

UNIVERSIDADE DO MINHO

Trabalho Prático 2

Elaborado no âmbito da Unidade Curricular de
Processamento de Linguagem Natural em Engenharia Biomédica

Grupo

André Sousa (PG52564)
Beatriz Macedo (PG55820)
Maria Vale (PG56144)

Docentes

José João Antunes Guimarães Dias Almeida
Luís Filipe Costa Cunha

junho de 2025

Índice

1	Introdução	4
2	Melhorias de Extração TP1	4
3	Construção do JSON Final	4
3.1	processar_multilingue_final.py	4
3.2	processar_glossarioNeo.py	5
3.3	glossario_mini_processor.py	5
3.4	Extração e Atualização de Definições em Inglês (definicoes_ingles.py)	6
3.5	Extração de Siglas e Abreviaturas (siglas_abrev.py)	6
4	Web scraping	7
4.1	Fonte 1: HGIS - Abreviações e Siglas	7
4.1.1	Leitura do ficheiro	8
4.1.2	Extração de conteúdo das tabelas	9
4.1.3	Filtragem e normalização	10
4.1.4	Classificação e inserção	11
4.1.5	Atualização do <i>dataset</i>	11
4.2	Fonte 2: Harvard Health Publishing	12
4.2.1	Leitura do ficheiro	12
4.2.2	Navegação pelas páginas do <i>website</i> :	12
4.2.3	Extração dos termos e definições	13
4.2.4	Associação com conceitos existentes	14
4.2.5	Atualização do ficheiro <i>JSON</i>	15
5	Backend	15
5.1	Funções de Apoio	15
5.1.1	Funções de Processamento do JSON	15
5.1.2	Função <i>processar_links()</i>	16
5.2	Funções de Pesquisa	17
5.3	Rotas Principais	17
5.4	Rotas de detalhe	18
5.5	Funções de Operações CRUD	19
6	Funcionalidades da aplicação web	19
6.1	Home	20
6.2	Conceitos	22
6.2.1	Listagem dos Conceitos	22
6.2.2	Pesquisa	22
6.2.3	Visualização Detalhada do Conceito	22
6.2.4	Ligação de Termos	23
6.2.5	Artigos Relacionados	23
6.2.6	Funcionalidades Adicionar, Editar, Eliminar	24
6.3	Siglas	25

6.3.1	Listagem	25
6.3.2	Pesquisa	25
6.3.3	Visualização da sigla	26
6.3.4	Funcionalidades Adicionar, Editar, Eliminar	26
6.4	Abreviaturas	26
6.4.1	Listagem	26
6.4.2	Pesquisa	27
6.4.3	Visualização da abreviatura	27
6.4.4	Funcionalidades Adicionar, Editar, Eliminar	27
6.5	Categorias	28
6.5.1	Listagem das Categorias	28
6.5.2	Pesquisa de Categorias	28
6.5.3	Visualização Detalhada de Categoria	29
6.5.4	Funcionalidades Adicionar, Editar, Eliminar	29
7	Secção Anexos	30
8	Conclusão	32

1 Introdução

Este relatório apresenta o desenvolvimento do Trabalho Prático 2 da unidade curricular de Processamento de Linguagem Natural em Engenharia Biomédica. O principal objetivo foi enriquecer o dataset previamente criado no TP1, que consistia na extração de conceitos médicos a partir de documentos PDF, complementando-o com novas informações obtidas através de fontes externas. Para isso, foram utilizadas técnicas de *web scraping* sobre websites médicos e glossários online, com vista à adição de definições, traduções em inglês, siglas, abreviações e outras propriedades relevantes.

Além do enriquecimento do dataset, foi também desenvolvida uma ferramenta de visualização e manipulação dos dados em formato JSON. Esta aplicação permite explorar relações entre conceitos, organizá-los por categorias e editar os conteúdos de forma dinâmica e persistente. O relatório documenta o processo de construção do sistema, detalha o contributo de cada ficheiro de código na estruturação do JSON final e apresenta registos da web app funcional.

2 Melhorias de Extração TP1

No âmbito do aperfeiçoamento da extração de conceitos do *Diccionari Multilingüe de la COVID-19*, foram implementadas melhorias significativas na fiabilidade e completude dos dados extraídos. Destaca-se, em particular, a reformulação do processo de identificação e separação das notas explicativas. Enquanto o código original apenas reconhecia blocos simples de texto com a formatação associada a notas, a nova abordagem introduziu uma função dedicada que procura de forma robusta o marcador “Nota:”, delimita automaticamente o seu fim com base no início do conceito seguinte e é capaz de **separar corretamente múltiplas notas numeradas** (por exemplo, “nota 1”, “nota 2”). Estas notas são agora extraídas em **formato corrido**, aumentando a legibilidade e a utilidade semântica da informação.

Outra melhoria importante foi a resolução de um erro de extração relacionado com abreviações. **Anteriormente, definições que continham a expressão "veg." ficavam incompletas**, comprometendo o conteúdo semântico logo nos primeiros conceitos. Esta situação foi corrigida e, atualmente, **o parser extrai corretamente as definições completas onde surgem abreviaturas como "veg.", sem perda de informação essencial**, contribuindo para uma base de dados terminológica mais coesa e fiel ao original.

3 Construção do JSON Final

A construção do ficheiro `glossario_final.json` resulta da combinação de vários scripts Python, cada um responsável por integrar diferentes fontes de dados. Nesta secção, explicamos passo a passo como o glossário final é construído, dividindo o processo consoante os ficheiros de processamento utilizados.

3.1 processar_multilingue_final.py

Este script tem como objetivo principal processar o ficheiro PDF multilingue da COVID-19, extraindo os termos e respetivas traduções. O processo envolve:

- Leitura e extração do texto do PDF;

- Identificação e separação das línguas (Português, Espanhol e Inglês);
- Criação de um dicionário para cada termo, onde são guardadas as traduções e a língua de origem;
- Atualização do glossário existente com as traduções encontradas, mantendo a estrutura dos conceitos;
- Caso o termo ainda não exista, é criada uma nova entrada com campos como: traduções, sinónimos, categoria léxica, entre outros.

Este script é responsável por garantir que o glossário final contenha equivalentes em várias línguas, enriquecendo a componente multilingue do dicionário.

3.2 processar_glossarioNeo.py

Este script tem como objetivo principal criar a estrutura base do JSON final, agregando informações do *GlossárioNeo*, abreviaturas, siglas e anexos.

Estrutura Inicial

Cria-se uma estrutura base com as chaves:

- SIGLAS
- ABREVS
- CONCEITOS
- CATEGORIAS
- ANEXOS

Processamento de Abreviaturas e Siglas

São lidos dois ficheiros distintos de abreviaturas. Cada chave é classificada como **sigla** se for composta apenas por letras maiúsculas. Caso contrário, é considerada uma abreviatura.

Processamento do Glossário

A informação lexical do glossário principal é lida de *Glossario/glossario.json*. Para cada termo:

- São guardadas traduções em espanhol e inglês.
- São inseridas definições associadas ao termo, com a fonte *GlossarioNeo*.
- É atribuída a sigla (se existir), distinguindo entre sigla válida e abreviatura.
- São registados campos como *categoria_lexica*, *artigos* e *info_enc*.

Processamento de Anexos

O ficheiro *Anexos/anexos_pesquisa.json* é adicionado à secção ANEXOS.

3.3 glossario_mini_processor.py

Este script complementa o *glossario_final.json*, atualizando-o com novas categorias, subcategorias, conceitos e siglas oriundas do **Glossário do Ministério da Saúde**.

Categorias e Subcategorias

- As **categorias principais** são lidas de `Areas.json` e adicionadas à secção CATEGORIAS, com uma breve definição.
- As **subcategorias** são lidas de `descritores.json` e associadas às respectivas categorias.

Siglas e Abreviaturas

As entradas de `siglas.json` são separadas entre siglas (compostas por letras maiúsculas) e abreviaturas. Estas são adicionadas às respectivas secções do glossário.

Conceitos

Cada conceito do ficheiro `conceitos.json` é processado:

- Se o termo já existir, é atualizado com novas definições e categorias.
- Se for um novo termo, é criada toda a estrutura base, incluindo campos como `definicoes`, `categoria_area`, `CAS`, `sinonimos`, entre outros.
- A fonte associada às definições é registada como `Glossario Ministerio da Saude`.

3.4 Extração e Atualização de Definições em Inglês (`definicoes_ingles.py`)

O script `definicoes_ingles.py` realiza *webscraping* de definições médicas em inglês a partir do site da Harvard Health Publishing. O processo segue os seguintes passos:

- Carrega um ficheiro JSON pré-existente que contém termos médicos e as suas traduções, organizados por conceitos.
- Accede a quatro páginas web, cada uma correspondente a um intervalo de letras do alfabeto (A-C, D-I, J-P, Q-Z).
- Para cada página, identifica os parágrafos que contêm termos destacados a negrito.
- Extrai o termo (removendo sinais como dois pontos) e a definição associada (texto imediatamente seguinte ao termo no parágrafo).
- Normaliza o termo para facilitar a comparação (tudo em minúsculas).
- Procura no JSON pré-existente se o termo extraído coincide com alguma tradução em inglês associada a um conceito.
- Se coincidir, adiciona a definição extraída, associando-a ao conceito correto e indicando a fonte (Harvard).
- Finalmente, guarda o JSON atualizado com as novas definições.

Este processo permite enriquecer o glossário com definições fiáveis e atualizadas retiradas diretamente de uma fonte reconhecida, garantindo uma ligação entre termos em inglês e conceitos estruturados.

3.5 Extração de Siglas e Abreviaturas (`siglas_abrev.py`)

O script `siglas_abrev.py` recolhe siglas e abreviaturas padronizadas de registos clínicos a partir de um site brasileiro. O processo consiste em:

- Carregar um ficheiro JSON já contendo o glossário completo com conceitos e outras informações.

- Aceder à página web que contém grupos organizados em painéis, onde cada painel tem uma tabela com siglas/abreviaturas e respetivos significados.
- Ignorar o grupo específico de símbolos, pois este não é relevante para o glossário.
- Percorrer todas as linhas das tabelas dos restantes grupos, extraindo as siglas/abreviaturas e os seus significados.
- Limpar os dados removendo caracteres especiais (ex: asteriscos) e espaços.
- Tratar casos onde existem múltiplas siglas numa mesma célula (exemplo: "ADA ou AEA"), separando-as.
- Classificar cada sigla/abreviatura como:
 - **Sigla**, se estiver em maiúsculas (ou não tiver letras minúsculas).
 - **Abreviatura**, caso contrário.
- Adicionar a sigla/abreviatura e significado ao respetivo dicionário dentro do JSON, apenas se ainda não existir.
- Guardar o JSON atualizado com as novas entradas.

Este processo contribui para completar o glossário com siglas e abreviaturas padronizadas utilizadas em registos médicos, garantindo a organização e distinção clara entre tipos de entradas.

4 *Web scraping*

De forma a enriquecer o *dataset* de conceitos médicos em formato *JSON* já existente, utilizou-se técnicas de *web scraping* para extrair informação adicional de duas fontes distintas. A integração destes novos dados teve como objetivo a adição de siglas/abreviações frequentemente utilizadas em contexto clínico e de definições em inglês a conceitos já existentes. Nas próximas subsecções serão mencionadas as fontes utilizadas.

4.1 Fonte 1: HGIS - Abreviações e Siglas

A primeira parte do processo de *web scraping* consistiu em complementar o glossário com siglas e abreviações padronizadas extraídas do *website* do HGIS (<https://hgis.org.br/abreviacoes-e-siglas-padronizadas-para-registros-do-prontuario/>).

Nas Figuras 1 e 2 é possível observar a página de abreviações e siglas padronizadas do HGIS.



Fig. 1: Página de abreviações e siglas padronizadas do HGIS.

— Grupo I.I - Diagnósticos

Abreviações / Siglas	Significado
A	Aborto
ADA ou AEA	Acrômio Direita Anterior ou Esquerda Anterior
ADP ou AEP	Acrômio Direito Posterior ou Esquerdo Posterior
AIG	Adequado para a idade gestacional
AIT ou TIA	Ataque isquêmico transitório (cerebral)
ANN	Anóxia Neonatal

Fig. 2: Página de abreviações e siglas padronizadas do HGIS.

O código implementado para a recolha dos dados realiza os seguintes passos:

4.1.1 Leitura do ficheiro

Carregamento do ficheiro 'glossario_final_completo.json', que já contém uma estrutura com conceitos médicos.

```

1 with open('../glossario_final_completo.json', 'r', encoding='
  utf-8') as f:
2     data = json.load(f)
3

```


4.1.2 Extração de conteúdo das tabelas

A página do HGIS organiza os dados em elementos do tipo *accordion*, cada um contendo uma tabela com siglas/abreviações e os seus significados (Figuras 3 e 4).

```
1 url = "https://hgis.org.br/abreviacoes-e-siglas-padronizadas-  
2 para-registros-do-prontuario/"  
3 response = requests.get(url)  
4 soup = BeautifulSoup(response.content, 'html.parser')  
  
5 # Encontrar todos os grupos (accordions)  
6 accordion_items = soup.select('.elementor-accordion-item')  
7  
8 for item in accordion_items:  
9
```

```
<div class="elementor-element elementor-element-5cd32ab elementor-widget eleme  
ntor-widget-accordion" data-id="5cd32ab" data-element_type="widget" data-  
widget_type="accordion.default">  
  <div class="elementor-widget-container">  
    <div class="elementor-accordion">  
      <div class="elementor-accordion-item">...</div>  
      <div class="elementor-accordion-item">...</div>  
      <div class="elementor-accordion-item">...</div>  
      <div class="elementor-accordion-item">...</div>  
      <div class="elementor-accordion-item">...</div>  
      <div class="elementor-accordion-item">...</div>  
      <div class="elementor-accordion-item">...</div>  
      <div class="elementor-accordion-item">...</div>  
      <div class="elementor-accordion-item">...</div>  
    </div>  
  </div>  
</div>
```

Fig. 3: Estrutura da página *HTML*.

[illegible]

Fig. 4: Estrutura da página *HTML*.

4.1.3 Filtragem e normalização

Cada linha das tabelas foi seguidamente processada, ignorando-se entradas inválidas ou grupos não relevantes (como o grupo de "Símbolos").

Foram também tratados casos com múltiplas formas para uma mesma sigla (ex: "ADA ou AEA").

```

1 for item in accordion_items:
2     # Verificar se e o grupo de simbolos
3     title = item.select_one('.elementor-accordion-title')
4     if title and "Simbolos" in title.get_text(strip=True):
5         print("Saltar grupo de simbolos")
6         continue
7
8     # Encontrar a tabela dentro do item
9     table = item.select_one('table')
10    if not table:
11        continue
12
13    # Processar cada linha da tabela (ignorando cabecalhos)
14    rows = table.select('tr:not(.FundoTabelaAzul)')
15    for row in rows:
16        cells = row.select('td')

```

```

17         if len(cells) < 2:
18             continue
19
20         abbr = cells[0].get_text(strip=True)
21         meaning = cells[1].get_text(strip=True)
22
23         # Pular entradas vazias ou sem significado
24         if not abbr or not meaning:
25             continue
26
27         # Remover asteriscos e espa os extras
28         abbr = abbr.replace('*', '').strip()
29
30         # Processar multiplas siglas (ex: "ADA ou AEA")
31         for single_abbr in [a.strip() for a in abbr.split(' ou ')]
32         if a.strip():

```

4.1.4 Classificação e inserção

Dependendo do formato dos dados extraídos, estes são classificados como siglas (SIGLAS), se as letras forem todas maiúsculas, ou abreviações (ABREVS) dentro do dicionário *JSON*. Apenas foram adicionadas siglas/abreviações que ainda não existem no *dataset*, evitando duplicações.

```

1     for single_abbr in [a.strip() for a in abbr.split(' ou ')] if
2     a.strip():
3         # Classificar como SIGLA (todas mai sculas) ou ABREV
4         if single_abbr.isupper() or not any(c.islower() for c in
5         single_abbr):
6             target_dict = data["SIGLAS"]
7             new_value = meaning
8         else:
9             target_dict = data["ABREVS"]
10            new_value = [meaning]
11
12            # Adicionar se nao existir
13            if single_abbr not in target_dict:
14                target_dict[single_abbr] = new_value
15                #print(f"adicionei {single_abbr} a {meaning}")

```

4.1.5 Atualização do *dataset*

Foi adicionado ao ficheiro 'glossario_siglas_abrevs.json' os dados atualizados.

```

1     with open('glossario_siglas_abrevs.json', 'w', encoding='utf
2     -8') as f:
3         json.dump(data, f, ensure_ascii=False, indent=4)

```

O *web scraping* realizado permitiu complementar o glossário com elementos abreviados que são frequentemente usados em procedimentos médicos e documentos clínicos, facilitando a interpretação e padronização da linguagem técnica.

4.2 Fonte 2: Harvard Health Publishing

A segunda parte do *web scraping* consistiu na extração de definições de termos médicos publicados no *website* da *Harvard Health Publishing* (<https://www.health.harvard.edu>).

Na Figura 5 é possível observar a página com conceitos médicos e a sua respetiva descrição da Harvard Medical School.



Fig. 5: Página com conceitos médicos e a sua respetiva descrição da Harvard Medical School.

Uma vez que os conceitos/descrições se encontram distribuídos por 4 páginas distintas seguindo uma ordem alfabética, foram extraídas as descrições de quatro páginas distintas, cada uma organizada por grupos alfabéticos (A-C, D-I, J-P, Q-Z).

O código implementado realiza os seguintes passos:

4.2.1 Leitura do ficheiro

Carregamento do ficheiro 'glossario_siglas_abrevs.json', já enriquecido com as siglas/abreviaturas anteriores.

4.2.2 Navegação pelas páginas do *website*:

Para cada *URL*, é feito um pedido *HTTP* e o conteúdo *HTML* é processado com a biblioteca *BeautifulSoup*.

```
1 # URLs para cada grupo de letras
2 urls = [
3     "https://www.health.harvard.edu/a-through-c",
4     "https://www.health.harvard.edu/d-through-i",
5     "https://www.health.harvard.edu/j-through-p",
6     "https://www.health.harvard.edu/q-through-z"]
7
```

```

8     for url in urls:
9         print(f"A processsar: {url}")
10        try:
11            response = requests.get(url)
12            response.raise_for_status()
13

```

4.2.3 Extração dos termos e definições

Após a visualização da estrutura *HTML* da página (Figura 6), realizou-se a análise de cada parágrafo da página. Se começar com um termo em negrito (), esse termo é considerado um potencial conceito. O restante do parágrafo é assumido como definição.

```

1     soup = BeautifulSoup(response.content, 'html.parser')
2     container = soup.select_one('div.content-repository-
content')
3
4     if not container:
5         print("Container principal nao encontrado")
6         continue
7
8     # Encontrar todos os elementos de paragrafo que contem
termos
9     for p in container.find_all('p'):
10        # Verificar se o par grafo contem um termo em
negrito
11        if p.strong:
12            term_text = p.strong.get_text(strip=True)
13
14            # Extrair o termo (remover os dois pontos finais)
15            term = term_text.rstrip(':').strip()
16            normalized_term = term.lower()
17
18            # Extrair defini o (texto ap s o termo)
19            definition = p.get_text().replace(term_text, '',
1).strip()
20
21

```

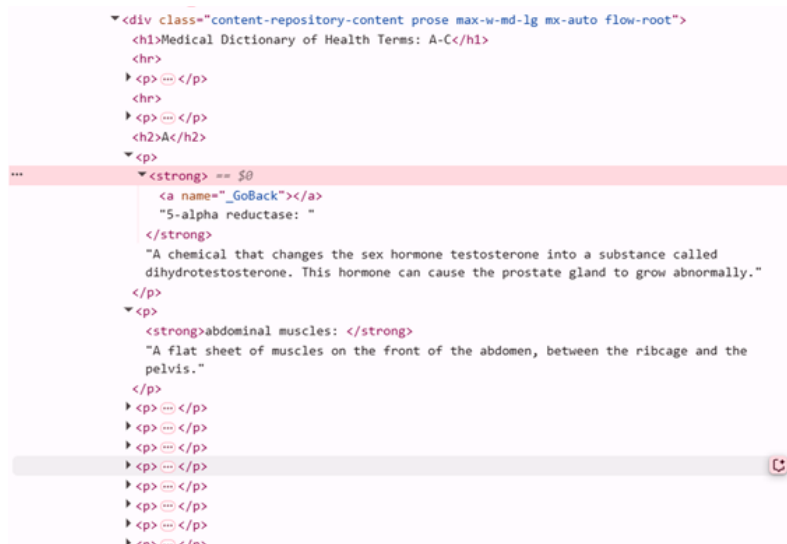


Fig. 6: Estrutura da página *HTML*.

4.2.4 Associação com conceitos existentes

O termo extraído é comparado com as traduções em inglês já existentes no ficheiro *JSON* carregado. Se for encontrado um termo coincidente, a nova definição é adicionada à lista de definições desse conceito, com a indicação da fonte ("Harvard").

As traduções em inglês presentes no ficheiro *JSON* estão organizadas como listas de strings e podem conter variações com hífens, apóstrofes, números ou espaços inconsistentes. De forma a garantir uma comparação eficaz, utilizou-se uma expressão regular aplicada ao conteúdo das traduções (convertido em texto com *json.dumps()*), com o objetivo de extrair termos compostos de forma robusta e normalizada. A regex aplicada possibilita:

- Capturar sequências de palavras (mínimo 2 caracteres alfanuméricos) com ou sem hífen ou apóstrofo;
- Preservar termos compostos como uma única unidade sem os separar;
- Ignorar pontuação irrelevante.

```

1   for concept, details in data["CONCEITOS"].items():
2       en_translations = details["traducoes"].get("en", [])
3       #print(f"en_translations:{en_translations}")
4
5       # Transformar a lista de traducoes em texto JSON simulado
6       translations_text = json.dumps(en_translations, indent=2)
7       #print(f"translations_text:{translations_text}")
8
9       # Aplicar a regex no texto
10      matches = re.findall(r'((?:(?:\w{2,})-?(?:\'\w*)?)|\d
    -?(?:\'\w*)?)\s?)+', translations_text)

```

```

11         #print(f"matches:{matches}")
12
13         # Extrair os termos encontrados pela regex
14         normalized_translations = [m.strip().lower() for m in
matches]
15         #print(f"normalized_translations:{normalized_translations
}")
16
17         for termo in normalized_translations:
18             #if normalized_term == "arrhythmia":
19                 #print(f"normalized_term: {normalized_term}")
20                 #print(f"termo: {termo}")
21             if termo == normalized_term:
22                 #print(f"normalized_term: {normalized_term}")
23                 #print(f"A adicionar definicao para: {concept}")
24                 details["definicoes"].append([definition, "
Harvard"])
25

```

4.2.5 Atualização do ficheiro *JSON*

O *dataset* é guardado com as novas definições no ficheiro 'glossario_final.json'.

Este processo permitiu adicionar explicações detalhadas a conceitos já existentes, reforçando a sua compreensão através de uma fonte de elevada credibilidade.

5 Backend

Nesta secção vai-se explorar o código desenvolvido para o *backend* da aplicação.

5.1 Funções de Apoio

As funções de apoio tratam da persistência e enriquecimento de dados e da lógica de pesquisa. `carregar_dados()` e `salvar_dados()` abstraem a leitura/escrita do ficheiro *JSON*, `processar_links()` converte termos em texto em hiperligação para navegação, e as cinco funções `pesquisar_*` aplicam filtros exatos ou parciais sobre conceitos, siglas, abreviaturas, categorias e anexos, retornando resultados com indicação de contexto.

5.1.1 Funções de Processamento do *JSON*

Estas funções são responsáveis por gerir a persistência dos dados da aplicação web, carregando e gravando os dados no ficheiro *dados.json* que serve como base de dados leve para todos os glossários, siglas, abreviaturas, categorias e anexos presentes.

Função carregar_dados()

O objetivo desta função é inicializar o estado da aplicação com os dados armazenados em disco, garantindo que todo o conteúdo previamente persistido no ficheiro *dados.json* fique disponível em memória para ser utilizado pelas rotas e funcionalidades implementadas. Ao ser invocada logo no início do código, assegura que a variável global `dados` reflete com precisão o

último *snapshot* dos conceitos, siglas, abreviaturas, categorias e anexos, de modo a apresentar ao utilizador uma visão consistente e atualizada da base de conhecimento.

O funcionamento pode ser descrito como um procedimento que abre o ficheiro *dados.json* em modo de leitura, especificando a codificação UTF-8 para suportar corretamente caracteres acentuados e símbolos médicos, e procede à desserialização do seu conteúdo para estruturas de dados *Python*. Em seguida, o objeto resultante é atribuído à variável global, permitindo que todas as rotas acessem de forma eficiente e sem necessidade de múltiplas leituras ao disco.

Função salvar_dados()

O objetivo desta função é assegurar a persistência de quaisquer modificações efetuadas em memória sobre o conjunto de dados, refletindo operações de criação, edição ou eliminação de conceitos, siglas, abreviaturas, categorias e anexos diretamente no ficheiro *dados.json*.

O funcionamento pode ser descrito como um processo atômico de escrita em disco, no qual o dicionário *Python* atualizado é passado como argumento, e então serializado para JSON com *json.dump*.

5.1.2 Função processar_links()

O objetivo desta função é enriquecer dinamicamente qualquer bloco de texto presente no frontend com hiperligações para os termos definidos no glossário, sejam estes conceitos, siglas ou abreviaturas, de modo a oferecer uma navegação contextual e facilitar a consulta imediata de definições relacionadas.

O funcionamento pode ser descrito por três etapas principais:

1. **Deteção de Conceitos:** Varre cada conceito presente na base de dados carregada, construindo para cada ocorrência uma tag `<a>` que aponta para a rota de detalhe desse conceito; quando se encontra no próprio contexto de exibição do conceito (origem), omite o parâmetro origem para evitar criar um link circular. A substituição utiliza expressões regulares sensíveis a limites de palavra, tendo em conta palavras maiúsculas/minúsculas.
2. **Deteção de Siglas:** Para cada sigla na base de dados, a função procura a sua presença no texto e envolve-a numa tag `<a>`. Este link abre reencaminha para página de detalhe da sigla e, quando o utilizador passa o rato por cima, exibe o significado completo da mesma através de um “tooltip”. A pesquisa no texto usa regex para garantir que a sigla é encontrada de forma exata e não no interior de outra palavra.
3. **Deteção de Abreviaturas:** De forma similar, para cada abreviatura função identifica no texto a sequência exata da abreviatura (por exemplo, “veg” ou “veg.”) e cria um link para a página dessa abreviatura. Além disso, cada link mostra, no “tooltip”, todas as definições armazenadas para essa abreviatura. Dessa maneira, sempre que um termo abreviado aparece, o utilizador pode clicar ou passar o rato por cima para consultar rapidamente o que ele significa.

No final, o texto transformado é marcado como HTML seguro (Markup) para que os `<a>` não sejam escapados pelo Jinja2, garantindo que todos os links inseridos apareçam corretamente na página. Esta abordagem assegura que, sempre que uma definição ou termo abreviado for mencionado em descrições, traduções, sinónimos ou informações enciclopédicas, o utilizador terá acesso imediato aos detalhes relevantes.

5.2 Funções de Pesquisa

As funções de pesquisa auxiliares partilham um mesmo propósito: permitir filtrar e encontrar entradas dentro de cada coleção (conceitos, siglas, abreviaturas, categorias e anexos) com base numa cadeia de caracteres fornecida pelo utilizador. Todas aceitam um parâmetro *query* (a sequência a procurar) e um booleano “exata” que determina se a correspondência deve ser literal ou parcial (presença em qualquer parte do texto, inclusiva no meio de uma palavra).

O objetivo genérico destas funções é devolver um conjunto de resultados relevantes acompanhado de informação contextual que ajude a identificar onde a consulta ocorreu seja no próprio nome do item, nas suas definições, traduções, etc, e assim orientar a interface a apresentar não apenas “o que” foi encontrado, mas também “onde”.

Olhando para o funcionamento detalhado de *pesquisar_conceitos()*, esta função quando invocada, converte a *query* para minúsculas e percorre todos os conceitos registados em memória. Para cada termo, verifica primeiro se o próprio nome do conceito corresponde (exatamente ou por contiguidade, dependendo de exata) e regista o contexto dessa correspondência. Em seguida, analisa cada definição associada, e procuram a *query* no texto da definição e, sempre que encontra, adiciona um fragmento dessa definição à lista de contextos. Depois, examina igualmente as traduções em cada língua, capturando menções exatas ou parciais e apontando a língua em que apareceram. Se pelo menos uma correspondência for identificada, o par “conceito + seus dados” integra o dicionário de resultados, cujo valor inclui tanto todas as informações originais do conceito como a lista de contextos recolhidos.

As restantes quatro funções seguem uma abordagem análoga, aplicando-a aos seus respetivos domínios:

- *pesquisar_siglas(query, exata)* opera sobre o dicionário de siglas, comparando a *query* tanto contra a sigla em si como contra o seu significado, e assinalando se a correspondência foi exata ou parcial.
- *pesquisar_abreviaturas(query, exata)* procura no código da abreviatura e, caso não encontre, varre cada um dos seus significados para correspondências parciais, reportando sempre o motivo de cada correspondência (código exato, contenção ou significado).
- *pesquisar_categorias(query, exata)* analisa o nome de cada categoria e a sua definição principal, bem como as subcategorias, assinalando se a *query* bateu exatamente ou aparece dentro de qualquer uma dessas partes.
- *pesquisar_anexos(query, exata)* itera sobre a lista de anexos (artigos), comparando a *query* com o título do artigo e devolvendo diretamente o registo completo sempre que a correspondência satisfaz o critério exato ou parcial.

5.3 Rotas Principais

As rotas principais definem o conjunto de entradas fundamentais à navegação e exploração do glossário: a página inicial (*index*) disponibiliza estatísticas sumárias sobre o conteúdo armazenado; a rota de pesquisa (*pesquisar*) concentra toda a lógica de busca integrada, permitindo consultas direccionadas ou universais; e as rotas de listagem (*listar_conceitos*, *listar_siglas*, *listar_abreviaturas*, *listar_categorias*, *listar_anexos*) exibem vistas filtráveis de cada tipo de recurso, suportando tanto pesquisas exatas como parciais diretamente sobre os seus respectivos títulos ou chaves. Juntas, estas rotas estruturam a experiência de descoberta e consulta, garantindo acesso rápido e consistente a qualquer segmento dos dados.

Função `index()`

O objetivo desta rota é encaminhar o utilizador para a página inicial da aplicação, fornecendo ao utilizador uma visão global e estatística do conteúdo disponível. O seu funcionamento consiste em contar, diretamente na variável global “dados”, o número de conceitos, siglas, abreviaturas, categorias e anexos registados, e passar esses valores para o template *index.html*. Deste modo, o utilizador é recebido com indicadores quantificados do tamanho da base de dados, servindo como ponto de partida para navegação ou pesquisa.

Função `pesquisar()`

O objetivo desta rota é centralizar toda a lógica de pesquisa integrada, permitindo ao utilizador procurar um termo livremente em qualquer um dos cinco domínios (conceitos, siglas, abreviaturas, categorias e anexos) ou restringir a pesquisa a um único tipo de dado. O funcionamento pode ser descrito como segue: a rota obtém os parâmetros *q* (texto de busca), *tipo* (área de pesquisa: tudo, conceitos, siglas etc.) e *exata* (booleano que indica comparação literal versus parcial). Se não for fornecida uma *query*, redireciona para o *index*. Caso contrário, invoca as funções auxiliares de pesquisa correspondentes ao(s) domínio(s) selecionado(s) e agrega os resultados num único dicionário, que inclui sempre listas ou mapas vazios para manter a coerência. Por fim, renderiza *resultados_pesquisa.html*, passando a *query*, o tipo selecionado, o modo *exato*/*parcial* e os resultados obtidos para exibição.

Funções de Listagem de Conceitos, Siglas, Abreviaturas, Categorias e Anexos

O objetivo destas rotas (*listar_conceitos()*, *listar_siglas()*, *listar_abreviaturas()*, *listar_categorias()* e *listar_anexos()*) é apresentar listas filtráveis dos diferentes recursos armazenados, facultando ao utilizador uma forma direta de navegar ou pesquisar dentro de cada coleção.

O funcionamento genérico pode ser ilustrado pela rota de *listar_conceitos()*: ao receber uma eventual *query* de filtro (*q*) e o indicador *exata*, a função verifica se o utilizador pretende ver todos os conceitos ou apenas aqueles cujo nome coincide totalmente com a *query* (modo *exato*) ou contém a sequência de caracteres (modo *parcial*). Em caso afirmativo, constrói um sub-dicionário apenas com as entradas que satisfazem o critério; caso contrário, devolve o dicionário completo. Finalmente, invoca o template *conceitos.html*, passando a lista resultante, a *query* usada e o modo de correspondência. As rotas de listagem para siglas, abreviaturas, categorias e anexos seguem abordagem idêntica.

5.4 Rotas de detalhe

As rotas de detalhe têm como propósito apresentar uma página dedicada a um único recurso, seja um conceito, uma sigla, uma abreviatura ou uma categoria. Reúne toda a informação relevante e interligando-a com o restante glossário. Funcionamento de *detalhe_conceito(nome)* Ao receber o nome de um conceito na URL, esta rota carrega o respetivo objeto e processa cada bloco de texto apresentado (definições, traduções, etc.) para a inserção automática de hiperligações a outros termos (outros conceitos/siglas/abreviaturas mencionadas). As restantes rotas de detalhe, *detalhe_sigla()*, *detalhe_categoria()* e *detalhe_abreviatura()* seguem estratégia semelhante.

5.5 Funções de Operações CRUD

Para uniformizar a manutenção de todos os recursos, sejam Conceitos, Siglas, Abreviaturas, Categorias ou Anexos, as operações de criação, atualização e remoção seguem um padrão consistente: cada entidade dispõe de rotas dedicadas para exibir formulários (quando aplicável), validar e processar o input do utilizador, atualizar os dados, e redirecionar com mensagens de feedback. A seguir detalham-se o funcionamento destas operações para Conceitos. As rotas equivalentes para Siglas, Abreviaturas, Categorias e Anexos aplicam exatamente a mesma lógica, adaptando apenas os campos e templates específicos de cada entidade.

Função adicionar_conceito()

O objetivo desta função é expor ao utilizador um formulário para criação de um novo conceito e, em submissão, validar e persistir essa entrada no glossário. O funcionamento inicia-se num **GET**, em que carrega todas as categorias disponíveis para popular os menus de seleção no formulário. Quando chega um **POST**, a rota extrai cada campo do formulário: nome, definição, informação enciclopédica, categoria léxica, sigla, CAS, sinónimos e traduções em várias línguas, categorias de área e artigos relacionados e aplica validações básicas (ex.: existência obrigatória de nome e definição, unicidade do nome, filtragem de artigos inválidos). Se tudo estiver correto, constrói um dicionário *novo_conceito* na ordem desejada, insere-o em na base de dados e redireciona para a página com a listagem dos conceitos com uma mensagem de sucesso. Erros de validação provocam redirecionamentos de volta ao formulário com parâmetros que exibem feedback ao utilizador.

Função editar_conceito()

O objetivo desta função é permitir a atualização de um conceito existente, apresentando um formulário pré-povoado e aplicando as alterações submetidas. Em **GET**, a rota recupera o objeto pelo seu nome e extrai os seus campos iniciais (primeira definição, info enciclopédica, categorias, sinónimos, traduções, CAS, sigla, artigos), passando-os ao mesmo template com o formulário, porém já com alguns campos preenchido. Em **POST**, valida-se a presença de definição e, caso exista, atualiza-se o primeiro item da lista de definições (ou cria-se um novo se estiver vazio), bem como todos os demais campos conforme o input do utilizador. Constrói-se um novo dicionário *conceito_atualizado* para manter a ordem dos campos, substitui-se a entrada na base de dados.

Função eliminar_conceito()

O objetivo desta função é remover de forma definitiva um conceito da base de dados. Funcionando apenas por **POST**, verifica se a chave “nome” existe na base de dados, se existir, procede-se à remoção com deste conceito. Posteriormente invoca-se *salvar_dados(dados)* para guardar as alterações no JSON e redireciona-se o utilizador para a lista de conceitos com uma mensagem de sucesso.

6 Funcionalidades da aplicação web

A aplicação *web* desenvolvida visa facilitar a consulta e gestão de termos médicos, incluindo conceitos, siglas, abreviaturas, categorias e anexos através de uma interface intuitiva. Seguidamente, detalhamos as funcionalidades chave de cada secção principal da aplicação.

6.1 Home

A página inicial apresenta uma *interface* com uma barra de pesquisa centralizada, permitindo aos utilizadores procurar conceitos, siglas, abreviaturas, categorias e anexos em todo o glossário (Figura 7).



Fig. 7: Página inicial da aplicação *web*.

Esta barra de pesquisa permite fazer uma pesquisa bastante abrangente não se limitando apenas aos termos em si. Isto é, realizando por exemplo uma pesquisa pelo termo "alzheimer" (Figura 8), é possível verificar nas Figuras 9 e 10 que os resultados da pesquisa abrangem tanto os termos em si quanto as suas definições e traduções proporcionando uma procura abrangente e rápida.

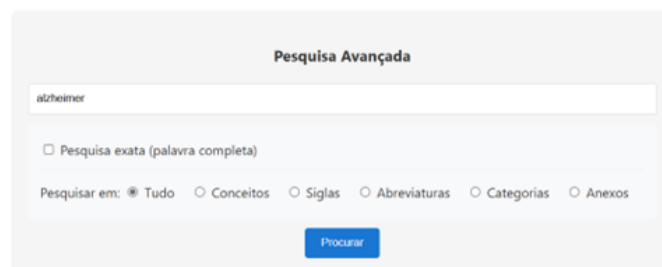


Fig. 8: Pesquisa pelo termo "alzheimer".

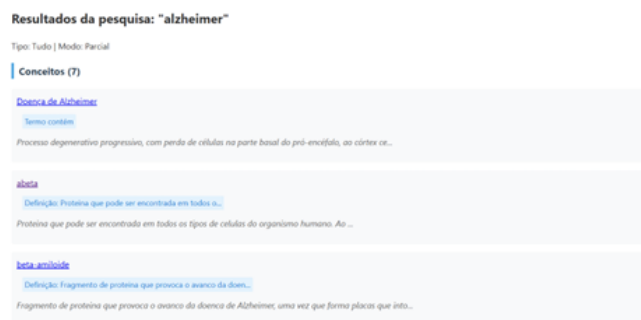


Fig. 9: Resultados da pesquisa pelo termo "alzheimer".



Fig. 10: Resultados da pesquisa pelo termo "alzheimer".

Além disso, também nesta página é possível visualizar o número total de conceitos, siglas, abreviaturas, categorias e anexos que fazem parte deste glossário médico, como se pode verificar na Figura 11.

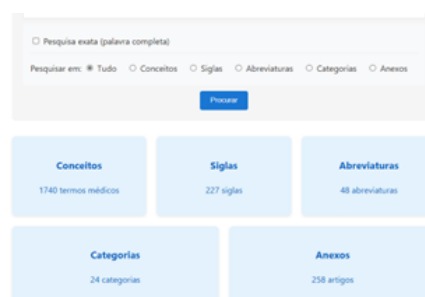


Fig. 11: Página inicial da aplicação web.

6.2 Conceitos

A secção dos Conceitos é dedicada à visualização dos conceitos médicos presentes no glossário e as suas respetivas definições detalhadas.

6.2.1 Listagem dos Conceitos

Esta página apresenta uma lista de todos os conceitos disponíveis, como se verifica na Figura 12.



Fig. 12: Lista de todos os conceitos disponíveis

6.2.2 Pesquisa

A funcionalidade de pesquisa foi implementada para permitir ao utilizador localizar rapidamente conceitos no glossário através dos seus nomes. Trata-se de uma pesquisa dinâmica, em que os resultados são atualizados em tempo real à medida que o utilizador digita.

Esta funcionalidade suporta dois modos principais de pesquisa:

- **Pesquisa parcial (dinâmica):** à medida que se introduzem caracteres, são apresentados todos os conceitos cujo nome contenha o texto inserido, independentemente da sua posição na palavra.
- **Pesquisa exata:** o sistema pode ser configurado para apresentar apenas os conceitos cujo nome corresponda exatamente ao termo inserido pelo utilizador.

6.2.3 Visualização Detalhada do Conceito

Ao clicar num conceito, o utilizador é direcionado para uma página específica que exhibe os detalhes completos relativos a esse mesmo conceito. São exibidas informações mediante as informações disponíveis no glossário, podendo incluir informações enciclopédicas, categoria lexical, siglas associadas, sinónimos em diferentes idiomas (Português, Inglês, Espanhol), traduções para diversas línguas (Inglês, Espanhol, Árabe, Catalão, Francês, Galego).

Na Figura 13 pode-se verificar os detalhes relativos ao conceito "abeta".

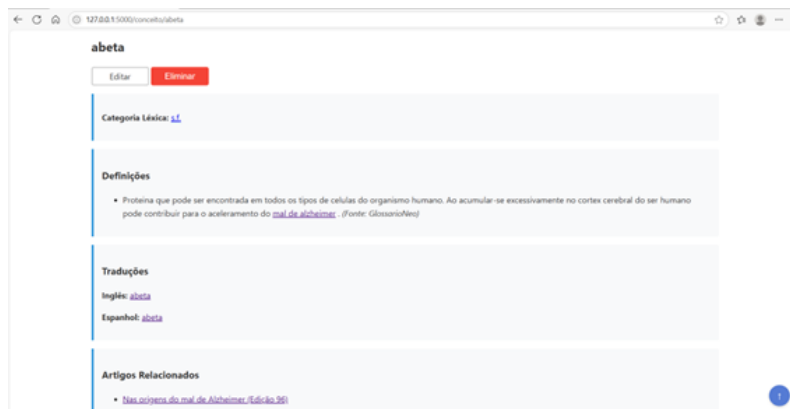


Fig. 13: Detalhes relativos ao conceito "abeta".

6.2.4 Ligação de Termos

Selecionando um conceito, nos detalhes do conceito em questão, a aplicação processa e transforma termos relevantes (outras siglas, abreviaturas ou conceitos) em hiperligações clicáveis, permitindo uma navegação fluída entre os termos relacionados.

Selecionando "mal de alzheimer" (Figura 13) vamos para uma outra página que exhibe os detalhes do conceito "mal de alzheimer", como é possível confirmar na Figura 14.

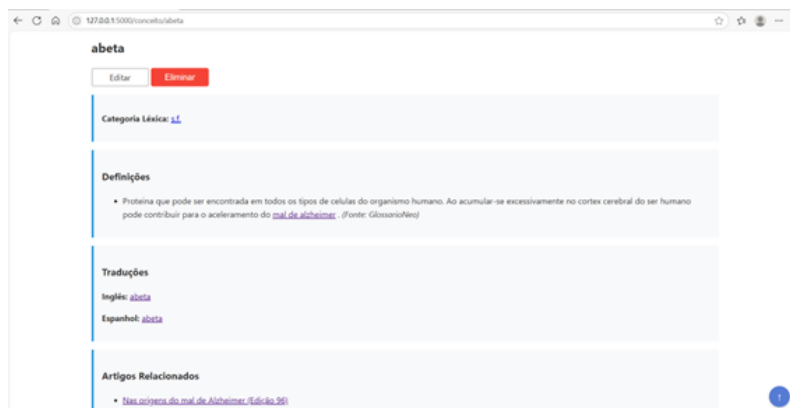


Fig. 14: Detalhes relativos ao conceito "abeta".

6.2.5 Artigos Relacionados

Os conceitos podem ter artigos relacionados. Selecionando um artigo, o utilizador é levado diretamente para a secção correspondente na tabela de anexos, com uma opção para regressar facilmente à página do conceito de origem.

Selecioneando "Nas origens do mal de Alzheimer (Edição 96)" ((Figura 13), o utilizador é direcionado para a tabela dos Anexos onde a primeira posição da tabela corresponde a esse mesmo artigo (Figura 15).

159	Nas origens do mal de Alzheimer	Nº 96 (Setembro)	2004	Editar Eliminar
160	No coração dos genes	Nº 94 (Setembro)	2003	Editar Eliminar
161	Nova base para drogas anti-HIV	Nº 87 (maio)	2003	Editar Eliminar
162	Nova técnica para tratar doença ocular	Nº 78 (agosto)	2003	Editar Eliminar
163	Novidade do oriente	Nº 97 (março)	2004	Editar Eliminar

Fig. 15: Seleção do artigo "Nas origens do mal de Alzheimer (Edição 96)".

6.2.6 Funcionalidades Adicionar, Editar, Eliminar

- **Adicionar Novo Conceito:** Existe um botão "Adicionar Conceito" na página dos Conceitos que redireciona o utilizador para um formulário onde é possível inserir um novo termo médico, se este ainda não existir no glossário médico. Apenas o campo "Definição" é de preenchimento obrigatório e apenas as categorias existentes no glossário podem ser escolhidas. Além disso, é somente aceitável adicionar artigos relacionados se estes já existirem no glossário.
- **Editar Conceito:** Na página de detalhes de cada conceito, existem 2 botões "Editar" e "Eliminar". O botão "Editar" leva o utilizador a um formulário pré-preenchido com os dados existentes do conceito, permitindo a modificação de todos os campos (definição, informação enciclopédica, categoria lexical, sigla, CAS, sinónimos, traduções e artigos relacionados), exceto o nome do conceito.
- **Eliminar Conceito:** O botão "Eliminar" inicia um processo de remoção do conceito, com uma confirmação para prevenir eliminações acidentais.

Adicionando um novo conceito "teste" ao glossário (Figura 16) e salvando, pode-se observar que o mesmo fica guardado no glossário, como se verifica na Figura 17.

Adicionar Conceito

Nome do Conceito:

teste

Definição *

teste

Informações Enciclopédicas:

Fig. 16: Criação de um novo conceito "teste".

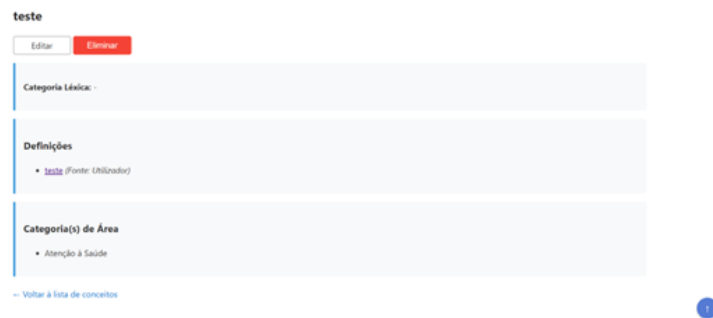


Fig. 17: Confirmação da criação do novo conceito "teste".

6.3 Siglas

A secção de Siglas permite a consulta e gestão de termos que são acrónimos ou abreviações.

6.3.1 Listagem

Esta página apresenta uma tabela que lista todas as siglas presentes no dicionário médico e os seus respetivos significados (Figura 18).



Fig. 18: Tabela com todas as siglas e os seus respetivos significados.

6.3.2 Pesquisa

A funcionalidade de pesquisa possibilita ao utilizador localizar rapidamente siglas específicas no glossário.

Esta funcionalidade suporta dois modos principais de pesquisa:

- **Pesquisa parcial (dinâmica):** são apresentadas todas as siglas que contenham o texto inserido, independentemente da sua posição na palavra.

- **Pesquisa exata:** somente são apresentadas as siglas que correspondam exatamente ao texto inserido pelo utilizador.

6.3.3 Visualização da sigla

Cada sigla pode ser selecionada para exibir o seu significado completo numa outra página, similar aos conceitos.

Na Figura 19 é exibida a sigla "ADT" e o seu respetivo significado após a seleção desta sigla.

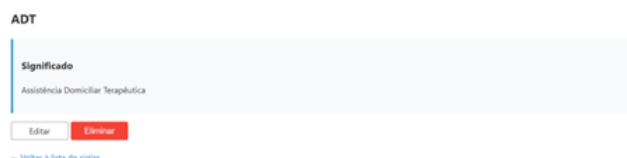


Fig. 19: Significado da sigla "ADT".

6.3.4 Funcionalidades Adicionar, Editar, Eliminar

- **Adicionar Nova Sigla:** Existe um botão "Adicionar Nova Sigla" que direciona o utilizador para um formulário de criação de novas siglas. Apenas é possível criar siglas que ainda não existam no glossário.
- **Editar Sigla:** Na página de detalhes de cada sigla, existem os botões "Editar" e "Eliminar" que permitem a modificação dos dados ou a remoção de siglas. O botão "Editar" abre um formulário pré-preenchido com os dados da sigla para alteração.
- **Eliminar Sigla:** O botão "Eliminar" permite a remoção da sigla após uma confirmação.

6.4 Abreviaturas

A secção de Abreviaturas é similar à anterior, focada na consulta e gestão de abreviaturas.

6.4.1 Listagem

Similar às Siglas, esta página exibe uma tabela com as abreviaturas e os seus significados (Figura 20).

Abreviaturas

[Adicionar Nova Abreviatura](#)

Digite uma abreviatura no pelo desta:

☐ Pesquisa exata (palavra completa)

[Pesquisar](#)

Abreviatura	Significados
ABR	Associação Brasileira de Enfermagem
BD-SUS	Banco de Dados Nacional do Sistema de Informações Ambulatoriais do SUS
Bilat	Bilateral
CAS	número CAS
CN-DST/AIDS	Coordenação Nacional de Doenças Sexualmente Transmissíveis e Aids

Fig. 20: Tabela com todas as abreviaturas e os seus respectivos significados.

6.4.2 Pesquisa

A funcionalidade de pesquisa possibilita ao utilizador localizar rapidamente abreviaturas específicas no dicionário médico.

Esta funcionalidade suporta dois modos principais de pesquisa:

- **Pesquisa parcial (dinâmica):** são apresentadas todas as abreviaturas que contenham o texto inserido, independentemente da sua posição na palavra.
- **Pesquisa exata:** somente são apresentadas as abreviaturas que correspondam exatamente ao texto inserido pelo utilizador.

6.4.3 Visualização da abreviatura

Selecionando uma abreviatura específica, o utilizador acede a uma página com o nome e significado.

Na Figura 21 é exibida a abreviatura "Kcal" e o seu respetivo significado após a seleção desta abreviatura.

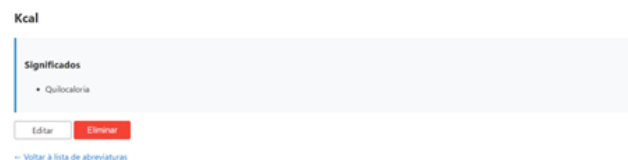


Fig. 21: Significado da abreviatura "Kcal".

6.4.4 Funcionalidades Adicionar, Editar, Eliminar

- **Adicionar Nova Abreviatura:** É apresentado um botão "Adicionar Nova Abreviatura" que possibilita a criação de novas abreviaturas que ainda não existam no glossário.

- **Editar Abreviatura:** Na página relativa a cada abreviatura, existem os botões "Editar" e "Eliminar" que estão disponíveis para modificação e remoção, respetivamente. O formulário de edição é pré-preenchido com os dados da abreviatura.
- **Eliminar Abreviatura:** A abreviatura pode ser removida com uma confirmação antes da eliminação final.

6.5 Categorias

A secção das Categorias organiza os conceitos do glossário em grandes temas médicos, permitindo uma navegação mais estruturada e intuitiva por áreas de conhecimento.

6.5.1 Listagem das Categorias

A página de categorias apresenta uma listagem de todas as categorias existentes, com uma breve descrição (definição) e, quando aplicável, uma listagem das subcategorias associadas. Esta organização facilita o acesso a grupos temáticos relacionados e oferece uma visão geral das áreas abordadas no glossário. A Figura 22 ilustra esta listagem.

Categorias Médicas

Adicionar Categoria

Digite o nome da categoria ou subcategoria...

☐ Pesquisa exata (palavra completa)

Pesquisar em:

☒ Nome da categoria

☐ Subcategorias

Procurar

Acidentes e Violência

Refere-se ao conjunto de agravos à saúde que pode levar a óbito ou sequelas irreversíveis que inclui as causas ditas acidentais: devidas ao trânsito, trabalho, quedas, envenenamentos, afogamentos e ou...

Subcategorias:

- Abuso financeiro em idosos

Administração e Planeamento em Saúde

Refere-se à organização, elaboração de planos e políticas públicas e governamentais, ao conjunto de normas e funções regidas por princípios e práticas administrativas, gestão, planificação e gerência...

Subcategorias:

- Acreditação hospitalar

Alimentação e Nutrição

Refere-se a todos os tipos de substâncias que têm por função alimentar ou nutrir seres humanos e animais. Inclui subtemas como: avaliação alimentar e nutricional, necessidades nutricionais e alimentares...

Subcategorias:

- Alimentação casual

Fig. 22: Listagem das categorias e respetivas subcategorias.

6.5.2 Pesquisa de Categorias

O sistema disponibiliza uma funcionalidade de pesquisa dedicada à consulta de categorias. Esta permite ao utilizador procurar tanto no nome das categorias como nas suas subcategorias, podendo ainda optar por uma correspondência exata ou parcial dos termos introduzidos. Estas opções de filtragem aumentam a flexibilidade e precisão na localização da informação pretendida.

- **Pesquisa parcial:** devolve todas as categorias ou subcategorias cujo nome contenha a expressão pesquisada.
- **Pesquisa exata:** devolve apenas os resultados que correspondem exatamente ao termo introduzido.
- **Campo de pesquisa:** o utilizador pode especificar se pretende pesquisar no nome da categoria ou nas subcategorias.

6.5.3 Visualização Detalhada de Categoria

Ao selecionar uma categoria, o utilizador é direcionado para a página correspondente que apresenta a definição da categoria, a lista completa das suas subcategorias (caso existam) e os conceitos relacionados. Estes conceitos estão interligados por hiperligações, permitindo uma navegação fluida entre termos, facilitando a exploração do glossário.

A Figura 23 mostra em detalhe uma categoria, incluindo a sua definição, subcategorias e ligações para os conceitos associados.

Comunicação em Saúde

Definição
Refere-se ao conjunto dos meios de comunicação de massa voltados a divulgação de produtos, serviços, ações preventivas e identificação de riscos relacionados à saúde ou morbididades de interesse individual ou coletivo. Inclui subtemas como: mídia (jornal, rádio, televisão, conteúdos e imagens digitais, etc), redes de informação especializadas, comunicação social, revistas, campanhas sanitárias, divulgação de descobertas científicas e tecnológicas, etc.

Subcategorias (3)

- Biblioteca Virtual em Saúde
- Biblioteca viva
- Propaganda de produtos farmacêuticos

Conceitos Relacionados

- [Biblioteca Virtual em Saúde](#)
- [Biblioteca viva](#)
- [Propaganda de produtos farmacêuticos](#)

[← Voltar à lista de categorias](#)

Fig. 23: Visualização detalhada de uma categoria.

6.5.4 Funcionalidades Adicionar, Editar, Eliminar

Tal como nas restantes secções, é possível gerir as categorias através de operações básicas:

- **Adicionar Categoria:** Através do botão "Adicionar Categoria", é possível inserir uma nova categoria no sistema, fornecendo o nome e a definição. Podem ainda ser adicionadas subcategorias.
- **Editar Categoria:** Cada categoria pode ser modificada, incluindo a sua definição e lista de subcategorias.
- **Eliminar Categoria:** É possível eliminar uma categoria, sendo solicitada uma confirmação para evitar remoções acidentais.

Estas funcionalidades permitem manter a organização temática do glossário sempre atualizada e coerente com os conceitos nele presentes.

7 Secção Anexos

Esta página apresenta uma tabela com os anexos existentes, onde cada linha representa um anexo, contém o número, o título do artigo, a edição (número e mês), o ano e ações disponíveis.

No topo da página está presente um botão “*Adicionar Anexo*”, que redireciona para um formulário de criação de novos anexos. Abaixo, existe uma barra de pesquisa que permite ao utilizador procurar anexos pelo título do artigo. Esta pesquisa pode ser feita por palavra-chave ou, opcionalmente, como uma **pesquisa exata**, ativada por uma checkbox.

Após uma pesquisa, caso existam resultados, é apresentada uma mensagem com os termos utilizados. A interface também contempla mensagens de sucesso e de erro, que são exibidas conforme o resultado das ações realizadas (por exemplo, ao adicionar ou remover anexos).

A listagem dos anexos é feita numa tabela com cinco colunas:

- **Número:** identificador do anexo.
- **Título do Artigo:** título ao qual o anexo está associado.
- **Edição:** número e mês da edição.
- **Ano:** ano da publicação.
- **Ações:** permite editar ou eliminar o anexo.

Cada linha da tabela possui dois botões de ação:

- **Editar:** redireciona para a página de edição do anexo correspondente.
- **Eliminar:** submete um formulário POST com confirmação, para remover o anexo da base de dados.

Adicionalmente, se a página for acedida a partir da visualização de um conceito, é apresentado um botão para voltar à página do conceito de origem, garantindo uma navegação fluida.

Esta secção segue a mesma estrutura e estilo das restantes páginas da aplicação, garantindo consistência visual e intuitiva funcionalidade ao utilizador.

Anexos

Adicionar Anexo

Digite o título do artigo...

☐ Pesquisa exata (palavra completa)

Procurar

Número	Título do Artigo	Edição	Ano	Ações
01	A base da diferença	Nº 68 (setembro)	2001	Editar Eliminar
02	A corrida pelo mapeamento	Nº 52 (abril)	2000	Editar Eliminar
03	A favor da mulher	Nº 101 (julho)	2004	Editar Eliminar
04	A flor da pele	Nº 88 (junho)	2003	Editar Eliminar
05	A frente dos desafios tropicais	Nº 100 (junho)	2004	Editar Eliminar

Fig. 24: Página de gestão de anexos, com pesquisa e tabela de ações.

8 Conclusão

Este projeto evidenciou a importância do processamento de linguagem natural na construção de um glossário médico integrado. Foi necessário recolher informação a partir de documentos em PDF, bases de dados online e glossários prévios, cada um com formatos e convenções próprias. Para obter os termos clínicos, siglas e abreviaturas, aplicámos web scraping, extração de texto e expressões regulares que permitiram isolar definições e traduções em várias línguas. Depois, toda essa informação foi limpa, normalizada e estruturada num único ficheiro JSON, assegurando consistência nos identificadores e qualidade nos dados.

A base de dados leve resultante foi utilizada para criar uma aplicação web em Flask capaz de apresentar definições, gerar hiperligações contextuais e suportar operações de pesquisa e edição de forma intuitiva. Em termos de aprendizagem, este trabalho permitiu consolidar conhecimentos em métodos de extração e transformação de texto clínico, no mapeamento multilingue de conceitos e na implementação de rotas RESTful com persistência em JSON. A experiência adquirida abrangeu todo o ciclo de vida de um projeto de PLN aplicado, desde a aquisição de dados até à entrega de uma interface funcional e colaborativa.