

Pietro Sales

Desenvolvimento de aplicativo móvel para dimensionamento de ciclos Rankine

Macaé - RJ

2019

Pietro Sales

Desenvolvimento de aplicativo móvel para dimensionamento de ciclos Rankine

Projeto de graduação apresentado ao Curso de Engenharia Mecânica, da Universidade Federal do Rio de Janeiro campus - Macaé, como parte dos requisitos necessários à obtenção do título de Engenheiro Mecânico.

Universidade Federal do Rio de Janeiro – *campus* Macaé

Engenharia Mecânica

Trabalho de Conclusão de Curso

Orientador: Prof. Dr. Diego Cunha Malagueta

Coorientador: Profa. MSc. Janaína Sant'anna Gomide

Macaé - RJ

2019

CIP - Catalogação na Publicação

S163d Sales, Pietro
Desenvolvimento de aplicativo móvel para dimensionamento de ciclos Rankine / Pietro Sales. - Rio de Janeiro, 2019. 82 f.

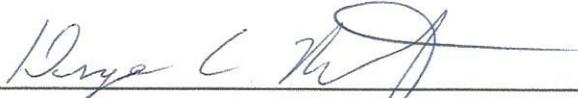
Orientador: Diego Cunha Malagueta.
Coorientadora: Janaína Sant'anna Gomide.
Trabalho de conclusão de curso (graduação) - Universidade Federal do Rio de Janeiro, Colegiado de Ensino de Graduação - Macaé, Bacharel em Engenharia Mecânica, 2019.

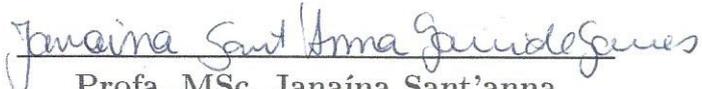
1. Rankine. 2. Dimensionamento. 3. Educação. 4. Algoritmos. 5. Aplicativo móvel. I. Cunha Malagueta, Diego, orient. II. Sant'anna Gomide, Janaína, coorient. III. Título.

Pietro Sales

Desenvolvimento de aplicativo móvel para dimensionamento de ciclos Rankine

Projeto de graduação apresentado ao Curso de Engenharia Mecânica, da Universidade Federal do Rio de Janeiro campus - Macaé, como parte dos requisitos necessários à obtenção do título de Engenheiro Mecânico.


Prof. Dr. Diego Cunha Malagueta


Profa. MSc. Janaína Sant'anna
Gomide


Profa. Dra. Raquel Jahara Lobosco


Profa. Dra. Máira Regina Rodrigues
Magini

Macaé - RJ

2019

Dedico este trabalho ao meu pai, Paulo, que sempre me deu forças por acreditar plenamente em mim; minha mãe, Ruth, que me demonstra um amor incondicional; meu avô, Agnelo, que me ensina a sempre ser feliz na vida; minha avó, Neuza, a pessoa mais carinhosa que já conheci; e minha namorada, Julia, que me faz todos os dias um homem melhor.

Agradecimentos

À UFRJ Macaé, que me proporcionou grandes experiências, amizades e um enorme crescimento pessoal durante esses cinco anos de jornada.

Aos professores que sempre buscaram ensinar muito além do que a parte acadêmica. Menções especiais para o professor Diego Malagueta e a professora Janaína Gomide, que foram muito importantes para meu interesse nas áreas deste trabalho e aceitaram o desafio de me apoiar nessa reta final da graduação. Também deixo um agradecimento especial ao professor Thiago Gomes, que mudou minha forma de o ver mundo.

Aos meus amigos da faculdade, do Magic e de várias outras esferas que frequentei nessa vida, que sempre me incentivam e acreditam que eu posso chegar cada vez mais longe.

À minha família, em especial meus pais, Ruth e Paulo, meus avós, Neuza e Agnelo, e minha companheira, Julia, que não poderiam ter feito um papel melhor em me amar, apoiar e incentivar a cada dia.

Muito Obrigado!

“It’s a long way to the top”
(AC/DC, 1975)

Resumo

Os ciclos de potência do tipo Rankine são responsáveis pela maior parte da geração de energia elétrica do mundo. Assim, para atender as necessidades financeiras e de sustentabilidade, é papel do engenheiro projetar máquinas que aproveitem a máxima quantidade de energia disponível nesses sistemas. Este trabalho teve como objetivo criar uma ferramenta para auxiliar estudantes de engenharia no dimensionamento de ciclos Rankine ideais de forma automática. Para isso, três tipos de arranjos foram modelados de forma qualitativa, identificando a sequência de passos necessária para calcular as variáveis de interesse. Os algoritmos foram implementados na linguagem de programação Python e, em seguida, em Java para o desenvolvimento de um aplicativo móvel nativo para a plataforma Android. Os softwares desenvolvidos se mostraram eficazes nos dimensionamentos propostos e capazes de indicar parâmetros para a otimização dos ciclos. Além disso, o aplicativo móvel garante um processo eficiente de experimentação para diferentes parâmetros de projeto, devido à sua portabilidade e interface intuitiva para o usuário. Por fim, os resultados alcançados por este trabalho podem ser utilizados como forma de motivação para o uso da programação na solução de problemas de engenharia.

Palavras-chaves: rankine, dimensionamento, educação, algoritmos, aplicativo móvel.

Abstract

Rankine power cycles are responsible for most of the world's electricity generation. Therefore, in order to meet financial and sustainability needs, it is the engineer's role to design machines that exploit the maximum amount of energy available in these systems. This work aimed at creating a tool to assist engineering students in modeling ideal Rankine cycles automatically. For this, three types of arrangements were implemented in the Python programming language, as well as in Java for the development of a native mobile application for the Android operational system. The software developed proved to be effective in the proposed modeling and capable of indicating parameters for the optimization of the cycles. In addition, the mobile application guarantees an efficient manner of experimentation for different design parameters, due to its portability and intuitive user interface. Finally, the results achieved by this work can be seen as a form of motivation for the use of programming in order to solve engineering problems.

Key-words: rankine, modeling, education, algorithms, mobile application.

Lista de ilustrações

Figura 1 – Sistema Cíclico	18
Figura 2 – Processo Termodinâmico	20
Figura 3 – Curva de saturação da água no diagrama T-s	22
Figura 4 – Princípio de Funcionamento de uma Máquina Térmica	25
Figura 5 – Defasagem entre ciclos reais e ideais	25
Figura 6 – Ciclo de Carnot	26
Figura 7 – Ciclo Rankine Simples	27
Figura 8 – Efeito da variação dos limites de pressão	28
Figura 9 – Ciclo Rankine Reaquecido	29
Figura 10 – Ciclo Rankine Regenerativo	30
Figura 11 – Fluxograma do Ciclo Rankine Simples	39
Figura 12 – Fluxograma do Ciclo Rankine com reaquecimento	43
Figura 13 – Fluxograma do Ciclo Rankine regenerativo	46
Figura 14 – Fluxograma de navegação do aplicativo móvel	47
Figura 15 – Entrada dos dados iniciais fornecidos pelo usuário	49
Figura 16 – Resultado do dimensionamento e indicação dos pontos de otimização	50
Figura 17 – Página do aplicativo na <i>Google Play Store</i>	51
Figura 18 – Tela inicial (à esquerda), calculadora (centro) e opções de arranjo (à direita)	52
Figura 19 – Arranjos físicos e Diagramas T-s	53
Figura 20 – Seleção de dados de entrada para os três arranjos	54
Figura 21 – Tela de resultados do ciclo Rankine reaquecido	55
Figura 22 – Otimização para o Ciclo Rankine Reaquecido	58
Figura 23 – Otimização para o Ciclo Rankine Regenerativo	59
Figura 24 – Instalação CoolProp	78

Lista de tabelas

Tabela 1 – Parâmetros de entrada Rankine simples	54
Tabela 2 – Resultado Rankine simples	55
Tabela 3 – Entrada Reaquecido e Regenerativo	56
Tabela 4 – Resultado Rankine com reaquecimento	56
Tabela 5 – Resultado Rankine regenerativo	57

Lista de símbolos

D	Densidade [kg/m ³]
g	Gravidade [m/s ²]
h	Entalpia específica [kJ/kg]
\dot{m}	Fração mássica [-]
P	Pressão [kPa]
q	Calor específico [kJ/kg]
s	Entropia específica [kJ/kg . K]
T	Temperatura [°C]
u	Energia interna [kJ/kg]
V	Velocidade [m/s]
x	Título [-]
w	Trabalho específico [kJ/kg]
Z	Altura [m]
η	Rendimento [-]
ν	Volume específico [m ³ /kg]

Sumário

1	INTRODUÇÃO	14
1.1	Contextualização do tema	14
1.2	Justificativa	15
1.3	Motivação	16
1.4	Objetivos	16
1.4.1	Objetivo geral	16
1.4.2	Objetivos específicos	16
1.5	Estrutura do trabalho	17
2	REVISÃO BIBLIOGRÁFICA	18
2.1	Fundamentos dos Ciclos Termodinâmicos	18
2.1.1	Conceitos Iniciais	18
2.1.2	Propriedades, estados e processos	19
2.1.2.1	Especificação de um Estado	20
2.1.2.2	Processos com propriedades constantes	20
2.1.2.3	Tabela de Propriedades	21
2.1.3	Curvas de saturação Líquido-Vapor	21
2.1.3.1	Título	21
2.1.4	Leis da Termodinâmica	22
2.1.4.1	Lei zero	22
2.1.4.2	Primeira lei	22
2.1.4.3	Segunda lei	23
2.1.5	Classificações	23
2.1.6	Funcionamento das máquinas térmicas	24
2.1.7	Tipos de Ciclos Ideais	24
2.1.8	Ciclo de Carnot	26
2.1.9	Ciclo Rankine simples	26
2.1.10	Ciclo Rankine com Reaquecimento	28
2.1.11	Ciclo Rankine Regenerativo	29
2.2	Softwares no processo de aprendizagem	31
2.2.1	Benefícios dos aplicativos móveis	31
2.2.2	Softwares no ensino de engenharia	32
3	METODOLOGIA	34
3.1	Considerações	34
3.2	Modelagem matemática	34

3.2.1	Balço de Energia	34
3.2.2	Dimensionamento de ciclos ideais a vapor	35
3.3	CoolProp	37
3.3.1	CoolProp em Python	37
3.4	Algoritmos para dimensionamento de Ciclos Rankine	38
3.4.1	Ciclo Rankine Simples	39
3.4.2	Implementação Python do Ciclo Rankine Simples	41
3.4.3	Ciclo Rankine com reaquecimento	42
3.4.4	Ciclo Rankine regenerativo	44
3.5	Função de otimização	45
3.6	Implementação em dispositivos Android	46
4	RESULTADOS E DISCUSSÕES	49
4.1	Aplicações em Python	49
4.2	Aplicativo Móvel	50
4.3	Validação dos resultados	53
4.3.1	Ciclo rankine simples	53
4.3.2	Ciclo Rankine com reaquecimento	54
4.3.3	Ciclo Rankine regenerativo	56
4.4	Otimização da pressão intermediária	56
5	CONCLUSÃO	60
	REFERÊNCIAS	62
	APÊNDICES	64
	APÊNDICE A – CÓDIGOS EM PYTHON	65
A.1	Ciclo Rankine Simples	65
A.2	Ciclo Rankine Reaquecido	68
A.3	Ciclo Rankine Regenerativo	72
	APÊNDICE B – FUNÇÃO DE OTIMIZAÇÃO	77
	APÊNDICE C – INSTRUÇÕES DE EXECUÇÃO DOS PROGRAMAS PYTHON NO WINDOWS	78
	APÊNDICE D – EXEMPLO DE CÓDIGO XML PARA INTERFACE DO APLICATIVO	80

1 Introdução

1.1 Contextualização do tema

A sustentação da vida moderna está fortemente ligada ao uso de energia. Quase toda atividade humana está atrelada, diretamente ou indiretamente, ao seu consumo. Isso faz com que exista uma alta correlação entre o uso de energia e o desenvolvimento econômico das sociedades (SOTOMONTE, 2015).

Qualquer previsão para o futuro aponta para um crescimento expressivo da necessidade energética global. Estima-se que, em relação aos patamares de 2010, haverá uma elevação de 20% em 2020 e 60% até 2040 no consumo (RAO, 2015).

Esse aumento acelerado desperta cada vez mais a atenção dos pesquisadores, governos e de toda população mundial acerca das consequências para o planeta. As principais preocupações são em relação à sustentabilidade do potencial futuro de geração de energia e à emissão de gases poluentes para a atmosfera.

Esses fatores, aliados a questões econômicas, resultam em uma grande necessidade de buscar formas de elevar a eficiência energética nos processos de geração de energia, reduzindo perdas para o meio ambiente sem que haja aproveitamento da energia disponível. Assim, a engenharia tem focado em formas de aumentar a eficiência dos dispositivos utilizados e na otimização de projetos, através de uma melhor seleção de fluidos de trabalho e parâmetros de funcionamento (DICKES et al., 2016).

Atualmente, os ciclos Rankine a água são responsáveis pela geração de cerca de 90% da energia elétrica no mundo. Esse tipo de máquina térmica, muito utilizada historicamente nas usinas termoelétricas, tem se tornado presente nas usinas nucleares, geotérmicas e solares (RAO, 2015). Algumas vantagens que favorecem seu uso são o baixo custo de geração de energia e a não toxicidade do fluido utilizado (SOTOMONTE, 2015).

O uso de simulações computacionais é crucial para uma boa modelagem de um ciclo de geração de potência, sobretudo quando o foco está na otimização do aproveitamento energético. No entanto, o acesso a softwares profissionais geralmente é inviável financeiramente para estudantes, pesquisadores e pequenas empresas. (DICKES et al., 2016) identificou a necessidade de desenvolver uma biblioteca de código aberto para a modelagem de ciclos Rankine orgânicos, de modo a viabilizar o acesso para a comunidade científica. De forma semelhante, (QUOILIN et al., 2014) apresenta um trabalho que busca prover uma ferramenta completa em código aberto para a modelagem de ciclos termodinâmicos.

Além disso, nesta última década, os *smartphones* têm evoluído rapidamente e alcan-

çaram uma capacidade de processamento comparável aos computadores pessoais. Este alto desempenho, aliado à sua portabilidade, teve como consequência natural o desenvolvimento de aplicações específicas para esses dispositivos (ISLAM; ISLAM; MAZUMDER, 2010). (JENG et al., 2010) apresenta os impactos positivos da utilização dos aplicativos móveis no processo de aprendizagem, visto que eles podem entregar uma experiência interativa e inovadora para os usuários.

Dado esse contexto, o presente trabalho propõe o desenvolvimento de um aplicativo móvel para a modelagem de ciclos Rankine ideais a água, incluindo funcionalidades para a otimização da escolha dos parâmetros de projeto. A aplicação tem como público alvo, sobretudo, estudantes e professores de engenharia, não havendo pretensão de competir com os resultados de softwares profissionais utilizados em projetos industriais.

1.2 Justificativa

Para alcançar bons resultados de eficiência nos processos de geração de energia, é crucial para o engenheiro a utilização de modelagens e simulações computacionais, que são capazes de avaliar diferentes tipos de arranjos e cenários.

Softwares profissionais de modelagem, em geral, possuem um custo elevado para aquisição, ficando restrito a projetos de grandes empresas. Assim, o ambiente acadêmico, repleto de estudantes e professores com alta capacidade de desenvolvimento tecnológico, geralmente fica restrito a uma baixa variedade de opções.

Com a enorme popularidade e capacidade de processamento dos *smartphones* no mundo atual, é natural que haja uma crescente importância do desenvolvimento de softwares de engenharia para essas plataformas. As vantagens do uso de aplicações móveis incluem sua portabilidade, eficiência de tempo e a familiaridade no uso do aparelho pelos usuários.

Ao levar essas ferramentas para um dispositivo tão presente no cotidiano, reduz-se a barreira de entrada para a realização de pesquisas e experimentações em qualquer ramo da engenharia. Como consequência, a probabilidade de alcançar avanços tecnológicos inovadores é fortemente elevada.

Sob essa ótica, identificou-se a relevância de desenvolver um aplicativo móvel de código aberto para servir como ferramenta no ensino de engenharia. Por desconsiderar as perdas dos sistemas reais, o produto desenvolvido não tem os mesmos objetivos que os softwares profissionais. Seus focos principais, no entanto, são a portabilidade e a velocidade para os usuários, abrindo novas possibilidades pedagógicas para alunos e professores universitários.

Ainda, o desenvolvimento deste trabalho também pode ser visto como uma forma

de motivação para a criação de outros aplicativos móveis como ferramenta de aprendizagem nos cursos de engenharia.

1.3 Motivação

A motivação para este trabalho surgiu através da experiência vivenciada pelo autor e seus colegas nos estudos da disciplina de Máquinas Térmicas, presente na grade de Engenharia Mecânica, na Universidade Federal do Rio de Janeiro *campus* Macaé.

A resolução de problemas e o dimensionamento de projetos de ciclos Rankine era limitado devido ao elevado tempo gasto na execução. Assim, o processo de aprendizagem era restringido, visto que era inviável analisar os resultados para diferentes arranjos e parâmetros iniciais.

No entanto, notou-se que o processo de resolução era determinístico e poderia ser compilado em uma sequência definida de passos. Alinhando essa possibilidade com a afinidade do autor pela área de programação computacional, foi identificada a oportunidade de criar uma solução prática para este problema.

Além disso, o fator de incentivo para o desenvolvimento do aplicativo móvel foi a possibilidade de disponibilizar uma ferramenta que pode ser aplicada efetivamente por diversos estudantes da área de Engenharia Mecânica em diferentes universidades, havendo espaço para ser aprimorada futuramente através de trabalhos colaborativos.

1.4 Objetivos

1.4.1 Objetivo geral

Desenvolver algoritmos para automatizar o processo de dimensionamento de ciclos Rankine ideais e implementá-los em um aplicativo móvel para o sistema operacional Android.

1.4.2 Objetivos específicos

- Determinar uma metodologia para o cálculo das variáveis que englobam um Ciclo Rankine;
- Automatizar a metodologia estabelecida através de linguagens de programação;
- Desenvolver um aplicativo móvel para o sistema operacional Android com uma interface de fácil utilização para os usuários;
- Avaliar os benefícios dessa ferramenta no processo de ensino de engenharia;

- Aplicar o método automatizado no processo de otimização dos ciclos.

1.5 Estrutura do trabalho

No [Capítulo 1](#) é feita uma contextualização geral do tema, além de apresentar as justificativas, motivações e objetivos do desenvolvimento deste trabalho.

No [Capítulo 2](#) é feita uma revisão bibliográfica sobre o tema, que pode ser dividida em dois tópicos principais. Primeiro, são apresentados os conceitos sobre ciclos termodinâmicos que são necessários para o entendimento e para a reprodução completa deste trabalho. Em seguida, é feito um compilado de outros trabalhos encontrados na literatura que tratam sobre o uso de softwares no ensino de engenharia, apresentando outras ferramentas com objetivos similares.

No [Capítulo 3](#) são descritas as equações matemáticas necessárias para os dimensionamentos, as ferramentas computacionais utilizadas, a modelagem de cada arranjo dos ciclos Rankine, a implementação dos algoritmos na linguagem de programação Python e a função de otimização desenvolvida. Também são estabelecidas as necessidades e o fluxograma de navegação do aplicativo móvel.

No [Capítulo 4](#) são apresentados os programas computacionais implementados, as interfaces com o usuário, a validação dos resultados e a aplicação do algoritmo no dimensionamento e otimização dos ciclos.

No [Capítulo 5](#) são feitas as considerações finais do trabalho, incluindo um compilado dos resultados alcançados, as contribuições proporcionadas por esse projeto, suas limitações e as possibilidades para trabalhos futuros.

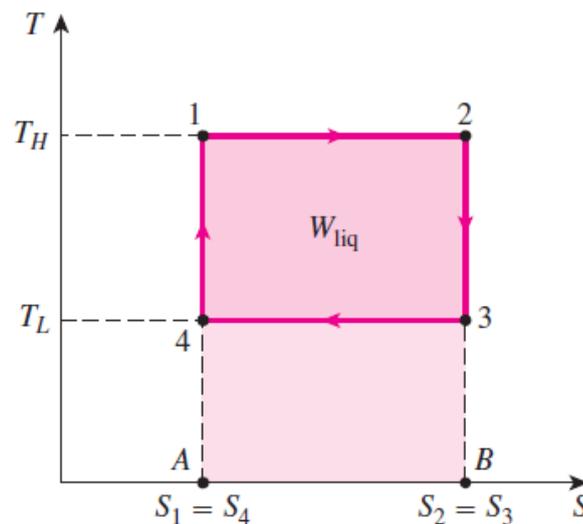
2 Revisão Bibliográfica

Neste capítulo, é feita uma revisão teórica dos principais fundamentos da termodinâmica utilizados no desenvolvimento dos algoritmos, de modo a contextualizar o leitor tanto com os conceitos necessários na elaboração deste trabalho quanto com a utilização e interpretação dos resultados do software implementado. Também é feita uma revisão de literatura sobre os benefícios da utilização de ferramentas computacionais, sobretudo com telefones móveis, no processo educacional na engenharia e sobre outras iniciativas de softwares com esse mesmo objetivo.

2.1 Fundamentos dos Ciclos Termodinâmicos

Um Ciclo Termodinâmico é uma sucessão de transformações termodinâmicas em um sistema, cujo estado final coincide com o estado inicial. No diagrama T-s (temperatura por entropia) da Figura 1, é possível observar a representação gráfica de um sistema cíclico composto por quatro estados, identificados pelos números 1, 2, 3 e 4. Ainda, a trajetória dos processos que compõe esse ciclo está representada pelo sentido das setas entre cada um destes estados.

Figura 1 – Sistema Cíclico



Fonte: (BOLES; ÇENGEL, 2013)

2.1.1 Conceitos Iniciais

A termodinâmica é a ciência que estuda a energia (BOLES; ÇENGEL, 2013). Na engenharia, o interesse geralmente é voltado para o estudo dos sistemas e suas interações

com o ambiente vizinho. Essas interações ocorrem nas formas de calor e trabalho¹, podendo ocorrer no sentido do sistema para a vizinhança ou o inverso.

Para um melhor entendimento deste trabalho, alguns termos frequentemente utilizados na termodinâmica estão descritos de forma resumida abaixo (MORAN; SHAPIRO; BOETTNER, 2013):

- Sistema: Uma região no espaço selecionada como objeto de análise.
- Vizinhança: Toda região externa ao sistema escolhido.
- Fronteira: Transição entre o sistema e a vizinhança. Pode ser físico (paredes de um recipiente) ou apenas imaginário.
- Estado: Condição de um sistema em equilíbrio.
- Propriedade²: Características de um estado do sistema, como temperatura, pressão, entalpia, entropia, volume específico e energia interna.
- Processo: Mudança sofrida por um sistema, levando-o de um estado de equilíbrio para outro.
- Percurso: Caminho percorrido em um processo.
- Ciclo: Sucessão de processos cujo estado final coincide com o estado inicial.
- Reservatório: Corpo com grande capacidade de energia térmica, capaz de fornecer ou receber calor sem alteração em sua temperatura.
- Calor: Energia térmica em transição devido à diferença de temperatura entre dois sistemas.
- Trabalho: Forma de energia transferida pela aplicação de uma força ao longo de um deslocamento.
- Eficiência: Razão entre o trabalho líquido realizado por uma máquina térmica e o calor de entrada fornecido.

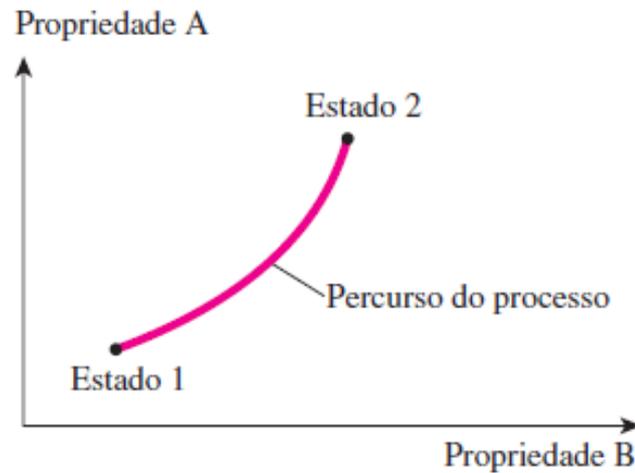
2.1.2 Propriedades, estados e processos

Conforme as definições anteriores, um ciclo termodinâmico é composto por uma sequência de processos, os quais são transformações em um sistema, levando-o de um estado ao outro. Estes, por sua vez, são caracterizados pelas propriedades termodinâmicas (BOLES; ÇENGEL, 2013). A Figura 2 retrata um processo termodinâmico entre os estados 1 e 2, onde os eixos representam duas propriedades independentes arbitrárias do fluido.

¹ Tratando-se de sistemas fechados

² Uma propriedade é chamada de específica quando considera-se seu valor por unidade de massa

Figura 2 – Processo Termodinâmico



Fonte: (BOLES; ÇENGEL, 2013)

Dessa forma, no dimensionamento de uma máquina térmica, é necessário analisar essas condições e mensurar seus efeitos sobre o resultado global do ciclo.

2.1.2.1 Especificação de um Estado

Pelo postulado de estado, o estado de uma substância pura compressível é totalmente definido a partir da determinação de duas propriedades independentes³ deste estado (SONNTAG; WYLEN; BORGNAKKE, 2003).

2.1.2.2 Processos com propriedades constantes

Nos ciclos ideais, é comum considerar processos nos quais uma das propriedades é mantida constante. Esses processos recebem uma nomenclatura específica, conforme a listagem abaixo (BOLES; ÇENGEL, 2013):

- Isobárica: Processo realizado com pressão constante;
- Isotérmico: Processo realizado com temperatura constante;
- Isocórico: Processo realizado com volume constante;
- Isentrópico: Processo realizado com entropia constante;
- Isentálpico: Processo realizado com entalpia constante.

Além de processos com propriedades constantes, também existem processos cuja transferência de calor é nula. Estes recebem o nome de processos adiabáticos.

³ Duas propriedades são ditas independentes quando é possível variar uma enquanto a outra permanece constante

2.1.2.3 Tabela de Propriedades

Decorre do postulado de estado para uma substância pura que, definindo duas propriedades termodinâmicas independentes, todas as outras propriedades referentes a este estado ficam definidas. No entanto, na maioria das substâncias, essas relações são complexas e não podem ser expressas a partir de equações simples (BOLES; ÇENGEL, 2013). Assim, diversos modelos empíricos e teóricos foram desenvolvidos para determinar essas propriedades, cujos resultados geralmente são apresentados em forma de tabela. Outros estudos, ainda, apresentam métodos numéricos que são capazes de aproximar com excelente precisão os resultados experimentais (BELL et al., 2014; AZIMIAN; ARRIAGADA; ASSADI, 2004).

2.1.3 Curvas de saturação Líquido-Vapor

Nos ciclos a vapor, o fluido de trabalho alterna entre as fases líquido e vapor, passando por regiões de equilíbrio de fases entre essas transições. As propriedades termodinâmicas dessas fases diferem consideravelmente. Assim, essa alternância é a chave para o resultado do ciclo, pois é possível aproveitar essa mudança de propriedades para obter um maior trabalho líquido.

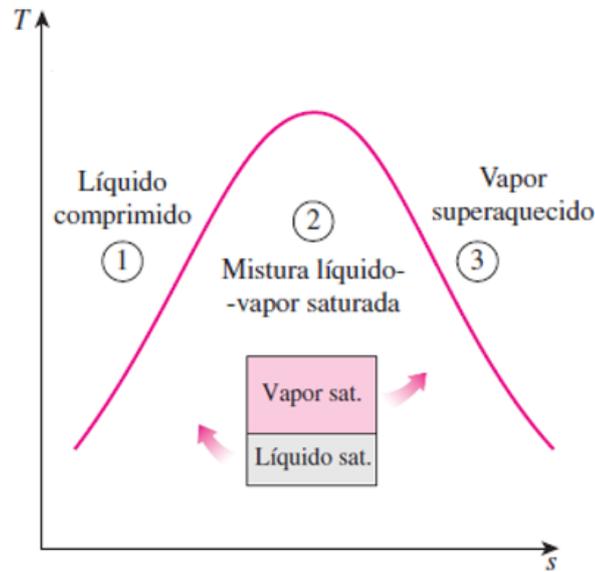
Para esse tipo de ciclo, é comum visualizá-lo graficamente junto à curva de saturação líquido-vapor do fluido de trabalho utilizado. A Figura 3 demonstra a curva de saturação da água no diagrama T-s, destacando três regiões. A curva rosa é formada pelo encontro das curvas de líquido saturado (à esquerda) e de vapor saturado (à direita). O ponto onde elas se encontram (no topo da curva) é chamado de ponto crítico. Na parte exterior dessas curvas, existem a região 1, chamada de líquido comprimido e a região 3, chamada de vapor superaquecido. Já no interior da curva, as duas fases coexistem em equilíbrio, sendo que a proporção entre líquido e vapor nessa mistura depende da distância horizontal em relação a curva de líquido e vapor saturado (BOLES; ÇENGEL, 2013).

2.1.3.1 Título

O título é uma propriedade utilizada na região do diagrama que há mistura entre as fases líquido e vapor (região 2 da Figura 3). Essa propriedade é definida como sendo a proporção entre a massa de vapor e a massa total da mistura. Assim, o título é uma grandeza cujo valor está compreendido entre 0 e 1, onde 0 representa um sistema composto por líquido saturado e 1 por vapor saturado (BOLES; ÇENGEL, 2013).

A relevância do cálculo do título nos ciclos à vapor é devido a restrições na operação das turbinas. Nesse caso, a presença de líquido em quantidades elevadas é prejudicial para a seu funcionamento e pode provocar erosão no equipamento (BOLES; ÇENGEL, 2013). Desse modo, o título torna-se uma variável importante a ser considerada no

Figura 3 – Curva de saturação da água no diagrama T-s



Fonte: (BOLES; ÇENGEL, 2013)

dimensionamento deste tipo de ciclo. Ainda, o título permite a identificação da quantidade de calor latente necessária nas trocas térmicas.

2.1.4 Leis da Termodinâmica

Os fenômenos da termodinâmica são regidos a partir de três leis fundamentais. Elas estão descritas sucintamente nas subseções seguintes:

2.1.4.1 Lei zero

A lei zero da termodinâmica declara que se dois corpos estão em equilíbrio térmico (possuem mesma temperatura) com outro corpo, eles estão em equilíbrio térmico entre si (SONNTAG; WYLEN; BORGNAKKE, 2003).

2.1.4.2 Primeira lei

A primeira lei da termodinâmica é uma constatação do princípio de conservação de energia. Visto que calor e trabalho são as duas formas possíveis que a energia pode cruzar a fronteira de um sistema fechado, a primeira lei resulta em:

$$(\text{Energia entrando}) - (\text{Energia saindo}) = (\text{Variação de energia do sistema})$$

ou ainda

$$Q - W = \Delta E \quad (2.1)$$

Em um Ciclo Termodinâmico, os estados inicial e final do fluido de trabalho são os mesmos. Sendo assim, $\Delta E_{ciclo} = 0$, resultando em:

$$Q = W \quad (2.2)$$

2.1.4.3 Segunda lei

A primeira lei da termodinâmica não se preocupa com o sentido da transferência de energia. Seguindo ela, seria possível aquecer um corpo de alta temperatura naturalmente a partir de um corpo de menor temperatura (de modo que o balanço de energia seja respeitado). A segunda lei impõe as restrições para esse tipo de processo, que, na prática, não acontecem (BOLES; ÇENGEL, 2013).

A segunda lei é normalmente expressa a partir dos enunciados de Kelvin-Planck e de Clausius. Ambos podem ser utilizados de forma equivalente.

- Enunciado de Kelvin-Planck: “É impossível para qualquer dispositivo que opera em um ciclo receber calor de um único reservatório e produzir uma quantidade líquida de trabalho.”
- Enunciado de Clausius: “É impossível construir um dispositivo que funcione em um ciclo e não produza qualquer outro efeito que não seja a transferência de calor de um corpo com temperatura mais baixa para um corpo com temperatura mais alta.”

2.1.5 Classificações

Diversos dispositivos presentes no cotidiano e no ambiente industrial operam a partir de fluidos que realizam um ciclo termodinâmico. Os motores (ou máquinas térmicas), cujo objetivo é a geração de potência, são chamados de ciclos de potência. Já aqueles responsáveis por gerar um efeito de refrigeração, são classificados como ciclos de refrigeração (BOLES; ÇENGEL, 2013).

Os ciclos termodinâmicos também podem ser classificados conforme os seguintes critérios:

1. Quanto à fase do fluido de trabalho:
 - Ciclo a Vapor;
 - Ciclo a Gás.
2. Quanto ao circuito do fluido de trabalho:
 - Ciclos Fechados;
 - Ciclos Abertos.

Nos ciclos a vapor, o fluido de trabalho passa por mudanças de fase durante a operação, existindo ora na fase de vapor e ora na fase líquida. Em contraste, nos ciclos a gás o fluido permanece no estado gasoso em todo o processo (BOLES; ÇENGEL, 2013).

Nos ciclos fechados, o fluido de trabalho retorna ao estado inicial após a sucessão de transformações termodinâmicas do sistema. Por outro lado, nos dispositivos que operam segundo um ciclo aberto, o fluido de trabalho é renovado a cada final de ciclo, o que não caracteriza um ciclo termodinâmico propriamente dito, mas sim um ciclo mecânico. Ainda assim, na modelagem de ciclos abertos, frequentemente adotam-se aproximações de forma a estudá-lo como sendo um ciclo fechado (BOLES; ÇENGEL, 2013; SONNTAG; WYLEN; BORGNACKE, 2003).

São exemplos cotidianos de dispositivos que operam segundo ciclos termodinâmicos os motores a combustão interna dos automóveis e as geladeiras. Os motores de automóveis são ciclos de geração de potência, que utilizam o ar como gás de trabalho e descrevem um circuito aberto (o ar é expelido ao fim de cada ciclo). Já as geladeiras são ciclos fechados de refrigeração utilizando um gás refrigerante como fluido de trabalho.

2.1.6 Funcionamento das máquinas térmicas

Apesar das diferentes classificações, o funcionamento macro das máquinas térmicas é comum para todas, conforme apresentado na Figura 4 (BOLES; ÇENGEL, 2013).

Assim, o funcionamento de qualquer máquina térmica tem as seguintes características (BOLES; ÇENGEL, 2013):

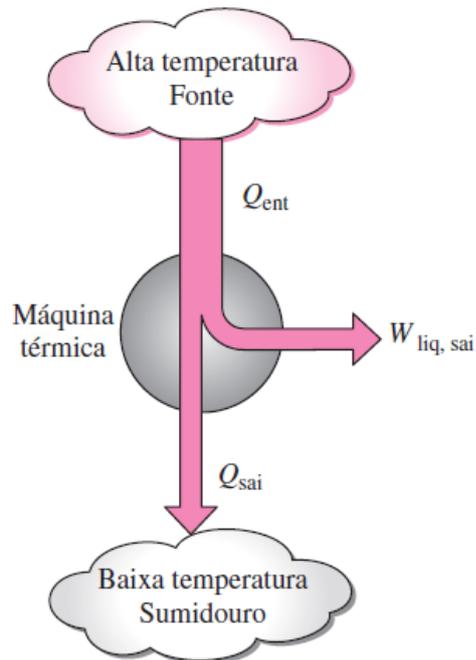
- O sistema recebe calor de um reservatório de alta temperatura;
- Parte do calor recebido é convertido em trabalho;
- O restante do calor é rejeitado para um reservatório de baixa temperatura (também chamado de sumidouro).

Portanto, na operação de uma máquina térmica, o calor flui em seu sentido natural (do ponto de maior temperatura para o ponto de menor temperatura), realizando trabalho neste caminho.

2.1.7 Tipos de Ciclos Ideais

Os processos que ocorrem nos dispositivos reais são bastante complexos de serem modelados devido às irreversibilidades dos sistemas. De acordo com o tipo de análise, é possível considerar equipamentos idealizados para simplificar o estudo. Como consequência, obtém-se um Ciclo Ideal, cujo resultado numérico pode diferir consideravelmente do real

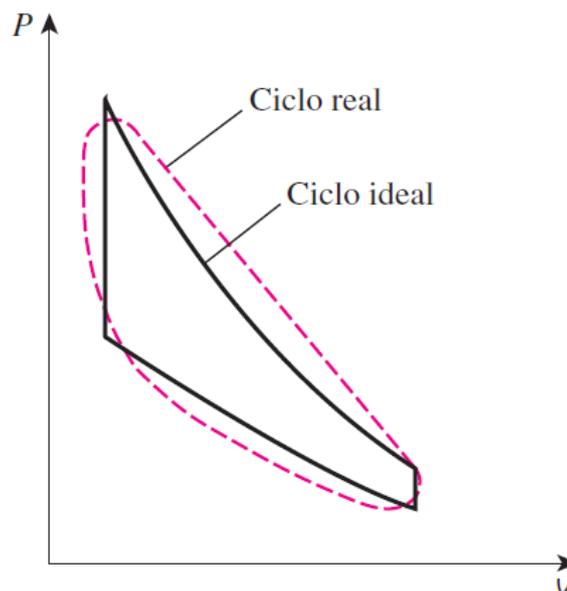
Figura 4 – Princípio de Funcionamento de uma Máquina Térmica



Fonte: (BOLES; ÇENGEL, 2013)

(Figura 5). Ainda assim, os Ciclos Ideais são interessantes objetos de estudo, pois permitem compreender a influência das propriedades termodinâmicas sobre o ciclo e estabelecem os limites superiores de rendimento dos ciclos reais (BOLES; ÇENGEL, 2013; SONNTAG; WYLEN; BORGNAKKE, 2003).

Figura 5 – Defasagem entre ciclos reais e ideais



Fonte: (BOLES; ÇENGEL, 2013)

2.1.8 Ciclo de Carnot

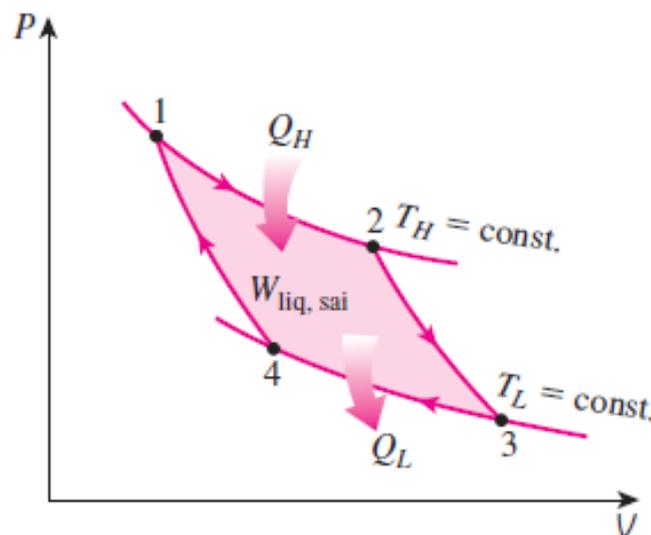
O Ciclo de Carnot é um ciclo idealizado de forma a possuir a maior eficiência possível entre dois patamares de temperatura. Isto é alcançado utilizando somente processos reversíveis. Na máquina de Carnot, esses processos são (BOLES; ÇENGEL, 2013):

- Expansão isotérmica reversível;
- Expansão adiabática reversível;
- Compressão isotérmica reversível;
- Compressão adiabática reversível.

As Figuras 1 e 6 representam esses processos do Ciclo de Carnot. Ainda, na Figura 6 é possível notar as regiões de entrada (representada por Q_H) e saída (Q_L) de calor. A região interna do gráfico representa o trabalho líquido realizado pela máquina que, seguindo a primeira lei, deve ser igual a diferença entre o calor de entrada e o calor rejeitado.

Portanto, mesmo sendo a máquina térmica ideal de melhor desempenho, a máquina de Carnot sempre possui um rendimento abaixo de 100%, condizente com a segunda lei da termodinâmica.

Figura 6 – Ciclo de Carnot



Fonte: (BOLES; ÇENGEL, 2013)

2.1.9 Ciclo Rankine simples

O Ciclo Rankine é o ciclo de potência a vapor ideal. Ele é responsável pela maior parte da geração mundial de energia, sendo utilizado em usinas como termoelétrica, nuclear, geotérmica e solar (RAO, 2015).

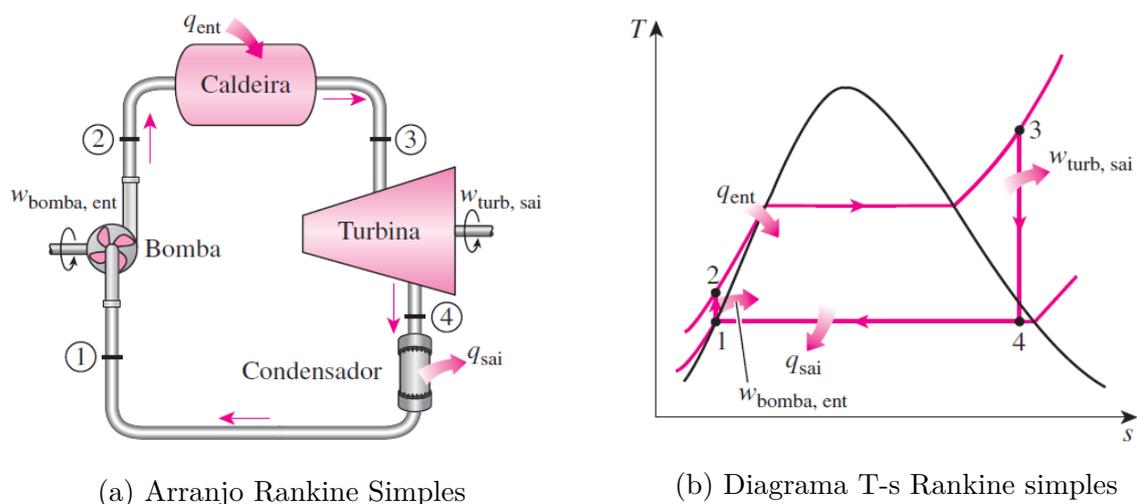
Esse ciclo é composto por quatro equipamentos: bomba, caldeira, turbina e condensador. Inicialmente, o fluido de trabalho no estado de líquido saturado é comprimido na bomba, elevando sua pressão. Em seguida, esse líquido pressurizado recebe calor em uma caldeira, atingindo a região de vapor saturado ou superaquecido. O passo seguinte é a expansão desse vapor em uma turbina, onde o fluido transfere parte da sua energia armazenada para a geração de trabalho e retorna à pressão inicial do sistema. Por fim, o fluido segue para um condensador, onde rejeita calor para a fonte fria e retorna ao estado inicial.

Esses quatro processos que compõem o Ciclo Rankine estão resumidos na listagem a seguir:

- **1-2)**⁴ Bomba - Compressão adiabática isoentrópica;
- **2-3)** Caldeira - Entrada de calor isobárica;
- **3-4)** Turbina - Expansão adiabática isoentrópica;
- **4-1)** Condensador - Saída de calor isobárica.

A configuração dos equipamentos está retratada na Figura 7a. O diagrama T-s da Figura 7b representa esses processos sobrepostos à curva de saturação da água, apresentando diversas informações de interesse, como os quatro estados de equilíbrio, os processos que o fluido é submetido e as regiões as quais o calor e trabalho cruzam a fronteira do sistema.

Figura 7 – Ciclo Rankine Simples



Fonte: (BOLES; ÇENGEL, 2013)

⁴ Notação: A-B) = Processo no qual o fluido sai do ponto A e atinge o ponto B

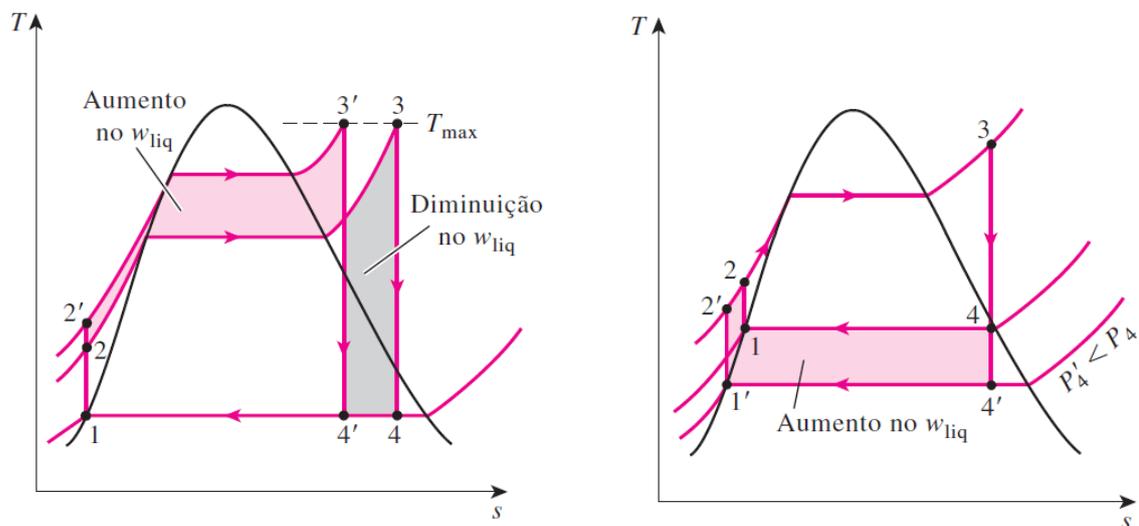
Como forma de elevar a eficiência térmica, é comum realizar modificações no Ciclo Rankine Simples. Essas modificações são baseadas em aumentar a temperatura média do fornecimento de calor ou reduzir a temperatura de rejeição de calor, resultando em uma maior capacidade de produção de trabalho líquido. Dois importantes tipos de alterações resultam nos ciclos com reaquecimento e nos ciclos regenerativos.

2.1.10 Ciclo Rankine com Reaquecimento

A temperatura de saturação de um fluido depende da pressão que ele está submetido. Duas formas possíveis de aumentar a eficiência do ciclo Rankine é pelo aumento da pressão máxima ou pela redução a pressão mínima do sistema. Desse modo, a temperatura média de fornecimento de calor aumenta no primeiro caso e a temperatura média de rejeição de calor diminui no segundo caso. (BOLES; ÇENGEL, 2013).

No entanto, é possível observar nas Figura 8a e 8b que há um efeito colateral para essas soluções: o título na saída da turbina é menor.

Figura 8 – Efeito da variação dos limites de pressão



(a) Efeito do aumento da pressão máxima

(b) Efeito da diminuição da pressão mínima

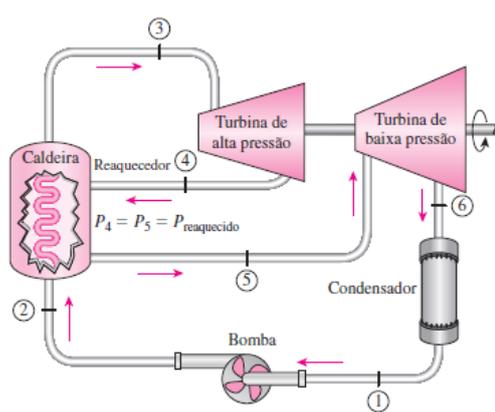
Fonte: (BOLES; ÇENGEL, 2013)

A solução para esse problema é o Ciclo Rankine com reaquecimento intermediário. Essa é uma modificação do ciclo Rankine simples que divide a geração de trabalho em duas turbinas, inserindo um reaquecimento entre essas elas.

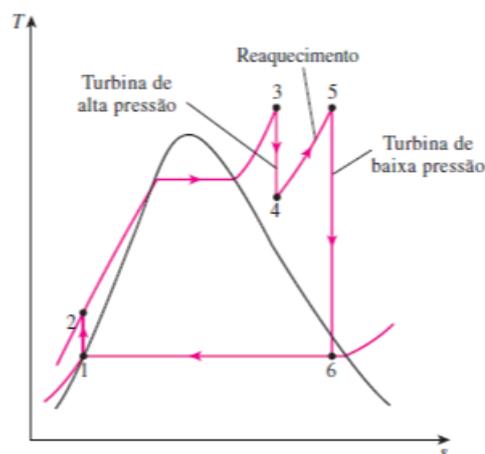
Assim, a etapa de compressão e aquecimento do fluido é idêntica ao Rankine simples. Na expansão, o fluido é parcialmente expandido em uma turbina de alta pressão até atingir uma pressão intermediária. Em seguida, o fluido retorna à caldeira, onde recebe calor

até retornar à temperatura máxima do sistema⁵. Então, o fluido passa por um segundo processo de expansão na turbina de baixa pressão, atingindo novamente a pressão mínima no sistema. Por fim, a condensação ocorre de forma idêntica ao Rankine simples. Essa configuração está apresentada na Figura 8.

Figura 9 – Ciclo Rankine Reaquecido



(a) Arranjo Rankine Reaquecido



(b) Diagrama T-s Rankine Reaquecido

Fonte: (BOLES; ÇENGEL, 2013)

É importante destacar que, para o ciclo Rankine com reaquecimento, há dois momentos de entrada de calor e de realização de trabalho que devem ser considerados no resultado final do ciclo.

Como é possível observar no diagrama T-s, o reaquecimento intermediário aumenta o valor do título na saída da turbina (ponto 6) em relação ao valor correspondente para o Rankine simples.

2.1.11 Ciclo Rankine Regenerativo

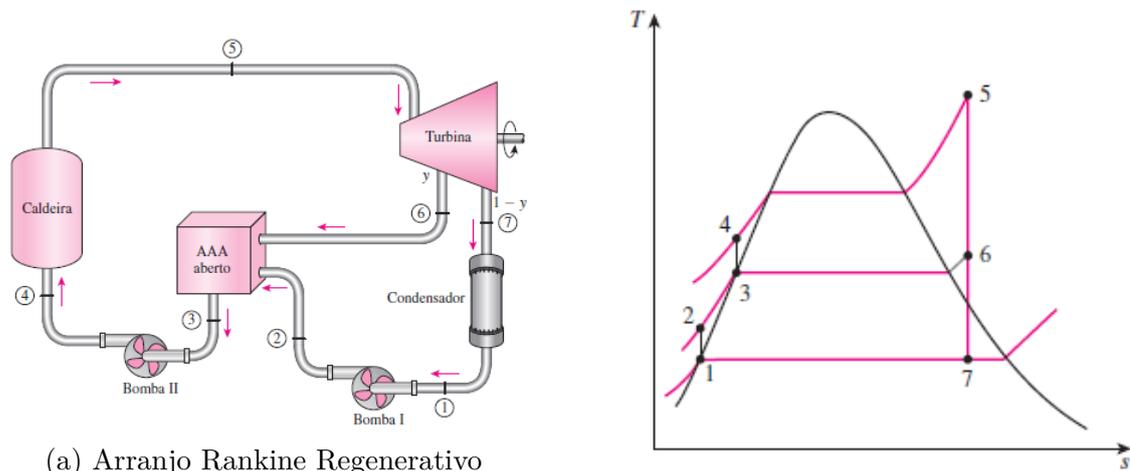
Outro método utilizado para elevar a eficiência dos ciclos Rankine é a inclusão de aquecedores da água de alimentação (AAA), resultando na elevação da temperatura média de recebimento de calor na caldeira. Esses aquecedores, também chamados de regeneradores, são trocadores de calor que aproveitam parte do fluido que está sendo expandido na turbina para aquecer o fluido de alimentação (SONNTAG; WYLEN; BORGNAKKE, 2003).

A Figura 10 representa o arranjo mais claramente. Após a compressão na bomba 1, o fluido de trabalho atinge a pressão intermediária do sistema e, em seguida, entra no regenerador. Esse fluido, então, é misturado diretamente com uma parcela de fluido que

⁵ Dependendo do arranjo, a temperatura de reaquecimento pode ser diferente da temperatura do aquecimento anterior. Neste trabalho, essas duas temperaturas serão consideradas iguais

foi extraída na turbina, cuja temperatura é superior e a pressão é igual. Essa mistura eleva a temperatura do fluido que irá ser aquecido na caldeira, aumentando também o rendimento total do ciclo.

Figura 10 – Ciclo Rankine Regenerativo



(a) Arranjo Rankine Regenerativo

(b) Diagrama T-s Rankine Regenerativo

Fonte: (BOLES; ÇENGEL, 2013)

Esse novo arranjo possui certas peculiaridades que devem ser observadas:

- A fração mássica de fluido que é extraída no regenerador deve ser equivalente à quantidade necessária para que o fluido entre em equilíbrio na temperatura de líquido saturado para o valor de pressão intermediária;
- A geração de trabalho no processo 6-7 (tanto quanto o trabalho da bomba no processo 1-2) ocorre apenas com a fração mássica restante que não foi extraída para o reaquecimento;
- Apesar do aquecimento do fluido no processo 2-3, só há entrada de calor no sistema no processo 4-5 (pela caldeira), visto que o regenerador apenas reaproveita o calor que já havia sido inserido anteriormente no sistema.

Portanto, a inclusão do regenerador tem com objetivo aumentar a eficiência total do ciclo. A inclusão de mais regeneradores resulta em ganhos ainda maiores de eficiência. No entanto, para cada novo regenerador o efeito se torna menos relevante, fazendo com que o número ideal de regeneradores seja obtido pelo equilíbrio entre custos e aumento de eficiência. Neste trabalho, o ciclo Rankine regenerativo foi modelado contendo apenas um regenerador.

2.2 Softwares no processo de aprendizagem

Nesta seção é realizada uma revisão da literatura a respeito dos benefícios do uso de softwares interativos no processo de aprendizagem, os aspectos importantes na elaboração do sistema, experimentos realizados e são apresentados outros trabalhos de implementação de softwares no ensino de engenharia.

2.2.1 Benefícios dos aplicativos móveis

Através das evoluções tecnológicas, surgem grandes oportunidades de inovações no setor educacional. Segundo (CHELLA et al., 2002), a utilização de ferramentas computacionais no processo de aprendizagem é caracterizada por quatro personagens: o computador, o software educativo, o instrutor e o estudante.

Na década 2000, como apresentado por (ISLAM; ISLAM; MAZUMDER, 2010), a utilização de aplicativos móveis ainda experimentava diversas limitações relacionados à tecnologia dos dispositivos da época, em consequência de suas pequenas telas, dificuldade de navegação e baixo poder de processamento. Nos últimos anos, no entanto, sendo um importante meio de comunicação e entretenimento, os celulares se tornaram um integrante essencial no cotidiano de maior parte da população mundial (ISLAM; ISLAM; MAZUMDER, 2010; JENG et al., 2010) e, superaram os computadores convencionais em popularidade.

Dessa forma, diversos pesquisadores voltaram-se para as possibilidades de utilização dos aplicativos móveis como estratégia de motivação na aprendizagem, desde a educação básica (SOUSA et al., 2016), até cursos de graduação. Os estudos realizados por (JENG et al., 2010) e (SUBRAMANYA; FARAHANI, 2012) apresentam os impactos positivos dessas ferramentas. Os benefícios incluem a mobilidade e disponibilidade, determinação do ritmo de estudo, auxílio na assimilação de conceitos abstratos, elevado grau de retenção de ideias, maior engajamento, maior motivação, eficiência de tempo e a liberdade de experimentação de diferentes possibilidades. (AMORIM et al., 2005) reforça a importância da experimentação no processo de aprendizagem na engenharia, pois a exploração individual dos alunos promove uma melhor capacidade de assimilação do através da teorização genérica dos conteúdos.

No quesito de software, é crucial que o sistema apresente um design de interface que seja intuitivo, de fácil interação e fácil visualização dos resultados (SUBRAMANYA; FARAHANI, 2012; JOU; LIN; TSAI, 2016). Essas considerações são importantes para reduzir o custo de adaptação, não sobrecarregar cognitivamente o estudante e facilitar o processo de imersão e experimentação. Além disso, eles devem ser desenhados de forma estratégica de modo que sirvam de complemento às aulas convencionais.

Outro ponto importante destacado é em relação ao desenvolvimento e acesso

a softwares livres, visto que programas especializados podem ser altamente custosos, inviabilizando a implementação em diversas instituições de ensino. Por outro lado, além de gratuito, os softwares livres permitem que outros pesquisadores trabalhem na evolução contínua do sistema (AMORIM et al., 2005; BACHEGA, 2018).

Assim, o principal propósito apontado pela literatura é propiciar meios inovadores para que os docentes possam complementar o modelo tradicional de ensino, almejando maior motivação, engajamento e performance por parte dos estudantes (AMORIM et al., 2005).

2.2.2 Softwares no ensino de engenharia

Ainda há uma grande escassez de softwares, principalmente de aplicativos móveis, nos modelos supracitados. Para que hajam crescimentos nesse sentido, é necessário reunir conhecimentos em áreas específicas da engenharia e de desenvolvimento de softwares.

Anteriormente à evolução dos *smartphones*, os trabalhos nessa direção estavam limitados às aplicações para computadores pessoais. Braga (BRAGA, 2000), avalia o uso de aplicativos de simulação para computadores no ensino de Transferência de Calor no curso de Engenharia Mecânica. O pesquisador identifica a consolidação de conceitos abstratos como um importante benefício alcançado pelos alunos, visto que o processo de experimentação, sobretudo quando auxiliado pelos professores, permite a análise de uma ampla quantidade de exemplos em um espaço de tempo reduzido.

No ramo da Engenharia Civil, (SALSA, 2018) desenvolveram um aplicativo multi-plataforma para os sistemas operacionais Android, iOS e Windows Phone a fim de dimensionar estruturas de concreto armado. Os questionários de opinião apresentaram uma boa aceitação da ferramenta por parte dos alunos e professores, que, de forma conjunta, contribuem com sugestões para a evolução da plataforma, a qual é atualizada constantemente.

(JOU; LIN; TSAI, 2016) conduziram experimentos quantitativos e qualitativos de modo a analisar os efeitos relacionados à performance, engajamento e carga cognitiva dos discentes. Os pesquisadores desenvolveram um aplicativo móvel temático para a área de Engenharia de Materiais e testaram sua eficácia com questionários de conhecimento e opinião. Para analisar os ganhos de performance no aprendizado, 45 estudantes formaram o grupo experimental, que utilizaram o aplicativo durante o curso, e 42 formaram o grupo de controle, o qual foi submetido somente ao método tradicional de ensino. A avaliação de performance foi realizada pela aplicação um teste de conhecimentos no início e ao final do curso. Os resultados dos testes reforçaram a melhora no desempenho proporcionada pelo uso do sistema como ferramenta auxiliar. Ainda, foram realizadas avaliações qualitativas com esses grupos, revelando uma postura mais positiva dos participantes que utilizaram o

aplicativo interativo no processo de aprendizagem. Os efeitos a respeito da carga cognitiva não foram significantes, salientando que, quando desenhado adequadamente, o sistema não apresenta um custo elevado de adaptação.

A partir dos conceitos físicos apresentados e dos principais aspectos destacados pelos pesquisadores na elaboração de aplicações efetivas, os próximos capítulos são dedicados aos processos de planejamento e desenvolvimento de um aplicativo Android com objetivo de auxiliar estudantes da Engenharia Mecânica no dimensionamento de ciclos termodinâmicos.

3 Metodologia

Neste capítulo, são descritas as equações matemáticas necessárias na resolução dos problemas, a modelagem dos algoritmos propostos e suas implementações na linguagem de programação Python. A linguagem Python foi escolhida para o desenvolvimento inicial e para a apresentação neste trabalho devido à sua sintaxe simples, que facilita a execução de testes rápidos e o entendimento por leitores com menor familiaridade com linguagens de programação.

As equações matemáticas utilizadas são derivadas das leis da termodinâmica e dos demais conceitos abordados na Seção 2.1.

Em seguida, são realizadas as modelagens dos ciclos Rankine, as quais são apresentadas por meios de fluxogramas derivados de uma análise qualitativa da sequência de passos necessários para resolver os problemas propostos.

Cada fluxograma foi dividido em fragmentos, de modo a discutir cada trecho individualmente com mais detalhes. Além da análise teórica, são apresentados os códigos referentes a cada fragmento individual. Os códigos completos são apresentados no Apêndice A e estão disponíveis em (SALES, 2019b).

Por fim, são feitas as considerações acerca da implementação do aplicativo móvel e a montagem de um fluxograma de navegação da plataforma. O código do aplicativo não é apresentado neste documento devido à sua extensão, mas pode ser acessado em (SALES, 2019b).

3.1 Considerações

Para todos os ciclos apresentados, foram considerados os seguintes pontos:

1. Processos ideais;
2. Água como fluido de trabalho;
3. O fluido na entrada das bombas está no estado de líquido saturado;

3.2 Modelagem matemática

3.2.1 Balanço de Energia

Os dispositivos utilizados no Ciclo Rankine (bomba, turbina, caldeira e condensador) operam com escoamento em regime permanente. Dessa forma, considerando um volume

de controle em cada um desses dispositivos, o balanço de energia da primeira lei da termodinâmica resulta em (SONNTAG; WYLEN; BORGNAKKE, 2003):

$$q + h_e + V_e^2/2 + gZ_e = h_s + V_s^2/2 + gZ_s + w \quad (3.1)$$

Onde os subscritos e e s referem-se a entrada e a saída do escoamento.

A Equação 3.1 pode ser rearranjada de forma a destacar cada forma de energia envolvida:

$$q - w = (h_e - h_s) + (V_e^2 - V_s^2)/2 + (gZ_e - gZ_s) \quad (3.2)$$

O termo $(q - w)$ refere-se às formas possíveis que a energia pode cruzar a fronteira do sistema. $(h_e - h_s)$ indica a variação de entalpia entre a entrada e saída do sistema. Por fim, $(V_e^2 - V_s^2)/2$ retrata a variação de energia cinética e $(gZ_e - gZ_s)$ a variação de energia potencial.

No entanto, a variação de energia cinética e potencial nesses dispositivos são desprezíveis em comparação a variação de entalpia (BOLES; ÇENGEL, 2013; ZOHURI; MCDANIEL, 2015). Assim, a Equação 3.2 pode ser simplificada, com ótima precisão, para:

$$q - w = (h_e - h_s) \quad (3.3)$$

Portanto, para dispositivos que operam em regime permanente e não apresentam variações de energia cinética e potencial significativas, a variação de energia de um sistema (referentes às transferências de calor e trabalho) pode ser determinada pela variação da entalpia entre dois pontos.

3.2.2 Dimensionamento de ciclos ideais a vapor

No dimensionamento de um ciclo Rankine ideal, as principais variáveis de interesse podem ser obtidas a partir da aplicação do balanço de energia representado na Equação 3.3. Independentemente do arranjo estudado, o ciclo é formado pelos equipamentos que trocam energia através de calor (caldeira e condensador) e os que trocam através de trabalho (bombas e turbinas).

A Equação 3.3 possui implicitamente uma convenção para o sentido do movimento de calor e trabalho. Nela, o calor é considerado positivo quando é transmitido ao sistema e negativo quando deixa o sistema. Já o trabalho é positivo quando é realizado pelo sistema ao meio externo, e negativo quando o meio exerce trabalho sobre o sistema. Nas análises seguintes, a convenção foi mantida, mas o resultado será apresentado como o módulo desses valores:

$$|q - w| = |h_e - h_s| \quad (3.4)$$

Apesar dos valores positivos, é importante destacar o sentido das transferências em cada equipamento. A energia entra no sistema através da aplicação de trabalho pela bomba e pela entrada de calor na caldeira. Em contrapartida, ela deixa o sistema pela realização de trabalho na turbina e na rejeição de calor no condensador. A Figura 7a representa visualmente essas afirmações.

O rendimento, uma das variáveis mais importantes no processo de dimensionamento de qualquer ciclo, também pode ser derivado da Equação 3.3. Como o estado final de um ciclo termodinâmico é igual ao inicial, tem-se $h_e = h_s$. Portanto, para um processo cíclico:

$$q - w = 0 \quad (3.5)$$

$$q = w \quad (3.6)$$

Assim, a Equação 3.6 revela que as transferências de calor e de trabalho em um ciclo são iguais. Isto implica que a diferença entre o calor de entrada e o calor rejeitado representa o trabalho líquido¹ realizado pela máquina. O rendimento expressa simplesmente a fração da energia total que é convertida em trabalho líquido. Assim:

$$n = (q_h - q_l)/q_h \quad (3.7)$$

Como identificado na Seção 2.1.3.1, outra variável importante para o bom funcionamento do ciclo é o título na saída das turbinas. Ele pode ser calculado a partir de uma interpolação de certas propriedades termodinâmicas para encontrar a proporção horizontal entre o estado de líquido saturado e o de vapor saturado (BOLES; ÇENGEL, 2013):

$$s = s_{líq} + x(s_{vap} - s_{líq}) \quad (3.8)$$

$$x = (s - s_{líq})/(s_{vap} - s_{líq}) \quad (3.9)$$

Para o ciclo Rankine regenerativo, outra grandeza que deve ser calculada é a fração mássica de extração de fluido para o regenerador. Ela é utilizada na determinação do trabalho gerado pelas turbinas deste ciclo, visto que a segunda parte da expansão ocorre com apenas uma fração do fluido.

¹ O trabalho líquido é a diferença entre o trabalho realizado pelas turbinas e o trabalho recebido pelas bombas

A fração de fluido extraído pode ser calculado através da aplicação do balanço de energia em um volume de controle que engloba o regenerador aplicando a equação:

$$\dot{m} = (h_3 - h_2)/(h_6 - h_2) \quad (3.10)$$

3.3 CoolProp

CoolProp é uma biblioteca² de código aberto escrita em C++ com suporte para diversas linguagens de programação, incluindo Python e Java (BELL et al., 2014; BELL, 2010a). Neste trabalho, a função PropsSI desta biblioteca é utilizada para determinar as propriedades termodinâmicas do fluido de trabalho a partir de um estado previamente definido por outras duas propriedades (BELL, 2010b).

As seguintes propriedades (e suas letras correspondentes para a função PropsSI³) são utilizadas nos códigos deste trabalho: pressão (P), temperatura (T), entalpia específica (H), entropia específica (S), energia interna específica (U), densidade (D) e qualidade⁴ (Q). A tabela com todas as opções disponíveis pode ser encontrada em (BELL, 2010b).

Para realizar sua implementação e acessá-la nos códigos, é necessário aplicar os seguintes passos:

1. Instalar o pacote CoolProp;
2. Importar a função PropsSI da biblioteca CoolProp dentro do código;
3. Utilizar a função PropsSI para determinar as propriedades desejadas.

3.3.1 CoolProp em Python

O código abaixo representa um exemplo da função PropsSI em Python, sendo utilizada na determinação da entalpia de um estado especificado a partir dos seus valores de pressão e temperatura.

```

1 from CoolProp.CoolProp import PropsSI #Importando a função PropsSI
2
3 p = 10000 #Pressão a 10KPa
```

² No contexto de softwares, bibliotecas são conjuntos de funções que podem ser utilizadas em outros códigos

³ Apesar das propriedades específicas serem representadas por letras minúsculas na literatura, os parâmetros devem ser inseridos em caixa alta nos argumentos da função

⁴ A qualidade é um valor binário que corresponde ao estado de líquido saturado (Q = 0) ou vapor saturado (Q = 1)

```
4 t = 500 #Temperatura a 500K
5 fluido = 'Water' #Água como fluido de trabalho
6
7 h = PropsSI('H', 'P', p, 'T', t, fluido) #Determinando a entalpia
```

O exemplo anterior pode ser dividido em três partes principais: a importação da função PropsSI (linha 1), a seleção dos dados de entrada (linhas 3 a 5) e a determinação do valor de saída (linha 7).

Os dados de entrada devem incluir o valor de duas propriedades e o fluido de trabalho. Neste caso, foram especificados os valores de pressão e temperatura da água.

A saída é o valor da propriedade desejada, resultado da função PropsSI. Essa função deve receber cinco parâmetros no seguinte formato: (Saída desejada, Propriedade conhecida 1, Valor da propriedade 1, Propriedade conhecida 2, Valor da propriedade 2, Fluido de trabalho)⁵. Para o exemplo anterior, os parâmetros utilizados são:

1. Saída desejada: Entalpia ('H');
2. Propriedade conhecida 1: Pressão ('P');
3. Valor da propriedade conhecida 1: 10kPa (através da *p*);
4. Propriedade conhecida 2: Temperatura ('T');
5. Valor da propriedade conhecida 2: 500K (através da variável *t*);
6. Fluido de trabalho: *Water* (através da variável *fluido*);

Se os valores de pressão e temperatura especificados forem referentes a um ponto da curva de saturação do fluido, a função retorna um erro, visto que esses parâmetros não representam um par de propriedades independentes e existem dois valores possíveis para o resultado. Neste caso, é necessário especificar se o fluido está na região de líquido saturado ou vapor saturado adicionando o sufixo “|liquid” (para líquido saturado) ou “|gas” (para vapor saturado) ao final da propriedade conhecida 1 ou 2⁶ ('P|gas', 'P|liquid', 'T|gas' ou 'T|liquid' são as possíveis opções para o exemplo anterior).

3.4 Algoritmos para dimensionamento de Ciclos Rankine

Os ciclos Rankine podem ser modelados por uma sequência de passos definida, de forma a calcular todas as variáveis relevantes para seu dimensionamento. Assim, faz-se

⁵ A ordem das propriedades 1 e 2 não é relevante e pode ser escolhida arbitrariamente

⁶ Outro erro é retornado caso o sufixo seja adicionado às duas propriedades simultaneamente

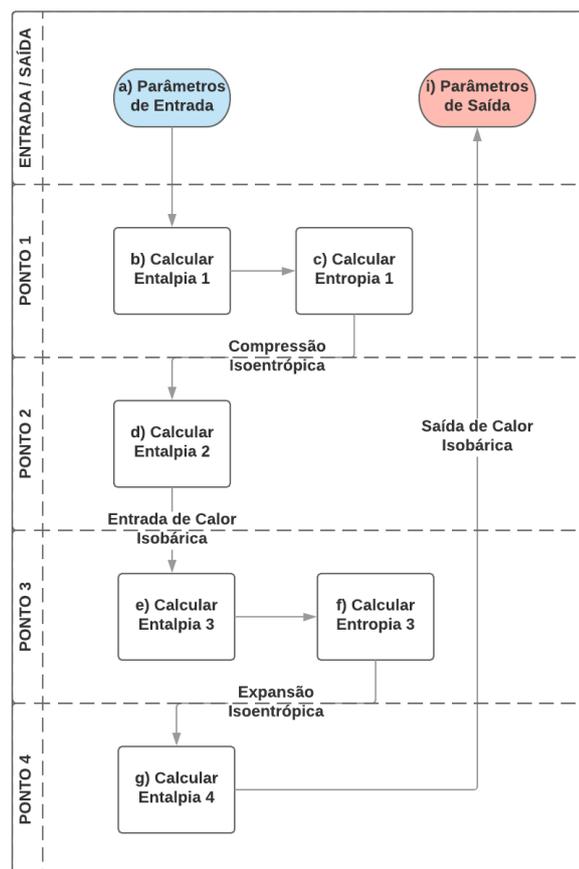
necessário definir previamente esses passos e as ações que devem ser feitas pelo software em cada um deles.

As subseções seguintes se dedicam à montagem de fluxogramas para a resolução dos ciclos rankine simples, reaquecido e regenerativo, além de definir as equações matemáticas necessárias e a implementação em Python para cada etapa estabelecida.

3.4.1 Ciclo Rankine Simples

O ciclo Rankine simples é composto por quatro processos termodinâmicos, cada qual sendo realizado em um equipamento específico (conforme mostrado nas Figuras 7a e 7b). Assim, é necessário aplicar o balanço de energia em cada um desses sistemas separadamente. No fluxograma da Figura 11, foi condensada a sequência de etapas para a resolução do Ciclo Rankine Simples, destacando as parcelas de cada sistema.

Figura 11 – Fluxograma do Ciclo Rankine Simples



Fonte: Autor

De acordo com a Equação 3.3, as transferências de calor e a realização de trabalho do sistema ou para o sistema podem ser calculados através das variações de entalpia. Desse modo, as ações realizadas na análise dos ciclos são focadas em determinar a entalpia para

cada estado de equilíbrio. Cada ação referente aos itens apresentados no fluxograma estão descritos a seguir:

a) Parâmetros de Entrada

Os parâmetros de entrada considerados são:

- Pressão inferior (no ponto 1);
- Pressão superior (no ponto 3);
- Temperatura superior (no ponto 3);

b) Calcular Entalpia 1

A entalpia do estado 1 pode ser obtida diretamente dos dados de entrada, utilizando a pressão inferior e a qualidade do fluido (líquido saturado).

c) Calcular Entropia 1

A entropia do estado 1 pode ser obtida analogamente ao item anterior.

d) Calcular Entalpia 2

Após a compressão na bomba, a pressão do estado 2 atinge o valor da pressão superior. Por ser isoentrópica, o valor da entropia é igual a do estado 1. Assim, a entalpia do estado 2 pode ser obtida através dos valores de pressão e entropia deste estado.

e) Calcular Entalpia 3

A entalpia do estado 3 pode ser obtida diretamente dos dados de entrada, utilizando a pressão superior e a temperatura superior.

f) Calcular Entropia 3

A entropia do estado 3 pode ser obtida analogamente ao item anterior.

g) Calcular Entalpia 4

Após a expansão na turbina, a pressão do estado 4 atinge o valor da pressão inferior. Por ser isoentrópica, o valor da entropia é igual a do estado 3. Assim, a entalpia do estado 4 pode ser obtida através dos valores de pressão e entropia deste estado.

h) Parâmetros de Saída

Os parâmetros de saída desejados são: rendimento, trabalho líquido, título na saída da turbina⁷, calor de entrada e calor de saída. Esses valores são obtidos através das equações listadas na seção 3.2.

⁷ O cálculo do título só pode ser feito se o fluido na saída da turbina estiver no interior da curva de saturação

As etapas e ações definidas para o ciclo Rankine simples são utilizadas como base para os próximos dois arranjos estudados.

3.4.2 Implementação Python do Ciclo Rankine Simples

Após a definição do algoritmo, foram elaborados os códigos para realizar as ações estabelecidas anteriormente.

a) Parâmetros de Entrada

```
1     #Parâmetros de Entrada
2     p1 = float(input("Insira a pressão inferior do sistema: "))
3     p3 = float(input("Insira a pressão superior do sistema: "))
4     t3 = float(input("Insira a temperatura superior do sistema: "))
5     fluido = 'Water'
```

b) Estado 1

```
1     #Estado 1
2     h1 = PropsSI('H', 'P', p1, 'Q', 0, fluido) #Entalpia 1
3     s1 = PropsSI('S', 'P', p1, 'Q', 0, fluido) #Entropia 1
```

c) Estado 2

```
1     #Estado 2
2     p2 = p3 #Compressão
3     s2 = s1 #Processo Isoentrópico
4     h2 = PropsSI('H', 'P', p2, 'S', s2, fluido) #Entalpia 2
```

d) Estado 3

```
1     #Estado 3
2     h3 = PropsSI('H', 'P|gas', p3, 'T', t3, fluido)
3     s3 = PropsSI('S', 'P|gas', p3, 'T', t3, fluido)
```

e) Estado 4

```

1     #Estado 4
2     s4 = s3
3     p4 = p1
4     h4 = PropsSI('H', 'P', p4, 'S', s4, fluido)
5
6     #Cálculo do título:
7     s4l = PropsSI('S', 'P', p4, 'Q', 0, fluido)
8     s4v = PropsSI('S', 'P', p4, 'Q', 1, fluido)
9
10    if (s4 < s4v):
11        x4 = (s4 - s4l)/(s4v - s4l) #Título na turbina
12    else:
13        x4 = 'Saída da turbina superaquecida'

```

f) Parâmetros de Saída

```

1     #Parâmetros de Saída
2     wb = h2 - h1 #Trabalho na bomba
3     wt = h3 - h4 #Trabalho na turbina
4     qh = h3 - h2 #Calor de Entrada
5     ql = h4 - h1 #Calor rejeitado
6     n = (qh - ql)/qh #Rendimento

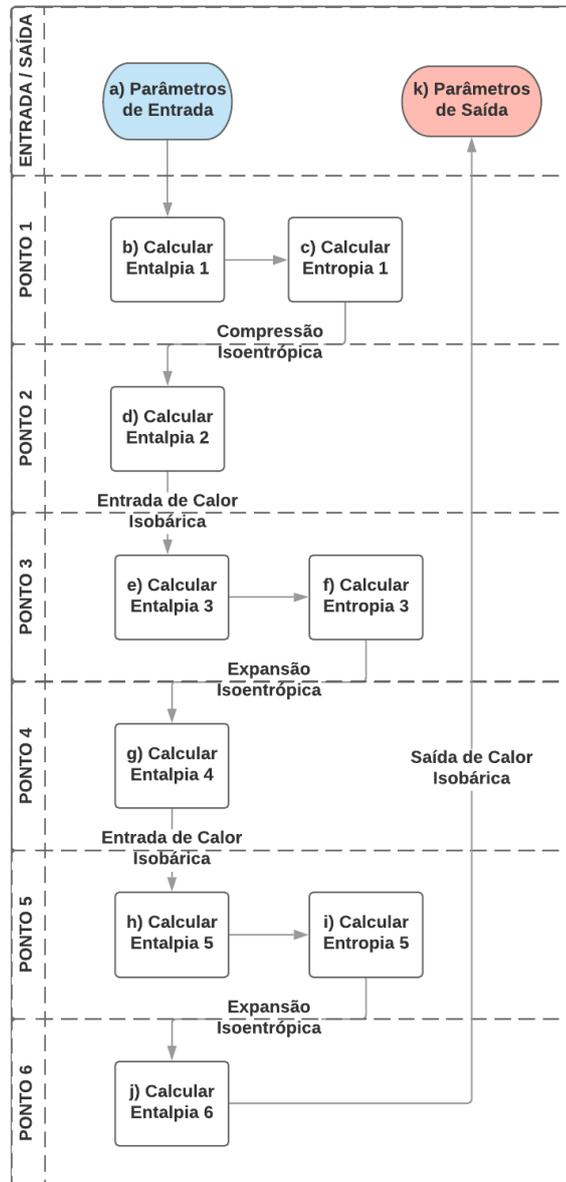
```

Desse modo, a implementação completa do código em Python para o ciclo Rankine simples é feita através da união dos fragmentos de código anteriores. O código completo é apresentado no Apêndice A.1 e em (SALES, 2019b).

3.4.3 Ciclo Rankine com reaquecimento

O arranjo do ciclo Rankine com reaquecimento é uma modificação do Rankine simples. Em sua modelagem, são adicionados um novo parâmetro de entrada, a pressão intermediária, e dois novos passos resultantes do processo de expansão parcial na turbina de alta pressão, o reaquecimento na caldeira e a expansão final na turbina de baixa pressão. Dessa modo, existem dois momentos de entrada de calor e de geração de trabalho no sistema. Um novo parâmetro de saída também deve ser adicionado, visto que, neste caso, é importante considerar o título da saída das turbinas de alta e de baixa pressão. O fluxograma da Figura 13 resume esses processos.

Figura 12 – Fluxograma do Ciclo Rankine com reaquecimento



Fonte: Autor

Considerando as pequenas diferenças no método de resolução em relação ao Rankine simples, não é necessário refazer a análise para cada trecho do fluxograma. No entanto, as peculiaridades deste arranjo estão descritas a seguir:

- A pressão intermediária em kPa é um novo parâmetro de entrada necessário;
- Os processos entre o estado 1 e o estado 3 são idênticos ao Rankine simples;
- O processo de expansão na turbina de alta pressão é análogo ao processo de expansão apresentado no Rankine simples. No entanto, a pressão final atingida é a pressão intermediária do sistema;

- O processo de aquecimento na caldeira é análogo ao aquecimento comum. No entanto, a pressão de entrada da caldeira é a pressão intermediária do sistema;
- O processo de expansão na turbina de baixa pressão é análogo ao processo de expansão apresentado no Rankine simples. No entanto, a pressão de entrada da turbina é a pressão intermediária do sistema;
- É importante avaliar o título na saída das turbinas de baixa e alta pressão.

O código completo para este arranjo é apresentado no Apêndice [A.2](#) e está disponível em ([SALES, 2019b](#)).

3.4.4 Ciclo Rankine regenerativo

Os mesmos princípios da resolução do ciclo Rankine simples são aplicados para o ciclo regenerativo. Entretanto, como destacado na Seção [2.1.11](#), certas particularidades devem ser consideradas em seu dimensionamento. A Figura [10a](#) auxilia no entendimento dessas características.

Assim como no ciclo reaquecido, é necessário definir a pressão intermediária do sistema como um parâmetro de entrada. Esse valor representa o patamar de pressão onde ocorre a extração de fluido na turbina.

Como consequência da extração, é importante considerar nos cálculos que apenas uma fração da água flui através dos processos que ocorrem abaixo da pressão intermediária. Os processos restantes, iguais ou acima da pressão intermediária, ocorrem com a totalidade da vazão mássica disponível. A fração de água extraída é definida utilizando a Equação [3.10](#).

Dessa forma, a Equação [3.4](#) deve ser modificada nos cálculos dos trabalhos nas bombas, na turbina e no condensador para considerar a extração. Sendo $q = 0$ nas bombas e na turbina, $w = 0$ no condensador e arranjando a equação de forma a obter o módulo do calor e trabalho transferido, as equações resultantes são:

- Bombas:

$$w_{b1} = (1 - \dot{m}) * (h_2 - h_1) \quad (3.11)$$

$$w_{b2} = h_4 - h_3 \quad (3.12)$$

$$w_b = w_{b1} + w_{b2} = (1 - \dot{m}) * (h_2 - h_1) + (h_4 - h_3) \quad (3.13)$$

- Turbina:

$$w_{t1} = h_5 - h_6 \quad (3.14)$$

$$w_{t2} = (1 - \dot{m}) * (h_6 - h_7) \quad (3.15)$$

$$w_t = w_{t1} + w_{t2} = (h_5 - h_6) + (1 - \dot{m}) * (h_6 - h_7) \quad (3.16)$$

- Caldeira:

$$q_H = h_5 - h_4 \quad (3.17)$$

- Condensador:

$$q_L = (1 - \dot{m}) * (h_7 - h_1) \quad (3.18)$$

O fluxograma da Figura 13 representa a sequência dos 7 passos necessários para a resolução do ciclo Rankine regenerativo. A implementação Python segue o mesmo padrão do Rankine simples, com a inclusão de novos passos e as modificações nas equações, conforme apresentado anteriormente.

O código completo para este arranjo é apresentado no Apêndice A.3 e está disponível em (SALES, 2019b).

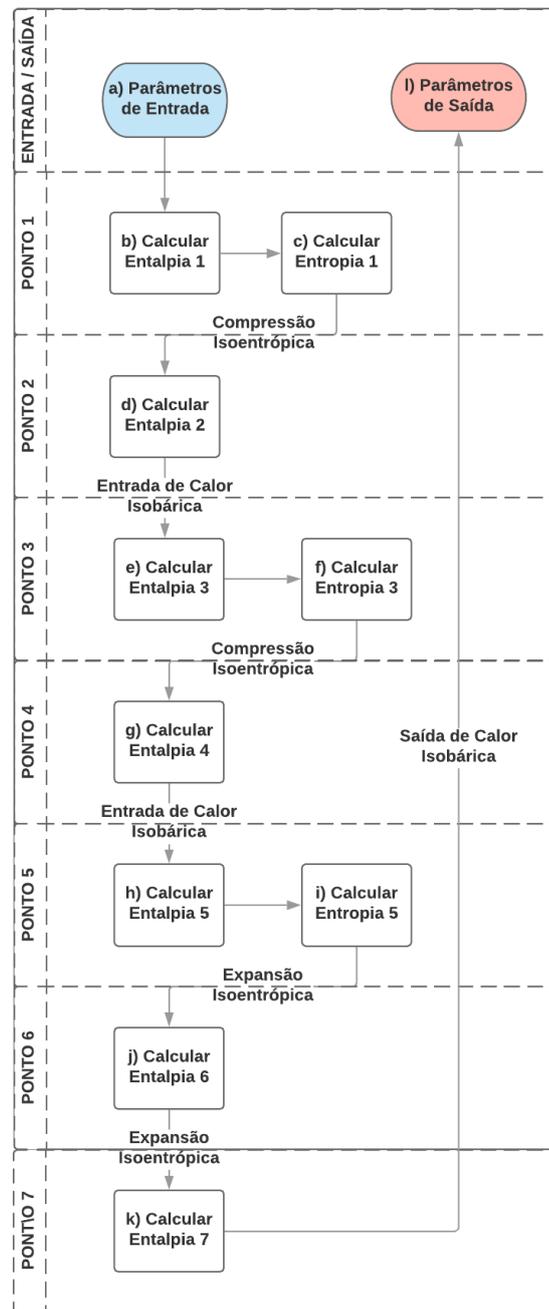
3.5 Função de otimização

Nos ciclos Rankine com reaquecimento e regenerativo, a escolha da pressão intermediária influencia no resultado global do ciclo. Utilizando os algoritmos estabelecidos para estes dois casos, é possível implementar uma função de busca para encontrar os valores de pressão intermediária que otimizam variáveis importantes, tais como o trabalho líquido específico e o rendimento. A otimização é realizada considerando os valores de temperatura⁸ e de pressão inferior e superior selecionados.

No Apêndice B é apresentada a função que foi aplicada para este propósito. Seu funcionamento é baseado na subdivisão do intervalo entre os patamares inferior e superior de pressão e na comparação dos resultados obtidos, utilizando cada subdivisão como o valor de pressão intermediária. O grau de precisão pode ser ajustado através da variável `divisoes`, sendo que valores maiores aumentam a precisão da otimização, em detrimento da velocidade de execução do algoritmo.

⁸ Neste trabalho, a temperatura de reaquecimento está sendo considerada igual à do aquecimento anterior

Figura 13 – Fluxograma do Ciclo Rankine regenerativo



Fonte: Autor

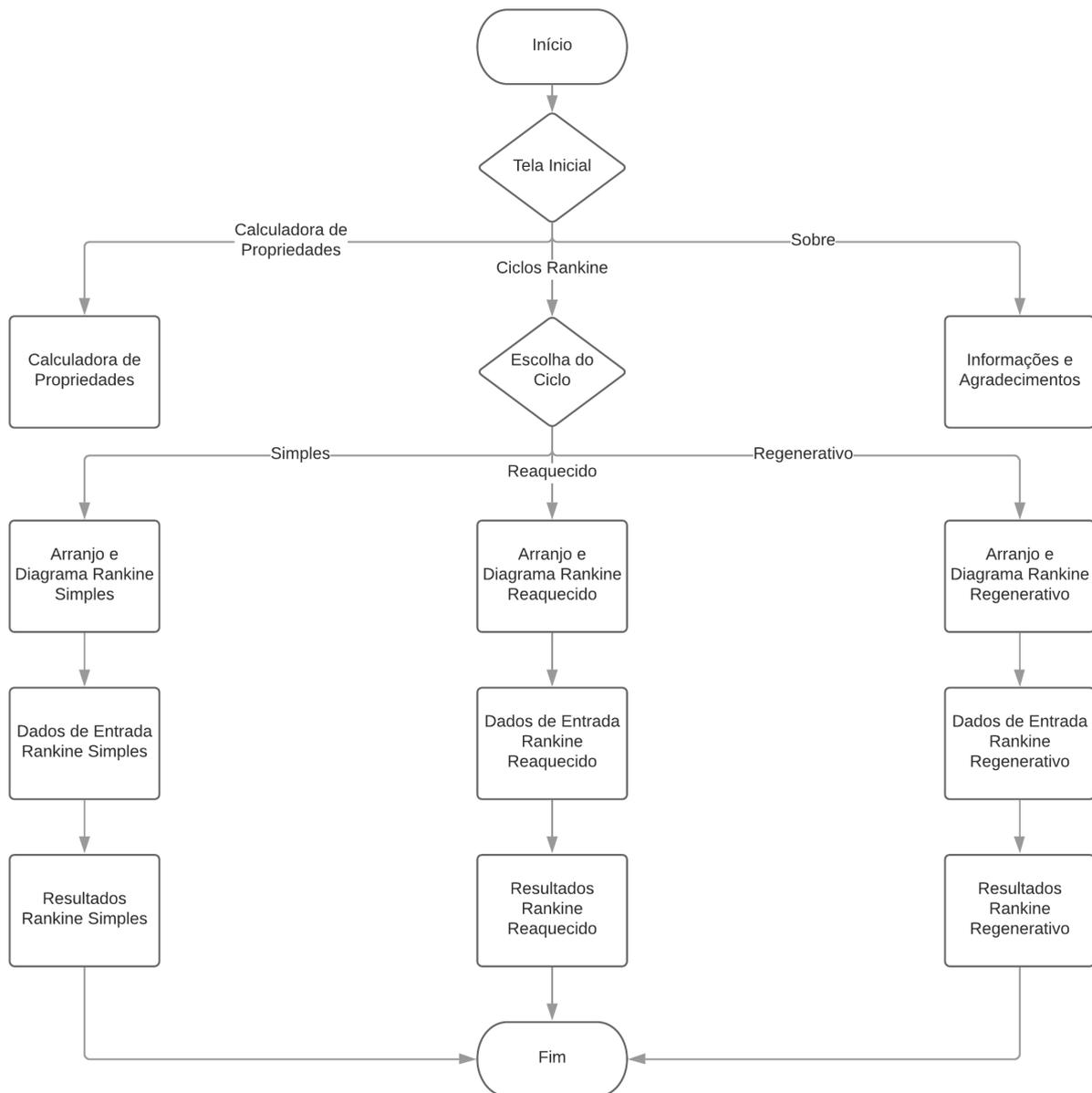
3.6 Implementação em dispositivos Android

O Java é a principal linguagem de programação utilizada no desenvolvimento de aplicações nativas para o sistema operacional Android. Assim, após a conclusão da etapa de implementação dos algoritmos na linguagem Python, os códigos foram adaptados para o Java, mantendo a mesma lógica de programação para o cálculo das variáveis termodinâmicas. No entanto, novas linhas de código foram adicionadas para permitir

interação do usuário com o aplicativo móvel, além da utilização da linguagem de marcação XML para a montagem dos layouts. Um exemplo de código XML, utilizado na criação de uma das interfaces do ciclo reaquecido, pode ser visto no Apêndice D.

Um fluxo de navegação foi elaborado com o propósito de obter uma plataforma intuitiva para os usuários e facilitar o processo de estudo exploratório. Este fluxo está apresentado na Figura 14.

Figura 14 – Fluxograma de navegação do aplicativo móvel



Fonte: Autor

Para o aplicativo móvel, além do dimensionamento dos ciclos Rankine, foi desenvolvida uma calculadora de propriedades termodinâmicas contendo os 122 fluidos disponíveis na função PropsSI. No próximo capítulo, são apresentadas as interfaces e a execução dos

softwares desenvolvidos. Adicionalmente, é realizada uma validação dos resultados dos algoritmos utilizando a literatura. Por fim, são discutidos os resultados obtidos pela função de otimização.

4 Resultados e Discussões

Nesse capítulo, são apresentados o funcionamento de cada algoritmo implementado, a comparação de seus resultados com os presentes na literatura e as análises propiciadas pela automatização desse processo, como a otimização na escolha da pressão intermediária para os ciclos regenerativo e reaquescido. Também é demonstrado o resultado final do aplicativo móvel desenvolvido e seus benefícios no estudo e análise das máquinas térmicas.

4.1 Aplicações em Python

Conforme planejado na Seção 3.4, foi desenvolvido um arquivo para cada um dos três ciclos trabalhados, contendo o código Python para o dimensionamento e a otimização dos ciclos. Em (SALES, 2019b), está disponível para *download* os seguintes arquivos .py¹:

- rankine_simples.py - Dimensionamento do ciclo Rankine simples
- rankine_reaquecido.py - Dimensionamento do ciclo Rankine reaquescido
- rankine_regenerativo.py - Dimensionamento do ciclo Rankine regenerativo

As instruções de instalação necessárias para executar essas aplicações no sistema operacional Windows estão no Apêndice C. A Figura 15 mostra um exemplo de execução para o caso do ciclo reaquescido, utilizando o ambiente de desenvolvimento oficial do Python.

Figura 15 – Entrada dos dados iniciais fornecidos pelo usuário

```

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\pietrosales\Documents\ciclos_rankine-master\rankine_reaquecido.py

**** CICLO RANKINE COM REAQUECIMENTO ****

----- Dados de Entrada -----

Insira a pressão inferior do sistema (kPa): 10
Insira a pressão intermediária do sistema (kPa): 400
Insira a pressão superior do sistema (kPa): 4000
Insira a temperatura superior do sistema (°C) [Insira -1 para Vapor Saturado]: |
Ln: 13 Col: 79

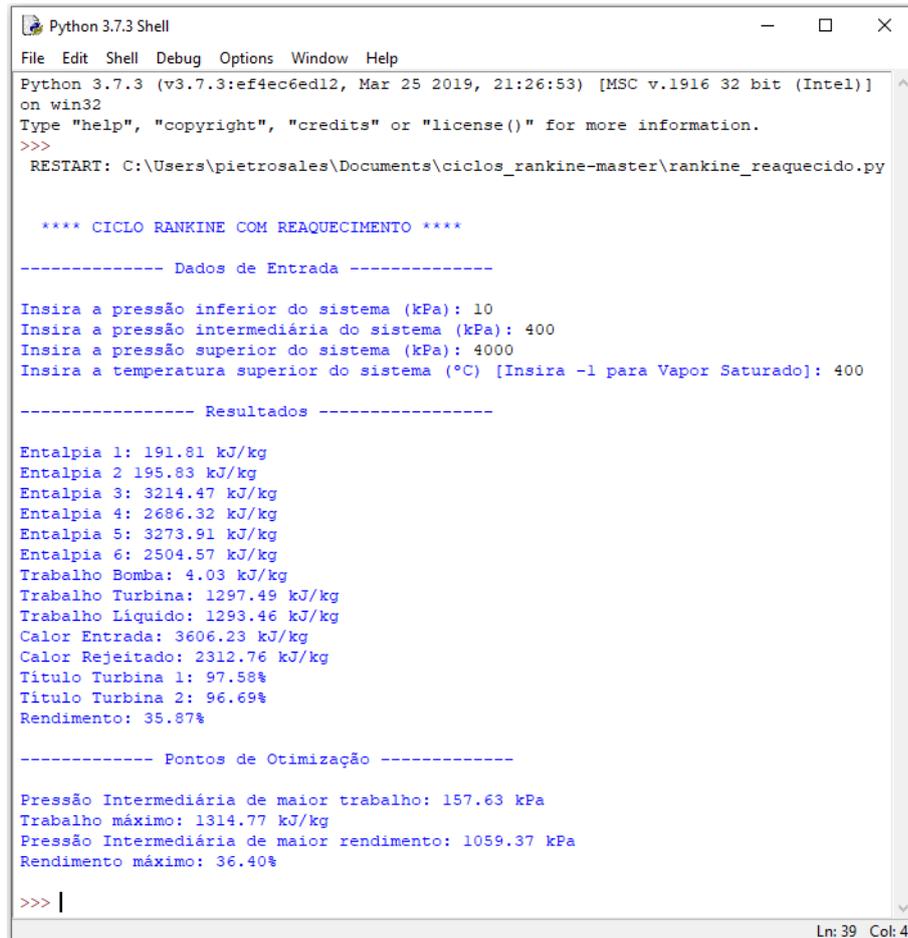
```

Fonte: Autor

¹ Extensão de arquivo referente aos códigos em Python

A primeira etapa da execução do programa é a solicitação dos parâmetros de entrada necessários ao dimensionamento do ciclo. Após a inserção manual dos dados, o algoritmo realiza os cálculos de dimensionamento e imprime na tela os resultados obtidos. Ainda, para os ciclos reaquecido e regenerativo, o código realiza uma busca para diversos valores de pressão intermediária e apresenta os pontos máximos de rendimento e geração de trabalho líquido. Os resultados completos podem ser vistos na Figura 16.

Figura 16 – Resultado do dimensionamento e indicação dos pontos de otimização



```

Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\pietrosales\Documents\ciclos_rankine-master\rankine_reaquecido.py

**** CICLO RANKINE COM REAQUECIMENTO ****

----- Dados de Entrada -----
Insira a pressão inferior do sistema (kPa): 10
Insira a pressão intermediária do sistema (kPa): 400
Insira a pressão superior do sistema (kPa): 4000
Insira a temperatura superior do sistema (°C) [Insira -1 para Vapor Saturado]: 400

----- Resultados -----

Entalpia 1: 191.81 kJ/kg
Entalpia 2 195.83 kJ/kg
Entalpia 3: 3214.47 kJ/kg
Entalpia 4: 2686.32 kJ/kg
Entalpia 5: 3273.91 kJ/kg
Entalpia 6: 2504.57 kJ/kg
Trabalho Bomba: 4.03 kJ/kg
Trabalho Turbina: 1297.49 kJ/kg
Trabalho Líquido: 1293.46 kJ/kg
Calor Entrada: 3606.23 kJ/kg
Calor Rejeitado: 2312.76 kJ/kg
Título Turbina 1: 97.58%
Título Turbina 2: 96.69%
Rendimento: 35.87%

----- Pontos de Otimização -----

Pressão Intermediária de maior trabalho: 157.63 kPa
Trabalho máximo: 1314.77 kJ/kg
Pressão Intermediária de maior rendimento: 1059.37 kPa
Rendimento máximo: 36.40%

>>> |
Ln: 39 Col: 4

```

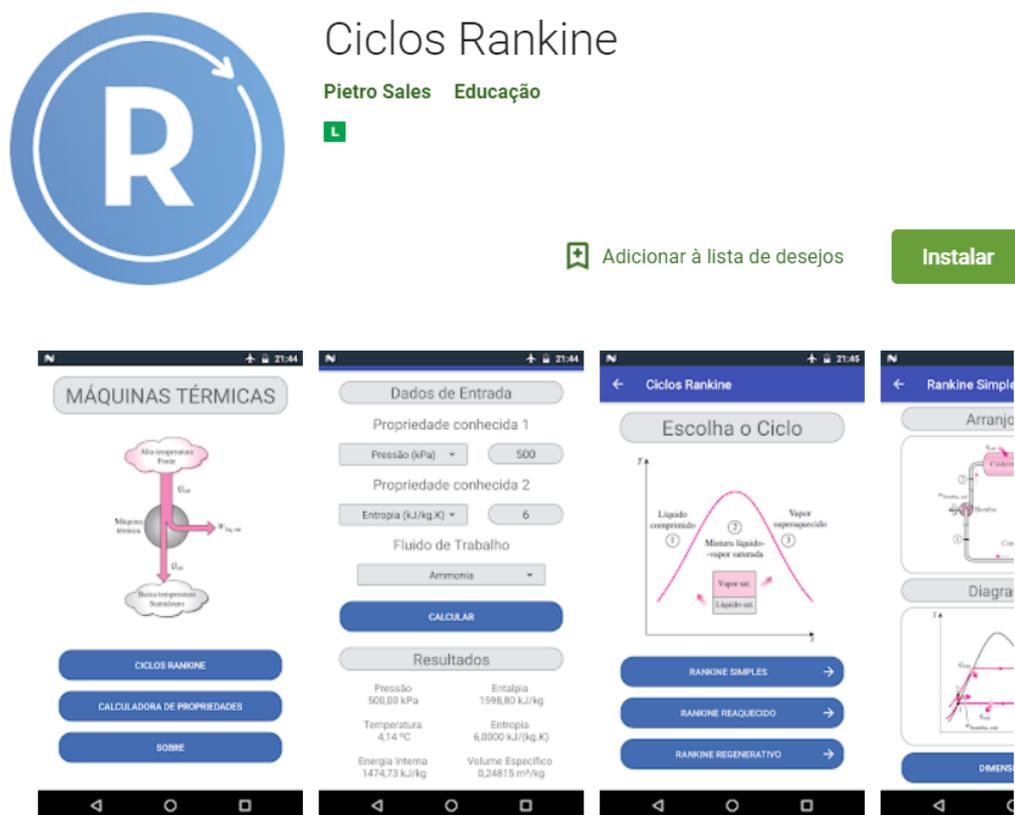
Fonte: Autor

4.2 Aplicativo Móvel

Assim como o programa anterior, o aplicativo móvel tem como objetivo calcular os resultados do dimensionamento para as três variações do ciclo Rankine e indicar os pontos de otimização para os ciclos regenerativo e reaquecido. Além disso, ele possui uma calculadora de propriedades termodinâmicas para 122 fluidos, podendo ser utilizada como ferramenta na análise de arranjos mais complexos. Os grandes diferenciais alcançados pelo desenvolvimento desta plataforma são sua portabilidade e sua interface amigável para os usuários.

O aplicativo está disponível gratuitamente na plataforma *Google Play Store*, podendo ser instalado pelo endereço (SALES, 2019a). Na Figura 17, é apresentada a página de instalação na loja do *Google*.

Figura 17 – Página do aplicativo na *Google Play Store*



Fonte: Autor

A seguir, são apresentadas as interfaces que foram desenvolvidas a partir da navegação previamente mapeada no fluxograma da Figura 14.

Na Figura 18, são exibidas a Tela Inicial (à esquerda), a Calculadora de Propriedades (centro) e a Escolha de Ciclo (à direita).

A Tela Inicial é composta por três opções: acessar as opções de ciclos Rankine, a Calculadora de Propriedades e a tela Sobre².

Na Calculadora de Propriedades, o usuário deve selecionar duas propriedades conhecidas (dentre as seis disponíveis) e informar os valores de cada uma delas. Além disso, deve selecionar um fluido de trabalho na lista contendo 122 opções. Assim, ao pressionar o botão “Calcular”, o aplicativo apresenta os resultados para as seis propriedades disponíveis³.

² A tela Sobre contém informações sobre o projeto, endereço para o código-fonte, referências e licenças utilizadas

³ Devido ao tamanho do dispositivo, pode ser necessário deslizar a tela para ver o resultado completo

Figura 18 – Tela inicial (à esquerda), calculadora (centro) e opções de arranjo (à direita)



Fonte: Autor

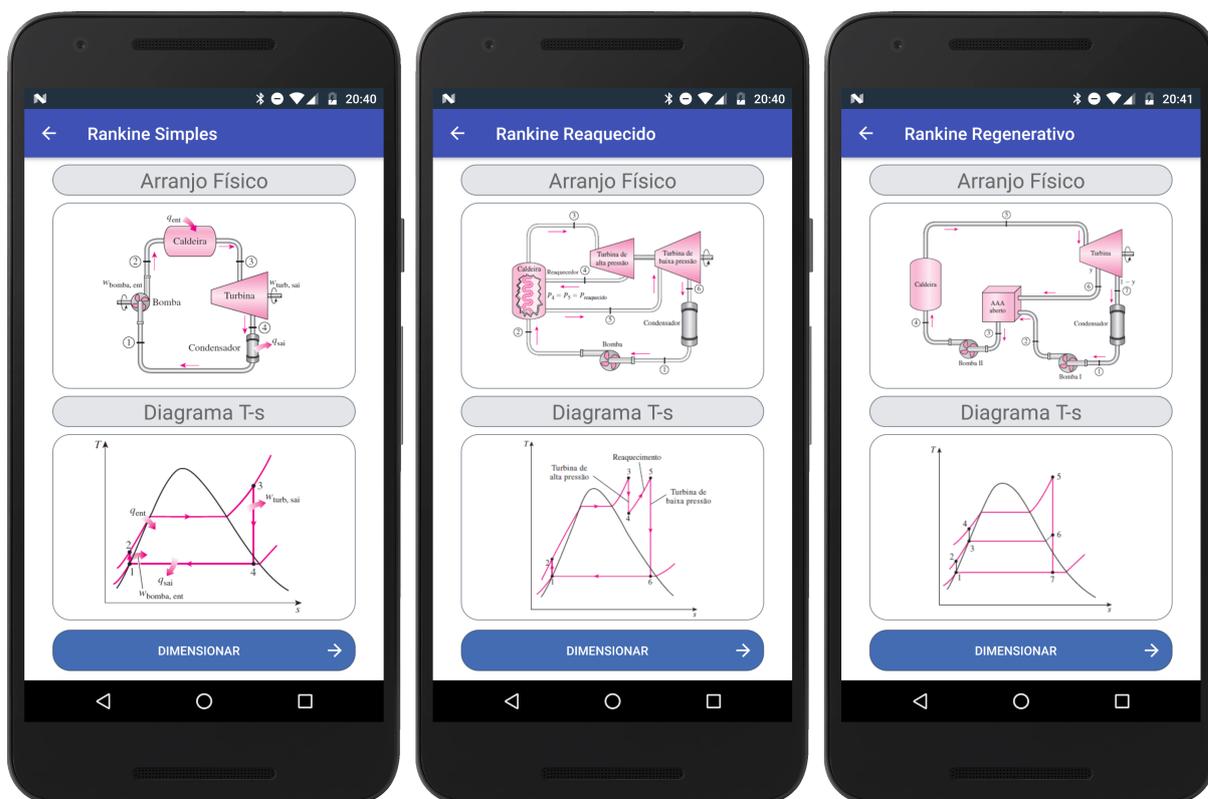
A tela de seleção de ciclo contém as três opções de arranjo discutidas neste trabalho. Ao selecionar uma delas, o usuário é enviado para a tela de Arranjo Físico e Diagrama T-s correspondente à sua escolha.

Na Figura 19, é possível observar os arranjos físicos e os diagramas T-s para os ciclos Rankine simples, reaquecido e regenerativo respectivamente. Essas telas foram postas antes da tela de inserção dos dados de entrada, de forma a apresentar ao usuário a convenção utilizada para nomear cada ponto do ciclo. Além disso, as imagens ajudam o estudante a recordar os componentes de cada arranjo, o sentido de escoamento do fluido, os processos com propriedades constantes e as regiões de transferência de calor e trabalho.

O passo seguinte é a escolha dos dados de entrada. A Figura 20 exibe a interface de inserção dos patamares de pressão e temperatura para os ciclos Rankine simples, reaquecido e regenerativo respectivamente. Caso o usuário deseje selecionar a temperatura máxima do sistema com o valor correspondente ao ponto de vapor saturado, a opção “Vapor Saturado na Caldeira” pode ser marcada. Assim, fica desabilitada a inserção manual para a temperatura superior, ficando a cargo do próprio sistema determinar este valor.

Após o preenchimento completo dos dados necessários, o usuário pressiona o botão “Calcular” e é enviado para a tela final, contendo os resultados do dimensionamento e

Figura 19 – Arranjos físicos e Diagramas T-s



Fonte: Autor

os valores das propriedades termodinâmicas para cada estado de equilíbrio. A Figura 21 apresenta os resultados para um exemplo de dimensionamento realizado para o arranjo com reaquecimento. Por fim, o aplicativo informa os pontos de otimização para a pressão intermediária, buscando maximizar o rendimento ou o trabalho líquido do sistema.

4.3 Validação dos resultados

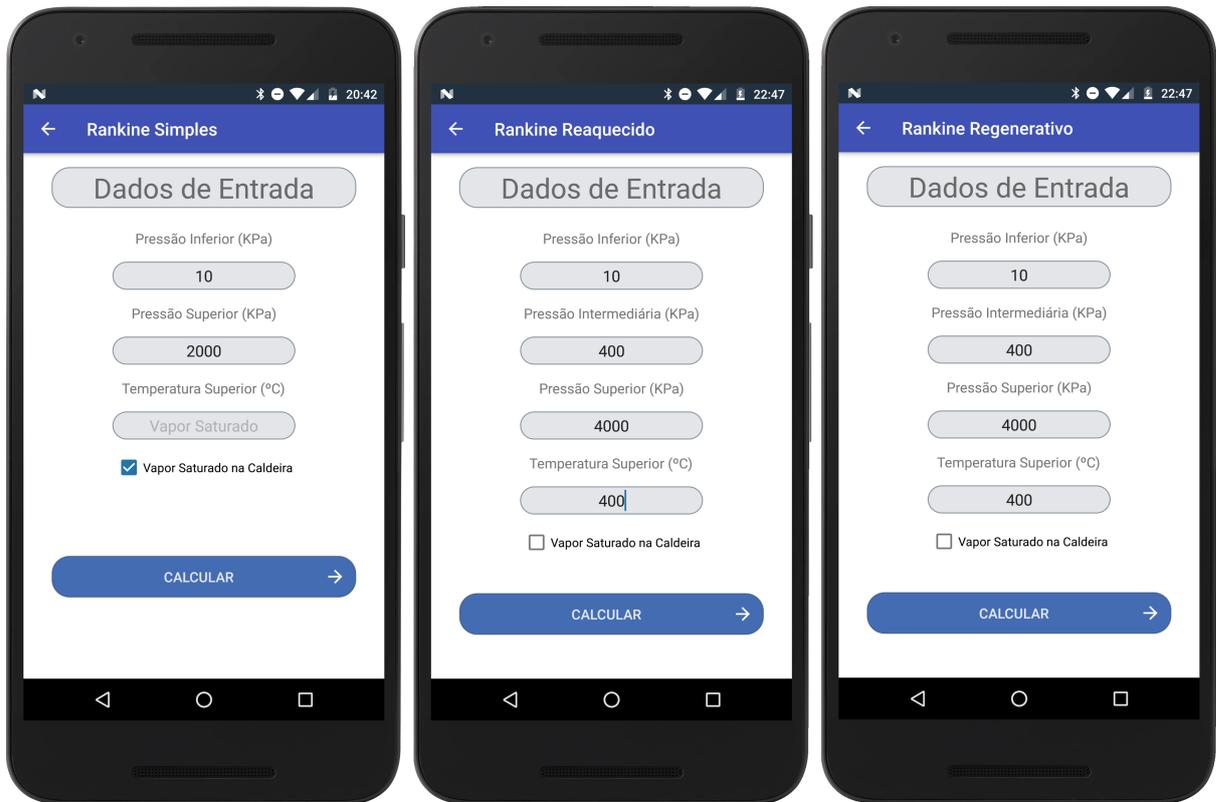
Nessa seção, é realizada uma comparação entre os resultados dos algoritmos e de alguns exemplos resolvidos encontrados na literatura (SONNTAG; WYLEN; BORGNAKKE, 2003).

4.3.1 Ciclo rankine simples

O primeiro exemplo consiste de um ciclo Rankine simples operando entre as pressões de 10 kPa e 2 MPa. O valor de temperatura do fluido na saída da caldeira é referente à temperatura de vapor saturado a 2 MPa. O fluido de trabalho é a água.

Assim, os parâmetros de entrada do algoritmo estão apresentados na Tabela 1.

Figura 20 – Seleção de dados de entrada para os três arranjos



Fonte: Autor

Tabela 1 – Parâmetros de entrada Rankine simples

Parâmetro	Valor
Pressão Inferior	10 kPa
Pressão Superior	2 MPa
Temperatura Superior	Saturação à 2MPa

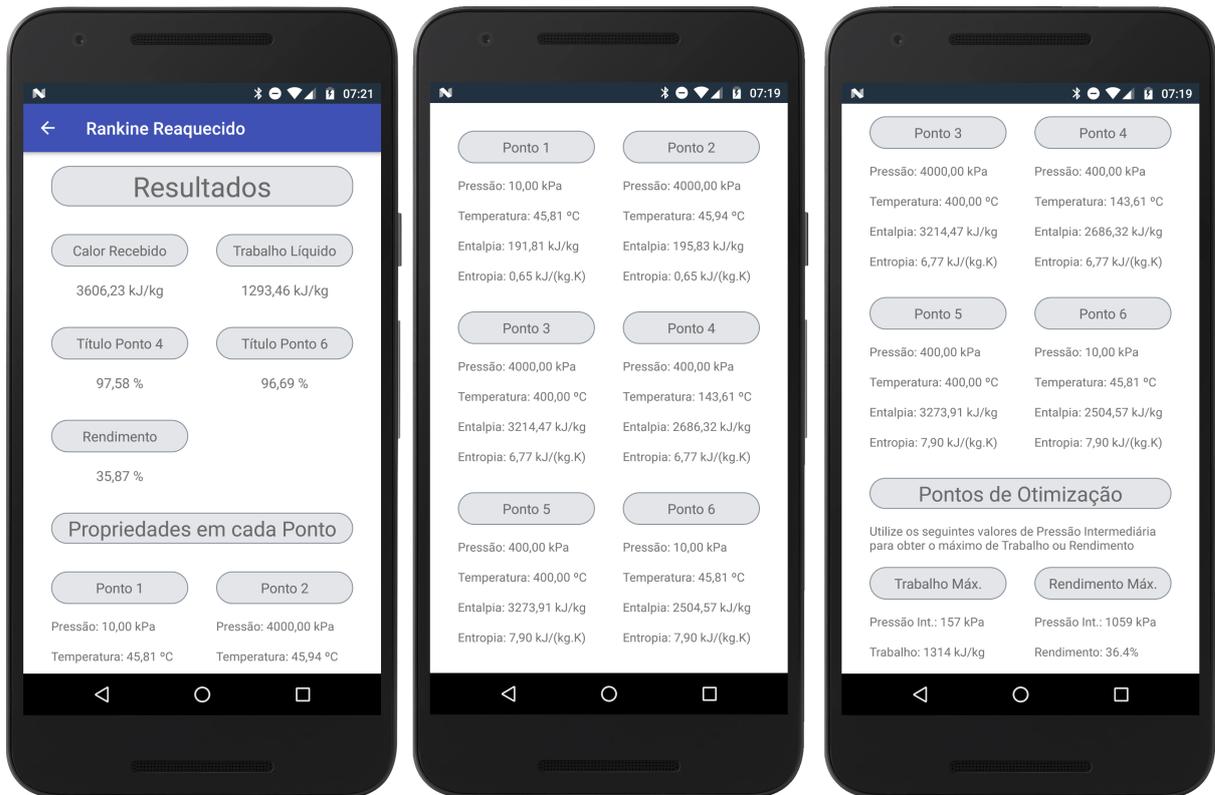
A Tabela 2 apresenta lado a lado os resultados encontrados no livro escolhido e pelo programa *rankine_simple.py*. Os valores de entalpia, trabalho e calor estão em kJ/kg . Como pode ser observado, a diferença entre os valores é desprezível e pode ser atribuída a pequenas incertezas e arredondamentos.

4.3.2 Ciclo Rankine com reaquecimento

O segundo exemplo diz respeito a um ciclo Rankine com reaquecimento que opera entre as pressões de 10 kPa e 4 MPa. A pressão intermediária na qual o fluido é reaquecido é de 400 kPa. A temperatura máxima alcançada pelo fluido após o aquecimento (e o reaquecimento) na caldeira é de 400 °C. O fluido de trabalho é a água.

Assim, os parâmetros de entrada do algoritmo estão apresentados na tabela 3.

Figura 21 – Tela de resultados do ciclo Rankine reaquecido



Fonte: Autor

Tabela 2 – Resultado Rankine simples

Parâmetro	Livro	Algoritmo	Diferença
Entalpia 1	191,8	191,8	0%
Entalpia 2	193,8	193,8	0%
Entalpia 3	2799,5	2798,3	0,04%
Entalpia 4	2007,5	2006,6	0,04%
Trabalho Bomba	2,0	2,0	0%
Trabalho Turbina	792,0	791,7	0,04%
Trabalho Líquido	790,0	789,7	0,04%
Calor de Entrada	2605,7	2604,5	0,05%
Calor Rejeitado	1815,7	1814,8	0,05%
Título Turbina	75,88%	75,87%	0,01%
Rendimento	30,3%	30,3%	0%

Fonte: (SONNTAG; WYLEN; BORGNAKKE, 2003)

Análogo ao caso anterior, a Tabela 4 mostra que o programa *rankine_reaquecido.py* apresentou uma diferença desprezível em relação aos valores encontrados na literatura. Os valores de entalpia, trabalho e calor estão em kJ/kg .

Tabela 3 – Entrada Reaquecido e Regenerativo

Parâmetro	Valor
Pressão Inferior	10 kPa
Pressão Intermediária	400 kPa
Pressão Superior	4 MPa
Temperatura Superior	400 °C

Tabela 4 – Resultado Rankine com reaquecimento

Parâmetro	Livro	Algoritmo	Diferença
Entalpia 1	191,8	191,8	0%
Entalpia 2	195,8	195,8	0%
Entalpia 3	3213,6	3214,5	0,03%
Entalpia 4	2685,6	2686,3	0,03%
Entalpia 5	3273,4	3273,9	0,02%
Entalpia 6	2504,3	2504,6	0,01%
Trabalho Bomba	4,0	4,0	0%
Trabalho Turbina	1297,1	1297,5	0,03%
Calor Entrada	3605,6	3606,2	0,02%
Calor Rejeitado	-	2312,8	-
Título Turbina 1	97,52%	97,58%	0,06%
Título Turbina 2	96,64%	96,69%	0,05%
Rendimento	35,9%	35,9%	0%

Fonte: (SONNTAG; WYLEN; BORGNAKKE, 2003)

4.3.3 Ciclo Rankine regenerativo

O terceiro exemplo diz respeito a um ciclo Rankine regenerativo operando sobre as mesmas circunstâncias apresentadas no exemplo anterior.

Assim, os parâmetros de entrada do algoritmo são os valores apresentados na Tabela 3.

Novamente, a Tabela 5 corrobora com a validação do algoritmo, visto que a diferença entre os resultados são desprezíveis. Os valores de entalpia, trabalho e calor estão em kJ/kg .

4.4 Otimização da pressão intermediária

Para os ciclos reaquecido e regenerativo, além dos limites de pressão e temperatura disponíveis, é necessário escolher o patamar intermediário de pressão. Essa otimização pode ser realizada por meio de uma busca iterativa, na qual o algoritmo testa diversos valores de pressão intermediária e retorna os melhores resultados de rendimento e de

Tabela 5 – Resultado Rankine regenerativo

Parâmetro	Livro	Algoritmo	Diferença
Entalpia 1	191,8	191,8	0%
Entalpia 2	192,2	192,2	0%
Entalpia 3	604,7	604,7	0%
Entalpia 4	608,6	608,6	0%
Entalpia 5	3213,6	3214,5	0,03%
Entalpia 6	2685,6	2686,3	0,03%
Entalpia 7	2144,1	2144,5	0,02%
Trabalho Bomba	4,3	4,3	0%
Trabalho Turbina	979,9	980,4	0,05%
Trabalho Líquido	975,7	976,1	0,04%
Calor Entrada	2605,0	2605,9	0,03%
Calor Rejeitado	-	1629,8	-
Título Turbina 1	-	97,6%	-
Título Turbina 2	-	81,6%	-
Fração de extração	16,54%	16,5%	0%
Rendimento	37,50%	37,5%	0%

Fonte: (SONNTAG; WYLEN; BORGNAKKE, 2003)

geração de trabalho líquido específico. Como apresentado a seguir, o ponto de máximo rendimento, em geral, não coincide com o ponto máximo de trabalho líquido. Portanto, a escolha da posição intermediária ideal depende das necessidades do projeto.

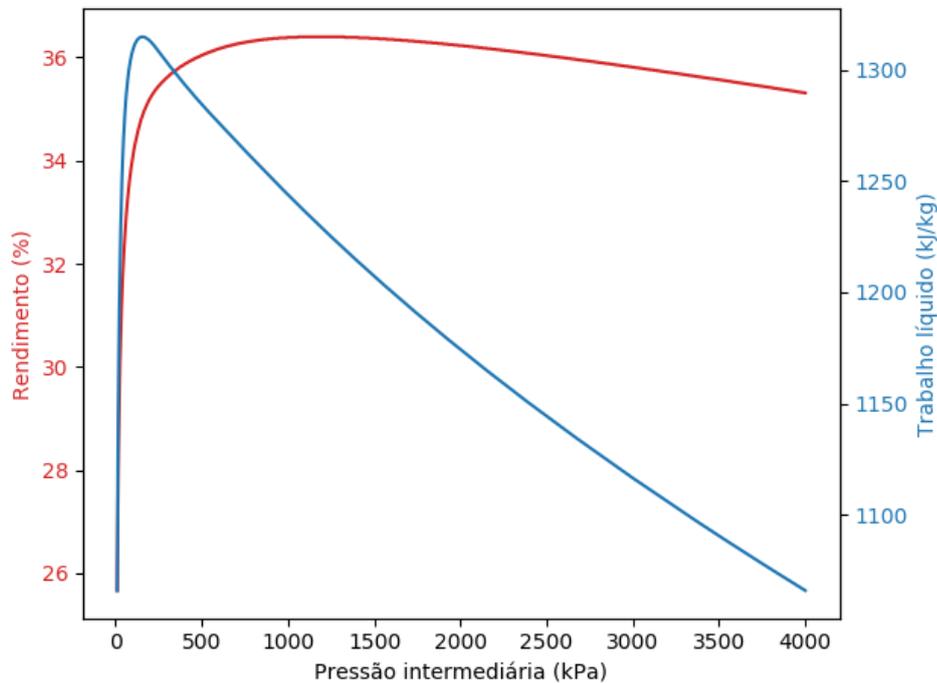
Nas Figuras 22 e 23, foram sobrepostas as curvas de rendimento e trabalho líquido, ambas em função da pressão. Os patamares de pressão e temperatura utilizados foram os mesmos da Tabela 3, exceto para a pressão intermediária, que foi a variável independente do processo de otimização. Em ambos os gráficos, o domínio dos valores de pressão intermediária está compreendido entre os valores de baixa e alta pressão, incluindo os extremos.

A Figura 22 representa o processo de otimização para o ciclo Rankine com reaquecimento⁴. Pode-se observar que os melhores resultados para trabalho líquido estão concentrados aplicando o reaquecimento em valores de pressão próximos ao patamar inferior de pressões. Entretanto, é importante observar a influência sobre o título, visto que a elevação do valor desta variável é o principal motivo de utilizar o processo de reaquecimento.

Na Figura 23, é apresentada a curva de otimização para o ciclo Rankine regenerativo. Como esperado, o trabalho reduz conforme o ponto de extração se aproxima da pressão superior. Isso acontece, pois quanto antes for realizada a extração no processo de expansão na turbina, menor é a região de geração de trabalho utilizando a vazão mássica total

⁴ Considerando a temperatura de reaquecimento igual à do aquecimento anterior

Figura 22 – Otimização para o Ciclo Rankine Reaquecido



Fonte: Autor

disponível. Consta-se, assim, que o arranjo regenerativo propõe elevar o rendimento em detrimento da produção de trabalho líquido.

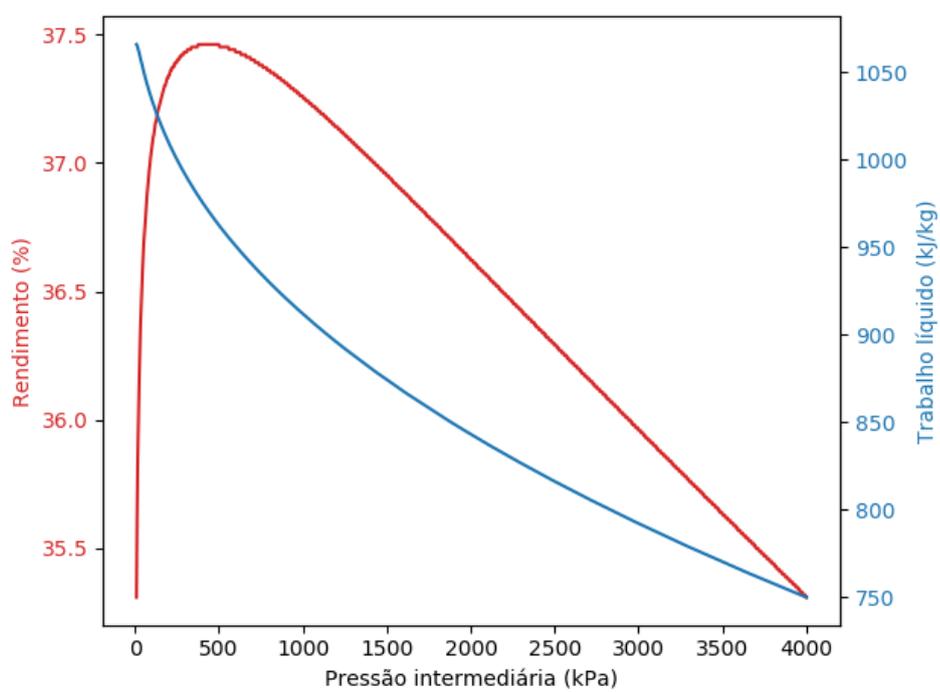
Em respeito ao rendimento, nota-se no exemplo da Figura 23 que há ganhos significativos ao selecionar a pressão de extração próxima à pressão inferior do sistema, em torno de 500 kPa.

Portanto, combinando as duas observações anteriores para o caso analisado, um bom projeto de ciclo Rankine regenerativo deve selecionar como pressão intermediária um valor na região próxima à pressão inferior do sistema.

É interessante destacar que, no ciclo regenerativo, ao selecionar a pressão intermediária igual à pressão inferior, a fração mássica extraída⁵ torna-se nula. Assim, o resultado obtido é equivalente ao de um Rankine simples operando entre os mesmos patamares de pressão e temperatura. Esta constatação não é verdade para a modelagem utilizada no ciclo com reaquecimento, visto que a etapa de reaquecimento continuaria presente.

⁵ Ver Equação 3.10

Figura 23 – Otimização para o Ciclo Rankine Regenerativo



Fonte: Autor

5 Conclusão

Este trabalho visou automatizar o processo de dimensionamento de ciclos Rankine ideais por meio da computação, considerando principalmente sua aplicação como instrumento no ensino de engenharia.

Os softwares desenvolvidos cumpriram o objetivo traçado, determinando os parâmetros relevantes no dimensionamento deste tipo de máquina térmica. Os resultados foram validados através de comparações com exemplos presentes na literatura, comprovando sua eficácia como ferramenta para o cálculo desse tipo de ciclo termodinâmico.

A automatização deste processo se mostrou altamente benéfica, sobretudo devido à capacidade de realizar diversos testes e colher resultados em um curto espaço de tempo. Como consequência, foi possível desenvolver a função de otimização, a qual aciona a rotina de dimensionamento de forma iterativa e busca os pontos máximos de trabalho líquido e rendimento.

Além do uso didático por alunos e professores, o aplicativo móvel pode ser utilizado por engenheiros para efetuar um dimensionamento inicial em projetos de máquinas térmicas. A velocidade e a portabilidade fornecida pela plataforma facilita a realização dessas análises, de modo que o engenheiro possa efetuar testes preliminares antes de passar para uma modelagem com o uso de softwares profissionais.

O projeto se restringiu a tratar de dispositivos ideais, de modo que os resultados numéricos não devem ser considerados em um dimensionamento real. No entanto, a ferramenta pode ser utilizada para realizar análises comparativas entre diferentes patamares de pressão e temperatura.

Além disso, este trabalho limitou-se a dimensionar e otimizar máquinas com apenas um estágio de reaquecimento ou regeneração. Entretanto, máquinas reais podem apresentar mais de um estágio intermediário. Neste caso, a otimização do ciclo inclui o número de estágios, os patamares intermediários de pressão e considerações financeiras de instalação e manutenção.

Baseado nas limitações indicadas, sugestões naturais para trabalhos futuros incluem a implementação de mais estágios intermediários e considerações de ineficiências e irreversibilidades nos dispositivos, aproximando os resultados das condições reais de funcionamento. Com isso, o aplicativo móvel pode evoluir para um produto comercial a ser utilizado na elaboração de projetos industriais.

Ainda, o aplicativo pode ser complementado adicionando novos ciclos de potência e ciclos de refrigeração, formando, assim, uma plataforma completa para análises de ciclos

termodinâmicos.

Por fim, espera-se que a metodologia de desenvolvimento e os resultados obtidos neste trabalho sejam utilizados como forma de estimular o uso da programação na solução de problemas de engenharia e no desenvolvimento de produtos relacionados.

Referências

- AMORIM, J. d. A. et al. Educação em engenharia: o desenvolvimento de um aplicativo de autoria para a elaboração de mapas conceituais e hipertextos. [sn], 2005. Citado 2 vezes nas páginas 31 e 32.
- AZIMIAN, A.; ARRIAGADA, J.; ASSADI, M. Generation of steam tables using artificial neural networks. *Heat Transfer Engineering*, Taylor Francis, v. 25, n. 2, p. 41–51, 2004. ISSN 1521-0537. Citado na página 21.
- BACHEGA, M. C. Desenvolvimento de um software para auxílio no projeto e análise de elementos de máquinas. 2018. Citado na página 32.
- BELL, I. H. *CoolProp: An open-source thermophysical property library*. 2010. Acesso em 12 de fevereiro de 2019. Disponível em: <<http://www.coolprop.org/>>. Citado na página 37.
- BELL, I. H. *CoolProp: An open-source thermophysical property library*. 2010. Acesso em 12 de fevereiro de 2019. Disponível em: <<http://www.coolprop.org/coolprop/HighLevelAPI.html#propssi-function>>. Citado na página 37.
- BELL, I. H. et al. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. *Industrial & Engineering Chemistry Research*, v. 53, n. 6, p. 2498–2508, 2014. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/ie4033999>>. Citado 2 vezes nas páginas 21 e 37.
- BOLES, M.; ÇENGEL, Y. *Termodinâmica. 7ª edição*. Bookman Editora, 2013. ISBN 9788580552010. Disponível em: <<https://books.google.com.br/books?id=jAI5AgAAQBAJ>>. Citado 15 vezes nas páginas 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35 e 36.
- BRAGA, W. Uso de applets java no ensino de engenharia. In: *CONGRESSO BRASILEIRO*. [S.l.: s.n.], 2000. Citado na página 32.
- CHELLA, M. T. et al. Ambiente de robótica para aplicações educacionais com superlogo. [sn], 2002. Citado na página 31.
- DICKES, R. et al. Orcmkit: an open-source library for organic rankine cycle modelling and analysis. In: *The 29th International Conference on Efficiency, Cost, Optimization, Simulation and Environmental impact of energy systems*. [S.l.: s.n.], 2016. p. 1–14. Citado na página 14.
- ISLAM, R.; ISLAM, R.; MAZUMDER, T. Mobile application and its global impact. *International Journal of Engineering & Technology (IJEST)*, v. 10, n. 6, p. 72–78, 2010. Citado 2 vezes nas páginas 15 e 31.
- JENG, Y.-L. et al. The add-on impact of mobile applications in learning strategies: A review study. *Journal of Educational Technology & Society*, JSTOR, v. 13, n. 3, p. 3–11, 2010. Citado 2 vezes nas páginas 15 e 31.

- JOU, M.; LIN, Y.-T.; TSAI, H.-C. Mobile app for motivation to learning: an engineering case. *Interactive Learning Environments*, Taylor & Francis, v. 24, n. 8, p. 2048–2057, 2016. Citado 2 vezes nas páginas 31 e 32.
- MORAN, M. J.; SHAPIRO, H. N.; BOETTNER, D. D. *Princípios de termodinâmica para engenharia*. [S.l.]: Grupo Gen-LTC, 2013. Citado na página 19.
- PYTHON. 2019. Acesso em 12 de março de 2019. Disponível em: <<https://www.python.org/>>. Citado na página 78.
- QUOILIN, S. et al. Thermocycle: A modelica library for the simulation of thermodynamic systems. In: LINKÖPING UNIVERSITY ELECTRONIC PRESS. *10th international Modelica conference*. [S.l.], 2014. Citado na página 14.
- RAO, A. *Sustainable Energy Conversion for Electricity and Coproducts: Principles, Technologies, and Equipment*. [S.l.]: John Wiley & Sons, 2015. Citado 2 vezes nas páginas 14 e 26.
- SALES, P. *Ciclos Rankine: Apps no Google Play*. 2019. Acesso em 31 de março de 2019. Disponível em: <https://play.google.com/store/apps/details?id=com.pietrosales.ciclos_rankine>. Citado na página 51.
- SALES, P. *Desenvolvimento de aplicativo móvel para dimensionamento de ciclos Rankine: Códigos-fonte*. 2019. Acesso em 12 de março de 2019. Disponível em: <https://github.com/PietroSas/ciclos_rankine>. Citado 5 vezes nas páginas 34, 42, 44, 45 e 49.
- SALSA, T. O. Aplicativo didático de dimensionamento de armaduras em estruturas de concreto armado. *Revista Técnico-Científica*, n. 10, 2018. Citado na página 32.
- SONNTAG, R.; WYLEN, G. V.; BORGNAKKE, C. *FUNDAMENTOS DA TERMODINAMICA: TRADUÇÃO DA 6ª EDIÇÃO AMERICANA*. Edgard Blucher, 2003. ISBN 9788521201670. Disponível em: <<https://books.google.com.br/books?id=QBlkAAAACAAJ>>. Citado 10 vezes nas páginas 20, 22, 24, 25, 29, 35, 53, 55, 56 e 57.
- SOTOMONTE, C. A. R. Otimização multiobjetivo para a seleção de fluidos de trabalho e parâmetros de projeto no ciclo rankine orgânico. 2015. Citado na página 14.
- SOUSA, R. P. de et al. *Teorias e práticas em tecnologias educacionais*. [S.l.]: SciELO-EDUEPB, 2016. Citado na página 31.
- SUBRAMANYA, S.; FARAHANI, A. Point-of-view article on: Design of a smartphone app for learning concepts in mathematics and engineering. *International Journal of Innovation Science*, Emerald Group Publishing Limited, v. 4, n. 3, p. 173–184, 2012. Citado na página 31.
- ZOHURI, B.; MCDANIEL, P. *Thermodynamics in nuclear power plant systems*. [S.l.: s.n.], 2015. 1-724 p. Citado na página 35.

Apêndices

APÊNDICE A – Códigos em Python

A.1 Ciclo Rankine Simples

```

1  from CoolProp.CoolProp import PropsSI
2
3  def rankine_simples(p1, p3, t3, fluido):
4
5      if t3 == 272.15:
6          t3 = PropsSI('T', 'P', p3, 'Q', 1, fluido)
7
8      #Estado 1
9      h1 = PropsSI('H', 'P|liquid', p1, 'Q', 0, fluido) #Entalpia 1
10     s1 = PropsSI('S', 'P|liquid', p1, 'Q', 0, fluido) #Entropia 1
11
12     #Estado 2
13     p2 = p3 #Compressão
14     s2 = s1 #Processo Isoentrópico
15     h2 = PropsSI('H', 'P|liquid', p2, 'S', s2, fluido) #Entalpia 2
16
17     #Estado 3
18     h3 = PropsSI('H', 'P|gas', p3, 'T', t3, fluido)
19     s3 = PropsSI('S', 'P|gas', p3, 'T', t3, fluido)
20
21     #Estado 4
22     s4 = s3
23     p4 = p1
24     h4 = PropsSI('H', 'P|gas', p4, 'S', s4, fluido)
25
26     #Para cálculo do título:
27     s4l = PropsSI('S', 'P|liquid', p4, 'Q', 0, fluido)
28     s4v = PropsSI('S', 'P|gas', p4, 'Q', 1, fluido)
29
30     if (s4 < s4v):
31         x4 = (s4 - s4l)/(s4v - s4l)
32     else:
33         x4 = 'Saída da turbina superaquecida'

```

```
34
35     #Parâmetros de Saída
36     wb = h2 - h1 #Trabalho na bomba
37     wt = h3 - h4 #Trabalho na turbina
38     w = wt - wb #Trabalho líquido
39     qh = h3 - h2 #Calor de Entrada
40     ql = h4 - h1 #Calor rejeitado
41     n = (qh - ql)/qh #Rendimento
42
43     #Padronizando unidades
44     h1 = round(h1/1000, 2)
45     h2 = round(h2/1000, 2)
46     h3 = round(h3/1000, 2)
47     h4 = round(h4/1000, 2)
48     wt = round(wt/1000, 2)
49     wb = round(wb/1000, 2)
50     w = round(w/1000, 2)
51     qh = round(qh/1000, 2)
52     ql = round(ql/1000, 2)
53     if type(x4) == float:
54         x4 = round(x4*100, 2)
55     n = round(n*100, 2)
56
57     return {'h1':h1, 'h2':h2, 'h3':h3, 'h4':h4, 'wt':wt,
58           'w':w, 'wb':wb, 'qh':qh, 'ql':ql, 'x4':x4,
59           'n':n}
60
61     #Função Principal - Parâmetros de Entrada
62     p1 = float(input("Insira a pressão inferior do sistema (kPa): "))
63     p3 = float(input("Insira a pressão superior do sistema (kPa): "))
64     t3 = float(input("Insira a temperatura superior do sistema (°C): "))
65     fluido = 'Water'
66
67     p1 *= 1000
68     p3 *= 1000
69     t3 += 273.15
70
71     r = rankine_simples(p1, p3, t3, fluido) #Resultados
72
73     #Imprimindo os resultados
74     print("\n----- Resultados ----- \n")
```

```
75 print(f"""Entalpia 1: {r['h1']} kJ/kg
76 Entalpia 2 {r['h2']} kJ/kg
77 Entalpia 3: {r['h3']} kJ/kg
78 Entalpia 4: {r['h4']} kJ/kg
79 Trabalho Bomba: {r['wb']} kJ/kg
80 Trabalho Turbina: {r['wt']} kJ/kg
81 Trabalho Líquido: {r['w']} kJ/kg
82 Calor Entrada: {r['qh']} kJ/kg
83 Calor Rejeitado: {r['ql']} kJ/kg
84 Título Turbina: {r['x4']}%
85 Rendimento: {r['n']}%\n""")
```

A.2 Ciclo Rankine Reaquecido

```
1 from CoolProp.CoolProp import PropsSI
2
3 def rankine_reaquecido(p1, p3, p4, t3, fluido):
4
5     if t3 == 272.15:
6         t3 = PropsSI('T', 'P', p3, 'Q', 1, fluido)
7
8         #Estado 1
9         h1 = PropsSI('H', 'P|liquid', p1, 'Q', 0, fluido) #Entalpia 1
10        s1 = PropsSI('S', 'P|liquid', p1, 'Q', 0, fluido) #Entropia 1
11
12        #Estado 2
13        p2 = p3 #Compressão
14        s2 = s1 #Processo Isoentrópico
15        h2 = PropsSI('H', 'P|liquid', p2, 'S', s2, fluido) #Entalpia 2
16
17        #Estado 3
18        h3 = PropsSI('H', 'P|gas', p3, 'T', t3, fluido)
19        s3 = PropsSI('S', 'P|gas', p3, 'T', t3, fluido)
20
21        #Estado 4
22        s4 = s3
23        h4 = PropsSI('H', 'P|gas', p4, 'S', s4, fluido)
24
25        #Para cálculo do título:
26        s4l = PropsSI('S', 'P|liquid', p4, 'Q', 0, fluido)
27        s4v = PropsSI('S', 'P|gas', p4, 'Q', 1, fluido)
28
29        if (s4 < s4v):
30            x4 = (s4 - s4l)/(s4v - s4l)
31        else:
32            x4 = 'Saída da turbina superaquecida'
33
34        #Estado 5
35        p5 = p4
36        t5 = t3
37        h5 = PropsSI('H', 'P|gas', p5, 'T', t5, fluido)
38        s5 = PropsSI('S', 'P|gas', p5, 'T', t5, fluido)
```

```
39
40     #Estado 6
41     s6 = s5
42     p6 = p1
43     h6 = PropsSI('H', 'P|gas', p6, 'S', s6, fluido)
44
45     #Para cálculo do título:
46     s6l = PropsSI('S', 'P|liquid', p6, 'Q', 0, fluido)
47     s6v = PropsSI('S', 'P|gas', p6, 'Q', 1, fluido)
48
49     if (s6 < s6v):
50         x6 = (s6 - s6l)/(s6v - s6l)
51     else:
52         x6 = 'Saída da turbina superaquecida'
53
54     #Parâmetros de Saída
55     wb = h2 - h1 #Trabalho na bomba
56     wt1 = h3 - h4 #Trabalho na turbina 1
57     wt2 = h5 - h6 #Trabalho na turbina 2
58     wt = wt1 + wt2 #Trabalho total realizado
59     w = wt - wb #Trabalho Líquido
60     qh = (h3 - h2) + (h5 - h4) #Calor de Entrada
61     ql = h6 - h1 #Calor rejeitado
62     n = (qh - ql)/qh #Rendimento
63
64     #Padronizando unidades
65     h1 = round(h1/1000, 2)
66     h2 = round(h2/1000, 2)
67     h3 = round(h3/1000, 2)
68     h4 = round(h4/1000, 2)
69     h5 = round(h5/1000, 2)
70     h6 = round(h6/1000, 2)
71     wt1 = round(wt1/1000, 2)
72     wt2 = round(wt2/1000, 2)
73     wt = round(wt/1000, 2)
74     wb = round(wb/1000, 2)
75     w = round(w/1000, 2)
76     qh = round(qh/1000, 2)
77     ql = round(ql/1000, 2)
78     if type(x4) == float:
79         x4 = round(x4*100, 2)
```

```
80     if type(x6) == float:
81         x6 = round(x6*100, 2)
82     n = round(n*100, 2)
83
84     return {'h1':h1, 'h2':h2, 'h3':h3, 'h4':h4, 'h5':h5,
85           'h6':h6, 'wt':wt, 'wb':wb, 'w':w, 'qh':qh, 'ql':ql,
86           'x4':x4, 'x6':x6, 'n':n}
87
88     #Função de busca dos pontos ótimos
89     def otimizacao(p_baixa, p_alta, t_alta, fluido):
90         lista_trabalhos = []
91         lista_rendimentos = []
92         lista_pressoes = []
93         intervalo_pressao = p_alta - p_baixa
94         divisoes = 1000
95         passo = intervalo_pressao / divisoes
96
97         for iteracao in range( 0, divisoes + 1 ):
98             p_intermediaria = p_baixa + iteracao * passo
99             r = rankine_reaquecido(p_baixa, p_alta, p_intermediaria,
100                                t_alta, fluido)
101             lista_pressoes.append(p_intermediaria)
102             lista_trabalhos.append(r['w'])
103             lista_rendimentos.append(r['n'])
104
105             #Ponto de maior produção de trabalho
106             idx_trabalho = lista_trabalhos.index(max(lista_trabalhos))
107             trabalho_max = lista_trabalhos[idx_trabalho]
108             p_trabalho_max = lista_pressoes[idx_trabalho] / 1000
109
110             #Ponto de maior rendimento
111             idx_rendimento = lista_rendimentos.index(max(lista_rendimentos))
112             rendimento_max = lista_rendimentos[idx_rendimento]
113             p_rendimento_max = lista_pressoes[idx_rendimento] / 1000
114
115             return {'trabalho_max':trabalho_max,
116                   'p_trabalho_max':p_trabalho_max,
117                   'rendimento_max':rendimento_max,
118                   'p_rendimento_max':p_rendimento_max}
119
120     #Função Principal - Parâmetros de Entrada
```

```
121 p1 = float(input("Insira a pressão inferior do sistema (kPa): "))
122 p4 = float(input("Insira a pressão intermediária do sistema (kPa): "))
123 p3 = float(input("Insira a pressão superior do sistema (kPa): "))
124 t3 = float(input("Insira a temperatura superior do sistema (°C): "))
125 fluido = 'Water'
126
127 p1 *= 1000
128 p3 *= 1000
129 p4 *= 1000
130 t3 += 273.15
131
132 r = rankine_reaquecido(p1, p3, p4, t3, fluido) #Resultados
133 maximos = otimizacao(p1, p3, t3, fluido)
134
135 #Imprimindo os resultados
136 print("\n----- Resultados ----- \n")
137 print(f"""Entalpia 1: {r['h1']} kJ/kg
138 Entalpia 2 {r['h2']} kJ/kg
139 Entalpia 3: {r['h3']} kJ/kg
140 Entalpia 4: {r['h4']} kJ/kg
141 Entalpia 5: {r['h5']} kJ/kg
142 Entalpia 6: {r['h6']} kJ/kg
143 Trabalho Bomba: {r['wb']} kJ/kg
144 Trabalho Turbina: {r['wt']} kJ/kg
145 Trabalho Líquido: {r['w']} kJ/kg
146 Calor Entrada: {r['qh']} kJ/kg
147 Calor Rejeitado: {r['ql']} kJ/kg
148 Título Turbina 1: {r['x4']}%
149 Título Turbina 2: {r['x6']}%
150 Rendimento: {r['n']}%\n""")
151
152 print("----- Pontos de Otimização -----")
153 print(f"""
154 Pressão Int. de maior trabalho: {maximos['p_trabalho_max']:.2f} kPa
155 Trabalho máximo: {maximos['trabalho_max']:.2f} kJ/kg
156 Pressão Int. de maior rendimento: {maximos['p_rendimento_max']:.2f} kPa
157 Rendimento máximo: {maximos['rendimento_max']:.2f}
158 """)
```

A.3 Ciclo Rankine Regenerativo

```
1 from CoolProp.CoolProp import PropsSI
2
3 def rankine_regenerativo(p1, p3, p5, t5, fluido):
4
5     if t5 == 272.15:
6         t5 = PropsSI('T', 'P', p5, 'Q', 1, fluido)
7
8         #Estado 1
9         h1 = PropsSI('H', 'P|liquid', p1, 'Q', 0, fluido) #Entalpia 1
10        s1 = PropsSI('S', 'P|liquid', p1, 'Q', 0, fluido) #Entropia 1
11
12        #Estado 2
13        p2 = p3 #Compressão
14        s2 = s1 #Processo Isoentrópico
15        h2 = PropsSI('H', 'P|liquid', p2, 'S', s2, fluido) #Entalpia 2
16
17        #Estado 3
18        h3 = PropsSI('H', 'P|liquid', p3, 'Q', 0, fluido) #Entalpia 3
19        s3 = PropsSI('S', 'P|liquid', p3, 'Q', 0, fluido) #Entropia 3
20
21        #Estado 4
22        s4 = s3
23        p4 = p5
24        h4 = PropsSI('H', 'P|liquid', p4, 'S', s4, fluido)
25
26        #Estado 5
27        h5 = PropsSI('H', 'P|gas', p5, 'T', t5, fluido)
28        s5 = PropsSI('S', 'P|gas', p5, 'T', t5, fluido)
29
30        #Estado 6
31        p6 = p3
32        s6 = s5
33        h6 = PropsSI('H', 'P|gas', p6, 'S', s6, fluido)
34
35        #Para cálculo do título ponto 6:
36        s6l = PropsSI('S', 'P|liquid', p6, 'Q', 0, fluido)
37        s6v = PropsSI('S', 'P|gas', p6, 'Q', 1, fluido)
38
```

```
39     if (s6 < s6v):
40         x6 = (s6 - s6l)/(s6v - s6l)
41     else:
42         x6 = 'Saída da turbina superaquecida'
43
44     #Estado 7
45     p7 = p1
46     s7 = s6
47     h7 = PropsSI('H', 'P|gas', p7, 'S', s7, fluido)
48
49     #Para cálculo do título ponto 7:
50     s7l = PropsSI('S', 'P|liquid', p7, 'Q', 0, fluido)
51     s7v = PropsSI('S', 'P|gas', p7, 'Q', 1, fluido)
52
53     if (s7 < s7v):
54         x7 = (s7 - s7l)/(s7v - s7l)
55     else:
56         x7 = 'Saída da turbina superaquecida'
57
58     #Fração mássica da extração
59     m = (h3 - h2)/(h6 - h2)
60
61     #Parâmetros de Saída
62     wb1 = h2 - h1 #Trabalho na bomba 1
63     wb2 = h4 - h3 #Trabalho na bomba 2
64     wb = wb1 + wb2 #Trabalho nas bombas
65     wt = (h5 - h6) + (1-m)*(h6 - h7) #Trabalho na turbina
66     w = wt - wb #Trabalho líquido
67     qh = h5 - h4 #Calor de Entrada
68     ql = (1-m)*(h7 - h1) #Calor rejeitado
69     n = (qh - ql)/qh #Rendimento
70
71     #Padronizando unidades
72     h1 = round(h1/1000, 2)
73     h2 = round(h2/1000, 2)
74     h3 = round(h3/1000, 2)
75     h4 = round(h4/1000, 2)
76     h5 = round(h5/1000, 2)
77     h6 = round(h6/1000, 2)
78     h7 = round(h7/1000, 2)
79     wt = round(wt/1000, 2)
```

```
80     wb = round(wb/1000, 2)
81     w = round(w/1000, 2)
82     qh = round(qh/1000, 2)
83     ql = round(ql/1000, 2)
84     if type(x6) == float:
85         x6 = round(x6*100, 2)
86     if type(x7) == float:
87         x7 = round(x7*100, 2)
88     n = round(n*100, 2)
89     m = round(m*100, 2)
90
91     return {'h1':h1, 'h2':h2, 'h3':h3, 'h4':h4, 'h5':h5,
92           'h6':h6, 'h7':h7, 'wt':wt, 'wb':wb, 'w':w, 'qh':qh,
93           'ql':ql, 'x6':x6, 'x7':x7, 'n':n, 'm':m}
94
95     #Função de busca dos pontos ótimos
96     def otimizacao(p_baixa, p_alta, t_alta, fluido):
97         lista_trabalhos = []
98         lista_rendimentos = []
99         lista_pressoes = []
100        intervalo_pressao = p_alta - p_baixa
101        divisoes = 1000
102        passo = intervalo_pressao / divisoes
103
104        for iteracao in range( 0, divisoes + 1 ):
105            p_intermediaria = p_baixa + iteracao * passo
106
107            r = rankine_regenerativo(p_baixa, p_intermediaria, p_alta,
108                                   t_alta, fluido)
109            lista_pressoes.append(p_intermediaria)
110            lista_trabalhos.append(r['w'])
111            lista_rendimentos.append(r['n'])
112
113            #Ponto de maior produção de trabalho
114            idx_trabalho = lista_trabalhos.index(max(lista_trabalhos))
115            trabalho_max = lista_trabalhos[idx_trabalho]
116            p_trabalho_max = lista_pressoes[idx_trabalho] / 1000
117
118            #Ponto de maior rendimento
119            idx_rendimento = lista_rendimentos.index(max(lista_rendimentos))
120            rendimento_max = lista_rendimentos[idx_rendimento]
```

```
121     p_rendimento_max = lista_pressoes[idx_rendimento] / 1000
122
123     return {'trabalho_max':trabalho_max,
124            'p_trabalho_max':p_trabalho_max,
125            'rendimento_max':rendimento_max,
126            'p_rendimento_max':p_rendimento_max}
127
128 #Função Principal - Parâmetros de Entrada
129 p1 = float(input("Insira a pressão inferior do sistema (kPa): "))
130 p3 = float(input("Insira a pressão intermediária do sistema (kPa): "))
131 p5 = float(input("Insira a pressão superior do sistema (kPa): "))
132 t5 = float(input("Insira a temperatura superior do sistema (°C): "))
133 fluido = 'Water'
134
135 p1 *= 1000
136 p3 *= 1000
137 p5 *= 1000
138 t5 += 273.15
139
140 r = rankine_regenerativo(p1, p3, p5, t5, fluido) #Resultados
141 maximos = otimizacao(p1, p5, t5, fluido)
142
143 #Imprimindo os resultados
144 print("\n----- Resultados ----- \n")
145 print(f"Entalpia 1: {r['h1']} kJ/kg
146 Entalpia 2: {r['h2']} kJ/kg
147 Entalpia 3: {r['h3']} kJ/kg
148 Entalpia 4: {r['h4']} kJ/kg
149 Entalpia 5: {r['h5']} kJ/kg
150 Entalpia 6: {r['h6']} kJ/kg
151 Entalpia 7: {r['h7']} kJ/kg
152 Trabalho Bomba: {r['wb']} kJ/kg
153 Trabalho Turbina: {r['wt']} kJ/kg
154 Trabalho Líquido: {r['w']} kJ/kg
155 Calor Entrada: {r['qh']} kJ/kg
156 Calor Rejeitado: {r['ql']} kJ/kg
157 Título Turbina 1: {r['x6']}%
158 Título Turbina 2: {r['x7']}%
159 Fração de extração: {r['m']}%
160 Rendimento: {r['n']}%\n")
161
```

```
162 print("----- Pontos de Otimização -----")
163 print(f"""
164 Pressão Int. de maior trabalho: {maximos['p_trabalho_max']:.2f} kPa
165 Trabalho máximo: {maximos['trabalho_max']:.2f} kJ/kg
166 Pressão Int. de maior rendimento: {maximos['p_rendimento_max']:.2f} kPa
167 Rendimento máximo: {maximos['rendimento_max']:.2f}
168 """)
```

APÊNDICE B – Função de Otimização

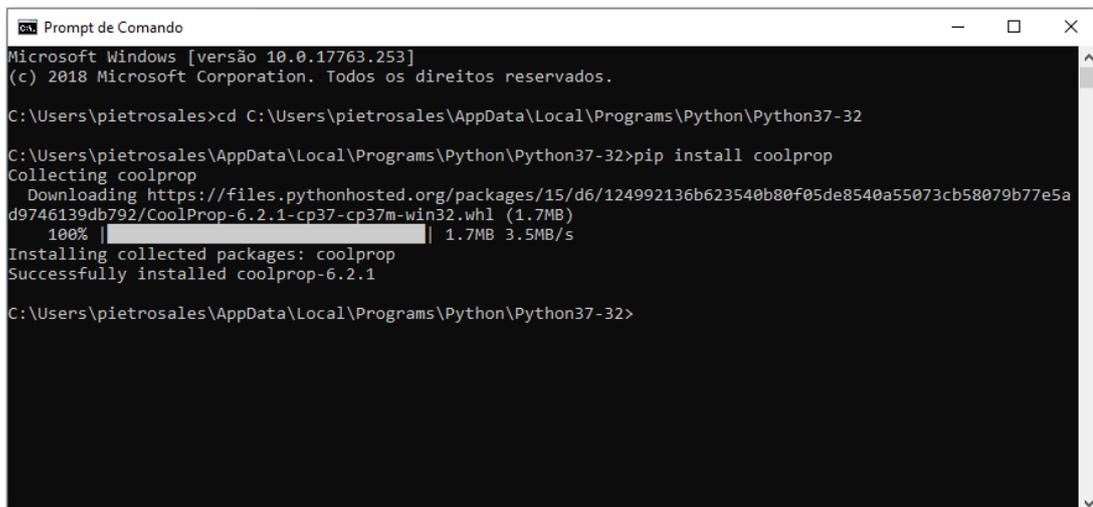
```
1 #Função de busca dos pontos ótimos
2 def otimizacao(p_baixa, p_alta, t_alta, fluido):
3     lista_trabalhos = []
4     lista_rendimentos = []
5     lista_pressoes = []
6     intervalo_pressao = p_alta - p_baixa
7     divisoes = 1000
8     passo = intervalo_pressao / divisoes
9
10    for iteracao in range( 0, divisoes + 1 ):
11        p_intermediaria = p_baixa + iteracao * passo
12        r = rankine_reaquecido(p_baixa, p_alta, p_intermediaria,
13                               t_alta, fluido)
14        lista_pressoes.append(p_intermediaria / 1000)
15        lista_trabalhos.append(r['w'])
16        lista_rendimentos.append(r['n'])
17
18    #Ponto de maior produção de trabalho
19    idx_trabalho = lista_trabalhos.index(max(lista_trabalhos))
20    trabalho_max = lista_trabalhos[idx_trabalho]
21    p_trabalho_max = lista_pressoes[idx_trabalho]
22
23    #Ponto de maior rendimento
24    idx_rendimento = lista_rendimentos.index(max(lista_rendimentos))
25    rendimento_max = lista_rendimentos[idx_rendimento]
26    p_rendimento_max = lista_pressoes[idx_rendimento]
27
28    return {'trabalho_max':trabalho_max,
29           'p_trabalho_max':p_trabalho_max,
30           'rendimento_max':rendimento_max,
31           'p_rendimento_max':p_rendimento_max}
```

APÊNDICE C – Instruções de execução dos programas Python no Windows

Para executar as aplicações no Windows, é necessário utilizar um interpretador Python e instalar o pacote CoolProp. É possível instalar o interpretador Python diretamente pelo site oficial em (PYTHON, 2019).

Após a instalação do Python, é possível instalar o pacote CoolProp. A Figura 24 exibe esse processo.

Figura 24 – Instalação CoolProp



```
Microsoft Windows [versão 10.0.17763.253]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\Users\pietrosales>cd C:\Users\pietrosales\AppData\Local\Programs\Python\Python37-32

C:\Users\pietrosales\AppData\Local\Programs\Python\Python37-32>pip install coolprop
Collecting coolprop
  Downloading https://files.pythonhosted.org/packages/15/d6/124992136b623540b80f05de8540a55073cb58079b77e5a
d9746139db792/CoolProp-6.2.1-cp37-cp37m-win32.whl (1.7MB)
    100% |#####| 1.7MB 3.5MB/s
Installing collected packages: coolprop
Successfully installed coolprop-6.2.1

C:\Users\pietrosales\AppData\Local\Programs\Python\Python37-32>
```

Fonte: Autor

Os seguintes passos foram realizados na Figura 24:

1. Acessar o prompt de comando do Windows:
 - Pode ser acessado digitando *CMD* na pesquisa do menu iniciar do Windows.
2. Acessar o diretório onde o executável do Python foi instalado¹:
 - Digitar o seguinte comando, substituindo *DIRETORIO_PYTHON* pelo diretório: *cd DIRETORIO_PYTHON*.
3. Instalar o pacote CoolProp:

¹ Caso este diretório tenha sido adicionado na variável *PATH* do sistema, não é necessário realizar esse passo

- Utilizar o gerenciador de pacotes PIP, com o seguinte comando: *pip install coolprop*.

APÊNDICE D – Exemplo de código XML para interface do Aplicativo

A criação de cada tela do aplicativo móvel é feita individualmente com códigos XML. O código apresentado neste apêndice foi utilizado na criação da interface, referente ao ciclo reaquecido, mostrada na Figura 19.

Os seguintes elementos estão presentes nessa interface:

- ScrollView: Responsável por tornar a tela deslizando;
- ConstraintLayout: Estrutura básica para a interface visual;
- Toolbar: Barra de tarefas;
- TextView: Elementos de texto;
- ImageView: Inserção de imagens;
- Button: Botão para interação com o usuário.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:fillViewport="true"
8      tools:context=".TelaRankReaquecido">
9
10     <android.support.constraint.ConstraintLayout
11         android:layout_width="match_parent"
12         android:layout_height="match_parent"
13         android:background="@android:color/white">
14
15         <include
16             android:id="@+id/toolbar_rankine"
17             layout="@layout/toolbar"
18             app:layout_constraintEnd_toEndOf="parent"

```

```
19         app:layout_constraintStart_toStartOf="parent"
20         app:layout_constraintTop_toTopOf="parent" />
21
22     <TextView
23         android:id="@+id/txt_arranjo"
24         android:layout_width="0dp"
25         android:layout_height="wrap_content"
26         android:layout_marginStart="32dp"
27         android:layout_marginTop="8dp"
28         android:layout_marginEnd="32dp"
29         android:background="@drawable/fundo_titulos"
30         android:gravity="center"
31         android:padding="2sp"
32         android:text="Arranjo Físico"
33         android:textSize="24sp"
34         app:layout_constraintEnd_toEndOf="parent"
35         app:layout_constraintStart_toStartOf="parent"
36         app:layout_constraintTop_toBottomOf="@+id/toolbar_rankine" />
37
38     <ImageView
39         android:id="@+id/img_arranjo"
40         android:layout_width="0dp"
41         android:layout_height="215dp"
42         android:layout_marginStart="32dp"
43         android:layout_marginTop="8dp"
44         android:layout_marginEnd="32dp"
45         android:adjustViewBounds="true"
46         android:foreground="@drawable/fundo_botao_dois"
47         android:padding="8sp"
48         android:scaleType="fitCenter"
49         app:layout_constraintEnd_toEndOf="parent"
50         app:layout_constraintStart_toStartOf="parent"
51         app:layout_constraintTop_toBottomOf="@+id/txt_arranjo"
52         app:srcCompat="@drawable/im_reaquecido" />
53
54     <TextView
55         android:id="@+id/txt_diagrama"
56         android:layout_width="0dp"
57         android:layout_height="wrap_content"
58         android:layout_marginStart="32dp"
59         android:layout_marginTop="8dp"
```

```
60         android:layout_marginEnd="32dp"
61         android:background="@drawable/fundo_titulos"
62         android:gravity="center"
63         android:padding="2sp"
64         android:text="Diagrama T-s"
65         android:textSize="24sp"
66         app:layout_constraintEnd_toEndOf="parent"
67         app:layout_constraintStart_toStartOf="parent"
68         app:layout_constraintTop_toBottomOf="@+id/img_arranjo" />
69
70     <ImageView
71         android:id="@+id/img_diagrama"
72         android:layout_width="0dp"
73         android:layout_height="215dp"
74         android:layout_marginStart="32dp"
75         android:layout_marginTop="8dp"
76         android:layout_marginEnd="32dp"
77         android:adjustViewBounds="true"
78         android:foreground="@drawable/fundo_botao_dois"
79         android:padding="2sp"
80         android:scaleType="fitCenter"
81         app:layout_constraintEnd_toEndOf="parent"
82         app:layout_constraintStart_toStartOf="parent"
83         app:layout_constraintTop_toBottomOf="@+id/txt_diagrama"
84         app:srcCompat="@drawable/im_reaquecido_ts" />
85
86     <Button
87         android:id="@+id/btn_dimensionar"
88         android:layout_width="0dp"
89         android:layout_height="wrap_content"
90         android:layout_marginStart="32dp"
91         android:layout_marginTop="8dp"
92         android:layout_marginEnd="32dp"
93         android:layout_marginBottom="8dp"
94         android:background="@drawable/fundo_botao"
95         android:clickable="true"
96         android:drawableEnd="@drawable/ic_arrow_forward_white_24dp"
97         android:foreground="?attr/selectableItemBackground"
98         android:paddingStart="24dp"
99         android:paddingEnd="12dp"
100        android:text="Dimensionar"
```

```
101         android:textColor="@android:color/white"
102         android:textSize="14sp"
103         app:layout_constraintBottom_toBottomOf="parent"
104         app:layout_constraintEnd_toEndOf="parent"
105         app:layout_constraintStart_toStartOf="parent"
106         app:layout_constraintTop_toBottomOf="@+id/img_diagrama" />
107
108     </android.support.constraint.ConstraintLayout>
109
110 </ScrollView>
```
