

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ÍTALO MOREIRA QUINTÃO

**RESOLUÇÃO DE MODELOS MATEMÁTICOS PARA ENGENHARIA QUÍMICA
UTILIZANDO API**

FRANCISCO BELTRÃO

2022

ÍTALO MOREIRA QUINTÃO

**RESOLUÇÃO DE MODELOS MATEMÁTICOS PARA ENGENHARIA QUÍMICA
UTILIZANDO API**

Resolution of mathematical models for chemical engineering using an API

Trabalho de conclusão de curso de graduação apresentada como requisito para obtenção do título de Bacharel em Engenharia Química da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Vilmar Steffen.

Co-orientador: Prof. Dr. Francisco Antonio Fernandes Reinaldo.

FRANCISCO BELTRÃO

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ÍTALO MOREIRA QUINTÃO

**RESOLUÇÃO DE MODELOS MATEMÁTICOS PARA ENGENHARIA QUÍMICA
UTILIZANDO API**

Trabalho de conclusão de curso de graduação apresentada como requisito para obtenção do título de Bacharel em Engenharia Química da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Vilmar Steffen.

Co-orientador: Prof. Dr. Francisco Antonio Fernandes Reinaldo.

Data de aprovação: 07/Dezembro/2022

Vilmar Steffen

Doutor

Universidade Tecnológica Federal do Paraná – UTFPR

Jeconias Rocha Guimarães

Doutor

Universidade Tecnológica Federal do Paraná – UTFPR

Francisco Antônio Fernandes Reinaldo

Doutor

Universidade Tecnológica Federal do Paraná – UTFPR

"A folha de aprovação assinada encontra-se na Coordenação do Curso"

FRANCISCO BELTRÃO

2022

AGRADECIMENTOS

Agradeço primeiramente a paciência, atenção e amor dos meus pais durante toda minha trajetória acadêmica, sem vocês eu não estaria aqui.

Agradeço à Luana, pelas palavras, incentivos, sugestões e por fazer parte da minha vida durante estes anos, sei que todo este trabalho teria sido muito mais árduo sem a sua companhia.

Agradeço a todos meus amigos cursistas, especialmente Bruno Henrique Manfrin Tormem e Carlos Catusso Della Pasqua, sem a amizade e parceria de vocês não estaria concluindo esta fase tão importante da minha vida.

Também aos amigos mais antigos, que mesmo à quilômetros de distância me incentivaram a concluir o curso e me profissionalizar cada vez mais, muito obrigado Gustavo Faria, Gabriel Lopes, Jefferson Andrade e Lucas Moreira.

Agradeço ao meu orientador Vilmar Steffen, do qual sempre me incentivou nas minhas ideias, sendo parte importantíssima do meu desenvolvimento profissional e pessoal ao longo do curso.

Agradeço também ao coorientador Francisco Antonio Fernandes Reinaldo, pelas inúmeras dicas e formas para poder valorizar ainda mais este trabalho.

Agradeço aos membros da banca, por fazer parte deste dia tão importante para minha formação e por apoiarem a minha ideia em um tema que sou tão apaixonado.

Por fim, agradeço a instituição UTFPR - Francisco Beltrão, que por anos foi uma segunda casa e parte tão importante da minha história.

RESUMO

Ao passar dos anos, a alta demanda tecnológica fez com que o mercado tivesse de se adequar a processos cada vez mais automatizados, gerando assim uma crescente demanda por insumos provenientes de linguagens de programação e engenharia como um todo. Neste presente trabalho implementou-se uma API (*Application Programming Interface*) utilizando a linguagem de programação Python, para solução de modelos matemáticos envolvendo a área da engenharia química, gerando assim espaço para uma possível implementação destes modelos em diversas outros *softwares*. Para o desenvolvimento do trabalho, foram empregados modelos de balanço de massa multicomponentes sem reação química e soluções para raízes de equação de estado cúbicas utilizando a equação de Cardano-Tartaglia como forma de exemplificação de modelos que podem ser empregados na API. Este trabalho também propôs e implementou uma aplicação Android como forma de simular uma aplicação real utilizando a API. Os resultados deste presente trabalho atenderam às expectativas, demonstrando que a resolução destes modelos por meio de API trouxe agilidade e possibilidades para diversas áreas envolto da engenharia química.

Palavras-chave: python; ferramenta computacional; engenharia química; api

ABSTRACT

Over the years, the high technological demand forced the market adapt to increasingly automated processes, generating a growing demand for inputs from programming and engineering. In this work, an API (Application Programming Interface) was implemented using the Python programming language to solve mathematical models involving the chemical engineering, making possible the implementation of these models in several other ways. For the development of the work, multicomponent mass balance models and roots finding for cubic equation of state using Cardano-Tartaglia were used as a way of exemplifying models that can be used in the API. This work also proposed the development of an Android application to simulate a application using the API. The results of this present work were positive, demonstrating that the resolution of these models through API brought agility and possibilities to several areas involved in chemical engineering.

Keywords: python; computational tool; chemical engineering; api

LISTA DE ILUSTRAÇÕES

Figura 1 - Fluxograma de funcionamento de uma API.....	14
Figura 2 - Exemplo comum de sistemas nos quais são feitos	15
Figura 3 - Fluxograma do funcionamento da API.....	27
Figura 4 - Fluxograma de resolução ferramenta de balanços de massa multicomponentes	28
Figura 5 - Fluxograma de resolução ferramenta de cálculo das raízes de estado cúbicas.....	29
Figura 6 - Fluxograma da utilização da API pelas ferramentas	29
Figura 7 - Fluxograma da utilização da API no desenvolvimento.....	30
Figura 8 - Modelos matemáticos dentro da API.....	31
Figura 9 - Tela inicial da aplicação Android.....	32
Figura 10 - Painel sobre os detalhes do desenvolvimento da aplicação	33
Figura 11 - Tela para seleção de modelos matemáticos disponíveis para resolução	34
Figura 12 - Tela de descrição do modelo “Gráfico do fator de compressibilidade”	35
Figura 13 - Tela de resolução de balanços de massa multicomponentes	36
Figura 14 - Erro de balanço envolvendo balanços impossíveis	37
Figura 15 - Erro de balanços por não atingir o número mínimo de variáveis	38
Figura 16 - Erro de balanço devido a frações mássicas fora do intervalo	39
Figura 17 - Telas (A) e (B) da inserção de dados para o modelo “Gráfico do fator de compressibilidade”	40
Figura 18 - Erro devido a campos vazios no modelo “Gráficos do fator de compressibilidade”	41
Figura 19 - Erro devido a grandes requisições no modelo “Gráfico do fator de compressibilidade”	42
Figura 20 - Erro envolvendo equação impossível no modelo “Gráfico do fator de compressibilidade”	43
Figura 21 - Exemplo de balanço de massa multicomponentes simples	44
Figura 22 - Inserção de dados no exemplo de balanço simples	45
Figura 23 – Telas (A) e (B) das inserções dos dados conhecidos no modelo de balanço simples	46
Figura 24 - Resultado do balanço simples desde E1 até S1 e seus componentes	47
Figura 25 - Resultado do balanço simples para componentes de S1 até S2 e seus componentes.....	48
Figura 26 - Resultado do balanço simples para componentes S2.....	49
Figura 27 - Exemplo de balanço de massa multicomponentes complexo	50
Figura 28 - Inserção das entradas, saídas e componentes do exemplo de balanço de massa complexo.....	51
Figura 29 – Telas (A) e (B) das inserções dos valores conhecidos de E1, E2 e seus componentes no exemplo de balanço de massa complexo	52
Figura 30 - Telas (A) e (B) das inserções dos valores conhecidos de E3, S1 e seus componentes no exemplo de balanço de massa complexo	53

Figura 31 - Telas (A) e (B) das inserções dos valores conhecidos de S2, S3 e seus componentes no exemplo de balanço de massa complexo	54
Figura 32 – Inserção dos valores conhecidos de S4 e seus componentes no exemplo de balanço de massa complexo	55
Figura 33 - Telas (A) e (B) dos resultados obtidos de E1, E2 e seus componentes no exemplo de balanço de massa complexo.....	56
Figura 34 - Telas (A) e (B) dos resultados obtidos de E3, S1 e seus componentes no exemplo de balanço de massa complexo.....	57
Figura 35 - Telas (A) e (B) dos resultados obtidos de S2, S3 e seus componentes no exemplo de balanço de massa complexo.....	58
Figura 36 - Resultados obtidos de S4 e seus componentes no exemplo de balanço de massa complexo	59
Figura 37 - Telas (A) e (B) das inserções dos dados no exemplo do modelo “Gráficos do fator de compressibilidade” para 3 raízes reais	60
Figura 38 - Resultado do exemplo do modelo “Gráficos do fator de compressibilidade” para 3 raízes reais	61
Figura 39 - Telas (A) e (B) das inserções dos dados no exemplo do modelo “Gráficos do fator de compressibilidade” para 1 raiz real	62
Figura 40 - Resultado do exemplo do modelo “Gráficos do fator de compressibilidade” para 1 raiz real	63

LISTA DE TABELAS

Tabela 1 - Parâmetros de equação de estado.....	23
---	----

SUMÁRIO

1	INTRODUÇÃO	10
2	OBJETIVOS	12
2.1	Objetivo geral	12
2.2	Objetivos específicos	12
3	FUNDAMENTAÇÃO TEÓRICA	13
3.1	Programação em Python	13
3.1.1	O que é uma API?	13
3.2	Modelos matemáticos a partir de balanços de massa	14
3.2.1	Balanço de massa para sistemas multicomponentes	15
3.3	Solução de sistemas não lineares pelo método de Newton	17
3.4	Método de Broyden	18
3.5	Raízes de equação de estado cúbicas	21
4	PROCEDIMENTOS METODOLÓGICOS	26
5	RESULTADOS E DISCUSSÕES	30
5.1	Testes da aplicação – Estudos de caso	43
5.1.1	Estudo de caso 1 – Balanço de massa em um sistema binário	43
5.1.2	Estudo de caso 2 – Balanço de massa multicomponente	49
5.1.3	Estudo de caso 3 – 3 raízes reais do fator de compressibilidade.....	59
5.1.4	Estudo de caso 4 – 1 raiz real do fator de compressibilidade	62
6	CONCLUSÃO	65
	REFERÊNCIAS	66

1 INTRODUÇÃO

Com a progressão da indústria 4.0 e o avanço da era da informação, tornam-se necessárias soluções tecnológicas em diversas áreas para resolução de problemas cotidianos.

Tais circunstâncias incentivaram o surgimento significativo de linguagens de programação, algumas tiveram seu devido destaque por serem de fácil uso e possuírem comunidade muito ativa. Um ótimo exemplo destas comunidades são os referentes à linguagem Python, linguagem de programação de sintaxe simples e de fácil compartilhamento de pacotes, onde os próprios usuários desenvolvem métodos para outros usuários utilizarem, método comum em softwares de código aberto. Sabemos que “Python é software livre, ou seja, pode ser utilizada gratuitamente, graças ao trabalho da Python Foundation e de inúmeros colaboradores. É possível utilizar Python em praticamente qualquer arquitetura de computadores ou sistema operacional” (MENEZES, 2010).

Outro notável resultado do avanço das linguagens de programação é o surgimento da API (*Application Programming Interface*), solução tecnológica que faz uma ponte entre o requerente e requerido, fazendo com que dados deixem de ser locais e passem a ter um acesso global ou privado, por meio de um protocolo *web*, podendo assim ser utilizado em diversos tipos de aplicações por requisição da sua URL (*Uniform Resource Locator*). Como por exemplo, projetos que envolvam internet das coisas, e até aplicações *mobile* com acesso à internet.

Devido a todos os avanços, áreas antes dominadas por procedimentos manuais e de longa duração, passaram a adotar novos métodos para sua resolução, e na área das engenharias, inclusive a engenharia química não foi diferente. Equacionamentos envolvendo balanços de massa, diagramas de fases, conversões de unidades em processos químicos, dentre outras áreas voltadas à engenharia química conseguiram, devido a este avanço, tornar prático a resolução destes equacionamentos. “[...] O perfil de profissional, ou seja, o engenheiro do futuro certamente deverá procurar ter uma visão holística dos processos; estar sempre em sintonia com sistemas produtivos, softwares, nanotecnologia e biotecnologia” (FREITAS, et al., 2019).

Desta maneira, várias aplicações após a atualização da lógica ou implementação de novas funcionalidades necessitam de uma atualização para todos

os usuários visando utilizar a nova versão. Porém, ao atualizar uma API, não é necessário que o usuário final faça download, tendo em vista que o software tão-somente faz requisições para a API, portanto, apenas a forma que ela lhe retorna informação é importante para a funcionalidade da aplicação.

Em virtude disto, o compartilhamento rápido de informações mediante a equipe de engenheiros e a formação de um conhecimento tecnológico acerca das inovações é um ponto crucial para resolução de problemas de maneira rápida e eficiente em meio científico, tornando assim a corrida pela resolução de novas soluções muito mais dinâmica e disputada em todo o mercado.

Deste modo, é proposto para este trabalho o desenvolvimento de uma API para auxílio na resolução modelos matemáticos relacionados à engenharia química, como compartilhamento de dados e resolução de modelos matemáticos disponibilizados pelos usuários da plataforma.

2 OBJETIVOS

2.1 Objetivo geral

O objetivo deste trabalho é a criação de uma API para facilitação e solução de modelos matemáticos específicos entre engenheiros químicos utilizando a linguagem de programação *Python* e a biblioteca de desenvolvimento *Flask*.

2.2 Objetivos específicos

- Desenvolvimento de uma API para resolução de modelos matemáticos para engenharia química;
- Possibilitar o desenvolvimento de novas ferramentas computacionais para engenharia química a partir da API;
- Agilidade e praticidade na utilização da API.

3 FUNDAMENTAÇÃO TEÓRICA

Com o intuito da realização deste projeto, é necessário base de conhecimento em áreas como programação em *Python*, elaboração e conhecimento de balanços de massa de processos químicos e raízes de equação de estado cúbicas, sendo estes dois últimos modelos matemáticos disponibilizados através da API. Com o intuito de melhor esclarecimento do conteúdo, este capítulo é dividido sobre estes temas.

3.1 Programação em Python

Python é uma linguagem de programação com estrutura simples e de fácil entendimento para desenvolvedores, a fim de facilitar o processo de produção de códigos e otimizar o desenvolvimento de lógicas envolvendo a linguagem. Sendo assim, é possível afirmar que, “o Python é uma linguagem de programação dinâmica e orientada a objetos, que pode ser utilizada no desenvolvimento de qualquer tipo de aplicação, científica ou não. O Python oferece suporte à integração com outras linguagens e ferramentas, e é distribuído com uma vasta biblioteca padrão.” (COELHO, 2007).

Desta forma, com intuito de explorar a parte do desenvolvimento científico da linguagem de programação, foi proposta a produção da API para a facilitação deste processo.

3.1.1 O que é uma API?

A sigla API é uma abreviação para *Application Programming Interface*, que nada mais é do que uma ponte entre usuário e aplicação, como demonstrado na Figura 1.

Figura 1 - Fluxograma de funcionamento de uma API



Fonte: Autoria própria (2022)

Por consequência, compartilhamento de informação via API se tornou um modelo amplamente utilizado pelo seu alto índice de uso *cross-plataform*, podendo então ser empregado para desenvolver softwares em outras linguagens de programação, assim, com o foco para atender a maioria dos sistemas operacionais.

Uma vez que os dados estão disponibilizados via URL em um *website*, qualquer aplicação cuja linguagem de programação detenha uma biblioteca para obtenção dos dados pode fazer o uso destes e então utilizar em seu desenvolvimento.

Um grande exemplo sobre esse tipo de facilidade são os cadastros em *websites* utilizando contas de outras plataformas maiores como Google ou Facebook, onde uma API disponibiliza seus dados básicos presentes nestas plataformas para cadastro de usuários em outras aplicações.

3.2 Modelos matemáticos a partir de balanços de massa

Ao longo de sua formação, são apresentados ao engenheiro químico, diversos fundamentos e princípios relacionados aos conhecimentos necessários para sua profissão. Uma das teorias extremamente difundidas durante o curso é o conhecimento acerca de balanços de massa, conteúdo com alto índice de aplicabilidade no cotidiano do engenheiro químico.

Desta maneira, um fluxograma é uma forma de se representar graficamente um processo, em que descreve um processo real com detalhes suficientes de modo a utilizá-lo para formular e resolver balanços de massa (e de energia), onde também estes fluxogramas são utilizados para resolução de problemas operacionais e otimização de desempenho de processos, sendo uma ferramenta chave para descrever de forma simples e completa, um processo químico. (HIMMELBLAU, 2014).

Fluxogramas, como o ilustrado na Figura 2, representam processos dos quais seguem os fenômenos presentes propostos por Lavoisier em 1774 pela sua observação da lei da conservação de massa, onde a massa total dos produtos é a mesma massa dos reagentes. Desta forma, em sua aplicação no processo de balanço de massa pode ser demonstrada a partir da Equação (1):

Figura 2 - Exemplo comum de sistemas nos quais são feitos



Fonte: Autoria própria (2022)

$$\text{Acúmulo} = \text{Entrada} - \text{Saída} + \text{Geração} - \text{Consumo} \quad (1)$$

Para os casos em que não há reação química e estão operando em estado estacionário, os termos de acúmulo, geração e/ou consumo são nulos, desta forma, a Equação (1) pode ser simplificada onde $\sum E_i$ é a representação do somatório das correntes de entrada do problema e $\sum S_j$ das correntes de saída do problema, sendo representada pela Equação (2).

$$\sum_{i=1}^n E_i - \sum_{j=1}^m S_j = 0 \quad (2)$$

3.2.1 Balanço de massa para sistemas multicomponentes

Como a maior parte dos sistemas envolvendo balanços de massa em sua prática apresenta uma grande gama de misturas, a resolução destes não se faz apenas por balanços globais, mas também, balanços por componentes presentes nestas correntes, onde se aumenta cada vez mais a complexidades destes balanços.

Levando em consideração o fluxograma demonstrado na Figura 2, é possível determinar um balanço de massa multicomponente pela Equação (3), onde E_i e S_j são os fluxos presentes nas correntes i e j de entrada e saída respectivamente e x_i^k e y_j^k representam as frações molares ou mássicas do componente k presentes respectivamente nas correntes de entrada e saída nestes fluxos onde o somatório das frações, representadas nas Equações (4) e (5) devem ser igual a 1 .

$$\sum_{i=1}^n E_i x_i^k - \sum_{j=1}^m S_j y_j^k = 0 \quad \forall k = 1, \dots, p \quad (3)$$

$$\sum_{k=1}^p x_i^k = 1 \quad \forall i = 1, \dots, m \quad (4)$$

$$\sum_{k=1}^p y_j^k = 1 \quad \forall j = 1, \dots, n \quad (5)$$

Porém, é possível perceber que quanto maior o número de entradas, saídas e componentes, maior o nível de complexidade do problema. Assim, geralmente são utilizados sistemas de equações e métodos numéricos para auxílio da resolução de balanços de massa cujo número total de variáveis é muito alto.

Tal número é definido pela Equação (6) que adota n como número total de correntes de entrada, m como número total de correntes de saída e p como número de componentes para cada corrente. Vale ressaltar que estas variáveis terão o mesmo significado em todas as equações a seguir neste item.

$$N_{inc} = (m + n) (p + 1) \quad (6)$$

Por conseguinte, estes sistemas precisam um número de variáveis conhecidas no problema para possibilitar sua resolução. Conceito conhecido como graus de liberdade.

Graus de liberdade são descritos como o número de variáveis para obtenção de uma única solução para o sistema, tendo em vista que caso não haja

conhecimento deste número, o sistema poderá ter infinitas soluções, onde caso fixado um número de variáveis maior que o grau de liberdade, existe a possibilidade de uma não solução do sistema. Conceito este descrito pela Equação (7).

$$GL = (m + n - 1) p \quad (7)$$

3.3 Solução de sistemas não lineares pelo método de Newton

Ao longo da graduação, é possível encontrar inúmeros sistemas de equações não lineares que necessitam de resolução, sendo necessários ferramentas robustas para resolução, gerando assim perda de tempo e capacidade computacional avançada para a resolução destes sistemas. Portanto, se faz necessário cada vez mais soluções e modelos matemáticos para facilitação e aperfeiçoamento destes métodos.

Dentre as diversas formas para resolução destes problemas, o método de Newton se destaca pelo uso da derivada (inclinação da reta tangente entre os pontos da função $f(x)$). Desta maneira, o método de Newton por meio destas retas tangentes aproxima a função proposta por uma função linear a partir da solução aproximada da iteração atual. É destacável que, “é fundamental que se perceba que o Método de Newton não determina a solução exata da equação trabalhada e sim sua raiz aproximada [...]” (MACHADO, ALVES, 2013).

Vale ressaltar que, todos os métodos iterativos utilizados para a resolução destes sistemas lineares, podem ser tratados com a função $x_{k+1} = \Phi(x_k)$, e a partir destes métodos nos é fornecido a sequência de vetores $\{x_k\}_{k=0}^{\infty}$, sendo que x_k é uma aproximação da solução, sendo k o número de interações. É de se esperar também que, após uma estimativa inicial x_0 , a solução para esta função pode convergir ou divergir para uma raiz do sistema onde $F(x) = 0$ (SOUZA, 2015).

A principal etapa do processo iterativo do método de Newton é representada pela Equação (8).

$$x_{k+1} = x_k - J^{-1}(x_k)F(x_k) \quad (8)$$

A partir desta equação é esperado obter uma convergência quadrática, sendo uma estimativa inicial fornecida para a solução do sistema de equações não lineares. Entretanto, é necessário que exista a inversa da matriz Jacobiana, onde se faz indispensável a resolução de um sistema linear para cada iteração. A matriz Jacobiana é representada pela Equação (9) (SOUZA, 2015).

$$J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \quad (9)$$

Em sua implementação, é possível evitar o cálculo de $J^{-1}(x)$ e do produto $J^{-1}(x_k)F(x_k)$ calculando a solução de um sistema de equações, expressa pela Equação (10) (SOUZA, 2015).

$$J(x_k)S_k = -F(x_k) \quad (10)$$

Desta forma é possível a determinação do vetor S_k , e a partir desta resolução, a Equação (11) demonstra (x_{k+1}) sendo a soma de S_k e x_k , reduzindo o esforço computacional entre as iterações (SOUZA, 2015).

$$(x_{k+1}) = x_k + S_k \quad (11)$$

Após a compreensão da usabilidade e falhas do método de Newton, é perceptível que o método pode convergir com um número pequeno de interações caso seja fornecida estimativa inicial de qualidade. (SOUZA, 2015)

3.4 Método de Broyden

Apesar de ser um método bastante utilizado, o método de Newton é computacionalmente caro devido ao cálculo complexo submetido pelo computador para resolução da matriz Jacobiana, assim como o método de quase-Newton, ou método de Broyden, onde seu diferencial é evitar a divergência e poupar parte de tal esforço computacional.

Desta maneira, o método de Broyden consiste no uso do fator de relaxamento diferente da unidade apenas caso haja necessidade para evitar uma possível divergência, desta maneira, a atualização da solução aproximada é feita segundo a Equação (12) (BROYDEN, 1965).

$$X_{(k+1)} = X_k + s_1^k \Delta X_k \quad (12)$$

Como forma padrão, em cada iteração do método, inicialmente tem-se como valor de s a unidade ($s_1^k = 1,0$). O motivo para o uso deste valor se dá por ele já ser comumente utilizado pelo método de Newton. Desta maneira, caso a desigualdade apresentada na Equação (13), seja verdadeira mantém-se o valor de $s = 1$ (BROYDEN, 1965).

$$\sqrt{\sum_{j=1}^n f_j^2 (X_k + s_1^k \Delta X_k)} < \sqrt{\sum_{j=1}^n f_j^2 (X_k)} \quad (13)$$

Entretanto, caso seja falso a afirmação proposta pela Equação (13), é necessário a correção do valor de s estabelecida pela Equação (14) (BROYDEN, 1965).

$$s_2^k = \frac{\sqrt{1 + 6\eta} - 1}{3\eta} \quad (14)$$

Sendo η a razão entre as normas Euclidianas apresentada na Equação (15) (BROYDEN, 1965).

$$\eta = \frac{\sum_{j=1}^n f_j^2 (X_k + s_1^k \Delta X_k)}{\sum_{j=1}^n f_j^2 (X_k)} \quad (15)$$

Desta forma, considerando a Equação (16) (BROYDEN, 1965).

$$\varphi(s) = \sum_{j=1}^n f_j^2 (X + s\Delta X) \quad (16)$$

Iremos encontrar o valor para η expressa pela Equação (17) (BROYDEN, 1965).

$$\eta = \frac{\varphi(1)}{\varphi(0)} \quad (17)$$

Sendo assim, é mostrado que se f_j possuir a característica de uma função linear de s , juntamente a uma jacobiana inversa, desta maneira, $\varphi(s)$ seria uma função quadrática de s com seu ponto nulo em $s = 1$. Porém, como esta função não pode ser negativa, cabe então possuir essa derivada nula no tempo igual a 1 (BROYDEN, 1965).

Desta maneira, é evidenciado que para satisfazer as condições de $s = 1$, nossa função $\varphi_q(s)$ é representada pela Equação (18) (BROYDEN, 1965).

$$\varphi_q(s) \equiv (1 - 2s + s^2) \varphi(0) \quad (18)$$

Porém, dificilmente o quadrado da norma Euclidianas no ponto $s = 1$ é zero, desta forma, é assumido que isto deve ser considerado apenas na presença de um termo cúbico adicional. Desta maneira, a aproximação cúbica $\varphi_c(s)$ é demonstrado pela Equação (19) (BROYDEN, 1965).

$$\varphi_c(s) \equiv \varphi_q(s) + a s^3 \quad (19)$$

Após a substituição de $s = 1$ demonstrada na Equação (19), é possível se obter a expressão representada pela Equação (20) (BROYDEN, 1965).

$$a = \varphi(1) \quad (20)$$

Substituindo a seguinte consideração nas Equações (18) - (20), $\varphi_c(s)$, definido de acordo com a Equação (21) (BROYDEN, 1965).

$$\varphi_c(s) \equiv (1 - 2s + s^2) \varphi(0) + \varphi(1) s^3 \quad (21)$$

Para a otimização da função, é necessário encontrar os valores de s que minimizem a Equação (21). Realizando a derivada da Equação (21) em relação a s , e igualando-a a 0, é obtido a Equação (22) (BROYDEN, 1965).

$$\frac{3\eta}{2}s^2 + s - 1 = 0 \quad (22)$$

Fazendo uso da fórmula de Bhaskara, é possível encontrar a equação isolada em s para valores de s positivos, como expresso na Equação (23) (BROYDEN, 1965).

$$s = \frac{-1 \pm \sqrt{1 + 6\eta}}{3\eta} \quad (23)$$

3.5 Raízes de equação de estado cúbicas

Equações de estado estão presentes em vários momentos da graduação do engenheiro químico, desde as primeiras aulas envolvendo o conteúdo de físico-química até termodinâmica e fenômenos de transferência. A primeira equação cúbica que nos é apresentada é a de Van der Waals, sendo representada pela Equação (24).

$$P = \frac{RT}{V - b} - \frac{a}{V^2} \quad (24)$$

Após a propostas da equação de Van der Waals, várias outras foram propostas, sendo que uma forma de representar várias delas é utilizando a Equação (25) (STEFFEN, SILVA, 2021):

$$P = \frac{RT}{V - b} - \frac{a}{V^2 + \lambda bV + \sigma b^2} \quad (25)$$

Sendo que como característica própria, a maioria dos trabalhos que envolvem equações cúbicas de estado levam em consideração a relação $\eta = b$, desta maneira, o termo de a e b geralmente são representados por expressões envolvendo pressão e temperatura crítica da substância. À vista disso, surgem novas variáveis como λ e σ que são independentes da substância, onde cada uma delas representa uma equação de estado (STEFFEN, SILVA, 2021).

Desta maneira, é possível reescrever a Equação (25) de maneira que se seja formada um polinômio cúbico com os valores de a e b , onde Λ e Γ são valores positivos que dependem da equação de estado, como demonstrado na Equação (26) (STEFFEN, SILVA, 2021).

$$V^3 - \left(b - \lambda b + \frac{RT}{P}\right)V^2 + \left(\sigma b^2 - \lambda b^2 - \lambda b \frac{RT}{P} + \frac{a}{P}\right)V - \left(\sigma b^3 - \sigma b^2 \frac{RT}{P} - b \frac{a}{P}\right) = 0 \quad (26)$$

$$a = \Lambda \frac{R^2 T_c^2}{P_c} \alpha(\omega, T_r) = \frac{\Lambda}{\Gamma} b R T_c \alpha(\omega, T_r) \quad (27)$$

$$b = \Gamma \frac{R T_c}{P_c} \quad (28)$$

Desta maneira, é possível representar o fator de compressibilidade pela Equação (29), sendo de grande importância em indústrias para dimensionamento de tanques por exemplo.

$$Z = \frac{PV}{RT} \quad (29)$$

São apresentados na Tabela 1 os parâmetros para os modelos de equação de estado mais conhecidas como van der Waals (vdW), Redlich-Kwong (RK), Soave-Redlich-Kwong (SRK) e Peng-Robinson (PR) (STEFFEN, SILVA, 2021).

Tabela 1 - Parâmetros de equação de estado

Equação de estado	λ	σ	Λ	Γ	$\alpha(\omega, T_r)$
vdW	0	0	$\frac{27}{64}$	$\frac{1}{8}$	1
RK	1	0	0,42748	0,08664	$\frac{1}{\sqrt{T_r}}$
SRK	1	0	0,42748	0,08664	$(1 + (0,480 + 1,574\omega - 0,176\omega^2)(1 - \sqrt{T_r}))^2$
PR	2	-1	0,45724	0,07780	$(1 + (0,37464 + 1,54226\omega - 0,26992\omega^2)(1 - \sqrt{T_r}))^2$

Fonte: Steffen e Silva (2021)

Por conseguinte, é possível a resolver a Equação (26) a partir da fórmula de Cardano-Tartaglia, fórmula esta que pode ser utilizada para resolução polinômios cúbicos que possuem coeficientes reais, sendo representada pela Equação (30) (STEFFEN, SILVA, 2021).

$$x^3 + \alpha x^2 + \beta x + \gamma = 0 \quad (30)$$

O modelo de Cardano-Tartaglia possui um parâmetro de teste para a verificação da quantidade de raízes reais e complexas, expresso pela Equação (31): (STEFFEN, SILVA, 2021).

$$D = Q^3 + R^2 \quad (31)$$

Sendo:

$$R = \frac{9\alpha\beta - 27\gamma - 2\alpha^3}{53} \quad (32)$$

$$Q = \frac{3\beta - \alpha^2}{9} \quad (33)$$

Portanto, há três possibilidades para valor de D : Caso $D > 0$, a equação possui uma raiz real e um par de raízes complexas conjugadas; Caso $D < 0$, a equação possui três raízes reais diferentes; e, caso $D = 0$, a equação contém três raízes reais com duas ou três raízes repetidas, onde caso $D < 0$, estas raízes podem ser calculadas utilizando as Equações (34) - (36) (STEFFEN, SILVA, 2021).

$$x_1 = 2\sqrt{-Q} \cos\left(\frac{\theta + 4\pi}{3}\right) - \frac{\alpha}{3} \quad (34)$$

$$x_2 = 2\sqrt{-Q} \cos\left(\frac{\theta + 2\pi}{3}\right) - \frac{\alpha}{3} \quad (35)$$

$$x_3 = 2\sqrt{-Q} \cos\left(\frac{\theta}{3}\right) - \frac{\alpha}{3} \quad (36)$$

Onde:

$$\theta = \arccos\left(\frac{R}{\sqrt{-Q^3}}\right) \quad (37)$$

Porém, quando se tem apenas uma raiz real e um par de raízes complexas conjugadas, sendo assim $D > 0$, o cálculo das raízes pode ser feito utilizando as Equações (38) - (40) (STEFFEN, SILVA, 2021).

$$x_1 = S + T - \frac{\alpha}{3} \quad (38)$$

$$x_2 = -\frac{1}{2}(S + T) - \frac{\alpha}{3} + i\frac{\sqrt{3}}{2}(S - T) \quad (39)$$

$$x_3 = -\frac{1}{2}(S + T) - \frac{\alpha}{3} - i\frac{\sqrt{3}}{2}(S - T) \quad (40)$$

Onde:

$$S = \operatorname{sgn}\left(R + \sqrt{Q^3 + R^2}\right) \sqrt[3]{\left|R + \sqrt{Q^3 + R^2}\right|} \quad (41)$$

$$T = \operatorname{sgn}\left(R - \sqrt{Q^3 + R^2}\right) \sqrt[3]{\left|R - \sqrt{Q^3 + R^2}\right|} \quad (42)$$

$$\operatorname{sgn}(y) = \begin{cases} +1 & \text{para } y > 0 \\ 0 & \text{para } y = 0 \\ -1 & \text{para } y < 0 \end{cases} \quad (43)$$

4 PROCEDIMENTOS METODOLÓGICOS

Logo no início do desenvolvimento da API, foi realizado um estudo para constatar as funcionalidades mínimas e esperadas para a aplicação, a fim de evitar problemas relacionados à etapa inicial do desenvolvimento que poderiam gerar perda de produtividade, prejudicando também a ponto chave do desenvolvimento que é a simplicidade na adição de novas aplicações.

Para o desenvolvimento da API, foi utilizada a linguagem de programação Python, linguagem de sintaxe simples, robusta, de fácil compreensão pelo desenvolvedor e que por possuir a política de código aberto, possui uma vasta comunidade com grande variedade de bibliotecas de desenvolvimento disponíveis.

A escolha da biblioteca para criação da API foi dividida entre o uso do Django, *FastAPI* ou *Flask*, porém, *Flask* se mostrou a melhor escolha devido a sua facilidade de desenvolvimento e grande comunidade ativa para estudo e auxílio na aplicação, sendo utilizado até por grandes empresas como Airbnb e Netflix.

Então, foi debatido a melhor forma para realização de envio de dados entre usuário e aplicação. Ainda na primeira fase do desenvolvimento, o envio de dados era realizado pelo corpo do URL do requerente, contudo, devido à complexidade do procedimento foi optado pelo envio dos dados utilizando JSON, pois,

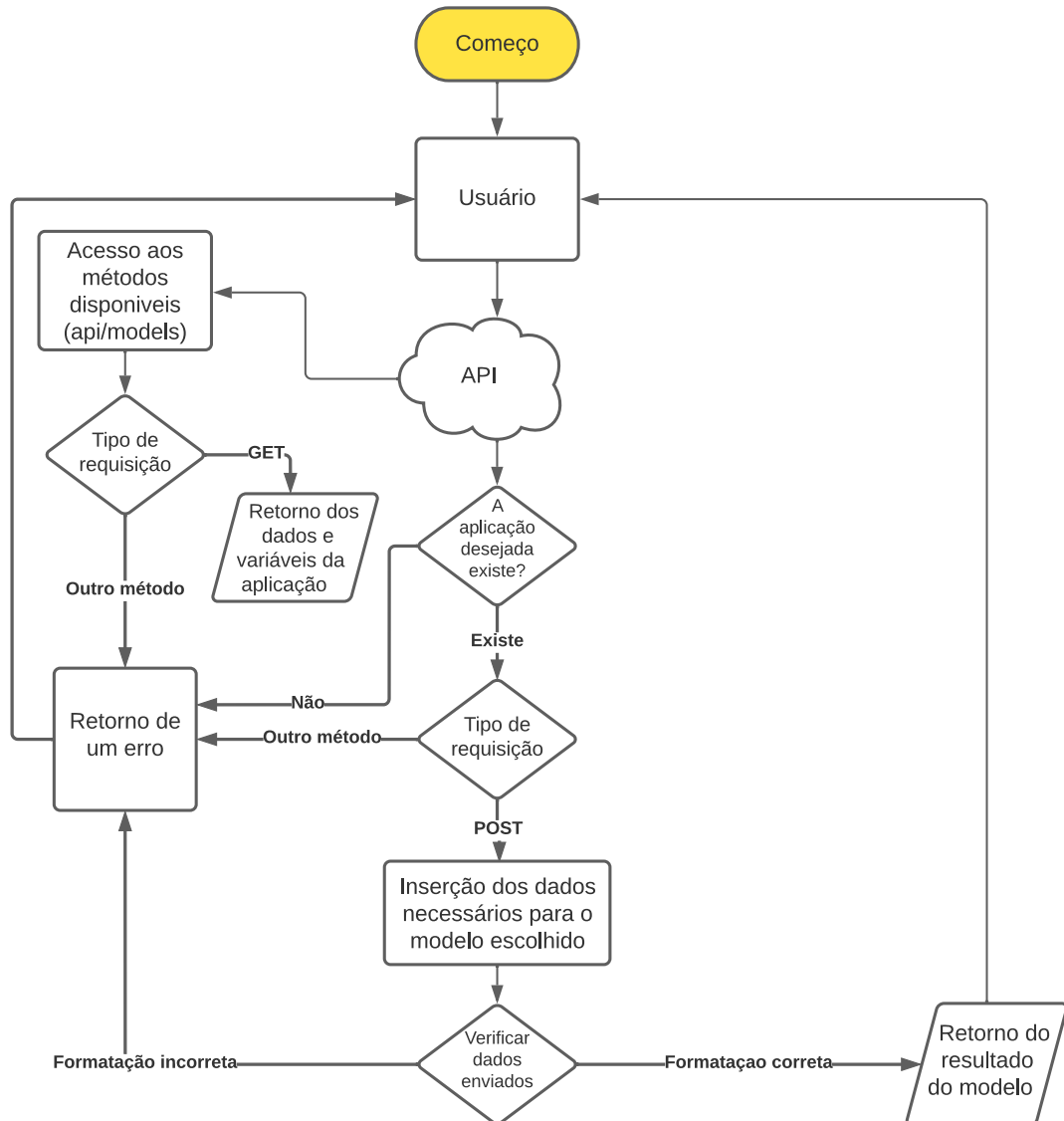
JSON é uma estrutura de texto no formato de objeto Javascript e pode ser composto de subobjetos. Sua estrutura possibilita uma diversificada representação de dados, pois o mesmo suporta quatro tipos de dados primitivos e estrutura de subobjetos, como *strings*, *number*, *integer*, *boolean*, *object* e *array*. (JÚNIOR, SILVA, 2018)

Após, houve novo debate acerca da liberdade do usuário para upload de seus próprios modelos matemáticos na aplicação, porém, esta característica foi vista como inadequada para a proposta do TCC, tendo em vista que para o arquivo ter sucesso em seu compartilhamento utilizando a API, é necessário certo tratamento prévio para conectar os dados do requerente com as variáveis mínimas requisitadas pela aplicação. Desta maneira, foi estabelecido um formulário para a realização do upload dos arquivos na aplicação, a fim de realizar uma avaliação prévia do pedido do requerente e então, a conexão com a API¹.

¹ Disponível em: <https://forms.gle/1yB7NfovLkx9XRfR6>

Desta forma, foi estipulado a lógica de funcionamento da API, visando a resolução dos modelos matemáticos e transferência de dados, como mostrado no fluxograma descrito na Figura 3.

Figura 3 - Fluxograma do funcionamento da API



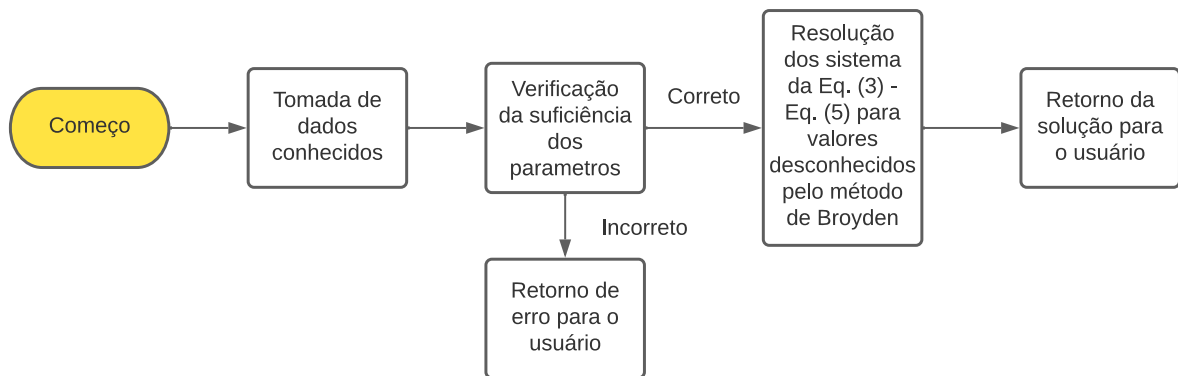
Fonte: Autoria própria (2022)

Em seguida, após o desenvolvimento de um modelo piloto da API, foi proposto o desenvolvimento de duas aplicações mobile para exemplificação de seu uso como fonte de processamento de dados, sendo, um modelo focado no balanço de massa de multicomponentes e outro no cálculo das raízes de estado cúbicas e visualização gráfica dos resultados.

O foco para o desenvolvimento destes aplicativos foi o sistema operacional *Android*, utilizando no seu desenvolvimento a IDE Android Studio juntamente a linguagem de programação Java, altamente utilizada para diversos tipos de sistema, tanto *desktop*, quanto *web* e *mobile*.

Para a produção dos aplicativos, foi proposto a utilização do modelo de desenvolvimento de software MVC (*Model-View-Controller*), que “é simples, todas as requisições da aplicação são direcionadas para a camada Controller, que acessa a camada Model para processar a tal requisição, e por fim exibe o resultado da camada View” (LUCIANO, ALVES, 2011).

Figura 4 - Fluxograma de resolução ferramenta de balanços de massa multicomponentes

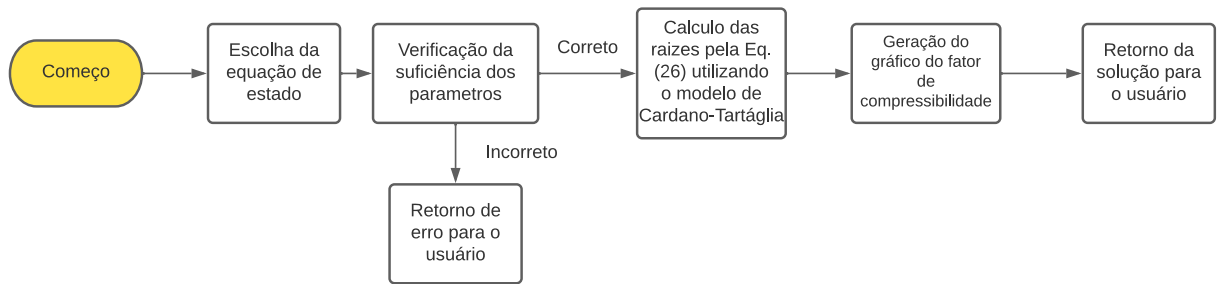


Fonte: Autoria própria (2022)

É apresentado na Figura 4 a lógica de uso da aplicação envolvendo balanço de massa para multicomponentes, sendo necessário seguir as etapas propostas como:

- Nos valores estabelecidos pelo usuário de entrada, saída e componentes, são necessários que o usuário forneça números do tipo inteiro;
- É necessário que o usuário respeite o número variáveis de graus de liberdade, como mostra a Equação (6) para este tipo de sistema;
- Se torna indispensável a atribuição de valores iniciais para os componentes dos quais o usuário tenha ciência, dados estes que dependendo do componente escolhido pelo usuário, é necessário que seja um fluxo mássico atribuído por números do tipo *float* ou até frações mássicas que vão de 0 até 1, também do tipo de dado *float*.

Figura 5 - Fluxograma de resolução ferramenta de cálculo das raízes de estado cúbicas



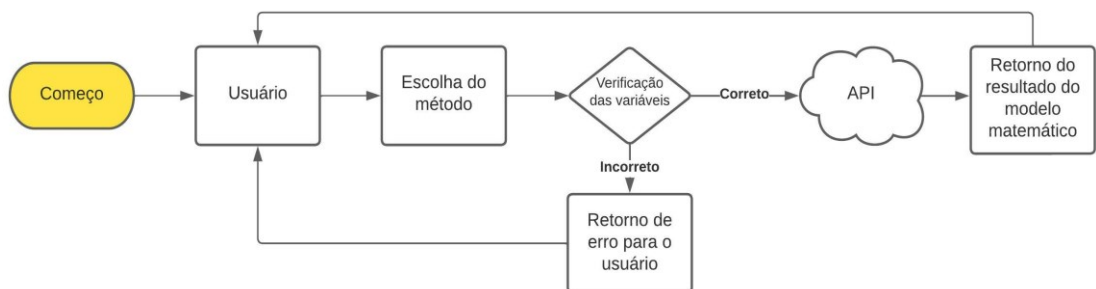
Fonte: Autoria própria (2022)

Já para a aplicação envolvendo cálculo de equações de estados cúbicas, também se faz necessário seguir passos para resolução do modelo como apresentado na Figura 5. Tais passos podem ser descritos como:

- Se faz obrigatório a estipulação de variáveis utilizadas para o cálculo, sendo estas variáveis: n , Z_{min} , Z_{max} , R , T , P , w , P_c , T_c , a fim de gerar o gráfico do fator de compressibilidade.
- O aplicativo pode gerar gráficos apenas com a tela na vertical, facilitando a visualização visando o uso de smartphones de todos os tamanhos.

Desta maneira, o funcionamento dos aplicativos diante a utilização da API é muito semelhante, levando em consideração que ambas as aplicações terão lógica própria para requisição correta dos dados necessários antes de fazer a conexão com a API, como apresentado na Figura 6:

Figura 6 - Fluxograma da utilização da API pelas ferramentas



Fonte: Autoria própria (2022)

5 RESULTADOS E DISCUSSÕES

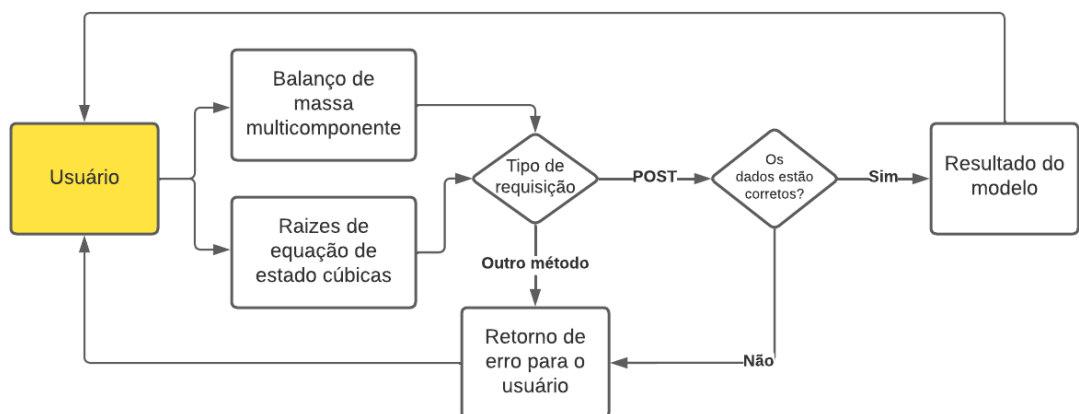
Conforme proposto, a API foi desenvolvida em Python utilizando o *framework* Flask, conseguindo atingir os objetivos desejados: a resolução de modelos matemáticos relacionados a problemas presentes na vida profissional de um engenheiro químico.

Os dois modelos matemáticos implementados como forma de exemplificação de uso da API, foram: a resolução de balanços de massa multicomponentes sem reação química e a resolução de equações de estado utilizando a fórmula de Cardano-Tartágia.

A API implementada pode ser acessada por navegadores, aplicações para testes de *request*, a exemplo, a aplicação Insomnia e linguagens de programação que possuem pacotes para obter tais requisições.

Com o auxílio do fluxograma apresentado na Figura 7, é possível visualizar o processo de utilização da API. Primeiramente o usuário escolhe qual modelo matemático gostaria de utilizar dentre os dois implementados, por meio do *end-point* do modelo. Após, o usuário deve utilizar uma requisição do tipo *POST* (utilizando linguagens de programação ou aplicações de *request* de dados web).

Figura 7 - Fluxograma da utilização da API no desenvolvimento



Fonte: A autoria própria (2022)

Então, o usuário alimenta os parâmetros requeridos pela API para resolução do modelo, como, volume, temperatura, fração molar, utilizando o modelo de dados JSON. Posteriormente, a API avalia os dados, se incompletos, ela retorna ao usuário

mensagem de erro na requisição. Caso os dados forem válidos, a API retorna ao usuário o resultado da sua requisição.

Um dos objetivos da aplicação era que qualquer pessoa poderia incluir modelos matemáticos a serem utilizados pelo usuário, gerando autonomia. Pensando nisso, para que o usuário possa visualizar quais os modelos matemáticos já abarcados pela aplicação, foi criado um *end-point* específico para visualização dos modelos matemáticos existentes dentro da API, como demonstrado na Figura 8. Desta forma, é retornado uma lista com características únicas para cada modelo em formato de JSON como: nome, descrição, *end-point* específico daquele modelo, desenvolvedor, data de implementação do modelo e um código UUID v4, sendo utilizado como forma de identificação universal para geração de códigos únicos com o intuito de identificar cada modelo usando apenas este código.

Figura 8 - Modelos matemáticos dentro da API

```
[
  {
    "Tipo": "Processos químicos",
    "Nome": "Balanço de massa para multicomponentes.",
    "Endpoint": "mass_balance",
    "Descricao": "Modelo matemático proposto para resolução de balanços de massa para multicomponentes utilizando pelo método de Broyden.",
    "Codigo": "3aa4b2ac-431a-45dc-9504-b4c4aa40b277",
    "Criador": "Vilmar Steffen",
    "Upload": "07/08/2022",
    "Funcao": ""
  },
  {
    "Tipo": "Termodinamica",
    "Nome": "Gráfico do fator de compressibilidade",
    "Endpoint": "eq_cub_st",
    "Descricao": "Por meio de variáveis termodinâmicas, este método é possível gerar o gráfico do fator de compressibilidade para um sistema de uma substância.",
    "Codigo": "81ffa5eb-eef2-41ab-95cc-fea46d62fa30",
    "Criador": "Vilmar Steffen",
    "Upload": "08/08/2022",
    "Funcao": "",
    "Artigo": "https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.17273?af=R&utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inbound&sid=researcher"
  }
]
```

Fonte: Autoria própria (2022)

Foi desenvolvido um aplicativo Android, para facilitar o acesso à API, focando apenas no desenvolvimento das interfaces visuais e apresentação dos resultados retornados. Como ilustrado nas Figura 9 e 10, foram criadas telas de Início para

apresentação da proposta do projeto e uma tela “Sobre” para explicar um pouco mais sobre os autores e o desenvolvimento.

Figura 9 - Tela inicial da aplicação Android



Fonte: Autoria própria (2022)

Figura 10 - Painel sobre os detalhes do desenvolvimento da aplicação

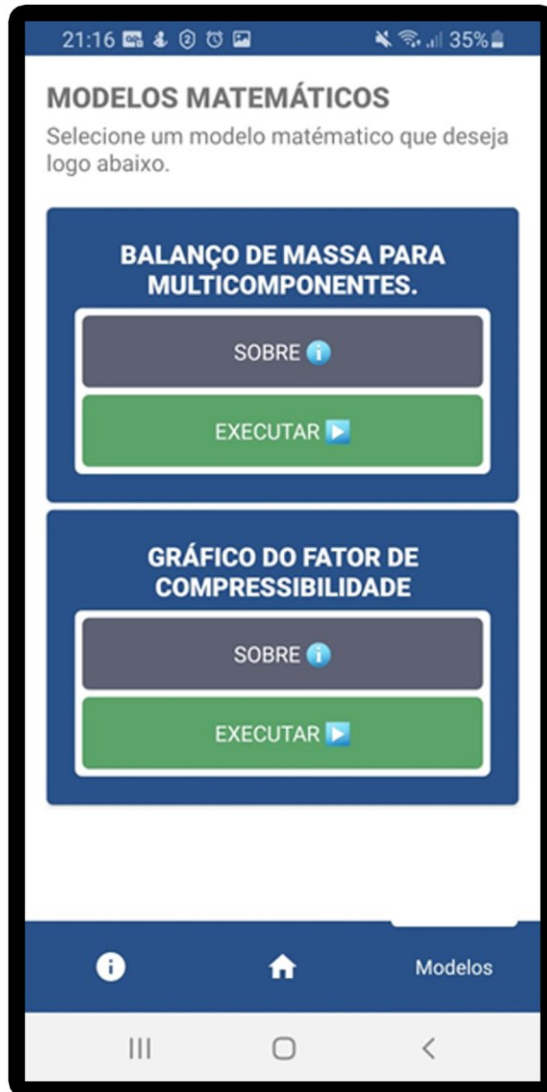


Fonte: Autoria própria (2022)

Sendo assim, é possível acessar o menu de “Modelos matemáticos” do qual é a aplicação do *end-point* para visualização dos modelos citados anteriormente. Desta maneira, para acesso aos modelos listados, é necessário que o usuário dê um *click* no item que deseja e, veja se este modelo está ou não implementado pela aplicação Android, onde foi utilizado a verificação do código UUID v4, mostrando uma clara aplicabilidade do método.

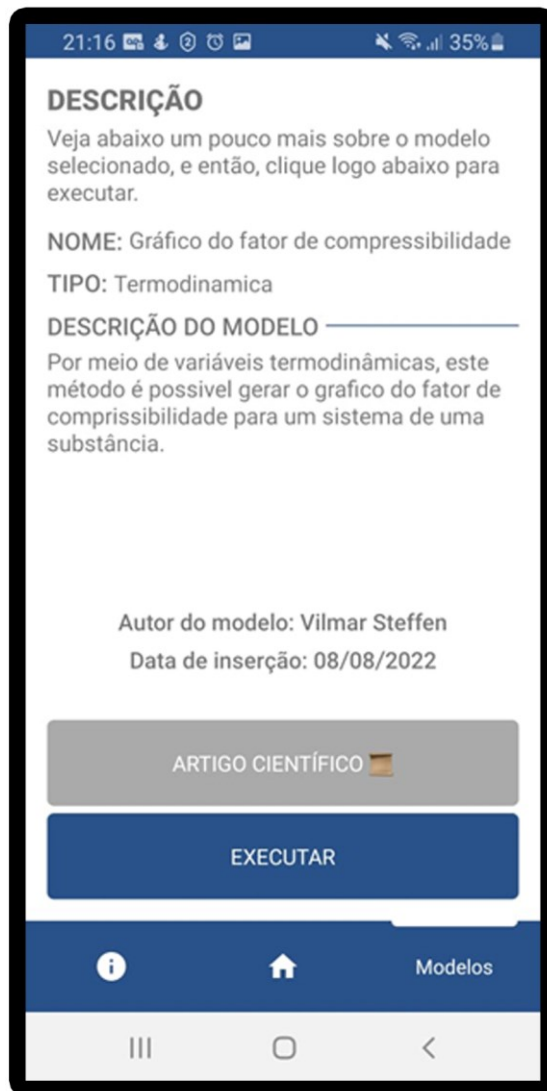
Em cada um dos itens listados, como apresentado na Figura 11, é possível que o usuário resolva o método diretamente ou acesse a tela de descrição do modelo, do qual consegue obter mais informações onde, caso disponível, pode acessar o artigo referente ao modelo matemático implementado. Ilustrado assim na Figura 12.

Figura 11 - Tela para seleção de modelos matemáticos disponíveis para resolução



Fonte: Autoria própria (2022)

Figura 12 - Tela de descrição do modelo “Gráfico do fator de compressibilidade”



Fonte: Autoria própria (2022)

Para a resolução dos modelos, foi desenvolvido uma interface para facilitação da implementação de dados dos balanços de massa multicomponentes sem reação química, do qual o usuário precisa primeiramente, informar a quantidade de entradas, saídas e componentes deste balanço.

Fornecidos os valores de entradas, saídas e números de componentes, como é possível ver na Figura 13, o aplicativo gera campos de inserção de entradas, saídas e suas respectivas composições, do qual o usuário deve respeitar as condições apresentadas anteriormente para possibilitar a resolução do modelo. Caso seja validado os dados enviados pelo usuário, a requisição é feita juntamente com a API e é apresentado o resultado do balanço para o usuário.

Figura 13 - Tela de resolução de balanços de massa multicomponentes

21:20 34%

RESOLUÇÃO DO MODELO

Conclua o processo de resolução do modelo abaixo para obtenção do resultado.

ENTRADAS

SAIDAS

COMPONENTES

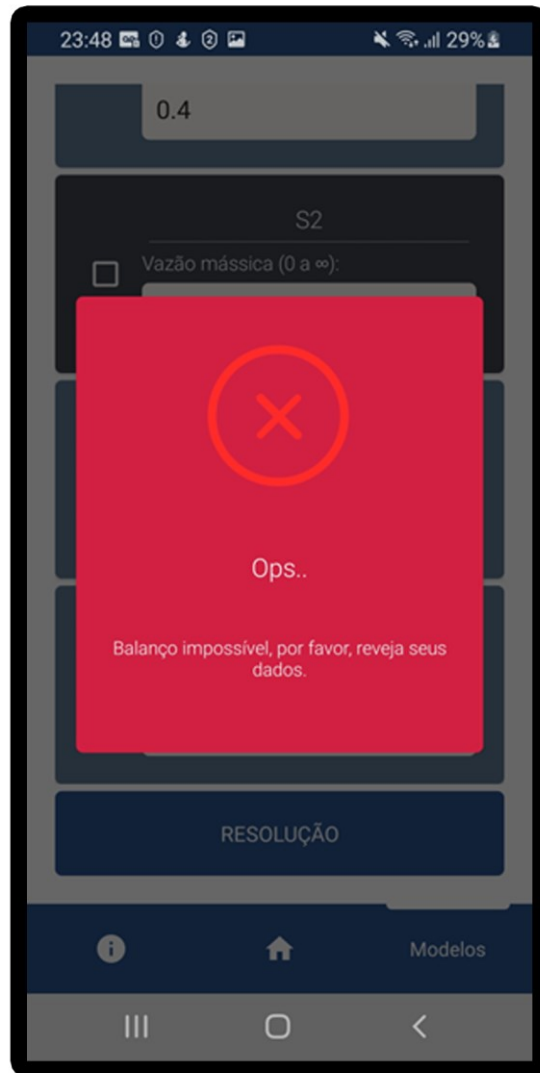
PRÓXIMO

Modelos

Fonte: Autoria própria (2022)

Como citado anteriormente, para todo o projeto foram gerados tratamentos de erro para guiar o usuário caso seja utilizado alguma informação fora dos conformes propostos pela aplicação, onde para o balanço de massa, são possíveis erros: não alcançar o número mínimo de variáveis necessárias, frações mássicas ou molares fora dos valores mínimos e máximo estipulados e o modelo proposto ser impossível de ser resolvido matematicamente.

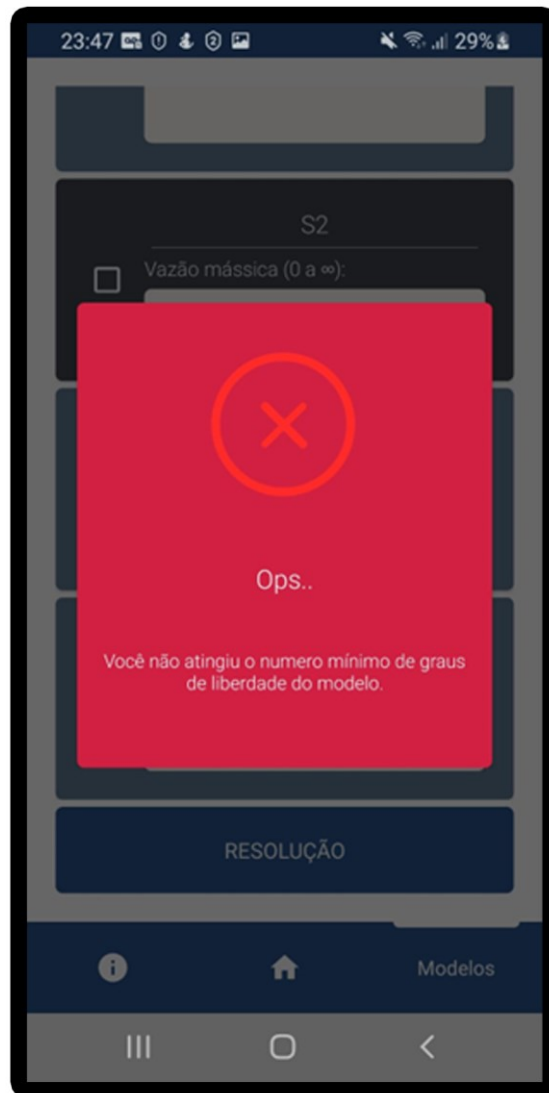
Desta forma, é apresentado na Figura 14 o erro caso o usuário insira um conjunto de valores para o balanço de massa que não é possível de resolver, sendo necessário a revisão dos dados para resolução do modelo.

Figura 14 - Erro de balanço envolvendo balanços impossíveis

Fonte: Aatoria própria (2022)

Conforme apresentado na Figura 15, este erro acontece devido ao fato de o usuário não informar o número de variáveis suficientes para resolução do modelo, sendo necessário assim, o conhecimento ou estipulação dos demais dados para atingir o valor mínimo de variáveis pra que consiga prosseguir com a resolução do modelo.

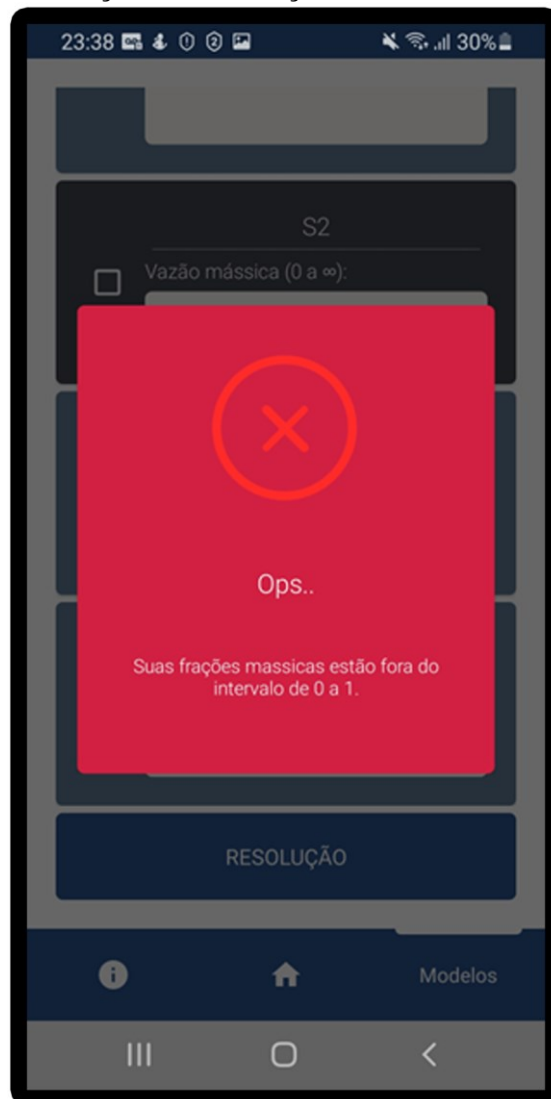
Figura 15 - Erro de balanços por não atingir o número mínimo de variáveis



Fonte: Autoria própria (2022)

Deste modo, na Figura 16 é apresentada a mensagem de erro para o caso de inserção de frações mássicas fora do intervalo de 0 e 1, sendo inconsistente fisicamente, o que torna necessária a revisão dos dados para o prosseguimento da resolução do modelo.

Figura 16 - Erro de balanço devido a frações mássicas fora do intervalo



Fonte: Autoria própria (2022)

Já para resolução das equações de estado cúbicas, seu desenvolvimento foi mais simples, dependendo apenas de variáveis dos quais o usuário pode passar em uma única tela, como ilustrado na Figura 17, e então, caso sejam parâmetros válidos para a resolução de tal modelo, é apresentado um gráfico do polinômio cúbico do fator de compressibilidade.

Desta maneira, tratamentos de erro para auxílio do usuário referente ao modelo é bastante simplório, tendo em vista que caso não seja aceito os parâmetros, é apresentado ao usuário um *card* avisando ao mesmo para correção destes dados.

Figura 17 - Telas (A) e (B) da inserção de dados para o modelo “Gráfico do fator de compressibilidade”

14:47 34%

RESOLUÇÃO DO MODELO
Insira os valores das variáveis abaixo para resolução do modelo.

PONTO MÍNIMO NO INTERVALO

PONTO MÁXIMO NO INTERVALO

NÚMERO DE PONTOS

TEMPERATURA

PRESSÃO

FATOR ACÊNTRICO

PRÓXIMO

Modelos

(A)

14:47 34%

RESOLUÇÃO DO MODELO
Insira os valores das variáveis abaixo para resolução do modelo.

PRESSÃO

FATOR ACÊNTRICO

TEMPERATURA CRÍTICA

PRESSÃO CRÍTICA

MÉTODO DE RESOLUÇÃO
VdW

PRÓXIMO

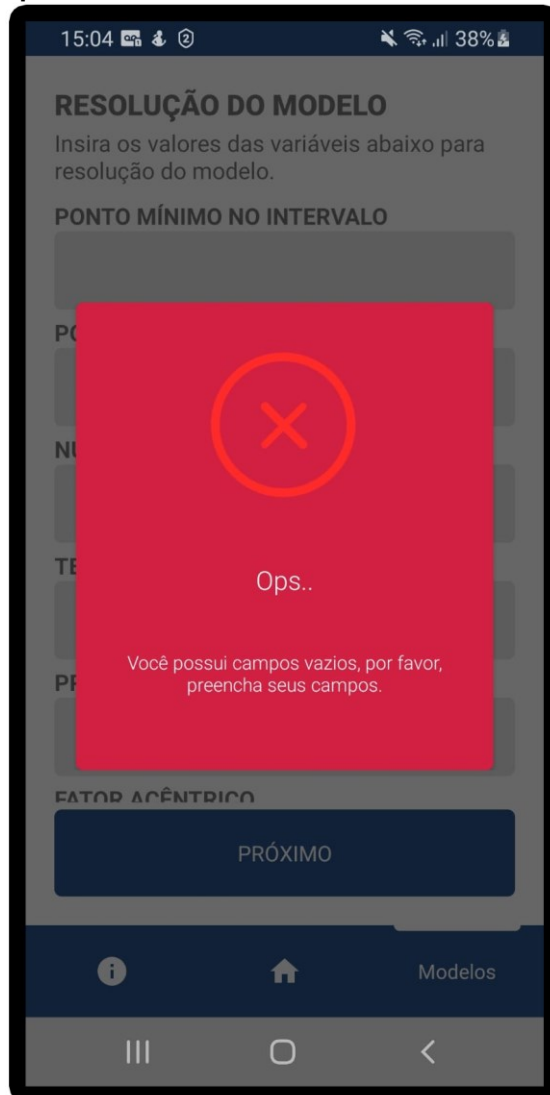
Modelos

(B)

Fonte: Autoria própria (2022)

É ilustrado na Figura 18 o erro apresentado pelo aplicativo quando o usuário não preenche todos os campos requisitados pela aplicação, o notificando e lhe exigindo revisão e inserção de tais dados.

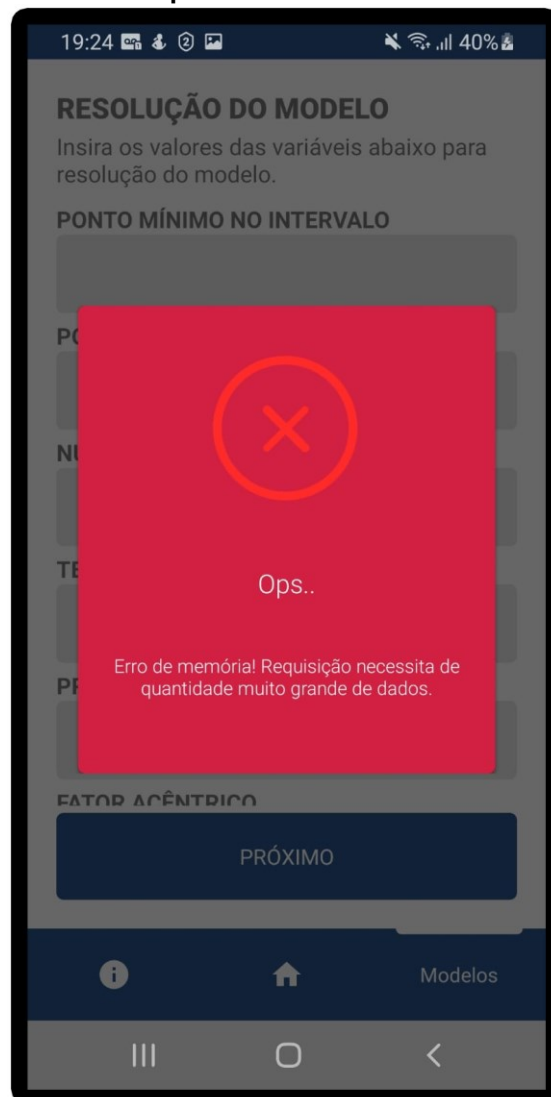
Figura 18 - Erro devido a campos vazios no modelo “Gráficos do fator de compressibilidade”



Fonte: Autoria própria (2022)

Na Figura 19, é apresentado um erro referente à requisições com grande quantidade de dados, pois, quando executada a requisição de resolução do modelo, é gerado uma grande lista JSON com os resultados referente a requisição, porém, caso sejam geradas estas grandes quantidades de dados, a API não suporta tal requisição e resulta na apresentação do erro e o pedido para revisão dos dados.

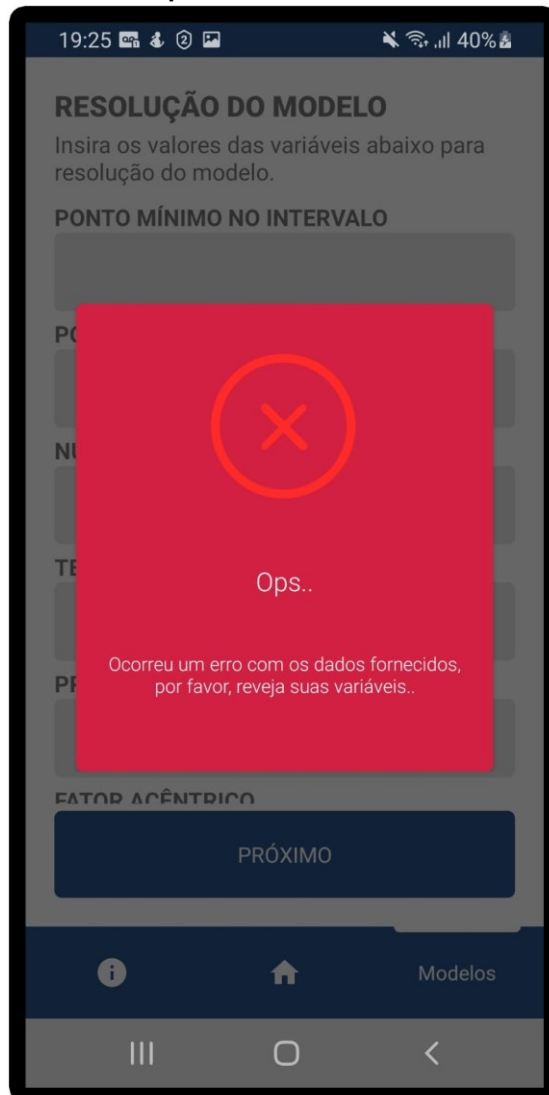
Figura 19 - Erro devido a grandes requisições no modelo “Gráfico do fator de compressibilidade”



Fonte: Autoria própria (2022)

Por fim, na Figura 20 é ilustrada uma situação de variáveis inconsistentes, que não conseguem gerar um resultado satisfatório para geração do gráfico proposto pelo modelo e novamente pede ao usuário que reveja seus dados.

Figura 20 - Erro envolvendo equação impossível no modelo “Gráfico do fator de compressibilidade”



Fonte: Autoria própria (2022)

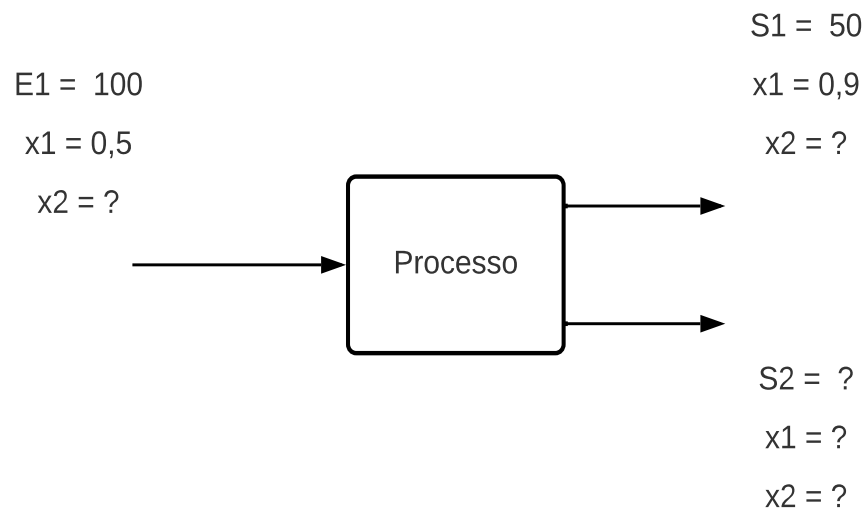
5.1 Testes da aplicação – Estudos de caso

Com o intuito de avaliar os resultados da aplicação, foram realizados quatro estudos de caso envolvendo os modelos implementados para demonstração da metodologia e resultados esperados.

5.1.1 Estudo de caso 1 – Balanço de massa em um sistema binário

O primeiro estudo de caso consiste em um processo de separação de uma mistura binária, como apresentado na Figura 21.

Figura 21 - Exemplo de balanço de massa multicomponentes simples



Fonte: Autoria própria (2022)

Desta forma, é possível extrair informações iniciais como o número de entradas, saídas e componentes como:

- O balanço possui 1 entrada, 2 saídas, ambas com 2 componentes cada;
- Para o componente 1 da entrada 1 tem-se uma fração mássica de 0,5;
- A vazão da entrada 1 é de 100 L/s.
- A vazão da saída 1 é de 50 L/s;
- Para o componente 1 da saída 1 tem-se uma fração mássica de 0,9.

As informações iniciais são transcritas para a aplicação, é necessário o *click* no botão “Próximo” para início da geração dos campos de inserção de valores das variáveis conhecidas, conforme apresentado na Figura 22.

Figura 22 - Inserção de dados no exemplo de balanço simples

The screenshot shows a mobile application interface with a dark blue header and a white background. At the top, the status bar displays the time 16:11, signal strength, Wi-Fi, airplane mode, and 39% battery. The main content area is titled "RESOLUÇÃO DO MODELO" in bold. Below the title, there is a paragraph: "Conclua o processo de resolução do modelo abaixo para obtenção do resultado." The interface is divided into three sections: "ENTRADAS" with a text input field containing the number "1"; "SAIDAS" with a text input field containing the number "2"; and "COMPONENTES" with a text input field containing the number "2". At the bottom of the main content area, there is a blue button labeled "PRÓXIMO". The bottom navigation bar is dark blue and contains three icons: an information icon, a home icon, and the text "Modelos". The very bottom of the screen shows the standard Android navigation bar with three icons: a square, a circle, and a triangle.

Fonte: Autoria própria (2022)

É necessário fornecer a(s) vazão(ões) e a(s) fração(ões) molar(es) ou mássica(s) dos componentes referentes ao balanço. Fazendo a inserção destes valores, se faz essencial realizar o click no botão “Resolução” ao final da tela para conclusão e demonstração do resultado para o usuário, como representado na Figura 23.

Figura 23 – Telas (A) e (B) das inserções dos dados conhecidos no modelo de balanço simples

Two screenshots of a mobile application interface for a simple balance model. The interface is divided into two panels, (A) and (B).

Panel (A) shows the following input fields:

- E1**: Vazão mássica (0 a ∞): 100
- xE11**: Fração mássica (0 a 1): 0.5
- xE12**: Fração mássica (0 a 1): (empty)
- S1**: Vazão mássica (0 a ∞): (empty)

Panel (B) shows the following input fields:

- S1**: Vazão mássica (0 a ∞): 50
- xS11**: Fração mássica (0 a 1): 0.9
- xS12**: Fração mássica (0 a 1): (empty)
- S2**: Vazão mássica (0 a ∞): (empty)

Both panels have a bottom navigation bar with icons for information, home, and models, and a status bar at the top showing the time and battery level.

Fonte: Autoria própria (2022)

Desta maneira é possível observar a resolução onde o *card* valores calculados ficam na cor verde. Desta forma, é observado os seguintes valores para E1 até S1, onde de acordo com a Figura 24:

- Entrada 1 se mantém inalterada com uma vazão de 100 L/s;
- Componente 1 da entrada 1 se mantém inalterada em 0,5;
- Componente 2 da entrada 2 foi calculado sendo 0,5;
- Saída 1 se mantém inalterada com uma vazão de 50 L/S.

Figura 24 - Resultado do balanço simples desde E1 até S1 e seus componentes

The screenshot displays a mobile application interface with a dark blue header showing the time 13:16 and battery level at 31%. The main content area is divided into four vertically stacked panels, each representing a different unit in a process flow:

- E1:** A dark grey panel with the title "E1" and the label "Vazão mássica (0 a ∞):". The input field contains the value "100.0".
- xE11:** A blue panel with the title "xE11" and the label "Fração mássica (0 a 1):". The input field contains the value "0.5".
- xE12:** A green panel with the title "xE12" and the label "Fração mássica (0 a 1):". The input field contains the value "0.5".
- S1:** A dark grey panel with the title "S1" and the label "Vazão mássica (0 a ∞):". The input field contains the value "50.0".

At the bottom, there is a navigation bar with three icons: an information icon, a home icon, and the text "Modelos". Below this is an Android-style navigation bar with three icons: a square, a circle, and a triangle.

Fonte: Autoria própria (2022)

Dando continuidade à resolução, é obtido para os componentes de S1 até S2 os seguintes valores, como ilustrado na Figura 25:

- Componente 1 da saída 1 se mantém inalterada 0,9;
- Componente 2 da saída 1 foi calculado sendo 0,1;
- Vazão da saída 2 foi calculada sendo 50 L/s;
- Componente 1 da saída 2 foi calculado sendo 0,1.

Figura 25 - Resultado do balanço simples para componentes de S1 até S2 e seus componentes

The screenshot displays a mobile application interface with a status bar at the top showing the time 13:16 and a battery level of 31%. The main content consists of four vertically stacked input fields, each with a title and a label:

- xS11**: Fração mássica (0 a 1): 0.9
- xS12**: Fração mássica (0 a 1): 0.1
- S2**: Vazão mássica (0 a ∞): 50.0
- xS21**: Fração mássica (0 a 1): 0.1

At the bottom, there is a navigation bar with three icons: an information icon, a home icon, and the text 'Modelos'. Below this is the standard Android navigation bar with three icons: a square, a circle, and a triangle.

Fonte: Autoria própria (2022)

Desta maneira seguindo a conclusão da resolução do modelo de exemplificação, se tem para os componentes de S2 de acordo com a Figura 26:

- Componente 2 da saída 2 foi calculado sendo 0,9.

Figura 26 - Resultado do balanço simples para componentes S2

The screenshot shows a mobile application interface with a status bar at the top displaying the time 13:17, signal strength, Wi-Fi, and 31% battery. The main content area contains four stacked input fields, each with a label and a value:

- xS12**: Fração mássica (0 a 1): 0.1
- S2**: Vazão mássica (0 a ∞): 50.0
- xS21**: Fração mássica (0 a 1): 0.1
- xS22**: Fração mássica (0 a 1): 0.9

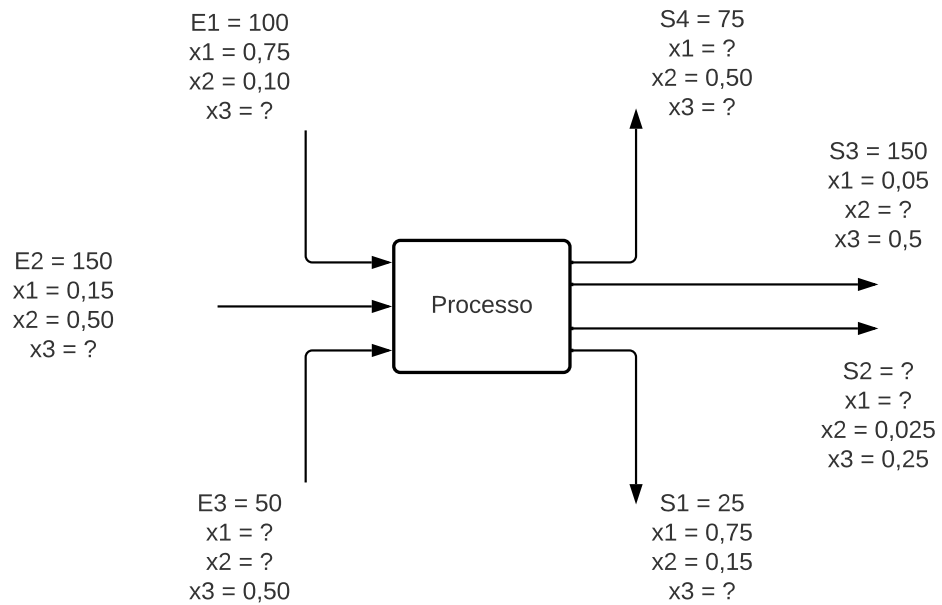
At the bottom, there is a navigation bar with three icons: an information icon, a home icon, and the text 'Modelos'. Below the navigation bar is the Android system bar with the back, home, and recents buttons.

Fonte: Autoria própria (2022)

5.1.2 Estudo de caso 2 – Balanço de massa multicomponente

É proposto também a resolução de um exemplo de complexidade maior, envolvendo maior número de entradas, saídas e componentes. Conforme ilustrado na Figura 27.

Figura 27 - Exemplo de balanço de massa multicomponentes complexo



Fonte: Autoria própria (2022)

As informações conhecidas para este estudo de caso são:

- O balanço possui 3 entradas, 4 saídas, ambas com 3 componentes cada;
- Entrada 1 com uma vazão de 100 L/s;
- Componente 1 da entrada 1 com fração mássica de 0,75;
- Componente 2 da entrada 1 com fração mássica de 0,10;
- Entrada 2 com uma vazão de 150 L/s;
- Componente 1 da entrada 2 com fração mássica de 0,15;
- Componente 2 da entrada 2 com fração mássica de 0,50;
- Entrada 3 com uma vazão de 50 L/s;
- Componente 3 da entrada 3 com uma fração mássica de 0,50;
- Saída 1 com uma vazão de 25 L/s;
- Componente 1 da saída 1 com uma fração mássica de 0,75;
- Componente 2 da saída 1 com uma fração mássica de 0,15;
- Componente 2 da saída 2 com uma fração mássica de 0,025;
- Componente 3 da saída 2 com uma fração mássica de 0,25;
- Saída 3 com uma vazão de 150 L/s;
- Componente 1 da saída 3 com uma fração mássica de 0,05;

- Componente 3 da saída 3 com uma fração mássica de 0,5;
- Saída 4 com uma vazão de 75 L/s;
- Componente 2 da saída 4 com uma fração mássica de 0,5.

Desta maneira, foram realizadas a inserção dos valores referentes as entradas, saídas e após o click no botão “Próximo” são gerados os componentes do balanço de massa. Como ilustrado na Figura 28.

Figura 28 - Inserção das entradas, saídas e componentes do exemplo de balanço de massa complexo



The screenshot shows a mobile application interface titled "RESOLUÇÃO DO MODELO". Below the title, there is a subtitle: "Conclua o processo de resolução do modelo abaixo para obtenção do resultado." The interface is divided into three sections: "ENTRADAS" with a text input field containing the number "3"; "SAIDAS" with a text input field containing the number "4"; and "COMPONENTES" with a text input field containing the number "3". At the bottom of these sections is a blue button labeled "PRÓXIMO". The top of the screen shows the time "19:12" and battery level "37%". The bottom navigation bar includes an information icon, a home icon, and the text "Modelos".

Fonte: Autoria própria (2022)

Tendo em vista a complexidade do exemplo, é necessário inserir as informações compreendidas inicialmente nos seus respectivos campos na aplicação,

sendo nesta imagem, aplicado às entradas 1 e 2 e seus respectivos componentes dos quais possuímos conhecimento. Sendo ilustrado na Figura 29.

Figura 29 – Telas (A) e (B) das inserções dos valores conhecidos de E1, E2 e seus componentes no exemplo de balanço de massa complexo

Two screenshots of a mobile application interface for mass balance calculations. Screen (A) shows input for stream E1 and its components xE11, xE12, and xE13. Screen (B) shows input for stream E2 and its components xE21, xE22, and xE23.

(A)

Nº MINIMO DE VARIÁVEIS: 18

E1

Vazão mássica (0 a ∞):
100

xE11

Fração mássica (0 a 1):
0.75

xE12

Fração mássica (0 a 1):
0.10

xE13

Fração mássica (0 a 1):
[empty]

(B)

E2

Vazão mássica (0 a ∞):
150

xE21

Fração mássica (0 a 1):
0.15

xE22

Fração mássica (0 a 1):
0.5

xE23

Fração mássica (0 a 1):
[empty]

Fonte: Autoria própria (2022)

É realizado o mesmo processo, a inclusão dos valores conhecidos é realizada, porém agora em relação a entrada 3, saída 1 e seus respectivos componentes. Sendo descrito na Figura 30.

Figura 30 - Telas (A) e (B) das inserções dos valores conhecidos de E3, S1 e seus componentes no exemplo de balanço de massa complexo

Two screenshots of a mobile application interface for mass balance calculations. Screen (A) shows input for stream E3 with a mass flow rate of 50 and mass fractions for components xE31, xE32, and xE33 (0.5). Screen (B) shows input for stream S1 with a mass flow rate of 25 and mass fractions for components xS11 (0.75) and xS12 (0.15).

(A)

(B)

Fonte: Autoria própria (2022)

Seguindo a mesma lógica, é realizado então a inserção dos dados já conhecidos referentes às saídas 2 e 3 e seus respectivos componentes. Como apresentado na Figura 31.

Figura 31 - Telas (A) e (B) das inserções dos valores conhecidos de S2, S3 e seus componentes no exemplo de balanço de massa complexo

Two screenshots of a mobile application interface for mass balance calculations. The interface is divided into two main sections, (A) and (B).

Section (A):

- S2:** Vazão mássica (0 a ∞):
- xS21:** Fração mássica (0 a 1):
- xS22:** Fração mássica (0 a 1):
- xS23:** Fração mássica (0 a 1):

Section (B):

- S3:** Vazão mássica (0 a ∞):
- xS31:** Fração mássica (0 a 1):
- xS32:** Fração mássica (0 a 1):
- xS33:** Fração mássica (0 a 1):

The bottom navigation bar contains an information icon, a home icon, and the text 'Modelos'. The Android navigation bar is visible at the very bottom.

Fonte: Autoria própria (2022)

Para finalizar a inserção de dados, os últimos dados a serem inseridos são referentes a saída 4 e seus respectivos componentes, sendo agora necessário que o usuário realize o *click* no botão “Resolução” localizado ao final da tela onde prosseguira para a realização do modelo matemático proposto. Como proposto na Figura 32.

Figura 32 – Inserção dos valores conhecidos de S4 e seus componentes no exemplo de balanço de massa complexo

The screenshot displays a mobile application interface for mass balance calculation. The screen is divided into four sections, each representing a different variable:

- S4:** A dark blue header with a white checkmark and the label "Vazão mássica (0 a ∞):". The input field contains the value "75".
- xS41:** A blue header with a white square and the label "Fração mássica (0 a 1):". The input field is empty.
- xS42:** A blue header with a white checkmark and the label "Fração mássica (0 a 1):". The input field contains the value "0.5".
- xS43:** A blue header with a white square and the label "Fração mássica (0 a 1):". The input field is empty.

The interface includes a status bar at the top showing the time "19:24", signal strength, Wi-Fi, and battery level "34%". At the bottom, there is a navigation bar with icons for information, home, and "Modelos".

Fonte: Autoria própria (2022)

Sendo assim, é ilustrado em seguinte a resolução do modelo onde, os valores determinados ficaram na cor verde. Os valores para as entradas 1 e 2 e seus respectivos componentes, onde se obteve os seguintes dados como ilustrado na Figura 33:

- Componente 3 da entrada 1 com fração mássica de 0,15;
- Componente 3 da entrada 2 com fração mássica de 0,35.

Figura 33 - Telas (A) e (B) dos resultados obtidos de E1, E2 e seus componentes no exemplo de balanço de massa complexo

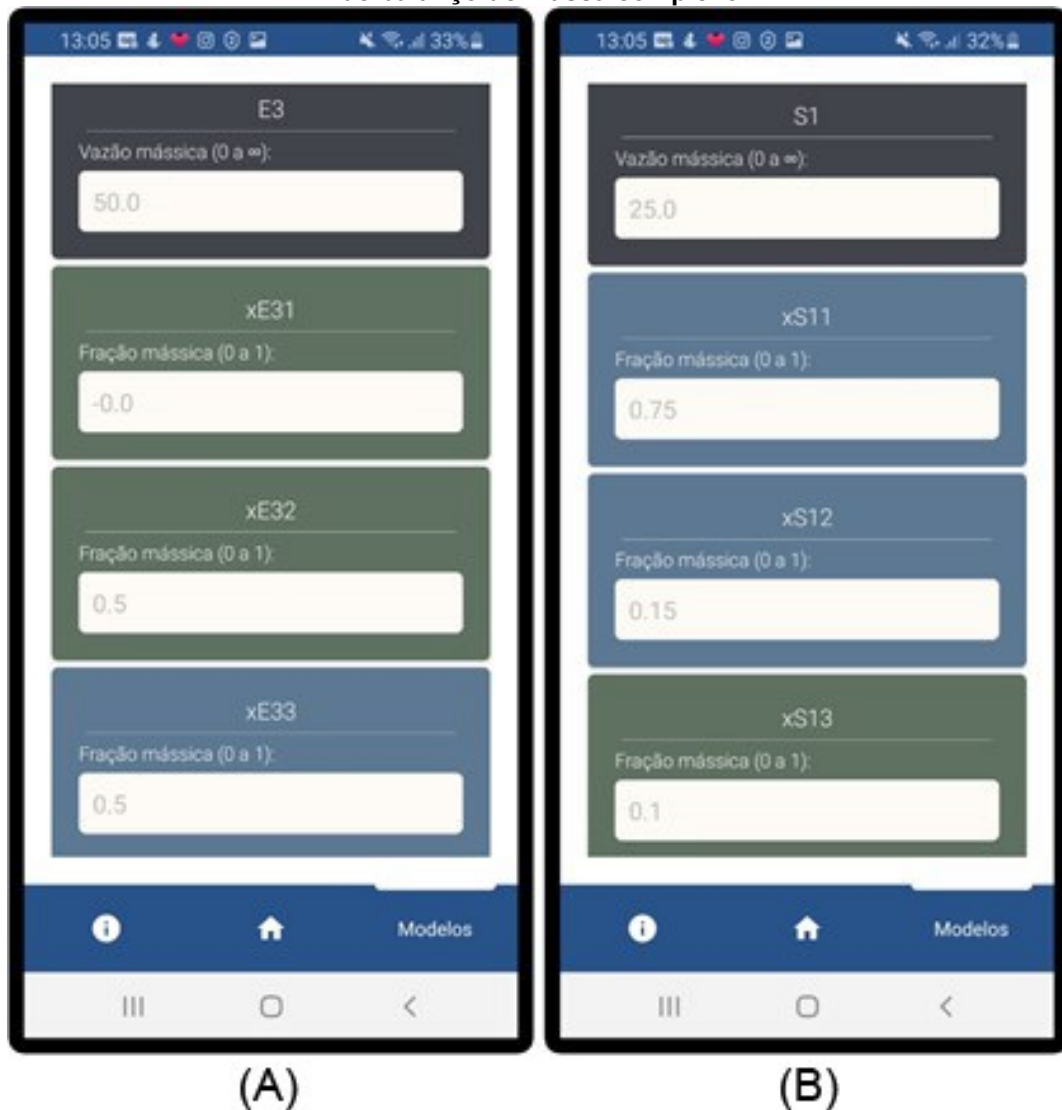


Fonte: Autoria própria (2022)

Os valores obtidos pela resolução do modelo para a entrada 3, saída 1 e seus respectivos componentes, como observado na Figura 34:

- Componente 1 da entrada 3 com fração mássica de 0;
- Componente 2 da entrada 3 com fração mássica de 0,5;
- Componente 3 da saída 1 com fração mássica de 0,1.

Figura 34 - Telas (A) e (B) dos resultados obtidos de E3, S1 e seus componentes no exemplo de balanço de massa complexo



(A)

(B)

Fonte: Autoria própria (2022)

Seguindo a resolução do modelo proposto anteriormente, é obtido os resultados relacionados a saída 2, saída 3 e seus respectivos componentes, sendo possível observar pela Figura 35 que:

- Saída 2 com vazão de 50 L/s;
- Componente 1 da saída 2 com fração mássica de 0,72;
- Componente 2 da saída 3 com fração mássica de 0,45.

Figura 35 - Telas (A) e (B) dos resultados obtidos de S2, S3 e seus componentes no exemplo de balanço de massa complexo



Fonte: Autoria própria (2022)

Seguindo a conclusão do modelo, o último componente a ser avaliado para a resolução do exemplo, sendo a saída 4 e seus respectivos componentes. Se obteve a partir da Figura 36:

- Componente 1 da saída 4 com fração mássica de 0,47;
- Componente 3 da saída 4 com fração mássica de 0,03.

Figura 36 - Resultados obtidos de S4 e seus componentes no exemplo de balanço de massa complexo

The screenshot displays a mobile application interface with a dark blue header bar showing the time 13:05 and battery level at 32%. The main content area consists of four vertically stacked input fields, each with a title and a label:

- S4**: Vazão mássica (0 a ∞): 75.0
- xS41**: Fração mássica (0 a 1): 0.47
- xS42**: Fração mássica (0 a 1): 0.5
- xS43**: Fração mássica (0 a 1): 0.03

At the bottom, there is a navigation bar with three icons: an information icon, a home icon, and a 'Modelos' label. Below this is the standard Android navigation bar with three icons: a menu icon, a home icon, and a back icon.

Fonte: Autoria própria (2022)

Sendo assim, foi observado a eficiência da aplicação para resolução de modelos simples até modelos complexos envolvendo a utilização da API como forma de resolução e aplicação Android como forma de apresentação visual dos dados.

5.1.3 Estudo de caso 3 – 3 raízes reais do fator de compressibilidade

Já para exemplificar a performance da aplicação referente à resolução de equações de estado cúbicas, é proposto a resolução do modelo conforme a Figura 37:

- Intervalo de valores do fator de compressibilidade de 0 a 1;

- Discretização dos valores do fator de compressibilidade de 1000 intervalos;
- Temperatura de 340 K;
- Pressão de 7 bar;
- Fator de compressibilidade de 0,193;
- Temperatura crítica de 425 K;
- Pressão crítica de 37,90 bar;
- Modelo de resolução SRK.

Figura 37 - Telas (A) e (B) das inserções dos dados no exemplo do modelo “Gráficos do fator de compressibilidade” para 3 raízes reais

22:28 40%

RESOLUÇÃO DO MODELO
Insira os valores das variáveis abaixo para resolução do modelo.

PONTO MÍNIMO NO INTERVALO
0

PONTO MÁXIMO NO INTERVALO
1

NÚMERO DE PONTOS
1001

TEMPERATURA
340

PRESSÃO
7

FATOR ACÊNTRICO

PRÓXIMO

Modelos

(A)

22:28 40%

RESOLUÇÃO DO MODELO
Insira os valores das variáveis abaixo para resolução do modelo.

PRESSÃO
7

FATOR ACÊNTRICO
0.193

TEMPERATURA CRÍTICA
425

PRESSÃO CRÍTICA
34.90

MÉTODO DE RESOLUÇÃO
SRK

PRÓXIMO

Modelos

(B)

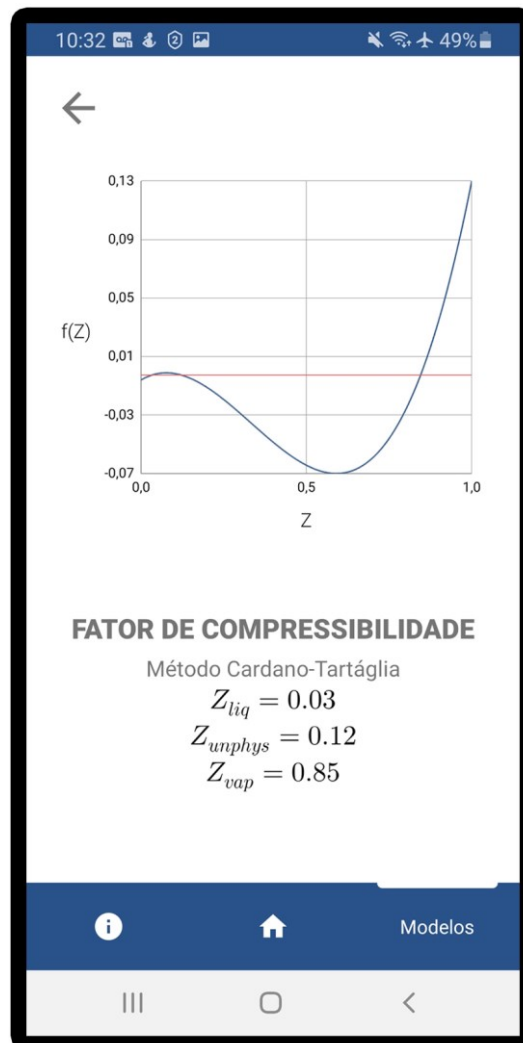
Fonte: Autoria própria (2022)

Assim, transcrevendo tais valores para a interface proposta pela aplicação, é necessário efetuar um click no botão “Próximo” ao final da tela para produção do gráfico proposto.

Após, o usuário é direcionado para a visualização do gráfico gerado pela resolução do modelo, assim como as variáveis que condizem com o problema proposto.

Com base no que é apresentado na Figura 38, é possível fazer a interpretação dos dados inseridos como ilustrado na Figura 37, onde para o conjunto de dados utilizados foram obtidas três raízes para o fator de compressibilidade, sendo 0,03 o fator de compressibilidade do líquido, 0,85 o valor do fator de compressibilidade do vapor e 0,12 a raiz que não possui significado físico

Figura 38 - Resultado do exemplo do modelo “Gráficos do fator de compressibilidade” para 3 raízes reais



Fonte: Autoria própria (2022)

5.1.4 Estudo de caso 4 – 1 raiz real do fator de compressibilidade

Outra maneira de exemplificar o uso a aplicação para a resolução de equações de estado cúbicas é a aplicação do modelo para um problema de uma única raiz, como apresentado na Figura 39 com as seguintes definições:

Figura 39 - Telas (A) e (B) das inserções dos dados no exemplo do modelo “Gráficos do fator de compressibilidade” para 1 raiz real



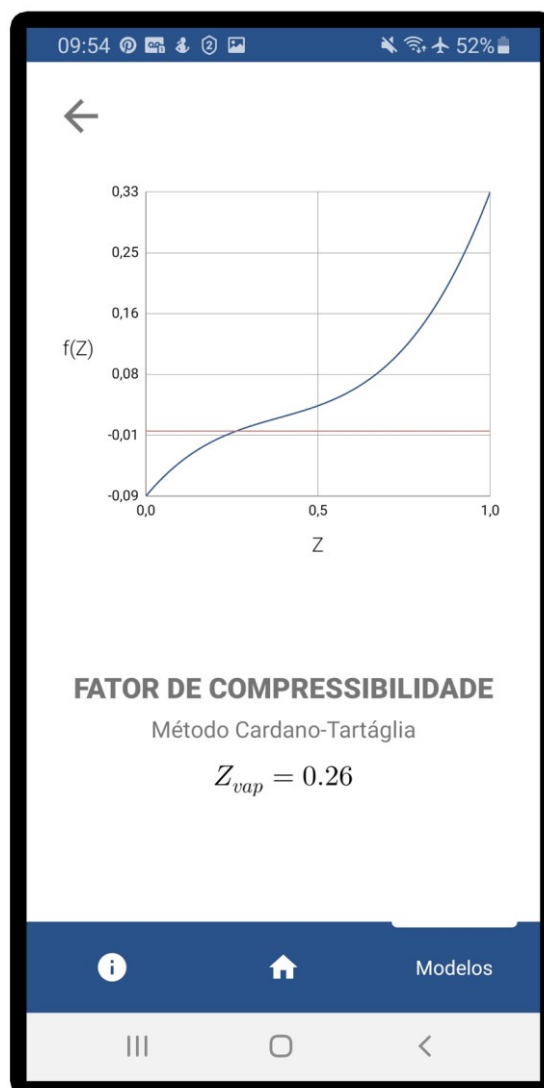
Fonte: Autoria própria (2022)

- Intervalo de valores do fator de compressibilidade de 0 a 1;
- Discretização dos valores do fator de compressibilidade de 100 intervalos;
- Temperatura de 500 K;

- Pressão de 25 bar;
- Fator de compressibilidade de 0,2;
- Temperatura crítica de 550 K;
- Pressão crítica de 22 bar;
- Modelo de resolução VdW.

Desta forma, transcrevendo os valores para a aplicação, é necessário realizar um click no botão “Próximo” ao final da tela para confecção do gráfico proposto, onde assim, o usuário é redirecionado a tela de resultado como ilustrado na Figura 40.

Figura 40 - Resultado do exemplo do modelo “Gráficos do fator de compressibilidade” para 1 raiz real



Fonte: Autoria própria (2022)

Para tais valores utilizados foi obtido apenas um raiz real para o fator de compressibilidade, sendo ela 0,26 para o fator de compressibilidade do vapor.

Com base em tais resultados, se prova novamente a fidelidade das resoluções dispostas pela API e apresentadas para o usuário pela aplicação Android.

6 CONCLUSÃO

No presente trabalho foi desenvolvido uma API visando versatilidade e portabilidade na resolução de modelos matemáticos no contexto da Engenharia Química. Tal ferramenta além de estar alinhada com a modernização da engenharia, propondo resolução de modelo matemáticos de forma cada vez mais computacional e moderna, também abre espaço para gerar de outras aplicações derivadas desta, contribuindo ainda mais para a modernização dos processos. A utilização da API mostrou ser extremamente vantajosa, facilitando na implementação, compartilhamento e resolução de modelos matemáticos implementados diretamente na aplicação, resultados estes claros quanto aos modelos utilizados como exemplo como balanços de massa para multicomponentes e equações de estado cúbicas utilizado a fórmula de Cardano-Tartaglia onde a resolução destes foram extremamente eficientes devido ao uso da aplicação. Deste modo, prova-se o potencial da aplicação de auxiliar desde o estudante da engenharia, até à docência e grandes indústrias.

REFERÊNCIAS

ABNT, Associação Brasileira de Normas Técnicas. **Norma Brasileira 6023**. 2018. Disponível em: <https://www.ufpe.br/documents/40070/1837975/ABNT+NBR+6023+2018+%281%29.pdf/3021f721-5be8-4e6d-951b-fa354dc490ed>. Acesso em: 20 mai. 2022.

BROYDEN, C. G.. A class of methods for solving nonlinear simultaneous equations. *Mathematics Of Computation*, v. 19, n. 92, p. 577-593, 1965. **American Mathematical Society (AMS)**. Disponível em: <http://dx.doi.org/10.1090/s0025-5718-1965-0198670-6>. Acesso em: 2 mai. 2022.

COELHO, Flávio Codeço. **Computação Científica com Python: uma introdução à programação para cientistas**. Petrópolis - RJ: 1ª ed., 2007.

FREITAS, Bruno de Almeida. CASTRO, Gabriel Gonçalves pessoa de. PEREIRA, Grazielly de Souza. CARVALHO, Renan Fernandes de. OLIVEIRA, Thalles Negrão de. **A Influência da Tecnologia no Futuro da Engenharia**. In: **SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA**, 16., 2019, Resende – RJ. Disponível em: <https://www.aedb.br/seget/arquivos/artigos19/31028222.pdf>. Acesso em: 15 mar. 2022.

HIMMELBLAU, D. M.; RIGGS, J. L. **Engenharia química: princípios e cálculos**. 8. ed. Rio de Janeiro, RJ: LTC, 2014. cap. 6, p.233.

JÚNIOR, João Batista da Silva. SILVA, Paulo Caetano. Análise da representação semântica de modelos de dados do formato JSON. Salvador: **Revista de Sistemas e Computação**, v. 8, n. 1, p. 196-209, jan./jun., 2018. Disponível em: <https://revistas.unifacs.br/index.php/rsc/article/view/5487>. Acesso em: 16 mai. 2022.

LUCIANO, Josué. ALVES, Wallison Joel Barberá. Padrão de Arquitetura MVC: Model-Viewcontroller. Bebedouro – SP: **Revista EPeQ Fafibe**, 3ª ed., vol. 01, 2011. Disponível em: <https://unifafibe.com.br/revistasonline/arquivos/revistaepeqfafibe/sumario/20/16112011142249.pdf>. Acesso em: 20 de mar. 2022.

MACHADO, Inácio de Araujo. ALVES, Ronaldo Ribeiro. Método de Newton. **Revista UniAraguaia**. 4ª ed., p. 30 – 45, 2013. Disponível em: <https://www.fara.edu.br/sipe/index.php/REVISTAUNIARAGUAIA/article/download/153/137>. Acesso em: 3 abr. 2022.

MENEZES, Nio Ney Coutinho. **Introdução à Programação com Python: Algoritmos e lógica de programação para iniciantes**. São Paulo: Novatec Editora, 1ª ed., 2010.

SOUZA, Elienai Alves de. **Métodos Iterativos para Problemas não Lineares**. Dissertação (Pós-graduação em Modelagem Computacional em Ciência e Tecnologia) – Universidade Federal Fluminense. Volta Redonda. 2015. Disponível em: http://www.mcct.sites.uff.br/wp-content/uploads/sites/454/2018/09/Dissertacao_17.pdf. Acesso em: 17 abr. 2022.

STEFFEN, Vilmar; SILVA, Edson Antonio. An analysis about analytical calculation of volume roots from cubic equations of state. **Aiche Journal**, v. 67, n. 7, 18 abr. 2021. Disponível em: <https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.17273>. Acesso em: 13 maio 2022.