



Escola de Engenharia
Universidade do Minho

Trabalho Prático 2

Processamento de Linguagem Natural em Engenharia Biomédica

Mestrado em Engenharia Biomédica - Informática Médica

2022/2023

Docentes:

Luís Filipe da Costa Cunha

José João Almeida

Diogo Guedes Lameira, PG50332

João Filipe Costa Alves, PG50471

José Miguel Moreira Santos, PG51190

Índice

Introdução	3
Arquitetura do projeto.....	4
Websites processados	5
Saúde de A-Z – CUF	5
Diseases & Conditions – Mayo Clinic	7
Enfermedades y Afecciones – Middlesex Health	8
Junção final	9
Interface	10
Conclusão.....	15
Bibliografia.....	15

Introdução

O objetivo do trabalho prático passa por dar continuidade aos resultados obtidos no trabalho anterior. Neste foi desenvolvida uma estrutura de dados *JSON* com termos médicos e respectivas traduções em inglês e espanhol, assim como a descrição do termo em português. Neste segundo trabalho pretende-se, então, enriquecer o conjunto de dados através da adição de informações provenientes de fontes externas, nomeadamente websites (*web scraping*). Para além disso pretendia-se encontrar uma relação de interesse entre os termos e que permitisse agrupá-los em categorias relevantes. Após obter um conjunto de dados satisfatório foi, então, desenvolvida uma ferramenta capaz de representar as informações de forma adequada e manipular os dados, nomeadamente um website desenvolvido através da ferramenta *Flask*.

Assim, foi desenvolvido um website que consiste num glossário de doenças. O objetivo do projeto passou por criar uma plataforma que disponibilizasse um extenso conjunto de informações sobre doenças em português, inglês e espanhol, agrupadas por categorias e permitisse que fossem adicionados e removidos termos, assim como efetuar pesquisas sobre os mesmos.

Este website poderá constituir uma ferramenta valiosa para profissionais da área médica, estudantes, ou mesmo qualquer pessoa interessada em aprender mais sobre algumas das mais comuns doenças. O glossário de doenças fornece, portanto, uma forma fácil de aceder a informações relevantes em vários idiomas, tornando o conhecimento médico mais acessível e compreensível para todos.

Arquitetura do projeto

Podem ser apontados alguns componentes importantes para o desenvolvimento do projeto: web scraping, junção dos dados, armazenamento dos dados e desenvolvimento de website Flask. O primeiro componente é responsável por extrair dados de fontes externas, neste caso websites, utilizando a ferramenta BeautifulSoup. Após a extração dos dados dos sites, os dados anteriores são enriquecidos com as novas informações extraídas. Após o enriquecimento dos dados, estes são armazenados numa estrutura adequada, neste caso um ficheiro json, para serem, posteriormente, utilizados na criação do site Flask. Este é responsável por exibir os dados de forma acessível e permitir a sua manipulação.

Pretende-se que o ficheiro json final tenha a seguinte estrutura:

```
Categoria (PT): {  
  Doença (PT): {  
    "PT": {  
      Termo (PT):  
      Info (PT):  
    }  
    "EN" {  
      Term (EN):  
      Info (EN):  
    }  
    "ES" {  
      Plazo (ES):  
      Info (ES):  
    }  
  }  
}
```

Websites processados

Saúde de A-Z – CUF [1]

Pretende-se fazer o *web scraping* de um glossário de doenças em português com vista à criação de um ficheiro *JSON*. De forma detalhada e passo a passo, foi implementado da seguinte forma:

Importação de bibliotecas:

As bibliotecas *requests*, *json*, *re* e *BeautifulSoup* foram importadas para permitir o acesso à web, manipulação de JSON, manipulação de expressões regulares e análise do HTML, respetivamente.

Definição da função *extractDiseasePage(url)*:

Esta função foi criada de forma a receber uma URL como entrada e realizar a extração das informações sobre uma doença específica na página correspondente. De forma resumida esta função executa os seguintes passos:

- Requisição HTTP para a URL fornecida e obtenção do conteúdo HTML da página.
- Utilização da biblioteca *BeautifulSoup* para analisar o HTML.
- Procura pelos títulos das descrições da doença no HTML, utilizando diferentes seletores CSS, caso exista, caso contrário extrai a descrição geral da doença a partir de uma classe CSS.
- Extração do conteúdo de cada descrição entre os seletores e remoção das quebras de linha.
- Retorno do dicionário com os títulos das descrições como chaves e o conteúdo das descrições como valores.

Definição da função *extractDiseaseListPage(div)*:

Essa função recebe um elemento *div* como entrada e extrai o título da doença listada nessa *div*. A função retorna o título da doença.

Após a implementação destas duas funções auxiliares, realizou-se os seguintes passos.

Definição de URLs e cabeçalhos:

Após isto, foram definidos os três URLs: *url* corresponde à página inicial do site "<https://www.cuf.pt/saude-a-z>", *url1* é usado para construir as URLs completas das páginas de doenças e *url2* é usado para construir as URLs das diferentes categorias de doenças. Também é definido um

cabeçalho para simular um navegador web ao fazer as requisições. Caso contrário o acesso a estes *urls* não era possível.

Obtenção do HTML da página inicial:

A biblioteca *requests* é usada para enviar uma solicitação HTTP à página inicial especificada em *url* e é obtido o conteúdo HTML da página.

Criação do objeto *BeautifulSoup*:

O conteúdo HTML obtido na etapa anterior é passado para a biblioteca *BeautifulSoup*, que cria um objeto *soup* para facilitar a análise e extração de dados.

Recolha das opções de categoria:

Através do objeto *soup*, a função *find_all* é usada para encontrar todas as *tags* *<option>*. Essas *tags* correspondem às diferentes categorias de doenças disponíveis no site.

Criação de um dicionário de categorias:

É criado um dicionário *category_dict* vazio para armazenar as informações extraídas sobre as doenças, organizadas por categoria.

Iteração sobre as opções de categoria:

Para cada opção de categoria encontrada na etapa anterior, são realizados os seguintes passos:

- Criação de uma *URL* para a categoria atual, adicionando o valor da opção à URL base *url2*.
- Obter o número de página e inicializando-o com 0.
- *Loop* enquanto for necessário para percorrer as páginas de doenças associadas à categoria.
- Construção da URL completa para a página atual, incluindo o número da página.
- Solicitação HTTP para a URL da página atual para obter conteúdo HTML.
- Novo objeto *BeautifulSoup* com o conteúdo HTML da página atual.
- Extração de todas as *divs* com a classe *views-row*, que contêm as informações das doenças.
- Iteração sobre as *divs* encontradas e após isto:
- Obtenção da URL da página de cada doença a partir do elemento HTML.
- Chamada da função *extractDiseasePage* para extrair as informações sobre a doença da página correspondente.
- Chamada da função *extractDiseaseListPage* para obter o título da doença listada.

- Modificação do título removendo texto entre parênteses e convertendo-o para minúsculas.
- Adicionar as informações da doença ao dicionário *category_dict* na categoria correspondente.
- Verificar se há uma próxima página utilizando o elemento *li* com a classe *pager__item pager__item-next*.
- Se houver uma próxima página, incrementa o número da página e continua para a próxima iteração do *loop*.
- Se não houver uma próxima página, sai do loop.

Por fim, guarda o *category_dict* no respetivo arquivo *JSON* com os resultados organizados por categorias e doenças.

Diseases & Conditions – Mayo Clinic [2]

Pretende-se fazer *web scraping* de um glossário de doenças em inglês. Para tal, recorreu-se a bibliotecas como *requests* e *BeautifulSoup* para realizar as operações necessárias de procura das páginas HTML, e as bibliotecas *re* e *json* para manipular os resultados e para guardar os resultados em ficheiros JSON e, respetivamente.

Primeiramente são declaradas algumas variáveis (URLs) importantes para o processo de *web scraping* e é efetuado um *request* HTTP à página principal de doenças da Mayo Clinic, obtendo-se o HTML da página. Através da biblioteca *BeautifulSoup* será possível extrair informações relevantes. Inicialmente, o código percorre todas as *anchor tags* que redirecionam para uma lista de doenças selecionadas pela letra inicial, guardando numa lista todos os novos *urls* recolhidos.

Seguidamente, a lista de *urls* é percorrida e são feitos *requests* HTTP às páginas contendo a lista de doenças ordenadas alfabeticamente. Realizado o varrimento é obtido um dicionário contendo nas chaves os nomes das doenças e nos valores respetivos os *urls* que redirecionam para a página respetiva de cada doença.

Posteriormente, o dicionário é percorrido e os *requests* HTTP às páginas contendo as informações de cada doença realizados. Novamente, o HTML das páginas é analisado com *BeautifulSoup*, e as informações relevantes, como os títulos e as respetivas informações relativas a cada doença extraídos e armazenados num dicionário denominado *disease_info*.

A extração dos títulos e respetivos parágrafos apresenta um processo complexo de definição do título através de uma *tag h2* e posterior soma de todos os parágrafos encontrados antes do título seguinte, tendo em conta que pelo caminho eram encontradas algumas *tags* irrelevantes para o processo.

De seguida, o dicionário final, *disease_all_info*, é composto contendo nas chaves o nome de cada doença e nos valores o respetivo dicionário *disease_info*.

Por fim, este dicionário é guardado num ficheiro JSON chamado "*en_diseases.json*". Tal permite que as informações sejam armazenadas de forma estruturada para posterior manipulação, em combinação com outros ficheiros.

Enfermedades y Afecciones – Middlesex Health [3]

Pretende-se fazer o *web scraping* de um glossário de doenças em espanhol. As bibliotecas necessárias para tal incluem a biblioteca *requests* para fazer *requests* HTTP, *json* para trabalhar com dados no formato JSON, *re* para manipulação de expressões regulares e *BeautifulSoup* para analisar o HTML. Assim, são definidos os URLs que serão necessários para o *web scraping*, nomeadamente *url_es* que é o URL principal que contém uma lista de doenças e *url_es_1* que é a parte inicial do URL que será concatenado com os links encontrados na página principal.

Ao analisar a estrutura da página principal verificou-se que os links para as páginas individuais para cada doença se encontravam dentro do elemento `<div>` com a classe "*service-content*", no atributo *href* de *tags <a>*. Assim, a lista com os URLs das páginas individuais é percorrida e um novo *request* HTTP é feito para cada uma delas e o HTML é analisado com a biblioteca *BeautifulSoup*. De seguida foram encontrados os elementos HTML que contêm informações relevantes sobre cada doença, como o nome, descrição, sintomas, causas, diagnóstico, tratamento e prevenção. Recorreu-se, ainda, a uma função *content()* que é utilizada para extrair o conteúdo de cada secção. Ao verificar que as informações relevantes que se pretendia extrair se encontravam delimitadas por *tags <h2>* que contêm o título da secção, a função percorre os elementos HTML e extrai o texto de cada elemento até encontrar um novo *heading <h2>*, construindo assim o conteúdo completo entre os mesmos. É ainda utilizada a função *get_text()* para remover as *tags* do conteúdo extraído e utilizada uma função *re.sub* para remover os caracteres `\n`. As informações extraídas para cada doença são armazenadas em um dicionário chamado

dici, em que cada termo da doença é usado como *key*, e as informações relacionadas são armazenadas num subdicionário com as *keys* correspondentes ao nome das secções: "Descrição", "Sintomas", "Causas", "Diagnóstico", "Tratamento" e "Prevención" e os dados armazenados são exportados para um arquivo no formato JSON.

Junção final

Por fim, e tendo em conta a estrutura delineada para o ficheiro *JSON* final foi desenvolvido um ficheiro *Python* a envolver os dados dos três ficheiros *JSON* anteriores (*pt_diseases.json*, *es_diseases.json* e *en_diseases.json*) e do ficheiro *JSON* final do último trabalho prático.

Desta forma, cria-se um novo dicionário chamado *merged_data* para armazenar os dados combinados dos três ficheiros. Os passos efetuados foram os seguintes:

- Percorreu-se os dados das doenças em português (*pt_diseases.json*) por categoria;
- Para cada doença, criou-se uma entrada correspondente no dicionário *merged_data* a conter informações em português da doença bem como entradas para informações das doenças em inglês e espanhol, caso existam.
- Verificação se a chave em português (doença) possui tradução em inglês e espanhol derivado do ficheiro *JSON* final do último trabalho prático. Caso não, utiliza o *Google Translate*.
- Se houver tradução em inglês, adicionar a tradução e procurar a mesma no *es_diseases.json*. Caso exista, adiciona outras informações disponíveis nesse dicionário ao dicionário *merged_data* no respetivo local.
- O mesmo se aplica à língua espanhola.

Por fim, guarda o *merged_data* no respetivo arquivo *JSON* com as informações combinadas das três fontes, incluindo termos em português, inglês e espanhol, bem como outras informações relacionadas às doenças.

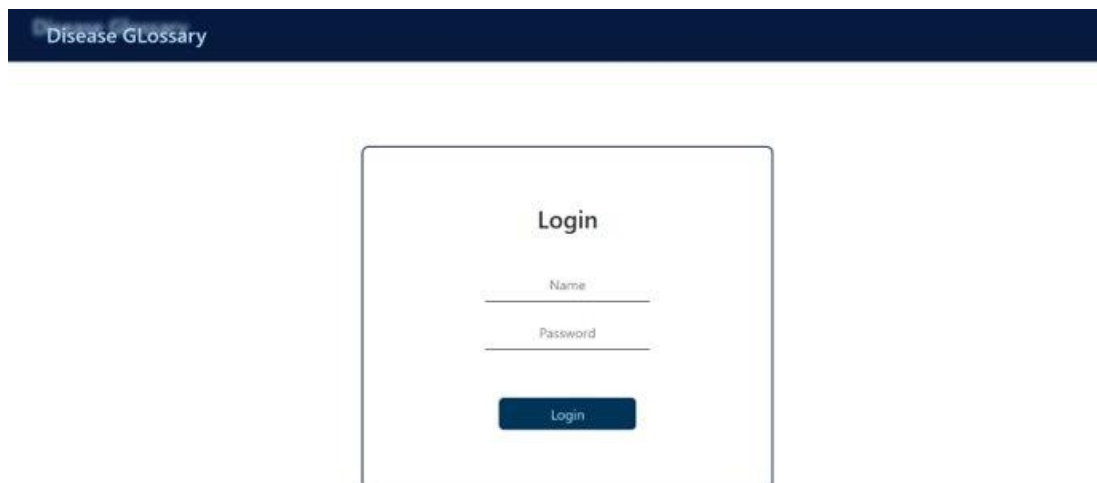
Para a realização da etapa da verificação da tradução da chave em português foram implementadas três funções auxiliares. A função *translate_to_english* traduz um texto de português para inglês através do *Google Translate*. A função *translate_to_spanish* traduz um texto de português para espanhol através do *Google Translate*. E por fim, a função *check_translation* que verifica se uma

chave em português existe nos dados de tradução do ficheiro *JSON* do trabalho prático anterior. Se existir, obtém as traduções em inglês e espanhol a partir desse dicionário. Caso contrário, utiliza as funções de tradução do *Google Translate* mencionadas acima para obter as traduções.

Interface

Inicialmente, os objetivos definidos para a formalização da interface foram baseados numa estrutura simples, intuitiva, dinâmica, logo consequentemente cativante, e fundamentalmente informática, ou seja, direta e sem imensas distrações. Desta forma, foram definidas as cores branco para o fundo, de modo a dar relevância às informações, e azul de modo a indicar os locais onde o utilizador teria de efetuar ações. Os blocos de informação e *inputs* seriam realçados em espaços limitados por uma borda.

A utilização de ações como eliminar e adicionar termos faziam apenas sentido para utilizadores competentes e definidos para o caso, dessa forma, foram criados dois tipos de utilizadores diferentes e no mesmo sentido uma página de *login* que é apresentada na figura seguinte:



The image shows a web interface for a 'Disease Glossary'. At the top, there is a dark blue header bar with the text 'Disease Glossary' in white. Below the header, there is a white rectangular box with a thin blue border. Inside this box, the word 'Login' is centered at the top. Below 'Login', there are two input fields: the first is labeled 'Name' and the second is labeled 'Password'. Both labels are in a small, light blue font. Below the input fields, there is a dark blue button with the word 'Login' in white text.

Figura 1 - Template da página de Login.

Após a autenticação bem-sucedida o utilizador é redirecionado para a página de *welcome*, onde serão apresentados os acessos principais para os mais variados espaços e ainda uma indicação do tipo de utilizador.

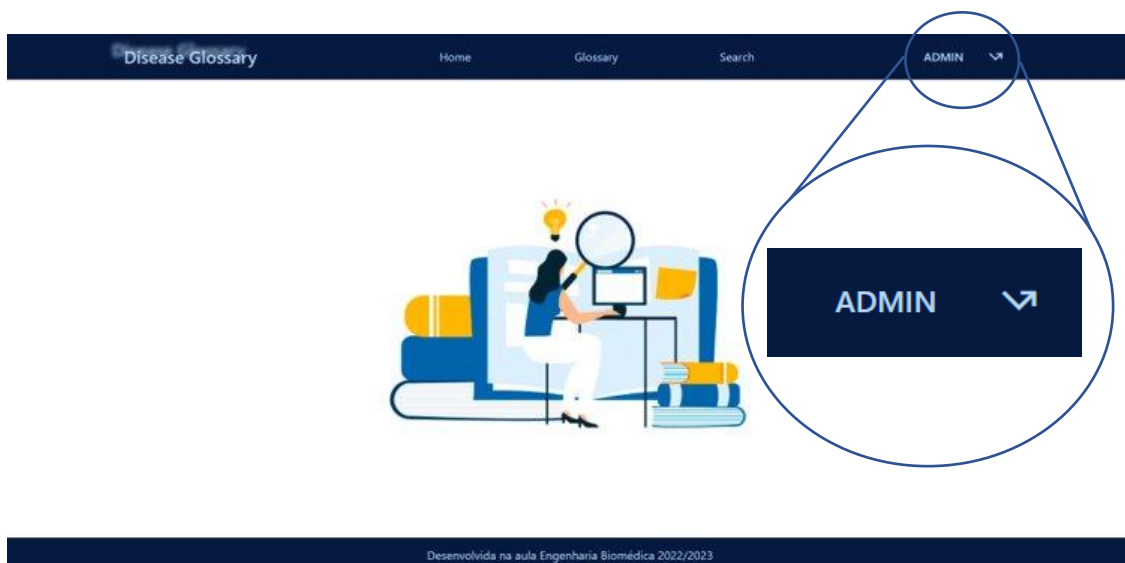


Figura 2 - Template da página Welcome com definição do tipo de utilizador autenticado.

As permissões de ações relativas a cada tipo de utilizador serão destacadas na apresentação ou omissão de algumas componentes, como é o caso dos botões de adicionar ou de eliminar termo, como é possível ver na página de apresentação do glossário realçados dentro de molduras vermelhas.

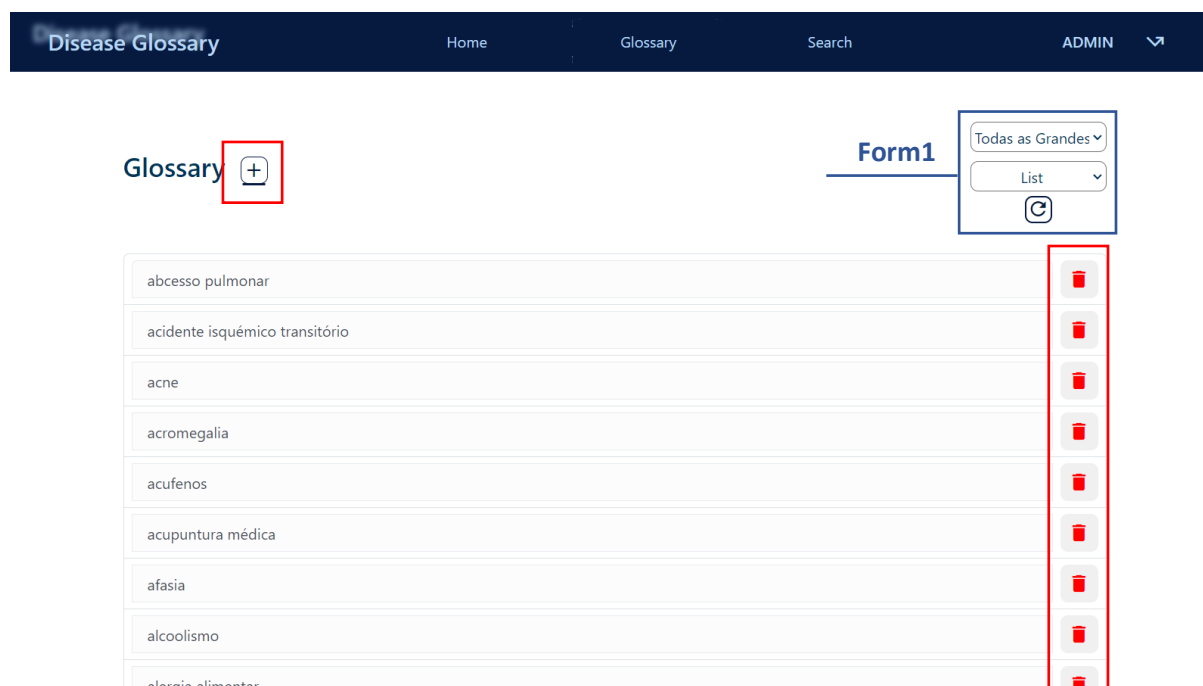


Figura 3 - Template da página do Glossary com apresentação das funcionalidades dos administradores.

Indicado na figura anterior, destacado por *Form1*, é apresentado um pequeno *form* que permite a filtragem do conteúdo apresentado:

- Primeiro input – permite a seleção da categoria sobre a qual um leque específico de doenças deve ser apresentado;
- Segundo input – permite a seleção do tipo de formatação dos dados, ora em forma de lista, ora em forma de tabela.

The screenshot shows the 'Disease Glossary' application interface. At the top, there is a dark blue header with the title 'Disease Glossary' and navigation links for 'Home', 'Glossary', 'Search', and 'ADMIN'. Below the header, the main content area has a title 'Todas as Grandes Áreas' with a plus icon. To the right of the title are two dropdown menus: 'Todas as Grandes' and 'List'. Below these is a search bar labeled 'Search:'. The main table displays a list of diseases with three columns: 'Term', 'Translation EN', and 'Translation ES'. The table is paginated, showing 'Showing 1 to 10 of 544 entries' and a 'Previous' button. The current page is '1' out of 55 pages.

Term	Translation EN	Translation ES
abscesso pulmonar	lung abscess	absceso pulmonar
acidente isquêmico transitório	transient ischemic attack	ataque isquémico transitorio
acne	acne	acné
acromegalia	acromegaly	acromegalia
acufenos	tinnitus	tinnitus
acupuntura médica	medical acupuncture	acupuntura medica
afasia	aphasia	afasia
alcoolismo	alcoholism	alcoholismo
alergia alimentar	food allergy	alergia a la comida
alergias	allergies	alergias

Figura 4 - Apresentação das doenças em formato de tabela.

A mudança dos resultados para uma categoria específica procede à mudança do título “Glossary” para o nome da categoria selecionada pelo utilizador.

A *tab Search* permite a pesquisa de termo(s) seja na lista de doenças, seja em todas as informações correspondentes a cada uma, sendo esta pesquisa feita por palavras completas e não por partes constituintes da mesma, ou seja, a pesquisa de “ato” não retorna resultados como “patologia”, que contem a pesquisa, mas sim resultados que contêm exatamente a palavra “ato”. Os resultados são apresentados numa lista de *links* que redirecionam para a página relativa a cada doença. Caso não sejam encontrados resultados é apresentada uma mensagem ao utilizador.

Figura 5 - Template da página de Search.

A página relativa a cada doença apresenta, inicialmente em português, todos os aspetos relativos à mesma (sintomas, tratamento, etc.). Contem ainda dois botões que permitem o acesso às páginas sobre a mesma doença em inglês e espanhol, e apresentação dos tópicos mencionados anteriormente caso estes campos existam.

Por fim, a página de adicionar um novo termo apresenta um *form*, com campos de preenchimento obrigatório, estes compreendem:

- Um *select*, que exclui a apresentação da categoria “Todas as Grandes Áreas”;
- Um espaço para preenchimento do nome da doença;
- Dois *inputs* relativos à tradução do nome da doença para inglês e espanhol;
- Uma *textarea* para preenchimento da descrição da doença em múltiplas linhas.

A adição de uma nova doença a uma categoria específica automaticamente procede à adição da nova doença à categoria que engloba todas as doenças. O *template* da página de adição é o apresentado a seguir.

Add Term

+65

▼

Disease

English Translation

Spanish Translation

Portuguese Description

↗

Add

Figura 6 - Template da página Add Term.

Conclusão

Neste projeto foi possível atingir o objetivo de enriquecer o conjunto de dados do trabalho prático passado e desenvolver um glossário de doenças juntamente com uma ferramenta funcional para representar e manipular essas informações. A adição de informações provenientes de fontes externas por meio do *web scraping* contribuiu para expandir o conteúdo disponível, tornando-o mais abrangente e útil para quem o consulta. A ferramenta *Flask* permitiu criar uma plataforma acessível onde os usuários podem explorar as informações sobre doenças em diferentes idiomas. Foi ainda possível desenvolver um sistema de autenticação que permite a administradores apagar, editar ou adicionar novas doenças, enquanto utilizadores comuns sem essa permissão apenas podem consultar as informações e efetuar pesquisas sobre as mesmas. A procura e navegação pelos conteúdos é bastante facilitada pela categorização dos termos, que fornece uma estrutura organizada às informações.

No entanto, poderiam ter sido implementadas algumas melhorias, desde a continuação do enriquecimento do conjunto de dados com mais termos, traduções e descrições, para tornar o glossário ainda mais completo, ou ainda aprimorar as funcionalidades do *website*, como por exemplo a inclusão de recursos multimídia como imagens, vídeos ou gráficos relacionados a doenças específicas.

Bibliografia

- [1] Saúde de A-Z. CUF. Available at: <https://www.cuf.pt/saude-a-z> (Accessed: 30 May 2023).
- [2] Medical Diseases and Conditions. Mayo Clinic. Available at: <https://www.mayoclinic.org/diseases-conditions> (Accessed: 31 May 2023).
- [3] Enfermedades y Afecciones. Middlesex Health. Available at: <https://middlesexhealth.org/learning-center/espanol/enfermedades-y-afecciones> (Accessed: 30 May 2023).