

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA MECÁNICA**

**“DISEÑO Y SIMULACIÓN DE UN EQUIPO DIDÁCTICO PARA  
ESTUDIAR EL FUNCIONAMIENTO REAL DE UNA BOMBA DE  
CALOR PARA LA PRODUCCIÓN DE AGUA CALIENTE  
SANITARIA”**

**“DESARROLLO DE UNA HERRAMIENTA INFORMÁTICA  
DIDÁCTICA PARA EL ESTUDIO TERMODINÁMICO DE UNA  
BOMBA DE CALOR PARA PRODUCCIÓN DE AGUA  
CALIENTE SANITARIA”**

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

**CRISTIAN SANTIAGO MEDINA ACOSTA**

**cristian.medina@epn.edu.ec**

**DIRECTOR: ING. NARANGO MENDOZA CARLOS ANDRÉS, PhD.**

**Carlos.naranjo@epn.edu.ec**

**DMQ, octubre 2023**

## **CERTIFICACIONES**

Yo, CRISTIAN SANTIAGO MEDINA ACOSTA 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.

---

**CRISTIAN MEDINA A.**

Certifico que el presente trabajo de integración curricular fue desarrollado por CRISTIAN SANTIAGO MEDINA ACOSTA, bajo mi supervisión.

---

**Ing. CARLOS NARANJO, PhD.**

**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.

CRISTIAN SANTIAGO MEDINA ACOSTA

Ing. CARLOS ANDRÉS NARANJO MENDOZA, PhD.

## DEDICATORIA

Dedico mi tesis principalmente a Dios, por nunca abandonarme y darme las fuerzas para culminar con éxito esta etapa.

Les dedico a mis padres con todo mi corazón, Ana Acosta y Sergio Medina, por su apoyo, amor incondicional y por motivarme con su ejemplo a seguir adelante ante cualquier circunstancia. Su compañía durante mi vida ha sido el faro que ilumina mi camino junto con su paciencia y sabiduría podré seguir enfrentándome a los desafíos que se presenten. Su dedicación y sacrificio no solo han sido evidentes en los momentos de triunfo, sino también en las horas de incertidumbre y esfuerzo.

A mis hermanos; Paúl, Pamela y Efrén, por su siempre cálida compañía que me mantuvo firme y determinado en cada paso que di. Su aliento constante y ejemplo me impulsa cada día más y estoy convencido que nuestra unión familiar y de amistad es inquebrantable.

También dedico a Melany. Tu constante apoyo, paciencia y amor incondicional han sido un amortiguador para el estrés y retos que he enfrentado durante esta etapa. Tu presencia es sin duda un recordatorio que el éxito no solo se encuentra en logros individuales, sino también en las conexiones que alimentan el alma, es por eso por lo que me siento afortunado de haberte tenido a mi lado en esta etapa.

## **AGRADECIMIENTO**

Agradezco profundamente a la mejor universidad del Ecuador, la Escuela Politécnica Nacional, por brindarme la oportunidad de adquirir conocimientos y desarrollarme en un entorno enriquecedor. Les agradezco por el crecimiento personal y profesional que he tenido, gracias a sus recursos que me proporcionaron que fueron fundamentales para mi formación.

Mi más sincero agradecimiento a mi tutor Carlos Naranjo, guía y conocimientos han sido una fuente constante de sabiduría para poder cumplir con este reto. Gracias por su paciencia y habilidad para transmitir conceptos complejos de manera sencilla. Su compromiso con el éxito de sus estudiantes es digno de admiración.

Un agradecimiento muy especial a mi amigo Joel Cachaguay, una amistad así no se encuentra en cualquier lugar, para mi es un tesoro muy valioso todos los momentos y desafíos que juntos hemos superado dentro y fuera de la universidad. Te deseo todo lo mejor y muchos éxitos.

A mi grupo de amigos “los reales”, que, aunque el nombre no les haga justicia, supieron brindarme su amistad e hicieron que esta etapa no este solo del éxito individual, sino también en el éxito de haber forjado una amistad sincera que alimenta el alma. Siempre contarán con mi amistad César, Andrés, Kevin, Sebastián, Alison, Alex.

# ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VII
ABSTRACT .....	VIII
1 INTRODUCCIÓN.....	1
1.1 Descripción de componente desarrollado .....	2
1.2 Objetivo general .....	3
1.3 Objetivos específicos .....	3
1.4 Alcance .....	4
1.5 Marco teórico .....	4
1.5.1 Sistemas de producción de agua caliente sanitaria.....	4
1.5.2 Bombas de Calor .....	6
1.5.3 Refrigerantes .....	12
1.5.4 Sistemas de adquisición de datos (SAD).....	13
1.5.5 Lenguajes de programación .....	16
2 METODOLOGÍA.....	19
2.1 Determinación de parámetros de análisis de la bomba de calor .....	21
2.1.1 Ubicación de sensores según bibliografía .....	21
2.1.2 Variables termodinámicas .....	22
2.1.3 Descripción del equipo – Especificaciones técnicas.....	23
2.2 Selección del lenguaje de programación.....	24
2.2.1 Opciones de lenguajes de programación .....	25
2.2.2 Requisitos del software para el sistema didáctico .....	26
2.2.3 Criterios para selección del lenguaje de programación .....	27
2.3 Desarrollo del software y diseño de la GUI .....	28
2.3.1 Librerías para Python .....	29
2.3.2 Programación para la adquisición de datos en Python.....	31
2.3.3 Programación para el cálculo de propiedades termodinámicas .....	33

2.3.4	Programación para almacenamiento y registro de datos.....	36
2.3.5	Programación para la obtención diagramas termodinámicos y gráficas para monitoreo de presión y temperatura .....	38
2.3.6	Diseño de la interfaz gráfica de usuario GUI .....	42
2.3.7	Programación de la Interfaz gráfica .....	44
2.3.8	Programación para obtener un archivo ejecutable .....	47
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	49
3.1	Resultados .....	49
3.1.1	Parámetros de análisis de la bomba de calor .....	49
3.1.2	Software e Interfaz gráfica desarrollada en Python .....	50
3.1.3	Funcionamiento del software: Heat Pump Monitor .....	51
3.1.4	Ventana “Práctica” .....	51
3.1.5	Manual de Usuario.....	54
3.1.6	Simulación y prueba del software: Heat Pump Monitor .....	59
3.2	Conclusiones.....	62
3.3	Recomendaciones.....	63
4	REFERENCIAS BIBLIOGRÁFICAS .....	64
5	ANEXOS.....	67
	ANEXO I.....	67
	ANEXO II.....	69
	ANEXO III.....	70
	ANEXO IV .....	71
	ANEXO V .....	72

## RESUMEN

El objetivo general de este proyecto fue desarrollar una herramienta informática para el análisis del comportamiento termodinámico real de una bomba de calor para el calentamiento de agua caliente sanitaria para el laboratorio de termodinámica de la FIM-EPN.

La metodología que se utilizó fue de tipo teórico-práctico. Se utilizó teoría de sistemas energéticos y el ciclo de refrigeración de compresión para determinar los parámetros de análisis del sistema didáctico, se establecieron las ubicaciones y puntos de medida de presión y temperatura y se especificaron los elementos que conforman el sistema de adquisición de datos a implementar en el laboratorio. Establecidos los parámetros y dispositivos se seleccionó el lenguaje de programación Python a través del método de ordinal de criterios ponderados. Se recopilaron las librerías necesarias para poder cumplir con los objetivos de la herramienta informática como el monitoreo de variables termodinámicas en tiempo real, obtención de diagramas termodinámicos y lectura y registro de datos obtenidos de un dispositivo DAQ a través de sensores de presión y temperatura. Finalmente, se establecieron requisitos de diseño de la interfaz gráfica mediante las normas ISO 9241.

Se obtuvo un software con interfaz gráfica que permite el análisis termodinámico de una bomba de calor para producción de agua caliente sanitaria. El código de programación obtenido fue en software libre (Python) y se utilizaron las librerías para leer, codificar, almacenar, registrar y utilizar los datos proporcionados por los sensores para poder realizar el estudio del comportamiento termodinámico de una bomba de calor para producción de agua caliente sanitaria.

### **PALABRAS CLAVE:**

Sistema de adquisición de datos, desarrollo de software e interfaces gráficas, GUI, bombas de calor, análisis de sistemas energéticos, equipos e instrumentación para equipos de laboratorio, programación en Python.



## **ABSTRACT**

The general objective of this project was to develop a computer tool for the analysis of the thermodynamic behavior of a heat pump for heating domestic hot water for the Thermodynamics Laboratory of FIM-EPN.

The methodology used was theoretical-practical. The theory of energy systems and the compression refrigeration cycle were used to determine the parameters for the analysis of the didactic system. The locations and measurement points for pressure and temperature were established, and the elements that make up the data acquisition system to be implemented in the laboratory were specified. Once the parameters and devices were established, the Python programming language was selected using the weighted ordinal criteria method. The necessary libraries were gathered to fulfill the objectives of the computer tool, such as real-time monitoring of thermodynamic variables, obtaining thermodynamic diagrams, and reading and recording data obtained from a DAQ device through pressure and temperature sensors. Finally, design requirements for the graphical interface were established following ISO 9241 standards.

A software with a graphical interface was obtained, allowing for the thermodynamic analysis of a heat pump for the production of domestic hot water. The obtained programming code was in open-source software (Python), and libraries were used to read, encode, store, record, and utilize data provided by sensors in order to carry out the study of the thermodynamic behavior of a heat pump for the production of domestic hot water.

### **KEYWORDS:**

Data acquisition system, software development and graphical interfaces, GUI, heat pumps, energy systems analysis, equipment and instrumentation for laboratory equipment, Python programming.

# 1 INTRODUCCIÓN

En Ecuador, la transformación de una matriz energética dependiente de combustibles fósiles hacia fuentes renovables y la implementación de medidas y políticas públicas del uso eficiente de la energía ha provocado el incremento del uso de sistemas energéticos más eficientes dentro de todos los sectores de consumo del país.

En el 2021 el 93.2% de producción de energía eléctrica correspondió a energías renovables, especialmente de recursos hídricos, lo que significó una reducción de uso de combustibles fósiles a un histórico mínimo [1]. Por otro lado, en el sector residencial existe un aumento del consumo de combustibles fósiles, se tiene un incremento anual promedio del 2% de gas licuado de petróleo (GLP) en el periodo 2011-2021 [2]. En este contexto donde existe una disminución de consumo de combustibles fósiles para generación de electricidad es preocupante notar su incremento en el sector residencial.

El sector residencial es el tercer sector de mayor consumo energético, con un 13.9% para el 2021. Sin embargo, es el mayor consumidor de GLP con un 70.4 %, esto se debe a que dentro del sector residencial su mayor consumo es el GLP, con un 52.9% y posterior la electricidad con 37.9% [2]. Esto es preocupante ya que el Ecuador importa el 84% de la demanda total de GLP [3].

Respecto al sector residencial, el uso final que se les da a estas energías secundarias (electricidad y GLP) es para calefacción, calentamiento y cocción de alimentos, enfriamiento, iluminación y en electrodomésticos [4]. Se estima que el 10% de energía eléctrica consumida en una vivienda corresponde al calentamiento de agua mientras que el GLP se destina en su mayoría solo a la cocción y calentamiento de agua, siendo específicamente el calentamiento de agua alrededor del 30% del consumo de GLP, así el calentamiento de agua se estima en un 20 % del consumo de energía en una vivienda [5].

En el Ecuador, los sistemas para producción de agua caliente sanitaria (ACS) más utilizados son de tipo eléctrico y de GLP, estos llegan a tener una eficiencia del 55 y 65 % respectivamente [5].

En este contexto el desarrollo del presente proyecto tiene como finalidad aportar al Laboratorio de Termodinámica de la facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional con el desarrollo un sistema didáctico para el análisis termodinámico del comportamiento de una bomba de calor real para la producción de agua caliente sanitaria, el cual será monitoreado mediante un software. Esto servirá para el análisis y

estudio del comportamiento de una bomba de calor de una forma práctica y para ayudará a formar profesionales capaces de resolver la problemática planteada.

## **1.1 Descripción de componente desarrollado**

En el presente proyecto se desarrolló un software (“Heat Pump Monitor”), como herramienta informática, para el análisis termodinámico de una bomba de calor para producción de agua caliente sanitaria. El software fue desarrollado en Python utilizando el paradigma de resolución de programación orientado a objetos (POO), el software interactúa con el usuario a través de una interfaz gráfica.

La interfaz gráfica de usuario incluye 4 ventanas, la ventana principal “Práctica” permite el monitoreo y análisis termodinámico de una bomba de calor a través de datos provenientes de sensores, las otras 3 ventanas están destinadas a la revisión teórica, resolución de un cuestionario de temas relacionados a bombas de calor y un manual para que el usuario tenga la guía necesaria para utilizar el software.

El análisis termodinámico, que se realiza en la ventana principal del software, se lo realiza mediante propiedades termodinámicas calculadas con CoolProp y finalmente para la obtención del desempeño del sistema se utilizó una regresión polinómica para estimar la eficiencia isentrópica del compresor.

El software desarrollado tiene pensado implementarse en un sistema didáctico de una bomba de calor para producción de ACS, una vez se encuentre disponible los equipos, para ello el presente proyecto cuenta con la siguiente estructura.

### **Estructura del trabajo**

#### **Capítulo 1: Introducción**

En este capítulo se realiza una revisión bibliográfica de sistemas de producción de agua caliente sanitaria incluida la bomba de calor. Se realiza una revisión de los fundamentos acerca del funcionamiento y elementos principales de una bomba de calor. Se encuentra una revisión literaria de sistemas de adquisición de datos, que son la forma más eficiente, en la actualidad, para el análisis de sistemas energéticos de equipos de laboratorio, y la revisión de elementos necesarios para implementar un sistema didáctico para una bomba de calor.

## **Capítulo 2: Metodología**

La metodología se modula en 4 etapas para el desarrollo y cumplimiento del objetivo principal, las cuales están descritas en las siguientes secciones. En este capítulo se determinaron los parámetros necesarios basados en la revisión bibliográfica anterior. Se detallan los algoritmos bases necesarios para el funcionamiento del software. Además, se implementa una metodología de diseño para la programación de la GUI en Python y algoritmos y manejo de librerías necesarias para el desarrollo del software a implementar en el sistema de adquisición de datos.

## **Capítulo 3: Resultados, Conclusiones y recomendaciones**

En este capítulo se tiene la interfaz gráfica de usuario (GUI) desarrollada, la cual es parte del software para el estudio de la bomba de calor. Se explica las ventanas que componen la GUI y sus funciones principales y como estas cumplen con los objetivos planteados y los requerimientos de diseño establecidos en la metodología.

Se concluye el nivel de cumplimiento del presente proyecto con los objetivos establecidos y recomendaciones para el manejo correcto del software y mejoras posibles para optimizar el funcionamiento del programa.

Finalmente, en **ANEXOS** se encuentra los códigos de programación obtenidos del desarrollo del proyecto presente y se encuentra disponible el software ejecutable.

### **1.2 Objetivo general**

Desarrollar una herramienta informática para el análisis del comportamiento termodinámico real de una bomba de calor para el calentamiento de agua caliente sanitaria para el laboratorio de termodinámica de la FIM-EPN.

### **1.3 Objetivos específicos**

1. Recopilar información bibliográfica sobre sistemas y herramientas informáticas de monitoreo de sistemas energéticos para laboratorio.
2. Determinar los parámetros para el análisis termodinámico de la bomba de calor.
3. Desarrollar un software con interfaz gráfica mediante la programación de código en software libre capaz de utilizar los datos tomados para el cálculo de variables termodinámicas y generar gráficos del comportamiento termodinámico para el estudio de la bomba de calor de uso doméstico.

4. Validar el funcionamiento del software mediante datos simulados de una bomba de calor.
5. Implementar un cuestionario y marco teórico, que sirva como refuerzo para el estudio de la bomba de calor, dentro del software implementado en el sistema didáctico del laboratorio de la FIM-EPN.
6. Elaborar un manual para el usuario sobre el manejo del software implementado.

## **1.4 Alcance**

El presente proyecto tiene como alcance desarrollar una herramienta informática para su implementación en el sistema didáctico de una bomba de calor de uso doméstico para el laboratorio de termodinámica de la Escuela Politécnica Nacional. La herramienta informática debe ser capaz de monitorear la bomba de calor y realizar cálculos termodinámicos relevantes para el estudio del comportamiento de una bomba de calor. Además, debe de permitir la visualización de datos entregados por sensores y diagramas termodinámicos en tiempo real.

## **1.5 Marco teórico**

### **1.5.1 Sistemas de producción de agua caliente sanitaria**

En el Ecuador la mayoría de los sistemas de producción de agua caliente sanitaria corresponden a sistemas que generan calor a través de la combustión de combustibles fósiles, especialmente GLP, y por sistemas eléctricos, estos dos sistemas son los más utilizados en el Ecuador, sin embargo, no son eficientes en el uso de la energía [6]. A continuación, se detallan los sistemas de calentamiento de agua más utilizados en el sector residencial.

#### **Clasificación de los sistemas de calentamiento**

Actualmente hay muchas tecnologías y sistemas térmicos capaces de suplir con la demanda de ACS en el sector residencial. Una forma de categorizar los tipos de calentadores existentes es mediante el origen de su fuente de energía. En la Figura 1.1 se muestra un esquema de los sistemas de calentamiento según su fuente primaria de energía.



**Figura 1.1.** Sistemas de calentamiento según su sistema [7]

### **Sistemas de calentamiento por combustión**

Los sistemas de calentamiento por combustión son de combustión sólida como leña o carbón y de combustión a gas como el GLP, siendo este último el más común y utilizado en el Ecuador dentro de esta categoría [3]. El tipo más común de estos sistemas son las calderas y calefones [5].

La eficiencia de este tipo de sistema puede variar, sin embargo, la eficiencia de los calefones domésticos a gas oscila entre el 55% al 80% [5]. Por lo que no resulta en un sistema eficiente en el uso de la energía, no obstante, se utiliza con gran frecuencia en Ecuador debido al subsidio para el GLP.

### **Sistemas de calentamiento por energía eléctrica**

Los sistemas de calentamiento por energía eléctrica se subdividen en dos categorías según su principio de calentamiento como se puede observar en la Figura 1.1.

El calentamiento de agua por resistencia eléctrica se basa en el efecto Joule, este tipo de calentamiento tiene una eficiencia alrededor del 65% [5].

El calentamiento de agua por inducción magnética resulta ser más eficiente, alrededor del 95% [5]. Sin embargo, la alta demanda de agua caliente sanitaria para los hogares hace que el consumo final de electricidad siga siendo alto, a pesar de su alta eficiencia.

### **Sistemas de calentamiento por captación de energía solar**

Existe una gran variedad de tipos de calentadores que utilizan la captación de energía solar para calentamiento de agua, los más comunes son los paneles planos y tubos de vacío, los cuales han ido adquiriendo popularidad dentro de Ecuador [5]. Este sistema tiene una eficiencia entre el 50% y el 70% [6]. Si bien la eficiencia no es una desventaja, ya que no

produce un costo de energía para el usuario final, este sistema depende del potencial de irradiación del lugar geográfico y un sistema de acumulación.

### **Sistemas de Producción ACS dentro del sector residencial en el Ecuador**

En Ecuador, entre el 70 y el 80 % de la producción de ACS, se da con GLP con sistemas tipo calefón o caldero, y en torno al 20 % se usan sistemas con resistencia eléctrica [6].

No se mencionó fuentes como biogás, biodiesel, etc. porque no ocupan un porcentaje representable dentro del Balance Energético Nacional (BEN) de usos de energía en el sector residencial, pero se puede decir que cualquier generador de ACS mediante combustión debería remplazarse, pues, aunque los costos pueden reducirse por usar fuentes renovables, contribuyen al calentamiento global.

### **Comparación entre los sistemas de producción de ACS**

En la Tabla 1.1 se puede apreciar de mejor manera los sistemas de calentamiento, su eficiencia y su fuente de energía.

**Tabla 1.1.** Comparación de sistemas de producción de ACS y la bomba de calor [5], [8]

<b>SISTEMAS PARA PRODUCCIÓN DE ACS</b>						
	<b>Combustión</b>		<b>Electricidad</b>		<b>Captación Solar</b>	<b>Híbrido</b>
<b>Categoría</b>	Combustión Sólida	Combustión de Gas	Resistencia Eléctrica	Inducción magnética	Colectores solares	Compresor eléctrico
<b>Fuente</b>	Leña, carbón	GLP, GN	Hidroeléctricas, Termoeléctricas		Sol	Electricidad y ambiente
<b>Tipo</b>	Caldera de leña o carbón	Calefón	Ducha eléctrica	Ducha de Inducción	Colector solar de tubos de vacío	Bomba de calor
<b>Eficiencia</b>	40 – 60 %	55 – 80 %	65 %	95 %	50 -70 %	>200 %

Con la comparativa planteada en la Tabla 1.1, una bomba de calor resulta en un sistema atractivo para la producción de ACS por su alta eficiencia y aprovechamiento de la energía. La bomba de calor puede mejorar aún más su eficiencia al combinarse con sistemas de captación de energía solar, ya sea para suplir la demanda de energía eléctrica o para agregar energía a la fuente [9].

#### **1.5.2 Bombas de Calor**

Una bomba de calor es una máquina térmica, por ende, trabaja con un foco de temperatura caliente y un foco de temperatura fría (generalmente el ambiente). Una bomba de calor transfiere energía en forma de calor del foco frío hacia el foco caliente, por lo tanto, invierte la dirección natural del flujo de calor al agregar una pequeña cantidad de trabajo adicional que generalmente es en forma de energía eléctrica. Al ser una máquina térmica la bomba de calor se rige a las leyes de la termodinámica [8].

### **Clasificación de las bombas de calor**

Las bombas de calor se pueden clasificar según su fuente de absorción de energía [8]

**Bombas de calor con fuente de aire:** Este tipo de bombas de calor absorben energía del aire que circula en los alrededores, este aire se encuentra a condiciones ambientales. La bomba de calor puede absorber energía incluso de un aire ambiente a  $-15^{\circ}\text{C}$ . Este tipo de bombas se utilizan más en el sector residencial o aplicaciones de baja escala.

**Bombas de calor con fuente de agua:** El agua es por excelencia un fluido de alta inercia térmica y que puede almacenar una gran cantidad de energía, además es el fluido del que mayor accesibilidad se tiene después del aire. El uso de este tipo de bombas se da principalmente en edificios residenciales.

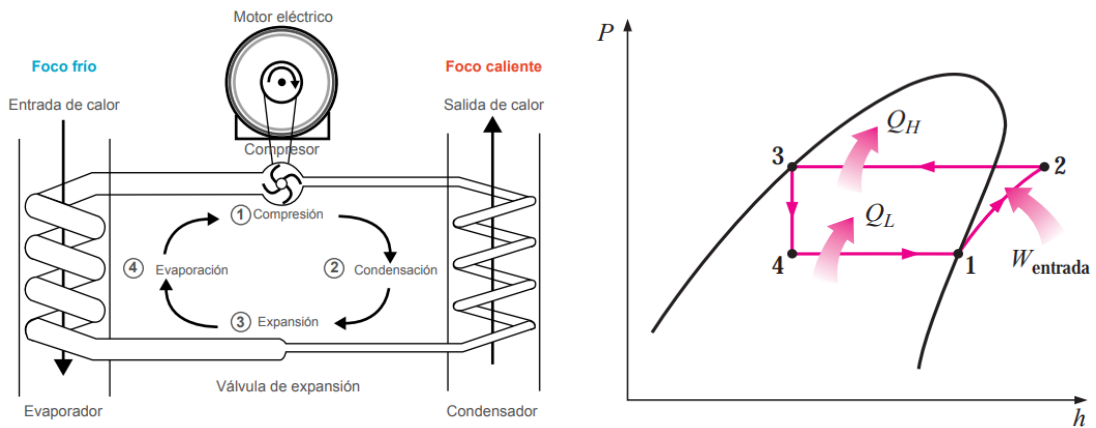
**Bomba de calor geotérmica:** La corteza terrestre capta y acumula la energía solar y al ser de gran profundidad tiene una alta inercia térmica, lo que le permite tener temperaturas muy estables sin importar la época del año y a una poca profundidad. Este tipo de bombas funciona con un fluido secundario para la transferencia de calor.

### **Funcionamiento y elementos principales de una bomba de calor**

En esta sección se describen las partes fundamentales de una bomba de calor de compresión de vapor mecánica accionada por un motor eléctrico.

La Figura 1.2 muestra el esquema (a) y el diagrama (b) del ciclo ideal de refrigeración por compresión de vapor, donde se pueden identificar 4 procesos que unen los 4 estados termodinámicos del fluido refrigerante que componen el ciclo y los 4 elementos principales para el funcionamiento: compresor, condensador, evaporador y válvula de estrangulamiento. A continuación, se describen los procesos asociados y su relación con los elementos para el funcionamiento de la bomba de calor.





**Figura 1.2** Esquema y Diagrama P-h del ciclo ideal de refrigeración por compresión de vapor [8], [10]

### Ciclo de refrigeración por compresión de vapor

Proceso de compresión de vapor (Estados 1-2): Se comprime el gas refrigerante desde un estado (1) de baja presión y temperatura hasta un estado (2) de alta presión y temperatura. El compresor es el encargado de transformar la energía eléctrica en trabajo de compresión sobre el fluido refrigerante.

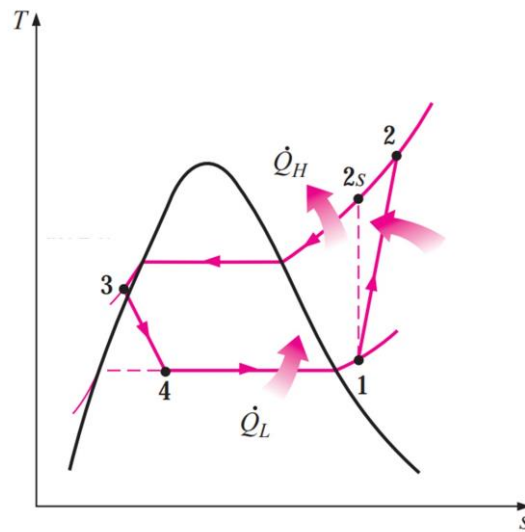
Proceso de condensación (Estados 2-3): Corresponde a la disipación de calor del gas refrigerante hacia el foco caliente a presión constante. El gas refrigerante pasa de estado gaseoso a estado líquido, e incluso a líquido subenfriado en el ciclo real. Este proceso es el principal y el de interés en una bomba de calor.

Proceso de expansión (Estados 3-4): Durante este proceso isentálpico el refrigerante en estado líquido y alta presión pasa por un dispositivo de estrangulamiento que disminuye su presión y por ende su temperatura y puede absorber calor del foco frío.

Proceso de evaporación: el gas refrigerante al estar a baja temperatura puede extraer calor del foco frío y, por ende, pasa de mezcla líquido-vapor a vapor saturado.

### Ecuaciones termodinámicas y coeficiente de desempeño para una bomba de calor

Las ecuaciones descritas en esta sección son desarrolladas mediante un balance energético del ciclo real de compresión de vapor, de un diagrama Temperatura-entropía (T-s), mostrada en la Figura 1.3, sin embargo, en este ciclo no se incluye las pérdidas de presión en tuberías y corresponde a una simplificación del ciclo real.



**Figura 1.3.** Diagrama T-s simplificado del ciclo real de compresión [10]

**Calor disipado – Proceso de condensación (Ecuación 1.1)**

$$q_{out} = h_4 - h_3$$

**Ecuación 1.1.** Calor disipado en el condensador

Donde:

$q_{out}$  Calor de salida  $\left[\frac{kJ}{kg}\right]$

$h_4$  Entalpía en el estado 4  $\left[\frac{kJ}{kg}\right]$

$h_3$  Entalpía en el estado 3  $\left[\frac{kJ}{kg}\right]$

**Calor absorbido – Proceso de evaporación (Ecuación 1.2)**

$$q_{in} = h_1 - h_6$$

**Ecuación 1.2.** Calor absorbido en el evaporador

Donde:

$q_{in}$  Calor de entrada  $\left[\frac{kJ}{kg}\right]$

$h_6$  Entalpía en el estado 6  $\left[\frac{kJ}{kg}\right]$

$h_7$  Entalpía en el estado 7  $\left[\frac{kJ}{kg}\right]$

## Trabajo del compresor

Existen distintos modelos para determinar el trabajo del compresor, cada uno de estos adaptándose según las consideraciones que rodean al proceso de compresión. Los tipos de compresión se describen a continuación.

- **Compresión isentrópica:** Se considera que no existe transferencia de calor con los alrededores y no existe fricción, por lo que la variación de entropía es cero.
- **Compresión isotérmica:** Existe transferencia de calor intencional con un fluido a sus alrededores para una disminución de trabajo de entrada requerido.
- **Compresión politrópica:** Considera un comportamiento entre el isentrópico e isotérmicos, sin embargo, se considera que no existen generación de calor y entropía por fricción, por lo que no se puede utilizar para determinar el trabajo real de un compresor con exactitud.

El proceso de compresión real incluye irreversibilidades las cuales afectan la eficiencia del compresor por lo que la entrada de trabajo no solo resulta en la variación de propiedades termodinámicas del refrigerante y resulta difícil cuantificar el trabajo del compresor mediante propiedades termodinámicas solamente.

### Eficiencia Isentrópica (Ecuación 1.3)

La eficiencia isentrópica es una medida que expresa de forma cuantitativa que tanto un proceso se acerca a un proceso isentrópico idealizado. La eficiencia isentrópica está dada por la Ecuación 1.3.

$$\eta_s = \frac{\text{Trabajo isentrópico}}{\text{Trabajo real del compresor}} = \frac{w_s}{w_a}$$

#### Ecuación 1.3. Eficiencia isentrópica

Donde:

$\eta_s$  Eficiencia isentrópica  $\left[ \frac{kJ}{kg} \right]$

$w_s$  Trabajo específico isentrópico  $\left[ \frac{kJ}{kg} \right]$

$w_a$  Trabajo específico real del compresor  $\left[ \frac{kJ}{kg} \right]$

El trabajo isentrópico por unidad de masa de refrigerante está dado por la Ecuación 1.4.

$$w_s = (h_{2s} - h_1)$$

**Ecuación 1.4.** Trabajo específico isentrópico del compresor

Donde:

$h_{2s}$  Entalpía en el estado 2s  $\left[\frac{kJ}{kg}\right]$

$h_1$  Entalpía en el estado 1  $\left[\frac{kJ}{kg}\right]$

Si bien la eficiencia isentrópica solo tiene sentido en compresores adiabáticos, la transferencia de calor en el proceso de compresión real es casi insignificante, por lo que, la eficiencia isentrópica está relacionada a las irreversibilidades del diseño del compresor y es de gran utilidad para determinar el trabajo real de un compresor si se conoce su desempeño isentrópico. Los mejores compresores tienen eficiencias isentrópicas que van de 80 a 90 por ciento [10].

**Coefficiente de rendimiento – COP (Ecuación 1.5)**

$$COP_{HP} = \frac{\text{Salida deseada}}{\text{Entrada requerida}} = \frac{Q_{out}}{W_{neto,in}}$$

**Ecuación 1.5.** COP para una bomba de calor

Donde

$COP_{HP}$  Coeficiente de desempeño para una bomba de calor

$Q_{out}$  Calor de salida  $[kJ]$

$W_{neto,in}$  Trabajo neto de entrada  $[kJ]$

Cabe mencionar que el trabajo neto de entrada para una bomba de calor, accionada por un compresor eléctrico, es el consumo de energía eléctrica que consume el compresor.

El trabajo de entrada neto se puede medir, y calcular mediante la Ecuación 1.6.

$$\dot{W}_{net,in} = V \cdot I$$

**Ecuación 1.6.** Potencia eléctrica del compresor

Donde

$\dot{W}_{net,in}$  Potencia de entrada consumida por un compresor eléctrico  $[W]$

$V$  Voltaje medido experimentalmente  $[V]$

$I$  Corriente medida experimentalmente  $[A]$

### 1.5.3 Refrigerantes

El refrigerante es el fluido de trabajo de la bomba de calor encargado de transportar el calor desde el foco frío y transferirlo al foco caliente en el ciclo de refrigeración [10]. Los primeros refrigerantes y más utilizados hasta antes de los años 30 fueron sustancias puras, estas eran Amoníaco, Propano y Dióxido de carbono; las cuales eran peligrosas, debido a la tecnología de la época, e incluso inflamables y tóxicos [11]. Debido a esta problemática se optó por sustancias halógenas o flouorcarbonos que posteriormente serían demostradas como causantes del deterioro de la capa de ozono y calentamiento global [12].

En el contexto actual, el deterioro de la capa de ozono y el calentamiento global se debe a procesos físicos y químicos que ocurren en la atmósfera [15]. Todo esto ha contribuido en la búsqueda continua de reemplazos para los refrigerantes que tengan un potencial de calentamiento global (GWP) alto, haciendo voltear a alternativas del pasado como refrigerantes de sustancias puras o nuevos refrigerantes.

#### **Indicadores ambientales y clasificación de seguridad de los refrigerantes más usados**

Actualmente los parámetros para elegir un refrigerante se basan en el potencial de calentamiento, potencial del deterioro de capa de ozono, y su seguridad y toxicidad [11].

- **Ozone Depletion Potential (ODP):** El potencial de agotamiento de la capa de ozono es la relación entre el impacto que tiene un refrigerante o sustancia química sobre la capa de ozono y el impacto de una masa idéntica de CFC-11 [12].
- **Global Warming Potential (GWP):** El potencial de calentamiento global es una medida de la capacidad que tienen los gases de efecto invernadero para atrapar calor de la atmósfera (eficiencia radiativa). Su potencial es comparado con la del CO<sub>2</sub> [12].
- **Clasificación de seguridad-ASHRAE 34: Los refrigerantes se pueden clasificar por su inflamabilidad en tres grupos, descritos en la norma ASHRAE 34.** Esta clasificación se basa en el LII (Límite Inferior de Inflamabilidad) a condiciones de temperatura ambiente y presión atmosférica [13].

Además, los refrigerantes se dividen en dos grupos según su toxicidad, y su clasificación están en base a al tiempo de exposición máximo que describe la ASHRAE 34 [13].

## Refrigerantes más utilizados en bombas de calor

Las bombas de calor, al igual que otros dispositivos de calentamiento, han sufrido adaptaciones tecnológicas bajo los márgenes políticos, ambientales y de investigación, por ende, se han utilizado todo tipo de refrigerantes en las bombas de calor a lo largo del tiempo. En la Actualidad, los refrigerantes más comunes son: R-134a, R-410A, R-407C, R-1234yf, R-454C, R-32 [14]. Aunque refrigerantes como el R-410A, R-134a, R-407C, son de los refrigerantes más comunes a nivel comercial, los R-290 y R-1234yf son los candidatos a ocupar su lugar en poco tiempo, ya que su rendimiento es mejor y tienen menor impacto ambiental.

En la Tabla 1.2 se muestran los refrigerantes más utilizados en bombas de calor y los potenciales reemplazos.

**Tabla 1.2.** Refrigerantes más utilizados en bombas de calor [12], [13]

Nombre genérico del Refrigerantes	ODP	GWP	Seguridad
<b>HIDROCARBUROS</b>			
Propano (HC – 290)	0	3	A3
Isobutano (HC – 600a)	0	3	A3
<b>HIDROFLUORCARBONADOS</b>			
HFC – 134a	0	1430	A1
HFC – 410A	0	2087.50	A1
HFC – 407C	0	1773.85	A1
<b>HIDROFLUOROLEFINAS</b>			
HFO – 1234yf	0	0	A2L
HFO – 1234ze	0	0	A2L

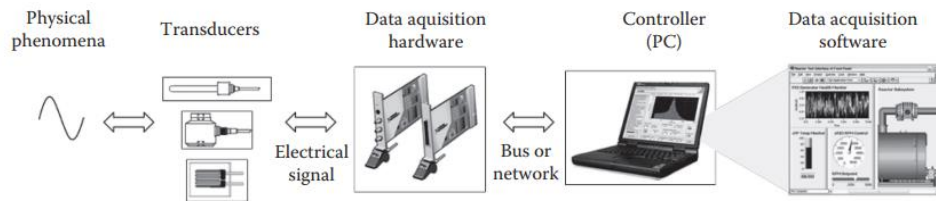
### 1.5.4 Sistemas de adquisición de datos (SAD)

Actualmente, bajo la supervisión de organismos internacionales como OLADE, Ecuador tiene compromisos de mejora en el uso de la energía y reemplazo de equipos ineficientes [15].

En este contexto, la renovación y actualización de equipos más eficientes para Ecuador exige que existan laboratorios y centros de estudio capaces de monitorear el comportamiento de los nuevos sistemas utilizando herramientas sofisticadas como un sistema de adquisición de datos.

Un sistema de adquisición de datos (SAD) permite que la medición de datos experimentales extraídos con instrumentos de medición (electrónicos) sean leídos,

almacenados, registrados y/o analizados mediante un entorno programable [16]. En la práctica es común utilizar los términos SAD y DAQ (Adquisición de datos) indistintamente, sin embargo, SAD hace referencia a todo el sistema y DAQ al proceso de adquisición de datos.



**Figura 1.4.** Diagrama de bloques de un SAD [16]

Como se muestra en la Figura 1.4, el DAQ es el corazón de un SAD, sin embargo, necesita de otros elementos para su funcionamiento los cuales son:

- a) **Fenómeno físico**
- b) **Transductores**
- c) **Proceso de adquisición de datos (DAQ)**
- d) **Controlador o software**

En estas secciones se detallan los elementos de un SAD, siendo el elemento en el que se enfoca el proyecto el controlador o software.

### **Fenómeno físico y teoría de señales**

El fenómeno físico corresponde al primer elemento de un SAD, y es la entrada principal para un sistema de adquisición de datos, pues es la señal el objeto sobre el cual se analizará y controlará un SAD.

Una señal es una abstracción matemática respecto a una magnitud física, la cual varía en función de las variables independientes como tiempo y espacio. El fenómeno físico se puede modelar matemáticamente para obtener una respuesta que representa la medición de la magnitud física que se quiere conocer, es decir, el fenómeno físico es la entrada y la salida es una medida [17].

### **Transductores**

Un transductor es el dispositivo electromecánico que transforma la señal de una magnitud física (presión, temperatura, fuerza) en una forma de energía eléctrica [18]. Es decir, el transductor es el sistema que transforma la señal de entrada y entrega una señal de salida.

Un sensor, es un transductor que sirve para medir una magnitud física, a diferencia del concepto de transductor, un sensor es capaz de interpretar la señal de salida y cuantificar el fenómeno físico a medir como voltaje, corriente, temperatura y presión [18].

### **Sensores para medición de Temperatura**

La temperatura es la magnitud física más común dentro de los parámetros de medición en equipos de laboratorio, los tipos de sensores más comunes son de termopares y resistivos.

### **Dispositivos de Medición de Presión**

La presión, en termodinámica, es una propiedad del fluido y es la segunda variable más importante en el monitoreo de sistemas energéticos [19]. Existe una variedad de dispositivos para la medición de presión basados en distintos principios físicos. Los dispositivos se pueden clasificar en mecánicos, elásticos y electrónicos/electromecánicos y los principales tipos son [19].

### **Proceso de adquisición de datos**

El DAQ se define como el proceso por el que un fenómeno físico del mundo real se almacena y registra en formato digital [20].

El Proceso de adquisición de datos es el corazón de todo SAD, pues se encarga de la transformación de la señal analógica en un formato digital o viceversa. Esta conversión continua la realiza el hardware de adquisición de datos.

### **Hardware de adquisición de datos**

El hardware DAQ puede tener distintas formas y aplicaciones. La mayoría de estos dispositivos no solo realizan la transformación analógica digital, también generan salidas analógicas y digitales e incluyen la capacidad de receptar señales digitales provenientes de otros dispositivos.

El hardware DAQ se puede dividir en 3 elementos fundamentales: circuito de acondicionamiento de señales, circuito de conversión de señales (analógico-digital) y bus de transferencia de datos hacia el controlador (PC)

### **Microcontroladores**

Los microcontroladores pueden procesar y manejar muchas entradas y salidas, y también se usan como dispositivos para DAQ. ARDUINO es el más conocido de esta categoría.



## Software de control y adquisición de datos

El software es la herramienta encargada de controlar y administrar los recursos de una computadora, mostrar los datos y sobre todo permite la interpretación y análisis de estos [21]. Por lo que, un software puede manejar, almacenar y ayudar a interpretar los datos entregados por el hardware DAQ.

## Interfaces gráficas

Como se mencionó, el software es la herramienta informática que permite al usuario el control y monitoreo de las señales, la interacción se puede dar mediante texto o una interfaz gráfica. Se denomina Interfaz Gráfica de Usuario (GUI) a todo software que admite la interacción del usuario con el sistema operativo mediante elementos visuales [22].

Algunos ejemplos de software con interfaz gráfica para análisis de sistemas energéticos se muestran en la Tabla 1.3.

**Tabla 1.3.** Interfaces gráficas para análisis de sistemas energéticos

GUI	Programación	Descripción	Ingreso de datos
Refriterm [23]	MATLAB	Análisis de ciclos de refrigeración	Manual
Software R290 [24]	C++	Diseño de plantas de refrigeración	Manual
Cooling Cycle Tester [25]	Python	Análisis de sistemas de refrigeración.	DAQ/ ARDUINO
Perfiles de temperaturas en hornos [26]	LabVIEW	Adquisición de datos para la generación de los perfiles de temperatura en hornos	DAQ/ NI 6002

### 1.5.5 Lenguajes de programación

Los lenguajes de programación cumplen un rol importante en el desarrollo de interfaces gráficas de usuario, vinculadas o no a un sistema de adquisición de datos.

Un lenguaje de programación es un sistema de notación que describe acciones y cálculos que se quiere que realice la computadora de forma legible, tanto para la computadora como para el ser humano. Se puede distinguir 5 generaciones durante la evolución de los lenguajes de programación [27].

- Primera generación: Conocido como lenguaje de máquina, donde su notación se basa en ceros y unos y es el lenguaje natural de las computadoras. Esta notación es difícil de entender para las personas.
- Segunda generación: se les denomina lenguaje ensamblador. En este lenguaje se hace uso de reglas mnemotécnicas que permiten la comunicación de forma más sencilla, aunque aún es muy básica y con pocas instrucciones.
- Tercera generación: Denominados lenguajes de alto nivel. Son notaciones aún más legibles para el ser humano, sin embargo, requieren de compiladores que traduzcan entre este nivel de notación y secuencias legibles para la computadora. Estos compiladores son programas complejos que han permitido el desarrollo y evolución del software. Los lenguajes más comunes son: FORTRAN, C++, PYTHON, JAVA y HTML.
- Cuarta generación: En esta generación se encuentran lenguajes como SQL, ABAP o NATURAL. Su propósito no es el desarrollo de aplicaciones o software, ya que están pensados para solucionar problemas específicos como manejo de bases de datos en el caso de SQL.
- Quinta generación: Son los lenguajes utilizados en el sector de la inteligencia artificial. Se los utiliza mediante restricciones orientadas a solucionar un problema que se le indica al sistema.

Los lenguajes de programación de tercera generación son los que permiten el desarrollo de aplicaciones, software e interfaces gráficas.

## **Python**

Python es un lenguaje de programación versátil y robusto fácil de leer y entender, es de software libre y tiene una gran comunidad alrededor de todo el mundo, lo que le permite que su desarrollo sea continuo y cada vez cuenta con funciones y librerías más amplias. Cuenta con características como herencias, librerías, distintos paradigmas de programación, etc. Que le permitió competir con lenguajes como C y Java desde sus inicios [28].

Python es considerado un lenguaje de programación orientado a objetos puro, pues todas sus funciones se basan en métodos y objetos.

## **Programación orientada a objetos**

La programación orientada a objetos (POO) es un paradigma de resolución de problemas en programación. Un paradigma en programación es un conjunto de teorías aceptadas por todos los integrantes que la utilicen y sirve de guía y modelo para resolver problemas [29].

La programación orientada a objetos se basa en encapsular objetos o identidades agrupando datos y comportamientos principales de un programa. Al encapsulamiento de las características y comportamiento de un objeto se le denomina clase. Está pensada con el enfoque de “modelar la realidad”, los objetos son abstracciones de la realidad que representan diferentes partes de la solución y que se comunican entre sí mediante mensajes. POO es la forma más práctica para el desarrollo de GUI, pues se abstrae objetos visuales que interactúan con el usuario y los encapsula en clases y les asigna un comportamiento específico según su interacción con el usuario [30].

## **2 METODOLOGÍA**

Con el fin de cumplir con los objetivos del presente proyecto, se plantea una metodología teórica-práctica modulada en 4 etapas que se pueden apreciar en la Figura 2.1. A continuación, se describe el contenido de cada etapa en forma de resumen.

### **Etapas 1: Revisión bibliográfica**

Esta etapa abarca una revisión bibliográfica de artículos y trabajos relacionados y similares al presente proyecto, además se define los conceptos fundamentales asociados al funcionamiento de una bomba de calor, los cuales servirán como material para el cuestionario, cálculo de variables termodinámicas y obtención del desempeño de la bomba de calor mediante las ecuaciones descritas en el Capítulo 1 del presente proyecto.

### **Etapas 2: Determinación de parámetros**

La determinación de parámetros está basada en revisión bibliográfica que permita determinar la ubicación de sensores y variables termodinámicas necesarias para poder monitorear y cuantificar el desempeño de la bomba de calor que forma parte del sistema didáctico propuesto. Además, se detallan las especificaciones técnicas de los componentes del sistema didáctico preseleccionado para así proceder a una correcta selección del lenguaje de programación compatible con los dispositivos preseleccionados.

### **Etapas 3: Selección del lenguaje de programación**

Se selecciona el lenguaje más apropiado después de una revisión bibliográfica de los lenguajes más utilizados para desarrollo de software, según requerimientos del proyecto y criterios de selección personales. El método de elección corresponde al método ordinal corregido de criterios ponderados. Después, se verifica que el lenguaje de programación sea capaz de manejar y manipular la información entregada por el dispositivo DAQ.

### **Etapas 4: Desarrollo del software e Interfaz Gráfica**

Una vez seleccionado el lenguaje de programación, se procede al desarrollo del software mediante la programación de código en un entorno de desarrollo que permita cumplir con los objetivos del proyecto. Finalmente, se determinan los parámetros de diseño de la GUI mediante las Normas ISO 9241 para que la GUI esté orientada al usuario, un fácil manejo y que sea intuitiva. Una vez desarrollados los códigos, estos se integran todos para obtener un software capaz de monitorear las variables termodinámicas establecidas en la Etapa 2 y la verificación de que el software funciona correctamente y cumple con los objetivos planteados.

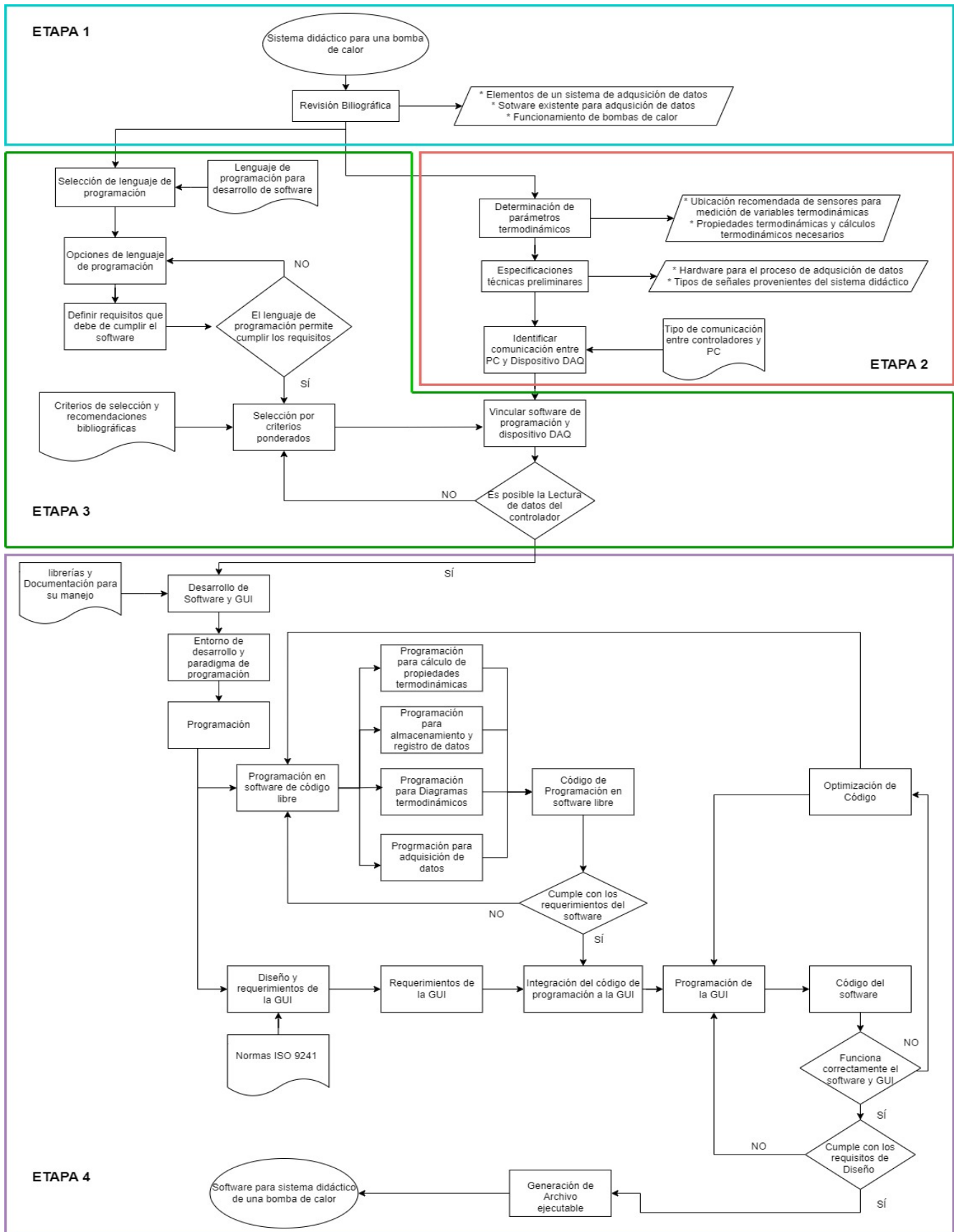


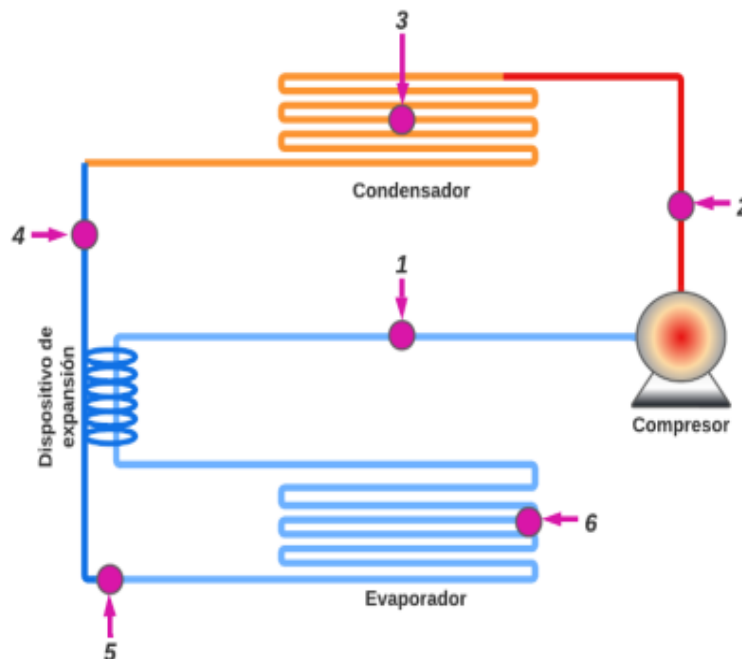
Figura 2.1. Diagrama de flujo de la metodología implementada

## 2.1 Determinación de parámetros de análisis de la bomba de calor

### 2.1.1 Ubicación de sensores según bibliografía

Mediante la revisión bibliográfica de proyectos similares y equipos de laboratorio destinados a la evaluación energética de sistemas de refrigeración y calefacción basados en compresión de vapor, se tiene que, la ubicación más apropiada de los sensores corresponde a la entrada y salida de cada elemento principal que conforma el ciclo de refrigeración por compresión de vapor (ver Sección 1.2.2).

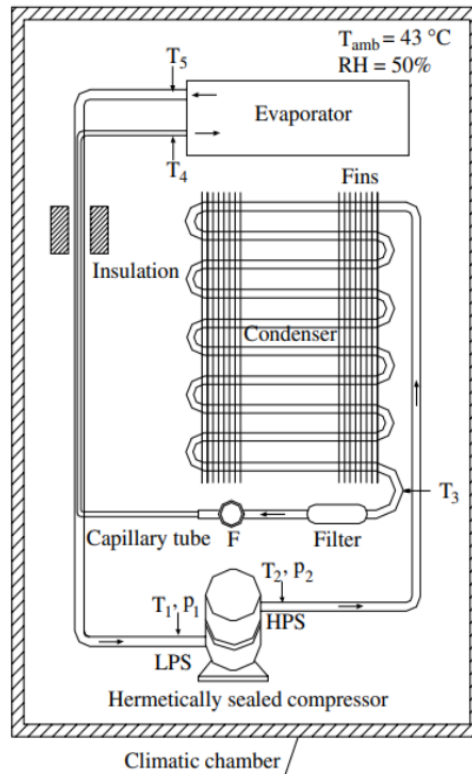
Por ejemplo, para el proyecto desarrollado para análisis de sistemas de refrigeración, la ubicación de la instrumentación estuvo basada en referencias bibliográficas y en ensayos experimentales, donde se optó por 6 puntos de medición de temperatura para poder monitorear el comportamiento de sistemas de refrigeración. En la Figura 2.2 se muestran las ubicaciones de los sensores. La ubicación de los sensores 3 y 6 se debe principalmente para conocer la temperatura de condensación y evaporación del refrigerante donde el punto medio es la mejor ubicación según [25].



**Figura 2.2.** Ubicación de sensores para análisis de sistemas de refrigeración por compresión [25]

Otro trabajo corresponde a un equipo experimental que consta de 5 puntos de medición de temperatura y 2 puntos de medición de presión a la entrada y salida del compresor.

En la Figura 2.3 se muestra el esquema de los puntos de medición, donde se puede notar que existe un punto de medición adicional comparado al ciclo de refrigeración ideal, esto se debe al intercambiador de calor que existe entre la línea de entrada y salida del evaporador, y resulta en un elemento extra a los componentes principales y por ende en un punto extra de medición.



**Figura 2.3.** Diagrama de la configuración del equipo experimental de un ciclo de refrigeración [31]

Existen muchos ejemplos adicionales de equipos de laboratorio con distintas ubicaciones de sensores e instrumentación, esto se debe a que cada equipo tiene su alcance y responde a una necesidad específica. La ubicación de los sensores se aconseja a la entrada y salida del compresor, ya que es el dispositivo más accesible para medir presión y temperatura en la mayoría de los equipos; los puntos de medición son los principales del ciclo de refrigeración por compresión de vapor y son los mínimos necesarios.

### 2.1.2 Variables termodinámicas

Las variables termodinámicas relevantes para el proyecto están basadas principalmente en el postulado de estado.

## **Postulado de estado**

Un ciclo termodinámico se describe a través de estados termodinámicos. Un estado a su vez se describe mediante sus propiedades, sin embargo, no se requiere de todas sus propiedades para poder especificar un estado. El postulado de estado indica que se puede especificar por completo el estado de un sistema compresible simple mediante dos propiedades intensivas independientes [10].

Así, las variables termodinámicas más fáciles de medir con instrumentación corresponden a presión y temperatura en las ubicaciones anteriormente identificadas y que cumplan con el postulado de estado.

## **Cálculo de propiedades y desempeño de la bomba de calor**

El cálculo de propiedades termodinámicas del refrigerante permite conocer el desempeño de la bomba de calor, por lo que, se obtendrán las propiedades mediante software, el cual pueda leer propiedades termodinámicas de distintos refrigerantes, por ejemplo, CoolProp.

Finalmente, el desempeño de la bomba de calor se obtendrá mediante las propiedades del refrigerante haciendo uso de las ecuaciones descritas en la Sección 1.2.3. y el cálculo de la potencia eléctrica mediante sensores de voltaje y corriente para la obtención del COP.

### **2.1.3 Descripción del equipo – Especificaciones técnicas**

Un sistema didáctico no siempre es un SAD, sin embargo, para el presente proyecto el sistema didáctico del laboratorio de termodinámica, que se desea implementar, corresponde a un SAD y se utilizará de forma indistinta ambos términos en este proyecto.

La descripción del equipo está organizada en términos de un sistema de adquisición de datos para un mejor entendimiento (ver Figura 1.4). En la Tabla 2.1 se especifica el componente preseleccionado de cada uno de los elementos del SAD, las cuales fueron especificadas en su mayoría durante la proforma para el financiamiento por parte de ASHRAE en la convocatoria abierta para equipos de laboratorio para universidades.

Como se observa en la Tabla 2.1 y en la Figura 1.4. el software es la parte final del SAD y es el componente a desarrollar en el presente proyecto. Es importante notar que el microcontrolador preseleccionado es compatible con las opciones de lenguaje de programación descritas en las siguientes secciones.



**Tabla 2.1.** Especificaciones preliminares del sistema didáctico a implementar en el laboratorio de termodinámica EPN.

Bomba de calor domiciliaria 300 litros Marca: Konen distribuido por Technova Sol Tipo: A01H-300L.OIH (Aire-Agua) Modelo: CKXRS-7.0IH Refrigerante: R - 410A		
Elemento SAD	Elemento para el banco de pruebas de bomba de calor	Selección preliminar
Fenómeno físico y Transductor	Presión	Transductor de presión SB69
	Temperatura	Módulo de temperatura MAX31865
	Voltaje	Sensor de voltaje AC ZMPT 101B
	Corriente	Sensor de corriente ACS712-30A
Hardware DAQ	Microcontrolador	Arduino Mega 2560
Software de adquisición de datos	Software con Interfaz Gráfica	Componente de desarrollo

## 2.2 Selección del lenguaje de programación

En la presente sección se detalla el procedimiento de selección del lenguaje de programación para el desarrollo de la interfaz gráfica. El lenguaje de programación debe de tener herramientas para el análisis termodinámico, proporcionar un monitoreo en tiempo real, etc. Y debe permitir el desarrollo de una interfaz gráfica para la interacción con el usuario.

Dentro de esta sección se analizará solamente lenguajes de programación de tercera generación. Los lenguajes de programación de tercera generación son conocidos como lenguajes de alto nivel, se caracterizan por su notación legible para el ser humano y el uso de compiladores, que hacen la tarea de traducción hacia lenguaje de máquina, y permiten el desarrollo de software y su evolución [27]. Por ende, la tercera generación es la indicada para establecer un lenguaje de programación adecuado para el desarrollo del software, y los lenguajes más comunes son:

- Java
- C++, C#
- Python
- Visual basic.NET

En esta sección se analizarán estos lenguajes de programación, aunque existan más lenguajes que puedan cumplir con los requisitos del software, los analizados en esta sección tienen una comunidad más amplia, son de software libre y se usan en proyectos similares.

### **2.2.1 Opciones de lenguajes de programación**

#### Visual Basic.NET

Visual Basic.Net es la última versión de Visual Basic (VB) y de su familia BASIC. La filosofía de esta familia se basa en un diseño de fácil uso, lo que permitió a múltiples usuarios y comunidades desarrollar todo tipo de proyectos. Visual Basic fue muy popular en la década de los 70 y 80 porque tenía la popularidad de ser un lenguaje fácil de aprender y era compatible con computadoras de baja memoria. VB proporcionó a sus usuarios una forma sencilla de construir interfaces gráficas de usuario (GUI). Con la implementación de VB.NET, los usuarios pueden usar el paradigma de programación orientada a objetos como resolución de problemas, lo que la hace más atractiva para los usuarios [32].

#### C++, C#

Son el lenguaje resultado de la evolución de C. C++ fue desarrollada en 1980, consta con mejores características imperativas y cuenta con soporte para programación orientada a objetos. C++ fue desarrollada por Microsoft, es software libre y es muy utilizado para interfaces gráficas, aunque menos que VB.NET. C# es basado en C# y Java e incluye conceptos y características de Visual Basic [32].

#### Java

Se desarrolló en 1990. Aunque Java se inspiró en C++, sus diseñadores mejoraron sus características, haciéndolo más ligero, fiable y simple. Además, Java es el pionero en la programación orientada a objetos, lo que la hizo popular entre sus usuarios. Las aplicaciones de Java son muy variadas, siendo el desarrollo Web una de las más populares [32].

#### Python

Es un lenguaje de scripting. Scripting se refiere al tipo de lenguaje de programación que requiere de un intérprete que procesa los comandos y da los resultados esperados. A diferencia de un lenguaje de programación convencional como C++ y Java, un lenguaje scripting no requiere de un compilador, pues todo el script se encuentra en un hilo conductor y se evita la gestión manual de almacenamiento. Este tipo de lenguaje facilita la

programación de software y la hace más directa y con menos texto fuente, aunque puede llegar a ser menos eficiente y ser propenso a errores por la poca participación del usuario en la gestión de almacenamiento [28]. En la Figura 2.4 se muestra una comparativa de un script en Java y en Python, donde se puede notar la diferencia de programación.

```
// Ejemplo "Hola mundo" en Java
class HolaMundo {
    static public void main( String args[] ) {
        System.out.println( "¡Hola, mundo!" );
    }
}

# Ejemplo "Hola, mundo" en Python
print "¡Hola, mundo!"
```

**Figura 2.4.** Comparativa de programación entre el lenguaje Java y Python [34]

Esta ventaja de simplificación de código fuente, reduce el trabajo para el programador, pero aumenta el trabajo para el procesador [33]. Es recomendable el uso de Python para aplicaciones pequeñas y medianas, muchos programadores utilizan Python para desarrollar aplicaciones de escritorio mediante bibliotecas de GUI [30].

### **2.2.2 Requisitos del software para el sistema didáctico de la bomba de calor para el laboratorio de la FIM-EPN**

Los requisitos del lenguaje de programación están basados en los objetivos del presente proyecto y deben permitir el análisis adecuado de la bomba de calor para el equipo de laboratorio de la FIM-EPN. Los requisitos principales son:

- Lenguaje de programación de software libre
- Capaz de leer tablas de propiedades termodinámicas
- Lectura, monitoreo y registro en tiempo real de las variables termodinámicas
- Desarrollo de una interfaz gráfica

Todos los lenguajes de programación cumplen a cierta medida los requerimientos. Todos los lenguajes de programación son de software libre, la librería con tablas termodinámicas de CoolProp está disponible para todos estos lenguajes (ver [34]), todos son capaces de leer variables termodinámicas provenientes de un dispositivo de adquisición de datos (Arduino para este proyecto) y todos son capaces de desarrollar interfaces gráficas y software ejecutable para las computadoras del laboratorio de la FIM-EPN. Sin embargo, la

sintaxis, paradigmas de resolución y compilación varía de lenguaje a lenguaje, por lo que es necesario establecer criterios personales para una selección adecuada.

### 2.2.3 Criterios para selección del lenguaje de programación

Los criterios de evaluación globales, establecidos de forma personal, para la selección del lenguaje de programación son los siguientes:

- Manejo del lenguaje
- Paradigmas de resolución permitidos
- Sencillez del lenguaje y facilidad de aprendizaje
- Versatilidad en aplicaciones

El Método Ordinal Corregido de Criterios Ponderados, propuesto por Carles Riba [35], se puede aplicar cuando se tiene distintas opciones para desarrollar un diseño, este método toma en cuenta los criterios de evaluación globales y no considera las características de las alternativas o sus propiedades [34]. Para este caso, las alternativas son las opciones de lenguaje de programación descritas en la sección 2.4.1.

Tras establecer los criterios de evaluación, mediante el Método Ordinal Corregido de Promedios Ponderados, se selecciona la mejor alternativa para lenguaje de programación.

En el **ANEXO I** se puede ver el resto de las tablas de cada criterio de selección.

#### Conclusión de Criterios Ponderados para las Alternativas

En la Tabla 2.2 Se concluyen las alternativas y su ponderación final para cada criterio de evaluación, por lo que se puede seleccionar un lenguaje según estos criterios.

**Tabla 2.2.** Conclusión método ordinal corregido de criterios ponderados para selección de lenguaje de programación

Conclusión						
Criterio	Manejo del lenguaje	Paradigmas	Sencillez y facilidad	Versatilidad	$\Sigma+1$	Prioridad
Python	0,16	0,12	0,0525	0,045	0,3775	1
Visual Basic.NET	0,08	0,045	0,0525	0,0225	0,2	3
C++, C#	0,08	0,045	0,0225	0,0225	0,17	4
Java	0,08	0,09	0,0225	0,06	0,2525	2

El Método Ordinal Corregido de Criterios Ponderados ayuda a la selección de una alternativa y sirve de guía para analizar las opciones, pero la selección final siempre la considera el diseñador.

Se selecciona Python como lenguaje de programación, debido no solo a la conclusión de la Tabla 2.2, sino también por la cantidad de información y guía en internet sobre el desarrollo de software e interfaces gráficas con Python.

## **2.3 Desarrollo del software y diseño de la GUI**

Una vez establecido el lenguaje de programación, hay que recopilar las librerías y herramientas para desarrollar el software y diseñar la interfaz gráfica.

El software hace referencia a todos los datos y programas que se ejecutan en una unidad de procesamiento, que puede ser una computadora o cualquier dispositivo electrónico. El software incluye en su código todos los algoritmos e instrucciones de las funciones y tareas que se debe realizar, que pueden ejecutarse con comandos de texto o mediante una interfaz gráfica [36].

Una interfaz gráfica de usuario es una forma de interacción entre el usuario y el software a través de objetos visuales como botones, ventanas, íconos etc. Lo que permite que la interacción sea más amigable e intuitiva.

### **Programación Orientada a Objetos (POO)**

La programación orientada a objetos es el paradigma de resolución más implementado para el desarrollo de software, debido a que su formulación es la más parecida al mundo real, además permite la reutilización de código, lo que hace más productivo el desarrollo de código [36]. Por eso, y la cantidad de material sobre el desarrollo de software mediante POO, se ha seleccionado como paradigma de resolución para el proyecto, pero Python permite aplicar varios paradigmas a la vez, y se pueden usar otros según la necesidad.

### **Entorno de desarrollo para software**

El IDE es un programa con interfaz gráfica que ayuda al desarrollo de programas en Python y contiene múltiples herramientas. Las siglas *IDE* en español significan Ambiente de desarrollo integrado, en este se puede escribir instrucciones de Python y ver resultados sobre la marcha, además de funciones propias de cada IDE [30].

Existen diversos *IDE*'s para Python y su selección es simple preferencia del programador y las características que ofrece según la necesidad del proyecto.

Para este proyecto se ha optado por utilizar Spyder como entorno de desarrollo integrado.

### **2.3.1 Librerías para Python**

Durante la instalación de Python se instalan librerías básicas, pero siempre es posible instalar librerías adicionales según las necesidades de cada proyecto. Existen librerías desarrolladas por distintas entidades para simplificar tareas y efectuar funciones de forma más sencilla mediante comandos simples [29].

A continuación, se describen las librerías principales utilizadas para el desarrollo del presente proyecto.

#### ***Tkinter y CustomTkinter***

Ambas librerías funcionan para el desarrollo y creación de aplicaciones de escritorio; facilita la creación de interfaces gráficas de usuario GUI [37]. *Tkinter* está incluida en el paquete de instalación de Python. *CustomTkinter* está basada en *Tkinter*, provee de *widgets* más modernos y personalizables, sin embargo, aún está en desarrollo por lo que no tiene gran cantidad de *widgets* como *Tkinter*. Ambas librerías se usaron para obtener una GUI con todos los *widgets*, pero con un aspecto y personalización más modernos.

#### ***Threading***

Este módulo permite la construcción de interfaces de hilado multinivel sobre el hilo principal de Python [37]. La creación de hilos permite que se ejecuten tareas en paralelo, mejorando y optimizando los programas, sin embargo, se debe de tener cuidado en el uso de hilos, ya que, estos no pertenecen al hilo principal y no se los puede manipular mediante consola fácilmente. Un hilo solo se termina al acabar la tarea que se ejecuta en este. Existen varios módulos que ayudan a gestionar los hilos y datos compartidos entre hilos como *semaphore* y *queue*.

#### ***PySerial***

Es una librería que permite la comunicación con periféricos a través de comunicaciones por serial (USB) [38]. *PySerial* hace posible que la GUI obtenga los datos del dispositivo DAQ en tiempo real, siempre y cuando, el serial se configure según la velocidad de baudios del dispositivo DAQ y el puerto conectado (COM #) en el código de Python.

#### ***Collections***

Es un módulo especializado y utilizado para contener o encapsular diferentes tipos de datos como listas, diccionarios, arreglos, tuplas, etc. [39]. Tiene varias funciones que permite manipular el contenido de una colección según la necesidad.

La clase de la librería *collections* utilizada en el presente proyecto es *deque*, el cual facilita agregar y eliminar datos comparado a las funciones originales de Python como *append* y *pop*.

### ***Numpy***

La ventaja de *Numpy* respecto a las librerías por defecto de Python, es que el procesamiento de datos es mucho más rápido y también permite el manejo de vectores y matrices mucho más grandes [39]. También, provee de funciones para que la manipulación de datos sea mucho más sencilla y directa para el programador.

### ***CoolProp***

Es una librería, desarrollada originalmente en C++, que implementa: Ecuaciones de estado y propiedades de transporte para más de 122 sustancias, propiedades termodinámicas, funciones para resolución de problemas de mezclas etc. [34]. Esta librería es la que se utilizará para la obtención de propiedades termodinámicas del refrigerante de la bomba de calor.

### ***Fluprodia***

Es una librería destinada a graficar diagramas termodinámicos. Su funcionamiento se basa en la obtención de propiedades termodinámicas mediante *CoolProp*, de manipulación de *arrays* mediante *Numpy* y finalmente visualización de los diagramas mediante *Matplotlib* [43].

### ***Matplotlib***

Es una librería destinada para crear visualizaciones estáticas, interactivas o animadas en Python [39]. Hace posible que se puedan obtener figuras y gráficas mediante instrucciones en el editor, es muy utilizada a la par con *Numpy*.

Para el desarrollo de este proyecto también se utilizaron librerías “secundarias” que solo se utilizaron pocas veces y tienen poca relevancia, aunque sean necesarias. A continuación, se realiza una descripción breve de estas librerías para que el lector entienda de mejor manera el desarrollo de la interfaz gráfica y como se puede obtener el resultado final. Se puede obtener más información en [40].

- *Statistics.* - Permite funciones estadísticas, está inspirada en Minitab.
- *Time.* – Módulo desarrollado para funcionalidades con el tiempo.
- *Os.* - Permite utilizar funcionalidades del sistema operativo.
- *Pillow.* - Módulo desarrollado para el manejo de imágenes en Python.
- *Shutil.* - Proporciona herramientas para el almacenamiento de archivos.
- *Pyinstaller.* – Librería desarrollada para obtener archivos ejecutables (.exe).

En las siguientes secciones se explicará la programación más importante para el desarrollo del software. Se comentará la sintaxis y algoritmos más importantes del programa y el uso de librerías descritas en esta sección

### **2.3.2 Programación para la adquisición de datos en Python**

Para el proceso de adquisición de datos provenientes del Arduino, es necesario establecer una conexión entre la computadora y el Arduino mediante una comunicación serial (USB).

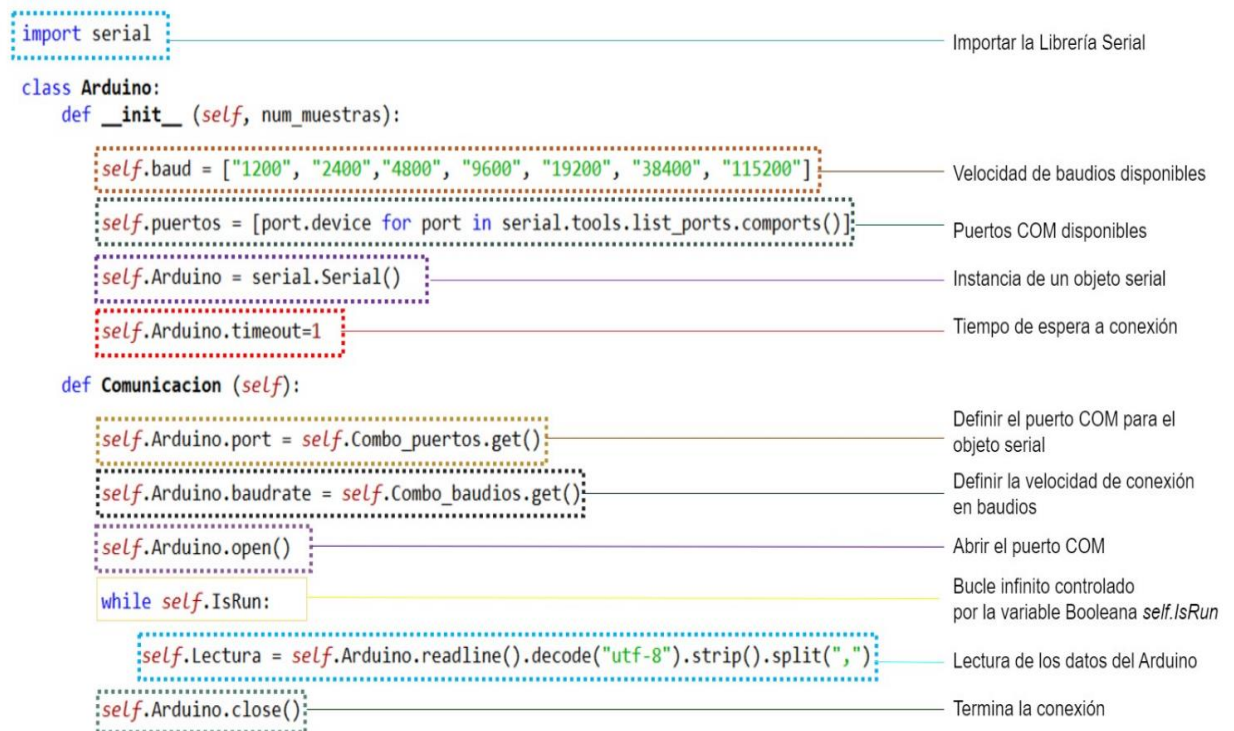
La librería serial, permite que al instanciar un objeto de esta clase se puedan utilizar distintos comandos para leer, almacenar y tratar los datos provenientes de un puerto serial. En la Figura 2.5 se muestra una forma de instanciar el objeto para lo cual es necesario asignar el puerto COM en el cual está conectado el Arduino y la velocidad de comunicación con el Arduino en baudios.

La asignación del puerto COM y la velocidad en baudios proviene de la selección del usuario mediante su interacción en la interfaz gráfica, que se explicará en las siguientes secciones.

El método *Comunicación* se utiliza para realizar una lectura infinita de los datos del Arduino, la lectura finaliza al cambiar el estado de la variable de control *self.IsRun*, la cual es una variable booleana y es muy utilizada para este tipo de acciones.

La Figura 2.5 es un extracto simplificado del código desarrollado para el software, para un mejor entendimiento del lector, dentro del código original se encuentran instrucciones adicionales y condicionales para el correcto funcionamiento (try, if, etc.), sin embargo, no son principales comandos para el proceso de lectura de datos provenientes del ARDUINO.





**Figura 2.5.** Algoritmo para comunicación serial con Arduino

### Decodificación y tratamiento de los datos provenientes del Arduino

Existen varias formas para tratar los datos que ingresan por el puerto USB, para que así sean compatibles con el resto de las librerías y puedan ser usados en Python. En el programa desarrollado se optó por utilizar los métodos propios de Python, los cuales son:

- *readline ()*. - lee y asigna los valores entregados por el DAQ hasta un salto de línea.
- *decode ("utf-8")*. - decodifica los valores del Arduino y los transforma en un formato legible para Python. Quita los caracteres b,\,r y n de la lista.
- *strip ()*, *split (",")*. - Elimina espacios y además separa los datos por comas.

El resultado del uso de todos estos métodos en conjunto es una lista de los valores de presión, temperatura, corriente y voltaje en una lista por cada medición que realiza el Arduino. En la Figura 2.6 se muestra la lista con datos tratados, la cual es la que se usará para los cálculos y gráficas del software.

```

Datos leídos directo del puerto serial
Cristian/Desktop/Archivos Programas/Python/Software Tesis/Bases serial')
[b'12.00,530.00,79.00,2022.00,18.00,18.00,-7.00,0.50\r\n']
[b'14.00,490.00,78.00,2165.00,8.00,18.00,-16.00,0.50\r\n']

Datos decodificados y tratados para su uso en Python
Cristian/Desktop/Archivos Programas/Python/Software Tesis/Bases serial')
['12.00', '530.00', '79.00', '2022.00', '18.00', '18.00', '-7.00', '0.50']
['14.00', '490.00', '78.00', '2165.00', '8.00', '18.00', '-16.00', '0.50']

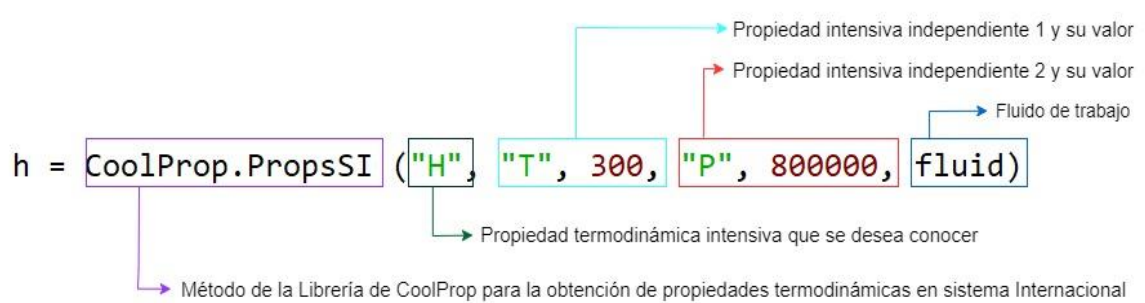
```

**Figura 2.6.** Resultado del tratamiento de datos para su uso en Python

### 2.3.3 Programación para el cálculo de propiedades termodinámicas

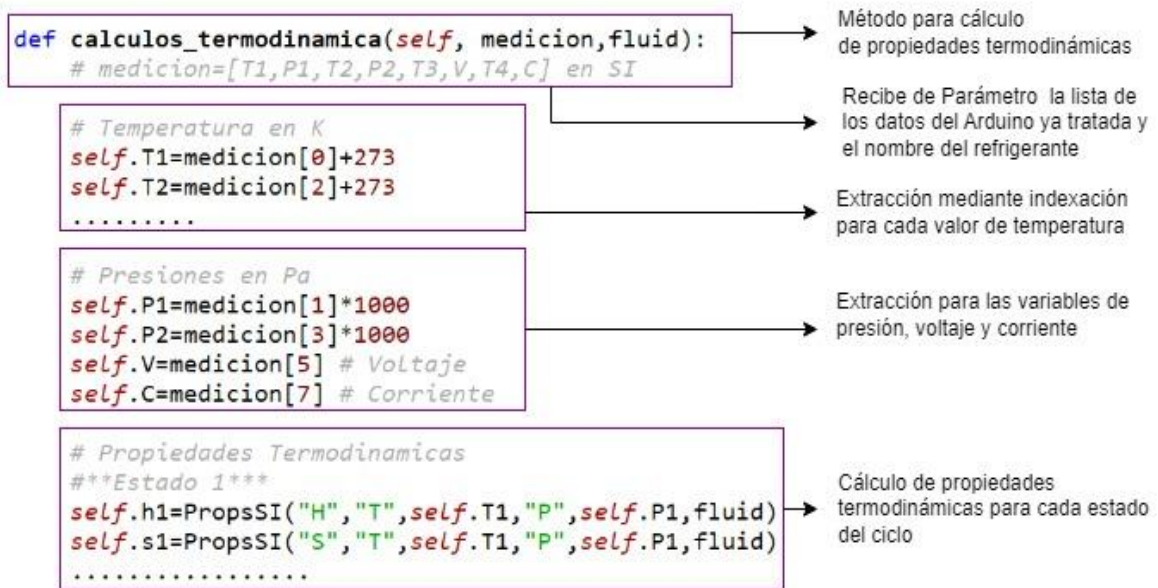
Una vez extraídos los datos del Arduino a través del puerto serial y asignados a la variable global dentro de la clase creada, es posible utilizar estos datos para determinar las propiedades termodinámicas, siempre que cumpla el postulado de estado, y continuar con los cálculos termodinámicos y de desempeño.

Se requiere de la librería CoolProp para lo anterior mencionado. La sintaxis de obtención de propiedades termodinámicas con esta librería se muestra en la Figura 2.7.



**Figura 2.7.** Sintaxis para la obtención de propiedades termodinámicas mediante *CoolProp*

Así, mediante la sintaxis descrita y los datos extraídos y asignados a una variable global, se define un método dentro de la clase Arduino para obtener las propiedades del refrigerante. En la Figura 2.8 se muestra un extracto de la programación de este método.



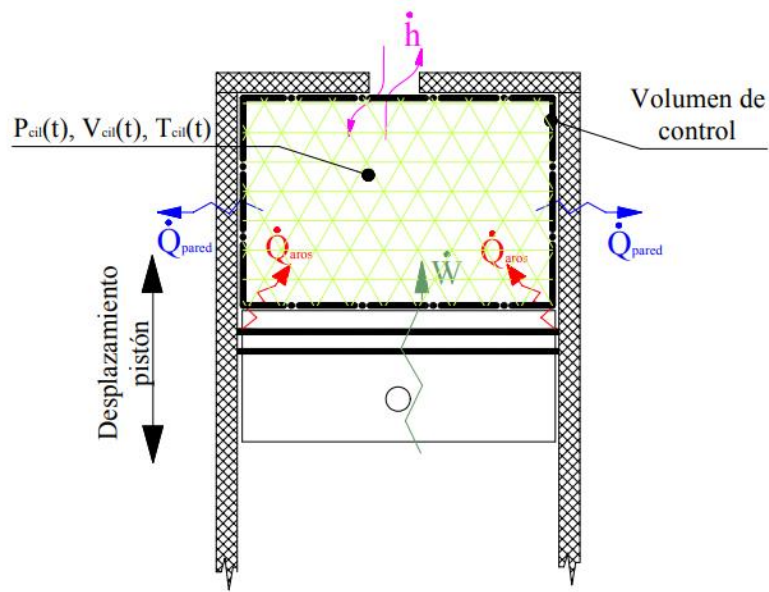
**Figura 2.8.** Asignación de mediciones provenientes del Arduino a variables globales

El algoritmo descrito se utiliza para obtener las propiedades termodinámicas del refrigerante R - 410A, tanto para el ciclo real como para el ciclo ideal y a cada estado termodinámico.

### Cálculo de desempeño de la bomba de calor

Para el ciclo ideal, el trabajo del compresor, por ende, el COP, se puede obtener únicamente con propiedades termodinámicas del refrigerante.

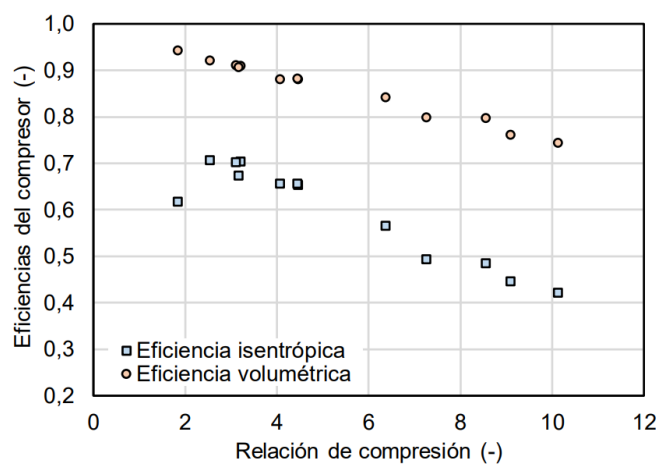
Sin embargo, para el cálculo del trabajo del compresor real es necesario realizar un balance de energía. En la Figura 2.9 se puede apreciar las formas de energía que intervienen en un proceso de compresión real, por lo que no es posible obtener el trabajo del compresor únicamente con las propiedades termodinámicas del refrigerante, ya que existen irreversibilidades como la transferencia de calor y las pérdidas de energía mecánica generadas por el rozamiento. Además, el modelo de compresión politrópica no incluye pérdidas de energía mecánicas ocasionadas por la calidad constructiva de cada compresor.



**Figura 2.9.** Balance de energía en un compresor tipo pistón [41]

Una forma de calcular el trabajo del compresor es mediante modelos matemáticos semiempíricos, los cuales toman en cuenta los aspectos constructivos del compresor. Sin embargo, para la correcta caracterización es necesario realizar pruebas experimentales, las cuales no están dentro del alcance de este proyecto.

Para una aproximación del trabajo del compresor real, se ha optado por utilizar una curva de eficiencia isentrópica, obtenida de un modelo matemático semiempírico, de un compresor rotativo, la cual se puede apreciar en la Figura 2.10. Esta curva es específica para un modelo de compresor, sin embargo, es de gran utilidad ya que toma en cuenta, pérdidas mecánicas, eléctricas, por succión/ descarga y fugas de refrigerante [42].



**Figura 2.10.** Curva de eficiencia isentrópica en función de relaciones de presión para compresor rotativo [45]

De la curva de eficiencia isentrópica en función de la relación de compresión, se puede calcular de forma aproximada el COP real de la bomba de calor durante la operación.

### **Trabajo específico del compresor real**

Se utilizó el concepto de eficiencia isentrópica para obtener el trabajo específico del compresor real [W/kg]. Además, se realizó una regresión polinómica de orden 4 de la curva de la Figura 2.10 para obtener una aproximación de la eficiencia isentrópica de un compresor rotativo y utilizarla para cuantificar el trabajo específico real del compresor del sistema a implementar, mediante la ecuación 2.1. Donde  $y$  es la eficiencia isentrópica y  $x$  es la relación de compresión.

$$y = -0.005x^4 + 0.0131x^3 - 0.127x^2 + 0.4651x + 0.1316$$

#### **Ecuación 2.1.** Regresión polinómica para la estimación de eficiencia isentrópica

Para el cálculo del trabajo real del compresor [W], se utiliza el producto de voltaje y corriente medidos de forma experimental con la Ecuación 1.6.

### **2.3.4 Programación para almacenamiento y registro de datos**

Por cada lectura del Arduino se debe realizar el cálculo de las propiedades termodinámicas y del COP, por ende, todas las variables utilizadas en este método se reescriben en cada iteración y es necesario contar un algoritmo de almacenamiento y registro de datos que permita guardar los valores medidos y calculados

Para la programación del algoritmo de almacenamiento y registro de datos se requiere de utilizar de las librerías `collections`, `statistics`, `os` y `time`.

#### **Almacenamiento en listas**

Una vez calculados y extraídos los datos de las propiedades térmicas del refrigerante y el desempeño de la bomba de calor se procede a almacenarlos y registrarlos. Existen dos listas principales creadas para el manejo de datos.

- La primera lista almacena de forma temporal los valores medidos y calculados de los últimos 10 segundos
- Una vez transcurrido los 10 segundos, se obtiene el promedio, mediante `statistics`, de las mediciones de los últimos 10 segundos almacenados en la primera lista, para así finalmente almacenarla en una lista de promedios no temporal.

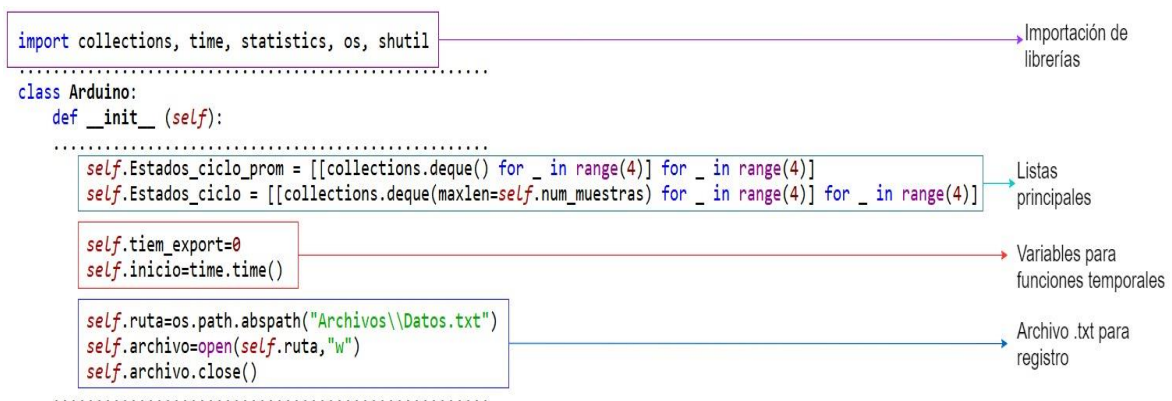
La primera lista se vacía cada 10 segundos para que el promedio solo corresponda a los datos de los últimos 10 segundos. Esta lógica de almacenamiento se incluye dentro del bucle infinito descrito en la Sección

### Registro de datos en CSV

Para el registro permanente de los datos medidos y calculados se realizó un algoritmo que exporta los valores de la lista de los promedios cada 10 segundos. Los datos se exportan continuamente en un archivo *.txt*, que puede estar fuera de la carpeta principal del programa, en formato de CSV.

### Algoritmo y lógica para almacenamiento y registro de datos

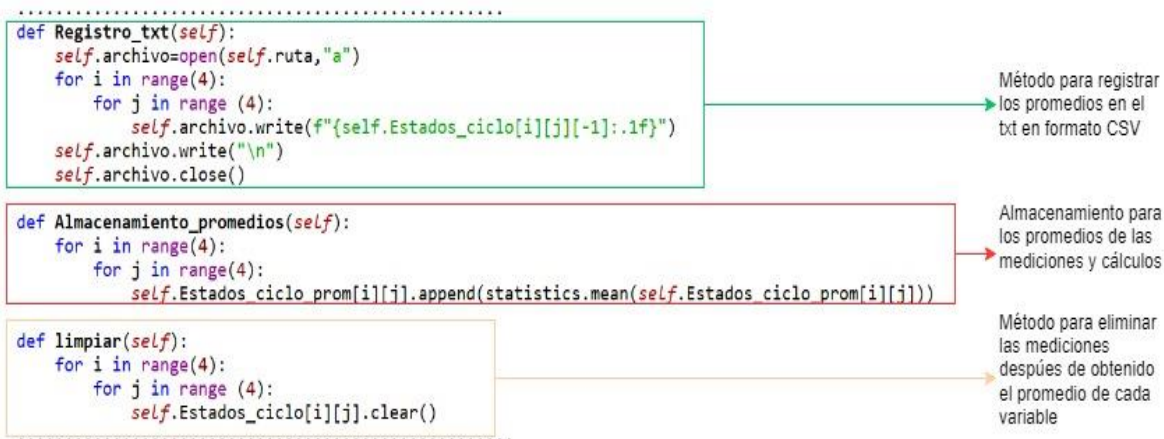
En la Figura 2.11 se muestran fragmentos del código implementado para explicar los algoritmos necesarios para almacenar y registrar. Estos fragmentos son para fines explicativos y no corresponden en su totalidad al código implementado, sin embargo, se utilizó la misma dinámica para el desarrollo final.



**Figura 2.11.** Código de programación para el registro de datos en formato CSV I

Las listas principales contienen sub-listas, cada sub-lista contiene las mediciones de cada estado termodinámico, además dentro de cada sub-lista existe otras sub-listas que contienen los datos según la variable termodinámica (Temperatura, Presión, Entalpía y Entropía), en total se tienen 16 sub-listas (4 estados del ciclo x 4 variables termodinámicas)

En la Figura 2.12 se muestran los métodos creados, que sirven para la actualización, limpieza de listas y registro de datos en formato CSV.



**Figura 2.12.** Código de programación para registro de datos en formato CSV II

Una forma de recorrer por todas las listas es mediante el uso de ciclos *for* anidados. El almacenamiento de las mediciones y cálculos en la lista temporal se lo realizó de forma manual con el método `append`, sin embargo, se lo omitió en la Figura 2.12. Este algoritmo, al igual que otros, forman parte del bucle principal del software.

Además de algoritmos que se ejecutan en cada iteración, existen partes del código que no se deben de ejecutar en cada lectura del Arduino como la obtención de los promedios de las últimas 10 mediciones de cada sensor, y su control está dado por variables temporales.

En esta sección, en conjunto con las anteriores sobre desarrollo del software, se mencionó los principales métodos, algoritmos y lógicas de programación para llegar a obtener datos medidos y calculados, los cuales están almacenados y organizados en listas y registrados en un archivo *.txt*.

Con los datos necesarios disponibles en atributos de la clase Arduino, es posible utilizar estos datos para la creación de figuras, diagramas y elementos visuales que permitan al usuario interpretar la información obtenida.

### 2.3.5 Programación para la obtención diagramas termodinámicos y gráficas para monitoreo de presión y temperatura

Una vez obtenidas todas las variables termodinámicas de interés en cada estado del ciclo es posible utilizar *Matplotlib* y *Fluprodia* para la obtención de gráficos y diagramas para su análisis.

## Monitoreo de presión y temperatura con *Matplotlib*

*Matplotlib* permite crear visualizaciones estáticas, interactivas o animadas en Python. El monitoreo de las variables de presión y temperatura exige que el tipo de visualización sea de tipo animado, para que se pueda monitorear estas variables en tiempo real y conocer el comportamiento de la bomba de calor durante su funcionamiento.

FuncAnimation es una clase de *Matplotlib* que es utilizada para generar animaciones mediante la ejecución repetitiva de una misma función [43]. En la Figura 2.13 Se muestra la sintaxis para realizar una animación 2D con FuncAnimation.

```
matplotlib.animation.FuncAnimation(fig, func, frames=None, init_func=None, fargs=None,
                                   save_count=None, *, cache_frame_data=True, **kwargs)
```

**Figura 2.13** Instancia de la clase FuncAnimation de *Matplotlib* y configuración

Como se puede observar en la Figura 2.13 al instanciar un objeto de la clase FuncAnimation, este admite muchos parámetros de entrada incluso más que los que se puede observar. Hay 2 parámetros mínimos necesarios para crear una animación.

- **Figura.** – es en donde se realizará la animación, la figura es un objeto que puede ser creado mediante la clase pyplotlib que pertenece a *Matplotlib* también.
- **Función.** – se refiere a las instrucciones que actualizarán los datos de la gráfica para que esta sea una animación. FuncAnimation se encarga de llamar a la función en cada iteración según se establezca en el intervalo de actualización. Si la función requiere de argumentos, estos deben de ser ingresados en el parámetro de fargs.

En el programa desarrollado la función asociada está relacionada a la extracción de los datos provenientes de las listas de los estados termodinámicos de la clase Arduino.

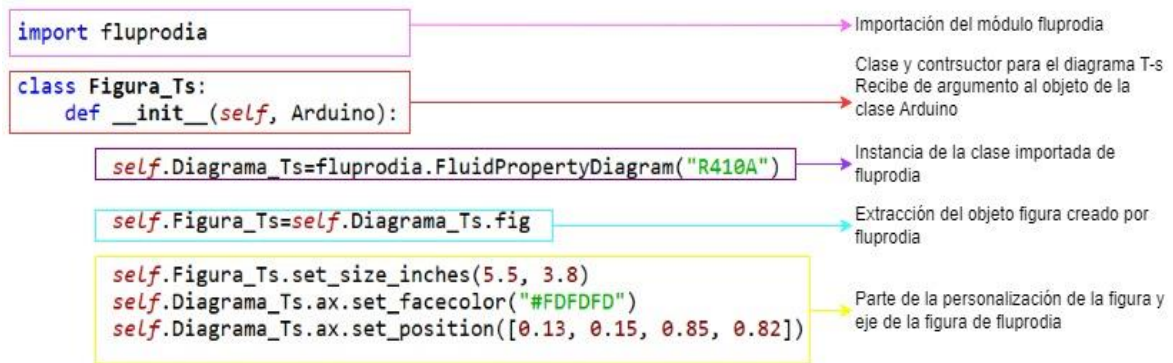
La creación y configuración de figuras en *Matplotlib* no es difícil y se cuenta con gran cantidad de documentación de la cual se puede obtener información del manejo de esta librería (ver [43]).

## Diagramas termodinámicos T-s y P-h con *Fluprodia*

Fluprodia es una biblioteca aún en desarrollo, está poco optimizada y requiere de artificios para obtener mejores resultados. Al ser una biblioteca que utiliza *Matplotlib* y *Numpy* es posible utilizar métodos de cada biblioteca para modificar los diagramas entregados por fluprodia y obtener mejores resultados.



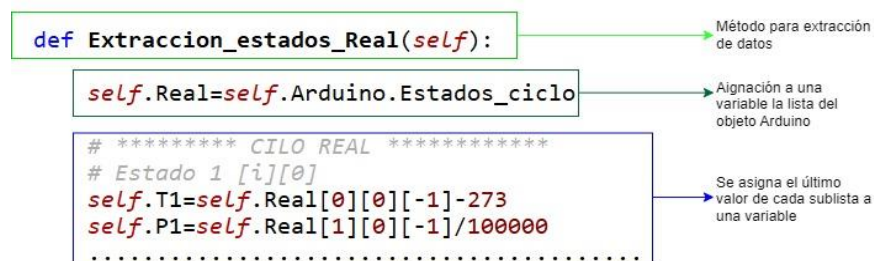
En la Figura 2.14 se muestra un fragmento de la programación para la obtención de diagramas termodinámicos. El algoritmo mostrado se usa para el diagrama T-s y P-h con las modificaciones necesarias.



**Figura 2.14.** Fragmento del constructor de la clase *Figura\_Ts* para la obtención del diagrama termodinámico con fluprodia

En la Figura 2.14 se muestran solo algunos atributos utilizados, pero más relevantes para la creación del diagrama y la forma de personalización de la figura. Se utiliza métodos de *Matplotlib* como `set_facecolor`. Además, es importante notar esta clase recibe como argumento un objeto *Arduino*, pues los atributos de este contienen la información necesaria para obtener los diagramas.

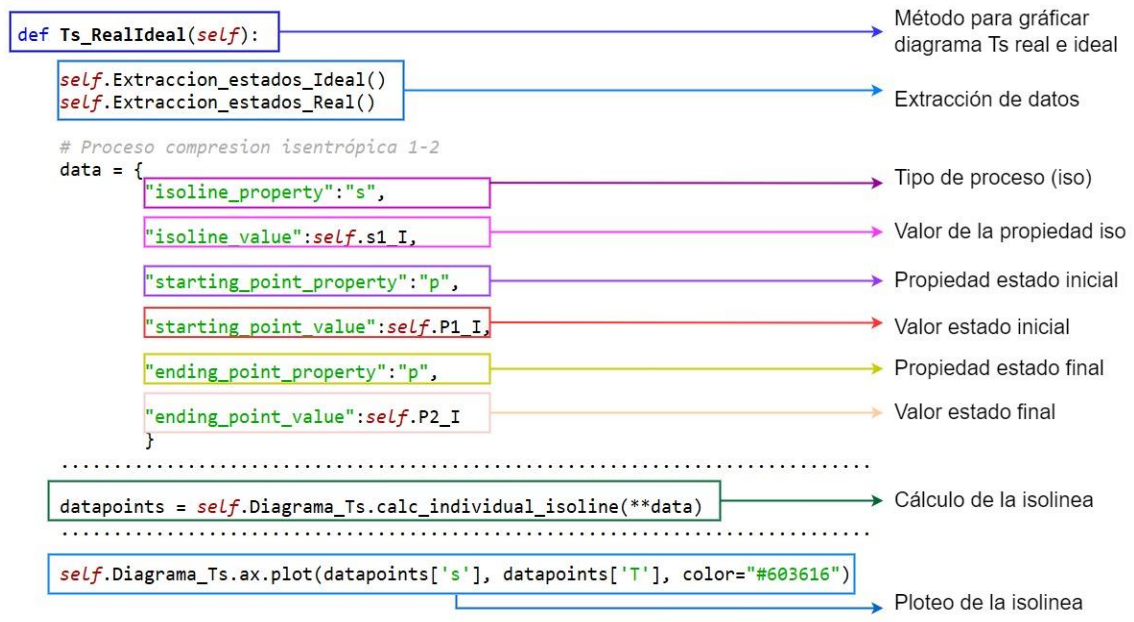
En la Figura 2.15 se muestra la forma de extracción de los datos provenientes del objeto *Arduino*.



**Figura 2.15.** Extracción de los valores medidos provenientes del Arduino y cálculos realizados con CoolProp.

La extracción de atributos de un objeto que pertenece a una clase se la realiza con un punto (.) así, para este caso se extrae la lista que contiene los valores de los estados del ciclo termodinámico de la bomba de calor y se los asigna a una variable global dentro de la clase *Figura\_Ts* para que así se tenga acceso a ese dato en cualquier método.

Una vez obtenido el algoritmo para extracción de datos, se pueden utilizar estos datos para obtener el diagrama T-s con fluprodia. En la Figura 2.16 se muestra parte del algoritmo utilizado para la graficar el diagrama T-s. La programación es similar para el resto de los procesos del ciclo termodinámico de la bomba de calor.

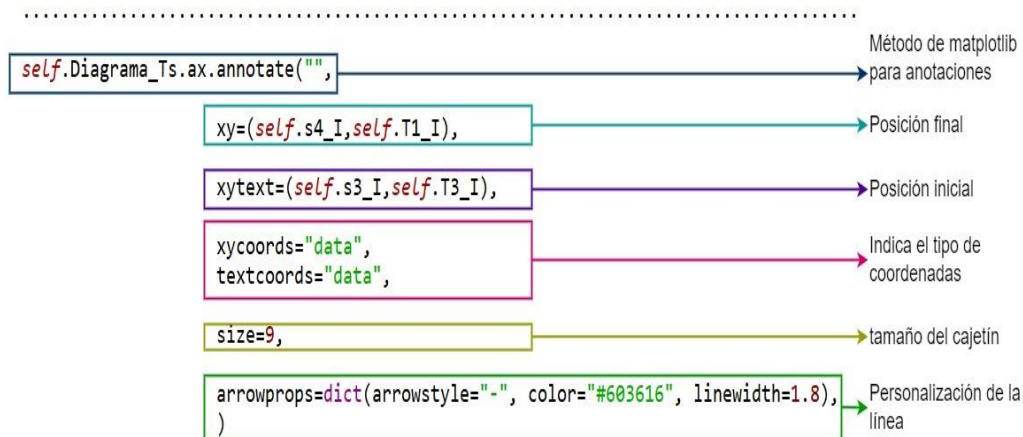


**Figura 2.16.** Código para graficar diagramas termodinámicos mediante isólinneas de fluprodia

Al tener poco tiempo de desarrollo, fluprodia tiene fallos durante la generación de diagramas y ciclos termodinámicos. En este proyecto fluprodia no genera la línea del proceso isentálpico, correspondiente a la expansión en la válvula utilizando el refrigerante “R-410A”. Sin embargo, fluprodia permite el uso de métodos de *Matplotlib*, y es posible graficar procesos termodinámicos con líneas mediante el método `annotate` de *Matplotlib* como se muestra en el Figura 2.17.

La programación descrita en esta sección se utilizó también para crear los diagramas P-h con todas sus variantes. Las variantes que se obtienen por cada tipo de diagrama T-s y P-h son:

- Diagrama del ciclo ideal
- Diagrama del ciclo real
- Diagrama del ciclo real e ideal



**Figura 2.17.** Código para crear líneas de procesos mediante métodos de *Matplotlib* para anotaciones.

### 2.3.6 Diseño de la interfaz gráfica de usuario GUI

Una vez se tienen todos los datos, y representaciones visuales para el usuario, es necesario diseñar la interfaz gráfica de usuario (GUI). La interfaz gráfica de usuario es la parte visual que interactúa con el usuario, por ende, el diseño debe de estar basado en la necesidad de este.

Los principios de usabilidad y experiencias sirven como guía para el diseño de la GUI; según la revisión bibliográfica, el diseño y desarrollo de una interfaz gráfica tiene un mayor alcance que la programación [44]. La metodología se caracteriza por un diseño centrado en el usuario, según la ISO 9241 [45].

La usabilidad es el grado en que sus usuarios pueden utilizar un sistema y su nivel de satisfacción, eficiencia y efectividad durante su uso [48]. En otras palabras, la usabilidad es la calidad de la experiencia del usuario durante su interacción con sistemas, productos, servicios, dispositivos o aplicaciones como interfaces gráficas. La usabilidad se basa en 4 objetivos, enfocados en una GUI, que son de utilidad, facilidad de aprendizaje, simpatía y eficacia [45].

La metodología de diseño de la interfaz gráfica propuesta para el presente proyecto se basa en el diseño enfocado en la usabilidad, para así garantizar que el software esté centrado en el usuario y se pueda cumplir con los objetivos del proyecto. El diseño enfocado en la usabilidad utiliza 3 principios fundamentales para así cumplir con los objetivos descritos anteriormente, estos 3 principios son [45]:

- Enfoque temprano en tareas y usuarios. – Se debe de identificar correctamente los usuarios que harán uso de la interfaz gráfica y las tareas que realizarán con ella. Se debe de tomar en cuenta características cognitivas, de comportamiento, actitud y la naturaleza del trabajo que se desea lograr.
- Mediciones empíricas. – Los potenciales usuarios son parte fundamental para el desarrollo de la GUI, se debe de tener en cuenta sus reacciones en situaciones reales donde se utilicen simulaciones o prototipos.
- Diseño iterativo. – El diseño es un proceso cíclico donde sus fases son: diseño, prueba y evaluación y se debe de repetir cuantas veces sea necesario.

En este contexto de diseño, el objetivo final de esta metodología es obtener una GUI que ofrezca la mejor experiencia para el usuario y que sea capaz de cumplir con la naturaleza del trabajo.

En la Tabla 2.3 se describen las características más relevantes del usuario, al que está enfocado este proyecto, los requisitos y tareas que debe ser capaz de cumplir la GUI y características de usabilidad enfocadas al estudio termodinámico de una bomba de calor real.

**Tabla 2.3. Requerimientos de la GUI basados en un diseño orientado el usuario**

Usuario	Tareas	Cualidades
La GUI está destinada para estudiantes universitarios, especialmente para estudiantes de la carrera de Ingeniería Mecánica y Mecatrónica de la FIM-EPN	Las tareas principales de la GUI son permitir: <ul style="list-style-type: none"> <li>- Monitoreo en tiempo real</li> <li>- Visualización de diagramas termodinámicos</li> <li>- Visualización de variables termodinámicas y desempeño de la bomba de calor en tiempo real</li> </ul>	Las cualidades de la GUI están orientadas a brindar la mejor experiencia posible durante su uso. Las principales características son: <ul style="list-style-type: none"> <li>- Fácil uso - intuitiva</li> <li>- Diseño agradable</li> <li>- Rápido aprendizaje</li> </ul>
El usuario utilizará la GUI en conjunto con el banco de pruebas descrito en la Sección 2.1.3. Por lo que se espera un ambiente académico y un comportamiento riguroso por parte de los usuarios	Adicionalmente, ya que la GUI tiene fines educativos, se debe de incluir funciones como <ul style="list-style-type: none"> <li>- Simulación de sistemas de bomba de calor</li> <li>- Cuestionario para mejorar el aprendizaje</li> <li>- Manual de usuario</li> </ul>	Una cualidad importante para una mejor experiencia del usuario es que la interfaz gráfica se pueda abrir desde un archivo ejecutable, sin la necesidad de instalarlo.

Una vez identificado al usuario, las tareas y cualidades que debe de tener la GUI en base a un diseño enfocado en la usabilidad se diseña la interfaz gráfica de forma iterativa para cumplir con los requerimientos presentados en esta sección.

### 2.3.7 Programación de la Interfaz gráfica

Existen gran variedad de librerías capaces de crear y diseñar interfaces gráficas de usuario con distinto nivel de complejidad y número de herramientas. Las librerías más populares son PyQt5 (u otras versiones), Kivy, WxPython, DearPyGui, Pyside y *Tkinter*. *CustomTkinter* no se encuentra en la lista debido a que es una librería relativamente nueva. Para la programación de la interfaz gráfica se utilizó las librerías de *Tkinter* y *CustomTkinter*, ambas en conjunto por su compatibilidad, por preferencia propia y recomendaciones literarias sobre el comienzo en el desarrollo de interfaces gráficas en Python [30]. El uso de estas librerías permite un mayor control sobre la personalización de los elementos de la interfaz gráfica, aunque tengan menos herramientas disponibles que la mayoría de las opciones nombradas, sin embargo, permite un mejor aprendizaje sobre la creación de interfaces gráficas.

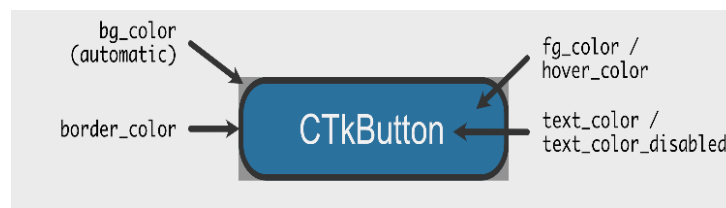
Una GUI, sin importar la librería que se utilice, está compuesta de diferentes *widgets* individuales. Un *Widget* es representado como un objeto de Python, instanciado de una clase y personalizado mediante atributos [46]. Ahí la importancia de la programación orientada a objetos (POO) como paradigma de solución para crear interfaces gráficas y aplicaciones.

Los *widgets* más utilizados para el desarrollo de este proyecto son:

- *Frame*: Un frame es un *widget* tipo contenedor, es decir, es el espacio donde se puede añadir otros *widgets*, incluso otros frames. Sirve para organizar de mejor forma la GUI y se puede configurar su color de fondo.
- *Label*: Es una etiqueta que permite mostrar texto o imagen, además se le pueden asociar variable para así cambiar el contenido del *Label*.
- *Button*: Es un objeto de tipo botón, en él se puede asociar una función para realizar un comando al momento de dar clic.
- *Entry*: Es una etiqueta que permite introducir texto corto del usuario en la GUI. El texto se puede extraer con el método *.get()*, propio de la clase.
- *Combobox*: Es una lista desplegable de opciones. Al igual que el *widget* *Entry*, se puede obtener el contenido seleccionado con el método *.get()*.

- *Scale*: Es un control deslizable que permite definir un número dentro de cierta escala. Se pueden asociar eventos para que ejecute comandos según se interactúe con el control deslizable.
- *RadioButton*: Este *widget* permite la selección de una opción entre varias, a diferencia del combobox, todas las opciones están visibles en la GUI.
- *Notebook*: Es un *widget* tipo menú que sirve como contenedor, la ventaja principal es que cuenta con paneles que comparten el mismo espacio y el usuario puede intercambiar entre ellos.
- *Message*: Muestra al usuario un mensaje programado. Los tipos de mensaje más comunes son de error, información o de advertencia.

En la Figura 2.18 Se muestra un ejemplo de un *widget* de *CustomTkinter* y algunos parámetros de configuración

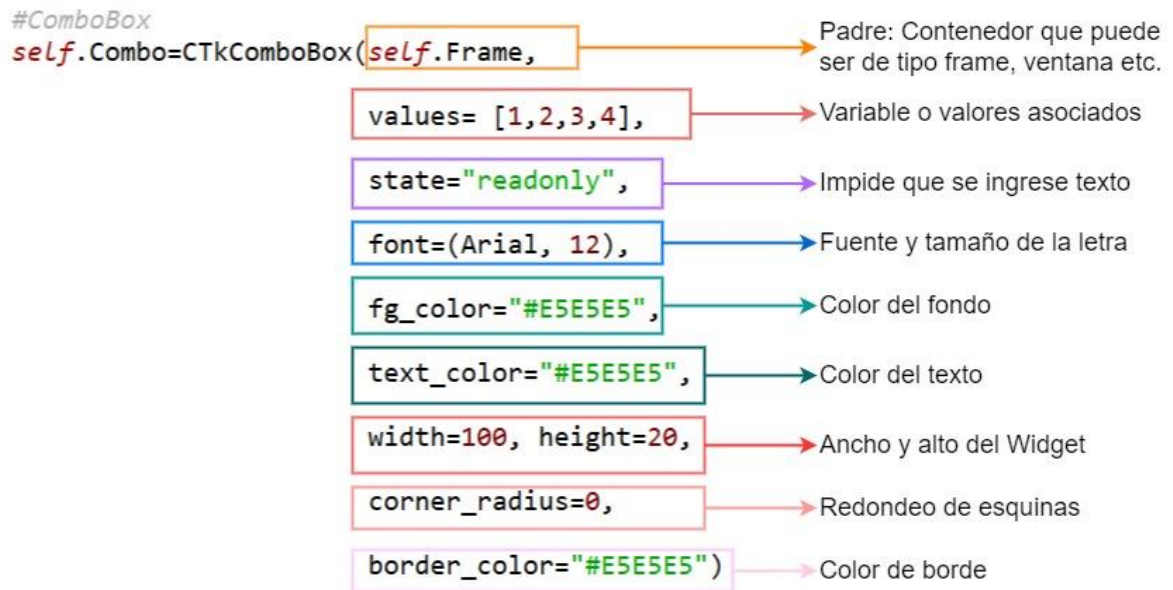


**Figura 2.18** Elementos de un botón de la librería CTKTkinter [47]

La mayoría de los *widgets* mencionados tienen su equivalente en *CustomTkinter*. En la Figura 2.19 se puede observar cómo es la sintaxis para crear un objeto de la clase *ComboBox* de la librería *CustomTkinter*. Sin embargo, esta sintaxis es muy parecida para configurar casi cualquier *widget* y puede usarse como guía para este fin. Para obtener información de como instanciar y personalizar de forma correcta un *widget* se puede ver la documentación [46], [47].

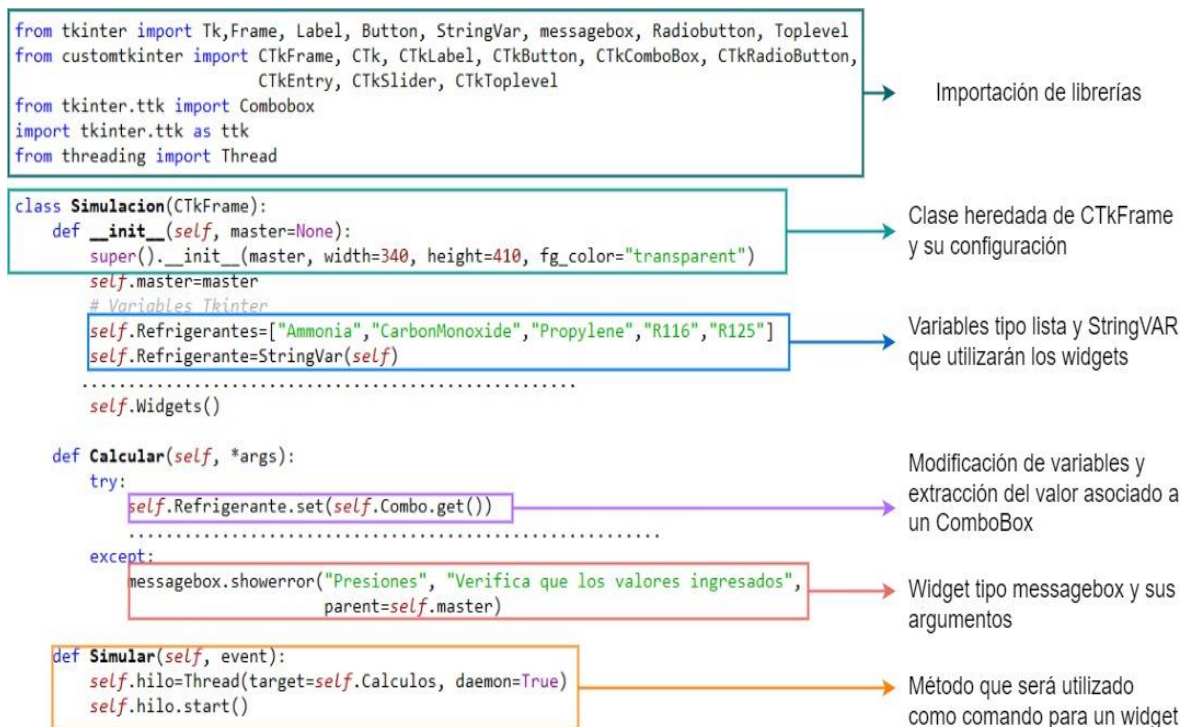
Los *widgets* se organizan en jerarquías, se puede establecer niveles superiores como contenedores de *widgets* de menor nivel dentro del marco raíz. Al momento de crear un *widget* hijo, se debe de pasar el argumento padre para identificar la dependencia o nivel [46].

Es necesario utilizar un administrador de geometrías para añadir a la GUI los *widgets* que han sido creados [46]. Los administradores son métodos, por ende, los objetos deben de pertenecer a una clase que admita estas funciones. Los métodos existentes en *Tkinter* y *CustomTkinter* para añadir *widgets* son, *place*, *pack* y *grid*. Solo se puede usar un tipo de administrador de geometrías en cada nivel.



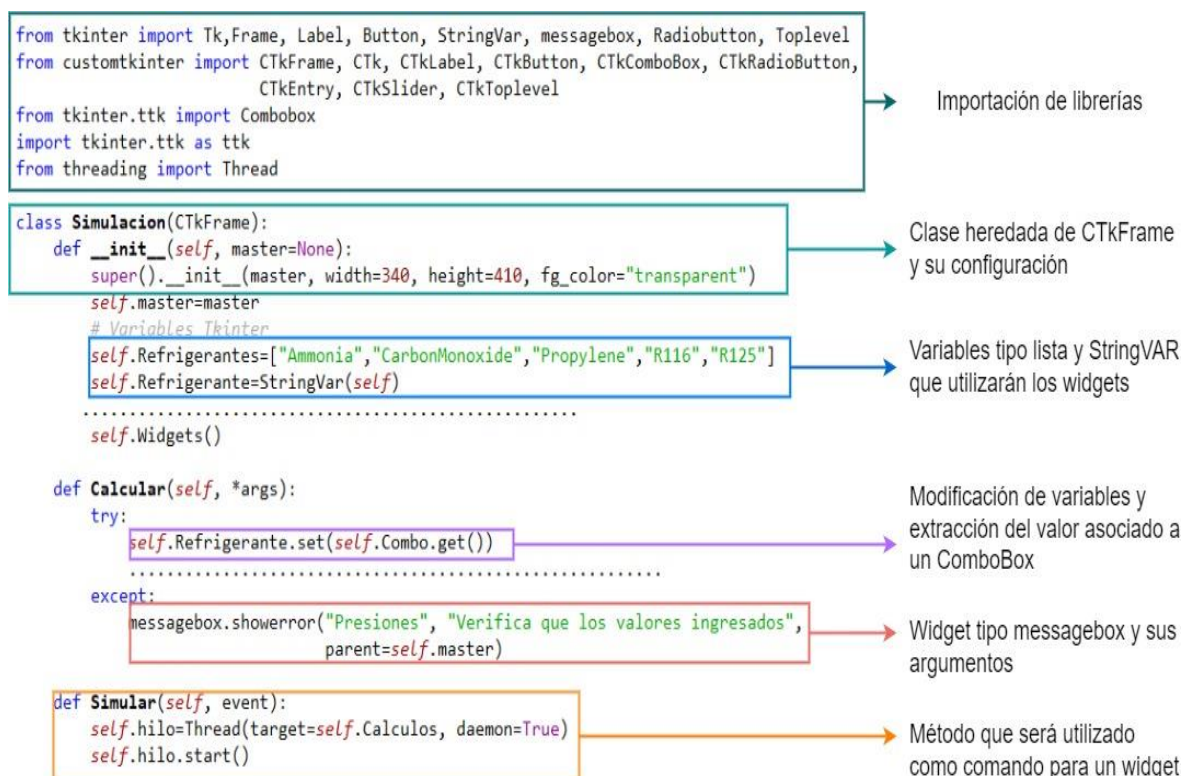
**Figura.2.19.** Instancia y configuración de un *widget* tipo ComboBox

En la Figura 2.20 se muestra un fragmento del código utilizado para crear una parte de la GUI desarrollada en este proyecto. La Figura 2.20 contiene código base que usarán por *widgets* como métodos, variables y objetos que serán utilizados para la configuración de un *widget*.



**Figura 2.20.** Código de Programación inicial para la creación de una GUI mediante programación orientada a objetos

En la Figura 2.21 se muestra la sintaxis para crear *widgets* y como personalizarlos a través de argumentos.



**Figura 2.21.** Código de programación para añadir *widgets* a la GUI

### 2.3.8 Programación para obtener un archivo ejecutable

Un requerimiento descrito en la Tabla 2.3 es que la GUI pueda ser ejecutada sin necesidad de instalarla. Se utiliza la librería *Pyinstaller* de Python para obtener este archivo. *Pyinstaller* lee el script realizado por el programador, lo analiza con el fin de obtener todos los archivos necesarios y dependencias y los compila incluido el intérprete de Python activo y los coloca en una sola carpeta o un solo archivo ejecutable [48].

La obtención de una sola carpeta tiene como ventaja que el script se pueda modificar, lo que permite corregir bugs y visualizar el código fuente. La desventaja es que para ciertos usuarios puede resultar difícil encontrar el ejecutable entre tantos archivos [51].

La opción de un solo archivo ejecutable crea una carpeta temporal donde se ejecuta un entorno virtual de Python y lee los archivos y dependencias necesarias en esta carpeta. Al crear carpetas y archivos temporales la aplicación puede tardar más tiempo en abrir, lo cual puede resultar en una desventaja [48].



El ejecutable obtenido es compatible con la arquitectura del PC desde donde se obtuvo, así para una PC (x64) Windows el ejecutable será compatible únicamente para ese sistema operativo y esa arquitectura.

En la Figura 2.22 Se muestra el código utilizado para obtener el archivo ejecutable mediante la opción de una sola carpeta.

```
(Tesis_BombadeCalor) C:\Users\Cristian\Desktop\Ejecutable HEAT PUMP MONITOR>pyinstaller HEAT_PUMP_MONITOR.py --collect-all customtkinter -w
1260 INFO: PyInstaller: 5.6.2
1260 INFO: Python: 3.10.11 (conda)
1260 INFO: Platform: Windows-10-10.0.22621-SP0
1260 INFO: wrote C:\Users\Cristian\Desktop\Ejecutable HEAT PUMP MONITOR\HEAT_PUMP_MONITOR.spec
1276 INFO: UPX is not available.
3002 INFO: Determining a mapping of distributions to packages...
```

**Figura 2.22.** Comandos en anaconda prompt para la obtención del software como ejecutable

En el código de la Figura 2. se puede notar que se agregó las instrucciones de *CustomTkinter -w*, esto es requerido ya que *Pyinstaller* no reconoce a *CustomTkinter* como dependencia y la aplicación no se ejecutará sin tener todas las dependencias necesarias.

### 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan como resultados principales las ventanas obtenidas que conforman parte de la GUI, su funcionamiento y elementos principales. Además, se verifica el funcionamiento del software desarrollado y los parámetros y variables termodinámicas necesarias para el análisis termodinámico de la bomba de calor a implementar en el laboratorio de termodinámica. Se detallan las interacciones principales disponibles entre el usuario y la interfaz gráfica como diagramas, figuras, entradas de texto y *widgets* de selección que están disponibles en la GUI.

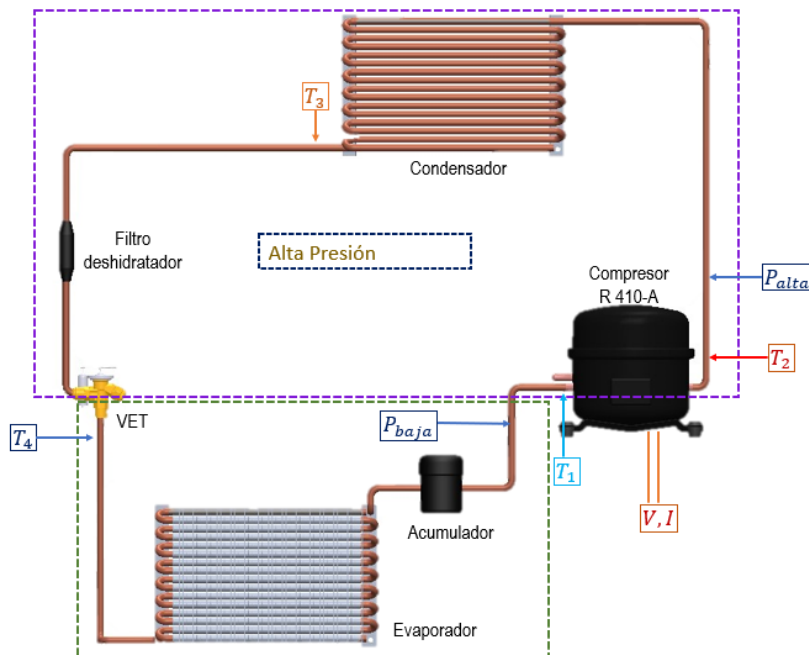
#### 3.1 Resultados

##### 3.1.1 Parámetros de análisis de la bomba de calor

###### Ubicación de sensores

La ubicación de los sensores se basó en las recomendaciones de las referencias bibliográficas y por criterios propios por accesibilidad a los puntos de medición. Además, se determinaron los puntos necesarios de medición para lograr cumplir con los objetivos de monitoreo y análisis de desempeño de la bomba de calor.

En la Figura 3.1 se muestra un esquema de la ubicación de los sensores que se seleccionaron para el desarrollo del proyecto.



**Figura 3.1.** Esquema de la ubicación de sensores para el equipo de laboratorio

Como se observa en la Figura 2. Se tiene 4 sensores de Temperatura y dos sensores de presión, puesto que se puede considerar despreciable la pérdida de presión en las tuberías. Es importante notar que, la distancia entre el compresor y el condensador en realidad es muy pequeña en la mayoría de los modelos de bombas de calor, por lo que, se puede utilizar un solo sensor para este tramo para sin afectar de forma significativa los resultados obtenidos de los cálculos.

### 3.1.2 Software e Interfaz gráfica desarrollada en Python

“Heat Pump Monitor” es el software desarrollado en Python, que cuenta con una interfaz gráfica de usuario para la interacción con el programa. En la Figura 3.2 se muestra la pantalla principal del programa, en este constan 4 botones que despliegan 4 ventanas.



**Figura 3.2.** Pantalla principal del programa “Heat Pum Monitor”

Iniciar Práctica: es el botón principal de la aplicación, abre una ventana secundaria (ventana “Práctica”) donde se monitorea y analiza el ciclo termodinámico del refrigerante de la bomba de calor del banco de pruebas en tiempo real. Además, permite la simulación de ciclos termodinámicos de compresión de vapor con distintos refrigerantes.

Ventana “Marco Teórico”: En esta ventana se muestra la teoría asociada a las bombas de calor como conceptos fundamentales de energía y el ciclo termodinámico de refrigeración por compresión de vapor.

Ventana “Cuestionario”: En esta ventana se puede realizar una pequeña evaluación de los conceptos descritos en el marco teórico, esto ayuda a que la práctica tenga un mayor aprovechamiento por parte del usuario final.

Ventana “Manual de Usuario”: En esta ventana se muestra material didáctico que ayuda al manejo e interacción con el software a través de la GUI.

### **3.1.3 Funcionamiento del software: Heat Pump Monitor**

En las siguientes secciones se detallan las funciones principales de la ventana “Práctica” y ventana “Manual de usuario”. Se explicará la arquitectura de funcionamiento de la ventana “Práctica”, donde se puede observar el flujo de información, objetos y métodos creados para su correcto funcionamiento y finalmente el contenido de la ventana “Manual de Usuario” que indica como utilizar la aplicación y sus principales elementos.

La ventana “Marco Teórico” cubre los principales fundamentos y conceptos clave sobre bombas de calor y ciclos termodinámicos, y no se detallará en las siguientes secciones, sin embargo, se puede observar sus principales elementos en el **ANEXO II**.

La ventana “Cuestionario” no se detallará en las siguientes secciones y se puede observar sus principales elementos en el **ANEXO III**.

### **3.1.4 Ventana “Práctica”**

Como se mencionó, la parte principal del software es la ventana “Práctica”, que se visualiza al pulsar el botón iniciar práctica.

En la Figura 3.3 se puede observar la ventana “Práctica”, esta ventana está conformada por 5 Frames, los cuales interactúan con el usuario mediante botones, sliders, elementos visuales, etc. Y que además interactúan entre sí. No toda la interacción se da dentro de elementos de los Frames, existe interacción entre Frames con listas, variables y funciones que no se puede observar en la GUI.

El funcionamiento del programa se basa en el flujo de información y recepción de comandos por parte del usuario a través de widgets. Un esquema de este flujo de información se observa en la Figura 3.4.



Figura 3.3. Ventana “Práctica” y los frames que la componen

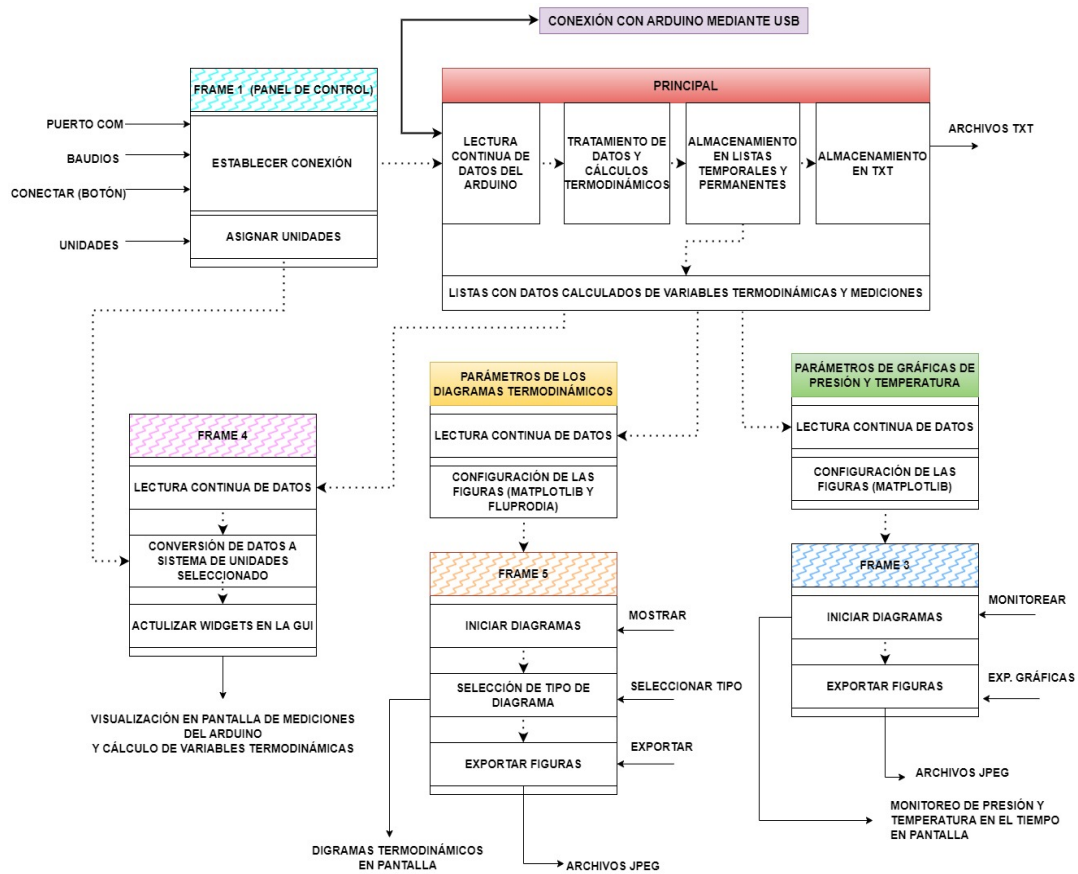
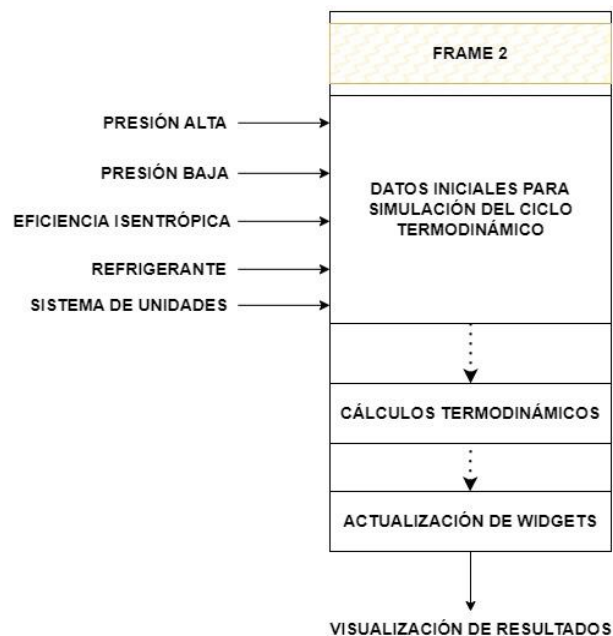


Figura 3.4. Diagrama de Funcionamiento de “Heat Pump Monitor”

En la Figura 3.4 se muestra un diagrama explicativo de la interacción de los objetos dentro de la interfaz y con el usuario, las rutinas y funciones principales del software. Además, se pueden observar salidas que son una forma de interacción con el usuario o alrededores. Cada bloque representa un objeto instanciado en el programa. Como se mencionó en la metodología (sección 2.6), se utilizó el paradigma de POO como método de desarrollo para el software, por lo que el programa está conformado por instancias de las clases creadas en el código fuente del software. Los objetos que son visibles para el usuario son los Frames 1,2,3,4 y 5. El Frame 2 no fue incluido en el diagrama debido a que no interactúa con ningún otro objeto y puede ser analizado como un programa independiente.

En la Figura 3.5 se muestra un diagrama del funcionamiento de la sección “Simulación” correspondiente al Frame 2.



**Figura 3.5.** Funcionamiento de la sección “Simulación” - Frame 2

En cada objeto, de las Figuras 3.4 y 3.5, se puede apreciar los métodos principales que ejecutan las rutinas necesarias para el funcionamiento del programa, sin embargo, existen muchas más funciones auxiliares. Las funciones auxiliares son destinadas principalmente a corregir errores que se pueden presentar durante la lectura de datos del Arduino, sin embargo, no se encuentran mencionadas en las Figuras anteriores.

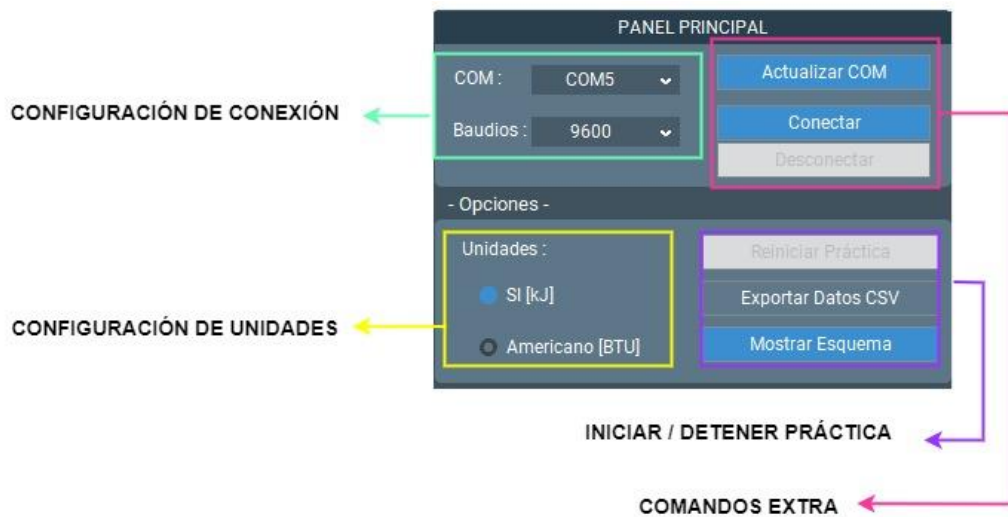
### 3.1.5 Manual de Usuario

Esta sección está orientada a explicar el funcionamiento de la ventana “Práctica” que es el contenido de la ventana “Manual de Usuario” en la GUI desarrollada.

Para una mejor comprensión se utilizará la distribución por frames descrita en la Figura 3.3 y se explicará su principales elementos y funciones.

#### Frame 1: Panel Principal

En la Figura 3.6 se muestra el Panel Principal, en este panel el usuario establece los parámetros principales de conexión y puede ejecutar los comandos principales para empezar la práctica de laboratorio.



**Figura 3.6.** Frame 1: Panel Principal de la Ventana “Práctica”

**COM:** En esta lista desplegable se encuentra los puertos COM disponibles de la computadora que está ejecutando el programa. Si el Arduino es conectado después de abrir el programa es necesario utilizar el botón “Actualizar COM”

**Baudios:** Establece la velocidad de conexión con el Arduino. Por lo general la velocidad en baudios es de 9600.

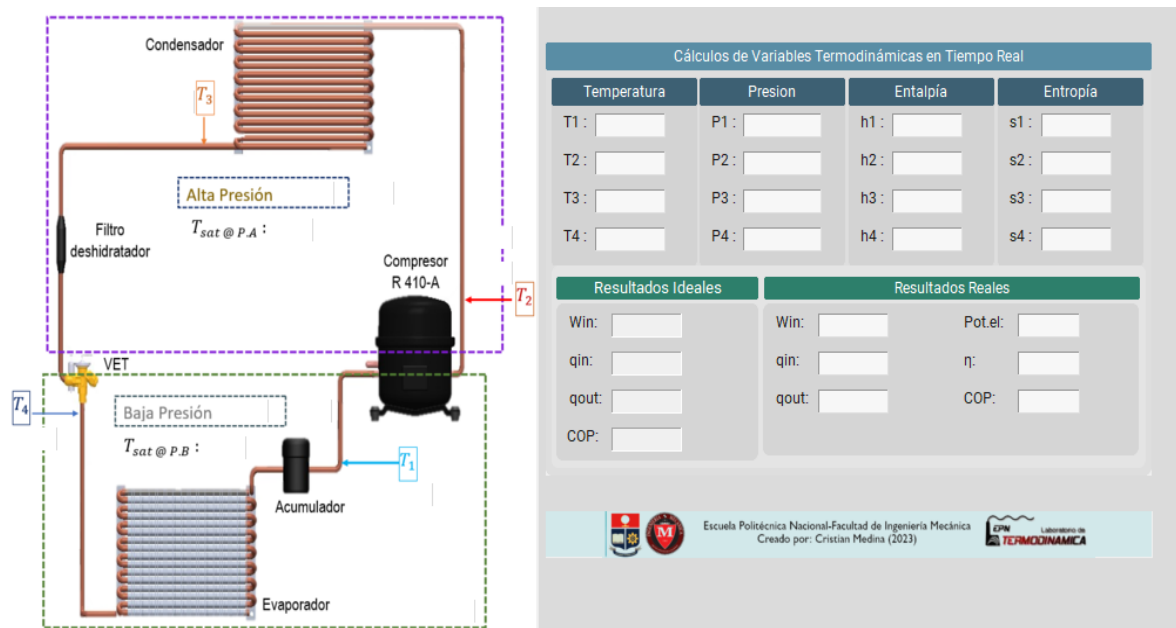
**Actualizar COM:** Ejecuta un comando para detectar los puertos COM donde se encuentran conectados periféricos mediante USB.

**Conectar:** Una vez establecida la configuración se puede ejecutar el comando Conectar. Este comando ejecuta e inicia los métodos de la clase principal (ver Figura 3.4).

Reiniciar Práctica: Reinicia la ventana “Práctica”, así se limpia el buffer del Arduino y de la aplicación para que no exista datos restantes en la interfaz.

Exportar Datos CSV: Este comando permite que los datos almacenados en el archivo *.txt*, que existe en el entorno de la aplicación, sea exportado a una ubicación diferente.

Mostrar Esquema: Despliega una Ventana secundaria donde se puede visualizar un esquema del ciclo termodinámico de la bomba de calor. En la Figura 3.7 se puede apreciar la ventana secundaria descrita.



**Figura 3.7.** Ventana Secundaria: Esquema del circuito de la bomba de calor

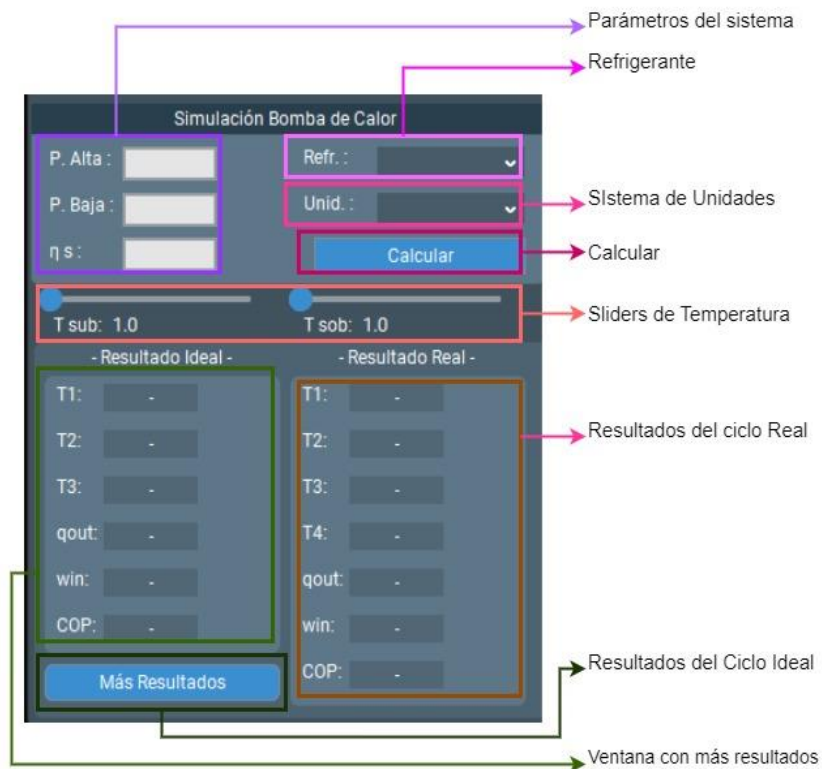
## Frame 2: Simulación con distintos refrigerantes

En la Figura 3.8 se muestra el frame destinado a la simulación del ciclo de compresión de vapor para una bomba de calor con distintos refrigerantes. El objetivo de esta sección es brindar al usuario la capacidad de modificar los parámetros principales del ciclo para así poder visualizar el comportamiento y desempeño según la configuración del sistema.

Parámetros del sistema: En esta sección se puede modificar la presión de alta, la presión de baja y la eficiencia isentrópica. Es importante que el ingreso de estos parámetros sean en las unidades que se seleccionará en la lista despegable de unidades

Refrigerantes: Lista despegable que contiene los refrigerantes disponibles para la simulación.





**Figura 3.8.** Frame 2: Simulación

Sliders de Temperatura: Las sliders están destinados a modificar la temperatura de sobrecalentamiento y la temperatura de subenfriamiento y solo afectan a los resultados del ciclo real.

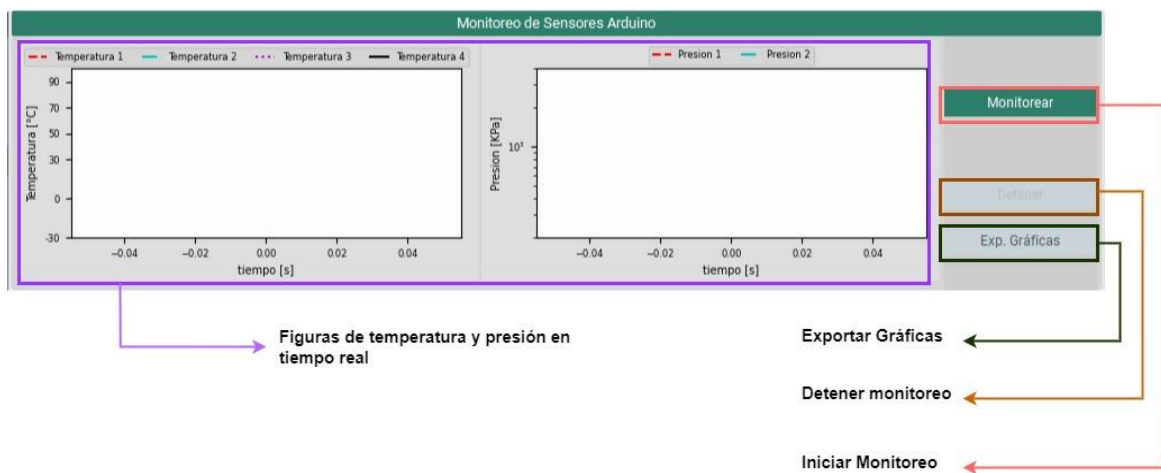
Ventana con más resultados: Se puede accionar para poder visualizar las variables termodinámicas de entalpía y entropía para cada punto del ciclo termodinámico. En la Figura 3.9 se muestra la ventana secundaria con más resultados.

Simulación Bomba de Calor											
Comportamiento Ideal					Comportamiento Real						
Temperatura		Entalpía		Entropía	Temperatura		Entalpía		Entropía		
T1:	-	h1:	-	s1:	-	T1:	-	h1:	-	s1:	-
T2:	-	h2:	-	s2:	-	T2:	-	h2:	-	s2:	-
T3:	-	h3:	-	s3:	-	T3:	-	h3:	-	s3:	-
T4:	-	h4:	-	s4:	-	T4:	-	h4:	-	s4:	-
Rendimiento					Rendimiento						
qout:					qout:						
win:					win:						
COP:					COP:						

**Figura 3.9.** Ventana con más resultados de la simulación

### Frame 3: Monitoreo de Sensores

En las anteriores secciones se detalló los frames con mayor interacción por parte del usuario, donde la modificación de los parámetros influye en la adquisición de datos y en la visualización de resultados. En las siguientes secciones se mostrará los frames destinados a la visualización de resultados del sistema de adquisición de datos y sus funciones principales. En la Figura 3.10 se muestra el frame destinado al monitoreo y visualización de la temperatura y presión en función del tiempo.



**Figura 3.10.** Frame 3: Monitoreo de los sensores del Arduino en tiempo real

Exportar Gráfica: Este comando una vez accionado despliega una ventana para seleccionar la ubicación donde se quiere guardar las gráficas de temperatura y presión. Por defecto las imágenes se almacenan en formato *.jpeg*.

Figuras: En ellas se muestran las gráficas de presión y temperatura. Se actualizan cada 10 segundos, debido a que se grafica el promedio de temperaturas y presiones cada 10 segundos.

Monitorear: Esta acción permite iniciar el monitoreo en tiempo real una vez establecida la conexión con el Arduino.

Detener: Esta función ayuda a pausar la actualización de la gráfica, para que el usuario pueda visualizar por más tiempo algún punto de interés.

### Frame 4: Visualización de Resultados

Este frame está destinado a la visualización de los cálculos y variables termodinámicas. Se puede visualizar los datos del ciclo ideal y real de la bomba de calor. En la Figura 3.11 se muestra el Frame 4 con las variables disponibles de visualización.

**Cálculos de Variables Termodinámicas en Tiempo Real**

Temperatura	Presion	Entalpía	Entropía
T1 : <input type="text"/>	P1 : <input type="text"/>	h1 : <input type="text"/>	s1 : <input type="text"/>
T2 : <input type="text"/>	P2 : <input type="text"/>	h2 : <input type="text"/>	s2 : <input type="text"/>
T3 : <input type="text"/>	P3 : <input type="text"/>	h3 : <input type="text"/>	s3 : <input type="text"/>
T4 : <input type="text"/>	P4 : <input type="text"/>	h4 : <input type="text"/>	s4 : <input type="text"/>

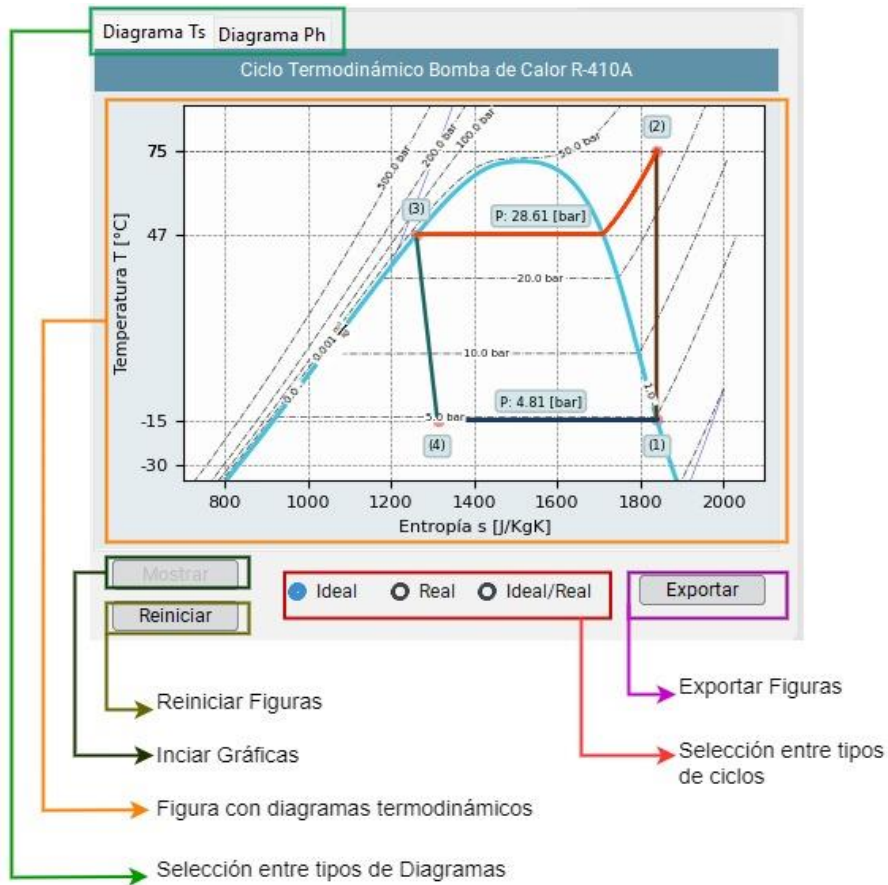
Resultados Ideales		Resultados Reales	
Win: <input type="text"/>		Win: <input type="text"/>	Pot.el: <input type="text"/>
qin: <input type="text"/>		qin: <input type="text"/>	η: <input type="text"/>
qout: <input type="text"/>		qout: <input type="text"/>	COP: <input type="text"/>
COP: <input type="text"/>			

→ Resultados Ideales
Resultados Reales ←

Variables Termodinámicas ←

**Figura 3.11.** Frame 4: Visualización de variables termodinámicas y resultados

**Frame 5: Diagramas Termodinámicos T-s y P-h**



**Figura 3.12.** Frame 5: Visualización de Diagramas termodinámicos

Este frame es desarrollado principalmente utilizando la librería fluprodia, por lo que permite la visualización de diagramas termodinámicos. La GUI desarrollada contempla la visualización de los diagramas T-s y Presión entalpía (Ph). En la Figura 3.12 se muestra el Frame 5 destinado a la visualización de diagramas termodinámicos y sus funciones.

Selección entre diagramas: El frame de diagramas termodinámicos es de tipo Notepad, cada pestaña permite la visualización de un tipo de diagrama termodinámico, que puede ser entre diagrama T-s y P-h.

Figura del diagrama termodinámico: Se visualiza el tipo de diagrama seleccionado y el tipo de ciclo termodinámico que selecciona el usuario.

Iniciar Gráfica: Es necesario, una vez establecida la conexión con el Arduino, iniciar la gráfica de los diagramas. Esta es una acción independiente del panel principal al iniciar la conexión para poder ahorrar recursos al sistema, puesto que fluprodia no está optimizado puede presentar inconvenientes si el computador utilizado no dispone suficientes recursos computacionales.

Reiniciar gráfica: comando que se debe utilizar junto con la opción de reiniciar práctica del panel principal.

Selección entre tipo de ciclo: El usuario puede seleccionar mediante un el tipo de ciclo a visualizar. La actualización del tipo de ciclo no es siempre inmediata ya que la actualización de la figura es cada 5 segundos.

Exportar gráfica: Las gráficas a exportar corresponden al tipo de ciclo seleccionado. Se exportan dos gráficas, una por cada tipo de diagrama, en formato JPEG en la ubicación que se requiera.

### **Complementos**

Adicional a la explicación de las funciones de la ventana “Práctica”, la ventana “Manual de Usuario” describe la secuencia de pasos correctos para empezar con el monitoreo de la bomba de calor y una pestaña con la información de la Sección 2.3 la cual se encuentra disponible dentro de la GUI y se puede observar en el **ANEXO IV**.

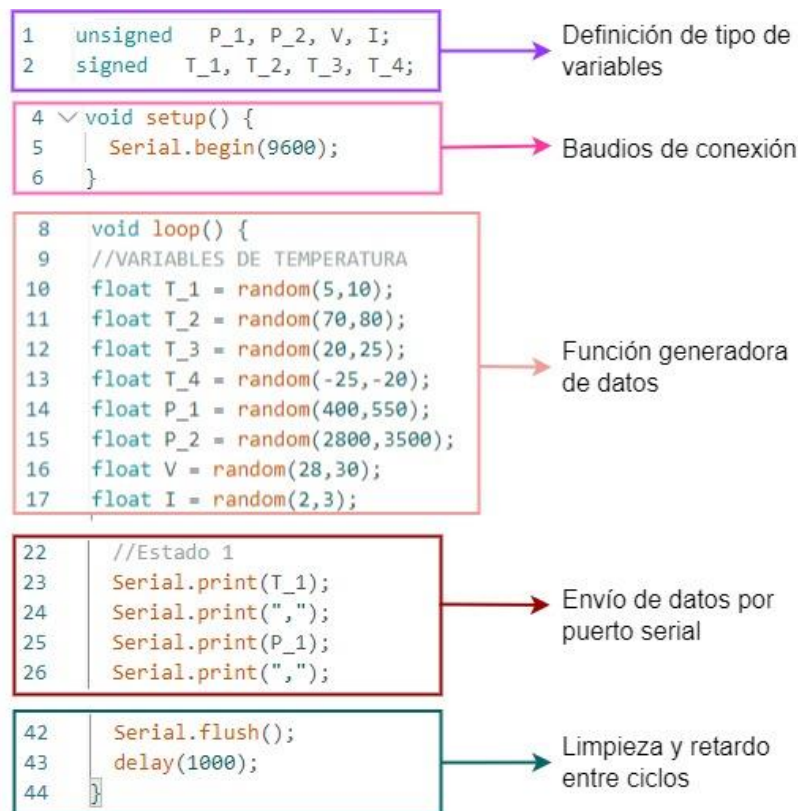
### **3.1.6 Simulación y prueba del software: Heat Pump Monitor**

Para comprobar el funcionamiento del software se realizó una simulación de lectura de datos con un Arduino Uno, que entrega datos mediante el puerto serial (USB), para validar el funcionamiento del software y verificar que cumpla con los requerimientos de la GUI.

## Programación en Arduino

La programación en Arduino resulta sencilla para simular datos que provengan de sensores por la cantidad de funciones incorporadas en el IDE de Arduino.

En la Figura 3.13 se puede observar las partes más relevantes del código para simular datos de sensores.



**Figura 3.13** Programación en Arduino para simulación de envío de datos

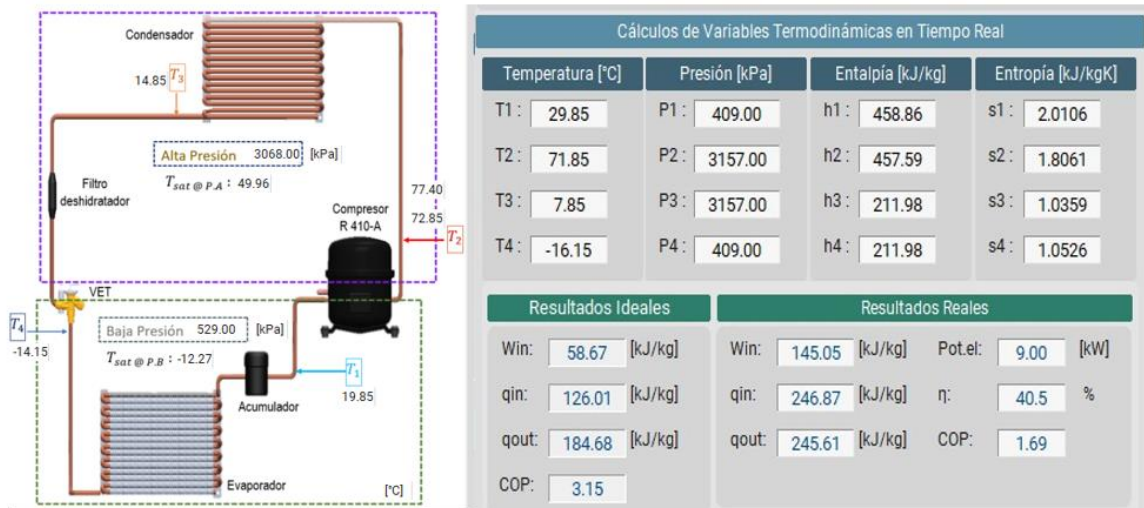
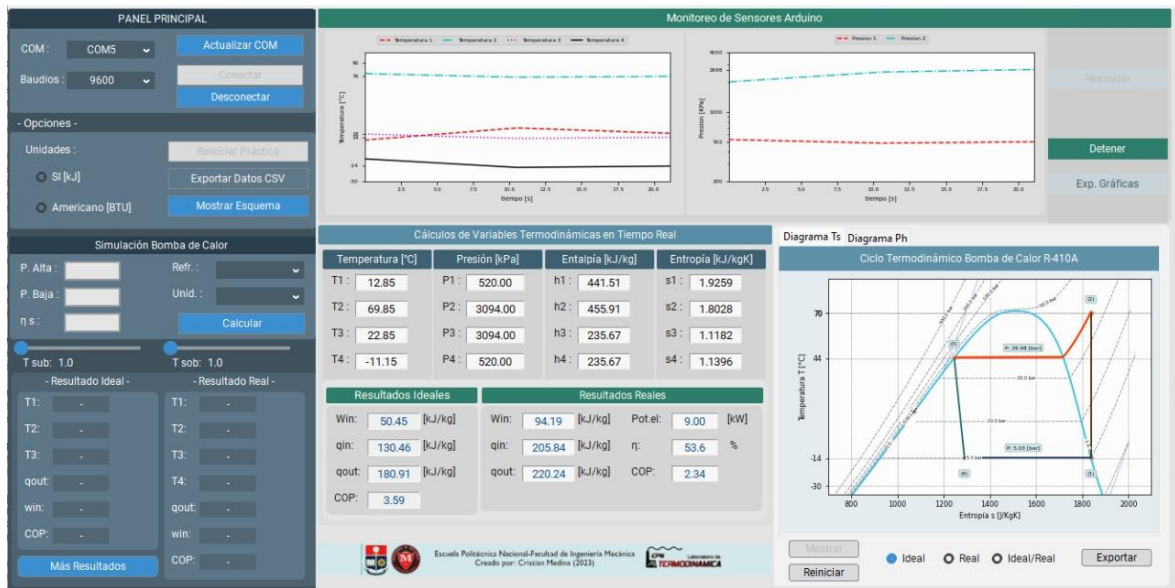
### Comprobación del funcionamiento del software: Heat Pump Monitor

Una vez programado el Arduino se puede comprobar el funcionamiento del software Heat Pump Monitor.

Es importante que, en la programación final de Arduino para el sistema didáctico, el envío de datos sea separado por “,” (comas) y en el siguiente  $T_1, P_1, T_2, P_2, T_3, V, T_4, I$  puesto que en la programación en Python el orden de los datos es importante para asignar a las variables correspondientes y que el cálculo de propiedades termodinámicas sea el correspondiente a cada estado termodinámico.

En la Figura 3.14 se puede observar el software puesto en marcha con datos simulados, sin embargo, el software está disponible y puede ser puesto a prueba por el lector teniendo

en consideración el orden para envío de datos de los sensores o datos simulados provenientes de un Arduino.



**Figura 3.14.** Verificación de funcionamiento del software Heat Pump monitor mediante datos simulados de una bomba de calor provenientes de un Arduino

Al utilizar el software desarrollado (Heat Pump Monitor), el usuario en la ventana "Práctica" es capaz de:

- Monitorear en tiempo real variables termodinámicas y almacenarlos en un archivo de texto para futuros análisis o registro de información.
- Visualizar los diagramas termodinámicos T-s y P-h y exportarlos en formato .jpeg.

- Visualizar un esquema de un sistema de bomba de calor para una mejor visualización y análisis de las variables termodinámicas y resultados en tiempo real.
- Visualizar el desempeño real e ideal en tiempo real de la bomba de calor que se implementará en el laboratorio de Termodinámica de la FIM-EPN.
- Simular el comportamiento y desempeño de una bomba de calor utilizando diferentes refrigerantes y variar los parámetros de funcionamiento.

Con las características descritas del programa, y con las ventanas adicionales de “Manual de Usuario” y “Cuestionario”, el software desarrollado cumple con los requisitos descritos del diseño de la GUI de la Sección 2.3.6 y el objetivo principal y específicos establecidos para el desarrollo del presente proyecto.

### 3.2 Conclusiones

Se desarrolló una herramienta informática para el análisis del comportamiento termodinámico real de una bomba de calor para el calentamiento de agua caliente sanitaria para el laboratorio de termodinámica de la FIM-EPN

Se recopiló referencias bibliográficas de proyectos de sistemas de adquisición de datos para monitoreo de sistemas energéticos, los cuales fueron guía para una correcta selección del lenguaje de programación y diseño de la GUI desarrollada.

Se determinó los parámetros necesarios para el análisis termodinámico de la bomba de calor y la ubicación de los sensores y los puntos necesarios de medición. Para el presente proyecto se optó por 4 puntos de medición de temperatura cerca de cada dispositivo principal del ciclo de refrigeración de vapor y dos sensores de presión a la entrada y salida del compresor. Estos parámetros permiten que el software desarrollado pueda calcular propiedades termodinámicas relevantes y el cálculo del COP de la bomba de calor.

Se desarrolló en Python un software ejecutable con interfaz gráfica utilizando el paradigma de programación orientada a objetos (POO) y se obtuvo un código para el software, disponible en el **ANEXO V**. El software es capaz de leer y registrar los datos entregados por un Arduino, realizar cálculos termodinámicos en tiempo real para el análisis termodinámico de la bomba de calor a implementar, permite la visualización de diagramas termodinámicos P-h y T-s en tiempo real y el monitoreo de los valores de presión y temperatura entregados por los sensores.

Se validó el funcionamiento del software mediante datos simulados en un Arduino, donde el software demostró ser capaz de cumplir con los requisitos establecidos.

Se desarrollaron las ventanas auxiliares “Marco teórico” y “Cuestionario”, las cuales permiten que el usuario tenga una mejor comprensión de los fundamentos asociados al funcionamiento de una bomba de calor.

El software cuenta con manual de usuario, dentro de la GUI, para ayudar en el manejo de la interfaz gráfica y que así pueda aprovechar todas las funciones disponibles del software Heat Pump Monitor.

### **3.3 Recomendaciones**

Puesto que está previsto que el sistema real esté montado con los respectivos sensores en el siguiente periodo académico se exhorta a poner a prueba el software con el banco de pruebas instalado y realizar un análisis sobre los posibles errores de cálculo y obtención del COP de la bomba de calor

Es importante que para la simulación con otros refrigerantes se tome en cuenta la variación de la eficiencia isentrópica a distintas condiciones de operación como presión y temperatura ambiente para que la simulación sea más precisa y didáctica. Para esto, se invita a los estudiantes que continúen con el desarrollo de este proyecto, revisar los modelos semiempíricos para la obtención de la eficiencia isentrópica descritos en la referencia [42], para el compresor utilizado en el sistema didáctico y mejorar la aproximación realizada en este proyecto.

El software se desarrolló utilizando fluprodia para los diagramas termodinámicos, y no tiene mucho tiempo de desarrollo, se recomienda actualizar la librería de fluprodia al momento del montaje del sistema didáctico para una mejor optimización del software Heat Pump Monitor.

Es conveniente utilizar Arduino como hardware DAQ, puesto que la decodificación en Python para los datos provenientes del DAQ está desarrollada para el formato de Arduino y pueda no ser compatible con otro hardware DAQ.



## 4 REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Blanco Bonilla *et al.*, “Panorama Energético de América Latina y el Caribe 2022,” Quito, 2022.
- [2] G. L. Mendoza, M. De Energía, Y. Minas, and X. V. Grunauer, “Balance Energético Nacional 2021,” Quito, jul. 2022. [Online]. Available: [www.rekursosyenergia.gob.ec](http://www.rekursosyenergia.gob.ec)
- [3] A. Ríos, J. Guamán, and C. Vargas, “Analysis of the implementation of a Strategy for Energy Consumption Reduction in the Residential Sector of Ecuador: Impact Evaluation in the Energy Matrix,” vol. I, no. 15, pp. 98–109, 2018.
- [4] Alexander Borja Cáceres, “Cálculo de la distribución espacial y temporal del consumo de energía en el sector residencial del Ecuador entre 2010 y 2050,” Trabajo experimental, Universidad Técnica de Ambato, Ambato, 2022.
- [5] J. Guamán, M. García, D. Guevara, and A. Ríos, “Evaluación del Impacto Económico en Diferentes Escenarios de Implementación de Tecnologías Eficientes de Calentamiento de Agua en el Ecuador.,” *Revista Técnica “Energía,”* vol. 1, pp. 270–283, 2016.
- [6] L. Manzano, “Estado actual y perspectivas de la energía solar térmica en el Ecuador,” 2011.
- [7] A. Medina, “Sistema eléctrico de calentamiento instantáneo de agua corriente con control inteligente de temperatura,” Instituto Politécnico Nacional, México, 2019.
- [8] J. García, C. Gómez, and F. Gómez, “Síntesis del Estudio Parque de Bombas de Calor en España,” Madrid, Dec. 2015. [Online]. Available: [www.idae.es](http://www.idae.es)
- [9] F.-B. I. Gorozabel-Chata and T. I. Carbonell-Morales, “Actualidad y perspectivas de una bomba de calor de expansión directa con energía solar,” *Ingeniería Mecánica,* vol. 19, no. 1, pp. 49–58, 2016, [Online]. Available:
- [10] Y. Çengel and M. Boles, *Termodinámica*, Séptima. México: McGRAW-HILL, 2012.
- [11] I. C. Figueroa, J. Fernández-Seara, and D. Fernández Cid, “Análisis teórico de los refrigerantes alternativos adecuados para bombas de calor de alta temperatura,” 2016. [Online]. Available: <http://www.uvigo.es>
- [12] G. Barletta and O. Acevedo, *Buenas prácticas en los procesos de instalación y mantenimiento de sistemas de refrigeración y aire acondicionado.* mcpceip2021.
- [13] ASHRAE, “Designation and safety classification of refrigerants,” Atlanta, Oct. 2019. [Online]. Available: [www.ashrae.org](http://www.ashrae.org)
- [14] C. Mateu-Royo, A. Mota-Babiloni, J. Navarro-Esbrí, and Á. Barragán-Cervera, “Comparative analysis of HFO-1234ze(E) and R-515 as low GWP alternatives to HFC,” *International Journal of Refrigeration,* vol. 124, pp. 197–206, Apr. 2021, doi: 10.16/j.ijrefrig.2020.12.023.
- [15] F. García *et al.*, “Eficiencia Energética en América Latina y el Caribe: Avance y Oportunidades,” Dec. 2017.
- [16] Y. S. González, E. R. Dunia-Amair, and J. A. Pérez-Rodríguez, “Prototipo didáctico para el aprendizaje de la mecánica en los laboratorios de física universitaria

- mediante un sistema de adquisición y procesamiento de datos,” *Revista Educación en Ingeniería*, vol. 12, no. 24, p. 9, jul. 2017, doi: 10.26507/rei.v12n24.781.
- [17] J. Fiallos, R. David Hernández, R. Ponce, and R. Mejía, “Sistemas de Adquisición de Datos Aplicados en la Medición de Variables Físicas,” *Revista de la Escuela de Física UNAH*, vol. 2, pp. 47–52, 2017.
- [18] R. Artés, F. Pérez, J. Sueiro, and R. López, *Señales y Sistemas*. 2015.
- [19] M. Sánchez, “Diseño e implementación de un banco de pruebas para el contraste de instrumentos de las variables físicas temperatura y presión,” Universidad Politécnica Salesiana, Guayaquil, 2015.
- [20] M. Abdallah and Elkeelany, “A Survey on Data Acquisition Systems DAQ,” *IEEE*, vol. 10, pp. 240–243, 2009.
- [21] J. Jerez and J. E. Vallejo, “Implementación del sistema de adquisición de datos en una máquina de ensayos de termofluencia para generar la curva de termofluencia en materiales ferrosos, 2021.
- [22] L. Yessid Alfonso Moreno Matilde Gutiérrez Luna, “Desarrollo de una tarjeta DAQ que permita interactuar en el monitoreo y control de las plantas FESTO MPS PA COMPACT,” Trabajo de titulación, Universidad Piloto de Colombia, Bogotá, 2015.
- [23] G. Romero, J. Cárdenas, and E. Espinel, “An innovative interactive educational software on refrigeration systems to support virtual thermodynamic Class ,” *Journal of Xi’an University of Architecture & Technology*, vol. 3, pp. 54–63, 2020.
- [24] H. Lastre and I. Xochimilco, “Desarrollo de un Software Didáctico para el Diseño de Plantas de Refrigeración de Gas Natural con Propano,” Universidad Industrial de Santander, 2017.
- [25] O. Pillajo, “Diseño y construcción de un prototipo portátil no invasivo para el análisis de sistemas de refrigeración por compresión de vapor para el laboratorio de termodinámica de la FIM-EPN, Quito, 2015.
- [26] M. Carreazo, G. Gómez, G. “Desarrollo de instrumento virtual enfocado en la adquisición de datos para generar perfiles de temperatura en hornos OVENS,” 2015.
- [27] F. Gortazar, R. Martínez, and V. Fresno, *Lenguajes de programación y procesadores*, 2nd ed. Madrid: Editorial Universitaria Ramón Areces, 2016.
- [28] Si. Yuill and H. Halpin, “Python,” 2006.
- [29] Ramírez Óscar, *Python a fondo*, II. Marcombo, 2021.
- [30] C. Guagliano, “Interfaz gráfica con Python,” in *Programación orientada a objetos en Python*, Peña. Claudia, Ed., USERS ebooks, 2019.
- [31] A. A. A. and A. B. A. M. Y. Taib, “Performance Analysis of a Domestic Refrigerator,” vol. 9501, pp. 320–325, May 2010.
- [32] M. Villalba, A. Urquía, and M. Rubio, *Lenguajes de Programación*, 1st ed. Madrid: Universidad Nacional de Educación a Distancia Madrid, 2021.

- [33] Desarrollo Web, “En qué se diferencian los lenguajes de scripting y de programación,” *Que son los lenguajes scripting*, 2020. <https://www.ionos.es> (accessed jul. 10, 2023).
- [34] CoolProp org, “Avaliable Wrappers for CoolProp,” *CoolProp*, 2022. <http://www.coolprop.org/coolprop/wrappers/index.html> (accessed Jul. 10, 2023).
- [35] C. Riba, *Diseño Concurrente*, 1st ed. Barcelona, España: Universidad Politécnica de Cataluña, 2002.
- [36] F. León, *Lógica y programación orientada a objetos: un inicio al desarrollo de software*. Fondo Editorial ITM, 2009.
- [37] KeepCoding, “Qué es Tkinter,” *Librerías para Python*. <https://keepcoding.io/blog/> (accessed Jul. 10, 2023).
- [38] elhacker.net, “Pyserial,” *Pyserial*. <https://wiki.elhacker.net/programacion/python/pyserial> (accessed Jul. 10, 2023).
- [39] Python org, “Documentación de Python - 3.8.17.” <https://docs.python.org/es/3.8/library/threading.html> (accessed jul. 10, 2023).
- [40] Python org, “PyPI,” *The Python package Index*. <https://pypi.org/> (accessed Jul. 10, 2023).
- [41] G. Bourges *et al.*, “Mecánica Computacional Vol XXV,” 2006.
- [42] F. M. Tello-Oquendo, D. C. Vásconez-Núñez, and E. Navarro-Peris, “Comparison of the influence of energy losses on efficiencies of a scroll compressor and a reciprocating compressor used in heat pumps,” *Informacion Tecnologica*, vol. 21, no. 5, pp. 205–214, Oct. 2020, doi: 10.4067/S0718-07642020000500205.
- [43] geeksforgeeks, “Matplotlib.animation.FuncAnimation class in Python,” *Matplotlib*. <https://www.geeksforgeeks.org/matplotlib-animation-funcanimation-class-in-python/> (accessed Jul. 11, 2023).
- [44] S. Sastoque, C. Narváez, and G. Garnica, “Metodología para la construcción de Interfaces Gráficas Centradas en el Usuario,” *Nuevas Ideas en Informática Educativa*, vol. 12, pp. 314–324, 2016.
- [45] International Organization for Standardization, “Human-centred design for interactive systems. Ergonomics of human-system interaction,” 2010.
- [46] Python org, “tkinter — Interface de Python para Tcl/Tk,” *Documentación Tk 3.11.4*. <https://docs.python.org/es/3/library/tkinter.html#important-tk-concepts> (accessed Jul. 12, 2023).
- [47] CustomTkinter, “Documentation Introduction,” *CTk Widgets*. <https://customtkinter.tomschimansky.com/documentation/widgets> (accessed Jul. 12, 2023).
- [48] GitHub, “What PyInstaller Does and How It Does It,” *Pysintaller*. <https://pyinstaller.org/en/stable/operating-mode.html> (accessed Jul. 12, 2023).

## 5 ANEXOS

### ANEXO I

#### Tablas y conclusiones de los criterios de selección para el lenguaje de programación

En este anexo se encuentran las tablas con los resultados obtenidos para la selección del lenguaje de programación por el *Método Ordinal Corregido de Criterios Ponderados*

#### Orden de criterios

**Tabla 5.1.** Tabla de ponderación entre los criterios de selección

Manejo > Paradigma > Sencillez = Versatilidad						
Criterio	Manejo del lenguaje	Paradigmas	Sencillez y facilidad	Versatilidad	$\Sigma+1$	Ponderación
Manejo del lenguaje		1	1	1	4	0,4
Paradigmas	0		1	1	3	0,3
Sencillez y facilidad	0	0		0,5	1,5	0,15
Versatilidad	0	0	0,5		1,5	0,15
<b>Total</b>					10	1

#### Manejo del lenguaje

**Tabla 5.2.** Tabla de ponderación entre lenguajes de programación para el criterio de: Manejo de lenguaje

Python > Visual Basic.NET = C++, C# = Java						
Manejo del lenguaje	Python	Visual Basic.NET	C++, C#	Java	$\Sigma+1$	Ponderación
Python		1	1	1	4	0,4
Visual Basic.NET	0		0,5	0,5	2	0,2
C++, C#	0	0,5		0,5	2	0,2
Java	0	0,5	0,5		2	0,2
<b>Total</b>					10	1

## Paradigmas de resolución de problemas

**Tabla 5.3.** Tabla de ponderación entre lenguajes de programación para el criterio de:  
Paradigmas de resolución de problemas

Python > Java > C++, C# = Visual Basic.NET						
Paradigmas	Python	Java	C++, C#	Visual Basic.NET	$\Sigma+1$	Ponderación
Python		1	1	1	4	0,4
Java	0		1	1	3	0,3
C++, C#	0	0		0,5	1,5	0,15
Visual Basic.NET	0	0	0,5		1,5	0,15
<b>Total</b>					10	1

## Sencillez del lenguaje y facilidad de aprendizaje

**Tabla 5.2.** Tabla de ponderación entre lenguajes de programación para el criterio de:  
Sencillez de lenguaje y facilidad de aprendizaje

Python > Visual Basic.NET = C++, C# = Java						
Sencillez y facilidad	Python	Visual Basic.NET	C++, C#	Java	$\Sigma+1$	Ponderación
Python		0,5	1	1	3,5	0,35
Visual Basic.NET	0		1	1	3,5	0,35
C++, C#	0	0		0,5	1,5	0,15
Java	0	0	0,5		1,5	0,15
<b>Total</b>					10	1

## Versatilidad en aplicaciones

**Tabla 5.3.** Tabla de ponderación entre lenguajes de programación para el criterio de:  
Sencillez de lenguaje y facilidad de aprendizaje

Python > Java > C++, C# = Visual Basic.NET						
Versatilidad	Python	Java	C++, C#	Visual Basic.NET	$\Sigma+1$	Ponderación
Java		1	1	1	4	0,4
Python			1	1	3	0,3
C++, C#	0	0		0,5	1,5	0,15
Visual Basic.NET	0	0	0,5		1,5	0,15
<b>Total</b>					10	1

## ANEXO II

### Elementos de la ventana “Marco Teórico”

En este apartado se muestra los elementos principales de la ventana “Marco Teórico” del software obtenido Heat Pump Monitor. Esta ventana es auxiliar al programa y parte de su contenido se puede observar en la Figura 5.1.

**PRODUCCION DE AGUA CALIENTE SANITARIA (ACS)**

- En Ecuador, entre el 70 al 80 % de la producción de ACS, se da mediante el uso de GLP con sistemas tipo calefón o caldero, mientras que el alrededor del 20 % se da mediante el uso de sistemas con resistencia eléctrica.
- Los sistemas de ACS se pueden clasificar según su fuente de energía

**SISTEMAS PARA PRODUCCIÓN DE ACS**

	Combustión	Electricidad	Captación Solar	Híbrido		
<b>Categoría</b>	Combustión Sólida	Combustión de Gas	Resistencia Eléctrica	Inducción magnética	Coletores solares	Compresor eléctrico
<b>Fuente</b>	Leña, carbón	GLP, GN	Hidroeléctricas, Termoeléctricas	Sol	Electricidad y ambiente	
<b>Tipo</b>	Caldera de leña o carbón	Calefón	Ducha eléctrica	Ducha de Inducción	Colector solar de tubos de vacío	Bomba de calor
<b>Eficiencia</b>	40 - 60 %	55 - 80 %	65 %	95 %	50 - 70 %	>200 %

En la lado derecho de la ventana "Marco Teórico" se encuentra teoría y conceptos fundamentales acerca de temas relacionados a bombas de calor

Al lado izquierdo se encuentra un menú, el cual ayuda a navegar entre los temas que son:

- \* Sistemas de Producción de Agua Caliente Sanitaria (ACS)
- \* Bombas de calor y su clasificación
- \* Funcionamiento de las bombas de calor
- \* Refrigerantes donde se encuentran los refrigerantes más utilizados en bombas de calor y circuitos de refrigeración

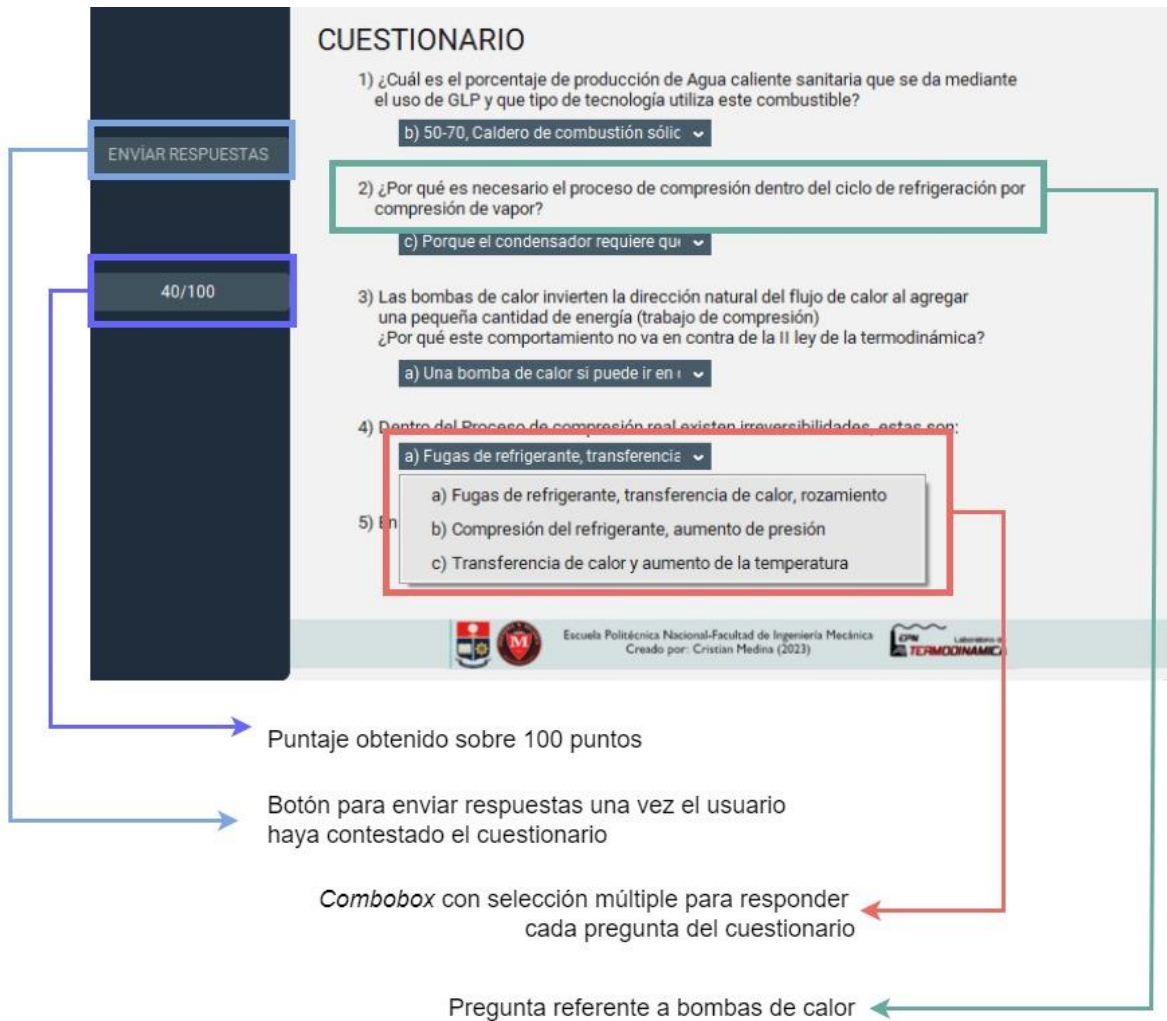
**Figura 5.1.** Ventana “Marco Teórico”

El usuario puede navegar a través del menú de la izquierda sin necesidad de seguir una secuencia específica.

## ANEXO III

### Elementos de la ventana “Cuestionario”

En este apartado se muestra los elementos principales de la ventana “Cuestionario” del software obtenido Heat Pump Monitor. Es una ventana auxiliar y se muestra en el Figura 5.2.



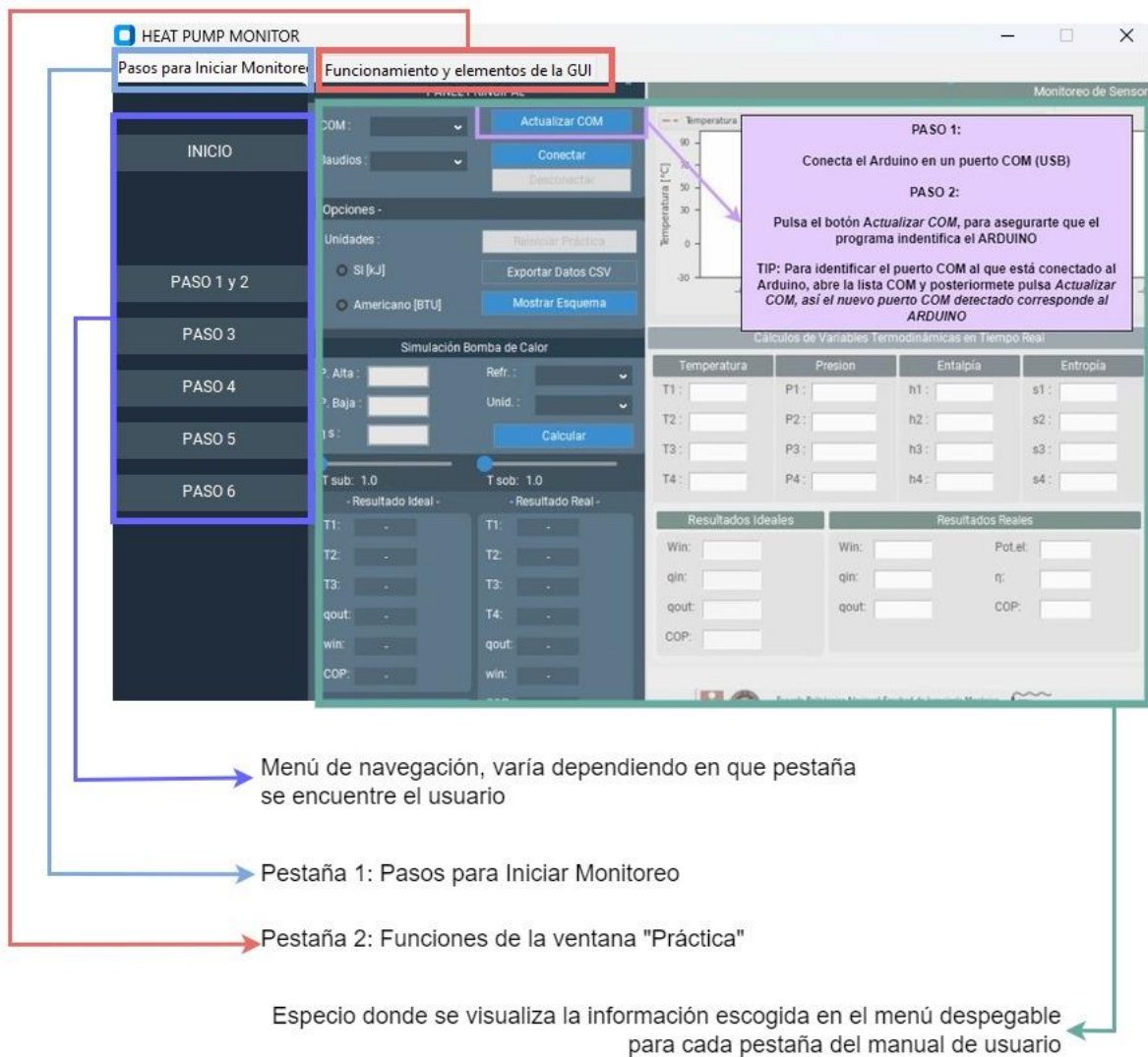
**Figura 5.2.** Ventana “Cuestionario”

El cuestionario está programado para admitir una sola respuesta por cada vez que se abre la ventana “Cuestionario”.

## ANEXO IV

### Elementos de la ventana “Manual de Usuario”

En este apartado se muestra los elementos principales de la ventana “Manual de Usuario”. En esta ventana se encuentra la secuencia de pasos a seguir para establecer la comunicación con el Arduino. Además, cuenta con otra pestaña “Funcionamiento y elementos de la GUI”, en esta pestaña se encuentra como utilizar las opciones disponibles de la ventana “Práctica” y el usuario pueda hacer uso de todas las funciones. En la Figura 5.3 se muestra los elementos de la ventana “Manual de Usuario”.



**Figura 5.3.** Ventana “Manual de Usuario”

Como se mencionó, esta ventana cuenta con dos pestañas, las cuales tienen información útil para el usuario.



## ANEXO V

### Código de programación en Python y ejecutable del software Heat Pump Monitor

#### Código de programación

En el siguiente enlace se encuentra disponible el código desarrollado en Python para la obtención del software Heat Pump Monitor. El archivo que permite abrir la aplicación tiene por nombre *Heat Pump Monitor.py*.

[https://epnecuador-my.sharepoint.com/:f:/g/personal/cristian\\_medina\\_epn\\_edu\\_ec/EIiHiAMHy1FFIjSkF\\_GHKrbcBlr\\_A\\_OPDGMuA-75ldsdQvg?e=uVecg8](https://epnecuador-my.sharepoint.com/:f:/g/personal/cristian_medina_epn_edu_ec/EIiHiAMHy1FFIjSkF_GHKrbcBlr_A_OPDGMuA-75ldsdQvg?e=uVecg8)

#### Ejecutable

Este software se obtuvo con el método de *Pyinstaller* de una sola carpeta y requiere todo el contenido de esta para su correcto funcionamiento.

Es importante que el usuario tenga en cuenta que el archivo ejecutable se encuentra en la misma carpeta que el resto de los archivos, y requiere abrir únicamente el archivo *Heat Pump Monitor.exe*.

[Heat Pump Monitor \(APP\).zip](#)