

# Relatório do Projeto: Sistema de Gestão de Publicações Científicas

---

## Autores

José Ferreira, A107278

Filipa Figueiredo, A107239

Andreia Ferreira, A107234

## Data

03/03/2025

## Visão Geral

No âmbito da unidade curricular de Algoritmos e Técnicas de Programação, foi-nos proposto o desenvolvimento de um sistema de gestão de publicações científicas em Python.

## Requisitos do Sistema

Em relação ao sistema que nos foi proposto desenvolver, havia alguns requisitos essenciais que o sistema devia cumprir para analisar com eficiência uma base de dados fornecida com as publicações científicas.

Requisitos:

### 1. Carregar Base de Dados

O programa deve ser capaz de carregar para a memória interna o conjunto de dados presente num ficheiro JSON com a seguinte estrutura:

```
[
  {
    "title": "Título da publicação",
    "abstract": "Resumo do conteúdo da publicação",
    "keywords": "Palavras-chave relacionadas com a publicação",
    "authors": [
      {
        "name": "Nome do autor",
        "affiliation": "Nome da afiliação do autor",
        "orcid": "Identificador aberto de investigador e contribuidor"
      }
    ],
    "doi": "Identificador de objeto digital",
    "pdf": "Caminho do ficheiro PDF da publicação",
    "publish_date": "Data da publicação (AAAA-MM-DD)",
    "url": "Endereço web da publicação"
  }
]
```

O dataset consiste numa lista de dicionários, e cada um corresponde a uma publicação. Estes incluem "keys" para título, resumo, palavras-chave, autores, DOI, PDF, data de publicação e URL. Relativamente aos autores, estes são representados por uma lista de dicionários, onde cada dicionário contém informações sobre nome, afiliação e ORCID.

## 2. Criar Publicações

O programa deve permitir que o utilizador crie uma nova publicação, especificando título, resumo, palavras-chave, DOI, autores (com nome, afiliação e ORCID), caminho para o PDF, data da publicação e URL da publicação.

## 3. Atualizar Publicações

O programa deve permitir que o utilizador atualize as informações de uma dada publicação. Estas informações incluem título, resumo, palavras-chave, autores (nome, afiliação e ORCID), DOI, caminho para o PDF, data da publicação e URL da publicação.

## 4. Consultar Publicações

O programa deve permitir que o utilizador consulte as publicações através de filtros por título, autor, afiliação, data de publicação e palavras-chave. Após encontrar as publicações, deve ser possível ordená-las pelos títulos ou pelas datas de publicação.

## 5. Analisar Publicações por Autor

O programa deve listar os autores, permitindo ao utilizador visualizar as publicações correspondentes a um dado autor. Esta listagem deve ser ordenada pela frequência de publicações e/ou por ordem alfabética.

## 6. Analisar Publicações por Palavras-Chave

O programa deve permitir a visualização das palavras-chave existentes no conjunto de dados, possibilitando que o utilizador visualize as publicações correspondentes a uma dada palavra-chave. As palavras-chave devem ser ordenadas pela sua frequência e/ou por ordem alfabética.

## 7. Estatísticas das Publicações

O programa deve exibir estatísticas referentes às publicações presentes no conjunto de dados, apresentando gráficos para os seguintes tópicos:

- Distribuição de publicações por ano.
- Distribuição de publicações por mês de um determinado ano.
- Número de publicações por autor (top 20 autores).
- Distribuição de publicações de um autor por anos.
- Distribuição de palavras-chave pela sua frequência (top 20 palavras-chave).
- Distribuição de palavras-chave mais frequentes por ano.

## 8. Armazenamento dos Dados

O programa deve guardar as informações alteradas ou adicionadas em memória no ficheiro de suporte.

## 9. Importação de Dados

O programa deve permitir que, a qualquer momento, seja possível importar novos registos de outro conjunto de dados com a mesma estrutura mencionada anteriormente.

## 10. Exportação Parcial de Dados

O programa deve permitir exportar os registos resultantes de uma pesquisa para um ficheiro.

# Interface

## 1. Gestão de Publicações

### 1.1 Adicionar novas publicações

Para adicionar novas publicações desenvolvemos uma interface clara onde é possível inserir todas as informações da publicação tendo como obrigatoriedade o título e o DOI, uma vez que são parametros importantes para a identificação da publicação como no exemplo a seguir:

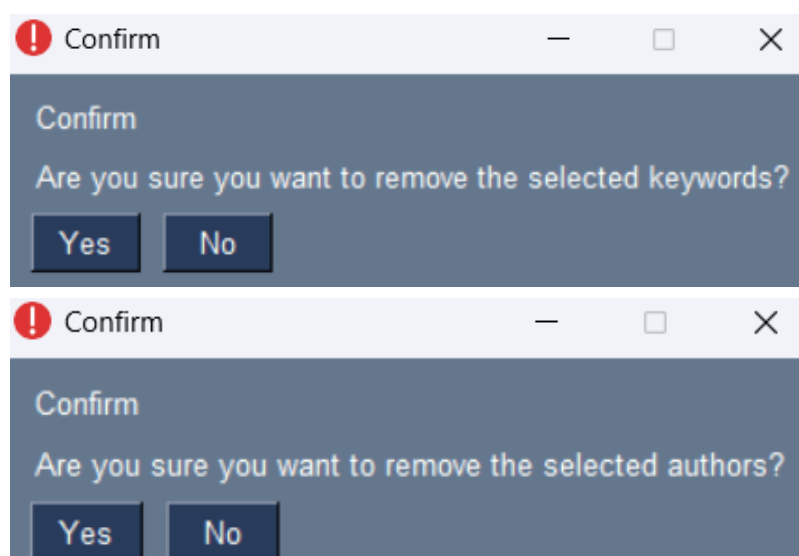
The screenshot shows a 'Add Publication' window with the following fields and content:

- Title:** TESTE
- Abstract:** TESTE
- Publish Date:** 2025-01-08 (with a 'Select' button)
- Keywords:** A list containing 'TESTE', 'TESTE 2', and 'TESTE 3', with 'Add' and 'Remove' buttons.
- Authors:** A list containing 'Name: TESTE; Affiliation: TESTE', 'Name: TESTE 2; Affiliation:', and 'Name: TESTE 3; Affiliation: TESTE 3', with 'Add' and 'Remove' buttons.
- DOI:** TESTE
- PDF (URL or path):** TESTE
- Article URL:** TESTE
- Buttons:** 'Save' and 'Cancel' at the bottom left.

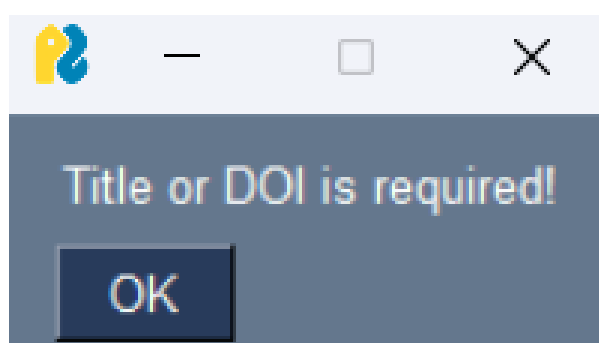
Para adicionar novas palavras-chave e novos autores, as seguintes janelas são apresentadas ao utilizador:



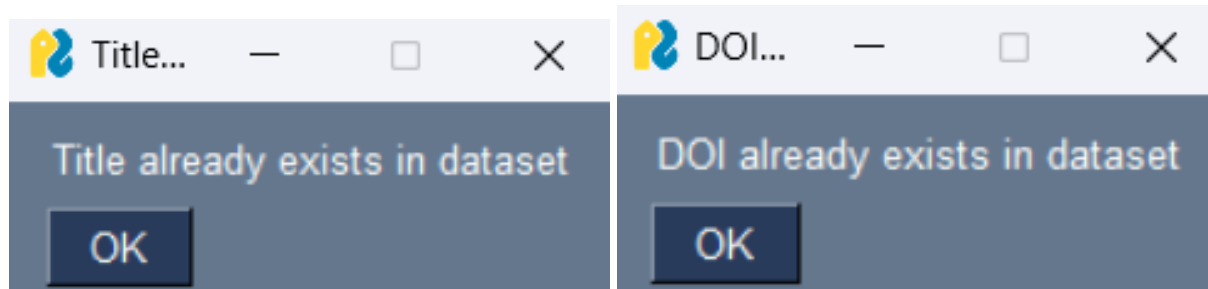
Caso o utilizador deseje remover alguma das palavras-chave ou autores que tenham adicionado, as seguintes janelas são apresentadas ao utilizador:



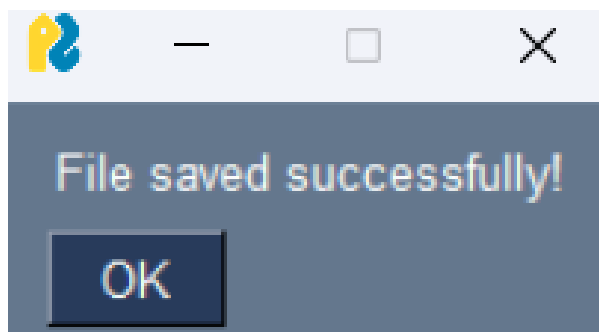
Em caso do utilizador tentar guardar a publicação sem título ou DOI, um aviso será mostrado como no exemplo a seguir:



Se porventura for inserido um título ou um DOI já existente, um aviso será mostrado como no exemplo a seguir:

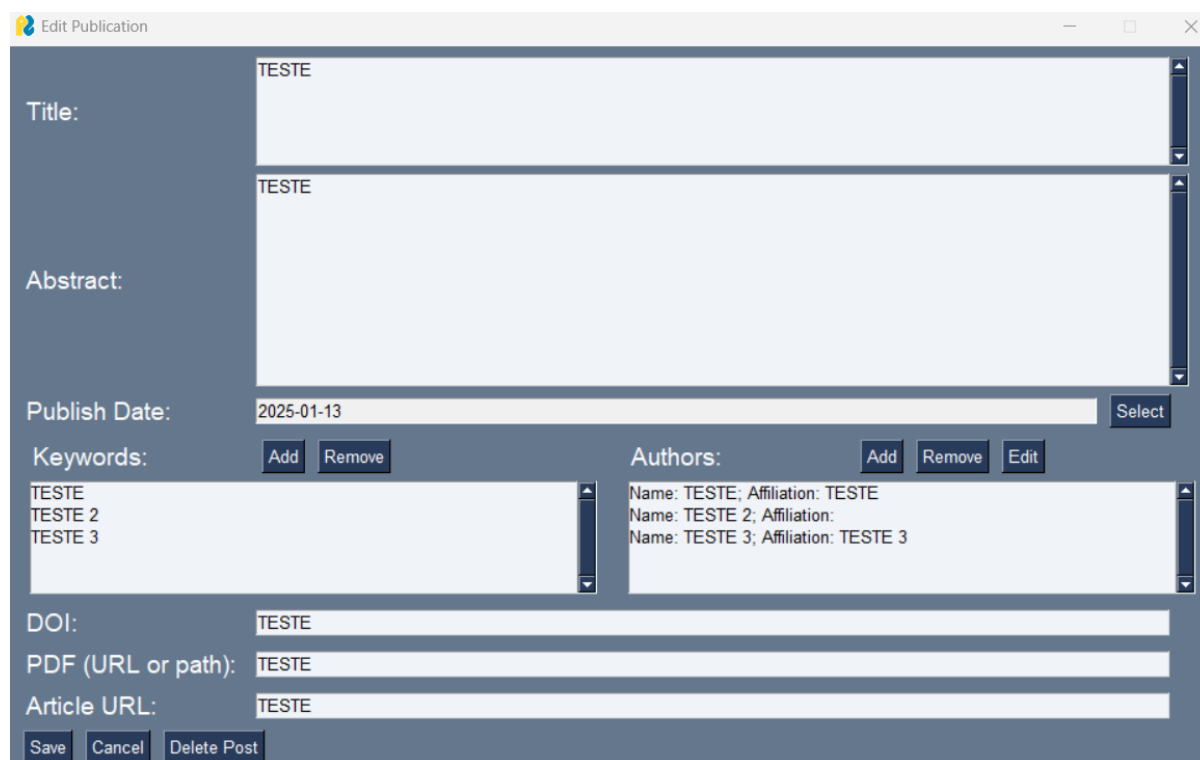


Após inserir as informações corretamente e guardar a nova publicação, a janela seguinte será exposta:



## 1.2 Atualizar publicações existentes

Para atualizar as informações de uma publicação já existente no dataset desenvolvemos uma interface clara onde serão exibidos todos os parâmetros devidamente preenchidos com as informações da publicação mas com a possibilidade de mudar. Apesar da estrutura e de funções muito semelhante à janela de adicionar publicação, esta apresenta dois novos botões: "Edit" para editar autores já existentes, e "Delete Post" para apagar a publicação do dataset.



The 'Edit Publication' window displays the following fields and controls:

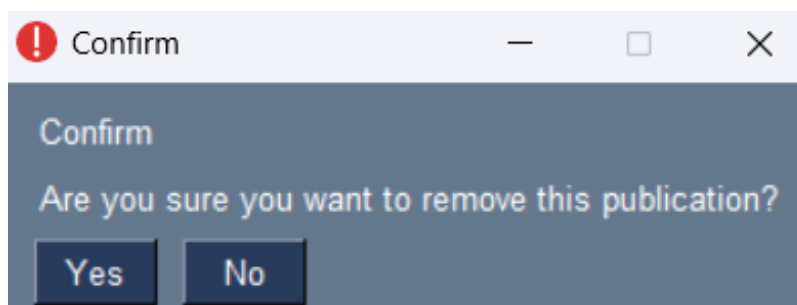
- Title:** A text input field containing 'TESTE'.
- Abstract:** A large text area containing 'TESTE'.
- Publish Date:** A date input field showing '2025-01-13' with a 'Select' button.
- Keywords:** A list box containing 'TESTE', 'TESTE 2', and 'TESTE 3', with 'Add' and 'Remove' buttons.
- Authors:** A list box containing three entries: 'Name: TESTE; Affiliation: TESTE', 'Name: TESTE 2; Affiliation:', and 'Name: TESTE 3; Affiliation: TESTE 3'. It includes 'Add', 'Remove', and 'Edit' buttons.
- DOI:** A text input field containing 'TESTE'.
- PDF (URL or path):** A text input field containing 'TESTE'.
- Article URL:** A text input field containing 'TESTE'.
- Buttons:** 'Save', 'Cancel', and 'Delete Post' buttons at the bottom.

No caso do utilizador querer editar os autores já existentes ao selecionar essa opção, o seguinte janela será exibida:



The 'Edit Author' dialog box has a title bar with a logo and the text 'Edit Author'. It contains two text input fields: 'Author name:' with the value 'TESTE' and 'Affiliation name:' with the value 'TESTE'. At the bottom are two buttons: 'Save' and 'Cancel'.

Na hipótese do utilizador querer apagar a publicação do dataset, o seguinte aviso será mostrado:

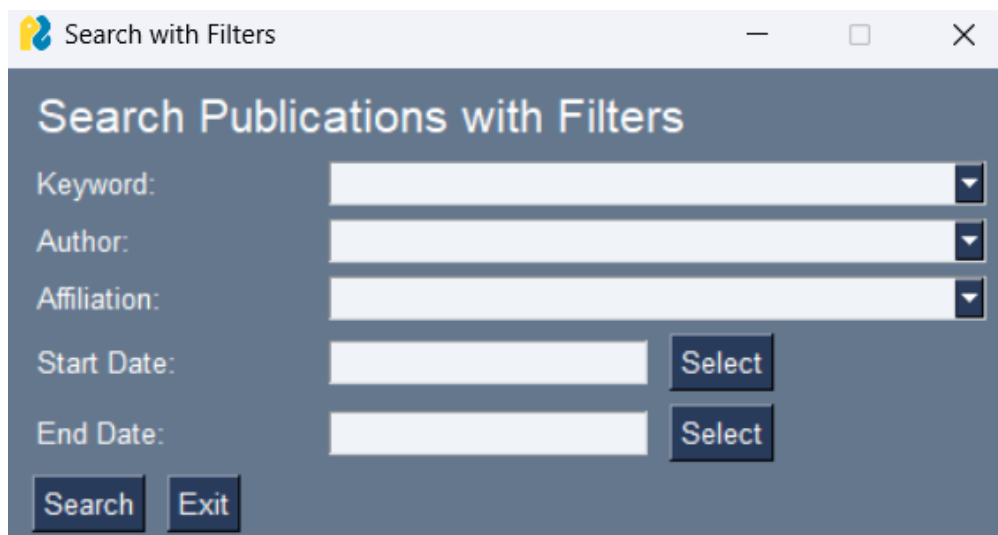


The 'Confirm' dialog box has a title bar with a red exclamation mark icon and the text 'Confirm'. The main text reads 'Confirm' followed by 'Are you sure you want to remove this publication?'. At the bottom are two buttons: 'Yes' and 'No'.

## 2. Capacidades de Pesquisa

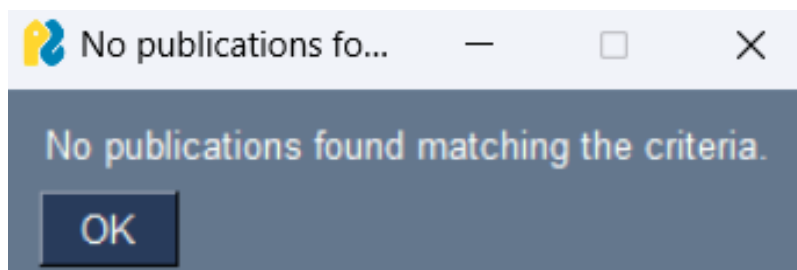
### 2.1 Pesquisa com filtros

O programa é capaz de encontrar publicações utilizando vários filtros, tais como, palavra-chave, autor, afiliação, intervalo de tempo. Para realizar a pesquisa a seguinte janela é exibida:



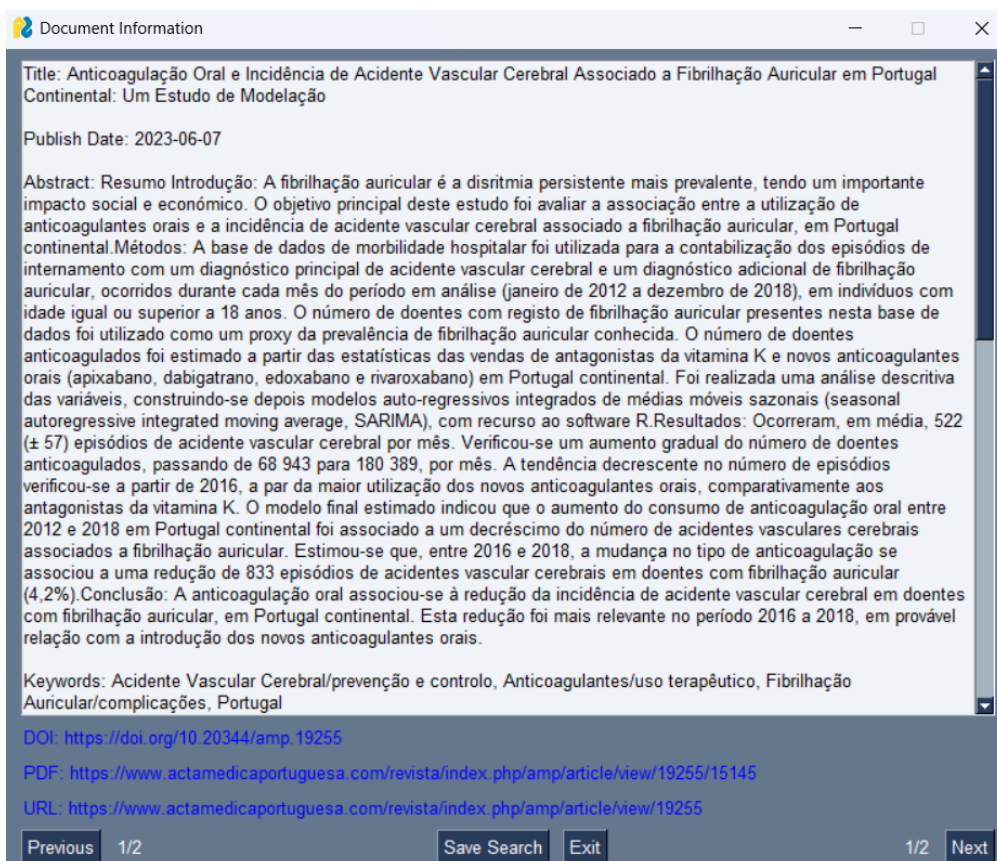
The 'Search with Filters' dialog box has a title bar with a logo and the text 'Search with Filters'. The main title is 'Search Publications with Filters'. It contains five input fields: 'Keyword:', 'Author:', 'Affiliation:', 'Start Date:', and 'End Date:'. The first three are text boxes with dropdown arrows on the right. The last two are text boxes with 'Select' buttons to their right. At the bottom are two buttons: 'Search' and 'Exit'.

Se não existir nenhuma publicação com os parâmetros submetidos, esta será apresentada:

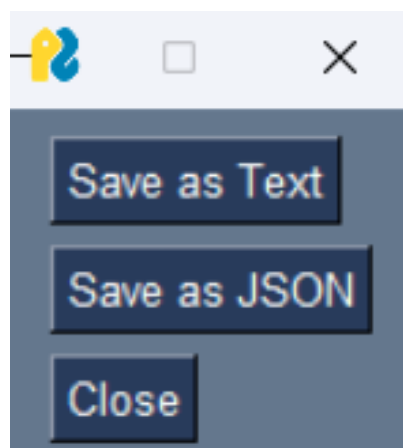


The 'No publications found' dialog box has a title bar with a logo and the text 'No publications fo...'. The main text reads 'No publications found matching the criteria.'. At the bottom is one button: 'OK'.

Após o programa procurar as publicações que apresentem os parâmetros preenchidos, será mostrada na seguinte janela as informações todas das publicações, os links que dão para interagir direcionando diretamente para a Web



O botão "Save Search" apresentado na janela serve para o utilizador poder guardar a pesquisa em ficheiro txt ou em um ficheiro json com a mesma estrutura do dataset.



TESTE.txt

Projeto > TESTE.txt

1

Title: A Importância do Peixe-Zebra na Investigação Biomédica

2

3

Publish Date: 2013-10-31

4

5

Abstract: Resumo Introdução: O peixe-zebra (Danio rerio) é um excelente organismo modelo para o estudo do desenvolvimento dos vertebrados

6

7

Keywords: N/A

8

9

Authors:

10

Author: Bárbara Tavares

11

Affiliation: Centro de Estudos de Doenças Crónicas. Faculdade de Ciências Médicas de Lisboa. Lisboa. Portugal.

12

ORCID: N/A

13

14

Author: Susana Santos Lopes

15

Affiliation: Centro de Estudos de Doenças Crónicas. Faculdade de Ciências Médicas de Lisboa. Lisboa. Portugal.

16

ORCID: N/A

17

18

----

19

20

DOI: <https://doi.org/10.20344/amp.4628>

21

22

PDF: <https://www.actamedicaportuguesa.com/revista/index.php/amp/article/view/4628/3779>

23

24

URL: <https://www.actamedicaportuguesa.com/revista/index.php/amp/article/view/4628>

25

26

////

27

Title: Actinomicetoma por Cellulosimicrobium cellulans num Jovem Guineense: Breve Revisão de Literatura a Propósito de um Caso Clínico

28

29

Publish Date: 2023-03-15

30

31

Abstract: Resumo O micetoma é causado pela inoculação subcutânea de fungos filamentosos ou bactérias filamentosas aeróbias. Cellulosimicrobium

32

33

Keywords: Actinomycetales, Micetoma/diagnóstico, Micetoma/tratamento

34

35

Authors:

36

Author: Margarida Trindade Torres

37

Affiliation: Serviço de Doenças Infecciosas. Hospital de Curry Cabral. Lisboa.

TESTE.json

Projeto > TESTE.json > ...

1

{

2

{

3

"abstract": "Resumo Introdu\u00e7\u00e3o: O peixe-zebra (Danio rerio) \u00e9 um excelente organismo modelo para o estudo do desenvolvimento dos vertebrados

4

"authors": [

5

{

6

"name": "B\u00e1rbara Tavares",

7

"affiliation": "Centro de Estudos de Doen\u00e7as Cr\u00f3nicas. Faculdade de Ci\u00eancias M\u00e9dicas de Lisboa. Lisboa. Portugal.

8

},

9

{

10

"name": "Susana Santos Lopes",

11

"affiliation": "Centro de Estudos de Doen\u00e7as Cr\u00f3nicas. Faculdade de Ci\u00eancias M\u00e9dicas de Lisboa. Lisboa. Portugal.

12

}

13

],

14

"doi": "https://doi.org/10.20344/amp.4628",

15

"pdf": "https://www.actamedicaportuguesa.com/revista/index.php/amp/article/view/4628/3779",

16

"publish\_date": "2013-10-31",

17

"title": "A Import\u00e2ncia do Peixe-Zebra na Investiga\u00e7\u00e3o Biom\u00e9dica",

18

"url": "https://www.actamedicaportuguesa.com/revista/index.php/amp/article/view/4628"

19

},

20

{

21

"abstract": "Resumo O micetoma \u00e9 causado pela inocula\u00e7\u00e3o subcut\u00e2nea de fungos filamentosos ou bact\u00e9rias filamentosas aer\u00f3bias. Cellulosimicrobium

22

"keywords": "Actinomycetales, Micetoma/diagn\u00f3stico, Micetoma/tratamento",

23

"authors": [

24

{

25

"name": "Margarida Trindade Torres",

26

"affiliation": "Servi\u00e7o de Doen\u00e7as Infecciosas. Hospital de Curry Cabral. Lisboa.",

27

"orcid": "https://orcid.org/0000-0002-5883-0723"

28

},

29

{

30

"name": "Beatriz Sousa Nunes",

31

"affiliation": "Servi\u00e7o de Infeciologia Pedi\u00e1trica. Hospital de Dona Estef\u00e2nia. Lisboa."

32

},

33

{

34

"name": "Lu\u00edsa Varandas",

35

"affiliation": "Servi\u00e7o de Infeciologia Pedi\u00e1trica. Hospital de Dona Estef\u00e2nia. Lisboa; NOVA Medical School"

36

}

37

}

2.2 Pesquisa por título

O programa tem a funcionalidade de apresentar uma lista com todos os títulos das publicações presentes no dataset, com a possibilidade de os organizar por ordem alfabética ou por data de publicação.





Se o utilizador selecionar um dos títulos apresentados, uma janela igual à função anterior aparecerá com as informações da publicação e com o botão para salvar a pesquisa.

## 2.3 Pesquisa por autor

O programa tem a funcionalidade de apresentar uma lista com todos os autores das publicações presentes no dataset, com a possibilidade de os organizar por ordem alfabética ou pela sua frequência.



Se o utilizador seleccionar um dos autores apresentados, uma janela igual à função anterior aparecerá com as informações da publicação e com o botão para salvar a pesquisa.

## 2.4 Pesquisa por afiliação

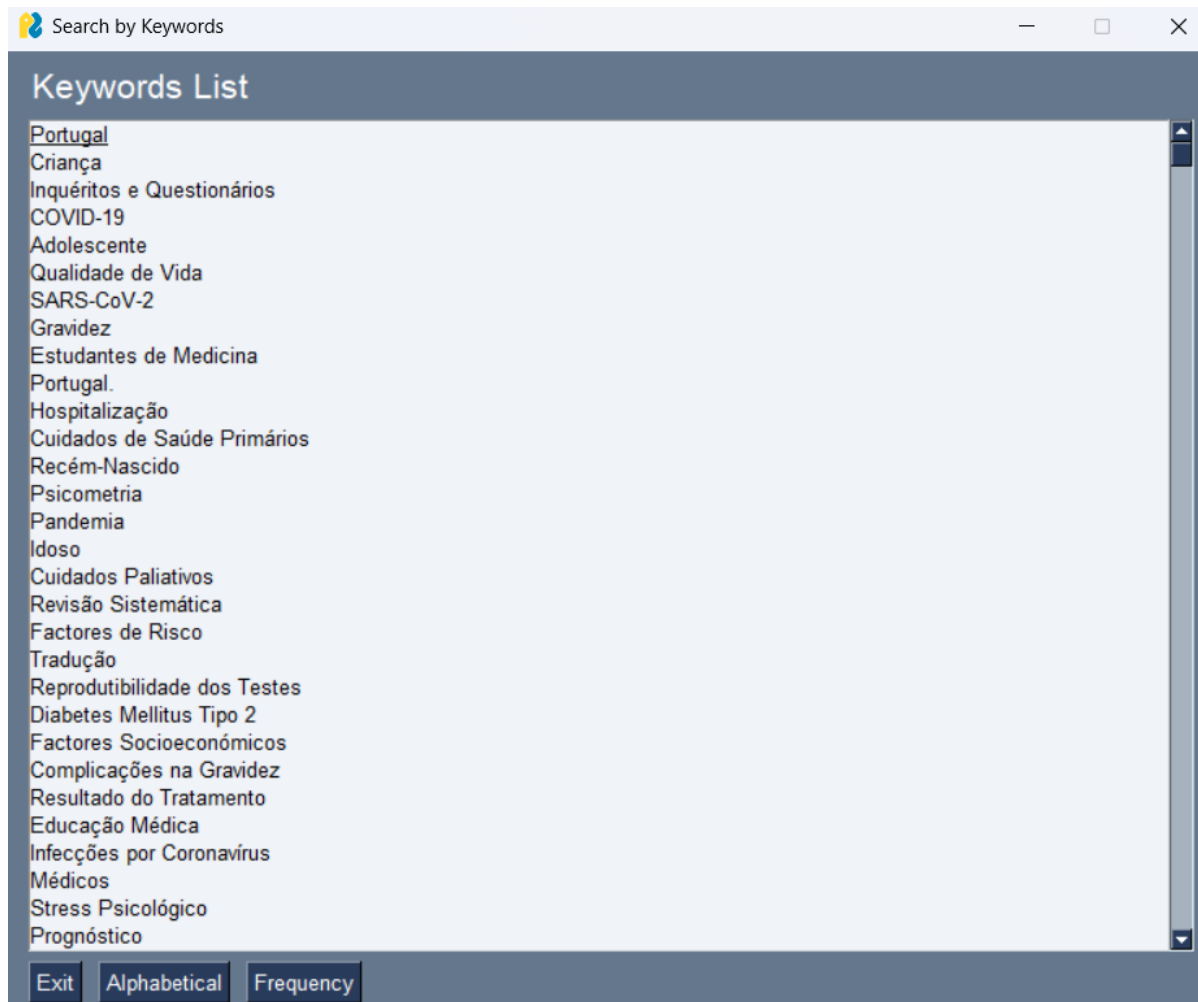
O programa tem a funcionalidade de apresentar uma lista com todas as afiliações das publicações presentes no dataset, com a possibilidade de os organizar por ordem alfabética ou pela sua frequência.



Se o utilizador selecionar uma das afiliações apresentados, uma janela igual à função anterior aparecerá com as informações da publicação e com o botão para salvar a pesquisa.

## 2.5 Pesquisa por palavras-chave

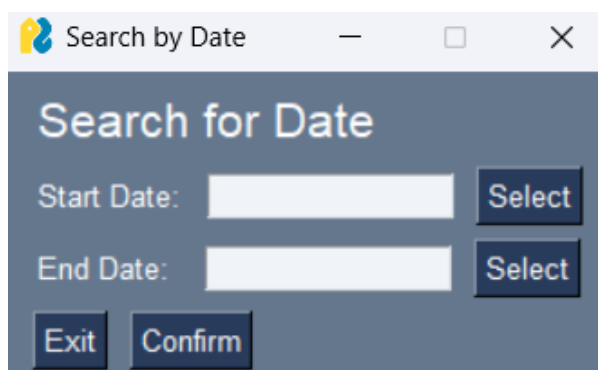
O programa tem a funcionalidade de apresentar uma lista com todas as palavras-chave das publicações presentes no dataset, com a possibilidade de os organizar por ordem alfabética ou pela sua frequência.



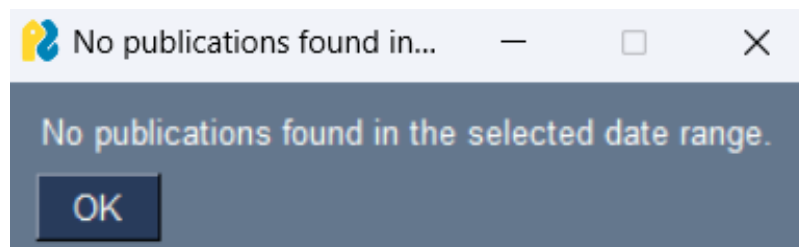
Se o utilizador seleccionar uma das palavras-chave apresentados, uma janela igual à função anterior aparecerá com as informações da publicação e com o botão para salvar a pesquisa.

## 2.6 Pesquisa por intervalo de datas

O programa tem a funcionalidade de procurar publicações num dado intervalo de tempo.



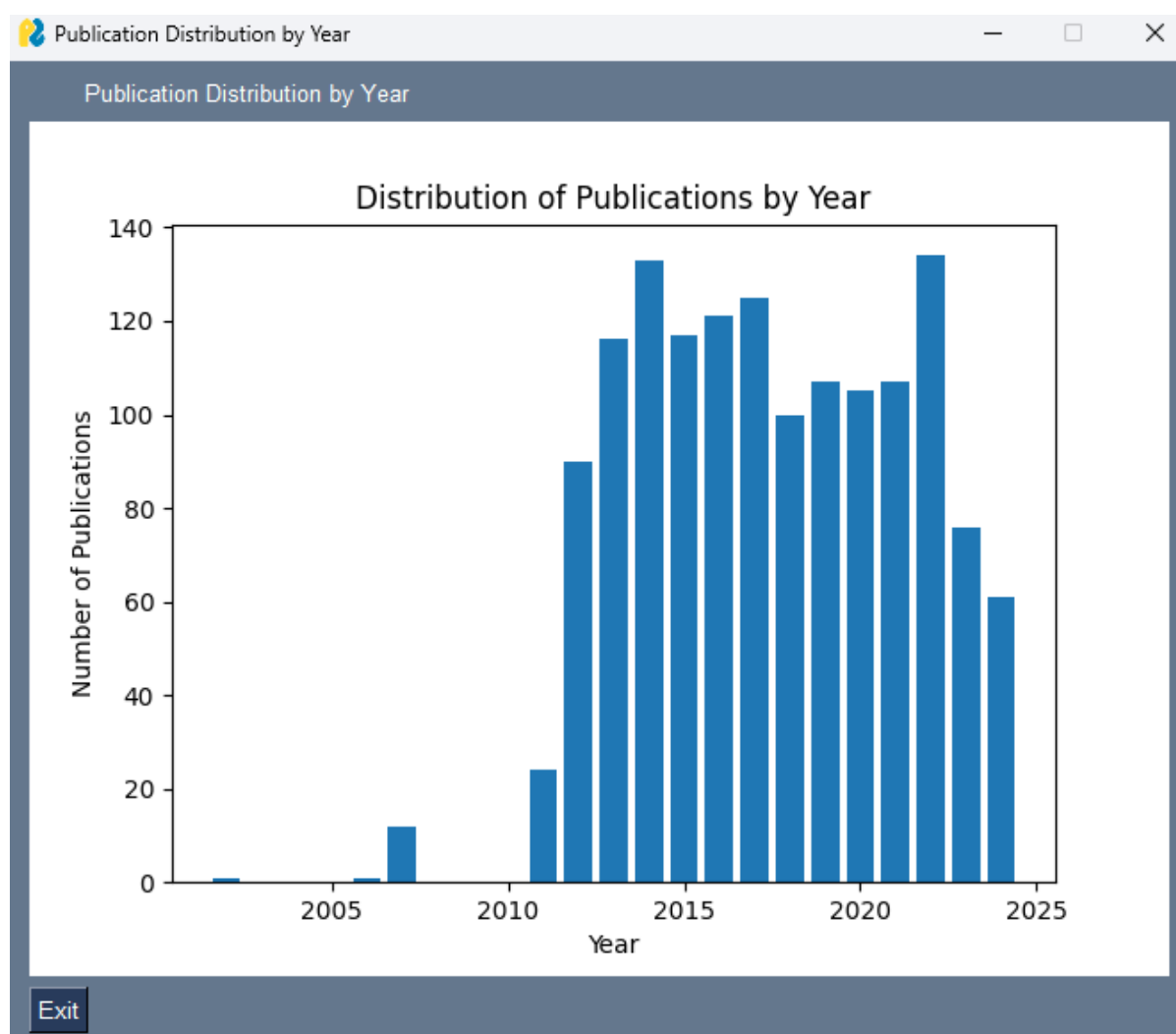
Após inserida as datas uma janela igual à função anterior aparecerá com as informações da publicação e com o botão para salvar a pesquisa. Se não houver nenhuma publicação no dado intervalo de tempo a seguinte janela será exibida:



### 3. Análise Estatística

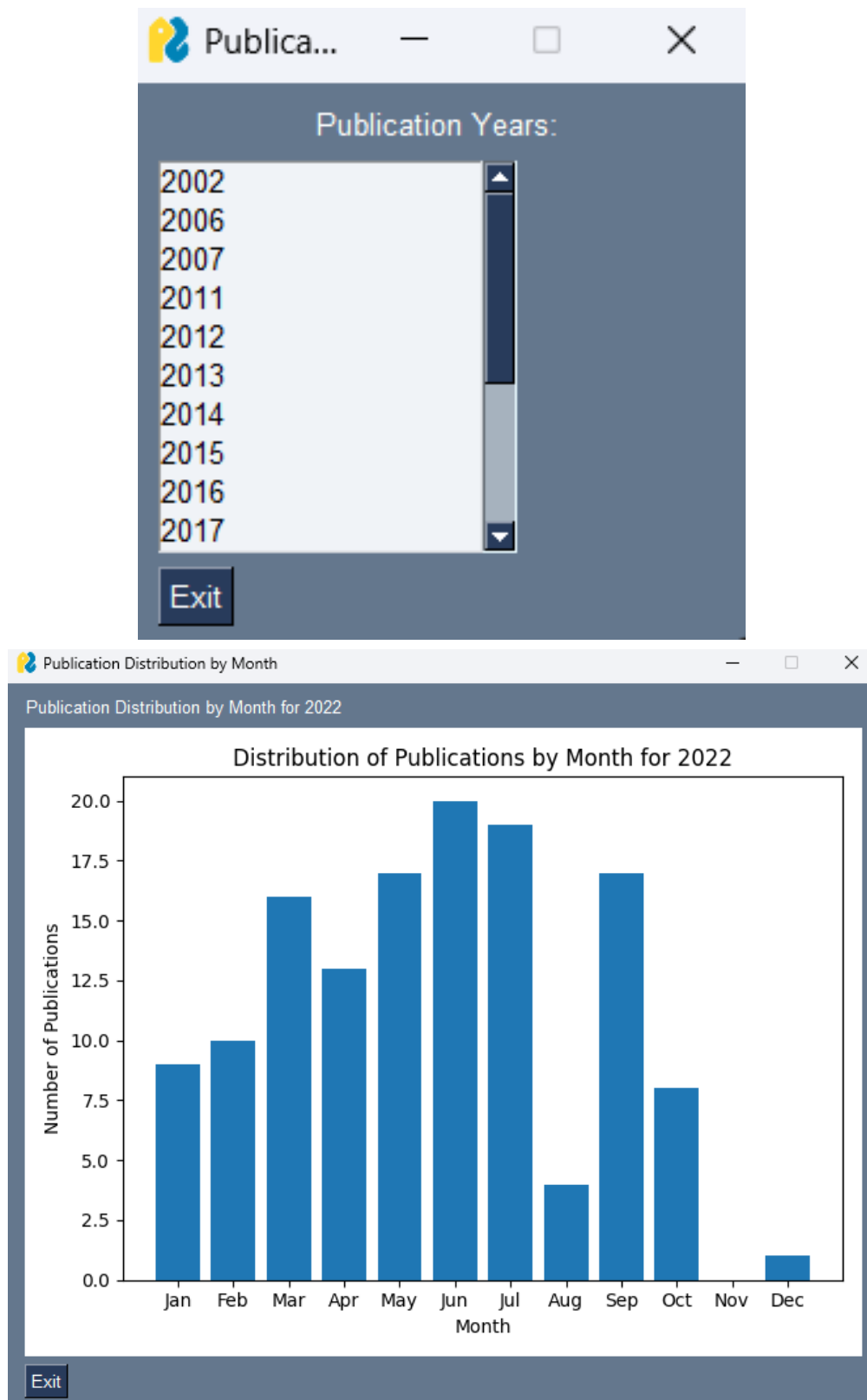
#### 3.1 Distribuição de publicações por ano

O programa é capaz de exibir um gráfico com a distribuição de publicações por ano.



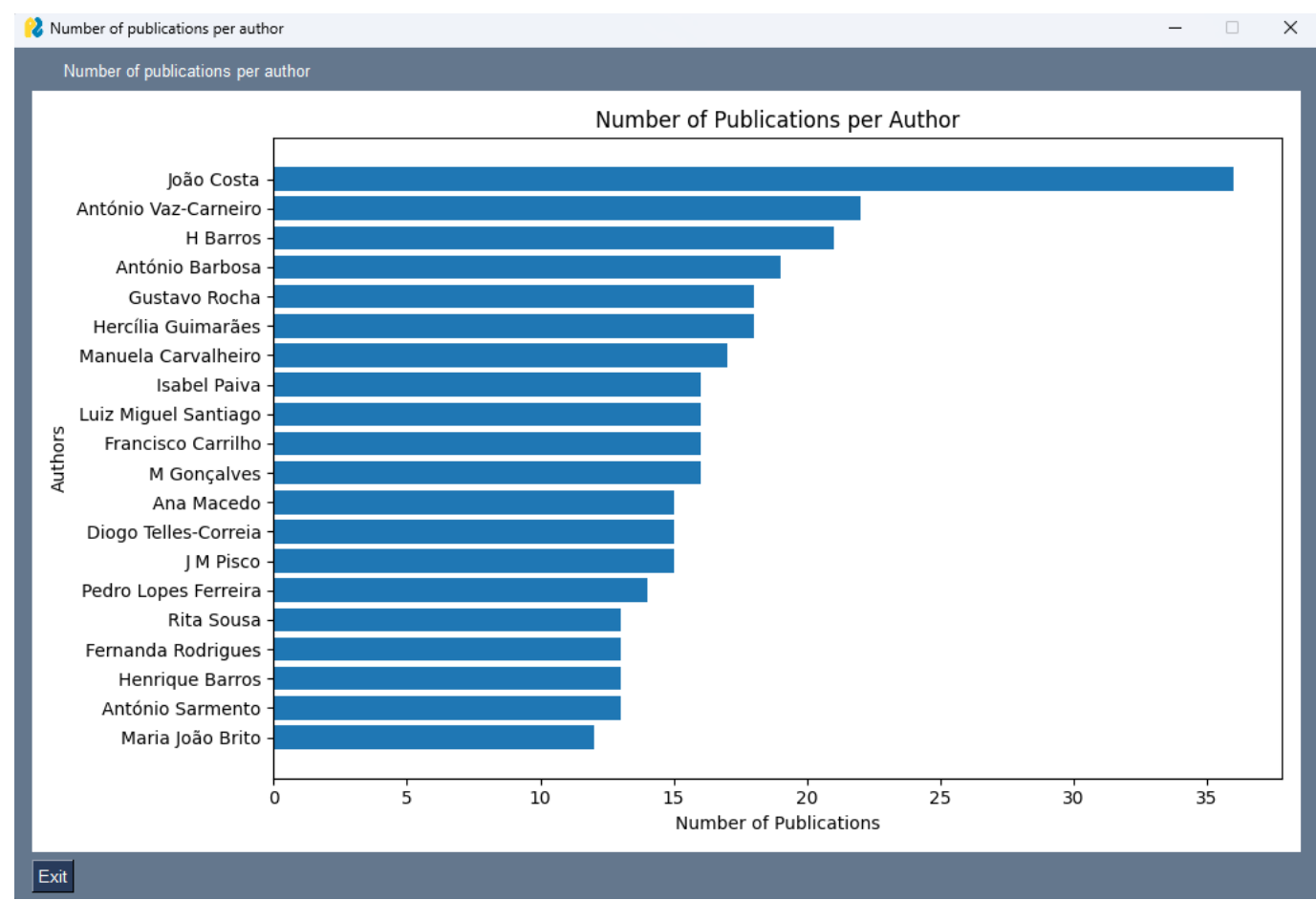
#### 3.2 Distribuição mensal de publicações

O programa é capaz de exibir um gráfico com a distribuição mensal de publicações num dado ano.



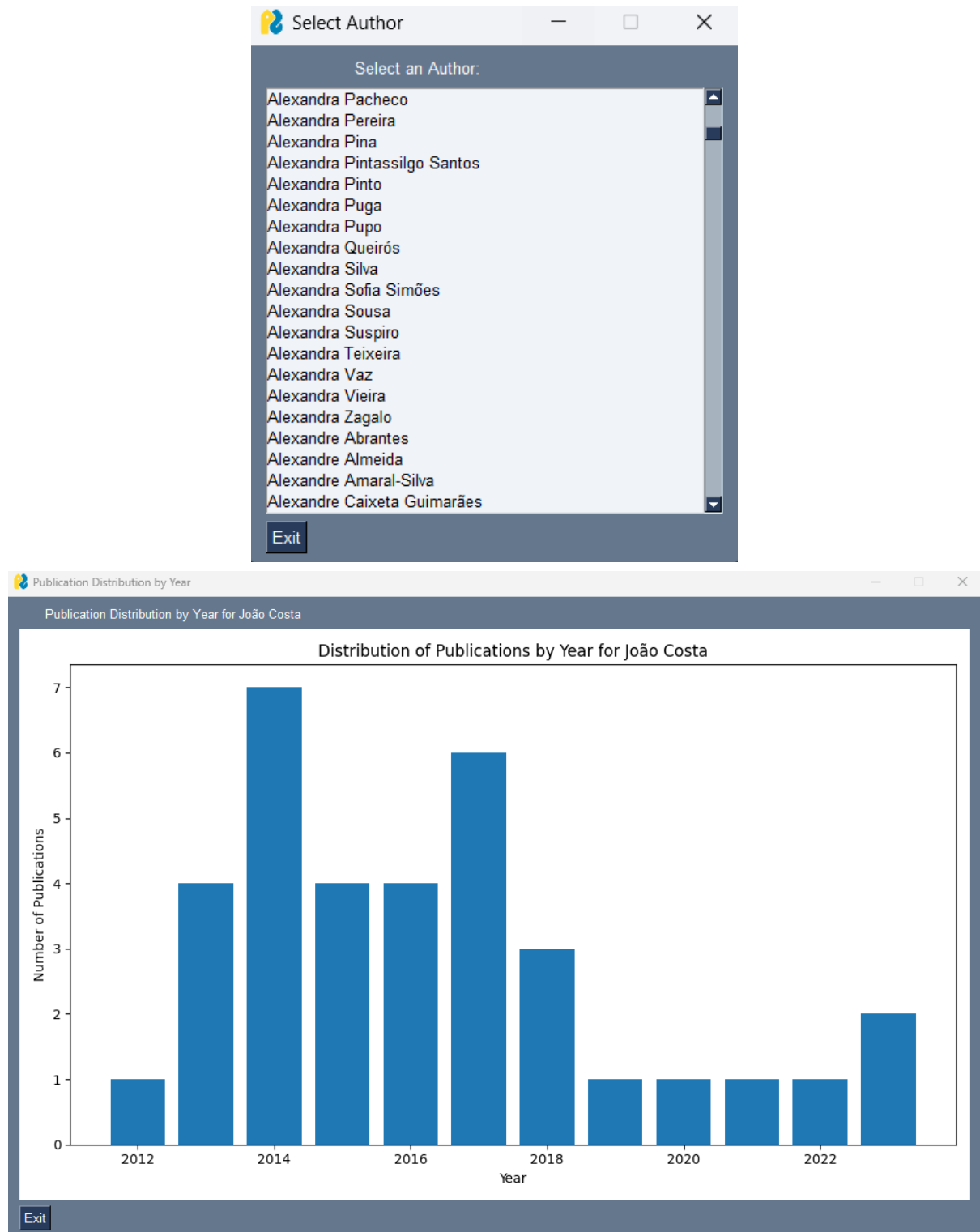
### 3.3 Número de publicação por autor (Top20)

O programa é capaz de exibir um gráfico com o número de publicações de um autor, mas apenas o Top20



3.4 Distribuição de publicações por ano de um dado autor

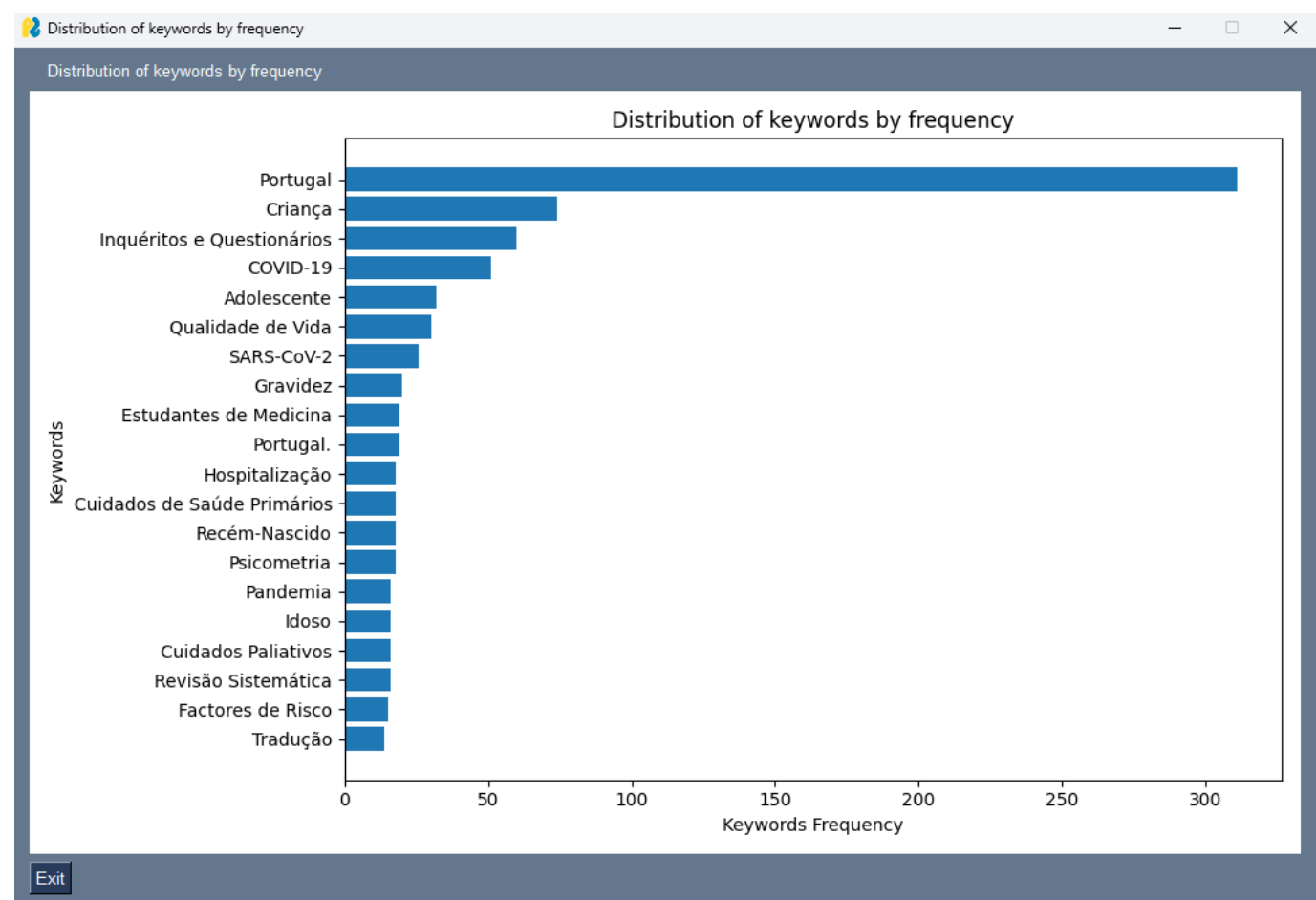
O programa é capaz de exibir um gráfico com a distribuição de publicações por ano após ser selecionado um autor específico.



### 3.5 Distribuição de palavras-chave pela sua frequência

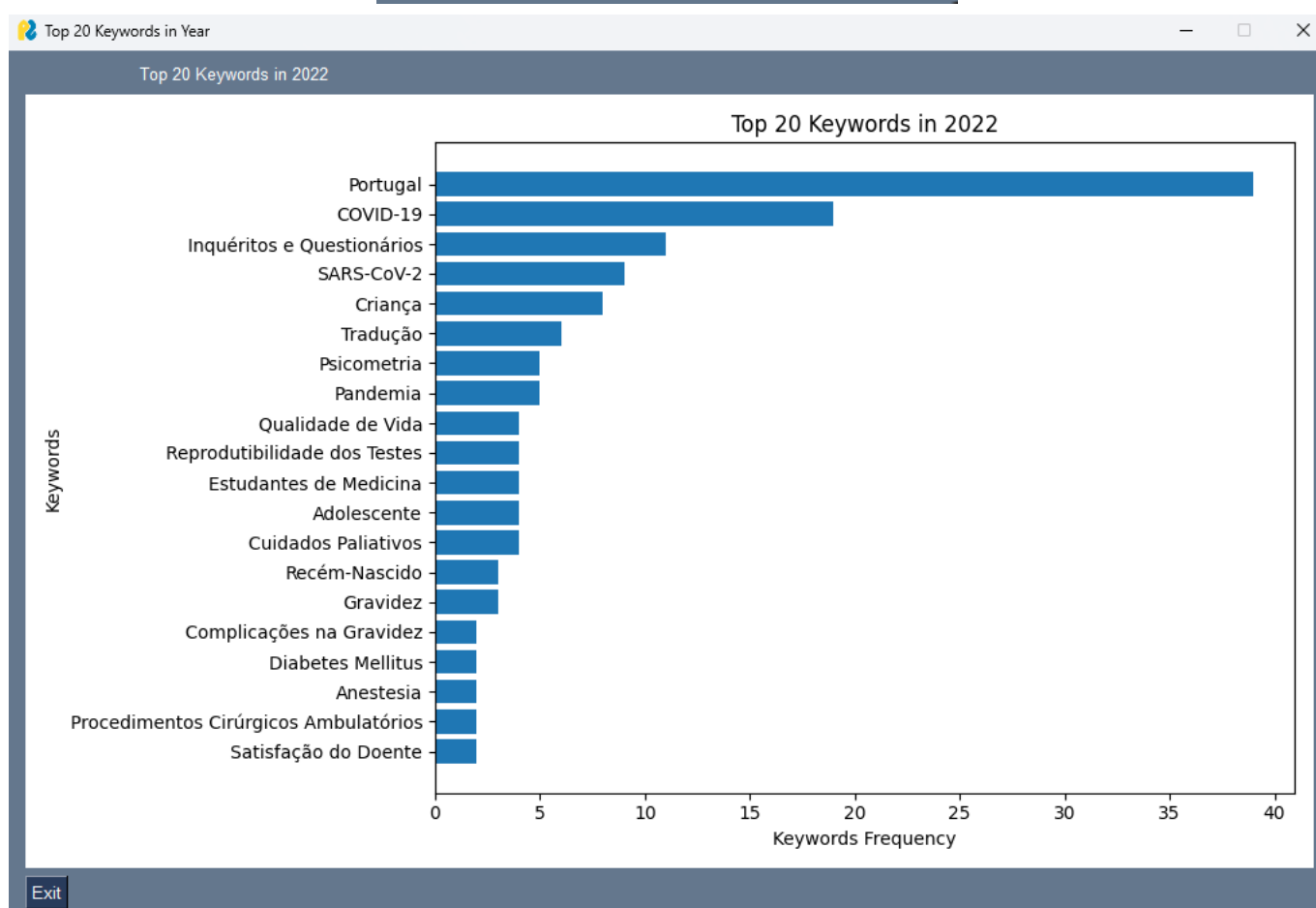
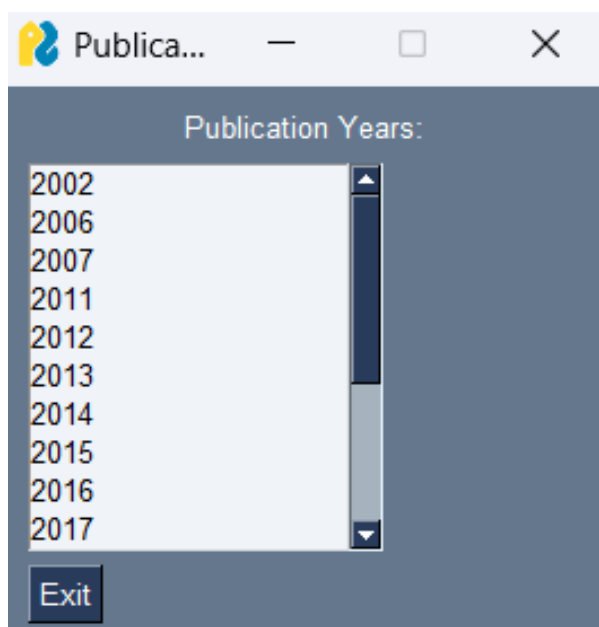
O programa é capaz de exibir um gráfico com a distribuição das palavras-chave pela sua frequência.





3.6 Número de palavras-chave num dado ano (Top20)

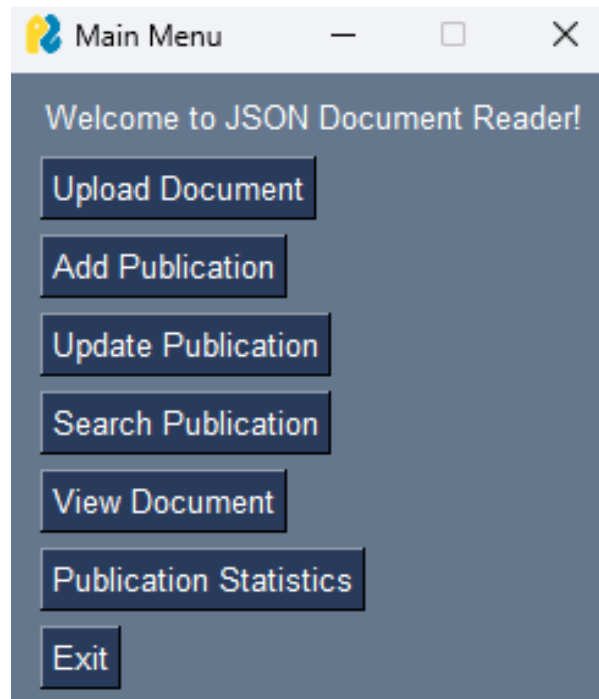
O programa é capaz de exibir um gráfico com o número de palavras-chave após ser selecionado um determinado ano



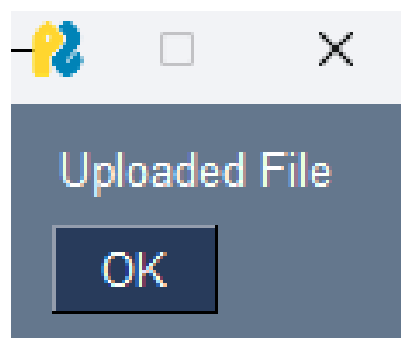
## 4. Menus

### 4.1 Menu Principal

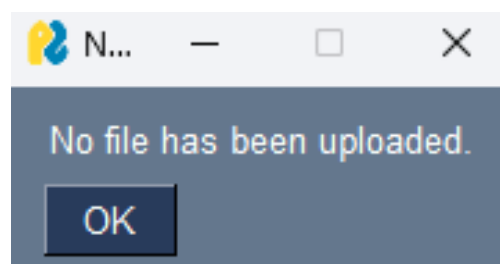
A primeira janela que aparece ao utilizador quando inicia o programa é a que se encontra a baixo. É a partir deste menu que se tem acesso a todas as funcionalidades do programa.



Quando o utilizador carregar o ficheiro a seguinte mensagem será exposta para ele:

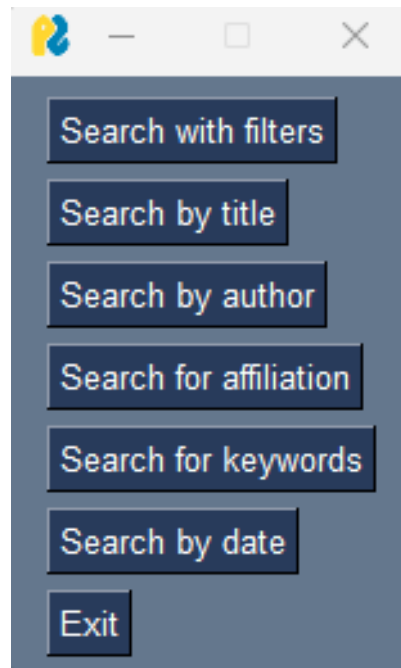


Caso contrário, ou seja, o utilizador não carregue nenhum ficheiro e tente selecionar outra opção que necessite do ficheiro um aviso será exibido para o mesmo:



#### 4.2 Menu de Pesquisa

Este menu é exibido quando o utilizador deseja procurar por publicações. É a partir daqui que o mesmo consegue selecionar o método para realizar a sua pesquisa.



### 4.3 Menu de Estatística

Quando o utilizador seleccionar o botão para visualizar as estatística o menu abaixo será apresentado. Aqui o utilizador pode escolher que tipo de estatística quer ver.



## Algoritmo

### Tecnologias Utilizadas

- Python
- PySimpleGUI
- Matplotlib
- JSON

### Importação de Bibliotecas e Módulos

Para o desenvolvimento de sistema foi necessária a importação de bibliotecas e módulos de modo a conseguir realizar o programa da forma mais eficiente possível. As bibliotecas e os módulos em questão são:

- `import PySimpleGUI as sg`
  - PySimpleGUI é uma biblioteca que simplifica a criação de interfaces gráficas em Python.
  - No contexto de projeto esta biblioteca foi utilizada para desenvolver toda a interface do sistema de forma clara e objetiva.
- `from datetime import datetime`
  - datetime é um módulo da biblioteca padrão do Python que fornece classes para manipulação de datas e horas.
  - No contexto do projeto este módulo foi utilizado para uma melhor manipulação das datas de forma a tornar o código mais eficiente.
- `import webbrowser`
  - webbrowser é um módulo da biblioteca padrão do Python que permite a interação com URL para abrir diretamente na Web.
  - No contexto do projeto foi utilizado para ser possível ter uma interação com os links fornecidos nas publicações.
- `import json`
  - json é um módulo da biblioteca padrão do Python que fornece uma maneira simples de codificar e decodificar dados no formato JSON (JavaScript Object Notation). Ele é usado para ler e escrever dados em arquivos JSON.
  - No contexto do programa, uma vez que o dataset se encontrava guardado em um ficheiro JSON, esta biblioteca foi usada para conseguir ler e escrever os dados das publicações
- `import matplotlib.pyplot as plt`
  - matplotlib.pyplot é um módulo da biblioteca Matplotlib que fornece uma interface de estilo MATLAB para a criação de gráficos e visualizações.
  - No contexto do programa este módulo da biblioteca Matplotlib foi usado para representar graficamente as estatísticas requeridas pela inuniciado.
- `from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg`
  - FigureCanvasTkAgg é uma classe do módulo matplotlib.backends.backend\_tkagg que permite integrar gráficos Matplotlib em interfaces Tkinter. Ele é usado para desenhar gráficos em um widget Tkinter Canvas
  - No contexto do programa este módulo foi necessário para apresentar os gráficos na interface da melhor forma possível.

## 1. Gestão de Ficheiros

### 1.1 Carregar ficheiros JSON

Para carregar o dataset presente no ficheiro JSON, com a estrutura anterior mente mencionada, foi utilizado código já desenvolvido nas aulas com algumas adaptações para a interface.

```
def carregar(caminho_ficheiro):
    try:
        with open(caminho_ficheiro, 'r', encoding='utf-8') as file:
            return json.load(file)
    except Exception as e:
        sg.popup_error(f"Error loading file: {e}")
        return None
```

## 1.2 Guardar publicações

Para guardar as publicações atualizadas ou criadas foi necessário desenvolver uma função para guardar os dados no mesmo ficheiro que foram carregados.

```
def guardar(caminho_ficheiro, dados):
    try:
        with open(caminho_ficheiro, 'w', encoding='utf-8') as file:
            json.dump(dados, file, indent=4)
            sg.popup("File saved successfully!")
    except Exception as e:
        sg.popup_error(f"Error saving file: {e}")
```

## 1.3 Exportar resultados de pesquisa

Para exportar resultados de uma pesquisa desenvolvemos duas maneiras possíveis para o fazer.

- Uma que permite exportar em formato txt com uma estrutura definida por nós

```
def guardar(caminho_ficheiro, dados):
    try:
        with open(caminho_ficheiro, 'w', encoding='utf-8') as file:
            json.dump(dados, file, indent=4)
            sg.popup("File saved successfully!")
    except Exception as e:
        sg.popup_error(f"Error saving file: {e}")
```

- Outra que permite exportar em formato de json com a mesma estrutura do dataset

```
elif event_save == "-SAVE_JSON-":
    caminho_arquivo_save = sg.popup_get_file("Save as", save_as=True,
no_window=True, file_types=(("JSON Files", "*.json"),))
    if caminho_arquivo_save:
        try:
            with open(caminho_arquivo_save, 'r', encoding='utf-8') as file:
                dados_existentes = json.load(file)
        except (FileNotFoundError, json.JSONDecodeError):
            dados_existentes = []

        if isinstance(dados, list):
            dados_existentes.extend(dados)
        else:
            dados_existentes.append(dados)

        guardar(caminho_arquivo_save, dados_existentes)
```

## 2. Gestão de Publicações

### 2.1 Adicionar novas publicações

Para adicionar novas publicações ao dataset desenvolvemos a seguinte função que recebe como argumentos `publicacoes`, que se refere à memória interna do dataset onde estão armazenadas todas as publicações, `values`, que são os dados obtidos pelo input do utilizador na interface, `autores`, que é uma lista de autores gerada por uma outra função, e `keywords`, que é uma string também criada por uma outra função onde as palavras-chave estão separadas por vírgulas.

```
def criar_publicacoes(publicacoes, values, autores, keywords):
    title = values["-TITLE-"].strip()
    abstract = values["-ABSTRACT-"].strip()
    doi = values["-DOI-"].strip()
    pdf = values["-PDF-"].strip()
    url = values["-URL-"].strip()
    publish_date = values["-PUBLISH_DATE-"].strip()

    if verificar_duplicados(publicacoes, title, doi):
        return False

    nova_publicacao = {
        "title": title,
        "abstract": abstract,
        "doi": doi,
        "pdf": pdf,
        "url": url,
        "publish_date": publish_date,
        "autores": autores,
        "keywords": keywords
    }

    publicacoes.append(nova_publicacao)
    return True
```

Primeiro, extraí os dados necessários dos `values` correspondentes, com a utilização do método `strip()` para remover quaisquer espaços em branco desnecessários. Em seguida, verifica se já existe alguma publicação com o mesmo título ou DOI através da função `verificar_duplicados`. Caso encontre duplicados, retorna `False` para indicar que esta publicação não pode ser adicionada.

Se não houver duplicados, cria um novo dicionário `nova_publicacao` com todos os dados fornecidos, incluindo os autores e palavras-chave. Este dicionário é então adicionado à lista de publicações existentes através do método `append()`. Por fim, retorna `True` para indicar que a publicação pode ser adicionada.

```
elif event == "-SAVE-":
    if not values["-TITLE-"].strip() and not values["-DOI-"].strip():
        sg.popup("Title or DOI is required!")
    else:
        if criar_publicacoes(publicacoes, values, autores, keywords):
```

```
if caminho_ficheiro:
    guardar(caminho_ficheiro, publicacoes)
    sg.popup("Publication successfully saved!")
    popup_running = False
else:
    sg.popup("No file selected.")
```

Este trecho de código mostra como o valor booleano é interpretado para guardar ou não a publicação dependendo se esta tem ou não um título ou DOI repetido. Também aqui tem uma verificação caso o utilizador não insira título nem DOI.

## 2.2 Atualizar Publicação

Para ser possível atualizar as informações de uma dada publicação existente no dataset nós desenvolvemos a seguinte função que recebe como argumentos `values`, que são os dados obtidos pelo input do utilizador na interface, `keywords`, que é uma string também criada por uma outra função onde as palavras-chave estão separadas por vírgulas, `autores`, que é uma lista de autores gerada por uma outra função, `publicacao_encontrada`, que são as informações de uma publicação que foi encontrada por uma outra função dados os dados de título ou DOI, `caminho_ficheiro`, que é o caminho guardado em memória onde as publicações com a publicação atualizada serão guardadas, e `publicacoes`, que se refere à memória interna do dataset onde estão armazenadas todas as publicações.

```
def atualizar_publicacao(values, keywords, autores, publicacao_encontrada,
                        caminho_ficheiro, publicacoes):
    title = values["-EDIT_TITLE-"].strip()
    doi = values["-EDIT_DOI-"].strip()

    if verificar_duplicados(publicacoes, title, doi, publicacao_encontrada):
        return

    updated_publication = {
        "title": values["-EDIT_TITLE-"].strip(),
        "abstract": values["-EDIT_ABSTRACT-"].strip(),
        "publish_date": values["-EDIT_DATE-"].strip(),
        "doi": values["-EDIT_DOI-"].strip(),
        "pdf": values["-EDIT_PDF-"].strip(),
        "url": values["-EDIT_URL-"].strip(),
        "keywords": keywords,
        "authors": autores
    }

    for key, value in updated_publication.items():
        publicacao_encontrada[key] = value

    if caminho_ficheiro:
        guardar(caminho_ficheiro, publicacoes)
        sg.popup("Publication successfully updated!")
    else:
        sg.popup("No file selected for saving")
```



Primeiro, assim com a função de adicionar uma nova publicação, extrai os dados necessários dos `values` correspondentes, com a utilização do método `strip()` para remover quaisquer espaços em branco desnecessários. Em seguida, assim como a outra também verifica se já existe alguma publicação com o mesmo título ou DOI através da função `verificar_duplicados`. Caso encontre duplicados, não retorna nada e nenhuma alteração é feita.

Se não houver duplicados, cria um novo dicionário `updated_publication` com todos os dados fornecidos, incluindo os autores e palavras-chave. Depois, a função atualiza a publicação original, substituindo cada campo pelos novos valores através de um ciclo que percorre todos os elementos do dicionário `updated_publication`. Por fim, se existir um caminho de ficheiro definido, a função guarda todas as alterações no ficheiro através da função `guardar`.

Ainda relacionado com a atualização de publicações, também desenvolvemos uma função para remover uma publicação do dataset. Esta recebe apenas três argumentos: `publicacao_encontrada`, `caminho_ficheiro` e `publicacoes`, já explicados anteriormente

```
def remover_publicacao(publicacao_encontrada, caminho_ficheiro, publicacoes):
    publicacoes.remove(publicacao_encontrada)
    sg.popup("Publication removed successfully!")
    guardar(caminho_ficheiro, publicacoes)
```

O seu funcionamento é direto e eficiente. Primeiro, utiliza o método `remove()` para eliminar a publicação selecionada da lista de publicações. Este método procura a publicação específica na lista e remove-a completamente. Por fim, a função `guardar` é chamada que atualiza o ficheiro guardado com a nova lista de publicações, já sem a publicação que foi removida.

## 2.3 Listagem

Como visto anteriormente na interface, nós desenvolvemos códigos para listar títulos, autores, afiliações e palavras-chave e que podem ser ordenados por ordem alfabética por frequência de ocorrência nas publicações e no caso dos títulos até mesmo por data de publicação.

```
def contar_frequencia_autores(publicacoes):
    author_frequency = {}
    for publicacao in publicacoes:
        for autor in publicacao.get("authors", []):
            name = autor.get("name", "")
            if name:
                if name not in author_frequency:
                    author_frequency[name] = 0
                author_frequency[name] += 1
    return author_frequency
```

Aqui tem um exemplo de como fizemos para calcular quantas vezes um dado autor aparece ao longo de todas as publicações. Esta função recebe como argumento `publicacoes`, que se refere à memória interna do

dataset onde estão armazenadas todas as publicações e e retorna um dicionário com a contagem de publicações por autor.

O seu funcionamento começa com a criação de um dicionário vazio chamado `author_frequency`, que irá armazenar o nome de cada autor como chave e o número de publicações como valor. Para cada publicação, a função examina a lista de autores associada. Utiliza o método `get()` para aceder à lista de autores, retornando uma lista vazia caso não existam autores. Para cada autor encontrado, extrai nome também. Caso seja a primeira vez que encontra o autor, cria uma nova entrada no dicionário com contagem inicial zero. Depois, incrementa a contagem para esse autor em uma unidade. Depois é só criar uma variável onde terá uma lista ordenada por frequência se utilizarmos como parâmetro de comparação a contagem no dicionário e por ordem alfabética se o parâmetro for os nomes no método `sorted()`.

Após calcular a frequência, são criadas duas estruturas adicionais. Uma lista com os nomes dos autores ordenada alfabeticamente, obtida com o método `sorted()` aplicado às chaves do dicionário. Outra lista de tuplos (`nome, frequência`), onde cada autor é associado ao número de publicações, ordenada de forma decrescente pela frequência. Esta lista é obtida utilizando o método `sorted()` com um critério de ordenação baseado na contagem. Sentimos a necessidade de criar esta lista de tuplos, uma vez que a contagem será útil para a estatística do projeto.

Para as afiliações e palavras-chave o código é muito semelhante ao explicado aqui apenas com algumas alterações no código das `keys` correspondentes no dicionário de cada elemento. No caso dos títulos uma vez que estes não são ordenados por ordem alfabética mas sim por data de publicação já existem diferenças significativas no código.

```
def ordenar_titulos(publicacoes):
    title_list = []
    for publicacao in publicacoes:
        title = publicacao.get("title", "")
        publish_date = publicacao.get("publish_date", "")

        if publish_date:
            try:
                data_publicacao = datetime.strptime(publish_date, "%Y-%m-%d")
            except ValueError:
                data_publicacao = None
            if data_publicacao:
                title_list.append((title, data_publicacao))
    ordered_title_list = [item[0] for item in sorted(title_list, key=lambda x:
x[0])]
    ordered_title_date = [item[0] for item in sorted(title_list, key=lambda x:
x[1])]
    ordered_title_date_rev = [item[0] for item in sorted(title_list, key=lambda x:
x[1], reverse=True)]
    return ordered_title_list, ordered_title_date, ordered_title_date_rev
```

Aqui a função `ordenar_titulos` organiza os títulos das publicações de três formas diferentes. Recebe a lista de publicações e começa por criar uma lista temporária que guarda tuplos com o título e a data correspondente a esse título. Para cada publicação que tenha data, a função tenta converter a data (que está em formato texto) para um objeto `datetime`, usando o formato `ano-mês-dia`. Se a conversão falhar, atribui

None à data. Por fim cria listas com ordenadas `ordered_title_list`: títulos ordenados alfabeticamente  
`ordered_title_date`: títulos ordenados por data, do mais antigo para o mais recente  
`ordered_title_date_rev`: títulos ordenados por data, do mais recente para o mais antigo.

## 2.4 Gráficos de estatística

Para possibilitar a visualização de gráficos criados pelo `Matplotlib` na nossa interface foi necessário que desenvolvessemos um função auxiliar.

```
def draw_figure(canvas, figure):
    figure_canvas_agg = FigureCanvasTkAgg(figure, canvas)
    figure_canvas_agg.draw()
    figure_canvas_agg.get_tk_widget().pack(side='top', fill='both', expand=1)
    return figure_canvas_agg
```

Esta função recebe dois argumentos: `canvas`, área de desenho, e `figura matplotlib`. Ela vai criar uma área de desenho específica para gráficos matplotlib em janelas tkinter, desenha a figura e ajusta-a para preencher todo o espaço disponível.

Um exemplo de uma função que cria um gráfico é a apresentada abaixo. Esta função recebe apenas o argumento `publicacoes` e tem a capacidade de gerar um gráfico utilizando a biblioteca `Matplotlib`.

```
def mostrar_distribuicao_palavras_chave(publicacoes):
    _, ordered_keywords = contar_frequencia_palavras_chave(publicacoes)

    top_20_keywords = [keyword for keyword, _ in ordered_keywords[:20]]
    keyword_counts = [count for _, count in ordered_keywords[:20]]

    fig, ax = plt.subplots(figsize=(10, 6))
    ax.barh(top_20_keywords, keyword_counts)
    ax.set_xlabel("Keywords Frequency")
    ax.set_ylabel("Keywords")
    ax.set_title("Distribution of keywords by frequency")
    plt.gca().invert_yaxis()
    plt.tight_layout()

    layout_keyword = [
        [sg.Text("Distribution of keywords by frequency", justification="center",
size=(30, 1))],
        [sg.Canvas(key="-CANVAS-")],
        [sg.Button("Exit", key="-EXIT-")]
    ]

    window_keyword = sg.Window("Distribution of keywords by frequency",
layout_keyword, modal=True, finalize=True)
    draw_figure(window_keyword["-CANVAS-"].TKCanvas, fig)

    keyword_running = True
    while keyword_running:
```

```
event_keyword, values_keyword = window_keyword.read()

if event_keyword in (sg.WINDOW_CLOSED, "-EXIT-"):
    keyword_running = False

window_keyword.close()
```

A função `mostrar_distribuicao_palavras_chave` gera um gráfico de barras horizontal que exibe as 20 palavras-chave mais frequentes nas publicações. Inicialmente, ela utiliza a função `contar_frequencia_palavras_chave` para obter uma lista ordenada de palavras-chave por frequência. Em seguida, seleciona as 20 palavras-chave mais frequentes e cria o gráfico utilizando o Matplotlib. Este gráfico é então mostrado em uma interface gráfica usando a função `draw_figure`, permitindo ao utilizador visualizar as palavras-chave mais comuns nas publicações.

## Linha de Comando

Foi também solicitado o desenvolvimento de uma linha de comandos com requisitos muito semelhantes aos anteriormente referidos, entre os quais:

- Criar Publicação
- Consultar Publicação
- Eliminar Publicação
- Relatório de Estatísticas
- Listar Autores
- Guardar Publicações

Tendo em conta que realizámos a maior parte da linha de comandos antes da interface gráfica, muitos dos códigos são semelhantes, com exceções, claro, uma vez que o algoritmo tem de ser ajustado quando os inputs são feitos na interface, mas toda a lógica geral das funções de ambos é idêntica.

Para manter o terminal mais limpo e facilitar a leitura da informação, utilizámos o módulo `import os`, que é uma biblioteca padrão do Python e fornece funcionalidades para interagir com o sistema operativo. No contexto deste projeto, o módulo `import os` foi utilizado para realizar tarefas como limpar a tela do terminal e garantir que o sistema seja capaz de identificar e manipular caminhos de ficheiros de forma compatível com diferentes sistemas operativos (Windows, Linux, etc.).

## Funcionalidades

### 1. Menu Principal

O menu principal é muito semelhante ao menu principal da interface gráfica como é possível ver na imagem a baixo.

```
=== Main Menu ===
1. Upload file
2. Add publication
3. Update publication
4. Search publications
5. View Docuemnt
6. Publication Statistics
0. Exit

Choose an option: █
```

## 2. Adicionar Publicação

O menu que desenvolvemos para adicionar uma publicação é o que se encontra abaixo. Neste menu, será pedido, por ordem, que cada parâmetro seja preenchido. No caso das palavras-chave e dos autores, pode-se adicionar quantos quisermos, e, quando não quisermos adicionar mais nenhum, basta pressionar ENTER.

```
=== Create New Publication ===
Title: TESTE
Abstract: TESTE
Enter keywords (press Enter with empty input to finish):
Keyword: TESTE
Keyword:
Author name (or ENTER to finish): TESTE
Affiliation: TESTE
Author name (or ENTER to finish):
DOI: TESTE
Publication date (YYYY-MM-DD): 2025-01-01
URL: TESTE
```

## 3. Atualizar Publicação

O menu que desenvolvemos para atualizar as informações de uma publicação de forma clara é o seguinte. Primeiramente, será solicitado o título ou o DOI da publicação que deseja procurar. Após isso, várias opções aparecerão, e basta selecionar a que desejamos alterar. O conteúdo atual será impresso no terminal e, em seguida, será pedido que seja inserido o novo conteúdo.

```
=== Update Publication ===

Search publication by:
1. Title
2. DOI
Choose option: 1
Enter title: O Teste do Reflexo Vermelho do Olho e a Leucocória na Criança

Update options:
1. Title
2. Abstract
3. Keywords
4. Authors
5. DOI
6. Publication date
7. URL
8. Delete publication
0. Return to menu
Choose field to update: 1
Current title: O Teste do Reflexo Vermelho do Olho e a Leucocória na Criança
New title: TESTE
```

#### 4. Procurar Publicação

No menu de pesquisa de publicações, inicialmente o utilizador será questionado sobre qual parâmetro deseja utilizar para a pesquisa. Após isso, um segundo menu será exibido, permitindo que o utilizador escolha como deseja visualizar a listagem do parâmetro selecionado: por ordem alfabética, por frequência ou, no caso dos títulos, por data de publicação.

```
=== Search Publication ===

Search by:
1. Title
2. Author
3. Affiliation
4. Keywords
5. Date
0. Close
Choose an option: 1

Title options:
1. Sort by newest first
2. Sort by oldest first
3. Sort alphabetically
4. Search specific title
0. Close
Choose an option: █
```

A seguir, uma lista será impressa no terminal de acordo com a escolha do utilizador. Assim, o utilizador poderá ver os nomes disponíveis, selecionar a opção de pesquisa no menu anterior e inserir o que deseja procurar.

```
Page 1/360 (Total items: 3597)
-----
1. TESTE
2. Caso Fatal de Miocardite e Miosite Imunomediadas Associadas ao Tratamento com Inibidores de Checkpoint Imunitário e de Tirosina-Quinase
3. Ataxia Espinocerebelosa Tipo 27B (SCA27B): Uma Ataxia Hereditária em Portugal
4. Desprescrição nos Idosos: Atitudes, Conhecimento, Formação e Prática Clínica dos Médicos Portugueses
5. Regressão Logística: Limitações na Estimção de Medidas de Associação com Desfechos de Saúde Binários
6. Avaliação da Carga da Doença Atribuída à Dermate Atópica em Portugal através de Experiências Centradas no Doente
7. Exposição ao Fumo Ambiental do Tabaco em Portugal Após a Implementação da Proibição de Fumar: Uma Revisão Sistemática
8. Hospitalizações por Perturbação Esquizofreniforme: Uma Análise Clínica e Demográfica de uma Base de Dados Nacional
9. Usabilidade da APIMedOlder: Uma Aplicação Web para a Gestão de Medicamentos Potencialmente Inapropriados em Idosos
10. Recomendações para a Redução do Impacto Ambiental dos Inaladores em Portugal: Documento de Consenso

Navigation:
n - Next page
p - Previous page
q - Return to menu

Enter choice: █
```

5. Vizualização das publicações

Existe uma opção no menu principal para visualizar todas as publicações do documento. No entanto, a estrutura em que elas aparecem é exatamente a mesma que é utilizada para visualizar, por exemplo, as publicações encontradas após uma pesquisa. Se houver mais de uma publicação, é possível navegar entre elas utilizando as opções apresentadas ao utilizador.

```
Publication 1/3597
-----
Title: Hospitalizações por Perturbação Esquizofreniforme: Uma Análise Clínica e Demográfica de uma Base de Dados Nacional
Abstract: Resumo Introdução: A perturbação esquizofreniforme manifesta-se com sintomas semelhantes aos da esquizofrenia, mas distingue-se desta pela s
O objetivo deste estudo foi descrever e analisar as hospitalizações por perturbação esquizofreniforme numa base de dados nacional de internamentos. Mét
ase de dados de hospitalizações a nível nacional, contendo todos os internamentos registados em hospitais públicos de Portugal Continental entre 2008
erturbação esquizofreniforme foram selecionadas com base no código de diagnóstico 295.4x da Classificação Internacional de Doenças versão 9, Modificaç
ata de nascimento, ao sexo, à zona de residência, aos diagnósticos, à duração do internamento (length of stay), ao estado de alta e às despesas hospit
pontuação do índice de Charlson. Foram efetuados testes t para amostras independentes para avaliar diferenças em variáveis contínuas com distribuição
não foi registada. Resultados: Nos hospitais públicos de Portugal Continental, ocorreu um total de 594 internamentos com o diagnóstico primário de per
s selecionado. A maioria, 72,1% (n = 428), estava associada a doentes do sexo masculino. A idade média de admissão foi de 34,34 anos nos homens e de 4
de 17,00 dias e a mortalidade intra-hospitalar foi de 0,5% (n = 3). Apenas 6,1% (n = 36) dos episódios de internamento tinham uma ou mais comorbilidade
clusão: As hospitalizações com o diagnóstico principal de perturbação esquizofreniforme ocorrem mais frequentemente em doentes jovens do sexo masculin
primeiro estudo de âmbito nacional que analisa todos os internamentos por este diagnóstico em Portugal.
Keywords: Hospitalização, Perturbações Psicóticas

Authors:
- Inês Teixeira da Cunha (Faculty of Medicine, Universidade do Porto, Porto.)
- Celeste Silveira (Psychiatry Service, Unidade Local de Saúde São João, Porto.)
- Alberto Freitas (CINTESIS@RISE, Department of Community Medicine, Information and Health Decision Sciences (MEDCIDS), Faculty of Medicine, Universid
- Manuel Gonçalves Pinho (CINTESIS@RISE, Department of Clinical Neurosciences and Mental Health, Faculty of Medicine, Universidade do Porto, Porto; De
de Saúde Tâmega e Sousa, Penafiel.)
DOI: https://doi.org/10.20344/amp.21714
Date: 2024-09-06
URL: https://www.actamedicaportuguesa.com/revista/index.php/amp/article/view/21714

Navigation:
n - Next publication
p - Previous publication
q - Return to menu

Enter choice: █
```

6. Estatística das Publicações

Por último, existe o menu para consultar as estatísticas, como o número de publicações por ano, o número de publicações por autor e as palavras-chave mais comuns no dataset.

```
=== Statistics Menu ===
1. Publications by year
2. Publications by author
3. Most common keywords
0. Return to menu

Choose an option: █
```

Selecionando qualquer uma das opções, o seguinte relatório estatístico será exibido para o utilizador, permitindo-lhe navegar utilizando as opções do menu.

```
Page 1/2 (Total items: 18)
-----
1. 2002: 1 publications
2. 2006: 1 publications
3. 2007: 12 publications
4. 2011: 24 publications
5. 2012: 90 publications
6. 2013: 116 publications
7. 2014: 133 publications
8. 2015: 117 publications
9. 2016: 121 publications
10. 2017: 125 publications

Navigation:
n - Next page
p - Previous page
q - Return to menu

Enter choice: █
```

## Desafios e Soluções

Um dos primeiros desafios que nos deparamos foi perceber todas as possibilidades que o **PySimpleGUI** nos possibilitava para organizar da melhor maneira possível todas as funcionalidades que desenvolvemos. Em relação à parte do algoritmo em si, algumas dificuldades que encontramos foram nomeadamente adaptar o código que tinha sido desenvolvido para a linha de comando para os inputs e outputs da interface e descobiri qual seria a melhor maneira de inserir os gráficos criados pelo **Matplotlib** na interface. Foi necessário realizar alguma pesquisa e estudo para superar estes objetivos, mas no final o sistema ficou exatamente como pretendíamos.

## Conclusão

Neste projeto, não só aplicámos os conhecimentos e habilidades adquiridos ao longo das aulas da Unidade Curricular de Algoritmos e Técnicas de Programação, como também desenvolvemos a nossa capacidade de aprendizagem autónoma, uma vez que surgiram necessidades de investigar e pesquisar novos conteúdos e informações. Além disso, tivemos a oportunidade de aprimorar a nossa competência na resolução de problemas, aprendendo a aplicá-la em situações práticas. Em resumo, apesar dos desafios e dificuldades enfrentados durante o desenvolvimento do projeto, consideramos ter alcançado os requisitos e objetivos propostos.