

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA MECÁNICA**

**DIAGRAMAS TERMODINÁMICOS INTERACTIVOS DE LOS  
CICLOS DE POTENCIA OTTO Y DIESEL**

**Componente: DIAGRAMAS TERMODINÁMICOS  
INTERACTIVOS DEL CICLO DIESEL**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO  
MECÁNICO**

**ANTHONY FERNANDO ENCALADA CAIZAPANTA**

**Anthony.encalada@epn.edu.ec**

**DIRECTOR: MIGUEL DARIO ORTEGA LÓPEZ**

**Miguel.ortega@epn.edu.ec**

**DMQ, agosto 2022**

## CERTIFICACIONES

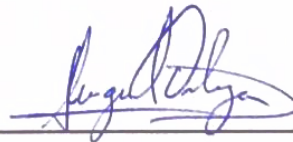
Yo, Anthony Fernando Encalada Caizapanta declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



---

ANTHONY ENCALADA

Certifico que el presente trabajo de integración curricular fue desarrollado por Anthony Fernando Encalada Caizapanta, bajo mi supervisión.



---

MIGUEL ORTEGA  
DIRECTOR

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Anthony Fernando Encalada Caizapanta

Miguel Darío Ortega López

## **DEDICATORIA**

Este trabajo al igual que toda la carrera que he cursado se lo dedico a mis padres Sofía y Johnny, por ser el apoyo incondicional durante todo mi desarrollo, a mis abuelos Lourdes, Fernando, Teresa y Antonio, a todos mis tíos y primos por aportar con mi crecimiento académico y personal y ser mi apoyo emocional, a Carlos por la cooperación en este último año, a Fatima por la ayuda día a día para poder lograr mis objetivos y a mis amigos de la universidad en especial a Geovanny, Walter, Brayan, Kevin, Eri y Steven.

Sin todas las personas nombradas anteriormente todo esto no sería posible.

## **AGRADECIMIENTO**

Agradezco en primer lugar a Dios, a mi familia y amigos cercanos por ayudarme siempre a mejorar, también agradezco a mi tutor el Ing. Miguel Ortega por ser el guía de este proyecto y brindarme el conocimiento necesario para cumplirlo, a los profesores que he tenido en todo mi proceso formativo en especial al Ing. Álvaro Puente<sup>†</sup> y la Escuela Politécnica Nacional por brindarme las herramientas suficientes para completar esta meta.

Agradezco también a las grandes comunidades de internet de páginas como DelftStack y Stackoverflow que brindan un gran servicio a la comunidad aclarando dudas sobre programación y brindando ejemplos y finalmente agradezco a mi mascota Papo que fue el acompañante de todas las malas noches durante esta carrera.

# ÍNDICE DE CONTENIDO

<b>CERTIFICACIONES</b> .....	<b>I</b>
<b>DECLARACIÓN DE AUTORÍA</b> .....	<b>II</b>
<b>DEDICATORIA</b> .....	<b>III</b>
<b>AGRADECIMIENTO</b> .....	<b>IV</b>
<b>ÍNDICE DE CONTENIDO</b> .....	<b>V</b>
<b>RESUMEN</b> .....	<b>IX</b>
<b>ABSTRACT</b> .....	<b>X</b>
<b>1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO</b> .....	<b>1</b>
1.1 Objetivo general .....	1
1.2 Objetivos específicos .....	1
1.3 Alcance .....	2
1.4 Marco teórico .....	3
Ciclos de potencia de gas .....	3
Ciclo Diesel Ideal .....	3
Energía .....	6
Entropía .....	7
Exergía .....	8
Python .....	8
Bibliotecas Python .....	9
Ventana Principal .....	10
Widgets de Tkinter .....	10
Marco .....	11
Etiqueta .....	11
Caja de texto .....	11
Botón .....	12
<b>2 METODOLOGÍA</b> .....	<b>13</b>
2.1 Recopilación de base teórica .....	15
Métodos de cálculo para resolución del ciclo Diesel .....	15
Método aproximado .....	15
Método exacto .....	15
Creación de los diagramas de energía, entropía, exergía y su respectiva interpretación .....	16
Interpretación diagrama de energía .....	18
Interpretación diagrama de entropía .....	20
Interpretación diagrama de exergía .....	21
2.2 Diseño del software .....	22
Resolución numérica del ciclo Diesel utilizando Python .....	22
Datos de entrada .....	22
CoolProp .....	23
Resolución con el método aproximado utilizando Python .....	24
Resolución con el método exacto utilizando Python .....	25
Resolución gráfica del ciclo Diesel utilizando Python .....	26
Matplotlib .....	26
Interfaz gráfica Tkinter utilizando Python .....	29
Integración de la interfaz gráfica con la resolución numérica y la resolución gráfica .....	33
Software resultante y corrección de errores .....	36
<b>3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES</b> .....	<b>38</b>
3.1 Resultados .....	38
Análisis de error .....	38

Sensibilidad de variables .....	41
Análisis de sensibilidad en el método aproximado .....	41
Análisis de sensibilidad en el método exacto .....	43
Eficiencia, relación de compresión y relación de corte- admisión .....	47
Producto final .....	48
3.2 Conclusiones .....	50
3.3 Recomendaciones .....	51
<b>4 REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>52</b>
<b>REFERENCIAS .....</b>	<b>52</b>
<b>5 ANEXOS .....</b>	<b>53</b>
<b>ANEXO I: MANUAL DE USUARIO. ....</b>	<b>53</b>
<b>SOFTWARE Y SUS PARTES .....</b>	<b>1</b>
Acerca del Software DT-Diesel.....	1
Partes del Software .....	2
<b>INGRESO DE DATOS .....</b>	<b>3</b>
<b>DIAGRAMAS DE ENERGÍA, ENTROPÍA, EXERGÍA Y SU RESPECTIVA INTERPRETACIÓN .....</b>	<b>4</b>
Interpretación diagrama de energía.....	7
Interpretación diagrama de entropía.....	9
Interpretación diagrama de exergía .....	10
<b>FUNCIONES DEL SOFTWARE .....</b>	<b>11</b>

### Índice de figuras:

<b>Figura 1.1:</b> Motor de automóvil [1].....	3
<b>Figura 1.2:</b> Gráficos del ciclo Diesel. ....	4
<b>Figura 1.3:</b> Importación de bibliotecas en Python.....	9
<b>Figura 1.4:</b> Ventana principal realizada con Tkinter.....	10
<b>Figura 1.5:</b> Widgets principales de Tkinter. ....	11
<b>Figura 2.1:</b> Metodología. ....	14
<b>Figura 2.2</b> Diagrama base.....	16
<b>Figura 2.3:</b> Diagrama de energía.....	18
<b>Figura 2.4:</b> Diagrama de entropía.....	20
<b>Figura 2.5:</b> Diagrama de exergía.....	21
<b>Figura 2.6:</b> Diagrama p-V referencial. ....	22
<b>Figura 2.7:</b> Importación biblioteca CoolProp.....	23
<b>Figura 2.8:</b> Ejemplo de uso de CoolProp.....	24
<b>Figura 2.9:</b> Código de resolución numérica con método aproximado. ....	24
<b>Figura 2.10:</b> Código de resolución numérica con método exacto (parte 1). ....	25
<b>Figura 2.11:</b> Código de resolución numérica con método exacto (parte 2). ....	25
<b>Figura 2.12:</b> Código de resolución numérica con método exacto (parte 3). ....	26
<b>Figura 2.13:</b> Ejemplo de construcción de gráficos con Matplotlib. ....	26
<b>Figura 2.14:</b> Código para graficar rectas con dos puntos. ....	27
<b>Figura 2.15:</b> Código para graficar circunferencia.....	27
<b>Figura 2.16:</b> Comando para definir el título de un gráfico. ....	27
<b>Figura 2.17:</b> Ejemplo definición de variables iniciales para graficar.....	27
<b>Figura 2.18:</b> Ejemplo Ciclo For para encontrar los puntos del gráfico.....	28

<b>Figura 2.19:</b> Ejemplo de cálculo de los puntos del cuarto cuadrante para el gráfico de energía.	28
<b>Figura 2.20:</b> Ejemplo de almacenamiento de los puntos del gráfico.	28
<b>Figura 2.21:</b> Ejemplo de anotaciones y etiquetas en el gráfico.	29
<b>Figura 2.22:</b> Ejemplo diagrama de energía.	29
<b>Figura 2.23:</b> Importación de biblioteca Tkinter.	30
<b>Figura 2.24:</b> Ejemplo de creación de ventana y marco principales.	30
<b>Figura 2.25:</b> Código ejemplo de creación de etiquetas.	30
<b>Figura 2.26:</b> Ejemplo de uso de caja de texto y recolección de datos.	31
<b>Figura 2.27:</b> Código para insertar imagen como etiqueta.	31
<b>Figura 2.28:</b> Definición de mensajes de error.	32
<b>Figura 2.29:</b> Condiciones para mostrar los mensajes de error.	32
<b>Figura 2.30:</b> Ejemplo de creación de un botón.	32
<b>Figura 2.31:</b> Comando para finalizar la construcción de la ventana principal.	32
<b>Figura 2.32:</b> Pantalla de ingreso de datos.	33
<b>Figura 2.33:</b> Código para obtener los datos ingresados.	33
<b>Figura 2.34:</b> Ejemplo de construcción de función con el módulo de resolución numérica.	34
<b>Figura 2.35:</b> Ejemplo de creación de la gráfica en la ventana principal.	35
<b>Figura 2.36:</b> Definición de clase para modificar características de Toolbar.	35
<b>Figura 2.37:</b> Código para añadir Toolbar al diagrama final.	35
<b>Figura 2.38:</b> Botones asociados a los módulos principales.	36
<b>Figura 2.39:</b> Ejemplo de obtención de resultados de exergía.	36
<b>Figura 3.1:</b> Comparación del diagrama de energía método aproximado.	43
<b>Figura 3.2:</b> Comparación del diagrama de energía método exacto.	45
<b>Figura 3.3:</b> Comparación del diagrama de entropía método exacto.	45
<b>Figura 3.4:</b> Comparación del diagrama de exergía método exacto.	46
<b>Figura 3.5:</b> Ejemplo de aumento de eficiencia del Ciclo Diesel (a).	48
<b>Figura 3.6:</b> Ejemplo de aumento de eficiencia del Ciclo Diesel (b).	48
<b>Figura 3.7:</b> Software final con sus partes.	49
<b>Figura 3.8:</b> Ejemplo de conversión a archivo ejecutable (.exe).	50

### Índice de tablas

<b>Tabla 1.1:</b> Simbología.	5
<b>Tabla 3.1:</b> Errores de cálculo numérico con el método aproximado.	39
<b>Tabla 3.2:</b> Errores de cálculo numérico con el método aproximado.	40
<b>Tabla 3.3:</b> Análisis sensibilidad método aproximado.	42
<b>Tabla 3.4:</b> Análisis sensibilidad método exacto.	43
<b>Tabla 3.5:</b> Análisis sensibilidad método exacto (continuación).	44
<b>Tabla 3.6:</b> Análisis sensibilidad método exacto exergía.	46

### Índice de ecuaciones

<b>Ecuación 1.1:</b> Ecuación de gas ideal.	6
<b>Ecuación 1.2:</b> Relación de compresión.	6
<b>Ecuación 1.3:</b> Relación de corte-admisión.	6
<b>Ecuación 1.4:</b> Proceso de compresión isentrópico.	6
<b>Ecuación 1.5:</b> Proceso de expansión isentrópico.	6
<b>Ecuación 1.6:</b> Balance de energía.	7
<b>Ecuación 1.7:</b> Calor añadido.	7
<b>Ecuación 1.8:</b> Calor rechazado.	7
<b>Ecuación 1.9:</b> Eficiencia térmica.	7
<b>Ecuación 1.10:</b> Exergía (sin flujo) [1].	8
<b>Ecuación 2.1:</b> Linealización para el primer cuadrante.	16



<b>Ecuación 2.2:</b> Linealización para el segundo cuadrante. ....	17
<b>Ecuación 2.3:</b> Linealización para el tercer cuadrante. ....	17
<b>Ecuación 2.4:</b> Linealización para el cuarto cuadrante. ....	17
<b>Ecuación 2.5:</b> Variación de energía interna de E1 a E2. ....	19
<b>Ecuación 2.6:</b> Variación de energía interna de E2 a E3. ....	19
<b>Ecuación 2.7:</b> Variación de energía interna de E3 a E4. ....	19
<b>Ecuación 2.8:</b> Variación de energía interna de E4 a E1. ....	19

## RESUMEN

En el presente proyecto se realiza un software de diagramas interactivos para ciclo Diesel en la plataforma de programación Python y se establecen como métodos de cálculo para la resolución numérica los métodos aproximado y exacto con las consideraciones del caso para cada uno, las propiedades del aire gas ideal para el cálculo del método exacto son proporcionados por la biblioteca CoolProp; el software tiene como base una interfaz gráfica con el uso de la biblioteca Tkinter como principal herramienta de diseño gráfico, los diagramas resultantes se realizan con la biblioteca Matplotlib que permite visualizar los resultados de forma interactiva y estética.

El software de diagramas interactivos para el ciclo Diesel muestra resultados numéricos de la resolución del Ciclo Diesel acompañado de los diagramas interactivos de: Energía, Entropía y Exergía, los cuales han sido creados específicamente para el presente proyecto por los autores de los dos componentes (Ciclo Otto y Ciclo Diesel) con la guía del Tutor el Ing. Miguel Ortega.

El resultado del componente es un software que cuenta con una interfaz gráfica amigable para el usuario, y brinda la posibilidad de variar los datos de entrada y obtener resultados inmediatos.

El análisis de variables en los resultados ayuda a comprender la importancia de las relaciones volumétricas del ciclo Diesel (relación de compresión y relación de corte-admisión), ya que definen la eficiencia térmica del ciclo.

El producto final del componente es un archivo ejecutable y un documento de manual de usuario.

**PALABRAS CLAVE:** Termodinámica, Ciclo Diesel, Software, Diagramas interactivos, Python

## ABSTRACT

Design a software to obtain Interactive thermodynamical diagrams for Diesel using Python with two calculation methods for numeric solving, exact method and approximate method using the assumptions for each one, Coolprop's library provides all the necessary properties of air to calculate using exact method; the graphic interface in this software is done using Tkinter library from Python as principal graphic tool, the thermodynamical diagrams are done using Matplotlib library, considering this library as the best option for result visualization at interactive and esthetic sense.

This software shows numerical results of Diesel cycle solution with the Energy, entropy and exergy interactive diagrams, these interactive diagrams have been created by the authors of this project under the supervision of Project tutor Ing. Miguel Ortega.

The output software has a friendly graphical interface for the user and offers the possibility to vary initial data and get immediate results.

The analysis from influence of properties at results helps to understand the importance of volumetric ratios of Diesel (compression ratio and cut-off ratio), because these ratios define the thermal efficiency of the Diesel cycle.

**KEYWORDS:** Thermodynamic, Diesel Cycle, Software, Interactive diagrams, Python.

# **1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO**

Este componente realiza el diseño de un nuevo módulo para el software diagramas termodinámicos interactivos, en este módulo se muestran resultados numéricos y gráficos del ciclo Diesel, conforme a los datos ingresados por el usuario, los cuales son almacenados para ser usados en los cálculos correspondientes, estos datos deber ser ingresados con el formato y unidades establecidas más adelante en el presente documento, el software tiene incorporadas funciones para resolver problemas del ciclo Diesel utilizando el método exacto y el método aproximado, tomando en cuenta al aire gas-ideal como fluido de trabajo y las suposiciones normalmente utilizadas en la termodinámica clásica; los resultados numéricos serán presentados en una ventana de interfaz gráfica en compañía de las gráficas de energía, entropía y exergía asociadas a este ciclo, para todos los casos evaluados por el usuario. Los resultados serán comprobados y analizados para garantizar el correcto funcionamiento del software; adicionalmente se establece la interpretación de los resultados gráficos explicando cada aspecto importante para su comprensión.

El módulo se realiza en la plataforma Python y será de libre acceso para los estudiantes que lo requieran, complementando el proceso de aprendizaje de la asignatura de Termodinámica con herramientas tecnológicas y gráficas de fácil uso.

El resultado del componente es un archivo ejecutable que se ejecuta en un sistema operativo Windows 8 o superiores y un manual de usuario que describe y explica el software.

## **1.1 Objetivo general**

Implementar el módulo de ciclo Diesel para el software diagramas interactivos

## **1.2 Objetivos específicos**

1. Realizar trabajo investigativo para obtener el estado del arte y fundamentación teórica básica del ciclo Diesel y de la programación numérica y visual en Python.
2. Diseñar el código base para el módulo del ciclo Diesel para resolución numérica

con el método exacto y aproximado.

3. Conceptualizar la representación gráfica de ciclos en sistema cerrado y desarrollar la programación de las gráficas de energía, entropía y exergía del ciclo Diesel que se presentarán asociadas a los resultados numéricos.
4. Comprobar los resultados obtenidos con el programa para corroborar la precisión y funcionalidad del mismo.
5. Implementar un manual del usuario para el uso del módulo de ciclo Diesel.

### **1.3 Alcance**

Entre los motores de combustión interna (MCI) se encuentra el motor Diesel, y al igual que otros MCI estos trabajan en un ciclo abierto por la imposibilidad de recircular los gases de combustión. El modelamiento de estos motores desde el punto de vista termodinámico se realiza incorporando algunas suposiciones, entre las que se encuentran el suponer que la sustancia de trabajo siempre es aire gas ideal y el suponer, que al no existir gases de combustión el ciclo no requiere ser un ciclo abierto y puede cerrarse si se incorpora un rechazo de calor que reemplaza al escape y admisión de los motores [1].

Tomando en cuenta herramientas computacionales existentes con funcionalidad y propósito parecido al que se implementará en este proyecto, como es el caso de [2], su autor García L. plantea una herramienta interactiva con el fin de la resolución numérica a los ciclos ideales. Debido a las limitaciones de estos trabajos mediante la presente propuesta se innovará al implementar gráficas interactivas de energía, entropía y exergía para el Ciclo Diesel.

En el desarrollo del módulo se requiere el estudio de conceptos de la plataforma Python en la cual se realizará el módulo de ciclo Diesel, se considera la programación de la interfaz, la construcción y desarrollo del algoritmo con sus respectivas funciones y las salidas esperadas que contarán con los diagramas interactivos de energía, entropía y exergía del ciclo Diesel y los resultados numéricos correspondientes.

Paralelamente se realizará una evaluación de los resultados numéricos y visuales obtenidos para comprobar la funcionalidad del módulo y asegurar que esta herramienta contribuya con el aprendizaje en la asignatura de termodinámica.

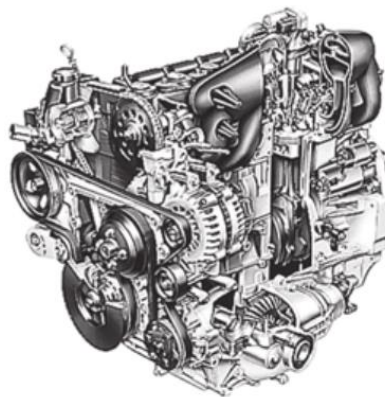
Finalmente, se realizará la implementación de un manual de usuario para el uso del programa de diagramas interactivos en lo que respecta al módulo ciclo Diesel.

## 1.4 Marco teórico

En esta sección se encuentra la recopilación de información sobre el Ciclo Diesel y la plataforma Python para establecer las bases del proyecto y definir la ruta para realizarlo de manera correcta.

### Ciclos de potencia de gas

Se define como ciclo de potencia a un ciclo que produce un trabajo neto, utilizando un fluido de trabajo que en todo el ciclo está en fase gaseosa, los ciclos de potencia de gas son el principio de funcionamiento de las máquinas utilizadas para producir una salida de potencia como los motores (**Figura 1.1**) [1].



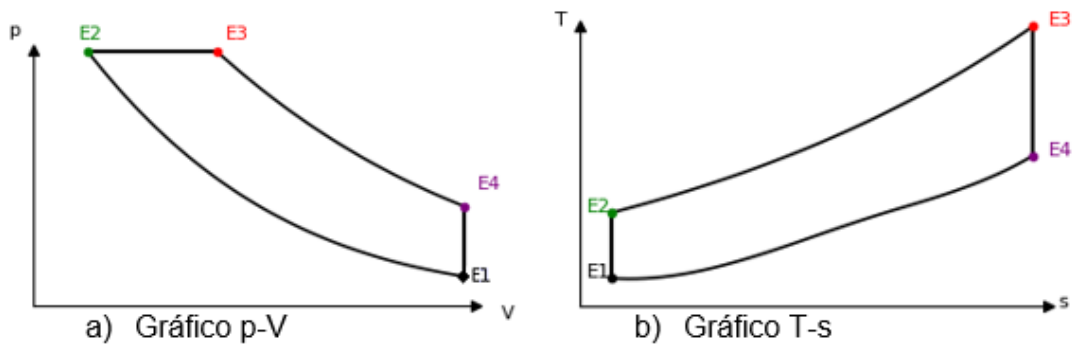
**Figura 1.1:** Motor de automóvil [1].

### Ciclo Diesel Ideal

El ciclo Diesel ideal es un ciclo de potencia de gas de tipo cerrado, ya que tiene como suposición inicial que el fluido de trabajo recircula a través del ciclo y vuelve a su estado inicial una vez cumplido el ciclo, el ciclo cuenta con 4 procesos termodinámicos [1].

Del estado 1 al 2 se da una compresión isentrópica en este proceso existe un ingreso de

trabajo, del estado 2 al 3 se produce la adición de calor a presión constante seguido de la combustión en este proceso existe una salida de trabajo, posteriormente del estado 3 al 4 se realiza la expansión isentrópica, en este proceso hay una salida de trabajo y finalmente del estado 4 al 1 se da el rechazo de calor a volumen constante cumpliendo así el ciclo y regresando al fluido de trabajo a su estado inicial (**Figura 1.2**) [1].



**Figura 1.2:** Gráficos del ciclo Diesel.

El principio de funcionamiento de los motores de combustión sin chispa está basado en este ciclo, porque el fluido de trabajo sobrepasa la temperatura de autoencendido del combustible y de este modo se inicia la combustión [1].

El ciclo Diesel ideal toma como consideraciones lo siguiente:

- No existe ningún tipo de fricción, por lo tanto, no se producen caídas de presión del fluido de trabajo al fluir por los diferentes componentes.
- La transferencia de calor entre componentes y tuberías del motor es despreciable se asume que todo se encuentra aislado de manera perfecta.
- Los procesos de compresión y expansión son isentrópicos.
- EL ciclo Diesel trabaja con relaciones de compresión altas comúnmente entre 12 y 24 y valores de relación de corte-admisión de máximo 2.

[3].

En la **Tabla 1.1** que se presenta a continuación se especifica la simbología utilizada para la resolución del ciclo Diesel, cuenta con la descripción correspondiente de cada propiedad y las unidades correspondientes para su uso en las ecuaciones que se presentan más adelante.

**Tabla 1.1.** Simbología.

<b>Letra Representativa</b>	<b>Propiedad</b>	<b>Unidades</b>
$p$	Presión	[kPa]
$V$	Volumen	[m <sup>3</sup> ]
$m$	Masa	[kg]
$R$	Constante de gas Para el aire=0.287	[kJ/kg·K]
$T$	Temperatura	[K]
$r$	Relación de compresión	-
$r_c$	Relación de corte admisión	-
$q_a$	Calor añadido	[kJ/kg]
$q_r$	Calor rechazado	[kJ/kg]
$\eta$	Eficiencia térmica	[%]
$C_p$	Calor específico a presión constante Para el aire=1.005	[kJ/kg·K]
$C_v$	Calor específico a volumen constante Para el aire=0.718	[kJ/kg·K]
$k$	Relación de calores específicos Para el aire=1.4	-
$u$	Energía interna	[kJ/kg]
$s$	Entropía	[kJ/kg·K]
$v$	Volumen específico	[m <sup>3</sup> /kg]
$\phi$	Exergía sin flujo	[kJ/kg]

A continuación, se establecen las ecuaciones para la resolver el ciclo Diesel, estas ecuaciones se encuentran en diversos libros de termodinámica, así como en el material referenciado en este documento (REFERENCIAS BIBLIOGRÁFICAS).



El fluido de trabajo definido para la resolución del ciclo Diesel es el aire gas ideal, por lo cual se utiliza la **Ecuación 1.1** para poder relacionar los diferentes estados del fluido.

$$pV = mRT$$

**Ecuación 1.1:** Ecuación de gas ideal.

La relación de volúmenes al inicio y al final de la compresión isentrópica del ciclo se denomina relación de compresión viene dada por la **Ecuación 1.2**.

$$r = \frac{V_1}{V_2}$$

**Ecuación 1.2:** Relación de compresión.

La relación de volúmenes al final de la adición de calor y al inicio de la adición de calor se denomina relación de corte-admisión y viene dada por la **Ecuación 1.3**.

$$r_c = \frac{V_3}{V_2}$$

**Ecuación 1.3:** Relación de corte-admisión.

Las ecuaciones que relacionan los estados en los procesos isentrópicos se muestran en **Ecuación 1.4** y **Ecuación 1.5**.

$$p_1 V_1^k = p_2 V_2^k$$

**Ecuación 1.4:** Proceso de compresión isentrópico.

$$p_3 V_3^k = p_4 V_4^k$$

**Ecuación 1.5:** Proceso de expansión isentrópico.

## Energía

La energía se considera como la capacidad de generar un cambio, puede ser un movimiento, un aumento de temperatura, un cambio de estado, etc. En el caso del ciclo Diesel la energía interna del sistema genera un trabajo neto.

La primera ley de la termodinámica o principio de conservación de la energía relaciona las diferentes formas de energía que existen para definir este cambio neto a través de los

procesos que realiza un ciclo se define el balance de energía (**Ecuación 1.6**) que corresponde a la diferencia entre la energía total que ingresa al sistema y la energía total que sale del sistema y esto es igual al cambio total de energía del sistema.

$$\Delta E = E_{entrada} - E_{salida}$$

**Ecuación 1.6:** Balance de energía.

Las ecuaciones de adición y rechazo de calor para el ciclo Diesel se derivan de la aplicación de la primera ley de la termodinámica al sistema físico del ciclo Diesel, el cual consiste en un sistema cerrado de émbolo pistón al que se le añade calor a presión constante y posteriormente rechaza calor a volumen constante (**Ecuación 1.7** y **Ecuación 1.8**).

$$q_a = h_3 - h_2 = Cp(T_3 - T_2)$$

**Ecuación 1.7:** Calor añadido.

$$q_r = u_4 - u_1 = Cv(T_4 - T_1)$$

**Ecuación 1.8:** Calor rechazado.

Con las suposiciones de aire estándar frío la eficiencia térmica del ciclo Diesel se determina con la **Ecuación 1.9**.

$$\eta = 1 - \frac{q_r}{q_a}$$

**Ecuación 1.9:** Eficiencia térmica.

## Entropía

La entropía se puede definir como el grado de desorden molecular de un sistema en equilibrio, esta propiedad es fundamental para la aplicación de la segunda ley de la termodinámica que establece la irreversibilidad de los fenómenos físicos, entonces la entropía surge como el indicador de la evolución en el tiempo de un sistema aislado.

La variación de entropía en los sistemas cerrados se da como resultado de la transferencia de entropía relacionada a la transferencia de calor y la generación de entropía dentro de las fronteras del sistema en el caso del ciclo Diesel solo habrá una variación de entropía relacionada a la transferencia de calor, ya que los procesos de compresión y expansión son isentrópicos.

## Exergía

La exergía se define como la cantidad de energía aprovechable de un sistema, teniendo en cuenta las leyes de la termodinámica existe un límite en cuanto al trabajo que se puede realizar y para realizar este trabajo el sistema debe tener almacenado una energía que no se pueda usar en contra a las condiciones ambientales, es decir que la energía para realizar un trabajo se consume por completo el momento que las variables de estado se igualan con las propiedades del entorno; cuando existe una variación de entropía existe una reducción del trabajo útil. La exergía se calcula con la **Ecuación 1.10**.

$$\phi = (u - u_0) + (p_0 \cdot (v - v_0)) - (T_0 \cdot (s - s_0))$$

**Ecuación 1.10:** Exergía (sin flujo) [1].

Donde:

$p_0, T_0, s_0, u_0$ : Representan a las propiedades del estado ambiente.

## Python

Python es un lenguaje de programación utilizado principalmente para aplicaciones web, que ofrece una gran cantidad de complementos y documentación para su desarrollo, es un software libre que permite una programación más eficiente y compacta que otros lenguajes de programación y su ejecución se puede realizar en varias plataformas [4].

Las principales ventajas de uso de Python para la programación de software son:

- Puede ser utilizado para realizar casi cualquier cosa, ya que cuenta con una biblioteca general integrada pero también cuenta con un sinnúmero de bibliotecas con diversas funciones las cuales se pueden agregar al programa.
- La legibilidad del código es uno de los principales aspectos que hacen de Python un lenguaje fácil y rápido de aprender.
- Existe una gran cantidad de información de Python, sus complementos, funciones, sintaxis, etc.

[5].

## Bibliotecas Python

Las bibliotecas o librerías de Python son un conjunto de componentes, funciones y objetos que contribuyen con la construcción de un script, con ellas se agrega mayor funcionalidad, brindando la posibilidad de tener más herramientas de utilidad en un mismo software. En la **Figura 1.3** se muestran las bibliotecas y funciones importadas para realizar el presente componente [6].

```
import tkinter as tk
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
import CoolProp.CoolProp as cp
from CoolProp.CoolProp import PropsSI
import textwrap
from PIL import ImageTk, Image
from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg, NavigationToolbar2Tk)
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as img
from matplotlib.patches import Arc
from scipy.interpolate import make_interp_spline
import random as rd
import os, sys, re
from os import path
from tkinter import filedialog as fd
import pyautogui
import subprocess
import winsound
```

**Figura 1.3:** Importación de bibliotecas en Python.

Las bibliotecas Python principales que se utilizan en el diseño del componente se las describe a continuación:

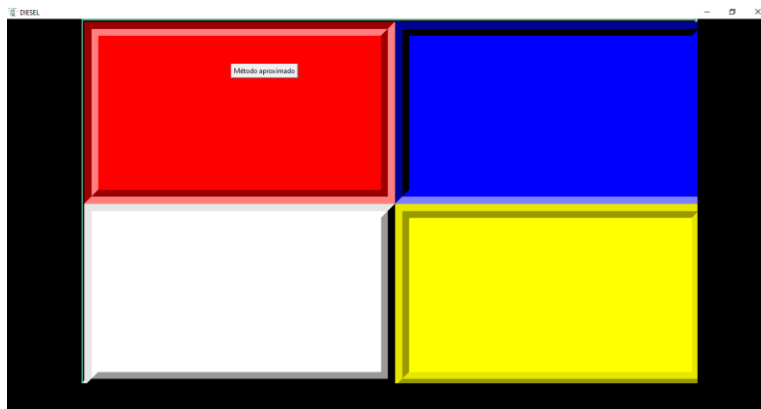
- Tkinter: utilizada para la creación de la interfaz gráfica del componente, integrada por widgets que proveen componentes para una mejora visual y una integración completa de los resultados numéricos y gráficos.
- Numpy: integrada en el componente para el uso de funciones matemáticas básicas y funciones de arreglo de tipo numérico.
- CoolProp: Provee las propiedades del fluido de trabajo para realizar los cálculos necesarios en la resolución numérica del componente.
- Matplotlib: permite realizar la representación gráfica de los resultados numéricos obtenidos, así como dota de interactividad a las gráficas con componentes de interacción directa con las mismas.
- Pillow: De uso específicamente para agregar imágenes en la interfaz gráfica, complementando la estética y uso de elementos visuales.

[6].

## Ventana Principal

Una ventana principal o raíz es el nombre que se le da a la función primaria de la biblioteca Tkinter, en ella se alberga todo tipo de componentes visuales ligados a la funcionalidad, interactividad y estética de un software

Se observa en la **Figura 1.4** un ejemplo de una Ventana raíz con algunos elementos visuales de la biblioteca Tkinter [7].



**Figura 1.4:** Ventana principal realizada con Tkinter.

Esta ventana principal es el primer elemento que se utiliza para armar la interfaz gráfica del programa, una vez iniciada su construcción se agregan las herramientas visuales llamadas “widgets” [7].

## Widgets de Tkinter

Los widgets son componentes incluidos en algunas las bibliotecas de Python tienen mucha funcionalidad relacionada con la biblioteca a la que pertenecen [6].

Los Widgets de Tkinter son herramientas visuales necesarias para conformar una interfaz gráfica, los más utilizados para la construcción de un software con interfaz gráfica son los que se muestran en la **Figura 1.5**, los mismos que se usan en el presente proyecto [8].

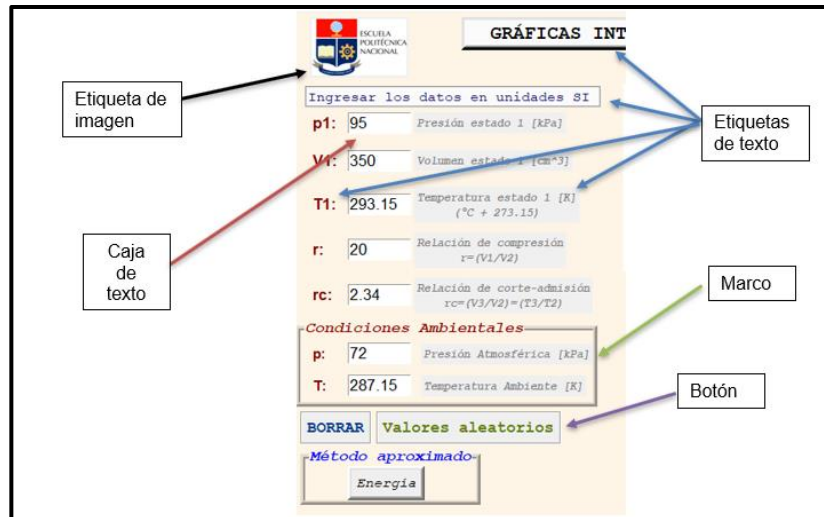


Figura 1.5: Widgets principales de Tkinter.

## Marco

Marco o “Frame” son espacios donde se pueden colocar widgets, cuenta con propiedades predeterminadas que son modificables como el tamaño, color, borde, etc. Y pueden ser colocadas en cualquier punto de la ventana principal [8].

## Etiqueta

Las etiquetas conocidas como “Label”, permiten mostrar textos o imágenes al usuario, como texto es posible definir el tamaño de letra, color y fuente a mostrar, como imagen muestra en pantalla una imagen definida anteriormente con ayuda de la biblioteca Pillow [8].

## Caja de texto

Para un software que requiere de datos proporcionados por el usuario es importante utilizar las cajas de texto “Entry”, permiten ingresar caracteres para su posterior almacenamiento, cabe mencionar que cualquier carácter ingresado en ellas es leído de forma predeterminada como una cadena de texto; para este caso particular se realiza una conversión de los datos a tipo número flotante antes de ser almacenados [8].

## **Botón**

Botón "Button" es un objeto de tipo interactivo que llama a una función definida en la programación, se activa con un "clic" por parte del usuario, cuenta con personalización de tamaño, color, borde, etc. Y se puede mostrar con texto o imagen [8].

## 2 METODOLOGÍA

El componente del proyecto se realiza conforme a la **Figura 2.1**, en el cual se detalla las actividades necesarias para cumplir con los objetivos planteados y obtener los resultados esperados.

Para realizar el componente se establecen tres fases: recopilación base teórica, diseño del software y diseño del manual de usuario, tomando en cuenta actividades específicas para cumplir con cada fase.

Inicialmente, se realiza la recopilación de la base teórica en la que se encuentra fundamentada los cálculos de cada una de las propiedades y estados necesarios del ciclo Diesel, también se establece la resolución del problema de forma numérica y gráfica la cual se utiliza posteriormente en la fase de diseño y se la desarrolla en la plataforma Python, del mismo modo se recopila información necesaria para el uso de la herramienta Python para la programación del software y se define las bibliotecas y componentes de Python más importantes para el diseño del software.

Una vez cumplida la fase de investigación previa se procede con la fase de diseño del software, en la que se implementa la resolución del problema con la plataforma Python, se realiza la construcción de todo el código del programa repasando cada módulo que lo compone hasta obtener el producto esperado.

Finalmente, se realiza un manual de usuario que contiene información y detalles del funcionamiento del software con el objetivo de mejorar la interpretación de resultados gráficos y el uso general del programa a los usuarios.



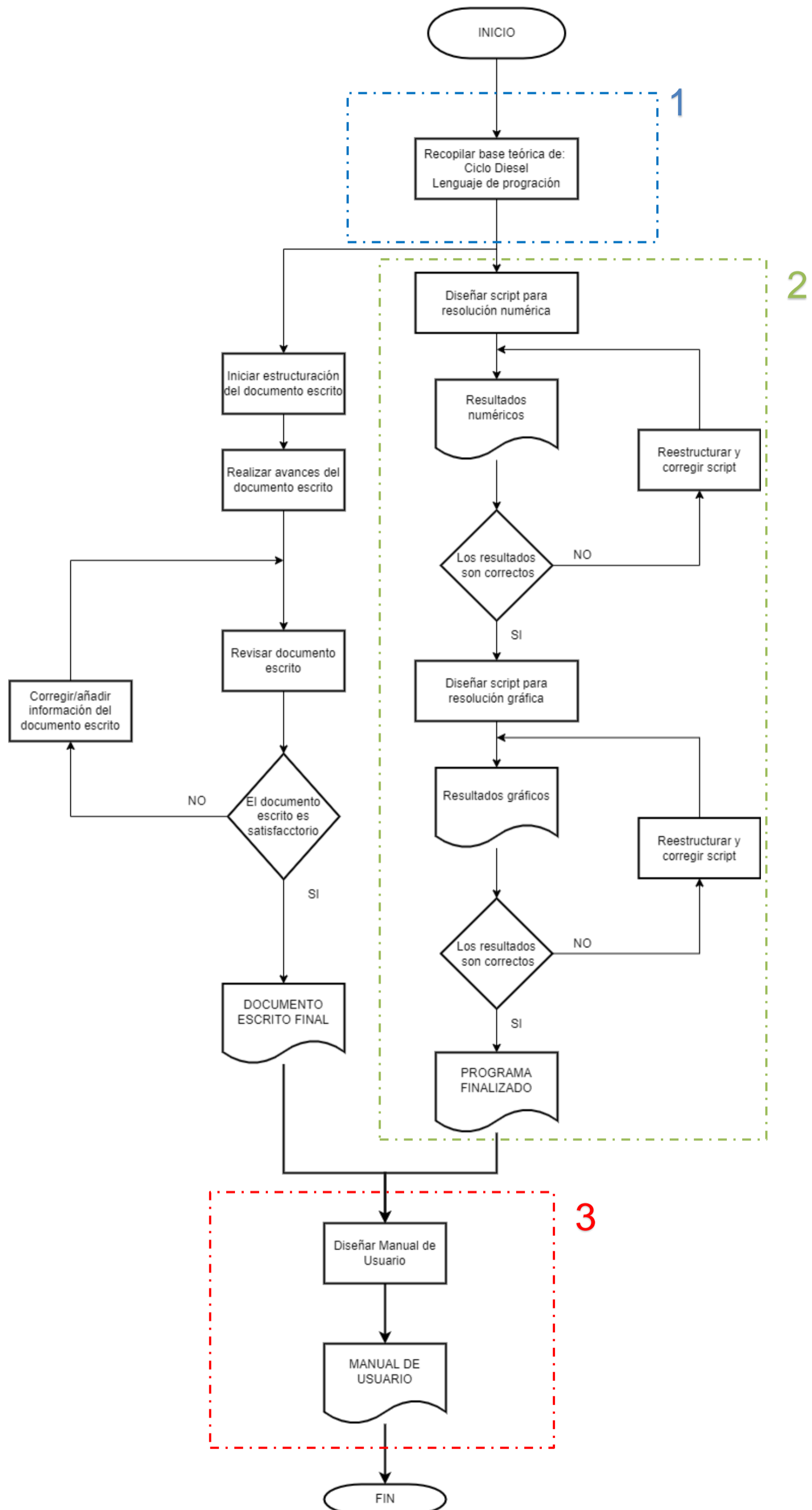


Figura 2.1: Metodología.

## 2.1 Recopilación de base teórica

Esta fase de investigación y recopilación se relaciona completamente con lo presentado en el capítulo anterior en la sección de Marco teórico, con toda la información recopilada se establece la resolución del problema para aplicarla posteriormente a la fase de diseño.

### Métodos de cálculo para resolución del ciclo Diesel

Para establecer los métodos de cálculo se debe tomar en cuenta el uso del fluido de trabajo como aire-gas ideal con las suposiciones en base a la termodinámica clásica.

#### Método aproximado

En el método aproximado se utilizan las ecuaciones de aire gas ideal, relación de compresión, relación de corte-admisión, relaciones de proceso isentrópico con  $C_p$ ,  $C_v$  y  $k$  constantes, las relaciones de calor de entrada y rechazo obtenidas del balance de energía y la relación para la eficiencia del ciclo Diesel.

Consideraciones del ciclo Diesel.

$$p_2 = p_3, V_1 = V_4, m = cte, C_v = 0.718, C_p = 1.005, k = 1.4$$

#### Método exacto

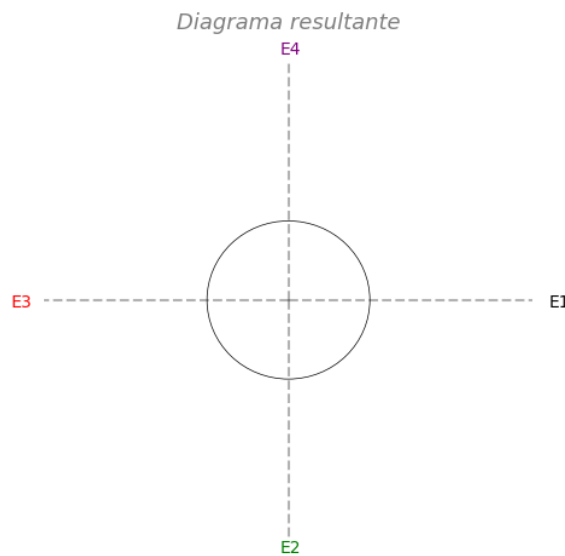
En el método exacto se utilizan las ecuaciones de aire ideal, relación de compresión, relación de corte-admisión, las relaciones de calor de entrada y rechazo obtenidas del balance de energía, la relación para la eficiencia del ciclo Diesel, la ecuación para el cálculo de la exergía sin flujo y los datos experimentales de las propiedades de aire gas ideal de la biblioteca CoolProp de Python.

Consideraciones del ciclo Diesel.

$$p_2 = p_3, V_1 = V_4, m = cte$$

## Creación de los diagramas de energía, entropía, exergía y su respectiva interpretación

La resolución gráfica que se plantea para la energía, entropía y exergía, no sigue un modelo establecido anteriormente, son diagramas diseñados por los autores de los componentes del presente Trabajo de Integración Curricular en conjunto con el director del trabajo Ing. Miguel Ortega; estos diagramas muestran la variación de la energía interna, entropía y exergía a través del ciclo Diesel, cuentan con una circunferencia centrada como base que representa el valor referencial de cada diagrama (**Figura 2.2**), la variación en el tiempo de una propiedad desde un estado hasta el siguiente se expresa de forma angular mediante arcos variables, los radios variables que se observan conformando dichos arcos corresponden a la variación de las propiedades en cada punto del ciclo; cada uno de estos puntos utilizados para realizar los diagramas resultantes se encuentran mediante una expresión de linealización con el uso de un ángulo beta que varía en el rango de 0° a 360°, rango propiamente de los diagramas creados.



**Figura 2.2** Diagrama base

Linealización para el primer cuadrante, beta de 0° a 90°, E4-E1, **Ecuación 2.1:**

$$x_n = \left( x_{E1} + (x_{E4} - x_{E1}) \cdot \left( \frac{beta}{90^\circ} \right) \right)$$

**Ecuación 2.1:** Linealización para el primer cuadrante.

Linealización para el segundo cuadrante, beta de 90° a 180°, E3-E4, **Ecuación 2.2:**

$$x_n = \left( x_{E4} + (x_{E3} - x_{E4}) \cdot \left( \frac{beta - 90^\circ}{90^\circ} \right) \right)$$

**Ecuación 2.2:** Linealización para el segundo cuadrante.

Linealización para el tercer cuadrante, beta de 180° a 270°, E2-E3, **Ecuación 2.3:**

$$x_n = \left( x_{E3} + (x_{E2} - x_{E3}) \cdot \left( \frac{beta - 180^\circ}{90^\circ} \right) \right)$$

**Ecuación 2.3:** Linealización para el tercer cuadrante.

Linealización para el cuarto cuadrante, beta de 270° a 360°, E1-E2, **Ecuación 2.4:**

$$x_n = \left( x_{E2} + (x_{E1} - x_{E2}) \cdot \left( \frac{beta - 270^\circ}{90^\circ} \right) \right)$$

**Ecuación 2.4:** Linealización para el cuarto cuadrante.

Donde:

$x$ : Representa la propiedad con la que se está trabajando.

$x_n$ : Es el valor numérico de la propiedad expresada en cualquier punto del ciclo.

Sea  $n$ :

$$n \in \mathbb{Z} / 1 \leq n \leq 100$$

Esto significa que se realiza el cálculo de 100 puntos en las representaciones gráficas, por lo tanto, la variación del ángulo beta entre cada punto es de 3.6°.

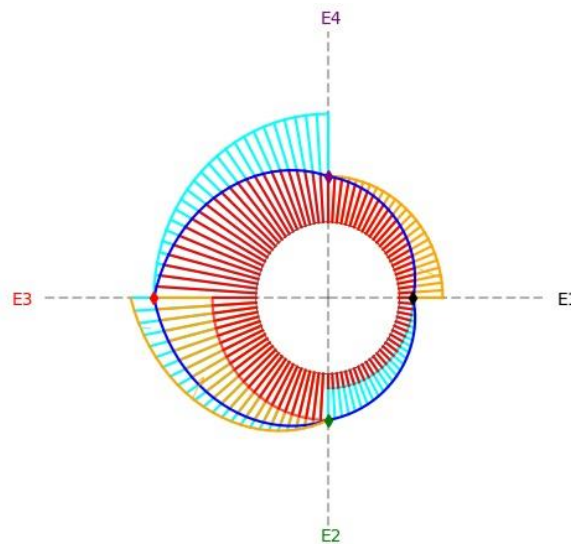
$beta$ : Representa el ángulo de apoyo para la variación de las propiedades a través del diagrama.

Nota: La linealización que se utiliza para calcular las propiedades en cada punto del ciclo

es el resultado de aplicación de teoremas geométricos en una representación gráfica con coordenadas polares.

### Interpretación diagrama de energía

En la **Figura 2.3** se muestra la variación de energía interna del ciclo Diesel y las propiedades involucradas en cada proceso para obtener dicha variación. El diagrama utiliza diferentes colores para representar estas propiedades, que son: el trabajo, el calor y la energía interna; los puntos marcados en cada eje en forma de rombo corresponden a los valores resultantes de energía interna ( $u$ ) en cada estado del ciclo ( $E_1, E_2, E_3, E_4$ ).



**Figura 2.3:** Diagrama de energía.

- Las líneas de color "celeste": Expresan el trabajo de entrada o salida en cada proceso del ciclo.
- Las líneas de color "naranja": Expresan la entrada o rechazo de calor en cada proceso del ciclo.
- Las líneas de color "rojo": Expresan la energía interna de cada proceso, sin tomar en cuenta el aumento de trabajo o calor.
- La línea de color "negro": Indica la variación de energía interna del ciclo completo a través de todos los procesos involucrados en el ciclo Diesel.

- La circunferencia central del diagrama representa el valor referencial del diagrama, en este caso será  $u_0=0$ .

De E1 a E2 se incrementa la energía interna, esto se debe a la entrada de trabajo en el proceso de compresión, por lo tanto, el trabajo se encuentra dentro del contorno de variación, ya que, en este caso la variación de energía es igual al trabajo que entra (**Ecuación 2.5**).

$$\Delta u_{12} = w_{12}$$

**Ecuación 2.5:** Variación de energía interna de E1 a E2.

De E2 a E3 ocurre un incremento de la energía interna debido al calor de entrada, pero también existe una salida de trabajo por el aumento del volumen a presión constante, por lo tanto, en este caso la variación de la energía interna será igual al calor añadido menos el trabajo de 2 a 3 (**Ecuación 2.6**).

$$\Delta u_{23} = q_a - w_{23}$$

**Ecuación 2.6:** Variación de energía interna de E2 a E3.

De E3 a E4 la energía interna disminuye debido a la salida de trabajo en el proceso de expansión isentrópica, por lo tanto, el trabajo se encuentra fuera del contorno que indica la variación, ya que, la variación de la energía interna será igual al trabajo de salida y su valor es negativo (**Ecuación 2.7**).

$$\Delta u_{34} = -w_{34}$$

**Ecuación 2.7:** Variación de energía interna de E3 a E4.

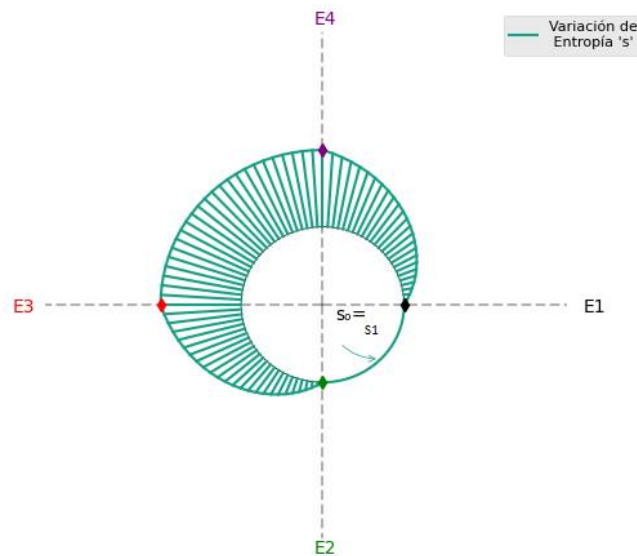
Finalmente, de E4 a E1 la energía interna disminuye debido al rechazo de calor a volumen constante, por lo tanto, la variación de energía interna es igual al calor rechazado y su valor es negativo, por eso el calor rechazado se encuentra fuera del contorno que indica la variación (**Ecuación 2.8**).

$$\Delta u_{41} = q_r \Rightarrow \Delta u_{41} = -q_{41}$$

**Ecuación 2.8:** Variación de energía interna de E4 a E1.

## Interpretación diagrama de entropía

En la **Figura 2.4** se muestra la variación de entropía del ciclo Diesel, los puntos marcados en cada eje en forma de rombo corresponden a los valores resultantes de entropía ( $s$ ) en cada estado del ciclo (E1, E2, E3, E4). La interpretación de este diagrama resulta más simple que la interpretación anterior, ya que solo cuenta con dicha propiedad representada en un mismo color (verde huevo de pato).



**Figura 2.4:** Diagrama de entropía.

Inicialmente se debe indicar que el valor de la circunferencia central referencial en este caso ya no representa un valor de  $s_0=0$ , sino es el valor de entropía más pequeño del ciclo que corresponde a “ $s_1$ ”, esto con el fin de observar con mayor claridad la variación de la entropía en el ciclo

De E1 a E2 la entropía se mantiene constante, ya que ocurre el proceso de compresión isentrópica y en este caso el arco entre ambos estados se mantiene constante en todo el cuadrante.

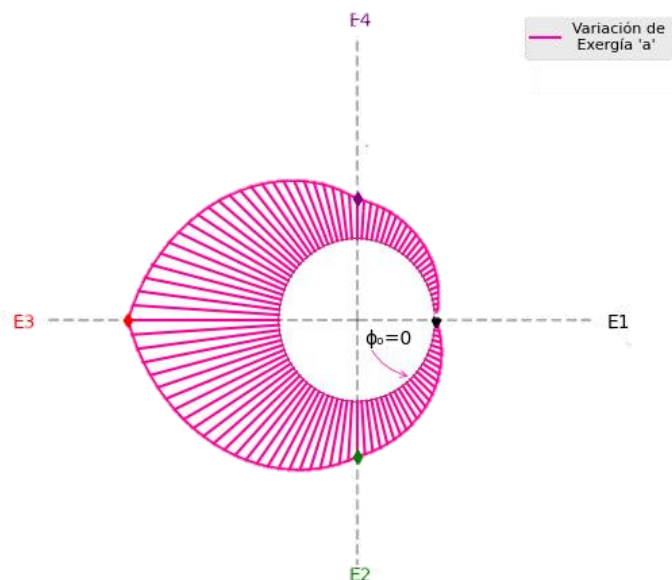
De E2 a E3 la entropía aumenta su valor por la entrada de calor en el ciclo a presión constante y se observa el aumento en el arco variable que conecta estos dos estados.

De E3 a E4 la entropía se mantiene constante, ya que ocurre el proceso de expansión isentrópica y en este caso el arco que une estos estados se mantiene constante en todo el cuadrante.

Por último, de E4 a E1 debido al rechazo de calor del ciclo, disminuye la entropía regresando al valor inicial el cual se toma como referencia en este diagrama.

### Interpretación diagrama de exergía

En la **Figura 2.5** mostrada a continuación se observa la variación de exergía (sin flujo) del ciclo Diesel, los puntos marcados en cada eje en forma de rombo corresponden a los valores resultantes de exergía ( $\phi$ ) en cada estado del ciclo (E1, E2, E3, E4). Este diagrama al igual que el de entropía cuenta solo con una propiedad en este caso exergía representada en un mismo color (magenta).



**Figura 2.5:** Diagrama de exergía.

De E1 a E2 la exergía aumenta debido al proceso de compresión isentrópica que ocurre entre estos estados.

De E2 a E3 la exergía se incrementa por la entrada de calor en el ciclo a presión constante y se observa el aumento en el arco variable que conecta estos dos estados.

De E3 a E4 existe una disminución de la exergía por el proceso de expansión isentrópica el cual tiene un trabajo de salida.

Por último, de E4 a E1 debido al rechazo de calor del ciclo la exergía disminuye. En este proceso existe destrucción de exergía.

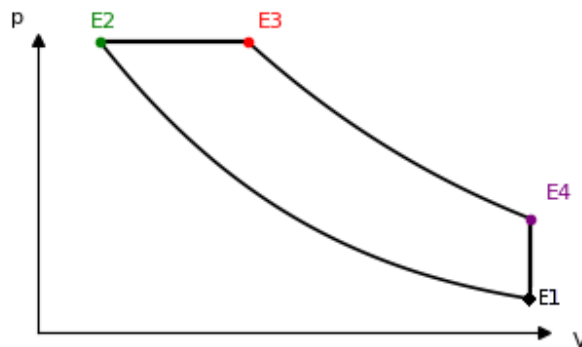


## 2.2 Diseño del software

### Resolución numérica del ciclo Diesel utilizando Python

#### Datos de entrada

Los datos de ingreso requeridos para el planteamiento y resolución del ejercicio están establecidos utilizando la **Figura 2.6** como una referencia para identificar los estados y sus unidades conforme al Sistema Internacional de Unidades, a excepción del volumen. Estos datos iniciales son requeridos al usuario para iniciar el script de resolución numérica y pueden ser ingresados como enteros o flotantes, el formato de escritura de los datos iniciales utiliza el punto (.) como separador de decimales, también se debe tener en cuenta que ninguno de los datos puede ser menor a cero y en el caso de la temperatura mínima  $T_1$  se recomienda usar valores mayores a 273.15 K.



**Figura 2.6:** Diagrama p-V referencial.

**Relación de compresión:**

$$r > 1$$

**Relación de corte-admisión:**

$$r_c > 1$$

**Estado 1:**

$p_1$ : en kilopascales [kPa].

V1: en centímetros cúbicos [cm<sup>3</sup>]

T1: en Kelvin [K]

### **Condiciones ambientales:**

p atmosférica: en kilopascales [kPa].

T ambiente: en Kelvin [K]

## **CoolProp**

La biblioteca CoolProp nos permite obtener los valores de propiedades termodinámicas de varios fluidos, en este caso se realiza la búsqueda de propiedades del aire gas-ideal para los diferentes estados del ciclo Diesel.

Primero se deben tener en cuenta algunas consideraciones:

- Identificar las propiedades termodinámicas que CoolProp proporciona y sus unidades.
- Verificar las unidades de los valores de entrada al utilizar CoolProp.
- Para encontrar una propiedad termodinámica en cualquier estado, se debe ingresar 2 propiedades diferentes que dependan del mismo estado termodinámico en que se busca la propiedad.

Teniendo en cuenta estas consideraciones no habrá ningún inconveniente al utilizar esta biblioteca.

Para comenzar con la búsqueda de propiedades se realiza un llamado a la biblioteca colocando su respectivo alias y añadiendo su principal función que nos permite obtener los resultados, como se muestra en la **Figura 2.7**.

```
import CoolProp.CoolProp as cp
from CoolProp.CoolProp import PropsSI
```

**Figura 2.7:** Importación biblioteca CoolProp.

Para obtener los resultados se define una variable que contenga la función “PropsSI”, dentro de esta función como primer parámetro se coloca entre comillas la letra que representa a la propiedad que se busca, como segundo y cuarto parámetro se colocan las letras que representan las propiedades que definen el estado termodinámico, como tercer y quinto parámetro se coloca los valores numéricos de las propiedades escogidas para definir el estado en su respectivo orden, finalmente como sexto parámetro se coloca entre comillas el nombre del fluido deseado. La **Figura 2.8** muestra un ejemplo del uso de CoolProp para encontrar la entropía y la densidad del aire gas-ideal en el estado uno (E1) del ciclo Diesel.

```
S1=PropsSI('S','T',T1i,'P',P1o,'Air')
D1=PropsSI('D','T',T1i,'P',P1o,'Air')
```

**Figura 2.8:** Ejemplo de uso de CoolProp.

Nota: La lista de letras que representan las propiedades del fluido puede ser consultada en la documentación oficial de CoolProp al igual que la lista de fluidos incluidos en la biblioteca.

### Resolución con el método aproximado utilizando Python

La resolución del método aproximado se realiza con el uso de las ecuaciones especificadas en Método aproximado.

Inicialmente se definen las constantes de Cp, Cv y k, posteriormente se establecen las consideraciones y relaciones necesarias como se muestra en la **Figura 2.9**.

```
367 k=1.4; 382 qr=cv*(T4i-T1i)
368 cp=1.005; 383 u1=cv*(T1i)
369 cv=0.718; 384 u2=cv*(T2i)
370 385 u3=cv*(T3i)
371 386 u4=cv*(T4i)
372 V4i=V1i 387 u12=cv*(T2i-T1i)
373 P2i=(P1i*((ri)**(k))) #Ec. 388 u23=cv*(T3i-T2i)
374 T2i=((P2i*T1i)/(P1i*ri)) 389 u34=cv*(T4i-T3i)
375 T3i=(T2i*rci) 390 u41=-qr
376 P3i=P2i 391
377 V2i=(V1i/ri) 392 uT=abs(u12+u23+u34+u41)
378 V3i=rci*V2i 393 uTi=round(uT,4)
379 P4i=(P3i*((rci/ri)**(k))) 394 η=(1-(qr/qa))*100
380 T4i=((ri*P4i*T2i)/(P2i))
381 qa=cp*(T3i-T2i)
```

**Figura 2.9:** Código de resolución numérica con método aproximado.

## Resolución con el método exacto utilizando Python

La resolución del método exacto se realiza con el uso de las ecuaciones especificadas en Método exacto.

En este método de resolución se utiliza la biblioteca CoolProp para obtener los valores de las propiedades que definen los estados del ciclo, además se establecen las consideraciones del ciclo y las ecuaciones para resolver completamente el problema, las figuras: **Figura 2.10**, **Figura 2.11** y **Figura 2.12** muestran el código de la resolución en Python con el método exacto.

```
P1o=P1i*1000
S1=PropsSI('S','T',T1i,'P',P1o,'Air')
D1=PropsSI('D','T',T1i,'P',P1o,'Air')
V4i=V1i
V2i=(V1i/ri)
S2=S1 #proceso isentrópico de compresión
D2=D1*ri
P2o=PropsSI('P','D',D2,'S',S2,'Air')
P2i=P2o/1000
T2i=((P2i*T1i)/(P1i*ri))
T3i=(T2i*rci)
P3o=P2o
V3i=(rci*V2i)
S3=PropsSI('S','T',T3i,'P',P3o,'Air')
S4=S3 #proceso isentrópico de expansión
D4=D1
T4i=PropsSI('T','S',S4,'D',D4,'Air')
P4o=PropsSI('P','S',S4,'D',D4,'Air')
```

**Figura 2.10:** Código de resolución numérica con método exacto (parte 1).

```
P3i=P3o/1000
P4i=P4o/1000

S1i=S2/1000
S2i=S2/1000
S3i=S3/1000
S4i=S4/1000

h1= PropsSI('H','P',P1o,'T',T1i,'Air')*(1/1000)
h2= PropsSI('H','P',P2o,'T',T2i,'Air')*(1/1000)
h3= PropsSI('H','P',P3o,'T',T3i,'Air')*(1/1000)
h4= PropsSI('H','P',P4o,'T',T4i,'Air')*(1/1000)

u1= PropsSI('U','P',P1o,'T',T1i,'Air')*(1/1000)
u2= PropsSI('U','P',P2o,'T',T2i,'Air')*(1/1000)
u3= PropsSI('U','P',P3o,'T',T3i,'Air')*(1/1000)
u4= PropsSI('U','P',P4o,'T',T4i,'Air')*(1/1000)

qa=h3-h2
qr=u4-u1

u12=u2-u1
u23=u3-u2
u34=u4-u3
u41=u1-u4

uT=abs(u12+u23+u34+u41)
uTi=round(uT,4)
eta=(1-(qr/qa))*100
```

**Figura 2.11:** Código de resolución numérica con método exacto (parte 2).

```

p0i=p0*1000

S0=PropsSI('S','P',p0i,'T',T0,'Air')
D0=PropsSI('D','P',p0i,'T',T0,'Air')
u0=PropsSI('U','P',p0i,'T',T0,'Air')
v0=1/D0
D1=PropsSI('D','T',T1i,'P',P1o,'Air')
v1ex=1/D1
v4ex=v1ex
v2ex=(v1ex/ri)
v3ex=(rci*v2ex)

S0i=S0/1000
u0i=u0/1000

phi1=((u1-u0i)+(p0*(v1ex-v0))-(T0*(S1i-S0i)))
phi2=((u2-u0i)+(p0*(v2ex-v0))-(T0*(S2i-S0i)))
phi3=((u3-u0i)+(p0*(v3ex-v0))-(T0*(S3i-S0i)))
phi4=((u4-u0i)+(p0*(v4ex-v0))-(T0*(S4i-S0i)))

```

**Figura 2.12:** Código de resolución numérica con método exacto (parte 3).

## Resolución gráfica del ciclo Diesel utilizando Python

La resolución gráfica es realizada utilizando la biblioteca Matplotlib, su uso se detalla a continuación.

### Matplotlib

La biblioteca Matplotlib permite realizar gráficos y dibujar figuras preestablecidas, tiene varias características, herramientas y parámetros que se pueden modificar y añadir, para un mejor resultado gráfico; inicialmente se define una figura con sus dimensiones y los puntos por pulgada (dpi) de resolución y en este caso particularmente se usa el método “subplot” para poder agregar varios elementos en un mismo gráfico. En la **Figura 2.13** se muestra un ejemplo.

```

figura=plt.Figure(figsize=(5.75,5.60),dpi=100)
ax=figura.add_subplot()
ax.axis('off')

```

**Figura 2.13:** Ejemplo de construcción de gráficos con Matplotlib.

Se puede graficar rectas estableciendo puntos como se muestra en la **Figura 2.14**, graficar figuras predefinidas como una circunferencia indicando las coordenadas y el radio como se observa en la **Figura 2.15**, arcos variables como se explicará más adelante, funciones trigonométricas, etc.

```
x1,y1=[150,300],[150,150]
x2,y2=[150,150],[150,0]
x3,y3=[150,-50],[150,150]
x4,y4=[150,150],[150,325]
ax.plot(x1,y1,color='black',ls='--',alpha=0.3)
ax.plot(x2,y2,color='black',ls='--',alpha=0.3)
ax.plot(x3,y3,color='black',ls='--',alpha=0.3)
ax.plot(x4,y4,color='black',ls='--',alpha=0.3)
```

**Figura 2.14:** Código para graficar rectas con dos puntos.

```
draw_circle = plt.Circle((150, 150), 50,fill=False,color='black')
ax.add_artist(draw_circle)
```

**Figura 2.15:** Código para graficar circunferencia.

En la **Figura 2.16** se muestra el comando para definir un título a la gráfica.

```
ax.set_title('Diagrama de Energía',style='italic',fontsize='13',color='red')
```

**Figura 2.16:** Comando para definir el título de un gráfico.

Los gráficos implementados utilizan coordenadas polares, por lo cual se inicia el código fijando los valores iniciales del gráfico, con el ángulo  $\beta=0$ , se define una escala para los valores resultantes en este caso “dr” para que no sobrepase el tamaño de la figura, también se crean listas vacías de datos como una herramienta de ayuda para almacenamiento, se puede observar un ejemplo en la **Figura 2.17**.

```
beta=0
dr=0.045
pi=np.pi
lx1=[]
ly1=[]
lx2=[]
ly2=[]
lx3=[]
ly3=[]
lx4=[]
ly4=[]
lx5=[]
ly5=[]
```

**Figura 2.17:** Ejemplo definición de variables iniciales para graficar.

En este caso se establece el número de puntos que se desea graficar en 100 para poder observar de forma clara las variaciones de las propiedades, para obtener estos puntos se utiliza un ciclo “for”, como se puede observar en la **Figura 2.18**, con las sentencias condicionales para cada cuadrante.

```

for i in range(102):
    if beta>=0 and beta<(0.5*pi):

```

**Figura 2.18:** Ejemplo Ciclo For para encontrar los puntos del gráfico.

Los gráficos en Matplotlib cuentan con coordenadas rectangulares, por lo tanto, en este caso se debe realizar una conversión de coordenadas polares a rectangulares; tomando en cuenta la sección Creación de los diagramas de energía, entropía, exergía y su respectiva interpretación, se definen las ecuaciones de linealización para cada cuadrante y la conversión de resultados. En la **Figura 2.19** podemos ver el ejemplo de la programación para graficar los puntos y radios en el cuarto cuadrante.

```

elif beta>=(1.5*pi):

    u=(u2+(u1-u2)*((beta-(1.5*pi))/(0.5*pi))) *dr
    uz=u1*dr
    x=((50+u)*(np.cos(beta)))+150
    y=((50+u)*(np.sin(beta)))+150

    xz=((50+uz)*(np.cos(beta)))+150
    yz=((50+uz)*(np.sin(beta)))+150

    xa=(50*(np.cos(beta)))+150
    ya=(50*(np.sin(beta)))+150
    xf,yf=[xa,x],[ya,y]
    xfz,yfz=[xa,xz],[ya,yz]
    ax.plot(xf,yf,color='cyan',ls='-',alpha=1)
    ax.plot(xfz,yfz,color='red',ls='-',alpha=1)
    xc=None
    yc=None
    xcr=None
    ycr=None
    xu=None
    yu=None

```

**Figura 2.19:** Ejemplo de cálculo de los puntos del cuarto cuadrante para el gráfico de energía.

Gracias al ciclo “for” y la función “append” se agregan todos los datos encontrados a las listas vacías (**Figura 2.20**), para poder finalmente graficar los arcos variables uniendo todos estos puntos, además se realizan anotaciones con la función “annotate” y añadir etiquetas con la función “text” (**Figura 2.21**).

```

lx4.append(xcr)
ly4.append(ycr)
lx5.append(xu)
ly5.append(yu)
beta=(i)*(pi/50)

```

**Figura 2.20:** Ejemplo de almacenamiento de los puntos del gráfico.

```

ax.plot((u1*dr)+200,150,color='black',marker='d',markersize=5,alpha=1,zorder=3)
ax.plot(150,(150-(u2*dr)+50),color='green',marker='d',markersize=5,alpha=1,zorder=3)
ax.plot((150-(u3*dr)+50),150,color='red',marker='d',markersize=5,alpha=1,zorder=3)
ax.plot(150,((u4*dr)+200),color='purple',marker='d',markersize=5,alpha=1,zorder=3)
ax.text((u1*dr)+225,135,(round(u1,2)),color='black')
ax.text(155,(-(u2*dr)+75),(round(u2,2)),color='black')
ax.text((100-(u3*dr)+50),165,(round(u3,2)),color='black')
ax.text(155,((u4*dr)+225),(round(u4,2)),color='black')
ax.text(310,145,'E1',color='black')
ax.text(145,-10,'E2',color='green')
ax.text(-70,145,'E3',color='red')
ax.text(145,330,'E4',color='purple')
ax.text(155,135,'us=0',color='black')
ax.annotate(r'',color='black',bbox={'facecolor':'orange','alpha':0.5,'pad':5},
            xy=(185,115),
            xytext=(155,142),
            fontsize=9,
            arrowprops=dict(arrowstyle="->",color='red',connectionstyle="arc3,rad=.3"))
ax.annotate(r'Entrada de Trabajo',color='black',bbox={'facecolor':'cyan','alpha':0.5,'pad':5},
            xy=((150+(u1*0.75*dr)+50),110),
            xytext=(250,0.4),
            fontsize=9,
            arrowprops=dict(arrowstyle="->",color='cyan',connectionstyle="arc3,rad=.3"))

```

Figura 2.21: Ejemplo de anotaciones y etiquetas en el gráfico.

En **Figura 2.22** se muestra un ejemplo de diagrama de energía graficado con el uso de Matplotlib.

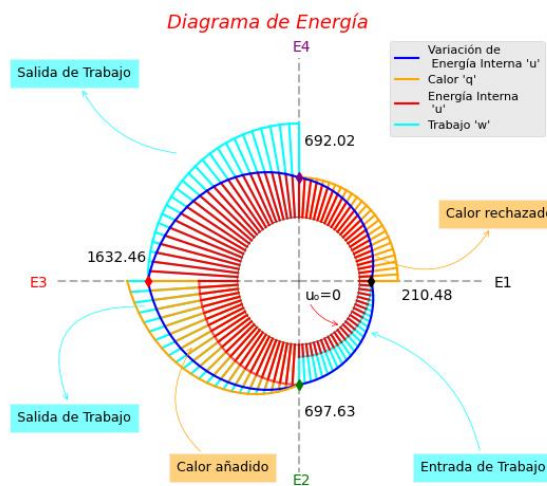


Figura 2.22: Ejemplo diagrama de energía.

## Interfaz gráfica Tkinter utilizando Python

La interfaz gráfica en un software es una parte importante si se desea brindar facilidades para el usuario al utilizarlo, además es necesaria para la interactividad de los diagramas los cuales se actualizan de forma inmediata al cambiar los datos de entrada y calcular las propiedades resultantes.

Para iniciar con la construcción de esta interfaz y añadir widgets se debe importar su respectiva biblioteca en Python.



```
import tkinter as tk
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
from tkinter import filedialog as fd
```

Figura 2.23: Importación de biblioteca Tkinter.

Como se muestra en la **Figura 2.23** adicionalmente se importan funciones de la misma biblioteca dependiendo la necesidad y se establece un alias para simplificar la sintaxis al momento de utilizar la biblioteca Tkinter en este caso el alias utilizado es "tk". Ahora se inicia con la construcción de la Ventana Principal o raíz, siendo el contenedor general de todos los complementos y herramientas que forman parte del software.

A continuación, en la **Figura 2.24** se presenta un ejemplo de la construcción de una ventana raíz con Tkinter en Python con la definición de características principales como: título, ícono, estado inicial y dimensiones.

```
#####Ventana raiz contenedora#####
principal = tk.Tk()
principal.title ("Gráficas interactivas ciclo Diesel")
principal.iconbitmap(resource_path("piston.ico"))
principal.state('normal')
principal.geometry('1315x740+0+0')
principal.resizable()
miframe = tk.Frame(principal)
miframe.pack(fill='both',expand='true')
miframe.config(bg='#FEF5E7')
```

Figura 2.24: Ejemplo de creación de ventana y marco principales.

En la misma **Figura 2.24** se muestra un ejemplo del código para crear el primer widget en la ventana raíz, en este caso un elemento de tipo Marco.

Los títulos, anotaciones y textos que se desea mostrar al usuario se insertan mediante Etiqueta, en la **Figura 2.25** se observa un ejemplo del código para colocar estas herramientas dentro del marco anteriormente creado.

```
##título y sub. Etiquetas#####
t1=Label(miframe, text="GRÁFICAS INTERACTIVAS CICLO DIESEL", fg='black', font= ('Courier new',15,'bold'),bg=
t1.grid(row=0,column=0,sticky='w',padx=180, pady=5)
st1=Label(miframe, text="Ingresar los datos en unidades SI ", fg='#191970', font= ('Courier new',11),bg='#f
st1.place(x=15,y=78)
pvacio=Label(miframe, text=" ", fg='black', font= ('Arial',12,'bold'),bg='#fbfbfb'),bg='#B0F2C2')
pvacio.grid(row=1,column=1, padx=0, pady=15,sticky='w')

resgra=tk.Label(miframe,text='',fg='black',bd=4,width=18, font= ('Courier new',12,'bold'),relief="raised")
resgra.place(x=475,y=285)
```

Figura 2.25: Código ejemplo de creación de etiquetas.

Inicialmente los datos de ingreso se recolectan con la ayuda de 2 herramientas: Caja de texto, las cuales permiten escribir cualquier tipo de carácter al usuario y la función “get” para poder obtener todo el texto escrito en las cajas.

A continuación, en la **Figura 2.26** se muestra un ejemplo de los comandos de configuración del widget “Caja de texto” y la función “get”.

```
P1=DoubleVar()  
cajaP1 = tk.Entry(miFrame, font= ('SimSum',13),width=7,textvariable=P1)  
cajaP1.grid(row=2,column=0, padx=60, pady=5,sticky='w')  
cajaP1.delete(0, tk.END)  
cajaP1.insert(0,95)  
Pli=float(P1.get())
```

**Figura 2.26:** Ejemplo de uso de caja de texto y recolección de datos.

Las funciones “delete” e “insert” permiten borrar inicialmente el contenido de la caja y añadir un dato predeterminado, usar estas funciones es opcional.

Las imágenes se insertan con la biblioteca “Pillow” y las funciones “Image” e “ImageTk”, simplemente se indica la imagen que se desea colocar y se crea una Etiqueta para colocar la imagen como se ejemplifica en la **Figura 2.27**; la imagen puede ser redimensionada antes de colocarla en el marco.

```
imagen=Image.open(resource_path('epnlogo.jpg'))  
imagen=imagen.resize((100,60))  
img=ImageTk.PhotoImage(imagen)  
logo=tk.Label(miFrame,image=img)  
logo.place(x=20,y=5)
```

**Figura 2.27:** Código para insertar imagen como etiqueta.

La definición para mensaje de error se realiza con la función “MessageBox” integrada en Tkinter, se define el mensaje superior de la ventana emergente y el mensaje de error a mostrar, en las figuras: **Figura 2.28** y **Figura 2.29** se muestra un ejemplo de la implementación de mensaje de error y la condición para ser mostrados usando un condicionante If.

```

#####Mensajes de error#####
def men1():
    messagebox.showinfo("INGRESO ERRÓNEO DE DATOS", "La relación de compresión")
def men2():
    messagebox.showinfo("INGRESO ERRÓNEO DE DATOS", "Ingresar condiciones ambientales")
def men5():
    messagebox.showinfo("INGRESO ERRÓNEO DE DATOS", "No puede ingresar datos")
def men6():
    messagebox.showinfo("INGRESO ERRÓNEO DE DATOS", "El 'Volumen 3' debe ser")
def men7():
    messagebox.showinfo("INGRESO ERRÓNEO DE DATOS", "Ingrese el valor de Presión")
def men8():
    messagebox.showinfo("INGRESO ERRÓNEO DE DATOS", "La relación de corte-admisión")

```

Figura 2.28: Definición de mensajes de error.

```

if (Pli<=0 or
Vli <=0 or
Tli <=0 or
ri <=0 or
rci <=0):
    men5()
    return
if (ri <=1):
    men1()
    return
if (rci <=1):
    men8()
    return

```

Figura 2.29: Condiciones para mostrar los mensajes de error.

El

Botón en la interfaz gráfica nos permiten llamar a diferentes funciones integradas en el programa, para permitir al usuario utilizarlas cuando las requieren. A continuación, en la **Figura 2.30** se ha colocado un ejemplo de las líneas de código para integrar un botón en el marco principal.

```

borrarb=tk.Button(miframe,text='BORRAR',
font= ('Courier new',13,'bold'),
fg='#003986',bd=2,
relief=GROOVE,
command=reset,
cursor='hand2')
borrarb.place(x=10,y=423)

```

Figura 2.30: Ejemplo de creación de un botón.

Con todos los componentes necesarios agregados solo resta indicar a la ventana raíz que no se agregará ningún componente y se debe construir usando la línea de código mostrada en la **Figura 2.31**;Error! No se encuentra el origen de la referencia..

```
3363 principal.mainloop()
```

**Figura 2.31:** Comando para finalizar la construcción de la ventana principal.

### **Integración de la interfaz gráfica con la resolución numérica y la resolución gráfica**

Una vez que se ha realizado la interfaz gráfica con los diferentes componentes necesarios para garantizar la funcionalidad y la interactividad al usuario se une la interfaz gráfica con el módulo de cálculos para resolución numérica y el módulo de resolución gráfica.

El ingreso de datos en la interfaz gráfica conforme a Datos de entrada se observa en la **Figura 2.32**.

Ingresar los datos en unidades SI	
<b>p1:</b>	92 Presión estado 1 [kPa]
<b>V1:</b>	380 Volumen estado 1 [cm <sup>3</sup> ]
<b>T1:</b>	294 Temperatura estado 1 [K] (°C + 273.15)
<b>r:</b>	18 Relación de compresión $r = (V1/V2)$
<b>rc:</b>	2.09 Relación de corte-admisión $rc = (V3/V2) = (T3/T2)$
Condiciones Ambientales	
<b>p:</b>	84 Presión Atmosférica [kPa]
<b>T:</b>	273 Temperatura Ambiente [K]

**Figura 2.32.** Pantalla de ingreso de datos.

Cabe señalar que para los cálculos que realiza el software los datos ingresados deben encontrarse en el formato propuesto en Datos de entrada, por lo que se le especifica a la caja de texto que los números ingresados se los maneje como “DoubleVar”, y la función “get” se encarga de obtener estos datos para su posterior uso como se muestra en **Figura 2.33**, si el usuario no cumple con el formato de ingreso de datos o ingresa un carácter que no sea numérico, la ejecución se interrumpe y se genera un mensaje de error.

```

Pli=float (Pl.get ())
Vli=float (Vl.get ())
Tli=float (Tl.get ())

ri=float (r.get ())
rci=float (rc.get ())
T0=float (Tamb.get ())
p0=float (patm.get ())

```

**Figura 2.33:** Código para obtener los datos ingresados.

Una vez implementado el nexo de datos ingresados por el usuario con el código general se procede a usar estos datos para ingresarlos en los módulos de resolución numérica y gráfica, para dar la posibilidad al usuario de llamar estos módulos se los define como funciones agregando únicamente el código de obtención de datos y las sentencias de tipo “if” para establecer las condiciones del ciclo Diesel, en la **Figura 2.34** se muestra un ejemplo de una parte del código de la resolución numérica utilizando el método aproximado.

```

344 def aproximado():
345     Pli=float (Pl.get ())
346     Vli=float (Vl.get ())
347     Tli=float (Tl.get ())
348     ri=float (r.get ())
349     rci=float (rc.get ())
350
351     if (Pli<=0 or
352         Vli <=0 or
353         Tli <=0 or
354         ri <=0 or
355         rci <=0):
356         men5 ()
357         return
358     if (ri <=1):
359         men1 ()
360         return
361     if (rci <=1):
362         men8 ()
363         return
364
365
366 #####Ecuaciones#####
367     k=1.4;
368     cp=1.005;

```

**Figura 2.34:** Ejemplo de construcción de función con el módulo de resolución numérica.

La resolución gráfica se encontrará dentro de esta misma función ya que depende de los resultados del cálculo numérico, pero para ser mostrada en el marco principal junto a las respuestas del ejercicio, se crea un nuevo marco y se establece un gráfico inicial que será la base para cada diferente diagrama programado, el código de ejemplo para crear este gráfico se muestra en la **Figura 2.35**.

```

grframe = tk.Frame(miframe,width="582",height="580")
grframe.place(x=730,y=0)
grframe.config(bd=3)
grframe.config(relief="groove")

figura=plt.Figure(figsize=(5.75,5.60),dpi=100)
ax=figura.add_subplot()

ax.axis('off')
Grafica_base=FigureCanvasTkAgg(figura,grframe)
Grafica_base.get_tk_widget().pack(fill='both', expand=True)

x1,y1=[150,300],[150,150]
x2,y2=[150,150],[150,0]
x3,y3=[150,0],[150,150]
x4,y4=[150,150],[150,300]
ax.plot(x1,y1,color='black',ls='--',alpha=0.3)
ax.plot(x2,y2,color='black',ls='--',alpha=0.3)
ax.plot(x3,y3,color='black',ls='--',alpha=0.3)
ax.plot(x4,y4,color='black',ls='--',alpha=0.3)

ax.set_title('Diagrama resultante',style='italic',fontsize='13',co

draw_circle = plt.Circle((150, 150), 50,fill=False,color='black')

```

Figura 2.35: Ejemplo de creación de la gráfica en la ventana principal.

A continuación, en las figuras: **Figura 2.36** y **Figura 2.37** se muestra la integración de la herramienta “Toolbar” que permite interactuar con el diagrama con funciones de acercamiento y alejamiento, mover, guardar y regresar al estado inicial del gráfico, la adición de esta herramienta es opcional, pero tiene mucha utilidad para una mejor visualización.

```

#####
class HorizontalNavigationToolbar3Tk(NavigationToolbar2Tk):
    def __init__(self, canvas, window):
        super().__init__(canvas, window, pack_toolbar=False)

        # colocar botones
        def _Button(self, text, image_file, toggle, command):
            b = super()._Button(text, image_file, toggle, command)
            b.pack(padx=3) # re-pack button in vertical direction
            return b

        # separador
        def _Spacer(self):
            s = tk.Frame(self, relief=tk.RIDGE)
            s.pack() # pack in vertical direction
            return s

        # desactivar posición del mouse
        def set_message(self, s):
            return s

```

Figura 2.36: Definición de clase para modificar características de Toolbar.

```

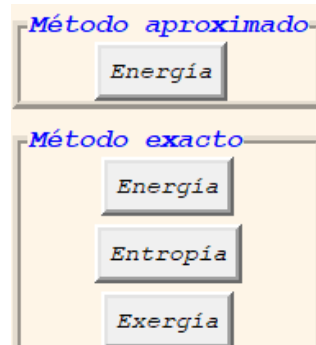
#####
toolbar = HorizontalNavigationToolbar3Tk(Grafica_base,principal)
toolbar.update()
toolbar.place(x=890,y=2)

```

Figura 2.37: Código para añadir Toolbar al diagrama final.

## Software resultante y corrección de errores

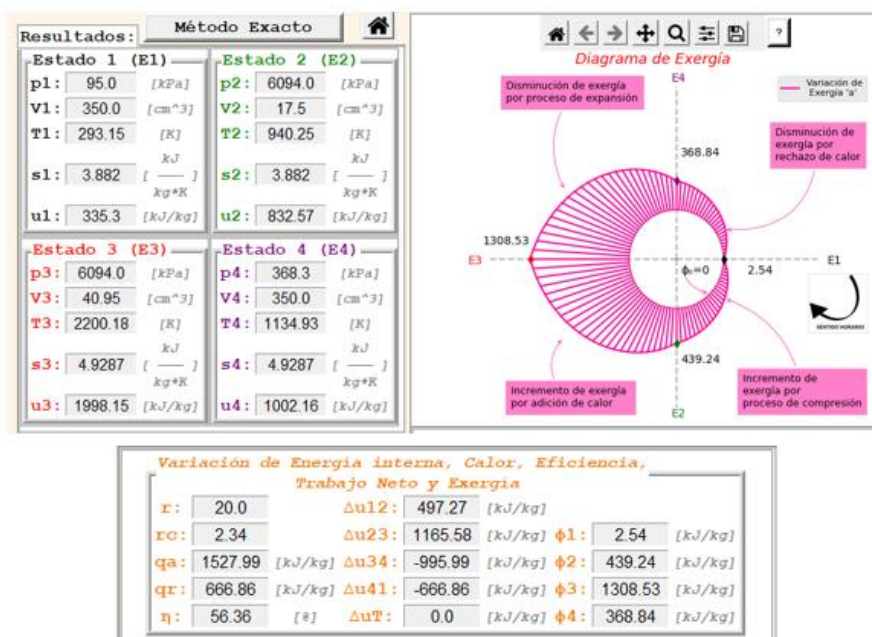
El software resultante que a su vez es un módulo adicional del software “Diagramas termodinámicos interactivos” cuenta con 4 módulos principales de cálculo asociado a los botones que se muestran en la **Figura 2.38**.



**Figura 2.38:** Botones asociados a los módulos principales.

Estos botones presentan los resultados numéricos de las propiedades del Ciclo Diesel en cada uno de sus estados y procesos calculados con los diferentes métodos de resolución: aproximado y exacto, también se presentan los diagramas interactivos de energía, entropía y exergía dependiendo del botón elegido por el usuario.

A continuación, en la **Figura 2.39** se muestra un ejemplo de obtención de resultados al elegir el botón de “Exergía”



**Figura 2.39:** Ejemplo de obtención de resultados de exergía.

Con las funciones principales implementadas, se procede a depurar el código eliminando líneas irrelevantes, espacios excesivos y líneas duplicadas.

Los errores más comunes que se han corregido son:

- Variables definidas con nombres iguales
- Falta de caracteres de agrupación “[”], “()”, “<>”, “:.”
- Espacios faltantes o excesivos
- Falta de comillas en expresiones
- Referenciar o llamar variables, funciones o clases que se han asignado posteriormente.

Adicionalmente se implementaron funciones extras para una mejor experiencia para el usuario.

- Botón de datos aleatorios para el ejercicio
- Limpieza de datos iniciales con el botón borrar
- Capturador de pantalla
- Gráficos p-V y T-s que se muestran adicionalmente con los resultados
- Atajos de teclado
- Botón de ayuda para interpretación de gráficas
- Botones de información del programa
- Botones para guardar y cargar datos iniciales.

Una vez concluida depuración del código final se inicia con la comprobación de resultados numéricos que arroja el programa lo cual se realiza en el capítulo a continuación.



## 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

### 3.1 Resultados

En este capítulo se analizan los resultados numéricos y gráficos del software y se presenta el producto final con todas sus partes.

Para realizar la comprobación de resultados numéricos del programa se ha realizado resolución de ejercicios propuestos de [1] y se ha comparado los resultados obtenidos en el software con los resultados de dicha resolución, adicionalmente se utilizó el Software TermoGraf v5.7 para una comparación adicional.

#### Análisis de error

Inicialmente se debe aclarar algunos aspectos del software para realizar el análisis de error de las propiedades resultantes.

1. El volumen que ingresa el usuario es un volumen absoluto, por lo cual no tiene relación con los estados termodinámicos, en el software se trabaja internamente con los volúmenes específicos, por lo que el dato de volumen no es relevante, sin embargo, se encuentra agregado para ilustrar al usuario como actúa la variación del volumen del fluido en un cilindro en cada estado producto de las relaciones de compresión y corte-admisión.
2. La biblioteca CoolProp trabaja con diferente estado ambiente de referencia para las propiedades de energía interna y entropía, si se hace la comparación con la tabla A-17 de [1], por lo tanto, el error se lo analiza en la variación de estas propiedades a través de los procesos con los que cuenta el ciclo.

A continuación, se muestra la comparación de resultados y el error calculado con respecto a un ejercicio de [1], utilizando los dos métodos de resolución: método aproximado en la **Tabla 3.1** y método exacto en la **Tabla 3.2**.

**Tabla 3.1:** Errores de cálculo numérico con el método aproximado.

<b>MÉTODO APROXIMADO</b>					
<b>Propiedad</b>		<b>Software</b>	<b>Teórico</b>	<b>Error</b>	<b>Error %</b>
<b>q<sub>a</sub></b>	[kJ/kg]	1234.28	1234.51	0.0001863	0.0186
<b>q<sub>r</sub></b>	[kJ/kg]	450.27	450.27	0	0
<b>n</b>	[%]	63.52	63.53	0.000157	0.0157
<b>Estado 1</b>					
<b>u1</b>	[kJ/kg]	210.48	210.48	0	0
<b>Estado 2</b>					
<b>p2</b>	[kPa]	6297.5	6297.46	0.000006	0.0006
<b>T2</b>	[K]	971.63	971.63	0	0
<b>u2</b>	[kJ/kg]	697.63	697.63	0	0
<b>Estado 3</b>					
<b>p3</b>	[kPa]	6297.5	6297.46	0.000006	0.0006
<b>T3</b>	[K]	2199.78	2200	0.000100	0.0100
<b>u3</b>	[kJ/kg]	1579.44	1579.6	0.000101	0.0101
<b>Estado 4</b>					
<b>p4</b>	[kPa]	298.2	298.23	0.000101	0.0101
<b>T4</b>	[K]	920.27	920.27	0	0
<b>u4</b>	[kJ/kg]	660.76	660.75	0.000015	0.0015

En la **Tabla 3.1** se encuentran las propiedades que se pueden comparar en el método aproximado, ya que en este método se utilizan calores específicos constantes y las relaciones isentrópicas como se indica en Método aproximado, los errores obtenidos son mínimos y en algunos casos los resultados son completamente iguales.

Estos errores corresponden al uso de un mayor número de decimales en el cálculo que realiza el software, ya que en las operaciones no existe ningún tipo de redondeo, los resultados mostrados al usuario sin embargo si se encuentran redondeados para una mejor visualización.

La cantidad más grande de error 0.0186% corresponde al Calor añadido, lo cual tiene sentido siendo que es de las propiedades que dependen de la temperatura T3 y la T2 las cuales son calculadas a partir de los datos de ingreso.

**Tabla 3.2:** Errores de cálculo numérico con el método aproximado.

<b>MÉTODO EXACTO</b>					
<b>Propiedad</b>		<b>Software</b>	<b>Teórico</b>	<b>Error</b>	<b>Error %</b>
<b>qa</b>	[kJ/kg]	1405.32	1388.71	0.0119607	1.1961
<b>qr</b>	[kJ/kg]	593.32	594.79	0.0024715	0.2471
<b>n</b>	[%]	57.78	57.17	0.0106699	1.0670
<b><math>\Delta s_{23}</math></b>	[kJ/kg·K]	0.9794	0.9806	0.0012237	0.1224
<b><math>\Delta s_{14}</math></b>	[kJ/kg·K]	-0.9794	-0.9806	0.0012237	0.1224
<b>Estado 1</b>					
<b><math>\phi_1</math></b>	[kJ/kg]	2.54	2.54	0	0
<b><math>\Delta u_{12}</math></b>	[kJ/kg]	481.19	476.335	0.0101924	1.0192
<b>Estado 2</b>					
<b>p2</b>	[kPa]	6094	5917.97	0.0297450	2.9745
<b>T2</b>	[K]	921.2	913.08	0.0088930	0.8893
<b><math>\phi_2</math></b>	[kJ/kg]	423.16	418.38	0.0114250	1.1425
<b><math>\Delta u_{23}</math></b>	[kJ/kg]	1070.3	1057.99	0.0116353	1.1635
<b>Estado 3</b>					
<b>p3</b>	[kPa]	6094	5917.97	0.0297450	2.9745
<b>T3</b>	[K]	2085.61	2067.21	0.0089009	0.8901
<b><math>\phi_3</math></b>	[kJ/kg]	1216.26	1198.42	0.0148863	1.4886
<b><math>\Delta u_{34}</math></b>	[kJ/kg]	-958.17	-939.53	0.0198397	1.9840
<b>Estado 4</b>					
<b>p4</b>	[kPa]	340.9	341.01	0.0003226	0.0323
<b>T4</b>	[K]	1050.44	1052.28	0.0017486	0.1749
<b><math>\phi_4</math></b>	[kJ/kg]	314.63	315.76	0.0035787	0.3579
<b><math>\Delta u_{41}</math></b>	[kJ/kg]	-593.32	-594.79	0.0024715	0.2471

En la **Tabla 3.2** se encuentran las propiedades que se pueden comparar en el método exacto, en este método se utilizan valores experimentales de las propiedades que dependen de los calores específicos variables, los cuales se encuentran en tablas, en el caso de los valores teóricos de referencia se utiliza la tabla A-17 de [1] y el software utiliza la base de datos de CoolProp.

Los errores obtenidos son mayores a comparación del método aproximado y esto se debe principalmente a las interpolaciones realizadas en el cálculo a mano usando las tablas, ya

que los valores obtenidos de las interpolaciones se usan con 3 decimales para posteriores cálculos y en el cálculo en el software se usan todos los decimales para encontrar los valores en la base de datos de CoolProp y los cálculos posteriores; sin embargo los errores siguen siendo mínimos en las propiedades de mayor relevancia como calores, energía, entropía y exergía.

El valor mayor de error es de 2.9745% en P2 y P3 y se debe tener en cuenta que en el software se encuentra estas propiedades con el uso del CoolProp mediante las temperaturas correspondientes a cada uno de los estados T2 y T3, a diferencia del cálculo a mano que se encuentran con la **Ecuación 1.1** y al tener también un error en las temperaturas entre el software y en el cálculo a mano se espera siempre un error como el obtenido.

En ambos casos se obtienen errores mínimos en los resultados por lo que se comprueba el funcionamiento de los módulos de cálculo en el software siendo satisfactorios los resultados expuestos.

### **Sensibilidad de variables**

El análisis de la sensibilidad de variables en el software se ha enfocado en las variables con mayor repercusión en los resultados numéricos y gráficos, en ambos casos de resolución: método aproximado y método exacto, se ha determinado que las variables que más sensibilidad tienen al momento de realizar variaciones son la relación de compresión ( $r$ ) y la relación de corte admisión ( $r_c$ ).

### **Análisis de sensibilidad en el método aproximado**

Los resultados de energía interna, calor añadido y calor rechazado en el método aproximado dependen completamente de las temperaturas de cada estado, el cálculo de las temperaturas involucra el uso de la relación de compresión y la relación de corte-admisión, por lo tanto, son las variables con mayor repercusión en los resultados de este método.

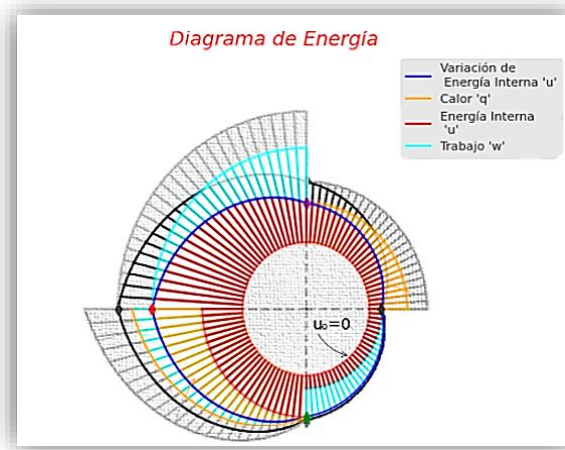
A continuación, en **Tabla 3.3** se muestra la tendencia del comportamiento de las variables.

**Tabla 3.3:** Análisis sensibilidad método aproximado.

Análisis sensibilidad método aproximado							
	u1	u2	u3	u4	qa	qr	n
	[kJ/kg]	[kJ/kg]	[kJ/kg]	[kJ/kg]	[kJ/kg]	[kJ/kg]	[%]
r: aumenta, rc: constante	-	aumenta	aumenta	-	aumenta	-	aumenta
r: disminuye, rc: constante	-	disminuye	disminuye	-	disminuye	-	disminuye
r: constante, rc: aumenta	-	-	aumenta	aumenta	aumenta	aumenta	disminuye
r: constante, rc: disminuye	-	-	disminuye	disminuye	disminuye	disminuye	aumenta

La tendencia que se encuentra es correcta teniendo en cuenta que las propiedades analizadas dependen de las temperaturas y mientras la temperatura de cada estado aumenta, las propiedades relacionadas a estos estados también aumentan y de la misma forma ocurre con la disminución, a excepción del caso del rendimiento con la relación de corte-admisión que disminuye si la otra propiedad aumenta.

El rendimiento presenta una tendencia de aumento cuando se aumenta la relación de compresión, pero cuando la relación de corte-admisión aumenta en cambio el rendimiento tiende a disminuir, esto ocurre porque al aumentar la relación de corte admisión se aumenta el calor de entrada y el calor rechazado haciendo que el ciclo pierda rendimiento térmico, y en el caso de solo aumentar la relación de compresión solo se aumenta la entrada de calor pero el rechazo de calor se mantiene constante, lo que implicaría teóricamente una mejoría del rendimiento.



**Figura 3.1:** Comparación del diagrama de energía método aproximado.

En la **Figura 3.1** se muestra la diferencia de un diagrama de energía con valores de relación de compresión ( $r$ ) = 20 y relación de corte-admisión ( $rc$ ) = 2.264 inicialmente con sus respectivos colores, debajo de ella en tonos grises se encuentra el diagrama de energía con valores de relación de compresión ( $r$ ) = 24 y relación de corte-admisión ( $rc$ ) = 3, para ilustrar en un solo gráfico la diferencia al aumentar estas variables.

### Análisis de sensibilidad en el método exacto

De la misma forma en este método las variables que más influyen en la variación de resultados son la relación de compresión y la relación de corte admisión, por lo cual se realiza el análisis en torno a estas variables en las tablas **Tabla 3.4** y **Tabla 3.5**, presentadas a continuación.

**Tabla 3.4:** Análisis sensibilidad método exacto.

Análisis sensibilidad método exacto							
	u1	u2	u3	u4	qa	qr	n
	[kJ/kg]	[kJ/kg]	[kJ/kg]	[kJ/kg]	[kJ/kg]	[kJ/kg]	[%]
<b>r: aumenta, rc: constante</b>	-	aumenta	aumenta	aumenta	aumenta	aumenta	aumenta
<b>r: disminuye, rc: constante</b>	-	disminuye	disminuye	disminuye	disminuye	disminuye	disminuye

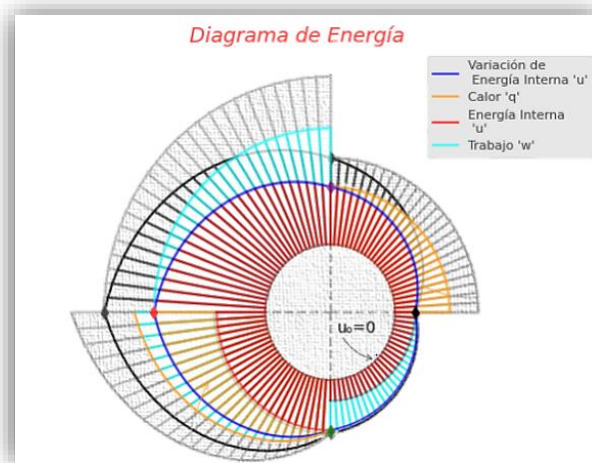
<b>r: constante, rc: aumenta</b>	-	-	aumenta	aumenta	aumenta	aumenta	disminuye
<b>r: constante, rc: disminuye</b>	-	-	disminuye	disminuye	disminuye	disminuye	aumenta

**Tabla 3.5:** Análisis sensibilidad método exacto (continuación).

<b>Análisis sensibilidad método exacto</b>						
	$\phi 1$	$\phi 2$	$\phi 3$	$\phi 4$	$s1=s2$	$s3=s4$
	[kJ/kg]	[kJ/kg]	[kJ/kg]	[kJ/kg]	[kJ/kg·K]	[kJ/kg·K]
<b>r: aumenta, rc: constante</b>	-	aumenta	aumenta	aumenta	-	aumenta
<b>r: disminuye, rc: constante</b>	-	disminuye	disminuye	disminuye	-	disminuye
<b>r: constante, rc: aumenta</b>	-	-	aumenta	aumenta	-	aumenta
<b>r: constante, rc: disminuye</b>	-	-	disminuye	disminuye	-	disminuye

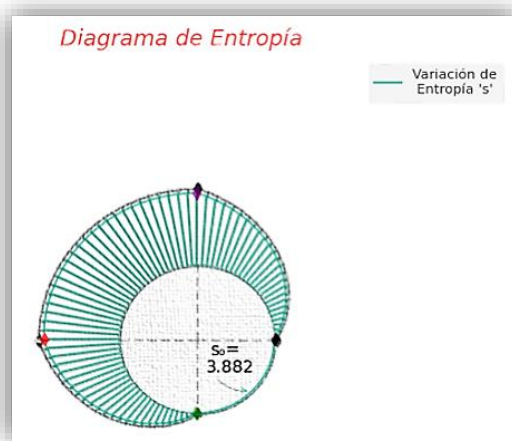
La tendencia en este análisis es similar a la tendencia del Análisis de sensibilidad en el método aproximado con algunas diferencias, en este caso la energía interna del estado 4 y el calor rechazado ya no se mantienen constantes cuando existe un aumento de la relación de compresión y es consecuencia de que en el método exacto las propiedades dependen de la temperatura y de calores específicos variables a diferencia del caso anterior.

Se observa también que las propiedades adicionales como la exergía y la entropía también tienden a aumentar con el aumento de la relación de compresión y la relación de corte-admisión.



**Figura 3.2:** Comparación del diagrama de energía método exacto.

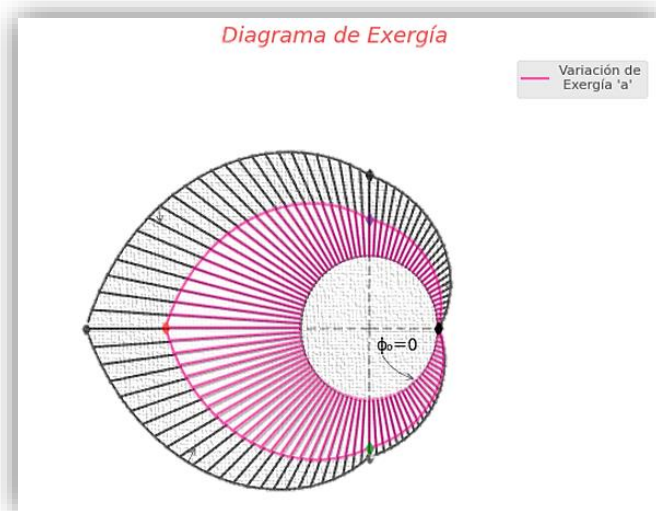
En la **Figura 3.2** se muestra la diferencia de un diagrama de energía con valores de relación de compresión ( $r$ ) = 20 y relación de corte-admisión ( $rc$ ) = 2.264 inicialmente con sus respectivos colores, debajo de ella en tonos grises se encuentra el diagrama de energía con valores de relación de compresión ( $r$ ) = 24 y relación de corte-admisión ( $rc$ ) = 3, para ilustrar en un solo gráfico la diferencia al aumentar estas variables.



**Figura 3.3:** Comparación del diagrama de entropía método exacto.

En la **Figura 3.3** se muestra la diferencia de un diagrama de entropía con valores de relación de compresión ( $r$ ) = 20 y relación de corte-admisión ( $rc$ ) = 2.264 inicialmente con su respectivo color, debajo de ella en tono gris se encuentra el diagrama de entropía con valores de relación de compresión ( $r$ ) = 24 y relación de corte-admisión ( $rc$ ) = 3, para ilustrar en un solo gráfico la diferencia al aumentar estas variables.





**Figura 3.4:** Comparación del diagrama de exergía método exacto.

En la **Figura 3.4** se muestra la diferencia de un diagrama de exergía con valores de relación de compresión ( $r$ ) = 20 y relación de corte-admisión ( $r_c$ ) = 2.264 inicialmente con su respectivo color, debajo de ella en tono gris se encuentra el diagrama de exergía con valores de relación de compresión ( $r$ ) = 24 y relación de corte-admisión ( $r_c$ ) = 3, para ilustrar en un solo gráfico la diferencia al aumentar estas variables.

En la exergía también resulta influenciado el resultado por el estado ambiente que el usuario establezca en los datos iniciales a continuación se muestra la tendencia en la **Tabla 3.6**.

**Tabla 3.6:** Análisis sensibilidad método exacto exergía.

Análisis sensibilidad método exacto exergía				
	$\phi_1$	$\phi_2$	$\phi_3$	$\phi_4$
	[kJ/kg]	[kJ/kg]	[kJ/kg]	[kJ/kg]
<b>p0: aumenta, T0: constante</b>	aumenta	aumenta	aumenta	aumenta
<b>p0: disminuye, T0: constante</b>	disminuye	disminuye	disminuye	disminuye
<b>P0: constante, T0: aumenta</b>	aumenta	aumenta	disminuye	disminuye
<b>P0: constante, T0: disminuye</b>	disminuye	disminuye	aumenta	aumenta

Al aumentar la presión del estado ambiente la exergía en cada uno de los estados tiende a aumentar y en el caso de disminuir la presión del estado ambiente la exergía en cada uno de los estados también disminuirá.

En el caso de aumentar la temperatura del estado ambiente la exergía del estado 1 y del estado 2 también aumenta, pero la exergía del estado 3 y 4 disminuye y esto se debe al aumento de la entropía en los estados 3 y 4, ya que por más que aumente la entropía del estado ambiente la diferencia entre la entropía de los estados 3 y 4 y la entropía del estado ambiente será siempre mucho mayor a la diferencia entre la entropía de los estados 1 y 2 y la entropía del estado ambiente.

Claramente si existe un aumento exagerado de la entropía del estado ambiente cambiará esta tendencia haciendo que las exergía en cada uno de los estados tiendan a aumentar conjuntamente, exclusivamente en la propiedad de exergía la tendencia no se encuentra marcada para todos los casos.

### **Eficiencia, relación de compresión y relación de corte- admisión**

La eficiencia del ciclo Diesel depende directamente de la adición y rechazo de calor en el ciclo, pero el papel que juega la relación de compresión y relación de corte-admisión es importante para mejorar esta propiedad.

Si hay un aumento de la relación de compresión y se mantiene el valor de la relación de corte-admisión en el ciclo aumenta la eficiencia del ciclo, pero al haber también un aumento del calor añadido se concluye que la eficiencia térmica aumentó porque teóricamente se mejoró la calidad del combustible.

En el caso contrario si se aumenta la relación de corte-admisión, pero se mantiene el valor de la relación de compresión, el valor de eficiencia térmica del ciclo disminuye, en este caso la razón para esta disminución de eficiencia térmica será que teóricamente la calidad del combustible empeoró.

Entonces para simular un aumento de la eficiencia térmica del ciclo usando teóricamente la misma calidad de combustible, se debe fijar el valor de calor añadido resultante en el ejercicio, posteriormente se aumenta la relación de compresión y se disminuye la relación de corte-admisión en una combinación de ambas propiedades que den como resultado dicho valor de calor añadido.

Y se podrá comprobar que efectivamente si se aumenta la relación de compresión usando teóricamente un combustible de la misma calidad, habrá una mejora de la eficiencia térmica del ciclo, a continuación, en las **Figura 3.5** y **Figura 3.6** se muestra un ejemplo.

<b>r:</b>	20	Relación de compresión $r = (V1/V2)$	<b>qa:</b>	1234.28	[kJ/kg]
<b>rc:</b>	2.264	Relación de corte-admisión $rc = (V3/V2) = (T3/T2)$	<b>qr:</b>	450.27	[kJ/kg]
			<b>η:</b>	63.52	[%]

**Figura 3.5:** Ejemplo de aumento de eficiencia del Ciclo Diesel (a).

<b>r:</b>	28	Relación de compresión $r = (V1/V2)$	<b>qa:</b>	1234.47	[kJ/kg]
<b>rc:</b>	2.105	Relación de corte-admisión $rc = (V3/V2) = (T3/T2)$	<b>qr:</b>	386.23	[kJ/kg]
			<b>η:</b>	68.71	[%]

**Figura 3.6:** Ejemplo de aumento de eficiencia del Ciclo Diesel (b).

### Producto final

El software finalizado cuenta con las funciones y módulos que se describen en Software resultante y corrección de errores, la ventana principal cuenta con las partes que se muestran a continuación en la **Figura 3.7**.

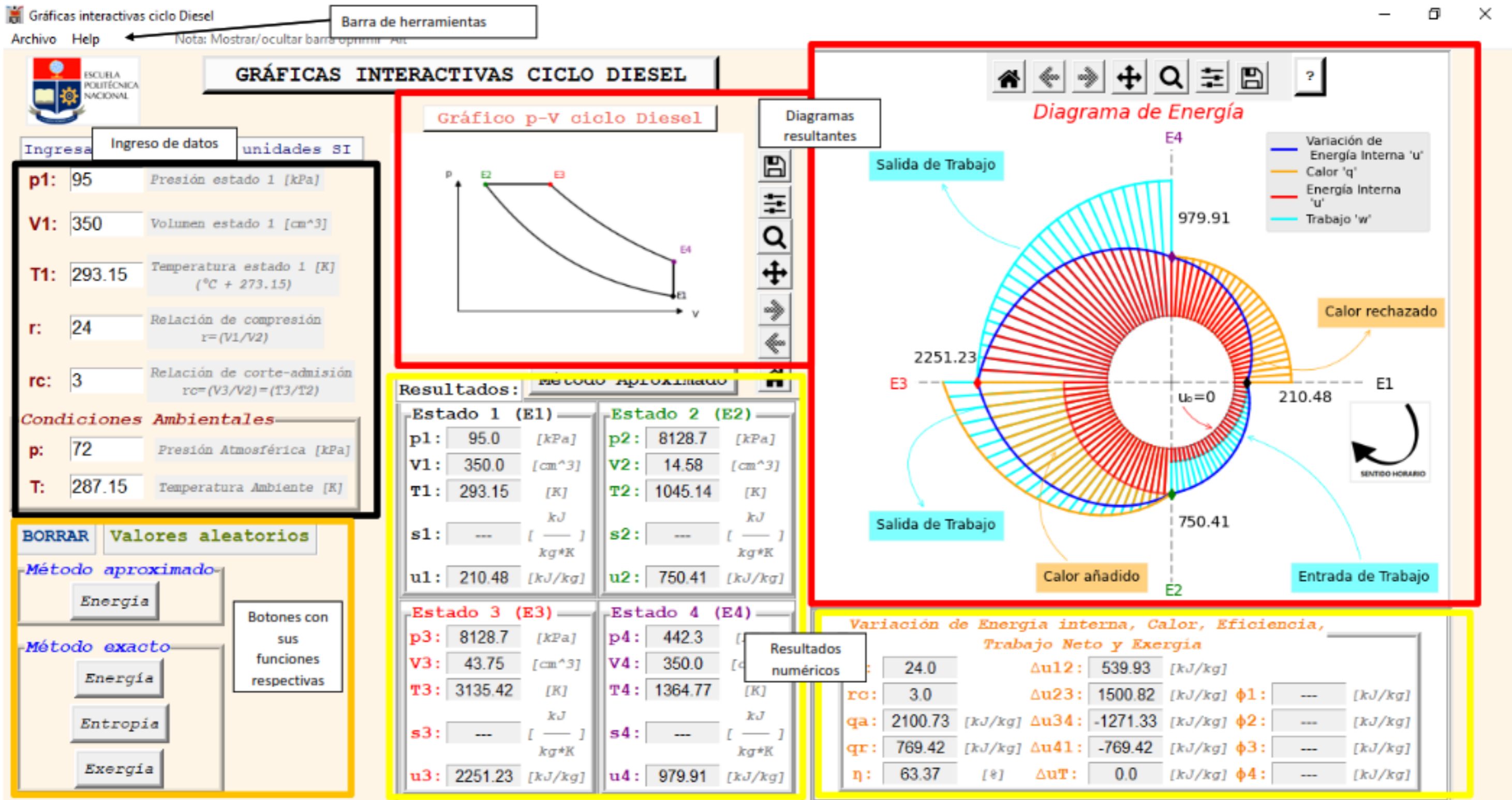


Figura 3.7: Software final con sus partes.

El archivo ejecutable se realiza mediante la herramienta “Pyinstaller” la cual se ejecuta en la consola Cmd del sistema, en la **Figura 3.8** se muestra un ejemplo.

```
Microsoft Windows [Versión 10.0.19044.1826]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\DELL>cd C:\Users\DELL\OneDrive - Escuela Politécnica Nacional\TESIS
C:\Users\DELL\OneDrive - Escuela Politécnica Nacional\TESIS> pyinstaller proyectofinal.py
```

**Figura 3.8:** Ejemplo de conversión a archivo ejecutable (.exe).

Finalmente, se realiza un manual de usuario que se adjunta en la sección de anexos, el cual es parte de los objetivos planteados en este proyecto y es importante para mejorar la experiencia del usuario y aclarar cualquier duda (ANEXO I).

## 3.2 Conclusiones

El software desarrollado en este proyecto ha generado resultados satisfactorios en su diseño e implementación final, teniendo como producto final un programa dinámico e interactivo de gran calidad con una interfaz gráfica amigable para el usuario, funciones útiles para cumplir con el objetivo de presentar resultados numéricos y gráficos de la resolución de ejercicios del Ciclo Diesel.

La información contenida en el marco teórico de este proyecto fue el pilar para establecer la ruta de realización de cada uno de los módulos del software, adicionalmente fue la referencia para la comprobación final de resultados y para el ajuste y depuración del código de programación.

El código final del software cuenta con módulos de cálculo para los métodos aproximado y exacto, estos módulos son la parte medular del funcionamiento y tienen la capacidad de funcionar sin necesidad de interactuar con las funciones extras del software si se desea, la interfaz gráfica y demás funciones giran en torno a estos módulos.

Los diagramas de energía, entropía y exergía, presentados en este componente cuentan con una sección de interpretación donde se describe cada diagrama a detalle y se explica su lectura, ya que no se encuentran basados en un diagrama existente, cabe añadir que fueron creados por los autores de los dos componentes que forman el proyecto “DIAGRAMAS TERMODINÁMICOS INTERACTIVOS DE LOS CICLOS DE POTENCIA OTTO Y DIESEL” en colaboración conjunta del Tutor el Ing. Miguel Ortega.

Para comprobar los resultados numéricos obtenidos por el software se comparó con resultados de ejercicios del Ciclo Diesel de [1], resultados de ejercicios recopilados de internet y finalmente resultados de ejercicios realizados con el software TermoGraf 5.7v, con ello se concluye que el programa tiene resultados satisfactorios con un mínimo error por factores de redondeo de cifras en cálculos e interpolaciones.

Se implementó un manual del software para garantizar el uso correcto de cada función que contine el programa y aclarar cualquier tipo de duda o inconveniente que pueda presentar el usuario.

Las variables con mayor relevancia en el cálculo de las propiedades resultantes del Ciclo Diesel son la relación de compresión y la relación de corte-admisión, ya que influyen directamente en el aumento o disminución de temperatura, volumen y presión, propiedades que son fundamentales para definir los estados termodinámicos.

Para simular un aumento de eficiencia en un ejercicio utilizando teóricamente un mismo combustible se debe fijar un valor de calor añadido y jugar con combinaciones de la relación de compresión y la relación de corte-admisión, en este caso aumentando la primera y disminuyendo la segunda respectivamente como se muestra en Eficiencia, relación de compresión y relación de corte- admisión.

El software DT-Diesel es una herramienta que servirá de complemento para el aprendizaje del Ciclo Diesel, su funcionalidad e interactividad al variar datos y obtener resultados inmediatos suponen mejorar la comprensión del usuario sobre lo que ocurre en este ciclo.

### **3.3 Recomendaciones**

Efectuar una investigación y recopilación teórica precisa para tener una base sólida al momento de realizar proyectos posteriores relacionados con el tema de este proyecto o sus derivaciones.

Se debe ingresar los datos iniciales siguiendo el formato establecido con las condiciones del caso para cada una de las propiedades tal como se describe en Datos de entrada, para evitar errores al utilizar el software.

Para el diseño del código de programación se recomienda realizar módulos de funcionamiento separados donde se alberguen las funciones principales que actúan de

forma independiente, para posteriormente incluirlas en un mismo código base que use estos módulos, con esto se evita tener que repetir líneas de programación y tener una mejor organización en la estructura del código.

Se recomienda incitar el uso de herramientas informáticas de tipo académico para complementar el aprendizaje de los estudiantes y apoyar al mismo tiempo a la creación e implementación de nuevas aplicaciones con contenido educativo.

Se recomienda la creación de más módulos para el software “Diagramas termodinámicos interactivos”, con los ciclos termodinámicos que aún no han sido tomados en cuenta, para crear una herramienta cada vez más completa y de uso libre para el aprendizaje de la asignatura de termodinámica.

## 4 REFERENCIAS BIBLIOGRÁFICAS

### Referencias

- [1] B. M. Cengel Yunus, Thermodynamics, An Engineering Approach (9th ed.), New York: McGraw-Hill, 2019.
- [2] L. F. García Rocha, «Desarrollo de una herramienta didáctica para el análisis de ciclos termodinámicos ideales,» Universidad Autónoma de Ciudad de Juárez, Ciudad de Juárez, 2019.
- [3] R. Stone, Introduction to Internal Combustion Engines, In Energy Conversion, Boca Ratón: MACMILLAN, 1999.
- [4] C. R. Severance, Python para todos (1ra. Ed.), USA, 2016.
- [5] Python, «Python,» 2022. [En línea]. Available: <https://www.python.org/about/>. [Último acceso: agosto 2022].
- [6] R. E. L. Briega, «Libro online de IAAR,» 22 julio 2017. [En línea]. Available: <https://iaarbook.github.io/python/>. [Último acceso: agosto 2022].
- [7] J. Shipman, Tkinter 8.5 reference a GUI for Python. 1-168, 2013.
- [8] Python, «tkinter.ttk — Tk widgets temáticos,» Python, 2022. [En línea]. Available: <https://docs.python.org/es/3/library/tkinter.ttk.html>. [Último acceso: agosto 2022].
- [9] J. S. Silva Cevallos, «Diseño y análisis de sistemas energéticos,» EPN, Quito, 2022.
- [10] Stackoverflow, «Stackoverflow,» 2022. [En línea]. Available: <https://stackoverflow.com/>.

[Último acceso: agosto 2022].

- [11] D. Stack, «DelftStack,» 2020. [En línea]. Available: <https://www.delftstack.com/es/>.  
[Último acceso: agosto 2022].

## **5 ANEXOS**

### **ANEXO I: Manual de Usuario.**



# Manual de Usuario

DT – DIESEL  
SOFTWARE DIAGRAMAS TERMODINÁMICOS  
INTERACTIVOS

MÓDULO CICLO DIESEL



Anthony Encalada

Quito- Ecuador

Agosto, 2022

## Contenido

<b><u>SOFTWARE Y SUS PARTES</u></b> .....	<b>1</b>
<u>Acerca del Software DT-Diesel</u> .....	1
<u>Partes del Software</u> .....	2
<b><u>INGRESO DE DATOS</u></b> .....	<b>3</b>
<b><u>DIAGRAMAS DE ENERGÍA, ENTROPÍA, EXERGÍA Y SU RESPECTIVA INTERPRETACIÓN</u></b> .....	<b>4</b>
<u>Interpretación diagrama de energía</u> .....	7
<u>Interpretación diagrama de entropía</u> .....	9
<u>Interpretación diagrama de exergía</u> .....	10
<b><u>FUNCIONES DEL SOFTWARE</u></b> .....	<b>11</b>

### Software y sus partes

#### Acerca del Software DT-Diesel

El Software DT-Diesel es un Trabajo de Integración Curricular presentado como requisito para la obtención del título de Ingeniero Mecánico en la Escuela Politécnica Nacional Quito- Ecuador. El autor del software: Anthony Encalada.

DT-Diesel es un software educativo para resolución de ejercicios de termodinámica específicamente del Ciclo Diesel, que cuenta con una interfaz gráfica intuitiva y funciones útiles para el usuario.

El principal objetivo de DT-Diesel es ser una herramienta para complementar el aprendizaje en el campo de la termodinámica, haciendo más fácil la comprensión de las propiedades de estado que forman parte del ciclo Diesel.

Los diagramas de energía, exergía y entropía que se obtienen con DT-Diesel fueron creados exclusivamente para este software por el Autor del software Sr. Anthony Encalada, el autor del Software DT-Otto Sr. Nicolas Nole y el Tutor del trabajo de Integración Curricular el Ing. Miguel Ortega.

Partes del Software

Gráficas interactivas ciclo Diesel

Archivo Help

Nota: Mostrar/ocultar barra optimizar Aut

Barra de herramientas

### GRÁFICAS INTERACTIVAS CICLO DIESEL

Ingreso de datos unidades SI

**p1:** 95 Presión estado 1 [kPa]  
**V1:** 350 Volumen estado 1 [cm<sup>3</sup>]  
**T1:** 293.15 Temperatura estado 1 [K] (°C + 273.15)  
**r:** 24 Relación de compresión  $r = (V1/V2)$   
**rc:** 3 Relación de corte-admisión  $rc = (V3/V2) = (T3/T2)$

**Condiciones Ambientales**  
**p:** 72 Presión Atmosférica [kPa]  
**T:** 287.15 Temperatura Ambiente [K]

BORRAR Valores aleatorios

Método aproximado

Método exacto

Botones con sus funciones respectivas

Energía  
Entropía  
Exergía

#### Gráfico p-v ciclo Diesel

#### Diagrama de Energía

Salida de Trabajo  
Calor rechazado  
Calor añadido  
Entrada de Trabajo

Variación de Energía interna 'u'  
Calor 'q'  
Energía Interna 'u'  
Trabajo 'w'

Resultados:

Estado 1 (E1)		Estado 2 (E2)	
p1:	95.0 [kPa]	p2:	8128.7 [kPa]
V1:	350.0 [cm <sup>3</sup> ]	V2:	14.58 [cm <sup>3</sup> ]
T1:	293.15 [K]	T2:	1045.14 [K]
s1:	---	s2:	---
u1:	210.48 [kJ/kg]	u2:	750.41 [kJ/kg]

Estado 3 (E3)		Estado 4 (E4)	
p3:	8128.7 [kPa]	p4:	442.3 [kPa]
V3:	43.75 [cm <sup>3</sup> ]	V4:	350.0 [cm <sup>3</sup> ]
T3:	3135.42 [K]	T4:	1364.77 [K]
s3:	---	s4:	---
u3:	2251.23 [kJ/kg]	u4:	979.91 [kJ/kg]

Resultados numéricos

rc:	3.0	$\Delta u_{12}$ :	539.93 [kJ/kg]	$\phi 1$ :	---	[kJ/kg]
qa:	2100.73 [kJ/kg]	$\Delta u_{23}$ :	1500.82 [kJ/kg]	$\phi 2$ :	---	[kJ/kg]
qr:	769.42 [kJ/kg]	$\Delta u_{34}$ :	-1271.33 [kJ/kg]	$\phi 3$ :	---	[kJ/kg]
$\eta$ :	63.37 [%]	$\Delta u_{41}$ :	-769.42 [kJ/kg]	$\phi 4$ :	---	[kJ/kg]
		$\Delta u_T$ :	0.0 [kJ/kg]			

Variación de Energía interna, Calor, Eficiencia, Trabajo Neto y Exergía

Fig. 1: Software y sus partes.



En la **Fig. 1** se observan las partes más importantes del software:

- Ingreso de datos: en esta sección se ingresan los datos del ejercicio
- Botones: los botones que se encuentran en este espacio son los encargados de las funciones principales de cálculo en el programa.
- Barra de herramientas: contiene más opciones de uso del programa como guardar, datos cargar datos, salir y la información del programa.
- Resultados numéricos: En estas cajas se puede visualizar los resultados de los cálculos realizados por el programa con cualquiera de los métodos.
- Diagramas resultantes: los diagramas resultantes de energía, entropía y exergía se muestran en estos espacios también cuentan con una barra de herramientas para interactuar con la gráfica.

## Ingreso de datos

Los datos de ingreso requeridos para el planteamiento y resolución del ejercicio están establecidos utilizando la **Fig. 29** como una referencia para identificar los estados y sus unidades conforme al Sistema Internacional de Unidades, a excepción del volumen. Estos datos iniciales son requeridos al usuario para iniciar el script de resolución numérica y pueden ser ingresados como enteros o flotantes, el formato de escritura de los datos iniciales utiliza el punto (.) como separador de decimales, también se debe tener en cuenta que ninguno de los datos puede ser menor a cero y en el caso de la temperatura mínima  $T_1$  se recomienda usar valores mayores a 273.15 K.

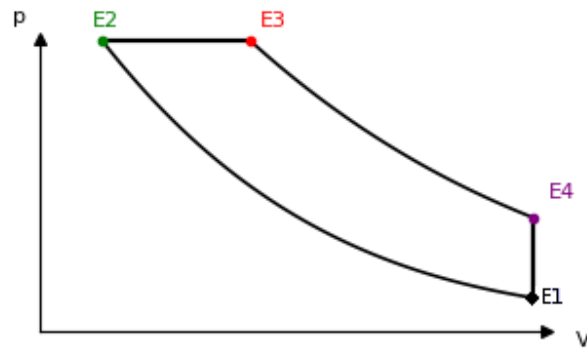


Fig. 29: Diagrama p-V referencial.

**Relación de compresión:**

$$r > 1$$

**Relación de corte-admisión:**

$$r_c > 1$$

**Estado 1:**

$p_1$ : en kilopascales [kPa].

$V_1$ : en centímetros cúbicos [ $\text{cm}^3$ ]

$T_1$ : en Kelvin [K]

**Condiciones ambientales:**

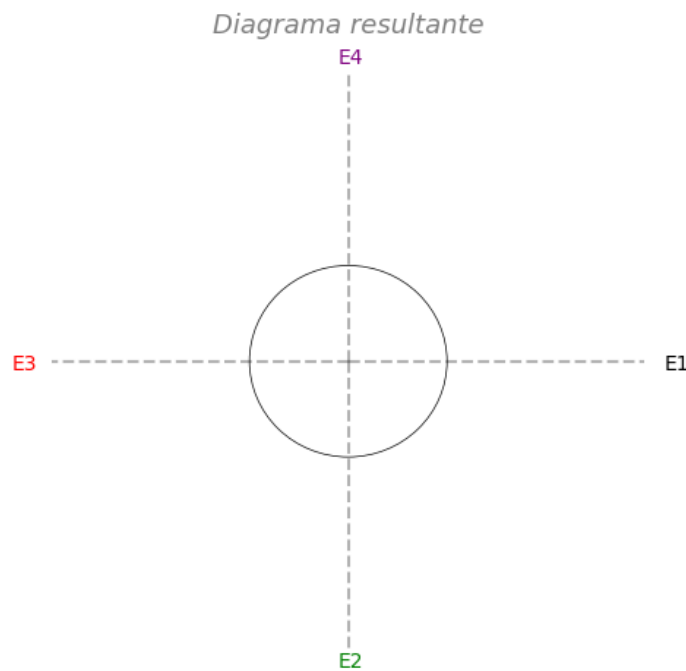
$p$  atmosférica: en kilopascales [kPa].

$T$  ambiente: en Kelvin [K]

## Diagramas de energía, entropía, exergía y su respectiva interpretación

La resolución gráfica que se plantea para la energía, entropía y exergía, no sigue un modelo establecido anteriormente, son diagramas diseñados específicamente

para este software; estos diagramas muestran la variación de la energía interna, entropía y exergía a través del ciclo Diesel, cuentan con una circunferencia centrada como base que representa el valor referencial de cada diagrama (**Fig. 3**), la variación en el tiempo de una propiedad desde un estado hasta el siguiente se expresa de forma angular mediante arcos variables, los radios variables que se observan conformando dichos arcos corresponden a la variación de las propiedades en cada punto del ciclo; cada uno de estos puntos utilizados para realizar los diagramas resultantes se encuentran mediante una expresión de linealización con el uso de un ángulo beta que varía en el rango de 0° a 360°, rango propiamente de los diagramas creados.



**Fig. 310:** Diagrama base.

Linealización para el primer cuadrante, beta de 0° a 90°, E4-E1:

$$x_n = \left( x_{E1} + (x_{E4} - x_{E1}) \cdot \left( \frac{\beta}{90^\circ} \right) \right)$$

Linealización para el primer cuadrante.

Linealización para el segundo cuadrante, beta de 90° a 180°, E3-E4:



$$x_n = \left( x_{E4} + (x_{E3} - x_{E4}) \cdot \left( \frac{\beta - 90^\circ}{90^\circ} \right) \right)$$

Linealización para el segundo cuadrante.

Linealización para el tercer cuadrante,  $\beta$  de  $180^\circ$  a  $270^\circ$ , E2-E3:

$$x_n = \left( x_{E3} + (x_{E2} - x_{E3}) \cdot \left( \frac{\beta - 180^\circ}{90^\circ} \right) \right)$$

Linealización para el tercer cuadrante.

Linealización para el cuarto cuadrante,  $\beta$  de  $270^\circ$  a  $360^\circ$ , E1-E2:

$$x_n = \left( x_{E2} + (x_{E1} - x_{E2}) \cdot \left( \frac{\beta - 270^\circ}{90^\circ} \right) \right)$$

Linealización para el cuarto cuadrante.

Donde:

$x$ : Representa la propiedad con la que se está trabajando.

$x_n$ : Es el valor numérico de la propiedad expresada en cualquier punto del ciclo.

Sea  $n$ :

$$n \in \mathbb{Z} / 1 \leq n \leq 100$$

Esto significa que se realiza el cálculo de 100 puntos en las representaciones gráficas, por lo tanto, la variación del ángulo  $\beta$  entre cada punto es de  $3.6^\circ$ .

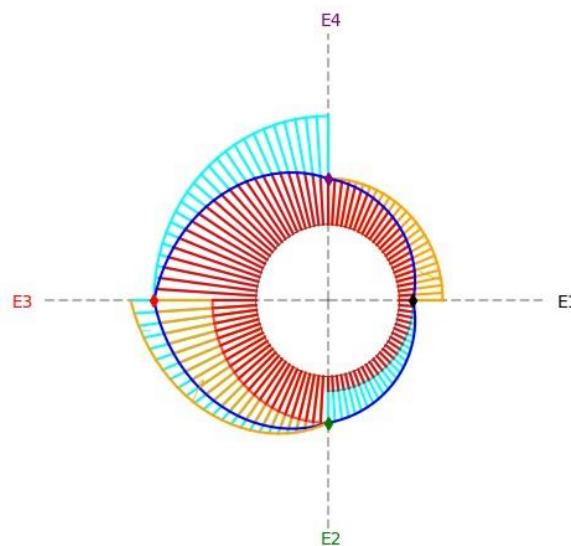
$\beta$ : Representa el ángulo de apoyo para la variación de las propiedades a través del diagrama.

Nota: La linealización que se utiliza para calcular las propiedades en cada punto

del ciclo es el resultado de aplicación de teoremas geométricos en una representación gráfica con coordenadas polares.

## Interpretación diagrama de energía

En la **Fig. 4** se muestra la variación de energía interna del ciclo Diesel y las propiedades involucradas en cada proceso para obtener dicha variación. El diagrama utiliza diferentes colores para representar estas propiedades, que son: el trabajo, el calor y la energía interna; los puntos marcados en cada eje en forma de rombo corresponden a los valores resultantes de energía interna ( $u$ ) en cada estado del ciclo (E1, E2, E3, E4).



**Fig. 4:** Diagrama de energía.

- Las líneas de color "celestes": Expresan el trabajo de entrada o salida en cada proceso del ciclo.
- Las líneas de color "naranja": Expresan la entrada o rechazo de calor en cada proceso del ciclo.
- Las líneas de color "rojo": Expresan la energía interna de cada proceso, sin tomar en cuenta el aumento de trabajo o calor.





- La línea de color "negro": Indica la variación de energía interna del ciclo completo a través de todos los procesos involucrados en el ciclo Diesel.
- La circunferencia central del diagrama representa el valor referencial del diagrama, en este caso será  $u_0=0$ .

De E1 a E2 se incrementa la energía interna, esto se debe a la entrada de trabajo en el proceso de compresión, por lo tanto, el trabajo se encuentra dentro del contorno de variación, ya que, en este caso la variación de energía es igual al trabajo que entra.

$$\Delta u_{12} = w_{12}$$

Variación de energía interna de E1 a E2.

De E2 a E3 ocurre un incremento de la energía interna debido al calor de entrada, pero también existe una salida de trabajo por el aumento del volumen a presión constante, por lo tanto, en este caso la variación de la energía interna será igual al calor añadido menos el trabajo de 2 a 3.

$$\Delta u_{23} = q_a - w_{23}$$

Variación de energía interna de E2 a E3.

De E3 a E4 la energía interna disminuye debido a la salida de trabajo en el proceso de expansión isentrópica, por lo tanto, el trabajo se encuentra fuera del contorno que indica la variación, ya que, la variación de la energía interna será igual al trabajo de salida y su valor es negativo.

$$\Delta u_{34} = -w_{34}$$

Variación de energía interna de E3 a E4.

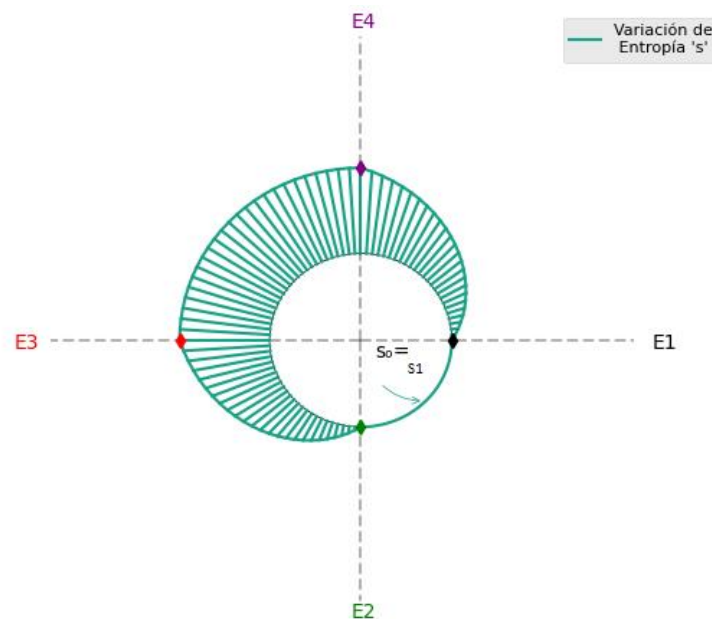
Finalmente, de E4 a E1 la energía interna disminuye debido al rechazo de calor a volumen constante, por lo tanto, la variación de energía interna es igual al calor rechazado y su valor es negativo, por eso el calor rechazado se encuentra fuera del contorno que indica la variación.

$$\Delta u_{41} = q_r \Rightarrow \Delta u_{41} = -q_{41}$$

Variación de energía interna de E4 a E1.

## Interpretación diagrama de entropía

En la **Fig. 5** se muestra la variación de entropía del ciclo Diesel, los puntos marcados en cada eje en forma de rombo corresponden a los valores resultantes de entropía ( $s$ ) en cada estado del ciclo (E1, E2, E3, E4). La interpretación de este diagrama resulta más simple que la interpretación anterior, ya que solo cuenta con dicha propiedad representada en un mismo color (verde huevo de pato).



**Fig. 5:** Diagrama de entropía.

Inicialmente se debe indicar que el valor de la circunferencia central referencial en este caso ya no representa un valor de  $s_0=0$ , sino es el valor de entropía más pequeño del ciclo que corresponde a " $s_1$ ", esto con el fin de observar con mayor claridad la variación de la entropía en el ciclo

De E1 a E2 la entropía se mantiene constante, ya que ocurre el proceso de compresión isentrópica y en este caso el arco entre ambos estados se mantiene

constante en todo el cuadrante.

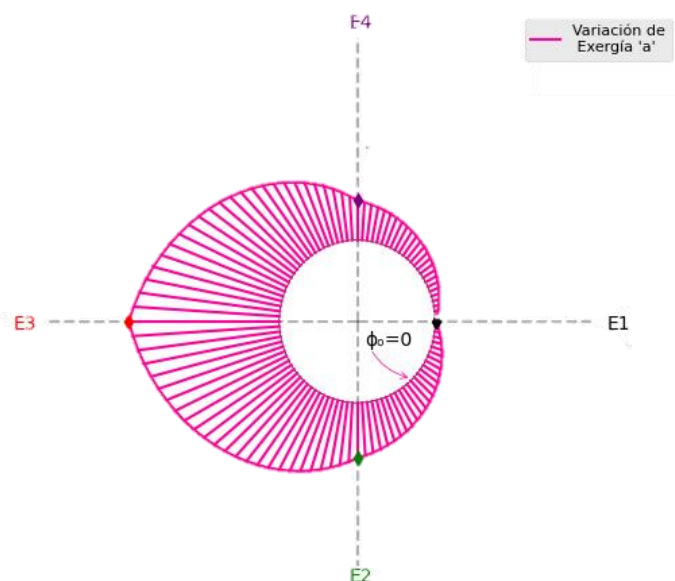
De E2 a E3 la entropía aumenta su valor por la entrada de calor en el ciclo a presión constante y se observa el aumento en el arco variable que conecta estos dos estados.

De E3 a E4 la entropía se mantiene constante, ya que ocurre el proceso de expansión isentrópica y en este caso el arco que une estos estados se mantiene constante en todo el cuadrante.

Por último, de E4 a E1 debido al rechazo de calor del ciclo, disminuye la entropía regresando al valor inicial el cual se toma como referencia en este diagrama.

## Interpretación diagrama de exergía

En la **Fig. 6** mostrada a continuación se observa la variación de exergía (sin flujo) del ciclo Diesel, los puntos marcados en cada eje en forma de rombo corresponden a los valores resultantes de exergía ( $\phi$ ) en cada estado del ciclo (E1, E2, E3, E4). Este diagrama al igual que el de entropía cuenta solo con una propiedad en este caso exergía representada en un mismo color (magenta).



**Fig. 6:** Diagrama de exergía.

De E1 a E2 la exergía aumenta debido al proceso de compresión isentrópica que ocurre entre estos estados.

De E2 a E3 la exergía se incrementa por la entrada de calor en el ciclo a presión constante y se observa el aumento en el arco variable que conecta estos dos estados.

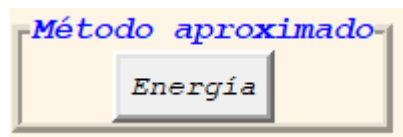
De E3 a E4 existe una disminución de la exergía por el proceso de expansión isentrópica el cual tiene un trabajo de salida.

Por último, de E4 a E1 debido al rechazo de calor del ciclo la exergía disminuye. En este proceso existe destrucción de exergía.

## Funciones del software

Como se mencionó anteriormente el software tiene funciones principales de cálculo que muestran los resultados numéricos acompañados de los diagramas interactivos.

Al oprimir el botón que se muestra en **Fig. 7**.



**Fig. 7:** Botón de cálculo por el método aproximado con diagrama de energía.

Se obtiene lo siguiente (**Fig. 8**):

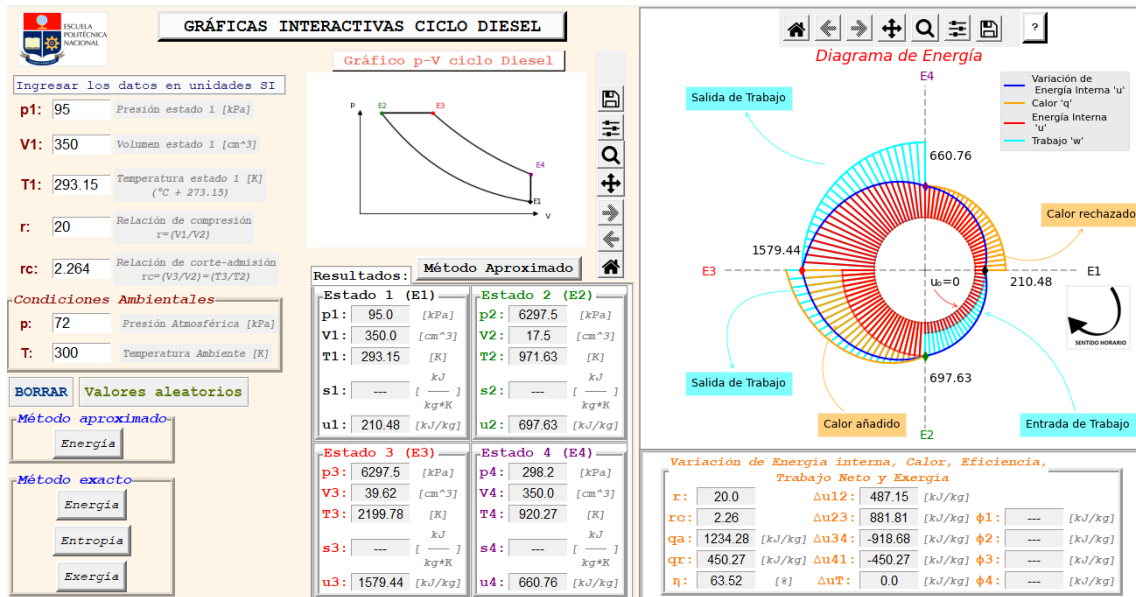


Fig. 8: Resultados de cálculo por el método aproximado con diagrama de energía.

Al oprimir los botones que se muestran en Fig. 9.

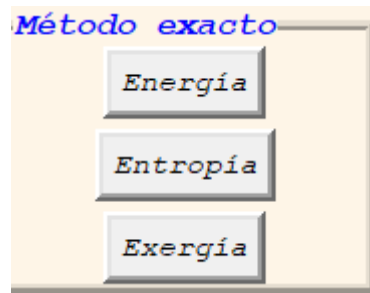


Fig. 9: Botones de cálculo por el método exacto con diagramas de energía, entropía y exergía respectivamente.

Se obtiene lo siguiente (Fig. 10, Fig. 11, Fig. 12):

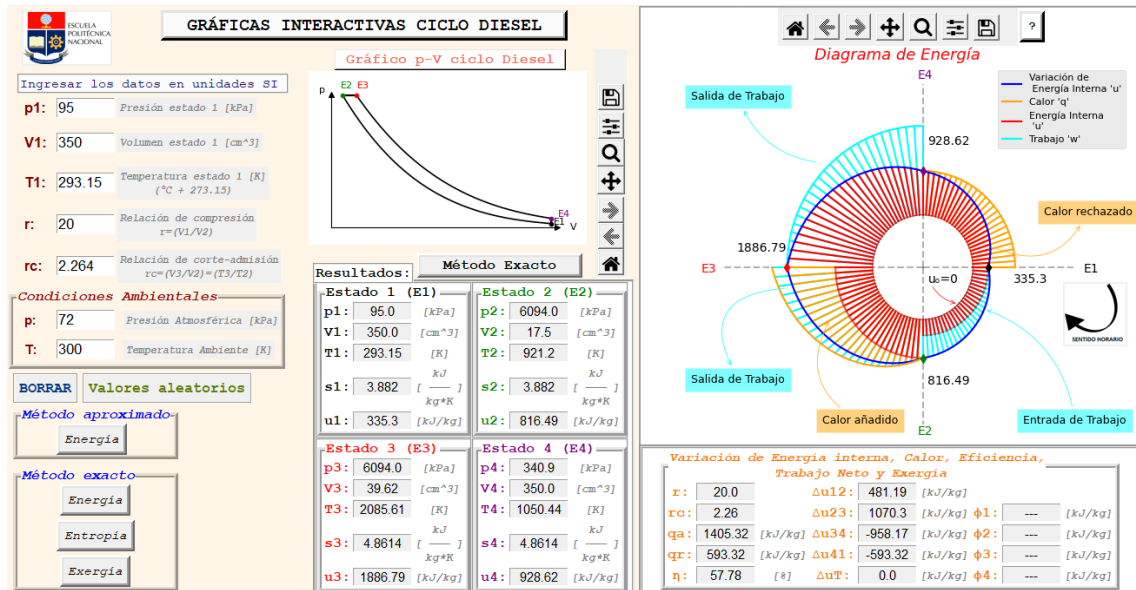


Fig. 10: Resultados de cálculo por el método exacto con diagrama de energía.

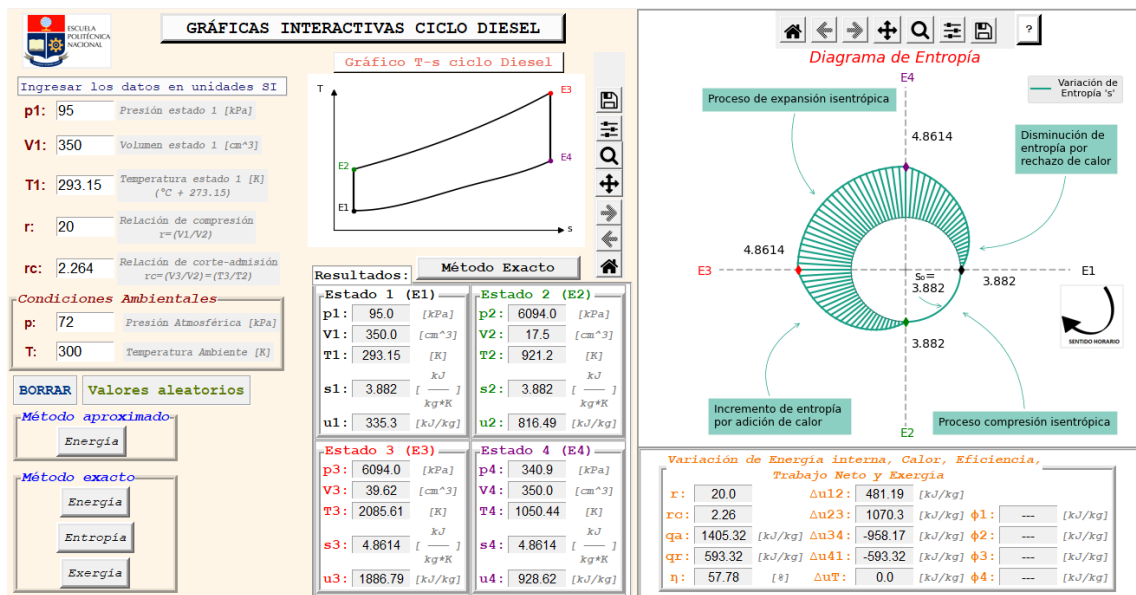


Fig. 11: Resultados de cálculo por el método exacto con diagrama de entropía.

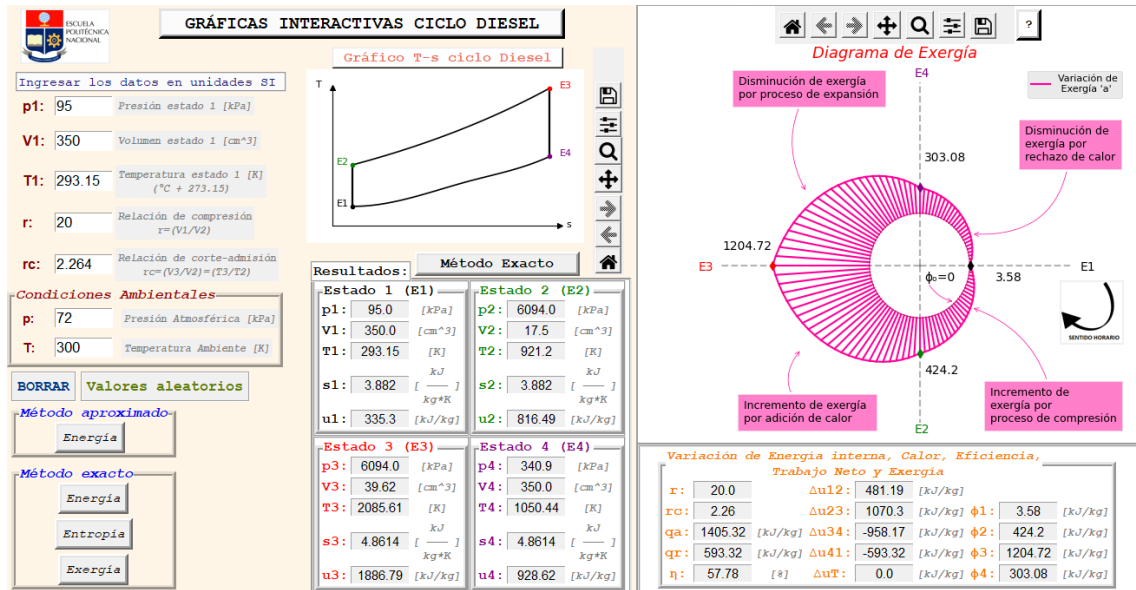


Fig. 12: Resultados de cálculo por el método exacto con diagrama de exergía.

Al oprimir el botón que se muestra en Fig. 13.



Fig. 13: Botón para borrar los datos de entrada.

Se borran los datos ingresados como se muestra en Fig. 14.

**p1:** 0 Presión estado 1 [kPa]  
**V1:** 0 Volumen estado 1 [cm<sup>3</sup>]  
**T1:** 0 Temperatura estado 1 [K] (°C + 273.15)  
**r:** 0 Relación de compresión  $r = (V1/V2)$   
**rc:** 0 Relación de corte-admisión  $rc = (V3/V2) = (T3/T2)$

**Condiciones Ambientales**  
**p:** 0 Presión Atmosférica [kPa]  
**T:** 0 Temperatura Ambiente [K]

Fig. 14: Datos iniciales borrados.

Al oprimir el botón que se muestra en **Fig. 15**.

Valores aleatorios

**Fig. 15:** Botón de valores aleatorios.

Se ingresan automáticamente datos iniciales aleatorios en el ejercicio como se muestra en **Fig. 16**.

<b>p1:</b>	70	Presión estado 1 [kPa]
<b>V1:</b>	425	Volumen estado 1 [cm <sup>3</sup> ]
<b>T1:</b>	297	Temperatura estado 1 [K] (°C + 273.15)
<b>r:</b>	23	Relación de compresión $r = (V1/V2)$
<b>rc:</b>	2.84	Relación de corte-admisión $rc = (V3/V2) = (T3/T2)$
<b>Condiciones Ambientales</b>		
<b>p:</b>	87	Presión Atmosférica [kPa]
<b>T:</b>	280	Temperatura Ambiente [K]

**Fig. 16:** Datos iniciales aleatorios.

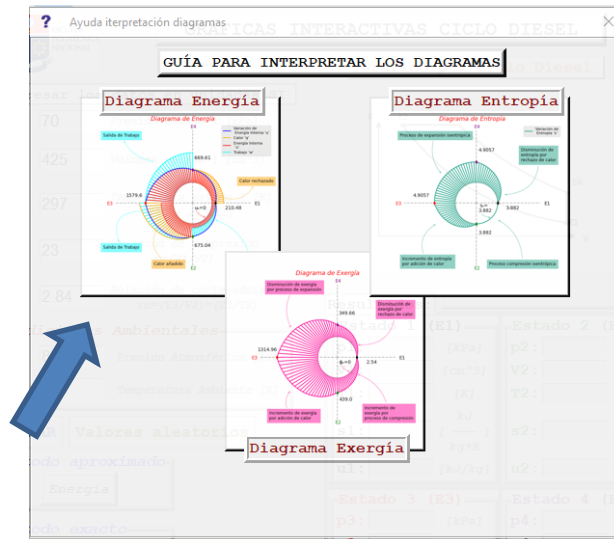
La función de ayuda para la interpretación de diagramas se activa al oprimir el botón mostrado en **Fig. 17**.



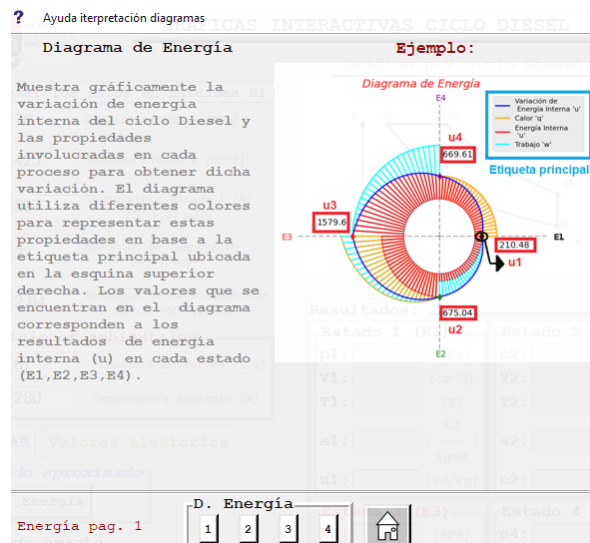
**Fig. 17:** Botón de ayuda para diagramas.



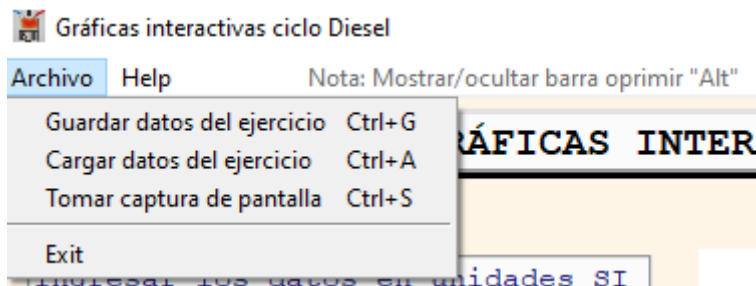
Se desplegará una ventana emergente donde se encuentra información para la interpretación de los diagramas al igual que se encuentra en la sección Diagramas de energía, entropía, exergía y su respectiva interpretación (**Fig. 18, Fig. 19**).



**Fig. 18:** Ventana de ayuda para diagramas (a).

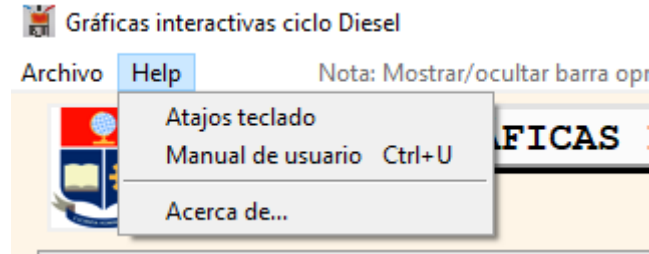


**Fig. 19:** Ventana de ayuda para diagramas (b).





**Fig. 20:** Barra de herramientas (a).



**Fig. 21:** Barra de herramientas (b).

Finalmente, en la barra de herramientas (**Fig. 20, Fig. 21**) existen las siguientes funciones:

- Guardar los datos iniciales del ejercicio: se guardan los datos iniciales en un documento tipo texto (.txt), para poder ser cargados posteriormente.
- Cargar datos iniciales del ejercicio previamente guardados: se cargan datos iniciales del ejercicio anteriormente guardados en formato de texto (.txt).
- Tomar captura de toda la pantalla: se guarda una imagen de toda la pantalla.
- Salir del programa:
- Atajos del teclado: muestra los atajos disponibles del teclado.
- Manual de usuario: abre el manual de usuario.
- Acerca del software: muestra la información del software en una ventana emergente.

**Software:** DT-Diesel.

**Realizado por:** Anthony Encalada, Estudiante de Ingeniería Mecánica de la Escuela Politécnica Nacional (EPN), Quito – Ecuador.

**Tutor:** Ing. Miguel Ortega.

**Agosto del 2022**