A blue and white symbols

Description automatically generated with medium confidence

**ANÁLISE E VISUALIZAÇÃO DE DADOS**

RELATÓRIO DE CONCLUSÃO DA DISCIPLINA

ANÁLISE E VISUALIZAÇÃO DE DADOS DAS ENTREVISTAS DO MUSEU DA PESSOA

DOCENTES: JOSÉ JOÃO E ÁLVARO IRIARTE

ALUNOS: DANIELLE PINTO ASSIS PG 51915

MAURICIO JOSE CASTRO BARBOSA PG52764

**SUMÁRIO**

INTRODUÇÃO 3

OBJETO 4

OBJETIVO 4

METODOLOGIA 4

CORPO DO TRABALHO 9

VISUALIZAÇÃO 21

DESTAQUES 24

CONCLUSÃO 25

BIBLIOGRAFIA 26

**INTRODUÇÃO**

O presente relatório trata-se do trabalho de conclusão da disciplina Análise e Visualização de Dados do curso de Mestrado em Humanidades Digitais da Universidade do Minho.

Os alunos foram convidados a realizar um trabalho de análise e visualização de dados em documentos caracterizados como entrevistas de sítios da Universidade do Minho, quais sejam: NOSUM, ComUM, UMDicas, acervo específico do Museu da Pessoa, objeto deste estudo, bem como realizar entrevistas recentes.

O Museu da Pessoa nasceu no Brasil, São Paulo, em 1991, criado por um grupo de historiadores que decidiu construir a história do país a partir de depoimentos de pessoas comuns, cujo objetivo é reunir depoimentos de cada ser humano, famoso ou anônimo, para perpetuar sua história, registrar histórias de famílias, comunidades ou instituições. Sua “coleção de arte” é composta por coisas intangíveis ou imateriais.

O Núcleo Português do Museu da Pessoa é uma iniciativa que teve início na Universidade do Minho em 1999, em estreita colaboração com o Museu da Pessoa do Brasil, com dois objectivos principais: por um lado concretizar e apoiar, no terreno e na Internet, projetos de coleções de histórias de vida. Por outro lado, desenvolver processos e programas que visem explorar em profundidade e de forma automatizada, os conteúdos das histórias de vida, através de técnicas informáticas relacionadas com a anotação documental.

Desta forma, pretende-se lançar em Portugal diversas iniciativas informáticas relacionadas com a anotação de documentos e lançar diversas iniciativas que permitam dar uma visão renovada de determinadas comunidades, instituições ou empresas, ou mesmo de famílias e cidadãos.

Há um fluxo de trabalho adotado pelos técnicos do Museu da Pessoa para aquisição de histórias de vida de pessoas comuns que consiste em: gravar o relato de um participante (áudio ou vídeo) feito por um entrevistador; transcrever as entrevistas; anotar as transcrições em Extensible Markup Language, marcando eventos, histórias independentes, etc. Usar entrevistas XML para produzir vários resultados.

**OBJETO**

Os objetos analisados neste relatório são 09 (nove) entrevistas realizadas nos anos 2000 a 2002 e 02 (duas) entrevistas recentes realizadas pelos alunos do curso de Mestrado em Humanidades Digitais com pessoas que tiveram ou têm alguma vivência na Universidade do Minho.

**OBJETIVO**

O projeto proposto pelos docentes considera a data alusiva aos 50 anos da Universidade que se realiza no corrente ano. Neste sentido, o objetivo é conhecer e formar competências que ajudem a representar, analisar e contextualizar dados, transformando-os em informação compreensível sobre a qual se possam fazer novas perguntas a partir do material analisado. Fizemos os exercícios de extração, tratamento, análise e visualização de dados lexicais e linguísticos por meio de Processamento de Linguagem Natural.

**METODOLOGIA**

MUSEU DA PESSOA

No caso do acervo específico do Museu da Pessoa, a metodologia aplicada iniciou com descarregamento e separação das pastas das entrevistas para leitura e análise. A seguir, escolhemos os arquivos a serem tratados: “bi.xml” e “entrevistaeditado.xml”. Após a leitura e análise dos conteúdos das entrevistas por todos os grupos em seus respectivos sítios, decidimos os metadados que seriam extraídos a comporem o corpus e fizemos os registros no repositório AVD-MHD /Arquivo\_UMSombra / Museu da Pessoa no github.

Em relação ao Museu da Pessoa convencionou-se os seguinte metadados:

| MARCAÇÃO | DEFINIÇÃO |
| --- | --- |
| Entrevistado | Pessoa a ser entrevistada |
| Entrevistador | Pessoa que concedeu a entrevista |
| Trabalho | Ocupação da pessoa que foi entrevistada |
| Local de nascimento | Local de nascimento da pessoa entrevistada |
| Data de nascimento | Dados de nascimento da pessoa entrevistada |
| Data | Dados da entrevista |
| Notas | Notas sobre uma entrevista |

Aplicamos o código abaixo para primeira extração dos metadados e formatação dos arquivos das entrevistas em extensão markdown:

A screenshot of a computer

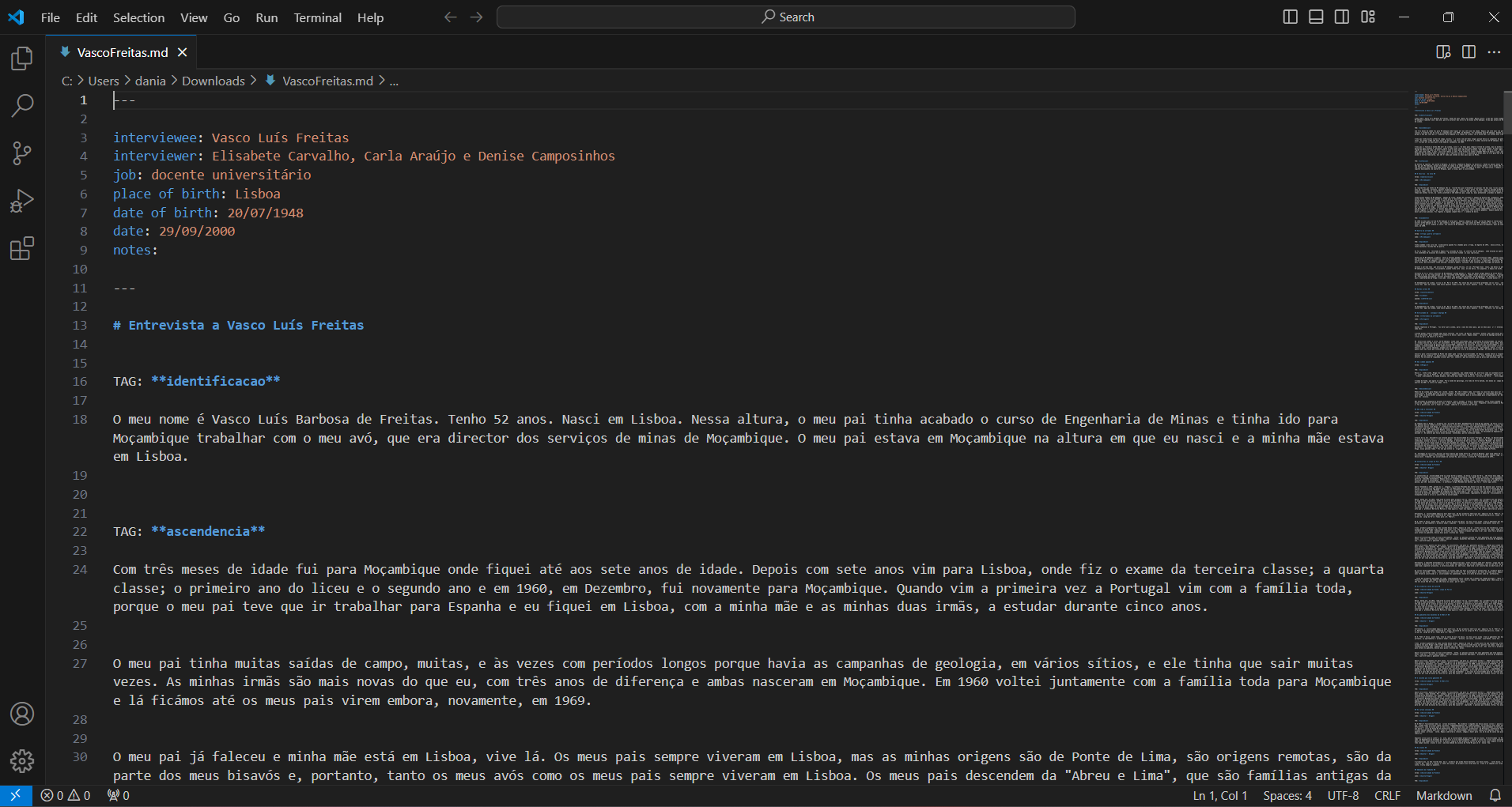
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



* Objetivo: processar os arquivos XML fornecidos, extrair os metadados e informações de entrevistas, e escrever os resultados em um arquivo Markdown, de maneira estruturada para visualizar e analisar os dados extraídos. O nosso objetivo era que cada ficheiro markdown começasse pelo título “Entrevista a (nome do entrevistado” e que o corpo da entrevista fosse dividido pelas secções marcadas pelas tags, de forma a não se perder também esse tipo de informação, valorizando mais o atributo “título”, que, quando existia, servia de título para determinada secção.

**Dificuldades:**

Numa fase inicial, deparámo-nos com alguns problemas quanto à extração dos metadados. Num código mais prematuro e em fase mais experimental, os resultados que obtínhamos contavam com algumas falhas, como por exemplo, um espaço antes de cada linha do cabeçalho, os metadados não estavam uniformizados entre si e os traços que marcavam o início e o final do cabeçalho também nem sempre batiam certo. Para além disso, decidimos que sempre que algum metadado estivesse vazio, imprimisse uma string vazia.

Para a marcação do começo de cada entrevista, e uma vez que o formato do ficheiro final é markdown, optámos por colocar um “#” antes do título “Entrevista a (nome do entrevistado)” de forma a também uniformizá-lo.

Quanto à extração do corpo das entrevistas, houve também algumas dificuldades que tiveram que ser ultrapassadas. O objetivo era, não só extrair os parágrafos das entrevistas, mas também as tags. Desta forma, uma vez que as tags variavam de entrevista para entrevista, optámos por colocar a string “TAG” antes de cada uma, e, sempre que havia um atributo, imprimi-lo por baixo da mesma. O atributo “título” pareceu-nos ter mais relevância do que qualquer outra tag ou atributo, pelo que decidimos que o mesmo deveria destacar-se dos restantes. Assim, sempre que houvesse um atributo “título” dentro de uma tag, o mesmo era imprimido entre “##”, de forma a ficar destacado como um sub-título.

A maior dificuldade centrou-se na extração dos parágrafos que se encontravam dentro de cada tag. Inicialmente, ao extrair as tags e os parágrafos que continham, uma vez que todas as tags se encontravam dentro de uma tag <mp>, era imprimido todo o corpo da entrevista, sem tags, para além do corpo das entrevistas separado por tags. Desta forma, tivemos que especificar que queríamos extrair todas as tags, exceto a tag <mp>, conseguindo, assim, o resultado final esperado.

Explicação do código:

* Uso da biblioteca beautiful soup usada para analisar arquivos XML e extrair informações específicas e salvar em arquivo de texto markdown. E **Glob** para obter os caminhos dos arquivos XML.
* Criação da função para extrair metadados: analisar o XML, extrair os metadados (como depoente, entrevistador, profissão, data de nascimento e notas) e escrever no arquivo de saída.
* Criação da função para encontrar depoente: encontrar e escrever o nome do depoente no arquivo de saída.
* Criação da função para extrair a entrevista: analisar o XML, extrair títulos, atributos e conteúdo dos parágrafos e escrever no arquivo de saída.
* Iterar sobre os arquivos “bi.xml” e “editado.xml”: percorrer e para cada arquivo bi.xml, abrir, ler, e chamar as funções extrair\_metadados e encontrar\_depoente. E fechar o arquivo markdown após todas as operações de escrita estarem completas.

ENTREVISTAS

Além dos projetos específicos por grupo, cada aluno ficou encarregado de realizar uma entrevista com pessoas que tiveram ou têm algum relacionamento ou vivência na Universidade do Minho a fim de recolher histórias e experiências alusivas ao tema de 50 anos da universidade que se realiza em 2024.

Os membros da equipa realizaram duas entrevistas com ex-aluna e professora da Universidade. Os produtos referentes à atividade foram apresentados em aula recente e salvos em pasta específica no repositório AVD-MHD /Arquivo\_UMSombra / EntrevistasAVD. As entrevistas foram realizadas online e gravadas. Posteriormente, realizamos a transcrição utilizando a ferramenta Pinpoint (Google) e tratamos o conteúdo de forma estruturada em formato Markdown para também compor o corpus do projeto.

A experiência com esta ferramenta não foi igual para os dois. Se, num caso, com uma entrevista mais curta, a experiência foi mais positiva do que negativa, no outro, com uma entrevista mais longa e detalhada, a experiência foi mais negativa do que positiva. A pontuação raramente estava correta, muitas palavras não coincidiam com aquilo que foi dito e todas as pausas verbais foram contabilizadas. Ainda assim, numa entrevista de tão longa duração, foi preferível ir corrigindo os erros que a ferramenta foi dando, e aproveitar os acertos, do que transcrever do zero.

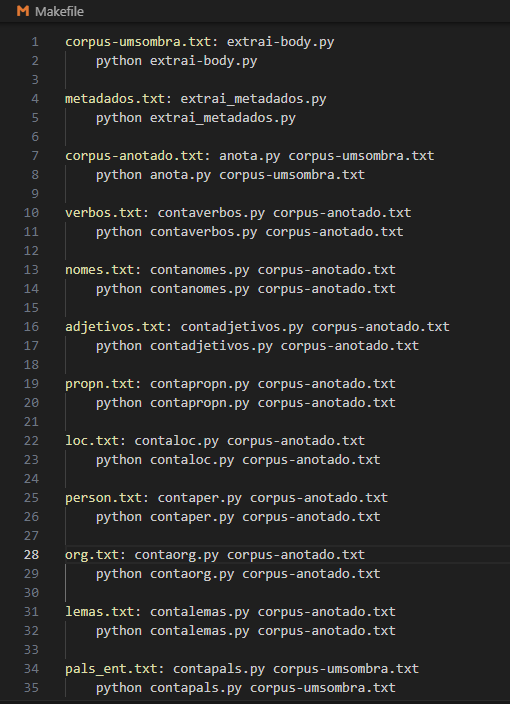
Em relação às Entrevistas podem-se incluir os seguinte metadados:

| MARCAÇÃO | DEFINIÇÃO |
| --- | --- |
| Trabalho | Ocupação da pessoa que foi entrevistada |
| Local de nascimento | Local de nascimento da pessoa que foi entrevistada |
| Data de nascimento | Dados de nascimento da pessoa entrevistada (por volta) |
| Data entrevista | Dados da entrevista |
| notas | Notas sobre uma entrevista |

**CORPO DO TRABALHO**

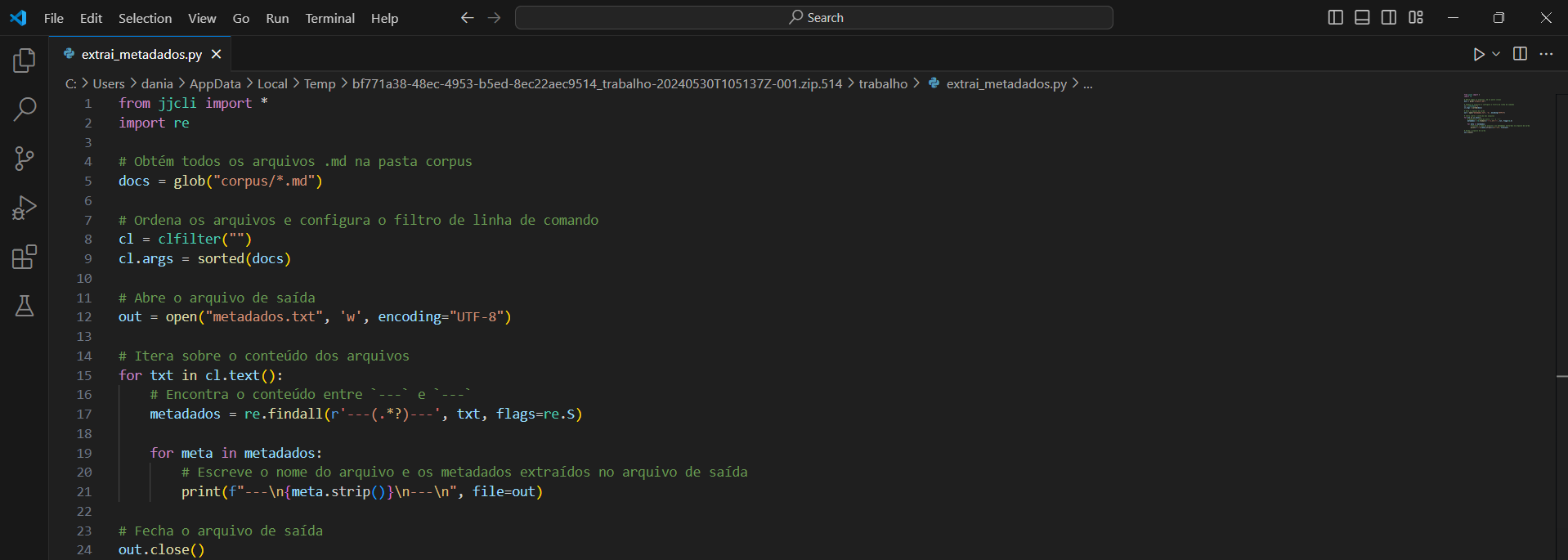
Para desenvolver a competência de analisar e visualizar dados, fomos sugeridos a realizar as extrações de metadados e conteúdo das entrevistas para composição de um corpus e posteriores contagens, segregações e análises a serem produzidas em formato .xlsx ou .csv como um dataset. Após definição e padronização das entrevistas no formato markdown, iniciamos o processo para criação do corpus para posterior extrações e visualizações.

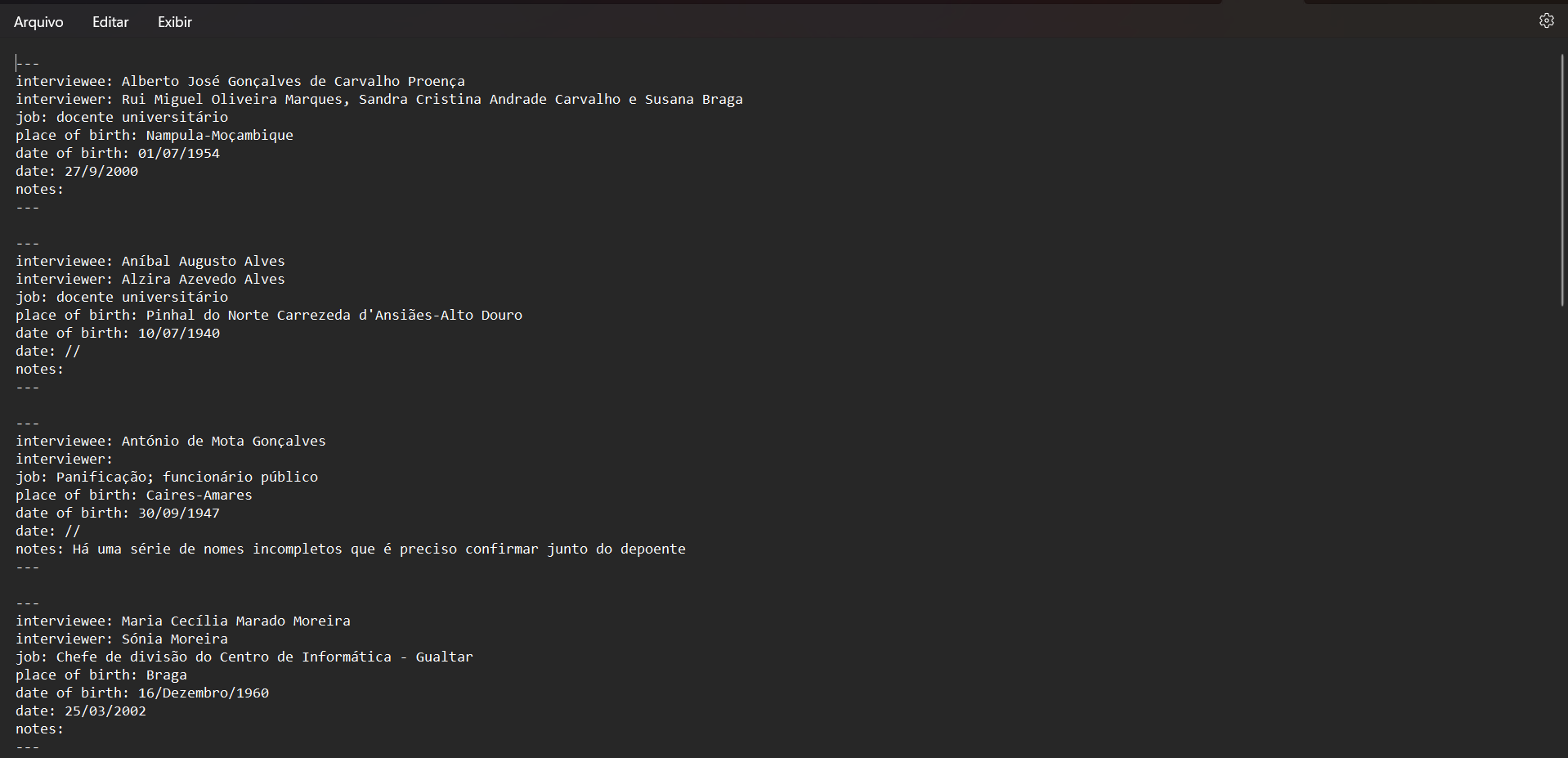
Utilizámos um ficheiro Makefile de forma a facilitar e automatizar os processos. Constatámos que é uma ferramenta de extrema utilidade, uma vez que simplifica todos os processos. Todas as extrações, exceto a contagem do número de palavras totais, foram feitas com recurso à Makefile.



**Criação do corpus:** Salvamos em uma pasta chamada Corpus todos os arquivos.md com as 11 (onze) entrevistas que seriam analisadas.

**Extração dos metadados:** Aplicamos o código abaixo para extração única de todos os metadados das entrevistas do corpus.



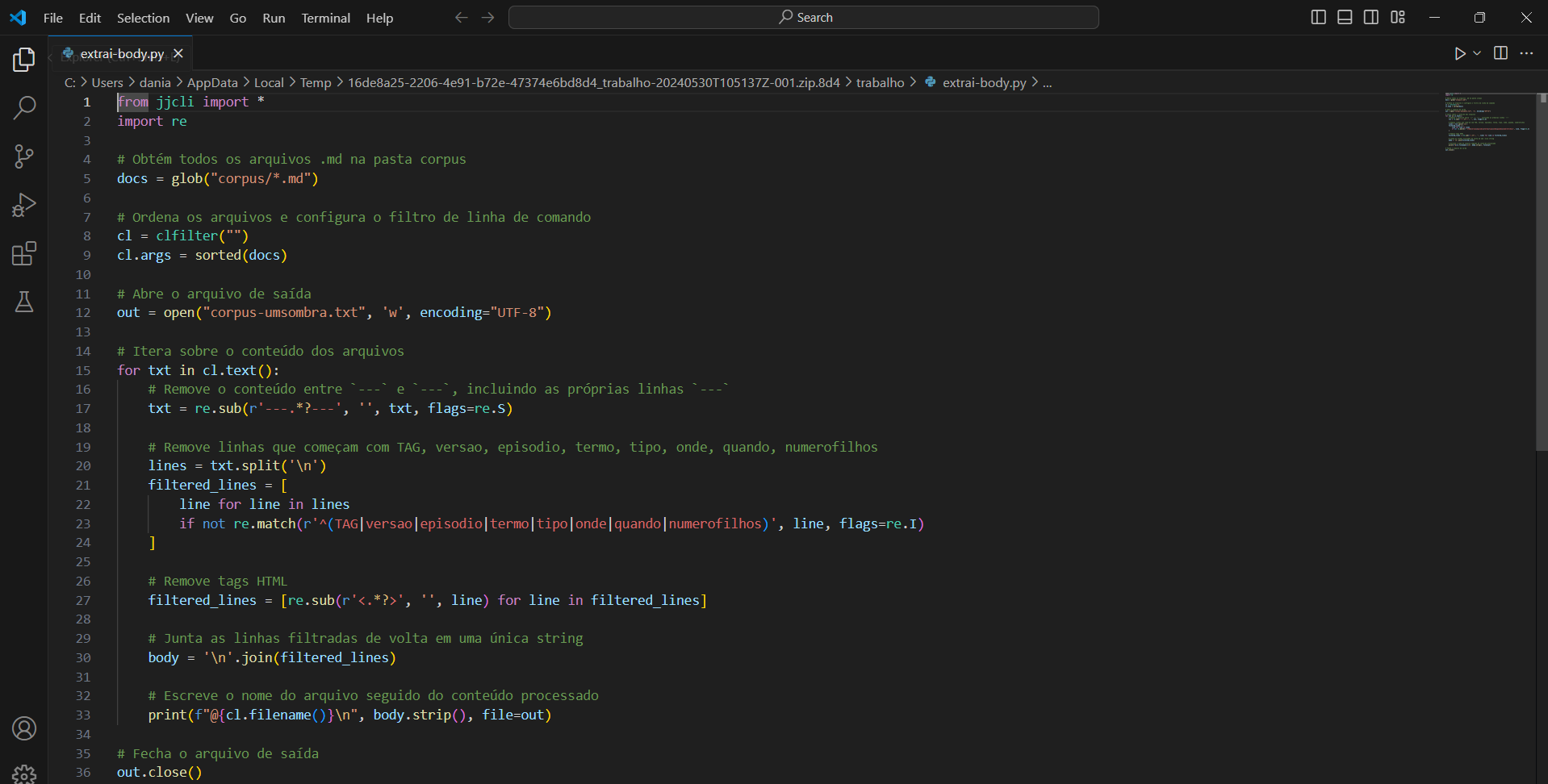


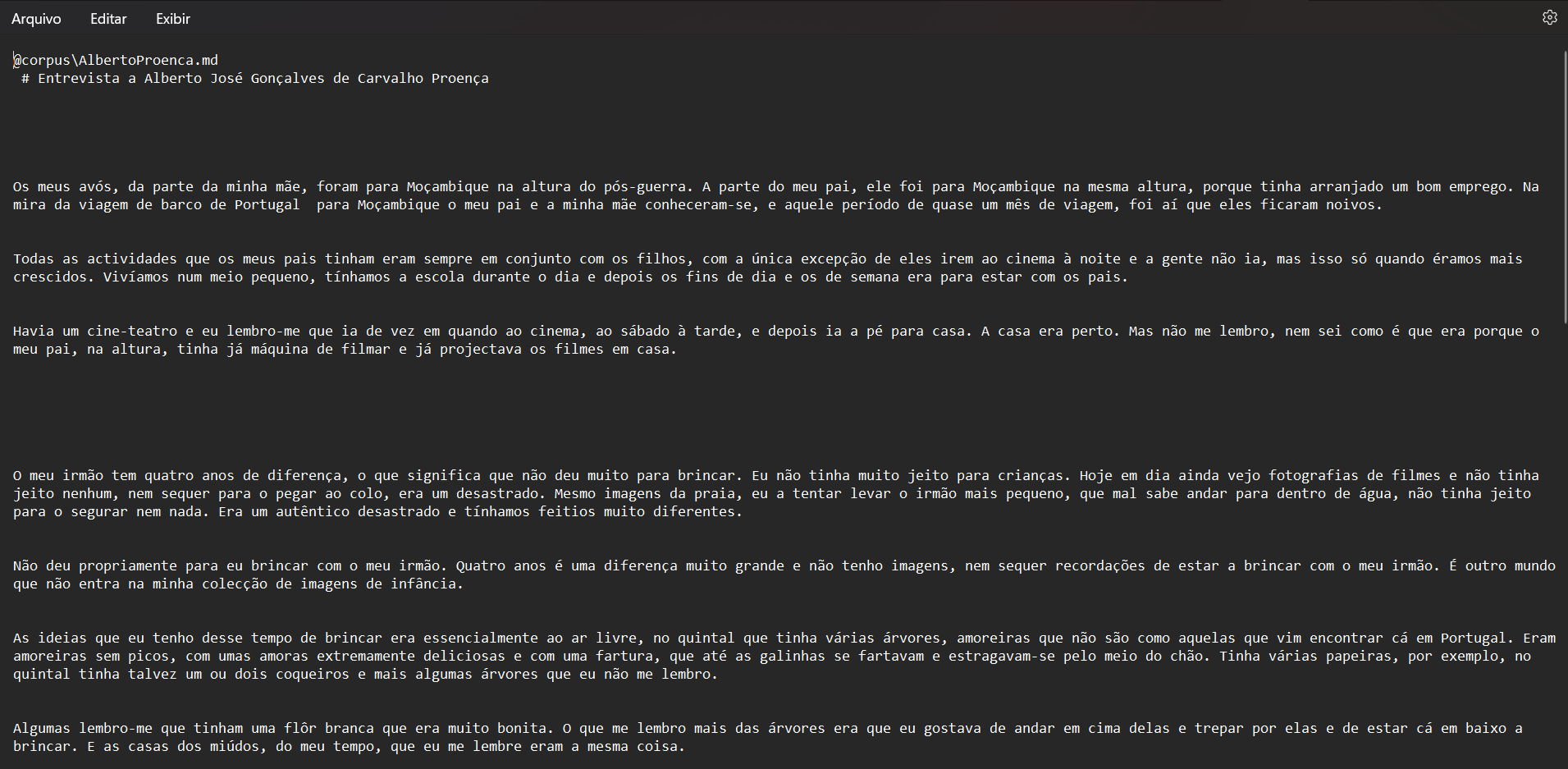
Objetivo: Coletar todos os arquivos Markdown (.md) na pasta corpus para extrair os metadados das entrevistas e salvar em arquivo específico “metadados.txt”, extraindo o conteúdo entre “--- e —” de cada arquivo.

Explicação do código:

* Importação do módulo “re”: manipulação e busca de padrões de texto.
* Uso de “clfilter("")”: para criar um filtro de linha de comando vazio.
* Uso de “cl.args = sorted(docs)”: Define os argumentos do filtro como a lista ordenada de arquivos markdown.
* Uso da expressão regular re findall(r'---(.\*?)---': para encontrar todo conteúdo que está entre “--- e —” no “texto (txt)”.

**Criação do corpus-umsombra:** Aplicamos o código abaixo para compilação apenas do conteúdo de todas as entrevistas do corpus.



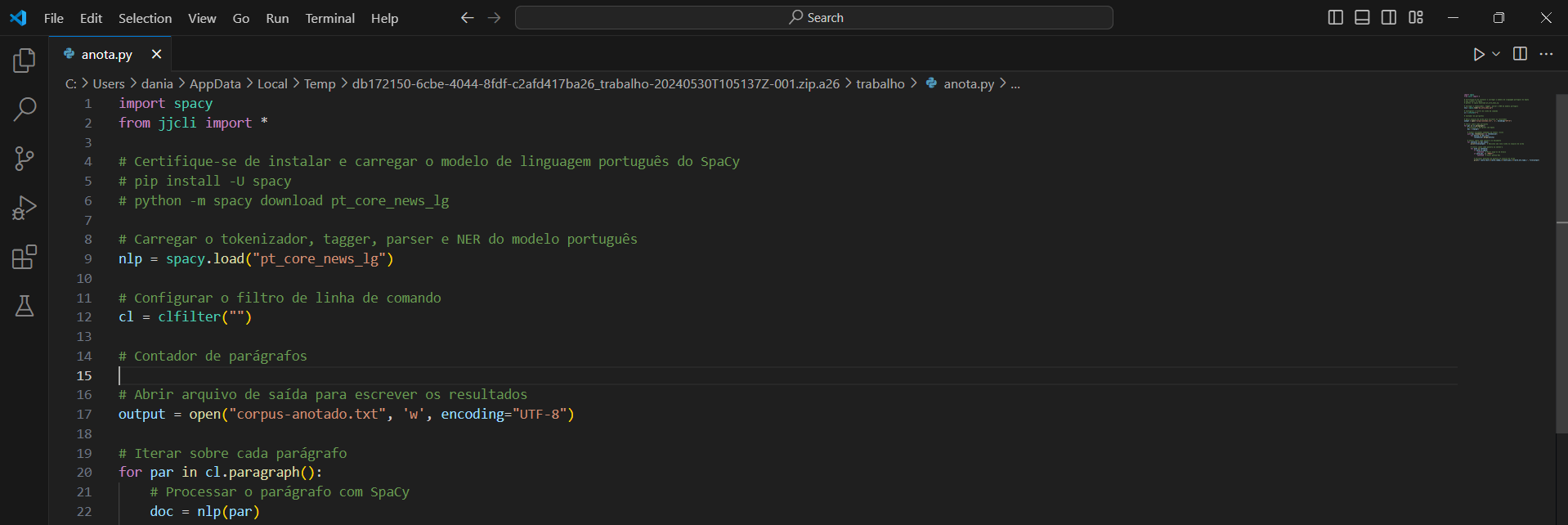


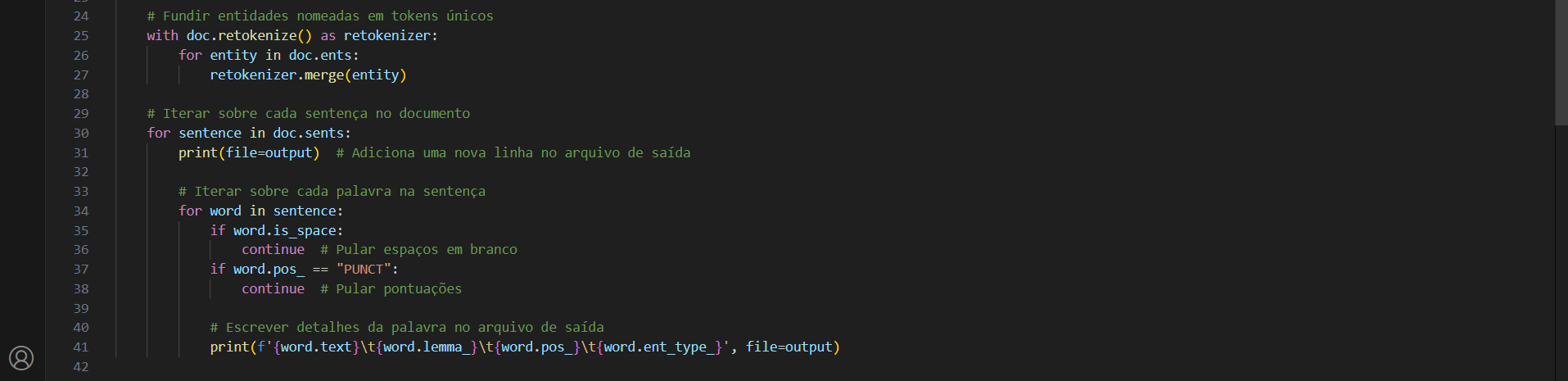
Objetivo: extrair apenas os conteúdos das entrevistas com a identificação de cada corpus, retirando espaços, algumas palavras-chave de linhas específicas, tags html e salvar num arquivo específico “corpus-umsombra.txt”.

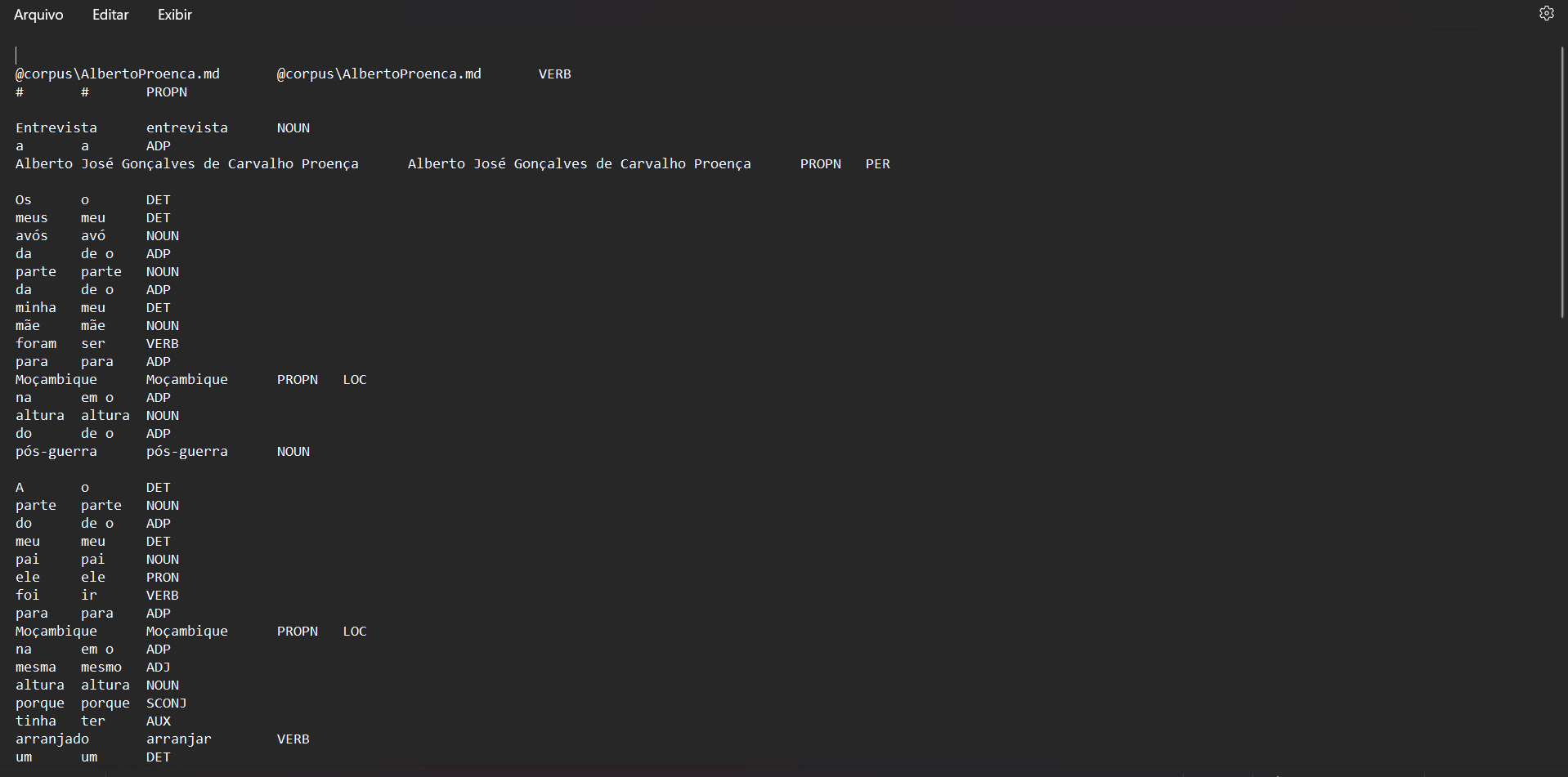
Explicação do código:

* Uso de “re.sub(r'---.\*?---', '', txt, flags=re.S)”: expressão regular para substituir o conteúdo entre “--- e ---", incluindo as próprias linhas “---”, por uma string vazia. A flag re.S permite que “.” corresponda a caracteres de nova linha, permitindo a busca em múltiplas linhas.
* Divisão do texto em linhas e remoção de linhas específicas: criação de uma lista de linhas que não começam com TAG, versao, episodio, termo, tipo, onde, quando, ou numerofilhos, ignorando a diferenciação entre maiúsculas e minúsculas (flags=re.I).
* Uso de “cl.filename()”: para obter o nome do arquivo atual.
* Uso de “body.strip()”: para remover espaços em branco no início e no final do texto.

**Criação do corpus-anotado:** Aplicamos o código abaixo para análise linguística e extração das palavras, seus lemas, suas partes do discurso e os tipos de entidades nomeadas existentes nas entrevistas:







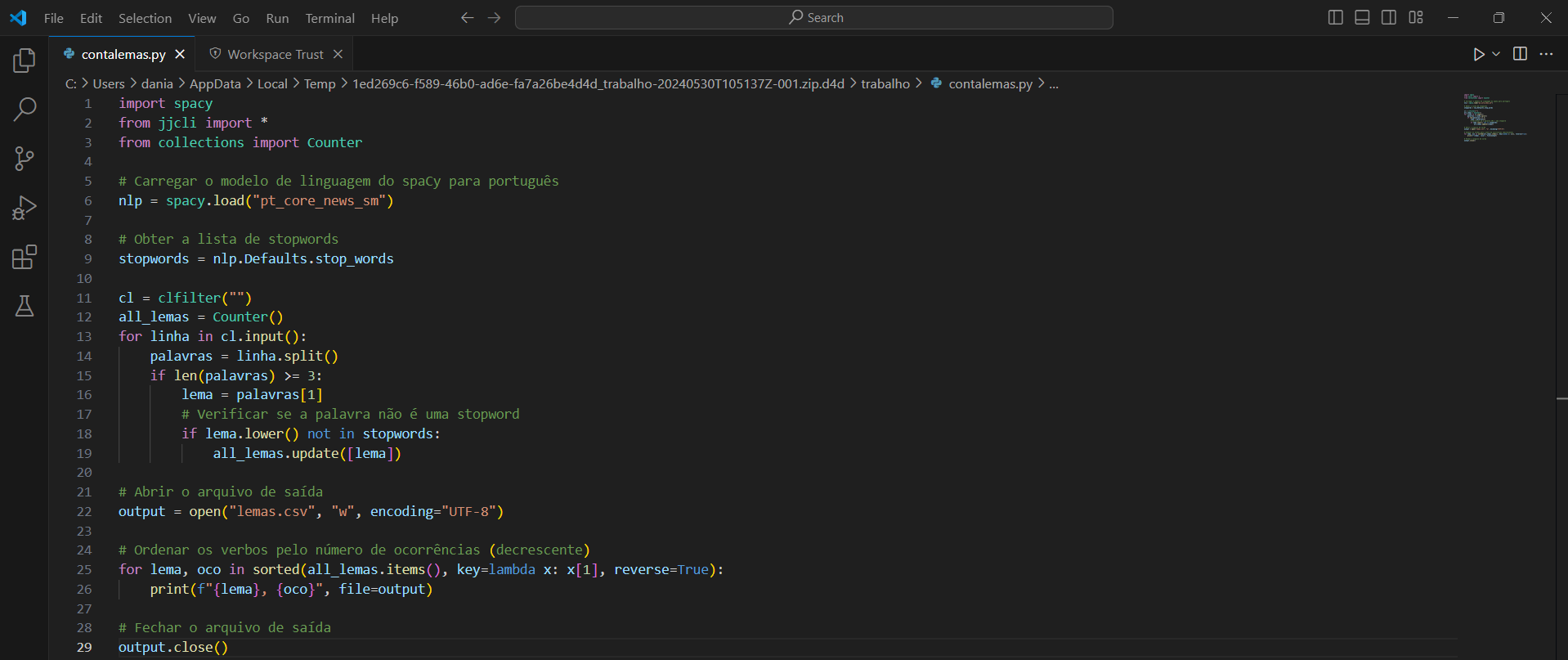
Objetivo: Realizar o processamento dos parágrafos de texto em português usando a biblioteca Spacy para análise linguística, retirar espaços, pontuações e gravar os resultados em um arquivo de saída “corpus-anotado”. Além disso, gravar o texto das palavras, seus lemas, suas partes do discurso e os tipos de entidades nomeadas.

Explicação do código:

* Uso da biblioteca spacy: para processamento de linguagem natural.
* Uso de “doc = nlp(par)”: processar cada parágrafo com o modelo spacy, gerando um objeto Doc que contém a análise linguística do parágrafo.
* Uso de “retokenizer.merge(entity)”: para fundir entidades nomeadas (como nomes de pessoas, lugares, etc.) em tokens únicos, facilitando o tratamento dessas entidades como unidades individuais.
* Ao final escreve os detalhes da palavra no arquivo de saída: word.text: texto da palavra; word.lemma\_: lema da palavra\_; word.pos\_: parte do discurso e word.ent\_type\_: tipo de entidade nomeada.

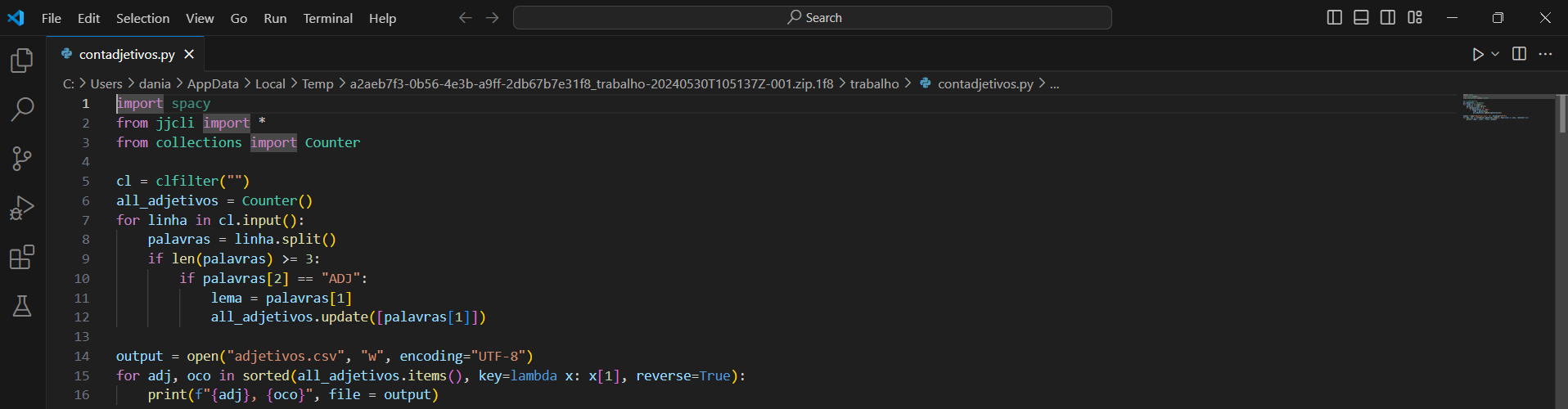
**Extrações, contagem e visualização:** A partir da criação do corpus, iniciamos as contagens de adjetivos, lemas, locais, nomes, organizações, palavras, pessoas, substantivos e verbos. Todos os arquivos serão compartilhados na pasta específica do projeto no github. Aplicamos o código abaixo para realizar as contagens, alterando especificamente para cada categoria de contagem.

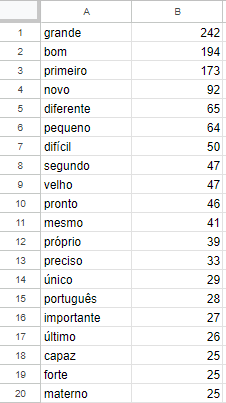
Lemas



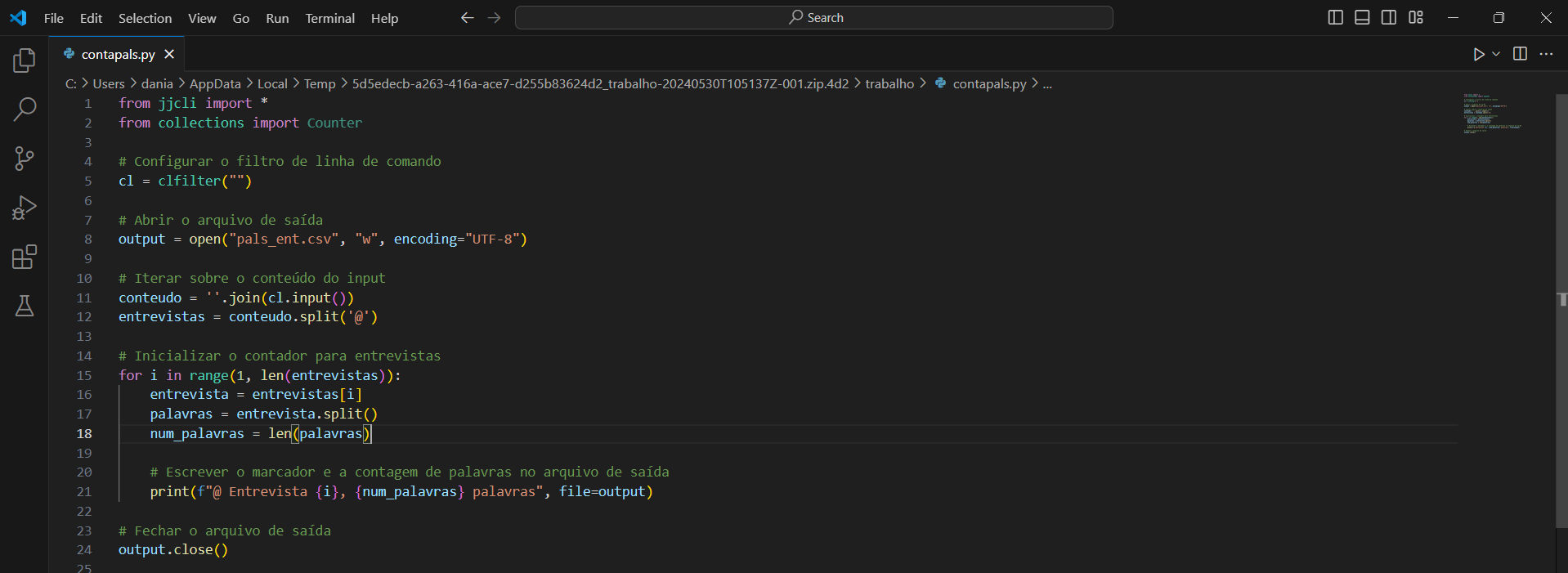


Adjetivos



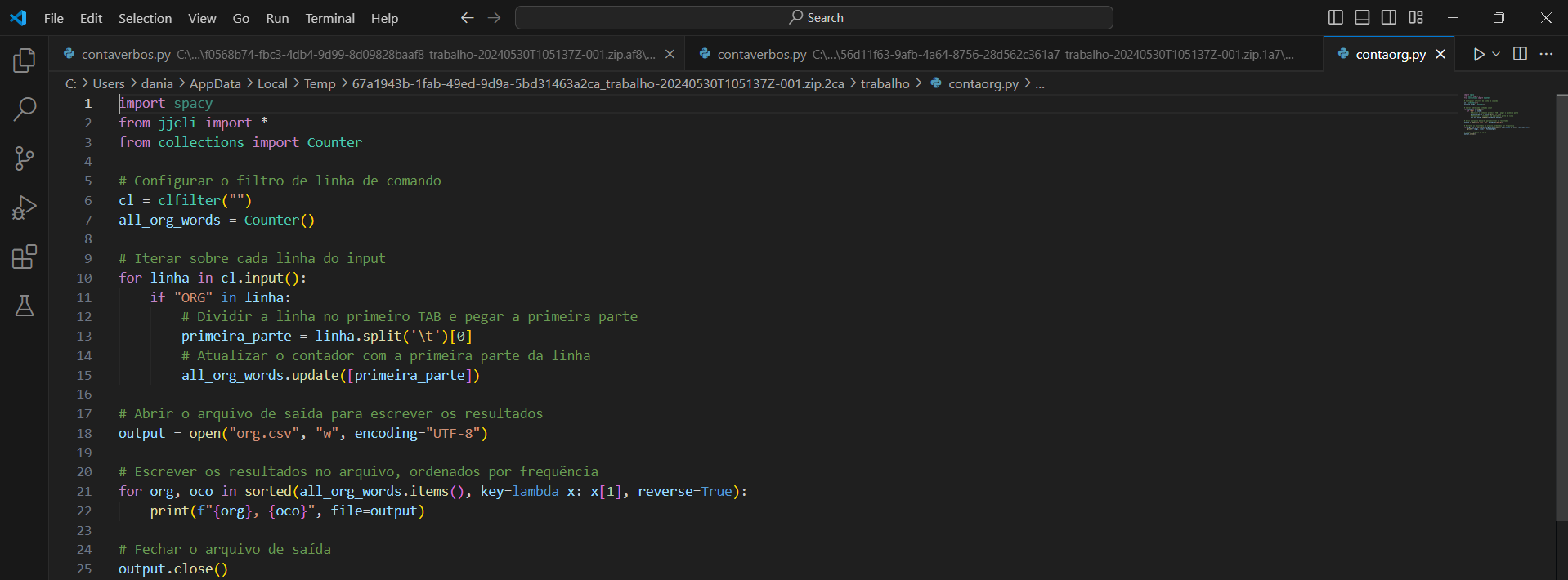


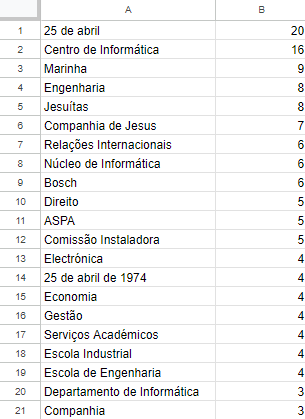
Palavras totais de cada entrevista



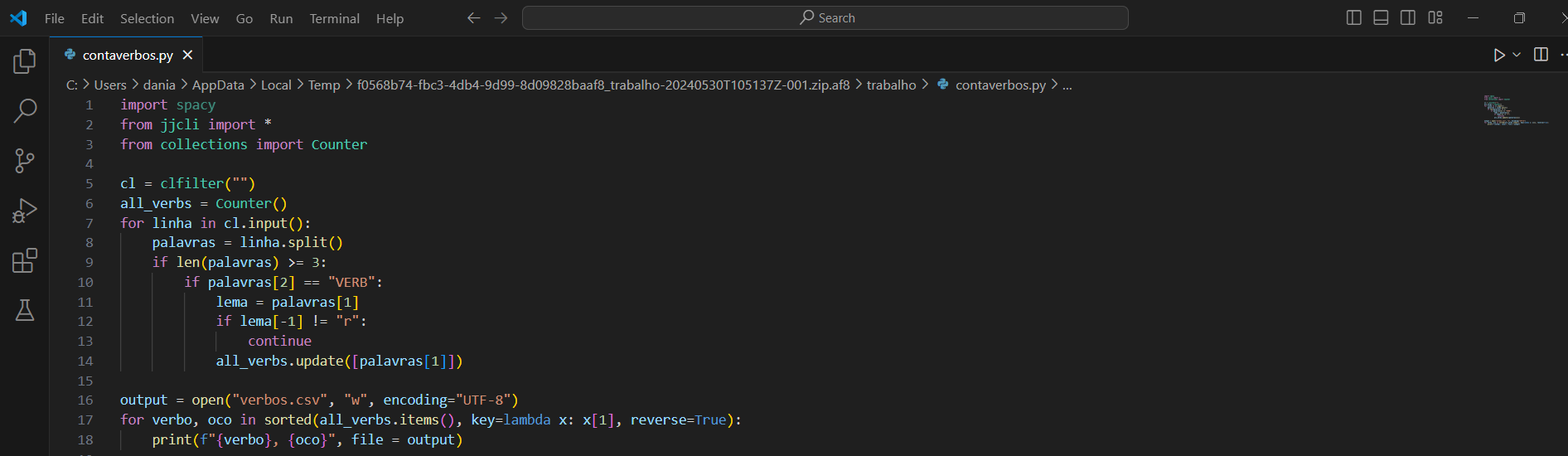


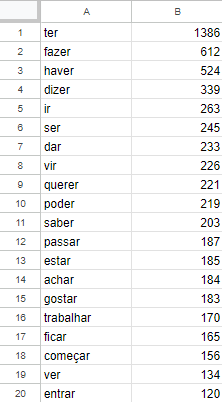
Entidades





Verbos





Objetivo: realizar a contagem dos verbos existentes no corpus em sua forma infinitiva (r ) e gravar o resultado num arquivo.csv

Explicação do código:

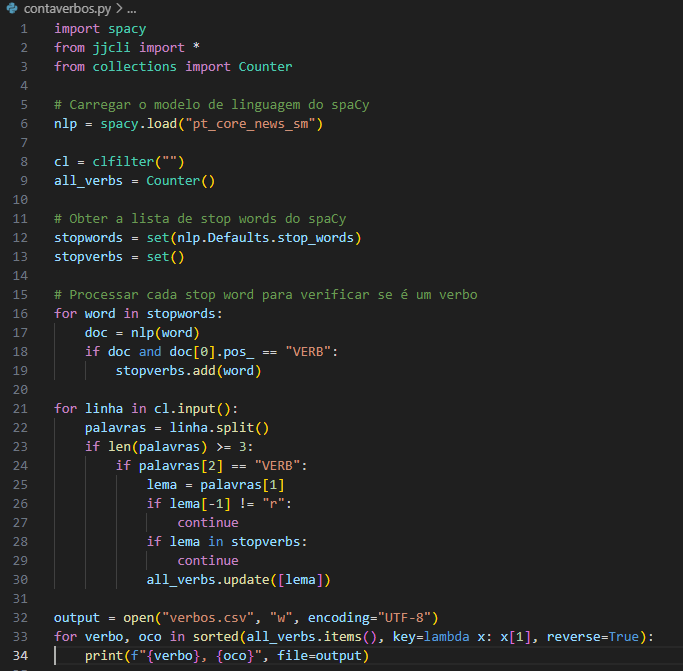
* Uso de “Counter”: trata-se de uma classe do módulo collections que facilita a contagem de itens únicos em uma lista.
* Uso de “Counter()”: para inicializar um contador para armazenar a contagem de verbos.
* Verifica “if palavras[2] == "VERB": verifica se a terceira palavra (índice 2) é "VERB", indicando que é um verbo.
* Verifica “if lema[-1] != "r": continue”: verifica se o lema do verbo termina com a letra "r". Se não, pula para a próxima iteração.
* Uso de “for verbo, oco in sorted(all\_verbs.items(), key=lambda x: x[1], reverse=True)”: percorre sobre os itens do contador de verbos, ordenando-os pela frequência em ordem decrescente.

Pretende-se que o código leia as linhas de um fluxo de entrada, analise cada linha para identificar verbos, conte as ocorrências de cada verbo cujo lema termine com a letra "r", e grave os resultados em um arquivo CSV. As linhas de entrada são esperadas no formato onde a terceira palavra indica a classe gramatical ("VERB" para verbos) e a segunda palavra é o lema do verbo.

No final, à exceção do código de extração dos verbos, os códigos de extração mudam muito pouco, uma vez que era necessário apenas mudar a categoria gramatical a extrair. Ainda assim, nas organizações, há uma ligeira alteração, uma vez que pretendíamos extrair não só a primeira palavra da organização, mas todas as palavras que a compunham. E, uma vez que havia organizações com mais do que uma palavra, explicitámos que devia ser extraído tudo até ao primeiro tab.

Reparámos também, como era de esperar, que nem todas as extrações foram bem conseguidas. Aquela que apresenta mais erros será os nomes próprios, uma vez que notámos a presença de inúmeras palavras que claramente não pertencem à mesma. Em categorias como verbos e adjetivos, consideramos que a taxa de acerto terá sido maior.

Posteriormente, uma vez que verificámos que a maior parte dos verbos que estavam a ser assinalados em posições cimeiras eram verbos auxiliares, considerados stopwords, decidimos fazer um código em que essas stopwords não fossem contadas, de forma a obter um resultado mais passível de ser estudado e analisado.



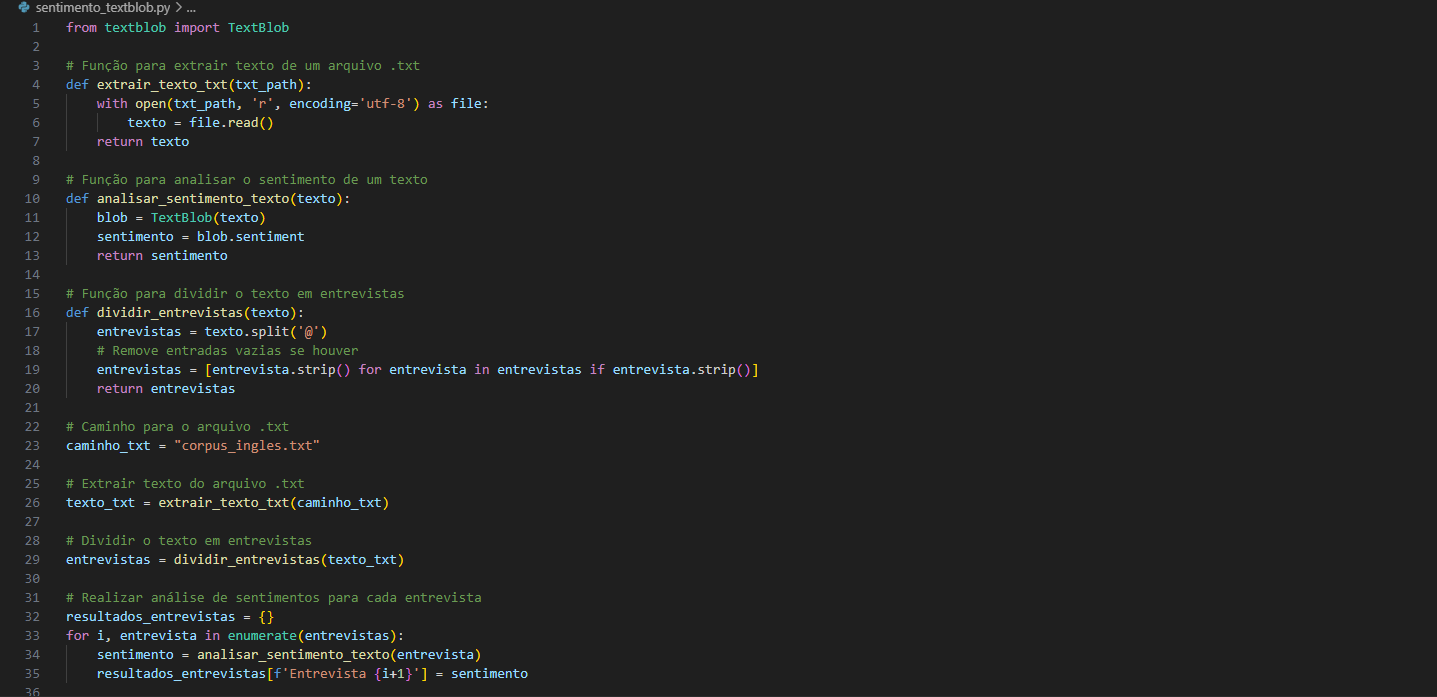
O resultado obtido foi o seguinte:

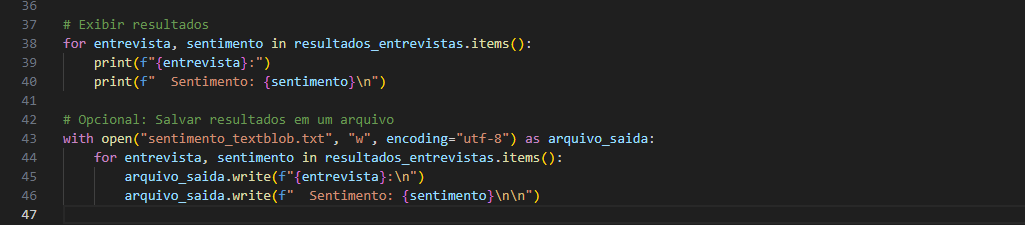


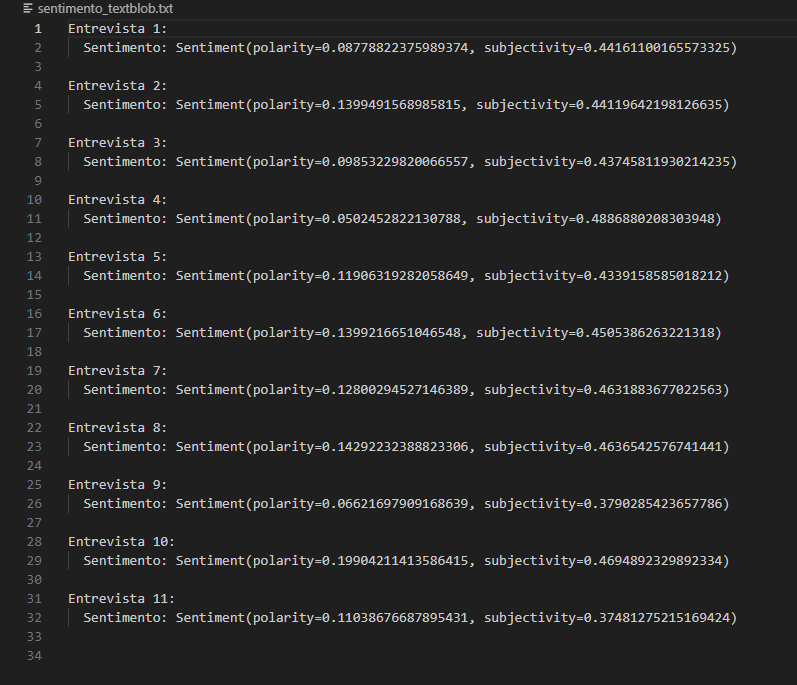
Apesar de termos especificado que não queríamos stopwords, e de alguns verbos terem sido excluídos, consideramos que a limpeza não foi totalmente bem sucedida, tendo em conta que verbos como “ser”, “estar” e “ficar” ainda foram contabilizados. Seria necessária uma posterior limpeza dos dados obtidos.

**Análise de sentimentos:** Realizamos a análise de sentimentos no corpus-umsombra. De forma a obter resultados mais precisos, traduzimos o corpus para inglês. Optámos por fazer esta análise de duas formas: utilizando uma biblioteca (TextBlob) e o sentillex. Os resultados obtidos foram os seguintes:

TextBlob:





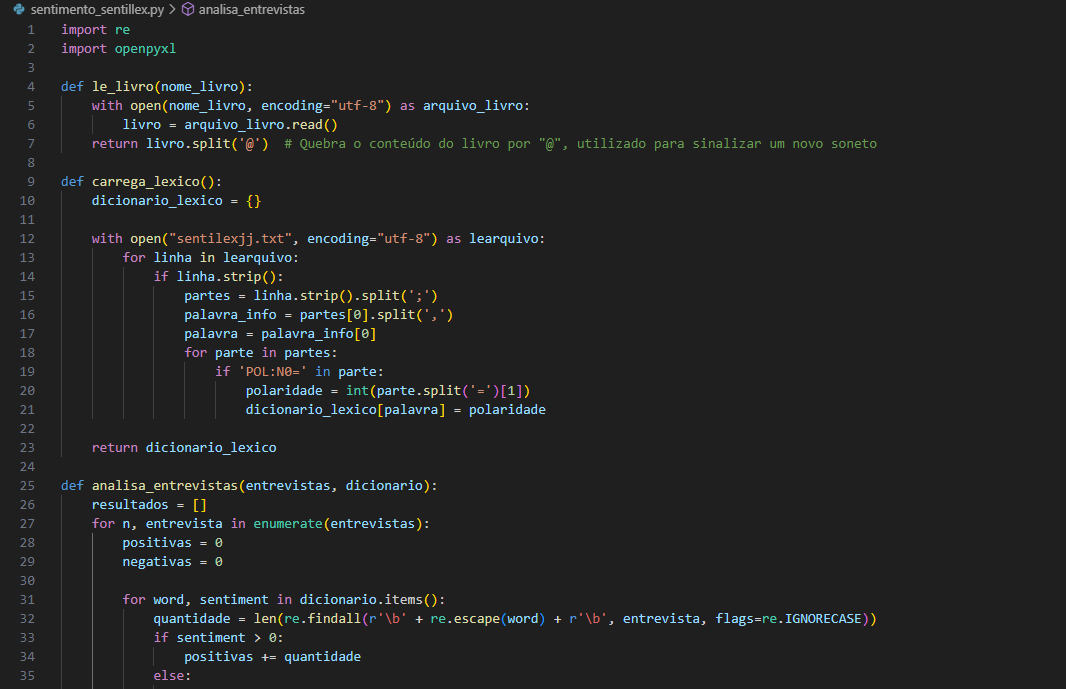


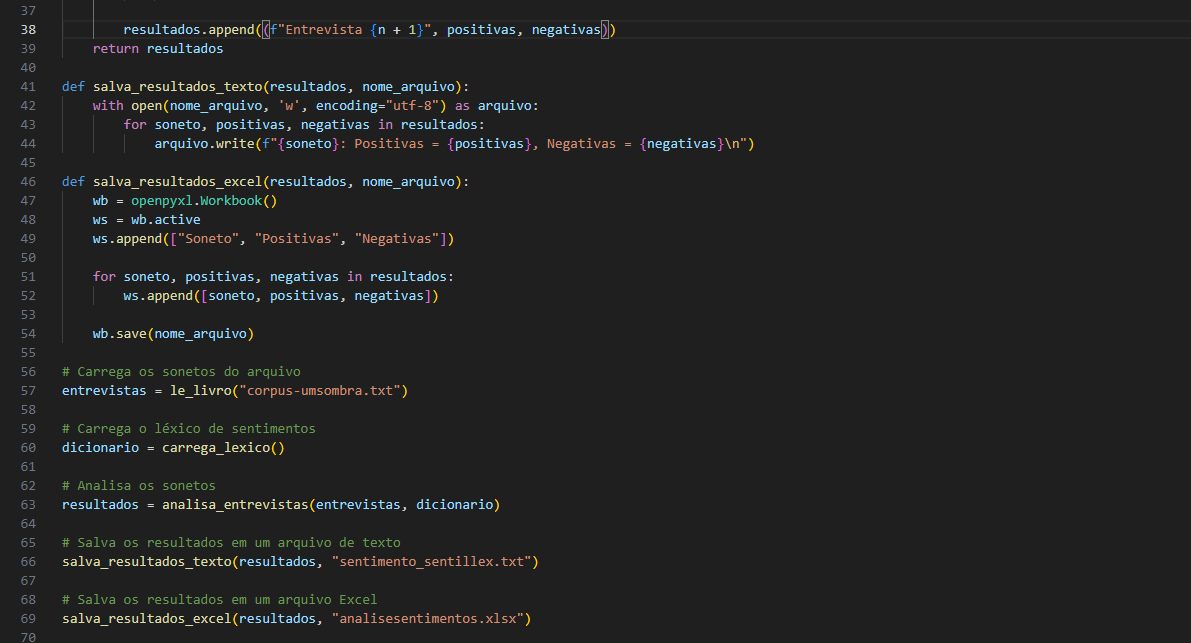
Objetivo: Realizar a análise de sentimento em frases individuais extraídas de um arquivo de texto (.txt) com a divisão em frases e análise do sentimento de cada uma delas.

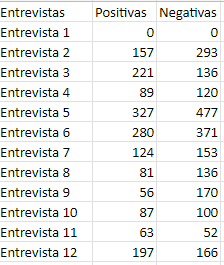
O código lê um arquivo de texto contendo várias entrevistas separadas pelo caractere @, divide o texto em entrevistas individuais, analisa o sentimento de cada entrevista utilizando TextBlob, e então exibe e salva os resultados da análise de sentimento.

Neste caso, quase todas as entrevistas apresentam resultados moderadamente positivos, sendo que a entrevista 10 é a mais positiva de todas.

Sentilex:

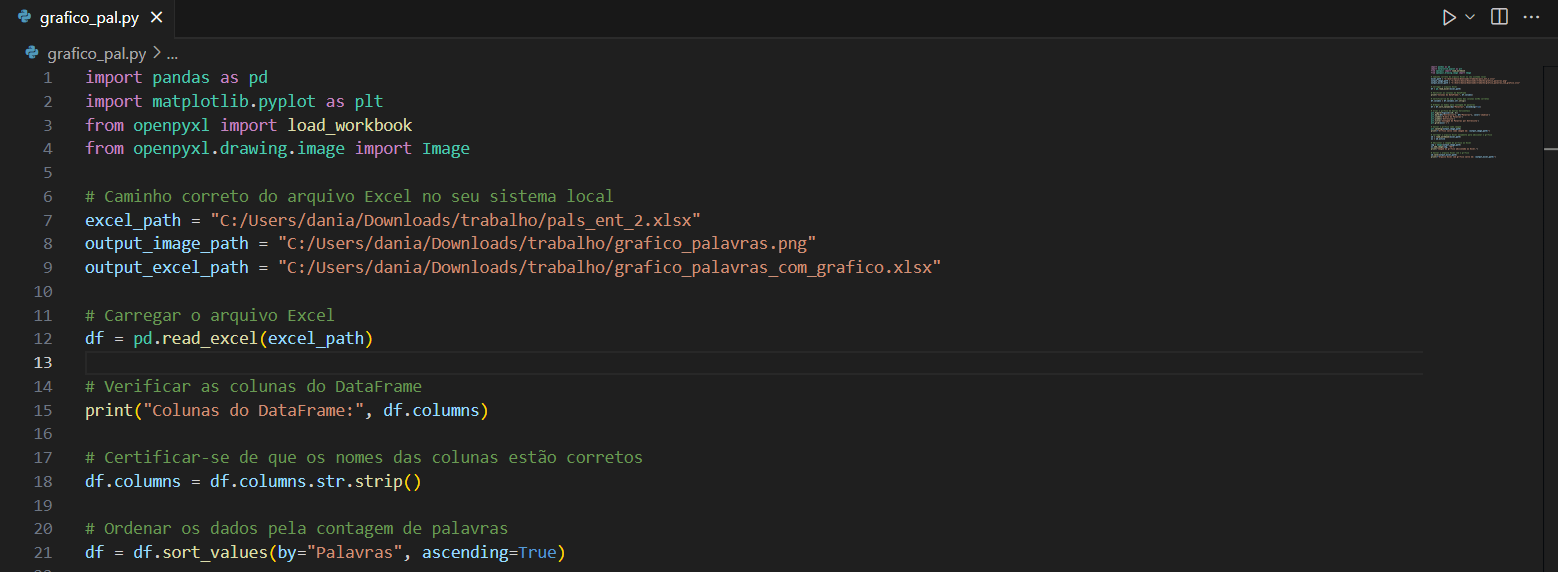


****

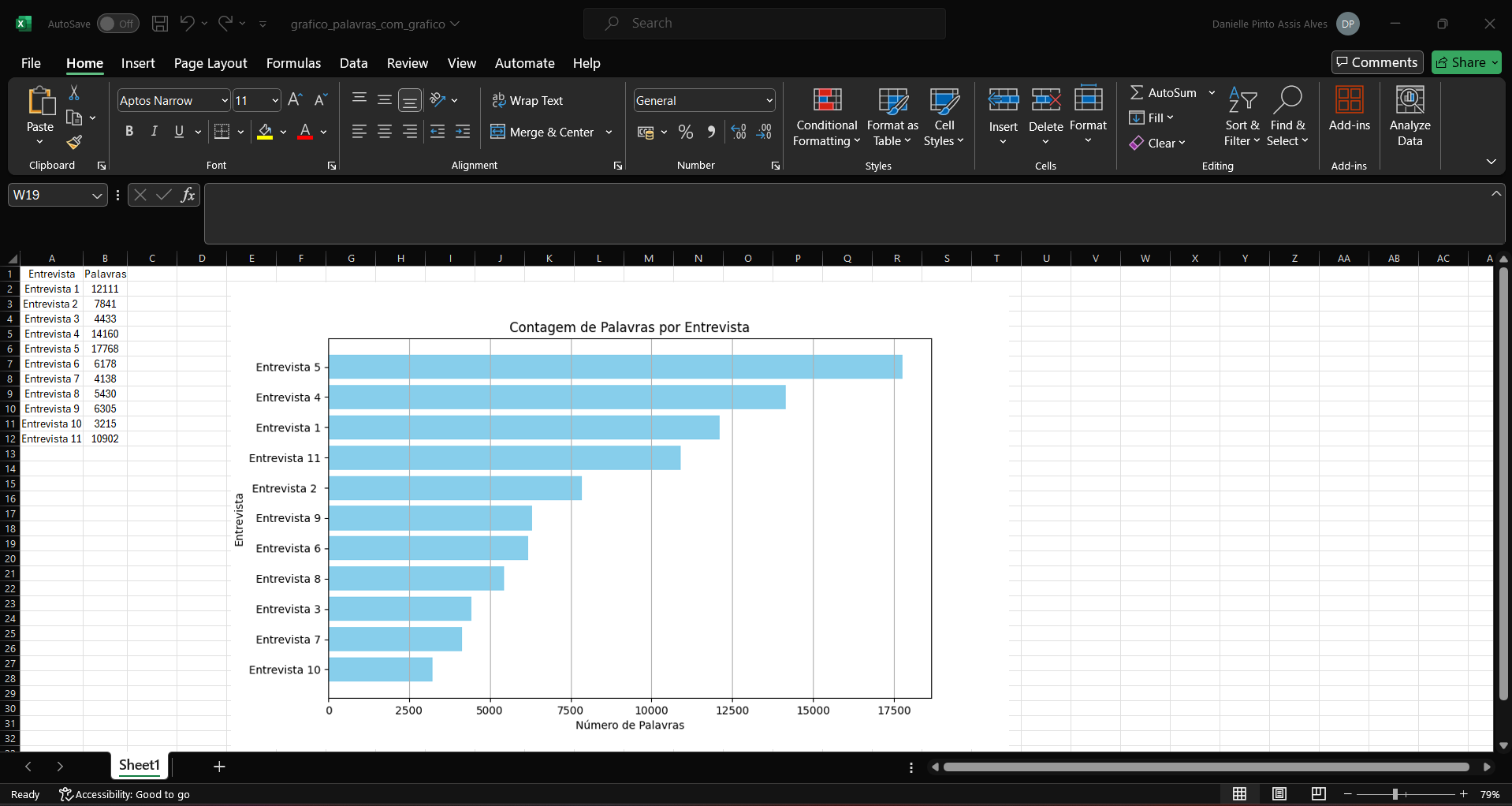


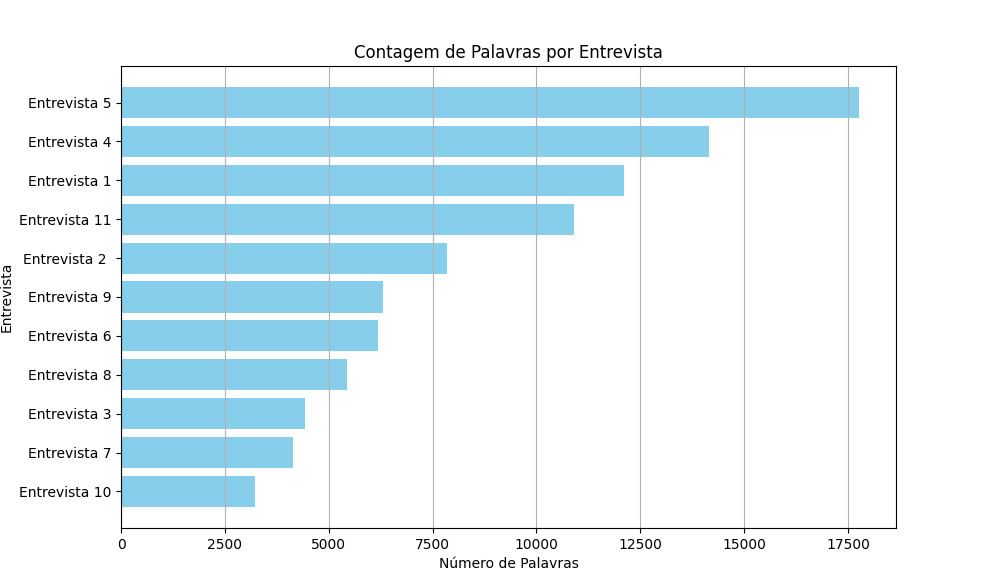
**VISUALIZAÇÃO**

**Gráfico com contagem de palavras totais:** Aplicamos o código abaixo para gerar um gráfico de barras horizontais com a contagem de palavras totais por entrevistas.







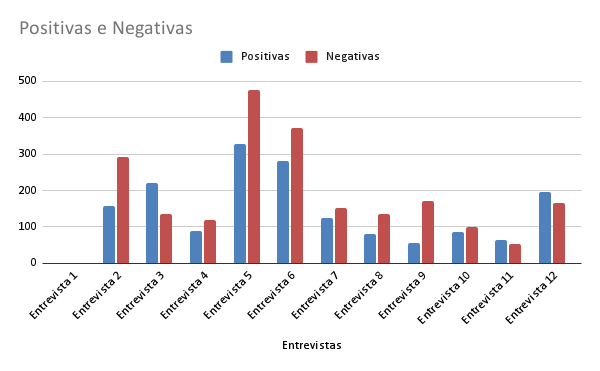


Objetivo: Gerar um gráfico de barras para visualização da quantidade de palavras por entrevistas. Pretendia-se que o código realizasse a leitura de dados de um arquivo excel, criasse um gráfico de barras horizontais com esses dados, salvasse o gráfico como uma imagem e depois inserisse essa imagem de volta em um novo arquivo excel.

Explicação do código:

* Uso das bibliotecas pandas, matplotlib.pyplot, openpyxml e image: para analisar e manipular dados de dataframe, criar gráficos e visualizações, ler e escrever arquivos excel, adicionar imagem em arquivos excel.
* Definição dos caminhos: locais onde estão salvos o arquivo base em excel e as pastas onde serão salvas a imagem e o novo gráfico.
* df = df.sort\_values(by="Palavras", ascending=True): para ordenar as dados pela contagem de palavras.
* Criação, salvamento e carregamento do gráfico.

**Palavras positivas e negativas de cada entrevista:**

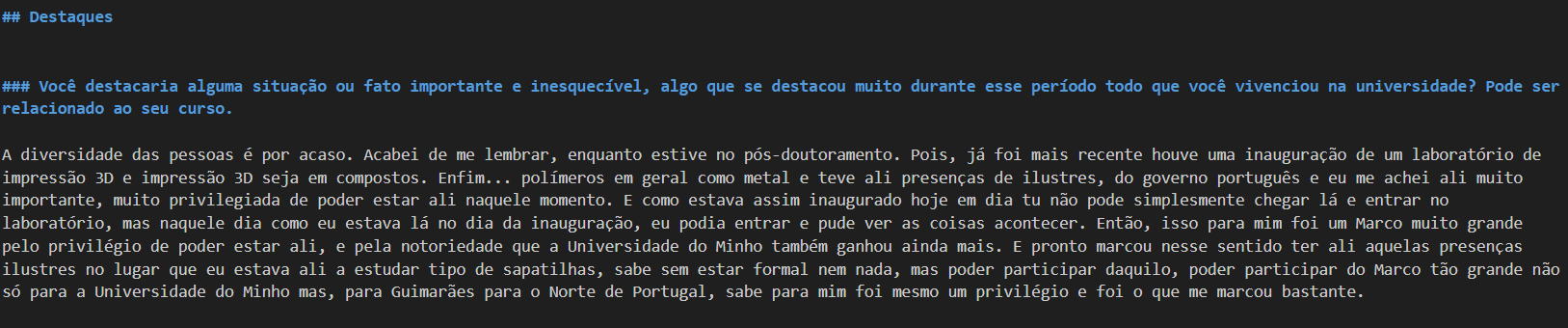


Este gráfico foi gerado no próprio Google Sheets, e permite-nos visualizar, de forma clara, a quantidade de palavras positivas e negativas de cada entrevista. Por algum motivo alheio ao nosso conhecimento, o próprio código interpretou o código como tendo 12 entrevistas, e imprimiu a primeira como uma vazia, fazendo com que a “Entrevista 2” seja a primeira entrevista, e a “Entrevista 1” não seja contada. É algo que podia facilmente ter sido corrigido no Google Sheets, eliminando a linha, no entanto decidimos manter, de forma a mostrar que é fruto do resultado do próprio código.

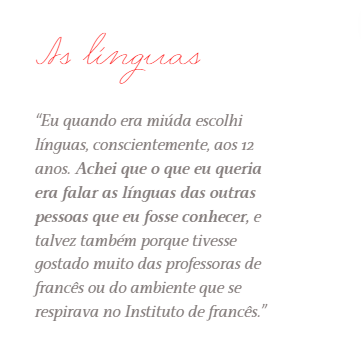
**DESTAQUES**

Sobre as entrevistas, destacamos os seguintes

Entrevistada Deborah Perrota de Andrade



Entrevistada Maria Filomena Louro

****

**CONCLUSÃO**

Assim, conclui-se que os objetivos do projeto e da disciplina foram cumpridos. Destacamos como pontos positivos a riqueza do acervo trabalhado, a possibilidade de desenvolvimento das competências, conhecimentos e habilidades linguísticas e tecnológicas para observar, selecionar, analisar e sintetizar os dados. Através da realização deste trabalho, pudemos inteirar-nos acerca da conveniência da Makefile e da sua utilidade, da elevada quantidade de informação que pode ser extraída de um corpus como este e da importância da escolha dos gráficos certos para a visualização de dados. Como pontos a melhorar identificamos a necessidade de aprimorar a composição dos códigos para geração de visualização. Contudo, há de se considerar o pouco tempo para exercitação e desenvolvimento de perguntas de investigação desse tópico da disciplina.

**BIBLIOGRAFIA**

Github. 2024. <https://github.com/jjoao/HD2023/tree/master/UmSombra/HV-ProjectoUM>. Acesso em 29/05/2024.

Núcleo Português do Museu da Pessoa. 2024. <https://npmp.epl.di.uminho.pt/overview.html>. Acesso em 29/05/2024.