticos y para administrar o retirar medicamentos u otras sustancias del organismo.

Reglas especiales:

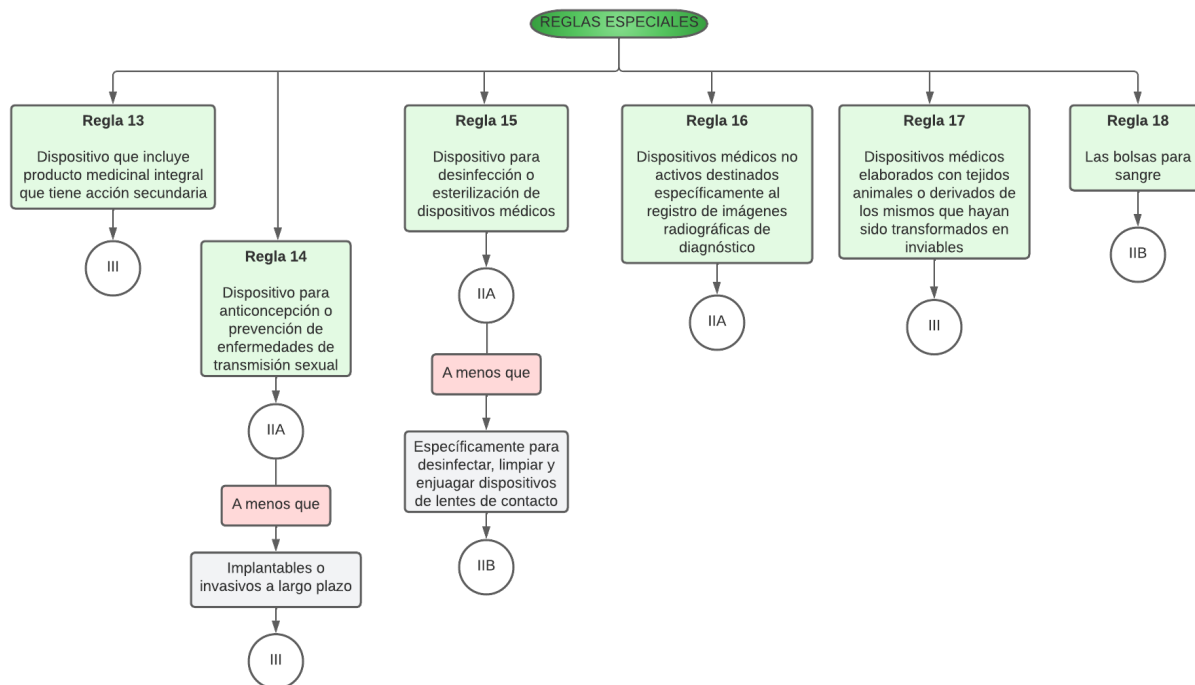


Figura 6.4: Reglas especiales. Tomado de Estándares semánticos para los dispositivos médicos, I Fase de Análisis [26]

Finalmente, en la figura 6.4 se mencionan los dispositivos los cuales cumplen con alguna regla especial y los cuales tienen la clasificación de riesgo más alta. Estos dispositivos son aquellos que incluyan producto de medicina integral, los usados para anticoncepción o prevención de enfermedades de transmisión sexual, los usados para desinfección o esterilización de dispositivos médicos, los que sean no activos usados para radiografías, los elaborados con tejidos animales o derivados de los mismos y las bolsas para sangre.

6.1.2. Alternativas mecánicas

Como se puede ver en las figuras 6.5 y 6.6 el primer modelo que se planteó para la implementación de la estación de calibración era una base de aluminio con orificios en la cual se acoplaría la jeringa y un actuador lineal. La jeringa estaría sujeta a la base con ayuda de las agarraderas de la izquierda las cuales se podrían ajustar en la base en cualquiera de los orificios con ayuda de un tornillo. Por otro lado, el actuador estaría acoplado a la jeringa con ayuda del acople de la derecha.

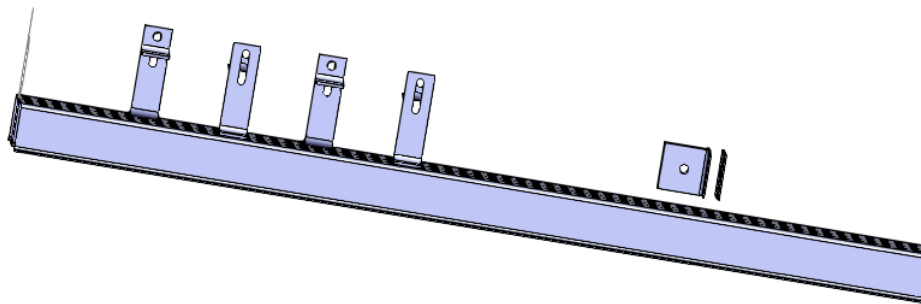


Figura 6.5: Vista lateral modelo con actuador lineal.

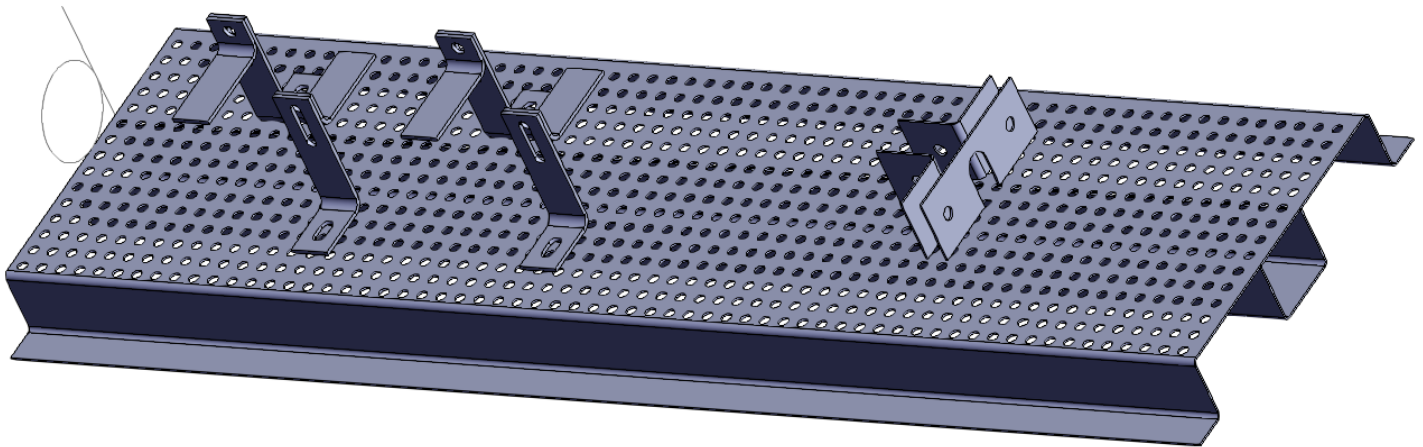


Figura 6.6: Vista diagonal modelo con actuador lineal.

En las figuras 6.7 y 6.8 se puede observar el modelo planteado con el tornillo sin fin. Para este modelo se planteó usar una estructura de aluminio, esta estructura sostendría la jeringa en la parte baja y en la parte superior estaría el motor con una base, el tornillo y un acople que movería el embolo de la jeringa a lo largo del tornillo y de dos ejes de acero.

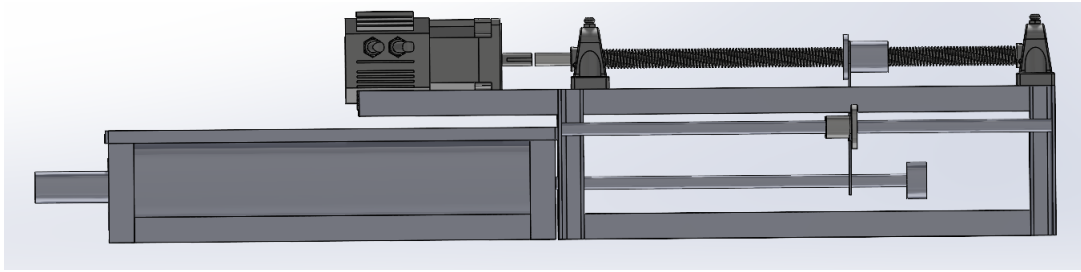


Figura 6.7: Vista lateral modelo con tornillo sin fin.

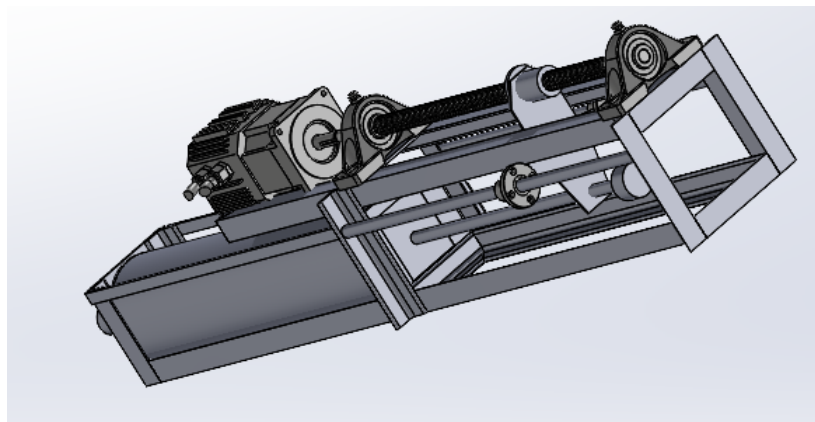


Figura 6.8: Vista diagonal modelo con tornillo sin fin.

6.1.3. Alternativas para los componentes

6.1.3.1. Alternativa Motor Paso a Paso

NEMA 23

El motor paso a paso NEMA 23 es una opción muy buena en caso de que el sistema necesite mover mas carga debido a que posee un torque mucho mas alto que el NEMA 17. Este motor es ampliamente usado para sistemas como impresoras 3D, maquinas CNC, cortadoras láser y actuadores lineales [39].

- **Precio:** Aproximadamente \$290.000.
- **Facilidad de acople:** Facilidad de implementación en todos los dispositivos.
- **Rangos de operación:** 14 Kg/cm de torque.
- **Precisión:** 1.8° que equivalen a 200 pasos por revolución.
- **Facilidad de comunicación:** Al ser un motor paso a paso unipolar o bipolar, se puede comunicar con las mismas librerías con las que se controla el motor actual.

6.1.3.2. Alternativa Driver

Driver DRV8825

El driver DRV8825 es el sustituto por defecto del A4988 debido a que el pinout de ambos es idéntico, lo que significa que las conexiones y las funciones son casi las mismas. Este driver también es usado para aplicaciones como impresoras 3D o maquinas CNC [33].

- **Precio:** Aproximadamente \$18.000.
- **Facilidad de acople:** Al tener los mismos pines que el A4988, su acople es el mismo.
- **Rangos de operación:** Requiere una duración mínima del pulso del paso de 1.9 us y entrega una corriente máxima de 2.2 A.
- **Precisión:** Ofrece 1/32 micro-pasos.

Driver L298N

El driver L298N posee dos canales de puente H, lo que significa que puede controlar dos motores eléctricos o un motor paso a paso unipolar o bipolar, controlando el sentido del motor y la velocidad. El control del sentido del motor se realiza mediante dos pines para cada motor y la velocidad se controla mediante la modulación por ancho de pulso (PWM) [40].

- **Precio:** Promedio \$15.000.
- **Facilidad de acople:** Facilidad de implementación para la mayoría de microcontroladores.
- **Rangos de operación:** Capacidad de corriente de 2 A hasta 3 A.

6.1.3.3. Alternativa Microcontrolador

Microcontrolador ESP-32

El microcontrolador ESP-32 es ideal para aplicaciones más robustas que necesiten de una comunicación por WIFI o bluetooth, aparte de estas dos opciones también cuenta con comunicación serial mediante USB. Adicional a esto, incluye seguridad criptográfica por hardware, un coprocesador para el modo bajo consumo e interfases periféricas como: I2C, SPI, CAN, PWM, entre otros [41].

- **Precio:** Promedio \$30.000.
- **Facilidad de acople:** 34 pines digitales y 18 canales analógicos.
- **Rangos de operación:** SRAM de 520 KiB.

- **Precisión:** 12 bits de resolución.
- **Facilidad de comunicación:** Se puede comunicar con el computador por medio de USB, WIFI o Bluetooth.

Microcontrolador STM32F103C8T6

El microcontrolador STM32F103C8T6 es una placa de desarrollo similar al Arduino con un núcleo RISC con una frecuencia de funcionamiento de 72 MHz, memorias integradas de alta velocidad, una amplia selección de entradas y salidas mejoradas y periféricos conectados a dos buses APB. En adición a esto, cuenta con un conjunto de modos de ahorro de energía, por lo que es ideal para aplicaciones de bajo consumo [42].

- **Precio:** Promedio \$43.000.
- **Facilidad de acople:** 37 pines de entradas y salidas de los cuales 10 son canales ADC.
- **Rangos de operación:** SRAM de 20 kB.
- **Precisión:** Terminales ADC con 12 bits de resolución.
- **Facilidad de comunicación:** Cuenta con los siguientes protocolos de comunicación: CANbus, I2C, IrDA, LINbus, SPI, UART/USART y USB.

6.1.3.4. Alternativas Sensor

Sensor HAFUHM0100L4AXT

El HAFUHM0100L4AXT de la marca honeywell es un sensor de flujo de alta precisión unidireccional, ofrece una conexión enroscada por puerto ISO 1179, formato de salida por protocolo I2C y proporciona una interfaz digital para la lectura del caudal de aire en rangos especificados de caudal a escala completa y temperatura compensada. El calentador aislado térmicamente y los elementos de detección de temperatura ayudan a este sensor proporciona una respuesta rápida al flujo de aire o gas. Está diseñado para medir el caudal másico de aire y otros gases no corrosivos [43].

- **Precio:** Aproximadamente \$1'320.000
- **Facilidad de acople:** Posee una conexión enroscada por puerto ISO 1179 lo que hace que tenga un tipo de conexión estándar para uso general.
- **Rangos de operación:** 0 - 100 L/min.
- **Precisión:** 0.5 %
- **Facilidad de comunicación:** Se puede comunicar por protocolo I2C.

Sensor SFM3200-AW

El caudalímetro másico SFM3200-AW de la marca SENSIRION, es el más adecuado para aplicaciones médicas, en específico para mediciones de flujo inspiratorio. Este sensor posee un amplio rango dinámico de mediciones de hasta 250 SLM y una alta precisión para variaciones de flujo bajo, el diseño especial del canal de flujo da como resultado una caída de presión muy baja a través del cuerpo de flujo del sensor, lo que lo hace extremadamente adecuado para ventilación médica y otras aplicaciones respiratorias. Además, el sensor térmico garantiza un tiempo de procesamiento de la señal muy rápido y un modo de medición bidireccional, con la señal linealizada internamente y compensada por temperatura [44].

- **Precio:** Aproximadamente \$850.000.
- **Facilidad de acople:** Debido a que es especial para aplicaciones respiratorias, su acople viene estandarizado para este tipo de aplicaciones.
- **Rangos de operación:** -100 a 250 SLM.
- **Precisión:** $\pm 2\%$ m.v.
- **Facilidad de comunicación:** Se puede comunicar por medio de protocolo I2C.

6.1.3.5. Alternativas DAQ

DAQ NI-6210

El módulo DAQ NI-6210 cuenta con 16 pines de entradas analógicas, 4 pines de entradas digitales, 4 pines de salidas digitales, 1 pin para fuente de 5 V, 3 pines para conexión a tierra, 1 pin de reserva y 1 pin de censado. También cuenta con 8 terminales diferenciales, un rango de -10 V a 10V para las entradas analógicas y una frecuencia máxima de la fuente de 80 MHz para contadores y temporizadores [45].

- **Precio:** Aproximadamente \$4'700.000.
- **Facilidad de acople:** Cuenta con los pines necesarios con terminales RSE para el sensor seleccionado.
- **Rangos de operación:** 250KS/s.
- **Precisión:** 16 bits de resolución.
- **Facilidad de comunicación:** Se puede conectar al computador por USB.

6.1.4. Manual de usuario

La estación de calibración es un sistema que tiene como objetivo ayudar a calibrar los diferentes tipos de dispositivos usados para terapias de respiración, como lo son los inspirómetros de incentivos. Para un funcionamiento óptimo del sistema se debe seguir el siguiente paso a paso:

1. Se deben conectar los cables USB del Arduino UNO y del modulo DAQ a los puertos USB del computador (1) como se puede ver en la figura 6.9.

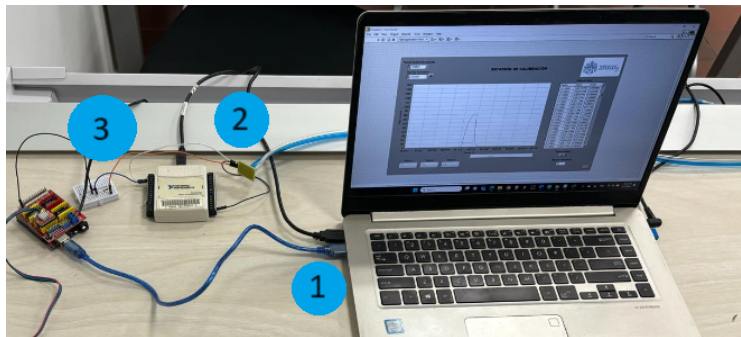


Figura 6.9: Conexiones del sistema.

2. Luego de conectar el Arduino y la DAQ al computador, se deben conectar los cables que salen del Arduino (1) como se ve en la figura 6.10 al motor de tal manera que en el motor queden en el siguiente orden: De izquierda a derecha, el cable negro, azul, verde y rojo. Como se ve en el figura 6.11.

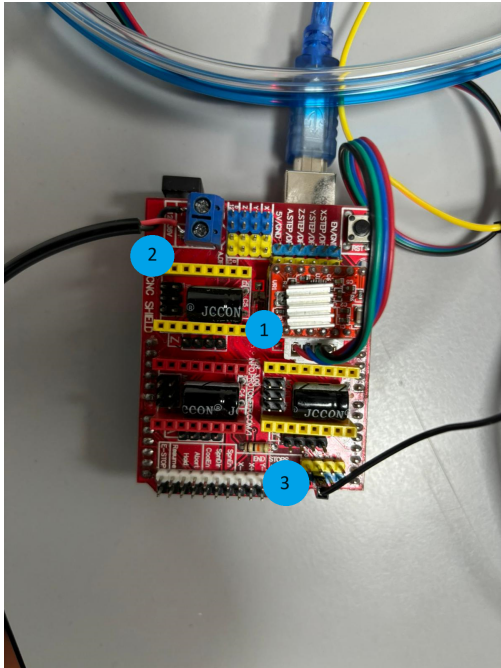


Figura 6.10: Conexiones del motor en el Arduino.

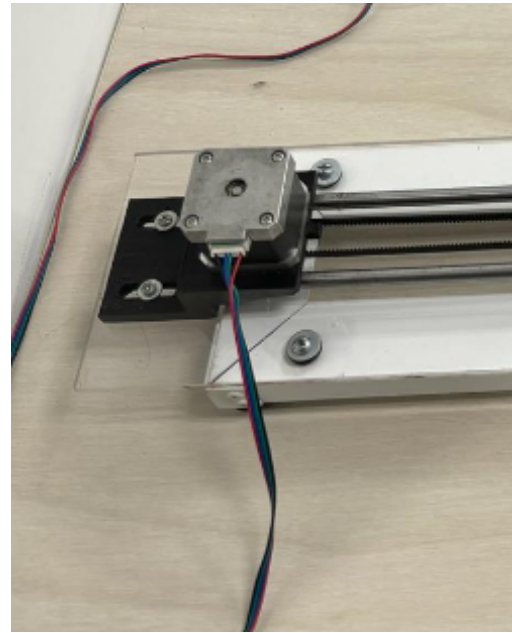


Figura 6.11: Conexiones del motor.

3. Una vez conectado el motor al Arduino se necesita energizarlo, para esto se debe conectar el adaptador negro que sale del Arduino (2) como se ve en la figura 6.10, a un toma corriente.
4. Luego de energizar el motor se acopla el sensor Hamilton conectando el orificio de mayor diámetro a la salida de la jeringa con ayuda de un tubo de plástico mas largo para su acople como se aprecia en la figura 6.12. Los tubos que salen del sensor Hamilton se conectan a la placa del sensor de presión diferencial (2) conectando el tubo azul a la entrada mas próxima a los pines y el tubo transparente a la entrada sobrante como se ve en la figura 6.13.



Figura 6.12: Conexiones del Sensor.

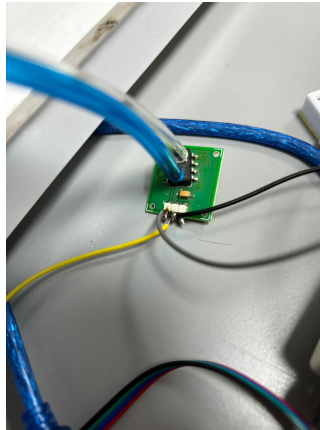


Figura 6.13: Conexiones del Sensor en la placa de presión.

5. Una vez acoplado el sensor, se conectan los pines de la placa del sensor XGZP6891A a el módulo DAQ y al filtro RC (3). Para esto se conecta el cable negro al puerto 1 del DAQ, el cable amarillo al puerto 31 de la DAQ y el cable gris a la entrada del filtro RC como se ve en la figura 6.14, luego de esto con un jumper (en este caso el azul) se conecta la salida del filtro con el puerto 2 del módulo DAQ y con otro la tierra del filtro al pin GND en el Arduino (3) como se ve en la figura 6.10.

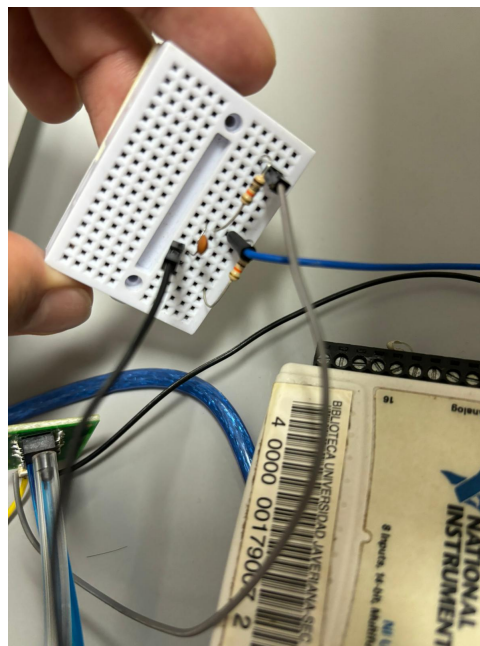


Figura 6.14: Conexiones del filtro.

6. Para iniciar la interfaz se debe de abrir el archivo “Estación de Calibración - HMI.vi”, en el software LabView.
7. Una vez abierto, se debe seleccionar el puerto en el que se conectó el Arduino al computador por medio de un caja desplegable en la que saldrán todos los puertos disponibles. Como se ve en la figura 6.15.

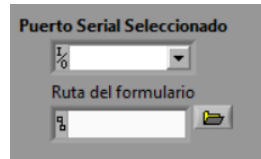


Figura 6.15: Selección de puertos.

8. Luego de seleccionar el puerto se debe seleccionar la ruta en donde se encuentra el archivo “Formato_Calibracion_Template”, el cual corresponde a la plantilla para el formato de calibración. Para seleccionarlo se da click izquierdo en la carpeta a la derecha de la caja de ruta de formulario que se ve en la figura 6.15, lo que abrirá una ventana con los archivos del PC, en esta se busca la ruta donde se tenga guardada la plantilla y se selecciona.
9. Una vez seleccionado el puerto y la ruta de la plantilla se inicializa la interfaz dándole click izquierdo en la flecha que se encuentra en la esquina superior izquierda de la ventana de la interfaz como se puede ver en la figura 6.16.

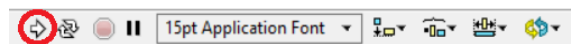


Figura 6.16: Inicio de la interfaz.

10. Luego de iniciar la interfaz esta empezará a tomar los datos así no se haya seleccionado ninguna rutina. Para seleccionar una rutina se debe dar click izquierdo a cualquiera de los tres botones que se encuentran en la esquina inferior izquierda de la interfaz como se ve en la figura 6.17.

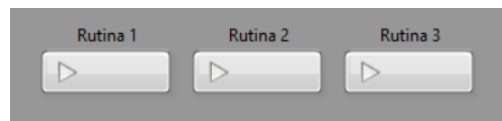


Figura 6.17: Botones para seleccionar rutinas.

11. Para tomar los datos que se están midiendo se debe dar click izquierdo en el botón "Toma de datos" que se encuentra en la esquina inferior derecha como se ve en la figura 6.18.

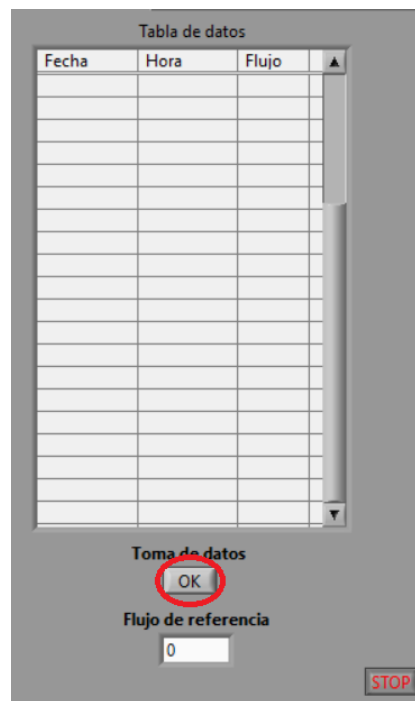


Figura 6.18: Botón para tomar datos.

12. Una vez se da click en este botón, se abrirá una ventana que permite cambiar el nombre del formato de calibración generado y el lugar donde se guardará este archivo.

13. Si se quiere volver a tomar otro dato se recomienda apagar la interfaz, limpiar los datos de la tabla y la gráfica y repetir los pasos a partir del número 9.

14. Para apagar la interfaz se debe dar click izquierdo en el botón de stop que se encuentra en la esquina inferior derecha como se puede ver en la figura 6.19.



Figura 6.19: Botón para apagar la interfaz.

15. Para borrar los datos de la tabla se debe dar click derecho sobre la tabla y seleccionar la opción de Data Operations y dentro de las opciones que se despliegan seleccionar Empty Table haciendo click izquierdo sobre esta como se ve en la figura 6.20.

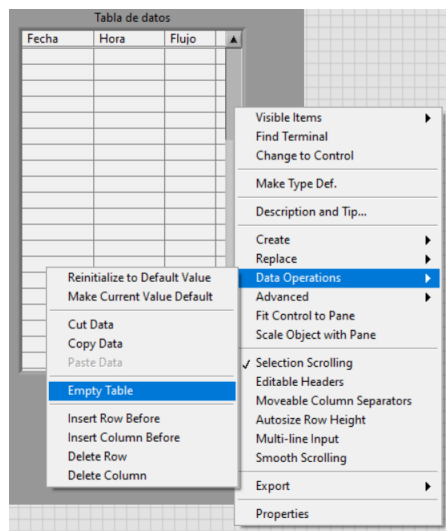


Figura 6.20: Botón para borrar datos de la tabla.

16. Por ultimo, para limpiar la gráfica se debe dar click izquierdo sobre esta y seleccionar la opción de Data Operations y dentro de las opciones que se despliegan seleccionar Clear Chart haciendo click izquierdo sobre esta como se ve en la figura 6.21.

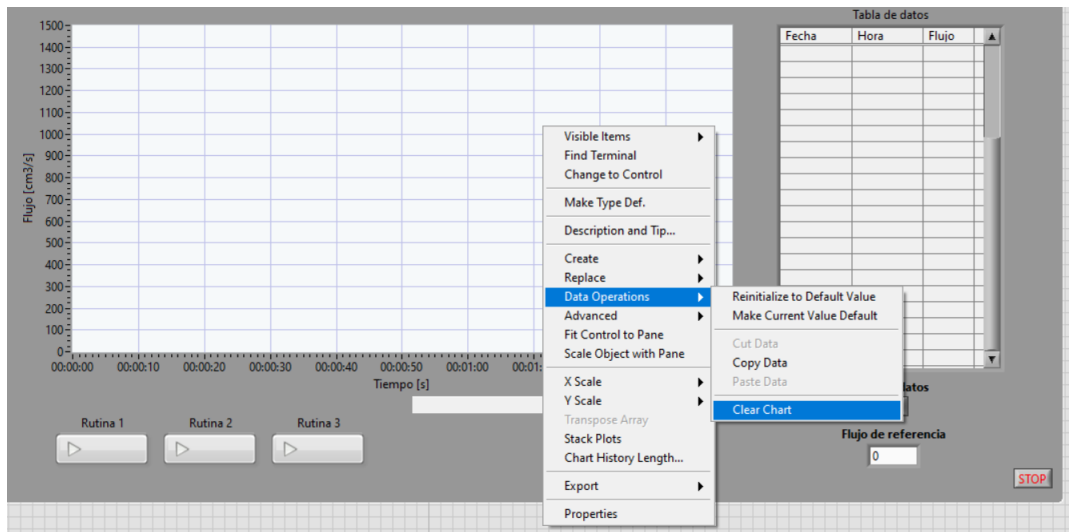


Figura 6.21: Botón para borrar datos de la gráfica.

6.1.5. Protocolo de pruebas

Para realizar la validación de un inspirómetro se recomienda seguir las siguientes indicaciones:

- Antes de conectar el equipo a validar, realizar pruebas de funcionamiento de la estación de calibración, siguiendo el paso a paso mencionado anteriormente en el manual de usuario y verificando que tanto el motor y el sistema estén funcionando correctamente.
- Realizar las validaciones únicamente en un laboratorio o en un ambiente limpio para evitar la contaminación de la estación de calibración y del dispositivo a probar.
- Usar un filtro a la salida de la estación de calibración para evitar la contaminación de los dispositivos a validar.
- Realizar una limpieza de todos los componentes de la estación de calibración luego de cada uso.

Al momento de realizar una validación se debe seguir el siguiente paso a paso:

1. Conectar la estación de calibración siguiendo el paso a paso mostrado en el manual de usuario.
2. Conectar el filtro entre la salida del sensor Hamilton de la estación de calibración y el dispositivo a validar.

3. Conectar el dispositivo a validar a la salida del filtro.
4. Seleccionar la rutina con la que se desea validar el dispositivo.
5. Generar el formato de calibración una vez que el sistema mecánico se encuentre detenido y recordar apagar la interfaz para evitar que el sistema siga tomando datos no deseados.
6. Realizar cinco (5) pruebas, es decir, seleccionar la rutina deseada, dejarla finalizar y generar el formato de calibración por cada prueba.
7. Una vez finalizada la validación, desmontar todas las conexiones y guardarlas en un lugar seguro, fresco y limpio.