UNIVERSIDADE
LUSÓFONA

Análise automática de mensagens de texto do Twitter

Trabalho Final de curso

Afonso Martins Marques
Prof. Dr. Manuel Marques Pita
Trabalho Final de Curso | LEI | 26/06/2020

Direitos de cópia

Índice

Índice de Figuras

Índice de Tabelas

Resumo

A cada dia que passa a quantidade de utilizadores de redes sociais está a aumentar. Neste momento 59% da população mundial é ativa nas redes sociais, sendo já, mais de 4.6 mil milhões de pessoas[1].

Este trabalho tem como objetivo analisar mensagens de texto em microblogs (*tweets*) com o intuito de aplicar técnicas de inteligência artificial de forma a ser possível identificar e anotar elementos que permitam a sua classificação em várias dimensões, como por exemplo tópicos e/ou sentimento [1]*Visual Sentiment Analysis on Twitter Data Streams*. A classificação automática de tweets poderá ajudar investigadores, jornalistas e cidadãos no geral na análise das dinâmicas complexas de conversa social para tentar identificar, por exemplo, vulnerabilidades, ideias que têm alta ressonância, capacidade viral, padrões, grupos, entre outros.

A plataforma utilizada para a recolha dos dados será o *Twitter*, visto que o mesmo é um sistema de 'microblogging' onde as pessoas comunicam através de pequenas mensagens (*tweets*). Outras plataformas como o Facebook e Instagram não tem acesso aberto aos seus dados públicos. Para a recolha de informação iremos criar uma biblioteca de acesso à plataforma Twitter de forma a extrairmos informações relevantes, relativas a mensagens de texto de interesse. Estes dados serão posteriormente guardados numa base de dados que irá ser acedida por uma aplicação cliente servidor. Esta irá apresentar os dados processados ao utilizador.

Este trabalho é deveras importante uma vez que irá permitir relacionar uma grande quantidade de conceitos teóricos com conceitos práticos.

Palavras chave: Ciência dos Dados, Inteligência Artificial, Twitter, Conversa social, noticias falsas, análise de sentimento.

[1] https://www.statista.com/statistics/617136/digital-population-worldwide/

Abstract

Every day that goes by, the number of people that use social networks increase. At this moment 59 percent of the world's population is active on social networks, with over three billion people.

The purpose of this work is to analyse text messages in micro-blogs (tweets) in order to apply artificial intelligence techniques with a view to be able to identify and annotate elements that allow their classification in various dimensions, such as example topics and / or sentiment [1]*Visual Sentiment Analysis on Twitter Data Streams*. The automatic classification of tweets may help researchers, journalists and citizens in general to analyse the complex dynamics of social conversation to try to identify, for example, vulnerabilities, ideas that have high resonance, viral capacity, patterns, groups, among others.

We are going to use twitter as the main platform. Twitter is a system of 'microblogging' where people communicate with short messages (tweets). Other platforms like Facebook and Instagram do not have open access to their public data. To collect information, we will build a library to access twitter in order to extract relevant information, including conversations parametrized by elements such as hashtags, keywords and others. This data will later be stored in a database which will be accessed by a client server application. In this application we are going to present processed data.

This work is very important since it will allow to relate theoretical concepts with practical concepts.

Keywords: Data Science, Artificial Intelligence, Twitter, Social Conversation, Fake news, sentiment analysis.

1. IDENTIFICAÇÃO DO PROBLEMA

Uma das áreas de maior interesse científico neste momento está relacionada com o estudo de interações em redes sociais. Em particular no efeito que o consumo de informação nestes meios pode ter em decisões individuais como por exemplo opções de voto em eleições, a compra de produtos, e como são valorizados certos aspetos relacionados com o ambiente e cultura. Por outro lado, as redes sociais têm permitido a difusão de notícias falsas e discursos de odio numa escala sem precedentes tal como foi visto em [2]*Rumor Detection and Classification for Twitter Data*.
Um outro problema é a dificuldade em encontrar fontes de informação que sejam confiáveis, para alem disso também é bastante difícil descobrir plataformas que façam a captura de dados, a análise e a apresentação dos resultados obtidos.

Neste TFC abordamos este problema através do estudo de mensagens de texto sobre temas específicos na plataforma *Twitter,* apresentado posteriormente o resultado desse estudo.

OBJETIVOS

Numa primeira fase do projeto, o objetivo foi aprender a consultar o API de *Twitter* respeitando limites de acesso e usando metodologias padrão na Ciência dos Dados que permita atingir os primeiros objetivos de aprendizagem deste TFC:

(O1) Aceder à plataforma *Twitter* e retirar dados específicos usando os padrões praticados pelos cientistas de dados, assim como respeitando os limites preestabelecidos por *Twitter*.

(O2) Construir uma biblioteca de acesso à plataforma *Twitter* para extrair informação relevante a mensagens de texto de interesse parametrizadas com elementos, tais como palavras chave, *hashtags,* entre outros e transferir esta informação para um repositório estruturado (base de dados relacional clássica em *PostgreSQL*, ou alternativa como por exemplo *MongoDB*).

(O3) Construir uma aplicação cliente servidor (web) que permita consultar a base de dados de *tweets*, e fazer estatísticas básicas.

Numa segunda fase, o foco passa a ser na análise de textos com técnicas de processamento estatístico de linguagem natural. O primeiro objetivo é extrair elementos informativos dos textos como por exemplo frases chave, sentimento e tópicos.

2. LEVANTAMENTO E ANÁLISE DOS REQUISITOS

Alguns requisitos funcionais identificados tanto para a Aplicação web em FLASK (descrita abaixo) como para a aplicação CISN (descrita abaixo) foram:

Área	**Nome**	**ID**	**Descrição do Requisito**	**Categoria**	**Estado**
1. CISN	1.1. Gestão de Funcionalidades	1.1.1	Captura de *tweets* em modo *archive.*	Recolha	Implementado
		1.1.2	Captura de tweets por hashtags.	Recolha	Implementado
		1.1.3	Captura de tweets por palavras.	Recolha	Implementado
		1.1.4	Captura de tweets por período de tempo.	Recolha	Implementado

	1.1.5	Respeitar limites do Twitter	Recolha	Implementado
	1.1.6	Reduzir o tweet para os campos necessários	Processamento	Implementado
	1.1.7	Processar/limpar texto do tweet	Processamento	Implementado
	1.1.8	Analisar sentimento de tweets	Processamento	Implementado
	1.1.9	Analisar subjetividade de tweets	Processamento	Implementado
	1.1.9	Validar se origem é proveniente de um Bot	Validação	Implementado
	1.1.10	Guardar tweets em formato JSON	Armazenamento	Implementado
	1.1.11	Inserir tweet na base de dados	Armazenamento	Implementado
	1.1.12	Criar uma base de dados	Armazenamento	Implementado
	1.1.13	Implementar um novo algoritmo	Processamento	Não implementado

Tabela 1 - Requisitos Funcionais: CISN

Área	Nome	ID	Descrição do Requisito	Categoria	Estado
2. Aplicação Cliente - Servidor	2.1. Gestão de Funcionalidades	2.1.1	Apresentar tweet com mais likes	Apresentação	Implementado
		2.1.2	Apresentar localização dos tweets	Apresentação	Implementado
		2.1.3	Apresentar *word cloud* dos sentimentos analisados	Apresentação	Implementado
		2.1.4	Apresentar percentagem de match	Apresentação	Implementado
		2.1.5	Fazer registo com email	Registo	Implementado
		2.1.6	Fazer Login com email	Identificação	Implementado
		2.1.7	Filtrar por temas existentes	Pesquisa	Implementado
		2.1.8	Filtrar por querie	Pesquisa	Parcialmente implementado
		2.1.9	Pedir dados à base de dados	Pesquisa	Parcialmente

Tabela 2 - Requisitos Funcionais: Aplicação Cliente-Servidor

Área	Nome	ID	Descrição do Requisito	Categoria	Estado
2. Aplicação Cliente - Servidor	2.2. Requisitos não funcionais	2.2.1	Facilidade de utilização	Usabilidade	Implementado
		2.2.2	Confiabilidade nos resultados	Confiabilidade	Implementado

Tabela 3 - Requisitos Não Funcionais: Aplicação Cliente-Servidor

Desde o planeamento inicial o ponto 1.1.12 (Criar uma base de dados). sofreu uma alteração uma vez que a base de dados foi construída na *Firebase* em vez de ter sido construída em *PostgreSQL* ou *MongoDB*. Apos temos feito esta alteração percebemos que a mesma tinha uma desvantagem que mais à frente nos condicionou. Esta limitação é referente à capacidade de fazer *queries*, sendo esta bastante limitada.

O ponto 1.1.9 (Validar se origem é proveniente de um Bot) está totalmente implementado e pronto a ser utilizado, no entanto não está ativo devido à demora na resposta da API.

Por fim o ponto 2.1.8 (Filtrar por querie) está parcialmente implementado devido à desvantagem existente na base de dados escolhida.

3. VIABILIDADE E PERTINÊNCIA

1. O trabalho a ser desenvolvido está alinhado às práticas comuns na Ciência dos Dados. Por outro lado, a fonte de acesso aos dados (Twitter) está sustentada por um API padrão completamente documentada.

2. O sistema permite a análise do sentimento e subjetividade de qualquer tema disponível na base de dados.

3. Os algoritmos foram implementados com uma abordagem de código fonte aberto, disponibilizados no Git ou BitBucket (após publicação de resultados).

Apos a conclusão e apresentação desde trabalho, existe a possibilidade do mesmo ser continuado uma vez toda a parte da captura de tweets e processamento está feita, para alem disso também tem a aplicação cliente-servidor que pode ser evoluída permitindo assim apresentar mais dados. Deste modo, poderia forcar-se apenas na análise dos tweets sem se ter que preocupar com todo o processo que está para trás, para isso apenas necessita de pedir as suas credenciais.

4. SOLUÇÃO DESENVOLVIDA

Durante a execução do TFC, percorremos/executamos os seguintes passos:

1. Plataforma desenhada e implementada em Python para acesso ao Twitter API através de bibliotecas já existentes;
2. Desenho e implementação de sistema, em Python, para parametrizar mensagens de texto de interesse através de *hashtags*, palavras chave e outras variáveis;
3. Processamento de objetos JSON (formato usado para representar tweets) para transformar dados recebidos em dados que possam ser assimilados dentro da base de dados do projeto (*Firebase*). Guardando apenas as partes mais importantes, uma vez que se fosse guardado na sua

totalidade, grande parte da informação iria ser desperdiçada para alem de ocupar muito espaço;

4. Criação de uma aplicação em Flask cliente-servidor na web para consulta dos conteúdos na base de dados;
5. Adicionamos à aplicação cliente servidor elementos que permitem visualizar estatísticas básicas usando, usando a biblioteca *gstatic*[3] da google;
6. Utilização da API da *google maps* para mostrar as cidades onde os tweets foram efetuados;
7. Usamos bibliotecas de processamento estatístico de linguagem natural para pré-processar texto, e retirar caraterísticas dos textos em várias dimensões: subjetividade, sentimento;

Iremos processar Tweets que estejam na língua inglesa[4] uma vez que esta corresponde a trinta e quatro porcento dos tweets [3]*Vicinitas*, sendo mais do dobro da segunda linguagem mais falada (japonês com dezasseis porcento). Uma vez que é a língua mais falada, vamos ter mais dados do que em qualquer outra, o que é bom pois precisamos de uma grande amostra para podemos tirar conclusões. Mesmo estando a escolher inglês como linguagem principal, não nos estamos apenas a focar num nicho de utilizadores, uma vez que possivelmente vamos poder generalizar os resultados para outras linguagens. Numa investigação futura seria interessante verificar se a afirmação anterior é verdadeira havendo mesmo uma correlação.

No nosso trabalho escolhemos o Twitter como plataforma de eleição uma vez que tem um API[5] de acesso, permitindo-nos assim obter informação através de ficheiros *JSON* como foi dito anteriormente. Escolhemos também esta

[3] https://developers.google.com/chart/interactive/docs/basic_load_libs
[4] https://mashable.com/2013/12/17/twitter-popular-languages/?europe=true
[5] https://developer.twitter.com/en/docs/tweets/search/api-reference

plataforma uma vez os dados que a mesma nos deixa analisar são bastante relevantes para as métricas que iremos utilizar.

Este trabalho poderá ser aplicado a outras plataformas, no entanto seria necessário fazer algumas adaptações, tendo em conta restrições da plataforma, tipos de utilizador e objetivos de estudo. Para o nosso caso, a plataforma que nos irá beneficiar será o *Twitter*.

A arquitetura do nosso projeto irá consistir em três pedras basilares, sendo estas: Flask, Python, Firebase.

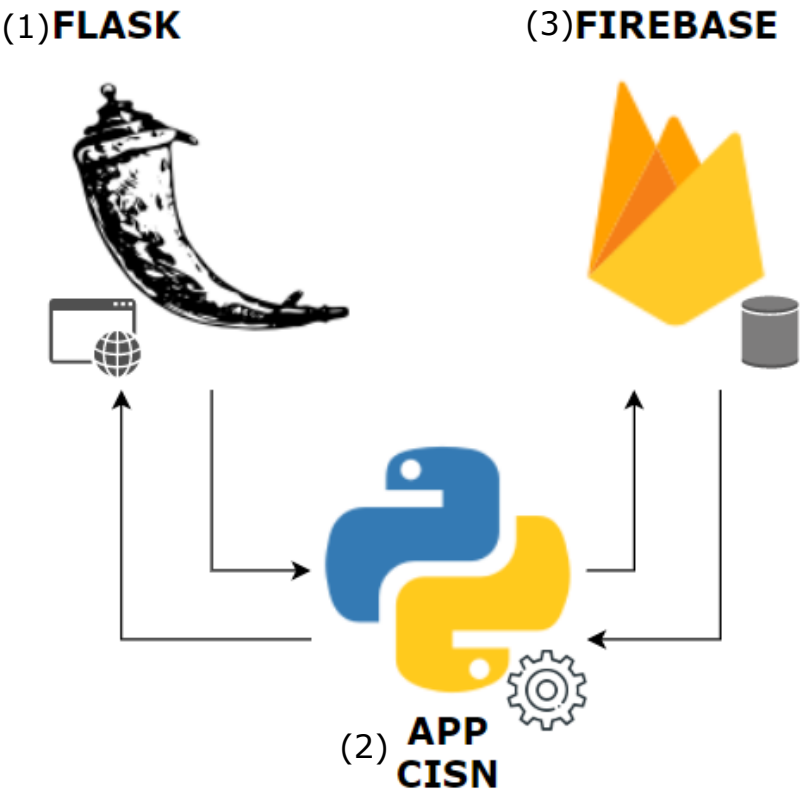

Figura 1 – Arquitetura do projeto

Vamos explicar o funcionamento de cada componente.

(1) Aplicação web em FLASK

Flask é uma pequena framework web escrita em *Python* e baseado na biblioteca *WSGI Werkzeug* e na biblioteca de *Jinja2.* Esta framework não tem uma camada de abstração para a base de dados, sendo assim

teremos que ser nós a desenvolver esta funcionalidade. É importante referir que o desenvolvimento desta aplicação não é principal objetivo deste trabalho, apenas é um meio para mostrar os resultados obtidos e também facilitar a analise de certos temas por pessoas que não são programadores.

Como já foi dito anteriormente esta aplicação tem o objetivo de "interagir" com os utilizadores permitindo que os mesmos façam pesquisas sobre temas existentes na nossa base de dados. Estas pesquisa poderão ser por palavras.

A nossa aplicação está assente na arquitetura *REST* tal como viemos em [4]Designing Consistent RESTful Web Service Interfaces.

REST significa *Representational State Transfer*, ou seja, transferência de estado representacional que é um estilo arquitetural que nos fornece padrões de sistemas computacionais na web, tornando assim mais fácil a comunicação entre sistemas. Normalmente sistemas que utilizam *REST* são *stateless*, ou seja, o servidor não mantem o estado do utilizador nem precisa de saber nada do cliente, separando assim o conceito de servidor e cliente.

Os inputs estão no formato padrão do protocolo HTTP (pares nome valor), tal como se fosse a submissão de um formulário HTML. Os outputs podem estar no formato XML ou JSON (no nosso caso estão em JSON).

Utiliza o protocolo HTTP sem extensões, ou seja, funciona em browsers e a maioria das linguagens inclui suporte nativo para estes webservices.

Na nossa aplicação não devemos mostrar os tweets na sua integra sem terem uma referência (link) para o mesmo, no entanto podemos fazer a sua análise e apresentar o resultado das mesmas, tal como vai ser explicado no próximo tópico. Estas análises serão mostradas em forma de *pie chart* no caso de ser análise de sentimento ou então em *word cloud* no caso de ser uma análise das palavras com mais peso nos tweets

referentes a um certo tema. Também mostramos ao utilizador a percentagem de tweets que fizeram match com a sua pesquisa, mostramos um mapa dos tweets cuja localização foi possível obter e por fim damos a hipótese ao utilizador de fazer o download dos tweets que foram analisados, no entanto apenas podemos fornecer os tweet Ids com as respetivas etiquetas analisadas. Não é possível fornecer o tweet completo uma vez que queremos seguir as políticas de partilha de dados impostas pelo Twitter.

A seguinte figura é referente ao formulário que o utilizador irá visualizar para fazer a sua pesquisa. Neste apenas terá que selecionar o tema pretendido, selecionar a subjetividade[6] dos tweets, colocar a intenção de pesquisa como explicamos anteriormente e por fim clicar em procurar.

Ao clicar em procurar irá ser gerado um pedido *POST* com o conteúdo do formulário.

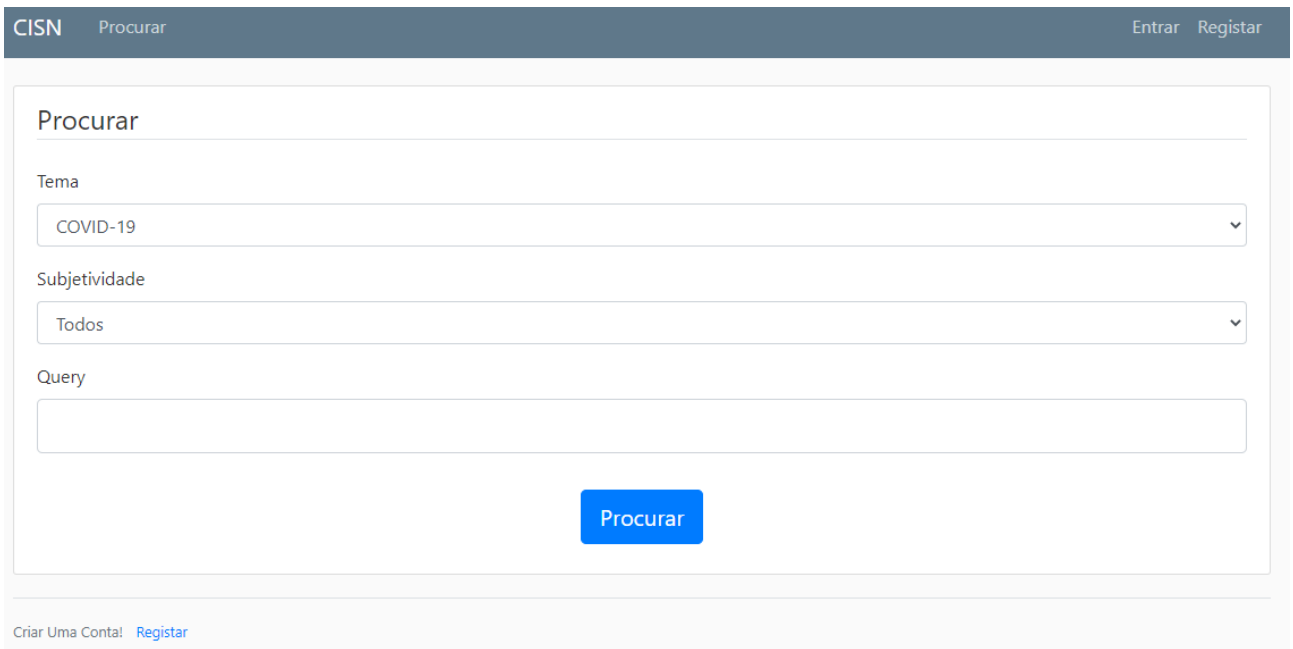

Figura 2 - Página de escolha do tema, subjetividade e palavras

[6] Subjetividade: Referente a tweets de opinião ou tweets factuais

Ao ser processado o pedido do utilizador, este começará por ver o tweet com mais likes, o que será representado na seguinte figura.

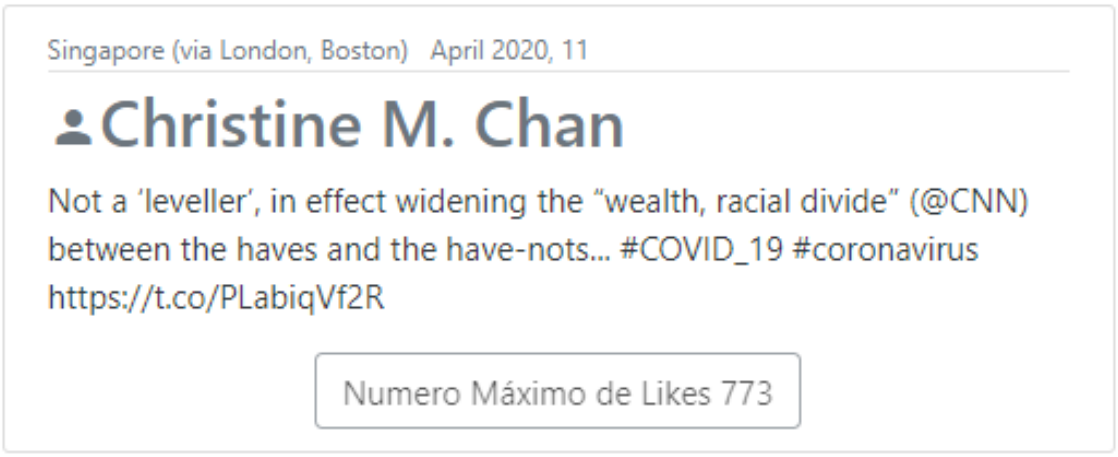

Figura 3 - Tweet com mais likes

Apos isto poderá ver a percentagem de tweets que fizeram *match* com os filtros escolhidos. Neste caso que vamos mostrar a percentagem foi 100%.

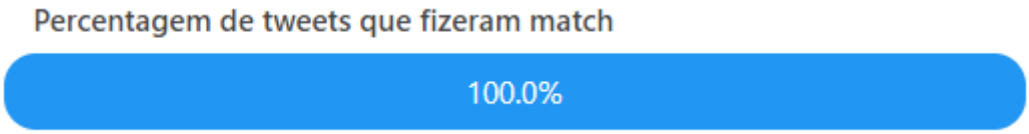

Figura 4 - Barra de percentagem de match

O cliente tem ainda a possibilidade de analisar a proveniência dos tweets, no entanto apenas damos a possibilidade de ver a cidade do mesmo por questões de privacidade. Pensamos que este mapa é uma mais valia uma vez que o utilizador poderá assim detetar padrões entre cidades, países e até continentes. A seguinte imagem representa o mapa apresentado ao utilizador. Como podemos ver este é bastante intuitivo não precisando assim de uma legenda.

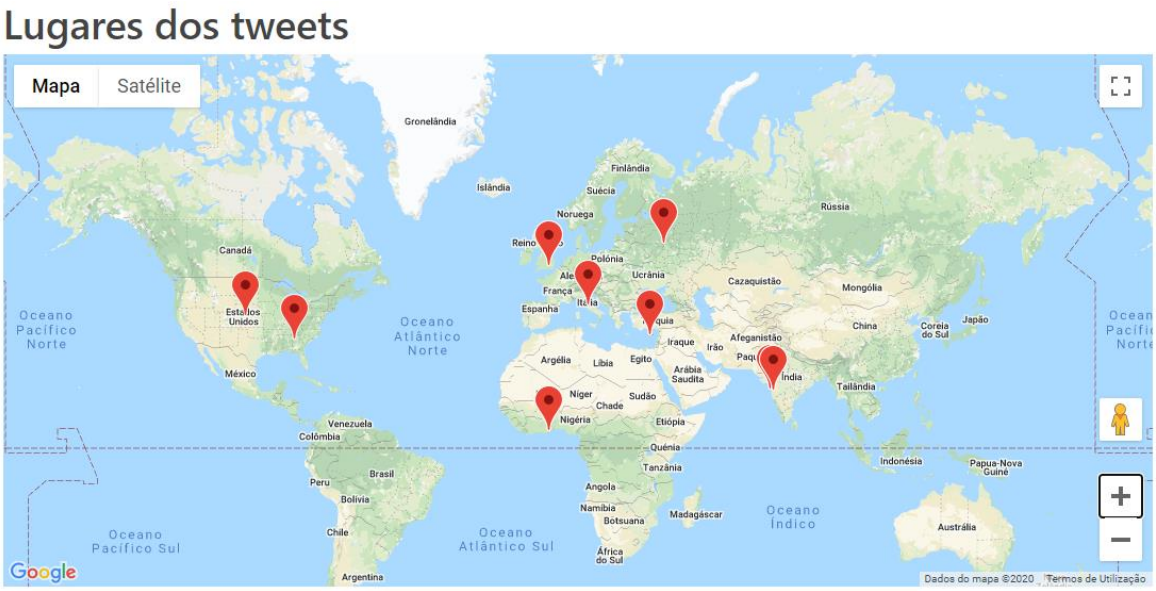

Figura 5 – Mapa com lugares dos tweets

A fim desta análise o utilizador irá ter acesso à análise de sentimento positivo e negativo em forma de *word cloud* referente ao tema que selecionou. A figura que se segue é um exemplo referente ao tema "COVID-19". Para serem mais explicitadas algumas palavras, vamos colocar de um dos lados da imagem algumas palavras que estão contidas no mesmo.

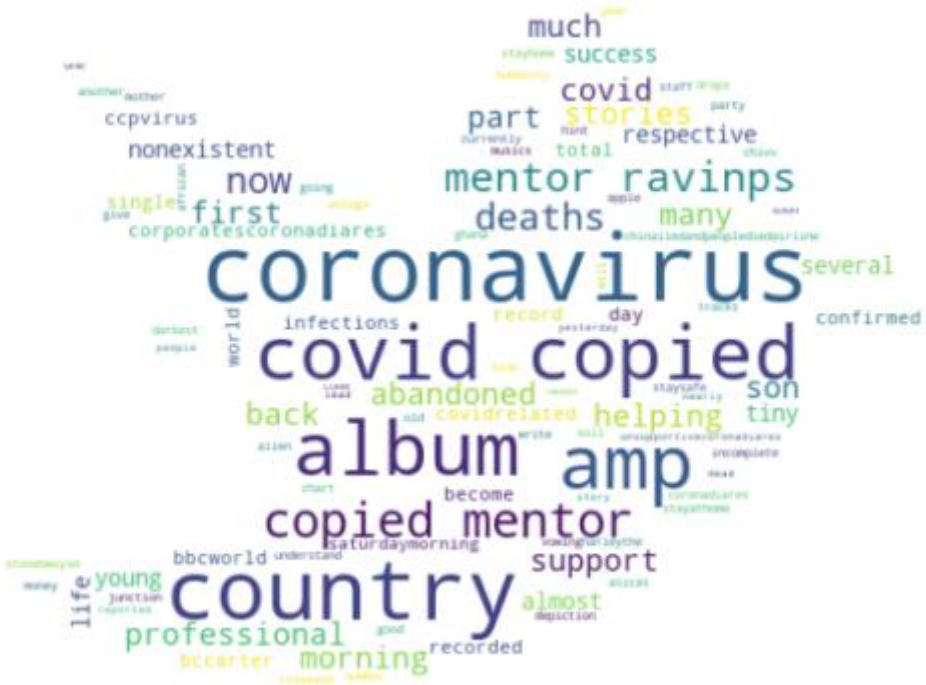

Figura 6 - Exemplo de uma Word Cloud

No próximo tópico iremos explicar o porque do seguinte resultado, no entanto facilmente percebemos que da nossa amostra 73.3% dos tweets analisados são positivos, 13.3% negativos e os restantes neutros. É importante referir que caso o utilizador passe com o cursor em cima de cada uma das percentagens irá ver a quantidade de tweets que correspondem a essa percentagem tal como podemos ver na seguinte imagem. Este *pie chart* é importante pois dá uma perceção geral do sentimento presente nos tweets analisados.

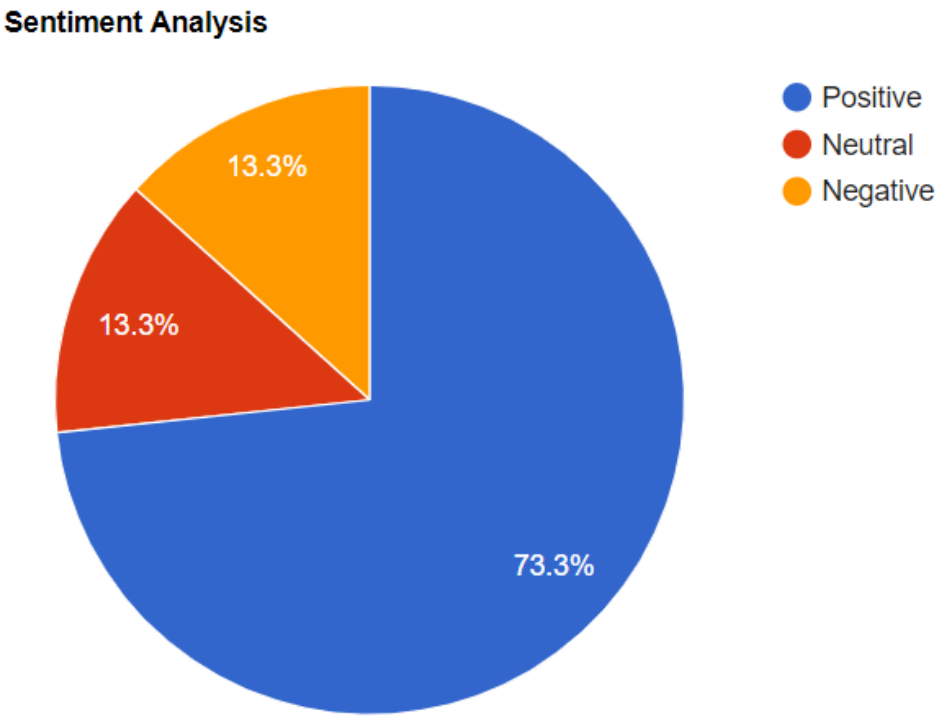

Figura 7 - Pie Chart de análise de sentimento

Neste projeto a criação de login e registo não era necessária, no entanto, existe uma grande probabilidade irmos necessitar das mesmas no futuro, quer seja para mostrar informações mais reservadas, quer seja para termos "controlo/ noção" dos nossos utilizadores.

Irei agora descrever cada uma destas funcionalidades começando pelo registo.

Na página de registo o utilizador irá encontrar um formulário onde terá que introduzir os seguintes campos: Nome de utilizador, email, password e confirmação da password. Esta página também nos oferece a possibilidade, caso o utilizador já tenha uma conta, de ir diretamente para a página de login através do link no canto inferior esquerdo.

Na seguinte figura, temos a possibilidade de ver tudo o que foi descrito.

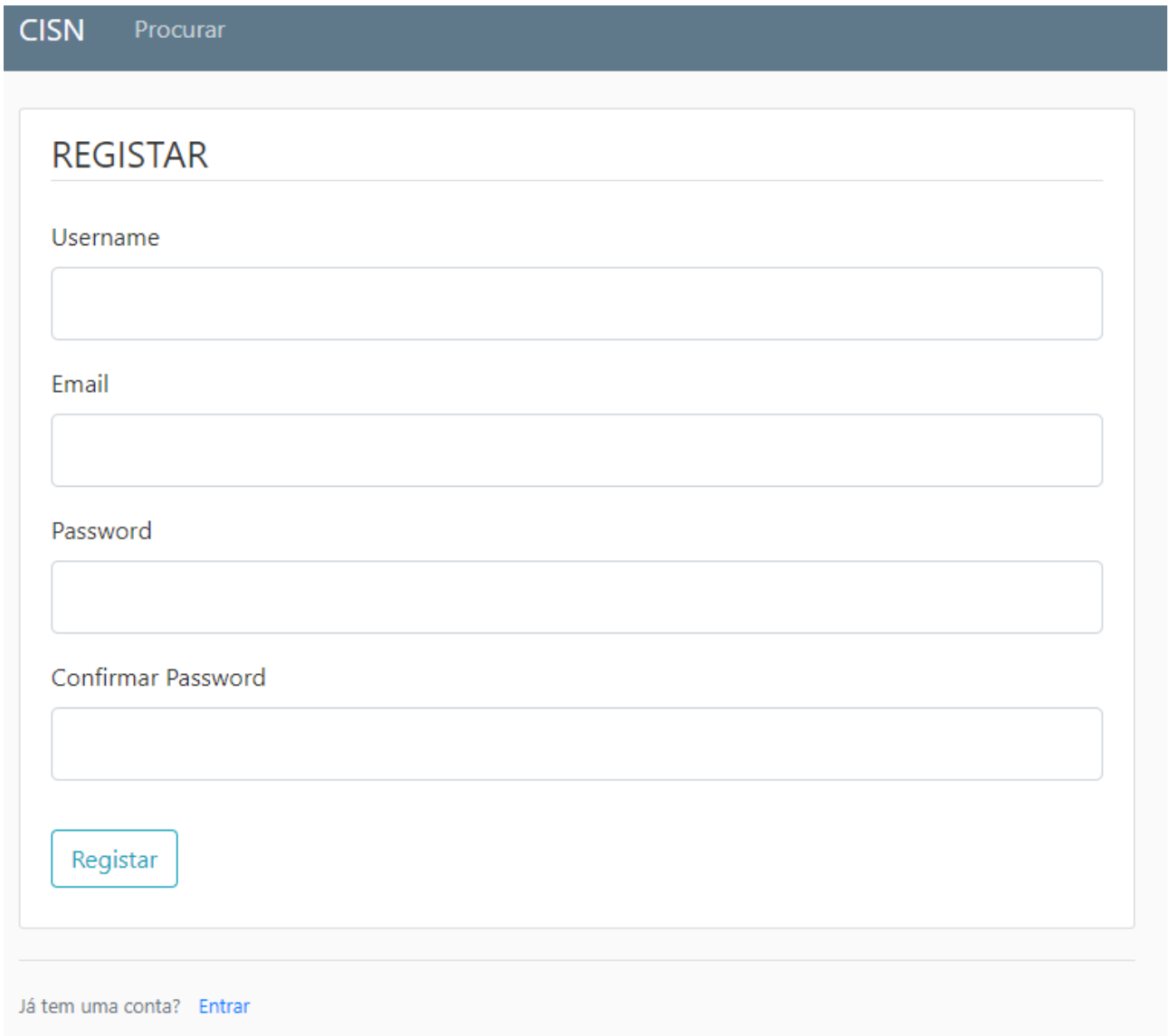

Figura 8 - Página de Registo

A fim de termos analisado a página de registo, vamos agora analisar a página de login. Nesta optamos por pedir o nome de utilizador em vez do email uma vez que no geral são mais simples de digitar, no entanto, se no futuro optarmos pelo email, a alteração é relativamente fácil tendo apenas que alterar o formulário de login. Nesta página como é natural, também pedimos a password. Tal como na página de login, damos a hipótese ao utilizador de se registar, utilizando para tal o link de registo.

A seguinte figura ilustra a página de login.

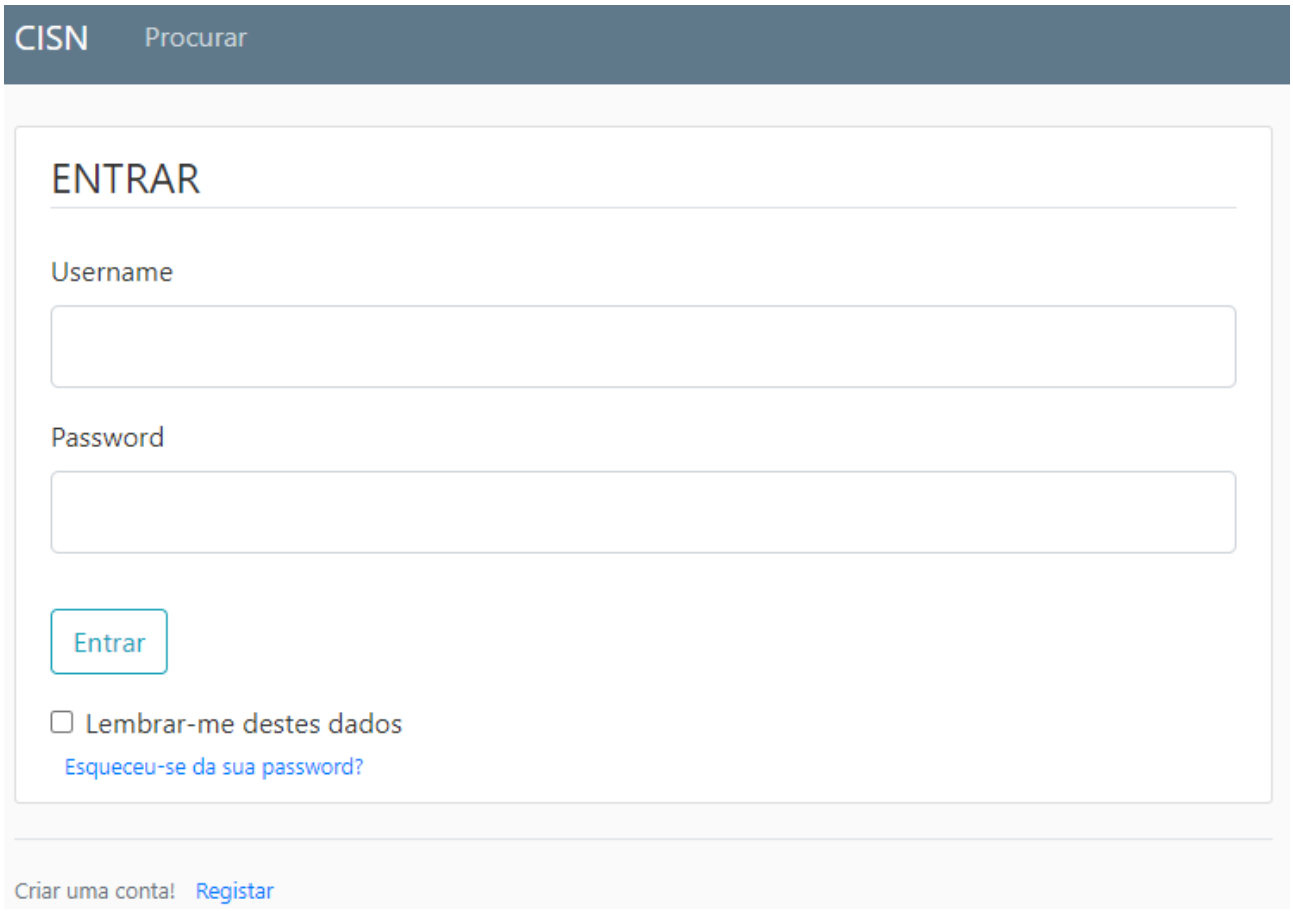

Figura 9 - Página de Login

(2) Aplicação CISN

Esta aplicação é a principal do nosso trabalho, podendo mesmo ser considerada o "cérebro". Demos o nome de CISN à nossa aplicação uma vez que a sua base são conversas nas redes sociais, neste caso tweets na rede social Twitter. CISN significa "Conversation In Social Networks", que em português significa, conversas em redes sociais.

Esta aplicação foi desenvolvida em Python e tem como objetivo a recolha, processamento e análise de tweets.

Vamos agora começar por perceber como funciona a parte da recolha dos tweets.

Para começarmos a fazer a recolha dos tweets, foi necessário fazermos um pedido para termos acesso à *API* do *Twitter*[7], onde foi necessário justificar o porquê de queremos a conta de desenvolvedor. Apos termos sido aprovados foi altura de estudar os limites que nos eram impostos. Esses limites iam desde a quantidade de pedidos 'GET' passiveis de serem feitos à quantidade de tweets que podíamos pedir em cada iteração, por exemplo, a cada 15 minutos, o número de tweets que podiam ser retirados são 18000, no entanto preferimos retirar um pouco menos para termos a certeza que ficávamos sempre a baixo do limite. Uma outra limitação como já dissemos é a quantidade de pedidos que podíamos fazer, sendo neste caso 180.

A preocupação com os limites surge porque no caso de passarmos os mesmos íamos estar a cometer uma "infração" e a nossa conta ficava marcada. No caso de cometermos 3 infrações, a nossa conta de desenvolvedor ia ser excluída não podendo fazer mais recolhas, impossibilitando-nos assim de continuar este trabalho.

A nossa aplicação lida também com possíveis erros do servidor, tais como os erros, 401 (não autorizado), 404 (não encontrado), 500 (erro interno do servidor) e 503 (serviço indisponível).

Esta *API* retorna-nos como resposta um dicionário, chave valor, ou seja, um *JSON*. Este *JSON*, vai começa por ser reduzido uma vez que grande parte da sua informação não nos é útil. Para isso tivemos que "desconstruir" o *JSON* e retirar as partes indesejadas. A fim deste passo, foi altura de voltar a construir o dicionário, para o efeito utilizamos a biblioteca *dumps*[8].

[7] https://developer.twitter.com/en/apply-for-access
[8] https://docs.python.org/3/library/json.html

Apos termos o nosso *tweet* final pronto, escrevíamos na nossa base de dados, contudo, a *API* por vezes retorna *tweets* repetidos logo tivemos que ter isso em consideração. Para resolver este problema, criamos uma solução de uma forma simples, guardar os ID's dos tweets já recolhidos e verificar se o id do tweet a ser escrito não estava nessa lista. É relevante dizer que esta lista é lida sempre que o programa inicia e é atualizada no decorrer da execução. A solução de colocar os dados dentro de um set seria uma solução apenas momentânea uma vez que iriamos perder os dados assim que o programa fosse encerrado.

Esta aplicação foi construída para podermos executá-la sem termos que estar a vigiá-la uma vez que a recolha de tweets é um processo bastante moroso.

Esta aplicação irá receber a query feita pelo utilizador na plataforma web e irá comunicar com a base de dados pedindo assim os dados. Apos ter os dados irá fazer o processamento e análise dos tweets. O resultado da análise será comunicado à nossa aplicação web para que esta possa dispor a informação ao utilizador.

O processamento do sentimento e subjetividade é feito com base no *Textblob*[9] que é uma biblioteca de Python para processar informação textual. Ela fornece-nos uma API simples para fazermos processamento de linguagem natural (PNL), assim sendo apenas necessitamos de fazer a limpeza previamente do texto e de seguida usar a função de análise de sentimento ou subjetividade. No caso da subjetividade vamos receber um valor entre 0 e 1 na qual valores mais próximos de zero indicam que é um tweet factual, o contrário indica que é um tweet de opinião. Em

[9] https://textblob.readthedocs.io/en/dev/

relação ao sentimento, obtemos um valor compreendido entre -1 e 1, sendo que valores negativos indicam sentimento negativo, valor zero indica que o tweet foi classificado como neutro e por fim, valores positivos indicam sentimento positivo.

Nestas duas abordagens valores mais próximos dos extremos indicam mais certeza em relação à classificação.

(3) Base de dados Firebase

Inicialmente tínhamos planeado que a base de dados seria mongoDB ou PostgreSQL, porem na primeira defesa do TFC foi nos sugerido fazer a alteração da base de dados para a *Firebase*, no início estávamos um pouco céticos, no entanto com alguma investigação decidimos avançar com essa sugestão.

Esta base de dados pertence à google e é bastante intuitiva de utilizar, para alem disse dá-nos a possibilidade de utilizar como armazenar ou utilizar como uma base de dados em tempo real.

Esta também nos oferece mecanismos de autenticação, segurança na transferência de dados, funcionamento em vários tipos de dispositivos e por fim, mas não menos importante dá-nos a hipótese de expansão consoante o uso. A Firebase permite-nos armazenar dados até 1 Gb depois será necessário efetuar um pagamento.

Na nossa base de dados por enquanto temos dois temas (“Poliamor” e “COVID-19”). No futuro podemos colocar quantos temas quisermos, mas por agora precisamos de um “ambiente” mais reservado. É indispensável salientar que apenas nós podemos escrever na base de dados, caso contrário os utilizadores poderiam tirar benefício disso. Estes benefícios podem ser por exemplo o aproveitamento das credenciais da *Firebase*

para uso indevido, inserção de tweets de forma a deturpar a veracidade dos resultados

A figura seguinte representa a forma como estruturámos os tweets, também é possível ver como ficam guardados.

Figura 10 - Representação da Base de Dados

5. ANÁLISE BIBLIOGRÁFICA

Apesar do interesse recente e crescente em utilizar os dados contidos no Twitter para examinar comportamentos, atitudes, demografia entre outras, ainda é possível afirmar que até ao momento existe um espaço significativo para crescimento em relação à capacidade de extração de dados do Twitter para análise e investigação. De seguida serão apresentados alguns trabalhos que foram publicados e que de certo modo vão ao encontro do trabalho que será desenvolvido neste TFC. Estes trabalhos têm algumas semelhanças, no entanto

não existe uma correspondência direta. As semelhanças poderão ser ao nível da análise demográfica, opções de voto em eleições, influencia na aquisição de um produto, deteção de rumores e outras coisas mais.

No trabalho [5]*Uma ferramenta para análise de sentimentos de tweets em português* foi desenvolvida uma aplicação cliente servidor. Esta aplicação tinha o objetivo de recolher e classificar (positivos ou negativos) Tweets de acordo com a procura de um determinado assunto. Esta aplicação também permitia verificar as pesquisas que os utilizadores faziam.

Ao longo do projeto [1]*Visual Sentiment Analysis on Twitter Data Streams* foi desenvolvido um trabalho de forma a ser possível fazer uma análise visual de sentimentos. Para isso foi criada um pipeline com 3 fases, sendo elas, análise de sentimentos baseada em tópicos, análise de fluxo e por fim, a criação de uma tabela de sentimentos baseados numa matriz de pixéis juntamente com um mapa de alta densidade geográfica.

No trabalho [2]*Rumor Detection and Classification for Twitter Data* foi desenvolvido um protótipo cujo a função era verificar a validade dos dados, com o objetivo de encontrar rumores contendo algum tipo de informação menos correta. Ao serem analisados os dados, alguns deles iam ser considerados rumores e por sua vez estes tinham que ser classificados.

6. MÉTODO E PLANEAMENTO

O seguinte cronograma representa o planeamento de execução feito no início do TFC.

No cronograma seguinte é possível ver a gestão e calendarização das etapas deste trabalho. Neste cronograma também é possível verificarmos as etapas existentes com o respetivo esforço necessário medido em semanas de trabalho. A sigla 'Qtr 1 2020' representa o intervalo de tempo entre 1-Janeiro-2020 e 31-Março-2020 (primeiro trimestre do ano).

A sigla ‘Qtr 2 2020’ representa o intervalo de tempo entre 1-Abril-2020 e 31-Junho-2020 (segundo trimestre do ano).

Toda esta calendarização revelou ser bastante precisa tanto a nível cronológico como a nível de esforço, no entanto os processos, 1) Aprender a utilizar API do Twitter e aceder à mesma, 2) Aprender a utilizar API do Botometer e utilizar a mesma, 3) Desenho e implementação de sistema, em Python, para parametrizar mensagens de texto de interesse, 4) Processamento dos objetos JSON foi despendido mais tempo do que estava planeado uma vez que existem bastantes detalhes para aceder e respeitar a API do Twitter, para alem disso foi necessário uma análise cautelosa da resposta proveniente da mesma uma vez que todo o trabalho ia depender dessas informações.
Sendo assim foi necessário adiar o tempo despendido a mais nestes processos referidos supra.
Para este trabalho adotamos uma metodologia semanal, sendo assim, todas as semanas tínhamos um objetivo que deveria ser cumprido de forma a completarmos a meta mensal. Esta estratégia revelou ser bastante vantajosa uma vez que conseguimos concluir todo o calendário de objetivos.

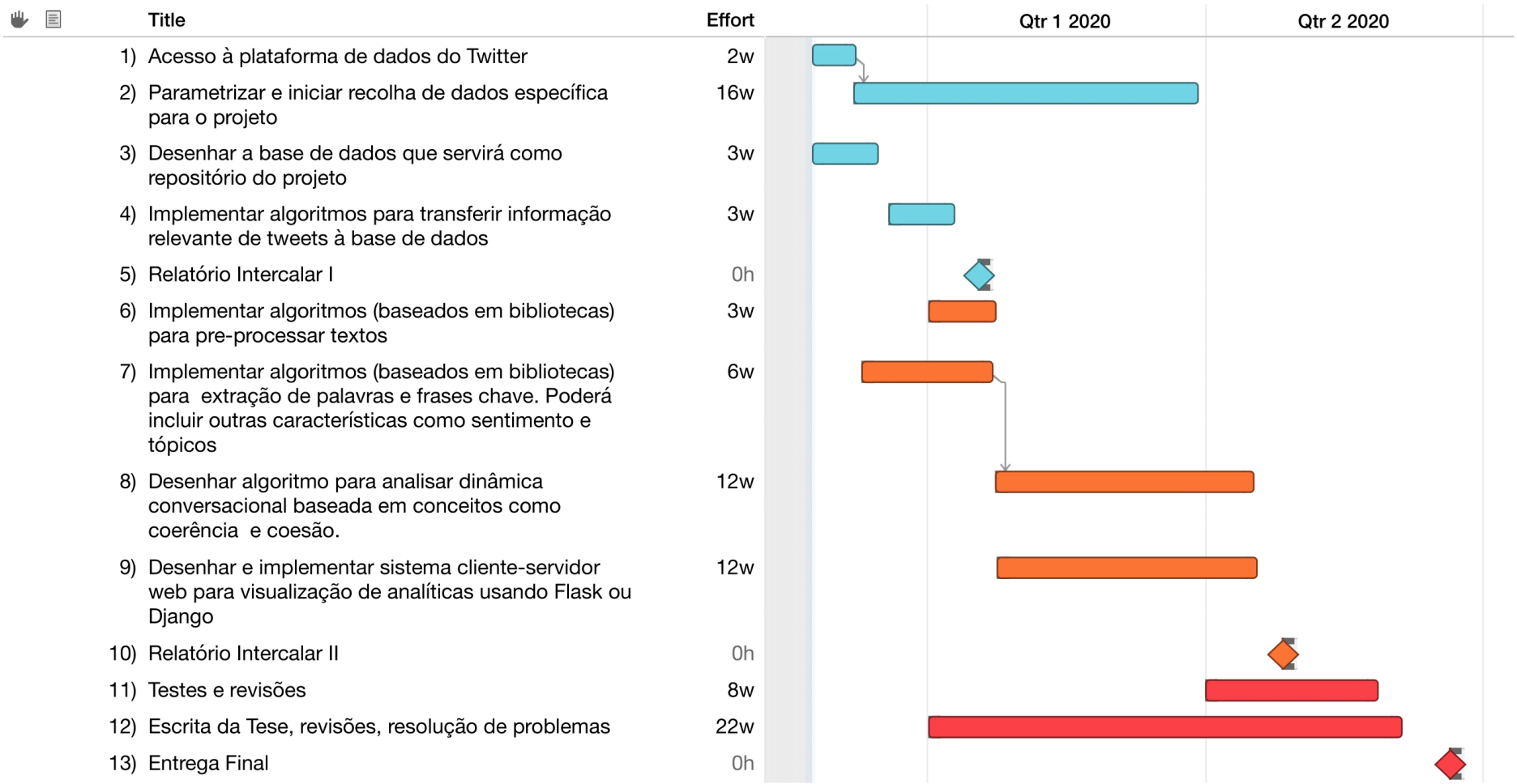

Figura 11 - Cronograma do TFC

7. RESULTADOS

Para a realização dos testes, recorremos a participantes que não têm qualquer ligação ao meio académico, as idades destes estavam compreendidos entre os 26 e os 54 e no total foram realizados apenas 3 testes devido ao *COVID-19*. Antes de cada testes dávamos a possibilidade de explorar a aplicação durante 5 minutos, sendo que todos os participantes optaram por esta possibilidade. Os testes efetuados irão em anexo.
Uma vez que amostra era muito pequena os testes não são totalmente conclusivos, no entanto obtivemos os seguintes resultados:

Para os testes: "Efetuar registo", "Efetuar Login ", "Procurar com determinados filtros", "Identificação do tweet com mais likes", "Identificação do significado dos marcadores no mapa", "Exemplo de 2 palavras positivas e negativas" e o teste "Percentagem de tweets neutros", todos os utilizadores responderam / executaram com destreza.

No teste "Significado da barra de percentagem", dois dos participantes tiveram dificuldades em perceber o seu significado, sendo assim, foi um fator obvio que necessitava de uma alteração. Para resolver este problema, colocamos uma legenda por cima da barra, ficando assim mais intuitivo.
Para o teste "Efetuar o download dos tweets utilizados", um dos participantes teve alguma dificuldade em perceber onde poderia ser feito, no entanto, sem ajuda conseguiu encontrar. Este pode ser um indicador que devemos alterar esta localização, no entanto não foi conclusivo uma vez que apenas um participante teve dificuldade.

8. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho foi sem dúvida dos mais complexos que desenvolvi durante todo o curso, quer a nível de exigência quer a nível de trabalho a desenvolver. Considero que fiz uma ótima escolha uma vez que para alem de ter sido aliciante durante todo o seu desenvolvimento senti também que cresci bastante em termos de conhecimento e profissionalismo.

Neste projeto, grande parte das ferramentas que utilizei, foram completamente novas. Não no sentido de serem recentes, mas no sentido de ter sido necessário todo o processo de aprendizagem para as utilizar. Através deste processo consolidei competências de investigação, seleção, organização e comunicação da informação.

Este trabalho abrangeu uma vasta panóplia de cadeiras que tivermos durante o curso, tais como: Base de dados, Programação Web, Sistemas de informação multimédia, Inteligência artificial e um pouco de cada uma das outras cadeiras. Sendo assim foi necessário um esforço adicional para fazer a interligação entre todas.

Este projeto foi ambicioso desde o primeiro momento uma vez que era bastante extenso e tal como foi dito anteriormente, grande parte das tecnologias eram desconhecidas, no entanto todos os objetivos à qual nos propusemos desde início foram alcançados com a exceção da implementação de novos algoritmos. Este último ponto penso não ser impossível de realizar por um aluno de licenciatura, no entanto considero ser mais adequado para uma tese de mestrado ou até de doutoramento.

Para trabalhos futuros, sugeria a implementação de outro tipo de base de dados que não fosse a *Firebase*. Esta base de dados tem muitas vantagens em relação às tradicionais tais como a possibilidade da obtenção de dados em tempo real. No entanto tem uma desvantagem que no nosso caso se revelou impeditiva,

sendo esta, a impossibilidade de fazer queries complexas dificultando assim a apresentação dos dados que realmente eram pretendidos.
Sugiro também que durante o desenvolvimento de projetos como este adotemos a mentalidade "keep it simple" quer a nível de design quer a nível de implementação de algoritmos, uma vez que a inserção de complexidade irá trazer problemas futuros.

Para finalizar é importante referir que durante todo o desenvolvimento contei o apoio do professor Manuel Pita, sem ele este trabalho teria sido muito mais difícil e no limiar impossível de realizar por um só aluno.

Para a realização dos testes, recorremos a participantes que não têm qualquer ligação

BIBLIOGRAFIA

[1] C. R. H. J. U. D. D. A. K. L.-E. H. M.-C. H. H.-P. L. U. o. K. Ming Hao, *Visual Sentiment Analysis on Twitter Data Streams ,* pp. 1-2, 15 Dezembro 2011.

[2] M. D. Sardar Hamidian, *Rumor Detection and Classification for Twitter Data,* pp. 1-7, 2015.

[3] Vicinitas, "Vicinitas," [Online]. Available: https://www.vicinitas.io/blog/twitter-social-media-strategy-2018-research-100-million-tweets#language. [Acedido em 25 01 2020].

[4] M. Masse, REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces., O'Reilly Media, 2011.

[5] Y. d. A. M. B. Rodrigo dos Santos Miguel, *UMA FERRAMENTA PARA ANÁLISE DE SENTIMENTOS DE TWEETS EM PORTUGUÊS,* pp. 1-19, 7 Agosto 2018.

[6] "Diário de notícias," 08 Dezembro 2018. [Online]. Available: https://life.dn.pt/as-redes-sociais-estao-a-mudar-o-nosso-cerebro/comportamento/346601/. [Acedido em 20 Novembro 2019].

[7] "Life Wire," [Online]. Available: https://www.lifewire.com/what-is-a-hashtag-on-twitter-3486592. [Acedido em 22 Novembro 2019].

[8] "Twitter Developer," Twitter, [Online]. Available: https://developer.twitter.com/en/products/tweets. [Acedido em 22 Novembro 2019].

[9] "Towards Data Science," [Online]. Available: https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa. [Acedido em 22 Novembro 2019].

[10] D. M. B. M. A. B. Y. B. A. J. G. K. M. M. F. .. &. S. Lazer, "The Science of Fake News," *Science,* pp. 1094-1096, 2018.

[11] "Marshable," [Online]. Available: https://mashable.com/2013/12/17/twitter-popular-languages/?europe=true. [Acedido em 21 09 2019].

ANEXOS

Id	Ação	Input	Output esperado	Output obtido	Resultado do teste	Comentários
1	Efetuar registo	**Nome**: Jose Oliveira **Email**: jose13@gmail.com **Password**: mexico	Registo com sucesso			
2	Efetuar Login	**Nome**: Jose Oliveira **Password**: mexico	Login com sucesso			
3	Procurar com filtros	**Tema:** Covid-19 **Subjetividade**: tweet factual **query**: --	Página de resultados			
4	Identificação do tweet com mais likes	----	Christine M. Chan			
5	Identificação do significado dos marcadores no mapa	----	Localização dos tweets			

6	Exemplo de 2 palavras positivas e negativas	----	Ex: Staysafe, young Globaldisaster death			
7	Percentagem de tweets neutros	----	20% de tweets neutros			
8	Significado da barra de percentagem	----	Percentagem de match com os filtros usados			
9	Efetuar o download dos tweets utilizados	----	Json com tweet Ids			

Tabela 4 -Test cases da aplicação Cliente-Servidor

GLOSSÁRIO

Endpoints: Utilizados para desbloquear dados contidos em Tweets. Endpoints possibilitam publicar, gerir, selecionar, filtrar e pesquisa por tópicos ou tendências de Tweets.

Hashtag: Palavra chave ou frase utilizada para descrever um tópico ou tema que é imediatamente processado pelo símbolo '*#*'. Hashtags ajudam utilizadores a encontrar tópicos que têm interesse para eles.

Inteligência artificial: Área da ciência da computação que enfatiza a criação de máquinas "inteligentes" que reagem e trabalham como humanos.

Machine Learning: É uma aplicação de inteligência artificial (*IA*) que oferece ao sistema a habilidade para aprender automaticamente e melhorar com base em experiências prévias sem serem explicitamente programadas.

Microblogging: Atividade ou prática de criar pequenos e frequentes '*posts*' nas redes sociais. Permitem aos utilizadores trocar entre si pequenas frases, imagens individuais ou links de vídeos.

Tweet: Pequenas mensagens enviadas no Twitter, contendo até 240 caracteres.

Word Embeddings: Representação mais popular do vocabulário de um documento. É capaz de capturar contexto de uma palavra num documento, similaridade semântica e sintática, relação com outras palavras, entre outras coisas.