

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS-
CAMPUS ARCOS
BACHARELADO EM ENGENHARIA MECÂNICA

Douglas Melo dos Santos

**FERRAMENTA COMPUTACIONAL PARA FINS DIDÁTICOS NAS ÁREAS DE
CIÊNCIAS TÉRMICAS E FLUIDOS**

Arcos, MG
2021

DOUGLAS MELO DOS SANTOS

**FERRAMENTA COMPUTACIONAL PARA FINS DIDÁTICOS NAS ÁREAS DE
CIÊNCIAS TÉRMICAS E FLUIDOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Mecânica do Campus Arcos do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG), como requisito parcial para aprovação na disciplina Trabalho Acadêmico Integrador X e obtenção do título de Bacharel em Engenharia Mecânica.

Orientador: Dr. Reginaldo Gonçalves Leão Junior

Coorientador: Dr. Francisco de Sousa Júnior

Arcos, MG

2021

AGRADECIMENTOS

Agradeço a Deus por me dar vida e saúde para chegar até o final dessa graduação e me manter firme diante de todos os desafios.

A todos os docentes e funcionários do IFMG Arcos, em especial os Doutores Reginaldo Gonçalves Leão Júnior que me orientou neste projeto e Francisco de Sousa Júnior que me coorientou.

Ao meu amigo Jaú por ter me ensinado desenvolvimento web e ter me ajudado em todas as dificuldades.

Aos meus pais, família, namorada e amigos por me dar todo o suporte necessário nessa jornada

MUITO OBRIGADO!

RESUMO

Este trabalho de conclusão de curso discorre sobre o desenvolvimento e criação de uma ferramenta computacional para fins didáticos nas áreas de ciências térmicas e fluidos, com o objetivo principal de servir como ferramenta confiável para obtenção de propriedades termodinâmicas de substâncias puras e de gás ideal do ar. A ideia surgiu da falta de praticidade para obter tais propriedades sem a necessidade de uso de *softwares* pagos. O conhecimento procedimental sobre o cálculo dos valores dessas propriedades pode ser valioso em um primeiro momento, porém a sua repetição é cansativa e onera no tempo disponível para o aprendizado de outros conhecimentos, sejam conceituais, factuais, procedimentais ou metacognitivos.. Pensando nisso, um *software* foi idealizado para trazer uma solução automatizada para obter as principais propriedades termodinâmicas de fluidos em uma interface simples com foco na experiência do usuário. A arquitetura do *software* chamado de *Termodesk* consiste em uma aplicação de *desktop* desenvolvida com *back end* em *Python*, responsável por todos os cálculos termodinâmicos e também pela importação das bibliotecas consagradas *Coolprop* e *PYroMat*, fundamentais para este projeto. A metodologia de desenvolvimento se baseou no *Scrum*, com *sprints* mais longas devido a equipe de desenvolvimento ser composta por apenas duas pessoas. A interface ou *front end* foi construída com ferramentas de desenvolvimento *web* (*HTML, JavaScript e CSS*), além do *framework EEL JS* que estrutura toda a comunicação da interface para o *back end*. Com a tecnologia definida, cinco funcionalidades foram desenvolvidas: propriedades de substâncias puras, propriedades do ar como gás ideal e também o cálculo e plotagem dos diagramas T-s e P-v dos principais tipos de ciclo *Rankine* ideal (Simples, com Regeneração e com Reaquecimento). Todas as funcionalidades foram exaustivamente validadas por meio das bibliografias consagradas da Termodinâmica com seus exemplos e exercícios práticos, mostrando uma excelente confiabilidade nos cálculos. A ferramenta funcionou conforme o esperado e será disponibilizada gratuitamente para uso dos alunos do *campus* Arcos, de modo que possa ser testado e possíveis melhorias sejam sugeridas para uma próxima versão. É esperado que mesmo a primeira versão do *Termodesk* contribua significativamente no ensino das disciplinas de Física II, Química Geral, Termodinâmica I e II, Mecânicas dos Fluidos e Máquinas de Fluxo e outras disciplinas do IFMG e de outras instituições de ensino. O registro do *software* será feito por meio das licenças da *Creative Commons*.

Palavras-chave: Termofluidos, Propriedades Termodinâmicas, Python, Termodesk, Software educacional

ABSTRACT

This scientific technical report expatiates about the development and creation of a computational didactic simulator for the thermofluids areas, with the main purpose to be a reliable tool for finding thermodynamic properties of pure substances and air as an ideal gas. The idea comes from the lack of practicality to find the values of such properties without need to use a paid software. The procedural knowledge to calculate the values of these properties can be valuable at first, but this repetition is tiresome and wastes the time available for learning other knowledges that can be conceptual, factual, procedural or metacognitive. With that in mind, a software was conceived to bring an automated solution to calculate the main thermodynamic properties of fluids in a simple interface focused on the user experience. The software was called Thermodesk and its architecture consists of a desktop application developed with Python backend, that is responsible for all thermodynamic calculation and the import of the well established packages Coolprop and PYroMat, essential for this project. The interface or frontend was built with web development tools (HTML, JavaScript and CSS), in addition to the EEL JS framework which structures all communication from the interface to the backend. After the technology definition, five functionalities were developed: properties of pure substances, properties of the air as an ideal gas and also the calculation and plotting of T-s and P-v diagrams of the main types of the ideal Rankine cycle (Simple, Regeneration and Reheat). All functionalities have been exhaustively validated through the reliable bibliographies of Thermodynamics using their examples and practical exercises that showed an excellent reliability in the calculations. The simulator will be launched for free to the students in the IFMG's Arcos campus, so that it can be tested and possible improvements could be suggested for a next version. The simulator will be launched for free to the students in the IFMG's Arcos campus, so that it can be tested and possible improvements could be suggested for a next version. It is expected that even the first version of Thermodesk can contribute significantly in the teaching of the disciplines Physics II, General Chemistry, Thermodynamics I and II, Fluid Mechanics, Turbomachinery and other disciplines at the IFMG and in other educational institutions. The software registration will be done under Creative Commons licenses.

Keywords: Thermofluid, Thermodynamics Properties, Python, Thermodesk, Educational Software

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama T-s - Ciclo Brayton	12
Figura 2 – Limites de fase das regiões críticas do ar	13
Figura 3 – Diagrama P-T das substâncias puras	14
Figura 4 – Ciclo <i>Rankine</i> Simples Ideal	15
Figura 5 – Ciclo <i>Rankine</i> Ideal com Reaquecimento	16
Figura 6 – Ciclo <i>Rankine</i> com reaquecimento e regeneração	17
Figura 7 – Arquitetura - <i>Termodesk</i>	18
Figura 8 – Estrutura de arquivos do aplicativo baseado no framework Eel.	20
Figura 9 – Tela inicial - <i>Termodesk</i>	20
Figura 10 – Tela-Propriedades de Substância Pura	21
Figura 11 – Tela - Propriedades do ar como gás ideal	22
Figura 12 – Tela - Ciclo <i>Rankine</i> com reaquecimento	23
Figura 13 – Resultado-Propriedades de substâncias puras	24
Figura 14 – Resultado-Propriedades do ar como gás ideal	25
Figura 15 – Resultado- Ciclo <i>Rankine</i> Simples	26
Figura 16 – Ciclo <i>Rankine</i> Simples - Diagrama T-s	26
Figura 17 – Ciclo <i>Rankine</i> Simples - Diagrama P-v	27
Figura 18 – Resultado - Ciclo <i>Rankine</i> com Reaquecimento	27
Figura 19 – Ciclo <i>Rankine</i> com Reaquecimento - Diagrama T-s	28
Figura 20 – Resultado - Ciclo <i>Rankine</i> com Regeneração	28
Figura 21 – Ciclo <i>Rankine</i> com Regeneração - Diagrama T-s	29

LISTA DE TABELAS

Tabela 1 – Comparativo - Bibliotecas	11
Tabela 2 – Parâmetros de entrada - Ciclo Rankine	22

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Justificativa	9
1.2	Objetivos	9
1.2.1	Objetivo geral	9
1.2.2	Objetivos específicos	9
1.3	Estrutura do trabalho	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	Cenário atual de <i>softwares</i> e bibliotecas como propriedades de fluidos e pseudo-fluidos	11
2.2	Propriedades de gás ideal do ar	12
2.3	Propriedades de Substância Pura	14
2.4	Ciclo <i>Rankine</i> Ideal	15
3	METODOLOGIA	18
3.1	Arquitetura do <i>software</i>	18
3.2	Metodologia de desenvolvimento - Scrum	19
3.3	Estrutura de arquivos	19
3.4	Funcionalidades do <i>Termodesk</i>	20
3.4.1	Propriedades de Substância Pura	21
3.4.2	Propriedades do ar como gás ideal	21
3.4.3	Ciclo <i>Rankine</i>	22
4	RESULTADOS E DISCUSSÕES	24
5	CONCLUSÃO	30
5.1	Sugestão para trabalhos futuros	30
5.2	Considerações Finais	31
	Referências	32

1 INTRODUÇÃO

A automação de tarefas repetitivas e onerosas é o motor da inovação, sendo o objetivo pelo qual os computadores foram criados. Porém, mesmo vivendo em uma sociedade já com um certo grau de automação ainda existem certas áreas do conhecimento que podem ser melhoradas pela tecnologia, sendo uma delas o ensino, onde as Tecnologias Digitais da Informação e Comunicação (TIDIC) ganham espaço na educação (ROSSI; SILVA; JUNIOR, 2020).

Pensando na trajetória de um estudante no ensino superior, particularmente na engenharia, alguns modelos de ensino dos conteúdos poderiam ser mais interativos. Aulas combinadas com o uso de novas tecnologias proporcionam um ambiente de aprendizagem em que os alunos podem explorar diferentes ferramentas para a resolução de problemas. A nova Diretriz curricular Nacional (DCN) para os cursos de Engenharia advoga para metodologias de ensino que privilegiem conhecimentos interdisciplinares e habilidades desenvolvidos mediante participação ativa, reflexiva dos alunos (BRASIL, 2019). Assim, as atividades educacionais direcionam-se para uma proposta e que aborde múltiplas dimensões do processo cognitivo: Lembrar, Entender, Aplicar, Analisar, Avaliar e Criar (CONKLIN, 2005).

Nesse contexto, surgem os chamados simuladores computacionais didáticos, para ajudar que o aluno tenha uma melhor experiência no processo de aprendizado. Estes são ferramentas computacionais que permitem experimentar sensorialmente um determinado fenômeno ou conceito de forma significativa e interativa, possibilitando simultaneamente um elevado nível de controle sobre a parametrização da execução de cada processo e sobre a configuração do sistema estudado (SILVA; BORGES, 2018).

Este mesmo conceito guiou o desenvolvimento da ferramenta educacional apresentada neste trabalho, que é defendida como complemento às abordagens de ensino ativas com o potencial de ser adaptável com diversos estilos de ensino e aprendizagem.

Após testes e correções de possíveis falhas, é esperado que esta ferramenta contribua para que o aluno possa reconhecer e descrever os conceitos de termofluidos na resolução de problemas; facilite a comparação entre cenários e suas diferentes formas de parametrização, conseqüentemente permitindo a identificação de semelhanças, diferenças, debilidades e oportunidades de melhoria de engenharia.

A ferramenta será distribuída de forma aberta (*open source*) e gratuita, disponibilizando também um manual de utilização das suas funcionalidades. O registro será feito por meio das licenças da *Creative Commons* e poderá ser incorporado no ensino e aprendizagem das disciplinas de Física II, Química Geral, Termodinâmica I e II, Mecânicas dos Fluidos e Máquinas de Fluxo, do campus Arcos do Instituto Federal de Minas Gerais e de outras instituições.

1.1 Justificativa

Desde o início dos anos 2000, com a expansão das políticas de fomento ao ingresso no ensino superior e também com democratização do acesso por meio do Exame Nacional do Ensino Médio houve um aumento significativo no ingresso de estudantes (RIOS; SANTOS; NASCIMENTO, 2001). Contudo, elevadas taxas de evasão principalmente nos cursos ligados às ciências exatas também caracterizaram esse período, levantando a questão de que o sistema poderia não estar preparado para lidar com esse quantitativo.

Independentemente dos fatores que contribuem para evasão, é imprescindível que a educação superior/técnica e de maioria pública busque alternativas para este fenômeno, considerando a realidade cultural e diversidade de cada região.

No contexto da engenharia, tem se tornado comum a disparidade de conhecimento entre a saída do ensino médio e a entrada no ensino superior, sendo esse choque de realidade inicial um agravante para a desistência do curso (RIOS; SANTOS; NASCIMENTO, 2001). Além disso, a alta carga de conteúdos novos e o modo arcaico de ensino intensifica o problema.

Seguindo essa linha de pensamento, o *software* proposto nesse trabalho pretende servir como ferramenta educacional especializado e inovador podendo colaborar com o aumento do desempenho acadêmico em disciplinas das áreas de Termofluidos, diminuindo a retenção e conseqüentemente a evasão logo no início do ensino das disciplinas específicas do curso de Engenharia Mecânica.

1.2 Objetivos

1.2.1 Objetivo geral

Criação de um *software de desktop* para fins didáticos nas áreas de ciências térmicas e fluidos.

1.2.2 Objetivos específicos

1. Desenvolvimento de classes de objetos pythonicos para o cálculo de propriedades de substâncias puras e do ar como gás ideal.
2. Desenvolvimento de classes de objetos pythonicos para o cálculo e análise gráfica de ciclos de potência do tipo Rankine.
3. Criação da arquitetura do *software* e estruturação do *backend*
4. Criação da interface gráfica do *software* com ferramentas de desenvolvimento *web*.
5. Teste das funcionalidades com exemplos didáticos das principais bibliografias de Termodinâmica de Mecânica dos Fluidos.
6. Distribuição do *software* na comunidade acadêmica

1.3 Estrutura do trabalho

No capítulo introdutório, é apresentado o conceito de um simulador didático computacional e como sua utilização pode impactar no aprendizado de disciplinas da engenharia contribuindo para o melhor desempenho e aprendizado do conteúdo ensinado.

No capítulo 2 é abordado sobre a disponibilidade de *softwares* que permitem, de forma robusta, trabalhar com propriedades de fluidos e pseudo-fluidos. É explicado sobre o conceito de substância pura e como suas propriedades são obtidas na bibliografia. Também é abordado sobre o ar como gás ideal e as equações que permitem o cálculo de suas propriedades. Por fim é apresentado o ciclo *Rankine* e suas variantes que foram incorporadas no *software*.

No capítulo 3, metodologia, é descrita a arquitetura do *software* e como todas suas funcionalidades foram construídas visando a automação de cada cálculo. As telas iniciais de cada funcionalidade foram apresentadas, demonstrando também como será a interação com o usuário.

No capítulo 4, os resultados são apresentadas todas as telas de saída de cada funcionalidade, onde as mesmas foram validadas com exercícios e situações encontradas na bibliografia principal. Por fim, a conclusão de todo o trabalho é apresentada no capítulo 5.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Cenário atual de *softwares* e bibliotecas como propriedades de fluidos e pseudo-fluidos

Várias bibliotecas e *softwares* existem com equações de alta acurácia para obter propriedades termofísicas dos fluidos. Atualmente a mais popular é o *REFPROP* (LEMMON; HUBER; MCLINDEN, 2013), de autoria do Instituto Nacional de Padrões e Tecnologia dos Estados Unidos (NIST).

Em um artigo relevante de (BELL; WRONSKI; QUOILIN, 2014) foi listado um comparativo ente as principais bibliotecas utilizadas, conforme a tabela a seguir:

Tabela 1 – Comparativo - Bibliotecas

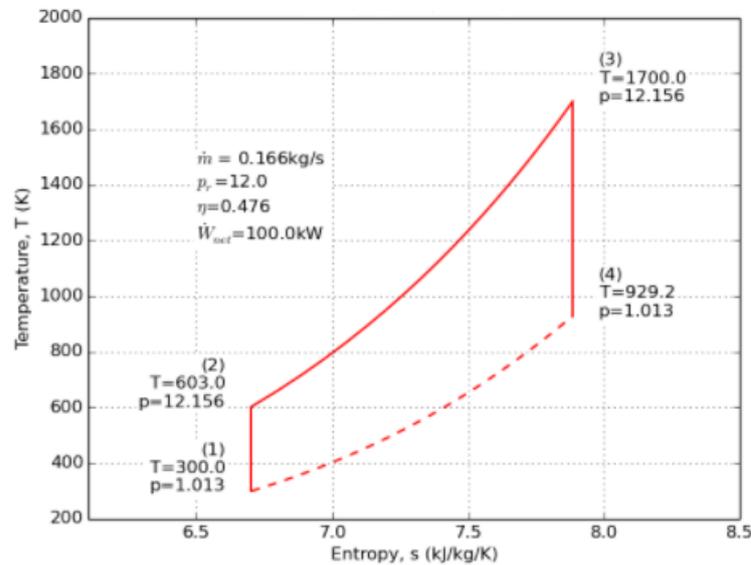
Biblioteca	Referência	Fluidos	Open-source	Misturas
REFPROP	(LEMMON; HUBER; MCLINDEN, 2013)	127	Não	Sim
CoolProp	(BELL, 2013)	110	Sim	Não
EES	(KLEIN; ALVARADO, 2000)	88	Não	Limitado
FLUIDCAL	(WAGNER, 2010)	70	Não	Não
Zittau	(KRETZSCHMAR, 2007)	34	Não	Não
FPROPS	(PYE, 2016)	36	Sim	Não
HelmholtzMedia	(SPAN et al., 2001)	9	Sim	Não

Observa-se então, que até 2014, a única biblioteca *open source* com uma quantidade relevante de fluidos era o *CoolProp*.

O cenário atual é semelhante, com destaque para a biblioteca *PYroMat* (MARTIN; RANALLI; MOORE, 2017), com um pacote de funcionalidades de propriedades termodinâmicas bem completo que facilita o processo de aprendizado dos estudantes. *PYroMat* está disponível apenas na linguagem *Python*, o que não é uma desvantagem devido a praticidade e facilidade de acesso para iniciantes dessa linguagem. Além disso outros pacotes da linguagem *Python* pode sem aproveitados, como por exemplo a biblioteca de visualizações *Matplotlib*, para a geração de visualizações de ciclos de potência a vapor e a gás.

A Figura 1, ilustra uma dessas possibilidades com a representação de um diagrama do tipo Temperatura-Entropia de um ciclo *Brayton*.

Figura 1 – Diagrama T-s - Ciclo Brayton



Fonte: (MARTIN; RANALLI; MOORE, 2017)

Com relação aos fluidos disponíveis, o *PYroMat* conta 934 substâncias, sendo 390 gases ideais, com suporte para fluidos bifásicos e misturas de gases ideais.

2.2 Propriedades de gás ideal do ar

Nos estudos e problemas que envolvem sistemas térmicos e combustão é necessário conhecer as propriedades de gás ideal do ar, as quais frequentemente são obtidas através de tabelas e interpolação. Uma solução para este problema foi apresentada nos estudos conduzidos por (LEMMON; JACOBSEN et al., 2000) com uma série de equações diferenciais com alta precisão para o cálculo das propriedades termodinâmicas do ar como gás ideal. Essas equações, definidas a seguir, são de grande relevância para automatizar a obtenção dessas propriedades.

A Equação 1 representa a energia livre de *Helmholtz*, que pode ser definida simplificada como a energia total necessária para criar o sistema, menos o calor que pode ser extraído livremente do ambiente a uma determinada temperatura.

A Equação 3 se refere à parcela de contribuição de gás ideal para a equação de *Helmholtz* e a Equação 4 se refere à contribuição residual para a equação de *Helmholtz*. O termo $\delta = \rho/\rho_j$ diz respeito à densidade reduzida, e $\tau = T_j/T$ é a temperatura reduzida recíproca. Os termos T_j e ρ_j são temperatura *maxcondentherm* e a densidade *maxcondentherm* respectivamente. Este termo “*maxcondentherm*” é utilizado no artigo para definir as propriedades no ponto de maior temperatura ao longo da linha de saturação, como mostrado na Figura 2. Os valores de N_k são dados em tabelas experimentais.

$$\alpha(\delta, \tau) = \alpha^0(\delta, \tau) + \alpha^r(\delta, \tau) \quad (1)$$

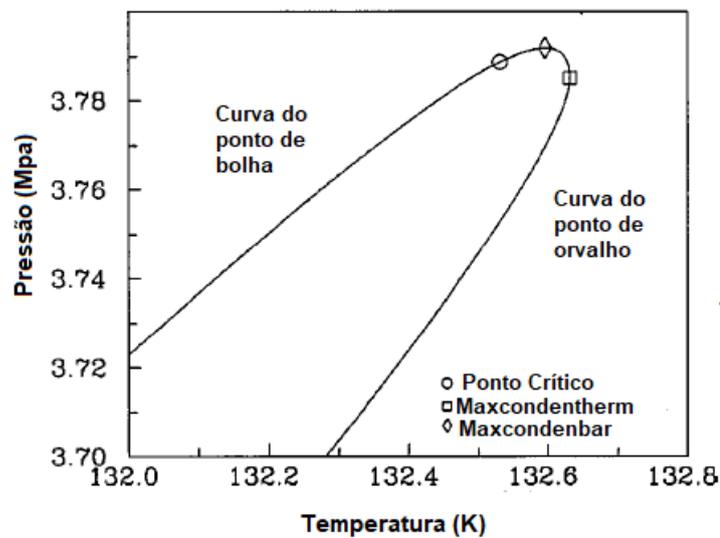
$$a^0 = RT + \sum_{i=1}^3 x_i(h_i^0 - Ts_i^0 + RT \ln X_i) \quad (2)$$

$$\alpha^0 = \ln \delta + \sum_{i=1}^5 N_i \tau^{i-4} + N_6 \tau^{1.5} + N_7 \ln \tau + N_8 \ln[1 - \exp(-N_1 \tau)] + \quad (3)$$

$$N_9 \ln[1 - \exp(-N_1 2\tau)] + N_{10} \ln[2/3 + \exp(N_1 3\tau)]$$

$$\alpha^r(\delta, \tau) = \sum_{k=1}^{10} N_k \delta^{i_k} \tau^{j_k} + \sum_{k=1}^{19} N_k \delta^{i_k} \tau^{j_k} \exp(-\delta^{l_k}) \quad (4)$$

Figura 2 – Limites de fase das regiões críticas do ar



Fonte: Adaptado (LEMMON; JACOBSEN et al., 2000)

As equações definidas anteriormente são utilizadas nas equações definidas a seguir para o cálculo das principais propriedades do ar como gás ideal. Por meio das Equações 5, 6, 7, 8, 9 e 10 são calculadas a energia interna, entalpia, entropia, energia de *Gibbs*, calor específico a volume constante e calor específico a volume constante respectivamente.

$$\frac{u}{RT} = \tau \left[\left(\frac{\partial \alpha^0}{\partial \tau} \right)_{\delta} + \left(\frac{\partial \alpha^r}{\partial \tau} \right)_{\delta} \right] \quad (5)$$

$$\frac{h}{RT} = \tau \left[\left(\frac{\partial \alpha^0}{\partial \tau} \right)_{\delta} + \left(\frac{\partial \alpha^r}{\partial \tau} \right)_{\delta} \right] + \delta \left(\frac{\partial \alpha^r}{\partial \delta} \right)_{\tau} + 1 \quad (6)$$

$$\frac{s}{R} = \tau \left[\left(\frac{\partial \alpha^0}{\partial \tau} \right)_{\delta} + \left(\frac{\partial \alpha^r}{\partial \tau} \right)_{\delta} \right] - \alpha^0 - \alpha^r \quad (7)$$

$$\frac{g}{RT} = 1 + \alpha^0 + \alpha^r + \delta \left(\frac{\partial \alpha^r}{\partial \delta} \right)_\tau \quad (8)$$

$$\frac{c_v}{R} = \tau^2 \left[\left(\frac{\partial^2 \alpha^0}{\partial \tau^2} \right)_\delta + \left(\frac{\partial^2 \alpha^r}{\partial \tau^2} \right)_\delta \right] \quad (9)$$

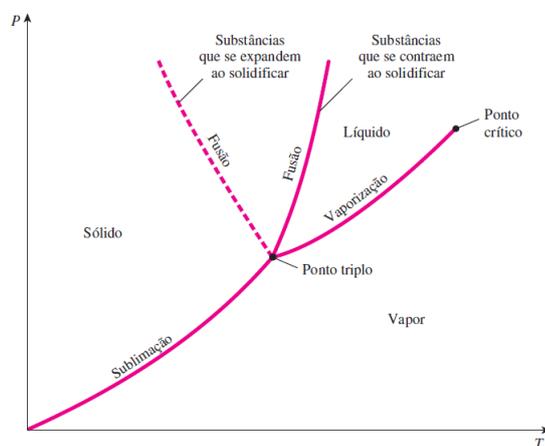
$$\frac{c_p}{R} = \frac{c_v}{R} + \frac{\left[1 + \delta \left(\frac{\partial \alpha^r}{\partial \delta} \right)_\tau - \delta \tau \left(\frac{\partial^2 \alpha^r}{\partial \delta \partial \tau} \right) \right]^2}{\left[1 + 2\delta \left(\frac{\partial \alpha^r}{\partial \delta} \right)_\tau + \delta^2 \left(\frac{\partial^2 \alpha^r}{\partial \delta^2} \right)_\tau \right]} \quad (10)$$

2.3 Propriedades de Substância Pura

De acordo com (ÇENGEL; BOLES; BUESA, 2006), uma substância pode ser classificada como pura quando tem a mesma composição química em toda a sua extensão. Mas essa definição se restringe a um único elemento químico, sendo que uma combinação de diferentes compostos pode entrar nessa classificação desde que a mistura seja homogênea em toda sua composição. Um exemplo que foge dessa definição é uma mistura "água e óleo", na qual é nítida a diferença entre as duas composições por causa da não solubilidade do óleo na água. Já o ar atmosférico que é composto de diversos gases, se enquadra na definição de substância pura, pela sua composição química uniforme.

As fases em que essas substâncias podem ser encontradas na natureza são a sólida, líquida e gasosa, havendo a possibilidade de serem encontradas combinadas em mais de uma fase, como é o caso de um copo de água com gelo, passando por uma transformação de fase. A Figura 3 a seguir retrata o diagrama P - T das substâncias puras, ilustrando como o aumento da temperatura e pressão afetam a mudança de fase.

Figura 3 – Diagrama P - T das substâncias puras



Fonte: (ÇENGEL; BOLES; BUESA, 2006)

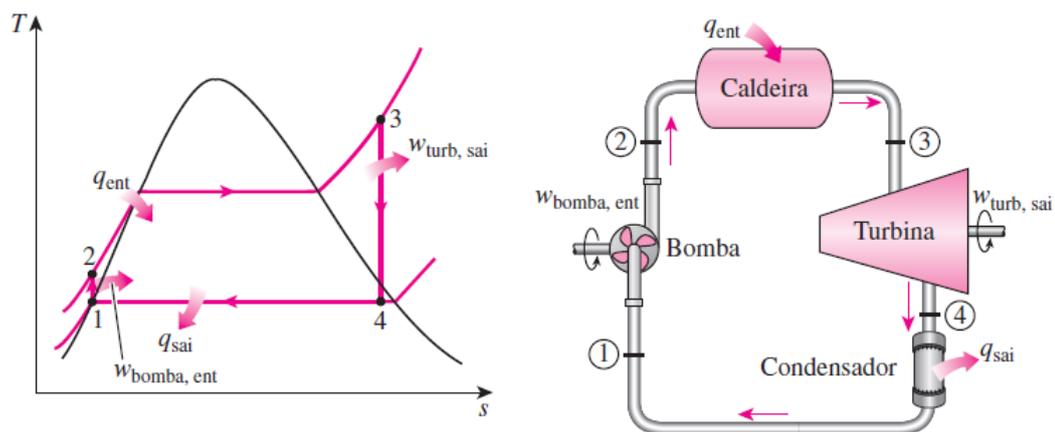
Observando Figura 3, pode-se supor a possibilidade de que uma substância pura em um processo térmico irá transitar por diferentes regiões nesse diagrama. Nessas mudanças de fase, as propriedades termodinâmicas também variam, sendo complexo de expressá-las por meio de funções simples ou relações. Com o tempo, essas propriedades foram medidas em diferentes estados, faixas de temperaturas e pressão, catalogando-as em tabelas termodinâmicas. Em seu livro *Termodinâmica*, (ÇENGEL; BOLES; BUESA, 2006) apresentam uma rica variedade dessas tabelas para as substâncias mais comuns, onde cada tabela corresponde à uma região de interesse do Diagrama $P-T$ (líquido comprimido, região de mistura e de vapor superaquecido).

2.4 Ciclo *Rankine* Ideal

O Ciclo *Rankine* Ideal é a idealização do Ciclo *Rankine* tradicional, utilizado como primeira introdução nos estudos de ciclos de potência a vapor e combinados. Esse tipo de ciclo é chamado de ideal por causa da premissa de que nenhuma irreversibilidade ocorre nos seus quatro processos definidos a seguir e mostrados na Figura 4.

1. O processo 1-2 representa a compressão isentrópica na bomba;
2. O processo 2-3 representa a entrada de calor a pressão constante em uma caldeira
3. O processo 3-4 representa a expansão isentrópica na turbina
4. O processo 4-1 representa a saída de calor a pressão constante no condensador

Figura 4 – Ciclo *Rankine* Simples Ideal

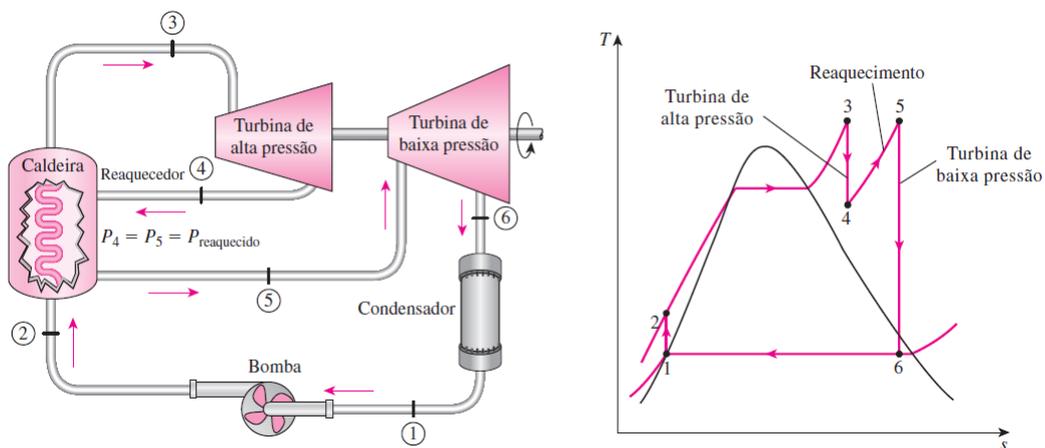


Fonte: (ÇENGEL; BOLES; BUESA, 2006)

Outras duas variações do ciclo *Rankine* ideal são o ciclo *Rankine* ideal com reaquecimento e ciclo *Rankine* ideal com regeneração. A primeira versão ocorre com a necessidade de elevar a pressão da caldeira, sem passar pelo problema de alta umidade nos últimos estágios da expansão na turbina, pelo que a eficiência térmica aumenta conforme o aumento da pressão.

Assim, no ciclo *Rankine* com reaquecimento, a expansão do vapor da turbina ocorre em dois estágios, com um reaquecimento entre eles, sendo a solução para o problema da umidade no último estágio. Esse processo de expansão ocorre em duas turbinas, uma de baixa e outra de alta pressão, conforme indicado na Figura 5.

Figura 5 – Ciclo *Rankine* Ideal com Reaquecimento



Fonte: (ÇENGEL; BOLES; BUESA, 2006)

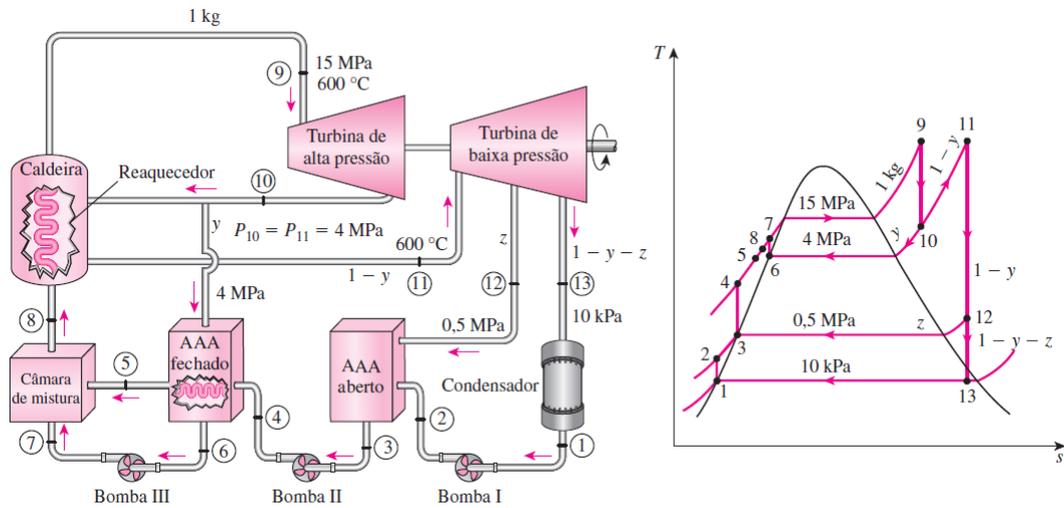
Nesse ciclo, no primeiro estágio da expansão com a turbina de alta pressão, o vapor é expandido isentropicamente até uma pressão intermediária, passando pela caldeira novamente com pressão constante. No segundo estágio na turbina de baixa pressão o vapor também se expande isentropicamente até a pressão inicial do condensador, resolvendo o problema da umidade.

Já o ciclo *Rankine* com regeneração busca aumentar a temperatura do fluido de trabalho durante o processo de fornecimento de calor, aumentando a temperatura do líquido que sai da bomba antes da entrada na caldeira. Esse método se dá pela extração do vapor da turbina em vários pontos, vapor este que não produzirá trabalho na turbina mas será usado para aquecer a água em um dispositivo chamado de regenerador, que basicamente funciona como um trocador de calor.

O regenerador pode ser do tipo aberto, no qual o vapor extraído da turbina se mistura com a água, ou do tipo fechado, sem o contato direto entre as duas correntes de fluido. Muitas usinas de potência combinam esses dois tipos dependendo da aplicação e dos custos de projeto, que podem ser combinados também com turbinas de alta e baixa pressão, utilizando do reaquecimento.

A Figura 6 ilustra o diagrama $T-s$ e também o esquema de um ciclo *Rankine* com regeneradores abertos e fechados e reaquecimento.

Figura 6 – Ciclo *Rankine* com reaquecimento e regeneração



Fonte: (ÇENGEL; BOLES; BUESA, 2006)

3 METODOLOGIA

3.1 Arquitetura do *software*

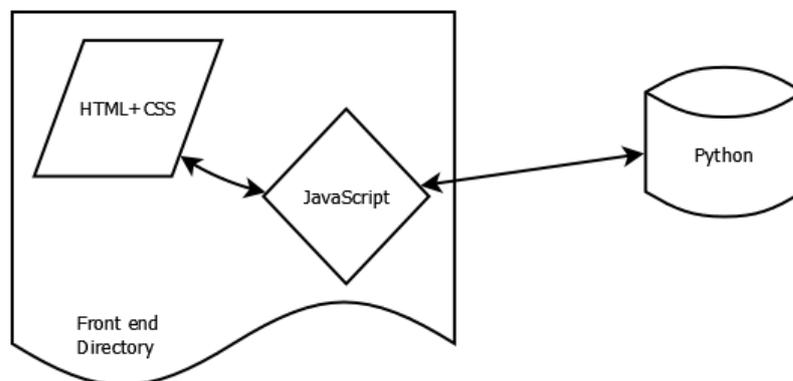
Antes do desenvolvimento de qualquer aplicação, uma POC (*Proof of concept*), prova de conceito em tradução livre, deve ser feita com o objetivo de testar a viabilidade da tecnologia escolhida (FONSECA et al., 2019). Nesse projeto a primeira abordagem para a construção do *software* foi com o *framework Electron JS*, um uma interface do usuário para acessar serviços do sistema operacional tanto via linha de comando (*CLI*) e interface gráfica (*GUI*). Com o *Electron* é possível desenvolver aplicações *desktop* usando *HTML*, *CSS* e *Javascript* e integração com *Node js* (MARINHO, 2020).

Embora as habilidades necessárias para desenvolver uma interface com o *Electron* sejam somente o conhecimento das ferramentas de desenvolvimento *web*, a comunicação entre os objetos criados no *Python* e a interface gráfica não é intuitiva, principalmente para renderização de gráficos.

Assim, optou-se por testar o *EEL*, um *framework* novo que se apresentou bastante intuitivo para executar a comunicação entre o *backend* em *Python* e a interface gráfica. O *EEL* é uma biblioteca que possibilita desenvolver aplicações *offline* usando ferramentas de desenvolvimento *web* semelhante ao *Electron*, criando um servidor na máquina local do usuário que permite que objetos criados em *Python* sejam chamados com *Javascript* e vice versa. Dessa forma, todo o potencial de processamento de dados e cálculos do *Python* pode ser combinado de forma simples com a interface.

Para entender de modo simples, a arquitetura escolhida foi desenvolvida com a linguagem *Python* no *backend* e com as tradicionais ferramentas de desenvolvimento *web*, *Java Script*, *HTML* e *CSS* no *frontend*, como indicado na Figura 7 a seguir:

Figura 7 – Arquitetura - *Termodesk*



Fonte: (DATTA, 2020)

Olhando pra Figura 7, percebe-se que todo o *front end* é criado com o *pack* de ferramentas *web*, e a interação com o *back end* em *Python* funciona com a integração com o *Eel*.

Esse *framework* que permite a criação de aplicações *offline* utilizando essa abordagem.

Nessa arquitetura, a linguagem de marcação de hipertexto *HTML* e o mecanismo de estilos *Cascading Style Sheets (CSS)* são utilizadas para criar toda a interface assim como em aplicações *web*. A linguagem *Java Script* permite que se crie toda a interatividade da aplicação, através de funções que capturam e interpretam os eventos da interface. Esses eventos após capturados, são processados nas funções criadas em *Python*, e o resultado é retornado então para a interface, com todo esse fluxo de informações entre *front* e *back end* em diferentes linguagem possibilitado pelo *EEL*.

3.2 Metodologia de desenvolvimento - Scrum

Como a equipe de desenvolvimento deste *software* é composta de apenas 2 pessoas, foi necessário utilizar uma metodologia que dividisse as tarefas em ciclos com a conclusão da entrega no tempo previsto. Não foi escolhida uma metodologia específica entre as mais conhecidas, porém a que mais se identificou com o projeto foi o *Scrum*.

Essa metodologia tem alguns termos associados ao *Rugby* e foi concebida pela primeira vez em um estudo publicado em *Harvard* em 1986 (TAKEUCHI; NONAKA, 1986). O termo *Scrum* refere-se a formação dos jogadores em uma equipe de *Hugby*, enfatizando os benefícios do trabalho em equipe. A divisão das tarefas é feita em ciclos chamados de *sprints*, termo associado ao atletismo que se refere a corridas curtas. Essas *sprints* duram entre 1 a 3 semanas, com uma lista de atividades a serem concluídas que são acompanhadas diariamente por reuniões chamadas de *daylies*. No final de cada *sprint* é feita uma revisão das tarefas entregues com o objetivo de identificar possíveis melhorias .

No desenvolvimento deste *software*, da mesma forma que é recomendado no *Scrum*, foram definidos períodos de tempo para cada tarefa, porém com a diferença que períodos mais longos foram estabelecidos, de no máximo 6 semanas.

As primeiras *sprints* foram focadas na construção das classes em *python* para a automação de cada funcionalidade. Posteriormente, foram criadas as interfaces de cada funcionalidade, inicialmente com um código de *html* simples, sendo estilizado e melhorado com *JavaScript* e *CSS*. Assim nas *sprints* seguintes, as integrações entre interface e *backend* foram executadas para utilizando o *EEL*. O projeto desde a concepção até a entrega final durou cerca de 10 meses considerando a escrita do relatório, com várias *sprints* sendo executadas.

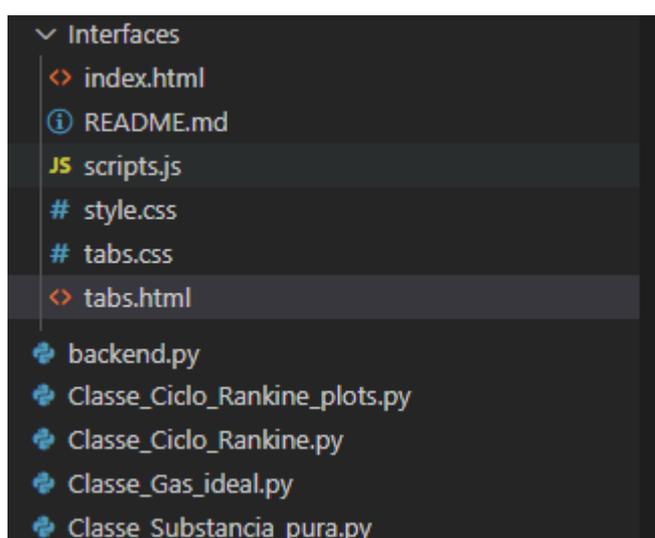
3.3 Estrutura de arquivos

A interface gráfica do aplicativo foi criada da mesma forma que um *site* tradicional. No *script HTML* principal, chamado de *index.html*, foram criados todos os elementos gráficos botões, formulários e títulos, por exemplo. No arquivo *CSS style.css*, toda a estilização e identidade visual do *software* foi construída, fontes, cores, alinhamentos. Algumas funções em *JavaScript*

foram criadas para dar uma maior interatividade a interface, como animações simples para os eventos do mouse e a transição entre os formulários.

Para o *backend*, cada funcionalidade com seus respectivos objetos foram estruturados como classes em arquivos *Python (.py)*. Dessa forma, essas classes são importadas em um arquivo principal, (*backend.py*), onde suas respectivas funções são "expostas" para o ambiente do *EEL* possibilitando a comunicação com a interface. A Figura 8 mostra a estrutura de arquivos do *software*.

Figura 8 – Estrutura de arquivos do aplicativo baseado no framework Eel.



3.4 Funcionalidades do *Termodesk*

A seção a seguir se dedica a explicar todas as 5 ferramentas do *software*, que recebeu o nome de *Termodesk*. A tela inicial é mostrada na Figura 9 a seguir, que poderá sofrer algumas alterações estéticas até o final do desenvolvimento.

Figura 9 – Tela inicial - *Termodesk*



3.4.1 Propriedades de Substância Pura

Esse módulo tem como objetivo fornecer as principais propriedades termodinâmicas de substâncias puras pela passagem de duas propriedades independentes já conhecidas. Conforme indicado na Figura 10, as entradas necessárias são os tipos das propriedades 1 e 2 com seus respectivos valores e o fluido desejado. Essas propriedades de entrada podem ser temperatura, pressão, entalpia, energia interna, entropia e título. As propriedades de saída são as mesmas, e o estado da substância.

Figura 10 – Tela-Propriedades de Substância Pura

The screenshot shows the 'Termodesk' software interface. On the left, there is a sidebar with the title 'Termodesk' and a description: 'O primeiro software didático para o estudo de disciplinas da área térmica e de fluidos.' Below this are five navigation buttons: 'Propriedades de Substância Pura' (highlighted), 'Propriedades de Gás Ideal do Ar', 'Ciclo Rankine Simples', 'Ciclo Rankine com Reaquecimento', and 'Ciclo Rankine com Regeneração'. The main content area is titled 'Propriedades de Substância Pura' and contains the following input fields:

- Propriedade 1:** A dropdown menu labeled 'Escolha' and a text input field labeled 'Valor:' with the placeholder 'Digite no SI'.
- Propriedade 2:** A dropdown menu labeled 'Escolha' and a text input field labeled 'Valor:' with the placeholder 'Digite no SI'.
- Fluido:** A dropdown menu labeled 'Escolha'.

A blue 'Calcular' button is positioned at the bottom right of the main area.

Esse módulo utiliza no *backend* a API para *Python* da biblioteca do *Coolprop* distribuída para a linguagem *Python*. Somente duas funções nativas *PropSI* e *PhaseSI* dessa biblioteca são chamadas nos cálculos. Todo o código foi desenvolvido orientado a objetos, com a vantagem de facilitar a abstração no processo de modelagem de processual e fenomenológica.

3.4.2 Propriedades do ar como gás ideal

Nessa funcionalidade, apenas de posse da temperatura e a pressão são calculados os valores da energia interna, entalpia, entropia, calor específico (pressão e volume constante) e velocidade do som do ar como gás ideal. Sem a necessidade de interpolação, todos os valores são calculados com base nas equações diferenciais parciais de (LEMMON; JACOBSEN et al., 2000) e alguns dados da biblioteca *PyroMat* (MARTIN; RANALLI; MOORE, 2017). A Figura 11 indica a tela do *software* nessa funcionalidade.

Figura 11 – Tela - Propriedades do ar como gás ideal

Termodesk

O primeiro software didático para o estudo de disciplinas da área térmica e de fluidos.

Propriedades de Substância Pura

Propriedades de Gás Ideal do Ar

Ciclo Rankine Simples

Propriedades de Gás Ideal do Ar

Pressão:
Digite no SI

Temperatura:
Digite em graus Kelvin

Calcular

O uso desse módulo será bastante útil nos cálculos de ciclos de termodinâmicos de motores de combustão interna e turbinas a gás.

3.4.3 Ciclo Rankine

Outra funcionalidade bastante completa deste *software* é o cálculo do trabalho, calor fornecido/perdido e eficiência do ciclo *Rankine* ideal. Além do ciclo simples, também é possível calcular o ciclo com reaquecimento e regeneração, com os gráficos $T-s$ (Temperatura e entropia) e $P-v$ (Pressão e volume específico).

Todo o algoritmo de resolução foi criado com base no método apresentado no livro *Termodinâmica* de cengel. Desde que as propriedades iniciais sejam fornecidas, essa funcionalidade permite o cálculo de qualquer ciclo *Rankine* ideal com as características descritas acima. A Tabela 2 a seguir mostra quais são as propriedades de entrada necessárias para o cálculo de cada tipo de ciclo *Rankine* ideal.

Tabela 2 – Parâmetros de entrada - Ciclo Rankine

Simples	Regeneração	Reaquecimento
Pressão inicial	Pressão inicial	Pressão inicial
Pressão na saída da bomba	Pressão na saída da bomba	Pressão na saída da bomba
Temperatura na caldeira	Temperatura na caldeira	Temperatura na caldeira
Fluido de trabalho	Pressão no regenerador	Pressão no reaquecedor
	Fluido de Trabalho	Fluido de Trabalho

A Figura 12 apresenta a tela inicial da funcionalidade do ciclo *Rankine* com reaquecimento.

Figura 12 – Tela - Ciclo *Rankine* com reaquecimento

Termodesk

O primeiro software didático para o estudo de disciplinas da área térmica e de fluidos.

- Propriedades de Substância Pura
- Propriedades de Gás Ideal do Ar
- Ciclo Rankine Simples
- Ciclo Rankine com Reaquecimento**
- Ciclo Rankine com Regeneração

Ciclo Rankine com Reaquecimento

Pressão inicial:
Digite no SI

Pressão na saída da bomba:
Digite no SI

Temperatura na saída da Caldeira:
Digite em graus Kelvin

Pressão no reaquecedor:
Digite no SI

Fluido
Escolha

Calcular Gerar gráfico Ts Gerar gráfico Pv

4 RESULTADOS E DISCUSSÕES

Para avaliar o desempenho do *software*, todas as funcionalidades foram testadas com resoluções de problemas encontrados no livro *Termodinâmica* (ÇENGEL; BOLES; BUESA, 2006). Para a primeira funcionalidade, propriedades de substâncias puras, o resultado da entalpia e energia interna foi padronizado com duas casas decimais, volume específico com seis e entropia com quatro, do mesmo modo que está nas tabelas do livro. Dessa forma, evita-se que o usuário seja prejudicado com erros de arredondamento. A saída do *software* para 3 estados da água é indicada na Figura 13 a seguir. O primeiro resultado foi calculado para a pressão de 100 kPa e temperatura de 300 K , Figura 13 (a). O segundo para um título de 0.5 e pressão de 2 MPa , Figura 13 (b). Já no terceiro resultado o cálculo foi feito para a temperatura de 600 K e pressão de 200 kPa , Figura 13 (c).

Figura 13 – Resultado-Propriedades de substâncias puras



(a) Liquido Comprimido

(b) Vapor Saturado



(c) Vapor Superaquecido

No campo "fluido", apenas estão listados para escolha os fluidos: amônia, água, R134A e R113, por serem mais comuns. Porém, o usuário contará com toda a lista de fluidos suportados pelo *Coolprop*, tendo que apenas digitar o código correto do fluido no campo, disponíveis na sua documentação http://coolprop.org/fluid_properties/PurePseudoPure.html

As casas decimais para o resultado da funcionalidade que calcula propriedades do ar como gás ideal foram padronizadas em três e todas as propriedades obtidas foram validadas e apresentam um erro praticamente desprezível comparado com as tabelas tradicionais. A Figura 14 indica o resultado das propriedades do ar como gás ideal para a temperatura de 300 K e pressão de 100 kPa .

Figura 14 – Resultado-Propriedades do ar como gás ideal

Propriedades:	
Temperatura	300.0 K
Pressão	100.0 kPa
Energia Interna	214.146 kJ/kg
Entalpia	300.231 kJ/kg
Entropia	1.704 kJ/kg.K
Calor Específico (volume constante)	0.717 kJ/kg.K
Calor Específico (pressão constante)	1.004 kJ/kg.K
Velocidade do som	347.229 m/s

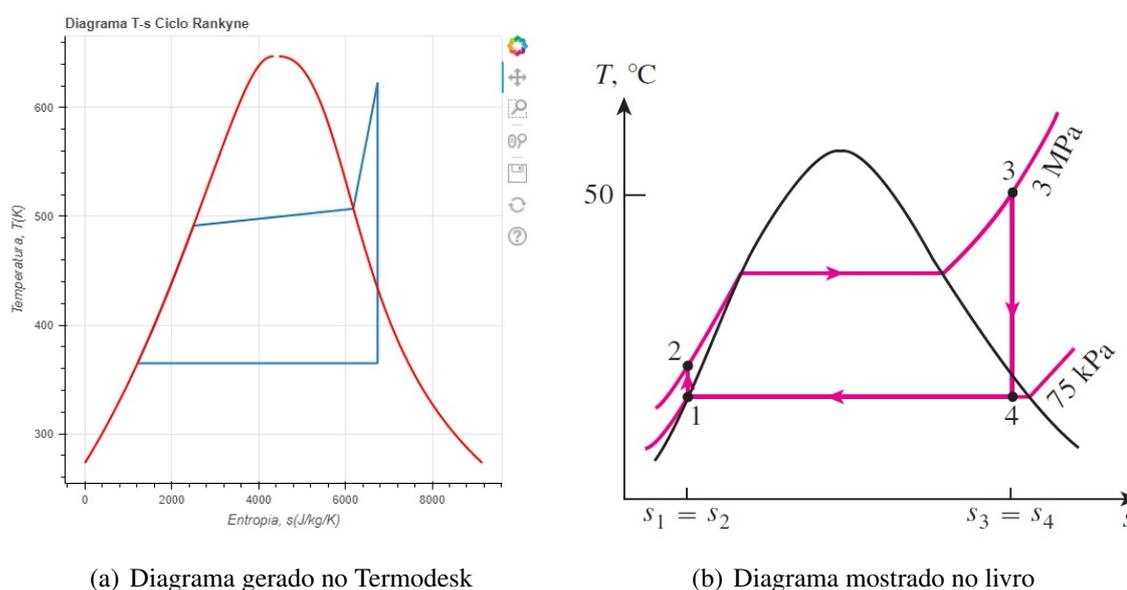
[← Voltar](#)

Para validar os cálculo de ciclo *Rankine*, foram utilizados os exemplos 10.1, 10.4 e 10.5 do capítulo de ciclos de potência a vapor e combinados do livro, além de outros exercícios do mesmo capítulo.

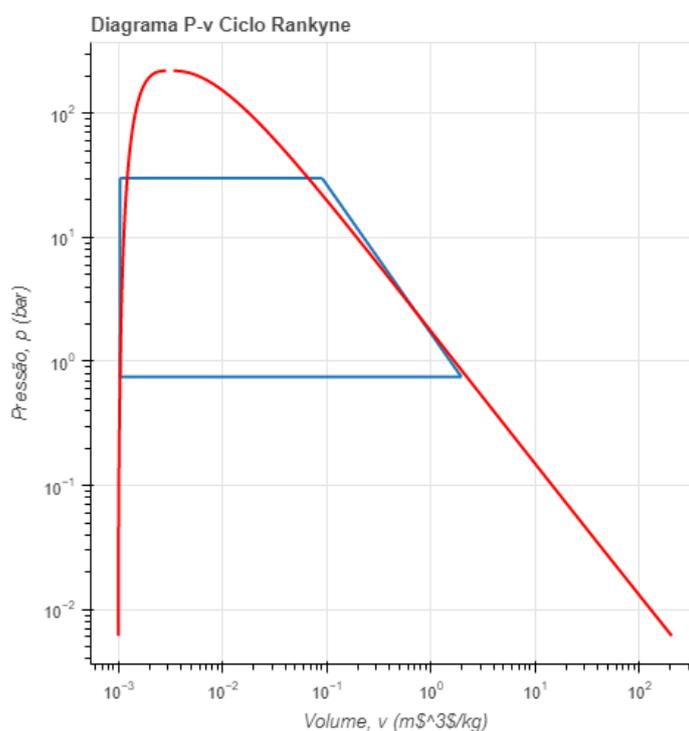
O exemplo 10.1 trata de uma usina de potência a vapor operando no ciclo *Rankine* ideal. O vapor entra na turbina com 3 MPa de pressão e $350\text{ }^{\circ}\text{C}$ de temperatura. Para esse caso a funcionalidade calcula o trabalho específico, calor perdido e fornecido em kJ/kg e eficiência. A Figura 15 a seguir indica o resultado dos cálculos para esse exemplo do livro:

Figura 15 – Resultado- Ciclo *Rankine* Simples

Também podem ser gerados os gráficos dinâmicos Temperatura-entropia (T-s) e Pressão-volume específico (P-v). A Figura 16 indica lado a lado os gráficos (T-s), plotado pelo *software*, Figura 16 (a), e o gráfico do exemplo 10.1 do livro, Figura 16 (b). Observa-se que as curvas são similares, apenas com a mudança da escala de graus *Celsius* para *Kelvin* no *software*.

Figura 16 – Ciclo *Rankine* Simples - Diagrama T-s

O Diagrama P-v do exemplo 10.1 é indicado na Figura 17 a seguir. O eixos foram colocados em escala logarítmica para melhorar visualização.

Figura 17 – Ciclo *Rankine* Simples - Diagrama P-v

Para validar os cálculos e gráficos de ciclo *Rankine* com reaquecimento o exemplo 10.4 foi utilizado. Nesse exercício são fornecidos os dados da pressão inicial de 10 kPa , pressão na caldeira de 15 MPa , temperatura na caldeira de 600°C e pressão no reaquecedor 4 MPa . Apenas com esses valores em mãos é possível calcular as principais variáveis de saída do ciclo e também o diagrama T-s.

Figura 18 – Resultado - Ciclo *Rankine* com Reaquecimento

Resultado:

Trabalho específico **1768.19 kJ/kg**

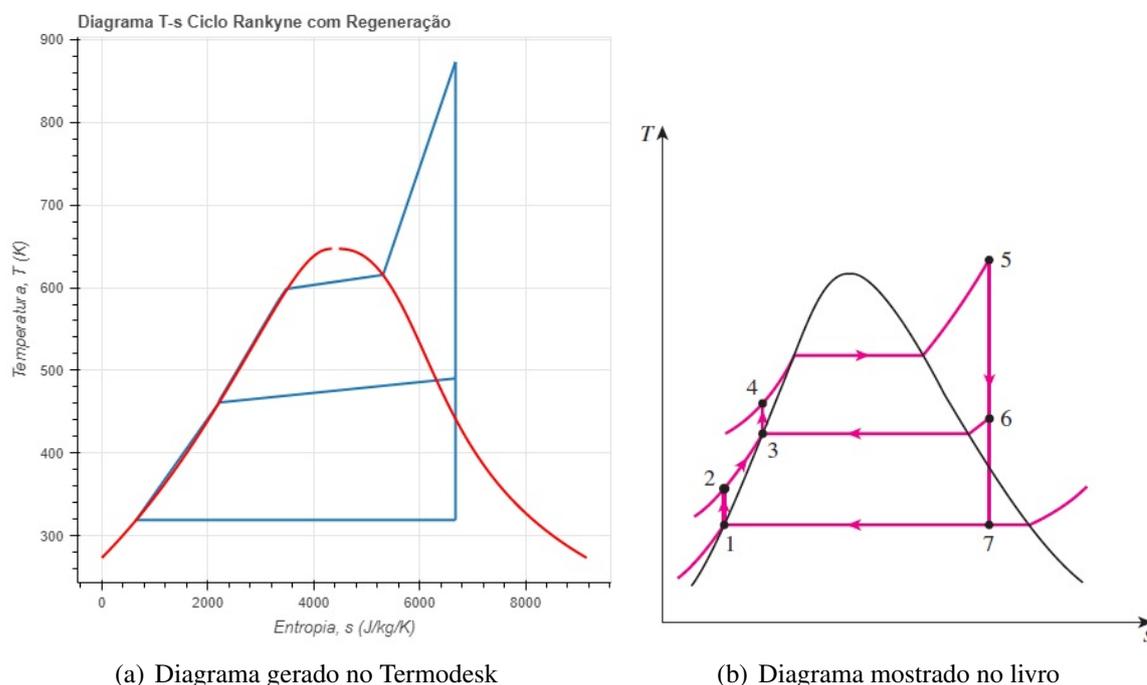
Eficiência **45.0%**

Calor Fornecido **3896.78 kJ/kg**

Calor Perdido **2143.69 kJ/kg**

[← Voltar](#)

O diagrama T-s é indicado da Figura 19 a seguir em comparação com o mesmo apresentado na resolução do exemplo do livro, com a mudança da escala de graus *Celsius* para *Kelvin* no *software*.

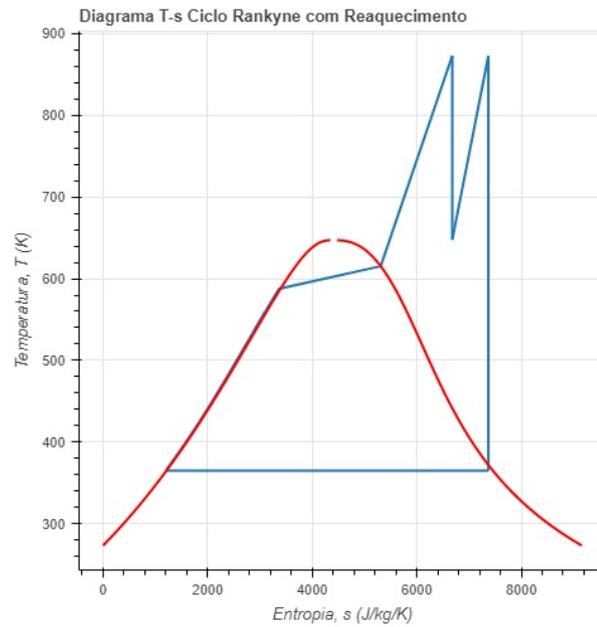
Figura 19 – Ciclo *Rankine* com Reaquecimento - Diagrama T-s

Já no exemplo 10.5, um ciclo *Rankine* regenerativo é apresentado. Os dados são quase os mesmos do exemplo anterior, pressão inicial de 10 kPa , pressão na caldeira de 15 MPa , temperatura na caldeira de 600°C , porém com pressão do regenerador de $1,2 \text{ MPa}$. Além do resultado padrão com as principais variáveis de saída do ciclo também é calculado a fração de vapor extraído da turbina de alta pressão para o regenerador, característica desse tipo de ciclo.

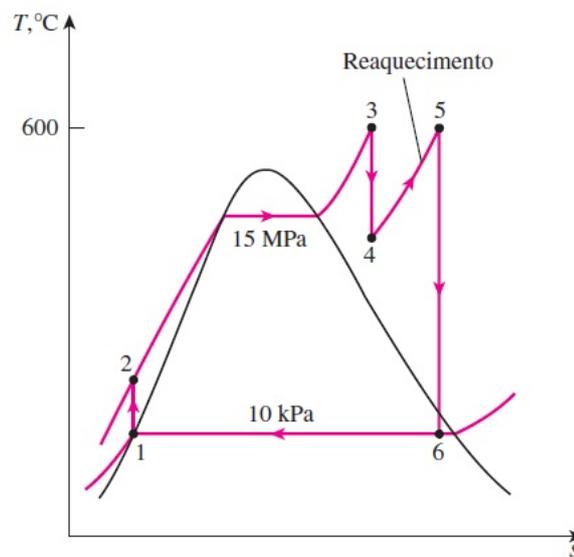
Figura 20 – Resultado - Ciclo *Rankine* com Regeneração

O diagrama T-s é mostrado da Figura 21 a seguir, também em comparação com o mesmo apresentado na resolução do exemplo do livro, com a mudança da escala de graus *Celsius* para *Kelvin* no *software*.

Figura 21 – Ciclo *Rankine* com Regeneração - Diagrama T-s



(a) Diagrama gerado no Termodesk



(b) Diagrama mostrado no livro

5 CONCLUSÃO

Com base na proposta inicial deste trabalho, todas as expectativas não somente foram cumpridas mas também superadas diante do desafio e o elevado grau de ineditismo já que até o presente momento, trata-se da primeira ferramenta nacional e gratuita com tais funcionalidades, conferindo-o elevado potencial de receptividade positiva pela comunidade acadêmica especializada na área. O *software* também é apresentado com alta confiabilidade de uso, visto que todos os cálculos foram embasados em métodos/equações tradicionais na bibliografia.

Assim, espera-se que a disponibilização gratuita da *software* traga mais dinamismo no estudo das disciplinas das áreas térmicas e de fluidos, utilizando suas ferramentas como aliado na resolução dos problemas. Há a expectativa que o *Termodesk* seja usado não só localmente, mas que se popularize na comunidade acadêmica de engenharia, o que será fundamental para a atualização de novas versões ainda mais completas.

5.1 Sugestão para trabalhos futuros

Primeiramente, as cinco funcionalidades podem ser avaliadas com relação a sua efetividade em contribuir com dinamismo e rapidez na resolução de problemas das disciplinas da área térmica e de fluidos, em especial as funcionalidades de obtenção das propriedades de substâncias puras e propriedades do ar como gás ideal, as quais podem eliminar a necessidade de consultar as tradicionais tabelas termodinâmicas, não sendo necessário fazer interpolações para se chegar no resultado esperado.

Outra possibilidade seria uma versão *mobile* já que atualmente os celulares são mais acessíveis do que computadores ou notebooks. A escolha de desenvolver o *Termodesk* para *desktop* se deu pelo fato do desenvolvimento ser mais rápido, entregando a primeira versão em um tempo menor.

Sugere-se também a adição de novas funcionalidades ao *software*, como por exemplo, balanço de energia de sistemas fechados, cálculo de ciclos de potência a gás, análise exergetica, projetos de máquinas de fluxo entre outros vários assuntos que pelo tempo disponível não foram abordados.

Por fim, com a tecnologia utilizada se mostrando confiável e prática na criação de interfaces com *backend* em *Python* para automatização de problemas de ensino da engenharia, recomenda-se que outras disciplinas possam ter seus conteúdos inseridos em ferramentas computacionais para fins didáticos semelhantes ao *Termodesk*. Essa característica da arquitetura apresentada neste trabalho se deve ao fato da linguagem *Python* contar com bibliotecas que permitem resolver problemas de diversas áreas de conhecimento. Além disso, as ferramentas de desenvolvimento *web* que foram utilizadas na construção da interface já são consagradas e contam com uma comunidade gigante de desenvolvedores.

5.2 Considerações Finais

É importante mencionar o caráter interdisciplinar deste trabalho, principalmente pelo fato de na engenharia mecânica a disciplina de *programação* ser ensinada com a abordagem bem inicial e científica, sendo necessário buscar conhecimento de diversas áreas para conseguir desenvolver o *software* por completo, como por exemplo o domínio de ferramentas de desenvolvimento *web* (*Html*, *CSS* e *JavaScript*). O conhecimento prévio em programação foi fundamental para o desenvolvimento dos cálculos automatizados pela aplicação, o que reforça a importância dessa área ser incentivada na engenharia.

O principal desafio na concepção do *software* foi a criação uma interface gráfica capaz de interagir com a linguagem *Python*, o qual foi resolvido pelo uso do *framework EEL JS*, que apesar de ter sido desenvolvido recentemente, se apresentou estável e com um potencial para aplicações com *backend* em *Python*.

REFERÊNCIAS

- BELL, Ian H.; WRONSKI, Jorrit; QUOILIN, Sylvain. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp. **Industrial & engineering chemistry research**, ACS Publications, v. 53, n. 6, p. 2498–2508, 2014.
- BELL, IH. **CoolProp: An Open-Source Thermophysical Property Library**. 2013.
- BRASIL. RESOLUÇÃO Nº 2, DE 24 DE ABRIL DE 2019. **Diário Oficial da República Federativa do Brasil**, Brasília, DF, 24 abr. 2019. Disponível em: <<https://www.in.gov.br/web/dou/-/resolu%C3%87%C3%83o-n%C2%BA-2-de-24-de-abril-de-2019-85344528>>.
- ÇENGEL, Yunus A; BOLES, Michael A; BUESA, Ignacio Apraiz. **termodinâmica**. McGraw-Hill São Paulo, 2006. v. 10.
- CONKLIN, Jack. **A taxonomy for learning, teaching, and assessing: A revision of Blooms taxonomy of educational objectives complete edition**. JSTOR, 2005.
- DATTA, Utsav. **Create HTML User Interface for Python using Eel Library**. Edição: Medium. Mai. 2020. Disponível em: <<https://utsav-datta.medium.com/create-html-user-interface-for-python-using-eel-library-bab101cc0f99>>.
- FONSECA, Stella Oggioni da et al. Mineração de dados orientada pelo domínio educacional: uma prova de conceito. **Estudos em Avaliação Educacional**, v. 30, n. 74, p. 420–453, 2019.
- KLEIN, SA; ALVARADO, FL. EES-Engineering equation solver, F-chart software. **Middleton, WI**, 2000.
- KRETZSCHMAR, Hans-Joachim. **Prof. Kretzschmar war an der Hochschule Zittau/Görlitz von 1993 bis 2018 für Technische Thermodynamik berufen**. 2007. Disponível em: <<https://www.thermo.ruhr-uni-bochum.de/thermo/wagner.html.en>>.
- LEMMON, Eric W.; HUBER, Marcia L.; MCLINDEN, Mark O. Reference fluid thermodynamic and transport properties (REFPROP). **NIST Standard Reference Database 23**, US Secretary of Commerce, 2013.
- LEMMON, Eric W.; JACOBSEN, Richard T et al. Thermodynamic properties of air and mixtures of nitrogen, argon, and oxygen from 60 to 2000 K at pressures to 2000 MPa. **Journal of physical and chemical reference data**, Physical, Chemical Properties Division, National Institute of Standards e Technology, v. 29, n. 3, p. 331–385, 2000.
- MARINHO, Thiago. **Electron - construindo aplicações desktop com JavaScript**. 2020. Disponível em: <<https://blog.rocketseat.com.br/electron-construindo-aplicacoes-desktop-com-javascript/>>.

MARTIN, Christopher Reed; RANALLI, Joseph; MOORE, Jacob Preston. Problem-based Learning Module for Teaching Thermodynamic Cycle Analysis using PYroMat. In: 2017 ASEE Annual Conference & Exposition. 2017.

PYE, John. **FPROPS**. 2016. Disponível em: <<https://ascend4.org/FPROPS>>.

RIOS, Jaime Roberto Teixeira; SANTOS, AP dos; NASCIMENTO, Caroline. Evasão e retenção no ciclo básico dos cursos de engenharia da Escola de Minas da UFOP. In: CONGRESSO Brasileiro de Ensino de Engenharia. 2001. v. 29.

ROSSI, Cláudia Maria Soares; SILVA, Jefferson Rodrigues da; JUNIOR, Niltom Vieira. **Tecnologias na educação: docência e tutoria EaD**. Instituto Federal de Minas Gerais, 2020. v. 10.

SILVA, Gabriel P; BORGES, José Antonio dos S. A Didactic Processor and Simulator for IoT. In: IEEE. 2018 3rd International Conference of the Portuguese Society for Engineering Education (CISPÉE). 2018. P. 1–7.

SPAN, R et al. Multiparameter equations of state—recent trends and future challenges. **Fluid Phase Equilibria**, Elsevier, v. 183, 2001.

TAKEUCHI, Hirotaka; NONAKA, Ikujiro. The new new product development game. **Harvard business review**, v. 64, n. 1, p. 137–146, 1986.

WAGNER, Wolfgang. **Software FLUIDCAL for Calculating Thermodynamic Properties using Fundamental Equations of State**. 2010. Disponível em: <<https://f-m.hszg.de/f-m/thermodynamik/prof-dr-ing-habil-hans-joachim-kretzschmar/prof-dr-ing-habil-hans-joachim-kretzschmar>>.

This work is licensed under a Creative Commons “Attribution-ShareAlike 4.0 International” license.

