



# Instituto Superior de Engenharia

Politécnico de Coimbra

DEPARTAMENTO DE ENGENHARIA MECÂNICA

## Desenvolvimento de uma aplicação para análise de combustão: *Flame Solver v1.0*

Trabalho de projeto no âmbito da unidade curricular de Métodos Computacionais em Engenharia do Mestrado em Engenharia Mecânica  
Especialização em Projeto, Instalação e Manutenção de Sistemas Térmicos

Autor

**Mário Mendes Loureiro**

Orientadores

**Prof. Raquel Almeida de Azevedo Faria**

**Prof. Arménio António da Silva Correia**



INSTITUTO POLITÉCNICO  
DE COIMBRA

INSTITUTO SUPERIOR  
DE ENGENHARIA  
DE COIMBRA

Coimbra, julho de 2025



*An experiment is a question which science poses to Nature,  
and a measurement is the recording of Nature's answer.*

**Max Planck**

## ÍNDICE

|  |     |
|--|-----|
| Índice.....  | ii  |
| Índice de figuras.....   | iii |
| Lista de abreviaturas.....                                       | iv  |
| 1 Introdução .....   | 1   |
| 1.1 Objetivos.....   | 1   |
| 2 Descrição da aplicação .....                                   | 3   |
| 2.1 Separador <i>Fuel</i> .....                                  | 3   |
| 2.2 Separador <i>Air</i> .....                                   | 3   |
| 2.3 Separador <i>Combustion Analysis</i> .....                   | 4   |
| 2.4 Separador <i>Flame Temperature</i> .....                     | 5   |
| 2.5 Separador <i>Ostwald Triangle</i> .....                      | 5   |
| 2.6 Integração entre aplicações para medição de gases.....       | 6   |
| 3 Funcionalidades e cálculos.....                                | 7   |
| 3.1 Análise da combustão.....                                    | 7   |
| 3.2 Cálculo da temperatura dos produtos de combustão .....       | 7   |
| 3.3 Cálculo do Triângulo de Ostwald .....                        | 8   |
| 4 Exemplo de aplicação.....                                      | 9   |
| 4.1 Análise da combustão.....                                    | 9   |
| 4.2 Análise da temperatura dos produtos .....                    | 10  |
| 4.3 Análise de um ensaio de combustão.....                       | 11  |
| 5 Conclusão .....  | 13  |
| 6 Referências.....   | 14  |
| Anexo A – Código-fonte da aplicação <i>Flame Solver</i> .....    | 15  |
| Anexo B – Código-fonte da aplicação <i>Probe Dashboard</i> ..... | 16  |
| Anexo C – Código-fonte da aplicação <i>About</i> .....           | 17  |

## **ÍNDICE DE FIGURAS**

|   |    |
|---|----|
| Figura 1 – Separador <i>Fuel</i> .....  | 3  |
| Figura 2 – Separador <i>Air</i> .....   | 4  |
| Figura 3 – Separador <i>Combustion Analysis</i> .....   | 4  |
| Figura 4 – Separador <i>Flame Temperature</i> .....   | 5  |
| Figura 5 – Separador <i>Ostwald Triangle</i> .....  | 5  |
| Figura 6 – Evolução da composição ponderal dos fumos secos em função do excesso de ar.....                | 10 |
| Figura 7 – Temperatura dos produtos da combustão em função do excesso de ar. ....                         | 10 |
| Figura 8 – Montagem da sonda analisadora para ensaio em automóvel com motor de combustão a gasolina. .... | 11 |
| Figura 9 – Traçado do Triângulo de <i>Ostwald</i> com base nas medidas da sonda. ....                     | 11 |

## **LISTA DE ABREVIATURAS**

MQTT      *Message Queuing Telemetry Transport*

## **1 INTRODUÇÃO**

A análise e otimização de processos de combustão é essencial em várias áreas da engenharia. Compreender as relações entre os combustíveis, ar necessário e produtos de combustão é fundamental para a otimização da eficiência energética e para redução das emissões de poluentes. Para isso, a utilização de aplicações computacionais permite acelerar os processos de otimização por via da simulação e cálculo das relações entre parâmetros.

Neste contexto, a aplicação *Flame Solver* foi desenvolvida com um objetivo de providenciar uma base para a análise e otimização de combustão destes sistemas térmicos. A aplicação permite ao utilizador combinar diversos tipos de combustíveis, calcular os produtos de combustão, a temperatura destes e visualizar gráficos que ilustram a relação entre as diversas variáveis envolvidas neste tipo de processos. A aplicação permite ainda gerar relatórios detalhados para auxiliar na interpretação e otimização dos processos de combustão.

Além disso, a aplicação integra-se com uma sonda de gases de combustão que permite realizar leituras de dados de sensores de temperatura e de gases numa situação de ensaio. Com base na leitura da composição volumétrica de gases de combustão é possível realizar uma análise mais completa determinando, ainda, a temperatura dos mesmos. Assim, a aplicação desenvolvida permite realizar análises baseadas tanto em dados teóricos quanto em medições de ensaios práticos, proporcionando uma análise mais rápida e que pode ser comparada com experiências reais.

A aplicação foi desenvolvida com uma interface gráfica intuitiva e dividida em separadores, onde o utilizador insere dados, podendo ajustar parâmetros e visualizar resultados da análise em tempo real e das medições da sonda.

Ao longo deste trabalho a aplicação será explorada de forma detalhada com foco nas suas funcionalidades e cálculos. A descrição das ferramentas e funcionalidades da interface do utilizador e os cálculos realizados forneceram uma compreensão completa sobre o seu funcionamento. Por fim, exemplifica-se um caso prático de utilização da aplicação.

### **1.1 Objetivos**

O objetivo principal desta aplicação é fornecer uma ferramenta para a análise de combustão de diferentes combustíveis, permitindo o cálculo e a visualização gráfica da influência dos diversos parâmetros no processo de combustão, nos produtos de combustão e sua temperatura, entre outros aspectos. A aplicação é especialmente desenvolvida para engenheiros mecânicos que trabalham em projeto

de sistemas térmicos que funcionem com base em combustão, fornecendo dados quantitativos e gráficos que permitem otimizar este processo e a compreensão da relação entre as diversas variáveis envolvidas.

Através da interface gráfica da aplicação, um utilizador pode selecionar e definir o teor volumétrico dos diferentes componentes dos combustíveis simulando diferentes misturas. Permite calcular os produtos da combustão em função do excesso de ar aplicado ao sistema e determinar a temperatura dos produtos de combustão com base nos dados de um balanço energético e nas condições higrotérmicas da combustão. Ademais visualizar o Triângulo de *Ostwald*, que representa a relação entre os teores volumétricos de dióxido de carbono, oxigénio, monóxido de carbono e excesso de ar permitindo uma análise completa de um ensaio de combustão por via da comunicação com uma sonda. Possibilita ainda gerar relatórios detalhados incluindo os cálculos realizados pela aplicação para posterior análise e otimização de processos de combustão.

## 2 DESCRIÇÃO DA APLICAÇÃO

Neste capítulo, apresenta-se uma descrição detalhada sobre os separadores da aplicação principal, *Flame Solver*. Esta aplicação é utilizada para realizar análise de combustão, considerando diferentes combustíveis e parâmetros de combustão e dos constituintes da mesma. A aplicação está organizada em separadores, cada um com um conjunto de funcionalidades específicas e encadeadas. Apresenta-se uma descrição de cada separador, referindo-se os constituintes, funcionalidades e princípio de funcionamento. Por fim, refere-se como ocorre a interligação entre a aplicação principal e a aplicação *Probe dashboard*, responsável por receber medições da sonda.

### 2.1 Separador *Fuel*

Este separador permite inserir até 6 componentes de combustível e definir o seu teor volumétrico. Os componentes principais são as caixas *dropdown* que permitem selecionar o tipo de combustível a partir de uma base de dados num ficheiro .xlsx que contém uma seleção de hidrocarbonetos líquidos e gasosos e respetivas propriedades e fatores multiplicativos para análise de combustão (Figura 1).

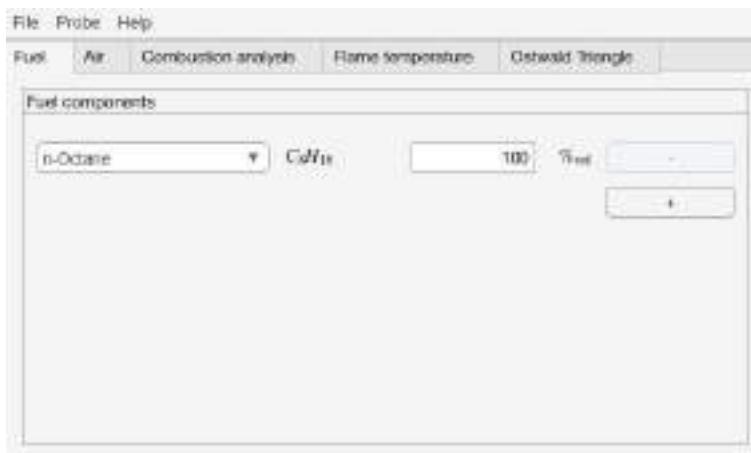


Figura 1 – Separador *Fuel*.

Apresenta etiquetas em TeX para ilustração das fórmulas químicas e ainda botões para adição ou remoção de campos dos combustíveis.

### 2.2 Separador *Air*

O separador *Air* tem como objetivos o cálculo das necessidades de ar estequiométrico, excesso de ar e contabilização de humidade. As necessidades de ar estequiométricos são calculadas automaticamente em função dos dados inseridos no separador *fuel*. Estas necessidades são os valores base para o cálculo das necessidades

com excesso de ar, sendo o valor de excesso de ar inserido pelo utilizador em campo apropriado (Figura 2).

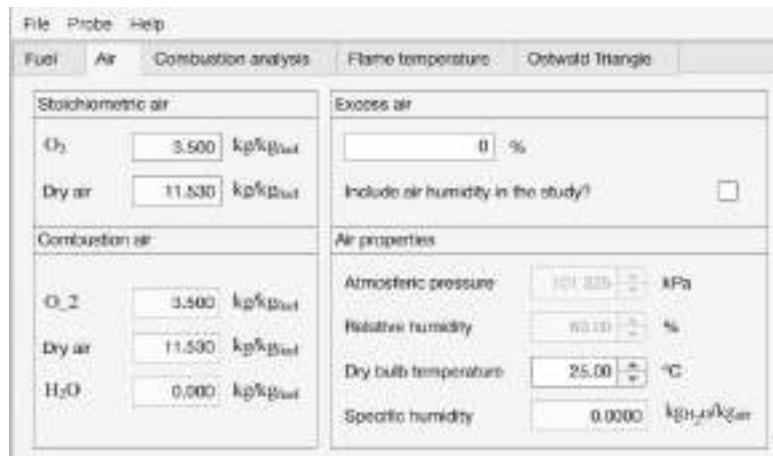


Figura 2 – Separador *Air*.

O teor de vapor de água no ar é contabilizado em função da humidade relativa, pressão atmosférica e temperatura de bolbo seco. Este valor é contabilizado tendo em conta o excesso de ar, sendo ativado ou desativado com recurso a uma *checkbox*.

### 2.3 Separador *Combustion Analysis*

O separador *Combustion Analysis* ilustra o cálculo automático do teor ponderal dos produtos de combustão em campos não editáveis. Estes valores apresentados são para os dados introduzidos nos dois separadores anteriores. Do lado direito apresenta-se um gráfico que compara a evolução do teor ponderal dos produtos de combustão em função do excesso de ar. Ao mover-se o *slider* lateral é possível ajustar o intervalo de excesso de ar que se pretende observar. O traçado deste gráfico é essencial para o processo de otimização da combustão (Figura 3).

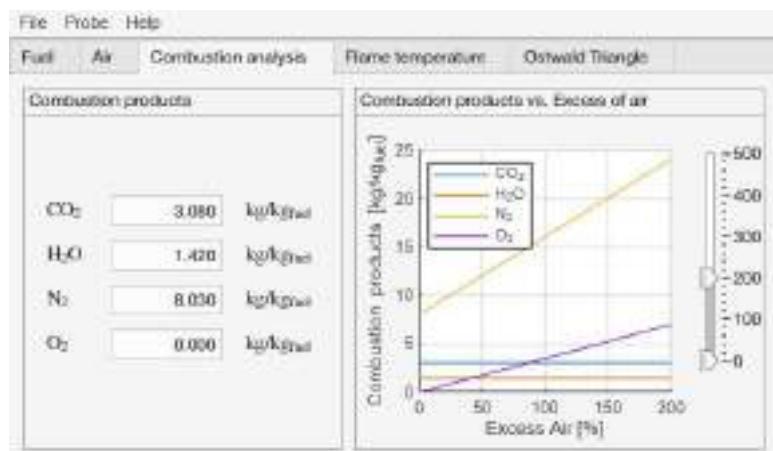


Figura 3 – Separador *Combustion Analysis*.

## 2.4 Separador *Flame Temperature*

Este separador permite simular a temperatura dos produtos de combustão em função do excesso de ar. O balanço de energia que permite considerar calor e trabalho efetuado sobre o sistema ou pelo sistema, dada a utilidade, nomeadamente para análise da combustão de motores de combustão interna.

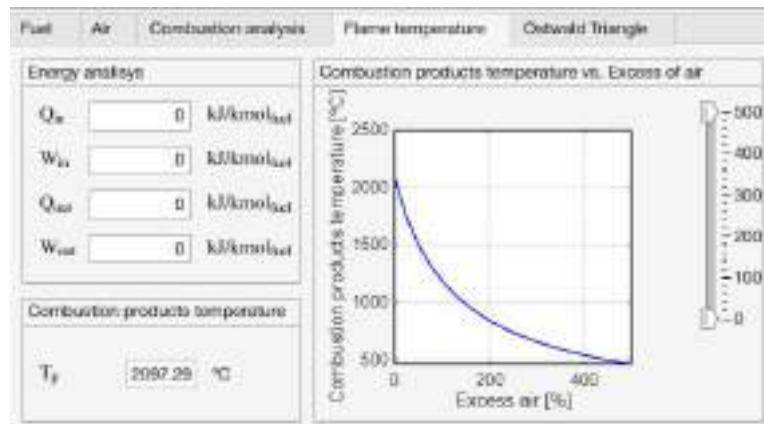


Figura 4 – Separador *Flame Temperature*.

À direita observa-se um gráfico que representa a relação da temperatura dos produtos em função do excesso de ar.

## 2.5 Separador *Ostwald Triangle*

O separador *Ostwald Triangle* permite, a partir das medições dos sensores ( $O_2$  e CO), determinar graficamente o triângulo de *Ostwald*. Determina-se as coordenadas correspondentes ao teor máximo de oxigénio, ao teor máximo de dióxido de carbono para a combustão estequiométrica e as coordenadas para a combustão incompleta. Obtém-se por esta via o teor volumétrico de dióxido de carbono e o excesso de ar com recurso a apenas duas medições e a dois cálculos de combustão. A Figura 5 representa o separador referido.

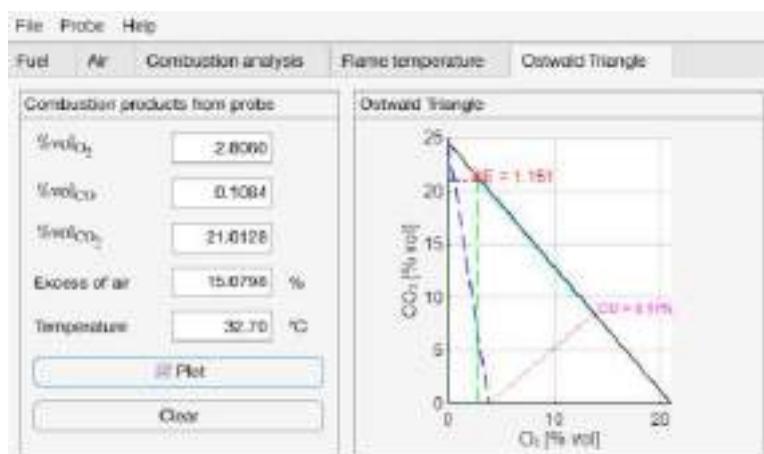


Figura 5 – Separador *Ostwald Triangle*.

## 2.6 Integração entre aplicações para medição de gases

A aplicação *Flame Solver* permite gerar uma análise de combustão e recuperação dos dados de uma sonda por via da função `getappdata()` e gerar o triângulo de *Ostwald* que permite calcular o teor (em volume) de dióxido de carbono nos fumos secos e o excesso de ar. A aplicação *Probe Dashboard* permite efetuar medições de temperatura, teor de monóxido de carbono e teor de oxigénio, necessários à geração gráfica do triângulo de *Ostwald*. Os dados são armazenados em variáveis partilhadas usando `setappdata()`.

A sonda consiste um microcontrolador ESP32 que efetua leitura dos sensores, MQ-7 e YYS SC24J24OS, para medição dos teores de monóxido de carbono e de oxigénio, respetivamente. A temperatura é medida com recurso a um termopar do tipo K. O microcontrolador, com interface wi-fi integrada comunica com um *broker* MQTT ([test.mosquitto.org](http://test.mosquitto.org)), sendo o *publisher* em `esp32/sensors/#`. O microcontrolador retira uma série de valores para garantir que existe pelo menos um valor de cada variável disponível no *broker*. Permite, assim, evitar a ausência de valores no *broker* por problemas de conexão ao wi-fi.

A aplicação principal, *Flame Solver*, utiliza uma biblioteca que permite obter os valores na modalidade *subscriber* como cliente MQTT. Seleciona o último valor de cada variável e faz uso das mesmas para gerar o Triângulo de *Ostwald*.

### 3 FUNCIONALIDADES E CÁLCULOS

Neste capítulo são descritos as funcionalidades e os cálculos que podem ser realizados com a aplicação. A aplicação foi desenvolvida para fornecer uma análise detalhada do processo de combustão, desde a seleção combustíveis até ao cálculo de produtos de combustão e da temperatura dos mesmos. Além disso, permite a visualização do triângulo de *Ostwald*, uma ferramenta essencial para a análise de excesso de ar e dos produtos de combustão de ensaios práticos a seguir são detalhados os principais cálculos e funcionalidades da aplicação.

#### 3.1 Análise da combustão

A análise da combustão inicia-se pela seleção dos combustíveis a combinar e por definição da sua composição volumétrica. A aplicação calcula automaticamente a composição ponderal obtendo-se a composição ponderal dos componentes do ar teoricamente necessário por via dos fatores multiplicativos associados. Os fatores multiplicativos, massa molar e outras propriedades relevantes para o cálculo são definidos numa base de dados que a aplicação consulta de modo a efetuar estes cálculos.

Posteriormente, a aplicação calcula os produtos de combustão considerando, ou não, a existência de vapor de água no ar. Aplica-se, posteriormente, se inserido pelo utilizador, um excesso de ar. Os produtos de combustão são obtidos, em composição ponderal, obtendo-se também um gráfico que relaciona estes com o excesso de ar.

#### 3.2 Cálculo da temperatura dos produtos de combustão

O cálculo da temperatura dos produtos de combustão é obtido em função de um balanço energético que considera o calor, trabalho e as entalpias de formação e sensíveis. A Equação 1 representa o balanço térmico referido.

$$Q + W + \sum N_P (\bar{h}_f^o + \bar{h} - \bar{h}^o)_P = Q + W + \sum N (\bar{h}_f^o + \bar{h} - \bar{h}^o) \quad 1$$

As Equações 2 a 5 estabelecem uma relação entre a entalpia sensível e a temperatura dos elementos químicos. No caso dos reagentes, considera-se que o combustível está a 25°C e 1 atm. No caso dos restantes reagentes considera-se a temperatura e pressão atmosférica inserida pelo utilizador.

$$\bar{h} = 3 \times 10^{-12}T^4 - 5 \times 10^{-7}T^3 + 0,039T + 26,352T + 447,29 \quad 2$$

$$\bar{h} = -9 \times 10^{-17}T^6 + 9 \times 10^{-T^5} - 3 \times 10^{-9}T^4 + 4 \times 10^{-6}T^3 + 0,0019T + 27,751T + 126,42 \quad 3$$

$$\begin{aligned}\bar{h} &= -2 \times 10^{-6}T^3 + 0,0118T + 34,701T - 2120,1 & 4 \\ \bar{h} &= -6 \times 10^{-17}T^6 + 8 \times 10^{-13}T^5 - 4 \times 10^{-9}T^4 - 8 \times 10^{-6}T^3 - 0,0028T + 33,367T + 5,7479 & 5\end{aligned}$$

No caso dos produtos da combustão, estas relações são estabelecidas tendo-se com incógnita a temperatura comum dos mesmos.

### 3.3 Cálculo do Triângulo de Ostwald

O triângulo de *Ostwald* é uma abordagem rápida para aumentar a eficiência da combustão, tendo a particularidade de entrar em conta com as condições reais de combustão relacionando quatro grandezas na constituição dos fumos secos:

- Teor volumétrico de CO<sub>2</sub> nos fumos secos;
- Teor volumétrico de CO nos fumos secos;
- Teor volumétrico de O<sub>2</sub> nos fumos secos;
- Excesso de ar.

Quando duas das grandezas são conhecidas, no caso em questão, o CO e o O<sub>2</sub>, assim com a composição do combustível, é possível determinar as outras duas sem ambiguidade. O cálculo efetua-se por via gráfica com uma análise de combustão completa e incompleta para determinação dos extremos das retas que compõem o Triângulo de *Ostwald*.

## 4 EXEMPLO DE APLICAÇÃO

Considera-se um exemplo de aplicação onde se queima octano num motor de combustão interna. O combustível e o ar encontram-se a 25°C. O ar apresenta uma humidade relativa de 90% e uma pressão atmosférica de 1 atm. Considerou-se um excesso de ar de 15%. Considerou-se que  $Q_{out} = 75000\text{kJ}/\text{kmol}$  e  $W_{out} = 120000\text{kJ}/\text{kmol}$ .

### 4.1 Análise da combustão

Considerando a massa molar do octano e respetivos fatores multiplicativos, obteve-se para uma combustão estequiométrica os seguintes valores para o ar teoricamente necessário:

- $O_2 = 3,5\text{kg/kg}_{\text{combustível}}$
- Ar seco =  $11,53\text{kg/kg}_{\text{combustível}}$

Com base nestas valores considera-se o vapor de água presente no ar, a temperatura de bolbo seco, humidade relativa e pressão atmosférica referidas inicialmente. Aplica-se o excesso de ar de 15%, obtendo-se os seguintes valores para o ar:

- $O_2 = 4,025 \text{ kg/kg}_{\text{combustível}}$
- Ar seco =  $13,259 \text{ kg/kg}_{\text{combustível}}$
- $H_2O = 0,239 \text{ kg/kg}_{\text{combustível}}$

Os produtos de combustão para esta condição de excesso de ar serão:

- $CO_2 = 3,080 \text{ kg/kg}_{\text{combustível}}$
- $H_2O = 1,659 \text{ kg/kg}_{\text{combustível}}$
- $N_2 = 9,234 \text{ kg/kg}_{\text{combustível}}$
- $O_2 = 0,525 \text{ kg/kg}_{\text{combustível}}$

A Figura 6 representa o traçado da evolução da composição ponderal dos fumos secos em função do excesso de ar obtido com a aplicação onde se verificam valores idênticos aos calculados anteriormente.

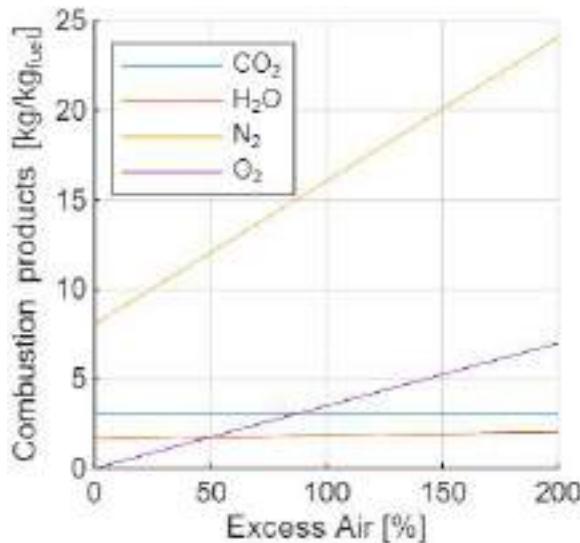


Figura 6 – Evolução da composição ponderal dos fumos secos em função do excesso de ar.

## 4.2 Análise da temperatura dos produtos

Para a análise da temperatura dos produtos de combustão considerou-se a análise anterior e que  $Q_{out} = 75000\text{kJ}/\text{kmol}$  e  $W_{out} = 120000\text{kJ}/\text{kmol}$ , obtendo-se uma temperatura para os produtos de combustão de  $T_p = 1733,41^\circ\text{C}$ , por realização do seguinte balanço de energia da Equação 6.

$$75000 + 120000 + \sum N_P (\bar{h}_f^o + \bar{h} - \bar{h}^o)_P = Q + W + \sum N (\bar{h}_f^o + \bar{h} - \bar{h}^o) \quad 6$$

A Figura 7 representa a evolução da temperatura dos produtos da combustão em função do excesso de ar obtido com a aplicação para o caso em estudo.

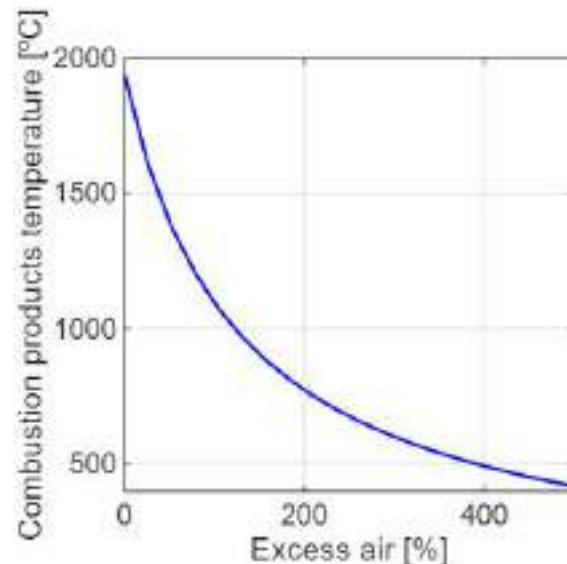


Figura 7 – Temperatura dos produtos da combustão em função do excesso de ar.

### 4.3 Análise de um ensaio de combustão

Posteriormente, realizou-se um ensaio nas condições referidas para o combustível e para o ar. Obteve-se medidas da composição volumétrica dos produtos de combustão (monóxido de carbono e oxigénio) para um período de funcionamento pouco após o arranque a frio do motor. A Figura 8 representa a montagem da sonda analisadora para ensaio num automóvel a gasolina. Considerou-se que a gasolina se assemelha completamente ao octano.



Figura 8 – Montagem da sonda analisadora para ensaio em automóvel com motor de combustão a gasolina.

As medidas da sonda, com recurso ao *Probe Dashboard*, resultaram no traçado do Triângulo de Ostwald da Figura 9.

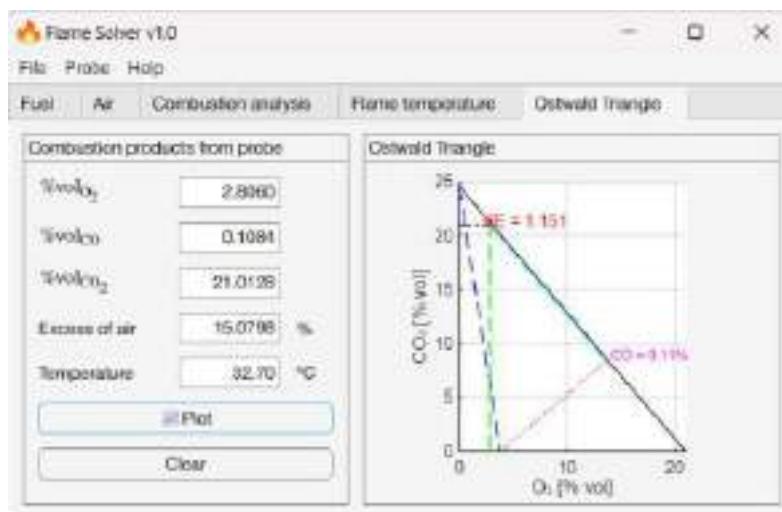


Figura 9 – Traçado do Triângulo de Ostwald com base nas medidas da sonda.

Observa-se que o motor se encontra a operar com cerca de 15% de excesso de ar. O teor volumétrico de O<sub>2</sub> reduziu de cerca de 24% para 2,806% em relação ao momento em que o motor ainda se encontrava parado. O teor de monóxido de carbono encontra-se em cerca de 0,1084 %, valor passível dado o limite superior da

sonda de 0,2 % ou 2000 ppm. A temperatura encontra-se mais elevada que ao ambiente, mas ainda de forma não significativa, algo que seria de esperar dado que o motor arrancou pouco tempo antes da leitura. No entanto, será esperado que a temperatura não atinja a temperatura teórica em função da longa linha de escape respetivos acessórios. Os valores de  $Q_{out}$  e  $W_{out}$  deverão ser corrigidos para os valores reais ou estimados, para que se possa realizar uma análise mais precisa.

No entanto, o Triângulo de *Ostwald* para o caso de estudo permite perceber oportunidades de melhoria ou verificação das condições ótimas de combustão.

## 5 CONCLUSÃO

A aplicação desenvolvida tem o objetivo de ser uma ferramenta para análise de combustão com vista à otimização de processos de combustão em equipamentos térmicos. Permite a combinação de diferentes combustíveis, o cálculo do ar necessário, dos produtos de combustão e a determinação da temperatura dos mesmos, além de possibilitar a visualização do Triângulo de *Ostwald*. Trata-se de uma ferramenta essencial para otimizar a combustão de combustíveis e controlar o excesso de ar. A integração de uma sonda para utilização em ensaios permite a monitorização dos gases de combustão dando uma ênfase mais prática a aplicação desenvolvida. Proporciona, assim, uma visão mais abrangente para o processo de otimização revelando-se uma ferramenta que acelera este processo.

A interface gráfica foi desenvolvida para facilitar a inserção de dados pelo utilizador e a capacidade de realização de cálculos automáticos e de gerar relatórios detalhados. Em termos de funcionalidades, a aplicação é eficiente para cálculo dos produtos de combustão e temperatura dos mesmos e permite gerar gráficos interativos e personalizáveis. O exemplo da aplicação demonstrada destaca a capacidade de aplicação em simulação e preparação de cenários teóricos de combustão com medidas ensaios práticos que, em conjunto, permitem o controlo das emissões e da eficiência de combustão. Embora a aplicação desenvolvida tenha alcançado os objetivos iniciais, existem ainda várias oportunidades de melhoria e de expansão das suas funcionalidades.

Primeiramente, refere-se a possibilidade de expansão da base de dados de combustíveis, abrangendo uma maior variedade de hidrocarbonetos, incluindo biocombustíveis. Futuramente, também se ambiciona adoção do software para combustíveis sólidos sendo adequado às mais variadas aplicações industriais.

A interface e funcionamento também pode ser melhorada fornecendo, por exemplo, feedback em tempo real e sugestões automáticas para ajustes nos parâmetros de combustão. Estes ajustes podem ser realizados com o auxílio de ferramentas de *machine learning*. Estas funcionalidades poderão adicionar grande valor tornando-o ainda numa ferramenta mais prática.

Outra oportunidade de melhoria seria a expansão para outras áreas da engenharia térmica onde se poderia analisar ciclos termodinâmicos em turbinas ou caldeiras combinando as funcionalidades que já existem para otimização de combustão.

Com introdução destas melhorias, a aplicação desenvolvida poderá tornar-se uma ferramenta ainda mais poderosa e útil para a engenharia térmica contribuindo para a otimização da eficiência energética e redução da emissão de poluentes nos sistemas de combustão, encontramos assim alinhada com os Objetivos do Desenvolvimento Sustentável.

## **6 REFERÊNCIAS**

Costa, R. da. (sem data). Caldeiras e Permutadores de Calor segundo as Lições do Eng. Ribeiro da Costa. Instituto Superior de Engenharia do Porto.

Develop Apps Using App Designer—MATLAB & Simulink. (sem data). Obtido 23 de outubro de 2024, de <https://www.mathworks.com/help/matlab/app-designer.html>

Holmgren, M. (2025, julho 17). X Steam, Thermodynamic properties of water and steam. <https://www.mathworks.com/matlabcentral/fileexchange/9817-x-steam-thermodynamic-properties-of-water-and-steam>

Industrial Communication Toolbox Documentation. (2025). Obtido 11 de janeiro de 2025, de <https://www.mathworks.com/help/icomm/index.html>

Symbolic Math Toolbox Documentation. (2025). Obtido 9 de setembro de 2024, de <https://www.mathworks.com/help/symbolic/index.html>

Turns, S. R. (2000). An introduction to combustion: Concepts and applications (2. ed). McGraw-Hill.

**ANEXO A – CÓDIGO-FONTE DA APLICAÇÃO *FLAME SOLVER***

```

classdef flame_solver < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
    UIFigure                      matlab.ui.Figure
    FileMenu                       matlab.ui.container.Menu
    ExampleMenu                    matlab.ui.container.Menu
    LoadexampleMenu               matlab.ui.container.Menu
    ExampleredocumentationMenu   matlab.ui.container.Menu
    GeneratereportMenu            matlab.ui.container.Menu
    ClearMenu                      matlab.ui.container.Menu
    ProbeMenu                      matlab.ui.container.Menu
    ProbashboardMenu               matlab.ui.container.Menu
    ProjectreportMenu              matlab.ui.container.Menu
    HelpMenu                       matlab.ui.container.Menu
    HelpMenu_3                     matlab.ui.container.Menu
    AboutMenu                      matlab.ui.container.Menu
    TabGroup                       matlab.ui.container.TabGroup
    FuelTab                        matlab.ui.container.Tab
    FuelcomponentsPanel           matlab.ui.container.Panel
    GridLayout                     matlab.ui.container.GridLayout
    volEditfield_6                 matlab.ui.control.NumericEditField
    volLabel_6                      matlab.ui.control.Label
    volEditfield_5                 matlab.ui.control.NumericEditField
    volLabel_5                      matlab.ui.control.Label
    volEditfield_4                 matlab.ui.control.NumericEditField
    volLabel_4                      matlab.ui.control.Label
    volEditfield_3                 matlab.ui.control.NumericEditField
    volLabel_3                      matlab.ui.control.Label
    volEditfield_2                 matlab.ui.control.NumericEditField
    volLabel_2                      matlab.ui.control.Label
    volEditfield_1                 matlab.ui.control.NumericEditField
    volLabel_1                      matlab.ui.control.Label
    Label_6                         matlab.ui.control.Label
    Label_5                         matlab.ui.control.Label
    Label_4                         matlab.ui.control.Label
    Label_3                         matlab.ui.control.Label
    Label_2                         matlab.ui.control.Label
    Label_1                         matlab.ui.control.Label
    DropDown_6                      matlab.ui.control.DropDown
    DropDown_5                      matlab.ui.control.DropDown
    DropDown_4                      matlab.ui.control.DropDown
    DropDown_3                      matlab.ui.control.DropDown
    DropDown_2                      matlab.ui.control.DropDown
    DropDown                         matlab.ui.control.DropDown
    Button_minus                    matlab.ui.control.Button
    Button_plus                     matlab.ui.control.Button
    AirTab                          matlab.ui.container.Tab
    AirpropertiesPanel              matlab.ui.container.Panel
    GridLayout4                     matlab.ui.container.GridLayout
    kg_H_20kg_airLabel             matlab.ui.control.Label
    SpecifichumidityLabel          matlab.ui.control.Label
    EditField_Spec_Humidity        matlab.ui.control.NumericEditField
    SpinnerDryBulbTemperature     matlab.ui.control.Spinner
    SpinnerRelativeHumidity       matlab.ui.control.Spinner
    SpinnerAtmosphericPressure   matlab.ui.control.Spinner
    CLabel                          matlab.ui.control.Label
    DrybulbtemperatureLabel       matlab.ui.control.Label
    Label_8                         matlab.ui.control.Label
    RelativehumidityLabel          matlab.ui.control.Label
    kPaLabel                        matlab.ui.control.Label
    AtmosfericpressureLabel      matlab.ui.control.Label
    CombustionairPanel             matlab.ui.container.Panel
    GridLayout2                     matlab.ui.container.GridLayout
    EditField_Air_H2O_Final        matlab.ui.control.NumericEditField
    EditField_Dry_Air_Final        matlab.ui.control.NumericEditField
    EditField_O2_Final              matlab.ui.control.NumericEditField
    kgkg_fuellabel_9               matlab.ui.control.Label
    H_20Llabel_2                   matlab.ui.control.Label
    kgkg_fuellabel_4               matlab.ui.control.Label
    DryairLabel_2                  matlab.ui.control.Label
    kgkg_fuellabel_3               matlab.ui.control.Label
    O_2label_2                     matlab.ui.control.Label
    StoichiometricairPanel         matlab.ui.container.Panel
    GridLayout3                     matlab.ui.container.GridLayout
    Editfield_Dry_Air              matlab.ui.control.EditField
    kgkg_fuellabel_2               matlab.ui.control.Label
    DryairLabel                    matlab.ui.control.Label
    EditField_O2                    matlab.ui.control.EditField
    kgkg_fuellabel_1               matlab.ui.control.Label
    O_2label_1                     matlab.ui.control.Label
    ExcessairPanel                 matlab.ui.container.Panel
    GridLayout5                     matlab.ui.container.GridLayout
    CheckBoxHumidAir               matlab.ui.control.CheckBox
    IncludeairhumidityinhestudyLabel matlab.ui.control.Label
    Label_7                         matlab.ui.control.Label
    EditfieldExcessAir              matlab.ui.control.NumericEditField
    CombustionanalysisTab          matlab.ui.container.Tab
    GridLayout9                     matlab.ui.container.GridLayout
    CombustionproductsvsExcessofairPanel matlab.ui.container.Panel
    ExcessofAirSlider               matlab.ui.control.RangeSlider
    UIAxes_CB                       matlab.ui.control.UIAxes
    CombustionproductsPanel         matlab.ui.container.Panel
    GridLayout6                     matlab.ui.container.GridLayout
    Editfield_CB_O2                 matlab.ui.control.NumericEditField
    Editfield_CB_N2                 matlab.ui.control.NumericEditField
    Editfield_CB_H2O                matlab.ui.control.NumericEditField
    Editfield_CB_CO2                matlab.ui.control.NumericEditField
    kgkg_fuellabel_8               matlab.ui.control.Label
    O_2label_3                     matlab.ui.control.Label
    kgkg_fuellabel_7               matlab.ui.control.Label
    N_2label                         matlab.ui.control.Label

```

```

kgkg_fuelLabel_6      matlab.ui.control.Label
H_20Label             matlab.ui.control.Label
kgkg_fuelLabel_5      matlab.ui.control.Label
CO_2label              matlab.ui.control.Label
FlametemperatureTab   matlab.ui.container.Tab
GridLayout8            matlab.ui.container.GridLayout
CombustionproductstemperaturevsExcessofairPanel matlab.ui.container.Panel
SliderT_p              matlab.ui.control.RangeSlider
UIAxesT_p              matlab.ui.control.UIAxes
CombustionproductstemperaturePanel matlab.ui.container.Panel
GridLayout10           matlab.ui.container.GridLayout
T_pLabel               matlab.ui.control.Label
CLabel_2               matlab.ui.control.Label
EditFielddT_p          matlab.ui.control.NumericEditField
EnergyanalysysPanel   matlab.ui.container.Panel
GridLayout5_2           matlab.ui.container.GridLayout
kJkmol_fuellabel_4    matlab.ui.control.Label
kJkmol_fuellabel_3    matlab.ui.control.Label
kJkmol_fuellabel_2    matlab.ui.control.Label
EditFieldWout          matlab.ui.control.NumericEditField
EditFieldQout          matlab.ui.control.NumericEditField
EditFieldWin            matlab.ui.control.NumericEditField
EditFieldQin            matlab.ui.control.NumericEditField
W_outLabel             matlab.ui.control.Label
Q_outLabel             matlab.ui.control.Label
W_inLabel              matlab.ui.control.Label
Q_inLabel              matlab.ui.control.Label
kJkmol_fuelLabel       matlab.ui.control.Label
OstwaldTriangleTab    matlab.ui.container.Tab
GridLayout11           matlab.ui.container.GridLayout
OstwaldTrianglePanel  matlab.ui.container.Panel
UIAxesOstwald          matlab.ui.control.UIAxes
CombustionproductsfromprobePanel matlab.ui.container.Panel
GridLayout5_3           matlab.ui.container.GridLayout
ClearButton             matlab.ui.control.Button
PlotButton              matlab.ui.control.Button
TemperatureLabel        matlab.ui.control.Label
Label_9                 matlab.ui.control.Label
ExcessofairLabel        matlab.ui.control.Label
vol_CO_2Label           matlab.ui.control.Label
vol_COLabel             matlab.ui.control.Label
CLabel_3                matlab.ui.control.Label
EditFieldOstT           matlab.ui.control.NumericEditField
EditFieldOstE           matlab.ui.control.NumericEditField
EditFieldOstCO2          matlab.ui.control.NumericEditField
EditFieldOstCO           matlab.ui.control.NumericEditField
EditFieldOstO2           matlab.ui.control.NumericEditField
vol_O_2Label             matlab.ui.control.Label
end

```

#### properties (Access = private)

```

FuelComponents ...
= struct('DropDown',{},{}, ...
'volEditField',{},{}, ...
'volLabel',{},{}, ...
'formulaLabel',{});
NumFuel = 1;
FuelsTable table
CollectedFuelData
end

```

#### methods (Access = private)

```

function updateFuelFields(app)
app.NumFuel = max(1, min(app.NumFuel, ...
    numel(app.FuelComponents)));
for i = 1: numel(app.FuelComponents)
    visible = i <= app.NumFuel;
    app.FuelComponents(i).DropDown.Visible = visible;
    app.FuelComponents(i).volEditfield.Visible = visible;
    app.FuelComponents(i).vollabel.Visible = visible;
    app.FuelComponents(i).formulaLabel.Visible = visible;
end
app.Button_minus.Enable = app.NumFuel > 1;
app.Button_plus.Enable = app.NumFuel < ...
    numel(app.FuelComponents);
end

function fuelValueChanged(app, event)
for i = 1:app.NumFuel
    app.FuelComponents(i).volEditField.BackgroundColor = ...
        'white';
end
vals = zeros(1, app.NumFuel);
for i = 1:app.NumFuel
    vals(i) = app.FuelComponents(i).volEditField.Value;
end
total = sum(vals);
if total > 100
    event.Source.BackgroundColor = [1, 0.6 ,0.6];
    uialert(app.UIFigure, ...
        'The sum of the volumetric percentages cannot exceed 100%.', ...
        'Invalid value');
    return
end
if total < 100
    for i = 1:app.NumFuel
        app.FuelComponents(i).volEditField.BackgroundColor ...
            = [1, 0.6 ,0.6];
    end
    uialert(app.UIFigure, ...
        'The sum of the volumetric percentages should be 100%.', ...

```

% Definition of private properties.

% Keeps references to the graphic elements of tab fuels.

% Number of active fuels in the fuels tab.

% Stores fuel data from the fuels.xlsx file.

% Compiles the data for the fuels selected in the fuel tab.

% Definition of private methods

```

% Controls the number of active dropdowns according to the
% buttons "+" e "-". Function declaration.
% Sets the number of active dropdowns between 1 and the number of
% suffixes _n (6 in total)
% For cycle runs through all the positions in the
% FuelComponents structure from 1 to 6.
% Visibility true if i is less than the number of fuels
% or false if it is larger, i.e. it hides it.
% Activates dropdowns up to the number of active dropdowns
% editfields, volume and chemical formula.
% Controlo da visibilidade dos botões "+" e "-"
% Enables the "-" button if more than one fuel is active
% Enables the "+" button if less than six fuels are active
% 'on' = 1 and 'off'=0.

```

% Sum validation function of the volumetric % editfields

% Defining the background colour of active volumetric % editfields
% as white.

% Creates the vector vals (with zeros) of dimension equal
% to the number of active fuels.
% Writes the value of each active fuel %vol in the array.

```

% Adds up the percentages of all active fuels, which must be 100%.
% Check if the total is more than 100%.
% If it is, put the background of the active editfield of the vol
% that originated the warning in light red.
% Creates the warning message.

```

% Check if the total is less than 100%.

% Set all active vol % edifield backgrounds to light red

% Creates the warning message.

```

        'Invalid total');
    return
end

function onFuelSelected(app, event)
try
    src = event.Source;
    idx = find(arrayfun(@(c) c.DropDown==src, ...
        app.FuelComponents),1);
    if isempty(idx), return; end
    selName = src.Value;
    row = find(strcmp(app.FuelsTable(:,2), selName),1);
    if isempty(row)
        warning('Fuel "%s" not found.', selName);
        return;
    end
    formula = app.FuelsTable{row,3}{1};
%
    app.FuelComponents(idx).formulaLabel.Text ...
        = ['$' formula '$'];
catch ME
    uialert( ...
app.UIFigure, ...
sprintf('Fuel "%s" does not exist', selName), ...
'Error', 'Icon','error');
end

function onTabChanged(app, event)
if event.NewValue == app.FuelTab
    app.CollectedFuelData = [];
end

if event.OldValue == app.FuelTab && ...
    event.NewValue ~= app.FuelTab
    vals = zeros(1, app.NumFuel);
    for i = 1:app.NumFuel
        vals(i) = app.FuelComponents(i).volEditField.Value;
    end
    totalVol = sum(vals);

    if totalVol ~= 100
        uialert(app.UIFigure, ...
            'The sum of the volumetric percentages should be 100%.', ...
            'invalid total');
        app.TabGroup.SelectedTab = app.FuelTab;
        return;
    end
    validateAndCollectFuels(app);
end
if event.NewValue == app.CombustionanalysisTab
    updateCombustionProducts(app);
end
if event.NewValue ~= app.FuelTab
    if isempty(app.CollectedFuelData)

        return;
    end
    T = app.CollectedFuelData;
    if ~ismember("WeightPct", T.Properties.VariableNames)
        try
            M = T.molar_mass;
            Vfrac = T.VolPct / 100;
            mi = M .* Vfrac;
            m_total = sum(mi);
            Wpct = mi / m_total;
            T.WeightPct = Wpct;
            T.nec_O2 = T.WeightPct .* T.fm_O2;
            T.nec_dry_air = T.WeightPct .* T.fm_dry_air;
            app.CollectedFuelData = T;
        catch ME
            return;
        end
    else
        end
    try
        T = app.CollectedFuelData;
        total_O2 = sum(T.nec_O2);
        total_air = sum(T.nec_dry_air);

        app.EditField_O2.Value = sprintf('%3f', total_O2);      % O2 (kg/kg_fuel)
        app.EditField_Dry_Air.Value = sprintf('%3f', total_air);    % Air (kg/kg_fuel)
    catch ME
    end
    updateFinalFields(app);
    updateHumidAir(app);
    updateCombustionProducts(app);
    plotCombustionVsExcessAir(app);
end
computeCurrentFlameTemperature(app);
plotFlameTemperatureVsExcessAir(app);
end

function validateAndCollectFuels(app)
N = app.NumFuel;
selNames = cell(N,1);
for i = 1:N
    selNames{i} = app.FuelComponents(i).DropDown.Value;
%
    % Function that validates fuel selection
    % and updates the chemical formula.
    % Stores the object that was selected.
    % Checks with @(c) c.DropDown = src which dropdown
    % originated the event (anonymous function).
    % Applies this function to each element arrayfun(app.FuelComponent
    % and returns a logical vector (true/1).
    % If the idx is empty ends the function.
    % Extract the second column (fuel name).
    % Check if there is a name in the second column.
%
    % Extracts the LaTeX formula from the fuels table.
%
    % Displays the formula in LaTeX.
%
    % Warning if the formula field is missing or there is an error.
%
    % Function that activates when the tab is changed.
    % Acquires the application and event that led to the change.
    % Verifica que a tab alterada
    % Clears the variable containing the data of the selected fuels

```

```

    end
    for i = 1:N
        name = selNames{i};
        if isempty(name) || (ischar(name) && any(isnan(name))) ...
            || (isstring(name) && ismissing(name))
            app.FuelComponents(i).DropDown.BackgroundColor ...
            = [1, 0.6, 0.6];
        for j = 1:N
            if j~=i
                app.FuelComponents(j).DropDown.BackgroundColor = 'white';
            end
        end

        uialert(app.UIFigure, ...
            sprintf('Invalid selection %d (empty fuel).', i), ...
            'Validation error');
        app.TabGroup.SelectedTab = app.FuelTab;
        return
    end
end
[uniqueNames, ~, idxUnique] = unique(selNames);
counts = histc(idxUnique, 1:numel(uniqueNames));
dupValues = uniqueNames(counts > 1);
if ~isempty(dupValues)
    for i = 1:N
        if any(strcmp(selNames{i}, dupValues))
            app.FuelComponents(i).DropDown.BackgroundColor ...
            = [1,0.6,0.6];
        else
            app.FuelComponents(i).DropDown.BackgroundColor ...
            = 'white';
        end
    end
    uialert(app.UIFigure, ...
        'The same fuel is being used more than once.', ...
        'Validation error');
    app.TabGroup.SelectedTab = app.FuelTab;
    return
end
for i = 1:N
    app.FuelComponents(i).DropDown.BackgroundColor = 'white';
end
rows = zeros(N,1);
vols = zeros(N,1);
for i = 1:N
    c = app.FuelComponents(i);
    rows(i) = find(strcmp(app.FuelsTable{:,2}, ...
        c.DropDown.Value),1);
    vols(i) = c.volEditField.Value;
end
T = app.FuelsTable(rows,:);
T.VolPct = vols;
app.CollectedFuelData = T;
end
function updateExcessAirField(app)
    if app.CheckBoxExcessAir.Value
        app.EditFieldWout.Editable = 'on';
    else
        app.EditFieldWout.Value = '100';
        app.EditFieldWout.Editable = 'off';
    end
end
function updateFinalFields(app)
    ea = (app.EditFieldExcessAir.Value + 100);
    T = app.CollectedFuelData;
    if isempty(T) || ~ismember('nec_02', ...
        T.Properties.VariableNames)
        return
    end
    finals_02 = T.nec_02 * ea/100;
    finals_dryAir = T.nec_dry_air * ea/100;
    app.EditField_O2_Final.Value = sum(finals_02);
    app.EditField_Dry_Air_Final.Value = sum(finals_dryAir);
    updateHumidAir(app);
end
function updateHumidAir(app, ~)
    if app.CheckBoxHumidAir.Value
        state = 'on';
    else
        state = 'off';
    end
    app.SpinnerAtmosfericPressure.Enable = state;
    app.SpinnerRelativeHumidity.Enable = state;
    app.SpinnerDryBulbTemperature.Enable = 'on';
    if ~app.CheckBoxHumidAir.Value
        app.EditField_Spec_Humidity.Value = 0;
        app.EditField_Air_H2O_Final.Value = 0;
    else
        T_C = app.SpinnerDryBulbTemperature.Value; % °C
        RH = app.SpinnerRelativeHumidity.Value/100; % (0-1)
        p_atm = app.SpinnerAtmosfericPressure.Value*1000; % kPa→Pa
        p_ws = XSteam('psat_T', T_C)*1e5;% psat em bar → Pa
        w = 0.622 * (RH*p_ws) / (p_atm - RH*p_ws);
        app.EditField_Spec_Humidity.Value = w;
        app.EditField_Air_H2O_Final.Value = w * ...
            str2double( app.EditField_Dry_Air.Value ) ...
            * (app.EditFieldExcessAir.Value + 100) /100;
    end
end
function updateCombustionProducts(app)

```

```

T = app.CollectedFuelData;
if isempty(T) || ~ismember("WeightPct", ...
    T.Properties.VariableNames)
    return
end

T.pc_co2      = T.WeightPct .* T.fm_co2;
app.EditField_CB_CO2.Value = sum(T.pc_co2);
T.pc_h2o      = T.WeightPct .* T.fm_h2o;
app.EditField_CB_H2O.Value = sum(T.pc_h2o) + app.EditField_Air_H2O_Final.Value;

remDryAir    = app.EditField_Dry_Air_Final.Value - app.EditField_O2_Final.Value;
app.EditField_CB_N2.Value = remDryAir;

o2_supplied = app.EditField_O2_Final.Value;
o2_stoich   = str2double(app.EditField_O2.Value);
app.EditField_CB_O2.Value = o2_supplied - o2_stoich;
app.CollectedFuelData = T;
end

function plotCombustionVsExcessAir(app)

T = app.CollectedFuelData;
if isempty(T) || ~ismember('WeightPct', T.Properties.VariableNames)
    return
end
range = app.ExcessofAirSlider.Value;
minExcess = range(1);
maxExcess = range(2);
if maxExcess <= minExcess
    warning('Invalid excess air interval');
    return;
end
excessRange = linspace(minExcess, maxExcess, 1000);
n = numel(excessRange);

CO2 = zeros(1,n);
H2O = zeros(1,n);
N2 = zeros(1,n);
O2 = zeros(1,n);

fm_co2      = T.fm_co2;
fm_h2o      = T.fm_h2o;
fm_o2       = T.fm_o2;
fm_dry_air  = T.fm_dry_air;
WeightPct   = T.WeightPct;
w           = app.EditField_Spec_Humidity.Value;
dryAir_Stoich = str2double(app.EditField_Dry_Air.Value); % kg/kg_fuel

if isnan(w) || isnan(dryAir_Stoich)
    warning('w ou dryAir inválido');
    return
end

for k = 1:n
    excess = excessRange(k);
    factor = (100 + excess)/100;

    O2_total     = sum(WeightPct .* fm_o2) * factor;
    dryAir_total = sum(WeightPct .* fm_dry_air) * factor;
    vapor_air   = w * dryAir_Stoich * factor;

    CO2(k) = sum(WeightPct .* fm_co2);
    H2O(k) = sum(WeightPct .* fm_h2o) + vapor_air;
    N2(k)  = dryAir_total - O2_total;
    O2(k)  = O2_total - sum(WeightPct .* fm_o2);
end

cla(app.UIAxes_CB, 'reset');
hold(app.UIAxes_CB, 'on');
plot(app.UIAxes_CB, excessRange, CO2, '-', 'DisplayName', 'CO_2');
plot(app.UIAxes_CB, excessRange, H2O, '-', 'DisplayName', 'H_2O');
plot(app.UIAxes_CB, excessRange, N2,  '-', 'DisplayName', 'N_2');
plot(app.UIAxes_CB, excessRange, O2,  '-', 'DisplayName', 'O_2');
hold(app.UIAxes_CB, 'off');

xlabel(app.UIAxes_CB, 'Excess Air [%]');
ylabel(app.UIAxes_CB, 'Combustion products [kg/kg_{fuel}]');
legend(app.UIAxes_CB, 'Location', 'northwest');
title(app.UIAxes_CB, '');
grid(app.UIAxes_CB, 'on');
app.UIAxes_CB.XLim = [range(1), range(2)];

end
function computeCurrentFlameTemperature(app)
T = app.CollectedFuelData;
if isempty(T) || ~all(ismember({'WeightPct', 'hf_o', 'm'}, T.Properties.VariableNames))
    return
end

m      = sum(T.m);
n      = m;
hf_o   = sum(T.hf_o);
T_air  = app.SpinnerDryBulbTemperature.Value;
RH    = app.SpinnerRelativeHumidity.Value / 100;
humidity = app.CheckBoxHumidAir.Value;

```

```

Q_in      = app.EditFieldQin.Value;
W_in      = app.EditFieldWin.Value;
Q_out     = app.EditFieldQout.Value;
W_out     = app.EditFieldwout.Value;
exc_air   = (app.EditFieldExcessAir.Value + 100) / 100;

try
    T_p_K = flame_temperature(m, n, hf_o, T_air, exc_air, RH, humidity, Q_in, W_in, Q_out, W_out);
    T_p_C = double(T_p_K - 273.15);
    app.EditFieldT_p.Value = T_p_C;
catch
    app.EditFieldT_p.Value = NaN;
end

plotFlameTemperatureVsExcessAir(app);

function plotFlameTemperatureVsExcessAir(app)
    T = app.CollectedFuelData;
    if isempty(T) || ~all(ismember({'WeightPct', 'hf_o', 'm'}, T.Properties.VariableNames))
        return
    end
    m = sum(T.m);
    n = m;
    hf_o = sum(T.hf_o);
    T_air = app.SpinnerDryBulbTemperature.Value;
    RH = app.SpinnerRelativeHumidity.Value / 100;
    humidity = app.CheckBoxHumidAir.Value;
    Q_in = app.EditFieldQin.Value;
    W_in = app.EditFieldWin.Value;
    Q_out = app.EditFieldQout.Value;
    W_out = app.EditFieldwout.Value;
    range = app.SliderT_p.Value;
    excessRange = linspace(range(1), range(2), 20);
    Tp = zeros(size(excessRange));

    for i = 1:numel(excessRange)
        exc = (excessRange(i) + 100) / 100;
        try
            Tp(i) = flame_temperature(m, n, hf_o, T_air, exc, RH, humidity, Q_in, W_in, Q_out, W_out) - 273.15;
        catch
            Tp(i) = NaN;
        end
    end
    cla(app.UIAxesT_p, 'reset');
    plot(app.UIAxesT_p, excessRange, Tp, 'b', 'LineWidth', 1);
    xlabel(app.UIAxesT_p, 'Excess air [%]');
    ylabel(app.UIAxesT_p, 'Combustion products temperature [°C]');
    grid(app.UIAxesT_p, 'on');
    app.UIAxesT_p.XLim = [range(1), range(2)];
end

function calcOstwaldTriangleTheo(app)

COData = getappdata(0, 'COData');
O2Data = getappdata(0, 'O2Data');
TemperatureData = getappdata(0, 'TemperatureData');

CO_probe = str2double(COData);
O2_probe = str2double(O2Data);
TemperatureValue = str2double(TemperatureData);

if isnan(CO_probe) || isnan(O2_probe) || isnan(TemperatureValue)
    uialert(app.UIFigure, ...
        'Invalid or missing probe data. Please check the input format or connection.', ...
        'Probe Error');
    return;
end

app.EditFieldOstT.Value = TemperatureValue;
T = app.CollectedFuelData;

if isempty(T) || ~ismember("WeightPct", T.Properties.VariableNames)
    return
end

T.pc_inc_co = (T.WeightPct*100./T.molar_mass) * 1;
T.pc_inc_n2 = (T.WeightPct*100./T.molar_mass) .* T.fm_n2;
T.pc_inc_o2 = (T.WeightPct*100./T.molar_mass) * 0.5;

total_co = sum(T.pc_inc_co);
total_n2 = sum(T.pc_inc_n2);
total_o2 = sum(T.pc_inc_o2);
total_sum = total_co + total_n2 + total_o2;

vol_CO_point_C = (total_co / total_sum)*100;
vol_O2_point_C = (total_o2 / total_sum)*100;

app.CollectedFuelData = T;

O2_total = sum(T.WeightPct .* T.fm_o2);
dryAir_total = sum(T.WeightPct .* T.fm_dry_air);
CO2 = sum(T.WeightPct .* T.fm_co2);
H2O = sum(T.WeightPct .* T.fm_h2o);
N2 = dryAir_total - O2_total;
O2 = O2_total - sum(T.WeightPct .* T.fm_o2);
CO2_point_A = (CO2 /(CO2+H2O+N2+O2))*100;

if app.TabGroup.SelectedTab ~= app.OstwaldTriangleTab
    return;
end

```

```

cla(app.UIAxesOstwald, 'reset')

if isnan(CO_probe) || isnan(O2_probe)
    uialert(app.UIFigure, ...
        'No measures available. Please try to use probe dashboard in first place.', ...
        'Missing Probe Data');
    return;
end

Ax = 0;
Ay = CO2_point_A;
Bx = 21;
By = 0;
Cx = vol_O2_point_C;
Cy = 0;

m_ab = (By - Ay) / (Bx - Ax);
b_ab = Ay - m_ab * Ax;

m_perp = -1 / m_ab;
b_perp = Cy - m_perp * Cx;

xD = (b_perp - b_ab) / (m_ab - m_perp);
yD = m_ab * xD + b_ab;

x_CD = [Cx, xD];
y_CD = [Cy, yD];

t = CO_probe / vol_CO_point_C;
xP = (1 - t) * xD + t * Cx;
yP = (1 - t) * yD + t * Cy;

vCD = [xD - Cx, yD - Cy];
v_perp_CD = [-vCD(2), vCD(1)];
m_perp_CD = v_perp_CD(2) / v_perp_CD(1);
b_perp_P = yP - m_perp_CD * xP;

xV = O2_probe;
yV_top = m_ab * xV + b_ab;
yV_bottom = 0;

xK = xV;
yK = m_perp_CD * xK + b_perp_P;
yK = min(yK, yV_top);

yK_limite = m_ab * xK + b_ab;
if yK < 0 || xK < 0 || yK > yK_limite
    uialert(app.UIFigure, ...
        ['The calculated point is out of the valid range for the Ostwald triangle.' newline ...
        'Please check the probe input values (CO and O2).'], ...
        'Invalid data');
    cla(app.UIAxesOstwald, 'reset');
    CO_probe = NaN;
    O2_probe = NaN;
    return;
end

hold(app.UIAxesOstwald, 'on');
plot(app.UIAxesOstwald, [Ax, Bx], [Ay, By], 'k-');
plot(app.UIAxesOstwald, [Ax, Cx], [Ay, Cy], 'b--');
plot(app.UIAxesOstwald, x_CD, y_CD, 'r:');
text(app.UIAxesOstwald, xP + 0.5, yP + 0.5, ...
    ['CO = ', num2str(CO_probe, '%.2f') '%'], ...
    'Color', 'm', 'FontSize', 10);

x_perp = linspace(xP, xK, 100);
y_perp = m_perp_CD * x_perp + b_perp_P;
plot(app.UIAxesOstwald, x_perp, y_perp, 'c-.');

plot(app.UIAxesOstwald, [xV, xV], [yV_bottom, yK], 'g--');

plot(app.UIAxesOstwald, [0, xK], [yK, yK], 'k-.');

vAC = [Cx - Ax, Cy - Ay];
m_AC = vAC(2) / vAC(1);
b_E = yK - m_AC * xK;

xE = (b_ab - b_E) / (m_AC - m_ab);
yE = m_AC * xE + b_E;

plot(app.UIAxesOstwald, [xK, xE], [yK, yE], 'm--');
plot(app.UIAxesOstwald, xE, yE, 'r*', 'MarkerSize');
text(app.UIAxesOstwald, xE + 0.5, yE, ...
    ['E = ', num2str(1 / (dot([xE - Bx, yE - By], [Ax - Bx, Ay - By]) / dot([Ax - Bx, Ay - By], [Ax - Bx, Ay - By])), '%.3f')], ...
    'Color', 'r');

vAB = [Ax - Bx, Ay - By];
vBE = [xE - Bx, yE - By];

```

```

AB_mag2 = dot(vAB, vAB);
lambda = dot(vBE, vAB) / AB_mag2;
Ostwald_E = (1 / lambda)*100 -100;

 xlabel(app.UIAxesOstwald, 'O2 [% vol]');
 ylabel(app.UIAxesOstwald, 'CO2 [% vol]');
 app.UIAxesOstwald.DataAspectRatio = [1 1 1];
 grid(app.UIAxesOstwald, 'on');
 Ostwald_CO2 = yK;

app.EditFieldOstE.Value    = Ostwald_E;
app.EditFieldOstCO2.Value = Ostwald_CO2;
app.EditFieldOstCO.Value  = CO_probe;
app.EditFieldOstO2.Value  = O2_probe;

end

function ClearAllButtonPushed(app, ~)
app.CollectedFuelData = [];
app.NumFuel = 1;
updateFuelFields(app);

for k = 1:numel(app.FuelComponents)
dd = app.FuelComponents(k).DropDown;
ve = app.FuelComponents(k).volEditField;
if ~isempty(dd.Items)
dd.Value = dd.Items{1};
end
ve.Value = [];
ve.BackgroundColor = 'white';
app.FuelComponents(k).formulaLabel.Text = '';
end
app.EditField_O2.Value = '';
app.EditField_Dry_Air.Value = '';
app.EditField_Spec_Humidity.Value = 0;
app.EditField_O2_Final.Value = 0;
app.EditField_Dry_Air_Final.Value = 0;
app.EditField_Air_H2O_Final.Value = 0;
app.EditFieldExcessAir.Value = 0;
app.SpinnerAtmosfericPressure.Value = 101.325;
app.SpinnerRelativeHumidity.Value = 60;
app.SpinnerDryBulbTemperature.Value = 25;
app.EditField_CB_CO2.Value = 0;
app.EditField_CB_H2O.Value = 0;
app.EditField_CB_N2.Value = 0;
app.EditField_CB_O2.Value = 0;
app.EditFieldT_p.Value = 0;
app.EditFieldqin.Value = 0;
app.EditFieldwin.Value = 0;
app.EditFieldqout.Value = 0;
app.EditFieldwout.Value = 0;
app.EditFieldOstE.Value = [];
app.EditFieldOstCO2.Value = [];
app.EditFieldOstCO.Value = [];
app.EditFieldOstO2.Value = [];
app.EditFieldOstT.Value = [];
cla(app.UIAxes_CB, 'reset');
cla(app.UIAxes_T_p, 'reset');
cla(app.UIAxesOstwald, 'reset');

end

end

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function startupFcn(app)

addpath('bin');
app.UIFigure.Name = 'Flame Solver v1.0';
app.UIFigure.Icon = './bin/icon.png';
addpath(fullfile(fileparts(mfilename('fullpath')), 'bin','xsteam'));
clc;

addlistener( app.TabGroup, ...
'SelectionChanged', ...
@(src,ev) onTabChanged(app,ev) );

addlistener(app.EditFieldExcessAir, 'ValueChanged', ...
@(src,ev) updateFinalFields(app));

app.EditFieldExcessAir.Limits = [0 1000];
app.EditFieldExcessAir.LowerLimitInclusive = 'on';
app.EditFieldExcessAir.UpperLimitInclusive = 'on';
app.EditFieldExcessAir.Value = 0;

fcn = createCallbackFcn(app, @updateHumidAir, true);

app.CheckBoxHumidAir.ValueChangedFcn      = fcn;
app.SpinnerAtmosfericPressure.ValueChangedFcn= fcn;
app.SpinnerRelativeHumidity.ValueChangedFcn = fcn;
app.SpinnerDryBulbTemperature.ValueChangedFcn= fcn;

```

```

app.EditField_Dry_Air_Final.ValueChangedFcn = fcn;

app.SpinnerAtmosphericPressure.Enable = 'off';
app.SpinnerRelativeHumidity.Enable = 'off';
app.SpinnerDryBulbTemperature.Enable = 'off';
updateHumidAir(app);

addlistener(app.CheckBoxHumidAir, 'ValueChanged', ...
    @(src,ev) updateHumidAir(app,ev));

addlistener(app.TabGroup, 'SelectionChanged', @(s,e) onTabChanged(app,e));

addlistener(app.ExcessofAirSlider, 'ValueChanged', ...
    @(src, ev) plotCombustionVsExcessAir(app));
addlistener(app.EditFieldQin, 'ValueChanged', @(src,ev) computeCurrentFlameTemperature(app));
addlistener(app.EditFieldWin, 'ValueChanged', @(src,ev) computeCurrentFlameTemperature(app));
addlistener(app.EditFieldQout, 'ValueChanged', @(src,ev) computeCurrentFlameTemperature(app));
addlistener(app.EditFieldWout, 'ValueChanged', @(src,ev) computeCurrentFlameTemperature(app));
addlistener(app.SliderT_P, 'ValueChanged', @(s,e) plotFlameTemperatureVsExcessAir(app));

```

```

sufixes = {'', '_2', '_3', '_4', '_5', '_6'};
app.FuelComponents(numel(sufixes)) = struct( ...
    'DropDown', [], ...
    'volEditField', [], ...
    'volLabel', [], ...
    'formulaLabel', [] );
for k = 1:numel(sufixes)
    app.FuelComponents(k).DropDown = app.(['DropDown' sufixes{k}]);
    app.FuelComponents(k).volEditField = app.(['volEditField' sufixes{k}]);
    app.FuelComponents(k).volLabel = app.(['volLabel' sufixes{k}]);
    app.FuelComponents(k).formulaLabel = app.(['Label' num2str(k)]);
    app.FuelComponents(k).formulaLabel.Visible = 'off';
end

fuelsFile = fullfile(fileparts(mfilename('fullpath')), 'bin', 'fuels.xlsx');
T = readable(fuelsFile);
idCol = T(:,1);
validRows = ~isnan(idCol);
T = T(validRows,:);
vars = T.Properties.VariableNames;
toKeep = false(1,numel(vars));
for k = 1:numel(vars)
    col = T.(vars{k});
    if isnumeric(col)
        toKeep(k) = any(~isnan(col));
    elseif iscellstr(col) || isstring(col)
        toKeep(k) = any(~ismissing(col));
    else
        toKeep(k) = true;
    end
end
T = T(:, vars(toKeep));
app.FuelsTable = T;
names = app.FuelsTable(:,2);
for k = 1:numel(app.FuelComponents)
    dd = app.FuelComponents(k).DropDown;
    dd.Items = names;
    dd.Value = names{1};
    dd.ValueChangedFcn = @(src, ev) onFuelSelected(app, ev);
end
app.NumFuel = 1;
updateFuelFields(app);
for k = 1:numel(app.FuelComponents)
    app.FuelComponents(k).volEditField.Limits = [0, 500];
    app.FuelComponents(k).volEditField.LowerLimitInclusive = 'off';
    app.FuelComponents(k).volEditField.UpperLimitInclusive = 'on';
    app.FuelComponents(k).valueChangedFcn = createCallbackFcn(app, @fuelValueChanged, true);
end
for k = 1:numel(app.FuelComponents)
    app.FuelComponents(k).volEditField.ValueChangedFcn = createCallbackFcn(app, @fuelValueChanged, true);
end
onFuelSelected(app, struct('Source',app.FuelComponents(1).DropDown));

```

end

```

% Menu selected function: AboutMenu
function AboutMenuSelected(app, event)
    run('bin/about.mlapp');
end

% Button pushed function: Button_minus
function Button_minusPushed(app, event)
    app.NumFuel = app.NumFuel - 1;
    updateFuelFields(app);
end

% Button pushed function: Button_plus
function Button_plusPushed(app, event)
    app.NumFuel = app.NumFuel + 1;
    updateFuelFields(app);
end

% Menu selected function: ProbashboardMenu
function ProbashboardMenuSelected(app, event)
    run('bin/probe_dashboard.mlapp');
end

```

```

% Button pushed function: PlotButton
function PlotButtonPushed(app, event)
    calcOstwaldTriangleTheo(app);
end

% Button pushed function: ClearButton
function ClearButtonPushed(app, event)
    cla(app.UIAxesOstwald, 'reset');

    app.EditFieldOstE.Value      = [];
    app.EditFieldOstCO2.Value    = [];
    app.EditFieldOstC0.Value     = [];
    app.EditFieldOstO2.Value     = [];
    app.EditFieldOstT.Value      = [];

end

% Menu selected function: GeneratereportMenu
function GeneratereportMenuSelected(app, event)

import mlreportgen.report.*
import mlreportgen.dom.*

timestamp = datestr(now, 'yyyy-mm-dd_HH-MM-SS');
filename  = fullfile(tempdir, ['FlameSolver_Report_' timestamp '.pdf']);

rpt = Report(filename, 'pdf');
rpt.Layout.Landscape = false;

add(rpt, Paragraph('Flame Solver Report'));
add(rpt, Paragraph(' '));

add(rpt, Paragraph('----- Fuel -----'));
Tfuel = app.CollectedFuelData;
if ~isempty(Tfuel)
    names = Tfuel(:,2);
    vois = Tfuel.VolPct;
    Ttbl = table(names, vois, 'VariableNames', {'Fuel','VolPct'});
    add(rpt, BaseTable(Ttbl));
else
    add(rpt, Paragraph('No fuel data available.'));
end
add(rpt, Paragraph(' '));
add(rpt, Paragraph('----- Combustion Analysis -----'));
vCO2 = app.EditField_CB_CO2.Value;
vH2O = app.EditField_CB_H2O.Value;
vN2 = app.EditField_CB_N2.Value;
vO2 = app.EditField_CB_O2.Value;
if all(isnumeric([vCO2 vH2O vN2 vO2]))
    Tcomb = table( ...
        ['CO2'; 'H2O'; 'N2'; 'O2'], ...
        [vCO2; vH2O; vN2; vO2], ...
        'VariableNames', {'Species','Value'} ...
    );
    add(rpt, BaseTable(Tcomb));
else
    add(rpt, Paragraph('No combustion data available.'));
end
add(rpt, Paragraph(' '));

add(rpt, Paragraph('----- Flame Temperature -----'));
Tp      = app.EditFieldT_p.Value;
Qin    = app.EditFieldQin.Value;
Win    = app.EditFieldWin.Value;
Qout   = app.EditFieldQout.Value;
Wout   = app.EditFieldWout.Value;
if isnumeric(Tp)
    add(rpt, Paragraph(sprintf('Flame Temperature (°C): %.2f', Tp)));
    add(rpt, Paragraph(sprintf('Qin : %.0f', Qin)));
    add(rpt, Paragraph(sprintf('Win : %.0f', Win)));
    add(rpt, Paragraph(sprintf('Qout : %.0f', Qout)));
    add(rpt, Paragraph(sprintf('Wout : %.0f', Wout)));
    fig2 = uifigure('Visible','off'); copyobj(app.UIAxesT_p, fig2);
    img2 = fullfile(tempdir, ['tempcurve_' timestamp '.png']);
    exportgraphics(fig2.Children, img2, 'Resolution',200);
    add(rpt, Image(img2));
else
    add(rpt, Paragraph('No flame temperature available.'));
end
add(rpt, Paragraph(' '));

add(rpt, Paragraph('----- Ostwald Triangle -----'));
co_probe   = app.EditFieldOstC0.Value;
o2_probe   = app.EditFieldOstO2.Value;
co2_k      = app.EditFieldOstCO2.Value;
E_value    = app.EditFieldOstE.Value;
T_probe    = app.EditFieldOstT.Value;

if isprop(app,'vol_CO_point_C') && isprop(app,'vol_O2_point_C')
    Cco = app.vol_CO_point_C;
    Co2 = app.vol_O2_point_C;
    add(rpt, Paragraph(sprintf('Triangle C: CO = %.2f%%, O2 = %.2f%%', Cco, Co2)));
end
add(rpt, Paragraph(sprintf('Probe CO : %g', co_probe)));
add(rpt, Paragraph(sprintf('Probe O2 : %g', o2_probe)));
add(rpt, Paragraph(sprintf('Ostwald CO2: %.4f', co2_k)));

```

```

add(rpt, Paragraph(sprintf('Ostwald Excess of air : %.6f', E_value)));
add(rpt, Paragraph(sprintf('Temperature: %.2f °C', T_probe)));

fig3 = uifigure('Visible','off'); copyobj(app.UIAxesOstwald, fig3);
img3 = fullfile(tempdir, ['ostwald_' timestamp '.png']);
exportgraphics(fig3.Children, img3, 'Resolution',200);
add(rpt, Image(img3));

close(rpt);
rptview(rpt);

end

% Menu selected function: ClearMenu
function ClearMenuSelected(app, event)

app.CollectedFuelData = [];
app.NumFuel = 1;
updateFuelFields(app);

for k = 1:numel(app.FuelComponents)
    dd = app.FuelComponents(k).DropDown;
    ve = app.FuelComponents(k).volEditField;
    if ~isempty(dd.Items)
        dd.Value = dd.Items{1};
    end
    ve.Value = [];
    ve.BackgroundColor = 'white';
    app.FuelComponents(k).formulaLabel.Text = '';
end
app.EditField_O2.Value = '';
app.EditField_Dry_Air.Value = '';
app.EditField_Spec_Humidity.Value = 0;
app.EditField_O2_Final.Value = 0;
app.EditField_Dry_Air_Final.Value = 0;
app.EditField_Air_H2O_Final.Value = 0;
app.EditFieldExcessAir.Value = 0;
app.SpinnerAtmosphericPressure.Value = 101.325;
app.SpinnerRelativeHumidity.Value = 60;
app.SpinnerDryBulbTemperature.Value = 25;
app.EditField_CB_CO2.Value = 0;
app.EditField_CB_H2O.Value = 0;
app.EditField_CB_N2.Value = 0;
app.EditField_CB_O2.Value = 0;
app.EditFieldT_p.Value = 0;
app.EditFieldqin.Value = 0;
app.EditFieldwin.Value = 0;
app.EditFieldQout.Value = 0;
app.EditFieldWout.Value = 0;
app.EditFieldOstE.Value = [];
app.EditFieldOstCO2.Value = [];
app.EditFieldOstCO.Value = [];
app.EditFieldOstO2.Value = [];
app.EditFieldOstT.Value = [];
cla(app.UIAxes_CB, 'reset');
cla(app.UIAxesT_p, 'reset');

end

% Menu selected function: LoadexampleMenu
function LoadexampleMenuSelected(app, event)

ClearAllButtonPushed(app, event);
app.NumFuel = 1;
updateFuelFields(app);
fuel = 'n-Octane';
names = app.FuelComponents(1).DropDown.Items;
idx = find(strcmp(names, fuel), 1);
if isempty(idx)
    uialert(app.UIFigure, 'Fuel not found!', 'Error');
    return;
end
app.FuelComponents(1).DropDown.Value = names{idx};
app.FuelComponents(1).volEditField.Value = 100;
onFuelSelected(app, struct('Source', app.FuelComponents(1).DropDown));
fuelValueChanged(app, struct('Source', app.FuelComponents(1).volEditField));
app.EditFieldExcessAir.Value = 15;
app.CheckBoxHumidAir.Value = true;
app.SpinnerRelativeHumidity.Value = 90;
app.SpinnerDryBulbTemperature.Value = 25;
app.EditFieldQout.Value = 75000;
app.EditFieldWout.Value = 120000;
validateAndCollectFuels(app);
updateFinalFields(app);
updateCombustionProducts(app);
computeCurrentFlameTemperature(app);
plotCombustionVsExcessAir(app);
plotFlameTemperatureVsExcessAir(app);

end

% Menu selected function: ProjectreportMenu
function ProjectreportMenuSelected(app, event)
open('./docs/Project_Report.pdf');

end

% Menu selected function: ExampledocumentationMenu
function ExampledocumentationMenuSelected(app, event)
open('./docs/Example.pdf');

end

% Menu selected function: HelpMenu_3
function HelpMenu_3Selected(app, event)

```

```

        open('./docs/Help.pdf');
    end
end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

        % Create UIFigure and hide until all components are created
        app.UIFigure = uifigure('Visible', 'off');
        app.UIFigure.Position = [100 100 556 308];
        app.UIFigure.Name = 'MATLAB App';

        % Create FileMenu
        app.FileMenu = uimenu(app.UIFigure);
        app.FileMenu.Text = 'File';

        % Create ExampleMenu
        app.ExampleMenu = uimenu(app.FileMenu);
        app.ExampleMenu.Text = 'Example';

        % Create LoadexampleMenu
        app.LoadexampleMenu = uimenu(app.ExampleMenu);
        app.LoadexampleMenu.MenuSelectedFcn = createCallbackFcn(app, @LoadexampleMenuSelected, true);
        app.LoadexampleMenu.Text = 'Load example';

        % Create ExampledocumentationMenu
        app.ExampledocumentationMenu = uimenu(app.ExampleMenu);
        app.ExampledocumentationMenu.MenuSelectedFcn = createCallbackFcn(app, @ExampledocumentationMenuSelected, true);
        app.ExampledocumentationMenu.Text = 'Example documentation';

        % Create GeneratereportMenu
        app.GeneratereportMenu = uimenu(app.FileMenu);
        app.GeneratereportMenu.MenuSelectedFcn = createCallbackFcn(app, @GeneratereportMenuSelected, true);
        app.GeneratereportMenu.Text = 'Generate report';

        % Create ClearMenu
        app.ClearMenu = uimenu(app.FileMenu);
        app.ClearMenu.MenuSelectedFcn = createCallbackFcn(app, @ClearMenuSelected, true);
        app.ClearMenu.Text = 'Clear';

        % Create ProbeMenu
        app.ProbeMenu = uimenu(app.UIFigure);
        app.ProbeMenu.Text = 'Probe';

        % Create ProbashboardMenu
        app.ProbashboardMenu = uimenu(app.ProbeMenu);
        app.ProbashboardMenu.MenuSelectedFcn = createCallbackFcn(app, @ProbashboardMenuSelected, true);
        app.ProbashboardMenu.Text = 'Probe dashboard';

        % Create ProjectreportMenu
        app.ProjectreportMenu = uimenu(app.UIFigure);
        app.ProjectreportMenu.MenuSelectedFcn = createCallbackFcn(app, @ProjectreportMenuSelected, true);
        app.ProjectreportMenu.Text = 'Project report';

        % Create HelpMenu
        app.HelpMenu = uimenu(app.UIFigure);
        app.HelpMenu.Text = 'Help';

        % Create HelpMenu_3
        app.HelpMenu_3 = uimenu(app.HelpMenu);
        app.HelpMenu_3.MenuSelectedFcn = createCallbackFcn(app, @HelpMenu_3Selected, true);
        app.HelpMenu_3.Text = 'Help';

        % Create AboutMenu
        app.AboutMenu = uimenu(app.HelpMenu);
        app.AboutMenu.MenuSelectedFcn = createCallbackFcn(app, @AboutMenuSelected, true);
        app.AboutMenu.Text = 'About...';

        % Create TabGroup
        app.TabGroup = uitabgroup(app.UIFigure);
        app.TabGroup.Position = [1 2 556 307];

        % Create FuelTab
        app.FuelTab = uitab(app.TabGroup);
        app.FuelTab.Title = 'Fuel';

        % Create FuelcomponentsPanel
        app.FuelcomponentsPanel = uipanel(app.FuelTab);
        app.FuelcomponentsPanel.Title = 'Fuel components';
        app.FuelcomponentsPanel.Position = [15 13 531 259];

        % Create GridLayout
        app.GridLayout = uigridlayout(app.FuelcomponentsPanel);
        app.GridLayout.ColumnWidth = {78.94, 80.94, 38.97, '1x', '2.63x', 30.98, '2.71x'};
        app.GridLayout.RowHeight = {'1x', 21.98, 21.98, 21.98, 21.98, 21.98, 26.98};

        % Create Button_plus
        app.Button_plus = uibutton(app.GridLayout, 'push');
        app.Button_plus.ButtonPushedFcn = createCallbackFcn(app, @Button_plusPushed, true);
        app.Button_plus.Layout.Row = 3;
        app.Button_plus.Layout.Column = 7;
        app.Button_plus.Text = '+';

        % Create Button_minus
        app.Button_minus = uibutton(app.GridLayout, 'push');
        app.Button_minus.ButtonPushedFcn = createCallbackFcn(app, @Button_minusPushed, true);
        app.Button_minus.Layout.Row = 2;
        app.Button_minus.Layout.Column = 7;
        app.Button_minus.Text = '-';
    end

```

```

% Create DropDown
app.DropDown = uidropdown(app.GridLayout);
app.DropDown.Layout.Row = 2;
app.DropDown.Layout.Column = [1 2];

% Create DropDown_2
app.DropDown_2 = uidropdown(app.GridLayout);
app.DropDown_2.Layout.Row = 3;
app.DropDown_2.Layout.Column = [1 2];

% Create DropDown_3
app.DropDown_3 = uidropdown(app.GridLayout);
app.DropDown_3.Layout.Row = 4;
app.DropDown_3.Layout.Column = [1 2];

% Create DropDown_4
app.DropDown_4 = uidropdown(app.GridLayout);
app.DropDown_4.Layout.Row = 5;
app.DropDown_4.Layout.Column = [1 2];

% Create DropDown_5
app.DropDown_5 = uidropdown(app.GridLayout);
app.DropDown_5.Layout.Row = 6;
app.DropDown_5.Layout.Column = [1 2];

% Create DropDown_6
app.DropDown_6 = uidropdown(app.GridLayout);
app.DropDown_6.Layout.Row = 7;
app.DropDown_6.Layout.Column = [1 2];

% Create Label_1
app.Label_1 = uilabel(app.GridLayout);
app.Label_1.Layout.Row = 2;
app.Label_1.Layout.Column = 3;
app.Label_1.Interpreter = 'latex';
app.Label_1.Text = '';

% Create Label_2
app.Label_2 = uilabel(app.GridLayout);
app.Label_2.Layout.Row = 3;
app.Label_2.Layout.Column = 3;
app.Label_2.Interpreter = 'latex';
app.Label_2.Text = '';

% Create Label_3
app.Label_3 = uilabel(app.GridLayout);
app.Label_3.Layout.Row = 4;
app.Label_3.Layout.Column = 3;
app.Label_3.Interpreter = 'latex';
app.Label_3.Text = '';

% Create Label_4
app.Label_4 = uilabel(app.GridLayout);
app.Label_4.Layout.Row = 5;
app.Label_4.Layout.Column = 3;
app.Label_4.Interpreter = 'latex';
app.Label_4.Text = '';

% Create Label_5
app.Label_5 = uilabel(app.GridLayout);
app.Label_5.Layout.Row = 6;
app.Label_5.Layout.Column = 3;
app.Label_5.Interpreter = 'latex';
app.Label_5.Text = '';

% Create Label_6
app.Label_6 = uilabel(app.GridLayout);
app.Label_6.Layout.Row = 7;
app.Label_6.Layout.Column = 3;
app.Label_6.Interpreter = 'latex';
app.Label_6.Text = '';

% Create volLabel
app.volLabel = uilabel(app.GridLayout);
app.volLabel.HorizontalAlignment = 'right';
app.volLabel.Layout.Row = 2;
app.volLabel.Layout.Column = 6;
app.volLabel.Interpreter = 'tex';
app.volLabel.Text = '%_{vol}';

% Create volEditField
app.volEditField = uieditfield(app.GridLayout, 'numeric');
app.volEditField.AllowEmpty = 'on';
app.volEditField.Layout.Row = 2;
app.volEditField.Layout.Column = 5;

% Create volLabel_2
app.volLabel_2 = uilabel(app.GridLayout);
app.volLabel_2.HorizontalAlignment = 'right';
app.volLabel_2.Layout.Row = 3;
app.volLabel_2.Layout.Column = 6;
app.volLabel_2.Interpreter = 'tex';
app.volLabel_2.Text = '%_{vol}';

% Create volEditField_2
app.volEditField_2 = uieditfield(app.GridLayout, 'numeric');
app.volEditField_2.AllowEmpty = 'on';
app.volEditField_2.Layout.Row = 3;
app.volEditField_2.Layout.Column = 5;

% Create volLabel_3
app.volLabel_3 = uilabel(app.GridLayout);
app.volLabel_3.HorizontalAlignment = 'right';
app.volLabel_3.Layout.Row = 4;

```

```

app.volLabel_3.Layout.Column = 6;
app.volLabel_3.Interpreter = 'tex';
app.volLabel_3.Text = '%_{vol}';

% Create volEditField_3
app.volEditField_3 = uieditfield(app.GridLayout, 'numeric');
app.volEditField_3.AllowEmpty = 'on';
app.volEditField_3.layout.Row = 4;
app.volEditField_3.layout.Column = 5;

% Create volLabel_4
app.volLabel_4 = uilabel(app.GridLayout);
app.volLabel_4.HorizontalAlignment = 'right';
app.volLabel_4.layout.Row = 5;
app.volLabel_4.layout.Column = 6;
app.volLabel_4.Interpreter = 'tex';
app.volLabel_4.Text = '%_{vol}';

% Create volEditField_4
app.volEditField_4 = uieditfield(app.GridLayout, 'numeric');
app.volEditField_4.AllowEmpty = 'on';
app.volEditField_4.layout.Row = 5;
app.volEditField_4.layout.Column = 5;

% Create volLabel_5
app.volLabel_5 = uilabel(app.GridLayout);
app.volLabel_5.HorizontalAlignment = 'right';
app.volLabel_5.layout.Row = 6;
app.volLabel_5.layout.Column = 6;
app.volLabel_5.Interpreter = 'tex';
app.volLabel_5.Text = '%_{vol}';

% Create volEditField_5
app.volEditField_5 = uieditfield(app.GridLayout, 'numeric');
app.volEditField_5.AllowEmpty = 'on';
app.volEditField_5.layout.Row = 6;
app.volEditField_5.layout.Column = 5;

% Create volLabel_6
app.volLabel_6 = uilabel(app.GridLayout);
app.volLabel_6.HorizontalAlignment = 'right';
app.volLabel_6.layout.Row = 7;
app.volLabel_6.layout.Column = 6;
app.volLabel_6.Interpreter = 'tex';
app.volLabel_6.Text = '%_{vol}';

% Create volEditField_6
app.volEditField_6 = uieditfield(app.GridLayout, 'numeric');
app.volEditField_6.AllowEmpty = 'on';
app.volEditField_6.layout.Row = 7;
app.volEditField_6.layout.Column = 5;

% Create AirTab
app.AirTab = uitab(app.TabGroup);
app.AirTab.Title = 'Air';

% Create ExcessairPanel
app.ExcessairPanel = uipanel(app.AirTab);
app.ExcessairPanel.Title = 'Excess air';
app.ExcessairPanel.Position = [229 173 316 99];

% Create GridLayout5
app.GridLayout5 = uigridlayout(app.ExcessairPanel);
app.GridLayout5.ColumnWidth = {'1.02x', 24.97, '1x', 24.97};
app.GridLayout5.RowHeaders = {21.96, 21.96};

% Create EditFieldExcessAir
app.EditFieldExcessAir = uieditfield(app.GridLayout5, 'numeric');
app.EditFieldExcessAir.layout.Row = 1;
app.EditFieldExcessAir.layout.Column = 1;

% Create Label_7
app.Label_7 = uilabel(app.GridLayout5);
app.Label_7.layout.Row = 1;
app.Label_7.layout.Column = 2;
app.Label_7.Text = '%';

% Create IncludeairhumidityinthestudyLabel
app.IncludeairhumidityinthestudyLabel = uilabel(app.GridLayout5);
app.IncludeairhumidityinthestudyLabel.layout.Row = 2;
app.IncludeairhumidityinthestudyLabel.layout.Column = [1 3];
app.IncludeairhumidityinthestudyLabel.Text = 'Include air humidity in the study?';

% Create CheckBoxHumidAir
app.CheckBoxHumidAir = uicheckbox(app.GridLayout5);
app.CheckBoxHumidAir.Text = '';
app.CheckBoxHumidAir.layout.Row = 2;
app.CheckBoxHumidAir.layout.Column = 4;

% Create StoichiometricairPanel
app.StoichiometricairPanel = uipanel(app.AirTab);
app.StoichiometricairPanel.Title = 'Stoichiometric air';
app.StoichiometricairPanel.Position = [15 173 205 99];

% Create GridLayout3
app.GridLayout3 = uigridlayout(app.StoichiometricairPanel);
app.GridLayout3.ColumnWidth = {24.96, 26.95, '1x', 54.9};
app.GridLayout3.RowHeaders = {21.96, 21.96};
app.GridLayout3.ColumnSpacing = 6.41160583496094;
app.GridLayout3.Padding = [6.41160583496094 10 6.41160583496094 10];

% Create O_2Label
app.O_2Label = uilabel(app.GridLayout3);
app.O_2Label.layout.Row = 1;

```

```

app.O_2Label.Layout.Column = 1;
app.O_2Label.Interpreter = 'tex';
app.O_2Label.Text = 'O_2';

% Create kgkg_fuellabel
app.kgkg_fuellabel = uilabel(app.GridLayout3);
app.kgkg_fuellabel.Layout.Row = 1;
app.kgkg_fuellabel.Layout.Column = 4;
app.kgkg_fuellabel.Interpreter = 'tex';
app.kgkg_fuellabel.Text = 'kg/kg_{fuel}';

% Create EditField_O2
app.EditField_O2 = uieditfield(app.GridLayout3, 'text');
app.EditField_O2.Editable = 'off';
app.EditField_O2.HorizontalAlignment = 'right';
app.EditField_O2.Layout.Row = 1;
app.EditField_O2.Layout.Column = 3;

% Create DryairLabel
app.Dryairlabel = uilabel(app.GridLayout3);
app.DryairLabel.Layout.Row = 2;
app.DryairLabel.Layout.Column = [1 2];
app.DryairLabel.Text = 'Dry air';

% Create kgkg_fuellabel_2
app.kgkg_fuellabel_2 = uilabel(app.GridLayout3);
app.kgkg_fuellabel_2.Layout.Row = 2;
app.kgkg_fuellabel_2.Layout.Column = 4;
app.kgkg_fuellabel_2.Interpreter = 'tex';
app.kgkg_fuellabel_2.Text = 'kg/kg_{fuel}';

% Create EditField_Dry_Air
app.EditField_Dry_Air = uieditfield(app.GridLayout3, 'text');
app.EditField_Dry_Air.Editable = 'off';
app.EditField_Dry_Air.HorizontalAlignment = 'right';
app.EditField_Dry_Air.Layout.Row = 2;
app.EditField_Dry_Air.Layout.Column = 3;

% Create CombustionairPanel
app.CombustionairPanel = uipanel(app.AirTab);
app.CombustionairPanel.Title = 'Combustion air';
app.CombustionairPanel.Position = [15 14 205 160];

% Create GridLayout2
app.GridLayout2 = uigridlayout(app.CombustionairPanel);
app.GridLayout2.ColumnWidth = {32.94, 18.97, '1x', 54.9};
app.GridLayout2.RowHeaders = {'1x', 21.96, 21.96, 21.96, '1.61x'};
app.GridLayout2.ColumnSpacing = 6.41161041259766;
app.GridLayout2.Padding = [6.41161041259766 10 6.41161041259766 10];

% Create O_2Label_2
app.O_2Label_2 = uilabel(app.GridLayout2);
app.O_2Label_2.Layout.Row = 2;
app.O_2Label_2.Layout.Column = 1;
app.O_2Label_2.Interpreter = 'tex';
app.O_2Label_2.Text = 'O_2';

% Create kgkg_fuellabel_3
app.kgkg_fuellabel_3 = uilabel(app.GridLayout2);
app.kgkg_fuellabel_3.Layout.Row = 2;
app.kgkg_fuellabel_3.Layout.Column = 4;
app.kgkg_fuellabel_3.Interpreter = 'tex';
app.kgkg_fuellabel_3.Text = 'kg/kg_{fuel}';

% Create DryairLabel_2
app.DryairLabel_2 = uilabel(app.GridLayout2);
app.DryairLabel_2.Layout.Row = 3;
app.DryairLabel_2.Layout.Column = [1 2];
app.DryairLabel_2.Text = 'Dry air';

% Create kgkg_fuellabel_4
app.kgkg_fuellabel_4 = uilabel(app.GridLayout2);
app.kgkg_fuellabel_4.Layout.Row = 3;
app.kgkg_fuellabel_4.Layout.Column = 4;
app.kgkg_fuellabel_4.Interpreter = 'tex';
app.kgkg_fuellabel_4.Text = 'kg/kg_{fuel}';

% Create H_20Label_2
app.H_20Label_2 = uilabel(app.GridLayout2);
app.H_20Label_2.Layout.Row = 4;
app.H_20Label_2.Layout.Column = 1;
app.H_20Label_2.Interpreter = 'tex';
app.H_20Label_2.Text = 'H_20';

% Create kgkg_fuellabel_9
app.kgkg_fuellabel_9 = uilabel(app.GridLayout2);
app.kgkg_fuellabel_9.Layout.Row = 4;
app.kgkg_fuellabel_9.Layout.Column = 4;
app.kgkg_fuellabel_9.Interpreter = 'tex';
app.kgkg_fuellabel_9.Text = 'kg/kg_{fuel}';

% Create EditField_O2_Final
app.EditField_O2_Final = uieditfield(app.GridLayout2, 'numeric');
app.EditField_O2_Final.ValueDisplayFormat = '%.3f';
app.EditField_O2_Final.Editable = 'off';
app.EditField_O2_Final.Layout.Row = 2;
app.EditField_O2_Final.Layout.Column = 3;

% Create EditField_Dry_Air_Final
app.EditField_Dry_Air_Final = uieditfield(app.GridLayout2, 'numeric');
app.EditField_Dry_Air_Final.ValueDisplayFormat = '%.3f';
app.EditField_Dry_Air_Final.Editable = 'off';
app.EditField_Dry_Air_Final.Layout.Row = 3;
app.EditField_Dry_Air_Final.Layout.Column = 3;

```

```

% Create EditField_Air_H2O_Final
app.EditField_Air_H2O_Final = uieditfield(app.GridLayout2, 'numeric');
app.EditField_Air_H2O_Final.ValueDisplayFormat = '%.3f';
app.EditField_Air_H2O_Final.Editable = 'off';
app.EditField_Air_H2O_Final.Layout.Row = 4;
app.EditField_Air_H2O_Final.Layout.Column = 3;

% Create AirpropertiesPanel
app.AirpropertiesPanel = uipanel(app.AirTab);
app.AirpropertiesPanel.Title = 'Air properties';
app.AirpropertiesPanel.Position = [229 14 316 160];

% Create GridLayout4
app.GridLayout4 = uigridlayout(app.AirpropertiesPanel);
app.GridLayout4.ColumnWidth = {96.87, 16.97, 85, 71.97};
app.GridLayout4.RowHeight = {21.97, 21.97, 21.97, 21.97, '1x'};

% Create AtmosphericpressureLabel
app.AtmosphericpressureLabel = uilabel(app.GridLayout4);
app.AtmosphericpressureLabel.Layout.Row = 1;
app.AtmosphericpressureLabel.Layout.Column = [1 2];
app.AtmosphericpressureLabel.Text = 'Atmospheric pressure';

% Create kPaLabel
app.kPaLabel = uilabel(app.GridLayout4);
app.kPaLabel.Layout.Row = 1;
app.kPaLabel.Layout.Column = 4;
app.kPaLabel.Text = 'kPa';

% Create RelativehumidityLabel
app.RelativehumidityLabel = uilabel(app.GridLayout4);
app.RelativehumidityLabel.Layout.Row = 2;
app.RelativehumidityLabel.Layout.Column = 1;
app.RelativehumidityLabel.Text = 'Relative humidity';

% Create Label_8
app.Label_8 = uilabel(app.GridLayout4);
app.Label_8.Layout.Row = 2;
app.Label_8.Layout.Column = 4;
app.Label_8.Text = '%';

% Create DrybulbtemperatureLabel
app.DrybulbtemperatureLabel = uilabel(app.GridLayout4);
app.DrybulbtemperatureLabel.Layout.Row = 3;
app.DrybulbtemperatureLabel.Layout.Column = [1 2];
app.DrybulbtemperatureLabel.Text = 'Dry bulb temperature';

% Create CLabel
app.CLabel = uilabel(app.GridLayout4);
app.CLabel.Layout.Row = 3;
app.CLabel.Layout.Column = 4;
app.CLabel.Text = '°C';

% Create SpinnerAtmosfericPressure
app.SpinnerAtmosfericPressure = uispinner(app.GridLayout4);
app.SpinnerAtmosfericPressure.ValueDisplayFormat = '%.3f';
app.SpinnerAtmosfericPressure.Layout.Row = 1;
app.SpinnerAtmosfericPressure.Layout.Column = 3;
app.SpinnerAtmosfericPressure.Value = 101.325;

% Create SpinnerRelativeHumidity
app.SpinnerRelativeHumidity = uispinner(app.GridLayout4);
app.SpinnerRelativeHumidity.ValueDisplayFormat = '%.2f';
app.SpinnerRelativeHumidity.Layout.Row = 2;
app.SpinnerRelativeHumidity.Layout.Column = 3;
app.SpinnerRelativeHumidity.Value = 60;

% Create SpinnerDryBulbTemperature
app.SpinnerDryBulbTemperature = uispinner(app.GridLayout4);
app.SpinnerDryBulbTemperature.ValueDisplayFormat = '%.2f';
app.SpinnerDryBulbTemperature.Layout.Row = 3;
app.SpinnerDryBulbTemperature.Layout.Column = 3;
app.SpinnerDryBulbTemperature.Value = 25;

% Create EditField_Spec_Humidity
app.EditField_Spec_Humidity = uieditfield(app.GridLayout4, 'numeric');
app.EditField_Spec_Humidity.ValueDisplayFormat = '%.4f';
app.EditField_Spec_Humidity.Editable = 'off';
app.EditField_Spec_Humidity.Layout.Row = 4;
app.EditField_Spec_Humidity.Layout.Column = 3;

% Create SpecifichumidityLabel
app.SpecifichumidityLabel = uilabel(app.GridLayout4);
app.SpecifichumidityLabel.Layout.Row = 4;
app.SpecifichumidityLabel.Layout.Column = [1 2];
app.SpecifichumidityLabel.Text = 'Specific humidity';

% Create kg_H_20kg_airLabel
app.kg_H_20kg_airLabel = uilabel(app.GridLayout4);
app.kg_H_20kg_airLabel.Layout.Row = 4;
app.kg_H_20kg_airLabel.Layout.Column = 4;
app.kg_H_20kg_airLabel.Interpreter = 'tex';
app.kg_H_20kg_airLabel.Text = 'kg_{H\_20}/kg_{air}';

% Create CombustionanalysisTab
app.CombustionanalysisTab = uitab(app.TabGroup);
app.CombustionanalysisTab.Title = 'Combustion analysis';

% Create GridLayout9
app.GridLayout9 = uigridlayout(app.CombustionanalysisTab);
app.GridLayout9.ColumnWidth = {'1x', '1.29x'};
app.GridLayout9.RowHeight = {'1x'};

```

```

% Create CombustionproductsPanel
app.CombustionproductsPanel = uipanel(app.GridLayout9);
app.CombustionproductsPanel.Title = 'Combustion products';
app.CombustionproductsPanel.Layout.Row = 1;
app.CombustionproductsPanel.Layout.Column = 1;

% Create GridLayout6
app.GridLayout6 = uigridlayout(app.CombustionproductsPanel);
app.GridLayout6.ColumnWidth = {32.94, '1x', 54.9};
app.GridLayout6.RowHeight = {'1x', 21.96, 21.96, 21.96, 21.96, '1.14x'};
app.GridLayout6.ColumnSpacing = 14.5220136642456;
app.GridLayout6.Padding = [14.5220136642456 10 14.5220136642456 10];

% Create CO_2Label
app.CO_2Label = uilabel(app.GridLayout6);
app.CO_2Label.Layout.Row = 2;
app.CO_2Label.Layout.Column = 1;
app.CO_2Label.Interpreter = 'tex';
app.CO_2Label.Text = 'CO_2';

% Create kgkg_fuelLabel_5
app.kgkg_fuellabel_5 = uilabel(app.GridLayout6);
app.kgkg_fuellabel_5.Layout.Row = 2;
app.kgkg_fuellabel_5.Layout.Column = 3;
app.kgkg_fuellabel_5.Interpreter = 'tex';
app.kgkg_fuellabel_5.Text = 'kg/kg_{fuel}';

% Create H_20Label
app.H_20Label = uilabel(app.GridLayout6);
app.H_20Label.Layout.Row = 3;
app.H_20Label.Layout.Column = 1;
app.H_20Label.Interpreter = 'tex';
app.H_20Label.Text = 'H_20';

% Create kgkg_fuelLabel_6
app.kgkg_fuellabel_6 = uilabel(app.GridLayout6);
app.kgkg_fuellabel_6.Layout.Row = 3;
app.kgkg_fuellabel_6.Layout.Column = 3;
app.kgkg_fuellabel_6.Interpreter = 'tex';
app.kgkg_fuellabel_6.Text = 'kg/kg_{fuel}';

% Create N_2Label
app.N_2Label = uilabel(app.GridLayout6);
app.N_2Label.Layout.Row = 4;
app.N_2Label.Layout.Column = 1;
app.N_2Label.Interpreter = 'tex';
app.N_2Label.Text = 'N_2';

% Create kgkg_fuelLabel_7
app.kgkg_fuellabel_7 = uilabel(app.GridLayout6);
app.kgkg_fuellabel_7.Layout.Row = 4;
app.kgkg_fuellabel_7.Layout.Column = 3;
app.kgkg_fuellabel_7.Interpreter = 'tex';
app.kgkg_fuellabel_7.Text = 'kg/kg_{fuel}';

% Create O_2Label_3
app.O_2Label_3 = uilabel(app.GridLayout6);
app.O_2Label_3.Layout.Row = 5;
app.O_2Label_3.Layout.Column = 1;
app.O_2Label_3.Interpreter = 'tex';
app.O_2Label_3.Text = 'O_2';

% Create kgkg_fuelLabel_8
app.kgkg_fuellabel_8 = uilabel(app.GridLayout6);
app.kgkg_fuellabel_8.Layout.Row = 5;
app.kgkg_fuellabel_8.Layout.Column = 3;
app.kgkg_fuellabel_8.Interpreter = 'tex';
app.kgkg_fuellabel_8.Text = 'kg/kg_{fuel}';

% Create EditField_CB_CO2
app.EditField_CB_CO2 = uieditfield(app.GridLayout6, 'numeric');
app.EditField_CB_CO2.ValueDisplayFormat = '%.3f';
app.EditField_CB_CO2.Editable = 'off';
app.EditField_CB_CO2.Layout.Row = 2;
app.EditField_CB_CO2.Layout.Column = 2;

% Create EditField_CB_H20
app.EditField_CB_H20 = uieditfield(app.GridLayout6, 'numeric');
app.EditField_CB_H20.ValueDisplayFormat = '%.3f';
app.EditField_CB_H20.Editable = 'off';
app.EditField_CB_H20.Layout.Row = 3;
app.EditField_CB_H20.Layout.Column = 2;

% Create EditField_CB_N2
app.EditField_CB_N2 = uieditfield(app.GridLayout6, 'numeric');
app.EditField_CB_N2.ValueDisplayFormat = '%.3f';
app.EditField_CB_N2.Editable = 'off';
app.EditField_CB_N2.Layout.Row = 4;
app.EditField_CB_N2.Layout.Column = 2;

% Create EditField_CB_O2
app.EditField_CB_O2 = uieditfield(app.GridLayout6, 'numeric');
app.EditField_CB_O2.ValueDisplayFormat = '%.3f';
app.EditField_CB_O2.Editable = 'off';
app.EditField_CB_O2.Layout.Row = 5;
app.EditField_CB_O2.Layout.Column = 2;

% Create CombustionproductsExcessofairPanel
app.CombustionproductsExcessofairPanel = uipanel(app.GridLayout9);
app.CombustionproductsExcessofairPanel.Title = 'Combustion products vs. Excess of air';
app.CombustionproductsExcessofairPanel.Layout.Row = 1;
app.CombustionproductsExcessofairPanel.Layout.Column = 2;

% Create UIAxes_CB

```

```

app.UIAxes_CB = uiaxes(app.CombustionproductsVsExcessofairPanel);
xlabel(app.UIAxes_CB, 'X')
ylabel(app.UIAxes_CB, 'Y')
zlabel(app.UIAxes_CB, 'Z')
app.UIAxes_CB.Position = [9 9 229 225];

% Create ExcessofAirSlider
app.ExcessofAirSlider = uislider(app.CombustionproductsVsExcessofairPanel, 'range');
app.ExcessofAirSlider.Limits = [0 500];
app.ExcessofAirSlider.Orientation = 'vertical';
app.ExcessofAirSlider.Position = [253 64 3 150];
app.ExcessofAirSlider.Value = [0 200];

% Create FlameTemperatureTab
app.FlameTemperatureTab = uitab(app.TabGroup);
app.FlameTemperatureTab.Title = 'Flame temperature';

% Create GridLayout8
app.GridLayout8 = uigridlayout(app.FlameTemperatureTab);
app.GridLayout8.ColumnWidth = {200, '1x'};
app.GridLayout8.RowHeight = {'1x', 90};

% Create EnergyanalysysPanel
app.EnergyanalysysPanel = uipanel(app.GridLayout8);
app.EnergyanalysysPanel.Title = 'Energy analysys';
app.EnergyanalysysPanel.Layout.Row = 1;
app.EnergyanalysysPanel.Layout.Column = 1;

% Create GridLayout5_2
app.GridLayout5_2 = uigridlayout(app.EnergyanalysysPanel);
app.GridLayout5_2.ColumnWidth = {27.97, 73, 62.97};
app.GridLayout5_2.RowHeight = {21.96, 21.96, 21.96, 21.96};

% Create kJkmol_fuelLabel
app.kJkmol_fuelLabel = uilabel(app.GridLayout5_2);
app.kJkmol_fuelLabel.Layout.Row = 1;
app.kJkmol_fuelLabel.Layout.Column = 3;
app.kJkmol_fuelLabel.Interpreter = 'tex';
app.kJkmol_fuelLabel.Text = 'kJ/kmol_{fuel}';

% Create Q_inLabel
app.Q_inLabel = uilabel(app.GridLayout5_2);
app.Q_inLabel.Layout.Row = 1;
app.Q_inLabel.Layout.Column = 1;
app.Q_inLabel.Interpreter = 'tex';
app.Q_inLabel.Text = 'Q_{in}';

% Create W_inLabel
app.W_inLabel = uilabel(app.GridLayout5_2);
app.W_inLabel.Layout.Row = 2;
app.W_inLabel.Layout.Column = 1;
app.W_inLabel.Interpreter = 'tex';
app.W_inLabel.Text = 'W_{in}';

% Create Q_outLabel
app.Q_outLabel = uilabel(app.GridLayout5_2);
app.Q_outLabel.Layout.Row = 3;
app.Q_outLabel.Layout.Column = 1;
app.Q_outLabel.Interpreter = 'tex';
app.Q_outLabel.Text = 'Q_{out}';

% Create W_outLabel
app.W_outLabel = uilabel(app.GridLayout5_2);
app.W_outLabel.Layout.Row = 4;
app.W_outLabel.Layout.Column = 1;
app.W_outLabel.Interpreter = 'tex';
app.W_outLabel.Text = 'W_{out}';

% Create EditFieldQin
app.EditFieldQin = uieditfield(app.GridLayout5_2, 'numeric');
app.EditFieldQin.Limits = [0 500000];
app.EditFieldQin.ValueDisplayFormat = '%.0f';
app.EditFieldQin.Layout.Row = 1;
app.EditFieldQin.Layout.Column = 2;

% Create EditFieldWin
app.EditFieldWin = uieditfield(app.GridLayout5_2, 'numeric');
app.EditFieldWin.Limits = [0 500000];
app.EditFieldWin.ValueDisplayFormat = '%.0f';
app.EditFieldWin.Layout.Row = 2;
app.EditFieldWin.Layout.Column = 2;

% Create EditFieldQout
app.EditFieldQout = uieditfield(app.GridLayout5_2, 'numeric');
app.EditFieldQout.Limits = [0 500000];
app.EditFieldQout.ValueDisplayFormat = '%.0f';
app.EditFieldQout.Layout.Row = 3;
app.EditFieldQout.Layout.Column = 2;

% Create EditFieldWout
app.EditFieldWout = uieditfield(app.GridLayout5_2, 'numeric');
app.EditFieldWout.Limits = [0 500000];
app.EditFieldWout.ValueDisplayFormat = '%.0f';
app.EditFieldWout.Layout.Row = 4;
app.EditFieldWout.Layout.Column = 2;

% Create kJkmol_fuelLabel_2
app.kJkmol_fuelLabel_2 = uilabel(app.GridLayout5_2);
app.kJkmol_fuelLabel_2.Layout.Row = 2;
app.kJkmol_fuelLabel_2.Layout.Column = 3;
app.kJkmol_fuelLabel_2.Interpreter = 'tex';
app.kJkmol_fuelLabel_2.Text = 'kJ/kmol_{fuel}';

% Create kJkmol_fuelLabel_3

```

```

app.kJkmol_fuelLabel_3 = uilabel(app.GridLayout5_2);
app.kJkmol_fuelLabel_3.Layout.Row = 3;
app.kJkmol_fuelLabel_3.Layout.Column = 3;
app.kJkmol_fuelLabel_3.Interpreter = 'tex';
app.kJkmol_fuelLabel_3.Text = 'kJ/kmol_{fuel}';

% Create kJkmol_fuelLabel_4
app.kJkmol_fuelLabel_4 = uilabel(app.GridLayout5_2);
app.kJkmol_fuelLabel_4.Layout.Row = 4;
app.kJkmol_fuelLabel_4.Layout.Column = 3;
app.kJkmol_fuelLabel_4.Interpreter = 'tex';
app.kJkmol_fuelLabel_4.Text = 'kJ/kmol_{fuel}';

% Create CombustionproductstemperaturePanel
app.CombustionproductstemperaturePanel = uipanel(app.GridLayout8);
app.CombustionproductstemperaturePanel.Title = 'Combustion products temperature';
app.CombustionproductstemperaturePanel.Layout.Row = 2;
app.CombustionproductstemperaturePanel.Layout.Column = 1;

% Create GridLayout10
app.GridLayout10 = uigridlayout(app.CombustionproductstemperaturePanel);
app.GridLayout10.ColumnWidth = {'1x', '1x', '1x'};
app.GridLayout10.RowHeight = {4, 21.96};

% Create EditFieldT_p
app.EditFieldT_p = uieditfield(app.GridLayout10, 'numeric');
app.EditFieldT_p.ValueDisplayFormat = '%.2f';
app.EditFieldT_p.Editable = 'off';
app.EditFieldT_p.Layout.Row = 2;
app.EditFieldT_p.Layout.Column = 2;

% Create CLabel_2
app.CLabel_2 = uilabel(app.GridLayout10);
app.CLabel_2.Layout.Row = 2;
app.CLabel_2.Layout.Column = 3;
app.CLabel_2.Text = '°C';

% Create T_pLabel
app.T_pLabel = uilabel(app.GridLayout10);
app.T_pLabel.Layout.Row = 2;
app.T_pLabel.Layout.Column = 1;
app.T_pLabel.Interpreter = 'tex';
app.T_pLabel.Text = 'T_p';

% Create CombustionproductstemperaturevsExcessofairPanel
app.CombustionproductstemperaturevsExcessofairPanel = uipanel(app.GridLayout8);
app.CombustionproductstemperaturevsExcessofairPanel.Title = 'Combustion products temperature vs. Excess of air';
app.CombustionproductstemperaturevsExcessofairPanel.Layout.Row = [1 2];
app.CombustionproductstemperaturevsExcessofairPanel.Layout.Column = 2;

% Create UIAxesT_p
app.UIAxesT_p = uiaxes(app.CombustionproductstemperaturevsExcessofairPanel);
xlabel(app.UIAxesT_p, 'X')
ylabel(app.UIAxesT_p, 'Y')
zlabel(app.UIAxesT_p, 'Z')
app.UIAxesT_p.Position = [10 9 229 200];

% Create SliderT_p
app.SliderT_p = uislider(app.CombustionproductstemperaturevsExcessofairPanel, 'range');
app.SliderT_p.Limits = [0 500];
app.SliderT_p.Orientation = 'vertical';
app.SliderT_p.Position = [282 65 3 150];
app.SliderT_p.Value = [0 500];

% Create OstwaldTriangleTab
app.OstwaldTriangleTab = uitab(app.TabGroup);
app.OstwaldTriangleTab.Title = 'Ostwald Triangle';

% Create GridLayout11
app.GridLayout11 = uigridlayout(app.OstwaldTriangleTab);
app.GridLayout11.ColumnWidth = {230, '1x'};
app.GridLayout11.RowHeight = {'18.5x'};

% Create CombustionproductsfromprobePanel
app.CombustionproductsfromprobePanel = uipanel(app.GridLayout11);
app.CombustionproductsfromprobePanel.Title = 'Combustion products from probe';
app.CombustionproductsfromprobePanel.Layout.Row = 1;
app.CombustionproductsfromprobePanel.Layout.Column = 1;

% Create GridLayout5_3
app.GridLayout5_3 = uigridlayout(app.CombustionproductsfromprobePanel);
app.GridLayout5_3.ColumnWidth = {88.97, 73, 25.97};
app.GridLayout5_3.RowHeight = {21.96, 21.96, 21.96, 21.96, 21.96, 21.96, 21.96};

% Create vol_0_2Label
app.vol_0_2Label = uilabel(app.GridLayout5_3);
app.vol_0_2Label.Layout.Row = 1;
app.vol_0_2Label.Layout.Column = 1;
app.vol_0_2Label.Interpreter = 'tex';
app.vol_0_2Label.Text = '%vol_{0_2}';

% Create EditFieldOst02
app.EditFieldOst02 = uieditfield(app.GridLayout5_3, 'numeric');
app.EditFieldOst02.Limits = [0 500000];
app.EditFieldOst02.ValueDisplayFormat = '%.4f';
app.EditFieldOst02.AllowEmpty = 'on';
app.EditFieldOst02.Layout.Row = 1;
app.EditFieldOst02.Layout.Column = 2;
app.EditFieldOst02.Value = [];

% Create EditFieldOstc0
app.EditFieldOstc0 = uieditfield(app.GridLayout5_3, 'numeric');
app.EditFieldOstc0.Limits = [0 500000];
app.EditFieldOstc0.ValueDisplayFormat = '%.4f';

```

```

app.EditFieldOstC0.AllowEmpty = 'on';
app.EditFieldOstC0.Layout.Row = 2;
app.EditFieldOstC0.Layout.Column = 2;
app.EditFieldOstC0.Value = [];

% Create EditFieldOstC02
app.EditFieldOstC02 = uieditfield(app.GridLayout5_3, 'numeric');
app.EditFieldOstC02.Limits = [0 500000];
app.EditFieldOstC02.ValueDisplayFormat = '%.4f';
app.EditFieldOstC02.AllowEmpty = 'on';
app.EditFieldOstC02.Layout.Row = 3;
app.EditFieldOstC02.Layout.Column = 2;
app.EditFieldOstC02.Value = [];

% Create EditFieldOstE
app.EditFieldOstE = uieditfield(app.GridLayout5_3, 'numeric');
app.EditFieldOstE.Limits = [0 50000];
app.EditFieldOstE.ValueDisplayFormat = '%.4f';
app.EditFieldOstE.AllowEmpty = 'on';
app.EditFieldOstE.Layout.Row = 4;
app.EditFieldOstE.Layout.Column = 2;
app.EditFieldOstE.Value = [];

% Create EditFieldOstT
app.EditFieldOstT = uieditfield(app.GridLayout5_3, 'numeric');
app.EditFieldOstT.Limits = [0 500000];
app.EditFieldOstT.ValueDisplayFormat = '%.2f';
app.EditFieldOstT.AllowEmpty = 'on';
app.EditFieldOstT.Layout.Row = 5;
app.EditFieldOstT.Layout.Column = 2;
app.EditFieldOstT.Value = [];

% Create CLabel_3
app.CLabel_3 = uilabel(app.GridLayout5_3);
app.CLabel_3.Layout.Row = 5;
app.CLabel_3.Layout.Column = 3;
app.CLabel_3.Text = '°C';

% Create vol_COLabel
app.vol_COLabel = uilabel(app.GridLayout5_3);
app.vol_COLabel.Layout.Row = 2;
app.vol_COLabel.Layout.Column = 1;
app.vol_COLabel.Interpreter = 'tex';
app.vol_COLabel.Text = '%vol_{CO}';

% Create vol_CO_2Label
app.vol_CO_2Label = uilabel(app.GridLayout5_3);
app.vol_CO_2Label.Layout.Row = 3;
app.vol_CO_2Label.Layout.Column = 1;
app.vol_CO_2Label.Interpreter = 'tex';
app.vol_CO_2Label.Text = '%vol_{CO_2}';

% Create ExcessofairLabel
app.ExcessofairLabel = uilabel(app.GridLayout5_3);
app.ExcessofairLabel.Layout.Row = 4;
app.ExcessofairLabel.Layout.Column = 1;
app.ExcessofairLabel.Text = 'Excess of air';

% Create Label_9
app.Label_9 = uilabel(app.GridLayout5_3);
app.Label_9.Layout.Row = 4;
app.Label_9.Layout.Column = 3;
app.Label_9.Text = '%';

% Create TemperatureLabel
app.TemperatureLabel = uilabel(app.GridLayout5_3);
app.TemperatureLabel.Layout.Row = 5;
app.TemperatureLabel.Layout.Column = 1;
app.TemperatureLabel.Text = 'Temperature';

% Create PlotButton
app.PlotButton = uibutton(app.GridLayout5_3, 'push');
app.PlotButton.ButtonPushedFcn = createCallbackFcn(app, @PlotButtonPushed, true);
app.PlotButton.Layout.Row = 6;
app.PlotButton.Layout.Column = [1 3];
app.PlotButton.Text = 'Plot';

% Create ClearButton
app.ClearButton = uibutton(app.GridLayout5_3, 'push');
app.ClearButton.ButtonPushedFcn = createCallbackFcn(app, @ClearButtonPushed, true);
app.ClearButton.Layout.Row = 7;
app.ClearButton.Layout.Column = [1 3];
app.ClearButton.Text = 'Clear';

% Create OstwaldTrianglePanel
app.OstwaldTrianglePanel = uipanel(app.GridLayout11);
app.OstwaldTrianglePanel.Title = 'Ostwald Triangle';
app.OstwaldTrianglePanel.Layout.Row = 1;
app.OstwaldTrianglePanel.Layout.Column = 2;

% Create UIAxesOstwald
app.UIAxesOstwald = uiaxes(app.OstwaldTrianglePanel);
app.UIAxesOstwald.XTick = [];
app.UIAxesOstwald.YTick = [];
app.UIAxesOstwald.ZTick = [];
app.UIAxesOstwald.TickDir = 'none';
app.UIAxesOstwald.Position = [6 5 282 230];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion

```

```
methods (Access = public)
    % Construct app
    function app = flame_solver
        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargout == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
```

*Mário Loureiro*

**ANEXO B – CÓDIGO-FONTE DA APLICAÇÃO *PROBE DASHBOARD***

```

classdef probe_dashboard < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure
        ProbeMenu
        StartmeasurementMenu
        ClearmeasurementsMenu
        ClearlogMenu
        CollecteddataPanel
        GridLayout2
        O_2EditField
        O_2EditFieldLabel
        COEditField
        COEditFieldLabel
        TemperatureEditField
        TemperatureEditFieldLabel
        volLabel_2
        volLabel
        CLabel
        MeasurementPanel
        Lamp
        LogTextArea
        LogTextAreaLabel
    end

    methods (Access = private)

function collectSensorData(app)
    try
        client = mqttclient("tcp://test.mosquitto.org");
        subscribe(client, "esp32/sensors/#");
        app.LogTextArea.Value = [app.LogTextArea.Value; 'Connecting...'];
        app.Lamp.Color = [1, 1, 0.5];
    catch ME
        uialert(app.UIFigure, [ME.message], 'Connection error', 'Icon', ...
            'error');
        app.Lamp.Color = [1, 0.502, 0.502];
        app.LogTextArea.Value = [app.LogTextArea.Value;...
            'Error: Connection to MQTT broker was not established.'];
        return;
    end
    nowTime = datetime('now');
    lastHeat = getappdata(0,'LastHeatTime');
    if isempty(lastHeat) || seconds(nowTime - lastHeat) >= 120
        app.LogTextArea.Value = [app.LogTextArea.Value; ...
            'Heating sensors...'];
        setappdata(0,'LastHeatTime', nowTime);
        app.Lamp.Color = [0.9 0.9 0.2];
        pause(30);
    end
    temperature = strings(6,1);
    co_percent = strings(6,1);
    o2 = strings(6,1);
    dataReceived = false;
    for i = 1:4
        pause(2);
        data = read(client);
        if ~isempty(data)
            dataReceived = true;
        end
        for j = 1:height(data)
            topic = string(data.Topic(j));
            value = string(data.Data(j));
            if topic == "esp32/sensors/temperature"
                temperature(i) = value;
                app.TemperatureEditField.Value = str2double(value);
            elseif topic == "esp32/sensors/co"
                co_percent(i) = value;
                app.COEditField.Value = str2double(value);
            elseif topic == "esp32/sensors/o2"
                o2(i) = value;
                app.O_2EditField.Value = str2double(value);
            end
        end
    end
    if ~dataReceived
        uialert(app.UIFigure, ...
            'Error: No data received. Possible causes: Internet disconnected or probe switched off.', ...
            'Error', 'Icon', 'error');
        app.Lamp.Color = [1 0.502 0.502];
        app.LogTextArea.Value = [app.LogTextArea.Value; ...

```

```

        'Error: No data received.'];
else
    app.LogTextArea.Value = [app.LogTextArea.Value; ...
        'Data collection complete.'];
    clear client;
    app.Lamp.Color = [0.5, 1, 0.5];
    validT = temperature( temperature~="" & temperature~="NaN" );
if isempty(validT)
    lastT = "";
else
    lastT = validT(end);
end
validCO = co_percent( co_percent~="" & co_percent~="NaN" );
if isempty(validCO)
    lastCO = "";
else
    lastCO = validCO(end);
end
validO2 = o2( o2~="" & o2~="NaN" );
if isempty(validO2)
    lastO2 = "";
else
    lastO2 = validO2(end);
end
setappdata(0, 'TemperatureData', lastT);
setappdata(0, 'COData', lastCO);
setappdata(0, 'O2Data', lastO2);

end
end
end

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function startupFcn(app)
    app.UIFigure.Name = 'Probe_dashboard';
    app.UIFigure.Icon = 'icon_2.png';
    app.Lamp.Color = [1 0.5 0.5];
    app.LogTextArea.Value = '';
    app.TemperatureEditField.Value = [];
    app.COEditField.Value = [];
    app.O_2EditField.Value = [];
end

% Menu selected function: StartmeasurementMenu
function StartmeasurementMenuSelected(app, event)
app.LogTextArea.Value = '';
app.TemperatureEditField.Value = [];
app.COEditField.Value = [];
app.O_2EditField.Value = [];
app.Lamp.Color = [1 0.5 0.5];
collectSensorData(app);

end

% Menu selected function: ClearmeasurementsMenu
function ClearmeasurementsMenuSelected(app, event)
app.TemperatureEditField.Value = [];
app.COEditField.Value = [];
app.O_2EditField.Value = [];
end

% Menu selected function: ClearlogMenu
function ClearlogMenuSelected(app, event)
    app.LogTextArea.Value = '';
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 406 264];
app.UIFigure.Name = 'MATLAB App';

% Create ProbeMenu
app.ProbeMenu = uimenu(app.UIFigure);
app.ProbeMenu.Text = 'Probe';

% Create StartmeasurementMenu

```

```

app.StartmeasurementMenu = uimenu(app.ProbeMenu);
app.StartmeasurementMenu.MenuSelectedFcn = createCallbackFcn(app, @StartmeasurementMenuSelected, true);
app.StartmeasurementMenu.Text = 'Start measurement';

% Create ClearmeasurementsMenu
app.ClearmeasurementsMenu = uimenu(app.ProbeMenu);
app.ClearmeasurementsMenu.MenuSelectedFcn = createCallbackFcn(app, @ClearmeasurementsMenuSelected, true);
app.ClearmeasurementsMenu.Text = 'Clear measurements';

% Create ClearlogMenu
app.ClearlogMenu = uimenu(app.ProbeMenu);
app.ClearlogMenu.MenuSelectedFcn = createCallbackFcn(app, @ClearlogMenuSelected, true);
app.ClearlogMenu.Text = 'Clear log';

% Create MeasurementPanel
app.MeasurementPanel = uipanel(app.UIFigure);
app.MeasurementPanel.Title = 'Measurement';
app.MeasurementPanel.Position = [12 158 384 92];

% Create LogTextAreaLabel
app.LogTextAreaLabel = uilabel(app.MeasurementPanel);
app.LogTextAreaLabel.HorizontalAlignment = 'right';
app.LogTextAreaLabel.Position = [41 42 25 22];
app.LogTextAreaLabel.Text = 'Log';

% Create LogTextArea
app.LogTextArea = uitextarea(app.MeasurementPanel);
app.LogTextArea.Editable = 'off';
app.LogTextArea.Position = [81 6 293 60];

% Create Lamp
app.Lamp = uilamp(app.MeasurementPanel);
app.Lamp.Position = [12 42 20 20];
app.Lamp.Color = [1 0.502 0.502];

% Create CollecteddataPanel
app.CollecteddataPanel = uipanel(app.UIFigure);
app.CollecteddataPanel.Title = 'Collected data';
app.CollecteddataPanel.Position = [13 32 383 126];

% Create GridLayout2
app.GridLayout2 = uigridlayout(app.CollecteddataPanel);
app.GridLayout2.ColumnWidth = {53.93, 26.96, '1.99x', 38.95, '1x'};
app.GridLayout2.RowHeight = {21.97, 21.97, 21.97};

% Create CLabel
app.CLabel = uilabel(app.GridLayout2);
app.CLabel.Layout.Row = 1;
app.CLabel.Layout.Column = 4;
app.CLabel.Interpreter = 'tex';
app.CLabel.Text = '%C';

% Create volLabel
app.volLabel = uilabel(app.GridLayout2);
app.volLabel.Layout.Row = 2;
app.volLabel.Layout.Column = 4;
app.volLabel.Interpreter = 'tex';
app.volLabel.Text = '%vol.';

% Create volLabel_2
app.volLabel_2 = uilabel(app.GridLayout2);
app.volLabel_2.Layout.Row = 3;
app.volLabel_2.Layout.Column = 4;
app.volLabel_2.Interpreter = 'tex';
app.volLabel_2.Text = '%vol.';

% Create TemperatureEditFieldLabel
app.TemperatureEditFieldLabel = uilabel(app.GridLayout2);
app.TemperatureEditFieldLabel.HorizontalAlignment = 'right';
app.TemperatureEditFieldLabel.Layout.Row = 1;
app.TemperatureEditFieldLabel.Layout.Column = [1 2];
app.TemperatureEditFieldLabel.Interpreter = 'tex';
app.TemperatureEditFieldLabel.Text = 'Temperature';

% Create TemperatureEditField
app.TemperatureEditField = uieditfield(app.GridLayout2, 'numeric');
app.TemperatureEditField.Limits = [0 100];
app.TemperatureEditField.ValueDisplayFormat = '%.5f';
app.TemperatureEditField.AllowEmpty = 'on';
app.TemperatureEditField.Editable = 'off';
app.TemperatureEditField.Layout.Row = 1;
app.TemperatureEditField.Layout.Column = 3;
app.TemperatureEditField.Value = [];

% Create COEditFieldLabel
app.COEditFieldLabel = uilabel(app.GridLayout2);
app.COEditFieldLabel.HorizontalAlignment = 'right';

```

```

app.COEditFieldLabel.Layout.Row = 2;
app.COEditFieldLabel.Layout.Column = 2;
app.COEditFieldLabel.Interpreter = 'tex';
app.COEditFieldLabel.Text = 'CO';

% Create COEditField
app.COEditField = uieditfield(app.GridLayout2, 'numeric');
app.COEditField.Limits = [0 100];
app.COEditField.ValueDisplayFormat = '%.5f';
app.COEditField.AllowEmpty = 'on';
app.COEditField.Editable = 'off';
app.COEditField.Layout.Row = 2;
app.COEditField.Layout.Column = 3;
app.COEditField.Value = [];

% Create O_2EditFieldLabel
app.O_2EditFieldLabel = uilabel(app.GridLayout2);
app.O_2EditFieldLabel.HorizontalAlignment = 'right';
app.O_2EditFieldLabel.Layout.Row = 3;
app.O_2EditFieldLabel.Layout.Column = 2;
app.O_2EditFieldLabel.Interpreter = 'tex';
app.O_2EditFieldLabel.Text = 'O_2';

% Create O_2EditField
app.O_2EditField = uieditfield(app.GridLayout2, 'numeric');
app.O_2EditField.Limits = [0 100];
app.O_2EditField.ValueDisplayFormat = '%.5f';
app.O_2EditField.AllowEmpty = 'on';
app.O_2EditField.Editable = 'off';
app.O_2EditField.Layout.Row = 3;
app.O_2EditField.Layout.Column = 3;
app.O_2EditField.Value = [];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = probe_dashboard

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

% Execute the startup function
runStartupFcn(app, @startupFcn)

if nargout == 0
    clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end

```

## **ANEXO C – CÓDIGO-FONTE DA APLICAÇÃO *ABOUT***

```

classdef about < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure      matlab.ui.Figure
        AboutPanel    matlab.ui.container.Panel
        LinkedInButton matlab.ui.control.Button
        Image         matlab.ui.control.Image
        TextArea       matlab.ui.control.TextArea
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Code that executes after component creation
        function startupFcn(app)
            app.UIFigure.Name = 'Flame Solver v1.0';
            app.UIFigure.Icon = './bin/icon.png';
        end

        % Button pushed function: LinkedInButton
        function LinkedInButtonPushed(app, event)
            web('https://www.linkedin.com/in/mario--loureiro', '-browser');
        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Get the file path for locating images
            pathToMLAPP = fileparts(fullfile('fullpath'));

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Position = [100 100 419 281];
            app.UIFigure.Name = 'MATLAB App';

            % Create AboutPanel
            app.AboutPanel = uipanel(app.UIFigure);
            app.AboutPanel.Title = 'About...';
            app.AboutPanel.Position = [9 9 404 264];

            % Create TextArea
            app.TextArea = uitextarea(app.AboutPanel);
            app.TextArea.Position = [18 6 231 223];
            app.TextArea.Value = {'Flame Solver v1.0 is a tool developed in MATLAB App Designer to analyse combustion processes of liquid or g'};

            % Create Image
            app.Image = uiimage(app.AboutPanel);
            app.Image.Position = [265 99 123 130];
            app.Image.ImageSource = fullfile(pathToMLAPP, 'photo.jpg');

            % Create LinkedInButton
            app.LinkedInButton = uibutton(app.AboutPanel, 'push');
            app.LinkedInButton.ButtonPushedFcn = createCallbackFcn(app, @LinkedInButtonPushed, true);
            app.LinkedInButton.Position = [264 14 124 22];
            app.LinkedInButton.Text = 'LinkedIn';

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = about

            % Create UIFigure and components
            createComponents(app);

            % Register the app with App Designer
            registerApp(app, app.UIFigure);

            % Execute the startup function
            runStartupFcn(app, @startupFcn);

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure);
        end
    end
end

```



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra