

**UNIVERSIDADE FEDERAL DA GRANDE DOURADOS
FACULDADE DE ENGENHARIA
ENGENHARIA MECÂNICA**

**PROTÓTIPO PARA ANÁLISE DE PARÂMETROS DE DESEMPENHO
DE UM CICLO BRAYTON DESENVOLVIDO EM LINGUAGEM
PYTHON**

LUCAS VASCONCELOS SALES

**DOURADOS
MATO GROSSO DO SUL
2023**

PROTÓTIPO PARA ANÁLISE DE PARÂMETROS DE DESEMPENHO DE UM CICLO BRAYTON DESENVOLVIDO EM LINGUAGEM PYTHON

Lucas Vasconcelos Sales

Orientador: Prof. Ramon Eduardo Pereira Silva

Trabalho de Conclusão de Curso apresentado à
Universidade Federal da Grande Dourados, como
parte dos requisitos para obtenção do título de
Bacharel em Engenharia Mecânica.

Dourados
Mato Grosso do Sul
2023



MINISTÉRIO DA EDUCAÇÃO
FUNDAÇÃO UNIVERSIDADE FEDERAL DA GRANDE DOURADOS

ANEXO D - AVALIAÇÃO FINAL DO TRABALHO DE CONCLUSÃO DE CURSO

Aluno: **LUCAS VASCONCELOS SALES**

Título do trabalho e subtítulo (se houver): **Protótipo para análise de parâmetros de desempenho de um ciclo Brayton desenvolvido em linguagem Python**

BANCA EXAMINADORA

1. Presidente (orientador):

Prof. Ramon Eduardo Pereira Silva, Universidade Federal da Grande Dourados - UFGD

2. Membro:

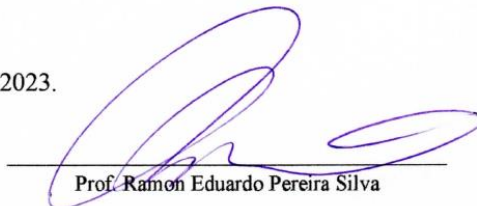
Prof. Augusto Salomão Bornschlegell, Universidade Federal da Grande Dourados – UFGD

3. Membro:

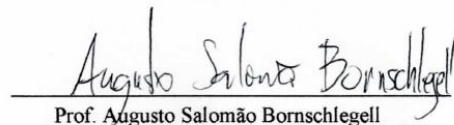
Prof. Antônio Carlos Caetano de Souza, Universidade Federal da Grande Dourados - UFGD

De acordo com o grau final obtido pelo aluno, nós da banca examinadora, declaramos APROVADO (Aprovado/Reprovado) o aluno acima identificado, na componente curricular Trabalho de Conclusão de Curso (TCC-II) de Graduação no Curso de Engenharia Mecânica da Universidade Federal da Grande Dourados.

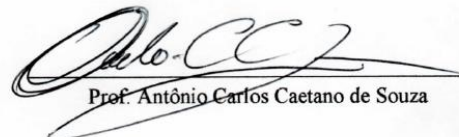
Dourados, 04 de maio de 2023.



Prof. Ramon Eduardo Pereira Silva



Prof. Augusto Salomão Bornschlegell



Prof. Antônio Carlos Caetano de Souza

Protótipo para análise de parâmetros de desempenho de um ciclo Brayton desenvolvido em linguagem Python

Lucas Vasconcelos Sales.¹

¹*Faculdade de Engenharia, Universidade federal da Grande Dourados
Dourados-MS, Brasil
lucasvasco68@hotmail.com*

Resumo. Este trabalho tem como objetivo produzir um protótipo que calcule os parâmetros de eficiência de um ciclo Brayton, desenvolvido em linguagem Python. O programa utiliza interface gráfica e tem a finalidade didática de ilustrar ciclos Brayton permitindo uma melhor compreensão de sua operação e projeto. O programa calcula os parâmetros de desempenho e dados de entalpia, temperatura e calor específico em cada estado, assim como também gera gráficos T-s e P-v. Utiliza como dados de entrada, temperatura ambiente, pressão ambiente, eficiência isentrópica do compressor, temperatura de saída da turbina, perda de carga no combustor, PCI do combustível, eficiência da turbina, fluxo mássico de ar e razão de compressão. Fornece como dados de saída *Heat rate*, potência consumida pelo compressor, potência produzida pela turbina, razão ar/combustível, razão combustível/ar, eficiência térmica, calor fornecido pelo combustível, vazão mássica de combustível, consumo específico e potência líquida, sendo esses parâmetros de desempenho calculados de duas formas, por calor específico médio e por entalpia específica.

Palavras-chave: Brayton, Python, Termodinâmica.

Abstract. This work aims to produce a prototype that calculates the efficiency parameters of a Brayton cycle, developed in Python language. The program uses a graphical interface and has the didactic purpose of illustrating Brayton cycles, allowing a better understanding of its operation and design. The program calculates performance parameters and enthalpy, temperature and specific heat data in each state, as well as generates T-s and P-v graphs. Using as input data, ambient temperature, ambient pressure, compressor isentropic efficiency, turbine outlet temperature, pressure drop in the combustor, fuel pci, turbine efficiency, air mass flow and compression ratio. Providing as output data Heat rate, power consumed by the compressor, power consumed by the turbine, air/fuel ratio, fuel/air ratio, thermal efficiency, heat supplied by the fuel, fuel mass flow, specific consumption and net power. These performance parameters are calculated in two ways, by average specific heat and by specific enthalpy.

Keywords: Brayton, Python, Thermodynamics.

1 Introdução

1.1 Objetivo

O objetivo deste memorial é descrever as necessidades, justificativas e requisitos de projeto para o desenvolvimento de um programa que calcule os parâmetros de desempenho de um ciclo Brayton aberto e real.

1.2 Necessidades específicas

O funcionamento e a análise energética de Turbinas a Gás (Gerador de Gases) são parte integrante da ementa da disciplina “Máquinas e Equipamentos Térmicos” tanto no curso de Engenharia de Energia quanto no curso de Engenharia Mecânica. Os balanços de massa e de energia são realizados de forma analítica pelo professor em um ponto de projeto determinado e para uma condição de operação. Desta forma, a necessidade de um programa que consiga fazer a simulação destes balanços, ainda que de forma analítica, mostrando a interferência decorrente da alteração das variáveis do processo na eficiência do ciclo Brayton torna-se premente de forma a facilitar o

entendimento da operação destes geradores de vapor. Assim, solicita-se o desenvolvimento de um protótipo de um programa de computador, em linguagem livre e aberta e que possua elementos gráficos e intuitividade com finalidade de auxílio pedagógico ao professor e aos alunos.

1.3 Requisitos de projeto

Assim, as necessidades apontadas anteriormente são traduzidas nos seguintes requisitos de projeto:

1. O programa deve permitir o cálculo dos parâmetros de desempenho de um ciclo Brayton.
2. O programa deve permitir o cálculo pelos seguintes métodos:
 - (a) Método das entalpias
 - (b) Método dos calores específicos médios.
3. O programa deve ser desenvolvido em linguagem aberta e gratuita.
4. O programa deve ter interface intuitiva para o usuário.
5. O programa deve permitir os seguintes dados de entrada utilizando as unidades descritas entre colchetes:
 - (a) Temperatura ambiente. [K]
 - (b) Pressão ambiente. [bar]
 - (c) Razão de pressão do compressor. [-]
 - (d) Eficiência isentrópica do compressor. [%]
 - (e) Vazão mássica de ar do compressor. [kg/s]
 - (f) Eficiência de combustão. [%]
 - (g) Perda de pressão no combustor. [%]
 - (h) Temperatura máxima na entrada da turbina. [K]
 - (i) Eficiência isentrópica da turbina. [%]
 - (j) Temperatura de saída da turbina. [K]
6. O programa deve mostrar os seguintes dados de saída utilizando as unidades descritas entre colchetes:
 - (a) Potência consumida pelo compressor. [kW]
 - (b) Potência fornecida pela turbina. [kW]
 - (c) Fluxo de calor fornecido pelo combustível. [kW]
 - (d) Potência líquida do gerador de gases. [kW]
 - (e) Vazão mássica de combustível. [kg/s]
 - (f) Vazão mássica dos gases de combustão. [kg/s]
 - (g) Razão ar/combustível. [-]
 - (h) Razão combustível/ar. [-]
 - (i) Consumo específico de combustível. [kg/s.kW]
 - (j) Heat Rate. [kJ/kWh]
 - (k) Eficiência térmica. [%]
 - (l) Temperaturas nos pontos do ciclo [K]
 - (m) Entalpias específicas nos pontos do ciclo [kJ/kg]
 - (n) Calores específicos médios [kJ/kg.K]
7. O programa deve mostrar o diagrama P-v.
8. O programa deve mostrar o diagrama T-s.

1.4 Validação

O programa deverá ser validado através da simulação de um exercício analítico cujos resultados deverão ser comparados e mostrados no projeto.

1.5 Entrega do projeto

Ao término do projeto deverão ser entregues:

1. O protótipo do programa operacional e validado.
 2. O relatório do projeto contendo:
 - (a) Objetivo do projeto.
 - (b) Fundamentação teórica.
 - (c) Validação do programa
-

3. Um tutorial de utilização do programa contendo:
 - (a) Instruções para instalação do programa.
 - (b) Instruções para utilização do programa.
 - (c) Exemplo da aplicação.
4. O script do programa em formato .pdf

2 O software

O software em questão foi desenvolvido através do uso da linguagem de programação Python. As bibliotecas utilizadas foram a Tkinter tendo o intuito de fornecer a interface gráfica do programa e a Pyromat visando fornecer as propriedades do ar (entalpia, entropia, pressão, calor específico). O programa tem por objetivo se tornar uma ferramenta com finalidade educativa, dessa forma ajudando discentes e docentes a compreender o funcionamento do ciclo Brayton com maior facilidade. Informando as temperaturas pedidas na tela inicial e as eficiências do compressor e turbina o programa consegue informar a eficiência térmica, o trabalho do compressor, trabalho da turbina e trabalho líquido assim como é plotado os gráficos pressão x volume específico e temperatura x entropia. A interface gráfica do usuário é mostrada na Fig 1, os parâmetros de entrada são adicionados em (I), os valores de temperatura, entalpia e calor específico referente à cada estado são mostrados em (II), os parâmetros de desempenho em (III), e os botões de executar, limpar, plot e exemplo com preenchimento automático são mostrados em (IV). Com o auxílio da biblioteca Pyinstaller o script se tornou em um programa executável onde não precisa do Python instalado no computador.

Figura 1. Interface gráfica do programa

The screenshot shows the 'Brayton Simples e real' software interface. It is organized into several functional areas:

- COMPRESSOR:** Includes input fields for 'Razão de compressão(P2/P1)', 'Temperatura ambiente (T0 ou T1)[K]', 'Pressão ambiente (P0 ou P1)[bar]', 'Eficiência do compressor[%]', and 'Vazão mássica de ar[kg/s]'. A red box labeled 'I' encompasses this section and the 'TURBINA' section.
- CÂMARA DE COMBUSTÃO:** Includes input fields for 'PCI do combustível[KJ/kg]', 'Eficiência de combustão [%]', 'Perda de carga na combustão [%]', and 'Temperatura de saída da câmara [K]'. A red box labeled 'II' encompasses this section and the 'TEMPERATURAS' section.
- TURBINA:** Includes input fields for 'Eficiência da turbina[%]' and 'Temperatura do gás na entrada da turbina[K]'. A red box labeled 'III' encompasses this section and the 'RESULTADOS CALOR ESPECÍFICO' section.
- TEMPERATURAS:** Includes input fields for 'T1=', 'T2 isentropica=', 'T2 real =', 'T3=', and 'T4='.
- ENTALPIAS:** Includes input fields for 'h1[kJ/kg]', 'h2[kJ/kg]', 'h3[kJ/kg]', and 'h4[kJ/kg]'.
- CALORES ESPECÍFICOS:** Includes input fields for 'cp, compressor=', 'cp, combustor=', and 'cp, turbina='.
- RESULTADOS ENALPIA:** Displays calculated values for 'Potencia do compressor=', 'Potencia da turbina=', 'Potência líquida do gerador de gases =', 'Heat rate=', 'Consumo específico=', 'Razão combustível/ar=', 'Vazão de gases na turbina=', 'Ganho de temperatura na câmara de combustão=', 'Quantidade de calor recebida pelos gases no interior da câmara =', and 'Eficiência térmica='.
- RESULTADOS CALOR ESPECÍFICO:** Displays calculated values for 'Potencia do compressor=', 'Potencia da turbina=', 'Potência líquida do gerador de gases =', 'Heat rate=', 'Consumo específico=', 'Razão combustível/ar=', 'Vazão de gases na turbina=', 'Ganho de temperatura na câmara de combustão=', 'Quantidade de calor recebida pelos gases no interior da câmara =', and 'Eficiência térmica='.
- Control Panel (IV):** Contains buttons for 'Executar', 'Exemplo', 'Limpar', 'PV plot', and 'T-s plot'.

Fonte: Autor, 2023

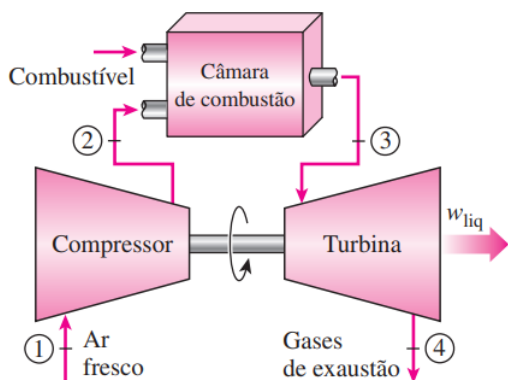
3 Fundamentação teórica

A base teórica do ciclo Brayton simples e com suas variações pode ser encontrada na literatura de Termodinâmica Clássica como Moran (2018), Çengel (2013), Borgnakke e Sonntag (2013). Para a implementação da interface gráfica usando a biblioteca Tkinter, o embasamento teórico teve como fonte a documentação da própria biblioteca no site oficial do Python. Desta forma, o embasamento teórico descrito nesta seção é baseado nas referências acima.

3.1 Ciclo básico

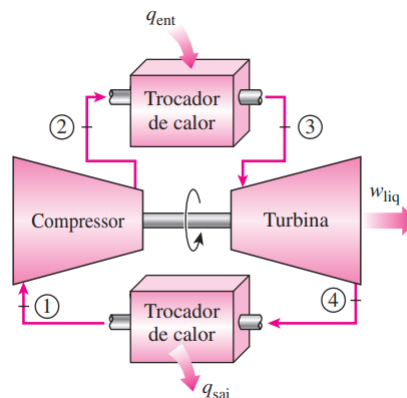
O ciclo Brayton simples pode ser modelado como ciclo fechado quando o ar que é o fluido de trabalho se mantém o mesmo durante os 4 processos principais que são: compressão isentrópica, fornecimento de calor a pressão constante, expansão isentrópica e rejeição de calor a pressão constante, conforme observado na Fig. 6. Quando se admite como ciclo aberto, o ar que é o fluido de trabalho é jogado fora (não recircula) e o ciclo é baseado em três processos principais: admissão de ar fresco em condições ambientes no compressor, combustão da mistura de ar e combustível à pressão constante e em seguida os gases resultantes entram na turbina onde se expandem até a pressão atmosférica produzindo potência, conforme observado na Fig. 3.

Figura 2. Ciclo aberto



Fonte: ÇENGEL e BOLES, 2013

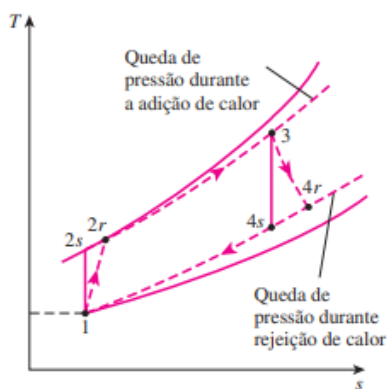
Figura 3. Ciclo fechado



Fonte: ÇENGEL e BOLES, 2013

Tanto o ciclo Brayton aberto quanto o ciclo fechado podem ser classificados em ciclo ideal ou ciclo real. “O ciclo da turbina a gás real difere do ciclo Brayton ideal em diversos aspectos. Um deles é que alguma queda de pressão durante os processos de adição e rejeição de calor é inevitável. Mais importante é o fato de que o trabalho de compressão real é maior e o trabalho realizado pela turbina é menor por causa das irreversibilidades”.(ÇENGEL; BOLES, 2013, p.513).

Figura 4. Diferença entre ciclo real e ideal



Fonte: ÇENGEL e BOLES, 2013

3.2 Cálculo de ciclo real

Temperatura isentrópica se trata da temperatura calculada sem acrescentar perdas do processo, essa temperatura no ponto 2 é obtida através da eq. (1).

$$T_{2s} = T_1 \frac{P_2}{P_1} \tag{1}$$

A eficiência do compressor descreve em porcentagem o quanto o compressor é capaz de converter a energia em trabalho e consequentemente o quanto de energia não é utilizado, conforme observado na eq. (2):

$$\eta_{comp} = \frac{T_{2s} - T_1}{T_{2r} - T_1}. \quad (2)$$

São feitas manipulações na eq. (2) como o objetivo de isolar T_{2r} . Dessa forma obtém-se na eq. (3) a temperatura no ponto 2 real, que no caso é diferente da temperatura isentrópica pois são consideradas as perdas do compressor.

$$T_{2r} = \frac{T_{2s} - T_1}{\eta_{comp}} + T_1. \quad (3)$$

Pressão no ponto 2 leva em consideração a pressão no ponto 1 e a razão de compressão que é o quanto a pressão P1 deve ser elevada e é descrita na eq. (4):

$$P_2 = RC \cdot P_1. \quad (4)$$

A potência que o compressor vai gastar para levar a pressão 1 ate a pressão 2 leva em consideração vazão mássica de ar, calor específico médio do ar que é a média entre os calores específicos do ar dos pontos 1 e 2 (antes e depois do compressor), e variação de temperatura do ar é descrito na eq. (5) quando é utilizado o método do calor específico médio, já quando é escolhido o método da entalpia específica é utilizado a diferença das entalpias entre os pontos 1 e 2 (antes e depois do compressor) e a vazão mássica de ar.

$$\dot{W}_{comp} = \dot{m}_{ar} \cdot cp_{frio} \cdot (T_{2r} - T_1) = \dot{m}_{ar} \cdot (h_{2r} - h_1). \quad (5)$$

Para calcular os valores de saída em relação à câmara de combustão foram usadas as fórmulas a seguir.

A eq. (6) calcula o ganho de temperatura na câmara de combustão.

$$\Delta T_{cc} = T_3 - T_2. \quad (6)$$

Quantidade de calor recebida pelos gases no interior da câmara utilizando o método do calor específico médio é descrita na eq. (7) onde \dot{m}_{ar} é o fluxo mássico de ar, cp_q é o calor específico médio do ar quente, que é calculado através da média aritmética do calor específico antes e depois da câmara de combustão, sendo equivalente à diferença das entalpias entre os pontos 3 e 2 (antes e depois da câmara de combustão) multiplicadas pela vazão mássica de ar.

$$Q_{ccg} = \dot{m}_{ar} \cdot cp_q \cdot (TIT - T_2) = \dot{m}_{ar} \cdot (h_3 - h_2). \quad (7)$$

Sabendo o fluxo mássico de combustível e o seu poder calorífico é possível obter a quantidade de calor liberada pelo combustível, mostrado na eq. (8).

$$Q_{cc} = \dot{m}_c \cdot PCI_c. \quad (8)$$

Razão combustível/ar se refere à porcentagem de combustível presente na mistura combustível/ar.

$$f = \frac{cp_q \cdot (TIT - T_2)}{\eta_{cc} \cdot PCI_c} = \frac{h_3 - h_2}{\eta_{cc} \cdot PCI_c}. \quad (9)$$

O volume de combustível por unidade de tempo que irá passar pela câmara de combustão é descrito por vazão mássica de combustível .

$$\dot{m}_c = \dot{m}_{ar} \cdot f. \quad (10)$$

Pressão na saída da câmara é descrita na eq. (11) onde ΔP_{cc} é a perda de pressão e P_2 é a pressão antes da câmara de combustão.

$$P_3 = P_2 \cdot (1 - \Delta P_{cc}). \quad (11)$$

Para calcular os valores relacionados à turbina

Vazão de gases da turbina, que é dada como a somatória de fluxo mássico de ar e de combustível descrito na eq. (12).

$$\dot{m}_{gturb} = \dot{m}_c + \dot{m}_{ar} \quad (12)$$

A potência gerada pela turbina é evidenciado na eq. (13), onde cp_q é calor específico médio do ar quente obtido pelas diferenças entre os calores específicos dos pontos 3 e 4 (antes e depois da turbina) \dot{m}_{gturb} é o fluxo mássico de gases, T_4 é a temperatura no ponto 4, dessa forma se torna equivalente a diferença entre as entalpias dos pontos 3 e 4 (antes e depois da turbina) multiplicadas pela vazão mássica de gases da turbina.

$$\dot{W}_{turb} = \dot{m}_{gturb} \cdot cp_q \cdot (T_4 - T_{IT}) = \dot{m}_{gturb} \cdot (h_4 - h_3) \quad (13)$$

Cálculo do gerador de gases:

Potência líquida é descrita na eq. (14) e se trata da diferença entre a potência consumida pelo compressor e a potência produzido pela turbina, e nesse caso também é considerado a eficiência mecânica.

$$\dot{W}_{liq} = (\dot{W}_{turb} - \dot{W}_{comp}) \cdot (\eta_{mec}) \quad (14)$$

A eficiência do ciclo é descrita na eq. (15) como a razão entre o trabalho líquido do processo e a quantidade de calor liberada pelo combustível.

$$\eta_{eng} = \frac{\dot{W}_{liq}}{\dot{m}_c \cdot PCI} \quad (15)$$

Consumo específico retrata a quantidade de combustível consumido por unidade de potência descrito na eq. (16).

$$sfc = \frac{\dot{m}_c}{\dot{W}_{liq}} \quad (16)$$

Heat rate é definido por quantidade de energia térmica consumida para gerar potência em um determinado tempo, mostrada em eq. (17).

$$HR = 3600 \cdot \frac{\dot{m}_c \cdot PCI}{\dot{W}_{liq}} \quad (17)$$

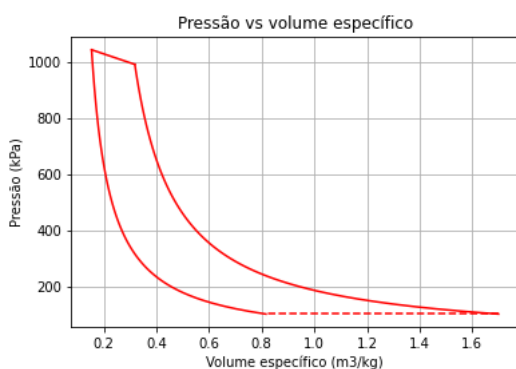
4 Resultados e discussões

Para poder verificar se o programa está calculando os dados do ciclo Brayton corretamente e validar os cálculos foram usados um exemplo analítico como forma de comparação. Analisando os dados presentes na tabela 1 é possível notar que não existe uma diferença significativa entre os valores obtidos a partir dos métodos da entalpia específica e do calor específico médio e os valores presentes no exemplo, assim como os gráficos T-s e P-v ilustram com fidelidade os estados do ciclo, portanto o programa é funcional.

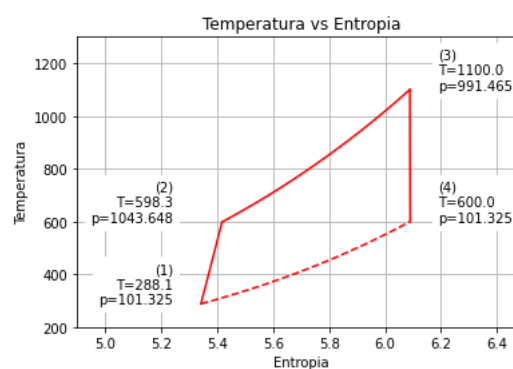
Tabela 1. Comparação para ciclo real

Parâmetro	Exemplo	Script cp	Script entalpia
T _{2s} [K]	561.05	561.05	561.05
T _{2r} [K]	598.26	598.26	598.26
P ₂ [kPa]	1043.64	1043.65	1043.65
W _c [kW]	33642.85	34401.41	3428.42
Δ_{tcc} [K]	501.74	501.73	501.73
Q _{tcc} [kW]	62256.21	59790.02	60021.67
f	0.014557	0.01398	0.01403
\dot{m}_c [kg/s]	1.572127	1.5098	1.5157
\dot{m}_{ar} [kg/s]	109.57	109.51	109.52
\dot{W}_{turb} [kW]	62943.71	60426.57	60660.92
\dot{W}_{liq} [kW]	29300.86	223162.39	23507.13
\dot{W}_{liq} [MW]	29.30	22.31	23.50
sfc [kg/kW]	5.37 E-05	6.519 E-05	6.423 E-05
η_t [%]	47.06	38.74	39.16
Heat rate [kJ/kWh]	7726.27	9386.69	9249.04

Fonte: Autor, 2023

Figura 5. Gráfico P-v gerado pelo *script*

Fonte: Autor, 2023

Figura 6. Gráfico T-s gerado pelo *script*

Fonte: Autor, 2023

5 Considerações finais

Conclui-se que a modelagem computacional do ciclo Brayton é efetiva, realizando os cálculos os resultados foram dentro do esperado, com grande facilidade de utilização, e o programa responde de forma rápida e sem travamentos, mas ainda necessita de aprimoramentos para abranger mais variedades de ciclos Brayton e gerar todos os gráficos pertinentes ao ciclo desejado.

6 Trabalhos futuros

6.1 Dispositivos móveis

A implementação desse script computacional juntamente com sua interface gráfica em *smartphones* vem com o objetivo de simplificar a análise de ciclos Brayton além de trazer portabilidade para a atividade.

6.2 Variações do ciclo Brayton

O presente projeto abordou ciclos reais apenas. Portanto acrescentar variações como resfriamento intermediário, regeneração, reaquecimento e turbina livre seria interessante para suprir as necessidades de pesquisa e trabalho existentes hoje em dia.

7 Referências

- BORGNAKKE, C., Sonntag . R. E. (2013). **Fundamentos da Termodinâmica** (Edgard Blucher Ltda. (ed.))
- ÇENGEL Y. A. e Boles M. A. (2013). **Termodinâmica**, Editora McGraw Hill: 7ª edição.
- MARTIN, C. R. **The PYroMat User and Developer Handbook** versão 2, v2.20. Pennsylvania State University, 2022
- MATTHES Eric, **Curso Intensivo de Python: Uma Introdução Prática e Baseada em Projetos à Programação**, Novatec Editora(16 maio 2016)
- MORAN M. J., Shapiro H. N., Boettner D. D. e Bailey M. B. (2014). **Princípios de Termodinâmica para Engenharia**, LTC Editora: 7ª edição, Rio de Janeiro.

Apêndice A

Script utilizado

```
from tkinter import *
from tkinter import ttk
import tkinter.ttk as ttk
import tkinter.messagebox
import pyromat as pyro
import numpy as np
import matplotlib.pyplot as plt

air = pyro.get('ig.air')

pyro.config['unit_energy'] = 'kJ'
pyro.config['unit_matter'] = 'kg'
pyro.config['unit_pressure'] = 'bar'
pyro.config['unit_temperature'] = 'K'

k=1.4 #razão entre calores específicos pro ar e 1.4

def calculos():

#ATRIBUINDO OS VALORES DE ENTRADA À VARIÁVEIS

Rp=float(textbox1.get()) #Razão de pressão
T1=float(textbox2.get()) #Temperatura antes do compressor
T4=float(textbox3.get()) #Temperatura antes da turbina
p1=float(textbox4.get()) #Pressão antes do compressor
nc=float(textbox9.get()) #Eficiência de compressão
ntt=float(textbox10.get()) #Eficiência mecânica
mar=float(textbox11.get()) #Fluxo mássico de ar
pci=float(textbox12.get()) #PCI do combustível
ncc=float(textbox13.get()) #Eficiência de compressão
tit=float(textbox15.get()) #TIT
dpcc=float(textbox14.get())

#CALOR ESPECÍFICO
```

```
cpf1=float (air.cp (T=T1))
#Cálculo da pressão depois do compressor
p2 = p1*Rp
#Temperatura 2 isoentropica
T2s=T1* ((p2/p1)**((k-1)/k))
#Temperatura 2 real
T2r=( (T2s-T1)/nc)+T1
#Calor especifico ponto 2
cpf2=float (air.cp (T=T2r))
#média dos calores especificos
cpfc=(cpf1+cpf2)/2
#Potencia do compressor utilizando a variação de temperatura
wc=mar*cpfc*(T2r-T1)
#Potencia do compressor utilizando a variação de entalpia
h1c=float (air.h (T1,p1))
h2c=float (air.h (T2r,p2))
wce=mar*(h2c-h1c)
#Ganho de temperatura na câmara de combustão
dtcc=(tit-T2r)
cptit=float (air.cp (T=tit))
cpcc=(cptit+cpf2)/2
#quantidade de calor recebida pelos gases utilizando temperatura
qccg=mar*cpcc*dtcc
#quantidade de calor recebida pelos gases utilizando entalpia
h1g=float (air.h (tit,p1))
dtcce=h1g-h2c
qccge=mar*dtcce
#Razão combustível/ar
f=(cpcc*dtcc)/(ncc*pci)
fi=1/f
#fe
fe=(dtcce)/(ncc*pci)
fei=1/fe
#Vazão mássica de combustível
mc=mar*f
mce=mar*fe
#quantidade de calor liberada pelo combustivel
qccc=mc*pci
#Pressão no ponto 3
P3=p2*(1-dpcc)

#Vazão de gases na turbina
mturb=mc+mar
mturbe=mce+mar

cp4=float (air.cp (T=T4))
cpturb=(cp4+cptit)/2
#Potencia da turbina utilizando temperatura
wturb=mturb*cpturb*(tit-T4)

#Potencia da turbina utilizando entalpia
h4t=float (air.h (T4,p1))
wturbe=mturb*(h1g-h4t)

#potencia líquida temperatura
wliq=(wturb-wc)*ntt

#potencia líquida entalpia
```

```
wliqe=(wturbe-wce)*ntt

#Eficiência do motor
nmotor=wliq/(mc*pci)

#Consumo específico
sfc=mc/wliq
sfce=mc/wliqe
#Heat rate
hr=3600*((mc*pci)/wliq)
hre=3600*((mc*pci)/wliqe)

Nt=(wliq/qccg)*100
Nte=(wliqe/qccge)*100

final1.set(str(round(T2s,2))+ " K")
final2.set(str(round(T2r,2))+ " K")
final3.set(str(round(p2,2))+ " kPa")
final4.set(str(round(mc,4))+ " Kg/s")
final5.set(str(round(wc,2))+ " kW")
final6.set(str(round(wturb,2))+ " kW")
final7.set(str(round(f,6)))
final8.set(str(round(mturb,2))+ " kg/s")
final9.set(str(round(qccg,2))+ " kW")
final10.set(str(round(wliq,2))+ " kW")
final11.set(str(round(sfc,8))+ " kg/kW")
final12.set(str(round(dtcc,2))+ " K")
final13.set(str(round(hr,2))+ " kJ/kWh")
final14.set(str(round(wce,2))+ " kW")
final15.set(str(round(wturbe,2))+ " kW")
final16.set(str(round(wliqe,2))+ " kW")
final17.set(str(round((h1c+302.75),2)))
final18.set(str(round((h2c+302.75),2)))
final19.set(str(round((h1g+302.75),2)))
final20.set(str(round((h4t+302.75),2)))
final21.set(str(round(T1,2))+ " K")
final22.set(str(round(T4,2))+ " K")
final23.set(str(round(tit,2))+ " K")
final24.set(str(round(hre,2))+ " kJ/kWh")
final25.set(str(round(sfce,8))+ " kg/kW")
final26.set(str(round(cpfc,4))+ " kJ/kg.K")
final27.set(str(round(cpcc,4))+ " kJ/kg.K")
final28.set(str(round(cpturb,4))+ " kJ/kg.K")
final29.set(str(round(fe,5)))
final30.set(str(round(mturbe,2))+ " kg/s")
final31.set(str(round(Nt,2))+ " %")
final32.set(str(round(Nte,2))+ " %")
final33.set(str(round(qccge,2))+ " kw")
final34.set(str(round(mce,4))+ " Kg/s")
final35.set(str(round(fi,4)))
final36.set(str(round(pei,4)))
#plot abal
```

```
def Pv_plot():
Rp=float(textbox1.get())      #Razão de pressão
T1=float(textbox2.get())     #Temperatura antes do compressor
T4=float(textbox3.get())     #Temperatura antes da turbina
p1=float(textbox4.get())     #Pressão antes do compressor
tit=float(textbox15.get())   #TIT
dpcc=float(textbox14.get())
#Cálculo da pressão depois do compressor
p2 = p1*Rp
#Temperatura 2 isoentropica
T2s=T1*((p2/p1)**((k-1)/k))
P3=p2*(1-dpcc)
T3=tit
R=0.287
p4=p1
#Volume específico
v1 = R * T1 / p1
v2 = R * T2s/ p2
v3 = R * T3 / P3
v4 = R * T4 / p4
T_12=np.linspace(T1,T2s,100)
T_34=np.linspace(T3,T4,100)
#1-2
plt.figure()
p12 = np.linspace(p1, p2, 100)
plt.plot(R*T_12/p12,p12,'r',linewidth=1.5)
#2-3
plt.plot([v2,v3],[p2,P3],'r',linewidth=1.5)
#3-4
p34 = np.linspace(P3, p4, 100)
plt.plot(R*T_34/p34,p34,'r',linewidth=1.5)
#4-1
plt.plot([v4,v1],[p4,p4],"r--",linewidth=1.5)
plt.grid('on')
plt.ylabel('Pressão (kPa)')
plt.xlabel('Volume específico (m3/kg)')
plt.title('Pressão vs volume específico')
plt.show()
#Preenchimento automático do exemplo
def ramon():

textbox1.delete(0, "end")
textbox1.insert(0, "10.3")

textbox2.delete(0, "end")
textbox2.insert(0, "288.15")

textbox3.delete(0, "end")
textbox3.insert(0, "600")

textbox4.delete(0, "end")
textbox4.insert(0, "101.325")

textbox9.delete(0, "end")
textbox9.insert(0, "0.88")

textbox10.delete(0, "end")
```

```
textbox10.insert(0,"0.89")

textbox11.delete(0,"end")
textbox11.insert(0,"108")

textbox12.delete(0,"end")
textbox12.insert(0,"40000")

textbox13.delete(0,"end")
textbox13.insert(0,"0.99")

textbox14.delete(0,"end")
textbox14.insert(0,"0.05")

textbox15.delete(0,"end")
textbox15.insert(0,"1100")
#Função limpaar
def limpar():
textbox1.delete(0,"end")
textbox2.delete(0,"end")
textbox3.delete(0,"end")
textbox4.delete(0,"end")
textbox9.delete(0,"end")
textbox10.delete(0,"end")
textbox11.delete(0,"end")
textbox12.delete(0,"end")
textbox13.delete(0,"end")
textbox14.delete(0,"end")
textbox15.delete(0,"end")
Label_R_T2S['text']='0'
Label_R_T2R['text']='0'
Label_R_P2['text']='0'
Label_R_MC['text']='0'
Label_R_WC['text']='0'
Label_R_WT['text']='0'
Label_R_F['text']='0'
Label_R_MT['text']='0'
Label_R_QG['text']='0'
Label_R_WL['text']='0'
Label_R_SFC['text']='0'
Label_R_DTCC['text']='0'
Label_R_HR['text']='0'

def T_s_plot():
#ATRIBUINDO OS VALORES DE ENTRADA À VARIÁVEIS

Rp=float(textbox1.get())      #Razão de pressão
T1=float(textbox2.get())     #Temperatura antes do compressor
T4=float(textbox3.get())     #Temperatura depoisda turbina
p1=float(textbox4.get())     #Pressão antes do compressor
nc=float(textbox9.get())     #Eficiência de compressão
tit=float(textbox15.get())   #TIT
dpcc=float(textbox14.get())

#Temperatura 3
T_3=tit
```

```
#Entropia 1
s1 = air.s(T1,p1)
#pressão 2
p2 = p1*Rp
#Temperatura 2 isoentropica
T2s=T1*((p2/p1)**((k-1)/k))
#Temperatura 2 real
T2r=(T2s-T1)/nc+T1
#Entropia 2
s2=air.s(T=T2r,p=p2)
#Entropia 3
s3 = air.s(T_3,p2)
#Temperatura 4
T_4=T4
#Pressão 3
p3=p2*(1-dpcc)
#Pressão 4
p4=p1
s4=air.s(T=T_4,p=p4)

plt.figure()
plt.plot([s1,s2],[T1,T2r],'r',linewidth=1.5)
T = np.linspace(T2r,T_3,20)
plt.plot(air.s(T=T,p=p2),T,'r',linewidth=1.5)
plt.plot([s3,s4],[T_3,T_4],'r',linewidth=1.5)
T = np.linspace(T1,T_4,20)
plt.plot(air.s(T=T,p=p1),T,'r--',linewidth=1.5)
plt.grid('on')
ax = plt.gca()
ax.set_xlim([4.9,6.5])
ax.set_ylim([200,1300])
# Adicionando labels de estado
plt.text(s1-.1,T1,'(1)\nT={:.1f}\nnp={:.3f}'.format(float(T1),float(p1)),
ha='right',backgroundcolor='white')
plt.text(s1-.1,T2r,'(2)\nT={:.1f}\nnp={:.3f}'.format(float(T2r),float(p2)),
ha='right',backgroundcolor='white')
plt.text(s3+.1,T_3,'(3)\nT={:.1f}\nnp={:.3f}'.format(float(T_3),float(p3)),
ha='left',backgroundcolor='white')
plt.text(s3+.1,T_4,'(4)\nT={:.1f}\nnp={:.3f}'.format(float(T_4),float(p4)),
ha='left',backgroundcolor='white')

plt.ylabel('Temperatura')
plt.xlabel('Entropia')
plt.title('Temperatura vs Entropia')
plt.show()

#Interface gráfica do usuário ABA1
root=Tk()
root.title("Brayton Simples e real ") #Título da janela
root.geometry("1300x800") #Dimensões da janela

final1=StringVar()
```

```
final2=StringVar()
final3=StringVar()
final4=StringVar()
final5=StringVar()
final6=StringVar()
final7=StringVar()
final8=StringVar()
final9=StringVar()
final10=StringVar()
final11=StringVar()
final12=StringVar()
final13=StringVar()
final14=StringVar()
final15=StringVar()
final16=StringVar()
final17=StringVar()
final18=StringVar()
final19=StringVar()
final20=StringVar()
final21=StringVar()
final22=StringVar()
final23=StringVar()
final24=StringVar()
final25=StringVar()
final26=StringVar()
final27=StringVar()
final28=StringVar()
final29=StringVar()
final30=StringVar()
final31=StringVar()
final32=StringVar()
final33=StringVar()
final34=StringVar()
final35=StringVar()
final36=StringVar()
#Widgets
#Label e textbox de dados de entrada

#frames
frame_3 = Frame( borderwidth=2, relief=SUNKEN)
frame_3.grid(row=1, column=2)
frame_2temp = Frame( borderwidth=2, relief=SUNKEN)
frame_2temp.grid(row=1, column=0)
frame_2ent = Frame( borderwidth=2, relief=SUNKEN)
frame_2ent.grid(row=1, column=1)
frame_21 = Frame( borderwidth=2, relief=SUNKEN)
frame_2 = Frame( borderwidth=2, relief=SUNKEN)
frame_2.grid(row=2, column=0)
frame_21.grid(row=2, column=1)
label_RE=Label(frame_2,text="RESULTADOS ENTALPIA",font="Arial 15",bd=1,relief="solid")
label_Rx=Label(frame_21,text="RESULTADOS CALOR ESPECÍFICO",font="Arial 15",bd=1,relief="solid")
label_ENT=Label(frame_2ent,text="ENTALPIAS",font="Arial 15",bd=1,relief="solid")
label_ENT.grid(row=0, column=0)
label_TEMPE=Label(frame_2temp,text="TEMPERATURAS",font="Arial 15",bd=1,relief="solid")
label_TEMPE.grid(row=0, column=0)
label_CPT=Label(frame_3,text="CALORES ESPECÍFICOS",font="Arial 13",bd=1,relief="solid")
```

```
frame_comp1 = Frame( borderwidth=2, relief=SUNKEN)
frame_comp1.grid(row=0, column=0)
frame_cc1 = Frame( borderwidth=2, relief=SUNKEN)
frame_cc1.grid(row=0, column=1)
frame_turb1 = Frame( borderwidth=2, relief=SUNKEN)
frame_turb1.grid(row=0, column=2)
frame_2cp = Frame( borderwidth=2, relief=SUNKEN)
frame_2cp.grid(row=1, column=2)
frame_but1 = Frame( borderwidth=2, relief=SUNKEN)
frame_but1.grid(row=2, column=2)

#TÍTULOS
label_COMPRESSOR=Label (frame_comp1,text="COMPRESSOR",font="Arial 15",bd=1,relief="solid")
label_TURBINA=Label (frame_turb1,text="TURBINA",font="Arial 15",bd=1,relief="solid")
label_CAMARA=Label (frame_cc1,text="CÂMARA DE COMBUSTÃO",font="Arial 15",bd=1,relief="solid")

#LABEL DE DADOS DE ENTRADA
label_RP=Label (frame_comp1,text="Razão de compressão (P2/P1) ")
textbox1=Entry (frame_comp1)
label_TA=Label (frame_comp1,text="Temperatura ambiente (T0 ouT1) [K] ")
textbox2=Entry (frame_comp1)
label_TG=Label (frame_turb1,text="Temperatura do gás na saída da turbina[K] ")
textbox3=Entry (frame_turb1)
label_PA=Label (frame_comp1,text="Pressão ambiente (P0 ouP1) [bar] ")
textbox4=Entry (frame_comp1)
label_NC=Label (frame_comp1,text="Eficiência do compressor[%] ")
textbox9=Entry (frame_comp1)
label_NT=Label (frame_turb1,text="Eficiência da turbina[%] ")
textbox10=Entry (frame_turb1)
label_MAR=Label (frame_comp1,text="Vazão mássica de ar[kg/s] ")
textbox11=Entry (frame_comp1)
label_PCI=Label (frame_cc1,text="PCI do combustivel[Kj/kg] ")
textbox12=Entry (frame_cc1)
label_NCC=Label (frame_cc1,text="Eficência de combustão [%] ")
textbox13=Entry (frame_cc1)
label_PC=Label (frame_cc1,text="Perda de carga na combustão [%] ")
textbox14=Entry (frame_cc1)
label_TS=Label (frame_cc1,text="Temperatura de saída da câmara [K] ")
textbox15=Entry (frame_cc1)

#BOTÕES
button1=Button (frame_but1,text="Executar",command=calculos)
button2=Button (frame_but1,text="Exemplo",command=ramon)
button3=Button (frame_but1,text="Limpar",command=limpar)
button4=Button (frame_but1,text="PV plot",command=Pv_plot)
button5=Button (frame_but1,text="T-s plot",command=T_s_plot)

#LABEL RESULTADO
Label_R_T1=Label (frame_2temp,textvariable=final21)
Label_R_T2S=Label (frame_2temp,textvariable=final1)
Label_R_T2R=Label (frame_2temp,textvariable=final2)
Label_R_T3=Label (frame_2temp,textvariable=final22)
Label_R_T4=Label (frame_2temp,textvariable=final23)
Label_R_P2=Label (frame_21,textvariable=final3)
Label_R_MC=Label (frame_21,textvariable=final4)
Label_R_WC=Label (frame_21,textvariable=final5)
Label_R_WT=Label (frame_21,textvariable=final6)
```

```
Label_R_F=Label (frame_21,textvariable=final7)
Label_R_MT=Label (frame_21,textvariable=final8)
Label_R_QG=Label (frame_21,textvariable=final9)
Label_R_WL=Label (frame_21,textvariable=final10)
Label_R_SFCC=Label (frame_21,textvariable=final11)
Label_R_DTCC=Label (frame_21,textvariable=final12)
Label_R_HR=Label (frame_21,textvariable=final13)
Label_R_WCE=Label (frame_2,textvariable=final14)
Label_R_WTE=Label (frame_2,textvariable=final15)
Label_R_WLE=Label (frame_2,textvariable=final16)
label_R_HRe=Label (frame_2,textvariable=final24)
label_R_SFCE=Label (frame_2,textvariable=final25)
Label_ent1=Label (frame_2ent,textvariable=final17)
Label_ent2=Label (frame_2ent,textvariable=final18)
Label_ent3=Label (frame_2ent,textvariable=final19)
Label_ent4=Label (frame_2ent,textvariable=final20)
Label_cp1=Label (frame_3,textvariable=final26)
Label_cp2=Label (frame_3,textvariable=final27)
Label_cp3=Label (frame_3,textvariable=final28)
Label_fe=Label (frame_2,textvariable=final29)
Label_mcee=Label (frame_2,textvariable=final30)
Label_R_DTCCe=Label (frame_2,textvariable=final12)
Label_R_nt=Label (frame_21,textvariable=final31)
Label_R_nte=Label (frame_2,textvariable=final32)
Label_R_qccge=Label (frame_2,textvariable=final33)
Label_R_mce=Label (frame_2,textvariable=final34)
Label_R_fi=Label (frame_21,textvariable=final35)
Label_R_fei=Label (frame_2,textvariable=final36)
#LABEL NOME DE RESULTADO
```

```
label_N_cpc=Label (frame_3,text="cp, compressor=")
label_N_cpcomb=Label (frame_3,text="cp, combustor=")
label_N_cpt=Label (frame_3,text="cp, turbina=")
label_N_qccge=Label (frame_2,text="Quantidade de calor recebida pelos gases no interior da câmara de combustão=")
label_N_fe=Label (frame_2,text="Razão combustível/ar=")
label_N_mcee=Label (frame_2,text="Vazão de gases na turbina=")
label_N_TITe=Label (frame_2,text="Ganho de temperatura na câmara de combustão=")
label_N_nt=Label (frame_21,text="Eficiência térmica=")
label_N_nte=Label (frame_2,text="Eficiência térmica=")
label_N_T3=Label (frame_2temp,text="T3=")
label_N_T4=Label (frame_2temp,text="T4=")
label_N_T1=Label (frame_2temp,text="T1=")
label_N_T2S=Label (frame_2temp,text="T2 isoentropica=")
label_N_T2R=Label (frame_2temp,text="T2 real =" )
label_N_P2=Label (frame_21,text="P2=")
label_N_WC=Label (frame_21,text="Potencia do compressor=")
label_N_F=Label (frame_21,text="Razão combustível/ar=")
label_N_Fi=Label (frame_21,text="Razão ar/combustível=")
label_N_Fei=Label (frame_2,text="Razão ar/combustível=")
label_N_MC=Label (frame_21,text="Massa de combustível=")
label_N_MCE=Label (frame_2,text="Massa de combustível=")
label_N_QG=Label (frame_21,text="Quantidade de calor recebida pelos gases no interior da câmara de combustão=")
label_N_TIT=Label (frame_21,text="Ganho de temperatura na câmara de combustão=")
label_N_WT=Label (frame_21,text="Potencia da turbina=")
label_N_MT=Label (frame_21,text="Vazão de gases na turbina=")
label_N_WL=Label (frame_21,text="Potência líquida do gerador de gases =" )
label_N_SFCC=Label (frame_21,text="Consumo específico=")
```

```
label_N_HR=Label (frame_21,text="Heat rate= ")
label_N_HRe=Label (frame_2,text="Heat rate= ")
label_N_WCE=Label (frame_2,text="Potencia do compressor=")
label_N_WTE=Label (frame_2,text="Potencia da turbina=")
label_N_WLE=Label (frame_2,text="Potência líquida do gerador de gases =")
label_N_SFCE=Label (frame_2,text="Consumo específico=")
Label_N_ent1=Label (frame_2ent,text="h1 [kJ/kg] ")
Label_N_ent2=Label (frame_2ent,text="h2 [kJ/kg] ")
Label_N_ent3=Label (frame_2ent,text="h3 [kJ/kg] ")
Label_N_ent4=Label (frame_2ent,text="h4 [kJ/kg] ")

#GRID
#BOTÕES
button1.grid(row=0,column=0)
button2.grid(row=1,column=0)
button3.grid(row=2,column=0)
button4.grid(row=3,column=0)
button5.grid(row=4,column=0)
#Label e textbox para dados iniciais
#TÍTULOS
label_COMPRESSOR.grid(row=0)
label_TURBINA.grid(row=0,column=0)
label_CAMARA.grid(row=0,column=0)

#DADOS DE ENTRADA E TEXTBOXES

label_RP.grid(row=1)
textbox1.grid(row=1,column=1)
label_TA.grid(row=2)
textbox2.grid(row=2,column=1)
label_TG.grid(row=2,column=0)
textbox3.grid(row=2,column=1)
label_PA.grid(row=3)
textbox4.grid(row=3,column=1)
label_NC.grid(row=4)
textbox9.grid(row=4,column=1)
label_NT.grid(row=1,column=0)
textbox10.grid(row=1,column=1)
label_MAR.grid(row=5)
textbox11.grid(row=5,column=1)
label_PCI.grid(row=1,column=0)
textbox12.grid(row=1,column=1)
label_NCC.grid(row=2,column=0)
textbox13.grid(row=2,column=1)
label_PC.grid(row=3,column=0)
textbox14.grid(row=3,column=1)
label_TS.grid(row=4,column=0)
textbox15.grid(row=4,column=1)

#Label nome do resultado

#TEMPERATURAS
label_N_T1.grid(row=1,column=0)
label_N_T2S.grid(row=2,column=0)
label_N_T2R.grid(row=3,column=0)
```

```
label_N_T3.grid(row=4, column=0)
label_N_T4.grid(row=5, column=0)
#RESULTADO
Label_R_T1.grid(row=1, column=1)
Label_R_T2S.grid(row=2, column=1)
Label_R_T2R.grid(row=3, column=1)
Label_R_T3.grid(row=5, column=1)
Label_R_T4.grid(row=4, column=1)
#ESTADO DE ENTALPIA
#NOME
Label_N_ent1.grid(row=1, column=0)
Label_N_ent2.grid(row=2, column=0)
Label_N_ent3.grid(row=3, column=0)
Label_N_ent4.grid(row=4, column=0)
#RESULTADO
Label_ent1.grid(row=1, column=1)
Label_ent2.grid(row=2, column=1)
Label_ent3.grid(row=3, column=1)
Label_ent4.grid(row=4, column=1)
#LABEL21 RESULTADOS CALOR ESPECIFICO
#NOME
label_N_WC.grid(row=2, column=0)
label_N_WT.grid(row=3, column=0)
label_N_WL.grid(row=4, column=0)
label_N_HR.grid(row=5, column=0)
label_N_SFC.grid(row=6, column=0)
label_N_F.grid(row=7, column=0)
label_N_MT.grid(row=8, column=0)
label_N_TIT.grid(row=9, column=0)
label_N_QG.grid(row=10, column=0)
label_N_MC.grid(row=11, column=0)
label_N_nt.grid(row=12, column=0)
label_N_Fi.grid(row=13, column=0)
#RESULTADO
Label_R_WC.grid(row=2, column=1)
Label_R_WT.grid(row=3, column=1)
Label_R_WL.grid(row=4, column=1)
Label_R_HR.grid(row=5, column=1)
Label_R_SFC.grid(row=6, column=1)
Label_R_F.grid(row=7, column=1)
Label_R_MT.grid(row=8, column=1)
Label_R_DTCC.grid(row=9, column=1)
Label_R_QG.grid(row=10, column=1)
Label_R_MC.grid(row=11, column=1)
Label_R_nt.grid(row=12, column=1)
label_Rx.grid(row=1, column=0)
Label_R_fi.grid(row=13, column=1)
#label 2 resultados entalpia
label_RE.grid(row=1, column=0)
label_N_WCE.grid(row=2, column=0)
label_N_WTE.grid(row=3, column=0)
label_N_WLE.grid(row=4, column=0)
label_N_HRe.grid(row=5, column=0)
label_N_SFCE.grid(row=6, column=0)
label_N_fe.grid(row=7, column=0)
label_N_mcee.grid(row=8, column=0)
label_N_TITe.grid(row=9, column=0)
label_N_qccge.grid(row=10, column=0)
```

```
label_N_MCE.grid(row=11, column=0)
label_N_nte.grid(row=12, column=0)
label_N_Fei.grid(row=13, column=0)
#resultados
Label_R_WCE.grid(row=2, column=1)
Label_R_WTE.grid(row=3, column=1)
Label_R_WLE.grid(row=4, column=1)
label_R_HRe.grid(row=5, column=1)
label_R_SFCE.grid(row=6, column=1)
Label_fe.grid(row=7, column=1)
Label_mcee.grid(row=8, column=1)
Label_R_DTCCe.grid(row=9, column=1)
Label_R_nte.grid(row=12, column=1)
Label_R_mce.grid(row=11, column=1)
Label_R_qccge.grid(row=10, column=1)
Label_R_fei.grid(row=13, column=1)
#RESULTADOS CP
#NOME
label_CPT.grid(row=0, column=0)
label_N_cpc.grid(row=1, column=0)
label_N_cpcomb.grid(row=2, column=0)
label_N_cpt.grid(row=3, column=0)

#RESULTADOS

Label_cp1.grid(row=1, column=1)
Label_cp2.grid(row=2, column=1)
Label_cp3.grid(row=3, column=1)

root.mainloop()
```