



FACULTAD DE INGENIERÍA

UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE ELECTRÓNICA

Tesis de grado en Ingeniería Electrónica

SISTEMA AUTOMÁTICO DE CALIBRACIÓN DE ANEMÓMETROS EN EL TÚNEL DE VIENTO DEL SERVICIO METEOROLÓGICO NACIONAL

Autor: Cristian Zozimo Aranda Cordero (93631)

Directores: Ing. Leonardo Carducci (FIUBA)

Dr. Ing. Lucas Sambuco (FIUBA)

Ing. Claudio Arencibia (SMN)

Jurado: Nombre y apellido

Nombre y apellido

Nombre y apellido

Buenos Aires, 12 de agosto de 2024

Resumen

En este trabajo se desarrolló un sistema automatizado para la calibración de instrumentos de medición de viento de tipo ultrasónico, utilizando el túnel de viento del Servicio Meteorológico Nacional. El sistema incluye un datalogger desarrollado con la placa EDU-CIAA, responsable de la adquisición de datos de viento de los sensores, así como de la comunicación y el control del motor del túnel, con un controlador de lazo cerrado PID, reemplazando el sistema manual de medición. Además, se desarrolló una aplicación web con el framework Django y un servidor WebSocket con Django Channels para configurar el datalogger, cargar información relevante de los sensores y del túnel de viento, presentar las mediciones en tiempo real, y procesar los datos utilizando un modelo por comparación, para calcular la incertidumbre y la corrección de las mediciones. Toda la información se guarda de manera estructurada en una base de datos. Este trabajo permitirá al laboratorio del Servicio Meteorológico Nacional alcanzar la capacidad de calibrar sensores de viento, satisfaciendo así la demanda de la red de estaciones automáticas y los sistemas de anemómetros instalados en el país.

Abstract

This work developed an automated system for calibrating ultrasonic wind measurement instruments using the wind tunnel of the National Meteorological Service. The system includes a datalogger developed with the EDU-CIAA board, responsible for acquiring wind data from the sensors, as well as for communication and control of the tunnel motor with a closed-loop PID controller, replacing the manual measurement system. Additionally, a web application was developed using the Django framework and a WebSocket server with Django Channels to configure the datalogger, load relevant information about the sensors and the wind tunnel, display real-time measurements, and process the data using a comparative model to calculate uncertainty and measurement corrections. All information is stored in a structured manner in a database. This work will enable the laboratory of the National Meteorological Service to achieve the capability to calibrate wind sensors, thus meeting the demand of the network of automatic stations and anemometer systems installed across the country.

Agradecimientos

Índice general

1. Introducción	1
1.1. Introducción	1
1.2. Estado del arte	1
1.3. Motivación	2
1.4. Objetivos	3
2. Viento	5
2.1. Viento en superficie	6
2.2. Instrumentos de medición del viento	9
2.3. Túnel de viento	12
3. Cálculo de Incertidumbre	18
3.1. Incertidumbre de la medición	18
3.2. Calibración y trazabilidad	20
3.3. Fuentes de incertidumbre	22
3.4. Modelo teórico de medición	24
4. Implementación del Datalogger	26
4.1. Desarrollo del hardware	27
4.1.1. Módulo de alimentación eléctrica	28
4.1.2. Módulo de adquisición de datos de viento	30
4.1.3. Módulo de comunicación Ethernet/SPI	32
4.1.4. Circuito PWM	33
4.1.4.1. Simulación del circuito	35
4.1.4.2. Implementación del circuito	38
4.1.5. Circuito de adquisición de tensión del variador	44
4.1.5.1. Simulación del circuito	44
4.1.5.2. Implementación del circuito	46

4.1.6. Circuito de LEDs indicadores	47
4.2. Diseño y construcción del PCB	49
4.3. Desarrollo del firmware	56
4.3.1. Configuración y conexión con los servidores	59
4.3.1.1. Servidor NTP	61
4.3.1.2. Servidor WebSocket	63
4.3.2. Muestreo, adquisición y procesamiento	67
4.3.3. Transmisión al servidor	72
4.3.4. Sistema de control PID	73
5. Implementación de la Aplicación WEB	77
5.1. Desarrollo aplicación WEB	77
5.1.1. Arquitectura del software	78
5.1.2. Back End	80
5.1.2.1. Base de datos	81
5.1.2.2. Servidor WebSocket	83
5.1.2.3. Generador de trayectoria y control del túnel	85
5.1.2.4. Cálculo de incertidumbre	87
5.1.3. Front End	89
5.1.3.1. Carga de metadatos	91
5.1.3.2. Configuración del sistema	92
5.1.3.3. Adquisición de datos	93
5.1.3.4. Procesamiento de datos	94
5.1.3.5. Resultados	96
5.2. Integración con el hardware	98
6. Mediciones y Resultados Experimentales	101
6.1. Caracterización de la zona de medición	102
6.2. Calibración DeltaOHM HD51.3	105
6.2.1. Banco de medición y configuración del hardware	106
6.2.2. Configuración de la aplicación web	107
6.2.3. Resultados	109
7. Conclusiones	113
7.1. Conclusiones finales	113
7.2. Trabajo futuro	114

A. Datalogger y sensores	115
A.1. Principio de medición para sensores de ultrasonido	115
A.2. Especificaciones técnicas y hojas de datos	117
B. Software	118
B.1. Algoritmo del generador de trayectoria	118
B.2. Comandos entre el servidor WebSocket y el datalogger	120
C. Cálculo de incertidumbre	121
C.1. Ejemplo de Presupuesto de incertidumbre para un punto	122

Índice de figuras

2.1. Tres componentes de la velocidad de viento v_x, v_y, v_z [m s^{-1}]. El vector de viento horizontal v_{xy} forma un ángulo de entrada θ con el vector de viento resultante v_{xyz} .	5
2.2. Campo de viento representado con flechas en el sur de Sudamérica. [18]	7
2.3. Representación gráfica del viento utilizando barbas. [18]	7
2.4. Representación del viento en 850 hPa para el día 14/4/2016 a las 12 UTC en la región central de Argentina. [18]	8
2.5. Viento (nudos) y líneas de corriente en 500hPa. [17]	9
2.6. En (a) el sensor de coperola y en (b) el sensor de veleta. [21]	10
2.7. Anemómetro de hilo caliente. [22]	10
2.8. Tubo Pitot. [22]	11
2.9. Anemómetro de ultrasonido. [16]	11
2.10. Anemómetro láser Doppler. [13]	12
2.11. Esquema del túnel de viento.	13
2.12. Vista general del túnel de viento cerrado del SMN.	13
2.13. En (a) se muestra la puerta cerrada de la zona de medición, en (b) se muestra la puerta abierta y en (c) se muestra la varilla telescopica.	14
2.14. En (a) se muestra la vista lateral izquierda de la zona de medición, en (b) se muestran los álabes instalados en la parte derecha de la zona de medición, en (c) una vista frontal de la zona de medición.	15
2.15. En (a) se muestra los protectores auditivos y en (b) se muestra la mesa de trabajo.	15
2.16. En (a) se muestra la vista lateral izquierda del motor, en (b) la vista lateral derecha, en (c) se muestra las correas conectadas a la polea superior y en (d) se muestra el eje del motor con la polea inferior y las correas.	16
2.17. En (a) se muestra el panel de control manual y en (b) el tablero eléctrico del variador de velocidad.	17

3.1. Diagrama de las fuentes que producen incertidumbre en la medición de un mensurando.	19
3.2. Trazabilidad metrológica.	22
4.1. Diagrama en bloques del sistema electrónico.	26
4.2. Placa de desarrollo EDU CIAA	27
4.3. Diagrama en bloques de la EDU-CIAA	27
4.4. En (a) se muestra el sensor Vaisala modelo WMT700, en (b) se muestran el sensor Delta OHM modelo HD51.3D.	28
4.5. Regulador de voltaje DC-DC basado en el chip LM2596.	29
4.6. Esquemático del sistema de alimentación con módulo LM2596, diodo Zener, divisor resistivo y filtros de bypass para el LM358.	30
4.7. Esquema de conexión de datos y alimentación del sensor HD51.3D con un Data-logger o un PLC. [8]	31
4.8. Puerto RS-485 integrado en la EDU-CIAA	31
4.9. Módulo convertidor de RS485 a UART (TTL)	31
4.10. Esquemático del módulo MAX485 para la recepción de datos del sensor bajo calibración.	32
4.11. Módulo W5100 utilizado para realizar la conectividad a internet a través de la interfaz SPI del microcontrolador.	33
4.12. Esquemático del módulo W5100 para la comunicación bidireccional con distintos servidores.	33
4.13. Esquemático circuital del tablero de control del tunel de viento.	34
4.14. Circuito diseñado para obtener distintos niveles de tensión mediante la variación del ciclo de trabajo de una señal cuadrada (PWM).	36
4.15. Señales a la salida del primer filtro LP (curva verde) y segundo filtro LP (curva azul).	36
4.16. Señales simuladas para un ciclo de trabajo del 100 %.	37
4.17. Señales simuladas para un ciclo de trabajo al 75 % de la señal de entrada.	38
4.18. Señales simuladas para un ciclo de trabajo al 55 % de la señal de entrada.	38
4.19. Esquemático del circuito PWM que controla la velocidad del túnel de viento por programación.	39
4.20. Banco de medición para caracterizar el circuito PWM.	39
4.21. Medición de la señal PWM (rojo) con un ciclo de trabajo de (1%) y la señal continua a la salida del amplificador operacional (azul) iguala 53,5 mV.	40

4.22. Medición de la señal PWM (rojo) con un ciclo de trabajo de (25 %) y la señal continua a la salida del amplificador operacional (azul) iguala 900 mV.	40
4.23. Medición de la señal PWM (rojo) con un ciclo de trabajo de (50 %) y la señal continua a la salida del amplificador operacional (azul) iguala 1,8 V.	41
4.24. Medición de la señal PWM (rojo) con un ciclo de trabajo de (75 %) y la señal continua a la salida del amplificador operacional (azul) iguala 2,65 V.	41
4.25. Medición de la señal PWM (rojo) con un ciclo de trabajo de (100 %) y la señal continua a la salida del amplificador operacional (azul) iguala 3,56 V.	42
4.26. Circuito diseñado para tomar muestras de la señal que se suministra al variador de velocidad y se las envía a una canal analógico-digital de la EDU-CIAA.	44
4.27. Señales simuladas a la salida del variador de velocidad y a la entrada del pin analógico-digital de la EDU-CIAA para un ciclo de trabajo del 100 %.	45
4.28. Señales simuladas a la salida del variador de velocidad y a la entrada del pin analógico-digital de la EDU-CIAA para un ciclo de trabajo del 55 %.	45
4.29. Esquemático de la implementación del circuito que toma muestras de la señal continua que ingresa al variador de velocidad.	46
4.30. Esquemático de los circuitos que encienden y apagan dos LEDs, verde y amarillo, cada vez que se recibe un dato de los anemómetros, tanto del patrón como del que está bajo calibración.	48
4.31. Esquemático del circuito de un LED dual que se enciende en rojo cuando el sistema no está conectado al servidor Websocket y en verde cuando hay una conexión estable con el servidor.	49
4.32. Template disponible para la realización de ponchos para la EDU-CIAA.	50
4.33. Diseño del esquemático que integra los módulos desarrollados.	51
4.34. Diseño PCB del Shield (poncho) que contiene todos los módulos descritos en las secciones anteriores.	52
4.35. Vista en 3D de la capa BOTTOM.	53
4.36. Vista en 3D de la capa TOP.	54
4.37. Shield conectado a la placa EDU-CIAA	55
4.38. Otra vista del Shield diseñado para esta aplicación.	55
4.39. Vista superior de la placa Shield conectada a la EDU-CIAA	56
4.40. Lanzador de aplicaciones integradas para el desarrollo de firmware para la EDU-CIAA.	57
4.41. Estructura de carpetas, del proyecto Datalogger	57

4.42. Lista de regiones con su respectiva prioridad.	58
4.43. Modos de configuración para el módulo W5100.	60
4.44. Diagrama de flujo para establecer una comunicación, en modo cliente, con un servidor utilizando sockets.	61
4.45. Máquina de estados para configurar el RTC del datalogger a través de un servidor NTP.	63
4.46. Máquina de estados para establecer una conexión bidireccional con un servidor WebSocket.	65
4.47. Máquina de estados para mantener la conexión con el servidor y elevar un raise en caso de que se reciba información del servidor.	66
4.48. Máquina de estados para procesar los mensajes recibidos del servidor WebSocket.	67
4.49.	69
4.50.	70
4.51. Máquina de estados para acumular muestras en una tabla y realizar un preprocesamiento de esos datos.	72
4.52. Máquina de estados para gestionar la transmisión de datos al servidor WebSocket.	73
4.53. Máquina de estados que ejecuta el control PID con valores de referencia enviados por el servidor.	76
5.1. Arquitectura del software implementada.	79
5.2. Patrón de diseño de software MVT, que divide la lógica del programa en tres elementos interconectados.	80
5.3. Diagrama entidad relación diseñada para la aplicación Web.	82
5.4. Segmento para pasar de un punto de velocidad v^B a otro v^C viiniendo desde v^A	86
5.5. Trayectoria generada utilizando el método de Paul para un ciclo ascendente y descendente de velocidades.	87
5.6. Diagrama de flujo del para el calculo de incertidumbre expandida.	87
5.7. Pantalla de inicio de la aplicación web.	90
5.8. Vista para confirmar si se desea realizar o no una calibración.	90
5.9. Barra lateral izquierda de navegación.	91
5.10. Vista para cargar los metadatos de la calibración.	92
5.11. Vista para configurar los equipos involucrados en la calibración.	92
5.12. Configuración de tiempos para el generador de trayectoria en 5 ms^{-1}	93
5.13. Vista de la adquisición de datos.	94
5.14. Vista para confirmar si se desea borrar o no los datos crudos de una calibración.	94

5.15. Vista para el procesamiento de datos y visualización de gráficos.	95
5.16. Vista para la revisión de datos parseados y cálculo de incertidumbre.	96
5.17. Curva de histéresis.	97
5.18. Tabla de valores promedio e histéresis.	97
5.19. Curva de calibración.	98
5.20. Tabla de correcciones, incertidumbre y grados de libertad.	98
5.21. Sistema integrado desarrollado en esta tesis.	100
 6.1. Sistema de referencia utilizado para realizar las mediciones adentro del túnel de viento.	101
6.2. En (a) se muestra el anemómetro midiendo a -20 cm, en (b) se muestra midiendo a 20 cm y en (c) se encuentra midiendo a 60 cm.	102
6.3. Velocidad de viento en cinco puntos, para distintas posiciones.	103
6.4. Nivel de PWM medido en el controlador, para distintas posiciones, obtenidas utilizando el software y hardware desarrollados en capítulos anteriores.	104
6.5. Niveles de PWM para distintas posiciones del anemómetro a 25 m s^{-1}	104
6.6. Niveles similares de PWM para distintas posiciones del anemómetro.	105
6.7. En (a) se muestra la vista frontal de los sensores patrón y bajo prueba, en (b) se observa una vista superior para ver la alineación, y en (c) se muestra una vista lateral.	106
6.8. Brazo con los sensores bajo calibración a izquierda y patrón a derecha dentro del túnel.	107
6.9. Mesa de trabajo con el datalogger y los insumos necesarios para su funcionamiento.	107
6.10. En (a), (b), (c) y (d) se muestran los metadatos ingresados el por el usuario para realizar la calibración.	108
6.11. En (a) y (b) se muestra las configuraciones realizada por el operador para el datalogger y el túnel de viento.	108
6.12. En (a) se muestra el ciclo ascendente y descendente de la velocidad de viento y en (b) se observa la dirección de viento.	109
6.13. En (a) se muestra la curva de histéresis para el sensor bajo calibración y en (b) se observa la tabla con los resultados numéricos.	110
6.14. En (a) se presenta la curva de calibración ascendente con sus respectivas incertidumbres expandidas, y en (b) se muestra la tabla con los resultados numéricos.	111
6.15. En (a) se presenta la curva de calibración descendente con sus respectivas incertidumbres expandidas, y en (b) se muestra la tabla con los resultados numéricos.	112

A.1. Esquema de los transductores y el tiempo de viaje de un pulso ultrasónico. [8] . . . 115

Índice de tablas

3.1. Fuentes de incertidumbre evaluadas para el presupuesto de incertidumbre de un anemómetro bajo calibracion en el tunel de viento del SMN.	23
4.1. Mediciones de tensión y corriente de la fuente del variador para distintas resistencias de carga.	35
4.2. Mediciones para ciclos de trabajo en modo Ascendente, VadcTunnel y la velocidad del viento.	43
4.3. Mediciones para ciclos de trabajo en modo Descendente, VadcTunnel y la velocidad del viento.	43
4.4. Mediciones a la entrada pin <i>VADC</i> del variador y en el pin ADC_CH2 de la EDU-CIAA, para un barrido ascendente del ciclo de trabajo.	47
4.5. Mediciones a la entrada pin <i>VADC</i> del variador y en el pin ADC_CH2 de la EDU-CIAA, para un barrido descendente del ciclo de trabajo.	47
4.6. Diferencias entre los tipos de máquinas de estados.	58
A.1. Especificaciones de sensor de viento, marca Delta Ohm, modelo HD51.3.	117
B.1. Listado de comandos transmitidos por el servidor WebSocket al datalogger para configurar los parámetros seleccionados por el usuario.	120
C.1. En la tabla se indica el calculo de incertidumbre para un punto en particular 10 m s^{-1}	122

Capítulo 1

Introducción

1.1. Introducción

Los anemómetros ultrasónicos o mecánicos son instrumentos capaces de medir el viento, una magnitud vectorial con dirección e intensidad. La medición del viento es crucial en una variedad de aplicaciones, tales como estudios de contaminación atmosférica, vigilancia y predicción meteorológica, aterrizaje de aeronaves, análisis climático en función de la carga de viento, evaluación de daños causados por el viento, estimación del potencial de energía eólica y aplicaciones agrarias. Como todo instrumento de medición, los anemómetros deben ser calibrados para asegurar la trazabilidad metrológica de las mediciones respecto a patrones estándar nacionales o internacionales [5].

Tradicionalmente, los túneles de viento utilizados para tales calibraciones se operan de forma manual o semi-automática. En este trabajo, se automatizó el proceso de medición mediante el control del motor del túnel de viento con un controlador PID. Este controlador recibe referencias de velocidad del viento proporcionadas por un anemómetro patrón. Además, estas mediciones se utilizan para llevar a cabo el proceso de calibración de otros anemómetros que también son afectados por el mismo flujo de viento.

En resumen, el sistema no solo controla las condiciones del viento de manera precisa, sino que también permite la calibración de instrumentos bajo prueba.

1.2. Estado del arte

A lo largo de los años, la tecnología de sensores de viento y los métodos de calibración han experimentado una evolución significativa, impulsada por la necesidad de mayor precisión y confiabilidad en las mediciones. En sus inicios, los anemómetros eran dispositivos mecánicos simples. Uno de los ejemplos más antiguos y utilizados es el anemómetro de copa, inventado por

Thomas Romney Robinson en 1846. Este dispositivo consistía en tres o cuatro copas montadas en los extremos de brazos horizontales que giraban alrededor de un eje vertical. La velocidad de rotación era proporcional a la velocidad del viento [23]. La calibración de estos anemómetros se realizaba mediante métodos empíricos en campo abierto.

Otro tipo histórico de anemómetro es el de hélice, que opera de manera similar al de copa pero utiliza una hélice en lugar de copas. Este diseño era especialmente útil para medir la velocidad del viento en una dirección específica [23].

En la actualidad, la tecnología ha avanzado hacia anemómetros más sofisticados y precisos. Los anemómetros de hilo caliente, por ejemplo, miden la velocidad del viento basándose en la tasa de enfriamiento de un hilo calentado eléctricamente, lo que les permite ser extremadamente sensibles a las fluctuaciones rápidas en la velocidad del viento [15]. Por otro lado, los anemómetros sónicos utilizan ondas ultrasónicas para medir la velocidad y dirección del viento, determinando el tiempo que tardan las ondas sonoras en viajar entre transductores. Estos anemómetros son altamente precisos y, al no tener partes móviles, requieren menos mantenimiento [11].

La calibración de los anemómetros en la actualidad se realiza en túneles de viento, instalando el anemómetro en la cámara de ensayos del túnel. El anemómetro se somete a un flujo de aire generado por el túnel, cuya velocidad se ajusta dentro de un rango específico. Durante el proceso, se compara la lectura del anemómetro con la de un instrumento patrón, asegurando que las mediciones sean precisas y trazables al Sistema Internacional (SI).

La incorporación de electrónica digital con microcontroladores y su programación, junto con una aplicación web desarrollada con un framework moderno, ha permitido obtener un sistema completo e integral que garantiza la repetibilidad del proceso de calibración y la correcta documentación en una base de datos que ofrece numerosas ventajas para un laboratorio, incluyendo la capacidad de almacenar grandes volúmenes de datos de manera organizada, facilitando el acceso rápido y eficiente a la información, mejorando la gestión de datos y permitiendo un seguimiento preciso de los experimentos y calibraciones. Además, aseguran la integridad y seguridad de los datos, protegiéndolos contra pérdidas y accesos no autorizados, y permiten la generación de informes detallados.

1.3. Motivación

El Servicio Meteorológico Nacional (SMN) representa al Centro Regional de Instrumentos de Buenos Aires (RIC III) de la Asociación Regional III [25]. Este Centro, dependiente de la Organización Mundial de Meteorología (WMO, por sus siglas en inglés), se destaca por sus capacidades en la verificación de instrumentos meteorológicos de presión atmosférica y viento, así como en

la calibración de sensores de temperatura y humedad. En ambos casos, se utilizan patrones de referencia y procedimientos que permiten establecer la trazabilidad de los instrumentos respecto al Sistema Internacional (SI).

La implementación de este trabajo permitirá actualizar y mejorar las capacidades del RIC, permitiendo incorporar la capacidad de **calibración** de anemómetros, no solo de los instrumentos propios de la red meteorológica del SMN, sino también de instrumentos de terceros tanto a nivel nacional como en la región latinoamericana.

Además, el laboratorio busca ser acreditado bajo la norma ISO 17025 [4], la cual establece requisitos generales para la competencia, imparcialidad y consistencia en la operación de laboratorios de ensayo y calibración. Esta tesis contribuye significativamente en esa línea de trabajo, proporcionando una base sólida para cumplir con los requisitos de acreditación, tales como la implementación de un sistema de calibración robusto, la mejora continua de los procesos y la demostración de la competencia técnica del laboratorio. Esto permitirá mejorar la calidad y confiabilidad de las calibraciones realizadas, asegurando resultados precisos y válidos.

1.4. Objetivos

La calibración de los anemómetros debe cumplir con normas estandarizadas [3] [1] [2] y ser realizada por laboratorios acreditados. Para seguir este enfoque, mediante la realización de un relevamiento y estudio del sistema actual de verificación de viento y su proceso, se propone desarrollar un sistema que:

1. Incorpore un datalogger basado en la placa de desarrollo EDU-CIAA, que permita la adquisición y transmisión de los datos de los sensores hacia un servidor mediante Ethernet.
2. Incorpore un sistema de control en el datalogger para automatizar el cambio de velocidades del túnel de viento, utilizando un circuito que simule las tensiones del potenciómetro mediante PWM.
3. Aplique los procedimientos actuales de calibración de anemómetros de ultrasonido en un túnel de viento.
4. Ponga a disposición una aplicación web desarrollada con el framework Django en Python que permita:
 - Configurar los sistemas de adquisición y control del datalogger.
 - Cargar los metadatos necesarios para elaborar el presupuesto de incertidumbre.

- Visualizar las mediciones en tiempo real, con la posibilidad de reiniciar las mediciones en caso de falla.
- Procesar las muestras y calcular la incertidumbre combinada para cada valor de velocidad de viento preconfigurada, una vez finalizada la adquisición de datos.
- Guardar en la base de datos PostgreSQL tanto los datos crudos como los procesados, incluyendo los gráficos, tablas y los metadatos pertinentes.
- Descargar los datos crudos, procesados, gráficos y toda la información necesaria para el análisis posterior y la generación del certificado de calibración.

Capítulo 2

Viento

La velocidad del viento es una magnitud vectorial tridimensional que experimenta fluctuaciones aleatorias de pequeña escala en el espacio y en el tiempo, que se superponen a un flujo organizado de mayor escala. Para los fines de la observación meteorológica, la meteorología sinóptica y el pronóstico del tiempo, es crucial analizar el movimiento horizontal del aire que se define como viento. No obstante, es importante considerar que, para algunas aplicaciones, también es necesario conocer el movimiento vertical del viento. Ejemplos de esto son el aterrizaje de aeronaves y el estudio de la dispersión de contaminantes en la atmósfera. En la Figura 2.1 se muestran las tres componentes de un vector de viento \mathbf{v}_{xyz} , y sobre el plano XY se muestra la proyección \mathbf{v}_{xy} que no interesa analizar.

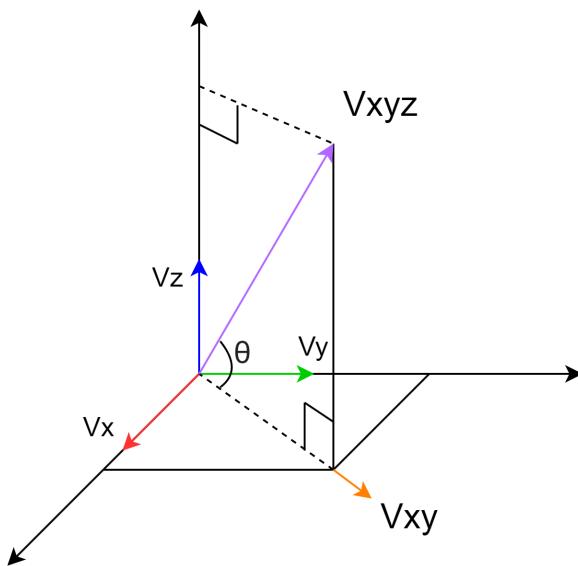


Figura 2.1: Tres componentes de la velocidad de viento v_x, v_y, v_z [m s^{-1}]. El vector de viento horizontal v_{xy} forma un ángulo de entrada θ con el vector de viento resultante v_{xyz} .

2.1. Viento en superficie

Se considerará que el viento de superficie es fundamentalmente una magnitud vectorial bidimensional definida por dos componentes

- **Intensidad:** Indica el valor de la velocidad del viento. En general se utilizan unidades como nudos (kn), metros por segundo (m s^{-1}) o kilómetros por hora (km h^{-1}).

- **Dirección y sentido:** Indica desde donde proviene el aire que se encuentra en movimiento.

Se mide en grados sexagesimales y considera el apartamiento con respecto al norte geográfico. Por ejemplo, un viento de dirección noreste (aire que se desplaza del noreste hacia el suroeste) tiene una dirección de 45° , ya que esta es la separación (en grados) entre el norte y el noreste.

El grado de fluctuación experimentado por el viento se denomina ráfagosidad, y las diferentes fluctuaciones son conocidas como ráfagas o rachas. La mayoría de los usuarios de datos sobre el viento necesitan conocer el viento horizontal en promedio, eliminando las fluctuaciones. Por ejemplo, en meteorología se utilizan promedios de entre 10 y 60 minutos con fines de predicción, mientras que en aeronáutica se utilizan promedios de 2 minutos. No obstante, cada vez son más las aplicaciones que precisan información acerca de la variabilidad o ráfagosidad del viento. Para este propósito, se utilizan tres magnitudes: la ráfaga o racha máxima, y las desviaciones típicas de la velocidad y de la dirección del viento [5].

La representación gráfica del viento se puede realizar de distintas formas, entre ellas se encuentran las flechas, las barbas y las líneas de corriente .

Las **flechas** [7] son la forma más común de representar un vector. La intensidad del viento se representa con la longitud de la flecha, mientras que la dirección se indica por el ángulo respecto al norte. El sentido de la flecha siempre apunta "hacia dónde va el viento". En la Figura 2.2 se muestra un ejemplo en el que se observa que el viento en la Patagonia proviene principalmente del sector oeste (es decir, el aire se desplaza de oeste a este), por lo tanto, las flechas apuntan hacia el este.

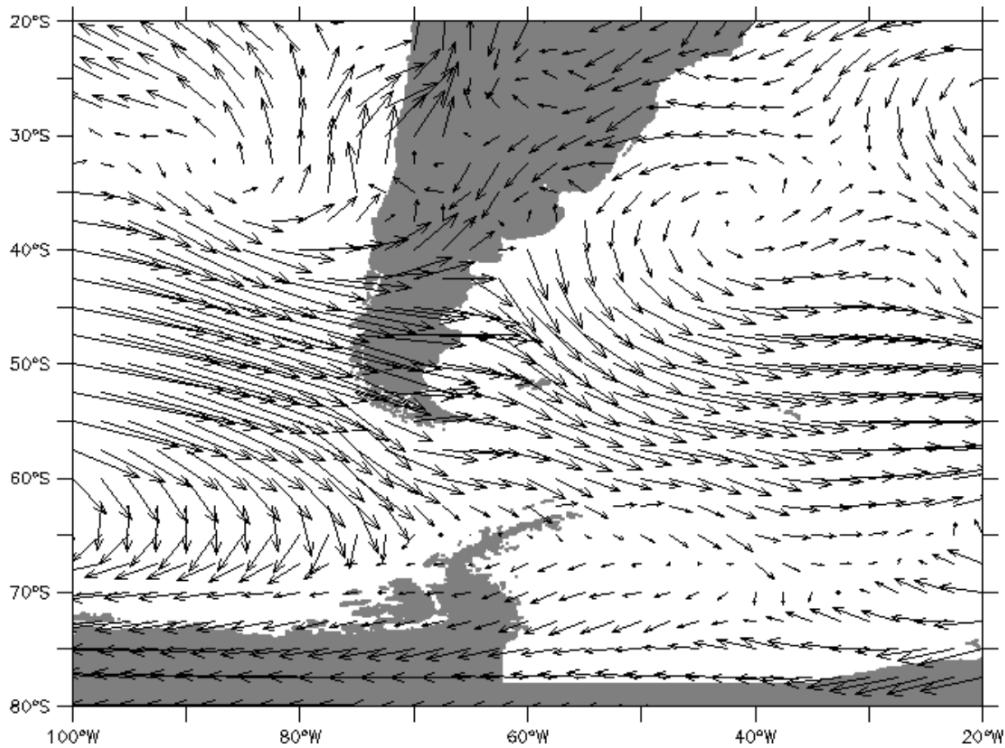


Figura 2.2: Campo de viento representado con flechas en el sur de Sudamérica. [18]

Las **barbas** [7] son ideogramas en los que la línea más larga indica la dirección del viento, mientras que las líneas más pequeñas, ubicadas en un extremo, indican la intensidad en múltiplos de 5 knot. En la Figura 2.3, a izquierda, se puede observar una representación del viento del oeste con una intensidad de 15 nudos, donde la media flecha representa 5 knot y la flecha entera representa 10 knot. En el centro, se observa una barba que representa el viento del sur con un banderín negro que indica una intensidad de 50 knot. Mas a la derecha se observa una barba que representa el viento proveniente del sureste con una intensidad dada por un banderín, dos lineas largas y una media linea, en total suman 75 knot. La dirección del viento está dada por el ángulo respecto al norte verdadero y, como se mencionó anteriormente, indica desde dónde viene el aire.



Figura 2.3: Representación gráfica del viento utilizando barbas. [18]

Las barbas son ampliamente utilizadas en meteorología sinóptica y aeronáutica. En la Figura 2.4 se muestra una representación gráfica del viento en 850 hPa utilizada en la meteorología aeronáutica. Nótese que en la elipse roja inferior, el viento es de 40 knot proveniente del sector

oeste en el norte de la Patagonia, mientras que en la zona central del país, en la elipse roja superior, prevalecen vientos del sector norte en torno a los 10 y 15 knot.

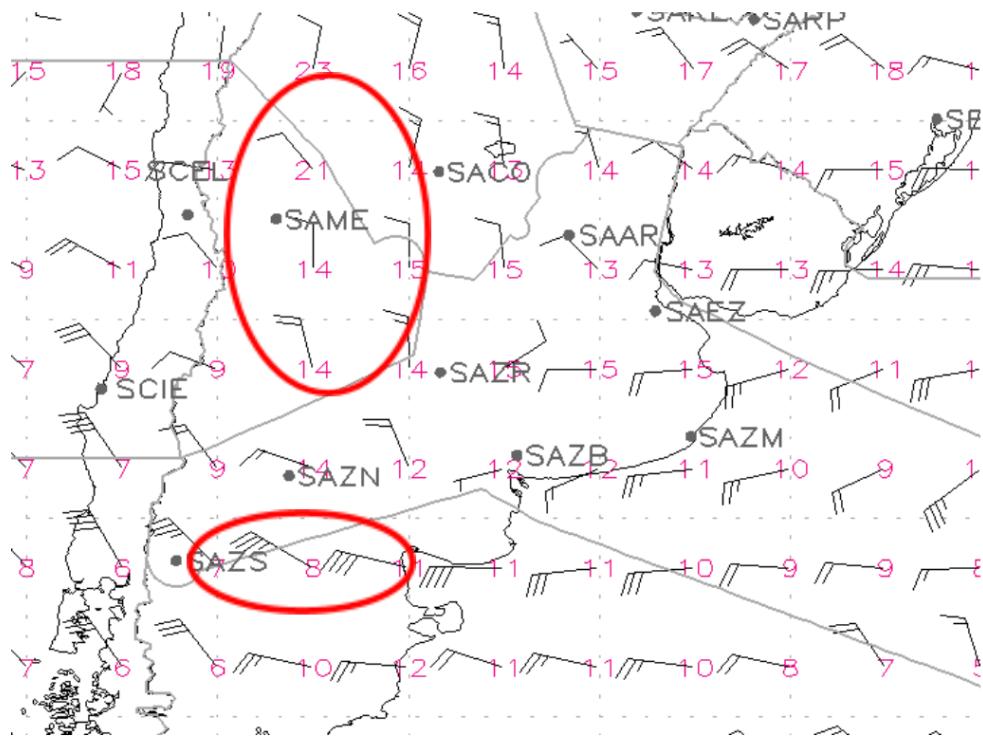


Figura 2.4: Representación del viento en 850 hPa para el día 14/4/2016 a las 12 UTC en la región central de Argentina. [18]

Las **líneas de corriente** [17] son la representación gráfica más comúnmente utilizada para analizar los factores de la circulación a gran escala. Una línea de corriente se define como la curva cuya tangente en cualquiera de sus puntos coincide con la dirección de la velocidad del aire. Generalmente, se emplean en niveles altos y en superficie en zonas tropicales debido a su capacidad para ilustrar patrones de flujo de aire de manera clara y precisa.

Un ejemplo de esta representación se puede observar en la Figura 2.5. En dicha figura, se muestra que un sistema de alta presión se desplazó hacia el sur de la Patagonia argentina. Este desplazamiento provocó el ingreso de aire templado a la península antártica, lo que a su vez generó condiciones de subsidencia. La subsidencia es el proceso mediante el cual el aire desciende en la atmósfera, aumentando su temperatura y reduciendo su humedad relativa. Este fenómeno intensificó el efecto de secado y calentamiento del aire al descender hacia la superficie, creando un ambiente más seco y cálido en la región.

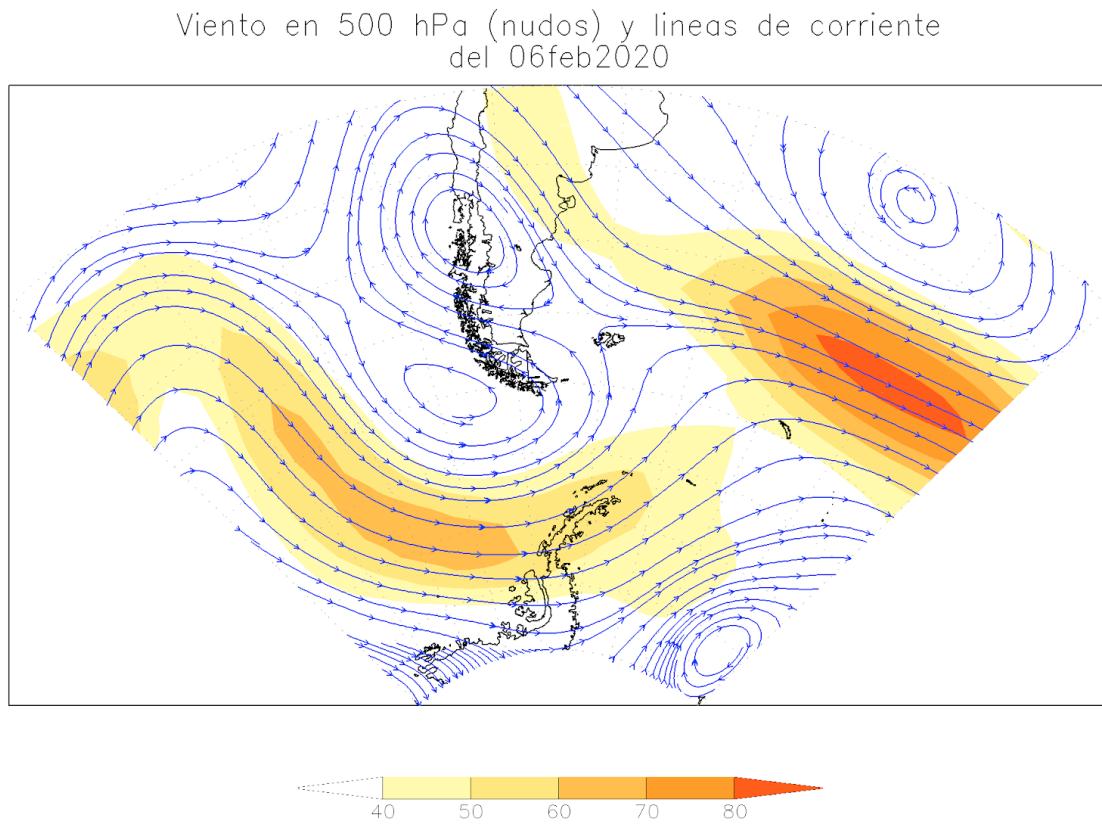


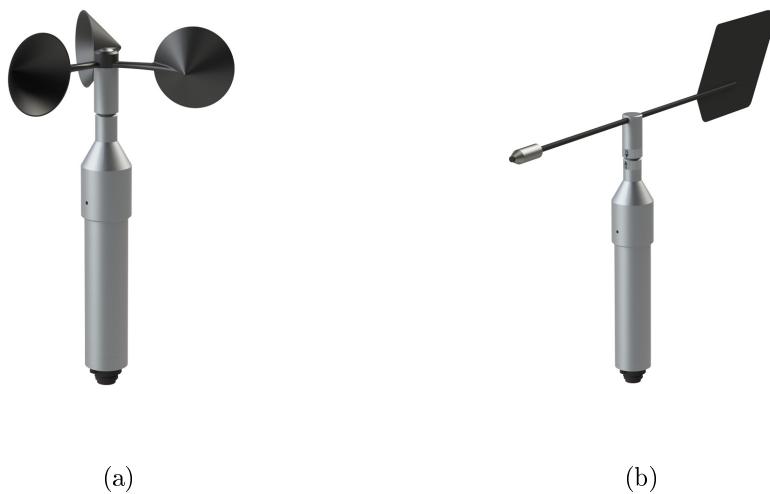
Figura 2.5: Viento (nudos) y líneas de corriente en 500hPa. [17]

2.2. Instrumentos de medición del viento

La medición precisa del viento en superficie requiere el uso de diversos instrumentos especializados, cada uno diseñado para capturar diferentes aspectos del flujo de aire. Estos instrumentos son esenciales para obtener datos confiables que se utilizan en múltiples aplicaciones meteorológicas y científicas. A continuación, se describen algunos de los instrumentos más comunes y su funcionamiento.

Sensor de Copas y Veleta

En la Figura 2.6 se muestra el sensor de copas y veleta, y es uno de los instrumentos más comunes para medir la velocidad y dirección del viento. Este dispositivo utiliza copas montadas en un eje vertical que giran con el viento, mientras que una veleta alineada con la dirección del viento proporciona datos direccionales. La velocidad de rotación de las copas se convierte en una medida de la velocidad del viento.



(a)

(b)

Figura 2.6: En (a) el sensor de coperola y en (b) el sensor de veleta. [21]

Anemómetro de Hilo Caliente

El anemómetro de hilo caliente que se muestra en la Figura 2.7 mide la velocidad del viento basándose en la tasa de enfriamiento de un hilo calentado eléctricamente. A medida que el viento pasa sobre el hilo, la tasa de enfriamiento varía con la velocidad del viento, permitiendo así su medición. Este tipo de anemómetro es especialmente útil para medir velocidades de viento bajas y en estudios de flujo de aire en túneles de viento.



Figura 2.7: Anemómetro de hilo caliente. [22]

Tubo Pitot

El tubo Pitot de la Figura 2.8, mide la velocidad del viento a través de la diferencia de presión entre un tubo orientado contra el flujo de aire y un tubo estático. Esta diferencia de presión se convierte en una medida de la velocidad del viento. Los tubos Pitot son ampliamente utilizados en aplicaciones aeronáuticas y en estudios de dinámica de fluidos.



Figura 2.8: Tubo Pitot. [22]

Anemómetro de Ultrasonido

El anemómetro de ultrasonido con el cual trabajaremos en esta tesis es similar al de la Figura 2.9, en general estos sensores utilizan ondas sonoras para medir la velocidad y dirección del viento. Emite pulsos ultrasónicos entre transductores y mide el tiempo que tardan en viajar entre ellos. La diferencia en el tiempo de viaje se utiliza para calcular la velocidad del viento. Este tipo de anemómetro es preciso y no tiene partes móviles, lo que reduce el mantenimiento. Se da una explicación mas detallada de su funcionamiento en el apéndice A.1.



Figura 2.9: Anemómetro de ultrasonido. [16]

Anemómetro Láser Doppler

El anemómetro láser Doppler mide la velocidad del viento utilizando el efecto Doppler de la luz láser. Como se ve en la Figura 2.10, un haz láser se dirige hacia partículas en el aire, y la luz dispersada se analiza para determinar el cambio en la frecuencia debido al movimiento de las partículas. Este cambio se convierte en una medida de la velocidad del viento. Este tipo de anemómetro es altamente preciso y se utiliza en investigaciones avanzadas de dinámica de fluidos.

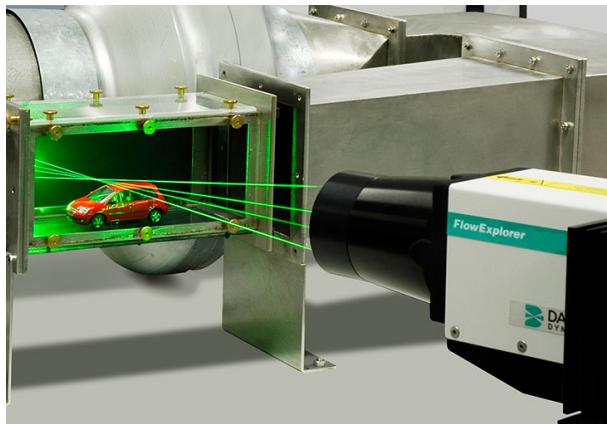


Figura 2.10: Anemómetro láser Doppler. [13]

2.3. Túnel de viento

El Servicio Meteorológico Nacional (SMN) cuenta con un túnel de viento con una antigüedad de aproximadamente 50 años, fabricado por la empresa *Santos Zaghi*. Este túnel se utiliza para la verificación de distintos tipos de anemómetros, detallados en la sección 2.2. El rango de velocidad del viento que actualmente alcanza está entre 1 m s^{-1} y 26 m s^{-1} , lo cual es adecuado para los anemómetros instalados en campo.

En la Figura 2.11 se muestra un esquema donde se observan las partes que lo componen: el motor que gira una helice, el variador con su panel de control y la zona de medición donde se instalan los sensores bajo prueba y su anemómetro patrón de referencia de la marca Vaisala, modelo WMT700, con un rango de operación de $0,01 \text{ m s}^{-1}$ a 65 m s^{-1} y una resolución de $0,01 \text{ m s}^{-1}$, montado en el techo del túnel. El sistema de referencia utilizado es el que se indica en la figura, donde la altura se mide en el eje vertical, la profundidad en el eje transversal y la longitud en el eje longitudinal.

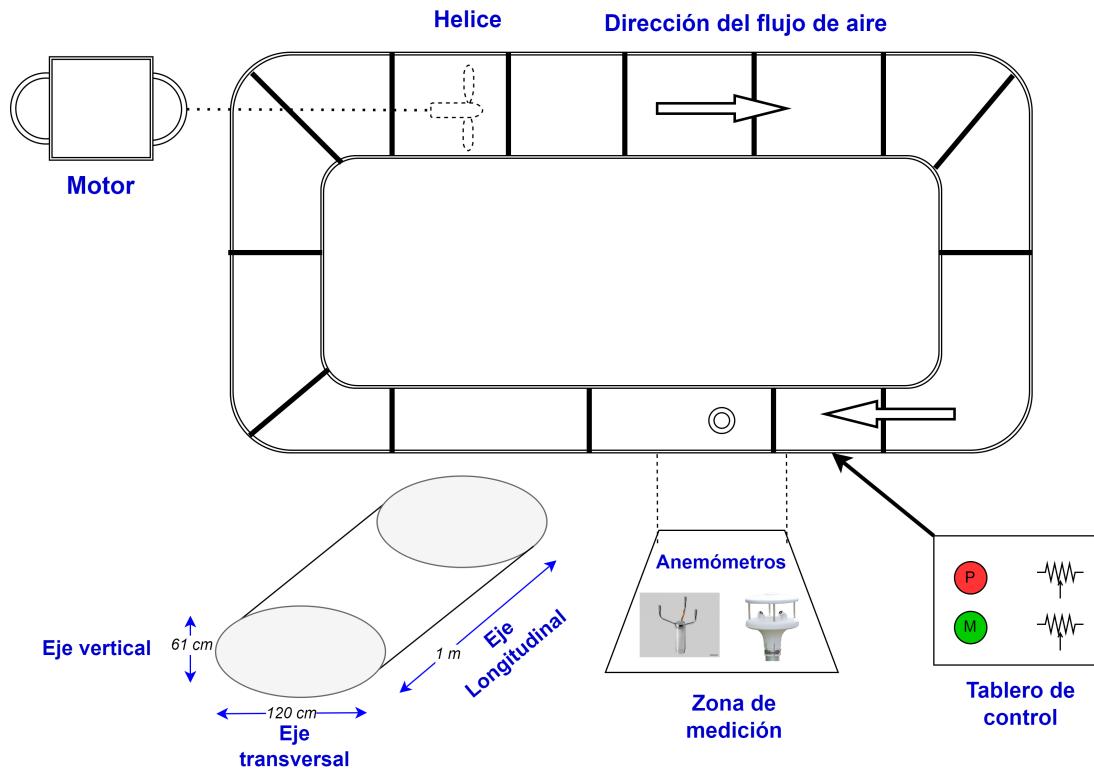


Figura 2.11: Esquema del túnel de viento.

En la Figura 2.12 se muestra una foto del túnel de viento, es de tipo cerrado con un perímetro medio de 21 m. A la izquierda, en la parte inferior se encuentra el motor.



Figura 2.12: Vista general del túnel de viento cerrado del SMN.

En la Figura 2.13a se muestra la parte externa de la zona de medición, la cual tiene una longitud de 1 m en la dirección longitudinal. Como se observa en la Figura 2.13b, en el centro hay una tapa con dimensiones de 33 cm × 28 cm, que se abre para permitir el ingreso de los instrumentos y sus respectivos soportes. En la Figura 2.13c, se muestra una barra cilíndrica

telescópica¹ montada en el suelo, la cual puede variar su altura hasta ingresar por un orificio en la parte baja del túnel, esta se utiliza para apoyar los sensores en un soporte.

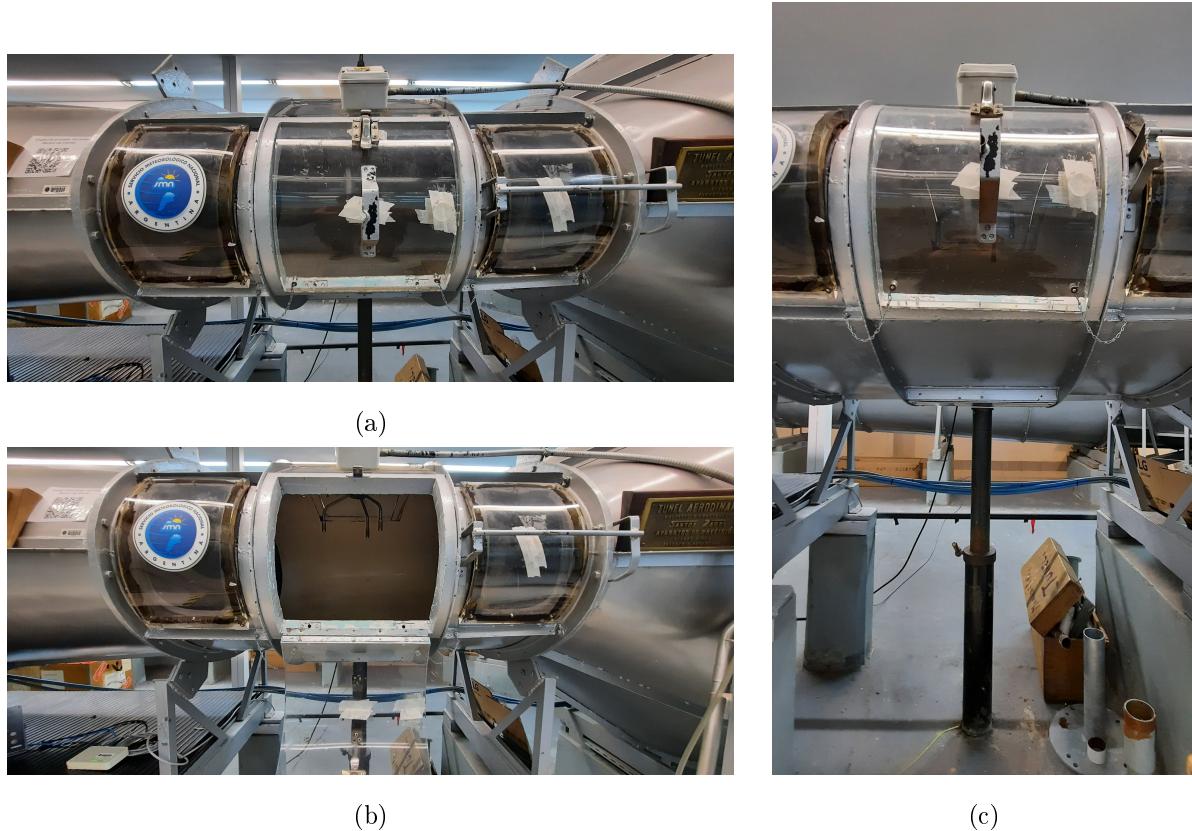


Figura 2.13: En (a) se muestra la puerta cerrada de la zona de medición, en (b) se muestra la puerta abierta y en (c) se muestra la varilla telescópica.

En la Figura 2.14a se muestra el interior de la zona de medición, cuya sección transversal elíptica se diseñó para disminuir la turbulencia en las paredes y mejorar la calidad del flujo de aire. Por otro lado, en la Figura 2.14b se observan los álabes (aleta de dirección), dispositivos esenciales que enderezan el flujo de aire, eliminando la componente rotacional inducida por el ventilador cuando atraviesa una esquina o cambio de sección abrupto. Esto no solo reduce la turbulencia, sino que también crea un flujo más uniforme y laminar, lo cual es crucial para obtener mediciones precisas y reproducibles en los ensayos aerodinámicos.

En la Figura 2.14c se observa la zona de medición de 1 m de largo, 120 cm de ancho y 61 cm de alto. Tanto el soporte como los anemómetros patrón y bajo calibración deben ser ubicados en esta zona, detallando la posición exacta en el volumen, para así lograr la reproducibilidad de las mediciones. Más adelante, en la sección 6.1, se mostrará un ensayo para caracterizar y encontrar dos posiciones tanto para el instrumento patrón, como para el instrumento bajo calibración.

¹La altura es ajustable en el eje vertical.

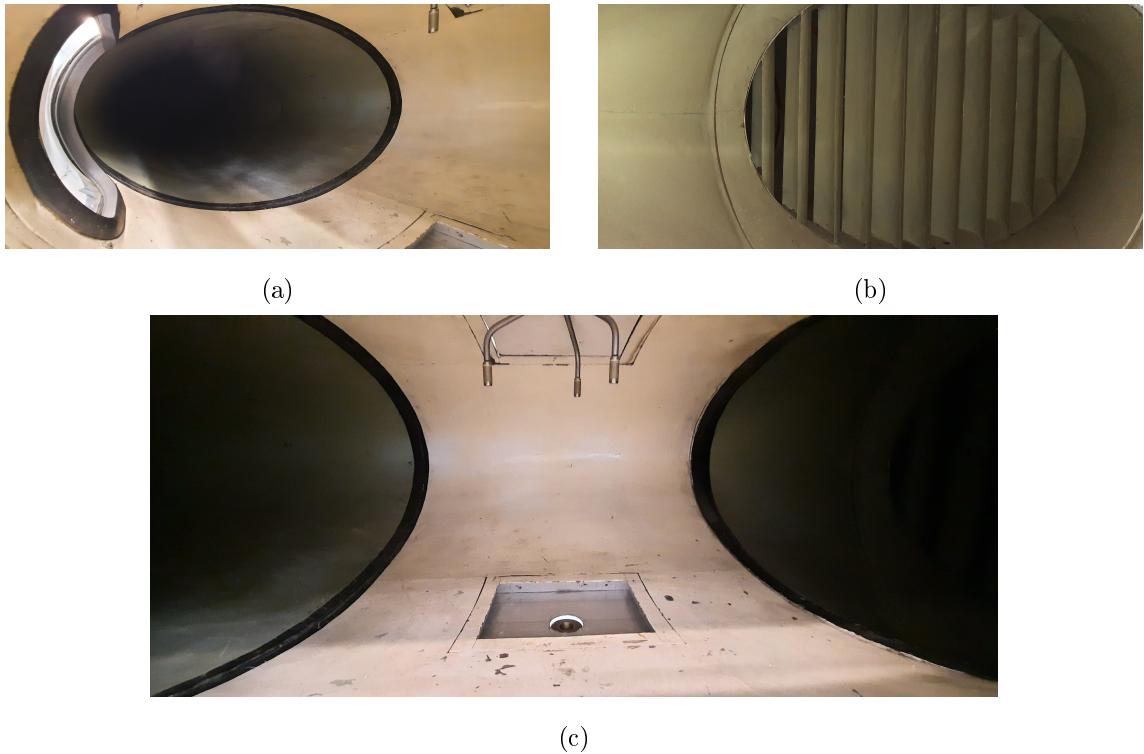


Figura 2.14: En (a) se muestra la vista lateral izquierda de la zona de medición, en (b) se muestran los álabe instalados en la parte derecha de la zona de medición, en (c) una vista frontal de la zona de medición.

En la Figura 2.15a se muestran los protectores auditivos necesarios para trabajar con el túnel de viento. Debido a que el motor y el flujo de aire dentro del túnel producen un alto nivel de ruido, estos protectores son indispensables para garantizar la seguridad laboral y proteger la audición del personal. En la Figura 2.15b se muestra la mesa de trabajo, que está equipada con una PC, una rosa de los vientos y el datalogger o adquisidor de datos. Además, la mesa cuenta con conexión a internet mediante un switch, lo cual facilita la transferencia de datos y la comunicación en tiempo real durante los ensayos.

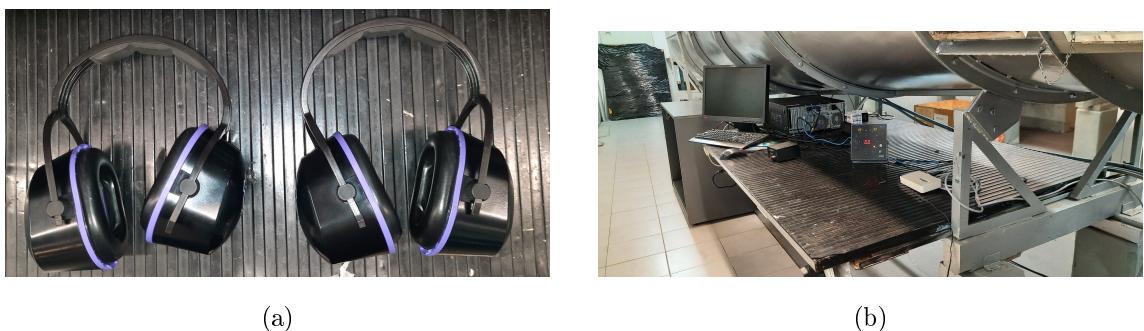


Figura 2.15: En (a) se muestra los protectores auditivos y en (b) se muestra la mesa de trabajo.

En las Figuras 2.16a y 2.16b se muestra el motor de corriente continua de 16 kW y 1800 rpm, el cual permite alcanzar velocidades de viento en el rango estimado de 1 m s^{-1} a 25 m s^{-1} . Este cuenta con cuatro poleas con sus respectivas correas para realizar la transmisión de movimiento, como se muestra en las Figuras 2.16c y 2.16d. Parte de la limitación de la velocidad del túnel se debe a la alineación de las poleas y al mantenimiento de las correas.

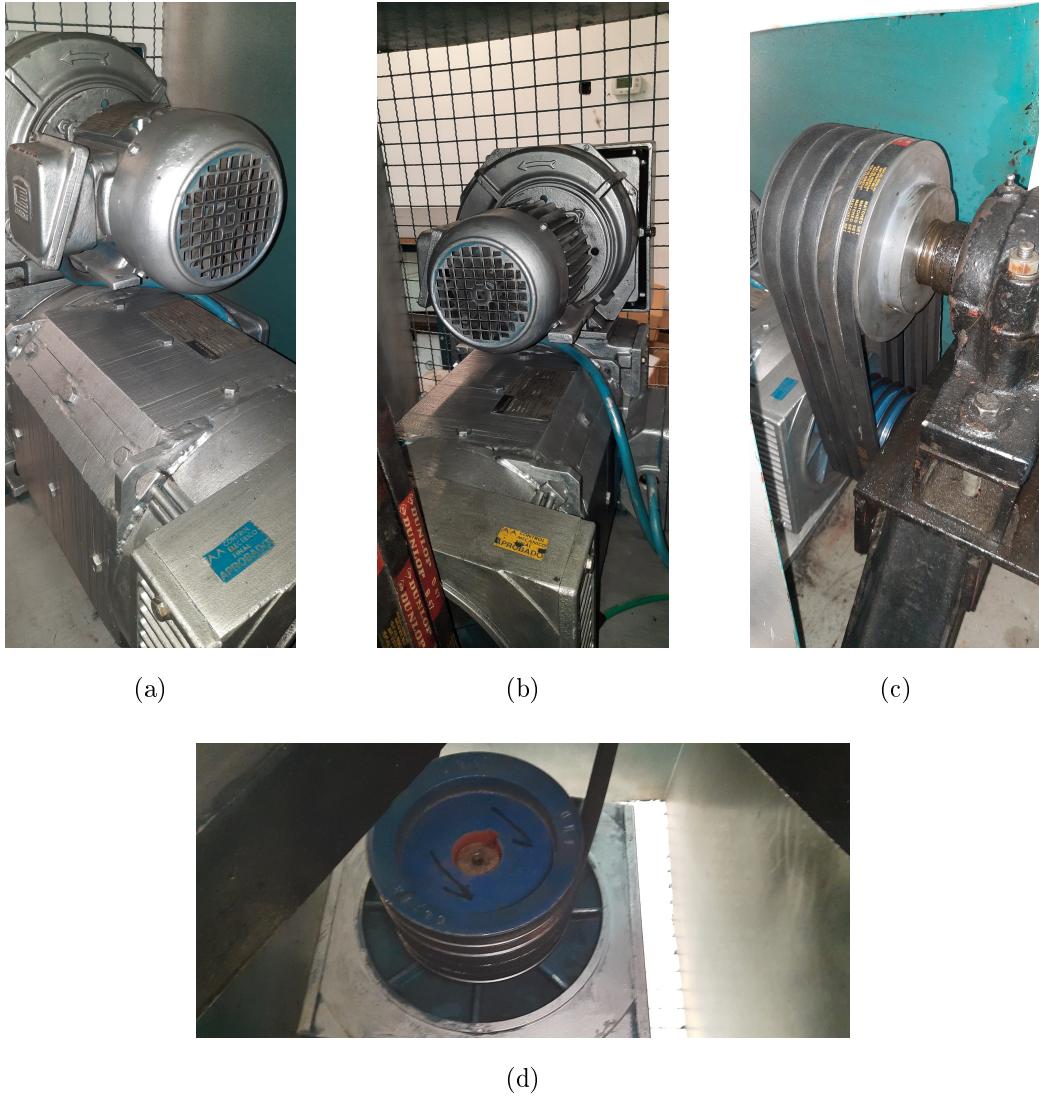


Figura 2.16: En (a) se muestra la vista lateral izquierda del motor, en (b) la vista lateral derecha, en (c) se muestra las correas conectadas a la polea superior y en (d) se muestra el eje del motor con la polea inferior y las correas.

En la Figura 2.17a se muestra el panel de control manual, que cuenta con dos botones, uno para iniciar y otro para detener el motor. La potencia del motor que varía la velocidad del viento dentro del túnel se regula manualmente mediante dos potenciómetros lineales: P1 de $1 \text{ k}\Omega$ y P2 de 500Ω . Además, dispone de dos medidores que indican el nivel de tensión suministrada al motor. En la Figura 2.17b se puede observar el circuito de electrónica de potencia del variador que

controla el túnel de viento. El variador recibe tensión trifásica, la cual se rectifica y se convierte en corriente continua para alimentar el motor del túnel.



(a)



(b)

Figura 2.17: En (a) se muestra el panel de control manual y en (b) el tablero eléctrico del variador de velocidad.

Capítulo 3

Cálculo de Incertidumbre

En este capítulo se abordarán los diferentes tipos de incertidumbre de medición y se explicará cómo identificar sus fuentes. Posteriormente, se revisará la importancia del certificado de calibración y su relación con la trazabilidad de las mediciones. A continuación, se definirán las fuentes de incertidumbre específicas que se consideran al calibrar los anemómetros de ultrasonido dentro del túnel de viento. Finalmente, se presentará el modelo teórico aplicado para la evaluación de la incertidumbre de medición. Este enfoque permitirá una comprensión integral de los aspectos que influyen en la precisión y fiabilidad de las mediciones realizadas en el contexto de los sensores de viento.

3.1. Incertidumbre de la medición

La incertidumbre en la medición es una característica inherente a cualquier proceso de medición y se refiere a la duda sobre el resultado de una medición. Esta incertidumbre puede surgir de diversas fuentes, como las limitaciones del instrumento de medición, las condiciones ambientales o el método de medición empleado. Para identificar las fuentes de incertidumbre, se puede utilizar el diagrama de Ishikawa (diagrama de pescado) mostrado en la Figura 3.1.

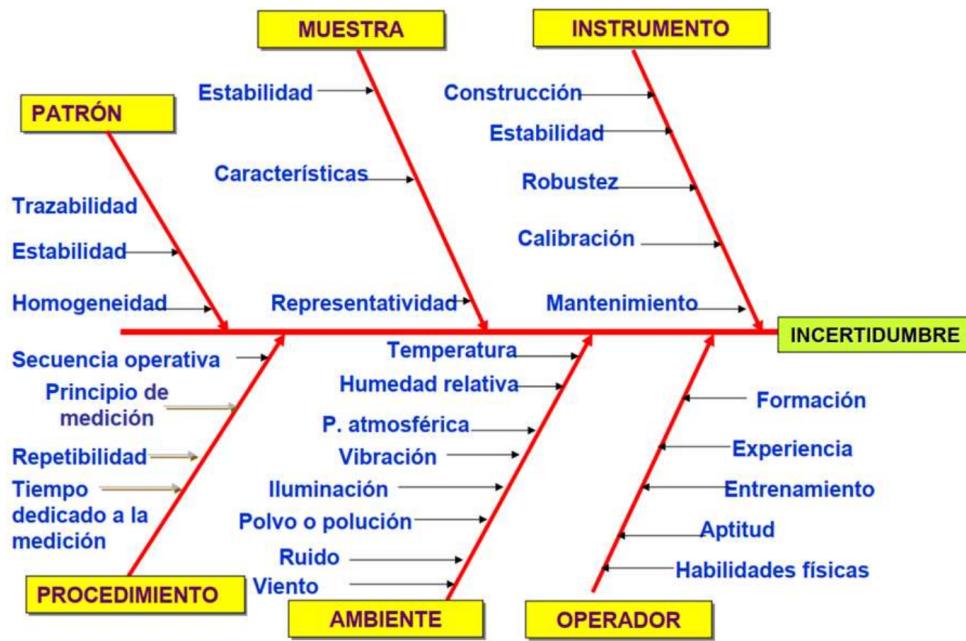


Figura 3.1: Diagrama de las fuentes que producen incertidumbre en la medición de un mensurando.

Existen dos tipos principales de incertidumbre: la incertidumbre de tipo A y la incertidumbre de tipo B. La incertidumbre de tipo A se evalúa mediante métodos estadísticos y se basa en datos experimentales. A partir de una serie de observaciones de una cantidad X_i , se obtiene el promedio (ecuación 3.1), el desvío estándar (ecuación 3.2) y la incertidumbre estándar (ecuación 3.3).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.1)$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (3.2)$$

$$u(\bar{x}) = \frac{\sigma}{\sqrt{n}} \quad (3.3)$$

La incertidumbre de tipo B se basa en conocimientos previos y no en datos experimentales directos. Esta puede derivarse de información contenida en manuales, certificados de calibración o la experiencia previa con el instrumento de medición. Normalmente, se especifica un entorno $\Delta = \frac{B-A}{2}$ definido como su incertidumbre expandida. Luego, para obtener la incertidumbre estándar, se divide por un número que depende de la función de distribución asumida. Para una distribución rectangular (la más usual), se divide por $\sqrt{3}$; para una distribución triangular, se divide por $\sqrt{6}$; y para una distribución de media cuadrática (RMS), donde las fuentes de incertidumbre son idénticas y aditivamente independientes, se divide por $\sqrt{2}$.

La combinación de las incertidumbres de tipo A y tipo B da lugar a la incertidumbre combinada, que proporciona una medida más completa de la incertidumbre total asociada a una medición. Generalmente, se elabora un presupuesto de incertidumbre (*uncertainty budget*) a partir de todas las fuentes identificadas, con el fin de obtener una visión detallada y precisa. La incertidumbre combinada, definida en la ecuación 3.4, puede ser ampliada mediante un factor de cobertura, el cual se determina utilizando la distribución t de Student. Esta distribución de probabilidad ajusta el intervalo de confianza deseado en función del tamaño de la muestra y del número de grados de libertad. Los grados de libertad efectivos, representados en la ecuación 3.5, representan el número de valores en un cálculo estadístico que son libres de variar. Esta ampliación de la incertidumbre se conoce como incertidumbre expandida, la cual se muestra en la ecuación 3.6, proporcionando un intervalo alrededor del resultado de la medición con una alta probabilidad de contener el valor verdadero del mensurando.

$$u_c = \sqrt{\sum_{i=1}^n (c_i \cdot u_i)^2} \quad (3.4)$$

$$V_{ef} = \frac{u_c^4}{\sum_{i=1}^n \frac{u_i^4}{v_i}} \quad (3.5)$$

$$U = k \cdot \sqrt{\sum_{i=1}^n (c_i \cdot u_i)^2} \quad (3.6)$$

3.2. Calibración y trazabilidad

La calibración es una operación que, bajo condiciones especificadas, establece en una primera etapa una relación entre los valores y sus incertidumbres de medida asociadas obtenidas a partir de los patrones de medida, y las correspondientes indicaciones con sus incertidumbres asociadas. En una segunda etapa, utiliza esta información para establecer una relación que permite obtener un resultado de medida a partir de una indicación. Es importante no confundir la calibración con el ajuste del instrumento, que es un proceso diferente.

Para expresar la calibración, se emiten certificados de calibración donde, en los puntos donde fue calibrado, se informa una función o tabla de corrección que se debe aplicar a las lecturas del instrumento, lo que contribuye a mejorar la exactitud de las mediciones. Otro aspecto crucial que se debe informar en el certificado es la incertidumbre de corrección, la cual es una componente fundamental de la incertidumbre total de la medición.

Un certificado de calibración debe incluir la siguiente información:

- **Identificación del instrumento calibrado:** Marca, modelo y número de serie del instrumento.
- **Identificación del patrón de referencia:** Marca, modelo y número de serie del patrón con el que se realizó la calibración.
- **Fecha de calibración:** Fecha en la que se realizó la calibración.
- **Condiciones ambientales:** Temperatura, humedad y presión atmosférica durante la calibración.
- **Resultados de la calibración:** Valores medidos, incertidumbres asociadas y cualquier desviación observada.
- **Curvas de calibración:** Gráficos de las curvas de calibración ascendente y descendente, si se incluyen.
- **Tablas de calibración:** Tablas con los datos de calibración, si se proporcionan.
- **Método de calibración:** Descripción del procedimiento utilizado para la calibración.
- **Certificación del laboratorio:** Información sobre la acreditación del laboratorio que realizó la calibración.
- **Firma y sello:** Firma del técnico responsable y sello del laboratorio.

El instrumento estándar también cuenta con estos parámetros en su propio certificado de calibración y son necesarios para completar el presupuesto de incertidumbre de la medición del instrumento bajo calibración (IBC), asegurando que todas las posibles fuentes de incertidumbre sean consideradas y cuantificadas.

La calibración es esencial para mantener la fiabilidad y trazabilidad a los estándares primarios. En la Figura 3.2 se muestra cómo, a partir de un patrón internacional, se calibra un patrón nacional tomando como referencia la incertidumbre del patrón internacional. De esta forma, en sentido descendente, se tienen incertidumbres cada vez mayores hasta llegar al equipo de medición. Por otro lado, en sentido inverso, la trazabilidad es una propiedad de un resultado de medida por la cual dicho resultado puede relacionarse con un valor de referencia, nacional o internacional, a través de una cadena ininterrumpida y documentada de calibraciones, donde cada una contribuye a la incertidumbre de la medición.



Figura 3.2: Trazabilidad metrológica.

3.3. Fuentes de incertidumbre

En particular se va a considerar las fuentes de incertidumbre debidas a:

- El sensor bajo calibración.
- El sensor patrón de referencia.
- El túnel de viento.

En la tabla 3.1 se detallan las fuentes de incertidumbre que se evaluarán en el presupuesto de incertidumbre para cada punto de viento configurado. En la segunda columna se nombran las fuentes y en la tercera columna se define cada una. Luego, cada incertidumbre se clasifica según el tipo, en función de cómo fue determinada. En la quinta columna se especifica el tipo de distribución probabilística. Finalmente, en la última columna se detalla el factor de normalización por el cual se debe dividir para obtener la incertidumbre estándar de cada fuente.

Fuentes de Incertidumbre		Descripción	Tipo	Distribución de probabilidad	Factor de normalización
Patrón	Calibración	Incertidumbre especificada en su certificado de calibración.	B	N	2
	Ajuste de calibración	Desvío estándar del ajuste lineal con los datos discretos del certificado de calibración	A	N	1
	Resolución del instrumento	Menor cambio detectable.	B	R	$\sqrt{3}$
	Repetibilidad	Desvío estándar de la media aritmética de las mediciones.	A	N	\sqrt{n} , n muestras
	Histéresis	Diferencia entre los valores de mediciones realizadas en los ciclos ascendente y descendente.	B	R	$\sqrt{3}$
	Factor de bloqueo	Incertidumbre asociada al factor de bloqueo que se define como la relación entre el área transversal del túnel de viento y el área efectiva del anemómetro, proyectada en el plano transversal.	B	RMS	1
Túnel de viento	Homogeneidad	Variación espacial del flujo de aire dentro del túnel.	B	R	$\sqrt{3}$
	Ajuste de la homogeneidad	Desvío estándar del ajuste lineal con los datos discretos del certificado de homogeneidad.	A	N	1
	Estabilidad	Variación temporal del flujo de aire dentro del túnel.	B	R	$\sqrt{3}$
	Ajuste de la estabilidad	Desvío estándar del ajuste lineal con los datos discretos del certificado de estabilidad.	A	N	1
	Factor de calibración	Incertidumbre asociada al factor de calibración que proporciona la relación entre las condiciones en la posición de medición de referencia y las condiciones en la posición del IBC.	B	RMS	1
IBC	Resolución del Instrumento	Menor cambio detectable.	B	R	$\sqrt{3}$
	Repetibilidad	Desvío estándar de la media aritmética de las mediciones.	A	N	\sqrt{n} , n muestras
	Histéresis	Diferencia entre los valores de mediciones realizadas en los ciclos ascendente y descendente.	B	R	$\sqrt{3}$
	Factor de bloqueo	Incertidumbre asociada al factor de bloqueo que se define como la relación entre el área transversal del túnel de viento y el área efectiva del anemómetro, proyectada en el plano transversal.	B	RMS	1

Tabla 3.1: Fuentes de incertidumbre evaluadas para el presupuesto de incertidumbre de un anemómetro bajo calibración en el túnel de viento del SMN.

Vale la pena destacar que la histéresis también se utiliza como indicador de la no linealidad del instrumento bajo calibración; una histéresis alta sugiere que el instrumento muestra un comportamiento diferente cuando la magnitud medida está aumentando en comparación a cuando está disminuyendo.

Por otro lado, tanto el factor de bloqueo como en factor de calibración deben ser calculados previamente.

3.4. Modelo teórico de medición

El modelo de medición utilizado para calcular la incertidumbre se muestra en la ecuación 3.7. Este modelo se basa en una comparación, donde se obtiene la corrección del mensurando bajo calibración a partir de la diferencia de medición en un punto determinado y bajo condiciones ambientales controladas, entre un patrón calibrado y el instrumento bajo calibración.

$$CV_{IBC} = \overbrace{V_p + CV_p}^{V_{ref}} - V_{IBC} \quad (3.7)$$

donde se define:

- V_{IBC} representa el valor de viento medido con el anemómetro bajo calibración.
- CV_{IBC} representa la corrección del anemómetro bajo calibración.
- V_p representa el valor de viento medido con el anemómetro patrón.
- CV_p representa la corrección de viento del anemómetro patrón, obtenida a partir de su certificado de calibración.

A partir de la ecuación 3.7 y tomando en cuenta las fuentes de incertidumbre, definidas en la tabla ?? se realiza el cálculo de la incertidumbre combinada de la corrección, como la suma de los cuadrados de las incertidumbres estándar de cada fuente, como se muestra en la ecuación 3.8.

$$u^2(CV_{IBC}) = u^2(V_{ref}) + u^2(V_{IBC}) + u^2(\text{túnel de Viento}) \quad (3.8)$$

donde se define:

- $u^2(CV_{IBC})$ es la incertidumbre estándar de la corrección del anemómetro bajo calibración.
- $u^2(V_{ref})$ representa la suma de todas las fuentes de incertidumbre estándar debidas al patrón de referencia.

- $u^2(V_{IBC})$ representa la suma de todas las fuentes de incertidumbre estándar debidas al anemómetro bajo calibración.
- $u^2(\text{túnel de Viento})$ es la contribución de incertidumbre estándar referida al túnel de viento, quien se encarga de generar condiciones de velocidad y dirección de viento.

Capítulo 4

Implementación del Datalogger

Este capítulo se centra en el diseño y la programación del datalogger, que será utilizado en el banco de medición para calibrar los anemómetros de ultrasonido. Este sistema está basado en la placa de desarrollo EDU-CIAA [20], con el agregado de un *shield* específico para la aplicación. El sistema se describe en la Figura 4.1 y consta de distintos módulos, a saber: un módulo para la adquisición de muestras de viento mediante el protocolo RS485, un módulo para la recepción y transmisión de datos a través de Ethernet, un módulo de alimentación de 12 V, y un módulo PWM con una etapa de filtrado seguida de otra de amplificación para entregar la señal de control del túnel de viento. Además, se toman muestras de esta señal y de la fuente para monitorear por software el comportamiento del sistema.

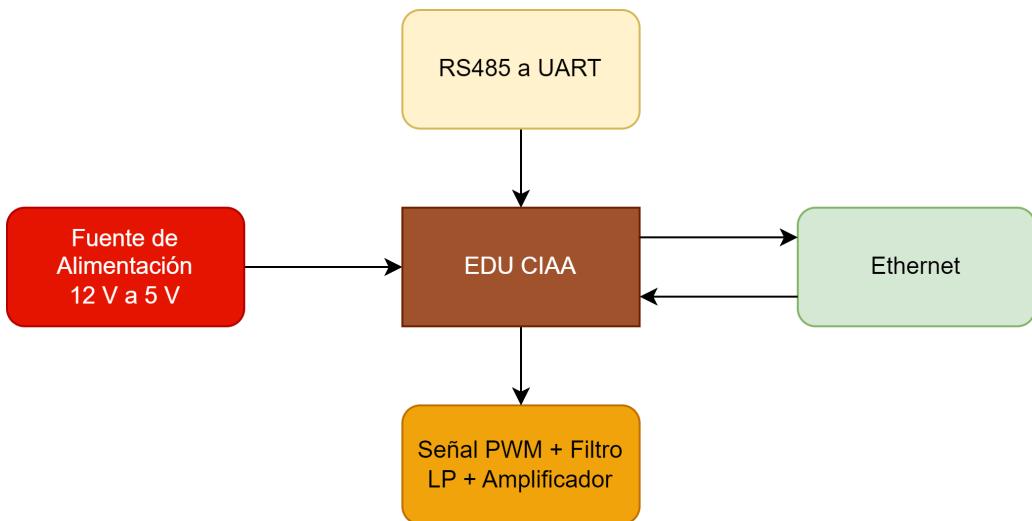


Figura 4.1: Diagrama en bloques del sistema electrónico.

4.1. Desarrollo del hardware

Para el desarrollo del Hardware, en una primera etapa se realizó un proceso de investigación de hojas de datos, manuales, artículos y videos de internet para ver qué módulos y componentes comprar para la implementación de la aplicación. Además, se utilizaron herramientas como LT Spice para la simulación de circuitos y KiCad para el diseño de PCB.

La placa de desarrollo EDU-CIAA-NXP, se muestra en la Figura 4.2, es una versión de bajo costo de la CIAA-NXP [20] pensada para la enseñanza universitaria, terciaria y secundaria.

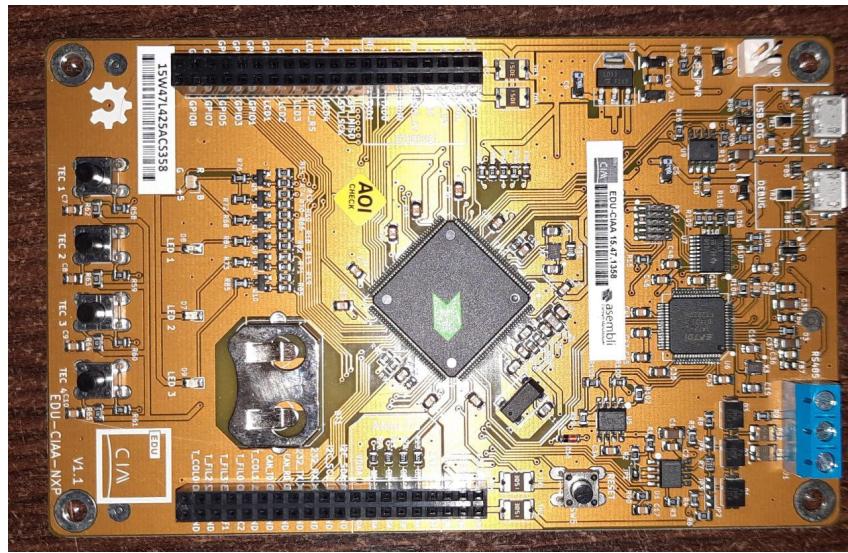


Figura 4.2: Placa de desarrollo EDU CIAA

En la Figura 4.3 se describe un diagrama en bloques de los periféricos de la placa de desarrollo.

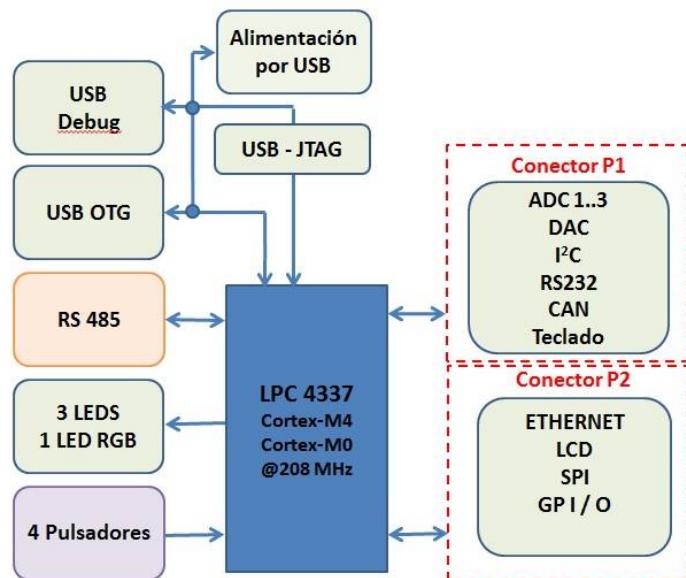


Figura 4.3: Diagrama en bloques de la EDU-CIAA

El SMN proporcionó los sensores de viento de ultrasonido. Se trabajó con los siguientes anemómetros: el modelo WMT700 de la marca VAISALA, Figura 4.4a , utilizado como anemómetro patrón, y el modelo HD51.3D de la marca Delta OHM, Figura 4.4b, utilizado como anemómetro bajo calibración. Ambos sensores operan con la interfaz eléctrica RS485 y vienen con un cable de aproximadamente 15 metros, puesto que estos sensores se instalan en torres de 10 metros de altura.



Figura 4.4: En (a) se muestra el sensor Vaisala modelo WMT700, en (b) se muestran el sensor Delta OHM modelo HD51.3D.

Algunas especificaciones técnicas de este tipo de anemómetros se muestran en la tabla A.1. Las más relevantes para el proceso de calibración son, el rango de medición, la resolución, la precisión, los protocolos de comunicación, la alimentación eléctrica y el intervalo de tiempo mínimo con el que puede tomar muestras.

4.1.1. Módulo de alimentación electrica

Los anemómetros funcionan con una alimentación de 12 V para el sistema principal y 24 V para el sistema de calefacción, ya que estos sensores pueden ser expuestos a entornos con temperaturas por debajo de cero grados. Por otro lado, los módulos Ethernet, RS-485 y los amplificadores operacionales trabajan con 5 V, mientras que el resto de la placa de desarrollo opera con 3,3 V. Por esta razón, se decide alimentar todo el sistema con una fuente de 12 V,

para garantizar el funcionamiento de los anemómetros en condiciones normales de temperatura dentro del laboratorio. Para lograr estos niveles de tensión, se optó por un regulador basado en el chip LM2596S (Step-Down), como se indica en la Figura 4.5. Se conectan 12 V a la entrada y se ajusta el potenciómetro hasta obtener 5 V a la salida. Esta salida estará disponible para conectar el resto de los módulos. Cabe aclarar que, si bien la EDU-CIAA tiene una salida de 5 V, esta no proporciona la corriente suficiente para alimentar todos los módulos, ya que está limitada por los 500 mA que entrega un puerto USB 2.0. En contraste, el regulador DC-DC basado en el LM2596 puede entregar hasta 3 A, satisfaciendo así las necesidades de corriente de todos los módulos conectados.



Figura 4.5: Regulador de voltaje DC-DC basado en el chip LM2596.

En la Figura 4.6 se observa el esquemático del sistema, donde se muestra cómo se conecta el módulo LM2596 con el resto de los componentes. Este esquemático incluye un divisor resistivo y un diodo Zener de 3,3 V para limitar y adaptar la tensión de 12 V a un rango de 3,3 V, ya que se toman muestras de la tensión de alimentación a través del canal analógico-digital ADC_CH1 de la EDU-CIAA. Esta información de la tensión se envía junto con los datos de los anemómetros, permitiendo un monitoreo en tiempo real del sistema de alimentación. Además, se han agregado filtros de bypass a la alimentación del amplificador operacional LM358 para asegurar una operación estable y con bajo ruido.

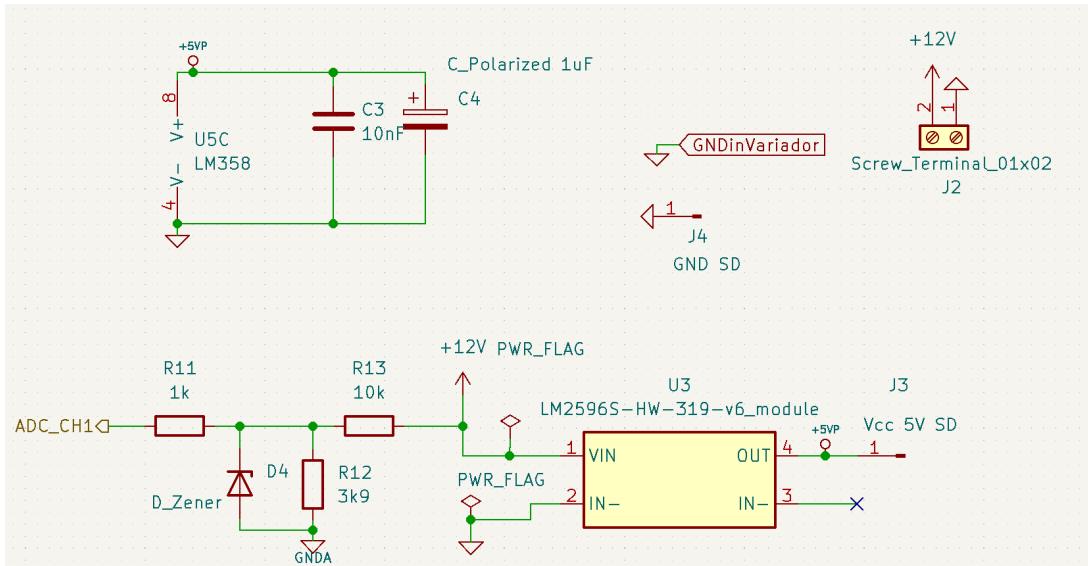


Figura 4.6: Esquemático del sistema de alimentación con módulo LM2596, diodo Zener, divisor resistivo y filtros de bypass para el LM358.

4.1.2. Módulo de adquisición de datos de viento

En la sección 2.2 del capítulo 2 describimos los distintos tipos de sensores para medir la intensidad y dirección del viento. En particular se trabajó con anemómetros de ultrasonido cuya interfaz electrónica utiliza el protocolo RS-485 [10], el mismo es un protocolo diferencial ampliamente utilizado en entornos industriales debido a su robustez y confiabilidad. Utiliza tres cables (A, B y GND) y transmite datos mediante la diferencia de tensión entre A y B, ofreciendo alta inmunidad al ruido. En la Figura 4.7 se muestra un esquema de conexión entre el sensor y el sistema. En particular, nos interesa dos cables para la alimentación V+ (12 V) y V- (GND), y tres cables para los datos, DATA+ (A), DATA- (B) y GND. Puesto que tanto el sensor patrón como el sensor bajo calibración trabajan con el puerto RS485 y la placa de desarrollo EDU-CIAA cuenta con un solo puerto RS485, se va a utilizar un adaptador para así tener disponible dos puertos RS485.

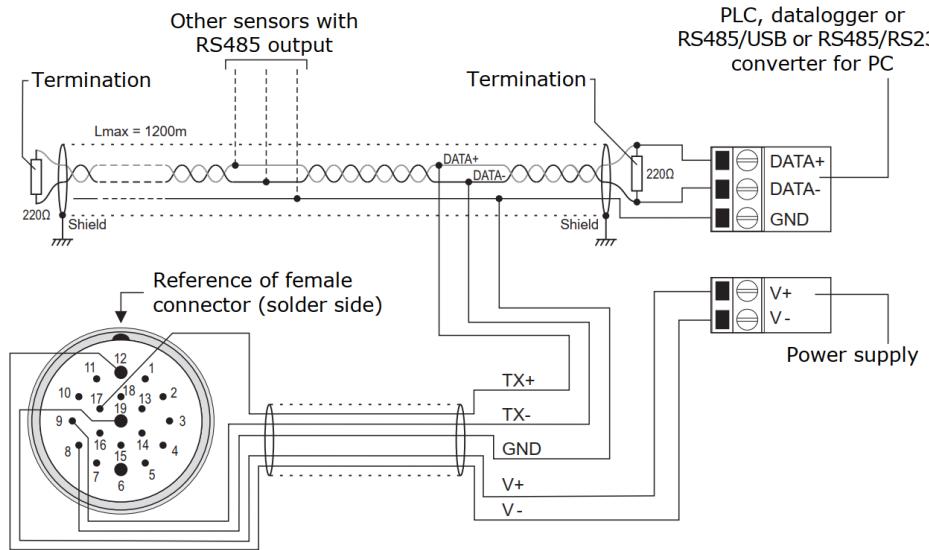


Figura 4.7: Esquema de conexión de datos y alimentación del sensor HD51.3D con un Datalogger o un PLC. [8]

En la Figura 4.8, se muestra el puerto RS485 integrado de la EDU-CIAA, a esta bornera se conecto directamente el anemómetro patrón WMT700. El sensor envía datos en modo ráfaga cada segundo, y este puerto permite una comunicación directa y eficiente sin necesidad de adaptadores adicionales.

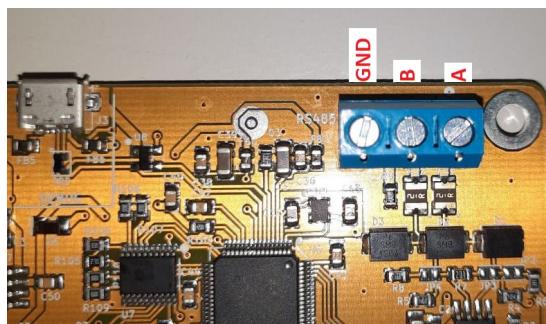


Figura 4.8: Puerto RS-485 integrado en la EDU-CIAA

Para conectar el anemómetro bajo calibración se utilizó el módulo MAX485, Figura 4.9, que convierte las señales RS-485 a niveles de la UART (TTL).

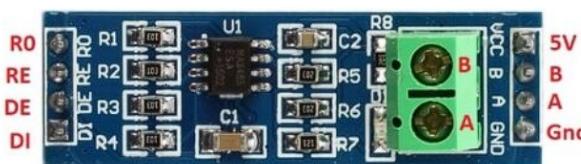


Figura 4.9: Módulo convertidor de RS485 a UART (TTL)

El módulo se alimentó con 5 V suministrados por el regulador LM2596S. Desde un extremo, se conectaron los cables A y B del sensor RS-485, mientras que del otro lado se conectó a la UART-RS232 de la EDU-CIAA. Para adaptar la señal TTL a la placa de desarrollo, se utilizó un resistor de $1\text{ k}\Omega$ entre el pin RO del MAX485 y la UART-RS232. Los pines DE y RE del MAX485 se conectaron a GND para permitir la recepción de datos desde el sensor hacia la placa. El pin DI no se conectó, ya que el sensor no requiere recibir datos o comandos desde la EDU-CIAA.

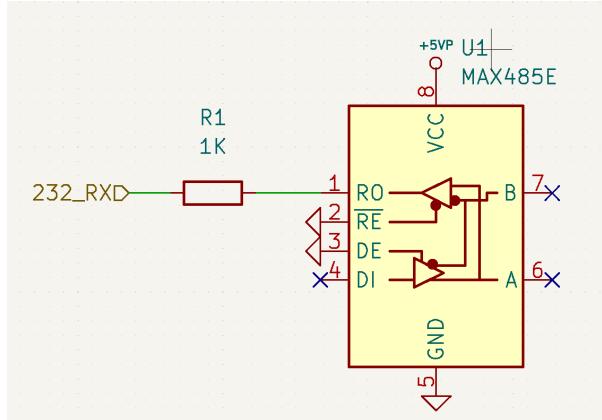


Figura 4.10: Esquemático del módulo MAX485 para la recepción de datos del sensor bajo calibración.

4.1.3. Módulo de comunicación Ethernet/SPI

Para comunicar el sistema embebido con distintos servidores, como el servidor WebSocket desarrollado en la sección 5.1.2, así como servidores NTP para sincronizar la fecha y hora, se utilizó el módulo W5100, Figura 4.11. Este módulo emplea el protocolo TCP/IP para proporcionar una comunicación estable. Se alimenta con 5 V y utiliza una interfaz SPI para comunicarse con la placa de desarrollo. Una de las principales ventajas del módulo W5100 [24] es su capacidad para manejar cuatro sockets independientes, lo que permite conectarse a hasta cuatro servidores distintos de manera concurrente. Esto mejora la eficiencia y la flexibilidad del sistema, facilitando la integración con múltiples servicios de red simultáneamente.

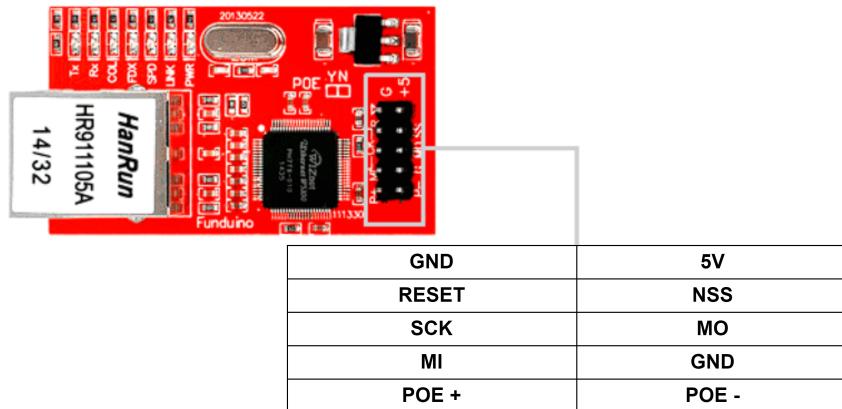


Figura 4.11: Módulo W5100 utilizado para realizar la conectividad a internet a través de la interfaz SPI del microcontrolador.

Como se muestra en la Figura 4.12, el módulo W5100 se alimentó con 5 V y se conectaron los pines del módulo a la EDU-CIAA de la siguiente manera: el pin MO a SPI_MOSI, el pin MI a SPI_MISO, y el pin SCK a SPI_SCK. Además, se conectó el pin NSS al GPIO1 para poder reiniciar el módulo cuando sea necesario por software. El resto de los pines del módulo quedaron sin conectar. Con esta configuración, se logró una comunicación bidireccional, permitiendo transmitir datos de los sensores de viento desde el datalogger hacia la aplicación web, así como recibir comandos desde la aplicación web hacia el datalogger.

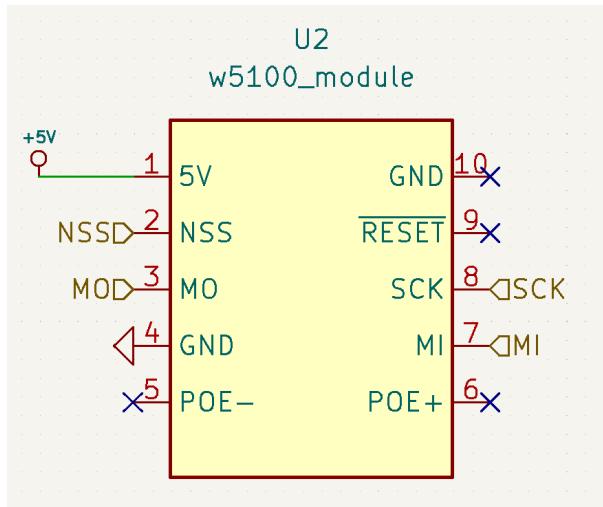


Figura 4.12: Esquemático del módulo W5100 para la comunicación bidireccional con distintos servidores.

4.1.4. Circuito PWM

En esta sección, se presenta el desarrollo de un circuito que permite controlar la potencia del motor del túnel de viento a través de modulación por ancho de pulso (PWM, por sus siglas en

inglés). Primero, se describe el funcionamiento original del variador de velocidad. En la Figura 4.13 se muestra un esquema del tablero de control que se conecta con el variador de velocidad del túnel de viento. El tablero posee dos pulsadores: uno de marcha (BT1-NC) y otro de parada (BT2-NA), que permiten encender y apagar, respectivamente, el conjunto variador-motor del túnel de viento. La velocidad del viento se regula manualmente mediante dos potenciómetros lineales: $P1 = 1\text{k}\Omega$ y $P2 = 500\Omega$. Además, el circuito incluye una resistencia $R1 = 1,2\text{k}\Omega$ conectada a una fuente de alimentación $VCC = 10\text{V}$, la cual es proporcionada por el tablero. Estos componentes, junto con los potenciómetros, conforman un divisor de tensión variable que permite ajustar el voltaje en un rango de 0V a 10V . Esta tensión se aplica al pin de entrada analógico-digital $VADC$ del tablero. Este nivel de baja tensión obtenido se amplifica y adapta a través de su variador a los niveles de tensión continua requeridos por el motor en el rango de 0V a 300V . El nivel de tensión aplicado al motor se puede visualizar en un voltímetro de corriente continua, V , que incluye el tablero.

Se realizó una medición con un amperímetro en serie con la resistencia $R1$ y se obtuvo que la corriente que pasa por $R1$ cuando se ajusta el potenciómetro para obtener una $VADC = 3,56\text{V}$ es de $140\mu\text{A}$. Cuando se ajusta el potenciómetro para obtener una $VADC = 5\text{V}$, la corriente es de $200\mu\text{A}$. Con estas mediciones podemos concluir que el pin $VADC$ del variador tiene una resistencia de entrada de alta impedancia, ya que el consumo de corriente no supera los $500\mu\text{A}$.

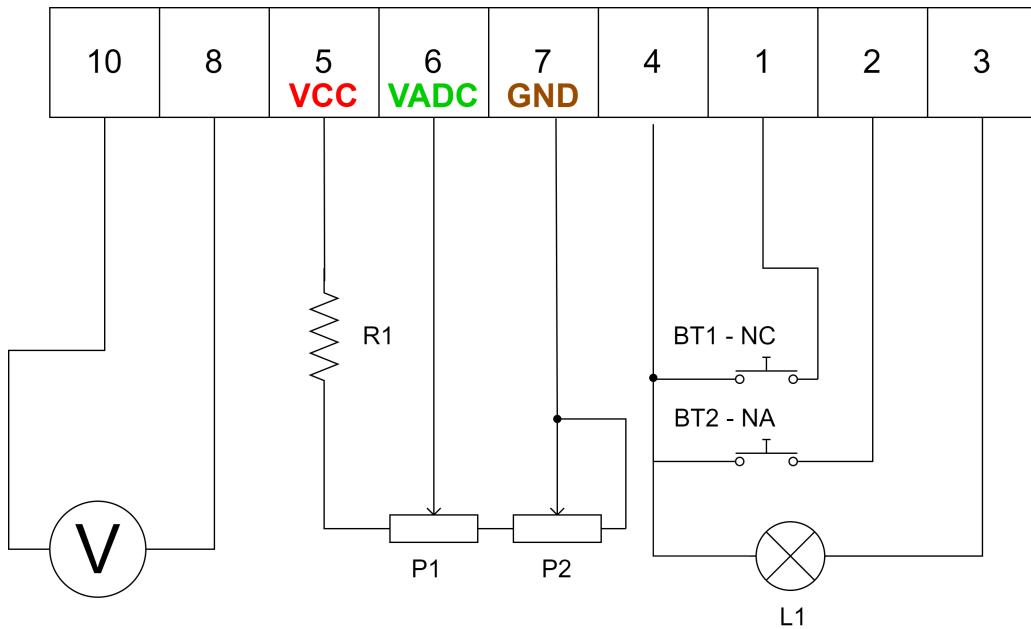


Figura 4.13: Esquemático circuital del tablero de control del tunel de viento.

Para lograr controlar el nivel de tensión $VADC$ por medio de software, se remplazó los dos potenciómetros y el resistor $R1$ por un circuito diseñado en la siguiente sección. Por otro lado,

se hizo un relevamiento de la corriente máxima que puede entregar la fuente del variador VCC conectando distintos valores de resistencia entre VCC y GND . Los resultados se muestran en la tabla 4.1. Como se observa en la tabla, a medida que disminuimos la carga R de prueba, tanto la tensión, como la corriente que puede entregar la fuente VCC disminuyen. Se alcanza la corriente maxima de 100 mA a partir de una carga de 100Ω , cuya tensión de la fuente cae a 2,29 V. Tomando como referencia el consumo de corriente y la tensión que precisa el datalogger, esta fuente VCC del tablero, no es capaz de suministrar la corriente necesaria y mantener la tensión constante, por esta razón se optó por alimentar el sistema como se explicó en la sección 4.1.1

R [Ω]	VCC [mV]	I_R[mA]
47K	9,9	2
10K	9,52	5,9
1K	9,29	9,4
560	7,55	13,48
330	5,63	16,97
100	2,29	100
47	0,859	100

Tabla 4.1: Mediciones de tensión y corriente de la fuente del variador para distintas resistencias de carga.

4.1.4.1. Simulación del circuito

En la Figura 4.14 se muestra el diseño del circuito del PWM [14], que reemplaza al divisor de tensión formado por los potenciómetros $P1$, $P2$ y la resistencia $R1$, mostrado en la Figura 4.13. La señal de entrada es una señal PWM (cuadrada) obtenida desde un pin de la EDU-CIAA. Esta señal, para un ciclo de trabajo del 100 %, tiene una amplitud de 3,3 V acorde con los niveles digitales de tensión de la EDU-CIAA. Luego, esta señal pasa por dos etapas de filtrado, cada una con una frecuencia de corte de 1592 Hz. La tensión vFilter1 (señal verde) en la primera etapa del filtro pasabajo tiene un tiempo de crecimiento corto, del orden de 8 ms, pero presenta mayor rizado. La segunda etapa del filtro genera la tensión voutFilter (señal azul), que tiene un mayor tiempo de crecimiento, del orden de 12 ms, pero con un rizado mucho menor. Con esta doble etapa de filtro pasa bajo, se obtiene una señal continua con menor rizado, como se muestra en la Figura 4.15.

Posteriormente, la señal rectificada ingresa a una etapa de amplificación no inversora, implementada con el amplificador operacional LM358. Este nivel de continua es el que se conecta al pin $VADC$ del variador de velocidad del túnel de viento, pero para la simulación se reemplaza

con una carga de $1\text{ M}\Omega$. La señal a la salida del amplificador tiene una ganancia de $\frac{3,5\text{ V}}{3,3\text{ V}} \approx 1,06$. Se decidió amplificar hasta 3,5 V porque se encontró que la máxima velocidad que logra alcanzar el túnel, de 25 m s^{-1} , se alcanza con 3,5 V a la entrada del *VADC*. Esta limitación viene dada por la mecánica del motor, por lo tanto, en estas condiciones no es necesario amplificar a valores mayores de 3,5 V, ya que el túnel no incrementa su velocidad de viento.

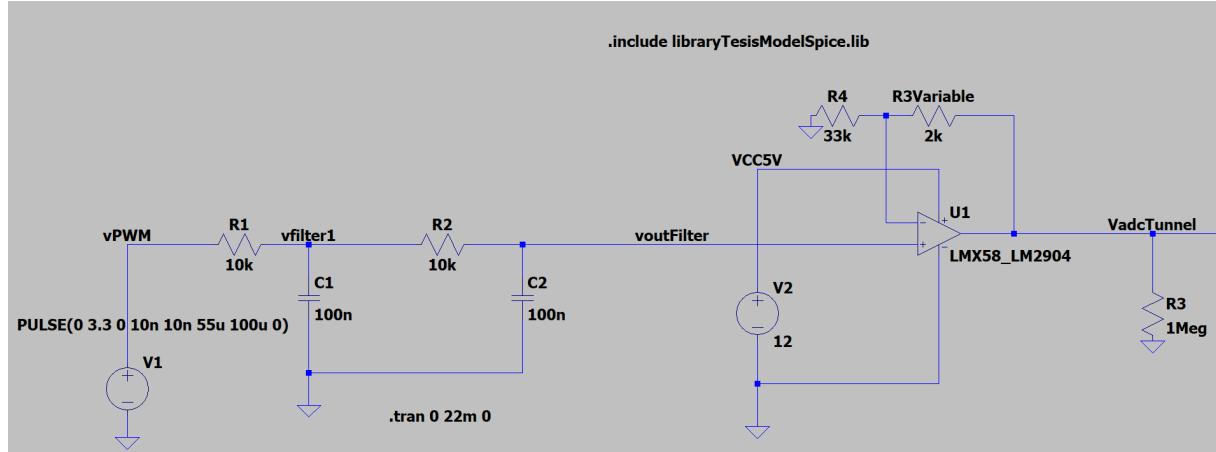


Figura 4.14: Circuito diseñado para obtener distintos niveles de tensión mediante la variación del ciclo de trabajo de una señal cuadrada (PWM).

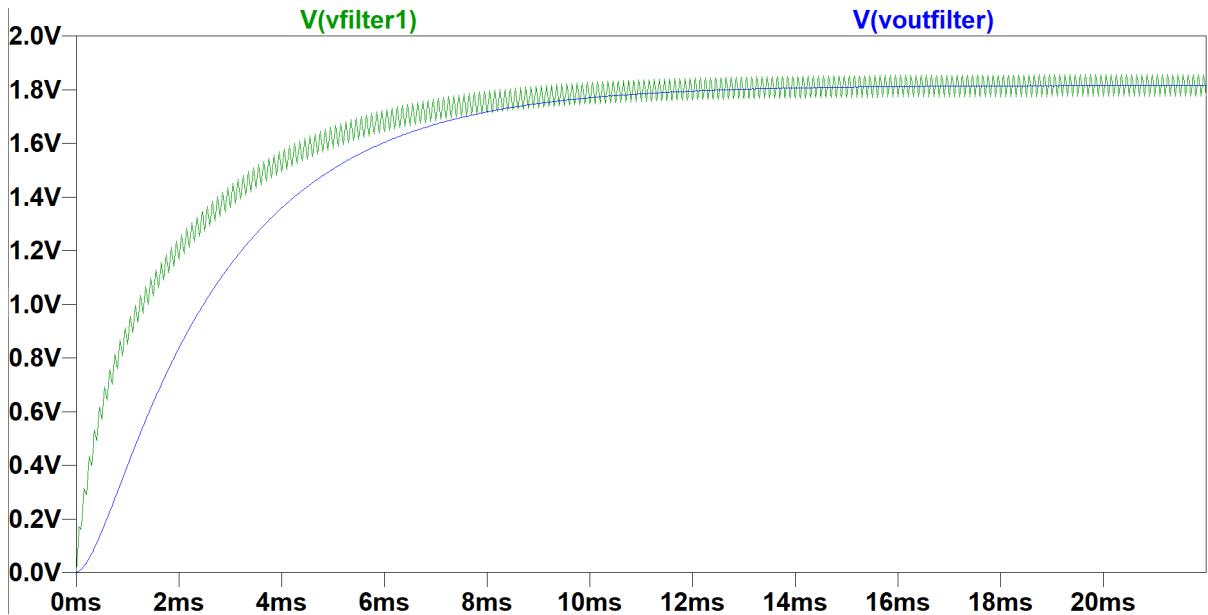


Figura 4.15: Señales a la salida del primer filtro LP (curva verde) y segundo filtro LP (curva azul).

Se dejó un resistor variable (Preset) $R3 = 10\text{ k}\Omega$ para poder modificar la ganancia del amplificador si fuera necesario. En este caso se lo ajustó para alcanzar 3,5 V a la salida del operacional. Sin embargo, para cuando sea mejorada la mecánica y transmisión del motor del túnel de viento,

podrá ser recalibrado para llegar a mayores niveles de tensión, y por ende mayores velocidades de viento(talvez lo pongo abajo).

En la Figura 4.16 se muestra el resultado de la simulación para un pulso vPWM (señal verde) de $100\text{ }\mu\text{s}$ de periodo (10 kHz) y una amplitud de 3,3 V con un ciclo de trabajo del 100 %. La señal vOutFilter (señal en azul) es la que ingresa al amplificador operacional alcanzando su valor máximo en 3,3 V y la señal vAdcTunnel (señal en rojo) se mide a la salida del amplificador operacional alcanzando un tension máxima de 3,5 V lo que produce la máxima velocidad del motor.

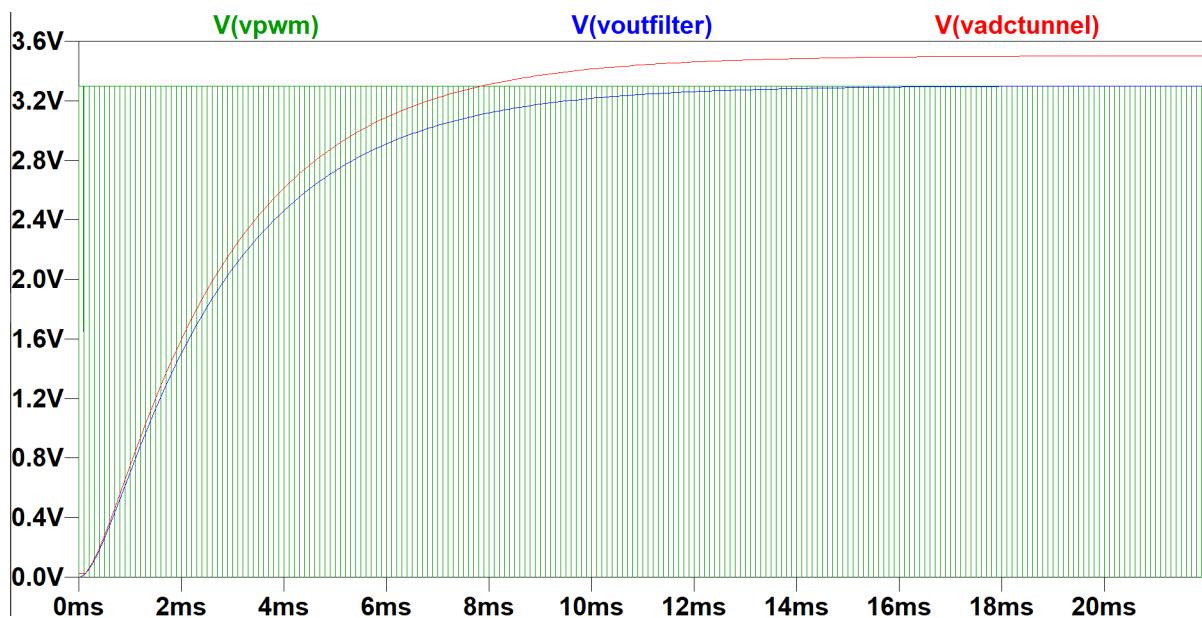


Figura 4.16: Señales simuladas para un ciclo de trabajo del 100 %.

En la simulación, al regular el ciclo de trabajo de la señal de entrada, se obtuvieron distintos niveles de tensión continua. Esto se puede observar en las Figuras 4.17 y 4.18, que muestran los resultados para ciclos de trabajo del 75 % y 55 %, respectivamente.

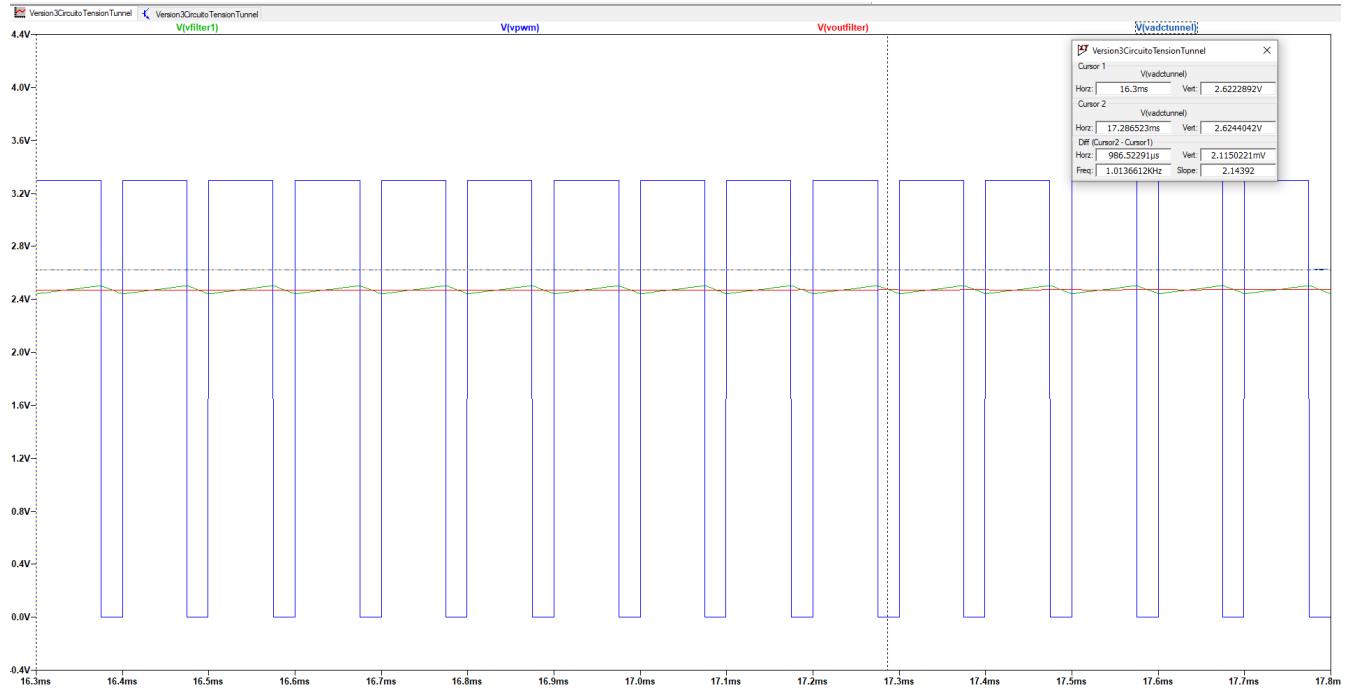


Figura 4.17: Señales simuladas para un ciclo de trabajo al 75 % de la señal de entrada.

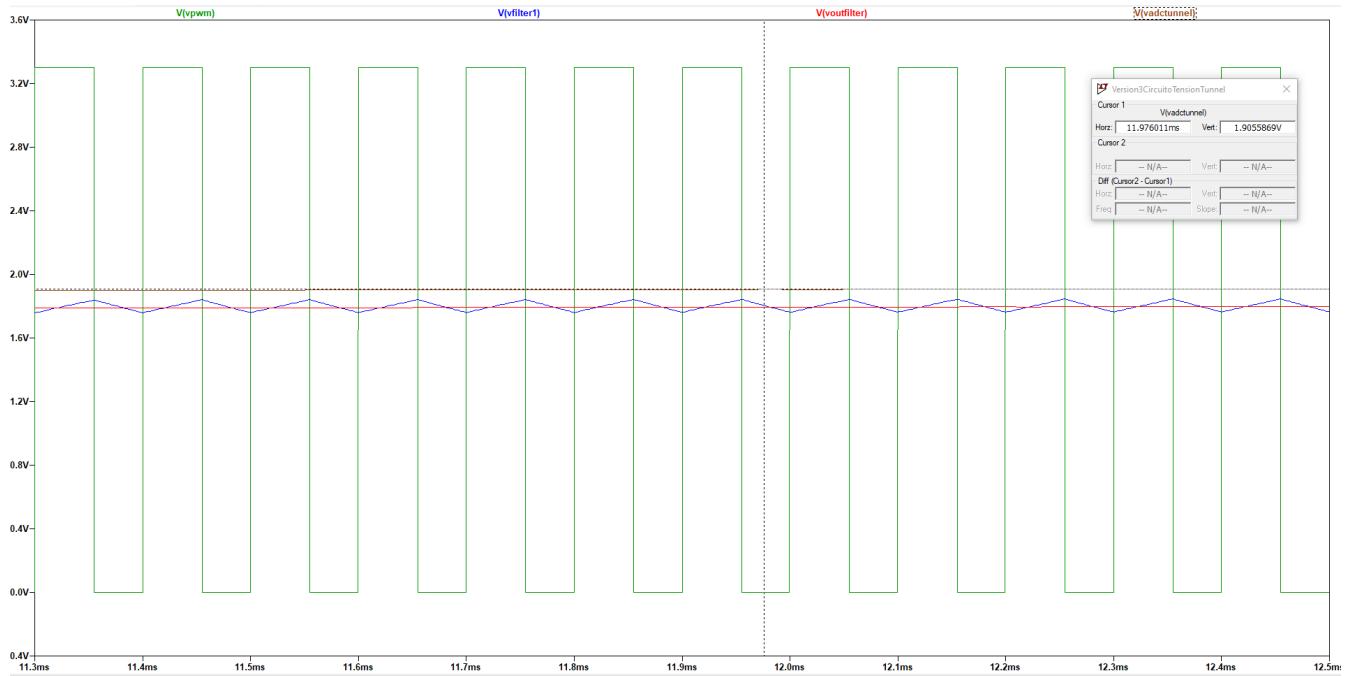


Figura 4.18: Señales simuladas para un ciclo de trabajo al 55 % de la señal de entrada.

4.1.4.2. Implementación del circuito

Se implementó el circuito de la Figura 4.19 en un protoboard de prueba con componentes Through-Hole Technology (THT), conectado a la EDU-CIAA, como se muestra en la Figura 4.20. Para comprobar y caracterizar el comportamiento del PWM, se utilizaron distintos valores

de ciclo de trabajo configurados manualmente en el firmware del microcontrolador. El ciclo de trabajo de la señal cuadrada se modificó desde 1 hasta 255, equivalente a un rango de 0 % a 100 %.

Para realizar estas mediciones, se utilizó un osciloscopio RIGOL DS2302A. En el canal 1 se midió la señal PWM de 10 kHz con amplitud de 3,3 V que sale de un pin digital de la placa de desarrollo y se conecta a la entrada al circuito PWM diseñado. En el canal 2 se midió la señal continua a la salida del amplificador operacional (ADCinVariador), ajustando el potenciómetro *RV1* del esquema de la Figura 4.19, para obtener 3,5 V cuando el ciclo de trabajo es 100 %.

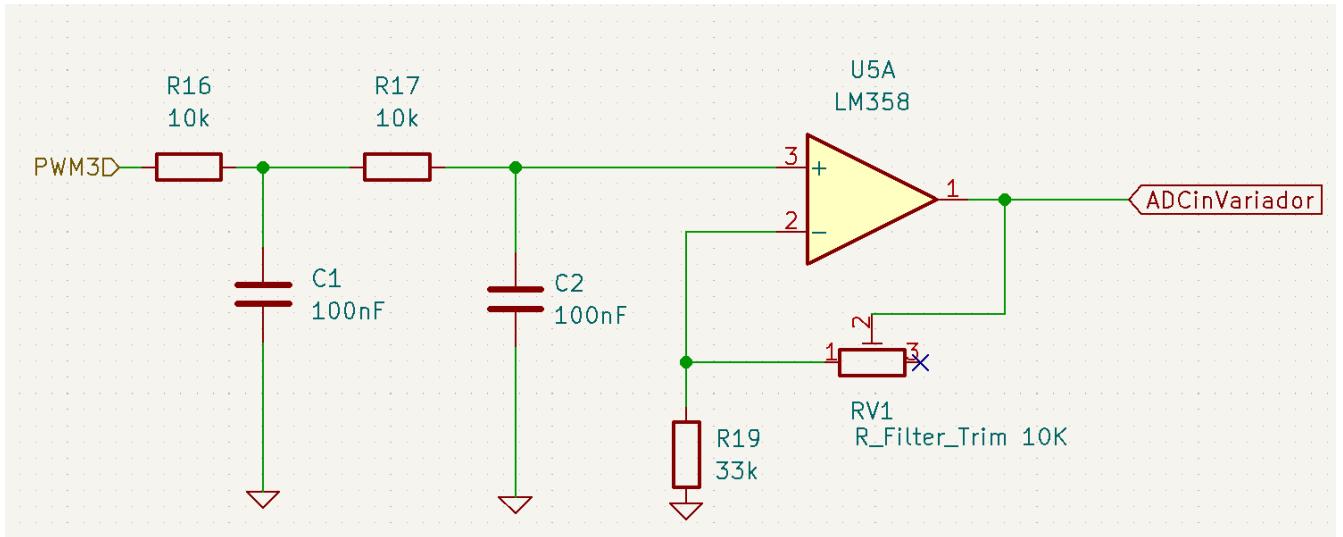


Figura 4.19: Esquemático del circuito PWM que controla la velocidad del túnel de viento por programación.

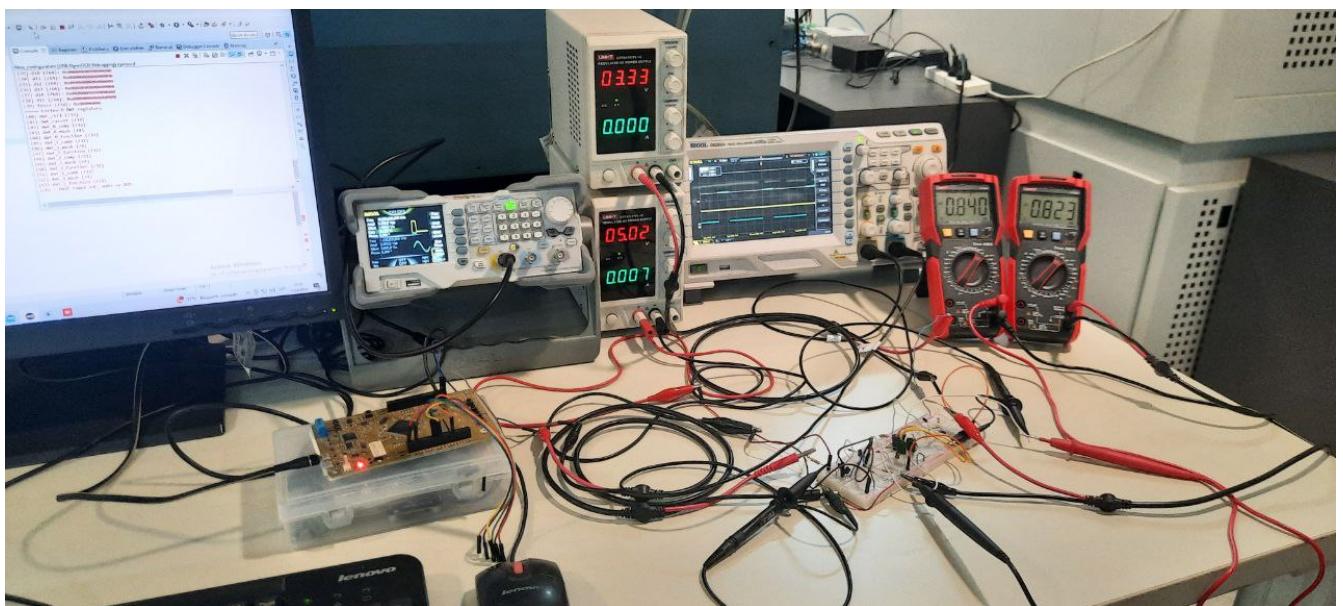


Figura 4.20: Banco de medición para caracterizar el circuito PWM.

Los resultados de las mediciones realizadas con el osciloscopio se muestran en las Figuras 4.21, 4.22, 4.23, 4.24 y 4.25, equivalente a ciclo de trabajo de 1%, 25%, 50%, 75% y 100% respectivamente.



Figura 4.21: Medición de la señal PWM (rojo) con un ciclo de trabajo de (1 %) y la señal continua a la salida del amplificador operacional (azul) iguala 53,5 mV.

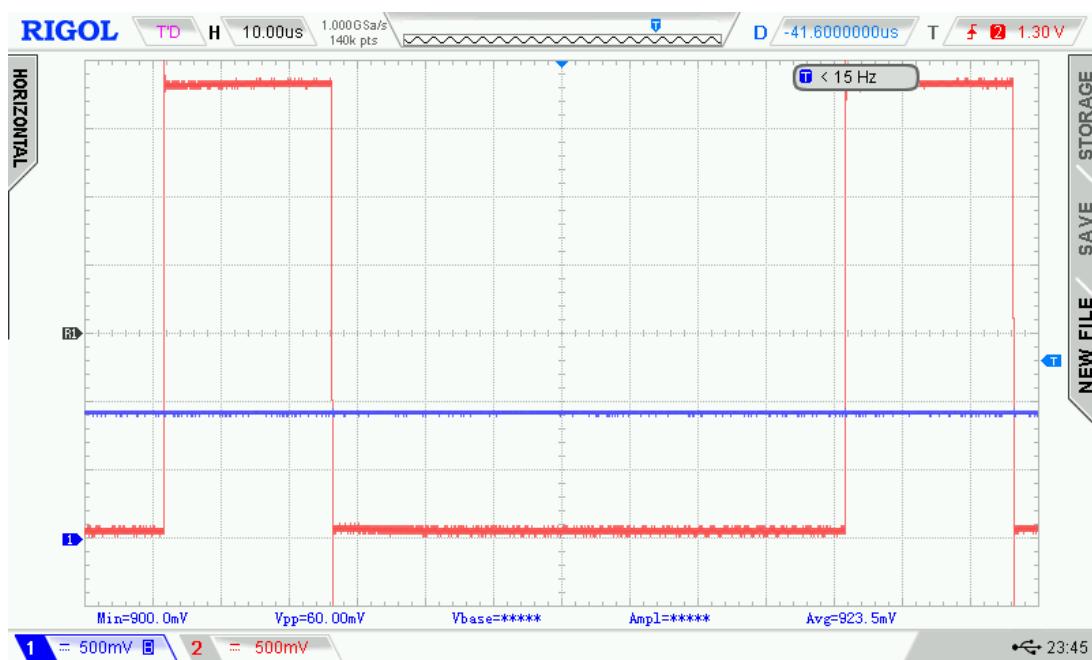


Figura 4.22: Medición de la señal PWM (rojo) con un ciclo de trabajo de (25 %) y la señal continua a la salida del amplificador operacional (azul) iguala 900 mV.

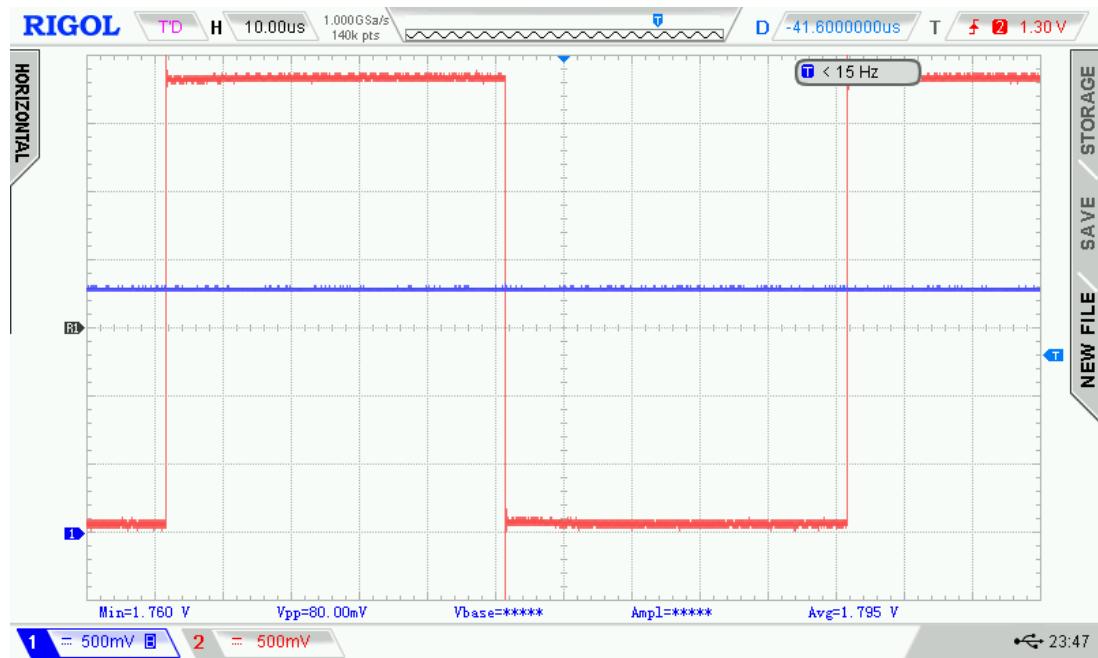


Figura 4.23: Medición de la señal PWM (rojo) con un ciclo de trabajo de (50 %) y la señal continua a la salida del amplificador operacional (azul) iguala 1,8 V.

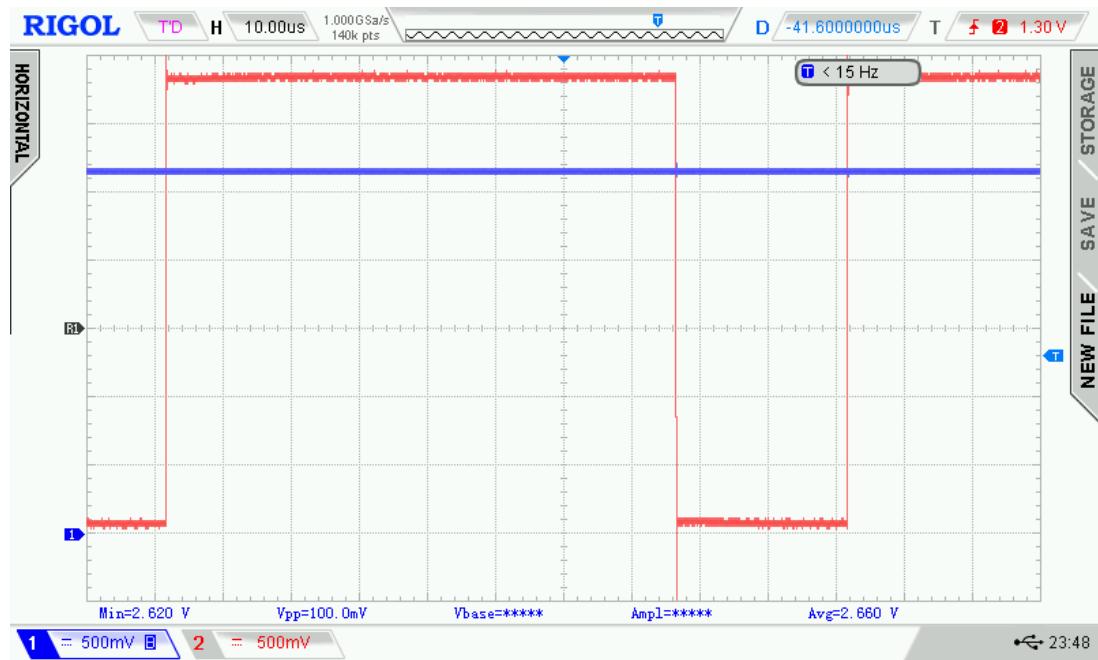


Figura 4.24: Medición de la señal PWM (rojo) con un ciclo de trabajo de (75 %) y la señal continua a la salida del amplificador operacional (azul) iguala 2,65 V.



Figura 4.25: Medición de la señal PWM (rojo) con un ciclo de trabajo de (100 %) y la señal continua a la salida del amplificador operacional (azul) iguala 3,56 V.

Luego de caracterizar el circuito PWM, se lo llevó al túnel de viento y se conectó la tensión de la salida del operacional al pin de entrada *VADC* del variador de velocidad. A continuación, se realizó un barrido del ciclo de trabajo desde 0 hasta 255 (0 % a 100 %), midiendo con un voltímetro, UNI-T modelo UT89X, la tensión entregada al túnel y, con el anemómetro VAISALA WMT700 dentro del túnel, la velocidad en metros por segundo obtenida para cada ciclo de trabajo. En la tabla 4.2 se muestran los resultados de un barrido ascendente, y en la tabla 4.3, los resultados de un barrido descendente.

Ciclo de trabajo [0 - 255]	VadcTunnel [mV]	Viento [m s^{-1}]
0	4,2	1
1	16,6	1,1
2	30,3	1,2
31	427	4,5
63	864	7,8
94	1291	11,1
127	1745	14,7
158	2170	18
191	2621	21,5
222	3048	24,5
255	3500	26,3

Tabla 4.2: Mediciones para ciclos de trabajo en modo Ascendente, VadcTunnel y la velocidad del viento.

Ciclo de trabajo [0 - 255]	VadcTunnel [mV]	Viento [m s^{-1}]
255	3501	25,9
222	3048	24,5
191	2622	21,3
158	2170	18
127	1744	14,7
94	1291	11,1
63	865	7,8
31	426	4,5
2	30	1,3
1	16	1,1
0	4	1,0

Tabla 4.3: Mediciones para ciclos de trabajo en modo Descendente, VadcTunnel y la velocidad del viento.

A partir de los resultados obtenidos para el ciclo de trabajo versus la tensión en el variador y el ciclo de trabajo versus la velocidad del viento medida con el anemómetro, se determina una velocidad máxima de viento de 26 m s^{-1} para una tensión de 3,5 V. Además, dado que se tienen 255 niveles de tensión, se obtiene una resolución en tensión de $\frac{3,5 \text{ V}}{255} = 13,7 \text{ mV}$ y una resolución en velocidad del viento de $\frac{26 \text{ m s}^{-1}}{255} = 0,101 \text{ m s}^{-1}$. Con estas mediciones y la caracterización del circuito PWM, éste quedó listo para ser integrado en un PCB. Este circuito será la interfaz electrónica para un controlador PID, el cual se explica en la sección 4.3.4.

4.1.5. Circuito de adquisición de tensión del variador

En esta sección se diseñó un circuito que permite tomar muestras de la tensión entregada al variador del túnel de viento. El circuito funciona como un voltímetro que mide la tensión instantánea y la envía a un canal analógico-digital (ADC_CH2) de la EDU-CIAA.

Debido a que la señal de continua, para un ciclo de trabajo del 100 %, es de 3,5 V, fue necesario adaptar esta señal a 3,3 V, ya que esta es la tensión máxima de entrada para los canales ADC de la EDU-CIAA.

4.1.5.1. Simulación del circuito

En la Figura 4.26 se muestra el diseño del circuito, la señal de continua VadcTunnel se conecta al segundo operacional del chip LM358, en modo seguidor para adaptar la impedancia y no cargar a la etapa anterior, luego a la salida del operacional se conecta un divisor de tensión para adaptar la tensión a 3,3 V, además se agrega una etapa de protección con dos diodos conectados entre la fuente de 3,3 V y GND para limitar el voltaje en caso de que la tensión a la salida del operacional sea superior 3,3 V. Finalmente, se pone una resistencia de $1\text{k}\Omega$ para limitar la corriente en VadcCIAA.

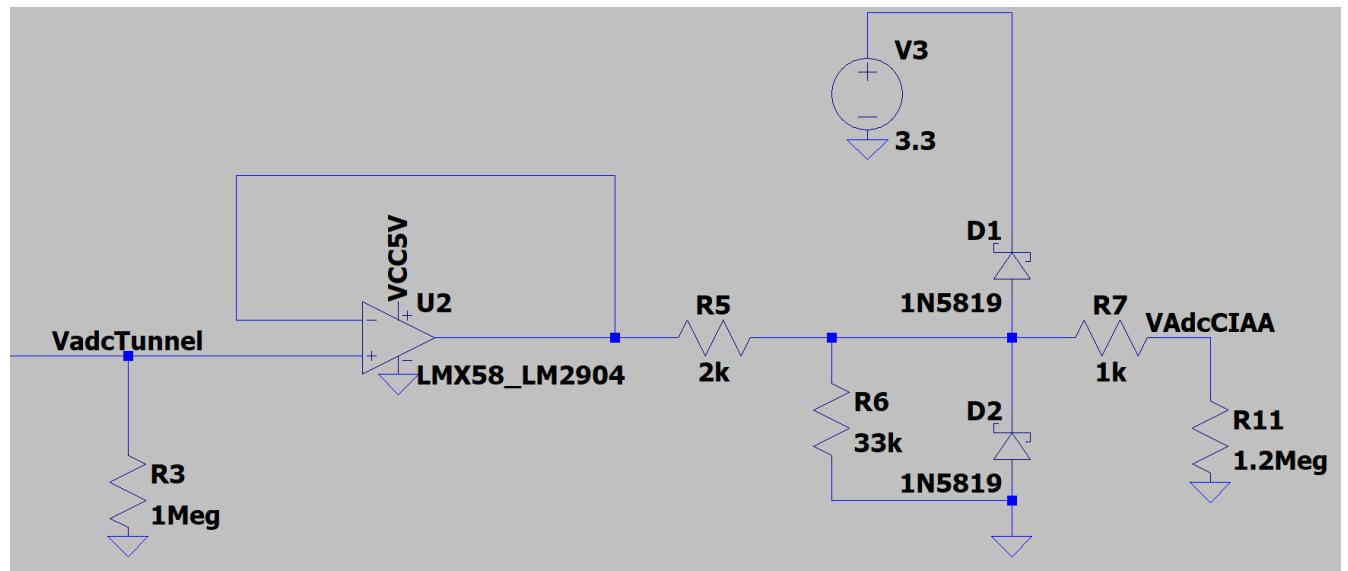


Figura 4.26: Circuito diseñado para tomar muestras de la señal que se suministra al variador de velocidad y se las envía a una canal analógico-digital de la EDU-CIAA.

Los resultados de la simulación del circuito descrito se muestran en la Figura 4.27 para un ciclo de trabajo del 100 % y en la Figura 4.28 para un ciclo de trabajo del 55 %. En ambos casos, se puede apreciar que la señal vadcCIAA (azul) está por debajo de la señal vadcTunel (verde), lo cual garantiza que no se superen los 3,3 V a la entrada del pin analógico-digital de la placa de

desarrollo.

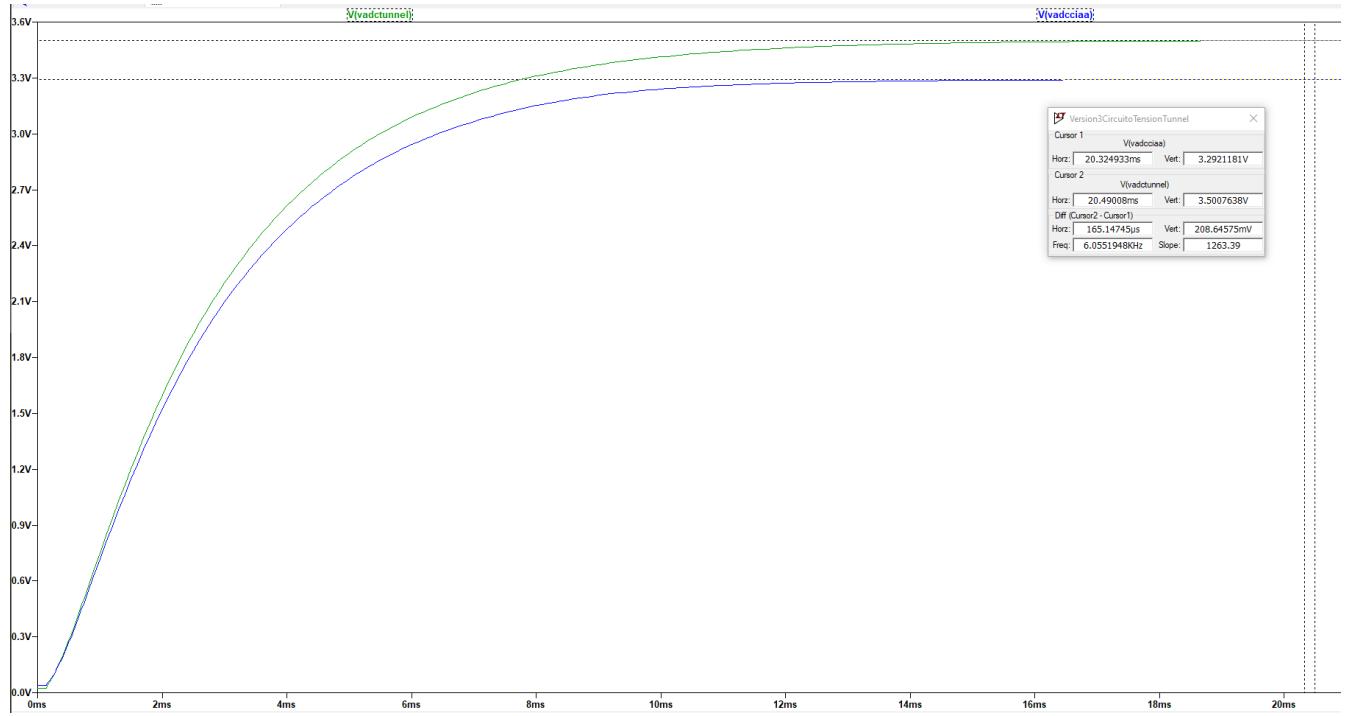


Figura 4.27: Señales simuladas a la salida del variador de velocidad y a la entrada del pin analógico-digital de la EDU-CIAA para un ciclo de trabajo del 100 %.

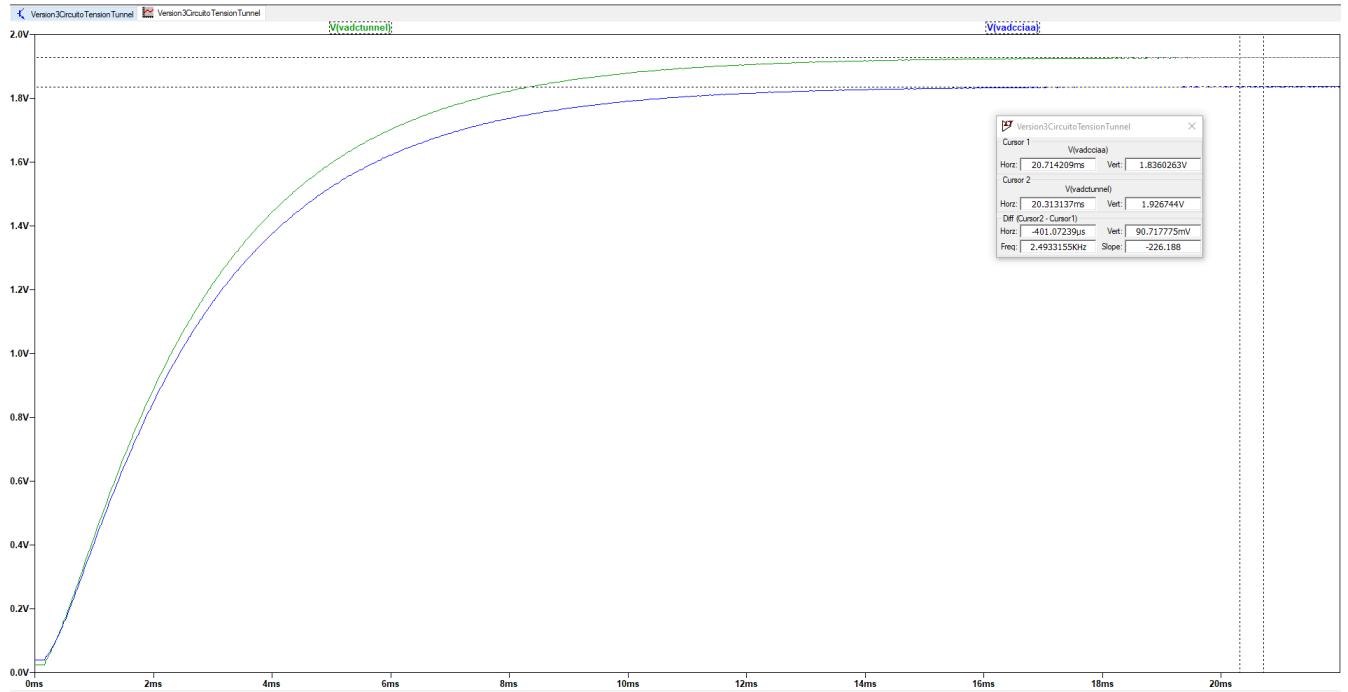


Figura 4.28: Señales simuladas a la salida del variador de velocidad y a la entrada del pin analógico-digital de la EDU-CIAA para un ciclo de trabajo del 55 %.

4.1.5.2. Implementación del circuito

Se implementó el circuito de la Figura 4.29. Para ajustar los niveles de tensión de 3,5 V a 3,3 V, se agregó en el divisor de tensión, un resistor variable (trimmer) de $RV2 = 10\text{ k}\Omega$. Luego se conectó los diodos a la fuente de 3,3 V de la EDU-CIAA y se agregó una resistencia limitadora de tensión de $R18 = 1\text{ k}\Omega$, esta señal de continua se muestrea con una resolución de $\frac{3,3\text{ V}}{1024} = 3,22\text{ mV}$ en el canal ADC_CH2 de la placa de desarrollo.

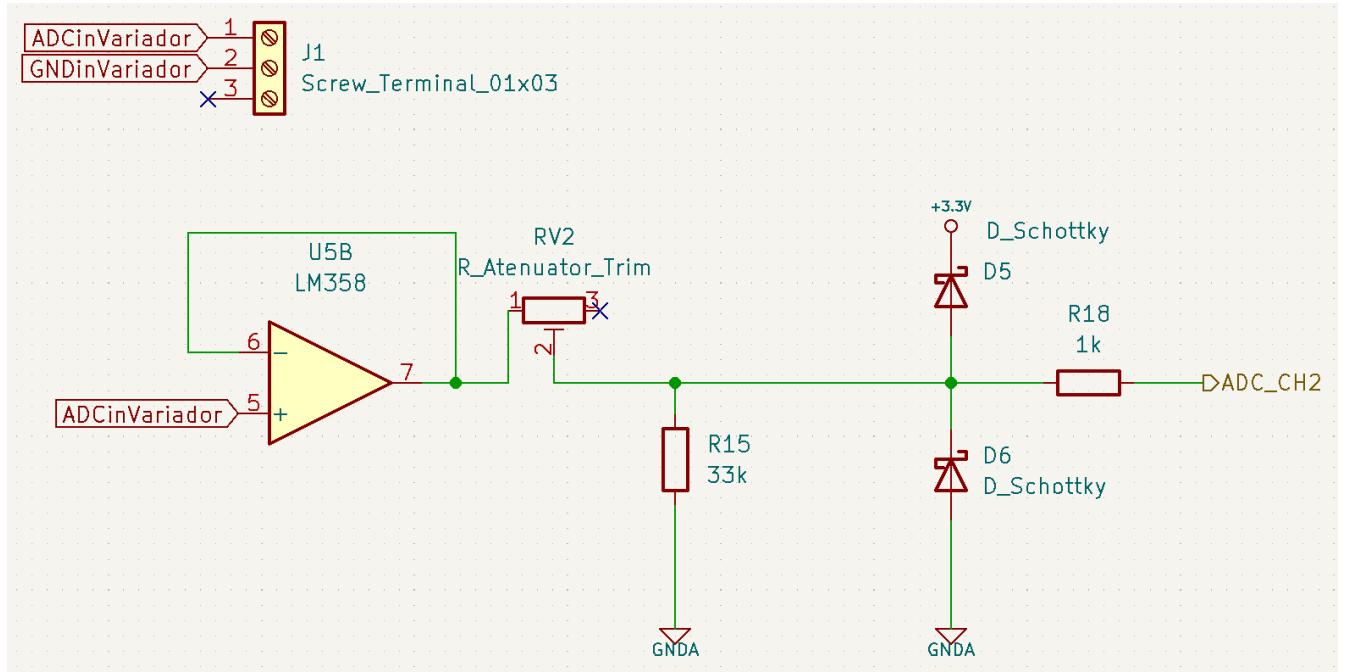


Figura 4.29: Esquemático de la implementación del circuito que toma muestras de la señal continua que ingresa al variador de velocidad.

Se conectó dos voltímetros, uno en la señal de entrada del variador y otro en la señal que ingresa al ADC de la EDU-CIAA, los resultados de estas mediciones se observan en las tablas 4.4 para un barrido ascendente y 4.5 para un barrido descendente.

Ciclo de trabajo [0 - 255]	VadcTunnel [mV]	VadcCIAA [mV]
0	4,2	20
1	16,6	16,7
2	30,3	37,3
31	427	421,3
63	864	856
94	1291	1280
127	1745	1725
158	2170	2146
191	2621	2593
222	3048	3013
255	3500	3364

Tabla 4.4: Mediciones a la entrada pin *VADC* del variador y en el pin *ADC_CH2* de la EDU-CIAA, para un barrido ascendente del ciclo de trabajo.

Ciclo de trabajo [0 - 255]	VadcTunnel [mV]	VadcCIAA [mV]
255	3501	3364
222	3048	3012
191	2622	2593
158	2170	2146
127	1744	1725
94	1291	1277
63	865	855
31	426	422
2	30	38
1	16	27
0	4	21

Tabla 4.5: Mediciones a la entrada pin *VADC* del variador y en el pin *ADC_CH2* de la EDU-CIAA, para un barrido descendente del ciclo de trabajo.

Este circuito permite monitorear por software los niveles de tensión que estamos entregando al variador de velocidad y su correspondiente ciclo de trabajo.

4.1.6. Circuito de LEDs indicadores

Se diseñó el circuito de la Figura 4.30 para encender y apagar LEDs como indicadores de la recepción de datos de los anemómetros patrón y bajo calibración. Se utilizaron dos LEDs de colores diferentes: uno verde y otro amarillo. El LED verde se asoció al canal RS-485-1 y el LED

amarillo al canal RS-485-2.

Cada circuito permite que el LED correspondiente parpadee cada vez que se recibe un dato del anemómetro, con un intervalo predeterminado por software. Si el LED no parpadea, indica que el anemómetro no está enviando datos.

El diseño de cada circuito incluye un transistor NPN y dos resistencias, conectados a un diodo LED. La señal de entrada llega al resistor $R_4 = 1\text{k}\Omega$, el cual está conectado a la base del transistor. El colector del transistor está conectado a una fuente de 5 V, mientras que el emisor está conectado a una resistencia limitadora de corriente $R_5 = 39\Omega$ y luego al ánodo del LED, con su cátodo a masa. Ambos circuitos son prácticamente idénticos, diferenciándose únicamente en los colores de los LEDs y los canales RS-485 a los que están conectados.

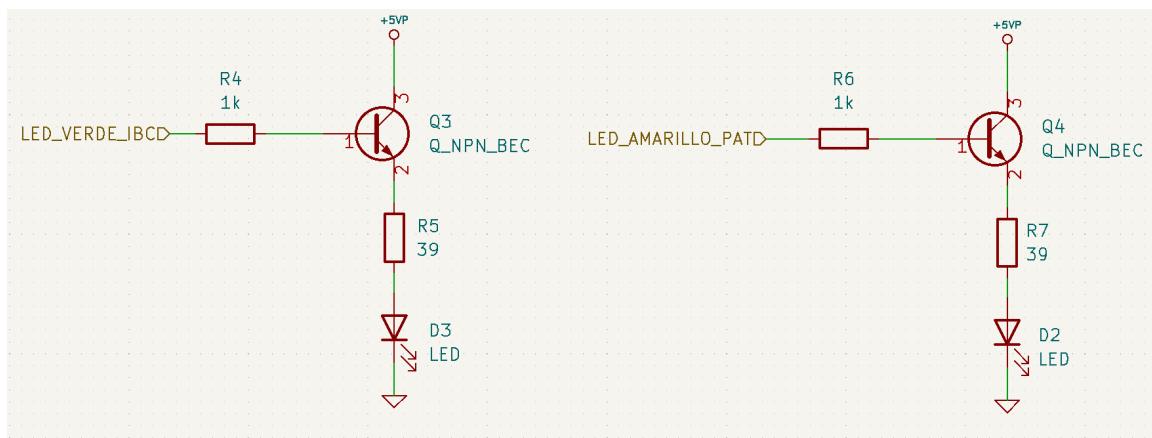


Figura 4.30: Esquemático de los circuitos que encienden y apagan dos LEDs, verde y amarillo, cada vez que se recibe un dato de los anemómetros, tanto del patrón como del que está bajo calibración.

En La Figura 4.31 se muestra un circuito similar al previamente descrito, en el que se utiliza un LED dual como indicador de la conectividad del sistema de la placa de desarrollo con el servidor Websocket descrito en la sección 5.1.2. El LED dual se enciende en rojo cuando el sistema de la placa de desarrollo está desconectado del servidor. Al establecerse la conexión a través de Ethernet, el LED cambia a verde. Este cambio de color ocurre automáticamente según el estado de la conectividad con el servidor. El LED dual tiene tres pines: dos ánodos y un cátodo común conectado a masa. Los ánodos están conectados a circuitos que controlan el encendido y apagado del LED según las señales de los pines digitales de la placa de desarrollo.

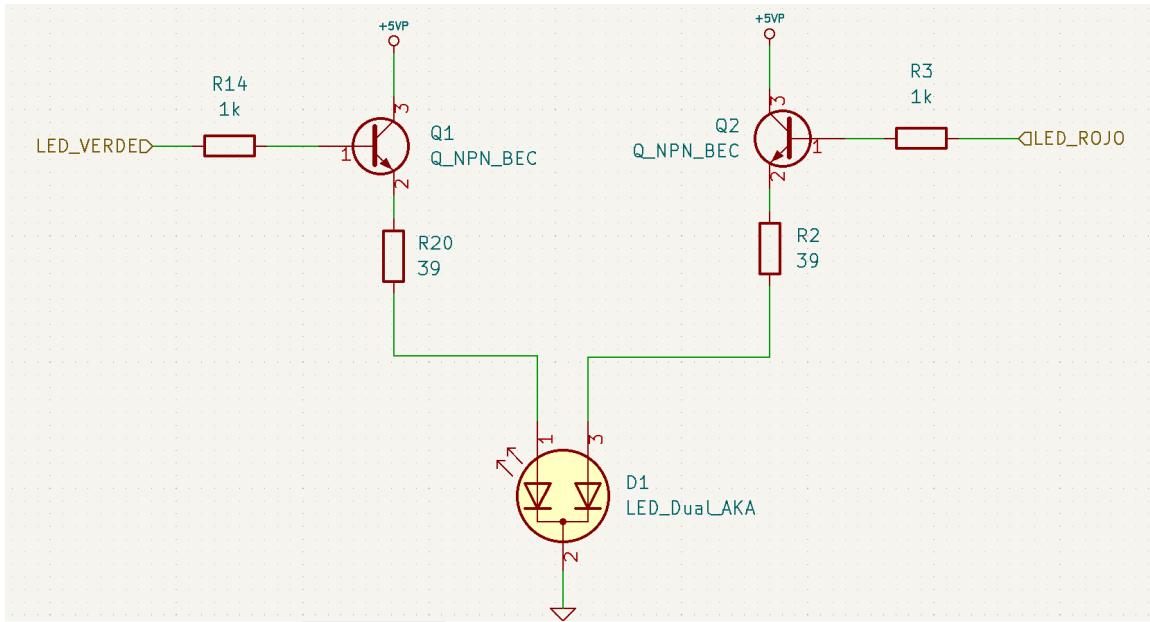


Figura 4.31: Esquemático del circuito de un LED dual que se enciende en rojo cuando el sistema no está conectado al servidor WebSocket y en verde cuando hay una conexión estable con el servidor.

4.2. Diseño y construcción del PCB

En las secciones anteriores se ha descrito el diseño y funcionamiento electrónico de los distintos módulos y circuitos que forman parte del datalogger. Para integrar todo en un solo lugar, se diseñó un shield (poncho) que se conecta sobre la EDU-CIAA. Para ello, se trabajó con la lista de templates que ofrece el proyecto CIAA [12]. Este template tiene un diseño acorde con las dimensiones de la placa EDU-CIAA.

Para el desarrollo del PCB, se utilizó el software KiCad. Primero se cargó el template de la Figura 4.32 que incluye una hoja jerárquica con los pines disponibles en los conectores P1 y P2 y luego se agregaron etiquetas jerárquicas por cada pin utilizado en este desarrollo.

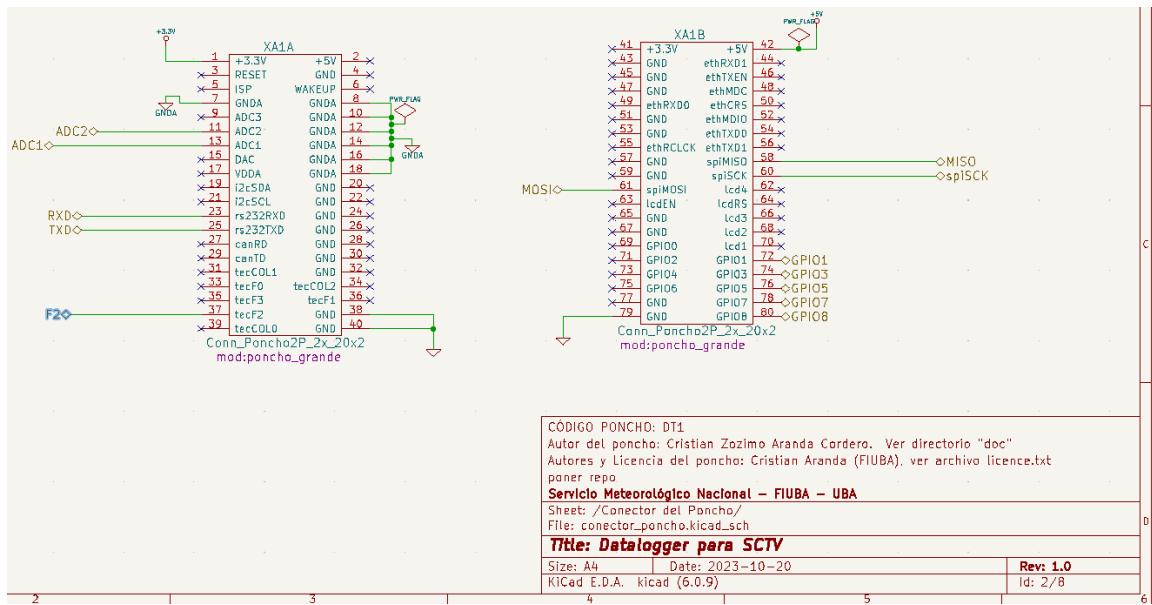


Figura 4.32: Template disponible para la realización de ponchos para la EDU-CIAA.

Por cada módulo, se agregó una hoja jerárquica que contiene el circuito esquemático como se indica en la Figura 4.33. En total, se dividió en seis hojas jerárquicas: una para el módulo RS485, otra para el módulo Ethernet W5100, otra para el circuito PWM, otra para la fuente de alimentación (power supply), otra para los LEDs y una más para el circuito que mide la tensión en el variador del túnel.

Además, se diseñaron los símbolos para los módulos W5100, RS485-TTL y el convertidor DC-DC usando el editor de símbolos de KiCad, ya que estos no vienen precargados en el software. También se diseñaron sus respectivos footprints con el editor de huellas, tomando las dimensiones y la distribución de pines de cada módulo para que se puedan agregar correctamente al PCB.



Figura 4.33: Diseño del esquemático que integra los módulos desarrollados.

En la Figura 4.34 se muestra el diseño del PCB de doble capa, con la capa inferior (*Bottom*) en azul y la capa superior (*Top*) en rojo. Las dimensiones de los bordes recortados y la tira de 20 pines en la parte superior e inferior forman parte del template; el resto del circuito se diseñó en función de las necesidades de cada módulo.

Las pistas tienen un ancho de entre 30 y 40 mils. Se agregaron vías de 2 mm de diámetro con un agujero de 1 mm para conectar la capa superior con la capa inferior. Además, los pads tienen un tamaño de 2 mm.

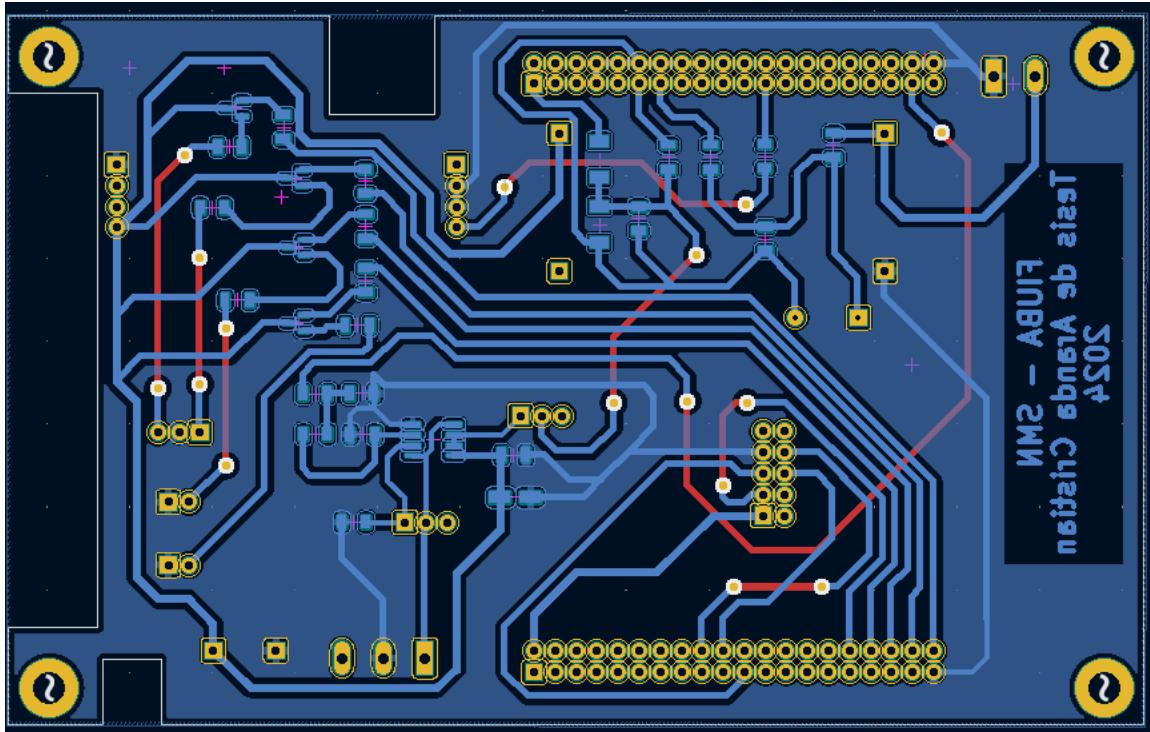


Figura 4.34: Diseño PCB del Shield (poncho) que contiene todos los módulos descritos en las secciones anteriores.

En la figura 4.35 se muestra la vista 3D de capa *Bottom* donde se diseñó los circuitos PWM, el adquisidor de tensión de la fuente y el adquisidor de tensión del variador, utilizando componentes SMD. Se emplearon diodos, capacitores y resistores con tamaño de paquete 1206 y para los transistores NPN MMBT3904 y el operacional LM358 se cargaron sus footprints SMD de la biblioteca de KiCad.

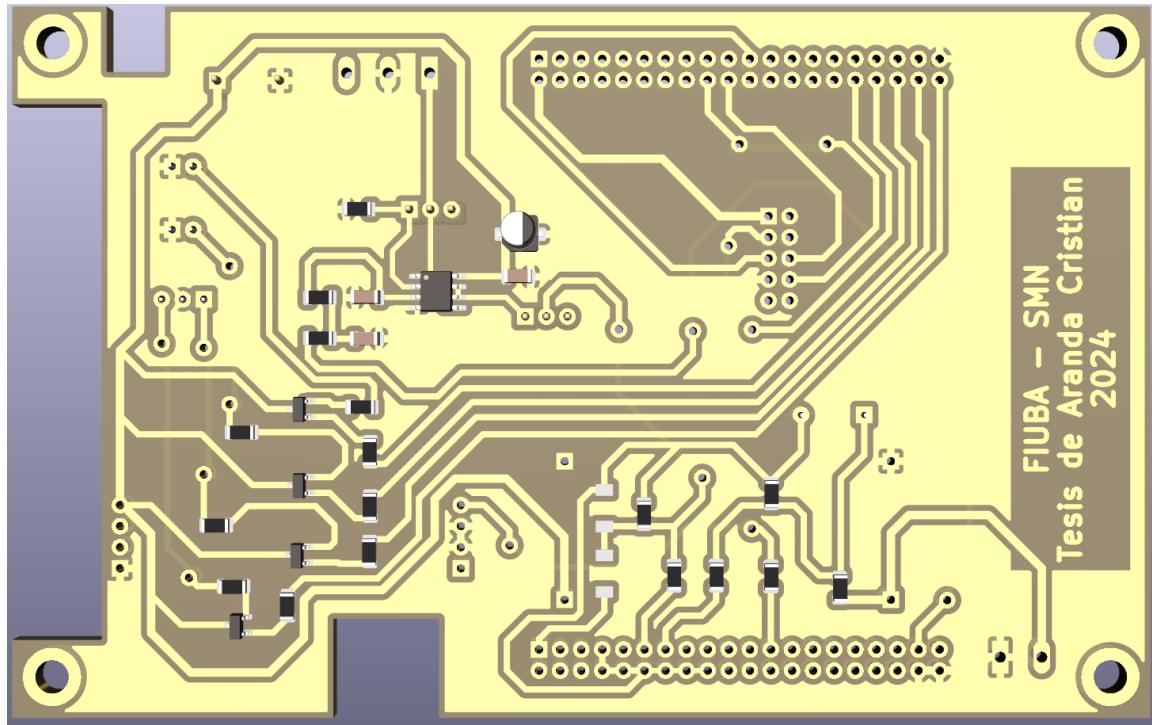


Figura 4.35: Vista en 3D de la capa BOTTOM.

Por otro lado, en la capa superior (TOP) se han añadido los tres LEDs con montaje THT y se han dispuesto dos tiras de pines 3x1 para la conexión física de los dos potenciómetros variables (presets). Además, debido a la falta de disponibilidad del diodo Zener SMD de 3,3 V para proteger el ADC-CH1 de la placa EDU-CIAA, se adquirió el mismo componente pero con montaje THT. También se han incorporado los footprints de los módulos previamente diseñados del módulo Ethernet, módulo RS485-UART (que incluye su propia bornera para la conexión del anemómetro bajo calibración), y módulo DC-DC step-down. Además, se han agregado una bornera de 2x1 para alimentar el circuito con 12 V, y otra bornera 3x1 para la conexión con el variador del túnel de viento.

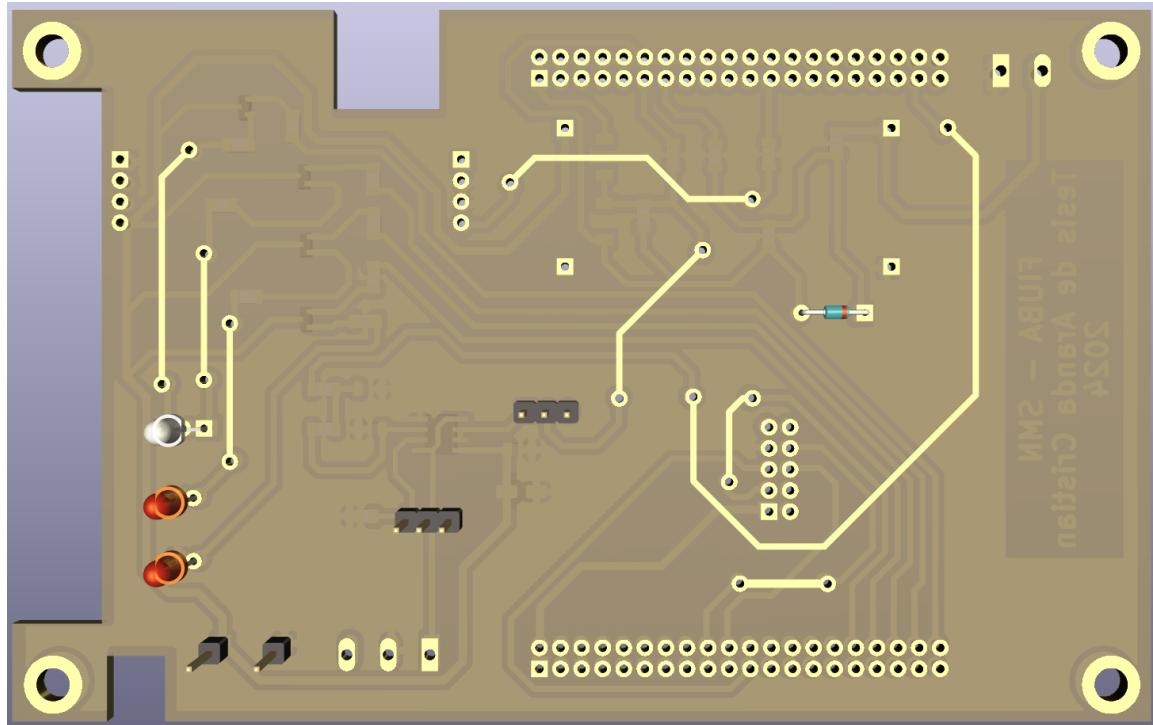


Figura 4.36: Vista en 3D de la capa TOP.

Para la construcción del *shield*, se seleccionó una placa simple FAS por su disponibilidad en el Servicio Meteorológico Nacional (SMN). Debido a restricciones de tiempo, se optó por ensamblar la placa en el laboratorio del SMN, siguiendo estos pasos:

Inicialmente, se empleó una insoladora de tubos UV y papel fotosensible para transferir el diseño a la placa. Los negativos del diseño se imprimieron en papel transparente.

Después, se utilizó ácido perclórico para retirar el cobre sobrante. Finalizado este procedimiento, se cortó la placa a la medida deseada y se perforaron los *pads* con una herramienta *Dremel*. Se aplicó una capa de máscara antisoldante para facilitar la soldadura de los componentes y proteger la placa. Dado que la placa era de tipo simple FAS, se añadieron manualmente las pistas de la capa superior (*Top*) usando pequeños cables. El proceso concluyó con la soldadura de todos los módulos y componentes. Las Figuras 4.37 y 4.38 ilustran el resultado final del montaje.

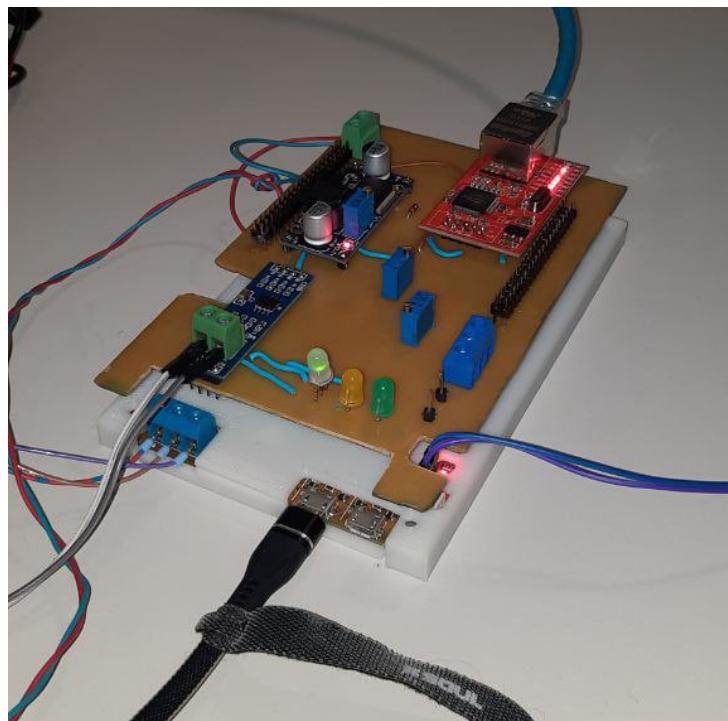


Figura 4.37: Shield conectado a la placa EDU-CIAA

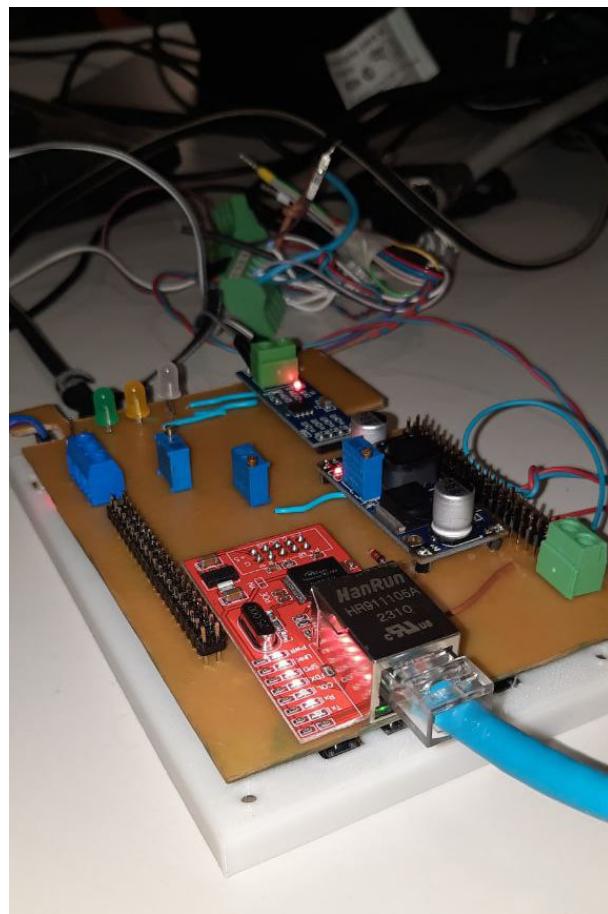


Figura 4.38: Otra vista del Shield diseñado para esta aplicación.

Como se indica en la Figura 4.39, se conectó el USB para programar y depurar la placa. Se conectó en la bornera azul de la EDU-CIAA el sensor patrón y en la bornera verde del módulo RS485 se conectó el anemómetro bajo calibración. En la parte superior de la figura está una bornera azul, ahí se conectaron los tres cables que vienen del variador de velocidad del túnel. En el lado izquierdo, se conectó un cable UTP que está conectado a la red local del SMN, y en la bornera verde que está abajo a la izquierda, se conectó la alimentación de 12 V.

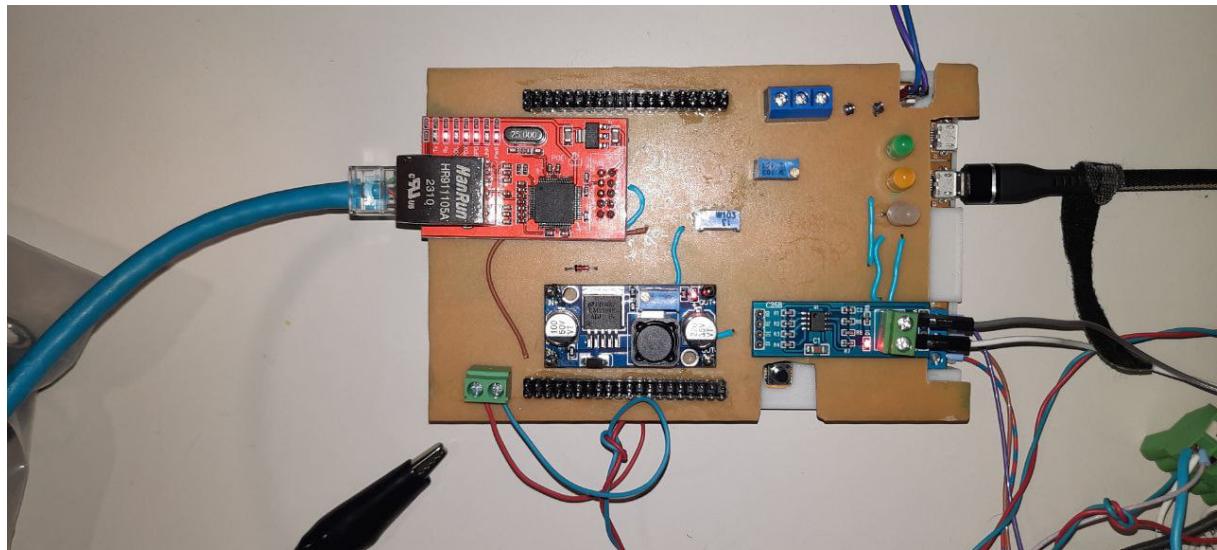


Figura 4.39: Vista superior de la placa Shield conectada a la EDU-CIAA

4.3. Desarrollo del firmware

El proyecto CIAA brinda un framework para programar las placas de desarrollo. La última versión, el [firmware_v3](#) [19], es un proyecto basado en makefile para desarrollar firmware en C/C++. En particular se utilizó la biblioteca sAPI, que ofrece ejemplos para programar los periféricos del microcontrolador.

Además, el proyecto incluye un lanzador de aplicaciones Figura 4.40, que trae integrado el IDE GNU MCU Eclipse como entorno de programación, OpenOCD para programación y depuración, drivers para conectar la EDU-CIAA a la PC, entre otras herramientas.



Figura 4.40: Lanzador de aplicaciones integradas para el desarrollo de firmware para la EDU-CIAA.

Se clonó el repositorio que contiene el `firmware_v3` de GitHub en el IDE y se abrió un nuevo proyecto llamado **Datalogger**, como se indica en la Figura 4.41.

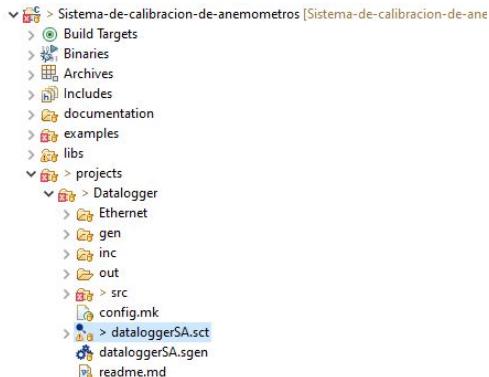


Figura 4.41: Estructura de carpetas, del proyecto Datalogger

En particular, la lógica del programa se implementó mediante el uso de máquinas de estado (*statecharts*) de Harel, empleando el software Itemis Yakindu [26]. Los *statecharts* de Harel amplían los diagramas de estados tradicionales incorporando conceptos de modularidad, jerarquía y estructura organizacional. Estos diagramas se caracterizan por el uso de estados compuestos y subdiagramas que preservan la claridad y la organización, y facilitan la ejecución concurrente de múltiples submáquinas de estado a través de regiones ortogonales. Además, los eventos habilitan la comunicación por difusión, mientras que las transiciones condicionales, los estados históricos, la lógica temporal y las acciones de entrada/salida potencian la funcionalidad de los *statecharts*,

optimizando la modelización de sistemas complejos de manera eficiente. La Tabla 4.6 presenta un análisis comparativo entre las máquinas de estado simples, como las de Mealy y Moore, y las máquinas de estado más avanzadas de UML y Harel.

Descripción	Mealy	Moore	Harel	UML
Estados y transiciones	✓	✓	✓	✓
Las transiciones producen salida	✓	✗	✓	✓
Los estados producen salida	✗	✓	✓	✓
Profundidad (jerarquías, estados compuestos)	✗	✗	✓	✓
Ortogonalidad (submáquinas de estado paralelas)	✗	✗	✓	✓
Comunicación por difusión (eventos)	✗	✗	✓	✓
Historial, acciones, retrasos, tiempos de espera, condiciones	✗	✗	✓	✓

Tabla 4.6: Diferencias entre los tipos de máquinas de estados.

Se crearon doce regiones que trabajan de forma concurrente (de forma ortogonal), como se muestra en la Figura 4.42, y se comunican entre sí mediante eventos internos. A cada región se le asigna una prioridad en función de su posición; las que están más arriba tienen mayor prioridad de ejecución respecto a las de más abajo.

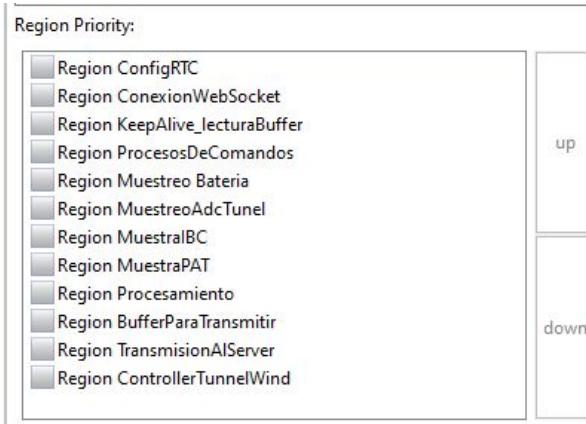


Figura 4.42: Lista de regiones con su respectiva prioridad.

Se utilizó una licencia gratuita de Yakindu, integrándola con el IDE de Eclipse para editar y generar el código a partir de los *statecharts*. En particular, se configuró el proyecto en Yakindu C/C++ para poder agregar código en C a las máquinas de estado, incluyendo la importación de archivos .h, declaración de variables, punteros globales y constantes, como se ejemplifica en el Código 4.1.

```

1 import: "inc/common.h"
2 import: "inc/da_acquisition.h"
3 import: "inc/da_processing.h"

```

```

4 import: "inc/da_transmision.h"
5 import: "inc/da_rtc.h"
6 import: "inc/da_configDatalogger.h"
7 import: "inc/controlTunnel.h"
8 /*Datalogger para Sistema anemometrico con EDU-CIAA-NXP */
9 interface:
10 //Variables
11 var startMesuare:uint8_t = FALSE
12 //Puntero
13 var ptrStartMesuare:ptrUnt8_t
14 var viTeclaOprim: uint16_t
15 //Constantes
16 const TRUE: uint8_t =1
17 const FALSE: uint8_t =0
18 const Tec1: uint16_t = 1<<0
19 const Tec2: uint16_t = 1<<1

```

Código 4.1: Configuración principal del modelo statechart dataloggerSA.

También se agregaron eventos internos que permiten transicionar del estado de una región a otro estado de otra región (*raiseEvent* en inglés), como se muestra en el Código 4.2.

```

1 internal:
2 //En forma interna tengo seniales para vincular las maquinas de estado
3 event siDatosListosParaBdtos
4 event siDatosProcesado
5 event siMuestrasNuevas
6 event siStartMeasure
7 event siKeepAlive
8 event siProcesMsjServer

```

Código 4.2: Declaración de eventos internos que permiten transicionar entre regiones del statechart.

4.3.1. Configuración y conexión con los servidores

Como se detalló en la sección 4.1.3, se utilizó el módulo W5100 para transferir datos desde el datalogger hacia los servidores y recibir datos de los mismos. El módulo cuenta con cuatro socket independientes y permite configurar un sistema embebido como servidor o cliente, como se muestra en la Figura 4.43.

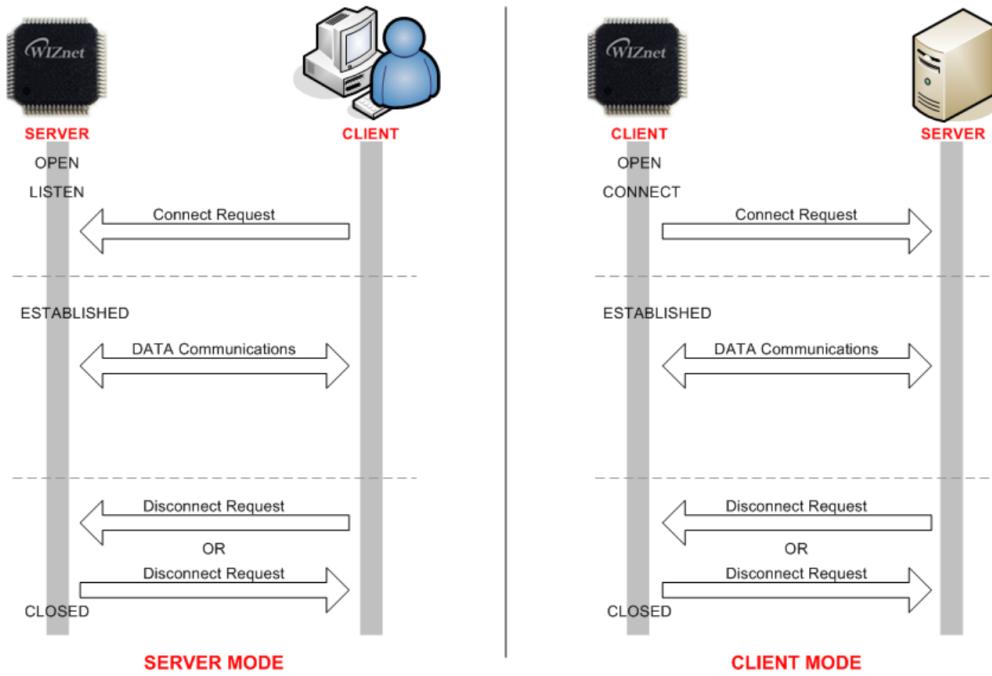


Figura 4.43: Modos de configuración para el módulo W5100.

El datalogger se comunica con los servidores en modo cliente (apertura activa), enviando una solicitud de conexión a un servidor, estableciendo la comunicación e intercambiando datos. Luego, realiza una solicitud de desconexión y se cierra el enlace. Utiliza, en la capa de transporte, el protocolo TCP y, en la capa de red, el protocolo IP. En la capa de aplicación, se utilizó el protocolo WebSocket para intercambio de datos y el protocolo NTP para configurar el RTC del datalogger. Dado que estos protocolos no están incluidos en la biblioteca del chip W5100, se implementó la conexión, el intercambio de datos y la desconexión de los servidores mediante sockets, como se indica en el diagrama de flujo de la figura 4.44.

Un socket es una interfaz que permite la comunicación entre dispositivos a través de redes. En términos simples, un socket es un punto final para enviar y recibir datos en una red, y está compuesto por una dirección IP y un número de puerto.

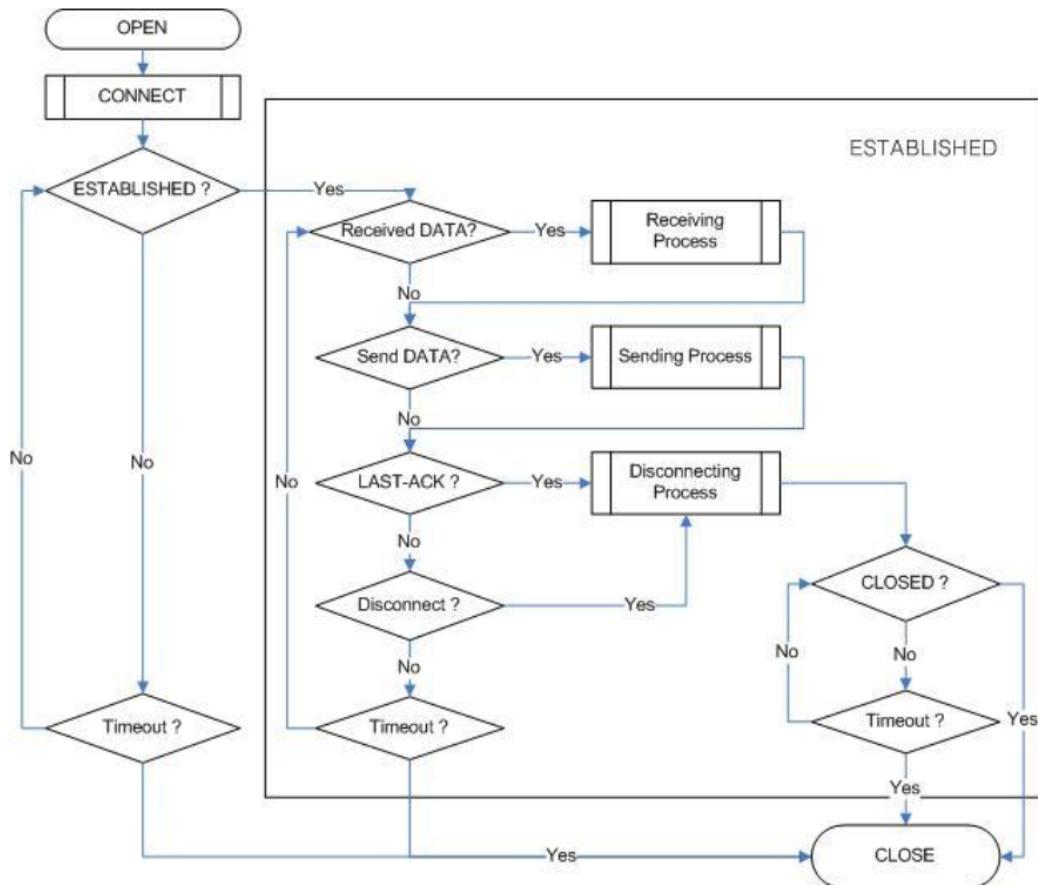


Figura 4.44: Diagrama de flujo para establecer una comunicación, en modo cliente, con un servidor utilizando sockets.

4.3.1.1. Servidor NTP

Para la configuración del RTC del datalogger, se estableció una conexión con los servidores NTP del Instituto Nacional de Estándares y Tecnología (*NIST*). Se seleccionó una de las múltiples IPs disponibles, específicamente la dirección 128.138.141.172 a través del puerto 13. Esta conexión retorna una cadena de caracteres, 59992 23-02-17 17:47:01 00 0 0 831.5 UTC(*NIST*)*, que se procesó e integró en la estructura del RTC del datalogger. Para gestionar y validar esta comunicación con el servidor NTP, se configuró una región específica dentro del sistema. Las variables declaradas en el Código 4.3 se utilizan para validar cada transición dentro de la máquina de estados ilustrada en la Figura 4.45.

```

1 // Variables para la conexión al servidor NTP
2 var ConfigNtpSocket: uint8_t = FALSE
3 var setRtc: uint8_t = FALSE
4 var ConfigNTP: uint8_t = FALSE

```

Código 4.3: Declaración de variables booleanas para validar la conexión con el servidor NTP.

La máquina comienza en el estado **INIT** y luego de un período de `DelatInit` segundos, pasa al estado **CONFIG_SOCKET**. En este estado, se llama a la función `opConfigNtpSocketViaTCP()`, que se encarga de configurar los parámetros de red del cliente: gateway, dirección MAC, máscara de subred e IP.

Luego, estos parámetros se asignan a uno de los cuatro sockets del módulo W5100, y se configura el protocolo de transporte (TCP en este caso), el puerto, y se abre la conexión para establecer una comunicación con el servidor. Esta función devuelve `True` si la apertura y configuración son correctas, almacenando este resultado en la variable `ConfigNtpSocket`. Si la configuración es exitosa, la máquina de estados transiciona al estado **CONFIG_NTP**.

En el estado **CONFIG_NTP**, se invoca la función `opConfigNtpViaTCP()`, encargada de configurar la conexión con el servidor NTP. Esta función establece la dirección IP y el puerto del servidor y realiza un intento de conexión. Si la conexión es exitosa, la función retorna `True` y el resultado se almacena en la variable `ConfigNTP`. Una vez configurado, el servidor responde con la hora y fecha actual en formato UTC, que se guarda en una variable interna. Posteriormente, la máquina de estados avanza al estado **SET_RTC_VIA_SOCKET**, donde se procesa la información recibida y se actualiza el RTC del datalogger. Finalmente, se cierra la conexión con el servidor NTP pasando al estado **IDLE**, cuando se realiza esta última transición se eleva un evento a través de la señal `siStartMeasure`, la cual será un disparador para iniciar otra máquina de estado en otra región. En caso de fallo en cualquiera de los estados previos, se transiciona al estado **SET_RTC_DEFAULT**, donde se establece una hora predeterminada.

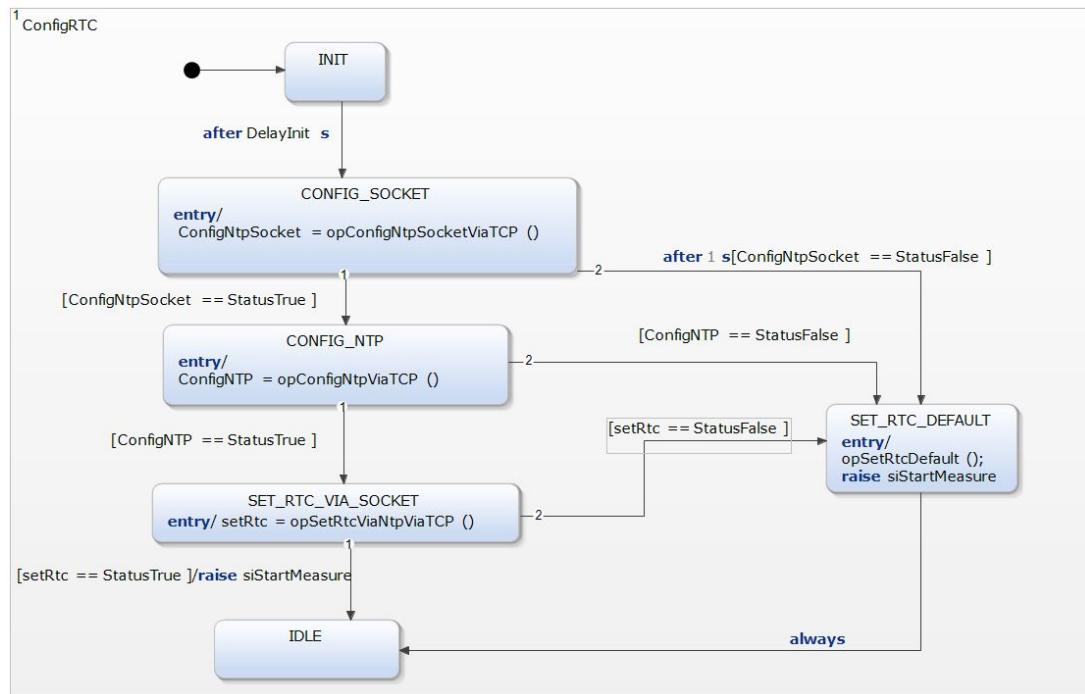


Figura 4.45: Máquina de estados para configurar el RTC del datalogger a través de un servidor NTP.

4.3.1.2. Servidor WebSocket

El protocolo WebSocket es una tecnología de comunicación que permite interacciones bidireccionales (*full-duplex*) entre un cliente y un servidor a través de un solo canal de conexión TCP. Es especialmente útil para aplicaciones web que requieren comunicación en tiempo real, como juegos en línea, chat en vivo y servicios de trading. Para establecer una conexión WebSocket, el cliente envía una solicitud de *handshake* (apretón de manos) al servidor. Esta solicitud sigue el protocolo HTTP con un *upgrade* para solicitar el cambio de protocolo a WebSocket. Aquí se muestra un ejemplo de cómo se ve una solicitud de *handshake* de WebSocket:

```

GET /mychat HTTP/1.1
Host: 10.10.13.153
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMBDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
Origin: http://example.com

```

El servidor, al recibir esta solicitud, responde con una respuesta de *handshake* si acepta la

conexión:

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGWk=
Sec-WebSocket-Protocol: chat
```

La clave `Sec-WebSocket-Accept` es generada por el servidor a partir de la clave `Sec-WebSocket-Key` enviada por el cliente, lo que confirma que el servidor entiende y acepta la solicitud de WebSocket. Una vez establecido el *handshake*, el cliente y el servidor pueden enviar datos en forma de *frames* de WebSocket, que pueden ser datos de texto o binarios.

A Continuación se pasa a explicar como se implementó la comunicación. Para ello se declaran las variables de estado que se muestran en el Código 4.4 y permiten la transición de estados de la región `ConexionWebSocket` que se muestra en la figura 4.46

```
1 // Variables para la conexion al servidor WebSocket
2 ar socketConnect: uint8_t = FALSE
3 var isConnectWebServer: uint8_t=FALSE
4 var buffertempLleno: uint8_t = TRUE
5 var txKeep:uint8_t=FALSE
```

Código 4.4: Declaración de variables booleanas para validar la conexión con el servidor WebSocket.

La máquina de estados inicia en el estado **IDLE**. La transición al estado **CONFIG_SOCKET** se desencadena por el evento de la señal `siStartMeasure`, generado tras la correcta configuración del tiempo por la máquina de estados del servidor NTP. Esto asegura que no se inicie la medición hasta que la hora esté precisamente sincronizada. En el estado **CONFIG_SOCKET**, se invoca la función `opConfigSocket()`, responsable de configurar los parámetros de red del módulo W5100, incluyendo el gateway, la máscara de subred, la dirección MAC y una dirección IP. También se establece el protocolo TCP, asignando un número de puerto al cliente. Tras abrir el socket, la máquina de estados avanza al estado **INI_WEB_SOCKET** sin requerir validación adicional.

En el estado **INI_WEB_SOCKET**, se ejecuta la función `opInitWebSocket()`, que prepara los datos necesarios para la conexión con el servidor WebSocket, como la dirección IP del servidor y el puerto. Esta configuración permite actualizar la variable `socketConnect` a `True`, habilitando el avance al estado **CONNECT_WEBSOCKET**. Aquí, la función `opConnectToWebSocket()` gestiona el envío de la solicitud de *handshake* al servidor y procesa su respuesta. Si la conexión

es exitosa, la variable `isConnectWebServer` se establece en `True`, indicando que la conexión está lista para el intercambio de datos en modo full-duplex.

Si se presenta un fallo en cualquiera de los estados previos, la máquina de estados regresa al estado `CONFIG_SOCKET` para intentar una nueva conexión con el servidor.

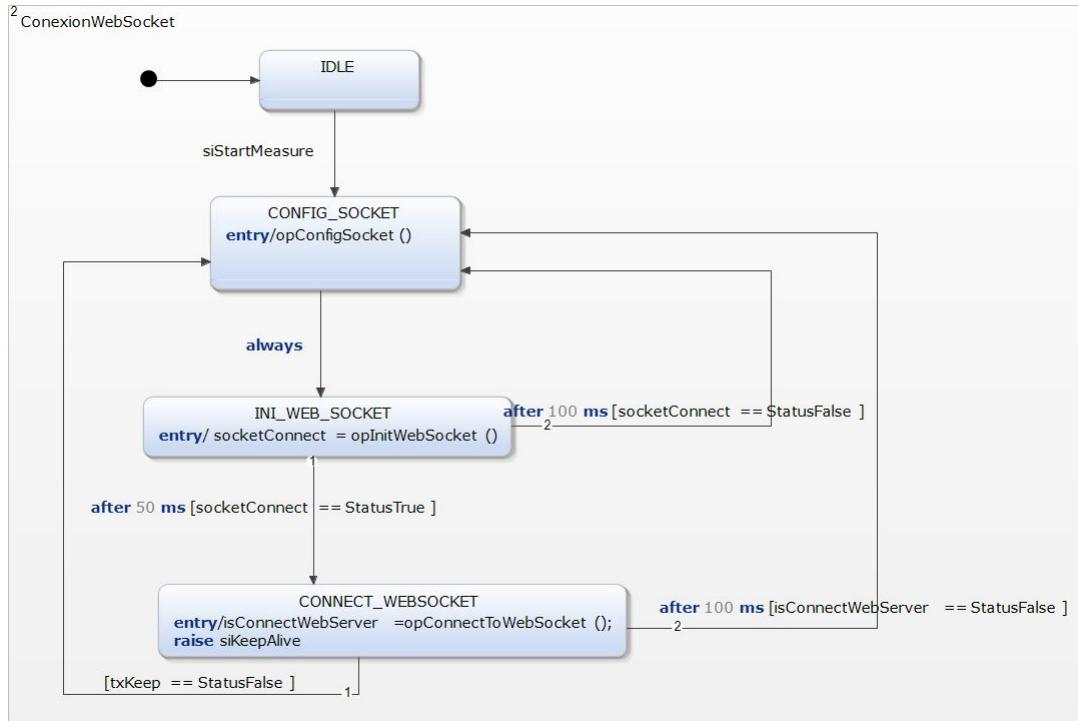


Figura 4.46: Máquina de estados para establecer una conexión bidireccional con un servidor WebSocket.

Para prevenir la desconexión automática del protocolo WebSocket en ausencia de intercambio de datos, se implementó una región denominada `KeepAlive_lecturaBuffer`. Esta región tiene un doble propósito: primero, mantener activa la comunicación con el servidor incluso cuando no hay tráfico de datos; y segundo, realizar una consulta periódica (*polling*) con un retardo no bloqueante para verificar si se ha recibido algún mensaje del servidor.

El estado final de la región `ConexionWebSocket` de la Figura 4.46, al completar su secuencia de operaciones, genera un evento con la señal `siKeepAlive`. Esta señal, en conjunción con la condición de que `isConnectWebServer` esté establecida en `True`, facilita la transición desde el estado `IDLE` hacia el estado `TX_MESSAGE`. En este último estado, se pueden transmitir mensajes al servidor, manteniendo así la conexión WebSocket activa y funcional.

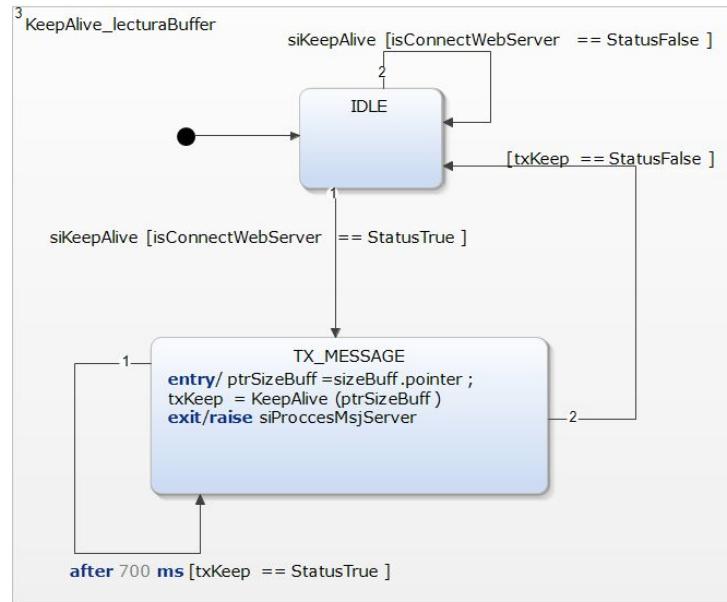


Figura 4.47: Máquina de estados para mantener la conexión con el servidor y elevar un raise en caso de que se reciba información del servidor.

Puesto que se puede recibir datos del servidor, se declaran las siguientes variables del Código

4.5

```

1 // Procesamiento de comandos
2 // Size de comandos
3 var sizeBuff: uint16_t = 0
4 var ptrSizeBuff: ptrUnt16_t
5 // Estructura de configuracion a traves de comandos
6 var configCommand: config_t
7 var ptrConfigCommand: ptrConfig_t = configCommand.pointer

```

Código 4.5: Declaración de variables para validar y procesar la recepción de datos del servidor.

En el estado **TX_MESSAGE**, independientemente de si se han recibido datos del servidor, se genera un evento hacia la región **ProcesoDeComandos**. Este evento facilita la transición del estado **IDLE** al estado **PROCESS_COMMAND**, condicionada a que la longitud del mensaje recibido del servidor sea mayor a cero. Si se cumple esta condición, en el estado **PROCESS_COMMAND** se invoca la función `opProcessMessageFromServer(ptrSizeBuff, ptrConfigCommand)`, que se encarga de procesar los mensajes provenientes del servidor. Estos mensajes pueden incluir comandos de configuración para el datalogger o valores de referencia para el control del túnel, los cuales serán detallados en la sección 4.3.4.

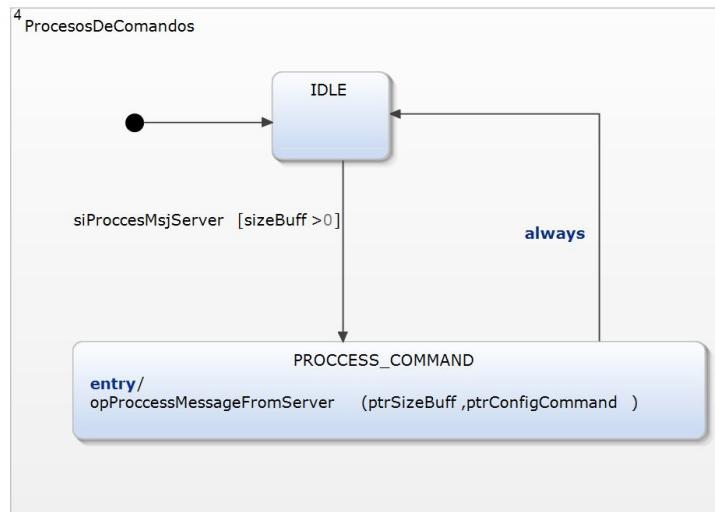


Figura 4.48: Máquina de estados para procesar los mensajes recibidos del servidor WebSocket.

Para gestionar eficientemente los comandos destinados a configurar diversas partes del datalogger con la información proporcionada por el software, se establece un diccionario de punteros a función. Este diccionario, ilustrado en el código 4.6, permite ejecutar una función específica en respuesta a cada comando o petición contenida en el mensaje recibido.

```

1 char *diccCommands []={
2     "start",
3     "stop",
4     "setIBC",
5     "setPAT",
6     "setTimes",
7     "refWindVel",
8     "NAN"};
9 bool_t (*diccProcesos [])(config_t * commandConfig)={
10     start,
11     stop,
12     setIBC ,
13     setPAT ,
14     setTimes ,
15     refWindVel};
```

Código 4.6: Funciones que permiten configurar o modificar el comportamiento del datalogger.

4.3.2. Muestreo, adquisición y procesamiento

Para almacenar las muestras de los sensores de viento, patrón y bajo calibración, así como los niveles de tensión de la fuente y la tensión que ingresa al variador de velocidad, se han declarado las variables, punteros y constantes mostrados en el Código 4.7. Cada variable está asociada a un puntero, ya que cada función de cada estado o transición requiere recibir un puntero específico a una variable para poder modificar su valor.

```

1 //Variable interna para guardar las muestras de voltaje DC
```

```

2 var viVoltajeMuestra: real32_t
3 //Puntero Correspondiente
4 var ptrVoltajeMuestra: ptrReal32_t
5 //MuestraDelAdcDelTunnel
6 //Variable interna para guardar las muestras de voltaje DC
7 var viTunelTensionMuestra: real32_t
8 //Puntero Correspondiente
9 var ptrTunelTensionMuestra: ptrReal32_t
10 // IBC
11 var ptrAnemoIBC:ptrAmenometerSerialParam_t = configCommand.AnemoIBC.pointer
12 // PAT
13 var ptrAnemoPat:ptrAmenometerSerialParam_t=configCommand.AnemoPAT.pointer

```

Código 4.7: Variables para almacenar en cada iteración una muestra de su magnitud correspondiente.

Para configurar cada anemómetro, se ha declarado una estructura como se muestra en el fragmento de código 4.8. Esta estructura se inicializa con datos ingresados por el operador a través de la aplicación web descrita en el capítulo 5. Cada estructura requiere configurar el número de **Uart**, el LED de recepción y el **BaudRate**, valores que deben coincidir con las especificaciones de cada anemómetro. Además, se dispone de un buffer de 100 caracteres para almacenar los strings de datos recibidos, con un puntero asociado para modificar su contenido. También se incluyen dos arreglos de números reales para almacenar los datos numéricos convertidos a **real32_t** a partir de los strings de cada sensor, cada uno con su respectivo puntero para su modificación.

Por último, se introduce el tipo de dato del sensor, **sensor_t**, que enumera los sensores utilizados.

```

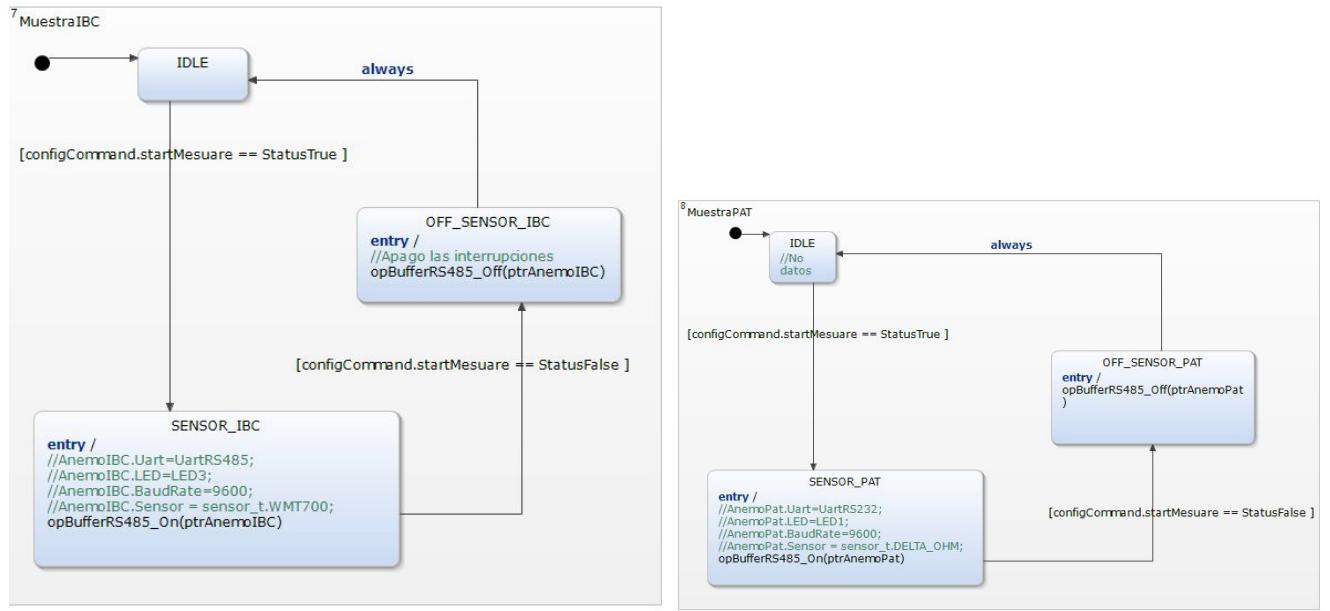
1 typedef struct{
2     uint16_t Uart;
3     uint16_t LED;
4     uint32_t BaudRate;
5     char Buffer[100];
6     char * ptrUartBuffer;
7     real32_t DataAnemometer[7];
8     real32_t * ptrDataAnemometer;
9     sensor_t Sensor;
10 }amenometerSerialParam_t;

```

Código 4.8: Estructura de datos para anemómetro.

En las Figuras 4.49a y 4.49b se muestran las regiones que contienen las máquinas de estado **MuestraIBC** y **MuestraPAT**, respectivamente, para adquirir los datos de viento. La máquina de estado, por ejemplo, del IBC comienza en el estado **IDLE**. Cuando el comando **startMeasure** es **True**, se transiciona al estado **SENSOR_IBC**, donde se invoca la función **opBufferRS485_On(prtAnemo)**. Esta función recibe un puntero con todos los datos de configuración del anemómetro y los carga en su respectiva UART. Además, inicializa un callback y habilita la interrupción correspondiente de la UART. De esta manera, cada vez que los sensores envían datos, se activa una función que se encarga de procesar y separar los datos recibidos.

Cuando `startMeasure` se configura como `false`, la máquina de estado transiciona al estado **OFF_SENSOR_IBC**, donde se desactiva la interrupción y el microcontrolador deja de tomar muestras de los anemómetros.



(a) Máquina de estados para tomar muestras de sensor bajo calibración.

(b) Máquina de estados para tomar muestras de sensor patrón.

Figura 4.49

El string del patrón VAISALA, modelo WMT700 tiene el formato `$-MWV,021,R,003.34,N,A*14`, mientras que el del instrumento bajo calibración DELTAOHM, modelo HD51.3 tiene el formato `28.30 359.3 998.3<CR><LF>`. Para procesar y separar estos datos, se utiliza un diccionario de punteros a funciones, como se muestra en el fragmento de Código 4.9. Desde el software, el operador preconfigura el modelo del anemómetro y, por lo tanto, a la máquina de estados, asegurándose de llamar a la función correspondiente para procesar los datos de cada anemómetro.

```

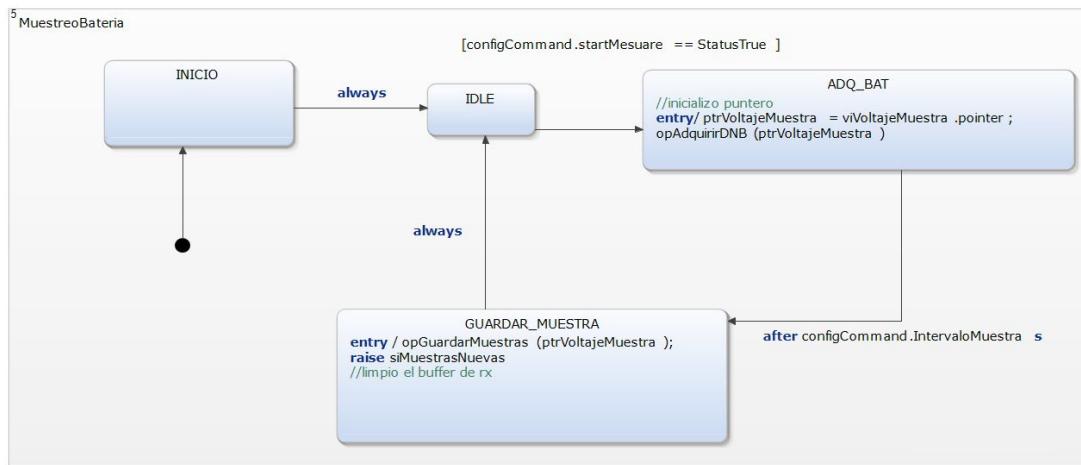
1 // Diccionario de punteros a función para procesar en función del sensor elegido
2 void (*opProcesoDatosViento [])(amenometerSerialParam_t * data) = {
3     opProcesoDeltaOHM,
4     opProcesoWMT700,
5     opProcesoVentus
6 };

```

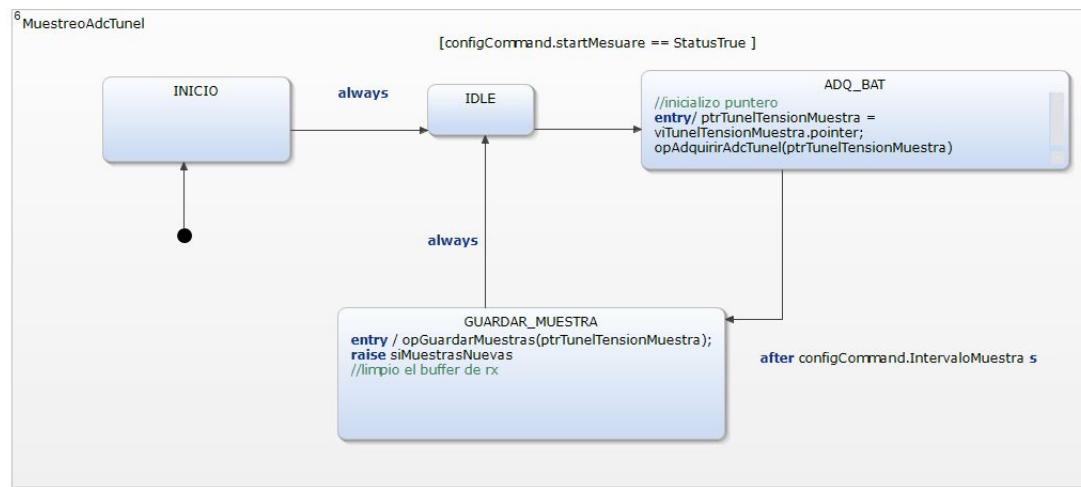
Código 4.9: Diccionario de punteros a función para procesar los datos recibidos de los anemómetros.

En las Figuras 4.50a y 4.50b se muestran las regiones que contienen las máquinas de estado **MuestreoBateria** y **MuestreoAdcTunel**, respectivamente, para adquirir los niveles de tensión de la fuente de alimentación y los niveles de tensión del variador de velocidad en las variables

declaradas en el Código 4.7. Por ejemplo, para el muestreo del ADC del variador, la máquina de estado comienza en el estado **INICIO** y automáticamente transita al estado **IDLE**. Luego, siempre que la condición `startMeasure` sea igual a `True`, se produce una transición al estado **ADQ_BAT**, donde se invoca la función `opAdquirirDNB(ptrVoltajeMuestra)` para tomar una muestra del nivel de tensión. Después de transcurrido el tiempo definido en `IntervaloMuestra` en segundos, se procede a guardar las muestras y se eleva (se hace un *raise*) a través del evento (señal) `siMuestrasNuevas` que permite activar una máquina de estado de otra región. Finalmente, se regresa al estado **IDLE** para repetir el ciclo indefinidamente o hasta que la condición `startMeasure` sea `False`.



(a) Máquina de estado para tomar muestras de la fuente de alimentación del datalogger.



(b) Máquina de estado para tomar muestras de la tensión que inyecta al variador de velocidad del túnel de viento.

Figura 4.50

En el Código 4.10 se muestra las variables declaradas para contabilizar el número de muestras y la cantidad de datos acumulados cada vez que se realiza una iteración para tomar muestras de

los sensores de viento y los niveles de tensión detallados arriba.

```

1 // Adquisicion y procesamiento
2 var NumMuestra: uint16_t =0
3 var ptrNumMuestra: ptrUnt16_t
4 var cantDatosProcesado: int32_t =0
5 var ptrCantDatosProcesado:ptrInt32_t

```

Código 4.10: Variables para controlar la adquisición y procesamiento de los datos .

La región **Procesamiento** de la figura 4.51 inicia en un estado de **IDLE**. Se produce la transición al estado **ACUMULO_MUESTRAS** cuando la señal elevada **siMuestrasNuevas**, generada en la región **MuestreoBateria**¹, está activada y se cumplen dos condiciones: primero, que el número de muestras (**NumMuestra**) sea menor al cociente entre el intervalo de Tabla y el intervalo de muestra; segundo, que la condición **startMeasure** sea **True**. En este nuevo estado, se llama a la función **opAcumular(ptrNumMuestra, ptrVoltajeMuestra, ptrTunelTensionMuestra, ptrAnemoIBC, ptrAnemoPat)**, que se encarga de guardar las muestras en un arreglo, se incrementa el número de muestras en una unidad y luego se regresa al estado **IDLE**.

Por ejemplo, si el operador definió un intervalo de muestras de 1 segundo y un intervalo de tabla de 10 segundos, el cociente es 10. Cuando el número de muestras (**NumMuestra**) llega a 10, se produce la transición al estado **PROCESO_y_ALMACENO**. En este estado, se llama a la función **opProceso(ptrNumMuestra, ptrCantDatosProcesado, ptrPwmValue)**, se calcula el promedio, mínimo y máximo de las 10 muestras tomadas cada 1 segundo. Luego, se reinicia el número de muestras a cero. Finalmente, se eleva la señal **siDatoProcesado** para activar la región de transmisión de datos y se vuelve al estado **IDLE** para acumular las próximas 10 mediciones. Todos estos datos, incluyendo el último dato considerado como el dato instantáneo, son guardados en el microcontrolador para posteriormente ser transmitidos al servidor.

¹Descripción anteriormente como un ejemplo de relación entre regiones mediante señales elevadas

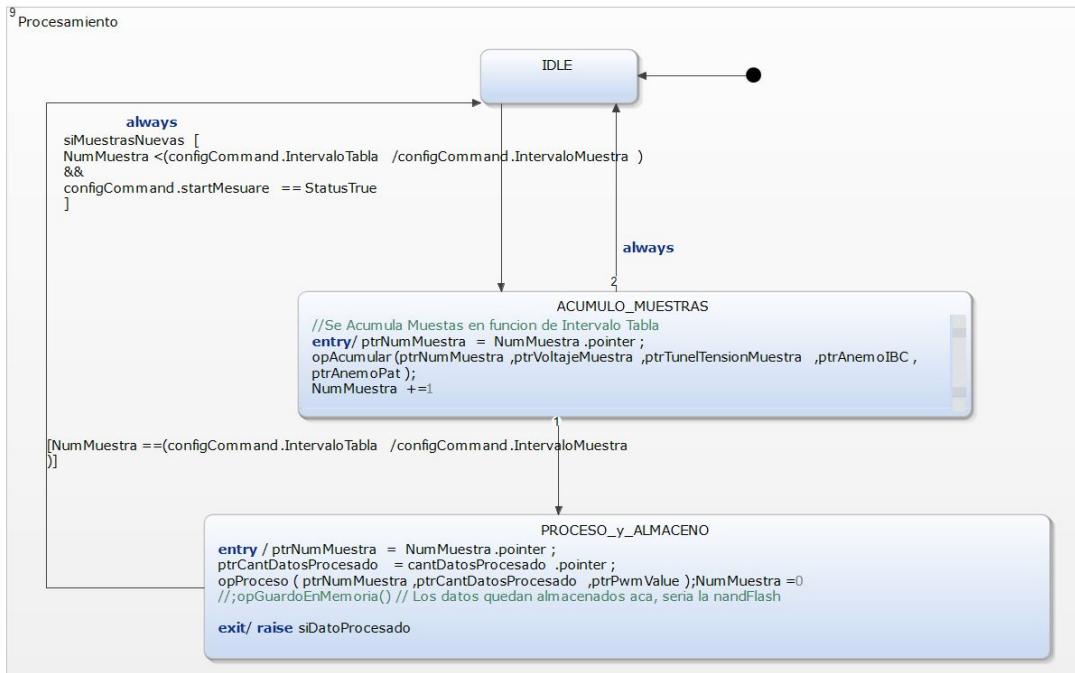


Figura 4.51: Máquina de estados para acumular muestras en una tabla y realizar un preprocesamiento de esos datos.

4.3.3. Transmisión al servidor

El protocolo WebSocket establece una comunicación en tiempo real entre clientes y servidores, adaptando el encabezado del mensaje según su tamaño. Para mensajes hasta 125 bytes, se usa un encabezado de 2 bytes. Para mensajes mayores, se añade un encabezado extendido con 2 bytes adicionales para la longitud. Independientemente del tamaño, todos los mensajes se enmascaran con una clave de 32 bits aplicada mediante XOR a la carga útil, protegiendo la transmisión contra vulnerabilidades y garantizando la integridad de los datos. Este mecanismo asegura que todos los mensajes, independientemente de su tamaño, se transmitan de manera confiable y conforme a las normas del protocolo. Para una transmisión exitosa desde el datalogger hacia el servidor, se declara la variable de Código 4.11, esta variable controla el estado de la transmisión en la máquina de estados de la región **TransmisionAlServer** de la figura 4.52.

```
1 // Transmision por Ethernet
2 var TransmisionExitosa: uint8_t=FALSE
```

Código 4.11: Declaración de variable para controlar el estado de una transmisión al servidor.

La máquina de estados de la figura 4.52 inicia en el estado **IDLE**. Cuando se detecta la señal `siDatosListosParaBdtos` y la condición `isConnectWebServer` está en `True`, se realiza una transición al estado **TX_MESSAGE**. En este estado, se invoca la función `opTransmitWebSocketEthernet(ptrCantDatosProcesado)`, la cual tiene como responsabilidad

codificar y enviar el mensaje a través del socket conectado al servidor web.

La codificación del mensaje depende de su longitud, para ello se implementaron dos funciones internas:

- Si la longitud es de hasta 125 bytes, se utiliza la función `encodeMessage125()`. El encabezado del mensaje codificado consta de 2 bytes.
- Si la longitud supera los 125 bytes, se emplea la función `encodeMessage126()`, que añade un campo adicional de 2 bytes para indicar la longitud extendida del mensaje.

Tras la codificación, se procede con el envío del mensaje. El servidor, al recibir el mensaje, responde con la longitud del mensaje recibido. Si la longitud coincide con la del mensaje enviado, se considera que la transmisión ha sido exitosa. De lo contrario, se asume que la transmisión ha fallado.

En caso de fallo en la transmisión, la máquina de estados transiciona al estado **BACK_UP**. Este estado se encarga de almacenar los datos hasta que se restablezca la comunicación con el servidor. Si la transmisión es exitosa, se incrementa un contador y se retorna al estado **IDLE**, quedando listo para una nueva transmisión.

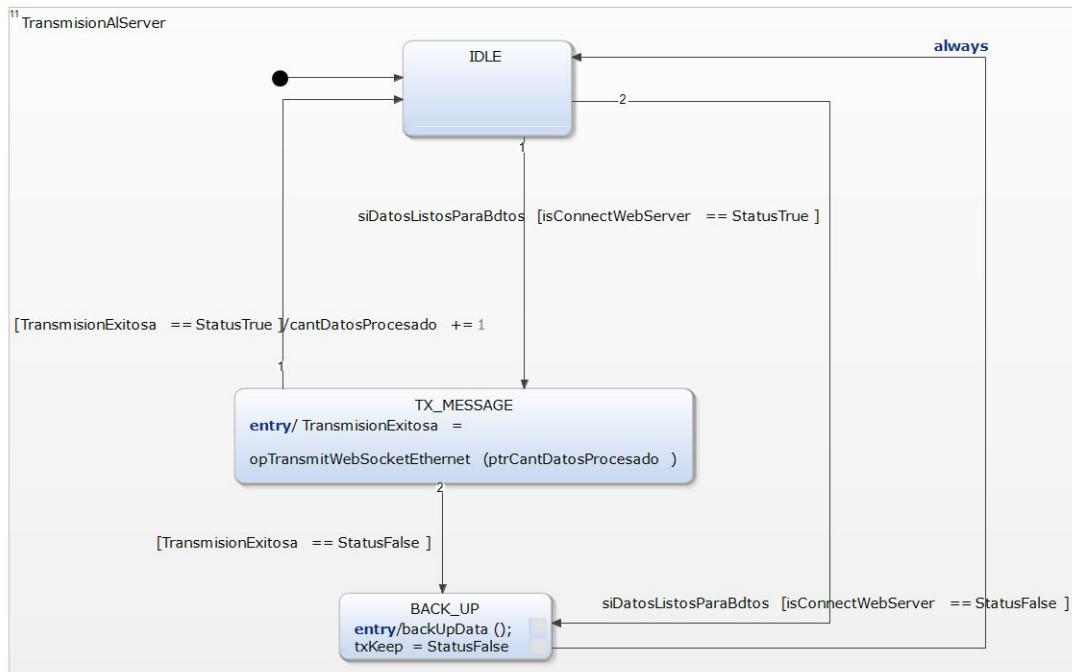


Figura 4.52: Máquina de estados para gestionar la transmisión de datos al servidor WebSocket.

4.3.4. Sistema de control PID

En la Sección 4.1.4, se detalla el desarrollo del circuito **PWM**. Este circuito, al recibir un valor digital entre 0 y 255, produce un voltaje de salida que oscila entre 53 mV y 3,5 V. Este

voltaje se utiliza para controlar el variador de velocidad del motor que trae el túnel de viento, permitiendo variar la velocidad del aire entre 1 m s^{-1} y 25 m s^{-1} .

El circuito PWM se utiliza para la implementación de un controlador **PID** de lazo cerrado. La referencia de la velocidad del viento, en metros por segundo, se obtiene del servidor. Paralelamente, se toma una muestra de la velocidad del viento a través de un sensor patrón, también en metros por segundo. La diferencia entre la referencia y la muestra se define como el error, sobre el cual se aplica el control PID.

Las variables necesarias para el control PID se declaran como se muestra en el Código 4.12. Las variables `viIntesidadPatron` y `viIntesidadReferencia` almacenan las muestras del viento patrón y la referencia respectivamente. La variable `viPwmValue` guarda el valor de PWM calculado por el controlador, y en `viPid` se almacenan los parámetros del controlador PID.

```

1 // Control del tunel
2 // Valor del patron de viento
3 var viIntesidadPatron: real32_t = 0
4 var ptriIntesidadPatron: ptrReal32_t
5 // Valor de referencia de viento
6 var viIntesidadReferencia: real32_t = 0
7 var ptriIntesidadReferencia: ptrReal32_t
8 // Valor de PWM
9 var viPwmValue: int32_t = 0
10 var ptrPwmValue: ptrInt32_t = &viPwmValue
11 // Controlador PID
12 var viPid: pidController_t
13 var prtPid: ptrPidController_t = &viPid

```

Código 4.12: Declaración de variables para el controlador PID del motor del túnel de viento.

En el Código 4.13, se presenta la declaración de una estructura que define los parámetros del controlador PID. Estos parámetros incluyen las ganancias proporcional (`kp`), integral (`ki`) y derivativa (`kd`). Además, se registran variables para el manejo del error actual (`error`), el error anterior (`lastError`), la suma acumulada del error (`integral`) y la derivada del error (`derivative`) y se incluye el tiempo de muestreo (`Ts`), puesto que es un controlador en tiempo discreto.

```

1 // Estructura del controlador PID
2 typedef struct {
3     real32_t kp;          // Ganancia proporcional
4     real32_t ki;          // Ganancia integral
5     real32_t kd;          // Ganancia derivativa
6     real32_t error;       // Error actual
7     real32_t lastError;   // Error anterior
8     real32_t integral;    // Integral del error
9     real32_t derivative; // Derivada del error
10    real32_t Ts;          // Tiempo de muestreo
11 } pidController_t;

```

Código 4.13: Estructura para los parámetros y variables del controlador PID.

Finalmente, se aplica el algoritmo de control **PID** sobre el error calculado para regular la salida del circuito PWM y, consecuentemente, la velocidad del viento en el túnel. La máquina de estados comienza en el estado **INICIO**, donde se ejecuta la función `configPID(prtPid)`. Esta función configura los parámetros del controlador PID, obtenidos heurísticamente, con $kp = 3$; $kd = 0,5$; $ki = 0$; y el tiempo de muestreo $Ts = 1\text{s}$. Además, prepara el pin digital de la EDU-CIAA para habilitar la salida PWM con un valor inicial de cero.

Si el indicador `starMesuare` se encuentra en `True`, se transiciona al estado **MESUARE_PAT**. En este estado, se invoca la función `opMesuareWindPAT(ptrAnemoPat, ptrIntesidadPatron, ptrIntesidadReferencia)`, que obtiene la intensidad instantánea del viento del anemómetro patrón y la intensidad de referencia enviada por el servidor, almacenando estos valores en variables correspondientes.

Posteriormente, se avanza al estado **CONTROL_TUNNEL**. Aquí, se llama a la función `pidControlTunnel(prtPid, ptrIntesidadPatron, ptrIntesidadReferencia, ptrPwmValue)`, que ejecuta las siguientes acciones:

- Calcula el error proporcional $e_i = vel_{ref_i} - vel_{pat_i}$.
- Determina el término integral $e_{int_i} = e_i \cdot Ts$.
- Estima el término derivativo $e_{dev_i} = \frac{e_i - e_{i-1}}{Ts}$.
- Actualiza el error y calcula la acción de control como $u_c = kp \cdot e_i + ki \cdot e_{int_i} + kd \cdot e_{dev_i}$.

El valor de u_c se somete a un saturador entre 0 y 255, y luego se aplica al pin de salida PWM. Este ciclo se repite cada 1s , correspondiente al tiempo de muestreo Ts .

Si el operador detiene las mediciones a través del software, se envía un comando que establece el indicador `starMesuare` en `False`. Esto hace que el controlador se ponga en cero y permanezca en espera para iniciar nuevas mediciones.

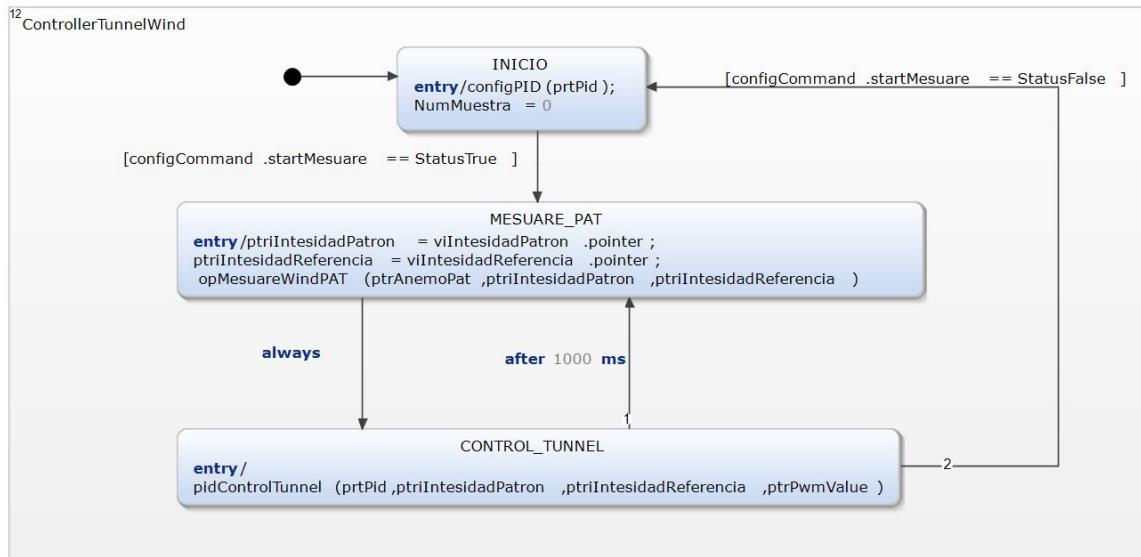


Figura 4.53: Máquina de estados que ejecuta el control PID con valores de referencia enviados por el servidor.

Capítulo 5

Implementación de la Aplicación WEB

En este capítulo, se introduce una aplicación web diseñada para interactuar con el datalogger descrito en el capítulo anterior, con el objetivo de optimizar la calibración de anemómetros. Esta herramienta de software simplifica la carga de metadatos esenciales, incluyendo la información de los sensores de referencia y los sensores bajo calibración, los certificados de calibración y las especificaciones de la zona de medición del túnel de viento.

La aplicación ofrece la posibilidad de configurar el datalogger, definiendo la interfaz eléctrica de los anemómetros, los intervalos de muestreo y procesamiento de datos, y los puntos de medición de la velocidad del viento. Además, permite establecer los tiempos de encendido y el periodo de estabilidad en cada punto de medición. Una vez configurado el sistema, el software inicia el proceso de medición, enviando referencias de viento al hardware. Este último interactúa con el túnel de viento a través de su controlador PID para ajustar la velocidad del viento al valor deseado. Las mediciones obtenidas son verificadas por el operador y, si son correctas, se calcula automáticamente la incertidumbre expandida. Los resultados se presentan en el front end, en gráficos y tablas, se almacenan en una base de datos y pueden descargarse para emitir el certificado de calibración correspondiente.

5.1. Desarrollo aplicación WEB

Durante el desarrollo de la aplicación web, se emplearon diversas herramientas para llevar a cabo el proyecto de manera eficiente. Se utilizó Django en su versión 4.2 como framework principal, junto con Python 3.10, proporcionando una base sólida y flexible para el desarrollo back end. Para la gestión de la base de datos, se optó por PostgreSQL en su versión 15 y se utilizó PGAdmin 4 para gestionar dicha base de datos. La conexión entre Django y PostgreSQL se realizó utilizando PSYCOPG2 en su versión 2.9.5.

En el frontend, se emplearon JavaScript y jQuery para hacer la página web dinámica y permitir la visualización de datos actualizados en tiempo real. También se utilizaron HTML y CSS para estructurar y estilizar el contenido de la aplicación web. Además, se emplearon herramientas como NumPy 1.24.2, SciPy 1.11.2 para el procesamiento de datos y Plotly 5.15 para generar gráficos dinámicos, así como Pandas 2.2 para la manipulación de datos.

Adicionalmente, se trabajó en un servidor local, dentro de un entorno virtual de Python para mantener las dependencias aisladas y facilitar la portabilidad del proyecto a otros entornos. Se gestionó el control de versiones con Git y GitHub, en el repositorio https://github.com/InstrumentalSMN/AppWebSistemaDecalibracion/tree/develop_apps, subiendo el código y trabajando en distintas ramas para luego integrar los cambios en la rama principal, lo que permitió mantener un historial claro de los cambios realizados.

5.1.1. Arquitectura del software

Para desarrollar la aplicación, se utilizó una arquitectura de cinco capas, como se muestra en la Figura 5.1. La primera capa está conformada por el servidor web Nginx, que actúa como un servidor *proxy* inverso. Nginx se encarga de manejar las solicitudes HTTP entrantes, distribuyéndolas eficientemente y proporcionando un equilibrio de carga, además de servir contenido estático como archivos CSS, JavaScript e imágenes. Esta capa permite mejorar el rendimiento y la seguridad de la aplicación al filtrar y dirigir el tráfico de manera óptima.

La segunda capa está compuesta por Gunicorn, un servidor WSGI (*Web Server Gateway Interface*) que sirve como intermediario entre Nginx y la aplicación web desarrollada en Django. Gunicorn se encarga de gestionar los procesos de la aplicación, proporcionando un entorno escalable y capaz de manejar múltiples solicitudes concurrentes. Esta capa es crucial para asegurar que la aplicación pueda responder de manera rápida y eficiente a las demandas de los usuarios.

En la tercera capa se encuentra la aplicación web desarrollada en Django, un *framework* de alto nivel que facilita el desarrollo rápido y eficiente de aplicaciones web seguras y mantenibles. Esta capa es responsable de la lógica de negocio de la aplicación, la gestión de las bases de datos y la interacción con los usuarios a través de interfaces web dinámicas.

La cuarta capa está constituida por el sistema de gestión de bases de datos PostgreSQL, encargado del almacenamiento y recuperación de datos de manera eficiente y segura. La integración con Django se realiza a través de su ORM (*Object-Relational Mapping*), que permite interactuar con la base de datos utilizando objetos en Python. Las tablas se definen creando una clase que hereda de `models.Model` de Django, donde los campos se establecen como atributos de la clase. Esto permite crear registros, guardar información y realizar consultas de manera sencilla, tra-

duciendo automáticamente las operaciones en Python a consultas SQL en la base de datos. Esta abstracción de los detalles subyacentes de la implementación facilita la gestión y manipulación de datos, mejorando la eficiencia en el desarrollo de aplicaciones web.

Finalmente, la quinta capa consiste en un servidor WebSocket que se conecta con un sistema embebido. Este servidor permite la comunicación bidireccional en tiempo real entre la aplicación y dispositivos embebidos, facilitando el control y monitoreo remoto. Esta capa es esencial para aplicaciones que requieren actualizaciones en tiempo real y una interacción continua con dispositivos de hardware.

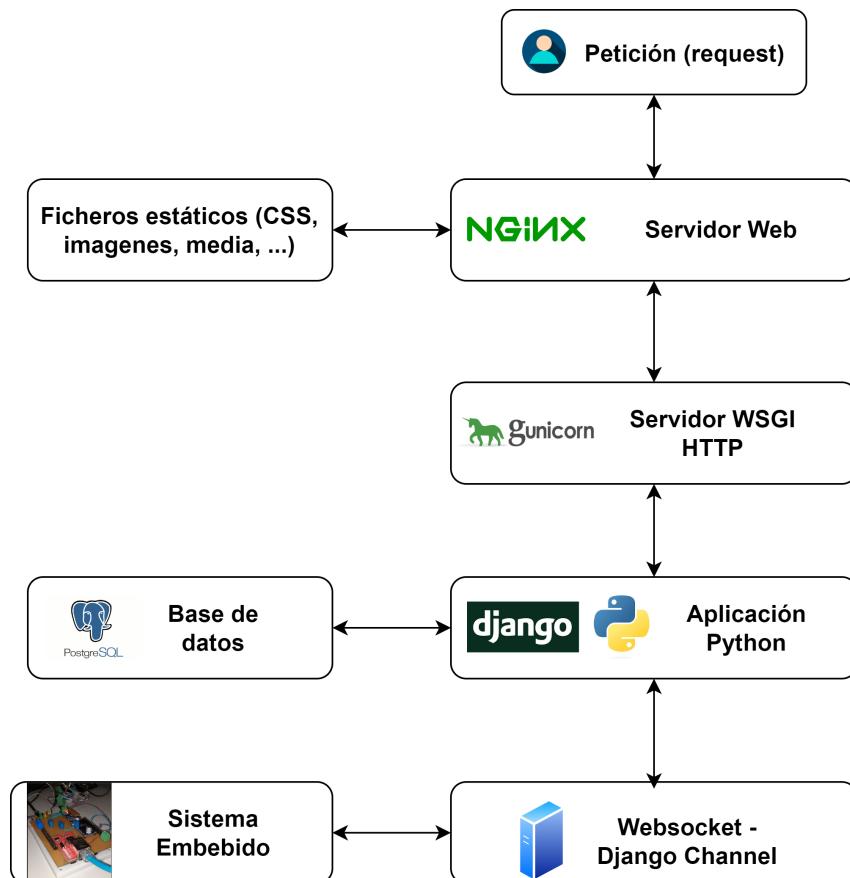


Figura 5.1: Arquitectura del software implementada.

Se empleó el patrón de diseño MVT (Modelo-Vista-Plantilla), ilustrado en la Figura 5.2. Este patrón es ampliamente utilizado en el desarrollo web con el *framework* de trabajo Django. El patrón MVT permite una distinción explícita de las responsabilidades, lo que facilita tanto el desarrollo como el mantenimiento del código.

- **Modelo (M):** Representa la capa de acceso a la base de datos (ORM). Contiene toda la información sobre los datos, cómo acceder a ellos, validarlos, su comportamiento y las relaciones entre ellos.

- **Vista (V)**: Corresponde a la capa de lógica de negocios. Se encarga de procesar las solicitudes del navegador y recuperar los datos necesarios del modelo. Luego, renderiza la plantilla (template) para mostrar el HTML resultante.
- **Template (T)**: (Plantilla), representa la capa de presentación, es la parte visual de la aplicación. El template integra los datos dinámicos recuperados del modelo y genera el HTML final que se envía al navegado

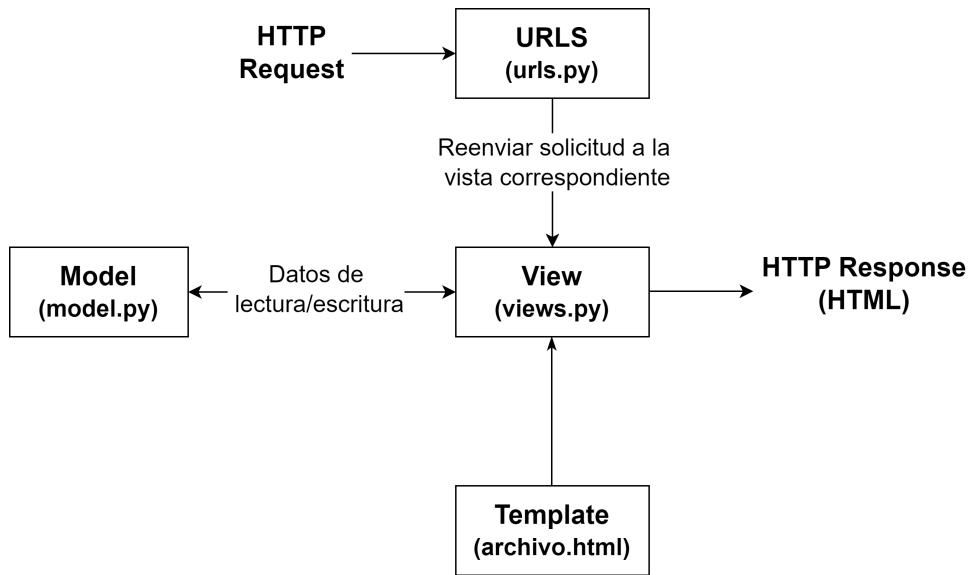


Figura 5.2: Patrón de diseño de software MVT, que divide la lógica del programa en tres elementos interconectados.

5.1.2. Back End

El back end se encarga de:

- Administrar la base de datos, incluyendo la ejecución de consultas, la realización de actualizaciones y la aplicación de validaciones para garantizar la integridad y consistencia de la información.
- Recibir o gestionar solicitudes HTTP y conexiones al servidor WebSocket.
- Enviar comandos de configuración hacia el datalogger y gestionar la recepción de mediciones de los sensor.
- Generar un perfil de referencias de velocidad de viento almacenadas en la base de datos, que se van a transmitir al datalogger para controlar el motor del túnel.
- A partir de las mediciones crudas de velocidades de viento, calcular el presupuesto de incertidumbre de la corrección.

5.1.2.1. Base de datos

Dado que la aplicación está diseñada para manejar una gran cantidad de datos que requieren almacenamiento y recuperación eficientes, se ha implementado una base de datos relacional. El diagrama entidad-relación, presentado en la figura 5.3, ilustra la estructura de la base de datos. Esta estructura incluye diversas tablas que almacenan información crucial para la aplicación. Entre estas tablas se encuentran las que registran los datos del sensor patrón y el sensor bajo calibración, así como los detalles relacionados con el certificado de calibración del patrón. Además, se documentan los registros del certificado de caracterización del túnel de viento. También se incluye información sobre la configuración del datalogger y del túnel de viento.

Cuando se inicia una calibración y se empiezan a recibir datos de los sensores, toda esta información se almacena en una tabla **DatosMedidos2023** de la base de datos. Asimismo, se guardan los resultados generados antes y después del procesamiento de los datos, como gráficos y tablas.

Todas estas tablas están relacionadas mediante claves foráneas (relación de uno a muchos) con la tabla de **Calibración**. Esto se debe a que una calibración está asociada a un único datalogger o a un único sensor IBC, y se puede recuperar la información de esa calibración a través clave primaria.

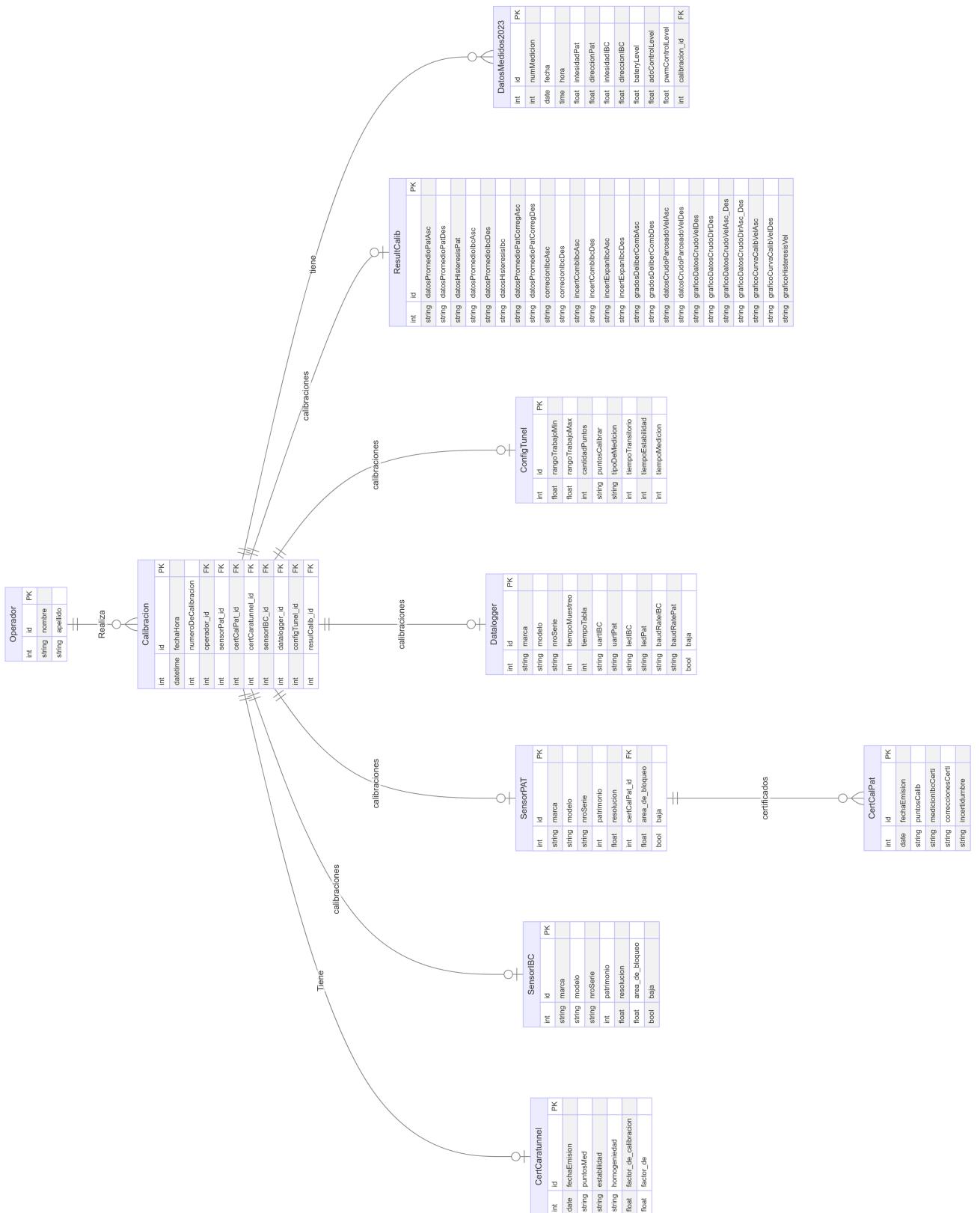


Figura 5.3: Diagrama entidad relación diseñada para la aplicación Web.

5.1.2.2. Servidor WebSocket

Se utilizó la biblioteca Django-channels para implementar un servidor WebSocket que opera de forma concurrente con el servidor web HTTP. Este servidor permite la conexión simultánea de múltiples clientes, en este caso, un datalogger que envía datos y un usuario que recibe dicha información a través de WebSockets. Los datos recibidos por el servidor, son graficados y actualizados en tiempo real en la aplicación web.

Para configurar el servidor, se creó el archivo `routing.py`, el cual define las rutas de los canales, es decir, los puntos de acceso a los consumidores (consumers) que gestionan las conexiones WebSocket. La configuración del archivo `routing.py` se presenta en el Código 5.1. Cada ruta mapea una URL de WebSocket a un consumidor específico. En este caso, `ChatConsumer` es una clase que maneja las conexiones a la URL `ws/socket-server/`.

```

1 from django.urls import re_path
2 from . import consumers
3
4 websocket_urlpatterns = [
5     re_path(r'ws/socket-server/', consumers.ChatConsumer.as_asgi())
6 ]

```

Código 5.1: Configuración del routing para gestionar las conexiones de clientes al servidor WebSocket.

También se creó el archivo `consumers.py` que define las clases de consumidores que manejan las conexiones WebSocket. Un consumidor en Django-channels es una clase que gestiona la conexión con el cliente, incluyendo el manejo de eventos como la recepción y el envío de mensajes. En el código 5.2 se muestra el consumidor implementado

```

1 class ChatConsumer(WebSocketConsumer):
2     def connect(self):
3         self.room_group_name = 'test'
4         sync_to_sync(self.channel_layer.group_add)(
5             self.room_group_name,
6             self.channel_name
7         )
8         self.accept()
9
10    def disconnect(self, close_code):
11        pass
12
13    def receive(self, text_data):
14        text_data_json = json.loads(text_data)
15        message = text_data_json['message']
16        # message = text_data
17        sync_to_sync(self.channel_layer.group_send)(
18            self.room_group_name,
19            {
20                'type': 'chat_message',
21                'message': message,

```

```

22         'sender_channel_name': self.channel_name,
23     }
24   )
25   if(len(message)>0):
26     if(message[0] == "M"):
27       saveMesuareDataBase(message)
28     print('Message:', message)
29
30   def chat_message(self, event):
31     message = event['message']
32     sender_channel_name = event['sender_channel_name']
33     if sender_channel_name != self.channel_name:
34       self.send(text_data=json.dumps({
35         'type': 'chat',
36         'message': message
37       }))

```

Código 5.2: Declaracion de las funciones que utiliza cada consumidor para interactuar con el servidor WebSocket.

En particular, se implementa la clase `ChatConsumer`, que hereda de `WebsocketConsumer`, para gestionar las conexiones WebSocket. Esta clase define varios métodos, incluyendo `connect`, `disconnect`, `receive` y `chat_message`.

El método `connect` se ejecuta cuando un cliente se conecta al servidor WebSocket. Dentro de este método, se asigna el nombre del grupo de la sala a `self.room_group_name` y se añade el canal del cliente al grupo utilizando `async_to_sync(self.channel_layer.group_add)`. Finalmente, se acepta la conexión mediante `self.accept()`. Por otro lado, el método `disconnect` se ejecuta cuando un cliente se desconecta del servidor WebSocket.

El método `receive` recibe y decodifica los mensajes desde JSON para luego enviarlos a un grupo de la sala utilizando `async_to_sync(self.channel_layer.group_send)`. Los mensajes que envía el datalogger tiene la forma del arreglo declarado en código 5.3, que contiene toda la información que ha recolectado el datalogger en un ciclo de iteración.

```

1 real32_t Tabla_Mediciones[25] = {
2   velocidadInst, velocidadMin, velocidadMax, velocidadPromedio,
3   velocidadInstIBC, velocidadMinIBC, velocidadMaxIBC, velocidadPromedioIBC,
4   direccionInst, direccionMin, direccionMax, direccionPromedio,
5   direccionInstIBC, direccionMinIBC, direccionMaxIBC, direccionPromedioIBC,
6   nv_bateriaInst, nv_bateriaMin, nv_bateriaMax, nv_bateriaPromedio,
7   nv_adcTunelInst, nv_adcTunelMin, nv_adcTunelMax, nv_adcTunelPromedio,
8   nv_pwmControlTunelInst
9 };

```

Código 5.3: Estructura del mensaje enviado por el datalogger

El mensaje decodificado tiene la siguiente forma:

```

poner numeros de la base de datos
M;634;14-06-2024;16:42:41;

```

```

10.08;10.08;10.08;10.08;
9.53;9.53;9.53;9.53;
169;169;169;169;
175;175;175;175;
12.02;12.02;12.02;12.02;
1.25;1.25;1.25;1.25;
90;

```

De esta forma, si el mensaje recibido comienza con la letra "M", se guarda en la base de datos mediante la función `saveMeasureDataBase(message)`. Esta función conoce el orden en que se creó el string y separa los datos para guardarlos en su respectivo campo de la base de datos.

Finalmente, el método `chat_message` se utiliza para enviar mensajes a los clientes. Si el mensaje recibido no se origina del mismo canal que el receptor, se envía el mensaje de vuelta al cliente utilizando `self.send()`. Por ejemplo, cuando un cliente está conectado al WebSocket desde el navegador web, puede ver cómo se grafican los datos en tiempo real a medida que estos llegan desde el datalogger hacia servidor.

5.1.2.3. Generador de trayectoria y control del túnel

En la sección 4.3.4 se ha detallado el funcionamiento del controlador PID del túnel de viento. Para que este funcione, necesita referencias que se construyen a partir de un generador de trayectoria utilizando el método de Paul [9]. El generador toma los valores de velocidad de viento deseados y los tiempos deseados desde la configuración realizada por el operador en el frontend, y con estos valores se genera una curva de referencia de velocidades de viento en función del tiempo.

El método de Paul consiste en definir, para cada punto, dos zonas: una de aceleración constante y otra de velocidad constante. Cada segmento tiene una duración T_j y un tiempo de aceleración t_{acc} , como se muestra en la figura 5.4, de forma tal que no se realicen cambios abruptos en el controlador, permitiendo que los cambios de velocidad del viento sean suaves.

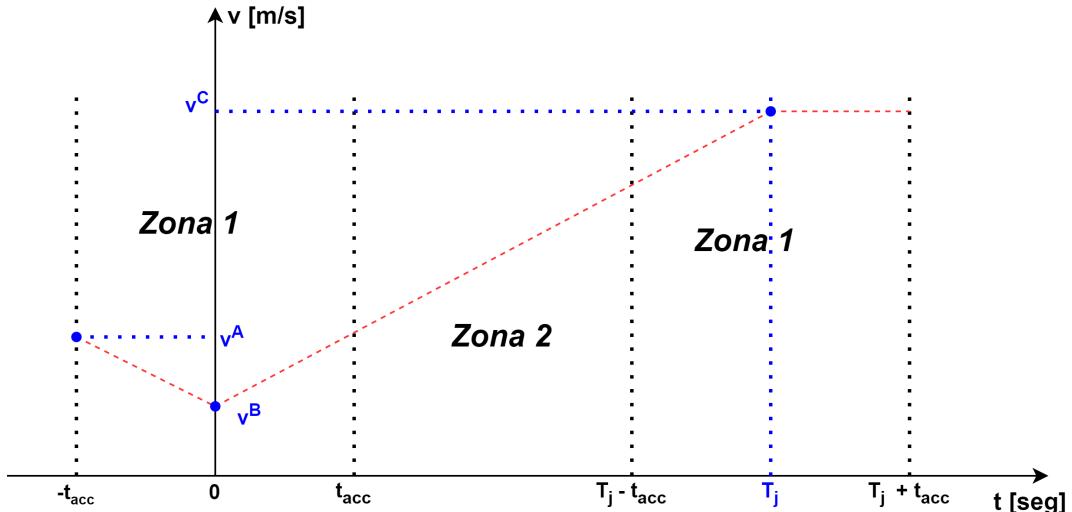


Figura 5.4: Segmento para pasar de un punto de velocidad v^B a otro v^C viendo desde v^A .

Estos cambios suaves se traducen, a través del controlador PID, en cambios suaves en los valores de PWM para el motor.

Para la zona 2 del segmento T_j , donde se define $\Delta C = v^C - v^B$, el método de Paul aplica la ecuación 5.1:

$$v(t_{seg}) = \frac{\Delta C}{T_j} + v^B \quad (5.1)$$

Para la zona 1 del mismo segmento, donde se define $\Delta A = v^A - v^B$, se aplica la ecuación 5.2:

$$v(t_{seg}) = \frac{\Delta C}{T_j} \frac{(t_{seg} + t_{acc})^2}{4t_{acc}} + \frac{\Delta A}{t_{acc}} \frac{(t_{seg} - t_{acc})^2}{4t_{acc}} + v^B \quad (5.2)$$

Esto garantiza que en la zona 1, donde se realiza el enganche de segmento, la aceleración del cambio sea constante, y que en la zona 2, donde se realiza la transición de un segmento a otro, la velocidad de cambio sea constante.

El algoritmo generador de trayectoria se muestra en el código B.1. Este se ejecuta en el back end y calcula la trayectoria en función de cada punto de viento (segmento) y los tiempos de crecimiento, estabilización y medición configurados por el operador. Luego, realiza el encadenamiento de todas las velocidades para obtener un perfil como el de la Figura 5.5, donde se muestra una trayectoria de ciclo ascendente y descendente para los puntos 5, 10, 15, 20 y 25 $m s^{-1}$, con un tiempo de crecimiento de 2 min, un tiempo de estabilización de 5 min y un tiempo de medición de 2 min.

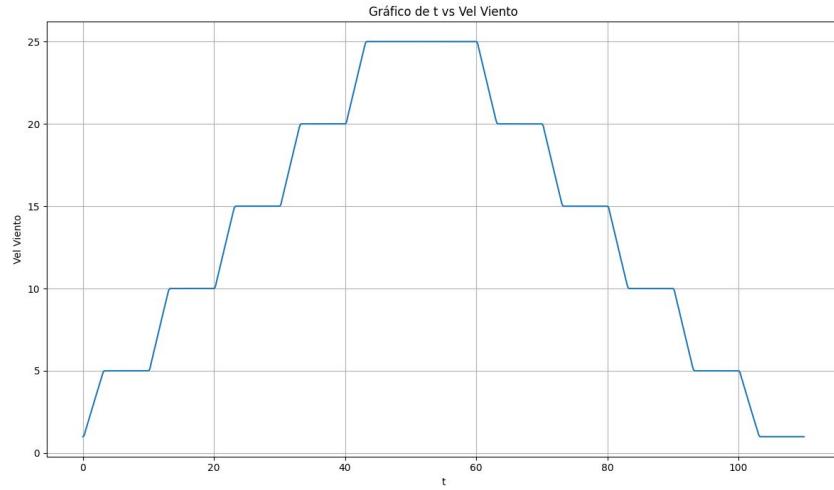


Figura 5.5: Trayectoria generada utilizando el método de Paul para un ciclo ascendente y descendente de velocidades.

5.1.2.4. Cálculo de incertidumbre

El proceso de calcular la incertidumbre expandida para todos los puntos medidos se realiza según el algoritmo del diagrama de flujo de la figura 5.6.

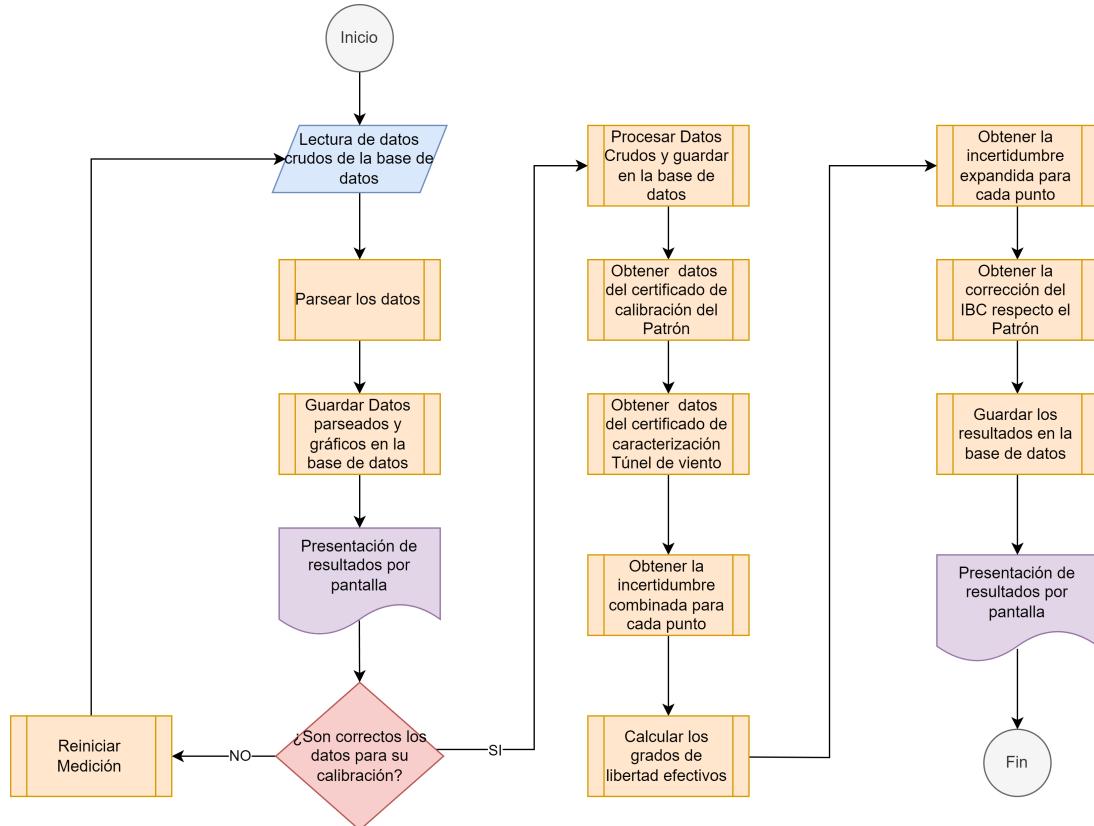


Figura 5.6: Diagrama de flujo para el cálculo de incertidumbre expandida.

Después de completar los ciclos ascendente y descendente de medición, se dispone de un banco de datos crudos de velocidad y dirección. El algoritmo lee estos datos y extrae las mediciones de la parte plana estable para cada punto de forma iterativa. Este proceso se denomina *parsear datos*. Luego, se muestran los datos crudos procesados en pantalla, donde el operador, desde el front-end, debe hacer clic en el botón “Calcular Incertidumbre”, siempre que esté de acuerdo con los datos. Si no, puede reiniciar las mediciones con un nuevo perfil.

Si los datos son correctos para la calibración, se calcula el promedio \bar{v} y el desvío estándar σ_v de las mediciones de velocidad del viento para cada punto. A continuación, se elabora el presupuesto de incertidumbre, obteniendo los datos de histéresis del patrón y del IBC a partir de la diferencia de los valores promedio. Se procede a la lectura de los datos ingresados por el operador, específicamente la resolución de los instrumentos y el área de bloqueo de los anemómetros para el cálculo del factor de bloqueo, ecuación 5.3, donde S_T corresponde al área transversal de la zona de medición del túnel de viento y S_A se corresponde al área efectiva de bloqueo del sensor y su soporte. Este valor se utiliza para corregir las mediciones del IBC y del patrón, y se incorpora su respectiva incertidumbre según la ecuación 5.4.

$$F_{bk} = \frac{S_A}{S_T} \quad (5.3)$$

$$\Delta F_{bk} = \sqrt{\left| \frac{\partial F}{\partial S_T} \right|^2 |\Delta S_T|^2 + \left| \frac{\partial F}{\partial S_A} \right|^2 |\Delta S_A|^2} = \left(\frac{-S_A}{S_T^2} \right)^2 (\Delta S_T)^2 + \left(\frac{1}{S_T} \right)^2 (\Delta S_A)^2 \quad (5.4)$$

Luego, se obtienen los datos de calibración del sensor patrón a partir de un ajuste lineal de los puntos discretos del certificado y evaluando los puntos configurados por el operador. Esto se hace así porque no siempre el patrón está calibrado en los mismos puntos en los que se lo va a utilizar. Lo mismo se realiza con el certificado del túnel, extrayendo los valores de estabilidad, homogeneidad y el factor de calibración del túnel para los puntos configurados por el operador.

Con todas estas incertidumbres normalizadas y multiplicadas por su coeficiente de sensibilidad c_i , se obtiene la incertidumbre estándar para cada punto. Luego, se realiza la suma de los cuadrados de cada incertidumbre y se calcula la raíz cuadrada para obtener la incertidumbre combinada para cada punto

$$u_c = \sqrt{\sum_{i=1}^n (u_i^2)}.$$

En base al número de mediciones, se calculan los grados de libertad efectivos para cada punto y con este número, solicitando un intervalo de confianza del 95 %, se evalúa la distribución t de Student para obtener el factor de cobertura (k). Luego, se multiplica este número por cada

incertidumbre combinada correspondiente para obtener la incertidumbre expandida

$$U = k \cdot u_c.$$

Finalmente, todos los resultados se presentan por pantalla y se dispone de botones para descargar los gráficos y tablas correspondientes a los datos crudos, la histéresis y los resultados de calibración.

5.1.3. Front End

El front end se encarga de:

- Interacción del usuario a través de formularios, botones y eventos.
- Comunicación con el backend a través de Django y WebSockets.
- Presentación de tablas y gráficos al usuario (UI/UX).

La aplicación web presenta una plantilla de inicio, como se muestra en la figura 5.7. En la cabecera se visualiza, el logotipo correspondiente al SMN, el nombre del sistema y el nombre y tipo de perfil de usuario. En el lado izquierdo, se encuentran cuatro botones: **Temperatura** y **humedad**, **Temperatura**, **Presión** y **Viento**, cada uno con su respectiva descripción. Este diseño permitirá escalar el software para incluir otras magnitudes con las que trabaja el laboratorio del SMN. En el pie de página, se presentan los enlaces a las herramientas utilizadas para desarrollar el proyecto.

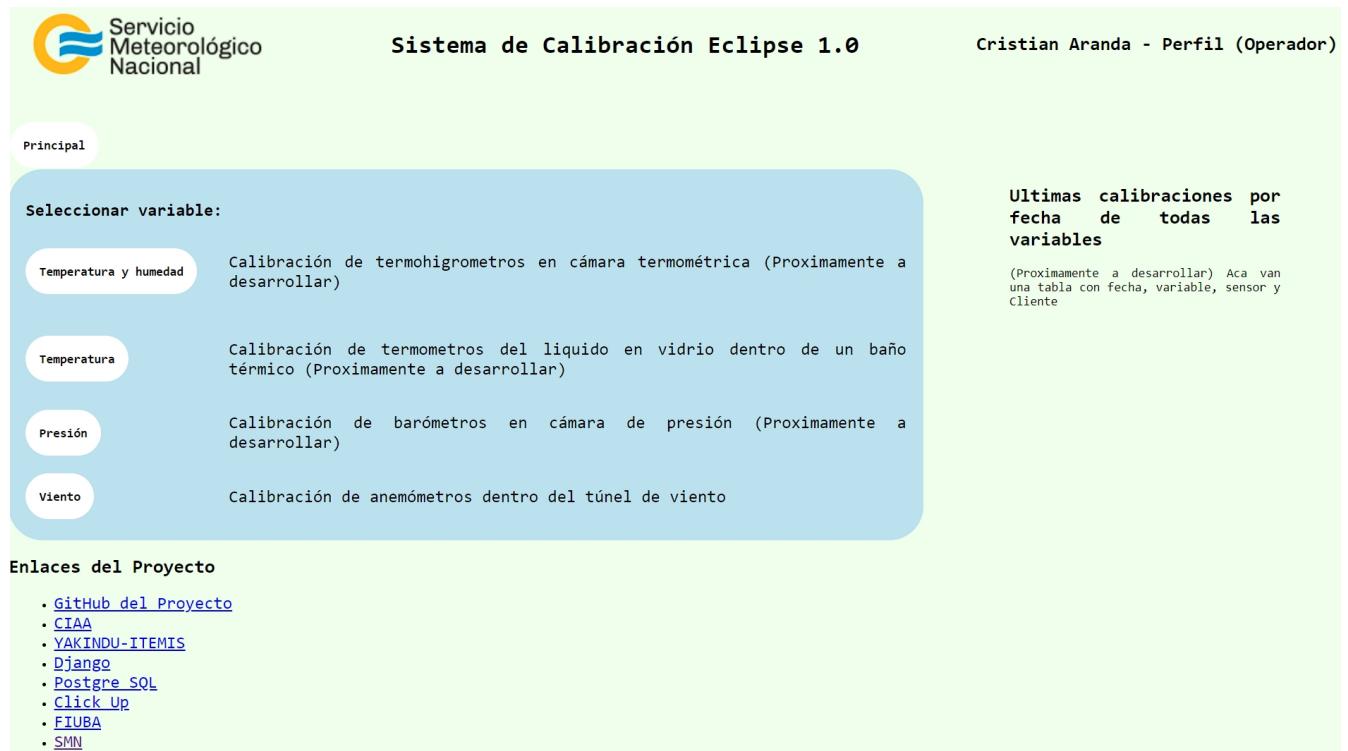


Figura 5.7: Pantalla de inicio de la aplicación web.

El usuario hace click en el botón **Viento** y se activa la vista de la Figura 5.8, donde le consulta al usuario, si desea iniciar o no una calibración de anemómetros.

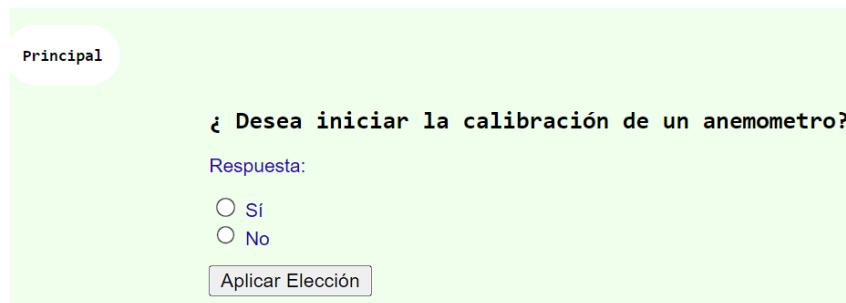


Figura 5.8: Vista para confirmar si se desea realizar o no una calibración.

Si el usuario responde afirmativamente, se abre a izquierda una barra de navegación, como se muestra en la figura 5.9, que permite navegar por la aplicación web. En el cuerpo de la aplicación se presenta una serie de formularios y botones para cargar los metadatos, configurar los equipos, iniciar las mediciones y, finalmente, procesar y presentar los resultados. Además, el usuario puede utilizar el botón **Salir**, que lo redirige a la pestaña de inicio.



Figura 5.9: Barra lateral izquierda de navegación.

5.1.3.1. Carga de metadatos

El contenido principal (*main*) de la primera vista, llamada **Carga de Metadatos**, se muestra en la Figura 5.10. En esta vista, se deben ingresar la marca, el modelo, el número de serie, el número de patrimonio (identificador de instrumentos del SMN), la resolución y el área de bloqueo del instrumento medido previamente con una cinta métrica, tanto del sensor patrón como del sensor bajo calibración. Luego, se debe cargar la información del certificado de calibración del instrumento patrón utilizado, incluyendo la fecha de generación, los puntos en los que fue calibrado, los valores obtenidos por cada punto, las correcciones obtenidas y su respectiva incertidumbre. Por último, se deben ingresar los datos del certificado de caracterización del túnel de viento, especificando la fecha de emisión, los puntos medidos en el ensayo, los valores de estabilidad y de homogeneidad del flujo de aire en la zona de medición. Además, se debe agregar el factor de calibración definido en la tabla 3.1. Una vez cargada toda la información, el operador debe presionar el botón **Cargar**, o si desea borrar toda la información del formulario, debe presionar **Limpiar campos**. Parte de esta información luego será utilizada para el cálculo de la incertidumbre expandida del sensor bajo calibración y toda los datos ingresados se guardan en la base de datos.

Figura 5.10: Vista para cargar los metadatos de la calibración.

5.1.3.2. Configuración del sistema

El contenido principal (*main*) de la segunda vista, denominada **Configuración**, se muestra en la Figura 5.11. En esta vista, a la izquierda, se configura el datalogger diseñado en el capítulo 4. Se pueden seleccionar la marca, el modelo y el número de serie del datalogger. Posteriormente, se especifica el tiempo de muestreo y el tiempo de tabla, se configuran los puertos de comunicación serie del datalogger, los LEDs de recepción y la velocidad en baudios para cada anemómetro.

Figura 5.11: Vista para configurar los equipos involucrados en la calibración.

A la derecha, se configura el túnel de viento. Aquí se establecen los puntos mínimo y máximo

de velocidad, la cantidad de puntos donde se desea medir, los valores de velocidad en metros por segundo, y el tipo de perfil de medición, que puede ser ascendente, descendente o ambos. También se configuran los tiempos de transitorio, de estabilidad y de medición para cada punto, necesarios para generar el perfil de trayectoria explicado en la sección 5.1.2.3. Estos tiempos se ilustran sobre el escalón de la Figura 5.12.

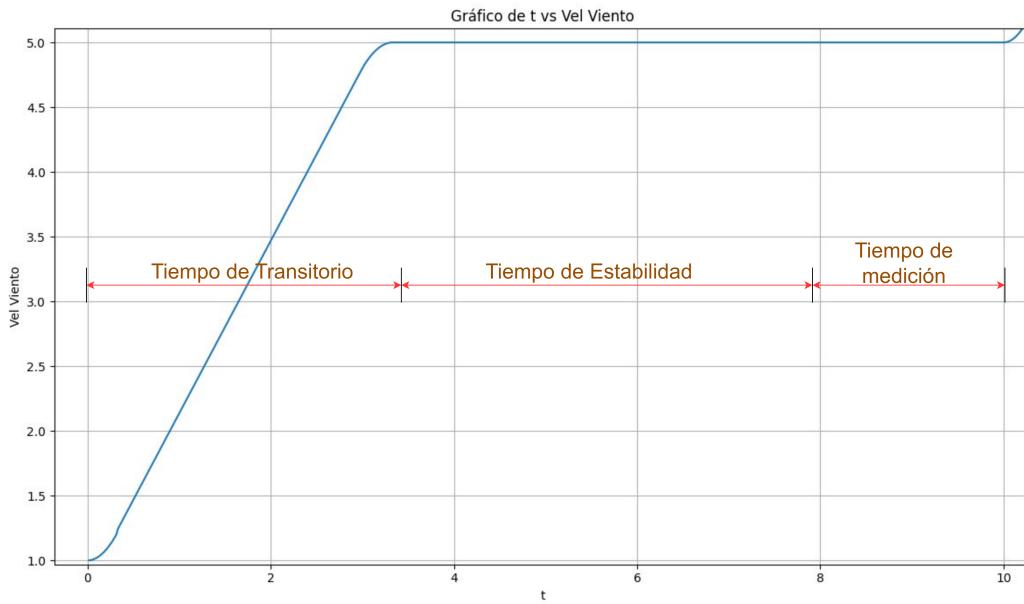


Figura 5.12: Configuración de tiempos para el generador de trayectoria en 5 m s^{-1} .

Una vez que los datos de configuración están cargados, el usuario puede hacer clic en el botón **Cargar** para subirlos a la base de datos o en **Limpiar campos** para borrar los campos de esta vista.

5.1.3.3. Adquisición de datos

Una vez cargados todos los datos, el software muestra la vista de la Figura 5.13. Al hacer clic en el botón **Iniciar Medición**, se envían todas las configuraciones al datalogger a través del servidor WebSocket mediante comandos detallados en la Tabla B.1. Luego, se activa un temporizador que multiplica el tiempo de un escalón, como el de la Figura 5.12, por la cantidad de puntos ingresados e indica el tiempo restante para que termine todas las mediciones.

El software también envía, a intervalos de un segundo, un valor de referencia del generador de trayectoria al datalogger para que éste genere un nivel de PWM y lo envíe al variador de velocidad del motor. Al mismo tiempo, se reciben las mediciones del sensor patrón e IBC, y se va construyendo el perfil de mediciones, con la dirección del viento a la izquierda y la intensidad a la derecha.

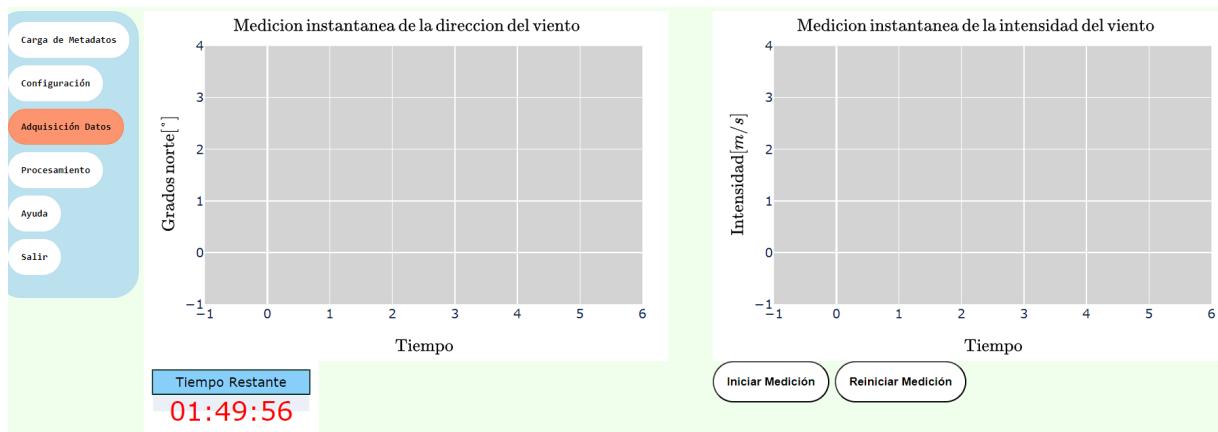


Figura 5.13: Vista de la adquisición de datos.

En caso de que el operador precise detener o reiniciar las mediciones puede activar el botón **Reiniciar Medición**, el cual genera la vista de la figura 5.14, que informa que si reinicia se borrarán las mediciones ya adquiridas.

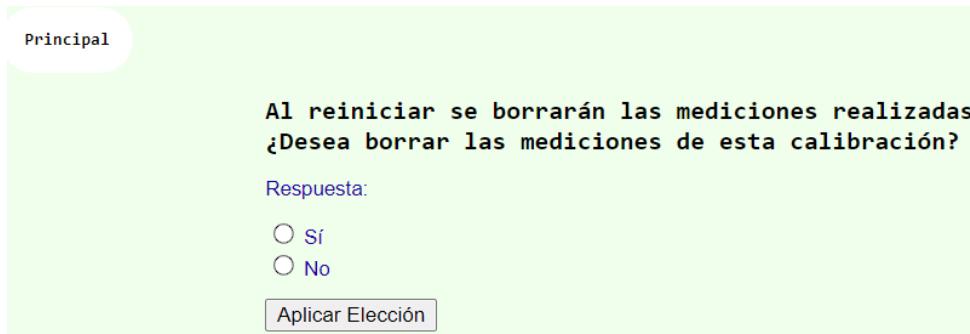


Figura 5.14: Vista para confirmar si se desea borrar o no los datos crudos de una calibración.

5.1.3.4. Procesamiento de datos

Luego de que se acabe el tiempo del temporizador, el software redirige a la vista **Procesamiento** que se muestra en la Figura 5.15. En esta vista, se grafican las curvas ascendente y descendente de las velocidades de viento del sensor patron y bajo calibración. En el título del gráfico se indica la magnitud, el número de serie y patrimonio del sensor bajo calibración. Toda la información y datos se leen desde la base de datos, ya que se fueron almacenando en la misma a medida que ingresaron. Además, cada gráfico cuenta con un botón para ser descargado.

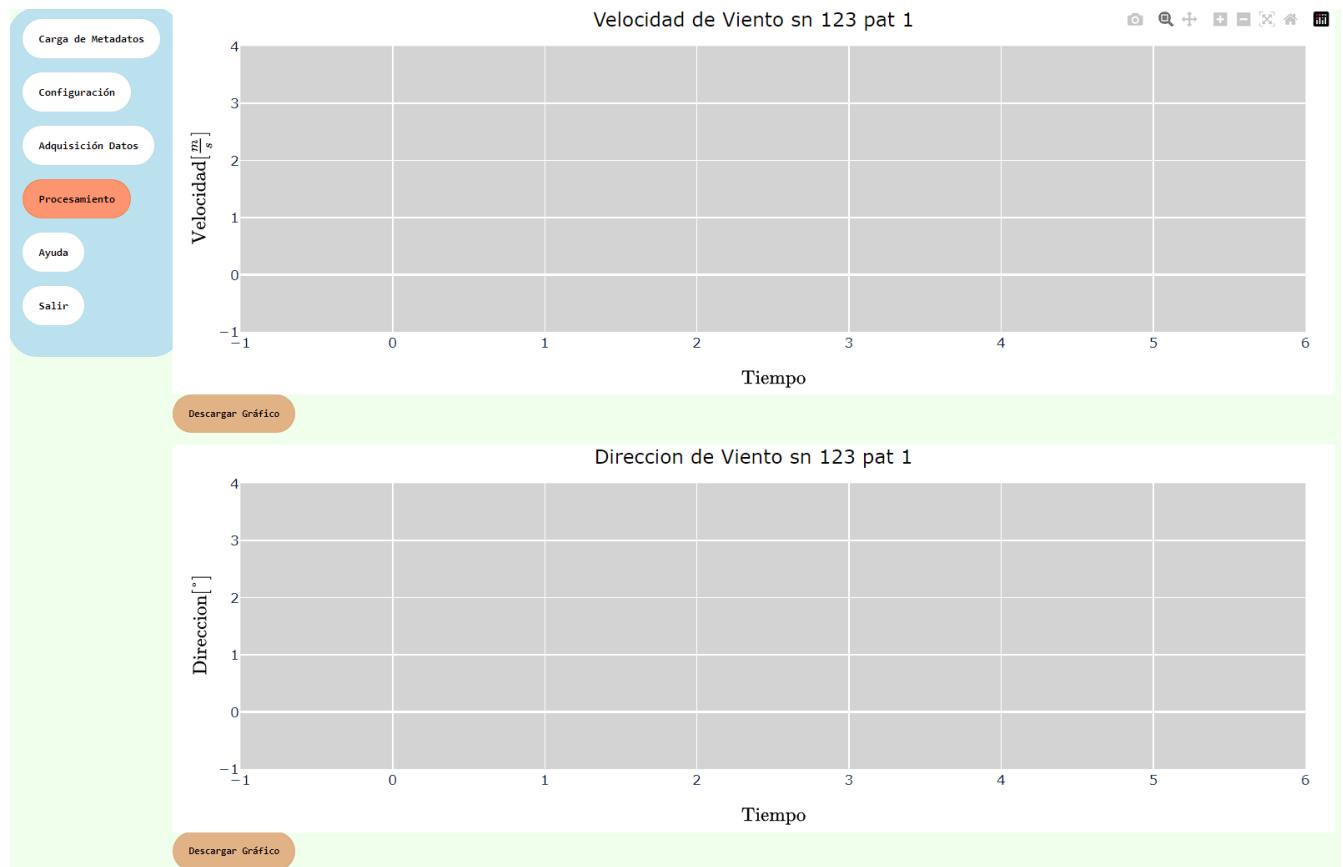


Figura 5.15: Vista para el procesamiento de datos y visualización de gráficos.

En la parte inferior de la vista, como se ve en la figura 5.16 se muestran tablas con los datos parseados, que corresponden a la extracción de los datos de la parte plana, equivalente al tiempo de medición, como por ejemplo en el gráfico de la Figura 5.12 es de alrededor de 120 muestras alrededor de 5 m s^{-1} . El operador revisa estos datos y, si están dentro del entorno del punto de medición configurado, debe presionar el botón **Calcular Incertidumbre**. En caso contrario, puede reiniciar las mediciones, lo que lo devolverá a la vista de adquisición de datos. De igual forma cada tabla cuenta con un botón para ser descargado en formato .csv



Figura 5.16: Vista para la revisión de datos parseados y cálculo de incertidumbre.

5.1.3.5. Resultados

Cuando se hace clic en el botón calcular incertidumbre, se activa el programa del diagrama de flujo de la Figura 5.6. En las Figuras 5.17 y 5.18 se muestra el espacio donde se carga el primer resultado, que consiste en un gráfico con la curva de histéresis y en su leyenda se agregan las funciones obtenidas a partir de un ajuste lineal de los puntos. Además, se presenta una tabla con los valores promedio y su histéresis, calculada como la diferencia de los datos ascendentes y descendentes. Tanto el gráfico como la tabla cuentan con botones para descargar.

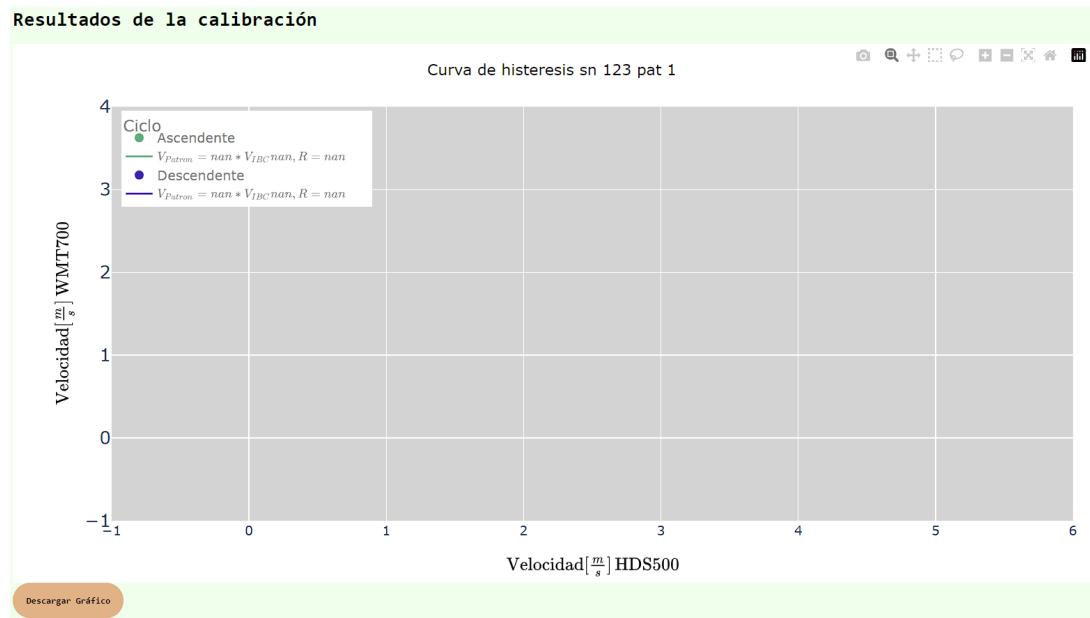


Figura 5.17: Curva de histéresis.

Tabla de Histeresis

Promedio Patron Asc	Promedio Patron Des	Histeresis Patrón	Promedio IBC Asc	Promedio IBC Des	Histeresis IBC
null	null	null	null	null	null
null	null	null	null	null	null

Descargar Tabla

Figura 5.18: Tabla de valores promedio e histéresis.

En las Figuras 5.19 y 5.20 se muestran los espacios donde se cargan los resultados de la calibración. Estos resultados comprenden las curvas de calibración ascendente y descendente con sus respectivas tablas, donde se presentan las correcciones, la incertidumbre combinada y expandida, y los grados de libertad para cada punto.

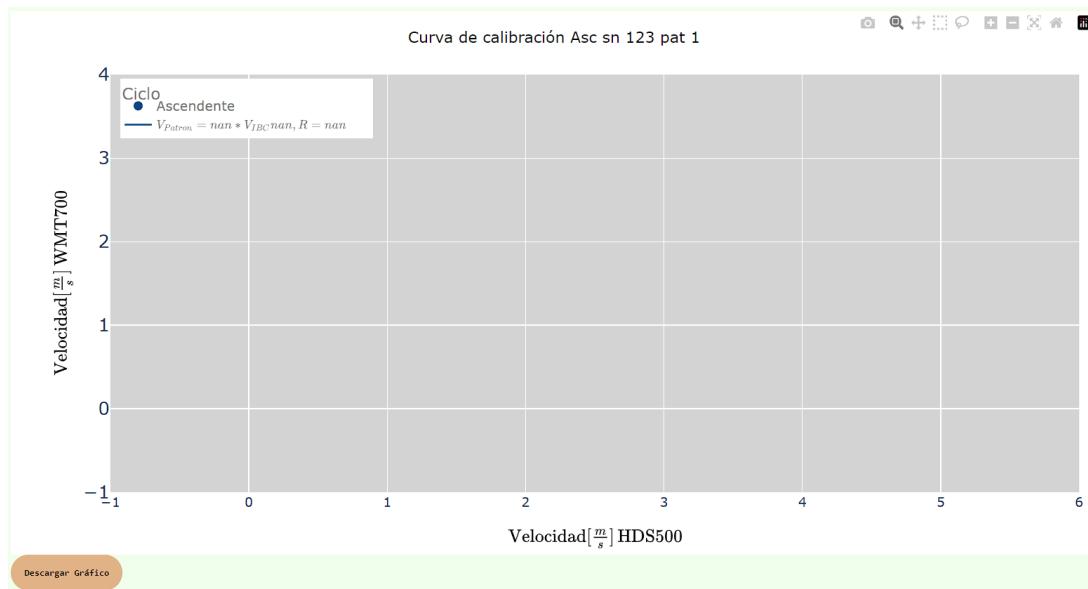


Figura 5.19: Curva de calibración.

Tabla con resultados del ciclo ascendente					
Velocidad Promedio Corregida del Patrón Asc	Velocidad Promedio IBC Asc HDS500	Correcciones IBC Asc	Incertidumbre Combinada IBC Asc	Incertidumbre Expandida IBC Asc	Grados De Libertad Combinada IBC Asc
null	null	null	null	null	null
null	null	null	null	null	null

[Descargar Tabla](#)

Figura 5.20: Tabla de correcciones, incertidumbre y grados de libertad.

5.2. Integración con el hardware

El software desarrollado interactúa con el hardware, permitiendo al operador configurar de manera sencilla e intuitiva la interfaz eléctrica de los anemómetros, la cadencia de toma de muestras y los puntos de velocidad del viento en el túnel. Se realiza automáticamente el cálculo de incertidumbre y se genera un reporte con los resultados, para el instrumento bajo estudio. Este trabajo estandariza un procedimiento automático para la calibración de sensores de viento ultrasónicos, reduciendo los tiempos operativos, el cálculo manual y los errores sistemáticos, mejorando la calidad en la calibración. Además, el sistema de adquisición, en conjunto con el software y otro sistema de posicionamiento (no incluido en esta tesis), permitirá estudiar en

mejor detalle el comportamiento del flujo de aire en la zona de medición.

Los pasos para realizar el proceso de calibración incluyendo el diagrama de la figura 5.21 se define a continuación:

- Se debe armar el banco de medición, montando los sensores en un soporte dentro del túnel de viento, asegurando que queden fijos y alineados.
- Se deben tomar las medidas de posición de los instrumentos respecto a un sistema de referencia.
- Se debe calcular el área de bloqueo de cada instrumento, incluyendo su soporte.
- Se deben conectar los anemómetros al datalogger y alimentar todo el sistema con una fuente de 12 voltios. Además, se debe conectar el datalogger a la red LAN local del laboratorio.
- Se debe acceder a la interfaz web y configurar los parámetros de calibración.
- Se debe iniciar la adquisición de mediciones mediante la interfaz web.
- La aplicación Django procesa las solicitudes y envía comandos al datalogger a través de WebSocket.
- El datalogger controla el variador de velocidad del motor mediante señales PWM, ajustando el flujo de aire en el túnel de viento.
- Los datos de los anemómetros y otros sensores se registran y envían a la aplicación Django.
- Terminada las mediciones, el operador debe revisar las mediciones y si son correctas activar el procesamiento.
- La aplicación procesa los datos. Primero, calcula el promedio y el desvío estándar de las mediciones para cada punto del escalón. Luego, con esos datos e información ingresada por el usuario, calcula las correcciones e incertidumbres. Finalmente, almacena los resultados en la base de datos.
- Los resultados pueden ser visualizados y descargados a través de la interfaz web.

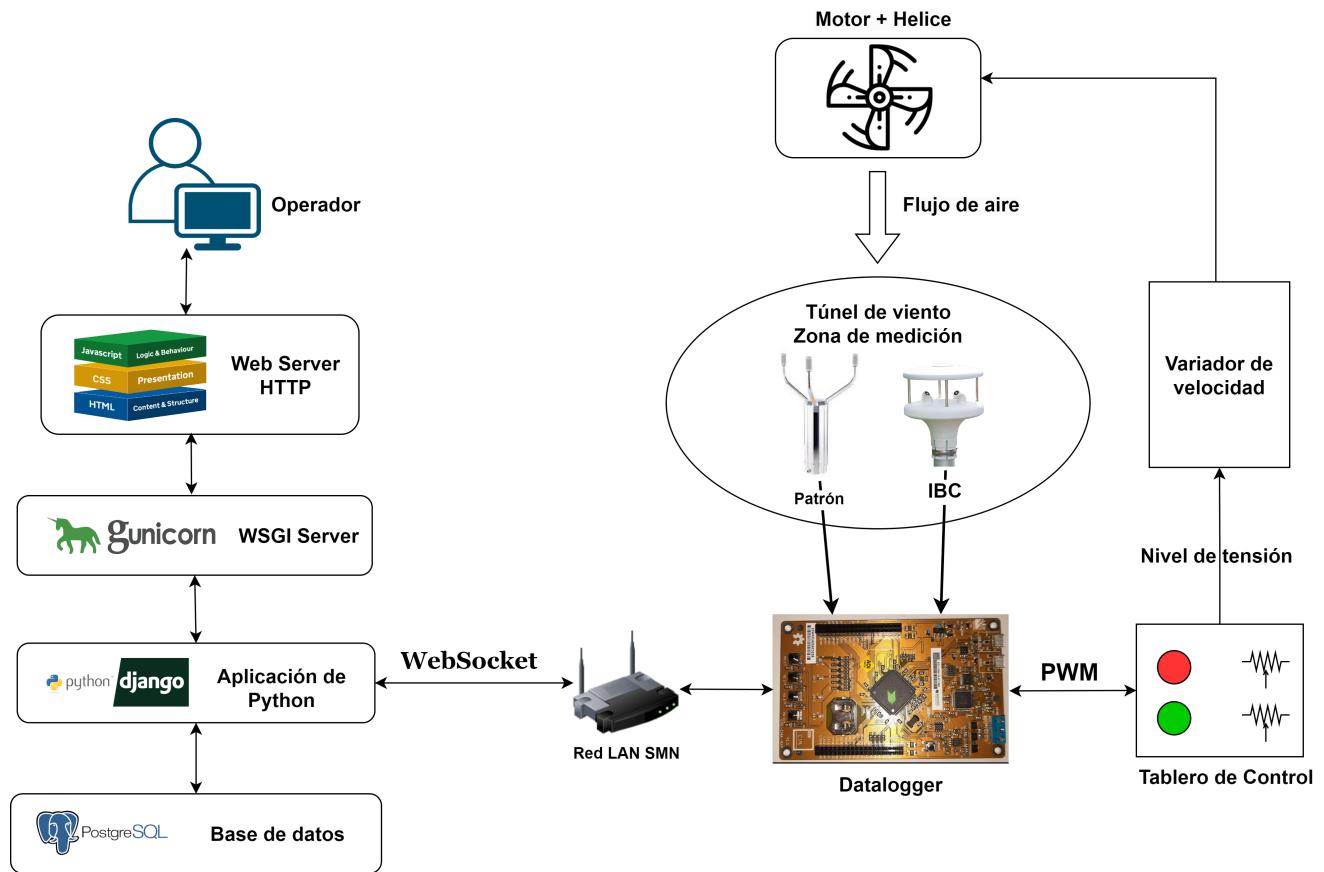


Figura 5.21: Sistema integrado desarrollado en esta tesis.

Capítulo 6

Mediciones y Resultados Experimentales

En este capítulo, se ha utilizado el sistema de calibración, compuesto por la placa desarrollada en el Capítulo 4 y el software descrito en el Capítulo 5, para realizar dos ensayos. Inicialmente, se identificarán dos posiciones dentro del túnel de viento que presenten condiciones similares de flujo de aire para la medición de los anemómetros. Posteriormente, se detallarán dos ejemplos de calibración de anemómetros en el túnel de viento, utilizando el sistema mencionado.

Para todas las mediciones usaremos el siguiente sistema de referencia definido en la figura 6.1, donde el eje **z** se corresponde con el eje vertical, el eje **y** se corresponde con el eje longitudinal y el eje **x** se corresponde con el eje trasversal.

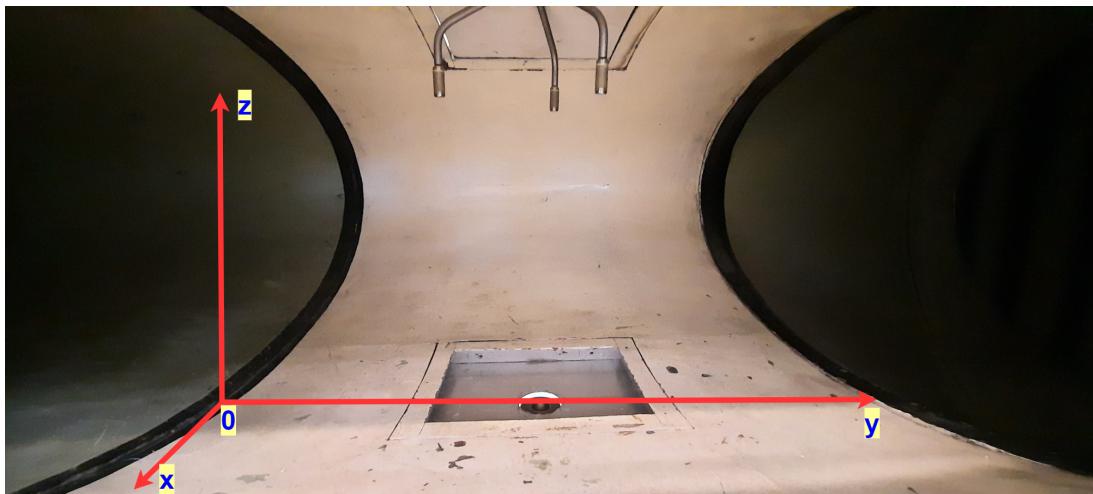


Figura 6.1: Sistema de referencia utilizado para realizar las mediciones adentro del túnel de viento.

6.1. Caracterización de la zona de medición

El flujo de aire a lo largo, alto y profundo en toda la zona de medición no es constante debido a los cambios de sección presentes al principio y al final de la zona de medición, lo cual provoca variaciones en la intensidad hasta alcanzar el régimen permanente del flujo. Para determinar dos posiciones donde el flujo de aire presenta la misma intensidad, una para el anemómetro patrón y otra para el anemómetro bajo calibración, se realizó un ensayo considerando las limitaciones de las dimensiones de los anemómetros dentro del volumen de medición. El objetivo del ensayo fue identificar dos posiciones en el espacio donde la potencia suministrada al motor resultara en la misma velocidad del viento en cada posición.

Para este ensayo, se utilizó el sensor patrón Vaisala, modelo WMT700. Se simplificó el problema a una dimensión, manteniendo constante la altura del sensor en 39,5 cm respecto la base de túnel y centrando el sensor en el medio. Solo se varió el eje longitudinal desde -20 cm hasta 60 cm en incrementos de 10 cm. En cada posición, se configuró la aplicación web para medir y controlar la velocidad del aire a 5 m s^{-1} , 10 m s^{-1} , 15 m s^{-1} , 20 m s^{-1} y 25 m s^{-1} .

En la Figura 6.2 se muestra el sensor en tres posiciones específicas: a -20 cm, a 20 cm y a 60 cm respecto del origen del sistema de referencia ilustrado en la Figura 6.1.

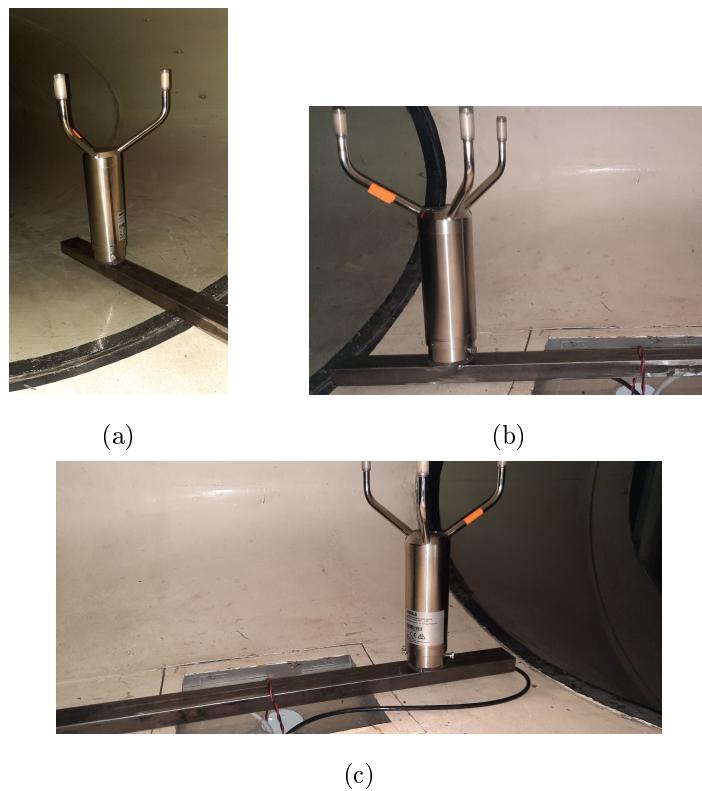


Figura 6.2: En (a) se muestra el anemómetro midiendo a -20 cm, en (b) se muestra midiendo a 20 cm y en (c) se encuentra midiendo a 60 cm.

En la figura 6.3 se observan las velocidades del viento para las distintas posiciones en los cinco puntos programados. El sistema de control PID se encarga de suministrar la potencia necesaria para que el viento dentro del túnel, en la posición del sensor patrón, alcance y se estabilice en el valor configurado.

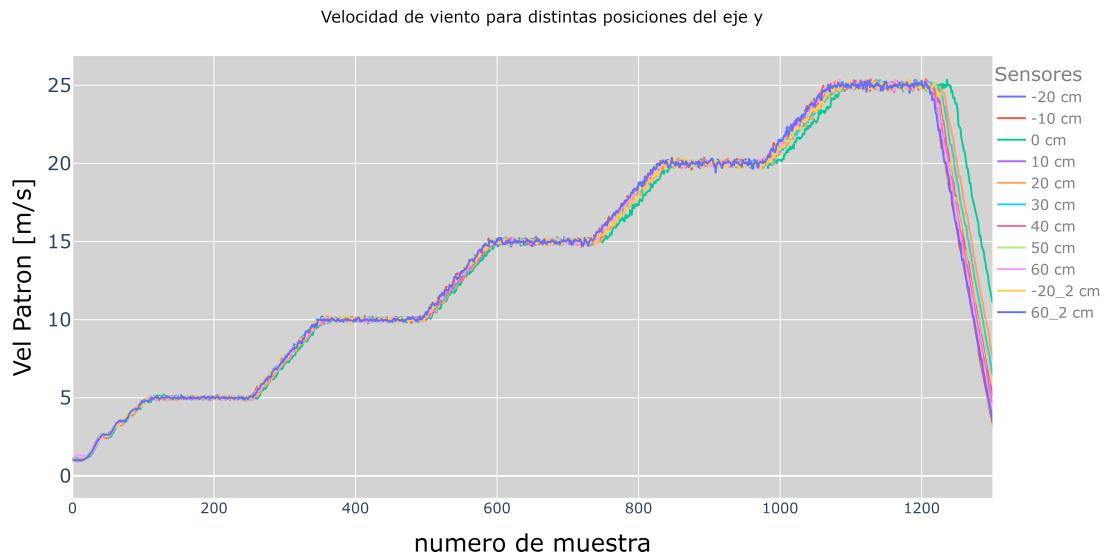


Figura 6.3: Velocidad de viento en cinco puntos, para distintas posiciones.

Los resultados de las mediciones del nivel de modulación por ancho de pulso (PWM) para cada posición, correspondientes a las cinco velocidades configuradas, se ilustran en la Figura 6.4. Se puede apreciar que, para una misma velocidad de viento, pero en diferentes posiciones a lo largo del eje longitudinal, el sensor requiere distintos niveles de PWM para generar la misma velocidad de viento.

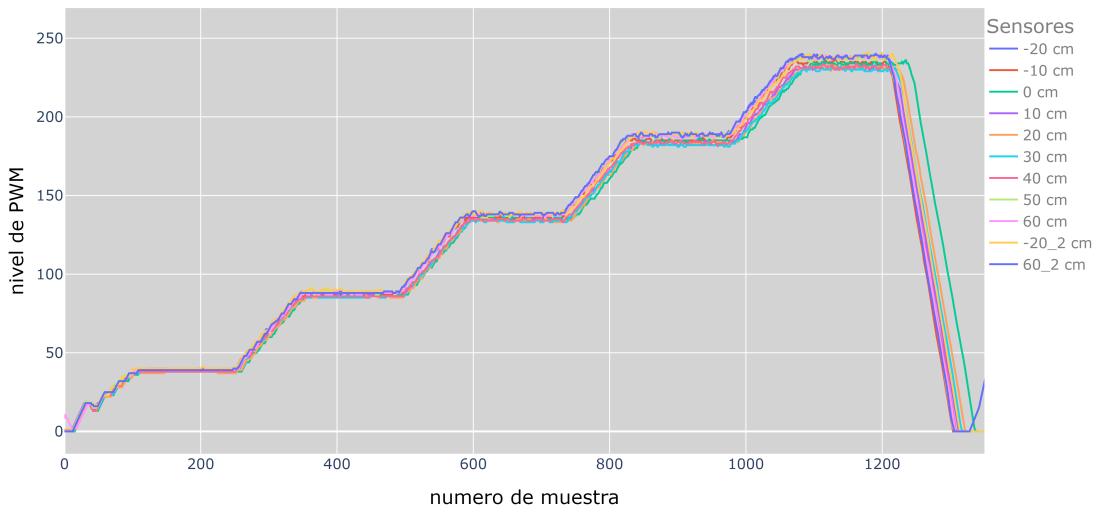


Figura 6.4: Nivel de PWM medido en el controlador, para distintas posiciones, obtenidas utilizando el software y hardware desarrollados en capítulos anteriores.

En la Figura 6.5, se realiza un acercamiento a 25 m s^{-1} , donde se pueden observar las variaciones de los niveles de PWM para la misma velocidad en función de la posición. Se destaca que para posiciones entre 0 cm y 40 cm, los niveles de PWM son más bajos, oscilando entre 229 y 235. Sin embargo, para posiciones alejadas a izquierda y posiciones más en el centro, como -10 cm , -20 cm , 50 cm y 60 cm , los niveles de PWM son más altos, aunque en promedio similares entre sí. Es importante que los sensores patrón e IBC estén suficientemente separados para minimizar el bloqueo cruzado [3] [1], manteniendo al mismo tiempo condiciones de flujo de aire idénticas.

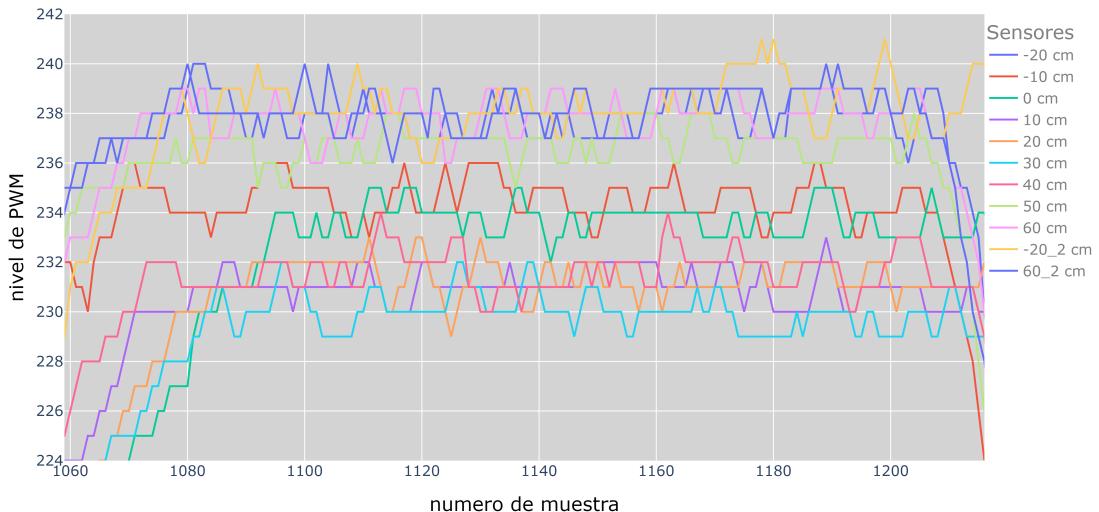


Figura 6.5: Niveles de PWM para distintas posiciones del anemómetro a 25 m s^{-1} .

Considerando lo anterior y basándose en los resultados gráficos, se concluye que los valores de PWM necesarios para obtener una velocidad determinada son equivalentes en las posiciones de -20 cm y entre 40 cm y 50 cm . Esto asegura que los sensores estén separados entre 60 cm y 70 cm . En la Figura 6.6, se muestran dos curvas correspondientes a -20 cm , dos a 60 cm y una a 50 cm . Aquí se puede apreciar que para una velocidad de viento determinada, el controlador entrega en promedio el mismo nivel de PWM, lo que indica que en estos puntos, el comportamiento del flujo de aire presenta características similares.

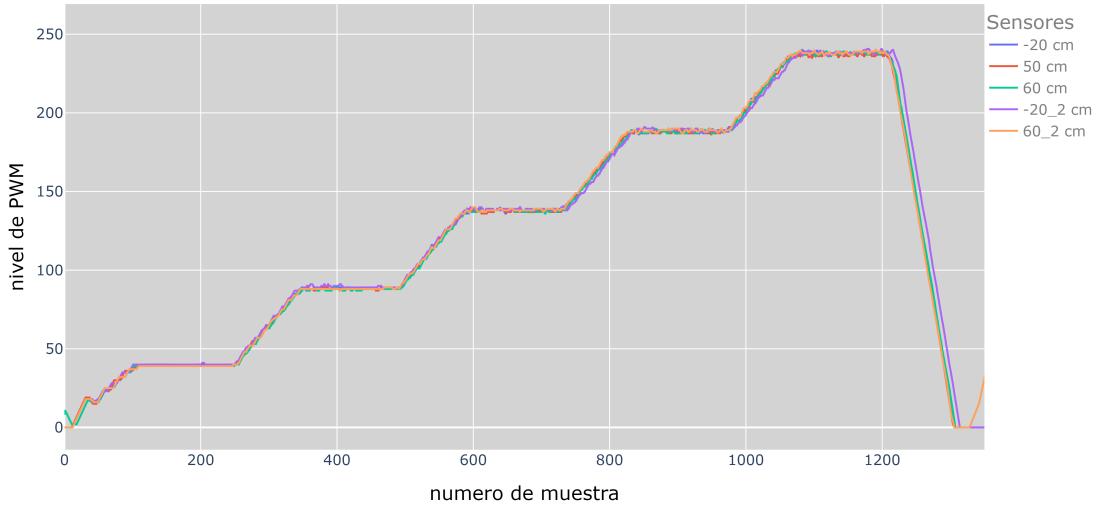


Figura 6.6: Niveles similares de PWM para distintas posiciones del anemómetro.

Esta investigación proporciona una primera aproximación al comportamiento del flujo de aire dentro del túnel de viento. Sin embargo, esta aproximación está limitada por varios factores. Por ejemplo, el sensor utilizado para el análisis presenta una altura de soporte que limita la variable z . Debido al área de bloqueo, la medición debe ser corregida. Para este tipo de ensayo, se podría utilizar un sensor con menor área de bloqueo, como un tubo Pitot o un anemómetro de hilo caliente. A pesar de estas limitaciones, como primera aproximación, es suficiente para realizar las primeras calibraciones. Esto se debe a que se garantiza que el flujo de aire en las posiciones seleccionadas tiene un comportamiento similar, generando una velocidad de viento comparable para el mismo nivel de modulación por ancho de pulso (PWM).

6.2. Calibración DeltaOHM HD51.3

Se realizó la calibración de un sensor ultrasónico de viento marca DeltaOhm, modelo HD51.3, mostrado en la figura 4.4b, utilizando como sensor patrón el de la figura 4.4a, modelo WMT700 de la marca Vaisala. El procedimiento de calibración se basa en los pasos detallados en la sección

5.2.

6.2.1. Banco de medición y configuración del hardware

En la Figura 6.7a se muestra el montaje de los sensores sobre una barra de acero, donde se instalaron con una distancia d de 65 cm y a una altura h respecto a la base del soporte de 36,5 cm. Los sensores quedan alineados y nivelados en el eje longitudinal de la barra, como se muestra en las Figuras 6.7b y 6.7c.

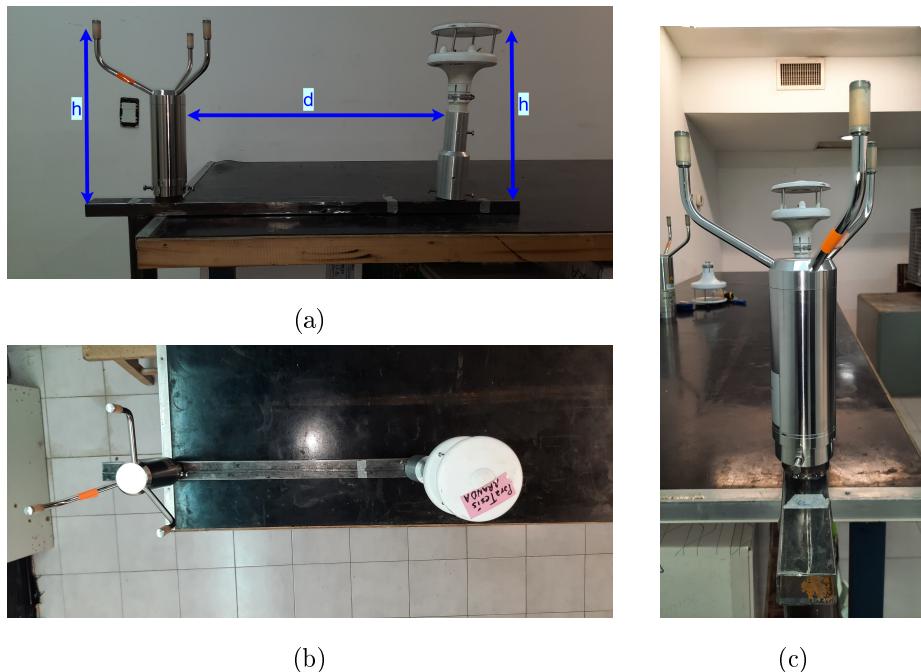


Figura 6.7: En (a) se muestra la vista frontal de los sensores patrón y bajo prueba, en (b) se observa una vista superior para ver la alineación, y en (c) se muestra una vista lateral.

Luego de ensamblar y alinear los sensores, se debe instalar el brazo dentro del túnel de viento. Es crucial conectar previamente los cables de los sensores antes de ajustar los prisioneros. En la Figura 6.8 se observa el sensor montado en el túnel de viento. El sensor bajo calibración se encuentra a -20 cm y el sensor de referencia está a 45 cm, ambos a una altura de 39,5 cm respecto al piso del túnel. Todo el conjunto está asegurado con la barra telescópica mostrada en la Figura 2.13c.



Figura 6.8: Brazo con los sensores bajo calibración a izquierda y patrón a derecha dentro del túnel.

En la mesa de trabajo se instala el datalogger, al cual se conectan el sensor bajo calibración y el sensor patrón a través de sus puertos RS485. Tanto los sensores como el datalogger se alimentan con una fuente de 12 V. El módulo Ethernet se conecta al switch, y el nivel de tensión continua generado por el PWM se conecta al variador de velocidad del túnel.

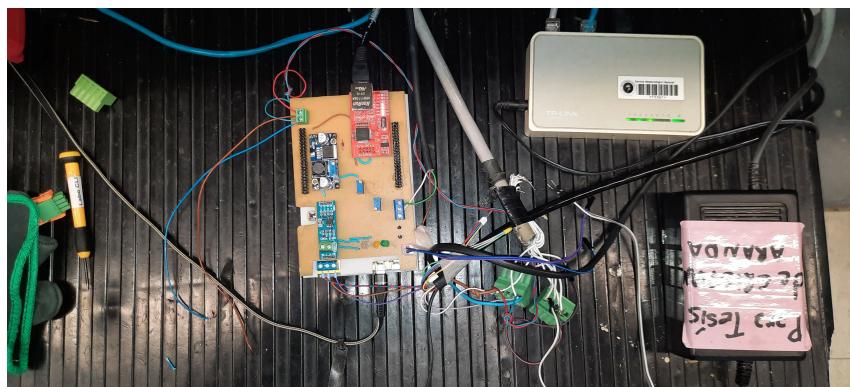


Figura 6.9: Mesa de trabajo con el datalogger y los insumos necesarios para su funcionamiento.

6.2.2. Configuración de la aplicación web

En las Figuras 6.10a y 6.10b se muestra la carga de metadatos de los sensores. En la Figura 6.10c se presenta la carga de datos del certificado de calibración del sensor patrón. Por otro lado, en la Figura 6.10d se deben cargar los datos de certificado de caracterización del túnel de viento, los valores se cargan como nulos, ya que aún no se ha realizado el estudio de homogeneidad y estabilidad de la zona de medición. Estos valores nulos aportan una incertidumbre nula al presupuesto de incertidumbre. No obstante, una vez realizados estos estudios, se podrán agregar los valores obtenidos sin mayor dificultad, y el sistema incorporará su aporte de incertidumbre al presupuesto.

Sensor IBC

- Marca: DELTA_OHM
- Modelo: HDS500
- Número de serie: 20017380
- Número de patrimonio: 22100
- Resolución: 0,01
- Área de bloqueo IBC: 0,036

Sensor PAT

- Marca: VAISALA
- Modelo: WMT700
- Número de serie: S3420313
- Número de patrimonio: 0
- Resolución: 0,01
- Área de bloqueo Pat: 0,031

Datos Cert. Calibración

- Fecha de emisión: 19/10/2022
- Puntos calibrados: 0,2,4,6,8,10,12,14,16,18
0,27,2,09,4,05,5,17,8,26,18,25,12,34,14,39,1
6,42,18,37
- Valor de la medición:
- Correcciones de la medición:
- Incertidumbre:

Datos Cert. Caracterización de túnel

- Fecha de emisión: 15/10/2021
- Puntos medidos en el ensayo: 5,10,15,20,25
0,0,0,0,0
- Valores de estabilidad:
- Valores de homogeneidad:
- Factor de calibración del túnel: 0,0

Figura 6.10: En (a), (b), (c) y (d) se muestran los metadatos ingresados el por el usuario para realizar la calibración.

En la figura 6.11a se muestra la configuración del datalogger. Se elige un tiempo de muestreo de 1 s y un tiempo de tabla también de 1 s, ya que nos interesa el dato instantáneo para realizar la calibración. Se selecciona el puerto COM2 (RS485 agregado con el módulo Maxrs485) para el IBC y el puerto COM1 (RS485 propio de la EDU-CIAA) para el sensor patrón. Ambos sensores tienen un baud rate de 9600.

En la figura 6.11b se muestra la configuración del túnel de viento. Se establece un punto mínimo de 1 m s^{-1} y un punto máximo de 25 m s^{-1} . Se indican 5 puntos para calibrar en 5 m s^{-1} , 10 m s^{-1} , 15 m s^{-1} , 20 m s^{-1} y 25 m s^{-1} . Se configura un ciclo ascendente-descendente de 5 m s^{-1} a 25 m s^{-1} y luego de 25 m s^{-1} a 5 m s^{-1} . Se establece un tiempo de transitorio para llegar de un punto a otro de 3 min, un tiempo para estabilizar las mediciones de 5 min y un tiempo de medición de 2 min.

Configurar Datalogger

- Marca del datalogger: ASEMBLI
- Modelo: EDU-CIAA
- Número de serie: 123
- Tiempo de muestreo: 1
- Tiempo de tabla: 1
- Puerto COM IBC: COM2
- Puerto COM PAT: COM1
- Led IBC: LED2
- Led PAT: LED1
- Baud Rate IBC: 9600
- Baud Rate PAT: 9600

Configurar Tunel de Viento

- Mínimo punto: 1,0
- Maxímo punto: 25,0
- Cantidad de puntos: 5
- Puntos a calibrar: 5,10,15,20,25
- Tipo de medición: Asc - Des
- Tiempo de transitorio: 3
- Tiempo de estabilidad: 5
- Tiempo de medición: 2

Figura 6.11: En (a) y (b) se muestra las configuraciones realizada por el operador para el datalogger y el túnel de viento.

6.2.3. Resultados

Una vez finalizada la adquisición de datos, se observan en las Figuras 6.12a y 6.12b los datos crudos de intensidad y dirección del viento, respectivamente. De estos datos se extrae la parte plana de cada punto, correspondiente al tiempo de medición. En este caso, se toman muestras cada 1 segundo, y el tiempo de medición es de 2 minutos, equivalente a 120 segundos. Por lo tanto, se tienen 120 muestras por punto. Estas muestras se presentan en tablas, donde el operador debe verificar que en cada punto las mediciones no se salgan del rango establecido.

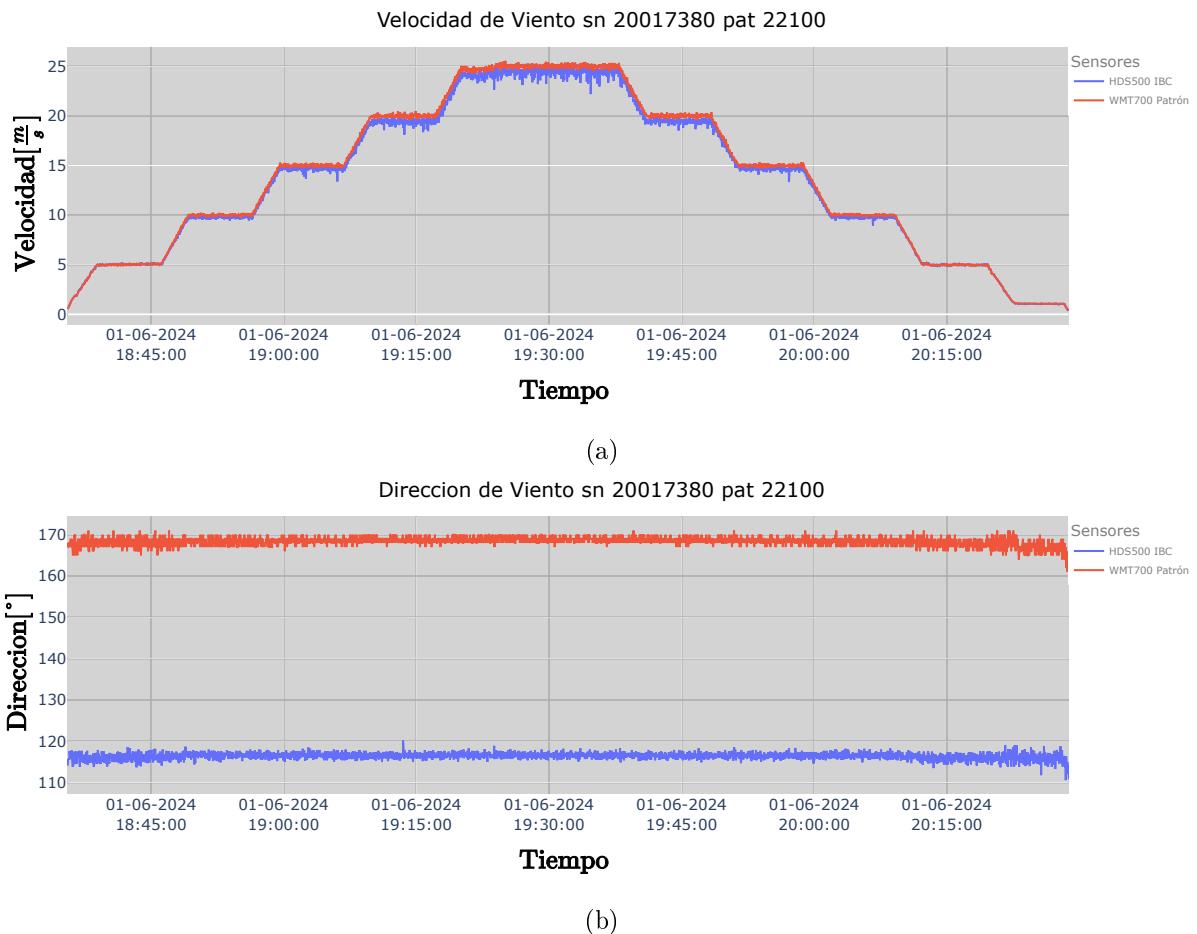


Figura 6.12: En (a) se muestra el ciclo ascendente y descendente de la velocidad de viento y en (b) se observa la dirección de viento.

A partir de la resta entre los valores promedio del ciclo ascendente y los valores promedio del ciclo descendente, se calcula la histéresis para cada punto. Tomando todos estos puntos, se puede construir una curva de histéresis como se muestra en la Figura 6.13a. En la misma se representan dos ajustes lineales: uno para el ciclo ascendente y otro para el ciclo descendente. Gráficamente, se puede observar que ambas curvas difieren sutilmente y presentan la misma pendiente, con una leve diferencia de 3 centésimos en la ordenada al origen. Además, presentan un factor de

correlación cercano a 1, lo que indica que el instrumento presenta baja alinealidad, de tal manera que cuando cambia de un valor en ascenso o descenso, lo hace por la misma curva. En la tabla de la Figura 6.13b se observa el resultado numérico de la histéresis para cada punto, tanto del patrón como del IBC.

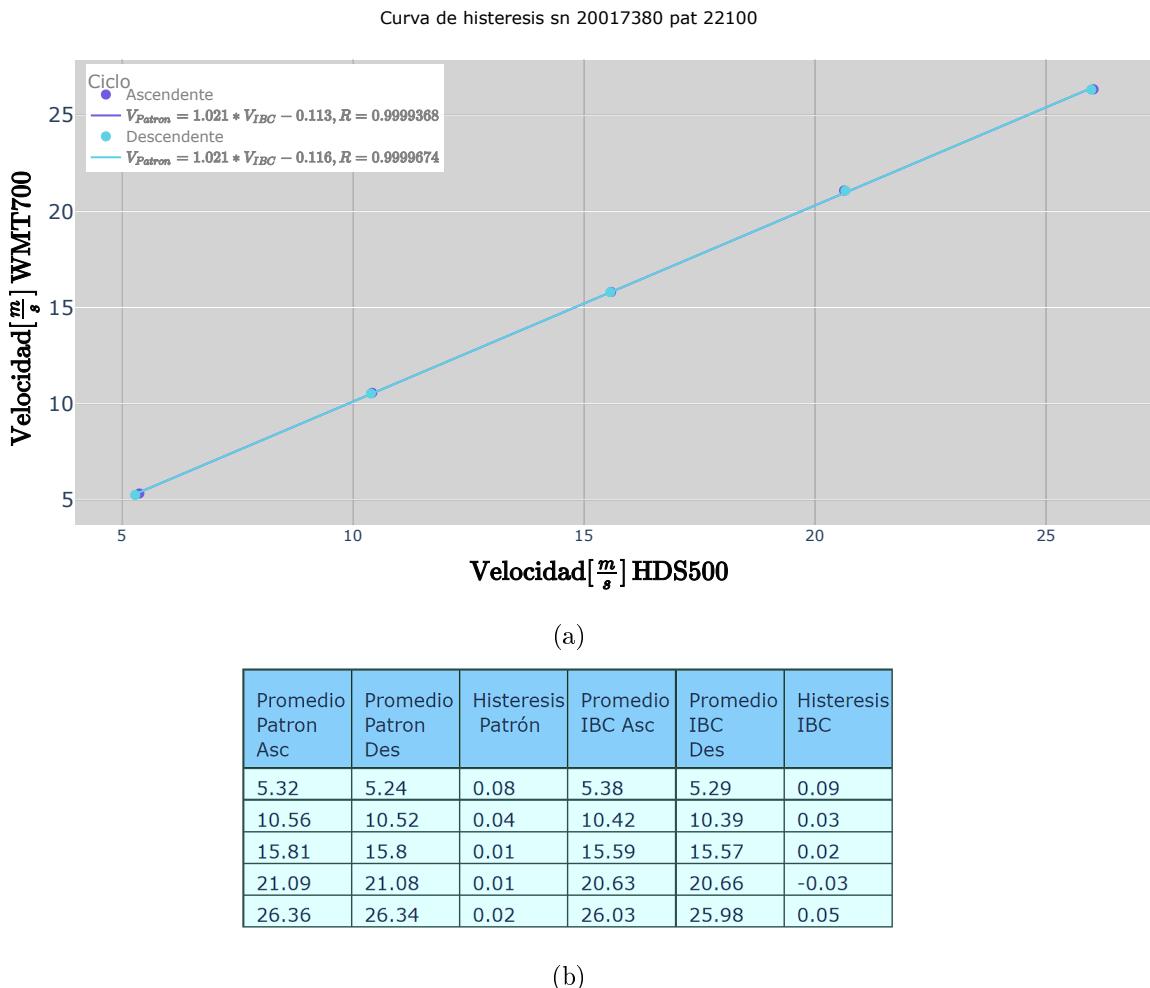


Figura 6.13: En (a) se muestra la curva de histéresis para el sensor bajo calibración y en (b) se observa la tabla con los resultados numéricos.

En la Figura 6.14a se presenta la curva de calibración, donde los datos del patrón han sido ajustados conforme a su certificado de calibración y comparados con los valores obtenidos mediante el IBC. Adicionalmente, en cada punto se incluye la incertidumbre expandida con un factor de cobertura aproximado de 2. De esta manera, se observa que el ajuste lineal de esta curva se encuentra dentro de los intervalos de incertidumbre, lo que permite afirmar que las mediciones verdaderas se encuentran en dicho intervalo con un nivel de confianza del 95 %. En la leyenda se muestra la función de calibración que se puede utilizar para aplicar las correcciones a las mediciones del IBC mediante software. En caso de que se deban ingresar manualmente las

correcciones para el sensor bajo calibración, se proporciona una tabla en la Figura 6.14b, que además cuenta con información de la incertidumbre combinada, la incertidumbre expandida y los grados de libertad. Toda esta información debe ser incluida en el certificado de calibración del IBC.

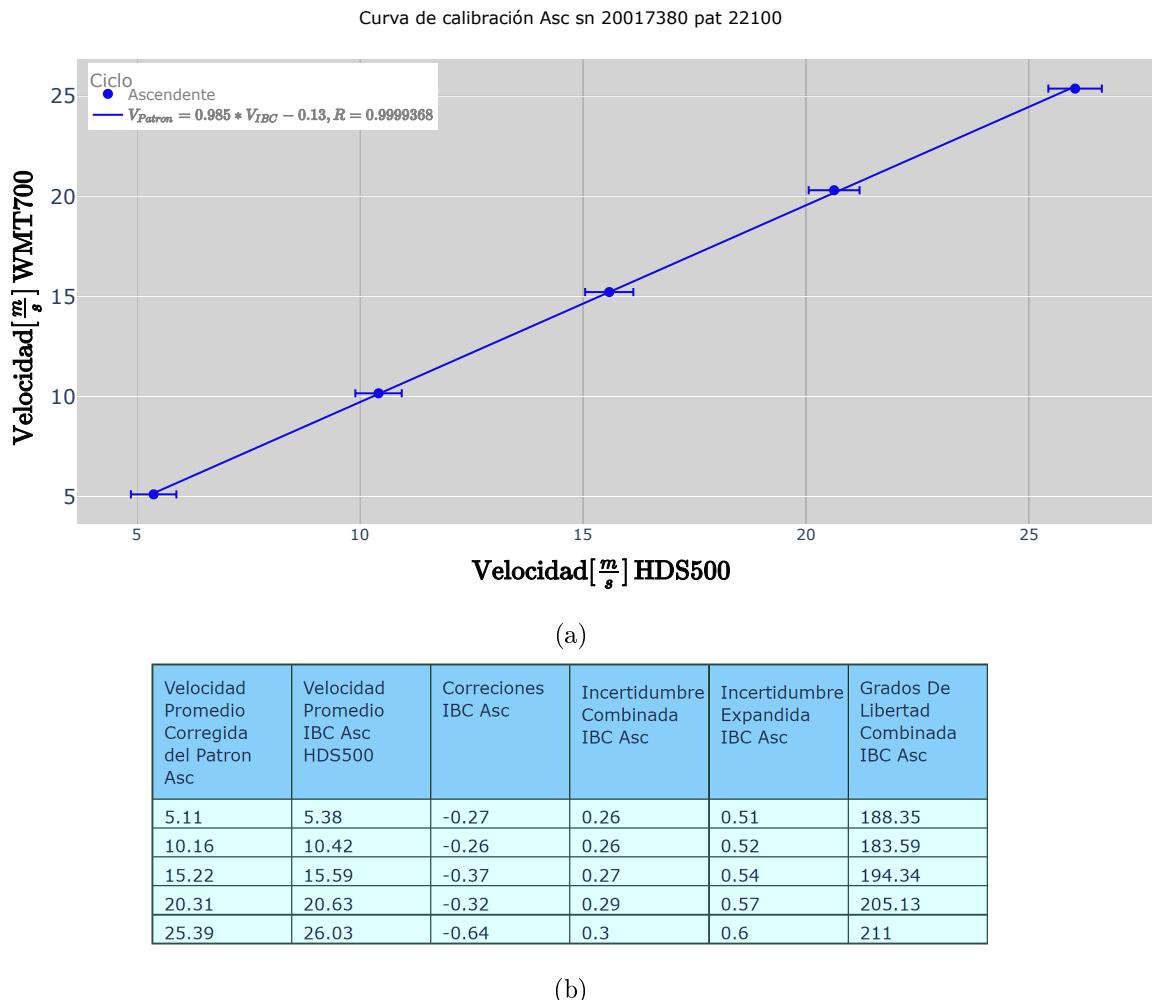


Figura 6.14: En (a) se presenta la curva de calibración ascendente con sus respectivas incertidumbres expandidas, y en (b) se muestra la tabla con los resultados numéricos.

Para las mediciones del ciclo descendente también se cuenta con una curva de calibración, la cual se puede utilizar para proporcionar información sobre la incertidumbre con la que se trabaja y aplicar las correcciones a las mediciones del instrumento. En la tabla de la Figura 6.15b también se muestran los resultados de esta calibración.

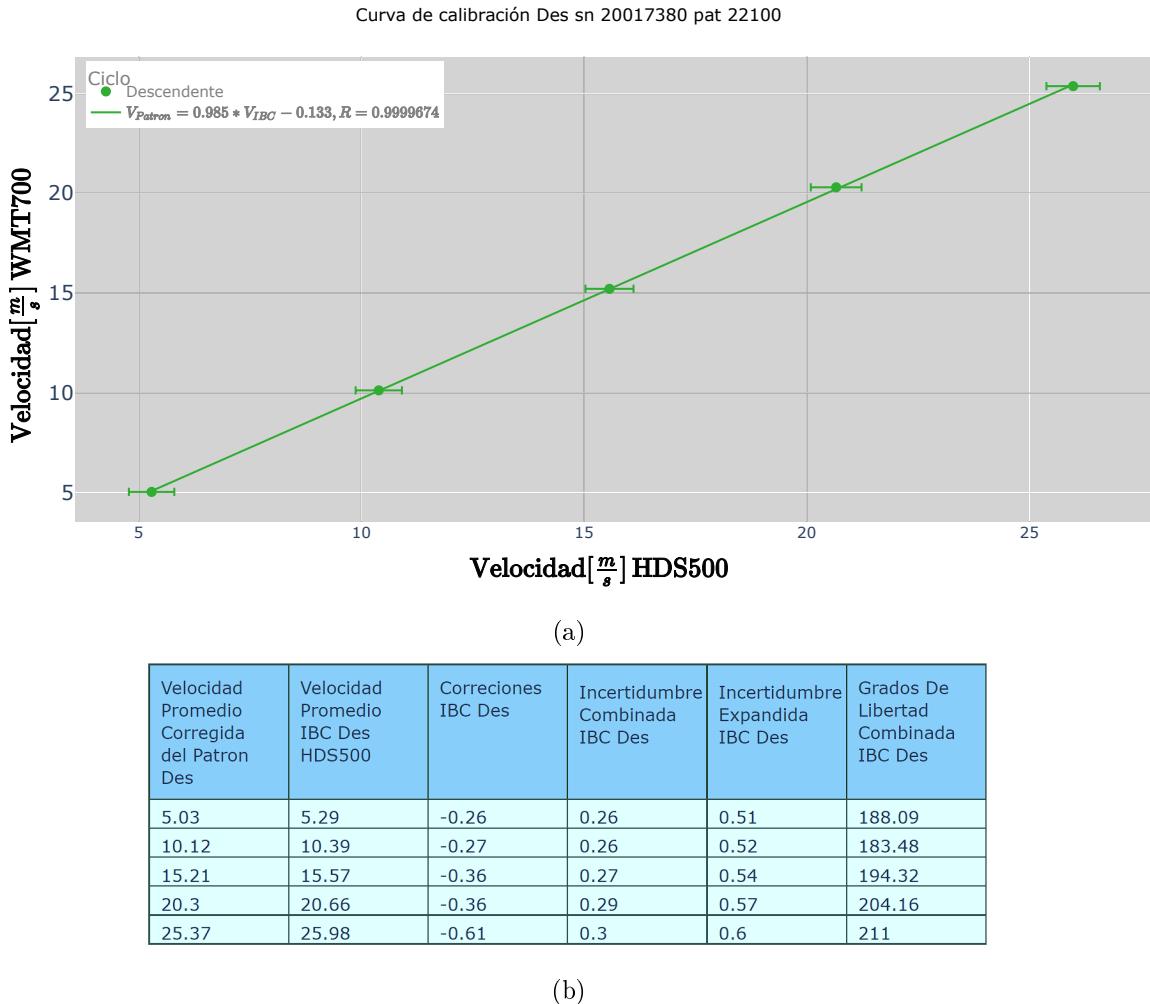


Figura 6.15: En (a) se presenta la curva de calibración descendente con sus respectivas incertidumbres expandidas, y en (b) se muestra la tabla con los resultados numéricos.

La información que se incluye en el certificado de calibración puede contener las dos curvas de calibración, tanto ascendente como descendente, o solo una de ellas. También es posible proporcionar únicamente alguna de las tablas de calibración, permitiendo que el usuario final realice una regresión lineal de los puntos y determine su propia curva de calibración. No obstante, el laboratorio no realiza el ajuste del instrumento, y la decisión final recae en el usuario final. En particular, los instrumentos que maneja el SMN tienen tolerancias máximas para cada magnitud física [6] establecidas por la WMO. En particular, para el viento se acepta una incertidumbre de velocidad de viento de $\pm 1 \text{ m s}^{-1}$ para ser considerado un instrumento de clase A, una incertidumbre de $\pm 2 \text{ m s}^{-1}$ para ser considerado de clase B, o una incertidumbre de $\pm 5 \text{ m s}^{-1}$ para ser considerado de clase C. Dependiendo de la categoría, el instrumento podrá ser utilizado para distintas aplicaciones.

Capítulo 7

Conclusiones

7.1. Conclusiones finales

El exhaustivo trabajo realizado en esta tesis ha permitido una comprensión profunda y detallada de diversos aspectos técnicos y científicos relacionados con la medición y calibración del viento. Desde el estudio del marco teórico de la metrología hasta la familiarización y entendimiento del uso del instrumental, se ha logrado establecer una base sólida para la investigación y desarrollo sobre estos temas.

Se abordaron los fundamentos del viento, el funcionamiento de los túneles de viento y los instrumentos utilizados para medir el flujo de aire. Además, se explicó el cálculo de la incertidumbre, la importancia de los certificados de calibración, los tipos de incertidumbre y la trazabilidad de las mediciones. Estos conocimientos fueron esenciales para garantizar la exactitud y confiabilidad de los resultados obtenidos.

La investigación incluyó el desarrollo de software, hardware y firmware necesarios para la aplicación, lo que permitió adquirir conocimientos detallados sobre el procedimiento de calibración de anemómetros en un túnel de viento. Este proceso no solo se comprendió teóricamente, sino que también se aplicó mediante el sistema de calibración automatizado, mejorando así la eficiencia en términos de tiempo y recursos humanos para realizar las mediciones y el procesamiento de datos.

El desarrollo de un datalogger para medir el viento con anemómetros y controlar la potencia del túnel de viento fue un componente clave de este trabajo. Este dispositivo permitió la recolección de datos precisos y su análisis en tiempo real, facilitando la calibración de los instrumentos de medición.

Asimismo, se implementó una aplicación web con una base de datos y una interfaz de usuario que permite configurar los equipos y cargar los metadatos. Esta herramienta proporciona una

plataforma accesible y eficiente para visualizar y gestionar el proceso de calibración.

Finalmente, utilizando el sistema desarrollado, se realizaron mediciones y se obtuvieron resultados experimentales que incluyeron la caracterización de la zona de medición y la calibración de un instrumento. Estos resultados demostraron la efectividad del sistema desarrollado y su capacidad para proporcionar mediciones precisas y confiables.

En resumen, este trabajo ha contribuido significativamente al avance del conocimiento en el campo de la metrología del viento y ha proporcionado herramientas prácticas y automatizadas para mejorar la precisión y eficiencia en la calibración de anemómetros.

7.2. Trabajo futuro

En cuanto al trabajo futuro, se podrá mejorar las prestaciones del sistema, expandiendo el hardware y el software para trabajar con otro tipo de sensores de viento. Ademas se podrá expandir su aplicación a otras variables meteorológicas con las que se trabaja en el SMN, como presión, temperatura, humedad y radiación dada la versatilidad y escalabilidad del hardware y software. También sera posible adaptar este sistema en otros túneles de viento.

Apéndice A

Datalogger y sensores

A.1. Principio de medición para sensores de ultrasonido

La velocidad y dirección del viento se determinan midiendo el tiempo que tardan los pulsos ultrasónicos en recorrer la distancia desde el transductor que genera el pulso hasta el transductor receptor. El instrumento utiliza dos pares de transductores orientados a lo largo de dos ejes ortogonales. Detectar la velocidad del viento a lo largo de dos ejes permite determinar no solo la intensidad sino también la dirección del viento.

El instrumento mide el tiempo de viaje del pulso ultrasónico entre los dos transductores de la misma pareja en ambas direcciones, como se muestra en la figura A.1. Los tiempos de viaje en las dos direcciones opuestas se definen como t_A (tiempo en dirección hacia adelante) y t_R (tiempo en dirección inversa). Si la velocidad del viento es cero, los valores de t_A y t_R son iguales. En presencia de viento, uno de los dos valores de tiempo es mayor que el otro y la comparación entre los dos valores de tiempo permite determinar la dirección y la intensidad del viento.

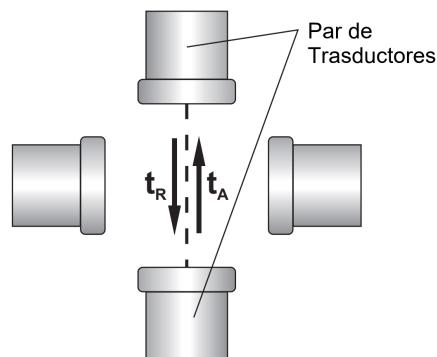


Figura A.1: Esquema de los transductores y el tiempo de viaje de un pulso ultrasónico. [8]

El tiempo de viaje del pulso de ultrasonido esta dado por las ecuaciones A.1 y A.2

$$t_A = \frac{D}{C + Vw} \quad (\text{A.1})$$

$$t_R = \frac{D}{C - Vw} \quad (\text{A.2})$$

donde se define:

- D: Distancia entre los dos transductores del mismo par.
- C: Velocidad del sonido.
- V_w : Componente de la velocidad del viento a lo largo del eje de medición.

Realizando la medición de los dos tiempos de viaje en los ejes cartesianos **U** y **V** podemos determinar las componentes de velocidad de viento como

$$V_{w_i} = \frac{D}{2} \left(\frac{1}{t_{a_i}} - \frac{1}{t_{R_i}} \right), \quad i = u, v \quad (\text{A.3})$$

El eje **U** es un eje que va de oeste a este, mientras que el eje **V** es un eje que va de sur a norte. Medir el tiempo de viaje en ambas direcciones también permite cancelar la dependencia de la velocidad de transmisión de los ultrasonidos en el aire de las condiciones ambientales de temperatura, humedad y presión barométrica.

A.2. Especificaciones técnicas y hojas de datos

Velocidad de viento	
Sensor	Ultrasonido
Rango de Medición	0 m s ⁻¹ a 75 m s ⁻¹
Resolución	0,1 m s ⁻¹
Precisión	±0,2 m s ⁻¹ o 2 % de la medición hasta 60 m s ⁻¹ 3 % de la medición para mayor que 60 m s ⁻¹
Dirección del viento	
Sensor	Ultrasonido
Rango de Medición	0° a 359,9°
Resolución	0,1°
Precisión	±2° RMSE (velocidad del viento 2 m s ⁻¹)
Fuente de alimentación	
Fuente de alimentación del calentador	24 V Vdc ±10 %
Consumo de energía del calentador	30 W
Fuente de alimentación del instrumento (sin calentador)	12 V a 30 V Vdc
Consumo de energía del instrumento (sin calentador)	60 mA @ 24 V Vdc
Otras especificaciones	
Salida Serial	RS232 aislado y RS485
Protocolo de comunicación	NMEA, MODBUS-RTU, ACII propietario
Intervalo de medición	Configurable de 250 ms a 1 s
Intervalo de velocidad promedio	Configurable de 1 s a 10 min
Conexión eléctrica	19-pole M23 conector Macho
Temperatura de operación	-40 °C a +70 °C
Grado de protección	IP66

Tabla A.1: Especificaciones de sensor de viento, marca Delta Ohm, modelo HD51.3.

Apéndice B

Software

B.1. Algoritmo del generador de trayectoria

```
1 def trajectoryGenerator(numCalib):
2     # Por ahora asumimos que el sistema empieza con una velocidad inicial del viento de
3     # 1 m/s que es lo que sucede generalmente al encender el tunel
4     vel_init = 1
5     vel_end = [1,1,1]
6     calibracion_instance = get_object_or_404(Calibracion, numeroDeCalibracion=numCalib)
7     datos_config_tunnel = ConfigTunel.objects.filter(calibracion=calibracion_instance)
8     puntosCalibrar = list(datos_config_tunnel)[0].puntosCalibrar
9     tipoDeMedicion = list(datos_config_tunnel)[0].tipoDeMedicion
10    if tipoDeMedicion == "Solo Ascendente":
11        velRef = puntosCalibrar.split(",")
12    if tipoDeMedicion == "Solo Descendente":
13        velRef = puntosCalibrar.split(",")
14        velRef = velRef[::-1]
15    if tipoDeMedicion == "Asc - Des":
16        velRefA = puntosCalibrar.split(",")
17        velRefB = velRefA[::-1]
18        velRef = velRefA + velRefB
19        pass
20
21    velRef = [int(x) for x in velRef]
22    aux = []
23    for i in range(len(velRef)):
24        aux.append([velRef[i]] * 3) # se multiplica por 3 ya que quiero que llegue,
25                                # estabilice y mida
26
27    velRef = sum(aux, []) # convertimos a un vector la matriz auxiliar
28    velRef = [vel_init] + velRef + vel_end
29    # Tiempos en minutos, ingresados por el usuario, en principio es el mismo para todos
30    # los puntos
31    tiempoTransitorio = list(datos_config_tunnel)[0].tiempoTransitorio*60
32    tiempoEstabilidad = list(datos_config_tunnel)[0].tiempoEstabilidad*60
33    tiempoMedicion = list(datos_config_tunnel)[0].tiempoMedicion*60
34    cantidadPuntos = list(datos_config_tunnel)[0].cantidadPuntos
35    if tipoDeMedicion == "Solo Ascendente" or tipoDeMedicion == "Solo Descendente" :
36        t_des = [tiempoTransitorio,tiempoEstabilidad, tiempoMedicion] * (cantidadPuntos +
37                           1)
38    else:
```

```

36     aux = [tiempoTransitorio , tiempoEstabilidad , tiempoMedicion]
37     t_des = (aux * (cantidadPuntos)*2 ) + aux
38
39
40 # Armamos la trayectoria / perfil de viento de referencia para el controlador
41 # trajectoryVector = [0]*1000
42 trajectoryVector = []
43 idx = 0
44 t_s = 1 # tiempo de muestreo en seg
45 t_acc = 10 # tiempo de aceleracion en seg
46 rapidez_max = 20 # pendiente maxima puede soportar el tunel, quiza tenga que ver con
    el saturador,255
47 qA = velRef[0] # punto de partida
48 for i in range(0,len(velRef)-1):
49     qB = velRef[i] # donde iba
50     qC = velRef[i+1] # donde voy
51     dA = qA-qB # de la zona 1
52     dC = qC-qB # de la zona 2
53     # Tj = max([dC/rapidez_max ,t_des[i]])
54     Tj = t_des[i]
55
56     for t_segm in range(-t_acc+t_s,t_acc,t_s): # zona 1
57         trajectoryVector.append((dC/Tj)*(((t_segm+t_acc)**2) /( 4*t_acc) ) + (dA/
            t_acc)*(((t_segm-t_acc)**2) / (4*t_acc)) + qB)
58
59     for t_segm in range(t_acc+t_s,Tj-t_acc+2,t_s): # zona 2
60         trajectoryVector.append( (dC/Tj)*t_segm + qB)
61     qA = trajectoryVector[idx-1]
62 # Redondeamos a 3 decimales los elementos de referencia
63 trajectoryVector = [round(numero , 3) for numero in trajectoryVector]
64 # Crear un arreglo de 1 a idx
65 t = np.arange(1, idx+1)
66 # Multiplicar cada elemento por t_sampling
67 t = t * t_s
68 # Graficamos la trayectoria armada
69 plt.plot(t/60,trajectoryVector )
70 plt.xlabel('t')
71 plt.ylabel('Vel Viento')
72 plt.title('Grafico de t vs Vel Viento')
73 plt.grid(True)
74 plt.show()
75 return trajectoryVector

```

Código B.1: Algoritmo generador de referencias para el controlador PID.

B.2. Comandos entre el servidor WebSocket y el datalogger

Comando	Descripción
<code>start</code>	El comando <code>start</code> inicia las mediciones del datalogger.
<code>stop</code>	El comando <code>stop</code> detiene las mediciones y la transmisión de datos del datalogger.
<code>setIBC;Uart_n;LED_n;BaudRate;sensorModel</code>	El comando <code>setIBC</code> configura: <ul style="list-style-type: none"> ■ El número de puerto COM. ■ El Led de recepción. ■ El baud Rate del sensor. ■ El modelo de sensor para llamar a la función que parsea los datos.
<code>setPAT;Uart_n;LED_n;BaudRate;sensorModel</code>	El comando <code>setPAT</code> configura: idem al comando anterior.
<code>setTimes;timeSampling;timeTable</code>	El comando <code>setTimes</code> configura: <ul style="list-style-type: none"> ■ El tiempo de muestreo. ■ El tiempo de tabla.
<code>refWindVel;valueWindRef</code>	El comando <code>refWindVel</code> envía <ul style="list-style-type: none"> ■ Un valor de velocidad de viento de referencia para el controlador.

Tabla B.1: Listado de comandos transmitidos por el servidor WebSocket al datalogger para configurar los parámetros seleccionados por el usuario.

Apéndice C

Cálculo de incertidumbre

En la tabla de la figura C.1 se muestra el cálculo, a través de una hoja de cálculo de Excel o una calculadora, para obtener la incertidumbre combinada en las mediciones de un anemómetro bajo calibración marca DeltaOhm, modelo HD51.3 en 10 m s^{-1} con un patrón Vaisala, modelo WMT700. Se cuantifican todas las incertidumbres, se normalizan por su factor de distribución, se multiplica por su coeficiente de sensibilidad que es la unidad en todos los casos y luego se realiza la suma de los cuadrados. En particular, se obtuvo una incertidumbre combinada de $u_{com} = \sqrt{0,07} = 0,26\text{ m s}^{-1}$ y, aplicando el factor de cobertura de 2, se obtiene una incertidumbre expandida de $U_{exp} = 2 \cdot 0,26 = 0,52\text{ m s}^{-1}$.

Por otro lado, se puede observar, en la última columna, el peso en porcentaje de la incertidumbre estándar al cuadrado para cada fuente. Se observa que las incertidumbre con mayor peso son el valor de calibración del patrón y el ajuste de la corrección de los datos del certificado del patrón. Por último, dado que el túnel no está caracterizado en su volumen, el aporte de las fuentes de incertidumbre se consideró nulo.

El software desarrollado realiza de forma iterativa estos cálculos para todos los puntos y devuelve un gráfico y una tabla con los resultados finales. De esta forma, el software permite optimizar en tiempo el procesamiento de los datos.

C.1. Ejemplo de Presupuesto de incertidumbre para un punto

Presupuesto de calibración para 10 m/s										
Fuentes de incertidumbre		Tipo de Incertidumbre	Valores $m s^{-1}$	Distribución	Factor de normalización	c_i	σ_i	$u_i = c_i \cdot \sigma_i$	u_i^2	Peso de u_i^2 (%)
Patron	Calibración	B	0,279	N	2,00	1,00	0,139	0,14	0,02	29,17
	Ajuste de calibración	A	0,049	N	1,00	1,00	0,049	0,05	0,00	3,64
	Resolución del instrumento	B	0,010	R	1,73	1,00	0,006	0,01	0,00	0,05
	Repetibilidad	A	0,010	N	9,80	1,00	0,010	0,01	0,00	0,15
	Histéresis	B	-0,020	R	1,73	1,00	- 0,012	-0,01	0,00	0,20
	Ajuste de la corrección del patron	A	0,210	N	1,00	1,00	0,210	0,21	0,04	66,13
	Factor de bloqueo	B	0,002	RMS	1,00	1,00	0,002	0,00	0,00	0,00
Túnel de viento	Homogeneidad	B	0,000	R	1,00	1,00	0,000	0,00	0,00	0,00
	Ajuste de la homogeneidad	A	0,000	N	1,00	1,00	0,000	0,00	0,00	0,00
	Estabilidad	B	0,000	R	1,73	1,00	0,000	0,00	0,00	0,00
	Ajuste de la estabilidad	A	0,000	N	1,00	1,00	0,000	0,00	0,00	0,00
	Factor de calibración	B	0,000	RMS	1,00	1,00	0,000	0,00	0,00	0,00
IBC	Resolución del Instrumento	B	0,010	R	1,73	1,00	0,006	0,01	0,00	0,05
	Repetibilidad	A	0,010	N	9,80	1,00	0,010	0,01	0,00	0,15
	Histéresis	B	-0,030	R	1,73	1,00	- 0,017	-0,02	0,00	0,45
	Factor de bloqueo	B	0,002	RMS	1,00	1,00	0,002	0,00	0,00	0,00
TOTAL								0,40	0,07	100,00

Tabla C.1: En la tabla se indica el calculo de incertidumbre para un punto en particular $10 m s^{-1}$.

Bibliografía

- [1] International Organization for Standardization. *ISO 16622:2002 Meteorology — Sonic anemometers/thermometers — Acceptance test methods for mean wind measurements.* 2002. URL: <https://www.iso.org/standard/29291.html>.
- [2] International Organization for Standardization. *ISO 17713-1:2007 Meteorology — Wind measurements — Part 1: Wind tunnel test methods for rotating anemometer performance.* 2007. URL: <https://www.iso.org/standard/31497.html>.
- [3] International Electrotechnical Commission. *IEC 61400-12-1:2017 Wind energy generation systems - Part 12-1: Power performance measurements of electricity producing wind turbines.* Annex F (normative): Wind tunnel calibration procedure for anemometers. 2017. URL: <https://webstore.iec.ch/publication/26603>.
- [4] International Organization for Standardization. *ISO/IEC 17025:2017 — General requirements for the competence of testing and calibration laboratories.* [Online; accessed 8-August-2024]. 2017. URL: <https://www.iso.org/standard/66912.html>.
- [5] World Meteorological Organization. «Surface Wind Measurement». En: *Guide to Meteorological Instruments and Methods of Observation.* WMO-No. 8. World Meteorological Organization, 2018, Chapter 5.
- [6] Infrastructure Commission for Observation e Information Systems. *APPENDIX 3. DECISIONS ADOPTED BY THE SESSION.* Interim Abridged Final Report of the First Session Virtual Session. WMO-No. 1251. Inclusion of the Measurement Quality Classification for Surface Observing Stations on Land in WMO-No. 8. Nov. de 2020.
- [7] Curso de Observaciones Meteorológicas - Centro regional de formación. *Parte 1 - Viento.* 2022. URL: <https://crf.smn.gob.ar/>.
- [8] Delta OHM. *DeltaOHM HD51.3D Series Operating Manual.* 2023. URL: https://www.deltaohm.com/wp-content/uploads/document/DeltaOHM_HD51.3D_manual_ENG.pdf.

- [9] Robótica (86.15), Facultad de Ingeniería, Departamento de Electrónica. *Teórica - Generación de Trayectoria*. <https://campusgrado.fi.uba.ar/enrol/index.php?id=384>. Notas de la materia sobre el método de Paul aplicado en generación de trayectorias. Jul. de 2023.
- [10] Embedded There. *RS485 Communication Protocol: Basics, Working Principle and Applications*. Accessed: July 23, 2024. 2023. URL: <https://embeddedthere.com/rs485-communication-protocol>.
- [11] Centre for Atmospheric Science, University of Manchester. *Sonic Anemometers*. [Online; accessed 8-August-2024]. 2024. URL: <http://www.cas.manchester.ac.uk/restools/instruments/meteorology/sonic/>.
- [12] Proyecto CIAA. *Ponchos de la CIAA - Desarrollo*. Sitio web. Accedido: 10 de julio de 2024, git clone <https://github.com/ciaa/Ponchos>. 2024. URL: <https://www.proyecto-ciaa.com.ar/devwiki/doku.php%3Fid=desarrollo:ciaa:ponchos.html>.
- [13] Dantec Dynamics. *Laser Doppler Anemometry (LDA)*. <https://www.dantecdynamics.com/solutions/fluid-mechanics/laser-doppler-anemometry-lda/>. Consultado el 1 de agosto de 2024. 2024.
- [14] EEVblog. *EEVblog #225 - Lab Power Supply Design Part 4 - PWM Control*. YouTube. Accedido: 4 de julio de 2024. 2024. URL: <https://www.youtube.com/watch?v=YaRDbw38x7Q>.
- [15] eFunda. *Introduction to Hot-Wire Anemometers*. [Online; accessed 8-August-2024]. 2024. URL: https://www.efunda.com/designstandards/sensors/hot_wires/hot_wires_intro.cfm.
- [16] Lufft. *VENTUS-UMB Ultrasonic Wind Sensor*. <https://www.lufft.com/products/wind-sensors-anemometers-289/ventus-umb-ultrasonic-wind-sensor-2303/>. Consultado el 1 de agosto de 2024. 2024.
- [17] Servicio Meteorológico Nacional. *Imagen extraída de Récords Antárticos y Cambio Climático*. Accedido: 31 de julio de 2024. 2024. URL: <https://www.smn.gob.ar/noticias/r%C3%A9cords-ant%C3%A1rticos-y-cambio-clim%C3%A1tico>.
- [18] Curso de observadores. *Imagen extraída del Curso de Observadores del Centro Regional de Formación del SMN, Parte 1: Unidad Viento*. Accedido: 31 de julio de 2024. 2024. URL: <https://crf.smn.gob.ar/>.
- [19] CIAA Project. *firmware_v3: Framework para desarrollo de Firmware de Sistemas Embedidos en C/C++*. GitHub. Accedido: 10 de julio de 2024. 2024. URL: https://github.com/ciaa/firmware_v3.

- [20] Proyecto CIAA. *Proyecto CIAA: Computadora Industrial Abierta Argentina.* <https://www.proyecto-ciaa.com.ar/>. Accedido: 1 de julio de 2024. 2024.
- [21] SIAP+MICROS. *Intensidad y Dirección del Viento.* <https://www.siapmicros.com/es/sistemi-di-misura/intensidad-y-direccion-del-viento/>. Consultado el 1 de agosto de 2024. 2024.
- [22] Testo. *Testo 425 – Anemómetros.* <https://www.testo.com/es-ES/products/flow-meter>. Consultado el 1 de agosto de 2024. 2024.
- [23] Windlogger. *History of the Anemometer.* [Online; accessed 8-August-2024]. 2024. URL: <https://www.windlogger.com/blogs/news/history-of-the-anemometer>.
- [24] WIZnet. *W5100 Datasheet.* Consultado el: 2 de Julio de 2024. 2024. URL: https://ar.mouser.com/datasheet/2/443/W5100_Datasheet_v1.2.5-586411.pdf.
- [25] World Meteorological Organization. *RIC Argentina — World Meteorological Organization.* [Online; accessed 8-August-2024]. 2024. URL: https://community.wmo.int/en/activity-areas/imop/Regional_Instrument_Centres/ric-argentina.
- [26] itemis CREATE: *Herramienta para desarrollar, simular y generar máquinas de estados finitos.* Sitio web de itemis. Accedido el 23 de julio de 2024. URL: https://www.itemis.com/en/products/itemis-create/documentation/user-guide/overview_what_are_itemis_create_statechart_tools.