



**Universidade do Minho**  
Escola de Engenharia

# Processamento de Linguagem Natural em Engenharia Biomédica

**Grupo 1:**

Alícia Soares Oliveira, PG50169

Ana Carolina Veloso, PG50178

Margarida Ferreira Fernandes, PG50588

**Docente:**

Luís Filipe Costa Cunha,

José João Antunes Guimarães Dias Almeida

19 de Abril de 2023

# Índice

<i>1.Introdução.....</i>	<i>3</i>
<i>2.Desenvolvimento .....</i>	<i>3</i>
2.1. Trabalho Individual do PDF obrigatório .....	3
2.2. Trabalho Individual do PDF anatomia geral .....	6
2.3. Trabalho Individual do PDF dicionário de termos médicos e de enfermagem .....	9
2.4. Resultado – combinação dos 3 jsons .....	13
<i>3.Conclusão.....</i>	<i>14</i>

## 1.Introdução

O projeto em causa tem como objetivo aplicar o conhecimento adquirido em sala de aula para extrair informações relevantes de documentos médicos em PDF e armazená-las para uso futuro. Para alcançar esse objetivo, é necessário criar *parsers* capazes de identificar e extrair as informações relevantes de cada arquivo PDF, que serão posteriormente preservadas em arquivos JSON.

## 2.Desenvolvimento

Numa primeira fase, escolheram-se o documento obrigatório e dois outros documentos PDF para selecionar a informação relevante de cada um. Os documentos escolhidos denominam-se:

- “Anatomia Geral.pdf”
- “Dicionario\_de\_termos\_medicos\_e\_de\_enfermagem.pdf”.

Inicialmente, a estratégia passou por converter os pdf's em txt e a partir daí limpá-los e selecionar a informação relevante. No entanto, o grupo verificou uma elevada complexidade na tarefa e, por essa razão, alterou o tipo de ficheiro para o qual o pdf é convertido. Deste modo, todos os ficheiros pdf's foram convertidos a xml através do comando: `pdftohtml -c -xml nomeFicheiro.pdf`.

De seguida, é explicada em detalhe a limpeza e seleção da informação em cada um dos documentos escolhidos.

### 2.1. Trabalho Individual do PDF português-inglês-espanhol

Numa fase inicial, realizou-se uma limpeza ao ficheiro xml, dividida nas seguintes etapas, que recorrem a uma expressão regular adequada:

- Eliminou-se todo o ficheiro antes do início do dicionário em português, porque a informação está triplicada ao longo do dicionário, mudando apenas a ordem das línguas. Nesta limpeza usou-se a expressão regular: **<b>Dicionário de termos médicos.\*** .
- Apagaram-se todas as partes auxiliares do ficheiro xml, com várias expressões regulares, nomeadamente a seguinte: `\n</page>\n<.*>`, que permite eliminar as quebras de página. Seguindo o mesmo raciocínio, foram também eliminadas as partes iniciais de cada linha de texto, conservando-se apenas a informação relativa ao texto, no caso da figura abaixo, mantendo apenas `<b>abaulamento</b>`. Na limpeza, foram também apagadas as partes finais de cada linha de texto, ou seja, os excertos `</text>`. Por fim, ainda se verificou que seria necessário retirar as indicações das palavras em itálico (ou seja, entre `<i>...</i>`) e também as letras m e f que apareciam no final de cada termo.

```
<text top="191" left="59" width="69" height="11" font="16"><b>abaulamento</b></text>
```

- De seguida procedeu-se à eliminação da parte inicial de cada página, uma vez que cada uma tinha um cabeçalho extenso. Da mesma forma, procurou-se eliminar a parte inicial do documento apresentada em baixo.

```
<b>Dicionário de termos médicos</b>
<b>português  [ ] inglês  [ ] espanhol</b>
<b>português</b>
<b>[ ]</b>
<b>inglês</b>
<b>[ ]</b>
<b>espanhol</b>
```

- Nesta fase verificou-se essencial corrigir as palavras que se encontravam divididas em 2 linhas, que se encontravam com um hífen, como na figura à esquerda a palavra abduc-ción.

```
<b>abdução</b>
U
abduction
E
abduc-
ción
```

- Reparou-se, porém, que ainda existiam alguns artefactos desnecessários ao longo do texto, para os quais se utilizaram as seguintes expressões regulares: `\s*<fontspec.*>/>`, esta para limpar o que aparece na figura à esquerda. Em relação a estas expressões foi necessária uma limpeza mais intensiva, de forma a eliminar ainda o que se apresenta na figura à direita. Na mesma ótica foi interessante eliminar as letras que anunciavam a mudança de letra, por exemplo, quando o dicionário acabava as palavras com A e passava para a letra B.

```
ting room (OR)
E
quirófano
<b>259</b>
<b>bloco</b>
<fontspec id="28" size="10" family="NMMLP+SSymbol" color="#131413"
<b>bloqueador</b>
β
U
```

```
<b>atetose</b>
<b>258</b>
<b>B</b>
<b>basile</b>
```

- Verificou-se que o ficheiro possuía cabeçalhos diferentes entre páginas de números ímpares e pares, nomeadamente, em números ímpares na figura à esquerda e em páginas pares o que é apresentado na figura à direita.

```
U
cerebrovascular accident, cere<b>249</b>
<b>acidente</b>
bral apoplexy
E
```

```
<b>anestésico</b>
<b>254</b>
<b>antiácido</b>
```

- De forma a juntar os termos que estavam separados por <\b> e por outros temas, verificaram-se 2 situações: uma geral que tratava de limpar a informação selecionada na figura à esquerda, e uma bastante específica para limpar a figura à direita.

```
<b>aberração</b>
<b>cromossómica</b>
```

```
<b>bloqueador</b>
β
U
```

- Por fim, foi fundamental limpar e uniformizar as traduções. Nomeadamente, para captar as traduções em inglês era necessário capturar todas as linhas entre U e E. No entanto, existiam situações com 2, 3 e 4 linhas, por isso uniformizaram-se estas informações de forma a aparecer todo o conteúdo numa só linha. Assim, realizou-se o mesmo procedimento para as traduções em espanhol. Nas figuras verifica-se a diferença que as expressões regulares provocaram.

```
<b>dióxido de carbono (CO 2 )</b>
U
carbon
dioxide (CO
2
)
E
dióxido
de carbono (CO
2
)
<b>diplopia</b>
```

```
<b>dióxido de carbono (CO 2 )</b>
U
carbon dioxide (CO 2 )
E
dióxido de carbono (CO 2 )
```

Após a limpeza do ficheiro xml, procurou-se extrair os termos e respetivas traduções. Os termos, caracterizavam-se por estarem sempre rodeados por `<b>termo</b>` e, por isso, foram captados através da expressão regular, `<b>(.*?)</b>`. A tradução inglesa foi identificada por estar rodeada por U e E, e, por isso, desenvolveu-se a expressão regular: `U\n(.*?)\nE`. O mesmo raciocínio foi aplicado para a tradução espanhola que se encontra sempre rodeada entre E e `<b>`, sendo que, para este caso se utilizou a expressão: `E\n(.*?)\n<b>`.

Nesta fase verificou-se que seria necessário organizar a informação num ficheiro json e, dessa forma, seria mais adequado organizá-la num dicionário. Assim sendo, definiram-se os termos como chaves do dicionário e as traduções como valores. De notar que as traduções continham um dicionário cuja chave era o nome da língua e o valor, a respetiva tradução. Assim, obtém-se no fim um ficheiro json da forma:

```
{
  "abcesso": {
    "ingles": "abscess",
    "espanhol": "absceso"
  },
  "abdómen": {
    "ingles": "belly, abdomen",
    "espanhol": "abdomen"
  },
  "abdominal": {
    "ingles": "abdominal",
    "espanhol": "abdominal"
  },
  "abdução": {
    "ingles": "abduction",
    "espanhol": "abducción"
  },
}
```

## 2.2. Trabalho Individual do PDF anatomia geral

Numa fase inicial, realizou-se uma limpeza ao ficheiro xml, com as seguintes etapas e em cada uma delas existe uma expressão regular adequada:

- Eliminaram-se todas as expressões irrelevantes para leitura e perceção do documento.
- Nesta fase, optou-se por não limpar as partes auxiliares (`</?text.*?>`), dado facilita o processo de identificação dos termos.

```
text = re.sub(r"</?page.*>", "", text) # Remover <page> e </page>
text = re.sub(r"</?image.*?>", "", text) # Remover <image> e </image>
text = re.sub(r"</?fontspec.*?>", "", text) # Remover <fontspec> e </fontspec>
text = re.sub(r"<[?]xml(.*)?>", "", text) # Remover <xml> e </xml>
text = re.sub(r"<[!DOCTYPE(.*)?>", "", text) # Remover <DOCTYPE> e </DOCTYPE>
text = re.sub(r"</?pdf(.*)?>", "", text) # Remover <pdf> e </pdf>
text = re.sub(r"<text(.*)>\s*(\s*)(.*)</text>", "", text) # Remover frases inicializadas com *, ** e ***
text = re.sub(r"<text[^>]*><b>[A-Z]</b>\s*</text>\n", "", text) # Remover letras isoladas s/ espaços
text = re.sub(r"<text[^>]*>\s*(\s*<b>)?[ABC](\s*[ABC])*(\s*<b>)?\s*</text>", "", text) # Remover letras e
combinções de letras c/ espaços
text = re.sub(r"A, F", "", text)
text = re.sub(r"C VII", "", text)
text = re.sub(r"Ver pág. 40", "", text)
text = re.sub(r"\s+", "", text) # Remover 1 ou mais "
```

- Dado que nem todos os termos do dicionário são delimitados pela expressão <b></b>, então, para identificação dos mesmos, recorreu-se à substituição direta dos números por um cardinal (#), que antecede sempre um termo.

```
text = re.sub(r'<text.*?>\s*d(\.?)+(\.\d+)?([;,:]? \s*d+(\.\d+)?)*\s*</text>', '#', text) # Substituir
números por marcador #
```

- Seguidamente, delimitaram-se todos os termos pela expressão <b></b>, e eliminaram-se os <b></b> e os # que estavam repetidos, como consequência da delimitação anterior.

```
text = re.sub(r"#\n<text(.*)>(.*)</text>", r"<text\1><b>\2</b></text>", text)
text = re.sub(r"><b></b>", ">", text) # Remover os <b></b> que ficaram a mais
text = re.sub(r"#+(.*)", "", text) # Remover os # que ficaram a mais
```

- Nesta fase da limpeza, procedeu-se à junção das expressões que ficaram separadas pela translineação, e ainda da remoção dos parênteses retos em algumas expressões.

```
text = re.sub(r"-</b></text>\n<text(.*)><b>", "", text) # Juntar as expressões que estão separadas
pelas regras de translineação
text = re.sub(r"\[([.*])\]", r"\1", text)
```

- No seguimento da limpeza, optou-se por alterar o marcador correspondente aos títulos, dado não serem relevantes para o dicionário. Para tal, substituiu-se o marcador <i><b></b></i> pelo marcador <t></t>. Todas as ocorrências de <i></i> também foram substituídas pelo marcador <b></b>. Finalmente, removeu-se o marcador <text></text>.

```
text = re.sub(r"<i><b>(.*)</b></i>([<]*)", r"<t>\1</t>\2", text) # Substituir o marcador b|i
text = re.sub(r"<i>(.*)</i>([<]*)", r"<b>\1</b>\2", text) # Substituir o marcador i
text = re.sub(r"</?text.*?>", "", text) # Remover <text> e </text>
```

- Por fim, removeram-se -\n resultantes da translineação, marcas e, ainda, numeração romana que estaria vinculada a imagens. O último passo consistiu em remover os espaçamentos restantes.

```

text = re.sub(r"-\\n", "", text) # Remover os -\\n que resulta da translineação
text = re.sub(r"\\bm+\\. ", "", text) # Remover marca m. e mm.
text = re.sub(r"<b> </b>", "", text) # Remover marca <b> </b>
text = re.sub(r"<b>\\. </b>", "", text)
text = re.sub(r"<b>(I+)</b>", r"\\1", text)
text = re.sub(r"\\bIII\\b\\n\\bII\\b\\n\\bI\\b\\n\\bI\\b\\n\\bV\\b", "", text) # Remover numeração romana extra
text = re.sub(r"\\n(\\s*)\\n", "\\n", text) # Remover espaços

```

- Finalizada a limpeza, o passo seguinte consiste na combinação dos termos e das designações num dicionário. A pesquisa dos termos consiste na expressão regular: `<b>(.)</b>([<]*)`. De notar que os termos considerados foram ajustados para letra minúscula, para posterior combinação com os restantes dicionários. Assegura-se também a remoção dos pontos finais, tanto na expressão do termo, como na designação.

```

# Pesquisar a expressão
list = re.findall(r"<b>(.)</b>([<]*)", text) # Pesquisar a expressão

# Cria um novo dicionário com as modificações desejadas
dicionario_modificado = {}
for termo, designacao in list:
    chave_modificada = termo.strip().replace('\\n', '').lower() # Remover tudo depois do último '.' na chave
    ultimo_ponto_chave = chave_modificada.rfind('.')
    if ultimo_ponto_chave != -1:
        chave_modificada = chave_modificada[:ultimo_ponto_chave]
    valor_modificado = designacao.strip().replace('\\n', '') # Remover tudo depois do último '.' no valor
    ultimo_ponto_valor = valor_modificado.rfind('.')
    if ultimo_ponto_valor != -1:
        valor_modificado = valor_modificado[:ultimo_ponto_valor]
    dicionario_modificado[chave_modificada] = {"descricao": valor_modificado}

```

- Parte do resultado final do processo anteriormente descrito é o seguinte:

```

"pescoço": {
    "descricao": "Seu limite superior passa por uma linha ao longo da margem inferior da mandíbula, processo mastóide, linha nugal superior, até a protuberância occipital externa; seu limite inferior estende-se da margem superior do manúbrio do esterno, ao longo da clavícula, ao acrômio e à espinha da escápula, até o processo espinhoso de "
},
"tronco": {
    "descricao": ""
},
"tórax": {
    "descricao": "Parte do tronco, entre o pescoço e o abdome. Sua estrutura básica é a caixa torácica. Seu limite inferior é a abertura torácica inferior e o diafragma"
},
"peito": {
    "descricao": ""
},
"abdome": {
    "descricao": "Parte do tronco entre o tórax, a margem superior do sacro, o ligamento inguinal e a sínfi se púbica"
},

```



### 2.3. Trabalho Individual do PDF dicionário de termos médicos e de enfermagem

Numa fase inicial, realizou-se uma limpeza ao ficheiro xml, com as seguintes etapas e em cada uma delas existe uma expressão regular adequada:

- Eliminou-se todo o ficheiro antes do início do dicionário, uma vez que se trata de textos de introdução sobre o dicionário em si. Nesta limpeza usou-se a expressão regular: <b>A, AN</b>.
- Importante limpar em xml todas as partes auxiliares (</?text.\*?>), como na figura abaixo, mantendo a informação essencial do texto apenas: para exame.

```
<text top="276" left="136" width="61" height="12" font="25">para exam.</text>
```

- De seguida procedeu-se com várias expressões regulares todos os cabeçalhos e rodapés do ficheiro, uma vez que estas seções do ficheiro são bastante diferentes. As expressões regulares usadas foram “\n[\s\*\o\s\*]+” , “[ÁÃÄÅËÊÏÎÍÔÕÖÜÇÑÀ-Z]{3}\s?\n[ÁÃÄÅËÊÏÎÍÓÔÕÖÜÇÑÀ-Z]{3}\nSou.\*\n(.\*\n){1,5}\d{2,3}\n”, “Sou.\*\n(.\*\n){1,5}\d{2,3}\n” e “Sou En.\*\n(.\*\n){2}”. Esta seção limpa excertos como:

Sou Enfermagem - Cadastre-se grátis <a href="https://souenfermagem.com.br">em: https://souenfermagem.com.br</a>

</page>

<page number="16" position="absolute" top="0" left="0" height="785" width="573">

<fontspec id="20" size="12" family="KLHFJ0+GillSans" color="#231f20"/>

ABR  
ABS  
Sou Enfermagem - Cadastre-se grátis <a href="https://souenfermagem.com.br">em: https://souenferma  
</page>  
<page number="18" position="absolute" top="0" left="0" height="785" width="573">  
20  
bactérias que invadem a pele, e são

[illegible]

- Nesta fase da limpeza tratou-se de limpar as descrições, nomeadamente, os hífens quando há parágrafos, com a expressão -\n.
- Ainda no texto das descrições verificou-se que existiam partes com <i>, por exemplo no caso da figura abaixo, deste modo desenvolveu-se esta expressão -</i>\n<i>(.)</i>, e substitui-se por \1</i>. Ou seja, foram realizadas estas expressões regulares, de forma a limpar os <i> e os hífens entre <i>'s, mantendo apenas a palavra lá dentro. Assim também é capaz de juntar os 2 <i>'s seguidos, como Levulose, o exemplo da imagem. O último passo, relativamente, aos <i>'s foi manter apenas as palavras no seu interior com a expressão regular \n<i>(.)</i>\n, substituindo pelo primeiro grupo de captura.

```
<b>AÇÚCAR DE CARVÃO DE PEDRA</b>
V.
<i>Sacarina</i>
.
<b>AÇÚCAR DE FRUTA</b>
V.
<i>Levu</i>
<i>lose</i>
.
<b>AÇÚCAR DE LEITE</b>
V.
<i>Lactose</i>
.
<b>AÇÚCAR DE MEL</b>
V.
<i>Glicose</i>
.
<b>AÇÚCAR MINERAL</b>
V.
<i>Sacarina</i>
.
```

- O ficheiro xml nesta fase ainda continha a capa da página inicial da letra XX, por isso, desenvolveu-se a expressão regular: ([A-Z])\1<b>, substituindo por apenas <b>. De notar que na parte final do ficheiro ainda existiam artefactos por limpar, por isso utilizou-se a expressão "<item.\*\n(.\*)\*\n\*</pdf2xml>" de forma a eliminar:

```
VÔM
VULXX<b>WIDAL, REACÃO DE</b>
-
Reação de
aglutinação para diagnóstico
<b>XANTELASMA</b>
```

```
<item page="1">Capa</item>
<item page="2">Expediente</item>
<item page="3">Introdução</item>
<item page="15">A</item>
<item page="69">B</item>
<item page="85">C</item>
<item page="139">D</item>
<item page="171">E</item>
<item page="206">F</item>
<item page="224">G</item>
<item page="239">H</item>
<item page="260">I</item>
<item page="275">J K L</item>
<item page="291">M</item>
<item page="313">N O</item>
<item page="335">P</item>
<item page="378">Q R</item>
<item page="397">S</item>
<item page="417">T</item>
<item page="441">U V Z</item>
</outline>
</pdf2xml>
```

- Nesta fase verificou-se que o ficheiro possuía muitos artefactos relacionados com os bolds, tanto após cada termo, como a fazer a transcrição das palavras pelo meio dos termos, assim como no meio das descrições, por isso aplicaram-se várias expressões regulares:

- `-</b>\n<b>` para limpar hífen dentro dos termos;
- `</b>\n\s?<b>` para limpar os bolds seguidos nas várias frases, de forma a ficar apenas `<b>` todo o texto de várias linhas `</b>` numa só linha, exemplo da imagem à direita;
- `<b>\s</b>` para limpar bolds com um espaço, exemplo na imagem central;
- `<b>([A-Z][^ÁÉÍÓÚÁÂÊÎÔÛÃÕÇÀ-Z]*?)</b>` responsável por substituir o que capta (s) pelo seu grupo de captura, o exemplo único na imagem da esquerda;
- `<b>(\d.*?)</b>` responsável por substituir o que capta (tudo, incluindo os números) pelo seu grupo de captura, o exemplo na imagem à direita fila 2 ;
- `<b>([A-Z][^ÁÉÍÓÚÁÂÊÎÔÛÃÕÇÀ-Z]*?)</b>` responsável por capturar o caso da imagem central fila 2;
- `<b>(.*)</b>[\n\s][^~]` desenvolvido para resolver a situação dos bolds que não verificavam a seguir um hífen, pelo que a substituição é feita para a expressão: `<b>\1</b>` - . Assim todos os bolds possuem a seguir possuem espaço hífen;
- Agora com a expressão: `</b>[\n\s]?-` é possível captar todos os bolds, já que neste momento todos se encontram com hífen, de modo a eliminá-lo em todos os casos;
- A expressão `[^~\n]<b>` vai a seguir resolver o problema de existirem bolds como na imagem à esquerda fila 2 que antes não possuem um `\n`. Assim, substitui-se o texto captado pela expressão referida por `\n<b>`;

```
<b>fi</b>
<b> cam freqüentemente surpresos </b>
<b>ao saber que seus testes de fígado dão anorma
<b>de rotina.</b>
```

```
<b> , </b>
```

```
<b>s</b>
êmen
```

```
<b>1 Visão normal</b>
- A luz de um objeto externo é trazida
para um foco na retina sensível do
fundo do globo ocular.
<b>2 Hipermetropia </b>
```

```
<b> , </b>
```

```
<b>ACRE</b>
V. Ácido.<b>ACRO</b>
```

- Por último, existia uma situação não uniformizada nas descrições, nomeadamente, a última descrição do ficheiro. Para resolver esta situação procedeu-se à substituição do que a expressão regular: `Vertigens\\.\\)` capta, por: `Vertigens\\.\\n<b>`. Desta forma a captura dos termos e descrições está facilitada.

Após a limpeza do ficheiro xml, procurou-se extrair os termos, identificados por estarem sempre rodeados por <b>TERMO</b>. Assim como as descrições que se encontram entre </b>DESCRICAO<b>, por isso desenvolveu-se a expressão: <b>(.\*?)</b>\s\*(.\*?)\s\*(?=<b>|\$, com a opção re.findall. Nesta opção adicionou-se a flag re.DOTALL, cujo objetivo é colocar o ponto (‘.’) a corresponder a quebras de linhas(‘\n’).

Nesta fase verificou-se que seria necessário organizar a informação num ficheiro json e, dessa forma, seria mais adequado organizá-la num dicionário, com termos como key do dicionário e com as descrições como values do mesmo. De notar apenas que aqui verificou-se essencial realizar o strip e a função limpa do value e a substituição da vírgula por vazio. A função limpa apenas é responsável por ‘\s+’ por um único espaço. Assim, obtém-se no fim um ficheiro json da forma:

```
{
  "ABDOMINAL": {
    "designacao": "Que se refere ou diz respeito ao abdome."
  },
  "ABDOMINO-HISTERECTOMIA": {
    "designacao": "Extirpação do útero através do abdome."
  },
  "ABDUÇÃO": {
    "designacao": "Movimento de afastamento de um membro ou de um segmento do eixo do corpo."
  },
  "ABDUTOR": {
    "designacao": "Músculo que ao contrair-se afasta do eixo do corpo alguma parte do organismo."
  },
  "ABERRAÇÃO": {
    "designacao": "Desvio do normal. Genética Anomalia na situação ou na conformação de um órgão."
  },
  "ABERRAÇÕES CROMOSSÓMICAS": {
    "designacao": "Alteração na anatomia dos cromossomos normais que geralmente afetam a função."
  },
  "ABERRANTE": {
    "designacao": "Que se desvia do normal, do padrão comum. Ex.: artéria aberrante, veia aberrante."
  },
  "ABLAÇÃO": {
    "designacao": "Separação por incisão ou amputação cirúrgica de qualquer parte do corpo, por exemplo."
  },
  "ABLEPSIA": {
    "designacao": "Cegueira, perda ou falta de visão."
  },
  "ABLUÇÃO": {
    "designacao": "Banho, lavagem. Ato de lavar-se, banhar-se."
  },
  "ABORTAMENTO": {
    "designacao": "Expulsão do feto antes de 180 dias de gestação. Depois desse prazo, chama-se parto."
  }
}
```

## 2.4. Resultado – combinação dos 3 jsons

- Para combinação dos três documentos json, realizou-se a abertura e a leitura dos ficheiros, e extraiu-se a chave de cada um deles.

```
# Lendo os arquivos JSON
with open('Output/dicionario_termos_medicos_pt_es_en.json', 'r') as f:
    json1 = json.load(f)

with open('Output/anatomiageral.json', 'r') as f:
    json2 = json.load(f)

with open('Output/Dicionario_de_termos_medicos_e_de_enfermagem.json', 'r') as f:
    json3 = json.load(f)

resultado = {}

termos_json1 = set(json1.keys())
termos_json2 = set(json2.keys())
termos_json3 = set(json3.keys())
```

- Numa fase inicial, extraem-se as chaves dos ficheiros 1 e 2 (interseção) e, ainda, as chaves comuns dos dois ficheiros (diferença).

```
# Chaves em comum em json1 e json2
termos_comuns = termos_json1.intersection(termos_json2)

# Chaves em json1 mas não em json2
termos_1 = termos_json1.difference(termos_json2)

# Chaves em json2 mas não em json1
termos_2 = termos_json2.difference(termos_json1)
```

- Seguidamente, inicializou-se a combinação e, para tal, verificou-se se o termo está aos termos comuns, ou se está apenas nos termos do ficheiro 1 ou 2.
- Para a comparação com o terceiro documento, inicializa-se pela verificação da existência do termo nos dois ficheiros combinados. De seguida, procede-se da mesma forma para o ficheiro 1 e, posteriormente, para o ficheiro 2.

```

for termo in termos_comuns:
    resultado[termo] = {
        "descricao": json2[termo]["descricao"],
        "ingles": json1[termo]["ingles"],
        "espanhol": json1[termo]["espanhol"]
    }

for termo in termos_1:
    resultado[termo] = {
        "ingles": json1[termo]["ingles"],
        "espanhol": json1[termo]["espanhol"]
    }

for termo in termos_2:
    resultado[termo] = {
        "descricao": json2[termo]["descricao"],
    }

for termo in termos_json3:
    info = {"designacao": json3[termo]["designacao"]}

    if termo in termos_comuns:
        info["ingles"] = json1[termo]["ingles"]
        info["espanhol"] = json1[termo]["espanhol"]
        info["descricao"] = json2[termo]["descricao"]
    elif termo in termos_1:
        info["ingles"] = json1[termo]["ingles"]
        info["espanhol"] = json1[termo]["espanhol"]
    elif termo in termos_2:
        info["descricao"] = json2[termo]["descricao"]

    resultado[termo] = info

```

### 3. Conclusão

Com base nas capacidades de processamento de linguagem natural adquiridas neste projeto, foi possível extrair informações úteis de documentos biomédicos em PDF e armazená-las num formato estruturado. Esta capacidade pode ser aplicada em diversas áreas da Engenharia Biomédica, incluindo análise de dados clínicos, descoberta de padrões de doenças e diagnóstico assistido por computador. Além disso, o projeto incentiva a exploração de técnicas de processamento de linguagem natural mais avançadas para lidar com documentos mais complexos e permitir a extração de informações ainda mais precisas.

Em resumo, o projeto prepara para enfrentar desafios no campo da saúde, melhorando a sua capacidade de analisar dados e informações para tomar decisões importantes e salvar vidas.