

Relatório Final

Realizado por : João Silva A103919, Leonor Matos A103920, Maria Miguel A103906.

CONTEÚDO

1. Introdução
 2. Metodologia
 3. Funcionamento da aplicação
 1. Descrição Informal do projeto
 2. Funções
 1. Carregar dados;
 2. Guardar dados;
 3. Criar Publicação;
 4. Consultar Publicação;
 5. Listar Publicações;
 6. Listar Autores;
 7. Eliminar Publicação;
 8. Gerar Relatório;
 3. Layouts
 1. Layout Inicial
 2. Layout Criar Publicação
 3. Layout Consultar Publicação
 4. Layout Atualizar Publicação
 5. Layout Listar Publicações
 6. Layout Listar Autores
 7. Layout Analisar Palavras-chave
 8. Gerar Relatório
 4. Conclusão
-

1. INTRODUÇÃO

Este projeto foi realizado no âmbito da Unidade Curricular de Algoritmos e Técnicas de Programação, com o intuito de criar um sistema que permite criar, atualizar e analisar publicações científicas, além de gerar relatórios com estatísticas úteis. Com base num dataset de publicações, o sistema possibilita a pesquisa de artigos usando filtros relevantes, tais como a data de publicação, as palavras-chave, autores, etc. Neste projeto, tivemos como objetivo atender a todas as funções pedidas, tentando solucionar eventuais erros que poderiam surgir aquando da utilização da aplicação pelo utilizador.

2. METODOLOGIA

Para a realização deste trabalho foi utilizado o editor "Visual Studio Code" e o "Jupyter notebook" em python 3.11.7 64-bit. Para o desenvolvimento da interface gráfica para a aplicação das funções criadas foi utilizado "PySimpleGUI". De forma a facilitar o desenvolvimento do projeto foi utilizado o documento:

"publicacoes.json" que contém todas as publicações associadas ao trabalho. O nosso código encontra-se organizado em duas secções: "linha de comando" e "interface". Esta divisão permite visualizar de forma mais clara o objetivo de cada funcionalidade. Dentro da "linha de comando" ocorre uma separação entre duas principais classes: "TaskManager" que é responsável por gerenciar as operações com publicações (carregar, guardar, criar, eliminar, etc.) e "Menu" que é a interface da linha de comando que vai conectar o utilizador às funcionalidades da aplicação.

3. FUNCIONAMENTO DA APLICAÇÃO

1. Descrição informal do Projeto

Este projeto tem por base o desenvolvimento de uma aplicação de manipulação de dados. Assim o sistema possui a capacidade de criar, atualizar e analisar publicações científicas, de forma a que permita ao utilizador a pesquisa de artigos usando filtros relevantes. Deste modo, foi desenvolvido um sistema possuir das seguintes opções: "help", "criar publicação", "consulta de publicação", "conuslta de publicações", "eliminar publicação", "relatório de estatísticas", "listar autores".

2. Funções

1. Carregar Dados

Primeiramente, para trabalhar com a base de dados que nos foi fornecida implementamos uma função que carrega todas os dados do ficheiro JSON de modo a que estes possam ser utilizados no restante trabalho. Caso o arquivo JSON não exista é retornada uma lista vazia.

```
def carregar_data(self):
    try:
        with open(self.data_file, 'r', encoding='utf-8') as file:
            self.data = json.load(file)
    except FileNotFoundError:
        return []
```

2. Guardar Dados

Criamos uma função capaz de guardar todas as alterações que são feitas no nosso sistema JSON. Este passo é crucial para que consigamos manter a persistência dos dados.

```
def save_data(self):
    with open(self.data_file, 'w', encoding='utf-8') as file:
        json.dump(self.data, file, indent=4)
```

3. Criar Publicação

Com a criação desta funcionalidade, solicitamos ao utilizador que insira os dados referentes ao título, resumo, palavras-chave, DOI, data de aplicação e URL, bem como as respectivas informações sobre os

autores, como o seu nome e afiliação. Os dados são, então, adicionados à lista de publicações e guardados no ficheiro JSON.

```
def criar_publicacao(self):
    new_publication = {
        "title": input("Título: "),
        "abstract": input("Resumo: "),
        "keywords": input("Palavras-chave (separadas por vírgula): "),
        "authors": [],
        "doi": input("DOI: "),
        "publish_date": input("Data de publicação (YYYY-MM-DD): "),
        "url": input("URL: ")
    }

    num_authors = int(input("Número de autores: "))
    for _ in range(num_authors):
        autor = {
            "name": input("Nome do autor: "),
            "affiliation": input("Afiliação: ")
        }
        new_publication["authors"].append(autor)

    self.data.append(new_publication)
    self.save_data()
    print("Publicação criada com sucesso!")
```

4. Consultar Publicação

Com a função "consultar publicação" permitimos ao utilizador que consulte publicações com base no seu título. Caso existam correspondências, as informações serão exibidas em detalhe (título, resumo e autores).

```
def consultar_publicacao(self):
    while True:
        title = input("Insira o título ou parte dele: ").lower()
        results = [pub for pub in self.data if title in
pub["title"].lower()]

        if results:
            for pub in results:
                print("-----")
                print(f"Título: {pub['title']}\n \nResumo:
{pub['abstract']}\n \nAutores: {[author['name'] for author in
pub['authors']]}\n-----\n")
                break
            else:
                print("Nenhuma publicação encontrada. Deseja tentar
novamente?")
```

```
if input().lower() != 'sim':  
    break
```

5. Listar Publicações

De modo a conseguirmos listar publicações, o nosso sistema tem funções responsáveis pela listagem de todos dos títulos, datas e autores de cada respectiva publicação.

```
def listar_publicacoes(self):  
    for pub in self.data:  
        print(f"Título: {pub['title']}\nData de publicação:  
{pub['publish_date']}\nAutores: {[author['name'] for author in  
pub['authors']]}\n")
```

6. Listar Autores

Para que nos fosse possível organizar as publicações por autores, criamos uma função que exibe o nome de cada autor e as suas respetivas publicações.

```
def listar_autores(self):  
    autores = {}  
    for pub in self.data:  
        for autor in pub["authors"]:  
            if autor["name"] not in autores:  
                autores[autor["name"]] = []  
            autores[autor["name"]].append(pub["title"])  
  
    for autor, artigos in sorted(autores.items()):  
        print(f"Autor: {autor}\nPublicações: {' '.join(artigos)}\n")
```

7. Excluir Publicação

Com esta função conseguimos remover uma publicação da nossa base de dados com base no seu título.

```
def delete_publicacao(self):  
    while True:  
        title = input("Insira o título da publicação a ser removida:  
").lower()  
        for i, pub in enumerate(self.data):  
            if title in pub["title"].lower():  
                print(f"Removendo publicação: {pub['title']}")  
                del self.data[i]  
                self.save_data()  
                print("Publicação removida com sucesso.")  
                return  
        print("Publicação não encontrada. Deseja tentar novamente?")
```

```
if input().lower() != 'sim':  
    break
```

8. Gerar Relatório

Para que conseguíssemos criar estatísticas sobre as publicações criamos esta aplicação que nos mostra quais foram as 20 palavras-chaves mais usadas, quais os 20 autores com mais publicações e as distribuições de publicações por ano.

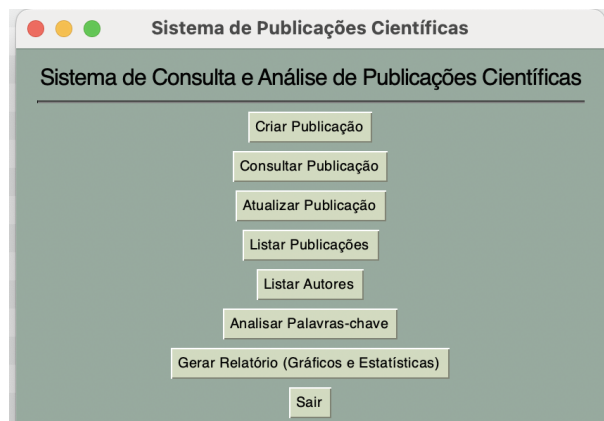
```
def gerar_relatorio(self):  
    if not self.data:  
        print("Nenhuma publicação disponível no sistema.")  
        return  
  
    print("\n-----Relatório de Estatísticas-----")  
    # Frequência de palavras-chave  
    keywords = [keyword for pub in self.data for keyword in  
pub["keywords"].split(",")]  
    keywords_freq = Counter(keywords).most_common(20)  
    print("\nPalavras-chave mais frequentes:")  
    for keyword, freq in keywords_freq:  
        keyword_limpo = keyword.replace(",", "")  
        print(f"{keyword_limpo}: {freq} vezes")  
  
    # Número de publicações por autor  
    autores = [author["name"] for pub in self.data for author in  
pub["authors"]]  
    autores_freq = Counter(autores).most_common(20)  
    print("\nAutores com mais publicações:")  
    for autor, freq in autores_freq:  
        print(f"{autor}: {freq} publicações")  
  
    # Número de publicações por ano  
    anos = [datetime.strptime(pub["publish_date"], "%Y-%m-%d").year  
for pub in self.data]  
    anos_freq = Counter(anos).most_common()  
    print("\nNúmero de publicações por ano:")  
    for ano, freq in anos_freq:  
        print(f"{ano}: {freq} publicações")  
    print("-----")
```

3. Layouts

Nesta parte do nosso código utilizamos o "PySimpleGUI" em Python de modo a criar um layout de uma interface gráfica.

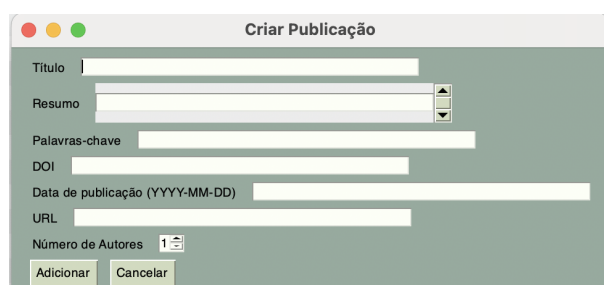
1. Layout Inicial

Começamos por criar um layout com todos os botões necessários para a utilização da nossa aplicação, sendo estes: "Criar Publicação", "Consultar Publicação", "Atualizar Publicação", "Listar Publicações", "Listar Autores", "Analisar Palavras-chave", "Gerar Relatório" e "Sair". Assim, criamos um layout simples de oito botões, onde cada botão irá executar uma função específica quando pressionado, como adicionar, consultar, atualizar, listar e analisar publicações bem como fornecer um relatório com gráficos e estatísticas.



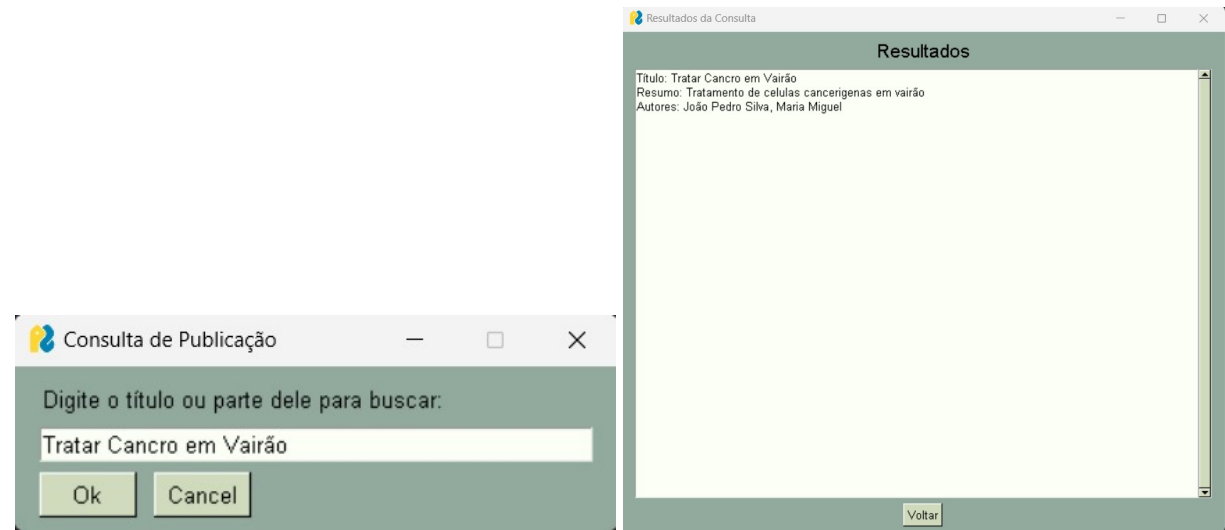
2. Layout Criar Publicação

Para esta parte do nosso código criamos um layout para a criação de uma nova publicação. As funções utilizadas servem para criar campos de entrada de texto, como por exemplo, "Título da publicação", local para colocarmos o nome da publicação; "Resumo" que é um síntese da publicação para que se possa ter uma descrição sucinta da publicação; "Palavras-chave" que são as palavras que melhor caracterizam a publicação para que esta possa ser acedida mais rapidamente; "DOI" para colocarmos o link da publicação em formato PDF (inalterável); "Data da publicação" em que escrevemos o ano seguido do mês e do dia (YYYY-MM-DD); "URL" para colocarmos o link do documento sem ser em PDF (pode sofrer atualizações/alterações) e por fim também existirá a opção de selecionar o "Número de Autores" da publicação. Se o utilizador desejar voltar ao menu inicial, poderá carregar no botão "Cancelar".



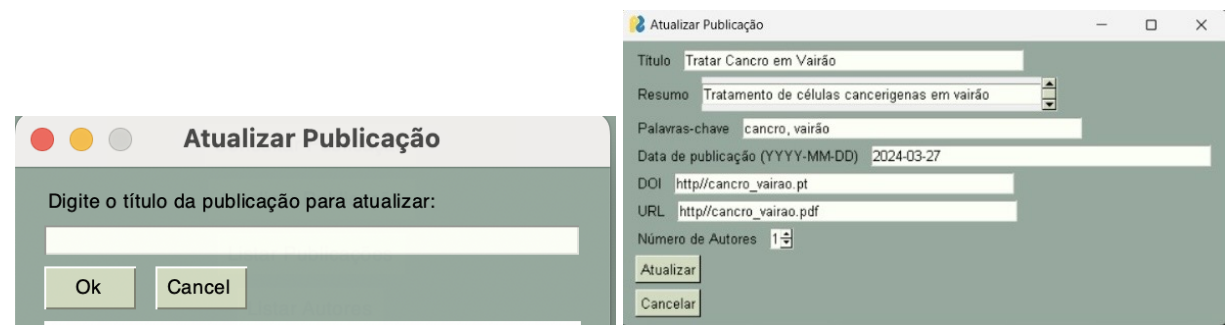
3. Layout Consultar Publicação

De modo a que o utilizador seja capaz de consultar uma publicação específica criamos um layout destinado à consulta de todas as publicações existentes. Para o utilizador conseguir encontrar a publicação pretendida terá que introduzir o título da publicação ou uma parte dele.



4. Layout Atualizar Publicação

De modo a que o utilizador seja capaz de alterar alguns dos dados da publicação criamos um layout destinado à atualização de informações de uma publicação já existente. O utilizador pode selecionar uma publicação através do título e depois será disponibilizado um layout igual ao layout "Criar Publicação", onde poderá alterar o título, o resumo da publicação, as palavras-chave, a data de publicação, DOI, URL e ainda o número de autores da publicação. Para guardar as alterações, basta carregar no botão "Atualizar" e a atualização da informação da publicação em questão será realizada com sucesso. Caso o utilizador decida que afinal não quer alterar a publicação, pode carregar no botão "Cancelar" e voltará ao menu inicial.



5. Layout Listar Publicações

De modo a que o utilizador seja capaz de visualizar todas as publicações existentes, criamos um layout destinado à listagem de todas as publicações. Para cada publicação, é possível visualizar o título, data de publicação e autores. Se o utilizador pretender voltar ao menu inicial, basta carregar no botão "Voltar".



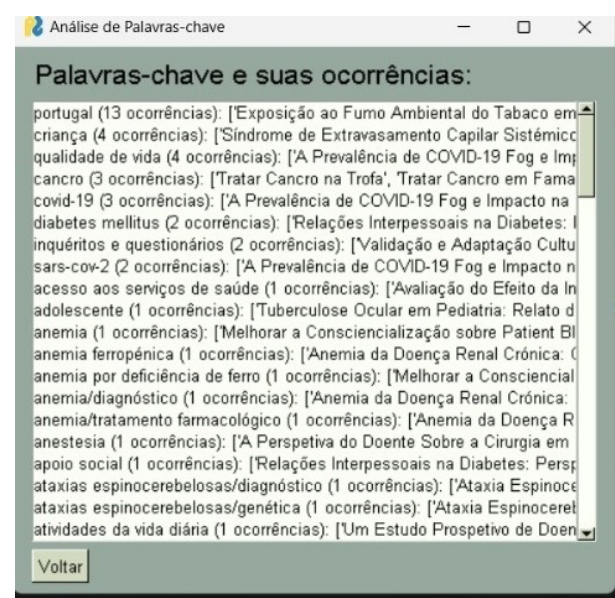
6. Layout Listar Autores

De modo a que o utilizador seja capaz de visualizar todos os autores das várias publicações existentes, criamos um layout destinado à listagem de todos os autores. Para cada autor, é possível visualizar o título da publicação que escreveu. Se o utilizador pretender voltar ao menu inicial, basta carregar no botão "Voltar".



7. Layout Analisar Palavras-chave

De modo a que o utilizador seja capaz de visualizar as 20 palavras-chave mais utilizadas, criamos um layout destinado à análise das mesmas. Para cada uma delas, é possível visualizar o seu número de ocorrências e as publicações que são definidas através dessa palavra-chave. Se o utilizador pretender voltar ao menu inicial, basta carregar no botão "Voltar".



8. Layout Gerar Relatório

O dashboard gráfico foi criado com o intuito de facilitar a visualização e compreensão de alguns dados. Assim, o utilizador tem acesso a gráficos estatísticos que melhoram a sua experiência durante a utilização da interface. Assim, temos acesso a: "Top 20 Palavras-chave", ou seja, as 20 palavras-chave mais utilizadas; "Top 20 Autores com mais Publicações"; "Distribuição de Publicações por Ano", que nos permite verificar, por exemplo, qual foi o ano com mais publicações; "Distribuição de Publicações por Mês em 2024", sendo possível visualizar os meses com maior número de textos publicados e "Top 20 Palavras-chave em 2024".





4. CONCLUSÃO

Este projeto foi desenvolvido com base nos conhecimentos adquiridos na UC de Algoritmos e Técnicas de Programação. O principal objetivo era criar um sistema e uma interface gráfica que permitisse a exploração e manipulação de uma base de dados. Durante a execução do trabalho, deparamo-nos com algumas dificuldades no funcionamento global da interface, o que nos levou a ajustar e adicionar funções conforme necessário. Apesar dos desafios encontrados, acreditamos ter atendido a todos os requisitos e propostas apresentados. As dificuldades encontradas foram, de certa forma, benéficas, pois impulsionaram o aprimoramento das nossas habilidades de programação em Python.