



# **Desarrollo de un software para la calibración del sistema térmico y gravimétrico de un motor de combustión interna**

**Jesús Ricardo Duarte Bejarano  
Daniela Ballesteros Lizcano**

Universidad Antonio Nariño  
Facultad de Ingeniería Mecánica  
Puerto Colombia, Colombia

2023



# **Desarrollo de un software para la calibración del sistema térmico y gravimétrico de un motor de combustión interna**

**Jesús Ricardo Duarte Bejarano  
Daniela Ballesteros Lizcano**

Proyecto de grado presentado como requisito parcial para optar al título de:  
**Ingeniero Mecánico**

Director:

PhD. José Daniel Hernández Vásquez

Línea de Investigación:

software

Grupo de Investigación:

GI-Fourier

Universidad Antonio Nariño  
Facultad de Ingeniería Mecánica  
Puerto Colombia, Colombia

2023



*Dedicado a mi familia y las personas que nos apoyaron en este logro tan importante, a dios por ayudarnos en el proceso del entendimiento y sabiduría, Tranquilidad, paciencia y fortaleza para alcanzar dichos logros.*

*Los autores.*



## **Agradecimientos**

Damos gracias a DIOS por ser el artífice de nuestra existencia, por otorgarnos la vida y colmarnos con sus bendiciones día tras día.

Extendemos nuestro más sincero agradecimiento a nuestras familias, quienes constituyen nuestro pilar fundamental y nuestra fuente inagotable de motivación a lo largo de nuestra jornada vital.

De manera especial, deseamos expresar nuestra gratitud hacia nuestro guía y maestro, el profesor José Daniel Hernández Vásquez, cuya orientación ha sido fundamental a lo largo de todo el proyecto.

Agradecer a la universidad por los espacios brindados para un aprendizaje adecuado y de la mejor calidad.





## Resumen

El trabajo realizado se centró en el desarrollo de un software programado en lenguaje C#. La base de este proyecto fue una investigación previa que se enfocó en evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna.

Se utilizó una plantilla de Excel como punto de partida, la cual se había investigado previamente. Esto proporcionó una visión inicial de los datos que deben ser presentados en la interfaz del software y cómo esta información será visualizada por el usuario. El software se complementa con un lenguaje de programación que permite estructurar los resultados de manera ordenada y prevenir posibles errores que podrían surgir debido a un uso incorrecto del programa o a la introducción de datos erróneos por parte del usuario que interactúa con la interfaz. La precisión en la presentación de los datos es una de las principales fortalezas de este software, ya que garantiza una representación exacta de la información.

Adicionalmente, el software cuenta con el respaldo de un lenguaje de programación que facilita la ejecución de instrucciones precisas y eficientes. De esta manera, se crea un entorno en el que el usuario puede tomar decisiones fundamentadas basadas en los resultados de la investigación, lo que contribuye a promover el avance del conocimiento y la generación de nuevos proyectos que impulsan el desarrollo continuo.

**Palabras clave:** Programación, C#, motores de combustión interna, temperatura, balanza, incertidumbre sensores.

## Abstract

The work carried out focused on the development of software programmed in the C# language. The basis of this project was a previous investigation that focused on evaluating the metrological reliability of the thermal and gravimetric system of the internal combustion engine.

An Excel template was used as a starting point, which had been previously researched. This provided an initial view of the data that should be presented in the software interface and how this information will be displayed by the user. The software is complemented by a programming language that allows the results to be structured in an orderly manner and prevent possible errors that could arise due to incorrect use of the program or the introduction of erroneous data by the user who interacts with the interface. Accuracy in the presentation of data is one of the main strengths of this software, as it guarantees an accurate representation of the information.

Additionally, the software is supported by a programming language that facilitates the execution of precise and efficient instructions. In this way, an environment is created in which the user can make informed decisions based on research results, which contributes to promoting the advancement of knowledge and the generation of new projects that drive continuous development.

**Keywords:** Programming, C#, internal combustion engines, temperature, scale, uncertainty, sensors.

# Contenido

<b>1. Introducción.....</b>	<b>16</b>
1.1 Formulación del problema.....	17
1.2 Objetivos.....	17
1.2.1 Objetivo general.....	17
1.2.2 Objetivos específicos.....	17
1.3 Justificación.....	18
<b>2. Funcionamiento del motor de combustión.....</b>	<b>19</b>
2.1 Motores de combustión interna.....	19
2.1.1 Motores de gasolina.....	19
2.1.2 Motores diésel.....	19
2.2 Tipos de motores de combustión interna.....	20
<b>3. Fundamentos de la incertidumbre.....</b>	<b>22</b>
3.1 Incertidumbre de medición.....	22
3.1.1 3.1.1 Calibración.....	22
3.1.2 Medición.....	22
3.1.3 Incertidumbre Estándar.....	23
3.1.4 Patrón.....	23
3.1.5 Precisión.....	23
3.1.6 Exactitud.....	23
3.1.7 Error Sistemático.....	23
3.1.8 Error Aleatorio.....	23
3.2 Método general del calculo.....	23
3.3 Evaluación de incertidumbre Tipo A.....	24
3.4 Evaluación de incertidumbre Tipo B.....	24
3.5 Distribución Rectangular.....	25
3.6 Distribución normal y t-student.....	26
3.7 Incertidumbre estándar combinada.....	26
3.8 Incertidumbre expandida.....	26
<b>4. Fundamentos de programación.....</b>	<b>27</b>
4.1 Programación orientada a objetos.....	27
4.1.1 Clases y Objetos.....	27
4.1.2 Encapsulación.....	29
4.1.3 Herencia.....	30
4.1.4 Polimorfismo.....	32
4.2 Descripción de modelo vista controlador.....	34
4.2.1 Modelo (Model).....	35

4.2.2	Vista (View).....	35
4.2.3	Controlador (Controller) .....	36
4.3	Administración de Windows Forms de .net.....	36
4.3.1	Diseño de la Interfaz Gráfica de Usuario (GUI).....	37
4.3.2	Propiedades y Eventos.....	37
4.3.3	Programación y Lógica de la Aplicación.....	37
4.3.4	Compilación y Ejecución.....	38
4.3.5	Manejo de la Interfaz de Usuario.....	38
4.4	variables en C# usadas en .NET.....	38
4.4.1	Variable Tipo de Dato .....	38
4.4.2	Nombre de Variable.....	39
4.4.3	Declaración .....	39
4.4.4	Asignación de Valor .....	39
4.4.5	Uso de Variables.....	39
4.4.6	Alcance de Variables .....	39
4.4.7	Inicialización.....	40
4.4.8	4.4.8 Tipos de Variables .....	40
4.5	Reglas de los principios de SOLID.....	40
4.5.1	Principio de Responsabilidad Única (SRP - Single Responsibility Principle) .	40
4.5.2	Principio de Abierto/Cerrado (OCP - Open/Closed Principle).....	40
4.5.3	Principio de Sustitución de Liskov (LSP - Liskov Substitution Principle) .....	41
4.5.4	Principio de Segregación de Interfaces (ISP - Interface Segregation Principle)	41
4.5.5	Principio de Inversión de Dependencia (DIP - Dependency Inversion Principle)	41
4.6	Miembros de clase y las clases abstractas y selladas en C#.....	42
4.6.1	Clases Abstractas .....	42
4.6.2	Clases Selladas .....	42
4.6.3	Miembros Sellados.....	42
<b>5.</b>	<b>Diseño metodológico.....</b>	<b>43</b>
<b>6.</b>	<b>Resultados y discusión.....</b>	<b>51</b>
6.1	Programación de calibración los sensores de temperatura en C#.....	52
6.1.1	Error sistemático no ajustado de la temperatura.....	55
6.1.2	Coeficientes de los polinomios interpoladores .....	58
6.1.3	Temperatura ajustada por cada polinomio .....	61
6.1.4	Incertidumbre del ajuste de la temperatura posteriormente al polinomio .....	63
6.1.5	Incertidumbre del ajuste unitario.....	64
6.1.6	factor de cobertura de t Student ( <i>k</i> ).....	66
6.1.7	Temperatura ajustada y su error sistemático .....	67
6.1.8	Análisis de Incertidumbre.....	68
6.2	Programación de calibración de balanza en C# .....	68
6.2.1	Masa nominal de los patrones para la balanza .....	71
6.2.2	Ajuste aparente vs indicada carga ascendente.....	72
6.2.3	Ajuste aparente vs el indicado descendente .....	74
6.2.4	Resultados finales carga ascendente.....	75
6.2.5	Resultados finales carga descendente.....	76

<b>7. Conclusiones y recomendaciones.....</b>	<b>78</b>
<b>Bibliografía.....</b>	<b>80</b>

## Lista de figuras

<b>Figura 1</b> – motor rotativo Wankel.....	20
<b>Figura 2</b> – motor gasolina 4 tiempos.....	21
<b>Figura 3</b> – motor Diesel 4 tiempos inyección directa.....	21
<b>Figura 4</b> – Distribución rectangular de probabilidad.....	25
<b>Figura 5</b> – Ejemplo de clases y objetos.....	29
<b>Figura 6</b> – Ejemplo de encapsulamiento.....	30
<b>Figura 7</b> – Ejemplo de herencia.....	32
<b>Figura 8</b> – Ejemplo de polimorfismo.....	34
<b>Figura 9</b> – Ejemplo de vista.....	35
<b>Figura 10</b> – Ejemplo de controlador.....	36

## Lista de tablas

<b>Tabla 1</b> – temperatura indicada por los sensores vs temperatura patrón .....	44
<b>Tabla 2</b> – Error sistemático no ajustado.....	45
<b>Tabla 3</b> – Análisis de incertidumbre de la temperatura ya ajustada .....	46
<b>Tabla 4</b> – Datos de calibración de la balanza electrónica.....	47
<b>Tabla 5</b> – Resultados finales de la calibración de la balanza con su incertidumbre .....	48
<b>Tabla 6</b> – Plantilla de Carga .....	52
<b>Tabla 7</b> – Datos iniciales de Sensores de Temperatura .....	53
<b>Tabla 8</b> – Resultado de la temperatura ajustada para cada sensor .....	54
<b>Tabla 9</b> – Error sistemático.....	56
<b>Tabla 10</b> – Error sistemático FuelCostUAN.....	57
<b>Tabla 11</b> – coeficientes de los polinomios.....	59
<b>Tabla 12</b> – Comparación de resultados entre polinomios de primer, segundo y tercer grado con las temperaturas registradas por los sensores .....	59
<b>Tabla 13</b> – Resultados de los polinomios en el software.....	60
<b>Tabla 14</b> – Temperatura ajustada por cada polinomio.....	62
<b>Tabla 15</b> – Temperatura ajustada por cada polinomio en el software.....	62
<b>Tabla 16</b> – Incertidumbre del ajuste del polinomio.....	64
<b>Tabla 17</b> – Incertidumbre del ajuste unitario .....	65
<b>Tabla 18</b> – Incertidumbre del ajuste mínimo .....	65
<b>Tabla 19</b> – Factor de cobertura k .....	66
<b>Tabla 20</b> – Temperatura ajustada vs error sistemático.....	67
<b>Tabla 21</b> – Incertidumbre expandida.....	68
<b>Tabla 22</b> – carga descendente con regreso a 0.....	69
<b>Tabla 23</b> – carga ascendente con regreso a 0.....	70
<b>Tabla 24</b> – Masa nominal de los patrones .....	72
<b>Tabla 25</b> – Ajuste aparente vs Indicado ascendente .....	73
<b>Tabla 26</b> – Ajuste aparente vs Indicado ascendente 2 .....	73
<b>Tabla 27</b> – Ajuste aparente vs Indicado descendente .....	74
<b>Tabla 28</b> – Ajuste aparente vs Indicado descendente 2 .....	75
<b>Tabla 29</b> – Resultado final carga ascendente.....	76
<b>Tabla 30</b> – Resultado final carga descendente.....	77

# 1. Introducción

Los avances en la tecnología y la ingeniería han llevado a una constante evolución en el diseño y desarrollo de motores de combustión interna [1,2]. Estos motores desempeñan un papel fundamental en diversas aplicaciones de la industria automotriz y otras áreas relacionadas. Sin embargo, la eficiencia y el rendimiento óptimo de estos motores dependen en gran medida de la precisión y confiabilidad de sus sistemas térmicos y gravimétricos [3].

En este contexto, el TIG que se presenta aborda un proceso de calibración estandarizado diseñado con el propósito de mejorar la precisión y confiabilidad en el sistema térmico y gravimétrico de un motor de combustión interna. Mediante un enfoque investigativo basado en la utilización de instrumentos de medición, se busca lograr una sistematización que facilite la comprensión y organización de los procesos previos de investigación relacionados con la calibración de estos sistemas críticos [4].

La calibración adecuada de los sistemas térmicos y gravimétricos es esencial para asegurar mediciones precisas y minimizar incertidumbres en el rendimiento y eficiencia del motor. Para lograrlo, en este estudio se emplearon pesas previamente certificadas para calcular la incertidumbre asociada a cada herramienta utilizada en el proceso de calibración [5].

El resultado de esta investigación es la creación de un software especializado que mejora significativamente el entendimiento y la interpretación de los datos recopilados durante el proceso de calibración. Este software simplifica el procesamiento de la información y permite generar informes más detallados y comprensibles, lo que se traduce en un avance significativo en el análisis de los resultados y en la obtención de mediciones más precisas y confiables [5].

La adopción de este enfoque estandarizado en la calibración de motores de combustión interna representa un paso crucial hacia una mejora en la investigación y desarrollo de estos sistemas vitales. Asimismo, se espera que los resultados obtenidos contribuyan a un análisis más riguroso y a futuras investigaciones que impulsen la eficiencia y el rendimiento de los motores de combustión interna en diversos campos de aplicación [1-5].



## **1.1 Formulación del problema**

En la Universidad Antonio Nariño, sede Puerto Colombia, los procesos de investigación relacionados con los motores de combustión interna se llevan a cabo de manera manual. Esto implica la calibración de los sensores, la búsqueda de la incertidumbre y la determinación de la incertidumbre de error en las lecturas, todo lo cual se combina con cálculos térmicos para estimar el calor generado por los motores y su valor real. Este proceso se realiza mediante una plantilla de Excel que, aunque actualmente está diseñada para proporcionar respuestas, presenta dificultades en cuanto a su comprensión. La gran cantidad de información presentada en el archivo, junto con su formato editable, hace que los cálculos deban realizarse manualmente y que la información no sea fácilmente legible a simple vista, lo que dificulta su edición sin correr el riesgo de dañarla.

Por este motivo, se ha desarrollado un software que tiene como objetivo automatizar la mayoría de los cálculos y la calibración de los sensores. Este software proporciona resultados precisos y de fácil comprensión, lo que facilita el proceso para el usuario. Además, permite que el usuario tenga un mayor control sobre la calibración de los sensores mediante la introducción de datos de muestreo que se obtienen en el laboratorio de motores de la Universidad Antonio Nariño, sede Puerto Colombia. La intención principal detrás de este software es que sea portable y pueda instalarse en diversos ordenadores sin el temor a obtener resultados incorrectos.

## **1.2 Objetivos**

### **1.2.1 Objetivo general**

Diseñar un software que ayude con los cálculos de análisis térmicos para estimar la incertidumbre de los sensores de temperatura y balanza de los instrumentos de medición con los que cuenta la universidad Antonio Nariño sede puerto Colombia.

### **1.2.2 Objetivos específicos**

Con el propósito de desarrollar el software, se establecieron los siguientes objetivos específicos:

- Llevar a cabo un análisis minucioso de los datos y las plantillas de Excel, evaluando su estructura y contenido, con el fin de implementar el lenguaje de C# y transformarlo en un software funcional.
- Desarrollar un software que incorpore una representación visual de la información y su disposición para su presentación al usuario.
- Desarrollar un software utilizando C# con el propósito de lograr una fácil instalación y portabilidad, permitiendo su implementación en diversos equipos y su uso en cualquier entorno.

### **1.3 Justificación**

Este software tiene como propósito principal optimizar los procesos de investigación, simplificar su comprensión y cálculo, con el objetivo de obtener resultados más precisos y confiables al abordar la incertidumbre asociada a los sensores de temperatura y la balanza. Es imperativo establecer un proceso estandarizado para prevenir errores y asegurar mediciones precisas de los sensores de temperatura, lo que facilita correcciones precisas. Además, el software proporciona una medición más precisa de la incertidumbre en el consumo de combustible en gramos y permite su calibración, lo que facilita la determinación del consumo del motor en un período específico, considerando la incertidumbre inherente a estas mediciones. Estos datos son de vital importancia para evaluar el rendimiento de un motor de combustión interna.

## **2. Funcionamiento del motor de combustión**

A continuación, se presentarán la teoría y los conceptos fundamentales relacionados con el software diseñado con el propósito de impulsar la investigación en este campo específico.

### **2.1 Motores de combustión interna**

Los motores de combustión interna son dispositivos mecánicos que convierten la energía química de la combustión de combustibles en energía mecánica para generar movimiento. Estos motores son ampliamente utilizados en la industria automotriz y en varios campos afines debido a su eficiencia y capacidad para proporcionar una fuente confiable de energía para la propulsión y otras aplicaciones [3]. Hay dos tipos principales de motores de combustión interna: motores de gasolina y motores diésel [3]. Ambos funcionan siguiendo el mismo principio básico de explosión-controlada dentro de una cámara de combustión para generar fuerza mecánica [3]. Aquí hay una descripción general de cada uno:

#### **2.1.1 Motores de gasolina**

También conocidos como motores de ciclo Otto, funcionan utilizando una mezcla de aire y gasolina. En un motor de gasolina, la mezcla se comprime en la cámara de combustión y luego se enciende mediante una chispa de una bujía. Esta ignición provoca una rápida expansión de los gases, lo que empuja un pistón hacia abajo y convierte la energía en movimiento rotativo. Estos motores suelen ser más ligeros y suaves en funcionamiento, lo que los hace comunes en automóviles de pasajeros y vehículos más pequeños [3].

#### **2.1.2 Motores diésel**

Los motores diésel, también conocidos como motores de ciclo diésel, funcionan con una mezcla de aire y combustible diésel. En este caso, el aire es comprimido dentro de la cámara de combustión hasta que alcanza una temperatura lo suficientemente alta como para encender el combustible diésel sin necesidad de una chispa. Al igual que con los motores de gasolina, la explosión generada empuja un pistón y produce movimiento

mecánico. Los motores diésel son conocidos por su eficiencia en el consumo de combustible y su alto torque, lo que los hace ideales para vehículos de carga, camiones, autobuses y maquinaria pesada [3].

## 2.2 Tipos de motores de combustión interna

Para los motores de combustión interna existen varios tipos de ciclos dentro de los que se encuentran el ciclo otto que es el de gasolina y el ciclo Diesel y sus diferencias pueden ir en la forma como encienden el combustible y en que tiempos lo pueden emplear.

También podemos encontrar ciclos 2 tiempos donde se puede tener en la carrera de admisión puede hacer el escape mediante lumbreras que ayudan al ingreso de gasolina, aceite y aire al motor, es uno de los motores más contaminantes al quemar aceite y gasolina al tiempo creando  $CO_2$  en mayores cantidades que los motores de 4 tiempos, otro tipo de motor que se tiene dentro de los más conocidos es el motor rotativo que usa en vez de un pistón, biela y cigüeñal, tiene un rotor en forma triangular que produce el movimiento de forma rotatoria y no lineal como lo hace el motor de ciclo otto.

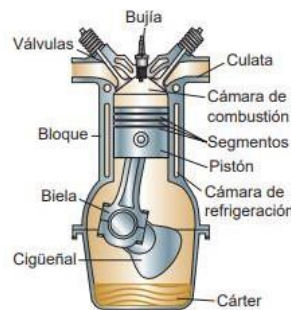


**Figura 1** – motor rotativo Wankel

Fuente: Diego López Donaire (16/11/2021) [6]

Otro de los motores más usados y el más común es el motor de gasolina es uno de los más usados dentro del sector automotriz y el más destacado por tener una eficiencia aceptable vs las emisiones lo que lo hace tan popular es que tiene muchas variantes dentro de los diferentes modelos y marcas pero su principio de funcionamiento es el mismo el

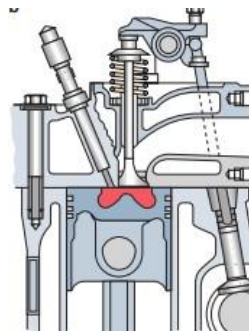
cual consiste en mover un tren motriz que está conformado por pistones , bielas , cigüeñal y dentro de estos un sin fin de complementos que hacen que un movimiento lineal sea convertido en rotativo partiendo del funcionamiento de la combustión de aire , gasolina y una chispa la cual produce una explosión en la cámara donde esta se alberga a una cierta compresión y como resultado obtenemos unos gases calientes que contienen partículas de  $CO_2$  y otros tipos de gases incluyendo residuos de combustible [7].



**Figura 2** – motor gasolina 4 tiempos

Fuente: Santiago Sanz Acebes [7]

Otro de los motores más usados en el sector industrial es el motor Diesel y a diferencia de los de gasolina estos son más eficientes , por otro lado la relación de compresión es más alta lo cual enciende el combustible por auto detonación lo que no requiere bujías de encendido , por otro lado en estos motores el combustible es inyectado directamente a la cámara de combustión o indirectamente según diseño y gusto del fabricante en el momento en que el pistón asciende en el proceso de compresión lo que en otros ciclos es necesario encender la mezcla con una chispa para este caso el motor no requiere sino de una buena compresión y una buena pulverización del combustible en la cámara de encendido del pistón [7]



**Figura 3** – motor Diesel 4 tiempos inyección directa

Fuente: Santiago Sanz Acebes [7]

## **3. Fundamentos de la incertidumbre**

El análisis de la incertidumbre enfatiza en comprender y de una u otra manera cuantificar los riesgos y la falta de certezas en varias situaciones, Es utilizado donde las precisiones de los resultados dependen de muchas variables, la evaluación de confiabilidad será basada en la medición de resultados obtenidos en el motor de combustión interna, estos resultados serán introducidos en el software propuesto el cual busca optimizar y aumentar la confiabilidad de los resultados.

### **3.1 Incertidumbre de medición**

La incertidumbre de medición es una medida que indica la viabilidad o el margen de error asociado con el valor obtenido al realizar una medición, “si la Incertidumbre está bien calculada y es suficientemente pequeña podemos decir que la medición es de buena calidad”. Por tanto, el resultado de una medición no está completo si no posee una declaración de la incertidumbre de la medición con un nivel de confianza determinado, o en otras palabras, se debe incluir información numérica que indique la cuantía de la duda acerca del resultado” [1]. La incertidumbre abarca varias componentes, las cuales pueden ser estimadas por dos tipos de evaluación, Una evaluación Tipo A y una evaluación Tipo B. En esta sección se utilizarán otros términos relacionados con la incertidumbre.

#### **3.1.1 Calibración**

La calibración se puede definir como “Conjunto de operaciones que establecen, bajo condiciones específicas, la relación entre los valores de las magnitudes que indiquen un instrumento de medición o un sistema de medición, o valores representados por una medida materializada o por un material de referencia y valores correspondientes determinados por medio de los patrones” [2].

#### **3.1.2 Medición**

Es una observación cuantitativa indicando un numero de determinadas características [4].

### **3.1.3 Incertidumbre Estándar**

Incertidumbre del resultado de una medición expresada como una desviación estándar [3].

### **3.1.4 Patrón**

Es un instrumento utilizado como referencia para obtener diferentes valores medidos e incertidumbres de medida que son asociadas para otras magnitudes de la misma naturaleza, de esta manera se logra establecer un seguimiento metrológico, a través de la calibración de otros patrones, instrumentos o sistemas de medidas [5].

### **3.1.5 Precisión**

Es el grado de conformidad que caracteriza las medidas independientes, generado por la aplicación repetida del proceso de medición bajo unas específicas condiciones.

### **3.1.6 Exactitud**

Es el grado de conformidad entre el valor real y el observado.

### **3.1.7 Error Sistemático**

Son los errores que dan a lugar siempre a una desviación del valor real en el mismo sentido, afectando de esta manera la exactitud.

### **3.1.8 Error Aleatorio**

Son los errores que no pueden predecirse y varían tanto en la magnitud como en el sentido, logrando afectar la exactitud

## **3.2 Método general del cálculo**

La incertidumbre de medida incluye algunos componentes que proceden de efectos sistemáticos, como son las correcciones y los patrones, así como la incertidumbre resultante de la definición.

Incluye diversos componentes como es la evaluación de la incertidumbre de las medidas de las estimaciones de entradas. [6]

### 3.3 Evaluación de incertidumbre Tipo A

Método para evaluar la incertidumbre mediante el análisis estadístico de una serie de observaciones [3].

La ecuación (1) corresponde al valor medio de una muestra, el cual es utilizado para, posteriormente, realizar una evaluación de un componente de la incertidumbre. La influencia que tiene en este resultado es aleatorio.

$$\bar{x} = \frac{1}{n} (x_1 + x_2 + \dots + \dots) = \left(\frac{1}{n}\right) \sum_{k=1}^n x_k \quad (1)$$

Las mediciones  $x_k$  varían por las magnitudes de fluencias, por ejemplo, la temperatura, humedad, entre otras. Están en variación constante.

La incertidumbre Tipo A puede ser calculada a partir de la ecuación (2):

$$s(x_k) = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2} / n \quad (2)$$

### 3.4 Evaluación de incertidumbre Tipo B

Método para evaluar la incertidumbre por otro medio que no sea el análisis estadístico de una serie de observaciones [3].

Puede llegar a ser más confiables que la evaluación Tipo A cuando la muestra de información es más grande.

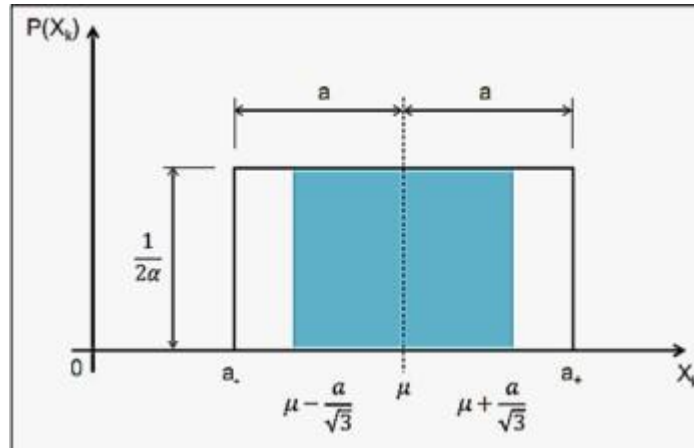
Ejemplo son las evaluaciones basadas en las informaciones obtenidas.

- Resultados de investigaciones
- Especificaciones del fabricante
- Datos certificados de calibración



### 3.5 Distribución Rectangular

La distribución rectangular es la distribución de probabilidad en la que todos los valores dentro de un intervalo tienen la misma probabilidad de ocurrir. Es decir; la probabilidad es constante dentro del intervalo y cero fuera de él.



**Figura 4** – Distribución rectangular de probabilidad

Fuente: ESTUDO DAS INCERTEZAS NA MEDIÇÃO DE ESPESSURA POR ULTRASSOM EM PEÇAS AERONÁUTICAS (Bruno César de Andrade Silva-2012)

Se deduce que la altura definida como  $f(x)$  es denominada con la siguiente ecuación

$$f(x) = \frac{1}{2a} \tag{3}$$

Se deduce un estimado para el cálculo de la incertidumbre cuando es asociado a los datos experimentales de una distribución rectangular de la probabilidad.

La varianza es definida por la siguiente ecuación.

$$s^2(x) = E(x^2) - [E(x)]^2 \tag{4}$$

$$s(x) = \sqrt{E(x^2) - [E(x)]^2} \tag{5}$$

Ecuación del valor esperado

$$E(x^2) = \int_{-\infty}^{+\infty} x^2 \cdot f(x) dx = \int_0^{2a} x^2 \cdot \frac{1}{2a} dx = \frac{1}{2a} \cdot \frac{x^3}{3} \Big|_0^{2a} = \frac{4a^3}{3} \tag{6}$$

Por último, la ecuación que denota la desviación estándar asociada a una distribución rectangular de probabilidad.

$$s(x) = \frac{a}{\sqrt{3}} \quad (7)$$

### 3.6 Distribución normal y t-student

La distribución normal también conocida como la Gaussiana, consiste en la distribución de probabilidades para las variables continuamente variables continuas presentan una gráfica en forma de campana en función de la densidad.

Su principal factor de importancia se debe a que hay muchas variables asociadas a fenómenos naturales que siguen el modelo de la normal.

Función de la densidad es la siguiente:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}; -\infty < x < \infty \quad (8)$$

Donde:

$\mu$ : Media;  $\sigma$ : Desviación;  $\pi = 3,1415$ ; e: 2.7182; x: abscisa.

### 3.7 Incertidumbre estándar combinada

Incertidumbre estándar del resultado de una medición cuando el resultado se obtiene a partir de los valores de algunas otras magnitudes, igual a la raíz cuadrada positiva de una suma de términos, siendo estos términos las varianzas y covarianzas de estas otras magnitudes ponderadas de acuerdo cómo el resultado de la medición varía con respecto a cambios en estas magnitudes [3].

### 3.8 Incertidumbre expandida

Cantidad que define un intervalo alrededor de una medición del que se puede esperar que abarque una fracción grande de la distribución de valores que razonablemente pudieran ser atribuidos al mensurando [3].

## 4. Fundamentos de programación

Los fundamentos de programación representan el cimiento esencial sobre el cual se construye el arte de dar vida a las máquinas. Son como las partículas subatómicas de nuestro universo digital, aquellas que, en su simplicidad aparente, permiten la creación de mundos complejos de lógica y funcionalidad. En la búsqueda de comprender estos fundamentos, nos aventuramos en un viaje donde los algoritmos se convierten en las estrellas, las variables y estructuras de datos en los planetas, y los bucles y condicionales en las leyes de la física que rigen nuestro programa. Estos fundamentos, como las piezas de un rompecabezas, se entrelazan para dar forma a aplicaciones y sistemas informáticos, exploraremos su importancia, su influencia en la resolución de problemas y su papel en la creación de soluciones innovadoras en el mundo de la programación [8].

### 4.1 Programación orientada a objetos

La programación orientada a objetos (POO) es un paradigma de programación que se basa en el concepto de "objetos". Un objeto es una entidad que agrupa datos (atributos) y funciones (métodos) que operan sobre esos datos. La POO se utiliza para modelar el mundo real de una manera más cercana, lo que la hace muy útil en el desarrollo de software complejo y en la resolución de problemas [9].

#### 4.1.1 Clases y Objetos

En la POO, las clases son los planos o moldes que definen la estructura de los objetos. Un objeto es una instancia de una clase, lo que significa que se crea a partir de la plantilla proporcionada por la clase. Por ejemplo, si tenemos una clase llamada "Coche", un objeto de esa clase podría ser un coche específico con sus propias características y comportamientos [9].

La programación orientada a objetos es una metodología crucial para la creación de sistemas de software. Este método se basa en la idea de que el diseño y modelado del software pueden ser similares a cómo percibimos y entendemos el mundo real. Para crear

un sistema que cumpla con los requisitos previamente establecidos, se siguen varias etapas que involucran una variedad de actividades [24].

El uso de un estilo de programación específico que se centra en el uso de objetos y sus interacciones es una de las técnicas más importantes de programación orientada a objetos. Estos objetos pueden agruparse en clases, que funcionan como moldes para crear instancias de objetos individuales. La programación orientada a objetos tiene como objetivo modelar el mundo real en el que todo es considerado un objeto [24]. Por ejemplo, al mirar a nuestro alrededor, podemos reconocer que cada objeto, desde una silla en la que nos sentamos hasta un lápiz en nuestra mano, es un objeto [24]. Al modelar un sistema de software, la programación orientada a objetos busca reflejar esto.

El elemento principal de este enfoque es el objeto en sí mismo. Cada objeto, como en la realidad, tiene características distintivas que lo distinguen de otros objetos similares [24]. Por ejemplo, las características de tres canicas podrían ser el tamaño y el color. Todas las canicas pueden tener un tamaño y un color diferentes, aunque sean iguales. Estas características se denominan atributos en la programación orientada a objetos [24].

Además de tener características, los objetos también pueden realizar una variedad de acciones o tareas específicas. Un lápiz, por ejemplo, tiene la capacidad de dibujar, borrar y escribir. En la programación orientada a objetos, los métodos se refieren a estas acciones que los objetos pueden realizar [24].

Los objetos similares se organizan en clases de acuerdo con una analogía con la realidad. Por ejemplo, podemos definir una clase llamada "felino" que incluye una variedad de animales, como gatos, tigres, leopardos y leones. Cada felino tiene su propio color, peso, tipo de alimentación y tamaño, así como su habilidad para comer, dormir y correr. En la programación orientada a objetos, la clase "felino" representa la categoría general, mientras que los objetos individuales como el gato, el tigre, el leopardo y el león son instancias de esa clase, cada uno con sus propios atributos y métodos [24].

En resumen, la programación orientada a objetos se basa en la idea de crear objetos, clases, atributos y métodos para modelar el mundo real. Este método ofrece una forma organizada y efectiva de diseñar y desarrollar sistemas de software que reflejen características y comportamientos del mundo real [24].

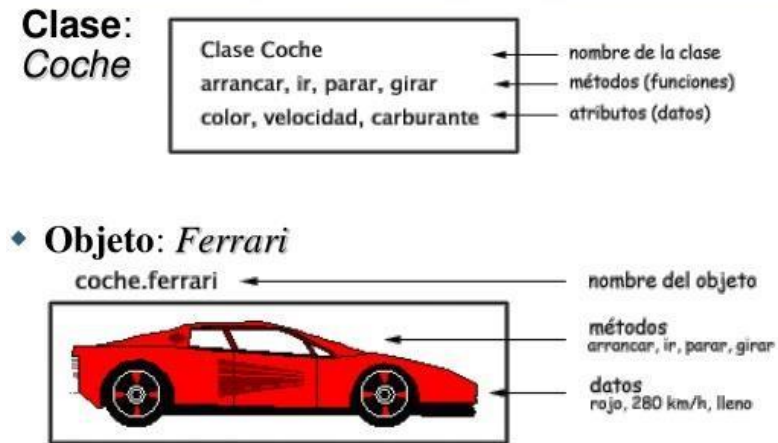


Figura 5 – Ejemplo de clases y objetos

Fuente: Alquicira, C. (2017, 05 de noviembre) Programación Orientada a Objetos. Conogasi, Conocimiento para la vida.  
Fecha de consulta: septiembre 24, 2023

## 4.1.2 Encapsulación

La encapsulación es un principio clave en la POO que se refiere a ocultar los detalles internos de un objeto y exponer solo la funcionalidad necesaria. Esto se logra utilizando modificadores de acceso como "public", "private" y "protected" para controlar el acceso a los atributos y métodos de una clase [9].

Esto permite que algunos atributos se puedan modificar al ser públicos para que los objetos puedan ser llamados en otra parte del código o puede ser privado para que solo sea usado en una operación específica del objeto y puede ser protegido puede tener acceso, pero es limitado para las operaciones [25].

- C#:

```
class Square
{
    private double side;

    public double Side
    {
        get
        {
            return side;
        }

        set
        {
            side = value;
        }
    }
}

class Program
{
    static void Main(string[] args)
    {
        Square s = new Square();
        s.Side = 1.23;
    }
}
```

**Figura 6** – Ejemplo de encapsulamiento.

Fuente: Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/visualstudio/ide/reference/encapsulate-field?view=vs-2022>

### 4.1.3 Herencia

La herencia permite que una clase herede atributos y métodos de otra clase. Esto fomenta la reutilización de código y la creación de jerarquías de clases. Por ejemplo, una clase "Animal" podría tener subclases como "Perro" y "Gato" que heredan sus características generales [26].

En la programación orientada a objetos, la herencia es un concepto fundamental que permite la creación de clases secundarias que reutilizan, amplían o modifican el comportamiento de una clase primaria. La clase primaria cuyos miembros son heredados se denomina clase base, mientras que la clase derivada es la clase primaria cuyos miembros son heredados [26].

La herencia única es permitida en C# y .NET, lo que significa que una clase puede heredar de una sola clase. Pero la herencia es transitiva, lo que permite establecer una jerarquía

de herencias en la que un tipo D puede heredar de un tipo C, que a su vez puede heredar de un tipo B, etc. Esto indica que debido a la transitividad de la herencia, los miembros de la clase base A estarán disponibles para la clase derivada D [26].

No obstante, existen ciertos miembros que no se heredan en las clases derivadas, como los constructores estáticos, que inicializan datos estáticos, los constructores de instancias, que se utilizan para crear nuevas instancias de la clase, y los finalizadores, que son llamados por el recolector de elementos no utilizados para destruir instancias de una clase. Cada clase debe definir sus propios constructores [26].

Los miembros heredados en las clases derivadas tienen diferentes niveles de visibilidad. Los miembros privados solo son visibles en las clases derivadas que están anidadas en su clase base; de lo contrario, no son accesibles en las clases derivadas. Esto implica que la accesibilidad de un miembro tiene un impacto en su visibilidad en las clases derivadas [26].

En resumen, la herencia en la programación orientada a objetos permite la reutilización y modificación del comportamiento de clases primarias en clases secundarias. C# y .NET admiten herencia única y transitiva, pero algunos miembros no se heredan, y la visibilidad de los miembros heredados en las clases derivadas depende de su nivel de accesibilidad [26].

```
public class A
{
    private int _value = 10;

    public class B : A
    {
        public int GetValue()
        {
            return _value;
        }
    }
}

public class C : A
{
    // public int GetValue()
    // {
    //     return _value;
    // }
}

public class AccessExample
{
    public static void Main(string[] args)
    {
        var b = new A.B();
        Console.WriteLine(b.GetValue());
    }
}

// The example displays the following output:
//     10
```

**Figura 7** – Ejemplo de herencia.

Fuente: Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/tutorials/inheritance>

#### 4.1.4 Polimorfismo

El polimorfismo permite que objetos de diferentes clases respondan de manera similar a mensajes o métodos. Esto se logra mediante el uso de interfaces y clases abstractas, lo que facilita la implementación de múltiples clases que cumplen con una interfaz común [27].

El polimorfismo, que se considera el tercer pilar de la programación orientada a objetos, permite que los objetos de clases derivadas se comporten como objetos de una clase base en tiempo de ejecución [27]. Esto se logra a través de dos componentes:

Los objetos de una clase derivada pueden tratarse como objetos de una clase base en tiempo de ejecución en una variedad de contextos, como parámetros de métodos o



elementos de colección. Esto implica que, en tiempo de ejecución, el tipo declarado del objeto ya no es necesariamente el mismo que su tipo declarado [27].

Las clases base pueden establecer métodos virtuales, y con su propia implementación, las clases derivadas pueden anular o sobre escribir estos métodos virtuales. El tiempo de ejecución, cuando se llama a un método virtual en el código cliente, busca el tipo real del objeto y ejecuta la versión del método definido en la clase derivada [27].

Este método de polimorfismo permite trabajar uniformemente con grupos de objetos relacionados. Por ejemplo, puede crear una jerarquía de clases en una aplicación de dibujo que incluya una clase base "Forma" y clases derivadas como "Rectangular", "Círculos" y "Triángulo". Cada una de estas clases proviene de la clase base "Forma" y sobrescribe el método virtual "Dibujar" para dibujar su forma particular. Después, se pueden almacenar objetos de estas clases en una lista (List) y, gracias al polimorfismo, se puede utilizar el método "Draw" en cada objeto de la lista sin tener conocimiento previo del tipo de forma [27].

En resumen, el polimorfismo permite que los objetos de clases derivadas se comporten como objetos de una clase base en tiempo de ejecución, lo que facilita trabajar de manera uniforme con objetos de una jerarquía de clases, como en el caso de una aplicación de dibujo que puede manipular diferentes tipos de formas sin conocer sus detalles específicos [27].

```
public class Shape
{
    // A few example members
    public int X { get; private set; }
    public int Y { get; private set; }
    public int Height { get; set; }
    public int Width { get; set; }

    // Virtual method
    public virtual void Draw()
    {
        Console.WriteLine("Performing base class drawing tasks");
    }
}

public class Circle : Shape
{
    public override void Draw()
    {
        // Code to draw a circle...
        Console.WriteLine("Drawing a circle");
        base.Draw();
    }
}

public class Rectangle : Shape
{
    public override void Draw()
    {
        // Code to draw a rectangle...
        Console.WriteLine("Drawing a rectangle");
        base.Draw();
    }
}

public class Triangle : Shape
{
    public override void Draw()
    {
        // Code to draw a triangle...
        Console.WriteLine("Drawing a triangle");
        base.Draw();
    }
}
```

**Figura 8** – Ejemplo de polimorfismo.

Fuente: Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/object-oriented/polymorphism>

## 4.2 Descripción de modelo vista controlador

El Modelo-Vista-Controlador (MVC) es un paradigma de diseño ampliamente utilizado en el desarrollo de aplicaciones de software. Su objetivo principal es separar las preocupaciones de una aplicación en tres componentes claramente definidos: el Modelo, la Vista y el Controlador. Esta separación facilita la organización, el mantenimiento y la

escalabilidad del código, lo que lo convierte en una elección popular para desarrollar aplicaciones de software complejas [10].

### 4.2.1 Modelo (Model)

El Modelo representa la capa de datos y la lógica de negocio de la aplicación. Contiene la información sobre el estado de la aplicación y las operaciones que se pueden realizar sobre estos datos [10].

En el contexto de una base de datos, el Modelo se encarga de acceder y gestionar los datos almacenados [10].

El Modelo no está directamente relacionado con la interfaz de usuario y no tiene conocimiento de cómo se presenta la información [10].

### 4.2.2 Vista (View)

La Vista es la capa encargada de mostrar la información al usuario y de recibir sus interacciones. Presenta los datos proporcionados por el Modelo de una manera que sea comprensible y significativa para el usuario [10].

Puede haber múltiples vistas que representen los mismos datos en formatos diferentes, como una vista de tabla y una vista de gráfico en una aplicación de gestión financiera.

La Vista no realiza ningún procesamiento de datos ni lógica de negocio. Su único propósito es la presentación [10].

```
<%@ Page Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc"
    <asp:Content ID="aboutContent" ContentPlaceHolderID="MainContent" runat="server">
      <h2>About</h2>
      <p>
        Put content here.
      </p>
    </asp:Content>
```

**Figura 9** – Ejemplo de vista.

Fuente: Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/aspnet/mvc/overview/older-versions-1/overview/understanding-models-views-and-controllers-cs>

### 4.2.3 Controlador (Controller)

El Controlador actúa como intermediario entre la Vista y el Modelo. Recibe las interacciones del usuario a través de la Vista y, en función de esas interacciones, realiza operaciones en el Modelo [10].

También puede actualizar la Vista para reflejar los cambios en el Modelo, la lógica de flujo de control y las decisiones se implementan en el Controlador, en una aplicación web, por ejemplo, el Controlador manejaría las solicitudes HTTP, procesaría la entrada del usuario y actualizaría el Modelo y la Vista según corresponda [10].

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MvcApplication1.Controllers
{
    [HandleError]
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewData["Title"] = "Home Page";
            ViewData["Message"] = "Welcome to ASP.NET MVC!";

            return View();
        }

        public ActionResult About()
        {
            ViewData["Title"] = "About Page";

            return View();
        }
    }
}
```

**Figura 10** – Ejemplo de controlador.

Fuente: Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/aspnet/mvc/overview/older-versions-1/overview/understanding-models-views-and-controllers-cs>

## 4.3 Administración de Windows Forms de .net

La programación de Windows Forms en .NET es un enfoque para crear aplicaciones de escritorio para el sistema operativo Windows. Windows Forms es parte de la plataforma .NET de Microsoft y proporciona una interfaz gráfica de usuario (GUI) basada en ventanas

que permite a los desarrolladores crear aplicaciones de Windows con una interfaz de usuario rica y altamente interactiva. A continuación [11], se presenta un resumen detallado de su funcionamiento.

### **4.3.1 Diseño de la Interfaz Gráfica de Usuario (GUI)**

La programación de Windows Forms comienza con el diseño de la interfaz gráfica de usuario de la aplicación. Esto se logra arrastrando y soltando controles (como botones, cuadros de texto, listas desplegables, etc.) desde la caja de herramientas en la ventana de diseño.

Los controles se colocan y se ajustan en una ventana o formulario que representa la interfaz de la aplicación [11].

### **4.3.2 Propiedades y Eventos**

Cada control en Windows Forms tiene propiedades que se pueden configurar para controlar su apariencia y comportamiento. Por ejemplo, puede establecer el texto de un botón, su color de fondo y sus dimensiones, además de las propiedades, los controles tienen eventos asociados, como "Click" para un botón. Los eventos permiten que la aplicación responda a las interacciones del usuario, como hacer clic en un botón o escribir en un cuadro de texto [11].

### **4.3.3 Programación y Lógica de la Aplicación:**

La lógica de la aplicación se implementa mediante el manejo de eventos. Cuando un usuario interactúa con un control, se desencadena un evento, y se puede escribir código para responder a ese evento, los desarrolladores pueden utilizar lenguajes de programación como C# o Visual Basic .NET para escribir el código detrás de la interfaz de usuario. Este código puede incluir operaciones como cálculos, acceso a bases de datos, llamadas a servicios web y más [11].

### **4.3.4 Compilación y Ejecución**

Una vez que se ha diseñado la interfaz de usuario y se ha escrito la lógica de la aplicación, el proyecto se compila en un archivo ejecutable (.exe) que los usuarios pueden abrir y ejecutar en sus computadoras con Windows [11].

La aplicación Windows Forms se ejecuta en el contexto del sistema operativo Windows y aprovecha la infraestructura de Windows para funciones como el manejo de ventanas, el enrutamiento de eventos y la administración de recursos [11].

### **4.3.5 Manejo de la Interfaz de Usuario:**

Windows Forms se encarga del manejo de la interfaz de usuario, lo que significa que gestiona la creación y administración de ventanas y controles, los desarrolladores pueden utilizar las características de Windows Forms para controlar la apariencia de la aplicación, como cambiar temas, personalizar iconos y gestionar la disposición de los controles [11].

## **4.4 variables en C# usadas en .NET**

Una variable es un contenedor utilizado en C# para almacenar y modificar datos en un programa. Cada variable tiene un tipo de dato que especifica qué tipo de valores puede contener. Estos tipos de dato pueden incluir números enteros, números de punto flotante, texto, caracteres y así sucesivamente. En el código, las variables tienen un nombre junto con su tipo de dato [12].

### **4.4.1 Variable Tipo de Dato**

En C#, cada variable tiene un tipo de dato que especifica el tipo de valores que se pueden almacenar [12]. Algunos tipos de datos comunes en C# son:

Int: Enteros.

Float y double: Números de punto flotante.

String: Cadenas de texto.

Char: Caracteres individuales.

Bool: Valores booleanos (verdadero o falso).

Y muchos otros tipos personalizados.

#### **4.4.2 Nombre de Variable**

El nombre de la variable es un identificador único utilizado en el código para hacer referencia a la variable. Debe cumplir con algunas reglas, como comenzar con una letra o un guion bajo, no dejar espacios en blanco y ser descriptivo para que el código sea fácil de entender [12].

#### **4.4.3 Declaración**

Para usar una variable, debes declararla primero. El tipo de dato y el nombre de la variable se especifican en la declaración. Después de la declaración, la variable se encuentra en la memoria, pero carece de un valor específico [12].

#### **4.4.4 Asignación de Valor**

Para darle un valor a una variable, se utiliza el operador de asignación (=). Esto asigna un valor del tipo de dato correspondiente a la variable. Por ejemplo, edad = 30; asigna el valor 30 a la variable edad [12].

#### **4.4.5 Uso de Variables**

Una variable puede usarse en expresiones y operaciones del programa una vez que tiene un valor asignado. Por ejemplo, se pueden concatenar cadenas de texto, realizar cálculos matemáticos utilizando variables o tomar decisiones basadas en el valor de una variable [12].

#### **4.4.6 Alcance de Variables**

En C#, las variables tienen un alcance que determina dónde son válidas. Por ejemplo, solo una variable declarada dentro de una función puede ser vista y accesible. Esto evita conflictos de nombres de variables en partes diferentes del código [12].

### **4.4.7 Inicialización**

La inicialización de las variables cuando se declaran, es decir, asignarles un valor inicial, es una buena práctica. Esto evita que las variables tengan valores inapropiados antes de que se les otorgue un valor real [12].

### **4.4.8 4.4.8 Tipos de Variables**

Además de los tipos de datos básicos mencionados anteriormente, C# permite la creación de estructuras (structs) y clases (classes), lo que permite una mayor encapsulación y abstracción de datos y funcionalidad [12].

## **4.5 Reglas de los principios de SOLID**

Robert C. Martin creó los principios SOLID, un conjunto de cinco pautas de diseño de software que se enfocan en la creación de código fuente limpio, modular y mantenible. Cada principio se centra en un aspecto particular del diseño de software y proporciona pautas para escribir código de alta calidad. Cada uno de los cinco principios SOLID se detalla a continuación [13].

### **4.5.1 Principio de Responsabilidad Única (SRP - Single Responsibility Principle)**

Este principio establece que una clase debe tener una sola responsabilidad, lo que significa que debe tener una sola razón para cambiar. En otras palabras, una clase debe hacer algo y hacerlo correctamente [13]. Al dividir las responsabilidades en clases independientes, se facilita el mantenimiento y la evolución del código porque los cambios en una responsabilidad no afectarán a otras partes del sistema [13].

### **4.5.2 Principio de Abierto/Cerrado (OCP - Open/Closed Principle)**

Según el OCP, las clases y módulos deben estar abiertos para la modificación, pero cerrados para la extensión. Esto significa que es posible incorporar nuevas funcionalidades al sistema sin alterar el código actual [13.] Esto se logra mediante la creación de interfaces



y abstracciones que permiten la extensión implementando nuevas clases o módulos en lugar de modificar el código existente [13].

### **4.5.3 Principio de Sustitución de Liskov (LSP - Liskov Substitution Principle)**

Los objetos de una clase derivada deben poder reemplazar a los objetos de la clase base sin comprometer la integridad del programa, según LSP. En otras palabras, si una clase B hereda de una clase A, entonces B puede reemplazar a A en cualquier lugar sin alterar el comportamiento esperado del programa. Esto garantiza la coherencia y la interoperabilidad del sistema [13].

### **4.5.4 Principio de Segregación de Interfaces (ISP - Interface Segregation Principle)**

Según el ISP, una interfaz no debe tener más métodos de los que necesita su cliente. Se deben crear interfaces específicas que se adapten a las necesidades de las clases que las implementan en lugar de crear interfaces monolíticas utilizando una variedad de métodos. Esto evita que las clases utilicen métodos que no utilizan y fomenta la cohesión del código [13].

### **4.5.5 Principio de Inversión de Dependencia (DIP - Dependency Inversion Principle)**

Según el DIP, los módulos de bajo nivel no deben depender de los módulos de alto nivel; en cambio, ambos deben depender de abstracciones. Además, los detalles no deben depender de las abstracciones, sino de las abstracciones [13]. Esto se logra mediante el uso de interfaces y abstracciones para desacoplar los elementos de un sistema, lo que facilita la sustitución y evolución de los componentes individuales [13].

## **4.6 Miembros de clase y las clases abstractas y selladas en C#**

Los conceptos clave relacionados con clases y miembros de clase abstractos y sellados se introducen en la guía de programación de C# [14], que son esenciales para crear código modular y seguro. La siguiente es una descripción detallada de estos conceptos:

### **4.6.1 Clases Abstractas**

Una característica importante de C# es la clase abstracta, que permite definir clases que no pueden ser instanciadas directamente, lo que significa que no se pueden crear objetos directamente a partir de ellas [14]. En cambio, se utilizan como base para clases derivadas que realizan funcionalidades particulares. La palabra "abstract" se utiliza para describir las clases abstractas, que pueden incluir métodos abstractos (sin implementación) o concretos. Las clases derivadas deben usar las técnicas abstractas de una clase abstracta [14].

### **4.6.2 Clases Selladas**

Una clase sellada, también conocida como "clase sellada" en inglés, es una que no puede ser heredada por otras clases [14]. Esto implica que no es posible crear clases derivadas de una clase sellada. La palabra clave "sellada" se usa para identificar una clase como sellada. Cuando se desea evitar que otras clases amplíen o alteren la funcionalidad de la clase sellada, se realiza esto. Las clases selladas son útiles para garantizar la inmutabilidad y la seguridad de ciertos componentes del código [14].

### **4.6.3 Miembros Sellados**

Es posible sellar partes específicas de una clase, como métodos o propiedades, además de una clase en su conjunto [14]. Esto se logra mediante el uso de la palabra clave "cerrado" y la definición del miembro en una clase derivada. Al sellar un miembro, las clases derivadas no pueden cambiar o anular ese miembro [14]. Esto es útil para garantizar que las clases derivadas no alteren ciertas partes del comportamiento de una clase [14].

## 5. Diseño metodológico

Para llevar a cabo este proyecto, se enfocó en el análisis, diseño y desarrollo de un software basado en el estudio de los datos de un trabajo previo que se realizó en una plantilla de Excel. Dicha plantilla había sido creada por un grupo de investigación en el contexto de un proyecto de investigación sobre un motor de combustión interna. En ese proyecto, se analizó la incertidumbre presente en la medición del consumo de combustible en gramos y la temperatura de funcionamiento del motor en diferentes sistemas de operación, tomando en cuenta la precisión de la balanza y los sensores de temperatura [15].

La plantilla de Excel se utilizó como punto de partida, lo que proporcionó una visión inicial de los datos que debían ser presentados en la interfaz del software. Esta plantilla, denominada "FuelCostUAN", definió cómo esta información sería visualizada por el usuario. El software también se complementó con un lenguaje de programación que permitió estructurar los resultados de manera organizada, evitando posibles errores que pudieran surgir por un uso incorrecto del programa. La precisión en la presentación de los datos se estableció como una de las fortalezas primordiales del software, asegurando una representación precisa de la información [15].

Además, el software se respaldó con un lenguaje de programación que facilitó la ejecución de instrucciones precisas y eficientes. De esta manera, se creó un entorno en el cual el usuario podría tomar decisiones cruciales en relación a los resultados de la investigación [15].

**Objetivo específico #1:** □ *Llevar a cabo un análisis minucioso de los datos y las plantillas de Excel, evaluando su estructura y contenido, con el fin de implementar el lenguaje de C# y transformarlo en un software funcional.*

**Etapa 1:** *en esta etapa de la investigación se analiza los resultados de la plantilla de Excel con los resultados obtenidos corrigiendo la incertidumbre de la temperatura y el peso del combustible en gramos.*

Al revisar los datos de la "Tabla 1", se evidenció que parte de los resultados obtenidos en Excel podían ser modificados. La tarea principal consistió en estandarizar el proceso para

evitar errores. Desde esta tabla surgieron las muestras de temperaturas tomadas de los sensores y los resultados obtenidos fueron los siguientes:

**Tabla 1** – temperatura indicada por los sensores vs temperatura patrón

No Puntos	Hora	S1	S2	S3	S4	S5	S6	Temperatura del Patrón de Medición			
		Temperaturas indicadas por los sensores						Medida 1	Medida 2	Valor medio	Desviación
		T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	°C	°C	°C	°C
	h:min:s	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C
1	3:56:09 p. m.	7,1	7,0	7,2	6,6	6,5	6,6	6,1	6,3	6,2	0,1414
2	4:00:00 p. m.	9,4	9,3	9,5	8,8	8,7	8,8	8,5	8,5	8,5	0,0000
3	4:09:40 p. m.	11,8	11,6	11,8	11,2	11,1	11,2	10,7	10,9	10,8	0,1414
4	11:07:04 a. m.	14,3	14,1	14,4	13,7	13,6	13,7	13,1	13,3	13,2	0,1414
5	4:16:50 p. m.	14,7	14,6	14,8	14,1	14,0	14,1	13,8	13,8	13,8	0,0000
6	11:13:28 a. m.	16,0	15,8	16,1	15,4	15,3	15,4	14,9	14,9	14,9	0,0000
7	11:18:05 a. m.	16,4	16,3	16,5	15,8	15,7	15,7	15,3	15,4	15,4	0,0707
8	4:23:58 p. m.	16,5	16,3	16,6	15,9	15,8	15,9	15,4	15,5	15,5	0,0707
9	4:30:36 p. m.	19,4	19,3	19,5	18,8	18,7	18,8	18,5	18,5	18,5	0,0000
10	4:39:35 p. m.	21,6	21,5	21,7	21,0	20,9	21,0	20,5	20,6	20,6	0,0707
11	11:26:20 a. m.	23,4	23,3	23,5	22,8	22,7	22,8	22,2	22,3	22,3	0,0707
12	4:43:51 p. m.	24,2	24,0	24,3	23,6	23,5	23,5	23,1	23,2	23,2	0,0707
13	4:50:09 p. m.	26,8	26,6	26,8	26,1	26,0	26,1	25,6	25,7	25,7	0,0707
14	11:46:10 a. m.	26,8	26,6	26,9	26,2	26,1	26,1	25,7	25,7	25,7	0,0000
15	11:56:35 a. m.	31,3	31,1	31,3	30,6	30,5	30,6	30,1	30,1	30,1	0,0000
16	4:54:47 p. m.	31,1	31,0	31,3	30,5	30,4	30,5	30,1	30,1	30,1	0,0000
17	12:09:01 p. m.	36,3	36,1	36,3	35,6	35,5	35,6	35,0	35,0	35,0	0,0000
18	12:17:00 p. m.	39,2	39,1	39,3	38,6	38,5	38,6	38,0	38,0	38,0	0,0000
19	12:26:14 p. m.	41,2	41,1	41,3	40,6	40,5	40,6	40,0	40,0	40,0	0,0000
20	12:31:09 p. m.	41,2	41,1	41,4	40,6	40,5	40,6	40,0	40,0	40,0	0,0000
21	5:07:31 p. m.	41,2	41,0	41,3	40,5	40,5	40,5	40,0	40,0	40,0	0,0000
22	12:51:25 p. m.	46,3	46,1	46,2	45,6	45,5	45,6	45,0	45,0	45,0	0,0000
23	1:02:10 p. m.	49,3	49,1	49,3	48,6	48,5	48,6	48,0	48,0	48,0	0,0000
24	5:20:20 p. m.	51,3	51,1	51,3	50,6	50,5	50,5	50,0	50,0	50,0	0,0000
25	1:15:39 p. m.	51,3	51,1	51,5	50,6	50,5	50,6	50,1	50,1	50,1	0,0000
26	1:19:14 p. m.	56,3	56,1	56,3	55,6	55,5	55,6	55,0	55,1	55,1	0,0707
27	5:32:31 p. m.	61,3	61,1	61,3	60,6	60,5	60,6	60,0	60,0	60,0	0,0000
28	1:23:44 p. m.	61,3	61,1	61,3	60,7	60,6	60,6	60,1	60,1	60,1	0,0000
29	1:29:12 p. m.	66,4	66,2	66,3	65,6	65,6	65,7	65,1	65,2	65,2	0,0707
30	1:34:38 p. m.	71,4	71,2	71,2	70,7	70,6	70,6	70,2	70,1	70,2	0,0707

Fuente: Buendia Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online]. Available <http://repositorio.uan.edu.co/handle/123456789/2536>

Con estos resultados que se obtuvieron en esta investigación donde se tomaron valores de temperatura de 6 sensores diferentes, se logró evidenciar que tenía valores diferentes respecto a las muestras tomadas, sin embargo, las lecturas de los 2 sensores patrones fueron diferentes y fue necesario calcular el valor medio de la temperatura, este valor fue fundamental en el cálculo del error sistemático para cada temperatura tomada por los 6 sensores como se muestra a continuación:

**Tabla 2** – Error sistemático no ajustado.

Error sistemático (No Ajustado)					
T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>
S1	S2	S3	S4	S5	S6
E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>
°C	°C	°C	°C	°C	°C
0,9	0,8	1,0	0,4	0,3	0,4
0,9	0,8	1,0	0,3	0,2	0,3
1,0	0,8	1,0	0,4	0,3	0,4
1,1	0,9	1,2	0,5	0,4	0,5
0,9	0,8	1,0	0,3	0,2	0,3
1,1	0,9	1,2	0,5	0,4	0,5
1,1	0,9	1,2	0,4	0,3	0,3
1,1	0,9	1,2	0,5	0,4	0,5
0,9	0,8	1,0	0,3	0,2	0,3
1,1	0,9	1,2	0,4	0,3	0,4
1,2	1,1	1,3	0,6	0,4	0,6
1,1	0,9	1,2	0,5	0,4	0,4
1,2	1,0	1,2	0,5	0,4	0,5
1,1	0,9	1,2	0,5	0,4	0,4
1,2	1,0	1,2	0,5	0,4	0,5
1,0	0,9	1,2	0,4	0,3	0,4
1,3	1,1	1,3	0,6	0,5	0,6
1,2	1,1	1,3	0,6	0,5	0,6
1,2	1,1	1,3	0,6	0,5	0,6
1,2	1,1	1,4	0,6	0,5	0,6
1,2	1,0	1,3	0,5	0,5	0,5
1,3	1,1	1,2	0,6	0,5	0,6
1,3	1,1	1,3	0,6	0,5	0,6
1,3	1,1	1,3	0,6	0,5	0,5
1,2	1,0	1,4	0,5	0,4	0,5
1,3	1,1	1,3	0,6	0,5	0,6
1,3	1,1	1,3	0,6	0,5	0,6
1,2	1,0	1,2	0,6	0,5	0,5
1,3	1,1	1,1	0,4	0,4	0,5
1,3	1,1	1,1	0,5	0,4	0,4

Fuente: Buendia Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online]. Available <http://repositorio.uan.edu.co/handle/123456789/2536>

También se evidenció que se obtuvieron los resultados de la temperatura ajustada según los cálculos obtenidos y el proceso descrito paso a paso en el proyecto de investigación en el que se basó y respaldó esta tesis [15]. Además, se encontraron los resultados de la temperatura ya ajustada, lo que revela el uso de polinomios de diferentes grados para obtener estos resultados.



con la lectura proporcionada por la balanza electrónica y su desviación, estos datos contribuyeron a la creación de las tablas citadas, como se indicó en la investigación [15].

**Tabla 4** – Datos de calibración de la balanza electrónica.

DATOS DE LA CALIBRACIÓN				
CARGA ASCENDENTE CON REGRESO A CERO				
Puntos	Masa patrón kg	Indicación de la balanza kg	Temperatura ambiente °C	Presión atmosférica mbar/abs
1	0	0,001	27,7	1021,7
2	3	2,997	27,7	1021,7
3	0	0,001	27,7	1021,7
4	6	5,993	27,7	1021,7
5	0	0,001	27,7	1021,7
6	9	8,998	27,7	1021,7
7	0	0,001	27,7	1021,7
8	12	11,977	27,7	1021,7
9	0	0,001	27,7	1021,7
10	15	14,964	27,7	1021,7
11	0	0,001	27,7	1021,7
12	18	17,996	27,7	1021,7
13	0	0,001	27,7	1021,7
14	21	20,992	27,7	1021,7
15	0	0,001	27,7	1021,7
16	24	23,994	27,7	1021,7
17	0	0,001	27,7	1021,7
18	27	26,989	27,6	1021,7
19	0	0,001	27,6	1021,7
20	30	29,987	27,5	1021,7
21	0	0,001	27,5	1021,7

CARGA DESCENDENTE CON REGRESO A CERO				
Puntos	Masa patrón kg	Indicación de la balanza kg	Temperatura ambiente °C	Presión atmosférica mbar/abs
1	0	0,001	27,5	1021,7
2	3	2,998	27,5	1021,7
3	0	0,001	27,5	1021,7
4	6	5,994	27,5	1021,7
5	0	0,001	27,5	1021,7
6	9	8,998	27,5	1021,7
7	0	0,002	27,5	1021,7
8	12	11,979	27,5	1021,7
9	0	0,002	27,5	1021,7
10	15	14,964	27,5	1021,7
11	0	0,002	27,5	1021,7
12	18	17,995	27,5	1021,7
13	0	0,001	27,5	1021,7
14	21	20,994	27,5	1021,7
15	0	0,001	27,5	1021,7
16	24	23,994	27,5	1021,7
17	0	0,001	27,5	1021,7
18	27	26,988	27,5	1021,7
19	0	0,001	27,5	1021,7
20	30	29,987	27,4	1021,7
21	0	0,001	27,4	1021,7

Fuente: Buendia Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online]. Available <http://repositorio.uan.edu.co/handle/123456789/2536>

Los investigadores de la Universidad Antonio Nariño llevaron a cabo una investigación exhaustiva en la que utilizaron cálculos y polinomios para corroborar un fenómeno de gran interés. Su objetivo principal consistía en detectar y comprender los errores en las mediciones de peso de objetos y cómo estos errores estaban relacionados con la incertidumbre, que a su vez estaba influenciada por la fuerza gravitatoria del lugar donde se realizaban las mediciones. Para alcanzar este objetivo, realizaron una comparación entre la gravedad experimentada a nivel del mar, que se caracteriza por ser más alta, y la gravedad experimentada en la ciudad de Medellín [15].

El procedimiento experimental involucró mediciones precisas de la carga ascendente y descendente de los objetos examinados, así como un detallado registro manual de los valores de peso. A través de este proceso riguroso, los investigadores recopilaron una variedad de resultados diversos. Estos resultados sirvieron como base para la creación de una plantilla que presentaba los valores de peso corregidos en función de los cálculos

realizados utilizando polinomios de diferentes grados. Mediante esta técnica, los investigadores pudieron determinar cuál polinomio se ajustaba mejor a cada uno de los parámetros analizados [15].

El estudio reveló hallazgos significativos que contribuyeron a una mejor comprensión de la relación entre la gravedad y el peso de los objetos. Además, se identificaron polinomios específicos que funcionaron de manera eficaz en diversas situaciones. En última instancia, esta investigación representó un avance importante en la mejora de las mediciones de peso, lo que aumenta la precisión en varios campos de la metrología y contribuye al desarrollo tecnológico que depende de estas mediciones fundamentales para la creación de nuevas formas de investigación [15].

**Tabla 5** – Resultados finales de la calibración de la balanza con su incertidumbre.

	Puntos	RESULTADO FINAL							Error
		Masa Indicada	Masa ajustada	Incertidumbre de ajuste	Incertidumbre del patrón	Incertidumbre del instrumento	Incertidumbre patrón (u)	Incertidumbre Expandida (U)	
		kg	kg	kg	kg	kg	kg	kg	
CARGA ASCENDENTE	1	0,001	0,001	0,00020	0,00000	0,00058	0,0006	0,0012	0,0000
	2	2,997	3,013	0,00020	0,00000186	0,00058	0,0006	0,0012	-0,0156
	3	5,993	6,024	0,00020	0,0000042	0,00058	0,0006	0,0012	-0,0312
	4	8,998	9,045	0,00020	0,0000048	0,00058	0,0006	0,0012	-0,0468
	5	11,977	12,059	0,00020	0,0000087	0,00058	0,0006	0,0012	-0,0623
	6	14,964	15,078	0,00020	0,0000095	0,00058	0,0006	0,0012	-0,1140
	7	17,996	18,094	0,00020	0,0000096	0,00058	0,0006	0,0012	-0,0976
	8	20,992	21,109	0,00020	0,0000165	0,00058	0,0006	0,0012	-0,1172
	9	23,994	24,125	0,00020	0,0000167	0,00058	0,0006	0,0012	-0,1308
	10	26,989	27,140	0,00020	0,0000171	0,00058	0,0006	0,0012	-0,1514
	11	29,987	30,156	0,00020	0,0000186	0,00058	0,0006	0,0012	-0,1690
CARGA DESCENDENTE	12	0,000	0,000	0,00020	0,0000000	0,00058	0,0006	0,0012	-0,0001
	13	3,000	3,016	0,00020	0,0000019	0,00058	0,0006	0,0012	-0,0156
	14	6,000	6,031	0,00020	0,0000042	0,00058	0,0006	0,0012	-0,0312
	15	9,000	9,047	0,00020	0,0000048	0,00058	0,0006	0,0012	-0,0468
	16	12,000	12,062	0,00020	0,0000087	0,00058	0,0006	0,0012	-0,0624
	17	15,000	15,078	0,00020	0,0000095	0,00058	0,0006	0,0012	-0,0780
	18	18,000	18,094	0,00020	0,0000096	0,00058	0,0006	0,0012	-0,0936
	19	21,000	21,109	0,00020	0,0000165	0,00058	0,0006	0,0012	-0,1092
	20	24,000	24,125	0,00020	0,0000167	0,00058	0,0006	0,0012	-0,1248
	21	27,000	27,140	0,00020	0,0000171	0,00058	0,0006	0,0012	-0,1404
	22	30,000	30,156	0,00020	0,000019	0,00058	0,0006	0,0012	-0,1560

Fuente: Buendia Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online]. Available <http://repositorio.uan.edu.co/handle/123456789/2536>



**Objetivo específico #2:** *Desarrollar un software que incorpore una representación visual de la información y su disposición para su presentación al usuario.*

**Etapa 2:** *La programación se crea en C# y luego se llevan a cabo las operaciones de cálculo con datos parametrizables. Esto implica identificar los parámetros ingresados y parámetros constantes para iniciar la ejecución del programa.*

Basándose en los hallazgos, surgió la idea de crear un software que utilizara los datos de esta investigación. El objetivo era diseñar un software que detallara paso a paso el proceso de obtención de cada uno de los parámetros. Además, se reconoció la necesidad de crear un código completamente nuevo, junto con sus variables, tomando como referencia un proyecto anterior y adaptándolo a un lenguaje programable en C#. La principal meta era garantizar que este software tuviera un entorno Modelo-Vista-Controlador (MVC) fácilmente accesible y comprensible para el usuario, lo que permitiría una interacción sencilla con resultados explícitos y claros.

La concepción del software se basó en la idea de desglosar y presentar de manera didáctica todo el proceso a través del cual se obtuvieron los parámetros cruciales. Como resultado, se creó una interfaz interactiva que desglosó cada etapa del análisis para facilitar la comprensión del usuario. Esto incluyó la traducción de los resultados anteriores al lenguaje C# y la creación completa del código y la estructura del programa desde cero.

Al implementar el entorno Modelo-Vista-Controlador (MVC), se aseguró que el software se comunicara de manera amigable y eficiente. El diseño se centró en permitir que el usuario comprendiera fácilmente los resultados y su contexto al explorar los datos. Este enfoque ayudó a crear una plataforma que, además de ser útil, promueve la comprensión y el aprendizaje continuo en el campo de la investigación científica y tecnológica.

**Objetivo específico #3:** *Desarrollar un software utilizando C# con el propósito de lograr una fácil instalación y portabilidad, permitiendo su implementación en diversos equipos y su uso en cualquier entorno.*

**Etapa 3:** *En esta fase del desarrollo del software, se verifica la compatibilidad con el sistema operativo de Windows, se garantiza la facilidad en el cargue de información, se busca una ejecución rápida y se verifica la ausencia de errores.*

En esta fase del desarrollo del software, se ha priorizado la compatibilidad con las diversas versiones de Windows 10 como uno de los principales objetivos. Para lograr esto, es necesario llevar a cabo una actualización del .NET Framework, un componente esencial que garantiza un rendimiento óptimo y una perfecta integración con el sistema operativo.

Una vez que se hayan cumplido estos requisitos técnicos fundamentales, el software estará completamente alineado con la interfaz de FuelCostUAN. Esto significa que los usuarios podrán experimentar una experiencia fluida y eficiente al utilizar el programa en su sistema Windows 10, sin preocuparse por posibles incompatibilidades o problemas de funcionamiento [15].

Sin embargo, la compatibilidad no es el único aspecto en el que se están enfocando. Para asegurarse de que el software cumple con los más altos estándares de calidad, se están llevando a cabo rigurosas pruebas. Estas pruebas se centran en la correcta carga de plantillas, buscando evitar cualquier error en este proceso crítico. El objetivo es garantizar que los usuarios puedan cargar sus datos sin contratiempos y obtener resultados precisos [15].

Además, la velocidad y eficiencia son factores clave en la evaluación de este software. Se están realizando comparaciones exhaustivas entre los resultados obtenidos por el software y los que se pueden lograr utilizando una plantilla de Excel. Este análisis permitirá medir con precisión el margen de precisión del software en comparación con Excel y ayudará a identificar oportunidades para mejorar aún más el rendimiento del software. El objetivo es garantizar que los cálculos sean lo más fluidos y rápidos posibles [15].

En resumen, esta etapa del proceso de desarrollo del software se enfoca en garantizar la compatibilidad con Windows 10, realizar pruebas exhaustivas de carga de plantillas y comparar los resultados con Excel para lograr la máxima precisión y eficiencia. El objetivo final es ofrecer a los usuarios una herramienta confiable y poderosa que simplifique sus tareas y les brinde resultados precisos de manera eficiente [15].

## **6. Resultados y discusión**

Mediante el desarrollo del software basado en el trabajo de investigación titulado "Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN" [15], se logró una mejora significativa en la precisión y velocidad de los cálculos realizados en la tabla de Excel. Esto se logró mediante la estandarización de los valores de temperatura utilizando el software, lo que permitió la calibración uniforme de todos los sensores. Como resultado, los valores obtenidos se volvieron comparables, lo que contribuyó a mediciones más coherentes y consistentes.

Una característica destacada del software fue su capacidad para visualizar de manera clara y detallada los valores corregidos y los desfases que surgieron en las mediciones. Esta función permitió una identificación visual precisa de la magnitud de las correcciones realizadas.

La información se introduce en el software a través de dos opciones. Una de ellas es la "Plantilla de Carga", que es una hoja de Excel donde se ingresan los datos necesarios para cargar al software. Esta plantilla permite la introducción manual de datos para obtener información específica relacionada con la investigación mencionada [15]. Contiene las entradas y salidas de seis sensores de temperatura. La otra opción implica ingresar los datos directamente al software de manera manual.

Una de las ventajas de utilizar la plantilla es la capacidad de validar los resultados obtenidos a partir de la investigación realizada por la Universidad Antonio Nariño y compararlos con los resultados almacenados en el software, lo que proporciona una forma efectiva de verificar la precisión de las mediciones [15].

**Tabla 6 – Plantilla de Carga.**

Hora	T1	T2	T3	T4	T5	T6	Medida 1	Medida 2
3:56:09 p. m.	7,1	7	7,2	6,6	6,5	6,6	6,1	6,3
4:56:09 p. m.	9,4	9,3	9,5	8,8	8,7	8,8	8,5	8,5
5:56:09 p. m.	11,8	11,6	11,8	11,2	11,1	11,2	10,7	10,9
6:56:09 p. m.	14,3	14,1	14,4	13,7	13,6	13,7	13,1	13,3
7:56:09 p. m.	14,7	14,6	14,8	14,1	14	14,1	13,8	13,8
8:56:09 p. m.	16	15,8	16,1	15,4	15,3	15,4	14,9	14,9
9:56:09 p. m.	16,4	16,3	16,5	15,8	15,7	15,7	15,3	15,4
10:56:09 p. m.	16,5	16,3	16,6	15,9	15,8	15,9	15,4	15,5
11:56:09 p. m.	19,4	19,3	19,5	18,8	18,7	18,8	18,5	18,5
12:56:09 a. m.	21,6	21,5	21,7	21	20,9	21	20,5	20,6
1:56:09 a. m.	23,4	23,3	23,5	22,8	22,7	22,8	22,2	22,3
2:56:09 a. m.	24,2	24	24,3	23,6	23,5	23,5	23,1	23,2
3:56:09 a. m.	26,8	26,6	26,8	26,1	26	26,1	25,6	25,7
4:56:09 a. m.	26,8	26,6	26,9	26,2	26,1	26,1	25,7	25,7
5:56:09 a. m.	31,3	31,1	31,3	30,6	30,5	30,6	30,1	30,1
6:56:09 a. m.	31,1	31	31,3	30,5	30,4	30,5	30,1	30,1
7:56:09 a. m.	36,3	36,1	36,3	35,6	35,5	35,6	35	35
8:56:09 a. m.	39,2	39,1	39,3	38,6	38,5	38,6	38	38
9:56:09 a. m.	41,2	41,1	41,3	40,6	40,5	40,6	40	40
10:56:09 a. m.	41,2	41,1	41,4	40,6	40,5	40,6	40	40
11:56:09 a. m.	41,2	41	41,3	40,5	40,5	40,5	40	40
12:56:09 p. m.	46,3	46,1	46,2	45,6	45,5	45,6	45	45

Fuente: propia

## 6.1 Programación de calibración los sensores de temperatura en C#.

Los resultados del estudio reflejaron una notable concordancia con los cálculos previamente efectuados, específicamente en relación con la medición de la carga ascendente y descendente de la balanza. Este nivel de coincidencia demuestra la eficacia operativa y la solidez estructural del software, respaldados por una gestión de recursos altamente eficiente. Estos factores desempeñaron un papel de suma importancia en la obtención de resultados precisos y confiables [15].

Es relevante destacar que los valores mencionados anteriormente se obtuvieron a partir de la tabla de datos correspondiente al proyecto mencionado previamente. Esta tabla proporciona información detallada sobre la cantidad de datos recopilados, la hora de su obtención, los seis conjuntos de datos de temperatura obtenidos de los sensores y, por último, la media de las temperaturas del patrón de medición, además de la desviación de la temperatura [15].

La capacidad de correlacionar los resultados del software con los valores establecidos en la tabla del proyecto representa un indicador de la precisión y confiabilidad del software desarrollado. Esta integración de datos y el análisis meticuloso de los mismos contribuyen significativamente a la mejora de la calidad de las mediciones y la efectividad en la interpretación de los resultados obtenidos en futuras investigaciones [15].

**Tabla 7–** Datos iniciales de Sensores de Temperatura.

No Puntos	Hora	S1	S2	S3	S4	S5	S6	Temperatura del Patrón de Medición			
		Temperaturas indicadas por los sensores						Medida 1	Medida 2	Valor medio	Desviación
		T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	°C	°C	°C	°C
	h:min:s	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C
1	3:56:09 p. m.	7,1	7,0	7,2	6,6	6,5	6,6	6,1	6,3	6,2	0,1414
2	4:00:00 p. m.	9,4	9,3	9,5	8,8	8,7	8,8	8,5	8,5	8,5	0,0000
3	4:09:40 p. m.	11,8	11,6	11,8	11,2	11,1	11,2	10,7	10,9	10,8	0,1414
4	11:07:04 a. m.	14,3	14,1	14,4	13,7	13,6	13,7	13,1	13,3	13,2	0,1414
5	4:16:50 p. m.	14,7	14,6	14,8	14,1	14,0	14,1	13,8	13,8	13,8	0,0000
6	11:13:28 a. m.	16,0	15,8	16,1	15,4	15,3	15,4	14,9	14,9	14,9	0,0000
7	11:18:05 a. m.	16,4	16,3	16,5	15,8	15,7	15,7	15,3	15,4	15,4	0,0707
8	4:23:58 p. m.	16,5	16,3	16,6	15,9	15,8	15,9	15,4	15,5	15,5	0,0707
9	4:30:36 p. m.	19,4	19,3	19,5	18,8	18,7	18,8	18,5	18,5	18,5	0,0000
10	4:39:35 p. m.	21,6	21,5	21,7	21,0	20,9	21,0	20,5	20,6	20,6	0,0707
11	11:26:20 a. m.	23,4	23,3	23,5	22,8	22,7	22,8	22,2	22,3	22,3	0,0707
12	4:43:51 p. m.	24,2	24,0	24,3	23,6	23,5	23,5	23,1	23,2	23,2	0,0707
13	4:50:09 p. m.	26,8	26,6	26,8	26,1	26,0	26,1	25,6	25,7	25,7	0,0707
14	11:46:10 a. m.	26,8	26,6	26,9	26,2	26,1	26,1	25,7	25,7	25,7	0,0000
15	11:56:35 a. m.	31,3	31,1	31,3	30,6	30,5	30,6	30,1	30,1	30,1	0,0000
16	4:54:47 p. m.	31,1	31,0	31,3	30,5	30,4	30,5	30,1	30,1	30,1	0,0000
17	12:09:01 p. m.	36,3	36,1	36,3	35,6	35,5	35,6	35,0	35,0	35,0	0,0000
18	12:17:00 p. m.	39,2	39,1	39,3	38,6	38,5	38,6	38,0	38,0	38,0	0,0000
19	12:26:14 p. m.	41,2	41,1	41,3	40,6	40,5	40,6	40,0	40,0	40,0	0,0000
20	12:31:09 p. m.	41,2	41,1	41,4	40,6	40,5	40,6	40,0	40,0	40,0	0,0000
21	5:07:31 p. m.	41,2	41,0	41,3	40,5	40,5	40,5	40,0	40,0	40,0	0,0000
22	12:51:25 p. m.	46,3	46,1	46,2	45,6	45,5	45,6	45,0	45,0	45,0	0,0000
23	1:02:10 p. m.	49,3	49,1	49,3	48,6	48,5	48,6	48,0	48,0	48,0	0,0000
24	5:20:20 p. m.	51,3	51,1	51,3	50,6	50,5	50,5	50,0	50,0	50,0	0,0000
25	1:15:39 p. m.	51,3	51,1	51,5	50,6	50,5	50,6	50,1	50,1	50,1	0,0000
26	1:19:14 p. m.	56,3	56,1	56,3	55,6	55,5	55,6	55,0	55,1	55,1	0,0707
27	5:32:31 p. m.	61,3	61,1	61,3	60,6	60,5	60,6	60,0	60,0	60,0	0,0000
28	1:23:44 p. m.	61,3	61,1	61,3	60,7	60,6	60,6	60,1	60,1	60,1	0,0000
29	1:29:12 p. m.	66,4	66,2	66,3	65,6	65,6	65,7	65,1	65,2	65,2	0,0707
30	1:34:38 p. m.	71,4	71,2	71,2	70,7	70,6	70,6	70,2	70,1	70,2	0,0707

Fuente: Buendia Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online]. Available <http://repositorio.uan.edu.co/handle/123456789/2536>

Considerando la información proporcionada, se procedió a ingresar estos datos en el software a través de la plantilla de carga. Este proceso implica la introducción manual de los datos, y como resultado, la interfaz del programa reflejó lo siguiente. Al analizar la interfaz, se pudo observar que los resultados obtenidos de los sensores de temperatura

coinciden con los de la plantilla base, lo que indica que el software está realizando una compilación precisa de los cálculos.

Esta integración exitosa de datos demuestra la eficacia y la coherencia del software en la interpretación de la información ingresada manualmente. Además, refuerza la confiabilidad de los cálculos realizados por el programa al garantizar que los resultados obtenidos sean consistentes con los valores de referencia.

**Tabla 8** – Resultado de la temperatura ajustada para cada sensor.

Puntos	Hora	S1-T1 C°	S2-T2 C°	S3-T3 C°	S4-T4 C°	S5-T5 C°	S6-T6 C°	Medida 1 C°	Medida 2 C°	Valor Medio C°	Desviacion C°
1	3:56:09 p. m.	7,1	7	7,2	6,6	6,5	6,6	6,1	6,3	6,2	0,1414
2	4:56:09 p. m.	9,4	9,3	9,5	8,8	8,7	8,8	8,5	8,5	8,5	0
3	5:56:09 p. m.	11,8	11,6	11,8	11,2	11,1	11,2	10,7	10,9	10,8	0,1414
4	6:56:09 p. m.	14,3	14,1	14,4	13,7	13,6	13,7	13,1	13,3	13,2	0,1414
5	7:56:09 p. m.	14,7	14,6	14,8	14,1	14	14,1	13,8	13,8	13,8	0
6	8:56:09 p. m.	16	15,8	16,1	15,4	15,3	15,4	14,9	14,9	14,9	0
7	9:56:09 p. m.	16,4	16,3	16,5	15,8	15,7	15,7	15,3	15,4	15,4	0,0707
8	10:56:09 p. m.	16,5	16,3	16,6	15,9	15,8	15,9	15,4	15,5	15,4	0,0707
9	11:56:09 p. m.	19,4	19,3	19,5	18,8	18,7	18,8	18,5	18,5	18,5	0
10	12:56:09 a. m.	21,6	21,5	21,7	21	20,9	21	20,5	20,6	20,6	0,0707
11	1:56:09 a. m.	23,4	23,3	23,5	22,8	22,7	22,8	22,2	22,3	22,2	0,0707
12	2:56:09 a. m.	24,2	24	24,3	23,6	23,5	23,5	23,1	23,2	23,2	0,0707
13	3:56:09 a. m.	26,8	26,6	26,8	26,1	26	26,1	25,6	25,7	25,6	0,0707
14	4:56:09 a. m.	26,8	26,6	26,9	26,2	26,1	26,1	25,7	25,7	25,7	0
15	5:56:09 a. m.	31,3	31,1	31,3	30,6	30,5	30,6	30,1	30,1	30,1	0
16	6:56:09 a. m.	31,1	31	31,3	30,5	30,4	30,5	30,1	30,1	30,1	0
17	7:56:09 a. m.	36,3	36,1	36,3	35,6	35,5	35,6	35	35	35	0
18	8:56:09 a. m.	39,2	39,1	39,3	38,6	38,5	38,6	38	38	38	0
19	9:56:09 a. m.	41,2	41,1	41,3	40,6	40,5	40,6	40	40	40	0
20	10:56:09 a. m.	41,2	41,1	41,4	40,6	40,5	40,6	40	40	40	0
21	11:56:09 a. m.	41,2	41	41,3	40,5	40,5	40,5	40	40	40	0
22	12:56:09 p. m.	46,3	46,1	46,2	45,6	45,5	45,6	45	45	45	0
23	1:56:09 p. m.	49,3	49,1	49,3	48,6	48,5	48,6	48	48	48	0
24	2:56:09 p. m.	51,3	51,1	51,3	50,6	50,5	50,5	50	50	50	0
25	3:56:09 p. m.	51,3	51,1	51,5	50,6	50,5	50,6	50,1	50,1	50,1	0
26	4:56:09 p. m.	56,3	56,1	56,3	55,6	55,5	55,6	55	55,1	55	0,0707
27	5:56:09 p. m.	61,3	61,1	61,3	60,6	60,5	60,6	60	60	60	0
28	6:56:09 p. m.	61,3	61,1	61,3	60,7	60,6	60,6	60,1	60,1	60,1	0
29	7:56:09 p. m.	66,4	66,2	66,3	65,6	65,6	65,7	65,1	65,2	65,2	0,0707
30	8:56:09 p. m.	71,4	71,2	71,2	70,7	70,6	70,6	70,2	70,1	70,2	0,0707

Fuente: Autor

La vista proporcionada muestra la versión del software en relación a la introducción de datos en el programa mediante el uso de la plantilla de carga. Se puede notar que los resultados son consistentes, con la distinción de que el software ajusta las cifras decimales en los casos en que el resultado es cero. Sin embargo, en los demás resultados, se mantienen las mismas cifras decimales que los valores de entrada. Esto indica que el software está recibiendo la información y la está procesando con precisión, presentando resultados como la temperatura patrón y la temperatura de los seis sensores, los cuales contribuyen al mantenimiento óptimo de la temperatura del motor, evitando daños en su funcionamiento.

Además, se observa la desviación de temperatura entre la primera y la segunda medición del patrón, y se realizó una comparación que reveló que las medidas se mantienen consistentes, lo que indica un funcionamiento adecuado del sistema. Esta capacidad del software para mostrar resultados precisos y mantener la coherencia en las mediciones es esencial para garantizar un funcionamiento eficiente y confiable del motor en diversas aplicaciones.

### **6.1.1 Error sistemático no ajustado de la temperatura**

En cuanto al error sistemático relacionado con la temperatura, se llevó a cabo un análisis exhaustivo de los datos que reveló un proceso específico. Se determinó que el valor obtenido de los seis sensores de temperatura se resta al valor medio de la temperatura del patrón. Esta operación proporciona una diferencia entre la temperatura medida por cada sensor conectado al motor y el valor medio de temperatura registrado por dos sensores patrón.

Este error sistemático representa la cantidad de discrepancia existente entre las dos mediciones, como se detalla en la tabla 9. La identificación y cuantificación de este error sistemático son aspectos críticos en la evaluación de la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna. Estos resultados contribuyen a una comprensión más profunda de las mediciones y ayudan a mejorar la precisión y la confiabilidad en el funcionamiento del motor.

**Tabla 9 – Error sistemático.**

Error sistemático (No Ajustado)					
T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>
S1	S2	S3	S4	S5	S6
E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>
°C	°C	°C	°C	°C	°C
0,9	0,8	1,0	0,4	0,3	0,4
0,9	0,8	1,0	0,3	0,2	0,3
1,0	0,8	1,0	0,4	0,3	0,4
1,1	0,9	1,2	0,5	0,4	0,5
0,9	0,8	1,0	0,3	0,2	0,3
1,1	0,9	1,2	0,5	0,4	0,5
1,1	0,9	1,2	0,4	0,3	0,3
1,1	0,9	1,2	0,5	0,4	0,5
0,9	0,8	1,0	0,3	0,2	0,3
1,1	0,9	1,2	0,4	0,3	0,4
1,2	1,1	1,3	0,6	0,4	0,6
1,1	0,9	1,2	0,5	0,4	0,4
1,2	1,0	1,2	0,5	0,4	0,5
1,1	0,9	1,2	0,5	0,4	0,4
1,2	1,0	1,2	0,5	0,4	0,5
1,0	0,9	1,2	0,4	0,3	0,4
1,3	1,1	1,3	0,6	0,5	0,6
1,2	1,1	1,3	0,6	0,5	0,6
1,2	1,1	1,3	0,6	0,5	0,6
1,2	1,1	1,4	0,6	0,5	0,6
1,2	1,0	1,3	0,5	0,5	0,5
1,3	1,1	1,2	0,6	0,5	0,6
1,3	1,1	1,3	0,6	0,5	0,6
1,3	1,1	1,3	0,6	0,5	0,5
1,2	1,0	1,4	0,5	0,4	0,5

Fuente: Buendía Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online].

Available <http://repositorio.uan.edu.co/handle/123456789/2536>

En el proceso de desarrollo del software, se consideró que los resultados detallados no deberían mostrarse de forma inmediata en la pantalla principal del programa. En cambio, se optó por diseñar el software de manera que el usuario pueda solicitar estos resultados específicos según sea necesario. Esta estrategia se implementó para evitar sobrecargar la pantalla principal y garantizar que los datos sean presentados de manera comprensible.

Un aspecto clave de este enfoque fue asegurarse de que, al solicitar los resultados, se obtuvieran los mismos valores que se encuentran en la tabla de resultados, sin ninguna diferencia significativa. Esta consistencia en los resultados es una indicación sólida de que el sistema ha sido programado de manera efectiva y está cumpliendo con las solicitudes necesarias. Esto, a su vez, proporciona una mayor confianza en la precisión y confiabilidad de los resultados generados por el software, como se puede observar en la siguiente tabla.



**Tabla 10 – Error sistemático FuelCostUAN**

Principal - FuelCostUAN

	T1-S1-E1 C°	T2-S2-E2 C°	T3-S3-E3 C°	T4-S4-E4 C°	T5-S5-E5 C°	T6-S6-E6 C°
▶	0,9	0,8	1	0,4	0,3	0,4
	0,9	0,8	1	0,3	0,2	0,3
	1	0,8	1	0,4	0,3	0,4
	1,1	0,9	1,2	0,5	0,4	0,5
	0,9	0,8	1	0,3	0,2	0,3
	1,1	0,9	1,2	0,5	0,4	0,5
	1	0,9	1,1	0,4	0,3	0,3
	1,1	0,9	1,2	0,5	0,4	0,5
	0,9	0,8	1	0,3	0,2	0,3
	1	0,9	1,1	0,4	0,3	0,4
	1,2	1,1	1,3	0,6	0,5	0,6
	1	0,8	1,1	0,4	0,3	0,3
	1,2	1	1,2	0,5	0,4	0,5
	1,1	0,9	1,2	0,5	0,4	0,4
	1,2	1	1,2	0,5	0,4	0,5
	1	0,9	1,2	0,4	0,3	0,4
	1,3	1,1	1,3	0,6	0,5	0,6
	1,2	1,1	1,3	0,6	0,5	0,6
	1,2	1,1	1,3	0,6	0,5	0,6
	1,2	1,1	1,4	0,6	0,5	0,6
	1,2	1	1,3	0,5	0,5	0,5
	1,3	1,1	1,2	0,6	0,5	0,6
	1,3	1,1	1,3	0,6	0,5	0,6
	1,3	1,1	1,3	0,6	0,5	0,5
	1,2	1	1,4	0,5	0,4	0,5
	1,3	1,1	1,3	0,6	0,5	0,6
	1,3	1,1	1,3	0,6	0,5	0,6
	1,2	1	1,2	0,6	0,5	0,5
	1,2	1	1,1	0,4	0,4	0,5
	1,2	1	1	0,5	0,4	0,4

Fuente: propia

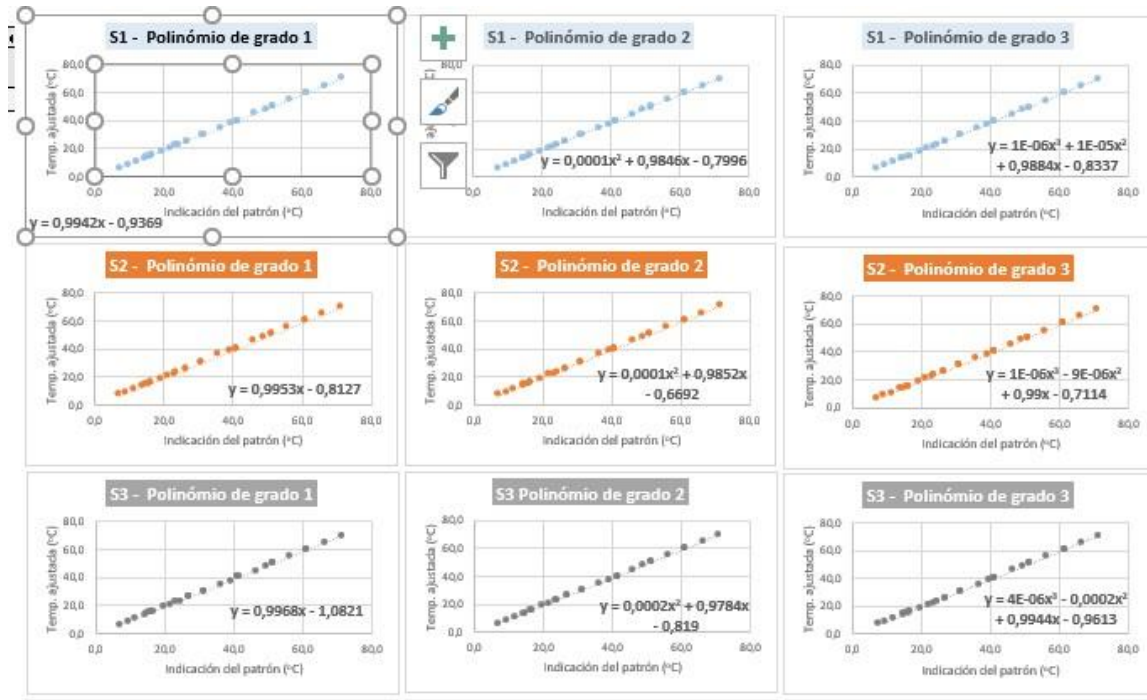
En la tabla 10, se puede apreciar que los valores obtenidos son consistentes con los resultados obtenidos en la investigación titulada "Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia" [15]. Al analizar las temperaturas registradas por los 6 sensores, se observa que no se presentan variaciones significativas en los resultados, lo que indica un funcionamiento estable y confiable del sistema.

Además, se implementó un enfoque en el diseño del software para garantizar que los resultados solo sean visibles para el usuario cuando este los solicite. Esta estrategia se adoptó con el objetivo de simplificar la comprensión del error sistemático, ya que el usuario no necesita ver las operaciones detrás de los resultados finales. De esta manera, se busca proporcionar una presentación de los datos de manera más clara y accesible para el usuario, lo que contribuye a una mejor comprensión de los resultados obtenidos.

### **6.1.2 Coeficientes de los polinomios interpoladores**

En esta etapa del software, se emplean polinomios de diversos grados con el propósito de obtener resultados precisos y determinar qué polinomio se ajusta mejor a la temperatura de acuerdo con los datos recopilados. Durante el desarrollo de esta fase, se comprendió que Excel genera el polinomio en función de la elección del usuario y crea una ecuación en los vectores  $x$  en relación con  $y$ . Esto permite analizar cómo se comporta la curva en diferentes valores de  $x$  y determinar qué polinomio produce una curva más ajustada. Esta información se basa en investigaciones previamente realizadas [15], tal como se muestra en la tabla 11.

**Tabla 11** – coeficientes de los polinomios



Fuente: Buendia Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online]. Available <http://repositorio.uan.edu.co/handle/123456789/2536>

con los resultados anteriores se creó la siguiente tabla que nos da el coeficiente del polinomio en sus diferentes grados.

**Tabla 12** – Comparación de resultados entre polinomios de primer, segundo y tercer grado con las temperaturas registradas por los sensores.

Grau	Coef.	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>
		S1	S2	S3	S4	S5	S6
1	a	0,994	0,9953	0,9968	0,9966	0,9962	0,9968
	b	-0,937	-0,8127	-1,0821	-0,3816	-0,2753	-0,37
2	a	1E-04	0,0001	0,0002	0,0001	0,0001	0,0001
	b	0,985	0,9852	0,9784	0,9884	0,9884	0,9882
	c	-0,8	-0,6692	-0,819	-0,2675	-0,1681	-0,251
3	a	1E-06	1E-06	4E-06	2E-06	2E-06	4E-06
	b	1E-05	-3E-06	-0,0002	-0,0001	-0,0001	-3E-04
	c	0,988	0,99	0,9944	0,9957	0,9952	1,002
	d	-0,834	-0,0711	-0,9613	-0,3304	-0,2257	-0,37

Fuente: Buendia Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online]. Available <http://repositorio.uan.edu.co/handle/123456789/2536>

Durante el proceso de programación, se encontró un desafío al trabajar con polinomios y sus diferentes grados, ya que comprender el funcionamiento algebraico desde el código resultó ser un obstáculo. Sin embargo, se identificó una librería que se encarga de realizar estas operaciones y generar los resultados necesarios. La ecuación correspondiente se expresa de la siguiente manera: "Polynomial.Fit(x.ToArray(), y.ToArray(), grado)", y es responsable de resolver los polinomios, proporcionando resultados como se puede apreciar en la tabla. Esta solución permitió superar las dificultades en la programación de polinomios.

**Tabla 13** – Resultados de los polinomios en el software.

Principal - FuelCostUAN  
Calibracion    Calculos

Coefficientes de los polinómios interpoladores

Polinomio Grado 1

	Coficiente	S1-T1	S2-T2	S3-T3	S4-T4	S5-T5	S6-T6
▶ A		0,994418	0,995542	0,997017	0,996837	0,996439	0,996978
B		-0,94265	-0,818447	-1,087925	-0,387309	-0,280977	-0,375435

Polinomio Grado 2

	Coficiente	S1-T1	S2-T2	S3-T3	S4-T4	S5-T5	S6-T6
▶ A		0,000154	0,000161	0,000271	0,000137	0,000131	0,000142
B		0,982881	0,983513	0,976671	0,986738	0,986803	0,986532
C		-0,778389	-0,64832	-0,797693	-0,247325	-0,147987	-0,230744

Polinomio Grado 3

	Coficiente	S1-T1	S2-T2	S3-T3	S4-T4	S5-T5	S6-T6
▶ A		2E-06	3E-06	6E-06	3E-06	3E-06	5E-06
B		-0,000122	-0,000144	-0,000371	-0,000245	-0,000234	-0,000441
C		0,99207	0,993629	0,998105	0,99922	0,998699	1,005551
D		-0,859764	-0,737257	-0,987955	-0,354424	-0,249517	-0,393856

Fuente: propia

Tras revisar los resultados obtenidos por el software, se destaca su mayor exactitud y precisión en los polinomios, lo que se traduce en resultados con mayor cantidad de decimales. Esto resulta especialmente útil para corregir la temperatura de los sensores y garantizar que todos marquen la misma temperatura sin variaciones entre ellos. Esta característica del software proporciona una visualización clara de la mejora que supone utilizar un programa que realice cálculos más precisos, permitiendo así obtener correcciones de mayor calidad y confiabilidad.

### **6.1.3 Temperatura ajustada por cada polinomio.**

En esta etapa del proceso, se llevó a cabo un análisis detallado de los resultados de temperatura obtenidos a través del software. Se ha observado que la corrección obtenida es consistente con los resultados de la investigación previamente mencionada [15]. Este análisis ha dado como resultado una presentación visual más comprensible en la interfaz del programa, donde se han delimitado y presentado claramente cada uno de los polinomios. Esto mejora significativamente la comprensión y el análisis de los resultados por parte del usuario.

Al examinar los resultados de los diferentes polinomios, se han identificado ligeras variaciones en las temperaturas respecto a las mediciones iniciales. Sin embargo, se ha observado que uno de los polinomios, específicamente el número 1, ajusta mejor los valores y presenta una mayor cantidad de similitudes en comparación con los otros polinomios. Estos últimos muestran una similitud principalmente en las temperaturas más bajas, lo que aumenta la incertidumbre. Es importante destacar que este análisis debe ser realizado por el usuario para comprender los valores obtenidos por el software, como se muestra en la tabla 14 y tabla 15.

Tabla 14 – Temperatura ajustada por cada polinomio.

Polinomio de grado 1						Polinomio de grado 2						Polinomio de grado 3					
T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>
S1	S2	S3	S4	S5	S6	S1	S2	S3	S4	S5	S6	S1	S2	S3	S4	S5	S6
°C	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C	°C
6,1	6,2	6,1	6,2	6,2	6,2	6,2	6,2	6,2	6,3	6,3	6,3	6,2	6,9	6,2	6,2	6,2	6,2
8,4	8,4	8,4	8,4	8,4	8,4	8,5	8,5	8,5	8,4	8,4	8,5	8,5	9,1	8,5	8,4	8,4	8,4
10,8	10,7	10,7	10,8	10,8	10,8	10,8	10,8	10,8	10,8	10,8	10,8	10,8	11,4	10,8	10,8	10,8	10,8
13,3	13,2	13,3	13,3	13,3	13,3	13,3	13,2	13,3	13,3	13,3	13,3	13,3	13,9	13,3	13,3	13,3	13,3
13,7	13,7	13,7	13,7	13,7	13,7	13,7	13,7	13,7	13,7	13,7	13,7	13,7	14,4	13,7	13,7	13,7	13,7
15,0	14,9	15,0	15,0	15,0	15,0	15,0	14,9	15,0	15,0	15,0	15,0	15,0	15,6	15,0	15,0	15,0	15,0
15,4	15,4	15,4	15,4	15,4	15,4	15,4	15,4	15,4	15,4	15,4	15,3	15,4	16,1	15,4	15,4	15,4	15,3
15,5	15,4	15,5	15,5	15,5	15,5	15,5	15,5	15,4	15,5	15,5	15,5	15,5	16,1	15,5	15,5	15,5	15,5
18,4	18,4	18,4	18,4	18,4	18,4	18,3	18,4	18,3	18,3	18,3	18,4	18,4	19,0	18,4	18,4	18,4	18,4
20,5	20,6	20,5	20,5	20,5	20,6	20,5	20,6	20,5	20,5	20,5	20,5	20,5	21,2	20,6	20,6	20,5	20,6
22,3	22,4	22,3	22,3	22,3	22,4	22,3	22,3	22,3	22,3	22,3	22,3	22,3	23,0	22,3	22,3	22,3	22,4
23,1	23,1	23,1	23,1	23,1	23,1	23,1	23,0	23,1	23,1	23,1	23,0	23,1	23,7	23,1	23,1	23,1	23,1
25,7	25,7	25,6	25,6	25,6	25,6	25,7	25,6	25,5	25,6	25,6	25,6	25,7	26,3	25,6	25,6	25,6	25,6
25,7	25,7	25,7	25,7	25,7	25,6	25,7	25,6	25,6	25,7	25,7	25,6	25,7	26,3	25,7	25,7	25,7	25,6
30,2	30,1	30,1	30,1	30,1	30,1	30,1	30,1	30,0	30,1	30,1	30,1	30,1	30,7	30,1	30,1	30,1	30,1
30,0	30,0	30,1	30,0	30,0	30,0	29,9	30,0	30,0	30,0	30,0	30,0	29,9	30,6	30,1	30,0	30,0	30,0
35,2	35,1	35,1	35,1	35,1	35,1	35,1	35,0	35,0	35,0	35,0	35,1	35,1	35,7	35,1	35,1	35,1	35,1
38,0	38,1	38,1	38,1	38,1	38,1	38,0	38,0	37,9	38,0	38,0	38,0	38,0	38,7	38,1	38,1	38,1	38,1
40,0	40,1	40,1	40,1	40,1	40,1	39,9	40,0	39,9	40,0	40,0	40,0	40,0	40,7	40,0	40,1	40,0	40,1
40,0	40,1	40,2	40,1	40,1	40,1	39,9	40,0	40,0	40,0	40,0	40,0	40,0	40,7	40,1	40,1	40,0	40,1
40,0	40,0	40,1	40,0	40,1	40,0	39,9	39,9	39,9	39,9	40,0	39,9	40,0	40,6	40,0	40,0	40,0	40,0
45,1	45,1	45,0	45,1	45,1	45,1	45,0	45,0	44,8	45,0	45,0	45,0	45,0	45,6	44,9	45,1	45,0	45,1
48,1	48,1	48,1	48,1	48,0	48,1	48,0	47,9	47,9	48,0	48,0	48,0	48,0	48,6	48,1	48,1	48,0	48,1
50,1	50,0	50,1	50,0	50,0	50,0	50,0	49,9	49,9	50,0	50,0	49,9	50,0	50,6	50,1	50,1	50,0	50,0
50,1	50,0	50,3	50,0	50,0	50,1	50,0	49,9	50,1	50,0	50,0	50,0	50,0	50,6	50,3	50,1	50,0	50,1
55,0	55,0	55,0	55,0	55,0	55,1	55,0	54,9	54,9	55,0	55,0	55,0	55,0	55,6	55,1	55,1	55,0	55,1
60,0	60,0	60,0	60,0	60,0	60,0	59,9	59,9	59,9	60,0	60,0	60,0	60,0	60,6	60,2	60,1	60,1	60,1
60,0	60,0	60,0	60,1	60,1	60,0	59,9	59,9	59,9	60,1	60,1	60,0	60,0	60,6	60,2	60,2	60,2	60,1
65,1	65,1	65,0	65,0	65,1	65,1	65,0	65,0	64,9	65,0	65,1	65,1	65,1	65,7	65,3	65,1	65,2	65,3
70,0	70,1	69,9	70,1	70,1	70,0	70,0	70,0	69,9	70,1	70,1	70,0	70,2	70,7	70,3	70,3	70,2	70,3

Fuente: Buendía Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online]. Available <http://repositorio.uan.edu.co/handle/123456789/2536>

Tabla 15 – Temperatura ajustada por cada polinomio en el software.

The screenshot displays a software interface with three calibration tables. The interface has a menu bar with 'Principal - FuelCostUAN', 'Calibracion', and 'Calculos'. The tables show temperature adjustments for various points (T1-S1 to T6-S6) across three polynomial degrees (1, 2, and 3). The data in the tables is as follows:

T1-S1	T2-S2	T3-S3	T4-S4	T5-S5	T6-S6
6,1	6,2	6,1	6,2	6,2	6,2
8,4	8,4	8,4	8,4	8,4	8,4
10,8	10,7	10,7	10,8	10,8	10,8
13,3	13,2	13,3	13,3	13,3	13,3
13,7	13,7	13,7	13,7	13,7	13,7
15	14,9	15	15	15	15
15,4	15,4	15,4	15,4	15,4	15,3
15,5	15,4	15,5	15,5	15,5	15,5
18,3	18,4	18,4	18,4	18,4	18,4
20,5	20,6	20,5	20,5	20,5	20,6
22,3	22,4	22,3	22,3	22,3	22,4
23,1	23,1	23,1	23,1	23,1	23,1
25,7	25,7	25,6	25,6	25,6	25,6
25,7	25,7	25,7	25,7	25,7	25,6
30,2	30,1	30,1	30,1	30,1	30,1
30	30	30,1	30	30	30
35,2	35,1	35,1	35,1	35,1	35,1
38	38,1	38,1	38,1	38,1	38,1
40	40,1	40,1	40,1	40,1	40,1
40	40,1	40,2	40,1	40,1	40,1
40	40	40,1	40	40,1	40
45,1	45,1	45	45,1	45,1	45,1
48,1	48,1	48,1	48,1	48	48,1
50,1	50,1	50,1	50,1	50	50
50,1	50,1	50,3	50,1	50	50,1
55	55	55	55	55	55,1
60	60	60	60	60	60
60	60	60	60,1	60,1	60
65,1	65,1	65	65	65,1	65,1
70,1	70,1	69,9	70,1	70,1	70

T1-S1	T2-S2	T3-S3	T4-S4	T5-S5	T6-S6
6,2	6,2	6,2	6,3	6,3	6,3
8,5	8,5	8,5	8,4	8,4	8,5
10,8	10,8	10,8	10,8	10,8	10,8
13,3	13,3	13,3	13,3	13,3	13,3
13,7	13,7	13,7	13,7	13,7	13,7
15	14,9	15	15	15	15
15,4	15,4	15,4	15,4	15,4	15,3
15,5	15,4	15,5	15,5	15,5	15,5
18,3	18,4	18,4	18,4	18,4	18,4
20,5	20,6	20,5	20,5	20,5	20,5
22,3	22,4	22,3	22,3	22,3	22,3
23,1	23	23,1	23,1	23,1	23
25,7	25,6	25,6	25,6	25,6	25,6
25,7	25,6	25,7	25,7	25,7	25,6
30,1	30,1	30	30,1	30,1	30,1
29,9	30	30	30	30	30
35,1	35,1	35	35,1	35	35,1
38	38,1	38	38	38	38,1
40	40	40	40	40	40,1
40	40	40,1	40	40	40,1
40	39,9	40	39,9	40	40
45,1	45	44,9	45	45	45,1
48,1	48	48	48	48	48,1
50	50	50	50	50	50
50	50	50,2	50	50	50,1
55	55	55	55	55	55,1
60,1	60	60,1	60,1	60	60,1
60,1	60	60,1	60,2	60,1	60,1
65,2	65,2	65,1	65,1	65,2	65,2
70,2	70,2	70,1	70,2	70,2	70,1

T1-S1	T2-S2	T3-S3	T4-S4
6,2	6,2	6,2	6,2
8,5	8,5	8,5	8,4
10,8	10,8	10,7	10,8
13,3	13,3	13,3	13,3
13,7	13,7	13,7	13,7
15	14,9	15	15
15,4	15,4	15,4	15,4
15,5	15,4	15,5	15,5
18,4	18,4	18,4	18,4
20,5	20,6	20,6	20,5
22,3	22,4	22,3	22,3
23,1	23,1	23,1	23,1
25,7	25,6	25,6	25,6
25,7	25,6	25,7	25,7
30,1	30,1	30,1	30,1
29,9	30	30,1	30
35,1	35,1	35	35
38	38,1	38	38
39,9	40,1	40	40
39,9	40,1	40,1	40
39,9	40	40	39,9
45	45,1	44,9	45
48	48,1	48	48
50	50,1	50	50
50	50,1	50,3	50
55	55,1	55,1	55
60	60,1	60,2	60
60	60,1	60,2	60,1
65,1	65,3	65,3	65
70,1	70,4	70,4	70,1

Fuente: propia

En la tabla 14 y tabla 15, se puede observar que los resultados de las temperaturas son idénticos a los resultados previamente investigados. Este hecho proporciona una sólida evidencia de que el software está realizando los cálculos de manera eficiente y precisa. Además, contribuye a que la información sea más comprensible para el usuario, mejorando significativamente la velocidad de ejecución de los procesos y estandarizando el proceso de investigación.

El uso del software garantiza que los resultados sean exactos y que las operaciones no puedan ser modificadas por el usuario. Esto se logra mediante el uso de listas en el lenguaje de programación, que permite utilizar los resultados ingresados por el usuario y realizar las operaciones de manera transparente y sin posibilidad de alteración. Esta característica brinda una mayor seguridad al investigador al utilizar la plantilla, evitando cambios no autorizados en los resultados o la formulación.

#### **6.1.4 Incertidumbre del ajuste de la temperatura posteriormente al polinomio.**

En esta etapa del desarrollo del software, se procede a realizar el cálculo de la incertidumbre de la temperatura corregida utilizando los polinomios previamente calculados para cada uno de los grados. Luego, se compara este valor corregido con el valor medio calculado utilizando las temperaturas patrón de medición. Se realiza un análisis para determinar la diferencia entre estos dos valores, y se espera que esta diferencia sea lo más pequeña posible, lo que indicaría que las temperaturas corregidas se están acercando cada vez más a las temperaturas patrón.

En los resultados obtenidos del software, se observa que algunos valores de diferencia son cero, lo que sugiere que el software está corrigiendo adecuadamente la temperatura de acuerdo con los polinomios seleccionados. La elección de qué polinomio utilizar es una decisión del usuario, y aunque algunos resultados muestran variaciones significativas, en otros casos, la diferencia es cero, lo que refuerza la confianza en la precisión y el funcionamiento del software.

Estos cálculos y la organización del software en su interfaz son claros y de fácil comprensión para el usuario, como se puede apreciar en la tabla 16.

**Tabla 16** – Incertidumbre del ajuste del polinomio.

	E1-S1 (C <sup>1</sup> ) <sup>2</sup>	E2-S2 (C <sup>2</sup> ) <sup>2</sup>	E3-S3 (C <sup>3</sup> ) <sup>2</sup>	E4-S4 (C <sup>4</sup> ) <sup>2</sup>	E5-S5 (C <sup>5</sup> ) <sup>2</sup>	E6-S6 (C <sup>6</sup> ) <sup>2</sup>
0,01	0	0,01	0	0	0	
0,01	0,01	0,01	0,01	0,01	0,01	
0	0,01	0,01	0	0	0	
0,01	0	0,01	0,01	0,01	0,01	
0,01	0,01	0,01	0,01	0,01	0,01	
0,01	0	0,01	0,01	0,01	0,01	
0	0	0	0	0	0,01	
0,01	0	0,01	0,01	0,01	0,01	
0,04	0,01	0,01	0,01	0,01	0,01	
0,01	0	0,01	0,01	0,01	0	
0,01	0,04	0,01	0,01	0,01	0,04	
0,01	0,01	0,01	0,01	0,01	0,01	
0,01	0,01	0	0	0	0	
0	0	0	0	0	0,01	
0,01	0	0	0	0	0	
0,01	0,01	0	0,01	0,01	0,01	
0,04	0,01	0,01	0,01	0,01	0,01	
0	0,01	0,01	0,01	0,01	0,01	
0	0,01	0,01	0,01	0,01	0,01	
0	0,01	0,01	0,01	0,01	0,01	
0	0,01	0,04	0,01	0,01	0,01	
0	0	0,01	0	0,01	0	
0,01	0,01	0	0,01	0,01	0,01	
0,01	0,01	0,01	0,01	0	0,01	
0,01	0,01	0,01	0,01	0	0	
0	0	0,04	0	0,01	0	
0	0	0	0	0	0,01	
0	0	0	0	0	0	
0,01	0,01	0,01	0	0	0,01	
0,01	0,01	0,04	0,04	0,01	0,01	
0,01	0,01	0,09	0,01	0,01	0,04	

Fuente: propia

Se puede observar que el modelo, la vista y el controlador están funcionando de manera eficiente en el software. Los resultados son claros, y la elección de colores y tipos de letra facilita la comprensión de los resultados obtenidos en las operaciones. La organización de las tablas y los resultados es de fácil comprensión, y la forma en que se ha estructurado la información es clara y lógica. Los encabezados de las tablas son entendibles, lo que contribuye a una presentación ordenada de la información.

### 6.1.5 Incertidumbre del ajuste unitario

En el ajuste de la incertidumbre unitaria, se realiza la suma de todos los valores de cada sensor y se divide este valor entre 7.5. Luego, se calcula la raíz cuadrada de este resultado para determinar cuál es el error entre las diferentes tomas de temperaturas y obtener la incertidumbre unitaria para cada polinomio y sensor. Este proceso permite identificar cuál de los polinomios se ajusta mejor a las diferentes temperaturas. En el Front del software,



se pueden observar datos más claros que facilitan la comprensión de estos resultados en la tabla 17.

**Tabla 17** – Incertidumbre del ajuste unitario.

	T1-S1 C°	T2-S2 C°	T3-S3 C°	T4-S4 C°	T5-S5 C°	T6-S6 C°
	0,0093	0,0075	0,0139	0,0079	0,0068	0,0096

Fuente: propia

Teniendo en cuenta que los resultados iniciales y de los polinomios no generaron fallas, se evidencia que los resultados son de calidad y con buena precisión. Se tiene una buena visualización de los resultados, la información está organizada y es comprensible. No se presentan problemas al ejecutar las operaciones, lo que brinda confiabilidad en estos resultados. Se tomó la decisión de calcular la incertidumbre de ajuste mínimo y se evidenció que para cada sensor hay un polinomio que ajusta mejor que otros, por lo que se selecciona el polinomio que ofrece una mayor precisión en la mayoría de los sensores, como se muestra en la tabla 18.

**Tabla 18** – Incertidumbre del ajuste mínimo.

	T1-S1 C° Grado 2	T2-S2 C° Grado 2	T3-S3 C° Grado 2	T4-S4 C° Grado 2	T5-S5 C° Grado 2	T6-S6 C° Grado 3
	0,0075	0,0064	0,0057	0,0061	0,0043	0,0071

Fuente: propia

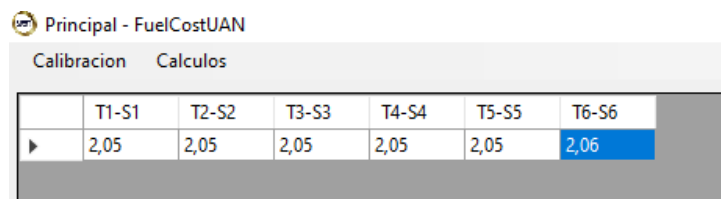
Se evidencia que los resultados pueden variar y que el polinomio de grado 2 es más preciso, lo que lo hace adecuado para la calibración de la mayoría de los sensores, aumentando su precisión. La presentación del Front es clara, los encabezados son comprensibles y los colores de fondo junto con los datos presentados están organizados

según las preferencias del usuario. No hay una saturación de información en la pantalla, y el tiempo de ejecución es muy rápido, lo que brinda confiabilidad a los resultados.

### 6.1.6 factor de cobertura de t Student ( $k$ )

En esta etapa del desarrollo, se evidenció la necesidad de obtener el inverso de la distribución t de Student de dos colas y calcular la cantidad de valores de la muestra de las temperaturas. La confiabilidad deseada para este caso es del 95%, lo que implica un error del 5%. Por lo tanto, en el sistema se programó un error del 0.5 como número entero. Se toman las muestras y se resta el grado del polinomio más uno, lo que significa que, si tenemos un polinomio de grado 3, se programa como "n - 4", y si tenemos un polinomio de grado 2, se programa como "n - 3", y así sucesivamente según el grado del polinomio. Se encontró que el código que nos ayuda a obtener este resultado es el siguiente: "return Math.Round(-StudentT.InvCDF(0d, 1d, gradosLibertad, nivelConf / 2), decimales)". Esta es una librería que devuelve el valor inverso, y el resultado obtenido se puede observar en la tabla 19.

**Tabla 19** – Factor de cobertura k



The screenshot shows a software window titled "Principal - FuelCostUAN" with two tabs: "Calibracion" and "Calculos". The "Calculos" tab is active, displaying a table with six columns labeled T1-S1, T2-S2, T3-S3, T4-S4, T5-S5, and T6-S6. The values in the table are 2,05 for T1-S1 through T5-S5, and 2,06 for T6-S6. The cell containing 2,06 is highlighted in blue.

	T1-S1	T2-S2	T3-S3	T4-S4	T5-S5	T6-S6
	2,05	2,05	2,05	2,05	2,05	2,06

Fuente: propia

El resultado obtenido se muestra como correcto, ya que utiliza los polinomios que mejor se ajustan a cada valor, lo que genera resultados de alta calidad. Esto demuestra que el lenguaje de programación empleado es claro y no propenso a errores. La interfaz es fácil de leer y no se encuentra saturada de información, lo que la hace cómoda de usar y proporciona resultados claros y precisos. Esto permite al investigador tomar decisiones informadas y acertadas.

### 6.1.7 Temperatura ajustada y su error sistemático

En los análisis de los resultados se determinó que el polinomio de grado 2 es el que mejor se ajusta a las temperaturas. Por esta razón, se ha permitido que el usuario obtenga los resultados de los tres polinomios y realice su propio análisis para identificar cuál de ellos ofrece los mejores resultados en su investigación. De esta manera, se puede determinar qué sensor tiene un margen de error más bajo y cuál se ajusta de manera óptima a la temperatura patrón. Los resultados se presentaron en una tabla que muestra la temperatura ajustada final y su error sistemático en comparación con los valores iniciales. Esto permite evidenciar cuál sensor presenta un mayor desfase, como se observa en la tabla 20.

**Tabla 20** – Temperatura ajustada vs error sistemático.

Principal - FuelCostUAN

Calibracion    Calculos

	Temp Ajustada T1-S1 C°	Temp Ajustada T2-S2 C°	Temp Ajustada T3-S3 C°	Temp Ajustada T4-S4 C°	Temp Ajustada T5-S5 C°	Temp Ajustada T6-S6 C°	Patrón Valor medio	Error T1-S1 C°	Error T2-S2 C°	Error T3-S3 C°
▶	6,2	6,2	6,2	6,3	6,3	6,3	6,2	0	0	0
	8,5	8,5	8,5	8,4	8,4	8,5	8,5	0	0	0
	10,8	10,8	10,8	10,8	10,8	10,8	10,8	0	0	0
	13,3	13,3	13,3	13,3	13,3	13,3	13,2	0,1	0,1	0,1
	13,7	13,7	13,7	13,7	13,7	13,7	13,8	-0,1	-0,1	-0,1
	15	14,9	15	15	15	15	14,9	0,1	0	0,1
	15,4	15,4	15,4	15,4	15,4	15,3	15,4	0	0	0
	15,5	15,4	15,5	15,5	15,5	15,5	15,4	0,1	0	0,1
	18,3	18,4	18,4	18,4	18,4	18,4	18,5	-0,2	-0,1	-0,1
	20,5	20,6	20,5	20,5	20,5	20,5	20,6	-0,1	0	-0,1
	22,3	22,4	22,3	22,3	22,3	22,3	22,2	0,1	0,2	0,1
	23,1	23	23,1	23,1	23,1	23	23,2	-0,1	-0,2	-0,1
	25,7	25,6	25,6	25,6	25,6	25,6	25,6	0,1	0	0
	25,7	25,6	25,7	25,7	25,7	25,6	25,7	0	-0,1	0
	30,1	30,1	30	30,1	30,1	30,1	30,1	0	0	-0,1
	29,9	30	30	30	30	30	30,1	-0,2	-0,1	-0,1
	35,1	35,1	35	35,1	35	35,1	35	0,1	0,1	0
	38	38,1	38	38	38	38,1	38	0	0,1	0
	40	40	40	40	40	40,1	40	0	0	0
	40	40	40,1	40	40	40,1	40	0	0	0,1
	40	39,9	40	39,9	40	40	40	0	-0,1	0
	45,1	45	44,9	45	45	45,1	45	0,1	0	-0,1
	48,1	48	48	48	48	48,1	48	0,1	0	0
	50	50	50	50	50	50	50	0	0	0
	50	50	50,2	50	50	50,1	50,1	-0,1	-0,1	-0,1

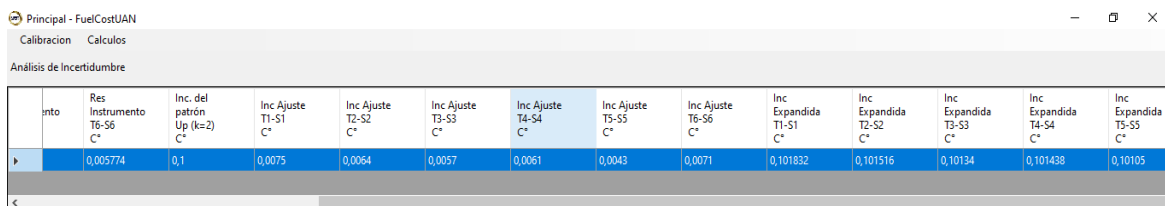
Fuente: propia

En los resultados, se observó que, para los sensores 1, 2 y 3, el error en la primera muestra es nulo, es decir, igual a 0. En cambio, para los sensores 4, 5 y 6, se detecta una corrección, ya que los resultados son aproximados a una décima. Esto demuestra que el software está funcionando de acuerdo con los objetivos planteados, lo que permite al usuario comprender rápidamente los resultados y entender cada uno de los valores presentados en la interfaz. Es importante destacar que el usuario debe comprender lo que está leyendo para que la interpretación de los resultados sea más efectiva.

### 6.1.8 Análisis de Incertidumbre

En última instancia, se procede a llevar a cabo un análisis de las temperaturas. El objetivo de este análisis es comprender cómo se comporta la incertidumbre para determinar cuánto se corrigió el valor de cada temperatura. Para resolver esta ecuación, se desarrolló un lenguaje de programación que permitió obtener resultados. Los valores obtenidos muestran que se encuentran dentro de rangos similares, lo que indica cuánto se aumentó la temperatura de cada sensor para compensar el error. A esta compensación se le conoce como incertidumbre, y los resultados se presentan en la tabla 21.

**Tabla 21** – Incertidumbre expandida.



Intero	Res Instrumento T6-S6 C*	Inc. del patrón Up (k=2) C*	Inc Ajuste T1-S1 C*	Inc Ajuste T2-S2 C*	Inc Ajuste T3-S3 C*	Inc Ajuste T4-S4 C*	Inc Ajuste T5-S5 C*	Inc Ajuste T6-S6 C*	Inc Expandida T1-S1 C*	Inc Expandida T2-S2 C*	Inc Expandida T3-S3 C*	Inc Expandida T4-S4 C*	Inc Expandida T5-S5 C*
	0,005774	0,1	0,0075	0,0064	0,0057	0,0061	0,0043	0,0071	0,101832	0,101516	0,10134	0,101438	0,10105

Fuente: propia

## 6.2 Programación de calibración de balanza en C#

En esta etapa del desarrollo, se procedió a revisar los cálculos previamente realizados por el equipo de investigación de la Universidad Antonio Nariño, quienes se enfocaron en el estudio de la incertidumbre de la balanza utilizada para pesar el combustible. Este proceso permitió determinar con precisión el consumo de combustible en gramos durante un período de tiempo y a un conjunto específico de revoluciones del motor, considerando

diferentes tipos de combustibles disponibles en el mercado, como la gasolina extra y la corriente [15].

Estos cálculos resultaron fundamentales para determinar el verdadero consumo de un motor en funcionamiento. Es importante tener en cuenta que el consumo de combustible puede variar según la carga a la que esté sometido el motor y las revoluciones a las que se desee operar. En este contexto, se consideró una balanza que se ajusta a cero y se tomó la medida de temperatura del entorno donde se llevó a cabo la calibración. Además, se consideró la presión atmosférica del lugar, que en este estudio fue muy cercana al nivel del mar.

Por otro lado, se utilizaron pesos previamente calibrados, como los de 1 kg, 2 kg (dos unidades), 5 kg y 20 kg. Estos pesos ya contaban con certificaciones. El proceso consistió en verificar si el valor de los pesos calibrados coincidía con los resultados mostrados en la balanza. Para este propósito, se cargaron los valores previamente estudiados en términos de confiabilidad de los resultados y se verificó que el software realizara los cálculos de manera precisa y eficiente. Los resultados obtenidos con el software se presentan en la tabla 22.

**Tabla 22** – carga descendente con regreso a 0.

Principal - FuelCostUAN  
Calibración    Cálculos

CARGA DESCENDENTE CON REGRESO A CERO

Puntos Medida	Masa Patrón kg	Indicación de la balanza kg	Temperatura ambiente °C	Presión atmosférica mbar/abs
1	0	0,001	27,5	1021,7
2	3	2,998	27,5	1021,7
3	0	0,001	27,5	1021,7
4	6	5,994	27,5	1021,7
5	0	0,001	27,5	1021,7
6	9	8,998	27,5	1021,7
7	0	0,002	27,5	1021,7
8	12	11,979	27,5	1021,7
9	0	0,002	27,5	1021,7
10	15	14,964	27,5	1021,7
11	0	0,002	27,5	1021,7
12	18	17,995	27,5	1021,7
13	0	0,001	27,5	1021,7
14	21	20,994	27,5	1021,7
15	0	0,001	27,5	1021,7

Añadir  
Cargar

Fuente: propia

Los resultados obtenidos en las mediciones de los gramos, realizadas a través del software, se compararon con los resultados obtenidos en una investigación previamente realizada en la Universidad Antonio Nariño [15]. Esta comparación permitió evidenciar que los resultados coinciden y no muestran variaciones significativas. Este hallazgo indica que el software es preciso y capaz de capturar y conservar las lecturas que se ingresan previamente. Esto mejora la precisión en la investigación, y demuestra que el lenguaje de programación empleado en el software ejecuta los cálculos de manera eficiente y asigna los valores correspondientes a cada celda y columna.

En este contexto, también se consideró la posibilidad de cargar valores con una carga ascendente, lo que ayuda a comprender el funcionamiento de la incertidumbre de la balanza. Esto implica realizar mediciones no solo en una carga descendente a cero sino también en una carga progresiva. Estos valores pueden variar ligeramente entre mediciones. Para este propósito, se implementó un módulo en el software que muestra estos resultados, como se detalla en la tabla 23.

**Tabla 23** – carga ascendente con regreso a 0.

Principal - FuelCostUAN  
Calibracion    Calculos

CARGA ASCENDENTE CON REGRESO A CERO

	Puntos Medida	Masa Patrón kg	Indicación de la balanza kg	Temperatura ambiente °C	Presión atmosférica mbar/abs
▶	1	0	0,001	27,7	1021,7
	2	3	2,997	27,7	1021,7
	3	0	0,001	27,7	1021,7
	4	6	5,993	27,7	1021,7
	5	0	0,001	27,7	1021,7
	6	9	8,998	27,7	1021,7
	7	0	0,001	27,7	1021,7
	8	12	11,977	27,7	1021,7
	9	0	0,001	27,7	1021,7
	10	15	14,964	27,7	1021,7
	11	0	0,001	27,7	1021,7
	12	18	17,996	27,7	1021,7
	13	0	0,001	27,7	1021,7
	14	21	20,992	27,7	1021,7
	15	0	0,001	27,7	1021,7

Añadir  
Cargar

Fuente: propia

Los resultados obtenidos son de alta calidad, y se observó que los valores son consistentes, incluso al mantener condiciones constantes de temperatura ambiente y presión atmosférica. Sin embargo, se notó que la variación en los pesos registrados en la balanza presenta un error de 0.001 gramos. Este hallazgo sugiere que la balanza no retorna al peso de 0 gramos de manera absoluta, lo que llevó al grupo de investigación de la Universidad Antonio Nariño [15] a determinar y cuantificar el error asociado a este instrumento de medición. El objetivo principal es comprender la incertidumbre relacionada con la medición de pesos en la balanza.

Los resultados obtenidos en el software son fundamentales para el cálculo de los siguientes parámetros, lo que ayuda a los investigadores a comprender la importancia de continuar con el desarrollo del software. Esto se debe a que el software estandariza el proceso de cálculo de la incertidumbre de la balanza, como se ilustra en las tablas 20 y 21.

### **6.2.1 Masa nominal de los patrones para la balanza**

En esta etapa del desarrollo, es evidente que, para determinar la incertidumbre asociada a la balanza, es necesario disponer de los valores de error de peso proporcionados en el certificado de calibración de las pesas, expresados en kilogramos. Además, se requiere la incertidumbre expandida de cada pesa, denotada como "u", y el factor de la pesa, que se representa como "k". Estos valores son específicos para cada pesa y son proporcionados en el certificado de calibración por el proveedor.

Con estos datos, se calcula la incertidumbre de cada pesa mediante la división de la incertidumbre expandida entre el factor "k". Es importante destacar que estos valores deben ser ingresados manualmente por el usuario en el software, y posteriormente son procesados por el programa, como se puede observar en la tabla 24. Los resultados obtenidos son consistentes con los cálculos realizados en la investigación previa [15], lo que indica que esta parte del software está correctamente programada, con operaciones y una interfaz de fácil uso. Es fundamental recordar que es esencial contar con todos los datos necesarios para obtener resultados precisos.

**Tabla 24** – Masa nominal de los patrones.

Principal - FuelCostUAN

Calibración    Cálculos    Datos Finales

MASA NOMINAL DE LOS PATRONES UTILIZADOS EN LA CALIBRACIÓN

No	Valor nominal	Identificación patrón	Error KG	Incertidumbre Expandida (U)	Factor K	Incertidumbre (u)
1	1	1KG	5,9E-07	1,7E-06	2	8,5E-07
2	2	2KG	3,06E-06	3,3E-06	2	1,65E-06
3	5	5KG	3,3E-06	8,3E-06	2	4,15E-06
4	10	10KG	2,2E-05	1,7E-05	2	8,5E-06
5	20	20KG	6,4E-05	3,3E-05	2	1,65E-05

Guardar

Fuente: propia

En la tabla, se pueden apreciar resultados satisfactorios que carecen de distorsiones y son de fácil comprensión. La interfaz de ejecución es clara y no se encuentra sobrecargada de información. Los colores utilizados son claros, lo que facilita su comprensión. Los datos no generan errores y, lo más relevante, los resultados obtenidos coinciden con los investigados previamente.

### 6.2.2 Ajuste aparente vs indicada carga ascendente

En esta fase del desarrollo, se buscó la presentación de datos de manera organizada y de fácil comprensión. Se ideó la forma de proporcionar información detallada sobre qué peso patrón se utilizó para cada medición, comparándolo con los resultados y mostrando los valores ingresados por el usuario en la interfaz. Esta presentación tenía la intención de guiar al usuario para que comprendiera a qué peso correspondía cada medición y con qué pesa se había calibrado. Además, se incluyó la incertidumbre de la pesa en kilogramos y el indicador aparente relacionado con la temperatura ambiente del entorno de la investigación. Estos valores variaban en función de los datos ingresados por el usuario. En esta tabla, también se obtuvieron los valores de la masa específica del aire, cuyos resultados eran similares, aunque podían variar según la cantidad de decimales



ingresados en los datos de entrada. Se sugiere utilizar la mayor cantidad de decimales posible para obtener resultados más precisos, como se muestra en la tabla 25-26.

**Tabla 25 – Ajuste aparente vs Indicado ascendente.**

	Puntos	Masa nominal kg	Incertidumbre (u) kg	Masa indicada kg	Masa aparente kg	Incertidumbre masa aparente kg	Indicado - aparente kg	Temperatura ambiente indicada C°
▶	1	0	0	0,001	0,001005215358...	0	-5,21535841777...	27,7
	2	3	1,856071119327...	2,997	3,012630429178...	1,899058102606...	-0,01563042917...	27,7
	3	0	0	0,001	0,001005215358...	0	-5,21535841777...	27,7
	4	6	4,236153915995...	5,993	6,024255642997...	4,315045390159...	-0,03125564299...	27,7
	5	0	0	0,001	0,001005215358...	0	-5,21535841777...	27,7
	6	9	4,761039802396...	8,998	9,044927795043...	4,907892238273...	-0,04692779504...	27,7
	7	0	0	0,001	0,001005215358...	0	-5,21535841777...	27,7
	8	12	8,658666179037...	11,977	12,03946434776...	8,807961939574...	-0,06246434776...	27,7
	9	0	0	0,001	0,001005215358...	0	-5,21535841777...	27,7
	10	15	9,458990432387...	14,964	15,04204262336...	9,657862501498...	-0,07804262336...	27,7
	11	0	0	0,001	0,001005215358...	0	-5,21535841777...	27,7
	12	18	9,639372386208...	17,996	18,08985559008...	9,930741152351...	-0,09385559008...	27,7

Fuente: propia

**Tabla 26 – Ajuste aparente vs Indicado ascendente 2.**

	Incertidumbre masa aparente kg	Indicado - aparente kg	Temperatura ambiente indicada C°	Temperatura ambiente incertidumbre C°	Presión atmosférica Indicación barómetro mbar/abs	Presión atmosférica Incertidumbre mbar/abs	Masa específica del aire Indicado kg/m3	Masa específica del aire Incertidumbre kg/m3s
▶	0	-5,21535841777...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	1,899058102606...	-0,01563042917...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	0	-5,21535841777...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	4,315045390159...	-0,03125564299...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	0	-5,21535841777...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	4,907892238273...	-0,04692779504...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	0	-5,21535841777...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	8,807961939574...	-0,06246434776...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	0	-5,21535841777...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	9,657862501498...	-0,07804262336...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	0	-5,21535841777...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...
	9,930741152351...	-0,09385559008...	27,7	0,245	1021,7	0,057735026918...	1,183279110656...	0,000965931500...

Fuente: propia

### 6.2.3 Ajuste aparente vs el indicado descendente

Esta fase de desarrollo se creó siguiendo un proceso similar al ajuste aparente ascendente. La distinción clave radica en esta etapa, los datos de entrada relativos a los pesos pueden variar, manteniendo constante la temperatura ambiente y la presión atmosférica del lugar de la investigación, que en este caso corresponde a la sede de la Universidad Antonio Nariño. Estos parámetros se pueden ajustar y personalizar según las condiciones específicas del entorno laboral y la época del año. El propósito fundamental de esta parte del software es determinar la masa específica del aire, junto con la incertidumbre asociada a este resultado, como se detalla en la tabla 27-28.

**Tabla 27** – Ajuste aparente vs Indicado descendente.

Principal - FuelCostUAN  
Calibración    Calculos    Datos Finales

Ajuste Aparente Vs Indicado Descendente

Puntos	Masa nominal kg	Incertidumbre (u) kg	Masa indicada kg	Masa aparente kg	Incertidumbre masa aparente kg	Indicado - aparente kg	Temperatura ambiente indicada C°
1	0	0	0	0	0	0	27,5
2	3	1,856071119327...	3	3,015646075253...	1,901051746396...	-0,01564607525...	27,5
3	0	0	0	0	0	0	27,5
4	6	4,236153915995...	6	6,031292150506...	4,320248626337...	-0,03129215050...	27,5
5	0	0	0	0	0	0	27,5
6	9	4,761039802396...	9	9,046938225759...	4,909306142519...	-0,04693822575...	27,5
7	0	0	0	0	0	0	27,5
8	12	8,658666179037...	12	12,06258430101...	8,825195730003...	-0,06258430101...	27,5
9	0	0	0	0	0	0	27,5
10	15	9,458990432387...	15	15,07823037626...	9,681552119283...	-0,07823037626...	27,5
11	0	0	0	0	0	0	27,5
12	18	9,639372386208...	18	18,09387645151...	9,933587034496...	-0,09387645151...	27,5

Fuente: propia

**Tabla 28** – Ajuste aparente vs Indicado descendente 2.

Principal - FuelCostUAN  
Calibracion    Calculos    Datos Finales

Ajuste Aparente Vs Indicado Descendente

	Incertidumbre masa aparente kg	Indicado - aparente kg	Temperatura ambiente indicada C°	Temperatura ambiente incertidumbre C°	Presión atmosférica Indicación barómetro mbar/abs	Presión atmosférica Incertidumbre mbar/abs	Masa específica del aire Indicado kg/m <sup>3</sup>	Masa específica del aire Incertidumbre kg/m <sup>3</sup> s
▶	0	0	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	1,901051746396...	-0,01564607525...	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	0	0	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	4,320248626337...	-0,03129215050...	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	0	0	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	4,909306142519...	-0,04693822575...	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	0	0	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	8,825195730003...	-0,06258430101...	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	0	0	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	9,681552119283...	-0,07823037626...	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	0	0	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...
	9,933587034496...	-0,09387645151...	27,5	0,245	1021,7	0,057735026918...	1,184066257911...	0,000967213972...

Fuente: propia

Los resultados alcanzados en esta fase del software coinciden con los previamente investigados por la Universidad Antonio Nariño. Esto respalda la precisión de los cálculos relacionados con la masa específica, confirmando que el código opera eficazmente al generar resultados rápidos, comprensibles y precisos.

### 6.2.4 Resultados finales carga ascendente

Esta etapa del software se considera la fase final, ya que a través de polinomios de diferentes grados se ha determinado la incertidumbre asociada al peso del instrumento. Se deben tener en cuenta factores como la gravedad, las incertidumbres de los patrones, la incertidumbre del instrumento y el ajuste de la incertidumbre, que se calcula a partir de estas incertidumbres. Se ha calculado la incertidumbre patrón mediante la raíz cuadrada de la suma de las incertidumbres elevadas al cuadrado, y con este valor, se ha calculado la incertidumbre expandida, que equivale al valor de la incertidumbre patrón multiplicado por 2. El error en gramos se obtiene restando la masa indicada de la masa ajustada para

las 11 muestras proporcionadas por el usuario, se evidenció que los resultados obtenidos en la tabla 29.

**Tabla 29** – Resultado final carga ascendente.

Principal - FuelCostUAN

Calibracion    Calculos    Datos Finales

RESULTADOS FINALES Carga Ascendente

	Masa indicada kg	Masa ajustada kg	Incertidumbre de ajuste kg	Incertidumbre del patrón kg	Incertidumbre del instrumento kg	Incertidumbre patrón (u) kg	Incertidumbre Expandida (U) kg	Error kg
▶	0,001000000000...	0,001005199999...	0,000202097320...	0	0,000580352490...	0,000614534246...	0,001229068493...	-5,19999997999...
	2,997	3,012584399999...	0,000202097320...	1,856071119327...	0,000580352490...	0,000614537049...	0,001229074099...	-0,01558439999...
	5,993	6,024163599999...	0,000202097320...	4,236153915995...	0,000580352490...	0,000614548847...	0,001229097694...	-0,03116359999...
	8,998	9,044789599999...	0,000202097320...	4,761039802396...	0,000580352490...	0,000614552689...	0,001229105378...	-0,04678959999...
	11,977	12,039280399999...	0,000202097320...	8,658666179037...	0,000580352490...	0,000614595243...	0,001229190486...	-0,06228039999...
	14,964	15,077999999999...	0,000202097320...	9,458990432387...	0,000580352490...	0,000614607039...	0,001229214078...	-0,11399999999...
	17,996	18,0936	0,000202097320...	9,639372386208...	0,000580352490...	0,000614609842...	0,001229219684...	-0,09759999999...
	20,992	21,109199999999...	0,000202097320...	1,652187943304...	0,000580352490...	0,000614756303...	0,001229512607...	-0,11719999999...
	23,994	24,124799999999...	0,000202097320...	1,666418314829...	0,000580352490...	0,000614760144...	0,001229520289...	-0,13079999999...
	26,989	27,1404	0,000202097320...	1,709371229429...	0,000580352490...	0,000614771937...	0,001229543875...	-0,15139999999...
	29,987	30,156	0,000202097320...	1,856071119327...	0,000580352490...	0,000614814476...	0,001229628952...	-0,16900000000...

Fuente: propia

Se observa que los resultados obtenidos muestran cierta precisión, y se destaca que la cantidad de decimales utilizada en el software tiene un impacto en la precisión de los resultados. Los resultados presentados en el Front del software son más exactos cuando se utilizan más decimales. Sin embargo, es importante señalar que las variaciones en los resultados en comparación con los calculados en Excel son mínimas. Para lograr esto, se ha utilizado un lenguaje de programación que incluye estructuras condicionales (verdadero o falso) para validar datos, así como bucles anidados para presentar los resultados de manera clara en la interfaz del software. Además, se ha diseñado el software de manera que sea claro y de fácil comprensión.

## 6.2.5 Resultados finales carga descendente

En esta etapa del desarrollo del software, se utilizó la misma programación que en el caso anterior, donde se generaron los resultados presentados en la tabla 26. La principal



diferencia radica en que, para este caso, los valores ingresados por el usuario son de forma descendente, lo que altera el orden de los resultados. A pesar de esta variación en los valores iniciales, el método de cálculo de los resultados sigue siendo el mismo para ambas situaciones, ya sea para cargas ascendentes o descendentes. Al revisar los resultados, se observa que son consistentes con los obtenidos en la investigación previamente realizada por la Universidad Antonio Nariño [15], lo que refuerza la confiabilidad de los resultados. Estos resultados se pueden apreciar en la tabla 30.

**Tabla 30** – Resultado final carga descendente.

Principal - FuelCostUAN

Calibración    Cálculos    Datos Finales

RESULTADOS FINALES Carga Descendente

	Masa indicada kg	Masa ajustada kg	Incertidumbre de ajuste kg	Incertidumbre del patrón kg	Incertidumbre del instrumento kg	Incertidumbre patrón (u) kg	Incertidumbre Expandida (U) kg	Error kg
▶	0	8,376666664666...	0,000202097320...	0	0,000580352490...	0,000614534246...	0,001229068493...	-8,376666664666...
	3	3,015599999999...	0,000202097320...	1,856071119327...	0,000580352490...	0,000614537049...	0,001229074099...	-0,01559999999...
	6	6,031199999999...	0,000202097320...	4,236153915995...	0,000580352490...	0,000614548847...	0,001229097694...	-0,03119999999...
	9	9,046799999999...	0,000202097320...	4,761039802396...	0,000580352490...	0,000614552689...	0,001229105378...	-0,04679999999...
	12	12,06239999999...	0,000202097320...	8,658666179037...	0,000580352490...	0,000614595243...	0,001229190486...	-0,06239999999...
	15	15,07799999999...	0,000202097320...	9,458990432387...	0,000580352490...	0,000614607039...	0,001229214078...	-0,07799999999...
	18	18,09359999999...	0,000202097320...	9,639372386208...	0,000580352490...	0,000614609842...	0,001229219684...	-0,09359999999...
	21	21,10919999999...	0,000202097320...	1,652187943304...	0,000580352490...	0,000614756303...	0,001229512607...	-0,10919999999...
	24	24,12479999999...	0,000202097320...	1,666418314829...	0,000580352490...	0,000614760144...	0,001229520289...	-0,12479999999...
	27	27,14039999999...	0,000202097320...	1,709371229429...	0,000580352490...	0,000614771937...	0,001229543875...	-0,14039999999...
	30	30,15599999999...	0,000202097320...	1,856071119327...	0,000580352490...	0,000614814476...	0,001229628952...	-0,15599999999...

Fuente: propia

## 7. Conclusiones y recomendaciones

Esta tesis ha resultado en el desarrollo de un software que agiliza los cálculos, permitiendo al usuario obtener resultados de manera clara y sin margen para posibles errores de interpretación. Los cálculos y operaciones se ejecutan en el módulo de modelo, donde se responde a las solicitudes del usuario a través de la vista y se controla mediante el controlador. Este software contribuye a una comprensión más profunda de la temperatura de funcionamiento del motor, lo que a su vez mejora la toma de muestras en el laboratorio y facilita la estandarización del proceso de investigación. Además, este enfoque ha permitido abordar con éxito cada uno de los objetivos específicos establecidos, contribuyendo así a la consecución del objetivo general del proyecto.

En relación con el primer punto específico - Llevar a cabo un análisis exhaustivo de los datos y las plantillas de Excel, evaluando su estructura y contenido, y comenzar la implementación del lenguaje C# con el fin de transformarlos en un software funciona. En el proceso de desarrollo, se llevó a cabo la programación de cada una de las variables que la vista del software solicitará. Estas solicitudes son luego enviadas al controlador, que a su vez las remite al modelo. En el modelo, se ejecutan las operaciones preprogramadas en segundo plano, y el resultado se devuelve a la vista. Una característica importante de este diseño es que las operaciones no pueden modificarse a menos que se altere el código en el que están programadas, lo que limita la capacidad del usuario para cambiar parámetros fundamentales y, al mismo tiempo, aumenta la confiabilidad de los resultados

Para el segundo objetivo - Desarrollar un software que incorpore una representación visual de la información y su disposición para su presentación al usuario. Se diseñó una vista agradable y de fácil manejo, que incorpora los colores representativos de la Universidad Antonio Nariño y coloca su logotipo en la parte superior izquierda. Esto personaliza el software para uso exclusivo de la universidad en sus procesos de investigación. La interfaz visual proporciona una presentación amigable que facilita la comprensión de los valores obtenidos, lo que a su vez simplifica la interpretación de los resultados y permite una comprensión más clara del estado del motor.

Para el tercer objetivo - Desarrollar un software utilizando C# con el propósito de lograr una fácil instalación y portabilidad, permitiendo su implementación en diversos equipos y su uso en cualquier entorno. En este punto, se llevó a cabo una prueba en varios PC con el sistema operativo Windows, y los resultados mostraron que el software funcionó correctamente en todos los casos. El único requisito es tener actualizado el .NET Framework a la versión 6.0, que es proporcionado de forma gratuita por Microsoft y se instala de manera sencilla

En última instancia, se puede concluir que el software ofrece oportunidades para futuros proyectos de mejora. Estas mejoras podrían incluir la capacidad de calcular el trabajo, establecer una conectividad directa con el motor a través de puertos USB para adquirir datos de temperatura y peso del combustible en tiempo real, sin necesidad de ingreso manual de valores. Esto reduciría significativamente la posibilidad de errores humanos y, en última instancia, mejoraría la calidad de los resultados en investigaciones futuras.

## Bibliografía

- [1] Heywood, John B. (1988) "Internal Combustion Engine Fundamentals"
- [2] Dekraker, P. Barba, D. Moskalik, A. & Butters, K. (2018). Constructing Engine Maps for Full Vehicle Simulation Modelling, documento técnico SAE 2018-01-1412, 2018, 1(1), 12. <https://doi.org/10.4271/2018-01-1412>.
- [3] Pulkrabek, W. W. (2004). Engineering fundamentals of the internal combustion engine.
- [4] Gómez Ormeño, Mauricio Andrés (2016-01-28). Evaluación de los datos del consumo de combustible calculado por módulos de control electrónico en motores de combustión interna. <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/6505>.
- [5] Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia. <http://repositorio.uan.edu.co/handle/123456789/2536>.
- [6] Diego López Donaire (16/11/2021). Motor Wankel: qué es, cómo funciona, partes, ventajas y desventajas. [Online]. Available: <https://www.actualidadmotor.com/ventajas-y-desventajas-del-motor-wankel/>
- [7] Santiago Sanz Acebes, motores: transporte y mantenimiento de vehículos, España: Editorial Editex, S. A, 2011.
- [8] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>
- [9] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/tutorials/oop>
- [10] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/aspnet/mvc/overview/older-versions-1/overview/understanding-models-views-and-controllers-cs>
- [11] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/desktop/winforms/overview/?view=netdesktop-7.0>
- [12] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/language-specification/variables>



- [13] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/archive/msdn-magazine/2014/may/csharp-best-practices-dangers-of-violating-solid-principles-in-csharp>
- [14] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/programming-guide/classes-and-structs/abstract-and-sealed-classes-and-class-members>
- [15] Buendia Waut, Jainer Jesus Arias Araujo, Gary, Diseño de una herramienta de cálculo para evaluar la confiabilidad metrológica del sistema térmico y gravimétrico del motor de combustión interna de la UAN-Sede Puerto Colombia, [Online]. Available <http://repositorio.uan.edu.co/handle/123456789/2536>
- [16] Aguiar García, J; Delgado Cabello, J. (2011). Física II OCW- Universidad de Málaga <http://ocw.uma.es>
- [17] NTC 2194, Numeral 6.11 Vocabulario de términos básicos y generales en metrología
- [18] NTC, GTC51 (1997-11-26).
- [19] Mendoza, H, Bautista, G. (2002). Probabilidad y Estadística. Universidad Nacional de Colombia, <http://www.virtual.unal.edu.co/cursos/ciencias/2001065/>. Licencia: Creative Commons BY-NC-ND.
- [20] VIM, 3ra Edición (2012)
- [21] VIM, 3ra Edición (2012) [sección. 2.26 (3.9)]
- [22] VIM, 3ra Edición (2012) [sección. 2.29]
- [23] Estudo das incertezas na medição de espessura por ultrassom em peças aeronáuticas(Bruno César de Andrade Silva-2012)
- [24] Alquicira, C. (2017, 05 de noviembre) Programación Orientada a Objetos. Conogasi, Conocimiento para la vida. Fecha de consulta: septiembre 24, 2023.
- [25] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/visualstudio/ide/reference/encapsulate-field?view=vs-2022>
- [26] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/tutorials/inheritance>
- [27] Microsoft. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/object-oriented/polymorphism>