



UNIVERSIDADE DO MINHO

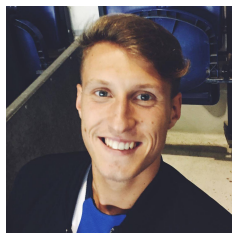
MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

LEI - LABORATÓRIO ENGENHARIA INFORMÁTICA

---

## Sistema de apoio à ingestão e catalogação de posts e comentários

---



*André Salgueiro (A77617)*

*Bruno Carvalho (A76987)*

*Fábio Araújo (A78508)*

1 de Julho de 2019

## Resumo

Este documento diz respeito ao projeto de LEI, Laboratórios de Engenharia Informática, relativo ao perfil de PLC, Processamento de Linguagens e Conhecimento, do quarto ano de Engenharia Informática.

De uma forma sucinta, este projeto encontra-se inserido num projeto internacional, denominado NetLang, cujo principal objetivo é a análise de comentários de teor discriminatório em plataformas online, como são exemplo, as redes sociais.

O objetivo deste projeto visa a criação de um método computacional que elimine a arduidade do processo de análise dos vários documentos um *corpus* manualmente.

O código desenvolvido durante a realização do projeto encontra-se disponível no repositório [Git](#).

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Enquadramento</b>	<b>5</b>
<b>3</b>	<b>Contexto</b>	<b>6</b>
<b>4</b>	<b>Problema</b>	<b>7</b>
<b>5</b>	<b>Objetivos</b>	<b>8</b>
<b>6</b>	<b>Arquitetura do Sistema</b>	<b>9</b>
<b>7</b>	<b>Implementação do Sistema</b>	<b>10</b>
7.1	Pré-processamento/Normalização . . . . .	10
7.2	Análise Léxica/Analisador . . . . .	10
7.3	Interface . . . . .	11
<b>8</b>	<b>Resultados</b>	<b>13</b>
<b>9</b>	<b>Conclusão</b>	<b>21</b>
<b>10</b>	<b>Anexos</b>	<b>23</b>

## Lista de Figuras

1	Arquitetura do Sistema . . . . .	9
2	Primeira parte da tabela “resultado.csv” . . . . .	15
3	Segunda parte da tabela “resultado.csv” . . . . .	16
4	Formulário de submissão de um ficheiro- 1ª Parte . . . . .	17
5	Formulário de submissão de um ficheiro- 2ª Parte . . . . .	18
6	Formulário de submissão de um ficheiro- Resultados do Analisador . . . . .	19
7	Formulário de submissão de uma keyword . . . . .	20
8	Visualização de um <i>post</i> previamente analisado/inserido . . . . .	20

## 1 Introdução

Este documento refere-se ao projeto de LEI, Laboratórios de Engenharia Informática, relativo ao perfil de Processamento de Linguagens e Conhecimento, do quarto ano de Engenharia Informática.

O projeto encontra-se inserido num projeto internacional, denominado NetLang, cujo objetivo é o estudo de comentários ofensivos e discriminatórios em plataformas online, como as redes sociais.

Ao longo deste documento iremos explicar o desenvolvimento deste projeto. Numa primeira fase procura-se apresentar o projeto, indicando no que consiste e o seu propósito. Numa segunda fase procura-se apresentar a abordagem tomada pelo grupo de trabalho ao longo deste projeto de maneira a atingir os objetivos anteriormente definidos.

Numa fase final deste documento serão expostos os resultados conseguidos, com a apresentação de exemplos concretos, e a projeção do trabalho futuro.

## 2 Enquadramento

Este projeto surge no âmbito dos projetos de Laboratórios de Engenharia Informática, sendo estes à escolha dos alunos que possuam o perfil que abrange o projeto em específico. Este projeto encontra-se inserido no perfil de Processamento de Linguagens e Conhecimento, e é suportado no aprendizado deste perfil até então fornecido que irá decorrer o trabalho desenvolvido sobre o projeto.

É um projeto internacional de investigação suportado pela FCT, Fundação para a Ciência e a Tecnologia, denominado NetLang. O NetLang tem como objetivo a criação de um corpora<sup>1</sup> anotado que permita estudar formas de insulto e discriminação presente nos painéis de comentários das redes sociais e sites de jornais.

O corpus construído vai conter o conjunto de *threads* de comentários e a respetiva metainformação de cada *thread*. A nossa componente no *NetLang* tem como objetivo construir um sistema que analise a *thread* de comentários e a partir de um lista de exemplos de preconceitos já classificada identifique palavras preconceituosas nos comentários. Desta forma irá classificar o tipo de preconceito presente nos comentários.

---

<sup>1</sup>plural de *corpus*- Colectânea acerca de um mesmo assunto.

## 3 Contexto

É importante perceber que a pretensão do projeto NetLang visa a compilação de uma vasta coleção de *posts* e os seus respetivos comentários, pelo que a obtenção da meta-informação por meios manuais representa uma tarefa quase impossível, senão impossível, impraticável quando associado o tempo de processamento de tal tarefa.

Por este motivo, a criação de um método computacional que possibilite a obtenção dos mesmos resultados mas que elimine a arduidade deste processo é estritamente necessária.

A ferramenta a criar, dentro das suas capacidades, deverá possibilitar a recolha de dados estatísticos, o processamento semi-automático da meta-informação e a representação dos *posts* e comentários em formato TEI.

## 4 Problema

Nos dias de hoje, cada vez mais a população recorre a redes sociais, quer via telemóvel quer via computador, para expressar os seus pensamentos, fazer críticas à sociedade e outros tipos de entidades.

No entanto, frequentemente verificamos que esses comentários podem ser ofensivos para certos indivíduos e são feitos sem qualquer tipo de filtro, sendo que muitas vezes até passam despercebidos para quem os visualiza.

No desenvolvimento deste projeto procura-se resolver estes problemas, sendo que dentro dos comentários ofensivos existe um determinado leque de preconceitos. Assim sendo, é feita uma análise aos *posts* e seus comentários para obtermos uma noção da frequência da existência destas ofensas, para posterior estudo na área da linguística .



# 5 Objetivos

Relembrando aquilo que foi referido anteriormente, com a introdução das temáticas relativas aos comentários e às palavras-chave (*keywords*), e principalmente a matéria de trabalho (ficheiros em análise) é possível traçar com clareza os principais objetivos deste projeto.

Primeiramente, dado que a informação relativa às palavras-chave encontra-se em contexto de tabela, é importante que a mesma seja representada num formato estruturado que permita um fácil manuseamento, como é o formato **JSON**. Na mesma linha de pensamento, uma vez que a informação relativa aos comentários pode encontrar-se no formato **CSV**<sup>2</sup> é necessário desenvolver um programa responsável pela transformação dessa informação para o formato **JSON**.

Seguidamente, depois de normalizada toda a informação a tratar, por forma a possibilitar a associação dos *posts* aos tipos de preconceito, é necessária a criação de um motor de procura das palavras-chave em cada um dos comentários.

Posteriormente, em resultado do trabalho realizado pelo motor de procura, é solicitada a criação de uma análise estatística, baseada na ocorrência das palavras-chave, com apresentação em formato à escolha, e por fim a criação de uma interface híbrida que para além de acolher os resultados do motor para a construção da meta-informação, dá oportunidade ao utilizador de a complementar.

Além destes objetivos, decidimos traçar também a hipótese do utilizador ter ao seu dispor a funcionalidade de acrescentar à nossa base de dados de *keywords* mais informação. Esta tarefa irá fazer com que o utilizador possa identificar *Keywords* mais específicas e tenha uma maior interação com a aplicação e com a análise dos respetivos documentos.

---

<sup>2</sup>Comma-separated values

## 6 Arquitetura do Sistema

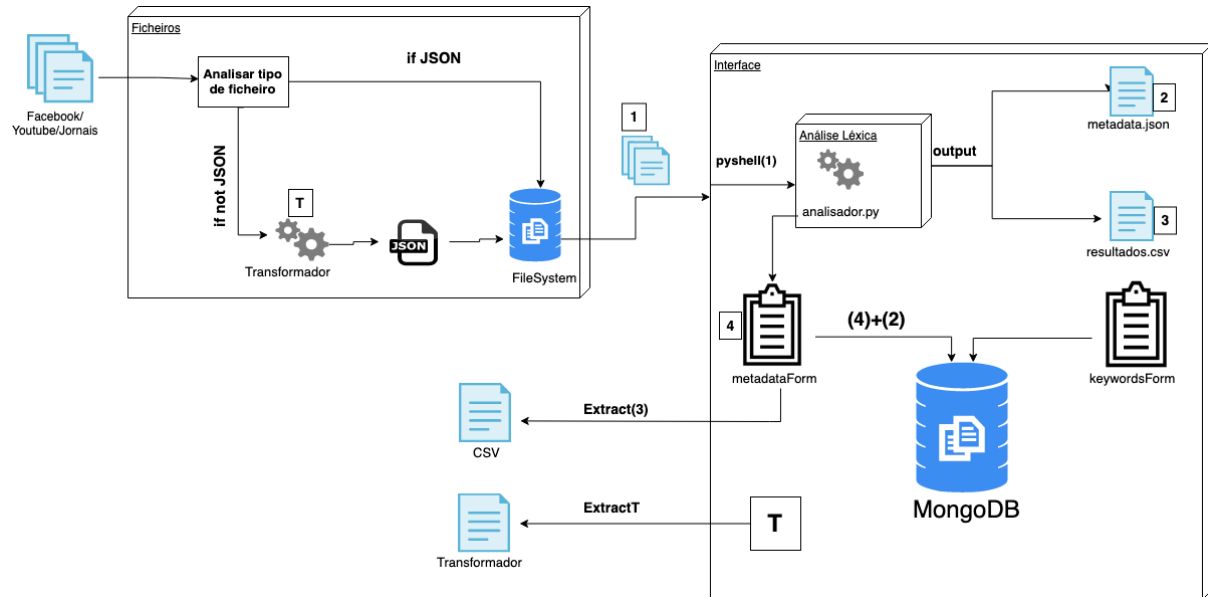


Figura 1: Arquitetura do Sistema

O trabalho a desenvolver ao longo do projeto, em forma de esquema pode ser observado na figura acima apresentada 8 proporcionando assim uma visão mais ampla e facilitada do projeto.

Com auxílio da figura, é possível traçar a sequência de eventos do sistema e os seus principais envoltentes.

Primeiramente existe uma fase de pré-processamento dos dados que consiste na conversão dos ficheiros para formato *json*. Estes ficheiros representam os dados extraídos dos *media*. Depois de estes dados se encontrarem devidamente normalizados, estes podem ser ingeridos pela *interface* para o processamento dos *meta-dados* no analisador, através da biblioteca *python-shell*, em *NodeJS*, que permite a execução de programas em *Python*. Da execução do analisador, são gerados dois documentos, um com os resultados estatísticos em *CSV* e outro com os valores de *meta-dados* que foram processados pelo analisador, que serão acrescentados ao ficheiro a submeter. Após a execução chegamos à submissão do documento no formulário “*metadataForm*”. Neste momento para além da submissão do ficheiro para o *MongoDB*, com parte da informação pré-processada pelo analisador (4) e outra parte introduzida pelo utilizador, é possível a extração do documento *CSV*, anteriormente gerado.

Pela figura é também visível a possibilidade de inserção de *keywords* para o *MongoDB*, a partir do formulário “*keywordsForm*”, e a possibilidade de extração do *script* de transformação dos dados para formato *json* (T).

# 7 Implementação do Sistema

Nesta secção irão ser descritas as tomadas de decisão mais relevantes que levaram à concretização da implementação do sistema tendo em conta todos os objetivos definidos anteriormente, que se encontra sucintamente ilustrado na secção 6.

Tomando, então, como guião para esta secção, a arquitetura do sistema, já referenciada, é possível identificar três grandes áreas de ação, **Pré-processamento/Normalização**, **Análise Léxica/Analizador** e **Interface**.

## 7.1 Pré-processamento/Normalização

Este *pré-processamento*, faz referência à transformação dos dados de extração das diferentes fontes de *media* sujeitas a análise e do *dataset* com a representação das *keywords*, para um formato que mais se adequasse às necessidades posteriores de processamento e representação dos dados.

Por conseguinte, ficou determinado que o formato mais adequado seria o *json*, dado que este apresenta características que beneficiam as ações sobre os documentos em questão. Esmiuçando, o formato *json* em relação aos demais, para além de facilitar a leitura, dado a leveza da sua sintaxe, e por consequência, a redução do tamanho do documento, o importante realçar é a maior capacidade de *parsing*, execução e transporte dos dados, características que serão realçadas nas posteriores áreas de ação.

Para o efeito, no caso da extração efetuada das diferentes plataformas de *media* foram criados *scripts* que convertessem os dados extraídos em formato *csv* para formato *json*. A escolha de linguagem para a construção dos transformadores recaiu sobre o *Python*, dado ao facto de se apresentar como uma linguagem de *scripting* e principalmente pelo facto de esta apresentar um grande suporte e simplificado nas mais diversas áreas, desde já no tratamento de ficheiros/objetos do tipo *json*.

Agora no caso da transformação do *dataset* relativo às *keywords*, este decorreu de forma manual.

## 7.2 Análise Léxica/Analizador

Esta talvez seja a área de ação mais importante no sistema, pois é nesta que se encontra implementado o analisador que trata do reconhecimento das *keywords* dentro do inventário de comentários em análise, da criação de uma tabela estatística das ocorrências das *keywords* e da associação das *keywords* e *variáveis socio-linguísticas* aos meta-dados que serão construídos na secção 7.3.

Para a construção do analisador, optou-se novamente pela utilização da linguagem *Python*, pelos motivos já referidos anteriormente, e por ter um grande suporte no que toca a questões de *Processamento de Linguagem Natural* (NLP).

De seguida apresentar-se-ão, os aspetos mais importantes na construção do analisador.

O primeiro aspeto a referir é a inclusão de uma classe *Comentário*, que será o objeto que transportará a informação do comentário ao longo do seu processamento. Esta classe é constituída pelas seguintes propriedades:

**comment\_id** Identificador do comentário

**commentMessage** Texto do comentário

**user** Identificador do utilizador com triplos cujos valores, são representados, respetivamente, pela *keyword*, pelo número de ocorrências dessa *keyword* e pelo número total de palavras no texto do comentário

A principal propriedade desta classe é a propriedade *occurrences*, pois é nela que é guardada o conteúdo chave da ação de *parsing* dos comentários, que é a obtenção ou não, e posterior contabilização, do *matching* das *keywords* nos comentários, basicamente que identifica se estamos na presença de um comentário preconceituoso e de que género, e o quantifica em termos de frequência relativa e absoluta.

Como será de esperar se tal propriedade toma tamanha importância, maior importância terá a função que carrega a informação. De facto é possível atribuir à função em questão, *checkNcount()*, o protagonismo principal, pois esta trata do *Processamento de Linguagem Natural* dos comentários com a ajuda do módulo *FuzzyWuzzy* do *Python*, que permite a comparação aproximada de *strings*. No caso específico deste sistema, o rácio estipulado para a aproximação encontra-se nos 75%. Toda a *string* dentro do comentário que corresponda em parte, com um valor igual ou superior a este rácio, a qualquer uma das *keywords*, fará desse comentário, um comentário preconceituoso.

Talvez, dos últimos aspetos relevantes, que façam sentido aqui serem mencionados, em relação à implementação do analisador, sejam a definição de duas funções, uma que trata da construção do ficheiro *CSV* com as estatísticas de ocorrências de *keywords* nos comentários, e outra que trata da construção de um ficheiro *json*, que representa os metadados do *post* (ficheiro) em análise, que contém a lista das *variáveis sociolinguísticas* e a lista das *keywords* que ocorreram no *post*. Ambas as funções, ingerem informação resultante do trabalho operado pela função anteriormente referida, para poderem construir os seus ficheiros.

### 7.3 Interface

A área de ação relativa à *Interface*, representa uma aplicação *web* que procura suprir as necessidades dos consumidores deste sistema, que neste casos são indivíduos da

## 7. IMPLEMENTAÇÃO DO SISTEMA

---

área da linguística, na obtenção automatizada de resultados significativos que permita avaliar a existência de comentários de teor preconceituoso e a sua respetiva profundidade, por outras palavras, o quão preconceituoso um *post* é, tendo em conta a avaliação dos comentários inerentes, a construção dos *meta-dados* e o auxílio na inserção de novas *keywords* ao *dataset* respetivo.

Para a construção da plataforma *web*, optou-se pela incursão num contexto já conhecido pelos membros do grupo, em âmbito de desenvolvimento *web*. Sendo que as ferramentas escolhidas para a implementação da plataforma passam pela utilização do *Express*, uma *framework* de desenvolvimento *web* para *NodeJS*, que é um ambiente de execução de código *javascript*.

Para a concretização das funcionalidades acima mencionadas, estabeleceu-se que a melhor abordagem passaria por uma forma de persistência de dados em *MongoDB*, que aproveitasse o facto dos documentos estarem em *json*, e que possibilitasse, para além da obtenção estática de resultados, uma visualização, dinâmica no tempo, dos ficheiros submetidos, já com a presença dos *meta-dados* acrescentados ao documento (ficheiro) em foco, e das *keywords* até ao momento definidas.

O estabelecimento do controlo da base de dados, é efetuado a partir da ferramenta *Mongoose*, que com a modelação de objetos *MongoDB*, permite a conexão à base de dados e a construção de *querys*. A partir dos modelos criados será possível transportar informação desde a base de dados até ao *Cliente*, e vice-versa. Isto é, permite a inserção de informação na base de dados a partir de um qualquer pedido HTTP POST do lado do *cliente* e a visualização de informação da base de dados a partir de pedidos HTTP GET. Quem trata do controlo destes pedidos HTTP é a biblioteca de *NodeJS*, chamada *Axios*.

Depois de estabelecida a base de dados, o principal objetivo da aplicação é acondicionar o analisador da secção anterior, para tal dispôs-se da biblioteca *python-shell*, que permite a execução de programas em *Python* dentro do *NodeJS*. Neste momento, reúnem-se todas as condições para a intercomunicação entre o analisador e a plataforma.

A base desta plataforma possibilita assim a persistência de dados das *keywords*, possibilitando a inserção de uma *keyword* numa dada *variável sociolinguística*, e a persistência de dados dos extratos *media* com a associação dos *meta-dados* resultantes da execução do analisador.

Por último foi necessária a adição de uma funcionalidade à aplicação, como forma de controlo de formatação do ficheiro a ser submetido. Isto representa um ponto fulcral da aplicação, pois caso este não coincida com o formato que o analisador espera, este incorrerá imediatamente em erro. A adição desta funcionalidade é possibilitada pela biblioteca *Ajv*, que faz a validação da estrutura do ficheiro submetido, comparando-a com um esquema previamente estabelecido, que representa a estrutura possível de ficheiros a serem submetidos.

## 8 Resultados

Tendo em conta a definição dos objetivos para este projeto, passemos agora a apresentar os resultados obtidos.

- Normalização

- ◊ Coletânea de palavras-chave (*keywords*) relativa a cada preconceito.

A seguinte tabela representa o exemplo dos dados fornecidos inicialmente para o projeto, neste caso de um excerto associado ao preconceito '*Sexismo*'.

Types od Prejudice	Sociolinguistic Variables	Keywords (English)	Keywords (Portuguese)
<b>Sexism</b>	Gender	- Male chauvinism /chauvinist. - Gender. Sex- (Sexual, Sexism... ). Misogyn- (-y, -ous, -nist, -e). Patriarchy. - Woman. Chick. Dame.	- Machismo. Machista. - Género. Sex- (Sexual, Sexismo). Misogin- (Misógino, Misoginia). Patriarcado. - Mulher. Gaja. Tipa. Garina. Chavala. Miúda.

Tabela 1: Extrato das Keywords fornecidas em forma tabular

Como já explicado anteriormente, esta tabela foi manualmente transformada num ficheiro do formato *json*, correspondendo assim o extrato apresentado à estrutura seguinte:

```
{
  "type_prejudice": "Sexism",
  "sociolinguistic_variables" : {
    "pt":{
      "gender":
        ↪ ["Machismo", "Machista", "Género", "Sexual",
          "Sexismo", "Misógino", "Misoginia",
          "Patriarcado", "Mulher",
          "Gaja", "Tipa", "Garina", "Chavala", "Miúda"],
        ....
    }
    "en":{
      ...
    }
    ...
  }
}
```

## 8. RESULTADOS

---

### ◇ Post

Inicialmente foi colocado o problema ao grupo de que a informação poderia estar em formato *csv*, sendo assim a tabela agora apresentada representa um extrato da estrutura excel que teria de ser posteriormente transformada em formato *json* pelas várias razões apresentadas anteriormente.

position	post-id	pos-by	post-text	post-published	comment-id
61-0	4699745104	c992ae	2019-01-25	10161	da39a3ee5e

Tabela 2: Extrato do conteúdo de um post em excel

De seguida, encontra-se um extrato da estrutura *json* resultante da transformação exercida pela ferramenta desenvolvida pelo grupo em *python* de nome “*forJSON-v1.py*” para, neste caso específico uma linha do ficheiro situado na pasta de Extratos de nome *facebook\_extraction\_portuguese\_1.tab*:

```
{
    "position": "61-0",
    "post_id" : "4699745104",
    "post_by" : "c992ae",
    "post_text" : "Eu não quero ser mártir",
    "post_published" : "2019-01-25",
    "comment_id": "10161",
    "comment_by" : "da39a3ee5e"
    "is_reply" : 1
    "comment_message" : "Ele já chegou",
    "comment_published" : "2019-01-25",
    "comment_like_count": 0,
    "attachment_type" : "",
    "attachment_url" : "",
    ...
}
```

### • Analisador

Pegando na ilustração 8, é possível apresentar como resultados para o analisador o ficheiro “*metadata.json*” e o ficheiro “*resultados.csv*”.

### ◇ *metadata.json*

Ficheiro de representação intermédia que serve para a posterior submissão do ficheiro com os *meta-dados* devidamente acrescentados.

Alteração dos headers dos extratos sujeitos a análise preenchendo os respetivos campos com as *keywords* e *variáveis sociolinguísticas* encontradas.

```

{
  ....
  "svs": [
    "gender",
    "ethnicity",
    "nationality",
    "social_class",
    "sexual_identity"
  ],
  "kws": [
    "Gaja",
    "Garina",
    "Mulher",
    "Raça",
    "Nação",
    "Classimo",
    "Homofóbico"
  ]
  ...
}

```

Acima encontra-se um exemplo do resultado do analisador no processamento da informação necessária para a construção dos *meta-dados* a adicionar ao *header* do extrato. O campo “svs” representa as *variáveis sociolinguísticas* que foram encontradas, e o campo “kws” representa as respectivas *keywords*. Para esta análise é utilizado o analisador que se encontra dentro da nossa aplicação na pasta publica com o nome de *pyscripts*.

◇ *resultados.csv*

Ficheiro de representação estatística da ocorrência de *keywords* no *post* (ficheiro). Abaixo encontra-se um pequeno excerto do que é possível observar a partir do ficheiro *CSV* proveniente da análise feita pelo *analisadorJSON-v6.py* do extrato de youtube de nome *Youtube\_extraction\_portuguese\_1.json*.

A primeira parte da tabela representa os comentários onde foram encontradas correspondências com as palavras reservadas (*keywords*)

Prejudice	Comentario
	A gaja não é burra de todo ORDINÁRIA isto tem nome, puta fina que já se desmarcou, e não foi agora, foi assim que percebeu que se continuasse a abrir as pernas ia parar à cadeia como cúmplice. O que eu constato de tudo isto é que essa Cândia é burra que nem uma porta, além de inculta: então namorou com ele, viveu em hotéis de luxo com ele, via a vida caríssima que ele lev Esta felicia ou por amizade ou por tb ser jornalista não ataca a cancio como devia ter feito, e ate tenta minimizar a atitude de cabra da cancio e mudar de todas as formas o assunto da t
Sexism	resumindo ...chupou bem o Socrates para seu beneficio.... e quase uma puta de alta roda resumindo ...chupou bem o Socrates para seu beneficio.... e quase uma puta de alta roda O Eduardo falou bem já a loira é mesmo burra Essa cancio é um nojo de mulher...oportunista, arrogante, feiosa, mentirosa, mesquinha etc Fogo, como é que eles conseguem manter a calma com aquela gaja a falar daquela maneira, eu ca dizia logo para mudar a postura ... a Fernanda Cancio é mesmo nojenta...a fazer-se de sonsa. VACA!

Figura 2: Primeira parte da tabela “resultado.csv”

A segunda parte da tabela representa as frequências relativas e absolutas



## 8. RESULTADOS

de cada palavra reservada, em cada comentário, representadas como um triplo, em que a primeira posição representa a palavra reservada, a segunda a contabilização de ocorrências e a última posição o total de palavras de cada comentário. Esta parte da tabela também apresenta o total de comentários preconceituosos, e o total de palavras reservadas por *post*.

Frequencia	Total	Ocorrencias
C6A5z1RTUK-y2A94AaABAg [['Gaja', 1, 7], ('Burra', 1, 7)]		Gaja ----> 2
:NZEky7rl1k02_0t4AaABAg [['Ordinária', 1, 1]]		Loira ----> 1
Vcg_wbFCVXy0_qd4AaABAg [['P*ta', 1, 30]]		Cabra ----> 2
Y04s840dmR8uYt94AaABAg [['Burra', 1, 86]]		Burra ----> 3
Gg597h03Ze-U-4l4AaABAg [['Cabra', 2, 56], ('Vaca', 1, 56)]		Vaca ----> 2
lVoPyjwDRBac_4l4AaABAg [['P*ta', 1, 15]]	11/42	Putá ----> 3
r6d-Zi5AFab6xNZ54AaABAg [['P*ta', 1, 15]]		Feiosa ----> 1
i41Jskp2s708xLD94AaABAg [['Burra', 1, 10], ('Loira', 1, 10)]		Ordinária ----> 1
Jlql0oCXRWSMRWx4AaABA [['Feiosa', 1, 12]]		
34c60BZB-p1LBhpd4AaABAg [['Gaja', 1, 25]]		
YGPEb2OJKqH4b4AaABAg [['Vaca', 1, 10]]		


Figura 3: Segunda parte da tabela “resultado.csv”

- Interface

Dentro da *Interface*, os resultados de realce são os dois formulários, anteriormente apresentados na figura 8, e a visualização de um *post* (ficheiro submetido).

- ◊ Formulário da submissão de um ficheiro

Como se pode observar pela figura 4, os valores que se encontram associados aos campos “Sociolinguistic Variables” e “Keywords”, são exatamente iguais ao processamento efetuado pelo analisador, que serve de apêndice ao formulário, que se encontra visível na representação do ficheiro “metadata.json” acima demonstrado.

Harambe

Analisi

Metadata Submission

**NLP - Natural Language Processing**

Sociolinguistic Variables

☐gender

☐ethnicity

☐nationality

☐social\_class

Add sociolinguistic variable

Add

Keywords and Expressions

☐Machismo

☐Machista

☐Mulher

☐Gaja

☐Racismo

☐Racista

☐Raça

☐Nação

☐Migrante

☐Imigrante

☐Emigrante

☐Estrangeiro

☐Estrangeira

☐Classismo

Add keyword or expression

Add

Title

Subtitle

Owner

CMC Source Text Type

Language

Choose your option

Date posted

dd/mm/aaaa

Figura 4: Formulário de submissão de um ficheiro- 1ª Parte

## 8. RESULTADOS

---

Date Extraction

dd/mm/aaaa

Source Type

URL

Type of online platform/channel

Choose your option

Post Text

Comments Permanently Open

☒ Yes

☐ No

Submit

Figura 5: Formulário de submissão de um ficheiro- 2ª Parte

Analyzer Results	
 Download Stats CSV	
./public/uploaded/Youtube_extraction_portuguese_2.json	
gender	
Ugy8CpxWIBVFrGMvGE94AaABA	--> ('Machismo', 1, 6)
UgxkiC9aoMHltahduB4AaABA	--> ('Machista', 1, 10)
UgzMmRFYadWylzXN8st4AaABA	--> ('Machista', 2, 59)
UgwUr_cVJjhxNVla_zJ4AaABA	--> ('Machismo', 1, 31)
UgxITN4B96vW3IDoxcJ4AaABA	--> ('Mulher', 1, 101)
UgyoHtMRKZsmO5XTbxt4AaABA	--> ('Machismo', 1, 45)
UgxeQgq_khjuffKm5rJ4AaABA	--> ('Machismo', 1, 14)
UgzLgR8VzOyV_pcw3K14AaABA	--> ('Machismo', 1, 13)

Figura 6: Formulário de submissão de um ficheiro- Resultados do Analisador

## 8. RESULTADOS

### ◇ Formulário da submissão de *keywords*

Na figura 7, podemos observar a interface que permite a inserção de uma nova *keyword* pela parte do utilizador. Como podemos verificar a página está dividida em duas zonas, em que uma se trata das *keywords* na língua portuguesa e a outra das *keywords* em língua Inglesa. É também possível verificar as *keywords* já existentes de forma a verificar que não existem sobreposições da mesma.

No router desta página já será indicado qual o tipo de preconceito e variável sociolinguística que pretendemos adicionar a *keyword*.

The screenshot shows the 'Adicionar Keywords' page. At the top, there's a navigation bar with 'Hara mbe' logo and links: 'Histórico de Análises', 'Adicionar Keywords', 'Submeter Ficheiros', 'Manual de atualização', and 'About'. The main title is 'Adicionar Keywords'. Below it, there are two sections: 'PT' and 'EN'. Each section contains a 'Variável Sociolinguística' dropdown menu (currently set to 'gender') and a 'Keywords' input field with a 'Click here' button. Below these are 'Add keyword or expression' input fields and 'Add' buttons. A 'Submit' button is located at the bottom left of the PT section.

Figura 7: Formulário de submissão de uma keyword

### ◇ Visualização de um *post* previamente analisado/inserido

Após a inserção através do formulário de submissão de ficheiros, cada um dos ficheiros inseridos pode ser visualizado no histórico de análises.

The screenshot shows the 'Ficheiros' page. At the top, there's a navigation bar with 'Hara mbe' logo and links: 'Histórico de Análises', 'Adicionar Keywords', 'Submeter Ficheiros', 'Manual de atualização', and 'About'. The main title is 'Ficheiros'. Below it, there's a search bar labeled 'Procurar...'. A list of files is displayed. The first file is 'O Interrogatório a Fernanda Cândia Ex Namorada de José Sócrates - Especial CMTV - 22 Abril 2018'. It includes details like 'Inserida por - canal de desporto', 'Plataforma - YouTube', 'URL - https://www.youtube.com/watch?v=3t43cWTYRX0&t=', 'Publicado a 21/04/2018', and 'Data de Extração - 2019-01-31'. There is an eye icon and a trash icon next to the file.

Figura 8: Visualização de um *post* previamente analisado/inserido

## 9 Conclusão

Concluído o trabalho e voltando a evidenciar que o principal objetivo deste projeto consistia na redução da arduidade que a tarefa de obtenção da meta-informação por meios manuais representa, podemos afirmar que o trabalho realizado vai de encontro às expectativas iniciais do grupo. Apesar de o *pré-processamento* não conseguir capturar todas as ocorrências de comentários de caráter ofensivo e discriminatório, uma vez que alguns utilizadores podem usar comentários mais subtis que não conseguem ser identificados pelo analisador, é possível aos utilizadores descarregar o documento *json* referente a um *post*, permitindo assim que o utilizador possa identificar com uma leitura mais cuidada e humana os comentários ofensivos.

Relegando para trabalho futuro o controlo de utilizadores, uma vez que é um ponto que não se enquadra no âmbito do projeto, o grupo acredita que era um aspeto que iria melhorar a experiência do utilizador dado que a cada utilizador estariam associadas apenas os ficheiros analisados pelos mesmos, contrariamente à coletânea de todos os ficheiros que já foram analisados comuns a qualquer utilizador da aplicação atual.

De forma a findar este relatório, gostaríamos de alegar que os desafios que o projeto apresentou, nomeadamente na parte da análise lexical e aplicação de algumas funcionalidades à nossa aplicação, potenciaram a nossa aptidão de manuseamento das ferramentas utilizadas durante o trabalho e promoveu um nível de compromisso elevado pelo mesmo durante toda a sua realização, visto que é um tema muito interessante, atual e evidente em qualquer rede social.

## **Referências**

[1]

## 10 Anexos

Nesta secção será apresentado em primeiro lugar o código referente ao analisador desenvolvido pelo grupo responsável por fazer a análise quando um documento é submetido na aplicação e em seguida um extrato de um documento exemplar do tipo de ficheiros que são submetidos na aplicação. Desta forma passamos a apresentar o conteúdo do *analisadorJSON-v6.py*:

```
#!/usr/bin/python3
##!/usr/local/bin/python3

import json,sys,xlsxwriter,os,glob
import re
from fuzzywuzzy import fuzz
from fuzzywuzzy import process
import pymongo

#class Post:
    #def __init__(newObject,postid,arrayComments,ocur):
        #newObject.post_id=postid
        #newObject.arrayComments=arrayComments
        #newObject.occurrences=ocur

#Classe Comentario
#comment_id : ID do comentário
#commentMessage : Texto do Comentario
#user : ID do utilizador
#occurencias : Objeto ocorrencias que permite guardar as ocorrencias de
↳ cada palavra reservada do respetivo comentario
class Comentario:
    def __init__(newObject,comment_id,commentMessag
,user,ocur):
        newObject.comment_id=comment_id
        newObject.commentMessage=commentMessage
        newObject.user=user
        newObject.occurrences=ocur

#Função que extrai informação dos ficheiros de comentarios para um array
def loadInfoExtract(inventory,com_id,com_txt,com_user):
    arrayComments = []
    i=0
    for item in inventory["commentThread"]:
        arrayComments.append(Comentario(item[com
_id],item[com_txt],item[com_user],[]))
        i+=1
    return arrayComments
```



```
#Função de inicialização do python para tratar ficheiro JSON
def loadInfo(file):
    info = open(file).read()
    inventory = json.loads(info)
    return inventory

#Função recursiva que trata de guardar para um array todos os valores do
↪ tipo lista de um objeto
def parseValues(objeto):
    lista = []
    for key in objeto.keys():
        if type(objeto[key]) is list:
            lista += (key,objeto[key])
        else:
            lista += parseValues(objeto[key])
    return lista

#Função que dado um preconceito e o inventário de keywords
#retorna um triplo com o tipo de preconceito, variavel sociolinguistica e
↪ as palavras reservadas (keywords)
def loadKeywordsRecAux(inventory,prejudice):
    value = ()
    parsedValues = ()
    for items in inventory:
        if (items['type_prejudice'] == prejudice):
            parsedValues = parseValues(items
            ['sociolinguistic_variables'])
            var_sociol = parsedValues[0]
            kws = parsedValues[1]
            value = (prejudice, var_sociol,kws)
    return value

#Função global, que dada a escolha do utilizador, se auxilia na
↪ função loadKeywordsRec
#para retornar a lista de tuplos de preconceitos-keywords
def loadKeywordsRec(inventory):
    prejudices = ["Sexism","Ageism","Racism","Nationalism",
    "Classism","Homophobia","Anti-Clericalism",
    "Body-Shaming","Addiction-Shaming",
    "Ideological-Shaming"]
```

```

    prej_sociol_kws_triples = []
    for prejudice in prejudices:
        prej_sociol_kws_triples.
            append(loadKeywordsRecAux(inventory
            ,prejudice))
    return prej_sociol_kws_triples

#Função Obsoleta
def loadKeywords(inventory,keyword):
    arrayKeywords = []
    i=0
    for items in inventory:
        if(items['type_prejudice'] == keyword):
            for values in items['Sociolinguistic variables']
                .values():
                    print(values)
                    arrayKeywords=values

    return arrayKeywords

#Função que realiza o trabalho de procura da string da Keyword na string
↪ do texto do comentário
#Caso encontre, cria um triplo com a keyword, o numero
de occorencias e o total de palavras do comentario em questao
def checkNcount(comentario,keywords):
    occurrences = []
    #print(keywords)
    nOcur = 0
    wordcount = 0
    for keyword in keywords:
        """
        ↪ #contabilizar o numero de occurencias da keyword dentro do
        ↪ comentario
        nOcur = len(re.findall(r"\b"+keyword+r"\b",
        comentario.commentMessage,re.I))
        if(nOcur > 0):
            #contabilizar o numero total de palavras
            wordcount = len(comentario.commentMessage
            .split())
            value = (keyword, nOcur, wordcount)
            occurrences.append(value)
            #print(comentario.commentMessage)
        """

```

```
        words = comentario.commentMessage.split()
        for word in words:
            if fuzz.ratio(word,keyword) >= 75:
                n0cur += 1
    if(n0cur > 0):
        #contabilizar o numero total de palavras
        wordcount = len(comentario.commentMessage
                        .split())
        value = (keyword, n0cur, wordcount)
        occurrences.append(value)
        #print(comentario.commentMessage)
        n0cur = 0
        wordcount = 0

    return occurrences

#Função que retorna um tuplo com o preconceito e um array de comentarios
# onde existe a ocorrencia desse preconceito
#
#Nesse array de comentarios, estes são carregados com a
#informação das suas ocorrencias no elemento ocorrencias
#da classe comentario
def lexicalAnalysisAux(comentarios,var_sociol,keywords):
    occurrences = []
    arraycoments = []
    #totalWordCount = 0
    for comentario in comentarios:
        #totalWordCount += len(comentario.commentMessage.split())
        occurrences = checkNcount(comentario,keywords)
        if occurrences:
            comentario_copia = Comentario
            (comentario.comment_id,comentario.
             commentMessage,comentario.user,[])
            comentario_copia.occurrences = occurrences
            arraycoments.
            append(comentario_copia)
```

```

    #Array de comentarios, em que cada um possui um array de
    ↪ ocorrencias
    var_sociolPost = (var_sociol,arraycoments)
    #print(var_sociolPost)
    return var_sociolPost

#Função que retorna um array com todos os tuplos
    ↪ variavel_sociol-comentarios
#resultantes da função auxiliar lexicalAnalysisAux
def lexicalAnalysis(comentarios, prej_sociol_kws_triples):
    var_sociolsPost = []
    for triple in prej_sociol_kws_triples:
        var_sociol = triple[1]
        keywords = triple[2]
        var_sociolsPost.append(lexicalAnalysisAux
                               (comentarios,var_sociol,keywords))
    return var_sociolsPost

#Função que retorna os tuplos de ocorrencias de keywords para o post
#Basicamente soma as ocorrencias dos comentarios para gerar as do post
def getPostOcur(var_sociolsPost):
    postOcur = []
    for prejComents in var_sociolsPost:
        for comentario in prejComents[1]:
            postOcur.extend(comentario
                           .occurrences)
    my_set = {x[0] for x in postOcur}
    postOcur = [(i,sum(x[1] for x in postOcur if x[0] == i)) for i in
                ↪ my_set]
    return postOcur

#Função de pretty printing para debbuging
def printOcorrencias(var_sociolsPost):
    str = ""
    for prejComents in var_sociolsPost:
        str += prejComents[0]
        str += "\n"
        for comentario in prejComents[1]:
            str += comentario.comment_id + " --> "
            for value in comentario.occurrences:
                str += "{}".format(value)
            str += "\n"
        str += "\n"

    print(str)

```

```
#Função que desenha o excell com as estatísticas
#Frequências relativas, absolutas e totais dos comentarios para cada post
def excelWriter(var_sociolsPost,nComents,totais,
worksheetName,workbook, file_name):
    #variaveis
    tam= len(comentarios[1])+3
    total_linhas = 0
    final=0
    final_id=0
    corrente = 4

    worksheet = workbook.add_worksheet(worksheetName)

    #Parte estatica
    bold = workbook.add_format({'bold': True})
    princ = workbook.add_format({'bold':
    ↪ True,'font_color':'white','font_size':'14',
    'bg_color':'green'})
    pre = workbook.add_format({'bold':
    ↪ True,'font_color':'black','font_size':'10',
    'valign': 'vcenter','align':'center',
    'border_color':'black'})
    worksheet.write('B1', 'Ficheiro',princ)
    worksheet.merge_range('C1:E1',file_name,pre)
    worksheet.write('B3', 'Prejudice',princ)
    worksheet.write('C3', 'Comentario',princ)
    worksheet.write('D3', 'ID',princ)
    worksheet.write('E3', 'Frequencia',princ)
    worksheet.write('G3', 'Total',princ)

    #Parte dinamica
    for prejComents in var_sociolsPost:
        tamC = len(prejComents[1])
        tam = tamC + corrente
        tam_str = str(tam-1)
        curr_str = str(corrente)
        if(tamC>1):
            worksheet.merge_range('B'+curr_str
            ↪ ':B'+tam_str,prejComents[0],pre)
        else:
            worksheet.write('B'+curr_str,
            ↪ prejComents[0],pre)
        for comentario in prejComents[1]:
            total_linhas += 1
            curr_str = str(corrente)
            maior_mm = len(comentario.commentMessage)
            maior_id = len(comentario.comment_id)
```

```

        if(maior_mm > final):
            final = maior_mm
        if(maior_id > final_id):
            final_id = maior_id
        worksheet.write('C'+curr_str,
            ↪ comentario.commentMessage)
        worksheet.write('D'+curr_str,
            comentario.comment_id)
        corrente+=1
        worksheet.write('E'+curr_str,
            str(comentario.occurrences))

worksheet.set_column('B:D',len('prejudice..'))
worksheet.set_column('C:D',final)
worksheet.set_column('D:E',final_id)

worksheet.write('I3','Ocorrencias',princ)
worksheet.set_column('I:J',20)

counter=4
for info in totais:
    str_counter=str(counter)
    worksheet.write('I'+str_counter,
        str(info[0])+ ' ----> '+str(info[1]))
    counter+=1

worksheet.merge_range('G4:G'+str(total_linhas+3)
    ,str(total_linhas)+'/'+str(nComents),pre)
print('terminei')

#Função que desmembra o array var_sociolsPost em prejsKW
# (array que representa as keywords por preconceito)
def kwNprej(var_sociolsPost):
    prejsKW = {}
    for prej,com in var_sociolsPost:
        if com:
            prejsKW[prej] = []
            for elem in com:
                ocur = elem.occurrences
                for item in ocur:
                    if item[0] not in prejsKW[prej]:
                        prejsKW
                        [prej]
                        .append
                        (item[0])

    #print(prejsKW)
    return prejsKW

```

```
#Função de criação do objeto JSON
def jsonMetadataWriter(var_sociolsPost):
    json_obj = {
        "fname": "",
        "cmc": "",
        "lang": "",
        "date_p": "",
        "date_e": "",
        "title_type": "",
        "url_type": "",
        "setting": "",
        "platform": "",
        "svs": [],
        "kws": [],
        "extract_file_type": "",
        "source_type": "",
        "cpo": ""
    }
    prejsKW = kWnpj(var_sociolsPost)

    for key, value in prejsKW.items():
        json_obj['svs'].append(key)
        for elem in value:
            json_obj['kws'].append(elem)

    print(json_obj)
    with open('metadata.json', 'w') as outfile:
        json.dump(json_obj, outfile, indent=4, ensure_ascii=False)

#Função geral
def main():
    file_path = sys.argv[1]

    myclient = pymongo
    .MongoClient("mongodb://localhost:27017/")
    mydb = myclient["harambe"]
    mycol = mydb["keywords"]

    # criação do inventário das keywords
    kw_inventory = []
    #x = mycol.find()
    for x in mycol.find():
        kw_inventory.append(x)

    # criação do xlsx
    workbook = xlsxwriter.Workbook('resultado.xlsx')

    counter = 0
```

```
print('A analisar ficheiro.....')
##Fazer análise do ficheiro escolhido
print(file_path)

com_inventory = loadInfo(file_path)
comentarios = loadInfoExtract(com_inventory, 'id', 'commentText',
    ↪ 'user')
prej_sociol_kws_triples = loadKeywordsRec(kw_inventory)

var_sociolsPost = lexicalAnalysis(comentarios,
    ↪ prej_sociol_kws_triples)
printOcurrencias(var_sociolsPost)
totais = getPostOcur(var_sociolsPost)

jsonMetadataWriter(var_sociolsPost)

nComents = len(comentarios)
excelWriter(var_sociolsPost, nComents, totais, f"sheet{counter}",
    ↪ workbook, file_path)

#counter += 1

#fechar o workbook
workbook.close()
myclient.close()

main()
```



Em seguida é apresentado um excerto do tipo de ficheiros **JSON** que são submetidos na aplicação para posterior análise.

```
{
  "header": {
    "title": "  O Interrogat\u00f3rio a Fernanda C\u00e2ncio Ex
    ↵ Namorada de Jos\u00e9 S\u00f3crates - Especial CMTV - 22
    ↵ Abril 2018 ",
    "subtitle": " NA ",
    "owner": " canal de desporto ",
    "views": " 36 478 visualiza\u00e7\u00f5es ",
    "likes": " 139 ",
    "dislikes": " 29 ",
    "shares": " NA ",
    "datePosted": " Publicado a 21/04/2018 ",
    "dateExtraction": " 2019-01-31 ",
    "language": " pt ",
    "plataform": " YouTube ",
    "url": " https://www.youtube.com/watch?v=3t43cWTYRX0&t= ",
    "postText": " ",
    "numberPosts": " 59 ",
    "srcType": " video ",
    "nameNewspaper": " NA ",
    "socioLingVar": " ",
    "listEvents": " ",
    "articleKeywords": " NA ",
    "keywords": " ",
    "commentsOpen": " yes ",
  },
  "commentThread": [
    {
      "id": "UgxhC6A5z1RTUk-y2A94AaABAg",
      "user": "Deolinda Martins",
      "date": "1 week ago",
      "timestamp": 1548344819936,
      "commentText": "A gaja n\u00e3o \u00e9 burra de todo",
      "likes": 0,
      "hasReplies": false,
      "numberOfReplies": 0,
      "hasKW": 0
    }
  ]
  ...
}
```

```
...
{
  "id": "Ugz2i79bEUTSPrxLoF4AaABAg",
  "user": "Hraki JAH1",
  "date": "8 months ago",
  "timestamp": 1527781619958,
  "commentText": "E lamentavel as sanguessugas que chupam a veia
↳ da nossa p\u00e9ltria. Sem ofensa (E um gajo aqui a tentar
↳ sobreviver com a mis\u00e9ria do ordenado m\u00e9dimo)
↳ ate o bruno de carvalho tem mais visualiza\u00e7\u00f5es
↳ a dar um peido nas antas. Como lavam os olhos com futebol
↳ e novelas.",
  "likes": 3,
  "hasReplies": true,
  "numberOfReplies": 1,
  "replies": [
    {
      "id": "Ugz2i79bEUTSPrxLoF4AaABAg
      .8gXEkr4Yc8b8hEoANdYEB8",
      "user": "N\u00e9dia Gomes",
      "date": "7 months ago",
      "timestamp": 1530373620297,
      "commentText": "Bem dito.",
      "likes": 1,
      "hasKW": 0
    }
  ],
  "hasKW": 0
},
...
```