



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA QUÍMICA
CURSO DE ENGENHARIA QUÍMICA

ARTUR AMARAL CORRÊA DE MORAES

APLICATIVO PYTHON PARA SUPORTE PEDAGÓGICO NO CÁLCULO DE
EQUAÇÕES DE ESTADO CÚBICAS

FORTALEZA

2022

ARTUR AMARAL CORRÊA DE MORAES

**APLICATIVO PYTHON PARA SUPORTE PEDAGÓGICO NO CÁLCULO DE
EQUAÇÕES DE ESTADO CÚBICAS**

Trabalho de Conclusão do Curso de Graduação
em Engenharia Química do Centro de
Tecnologia da Universidade Federal do Ceará,
como requisito parcial à obtenção do grau de
Engenheiro Químico.

Orientadora: Profa. Dra. Andrea da Silva Pereira

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- M818a Moraes, Artur Amaral Corrêa de.
Aplicativo Python para suporte pedagógico no cálculo de equações de estado cúbicas / Artur Amaral Corrêa de Moraes. – 2022.
83 f. : il. color.
- Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Química, Fortaleza, 2022.
Orientação: Profa. Dra. Andrea da Silva Pereira.
Coorientação: Prof. Dr. João José Hiluy Filho.
1. Termodinâmica. 2. Equações de estado cúbicas. 3. Python. 4. Van der Waals. 5. Redlich-Kwong. I.
Título.

CDD 660

ARTUR AMARAL CORRÊA DE MORAES

**APLICATIVO PYTHON PARA SUPORTE PEDAGÓGICO NO CÁLCULO DE
EQUAÇÕES DE ESTADO CÚBICAS**

Trabalho de Conclusão do Curso de Graduação em Engenharia Química do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de Engenheiro Químico.

Aprovado em: ___ / ___ / _____

BANCA EXAMINADORA

Profa. Dra. Andrea da Silva Pereira (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. João José Hiluy Filho
Universidade Federal do Ceará (UFC)

Prof. Dr. Rodrigo Silveira Viera
Universidade Federal do Ceará (UFC)

À minha tia e amiga Iracema,
que partiu no dia 23 de março de 2015.

AGRADECIMENTOS

A Deus, pela minha vida, e por me permitir ultrapassar todos os obstáculos encontrados ao longo da realização deste trabalho.

Aos meus pais, Libia e Jacques, minha avó Irandi e minhas tias: Rejane e Laís, por todo o apoio e pela ajuda ao longo da vida.

Às pessoas com quem convivi ao longo desses anos de curso, que me incentivaram e que certamente tiveram impacto na minha formação acadêmica.

Aos meus amigos: Alan, Brenda, Emerson, Fábio, Felipe, Henrique, Higor, Ítalo, Joanna Karen, João Victor, João Marcos, Karla, Karlos, Maria Rafaela, Lucivando, Pedro, Raimundo, Renato, Roniele e a quem eu esqueci de mencionar.

A minha orientadora, Professora Andrea, que conduziu o trabalho com paciência e dedicação, sempre disponível a compartilhar todo o seu vasto conhecimento. Ao Professor Hiluy, essencial no meu processo de formação profissional, pela dedicação, e por tudo o que aprendi ao longo dos anos do curso.

Aos professores, por todos os conselhos, pela ajuda e pela paciência com a qual guiaram o meu aprendizado.

RESUMO

As ciências e engenharias transformaram-se nas últimas décadas com o desenvolvimento tecnológico e também o advento da programação. A engenharia química foi impactada tanto na atividade industrial, bem como nas pesquisas científicas e em seu ensino, com o surgimento de softwares, cada vez mais modernos, capazes de solucionar problemas por meio de cálculos computacionais. Nesse contexto, o ensino tradicional ganha novas possibilidades com o acréscimo de outras estratégias de aprendizagem. Dentre as áreas de conhecimento da Engenharia Química, escolheu-se a disciplina de Termodinâmica Química para inserir-se a utilização de um programa auxiliar no cálculo de equações de estado cúbicas. Essas equações descrevem e preveem propriedades de misturas de líquidos e gases a partir de propriedades tais como temperatura, pressão e volume, caracterizando um estado termodinâmico, auxiliando por exemplo no dimensionamento de equipamentos industriais dos mais diversos. Selecionou-se Equações de Estado Cúbicas tais como Van der Waals, Soave-Redlich-Kwong, Peng-Robinson, dentre outras, que compreendessem a maior faixa possível das propriedades físicas citadas, fizessem uso de diferentes parâmetros e considerações e que englobasse a lista dos fluidos mais utilizados na indústria. Foi utilizado a linguagem computacional Python e bibliotecas de código aberto como Thermo e Tkinter, referências em suas áreas, para construir-se uma interface gráfica que permitisse ao usuário obter resultados intermediários de seus problemas, valores específicos onde ocorra mudanças de estado ou pontos críticos e gráficos, possa comparar os valores obtidos de acordo com o modelo escolhido e verificar o erro associado, por exemplo, ao selecionar uma faixa inapropriada para a equação.

PALAVRAS-CHAVE: Termodinâmica; Equações de Estado Cúbicas; Python. Van der Waals; Redlich-Kwong.

ABSTRACT

Science and engineering have changed in recent decades with technological development and also the advent of programming. Chemical Engineering and its activity was impacted both in industrial activity, as well as in scientific research and teaching, with the emergence of increasingly modern software capable of solving problems through computational calculations. In this context, traditional teaching gains possibilities with the addition of other learning strategies. Among the areas of knowledge of Chemical Engineering, the discipline of Chemical Thermodynamics was chosen to insert the use of an auxiliary program in the calculation of Cubic Equations of State. These equations describe and predict properties of mixtures of liquids and gases from properties such as temperature, pressure and volume, characterizing a thermodynamic state, helping, for example, in the design of the most diverse industrial equipment. Cubic Equations of State such as Van der Waals, Soave-Redlich-Kwong, Peng-Robinson, among others, were selected, covering the widest possible range of the aforementioned physical properties, making use of different parameters and considerations and encompassing the list of fluids most used in the industry. For this, the Python computational language and open source libraries such as Thermo and Tkinter will be used, references in their areas, to build a graphical interface that allows the user to obtain intermediate results of their problems, specific values where state changes or critics points occur, graphics, etc. and can compare the values obtained according to the chosen model and verify the associated error, for example, when selecting an inappropriate range for the equation.,.

KEYWORDS: Thermodynamics; Cubic Equations of State; Python; Van der Waals; Redlich-Kwong.

LISTA DE FIGURAS

Figura 1- Superfície das fases	19
Figura 2- Logo da biblioteca Tkinter	23
Figura 3- Interface default da aplicação.....	24
Figura 4- Home da biblioteca Thermo	26
Figura 5- Home do banco de dados PubChem.....	27
Figura 6- Buscador do banco de dados PubChem	27
Figura 7- Logo da Aplicação	28
Figura 8- Fluxograma da Aplicação.....	29
Figura 9- Menu de Ajuda.....	30
Figura 10- Tela Inicial.....	30
Figura 11- Tela dos Resultados.....	31
Figura 12- Menu Sobre	31
Figura 13- Enunciado do Problema	32
Figura 14- Planilha volume molar do nitrogênio	33
Figura 15- Tela inicial com os dados de entrada do problema	33
Figura 16- Tela de resultados com o volume molar de nitrogênio	34
Figura 17- Planilha volume molar do oxigênio	36
Figura 18- Tela de início com os dados de entrada do problema	37
Figura 19- Tela de resultados com o volume molar de oxigênio.....	37
Figura 20- Enunciado do Problema	39
Figura 21- Tela inicial com os dados de entrada do problema	42
Figura 22- Tela de resultados com o volume de Etano.....	42
Figura 23- Tela inicial com os dados de entrada do sistema	44
Figura 24- Tela de resultados com o volume de Etano.....	44
Figura 25- Enunciado do Problema	45
Figura 26- Planilha com o volume molar de Propano	45
Figura 27- Tela Inicial com os valores de entrada do problema.....	46
Figura 28- Tela de resultados com o volume de Propano.....	46
Figura 29- Tela de resultado e gráfico com o volume molar de Propano.....	46
Figura 30- Tela de início com os dados de entrada do problema	50
Figura 31- Tela de resultado com o volume molar de Propano	50

Figura 32- Tela de resultado e gráfico com o volume molar de Propano	51
Figura 33- Enunciado do Problema	52
Figura 34- Tela de início com os valores de entrada do problema	54
Figura 35- Tela de resultado com o volume molar da Amônia	55
Figura 36- Tela de início com a seleção das Equações.....	55
Figura 37- Tela de resultado com volume molar da Amônia	56
Figura 38- Enunciado do Problema	56
Figura 39- Planilha com o volume molar da Mistura	58
Figura 40- Tela de resultado com volume molar da Mistura.....	59
Figura 41- Tela de resultado com volume molar da Mistura.....	59
Figura 42- Erros associados de cada problema.....	60

LISTA DE SÍMBOLOS

α	Parâmetro de interação binária
β	Fase do sistema
ω	Fator de acentricidade
σ	Distância potencial de Sutherland
f	Fugacidade
φ	Coefficiente de fugacidade
a	Parâmetro de atração
b	Parâmetro de atração
J	Joule
h	Entalpia
s	Entropia
g	Energia Livre de Gibbs
m	Massa
n	Número de mols
r	Constante dos Gases
V	Volume
v	Volume molar
T	Temperatura
T_c	Temperatura Crítica
P_c	Pressão Crítica
T_r	Temperatura Reduzida
P_r	Pressão Reduzida
P	Pressão
Z	Fator de Compressibilidade

SUMÁRIO

1	INTRODUÇÃO.....	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Equação do Gás Ideal.....	14
2.2	Equação de Van der Waals	14
2.3	Equação de Redlich-Kwong.....	16
2.4	Equação de Soave Redlich-Kwong.....	17
2.5	Equação de Peng-Robinson.....	18
2.6	Equilíbrio de Fases.....	19
2.7	Cálculo para Misturas	20
2.8	Fugacidade.....	21
3	METODOLOGIA	23
3.1	TKinter	23
3.2	Thermo.....	25
3.2	PubChem	26
3.3	Thermo PySoup.....	28
4	RESULTADOS	32
4.1	Companhia de Gás Castor - Nitrogênio.....	32
4.2	Companhia de Gás Castor - Oxigênio.....	36
4.3	Determinação do Volume Molar de Etano – Item A	39
4.4	Determinação do Volume Molar de Etano – Item B	43
4.5	Determinação do Volume Molar de Propano – Redlich-Kwong.....	45
4.6	Determinação do Volume Molar de Propano – Peng-Robinson.....	48
4.7	Determinação do Volume Molar da Amônia – Soave Redlich-Kwong e Peng-Robinson	52
4.8	Determinação do Volume Molar do Acetileno – Redlich-Kwong	56
4.9	Compilação dos Erros	60
5	CONCLUSÃO.....	61
	REFERÊNCIAS.....	66
	APÊNDICE A – CÓDIGO DO PROGRAMA	63

1 INTRODUÇÃO

O desenvolvimento das ciências e suas tecnologias promoveu significativas mudanças na sociedade ao longo dos séculos e alcançou feitos outrora improváveis. Desde supercomputadores quânticos ao pouso de uma sonda na superfície de um cometa viajando a 135 mil km/h, o conhecimento gerado nessas pesquisas se dispersa no nosso cotidiano. A World Wide Web, por exemplo, que foi pensada na década de 80 e surgiu em um laboratório europeu da CERN, hoje é utilizada por mais de 5 bilhões de pessoas em todo o mundo. Nesse contexto, a engenharia química contribui em diversas áreas e seus campos de conhecimento abrangem múltiplas ciências e saberes.

Dentre estes, a termodinâmica, cujo esboço surgiu ao longo do século XVIII a partir da Física e de fato foi criada em 1824 por Sardi Carnot, trouxe ao mundo a Máquina de Carnot que foi imprescindível no período da Primeira Revolução Industrial. Em 1850, Lorde Kelvin e Clausius, influenciados pelo trabalho de Joule, criaram a Primeira e Segunda Leis da Termodinâmica que mudariam a compreensão do calor como forma de energia e ampliou o campo de atuação da termodinâmica (THOMSON, 1851).

No século XX, Max Planck e Walther Nernst em trabalhos distintos formularam a Terceira Lei da Termodinâmica afirmando que a entropia de um sistema próxima ao zero absoluto é uma constante, também definido como: "É impossível através de qualquer procedimento, não importa o quão idealizado, reduzir a temperatura de qualquer sistema à temperatura zero em um finito número de finitas operações" (GUGGENHEIM, 1967).

O interesse pela definição de um sistema termodinâmico gerou a busca de diversos pesquisadores por equações que representassem o comportamento de gases e fluidos e em 1873, Van der Waals introduziu a primeira equação de estado derivada pela pressuposição de um volume finito ocupado pelas moléculas constituintes. Esta equação serviu como base para os modelos seguintes, tendo partes dela inalteradas mesmo em versões mais modernas como a proposta pelo trabalho de Redlich-Kwong(1949) e sua posterior mudança feita por Soave(1972).

No presente trabalho foi apresentada uma aplicação feita na linguagem de programação Python que visa ajudar alunos da disciplina de Termodinâmica Química com os resultados obtidos por meio de soluções analíticas, comparando-os com valores obtidos da aplicação. Para isso, utilizou-se módulos de bibliotecas Python tais como Thermo e para a criação de uma interface gráfica, utilizou-se a biblioteca Tkinter, ambas de código aberto e amplamente utilizadas em suas respectivas áreas.

2 FUNDAMENTAÇÃO TEÓRICA

Uma equação de estado é uma relação matemática entre as grandezas termodinâmicas de estado. (PERROT, 1998). Há propriedades termodinâmicas intensivas que são mensuráveis tais como pressão, volume molar, temperatura e concentração, e estas são independentes se analisadas aos pares, definindo um plano no qual podemos definir o valor da terceira propriedade. Matematicamente, expressa-se essa relação como:

$$f(P, v, T) = 0 \quad (2.1)$$

O objetivo de uma equação de estado é ajustar dados experimentais e prever o estado e propriedades termodinâmicas. Entretanto, diferentemente de uma equação fundamental que explicita todas as variáveis desejadas de um sistema, as equações de estado são constitutivas e não obrigatoriamente definem todos esses valores, além de muitas vezes não possuírem justificativa física para sua forma.

Por serem equações dita cúbicas, elas apresentaram três raízes para o volume. Dependendo de onde estas se encontrarem em relação ao ponto crítico o comportamento é alterado. Acima do ponto crítico há uma raiz real e positiva e as outras duas são negativas, não apresentando sentido físico. Abaixo do ponto crítico pode ocorrer o caso de três raízes positivas, porém utilizam-se apenas a de menor valor, representando o volume molar do estado líquido, e o maior valor encontrado, representando o volume molar do estado gasoso. A seguir cinco equações de estado utilizadas no programa Thermo PySoup.

2.1 Equação do Gás Ideal

A Equação dos Gases Ideais é obtida por meio da teoria cinética dos gases onde se aplicam simplificações de forma a considerar que o gás é composto por partículas perfeitamente esféricas, rígidas e infinitesimalmente pequenas e que só interagem por meio de colisões perfeitamente elásticas. As forças intermoleculares são desprezadas, de forma que a energia interna do sistema seja independente da pressão e exclusivamente dependente da temperatura. Nota-se o comportamento de gás ideal para todos os gases para pressões tendendo a zero.

$$PV = nRT \quad (2.2)$$

$$P = \frac{RT}{v} \quad (2.3)$$

2.2 Equação de Van der Waals

Ao contrário do modelo anterior, a Equação de Van der Waals contabiliza as interações intermoleculares por meio da função potencial de Sutherland. Para o modelo de gás ideal e simplificação de esferas rígidas tinha-se o seguinte comportamento para um par de moléculas:

$$\Gamma = \begin{cases} 0 & \text{para } r > \sigma \\ \sigma & \text{para } r < \sigma \end{cases} \quad (2.4)$$

É adicionado um termo atrativo para as interações, com uma descontinuidade quando r igual a σ , de forma a descrevê-las agora como:

$$\Gamma = \begin{cases} \frac{-C_6}{r_6} & \text{para } r > \sigma \\ \sigma & \text{para } r < \sigma \end{cases} \quad (2.5)$$

A descontinuidade evidencia que os centros das moléculas sempre distam mais que o valor de σ e para esse espaço vazio gerado há uma correção para o volume molar não-ocupado, de forma a termos:

$$P = \frac{RT}{v - b} \quad (2.6)$$

No caso das forças intermoleculares atrativas a variável mais significativa independentemente do tipo destas (dipolo-dipolo, indução, dispersão) é a pressão e as forças são tratadas de forma geral como forças de van der Waals. Como elas diminuem a pressão, dificultando as colisões entre as moléculas, o sinal de acordo com a referência adotada é negativo. O termo “b” para forças repulsivas e o posterior termo “a” para forças atrativas são constantes empíricas que modificam a equação proposta dos gases ideais de forma a ter o seguinte formato:

$$P = \frac{RT}{v - b} - \frac{a}{v^2} \quad (2.7)$$

A equação de Van der Waals necessita da determinação dos parâmetros a e b e estes estão relacionados com o princípio dos estados correspondentes. Dessa forma, as propriedades são avaliadas como se estivessem no ponto crítico e deve-se notar o ponto de inflexão da isoterma crítica, que matematicamente é expressa por:

$$\left(\frac{\partial P}{\partial v}\right)_{T_c} = \left(\frac{\partial^2 P}{\partial v^2}\right)_{T_c} = 0 \quad (2.8)$$

Analisando para o ponto crítico, tem-se:

$$P_c = \frac{RT_c}{v_c - b} - \frac{a}{v_c^2} \quad (2.9)$$

Igualando com as derivadas do ponto de inflexão e realizando o tratamento matemático adequado, encontram-se as seguintes expressões para determinação dos parâmetros a e b:

$$a = \frac{27 (RT_c)^2}{64 P_c} \quad (2.10)$$

$$b = \frac{RT_c}{8P_c} \quad (2.11)$$

2.3 Equação de Redlich-Kwong

Após a Equação de Van der Waals vários novos modelos foram surgindo de forma a aumentar a precisão dos modelos anteriores. O termo de repulsão da equação de Van der Waals se mantém intacta em várias destas, entretanto o termo de atração foi sendo refinado a cada modelo novo proposto. Uma forma de mensurar a qualidade de uma equação de estado cúbico está no comportamento do modelo em regiões próximas ao ponto crítico e pela quantidade de parâmetros ajustáveis nessa região. A Equação de Redlich-Kwong (1949) é escrita na forma:

$$P = \frac{RT}{v - b} - \frac{a}{T^{1/2}v(v + b)} \quad (2.12)$$

Analogamente ao que foi discutido para o modelo anterior, neste encontraremos as seguintes expressões para os parâmetros a e b, sendo estes não permutáveis entre as equações.

$$a = \frac{0,42748R^2T_c^{2,5}}{P_c} \quad (2.13)$$

$$b = \frac{0,08664R^2T_c}{P_c} \quad (2.14)$$

A Equação de Redlich-Kwong se destacou na época que foi desenvolvida e foi um avanço considerável frente Van der Waals. Entretanto, ela tem uma imprecisão para

equilíbrios líquido-vapor e para isso necessita de conjuntos de correlações para validar os resultados nessa situação. O intervalo de utilização dessa equação gera resultados mais satisfatórios quando.

$$\frac{p}{p_c} < \frac{T}{2T_c} \quad (2.15)$$

2.4 Equação de Soave Redlich-Kwong

A Equação de Soave-Redlich-Kwong (1972) modificou termos da equação passada e agora utiliza a esfericidade da molécula, conhecido como fator acêntrico, e envolve também a temperatura no seu cálculo. Possui leve diferença no denominador se comparada a Redlich-Kwong por não ter mais o termo da temperatura explícito no equacionamento e sim no novo parâmetro, α .

$$P = \frac{RT}{v - b} - \frac{\alpha a(T)}{v(v + b)} \quad (2.16)$$

$$a = \frac{0,42748R^2T_c^{2,5}}{P_c} \quad (2.17)$$

$$b = \frac{0,08664R^2T_c}{P_c} \quad (2.18)$$

$$\alpha = \left(1 + (0,48508 + 1,5571\omega - 0,17613\omega^2)(1 - T_r^{0,5})\right)^2 \quad (2.19)$$

2.5 Equação de Peng-Robinson

A Equação de Peng-Robinson (1976) assim como a Equação de Soave Redlich-Kwong também leva em conta o fator acêntrico das moléculas. Sua vantagem frente o modelo anterior é de possuir maior precisão na região crítica principalmente para líquidos e compostos apolares. O parâmetro de interação binária é independente das propriedades termodinâmicas intensivas e deve ser único. Na maior parte dos casos este possui resultados semelhantes ao modelo de Soave mas possui diferenças nos parâmetros internos como visto a seguir:

$$P = \frac{RT}{v-b} - \frac{\alpha a(T)}{(v-b)^2} \quad (2.20)$$

$$a = \frac{0,45724R^2T_c^2}{P_c} \quad (2.21)$$

$$b = \frac{0,07780RT_c}{P_c} \quad (2.22)$$

$$\alpha = \left(1 + (0,3764 + 1,54226\omega - 0,26992\omega^2)(1 - T_r^{0,5})\right)^2 \quad (2.24)$$

E escrevendo na forma polinomial, com Z sendo o fator de compressibilidade, temos a expressão:

$$\begin{aligned} Z^3 - (1-B)Z^2 + (A-2B-3B^2)Z \\ - (AB - B^2 - B^3) = 0 \end{aligned} \quad (2.25)$$

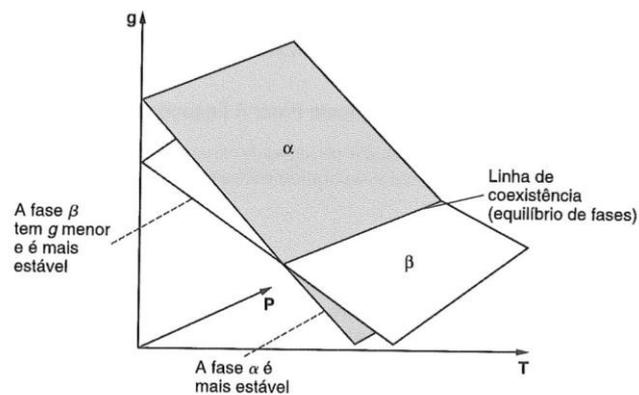
2.6 Equilíbrio de Fases

As misturas trabalhadas na aplicação Python podem estar em fases diferentes e dessa maneira é de interesse saber a distribuição da espécie química. A análise do equilíbrio químico parte de não haver gradientes mecânico ou térmico e se baseia em determinar o mínimo da energia livre de Gibbs (g) do sistema:

$$g_i^\alpha = g_i^\beta \quad (2.26)$$

α e β representando as fases vapor, líquida ou sólida. A partir de um gráfico de Temperatura pela Pressão, obteve-se as superfícies de α e β e é possível analisar o equilíbrio de fases a partir da linha de interseção das superfícies como é visto na Figura 1:

Figura 1- Superfície das fases



Fonte: Koretsky (2007)

A análise do sistema é realizada a partir de alterações infinitesimais das propriedades de forma a ser válida em toda a curva de coexistência do equilíbrio de fases. Dessa forma a equação 2.26 passa a forma diferencial:

$$dg_i^\alpha = dg_i^\beta \quad (2.27)$$

E com auxílio da definição da energia livre de Gibbs:

$$dg = dh - d(Ts) \quad (2.28)$$

$$dg = -sdT + vdP \quad (2.29)$$

Tem-se:

$$v_i^\alpha dP - S_i^\alpha dT = v_i^\beta dP - S_i^\beta dT \quad (2.30)$$

Rearranjando:

$$\frac{dP}{dT} = \frac{s_i^\alpha - s_i^\beta}{v_i^\alpha - v_i^\beta} \quad (2.31)$$

Da diferença de entropia tem-se:

$$s_i^\alpha - s_i^\beta = \frac{h_i^\alpha - h_i^\beta}{T} \quad (2.32)$$

E combinando com a equação 2.32 obtém-se a equação de Clayperon:

$$\frac{dP}{dT} = \frac{h_i^\alpha - h_i^\beta}{(v_i^\alpha - v_i^\beta)T} \quad (2.33)$$

2.7 Cálculo para Misturas

Para calcular as propriedades parciais de uma mistura faz-se uso da equação de Gibbs-Duhem. Quando o valor das propriedades de um componente é conhecido a utilização desta equação resulta nos valores do outro componente. A formulação parte da definição de propriedade extensiva total de um sistema multicomponente:

$$K = \sum n_i \bar{K}_i \quad (2.34)$$

Diferenciando a Equação 2.34 a T e P constantes:

$$dK_{PT} = \sum [n_i \overline{dK}_i + \overline{K}_i dn_i] \quad (2.35)$$

A temperatura e pressão constantes a equação se reduz a:

$$dK_{PT} = \sum \overline{K}_i dn_i \quad (2.36)$$

Para satisfazer ambas as equações:

$$0 = \sum n_i \overline{dK}_i \quad (2.37)$$

Determinando assim a Equação de Gibbs-Duhem.

2.8 Fugacidade

fugacidade, do latim ‘tendência a escapar’, propriedade termodinâmica, é considerada uma função de estado e foi desenvolvida por G.N Lewis de forma a generalizar a equação diferencial do potencial químico. Ela é definida com unidade de pressão e matematicamente é expressa por:A

$$\mu_i - \mu_i^0 \equiv RT \ln \left[\frac{\hat{f}_i}{f_i^0} \right] \quad (2.38)$$

A equação 2.38 atua como uma pressão corrigida e é válida para uma isotérmica do potencial químico, estejam as substâncias sólidas, líquidas ou gasosas. Quando a pressão se aproxima de zero, isto é, condição onde os gases se comportam como gases ideais, tem-se uma equação auxiliar a equação 2.38 que complementa a definição da propriedade, a seguir:

$$\lim_{P \rightarrow 0} \left(\frac{\hat{f}_i}{p_i} \right) \equiv 1 \quad (2.39)$$

O termo do denominador é denominado coeficiente de fugacidade e é normalmente representado pela letra φ . É uma grandeza adimensional que avalia como o sistema se encontra em comparação a idealidade em relação a pressão parcial do sistema. Se $\varphi < 1$ as forças atrativas dominam o comportamento do sistema, bem como, para $\varphi > 1$ as forças atrativas são dominantes.

Há outras equações que definem fugacidade e para a fugacidade total do sistema podemos escrever:

$$g - g^0 \equiv RT \ln \left[\frac{\hat{f}_i}{f_i^0} \right] \quad (2.40)$$

$$\lim_{P \rightarrow 0} \left(\frac{f}{P} \right) \equiv 1 \quad (2.41)$$

$$\varphi \equiv \frac{f}{P_{sis}} \quad (2.42)$$

Bem como para uma substância pura:

$$g - g^0 \equiv RT \ln \left[\frac{f_i}{f_i^0} \right] \quad (2.43)$$

$$\lim_{P \rightarrow 0} \left(\frac{f_i}{P} \right) \equiv 1 \quad (2.44)$$

$$\varphi_i \equiv \frac{f_i}{P_{sis}} \quad (2.45)$$

3 METODOLOGIA

A escolha pela linguagem de programação Python, lançada em 1991, justifica-se por ser uma linguagem de alto nível, orientada a objetos e possuir um desenvolvimento de código aberto. Ademais, por ter um foco em legibilidade facilita sua implementação, correções e eventuais evoluções no algoritmo. Pelo caráter pedagógico da aplicação, utilizar Python é um fator importante haja visto a sua facilidade no *download* nos três sistemas operacionais mais utilizados no mundo: Microsoft Windows, Linux e Mac.

Na implementação do código optou-se pela utilização de bibliotecas Python pela facilidade de utilização de métodos internos especializados para atividade em questão. Além disso, construiu-se uma interface gráfica para melhor interação e visualização por parte do usuário, a partir da biblioteca Tkinter, bem como para o cálculo das Equações de Estado Cúbicas, no qual foi utilizada a biblioteca Thermo. Internamente na determinação dos valores das propriedades de compostos químicos, usou-se o banco de dados da instituição PubChem que será abordado nas próximas seções.

3.1 TKinter

Figura 2- Logo da biblioteca Tkinter



Fonte: <iotbeginners.com>. Acesso em: 15 nov. 2022.

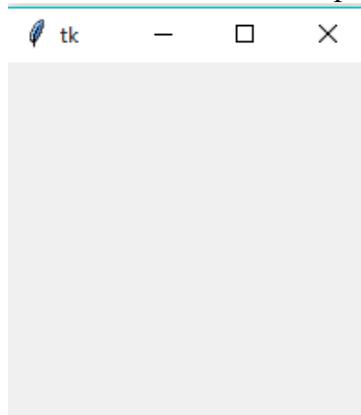
Bibliotecas como Tkinter são frameworks gráficos que criam interfaces para o usuário, conhecidas como GUI (Graphic User Interface). A principal vantagem da construção de GUI's é tornar o uso mais fácil além de aumentar a produtividade. O único pré-requisito necessário para sua utilização é ter instalado o pacote Python na máquina em questão. Por ser nativo do Python, sua utilização consiste apenas em importar o módulo e seus recursos se tornam disponíveis.

Alguns outros frameworks disponíveis no mercado como: Kivy, PyGTK, PySide e QT são boas alternativas, entretanto, Tkinter é o mais amplamente utilizado, principalmente para aplicações do nível do presente trabalho.

As GUI têm conceitos em comum, no qual pode-se citar: a estrutura no qual o programa estará contido. Os objetos armazenados, chamados de widgets. As rotinas, que tratam eventos disparados por ações dos usuários em widgets específicos (botões, entradas de texto, etc.). Eventos associados, no caso de ações que são executadas a partir de eventos prévios, na aplicação Thermo PySoup pode-se destacar o Top Level gerado a partir das informações adicionadas pelos usuários que permite o cálculo da propriedade Volume e demais informações como Pressão de Saturação, gráficos, etc.

Ao iniciar o projeto, tem-se a seguinte interface inicial, na qual o programador adiciona objetos e demais práticas descritas acima.

Figura 3- Interface default da aplicação



Fonte: elaborado pelo autor (2022)

E como desvantagens da biblioteca, pode-se citar que faltam alguns componentes importantes, como notebooks e combo box, que não são definidos nativamente, ou seja, o programador precisa combinar elementos para criá-los, bem

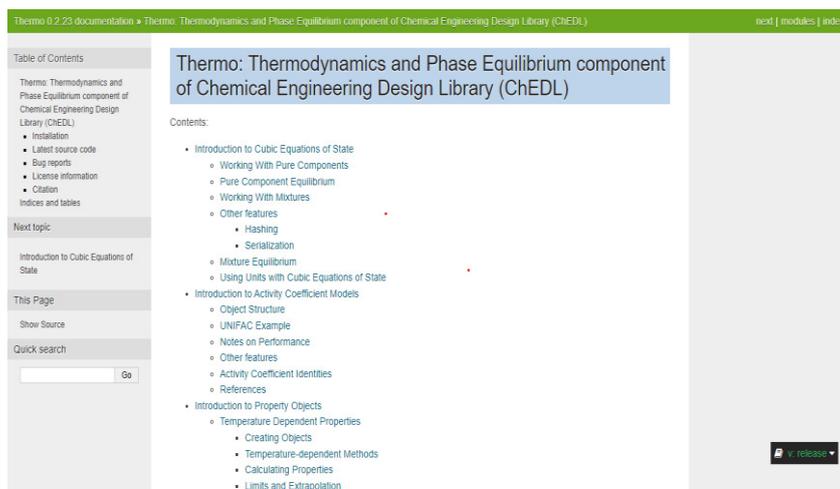
como, alguns elementos limitados. A biblioteca Tkinter usa uma função própria para acessar diretamente as funções do sistema operacional e desenhar seus próprios elementos na tela. Como vantagem gera portabilidade, mas gera a vantagem de destoar da aparência nativa do ambiente. Por último, a performance é baixa dado o que a biblioteca disponibiliza, ou seja, pode ser otimizada de forma a diminuir o tempo de processamento, principalmente a primeira sessão de interpretação feita pelo ambiente Python na qual é desenvolvida a aplicação.

3.2 Thermo

A biblioteca Thermo se enquadra na divisão de bibliotecas implementadas em código aberto para engenheiros, cientistas e demais técnicos trabalharem especificamente em determinada área das ciências e engenharias. Desenvolvida e mantida pelo engenheiro químico Caleb Bell da University of New Brunswick, ela facilita a busca por valores de propriedades de compostos químicos, calcula propriedades termodinâmicas diversas (temperatura, pressão, volume, fugacidade, entalpia e derivadas relacionadas a Termodinâmica) para substâncias puras e também para misturas, aplicando diversos métodos internos que podem ser encontrados na documentação.

Sua utilização depende de importação da biblioteca que pode ser feita facilmente a partir do ambiente de programação utilizado e sua importação é feita de maneira análoga ao descrito na seção dedicado ao Tkinter. A biblioteca é poderosa e intuitiva, segue a utilização do Sistema Internacional de Unidades e disponibiliza exemplos práticos na demonstração das Equações de Estado Cúbicas e/ou demais instanciações. Sua documentação encontra-se totalmente em inglês e a página web referente ao projeto consta a seguir na Figura 4

Figura 4- Home da biblioteca Thermo



Fonte: <<https://thermo.readthedocs.io/>> Acesso em: 15 nov. 2022.

3.2 PubChem

Conforme a utilização dos métodos da biblioteca Thermo surgiu a necessidade de validação dos valores usados de referência no cálculo do volume molar a partir de Equações de Estado Cúbicas. Dessa forma, internamente a biblioteca utiliza o banco de dados de moléculas PubChem. Este é operado e mantido pelo National Center for Biotechnology Information (NCBI), que faz parte da National Library of Medicine, integrante da National Institutes of Health dos Estados Unidos da América. A coleção de dados aberta a comunidade informa que o banco de dados dispõe de mais de 110000 compostos catalogados, desde estruturas das mais simples até moléculas robustas próximas a 2000 unidades de massa atômica. É uma fonte de pesquisa de patentes, especialmente para a área da biotecnologia. Para a termodinâmica também é uma ferramenta poderosa, podendo com o nome do composto em inglês ou sua nomenclatura/ID da União Internacional de Química Pura e Aplicada (IUPAC) encontrar valores de propriedades, artigos, patentes, aplicações, etc.

Figura 5- Home do banco de dados PubChem

Fonte: <<https://pubchem.ncbi.nlm.nih.gov/>>. Acesso em: 15 nov. 2022.

Figura 6- Buscador do banco de dados PubChem

Fonte: <<https://pubchem.ncbi.nlm.nih.gov/#query=water>>. Acesso em: 15 nov. 2022.

3.3 Thermo PySoup

Figura 7- Logo da Aplicação

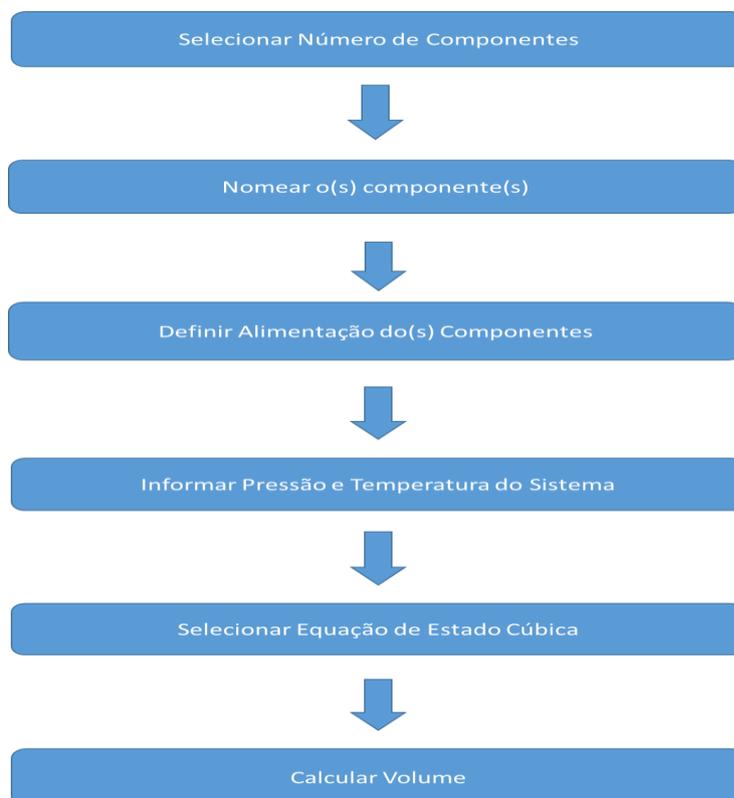


Fonte: elaborado pelo autor (2022)

Contando com as bibliotecas apresentadas nas seções anteriores, deu-se a construção da aplicação Thermo PySoup. O objetivo principal da aplicação é calcular volumes para o caso de substâncias puras ou misturas de dois componentes, explicitar as quantidades nos estados físicos a quais se encontram, determinar qual estado físico é mais estável e por fim, gerar um gráfico com um envelope de saturação para a situação implementada pelo usuário a partir da Equação de Estado Cúbica escolhida.

O projeto conta com uma sequência de passos que se inicia com a determinação de quantas substâncias estão envolvidas. Em seguida, explicitar o nome das substâncias e suas proporções molares alimentações. Tem-se valores de entrada para as propriedades Temperatura e Pressão e por fim um menu composto pelas equações disponíveis, para em seguida o cálculo do volume ser executado pelo computador.

Figura 8- Fluxograma da Aplicação



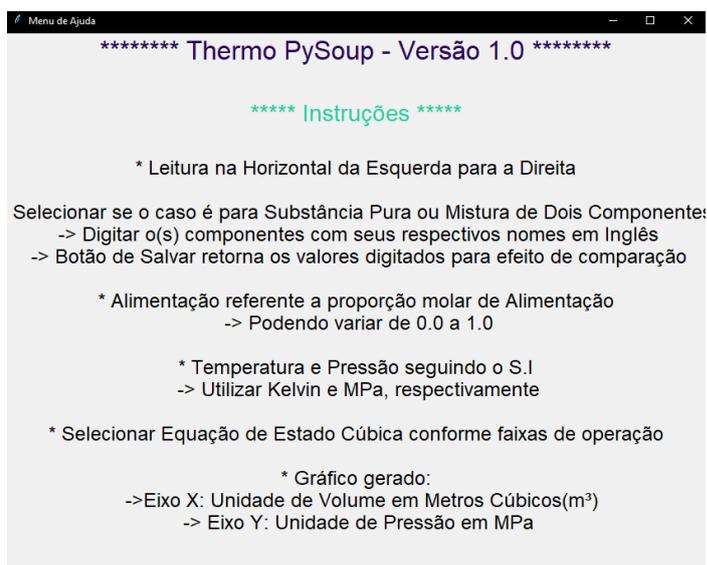
Fonte: elaborado pelo autor (2022)

Há no total cinco Equações de Estado Cúbicas, variando desde Gás Ideal até Equações mais robustas como Soave Redlich-Kwong para dois componentes.

A aplicação dispõe de um Menu guia de Ajuda e outro com informações do desenvolvedor e da aplicação, reforçando o cunho acadêmico da mesma.

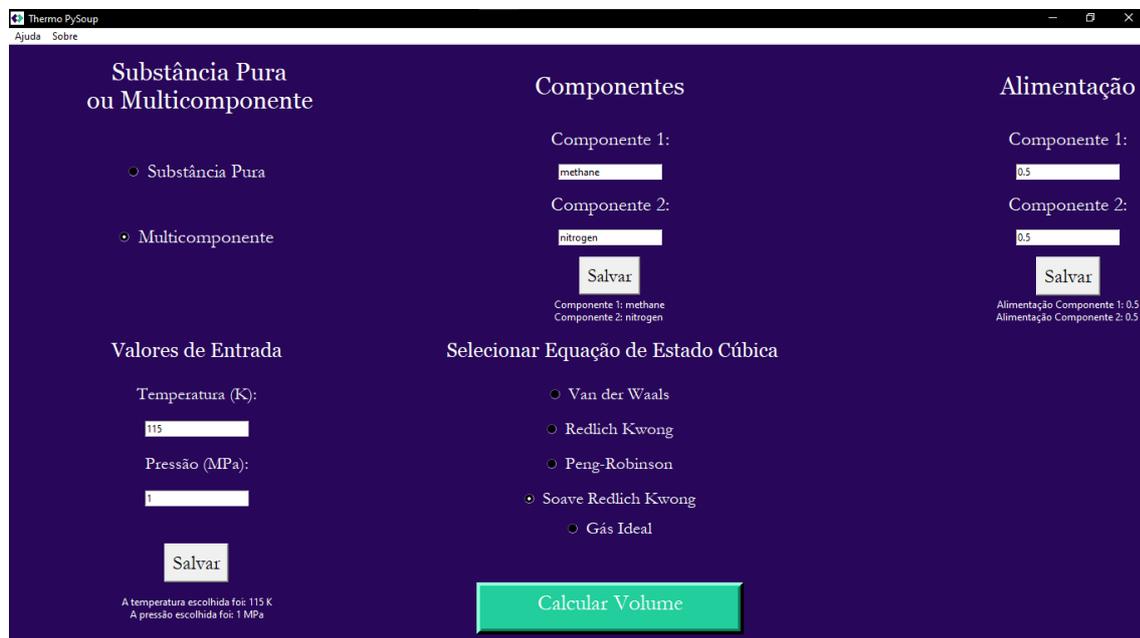
O código fonte está no Anexo A e um guia de ajuda, além da interface encontram-se a seguir:

Figura 9- Menu de Ajuda



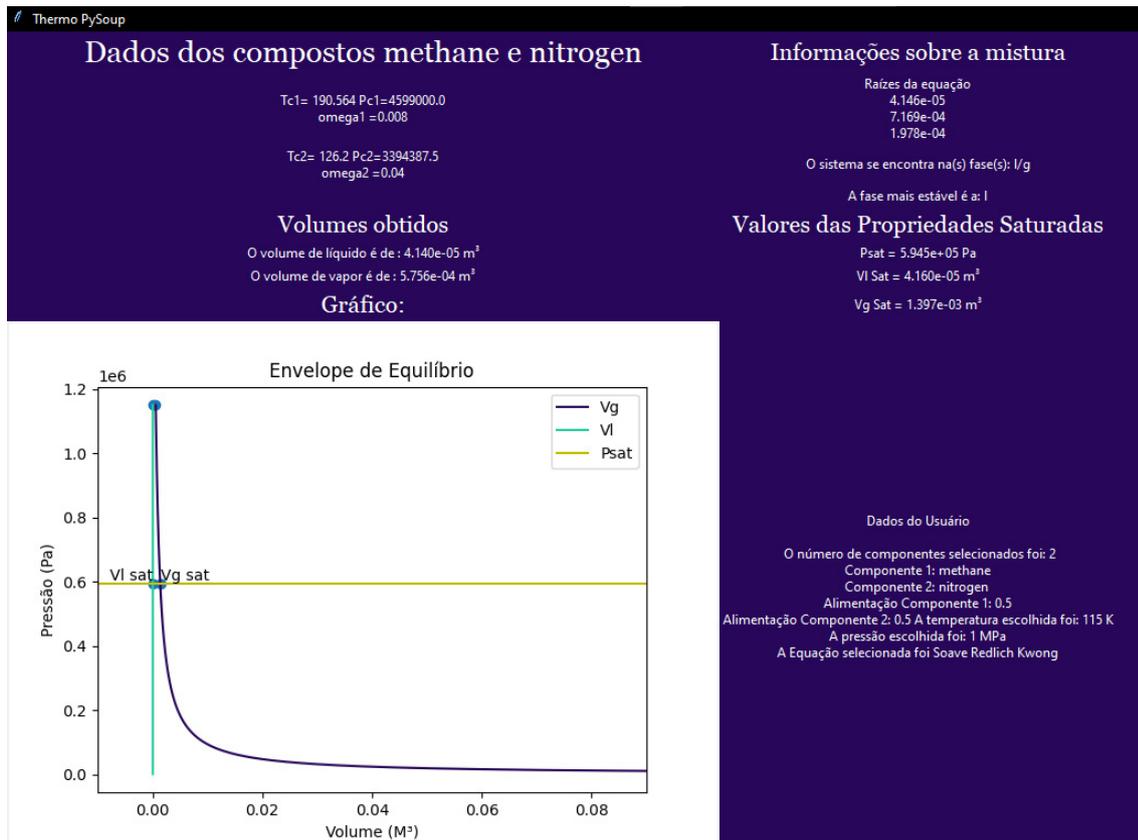
Fonte: elaborado pelo autor (2022)

Figura 10- Tela Inicial



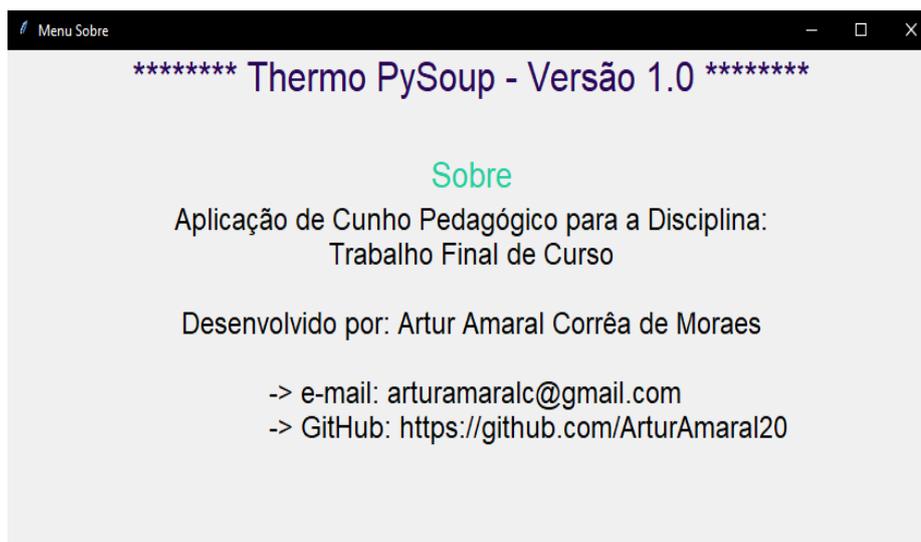
Fonte: elaborado pelo autor (2022)

Figura 11- Tela dos Resultados



Fonte: elaborado pelo autor (2022)

Figura 12- Menu Sobre



Fonte: elaborado pelo autor (2022)

4 RESULTADOS

Finalizado a implementação do código e desenvolvimento da interface gráfica, testou-se o programa para validar a eficácia do algoritmo em determinar o volume molar das substâncias. Por apresentar caráter pedagógico, foi utilizado o livro-texto da disciplina de Termodinâmica Química da Universidade Federal do Ceará da autoria de Milo D. Koretsky, comparando resultados analíticos com os obtidos pelo Thermo PySoup.

Os problemas foram apresentados, seguidos da solução analítica e tela da aplicação. Os envelopes de equilíbrio são apresentados quando a substância se encontrar abaixo do ponto crítico e soluções analíticas mais complexas foram resolvidas por meio do Solver do Excel.

4.1 Companhia de Gás Castor - Nitrogênio

Figura 13- Enunciado do Problema

- 4.27 Bem-vindo à Cia. de Gás Castor! Sua primeira tarefa é fazer um cálculo aproximado da venda anual dos nossos gases, grau superpuro, nitrogênio e oxigênio.
- (a) A quantidade aproximada de vendas de N_2 é de 30.000 unidades por ano. Considere que o volume do cilindro é de 43 L, a pressão é 12.400 kPa e o preço é R\$20,00/kg. Compare seu resultado àquele que você obteria através do modelo do gás ideal.
- (b) Repita para 30.000 unidades de O_2 a 15.000 kPa e R\$30,00/kg.

Fonte: Koretsky (2007)

Inicialmente, calculam-se os parâmetros a e b:

$$a = \frac{0,42748R^2T_c^{2,5}}{P_c} \quad (4.1)$$

$$a = 1,56[J.m^3.K^{1,5}.mol^{-2}] \quad (4.2)$$

$$b = \frac{0,08664R^2T_c}{P_c} \quad (4.3)$$

$$b = 2,65 \cdot 10^{-5} [m^3 \cdot mol^{-1}] \quad (4.4)$$

Utilizando a Equação de Redlich-Kwong

$$P = \frac{RT}{v-b} - \frac{a}{T^{1/2}v(v+b)} \quad (4.5)$$

Assumindo a temperatura ambiente em 22°C e pressão de 12400 kPa, substituindo os valores e utilizando o método Solver do Excel, tem-se:

Figura 14- Planilha volume molar do nitrogênio

	A	B	C	D	E	F	G	H	I	J	K	L
1	P=	12400000	Pa	Pressão Calculada		Diferença						
2	R=	8,314	J/m.K	12400000	Pa	7,82311E-08						
3	T=	299,15	K									
4	a=	1,56	J.m ³ .(K ^{1,5}).{mol ⁻² }									
5	b=	0,0000265	m ³ /mol									
6	v=	0,0001991	m ³ /mol									
7												

$$P = \frac{RT}{v-b} - \frac{a}{T^{1/2}v(v+b)}$$

Teste da Fórmula com o valor de v do Solver

14407504	-	2007504
P=	12400000	Pa

Fonte: elaborado pelo autor (2022)

$$v = 1,91 \cdot 10^{-4} [m^3 \cdot mol^{-1}] \quad (4.6)$$

Com auxílio da aplicação Thermo PySoup, o valor encontrado para o volume molar foi:

$$v = 1,99 \cdot 10^{-4} [m^3 \cdot mol^{-1}] \quad (4.7)$$

Figura 15- Tela inicial com os dados de entrada do problema

Fonte: elaborado pelo autor (2022)

Figura 16- Tela de resultados com o volume molar de nitrogênio

Dados dos composto nitrogen Informações sobre o composto	
Temperatura Crítica Composto 1= 126.2	Raízes da equação
Pressão Crítica Composto1= 3394387.5	1.996e-04
Fator de Acentricidade1 =0.04	0.000e+00
	0.000e+00

Fonte: elaborado pelo autor (2022)

Sendo assim, a diferença do resultado das duas resoluções é de 4,02%, ou seja, dentro de uma margem aceitável, levando em conta que a equação de Redlich-Kwong é um pouco mais robusta e o Solver não conseguiu, para esse caso A, reduzir o parâmetro a zero para determinar precisamente o volume molar. Esse exemplo foi o que apresentou maior diferença entre a solução analítica (finalizada com o Solver) e a solução da aplicação. No final da seção foi abordado detalhadamente a porcentagem diferencial dos modelos comparativos de solução.

Continuando a solução do problema com o valor que a aplicação Thermo PySoup nos fornece tem-se o volume total de gás para 30000 unidades.

$$V_{total} = (30000 \text{ cilindros}) \cdot \left(43 \frac{L}{\text{cilindro}} \right) = 1290 \text{ m}^3$$

O número de mols do sistema é dado por:

$$n = \frac{V_{total}}{v} \quad (4.8)$$

$$n = \frac{1290 \text{ m}^3}{1,91 \cdot 10^{-4} [\text{m}^3 \cdot \text{mol}^{-1}]} = 6,48 \cdot 10^{-6} \text{ mol} \quad (4.9)$$

A massa de gás é dada por:

$$\begin{aligned} m &= (6,48 \cdot 10^{-6} \text{ mol}) \cdot (0,02801 \text{ kg} \cdot \text{mol}^{-1}) \\ &= 1,81 \cdot 10^5 \text{ Kg} \end{aligned} \quad (4.10)$$

E o valor anual de vendas, em real é dado por:

$$Valor \text{ (R\$)} = (1,81 \cdot 10^5 \text{ Kg})(\text{R\$ } 20,00 \cdot \text{Kg}^{-1}) \quad (4.11)$$

$$Valor = \text{R\$ } 3.629.623,31 \text{ reais} \quad (4.12)$$

Um total anual de venda de R\$ 3,63 milhões de reais.

Comparativamente, a utilização da Equação dos Gases Ideais resultaria em um valor próximo.

$$v = \frac{RT}{P} \quad (4.13)$$

$$v = \frac{(8,314 [\text{J} \cdot \text{m}^{-1} \cdot \text{K}^{-1}]) \cdot (295,15 \text{ K})}{12400000 \text{ Pa}} \quad (4.14)$$

$$v = 1,98 \cdot 10^{-4} [\text{m}^3 \cdot \text{mol}^{-1}] \quad (4.15)$$

Volume molar semelhante ao determinado pela equação de Redlich-Kwong, o que confirma que as condições de pressão baixa, temperatura e um componente gás puro se comportaria próximo a um gás ideal

4.2 Companhia de Gás Castor - Oxigênio

Inicialmente, calculam-se os parâmetros a e b:

$$a = \frac{0,42748R^2T_c^{2,5}}{P_c} \quad (4.16)$$

$$a = 1,74 [J \cdot m^3 \cdot K^{1,5} \cdot mol^{-2}] \quad (4.17)$$

$$b = \frac{0,08664R^2T_c}{P_c} \quad (4.18)$$

$$b = 2,21 \cdot 10^{-5} [m^3 \cdot mol^{-1}] \quad (4.19)$$

Utilizando a Equação de Redlich-Kwong

$$P = \frac{RT}{v - b} - \frac{a}{T^{\frac{1}{2}}v(v + b)} \quad (4.20)$$

Assumindo a temperatura ambiente em 22°C e pressão de 15000 kPa, substituindo os valores e utilizando o método Solver do Excel, tem-se:

Figura 17- Planilha volume molar do oxigênio

	A	B	C	D	E	F	G	H	I	J	K	L
1	P=	15000000	Pa	Pressão Calculada		Diferença		$P = \frac{RT}{v - b} - \frac{a}{T^{\frac{1}{2}}v(v + b)}$				
2	R=	8,314	J/m.K		15000011	Pa	10,61049					
3	T=	299,15	K									
4	a=	1,74	J.m ³ .(K ^{1,5}). (mol ⁻²)									
5	b=	0,0000221	m ³ /mol									
6	v=	0,0001555	m ³ /mol									
7								Teste da Fórmula com o valor de v do Solver				
								18642136	-	3642126		
									15000011			

Fonte: elaborado pelo autor (2022)

$$v = 1,55 \cdot 10^{-4} [m^3 \cdot mol^{-1}] \quad (4.21)$$

Figura 18- Tela de início com os dados de entrada do problema



Fonte: elaborado pelo autor (2022)

Figura 19- Tela de resultados com o volume molar de oxigênio

Dados dos composto oxygen Informações sobre o composto	
Temperatura Crítica Composto 1= 154.58	Raízes da equação
Pressão Crítica Composto1=5042945.25	1.555e-04
Fator de Acentricidade1 =0.021	0.000e+00
	0.000e+00

Fonte: elaborado pelo autor (2022)

Sendo assim, os valores encontrados são iguais utilizando três algarismos significativos após a vírgula e o erro associado entre as resoluções pode ser considerado zero.

Continuando a solução do problema com o valor que a aplicação Thermo PySoup nos fornece, tem-se o volume total de gás para 30000 unidades.

$$V_{total} = (30000 \text{ cilindros}) \cdot \left(43 \frac{L}{\text{cilindro}} \right) = 1290 m^3$$

O número de mols do sistema é dado por:

$$n = \frac{V_{total}}{v} \quad (4.22)$$

$$n = \frac{1290 \text{ m}^3}{1,55 \cdot 10^{-4} [\text{m}^3 \cdot \text{mol}^{-1}]} = 8,29 \cdot 10^{-6} \text{ mol} \quad (4.23)$$

A massa de gás é dada por:

$$m = (8,29 \cdot 10^{-6} \text{ mol}) \cdot (0,032 \text{ kg} \cdot \text{mol}^{-1}) = 2,65 \cdot 10^{-5} \text{ Kg} \quad (4.24)$$

E o valor anual de vendas, em real é dado por:

$$\text{Valor (R\$)} = (2,65 \cdot 10^5 \text{ Kg})(\text{R\$ } 30,00 \cdot \text{Kg}^{-1}) \quad (4.25)$$

$$\text{Valor} = \text{R\$ } 7.961.427,20 \text{ reais} \quad (4.26)$$

Um total anual de venda de aproximadamente R\$ 7,96 milhões de reais.

Comparativamente, a utilização da Equação dos Gases Ideais resultaria em um erro maior do que o apresentado no caso anterior

$$v = \frac{RT}{P} \quad (4.27)$$

$$\begin{aligned} v &= \frac{(8,314 [\text{J} \cdot \text{m}^{-1} \cdot \text{K}^{-1}]) \cdot (295,15 \text{ K})}{15000000 \text{ Pa}} \\ &= 1,64 \cdot 10^{-4} [\text{m}^3 \cdot \text{mol}^{-1}] \end{aligned} \quad (4.28)$$

Valor significativamente diferente do determinado pela equação de Redlich-Kwong. Com o aumento da pressão do sistema e um gás com maior peso molecular e eletronegativo, a diferença do modelo ideal é mais perceptível que no primeiro caso para o gás nitrogênio.

4.3 Determinação do Volume Molar de Etano – Item A

Figura 20- Enunciado do Problema

4.31 Calcule o seguinte:

(a) o volume ocupado por 20 kg de etano a 70°C e 30 bar

(b) a pressão necessária para encher um recipiente de 0,1 m³, à temperatura ambiente, de modo a armazenar 40 kg de etano

Fonte: Koretsky (2007)

Dada a Pressão e a Temperatura do Sistema, usa-se carta de compressibilidade para se determinar as propriedades reduzidas:

$$T_r = \frac{T}{T_c} \quad (4.29)$$

$$\frac{T}{T_c} = \frac{343,15 \text{ K}}{305,4 \text{ K}} = 1,12 \quad (4.30)$$

$$P_r = \frac{P}{P_c} \quad (4.31)$$

$$\frac{P}{P_c} = \frac{30 \text{ bar}}{48,74 \text{ bar}} = 0,616 \quad (4.32)$$

Dessa forma, temos:

$$\omega = 0,099 \quad (4.33)$$

Interpolando o gráfico:

$$z^{(0)} = 0.8376 \quad (4.34)$$

$$z^{(1)} = 0.0168 \quad (4.35)$$

Aplicando a fórmula:

$$z = z^{(0)} + \omega z^{(1)} \quad (4.36)$$

Substituindo os valores:

$$z = 0.8376 + (0,099)(0.0168) \quad (4.37)$$

$$z = 0.8393 \quad (4.38)$$

Calculando o volume pela equação a seguir:

$$V = \frac{z \cdot n \cdot R \cdot T}{P} \quad (4.39)$$

$$V = \frac{(0.8393) \cdot \left(\frac{30 \text{ kg}}{0.03007 \text{ Kg. mol}^{-1}} \right) \cdot (8,314 [\text{J. m}^{-1} \cdot \text{K}^{-1}]) 343.15 \text{ K}}{3.10^6 \text{ Pa}} \quad (4.40)$$

$$V = 0.796 \text{ m}^3 \quad (4.41)$$

$$v = \frac{0.796}{\frac{30 \text{ kg}}{0.03007 \text{ Kg. mol}^{-1}}} \quad (4.42)$$

$$v = 7,978. 10^{-4} \text{ m}^3 \cdot \text{mol}^{-1} \quad (4.43)$$

Para as mesmas condições, é testado o modelo na aplicação Thermo PySoup e encontra-se o valor de volume molar igual a:

$$v = 7,970. 10^{-4} \text{ m}^3 \cdot \text{mol}^{-1} \quad (4.44)$$

Sendo representado os valores de entrada na Figura 21 e o resultado do volume molar na Figura 22, como visto a seguir:

Figura 21- Tela inicial com os dados de entrada do problema

The screenshot shows the Thermo PySoup application interface. It is divided into three main sections: 'Substância Pura ou Multicomponente', 'Componentes', and 'Alimentação'. In the 'Substância Pura ou Multicomponente' section, 'Substância Pura' is selected. The 'Valores de Entrada' section shows 'Temperatura (K): 343.15' and 'Pressão (MPa): 3'. In the 'Componentes' section, 'Componente 1' is 'ethane' and 'Componente 2' is '-'. In the 'Alimentação' section, 'Componente 1' is '1.0' and 'Componente 2' is '-'. The 'Selecionar Equação de Estado Cúbica' section has radio buttons for 'Van der Waals', 'Redlich Kwong', 'Peng-Robinson', 'Soave Redlich Kwong', and 'Gás Ideal'. A 'Calcular Volume' button is at the bottom right. A status bar at the bottom left indicates 'A temperatura escolhida foi: 343.15 K' and 'A pressão escolhida foi: 3 MPa'.

Fonte: elaborado pelo autor (2022)

Figura 22- Tela de resultados com o volume de Etano

The screenshot shows the results page of the Thermo PySoup application. The title is 'Dados dos composto ethane Informações sobre o composto'. The results are displayed in two columns:

Dados dos composto ethane		Informações sobre o composto	
Temperatura Crítica Composto 1=	305.32	Raízes da equação	
Pressão Crítica Composto1=	4872000.0		7.970e-04
Fator de Acentricidade1 =	0.098		0.000e+00
			0.000e+00

Fonte: elaborado pelo autor (2022)

A diferença nos valores encontrados é de 0,1%, o que indica convergência entre a solução analítica e a solução computacional da aplicação.

4.4 Determinação do Volume Molar de Etano – Item B

Assumindo uma temperatura ambiente de 25°C e utilizando a Equação de Redlich-Kwong:

$$v = \frac{V_{total}}{n} \quad (4.45)$$

$$v = \frac{0.1 \text{ m}^3}{\frac{40 \text{ kg}}{0.03007 \text{ Kg. mol}^{-1}}} \quad (4.46)$$

$$v = 7,518 \cdot 10^{-5} [\text{m}^3 \cdot \text{mol}^{-1}] \quad (4.47)$$

$$a = \frac{0,42748R^2T_c^{2,5}}{P_c} = 9,88 [\text{J} \cdot \text{m}^3 \cdot \text{K}^{1,5} \cdot \text{mol}^{-2}] \quad (4.48)$$

$$b = \frac{0,08664R^2T_c}{P_c} = 4,51 \cdot 10^{-5} [\text{m}^3 \cdot \text{mol}^{-1}] \quad (4.49)$$

E a Pressão total do sistema é:

$$P = 191.3\text{bar} \quad (4.50)$$

Utilizando a aplicação com os mesmos valores e equação de estado cúbica do problema, tem-se na Figura 23 a página de entrada das variáveis e na Figura 24 o valor do volume molar, o qual foi encontrado valendo:

$$v = 7,529 \cdot 10^{-5} [m^3 \cdot mol^{-1}] \quad (4.51)$$

Figura 23- Tela inicial com os dados de entrada do sistema

The screenshot shows the Thermo PySoup application interface. It is divided into three main sections: 'Substância Pura ou Multicomponente', 'Componentes', and 'Alimentação'. The 'Substância Pura ou Multicomponente' section has radio buttons for 'Substância Pura' and 'Multicomponente'. The 'Componentes' section has input fields for 'Componente 1' (filled with 'ethane') and 'Componente 2' (empty), with a 'Salvar' button below. The 'Alimentação' section has input fields for 'Componente 1' (filled with '1.0') and 'Componente 2' (empty), with a 'Salvar' button below. The 'Valores de Entrada' section has input fields for 'Temperatura (K):' (filled with '298.15') and 'Pressão (MPa):' (filled with '19.13'), with a 'Salvar' button below. The 'Selecionar Equação de Estado Cúbica' section has radio buttons for 'Van der Waals', 'Redlich Kwong', 'Peng-Robinson', 'Soave Redlich Kwong', and 'Gás Ideal'. A large green 'Calcular Volume' button is at the bottom center. At the bottom left, it says 'A temperatura escolhida foi: 298.15 K' and 'A pressão escolhida foi: 19.13 MPa'.

Fonte: elaborado pelo autor (2022)

Figura 24- Tela de resultados com o volume de Etano

The screenshot shows the results page of the Thermo PySoup application. It is divided into two main sections: 'Dados dos composto ethane' and 'Informações sobre o composto'. The 'Dados dos composto ethane' section shows 'Temperatura Crítica Composto 1= 305.32', 'Pressão Crítica Composto 1=4872000.0', and 'Fator de Acentricidade1 =0.098'. The 'Informações sobre o composto' section shows 'Raízes da equação' with values '7.529e-05', '0.000e+00', and '0.000e+00'.

Dados dos composto ethane	Informações sobre o composto
Temperatura Crítica Composto 1= 305.32	Raízes da equação
Pressão Crítica Composto 1=4872000.0	7.529e-05
Fator de Acentricidade1 =0.098	0.000e+00
	0.000e+00

Fonte: elaborado pelo autor (2022)

A diferença nos valores encontrados é de 0,146 %, o que indica convergência entre a solução analítica e a solução computacional da aplicação.

4.5 Determinação do Volume Molar de Propano – Redlich-Kwong

Figura 25- Enunciado do Problema

4.32 Calcule o volume ocupado por 50 kg de propano a 35 bar e 50°C, usando o seguinte:

- (a) O modelo do gás ideal
- (b) A equação de estado de Redlich-Kwong
- (c) A equação de estado de Peng-Robinson

Fonte: Koretsky (2007)

A partir da Equação de Estado Cúbica de Redlich-Kwong, define-se os valores dos parâmetros a e b:

$$P = \frac{RT}{v - b} - \frac{a}{T^{1/2}v(v + b)} \quad (4.52)$$

$$a = \frac{0,42748R^2T_c^{2,5}}{P_c} = 18,33[J \cdot m^3 \cdot K^{1,5} \cdot mol^{-2}] \quad (4.53)$$

$$b = \frac{0,08664R^2T_c}{P_c} = 6,28 \cdot 10^{-5}[m^3 \cdot mol^{-1}] \quad (4.54)$$

Figura 26- Planilha com o volume molar de Propano

	A	B	C	D	E	F	G	H	I	J	K	L	
1	P=	3500000	Pa	Pressão Calculada		Diferença	$P = \frac{RT}{v - b} - \frac{a}{T^{1/2}v(v + b)}$						
2	R=	8,314	J/m.K		3503618	Pa							3618,011
3	T=	323,15	K										
4	a=	18,33	K ^{1,5} .(m										
5	b=	0,0000628	m ³ /mol					Teste da Fórmula com o valor de v do Solver					
6	v=	0,0001095	m ³ /mol					57585198	-	54081580			
7								P=	3503618	Pa			

Fonte: elaborado pelo autor (2022)

O volume molar de propano:

$$v = 0.0001095 [m^3 \cdot mol^{-1}] \quad (4.55)$$

Em seguida, utilizando a aplicação Thermo PySoup, foi definido um valor para o volume molar, bem como, estando abaixo do ponto crítico, um gráfico do equilíbrio para a situação proposta.

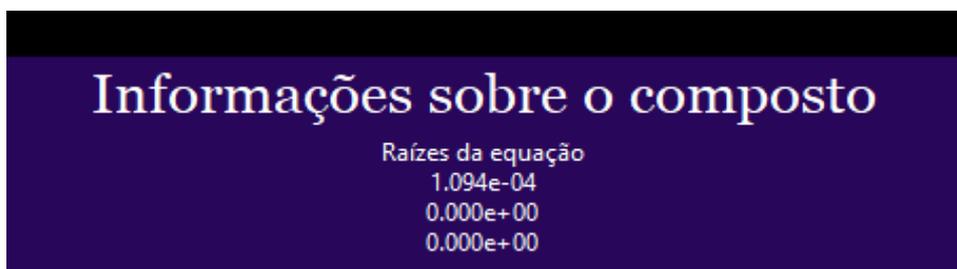
$$v = 0.0001094 [m^3 \cdot mol^{-1}] \quad (4.56)$$

Figura 27- Tela Inicial com os valores de entrada do problema

The screenshot shows the Thermo PySoup application interface. It is divided into three main sections: 'Substância Pura ou Multicomponente', 'Componentes', and 'Alimentação'. Under 'Substância Pura ou Multicomponente', 'Substância Pura' is selected. Under 'Componentes', 'Componente 1' is set to 'propane'. Under 'Alimentação', 'Componente 1' is set to '1.0'. In the 'Valores de Entrada' section, 'Temperatura (K)' is 323.15 and 'Pressão (MPa)' is 3.5. The 'Selecionar Equação de Estado Cúbica' section has 'Soave Redlich Kwong' selected. A green 'Calcular Volume' button is at the bottom. A 'Salvar' button is also present. At the bottom left, it says 'A temperatura escolhida foi: 323.15 K' and 'A pressão escolhida foi: 3.5 MPa'.

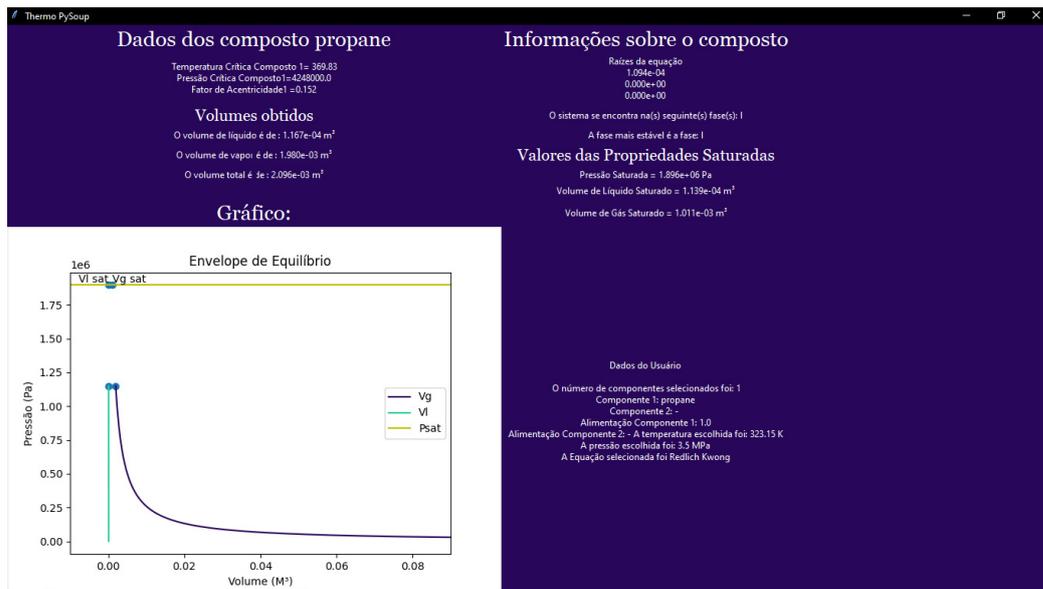
Fonte: elaborado pelo autor (2022)

Figura 28- Tela de resultados com o volume de Propano



Fonte: elaborado pelo autor (2022)

Figura 29- Tela de resultado e gráfico com o volume molar de Propano



Fonte: elaborado pelo autor (2022)

A diferença nos valores encontrados é de 0,365 %, o que indica convergência entre a solução analítica e a solução computacional da aplicação.

Determinado o volume total ocupado:

$$v = \frac{V_{total}}{n} \quad (4.57)$$

$$0.000109 [m^3 \cdot mol^{-1}] = \frac{V_{total}}{1130 mol} \quad (4.58)$$

$$V_{total} = 0.124 m^3 \quad (4.59)$$

4.6 Determinação do Volume Molar de Propano – Peng-Robinson

Determinando o Volume total pela Equação de Peng-Robinson:

$$P = \frac{RT}{v-b} - \frac{\alpha a(T)}{(v-b)^2} \quad (4.60)$$

$$a = \frac{0,45724R^2T_c^2}{P_c} \quad (4.61)$$

$$b = \frac{0,07780RT_c}{P_c} \quad (4.62)$$

Determinando a temperatura reduzida:

$$T_r = \frac{T}{T_c} \quad (4.63)$$

$$\frac{T}{T_c} = \frac{323,15 \text{ K}}{370 \text{ K}} = 0,873 \quad (4.64)$$

Determinando os parâmetros a e b:

$$a = \frac{0,45724R^2T_c^2}{P_c} = 1,02 \text{ J} \cdot \text{m}^3 \cdot \text{mol}^{-1} \quad (4.65)$$

$$b = \frac{0,07780RT_c}{P_c} = 5.64 \cdot 10^{-5} m^3 \cdot mol^{-1} \quad (4.66)$$

Substituindo em α :

$$\alpha = \left(1 + (0,3764 + 1,54226\omega - 0,26992\omega^2)(1 - T_r^{0,5})\right)^2 \quad (4.67)$$

$$\alpha = 1,081 \quad (4.68)$$

Definindo todas as variáveis, pode-se substituir na equação de Peng-Robinson:

$$P = \frac{RT}{v - b} - \frac{\alpha a(T)}{(v - b)^2} \quad (4.69)$$

Encontrando para o volume molar o seguinte valor:

$$v = 0.0000942 [m^3 \cdot mol^{-1}] \quad (4.70)$$

Em seguida, utilizando a aplicação Thermo PySoup, foi definido um valor para o volume molar, bem como, estando abaixo do ponto crítico, um gráfico do equilíbrio para a situação proposta.

$$v = 0.0000943 [m^3 \cdot mol^{-1}] \quad (4.71)$$

Figura 30- Tela de início com os dados de entrada do problema

The screenshot shows the Thermo PySoup software interface with the following sections:

- Substância Pura ou Multicomponente:** Radio buttons for "Substância Pura" (selected) and "Multicomponente".
- Componentes:** Input fields for "Componente 1:" (containing "propane") and "Componente 2:" (containing "-"). A "Salvar" button is below.
- Alimentação:** Input fields for "Componente 1:" (containing "1.0") and "Componente 2:" (containing "-"). A "Salvar" button is below.
- Valores de Entrada:** Input fields for "Temperatura (K):" (containing "323.15") and "Pressão (MPa):" (containing "3.5"). A "Salvar" button is below.
- Selecionar Equação de Estado Cúbica:** Radio buttons for "Van der Waals", "Redlich Kwong" (selected), "Peng-Robinson", "Soave Redlich Kwong", and "Gás Ideal".
- Footer:** "A temperatura escolhida foi: 323.15 K" and "A pressão escolhida foi: 3.5 MPa".
- Buttons:** A large green "Calcular Volume" button is at the bottom center.

Fonte: elaborado pelo autor (2022)

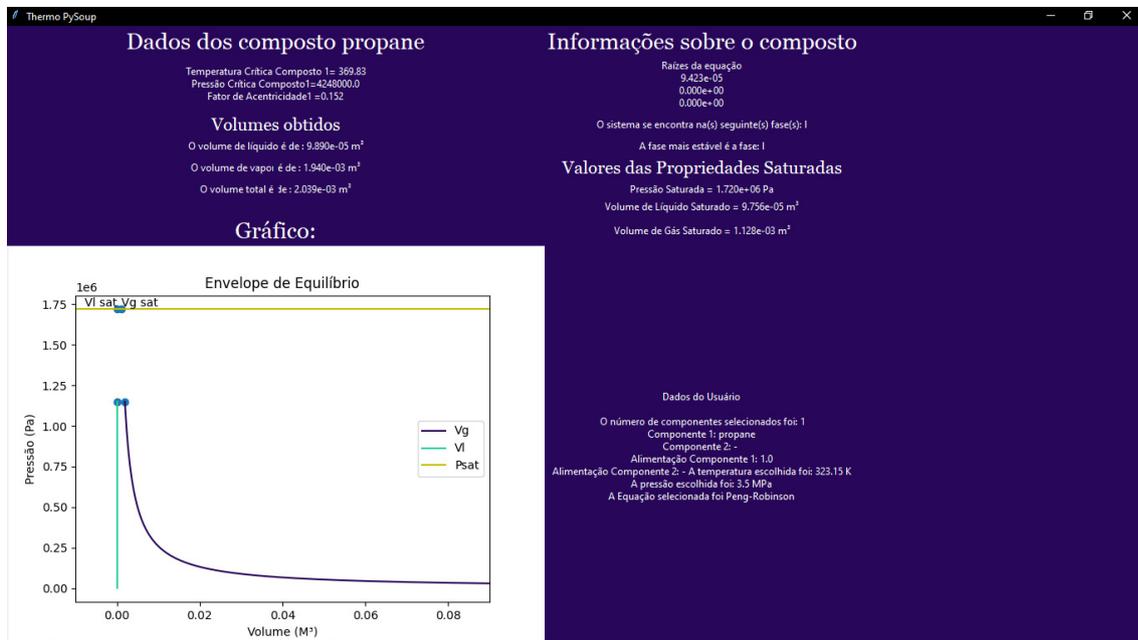
Figura 31- Tela de resultado com o volume molar de Propano

The screenshot shows the results of the calculation on a dark blue background with white text:

- Informações sobre o composto**
- Raízes da equação**
- 9.423e-05
- 0.000e+00
- 0.000e+00

Fonte: elaborado pelo autor (2022)

Figura 32- Tela de resultado e gráfico com o volume molar de Propano



Fonte: elaborado pelo autor (2022)

A diferença nos valores encontrados é de 0,106 %, o que indica convergência entre a solução analítica e a solução computacional da aplicação.

Continuando a questão com o proposto pelo exercício e determinado o volume total ocupado:

$$v = \frac{V_{total}}{n} \quad (4.72)$$

$$0.0000942 [m^3 \cdot mol^{-1}] = \frac{V_{total}}{1130 mol} \quad (4.73)$$

$$V_{total} = 0.107 m^3 \quad (4.74)$$

4.7 Determinação do Volume Molar da Amônia – Soave Redlich-Kwong e Peng-Robinson

Figura 33- Enunciado do Problema

4.35 Usando os gráficos/tabelas de compressibilidade generalizada, calcule o volume molar da amônia a 92°C e 306,5 bar. Qual é o estado físico da amônia?

Fonte: Koretsky (2007)

Da Solução Analítica utilizando gráficos/tabelas de compressibilidade, tem-se:

$$T_c = 405,6 \text{ K} \quad (4.75)$$

$$P_c = 112.77 \text{ bar} \quad (4.76)$$

$$\omega = 0,25 \quad (4.77)$$

Calculando a Temperatura e a Pressão reduzidas:

$$T_r = \frac{T}{T_c} \quad (4.78)$$

$$\frac{T}{T_c} = \frac{(92 + 273.15) \text{ K}}{405,6 \text{ K}} = 0,9 \quad (4.79)$$

$$P_r = \frac{P}{112.77 \text{ bar}} \quad (4.80)$$

$$\frac{P}{P_c} = \frac{306.5 \text{ bar}}{48,74 \text{ bar}} = 2.718 \quad (4.81)$$

Interpolando para determinar z :

$$z^{(0)} = 0.4133 \quad (4.82)$$

$$z^{(1)} = -0.1351 \quad (4.83)$$

Aplicando a fórmula:

$$z = z^{(0)} + \omega z^{(1)} \quad (4.84)$$

Substituindo os valores:

$$z = 0.4133 + (0,25)(-0.1351) \quad (4.85)$$

$$z = 0.38 \quad (4.86)$$

Calculando o volume molar pela equação a seguir:

$$v = \frac{z \cdot R \cdot T}{P} \quad (4.87)$$

$$v = \frac{(0,38) \cdot (8,314 [J \cdot m^{-1} \cdot K^{-1}]) (365,15) K}{30,65 MPa} \quad (4.88)$$

$$v = 3,76 \cdot 10^{-5} m^3/mol \quad (4.89)$$

Como no sistema a amônia se encontra abaixo da temperatura crítica e a pressão é maior do que a pressão crítica, a substância se encontra líquida no sistema. Isso corrobora com os resultados da aplicação a seguir.

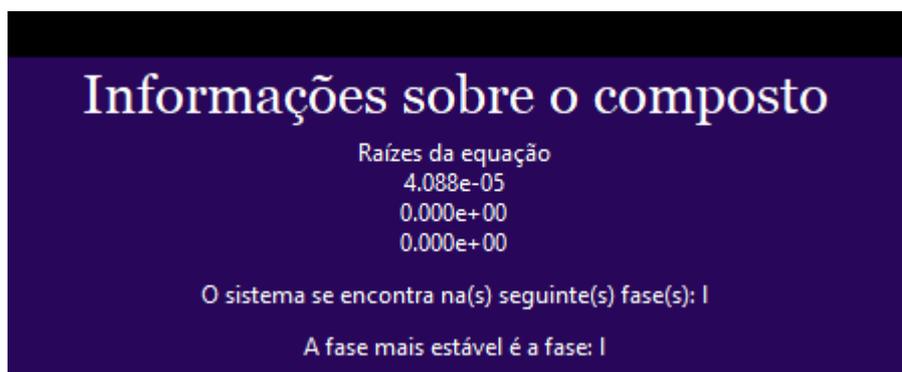
Figura 34- Tela de início com os valores de entrada do problema

The screenshot shows the 'ThermoPySeup' software interface with the following sections and values:

- Substância Pura ou Multicomponente:**
 - Substância Pura
 - Multicomponente
- Componentes:**
 - Componente 1: ammonia
 - Componente 2: (empty)
 - Buttons: Salvar
 - Summary: Componente 1: ammonia, Componente 2: -
- Alimentação:**
 - Componente 1: 14
 - Componente 2: (empty)
 - Buttons: Salvar
 - Summary: Alimentação Componente 1: 14, Alimentação Componente 2: -
- Valores de Entrada:**
 - Temperatura (K): 365,15
 - Pressão (MPa): 30,65
 - Buttons: Salvar
 - Summary: A temperatura escolhida foi: 365,15 K, A pressão escolhida foi: 30,65 MPa
- Selecionar Equação de Estado Cúbica:**
 - Van der Waals
 - Redlich Kwong
 - Peng-Robinson
 - Soave Redlich Kwong
 - Gás Ideal
- Buttons:** Salvar (bottom left), Calcular Volume (bottom center)

Fonte: elaborado pelo autor (2022)

Figura 35- Tela de resultado com o volume molar da Amônia



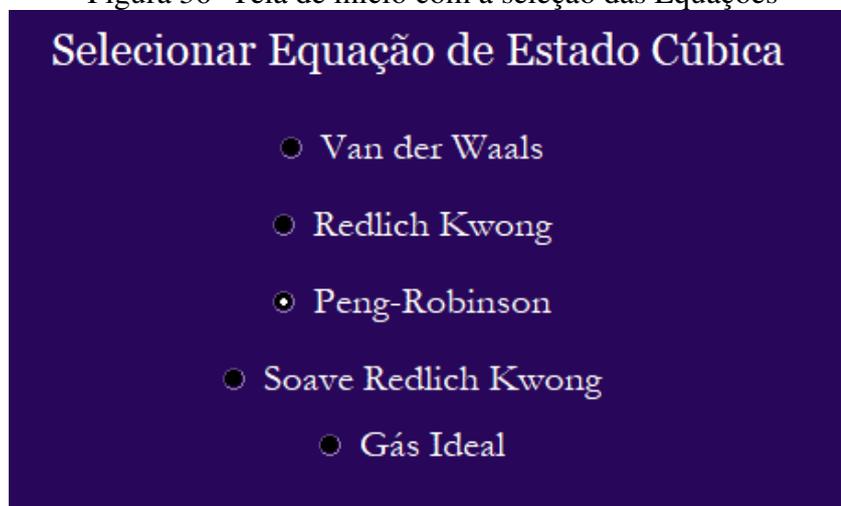
Fonte: elaborado pelo autor (2022)

Por meio da aplicação, utilizando a equação de Soave Redlich-Kwong, obteve-se o seguinte valor para o volume molar:

$$v = 4.08 \cdot 10^{-5} \text{ m}^3/\text{mol} \quad (4.90)$$

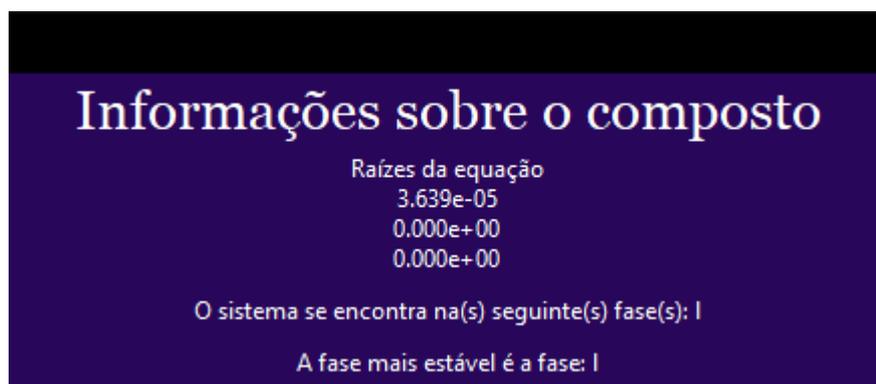
A diferença nos valores encontrados é de 7,84 %, um erro comparativo significativo, mas que com mais testes será possível entender melhor as diferenças de valores para as duas resoluções. Continuando a questão com o proposto pelo exercício e determinado o volume molar agora utilizando a Equação de Estado Cúbica de Peng-Robinson:

Figura 36- Tela de início com a seleção das Equações



Fonte: elaborado pelo autor (2022)

Figura 37- Tela de resultado com volume molar da Amônia



Fonte: elaborado pelo autor (2022)

A diferença nos valores encontrados é de 3,21%, menor do que no modelo anterior. Um fato importante neste exemplo é um baixo fator de acentricidade para a molécula, ou seja, ela não tem um formato próximo do esférico o que dificulta a análise para as interações intermoleculares, além de na solução analítica ser utilizado cartas de compressibilidade que podem gerar erros. Ademais, as condições do problema podem gerar dificuldade na implementação das equações de estado que sofrem desvios significativos próximo da região crítica.

4.8 Determinação do Volume Molar do Acetileno – Redlich-Kwong

Figura 38- Enunciado do Problema

4.36 Use a equação de Redlich–Kwong para calcular o tamanho do recipiente necessário para armazenar 30 kg de acetileno misturados com 50 kg de *n*-butano a 30 bar e 450 K. O coeficiente de interação binária é dado por $k_{12} = 0,092$.

Fonte: Koretsky (2007)

Determinar inicialmente o número de mols para cada substância, sendo o sufixo a para o acetileno e b para o butano:

$$n_a = \frac{m}{M} \quad (4.91)$$

$$n_a = \frac{30 \text{ kg}}{26,038 \cdot 10^{-3} \text{ Kg} \cdot \text{mol}^{-1}} \quad (4.92)$$

$$n_a = 1152,2 \text{ mol} \quad (4.93)$$

$$n_b = \frac{m}{M} \quad (4.94)$$

$$n_b = \frac{50 \text{ kg}}{58,123 \cdot 10^{-3} \text{ Kg} \cdot \text{mol}^{-1}} = 860,2 \text{ mol} \quad (4.95)$$

E a alimentação do sistema é dada pela proporção do número de mols quanto ao número de mols total do sistema, dessa forma:

$$y_a = 0,573 \quad (4.96)$$

$$y_b = 0,427 \quad (4.97)$$

Determinando os parâmetros a e b dos compostos:

$$a_a = \frac{0,42748R^2T_c^{2,5}}{P_c} = 8,03[J.m^3.K^{1,5}.mol^{-2}] \quad (4.98)$$

$$b_a = \frac{0,08664R^2T_c}{P_c} = 3,62.10^{-5}[m^3.mol^{-1}] \quad (4.99)$$

$$a_b = \frac{0,42748R^2T_c^{2,5}}{P_c} = 29,07[J.m^3.K^{1,5}.mol^{-2}] \quad (4.100)$$

$$b_b = \frac{0,08664R^2T_c}{P_c} = 8,081.10^{-5}[m^3.mol^{-1}] \quad (4.101)$$

$$a_{mis} = y_1^2 a_1 + 2y_1 y_2 a_{12} + y_2^2 a_2 \quad (4.102)$$

$$a_{mis} = 14,72 J.m^3.K^{1,5}.mol^{-2} \quad (4.103)$$

$$b_{mis} = y_1 b_1 + y_2 b_2 \quad (4.104)$$

$$b_{mis} = 5,53.10^{-5} m^3/mol \quad (4.105)$$

Substituindo os parâmetros na equação de Redlich-Kwong e calculando no Solver do Excel, tem-se:

Figura 39- Planilha com o volume molar da Mistura

	A	B	C	D	E	F	G	H	I	J	K	L
1	P=	3000000	Pa	Pressão Calculada		Diferença		$P = \frac{RT}{v-b} - \frac{a}{T^{1/2}v(v+b)}$				
2	R=	8,314	J/m.K		3000000	Pa	0,084354					
3	T=	365,15	K									
4	a=	14,72	K ^{1,5} .(m ³ /mol)									
5	b=	0,0000553	m ³ /mol									
6	v=	0,0007824	m ³ /mol									
7								Teste da Fórmula com o valor de v do Solver				
					4175338	-	1175338					
					P=	3000000	Pa					

Fonte: elaborado pelo autor (2022)

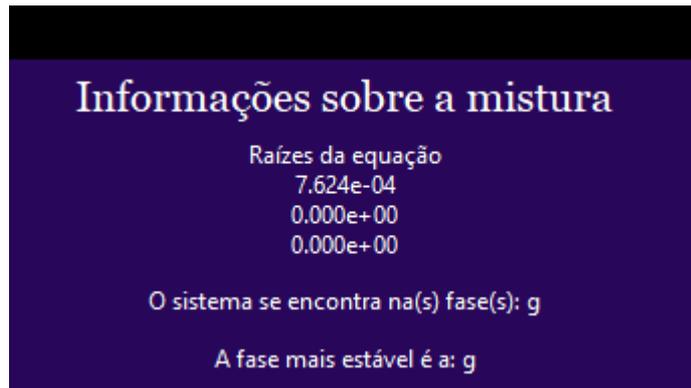
Volume molar da mistura

$$v = 7,824.10^{-4} m^3/mol \quad (4.106)$$

Por meio da aplicação, utilizando a equação de Soave Redlich-Kwong, obteve-se o seguinte valor para o volume molar:

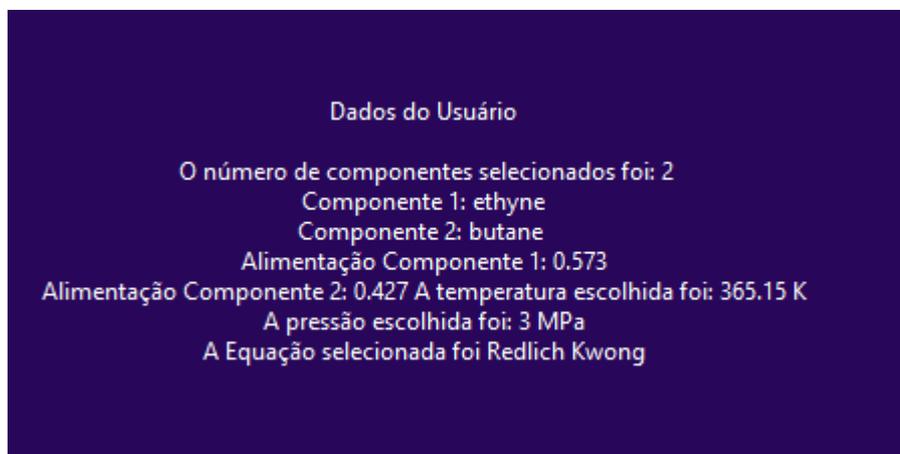
$$v = 7,624 \cdot 10^{-4} \text{ m}^3/\text{mol} \quad (4.107)$$

Figura 40- Tela de resultado com volume molar da Mistura



Fonte: elaborado pelo autor (2022)

Figura 41- Tela de resultado com volume molar da Mistura



Fonte: elaborado pelo autor (2022)

A diferença nos valores encontrados é de 2,55 %. Considerando a mistura e as condições de operação, é um erro aceitável e valida o algoritmo para uma situação de maior robustez nos cálculos computacionais feitos pela aplicação Thermo PySoup.

4.9 Compilação dos Erros

A figura a seguir demonstra uma compilação dos erros após a demonstração da resolução analítica e computacional das seções anteriores. O maior erro foi encontrado para o cálculo do volume molar da amônia pela Equação de Estado Cúbica de Redlich-Kwong e o menor erro também seguiu a escolha da mesma equação. Mais da metade dos problemas apresentou diferença inferior a 1% entre as duas soluções e para alguns erros maiores, há grande possibilidade de eles terem sido acrescidos pela utilização do Solver ao se buscar valores que zerassem a diferença do valor obtido pela equação e o valor real do problema.

Figura 42- Erros associados de cada problema

	Solução Analítica	Thermo PySoup	Diferença dos Valores(%)
Empresa Castor - Nitrogênio	1,910E-04	1,990E-04	4,02%
Empresa Castor - Oxigênio	1,550E-04	1,550E-04	0%
Etano – Item A	7,978E-04	7,970E-04	0,1 %,
Etano – Item B	7,518E-05	7,529E-05	0,146 %,
Propano- RK	1,095E-04	1,094E-04	0,37%
Propano - PG	9,420E-05	9,430E-05	0,106 %,
Amônia - RK	3,760E-05	4,080E-05	7,84 %,
Amônia - PG	3,760E-05	3,630E-05	3,21%
Acetileno	7,824E-04	7,624E-04	2,55 %.

Fonte: elaborado pelo autor (2022)

5 CONCLUSÃO

O desenvolvimento da aplicação tem como objetivo ser uma ferramenta de auxílio no cálculo de propriedades termodinâmicas para ensino didático e dado os resultados apresentados na seção anterior, esta é funcional. Na aprendizagem de termodinâmica química a ferramenta depende do conhecimento prévio do usuário para a utilização devida do programa, principalmente para selecionar a (s) equação (es) de estado que adequadamente resultam em valores próximos dos reais e então não substitui as horas de estudo necessárias para os alunos da graduação se tornarem aptes para resolver tais problemas.

Por se tratar de uma primeira versão, o programa necessita de mais testes e melhorias, principalmente no tratamento de erros e exceções, principalmente em situações que o sistema se encontra acima da região crítica. No menu “Sobre” da aplicação estão presentes os dados do desenvolvedor e quaisquer dúvidas e sugestões, bem como apresentação de bugs e demais erros do programa podem ser realizados por meio de e-mail ou comentários no projeto GitHub referente ao Thermo PySoup.

A utilização da biblioteca Tkinter resultou em algumas limitações, dentre elas a impossibilidade de construção de gráficos acima do primeiro Top Level. Dessa forma, eventuais trabalhos futuros precisarão de pesquisa de bibliotecas com mais recursos ou até mesmo um desenvolvimento front-end em outro ambiente de programação.

Ademais, a implementação do cálculo de pressão é uma funcionalidade que foi desenvolvida em versões futuras, bem como, a comparação de resultados entre modelos de Equações de Estado Cúbicas diferentes, com a possibilidade de salvar. Há ainda a possibilidade de se implementar mais Equações de Estado Cúbicas e se possível, expandir o cálculo para sistema com três componentes.

Para futuros trabalhos é possível a expansão do programa para a plataforma Android, de forma a facilitar o download e ampliação da base de usuários, reiterando o caráter pedagógico da aplicação Thermo PySoup.

REFERÊNCIAS

Caleb Bell and Contributors. **Thermo: Chemical properties component of Chemical Engineering Design Library (ChEDL)**. 2021

GUGGENHEIM, E.A. **Thermodynamics. An Advanced Treatment for Chemists and Physicists**, fifth revised edition, North-Holland Publishing Company, Amsterdam, page 157, 1967

KORETSKY, Milo D. **Termodinâmica para engenharia química** / Milo D. Koretsky; tradução Márcio José Estillac de Mello Cardoso, Oswaldo Esteves Barcia, Rosana Janot Martins, -Rio de Janeiro: LTC,2007

PENG, D.Y., and D.B. Robinson. **A New Two-Constant Equation of State**. Ind. Eng. Chem. Fundam. 15, 59–64. 1976

PERROT, Pierre. **A to Z of Thermodynamics**. [S.l.]: Oxford University Press. 1998

PITZER, K.S., **Thermodynamics of Eletrolytes. I. Theoretical Basis and General Equations** J. Phys. Chem., 77, No. 2, 268–277. 1973

REDLICH, O., and J.N.S. Kwong, **An Equation of State. Fugacities of Gaseous Solutions** Chem. Rev., 44, 233–244. 1949

SOAVE, G., Chem. **Equilibrium constants from a modified Redlich-Kwong equation of state** Eng. Sci., 27, 1197–1203. 1972

THOMSON, W. (Lord Kelvin) **On the Dynamical Theory of Heat, with Numerical Results Deduced from Mr Joule's Equivalent of a Thermal Unit, and M. Regnault's Observations on Steam** Transactions of the Royal Society of Edinburgh. Vol. 20. pp. 261-268, 289-298, 1851.

VAN DER WAALS, J. D. **On the Continuity of the Gaseous and Liquid States** (doctoral dissertation). [S.l.]: Universiteit Leiden. 1873

APÊNDICE A – CÓDIGO DO PROGRAMA

```

#Importações
from tkinter import *
from thermo.chemical import Chemical
from thermo import ChemicalConstantsPackage
from thermo.interaction_parameters import IPDB
import matplotlib.pyplot as plt
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

#-----
#-----

#GUI - Início - Primeiro Nível
root = Tk()
root.title('Thermo PySoup')
root.geometry('1400x1500')
root.configure(background='#280659',relief='solid')
root.iconbitmap('iconeSoup.ico')
#-----
#-----

# Menus da Aplicação

apresentacao = '***** Thermo PySoup - Versão 1.0 *****\n'
instrucao = '***** Instruções *****\n'

texto_ajuda = '* Leitura na Horizontal da Esquerda para a Direita\n\n' \
              '* Selecionar se o caso é para Substância Pura ou Mistura de\n\n' \
              'Dois Componentes\n\n' \
              '* -\n\n' \
              '> Digitar o(s) componentes com seus respectivos nomes em Inglês\n\n' \
              '* -\n\n' \
              '> Botão de Salvar retorna os valores digitados para efeito de comparação\n\n' \
              '* Alimentação referente a proporção molar de Alimentação\n\n' \
              '\n\n' \
              '* -> Podendo variar de 0.0 a 1.0\n\n' \
              '* Temperatura e Pressão seguindo o S.I\n\n' \
              '* -> Utilizar Kelvin e MPa, respectivamente\n\n' \
              '* Selecionar Equação de Estado Cúbica conforme faixas de ope\n\n' \
              'ração\n\n' \
              '* Gráfico gerado:\n\n' \
              '>Eixo X: Unidade de Volume em Metros Cúbicos(m³)\n\n' \
              '* -> Eixo Y: Unidade de Pressão em MPa'

```

```

#-----
# Ajuda ao Usuário

def caixa_de_ajuda():
    ajuda = Toplevel()
    ajuda.geometry('850x800')
    ajuda.title('Menu de Ajuda')

    apresentação = Label(ajuda, text= apresentacao, font='Arial 24',fg='#28
0659').pack()
    instrução = Label(ajuda, text= instrucao, font='Arial 21',fg='#21ce9c')
.pack()
    texto_de_ajuda = Label(ajuda, text= texto_ajuda, font='Arial 18').pack(
)

#-----
# Sobre
textosobre = 'Aplicação de Cunho Pedagógico para a Disciplina:\nTrabalho Fi
nal de Curso\n\n' \
            'Desenvolvido por: Artur Amaral Corrêa de Moraes\n\n' \
            ' -> e-mail: arturamaralc@gmail.com\n' \
            ' \t-> GitHub: https://github.com/ArturAmaral20'

def caixa_Sobre():
    sobre = Toplevel()
    sobre.geometry('850x400')
    sobre.title('Menu Sobre')

    inicio = Label(sobre, text=apresentacao, font='Arial 24',fg='#280659').
pack()
    sobre_label = Label(sobre, text='Sobre', font='Arial 21', fg='#21ce9c')
.pack()
    texto_sobre = Label(sobre, text=textosobre, font='Arial 18').pack()

#-----

```

```

meuMenu = Menu(root)
meuMenu.add_command(label='Ajuda',command= caixa_de_ajuda)
meuMenu.add_command(label='Sobre',command = caixa_Sobre)

#-----
#-----

# Número de Componentes

numero_de_componentes = IntVar()
def mostrarvalorcomponentes():
    print(f' O número de componentes selecionados foi: {numero_de_componentes.get()}')
    return (f' O número de componentes selecionados foi: {numero_de_componentes.get()}')

label = Label(text=' Substância Pura\n ou Multicomponente',font='Georgia 22',
              padx=5,pady=10,bd=5,background='#280659',fg='white').grid(row=0,column=0)
ra_1 = Radiobutton(text='Substância Pura',font='Garamond 17', variable=numero_de_componentes, value=1, command=mostrarvalorcomponentes,
                  padx=5,pady=5,bd=2,background='#280659',fg='white',selectcolor='black').grid(row=2,column=0)
ra_2 = Radiobutton(text='Multicomponente',font='Garamond 17', variable=numero_de_componentes, value=2, command=mostrarvalorcomponentes,
                  padx=5,pady=5,bd=2,background='#280659',fg='white',selectcolor='black').grid(row=4,column=0)
label_1 = Label(text='',padx=5,pady=5,bd=2,background='#280659',fg='white').grid(row=1,column=0)
label_2 = Label(text='',padx=5,pady=5,bd=2,background='#280659',fg='white').grid(row=3,column=0)
label_3 = Label(text='',padx=5,pady=5,bd=2,background='#280659',fg='white').grid(row=5,column=0)
label_4 = Label(text='',padx=5,pady=5,bd=2,background='#280659',fg='white').grid(row=6,column=0)

#-----
#-----

#Determinando os Componentes

comp = StringVar()
comp2 = StringVar()

```

```

comp_text = StringVar()

def mostrarcomponentes():
    print(comp.get(), comp2.get())
    comp_text.set(f'Componente 1: {comp.get()}\n'
                  f'Componente 2: {comp2.get()} ')
    return (f'Componente 1: {comp.get()}\n'
            f'Componente 2: {comp2.get()} ')

label1 = Label(text='Componentes', font='Georgia 22',border=20,
               padx=10,pady=10,bd=5,width=30,background='#280659',fg='white')
label1.grid(row=0,column=1)

labelc1 = Label(text='Componente 1:', font='Garamond 17',background='#280659',fg='white')
labelc1.grid(row=1,column=1)
entryc1 = Entry(textvariable=comp).grid(row=2,column=1)

labelc2 = Label(text='Componente 2:', font='Garamond 17',background='#280659',fg='white')
labelc2.grid(row=3,column=1)
entryc2 = Entry(textvariable=comp2).grid(row=4,column=1)

cmd = Button(text='Salvar', font='Garamond 17', command=mostrarcomponentes)
cmd.grid(row=5,column=1)
labelRetornoComp = Label(textvariable=comp_text,background='#280659',fg='white')
labelRetornoComp.grid(row=6,column=1)

#-----
#Alimentação dos Componentes

entrada1 = StringVar()
entrada2 = StringVar()
comp_ali_text = StringVar()

def mostrarAlimentacaoComponentes():
    print(entrada1.get(), entrada2.get())
    comp_ali_text.set(f'Alimentação Componente 1: {entrada1.get()}\n'
                     f'Alimentação Componente 2: {entrada2.get()} ')
    return (f'Alimentação Componente 1: {entrada1.get()}\n'
            f'Alimentação Componente 2: {entrada2.get()} ')

```

```

label1 = Label(text='Alimentação', font='Georgia 22',border=20,
               padx=10,pady=10,bd=5,width=31,background='#280659',fg='white'
               ).grid(row=0,column=2)

labelc1 = Label(text='Componente 1:', font='Garamond 17',background='#280659',fg='white').grid(row=1,column=2)
entryc1 = Entry(textvariable=entrada1).grid(row=2,column=2)

labelc2 = Label(text='Componente 2:', font='Garamond 17',background='#280659',fg='white').grid(row=3,column=2)
entryc2 = Entry(textvariable=entrada2).grid(row=4,column=2)

cmd = Button(text='Salvar', font='Garamond 18', command=mostrarAlimentacaoComponentes).grid(row=5,column=2)
labelRetornoComp = Label(textvariable=comp_ali_text,background='#280659',fg='white').grid(row=6,column=2)

#-----
#Entrada das Variáveis
T = StringVar()
P = StringVar()
temp_text = StringVar()

def mostrarentradaTeP():
    print(T.get(),P.get())
    temp_text.set(f'A temperatura escolhida foi: {T.get()} K\n'
                 f'A pressão escolhida foi: {P.get()} MPa')

    return (f'A temperatura escolhida foi: {T.get()} K\n'
           f'A pressão escolhida foi: {P.get()} MPa')

label1 = Label(text='Valores de Entrada',font='Georgia 18',border=20,width=30,
               padx=10,pady=10,bd=5,background='#280659',fg='white').grid(row=7,column=0)

labelt = Label(text='Temperatura (K):', font='Garamond 16',background='#280659',fg='white',
               padx=10,pady=10).grid(row=8,column=0)
entryt = Entry(textvariable=T).grid(row=9,column=0)

```

```

labelp = Label(text='Pressão (MPa):', font='Garamond 16',background='#280659',fg='white',
              padx=10,pady=10).grid(row=10,column=0)
labelp_vazio = Label(text='',background='#280659').grid(row=12,column=0)
entryp = Entry(textvariable=P).grid(row=11,column=0)

cmd = Button(text='Salvar',font='Garamond 18',command=mostrarentradaTeP).grid(row=13,column=0)
labelRetornoTemp = Label(textvariable=temp_text,background='#280659',fg='white').grid(row=14,column=0)

#-----
#-----
#Equações
valor_equacao = IntVar()

def mostrarEquacao():
    if(valor_equacao.get() == 1):
        print(f'A Equação selecionada foi Van der Waals')
        return 'A Equação selecionada foi Van der Waals'
    if (valor_equacao.get() == 2):
        print(f'A Equação selecionada foi Redlich Kwong')
        return 'A Equação selecionada foi Redlich Kwong'
    if (valor_equacao.get() == 3):
        print(f'A Equação selecionada foi Peng-Robinson')
        return 'A Equação selecionada foi Peng-Robinson'
    if (valor_equacao.get() == 4):
        print(f'A Equação selecionada foi Soave Redlich Kwong')
        return 'A Equação selecionada foi Soave Redlich Kwong'
    if (valor_equacao.get() == 5):
        print(f'A Equação selecionada foi para Gás Ideal')
        return 'A Equação selecionada foi Gás Ideal'

label = Label(text=' Selecionar Equação de Estado Cúbica',font='Georgia 18',border=20,width=30,
              padx=10,pady=10,bd=5,background='#280659',fg='white').grid(row=7,column=1)
ra_1 = Radiobutton(text='Van der Waals',font='Garamond 16', variable=valor_equacao, value=1, command=mostrarEquacao,

```

```

        background='#280659',fg='white',selectcolor='black').grid(
row=8,column=1)
ra_2 = Radiobutton(text='Redlich Kwong',font='Garamond 16', variable=valor_
equacao, value=2, command=mostrarEquacao,
        background='#280659',fg='white',selectcolor='black').grid(
row=9,column=1)
ra_3 = Radiobutton(text='Peng-
Robinson',font='Garamond 16', variable=valor_equacao, value=3, command=most
rarEquacao,
        background='#280659',fg='white',selectcolor='black').grid(
row=10,column=1)
ra_4 = Radiobutton(text='Soave Redlich Kwong',font='Garamond 16', variable=
valor_equacao, value=4, command=mostrarEquacao,
        background='#280659',fg='white',selectcolor='black').grid(
row=11,column=1)
ra_5 = Radiobutton(text='Gás Ideal',font='Garamond 16', variable=valor_equa
cao, value=5, command=mostrarEquacao,
        background='#280659',fg='white',selectcolor='black').grid(
row=12,column=1)

#-----
#-----

#Parte 2 - Top Level

def abrir():

#-----
#-----

#Termodinâmica

#-----
#-----

#Propriedades Físicas dos Compostos

compostol = Chemical(comp.get())
Tc1 = compostol.Tc
Pc1 = compostol.Pc
omegal = compostol.omega

if numero_de_componentes.get() == 2:
    composto2 = Chemical(comp2.get())

```

```

Tc2 = composto2.Tc
Pc2 = composto2.Pc
omega2 = composto2.omega

constants, correlations = ChemicalConstantsPackage.from_IDs([compos
to1.ID, composto2.ID])
kijs = IPDB.get_ip_asymmetric_matrix('ChemSep PR', constants.CASs,
'kij')

#-----
# Modificando o tipo das variáveis para se ajustar ao cálculo

Tcalc = float(T.get())
Pcalc = float(P.get())
Pcalc = Pcalc*1000000
Tc1calc = float(Tc1)
Pc1calc = float(Pc1)
entradalcalc = float(entrada1.get())
omegalcalc = float(omega1)

if numero_de_componentes.get() == 2:
    Tc2calc = float(Tc2)
    Pc2calc = float(Pc2)
    entrada2calc = float(entrada2.get())
    omega2calc = float(omega2)

#-----
#Definindo Equação de Estado Cúbica

if numero_de_componentes.get() == 1:
    if (valor_equacao.get() == 1):
        from thermo.eos import VDW

        eos = VDW(Tc1calc, Pc1calc, Tcalc, Pcalc)
        print(eos)

    if (valor_equacao.get() == 2):
        from thermo.eos import RK

        eos = RK(Tc1calc, Pc1calc, Tcalc, Pcalc)
        print(eos)

```

```

if (valor_equacao.get() == 3):
    from thermo.eos import PR

    eos = PR(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc)
    print(eos)

if (valor_equacao.get() == 4):
    from thermo.eos import SRK

    eos = SRK(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc)
    print(eos)

if (valor_equacao.get() == 5):
    from thermo.eos_volume import volume_solutions_ideal

    eos = volume_solutions_ideal(Tcalc, Pcalc)
    print(eos)

if numero_de_componentes.get() == 2:
    if (valor_equacao.get() == 1):
        from thermo.eos_mix import VDWMIX

        eos = VDWMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], Pcs=[Pc1
calc, Pc2calc], omegas=[1, 1], kijs=kijs, zs=[entrada1calc, entrada2calc])
        print(eos)
        print(eos.V_l, eos.V_g)
        print(eos.fugacities_l, eos.fugacities_g)

    if (valor_equacao.get() == 2):
        from thermo.eos_mix import RKMIX

        eos = RKMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], Pcs=[Pc1c
alc, Pc2calc], kijs=kijs, zs=[entrada1calc, entrada2calc])
        print(eos)
        #print(eos.V_l, eos.V_g)
        #print(eos.fugacities_l, eos.fugacities_g)

if (valor_equacao.get() == 3):
    from thermo.eos_mix import PRMIX

```

```

        eos = PRMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], Pcs=[Pc1calc, Pc2calc],
        omegas=[omegalcalc, omega2calc], kijs=kijs, zs=[entrada1calc, entrada2calc])
        print(eos)
        print(eos.V_l, eos.V_g)
        print(eos.fugacities_l, eos.fugacities_g)

    if (valor_equacao.get() == 4):
        from thermo.eos_mix import SRKMIX

        eos = SRKMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], Pcs=[Pc1calc, Pc2calc],
        omegas=[omegalcalc, omega2calc], kijs=kijs, zs=[entrada1calc, entrada2calc])
        print(eos)
        print(eos.V_l, eos.V_g)
        print(eos.fugacities_l, eos.fugacities_g)

    if (valor_equacao.get() == 5):
        from thermo.eos_volume import volume_solutions_ideal

        eos = volume_solutions_ideal(Tcalc, Pcalc)
        print(eos)

#-----
#-----
# Informações sobre a mistura
    if (valor_equacao.get() != 5):
        raizes_da_equacao = eos.raw_volumes
        raizes_da_equacao = (f'Raízes da equação\n'
        f'{raizes_da_equacao[0]:.3e}\n'
        f'{raizes_da_equacao[1]:.3e}\n'
        f'{raizes_da_equacao[2]:.3e}')
        fase_do_sistema = eos.phase
        fase_mais_estavel = eos.more_stable_phase

# Determinando o volume de líquido e gás para a situação-problema:

    if numero_de_componentes.get() == 1:
        if (valor_equacao.get() == 1):
            try:
                Vg = VDW(Tc1calc, Pc1calc, Tcalc, Pcalc).V_g
            except:

```

```

        Vg = 0
    try:
        Vl = VDW(Tc1calc, Pc1calc, Tcalc, Pcalc).V_l
    except:
        Vl = 0

    print('Equação Cúbica de Estado de VDW')
    print(f'O volume de líquido é de : {Vl} m³')
    print(f'O volume de vapor é de : {Vg} m³')

if (valor_equacao.get() == 2):
    try:
        Vg = RK(Tc1calc, Pc1calc, Tcalc, Pcalc).V_g
    except:
        Vg = 0
    try:
        Vl = RK(Tc1calc, Pc1calc, Tcalc, Pcalc).V_l
    except:
        Vl = 0

    print('Equação Cúbica de Estado de RK')
    print(f'O volume de líquido é de : {Vl} m³')
    print(f'O volume de vapor é de : {Vg} m³')

if (valor_equacao.get() == 3):
    try:
        Vg = PR(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc).V_g
    except:
        Vg = 0
    try:
        Vl = PR(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc).V_l
    except:
        Vl = 0

    print('Equação Cúbica de Estado de PR')
    print(f'O volume de líquido é de : {Vl} m³')
    print(f'O volume de vapor é de : {Vg} m³')

if (valor_equacao.get() == 4):
    try:
        Vg = SRK(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc).V_g
    except:
        Vg = 0

```

```

try:
    Vl = SRK(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc).V_l
except:
    Vl = 0

print('Equação Cúbica de Estado de SRK')
print(f'O volume de líquido é de : {Vl} m³')
print(f'O volume de vapor é de : {Vg} m³')

if numero_de_componentes.get() == 2:
    if (valor_equacao.get() == 1):
        try:
            Vg = eos.V_g
        except:
            Vg = 0
        try:
            Vl = eos.V_l
        except:
            Vl = 0

        print('Equação Cúbica de Estado de VDW')
        print(f'O volume de líquido é de : {Vl} m³')
        print(f'O volume de vapor é de : {Vg} m³')

    if(valor_equacao.get() == 2):
        try:
            Vg = eos.V_g
        except:
            Vg = 0
        try:
            Vl = eos.V_l
        except:
            Vl = 0

        print('Equação Cúbica de Estado de RK')
        print(f'O volume de líquido é de : {Vl} m³')
        print(f'O volume de vapor é de : {Vg} m³')

    if(valor_equacao.get() == 3):
        try:
            Vg = eos.V_g
        except:

```

```

        Vg = 0
    try:
        Vl = eos.V_l
    except:
        Vl = 0

    print('Equação Cúbica de Estado de PR')
    print(f'O volume de líquido é de : {Vl} m³')
    print(f'O volume de vapor é de : {Vg} m³')

if (valor_equacao.get() == 4):
    try:
        Vg = eos.V_g
    except:
        Vg = 0
    try:
        Vl = eos.V_l
    except:
        Vl = 0

    print('Equação Cúbica de Estado de SRK')
    print(f'O volume de líquido é de : {Vl} m³')
    print(f'O volume de vapor é de : {Vg} m³')

#-----
#-----
#Determinando as Propriedades Saturadas
    try:
        Psat = eos.Psat(Tcalc)
    except:
        print('Está além da temperatura crítica')

    try:
        VlSat = eos.V_l_sat(Tcalc)
        VgSat = eos.V_g_sat(Tcalc)
        print(f'Vl sat = {VlSat} m³\nVg sat = {VgSat} m³')
    except:
        print('Está além da temperatura crítica')

#-----
#-----
# Gráfico
    try:
        Volume_liq_eiox = list()

```

```

Volume_gas_eixox = list()
Pressao_eixoy = list()

Pcalc = 1000

if numero_de_componentes.get() == 1:
    if (valor_equacao.get() == 1):
        for Pcalc in range(Pcalc, 1150 * Pcalc, 100):
            Vg = VDW(Tc1calc, Pc1calc, Tcalc, Pcalc).V_g
            Volume_gas_eixox.append(Vg)

            Vl = VDW(Tc1calc, Pc1calc, Tcalc, Pcalc).V_l
            Volume_liq_eixox.append(Vl)

            Pressao_eixoy.append(Pcalc)

        print(Pressao_eixoy)
        print(Volume_liq_eixox)
        print(Volume_gas_eixox)

    if (valor_equacao.get() == 2):
        for Pcalc in range(Pcalc, 1150 * Pcalc, 100):
            Vg = RK(Tc1calc, Pc1calc, Tcalc, Pcalc).V_g
            Volume_gas_eixox.append(Vg)

            Vl = RK(Tc1calc, Pc1calc, Tcalc, Pcalc).V_l
            Volume_liq_eixox.append(Vl)

            Pressao_eixoy.append(Pcalc)

        print(Pressao_eixoy)
        print(Volume_liq_eixox)
        print(Volume_gas_eixox)

    if (valor_equacao.get() == 3):
        for Pcalc in range(Pcalc, 1150 * Pcalc, 100):
            Vg = PR(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc).V_g
            Volume_gas_eixox.append(Vg)

            Vl = PR(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc).V_l
            Volume_liq_eixox.append(Vl)

            Pressao_eixoy.append(Pcalc)

```

```

        print(Pressao_eixoy)
        print(Volume_liq_eixox)
        print(Volume_gas_eixox)

    if (valor_equacao.get() == 4):
        for Pcalc in range(Pcalc, 1150 * Pcalc, 100):
            Vg = SRK(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc).V_
g
            Volume_gas_eixox.append(Vg)

            Vl = SRK(Tc1calc, Pc1calc, omegalcalc, Tcalc, Pcalc).V_
l
            Volume_liq_eixox.append(Vl)

            Pressao_eixoy.append(Pcalc)

        print(Pressao_eixoy)
        print(Volume_liq_eixox)
        print(Volume_gas_eixox)
    except:
        print('Gráfico não será criado\nValores acima do ponto crítico!')

    try:
        if numero_de_componentes.get() == 2:
            if (valor_equacao.get() == 1):
                for Pcalc in range(Pcalc, 1150 * Pcalc, 100):
                    Vg = VDWMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], P
cs=[Pc1calc, Pc2calc], omegas=[1, 1], kijs=kijs, zs=[entrada1calc, entrada2
calc]).V_g
                    Volume_gas_eixox.append(Vg)

                    Vl = VDWMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], P
cs=[Pc1calc, Pc2calc], omegas=[1, 1], kijs=kijs, zs=[entrada1calc, entrada2
calc]).V_l
                    Volume_liq_eixox.append(Vl)

                    Pressao_eixoy.append(Pcalc)

                print(Pressao_eixoy)
                print(Volume_liq_eixox)
                print(Volume_gas_eixox)

            if (valor_equacao.get() == 2):

```

```

    for Pcalc in range(Pcalc, 1150*Pcalc,100):
        Vg = RKMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], Pcs=[P
c1calc, Pc2calc], kijs=kijs, zs=[entrada1calc, entrada2calc]).V_g
        Volume_gas_eiox.append(Vg)

        Vl = RKMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], Pcs=[P
c1calc, Pc2calc], kijs=kijs, zs=[entrada1calc, entrada2calc]).V_l
        Volume_liq_eiox.append(Vl)

        Pressao_eioxoy.append(Pcalc)

    print(Pressao_eioxoy)
    print(Volume_liq_eiox)
    print(Volume_gas_eiox)

if(valor_equacao.get() == 3):
    for Pcalc in range(Pcalc, 1150*Pcalc,100):
        Vg = PRMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], Pcs=[P
c1calc, Pc2calc], omegas=[omegalcalc, omega2calc], kijs=kijs, zs=[entrada1c
alc, entrada2calc]).V_g
        Volume_gas_eiox.append(Vg)

        Vl = PRMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], Pcs=[P
c1calc, Pc2calc], omegas=[omegalcalc, omega2calc], kijs=kijs, zs=[entrada1c
alc, entrada2calc]).V_l
        Volume_liq_eiox.append(Vl)

        Pressao_eioxoy.append(Pcalc)

    print(Pressao_eioxoy)
    print(Volume_liq_eiox)
    print(Volume_gas_eiox)

if (valor_equacao.get() == 4):
    for Pcalc in range(Pcalc, 1150 * Pcalc, 100):
        Vg = SRKMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], P
cs=[Pc1calc, Pc2calc],
                                omegas=[omegalcalc, omega2calc], kijs=kijs,
zs=[entrada1calc, entrada2calc]).V_g
        Volume_gas_eiox.append(Vg)

```

```

        Vl = SRKMIX(T=Tcalc, P=Pcalc, Tcs=[Tc1calc, Tc2calc], P
cs=[Pc1calc, Pc2calc],
                    omegas=[omegalcalc, omega2calc], kijs=kijs,
zs=[entradalcalc, entrada2calc]).V_l
        Volume_liq_eiox.append(Vl)

        Pressao_eiox.append(Pcalc)

        print(Pressao_eiox)
        print(Volume_liq_eiox)
        print(Volume_gas_eiox)
    except:
        print('Não será criado o Gráfico.\nEstá acima do ponto crítico!')
#-----
#Criando o Gráfico

    try:
        plt.title(f'Envelope de Equilíbrio')
        plt.ylabel('Pressão (Pa)')
        plt.xlabel('Volume (M³)')
        plt.plot(Volume_gas_eiox, Pressao_eiox,color='#280659')
        plt.plot(Volume_liq_eiox, Pressao_eiox,color='#21ce9c')
        plt.rcParams["figure.figsize"] = (5, 3)
        plt.xlim(-0.01,0.09)
        plt.axhline(y=Psat, xmin=0, xmax=1,color='y')
        data = [
            [VlSat, Psat],
            [VgSat, Psat],
            [Vl, Pcalc],
            [Vg, Pcalc],
        ]

        x, y = zip(*data)
        plt.scatter(x, y)

        plt.annotate('Vl sat',
                    horizontalalignment = 'right',
                    verticalalignment = 'bottom',
                    xy = (VlSat, Psat),
                    arrowprops = dict(facecolor = 'green', shrink = 0.05))

        plt.annotate('Vg sat',
                    horizontalalignment = 'left',

```

```

        verticalalignment = 'bottom',
        xy = (VgSat, Psat),
        arrowprops = dict(facecolor = 'green', shrink = 0.05))

plt.legend(['Vg', 'Vl', 'Psat'], loc=0) #Varia de 0 a 9
#plt.grid()
#plt.show()
plt.savefig('grafico.png')
#plt.close()
except:
    print('Sem gráfico para a situação!\nAcima do ponto crítico')

#-----
#GUI - Início - Segundo Nível

top = Toplevel()
top.geometry('1400x800')
top.config(background='#280659')

if numero_de_componentes.get() == 1:
    label_dados_componentes = Label(top, text=f'Dados dos composto {com
p.get()}', font='Georgia 20',
                                   background='#280659', fg='white').gr
id(row=0, column=0)
    label_componente1 = Label(top, text=f'Temperatura Crítica Composto
1= {Tc1}\nPressão Crítica Composto1={Pc1}\nFator de Acentricidade1 = {omega1
}',
                              background='#280659', fg='white').grid(row
=1, column=0)

    label_informacoes = Label(top, text='Informações sobre o composto',
font='Georgia 20',
                              background='#280659', fg='white').grid(row
=0, column=1)
    label_raizes = Label(top, text=raizes_da_equacao,
                          background='#280659', fg='white').grid(row=1, c
olumn=1)
    label_fase = Label(top, text=f'O sistema se encontra na(s) seguinte
(s) fase(s): {fase_do_sistema}',

```

```

        background='#280659',fg='white').grid(row=2, col
umn=1)
        label_mais_estavel = Label(top, text=f'A fase mais estável é a fase
: {fase_mais_estavel}',
        background='#280659',fg='white').grid(ro
w=3, column=1)

        label_volumes = Label(top, text='Volumes obtidos',font='Georgia 16'
,
        background='#280659',fg='white').grid(row=2,
column=0)
        label_Vl = Label(top, text=f'O volume de líquido é de : {Vl:.3e} m³
',
        background='#280659',fg='white').grid(row=3, colum
n=0)
        label_Vg = Label(top, text=f'O volume de vapor é de : {Vg:.3e} m³',
        background='#280659',fg='white').grid(row=4, colum
n=0)
        label_soma = Label(top, text=f'O volume total é de : {(Vg + Vl):.3e
} m³',
        background='#280659',fg='white').grid(row=
5, column=0)
        label_usu6 = Label(top, text='',
        background='#280659',fg='white').grid(row=
6, column=0)

        label_Prop_Saturadas = Label(top, text='Valores das Propriedades Sa
turadas',font='Georgia 16',
        background='#280659',fg='white').grid(
row=4, column=1)
        label_PSat = Label(top, text=f'Pressão Saturada = {Psat:.3e} Pa',
        background='#280659',fg='white').grid(row=5, col
umn=1)
        label_Vl_Sat = Label(top, text=f'Volume de Líquido Saturado = {VlSa
t:.3e} m³',
        background='#280659',fg='white').grid(row=6, c
olumn=1)
        label_Vg_Sat = Label(top, text=f'Volume de Gás Saturado = {VgSat:.3
e} m³',
        background='#280659',fg='white').grid(row=7, c
olumn=1)

        label_usu1 = Label(top, text='',

```

```

        background='#280659',fg='white').grid(row=
4, column=0)
        label_usu2 = Label(top, text='',
        background='#280659',fg='white').grid(row=
5, column=0)
        label_usu3 = Label(top, text='',
        background='#280659',fg='white').grid(row=
6, column=0)

text_usuario = StringVar()
text_usuario.set(f'Dados do Usuário\n\n'
        f'{mostrarvalorcomponentes()}\n'
        f'{mostrarcomponentes()}\n'
        f'{mostrarAlimentacaoComponentes()}'
        f'{mostrarentradaTeP()}\n'
        f'{mostrarEquacao()}'')
        labelRetornoUsuario = Label(top, textvariable=text_usuario, backgro
und='#280659', fg='white').grid(row=8,column=1)

        label_Grafico = Label(top, text='Gráfico:',font='Georgia 20',
        background='#280659',fg='white').grid(row=7,
column=0)
        img = PhotoImage(file='grafico.png')
        label_imagem = Label(top, image=img).grid(row=8, column=0)

        if numero_de_componentes.get() == 2:
            label_dados_componentes = Label(top,text=f'Dados dos compostos {com
p.get()} e {comp2.get()}',font='Georgia 20',
            background='#280659',fg='white').gr
id(row=0,column=0)
            label_componente1 = Label(top,text=f'Tc1= {Tc1} Pc1={Pc1}\nomega1 =
{omegal}',
            background='#280659',fg='white').grid(row
=1,column=0)
            label_componente2 = Label(top, text=f'Tc2= {Tc2} Pc2={Pc2}\nomega2
={omega2}',
            background='#280659',fg='white').grid(row
=2,column=0)
            label_matriz_Kij = Label(top, text='',
            background='#280659',fg='white').grid(row=
3,column=0)

```

```

        label_informacoes= Label(top, text='Informações sobre a mistura', font='Georgia 16',
                                background='#280659', fg='white').grid(row=
0, column=1)
        label_raizes= Label(top, text=raizes_da_equacao,
                            background='#280659', fg='white').grid(row=1, col
umn=1)
        label_fase= Label(top, text=f'O sistema se encontra na(s) fase(s):
{fase_do_sistema}',
                           background='#280659', fg='white').grid(row=2, colum
n=1)
        label_mais_estavel= Label(top, text=f'A fase mais estável é a: {fas
e_mais_estavel}',
                                   background='#280659', fg='white').grid(row
=3, column=1)

        label_usu1 = Label(top, text='', background='#280659', fg='white').gr
id(row=4, column=1)
        label_usu2 = Label(top, text='', background='#280659', fg='white').gr
id(row=5, column=1)
        label_usu3 = Label(top, text='', background='#280659', fg='white').gr
id(row=6, column=1)
        label_usu4 = Label(top, text='', background='#280659', fg='white').gr
id(row=7, column=1)

        text_usuario = StringVar()
        text_usuario.set(f'Dados do Usuário\n\n'
                        f'{mostrarvalorcomponentes()}\n'
                        f'{mostrarcomponentes()}\n'
                        f'{mostrarAlimentacaoComponentes()}'
                        f'{mostrarentradaTeP()}\n'
                        f'{mostrarEquacao()}')

        labelRetornoUsuario = Label(top, textvariable=text_usuario, backgrou
nd='#280659', fg='white').grid(row=8, column=1)

        label_volumes = Label(top, text='Volumes obtidos', font='Georgia 16'
, background='#280659', fg='white').grid(row=4, column=0)
        label_Vl = Label(top, text=f'O volume de líquido é de : {Vl:.3e} m³
', background='#280659', fg='white').grid(row=5, column=0)
        label_Vg = Label(top, text=f'O volume de vapor é de : {Vg:.3e} m³',
background='#280659', fg='white').grid(row=6, column=0)

```

```

        label_Prop_Saturadas = Label(top, text='Valores das Propriedades Sa
turadas',font='Georgia 16',
                                background='#280659',fg='white').grid(
row=4,column=1)
        label_PSat = Label(top, text=f'Psat = {Psat:.3e} Pa',background='#2
80659',fg='white').grid(row=5,column=1)
        label_Vl_Sat = Label(top, text=f'Vl Sat = {VlSat:.3e} m³',backgroun
d='#280659',fg='white').grid(row=6,column=1)
        label_Vg_Sat = Label(top, text=f'Vg Sat = {VgSat:.3e} m³',backgroun
d='#280659',fg='white').grid(row=7,column=1)

        label_Grafico = Label(top, text='Gráfico:',font='Georgia 16',backgr
ound='#280659',fg='white').grid(row=7,column=0)
        img = PhotoImage(file='grafico.png')
        label_imagem = Label(top, image=img).grid(row=8,column=0)

#-----
#-----

#Botão que executa o Top Level
#label_volume = Label(root,text='').grid(row=14,column=0)
label_calcular_volume = Label(root,text='', background='#280659').grid(row=
13,column=1)
btn = Button(root, text='Calcular Volume',relief='raised',bd=7,font='Garamo
nd 20',bg='#21ce9c',fg='white',anchor='n',
            width=20,command=abrir).grid(row=14,column=1)

#-----
#-----

root.config(menu=meuMenu)
root.mainloop()

```