

UNIVERSIDADE FEDERAL DA GRANDE DOURADOS
FACULDADE DE ENGENHARIA – CURSO DE ENGENHARIA MECÂNICA
TRABALHO DE CONCLUSÃO DE CURSO

**DESENVOLVIMENTO DE UM PROGRAMA EM PYTHON PARA DETERMINAÇÃO
DA EFICIÊNCIA DE GERADORES DE VAPOR**

Por

João Victor Schluchting Nunes

Dourados, Abril de 2023

DESENVOLVIMENTO DE UM PROGRAMA EM PYTHON PARA DETERMINAÇÃO
DA EFICIÊNCIA DE GERADORES DE VAPOR

Por

João Victor Schluchting Nunes

TCC apresentado ao Curso de Engenharia Mecânica da Faculdade de Engenharia da Universidade Federal da Grande Dourados, como parte dos requisitos para obtenção do diploma de Engenheiro Mecânico, sob a orientação do Prof. Dr. Ramon Eduardo Pereira Silva

Área de concentração: 3.05.02.01-2
Termodinâmica

Dourados, Abril de 2023

RESUMO

A eficiência na geração de vapor pode ser vista de forma direta, que foca apenas na quantidade de energia produzida na forma de vapor, e de forma indireta, que visa detalhar cada perda de energia relacionada na produção de vapor. O cálculo de forma indireta traz grandes vantagens na hora de aplicar melhorias no sistema. Para auxiliar a visualização dessas perdas o presente projeto traz o desenvolvimento de um aplicativo em linguagem de código aberto com interface interativa. A partir da definição do combustível e dados da operação da caldeira, o programa retorna a quantidade de energia perdida e também um gráfico para melhor visualização.

Palavras-Chave: Vapor, Eficiência, Aplicativo, Interface

ABSTRACT

The efficiency in steam generation can be obtained directly, which focuses only on the amount of energy produced in the form of steam, and indirectly, which aims to detail each energy loss in steam production. Indirect calculation brings great advantages when applying improvements to the system. To help visualize these losses, this project brings the development of an application in free code language with an interactive interface. From the fuel definition and boiler operation data, the program returns the amount of energy lost and also a chart for better visualization.

Keywords: Steam, Efficiency, Application, Interface

1. MEMORIAL DESCRITIVO DE SOLICITAÇÃO DE DESENVOLVIMENTO DE PROTÓTIPO

1.1 OBJETIVO

O objetivo deste Memorial é descrever as necessidades, justificativas e requisitos de projeto para o desenvolvimento de um programa que calcule a eficiência energética de um gerador de vapor pelo método indireto.

1.2 NECESSIDADES ESPECÍFICAS

O funcionamento e a análise energética de Geradores de Vapor são parte integrante da ementa da disciplina “Projetos de Sistemas Térmicos” tanto no curso de Engenharia de Energia quanto no curso de Engenharia Mecânica. Os balanços de massa e de energia são realizados de forma analítica pelo professor em um ponto de projeto determinado e para uma condição de operação.

Desta forma a necessidade de um programa que consiga fazer a simulação destes balanços, ainda que de forma analítica, mostrando a interferência decorrente da alteração das variáveis do processo na eficiência do gerador de vapor torna-se premente de forma a facilitar o entendimento da operação destes geradores de vapor.

Assim, solicita-se o desenvolvimento de um protótipo de um programa de computador, em linguagem livre e aberta e que possua elementos gráficos e intuitividade com finalidade de auxílio pedagógico ao professor e aos alunos.

1.3 REQUISITOS DE PROJETO

Assim, as necessidades apontadas anteriormente são traduzidas nos seguintes requisitos de projeto:

1. O programa deve permitir o cálculo da eficiência de geradores de vapor pelo método indireto.
2. O programa deve ser desenvolvido em linguagem aberta e gratuita.
3. O programa deve ter interface intuitiva para o usuário.
4. O programa deve permitir a utilização, no mínimo, dos seguintes combustíveis:
 - a. Bagaço de cana-de-açúcar.

- b. Lenha ou cavacos de eucalipto.
5. Para o cálculo utilizando-se os dois combustíveis listado no item 4, o programa deve permitir a entrada da variação do teor de umidade do combustível.
6. Para a utilização do bagaço da cana-de-açúcar, o programa deve permitir também a entrada da variação do teor de açúcar no combustível
7. O programa deve permitir também a entrada da composição de combustíveis tanto em base molar quanto em base mássica.
8. O programa deve permitir os seguintes dados de entrada utilizando as unidades descritas entre colchetes:
 - a. Vazão mássica de vapor [ton/h]
 - b. Temperatura de saída do vapor [°C]
 - c. Pressão de trabalho do gerador de vapor [MPa]
 - d. Excesso de ar [-]
 - e. Temperatura da água de alimentação [°C]
 - f. Temperatura de entrada do ar [°C]
 - g. Teor de cinzas do combustível [%]
 - h. Temperatura dos gases na chaminé [°C]
 - i. Calor específico dos gases da chaminé [kJ/kg.K]
 - j. Perdas de calor para o ambiente [%]
 - k. Perdas por combustão incompleta [%]
 - l. Perdas por vazão de purgas [%]
9. O programa deve mostrar os seguintes dados de saída utilizando as unidades descritas entre colchetes:
 - a. Poder Calorífico Inferior (PCI) do combustível [kJ/kg]
 - b. Razão ar/combustível estequiométrica [-]
 - c. Vazão mássica de combustível [ton/h].
 - d. Calor injetado pelo combustível na caldeira [kW]
 - e. Perdas de calor para o ambiente [kW]
 - f. Perdas de calor por combustão incompleta [kW]
 - g. Perdas de calor pelos gases de combustão [kW]
 - h. Perdas de calor por purgas de lama [kW]
10. O programa deve mostrar eficiência térmica e a distribuição das perdas de forma percentual.

1.4 VALIDAÇÃO

O programa deverá ser validado através da simulação de um exercício analítico cujos resultados deverão ser comparados e mostrados no projeto.

1.5 ENTREGA DO PROJETO

Ao término do projeto deverão ser entregues:

1. O protótipo do programa operacional e validado.
2. O relatório do projeto contendo:
 - a. Objetivo do projeto.
 - b. Fundamentação teórica.
 - c. Validação do programa
3. Um tutorial de utilização do programa contendo:
 - a. Instruções para instalação do programa.
 - b. Instruções para utilização do programa.
 - c. Exemplo da aplicação.
4. O script do programa em formato .pdf

2. DESENVOLVIMENTO DE UM PROGRAMA EM PYTHON PARA DETERMINAÇÃO DA EFICIÊNCIA DE GERADORES DE VAPOR

2.1 OBJETIVO DO PROJETO

2.1.1 Objetivo Geral

O objetivo deste TCC é desenvolver um programa em linguagem aberta para calcular a eficiência de caldeiras de forma indireta com elementos gráficos e intuitivos.

2.1.2 Objetivos Específicos

- Detalhar as perdas em um sistema de gerador de vapor e demonstrar sua eficiência de forma gráfica;
- Promover uma ferramenta para auxílio pedagógico que demonstre a interferência causada na alteração de variáveis na geração de vapor de forma rápida.

2.2 NECESSIDADES E JUSTIFICATIVAS

A eficiência dos geradores de vapor é importante para determinar seu desempenho, uma alta eficiência implica em economia de combustível. Para auxiliar a reduzir perdas e desperdícios de energia, faz-se necessário um programa que demonstre fatores que podem ser modificados em curto prazo de forma gráfica, facilitando o entendimento do usuário.

2.3 INTRODUÇÃO

2.3.1 Geradores de vapor

A geração de vapor ocorre em caldeiras, que são chamadas também de geradores de vapor. As caldeiras podem variar significativamente em tamanho e forma e trabalham com pressão superior à pressão atmosférica, produzindo vapor a partir da energia térmica fornecida por uma fonte qualquer. O vapor é gerado em uma caldeira pela transferência do calor dos gases quentes para a água que nela circula. A energia para gerar o vapor pode ser obtida através da combustão de combustíveis ou recuperação de calor residual de processo. Após a geração e vapor, o efeito da pressão faz o vapor fluir da caldeira para o sistema de distribuição. L. A. H. Nogueira (2005) afirma que as partes mais importantes do sistema de geração de vapor são: caldeira, sistema de alimentação do combustível, sistema de injeção de ar de combustão, sistema de alimentação de

água e sistema de tiragem dos gases de exaustão. Dentro do volume de controle da caldeira o processo de geração de vapor pode ser esquematizado como na Figura 2.1.

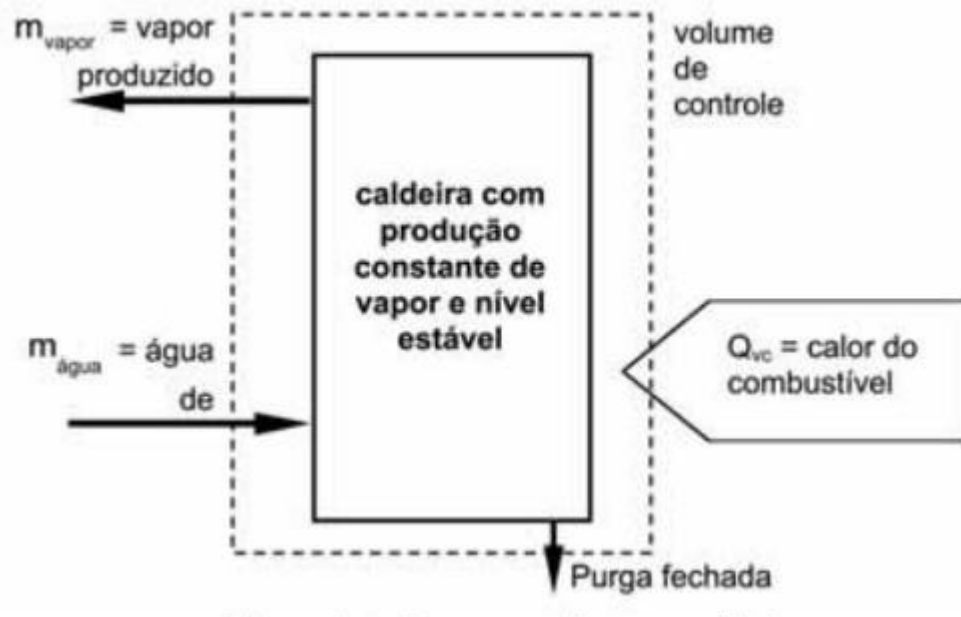


Figura 2.1 – Esquema da caldeira. Fonte: L.A. H. Nogueira (2005)

Na caldeira com produção constante pode ser considerado um processo com escoamento em regime permanente, dessa forma a massa contida no volume de controle não sofre variação com o tempo. Portanto, pelo princípio de conservação da massa pode-se afirmar que:

$$\dot{m}_{\text{água}} = \dot{m}_{\text{vapor}} \quad (2.1)$$

2.3.2 Método Direto e Método Indireto

A eficiência energética de um sistema, indica o quanto um equipamento real se aproxima de um comportamento ideal, quando não existem perdas. (NOGUEIRA, L. A. H., 2005) A eficiência é expressada pela razão da energia aproveitada e a energia fornecida.

$$\eta = \frac{\dot{Q}_u}{\dot{Q}_c} \quad (2.2)$$

Onde:

\dot{Q}_u – Calor transferido ao vapor; [kW]

\dot{Q}_c – Fluxo de calor produzido pelo combustível. [kW]

Através do método direto, a eficiência térmica dos geradores de vapor é dada pela razão entre a energia produzida em forma de vapor e consumo de energia fornecida pelo combustível.

A energia presente no vapor é definida pelo produto entre sua vazão em massa e o ganho de entalpia no aquecimento (NOGUEIRA, L. A. H., 2005), portanto:

$$\eta_d = \frac{\dot{m}_v(h_v - h_a)}{\dot{m}_c PCI} \quad (2.3)$$

Onde:

η_d – eficiência pelo método direto;
 \dot{m}_v – vazão mássica de vapor [kg/s];
 \dot{m}_c – vazão mássica de combustível [kg/s];
 h_v – entalpia do vapor [kJ/kg];
 h_a – entalpia da água de alimentação [kJ/kg];
 PCI – poder calorífico inferior do combustível [kJ/kg].

Outro modo que pode ser usado para determinar a eficiência térmica da caldeira é o método indireto onde são analisadas as perdas na energia fornecida pelo combustível de forma que a eficiência é dada por:

$$\eta_i = 1 - \frac{\Sigma_{perdas}}{\dot{m}_c PCI} \quad (2.4)$$

Quando utilizado o método direto, tem-se a vantagem de obter um cálculo de forma rápida e com menos variáveis, porém fica difícil de identificar os fatores que estão influenciando na eficiência do gerador de vapor. No método indireto é possível detalhar quais são as perdas de energia no sistema, gerando assim oportunidades de melhoria no desempenho do equipamento, a desvantagem desse método é existência de perdas de difícil determinação e necessitam ser aproximadas.

2.3.3 Desenvolvimento de programas.

Foi desenvolvido um aplicativo para detalhar as perdas usando a linguagem de programação Python. Para criar uma interface foi utilizado o módulo PyQt5. O objetivo do programa é facilitar a apresentação do cálculo da eficiência da geração de vapor utilizando o método indireto, detalhando onde se encontram as perdas de eficiência, dessa forma possibilita-se demonstrar quais alterações de parâmetros podem reduzir mais significativamente as perdas, levando à maior eficiência do sistema. A presente aplicação pode ser utilizada para geradores de vapor reais, e também, por ser desenvolvida em uma linguagem de código aberto é uma grande ferramenta para uso didático. Utilizando os parâmetros inicialmente definidos, dados do combustíveis e as condições de operação que são facilmente obtidas, o programa retorna não só a quantidade de

energia perdida, mas também informações relevantes, como o consumo horário de combustível e seu poder calorífico inferior.

2.3.4 Python

Python é uma linguagem de programação amplamente utilizada, interpretada, orientada a objetos e de alto nível com semântica dinâmica, usada para programação de propósito em geral.(PYTHON INSTITUTE, 2023) Foi criado por Guido van Rossum e lançado em 1991 com objetivos de ser uma linguagem fácil, intuitiva, poderosa e também de código aberto, para que qualquer pessoa possa contribuir para seu desenvolvimento. A principal desvantagem do Python é a sua velocidade de execução, que é lenta em comparação às linguagens compiladas, como C. Mesmo assim está entre as linguagens mais utilizadas do mundo (SLASHDATA, 2022). E, de acordo com (LUTZ; ASCHER, 2007), há muitas vantagens em utilizar Python, entre elas estão:

- Orientado a objetos;
- Gratuito;
- Portável;
- Poderoso;
- Fácil de usar e aprender;

2.4 FUNDAMENTAÇÃO TEÓRICA

2.4.1 Combustíveis

A combustão é a rápida combinação química do oxigênio com os elementos do combustível, resultando na produção de calor. É feita misturando combustível e ar a temperaturas elevadas. O ar fornece oxigênio, que reage quimicamente com carbono, hidrogênio e outros elementos do combustível para produzir calor (WOODRUFF; LAMMERS, H.B.; LAMMERS, T.F., 2004). Os combustíveis podem ser classificados em sólidos, líquidos ou gasosos. Os combustíveis mais utilizados são, óleo diesel, óleo combustível, gás natural, gás GLP e entre os combustíveis sólidos o mais importante é a biomassa, como lenha e bagaço de cana, sendo o bagaço de cana açúcar a maior fonte de biomassa no Brasil (ICATERM, 2023)

PODER CALORÍFICO INFERIOR

A energia térmica fornecida durante o processo de combustão dos combustíveis pode ser avaliada pelo seu poder calorífico. São definidos dois tipos de poder calorífico: Poder Calorífico

Superior (PCS), quando a água presente nos produtos de combustão está na forma líquida; e Poder Calorífico Inferior (PCI), quando esta água apresenta-se como vapor, situação que efetivamente ocorre nos produtos de combustão nas chaminés. (NOGUEIRA, L. A. H., 2005)

2.4.2 Balanço de massa e energia no gerador de vapor

Caldeiras em operação normal produzem vapor em um processo constante. Ar, água e combustível são fornecidos enquanto o vapor é produzido e produtos de combustão como cinzas e gases são descartados. Para o bom funcionamento do gerador de vapor, instrumentos de controle devem sempre ser checados e calibrados para fazer ajustes de rotina e detectar dificuldades (WOODRUFF; LAMMERS, H.B.; LAMMERS, T.F., 2004).

É possível determinar a eficiência da caldeira de forma mais detalhada com o balanço de energia, calculando as perdas de calor. O balanço típico de energia pode ser visto na Figura 2.2

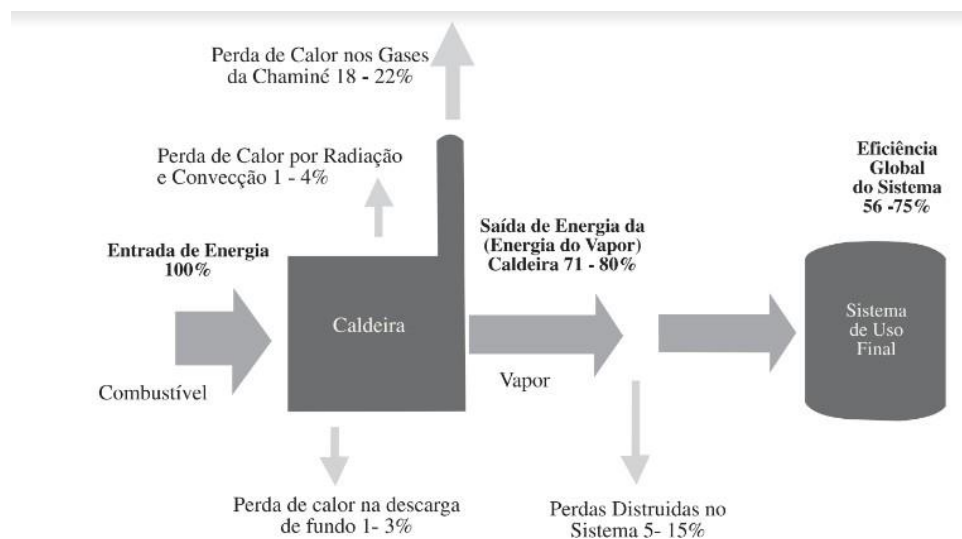


Figura 2.2 – Balanço típico de energia com perdas (NOGUEIRA, L. A. H.; ROCHA; NOGUEIRA, F.J.H., 2005)

A Primeira Lei da Termodinâmica enuncia que a energia não pode ser criada nem destruída durante um processo; ela pode apenas mudar de forma. Embora a essência da Primeira Lei seja a existência da propriedade da energia total, a conservação da energia está implícita no enunciado da Primeira Lei (ÇENGEL; BOLES, 2013). Dessa forma, tem-se o balanço de energia:

$$\dot{Q}_{entra} = \dot{Q}_{sai} \quad (2.5)$$

Portanto:

$$\dot{Q}_c = \dot{Q}_u + \dot{Q}_1 + \dot{Q}_2 + \dot{Q}_3 + \dot{Q}_4 + \dot{Q}_5 \quad (2.6)$$

Onde:

\dot{Q}_c – Fluxo de calor fornecido pelo combustível; [kW]

\dot{Q}_u – Fluxo de calor utilizado na geração de vapor; [kW]

\dot{Q}_1 – Fluxo de calor perdido nos gases de exaustão; [kW]

\dot{Q}_2 – Fluxo de calor perdido nas cinzas; [kW]

\dot{Q}_3 – Fluxo de calor perdido em purgas; [kW]

\dot{Q}_4 – Fluxo de calor perdido para o ambiente; [kW]

\dot{Q}_5 – Fluxo de calor perdido na combustão incompleta. [kW]

Implicando assim que o calor efetivamente utilizado na produção de vapor é da diferença entre o calor produzido pelo combustível e a soma das perdas no sistema, conforme mostra a Equação 2.7.

$$\dot{Q}_u = \dot{Q}_c - \Sigma_{perdas} \quad (2.7)$$

Portanto, de acordo com a Equação 2.2, a eficiência da caldeira também pode ser expressa como:

$$\eta = \frac{\dot{Q}_c - \Sigma_{perdas}}{\dot{Q}_c} \quad (2.8)$$

Ou:

$$\eta = 1 - \frac{\Sigma_{perdas}}{\dot{Q}_c} \quad (2.9)$$

2.4.3 Perdas

- **Perda nos gases de exaustão**

As perdas na temperatura dos gases de escape é a maior parcela das perdas. Quanto maior a temperatura dos gases que saem pela chaminé, maior é a perda de energia. Pela conservação da massa, o que entra de ar e combustível deve sair da caldeira, não se misturando com a água de alimentação. Pode se considerar para cálculos práticos que grande parte dos gases é constituída por nitrogênio atmosférico, assim, para mesma temperatura, o calor específico dos gases de combustão é semelhante ao do ar. Essa perda é calculada pela Equação 2.10.

$$P_c = (\dot{m}_{ar} + \dot{m}_c)C_{p,ar}(T_g - T_{ref}) \quad (2.10)$$

Onde:

\dot{m}_{ar} – Fluxo mássico de ar [kg/s];
 \dot{m}_c – Fluxo mássico de combustível[kg/s];
 $C_{p,ar}$ – Calor específico do ar[kJ/kgK];
 T_g – Temperatura dos gases [K];
 T_{ref} – Temperatura de referência adotada[K];
 P_c – Perda nos gases [kW].

- **Perda no teor de cinzas do combustível**

Em combustíveis sólidos, existe a perda de calor nas cinzas, o teor das cinzas é a quantidade de cinzas presente no combustível, estas são descartadas em altas temperaturas. É calculada por:

$$P_{tc} = 1170 * m_{cinza} \quad (2.11)$$

Onde o valor 1170 representa um valor médio aproximado da entalpia sensível das cinzas e m_{cinza} é a razão entre a massa de cinzas e a massa de combustível.

- **Perda por vazão de purgas**

A água de alimentação da caldeira costuma sempre apresentar algumas impurezas. Conforme a caldeira gera vapor, essas impurezas permanecem dentro da caldeira, e com o tempo sua concentração aumenta em forma de lama. Para realizar o controle dessas impurezas, é realizada a purga da caldeira, conhecida como purgas

de lama. Essa retirada de água também está associada com a perda de energia, que pode ser calculada pela Equação 2.12.

$$P_p = \dot{m}_p C_{p,agua} (T_p - T_{ref}) \quad (2.12)$$

Onde:

$C_{p,agua}$ – Calor específico da água [kJ/kgK]

\dot{m}_p – Vazão média de água purgada em relação a vazão de combustível

T_p – Temperatura da água purgada [K]

- **Perda para o ambiente**

Devido à alta temperatura de operação da caldeira, também é perdido calor para o ambiente externo por convecção e radiação.

Devido às diversas e complexas medições necessárias para sua obtenção, a determinação dessa perda é difícil de avaliar, e de maneira geral adota-se perdas de 1 a 5% da energia fornecida (NOGUEIRA, 2005).

- **Perda por combustão incompleta**

Essa perda é característica do combustível sólido, e está relacionada à parte não queimada de combustível presente nas cinzas, parte do combustível é envolvido pela escória, isolando-o do ar.

2.4.4 Calor fornecido pelo combustível

O fluxo de calor fornecido pelo combustível no processo de geração de vapor é obtido pelo produto do seu fluxo em massa e seu PCI.

$$\dot{Q}_c = \dot{m}_c PCI \quad (2.13)$$

Obtenção do PCI

De acordo com Strobel (s.d.), é possível obter o PCI para combustíveis líquidos e sólidos, a partir da sua fração mássica, em [kJ/kg] utilizando a equação 2.14

$$PCI_{l,s} = 339,13C + 1029,95H + 108,85(S - O) - 25,12W \quad (2.14)$$

Onde:

C – Carbono;

H – Hidrogênio;

O – Oxigênio;

S – Enxofre;

W – Umidade

A figura 2.2 traz uma tabela do PCI de alguns combustíveis em sua condição típica de utilização na caldeira, de acordo com Nogueira (2005).

COMBUSTÍVEL	PODER CALORÍFICO INFERIOR	DENSIDADE
Óleo combustível B1	9.590 kcal/kg	1000 kg/m ³
Gás natural (típico)	8.800 kcal/m ³	-
GLP	11.100 kcal/kg	-
Lenha	3.100 kcal/kg	400 kg/m ³
Bagaço de cana	2.130 kcal/kg	-
Carvão vegetal	6.460 kcal/kg	260 kg/m ³
Carvão mineral ¹	2.850 kcal/kg	-

Figura 2.3 – PCI de alguns combustíveis (NOGUEIRA. L. A. H., 2005)

Bagaço de Cana

O PCI do bagaço de cana é obtido conforme sua umidade e teor de sacarose no bagaço em [kcal/kg] utilizado a fórmula de Hugot(1977) de acordo com Carvalho et al. (2019).

$$PCI = 4250 - 12s - 48,5w \quad (2.15)$$

Onde s é o teor de sacarose no bagaço e w é o teor de umidade em % de massa.

Lenha de Eucalipto

O PCI da lenha de eucalipto é influenciado por sua umidade, a evaporação da água vai consumir parte do calor liberado pela combustão. (NOGUEIRA. L. A. H., 2005) Neste caso especial, o PCI da lenha pode ser obtido através de interpolações usando a tabela mostrada na Figura 2.4.

Tabela II.2 - Influência da umidade da lenha no PCI e do tempo na umidade

UMIDADE	PCI	UMIDADE	PCI	UMIDADE DA LENHA DE EUCALIPTO	
(%)	[kcal / kg]	(%)	[kcal / kg]	(dias após)	(%)
0	4438,3	45	2171,1	No corte	45
5	4186,4	50	1919,2	30	36 %
10	3934,5	55	1667,3	60	30 %
15	3682,6	60	1415,3	90	27 %
20	3430,7	65	1163,4	120	25 %
25	3178,8	70	911,5	150	23 %
30	2926,8	75	659,6		
35	2674,9	80	407,7		
40	2423,0	85	155,8		

Figura 2.4 – PCI da lenha conforme umidade (NOGUEIRA, L. A. H., 2005)

2.4.5 Relação ar combustível e excesso de ar

Para reações de combustão de cada combustível, existe uma quantidade determinada de ar que deve ser fornecido. A proporção de ar e combustível para combustão completa é chamada relação ar combustível estequiométrica. Através da Equação 2.16, podemos obter a razão ar combustível. (STROBEL, s.d.)

$$AC_{massa} = 0,1149C + 0,3448H + 0,0431(S - O) \quad (2.16)$$

Porém, para que todo o combustível seja queimado, é necessário garantir uma certa quantidade de excesso de ar, entretanto, altas quantidades de ar em excesso reduzem significativamente a eficiência. O excesso mínimo de ar adotado depende do tipo de combustível e do sistema de combustão, pois trata buscar uma mistura adequada entre o combustível e o comburente. Gases combustíveis costumam necessitar de 5 a 10% de excesso de ar, para combustíveis líquidos pode ser requerido desde menos de 10 até mais de 30% de excesso, já para combustíveis sólidos pode ser necessário maior proporção de excesso de ar. (NOGUEIRA, L. A. H., 2005)

2.4.6 Eficiência indireta

Conhecendo as perdas e energia fornecida, pode-se obter a eficiência pelo método indireto. Substituindo a Equação 2.13 na Equação 2.9 tem-se:

$$\eta = 1 - \frac{\Sigma_{perdas}}{\dot{m}_c PCI} \quad (2.17)$$

Sabendo que a maior parte das perdas está na temperatura de saída dos gases da chaminé, a somatória das perdas, com exceção da perda na chaminé, pode ser definida como k , e a Equação 2.17 rescrita como:

$$\eta = 1 - \left[k + \frac{(\dot{m}_{ar} + \dot{m}_c) c_{p,ar} (T_g - T_{ref})}{\dot{m}_c PCI} \right] \quad (2.18)$$

Ou também:

$$\eta = 1 - \left[k + \frac{\lambda (AC_{massa} + 1) c_{p,ar} (T_g - T_{ref})}{PCI} \right] \quad (2.19)$$

Onde λ é o coeficiente de excesso de ar adotado, que é a razão entre a relação ar combustível utilizada no processo e a relação ar combustível estequiométrica.

2.5 VALIDAÇÃO DO PROGRAMA

Para realizar a validação do programa, os resultados obtidos no programa serão comparados com os cálculos de forma analítica do exercício proposto por Strobel(s.d):

Exercício 4) Caldeira a óleo combustível, com C=80%, H=10%, S=3%, O=5%, W=2%, produzindo 100 ton/h de vapor a 40 bar, 340°C, apresenta temperatura de saída dos produtos de combustão de 345°C. Com a instalação de um pré-aquecedor de ar, esta temperatura baixou para 240°C. Devido ao ar aquecido na entrada do queimador, foi possível modificar o valor do excesso de ar de 15%, na situação anterior, para 8% com o pré-aquecedor. A temperatura da água de alimentação é 65°C. Qual a economia de combustível proporcionada? Considere a temperatura ambiente em 20°C.

Primeira condição:

Na primeira condição a tem-se:

Produção de vapor: 100 [ton/h];
 Pressão de trabalho: 40 [bar];
 Temperatura do vapor: 340[°C];
 Temperatura dos gases na chaminé: 345 [°C];
 Temperatura do ar de alimentação: 20 [°C];
 Temperatura da água de alimentação: 65 [°C];
 Coeficiente de excesso de ar: 1,15.

Obtem-se o PCI do combustível com a Equação 2.14:

$$PCI = 339,13 * 80 + 1029,95 * 10 + 108,85(3 - 5) - 25,12 * 2 \text{ [kJ/kg]}$$

$$PCI = 37161,96 \text{ [kJ/kg]}$$

A relação ar combustível é calculada com a Equação 2.16:

$$AC = 0,1149 * 80 + 0,3448 * 10 + 0,0431(3 - 5)$$

$$AC = 12,5538 \text{ [kg de ar / kg de combustível]}$$

O calor utilizado na geração de vapor é:

$$\dot{Q}_u = \dot{m}_v(h_{vapor} - h_{agua}) \text{ [kW]}$$

$$\dot{Q}_u = 100 * \frac{1000}{3600} (3068,14 - 272,12) \text{ [kW]}$$

$$\dot{Q}_u = 77667,22 \text{ [kW]}$$

A eficiência é dada pela Equação 2.19:

$$\eta = 1 - \left[0 + \frac{1,15(12,5538+1)C_{p,ar}(345-20)}{37161,96} \right]$$

Portanto:

$$\eta = 0,8609$$

O fluxo de combustível é, de acordo com a Equação 2.2:

$$\dot{m}_c = \frac{77667,22}{37161,96 * 0,8609} \text{ [kg/s]}$$

$$\dot{m}_c = 2,43 \text{ [kg/s]}$$

Segunda condição:

Na segunda condição o coeficiente de excesso de ar é alterado para 1,08 e a temperatura de saída na chaminé é reduzida para 240°C.

Como o calor específico dos gases de combustão é muito próximo ao do ar, e considerando que todo o calor perdido nos gases foi transferido ao ar de alimentação e o princípio de conservação da massa, pode-se afirmar que:

$$\Delta T_{ar} = \Delta T_{gás} \tag{2.20}$$

Dessa forma a temperatura de alimentação do ar é elevada para 125°C, e a nova eficiência é:

$$\eta = 1 - \left[0 + \frac{1,08(12,5538+1)c_{p,ar}(240-125)}{37161,96} \right]$$

$$\eta = 0,9537$$

Nessas condições o fluxo de combustível é:

$$\dot{m}_c = \frac{77667,22}{37161,96 \cdot 0,9537} \text{ [kg/s]}$$

$$\dot{m}_c = 2,19 \text{ [kg/s]}$$

Diferença de fluxo de combustível:

$$\Delta \dot{m}_c = 2,43 - 2,19 = 0,24 \text{ [kg/s]} \text{ ou } 864 \text{ [kg/h]}$$

A economia de combustível proporcionada pela instalação de um pré-aquecedor de ar é de 864 kg/h

Utilizado o programa

Com a utilização do programa para realizar os cálculos, as figuras 2.5 e 2.6 mostram os resultados para a primeira condição. Para segunda condição proposta, os resultados estão mostrados nas Figuras 2.7 e 2.8.

Rendimento Indireto de Caldeiras

Arquivo

Combustível		Dados		Resultados	
Fração Mássica		Produção de vapor	100 [Ton/h]	Calor gerado pelo combustível: 90113.20 kW	
C	80 % Massa	Temperatura do Vapor	340 [°C]	Calor útil: 77583.75 kW	
H	10 % Massa	Pressão de trabalho	4 [Mpa]	PERDAS	
O	5 % Massa	Coefficiente de excesso de Ar	1.15 [-]	Perda combustão incompleta: 0.00 kW	
S	3 % Massa	Temperatura da água de alimentação	65 [°C]	Perda no calor das cinzas: 0.00 kW	
W	2 % Massa	Temperatura do ar de entrada	20 [°C]	Perda nos gases de escape: 12529.46 kW	
PCI do combustível: 37161.96 kJ/kg		Teor de cinzas do combustível	0 [%]	Perda por purgas: 0.00 kW	
Vazão ar/combustível estequiométrica: 12.55		Temperatura dos gases na chaminé	345 [°C]	Perda para o ambiente: 0.00 kW	
Vazão mássica de combustível: 8729.56 kg/h		Calor específico dos gases	1.02 [KJ/KgK]		
		Perdas			
		Transferência de calor para o ambiente	0 %		
		Combustão Incompleta	0 %		
		Purgas de lama	0 %		
[Calcular Eficiência]					
Exemplo		Novos dados			

Figura 2.5 – Interface do programa com resultados do primeiro caso. Fonte: O autor

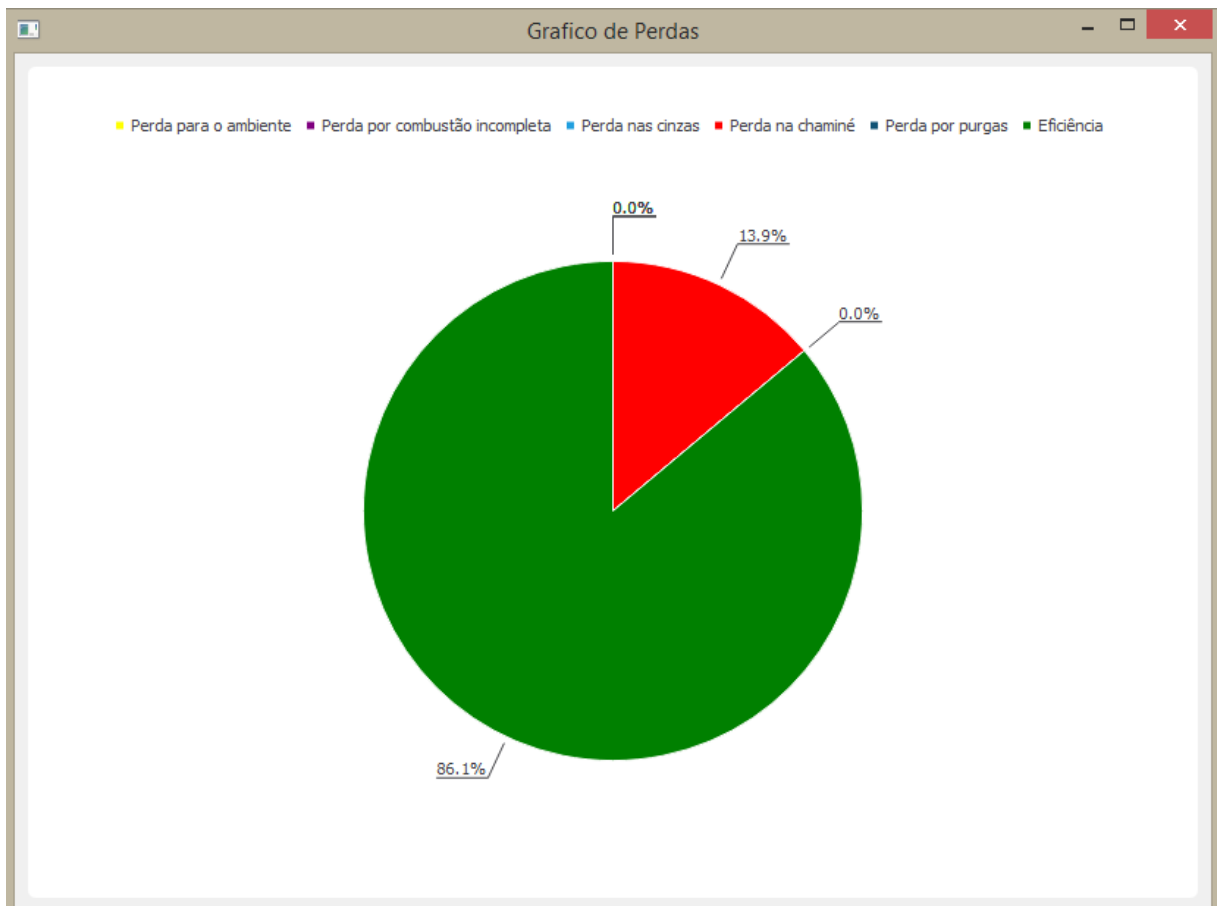


Figura 2.6 – Gráfico da eficiência no primeiro caso. Fonte: O autor

Rendimento Indireto de Caldeiras

Arquivo

Combustível		Dados		Calor gerado pelo combustível: 81342.12 kW Calor útil: 77583.75 kW PERDAS Perda combustão incompleta: 0.00 kW Perda no calor das cinzas: 0.00 kW Perda nos gases de escape: 3758.37 kW Perda por purgas: 0.00 kW Perda para o ambiente: 0.00 kW
Fração Mássica		Produção de vapor	100 [Ton/h]	
C	80 % Massa	Temperatura do Vapor	340 [°C]	
H	10 % Massa	Pressão de trabalho	4 [Mpa]	
O	5 % Massa	Coefficiente de excesso de Ar	1.08 [-]	
S	3 % Massa	Temperatura da água de alimentação	65 [°C]	
W	2 % Massa	Temperatura do ar de entrada	125 [°C]	
		Teor de cinzas do combustível	0 [%]	
		Temperatura dos gases na chaminé	240 [°C]	
		Calor específico dos gases	1.02 [KJ/KgK]	
PCI do combustível: 37161.96 kJ/kg Vazão ar/combustível estequiométrica: 12.55 Vazão mássica de combustível: 7879.88 kg/h		Perdas		
		Transferência de calor para o ambiente	0 %	
		Combustão Incompleta	0 %	
		Purgas de lama	0 %	

Figura 2.7 – Resultados do segundo caso. Fonte: O autor

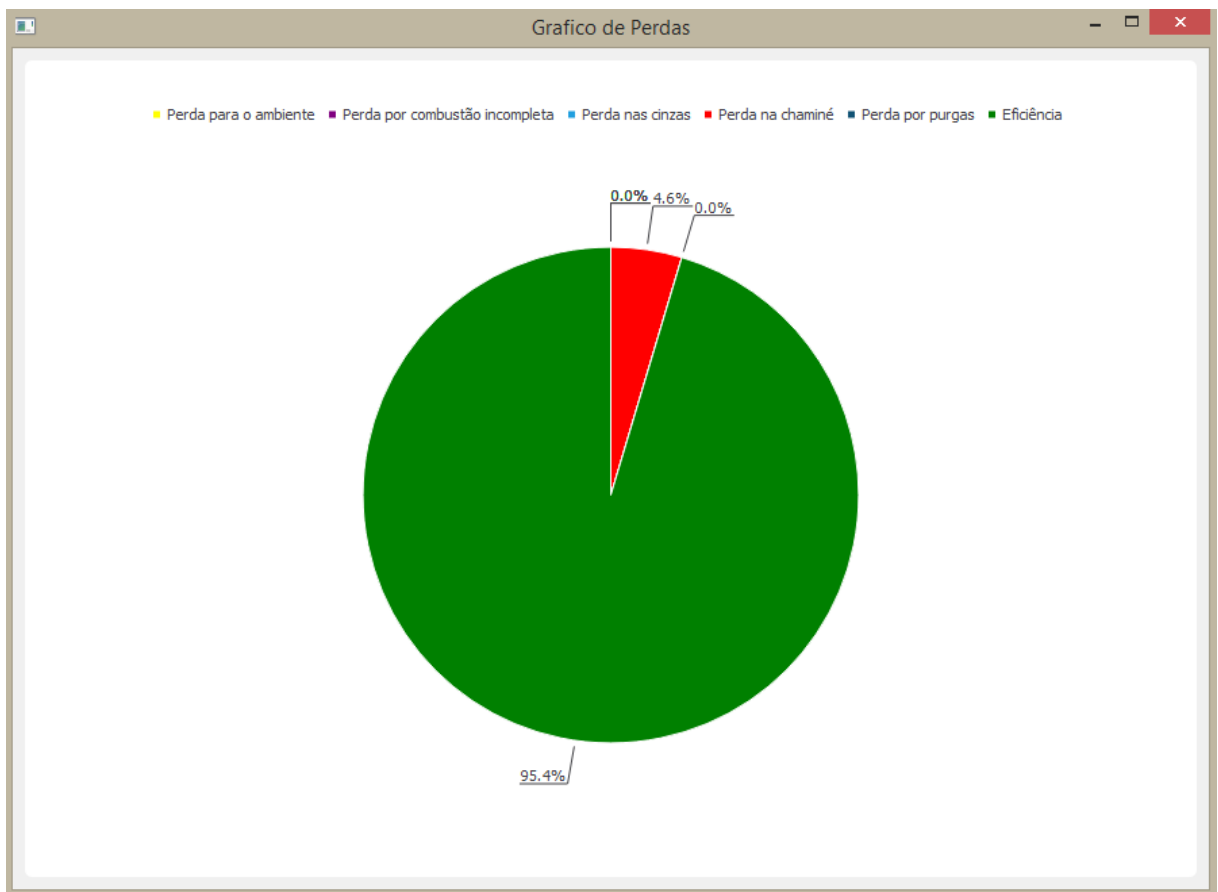


Figura 2.8 – Gráfico de eficiência do segundo caso. Fonte: O autor

Comparando o resultado do fluxo mássico de combustível na Figura 2.5 com o resultado na Figura 2.7, a economia proporcionada de combustível é:

$$\Delta\dot{m}_c = 8729,56 - 7879,88 = 849,68 \text{ [kg/h]}$$

A Tabela 2.1 mostra a comparação entre alguns resultados dos cálculos realizados de forma analítica e utilizando o programa.

Tabela 2.1 – Comparação de alguns resultados do exemplo

Dados	Analítico	Programa	Diferença
Calor útil	77667,22 [kW]	77583,75 [kW]	0,1 %
$\Delta\dot{m}_c$	864 [kg/h]	849,68 [kg/h]	1,6%
$\Delta\eta$	9,28 %	9,3 %	0,2%
PCI	37161,96[kJ/kg]	37161,96 [kJ/kg]	0%

A Tabela 2.1 mostra que não há grandes desvios nos resultados quando utilizado o programa, sendo o maior desvio na economia proporcionada de combustível.

2.6 CUMPRIMENTO DOS REQUISITOS DE PROJETO

O programa permite o cálculo da eficiência de geradores de vapor pelo método indireto com uma interface intuitiva, mostrada nas Figuras 2.5 e 2.7, e foi desenvolvido em linguagem aberta e gratuita. É possível utilizar o bagaço de cana-de-açúcar e lenha de eucalipto como combustível, onde é permitido a entrada da variação do teor de umidade e, no caso do bagaço de cana-de-açúcar, a variação o teor de açúcar no combustível. Também é possível utilizar combustíveis com a entrada em base molar ou base mássica.

2.7 TUTORIAL DE UTILIZAÇÃO DO PROGRAMA

2.7.1 Instruções para instalação do programa.

Não há necessidade de instalação do programa, para utilizar deve-se apenas executar o programa.

2.7.2 Instruções para utilização do programa

Utilizando os parâmetros inicialmente definidos, dados do combustíveis e as condições de operação que são facilmente obtidas, o programa retorna não só a quantidade de energia perdida, mas também informações relevantes, como o consumo horário de combustível e seu poder calorífico inferior. A interface do aplicativo pode ser vista na Figura 2.9. Iniciando pelo quadro I, é possível escolher o tipo de combustível utilizado entre lenha de eucalipto, diesel, bagaço de cana e também utilizando sua fórmula ou base mássica. Caso o combustível escolhido seja o diesel, não é possível preencher sua composição, para lenha de eucalipto deve-se indicar apenas a umidade, como mostra o exemplo da Figura 2.10. No quadro II deve-se informar as condições de operação, requisitos e temperaturas analisadas. Em III informa-se as perdas que geralmente já são conhecidas e não há muita variação. Com todos os dados inseridos, após clicar em calcular eficiência, a primeira lista (V) apresenta o PCI do combustível utilizado, a relação ar/ combustível estequiométrica e o fluxo de combustível necessário. Na segunda lista (VI) são mostradas as perdas na produção e a energia total. No quadro IV também há os botões para limpar os dados inseridos e um botão de exemplo, que insere alguns dados arbitrários para demonstrar o uso do programa.

The screenshot shows the main window of the 'Rendimento Caldeira' application. The window title is 'Rendimento Caldeira' and it has a menu bar with 'Arquivo'. The interface is divided into several sections:

- I (Fuel Composition):** A section for entering fuel properties. It includes a dropdown for 'Base Molar' and input fields for C, H, O, S, and W, each followed by '% Massa'.
- II (Operational Parameters):** A section for entering operating conditions. It includes input fields for 'Produção Necessária' (Ton/h), 'Temperatura do Vapor' (°C), 'Pressão de trabalho' (Mpa), 'Excesso de Ar' (-), 'Temperatura de alimentação da água' (°C), 'Temperatura de alimentação do ar' (°C), 'Teor de cinzas do combustível' (%), 'Temperatura dos gases na chaminé' (°C), and 'Calor específico dos gases' (KJ/kgK).
- III (Losses):** A section for entering losses. It includes input fields for 'Perdas Ambiente' (%), 'Combustão Mecânica Incompleta' (%), and 'Vazão de Purgas' (%).
- IV (Buttons):** A section with two buttons: 'Calcular Eficiência' and 'Novos dados'.
- V (Output Results 1):** A section for displaying results, including 'PCI', 'Relação ar/combustível', and 'Fluxo de combustível'.
- VI (Output Results 2):** A section for displaying results, including 'Perdas' and 'Energia total'.

Figura 2.9 – Janela principal do aplicativo. Fonte: O autor

Combustível

Lenha de eucalipto

C % Massa

H % Massa

O % Massa

S % Massa

W % Massa

Figura 2.10 – Exemplo da mudança de combustível. Fonte: O autor

2.7.3 Exemplo de aplicação

Para realizar os cálculos utilizando o programa serão utilizados os seguintes pré requisitos adaptados do exemplo de Strobel:

Produção de vapor necessária: [15 ton/h];
 Condição do vapor: 8 [Mpa] e 520 [°C];
 Combustível: C = 50%; H=6%; S=5%; O=8%; umidade=9%; cinzas=22% (base mássica)
 Coeficiente de excesso de ar: 1,4;
 Coeficiente de excesso de ar: 1,4;
 Temperatura de alimentação da água: 50 [°C];
 Temperatura de alimentação do ar: 60 [°C];
 Temperatura de dos gases de combustão: 400 [°C];
 Calor específico dos gases de combustão: 1,6 [kJ/kgK];
 Perda de calor por combustão mecânica incompleta: 4 [%];
 Perda de calor para o ambiente: 5 [%]

Com estes dados pode-se calcular a eficiência da geração de vapor com o aplicativo.

A partir dos dados inseridos, a Figura 2.11 mostra os resultados obtidos. À esquerda é mostrado os dados relacionados ao combustível, à direita estão as perdas de energia na geração de vapor. Para visualização da eficiência e perdas de forma percentual, há um gráfico, como na Figura 2.12.

Rendimento Indireto de Caldeiras

Arquivo

Combustível		Dados		Resultados	
Fração Mássica		Produção de vapor	15 [Ton/h]	Calor gerado pelo combustível: 21872.61 kW	
C	50 % Massa	Temperatura do Vapor	520 [°C]	Calor útil: 13498.18 kW	
H	6 % Massa	Pressão de trabalho	8 [Mpa]	PERDAS	
O	8 % Massa	Coefficiente de excesso de Ar	1.4 [-]	Perda combustão incompleta: 874.90 kW	
S	5 % Massa	Temperatura da água de alimentação	50 [°C]	Perda no calor das cinzas: 249.30 kW	
W	9 % Massa	Temperatura do ar de entrada	60 [°C]	Perda nos gases de escape: 6156.60 kW	
PCI do combustível: 22583.57 kJ/kg		Teor de cinzas do combustível	22 [%]	Perda por purgas: 0.00 kW	
Vazão ar/combustível estequiométrica: 7.68		Temperatura dos gases na chaminé	400 [°C]	Perda para o ambiente: 1093.63 kW	
Vazão mássica de combustível: 3486.67 kg/h		Calor específico dos gases	1.6 [kJ/kgK]		
		Perdas			
		Transferência de calor para o ambiente	5 %		
		Combustão Incompleta	4 %		
		Purgas de lama	0 %		
		Calcular Eficiência			
Exemplo		Novos dados			

Figura 2.11 – Resultados. Fonte: O autor

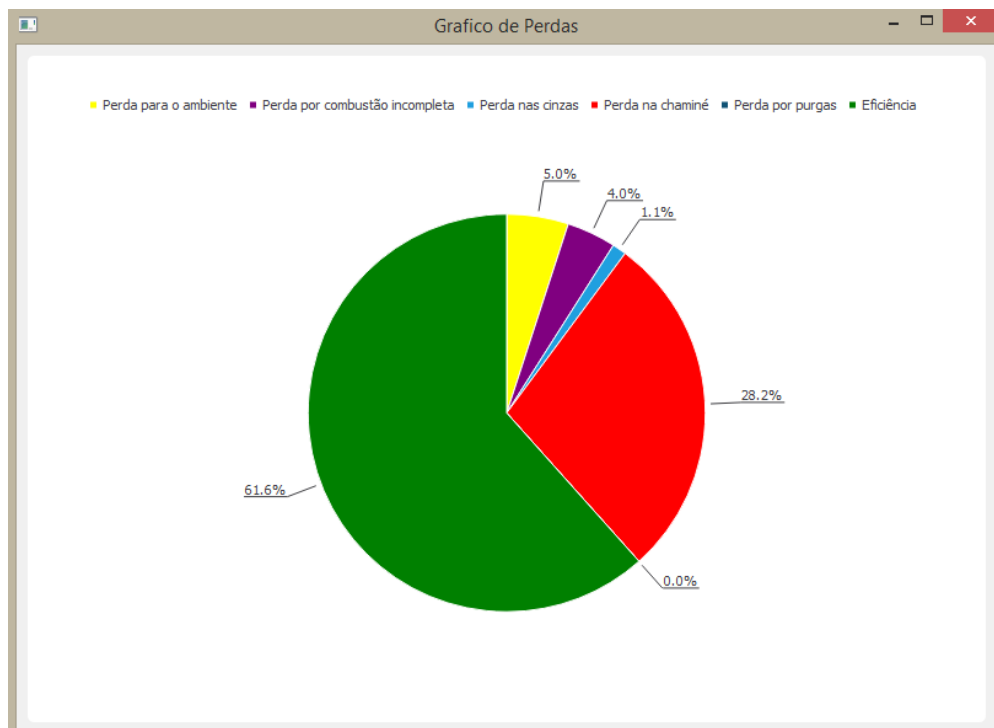


Figura 2.12 – Gráfico da eficiência. Fonte: O autor

Mudança no combustível

Mantendo os mesmos dados de entrada utilizados no Exemplo 2.6.3 e alterando apenas o combustível, foram obtidos os dados presentes na Tabela 2.2, com o PCI em [kJ/kg].

Tabela 2.2 – Variação da eficiência conforme combustível

Combustível	PCI	AC	Eficiência
Cana-de-açúcar	12507	6,06	48%
Lenha	13312	5,41	54,3%
Diesel	43000	14,5	63,5%
Exemplo	22583	7,68	61,7%

Coeficiente de excesso de ar

Com uma variação do excesso de ar entre 10 e 50%, foi comparada a variação da eficiência para cada combustível, mostrada no gráfico da Figura 2.13

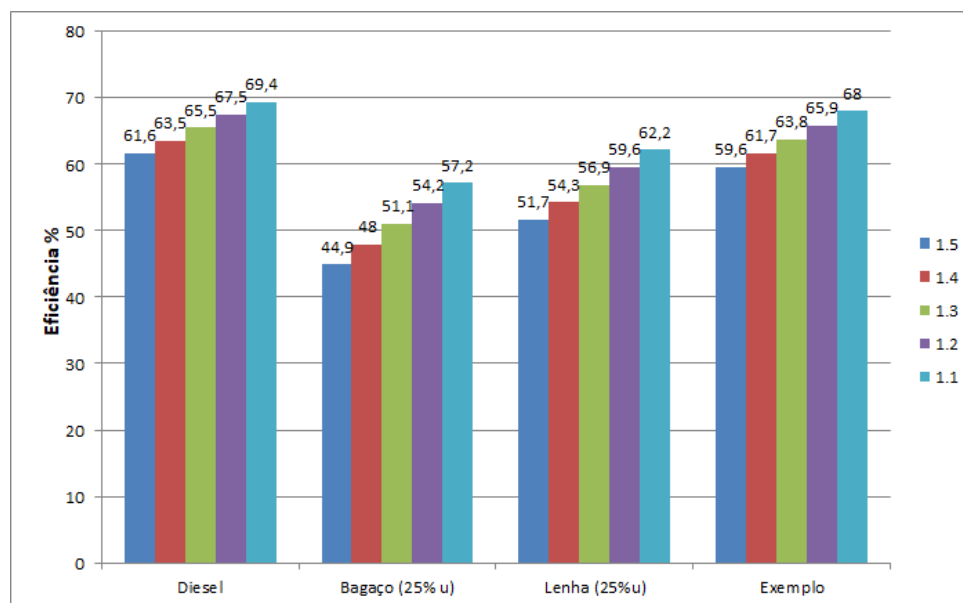


Figura 2.13 – Variação da eficiência conforme coeficiente de excesso de ar

Temperatura dos gases

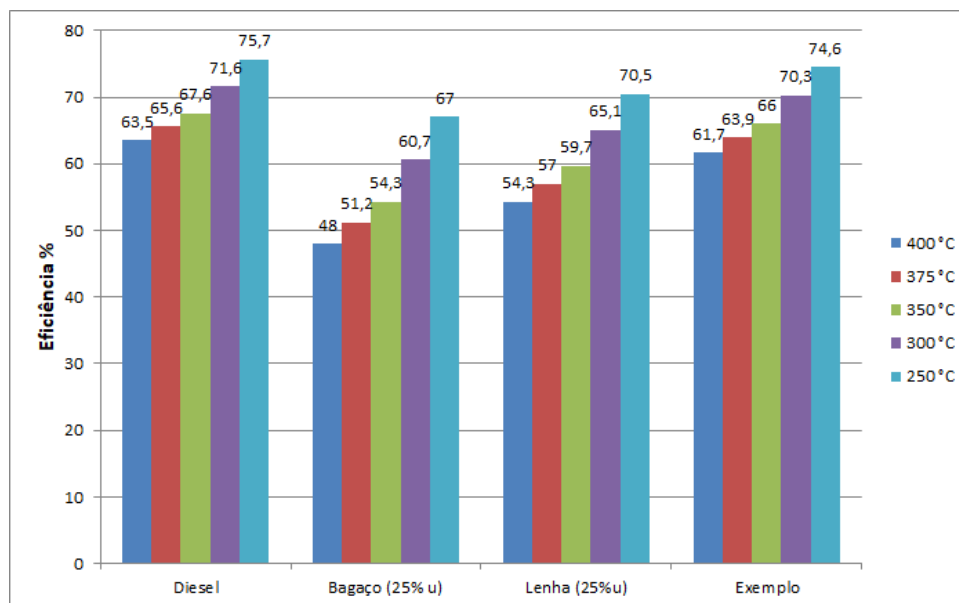


Figura 2.14 – Variação da eficiência conforme a temperatura dos gases de escape

2.8 CONCLUSÃO

A partir dos dados analisados, observa que o aplicativo cumpre o objetivo de demonstrar a eficiência da geração de vapor, sua interface intuitiva garante ampla utilização também por estudantes no intuito de compreender de forma gráfica onde se encontram as maiores perdas no sistema e como alterações nos parâmetros afetam a eficiência térmica da caldeira. Com a utilização do programa, também pode se observar que:

- Combustíveis com PCI mais elevados demonstram melhor eficiência na geração de vapor, mas como pode ser observado na Tabela 2.2, apesar do Diesel ter um PCI consideravelmente mais elevado que o combustível utilizado no exemplo, sua relação ar combustível também é mais elevada, causando pouca variação de eficiência. Portanto, para a escolha do combustível outros fatores devem ser considerados, não apenas seu poder calorífico.
- O fator que tem maior impacto na eficiência da caldeira é a temperatura de saída dos gases de escape, a eficiência é mais sensível à mudança dessa temperatura. É possível reaproveitar o calor dos gases de escape utilizando um recuperador, portanto, também foi analisada a variação da eficiência conforme a temperatura de saída dos gases de combustão.

Sugere-se para novos trabalhos, então, o estudo do impacto da aplicação de formas de recuperar calor na geração de vapor, e também melhorias no desenvolvimento de programas, como em plataformas móveis e mecanismos de controle simultâneo para uso em aplicações práticas.

3. BIBLIOGRAFIA

- CARVALHO, A. H. S. et al. **Influência da umidade do bagaço da cana-de-açúcar no sistema de cogeração de energia**. FATEC - Sertãozinho: [s.n.], 2019. P. 12.
- ÇENGEL, Y. A.; BOLES, M. A. **Termodinâmica**. [S.l.]: McGraw-Hill, 2013. P. 1048. ISBN 8580552001.
- LUTZ, MARK; ASCHER, DAVID. **Aprendendo Python**. Porto Alegre: Bookman, 2007. 568 P.
- NEIVA, P. S.; FURTADO, D. B.; FINZER, J. R. D. **Capacidade térmica e poder calorífico de biomassa eucalipto**. Universidade de Uberaba, Departamento de engenharia química: [s.n.], 2018.
- NOGUEIRA, L. A. H. **Eficiência Energética no Uso de Vapor**. Rio de Janeiro: Eletrobrás/Procel, 2005. P. 196.
- NOGUEIRA, L. A. H.; ROCHA, C. R.; NOGUEIRA, F. J. H. **Eficiência Energética no Uso de Vapor**. Rio de Janeiro: Eletrobrás/Procel, 2005. P. 95.
- SILVA, L. R. S.; FINZER, J. R. D.; BEGNINI, M. L. **Cálculo da relação ar/combustível e a eficiência em caldeira alimentada com bagaço de cana-de-açúcar**. Universidade de Uberaba, Departamento de engenharia química: [s.n.], 2018.
- SILVA, R. E. P. **TROCADORES DE CALOR, GERAÇÃO E DISTRIBUIÇÃO DE VAPOR**. Engenharia de Energia, UFGD: [s.n.], 2016.
- STROBEL, C. **Balço energético em caldeiras**. Departamento de engenharia mecânica, UFPR: [s.n.]. Disponível em: <http://ftp.demec.ufpr.br/disciplinas/EngMec_NOTURNO/>

TM364/Material%5C%20de%5C%20Aula/Balan%5C%E7o%5C%20energ%5C%E9ti
co%5C%20em%5C%20caldeiras.pdf>.

WOODRUFF, E. B.; LAMMERS, H. B.; LAMMERS, T. F. **Steam Plant Operation**. [S.l.]:
McGraw-Hill Professional, 2004. P. 850. ISBN 0071418466.

**[https://www.slashdata.co/blog/state-of-the-developer-nation-coding-language-
popularity-chinas-developer-market-how-developers-make-revenue-and-more](https://www.slashdata.co/blog/state-of-the-developer-nation-coding-language-popularity-chinas-developer-market-how-developers-make-revenue-and-more)**:
acessado em: 18/04/2023

<https://pythoninstitute.org/about-python>: acessado em 18/04/2023

**[https://www.icaterm.com.br/quais-sao-os-tipos-de-combustiveis-utilizados-em-
caldeiras](https://www.icaterm.com.br/quais-sao-os-tipos-de-combustiveis-utilizados-em-caldeiras)**: acessado em 18/04/2023

APÊNDICE

A. CÓDIGO DO PROGRAMA EM PYTHON

```

from PyQt5 import QtCore, QtGui, QtWidgets, QtChart
from iapws import IAPWS97

cp_ar = 1.022
cp_agua = 4.180
const_gas = 8.31
perda_costado = 0.04 # 4%
pci_diesel = 43000 # KJ/kg
arcomb_diesel = 14.5

p_atm = 101.325 / 1000

# PCI da lenha conforme umidade (em kcal/kg)

umidade = list(range(0, 86))
f = 1051.43 / 5
pci = [18569.84 - i * f for i in umidade]
pci_lenha = {}
for i in umidade:
    for j in pci:
        pci_lenha[i] = j
        pci.remove(j)
        break

# CALOR UTIL
def calor_util(mv, hv, ha, ):
    calor = mv * (hv - ha)
    return calor

# CALOR PERDIDO NOS GASES DE ESCAPE
def perda_gases(mar, mcomb, tref, tgas):
    perda = (mar + mcomb) * cp_ar * (tgas - tref)
    return perda

# Perdas associadas à temperatura das cinzas
def perda_tempcinza(mr):
    perda = mr * 1170
    return perda

# Perdas associadas ao combustível não convertido presente nas cinzas
def perda_carbono(mc):
    perda = mc * 33780

```

```

return perda

# Perdas por purgas
def perda_purga(mp, tp, tref):
    perda = mp * cp_agua * (tp - tref)
    return perda

# PCI para combustiveis solidos e liquidos
def pci_ls(c, h, s, o, w):
    pci = 339.13 * c + 1029.95 * h + 108.85 * (s - o) - 25.12 * w
    return pci

# PCI para combustiveis gasosos
def pci_g(h, co, h2s, ch4, c2h4, c2h6, c3h6, c3h8, c4h8, c4h10, c5h12, c6h6):
    pci = float(
        108 * h + 126 * co + 234 * h2s + 358 * ch4 + 591 * c2h4 + 638 * c2h6 + 860 * c3h6
+ 913 * c3h8 + 1135 * c4h8
        + 1187 * c4h10 + 1461 * c5h12 + 1403 * c6h6)
    return pci

# relação ar combustível
def ac(c, h, s, o):
    rel = 0.1149 * c + 0.3448 * h + 0.0431 * (s - o)
    return rel

def eficiencia(k, ea, ac, cp, t, tref, pci):
    efi = 1 - k - ((ea * (ac + 1) * cp * (t - tref)) / pci)
    return efi

def consumo(util, pci, efi):
    mcomb = util / (efi * pci)
    return mcomb

def pchamine(ea, ac, cp, t, tref):
    pch = ea * (ac + 1) * cp * (t - tref)
    return pch

def p_umidade(mv, cpv, tref, tch):
    perda = mv * (4.186 * (100 - tref) + cpv * (tch - 100))
    return perda

```



```

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1000, 600)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.widget = QtWidgets.QWidget(self.centralwidget)
        self.widget.setGeometry(QtCore.QRect(20, 30, 958, 495))
        self.gridLayout_4 = QtWidgets.QGridLayout(self.widget)
        self.gridLayout_4.setContentsMargins(0, 0, 0, 0)
        self.groupBox_comb = QtWidgets.QGroupBox(self.widget)
        self.gridLayout_3 = QtWidgets.QGridLayout(self.groupBox_comb)
        self.comboBox = QtWidgets.QComboBox(self.groupBox_comb)
        self.comboBox.addItem("Fração Mássica")
        self.comboBox.addItem("Bagaço de cana")
        self.comboBox.addItem("Diesel")
        self.comboBox.addItem("Lenha de eucalipto")
        self.comboBox.addItem("Fórmula molar")
        self.gridLayout_3.addWidget(self.comboBox, 0, 1, 1, 1)
        self.label_7 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_7, 1, 0, 1, 1)
        self.lineEdit_carbono = QtWidgets.QLineEdit(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.lineEdit_carbono, 1, 1, 1, 1)
        self.label_28 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_28, 1, 2, 1, 1)
        self.label_8 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_8, 2, 0, 1, 1)
        self.lineEdit_hidrogenio = QtWidgets.QLineEdit(self.groupBox_comb)
        self.lineEdit_hidrogenio.setValidator(QtGui.QIntValidator())
        self.gridLayout_3.addWidget(self.lineEdit_hidrogenio, 2, 1, 1, 1)
        self.label_29 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_29, 2, 2, 1, 1)
        self.label_9 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_9, 3, 0, 1, 1)
        self.lineEdit_oxigenio = QtWidgets.QLineEdit(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.lineEdit_oxigenio, 3, 1, 1, 1)
        self.label_10 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_10, 5, 0, 1, 1)
        self.lineEdit_enxofre = QtWidgets.QLineEdit(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.lineEdit_enxofre, 5, 1, 1, 1)
        self.label_30 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_30, 3, 2, 1, 1)
        self.label_31 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_31, 5, 2, 1, 1)
        self.lineEdit_wet = QtWidgets.QLineEdit(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.lineEdit_wet, 6, 1, 1, 1)
        self.label_13 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_13, 6, 0, 1, 1)
        self.label_32 = QtWidgets.QLabel(self.groupBox_comb)
        self.gridLayout_3.addWidget(self.label_32, 6, 2, 1, 1)

```

```

self.gridLayout_4.addWidget(self.groupBox_comb, 0, 0, 1, 1)
self.groupBox_dados = QtWidgets.QGroupBox(self.widget)
self.gridLayout_2 = QtWidgets.QGridLayout(self.groupBox_dados)
self.label_3 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_3, 4, 0, 1, 1)
self.label_24 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_24, 4, 2, 1, 1)
self.lineEdit_requisito = QtWidgets.QLineEdit(self.groupBox_dados)
self.gridLayout_2.addWidget(self.lineEdit_requisito, 0, 1, 1, 1)
self.lineEdit_Tvapor = QtWidgets.QLineEdit(self.groupBox_dados)
self.gridLayout_2.addWidget(self.lineEdit_Tvapor, 1, 1, 1, 1)
self.lineEdit_pressao = QtWidgets.QLineEdit(self.groupBox_dados)
self.gridLayout_2.addWidget(self.lineEdit_pressao, 2, 1, 1, 1)
self.lineEdit_excessoAr = QtWidgets.QLineEdit(self.groupBox_dados)
self.gridLayout_2.addWidget(self.lineEdit_excessoAr, 3, 1, 1, 1)
self.lineEdit_Tagua = QtWidgets.QLineEdit(self.groupBox_dados)
self.gridLayout_2.addWidget(self.lineEdit_Tagua, 4, 1, 1, 1)
self.lineEdit_Tar = QtWidgets.QLineEdit(self.groupBox_dados)
self.gridLayout_2.addWidget(self.lineEdit_Tar, 5, 1, 1, 1)
self.label_2 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_2, 5, 0, 1, 1)
self.label_21 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_21, 1, 2, 1, 1)
self.label_5 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_5, 2, 0, 1, 1)
self.label = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label, 3, 0, 1, 1)
self.label_22 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_22, 2, 2, 1, 1)
self.label_23 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_23, 3, 2, 1, 1)
self.lineEdit_Tcinzas = QtWidgets.QLineEdit(self.groupBox_dados)
self.gridLayout_2.addWidget(self.lineEdit_Tcinzas, 6, 1, 1, 1)
self.label_26 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_26, 6, 2, 1, 1)
self.label_12 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_12, 7, 0, 1, 1)
self.label_25 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_25, 5, 2, 1, 1)
self.lineEdit_tchamine = QtWidgets.QLineEdit(self.groupBox_dados)
self.gridLayout_2.addWidget(self.lineEdit_tchamine, 7, 1, 1, 1)
self.lineEdit_cpgas = QtWidgets.QLineEdit(self.groupBox_dados)
self.gridLayout_2.addWidget(self.lineEdit_cpgas, 8, 1, 1, 1)
self.label_16 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_16, 6, 0, 1, 1)
self.label_27 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_27, 7, 2, 1, 1)
self.label_4 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_4, 1, 0, 1, 1)
self.label_20 = QtWidgets.QLabel(self.groupBox_dados)

```

```

self.gridLayout_2.addWidget(self.label_20, 0, 2, 1, 1)
self.label_req = QtWidgets.QLabel(self.groupBox_dados)
self.label_req.setText("Produção de vapor")
self.gridLayout_2.addWidget(self.label_req, 0, 0, 1, 1)
self.gridLayout_4.addWidget(self.groupBox_dados, 0, 1, 1, 1)
self.listWidget_perdas = QtWidgets.QListWidget(self.widget)
self.gridLayout_4.addWidget(self.listWidget_perdas, 0, 2, 1, 1)
for i in range(11):
    item = QtWidgets.QListWidgetItem()
    self.listWidget_perdas.addItem(item)
self.listWidget_dados = QtWidgets.QListWidget(self.widget)
item = QtWidgets.QListWidgetItem()
self.listWidget_dados.addItem(item)
item = QtWidgets.QListWidgetItem()
self.listWidget_dados.addItem(item)
item = QtWidgets.QListWidgetItem()
self.listWidget_dados.addItem(item)
item = QtWidgets.QListWidgetItem()
self.listWidget_dados.addItem(item)
self.gridLayout_4.addWidget(self.listWidget_dados, 1, 0, 2, 1)
self.groupBox_perdas = QtWidgets.QGroupBox(self.widget)
self.gridLayout = QtWidgets.QGridLayout(self.groupBox_perdas)
self.label_15 = QtWidgets.QLabel(self.groupBox_perdas)
self.gridLayout.addWidget(self.label_15, 1, 0, 1, 1)
self.lineEdit_Tamb = QtWidgets.QLineEdit(self.groupBox_perdas)
self.gridLayout.addWidget(self.lineEdit_Tamb, 0, 1, 1, 1)
self.label_18 = QtWidgets.QLabel(self.groupBox_perdas)
self.gridLayout.addWidget(self.label_18, 1, 2, 1, 1)
self.label_11 = QtWidgets.QLabel(self.groupBox_perdas)
self.gridLayout.addWidget(self.label_11, 0, 2, 1, 1)
self.lineEdit_combMecInc = QtWidgets.QLineEdit(self.groupBox_perdas)
self.gridLayout.addWidget(self.lineEdit_combMecInc, 1, 1, 1, 1)
self.lineEdit_Vpurga = QtWidgets.QLineEdit(self.groupBox_perdas)
self.gridLayout.addWidget(self.lineEdit_Vpurga, 2, 1, 1, 1)
self.label_undPurga = QtWidgets.QLabel(self.groupBox_perdas)
self.label_tituloPurga = QtWidgets.QLabel(self.groupBox_perdas)
self.gridLayout.addWidget(self.label_tituloPurga, 2, 0, 1, 1)
self.gridLayout.addWidget(self.label_undPurga, 2, 2, 1, 1)
self.label_undPurga.setText("%")
self.label_tituloPurga.setText("Purgas de lama")
self.lineEdit_Vpurga.setText("0")
self.label_14 = QtWidgets.QLabel(self.groupBox_perdas)
self.gridLayout.addWidget(self.label_14, 0, 0, 1, 1)
self.gridLayout_4.addWidget(self.groupBox_perdas, 1, 1, 1, 1)
self.pushButton = QtWidgets.QPushButton(self.widget)
self.gridLayout_4.addWidget(self.pushButton, 2, 1, 1, 1)
self.pushButton_2 = QtWidgets.QPushButton(self.widget)
self.gridLayout_4.addWidget(self.pushButton_2, 3, 1, 1, 1)
self.pushButton_2.setText("Novos dados")
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 940, 21))

```

```

self.menuArquivo = QtWidgets.QMenu(self.menubar)
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
MainWindow.setStatusBar(self.statusbar)
self.actionSair = QtWidgets.QAction(MainWindow)
self.menuArquivo.addAction(self.actionSair)
self.menubar.addAction(self.menuArquivo.menuAction())
self.label_cp = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_cp, 8, 0, 1, 1)
self.label_cp.setText("Calor específico dos gases")
self.label_cp2 = QtWidgets.QLabel(self.groupBox_dados)
self.gridLayout_2.addWidget(self.label_cp2, 8, 2, 1, 1)
self.label_cp2.setText("[KJ/KgK]")
self.pushButton_exemplo = QtWidgets.QPushButton(self.widget)
self.gridLayout_4.addWidget(self.pushButton_exemplo, 3, 0)
self.pushButton_exemplo.setText("Exemplo")

self.retranslateUi(MainWindow)

# validator
ListaLineEdit = [self.lineEdit_carbono, self.lineEdit_hidrogenio,
self.lineEdit_oxigenio, self.lineEdit_enxofre,
self.lineEdit_wet, self.lineEdit_requisito, self.lineEdit_Tvapor,
self.lineEdit_pressao,
self.lineEdit_excessoAr, self.lineEdit_Tagua, self.lineEdit_Tar,
self.lineEdit_Tcinzas,
self.lineEdit_tchamine, self.lineEdit_cpgas, self.lineEdit_Tamb,
self.lineEdit_combMecInc]
for item in ListaLineEdit:
    item.setValidator(QtGui.QDoubleValidator())

self.comboBox.activated.connect(self.combuschanged)
self.pushButton.clicked.connect(self.eficiencia)
self.pushButton_2.clicked.connect(self.limpar)
self.pushButton_exemplo.clicked.connect(self.exemplo)
self.actionSair.triggered.connect(MainWindow.close) # type: ignore
QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "Eficiencia Indireta de
Caldeiras"))
    self.groupBox_comb.setTitle(_translate("MainWindow", "Combustível"))
    self.label_7.setToolTip(_translate("MainWindow",
"<html><head/><body><p>Carbano</p></body></html>"))
    self.label_7.setText(_translate("MainWindow", "C"))
    self.label_28.setText(_translate("MainWindow", "% Massa"))
    self.label_8.setToolTip(_translate("MainWindow",
"<html><head/><body><p>Hidrogênio</p></body></html>"))
    self.label_8.setText(_translate("MainWindow", "H"))

```

```

        self.label_29.setText(_translate("MainWindow", "% Massa"))
        self.label_9.setToolTip(_translate("MainWindow",
"<html><head/><body><p>Oxigênio</p></body></html>"))
        self.label_9.setText(_translate("MainWindow", "O"))
        self.label_10.setToolTip(_translate("MainWindow",
"<html><head/><body><p>Enxofre</p></body></html>"))
        self.label_10.setText(_translate("MainWindow", "S"))
        self.label_30.setText(_translate("MainWindow", "% Massa"))
        self.label_31.setText(_translate("MainWindow", "% Massa"))
        self.label_13.setToolTip(_translate("MainWindow",
"<html><head/><body><p>Umidade</p></body></html>"))
        self.label_13.setText(_translate("MainWindow", "W"))
        self.label_32.setText(_translate("MainWindow", "% Massa"))
        self.groupBox_dados.setTitle(_translate("MainWindow", "Dados"))
        self.label_3.setText(_translate("MainWindow", "Temperatura da água de
alimentação"))
        self.label_24.setText(_translate("MainWindow", "[°C]"))
        self.label_2.setText(_translate("MainWindow", "Temperatura do ar de entrada"))
        self.label_21.setText(_translate("MainWindow", "[°C]"))
        self.label_5.setText(_translate("MainWindow", "Pressão de trabalho"))
        self.label.setText(_translate("MainWindow", "Coeficiente de excesso de Ar"))
        self.label_22.setText(_translate("MainWindow", "[Mpa]"))
        self.label_23.setText(_translate("MainWindow", "[-]"))
        self.label_26.setText(_translate("MainWindow", "[%]"))
        self.label_12.setText(_translate("MainWindow", "Temperatura dos gases na
chaminé"))
        self.label_25.setText(_translate("MainWindow", "[°C]"))
        self.label_16.setText(_translate("MainWindow", "Teor de cinzas do combustível"))
        self.label_27.setText(_translate("MainWindow", "[°C]"))
        self.label_4.setText(_translate("MainWindow", "Temperatura do Vapor"))
        self.label_20.setText(_translate("MainWindow", "[Ton/h]"))
        __sortingEnabled = self.listWidget_dados.isSortingEnabled()
        self.listWidget_dados.setSortingEnabled(False)
        item = self.listWidget_dados.item(0)
        item.setText(_translate("MainWindow", "PCI"))
        item = self.listWidget_dados.item(1)
        item.setText(_translate("MainWindow", "Vazão de ar/combustível
estequiométrica"))
        item = self.listWidget_dados.item(2)
        item.setText(_translate("MainWindow", "Vazão mássica de combustível"))
        self.listWidget_dados.setSortingEnabled(__sortingEnabled)
        self.groupBox_perdas.setTitle(_translate("MainWindow", "Perdas"))
        self.label_15.setText(_translate("MainWindow", "Combustão Incompleta"))
        self.lineEdit_Tamb.setText(_translate("MainWindow", "5"))
        self.label_18.setText(_translate("MainWindow", "%"))
        self.label_11.setText(_translate("MainWindow", "%"))
        self.lineEdit_combMecInc.setText(_translate("MainWindow", "4"))
        self.label_14.setText(_translate("MainWindow", "Transferência de calor para o
ambiente"))
        self.pushButton.setText(_translate("MainWindow", "Calcular Eficiência"))

```

```

self.menuArquivo.setTitle(_translate("MainWindow", "Arquivo"))
self.actionSair.setText(_translate("MainWindow", "Sair"))

def combuschanged(self, indice):
    if not indice == 1:
        self.label_10.show()
        self.label_31.setText("% Massa")
    else:
        self.label_10.hide()
        self.label_31.setText("Açúcar")
    if indice == 0 or indice == 4:
        self.lineEdit_carbono.setEnabled(True)
        self.lineEdit_wet.setEnabled(True)
        self.lineEdit_enxofre.setEnabled(True)
        self.lineEdit_oxigenio.setEnabled(True)
        self.lineEdit_hidrogenio.setEnabled(True)
    else:
        self.lineEdit_enxofre.setEnabled(False)
        self.lineEdit_carbono.setEnabled(False)
        self.lineEdit_wet.setEnabled(False)
        self.lineEdit_oxigenio.setEnabled(False)
        self.lineEdit_hidrogenio.setEnabled(False)
    if indice == 1 or indice == 3:
        self.lineEdit_wet.setEnabled(True)
    if indice == 1:
        self.lineEdit_enxofre.setEnabled(True)
    if indice == 4:
        self.label_28.hide()
        self.label_29.hide()
        self.label_30.hide()
        self.label_31.hide()
    else:
        self.label_28.show()
        self.label_29.show()
        self.label_30.show()
        self.label_31.show()
        self.label_32.show()

def limpar(self):
    ListaLineEdit = [self.lineEdit_carbono, self.lineEdit_hidrogenio,
self.lineEdit_oxigenio, self.lineEdit_enxofre,
self.lineEdit_wet, self.lineEdit_requisito, self.lineEdit_Tvapor,
self.lineEdit_pressao,
self.lineEdit_excessoAr, self.lineEdit_Tagua, self.lineEdit_Tar,
self.lineEdit_Tcinzas,
self.lineEdit_tchamine, self.lineEdit_cpgas, self.lineEdit_Vpurga]
    for item in ListaLineEdit:
        item.setText("0")
    for i in range(11):
        self.listWidget_perdas.item(i).setText("")

```

```

for i in range(3):
    self.listWidget_dados.item(i).setText("")

def exemplo(self):
    ListaLineEdit = [self.lineEdit_carbono, self.lineEdit_hidrogenio,
self.lineEdit_oxigenio, self.lineEdit_enxofre,
self.lineEdit_wet, self.lineEdit_requisito, self.lineEdit_Tvapor,
self.lineEdit_pressao,
self.lineEdit_excessoAr, self.lineEdit_Tagua, self.lineEdit_Tar,
self.lineEdit_Tcinzas,
self.lineEdit_tchamine, self.lineEdit_cpogas, self.lineEdit_Vpurga]
    ValorExemplo = [50, 6, 8, 5, 9, 15, 520, 8, 1.4, 50, 60, 22, 400, 1.6, 0]

    self.comboBox.setCurrentIndex(0)

    for item in ListaLineEdit:
        for j in ValorExemplo:
            item.setText(f'{j}')
            ValorExemplo.remove(j)
            break
    self.lineEdit_Tamb.setText("5")
    self.lineEdit_combMecInc.setText("4")
    self.eficiencia()

def eficiencia(self):
    try:
        if self.comboBox.currentIndex() == 0:
            c = float(self.lineEdit_carbono.text())
            h = float(self.lineEdit_hidrogenio.text())
            o = float(self.lineEdit_oxigenio.text())
            s = float(self.lineEdit_enxofre.text())
            w = float(self.lineEdit_wet.text())
            pci = pci_ls(c, h, s, o, w)
            arcomb = ac(c, h, s, o)
        elif self.comboBox.currentIndex() == 2:
            pci = pci_diesel
            arcomb = arcomb_diesel
        elif self.comboBox.currentIndex() == 3:
            w = float(self.lineEdit_wet.text())
            pci = pci_lenha[w]
            arcomb = 5.41
        elif self.comboBox.currentIndex() == 4:
            c = float(self.lineEdit_carbono.text())
            h = float(self.lineEdit_hidrogenio.text())
            o = float(self.lineEdit_oxigenio.text())
            s = float(self.lineEdit_enxofre.text())
            w = float(self.lineEdit_wet.text())
            w = w / 100
            coef = 1 - w
            m = c * 12 + h + o * 16 + s * 32

```

```

c = ((c * 12) / m) * 100 * coef
h = (h / m) * 100 * coef
o = ((o * 16) / m) * 100 * coef
s = ((s * 32) / m) * 100 * coef
pci = pci_ls(c, h, s, o, w)
arcomb = ac(c, h, s, o)
else:
w = float(self.lineEdit_wet.text())
if self.lineEdit_enxofre.text() == "":
a = 4
else:
a = float(self.lineEdit_enxofre.text())

pci = 17782 - 50.256 * a - 202.924 * w
arcomb = 6.06
pressao = float(self.lineEdit_pressao.text())
Tagua = float(self.lineEdit_Tagua.text())
Tar = float(self.lineEdit_Tar.text())
Tcinza = float(self.lineEdit_Tcinzas.text())
ExcessoAr = float(self.lineEdit_excessoAr.text())
Tchamine = float(self.lineEdit_tchamine.text())
cpgas = float(self.lineEdit_cp_gas.text())
perdaAmbiente = float(self.lineEdit_Tamb.text())
perdaCombustaoIncompleta = float(self.lineEdit_combMecInc.text())
perdaPurgas = float(self.lineEdit_Vpurga.text()) / 100
Tsat = IAPWS97(P=pressao, x=1)
Tsat = Tsat.T
perdaPurgas = perdaPurgas * cp_agua * (Tsat - Tar)
perdacinza = perda_tempcinza(Tcinza / 100)
k = perdaAmbiente + perdaCombustaoIncompleta
k = k / 100
Tvapor = float(self.lineEdit_Tvapor.text())
requisito = float(self.lineEdit_requisito.text())
Entalpia Vapor = IAPWS97(T=Tvapor + 273, P=pressao)
Entalpia Vapor = Entalpia Vapor.h
Entalpia Agua = IAPWS97(T=Tagua + 273, P=pressao)
Entalpia Agua = Entalpia Agua.h
CalorUtil = calor_util(requisito * 1000/3600, Entalpia Vapor, Entalpia Agua)
Eficiencia = eficiencia(k, ExcessoAr, arcomb, cpgas, Tchamine, Tar, pci)

FluxoCombustivel = consumo(CalorUtil, pci, Eficiencia)
perdaPurgas = FluxoCombustivel * perdaPurgas
perdacinza = FluxoCombustivel * perdacinza
etotal = FluxoCombustivel * pci
purg = perdaPurgas / etotal
ash = (perdacinza / etotal)
perdaChamine = 1 - Eficiencia - k - ash
enerchamine = perdaChamine * etotal
incomb = etotal * perdaCombustaoIncompleta / 100

```



```

eneramb = etotal * perdaAmbiente / 100

self.listWidget_dados.item(0).setText(f'PCI do combustivel: {pci:.2f} kJ/kg')
self.listWidget_dados.item(1).setText(f'Vazão ar/combustível estequiométrica:
{arcomb:.2f}')
self.listWidget_dados.item(2).setText(
    f'Vazão mássica de combustível: {FluxoCombustivel * 3600:.2f} kg/h')
self.listWidget_perdas.item(8).setText(f'Perda para o ambiente: {eneramb:.2f}
kW')
self.listWidget_perdas.item(4).setText(f'Perda combustão incompleta:
{incomb:.2f} kW')
self.listWidget_perdas.item(5).setText(f'Perda no calor das cinzas:
{perdacinza:.2f} kW')
self.listWidget_perdas.item(6).setText(f'Perda nos gases de escape:
{enerchamine:.2f} kW')
self.listWidget_perdas.item(7).setText(f'Perda por purgas: {perdaPurgas:.2f}
kW')
self.listWidget_perdas.item(0).setText(f'Calor gerado pelo combustível:
{etotal:.2f} kW')
self.listWidget_perdas.item(1).setText(f'Calor útil: {CalorUtil:.2f} kW')
self.listWidget_perdas.item(3).setText(f'PERDAS")

# Grafico percentual de perdas

series = QtChart.QPieSeries()
series.append("Perda para o ambiente", perdaAmbiente)
series.append("Perda por Combustão incompleta", perdaCombustaoIncompleta)
series.append("Perda nas Cinzas", ash * 100)
slicham = QtChart.QPieSlice("Perda na chaminé", perdaChamine * 100)
slicham.setBrush(QtGui.QColor("red"))
series.append(slicham)
series.append("perda por purgas", purg * 100)
rend = QtChart.QPieSlice("Eficiência", Eficiencia * 100)
rend.setBrush(QtGui.QColor("green"))
series.append(rend)
series.setLabelsVisible()
for slices in series.slices():
    slices.setLabel("{:.1f} %".format(100 * slices.percentage()))

series.slices().__getitem__(0).setBrush(QtGui.QColor("yellow"))
series.slices().__getitem__(1).setBrush(QtGui.QColor("purple"))

chart = QtChart.QChart()
chart.addSeries(series)
chartview = QtChart.QChartView(chart)
chartview.setRenderHint(QtGui.QPainter.Antialiasing)
chart.legend().markers(series)[0].setLabel("Perda para o ambiente")
chart.legend().markers(series)[1].setLabel("Perda por combustão incompleta")
chart.legend().markers(series)[2].setLabel("Perda nas cinzas")
chart.legend().markers(series)[3].setLabel("Perda na chaminé")

```

```

chart.legend().markers(series)[4].setLabel("Perda por purgas")
chart.legend().markers(series)[5].setLabel("Eficiência")
self.janela = QtWidgets.QMdiSubWindow()
self.janela.resize(1000, 900)
self.janela.setWidget(chartview)
self.janela.show()
self.janela.move(-1, 1)
self.janela.setWindowTitle("Grafico de Perdas")
except:
    for i in range(11):
        self.listWidget_perdas.item(i).setText("")
    for i in range(3):
        self.listWidget_dados.item(i).setText("")
    self.listWidget_perdas.item(0).setText("Verificar valores inseridos")

if __name__ == "__main__":
    import sys

    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```