

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**VINÍCIUS FILIPAKI**

**BIBLIOTECA GENÉRICA DE RESOLUÇÃO DE CICLOS TERMODINÂMICOS  
EM REGIME PERMANENTE**

**GUARAPUAVA**

**2023**

**VINÍCIUS FILIPAKI**

**BIBLIOTECA GENÉRICA DE RESOLUÇÃO DE CICLOS TERMODINÂMICOS  
EM REGIME PERMANENTE**

**Library for solving generic thermodynamic cycles in steady state regime**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Mecânica do Curso de Bacharelado em Engenharia Mecânica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Christian Naaktgeboren

**GUARAPUAVA**

**2023**



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

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

**VINÍCIUS FILIPAKI**

**BIBLIOTECA GENÉRICA DE RESOLUÇÃO DE CICLOS TERMODINÂMICOS  
EM REGIME PERMANENTE**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia Mecânica  
do Curso de Bacharelado em Engenharia  
Mecânica da Universidade Tecnológica Federal  
do Paraná.

Data de aprovação: 07/dezembro/2023

---

Christian Naaktgeboren  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Sergio Dalmás  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Douglas Santos de Oliveira  
Doutorado  
Universidade Tecnológica Federal do Paraná

**GUARAPUAVA**  
**2023**

## RESUMO

Ciclos termodinâmicos são usados em processos de geração de trabalho e de refrigeração, avanços em sua área podem apoiar o desenvolvimento de sistemas mais sustentáveis e eficientes. Desse modo, para promover a agilidade no processo de modelagem e análise de ciclos termodinâmicos, o presente trabalho busca apresentar uma biblioteca que tem essa capacidade. A ferramenta, desenvolvida na linguagem de programação Julia, disponibiliza uma estrutura de formatação própria para a representação dos ciclos e fornecimento de propriedades já conhecidas, com base nisso, o algoritmo procura encontrar as outras propriedades desconhecidas do sistema de forma automática. Além da resolução de ciclos, a biblioteca apresentada também oferece recursos de exibição dos resultados obtidos, com a geração de tabelas e gráficos para ilustrar com clareza as propriedades do ciclo. Dessa forma, a ferramenta permite uma análise rápida e confiável de ciclos termodinâmicos que pode ser usada por profissionais e estudantes interessados no desenvolvimento desta área.

**Palavras-chave:** ciclos termodinâmicos; modelagem de ciclos; regime permanente; resolução de ciclos; análise de ciclos.

## ABSTRACT

Thermodynamic cycles are used in work generation and cooling processes, advances in their area can support the development of more sustainable and efficient systems. Therefore, to promote agility in the process of modeling and analyzing thermodynamic cycles, this work seeks to present a library that has this capacity. The tool, developed in the Julia programming language, provides its own formatting structure for representing cycles and providing already known properties, based on this, the algorithm seeks to find the other unknown properties of the system automatically. In addition to cycle resolution, the presented library also offers resources for displaying the results obtained, with the generation of tables and graphs to clearly illustrate the properties of the cycle. In this way, the tool allows a quick and reliable analysis of thermodynamic cycles that can be used by professionals and students interested in the development of this area.

**Keywords:** thermodynamic cycles; cycle modeling; steady state; cycle resolution; cycle analysis.

## LISTA DE FIGURAS

- Figura 1** – Representação esquemática de uma turbina. Os rótulos 1 e 2 representam, estados de entrada e saída da turbina, respectivamente, do fluido de trabalho, o qual pode ser um gás, um vapor ou uma mistura. As setas coloridas indicam interações energéticas, sendo as de calor indicadas em vermelho, e as de trabalho em seta azul mais espessa, juntamente com os respectivos rótulos  $q$  ou  $w$ , com subscrito indicando a direção da interação, seja *ent* e *sai*, respectivamente para entrando e saindo. O escoamento do fluido de trabalho se dá no sentido 1-2 em regime de expansão, comunicado pelo contorno da turbina. . . . . 20
- Figura 2** – Representação esquemática de uma bomba. Os rótulos 1 e 2 representam, estados de entrada e saída da bomba, respectivamente, do fluido de trabalho. A seta azul indica interação energética de trabalho, juntamente com o rótulo  $w$ , com subscrito *ent* indicando a direção de entrada da interação. O escoamento do fluido de trabalho se dá no sentido 1-2 em regime de compressão. . . . . 21
- Figura 3** – Representação esquemática de um condensador. Os rótulos 1 e 2 representam, estados de entrada e saída do condensador, respectivamente, do fluido de trabalho. A seta vermelha indica interação energética de calor, juntamente com o rótulo  $q$ , com subscrito *sai* indicando a direção de saída da interação. O escoamento do fluido de trabalho se dá no sentido 1-2 em um processo isobárico, que vai transferir calor para fora do sistema fazendo o estado de saída do componente ser líquido comprimido saturado. . . . . 22
- Figura 4** – Representação esquemática de uma caldeira. Os rótulos 1 e 2 representam, estados de entrada e saída da caldeira, respectivamente, do fluido de trabalho. A seta vermelha indica interação energética de calor, juntamente com o rótulo  $q$ , com subscrito *ent* indicando a direção de entrada da interação. O escoamento do fluido de trabalho se dá no sentido 1-2 em um processo que pode ser isobárico, se não houver perda de carga, e vai transferir calor para dentro do sistema. . . . . 23

- Figura 5 – Representação esquemática de um evaporador. Os rótulos 1 e 2 representam, estados de entrada e saída do evaporador, respectivamente, do fluido de trabalho. A seta vermelha indica interação energética de calor, juntamente com o rótulo  $q$ , com subscrito  $ent$  indicando a direção de entrada da interação. O escoamento do fluido de trabalho se dá no sentido 1-2 em um processo que pode ser isobárico, se não houver perda de carga, e vai transferir calor para dentro do sistema fazendo o estado de saída do componente ser vapor saturado. . . . . 24**
- Figura 6 – Representação esquemática de uma válvula de expansão. Os rótulos 1 e 2 representam, estados de entrada e saída da válvula de expansão, respectivamente, do fluido de trabalho. O escoamento do fluido de trabalho se dá no sentido 1-2 em um processo que reduz a pressão e pode ser isoentálpico, se também for adiabático. . . . . 25**
- Figura 7 – Representação esquemática de um compressor. Os rótulos 1 e 2 representam, estados de entrada e saída do compressor, respectivamente, do fluido de trabalho. As setas coloridas indicam interações energéticas, sendo as de calor indicadas em vermelho, e as de trabalho em seta azul mais espessa, juntamente com os respectivos rótulos  $q$  ou  $w$ , com subscrito indicando a direção da interação, seja  $ent$  e  $sai$ , respectivamente para entrando e saindo. O escoamento do fluido de trabalho se dá no sentido 1-2 em regime de compressão. . . . . 25**
- Figura 8 – Representação esquemática de uma câmara de mistura. Neste componente é realizada a mistura de dois ou mais estados por meio da equação de balanço de energia. Cada linha ligada ao retângulo representa um estado, que pode ser de entrada ou saída. . . . . 26**
- Figura 9 – Representação esquemática de um separador. O rótulo 1 representa o estado de entrada e os rótulos 2 e 3 representam os estados de saída do separador, que são líquido comprimido saturado e vapor saturado, respectivamente. O componente irá separar a parte líquida e gasosa, de acordo com o título do estado de entrada. . . . . 27**

<b>Figura 10 – Representação esquemática de um trocador de calor. Os rótulos <i>a</i> e <i>b</i> representam respectivamente os estados de entrada e saída de uma determinada corrente. Já os rótulos 1 e 2 representam respectivamente os estados de entrada e saída de uma outra corrente, que não se misturam com a primeira corrente. Neste componente há a interação entre diferentes correntes, sem haver a mistura entre elas. Essa interação depende da efetividade <math>\epsilon</math> do trocador de calor. . . . .</b>	<b>28</b>
<b>Figura 11 – Representação do ciclo Rankine simples, em que se tem uma bomba, uma caldeira, uma turbina e um condensador. . . . .</b>	<b>36</b>
<b>Figura 12 – Representação do ciclo Rankine com reaquecimento, em que se tem uma bomba, duas caldeiras, duas turbinas e um condensador. . . . .</b>	<b>47</b>
<b>Figura 13 – Representação do ciclo Rankine regenerativo, em que se tem duas bombas, uma caldeira, uma turbina com duas saídas, um condensador, e uma câmara de mistura. . . . .</b>	<b>50</b>
<b>Figura 14 – Representação do ciclo do Brayton, em que se tem um compressor, uma turbina e duas câmaras de combustão. . . . .</b>	<b>53</b>
<b>Figura 15 – Representação do ciclo de refrigeração, em que se tem um compressor, um evaporador, uma válvula de expansão e um condensador. . . . .</b>	<b>56</b>
<b>Figura 16 – Representação do sistema de refrigeração em cascata, em que se tem um ciclo com um condensador, uma válvula de expansão e um compressor, outro ciclo com um evaporador, uma válvula de expansão e um compressor, e um trocador de calor que liga ambos os ciclos. . . .</b>	<b>60</b>
<b>Figura 17 – Representação do sistema com 3 ciclos, em que se tem um ciclo de Dióxido de carbono com dois compressores, uma câmara de combustão, e uma turbina, um ciclo de água com uma bomba, um condensador, e uma turbina, um ciclo de Refrigerante-134a com uma turbina, uma bomba e um condensador, e quatro trocadores de calor fazendo a ligação entre os ciclos de Dióxido de carbono e água, e um trocador de calor fazendo a ligação entre os ciclos de Refrigerante-134a e Dióxido de carbono. . . . .</b>	<b>66</b>



<b>Figura 18 – Captura de tela da página GitHub que hospeda a biblioteca desenvolvida no trabalho, proporcionando uma visão geral do projeto e seu ambiente de código-fonte. . . . .</b>	<b>88</b>
<b>Figura 19 – Captura de tela da página de documentação da biblioteca desenvolvida no trabalho, proporcionando um manual de como usar a biblioteca e exemplos de uso. . . . .</b>	<b>89</b>

## LISTA DE GRÁFICOS

Gráfico 1 – Diagrama $T - s$ do exemplo 1 . . . . .	39
Gráfico 2 – Diagrama $T - s$ do exemplo 2 . . . . .	42
Gráfico 3 – Diagrama $T - s$ do exemplo 3 . . . . .	44
Gráfico 4 – Diagrama $T - s$ do exemplo 5 . . . . .	49
Gráfico 5 – Diagrama $T - s$ do exemplo 6 . . . . .	52
Gráfico 6 – Diagrama $T - s$ do exemplo 7 . . . . .	55
Gráfico 7 – Diagrama $T - s$ do exemplo 8 . . . . .	58
Gráfico 8 – Diagrama $P - h$ do exemplo 8 . . . . .	58
Gráfico 9 – Diagrama $T - s$ do 1º ciclo do exemplo 9 . . . . .	62
Gráfico 10 – Diagrama $T - s$ do 2º ciclo do exemplo 9 . . . . .	63
Gráfico 11 – Diagrama $T - s$ do sistema do exemplo 9 . . . . .	64
Gráfico 12 – Diagrama $T - s$ do ciclo de CO2 do exemplo 10 . . . . .	69
Gráfico 13 – Diagrama $T - s$ do ciclo de água do exemplo 10 . . . . .	70
Gráfico 14 – Diagrama $T - s$ do ciclo de R134a do exemplo 10 . . . . .	71
Gráfico 15 – Diagrama $T - s$ do sistema do exemplo 10 . . . . .	73
Gráfico 16 – Diagrama $T - s$ do sistema do exemplo 10, sem os nomes dos estados	74
Gráfico 17 – Diagrama $T - s$ do sistema do exemplo 10, multiplicando a entropia pelo fluxo mássico . . . . .	75

## LISTA DE TABELAS

Tabela 1 – Relação das propriedades termodinâmicas e símbolos dos atributos dos estados. . . . .	37
Tabela 2 – Propriedades de cada estado do exemplo 1 . . . . .	38
Tabela 3 – Propriedades do ciclo do exemplo 1 . . . . .	38
Tabela 4 – Propriedades de cada estado do exemplo 2. . . . .	41
Tabela 5 – Propriedades do ciclo do exemplo 2 . . . . .	41
Tabela 6 – Propriedades de cada estado do exemplo 3. . . . .	43
Tabela 7 – Propriedades do ciclo do exemplo 3 . . . . .	44
Tabela 8 – Propriedades de cada estado do exemplo 4. . . . .	45
Tabela 9 – Propriedades do ciclo do exemplo 4 . . . . .	46
Tabela 10 – Propriedades de cada estado do exemplo 5 . . . . .	48
Tabela 11 – Propriedades do ciclo do exemplo 5 . . . . .	49
Tabela 12 – Propriedades de cada estado do exemplo 6 . . . . .	51
Tabela 13 – Propriedades do ciclo do exemplo 6 . . . . .	52
Tabela 14 – Propriedades de cada estado do exemplo 7 . . . . .	54
Tabela 15 – Propriedades do ciclo do exemplo 7 . . . . .	55
Tabela 16 – Propriedades de cada estado do exemplo 8 . . . . .	57
Tabela 17 – Propriedades do ciclo do exemplo 8 . . . . .	57
Tabela 18 – Propriedades de cada estado do 1º ciclo do exemplo 9 . . . . .	62
Tabela 19 – Propriedades do 1º ciclo do exemplo 9 . . . . .	62
Tabela 20 – Propriedades de cada estado do 2º ciclo do exemplo 9 . . . . .	63
Tabela 21 – Propriedades do 2º ciclo do exemplo 9 . . . . .	63
Tabela 22 – Propriedades do sistema do exemplo 9 . . . . .	64
Tabela 23 – Propriedades de cada estado do ciclo de CO <sub>2</sub> do exemplo 10 . . . . .	68
Tabela 24 – Propriedades do ciclo de CO <sub>2</sub> do exemplo 10 . . . . .	68
Tabela 25 – Propriedades de cada estado do ciclo de água do exemplo 10 . . . . .	69
Tabela 26 – Propriedades do ciclo de água do exemplo 10 . . . . .	70
Tabela 27 – Propriedades de cada estado do ciclo de R134a do exemplo 10 . . . . .	71
Tabela 28 – Propriedades do ciclo de R134a do exemplo 10 . . . . .	71
Tabela 29 – Propriedades do sistema do exemplo 10 . . . . .	72

<b>Tabela 30 – Propriedades exemplo 11 - SYSTEM . . . . .</b>	<b>77</b>
<b>Tabela 31 – Propriedades exemplo 11 - compressor: st1 » st2 . . . . .</b>	<b>77</b>
<b>Tabela 32 – Propriedades exemplo 11 - compressor: st3 » st4 . . . . .</b>	<b>77</b>
<b>Tabela 33 – Propriedades exemplo 11 - combustion_chamber: st4 » st5 . . . . .</b>	<b>78</b>
<b>Tabela 34 – Propriedades exemplo 11 - turbine: st5 » st6 . . . . .</b>	<b>78</b>
<b>Tabela 35 – Propriedades exemplo 11 - pump: sta » stb . . . . .</b>	<b>78</b>
<b>Tabela 36 – Propriedades exemplo 11 - turbine: stf » stg . . . . .</b>	<b>79</b>
<b>Tabela 37 – Propriedades exemplo 11 - condenser: stg » sta . . . . .</b>	<b>79</b>
<b>Tabela 38 – Propriedades exemplo 11 - pump: stj » stk . . . . .</b>	<b>79</b>
<b>Tabela 39 – Propriedades exemplo 11 - turbine: stl » stm . . . . .</b>	<b>80</b>
<b>Tabela 40 – Propriedades exemplo 11 - condenser: stm » stj . . . . .</b>	<b>80</b>
<b>Tabela 41 – Propriedades exemplo 11 - heater_exchanger: [st2, stb] » [st3, stc] . . .</b>	<b>80</b>
<b>Tabela 42 – Propriedades exemplo 11 - heater_exchanger: [st6, ste] » [st7, stf] . . .</b>	<b>81</b>
<b>Tabela 43 – Propriedades exemplo 11 - heater_exchanger: [std, st7] » [ste, st8] . . .</b>	<b>81</b>
<b>Tabela 44 – Propriedades exemplo 11 - heater_exchanger: [stc, st8] » [std, st9] . . .</b>	<b>81</b>
<b>Tabela 45 – Propriedades exemplo 11 - heater_exchanger: [stk, st9] » [stl, st10] . . .</b>	<b>82</b>

## LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Código do exemplo 1 . . . . .	37
Listagem 2 – Código do exemplo 2. . . . .	40
Listagem 3 – Código do exemplo 3. . . . .	43
Listagem 4 – Código do exemplo 4 . . . . .	45
Listagem 5 – Código do exemplo 5 . . . . .	48
Listagem 6 – Código do exemplo 6 . . . . .	51
Listagem 7 – Código do exemplo 7 . . . . .	54
Listagem 8 – Código do exemplo 8 . . . . .	56
Listagem 9 – Código do exemplo 9 . . . . .	61
Listagem 10 – Código do exemplo 10 . . . . .	67
Listagem 11 – Código do exemplo 11 . . . . .	76

## LISTA DE ABREVIATURAS E SIGLAS

### **Siglas**

COP	Coeficiente de Performance
CSS	<i>Cascading Style Sheet</i>
HTML	<i>HyperText Markup Language</i>

## LISTA DE SÍMBOLOS

### Letras Latinas

$T$	Temperatura	[K]
$P$	Pressão	[kPa]
$h$	Entalpia	$\left[\frac{\text{kJ}}{\text{kg}}\right]$
$s$	Entropia	$\left[\frac{\text{kJ}}{\text{kg}\cdot\text{K}}\right]$
$x$	Titulo	
$\dot{m}$	Fluxo mássico	$\left[\frac{\text{kg}}{\text{s}}\right]$
$\dot{Q}$	Transferência de calor	[kW]
$\dot{W}$	Trabalho	[kW]

### Letras Gregas

$\epsilon$	Efetividade
$\eta$	Eficiência

### Subscritos

$ent$	Total que entra no sistema
$sai$	Total que sai do sistema
$t$	Referente à eficiência térmica

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>1.1</b>	<b>Objetivos</b>	<b>18</b>
1.1.1	Objetivo geral	18
1.1.2	Objetivos específicos	18
<b>1.2</b>	<b>Justificativa</b>	<b>18</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>20</b>
<b>2.1</b>	<b>Equações para cada componente</b>	<b>20</b>
2.1.1	Turbina	20
2.1.2	Bomba	21
2.1.3	Condensador	22
2.1.4	Caldeira	23
2.1.5	Evaporador	24
2.1.6	Válvula de expansão	25
2.1.7	Compressor	25
2.1.8	Câmara de mistura	26
2.1.9	Separador	27
2.1.10	Trocador de calor	28
<b>2.2</b>	<b>Cálculo do desempenho dos ciclos</b>	<b>29</b>
2.2.1	Ciclos de geração de potência	29
2.2.2	Ciclos de refrigeração	29
<b>2.3</b>	<b>Imbalanços e Geração de entropia</b>	<b>29</b>
2.3.1	Imbalanço de massa	29
2.3.2	Imbalanço de energia	30
2.3.3	Geração de entropia	30
<b>3</b>	<b>METODOLOGIA</b>	<b>31</b>
<b>3.1</b>	<b>Levantamento de variáveis do sistema</b>	<b>31</b>
3.1.1	Fluido de trabalho	31
3.1.2	Componentes do ciclo	31
<b>3.2</b>	<b>Desenvolvimento do código</b>	<b>31</b>
3.2.1	Inserção das informações do ciclo	31



3.2.2	Controle de equações do ciclo . . . . .	32
3.2.3	Manipulação de equações condicionais . . . . .	33
3.2.4	Obtenção das propriedades térmicas . . . . .	33
3.2.5	Gerenciamento dos estados termodinâmicos . . . . .	33
3.2.6	Formulação de hipóteses . . . . .	34
3.2.7	Gerenciamento dos ciclos do sistema . . . . .	34
3.2.8	Cálculo das características termodinâmicas . . . . .	34
3.2.9	Apresentação dos resultados . . . . .	35
3.2.10	Manipulação dos resultados . . . . .	35
<b>4</b>	<b>RESULTADOS . . . . .</b>	<b>36</b>
<b>4.1</b>	<b>Exemplo 1 - Ciclo Rankine simples ideal . . . . .</b>	<b>36</b>
<b>4.2</b>	<b>Exemplo 2 - Ciclo Rankine simples real . . . . .</b>	<b>40</b>
<b>4.3</b>	<b>Exemplo 3 - Mudança do fluido do ciclo . . . . .</b>	<b>42</b>
<b>4.4</b>	<b>Exemplo 4 - Definir o fluxo mássico principal do ciclo . . . . .</b>	<b>44</b>
<b>4.5</b>	<b>Exemplo 5 - Ciclo Rankine supercrítico com reaquecimento . . . . .</b>	<b>46</b>
<b>4.6</b>	<b>Exemplo 6 - Ciclo Rankine regenerativo . . . . .</b>	<b>50</b>
<b>4.7</b>	<b>Exemplo 7 - Ciclo Brayton . . . . .</b>	<b>53</b>
<b>4.8</b>	<b>Exemplo 8 - Ciclo de refrigeração . . . . .</b>	<b>55</b>
<b>4.9</b>	<b>Exemplo 9 - Sistema de refrigeração em cascata . . . . .</b>	<b>59</b>
4.9.1	Ciclo superior . . . . .	62
4.9.2	Ciclo inferior . . . . .	63
4.9.3	Propriedades do sistema . . . . .	64
<b>4.10</b>	<b>Exemplo 10 - Interação de 3 ciclos . . . . .</b>	<b>65</b>
4.10.1	Ciclo de CO <sub>2</sub> . . . . .	68
4.10.2	Ciclo de água . . . . .	69
4.10.3	Ciclo de R134a . . . . .	71
4.10.4	Propriedades do sistema . . . . .	72
<b>4.11</b>	<b>Exemplo 11 - Imbalanço de massa e de energia, e geração de entropia . . . . .</b>	<b>76</b>
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>83</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>84</b>

<b>APÊNDICES</b>	<b>86</b>
<b>APÊNDICE A – CÓDIGO DESENVOLVIDO . . . . .</b>	<b>88</b>

## 1 INTRODUÇÃO

Os ciclos termodinâmicos estão presentes no funcionamento de vários sistemas de engenharia, como na geração de energia, em processos industriais e em sistemas de refrigeração. Ao compreender e otimizar esses ciclos, é possível desenvolver sistemas de energia mais eficientes e sustentáveis para atender à demanda global por energia.

O seu funcionamento consiste em uma sequência de processos termodinâmicos que dependem das propriedades de cada componente e a ordem em que esses elementos são distribuídos. Para resolvê-los é importante ter um entendimento das propriedades dos componentes que compõem o ciclo.

Cada componente possui suas próprias propriedades, as quais devem ser consideradas para que o ciclo seja resolvido corretamente. Essas propriedades estão diretamente relacionadas aos processos termodinâmicos que ocorrem dentro de cada componente, estabelecendo uma relação entre os estados de entrada e saída do mesmo.

Além disso, é necessário realizar um estudo para determinar a estratégia de resolução que será necessária para resolver o ciclo. Essa estratégia se refere à ordem em que as equações devem ser resolvidas para que as propriedades do ciclo possam ser determinadas. Isso pode ser influenciado por fatores como a forma que os componentes são distribuídos no ciclo, e as propriedades já conhecidas do sistema.

Assim sendo, a resolução de ciclos termodinâmicos pode ser automatizada, para evitar as dificuldades que os métodos convencionais apresentam. Já existem bibliotecas para a obtenção de propriedades termodinâmicas, em que podemos citar a biblioteca CoolProp (2023), que é capaz de calcular várias propriedades de uma ampla variedade de fluidos, porém não trabalha com a resolução de ciclos termodinâmicos.

Segundo as informações do CoolProp (2023), é possível encontrar outros projetos que o possuem em suas dependências Dechesne *et al.* (2019), Bell e Bradshaw (2012), Bell (2011), DWSIM (2023), SmoWeb (2018), PropiedadesDeFluidos (2022), WebThermodynamics (2023). No entanto, esses projetos possuem escopos distintos do presente trabalho, sendo que são desenvolvidos em diversas linguagens, e alguns projetos apresentando pouca documentação e longos períodos sem atualização.

Diante disso, o objetivo deste estudo é desenvolver uma biblioteca em linguagem Julia capaz de resolver ciclos termodinâmicos em regime permanente de potência e refrigeração, visando uma análise mais rápida e eficiente. Atualmente, os programadores continuam a preferir linguagens dinâmicas para o desenvolvimento de algoritmos em análise de dados, matemática aplicada e engenharia, apesar dos avanços na tecnologia de compiladores e execução para computação de alto desempenho. Nesse sentido, a linguagem Julia foi escolhida por combinar alta performance, comparável a linguagens como C e Fortran, com facilidade de desenvolvimento, similar a linguagens dinâmicas. Além disso, a implementação do Julia é licenciada sob a licença MIT e está disponível gratuitamente para uso e modificação por qualquer pessoa, e a

comunidade Julia já desenvolveu mais de 2000 pacotes para várias aplicações, o que possibilita uma grande gama de funcionalidades para os usuários da linguagem (BEZANSON *et al.*, 2012).

Portanto, a escolha da linguagem torna o desenvolvimento mais ágil, proporcionando uma maior legibilidade e facilidade na manutenção do código, além de garantir que o resultado final apresente alta performance e eficiência. Dessa forma, a biblioteca tem a capacidade de resolver os ciclos termodinâmicos, e certificar que seus resultados sejam rápidos e confiáveis para o uso dos usuários.

## 1.1 Objetivos

### 1.1.1 Objetivo geral

Desenvolver uma biblioteca eficiente que possa calcular as propriedades de ciclos termodinâmicos em regime permanente de potência e de refrigeração a partir das informações disponíveis, como a distribuição dos componentes do ciclo e as propriedades térmicas de estados conhecidos, e com base nisso, apresentar as informações obtidas de maneira clara e objetiva.

### 1.1.2 Objetivos específicos

1. Desenvolvimento do programa.

Fazer um programa que resolva ciclos termodinâmicos utilizando a linguagem Julia.

2. Apresentação dos resultados.

Programar uma função que apresente as informações da resolução do ciclo da melhor forma.

3. Validação do programa.

Validar o programa através da comparação dos resultados obtidos com os resultados de métodos convencionais.

4. Estruturar o programa como uma biblioteca.

Elaboração de uma biblioteca testável e documentada.

## 1.2 Justificativa

A biblioteca apresentada neste trabalho é uma ferramenta valiosa para a análise eficiente de ciclos termodinâmicos. Ao incorporar o comportamento de cada componente do sistema, o

código elimina as dificuldades apresentadas pelos métodos convencionais Çengel, Boles e Kanoğlu (2019), Wylen (1985), Jones e Hawkins (1986), Moran e Shapiro (2002), Bejan (2016) e busca calcular todas as propriedades do sistema de forma automática. Além disso, diferentemente dos outros projetos relacionados Dechesne *et al.* (2019), Bell e Bradshaw (2012), Bell (2011), DWSIM (2023), SmoWeb (2018), PropiedadesDeFluidos (2022), WebThermodynamics (2023), os quais são desenvolvidos em diversas linguagens, a implementação do algoritmo tem como objetivo agilizar o processo de modelagem e análise de ciclos termodinâmicos usando a linguagem Julia, proporcionando resultados confiáveis e rápidos, com benefícios tanto para a indústria quanto para a pesquisa acadêmica.

## 2 REFERENCIAL TEÓRICO

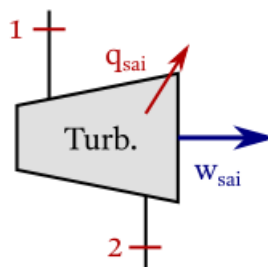
O funcionamento de um ciclo termodinâmico é determinado pela interação de diversos componentes, como turbinas, compressores, caldeiras e condensadores. Cada componente do ciclo possui propriedades diferentes, que podem ser descritas utilizando equações termodinâmicas que definem as relações dos estados de entrada e saída dos componentes.

### 2.1 Equações para cada componente

A seguir apresenta-se o equacionamento de componentes típicos de ciclos de regime permanente com base em Çengel e Boles (2013), Struchtrup (2014), Bejan (2016), Kroos e Potter (2015), Çengel, Boles e Kanoğlu (2019), Jones e Hawkins (1986), Wylen (1985), Moran e Shapiro (2002) e Bejan (1993).

#### 2.1.1 Turbina

**Figura 1 – Representação esquemática de uma turbina. Os rótulos 1 e 2 representam, estados de entrada e saída da turbina, respectivamente, do fluido de trabalho, o qual pode ser um gás, um vapor ou uma mistura. As setas coloridas indicam interações energéticas, sendo as de calor indicadas em vermelho, e as de trabalho em seta azul mais espessa, juntamente com os respectivos rótulos  $q$  ou  $w$ , com subscrito indicando a direção da interação, seja *ent* e *sai*, respectivamente para entrando e saindo. O escoamento do fluido de trabalho se dá no sentido 1-2 em regime de expansão, comunicado pelo contorno da turbina.**



Fonte: Naaktgeboren (2023).

O processo do componente representado na Figura 1 vai diminuir a pressão e produzir trabalho, além disso, o processo pode ser isentrópico.

O trabalho realizado pelo componente é descrito pelas equações:

$$\dot{W}_{sai} = \dot{m}_1 \cdot h_1 - \dot{m}_2 \cdot h_2 - \dot{Q}_{sai} \quad (1)$$

e

$$w_{sai} = h_1 - h_2 - q_{sai} \quad (2)$$

O componente pode ter as propriedades descritas na Equação 3, Equação 4 e Equação 5, dependendo das características da turbina.

Se o processo do componente for adiabático:

$$\dot{Q}_{sai} = q_{sai} = 0 \quad (3)$$

Se for isentrópico:

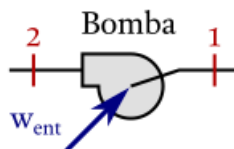
$$s_2 = s_1 \quad (4)$$

Se não for isentrópico:

$$h_2 = h_1 - \eta \cdot (h_1 - h_{2,s=s_1}) \quad (5)$$

### 2.1.2 Bomba

**Figura 2 – Representação esquemática de uma bomba. Os rótulos 1 e 2 representam, estados de entrada e saída da bomba, respectivamente, do fluido de trabalho. A seta azul indica interação energética de trabalho, juntamente com o rótulo  $w$ , com subscrito  $ent$  indicando a direção de entrada da interação. O escoamento do fluido de trabalho se dá no sentido 1-2 em regime de compressão.**



**Fonte: Naaktgeboren (2023).**

O processo do componente representado na Figura 2 vai aumentar a pressão e utilizar trabalho, além disso, o processo pode ser isentrópico.

O trabalho consumido pelo componente é descrito pelas equações:

$$\dot{W}_{ent} = \dot{m}_2 \cdot h_2 - \dot{m}_1 \cdot h_1 \quad (6)$$

e

$$w_{ent} = h_2 - h_1 \quad (7)$$

O componente pode ter as propriedades descritas na Equação 8 e Equação 9, dependendo das características da bomba.

Se for isentrópico:

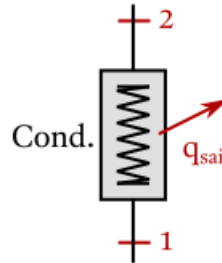
$$s_2 = s_1 \quad (8)$$

Demais casos:

$$h_2 = h_1 - \frac{h_1 - h_{2,s=s_1}}{\eta} \quad (9)$$

### 2.1.3 Condensador

**Figura 3 – Representação esquemática de um condensador. Os rótulos 1 e 2 representam, estados de entrada e saída do condensador, respectivamente, do fluido de trabalho. A seta vermelha indica interação energética de calor, juntamente com o rótulo  $q$ , com subscrito *sai* indicando a direção de saída da interação. O escoamento do fluido de trabalho se dá no sentido 1-2 em um processo isobárico, que vai transferir calor para fora do sistema fazendo o estado de saída do componente ser líquido comprimido saturado.**



Fonte: Naaktgeboren (2023).

O calor transferido para fora do componente representado na Figura 3 é descrito pelas equações:

$$\dot{Q}_{sai} = \dot{m}_1 \cdot h_1 - \dot{m}_2 \cdot h_2 \quad (10)$$

e

$$q_{sai} = h_1 - h_2 \quad (11)$$

A pressão se manterá constante, representada por:

$$P_2 = P_1 \quad (12)$$

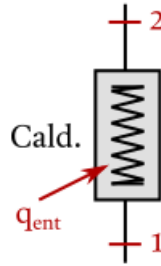
O estado de saída do componente será líquido comprimido saturado, representado por:

$$x_2 = 0 \quad (13)$$



### 2.1.4 Caldeira

**Figura 4 – Representação esquemática de uma caldeira. Os rótulos 1 e 2 representam, estados de entrada e saída da caldeira, respectivamente, do fluido de trabalho. A seta vermelha indica interação energética de calor, juntamente com o rótulo  $q$ , com subscrito  $ent$  indicando a direção de entrada da interação. O escoamento do fluido de trabalho se dá no sentido 1-2 em um processo que pode ser isobárico, se não houver perda de carga, e vai transferir calor para dentro do sistema.**



Fonte: Naaktgeboren (2023).

O calor transferido para dentro do componente representado na Figura 4 é descrito pelas equações:

$$\dot{Q}_{ent} = \dot{m}_2 \cdot h_2 - \dot{m}_1 \cdot h_1 \quad (14)$$

e

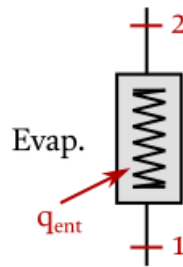
$$q_{ent} = h_2 - h_1 \quad (15)$$

A pressão pode se manter constante, dependendo das características da caldeira, representada por:

$$P_2 = P_1 \quad (16)$$

### 2.1.5 Evaporador

**Figura 5 – Representação esquemática de um evaporador. Os rótulos 1 e 2 representam, estados de entrada e saída do evaporador, respectivamente, do fluido de trabalho. A seta vermelha indica interação energética de calor, juntamente com o rótulo  $q$ , com subscrito  $ent$  indicando a direção de entrada da interação. O escoamento do fluido de trabalho se dá no sentido 1-2 em um processo que pode ser isobárico, se não houver perda de carga, e vai transferir calor para dentro do sistema fazendo o estado de saída do componente ser vapor saturado.**



**Fonte: Naaktgeboren (2023).**

O calor transferido para dentro do componente representado na Figura 5 é descrito pelas equações:

$$\dot{Q}_{ent} = \dot{m}_2 \cdot h_2 - \dot{m}_1 \cdot h_1 \quad (17)$$

e

$$q_{ent} = h_2 - h_1 \quad (18)$$

O estado de saída do componente será vapor saturado, representado por:

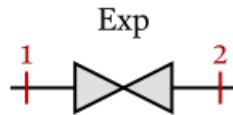
$$x_2 = 1 \quad (19)$$

A pressão pode se manter constante, dependendo das características do evaporador, representada por:

$$P_2 = P_1 \quad (20)$$

### 2.1.6 Válvula de expansão

**Figura 6 – Representação esquemática de uma válvula de expansão. Os rótulos 1 e 2 representam, estados de entrada e saída da válvula de expansão, respectivamente, do fluido de trabalho. O escoamento do fluido de trabalho se dá no sentido 1-2 em um processo que reduz a pressão e pode ser isoentálpico, se também for adiabático.**



Fonte: Naaktgeboren (2023).

Se o processo do componente representado na Figura 6 for adiabático seguirá a equação:

$$h_2 = h_1 \quad (21)$$

Caso não seja adiabático, então calor será transferido pelo componente, descrito pelas equações:

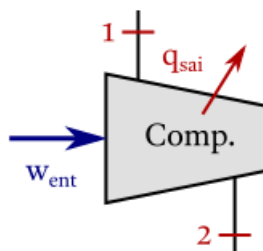
$$\dot{Q} = \dot{m}_2 \cdot h_2 - \dot{m}_1 \cdot h_1 \quad (22)$$

e

$$q = h_2 - h_1 \quad (23)$$

### 2.1.7 Compressor

**Figura 7 – Representação esquemática de um compressor. Os rótulos 1 e 2 representam, estados de entrada e saída do compressor, respectivamente, do fluido de trabalho. As setas coloridas indicam interações energéticas, sendo as de calor indicadas em vermelho, e as de trabalho em seta azul mais espessa, juntamente com os respectivos rótulos  $q$  ou  $w$ , com subscrito indicando a direção da interação, seja *ent* e *sai*, respectivamente para entrando e saindo. O escoamento do fluido de trabalho se dá no sentido 1-2 em regime de compressão.**



Fonte: Naaktgeboren (2023).

O processo do componente representado na Figura 7 vai aumentar a pressão e utilizar trabalho, além disso, o processo pode ser isentrópico.

O trabalho consumido pelo componente é descrito pelas equações:

$$\dot{W}_{ent} = \dot{m}_2 \cdot h_2 - \dot{m}_1 \cdot h_1 - \dot{Q}_{sai} \quad (24)$$

e

$$w_{ent} = h_2 - h_1 - q_{sai} \quad (25)$$

O componente pode ter as propriedades descritas na Equação 26, Equação 27 e Equação 28, dependendo das características do compressor.

Se o processo do componente for adiabático:

$$\dot{Q}_{sai} = q_{sai} = 0 \quad (26)$$

Se for isentrópico:

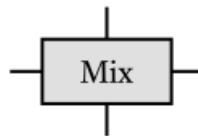
$$s_2 = s_1 \quad (27)$$

Se não for isentrópico:

$$h_2 = h_1 - \frac{h_1 - h_{2,s=s_1}}{\eta} \quad (28)$$

### 2.1.8 Câmara de mistura

**Figura 8 – Representação esquemática de uma câmara de mistura. Neste componente é realizada a mistura de dois ou mais estados por meio da equação de balanço de energia. Cada linha ligada ao retângulo representa um estado, que pode ser de entrada ou saída.**



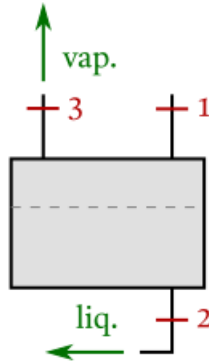
**Fonte: Naaktgeboren (2023).**

O componente representado na Figura 8 poder ter varias entradas e saídas, e seu processo é descrito pela equação:

$$\sum \dot{m}_{ent} \cdot h_{ent} = \sum \dot{m}_{sai} \cdot h_{sai} \quad (29)$$

## 2.1.9 Separador

**Figura 9 – Representação esquemática de um separador. O rótulo 1 representa o estado de entrada e os rótulos 2 e 3 representam os estados de saída do separador, que são líquido comprimido saturado e vapor saturado, respectivamente. O componente irá separar a parte líquida e gasosa, de acordo com o título do estado de entrada.**



Fonte: Naaktgeboren (2023).

O estado de saída com rótulo 3 da Figura 9 será vapor saturado, representado por:

$$x_3 = 1 \quad (30)$$

e a fração de massa que vai para essa saída é proporcional ao título do estado de entrada, representada por:

$$\dot{m}_3 = \dot{m}_1 \cdot x_1 \quad (31)$$

O estado de saída com rótulo 2 da Figura 9 será líquido comprimido saturado, representado por:

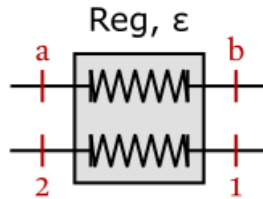
$$x_2 = 0 \quad (32)$$

e a fração de massa que vai para essa saída é proporcional ao título do estado de entrada, representada por:

$$\dot{m}_2 = \dot{m}_1 \cdot (1 - x_1) \quad (33)$$

## 2.1.10 Trocador de calor

**Figura 10 – Representação esquemática de um trocador de calor. Os rótulos  $a$  e  $b$  representam respectivamente os estados de entrada e saída de uma determinada corrente. Já os rótulos 1 e 2 representam respectivamente os estados de entrada e saída de uma outra corrente, que não se misturam com a primeira corrente. Neste componente há a interação entre diferentes correntes, sem haver a mistura entre elas. Essa interação depende da efetividade  $\epsilon$  do trocador de calor.**



Fonte: Naaktgeboren (2023).

Se a efetividade do componente representado na Figura 10 for de 100%, a relação entre as correntes será dada pela equação de balanço de energia, caso contrário a relação dependerá das capacidades térmicas das correntes que estão interagindo.

- Se a capacidade térmica da corrente inferior for maior que a superior seguirá as equações:

$$h_b = h_a + \epsilon \cdot \frac{\dot{m}_1}{\dot{m}_a} \cdot (h_1 - h_x) \quad (34)$$

e

$$h_2 = h_1 - \epsilon \cdot (h_1 - h_x) \quad (35)$$

Considerando  $h_x$  a entalpia do estado em que a temperatura é a mesma do estado de entrada da corrente superior, e a pressão a mesma da corrente inferior.

- Se a capacidade térmica da corrente superior for maior que a inferior seguirá as equações:

$$h_2 = h_1 + \epsilon \cdot \frac{\dot{m}_a}{\dot{m}_1} \cdot (h_a - h_x) \quad (36)$$

e

$$h_b = h_a - \epsilon \cdot (h_a - h_x) \quad (37)$$

Considerando  $h_x$  a entalpia do estado em que a temperatura é a mesma do estado de entrada da corrente inferior, e a pressão a mesma da corrente superior.

## 2.2 Cálculo do desempenho dos ciclos

Os ciclos termodinâmicos podem ter diferentes objetivos e é importante ter meios de avaliar a performance de cada um, para analisar se cumprem com seu propósito. Para isso há expressões diferentes para avaliar diferentes tipos de ciclos. A seguir serão apresentados os meios para o cálculo da eficiência de cada ciclo com base em Çengel e Boles (2013) e Struchtrup (2014).

### 2.2.1 Ciclos de geração de potência

Um ciclo de geração de potência busca transformar calor em trabalho, e seu desempenho pode ser medido com a eficiência térmica, que pode ser calculada pela equação:

$$\eta_t = \frac{w_{sai} - w_{ent}}{q_{ent}} \quad (38)$$

### 2.2.2 Ciclos de refrigeração

Já um ciclo de refrigeração busca transferir calor de um ambiente mais frio para um mais quente, e seu desempenho é verificado com o Coeficiente de Performance (COP), que se refere a equação:

$$COP = \frac{q_{ent}}{w_{ent}} \quad (39)$$

## 2.3 Imbalanços e Geração de entropia

Para validar os resultados obtidos, e comprovar que os cálculos das propriedades do sistema estão corretos, será obtido os valores de geração de entropia e imbalanços de massa e de energia.

### 2.3.1 Imbalanço de massa

Com base em Çengel e Boles (2013), o cálculo do balanço de massa é feito pela equação:

$$\sum \dot{m}_{sai} - \sum \dot{m}_{ent} = 0 \quad (40)$$

Com isso, o balanço de massa será validado se o resultado após a substituição dos valores obtidos for igual a zero.

### 2.3.2 Imbalanço de energia

Com base em Çengel e Boles (2013), o cálculo do balanço de energia é feito pela equação:

$$\sum \dot{m}_{sai} \cdot h_{sai} - \sum \dot{m}_{ent} \cdot h_{ent} = 0 \quad (41)$$

Com isso, o balanço de energia será validado se o resultado após a substituição dos valores obtidos for igual a zero.

### 2.3.3 Geração de entropia

Com base em Çengel e Boles (2013), o cálculo da geração de entropia é feito pela equação:

$$\dot{S}_{gen} = \sum \dot{m}_{sai} \cdot s_{sai} - \sum \dot{m}_{ent} \cdot s_{ent} - \sum \frac{\dot{Q}_k}{T_k} \quad (42)$$

Com isso, o valor da geração de entropia será validado se o resultado após a substituição dos valores obtidos for superior a zero, para que cumpra com a segunda lei da termodinâmica, que estabelece que a entropia não pode ser destruída, segundo Çengel e Boles (2013).



## **3 METODOLOGIA**

### **3.1 Levantamento de variáveis do sistema**

O comportamento dos ciclos termodinâmicos é influenciado por fatores, como o fluido utilizado e os componentes presentes no sistema, junto com a sua distribuição.

#### **3.1.1 Fluido de trabalho**

A escolha do fluido de trabalho pode ter grande impacto no comportamento do ciclo, devido às diferenças entre as propriedades termodinâmicas, como o calor específico e a temperatura crítica.

Portanto é essencial definir o tipo de fluido de trabalho e ter ferramentas que possibilitem a obtenção das propriedades térmicas referentes a cada estado do ciclo, levando em consideração o fluido utilizado.

#### **3.1.2 Componentes do ciclo**

Cada componente apresenta características diferentes, fazendo com que a ordem de distribuição dos componentes seja fundamental para definir o funcionamento do ciclo termodinâmico.

Assim, é necessário desenvolver um método adequado para obter e armazenar as informações referentes à formatação do ciclo em análise, pois a estrutura seguirá o comportamento de um grafo, podendo ainda apresentar bifurcações, fazendo com que o ciclo tenha mais de um caminho.

### **3.2 Desenvolvimento do código**

#### **3.2.1 Inserção das informações do ciclo**

Levando em consideração que a formatação do ciclo, incluindo a ordem dos componentes, exerce uma influência significativa em suas propriedades, torna-se fundamental o emprego de ferramentas que possibilitem a inserção dessas informações com facilidade pelo usuário.

Portanto, escolheu-se adotar a linguagem de programação Julia, que apresenta suporte para metaprogramação, o qual permite a criação de uma sintaxe de programação específica para a aplicação em questão.

Metaprogramação é uma técnica que permite manipular o código em tempo de execução antes da compilação, permitindo aos desenvolvedores criar uma estrutura de codificação

personalizada e gerenciar a entrada de código nessa formatação antes da compilação. Julia fornece funcionalidades para suportar a metaprogramação, sendo uma delas o uso de macros, que são funções que recebem expressões como argumentos e retornam novas expressões que são então avaliadas pelo compilador. O suporte da Julia para metaprogramação a torna uma ferramenta poderosa para gerar códigos eficientes e flexíveis em tempo de execução (BEZANSON *et al.*, 2012).

Com esse recurso, é possível criar uma biblioteca de programação própria que interprete a formatação dos ciclos termodinâmicos, simplificando a inserção dessas informações pelo usuário. Além disso, é possível declarar as variáveis do sistema através da manipulação do código de entrada. Com isso, a complexidade do código gerado pelo usuário é reduzida.

### 3.2.2 Controle de equações do ciclo

Na resolução dos ciclos haverá várias equações para cada componente no processo, e para alguns ciclos, será necessário o uso de sistemas de equações para essa resolução, com isso é essencial uma forma de manipular essas equações para que possam ser resolvidas de maneira ordenada e realizar a verificação se algum subconjunto dessas equações pode ser formado para gerar um sistema de equações.

A biblioteca Symbolics (2023) é uma ferramenta importante para pesquisadores e profissionais em áreas da engenharia que precisam realizar cálculos simbólicos complexos rapidamente e com precisão (GOWDA *et al.*, 2021). Essa biblioteca pode fazer o gerenciamento de variáveis das equações, transformando-as em símbolos e gerando uma estrutura da equação com a sequência de operações que esses símbolos estão envolvidos.

Dessa forma, a biblioteca é uma ferramenta capaz de lidar com a manipulação de equações. Além de permitir o armazenamento de equações e suas variáveis correspondentes, a biblioteca também pode resolver sistemas de equações e equações isoladas de maneira eficiente. Outra característica é a possibilidade de simplificação, expansão e fatoração de equações, o que pode ser adequado para reduzir o tamanho e a complexidade das equações.

Uma alternativa para atingir esse objetivo é a biblioteca SymPy (2021), porém ela é desenvolvida em Python e requer o PyCall (2023) para funcionar, enquanto o Symbolics (2023) é escrito em Julia. Portanto, o Symbolics (2023) tem uma maior performance e não depende de outras ferramentas para sua instalação, fazendo ele mais vantajoso em relação ao uso da SymPy (2021).

Assim sendo, a biblioteca Symbolics (2023) atende às demandas exigidas e destaca-se em relação às ferramentas alternativas, devido à sua melhor performance, o que justifica sua utilização.

### 3.2.3 Manipulação de equações condicionais

Existem componentes que possuem propriedades condicionais, como o trocador de calor, em que a obtenção das equações que caracterizam o seu comportamento depende do resultado de uma inequação.

Desse modo, será utilizada uma classe para manipular esse tipo de equação. Tendo como seus atributos a condição que será analisada, o conjunto de equação caso a condição seja verdadeira, e o conjunto de equação caso a condição seja falsa.

Dessa maneira, quando a inequação for resolvida, o conjunto de equações referentes a resposta da condição será adicionado nas equações do sistema. Com essa abordagem o controle das equações condicionais fica mais eficiente, separando-as das equações convencionais e fazendo um tratamento específico.

### 3.2.4 Obtenção das propriedades térmicas

Para obter as propriedades termodinâmicas dos estados necessárias para o desenvolvimento do ciclo, a biblioteca CoolProp (2023) será empregada.

CoolProp (2023) é uma biblioteca de propriedades termodinâmicas de código aberto que fornece dados termodinâmicos e de transporte para mais de 110 fluidos puros e pseudo-puros. Projetada para ser fácil de usar e bem documentada, a biblioteca é capaz de calcular várias propriedades termodinâmicas, como entalpia, entropia e capacidade calorífica específica, a partir de outras propriedades conhecidas (BELL *et al.*, 2014).

### 3.2.5 Gerenciamento dos estados termodinâmicos

Cada estado termodinâmico possui vários atributos que o caracteriza. Sendo um grupo de atributos envolvendo as suas propriedades térmicas, que são obtidos pela biblioteca CoolProp (2023), e outros atributos que completam a sua identidade como o ciclo que o estado pertence, o fluido e o fluxo mássico do estado.

O controle dos estados envolve a análise se o estado possui duas propriedades térmicas compatíveis que possibilitem a obtenção das outras propriedades desconhecidas. Com isso é importante ter a estrutura de cada estado bem definida, então as características dos estados são formatadas como uma classe, para organizar cada estado como um objeto único, facilitando a sua manipulação e possibilitando a avaliação da presença de duas propriedades que permitam a obtenção das restantes.

### 3.2.6 Formulação de hipóteses

Em algumas aplicações não haverá informações suficientes para calcular todas as propriedades requeridas, como casos que o resultado das inequações das equações condicionais não podem ser obtidas. Com isso, para continuar com a resolução do sistema hipóteses devem ser formuladas.

Essas hipóteses se referem à suposição de valores desconhecidos que possibilitem a continuação do cálculo das outras propriedades do sistema. Com isso, no casos das equações condicionais não resolvidas, um resultado para a condição da equação pode ser pressuposto.

Dessa forma, com a aplicação da hipótese formulada e depois de fazer os cálculos das propriedades restantes será verificado se a hipótese se confirma. Essa análise será feita pela verificação do cumprimento dos resultados obtidos com as equações iniciais.

Caso a hipótese se prove falsa, o estado do sistema deve ser restaurado ao estado anterior à adoção da hipótese e então formular uma hipótese diferente, e caso a hipótese seja verdadeira, o sistema continua com a sua execução.

### 3.2.7 Gerenciamento dos ciclos do sistema

O sistema em estudo pode ser composto por mais de um ciclo. Com isso é necessário fazer uma análise individual de cada ciclo e uma análise geral do sistema considerando as interações entre os ciclos.

Com esse fim é importante a organização dos atributos dos ciclos. A partir disso, os ciclos são estruturados em uma classe que contém a lista de estados do ciclo e as características termodinâmicas do ciclo, além de informações como o fluido do ciclo, se o ciclo é a vapor, se é o ciclo é de refrigeração e o fluxo mássico principal do ciclo se existir.

### 3.2.8 Cálculo das características termodinâmicas

Para a avaliação do sistema e verificar o seu desempenho é preciso ter as informações das características termodinâmicas, que envolvem as transferências de calor para dentro e para fora do sistema, os trabalhos de entrada e saída e a eficiência térmica.

Dessa forma, as equações que definem essas informações são geradas na inicialização do ciclo, nas funções de cada componente, estabelecendo as características termodinâmicas individualmente para os componentes dos ciclos.

A partir desse ponto, as equações geradas serão executadas e com isso se obtém as características de cada componente e com base nisso, para definir as características termodinâmicas de cada ciclo e do sistema, se deve avaliar em quais subsistemas cada componente pertence.

A única propriedade que não será calculada dessa forma é a eficiência térmica, que será calculada isoladamente para cada subsistema a partir das outras características termodinâmicas obtidas anteriormente.

### 3.2.9 Apresentação dos resultados

Visando ter uma análise completa dos resultados obtidos é fundamental ter uma interface que apresente as informações de forma organizada e clara. Com base nisso, uma tabela deve ser gerada a partir dos resultados do ciclo, contendo as propriedades térmicas de cada estado, suas frações de fluxo mássico e fluxos mássicos, se existirem.

Com o objetivo de resumir o ciclo, uma tabela com os trabalhos realizados e os calores trocados será gerada, mostrando essas propriedades para cada componente e o total do ciclo. Além disso, a eficiência do ciclo também deve estar na tabela, e para fornecer uma visão geral do ciclo, é importante que um gráfico temperatura-entropia seja ilustrado, para indicar a sequência do fluxo dos componentes e as suas influências nas propriedades do ciclo.

Para a produção das tabelas e gráficos, serão utilizadas as bibliotecas *PrettyTables* (2023) e *JuliaPlots* (2023), respectivamente. Posteriormente, para melhor organização, esses elementos serão formatados utilizando *HyperText Markup Language* (HTML) e *Cascading Style Sheet* (CSS), de modo que todas as informações relevantes do ciclo sejam apresentadas de forma clara ao usuário. Além disso, é relevante destacar que a formatação destes elementos pode ser realizada de forma nativa em Julia, não requerendo nenhuma outra ferramenta externa.

### 3.2.10 Manipulação dos resultados

Além da possibilidade de visualizar os resultados graficamente, também é importante ter métodos de extrair as informações por meio de código, a fim de habilitar a manipulação dos dados.

Para alcançar esse objetivo, será estruturado um sistema que permitirá obter as propriedades desejadas de forma intuitiva. Dessa forma, será possível realizar uma análise mais detalhada dos resultados do ciclo, fornecendo uma compreensão mais abrangente do comportamento do sistema.

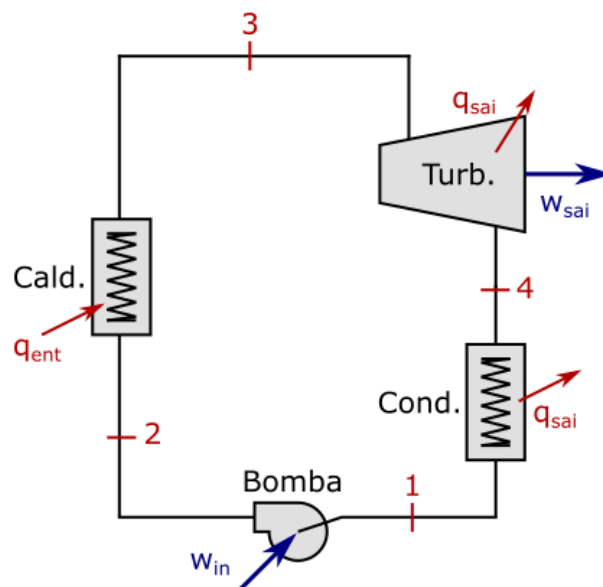
## 4 RESULTADOS

A seguir haverá a demonstração do funcionamento do código, mostrando a estrutura da programação, com base na formatação do ciclo em análise e as propriedades conhecidas, e os resultados obtidos com a execução do programa.

### 4.1 Exemplo 1 - Ciclo Rankine simples ideal

A Figura 11 ilustrará a formatação do ciclo e as propriedades conhecidas.

**Figura 11 – Representação do ciclo Rankine simples, em que se tem uma bomba, uma caldeira, uma turbina e um condensador.**



**Fonte: Autoria própria (2023).**

As propriedades do ciclo apresentado na Figura 11 são:

- Temperatura na saída da caldeira de 250 °C;
- Pressão antes da turbina de 1000 kPa;
- Pressão depois da turbina de 10 kPa.

Com base nessas informações, o código para resolução do ciclo pode ser desenvolvido conforme a Listagem 9.

**Listagem 1 – Código do exemplo 1**

```

1 using CycleSolver
2
3 @solve begin
4     st3.T = 250 + 273
5     st3.p = 1000
6     st1.p = 10
7
8     newCycle []
9         pump(st1 , st2)
10        boiler(st2 , st3)
11        turbine(st3 , st4)
12        condenser(st4 , st1)
13 end
14
15 PrintResults ()

```

**Fonte: Autoria própria (2023).**

A programação do ciclo será feita dentro da macro `@solve`. Uma macro é uma função que leva um código que pode ser manipulado antes de ser compilado.

Cada estado é declarado usando o prefixo 'st' com o número do estado na representação do ciclo, mas os estados poderiam ser nomeados de outra forma caso necessário.

Para definir as propriedades já conhecidas do sistema, deve-se especificar o estado e o símbolo da propriedade conhecida, seguindo a Tabela 1.

**Tabela 1 – Relação das propriedades termodinâmicas e símbolos dos atributos dos estados.**

Propriedade termodinâmicas	Unidade	Símbolo dos atributos do estado
Temperatura	[K]	.T
Pressão	[kPa]	.p
Entalpia	[kJ/kg]	.h
Entropia	[kJ/kg·K]	.s
Titulo		.Q
Densidade	[kg/m <sup>3</sup> ]	.rho
Fluxo mássico	[kg/s]	.m

**Fonte: Autoria própria (2023).**

Com base nisso, para atribuir as propriedades, é concatenado ao nome do estado com o símbolo do atributo conhecido, seguido pelo sinal de igualdade e pelo valor que se deseja atribuir, levando em conta as unidades de cada propriedade.

A definição das propriedades conhecidas pode ser feita em qualquer linha do código, já a especificação da estrutura do ciclo deve ser feita depois da inicialização do ciclo, com o comando `newCycle []`, que inicializa um ciclo utilizando água como fluido de trabalho.

Depois da inicialização, a estrutura do ciclo é criada, usando funções nomeadas com base nos componentes que elas representam e o estado de entrada do componente sendo informado no primeiro parâmetro da função e o estado de saída sendo informado no segundo parâmetro.

Após a programação do ciclo ser realizada dentro da macro, os resultados são apresentados por meio de tabelas e gráficos chamando a função `PrintResults()`. Para o ciclo programado, os resultados obtidos são apresentados nas Tabela 2 e na Tabela 3.

**Tabela 2 – Propriedades de cada estado do exemplo 1**

<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
st1	318.956	10.0	191.806	0.6492	0.0	...	1.0
st2	318.989	1000.0	192.806	0.6492	...	...	1.0
st3	523.0	1000.0	2942.79	6.9258	...	...	1.0
st4	318.956	10.0	2193.78	6.9258	0.8369	...	1.0

**Fonte: Autoria própria (2023).**

**Tabela 3 – Propriedades do ciclo do exemplo 1**

	<i>Total</i>	<i>Component</i>	<i>Value</i>
<i>q<sub>in</sub></i>	2749.9849 kJ/kg	boiler: st2 » st3	2749.9849 kJ/kg
<i>Q<sub>in</sub></i>	...		...
<i>q<sub>out</sub></i>	2001.9783 kJ/kg	condenser: st4 » st1	2001.9783 kJ/kg
<i>Q<sub>out</sub></i>	...		...
<i>w<sub>in</sub></i>	1.0 kJ/kg	pump: st1 » st2	1.0 kJ/kg
<i>W<sub>in</sub></i>	...		...
<i>w<sub>out</sub></i>	749.0066 kJ/kg	turbine: st3 » st4	749.0066 kJ/kg
<i>W<sub>out</sub></i>	...		...
		Thermal efficiency ( <i>n</i> ) =	27.2004 %

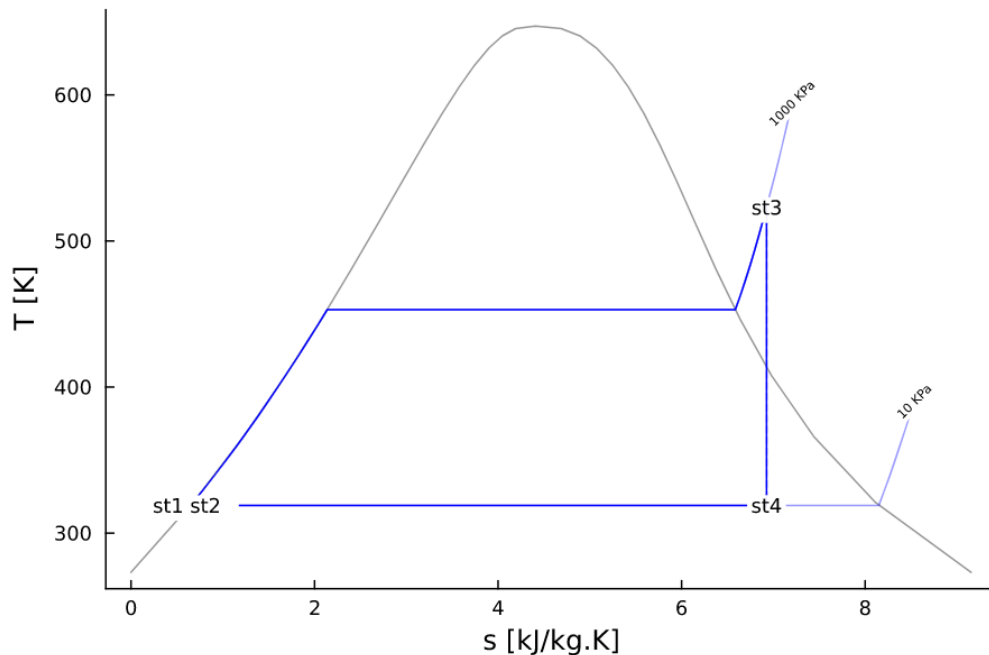
**Fonte: Autoria própria (2023).**

A função retorna a Tabela 2 com as propriedades de cada estado do ciclo, e a Tabela 3 com as propriedade termodinâmicas do ciclo, descrevendo os processos de transferência de calor e realização de trabalho ocorridos no ciclo, além de mostrar os componentes responsáveis por esses processos, e concluindo com a eficiência térmica do ciclo.

Com a Tabela 2 é possível analisar as etapas do ciclo em que a pressão se mantém constante, e as etapas que a entropia se mantém constante. Além de visualizar a variação das outras propriedades do ciclo.

Na Tabela 3 pode-se analisar individualmente o comportamento dos componentes em relação a direção da interação de transferência de calor ou de realização de trabalho, sendo de entrada ou saída.



Gráfico 1 – Diagrama  $T - s$  do exemplo 1

Fonte: Autoria própria (2023).

Além das tabelas, também é gerado um gráfico que relaciona a temperatura e a entropia de cada estado do ciclo, apresentando o trajeto desenvolvido pelo ciclo em relação ao domo e as linhas de pressão constante.

Através do Gráfico 1 se obtém uma melhor visualização dos processos do ciclo, podendo visualizar a isentropia no processo da turbina, e a constância da pressão nos processos da caldeira e do condensador.

Com o comando `CycleSolver.SystemCycles` é possível obter a lista de ciclos, para poder analisar e manipular os dados relacionados aos ciclos termodinâmicos.

Cada ciclo possui propriedades termodinâmicas específicas, as quais podem ser acessadas. Por exemplo, para obter as propriedades termodinâmicas do ciclo no índice 1, basta utilizar `CycleSolver.SystemCycles[1].thermoProperties`. Dentro dessas propriedades, é possível acessar informações como a transferência de calor para dentro do ciclo, representada por `CycleSolver.SystemCycles[1].thermoProperties.qin`, e a eficiência do ciclo, acessada com `CycleSolver.SystemCycles[1].thermoProperties.n`.

Além disso, cada ciclo é composto por uma série de estados termodinâmicos. Para visualizar a lista de estados no ciclo de índice 1, é utilizado o comando `CycleSolver.SystemCycles[1].states`.

Para acessar as propriedades específicas de cada estado, além da sintaxe apresentada anteriormente, também é possível obter as propriedades com o comando `CycleSolver.st1`, que proporciona acesso às propriedades do estado identificado como 'st1'. Para visualizar a entalpia deste estado, basta utilizar `CycleSolver.st1.h`.

É importante ressaltar que, caso haja mais de um ciclo, além das propriedades individuais de cada ciclo, haverá também propriedades referentes ao sistema de ciclos como um todo. Estas propriedades podem ser acessadas através do comando `CycleSolver.System`. A transferência de calor para dentro do sistema pode ser visualizada com `CycleSolver.System.qin`, enquanto a eficiência do sistema é obtida através de `CycleSolver.System.n`.

#### 4.2 Exemplo 2 - Ciclo Rankine simples real

Nesse exemplo, além de considerar a mesma formatação e valores conhecidos que o exemplo da seção 4.1, a eficiência isentrópica da bomba e turbina também será observada, possuindo os seguintes valores:

- A bomba possui eficiência isentrópica de 90%;
- A turbina possui eficiência isentrópica de 80%.

##### Listagem 2 – Código do exemplo 2.

```

1 using CycleSolver
2
3 @solve begin
4     st3.T = 250 + 273
5     st3.p = 1000
6     st1.p = 10
7
8     newCycle []
9         pump(st1, st2, 90)
10        boiler(st2, st3)
11        turbine(st3, st4, 80)
12        condenser(st4, st1)
13 end
14
15 PrintResults ()

```

Fonte: Autoria própria (2023).

Para inclusão das eficiência isentrópica dos componentes na programação do ciclo, o valor em porcentagem é definido no terceiro parâmetro da função do componente, depois da definição dos estados de entrada e saída.

Considerando a eficiência isentrópica da bomba e da turbina os resultados obtidos são apresentados na Tabela 4.

Tabela 4 – Propriedades de cada estado do exemplo 2.

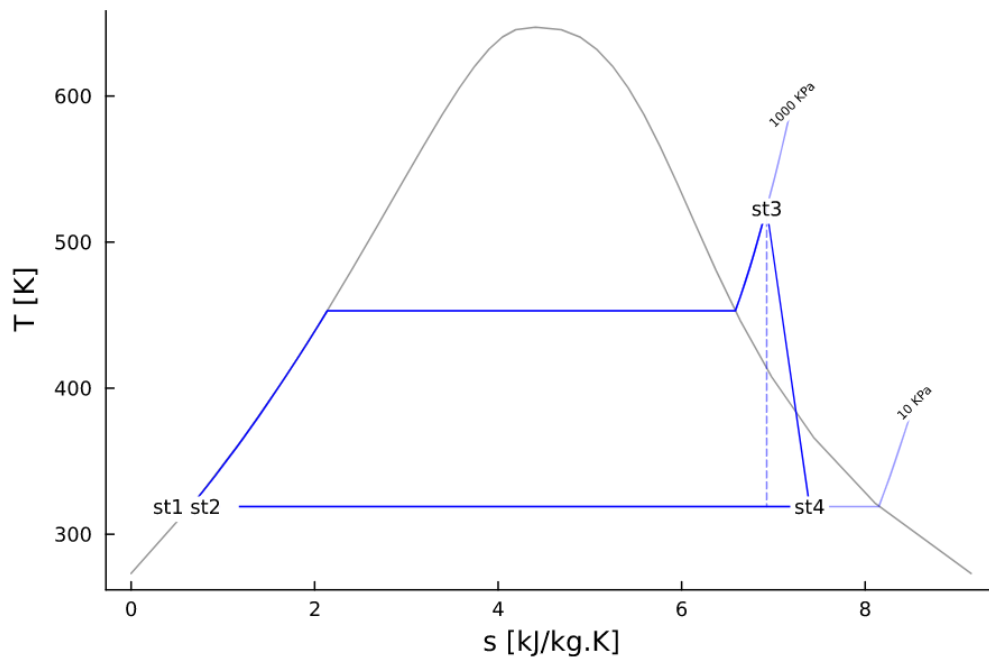
<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
st1	318.956	10.0	191.806	0.6492	0.0	...	1.0
st2	319.016	1000.0	192.917	0.6495	...	...	1.0
st3	523.0	1000.0	2942.79	6.9258	...	...	1.0
st4	318.956	10.0	2343.59	7.3955	0.8996	...	1.0

Fonte: Autoria própria (2023).

Tabela 5 – Propriedades do ciclo do exemplo 2

	<i>Total</i>	<i>Component</i>	<i>Value</i>
<i>q<sub>in</sub></i>	2749.8738 kJ/kg	boiler: st2 » st3	2749.8738 kJ/kg
<i>Q<sub>in</sub></i>	...		...
<i>q<sub>out</sub></i>	2151.7796 kJ/kg	condenser: st4 » st1	2151.7796 kJ/kg
<i>Q<sub>out</sub></i>	...		...
<i>w<sub>in</sub></i>	1.1111 kJ/kg	pump: st1 » st2	1.1111 kJ/kg
<i>W<sub>in</sub></i>	...		...
<i>w<sub>out</sub></i>	599.2053 kJ/kg	turbine: st3 » st4	599.2053 kJ/kg
<i>W<sub>out</sub></i>	...		...
		Thermal efficiency ( <i>n</i> ) =	21.7499 %

Fonte: Autoria própria (2023).

Gráfico 2 – Diagrama  $T - s$  do exemplo 2

Fonte: Autoria própria (2023).

No Gráfico 2, além de ilustrar o processo real, também é ilustrado o processo caso ele fosse isentrópico, fornecendo uma comparação ao considerar a eficiência isentrópica dos componentes.

### 4.3 Exemplo 3 - Mudança do fluido do ciclo

Neste exemplo será apresentada a sintaxe para a troca de fluido de trabalho no ciclo. A formatação do ciclo é a mesma que a Figura 11, e os valores conhecidos são os seguintes:

- Refrigerante-134a é utilizado como fluido de trabalho;
- Temperatura na saída da caldeira de 70 °C;
- Pressão antes da turbina de 1000 kPa;
- Pressão depois da turbina de 10 kPa;
- A bomba possui eficiência isentrópica de 90%;
- A turbina possui eficiência isentrópica de 90%.

## Listagem 3 – Código do exemplo 3.

```

1 using CycleSolver
2
3 @solve begin
4     st3.T = 70 + 273
5     st3.p = 1000
6     st1.p = 10
7
8     newCycle[R134a]
9         pump(st1, st2, 90)
10        boiler(st2, st3)
11        turbine(st3, st4, 90)
12        condenser(st4, st1)
13 end
14
15 PrintResults()

```

Fonte: Autoria própria (2023).

A mudança no fluido de trabalho é feita no comando de inicialização do ciclo, inserindo o novo fluido entre colchetes no final do comando. Se nenhum fluido for especificado é utilizado água como fluido de trabalho.

Tabela 6 – Propriedades de cada estado do exemplo 3.

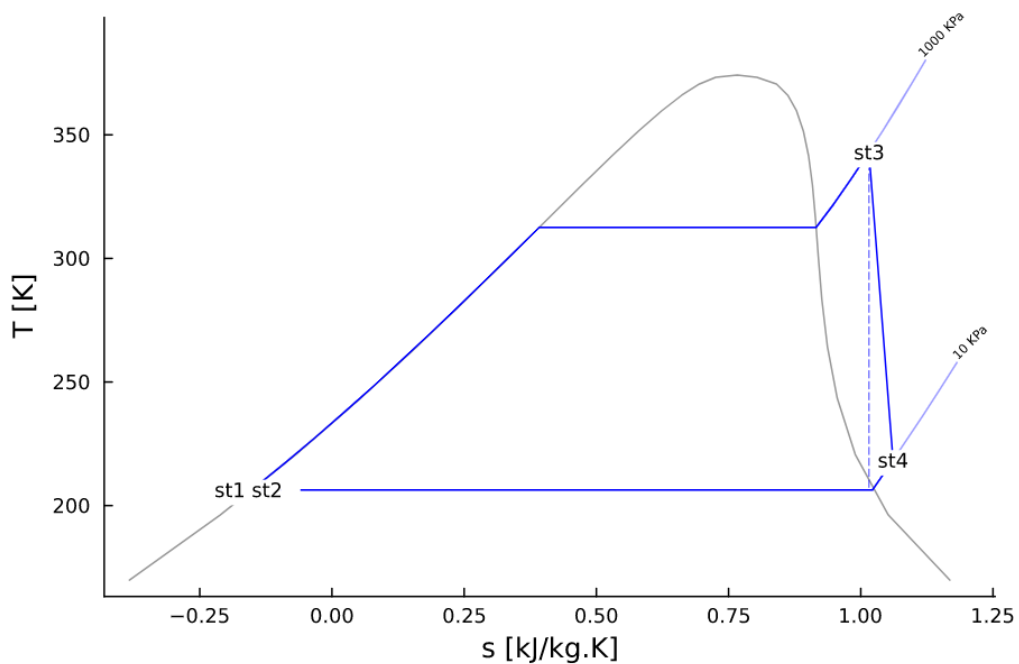
<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
st1	206.294	10.0	-33.1351	-0.1508	0.0	...	1.0
st2	206.561	1000.0	-32.3988	-0.1505	...	...	1.0
st3	343.0	1000.0	303.7	1.0155	...	...	1.0
st4	218.361	10.0	217.093	1.061	...	...	1.0

Fonte: Autoria própria (2023).

Tabela 7 – Propriedades do ciclo do exemplo 3

	<i>Total</i>	<i>Component</i>	<i>Value</i>
$q_{in}$	336.099 kJ/kg	boiler: st2 » st3	336.099 kJ/kg
$\dot{Q}_{in}$	...		...
$q_{out}$	250.2279 kJ/kg	condenser: st4 » st1	250.2279 kJ/kg
$\dot{Q}_{out}$	...		...
$w_{in}$	0.7364 kJ/kg	pump: st1 » st2	0.7364 kJ/kg
$\dot{W}_{in}$	...		...
$w_{out}$	86.6074 kJ/kg	turbine: st3 » st4	86.6074 kJ/kg
$\dot{W}_{out}$	...		...
Thermal efficiency ( $\eta$ ) =			25.5493 %

Fonte: Autoria própria (2023).

Gráfico 3 – Diagrama  $T - s$  do exemplo 3

Fonte: Autoria própria (2023).

#### 4.4 Exemplo 4 - Definir o fluxo mássico principal do ciclo

Neste exemplo será apresentada a sintaxe para a definição do fluxo mássico principal do ciclo. A formatação do ciclo e os valores conhecidos são os mesmos que o exemplo da seção 4.3, mas também será considerado um fluxo mássico de 3,5 kg/s.

## Listagem 4 – Código do exemplo 4

```

1 using CycleSolver
2
3 @solve begin
4     st3.T = 70 + 273
5     st3.p = 1000
6     st1.p = 10
7
8     newCycle[R134a: 3.5]
9         pump(st1, st2, 90)
10        boiler(st2, st3)
11        turbine(st3, st4, 90)
12        condenser(st4, st1)
13 end
14
15 PrintResults()

```

Fonte: A autoria própria (2023).

A definição do fluxo mássico também é feita no comando de inicialização do ciclo, logo depois de definir o fluido de trabalho do ciclo, dentro do colchetes o valor a ser atribuído é inserido depois de um sinal de dois pontos.

Tabela 8 – Propriedades de cada estado do exemplo 4.

<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
st1	206.294	10.0	-33.1351	-0.1508	0.0	3.5	1.0
st2	206.561	1000.0	-32.3988	-0.1505	...	3.5	1.0
st3	343.0	1000.0	303.7	1.0155	...	3.5	1.0
st4	218.361	10.0	217.093	1.061	...	3.5	1.0

Fonte: A autoria própria (2023).

Tabela 9 – Propriedades do ciclo do exemplo 4

	<i>Total</i>	<i>Component</i>	<i>Value</i>
$q_{in}$	336.099 kJ/kg	boiler: st2 » st3	336.099 kJ/kg
$\dot{Q}_{in}$	1176.3463 kW		1176.3463 kW
$q_{out}$	250.2279 kJ/kg	condenser: st4 » st1	250.2279 kJ/kg
$\dot{Q}_{out}$	875.7978 kW		875.7978 kW
$w_{in}$	0.7364 kJ/kg	pump: st1 » st2	0.7364 kJ/kg
$\dot{W}_{in}$	2.5773 kW		2.5773 kW
$w_{out}$	86.6074 kJ/kg	turbine: st3 » st4	86.6074 kJ/kg
$\dot{W}_{out}$	303.1258 kW		303.1258 kW
		Thermal efficiency ( $\eta$ ) =	25.5493 %

Fonte: Autoria própria (2023).

O gráfico temperatura-pressão gerado é o mesmo que o ciclo anterior, pois o fluxo mássico não influencia a sua geração.

Na primeira tabela é apresentado o fluxo mássico de cada estado, e com isso é possível calcular a taxa de transferência de calor e a taxa de realização de trabalho, que são apresentadas na segunda tabela.

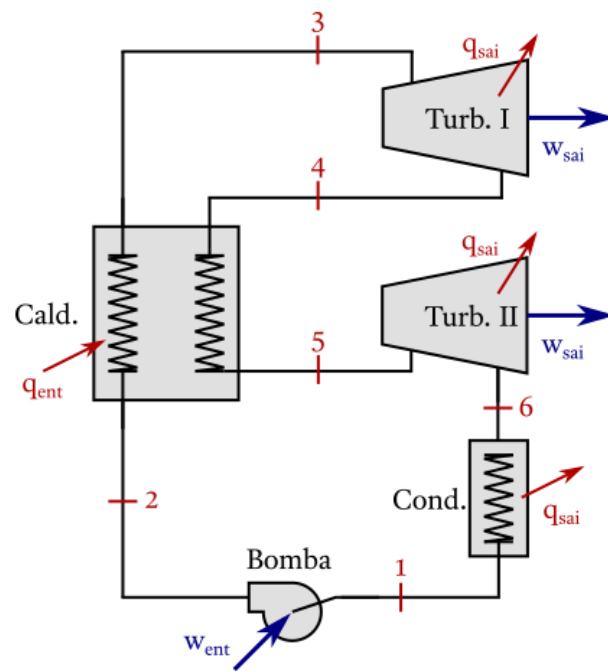
#### 4.5 Exemplo 5 - Ciclo Rankine supercrítico com reaquecimento

Este exemplo vai continuar com a demonstração dos resultados obtidos com o ciclo da Figura 12.

- Água é utilizada como fluido de trabalho;
- Pressão na saída da turbina de baixa pressão de 10 kPa;
- Pressão na saída da turbina de alta pressão de 200 kPa;
- Pressão na saída da bomba de 1000 kPa;
- Temperaturas depois de sair da caldeira de 250 °C.
- A bomba possui eficiência isentrópica de 90%;
- As turbinas possuem eficiência isentrópica de 80%.



Figura 12 – Representação do ciclo Rankine com reaquecimento, em que se tem uma bomba, duas caldeiras, duas turbinas e um condensador.



Fonte: Autoria própria (2023).

## Listagem 5 – Código do exemplo 5

```

1 using CycleSolver
2
3 @solve begin
4     st3.T = 600 + 273
5     st5.T = 300 + 273
6     st3.p = 30000
7     st5.p = 500
8     st1.p = 10
9
10    newCycle []
11        pump(st1 , st2 , 90)
12        boiler(st2 , st3)
13        turbine(st3 , st4 , 80)
14        boiler(st4 , st5)
15        turbine(st5 , st6 , 80)
16        condenser(st6 , st1)
17 end
18
19 PrintResults ()

```

Fonte: Autoria própria (2023).

Tabela 10 – Propriedades de cada estado do exemplo 5

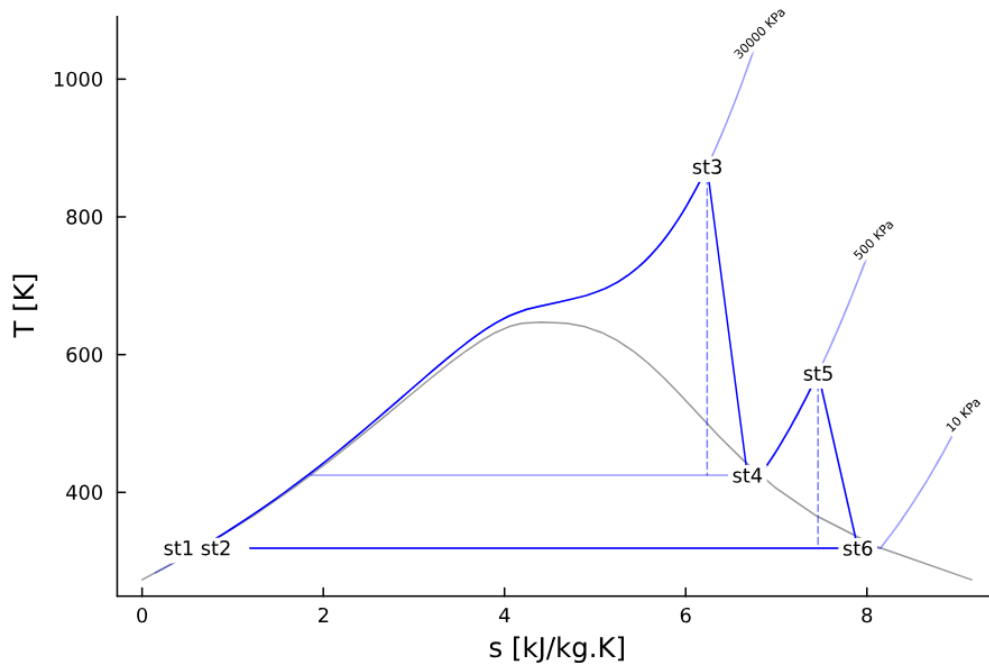
<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
st1	318.956	10.0	191.806	0.6492	0.0	...	1.0
st2	320.772	30000.0	225.261	0.6596	...	...	1.0
st3	873.0	30000.0	3446.26	6.2368	...	...	1.0
st4	424.981	500.0	2689.23	6.6821	0.9721	...	1.0
st5	573.0	500.0	3064.24	7.4609	...	...	1.0
st6	318.956	10.0	2504.4	7.8997	0.9668	...	1.0

Fonte: Autoria própria (2023).

Tabela 11 – Propriedades do ciclo do exemplo 5

	<i>Total</i>	<i>Component</i>	<i>Value</i>
$q_{in}$	3596.0169 kJ/kg	boiler: st2 » st3	3221.0036 kJ/kg
	...	boiler: st4 » st5	375.0133 kJ/kg
$\dot{Q}_{in}$	...		...
$q_{out}$	2312.5901 kJ/kg	condenser: st6 » st1	2312.5901 kJ/kg
	...		...
$\dot{Q}_{out}$	...		...
$w_{in}$	33.4554 kJ/kg	pump: st1 » st2	33.4554 kJ/kg
	...		...
$\dot{W}_{in}$	...		...
$w_{out}$	1316.8822 kJ/kg	turbine: st3 » st4	757.0347 kJ/kg
	...	turbine: st5 » st6	559.8475 kJ/kg
$\dot{W}_{out}$	...		...
Thermal efficiency ( $\eta$ ) =			35.6902 %

Fonte: Autoria própria (2023).

Gráfico 4 – Diagrama  $T - s$  do exemplo 5

Fonte: Autoria própria (2023).

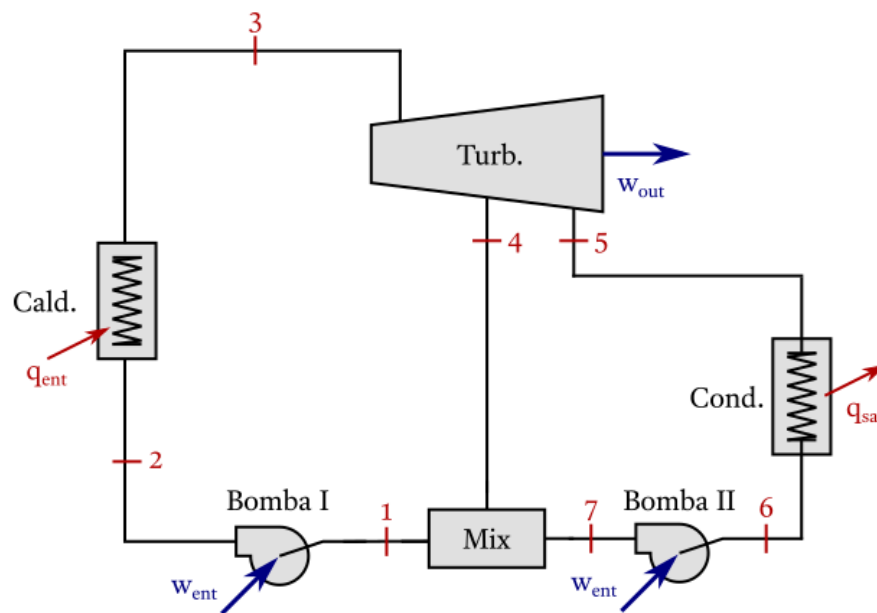
Como resultado da utilização de mais de um componente do mesmo tipo observa-se múltiplos componentes para cada propriedade termodinâmica do ciclo na Tabela 11, sendo apresentados os valores referentes aos componentes isolados e o valor total do ciclo, somando os valores isolados.

#### 4.6 Exemplo 6 - Ciclo Rankine regenerativo

Com o ciclo representado em Figura 13 será introduzida a sintaxe para a definição de componentes com mais de um estado de entrada ou saída, fazendo que o ciclo apresente ramificações.

- Água é utilizada como fluido de trabalho;
- Pressão que passa pelo condensador de 10 kPa;
- Pressão que vai para a câmara de mistura de 100 kPa;
- Pressão na caldeira de 1000 kPa;
- Temperaturas depois de sair da caldeira de 250 °C.

Figura 13 – Representação do ciclo Rankine regenerativo, em que se tem duas bombas, uma caldeira, uma turbina com duas saídas, um condensador, e uma câmara de mistura.



Fonte: Autoria própria (2023).

## Listagem 6 – Código do exemplo 6

```

1 using CycleSolver
2
3 @solve begin
4     st3.T = 250 + 273
5     st3.p = 1000
6     st4.p = 100
7     st5.p = 10
8
9     newCycle []
10        pump(st1 , st2)
11        boiler(st2 , st3)
12        turbine(st3 , [st4 , st5 ])
13        condenser(st5 , st6)
14        pump(st6 , st7)
15        heater_open([st7 , st4] , st1)
16 end
17
18 PrintResults ()

```

Fonte: Autoria própria (2023).

Para a definição que o componente possui múltiplos estados de entrada ou de saída, os estados são inseridos dentro de uma lista, que será o parâmetro de entrada ou de saída do componente.

Tabela 12 – Propriedades de cada estado do exemplo 6

<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
st1	372.756	100.0	417.504	1.3028	0.0	...	1.0
st2	372.818	1000.0	418.443	1.3028	...	...	1.0
st3	523.0	1000.0	2942.79	6.9258	...	...	1.0
st4	372.756	100.0	2513.54	6.9258	0.9285	...	0.0972
st5	318.956	10.0	2193.78	6.9258	0.8369	...	0.9028
st6	318.956	10.0	191.806	0.6492	0.0	...	0.9028
st7	318.959	100.0	191.897	0.6492	...	...	0.9028

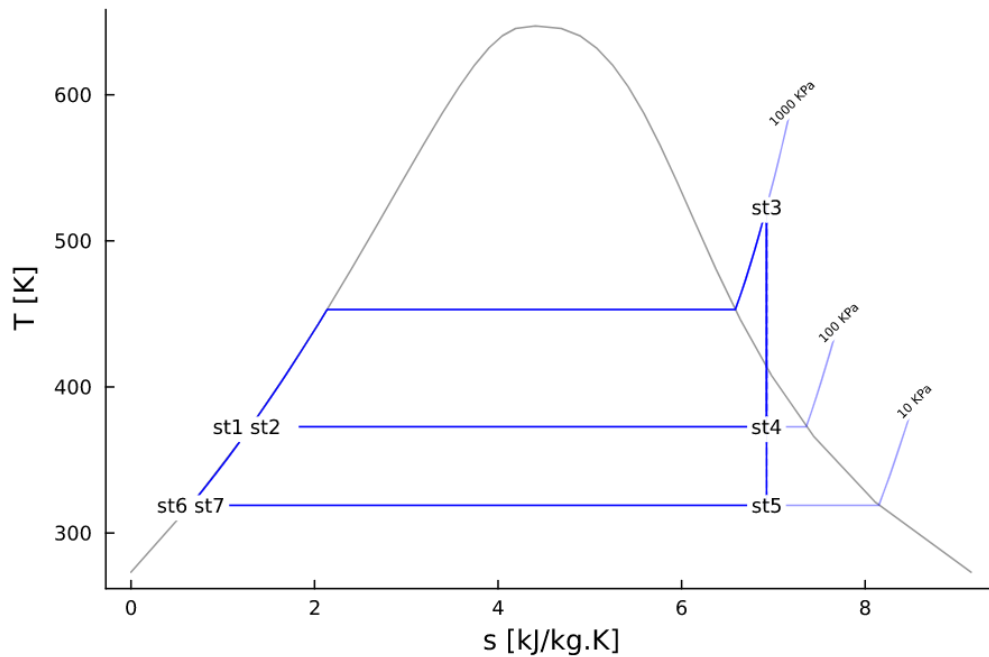
Fonte: Autoria própria (2023).

Devido as bifurcações geradas, é possível observar a fração de fluxo mássico que vai para cada ramificação do ciclo, através da tabela que apresenta as propriedades de cada estado do ciclo.

Tabela 13 – Propriedades do ciclo do exemplo 6

	<i>Total</i>	<i>Component</i>	<i>Value</i>
$q_{in}$	2524.3482 kJ/kg	boiler: st2 » st3	2524.3482 kJ/kg
$\dot{Q}_{in}$	...		...
$q_{out}$	2001.9783 kJ/kg	condenser: st5 » st6	2001.9783 kJ/kg
$\dot{Q}_{out}$	...		...
$w_{in}$	1.0296 kJ/kg	pump: st6 » st7	0.0909 kJ/kg
$\dot{W}_{in}$	...	pump: st1 » st2	0.9387 kJ/kg
			...
$w_{out}$	717.9338 kJ/kg	turbine: st3 » [st4, st5]	717.9338 kJ/kg
$\dot{W}_{out}$	...		...
		Thermal efficiency ( $\eta$ ) =	28.3995 %

Fonte: Autoria própria (2023).

Gráfico 5 – Diagrama  $T - s$  do exemplo 6

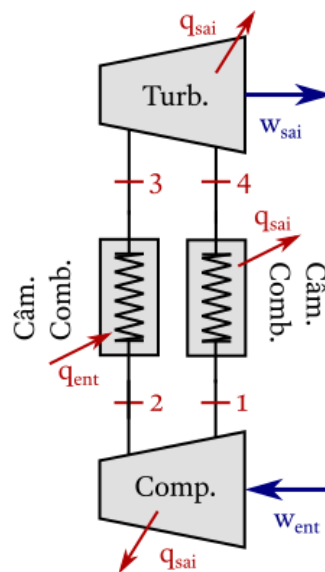
Fonte: Autoria própria (2023).

#### 4.7 Exemplo 7 - Ciclo Brayton

O exemplo representado na Figura 14 aplicará o ciclo Brayton no programa, mostrando a sintaxe para atingir esse objetivo:

- Ar é utilizado como fluido de trabalho;
- Pressão na entrada do compressor 100 kPa;
- Temperatura na entrada do compressor 20 °C;
- Razão de pressão igual a 7;
- Temperatura na entrada da turbina 500 °C;
- A turbina e o compressor possuem eficiência isentrópica de 90%.

Figura 14 – Representação do ciclo do Brayton, em que se tem um compressor, uma turbina e duas câmaras de combustão.



Fonte: Autoria própria (2023).

## Listagem 7 – Código do exemplo 7

```

1 using CycleSolver
2
3 @solve begin
4     st2.p / st1.p = 7
5     st1.T = 20 + 273
6     st1.p = 100
7     st3.T = 500 + 273
8     newCycle[ Air ]
9         compressor(st1 , st2 , 90)
10        combustion_chamber(st2 , st3)
11        turbine(st3 , st4 , 90)
12        combustion_chamber(st4 , st1)
13 end
14
15 PrintResults ()

```

Fonte: Autoria própria (2023).

Nesse exemplo, além de mostrar a mudança do fluido de trabalho para o ar, também é exibido a operação de divisão entre duas propriedades, demonstrando que expressões envolvendo as propriedades podem ser desenvolvidas.

Tabela 14 – Propriedades de cada estado do exemplo 7

<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
st1	293.0	100.0	419.257	3.8667	...	...	1.0
st2	532.122	700.0	662.359	3.9135	...	...	1.0
st3	773.0	700.0	918.995	4.3107	...	...	1.0
st4	487.236	100.0	616.266	4.3821	...	...	1.0

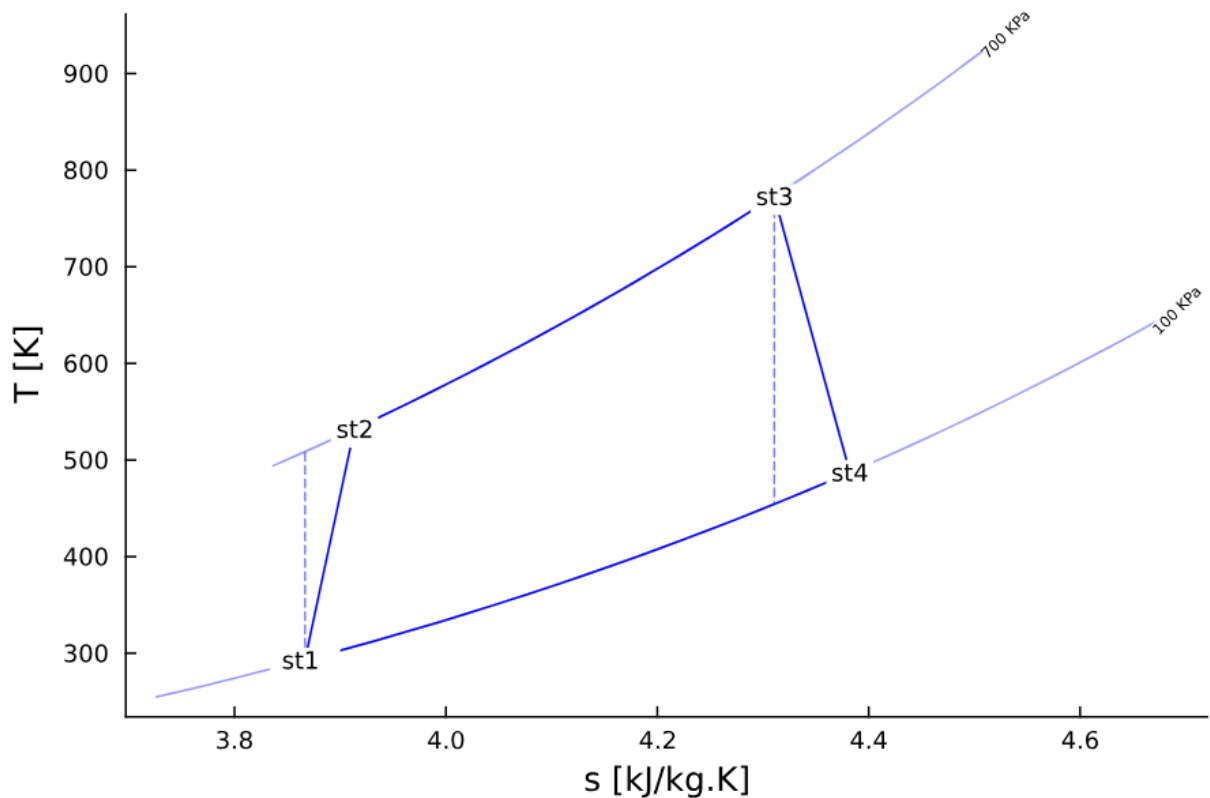
Fonte: Autoria própria (2023).



Tabela 15 – Propriedades do ciclo do exemplo 7

	<i>Total</i>	<i>Component</i>	<i>Value</i>
$q_{in}$	256.6365 kJ/kg	combustion_chamber: st2 » st3	256.6365 kJ/kg
$\dot{Q}_{in}$	...		...
$q_{out}$	197.0086 kJ/kg	combustion_chamber: st4 » st1	197.0086 kJ/kg
$\dot{Q}_{out}$	...		...
$w_{in}$	243.1014 kJ/kg	compressor: st1 » st2	243.1014 kJ/kg
$\dot{W}_{in}$	...		...
$w_{out}$	302.7293 kJ/kg	turbine: st3 » st4	302.7293 kJ/kg
$\dot{W}_{out}$	...		...
		Thermal efficiency ( $\eta$ ) =	23.2344 %

Fonte: Autoria própria (2023).

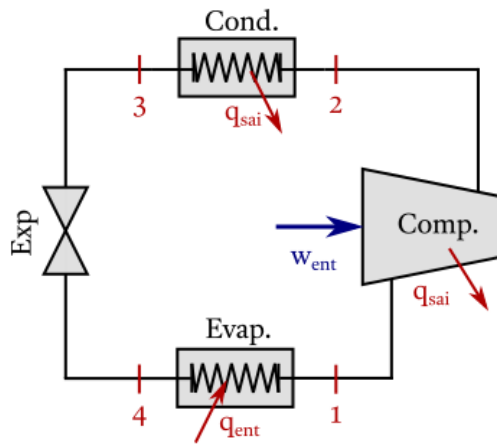
Gráfico 6 – Diagrama  $T - s$  do exemplo 7

Fonte: Autoria própria (2023).

#### 4.8 Exemplo 8 - Ciclo de refrigeração

Nesse exemplo será demonstrado a sintaxe para representar o ciclo da Figura 15.

Figura 15 – Representação do ciclo de refrigeração, em que se tem um compressor, um evaporador, uma válvula de expansão e um condensador.



Fonte: Autoria própria (2023).

- Refrigerante-134a é utilizado como fluido de trabalho;
- Pressão na entrada do compressor 100 kPa;
- Pressão na saída do compressor 1500 kPa;
- O compressor possui eficiência isentrópica de 80%.

#### Listagem 8 – Código do exemplo 8

```

1 using CycleSolver
2
3 @solve begin
4     st1.p = 100
5     st2.p = 1500
6     newRefrigerationCycle [R134a]
7         compressor(st1 , st2 , 80)
8         condenser(st2 , st3)
9         expansion_valve(st3 , st4)
10        evaporator(st4 , st1)
11 end
12
13 PrintResults(3)

```

Fonte: Autoria própria (2023).

A inicialização do ciclo de refrigeração é feita fazendo a mudança de `newCycle[]` para `newRefrigerationCycle[]`, de modo que seja calculado o COP no lugar da eficiência térmica do ciclo.

Tabela 16 – Propriedades de cada estado do exemplo 8

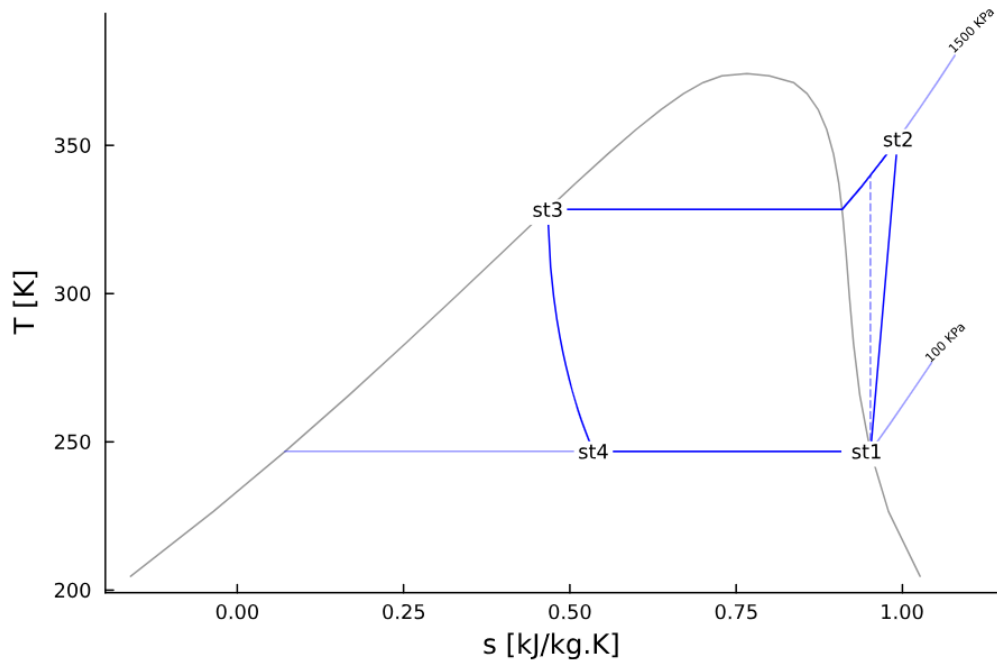
<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
st1	246.789	100.0	234.455	0.9519	1.0	...	1.0
st2	352.08	1500.0	305.522	0.993	...	...	1.0
st3	328.383	1500.0	131.695	0.4665	0.0	...	1.0
st4	246.789	100.0	131.695	0.5355	0.5268	...	1.0

Fonte: Autoria própria (2023).

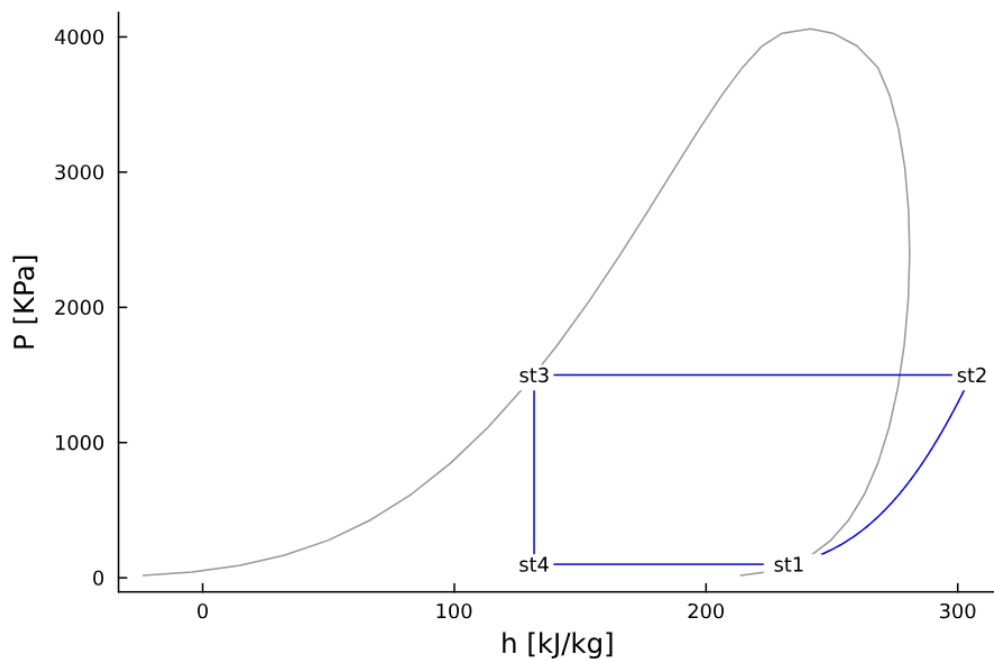
Tabela 17 – Propriedades do ciclo do exemplo 8

	<i>Total</i>	<i>Component</i>	<i>Value</i>
$q_{in}$	102.7603 kJ/kg	evaporator: st4 » st1	102.7603 kJ/kg
$\dot{Q}_{in}$	...		...
$q_{out}$	173.8267 kJ/kg	condenser: st2 » st3	173.8267 kJ/kg
$\dot{Q}_{out}$	...		...
$w_{in}$	71.0665 kJ/kg	compressor: st1 » st2	71.0665 kJ/kg
$\dot{W}_{in}$	...		...
$w_{out}$	...		...
$\dot{W}_{out}$	...		...
Coefficient of performance ( <i>COP</i> ) =			1.446

Fonte: Autoria própria (2023).

Gráfico 7 – Diagrama  $T - s$  do exemplo 8

Fonte: Autoria própria (2023).

Gráfico 8 – Diagrama  $P - h$  do exemplo 8

Fonte: Autoria própria (2023).

Devido à modificação na chamada da função `PrintResults(3)`, a qual possui o valor 3 como seu primeiro parâmetro, foi possível gerar não apenas o Gráfico 7, que correlaciona temperatura e entropia, mas também foi gerado o Gráfico 8, que relaciona pressão e entalpia.

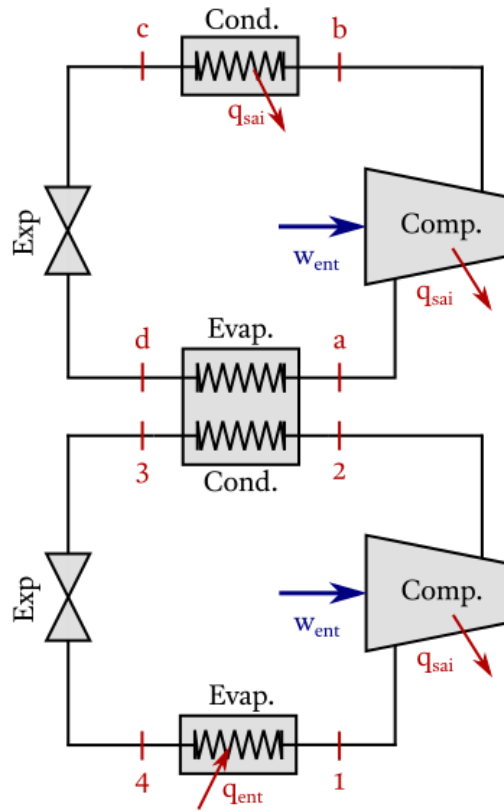
O valor padrão do primeiro parâmetro da função é o número 1, que gera apenas o gráfico  $T - s$ . Com o número 2, será gerado apenas o gráfico  $P - h$ , e com o número 3, serão gerados ambos os gráficos.

#### 4.9 Exemplo 9 - Sistema de refrigeração em cascata

O exemplo representado na Figura 16 mostra os funcionamento de múltiplos ciclos combinados, formando um sistema:

- Refrigerante-134a é utilizado como fluido de trabalho em ambos os ciclos;
- O ciclo inferior possui fluxo mássico de 2,5 kg/s;
- A pressão na entrada do compressor inferior é de 250 kPa, e de 600 kPa em sua saída;
- A pressão na entrada do compressor superior é de 500 kPa, e de 1500 kPa em sua saída;
- O compressor superior possui eficiência isentrópica de 80%, e o inferior de 90%.

Figura 16 – Representação do sistema de refrigeração em cascata, em que se tem um ciclo com um condensador, uma válvula de expansão e um compressor, outro ciclo com um evaporador, uma válvula de expansão e um compressor, e um trocador de calor que liga ambos os ciclos.



Fonte: Autoria própria (2023).

## Listagem 9 – Código do exemplo 9

```

1 using CycleSolver
2
3 @solve begin
4
5     evaporator_condenser([stD, st2], [stA, st3])
6
7     stA.p = 500
8     stB.p = 1500
9     newRefrigerationCycle[R134a]
10         compressor(stA, stB, 80)
11         condenser(stB, stC)
12         expansion_valve(stC, stD)
13
14     st1.p = 250
15     st2.p = 600
16     newRefrigerationCycle[R134a: 2.5]
17         compressor(st1, st2, 90)
18         expansion_valve(st3, st4)
19         evaporator(st4, st1)
20
21 end
22
23 PrintResults()

```

Fonte: Autoria própria (2023).

Foram inicializados dois ciclos para a criação do sistema, cada um deles com seus componentes localizados logo depois de cada inicialização. A declaração das propriedades conhecidas segue não sendo influenciada pela sua localização, compartilhando a mesma característica que o `evaporator_condenser()`, que é um componente que interage com ambos os ciclos, desse modo, não é pertencente a nenhum ciclo exclusivamente.

Neste componente também há múltiplos estados de entrada e de saída em que não ocorrem a mistura entre eles. Dessa forma, para definir corretamente as correntes é importante seguir a ordem da lista de estados, fazendo com que os estados com mesmo ciclo fiquem na mesma posição na lista de entrada e saída.

Devido a existência de múltiplos ciclos, será apresentado os gráficos e tabelas de ambos os ciclos individualmente e concluído com o gráfico e tabela do sistema, considerando a interação entre os ciclos.

## 4.9.1 Ciclo superior

Tabela 18 – Propriedades de cada estado do 1º ciclo do exemplo 9

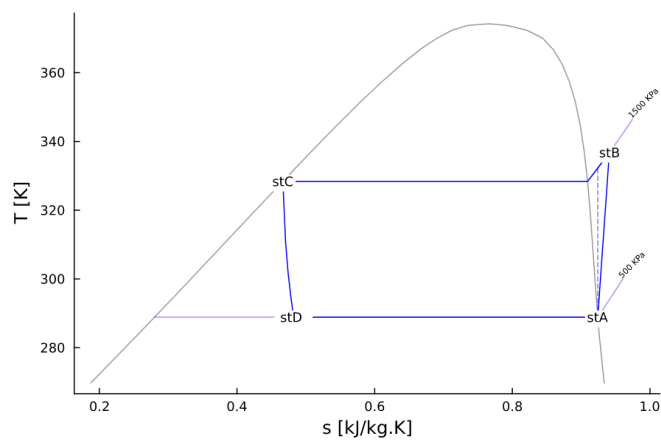
State name	$T$ [K]	$P$ [kPa]	$h$ [kJ/kg]	$s$ [kJ/kg·K]	$x$	$\dot{m}$ [kg/s]	Mass – flux fraction
stD	288.885	500.0	131.695	0.4823	0.3137	3.653	1.0
stA	288.885	500.0	259.327	0.9241	1.0	3.653	1.0
stB	336.769	1500.0	287.628	0.941	...	3.653	1.0
stC	328.383	1500.0	131.695	0.4665	0.0	3.653	1.0

Fonte: Autoria própria (2023).

Tabela 19 – Propriedades do 1º ciclo do exemplo 9

	Total	Component	Value
$q_{in}$	127.6324 kJ/kg	evaporator_condenser: [stD, st2] » [stA, st3]	127.6324 kJ/kg
$\dot{Q}_{in}$	466.2367 kW		466.2367 kW
$q_{out}$	155.9332 kJ/kg	condenser: stB » stC	155.9332 kJ/kg
$\dot{Q}_{out}$	569.6185 kW		569.6185 kW
$w_{in}$	28.3008 kJ/kg	compressor: stA » stB	28.3008 kJ/kg
$\dot{W}_{in}$	103.3818 kW		103.3818 kW
$w_{out}$	...		
$\dot{W}_{out}$	...		
Coefficient of performance ( $COP$ ) =			4.5099

Fonte: Autoria própria (2023).

Gráfico 9 – Diagrama  $T - s$  do 1º ciclo do exemplo 9

Fonte: Autoria própria (2023).



## 4.9.2 Ciclo inferior

Tabela 20 – Propriedades de cada estado do 2º ciclo do exemplo 9

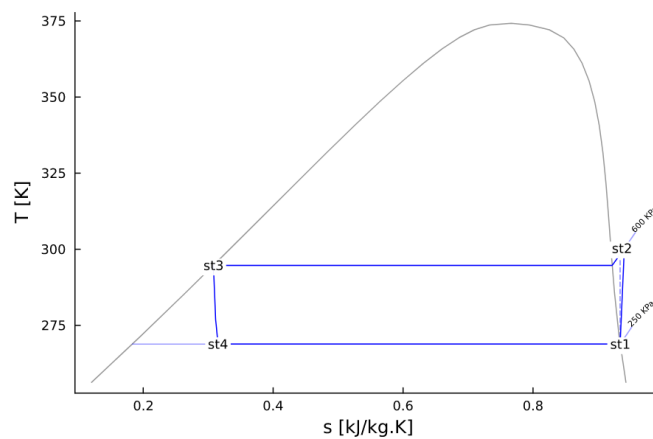
State name	$T$ [K]	$P$ [kPa]	$h$ [kJ/kg]	$s$ [kJ/kg·K]	$x$	$\dot{m}$ [kg/s]	Mass – flux fraction
st2	300.331	600.0	268.033	0.9407	...	2.5	1.0
st3	294.722	600.0	81.5384	0.3081	0.0	2.5	1.0
st1	268.866	250.0	247.94	0.934	1.0	2.5	1.0
st4	268.866	250.0	81.5384	0.3151	0.1755	2.5	1.0

Fonte: Autoria própria (2023).

Tabela 21 – Propriedades do 2º ciclo do exemplo 9

	Total	Component	Value
$q_{in}$	166.4013 kJ/kg	evaporator: st4 » st1	166.4013 kJ/kg
$\dot{Q}_{in}$	416.0033 kW		416.0033 kW
$q_{out}$	186.4947 kJ/kg	evaporator_condenser: [stD, st2] » [stA, st3]	186.4947 kJ/kg
$\dot{Q}_{out}$	466.2367 kW		466.2367 kW
$w_{in}$	20.0934 kJ/kg	compressor: st1 » st2	20.0934 kJ/kg
$\dot{W}_{in}$	50.2334 kW		50.2334 kW
$w_{out}$	...		
$\dot{W}_{out}$	...		
Coefficient of performance ( $COP$ ) =			8.2814

Fonte: Autoria própria (2023).

Gráfico 10 – Diagrama  $T - s$  do 2º ciclo do exemplo 9

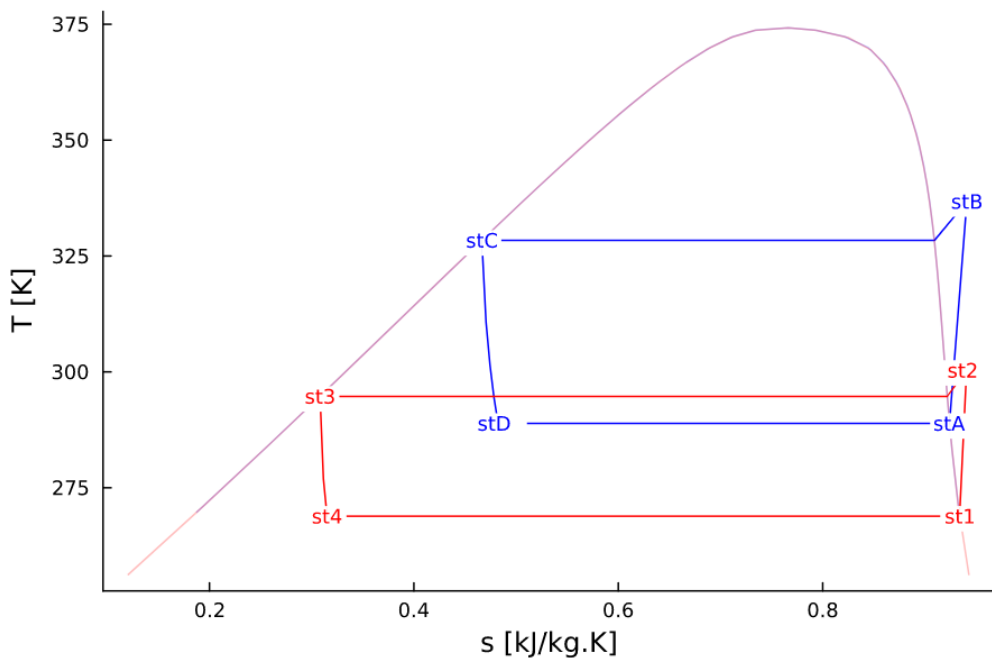
Fonte: Autoria própria (2023).

## 4.9.3 Propriedades do sistema

Tabela 22 – Propriedades do sistema do exemplo 9

	<i>Total</i>	<i>Component</i>	<i>Value</i>
$q_{in}$	166.4013 kJ/kg	evaporator: st4 » st1	166.4013 kJ/kg
$\dot{Q}_{in}$	416.0033 kW		416.0033 kW
$q_{out}$	155.9332 kJ/kg	condenser: stB » stC	155.9332 kJ/kg
$\dot{Q}_{out}$	569.6185 kW		569.6185 kW
$w_{in}$	48.3941 kJ/kg	compressor: stA » stB	28.3008 kJ/kg
$\dot{W}_{in}$	153.6152 kW	compressor: st1 » st2	103.3818 kW
			20.0934 kJ/kg
			50.2334 kW
$w_{out}$	...		
$\dot{W}_{out}$	...		
Coefficient of performance ( <i>COP</i> ) =			2.7081

Fonte: Autoria própria (2023).

Gráfico 11 – Diagrama  $T - s$  do sistema do exemplo 9

Fonte: Autoria própria (2023).

Primeiramente é possível verificar que o algoritmo calculou o fluxo mássico do ciclo superior com base nos dados de entrada, respeitando todas as relações impostas e apresentando o resultado nas tabelas das propriedades dos estados do ciclo.

A interação entre os ciclos pode ser vista ao analisar o comportamento do `evaporator_condenser()` nas tabelas de propriedades do ciclo. Na Tabela 19 o componente está apresentado como responsável pelo calor que entra no determinado ciclo, já na Tabela 21 o componente está apresentado como responsável pelo calor que sai, e na Tabela 22 o componente não está presente, pois o componente não realiza transferência de calor fora do sistema, apenas entre os ciclos.

Além disso, é importante apontar a geração de gráficos, que por se tratar de um sistema de ciclos gera os gráficos individuais, e um gráfico fazendo a sobreposição de ambos, com cores diferentes para melhor visualização.

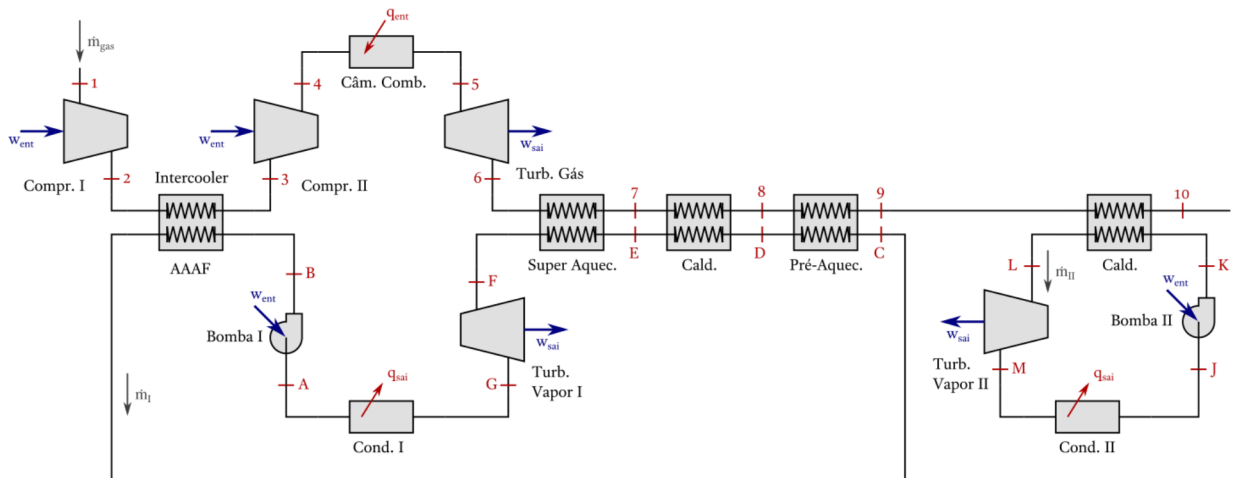
#### 4.10 Exemplo 10 - Interação de 3 ciclos

O exemplo a seguir mostra o funcionamento da combinação de 3 ciclos distintos, conforme a Figura 17.

- Dióxido de carbono é utilizado como fluido de trabalho no ciclo superior;
- Água é utilizada como fluido de trabalho no ciclo inferior esquerdo;
- Refrigerante-134a é utilizado como fluido de trabalho no ciclo inferior direito;
- O ciclo superior possui fluxo mássico de 2 kg/s;
- A pressão na entrada do Compr. I é de 100 kPa, e sua temperatura de 10°C;
- A temperatura na entrada da Turb. Gás é de 1000°C;
- A razão de pressão do ciclo a gás é de 6 a qual é igualmente balanceada em todos os estágios de compressores;
- A Turb. Gás possui eficiência isentrópica de 90%;
- Os compressor do ciclo superior possui eficiência isentrópica de 85%;
- A pressão na entrada da Turb. Vapor I é de 2000 kPa;
- A temperatura na entrada da Bomba I é de 15°C;
- A pressão na entrada da Turb. Vapor II é de 1400 kPa, e sua temperatura de 100°C;
- A temperatura na entrada da Bomba II é de 15°C;

- Os regeneradores Intercooler/AAAF, Caldeira I, Superaquecedor I e Caldeira II possuem todos efetividade de 80%, avaliada com base na corrente de menor taxa de capacidade térmica.

**Figura 17 – Representação do sistema com 3 ciclos, em que se tem um ciclo de Dióxido de carbono com dois compressores, uma câmara de combustão, e uma turbina, um ciclo de água com uma bomba, um condensador, e uma turbina, um ciclo de Refrigerante-134a com uma turbina, uma bomba e um condensador, e quatro trocadores de calor fazendo a ligação entre os ciclos de Dióxido de carbono e água, e um trocador de calor fazendo a ligação entre os ciclos de Refrigerante-134a e Dióxido de carbono.**



Fonte: Naaktgeboren (2023).

## Listagem 10 – Código do exemplo 10

```

1 using CycleSolver
2
3 @solve begin
4     st1.p = 100
5     st1.T = 10 + 273.15
6     st2.p/st1.p = 6^(1/2)
7     st4.p/st3.p = 6^(1/2)
8     st5.T = 1000 + 273.15
9     st10.p = st1.p
10    newCycle[CO2: 2]
11        compressor(st1 , st2 , 85)
12        compressor(st3 , st4 , 85)
13        combustion_chamber(st4 , st5)
14        turbine(st5 , st6 , 90)
15
16    std.Q = 0
17    ste.Q = 1
18    stf.p = 2000
19    sta.T = 15 + 273.15
20    newCycle[Water]
21        pump(sta , stb)
22        turbine(stf , stg)
23        condenser(stg , sta)
24
25    stl.p = 1400
26    stl.T = 100 + 273.15
27    stj.T = 15 + 273.15
28    newCycle[R134a]
29        pump(stj , stk)
30        turbine(stl , stm)
31        condenser(stm , stj)
32
33    heater_exchanger([st2 , stb] , [st3 , stc] , 80)
34    heater_exchanger([st6 , ste] , [st7 , stf] , 80)
35    heater_exchanger([std , st7] , [ste , st8] , 80)
36    heater_exchanger([stc , st8] , [std , st9] , find)
37    heater_exchanger([stk , st9] , [stl , st10] , 80)
38 end
39
40 PrintResults()

```

Fonte: Autoria própria (2023).

O trocador de calor Pré-Aquec. possui efetividade desconhecida. Com isso, na sua representação na Listagem 11, o seu terceiro parâmetro encontra-se definido como `find`, para que o código encontre essa propriedade desconhecida e apresente o resultado.

4.10.1 Ciclo de CO<sub>2</sub>**Tabela 23 – Propriedades de cada estado do ciclo de CO<sub>2</sub> do exemplo 10**

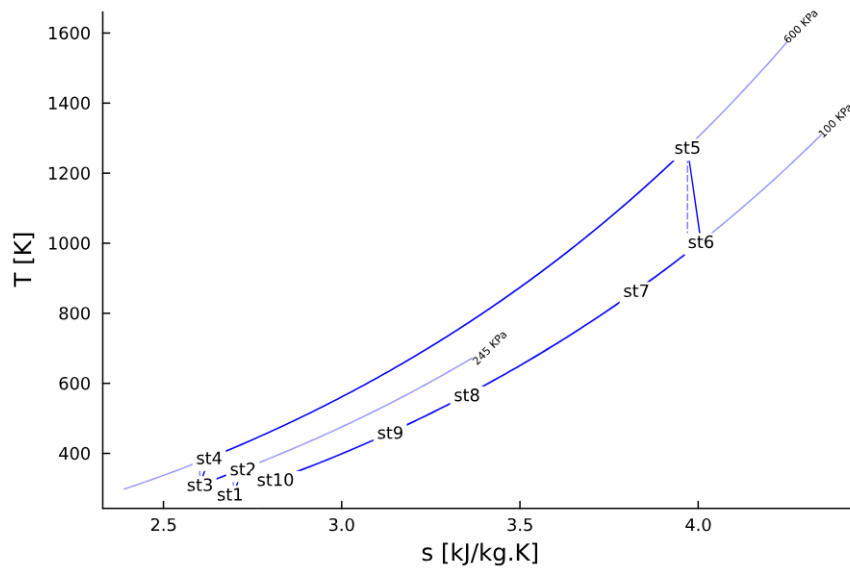
<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
st1	283.15	100.0	493.2	2.6958	...	2.0	1.0
st2	355.439	244.949	555.207	2.7223	...	2.0	1.0
st3	310.464	244.949	515.163	2.6019	...	2.0	1.0
st4	387.201	600.0	582.591	2.6284	...	2.0	1.0
st5	1273.15	600.0	1611.59	3.9696	...	2.0	1.0
st6	1004.09	100.0	1270.81	4.0079	...	2.0	1.0
st7	864.438	100.0	1101.14	3.8261	...	2.0	1.0
st8	567.007	100.0	764.726	3.3511	...	2.0	1.0
st9	459.425	100.0	654.611	3.136	...	2.0	1.0
st10	325.109	100.0	529.135	2.8141	...	2.0	1.0

Fonte: Autoria própria (2023).

**Tabela 24 – Propriedades do ciclo de CO<sub>2</sub> do exemplo 10**

	<i>Total</i>	<i>Component</i>	<i>Value</i>
<i>q<sub>in</sub></i>	1029.0002 kJ/kg	combustion_chamber: st4 » st5	1029.0002 kJ/kg
<i>Q<sub>in</sub></i>	2058.0004 kW		2058.0004 kW
		heater_exchanger: [st2, stb] » [st3, stc]	40.044 kJ/kg 80.088 kW
		heater_exchanger: [stk, st9] » [stl, st10]	125.476 kJ/kg 250.9519 kW
<i>q<sub>out</sub></i>	781.7197 kJ/kg	heater_exchanger: [std, st7] » [ste, st8]	336.4122 kJ/kg
<i>Q<sub>out</sub></i>	1563.4394 kW	heater_exchanger: [st6, ste] » [st7, stf]	672.8244 kW 169.6726 kJ/kg 339.3452 kW
		heater_exchanger: [stc, st8] » [std, st9]	110.1149 kJ/kg 220.2298 kW
<i>w<sub>in</sub></i>	129.434 kJ/kg	compressor: st1 » st2	62.0066 kJ/kg 124.0132 kW
<i>W<sub>in</sub></i>	258.8681 kW	compressor: st3 » st4	67.4274 kJ/kg 134.8549 kW
<i>w<sub>out</sub></i>	340.7802 kJ/kg	turbine: st5 » st6	340.7802 kJ/kg
<i>W<sub>out</sub></i>	681.5603 kW		681.5603 kW
		Thermal efficiency ( <i>n</i> ) =	20.539 %

Fonte: Autoria própria (2023).

Gráfico 12 – Diagrama  $T - s$  do ciclo de CO2 do exemplo 10

Fonte: Autoria própria (2023).

#### 4.10.2 Ciclo de água

Tabela 25 – Propriedades de cada estado do ciclo de água do exemplo 10

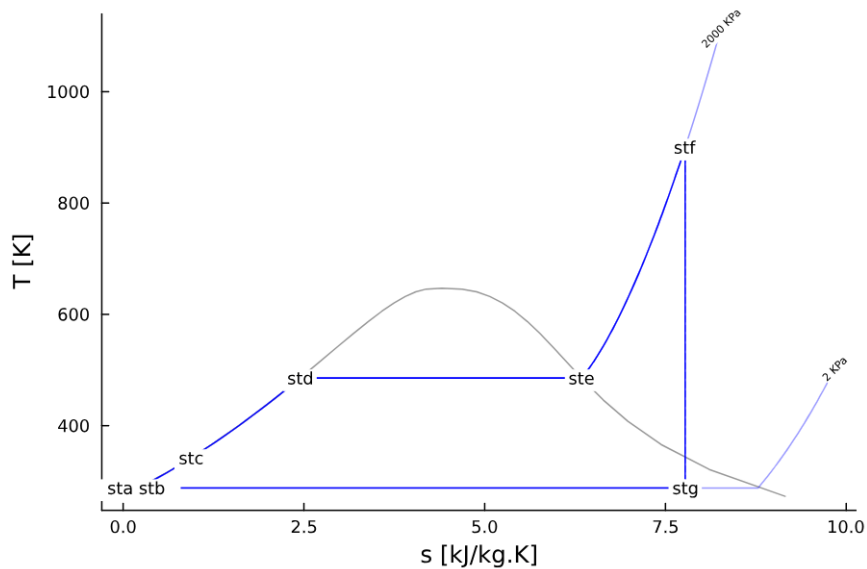
<i>State name</i>	$T$ [K]	$P$ [kPa]	$h$ [kJ/kg]	$s$ [kJ/kg.K]	$x$	$\dot{m}$ [kg/s]	<i>Mass – flux fraction</i>
sta	288.15	1.7058	62.9815	0.2245	0.0	0.356	1.0
stb	288.171	2000.0	64.9807	0.2245	...	0.356	1.0
stc	342.016	2000.0	289.928	0.9401	...	0.356	1.0
std	485.527	2000.0	908.498	2.4468	0.0	0.356	1.0
stf	900.074	2000.0	3751.43	7.7728	...	0.356	1.0
stg	288.15	1.7058	2238.02	7.7728	0.8822	0.356	1.0
ste	485.527	2000.0	2798.29	6.339	1.0	0.356	1.0

Fonte: Autoria própria (2023).

Tabela 26 – Propriedades do ciclo de água do exemplo 10

	<i>Total</i>	<i>Component</i>	<i>Value</i>
$q_{in}$ $\dot{Q}_{in}$	3686.447 kJ/kg 1312.4875 kW	heater_exchanger: [st2, stb] » [st3, stc]	224.9472 kJ/kg 80.088 kW
		heater_exchanger: [std, st7] » [ste, st8]	1889.7945 kJ/kg 672.8244 kW
		heater_exchanger: [st6, ste] » [st7, stf]	953.1351 kJ/kg 339.3452 kW
		heater_exchanger: [stc, st8] » [std, st9]	618.5702 kJ/kg 220.2298 kW
$q_{out}$ $\dot{Q}_{out}$	2175.0406 kJ/kg 774.3807 kW	condenser: stg » sta	2175.0406 kJ/kg 774.3807 kW
$w_{in}$ $\dot{W}_{in}$	1.9993 kJ/kg 0.7118 kW	pump: sta » stb	1.9993 kJ/kg 0.7118 kW
$w_{out}$ $\dot{W}_{out}$	1513.4057 kJ/kg 538.8185 kW	turbine: stf » stg	1513.4057 kJ/kg 538.8185 kW
Thermal efficiency ( $\eta$ ) =			40.999 %

Fonte: Autoria própria (2023).

Gráfico 13 – Diagrama  $T - s$  do ciclo de água do exemplo 10

Fonte: Autoria própria (2023).



## 4.10.3 Ciclo de R134a

Tabela 27 – Propriedades de cada estado do ciclo de R134a do exemplo 10

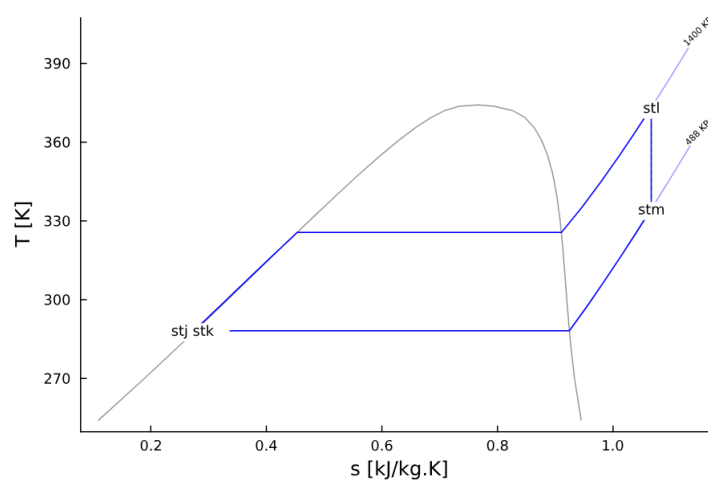
<i>State name</i>	<i>T</i> [K]	<i>P</i> [kPa]	<i>h</i> [kJ/kg]	<i>s</i> [kJ/kg·K]	<i>x</i>	<i>m</i> [kg/s]	<i>Mass – flux fraction</i>
stj	288.15	488.374	72.3364	0.2768	0.0	0.9756	1.0
stk	288.592	1400.0	73.0687	0.2768	...	0.9756	1.0
stl	373.15	1400.0	330.302	1.0661	...	0.9756	1.0
stm	334.496	488.374	302.957	1.0661	...	0.9756	1.0

Fonte: Autoria própria (2023).

Tabela 28 – Propriedades do ciclo de R134a do exemplo 10

	<i>Total</i>	<i>Component</i>	<i>Value</i>
<i>q<sub>in</sub></i>	257.2336 kJ/kg	heater_exchanger: [stk, st9] » [stl, st10]	257.2336 kJ/kg
<i>Q<sub>in</sub></i>	250.9519 kW		250.9519 kW
<i>q<sub>out</sub></i>	230.6203 kJ/kg	condenser: stm » stj	230.6203 kJ/kg
<i>Q<sub>out</sub></i>	224.9885 kW		224.9885 kW
<i>w<sub>in</sub></i>	0.7323 kJ/kg	pump: stj » stk	0.7323 kJ/kg
<i>W<sub>in</sub></i>	0.7144 kW		0.7144 kW
<i>w<sub>out</sub></i>	27.3456 kJ/kg	turbine: stl » stm	27.3456 kJ/kg
<i>W<sub>out</sub></i>	26.6778 kW		26.6778 kW
		Thermal efficiency ( <i>η</i> ) =	10.346 %

Fonte: Autoria própria (2023).

Gráfico 14 – Diagrama *T – s* do ciclo de R134a do exemplo 10

Fonte: Autoria própria (2023).

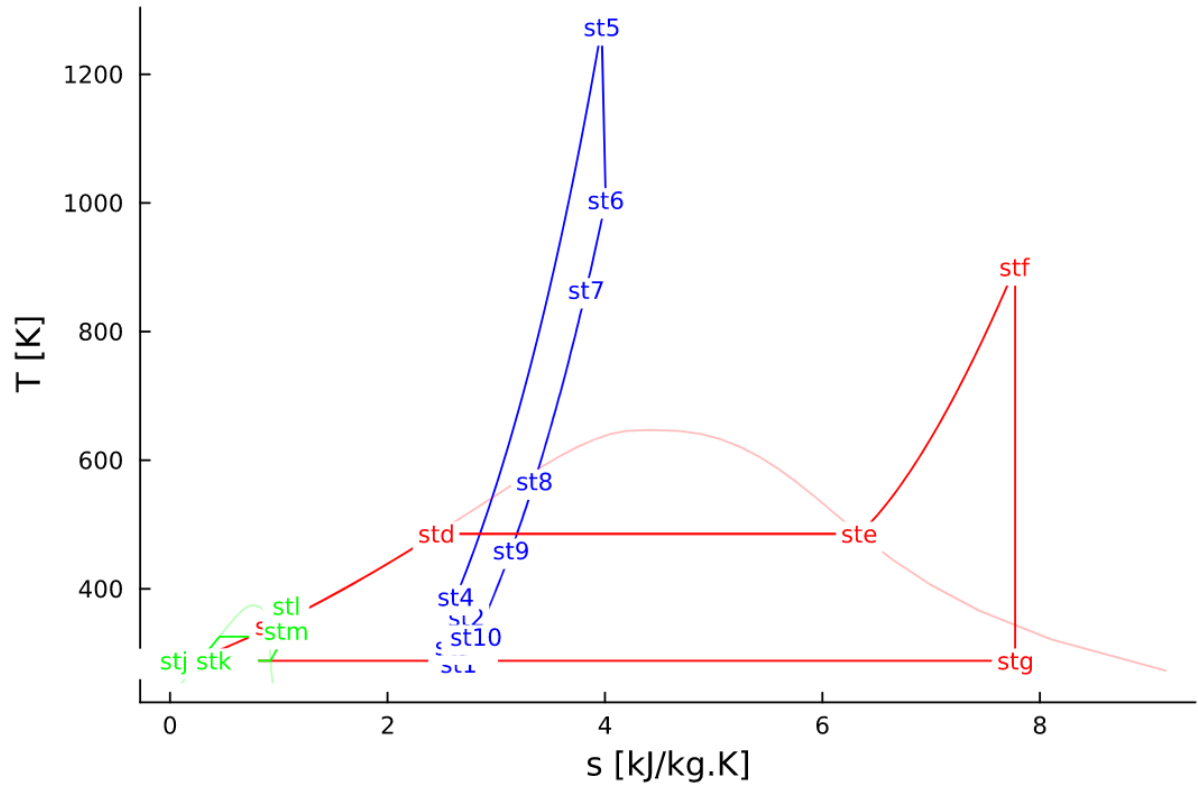
## 4.10.4 Propriedades do sistema

Tabela 29 – Propriedades do sistema do exemplo 10

	<i>Total</i>	<i>Component</i>	<i>Value</i>
$q_{in}$	1029.0002 kJ/kg	combustion_chamber: st4 » st5	1029.0002 kJ/kg
$\dot{Q}_{in}$	2058.0004 kW		2058.0004 kW
		condenser: stm » stj	230.6203 kJ/kg
$q_{out}$	2405.6609 kJ/kg		224.9885 kW
$\dot{Q}_{out}$	999.3692 kW	condenser: stg » sta	2175.0406 kJ/kg
			774.3807 kW
		compressor: st1 » st2	62.0066 kJ/kg
			124.0132 kW
		compressor: st3 » st4	67.4274 kJ/kg
$w_{in}$	132.1656 kJ/kg		134.8549 kW
$\dot{W}_{in}$	260.2943 kW	pump: stj » stk	0.7323 kJ/kg
			0.7144 kW
		pump: sta » stb	1.9993 kJ/kg
			0.7118 kW
		turbine: stf » stg	1513.4057 kJ/kg
			538.8185 kW
$w_{out}$	1881.5315 kJ/kg	turbine: stl » stm	27.3456 kJ/kg
$\dot{W}_{out}$	1247.0567 kW		26.6778 kW
		turbine: st5 » st6	340.7802 kJ/kg
			681.5603 kW
		Thermal efficiency ( $\eta$ ) =	47.9476 %
		Calculated Component Properties	
		$\epsilon$ ) heater_exchanger: [stc, st8] » [std, st9]	49.9063 %

**Fonte: Autoria própria (2023).**

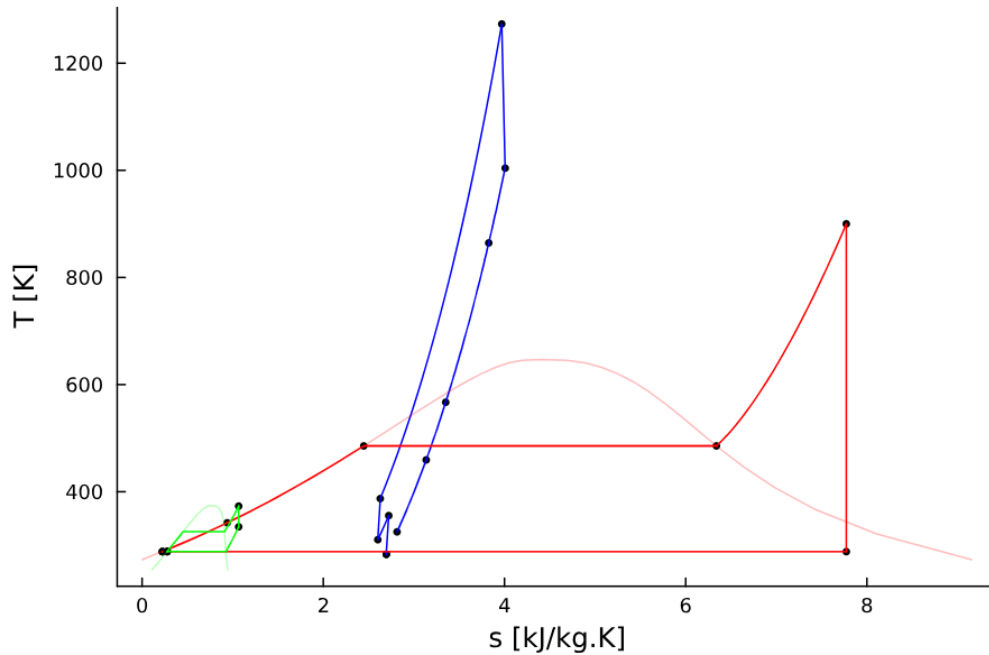
Na Tabela 29 é apresentada a efetividade calculada do trocador de calor Pré-Aquec., que era desconhecida anteriormente, e foi obtida ao definir o terceiro parâmetro da função do componente como `find`.

Gráfico 15 – Diagrama  $T - s$  do sistema do exemplo 10

Fonte: Autoria própria (2023).

Para uma melhor visualização do gráfico do sistema, é possível adicionar alguns atributos na sua geração, modificações na função `PrintResults()`, que é composta por mais dois atributos opcionais, além da seleção dos gráficos: `PrintResults(graphs=1, showStateNames=true, multiplyEntropyByMass=false)`. Ao chamar a função como `PrintResults(1, false)`, o gráfico será exibido sem os nomes dos estados, conforme demonstrado no Gráfico 16.

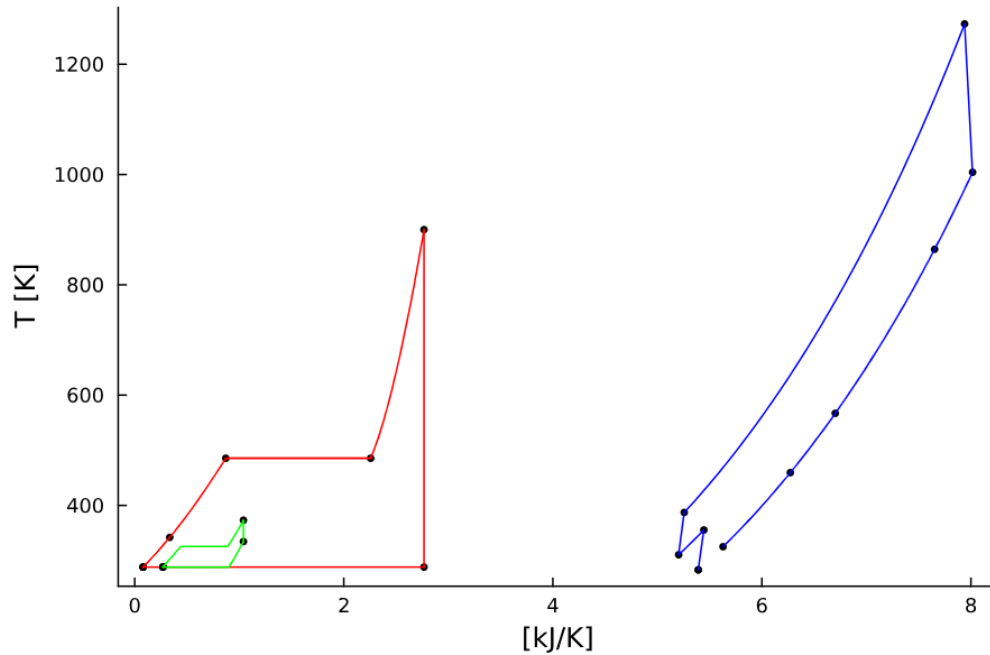
**Gráfico 16 – Diagrama  $T - s$  do sistema do exemplo 10, sem os nomes dos estados**



Fonte: Autoria própria (2023).

Além dessa modificação, é possível exibir o gráfico com as entropias do ciclo sendo multiplicadas pelo seu fluxo mássico, chamando a função como `PrintResults(1, false, true)`, conforme demonstrado no Gráfico 17.

**Gráfico 17 – Diagrama  $T - s$  do sistema do exemplo 10, multiplicando a entropia pelo fluxo mássico**



**Fonte: Autoria própria (2023).**

#### 4.11 Exemplo 11 - Imbalanço de massa e de energia, e geração de entropia

O exemplo a seguir mostra como se obtêm o imbalanço de massa, de energia, e a geração de entropia do sistema, seguindo as mesmas propriedades do exemplo da seção 4.10.

##### Listagem 11 – Código do exemplo 11

```

1  using CycleSolver
2
3  @solve begin
4      st1.p = 100
5      st1.T = 10 + 273.15
6      st2.p/st1.p = 6^(1/2)
7      st4.p/st3.p = 6^(1/2)
8      st5.T = 1000 + 273.15
9      st10.p = st1.p
10     newCycle[CO2: 2]
11         compressor(st1 , st2 , 85)
12         compressor(st3 , st4 , 85)
13         combustion_chamber(st4 , st5)
14         turbine(st5 , st6 , 90)
15
16     std.Q = 0
17     ste.Q = 1
18     stf.p = 2000
19     sta.T = 15 + 273.15
20     newCycle[Water]
21         pump(sta , stb)
22         turbine(stf , stg)
23         condenser(stg , sta , 14 + 273.15)
24
25     stl.p = 1400
26     stl.T = 100 + 273.15
27     stj.T = 15 + 273.15
28     newCycle[R134a]
29         pump(stj , stk)
30         turbine(stl , stm)
31         condenser(stm , stj , 14 + 273.15)
32
33     heater_exchanger([st2 , stb] , [st3 , stc] , 80)
34     heater_exchanger([st6 , ste] , [st7 , stf] , 80)
35     heater_exchanger([std , st7] , [ste , st8] , 80)
36     heater_exchanger([stc , st8] , [std , st9] , find)
37     heater_exchanger([stk , st9] , [stl , st10] , 80)
38 end
39
40 PrintImbalance()

```

Fonte: Autoria própria (2023).

A visualização dos resultados de balanço e de geração de entropia é feita por meio da função `PrintImbalance()`, que apresenta os valores para cada componente individualmente e também para o sistema.

Além disso, para o cálculo da geração de entropia no condensador é necessário a temperatura na qual ocorre a transferência de calor no componente. Dessa forma, esse valor é inserido no terceiro parâmetro da função do condensador, em que nesse exemplo se considerou 14°C.

Os resultados obtidos com a execução do código são os seguintes:

**Tabela 30 – Propriedades exemplo 11 - SYSTEM**

SYSTEM	
Property	Value
Energy Imbalance	0.0
Mass Imbalance	0.0
Entropy Generation	3.7169

**Fonte: Autoria própria (2023).**

**Tabela 31 – Propriedades exemplo 11 - compressor: st1 » st2**

compressor: st1 » st2				
Property	State	[kg . kJ/kg]	[kg/s]	[kg . kJ/kg.K]
Input States	st1	986.4011	2.0	5.3916
Output States	st2	1110.4143	2.0	5.4447
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	124.0132		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.0531

**Fonte: Autoria própria (2023).**

**Tabela 32 – Propriedades exemplo 11 - compressor: st3 » st4**

compressor: st3 » st4				
Property	State	[kg . kJ/kg]	[kg/s]	[kg . kJ/kg.K]
Input States	st3	1030.3262	2.0	5.2039
Output States	st4	1163.4528	2.0	5.2524
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	133.1266		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.0485

**Fonte: Autoria própria (2023).**

**Tabela 33 – Propriedades exemplo 11 - combustion\_chamber: st4 » st5**

<b>combustion_chamber: st4 » st5</b>				
<b>Property</b>	<b>State</b>	<b>[kg . kJ/kg]</b>	<b>[kg/s]</b>	<b>[kg . kJ/kg.K]</b>
Input States	st4	1163.4528	2.0	5.2524
Output States	st5	3223.1815	2.0	7.9393
Properties	$\dot{Q}$	2059.7287		
	$\dot{W}$	0.0		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				2.6869

**Fonte: Autoria própria (2023).**

**Tabela 34 – Propriedades exemplo 11 - turbine: st5 » st6**

<b>turbine: st5 » st6</b>				
<b>Property</b>	<b>State</b>	<b>[kg . kJ/kg]</b>	<b>[kg/s]</b>	<b>[kg . kJ/kg.K]</b>
Input States	st5	3223.1815	2.0	7.9393
Output States	st6	2541.6212	2.0	8.0159
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	-681.5603		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.0766

**Fonte: Autoria própria (2023).**

**Tabela 35 – Propriedades exemplo 11 - pump: sta » stb**

<b>pump: sta » stb</b>				
<b>Property</b>	<b>State</b>	<b>[kg . kJ/kg]</b>	<b>[kg/s]</b>	<b>[kg . kJ/kg.K]</b>
Input States	sta	22.4233	0.356	0.0799
Output States	stb	23.1351	0.356	0.0799
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	0.7118		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.0

**Fonte: Autoria própria (2023).**



**Tabela 36 – Propriedades exemplo 11 - turbine: stf » stg**

<b>turbine: stf » stg</b>				
<b>Property</b>	<b>State</b>	<b>[kg . kJ/kg]</b>	<b>[kg/s]</b>	<b>[kg . kJ/kg.K]</b>
Input States	stf	1335.6226	0.356	2.7673
Output States	stg	796.804	0.356	2.7673
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	-538.8185		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.0

**Fonte: Autoria própria (2023).**

**Tabela 37 – Propriedades exemplo 11 - condenser: stg » sta**

<b>condenser: stg » sta</b>				
<b>Property</b>	<b>State</b>	<b>[kg . kJ/kg]</b>	<b>[kg/s]</b>	<b>[kg . kJ/kg.K]</b>
Input States	stg	796.804	0.356	2.7673
Output States	sta	22.4233	0.356	0.0799
Properties	$\dot{Q}$	-774.3807		
	$\dot{W}$	0.0		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.0094

**Fonte: Autoria própria (2023).**

**Tabela 38 – Propriedades exemplo 11 - pump: stj » stk**

<b>pump: stj » stk</b>				
<b>Property</b>	<b>State</b>	<b>[kg . kJ/kg]</b>	<b>[kg/s]</b>	<b>[kg . kJ/kg.K]</b>
Input States	stj	70.57	0.9756	0.2701
Output States	stk	71.2844	0.9756	0.2701
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	0.7144		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.0

**Fonte: Autoria própria (2023).**

Tabela 39 – Propriedades exemplo 11 - turbine: stl » stm

turbine: stl » stm				
Property	State	[kg . kJ/kg]	[kg/s]	[kg . kJ/kg.K]
Input States	stl	322.2363	0.9756	1.04
Output States	stm	295.5585	0.9756	1.04
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	-26.6778		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.0

Fonte: Aatoria própria (2023).

Tabela 40 – Propriedades exemplo 11 - condenser: stm » stj

condenser: stm » stj				
Property	State	[kg . kJ/kg]	[kg/s]	[kg . kJ/kg.K]
Input States	stm	295.5585	0.9756	1.04
Output States	stj	70.57	0.9756	0.2701
Properties	$\dot{Q}$	-224.9885		
	$\dot{W}$	0.0		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.0135

Fonte: Aatoria própria (2023).

Tabela 41 – Propriedades exemplo 11 - heater\_exchanger: [st2, stb] » [st3, stc]

heater_exchanger: [st2, stb] » [st3, stc]				
Property	State	[kg . kJ/kg]	[kg/s]	[kg . kJ/kg.K]
Input States	st2	1110.4143	2.0	5.4447
	stb	23.1351	0.356	0.0799
Output States	st3	1030.3262	2.0	5.2039
	stc	103.2232	0.356	0.3347
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	0.0		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.014

Fonte: Aatoria própria (2023).

**Tabela 42 – Propriedades exemplo 11 - heater\_exchanger: [st6, ste] » [st7, stf]**

heater_exchanger: [st6, ste] » [st7, stf]				
Property	State	[kg . kJ/kg]	[kg/s]	[kg . kJ/kg.K]
Input States	st6	2541.6212	2.0	8.0159
	ste	996.2774	0.356	2.2569
Output States	st7	2202.276	2.0	7.6521
	stf	1335.6226	0.356	2.7673
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	0.0		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.1467

Fonte: Autoria própria (2023).

**Tabela 43 – Propriedades exemplo 11 - heater\_exchanger: [std, st7] » [ste, st8]**

heater_exchanger: [std, st7] » [ste, st8]				
Property	State	[kg . kJ/kg]	[kg/s]	[kg . kJ/kg.K]
Input States	std	323.453	0.356	0.8711
	st7	2202.276	2.0	7.6521
Output States	ste	996.2774	0.356	2.2569
	st8	1529.4516	2.0	6.7022
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	0.0		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.4358

Fonte: Autoria própria (2023).

**Tabela 44 – Propriedades exemplo 11 - heater\_exchanger: [stc, st8] » [std, st9]**

heater_exchanger: [stc, st8] » [std, st9]				
Property	State	[kg . kJ/kg]	[kg/s]	[kg . kJ/kg.K]
Input States	stc	103.2232	0.356	0.3347
	st8	1529.4516	2.0	6.7022
Output States	std	323.453	0.356	0.8711
	st9	1309.2218	2.0	6.272
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	0.0		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.1062

Fonte: Autoria própria (2023).

**Tabela 45 – Propriedades exemplo 11 - heater\_exchanger: [stk, st9] » [stl, st10]**

<b>heater_exchanger: [stk, st9] » [stl, st10]</b>				
<b>Property</b>	<b>State</b>	<b>[kg . kJ/kg]</b>	<b>[kg/s]</b>	<b>[kg . kJ/kg.K]</b>
Input States	stk	71.2844	0.9756	0.2701
	st9	1309.2218	2.0	6.272
Output States	stl	322.2363	0.9756	1.04
	st10	1058.2698	2.0	5.6281
Properties	$\dot{Q}$	0.0		
	$\dot{W}$	0.0		
Imbalance (Energy/Mass)		0.0	0.0	
Entropy Generation				0.1261

**Fonte: Autoria própria (2023).**

A partir dos resultados obtidos é possível validar os cálculos das propriedades do ciclo, pois os balanços de massa e de energia estão iguais a zero, e os valores de geração de entropia estão superiores a zero, comprovando que a resolução está correta.

## 5 CONCLUSÃO

Com base nos resultados expostos, é evidente a versatilidade da biblioteca desenvolvida neste trabalho. Proporcionando, desde ferramentas para a modelagem dos ciclos, até ferramentas para análise e visualização do desempenho dos ciclos. Além disso, demonstra a habilidade de resolver diversos tipos de ciclos termodinâmicos, como o ciclo Rankine, ciclo Brayton, ciclo de refrigeração, e além da possibilidade de desenvolvimento de sistemas com múltiplos ciclos, com diferentes fluidos de trabalho.

A biblioteca se mostrou capaz de facilitar o estudo de ciclos termodinâmicos, pois diferentemente dos modelos convencionais Çengel, Boles e Kanoğlu (2019), Wylen (1985), Jones e Hawkins (1986), Moran e Shapiro (2002), Bejan (2016), possui a habilidade de incorporar a relação de cada componente do sistema e resolver o ciclo automaticamente com base na estrutura do ciclo e as propriedades previamente conhecidas.

Portanto, o objetivo do trabalho, que consistiu em otimizar o processo de modelagem e análise de ciclos termodinâmicos, ao oferecer resultados confiáveis e rápidos, foi atingido. Por meio do desenvolvimento da biblioteca `CycleSolver.jl`, que se evidenciou como uma ferramenta valiosa tanto para a indústria quanto para a pesquisa acadêmica.

## REFERÊNCIAS

- BEJAN, A. **Heat Transfer**. [S.l.]: John Wiley & Sons, 1993.
- BEJAN, A. **Advanced Engineering Thermodynamics**. 4th. ed. New Jersey: John Wiley Sons, 2016.
- BELL, I. 2011. Online. Acesso em 17 de agosto de 2023. Disponível em: <https://achp.sourceforge.net/index.html>.
- BELL, I.; BRADSHAW, C. **PDSim**. 2012. Online. Acesso em 17 de agosto de 2023. Disponível em: <https://pdsim.sourceforge.net/#>.
- 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>.
- BEZANSON, J. *et al.* Julia: A fast dynamic language for technical computing. **Computing Research Repository**, 2012.
- CoolProp. 2023. Online. Acesso em 18 de agosto de 2023. Disponível em: <http://www.coolprop.org>.
- DECHESNE, B. *et al.* **ThermoCycle**. 2019. Online. Disponível em: <https://github.com/thermocycle/Thermocycle-library>.
- DWSIM. 2023. Online. Acesso em 17 de agosto de 2023. Disponível em: <https://dwsim.org/>.
- GOWDA, S. *et al.* High-performance symbolic-numeric via multiple dispatch. **arXiv preprint arXiv:2105.03949**, 2021.
- JONES, J. B.; HAWKINS, G. A. **Engineering Thermodynamics: An Introductory Textbook**. [S.l.]: John Wiley & Sons, 1986.
- JuliaPlots. 2023. Online. Acesso em 18 de agosto de 2023. Disponível em: <https://docs.juliaplots.org/stable/>.
- KROOS, K. A.; POTTER, M. C. **Termodinâmica para engenheiros**. First. São Paulo, SP: Cengage Learning, 2015. ISBN 978-85-221-2198-4.
- MORAN, M. J.; SHAPIRO, H. N. **Princípios de termodinâmica para engenharia**. Fourth. Rio de Janeiro, RJ: Livros Técnicos e Científicos Editora, 2002. ISBN 978-85-216-1340-4.
- NAAKTGEBOREN, C. libTherm — vector schematics of thermodynamic components (unpublished). Private communication. 2023.
- PrettyTables. 2023. Online. Acesso em 18 de agosto de 2023. Disponível em: <https://ronisbr.github.io/PrettyTables.jl/stable/>.
- PropiedadesDeFluidos. 2022. Online. Acesso em 17 de agosto de 2023. Disponível em: <https://personal.us.es/jfc/PropiedadesDeFluidos/descripcion/>.
- PyCall. 2023. Online. Acesso em 18 de agosto de 2023. Disponível em: <https://github.com/JuliaPy/PyCall.jl>.

SmoWeb. 2018. Online. Acesso em 17 de agosto de 2023. Disponível em: <http://platform.sysmoltd.com/>.

STRUCHTRUP, H. **Thermodynamics and energy conversion**. Springer Berlin Heidelberg, 2014. ISBN 978-3-662-43715-5. Disponível em: <http://doi.org/10.1007/978-3-662-43715-5>.

Symbolics. 2023. Online. Acesso em 18 de agosto de 2023. Disponível em: <https://symbolics.juliasymbolics.org/stable/>.

SymPy. 2021. Online. Acesso em 18 de agosto de 2023. Disponível em: <https://www.sympy.org/pt/index.html>.

WebThermodynamics. 2023. Online. Acesso em 17 de agosto de 2023. Disponível em: <https://github.com/dvd101x/web-thermodynamics>.

WYLEN, G. **Fundamentals of classical thermodynamics**. Fourth. New York: Wiley, 1985. ISBN 978-04-718-0014-9.

ÇENGEL, Y. A.; BOLES, M. A. **Termodinâmica: uma abordagem de engenharia**. 7ª edição em português. ed. [S.l.]: McGraw-Hill, 2013.

ÇENGEL, Y. A.; BOLES, M. A.; KANOĞLU, M. **Thermodynamics: an engineering approach**. 9th. ed. New York: McGraw-Hill book company, inc., 2019.

## APÊNDICES



## **APÊNDICE A – Código desenvolvido**

O código desenvolvido neste trabalho está disponível em <https://github.com/vFilipaki/CycleSolver.jl>.

**Figura 18 –** Captura de tela da página GitHub que hospeda a biblioteca desenvolvida no trabalho, proporcionando uma visão geral do projeto e seu ambiente de código-fonte.

The screenshot shows the GitHub repository page for `CycleSolver.jl`. The repository is public and has 7 branches and 2 tags. The main content area displays the README.md file, which features a logo for CycleSolver and a description: "Package for solving thermodynamic cycles in steady state". The right sidebar contains information about the package, including a description, tags, and release information.

**Repository Information:**

- Repository: `CycleSolver.jl` (Public)
- Navigation: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings
- Branches: 7 branches, Tags: 2 tags
- Actions: Go to file, Add file, Code

**Recent Commits:**

Commit Message	Author	Time	Commits
vFilipaki added Visualization.jl options	e8ee3d	27 minutes ago	81
.github/workflows: codecov.yml moved to src		4 days ago	
docs: Update Docs sample images		1 hour ago	
src: added Visualization.jl options		27 minutes ago	
test: update hypotheses.test - more tests added		4 days ago	
.gitignore: Settings		3 weeks ago	
LICENSE: Settings		3 weeks ago	
Project.toml: states.test added		2 weeks ago	
README.md: Update README.md - Change installation		18 hours ago	
codecov.yml: Update codecov.yml		4 days ago	

**README Content:**

**CycleSolver**

Package for solving thermodynamic cycles in steady state  
[Explore the docs »](#)

How to install · How to cite this project · See examples

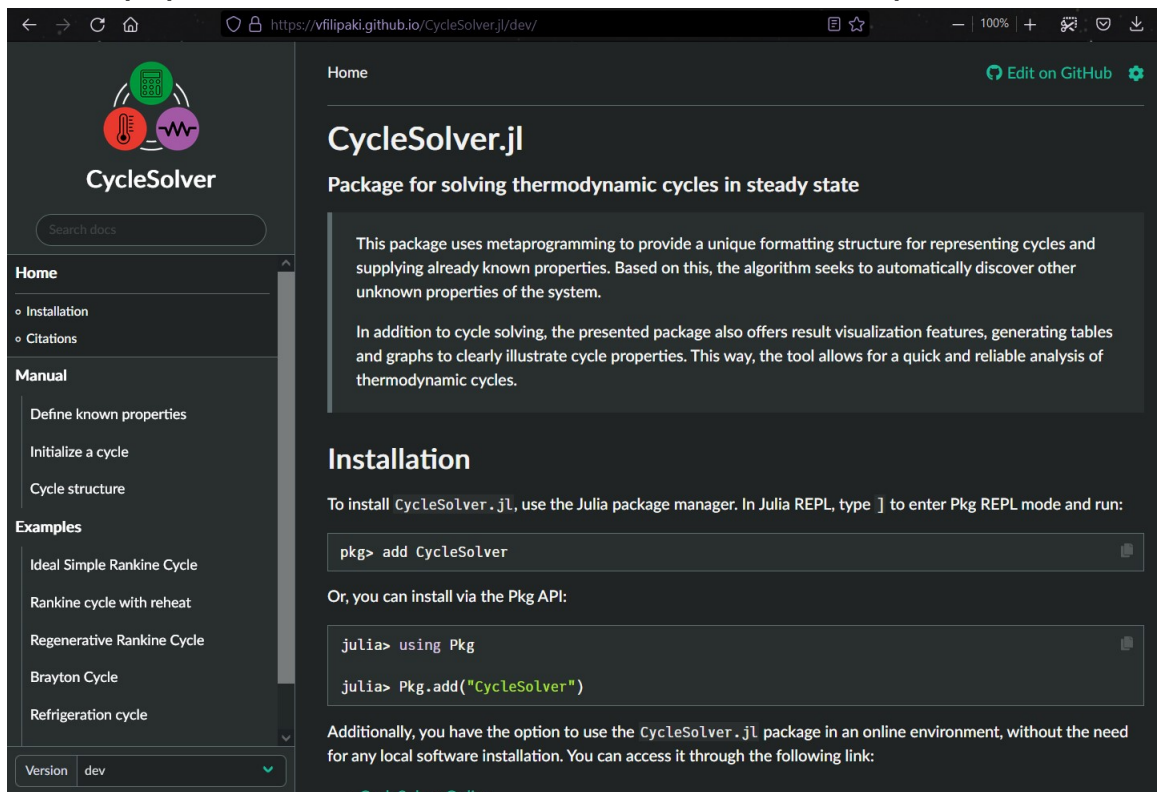
**Right Sidebar:**

- About:** The package offers a unique framework for formatting the cycle's structure and providing already known properties, enabling the resolution of thermodynamic cycles in steady state.
- Tags:** graphics, solver, julia-language, cycle, efficiency, thermodynamics, refrigeration, rankine-cycle, brayton-cycle
- Releases:** 2 releases. Latest: release 0.2.0 (2 weeks ago).
- Deployments:** 68 deployments. Latest: github-pages (21 minutes ago).
- Languages:** Julia 100.0%

Fonte: Autoria própria (2023).

A documentação da biblioteca desenvolvida está disponível em <https://vfilipaki.github.io/CycleSolver.jl>.

**Figura 19 – Captura de tela da página de documentação da biblioteca desenvolvida no trabalho, proporcionando um manual de como usar a biblioteca e exemplos de uso.**



Home [Edit on GitHub](#)

# CycleSolver.jl

Package for solving thermodynamic cycles in steady state

This package uses metaprogramming to provide a unique formatting structure for representing cycles and supplying already known properties. Based on this, the algorithm seeks to automatically discover other unknown properties of the system.

In addition to cycle solving, the presented package also offers result visualization features, generating tables and graphs to clearly illustrate cycle properties. This way, the tool allows for a quick and reliable analysis of thermodynamic cycles.

## Installation

To install `CycleSolver.jl`, use the Julia package manager. In Julia REPL, type `]` to enter Pkg REPL mode and run:

```
pkg> add CycleSolver
```

Or, you can install via the Pkg API:

```
julia> using Pkg
julia> Pkg.add("CycleSolver")
```

Additionally, you have the option to use the `CycleSolver.jl` package in an online environment, without the need for any local software installation. You can access it through the following link:

Version: `dev`

**Fonte: Autoria própria (2023).**