

Proyecto y Diseño Final

Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería

Autor: Ricardo Ezequiel Lick.

Carrera: Ingeniería Electromecánica (Plan 2015).

Tutor: Ing. Diego Alberto Vicente.

Cotutor: Ing. Hugo Berti.

Jurado:

- Ing. Jorge Luis Amigone – Facultad de Ing. UNLPam.
- Ing. Gustavo Marcelo Flores – Facultad de Ing. UNLPam.
- Ing. Lucas Oscar Leguizamon – Facultad de Ing. UNLPam.

Fecha de aprobación: 15 de Agosto de 2023.

Lugar: General Pico, La Pampa, Argentina.

Resumen

Este trabajo presenta el diseño, fabricación y puesta en marcha del sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería de la Universidad Nacional de La Pampa. El sistema posee la capacidad de registrar la fuerza de sustentación, la fuerza de arrastre y el momento de cabeceo, que actúan sobre el objeto de estudio, en función de la velocidad del aire y del ángulo de ataque. Cuenta con una interfaz de usuario que permite visualizar los datos al pie del túnel de viento. Asimismo, posee un sitio web embebido capaz de proveer gráficas en tiempo real de las magnitudes aerodinámicas antes mencionadas. La interfaz de usuario permite, además, almacenar los datos en una memoria portátil para un posterior análisis de los mismos.

Palabras claves: balanza aerodinámica, instrumentación, túnel de viento, fuerzas aerodinámicas.

Abstract

This work presents the design, fabrication, and commissioning of the instrumentation and data acquisition system for the wind tunnel at the Faculty of Engineering of the National University of La Pampa. The system can measure lift force, drag force, and pitch moment acting on the object, as a function of air speed and angle of attack. It features a user interface located on the side of the wind tunnel for data visualization. Additionally, it includes an embedded website that provides real-time graphs of the aforementioned aerodynamic quantities. The user interface also allows for data storage on a portable memory device for later analysis.

Key Words: aerodynamic balance, instrumentation, wind tunnel, aerodynamic forces.

Índice

Capítulo 1: Introducción	1
1.1 Objetivos	2
1.2 Estructura del trabajo	2
Capítulo 2: Conceptos preliminares	4
2.1 Perfil aerodinámico	4
2.2 Fuerzas aerodinámicas	6
2.3 Túnel de viento.....	12
2.4 Balanza aerodinámica	13
2.4.1 Balanzas externas	14
2.4.2 Balanzas internas.....	17
Capítulo 3: Instrumentación del túnel de viento	19
3.1 Consideraciones de diseño	19
3.2 Medición de fuerzas	22
3.2.1 Selección del transductor de fuerza.....	22
3.2.2 Diseño de la balanza aerodinámica.....	27
3.3 Variación del ángulo de ataque	28
3.3.1 Selección del actuador.....	28
3.3.2 Diseño del mecanismo	29
3.3.3 Análisis de fuerzas en la balanza aerodinámica.....	30
3.4 Medición de la velocidad del aire	35
3.4.1 Selección del sensor de presión.....	36
Capítulo 4: Adquisición y presentación de datos.....	38
4.1 Selección de componentes	38
4.2 Diseño de la placa de circuito impreso	40
4.2.1 Diseño del diagrama esquemático.....	41

4.2.2 Diseño y ensamble de la placa	45
4.3 Interfaz de usuario	48
4.3.1 Panel principal.....	49
4.3.2 Panel web	50
Capítulo 5: Calibración y puesta en marcha	52
5.1 Calibración de la balanza aerodinámica.....	52
5.1.1 Resultados de la calibración.....	54
5.2 Verificación del sensor de velocidad del aire	59
5.3 Montaje del sistema en túnel de viento	61
5.4 Evaluación del funcionamiento del sistema.....	62
Capítulo 6: Conclusiones	65
Bibliografía	66
Anexos	68

Índice de figuras

Figura 1: Características geométricas de un perfil aerodinámico.	4
Figura 2: Geometría de los perfiles alares (a) NACA 0012 y (b) NACA 2412.	6
Figura 3: Ángulo de ataque de un perfil aerodinámico.	6
Figura 4: Perfil alar inmerso en flujo de aire: (a) campo de velocidades y (b) campo de presiones.	7
Figura 5: Fuerzas aerodinámicas en un perfil alar.	7
Figura 6: Coeficiente de sustentación en función del ángulo de ataque para los perfiles NACA 0012 y NACA 2412.	9
Figura 7: Coeficiente de arrastre en función del ángulo de ataque para los perfiles NACA 0012 y NACA 2412.	10
Figura 8: Momento de cabeceo en un perfil alar.	10
Figura 9: Coeficiente de momento de cabeceo en función del ángulo de ataque para los perfiles NACA 0012 y NACA 2412.	12
Figura 10: Componentes de un túnel de viento.	13
Figura 11: Balanza externa de cable.	14
Figura 12: Balanza externa de plataforma.	15
Figura 13: Balanza externa de yugo.	16
Figura 14: Balanza externa piramidal.	17
Figura 15: Balanza interna.	17
Figura 16: Sensor de fuerzas de una balanza interna.	18
Figura 17: Túnel de viento de la Facultad de Ingeniería.	19
Figura 18: Cámara de ensayo del túnel de viento de la Facultad de ingeniería.	20
Figura 19: Celda de carga de punto único.	23
Figura 20: Celda de carga de punto único, (a) esquema de montaje y funcionamiento, (b) esquema del puente de Wheatstone.	24
Figura 21: Análisis de sensibilidad a esfuerzos de la celda de carga de punto único.	24
Figura 22: Módulo Hx711.	25
Figura 23: Diagrama de bloques de una aplicación típica del CI Hx711.	26
Figura 24: Diseño de la balanza aerodinámica.	28
Figura 25: Servomotor SG92R.	29
Figura 26: Diseño del mecanismo de variación del ángulo de ataque.	30

Figura 27: Esquema simplificado de la balanza aerodinámica mediante 2 chapas.....	31
Figura 28: Diagrama de cuerpo libre de las chapas.	32
Figura 29: Análisis medición del momento de cabeceo en función del ángulo de ataque.	32
Figura 30: Análisis medición de fuerzas de sustentación y arrastre.	34
Figura 31: Tubo Pitot.	35
Figura 32: Sensor de presión MPX5004.	36
Figura 33: Tensión de salida del sensor MPX5004 en función de la presión aplicada...	37
Figura 34: Placa ESP32-DevKit.....	38
Figura 35: Módulo MSP1803, (a) pantalla y (b) ranura SD.	39
Figura 36: Encoder KY-040.....	40
Figura 37: Esquema de conexiones entre los componentes del sistema de instrumentación y el sistema de adquisición y presentación de datos.	41
Figura 38: Detalle del diagrama esquemático de la placa ESP32-DevKit.....	42
Figura 39: Detalle del diagrama esquemático del módulo Hx711.....	43
Figura 40: Detalle del diagrama esquemático del sensor MPX5004.	44
Figura 41: Detalle del diagrama esquemático de la fuente de alimentación.....	45
Figura 42: Cara frontal de la placa PCB con componentes soldados.	46
Figura 43: Cara posterior de la placa PCB con componentes soldados.	47
Figura 44: Ensamble de componentes sobre la tapa del gabinete.....	47
Figura 45: Gabinete ensamblado final.	47
Figura 46: Diagrama de flujo del proceso de ejecución del programa.	48
Figura 47: Vista de la pantalla del panel principal.....	49
Figura 48: Vista del panel web.....	51
Figura 49: Disposición de la balanza para calibración por fuerza de sustentación.....	53
Figura 50: Disposición de la balanza para calibración por fuerza de arrastre.	53
Figura 51: Disposición de la balanza para calibración por momento de cabeceo.	54
Figura 52: Respuesta de las salidas frente a fuerza de sustentación.	57
Figura 53: Respuesta de las salidas frente a fuerza de arrastre.	57
Figura 54: Respuesta de las salidas frente a momento de cabeceo para $\alpha=0^\circ$	58
Figura 55: Respuesta de las salidas frente a momento de cabeceo para $\alpha=20^\circ$	58
Figura 56: Transmisor Dwyer Magnesense.	60

Figura 57: Respuesta de la salida del sensor MPX5004 y del transmisor Dwyer a diferentes velocidades del sistema de ventilación.....	60
Figura 58: Sistema de instrumentación y adquisición de datos instalado en túnel de viento.	61
Figura 59: Vista frontal y lateral de la cámara de ensayo con el perfil de ala NACA 2412 en su interior.....	62
Figura 60: Perfiles alares utilizados en la evaluación del sistema (a) NACA 0012 y (b) NACA 2412.	62
Figura 61: Resultados de la evaluación del perfil alar NACA 0012.....	64
Figura 62: Resultados de la evaluación del perfil alar NACA 2412.....	64

Capítulo 1: Introducción

En el ámbito de la ingeniería aeroespacial y la aerodinámica, el túnel de viento es una herramienta fundamental para comprender y analizar el comportamiento del aire sobre la superficie de objetos en movimiento. El túnel de viento es una instalación diseñada para recrear condiciones atmosféricas controladas en un entorno cerrado y a escala reducida. Utiliza un ventilador potente y conductos especialmente diseñados para generar un flujo de aire constante y ajustable que simula diferentes velocidades del viento. El flujo se canaliza a través de una cámara de ensayo donde se colocan objetos a escala o prototipos para estudiar su comportamiento aerodinámico.

La principal tarea en un túnel de viento la constituye la medición de las fuerzas aerodinámicas, que permiten evaluar y mejorar el rendimiento del objeto en estudio: en aeronaves resulta importante el diseño de las alas para garantizar la sustentación necesaria; en vehículos terrestres resulta fundamental para mejorar la eficiencia del combustible; entre otros aspectos.

En este sentido, la Facultad de Ingeniería cuenta con un túnel de viento que está equipado con un kit académico. Entre los principales componentes de este kit se encuentran un manómetro de múltiples tubos y un perfil de ala con tomas de presión (orificios). Las tomas de presión están distribuidas en la superficie del perfil, y cada una está conectada a un tubo del manómetro mediante tubería flexible. Al exponer el perfil de ala a un flujo de aire, cada tubo del manómetro registra de manera visual la presión en un punto específico de la superficie del perfil.

Las pruebas realizadas con el kit académico son valiosas para comprender el origen de las fuerzas aerodinámicas; sin embargo, no proporcionan mediciones precisas de estas fuerzas. Además, el kit incluye solo un modelo de perfil de ala y su utilización conlleva dificultades. Estas dificultades suponen limitaciones para el estudio aerodinámico en el túnel de viento, y sugieren la necesidad de contar con equipamiento más preciso y versátil para ampliar las capacidades de investigación en este campo.

1.1 Objetivos

El objetivo principal de este proyecto es desarrollar un sistema de instrumentación y adquisición de datos digital para el túnel de viento de la Facultad de Ingeniería (UNLPam) y, por ende, facilitar la utilización del túnel de viento en actividades didácticas y experimentales.

Los objetivos específicos del proyecto y diseño son:

- Diseñar y construir una balanza aerodinámica para medir la fuerza de sustentación, la fuerza de arrastre y el momento de cabeceo que actúan sobre el objeto de estudio.
- Diseñar y construir un mecanismo para variar el ángulo de ataque del objeto de estudio.
- Incorporar un sensor para medir la velocidad del aire dentro del túnel de viento.
- Desarrollar y construir un sistema para la adquisición de la información proveniente de los sensores, la visualización y el almacenamiento de la información procesada.

1.2 Estructura del trabajo

El capítulo 2 presenta conceptos preliminares que facilitan la comprensión de aspectos que se abordan en el desarrollo del proyecto. El capítulo comienza con una definición del término perfil de ala en el ámbito de la aeronáutica y luego se presentan los principales parámetros que definen la geometría del perfil. A continuación se describe el origen de las fuerzas aerodinámicas sobre los mismos. Luego, se describe en forma breve el túnel de viento, y por último, se presenta una descripción de los diseños más destacados de balanzas utilizadas para medir las fuerzas aerodinámicas en túnel de viento.

El capítulo 3 presenta el desarrollo del sistema de instrumentación, específicamente el desarrollo de la balanza de fuerzas, el diseño del mecanismo para variar el ángulo de ataque del perfil de ala que se estudia y la selección del sensor de velocidad de flujo. El capítulo comienza con un análisis del túnel de viento para identificar y determinar los parámetros de diseño de la balanza de fuerzas. Luego, se abordan el diseño de la balanza aerodinámica y el mecanismo para variar el ángulo de ataque. El capítulo concluye con

la selección del sensor de presión diferencial que se utiliza para medir la velocidad del aire.

El capítulo 4 describe el desarrollo del sistema de adquisición y presentación de datos. El capítulo comienza con la selección de la unidad microcontrolada y los componentes para el desarrollo de la interfaz con el usuario. Luego, se presenta el diseño del diagrama esquemático de la placa de circuito impreso, haciendo hincapié en algunos detalles de diseño y adaptación de señales. Por último, se presenta la lógica del programa que se ejecuta en la unidad microcontrolada y se describe el funcionamiento de la interfaz para el usuario.

El capítulo 5 presenta los resultados obtenidos durante la calibración de los instrumentos, el montaje del sistema en el túnel de viento y la evaluación experimental del mismo. El capítulo comienza con los resultados obtenidos en la calibración de la balanza aerodinámica. Se obtienen los coeficientes de sensibilidad de la balanza respecto de las fuerzas aerodinámicas y los coeficientes de sensibilidad cruzada. Luego, se contrasta el sensor de velocidad del aire desarrollado con un transmisor comercial de la marca Dwyer. Por último, se presentan los resultados experimentales de los ensayos con perfiles de alas NACA 0012 y NACA 2412.

Para finalizar, en el capítulo 6 se presentan las principales conclusiones obtenidas a lo largo del trabajo.

Capítulo 2: Conceptos preliminares

El propósito de este capítulo es comprender el origen de las fuerzas aerodinámicas y el modo en que estas fuerzas se miden en el túnel de viento mediante la utilización de la balanza aerodinámica. El capítulo comienza con un estudio de perfiles aerodinámicos y de las fuerzas que aparecen cuando estos perfiles se mueven en relación con un fluido circundante gaseoso. Luego, se exploran varios aspectos teóricos del túnel de viento, que incluyen su definición, modo de operación, aplicaciones y componentes. Por último, se realiza un análisis de los diseños de balanzas aerodinámicas más destacadas.

2.1 Perfil aerodinámico

En el ámbito de la aeronáutica el término perfil aerodinámico o perfil alar se refiere a la geometría de la sección transversal de un objeto que genera fuerzas aerodinámicas cuando se somete al movimiento relativo con un fluido gaseoso [1]. Entre los principales ejemplos de estos objetos se encuentran: ala de avión, pala de hélice, pala de helicóptero y álabe de turbina. La Figura 1 ilustra un esquema de perfil alar e identifica las principales características geométricas.

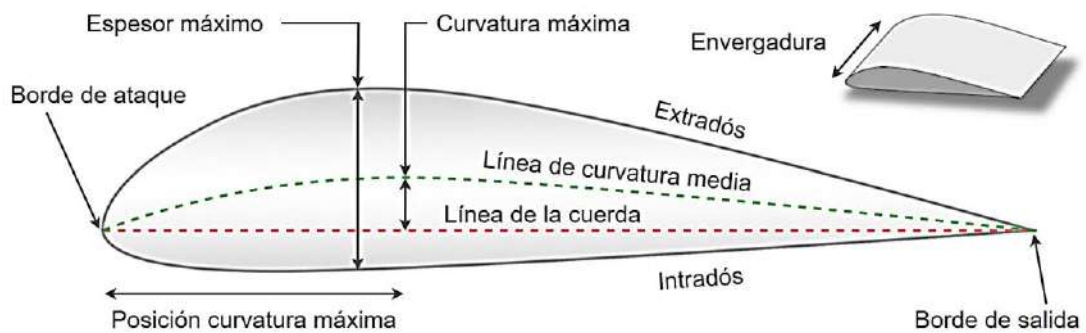


Figura 1: Características geométricas de un perfil aerodinámico.

Donde:

- *Borde de ataque*, punto más adelantado del perfil.
- *Borde de salida*, punto más retrasado del perfil.
- *Extradós*, parte superior del perfil.
- *Intradós*, parte inferior del perfil.

- *Línea de la cuerda*, línea recta que une el borde de ataque con el borde de salida.
- *Línea de curvatura media*, línea formada por los puntos equidistantes entre intradós y extradós.
- *Espesor máximo*, máxima distancia entre intradós y extradós, medida en forma perpendicular a la cuerda.
- *Curvatura máxima*, máxima distancia entre la línea de curvatura media y la línea de la cuerda, medida en forma perpendicular a la cuerda.
- *Posición curvatura máxima*, distancia entre el borde de ataque y la curvatura máxima.
- *Envergadura*, distancia entre los extremos laterales del ala.

La elección de cada uno de estos parámetros implica las consideraciones más importantes del diseño de un perfil alar. En particular, la línea de curvatura media permite definir la simetría del perfil; si la línea de curvatura media coincide con la línea de la cuerda define un perfil alar simétrico. Por ejemplo, el perfil de la Figura 1 es asimétrico, ya que la línea de curvatura media no coincide con la línea de la cuerda y, además, posee una línea de curvatura media positiva, es decir, cóncava hacia abajo. El diseño de perfiles tiene diferentes objetivos, por ende, estas geometrías pueden variar en cuanto al espesor, la simetría e incluso variar a lo largo de la envergadura.

Series NACA

Con el propósito de establecer una nomenclatura estandarizada para los perfiles alares y simplificar la identificación de cada diseño se han creado varias series. Entre ellas, las series NACA (National Advisory Committee for Aeronautics) son las más utilizadas en la actualidad y fueron pioneras en el desarrollo sistemático de perfiles alares mediante el empleo de ecuaciones analíticas [1].

A lo largo de este trabajo se utilizan los perfiles alares NACA 0012 y NACA 2412, resulta necesario identificar su nomenclatura y definir su geometría. Cada cifra detrás del sobrenombre NACA contiene los datos necesarios para construir el perfil, por ejemplo NACA *MPXX*:

- *M* expresa el valor de la curvatura máxima en porcentaje de la longitud de la cuerda.

- P expresa la posición de la curvatura máxima en décimas de la longitud de la cuerda medido desde el borde de ataque.
- XX indica el espesor máximo del perfil en porcentaje de la longitud de la cuerda.

La Figura 2 muestra los perfiles alares (a) NACA 0012 y (b) NACA 2412. El perfil NACA 0012 posee $M=P=0$, lo que indica que su curvatura es nula y, por ende, se define como un perfil simétrico. Por otro lado, el perfil NACA 2412 tiene $M=2$ y $P=4$, lo que significa que posee una curvatura máxima del 2% ubicada a una distancia del 40% de la longitud de la cuerda, medida desde el borde de ataque. Esta característica define al NACA 2412 como un perfil asimétrico. Por último, dado que en ambos perfiles $XX=12$, entonces resulta que sus geometrías tienen un espesor máximo del 12% de la longitud de la cuerda.

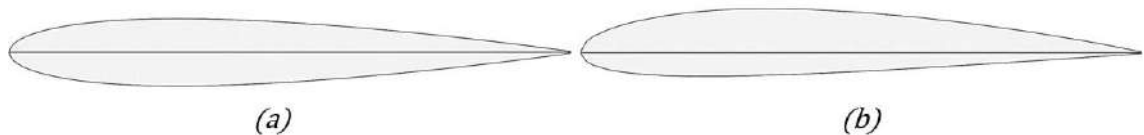


Figura 2: Geometría de los perfiles alares (a) NACA 0012 y (b) NACA 2412.

2.2 Fuerzas aerodinámicas

En el estudio de las fuerzas aerodinámicas se considera un perfil asimétrico, con curvatura positiva, sometido a un flujo de aire con un ángulo de ataque α , siendo α el ángulo entre la dirección del flujo de aire y la línea de la cuerda del perfil (Figura 3).

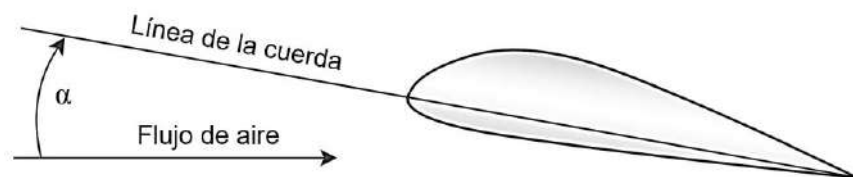


Figura 3: Ángulo de ataque de un perfil aerodinámico.

En estas condiciones, el flujo de aire que pasa sobre el extradós (parte superior del perfil) se mueve a mayor velocidad y la presión disminuye, mientras que, sobre el intradós (parte inferior del perfil), el flujo de aire se mueve a menor velocidad y la presión aumenta [1], como se muestra en la Figura 4.

Para explicar la relación entre el campo de velocidades y el de presiones es necesario recurrir al Teorema de Bernoulli, que deriva de la ley de conservación de energía de Mecánica de fluidos [2]. Este teorema demuestra que para un fluido ideal en condición subsónica, cuando el flujo de aire se acelera la presión disminuye, y por el contrario, cuando se desacelera la presión aumenta [1].

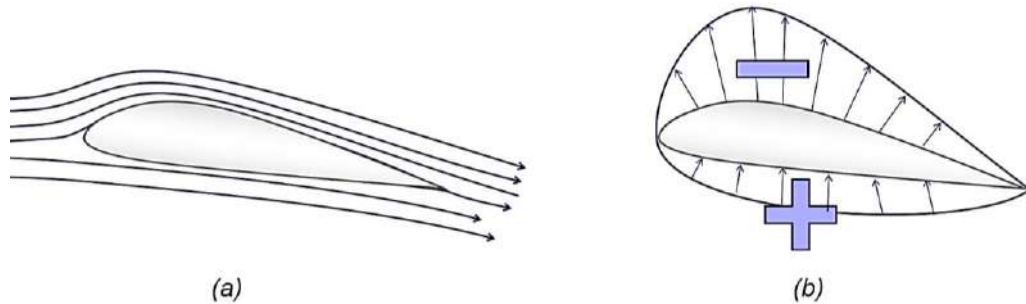


Figura 4: Perfil alar inmerso en flujo de aire: (a) campo de velocidades y (b) campo de presiones.

La suma de todas las fuerzas de presión individuales que actúan sobre el perfil aerodinámico deriva en una única fuerza resultante con un efecto equivalente. Como se muestra en la Figura 5, esta fuerza resultante se divide siempre en dos componentes: una componente perpendicular al flujo de aire denominada fuerza de sustentación y una componente paralela al flujo de aire denominada fuerza de arrastre [1]. El punto sobre la línea de la cuerda donde la fuerza resultante actúa se denomina centro de presiones (CP), y su posición suele estar comprendido entre el 25% y el 60% de la longitud de la cuerda, medido desde el borde de ataque [3].

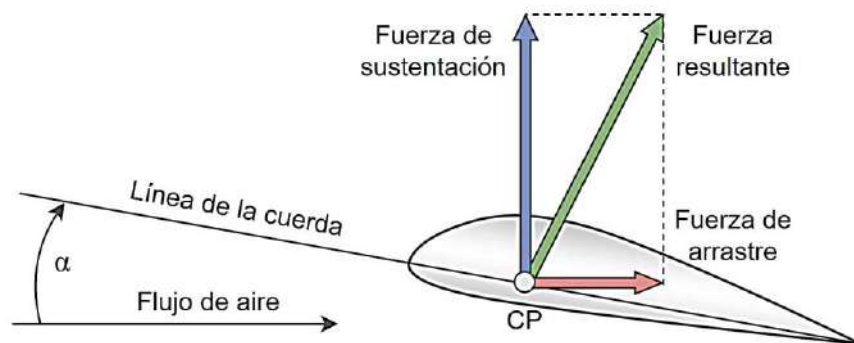


Figura 5: Fuerzas aerodinámicas en un perfil alar.

Fuerza y coeficiente de sustentación

La fuerza de sustentación, representada por F_L , es la componente perpendicular al flujo de aire, que actúa de abajo hacia arriba. Su origen depende exclusivamente de la diferencia de presiones entre extradós e intradós [1]. La expresión matemática para la fuerza de sustentación es la siguiente:

$$F_L = \frac{1}{2} \cdot \rho \cdot v^2 \cdot A \cdot C_L \quad (1)$$

Dónde F_L es la fuerza de sustentación [N], ρ es la densidad del fluido [kg/m^3], v es la velocidad del flujo [m/s], A es la superficie del ala [m^2] y C_L es el coeficiente de sustentación.

El coeficiente de sustentación C_L es una medida adimensional que describe la capacidad de un perfil para generar sustentación. En términos más simples, el coeficiente C_L indica cuánta sustentación puede generar un perfil alar en relación a su área y a la velocidad del flujo de aire que lo atraviesa. Su valor depende de la forma del perfil y del ángulo de ataque α .

En la Figura 6 se presenta el coeficiente de sustentación (C_L) en función del ángulo de ataque (α) para los perfiles NACA 0012 y NACA 2412. La figura muestra que a medida que aumenta el ángulo de ataque, el coeficiente C_L se incrementa hasta alcanzar su máximo, después del cual, la sustentación disminuye debido a la separación del flujo de aire. Los valores positivos del coeficiente C_L indican que el perfil es empujado hacia arriba, mientras que los valores negativos señalan que el perfil es empujado hacia abajo. Asimismo, se aprecia que en un perfil simétrico (NACA 0012) el coeficiente C_L resulta nulo cuando el ángulo de ataque es nulo, en cambio, en un perfil asimétrico (NACA 2412) el coeficiente C_L es distinto de cero cuando el ángulo de ataque es nulo. La información sobre estos perfiles está disponible en el sitio web de Airfoil Tools [4].

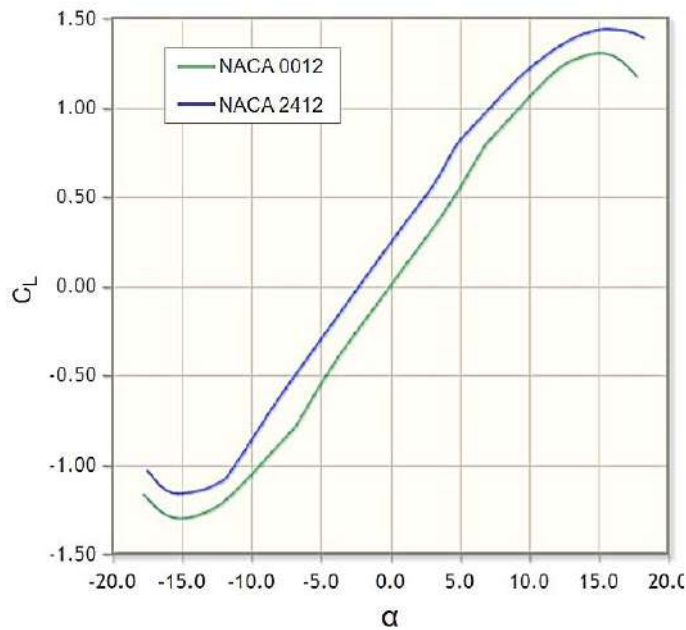


Figura 6: Coeficiente de sustentación en función del ángulo de ataque para los perfiles NACA 0012 y NACA 2412.

Fuerza y coeficiente de arrastre

La fuerza de arrastre, representada por F_D , es la fuerza que se opone al avance del perfil alar. Esta fuerza actúa en forma paralela y en la misma dirección que el flujo de aire relativo incidente. A diferencia de la fuerza de sustentación, la fuerza de arrastre no solo depende de la diferencia de presiones entre extradós e intradós, sino que aparece una componente más que deriva de la fricción del flujo de aire sobre el perfil [1]. La expresión matemática para la fuerza de arrastre es la siguiente:

$$F_D = \frac{1}{2} \cdot \rho \cdot v^2 \cdot A \cdot C_D \quad (2)$$

Donde F_D es la fuerza de arrastre [N], ρ es la densidad del fluido [kg/m^3], v es la velocidad del flujo [m/s], A es la superficie del ala [m^2] y C_D es el coeficiente de arrastre.

De manera análoga al coeficiente de sustentación C_L , el coeficiente de arrastre C_D es una medida adimensional, que indica cuánto arrastre puede generar el perfil de ala en relación a su área y la velocidad del aire que lo atraviesa. Como sucede con el coeficiente C_L , el coeficiente C_D depende de la forma del perfil y del ángulo de ataque α .

En la Figura 7 se presenta el coeficiente de arrastre (C_D) en función del ángulo de ataque (α) para los perfiles NACA 0012 y NACA 2412. La figura muestra que a medida que

aumenta el ángulo de ataque, el coeficiente C_D aumenta gradualmente hasta un cierto punto, a partir del cual, experimenta un aumento más pronunciado. Asimismo, se aprecia que no hay diferencias significativas en los valores del coeficiente C_D entre el perfil simétrico NACA 0012 y el perfil asimétrico NACA 2412. La información sobre estos perfiles está disponible en el sitio web de Airfoil Tools [4].

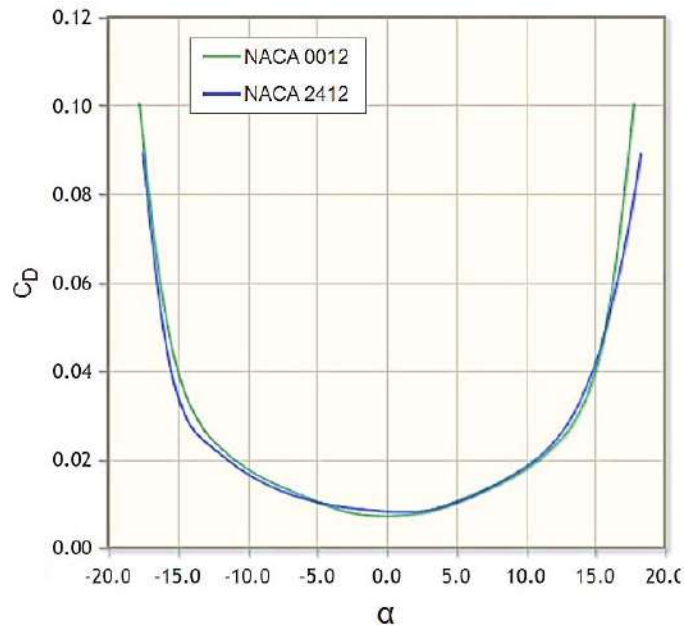


Figura 7: Coeficiente de arrastre en función del ángulo de ataque para los perfiles NACA 0012 y NACA 2412.

Momento de cabeceo

El momento de cabeceo se refiere a la tendencia que presenta el perfil para girar alrededor de un punto. En la Figura 8 se muestra un perfil alar con dos puntos identificados: el centro de presiones (CP) y el centro aerodinámico (CA). El CP es el punto donde actúa la fuerza resultante, lo que significa que el momento producido por esta fuerza respecto del CP es nulo.

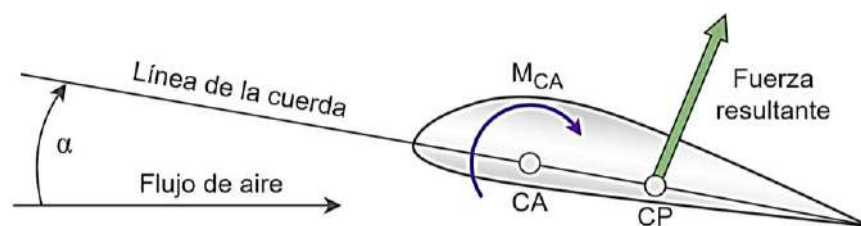


Figura 8: Momento de cabeceo en un perfil alar.

El centro aerodinámico CA es un punto de interés para el análisis de las fuerzas aerodinámicas en el cual se puede considerar aplicado un sistema de fuerza–momento [3]. Para un perfil inmerso en un flujo de aire subsónico, el CA se encuentra situado sobre la cuerda del perfil aproximadamente a 25% de la longitud de la cuerda medido desde el borde de ataque [3]. Al variar el ángulo de ataque, el CP se desplaza sobre la línea de la cuerda, típicamente en un rango entre el 25 y el 60% de la longitud de la cuerda medido desde el borde de ataque [3]. Este desplazamiento del CP permite que la fuerza resultante genere un momento alrededor del CA. La expresión matemática para el momento de cabeceo respecto del centro aerodinámico es la siguiente:

$$M_{CA} = \frac{1}{2} \cdot \rho \cdot v^2 \cdot A \cdot C \cdot C_M \quad (3)$$

Donde M_{CA} es el momento de cabeceo respecto del centro aerodinámico [$N \cdot m$], ρ es la densidad del fluido [kg/m^3], v es la velocidad del flujo [m/s], A es la superficie del ala [m^2], C es la longitud de la cuerda [m] y C_M es el coeficiente de momento de cabeceo respecto del centro aerodinámico.

El coeficiente de momento de cabeceo C_M es una medida adimensional, que indica cuánto momento puede generar el perfil de ala en relación a su área, a la cuerda y a la velocidad del aire que lo atraviesa. El coeficiente C_M depende principalmente de la forma del perfil.

En la Figura 9 se presenta el coeficiente de momento de cabeceo (C_M) en función del ángulo de ataque (α) para los perfiles NACA 0012 y NACA 2412. La figura muestra que en un perfil simétrico (NACA 0012) el coeficiente C_M resulta nulo cuando el ángulo de ataque es nulo, en cambio, en un perfil asimétrico (NACA 2412) el coeficiente C_M es distinto de cero cuando el ángulo de ataque es nulo.

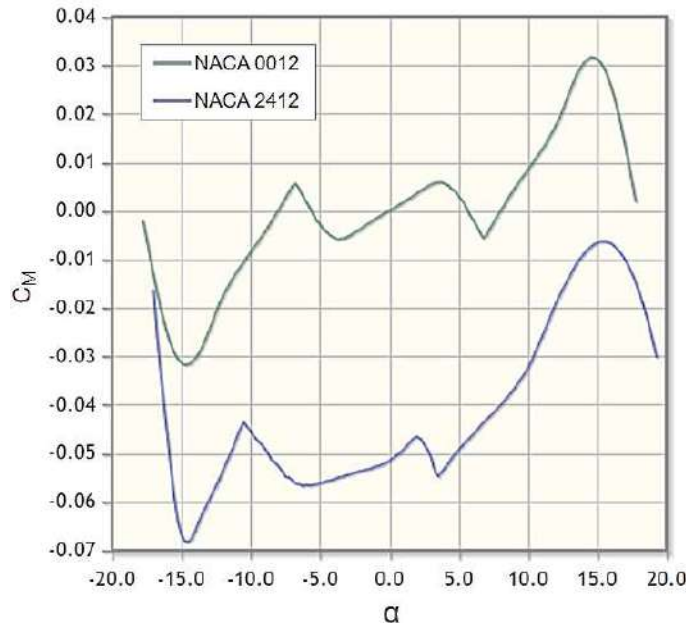


Figura 9: Coeficiente de momento de cabeceo en función del ángulo de ataque para los perfiles NACA 0012 y NACA 2412.

2.3 Túnel de viento

El túnel de viento es una herramienta de investigación que permite simular el desplazamiento de un objeto a través de un fluido gaseoso, lo que facilita la medición de las fuerzas aerodinámicas que actúan sobre el objeto de estudio. El túnel de viento está conformado por un conducto por el cual se impulsa un flujo de aire uniforme y rectilíneo a una velocidad predefinida, mientras el objeto de análisis permanece estacionario. La principal aplicación del túnel de viento radica en realizar pruebas a objetos aerodinámicos (ala de avión, álabes de turbina, etc.) a una escala reducida. Estas pruebas proporcionan datos que permiten comprender y mejorar el rendimiento antes de proceder con la producción o construcción a escala real del objeto. Esta metodología contribuye a evitar situaciones peligrosas en ciertas circunstancias de funcionamiento del objeto de estudio, ahorrando grandes costos.

La Figura 10 muestra las distintas partes de un túnel de viento típico: cono de aceleración, cámara de ensayo, difusor y ventilador. Una prueba en el túnel de viento se inicia con la ubicación del objeto de estudio dentro de la cámara de ensayo, sostenido por un soporte que está vinculado a los sensores de fuerza. Luego se establece la velocidad del flujo de aire mediante el control del ventilador. El flujo de aire ingresa por el cono de aceleración, diseñado para aumentar la velocidad del aire y concentrar el flujo hacia la entrada de la

cámara de ensayo. En el interior de esta cámara el objeto interactúa con el flujo de aire y se recopilan los datos de los sensores. Por último, el aire sale por el difusor, que reduce su velocidad.

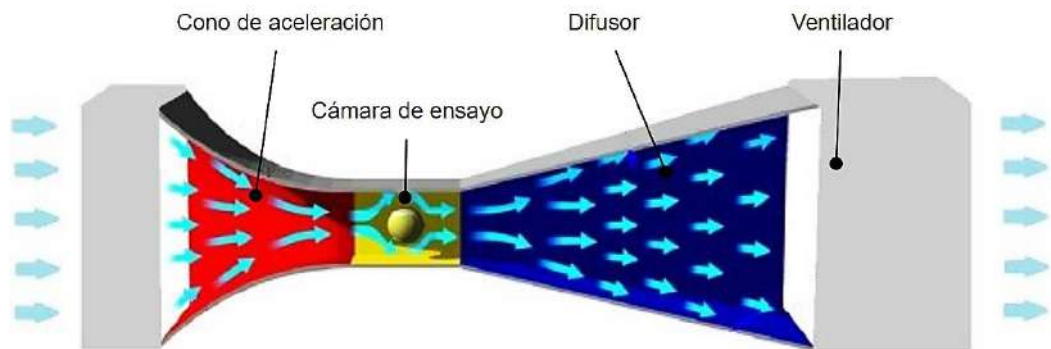


Figura 10: Componentes de un túnel de viento.

Los datos recolectados de los sensores ofrecen información para predecir el comportamiento que tendría el objeto en su escala real (prototipo), bajo condiciones semejantes a las que fue sometido en el túnel de viento. Una de las condiciones de semejanza es tener tanto en el objeto de estudio como en el prototipo un mismo número de Reynolds [5]. El número de Reynolds se calcula a partir de la siguiente fórmula:

$$Re = \frac{\rho \cdot v \cdot C}{\mu} \quad (4)$$

Donde Re es el número de Reynolds, ρ es la densidad del fluido [kg/m^3], v es la velocidad del flujo [m/s], C es la longitud de la cuerda del perfil de ala [m] y μ es la viscosidad dinámica del fluido [$kg/m \cdot s$].

2.4 Balanza aerodinámica

La balanza aerodinámica es el instrumento de precisión que se incorpora al túnel de viento para sostener el objeto de estudio dentro de la cámara de ensayo y medir las fuerzas aerodinámicas que actúan sobre el objeto cuando es expuesto al flujo de aire. Las balanzas aerodinámicas se dividen en dos categorías principales: externas e internas, cada una con sus aplicaciones y ventajas específicas.

2.4.1 Balanzas externas

Estas balanzas se ubican fuera del objeto de estudio y emplean varios sistemas mecánicos que permiten desacoplar las fuerzas de arrastre, de sustentación y el momento de cabeceo, lo que facilita la medición directa de cada una de estas fuerzas. El objeto se coloca dentro de la cámara de ensayo del túnel mediante soportes que lo vinculan a la balanza.

La ventaja de estas balanzas radica en su versatilidad, ya que se pueden utilizar para una variedad de estudios con diferentes objetos, simplemente cambiando el tipo de soporte. Además, esta disposición facilita reparaciones y modificaciones de la balanza por situarse separada del objeto.

Sin embargo, estas balanzas también tienen sus desventajas. En muchos casos, interfieren en el estudio al desviar el flujo de aire en el espacio que ocupan. Además, suelen tener un peso relativo elevado debido a la inercia de los sistemas de medición, lo que limita su uso en túneles pequeños. Entre los tipos de balanzas externas destacadas se encuentran:

Balanza de cable: es uno de los primeros tipos de balanzas utilizadas en túneles de viento. Este tipo de balanza consiste en sostener el objeto de estudio con cables y mediante la tensión generada en los cables se determinan las fuerzas aerodinámicas. En la Figura 11 se presenta un esquema de la balanza de cable y una estimación de los resultados de las fuerzas aerodinámicas que se pueden obtener con este método. La figura muestra el objeto sostenido por cables verticales y horizontales, entre los que se ubican los sensores para medir las fuerzas aerodinámicas que se desarrollan sobre el objeto.

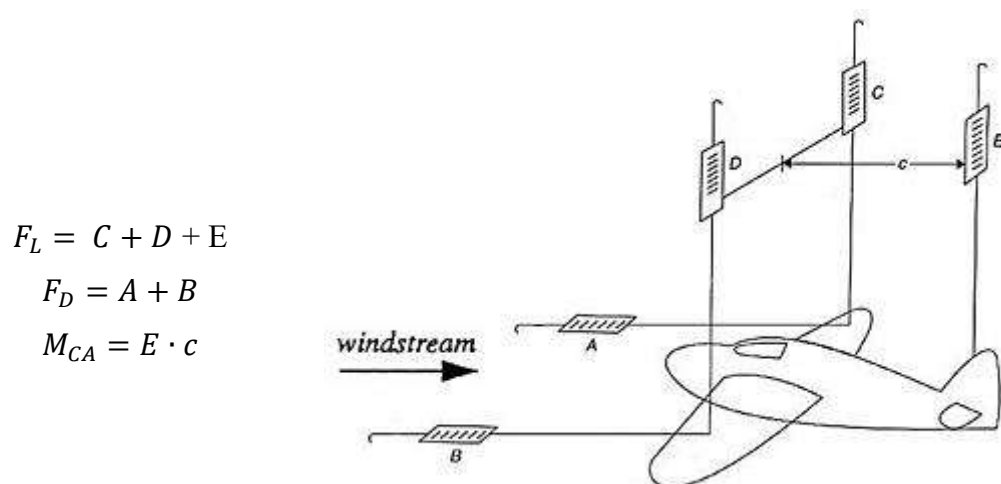
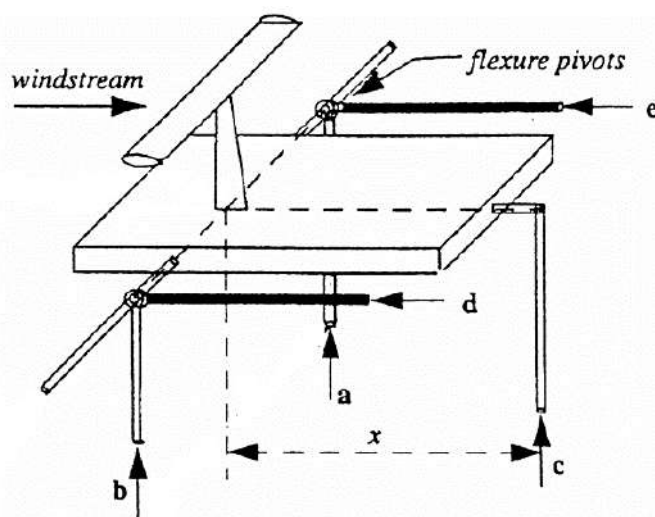


Figura 11: Balanza externa de cable.

Como desventaja, este tipo de balanza tiene el problema que debido al peso del objeto y la sustentación aerodinámica, la tensión de los cables nunca puede llegar a ser cero. Otro inconveniente es el arrastre generado por los cables que resulta difícil de medir. Además, los cables pueden romperse, lo que podría llevar a la pérdida del objeto, por lo que este tipo de balanza resulta menos versátil y robusta que otras balanzas externas. Por estos motivos, este tipo de balanza tiene un uso limitado.

Balanza de plataforma: este tipo de balanza se caracteriza por utilizar barras para sostener una placa sobre la cual se coloca el objeto de estudio. En la Figura 12 se presenta un esquema de la balanza de plataforma y una estimación de los resultados de las fuerzas aerodinámicas que se pueden obtener con este método. La figura muestra el objeto sobre la placa, sostenida por tres barras verticales para medir la fuerza de sustentación y el momento de cabeceo, y dos barras horizontales para obtener una medida de la fuerza de arrastre.

Las balanzas de plataforma son las más empleadas en el ámbito de la aeronáutica en túneles de viento debido a su robustez y relativa facilidad de construcción. Sin embargo, como desventaja, requieren correcciones en la medición del momento de cabeceo ya que se debe transferir la lectura del momento desde el centro de la balanza hasta el centro del objeto. Asimismo, la fuerza de arrastre produce un falso momento de cabeceo que debe ser restado del resultado final.



$$F_L = a + b + c$$

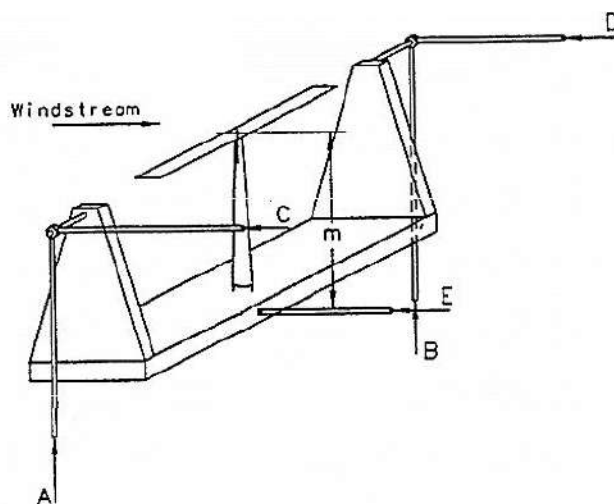
$$F_D = d + e$$

$$M_{CA} = c \cdot x$$

Figura 12: Balanza externa de plataforma.

Balanza de yugo: este tipo de balanza también utiliza barras para sostener una estructura sobre la cual se coloca el objeto de estudio. En la Figura 13 se presenta un esquema de la balanza de yugo y una estimación de los resultados de las fuerzas aerodinámicas que se pueden obtener con este método. La figura muestra el objeto sobre la estructura, sostenida por dos barras verticales para medir la fuerza de sustentación y tres barras horizontales para obtener una medida de la fuerza de arrastre y del momento de cabeceo.

En este tipo de balanza el centro del objeto coincide con el centro de la balanza, este aspecto representa una ventaja respecto de la balanza de plataforma. Sin embargo, este diseño conduce a mayores desviaciones en las medidas respecto de la balanza de plataforma, en particular en el momento de cabeceo, que se mide con las barras horizontales en lugar de las barras verticales.



$$F_L = A + B$$

$$F_D = C + D + E$$

$$M_{CA} = E \cdot m$$

Figura 13: Balanza externa de yugo.

Balanza Piramidal: en la Figura 14 se presenta un esquema de la balanza piramidal y una estimación de los resultados de las fuerzas aerodinámicas que se pueden obtener con este método. Como ventajas de estas balanzas se tiene que la lectura del momento de cabeceo resulta precisa y que la lectura de las fuerzas de sustentación y de arrastre resulta directa e independiente.

Entre las principales desventajas de este tipo de balanza se presenta la gran complejidad en la construcción, calibración y alineación de sus barras inclinadas. Asimismo, la existencia de deformaciones en las barras inclinadas afecta la correcta alineación y altera la precisión en la lectura del momento de cabeceo.

$$F_L = A$$

$$F_D = B$$

$$M_{CA} = C \cdot m$$

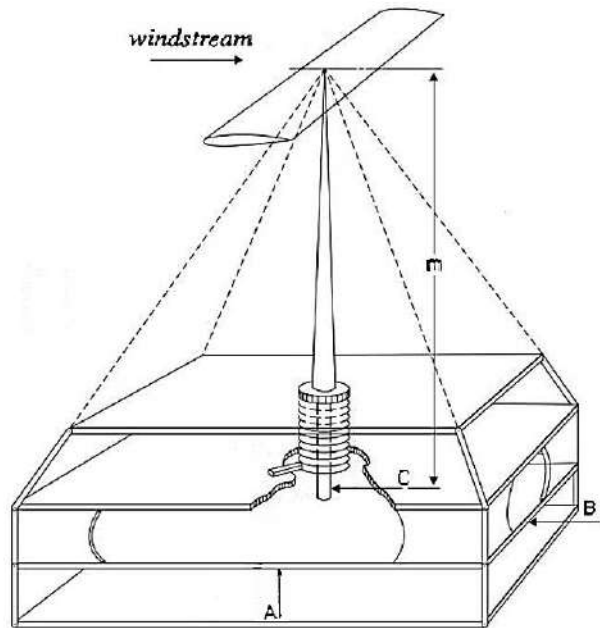


Figura 14: Balanza externa piramidal.

2.4.2 Balanzas internas

Las balanzas internas están desarrolladas para excluir las fuerzas actuantes en los soportes. Este tipo de balanza se puede instalar en el interior del objeto de estudio o en la unión del objeto-soporte. En la Figura 15 se presenta un esquema típico de balanza interna, que posee el sensor de fuerza situado en el interior del objeto de estudio. La Figura 16 muestra en detalle el sensor de fuerzas de la balanza interna.

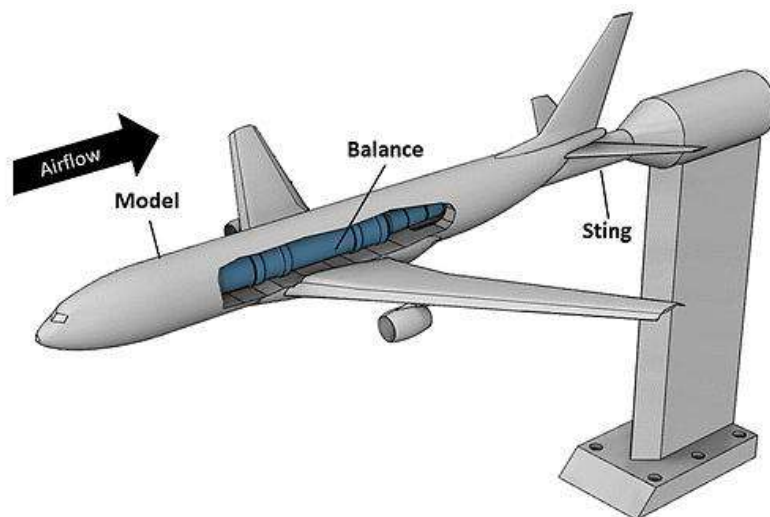


Figura 15: Balanza interna.

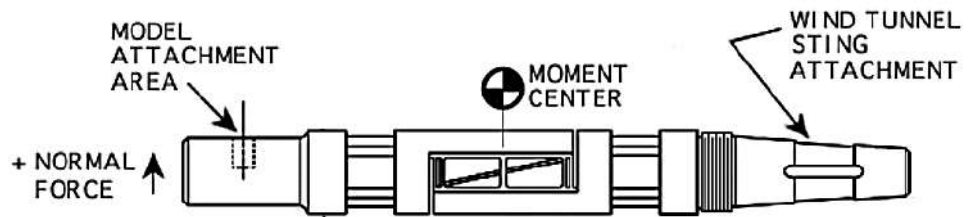


Figura 16: Sensor de fuerzas de una balanza interna.

Cuando se instala una balanza interna dentro del objeto, solo se miden las fuerzas actuantes sobre el objeto, el soporte en este caso solo perturba el flujo de aire. Cuando la balanza interna se instala en la unión objeto-soporte, ésta es protegida del flujo por un carenado cilíndrico o cónico. La principal ventaja de la balanza interna es que no interfiere el flujo de aire y por lo tanto no afecta las medidas de fuerzas y de momento de cabeceo, resulta entonces más precisa que la balanza externa. La desventaja es que se debe utilizar una balanza por cada objeto de estudio, aumentando la complejidad y los costos.

Capítulo 3: Instrumentación del túnel de viento

En este capítulo se presenta el desarrollo del sistema de instrumentación para el túnel de viento. En una primera fase se examina el diseño del túnel de viento de la Facultad de Ingeniería y se identifican los principales parámetros para la instrumentación. Luego, se aborda el diseño de la balanza aerodinámica, que incluye además una descripción de los componentes utilizados y un análisis de cómo se miden las fuerzas. Esta etapa crucial se complementa con la implementación de un mecanismo que permite realizar una variación controlada del ángulo de ataque del objeto de estudio. El capítulo concluye con la selección minuciosa del sensor para medir la velocidad del aire.

3.1 Consideraciones de diseño

Resulta fundamental comenzar este apartado con un análisis del túnel de viento de la Facultad de Ingeniería. Este análisis permite tomar decisiones vinculadas al correcto diseño del sistema de instrumentación y asegurar una implementación precisa y efectiva del sistema. En el apartado 2.3 se abordaron diversos aspectos relacionados con el túnel de viento, tales como su definición, modo de operación, aplicaciones y componentes. En esta instancia se retoman algunos conceptos para describir el túnel de viento de la Facultad de Ingeniería, representado en la Figura 17.

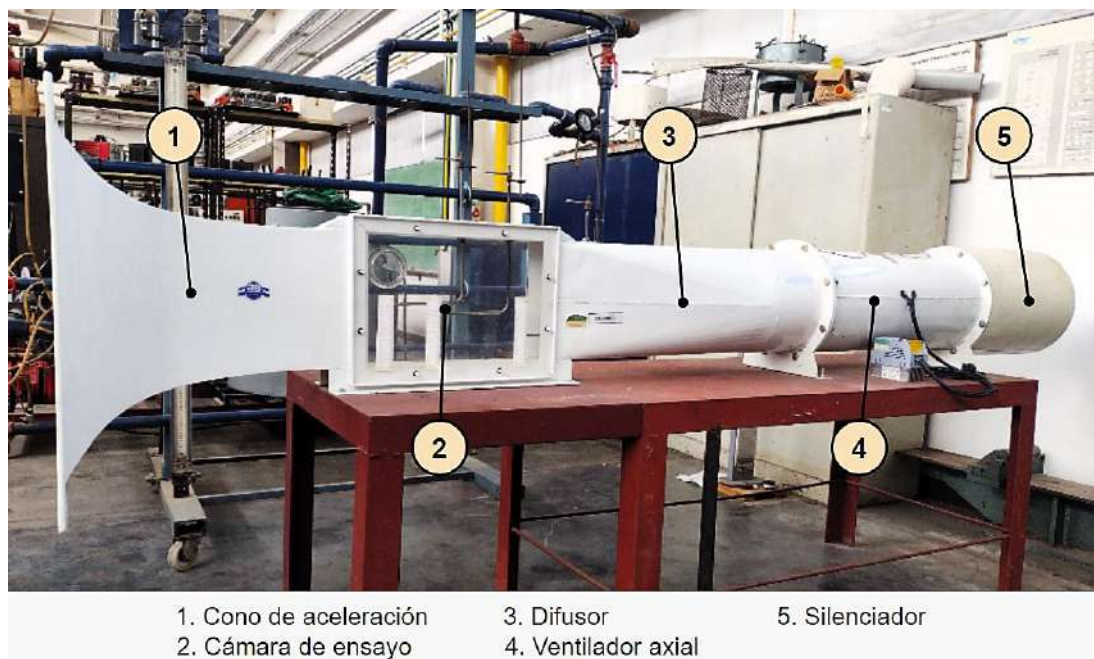


Figura 17: Túnel de viento de la Facultad de Ingeniería.

Los componentes que conforman el túnel de viento de la Facultad de Ingeniería incluyen el cono de aceleración (1), la cámara de ensayo (2), el difusor (3), el sistema de ventilación, compuesto por dos ventiladores axiales con variador de velocidad (4) y el silenciador (5). Como muestra la Figura 18, la cámara de ensayo presenta una estructura de hierro en las caras superior e inferior y de placas extraíbles de acrílico en las caras laterales. Sus dimensiones internas son 8 cm de ancho, 47 cm de largo y 31 cm de alto, y está equipada con 3 tubos Pitot para medir la velocidad del flujo de aire. La potencia de los ventiladores y la geometría del túnel permiten alcanzar en la cámara de ensayo una velocidad del flujo de aire de 35 m/s como máximo.



Figura 18: Cámara de ensayo del túnel de viento de la Facultad de ingeniería.

El diseño de la balanza aerodinámica requiere definir el rango a medir de las fuerzas de sustentación y de arrastre y del momento de cabeceo. Estas fuerzas presentan una relación directa con los coeficientes C_L , C_D y C_M (ecuaciones (1), (2) y (3)). Se parte entonces con los rangos máximos alcanzados de tales coeficientes en los perfiles NACA 0012 y NACA 2412 presentados en el apartado 2.2 (Figura 6, Figura 7 y Figura 9). Asimismo, se determina el rango del ángulo de ataque. De esta manera, los intervalos considerados como referencia para el diseño de la balanza aerodinámica son los siguientes:

$$-1,5 \leq C_L \leq 1,5 \quad (5)$$

$$0 \leq C_D \leq 0,15 \quad (6)$$

$$-0,1 \leq C_M \leq 0,1 \quad (7)$$

$$-20^\circ \leq \alpha \leq 20^\circ \quad (8)$$

La cota inferior del coeficiente C_L en (5) resulta negativa, que indica que en ciertos ángulos de ataque el perfil recibe empuje hacia abajo. Por lo tanto, la balanza aerodinámica debe medir fuerza de sustentación tanto positiva como negativa.

El rango del coeficiente de sustentación determinado en (5) y la aplicación de la ecuación (1) permiten estimar el rango requerido de la balanza aerodinámica para medir la fuerza de sustentación, el rango estimado resulta: $[-7,88 \text{ a } 7,88] \text{ N}$. En el cálculo de esta medida se consideraron los siguientes valores: $\rho = 1,225 \text{ kg/m}^3$ que corresponde al aire a 15°C y 1 atm , $v = 35 \text{ m/s}$, $A = 0,007 \text{ m}^2$ que corresponde al área de planta de un perfil alar con envergadura $0,07 \text{ m}$ y cuerda $0,1 \text{ m}$, $C_L = 1,5$.

De manera análoga, el rango del coeficiente de arrastre determinado en (6) y la aplicación de la ecuación (2) permiten estimar el rango requerido de la balanza aerodinámica para medir la fuerza de arrastre, el rango estimado resulta: $[0 \text{ a } 0,78] \text{ N}$. En el cálculo de esta medida se consideraron los siguientes valores: $\rho = 1,225 \text{ kg/m}^3$, $v = 35 \text{ m/s}$, $A = 0,007 \text{ m}^2$, $C_D = 0,15$.

Por último, el rango del coeficiente de momento de cabeceo determinado en (7) y la aplicación de la ecuación (3) permiten estimar el rango requerido de la balanza aerodinámica para medir el momento de cabeceo, el rango estimado resulta: $[-0,053 \text{ a } 0,053] \text{ N} \cdot \text{m}$. En el cálculo de esta medida se consideraron los siguientes valores: $\rho = 1,225 \text{ kg/m}^3$, $v = 35 \text{ m/s}$, $A = 0,007 \text{ m}^2$, $C = 0,1 \text{ m}$, $C_M = 0,1$.

En la Tabla 1 se presenta un resumen de los parámetros contemplados para el proceso de diseño del sistema de instrumentación. Además, el diseño requiere la consideración de los siguientes objetivos:

- Minimizar la interferencia de la balanza con el flujo de aire.
- Reducir el acoplamiento de las fuerzas de sustentación y arrastre. Esto significa que la salida asociada a la fuerza de sustentación debe tender a cero cuando no exista tal esfuerzo, aunque la magnitud de la fuerza de arrastre sea importante. De manera

análoga, la salida asociada con la fuerza de arrastre debe tender a cero cuando no hay arrastre, aunque la sustentación sea elevada.

- Brindar facilidades en la balanza para el montaje y desmontaje del objeto de estudio.
- Posibilitar la variación del ángulo de ataque del objeto de estudio.
- Facilitar el uso y mantenimiento del sistema completo: balanza aerodinámica, actuador que gobierna el ángulo de ataque, sensor de velocidad de aire y equipo electrónico.

Tabla 1: Parámetros de diseño para el sistema de instrumentación.

<i>Máxima velocidad del aire</i>	<i>35 m/s</i>
<i>Dimensiones de la cámara de ensayo</i>	<i>8 cm x 47 cm x 31 cm</i>
<i>Rango del coeficiente de sustentación</i>	<i>[-1,5 a 1,5]</i>
<i>Rango del coeficiente de arrastre</i>	<i>[0 a 0,15]</i>
<i>Rango del coeficiente de momento de cabeceo</i>	<i>[-0,1 a 0,1]</i>
<i>Rango de variación del ángulo de ataque</i>	<i>[-20 a 20] °</i>
<i>Rango de fuerza de sustentación</i>	<i>[-7,88 a 7,88] N</i>
<i>Rango de fuerza de arrastre</i>	<i>[0 a 0,78] N</i>
<i>Rango de momento de cabeceo</i>	<i>[-0,053 a 0,053] N·m</i>

3.2 Medición de fuerzas

Este apartado trata el proceso de diseño de la balanza aerodinámica, un elemento central en la instrumentación del túnel de viento ya que permite obtener una medida de las fuerzas aerodinámicas. Esta etapa del estudio se focaliza en la materialización concreta de las ideas y objetivos previamente delineados. Cada decisión tomada en este proceso tiene un impacto directo en la estabilidad y capacidad operativa del sistema.

3.2.1 Selección del transductor de fuerza

El elemento seleccionado para medir las fuerzas aerodinámicas es una celda de carga de punto único. Este transductor de fuerza utiliza el principio de deformación elástica para medir fuerzas mecánicas y convertirlas en señales eléctricas. La Figura 19 presenta la

celda de carga seleccionada, que está construida en aluminio y dispone de orificios roscados en sus extremos para facilitar la fijación.

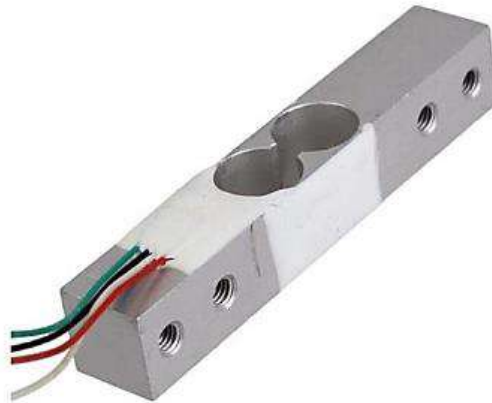


Figura 19: Celda de carga de punto único.

Para medir la fuerza, la celda de carga emplea galgas extensométricas como elementos sensibles. Estas galgas consisten en alambres metálicos delgados que se adhieren a la superficie del material elástico de la celda. Cuando se aplica una fuerza sobre la celda de carga, el material elástico (aluminio) se deforma ligeramente, lo que provoca que las galgas extensométricas se estiren o se compriman, alterando así su resistencia eléctrica. La Figura 20 (a) muestra, en forma esquemática, la deformación de una celda de carga cuando está sometida a un esfuerzo. También se aprecia en la figura que la geometría de la celda está diseñada para facilitar las deformaciones del material elástico en áreas específicas, precisamente donde se instalan las galgas extensométricas.

La variación en la resistencia de las galgas se mide utilizando un puente de Wheatstone, que es un circuito eléctrico que combina cuatro galgas. La Figura 20 (b) muestra un esquema de este circuito. Al aplicar una tensión de excitación V_{EX} al puente de Wheatstone se puede detectar un desequilibrio en el circuito causado por el cambio en la resistencia de las galgas, lo que produce una tensión de salida V_O proporcional a la fuerza aplicada. La relación entre V_O y V_{EX} , cuando la celda está cargada a su capacidad nominal, se conoce como sensibilidad y se expresa en mV/V .

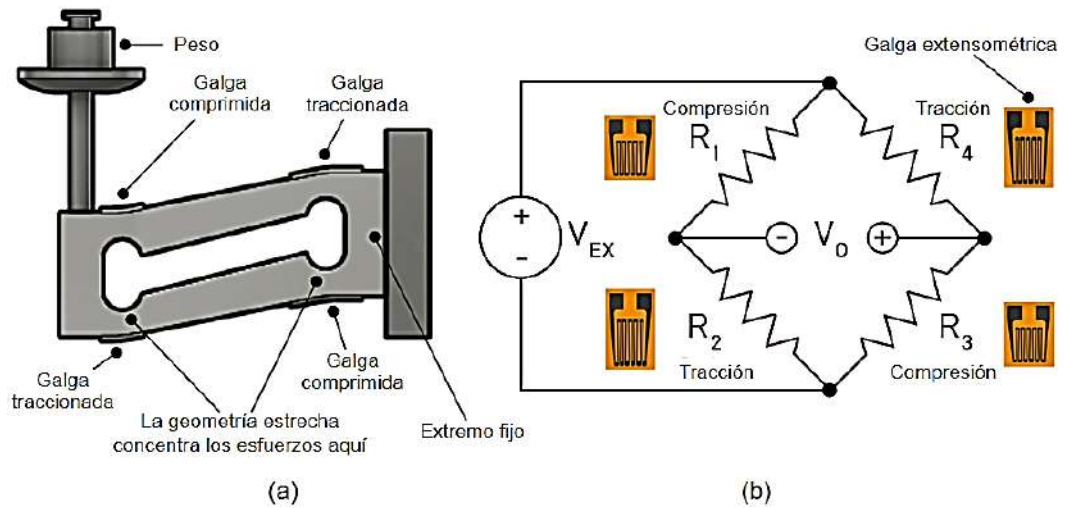


Figura 20: Celda de carga de punto único, (a) esquema de montaje y funcionamiento, (b) esquema del puente de Wheatstone.

La celda de carga de punto único se distingue de otros tipos de celdas de carga (como las de tipo viga) en que la geometría de sus elementos deformables la hace prácticamente insensible al efecto de la excentricidad en la aplicación de la carga [6] [7]. Por ejemplo, la Figura 21(a) muestra una celda de carga de punto único sometida a una fuerza F_1 aplicada a una distancia d_1 desde su extremo fijo. La Figura 21(b) presenta la misma celda de carga sometida a una fuerza F_2 aplicada a una distancia d_2 , donde $d_2 > d_1$. A pesar de las diferencias en la magnitud y el punto de aplicación de las fuerzas, las salidas de ambas celdas de carga serían iguales o prácticamente iguales. Esto se debe a que los elementos deformables de las celdas de carga de punto único solo son sensibles a las componentes perpendiculares de las cargas aplicadas. Por lo tanto, como la componente perpendicular de la carga en la Figura 21(b) F_{2v} es igual a la carga aplicada en la Figura 21(a) F_1 , las salidas de las celdas de carga serán iguales, independientemente del punto de aplicación de la carga en cada caso.

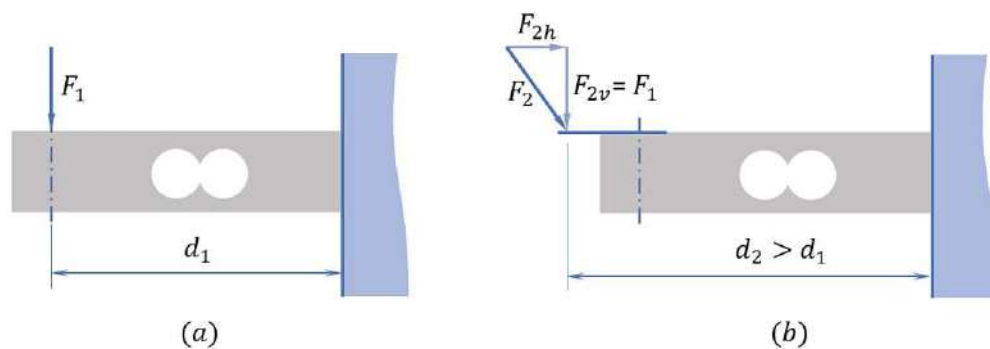


Figura 21: Análisis de sensibilidad a esfuerzos de la celda de carga de punto único.

La elección del modelo apropiado de celda de carga requiere plantear una situación de compromiso que permita obtener buena sensibilidad con un rango de medición adecuado. Con este planteo se determina utilizar 3 celdas de carga, donde cada celda posee una capacidad nominal de 1 kg. En la Tabla 2 se presentan las principales características técnicas de las celdas de carga.

Tabla 2: Principales características de la celda de carga seleccionada.

<i>Carga nominal</i>	<i>1 kg</i>
<i>Sensibilidad</i>	<i>0,9 mV/V</i>
<i>Conexión</i>	<i>Cable rojo: Tensión de excitación +</i> <i>Cable negro: Tensión de excitación -</i> <i>Cable blanco: Tensión de salida +</i> <i>Cable verde: Tensión de salida -</i>

Módulo de acondicionamiento de señal

La señal analógica procedente de la celda de carga es relativamente tenue, por lo que resulta necesario emplear un dispositivo que acondicione la señal antes de llegar a la unidad microcontrolada (MCU). Para este fin se selecciona un módulo que incorpora el circuito integrado (CI) Hx711 [8], diseñado en forma específica para acondicionar señales de un puente Wheatstone. La Figura 22 presenta el módulo Hx711.

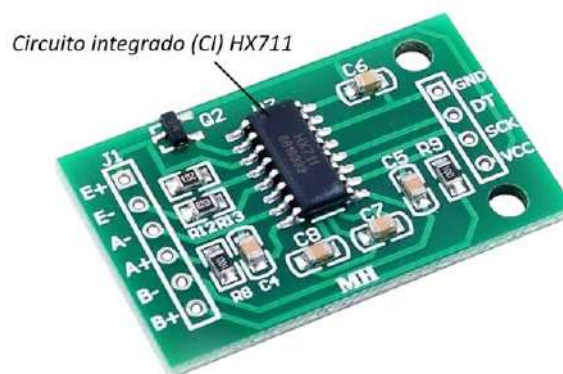


Figura 22: Módulo Hx711.

La Figura 23 muestra un diagrama de bloques de una aplicación típica del CI Hx711. Este dispositivo posee 2 canales de entrada analógica (A y B), que se conectan a un amplificador de ganancia programable por medio de un multiplexor, la salida se realiza a

través de un conversor analógico digital (ADC) de 24 bits. Estos canales funcionan en modo diferencial, lo que significa que miden la diferencia de tensión entre dos puntos. La señal del canal A se amplifica mediante una ganancia ajustable de 64 o 128, mientras que la señal del canal B solo se amplifica con una ganancia de 32.

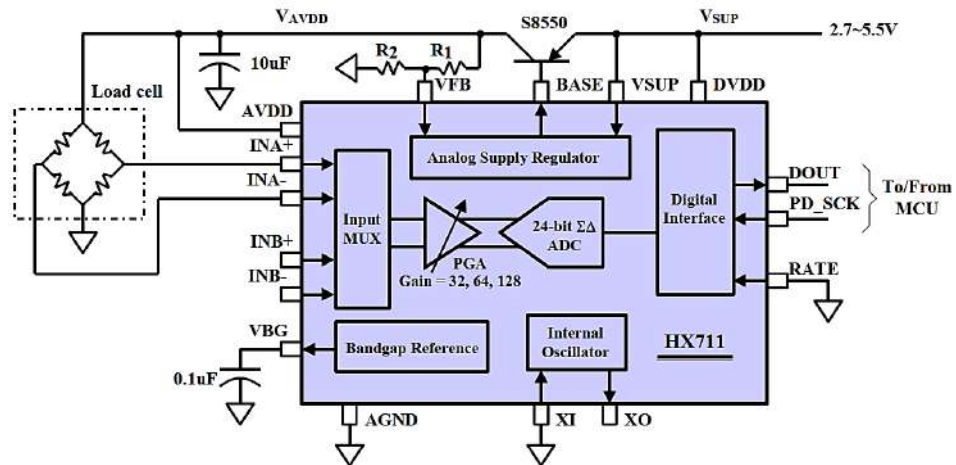


Figura 23: Diagrama de bloques de una aplicación típica del CI Hx711.

El módulo Hx711 posee una interfaz de comunicación serial sincrónica de 2 hilos (datos y reloj), que permite la comunicación con la MCU que recibe la información. Este módulo posee además una salida de tensión regulada, especialmente útil para proveer excitación a un puente Wheatstone. Por todo esto el módulo Hx711 es especialmente apropiado para el acondicionamiento de la señal proveniente de una celda de carga basada en galgas extensométricas. En la Tabla 3 se presentan las principales características técnicas del módulo Hx711.

Tabla 3: Principales características del módulo Hx711.

<i>Tensión de alimentación</i>	<i>[2,6 a 5,5] VDC</i>
<i>Comunicación con MCU</i>	<i>Serial: Clock y Data</i>
<i>Conexión a celda de carga</i>	<i>E+ : Tensión de excitación +</i> <i>E- : Tensión de excitación –</i> <i>A+ / A- : Canal A</i> <i>B+ / B- : Canal B</i>

3.2.2 Diseño de la balanza aerodinámica

Para el diseño de la balanza aerodinámica se evalúan las características del túnel de viento y se atienden las consideraciones de diseño definidas en el apartado 3.1. Asimismo, se evalúan los modelos de balanzas externas mencionados en el apartado 2.4.1. Conforme a las desventajas indicadas para cada modelo de balanza externa, se opta por descartar el empleo del modelo de cable y del modelo piramidal. Los modelos restantes, de plataforma y de yugo, se consideran los más convenientes para cumplir con los objetivos establecidos, sin embargo, sus estructuras resultan demasiado grande en comparación con las dimensiones de la cámara de ensayo del túnel de viento. Por lo tanto, se opta por desarrollar un diseño nuevo.

Como resultado se logra un diseño compacto, que se destaca por la robustez lograda, los bajos requisitos de mantenimiento, la interacción reducida entre los componentes y una interferencia despreciable en el flujo de aire. La Figura 24 ofrece un detalle de los distintos elementos que integran el diseño de la balanza aerodinámica desarrollada. En dicha figura, se presenta la balanza ubicada sobre uno de los cerramientos de acrílico de la cámara de ensayo. Cabe destacar que la balanza queda del lado exterior de la cámara de ensayo y por lo tanto minimiza la interferencia con el flujo de aire.

La Figura 24 muestra que el soporte base A sostiene la estructura completa de la balanza aerodinámica sobre la placa de acrílico H. La celda C1 se encuentra anclada a la placa de acrílico H en uno de sus extremos mediante el soporte base A. En el extremo opuesto de la celda C1, el codo unión B sujeta de manera ortogonal la celda C2. A su vez, la celda C2, en su extremo opuesto, soporta la celda C3 mediante el ángulo D. Asimismo, la celda C2 cumple la función de servir como soporte para la varilla de hierro C. Las celdas de carga C1, C2 y C3 se encargan de medir las fuerzas de sustentación, de arrastre y el momento de cabeceo respectivamente, como se demuestra en el apartado 3.3.3.

La varilla C sostiene el perfil alar G en el lado opuesto de la placa de acrílico H y también actúa como un punto de pivote permitiendo el giro libre del perfil alar G y de la palanca F. Sin embargo, el giro del perfil alar G se encuentra limitado por un vínculo con la palanca F, la cual se apoya en la celda C3 mediante la unión E. La varilla C permite desmontar el perfil alar G; para montar un nuevo perfil alar se lo inserta en la varilla C y se lo acopla a la palanca F.

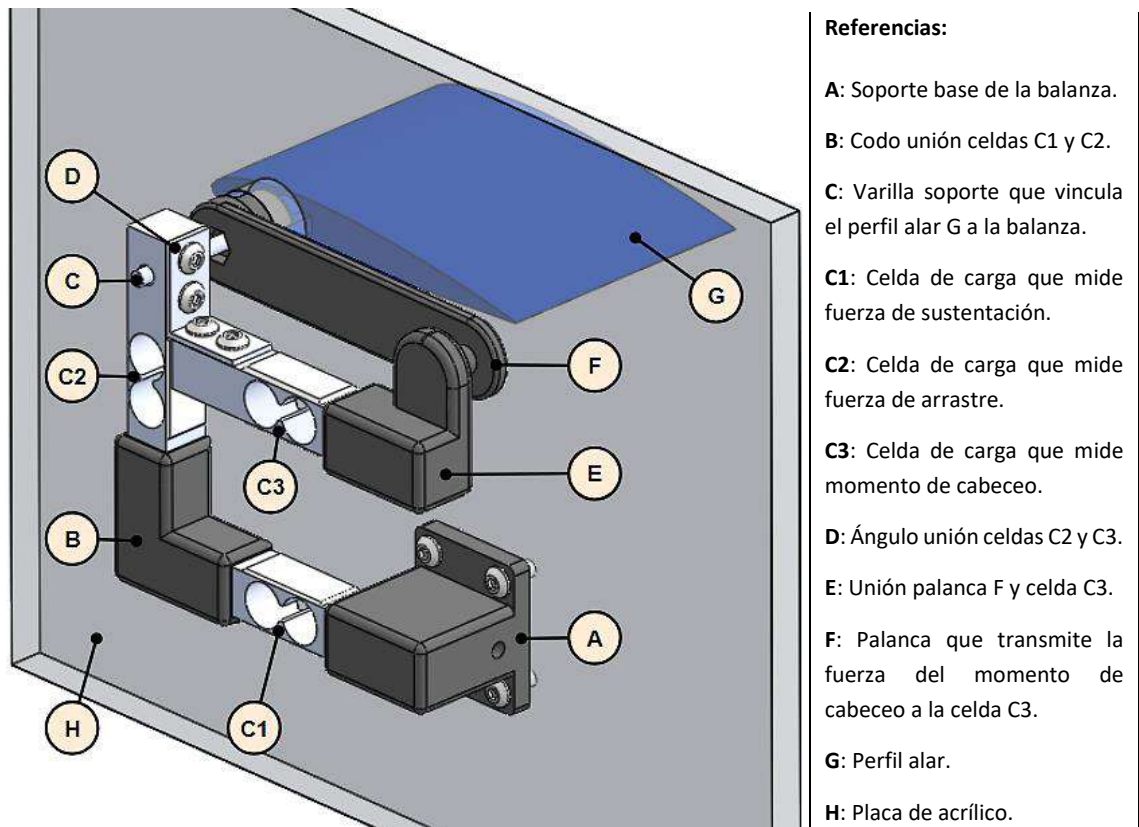


Figura 24: Diseño de la balanza aerodinámica.

3.3 Variación del ángulo de ataque

En este apartado se aborda el proceso de incorporar a la balanza aerodinámica un mecanismo que permita la variación del ángulo de ataque del perfil aerodinámico. En primer lugar se presenta la selección del servomotor utilizado como actuador. Luego, se describe el diseño del mecanismo que ajusta el ángulo de ataque. Por último, se efectúa un análisis de la balanza en relación a la medición de las fuerzas aerodinámicas.

3.3.1 Selección del actuador

Para variar el ángulo de ataque se opta por el servomotor SG92R [9], que es un dispositivo electromecánico que controla la posición angular de un eje de acuerdo a la señal PWM proveniente de la MCU. La Figura 25 muestra el servomotor seleccionado.

El funcionamiento del servomotor requiere de dos pines de alimentación y un pin para la señal PWM. Este dispositivo contiene un motor conectado a un sistema de engranajes que reduce la velocidad y aumenta el torque para lograr movimientos precisos. Asimismo, el servomotor está equipado con un potenciómetro interno que se vincula en forma mecánica

al eje principal. Este potenciómetro actúa como sensor de retroalimentación e informa al circuito interno sobre la posición actual del eje. Este circuito, a su vez, interpreta la señal de control PWM proveniente de la MCU y ajusta la energía suministrada al motor a través de un controlador y un amplificador para lograr el posicionamiento. En la Tabla 4 se presentan las principales características técnicas del servomotor SG92R.



Figura 25: Servomotor SG92R.

Tabla 4: Principales características del servomotor SG92R.

Tensión de alimentación	[3 a 6] VDC
Comunicación con MCU	PWM
Torque	2,5 kg·cm
Ángulo de rotación	[-90 a 90] °

3.3.2 Diseño del mecanismo

En el apartado 3.1 se estableció la importancia de variar el ángulo de ataque del perfil aerodinámico en un rango de $\pm 20^\circ$. Sin embargo, el esquema de la balanza presentado hasta aquí (Figura 24) no permite el movimiento angular de la palanca F y, por lo tanto, no permite variar el ángulo de ataque del perfil alar G.

Para lograr la capacidad de variar el ángulo de ataque se incorporan modificaciones sobre la balanza que resultan en el esquema presentado en la Figura 26. La figura muestra la incorporación del servomotor I, que está montado sobre la celda de carga C3 y está vinculado mediante la biela J a la palanca F. El servomotor acciona la biela J, que desplaza la palanca F y modifica el ángulo de ataque del perfil alar G.

La longitud de la biela J es tal que una variación de 90° en el servomotor I implica una variación de 20° en el ángulo del perfil alar G. En consecuencia, una variación en el servomotor de $\pm 90^\circ$ resulta en una variación equivalente de $\pm 20^\circ$ en el ángulo de ataque del perfil alar.

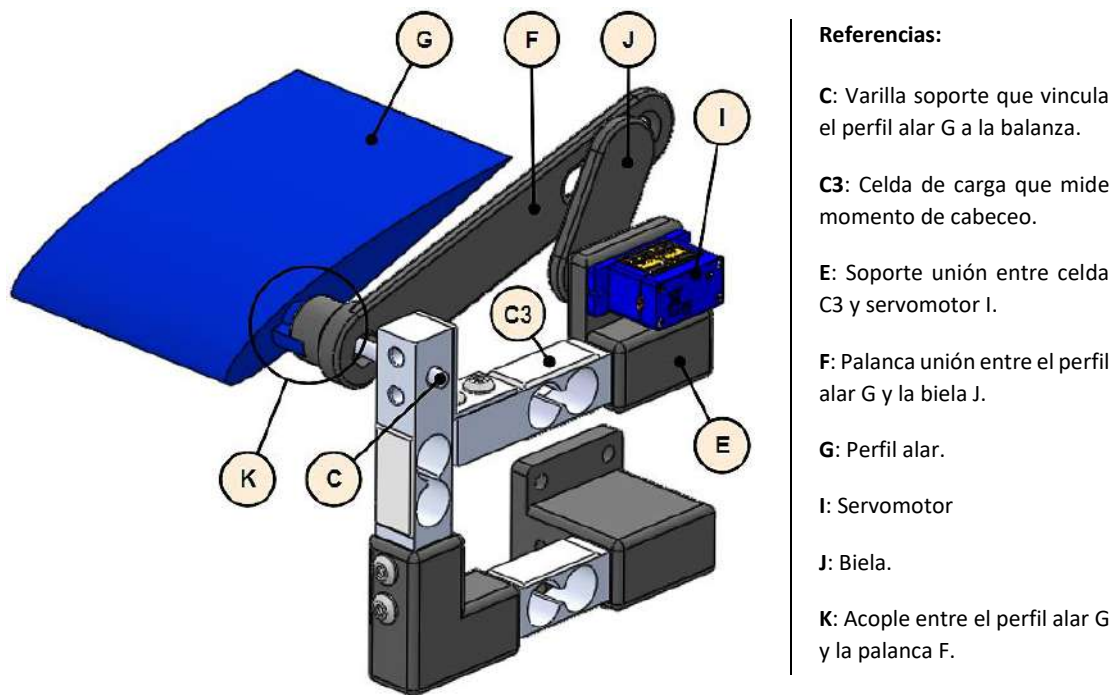


Figura 26: Diseño del mecanismo de variación del ángulo de ataque.

3.3.3 Análisis de fuerzas en la balanza aerodinámica

Para analizar la medición de las fuerzas aerodinámicas, se inicia con la Figura 27 (a), que muestra el perfil de ala G en la balanza aerodinámica. La figura también ilustra la fuerza aerodinámica resultante F_R , que actúa sobre el perfil de ala en el punto CP. Por otro lado, la Figura 27 (b) presenta un esquema simplificado de la balanza mediante 2 chapas. La chapa 1 está compuesta por las celdas de carga C1, C2 y C3, y por la biela J. La chapa 2 está compuesta por el perfil G y la palanca F (la conexión entre el perfil G y la palanca F tiene restricciones de movimiento en todas las direcciones, por lo que puede considerarse una sola pieza). Los puntos 1 y 2 son vínculos que restringen tres grados de libertad, el punto 3 restringe dos grados de libertad (es una articulación) y el punto 4 restringe un grado de libertad (es una corredera). En el esquema, se considera que la fuerza aerodinámica F_R actúa en el punto 3, que corresponde al punto de sujeción del perfil G.

Asimismo, en el punto 3 se considera aplicado el momento de cabeceo M_{CA} debido a la traslación de la fuerza aerodinámica F_R desde el centro de presión (CP) hasta el centro aerodinámico (CA).

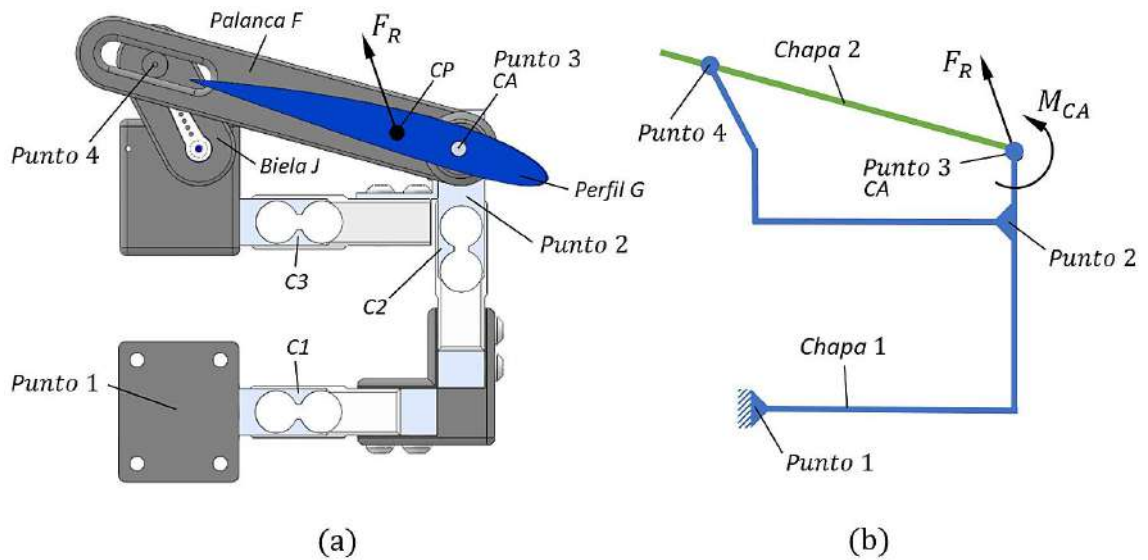


Figura 27: Esquema simplificado de la balanza aerodinámica mediante 2 chapas.

Medición del momento de cabeceo

La Figura 28 muestra el diagrama de cuerpo libre para las chapas 1 y 2. Las fuerzas F_{3H} y F_{3V} son las reacciones de vínculo, horizontal y vertical, en el punto 3, F_4 es la reacción de vínculo en el punto 4, α es el ángulo de ataque y r_1 es la distancia entre los puntos 3 y 4. Nótese que la distancia r_1 y la dirección de la fuerza F_4 no son constantes, sino que varían en función de α . De esta manera, si se plantea la condición de equilibrio en la chapa 2 resulta que:

$$M_{CA} = F_4 \cdot r_1 \quad (9)$$

La ecuación (9) indica que el momento de cabeceo M_{CA} se puede calcular mediante la determinación de r_1 y la medición de F_4 . Dado que tanto la distancia r_1 como la dirección de la fuerza F_4 varían en función de α , es necesario obtener una relación entre estas variables y el ángulo de ataque α . La Figura 29 muestra un esquema que contribuye a este análisis, donde β es el ángulo de la biela J, es decir, el ángulo aplicado en el servomotor, $d_1 = 7,027 \text{ cm}$ y $r_2 = 2,55 \text{ cm}$.

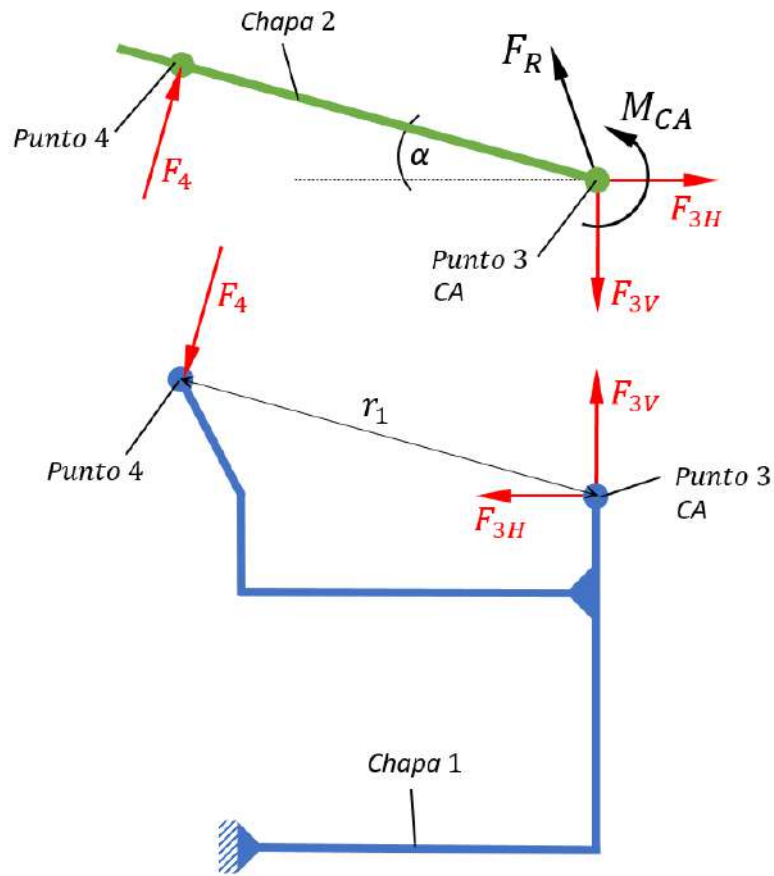


Figura 28: Diagrama de cuerpo libre de las chapas.

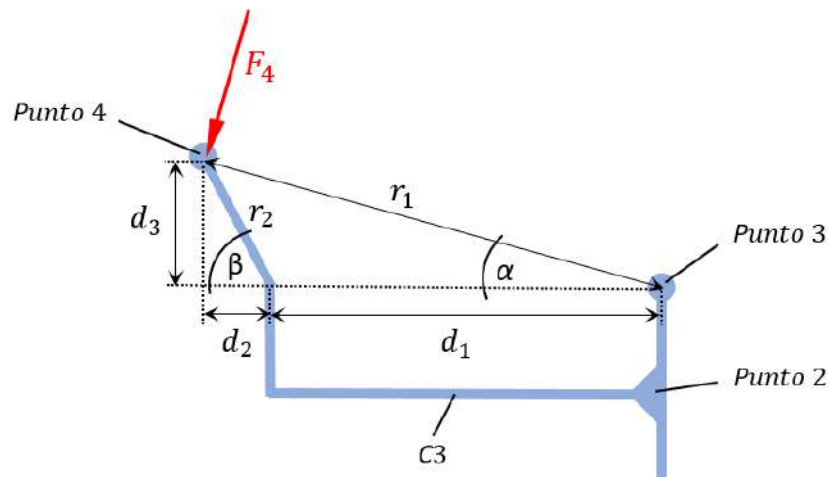


Figura 29: Análisis medición del momento de cabeceo en función del ángulo de ataque.

Como se mencionó en el apartado 3.2.1, las celdas de carga de punto único solo son sensibles a los esfuerzos aplicados en dirección perpendicular a su eje longitudinal y son insensibles a la excentricidad de la carga. Por lo tanto, a partir de la Figura 29, se puede determinar que la fuerza medida en la celda C3 (F_{C3}) es:

$$F_{C3} = F_4 \cdot \cos(\alpha) \quad (10)$$

Al reemplazar la ecuación (10) en la ecuación (9) se obtiene:

$$M_{CA} = \frac{F_{C3}}{\cos(\alpha)} \cdot r_1 \quad (11)$$

Para determinar el valor de r_1 , se establece el siguiente sistema de ecuaciones a partir de la Figura 29:

$$\begin{cases} r_1^2 = (d_1 + d_2)^2 + d_3^2 \\ r_2^2 = d_2^2 + d_3^2 \\ r_1 \cdot \cos(\alpha) = d_1 + d_2 \end{cases} \quad (12)$$

Si se desarrolla el binomio al cuadrado en la primera ecuación del sistema (12) y luego se aplica la segunda ecuación del sistema, se obtiene:

$$r_1^2 = d_1^2 + 2 \cdot d_1 \cdot d_2 + r_2^2 \quad (13)$$

Al reemplazar la tercera ecuación del sistema (12) en la ecuación (13) resulta:

$$r_1^2 - 2 \cdot d_1 \cdot r_1 \cdot \cos(\alpha) = r_2^2 - d_1^2 \quad (14)$$

Si se completan ambos términos de la ecuación (14) con $(d_1 \cdot \cos(\alpha))^2$ y se resuelve, se obtiene la siguiente expresión matemática para r_1 en función del ángulo de ataque α :

$$r_1(\alpha) = d_1 \cdot \cos(\alpha) + \sqrt{r_2^2 - d_1^2 + (d_1 \cdot \cos(\alpha))^2} \quad (15)$$

Además, la Figura 29, permite obtener una expresión matemática para el ángulo β (ángulo aplicado al servomotor) en función del ángulo de ataque α :

$$\beta(\alpha) = \sin^{-1} \left(\frac{r_1(\alpha) \cdot \sin(\alpha)}{r_2} \right) \quad (16)$$

Medición de la fuerza de sustentación y de la fuerza de arrastre

La Figura 30 muestra un esquema que contribuye al análisis de fuerzas. Las fuerzas F_{1H} y F_{1V} , y el momento M_1 son las reacciones de vínculo en el punto 1. Como se mencionó en el apartado 3.2.1, las celdas de carga de punto único son insensibles a los momentos y a la excentricidad de la carga. Por lo tanto, si se plantea la condición de equilibrio resulta que la fuerza registrada por la celda C1 (F_{C1}) es la fuerza de sustentación F_L y la fuerza registrada por la celda C2 (F_{C2}) es la fuerza de arrastre F_D .

$$F_{C1} = F_{1V} = F_L \quad (17)$$

$$F_{C2} = F_{1H} = F_D \quad (18)$$

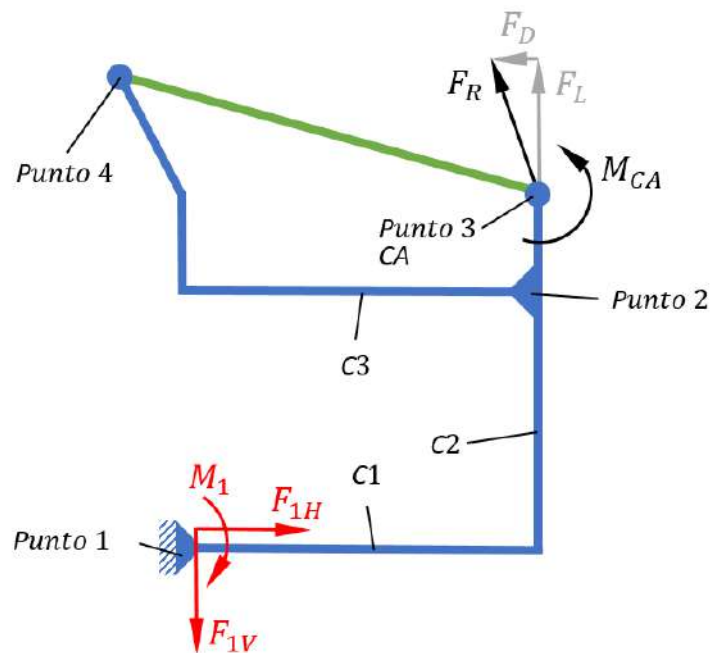


Figura 30: Análisis medición de fuerzas de sustentación y arrastre.

3.4 Medición de la velocidad del aire

En este apartado se aborda el proceso de selección de los elementos para medir la velocidad del aire. Entre ellos encontramos el tubo Pitot, cuyo funcionamiento se basa en el principio de Bernoulli [2]. La elección de este dispositivo se justifica por la presencia de 3 tubos Pitot en el túnel de viento, lo que facilita la medición de la velocidad del aire.

El tubo Pitot está formado por 2 tubos concéntricos, el tubo interior posee un orificio paralelo al flujo, mientras que el tubo exterior posee orificios perpendiculares a la dirección del flujo, como muestra la Figura 31.



Figura 31: Tubo Pitot.

El flujo de aire que ingresa por el tubo interior permanece en estado estacionario (velocidad nula). Si se conectan las salidas del tubo Pitot a un sensor de presión diferencial, se obtiene la diferencia entre la presión total y la presión estática. Esta diferencia es la denominada presión dinámica, que se debe al incremento de presión debido a la reducción a cero en la velocidad del flujo. De esta manera, la velocidad del flujo de aire se obtiene como:

$$v = \sqrt{\frac{2 \cdot (P_{total} - P_{estática})}{\rho}} \quad (19)$$

La presión dinámica máxima que el sensor debe medir se calcula al considerar en la ecuación (19) la velocidad máxima del flujo de aire dentro de la cámara de ensayo ($v_{máx} = 35 \text{ m/s}$). De esta manera se obtiene que la presión dinámica máxima que el sensor debe medir en el túnel de viento resulta $P_{dmáx} = 750 \text{ Pa}$. En el cálculo se considera $\rho = 1,225 \text{ kg/m}^3$ que corresponde a aire a $15 \text{ }^\circ\text{C}$ y 1 atm .

3.4.1 Selección del sensor de presión

Para convertir la presión diferencial proveniente del tubo Pitot en una señal eléctrica que la MCU pueda interpretar, se selecciona el transductor piezorresistivo de la serie MPXV5004DP [10]. Este dispositivo es un sensor de presión cuyo elemento sensible es una celda piezorresistiva capaz de proveer una señal eléctrica de tensión especialmente apropiada para la interfaz ADC de la MCU. La celda piezorresistiva está expuesta a dos entradas de presión, por lo que la salida del dispositivo es proporcional a la diferencia de presión entre las presiones presentes en las entradas. En la Figura 32 se muestra el sensor seleccionado.

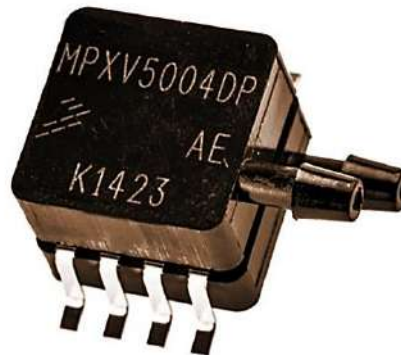


Figura 32: Sensor de presión MPX5004.

El funcionamiento del sensor requiere de 2 pines de alimentación y 1 pin para la salida de la señal analógica. El rango de medición de presión diferencial de este sensor es $[0 \text{ a } 3,92] \text{ kPa}$. La Figura 33 muestra la relación entre la diferencia de presión aplicada entre los orificios y la tensión de salida. Se aprecia en la figura que el nivel de tensión de salida tiene un offset de 1 VDC , es decir para una diferencia de presión nula la tensión de salida es de 1 VDC . Asimismo, se calculó mediante la ecuación (19) que el tubo Pitot en túnel de viento opera en un rango de presión diferencial comprendido en el intervalo de $[0 \text{ a } 750] \text{ Pa}$, por lo tanto, se deduce de la Figura 33 que el rango de tensión en la salida del sensor es de $[1 \text{ a } 1,75] \text{ VDC}$ en túnel de viento. La Tabla 5 presenta las principales características técnicas del sensor.

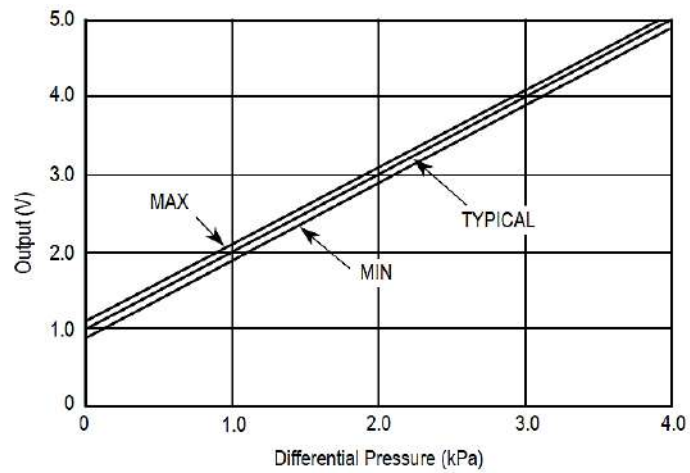


Figura 33: Tensión de salida del sensor MPX5004 en función de la presión aplicada.

Tabla 5: Principales características del sensor MPX5004.

<i>Tensión de alimentación</i>	<i>[4,75 a 5,25] VDC</i>
<i>Tensión de salida</i>	<i>[1 a 5] VDC</i>
<i>Rango de medición</i>	<i>[0 a 3,92] kPa</i>
<i>Sensibilidad</i>	<i>1 mV/Pa</i>

Capítulo 4: Adquisición y presentación de datos

En este capítulo se presenta el desarrollo de la placa de circuito impreso y se describe la interfaz de usuario del sistema de adquisición y presentación de datos. El capítulo comienza con la selección de la unidad microcontrolada, la pantalla y los botones de comando. Luego, se aborda el análisis del diseño del diagrama esquemático, y la posterior aplicación de este diagrama en el desarrollo de la placa de circuito impreso. Por último, se explican las principales características y funcionalidades de la interfaz de usuario, que ofrece facilidades de navegación desde un panel principal y desde un sitio web.

4.1 Selección de componentes

Selección de la unidad microcontrolada (MCU)

Se optó por utilizar la placa ESP32-DevKit [11] (Figura 34) basada en la MCU ESP32 [12]. La MCU ESP32 está diseñada principalmente para aplicaciones de Internet de las cosas (IoT) y sistemas embebidos que requieren una combinación de potencia de procesamiento, conectividad inalámbrica y eficiencia energética. La MCU ESP32 se basa en una arquitectura de procesador Tensilica Xtensa LX6 de 32 bits, que consta de 2 núcleos de procesamiento que pueden alcanzar una velocidad de hasta 240 MHz cada uno. En términos de conectividad inalámbrica, la MCU ESP32 incorpora Wi-Fi 802.11 b/g/n y Bluetooth Low Energy v4.2. Por otro lado, la MCU ESP32 ofrece una amplia gama de periféricos e interfaces que incluye UART, PWM, SPI, I2C, ADC y DAC, que facilitan la conexión de sensores, actuadores y otros dispositivos externos.

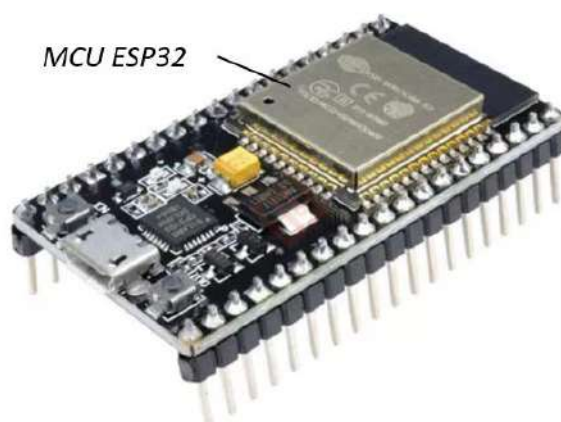


Figura 34: Placa ESP32-DevKit.

La placa ESP32-DevKit además cuenta con un regulador de tensión, una interfaz USB y 38 pines, de los cuales 24 pueden utilizarse para operar con otros dispositivos (sensores, actuadores, etc). El regulador de tensión ofrece la posibilidad de alimentar la placa ESP32-DevKit a 3,3 o 5 VDC y garantiza que la MCU ESP32 siempre reciba la tensión adecuada para su funcionamiento (3,3 VDC). La interfaz USB permite la comunicación entre la MCU ESP32 y una PC, que facilita el proceso de programación y depuración del programa desarrollado.

Selección de la pantalla

Para la presentación de los datos se optó por el módulo MSP1803 [13]. Este dispositivo incluye una pantalla a color TFT ST7735 y una ranura para tarjeta SD. La Figura 35 (a) muestra la pantalla, que tiene una longitud diagonal de 1.8", una resolución de 128 x 160 píxeles y retroiluminación led. La comunicación con la pantalla se realiza mediante el protocolo SPI, que es compatible con la MCU ESP32 y necesita 4 pines para funcionar (datos, reloj, selección de chip y selección de comando/datos). La Figura 35 (b) muestra la ranura para tarjeta SD incorporada sobre la cara posterior del módulo. Al igual que la pantalla, la ranura para tarjeta SD opera mediante el protocolo SPI lo que implica la necesidad de 4 pines para su funcionamiento (2 líneas de datos, reloj y selección de chip). El módulo requiere una tensión de alimentación de 3,3 VDC.

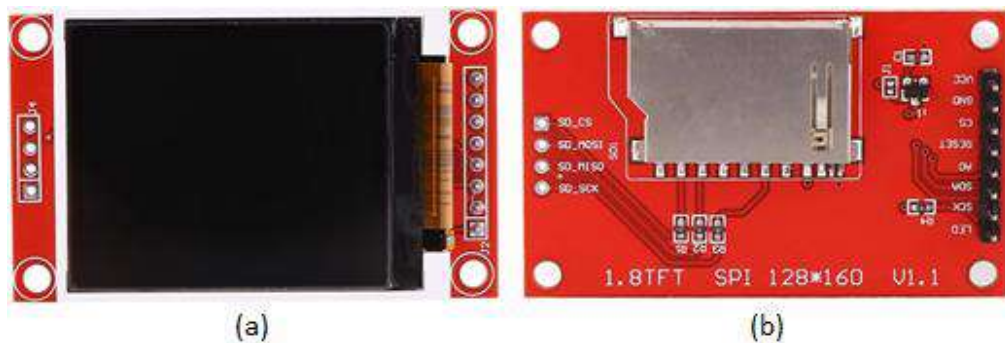


Figura 35: Módulo MSP1803, (a) pantalla y (b) ranura SD.

Selección de los botones de comando

Para interactuar con el sistema de adquisición de datos se seleccionó el encoder rotativo KY-040 [14] (Figura 36). Este dispositivo electromecánico cuenta con un eje que permite el giro en ambos sentidos y un pulsador que se activa al presionar el eje del encoder. El encoder tiene 5 pines de conexión: 2 pines para alimentación eléctrica, 1 pin para la salida

de señal digital del pulsador, y 2 pines para las salidas digitales de rotación, que proporcionan una indicación de cuánto el eje ha sido girado y en qué dirección.

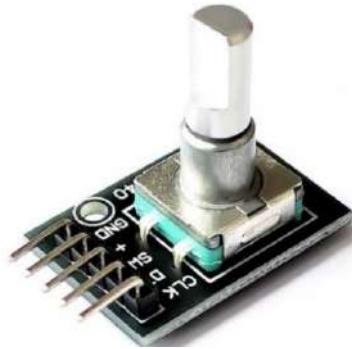


Figura 36: Encoder KY-040.

4.2 Diseño de la placa de circuito impreso

La placa de circuito impreso (PCB) se diseñó en KiCad, una herramienta de diseño electrónico gratuita y de código abierto. Este proceso constó de dos etapas: en la primera etapa se elaboró el esquemático, que ofrece una representación visual de los componentes del circuito eléctrico. En la segunda etapa se confeccionó la placa, que sirve de enlace eléctrico para los componentes físicos del sistema.

La Figura 37 presenta un esquema que proporciona una visión simplificada y clara de las conexiones entre los componentes del sistema de instrumentación y el sistema de adquisición y presentación de datos. En el centro de la figura destaca un rectángulo con un tono verde que contiene la placa ESP32-DevKit, los módulos Hx711 y el sensor MPX5004. El rectángulo indica que los componentes en su interior están integrados en la placa PCB.

En el lado izquierdo de la Figura 37 se muestran los componentes del sistema de instrumentación. Entre estos componentes se encuentran las tres celdas de carga responsables de traducir las fuerzas aerodinámicas en señales analógicas. Estas señales analógicas se amplifican y digitalizan por los módulos Hx711, que luego son transmitidas a la MCU ESP32 mediante comunicación serial. Asimismo, la figura muestra el tubo Pitot que se conecta al sensor de presión MPX5004 mediante un conducto flexible. Este sensor traduce la presión en una señal analógica que se envía al ADC de la MCU ESP32. Por último, el lado izquierdo de la figura ilustra el servomotor SG92R que establece el ángulo de ataque a partir de la señal PWM generada por la MCU ESP32.

En el lado derecho de la Figura 37 se muestran los componentes que permiten al usuario interactuar con el sistema. Entre estos componentes se encuentra el encoder KY-040 que transmite las instrucciones del usuario a la MCU ESP32 mediante señales digitales. Asimismo, la figura incluye el módulo MSP1803 de pantalla y ranura SD, que recibe los datos de la MCU ESP32 mediante comunicación SPI. Por último, la notebook representa la interfaz web con el usuario, que accede desde un sitio web por la conexión WiFi de la MCU ESP32.

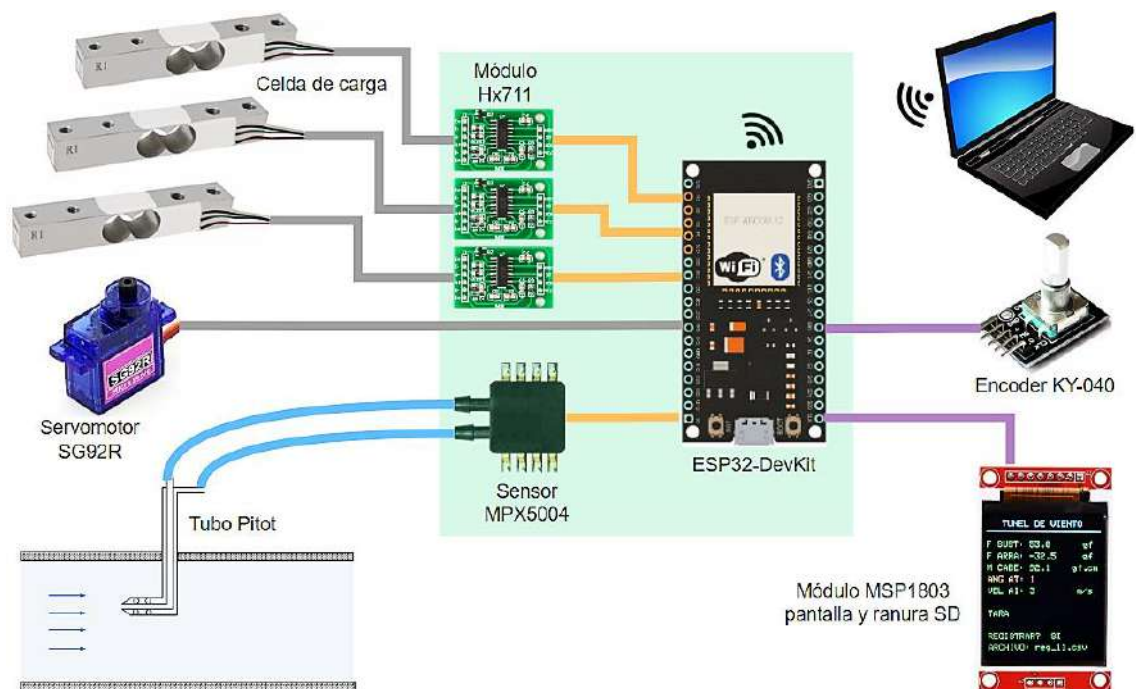


Figura 37: Esquema de conexiones entre los componentes del sistema de instrumentación y el sistema de adquisición y presentación de datos.

4.2.1 Diseño del diagrama esquemático

El diagrama esquemático se realizó en el editor de esquemáticos de KiCad. El proceso consistió en ubicar los componentes electrónicos sobre la hoja de trabajo y establecer las conexiones eléctricas entre tales componentes según las pautas establecidas en la Figura 37. En el anexo se muestra el diagrama esquemático completo del sistema de adquisición y presentación de datos. A continuación, se detallan las consideraciones más importantes del diseño.

componente se incorpora al diagrama esquemático y se establecen las conexiones eléctricas correspondientes.

La implementación de este circuito permite la comunicación efectiva entre el módulo Hx711 y la MCU ESP32 y asegura una correcta adaptación en los niveles de tensión de los dispositivo. La Figura 39 muestra el detalle del diagrama esquemático de uno de los módulos Hx711.

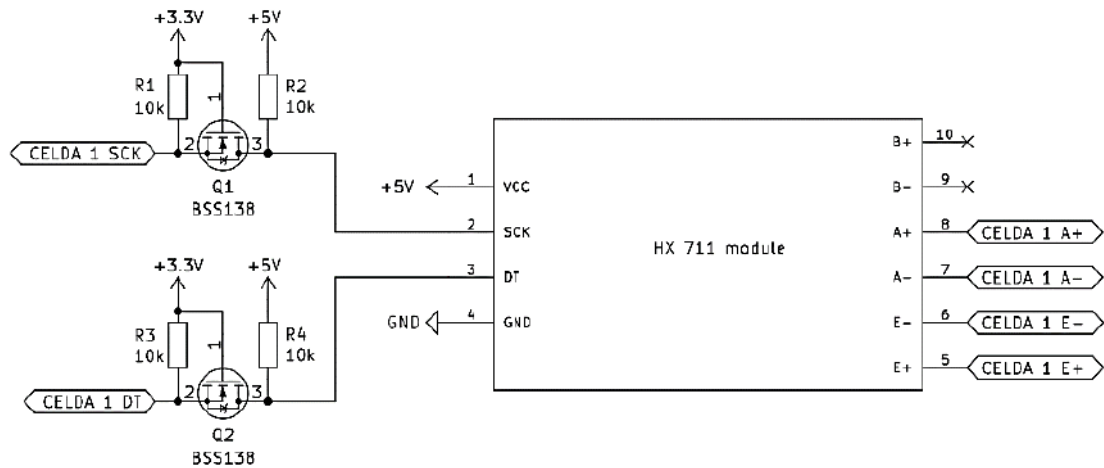


Figura 39: Detalle del diagrama esquemático del módulo Hx711.

Acondicionamiento de la señal analógica del sensor de presión MPX5004

Cada entrada ADC en la MCU ESP32 permite configurar un valor de atenuación que define el rango de tensión a medir. La Tabla 6 indica las atenuaciones con sus rangos de medición. Más allá de la atenuación seleccionada, el valor máximo admisible en la entrada ADC es de 3,3 VDC, un valor mayor a este puede causar daños.

Tabla 6: Atenuaciones disponibles del ADC de la MCU ESP32 con sus rangos medibles.

<i>Atenuación</i>	<i>Rango medible</i>
<i>0 dB</i>	<i>[0 a 1,1] VDC</i>
<i>-2,5 dB</i>	<i>[0 a 1,5] VDC</i>
<i>-6 dB</i>	<i>[0 a 2,2] VDC</i>
<i>-11 dB</i>	<i>[0 a 3,3] VDC</i>

El sensor MPX5004 utilizado para medir la presión del tubo Pitot provee una tensión menor a 2 VDC en condiciones de máxima velocidad del flujo de aire en el túnel de

viento. Sin embargo, ante un funcionamiento anormal el sensor podría proveer una tensión de hasta 5 VDC y, por ende, causar daños en la entrada ADC de la MCU ESP32. Para asegurar la integridad de la entrada ADC se agrega un divisor resistivo y un circuito seguidor entre la salida del sensor MPX5004 y la entrada ADC de la MCU ESP32, como se muestra en la Figura 40. El divisor resistivo permite atenuar la salida del sensor de presión, de modo que la tensión de entrada en el ADC de la MCU ESP32 sea siempre inferior a la admisible (3,3 VDC). Asimismo, el circuito seguidor provee una adaptación de impedancias entre el divisor resistivo y la entrada ADC de la MCU ESP32. Para el circuito seguidor se utiliza el amplificador operacional MCP6002 [16], que es alimentado con 3,3 VDC. El resto de los elementos pasivos ilustrados en el diagrama esquemático de la Figura 40 forman parte de filtros recomendados en las hojas de datos del sensor [10], el amplificador operacional [16] y la MCU ESP32 [17].

La elección apropiada de las resistencias R16 y R17, junto con la elección de la atenuación en la entrada ADC de la MCU ESP32 (Tabla 6), permiten proteger la entrada ADC y ajustar la señal al rango medible. En este caso se selecciona $R16 = R17$, por lo que el divisor resistivo reduce a la mitad el valor de la salida del sensor MPX5004. De esta manera, aun cuando la salida del sensor puede alcanzar los 5 VDC, en la entrada ADC de la MCU ESP32 solo se tendrá 2,5 VDC (un valor menor al máximo admisible de 3,3 VDC). Asimismo, en condiciones de máxima velocidad del flujo de aire en el túnel de viento, en la entrada ADC de la MCU ESP32 se tendrá una tensión menor a 1 VDC. Por lo tanto, se selecciona una atenuación de 0 dB (Tabla 6) en la entrada ADC de la MCU ESP32, ya que permite tener un buen ajuste del rango medible (0 a 1,1 VDC) al valor máximo que provee el sensor en túnel viento (un valor menor a 1 VDC).

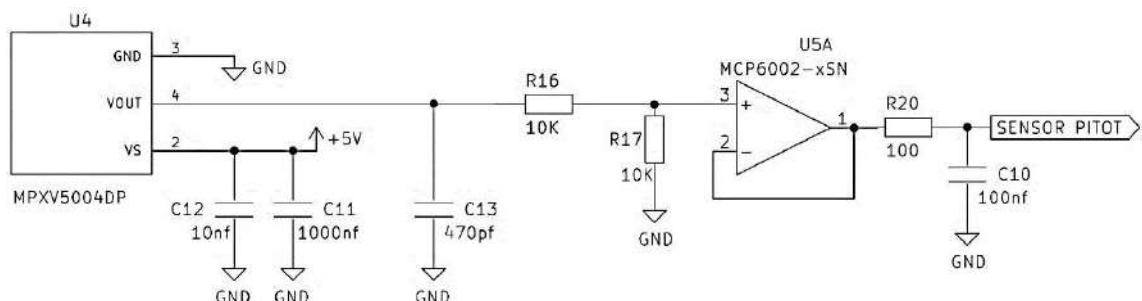


Figura 40: Detalle del diagrama esquemático del sensor MPX5004.

Fuente de alimentación

Por último, resulta necesario incorporar una fuente de alimentación en la placa PCB para cumplir con las demandas de tensión de 3,3 y 5 VDC. Para este propósito, se utiliza un diseño que consta de un regulador de tensión LM2576 [18] para proporcionar una tensión de 5 VDC y un regulador TC1262 [19] para proporcionar una tensión de 3,3 VDC. El circuito se diseña conforme a las especificaciones proporcionadas en las hojas de datos de los componentes utilizados. La incorporación de esta fuente de alimentación permite que el sistema opere con un rango de tensión de suministro comprendido entre 12 y 24 VDC. De este modo se garantiza que todos los componentes del circuito trabajen con los niveles de tensión adecuados para su funcionamiento. El detalle del diagrama esquemático de la fuente de alimentación se muestra en la Figura 41.

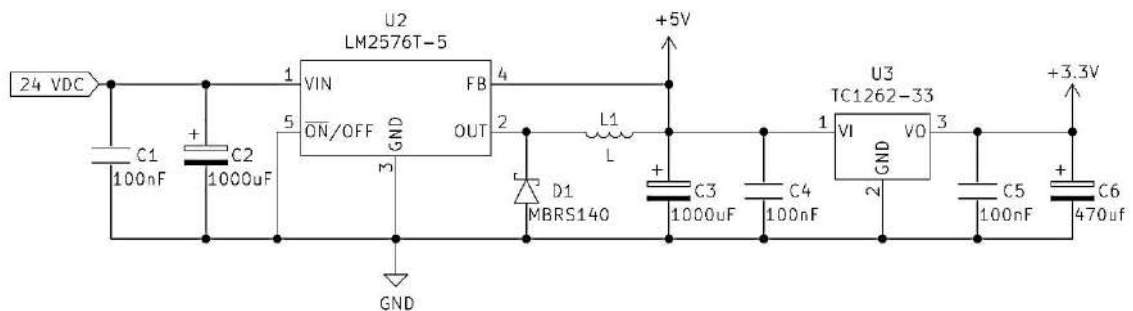


Figura 41: Detalle del diagrama esquemático de la fuente de alimentación.

4.2.2 Diseño y ensamble de la placa

El diseño de la placa PCB se realizó en el editor de placas de KiCad. En el diseño se consideraron aspectos como la disposición física de los componentes, las rutas de conexión eléctrica (pistas) y la optimización del espacio. Además, el correcto enrutamiento de las pistas requirió la minimización de interferencias electromagnéticas, la separación adecuada entre pistas y la reducción de la longitud de las pistas; la consideración de todos estos requerimientos repercute en la calidad e integridad de las señales. Completada la etapa de diseño se contrató un servicio de fabricación de la placa PCB. Con la recepción de la placa PCB fabricada se soldaron los componentes. Por último, la placa PCB fue ensamblada dentro de un gabinete diseñado en SolidWorks y fabricado mediante impresión 3D.

La Figura 42 presenta la cara frontal de la placa PCB con todos los componentes soldados. La figura muestra en la parte superior izquierda de la placa un conector para alimentación con un rango de entrada de 12 a 24 VDC. Justo debajo del conector se encuentra la fuente de alimentación que proporciona 3,3 y 5 VDC. En el centro de la placa PCB se encuentra la placa ESP32-DevKit y en la parte derecha de la placa están ubicados los módulos Hx711, el sensor MPX5004 y las borneras RJ45 para la conexión externa de las celdas de carga y el servomotor. Además, la placa PCB dispone de 3 borneras internas para la conexión cableada del encoder KY-040 y del módulo MSP1803 de pantalla y ranura SD.

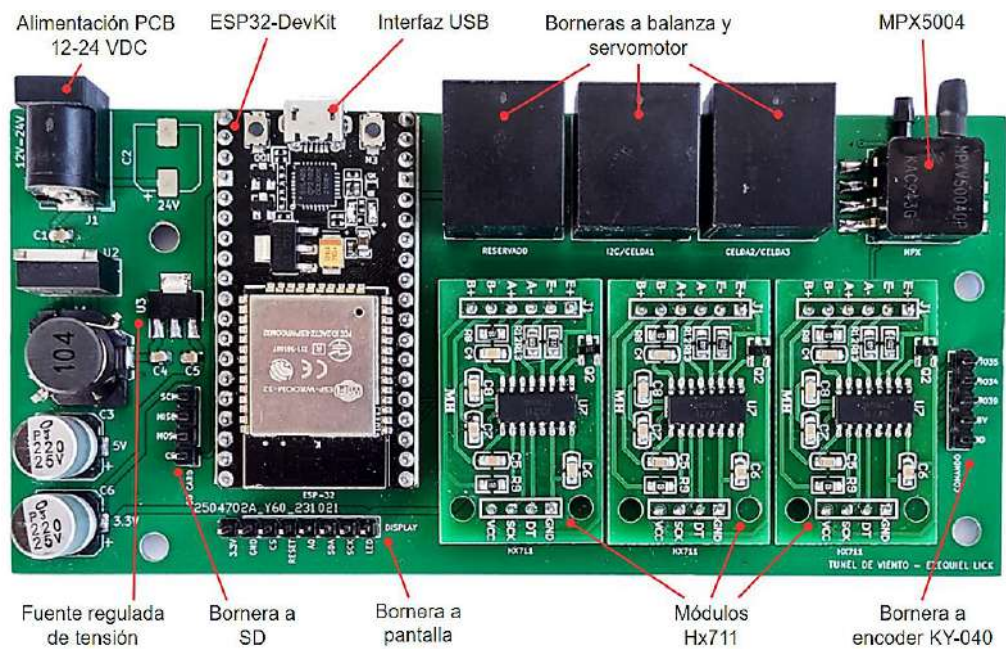


Figura 42: Cara frontal de la placa PCB con componentes soldados.

La Figura 43 presenta la cara posterior de la placa PCB con todos los componentes soldados. En esta cara de la placa PCB se ubica el circuito acondicionador de la señal analógica del sensor MPX5004, así como los adaptadores de niveles lógicos para la comunicación serial entre los módulos Hx711 y la MCU ESP32.

La Figura 44 presenta la parte interior de la tapa del gabinete, que tiene fijados el módulo MSP1803 de pantalla y ranura SD y el encoder KY-040. Estos componentes se conectan mediante cables a las 3 borneras internas de la placa PCB.

La Figura 45 exhibe el gabinete armado e instalado en una de las placas de acrílico del túnel de viento. La figura muestra la utilización de 3 cables de red para la comunicación entre el sistema de instrumentación y la placa PCB.

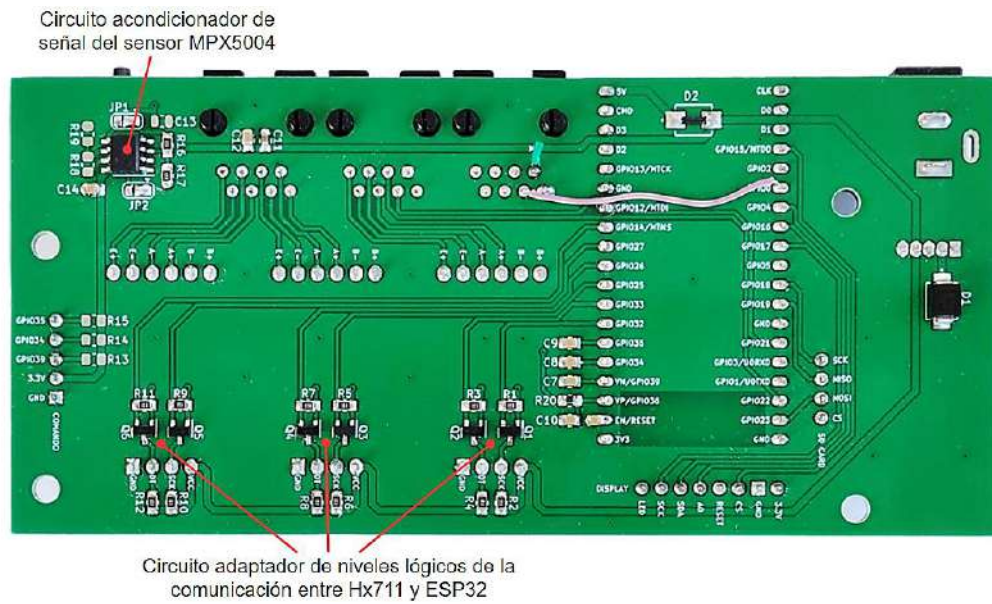


Figura 43: Cara posterior de la placa PCB con componentes soldados.



Figura 44: Ensamble de componentes sobre la tapa del gabinete.



Figura 45: Gabinete ensamblado final.

4.3 Interfaz de usuario

La interfaz con el usuario se realiza por dos vías, desde un panel principal en el gabinete y desde un panel web en una computadora remota. Las funcionalidades requeridas para estas dos vías forman parte del programa desarrollado para la MCU ESP32 mediante las herramientas integradas de Visual Studio y PlatformIO. La Figura 46 presenta el diagrama de flujo que describe el proceso de ejecución del programa implementado.

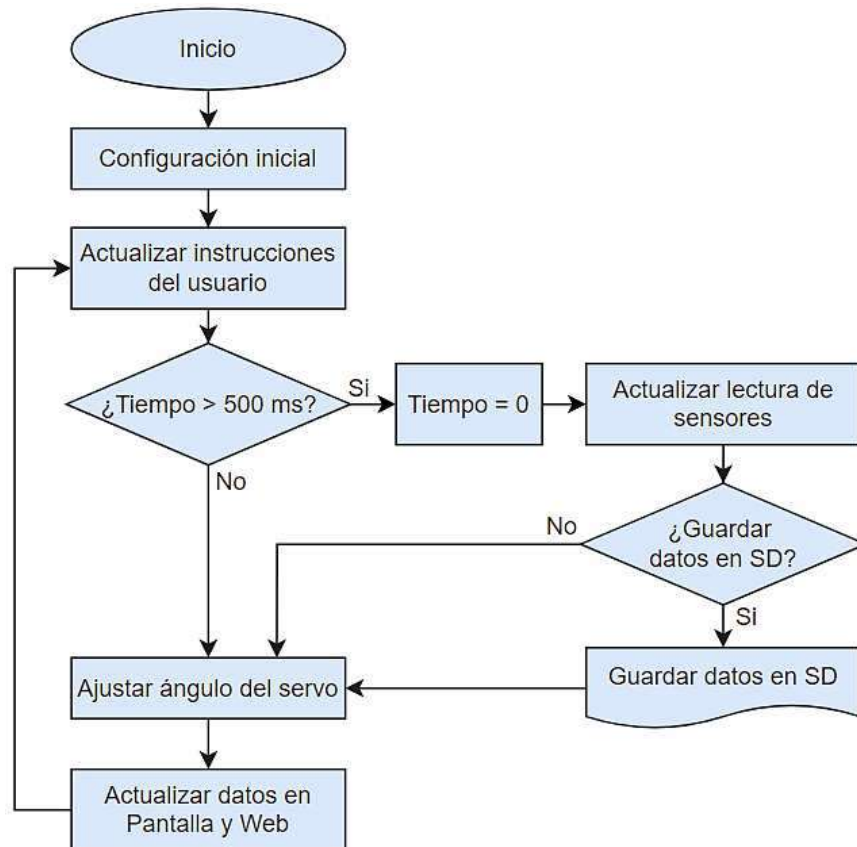


Figura 46: Diagrama de flujo del proceso de ejecución del programa.

La Figura 46 muestra que el proceso de ejecución del programa comienza con la configuración inicial de periféricos y protocolos de comunicación, seguido por la configuración de un punto de acceso WiFi y un servidor web. Luego se activa una tarea en bucle que comienza con la actualización de las instrucciones emitidas por el usuario. A continuación, se verifica el tiempo transcurrido desde la última actualización de las lecturas de los sensores. Si este tiempo excede los 500 ms, se actualizan nuevamente las lecturas de los sensores y se almacenan los datos en la tarjeta SD, siempre que el registro

de datos esté activado. La tarea concluye con el ajuste del ángulo del servomotor y la actualización de los datos en la pantalla y en el sitio web.

4.3.1 Panel principal

El panel principal muestra en pantalla los valores de la fuerza de sustentación, la fuerza de arrastre, el momento de cabeceo, la velocidad del aire, el ángulo de ataque y el estado de almacenamiento de los datos en la tarjeta SD. La Figura 47 ilustra una vista de la pantalla con el sistema en funcionamiento.



Figura 47: Vista de la pantalla del panel principal.

La rotación de la perilla del encoder permite al usuario navegar por las opciones del menú que se presenta en pantalla. La opción seleccionada se indica con el texto en tono naranja. En este menú existen tres posibles ajustes: modificar el ángulo de ataque, realizar un ajuste de cero y activar/desactivar el almacenamiento de datos en la tarjeta SD.

Para modificar el ángulo de ataque se selecciona la opción "ANG AT" y luego se presiona la perilla del encoder. El ángulo se muestra en tono rojo, que indica que el ajuste está activo. En esta situación, se gira la perilla, con saltos de 1°, para ajustar el ángulo de ataque en valores comprendidos en el rango de -20° a 20° de la horizontal. Para salir de este ajuste se pulsa nuevamente la perilla del encoder.

Para realizar un ajuste de cero se selecciona la opción de "AJUSTE DE CERO" y luego se presiona la perilla del encoder. De este modo, se reinician las medidas de los sensores (valor 0), opción que resulta útil antes de iniciar un nuevo ensayo.

Por último, para activar el almacenamiento de datos en la tarjeta SD se selecciona la opción "REGISTRAR?" y luego se presiona la perilla del encoder. Una vez activado, en el renglón siguiente se muestra el nombre del archivo actual. Para detener el almacenamiento basta con seleccionar nuevamente la opción "REGISTRAR?" y presionar la perilla del encoder. Esta funcionalidad permite registrar la fuerza de sustentación, la fuerza de arrastre, el momento de cabeceo, el ángulo de ataque y la velocidad del aire en una planilla de cálculo para una visualización y análisis posterior de los resultados del ensayo.

4.3.2 Panel web

El panel web, al igual que la pantalla, muestra lecturas actualizadas de los sensores y permite modificar el ángulo de ataque, realizar un ajuste de cero y activar/desactivar el almacenamiento de datos en la tarjeta SD (Figura 48). El acceso al panel web se realiza desde un dispositivo remoto, que se conecta a la MCU ESP32 con una conexión WiFi desde un navegador web (dirección IP 192.168.4.1).

El panel web ofrece la capacidad de visualizar gráficos en tiempo real de las fuerzas aerodinámicas en función de la velocidad del aire y/o del ángulo de ataque. La representación gráfica le brinda al usuario facilidades para tomar decisiones en tiempo real.

El panel web, representado en la Figura 48, está organizado en cuatro secciones. La primera sección permite activar/desactivar el almacenamiento de datos en la tarjeta SD. La segunda sección proporciona las opciones de ajustar el ángulo de ataque y realizar un ajuste de cero. En la tercera sección del panel web se presentan los valores de la fuerza de sustentación, la fuerza de arrastre, el momento de cabeceo y la velocidad del aire. Por último, la cuarta sección contiene dos gráficos; el gráfico superior muestra la evolución de las fuerzas aerodinámicas y el gráfico inferior representa la velocidad del aire y el ángulo de ataque. Esta disposición de los datos facilita la distinción entre variables dependientes y variables de control; el eje horizontal de los gráficos representa el número de lecturas de los sensores, realizadas cada 500 ms.

Registro de datos en SD

Nombre de archivo: -

Iniciar registro

Ángulo de ataque y ajuste de cero

Ángulo de ataque: 1 - +

Ajuste de cero

Última lectura de los sensores

Fuerza de sustentación: 26.66 [g]

Fuerza de arrastre: 18.72 [g]

Momento de cabeceo: 102.34 [g.cm]

Velocidad del aire: 33 [m/seg]

Gráfico

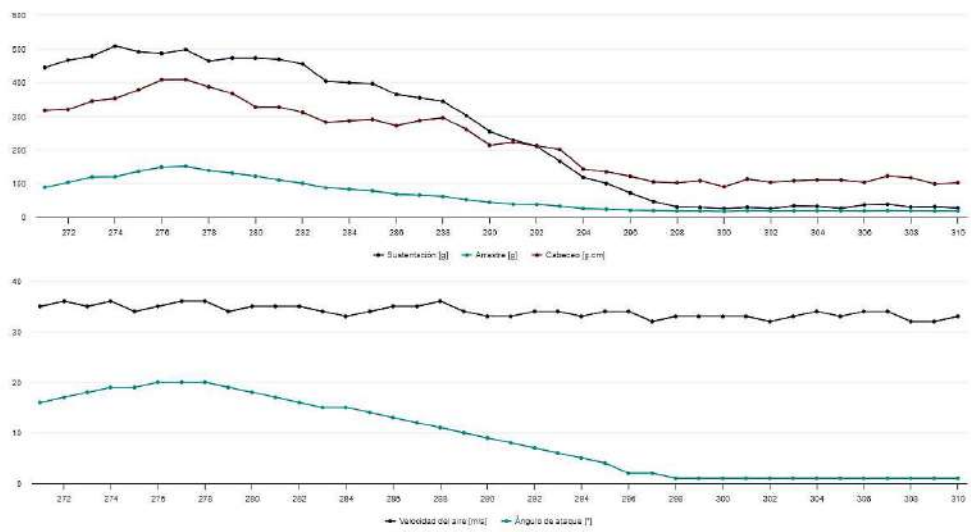


Figura 48: Vista del panel web.

Capítulo 5: Calibración y puesta en marcha

En este capítulo se exponen las tareas experimentales realizadas para la calibración de los instrumentos y la posterior puesta en marcha del sistema desarrollado. En primer lugar, se presentan las tareas experimentales implicadas en la calibración de la balanza aerodinámica. Se describe el procedimiento de calibración, se muestran las curvas obtenidas y se hace un análisis del acoplamiento de fuerzas. Luego, se comparan los resultados obtenidos en la medición de la velocidad del aire entre el sensor de presión diferencial que se utiliza en este proyecto y un transmisor de presión diferencial de la marca Dwyer. Por último, se muestran los resultados experimentales obtenidos en los ensayos realizados con perfil de ala NACA 0012 y NACA 2412.

5.1 Calibración de la balanza aerodinámica

Para calibrar la balanza aerodinámica se relevan las curvas que relacionan las fuerzas de sustentación, de arrastre y el momento de cabeceo con las salidas, en *cuentas*, que proveen los módulos de acondicionamiento de señal de las celdas de carga (módulos Hx711).

Las curvas que relacionan las tres salidas de la balanza con la fuerza de sustentación se obtienen mediante la aplicación de cargas en la dirección de F_L sobre la varilla C (Figura 49), mientras se registran las salidas de los tres módulos Hx711. La Figura 49 (a) muestra el arreglo utilizado para aplicar las cargas que corresponden a una fuerza de sustentación negativa. Para aplicar cargas equivalentes de fuerza de sustentación positiva se gira la balanza 180° respecto de su posición normal y se aplica peso sobre la varilla C, como se muestra en la Figura 49 (b). De esta manera se evita utilizar un sistema de poleas para aplicar cargas de sustentación positiva, ya que pequeñas fuerzas de fricción en las poleas pueden producir errores relativamente grandes. Sin embargo, es necesario ajustar el corrimiento en las mediciones obtenidas con la balanza invertida, debido a la diferencia en la carga que soporta cada celda por el propio peso de la balanza en esta posición, en comparación con su posición normal.

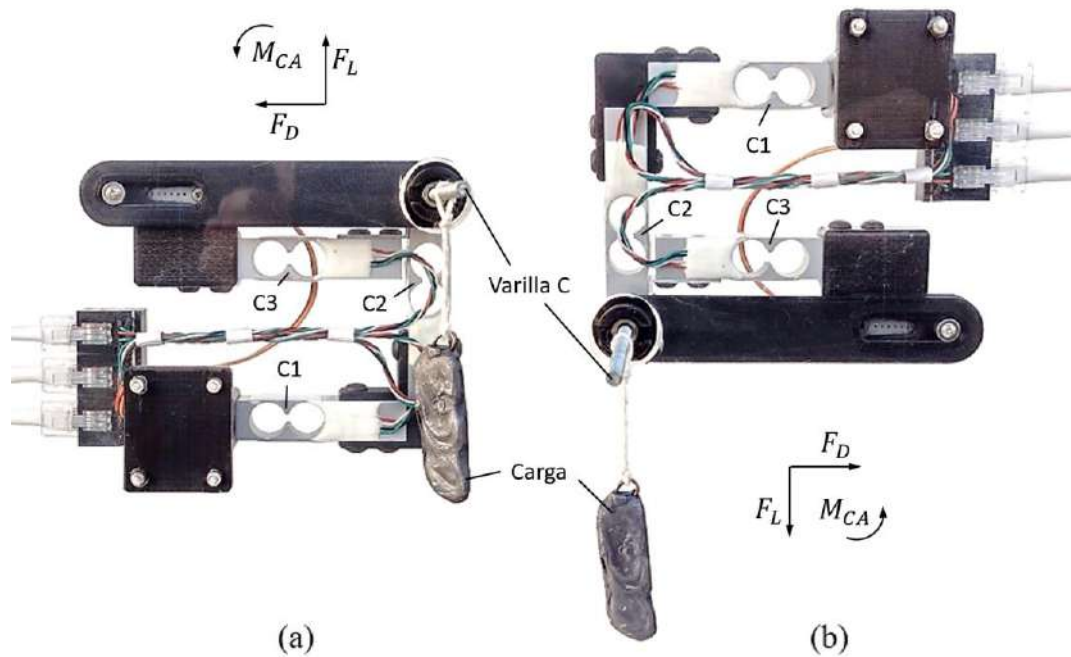


Figura 49: Disposición de la balanza para calibración por fuerza de sustentación.

De manera análoga, las curvas que relacionan la fuerza de arrastre con cada una de las salidas se obtienen con la aplicación de diferentes cargas en la dirección de F_D sobre la varilla C, mientras se registran las salidas de los módulos Hx711. Para generar cargas equivalentes a fuerza de arrastre se gira la balanza 90° respecto de su posición natural y se aplica peso en la varilla C como se muestra en la Figura 50.

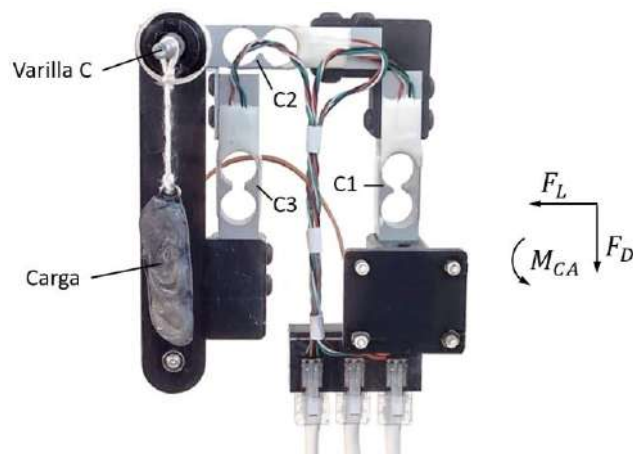


Figura 50: Disposición de la balanza para calibración por fuerza de arrastre.

Por último, las curvas que relacionan las tres salidas de la balanza con el momento de cabeceo se obtienen mediante la aplicación de diferentes momentos sobre la palanca F,

mientras se registran las salidas de los tres módulos Hx711. La Figura 51 muestra el arreglo utilizado para aplicar momento de cabeceo positivo mediante una herramienta que actúa como brazo de palanca. La carga se mantiene constante y se varía la distancia de aplicación de la carga respecto de la varilla C para generar diferentes momentos. Para aplicar momento negativo se gira la herramienta 180° y se aplica peso a diferentes distancias de la varilla C. El proceso se realiza para ángulo de ataque nulo y ángulo de ataque de 20°.

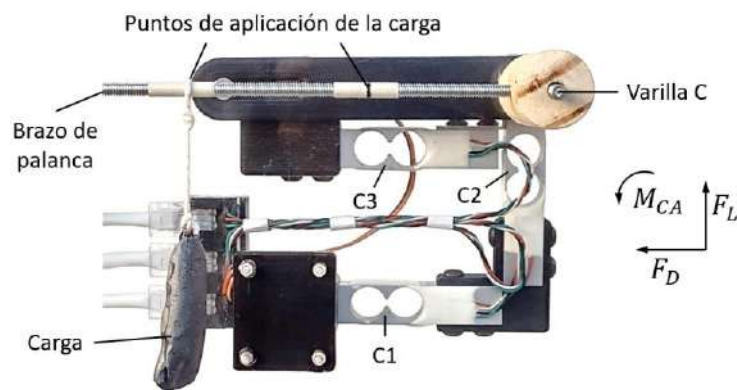


Figura 51: Disposición de la balanza para calibración por momento de cabeceo.

Con el propósito de facilitar la comprensión de los resultados que se exponen a continuación, se identifica como S_L a la salida del módulo Hx711 que provee la medición de la fuerza de sustentación (salida asociada a la celda C1), como S_D a la salida que provee la medición de la fuerza de arrastre (salida asociada a la celda C2) y como S_M a la salida que provee la medición del momento de cabeceo (salida asociada a la celda C3).

5.1.1 Resultados de la calibración

La Figura 52 muestra la respuesta de las 3 salidas de la balanza a las fuerzas de sustentación, mientras que la Figura 53 muestra la respuesta a las fuerzas de arrastre. De manera similar, la Figura 54 y la Figura 55 muestran la respuesta de las salidas de la balanza al momento de cabeceo, para los ángulos de ataque de 0° y 20° respectivamente. En las cuatro figuras también se presentan las ecuaciones de las rectas de ajuste lineal de las respuestas de cada una de las salidas a las fuerzas y momentos aplicados.

Coefficientes de sensibilidad a las fuerzas de sustentación y de arrastre

La Figura 52 muestra que la fuerza de sustentación se puede determinar a partir de la ecuación $y = -2091,1 \cdot x + 485010$ donde y es la salida S_L medida en cuentas, y x es la fuerza de sustentación aplicada expresada en gramos. El coeficiente $-2091,1$ [cuentas/g] (pendiente de la recta) representa la sensibilidad de la salida S_L respecto de la fuerza de sustentación y , por lo tanto, es la razón de cambio de la salida S_L respecto de la fuerza F_L . El coeficiente 485010 [cuentas] (término independiente) representa el corrimiento de cero en la salida S_L .

De manera análoga, la Figura 53 muestra que la fuerza de arrastre se puede obtener a partir de la ecuación $y = -1937,4 \cdot x + 64454$ donde y es la salida S_D medida en cuentas, y x es la fuerza de arrastre aplicada expresada en gramos. El coeficiente $-1937,4$ [cuentas/g] (pendiente de la recta) representa la sensibilidad de la salida S_D respecto de la fuerza de arrastre y , por lo tanto, es la razón de cambio de la salida S_D respecto de la fuerza F_D . El coeficiente 64454 [cuentas] (término independiente) representa el corrimiento de cero en la salida S_D .

Coefficiente de sensibilidad del momento de cabeceo en función de alfa

La Figura 54 muestra que el momento de cabeceo para ángulo de ataque $\alpha = 0^\circ$ se puede determinar a partir de la ecuación $y = -191,3 \cdot x - 132698$ donde y es la salida S_M medida en cuentas, y x es el momento de cabeceo aplicado, expresado en gramos por centímetro. De manera similar, la Figura 55 muestra que el momento de cabeceo para $\alpha = 20^\circ$ se puede obtener a partir de la ecuación $y = -231,6 \cdot x - 132708$.

La diferencia entre los coeficientes de sensibilidad para $\alpha = 0^\circ$ ($-191,3$ [cuentas/g · cm]) y para $\alpha = 20^\circ$ ($-231,6$ [cuentas/g · cm]) se debe a que, aunque el momento de cabeceo aplicado es el mismo, la carga que mide la celda C3 es función de α , cómo se explicó en el apartado 3.3.3. Por lo tanto, para el mismo momento de cabeceo la salida S_M produce valores diferentes para ángulos de ataque α diferentes.

De la ecuación (11) resulta que dado un M_{CA} , la carga medida por la celda C3 es $F_{C3} = M_{CA} \cdot \cos(\alpha)/r_1(\alpha)$, donde $r_1(\alpha)$ se puede calcular con la ecuación (15). Por lo tanto, dado un momento de cabeceo cualquiera, tal que $M_{CA}(\alpha) = M_{CA}(0) = M_{CA}$, se puede obtener una expresión para el coeficiente de sensibilidad de la salida S_M en función

de α . La salida de la celda C3 es proporcional a la carga F_{C3} ($S_M \propto F_{C3}$) por lo tanto resulta la siguiente igualdad:

$$\frac{S_M(\alpha)}{S_M(0)} = \frac{M_{CA}(\alpha) \cdot \cos(\alpha)/r_1(\alpha)}{M_{CA}(0) \cdot \cos(0)/r_1(0)} \quad (20)$$

Dado que $M_{CA}(\alpha) = M_{CA}(0) = M_{CA}$ y reacomodando la ecuación (20) resulta:

$$\frac{S_M(\alpha)}{M_{CA}} = \frac{S_M(0) \cdot \cos(\alpha)/r_1(\alpha)}{M_{CA}(0) \cdot \cos(0)/r_1(0)} \quad (21)$$

del resultado de calibración para $\alpha = 0^\circ$, se obtuvo que $S_M(0)/M_{CA}(0) = -191,3$, si se reemplaza este resultado en la ecuación (21) resulta:

$$\frac{S_M(\alpha)}{M_{CA}} = \frac{-191,3 \cdot \cos(\alpha)/r_1(\alpha)}{\cos(0)/r_1(0)} = \frac{-191,3 \cdot \cos(\alpha) \cdot 9,577}{r_1(\alpha)} \quad (22)$$

La ecuación (22) representa el coeficiente de sensibilidad de la salida S_M al momento de cabeceo M_{CA} para cualquier ángulo de ataque α . De esta manera, dado un momento M_{CA} la ecuación (22) permite obtener el valor de la salida S_M en función del ángulo de ataque. Si la ecuación (22) se resuelve para $\alpha = 20^\circ$ se obtiene $S_M(\alpha)/M_{CA} = -230,3$, que es aproximadamente igual al resultado encontrado experimentalmente ($-231,6$) para $\alpha = 20^\circ$.

Coefficientes de sensibilidad cruzada

La Figura 52 muestra que las salidas asociadas a la fuerza de arrastre (S_D) y al momento de cabeceo (S_M) permanecen casi inalteradas cuando se aplica fuerza de sustentación F_L en todo el rango ensayado. La relación entre S_D y F_L está representada por la ecuación $y = -4,7 \cdot x + 64526$, donde el coeficiente $-4,7$ [cuentas/g] representa la sensibilidad de la salida S_D a la fuerza F_L . Asimismo, la ecuación $y = -0,2 \cdot x - 132805$ representa la relación entre la fuerza de sustentación F_L y la salida asociada al momento de cabeceo S_M , donde el coeficiente $-0,2$ [cuentas/g] representa la sensibilidad de la salida S_M a la fuerza F_L .

De manera similar, de la Figura 53 resulta que el coeficiente $-8,7$ [cuentas/g] es la sensibilidad de la salida que mide la fuerza de sustentación S_L respecto de la fuerza de

arrastre F_D , y el coeficiente $0,3$ [cuentas/g] es la sensibilidad de la salida S_M respecto de la fuerza F_D . Por último, la Figura 54 y la Figura 55 muestran que para $\alpha = 0^\circ$ y $\alpha = 20^\circ$ las salidas asociadas con las fuerzas de sustentación y de arrastre (S_L y S_D) permanecen casi inalteradas cuando se aplica momento de cabeceo M_{CA} . La sensibilidad de la salida S_L respecto del momento M_{CA} es -0.3 [cuentas/g · cm] para $\alpha = 0^\circ$ y para $\alpha = 20^\circ$. La sensibilidad de la salida S_D respecto del momento M_{CA} es -0.4 [cuentas/g · cm] para $\alpha = 0^\circ$ y para $\alpha = 20^\circ$.

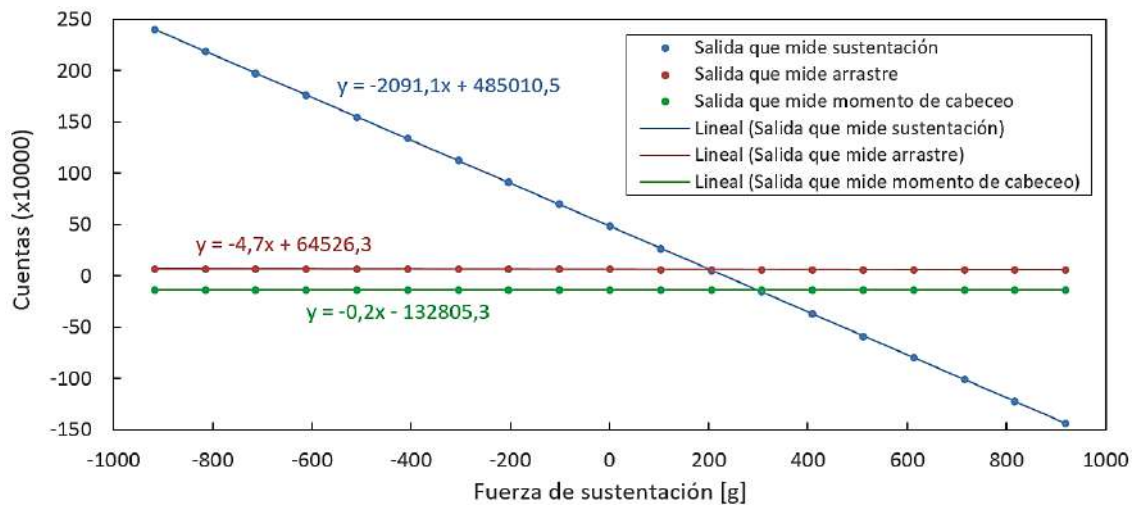


Figura 52: Respuesta de las salidas frente a fuerza de sustentación.

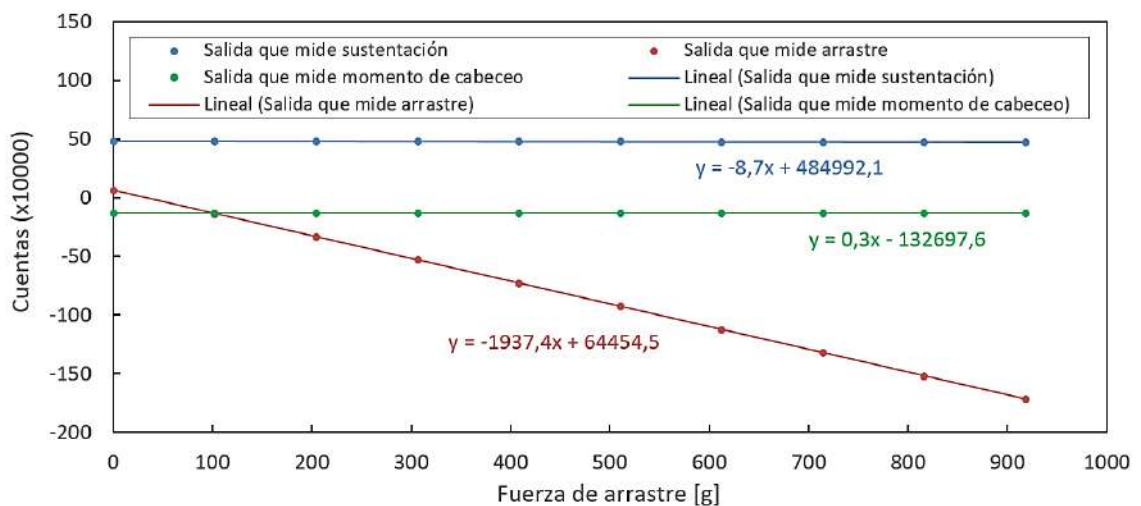


Figura 53: Respuesta de las salidas frente a fuerza de arrastre.

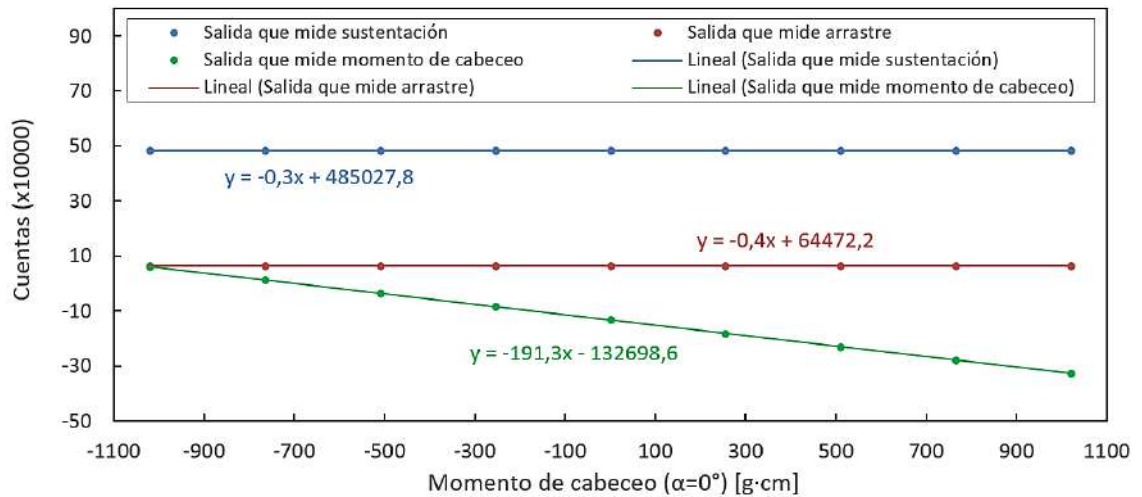


Figura 54: Respuesta de las salidas frente a momento de cabeceo para $\alpha=0^\circ$.

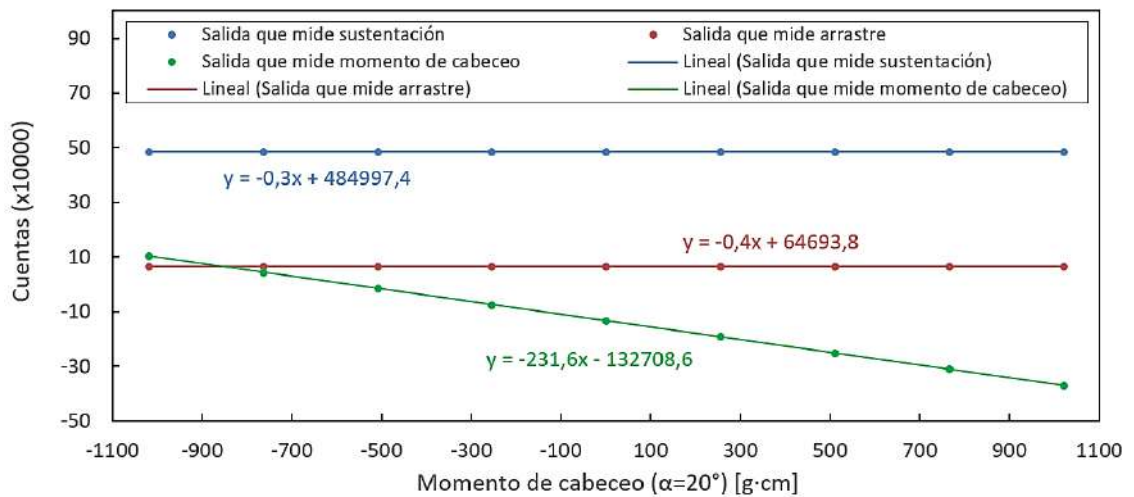


Figura 55: Respuesta de las salidas frente a momento de cabeceo para $\alpha=20^\circ$.

Modelo del comportamiento de las salidas de la balanza

Con los resultados obtenidos en la calibración de la balanza se propone el siguiente modelo del comportamiento de las salidas que proveen las medidas de la fuerza de arrastre, la fuerza de sustentación y el momento de cabeceo:

$$\begin{bmatrix} S_L \\ S_D \\ S_M \end{bmatrix} = \begin{bmatrix} -2091,1 & -8,7 & -0,3 \\ -4,7 & -1937,4 & -0,4 \\ -0,2 & 0,3 & -191,3 \cdot \cos(\alpha) \cdot 9,577/r_1(\alpha) \end{bmatrix} \begin{bmatrix} F_L \\ F_D \\ M_{CA} \end{bmatrix} + \begin{bmatrix} 485010 \\ 64454 \\ -132698 \end{bmatrix}$$

Por simple inspección de la matriz de los coeficientes de sensibilidad se puede ver que los coeficientes de la diagonal son varios órdenes de magnitud mayores a los coeficientes fuera de la diagonal. Eso indica el bajo impacto de cada una de las magnitudes de entrada

F_L , F_D y M_{CA} en las salidas de las dos celdas de carga que no están dedicadas a su medición. Por ejemplo, del cociente entre los coeficientes (1,2) y (1,1) ($8,7/2091,1$) resulta que el efecto de la fuerza F_D en la salida S_L es menor al 0,42%. De manera análoga, el cociente entre los coeficientes (2,1) y (2,2) ($4,7/1937,4$) indica que el impacto de la fuerza F_L en la salida S_D es menor al 0,25%. Con este razonamiento se puede comprobar el bajo impacto que tiene el efecto de la sensibilidad cruzada, lo que marca el fuerte desacople entre cada una de las salidas.

Comportamiento de la celda de carga de tipo punto único.

En la Figura 50 se ve que la fuerza de arrastre produce un momento sobre la celda de carga C1. Sin embargo, en el apartado anterior se calculó que el impacto de la fuerza de arrastre en la salida de la celda C1 (S_L) es menor al 0,42%, esto se debe a que la salida de la celda C1 es insensible a los momentos de fuerza sobre la misma. Asimismo, la Figura 51 muestra que la fuerza asociada con el momento de cabeceo, que actúa sobre la celda C3, produce un momento sobre las celdas C1 y C2, sin embargo, con la matriz de coeficientes, se puede comprobar que el impacto del momento de cabeceo en las salidas S_L y S_D es despreciable. Esto solo es posible si las celdas C1 y C2 son insensibles a los momentos de fuerza. De esta manera se confirma, de manera experimental, lo expresado en el apartado 3.2.1 acerca del comportamiento de las celdas de carga de punto único respecto de los momentos de fuerza.

5.2 Verificación del sensor de velocidad del aire

El sensor de presión diferencial MPX5004 posee una salida calibrada de fábrica cuya curva se muestra en la Figura 33. Por lo tanto, en este caso no se requiere un procedimiento riguroso de calibración, como en el caso de la balanza. Sin embargo, para garantizar la efectividad del sistema desarrollado resulta necesario comparar los valores de velocidad de flujo obtenidos a partir del sensor MPX5004 y los valores provistos por el transmisor de presión diferencial Dwyer serie MS Magnesense [20] (Figura 56). Tanto el sensor MPX5004, como el transmisor Dwyer, se conectan a la misma toma de presión del tubo Pitot para garantizar una buena comparación.



Figura 56: Transmisor Dwyer Magnesense.

La Figura 57 presenta los datos que relacionan las salidas de los dispositivos y la frecuencia configurada en el variador de velocidad de los ventiladores en túnel de viento. Se comprueba que en el rango de velocidades comprendido entre 5 y 35 m/s los valores provistos por ambos dispositivos son muy similares. Las pequeñas discrepancias en las medidas no representan un inconveniente significativo, ya que, el sistema desarrollado provee valores muy confiables en todo el rango de medición.

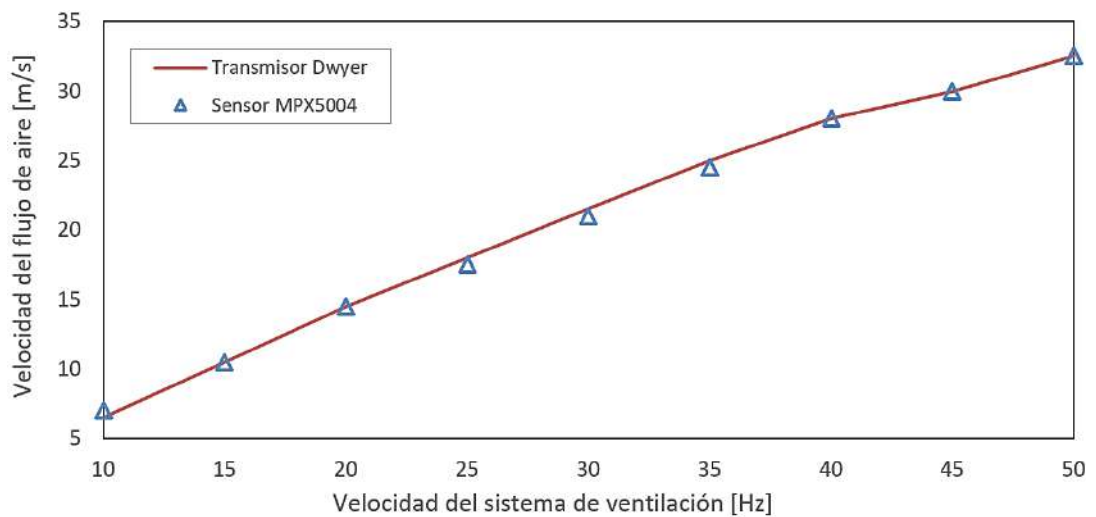


Figura 57: Respuesta de la salida del sensor MPX5004 y del transmisor Dwyer a diferentes velocidades del sistema de ventilación.

5.3 Montaje del sistema en túnel de viento

La Figura 58 presenta el sistema de instrumentación y adquisición de datos montado en el túnel de viento. La figura muestra a la balanza ubicada fuera de la cámara de ensayo, lo que minimiza la interferencia con el flujo de aire. El sistema de adquisición de datos está instalado sobre la misma placa de acrílico que la balanza y se utiliza un conducto flexible para conectar el tubo Pitot con el sensor de presión. Dado que todo el sistema está ubicado en la misma placa se evita la necesidad de cables largos entre la balanza y el sistema de adquisición de datos y, además, se facilita el montaje y desmontaje del sistema en el túnel de viento. La Figura 59 muestra dos vistas (frontal y lateral) de la cámara de ensayo con el perfil de ala NACA 2412 en su interior.

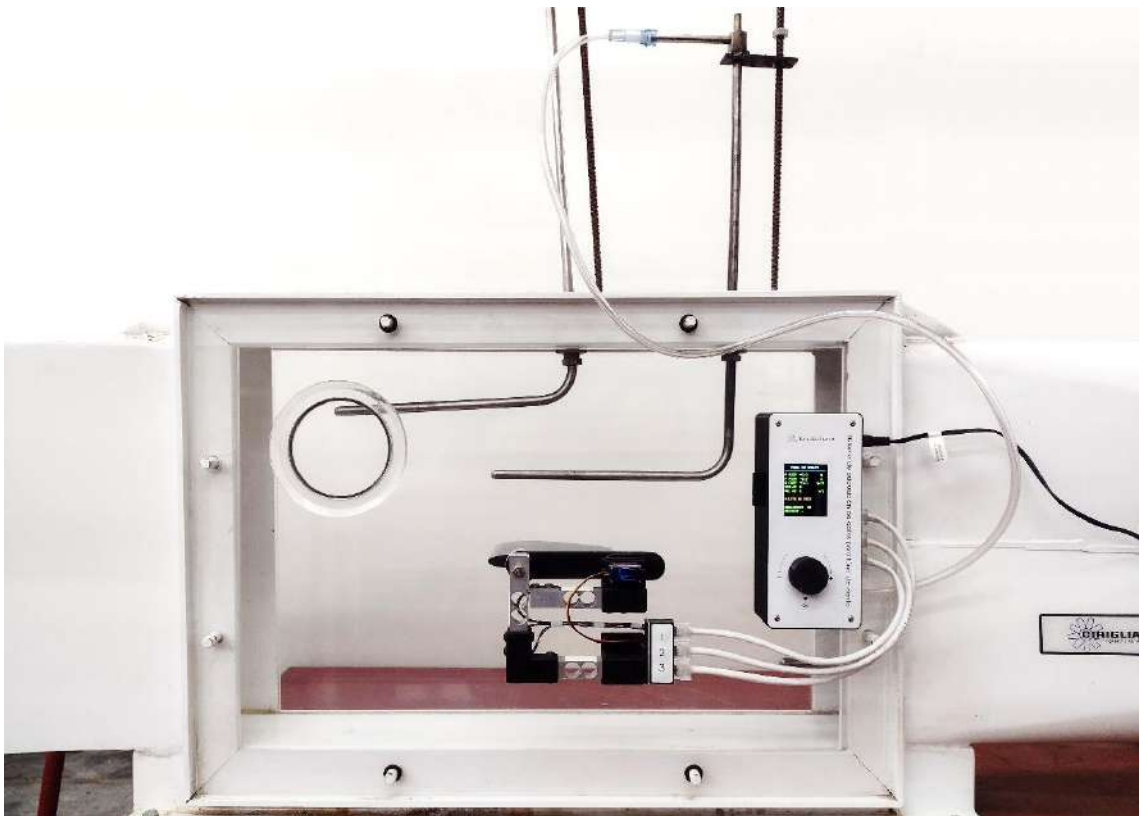


Figura 58: Sistema de instrumentación y adquisición de datos instalado en túnel de viento.



Figura 59: Vista frontal y lateral de la cámara de ensayo con el perfil de ala NACA 2412 en su interior.

5.4 Evaluación del funcionamiento del sistema

Para evaluar el sistema en su conjunto se ensayaron los perfiles de ala NACA 0012 [21] y NACA 2412 [22] (Figura 60). Los perfiles se diseñaron con el software SolidWorks y luego se fabricaron mediante impresión 3D a partir de los parámetros geométricos disponibles en el sitio web de Airfoil Tools [4]. Los perfiles fabricados presentan un orificio que se utiliza para soporte y, presentan además, una moldura que se utiliza para establecer la posición angular del perfil y para transmitir el momento de cabeceo.



Figura 60: Perfiles alares utilizados en la evaluación del sistema (a) NACA 0012 y (b) NACA 2412.

Ejecución y resultados de los ensayos

Realizar un ensayo requiere los siguientes pasos: (1) colocar el perfil de ala en la balanza, (2) insertar la tarjeta de memoria SD que permite registrar los datos, (3) encender el sistema, (4) ejecutar el procedimiento de ajuste de cero, (5) ajustar la velocidad deseada

del flujo de aire, (6) activar el registro de datos en la memoria SD, (7) variar el ángulo de ataque, y (8) detener el registro de datos en la memoria SD.

El ensayo del perfil NACA 0012 se realizó con una velocidad del aire $v = 15 \text{ m/s}$, por lo que de la ecuación (4) resulta un flujo con un número de Reynolds $Re = 100000$. Para el cálculo del número de Reynolds se considera además una densidad de aire $\rho = 1,225 \text{ kg/m}^3$, una longitud de cuerda $C = 0,1 \text{ m}$ y una viscosidad dinámica $\mu = 1,802 \times 10^{-5} \text{ kg/m} \cdot \text{s}$.

La Figura 61 muestra los resultados obtenidos en el ensayo del perfil NACA 0012 para los coeficientes C_L , C_D y C_M en función del ángulo de ataque α . La figura permite visualizar las características esperadas de un perfil simétrico: (1) el coeficiente de sustentación resulta nulo ante un ángulo de ataque nulo, (2) el coeficiente de sustentación presenta un comportamiento en forma anti simétrica (para un ángulo negativo, C_L tiene mismo valor y distinto signo que para un ángulo positivo), (3) el coeficiente de arrastre presenta un valor mínimo ante un ángulo de ataque nulo, (4) el coeficiente de arrastre presenta un comportamiento en forma simétrica (para un ángulo negativo, C_D tiene mismo valor e igual signo que para un ángulo positivo), y (5) el coeficiente de momento de cabeceo presenta un valor nulo ante un ángulo de ataque nulo.

El ensayo del perfil NACA 2412 se realizó con una velocidad del aire $v = 30 \text{ m/s}$, por lo que de la ecuación (4) resulta un flujo con un número de Reynolds $Re = 200000$. Para el cálculo del número de Reynolds se considera además una densidad de aire $\rho = 1,225 \text{ kg/m}^3$, una longitud de cuerda $C = 0,1 \text{ m}$ y una viscosidad dinámica $\mu = 1,802 \times 10^{-5} \text{ kg/m} \cdot \text{s}$.

En forma análoga, la Figura 62 muestra los resultados obtenidos en el ensayo del perfil NACA 2412 para los coeficientes C_L , C_D y C_M en función del ángulo de ataque α . La figura permite visualizar las características esperadas de un perfil asimétrico: (1) el coeficiente de sustentación resulta positivo ante un ángulo de ataque nulo, (2) el coeficiente de arrastre presenta un comportamiento en forma simétrica (para un ángulo negativo, C_D tiene mismo valor e igual signo que para un ángulo positivo), y (3) el coeficiente de momento de cabeceo resulta negativo para un ángulo de ataque nulo.

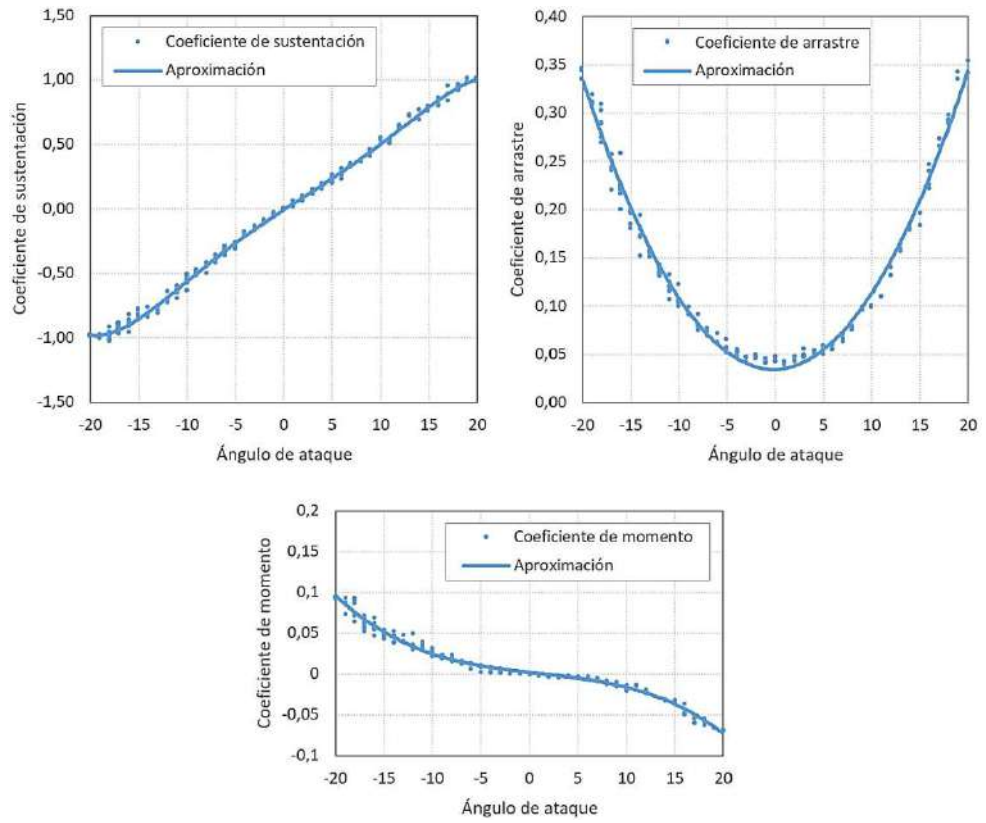


Figura 61: Resultados de la evaluación del perfil alar NACA 0012.

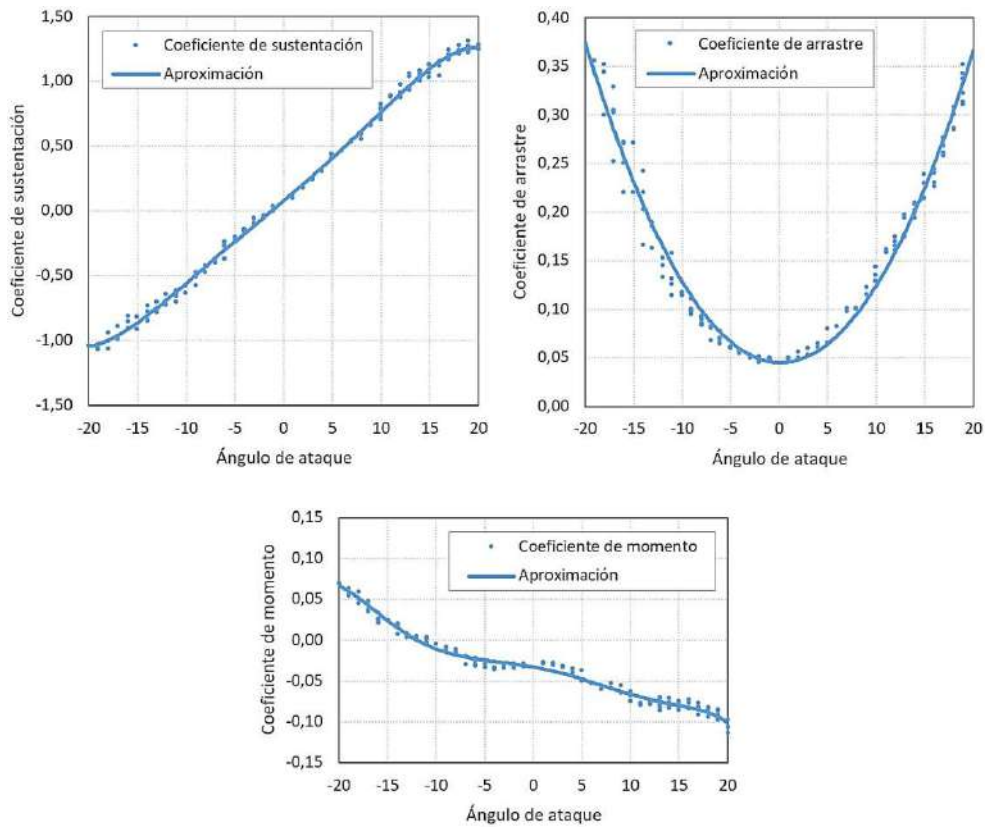


Figura 62: Resultados de la evaluación del perfil alar NACA 2412.

Capítulo 6: Conclusiones

En cuanto a la instrumentación, se puede mencionar que la balanza desarrollada se destaca por su robustez, bajos requisitos de mantenimiento, interacción reducida entre los componentes, interferencia despreciable en el flujo de aire y facilidad para el montaje y desmontaje del objeto de estudio. Además, el mecanismo para variar el ángulo de ataque ofrece una solución automatizada que amplía las posibilidades de ensayo de la balanza.

Los resultados obtenidos en la calibración de la balanza demuestran que las celdas de carga utilizadas presentan un comportamiento lineal en todo el rango de medición y permiten obtener un excelente desacople de las fuerzas. Asimismo, se comprobó que el sistema conformado por el tubo Pitot y el sensor de presión permite obtener una medición precisa de la velocidad del aire con bajos requisitos de mantenimiento.

Respecto al sistema de adquisición y presentación de datos, el desarrollo de la placa de circuito impreso resultó clave para asegurar la correcta conexión eléctrica entre los dispositivos electrónicos. Además, permitió integrar correctamente el circuito de adaptación de niveles lógicos de los módulos Hx711 y el circuito de acondicionamiento de la señal del sensor de presión.

En cuanto a la interfaz de usuario, el panel principal ofrece simplicidad para controlar el sistema, mientras que el panel web se destaca por los gráficos en tiempo real, permitiendo visualizar las variables en una pantalla remota. Además, el almacenamiento de datos en una unidad portátil brinda flexibilidad para el análisis posterior de los datos.

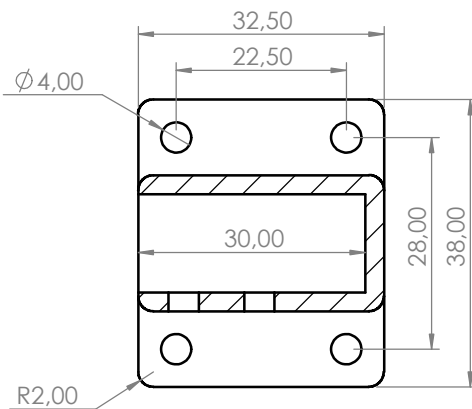
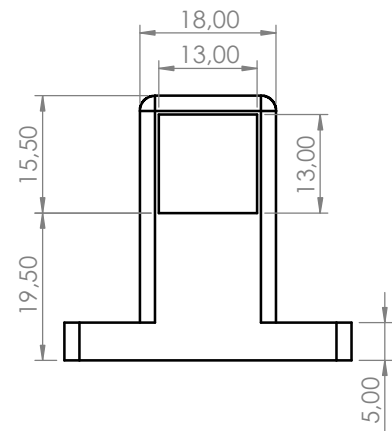
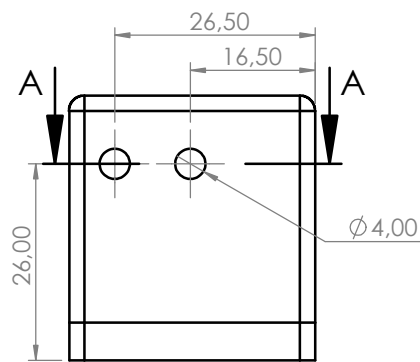
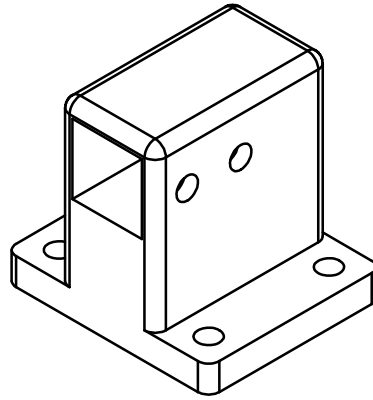
Por lo tanto, se puede concluir que el sistema de instrumentación y adquisición de datos para túnel de viento desarrollado cumple con los objetivos propuestos, ofreciendo una herramienta flexible y de fácil uso para futuras actividades didácticas y experimentales en el campo de la aerodinámica.

Bibliografía

- [1] A. R. Corpa, «Diseño de perfiles aerodinámicos mediante metodología inversa,» Sevilla, 2015.
- [2] Y. A. Cengel y J. M. Cimbala, Mecánica de los fluidos, México: McGraw-Hill Interamericana, 2006.
- [3] A. I. Carmona, Aerodinámica y actuaciones del avión, España: Paraninfo, 1999.
- [4] Airfoil Tools, <http://airfoiltools.com/index>.
- [5] Calculadora del numero de reynolds, <http://airfoiltools.com/calculator/reynoldsnumber>.
- [6] Celda de carga de un solo punto Flintek, <https://www.flintec.com/mx/sensores-de-peso/celulas-de-carga/c%C3%B3mo-funciona-una-celda-de-carga-de-un-solo-punto>.
- [7] Celda de carga de un solo punto, <https://www.800loadcel.com/blog/single-point-load-cells-what-they-are-and-how-they-work.html>.
- [8] AVIA Semiconductor, Datasheet Hx711, https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf.
- [9] TowerPro, Datasheet SG92R, <https://www.towerpro.com.tw/product/sg92r-7/>.
- [10] NXP Semiconductor, Datasheet MPXV5004, <https://www.nxp.com/docs/en/data-sheet/MPXV5004G.pdf>.
- [11] Espressif Systems, ESP32-DevKit Getting Started Guide, <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/hw-reference/esp32/get-started-devkitc.html#>.
- [12] Espressif Systems, Datasheet ESP32, https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

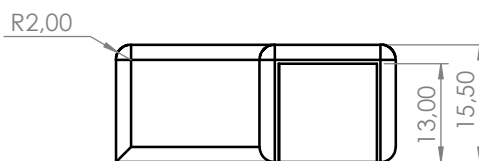
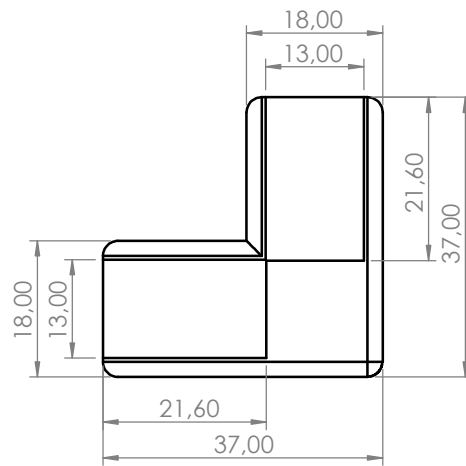
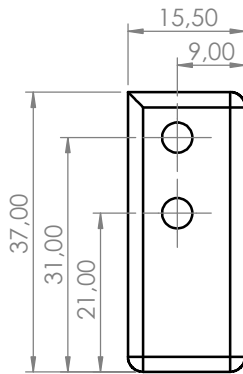
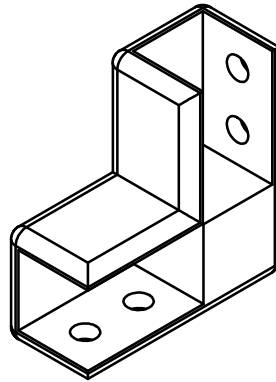
- [13] Datasheet modulo MSP1803, http://www.lcdwiki.com/res/MSP1803/1.8inch_SP_I_Module_MSP1803_User_Manual_EN.pdf.
- [14] Handson Technology, Datasheet KY-040, <https://www.handsontec.com/dataspecs/module/Rotary%20Encoder.pdf>.
- [15] Fairchild Semiconductor, Datasheet BSS138, <https://cdn.sparkfun.com/datasheets/BreakoutBoards/BSS138.pdf>.
- [16] Microchip, Datasheet MCP6002, <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP6001-1R-1U-2-4-1-MHz-Low-Power-Op-Amp-DS20001733L.pdf>.
- [17] Espressif Systems, Analog to Digital Converter ESP32, <https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32/api-reference/peripherals/adc.html>.
- [18] Texas Instrument, LM2576 Datasheet, <https://www.ti.com/lit/ds/symlink/lm2576hv.pdf>.
- [19] Microchip Technology, TC1262 Datasheet, <https://ww1.microchip.com/downloads/en/DeviceDoc/21373C.pdf>.
- [20] Differential pressure transmitter MSX Magnesense, https://intl.dwyer-inst.com/Products/Product.cfm?Group_ID=20672.
- [21] NACA 0012, <http://airfoiltools.com/airfoil/details?airfoil=n0012-il>.
- [22] NACA 2412, <http://airfoiltools.com/airfoil/details?airfoil=naca2412-il>.
- [23] F. M. White, Mecánica de fluidos, España: McGraw-Hill Interamericana, 2004.
- [24] A. M. Domingo, Apuntes de mecánica de fluidos, 2011.
- [25] J. M. Sanchez López, «Diseño de una balanza para tunel de viento subsónico,» Valencia, 2012.
- [26] Aerodinámica básica para pilotos, <https://greatbustardsflight.blogspot.com/2020/06/aerodinamica-basica-para-pilotos.html>.

Anexos

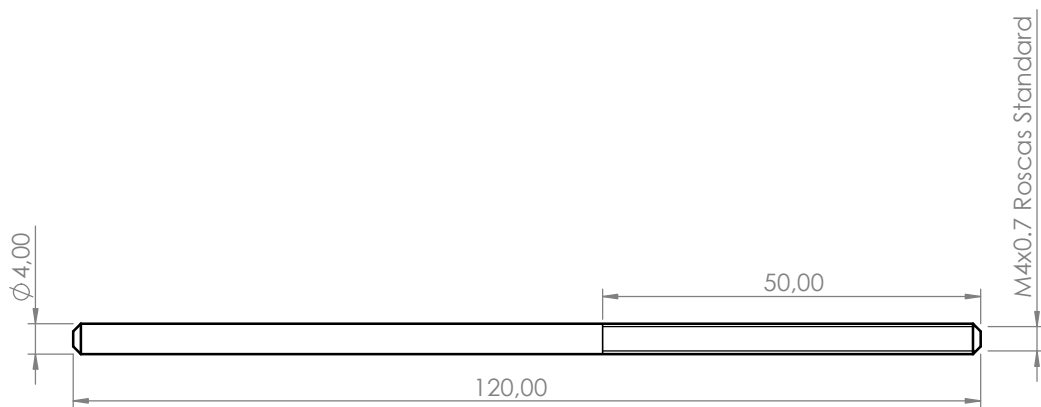
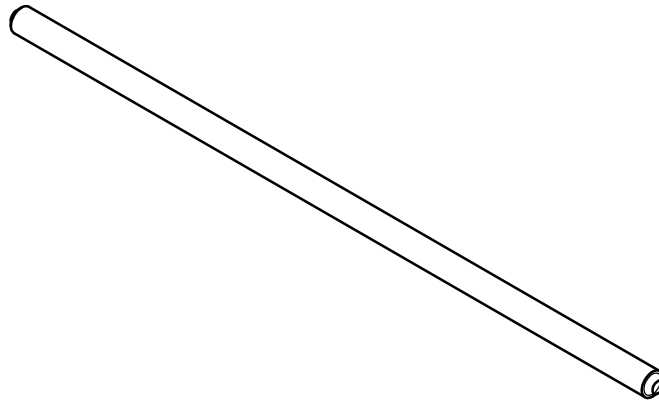


SECCIÓN A-A

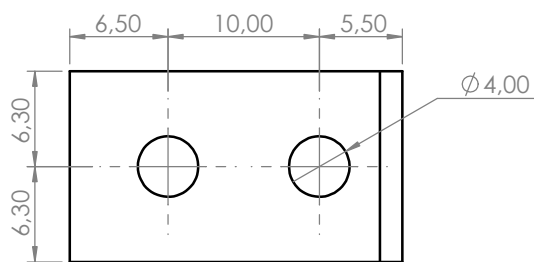
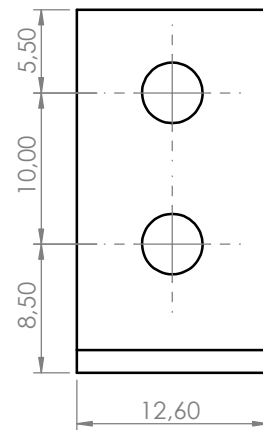
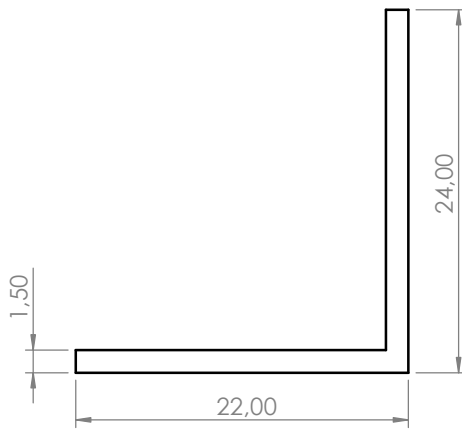
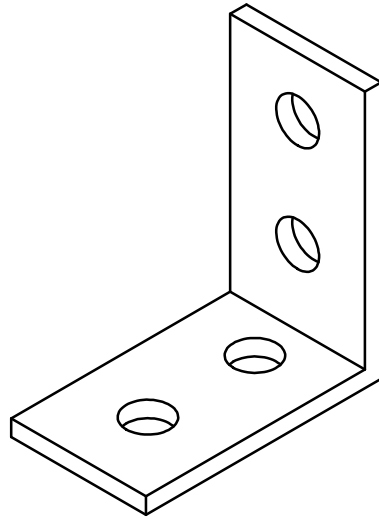
Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	1:1	Balanza: base (A)			Nombre archivo Balanza_base.SLDPRT
					Nº plano 1.01
Formato A4					



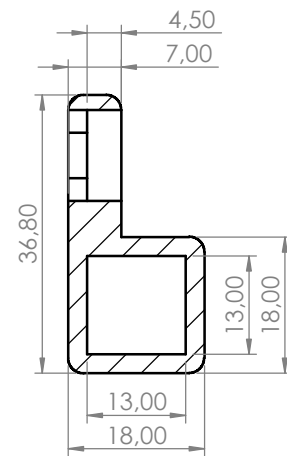
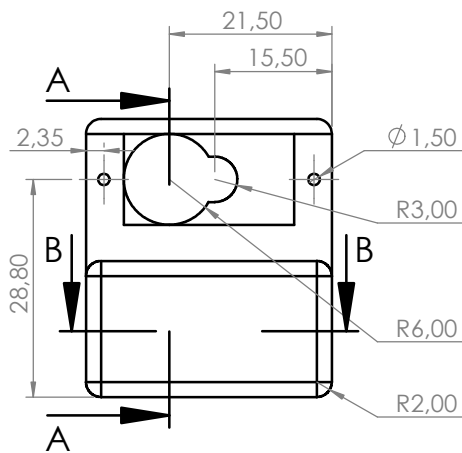
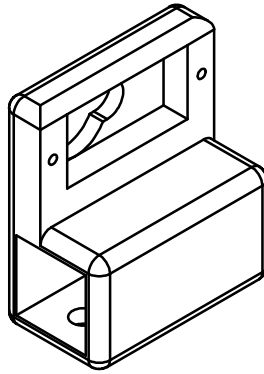
Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación Balanza: codo (B)			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
					Nombre archivo Balanza_codo.SLDPRT
	Formato A4				Nº plano 1.02



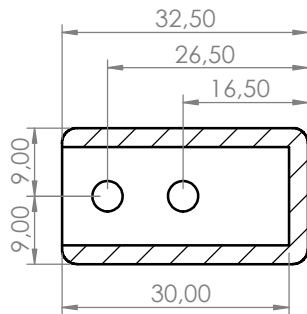
Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material Acero	Escala	Denominación Balanza: varilla (C)			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
					Nombre archivo Balanza_varilla.SLDPRТ
	Formato A4				Nº plano 1.03



Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material Aluminio	Escala	Denominación Balanza: ángulo (D)			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	2:1				Nombre archivo Balanza_ángulo.SLDPRT
					Nº plano 1.04
	Formato A4				

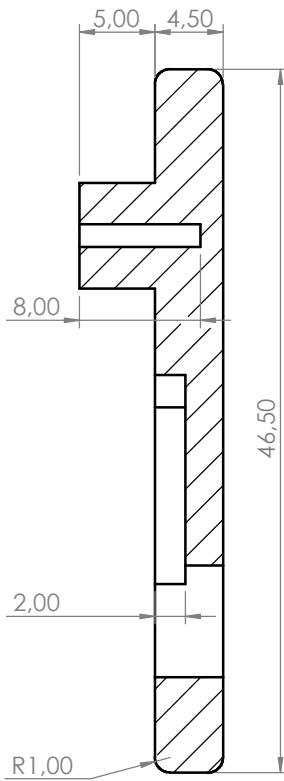
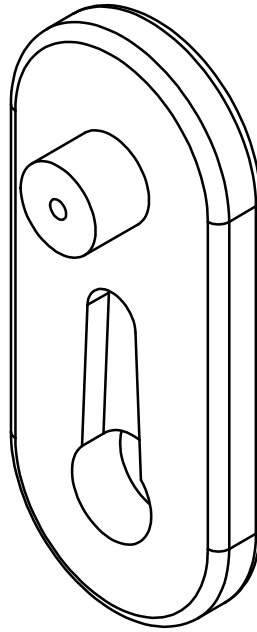


SECCIÓN A-A

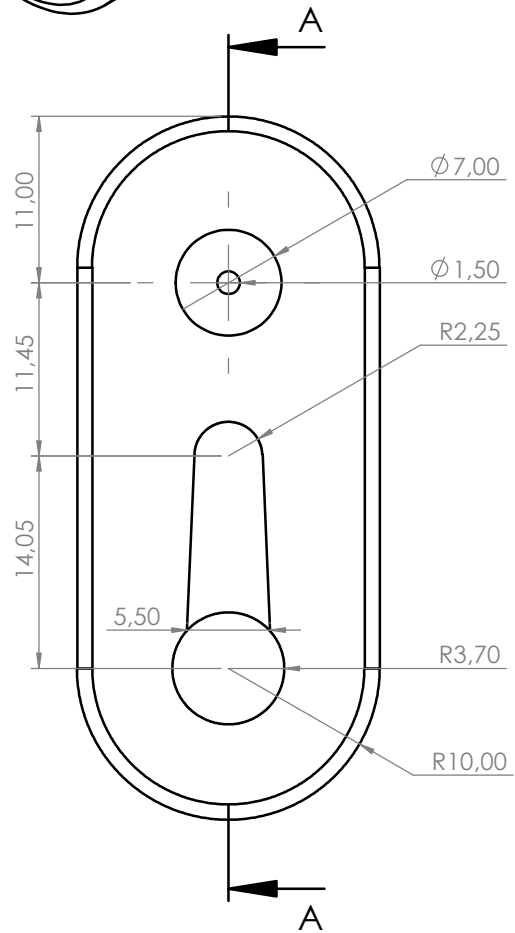


SECCIÓN B-B

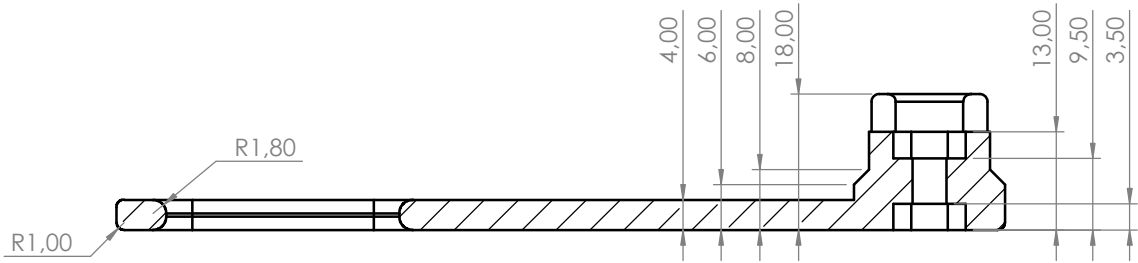
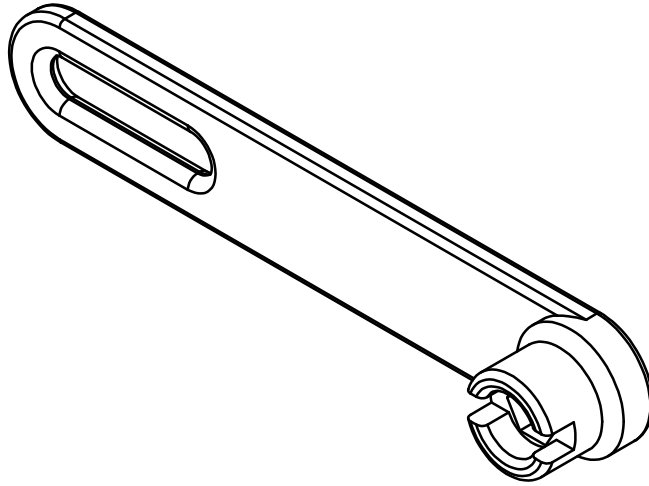
Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	1:1	Balanza: soporte servo (E)			
					



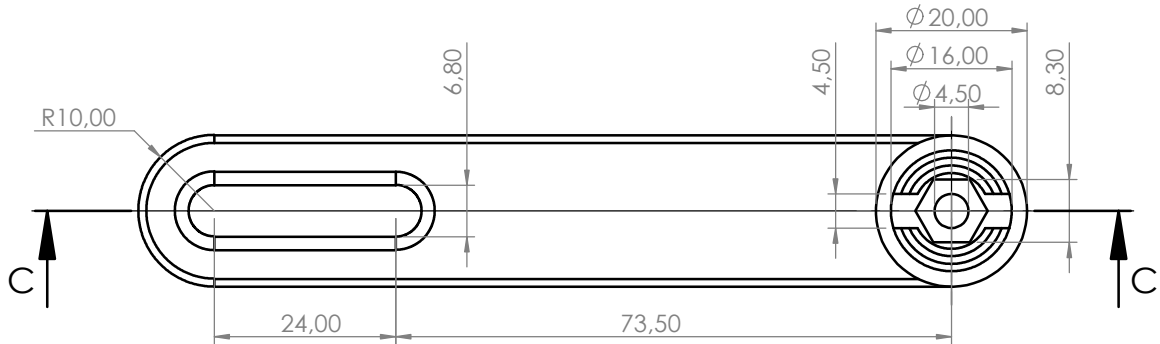
SECCIÓN A-A



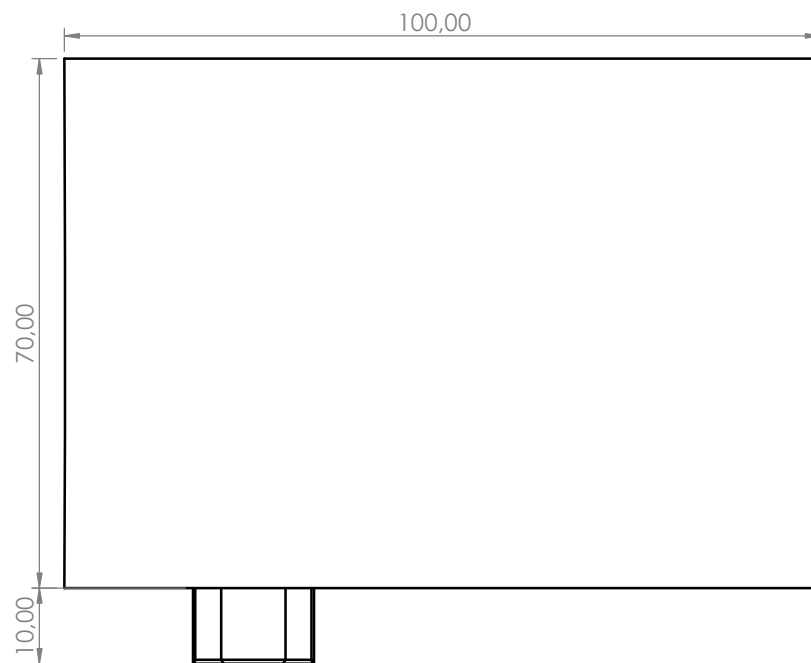
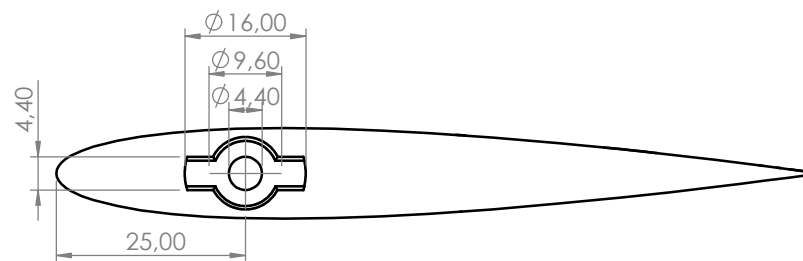
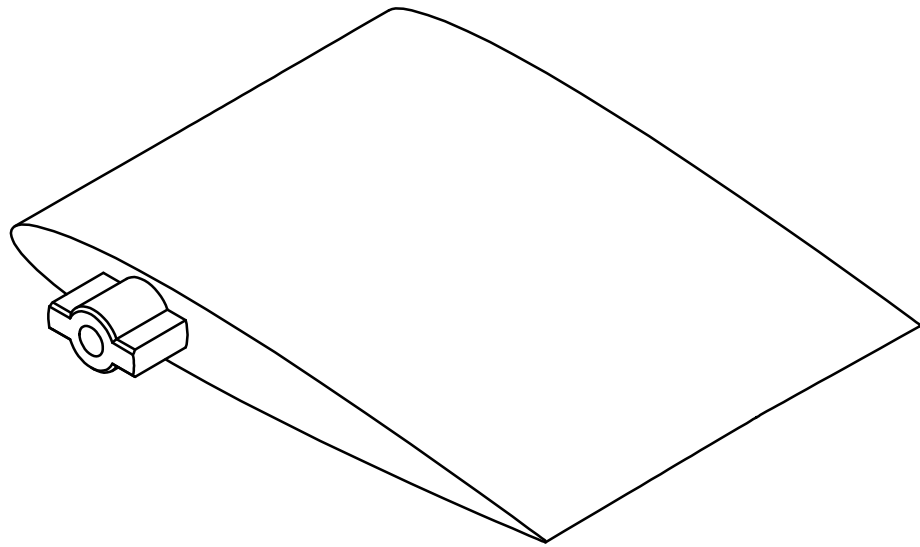
Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	2:1	Balanza: biela (J)			Nombre archivo Balanza_biela.SLDPRT
					Nº plano 1.06
Formato A4					



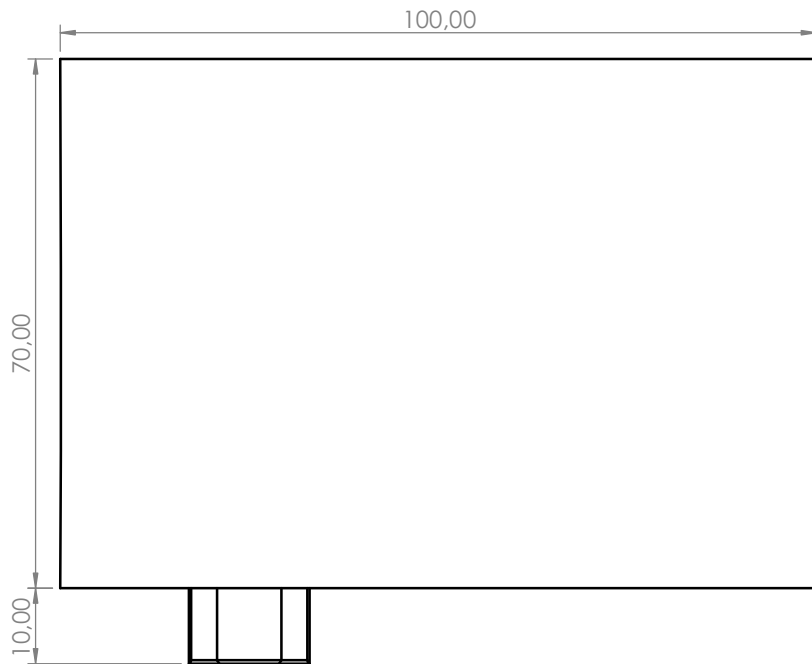
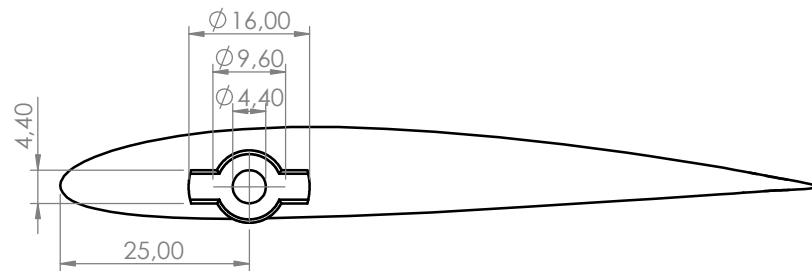
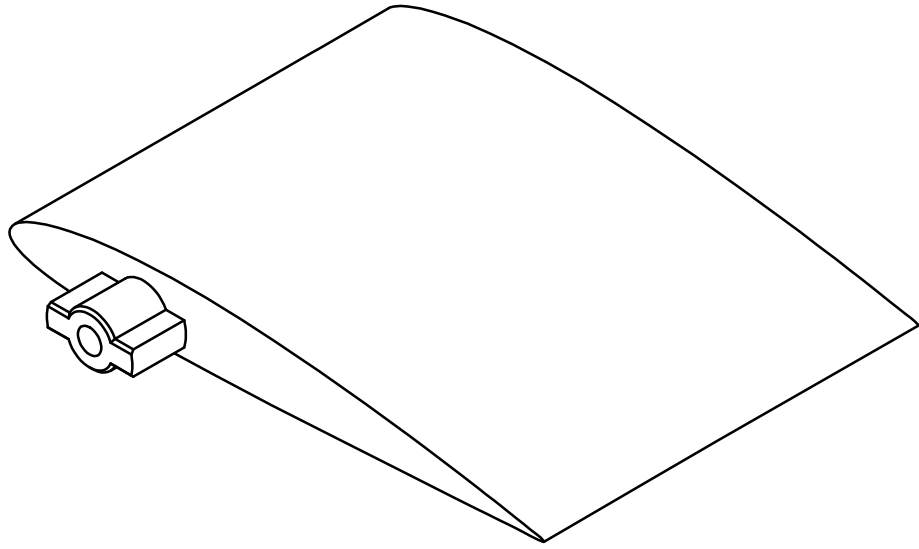
SECCIÓN C-C



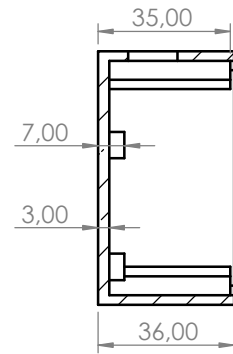
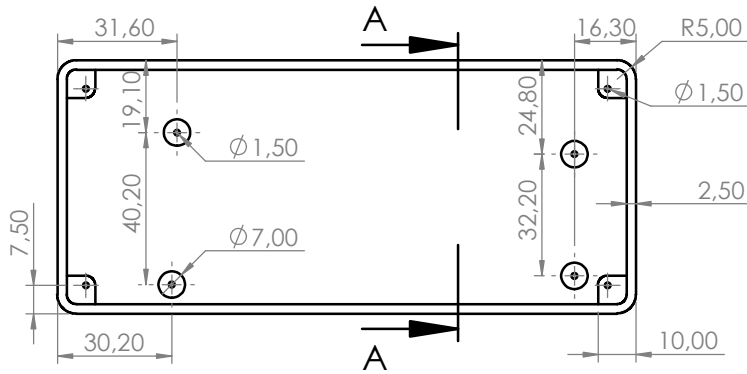
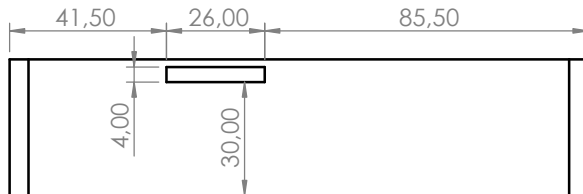
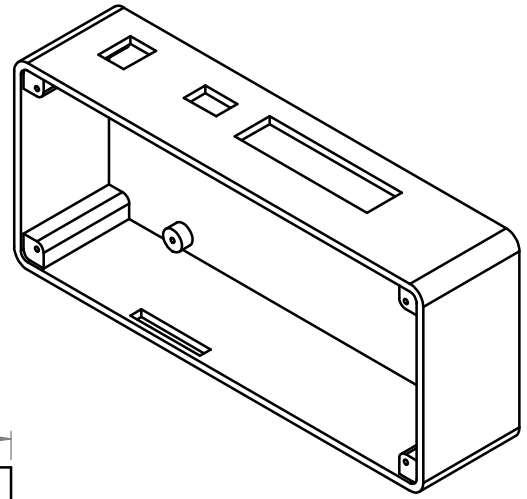
Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	1:1	Balanza: palanca (F)			
					Formato A4



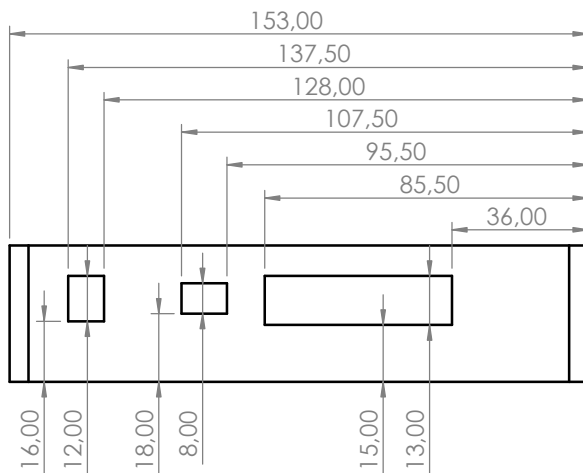
Tolerancias generales ± 0,2	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	1:1	Perfil de ala NACA 0012			Nombre archivo NACA_0012.SLDPRT
					Nº plano 2.01
Formato A4					



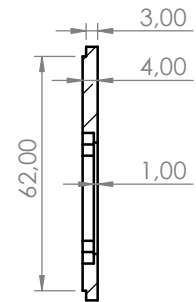
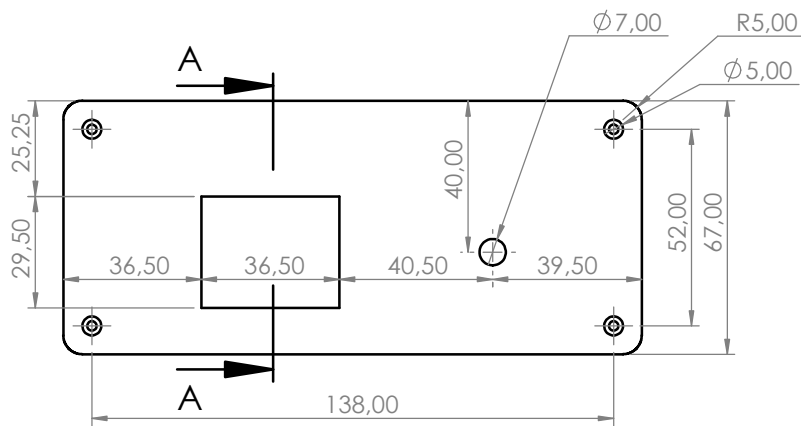
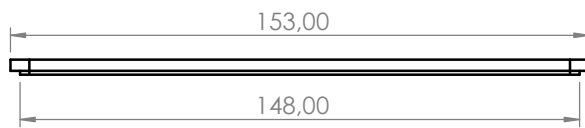
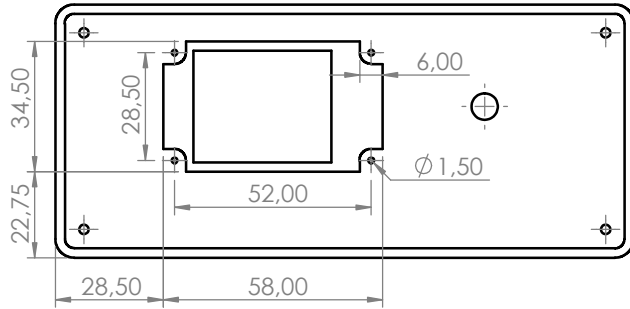
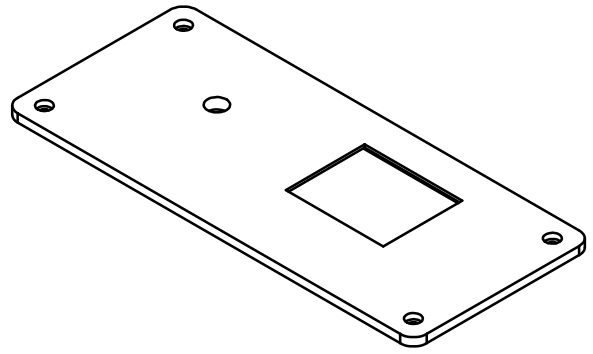
Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	1:1	Perfil de ala NACA 2412			Nombre archivo NACA_2412.SLDPRT
					Nº plano 2.02
Formato A4					



SECCIÓN A-A

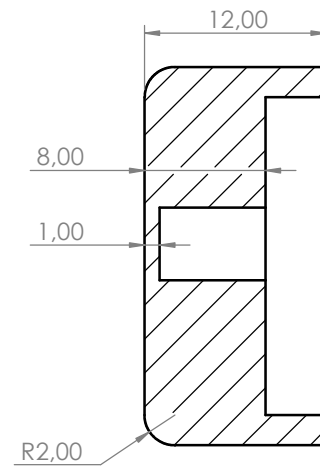
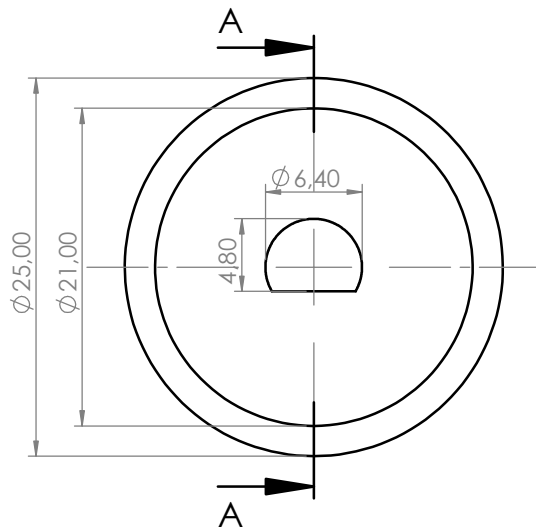
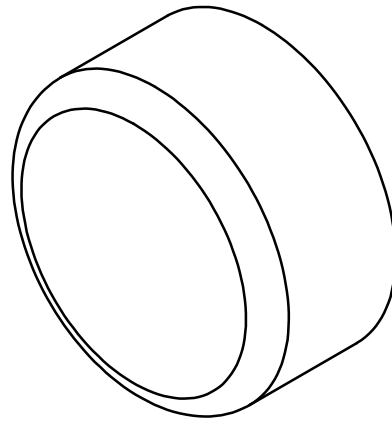


Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	1:2	Base gabinete			Nombre archivo Gabinete_base.SLDPRT
					Nº plano 3.01
Formato A4					



SECCIÓN A-A

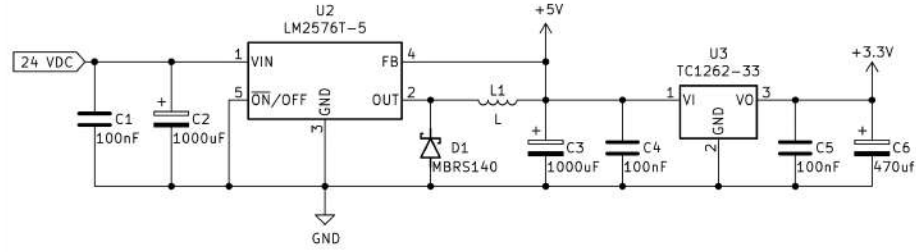
Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
		Tapa gabinete			
		1:2			
	Formato A4				Nº plano 3.02



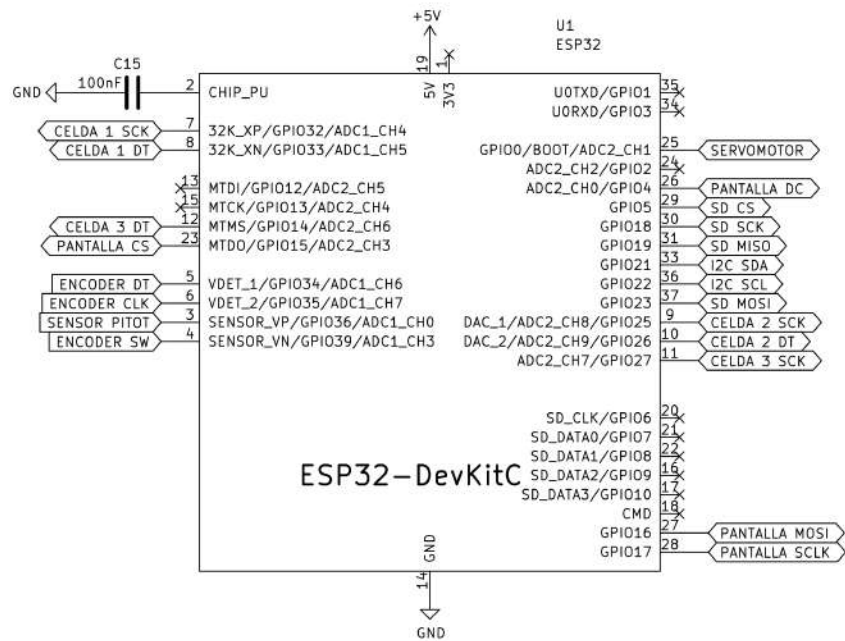
SECCIÓN A-A

Tolerancias generales $\pm 0,2$	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material PLA	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	2:1	Perilla gabinete			
					Nombre archivo Gabinete_perilla.SLDPRT
Formato A4				Nº plano 3.03	

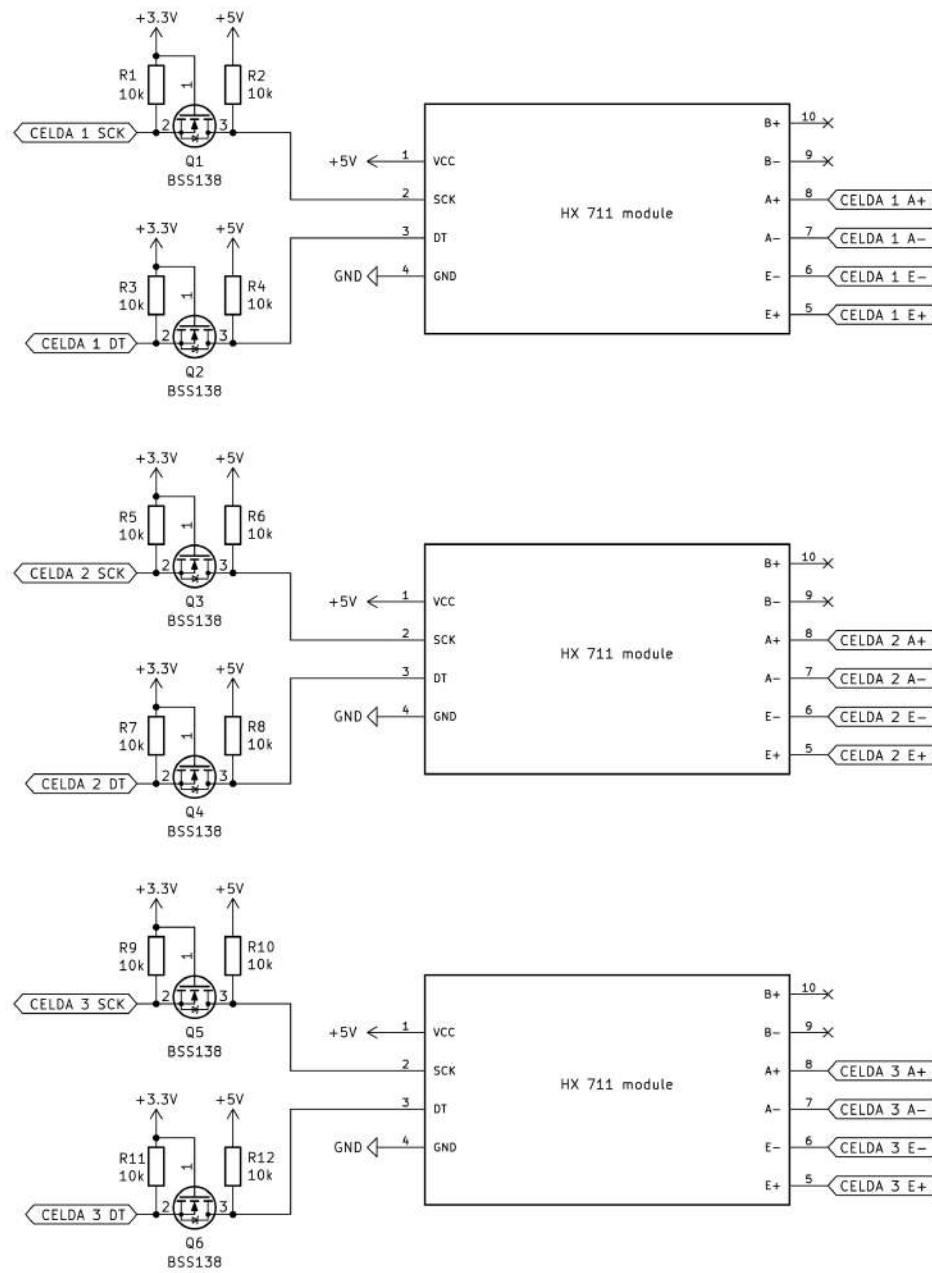
Fuente de tensión regulada



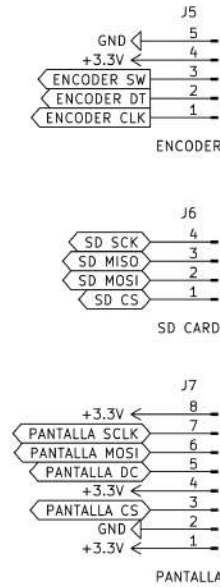
Esp32-DevKit



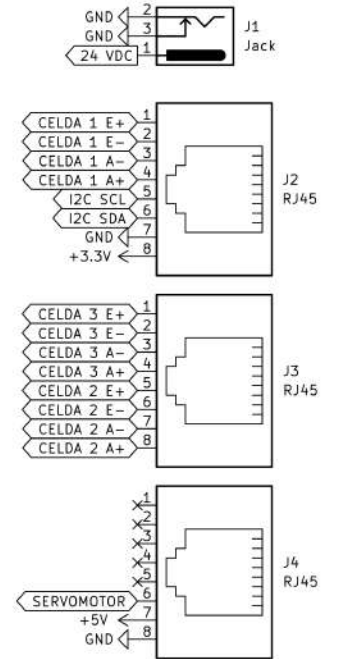
Acondicionamiento de señal de los módulos HX711



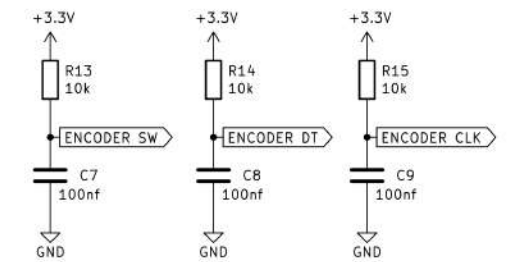
Borneras Internas



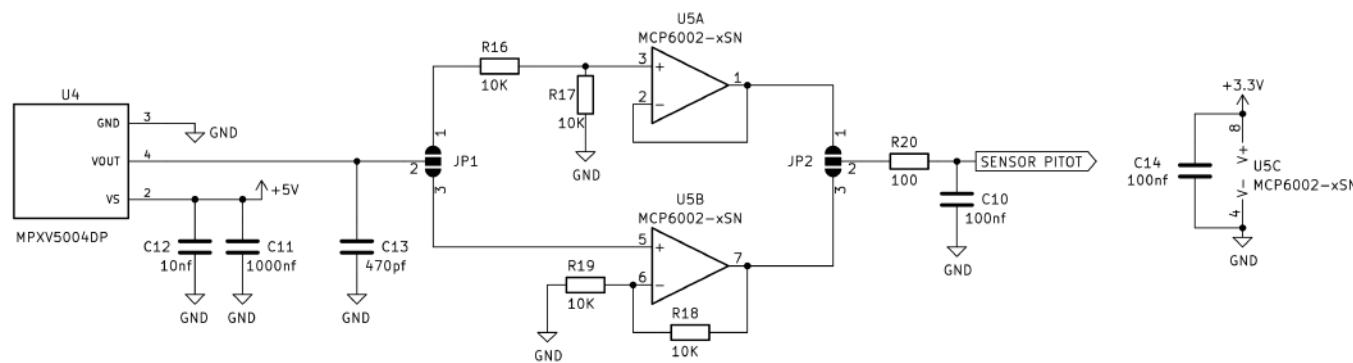
Borneras externas



Filtro encoder

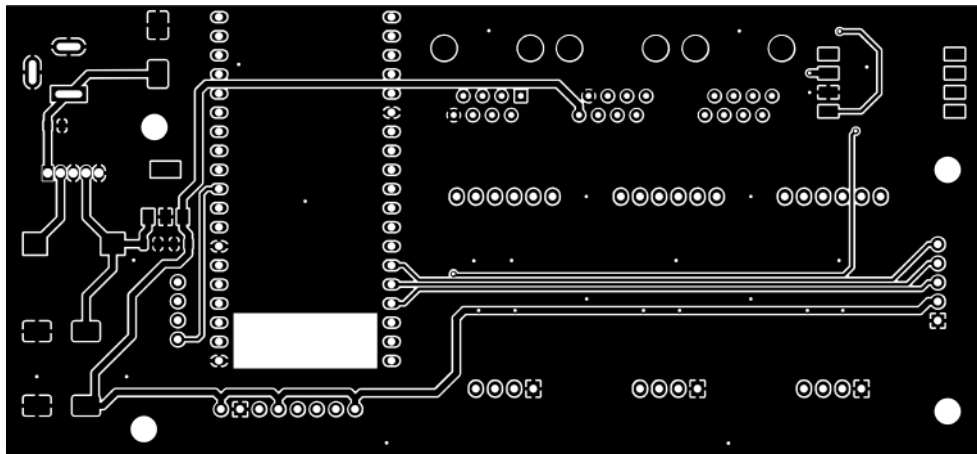


Acondicionamiento de señal del sensor de presión MPX5004

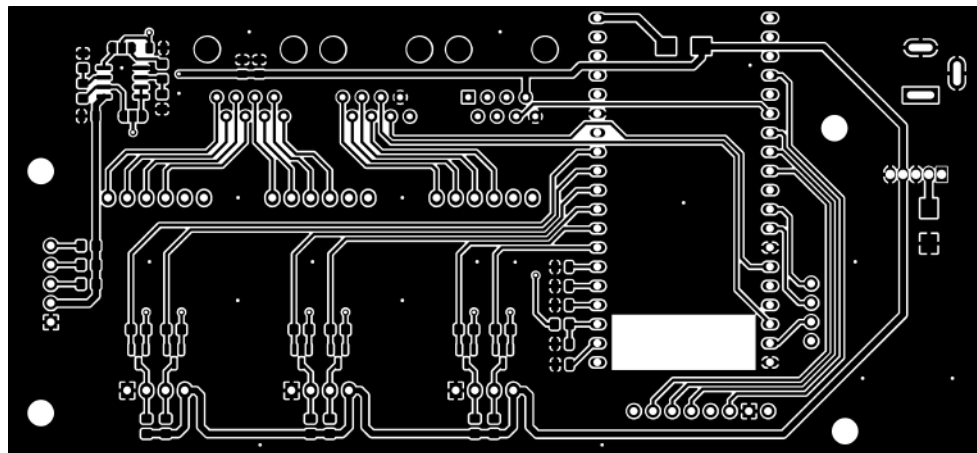


Tolerancias generales	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	S/E	Diagrama esquemático			
	Formato A3	Nombre archivo PCB.kicad_pro			
		Nº plano			4.01

Vista frontal



Vista posterior



Tolerancias generales	Proyectó	10/2023	Ricardo E. Lick		Proyecto y Diseño Final
	Dibujó	10/2023	Ricardo E. Lick		
	Revisó				
	Aprobó				
Material	Escala	Denominación			Desarrollo de un sistema de instrumentación y adquisición de datos para el túnel de viento de la Facultad de Ingeniería
	1:1	Placa de circuito impreso			
		Formato			
A4					Nº plano 4.02

Código fuente de la MCU ESP32 - sistema de adquisición de datos para túnel de viento

File: platformio.ini

```
01: [env:nodemcu-32s]
02: platform = espressif32
03: board = nodemcu-32s
04: framework = arduino
05: lib_deps =
06:     bogde/HX711@^0.7.5
07:     bodmer/TFT_eSPI@^2.5.31
08:     SPI@^2.0.0
09:     Wire@^2.0.0
10:     FS@^2.0.0
11:     SPIFFS@^2.0.0
12:     EEPROM@^2.0.0
13:     WiFi@^2.0.0
14:     ottowinter/ESPAsyncWebServer-esphome@^3.0.0
15:     bblanchon/ArduinoJson@^6.21.3
16:     roboticsbrno/ServoESP32@1.0.3
```

File: main.h

```
01: #ifndef MAIN_H
02: #define MAIN_H
03:
04: #include <WiFi.h>
05:
06: #define WIFI_SSID          "TunelDeViento"
07: #define WIFI_PASS         "12345678"
08: #define TMR_UPDATE_SENSORS 500
09:
10: #endif /*MAIN_H*/
```

File: main.cpp

```
01: #include "main.h"
02: #include "config.h"
03: #include "pin_manager.h"
04: #include "servo_manager.h"
05: #include "mpx_manager.h"
06: #include "sd_manager.h"
07: #include "bascula_manager.h"
08: #include "display_manager.h"
09: #include "web_server_manager.h"
10:
11: struct sys_t {
12:     uint8_t menu;
13:     bool hold;
14: };
15: sys_t sys = {0, false};
16:
17:
18: void CONTROLADOR_Initialize_Wifi(void) {
19:     WiFi.mode(WIFI_AP);
20:     WiFi.softAP(WIFI_SSID, WIFI_PASS, 1, 0, 1);
21: }
22:
23: void CONTROLADOR_Update_Sensors(void) {
24:     static uint32_t lastMillis = 0;
25:
26:     if ((millis() - lastMillis) < TMR_UPDATE_SENSORS) return;
27:     lastMillis = millis();
28:
```

```

29: MPX_MANAGER_Read();
30: BASCULA_MANAGER_Read();
31: SD_MANAGER_WriteFile();
32: }
33:
34: void CONTROLADOR_Menu(void) {
35:   if (btn.menu) {
36:     switch (sys.menu) {
37:       case 4: sys.hold = !sys.hold; break;
38:       case 6: BASCULA_MANAGER_Tara(); MPX_MANAGER_Tara(); break;
39:       case 7: SD_MANAGER_InitializeFile(); break;
40:     }
41:   }
42:
43:   if (btn.up) {
44:     if (sys.hold) SERVO_MANAGER_SetAnguloAtaque(SERVO_MANAGER_GetAnguloAtaque() + 1);
45:     if (!sys.hold && sys.menu < 8) sys.menu++;
46:   }
47:
48:   if (btn.down) {
49:     if (sys.hold) SERVO_MANAGER_SetAnguloAtaque(SERVO_MANAGER_GetAnguloAtaque() - 1);
50:     if (!sys.hold && sys.menu > 0) sys.menu--;
51:   }
52: }
53:
54: void setup() {
55:   PIN_MANAGER_Initialize();
56:   CONFIG_Initialize();
57:   CONTROLADOR_Initialize_Wifi();
58:   SD_MANAGER_Initialize();
59:   BASCULA_MANAGER_Initialize();
60:   SERVO_MANAGER_Initialize();
61:   MPX_MANAGER_Initialize();
62:   DISPLAY_MANAGER_Initialize();
63:   WEB_MANAGER_Initialize();
64: }
65:
66: void loop() {
67:   PIN_MANAGER_Loop();
68:   CONTROLADOR_Update_Sensors();
69:   CONTROLADOR_Menu();
70:   DISPLAY_MANAGER_Update(sys.menu, sys.hold);
71:   WEB_MANAGER_Loop();
72: }

```

File: pin_manager.h

```

01: #ifndef PIN_MANAGER_H
02: #define PIN_MANAGER_H
03:
04: #include <WiFi.h>
05:
06: #define LED_Pin          2          //Led integrado en la placa ESP32 DevKit
07: #define ENCODER_SW_Pin  39
08: #define ENCODER_DT_Pin  34
09: #define ENCODER_CLK_Pin 35
10: #define CELDA_1_DT_Pin  33
11: #define CELDA_1_SCK_Pin 32
12: #define CELDA_2_DT_Pin  26
13: #define CELDA_2_SCK_Pin 25
14: #define CELDA_3_DT_Pin  14
15: #define CELDA_3_SCK_Pin 27
16: #define SCL_Pin         22
17: #define SDA_Pin         21
18: #define DISPLAY_SCLK_Pin 17        //SCK
19: #define DISPLAY_MOSI_Pin 16        //SDA
20: #define DISPLAY_DC_Pin  4          //A0
21: #define DISPLAY_RST_Pin -1         //RESET

```

```

22: #define DISPLAY_CS_Pin      15      //CS
23: #define SD_CARD_CS_Pin     5
24: #define SD_CARD_MOSI_Pin   23
25: #define SD_CARD_MISO_Pin   19
26: #define SD_CARD_SCK_Pin    18
27: #define SERVO_Pin          0
28: #define MPX_Pin            36
29:
30: #define LED_SetDigitalOutput      pinMode(LED_Pin,OUTPUT)
31: #define LED_GetValue              digitalWrite(LED_Pin)
32: #define LED_SetHigh               digitalWrite(LED_Pin, HIGH)
33: #define LED_SetLow                digitalWrite(LED_Pin, LOW)
34: #define LED_Toggle                digitalWrite(LED_Pin, !LED_GetValue)
35: #define ENCODER_SW_SetDigitalInput pinMode(ENCODER_SW_Pin,INPUT)
36: #define ENCODER_SW_GetValue       !digitalRead(ENCODER_SW_Pin)
37: #define ENCODER_DT_SetDigitalInput pinMode(ENCODER_DT_Pin,INPUT)
38: #define ENCODER_DT_GetValue       digitalRead(ENCODER_DT_Pin)
39: #define ENCODER_CLK_SetDigitalInput pinMode(ENCODER_CLK_Pin,INPUT)
40: #define ENCODER_CLK_GetValue      digitalRead(ENCODER_CLK_Pin)
41:
42: struct btn_t {
43:     uint8_t menu;
44:     uint8_t up;
45:     uint8_t down;
46: };
47: extern btn_t btn;
48:
49: void PIN_MANAGER_Initialize(void);
50: void PIN_MANAGER_Loop(void);
51:
52: #endif /*PIN_MANAGER_H*/

```

File: pin_manager.cpp

```

01: #include "pin_manager.h"
02:
03: void ICACHE_RAM_ATTR PIN_MANAGER_ISR_EncoderA();
04: #define TMR_LED 500
05:
06: btn_t btn;
07: volatile int ISRCounter;
08:
09: void PIN_MANAGER_ISR_EncoderA() {
10:
11:     static uint32_t lastMillis = 0;
12:
13:     if ((millis() - lastMillis) < 5) return;
14:     lastMillis = millis();
15:
16:     if (ENCODER_DT_GetValue) ISRCounter++;
17:     else ISRCounter--;
18: }
19:
20: void PIN_MANAGER_blinkLed(void) {
21:
22:     static uint32_t lastMillis = 0;
23:
24:     if ((millis() - lastMillis) < TMR_LED) return;
25:     lastMillis = millis();
26:
27:     LED_Toggle;
28: }
29:
30: void PIN_MANAGER_ReadBtn(void) {
31:
32:     static uint8_t last_btn_menu = 0, last_btn_up = 0, last_btn_down = 0;
33:     static uint32_t lastMillis = 0;
34:

```

```

35: btn.menu = 0;
36: btn.up = 0;
37: btn.down = 0;
38:
39: if (((millis() - lastMillis) < 100) && (lastMillis != 0)) return;
40: lastMillis = millis();
41:
42: if (ENCODER_SW_GetValue && !last_btn_menu) {
43:     last_btn_menu = 1;
44:     btn.menu = 1;
45: }
46: if (!ENCODER_SW_GetValue && last_btn_menu) {
47:     last_btn_menu = 0;
48: }
49:
50: if (ISRCounter > 0) { btn.up = ISRCounter; }
51: if (ISRCounter < 0) { btn.down = -ISRCounter; }
52:
53: ISRCounter = 0;
54: }
55:
56: void PIN_MANAGER_Initialize(void) {
57:
58:     Serial.begin(115200);
59:     LED_SetDigitalOutput;
60:     ENCODER_SW_SetDigitalInput;
61:     ENCODER_DT_SetDigitalInput;
62:     ENCODER_CLK_SetDigitalInput;
63:
64:     attachInterrupt(digitalPinToInterrupt(ENCODER_CLK_Pin), PIN_MANAGER_ISR_EncoderA, RISING);
65: }
66:
67: void PIN_MANAGER_Loop(void) {
68:     PIN_MANAGER_ReadBtn();
69:     PIN_MANAGER_blinkLed();
70: }

```

File: bascula_manager.h

```

01: #ifndef BASCULA_MANAGER_H
02: #define BASCULA_MANAGER_H
03:
04: #include <WiFi.h>
05:
06: enum {
07:     SUSTENT,
08:     ARRASTR,
09:     CABECEO,
10: } basc_type_t;
11:
12: void BASCULA_MANAGER_Initialize(void);
13: void BASCULA_MANAGER_Read(void);
14: void BASCULA_MANAGER_Tara(void);
15: double BASCULA_MANAGER_GetValue(uint8_t i);
16: long BASCULA_MANAGER_GetCuentas(uint8_t i);
17:
18: #endif /*BASCULA_MANAGER_H*/

```

File: bascula_manager.cpp

```

01: #include "bascula_manager.h"
02: #include "pin_manager.h"
03: #include "servo_manager.h"
04: #include "HX711.h"
05:
06: HX711 scale[3];
07:
08: // Matriz inversa de los coeficientes de sensibilidad

```

```

09: const double sensibilidad_s[3] = {-1/2091.1, 2.1E-6, 7.4E-7};
10: const double sensibilidad_a[3] = {1.1E-6, -1/1937.4, 1.0E-6};
11: const double sensibilidad_m[3] = {5.0E-7, -8.0E-7, -1/191.3};
12:
13: // Vector offset
14: const long offset[3] = {485010, 64454, -132698};
15:
16: // Escalar cos(alfa)*9,577/r1(alfa)
17: const double correccion_momento[] = {1.000,1.000,1.001,1.002,1.004,1.007,1.010,1.014,1.018,
    1.024,1.030,1.037,1.045,1.055,1.066,1.079,1.094,1.113,1.136,1.165,1.207};
18:
19: struct balanza_t {
20:     long salida;
21:     long offset;
22:     long diferencia;
23: };
24:
25: balanza_t balanza[3] = {{0,0,0},{0,0,0},{0,0,0}};
26: double fuerza[3] = {0,0,0};
27:
28: void BASCULA_MANAGER_Initialize(void) {
29:     scale[SUSTENT].begin(CELDA_1_DT_Pin, CELDA_1_SCK_Pin);
30:     scale[ARRASTR].begin(CELDA_2_DT_Pin, CELDA_2_SCK_Pin);
31:     scale[CABECE0].begin(CELDA_3_DT_Pin, CELDA_3_SCK_Pin);
32:
33:     balanza[SUSTENT].offset = offset[SUSTENT];
34:     balanza[ARRASTR].offset = offset[ARRASTR];
35:     balanza[CABECE0].offset = offset[CABECE0];
36: }
37:
38: void BASCULA_MANAGER_Read(void) {
39:     if (scale[SUSTENT].is_ready()) balanza[SUSTENT].salida = scale[SUSTENT].read();
40:     if (scale[ARRASTR].is_ready()) balanza[ARRASTR].salida = scale[ARRASTR].read();
41:     if (scale[CABECE0].is_ready()) balanza[CABECE0].salida = scale[CABECE0].read();
42:
43:     balanza[SUSTENT].diferencia = (double)(balanza[SUSTENT].salida - balanza[SUSTENT].offset);
44:     balanza[ARRASTR].diferencia = (double)(balanza[ARRASTR].salida - balanza[ARRASTR].offset);
45:     balanza[CABECE0].diferencia = (double)(balanza[CABECE0].salida - balanza[CABECE0].offset);
46:
47:     double f[3];
48:
49:     //fuerza de sustentacion
50:     f[SUSTENT] = balanza[SUSTENT].diferencia * sensibilidad_s[SUSTENT];
51:     f[ARRASTR] = balanza[ARRASTR].diferencia * sensibilidad_s[ARRASTR];
52:     f[CABECE0] = balanza[CABECE0].diferencia * sensibilidad_s[CABECE0];
53:     fuerza[SUSTENT] = f[SUSTENT] + f[ARRASTR] + f[CABECE0];
54:
55:     //fuerza de arrastre
56:     f[SUSTENT] = balanza[SUSTENT].diferencia * sensibilidad_a[SUSTENT];
57:     f[ARRASTR] = balanza[ARRASTR].diferencia * sensibilidad_a[ARRASTR];
58:     f[CABECE0] = balanza[CABECE0].diferencia * sensibilidad_a[CABECE0];
59:     fuerza[ARRASTR] = f[SUSTENT] + f[ARRASTR] + f[CABECE0];
60:
61:     //momento de cabeceo
62:     int ang = SERVO_MANAGER_GetAnguloAtaque();
63:     f[SUSTENT] = balanza[SUSTENT].diferencia * sensibilidad_m[SUSTENT];
64:     f[ARRASTR] = balanza[ARRASTR].diferencia * sensibilidad_m[ARRASTR];
65:     f[CABECE0] = balanza[CABECE0].diferencia * sensibilidad_m[CABECE0] /
        correccion_momento[abs(ang)];
66:     fuerza[CABECE0] = f[SUSTENT] + f[ARRASTR] + f[CABECE0];
67:
68: }
69:
70: void BASCULA_MANAGER_Tara(void) {
71:     balanza[SUSTENT].offset = balanza[SUSTENT].salida;
72:     balanza[ARRASTR].offset = balanza[ARRASTR].salida;
73:     balanza[CABECE0].offset = balanza[CABECE0].salida;
74: }
75:

```

```

76: double BASCULA_MANAGER_GetValue(uint8_t i) {
77:     return fuerza[i];
78: }
79:
80: long BASCULA_MANAGER_GetCuentas(uint8_t i) {
81:     return balanza[i].salida;
82: }

```

File: mpx_manager.h

```

01: #ifndef MPX_MANAGER_H
02: #define MPX_MANAGER_H
03:
04: #include <WiFi.h>
05:
06: void MPX_MANAGER_Initialize(void);
07: void MPX_MANAGER_Read(void);
08: void MPX_MANAGER_Tara(void);
09: int MPX_MANAGER_GetVelAire(void);
10: int MPX_MANAGER_GetPressure(void);
11:
12: #endif /*MPX_MANAGER_H*/

```

File: mpx_manager.cpp

```

01: #include "mpx_manager.h"
02: #include "pin_manager.h"
03:
04: int pressure = 0;
05: int offsetPressure = 0;
06:
07: void MPX_MANAGER_Initialize(void) {
08:
09:     analogReadResolution(11);
10:     analogSetPinAttenuation(MPX_Pin, ADC_0db);
11:
12:     MPX_MANAGER_Read();
13:     MPX_MANAGER_Tara();
14: }
15:
16: void MPX_MANAGER_Read(void) {
17:
18:     uint32_t mv = 0;
19:
20:     for (int i=0 ; i<200 ; i++) //se suman 200 lecturas y se promedia para mayor precisión
21:         mv += analogReadMilliVolts(MPX_Pin);
22:
23:     pressure = mv/100; //se multiplica x2 el promedio por el divisor resistivo en pcb
24: }
25:
26: void MPX_MANAGER_Tara(void) {
27:     offsetPressure = pressure;
28: }
29:
30: int MPX_MANAGER_GetVelAire(void) {
31:     return (int)sqrt((double)(abs(pressure - offsetPressure) * 2) / (double)1.225);
32: }
33:
34: int MPX_MANAGER_GetPressure(void) {
35:     return (pressure - offsetPressure);
36: }

```

File: servo_manager.h

```

01: #ifndef SERVO_MANAGER_H
02: #define SERVO_MANAGER_H
03:
04: #include <WiFi.h>

```



```

05:
06: void SERVO_MANAGER_Initialize(void);
07: void SERVO_MANAGER_SetAnguloAtaque(int);
08: int SERVO_MANAGER_GetAnguloAtaque(void);
09:
10: #endif /*SERVO_MANAGER_H*/

```

File: servo_manager.cpp

```

01: #include "servo_manager.h"
02: #include "pin_manager.h"
03: #include <Servo.h>
04:
05: Servo servo;
06:
07: int anguloAtaque = 0;
08: const int anguloServo[] = {0,4,8,11,15,19,23,27,31,35,39,43,47,51,56,60,65,71,76,83,90};
09:
10: void SERVO_MANAGER_Initialize(void) {
11:
12:     servo.attach(SERVO_Pin);
13:     SERVO_MANAGER_SetAnguloAtaque(anguloAtaque);
14: }
15:
16: void SERVO_MANAGER_SetAnguloAtaque(int ang) {
17:     int as = 0;
18:
19:     if (ang < -20 || ang > 20) return;
20:
21:     anguloAtaque = ang;
22:
23:     as = (ang < 0)? 90 - anguloServo[-ang] : 90 + anguloServo[ang];
24:
25:     servo.write(as);
26: }
27:
28: int SERVO_MANAGER_GetAnguloAtaque(void) {
29:
30:     return anguloAtaque;
31: }

```

File: sd_manager.h

```

01: #ifndef SD_MANAGER_H
02: #define SD_MANAGER_H
03:
04: #include <WiFi.h>
05:
06: void SD_MANAGER_Initialize(void);
07: void SD_MANAGER_InitializeFile(void);
08: void SD_MANAGER_WriteFile(void);
09: bool SD_MANAGER_RegistrarIsOn(void);
10: bool SD_MANAGER_SdIsOk(void);
11: String SD_MANAGER_FileName(void);
12:
13: #endif /*SD_MANAGER_H*/

```

File: sd_manager.cpp

```

01: #include "sd_manager.h"
02: #include "pin_manager.h"
03: #include "bascula_manager.h"
04: #include "servo_manager.h"
05: #include "mpx_manager.h"
06: #include "config.h"
07: #include "FS.h"
08: #include "SD.h"
09: #include "SPI.h"

```

```

10:
11: SPIClass spiSd = SPIClass(VSPI);
12:
13: #define ENCABEZADO_CSV "Fuerza sustentacion [g],Fuerza arrastre [g],Momento cabeceo
    [g.cm],Angulo de ataque [°],Velocidad Aire [m/seg]"
14:
15: bool sdIsOk = true;
16: bool registrarIsOn = false;
17: String fileName = "-";
18:
19: void SD_MANAGER_Initialize(void) {
20:     spiSd.begin(SD_CARD_SCK_Pin, SD_CARD_MISO_Pin, SD_CARD_MOSI_Pin, SD_CARD_CS_Pin);
21: }
22:
23: bool SD_MANAGER_AppendFile(String message) {
24:
25:     File file = SD.open("/") + fileName, FILE_APPEND);
26:     if (!file) return false;
27:     file.println(message);
28:     file.close();
29:
30:     return true;
31: }
32:
33: void SD_MANAGER_InitializeFile(void) {
34:
35:     if (registrarIsOn) {
36:         registrarIsOn = false;
37:         fileName = "-";
38:         SD.end();
39:         return;
40:     }
41:
42:     if (!SD.begin(SD_CARD_CS_Pin, spiSd, 1000000)) {
43:         sdIsOk = false;
44:         fileName = "-";
45:         SD.end();
46:         return;
47:     }
48:
49:     uint32_t file_number = CONFIG_FileNumber_Read();
50:     fileName = "reg_" + String(file_number) + ".csv";
51:
52:     if (!SD_MANAGER_AppendFile(ENCABEZADO_CSV)) {
53:         sdIsOk = false;
54:         fileName = "-";
55:         SD.end();
56:         return;
57:     }
58:
59:     CONFIG_FileNumber_Write(file_number+1);
60:     sdIsOk = true;
61:     registrarIsOn = true;
62: }
63:
64: void SD_MANAGER_WriteFile(void) {
65:
66:     if (!registrarIsOn) return;
67:
68:     String message = String(BASCULA_MANAGER_GetValue(SUSTENT),1) + "," +
        String(BASCULA_MANAGER_GetValue(ARRASTR),1) + "," +
        String(BASCULA_MANAGER_GetValue(CABECEO),1) + "," +
        String(SERVO_MANAGER_GetAnguloAtaque()) + "," +
        String(MPX_MANAGER_GetVelAire());
69:     SD_MANAGER_AppendFile(message);
70: }
71:
72: bool SD_MANAGER_RegistrarIsOn(void) {
73:     return registrarIsOn;

```

```

74: }
75:
76: bool SD_MANAGER_SdIsOk(void) {
77:     return sdIsOk;
78: }
79:
80: String SD_MANAGER_FileName(void) {
81:     return fileName;
82: }

```

File: display_manager.h

```

1: #ifndef DISPLAY_MANAGER_H
2: #define DISPLAY_MANAGER_H
3:
4: #include <WiFi.h>
5:
6: void DISPLAY_MANAGER_Initialize(void);
7: void DISPLAY_MANAGER_Update(uint8_t menu, bool hold);
8:
9: #endif /*DISPLAY_MANAGER_H*/

```

File: display_manager.cpp

```

01: #include "display_manager.h"
02: #include "pin_manager.h"
03: #include "mpx_manager.h"
04: #include "sd_manager.h"
05: #include "bascula_manager.h"
06: #include "servo_manager.h"
07: #include <TFT_eSPI.h>
08: #include <SPI.h>
09:
10: #define TMR_UPDATE_DISPLAY      200
11:
12: TFT_eSPI tft = TFT_eSPI(); // Invoke library, pins defined in User_Setup.h
13:
14: String StringFormatNum(String buff, uint8_t length) {
15:
16:     uint8_t buffLength = buff.length();
17:     for ( ; buffLength < length ; buffLength++) buff += " ";
18:     return buff;
19: }
20:
21: void DISPLAY_MANAGER_Initialize(void) {
22:
23:     tft.init();
24:     tft.setRotation(0);
25: }
26:
27: void DISPLAY_MANAGER_WriteText(String buff, int16_t y, int16_t x, uint8_t s, uint16_t c,
    uint16_t bg) {
28:
29:     x = x*6;
30:     y = (y==0)? (y+3):(y*(8+5)+13);
31:     tft.setTextColor(c,bg);
32:     tft.drawString(buff, x, y, s);
33: }
34:
35: void DISPLAY_MANAGER_Update(uint8_t menu, bool hold) {
36:
37:     static uint32_t lastMillis = 0;
38:     static uint8_t last_menu = 255;
39:
40:     if ((millis() - lastMillis) < TMR_UPDATE_DISPLAY) return;
41:     lastMillis = millis();
42:
43:     if (menu != last_menu) {

```

```

44:     tft.fillScreen(TFT_BLACK);
45:         //123456789012345678901
46:     DISPLAY_MANAGER_WriteText(" TUNEL DE VIENTO ", 0, 0, 1, TFT_CYAN, TFT_BLACK);
47:     tft.drawFastHLine(0, 15, tft.width(), TFT_CYAN);
48:     DISPLAY_MANAGER_WriteText("F SUST:          g ", 1, 0, 1, (menu==1)?
TFT_ORANGE:TFT_GREEN, TFT_BLACK);
49:     DISPLAY_MANAGER_WriteText("F ARRA:          g ", 2, 0, 1, (menu==2)?
TFT_ORANGE:TFT_GREEN, TFT_BLACK);
50:     DISPLAY_MANAGER_WriteText("M CABE:          g.cm", 3, 0, 1, (menu==3)?
TFT_ORANGE:TFT_GREEN, TFT_BLACK);
51:     DISPLAY_MANAGER_WriteText("ANG AT:          ", 4, 0, 1, (menu==4)?
TFT_ORANGE:TFT_GREEN, TFT_BLACK);
52:     DISPLAY_MANAGER_WriteText("VEL AI:          m/s", 5, 0, 1, (menu==5)?
TFT_ORANGE:TFT_GREEN, TFT_BLACK);
53:     DISPLAY_MANAGER_WriteText("AJUSTE DE CERO ", 7, 0, 1, (menu==6)?
TFT_ORANGE:TFT_GREEN, TFT_BLACK);
54:     DISPLAY_MANAGER_WriteText("REGISTRAR?     ", 9, 0, 1, (menu==7)?
TFT_ORANGE:TFT_GREEN, TFT_BLACK);
55:     DISPLAY_MANAGER_WriteText("ARCHIVO:     ", 10,0, 1, (menu==8)?
TFT_ORANGE:TFT_GREEN, TFT_BLACK);
56: }
57:
58: DISPLAY_MANAGER_WriteText(StringFormatNum(String(BASCULA_MANAGER_GetValue(SUSTENT), 1),
11), 1, 8, 1, (menu==1)? TFT_ORANGE:TFT_GREEN, TFT_BLACK);
59: DISPLAY_MANAGER_WriteText(StringFormatNum(String(BASCULA_MANAGER_GetValue(ARRASTR), 1),
11), 2, 8, 1, (menu==2)? TFT_ORANGE:TFT_GREEN, TFT_BLACK);
60: DISPLAY_MANAGER_WriteText(StringFormatNum(String(BASCULA_MANAGER_GetValue(CABECE0), 1),
9), 3, 8, 1, (menu==3)? TFT_ORANGE:TFT_GREEN, TFT_BLACK);
61: DISPLAY_MANAGER_WriteText(StringFormatNum(String(SERVO_MANAGER_GetAnguloAtaque()),
3), 4, 8, 1, (menu==4)? ((hold==true)? TFT_RED:TFT_ORANGE):TFT_GREEN, TFT_BLACK);
62: DISPLAY_MANAGER_WriteText(StringFormatNum(String(MPX_MANAGER_GetVelAire()),
9), 5, 8, 1, (menu==5)? TFT_ORANGE:TFT_GREEN, TFT_BLACK);
63: DISPLAY_MANAGER_WriteText((SD_MANAGER_RegistrarIsOn())? "SI":"NO",
9,12, 1, (menu==7)? TFT_ORANGE:TFT_GREEN, TFT_BLACK);
64: DISPLAY_MANAGER_WriteText(StringFormatNum(SD_MANAGER_FileName(),
12), 10,9, 1, (menu==8)? TFT_ORANGE:TFT_GREEN, TFT_BLACK);
65:
66: last_menu = menu;
67: }

```

File: config.h

```

01: #ifndef CONFIG_H
02: #define CONFIG_H
03:
04: #include <WiFi.h>
05:
06: void CONFIG_Initialize(void);
07: uint32_t CONFIG_FileNumber_Read(void);
08: void CONFIG_FileNumber_Write(uint32_t);
09:
10: #endif /*CONFIG_H*/

```

File: config.cpp

```

01: #include "config.h"
02: #include <EEPROM.h>
03:
04: const uint8_t BYTE_INIT_EE = 0x46;
05: const uint8_t EE_POS_BYTE_INIT = 0;
06: const uint8_t EE_POS_FILE_NUMBER = 5;
07:
08: uint32_t CONFIG_FileNumber_Read(void) {
09:     uint32_t value;
10:     EEPROM.get(EE_POS_FILE_NUMBER, value);
11:     return value;
12: }
13:

```

```

14: void CONFIG_FileNumber_Write(uint32_t value) {
15:     EEPROM.put(EE_POS_FILE_NUMBER, value);
16:     EEPROM.commit();
17: }
18:
19: void CONFIG_Initialize(void) {
20:     uint8_t byte_init_ee;
21:
22:     EEPROM.begin(100);
23:
24:     EEPROM.get(EE_POS_BYTE_INIT, byte_init_ee);
25:
26:     if (byte_init_ee != BYTE_INIT_EE) {
27:         CONFIG_FileNumber_Write(1);
28:         EEPROM.put(EE_POS_BYTE_INIT, BYTE_INIT_EE);
29:         EEPROM.commit();
30:     }
31: }

```

File: web_server_manager.h

```

1: #ifndef WEB_SERVER_MANAGER_H
2: #define WEB_SERVER_MANAGER_H
3:
4: #include <WiFi.h>
5:
6: void WEB_MANAGER_Initialize(void);
7: void WEB_MANAGER_Loop(void);
8:
9: #endif /*WEB_SERVER_MANAGER_H*/

```

File: web_server_manager.cpp

```

01: #include "main.h"
02: #include "web_server_manager.h"
03: #include "bascula_manager.h"
04: #include "servo_manager.h"
05: #include "mpx_manager.h"
06: #include "sd_manager.h"
07: #include <ESPAsyncWebServer.h>
08: #include <ArduinoJson.h>
09: #include "SPIFFS.h"
10:
11: // Create AsyncWebServer object on port 80
12: AsyncWebServer server(80);
13:
14: // Create an Event Source on /events
15: AsyncEventSource events("/events");
16:
17: String getSensorReadings(){
18:     DynamicJsonDocument jsonDoc(1024);
19:     jsonDoc["fSust"] = String(BASCULA_MANAGER_GetValue(SUSTENT), 2);
20:     jsonDoc["fArra"] = String(BASCULA_MANAGER_GetValue(ARRASTR), 2);
21:     jsonDoc["fCabe"] = String(BASCULA_MANAGER_GetValue(CABECE0), 2);
22:     jsonDoc["velAire"] = MPX_MANAGER_GetVelAire();
23:     jsonDoc["angAtaq"] = SERVO_MANAGER_GetAnguloAtaque();
24:     jsonDoc["fileSave"] = SD_MANAGER_RegistrarIsOn();
25:     jsonDoc["fileName"] = SD_MANAGER_FileName();
26:     // Serializar el JSON a una cadena
27:     String jsonString;
28:     serializeJson(jsonDoc, jsonString);
29:     return jsonString;
30: }
31:
32: void WEB_MANAGER_Initialize(void)
33: {
34:     if (!SPIFFS.begin()) {
35:         Serial.println("Error al iniciar SPIFFS");

```

```

36:         return;
37:     }
38:
39:     server.serveStatic("/", SPIFFS, "/").setDefaultFile("index.html");
40:
41:     server.on("/api/botonTara", HTTP_GET, [](AsyncWebServerRequest *request){
42:         BASCULA_MANAGER_Tara();
43:         MPX_MANAGER_Tara();
44:         request->send(200, "text/plain", "OK");
45:     });
46:
47:     server.on("/api/botonAnguloAtaqueMas", HTTP_GET, [](AsyncWebServerRequest *request){
48:         SERVO_MANAGER_SetAnguloAtaque(SERVO_MANAGER_GetAnguloAtaque() + 1);
49:         request->send(200, "text/plain", "OK");
50:     });
51:
52:     server.on("/api/botonAnguloAtaqueMenos", HTTP_GET, [](AsyncWebServerRequest *request){
53:         SERVO_MANAGER_SetAnguloAtaque(SERVO_MANAGER_GetAnguloAtaque() - 1);
54:         request->send(200, "text/plain", "OK");
55:     });
56:
57:     server.on("/api/botonGuardarDatos", HTTP_GET, [](AsyncWebServerRequest *request){
58:         SD_MANAGER_InitializeFile();
59:         request->send(200, "text/plain", "OK");
60:     });
61:
62:     server.onNotFound([](AsyncWebServerRequest *request) {
63:         request->send(400, "text/plain", "Not found");
64:     });
65:
66:     events.onConnect([](AsyncEventSourceClient *client){
67:         if(client->lastId()){
68:             Serial.printf("Client reconnected! Last message ID that it got is: %u\n",
69:                 client->lastId());
70:         }
71:         client->send("hello!", NULL, millis());
72:     });
73:     server.addHandler(&events);
74:
75:     server.begin();
76: }
77:
78: void WEB_MANAGER_Loop(void) {
79:     static uint32_t lastMillis = 0;
80:
81:     if ((millis() - lastMillis) > 1000) {
82:         lastMillis = millis();
83:         events.send(getSensorReadings().c_str(), "new_readings" ,millis());
84:     }
85: }

```

File: index.html

```

01: <!DOCTYPE html>
02: <html>
03: <head>
04:     <title>Tunel de viento</title>
05:     <meta name="viewport" content="width=device-width, initial-scale=1">
06:     <link rel="stylesheet" type="text/css" href="style.css">
07:     <script src="highcharts.js"></script>
08: </head>
09: <body>
10:     <div class="topnav">
11:         <h1>Sistema de adquisici&oacuten de datos para t&uacutenel de viento - UNLPam</h1>
12:     </div>
13:     <div class="content">
14:         <div class="data">

```

```

15:         <h2>Registro de datos en SD</h2>
16:         <p>Nombre de archivo: <span id="TextNombreArchivo"></span></p>
17:         <p><button id="botonGuardarDatos">Iniciar registro</button></p>
18:     </div>
19: </div>
20: <div class="content">
21:     <div class="data">
22:         <h2>&Aacutengulo de ataque y ajuste de cero</h2>
23:         <p>&Aacutengulo de ataque: <span type="number"
                id="textAnguloAtaque">[°]</span><button style="margin-left: 10px;"
                id="botonAnguloAtaqueMenos">-</button><button style="margin-left: 10px;"
                id="botonAnguloAtaqueMas">+</button></p>
24:         <p><button id="botonTaraCeldasCarga">Ajuste de cero</button></p>
25:     </div>
26: </div>
27: <div class="content">
28:     <div class="data">
29:         <h2>&Uacutelultima lectura de los sensores</h2>
30:         <p>Fuerza de sustentaci&oacuten: <span id="textFuerzaSustent">1.0</span> [g]</p>
31:         <p>Fuerza de arrastre: <span id="textFuerzaArrastr">1.0</span> [g]</p>
32:         <p>Momento de cabeceo: <span id="textFuerzaCabeceo">1.0</span> [g.cm]</p>
33:         <p>Velocidad del aire: <span id="textAireVelocidad">1.0</span> [m/seg]</p>
34:     </div>
35: </div>
36: <div class="content">
37:     <div class="card-grid">
38:         <div class="card">
39:             <h2 class="card-title">Gr&aacutefico</h2>
40:             <div id="chartF" class="chart-container"></div>
41:             <div id="chartV" class="chart-container"></div>
42:         </div>
43:     </div>
44: </div>
45:
46:     <script src="scriptIndex.js"></script>
47:
48: </body>
49: </html>

```

File: scriptIndex.js

```

001: // ----- Chart -----
002:
003: var chartF = new Highcharts.Chart({
004:   chart:{
005:     renderTo:'chartF'
006:   },
007:   series: [
008:     {
009:       name: 'Sustentaci&oacuten [g]',
010:       type: 'line',
011:       color: '#101D42',
012:       marker: {
013:         symbol: 'circle',
014:         radius: 3,
015:         fillColor: '#101D42',
016:       }
017:     },
018:     {
019:       name: 'Arrastre [g]',
020:       type: 'line',
021:       color: '#00A6A6',
022:       marker: {
023:         symbol: 'circle',
024:         radius: 3,
025:         fillColor: '#00A6A6',
026:       }
027:     },

```

```

028:     {
029:       name: 'Cabeceo [g.cm]',
030:       type: 'line',
031:       color: '#8B2635',
032:       marker: {
033:         symbol: 'circle',
034:         radius: 3,
035:         fillColor: '#8B2635',
036:       }
037:     },
038:   ],
039:   title: {
040:     text: undefined
041:   },
042:   xAxis: {
043:     //type: 'datetime',
044:     //dateTimeLabelFormats: { second: '%H:%M:%S' }
045:   },
046:   yAxis: {
047:     title: {
048:       text: undefined
049:     }
050:   },
051:   credits: {
052:     enabled: false
053:   }
054: });
055:
056: var chartV = new Highcharts.Chart({
057:   chart: {
058:     renderTo: 'chartV'
059:   },
060:   series: [
061:     {
062:       name: 'Velocidad del aire [m/s]',
063:       type: 'line',
064:       color: '#101D42',
065:       marker: {
066:         symbol: 'circle',
067:         radius: 3,
068:         fillColor: '#101D42',
069:       }
070:     },
071:     {
072:       name: '&Aacutengulo de ataque [&#176]',
073:       type: 'line',
074:       color: '#00A6A6',
075:       marker: {
076:         symbol: 'circle',
077:         radius: 3,
078:         fillColor: '#00A6A6',
079:       }
080:     },
081:   ],
082:   title: {
083:     text: undefined
084:   },
085:   xAxis: {
086:     //type: 'datetime',
087:     //dateTimeLabelFormats: { second: '%H:%M:%S' }
088:   },
089:   yAxis: {
090:     title: {
091:       text: undefined
092:     }
093:   },
094:   credits: {
095:     enabled: false
096:   }

```



```

097: });
098:
099: function updateChart(jsonValue) {
100:
101:   var keysChartF = ['fSust', 'fArra', 'fCabe'];
102:   var keysChartV = ['velAire', 'angAtaq'];
103:
104:   for (var i = 0; i < keysChartF.length; i++) {
105:     const key = keysChartF[i];
106:     var y = Number(jsonValue[key]);
107:
108:     while (chartF.series[i].data.length > 39) {
109:       chartF.series[i].data[0].remove(false);
110:     }
111:
112:     chartF.series[i].addPoint(y, true, false, true);
113:   }
114:
115:   for (var i = 0; i < keysChartV.length; i++) {
116:     const key = keysChartV[i];
117:     var y = Number(jsonValue[key]);
118:
119:     while (chartV.series[i].data.length > 39) {
120:       chartV.series[i].data[0].remove(false);
121:     }
122:
123:     chartV.series[i].addPoint(y, true, false, true);
124:   }
125: }
126:
127: // ----- Data -----
128:
129: function updateData(jsonValue) {
130:
131:   var text_FuerzaSustent = document.getElementById("textFuerzaSustent");
132:   var text_FuerzaArrastr = document.getElementById("textFuerzaArrastr");
133:   var text_FuerzaCabeceo = document.getElementById("textFuerzaCabeceo");
134:   var text_AireVelocidad = document.getElementById("textAireVelocidad");
135:   var text_AnguloAtaque = document.getElementById("textAnguloAtaque");
136:
137:   text_FuerzaSustent.textContent = Number(jsonValue['fSust']);
138:   text_FuerzaArrastr.textContent = Number(jsonValue['fArra']);
139:   text_FuerzaCabeceo.textContent = Number(jsonValue['fCabe']);
140:   text_AireVelocidad.textContent = Number(jsonValue['velAire']);
141:   text_AnguloAtaque.textContent = Number(jsonValue['angAtaq']);
142: }
143:
144: // ----- File -----
145:
146: var boton_GuardarDatos = document.getElementById("botonGuardarDatos");
147:
148: function updateFile(jsonValue) {
149:
150:   var text_NombreArchivo = document.getElementById("TextNombreArchivo");
151:
152:   if (jsonValue['fileSave'] == true) {
153:     boton_GuardarDatos.textContent = "Finalizar registro"
154:   } else {
155:     boton_GuardarDatos.textContent = "Iniciar registro"
156:   }
157:
158:   text_NombreArchivo.textContent = jsonValue['fileName'];
159: }
160:
161: boton_GuardarDatos.addEventListener("click", function() {
162:
163:   fetch("/api/botonGuardarDatos")
164:   .then(response => {
165:     })

```

```

166:   .catch(error => {
167:   });
168:
169: });
170:
171: // ----- Control -----
172:
173: var boton_TaraCeldasCarga = document.getElementById("botonTaraCeldasCarga");
174: var boton_AnguloAtaqueMas = document.getElementById("botonAnguloAtaqueMas");
175: var boton_AnguloAtaqueMenos = document.getElementById("botonAnguloAtaqueMenos");
176:
177: boton_TaraCeldasCarga.addEventListener("click", function() {
178:
179:   fetch("/api/botonTara")
180:   .then(response => {
181:   })
182:   .catch(error => {
183:   });
184:
185: });
186:
187: boton_AnguloAtaqueMas.addEventListener("click", function() {
188:
189:   fetch("/api/botonAnguloAtaqueMas")
190:   .then(response => {
191:   })
192:   .catch(error => {
193:   });
194:
195: });
196:
197: boton_AnguloAtaqueMenos.addEventListener("click", function() {
198:
199:   fetch("/api/botonAnguloAtaqueMenos")
200:   .then(response => {
201:   })
202:   .catch(error => {
203:   });
204: });
205:
206: // ----- Events -----
207:
208: if (!!window.EventSource) {
209:   var source = new EventSource('/events');
210:
211:   source.addEventListener('open', function(e) {
212:     console.log("Events Connected");
213:   }, false);
214:
215:   source.addEventListener('error', function(e) {
216:     if (e.target.readyState !== EventSource.OPEN) {
217:       console.log("Events Disconnected");
218:     }
219:   }, false);
220:
221:   source.addEventListener('message', function(e) {
222:     console.log("message", e.data);
223:   }, false);
224:
225:   source.addEventListener('new_readings', function(e) {
226:     var myObj = JSON.parse(e.data);
227:     updateFile(myObj);
228:     updateData(myObj);
229:     updateChart(myObj);
230:   }, false);
231: }

```

File: style.css

```

001: html {
002:   font-family: Arial, Helvetica, sans-serif;
003:   display: inline-block;
004:   text-align: center;
005: }
006: h1 {
007:   font-size: 1.8rem;
008:   color: white;
009: }
010: h2 {
011:   font-size: 1.5rem;
012:   color: #034078;
013: }
014: p {
015:   font-size: 1.2rem;
016:   color: #000000;
017: }
018: .topnav {
019:   overflow: hidden;
020:   background-color: #1a3ba5;
021: }
022: body {
023:   margin: 0;
024:   background-color: #eff3fa;
025: }
026: .content {
027:   padding: 1% 2%;
028: }
029: .card-grid {
030:   margin: 0 auto;
031:   display: grid;
032:   grid-gap: 2rem;
033:   grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
034: }
035: .card {
036:   background-color: white;
037:   box-shadow: 2px 2px 12px 1px rgba(140,140,140,.5);
038:   border-radius: 10px;
039: }
040: .card-title {
041:   font-size: 1.5rem;
042:   font-weight: bold;
043:   color: #034078;
044:   text-align: left;
045:   padding: 1.5%;
046: }
047: .chart-container {
048:   padding-right: 5%;
049:   padding-left: 5%;
050: }
051: .data {
052:   padding: 1.5%;
053:   background-color: white;
054:   border-radius: 10px;
055:   box-shadow: 2px 2px 12px 1px rgba(140,140,140,.5);
056:   text-align: left;
057: }
058: ul {
059:   list-style-type: none;
060:   overflow: hidden;
061: }
062: li {
063:   float: left;
064:   text-align: center;
065:   font-size: 15px;
066:   border-radius: 10px;
067:   background-color: #1a3ba5;
068:   margin: 0.25%;
069: }

```

```
070: li a {
071:   font-size: 1.5rem;
072:   padding: 20px 20px;
073:   display: block;
074:   color: #ffffff;
075:   border-radius: 10px;
076:   text-decoration: none;
077: }
078: li a:hover {
079:   background-color: #8dbbe3;
080: }
081: input {
082:   background-color: #cde1f3;
083:   color: #000000;
084:   border: none;
085:   border-radius: 10px;
086:   padding: 10px 10px;
087:   margin: 8px 8px;
088:   text-decoration: none;
089:   font-family: monospace;
090:   font-size: 1.2rem;
091:   text-align: center;
092: }
093: input:focus {
094:   outline: none;
095: }
096: button {
097:   border: none;
098:   background-color: #cde1f3;
099:   border-radius: 10px;
100:   padding: 10px 10px;
101:   color: #000000;
102:   text-decoration: none;
103:   font-family: Verdana, Geneva, Tahoma, sans-serif;
104:   font-size: 1.2rem;
105: }
106: button:hover {
107:   background-color: #bdd6ec;
108: }
```