

Escola Superior de Saúde do Instituto Politécnico do Porto  
1.º ano de Mestrado em Bioestatística e Bioinformática Aplicadas à Saúde  
Ano letivo: 2023/2024  
UC: Programação e Base de Dados  
Docentes: Vítor Sá e Paulo Gomes

## **GESStor de Referências – Comentários**

Dificuldades sentidas na implementação do sistema de Gestão de Referências  
Bibliográficas na linguagem Python

Trabalho realizado no âmbito da unidade curricular Programação e Base de Dados lecionada  
pelos docentes Vítor Sá e Paulo Gomes.

Ana Sofia Lima, 10200763

Carolina Silva Coutinho, 10230053

Porto, 10 de novembro de 2023

## Índice

Curva de Aprendizagem do Python.....	1
Introdução .....	1
Módulo <i>tkinter</i> – vantagens e limitações .....	2
Conclusão .....	4

## Introdução

O objetivo na criação desta ferramenta foi implementar na linguagem Python um programa que simulasse a gestão de referências bibliográficas de forma intuitiva, incluindo a inserção das referências no programa, a sua edição posterior e a exploração dos seus conteúdos. No presente texto será descrita a experiência com este projeto, incluindo as dificuldades, soluções e preocupações que nele constaram.

## Curva de Aprendizagem do Python

A principal dificuldade encontrada, como foi evidente pelo número de vezes que foi necessário corrigir este erro, foi iterar sobre as diferentes listas e dicionários das referências. Conforme a variável em questão foi necessário adaptar o código constantemente de forma que fosse possível chamar o valor necessário para a tarefa. Isto aconteceu principalmente quando a função envolvia listar autores, visto que o nome e apelido de cada autor estava inserido dentro de um dicionário, dentro de uma lista, dentro de outro dicionário, dentro da lista das referências. Desde logo a variável dos autores foi a mais desafiante, visto que exigiu a divisão do nome dos autores em nome próprio e apelido num dicionário, pelo que se destaca das restantes chaves em cada referência. Este problema foi ultrapassado através do estudo do programa, ferramentas *on-line* (como o Python Visualizer) e com a prática adquirida ao longo do tempo. A função de listar as publicações por uma ou mais palavras-chave também foi um desafio pois foi a única em que o *input* consistia em mais do que uma palavra, pelo que teve de ser transformado numa lista, e a comparação de listas (no caso, com a lista de palavras-chave presentes nas referências) não é facilmente implementada. Este desafio foi contornado através de um *loop* e uma verificação de condição numa única linha.

Outra dificuldade encontrada foi a legibilidade do código que por vezes se tornava demasiado complexo. A utilização de comentários utilizando o cardinal e a divisão do código em diferentes células veio melhorar a sua legibilidade.

Apesar de inicialmente o raciocínio ser um pouco desafiante devido à falta de experiência com o programa, com o passar das horas dedicadas à programação tornou-se mais natural e claro, visto que se evidenciou a forma como utilizar e programar no Python corretamente. Assim, a tarefa mais simples foi a implementação das funções de estatística visto que o raciocínio foi menos exigente, tendo em conta que foi a última parte a ser criada, pelo que foi possível reutilizar conteúdo de funções anteriores.

Em geral, uma preocupação com o programa era que este fosse o mais *user-friendly* possível e sentimos que isto foi atingido através da função de leitura dos ficheiros RIS (Research Information Systems), da utilização da biblioteca *unidecode*, que tem em conta as várias formas diferentes que o utilizador pode inserir o *input*, da formatação dos *outputs* e através principalmente da aplicação desenvolvida, que é bastante intuitiva.

### **Módulo *tkinter* – vantagens e limitações**

A crescente complexidade do código ao longo do trabalho, por sua vez, obrigou à procura de uma solução que tornasse a navegação do programa fácil e intuitiva. A ideia inicial passou pela implementação de menus e sub-menus que saíam automaticamente na consola aquando da inicialização da aplicação, de forma a guiar o utilizador de forma mais eficaz. No entanto, achamos que esta solução ficou aquém das expectativas, levando à exploração do módulo *tkinter*, de forma a criar um GUI (*graphical user interface*), que permitiu dinamizar o código e transformá-lo em algo legível e esteticamente agradável. A utilização deste módulo auxiliou também na resolução de alguns problemas e dificuldades, tais como:

- Permitiu limitar o *input* dos utilizadores, como no caso da variável “Tipo de publicação”, ao substituir o *input* de uma *string* por uma *combobox* (menu suspenso) onde as únicas opções disponíveis são aquelas definidas pelo programador. Isto eliminou a necessidade de código extra para verificar se o *input* dado era válido no contexto da aplicação ou não;
- Eliminou a necessidade do utilizador de copiar os caminhos dos ficheiros RIS que pretendesse inserir na biblioteca, ao abrir uma janela que lhe permite seleccionar o ficheiro diretamente de uma pasta;

- O utilizador deixou de ter de reinicializar a aplicação ou inserir *inputs* extra caso pretendesse recuar para os menus prévios ou realizar outras operações – é possível correr diferentes partes do código em simultâneo graças à implementação de janelas *pop-up*;
- De uma forma geral, foi possível minimizar a quantidade de erros de *parsing* resultantes da necessidade de obter um grande volume de *inputs* por parte do utilizador, como por exemplo:
  - No código original, era necessário fornecer ao utilizador instruções para o *input* dos autores e palavras-chave, devido ao facto destas variáveis poderem tomar mais do que um valor (por exemplo, pedir que os autores/palavras-chave fossem separados por vírgulas e seguissem uma determinada ordem), e caso essas instruções não fossem seguidas, o *input* não seria corretamente tratado e poderia até levar a erros fatais no programa. O módulo utilizado permite a inserção de cada parâmetro separadamente (como no caso dos autores, o nome próprio e o apelido são inseridos em caixas de texto diferentes devidamente identificadas), o que facilita o correto tratamento dos *inputs*.
  - Antes da implementação do GUI, era necessário escrever código extra para casos onde o utilizador tinha de seleccionar um artigo, autor, palavra-chave, etc. em específico para eliminar/alterar, uma vez que a inserção de um valor para o *index* fora do intervalo levaria a um erro do tipo "*Index out of range*". A aplicação contorna esta limitação ao deixar o utilizador apenas seleccionar um parâmetro que esteja na lista fornecida pelo próprio programa.

Posto isto, existiram dificuldades associadas à utilização deste módulo também. A mais óbvia e proeminente foi a necessidade de perceber um módulo novo do zero e aprender a implementar as suas funcionalidades. Outra dificuldade foi conseguir adaptar o código original (completado anteriormente à implementação do GUI) ao *tkinter*, tendo sido necessário reescrever certas secções de novo. Finalmente, o módulo em si apresenta certas limitações; quando uma referência é inserida manualmente, é necessário por vezes inserir mais do que um autor ou palavra-chave. Esta funcionalidade é possível e foi implementada, no entanto não foi

possível fazer com que as caixas de entrada extra aparecessem diretamente acima do botão e junto às outras do mesmo tipo. Ao invés, estas aparecem no fundo da página, o que pode ser confuso e contraintuitivo para o utilizador.

Por outro lado, existem limitações no programa que são impossíveis de contornar; um ficheiro RIS mal formatado pela própria instituição de origem vai gerar uma referência potencialmente incorreta, uma vez que o código apenas está preparado para ler ficheiros codificados de acordo com as normas oficiais. Uma solução para este problema seria permitir a posterior edição de todos os parâmetros (ao invés de só autores ou palavras-chave, como era pedido no enunciado), tal como acontece noutros gestor de referências que aceite ficheiros RIS (e.x.: Mendeley).

## **Conclusão**

Em suma, e tendo aprendido a ultrapassar vários desafios, sentimos que o programa final é bastante claro não só para o utilizador como para o olhar de programador, devido ao esforço colocado sobre a sua organização em funções, nomeação das variáveis, entre outros. Estes desafios agregaram-nos experiência e conhecimento novo e renovado acerca da linguagem de programação Python e do módulo *tkinter*, pelo que é com grande satisfação e realização que subtemos a aplicação GESStor de Referências.