

Relatório Final

UC- Algoritmos e técnicas de Programação

Nome - Sistema de Consulta e Análise de Publicações Científicas

Docentes - José Carlos Ramalho e Luís Filipe Cunha

Patricia Gomes (A105238, a105238.uminho.pt)

Joana Santos (A103904, a103904@uminho.pt)

Marta Silva (A92041, a92041.uminho.pt)

5 de janeiro de 2025

Conteúdo

- 1. Introdução 1
- 2. Descrição do Projeto 1
 - 2.1. Requisitos técnicos 2
 - 2.2. Principais Objetivos do Projeto 2
- 3. Concepção e desenho da aplicação 3
 - 3.1. Estrutura de Dados utilizados 3
 - 3.2. Explicação do código (parte gráfica) 4
- 4. Conclusão 8

Introdução

O presente relatório descreve a elaboração do projeto, cuja finalidade é o desenvolvimento de uma aplicação que funcione como um sistema de consulta e análise de publicações científicas.

O sistema foi desenvolvido utilizando o Python com o uso de estruturas de dados adequadas, nomeadamente listas e dicionários, além do uso de bibliotecas como o **matplotlib**. Para uma interação mais dinâmica, utilizamos uma interface gráfica com o módulo **PySimpleGUI**, que permite visualizar a aplicação através de um menu que contém todas as funções disponíveis na linha de comandos.

Desta forma, o utilizador pode usufruir de todas as componentes criadas no código da aplicação para a consulta e análise das publicações que deseje.

Descrição do Projeto

O projeto a desenvolver na edição 2024/25 da unidade curricular tem como objetivo criar uma aplicação que permita ao utilizador poder aceder a publicações através de uma pesquisa dinâmica, intuitiva e eficiente para que encontre o que necessita.

Requisitos Técnicos do Sistema

Os requisitos técnicos deste sistema incluem a criação de funções para:

Carregamento da Base de Dados

O programa, no arranque, deverá carregar para a memória o dataset que estará guardado no ficheiro de suporte à aplicação.

Criação de Publicações

O utilizador deve poder criar um artigo especificando:

- Título
- Resumo
- Palavras-chave
- DOI
- Lista de autores e sua afiliação correspondente
- URL para o ficheiro PDF do artigo
- Data de publicação
- URL do artigo

Atualização de Publicações

O sistema deve permitir a atualização da informação de uma publicação, incluindo:

- Data de publicação
- Resumo
- Palavras-chave
- Autores e afiliações

Consulta de Publicações

O sistema deve permitir pesquisar publicações com filtros por:

- Título
- Autor
- Afiliação
- Data de publicação
- Palavras-chave

Além disso, deve ser possível ordenar as publicações encontradas pelos títulos ou pela data de publicação.

Análise de Publicações por Autor

O sistema deve:

- Listar os autores
- Permitir o acesso aos artigos de cada autor da lista
- Ordenar os autores pela frequência dos seus artigos publicados e/ou por ordem alfabética

Análise de Publicações por Palavras-chave

O sistema deve:

- Permitir a pesquisa e visualização das palavras-chave do dataset
- Ordenar as palavras-chave pelo número de ocorrências nos artigos e/ou por ordem alfabética
- Visualizar a lista das publicações associadas a cada palavra-chave

Principais Objetivos do Projeto

Os principais objetivos deste projeto são:

- Implementação de uma interface de linha de comando (CLI) e uma interface gráfica para a interação do utilizador com a aplicação
- Utilização de estruturas de dados adequadas para facilitar a manipulação de dados
- Integração de funcionalidades de gráficos através do uso de bibliotecas Python, como o **matplotlib**
- Garantir que os dados estejam armazenados em um mecanismo de armazenamento, como um ficheiro **JSON**

Concepção do Desenho do Projeto

Antes do começo do desenvolvimento da aplicação, os elementos do grupo fizeram um esboço para organizar as ideias e garantir uma abordagem coerente. Este planeamento permitiu determinar as principais funcionalidades do programa e identificar as ferramentas necessárias para que o utilizador pudesse tirar o máximo proveito da aplicação.

Como parte deste processo, desenvolvemos um protótipo básico do programa, estabelecido num Jupyter Notebook, para explorar e testar as funcionalidades essenciais da aplicação. Durante a conceção deste esboço, concluímos que o programa seria constituído, sobretudo, por **dicionários** e **listas**, de forma a facilitar a manipulação dos dados.

Estrutura de Dados

O código é composto por **Listas** e **Dicionários** para armazenamento e manipulação de dados relacionados com as publicações científicas.

Listas:

- **mybd**: É uma lista que armazena todas as publicações carregadas do ficheiro JSON principal (`ata_medica_papers.json`). Cada elemento da lista é um dicionário representando uma publicação com atributos como título, resumo, palavras-chave, autores, etc.
- **autores**: Na função `Criar_Pub_Layout`, os autores de cada publicação são armazenados numa lista de dicionários onde cada autor corresponde a um dicionário, com detalhes como nome, afiliação e ORCID, permitindo adicionar e manipular vários autores.
- **pubs_removidas**: Na função `Apagar_Pub`, definimos uma lista em compreensão que nos permite filtrar as publicações que não queremos manter no ficheiro principal (base de dados). Todas as publicações que se encontram na lista serão transferidas para um segundo ficheiro (`pubs_removidas.json`), permitindo recuperar informações eliminadas se o utilizador desejar.

- **bd_atualizada**: Na função **Apagar_Pub**, definimos uma lista em compreensão que permite atualizar o ficheiro principal sem as publicações eliminadas pelo utilizador.
- **pub_dupla**: É uma lista utilizada na função **Listar_Pub_Afil** para garantir que não há publicações repetidas em consultas relacionadas a afiliações. Quando corremos a função serão armazenados os títulos das publicações já processadas.
- **lista_palavras**: Na função **Distribuição_PC**, as palavras-chaves associadas a uma publicação são separadas por vírgulas e serão processadas como elementos na lista, facilitando a sua contagem e ordenação.
- **novos_registos**: Na função **Importar_Pub**, a lista referida será utilizada para armazenar publicações carregadas de um ficheiro externo escolhido pelo utilizador, permitindo validar e transferir esses registos para a base de dados principal.

Dicionários:

Cada publicação é representada por um dicionário composto pelos seguintes campos:

- **"title"**: Título da publicação.
- **"abstract"**: Resumo da publicação.
- **"keywords"**: Palavras-chave associadas.
- **"authors"**: Lista de autores com informações detalhadas.
- **"doi"**: DOI da publicação.
- **"pdf"** e **"url"**: Links para o PDF e o artigo online.
- **"publish_date"**: Data de publicação.

Além disso, cada autor também é representado como um dicionário dentro de uma lista, com atributos como:

- **"name"**: Nome do autor.
- **"affiliation"**: Afiliação do autor.
- **"orcid"**: ORCID do autor.

Arquivos JSON

A biblioteca **JSON** é utilizada para salvar e guardar dados das publicações em arquivos JSON.

Ao longo da concepção deste sistema, para todas as funcionalidades responsáveis por uma alteração da informação contida na base de dados foi utilizado o **context manager constructor** designado como **with open()** disponível em Python, permitindo que qualquer ficheiro aberto para leitura ou escrita seja automaticamente fechado após a ocorrência de **json.load()** ou **json.dump()**. Graças a este mecanismo todas as alterações registadas no ficheiro principal serão salvas de uma forma segura.

Explicação do Código (Parte Gráfica)

O código delineado pelos membros do grupo contém diversas aplicações específicas que garantem o seu bom funcionamento. A cada uma das respetivas funções é associada uma nova janela visualizável na interface gráfica.

Criação da Janela Inicial da Aplicação:

Função `criar_layout`

Criamos o layout principal da aplicação usando a biblioteca PySimpleGUI. O layout é composto quatro colunas de botões, a cada um está associada uma função definida quer na interface da linha de comandos (CLI), quer na interface gráfica.

Salvar Novas Publicações:

Função `Salvar_Pub (Criar_Pub_Layout)`

Os dados inseridos pelo usuário nos respetivos campos de input visíveis na janela "Criar Nova Publicação" são processados de modo a criar uma nova publicação, estruturando-os num dicionário que é então salvo na base de dados.

Apagar Publicações da Base de Dados:

Função `Apagar_Pub_Layout`

O utilizador possui a capacidade de apagar uma publicação da base de dados ao fornecer o seu respetivo título. A função irá verificar se o título inserido corresponde a alguma publicação na ficheiro principal do sistema, exibe mensagens apropriadas e usa uma função lógica (`Apagar_Pub`) para efetuar a eliminação da publicação se o utilizador assim desejar fazê-lo. A janela "Apagar Publicação" apresenta o local adequado para o utilizador providenciar a informação correspondente ao título da publicação a eliminar.

Atualizar Informação das Publicações:

Função `Atualizar_Pub_Layout`

O utilizador é capaz de realizar alterações à informação contida numa publicação específica. Na janela "Atualizar Publicação" é inicialmente pedido o título da publicação a modificar. Após ser encontrada na base da dados, serão apresentados na janela os detalhes dessa publicação, como o seu título, abstract, data de publicação, DOI, entre outros. A alteração bem-dita das informações deve ser fornecida nos campos de input visíveis na janela sendo, de seguida, salvas no ficheiro principal. Esta funcionalidade permite a atualização de informação da publicação geral, bem como a alteração dos campos de um respetivo autor. É também possível criar novos campos gerais à publicação aquando da sua anterior inexistência.

Consulta de Publicações:

Função `Consultar_Title_Layout`

Permite ao usuário pesquisar publicações pelo seu título e exibir os detalhes completos da publicação encontrada.

As seguintes funções seguem a mesma estrutura da função mencionada anteriormente:

- `Consultar_PDF_Layout`: Consulta por link PDF.
- `Consultar_DOI_Layout`: Consulta por link DOI.
- `Consultar_URL_Layout`: Consulta por link URL.

Todas as funções referidas utilizam critérios de pesquisa diferentes, porém seguem o mesmo fluxo de exibição dos detalhes das publicações encontradas.

Consultar Autores na Base de Dados:

Função `Listar_Autores_Layout`

A função tem como objetivo apresentar uma lista completa dos autores presentes no ficheiro principal, acompanhada dos títulos das publicações associadas a cada um, de maneira clara e estruturada. Para isso, utiliza a função `Listar_Autores` de modo a obter os dados e exibi-los numa janela ("Lista de Autores na Base de Dados") organizada, facilitando a consulta pelo utilizador.

Consultar Todas as Publicações Associadas a Um Autor:

Função `Listar_Pub_Autor_Layout`

O utilizador é capaz de realizar uma pesquisa que tenha como resultado todas as publicações associadas a um autor específico, exibindo uma lista organizada de informação na janela "Publicações por Autor". O utilizador interage com a janela inserindo o nome do autor e, caso o autor pretendido possua publicações na base de dados, essas serão apresentadas de forma clara. Em caso contrário, o utilizador é informado da situação por meio de mensagens de erro.

As seguintes funções seguem a mesma estrutura da função mencionada anteriormente:

- `Listar_Pub_Afil_Layout`: Consulta por afiliação.
- `Listar_Pub_PC_Layout`: Consulta por palavras-chave.
- `Listar_Data_Pub_Layout`: Consulta por data de publicação.

Todas as funções referidas utilizam critérios de listagem diferentes, porém seguem o mesmo fluxo de exibição das publicações encontradas.

Gráficos Visuais Baseados em Análise de Dados:

É possível ao usuário gerar gráficos baseados em estatísticas específicas às publicações presentes na base de dados através da função `Gerar_Gráficos` desenvolvida pelo grupo.

A janela "Gráficos" exibe um conjunto de botões permitindo ao utilizador escolher o tipo de gráfico que deseja visualizar.

Os botões disponíveis correspondem às seguintes opções de gráficos:

- **Top 20 Autores**: Exibe os 20 autores mais frequentes na base de dados.
- **Top 20 Palavras-Chave**: Exibe as 20 palavras-chave mais comuns em publicações.
- **Palavra-Chave Mais Frequente Por Ano**: Exibe a palavra-chave mais frequente por cada ano disponível na base de dados.
- **Publicações Por Ano**: Apresenta o número de publicações por cada ano disponível na base de dados.
- **Publicações Por Mês Num Ano**: Exibe o número de publicações por mês num ano específico à escolha do usuário.

- **Publicações de um Autor Por Ano:** Exibe o número de publicações de um autor à escolha do usuário por ano.

A função irá executar um loop contínuo de modo a exibir a janela "Gráficos" até o utilizador decidir encerrá-la.

Para cada tipo de gráfico, `Gerar_Gráficos` chama uma função específica da interface da linha de comandos (CLI) importada como `logic` para obter os dados necessários. Em seguida, a função correspondente do ficheiro importado `gráficos_distribuições` como `gd` é chamada para gerar e exibir o gráfico.

- **TOP20A:** Chama a função `Distribuição_20A` da lógica para obter os 20 autores mais frequentes e passa essa distribuição para a função `graf_20A`.
- **TOP20PC:** Chama a função `Distribuição_20PC` da lógica para obter as 20 palavras-chave mais frequentes e passa essa distribuição para a função `graf_20PC`.
- **TOP1PC:** Chama a função `Distribuição_PC` da lógica para obter a palavra-chave mais frequente por ano e passa essa distribuição para a função `graf_PC`.
- **PUB_ANO:** Chama a função `Distribuição_Ano` da lógica para obter o número de publicações por ano e passa essa distribuição para a função `graf_Ano`.
- **PUB_MES:** Chama a função `Distribuição_Mês` da lógica para obter o número de publicações por mês num ano específico e passa essa distribuição para a função `graf_Mês`.
- **PUB_AUTOR:** Chama a função `Distribuição_Autor` da lógica para obter o número de publicações por ano de um autor específico e passa essa distribuição para a função `graf_Autor`.

Todas as funções do tipo `graf` foram concebidas utilizando o módulo `pyplot` da biblioteca `matplotlib` para criar as representações visuais.

Relatórios Baseados em Análise de Dados:

É possível ao usuário gerar relatórios baseados em estatísticas específicas às publicações presentes na base de dados através da função `Gerar_Relatórios` desenvolvida pelo grupo.

A janela "Relatórios" é composta por vários botões que permitem ao utilizador seleccionar o tipo de relatório que deseja criar. Os relatórios disponíveis correspondem aos dados que são apresentados na função `Gerar_Gráficos`. Os botões disponíveis na janela correspondem aos mesmos botões disponíveis para a função anteriormente descrita (**Top 20 Autores**, **Top 20 Palavras-Chave**, **Palavra-Chave Mais Frequente Por Ano**, **Publicações Por Ano**, **Publicações Por Mês Num Ano**, **Publicações de um Autor Por Ano**).

Aquando da escolha de um relatório, os seus dados são inseridos numa área de texto multiline exibida na janela "Relatórios" onde os resultados são listados de forma clara e organizada.

Main:

A função `main()` é o ponto central da aplicação, controlando o fluxo de interação com o utilizador. A variável `stop` controla o loop principal, permitindo que ele continue até o utilizador fechar a aplicação. A janela principal é criada usando o `PySimpleGUI`, com o título "Consulta e Análise de Publicações Científicas", e o layout é organizado pela função `criar_layout(publicações)`, que define os botões e campos da interface.

O programa entra num loop de eventos, aguardando interações do usuário, como cliques nos botões ou o encerrar da janela. Quando o evento `sg.WINDOW_CLOSED` ocorre, o loop é encerrado. Diversos eventos são definidos, como a criação de publicações, onde os campos para inserção dos dados são exibidos ao acionar o botão `–CRIAR–`. Quando o evento `–SALVAR_PUBLICACAO–` é acionado, os dados da publicação são salvos na base de dados e os campos são ocultados. Se o utilizador optar por cancelar a criação, o painel original é restaurado ao acionar o evento `–CANCELAR_PUBLICACAO–`.

Conclusão

Para conseguirmos desenvolver o sistema de "Consulta e Análise de Publicações Científicas", foi necessária a utilização de todos os conhecimentos prévios em programação, bem como a utilização de diversas bibliotecas, como o PySimpleGUI para a interface gráfica e o gerenciamento de eventos. Durante o desenvolvimento, tivemos de adaptar e aplicar funcionalidades específicas, como a criação, atualização, listagem e consulta de publicações científicas. Além disso, a implementação de gráficos e relatórios, a partir da análise dos dados, exigiu uma compreensão aprofundada de como lidar com bases de dados e gerar representações visuais e relatórios úteis para o utilizador.

Apesar de existirem diversos pontos nos quais ainda é possível melhorar e aprender mais, com o desenvolvimento deste projeto, conseguimos consolidar e completar os nossos conhecimentos de programação, adquiridos ao longo de todo o semestre. Além disso, tivemos a oportunidade de desenvolver uma aplicação que pode ser muito útil na área da pesquisa científica, permitindo que os utilizadores consultem e analisem publicações de forma prática e eficiente. Este projeto não só representa um avanço nas nossas competências de programação, mas também cria uma ferramenta que pode ser utilizada por profissionais da área para gestão e análise de publicações científicas.