Escola de Tecnologias e Arquitectura
Departamento de Ciências e Tecnologias de Informação

Sistema Inteligente de Recolha e Armazenamento de Informação proveniente do Twitter

Gaspar Manuel Rocha Brogueira

Dissertação submetida como requisito parcial para obtenção do grau de Mestre em Software de Código Aberto

Orientador

Professor Doutor Fernando Manuel Marques Batista, ISCTE-IUL

Co-Orientador

Professor Doutor João Paulo Baptista de Carvalho, Instituto Superior Técnico, Universidade de Lisboa

Setembro de 2015

Resumo

Independentemente do grau de conhecimento e utilização das redes sociais é inegável a sua importância na sociedade contemporânea. Publicitar um evento, comentar ou divulgar uma ideia são práticas comuns nas redes sociais, tornando-as num meio propício à expressão da opinião individual e sua disseminação através dos vários canais levando, consequentemente, à conceção e formação de juízos de valor e facto acerca das mudanças e acontecimentos no mundo que nos rodeia. Analisar e monitorizar sentimentos relativos a uma organização em especifico, prever vendas e aceitação de um produto ou serviço por parte do consumidor, antecipar a propagação de um vírus pela população, são exemplos concretos de como a informação recolhida nas redes sociais, pode ser útil em diversos campos da investigação (áreas como o turismo, marketing e saúde são as que mais se tem vindo a fortalecer mediante este fenómeno). Considerando tal relevância, levantam-se questões acerca do impacto que as redes sociais têm na atual sociedade e indubitavelmente debate-se a temática de como tratar e abordar essa informação de forma analítica e efetivamente útil. Para construir (ou desconstruir) um fato credível, é necessário um volume considerável de dados e uma cobertura assinalável do conjunto de utilizadores do Twitter. Diversos autores que desenvolveram trabalhos relacionados com esta problemática, têm constatado dificuldade em obter volumes significativos de informação, por limitação do Twitter em fornecer acesso aos seus dados. Perante estas circunstâncias, os dados recolhidos estão muitas vezes condicionados a uma análise limitada onde se torna complexo compreender os verdadeiros contornos das questões, ou por vezes são consideradas apenas algumas das suas características, de modo a simplificar a modelação e armazenamento. Tendo como premissa reduzir este enviesamento de informação, o objetivo deste trabalho consiste em desenvolver uma arquitetura para construção de um *corpus* de tweets tentando ultrapassar as limitações impostas pelo Twitter. Explora-se o paradigma das bases de dados NoSQL de modo a armazenar integralmente cada tweet, resultando num Sistema de Informação que automatiza a recolha, processamento, armazenamento e acesso a um volume considerável de tweets, produzidos em Portugal por autores portugueses e escritos em Português Europeu. A arquitetura apresentada produz um *corpus* de tweets produzidos em tempo real, que contêm indicação da sua geolocalização. A partir de tweets geolocalizados é efetuada a expansão do *corpus* pela leitura da timeline dos autores de tweets geolocalizados, conseguindo-se a recuperação de grande parte da informação produzida por estes. Em média são recuperados cerca de 530 mil tweets por dia.

Palavras Chave

Sistema de Informação, Twitter, Redes Sociais, Big Data, MongoDB, REST API, Visualização de Dados.

Abstract

Regardless the degree of knowledge and use of social networks, it is undeniable its importance in contemporary society. Advertise an event, comment or release an idea are common practices in social networks, making them an environment conducive to the expression of individual opinion and its dissemination through the main channels, leading consequently to the build of judgments of value and fact about changes and developments in the world around us. Analyze and monitor feelings relating to a specific organization, sales forecasting and acceptance of a product or service by the consumer, anticipate propagation of a virus among the population, are concrete examples of how the information collected on social networks can be useful in several fields of research (areas such as tourism, marketing and health are the most contemplated by this phenomenon). Considering such relevance, arise questions about the impact that social networks have in society and, undoubtedly, it is debated how to treat analytically and effectively this information, making it really useful information. To construct (or deconstruct) a credible fact, it is needed a considerable amount of data and a remarkable coverage of Twitter users. Several authors, who developed works related to this issue, have found difficulty in obtaining large volumes of information, having in account the limitation of Twitter concerning to give access to private data. In those circumstances, the data collected are often constrained to a limited analysis and becomes complex to understand the true contours of the themes. Sometimes it is even considered only some of the many characteristics in order to simplify the modeling and storage. Having as a premise reduce this skewing of information, the objective of this work is to develop an architecture having as a foundation the building of a *corpus* of tweets in attempt to overcome the limitations imposed by Twitter. It is exploited the paradigm of NoSQL databases in order to fully store each tweet, resulting in an Information System that automates the collection, processing, storage and access to a considerable volume of tweets, produced in Portugal, by Portuguese authors and written in European Portuguese. The presented architecture produces a *corpus* of tweets done in real time containing indication of its geolocation. From geolocated tweets is made the expansion of *corpus* by reading the timeline of the authors of geolocated tweets and it is possible to recover much of the information produced by them. On average, are recovered 530K tweets per day.

Keywords

Information System, Twitter, Social Networks, Big Data, MongoDB, REST API, Data Visualization.

Agradecimentos

Nós somos aquilo que fazemos repetidamente.
Excelência, então, não é um ato, mas um hábito.

Aristóteles

Finalizado o trabalho que conduziu à escrita da Dissertação de Mestrado apresentada neste documento é merecido o reconhecimento e os respetivos agradecimentos a quem de alguma forma contribuiu para que tal fosse uma realidade.

É com o maior prazer que manifesto ao Professor Fernando Batista, o meu agradecimento por acreditar no meu trabalho, pela confiança transmitida durante esta etapa, pelas palavras de incentivo, pelos sábios e oportunos conselhos e sugestões, que foram fundamentais para a realização desta investigação. Quero expressar um especial agradecimento ao Professor João Paulo Carvalho, pela sua disponibilidade, pelas suas sugestões e pelo otimismo que demonstrou com o meu trabalho e com quem tive o privilégio de partilhar opiniões e ideias. O meu obrigado a ambos!

Agradeço ao Laboratório de Sistemas de Língua Falada (L2F) do INESC-ID pelos recursos que me foram disponibilizados e que se revelaram fundamentais para o desenvolvimento deste trabalho. Um agradecimento especial ao Pedro Fialho e ao Hugo Rosa pela permanente colaboração e auxílio na administração dos recursos de hardware e software utilizados neste trabalho. Foi um prazer trabalhar convosco.

É devido igualmente um agradecimento especial às sempre pertinentes observações da Professora Manuela Aparício e do Professor Carlos Costa, em diversas reuniões promovidas por ambos, de onde resultaram relevantes contribuições para esta Tese. O meu obrigado também a ambos.

À Silvia Cardoso, Joana Correia e Salomé Brogueira um agradecimento também especial pela sua amizade e tempo dedicado à revisão deste documento. Uma palavra de apreço é também devida a todos os colegas da equipa de CQOS que integra o serviço NOESIS na NOS, pelo seu encorajamento e companheirismo ao longo de todo este processo.

Por último, mas não menos importante, um agradecimento muito especial à minha família, nomeadamente aos meus pais, irmã e avós com os quais sempre pude contar. Obrigado pela vossa compreensão e incentivo mesmo nos momentos em que me esqueci de ser filho, irmão e neto por estar absorvido neste trabalho. Sem vós, não teria sido possível!

“Perante grandes desafios 88% das pessoas chamar-te-ão de louco. Mas se ultrapassares tais desafios com distinção, apenas 8% te felicitarão”. Este trabalho é dedicado aos 8% que sempre acreditaram!

Lisboa, setembro de 2015
Gaspar Manuel Rocha Brogueira

Conteúdo

Lista de Figuras

Lista de Tabelas

Nomenclatura

No texto desta dissertação são utilizados acrónimos que referenciam alguns conceitos, cujo significado se encontra descrito de seguida:

ACID - Atomicity, Consistency, Isolation, Durability

AFS - Andrew File System

API - Application Programming Interface

CAP - Consistency, Availability, Tolerance

CTT - Correios, Telégrafos e Telefones; Correios de Portugal, S.A.

BSON - Binary JSON

GPS - Global Positional System

HTTP - Hypertext Transfer Protocol

JSON - JavaScript Object Notation

PHP - PHP: Hypertext Preprocessor

MOOC - Massive Open Online Course

NoSQL - Not Only SQL

REST - Representational State Transfer

SGBD - Sistema de Gestão de Bases de Dados

SMS - Short Message Service

UnQL - Unstructured Query Language

URI - Uniform Resource Identifier

URL - Uniform Resource Locator

XML - eXtensible Markup Language

WSGI - Web Server Gateway Interface

1 Introdução

> In April 2010, the U.S. Library of Congress and the popular micro-blogging company Twitter announced an agreement providing the Library a digital archive of all public tweets - short Web messages of up to 140 characters - from March 2006 (when Twitter first launched) through April 2010. Additionally, Twitter agreed to provide the Library all future public tweets on an ongoing basis (Raymond, 2010). The Library of Congress' planned digital archive of all public tweets holds great promise for the research community, providing long-term curation and access to this valuable information resource. Yet, over five years since its announcement, the archive remains unavailable.
>
> *Michael Zimmer (2015)*

Neste capítulo é introduzida a investigação desenvolvida na presente dissertação, enunciando os objetivos e a motivação que estiveram na sua origem. Na Secção 1.1 é apresentada a motivação para o estudo realizado, prosseguindo-se na Secção 1.2 com a exposição da questão de investigação que esteve na base do presente trabalho, enquadrada no contexto da análise das redes sociais. Segue-se na Secção 1.3 um breve resumo de cada um dos módulos do Sistema de Informação resultante desta dissertação. A Secção 1.4 enuncia as principais contribuições da tese. Na Secção 1.5 apresenta-se o plano geral da dissertação com um breve resumo de cada Capítulo.

1.1 Motivação

O sociólogo espanhol Manuel Castells, defende na sua famosa triologia *The Information Age* que a era industrial está a chegar ao fim e que começou uma era da informação, onde a necessidade de poder das redes como infra-estrutura organizativa está na base de uma economia baseada na informação (Castells, 2000). De facto, a sociedade atual vive hiperconectada. Apesar da distância, a todo o momento estamos próximos de amigos ou de outras pessoas cujas vidas queremos acompanhar, vendo as suas fotografias no Instagram, os conteúdos que partilham no Twitter e no Facebook, os vídeos que colocam no Youtube ou as experiências profissionais publicadas no LinkedIn. A facilidade com que se pode acompanhar a vida de alguém, seja pela sua localização geográfica difundida no Foursquare, ou pelas fotografias das férias colocadas no Flickr ou ainda pelos interesses partilhados no Badoo, faz com que os gostos, interesses e demais características sejam expostas ao mundo, através das redes sociais.

O conceito de rede social é definido por D.M. Boyd (2007) como um serviço baseado na Web que permite construir um perfil público ou semipúblico num sistema delimitado, onde é permitido a liberdade de partilha de informações, opiniões e pensamentos. Cada utilizador gere uma lista de potenciais amigos com os quais mantém uma conexão, visualizando as suas publicações e atualizações de estado, podendo aceder à lista dos respetivos amigos. Quando surgiram as primeiras redes sociais como o Friendster e o MySpace, destinavam-se a permitir que os utilizadores conhecessem mais pessoas, nomeadamente os amigos de amigos, que poderiam ter os mesmos interesses, gostos ou paixões. Essencialmente, as redes sociais foram concebidas para as pessoas alargarem as suas relações sociais. Mas o que tornou estes serviços tão populares foi o facto de também constituírem uma plataforma para as pessoas contactarem com os seus amigos. As redes sociais constituem atualmente uma forma privilegiada de partilha de informação, indiscutivelmente mais rápida e mais democratizada por comparação com os meios de acesso à informação utilizados em gerações passadas. O conteúdo partilhado nas redes sociais é *persistente,* o que frequentemente tem implicações significativas, permitindo a ocorrência de interações ao longo do tempo de forma assíncrona. A persistência significa que as conversas e conteúdos partilhados estão longe de serem efémeros. Esta persistência poderá indicar também que quem utiliza as redes sociais está a ficar "registada" de uma forma sem precedentes, deixando um rasto na Internet que perdurará ao longo do tempo.

A Internet delineia a cada *bit* um espaço onde cada pessoa pode exercer a sua individualidade e as suas vontades, exteriorizando os seus desejos e necessidades, numa sociedade cada vez mais baseada na era da informação. A Internet captou e capta a cada dia, a cada pesquisa no Google, a cada *post* nas redes sociais, o íntimo de uma sociedade global, constituindo-se como o meio natural para a expressão e divulgação de ideias, pensamentos, opiniões e críticas para um público aparentemente ilimitado (Vaz, 2012). Através das redes sociais as pessoas podem fazer partilhas com amplas audiências e aceder a conteúdos a grandes distâncias, o que aumenta a visibilidade potencial de qualquer mensagem particular. Em espaços físicos é bem diferente, onde as pessoas têm de fazer um esforço adicional para tornar o conteúdo visível por audiências bastante grandes. Nas redes sociais, as interações são públicas por predefinição. As redes sociais são essencialmente utilizadas por jovens, que embora atualmente quase todos tenham acesso à tecnologia, tais acessos variam extraordinariamente. Alguns têm *smartphones* topo de gama com plano de dados ilimitado, computador portátil próprio e acesso à Internet sem fios em casa. Outros estão limitados a telefones básicos e acesso à Internet apenas através dos computadores da escola ou da biblioteca. A desigualdade económica desempenha um papel fundamental. O acesso à tecnologia não é o único constrangimento, visto que os conhecimentos técnicos, a literacia em termos de redes sociais e até o domínio básico do inglês moldam o modo como os jovens utilizam as novas tecnologias. Embora as redes sociais possam permitir um contacto permanente entre a rede de amigos, os diferentes conhecimentos e possibilidades de cada indivíduo relativamente ao acesso à tecnologia, poderá estar na base da destruição do tecido social, originando a exposição a mundos sinistros, nomeadamente de predadores sexuais. As redes sociais são concebidas de modo a ajudar na disseminação de informações, incentivando a partilha de *links*, fornecendo ferramentas que permitem a republicação de textos ou imagens, facilitando igualmente copiar e colar conteúdos de um lugar para outro. Muito do que é publicado online, é facilmente disseminável mediante o clique de algumas teclas ou através de botões simples de "republicar" ou "partilhar", como é o caso do Twitter e Facebook, respetivamente.

A crescente disseminação das tecnologias de comunicação e informação tem tido um forte impacto social e económico (Heidemann et al., 2012), sendo cada vez mais fácil entrar em contacto direto

com os consumidores e aceder a informações sobre quase tudo o que existe. Tal quantidade de informação disponível atualmente, tem implicações num mundo que sempre conviveu com a escassez de informação. Como é que o volume de dados produzido diariamente tem implicações a nível global, nomeadamente ao nível do armazenamento, processamento e análise de forma eficiente para extração de informações úteis, é um problema com que se debate a sociedade atual. Para tal, em muito tem contribuído a utilização massiva das redes sociais, sendo um dos meios preferenciais para a partilha e divulgação de informações sobre os mais diversos assuntos. A importância das redes sociais vai muito para além da utilização comum, efetuada diariamente, pelos diversos elementos que as compõem. Sendo um espaço normalmente populado por grupos de amigos, que no seu todo formam uma comunidade culturalmente distinta, é imediata a perceção do contributo que as redes sociais poderão ter para o tecido empresarial. As grandes vantagens de que uma empresa pode beneficiar com a criação de um perfil numa rede social, vão para além da partilha de informação institucional ou da publicidade a produtos e serviços que disponibiliza. A principal vantagem da presença na Web e em particular nas redes sociais, é o rápido *feedback* que as empresas podem obter relativamente a inúmeros aspetos, desde a opinião que terceiros possam ter da empresa, de produtos lançados, de campanhas publicitárias, entre outros. É seguramente uma mais-valia ter esta proximidade com o público através de uma rede social, na medida em que a relação virtual estabelecida com o possível cliente, possibilita ouvir e responder com base nos seus comentários, podendo a empresa transformar e moldar os seus produtos/serviços de forma célere, adequando-os às necessidades do mercado e dos seus clientes. Além das múltiplas vantagens da presença online, nomeadamente nas redes sociais, as opiniões e pensamentos partilhados pela comunidade de utilizadores nas redes socais, quando devidamente analisada, pode constituir uma valiosa fonte de informação para diversos domínios, tais como, a política, a saúde, a segurança ou o turismo. Diversos autores têm apresentado diferentes propostas de metodologias para recolha e armazenamento de dados partilhados nas redes sociais, nomeadamente no Twitter. No entanto, devido a limitações impostas pelo próprio Twitter no acesso aos seus dados, a recolha de dados em volume considerável nem sempre é possível, limitando em grande medida o tipo de análise que é possível efetuar tendo em conta o reduzido volume de dados. Boyd (2014) relembra que as mensagens do Twitter e as atualizações de estado não estão disponíveis apenas para a audiência que, por acaso, está a seguir o fio enquanto este se desenrola; tornam-se rapidamente vestígios arquivados, acessíveis aos espectadores num momento posterior. Estes vestígios podem ser pesquisados e são facilmente republicados e difundidos. A possibilidade de aceder, guardar e analisar estes vestígios de informação persistente é a principal motivação deste trabalho.

1.2 Questão de Investigação

A tarefa de recolher, armazenar, pesquisar e analisar elevadas quantidades de tweets constitui um grande desafio por diversos motivos. A taxa de amostragem imposta pela API (*Application Programming Interface*) do Twitter e o elevado número de tweets produzidos diariamente faz com que a recolha de todos os tweets publicados se torne praticamente impossível (Oussalah et al., 2013). Além das limitações no acesso à API do Twitter, as características da infra-estrutura de hardware e software utilizado na obtenção dos dados, a velocidade de ligação à Internet e a metodologia de armazenamento dos dados influencia em grande medida o volume de tweets capturados.

A combinação do facto de um tweet ser internamente representado por um conjunto de campos

variável, cada um deles podendo conter outros campos, numa estrutura hierárquica, com a necessidade de armazenar grandes quantidades de dados é uma questão preponderante no desempenho de todo o processo. A base de dados a utilizar neste contexto, deverá possibilitar guardar dados sem estrutura fixa, de forma escalável e com elevada performance no armazenamento, sem no entanto desconsiderar a necessidade de efetuar pesquisas de forma eficiente sobre um volume considerável de dados.

Considerando os pressupostos referidos, o problema para o qual se pretende propor uma abordagem nesta dissertação, resulta da seguinte questão de investigação: como obter e armazenar de forma eficiente um *corpus* de tweets escritos em Português Europeu e publicados em Portugal por utilizadores portugueses, ultrapassando as limitações impostas pela API do Twitter?

A questão de investigação desta tese advém da importância que a informação partilhada nas redes sociais pode conter, conduzindo à formulação de muitas perguntas sobre o efeito das redes sociais na sociedade. A pesquisa de respostas para tais perguntas carece do acesso a um volume considerável de dados e a uma cobertura assinalável do conjunto de utilizadores do Twitter. Algumas das questões que têm suscitado mais interesse no meio académico, estão relacionadas com: quais os temas, tópicos ou eventos importantes partilhados nas redes sociais; quais os atores principais, a origem e propagação dos temas; o que faz determinado evento tornar-se importante e quanto tempo é necessário para que tal tenha impacto tanto nas redes sociais como na sociedade ou qual o papel dos "principais atores" na propagação dos eventos. A necessidade de resposta para algumas destas questões resultou na proposta de uma *framework* para analisar a influência das redes sociais na sociedade atual (Carvalho et al., 2013).

O trabalho apresentado nesta tese enquadra-se no projeto *Intelligent Mining of Public Social Networks' Influence in Society* (MISNIS), com o qual se pretende identificar eventos importantes (tópicos) e os atores-chave dentro desses eventos, bem como identificar a sua origem e propagação. A proposta de medidas e indicadores que identifiquem e caracterizem os eventos, a contribuição com informações essenciais para compreender fenómenos sociais da sociedade em rede e a sua relevância no mundo atual, pressupõe a análise de um volume de dados extraídos das redes sociais, neste caso o Twitter.

Com o intuito de definir uma arquitetura capaz de recolher, processar, armazenar e partilhar a informação produzida pela comunidade portuguesa do Twitter e escrita em Português Europeu, foram definidos os seguintes objetivos:

1. Desenvolvimento de um algoritmo para recolha e armazenamento eficiente de tweets produzidos em tempo real, geolocalizados em Portugal e escritos em Português Europeu;
2. Definir um método para expansão do repositório de tweets geolocalizados resultante do objetivo 1, por recuperação das publicações mais recentes dos autores dos tweets geolocalizados;
3. Implementação de uma REST API (*Representational State Transfer*), que permita o acesso ao *corpus* de tweets, por aplicações de terceiros;
4. Desenvolvimento de um Dashboard Web para visualização de métricas e indicadores sobre os dados armazenados, alguns inclusivamente em tempo real, de modo a possibilitar o acompanhamento de todo o processo;
5. Análise e extração de informação através do processamento do volume de dados recolhidos, com o objetivo de caracterizar a comunidade portuguesa do Twitter.

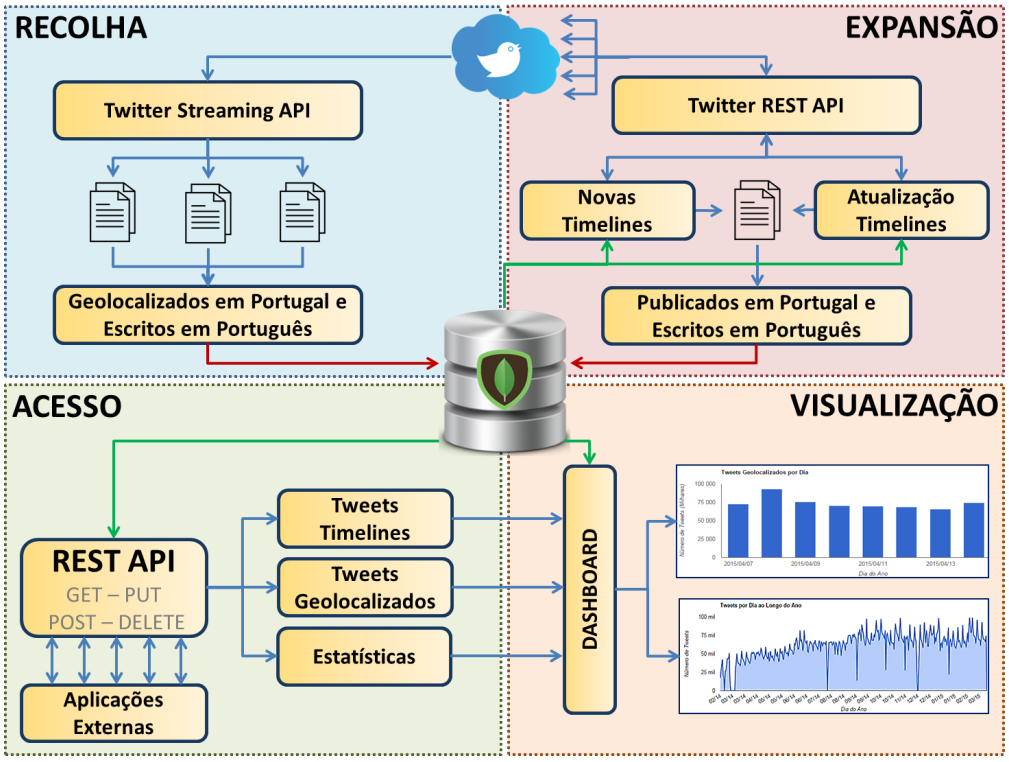


Figura 1.1: Sistema de informação para recolha e visualização de dados do Twitter.

Em resultado desta dissertação pretende-se o desenvolvimento de um Sistema de Informação modular, que de forma autónoma recolha e armazene tweets, permitindo o acesso aos mesmos por aplicações de terceiros. A visualização de métricas e indicadores relativos aos dados recolhidos na forma de um Dashboard Web, será uma mais-valia para avaliação do desempenho de todo o processo.

1.3 Solução Proposta

O Sistema de Informação resultante deste trabalho é constituído por 4 módulos que, embora distintos, estão diretamente correlacionados dado que o repositório da informação armazenada é partilhado por todos os módulos do sistema. A Figura 1.1 resume graficamente a sequência de ações de cada um dos módulos e o fluxo de informação entre todos eles. O módulo de *Recolha* captura e armazena tweets geolocalizados que estão a ser produzidos em tempo real. O módulo de *Expansão* obtém a timeline para cada um dos utilizadores portugueses identificados no módulo de *recolha*, complementando desta forma a informação existente para cada utilizador. Estes dois módulos permitem: i) identificar quais os utilizadores portugueses que produzem tweets em Portugal; ii) obter o histórico do conteúdo produzido por cada um desses utilizadores. O módulo *Acesso* define através de uma REST API uma camada de abstração para o acesso ao repositório de dados, possibilitando a sua utilização por aplicações de terceiros, de forma transparente sem a necessidade de conhecimento do modelo de dados. O módulo de *Visualização* de dados proporciona um Dashboard Web para visualização de métricas e indicadores sobre os dados recolhidos. Para alguns dos indicadores é possível a visualização em tempo real. Nos casos em que o volume de dados é elevado, é necessário o pré-processamento dos mesmos.

1.4 Contribuições da Tese

O trabalho realizado no âmbito desta tese tem como principal contribuição a proposta de uma arquitetura para a produção e partilha de um *corpus* de tweets produzidos em Portugal e escritos em Português Europeu. O método de exploração da Twitter API permite contornar de alguma forma as limitações impostas pelo próprio Twitter, conseguindo-se que o volume de tweets recolhidos cresça linearmente ao longo do tempo. O grande volume de dados obtidos irá permitir a pesquisa de respostas para as questões enunciadas por Carvalho et al. (2013).

A realização da investigação que originou esta tese decorreu entre janeiro de 2014 e maio de 2015, tendo sido desenvolvidos alguns estudos preliminares de análise da comunidade portuguesa do Twitter sob diversas perspetivas, tendo em consideração subconjuntos dos dados obtidos. Desses estudos resultaram as seguinte publicações:

- Gaspar Brogueira, Fernando Batista, João Paulo Carvalho e Helena Moniz, *Expanding a Database of Portuguese Tweets*. SLATE 2014 - Symposium on Languages, Applications and Technologies, pág. 275-282, 2014
- Gaspar Brogueira, Fernando Batista, João Paulo Carvalho e Helena Moniz, *Portuguese geolocated tweets: an overview*. International Conference on Information Systems and Design of Communication (ISDOC 2014), pág. 178-179, 2014
- Gaspar Brogueira, Fernando Batista e João Paulo Carvalho, *Arquitetura e Desenvolvimento de um Repositório de Tweets em Português Europeu*, 5ª Jornadas de Informática da Universidade de Évora, JIUE'15, Universidade de Évora, 2015
- Gaspar Brogueira, Fernando Batista e João Paulo Carvalho, *Using Geolocated Tweets for Characterization of Portuguese Administrative Regions*, 18º AGILE 2015 - Association of Geographic Information Laboratories for Europe, Lisboa, 2015
- Gaspar Brogueira, Fernando Batista e João Paulo Carvalho, *Sistema Inteligente de Recolha, Armazenamento e Visualização de Informação proveniente do Twitter*, 15ª Conferência da Associação Portuguesa de Sistemas de Informação 2015, Lisboa, 2015

Uma mais-valia do Sistema de Informação apresentado reflete-se na simplicidade com que o *corpus* pode ser utilizado no âmbito de outros estudos. Alguns parâmetros relativos aos autores dos tweets capturados, foram a fonte de informação para análise da comunidade portuguesa presente no Twitter, inferindo a idade e o género de cada autor, por aplicação de algoritmos desenvolvidos por Vicente et al. (2015). A deteção de tópicos e tendências no Twitter, nomeadamente a convocação de *tweetups* e a identificação dos respetivos autores ou as opiniões e mensagens partilhadas sobre o tema foi o foco do trabalho de Rosa et al. (2014). Ambos os trabalhos utilizaram a REST API como meio de acesso aos dados contidos no *corpus*, permitindo de certa forma validar o trabalho desenvolvido.

1.5 Estrutura do Documento

A dissertação apresentada neste documento encontra-se dividida em 6 Capítulos. O Capítulo 1 contextualiza o trabalho desenvolvido tendo em consideração a importância da análise dos conteúdos

partilhados nas redes sociais, dado que as mensagens e atualizações de estado publicadas nas redes sociais tornam-se rapidamente vestígios arquivados, acessíveis a uma vasta comunidade muito para além do momento da sua publicação. Estes vestígios podem ser pesquisados e facilmente republicados ou difundidos. A possibilidade de aceder, guardar e analisar tais vestígios de informação persistente constitui a principal motivação deste trabalho.

No Capítulo 2 é efetuada uma revisão da literatura, focando o Twitter como a fonte de informação que está na origem do desenvolvimento de diversos estudos sobre o conteúdo das mensagens partilhadas nesta rede sociais. De facto a enorme quantidade de informação que é produzida nas redes sociais, poderá ter um elevado valor para académicos, jornalistas, sociólogos, agências de marketing ou organizações com interesse em compreender o comportamento online e monitorizar tendências que são expressas entre os utilizadores das redes socais. Neste Capítulo são também abordadas algumas arquiteturas de recolha, armazenamento e visualização de dados provenientes das redes sociais.

O Capítulo 3 é focado no conceito de Big Data, sendo apresentadas as teorizações dos principais autores relativamente a este conceito, assim como as caraterísticas que lhes estão associadas. Em consequência da necessidade de armazenar grandes volumes de dados, provenientes de diversas fontes e possivelmente em formatos complexos e não estruturados, surgiu um novo paradigma de bases de dados, o NoSQL (*Not Only SQL*). O Capítulo prossegue então com uma introdução a este tipo de bases de dados, apresentando-se o teorema CAP (*Consistency, Availability, Tolerance*) que está na base da filosofia NoSQL. No entanto, para uma compreensão plena do teorema CAP são revistas as propriedades ACID (*Atomicity, Consistency, Isolation, Durability*) inerentes aos sistemas de bases de dados relacionais, e o contraste com as propriedades BASE que derivam do teorema CAP. São também referidos os diversos tipos de bases de dados NoSQL, terminando o Capítulo com uma análise mais detalhada às bases de dados NoSQL orientadas ao documento.

O Capítulo 4 expõe de forma pormenorizada a metodologia implementada na recolha e expansão de um *corpus* de tweets publicados em Portugal e escritos em Português Europeu, recorrendo à conjugação de diversas API's públicas do Twitter. Deste Capítulo resulta a definição de um conjunto de utilizadores portugueses, dos quais é capturado o seu histórico de publicações no Twitter, de modo a permitir uma análise individualizada de cada perfil ou uma análise geral sobre a comunidade portuguesa nesta rede social. Este Capítulo aborda os módulos *Recolha* e *Expansão* do Sistema de Informação decorrente deste trabalho.

No início do Capítulo 5 são discutidos alguns aspetos a considerar no desenho e implementação de REST APIs, seguindo-se a descrição de alguns dos detalhes da REST API implementada no âmbito do módulo *Acesso*. Esta REST API além de permitir o consumo dos dados contidos no *corpus* por aplicações de terceiros, poderá ser utilizada como *input* de dados para o Dashboard Web. Este Dashboard inserido no módulo *Visualização* permite o resumo de forma visual e dinâmica de alguns indicadores relativos não só ao desempenho e eficiência do Sistema de Informação, como à extração de informações relevantes sobre os dados armazenados. Alguns dos indicadores presentes no Dashboard são objeto de uma análise mais aprofundada no Capítulo 6.

Também no Capítulo 6 são apresentados dois estudos preliminares relativos a subconjuntos de dados do *corpus* obtidos em momentos distintos ao longo do trabalho: i) o primeiro estudo consiste numa breve caracterização da comunidade portuguesa do Twitter baseada na análise da recolha de uma semana de tweets; ii) o segundo estudo remete-nos para uma análise da distribuição da produção de tweets ao longo de 2014 em cada um dos distritos no território continental português, permitindo

caracterizar a dispersão populacional em diversas alturas no ano, assim como os períodos de maior atividade ao longo do dia em diversas regiões de Portugal. Em virtude das limitações impostas pelo Twitter no acesso à informação levando à escassez de dados que possibilitem o desenvolvimentos de estudos objetivos sobre os mais diversos temas, é efetuada uma breve discussão dos dados recolhidos por comparação com o volume de dados obtidos por trabalhos de outros autores.

O Capítulo 7 resume as principais conclusões deste trabalho, sugerindo diversas questões em ainda aberto que poderão dar continuidade ao trabalho realizado, explorando o *corpus* de tweets produzido.

2 Revisão da Literatura

> The main prompt for all contact on Twitter is a simple question: "What are you doing?" In practice, that question is usually interpreted as, "What interesting thought do you want to share at this moment?"
>
> *Kevin Makice*

Este capítulo encontra-se dividido em duas secções, centradas em dois temas principais: a utilização das mensagens publicamente partilhadas no Twitter como fonte de informação para a realização de diversos tipos de estudos e análises (Secção 2.1) e a análise de diversas arquiteturas desenvolvidas por outros autores igualmente com o objetivo de recolha, armazenamento e processamento de dados provenientes do Twitter (Secção 2.2).

2.1 Twitter como Fonte de Informação

O Twitter surgiu em março de 2006, desenvolvido por Jack Dorsey, Evan Williams, Biz Stone e Noah Glass, como um serviço de *microblogging*, por vezes apelidado como o serviço de SMS (*Short Message Service*) da Internet e que proporciona a possibilidade de partilhar mensagens com o máximo de 140 caracteres. Uma das características que diferencia o Twitter de algumas outras redes sociais e que o torna por isso numa das mais populares, é a sua assimetria no que respeita às conexões sociais. Ao contrário do Facebook, o Twitter não exige a conexão bidirecional entre dois membros da rede social, pelo que um utilizador pode seguir um outro, sem que o inverso seja necessariamente verdade. Este comportamento, encoraja a que o utilizador comum siga personalidades que de alguma forma se destacam na sociedade, pela sua notoriedade ou influência junto dos demais. Os utilizadores mais influentes, assim como as organizações noticiosas, tendem a ter os seus perfis no Twitter sincronizados e atualizados com o intuito de alcançar o seu público-alvo em escassos segundos (Oussalah et al., 2013).

Os pequenos comentários que constituem um tweet podem retornar um grande valor quando partilhados com o mundo. É certo que os tweets não serão lidos por todos os utilizadores do Twitter, no entanto, o número de pessoas com quem nos é possível partilhar um pouco da nossa vida e do que nos rodeia é bem superior, dado que anteriormente apenas o faríamos com quem nos cruzávamos num corredor ou na rua, residindo neste aspeto a principal mais-valia do Twitter, ou seja, a capacidade de contacto informal que permite entre os seus utilizadores. O Twitter é sem dúvida uma das redes sociais

com maior popularidade, nomeadamente nos Estados Unidos da América, Índia e Japão, países que lideram em termos percentuais relativamente ao número utilizadores (Alexa, 2014). No início de 2015, estavam registados cerca de 646 milhões de utilizadores, dos quais perto de 200 milhões permaneciam ativos, contribuindo em média com 500 milhões de novos tweets por dia (Goonetilleke et al., 2014).

A enorme quantidade de informação que é produzida nas redes sociais, poderá ter um elevado valor para académicos, jornalistas, sociólogos, agências de marketing ou organizações com interesse em compreender o comportamento online e monitorizar tendências que são expressas entre os utilizadores das redes socais. Uma das características que tem contribuído para o crescimento do Twitter, é a ausência de barreiras de privacidade, presentes em outras redes sociais, de que são exemplos o Facebook, o MySpace ou o Badoo, tornando-se uma excelente fonte de informação para a identificação de padrões de comportamento social, nomeadamente, no que diz respeito à tomada de decisões (Gruber et al., 2015), análise de sentimentos (Saif et al., 2012; Spencer and Uchyigit, 2012; Rill et al., 2014), análise de tendências de consumo (Lau et al., 2012; Kaleel and Abhari, 2015), previsões eleitorais e no treino de modelos de fala (Kong et al., 2014).

Embora o Twitter seja uma das redes sociais mais populares atualmente, a limitação de 140 caracteres por mensagem acrescida da utilização de vocabulário e expressões próprias da escrita rápida na Internet, como por exemplo os *emoticons* e diversas abreviaturas, assim como erros ortográficos ocasionais, causa dificuldades na extração de informação útil tendo por base textos com um número restrito de palavras (Moreira et al., 2014). Ainda assim, um dos tópicos de maior interesse na análise textual dos tweets, está relacionado com a deteção de tópicos (Sundar and Kumaresh, 2013), opiniões formuladas pelos utilizadores acerca de produtos, serviços ou questões sócio-políticas, entre outras. Sistemas híbridos para análise de n-gramas e redes neuronais dinâmicas foram utilizadas por Ghiassi et al. (2013) para detetar alterações no sentimento dos consumidores relativamente a marcas ou produtos. Estas opiniões/sentimentos podem revelar-se de grande interesse para áreas como o marketing ou agências governamentais (Dehkharghani et al., 2014). O mesmo autor, refere que a análise de sentimentos pode oferecer vantagens numa variedade de domínios, como por exemplo na previsão de vendas (Liu et al., 2007), na política (Tumasjan et al., 2010) ou no turismo (Claster et al., 2010). A análise de sentimentos expressa pelos utilizadores nas redes sociais é sem dúvida um dos tópicos que tem gerado maior interesse, avaliando pela quantidade de estudos recentes que têm sido produzidos sobre o tema. Para Kontopoulos et al. (2013) o conceito de análise de sentimentos é definido como um processo objetivo de determinar a polaridade de um *corpus* textual (documento, parágrafo, frase, ...) tendendo para positivo, negativo ou neutro.

No domínio da Língua Portuguesa têm sido desenvolvidos alguns estudos de análise de sentimentos, tendo como base a informação recolhida do Twitter. Avaliando o impacto de diferentes técnicas de pré-processamento de tweets produzidos em Português, Souza and Vieira (2012) demonstra que a utilização de pré-processamento e de modelos de negação tem baixo impacto na classificação dos tweets tendo em conta a polaridade lexical dos mesmos. Usando uma abordagem semântica Duarte (2013), extraiu entidades relevantes da mensagem de modo a atribuir um valor de sentimento para cada entidade encontrada. A classificação de sentimentos e a classificação de entidades tem em conta a construção gramatical da mensagem. A sua análise forneceu uma forma de visualizar e comparar o sentimento do público em relação a entidades, mostrando as preferências sobre marcas, empresas e pessoas, bem como mostrar a variação do sentimento ao longo do tempo.

A massificação da utilização de dispositivos móveis, nomeadamente de *smartphones* com acesso à

Internet e, como tal, com acesso às redes sociais, conduz a que a rápida comunicação entre pessoas, impulsione o interesse na compreensão das interações sociais, mediante a análise do conteúdo das mensagens partilhadas. A facilidade de comunicação proporcionada pela tecnologia atual, pode revelar-se de extrema importância em diversas situações, nomeadamente em casos de emergência, como por exemplo, na amaragem do avião US Airways Flight 1549 no rio Hudson em janeiro de 2009, onde um dos passageiros partilhou uma fotografia do avião e de todo o aparato envolvido, mesmo antes dos média chegaram ao local. Outro exemplo, ocorreu durante um incêndio de grandes dimensões na Austrália, em que a Autoridade de Fogos Australiana, utilizou o Twitter para enviar alertas e atualizações regulares sobre o incêndio, assim como, no terramoto no Haiti em 2010, onde nas redes sociais foram sendo partilhadas informações de auxílio para reação à situação vivida. Igualmente o terramoto seguido de tsunami no Japão em 2010, as eleições presidenciais no Irão em 2009 e os protestos no médio Oriente em 2013, demonstraram a extrema utilidade do rápido acesso e partilha de informações em tempo real, utilizando como canal de comunicação as redes sociais.

A popularidade do Twitter como fonte de informação tem conduzido ao desenvolvimento de aplicações e investigações em diversos domínios. Situações de catástrofe e assistência humanitária são fatalidades onde a informação do Twitter se tem revelado muito importante, não apenas para obter um maior conhecimento sobre a situação *per si*, mas pela forma rápida com que a informação é disseminada. Sakaki et al. (2010) usou o Twitter como fonte de informação auxiliar na identificação e localização da ocorrência de sismos, uma vez que segundo o autor "quando ocorre um sismo, as pessoas produzem muitos *posts* do Twitter relacionados com o acontecimento, o que permite a deteção do sismo prontamente, simplesmente pela observação dos tweets. Outra abordagem foi proposta por (Kumar et al., 2013b) para identificar um subconjunto de utilizadores e sua localização, que justifique serem seguidos em situações de catástrofe, de modo a obter um acesso rápido a informação útil sobre o acontecimento. Durante uma situação de crise, a localização de determinado utilizador é um fator importante para determinar se é provável que publique informações relevantes sobre o estado da ocorrência. Por exemplo, no caso de um sismo, os tweets produzidos numa área próxima do local do sismo, são provavelmente mais pertinentes para avaliação do estado da situação, que tweets produzidos em locais mais distantes. Outros estudos foram produzidos, tendo por base tópicos semelhantes (Mendoza et al., 2010), (Qu et al., 2011) e (Lachlan et al., 2014).

As publicações no Twitter podem incluir informação relativamente ao local exato onde as mensagens são emitidas, conduzindo a diversos estudos e ao desenvolvimento de aplicações baseadas na localização geográfica de tweets. Recorrendo a uma *framework* de mineração de dados para análise de sentimentos Widener and Li (2014) propôs-se a compreender como os tweets geolocalizados podem ser utilizados para explorar a prevalência de hábitos alimentares saudáveis nos Estados Unidos da América. Outra das áreas de aplicação de tweets geolocalizados é a deteção de eventos reais (Hiruta et al., 2012) ou a previsão da distribuição populacional (Lin and Cromley, 2015). A identificação de padrões na rotina dos utilizadores através do seguimento contínuo da sua localização através de tweets geolocalizados e com possível utilidade em áreas como o turismo ou a restauração, levou a que Fabio Pianese and Ishizuka (2013) tenha desenvolvido técnicas automáticas para filtrar, agregar e processar publicações no Twitter e no Foursquare, com o objetivo de extrair descrições de atividades efetuadas regularmente pelos utilizadores. Por outro lado, a identificação dos utilizadores através dos locais de onde publicam mensagens nas redes sociais pode ser usada para inferir padrões de migração a nível internacional (Zagheni et al., 2014).

Também com informação recolhida do Twitter, (Santos and Matos, 2013) e (Santos and Matos,

2014) com um conjunto de aproximadamente 2700 tweets produzidos em Portugal, conseguiu prever a taxa de incidência do vírus influenza, na população Portuguesa. Estudos idênticos foram realizados por (Lampos and Cristianini, 2010), (Chew and Eysenbach, 2010) ou (Culotta, 2010). Ainda no domínio da saúde pública, estudos realizados por (Paul and Dredze, 2011), (Scanfeld et al., 2010) e (Prieto et al., 2014) permitem estimar e monitorizar a incidência de certos problemas de saúde na sociedade, como por exemplo a ocorrência de sintomas de depressão ou distúrbios alimentares.

O Twitter é uma fonte de dados para problemas de suporte à decisão. Gerber (2014) utilizando tweets marcados no espaço e no tempo tentou prever a atividade criminal na maior cidade dos Estados Unidos. Dado que os tweets são informação pública, disponibilizados oficialmente por serviços do Twitter, o desenvolvimento de modelos de análise linguística que permitam a identificação automática de tópicos relacionados com a prática de crime, pode ter bastante relevância não só na prevenção do mesmo, como na tomada de decisão em fase de julgamento em tribunal.

2.2 Recolha, Armazenamento e Visualização de Dados

O volume de conteúdos partilhados publicamente nas redes sociais continua a crescer, havendo um grande interesse em tecnologia que possa ajudar na recolha e mineração desse conteúdo. O potencial do Twitter como fonte de dados, é sublinhado pelo seu rápido crescimento tendo-se tornado num canal privilegiado para comunicação e partilha de informação em tempo real na Web. O desenho de uma arquitetura de software para a construção automática de um *corpus* de tweets, que possibilite a recuperação de informação contida nos mesmos de forma facilitada, configura-se como um desafio tendo em consideração as questões ainda em aberto relativamente aos sistemas de recuperação de informação, nomeadamente relacionadas com o aspeto semântico, dado que as mensagens no Twitter são limitadas a 140 caracteres, conduzindo à escrita abreviada e recorrendo a poucas palavras. A limitação no acesso à informação imposta pelo Twitter é outro dos constrangimentos com que este tipo de sistemas se deparam (Oussalah et al., 2013).

No que concerne a abordagens para recolha massiva de dados do Twitter, diversos autores têm proposto diferentes abordagens, no entanto, sem conseguirem recolher um volume de dados significativo. É de salientar que o acesso mais restrito à informação do Twitter é uma das consequências da versão 1.1 da API do Twitter lançada em setembro de 2012. Na versão 1.0 era permitido o acesso a mais 80% de informação que na atual versão 1.1, no entanto, a nova versão trouxe diversos aspetos positivos como por exemplo, a substituição do XML (*eXtensible Markup Language*) por JSON (*JavaScript Object Notation*), assim como a introdução de novos *endpoints*, possibilitando o acesso a outros tipos de dados (Tornes, 2013). Com a API na versão 1.0, era possível o acesso a uma maior quantidade de dados, mediante a inclusão do IP de ligação à API numa lista com acesso privilegiado, designada por *whitelist*. Com este modo de acesso diversos autores tiveram a possibilidade de recolher consideráveis quantidades de dados, até um máximo teórico de 500 milhões de tweets por dia (Anderson and Schram, 2011).

A abordagem apresentada por Anderson and Schram (2011) tem por base uma arquitetura com elevado grau de concorrência e escalável, tendo havido o cuidado de implementar o código adaptado ao processamento *multithreaded* de forma a executar em máquinas com diversos processadores. Recorrendo a 5 IP's na *whitelist* do Twitter, foram recolhidos 1000 milhões de tweets em 2010.

Com 20 contas inscritas na *whitelist* do Twitter Kwak et al. (2010) capturou, entre 6 e 31 de julho de 2009, 106 milhões de tweets recorrendo à Search API, iniciando a pesquisa de tweets relacionados com ”Perez Hilton”. Pesquisou tweets dos seus seguidores, ,analisando 41.7 milhões de utilizadores distintos. De forma a desenvolver uma análise empírica dos padrões de influência em redes sociais, Cha et al. (2010) comparou três diferentes medidas de influência: número de seguidores de determinado utilizador, *retweets* e menções. Utilizando 58 servidores na *whitelist* foram colecionados 1800 milhões de tweets de 55 milhões de utilizadores. Embora o período de recolha de dados tenha sido semelhante ao de Kwak et al. (2010), Cha et al. (2010) conseguiram obter mais 13 milhões de utilizadores e, por conseguinte, muitos mais tweets.

A introdução da versão 1.1 da API do Twitter, alterou radicalmente o paradigma de obtenção de dados do Twitter, tornando-se bem mais escassa a informação disponibilizada. Apesar do grande interesse na informação proveniente do Twitter, há apenas um pequeno número de *corpora* disponíveis, sem que nenhum dos quais seja adequado para avaliações em grande escala. O Twitter pode ajudar a identificar ou a localizar certo tipo de eventos, com por exemplo a realização de um *meet* de jovens num centro comercial ou a ocorrência de manifestação numa qualquer cidade do mundo, no entanto, num *corpus* com um reduzido número de tweets, esta não é uma análise trivial. Tal deve-se não só à grande escala como à dispersão das publicações diárias no Twitter, o que dificulta a recolha de um conjunto significativo de informação sobre um mesmo tópico, sendo uma tarefa demorada e por vezes com elevados custos (McMinn et al., 2013).

Com o objetivo de detetar eventos através da análise de um conjunto significativo de tweets McMinn et al. (2013) colecionou tweets recorrendo à Twitter Streaming API durante 28 dias, entre 10 de outubro de 2012 e 7 de novembro de 2012, conseguindo 120 milhões de tweets, após a filtragem de tweets cuja língua de escrita não fosse o Inglês. Foram igualmente filtrados tweets considerados spam, ou seja, os tweets com mais de 3 hashtags ou com mais de 3 menções ou mais de 2 URLs (*Uniform Resource Locator*) são considerados spam e como tal foram descartados. Também com recurso à Twitter Streaming API, Petrović et al. (2010) construiu um *corpus* com 97 milhões de tweets em cerca de três meses, entre 11 de novembro de 2009 e 1 de fevereiro de 2010.

Outros autores tiveram de desenvolver os seus estudos recorrendo a conjuntos de tweets bem mais reduzidos, como foi o caso de Pak and Paroubek (2010) que obteve 300 mil tweets em resultado da aplicação de um método que permite a constituição de um *corpus* de tweets que expressam sentimentos positivos ou negativos e um *corpus* de textos objetivos, sem qualquer sentimento implícito. Tal método permite recolher amostras de sentimentos negativos e positivos de forma automática e sem a necessidade de esforço humano para a classificação dos documentos, embora os autores refiram que o tamanho do *corpus* possa ser arbitrariamente grande.

Um dos primeiros estudos relativamente à utilização do Twitter foi apresentado por Java et al. (2007), focando nas propriedades topológicas e geográficas. Os membros da comunidade do Twitter podem optar por produzir as suas mensagens de forma pública ou disponíveis apenas para os amigos. Caso o perfil do utilizador seja tornado público, as suas mensagens são incluídas numa *timeline* ("linha de tempo pública") ordenadas de forma decrescente cronologicamente, ou seja, da mais recente para a mais antiga. O acompanhamento da timeline de cerca de 76 mil utilizadores, por um período de dois meses, a contar de 1 de abril de 2007 a 30 de maio de 2007, permitiu a recolha de 1.3 milhões de tweets, através de métodos da REST API do Twitter. De notar que o período de abril a maio de 2007, corresponde a uma fase ainda embrionária do Twitter, pelo que a produção de tweets não era muito

elevada e daí a recolha de relativamente poucos tweets.

Com propósito idêntico, Wang (2010) durante o período entre 3 e 24 de janeiro de 2010 colecionou sensivelmente 500 mil tweets de 25 mil utilizadores. Wang utilizou os mesmos métodos da Twitter API que Java et al. (2007) para extrair tweets recentes e respetivos autores. De cada utilizador foram analisadas as relações com outros utilizadores e lidos os seus primeiros 20 tweets. Foi utilizado um sistema distribuído com limite a 120 solicitações por hora à Twitter API.

Para analisar comentários, sentimentos e opiniões sobre diversas marcas e empresas Jansen et al. (2009) analisou cerca de 150 mil mensagens publicadas no Twitter durante 13 semanas relativamente a 50 marcas, recolhidas entre 4 de abril e 3 de julho de 2008 através da ferramenta Summize[1], adquirida em 2008 pelo Twitter.

Em outras abordagens para armazenamento de dados do Twitter, foram propostas arquiteturas sustentadas em base de dados relacionais. Da utilização de bases de dados relacionais para guardar dados não estruturados que se apresentam no formato JSON ou XML, decorre a necessidade de os transformar em estruturas relacionais, adequadas ao seu armazenamento de acordo com o paradigma relacional. Nestes casos, a base de dados mais utilizada é o MySQL.

Uma arquitetura de software baseada na Twitter API em conjunção com Python e MySQL é proposta por (Perera et al., 2010) para recolher os tweets enviados para utilizadores específicos. Para a obtenção de dados espaciais (localização, nome, descrição, ...) dos autores dos tweets, foi utilizado a Twython API. O processo de recolha é executado em intervalos de 5 minutos, sendo recolhidos os tweets enviados para determinado *id* de utilizador, nomeadamente o Presidente Barack Obama.

Por outro lado, Oussalah et al. (2013) propõe uma arquitetura para análise semântica e espacial dos dados recolhidos do Twitter. Para a recolha dos tweets foi utilizada a Streaming API, por permitir o acesso aos *tweets* em tempo real e de forma contínua. Os tweets recolhidos são restringidos a uma região retangular delimitada por coordenadas geográficas (longitude e latitude). Na implementação do software foi utilizado Django (*framework* para desenvolvimento de aplicações web em Python), Apache Lucene (para indexação dos tweets) e MySQL para armazenamento dos dados recolhidos. Esta arquitetura também descreve formas de pesquisa de tweets por texto, nome dos utilizadores, localização, entre outras.

A arquitetura proposta por Anderson and Schram (2011), já referida anteriormente, utiliza as frameworks Spring, MVC, Hibernate e JPA e componentes de infra-estrutura Tomcat, MySQL e Lucene para recolher, armazenar e efetuar pesquisas nos dados capturados.

Diversas plataformas de extração de dados do Twitter integram mecanismos de visualização dos dados sob diversas perspetivas, de modo a facilitar a sua análise na dimensão temporal e espacial. Sistemas como o TwitInfo (Marcus et al., 2011), Twitcident (Abel et al., 2012) ou Toretter (Sakaki et al., 2010) permitem a visualização e exploração de tweets com diferentes graus de detalhe relativamente a aplicações específicas, como por exemplo, catástrofes naturais nomeadamente no combate a incêndios ou na deteção da ocorrência de sismos. O TwitInfo em particular possibilita a navegação numa vasta coleção de tweets usando uma exibição ordenada cronologicamente que destaca picos de grande atividade de produção de tweets.

Ainda no âmbito de causas humanitárias, ferramentas como o tweetTracker (Kumar et al., 2011) tem como objetivo auxiliar na monitorização de tweets em situações de emergência, provocadas por

[1] https://blog.twitter.com/2008/finding-perfect-match (acedido em 20/07/2015)

causas naturais. Aplicações web e soluções de negócio a nível empresarial, têm extrema importância quando em tempo real permitem o acesso a informação privilegiada, de modo a proporcionar uma certa vantagem em análises e decisões de negócio, das quais são exemplo o Trendsmap[2] e o Twitalyser[3].

No contexto do marketing e em campanhas de publicidade, aplicações como o tweetXplorer desenvolvida por Morstatter et al. (2013), ajudam os analistas a visualizarem grandes quantidades de dados em diferentes perspetivas, como por exemplo, a melhor altura para o lançamento de determinado produto ou serviço (quando), a importância de determinados utilizadores (quem/como) e a localização dos utilizadores chave (onde).

Aplicações para análise geográfica de tópicos de interesse partilhados e comentados no Twitter, através de interfaces web interativas permitem a exploração e visualização da informação pelos utilizadores com diferentes graus de granuralidade, ou seja, possibilitam a pesquisa por regiões, por tópicos ou por termos específicos, como por exemplo, a aplicação Limosa (Vosecky et al., 2013). A análise geográfica é igualmente relevante para outro tipo de aplicações, como é o caso da deteção de eventos do mundo real através da informação partilhada nas redes sociais (Hiruta et al., 2012), denotando-se a preocupação por parte do autor com o aspeto da visualização dos dados ao ponto de sugerir duas formas alternativas de interagir com os dados.

[2]http://trendsmap.com
[3]http://twitalyzer.com

3 Sistemas de Armazenamento de Dados

> There were 5 exabytes of information created between the dawn of civilization through 2003, but that much information is now created every two days.
>
> *Eric Schmidt, Chefe Executivo da Google*

Este Capítulo, estruturado em cinco Secções, aborda os principais conceitos relacionados com os sistemas de armazenamento de dados. Na Secção 3.1 é exposta a problemática do armazenamento da informação no contexto de Big Data, prosseguindo-se na Secção 3.2 com a análise do paradigma de bases de dados NoSQL como forma de armazenamento de dados não estruturados em larga escala. Na Secção 3.3 é efetuada uma breve análise das principais propriedades das bases de dados relacionais por contraste com as bases de dados não relacionais. A Secção 3.4 descreve os conceitos gerais associados às bases de dados NoSQL orientadas a documentos, sendo na Secção 3.5 apresentado um exemplo concreto de uma base de dados NoSQL orientada ao documento (MongoDB) e respetiva hierarquia de dados.

3.1 Dados e Informação

A era da informação tem conhecido muitos progressos. Estamos cada vez mais dependentes dos dados e da informação que deles se pode retirar. Atualmente consomem-se e produzem-se grandes quantidades de dados nomeadamente em formato digital, facilitando a sua captura, armazenamento, processamento e difusão. Todas as transações numa loja, cada pesquisa no Google ou praticamente todas as ações que reproduzimos num *smartphone* produzem grandes quantidades de dados. Com a proliferação de dados a que temos assistido e dada a redução nos custos do seu armazenamento, tem sido colocado um maior ênfase no processamento, procurando processar em tempo real elevados volumes de dados, produzidos a grande velocidade e provenientes de diversas fontes. Estes pressupostos estão na génese do Big Data podendo ter um forte impacto de um modo geral em toda a sociedade.

O conceito de Big Data surgiu pela primeira vez em 2005 numa conferência Strata[1] através de Roger Magoulas, que o utilizou para definir aquilo que descrevia como um conjunto de dados que devido ao seu volume e complexidade, impossibilitavam a devida gestão pelos sistemas de bases de dados tradicionais. Embora seja um conceito relativamente recente, já na década de 1960 várias organizações, como o CERN (Smith, 2013), enfrentavam problemas atualmente associados ao Big Data, residindo a diferença na quantidade de dados que no passado estava na ordem dos megabytes e hoje atingem os

[1] Conjunto de conferências sobre temas ligados a Data Science e Big Data, organizadas pela editora O'Reilly Media.

diversos terabytes ou mesmo petabytes (Borkar et al., 2012). Segundo Minelli et al. (2013), Big data corresponde à terceira era da informação. A primeira foi em 1954, com o surgimento dos sistemas de informação nas organizações. Passados 35 anos, o foco nos processos internos, deu lugar às interações externas, iniciando-se a época da ligação em rede, que juntamente com a globalização tornou o ambiente empresarial ainda mais complexo. Atualmente, o fenómeno do Big Data surge em consequência do aumento do poder de processamento que de acordo com a lei de Moore, duplica a cada 18 meses. O conceito de Big Data com origem na necessidade de recolher, armazenar e processar em tempo real, grandes volumes de dados, provenientes de variadas fontes e em diversos formatos, tornou-se omnipresente. Devido à sua origem repartida entre a academia, a indústria e os meios de comunicação, não existe uma definição consensual e vários têm sido os autores a proporem a sua própria definição (Ward and Barker, 2013). Um dos autores mais citados, Laney (2001), apresentou a definição conhecida pelos ”3Vs”, remetendo para os conceitos de Velocidade, Volume e Variedade. Para Doug Laney o aumento do volume de dados, a taxa a que são produzidos e a diversidade de formatos e representações em que estes podem surgir, estão na base da definição geral de Big Data. É estimado que desde o início da humanidade até ao ano de 2003 tenham sido armazenados cerca de 5 exabytes de dados, enquanto que atualmente a cada 2 dias é armazenado a mesma quantidade de dados. Em 2012 tinham-se alcançado os 2.72 zettabytes (Sagiroglu and Sinanc, 2013) e prevê-se que até 2020 a totalidade de dados armazenados alcançará os 35 Zettabytes (Zikopoulos et al., 2011). Apesar de todas as estimativas, a realidade é que grande parte desses dados não está a ser analisada e a necessidade de manter todos esses dados acessíveis, tornam o ”Volume” a característica mais importante do desafio que é o Big Data (Demchenko et al., 2013). Um dos pressupostos do Big Data é a análise dos dados “ainda em movimento e não quando já se encontram em repouso” (Zikopoulos et al., 2011), facto que se pode tornar num problema para algumas organizações, pois a velocidade com que os dados são produzidos, ultrapassa em muito a capacidade de processamento em tempo real. Os dados não se apresentam apenas sob a forma tradicional (estruturada), em muito devido à elevada proliferação de sensores, dispositivos inteligentes, redes sociais, blogs, sendo que grande parte se apresenta sob numa forma semiestruturada ou completamente não estruturada, caracterizados pela sua aleatoriedade, complexidade e dificuldade em processar.

Big Data é também definido por Hashem et al. (2015) como a conjugação de um conjunto de métodos e tecnologias para através destes, se obter valor até então oculto nos dados, podendo estes estar em diversos formatos, apresentar estrutura complexa e em elevada quantidade. Nesta definição é patente a preocupação com uma quarta dimensão, ou seja, com o Valor dos dados. Os três primeiros ”Vs” estão focados na engenharia dos dados, isto é, na recolha, transferência e armazenamento dos mesmos. O quarto ”V” está focado na ciência dos dados, nomeadamente em métodos analíticos e estatísticos para a extração de conhecimento, de modo a auxiliar na tomada de decisões (Huang et al., 2015). Recentemente outros autores, (López et al., 2015; Yang et al., 2014), têm acrescentado uma quinta dimensão, relacionada com a Veracidade, ou seja, consideram ser importante a preocupação com a precisão dos dados, uma vez que estes devem ser adquiridos de fontes credíveis e a sua segurança e confidencialidade deverá ser garantida. A Microsoft (2013) apresenta uma definição um pouco mais técnica de Big Data, referindo tratar-se de um termo cada vez mais utilizado na descrição do processo de aplicação de grande poder de computação, para que através dos mais recentes avanços nas áreas de aprendizagem automática e inteligência artificial, se possam obter informações relevantes de conjuntos de dados altamente complexos.

3.2 Bases de Dados NoSQL

A complexidade e o volume de dados obtidos de uma variedade de fontes tão vasta como são os blogs, redes sociais, sensores, vídeos, texto, sons, imagens e outras formas de dados que variam em tamanho, estrutura e formato torna praticamente impossível o seu armazenamento e manipulação através dos Sistemas de Gestão de Bases de Dados (SGBDs) tradicionais. De notar que a tecnologia das bases de dados relacionais foi proposta na década de 1970, quando as aplicações que necessitavam de guardar informação se caracterizavam por lidar com dados estruturados, ou seja, que possuem uma estrutura fixa e bem definida. Além disso, por maior que seja o volume de dados gerado em tais aplicações, ainda assim não se compara ao volume de dados produzido atualmente nas redes sociais ou na quantidade de vídeos enviados para o Youtube a cada instante, onde o universo de utilizadores tem um ritmo de crescimento acentuado, o que não acontece nas aplicações que utilizam base de dados convencionais.

A maioria dos dados disponíveis atualmente são não estruturados. Por exemplo, a evolução dos sistemas de gestão de conteúdos tem permitido guardar e manipular além dos tradicionais documentos, conteúdo proveniente da web, imagens, áudio ou vídeo. Neste contexto, surgiu um novo paradigma de Base de Dados, designado por NoSQL ("Not Only SQL"), proposto como solução para o armazenamento e gestão de grandes volumes de dados, semiestruturados ou não estruturados, que necessitam de alta disponibilidade e escalabilidade. A necessidade de uma nova tecnologia de base de dados surgiu em consequência da ineficiência das bases de dados relacionais relativamente a este tipo de tarefas. Esta necessidade decorre do aparecimento de grandes empresas baseadas na Web, surgindo deste modo a exigência de se processarem grandes quantidades de dados, não estruturados, de forma rápida e eficiente. A escalabilidade horizontal, que consiste no aumento do número de unidades de armazenamento, é uma das principais características disponibilizadas pelas bases de dados NoSQL (Abramova and Bernardino, 2013). A integridade e consistência dos dados são fatores críticos, logo as bases de dados relacionais não apresentam os requisitos que permitam suportar a escalabilidade para grandes volumes de dados (Krishnan, 2013). A maioria das principais tecnologias no âmbito do NoSQL teve a sua génese em projetos proprietários desenvolvidos pela Google (BigTable), IBM (Cloudant) e Amazon (DynamoDB). Como alternativas de código aberto tem-se o Apache Cassandra (originalmente desenvolvido para o Facebook) ou o Apache HBase.

NoSQL é literalmente a combinação das palavras *No* (não) e SQL (*Structured Query Language*). Na realidade o termo NoSQL é um acrónimo para "Not Only SQL" e é usado como definição genérica para todas as bases de dados que não seguem os princípios das bases de dados relacionais, por especificamente não ser utilizada a linguagem SQL na recuperação e manipulação dos dados armazenados e também pela possibilidade de serem guardados dados não estruturados. O nome teve origem na filosofia dos seus criadores, que ambicionavam a criação de um modelo distinto do relacional e dos seus princípios ACID. No entanto, NoSQL acabou por passar a ser interpretado como "Not Only SQL" (não apenas SQL), referindo-se não apenas a um produto ou tecnologia específica, mas a um conjunto de termos, por vezes relacionados, associados ao rápido e eficiente processamento de dados com foco na performance, confiança, segurança e agilidade (McCreary and Kelly, 2013). De entre as características fundamentais das bases de dados NoSQL, destacam-se:

- **Ausência de estrutura**: uma característica evidente no NoSQL é a ausência total ou quase total do esquema que define a estrutura dos dados armazenados. Esta ausência de estrutura facilita

a escalabilidade contribuindo para o aumento da disponibilidade. No entanto não há garantia da integridade dos dados;

- **Escalabilidade horizontal**: com o crescimento do volume de dados, aumenta a necessidade de escalabilidade e consequentemente a necessidade de desempenho. Uma das soluções para este problema, é a escalabilidade vertical, que consiste em aumentar o poder de processamento e armazenamento das máquinas ou a escalabilidade horizontal, onde ocorre um aumento no número de máquinas disponíveis para o armazenamento e processamento de dados. A escalabilidade horizontal tende a ser a solução mais viável, porém requer que uma tarefa seja dividida em diversos processos que se devem executar de forma distribuída. O termo ”escalabilidade horizontal” é a característica fundamental dos sistemas NoSQL (Cattell, 2011), referindo-se à capacidade para distribuir os dados e a carga de processamento ao longo de muitos servidores, sem RAM ou disco partilhados entre os diversos servidores. ”Escalabilidade horizontal” difere da ”escalabilidade vertical”, visto que a ”escalabilidade vertical” utiliza muitos *cores* e/ou CPUs com memória RAM e discos partilhados;

- **Suporte à replicação**: outra forma de aumentar escalabilidade é através da replicação. Permitir a replicação de forma nativa, diminui o tempo gasto na recuperação informação;

- **API simples para acesso aos dados**: o objetivo da solução NoSQL é tornar eficiente o acesso aos dados, oferecendo alta disponibilidade e escalabilidade, ou seja, o foco não está em como os dados são armazenados e sim como poderemos recuperá-los de forma eficiente. Como tal, é necessário o desenvolvimento de APIs de modo a facilitar o acesso a estas informações, permitindo que qualquer aplicação possa utilizar os dados armazenados rápida e eficientemente;

- **Eventualmente consistente**: uma característica das bases de dados NoSQL está relacionada com o facto da consistência nem sempre ser mantida entre os diversos blocos de dados, tal como referido anteriormente.

Atualmente são conhecidas cerca de centena e meia de bases de dados seguindo o paradigma NoSQL, diferenciando-se pelo modelo não-relacional de dados que adotam, não existindo unanimidade quanto a uma taxonomia oficial para todas estas bases de dados (Tudorica and Bucur, 2011). Contundo têm surgido diversas classificações baseadas em características como (Dharmasiri and Goonetillake, 2013): i) modelo de dados; ii) linguagem utilizada na consulta de dados; iii) propriedades do teorema CAP suportados pelo sistema; iv) modelo de consistência implementado; v) integridade, indexação e distribuição dos dados (Moniruzzaman e Hossain, 2013).

Segundo Dharmasiri and Goonetillake (2013) a classificação dos sistemas de bases de dados NoSQL é atualmente dividido em dois grupos (ver Figura 3.1): *Core NoSQL* cujas bases de dados foram desenvolvidas para responder a necessidades da Web 2.0 (Tudorica and Bucur, 2011) e *Soft NoSQL* que embora não estejam ligadas à Web 2.0, partilham características NoSQL, como por exemplo, não garantem as propriedades ACID. Na continuação deste Capítulo serão apenas considerados os sistemas *Core NoSQL.*

Na Tabela 3.1 são apresentados alguns exemplos de bases de dados *Core NoSQL* tendo em consideração o modelo de dados. As bases de dados da família chave-valor, também conhecidas por tabelas de *hash* distribuídas, armazenam objetos indexados por chaves, possibilitando a sua pesquisa a partir das suas chaves. Este modelo é baseado numa estrutura de dados com diversas aplicações

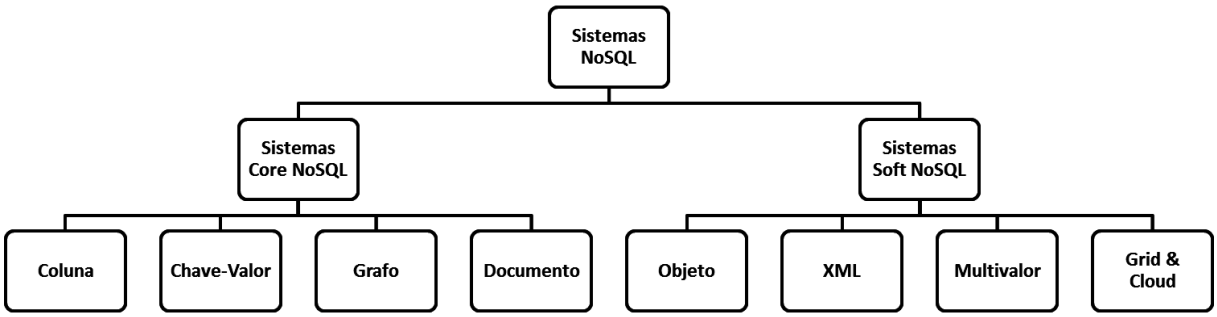


Figura 3.1: Classificação das bases de dados NoSQL:*Core NoSQL* e *Soft NoSQL* (adaptado de (Dharmasiri and Goonetillake, 2013)).

nomeadamente no desenvolvimento de sistemas de ficheiros (Indrawan-Santiago, 2012) onde os dados a armazenar são constituídos por duas partes: uma chave representada por uma *string* e o por um valor que representa os dados a armazenar originando deste modo um par "chave-valor" (Nayak et al., 2013).

A família com o modelo de dados em colunas possui uma grande capacidade de escalabilidade horizontal, o que lhes permite conseguir albergar grandes quantidades de dados (McCreary and Kelly, 2013). Estes sistemas foram bastante influenciados BigTable, sistema igualmente distribuído que permite gerir dados estruturados e com escalabilidade horizontal de forma natural, apresentado em 2006 pela Google (Chang et al., 2006). Estas bases de dados, devido ao formato tabular apresentam algumas semelhanças com as bases de dados relacionais, estando a principal diferença na forma como são geridos valores nulos. Sendo necessário armazenar um grande número de atributos, um sistema relacional guarda um valor nulo para cada coluna sem dados, por seu lado num sistema NoSQL por colunas apenas são armazenados pares chaves-valor numa linha caso esses dados existam e sejam necessários (Hecht and Jablonski, 2011). Este tipo de sistemas pode ser considerado como um tipo específico do modelo par chave-valor, onde a estrutura do valor (no par chave-valor) é definido como um conjunto predefinido de colunas (Indrawan-Santiago, 2012).

A introdução dos primeiros conceitos associados à Web 3.0, projetos de Linked Data (Bizer et al., 2009) e a omnipresença das redes sociais, tem conduzido ao desenvolvimento das bases de dados baseadas na teoria dos grafos. A ideia desse modelo é representar os dados e/ou o esquema dos dados como grafos dirigidos ou como estruturas que generalizem a noção de grafos. No modelo de grafos as informações sobre a inter-conectividade ou a topologia dos dados são tão ou mais importantes quanto os dados propriamente ditos. O modelo orientado a grafos possui três componentes básicos: os nós (são os vértices do grafo), os relacionamentos (são as arestas) e as propriedades (ou atributos) dos nós e seus relacionamentos. Neste caso, a base de dados pode ser vista como um multigrafo rotulado e direcionado, onde cada par de nós pode ser conectado por mais de uma aresta.

A família de bases de dados orientadas a documentos utiliza o conceito de dados e documentos autocontidos e auto descritivos, o que implica que o documento em si já define como deve ser apresentado e o significado dos dados em cuja estrutura estão armazenados. Este tipo de bases de dados será abordado com maior detalhe na Secção 3.4.

Tabela 3.1: Lista de bases de dados NoSQL

Documento	Chave-Valor	Coluna	Grafo
MongoDB	Redis	BigTable	Neo4J
CouchDB	Membase	Hadoop / HBase	FlockDB
RavenDB	Voldemort	Cassandra	InfiniteGraph
Terrastore	MemcacheDB	SimpleDB	InfoGrid
Elasticsearch	Riak	Cloudera	HyperGraphDB

3.3 Bases de Dados Relacionais *vs.* Não Relacionais

As bases de dados SQL são essencialmente designadas por bases de dados relacionais (RDBMS) por contraste com as bases de dados NoSQL designadas por bases de dados não-relacionais ou distribuídas. Em termos gerais, uma das principais caraterísticas do NoSQL é a não obrigatoriedade de definição de uma estrutura rígida para os dados ao contrário das bases de dados SQL que representam os dados em forma de tabelas, onde os dados são dispostos em n número de linhas com m atributos correspondentes às colunas da tabela. O modelo relacional apresenta escalabilidade vertical através da melhoria do hardware utilizado, enquanto o modelo não-relacional tem na escalabilidade horizontal um dos seus pontos fortes, facilitando o incremento do número de servidores de bases de dados como forma de aumentar o volume de trabalho suportado pela base de dados. Outra das caraterísticas que diferenciam ambas as filosofias de base de dados é a linguagem usada na recuperação dos dados armazenados, dado que as bases de dados relacionais utilizam o padrão SQL (Structured Query Language), não existindo uma linguagem padrão para a manipulação dos dados em bases de dados não-relacionais, variando de base de dados para base de dados e a sintaxe utilizada é por vezes designada por UnQL (*Unstructured Query Language*) (Buneman et al., 2000). Em termos de propriedades as bases de dados relacionais asseguram o conjunto de propriedades ACID enquanto que as bases de dados não-relacionais asseguram as propriedades relacionadas com o Teorema de CAP. Ambos os conjuntos de propriedades, serão abordados nas Secções seguinte.

3.3.1 Bases de Dados Relacionais: Propriedades ACID

O conceito de transação é um aspeto fulcral quando se aborda a performance e consistência de bases de dados implementadas num ambiente de computação distribuída (McCreary and Kelly, 2013). Durante anos, os SGBDs relacionais que garantiam as propriedades ACID dominaram o mercado e respondiam às necessidades de armazenamento da época (Shim, 2012). Todavia, a atual explosão na produção de dados conduziu a diferentes necessidades de armazenamento bem como à alteração do paradigma de controlo de transações, assumindo as propriedades BASE (*Basically Available, Soft-state, Eventually consistent*). As tradicionais bases de dados relacionais são focadas nas transações ACID, devendo garantir as seguintes propriedades em simultâneo:

Atomicidade: todas as operações numa transação têm de terminar com sucesso, senão a transação é revertida. Não existe um estado intermédio que seja aceitável (Tiwari, 2011). Se uma parte da transação falha, toda a transação falha; só se todos os componentes da transação forem bem-sucedidos é que a transação é considerada bem-sucedida. Se algum componente da transação falhar, todas as alterações efetuadas até àquele ponto têm de ser revertidas e a base de dados

tem de voltar ao estado em que se encontrava antes de se iniciar a transação. A atomicidade tem que ser assegurada em toda e qualquer situação, como problemas de hardware/software ou falhas de energia (Mohanty et al., 2013);

Consistência: esta propriedade assegura que a execução de uma transação irá deixar a base de dados num estado válido. Tal significa que qualquer transação altera a base de dados de um estado válido para um outro estado igualmente válido, onde todos os dados escritos na base de dados necessitam de ser validados segundo as regras definidas (Leonard, 2013). Assim, os dados inseridos têm de respeitar as restrições impostas pelo esquema da base de dados, tipos de dados e integridade referencial de tabelas ou linhas de dados (Mohanty et al., 2013);

Isolamento: torna-se relevante quando determinados dados são acedidos por dois processos de forma concorrente (Tiwari, 2011). Cada transação tem de ser executada com total isolamento, logo duas transações a ocorrer ao mesmo tempo devem permanecer sem qualquer interação. Dependendo do nível de isolamento pretendido tal pode significar que o mecanismo de gestão da base de dados pode bloquear temporariamente a execução de uma transação até que outra tenha terminado (Mohanty et al., 2013).

Durabilidade: assegura que numa transação executada com sucesso, o seu resultado é guardado permanentemente (Tiwari, 2011), mesmo na eventualidade da ocorrência de erros ou falha do sistema. Pode implicar o registo da sequência de operações levadas a cabo pela transação numa outra localização, de modo a que em caso de erro seja possível repor a base de dados no estado em que se encontrava antes do início da transação (McCreary and Kelly, 2013).

3.3.2 Bases de Dados Não Relacionais: Teorema CAP e Propriedades BASE

Considerando as necessidades de escalabilidade das bases de dados NoSQL, as propriedades ACID não podem ser todas garantidas em simultâneo. Em 2000, Eric Brewer (2000) definiu que uma forma de assegurar as garantias ACID em sistemas distribuídos é compreender como os fatores Consistência, Disponibilidade e Tolerância à Partição têm impacto em tais sistemas, ficando conhecida pelo teorema CAP. Este teorema foi formalizado em 2002 por Gilbert and Lynch (2002) com base nos seguintes pressupostos:

Consistência: um serviço consistente deve garantir que as atualizações se propagam por todas as réplicas de um *cluster,* de forma ordenada resultando em que "o que se escreve é o que se lê", independentemente da localização. A propriedade de consistência está relacionada com a atomicidade e isolamento, implicando que todos os processos executados de forma concorrente visualizem a mesma versão dos dados (Tiwari, 2011);

Disponibilidade: o sistema deve estar disponível, garantindo que cada pedido recebe uma resposta de sucesso ou erro. Quando o sistema possa não estar disponível pode, no entanto, permanecer consistente. A consistência e disponibilidade não podem ser alcançadas em simultâneo. Diminuindo a importância da consistência permitirá que o sistema permaneça altamente disponível, mas uma forte consistência poderá significar que em certas condições o sistema não estará disponível;

Tolerância à Partição: o sistema continua a funcionar apesar da falha de uma parte do sistema. Considerando um conjunto de *n clusters*, se por algum motivo a rede não estiver disponível entre alguns dos nós, irá provocar a incapacidade sincronização dos dados, mas apenas uma parte do sistema não funcionará perdendo-se consistência (visto não haver a sincronização de todos os nós) ou disponibilidade (caos se desligue todo os sistema em consequência da falha).

O teorema CAP assume que qualquer sistema de bases de dados distribuído apenas garante, em determinado instante, duas das propriedades anteriores em simultâneo (Mohanty et al., 2013). A aplicação do teorema CAP apresenta três combinações possíveis:

- **Consistência e tolerância a partição, comprometendo a disponibilidade (CP)**: quando um nó falha o sistema pode ficar indisponível por tempo indeterminado, até que seja restaurado para um estado consistente. Este modelo é comumente utilizado em aplicações com transações financeiras, onde o tempo de execução é um fator crítico. Na maioria das ocorrências os sistemas conseguem recuperar autonomamente e replicar rapidamente informação, minimizando o período de indisponibilidade do sistema (Tiwari, 2011);
- **Consistência e disponibilidade, comprometendo a tolerância a partição (CA):** parte da base de dados não apresenta a preocupação de ser tolerante a falhas, recorrendo à replicação para garantir a disponibilidade e consistência da informação, situação verificada nas tradicionais bases de dados relacionais (Han et al., 2011);
- **Tolerância a partição e disponibilidade, comprometendo a consistência (AP):** em determinados casos a disponibilidade não pode ser comprometida e se o sistema for bastante distribuído também não é possível comprometer a tolerância à partição, podendo-se no entanto abdicar da consistência, passando o sistema a ser eventualmente consistente. Uma eventual consistência significa que, após a atualização de um atributo, "eventualmente" todos os nós do sistema irão sincronizar essa informação (Tiwari, 2011). Esta é uma das características basilares do modelo BASE apresentado de seguida.

As propriedades BASE (básico) surgiram na sequência do teorema CAP por oposição às propriedades ACID (ácido). A diferença entre ambos os conjuntos de propriedades reside no tipo de consistência garantido, dado que os sistemas ACID se focam na consistência e os sistemas BASE na disponibilidade. As propriedades BASE estão associadas aos sistemas NoSQL (Abramova and Bernardino, 2013), garantindo que o volume de dados produzido a cada instante é armazenado de imediato, mesmo que tal implique que o sistema possa ficar inconsistente por breves instantes (McCreary and Kelly, 2013).. Segue uma breve descrição de cada uma das propriedades BASE:

- **Basicamente disponível**: o sistema garante a disponibilidade dos dados de acordo com o teorema CAP. A resposta obtida pode ser "Falha" ou "Inconsistência", se os dados solicitados estiverem inconsistentes (Celko, 2013);
- **Estado flexível**: assume-se que os dados possam estar inconsistentes durante um período de tempo e inclusivamente podem ser alterados no momento em que são pesquisados ou utilizados (McCreary and Kelly, 2013);
- **Eventual consistência**: o sistema tornar-se-á eventualmente consistente dado que após um determinado período de tempo sem alterações, estas serão replicadas por todos os *clusters* do

Tabela 3.2: Tipos de dados JSON.

Primitivos	**Strings** unicode, delimitadas por aspas **Números** de dupla precisão Valores **booleanos**: verdadeiro ou falso **Null**, que representa o valor nulo
Estruturados	**Objetos**: conjunto de pares chave (string) valor (qualquer tipo) **Arrays**: lista de valores, cujos valores não têm de ser todos o mesmo tipo

sistema e todas as réplicas da base de dados assumirão o mesmo estado (McCreary and Kelly, 2013).

3.4 Bases de Dados Orientadas ao Documento

O modelo de dados orientado ao documento é considerado o mais flexível e popular de entre todos os modelos NoSQL (McCreary and Kelly, 2013). As bases de dados que implementam este modelo armazenam os dados em forma de documentos, essencialmente no formato de documentos XML, JSON ou PDF proporcionando boa performance ao mesmo tempo que possibilita elevada escalabilidade horizontal (Nayak et al., 2013). Este tipo de sistemas associa a cada documento uma chave única na forma de uma string, utilizando índices para tornar mais eficiente o acesso a documentos, dado que sempre que um novo documento é guardado, todo o conteúdo desse mesmo documento é indexado (McCreary and Kelly, 2013). Tal como nos índices das bases de dados relacionais, também neste tipo de bases de dados existe um compromisso entre a performance de escrita e uma maior performance de leitura, no entanto, os índices dos sistemas não-relacionais são bastante amplos, o que significa que todos os campos de um documento são indexáveis e pesquisáveis (McCreary and Kelly, 2013). Basta conhecer um campo de um documento, para assim conseguir encontrar rapidamente o mesmo campo em outros documentos. Documentos são agrupados em coleções (Kaur and Rani, 2013) e se se efetuar um paralelismo com as bases de dados relacionais, pode-se considerar que as tabelas do modelo relacional são o equivalente às coleções do modelo de armazenamento orientado ao documento, e como tal cada registo de uma tabela é equivalente a um documento de uma dada coleção. Estas bases de dados são bastante flexíveis por oposição ao modelo relacional, visto que documentos distintos numa mesma coleção podem ter diferentes campos, não sendo necessário desperdiçar espaço em disco em campos com valor nulo nos documentos em que não sejam necessários (Kaur and Rani, 2013). Outra diferença para o modelo relacional reside no fato de que uma tabela não pode possuir numa das suas células uma outra tabela. No armazenamento orientado ao documento, é possível criar coleções armazenadas dentro de uma outra coleção, ou um documento embebido noutro documento (McCreary and Kelly, 2013). Este tipo de bases de dados NoSQL são as mais utilizadas em aplicações web que necessitem de armazenar grande quantidade de dados semiestruturados, que em simultâneo necessitem de executar consultas dinâmicas a esses mesmos dados (Kaur and Rani, 2013).

O modelo de dados JSON, desenvolvido por Douglas Crockford (Crockford, 2006), segue um formato de texto que é independente do idioma mas que utiliza convenções comuns a diversas linguagens de programação, como por exemplo, Python, Perl, PHP (*PHP: Hypertext Preprocessor*) ou JavaScript. Documentos JSON podem com alguma facilidade ser comparados aos *arrays* associativos do PHP ou aos dicionários em Python. Os objetos JSON são constituídos por quatro tipos primitivos e dois tipos

```
1  {
2          "_id": "___",
3          "text": "Olá! Bom dia!",
4          "favorite_count": 0,
5          "retweeted": false,
6          "retweet_count": 0,
7          "user": {
8                  "id": "___",
9                  "followers_count": 603,
10                 "statuses_count": 20335,
11                 "friends_count": 216,
12                 "location": "Portugal, Sintra",
13                 "geo_enabled": true,
14         },
15         "geo": {
16                 "type": "Point",
17                 "coordinates": [38.7427123,-9.1546237]
18         },
19         "created_at": "Thu Feb 20 16:18:59 +0000 2014"
20 }
```

Figura 3.2: Excerto de um objeto JSON que representa um tweet.

estruturados, tal como se apresenta na Tabela 3.2.

A notação JSON tem sido adotada como o modelo de dados preferencial para as bases de dados NoSQL, permitindo um elevado grau de liberdade na definição dos campos e respetivos valores que constituem cada objeto, sendo que um valor de um objeto ou array pode ser de um tipo primitivo ou de um tipo estruturado, possibilitando o aninhamento de objetos e arrays em outros objetos do mesmo documento (Chasseur et al., 2013). Na Figura 3.2 encontra-se um um excerto de um objeto JSON, que representa um tweet, onde na linha 3 se encontra um par chave-valor cujo valor é uma *string*, na linha 4 o valor é um número inteiro e na linha 5 o valor da chave ”retweeted” é um booleano. As linhas de 8 a 15 representam um documento embebido dentro do corpo do tweet.

3.5 Hierarquia de Dados em MongoDB

MongoDB é um sistema NoSQL open source, orientado ao documento, escalável, com alta performance, suporta *full index*, distribuição, replicação e compatibilidade com o paradigma Map/Reduce. Foi desenvolvido em C++ e o seu nome advém da palavra anglo-saxónica *Hu**mongo**us*, normalmente utilizada para qualificar algo gigantesco. O suporte comercial é fornecido pela organização 10Gen, tendo iniciado o seu desenvolvimento em 2007 (Liu et al., 2007).

No contexto das bases de dados orientadas ao documento, o MongoDB é a base de dados Open Source com maior popularidade, sendo utilizada por grandes e pequenas empresas, como por exemplo o eBay, a Sourceforge, o The New York Times e o LinkedIn. No MongoDB os dados são armazenados no formato JSON, mais concretamente em BSON[2] (Binary JSON), explorando o facto de que a necessidade de mapeamento em tempo real dos objetos JSON, torna ineficiente o seu armazenamento em bases de dados relacionais (Chasseur et al., 2013). Os ficheiros BSON geridos pelo sistema não

[2]JSON em formato binário

podem ultrapassar um tamanho máximo de 16 Mb, normalmente agrupados em coleções segundo uma determinada estrutura, embora documentos com diferentes estruturas possam ser armazenados numa mesma coleção. No entanto, por efeitos de performance é aconselhável que determinada coleção contenha apenas documentos com estruturas similares (Abramova and Bernardino, 2013).

O armazenamento dos dados em MongoDB segue a hierarquia ilustrada na Figura 3.3: base de dados, coleções e documentos. Uma base de dados é um conjunto de coleções e uma coleção é um conjunto de documentos. Cada documento pode conter documentos embebidos ou ligados, tentando de alguma forma, mapear o conceito de JOIN de tabelas no modelo relacional, embora com repetição de informação e desperdício de espaço.

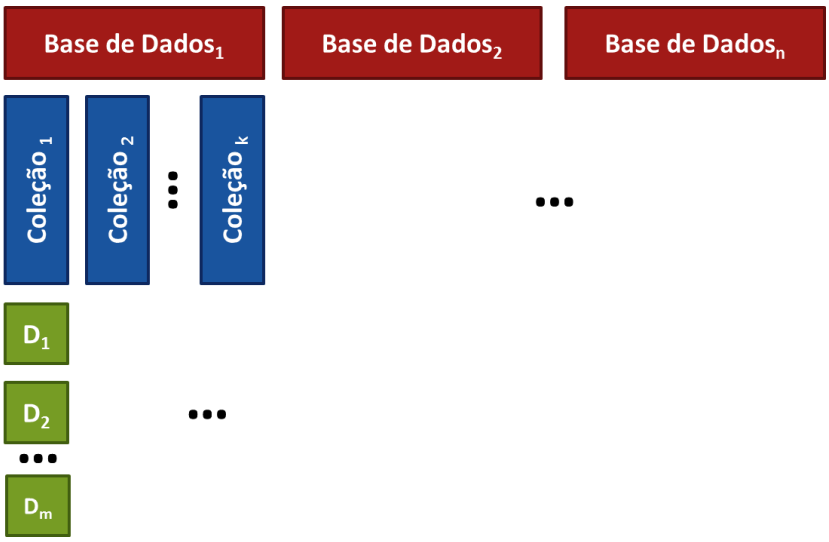


Figura 3.3: Hierarquia do modelo de dados para armazenamento de informação em MongoDB (adaptado de Kumar et al. (2013a)).

O MongoBD utiliza um sistema de indexação idêntico ao das bases de dados relacionais, onde cada documento é identificado por um campo chave com o nome "*_id*" com o qual é criado um índice que identifica univocamente cada documento. A indexação é imprescindível para a eficiência das operações de leitura, no entanto pode impactar negativamente a performance de operações de escrita de dados. Outros índices podem ser livremente criados sobre cada um dos campos dos documentos (Abramova and Bernardino, 2013). Uma das características mais relevantes do MongoDB é a forma como é gerido o mecanismo de concorrência e os processos que utiliza para garantir a durabilidade dos dados, através da criação de réplicas por definição de um nó mestre e de um ou vários nós escravos. O nó mestre pode ler ou escrever ficheiros enquanto que os nós escravos estão limitados a servir como *backups*, sendo apenas permitida a operação de leitura a esses nós. Se o nó mestre falhar, o nó escravo que possua a réplica de dados mais atualizada torna-se no nó mestre. Todas as réplicas são atualizadas assincronamente, assim alterações aos dados não são propagadas imediatamente (Abramova and Bernardino, 2013), logo a consistência dos dados enquadra-se no nível de "consistência eventual" dos sistemas que seguem as propriedades BASE. O MongoDB oferece nativamente a possibilidade de conexão com aplicações desenvolvidas em linguagens, como C, C++, Java, PHP

ou Python. Existem ainda várias frameworks e bibliotecas disponibilizadas por terceiros que permitem interagir com o MongoDB de forma simples, providenciando do rápido desenvolvimento de aplicações que utilizam o MongoDB como repositório de dados.

3.6 Sumário

As bases de dados NoSQL foram criadas com o objetivo de oferecer bom desempenho (tanto em termos de velocidade como em termos de tamanho) e disponibilidade, sabendo que deste modo não poderão garantir as propriedades ACID características das bases de dados relacionais, mantendo apenas as propriedades BASE. O MongoDB, como base de dados NoSQL orientada ao documento, é perfeitamente adequada ao armazenamento de tweets, não só pelo facto do formato dos tweets serem documentos JSON que também é o formato de armazenamento de dados do MongoBD não havendo necessidade de conversão, mas também por ser uma base de dados distribuída que vem ao encontro na necessidade de armazenar a grande quantidade de dados que se pretende recolher, usando múltiplas fontes.

4 Recolha e Expansão de Dados

> Twitter will no longer grant whitelisting requests.
>
> *Ryan Sarver*

> They're essentially saying if you can't get your data in real time or in less than 350 requests in an hour, find something else to do. Every API has some kind of rate limit and a lot of them don't have whitelisting.
>
> *Abraham Williams*

Num processo de investigação devem explicar-se, detalhadamente, os princípios metodológicos e técnicas utilizadas. Este Capítulo inicia na Secção 4.1 com uma breve referência aos principais conceitos relacionados com a dinâmica de publicação de mensagens no Twitter. Na Secção 4.2 são descritas as principais APIs que o Twitter disponibiliza como forma de acesso a uma parte da informação partilhada de forma pública pelos seus utilizadores, seguindo-se na Secção 4.3 a apresentação dos processos e configurações adotados para a recolha de tweets geolocalizados em Portugal dando origem a um *corpus* de tweets produzidos por utilizadores portugueses. Na Secção 4.4 são apresentados os métodos utilizados para a expansão do histórico de tweets recolhidos sobre cada um dos utilizadores portugueses. O Capítulo termina com a Secção 4.5 onde são referidas algumas otimizações introduzidas no processo de recolha de tweets.

4.1 Twitter

A restrição de 140 caracteres por tweet é motivada pelo interesse do Twitter em se manter compatível com o envio de mensagens SMS, através de dispositivos móveis. Este facto induz à escrita de uma forma não estruturada, muitas vezes recorrendo a abreviaturas e a URL encurtados por serviços como bitly.com, goo.gl ou o t.co, propriedade do próprio Twitter. Embora dispondo apenas de pouco mais de uma centena de caracteres, os utilizadores do Twitter expressam nas suas mensagens informações sobre os mais variados assuntos, nomeadamente notícias, anúncios, experiências, pensamentos ou convites.

O Twitter é utilizado com diversas finalidades, onde se incluem: a partilha de *web links;* relatos de notícias a que se presenciou ou propagação de novas informações sobre algo; comentários dirigidos a outra pessoa; transmissão de sentimentos recorrendo à escrita com *emoticons;* divulgação de alteração do local de habitação ou dos hábitos diários; filosofar, questionar ou indagar sobre determinado

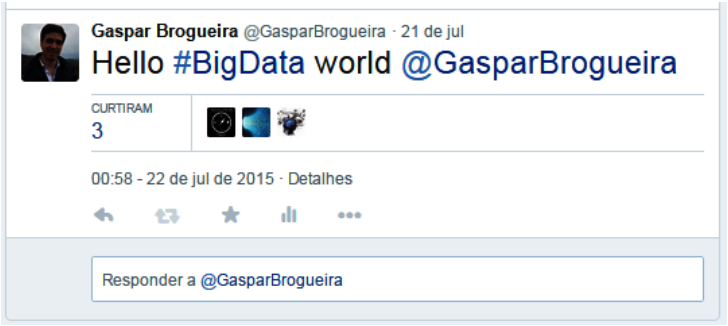


Figura 4.1: Anatomia de um tweet.

assunto; *crowdsourcing* ou organizar *flash mobs* e *tweetups* (*meetups* pessoais com amigos do Twitter). Além do envio e receção de mensagens, o Twitter disponibiliza outras funcionalidades extra. Neste conjunto de funcionalidades extra, está incluído o *retweet* (partilha) de qualquer mensagem produzida por outro utilizador ou o envio de mensagens diretamente entre utilizadores, mediante a colocação do caracter "@" antes do nome do utilizador de destino da mensagem. Para Yang and Rim (2014) a ação de *retweeting* é similar a reencaminhar um email, sendo fundamental para a propagação de mensagens no Twitter. A possibilidade de fazer *retweet* conduz à partilha instantânea de um tweet com todos os seguidores de determinado utilizador. Por vezes, adiciona-se RT no início de um tweet para indicar que se está a publicar novamente o conteúdo de um outro tweet. Não se trata de um comando ou recurso oficial do Twitter, significa apenas que se está a copiar o conteúdo do tweet de outro autor. Um *retweet* parece um tweet normal, com o nome do autor e o nome de utilizador ao lado, mas é diferenciado pelo ícone de *retweet* e pelo nome do utilizador que efetuou o *retweet*. Outra das funcionalidades do Twitter é a possibilidade de manter uma conversa com os demais membros da rede social, respondendo diretamente aos tweets de determinado utilizador ou mencionando o utilizador com quem se pretende comunicar, aquando da publicação de um tweet. Uma resposta é qualquer atualização desencadeada pelo clique no botão "Responder" de um tweet (ver Figura 4.1). Todo o tweet que seja uma resposta começa com o nome do utilizador (menção) a quem se responde. Uma menção corresponde à atividade no Twitter que contenha o nome do utilizador precedido do caracter "@" no texto do tweet, o que significa que a identificação de determinado utilizador numa resposta também é considerada uma menção, tal como no exemplo da Figura 4.1, onde aparece a menção ao utilizador "@Gaspar Brogueira". Se se incluir o nome de mais de uma pessoa no texto de um tweet e usar o formato *@nome_utilizador*, todas essas pessoas terão acesso ao tweet no seu menu de Menções. A criação de hastags (palavras ou frases precedidas do caracter "#") podem originar palavras-chave associadas a determinado assunto, tópico ou discussão e que se pretende indexar de forma explícita, facilitando a pesquisa por outros utilizadores. Na Figura 4.1 o texto do tweet apresenta a hashtag #BigData. As hashtags mais referenciadas no Twitter são agrupadas no menu *Trending Topics*, sempre disponível na interface do Twitter para qualquer utilizador. A frequência de ocorrência de menções (tweets em que se reconhece o caracter "@" seguido de um nome de utilizador) ou hastags, são elementos que podem caracterizar a utilização do Twitter em diversas línguas, tal como concluiu Weerkamp et al. (2011a).

4.1.1 Geolocalização

Normalmente, as aplicações de geolocalização podem fazer duas coisas: relatam a localização de determinado utilizador aos demais membros da comunidade e associam locais ou eventos de interesse (como restaurantes ou viagens) ao histórico de localizações onde o utilizador já esteve fisicamente. Aplicações de geolocalização executadas em dispositivos móveis oferecem uma experiência mais rica do que aplicações executadas em *desktops* porque os dados relevantes relativamente à localização variam bastante. Os *smartphones* atuais têm incorporado um *chip* GPS (Global Positional System), que usa dados de satélite para calcular a sua posição exata, tal como os serviços do Google Maps. Sempre que um sinal de GPS não está disponível, as aplicações de geolocalização podem usar informações das antenas de emissão/receção de sinal de rádio para triangular a sua posição aproximada, um método que não é tão preciso como o GPS, mas que tem vindo a ser bastante melhorado nos últimos anos.

A informação da geolocalização no âmbito das redes sociais pode, portanto, ser obtida de duas formas distintas. A primeira está relacionada com a informação que os utilizadores decidem partilhar publicamente na descrição do seu perfil, dando indicações relativamente à sua zona de residência. A segunda possibilidade está relacionada com informações obtidas do dispositivo utilizado para publicar o tweet. Tal é possível por dedução da localização através do endereço IP utilizando uma solução de geolocalização por IP, tais como as apresentadas por (Eriksson et al., 2010) e (Poese et al., 2011). Poderá também ser utilizada a triangulação de antenas de sinal de rádio, sendo que as antenas recebem o sinal de rádio de um determinado equipamento (*smartphone*) e com base nesse sinal e na posição conhecida das antenas, calcula a posição geográfica do equipamento. Outra possibilidade é através do sinal GPS, onde o equipamento determina a sua própria posição após conexão com pelo menos 4 satélites de um conjunto alargado de satélites. Com os dados recebidos, o equipamento pode calcular a sua posição, e por consequência é possível determinar a localização do utilizador que está a manipular o equipamento. Caso o céu esteja limpo, a aplicação de geolocalização do *smartphone* pode verificar a sua posição com razoável precisão. Em espaços interiores, no entanto, é menos preciso e em edifícios muito próximos, por vezes é necessário selecionar manualmente a localização.

A informação contida no perfil do utilizador é estática e por vezes pouco precisa. Qualquer utilizador ao criar o seu perfil no Twitter pode indicar como zona de residência a localidade de Lisboa, mas caso se desloque ao Porto e publique um tweet geolocalizado, a localização indicada no seu perfil não corresponderá à sua localização física aquando da publicação do tweet. Nos três exemplos da Figura 4.2 verifica-se que a localização que determinado utilizador partilha no seu próprio perfil através do campo *user.location* não tem obrigatoriamente correspondência direta aos diversos lugares de onde o utilizador vai publicando os seus tweets. Como tal, a informação presente no perfil de utilizador não é viável e como tal não será considerada.

Um dos focos desta dissertação, irá incidir na geolocalização através da informação obtida do equipamento usado na interação com o Twitter. A informação relativa à geolocalização pode ser partilhada segundo dois níveis de granularidade: pela precisão das coordenadas (longitude/latitude) do lugar exato onde a mensagem foi publicada ou pela descrição do lugar onde o utilizador se encontra, por exemplo, o nome da cidade. Como se depreende, a descrição do lugar é bem menos precisa que as coordenadas exatas do GPS, no entanto, ambas as informações fazem parte da generalidade dos tweets, tal como se ilustra na Figura 4.3.

Se os serviços de localização estiverem ativos, o Twitter poderá utilizar uma variedade de sinais para determinar a localização precisa do dispositivo móvel, como por exemplo, o sinal de GPS, o sinal

```
{
        "_id" : "___",
        "user" : { "location" : "paços de ferreira" },
        "place" : { "full_name" : "Vila Real, Vila Real" }
        ...
}

{
        "_id" : "___",
        "user" : { "location" : "Beja" },
        "place" : { "full_name" : "Matosinhos, Porto" }
        ...
}

{
        "_id" : "___",
        "user" : { "location" : "Lisboa" },
        "place" : { "full_name" : "Vila Franca de Xira, Lisboa" }
        ...
}
```

Figura 4.2: A informação da localização do utilizador na descrição do seu perfil nem sempre corresponde ao local de publicação de tweets.

da antena de rádio ou pelos dados sobre os pontos de acesso a rede sem fio nas proximidades do local onde o utilizador se encontra. Mesmo quando os serviços de localização não estejam ativos e caso se possua um ponto de acesso sem fios (por exemplo, uma rede sem fios), o Twitter poderá usar certas informações transmitidas publicamente por esse ponto de acesso, como o nome/SSID da rede, o endereço MAC, a frequência ou a intensidade do sinal, para permitir aos serviços do Twitter, a localização da publicação de determinado tweet.

4.2 Twitter API

Uma Web API permite que uma dada aplicação partilhe os seus dados com aplicações desenvolvidas por terceiros, abstraindo a forma de acesso e facilitando a utilização dos mesmos. Os dados retornados por uma API são estruturados de modo a facilitar a sua análise e o processamento da informação pretendida. As APIs tendem a separar todas as suas funções em ações específicas individuais, como por exemplo, “obter uma lista de tweets“ ou “obter a lista de *followers* de determinado utilizador“. O desenho da Twitter API segue os princípios do sistema RESTful (*REpresentational State Transfer*) ou simplesmente REST, concebido por Fielding (2000). Esta abordagem aumenta a facilidade de desenvolvimento, bem como a escalabilidade e flexibilidade de aplicações.

Uma REST API é desenvolvida tendo por base a invocação de serviços Web recorrendo ao protocolo de comunicação HTTP (*Hypertext Transfer Protocol*). Este protocolo é baseado no modelo computacional cliente-servidor, mediando a troca de pedidos e respetivas respostas entre determinada aplicação cliente (p.ex. navegador web) e um servidor onde, por exemplo, se encontra alojada uma aplicação web que atende aos pedidos do cliente. O cliente envia um pedido HTTP para determinado servidor que disponibiliza um certo tipo de recursos, essencialmente no formato HTML, retornando uma mensagem de resposta para o cliente. A resposta contém informações de estado sobre o pedido e pode também

```
{
        "_id": "___",
        "coordinates": {
               "coordinates": [-9.1546237,38.7427123]
        },
        "geo": {
               "coordinates": [38.7427123,-9.1546237]
        },
        "place": {
               "country_code": "PT",
               "country": "Portugal",
               "place_type": "city",
               "bounding_box": {
                      "coordinates": [
                         [
                               [-9.2298264,38.6913748],
                               [-9.2298264,38.7958529],
                               [-9.0901639,38.7958529],
                               [-9.0901639,38.6913748]
                         ]
                       ]
                },
                "full_name": "Lisboa, Lisboa",
                "name": "Lisboa"
        },
}
```


Figura 4.3: Excerto de um tweet evidenciando os dados que permitem a geolocalização.

conter o conteúdo solicitado no corpo de sua mensagem. O protocolo HTTP define oito métodos (GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS e CONNECT) que indicam a ação a ser realizada sobre o recurso especificado. Cada método determina a ação despoletada no servidor indicando o que este deve fazer com o URL fornecido no momento do pedido enviado sobre determinado recurso.

A Twitter API permite três tipos de pedidos HTTP: GET, POST e DELETE. O método GET aceita um URL, através do qual solicita o acesso a determinada informação de um servidor. Se o URL é uma página Web PHP por exemplo, o método GET irá receber como resposta o HTML gerado pela interpretação do PHP e não o código PHP. O método POST tem um comportamento idêntico ao GET, adquirindo os seus resultados de uma forma diferente. Existe um limite máximo para o tamanho do URL de um pedido GET, sendo que um pedido POST encapsula os dados a enviar para o servidor, permitindo que mais informações sejam transferidas da aplicação cliente para o servidor. O envio de dados num pedido POST pode ser necessário nas funções da API que efetuam alterações nos servidores do Twitter, em vez de apenas a recuperação de dados. É uma ação típica e disponibilizada de uma forma geral por todas as Web APIs, não sendo exclusiva do Twitter. A invocação do método DELETE tem como objetivo indicar ao servidor a remoção do acesso ao recurso referenciado pelo URL enviado no pedido.

A maioria das invocações à Twitter API requerem autenticação com um nome de utilizador e uma senha válida. A autenticação é necessária por duas razões: em primeiro porque algumas das informações disponibilizadas pela API são específicas do utilizador autenticado e em segundo lugar porque a autenticação é a forma mais confiável de aplicar a taxa de limitação de acessos à API. A Twitter API pode ser acedida apenas com pedidos autenticados. O Twitter utiliza como método de autenticação o

OAuth[1] (*Open Authentication*) e cada pedido é autenticado com as credenciais de um utilizador com perfil registado no Twitter.

A Twitter API pode ser dividida em três classes, tendo em consideração o método de acesso aos dados: Search API, Streaming API e REST API (Oussalah et al., 2013). As características de cada API variam em função do tipo e da quantidade de informação que disponibilizam.

A Search API é indicada para a pesquisa de tweets publicados num período de tempo próximo à data da pesquisa. É dada a possibilidade de efetuar múltiplas pesquisas num único pedido, enviadas numa lista separadas por vírgulas. Na resposta à pesquisa são devolvidos os tweets que estão de acordo com a consulta efetuada. A REST API pode ser utilizada para obter uma fração dos tweets publicados recentemente por determinado utilizador. A Streaming API fornece um fluxo contínuo de tweets públicos, podendo os resultados serem filtrados por *hashtags*, palavras-chave, identificações de utilizador (nome, *id*) ou por regiões geográficas.

Dado que no decorrer dos próximos Capítulos será abordada a utilização da Streaming API e a REST API, as Secções seguintes apresentam ambas as APIs de forma mais detalhada.

4.2.1 Twitter Streaming API

A Streaming API disponibiliza o acesso quase em tempo real a um fluxo de tweets públicos, incluindo respostas e menções criadas por contas públicas, exigindo uma conexão persistente a um *streaming endpoint*, que após ser estabelecida, mantém continuamente o acesso ao fluxo de tweets. Para este efeito, o processo que comunica com a Streaming API deverá executar em segundo plano no sistema. Esta API é baseada no protocolo HTTP utilizando principalmente os comandos GET, POST e DELETE para aceder e manipular os dados.

O Twitter disponibiliza diversos *streaming endpoints*, agrupados em função do seu tipo de utilização:

- ***Public Streams***: adequado para o acesso a dados públicos, acompanhando utilizadores e temas específicos ou para tarefas de mineração de dados;
- ***User Streams***: acesso a aproximadamente todos os dados de um único utilizador do Twitter;
- ***Site Streams***: versão multi-utilizador da *User Stream*, destinado a aplicações que se conectam ao Twitter para obter informações em tempo real de muitos utilizadores.

As *Public Streams* são constituídas por três *endpoints* que oferecem amostras dos dados públicos partilhados no Twitter:

- ***POST statuses/filter***: devolve os tweets públicos de acordo com um ou mais filtros. Podem ser especificados múltiplos parâmetros, pelo que na maioria das aplicações apenas uma conexão é suficiente para se alcançar os dados pretendidos. São permitidas as operações GET e POST, no entanto, os pedidos GET com demasiados parâmetros podem causar a rejeição do URL devido à limitação do número de caracteres. A invocação pelo método POST evita esta questão;

[1] http://oauth.net/

- ***GET statuses/sample***: retorna uma amostra aleatória de todos os tweets públicos. A amostragem é igual nos casos em que a autenticação utilizada tem o nível de acesso padrão, pelo que, dois clientes diferentes que se conectam a este *endpoint*, irão receber o mesmo conjunto de tweets;
- ***GET statuses/firehose***: retorna todos os tweets públicos, mediante o necessário nível de acesso. Em conjugação com outros recursos, com diversos níveis de acesso, podem permitir o acesso a quase toda a informação partilhada pelos diversos utilizadores no Twitter.

O tipo de fluxo de dados da *statuses/sample* retorna uma amostragem aleatória de tweets numa percentagem estatisticamente válida do total de tweets produzidos a nível global. A informação retornada por esta API é baseada em tweets partilhados na rede social, a partir de contas não protegidas. Os tweets criados por utilizadores com contas protegidas e a troca de mensagens pessoais diretamente entre utilizadores não estão disponíveis na Streaming API (Kendall et al., 2011).

Com a *statuses/filter*[2] é permitido o acesso a um fluxo contínuo de tweets públicos, que estão de acordo com algum critério de seleção, como por exemplo, palavras-chave, *hashtags*, ID de utilizador, nome do utilizador ou regiões delimitadas geograficamente (Kumar et al., 2013a). O acesso a tweets geolocalizados é, portanto, efetuado mediante a recolha dos tweets disponibilizados pelo *endpoint statuses/filter*, utilizando na filtragem dos tweets o parâmetro *locations.* Este parâmetro permite indicar uma lista, separada por vírgulas, de pares de valores correspondentes à longitude e latitude que especifica a delimitação de regiões geográficas. O valor para a longitude e latitude correspondente ao sudoeste vem em primeiro lugar na lista. Apenas os tweets cujos autores tenham autorizado a sua geolocalização e que tenham sido produzidos nas regiões indicadas no parâmetro *locations* serão incluídos no retorno da *statuses/filter* API.

A Twitter Streaming API foi concebida para proporcionar o acesso a volumes limitados de informação, verificando-se a preferência de utilização da Streaming API pela natureza de *streaming* dos dados retornados que estão bastante perto de serem considerados em tempo real. No entanto, a Streaming API não foi projetada para fornecer resultados com uma cobertura total e por isso tem algumas limitações: i) o volume de tweets acessível através do *endpoint statuses/filter* é limitado de modo que nunca seja superior a uma determinada percentagem do volume de dados disponibilizados no *endpoint statuses/firehose*. Como resultado, apenas as consultas de baixo volume de dados retornados podem ser acomodadas de forma confiável; ii) o Twitter impõe um limite no número de consultas que é permitido efetuar em cada pedido HTTP à API. Cada pedido admite como parâmetros o máximo de 400 palavras-chave, 25 delimitações de regiões geográficas ou 5000 IDs de utilizador. Este é um desafio significativo para quem pretende o acesso a grandes quantidades de informação; iii) os operadores *booleanos* não são suportados por esta API, tal como o são pela API do GNIP[3], por exemplo. E, finalmente, não há garantia de que os níveis de acesso do Twitter permaneçam inalterados no futuro. Empresas que desenvolvem aplicações que necessitem de acesso garantido aos dados ao longo do tempo, devem compreender que a construção de um negócio baseado no acesso livre a todos os dados através de APIs públicas pode ser um pouco arriscado.

O total de informação retornado pela *statuses/filter* API pode variar entre 1% a 10% do volume total de tweets produzidos no Twitter, em determinado instante (Kumar et al., 2013b). A variação na percentagem de dados a que se pode ter acesso, está relacionado com o nível de acesso tendo em

[2]https://dev.twitter.com/streaming/reference/post/statuses/filter
[3]https://gnip.com/

consideração as credenciais de autenticação utilizadas. Na versão 1.0 da Twitter API era permitido o acesso a cerca de 10% do volume total de tweets, utilizando IP's inseridos na *whitelist* do Twitter (Goonetilleke et al., 2014). O volume de dados disponibilizados foi drasticamente reduzido na versão 1.1 da Twitter API. A taxa de acesso à Streaming API designada por "spritzer", está atualmente fixada entre 1% a 2% do volume total de tweets publicados num determinado instante (Van Kleek et al., 2012). Considerando que, segundo estatísticas recentes, são produzidos diariamente a nível global cerca de 500 milhões de tweets, poderão ser recolhidos da s*tatuses/filter* cerca de 5 milhões de tweets, correspondentes ao 1% do volume total de tweets. Dependendo do tipo de dados a que se pretenda aceder, a devida conjugação dos parâmetros admitidos por esta API, poderá permitir o acesso praticamente total ao volume de informação partilhada sobre determinado contexto.

4.2.2 Twitter REST API

A Twitter REST API[4] permite o acesso a uma fração de informações e atualizações de *status* dos utilizadores, à lista de *friends* e *followers* do utilizador autenticado, permitindo uma infinidade de oportunidades de integração e interação com o Twitter, através da publicação de tweets, resposta a tweets, fazer *retweet* noutros tweets e muitas outras possibilidades. O acesso a cada tipo de informação ou funcionalidade é fornecido por um *endpoint* específico, como é o caso de POST *statuses/update* que permite a publicação de um novo tweet ou GET *followers/ids* que retorna a lista de seguidores do utilizador autenticado.

O *endpoint* GET *statuses/user_timeline* retorna o histórico de mensagens produzidas por determinado utilizador (timeline), até ao máximo de 3200 tweets. A recolha da timeline de um utilizador em específico, impõe que o mesmo seja indicado explicitamente pelos parâmetros *user_id* ou *screen_name*. Utilizadores com elevada atividade de produção de tweets rapidamente fazem crescer a sua timeline, havendo limite no número de tweets da timeline que são retornados a cada pedido, não sendo retornados os 3200 tweets num só pedido.

Além dos parâmetros *user_id* ou *screen_name* essenciais para definir o utilizador do qual se pretende recolher a timeline, o parâmetro *count* especifica o número de tweets retornados na resposta à invocação do *endpoint*, até ao máximo de 200 tweets. No caso limite em que a timeline de determinado utilizador contém um número de tweets igual ou superior a 3200, a recolha do máximo de tweets permitidos pressupõe no mínimo 16 invocações ao *endpoint* GET *statuses/user_timeline*, sendo que no parâmetro *count* deverá ser indicado o valor 200.

O acesso à Twitter API é limitado também no número de pedidos que é possível efetuar num determinado intervalo de tempo, designado por *rate limit*. A janela de tempo correspondente ao *rate limit* é de 15 minutos na versão 1.1 da Twitter API e é utilizada para renovar periodicamente a quota de invocações permitidas à API, sendo permitido efetuar 180 pedidos à API em cada período com autenticação do nível de utilizador ou 300 pedidos no caso de autenticação com nível de aplicação (Kumar et al., 2013a). As contas[5] (ou aplicações) com as quais é permitido o acesso à API do Twitter, deverão ser criadas e devidamente registadas no Twitter.

[4] https://dev.twitter.com/rest/public
[5] https://www.apps.twitter.com/

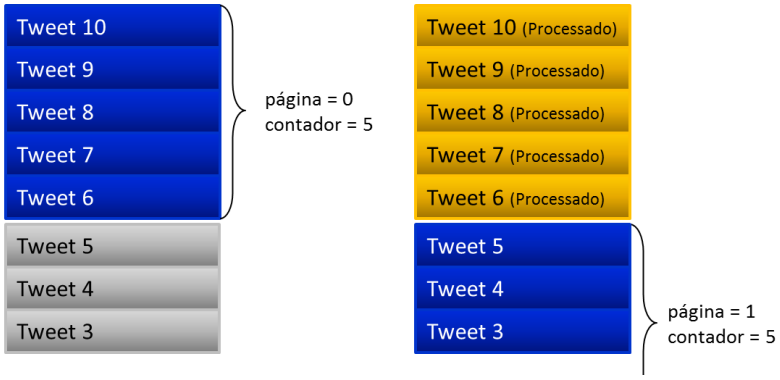


Figura 4.4: Leitura de uma timeline com 10 tweets pelo método da paginação (adaptado de (Twitter, 2015b)).

4.2.2.1 Processamento de Timelines

Além da API GET *statuses/user_timeline,* as APIs GET *statuses/home_timeline* ou GET *search/tweets*, retornam uma coleção de tweets e *retweets* recentemente publicados por determinado utilizador e pelos utilizadores que o seguem ou apenas uma coleção de tweets que estão de acordo com um filtro utilizado numa pesquisa. As timelines podem crescer muito rapidamente e, como tal, existem limites quanto à quantidade de tweets de uma timeline a que uma aplicação de terceiros pode aceder num único pedido. A lógica consiste, portanto, em iterar sobre subconjuntos de tweets da timeline, a fim de reunir uma lista mais completa de tweets. Considerando a natureza de tempo real do Twitter assim como o volume de dados que está constantemente a ser adicionado às timelines, abordagens de iteração sobre conjuntos de resultados de forma paginada nem sempre são eficazes.

Em teoria a paginação é simples de implementar. Considere-se o caso em que uma timeline contém 10 tweets ordenados de forma decrescente cronologicamente, ou seja, ordenados decrescentemente em função do instante em que o tweet foi publicado ou em função do ”id” que é incrementado a cada tweet publicado. Tal como demonstra a Figura 4.4, uma aplicação pode tentar ler esta timeline em dois pedidos definindo um tamanho de página de 5 tweets, solicitando a primeira página e de seguida a segunda página.

Esta abordagem não é de todo a mais eficiente, visto que as timelines são constantemente atualizadas com novos tweets dos respetivos utilizadores, ou seja, prosseguindo com o cenário da Figura 4.4, caso sejam adicionados dois novos tweets à timeline entre a primeira e a segunda invocação à API para a leitura de uma página, a segunda invocação irá ler dois tweets que já tinham sido lidos e processados na primeira invocação, tal como se pode observar na Figura 4.5. Na situação limite, em que entre as duas invocações fossem produzidos 5 tweets, a segunda invocação retornaria apenas dados repetidos, logo este método é ineficiente e redundante.

A solução para o problema descrito, consiste na leitura da timeline em relação ao topo da lista ordenada de tweets, que ao mudar frequentemente, deverá ser lida em relação ao *id* dos tweets já processados. Isto é possível através da utilização do parâmetro *max_id,* que introduz um filtro adicional na leitura da timeline, sendo apenas retornados os tweets cujos *ids* são inferiores ao valor indicado no parâmetro *max_id.*

A utilização correta do parâmetro *max_id* pressupõe que na primeira invocação ao endpoint da timeline só se deva especificar o número de tweets que se pretende ler. Após completado o processa-

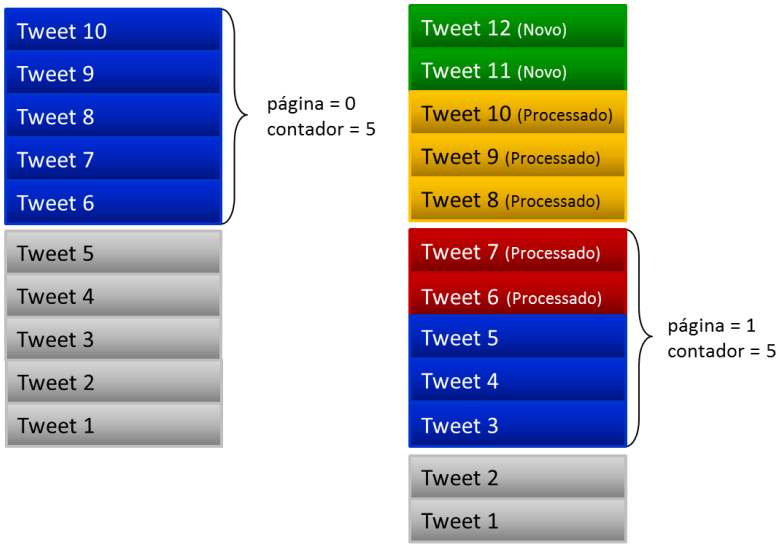


Figura 4.5: Repetição de tweets entre a leitura de duas páginas da timeline (adaptado de (Twitter, 2015b)).

mento da resposta de uma leitura da timeline, deve ser indicado nas invocações seguintes o parâmetro *max_id* com o valor do menor tweet *id* retornado na invocação anterior. Desta forma garante-se a leitura de tweets com *ids* inferiores ou iguais ao valor do parâmetro *max_id*. Embora tenha sido minimizada a leitura de tweets repetidos, a possibilidade de redundância persiste. Ou seja, na resposta é incluído o tweet cujo *id* é igual ao parâmetro *max_id*, tal como é ilustrado na Figura 4.6, ocorrendo a repetição da leitura de um tweet.

Embora a redundância de um tweet possa não ser justificação suficiente para considerar esta abordagem ineficiente, é possível otimizar um pouco mais. A otimização consiste em subtrair 1 (um) ao menor tweet *id* retornado na solicitação anterior e usar esse valor no parâmetro *max_id*. É irrelevante o facto do valor de *max_id* corresponder a um tweet *id* válido no contexto da timeline que se está a ler ou se corresponde a um tweet publicado por outro utilizador, visto que o valor de *max_id* é apenas usado na decisão de quais os tweets a filtrar. Desta forma, é possível ler e processar uma timeline completa sem redundância, como é possível confirmar pela Figura 4.7.

Outra otimização possível, relaciona-se com o facto de que as aplicações que processam as timelines, efetuam o processamento de forma iterativa, aguardando um certo intervalo de tempo entre cada iteração, processando os novos tweets que entretanto foram adicionadas desde a última vez que a timeline foi processada. A otimização passa por utilizar o parâmetro *since_id*. Prosseguindo com o exemplo anterior, foram adicionados à timeline os tweets de 11 a 18 desde o processamento inicial. Uma abordagem ineficiente para processar os novos tweets seria ler desde o tweet mais recente da timeline até ao tweet 10, mostrado na Figura 4.8, o que provoca a leitura repetida de dois tweets.

Esta particularidade pode ser evitada definindo o parâmetro *since_id* com o maior tweet *id* de todos os tweets já processados. Ao contrário do parâmetro *max_id*, o valor indicado em *since_id* não é incluído na pesquisa de tweets, por isso não é necessário decrementar o valor de *since_id* em uma unidade. Como se pode observar na Figura 4.9, serão processados apenas tweets com *ids* superiores ao valor do parâmetro *since_id*. A correta utilização dos parâmetros *max_id* e *since_id* minimiza o processamento de dados redundantes, mantendo a capacidade de leitura sobre todo o

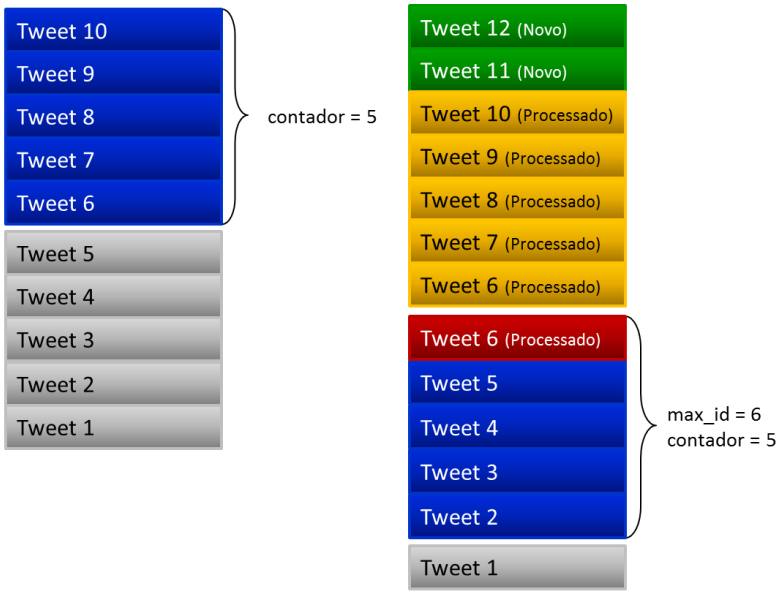


Figura 4.6: Leitura de um tweet repetido cujo *id* é igual ao valor do parâmetro *max_id* (adaptado de (Twitter, 2015b)).

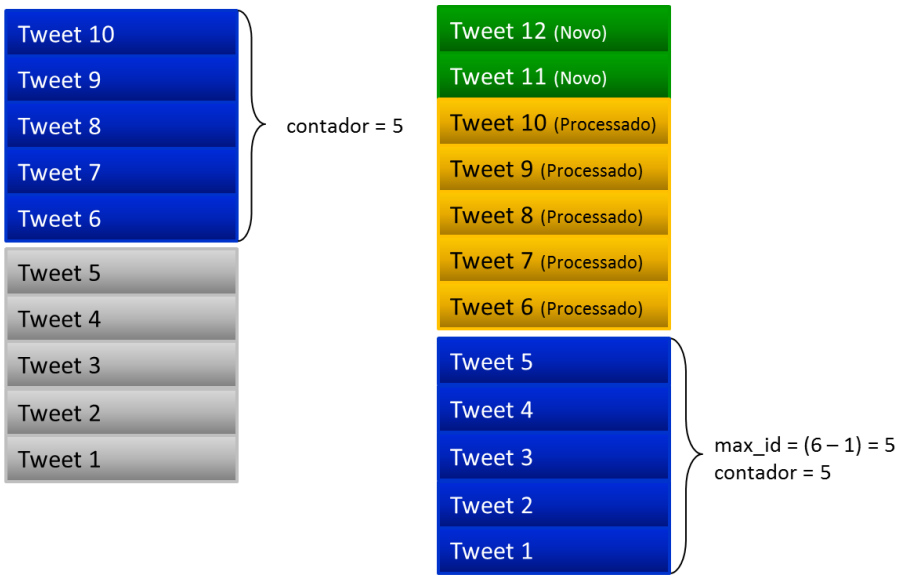


Figura 4.7: Ajuste do parâmetro *max_id* para evitar a leitura de tweets repetidos (adaptado de (Twitter, 2015b)).

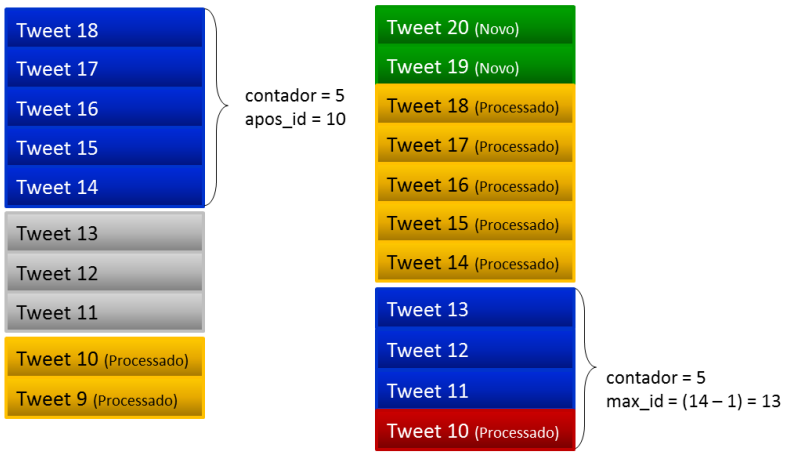


Figura 4.8: Repetição de tweets em leituras da timeline espaçadas no tempo (adaptado de (Twitter, 2015b)).

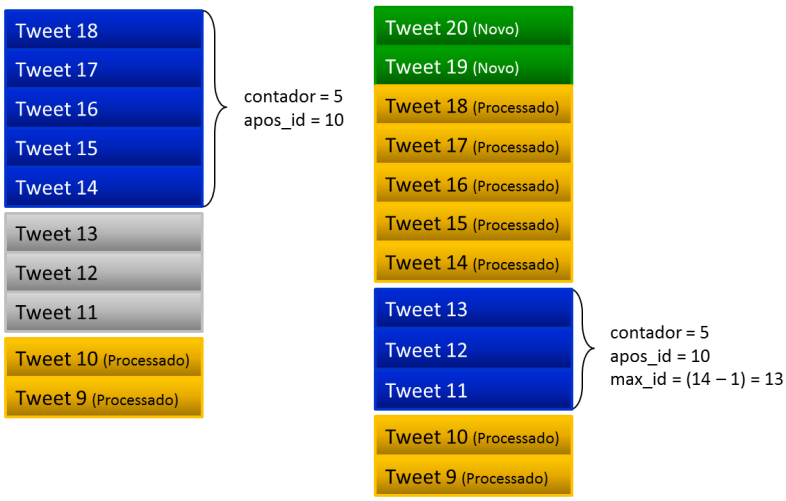


Figura 4.9: Utilização do parâmetro *since_id* para evitar a repetição na leitura de tweets (adaptado de (Twitter, 2015b)).

conteúdo disponível de uma timeline.

4.3 Processo de Recolha de Tweets

O Twitter permite o acesso aos dados da sua API por aplicações de terceiros após autenticação pelo método OAuth, tal como mencionado na Secção 4.2. A utilização deste método de autenticação requer a criação e o registo de uma aplicação associada a uma conta de utilizador do Twitter em `https://apps.twitter.com`. O registo da aplicação é concluído com a geração das chaves que permitem a autenticação na Twitter API. Para referência a uma aplicação que utiliza o método OAuth para efetuar a autenticação no Twitter a fim de proceder à recolha de dados, será utilizado o conceito de ”cliente”, seguindo a lógica do modelo Cliente-Servidor, onde os clientes são as aplicações registadas no Twitter e o Servidor o próprio Twitter, nomeadamente a Twitter API. Esta secção começa por apresentar as opções tomadas com vista a minimizar as limitações da Twitter API descritas na Secção 4.2. De seguida

```
{
        "_id": String,
        "user": Integer,
        "client": Integer,
        "timestamp": ISODate,
        "timeline_status": String,
        "APIKey": String,
        "APISecret": String,
        "AccessToken": String,
        "AccessTokenSecret": String
}
```

Figura 4.10: Estrutura JSON com a informação de autenticação de um cliente na Twitter API.

descreve como foram delimitadas as coordenadas geográficas e finalmente descreve o processo de recolha de tweets geolocalizados.

4.3.1 Gestão dos *Access Tokens*

Para realizar este trabalho foram registadas diversas aplicações associadas a perfis de utilizadores distintos, dadas as limitações impostas pelo Twitter relativamente à utilização da sua API, referentes tanto ao volume de informação que é disponibilizada como ao número de invocações permitidas num determinado intervalo de tempo. Com recurso a diversos clientes em simultâneo, é possível a paralelização de alguns dos procedimentos apresentados nas Secções seguintes, capturando-se um maior volume de dados por unidade de tempo.

As chaves de autenticação (*access tokens*) de cada cliente (*Consumer Key, Consumer Secret, OAuth Access Token e OAuth Access Token Secret*) são guardadas e geridas numa coleção MongoDB, segundo a estrutura JSON apresentada na Figura 4.10.

Os campos *user* e *client* não têm significativa relevância no contexto do trabalho, servindo apenas para fazer corresponder o *access token* ao perfil de utilizador registado no Twitter a que este se encontra associado. O campo *timestamp* contém a data correspondente ao instante em que foi terminado com sucesso o processamento da resposta vinda da API após invocação da mesma. Nos casos em que a invocação termina em erro, o campo *timestamp* não será atualizado. Este campo tem bastante relevância considerando a sua múltipla utilização, permitindo detetar quando determinado cliente se encontra ocioso, ou seja, quando este não efetua qualquer pedido à Twitter API com sucesso num período de 5 minutos. Poderá deduzir-se que algo decorreu de forma inesperada, terminando-se a execução do referido cliente e dando-se início à execução de um novo cliente com as credenciais de acesso, que se encontravam inativas nos últimos 5 minutos. Sempre que é instanciado um novo cliente, é dada prioridade à utilização do *access token* cuja data registada no campo *timestamp* seja a mais antiga do conjunto de todos os *access tokens*.

O campo *timeline_type* está relacionado, como se verá mais adiante, com o tipo de parâmetros a utilizar na invocação da API *statuses/user_timeline*. Os campos *APIKey*, *APISecret*, *AccessToken* e *AccessTokenSecret* contêm as *strings* geradas pelo Twitter que serão utilizadas no processo de autenticação na Twitter API pelo método OAuth.

Como se verá na Secção 4.4 a pesquisa e leitura de informação na Twitter API é inviável utilizando

Tabela 4.1: Alocação de *access tokens* a diferentes tipos de tarefas.

Access tokens	**Utilização**
1	Recolha contínua de tweets geolocalizados do fluxo da Streaming API
1	Leitura integral da timeline dos novos utilizadores integrados na base de dados
1	Processo de ”recuperação” de timelines dos utilizadores em estado de erro
12	Atualização de timelines

apenas um *access token,* dadas as limitações impostas pelo próprio Twitter. Com base neste pressuposto, tomou-se a opção de explorar a possibilidade de criação de diversas aplicações e respetivos *access tokens* associados a um mesmo perfil de utilizador do Twitter. Por cada perfil é permitido registar 7 aplicações, pelo que recorrendo a dois perfis distintos foram criadas 14 aplicações, utilizadas exclusivamente do processo apresentado na Secção 4.4. No processo descrito na Secção 4.3.3 foi utilizado um outro *access token* associado a um terceiro perfil de utilizador no Twitter, sendo também este utilizado exclusivamente apenas neste processo. A utilização associada a cada *access token* é referida na Tabela 4.1, sendo apresentada a descrição detalhada de cada uma das referidas tarefas, na continuação deste Capítulo.

4.3.2 Delimitação de Coordenadas Geográficas

Na recolha dos tweets geolocalizados produzidos em Portugal Continental e nos Arquipélagos dos Açores e da Madeira, foi utilizada a Streaming API *statuses/filter,* indicando no parâmetro *locations* as coordenadas que delimitam a região geográfica pela qual será efetuada a filtragem dos tweets devolvidos pela API. A delimitação do território Português foi definida segundo os dados retornados pelo próprio Twitter através da REST API *geo/search*[6]. Na invocação da REST API *geo/search* foram utilizados os parâmetros *granularity* e *query* com os valores ”country” e ”Portugal”, respetivamente, obtendo-se as coordenadas geográficas com as quais é possível definir um polígono que delimita o território Português. Parte da informação retornada é apresentada na Figura 4.11, sendo indicado nas linhas 11 a 15 os quatro pontos geográficos que permitem delimitar o território Português tal como é visível na Figura 4.12.

É possível observar na Figura 4.12 que para além da inclusão de Portugal Continental e dos Arquipélagos dos Açores e da Madeira, fazem parte da área considerada pelo conjunto de coordenadas, algum território de Espanha junto à fronteira norte e interior de Portugal e uma considerável extensão de Marrocos, o que irá originar o retorno de tweets para além dos produzidos em Portugal. Tendo em vista o objetivo de obter um *corpus* de tweets publicados somente em Portugal, na Secção 4.3.3 é descrito o método utilizado na filtragem dos tweets recolhidos, de modo a considerarem-se apenas os que efetivamente são produzidos em território português.

[6] *https://dev.twitter.com/rest/reference/get/geo/search*

```
{
    "places": {
        "id": "8198e85105936d3c",
        "place_type": "country",
        "name": "Portugal",
        "country_code": "PT",
        "country": "Portugal",
        "centroid":  [-8.27647113432668,39.5589331],
        "bounding_box":  {
          "coordinates":  [ [
                [-31.2688154,30.0302839],
                [-31.2688154,42.1542048],
                [-6.1902091,42.1542048],
                [-6.1902091,30.0302839],
                [-31.2688154,30.0302839] ]
        (...)
}
```


Figura 4.11: Delimitação geográfica de Portugal retornada pela REST API *geo/search.*

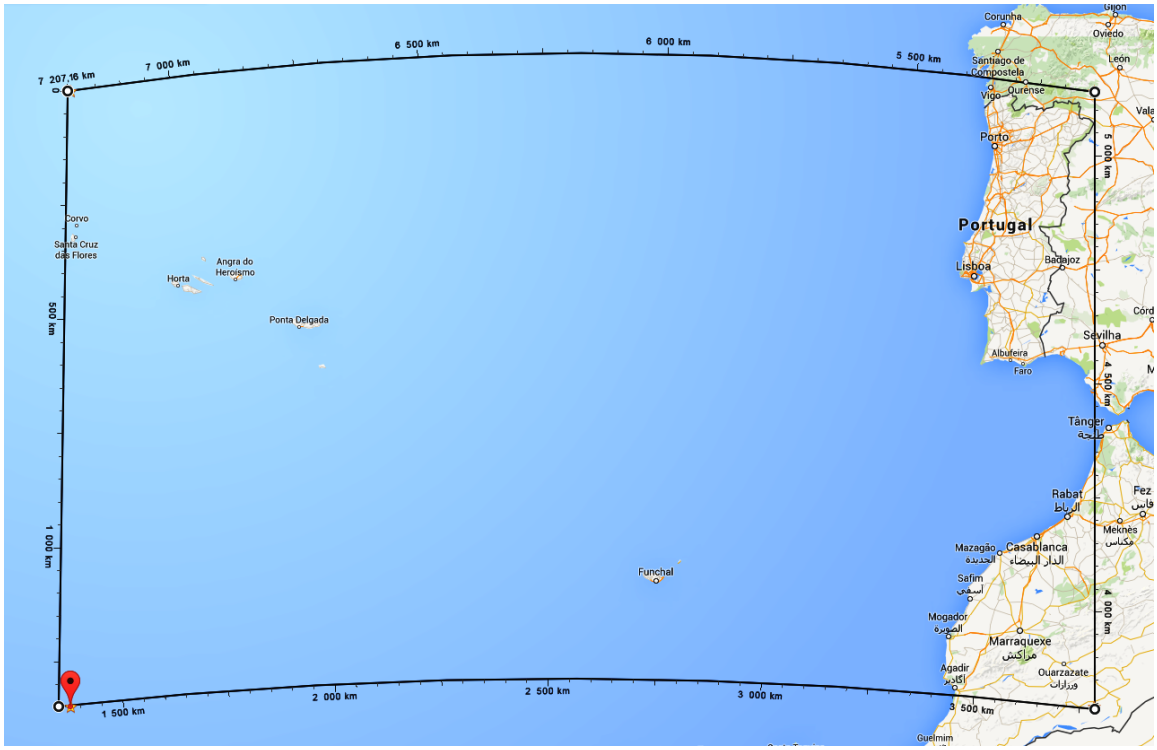


Figura 4.12: Representação da área considerada na recolha de tweets geolocalizados.

4.3.3 Recolha de Tweets Geolocalizados

Existem diversas alternativas para colecionar dados provenientes da Twitter API: i) Search API; ii) Streaming API ou iii) REST API, tal como foi referido na Secção 4.2. Cada API tem características distintas, quanto ao tipo e à quantidade de informação que devolve. A Streaming API, ao possibilitar a extração de dados do Twitter de acordo com determinadas delimitações geográficas, é a API indicada para a recolha de tweets geolocalizados. De entre os diversos *endpoints* da Streaming API, foi utilizada a Streaming API *statuses/filter*. Os tweets são lidos da *statuses/filter* por um script Python, que representa um cliente da Twitter API. Após autenticação com sucesso e envio do primeiro pedido HTTP é disponibilizado o acesso ao fluxo de tweets produzidos em tempo real, permanecendo o acesso ativo continuamente sem necessidade de posterior interação por parte do cliente. O cliente deverá igualmente permanecer em execução, de modo a tratar os tweets recebidos da *statuses/filter* API. O *access token* associado a este cliente, é exclusivamente utilizado nesta tarefa.

O armazenamento dos tweets recebidos é efetuado diretamente em ficheiros comprimidos em disco, sem quaisquer alterações ao conteúdo dos tweets. A estrutura de ficheiros utilizada para armazenar os tweets é criada dinamicamente, sendo os ficheiros organizados em diretórios cuja nomenclatura contém a referência ao ano, mês e dia em que os dados são recolhidos. A cada hora é criado um novo ficheiro, que irá conter os tweets recolhidos na hora seguinte e que será guardado no diretório correspondente ao ano, mês e dia em que os tweets estão a ser capturados.

O conjunto de tweets recebidos da *statuses/filter* API, armazenado na estrutura de ficheiros referida anteriormente, tem um crescimento contínuo, pelo que a sua manipulação e processamento tornam-se inviáveis caso permaneçam apenas armazenados desta forma. Deste modo, um processo paralelo à recolha dos tweets irá inserir os referidos tweets numa base de dados MongoDB.

No processo de armazenamento dos tweets na base de dados MongoDB, é efetuada uma filtragem dos tweets de forma a verificar se foram efetivamente escritos em Português Europeu e publicados em Portugal. Muito embora a invocação da *statuses/filter* API, tenha como filtro as delimitações geográficas de Portugal, a dificuldade de representação exata dos limites geográficos de Portugal, tal como se verificou na Figura 4.12, tem como consequência a recolha de alguns tweets provenientes de regiões geograficamente próximas de Portugal, tal como Espanha ou Marrocos. Estes tweets são filtrados pela verificação do campo *lang* que deverá conter o valor igual a ”pt” e o campo *place.country* que deverá conter o valor igual a “Portugal”.

Esta dupla verificação permite igualmente identificar os casos em que os algoritmos de deteção automática de linguagem (Twitter, 2015a) utilizados pelo Twitter, não identificam de forma correta a língua em que a mensagem do tweet foi escrita, ou seja, o texto contido no campo *text.* É frequente verificar a ocorrência de tweets cujo campo *lang* contém erradamente o valor “pt”, indicando nesse sentido que o tweet estaria escrito em Português, sem que tal se verifique na realidade. Comprovando o erro, verifica-se em certos casos que o valor do campo *place.country* encontra-se aparentemente de acordo com local onde o tweet foi produzido, tendo em conta a língua em que o tweet foi realmente escrito. A Figura 4.13 apresenta o excerto de um tweet em que o valor do campo *lang* não corresponde à língua em que o texto do tweet foi escrito, verificando-se facilmente que a mensagem do tweet se encontra escrita em Espanhol, estando de acordo com o valor do campo *place.country* onde é indicado *España*.

Como resultado do processo de filtragem, são considerados como “válidos” todos os tweets cujo

```
{
        "_id": "___",
        "text": "a mi o me aportas o te apartas.",
        "lang": "pt",
        "created_at": "Thu Feb 20 16:23:59 +0000 2014",
        "user": {
                "id": "___",
                "geo_enabled": true
         },
         "place": {
                "full_name": "Lucena del Puerto, Huelva",
                "country": "España",
                "country_code": "ES"
         }
}
```


Figura 4.13: Tweet cujo campo *lang* foi incorretamente classificado pelo Twitter.

valor do campo *lang* é igual a “pt” e o campo *place.country* é igual a ”Portugal”. Estes tweets serão armazenados na base de dados MongoDB, que como se depreende, conterá apenas tweets geolocalizados em Portugal e escritos em Português Europeu, recolhidos da *statuses/filter* API. Os tweets cujo campo *lang* é diferente de “pt” ou o campo *place.country* é diferente de ”Portugal”, não serão incluídos na base de dados de MongoDB, no entanto permanecerão guardados nos ficheiros em disco, tal como os restantes tweets, como primeira linha de backup dos dados.

O objeto JSON correspondente a cada tweet considerado como válido no contexto deste trabalho é inserido no MongoDB tal como é retornado pela Twitter API, com a adição de um um novo campo a que se convencionou designar por *created_at_object* e que conterá o valor presente no campo *created_at* num formato mais conveniente à sua futura utilização. O campo *created_at* contém a data da criação de determinado tweet no formato UTC (p.ex. “Sun Jan 04 00:00:14 +0000 2015“), no entanto, na operação de pesquisa de tweets em MongoDB onde o campo *created_at* é utilizado na filtragem do intervalo de tweets a retornar, torna-se mais simples a indicação da data caso se utilize uma máscara com o seguinte formato: ’<AAAA-MM-DD>T<HH:MM:SS>Z’. Além de que, na interação com a base de dados MongoDB foi utilizada essencialmente a linguagem de programação Python, pelo que também por este motivo a manipulação de datas no formato ’<AAAA-MM-DD>T<HH:MM:SS>Z’ se revelou ser o mais indicado. Na Figura 4.14 é ilustrada a adição do campo *created_at_object* em resultado da formatação do campo *created_at*, assumindo o campo *created_at_object* o valor de um objeto do tipo Date em resultado do retorno do construtor de datas ISODate[7], função interna do MongoDB.

Paralelamente ao armazenamento dos tweets geolocalizados numa *collection* específica, é registado numa outra *collection* MongoDB, a informação relativa aos respetivos autores dos tweets. Para cada autor é guardada a informação com a estrutura do documento JSON da Figura 4.15. O campo *_id* corresponde ao identificador único do utilizador, que é obtido do campo *user.id* de cada tweet, associando o tweet ao seu autor. Os campos *timeline_status* e *last_timeline_status_update* permitem controlar o estado do utilizador relativamente à recolha da sua timeline, cujo significado será explicado na Secção 4.4. No campo *user_date* é guardada a data em que o utilizador foi inserido na base de dados, permitindo saber quantos novos utilizadores são integrados na base de dados por cada dia.

[7] http://docs.mongodb.org/manual/core/shell-types/

```
Tweet original recebido do Twitter:
{
        "id": "___",
        "created_at": "Sun Jan 04 00:00:14 +0000 2015",
        (...)
}

Adição do campo created_at_object antes de inserir no MongoDB:
{
        "id": "___",
        "created_at": "Sun Jan 04 00:00:14 +0000 2015",
        "created_at_object": ISODate("2015-01-04T00:00:14Z"),
        (...)
}
```

Figura 4.14: Exemplo de adição do campo *created_at_object*.

```
{
        "_id": Integer,
        "timeline_status": String,
        "last_timeline_status_update": ISODate,
        "user_date": ISODate
}
```

Figura 4.15: Documento JSON que guarda a informação sobre um utilizador.

4.4 Processo de Expansão da Base de Dados

O conjunto de utilizadores identificados como tendo publicado tweets geolocalizados e cuja informação é armazenada pelo método apresentado na Secção 4.3.3, irá permitir a expansão da base de dados de tweets geolocalizados pela recolha da timeline de cada um dos utilizadores, isto é, pela recolha dos tweets produzidos pela atividade recente de cada utilizador. A timeline é disponibilizada pela REST API *statuses/user_timeline*[8], que dado um *id* de utilizador retorna os últimos 3200 tweets produzidos por esse mesmo utilizador.

O procedimento de recolha da timeline de todos os utilizadores, não é um processo trivial, dadas as restrições impostas pelo Twitter na utilização da API *statuses/user_timeline*. Tal como referido na Secção 4.2.2, o número de acessos à API do Twitter na sua versão 1.1, é contabilizado em períodos de 15 minutos (*rate limit window*), sendo permitido efetuar até 180 invocações à API em cada período por cada cliente, ou seja, por cada *access token*. Por cada invocação da API *statuses/user_timeline* é possível ler no máximo 200 tweets, logo para completar a leitura integral de uma timeline com 3200 tweets serão necessárias 16 invocações, portanto, tendo em consideração o limite de 180 pedidos à referida API no período de 15 minutos, será lida a timeline completa de 11 utilizadores em cada *rate limit window*. Este cenário pressupõe que todos os utilizadores tenham produzido pelo menos 3200 tweets.

Ainda considerando o cenário anterior, serão recolhidas somente 45 timelines por hora e cerca de 1080 timelines completas por dia, utilizando apenas um cliente para acesso à Twitter API. Além de uma

[8]https://dev.twitter.com/rest/reference/get/statuses/user_timeline

Tabela 4.2: Estados possíveis no processo de leitura integral e atualização da timeline.

Estado	Descrição
Leitura Integral	Recolher todos os tweets da timeline
Atualizar	Recolher os novos tweets da timeline
Em Atualização	"Bloqueia" o utilizador durante a atualização
Em Erro	O utilizador não permite o acesso à timeline
Bloqueado	Os novos tweets não foram publicados em Portugal
Casual	O processo de recolha terminou em erro

leitura inicial da timeline, que compreende a leitura do máximo de tweets possível, pretende-se que periodicamente sejam lidos as novas mensagens produzidas por cada autor, lendo-se apenas o que é novo desde a última leitura. Neste caso e perspetivando-se que a comunidade Portuguesa presente no Twitter fosse constituída por um número assinalável de utilizadores, a rondar os 100 mil, o que na realidade se veio a verificar, seria inviável implementar este processo recorrendo apenas a um cliente para acesso à Twitter API.

A leitura da timeline pode assumir diversos estados, apresentados na Tabela 4.2, tendo em consideração não só o momento em que o utilizador foi integrado na base de dados, como o resultado de leituras anteriores, ou seja, a primeira leitura pressupõe a integração de todos os tweets produzidos pelo utilizador, em Portugal e escritos em Português Europeu. Em iterações posteriores, são integrados apenas os novos tweets produzidos após a primeira leitura e assim sucessivamente nas leituras seguintes. Todos os utilizadores, ao serem integrados na base de dados, assumem por defeito o estado "Leitura Integral" implicando que a primeira leitura contemple os últimos 3200 tweets. Durante o processamento da timeline, o valor *timeline_status* é passado para o estado "Em atualização". Após terminar o processamento, o utilizador poderá assumir um de quatro estados possíveis: "Atualizar" significando que o processamento decorreu com normalidade e na próxima iteração, serão apenas pesquisados os tweets mais recentes; "Em Erro" o que indica que algo de anormal ocorreu durante o processamento; "Casual" é utilizado no caso dos utilizadores em que nenhum dos tweets da resptiva timeline formam publicados segundo as condições exigidas no âmbito deste trabalho. Por último, o estado "Bloqueado" indica que o utilizador não permite o acesso à sua timeline, pelo que não será possível recolher a sua timeline.

O campo *timeline_status* é comum à informação registada sobre cada utilizador e cada cliente. Relativamente à informação de cada cliente o campo *timeline_status,* remete-nos para o tipo de processamento a efetuar a determinada timeline, podendo assumir apenas dois estados: "Leitura Integral" e "Atualizar". Como se verificará na Secção 6 são incluídos na base de dados diariamente ém média de 237 novos utilizadores. Considerando que por cada *access token* é possível ler cerca de 1000 timelines completas por dia, apenas um *access token* é suficiente para ler e processar todas as timelines dos novos utilizadores integrados em cada dia. O contínuo crescimento do número de utilizadores, faz prever que uma iteração completa para atualização das respetivas timelines demore cada vez mais tempo. Ao reduzir o tempo necessário para a execução de uma iteração completa o número de tweets produzidos por cada utilizador tenderá a ser menor, reduzindo igualmente o tempo necessário à atualização de cada uma das timelines, maximizando a possibilidade de recolher toda a atividade entretanto gerada, caso contrário, para um utilizador com elevada atividade existirá grande probabilidade de este ter publicado mais de 3200 novos tweets, desde a última leitura da timeline, perdendo-se alguma da sua atividade. Com vista a minimizar o tempo entre iterações maximizando o número de tweets reco-

```
URL da API
endpoint = https://api.twitter.com/1.1/statuses/user_timeline.json?

Primeiro pedido:
url = endpoint + user_timeline.json?user_id=user_id&count=200

Pedidos seguintes:
url = endpoint + user_timeline.json?user_id=user_id&count=200&max_id=(last_max-1)
```

Figura 4.16: Parâmetros utilizados na leitura integral da timeline.

lhidos, 12 dos 14 *access tokens* para processamento das timelines, foram dedicados exclusivamente aos utilizadores que se encontram no estado ”Atualizar” e como tal o valor do campo *timeline_status* relativamente a estes clientes, contém igualmente o estado ”Atualizar”. Deste modo é possível efetuar um mapeamento entre o estado de determinado utilizador e o cliente que o poderá processar.

O processo de leitura das timelines é precedido pela seleção do *access token* disponível que não é utilizado à mais tempo e pela seleção de um utilizador que se encontre no estado correspondente ao *access token* a selecionado para a autenticação do cliente. É igualmente dada prioridade ao utilizador cuja data de processamento/atualização da timeline (*last_timeline_status_update*) seja a mais antiga, de entre o conjunto de utilizadores que se encontram em determinado estado.

Após autenticação com sucesso na Twitter API, é iniciado o processo de leitura da timeline, estando este inteiramente dependente do estado do utilizador, relativamente à recolha/atualização da respetiva timeline. Caso o utilizador se encontre no estado ”Leitura Integral”, na primeira invocação à API *statuses/user_timeline* é indicado o parâmetro *user_id* correspondente ao identificador que distingue univocamente o utilizador de cuja timeline se pretende efetuar a leitura e o parâmetro *count* com o valor de tweets que se pretende obter no resultado da invocação. Para minimizar o número de invocações no parâmetro *count* é indicado o valor 200. Nos pedidos posteriores, além dos dois parâmetros já referidos, é utilizado o parâmetro *max_id* que conterá o *id* do tweet mais elevado em resultado da invocação anterior decrementado em uma unidade, seguindo as otimizações referidas na Secção 4.2.2.1. Deste modo pretende-se evitar a recolha de tweets redundantes, considerando que durante o procedimento de leitura da timeline, poderá ocorrer a publicação de novos tweets, por parte do utilizador cuja timeline está a ser processada em determinado momento. Na Figura 4.16 são indicados os parâmetros utilizados no acesso à API *statuses/user_timeline* para leitura integral da timeline.

O procedimento de atualização da timeline dos utilizadores no estado ”Atualizar” é ligeiramente diferente da sua leitura completa, no sentido em que é tido em consideração o tweet mais recente de entre o conjunto de tweets recolhidos para cada utilizador. Deste modo, além dos parâmetros *user_id*, *count* e *max* é utilizado também o parâmetro *since_id* que indica o *id* do tweet a partir do qual deverá ser efetuada a leitura da timeline. Na Figura 4.17 são indicados os parâmetros utilizados no acesso à API *statuses/user_timeline* para atualização das timelines. Tal como no processo de leitura integral, no primeiro pedido à API *statuses/user_timeline* não é indicado o parâmetro *max_id,* sendo apenas utilizado nas invocações posteriores tendo em consideração o retorno na invocação anterior. De referir que os parâmetros *since_id* e *max_id* contêm sempre valores distintos, visto que o valor de *since_id* permanecerá estático em todos os pedidos de determinada iteração e o valor de *max_id* variará em função dos tweets que vão sendo recolhidos.

Concluída a recolha ou atualização da timeline de um utilizador, o campo

```
URL da API
endpoint = https://api.twitter.com/1.1/statuses/user_timeline.json?

Primeiro pedido:
url = endpoint + user_timeline.json?user_id=user_id&count=200&since_id=last_tweet_id

Pedidos seguintes:
url = endpoint + user_timeline.json?user_id=user_id&count=200&since_id=last_tweet_id
          &max_id=(last_max-1)
```

Figura 4.17: Parâmetros utilizados na atualização da timeline.

last_timeline_status_update é atualizado com a data correspondente ao dia em que o processamento foi realizado, de modo a que na iteração seguinte seja dada prioridade ao utilizador cujo campo *last_timeline_status_update* contenha a data mais antiga.

A transição entre os estados relativos à recolha da timeline é apresentado na Figura 4.18. A leitura com sucesso da totalidade da timeline, resulta na transição do estado ”Leitura Integral” para o estado ”Atualizar”. Durante o processo de leitura da timeline, o utilizador é colocado temporariamente no estado ”Em Atualização”, com o intuito de tentar evitar que este possa vir a ser processado por outro cliente em simultâneo. No entanto, o estado “Em Atualização” pode originar a passagem a outros estados, que não o estado ”Atualizar”, quando ocorre algo de inesperado durante a execução, sendo o utilizador colocado no estado de “Erro”. Estes erros podem dever-se a quebras na ligação à Internet, falha da própria API do Twitter ou até algum problema com o cliente que efetua a leitura e processamento da timeline. É igualmente possível que um utilizador seja colocado no estado “Bloqueado” quando este não permite o acesso à sua timeline ao público em geral. A passagem ao estado “Casual” ocorre nas situações em que um utilizador em determinado momento publicou tweets em Portugal e escritos em Português Europeu e cujas posteriores publicações ou não foram publicadas em Portugal ou se encontram escritas em outra língua. A passagem a cada um destes estados não provoca a atualização do campo *last_timeline_status_update*.

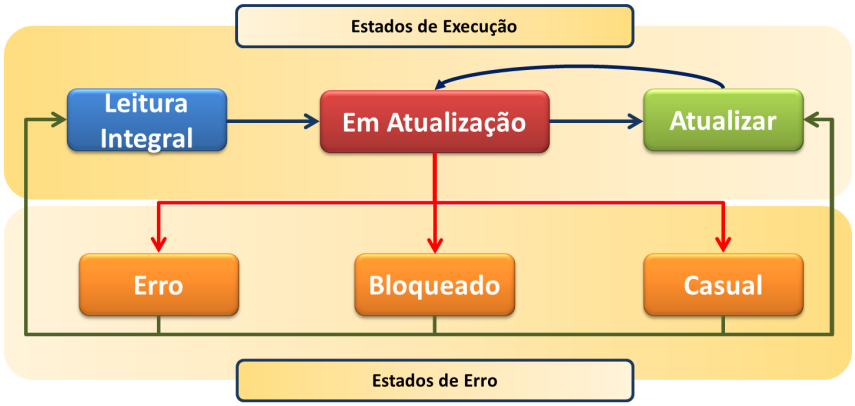


Figura 4.18: Diagrama de transição de estados para a recolha da timeline de cada utilizador.

Um outro *access token* é utilizado na tentativa de ”recuperar” os utilizadores que se encontram num dos estados de erro. Por vontade própria os utilizadores cuja timeline esteja propositadamente bloqueda podem optar por torná-la pública e como tal desta forma será detetada essa mesma alteração. Será efetuada a leitura da respetiva timeline e alterado o estado do utilizador para o estado ”Atualizar”. A não

atualização do campo *last_timeline_status_update* na passagem a um estado de erro tem influência no processo de ”recuperação”, ou seja, se o campo *last_timeline_status_update* permanecer igual ao campo *date* significa que a passagem ao estado de erro ocorreu quando o utilizador se encontrava no estado ”Leitura Integral”. Este cenário ocorre quando ainda não foi lido qualquer tweet da timeline, o que conduzirá a que o processo de ”recuperação” seja testado pela recolocação do utilizador no estado ”Leitura Integral” testando-se a leitura completa da timeline.

Existe grande probabilidade de entre iterações consecutivas da leitura das timelines sejam produzidos novos tweets geolocalizados, no entanto estes são armazenados numa *collection* distinta de onde são armazenados os tweets das timelines, pelo que a atualização das timelines é efetuada tendo sempre por base o último tweet recolhido pelo procedimento de leitura da timeline, garantindo a captura de todos os novos tweets sejam eles geolocalizados ou não. À imagem do procedimento de recolha dos tweets geolocalizados, os tweets lidos das timelines são armazenados em ficheiros comprimidos em disco. Neste caso, os ficheiros são guardados numa estrutura de ficheiros organizada de forma um pouco diferente, onde o nome de cada diretório é definido pela sequência de caracteres 000, 001, 002 . . . 999. Por cada invocação da API *statuses/user_timeline* é gerado um novo ficheiro cujo nome é composto pelo *id* do utilizador (indicado no campo *user.id*) que está a ser processado além de outras informações, sendo este guardado no diretório correspondente aos três últimos dígitos do *id* de utilizador. Esta nomenclatura dos diretórios permite uma distribuição uniforme dos documentos pela estrutura de ficheiros, visto que a elevada quantidade de invocações à API gera igual número de ficheiros, o que faz atingir rapidamente o valor máximo de ficheiros que é possível guardar num único diretório, caso sejam guardados num só diretório. O cuidado a ter na gestão do número de ficheiros por diretoria está também relacionado com o facto da máquina utilizada neste trabalho conter o sistema de gestão de ficheiros AFS (*Andrew File System*). Neste sistema é possível que uma diretoria contenha até 64 mil ficheiros quando o nome dos ficheiros contém menos de 16 caracteres. Caso o nome dos ficheiros contenha entre 16 e 32 caracteres o número de ficheiros permitidos por diretório será bastante mais reduzido. Existem 64 mil *slots* por diretório, logo cada ficheiro cujo nome tem menos de 16 caracteres ocupa apenas 1 *slot*, mas ficheiros cujo nome têm entre 16 e 32 caracteres necessitam de 2 *slots* (Arpaci-Dusseau and Arpaci-Dusseau, 2012). A nomenclatura usada nos ficheiros criados em resultado da resposta da API *statuses/user_timeline*, é superior a 16 caracteres e inclui além do *id* do utilizador, um valor que varia de 1 a 16 e que indica o número da invocação (permite tornar único, o nome do ficheiro dentro do diretório) e a data do dia de processamento, como no seguinte exemplo: ”1536514902_1_2015-03-10.gz”.

A Figura 4.19 resume o fluxo de processamento do algoritmo apresentado nesta Secção. O processo inicia com a criação de um cliente que efetua a leitura a partir da base de dados MongoDB das credenciais que o permitirá conectar-se à Twitter API. Cada cliente pode desempenhar dois tipos de tarefas: i) leitura completa de uma timeline ou ii) atualização de uma timeline. Enquanto existirem utilizadores no estado correspondente ao tipo de tarefa suportada pelo cliente, este mantém-se em atividade, selecionando consecutivamente o próximo utilizador a processar. Sabendo o estado em que se encontra o utilizador seleccionado é despoletado o respetivo processamento, onde a parte esquerda da Figura 4.19 corresponde à sequência de ações necessárias à leitura integral de timeline de um utilizador e a parte esquerda, por conseguinte, corresponde ao processo de atualização de uma timeline.

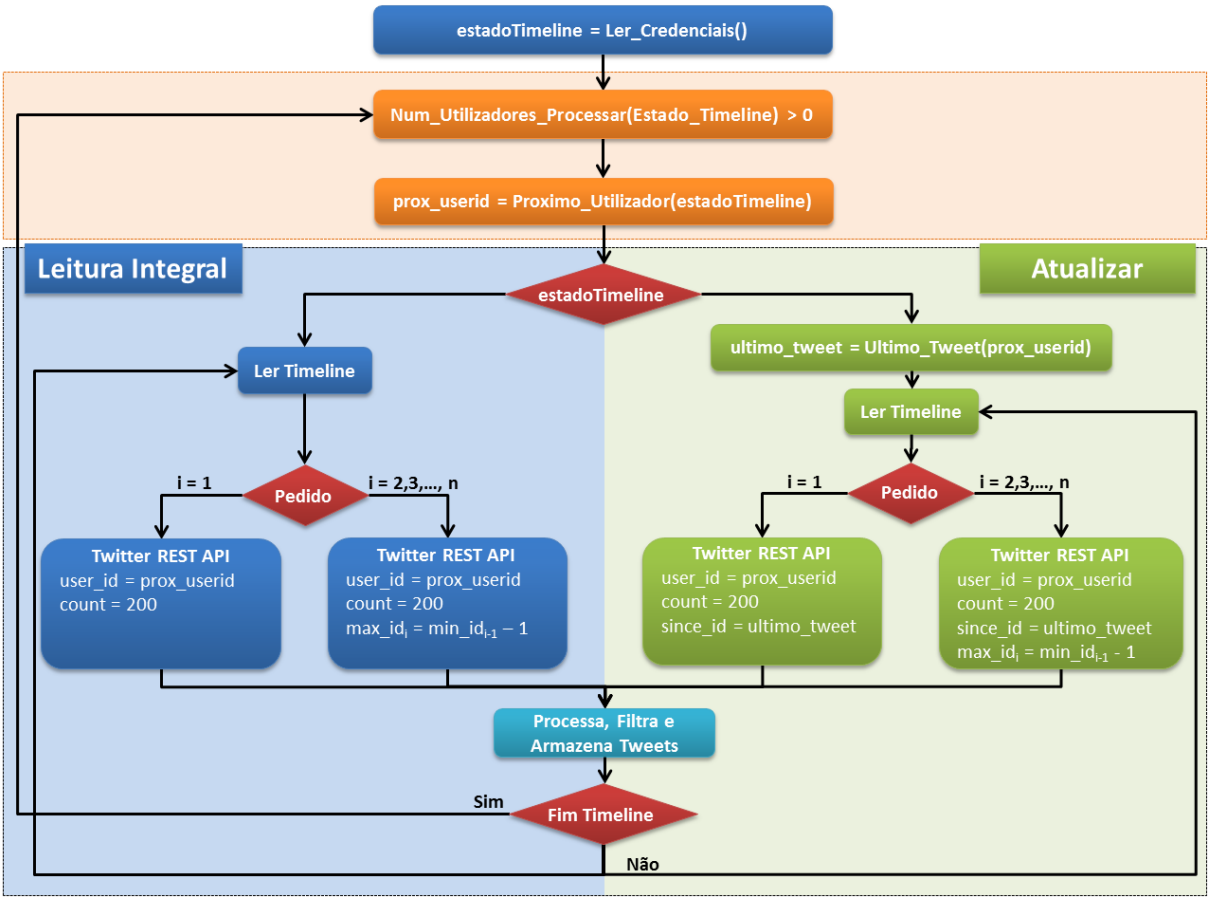


Figura 4.19: Algoritmo de expansão da base de dados de tweets.

4.5 Optimizações e Correção de Erros

No processamento dos ficheiros resultantes de cada invocação à REST API *statuses/user_timeline*, verificou-se que uma elevada percentagem de ficheiros continha algumas particularidades que impossibilitavam a correta leitura dos tweets. Deste modo, foram implementadas duas verificações iniciais que resolveram grande parte dos erros. Na maioria dos casos, a escrita do último tweet no ficheiro não é completamente terminada, causando erro aquando da tentativa de leitura e construção do objeto JSON referente ao último tweet. A correção deste tipo de erro demonstrou-se ser trivial, dado que o conjunto de tweets é escrito de forma sucessiva numa única linha do ficheiro, representando uma lista de tweets, delimitada por parênteses retos $[tweet_1, tweet_2, ..., tweet_n]$, pelo que a referida linha deverá terminar com o caracter ']' seguido do caracter de nova linha. Tendo presente que todo o tweet retornado pela Twitter API é iniciado por "$\{created\_at : ...$", nos casos em que o penúltimo caracter da linha que contém a lista de tweets não corresponda a ']', indica que a lista não foi corretamente terminada e, por conseguinte, será desconsiderado o último tweet. É pesquisa a última ocorrência da sequência de caracteres "$,\{created\_at : ...$", sendo esta substituídos por ']', eliminando-se deste modo o último tweet, possibilitando o processamento do ficheiro com sucesso. Outro problema identificado, está relacionado com as respostas da REST API *statuses/user_timeline* onde é retornado apenas um tweet. Nestes casos, a linha do ficheiro que contém a lista de tweets termina com os caracteres ']]', precedido do caracter de nova linha. A solução para este sito de ocorrências baseou-se na eliminação o último caracter

']', resultando na definição correta de uma lista de tweets, embora contendo apenas um tweet. Estas melhorias no processo de leitura das timelines, foram sendo implementadas ao longo do desenvolvimento do trabalho, pelo que se assume que numa fase inicial se tenham desperdiçado alguns tweets, visto que quando a criação da representação JSON da lista de tweets originava erro, todos os tweets eram descartados. Além destas correções, continuaram a persistir outro tipos de erros, nomeadamente a ocorrência de ficheiros comprimidos que se encontravam corrompidos, cuja descompressão não era possível. A baixa frequência deste tipo de erros, permite considerar residual o volume de tweets não incluídos na base de dados.

A utilização de 14 clientes em simultâneo para leitura e atualização das timelines tal como indicado na Tabela 4.1, origina com relativa frequência alguns problemas de concorrência ao nível da seleção do próximo utilizador cuja timeline deverá ser processada, podendo verificar-se o cenário de dois clientes distintos processarem a timeline de um mesmo utilizador em simultâneo. Tal facto não produz efeitos ao nível da duplicação de informação na coleção de tweets lidos das timelines, uma vez que na chave (*_id*) de cada documento JSON correspondente a determinado tweet é utilizado o campo *id* do respetivo tweet e que por ser único não permite a duplicação de tweets. No entanto, este problema de concorrência provoca o desperdício de invocações à Twitter API, dada a repetição de invocações sobre um mesmo utilizador. Para mitigar tal situação, o campo *timeline_status* associado a cada utilizador foi usado inicialmente como uma *flag*, ou seja, após a seleção do próximo utilizador a processar, o campo *timeline_status* é atualizado para o estado "Em Atualização", indicando que a respetiva timeline está a ser processada de modo a tentar bloquear o seu processamento por parte de outro cliente em simultâneo. Esta primeira abordagem permitiu reduzir de forma significativa o número de pedidos repetidos sem, no entanto, resolver por completo a questão. Tal facto deve-se essencialmente à não existência do conceito de transação no MongoDB, pelo que entre a pesquisa do próximo utilizador e a respetiva atualização de estado, é possível que um ou mais clientes possam iniciar a mesma pesquisa, e por conseguinte, pode ser retornado o mesmo *id* de utilizador para mais do que um cliente até que seja terminada a operação de atualização de estado do campo *timeline_status,* iniciada por algum dos clientes a executar em concorrência. A solução baseou-se utilização da função *findAndModify()*[9]*,* pertencente à biblioteca de funções do MongoDB, que permite a pesquisa de documentos seguida da atualização de um ou mais documentos pertencentes ao conjunto retornado pela pesquisa. Deste modo, a atualização do campo *timeline_status* do utilizador selecionado é concluída na sequência da operação de pesquisa, não permitindo que dois ou mais clientes iniciem o processo de seleção do próximo utilizador em simultâneo, otimizando o número de pedidos à Twitter API.

Na Secção 4.2.2 foi referido que o *endpoint* da Twitter API *statuses/user_timeline* permite apenas 180 invocações em cada intervalo de 15 minutos. Quando tal limitação é ultrapassada é retornada a mensagem de erro "Rate limit exceeded" correspondente ao código de erro HTTP 429 e também nestes casos é gerado um ficheiro com o conteúdo da resposta da API. Embora se tenha previsto um automatismo para eliminação dos ficheiros com respostas de erro, o tráfego gerado por tais invocações é desnecessário. Por isso, entre cada invocação à API *statuses/user_timeline* é efetuada uma pausa de 5 segundos, evitando os erros por excesso de invocações.

[9] *http://docs.mongodb.org/manual/reference/method/db.collection.findAndModify/*

4.6 Sumário

A Figura 4.20 resume a arquitetura de recolha de tweets apresentada neste Capítulo. Do lado esquerdo é sintetizado o processo de captura de tweets geolocalizados e do lado direito o método de expansão do *corpus* pela leitura do histórico de mensagens dos autores dos tweets geolocalizados. Este procedimento permite a constituição de um *corpus* de tweets publicados por utilizadores portugueses, escritos em português e produzidos em Portugal. Ao longo do desenvolvimento de arquitetura proposta, foram sendo consideradas diversas questões de performance tanto no armazenamento dos dados como na forma de os obter e processar.

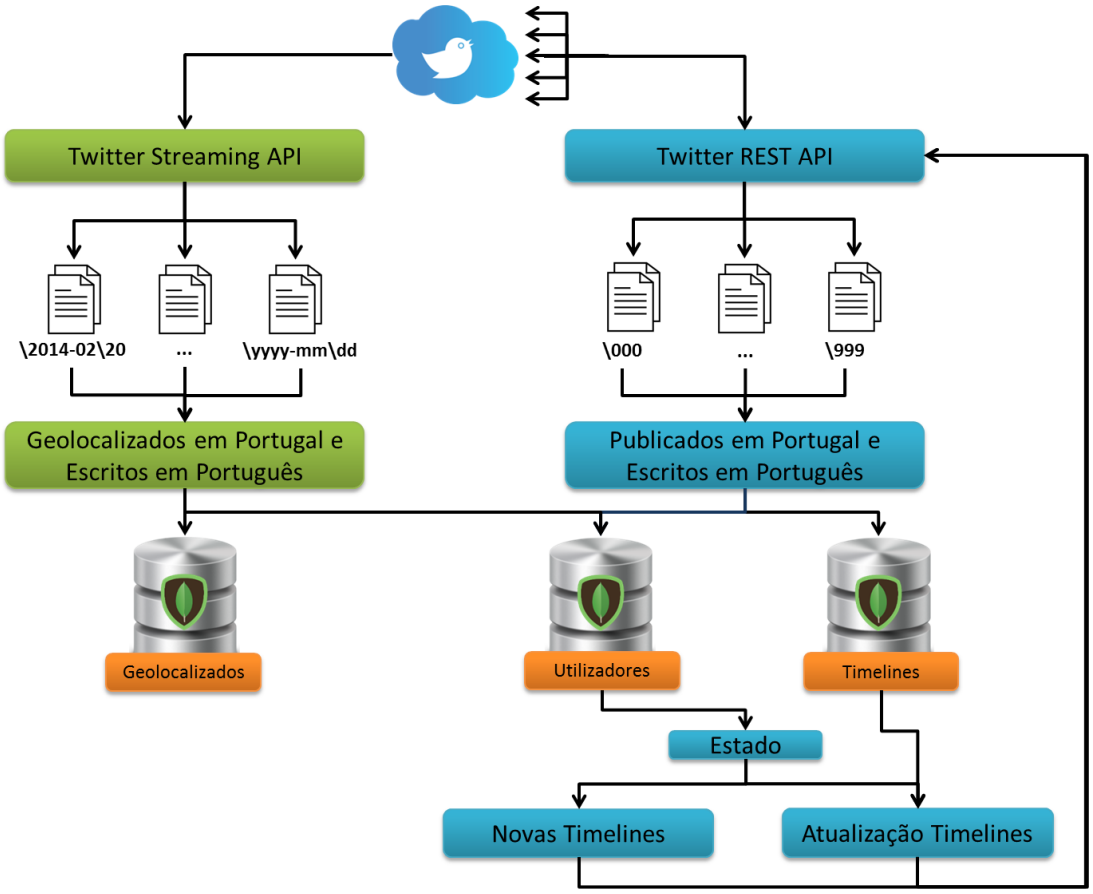


Figura 4.20: Arquitetura para recolha e expansão de um *corpus* de tweets portugueses.

5 Acesso e Visualização de Dados

> Graphics compress data and provide it in a visual form that is easily recollected by humans. For businesses and people to process more data, data compression and data visualization are needed.
>
> *Russell Walker*

Neste Capítulo são apresentados os módulos *Acesso* e *Visualização* do Sistema de Informação. A Secção 5.1 aborda os principios básicos a ter em consideração no desenho e implementação de uma REST API. Também nesta Secção são descritas as opções tomadas na implementação da REST API que permite o acesso à base de dados de tweets a aplicações de terceiros e que constitui o móludo *Acesso* do Sistema de Informação. Na Secção 5.2 são enunciados os cuidados a ter no desenho e na apresentação de um Dashboard. As Secções 5.3 e 5.4 referem as opções tomadas no desenvolvimento do Dashboard que está na génese do módulo *Visualização* do Sistema de Informação.

5.1 Implementação de uma REST API para Acesso ao *Corpus*

Esta secção começa por abordar os aspetos principais de desenho de REST APIs que devem ser tidos em conta na sua implementação. De seguida descreve-se a implementação de uma REST API que irá permitir o acesso aos dados armazenados. Finalmente, apresenta-se um exemplo de integração desta base de dados numa aplicação concreta, com base na API implementada.

5.1.1 Aspetos de Desenho e Implementação de REST APIs

Nos últimos anos a tecnologia REST tem emergido como uma arquitetura padrão para o projeto de web services e web APIs. A criação de serviços REST é bastante simples, no entanto, deverá ter-se em atenção algumas regras básicas de modo a seguir o padrão instituido. As características de uma REST API são definidas pelas seguintes regras de desenho de APIs:

Cliente-Servidor: Deve haver uma separação entre o servidor que oferece serviços e o cliente que consome os serviços;

***Stateless*:** Cada solicitação de um cliente deve conter todas as informações exigidas pelo servidor para realizar o pedido. Por outras palavras, o servidor não pode armazenar informações fornecidas pelo cliente em determinado pedido e usá-lo num outro pedido, ou seja, não será guardado o estado da aplicação num dado instante;

Tabela 5.1: Métodos de invocação de uma REST API via HTTP.

Método HTTP	Ação	Exemplo
GET	Obter informação sobre um recurso	http://hostname/api/exemplo (retorna uma lista de recursos)
GET	Obter informação sobre um recurso	http://hostname/api/exemplo/1 (retorna o recurso #1)
POST	Criar um novo recurso	http://hostname/api/exemplo (cria um novo recurso, mediante os dados da invocação)
PUT	Atualizar um recurso	http://hostname/api/exemplo/1 (atualiza o recurso #1, com os novos dados submetidos na invocação do serviço)
DELETE	Apagar recurso	http://hostname/api/exemplo/1 (apaga o recurso #1)

***Cacheable*:** O servidor deve indicar ao cliente se os pedidos podem ser armazenados em *cache* ou não;

Sistema por camadas: A comunicação entre um cliente e o servidor deve ser padronizada, possibilitando que sistemas intermédios possam responder aos pedidos, em vez do servidor final, sem que o cliente tenha de alterar algo;

Interface uniforme: O método de comunicação entre um cliente e um servidor deve ser uniforme;

Código a pedido: Os servidores podem fornecer o código executável ou os *scripts* que os clientes devem executar. Esta restrição é a única opcional.

A arquitetura REST foi originalmente projetada de acordo com o protocolo HTTP (protocolo base para a comunicação de dados da World Wide Web). Um conceito fundamental na definição de serviços REST é a noção de recurso. Cada recurso é representado por uma URI (*Uniform Resource Identifier*). Os clientes enviam pedidos aos URIs usando os métodos definidos pelo protocolo HTTP e recebem, possivelmente, como resultado o estado das alterações dos recursos afetados. Os métodos de invocação de recursos via HTTP são normalmente concebidos para aceder a um determinado recurso, seguindo uma forma padrão apresentada na Tabela 5.1.

O desenho de REST APIs não requer um formato específico para os dados enviados na invocação de determinado recurso. Geralmente é fornecido no corpo da solicitação uma estrutura JSON ou por vezes como parâmetro incluído no URL da consulta. O projeto de um web service ou de uma API que siga as diretrizes REST, resume-se a um exercício de identificação de quais os recursos podem ser expostos e de como estes serão afetados pelos diferentes pedidos HTTP.

Existem diversas *frameworks* através das quais é possível efetuar a implementação de REST APIs. Foram analisadas diversas soluções para implementação de REST APIs das quais se destacam: Django REST Framework[1] baseada na linguagem Python; Express[2] uma biblioteca de NodeJS[3] ou Slim[4] são opções de implementação em JavaScript e PHP respetivamente. A *framework* Flask[5]

[1] http://www.django-rest-framework.org
[2] http://expressjs.com/
[3] https://nodejs.org
[4] http://www.slimframework.com
[5] http://flask.pocoo.org

revelou-se ser a simples de utilizar permitindo um rápido desenvolvimento de uma REST API e por também ser implementada em Python, foi a solução escolhida.

Flask tem uma estrutura relativamente pequena quando comparada com a maioria das restantes *frameworks*, podendo inclusivamente ser chamada de "*microframework*" facilitando a familiarização com sua arquitetura, mas a sua simplicidade não significa menor funcionalidade ou desempenho que outras *frameworks*. O seu desenvolvimento foi concebido de modo a ser extensível, focando-se em proporcionar um núcleo sólido com os serviços básicos, enquanto as extensões desenvolvidas posteriormente complementas as suas funcionalidades (Grinberg, 2014). Flask tem duas dependências essenciais: os sistemas de *routing* (encaminhamento), *debugging*, e Web Server Gateway Interface[6] (WSGI) são disponibilizados pela extensão Werkzeug, enquanto o modelo de criação e formatação de templates HTML é fornecido pela extensão Jinja2. As extensões Werkzeug e Jinja2 são essenciais ao correto funcionamento do Flask e foram desenvolvidas pelo pelo mesmo autor, no entanto, não é garantido o suporte nativo no acesso a bases de dados, validação de formulários Web, autenticação ou outras tarefas de alto nível. Diversos serviços e funcionalidades essenciais à maioria das aplicações web podem ser adicionadas, selecionando quais as extensões que melhor se adaptam ao projeto em causa ou no limite poderão ser desenvolvidas extensões em conformidade com as necessidades. Tal flexibilidade contrasta com as *frameworks* de maior estrutura, nas quais por vezes não existe a flexibilidade de optar por quais extensões utilizar.

5.1.2 Implementação da REST API

O *corpus* de tweets resultante do método apresentado neste artigo, caso não seja devidamente analisado e tratado, a informação que dele se poderá retirar não constituirá uma mais-valia para as diversas áreas em que poderá ser útil. Por vezes, este tipo de análises são efetuadas por especialistas em áreas cujo domínio de alguns aspetos da informática não é muito acentuado. Neste sentido e de modo a criar uma forma de abstração, nomeadamente no acesso aos tweets armazenados na base de dados MongoDB, foi implementada uma REST API que disponibiliza um conjunto de serviços (*endpoints*) através dos quais é possível o acesso à informação

Os serviços implementados podem ser vistos como uma interface de acesso à base de dados, facilitando o desenvolvimento de aplicações externas baseadas na informação armazenada, abstraindo a estrutura dos dados e o acesso aos mesmos. Na implementação da REST API foi utilizada a microframework Flask que por ser desenvolvida em Python, proporciona facilmente a integração com os restantes módulos do Sistema de Informação. Todos serviços disponibilizados, permitem somente a leitura de dados, de entre os quais se destacam: acesso à coleção de tweets geolocalizados, à timeline de um utilizador em específico, a estatísticas sobre os dados ou ao perfil de um determinado utilizador.

A possibilidade de inserção de novos tweets no MongoDB via REST API não foi considerada visto não contribuir para um melhor desempenho global do sistema. Para manter a consistência dos dados, não foram implementados quaisquer *endpoints* que efetuem alterações aos mesmos. No entanto, poderá ser considerado o desenvolvimento de *endpoints* que possibilitem a atualização ou a introdução de novos campos na estrutura JSON associada a cada tweet, com o intuito de enriquecer a base de dados com informação proveniente do exterior, tal como a idade ou o género de determinado utilizador.

[6] Especificação de uma interface entre servidores e aplicações/frameworks web desenvolvidas na linguagem de programação Python.

Tabela 5.2: Serviços da REST API para acesso aos tweets geolocalizados.

Nome Serviço	**Parâmetros**	**Dados retornados pelo serviço**
/api/<collection>/tweet/	<int: tweet_id>	Tweet correspondente ao ID indicando em “tweet_id”
/api/<collection>/page/	<int: page_id>	Conjunto de 1000 tweets, ordenados decrescentemente pela ordem de publicação
/api/<collection>/day/	<int: day>	Tweets produzidos no dia indicado em ”day”
/api/<collection>/user/	<int: screen_name>	Todos os tweets produzidos pelo utilizador indicado em ”screen_name”
/api/<collection>/query/	<String: query>	Conjunto de tweets de acordo com o filtro especificado pelo parâmetro ”query”

Na Tabela 5.2 são apresentados os principais *endpoints* para acesso aos tweets. De notar que para selecionar qual a coleção de tweets em que será efetuada a pesquisa, em cada *endpoint* deverá ser subtituído o valor de <collection> por ”geolocated” ou ”timeline”, indicando que se pretende pesquisar na collection de tweets geolocalizados ou na collection de tweets das timelines, respetivamente. Na Figura 5.1 são apresentados dois exemplos de utilização dos *endpoints* indicados na Tabela 5.2. No primeiro exemplo apresentado na Figura 5.1, é invocado o *endpoint* /api/geolocated/tweet/ onde é pesquisada a informação relativa ao tweet que cujo ID é o valor 551528493021147136. O segundo exemplo invoca o *endpoint* /api/timeline/query/ pretendendo-se obter a timeline do utilizador com o *screen_name* ”Razzo”. A query que permitirá retornar tal resultado é incluída no ficheiro query.json, passado como parâmetro no comando *curl,* cujo conteúdo é {"user.screen_name":"Razzo"} de modo a permitir filtrar de entre todos os tweets incluídos na collection das timelines dos utilizadores, somente os tweets produzidos por @Razzo.

A informação relativa a cada um dos utilizadores que utilizam a rede social do Twitter, poderá revelar-se bastante útil para estudos de análise de perfis, com o objetivo de caracterizar a comunidade em termos de idade, género, podendo estimar o seu nível económico que de certo modo poderá estar interligado com o tipo dispositivo utilizado na publicação dos tweets (Android ou iOS). Estudos relativos à migração, considerando os diferentes locais onde os demais utilizadores vão publicando os seus tweets, poderá dar uma ideia sobre a movimentação de pessoas numa determinada região geográfica. Para tal, foram criados *endpoints* específicos para possibilitar o acesso à informação do perfil de cada utilizador, descritos na Tabela 5.3.

Em particular, o último serviço identificado na Tabela 5.3 com o nome user/ageAndGender foi definido para retornar apenas alguma informação específica de cada utilizador, que se julga permitir inferir a sua idade e género, nomeadamente os campos: *user_description, user_profile_image_url, user_statuses_count, user_friends_count, user_favourites_count, user_created_at, user_lang* e *user_location.* A informação retornada, tem em consideração o último tweet da base de dados relativo ao utilizador em indicado no parâmetro do serviço. Este serviço da REST API foi utilizado no treino e teste do algoritmo desenvolvido por Vicente et al. (2015), que pretende detetar de forma automática a idade e o género de cada utilizador.

Devido ao elevado volume de dados armazenados, determinadas pesquisas podem tornar-se bas-

```
> curl -i http://localhost:27016/api/geolocated/tweet/551528493021147136

200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 102
Server: Werkzeug/0.9.6 Python/2.7.8
Date: Sun, 26 Jul 2015 17:00:36 GMT
{
  "_id" : "551528493021147136",
  "text" : "Boa noite!",
  "created_at_object" : "2015-01-04 00:00:14",
  ...
}

> curl -i -H "Content-Type: application/json" -X POST -d @query.json
http://localhost:27016/api/timeline/query/

200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 2481
Server: Werkzeug/0.9.6 Python/2.7.8
Date: Sun, 26 Jul 2015 17:14:36 GMT
{
  "_id": "436945853522399232",
  "text": "Fora do S. Martinho continua a existir Golegã. @Golegã
                  http://t.co/mkpveaR8SA",
  "created_at_object": "2014-02-21 19:29:44",
  "user": {
                "screen_name": Razzo,
                ...
  }
  ...
}
{
   "_id": "436946283757309953",
   "text": "@0xJoao vim cá passear um fim de semana com amigos.",
   "created_at_object": "2014-02-21 19:31:26",
   "user": {
                "screen_name": "Razzo",
                ...
   }
  ...
}
```

Figura 5.1: Exemplos de utilização da REST API.

Tabela 5.3: Serviços da REST API para acesso à informação de cada utilizador.

Nome Serviço	Parâmetros	Dados retornados pelo serviço
/api/user/state/	<int:user_id>	Estado em que o utilizador se encontra, no contexto deste trabalho. Ou seja, se a sua timeline já foi recolhida, se está em atualização ou se se encontra bloqueada pelo próprio utilizador
/api/user/first_profile/	<String:screen_name>	Perfil do utilizador *@screen_name* relativamente ao seu primeiro tweet incluído na base de dados
/api/user/last_profile/	<String:screen_name>	Perfil do utilizador *@screen_name* relativamente ao seu último tweet incluído na base de dados
/api/user/ageAndGender/	<String:screen_name>	Campos do perfil do utilizador que possam permitir inferir a idade e o género do utilizador *@screen_name*

tante demoradas. Para tal, o resultado de pesquisas mais complexas, é guardado em novas collections MongoDB, permitindo a visualização de determinadas estatísticas sobre os dados de forma consideravelmente mais rápida. Algumas das pesquisas pré-processadas são disponibilizadas pelos serviços da REST API apresentados na Tabela 5.4.

Tabela 5.4: Serviços da REST API para acesso a estatísticas pré-processadas.

Nome Serviço	Dados retornados pelo serviço
/stats_geolocated_day/	Total por dia de tweets geolocalizados recolhidos (desde 20 de Fevereiro de 2014)
/stats_geolocated_hour_day/	Soma dos tweets geolocalizados recolhidos por cada hora
/stats_geolocated_week_day/	Soma dos tweets geolocalizados recolhidos por dia da semana
/stats_timeline_day/	Soma dos tweets da timeline em cada dia (desde 20 de Fevereiro de 2014)
/stats_users_day/	Número de novos utilizadores identificados por dia

Na Figura 5.2 é demonstrada a abstração introduzida pela REST API no acesso à informação armazenada na base de dados MongoDB. A REST API permite a comunicação bidirecional com web browsers (Internet Explorer, Chrome, Firefox), com aplicações da linha de comandos (curl, http) ou com aplicações móveis (Android, iOS).

As invocações à REST API são efetuadas através de pedidos HTTP aos serviços disponibilizados, que consultam a informação solicitada na base de dados MongoDB. Na resposta às invocações são utilizadas as bibliotecas Werkzeug e Jinja2.

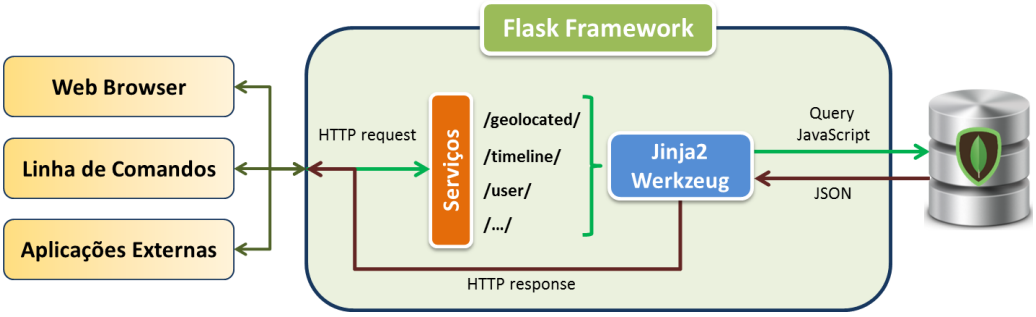


Figura 5.2: Interação entre aplicações externas e o MongoDB via REST API.

5.1.3 Exemplo de Integração da Base de Dados via REST API

Além dos *endpoints* apresentados anteriormente, foi implementado um serviço específico para possibilitar a interligação com uma aplicação externa que requer a interação com os tweets armazenados de uma forma específica. Tal como referido do Capítulo 1, no âmbito do projeto MISNIS pretende-se efetuar a deteção de tópicos e analisar a sua influência nas redes sociais Carvalho et al. (2013). Com recurso a tweets Portugueses, é possível criar a impressão digital de um dado tópico/hashtag do Twitter e detetar outros tweets sobre um mesmo tema. Dada uma hashtag, grande parte dos tweets não se encontram classificados como sendo relacionados com determinado tópico, pelo que a utilização do método Twitter Topic Fuzzy Fingerprints, desenvolvido por Rosa et al. (2014), permite expandir o universo de tweets correlacionados com o tema. A impressão digital de um assunto, é então um conjunto (*rank)* de palavras ordenadas por ordem de importância, que possam de alguma forma estar relacionadas com o tema do tópico. Por exemplo, a hastag '#Socrates', poderia ter como impressão digital as palavras {"prisao", "socas", "evora", "ministro"} entre outras. Estas palavras são posteriormente usadas no cálculo da semelhança entre tweets. Este método procura igualmente definir quem são os utilizadores mais importantes na propagação e criação de um tópico. Usando algoritmos de análise de rede como o sobejamente conhecido PageRank da Google (Page et al., 1998), estabelecem-se ligações entre os utilizadores com base nas menções efetuadas pelos utilizadores que publicam mensagens relacionadas com o tema. Por exemplo, o utilizador @user_a poderia escrever "não detestas política, @user_b?" o que criaria um elo de ligação do @user_a para o @user_b.

Dada a limitação do Twitter em permitir apenas o acesso a cerca de 1% dos tweets publicados num dado momento, a utilização do *corpus* resultante desta tese, foi imprescindível para melhorar a qualidade dos resultados obtidos por Rosa et al. (2014). Na definição deste *endpoint,* que se convencionou designar por */api/fingerprint/,* foi essencialmente considerada a sua utilização via browser, sendo solicitado através de um formulário, a introdução de alguns parâmetros necessários para a execução do algoritmo Twitter Topic Fuzzy Fingerprints.

A Figura 5.3 demonstra a utilização do *endpoint /api/fingerprint/* pelo algoritmo Twitter Topic Fuzzy Fingerprints. Neste exemplo, pretende-se detetar a ocorrência de tweets relacionados com a detenção do ex-Primeiro Ministro de Portugal, Eng. José Sócrates. Na imagem 1 da Figura 5.3 é solicitado o intervalo de datas aproximado à ocorrência do evento assim como a indicação de possíveis hashtags que possam estar relacionadas com o evento. Na imagem 2 é apresentado um resultado intermédio do algoritmo Twitter Topic Fuzzy Fingerprints com a sugestão de hastags que foram detetadas como estando relacionadas com o evento em causa e sobre este conjunto de hashtags é dada a possibili-

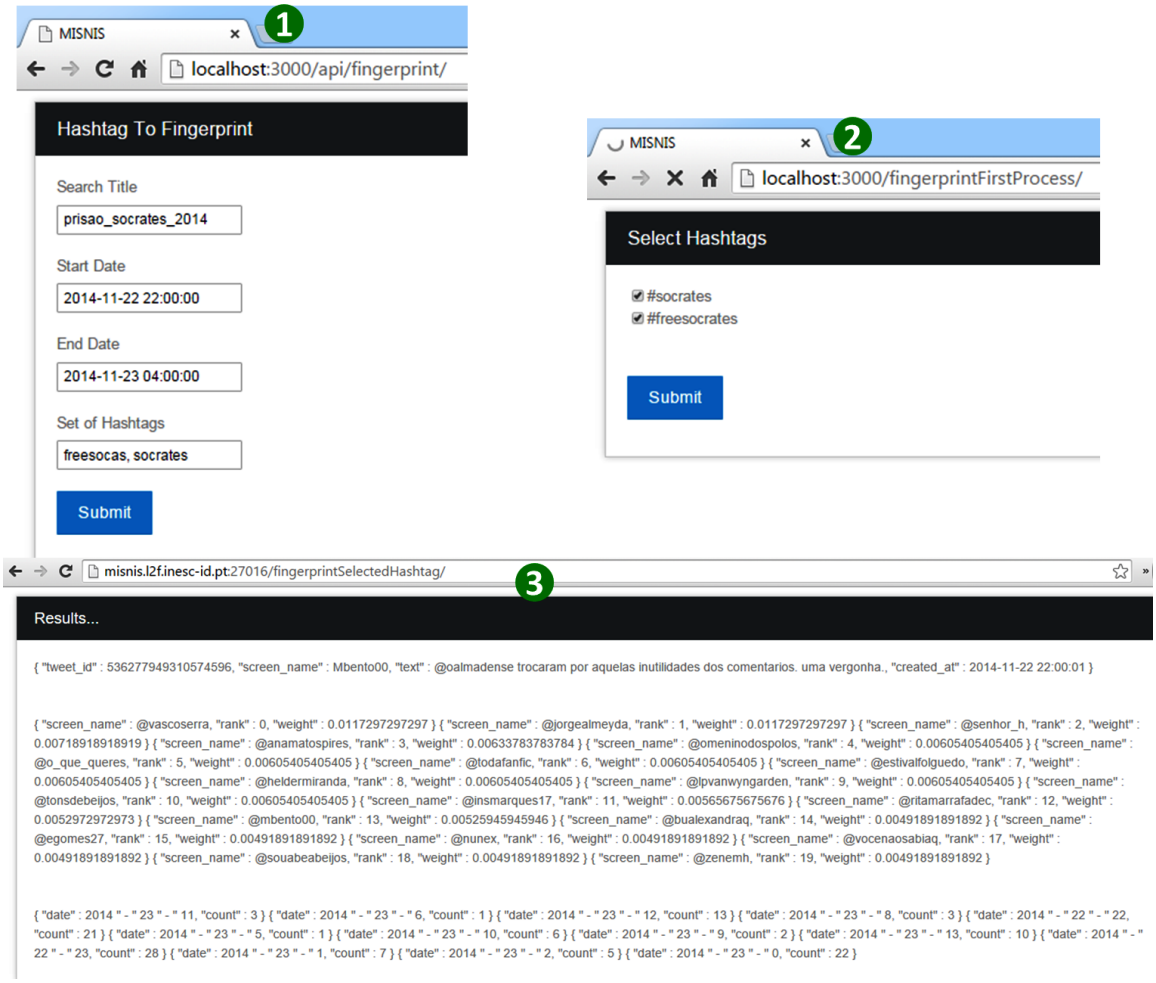


Figura 5.3: Exemplo de utilização do *endpoint /api/fingerprint/* da REST API.

dade de se escolherem as que se julgam estar efetivamente relacionadas com o tópico, dando início à segunda fase de processamento do algoritmo. Na imagem 3 são mostrados alguns dos resultados finais do algoritmo, com a indicação do primeiro tweet sobre o tema assim como a quantidade de tweets produzidos em cada hora do período indicado ao início e que estão previsivelmente correlacionados com o evento. Um dos resultados mais interessantes é o retorno do conjunto de utilizadores que o algoritmo considera estarem relacionados com o evento. Com este exemplo, demonstra-se a facilidade de integração do Sistema de Informação desenvolvido com o trabalho de outros autores, nomeadamente através da REST API.

5.2 Dashboards Web para Visualização de Informação

O Dashboard não é mais do que um painel de gestão essencialmente suportado em gráficos, capazes de comunicar instantaneamente e de forma eficaz, a informação pertinente sobre uma determinada realidade, apresentando a informação mais relevante, de forma apelativa e inteligente, através de um design elegante e inovador (Caldeira, 2010). O mesmo autor enuncia alguns dos princípios básicos a ter em consideração no desenho e construção de um dashboard:

- **Visualização numa única página** - evita que o utilizador a passar de página para página, visto que toda a informação deverá estar visível imediatamente. Poderá ter-se mais do que uma página, mas visualizadas através de separadores que agregam diferentes tipos de informação;
- **Apresentar a informação através de gráficos** - os gráficos são muito poderosos na comunicação da informação sobre determinada realidade, transmitindo métricas e indicadores de forma clara, objetiva e rápida;
- **Apresentar apenas a informação realmente necessária** - o dashboard deve ter apenas a informação necessária para que possa cumprir o seu objetivo;
- **Destacar a informação relevante** - é fundamental a apresentação da informação de forma a facilitar a perceção da mesma;
- **Realçar o presente em detrimento do passado** - deve-se focar, em primeiro lugar, aquilo que está a acontecer no ano presente, deixando para segundo plano o que aconteceu no passado, que é sempre menos importante do que o que se passa agora. No entanto, a análise conjunta do presente e do passado permite percecionar a dimensão da evolução;
- **Organizar a informação de forma lógica** - numa mesma página a informação deve estar arrumada em grupos com relacionamento de assuntos;
- **Utilizar o espaço de forma eficiente** - sabendo a que o espaço se restringe à área equivalente de uma página e existindo uma grande quantidade de informação é imprescindível que se consiga rentabilizar o espaço disponível da forma mais eficiente;
- **Apresentar um visual apelativo** - o dashboard deve ser eficaz em termos de comunicação da informação, com uma apresentação limpa, elegante, equilibrada e esteticamente agradável.

Existem diversas soluções de software proprietário especializado na construção de Dashboards embora com um custo assinalável, onde o QlikView[7] é um dos exemplos. No entanto, o desenho e implementação de um dashboard integrado no Sistema de Informação resultante desta dissertação, deverá ter por base tecnologias Open Source. Como tal, foram testadas e analisadas algumas bibliotecas de geração de gráficos dinâmicos, que possam ser disponibilizados na Web. Uma das primeiras opções analisadas foi o D3[8], uma biblioteca JavaScript para a manipulação de documentos baseados em dados, auxiliando na visualização dos dados em HTML. Outra alternativa considerada foi a biblioteca de gráficos HighCharts[9], também desenvolvida em JavaScript e baseada na tecnologia nativa dos browsers nomeadamente o HTLM 5, é de utilização gratuita para ambientes não empresariais. Ambas as opções se mostraram bastante interessantes na variedade e no aspeto dos gráficos que disponibilizam, no entanto implicam um esforço considerável na sua parametrização.

A escolha da biblioteca a utilizar recaiu sobre o Google Charts[10], biblioteca desenvolvida e disponibilizada online pela Google. Os fatores que contribuíram para a escolhida desta biblioteca devem-se essencialmente a alguma experiência de utilização da referida biblioteca, assim como à simplicidade de parametrização e inclusão dos gráficos em ambientes web. Uma desvantagem deste recurso prende-se

[7] http://www.qlik.com
[8] http://d3js.org/
[9] http://www.highcharts.com/
[10] https://developers.google.com/chart

com a necessidade permanente de acesso à Internet para a geração e visualização dos gráficos obtidos recorrendo à referida biblioteca. No seguimento da implementação dos procedimentos de recolha de tweets apresentados no Capítulo 4 e do desenvolvimento da REST API apresentada neste Capítulo, foi utilizada a linguagem Python. Como tal seria natural a escolha de uma biblioteca igualmente em Python para a criação do Dashboard, embora tal opção não tenha sido considerada inicialmente. No entanto, já numa fase em que o Dashboard se encontrava praticamente concluído, foi analisa a biblioteca Bokeh[11]. Esta biblioteca desenvolvida em Python, permitiria uma integração mais objetiva com os restantes módulos do Sistema de Informação, nomeadamente na reutilização das pesquisas de extração de informação do MongoDB, evitando os ajustes que foi necessário efetuar em algumas *queries* de modo a possibilitar a sua inclusão em páginas PHP que executam a biblioteca Google Charts.

5.3 Dashboard Inicial usando MySQL

Numa primeia abordagem ao desenvolvimento do Dashboard foi utilizada a base de dados relacional MySQL (Brogueira et al., 2015a). Esta base de dados foi utilizada exclusivamente para armazenar algumas estatísticas e indicadores pré-processados relativos ao processo de recolha de tweets. Durante algum tempo em que o trabalho foi decorrendo, houve a necessidade de duplicação de alguma informação entre as bases de dados MySQL e MongoDB, dado que o Dashboard apresentado na Secção 5.4 foi inicialmente desenhado tendo por base a informação contida no MySQL.

A utilização do MySQL revelou-se bastante útil num estado preliminar do desenvolvimento do trabalho dada a maior familiarização com os conceitos de bases de dados relacionais, permitindo em simultâneo a aprendizagem e familiarização com o paradigma das bases de dados NoSQL, visto que o MongoDB sempre foi utilizado como o sistema de base de dados para armazenamento dos tweets no formato JSON original. No processo de aprendizagem foram sendo otimizadas e automatizadas algumas tarefas relacionadas com a utilização do MongoDB, pelo que por vezes foi necessário proceder ao reprocessamento de todos os tweets contidos nos ficheiros em disco voltanto a ser inseridos no MongoDB, pelo que as estatísticas previamente armazenadas no MySQL foram preponderantes na confirmação de que não ocorria perda de dados e que os processos apresentavam as melhorias esperadas.

Ao longo do trabalho e com o evoluir do conhecimento adquirido relativamente ao MongoDB, foram sendo migradas todas funcionalidades do Dashboard que tinham sido desenvolvidas com base na informação guardada em MySQL, pelo que o MongoDB passou a ser a única base de dados utilizada até à consulsão do trabalho ficando todo o fluxo de dados do Sistema de Informação centralizado num único repositório de dados.

5.4 Visualização de Dados via Dashboard Web

A execução contínua do processo de recolha e expansão da base de dados de tweets descrita nesta tese, resulta na captação de um grande volume de dados. Para a visualização de estatísticas relativas aos dados armazenados foi desenvolvido um dashboard web. O dashboard atual permite

[11] http://bokeh.pydata.org

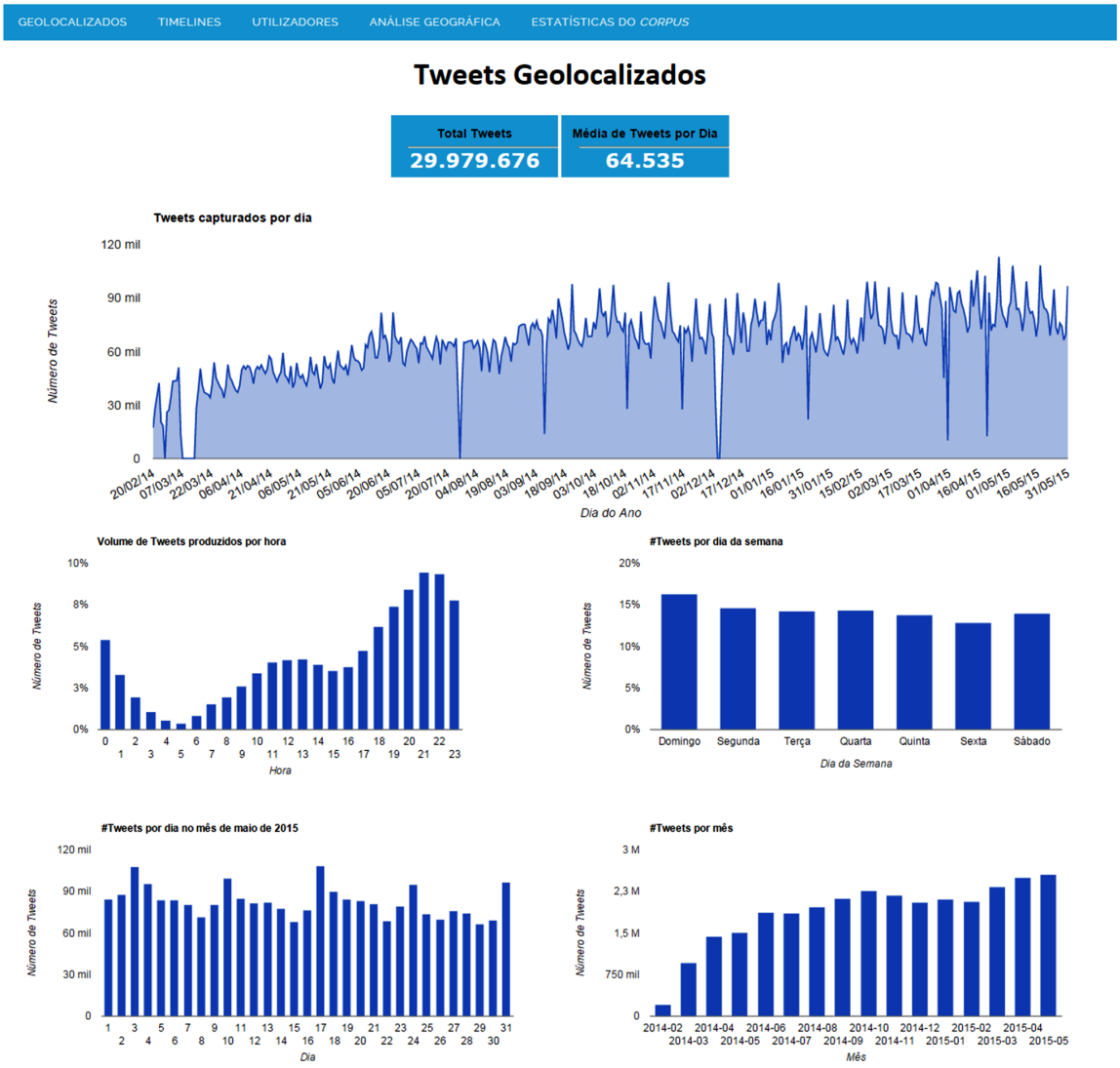


Figura 5.4: Visualização de indicadores relativos aos tweets geolocalizados.

agregar informação relativa a 5 tópicos distintos: tweets geolocalizados, tweets provenientes das timelines, utilizadores, análises geográficas e estatísticas gerais sobre o *corpus*.

5.4.1 Tweets Geolocalizados

Na Figura 5.4 são apresentadas algumas das estatísticas gerais relativas à recolha de tweets geolocalizados. Determinados gráficos, dado o volume de dados que resumem necessitam que os mesmos sejam pré-processados, pelo que não apresentam os valores em tempo real, como é o caso dos gráficos "Tweets capturados por dia" ou "Volume de Tweets produzidos por hora". O cálculo de indicadores como o apresentado em "Total Tweets", é instatâneo pelo que a cada visualização do gráfico será mostrado o valor mais atual.

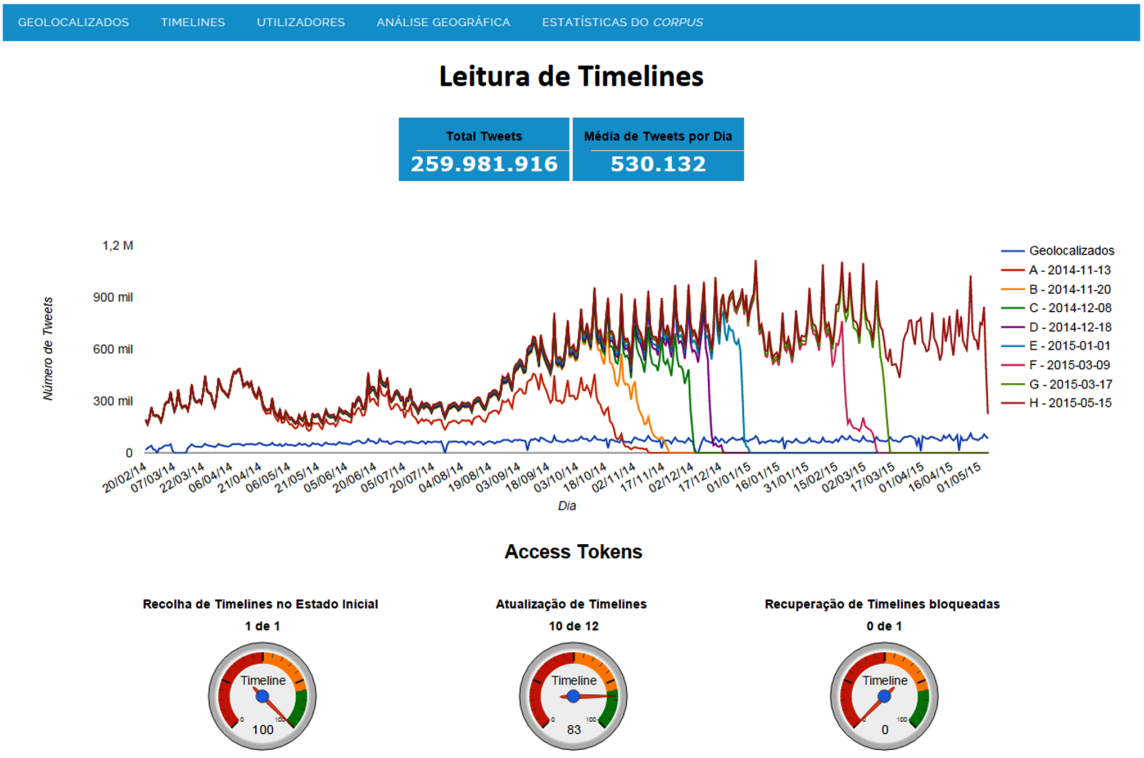


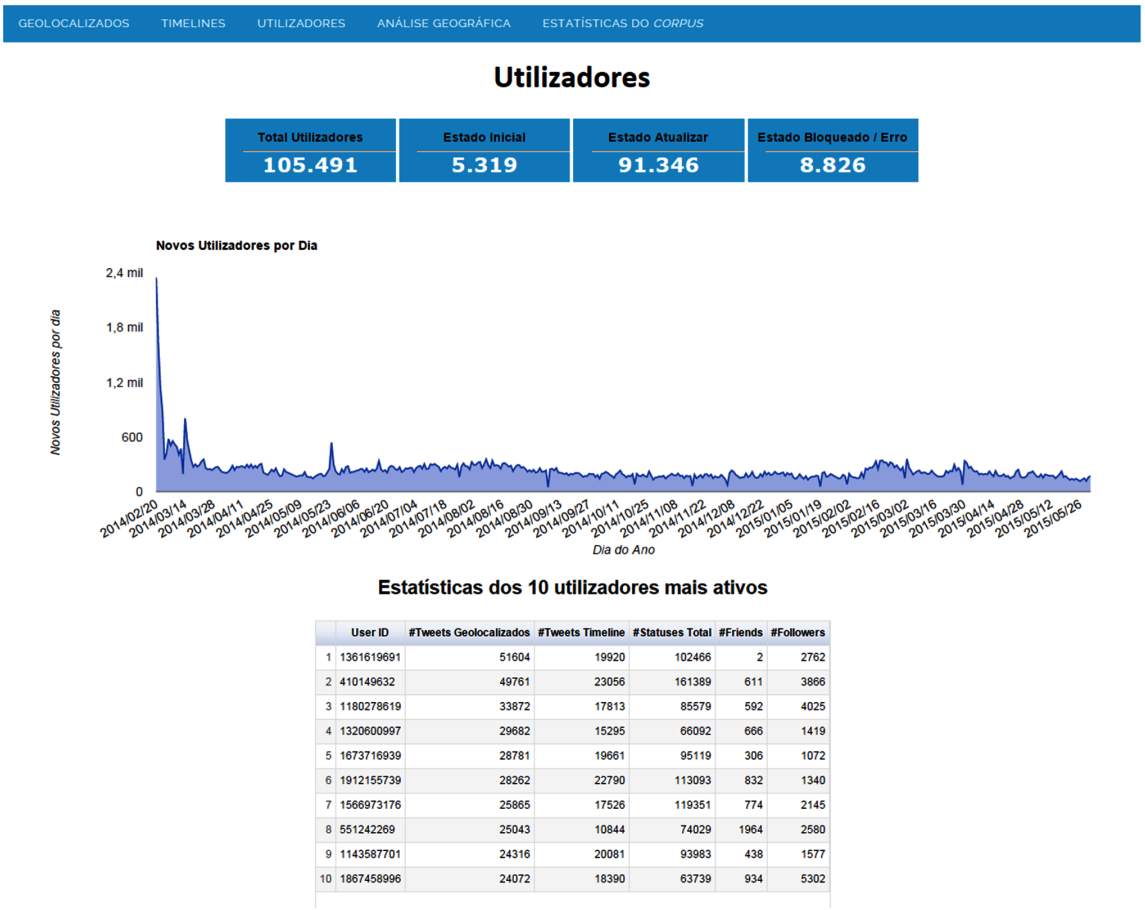
Figura 5.5: Visualização de indicadores relativos aos tweets lidos das timelines.

5.4.2 Tweets Provenientes da Timeline

A Figura 5.5 mostra o volume de tweets recuperado das timelines dos diversos utilizadores, comparando com o volume de tweets geolocalizados recolhidos em cada dia. Uma análise mais pormenorizada ao gráfico inicial da Figura 5.5 será apresentada no Capítulo 6. Tal como referido no Capítulo 4, considera-se que um cliente utilizado no processamento das timelines se encontra ocioso quando este não efetua nenhum pedido à Twitter API num período de 5 minutos. No quadro de acompanhamento do processo de leitura das timelines (Figura 5.5) foi também incluída informação que permite a visualização em tempo real do estado da alocação dos diversos *access tokens* no processamento das timelines. Os três gráficos de Gauge representam da esquerda para a direita, respetivamente: i) atividade do cliente dedicado à recolha da totalidade da timeline dos novos utilizadores integrados diariamente na base de dados; ii) atividade do conjunto de 12 clientes que procedem à atualização das timelines; iii) atividade do cliente que verifica se as timelines bloquedas ainda permanecem nesse mesmo estado sendo igualmente utilizado no reprocessamento das timelines que terminaram num estado de erro em execuções anteriores.

5.4.3 Utilizadores

Na Figura 5.6 são mostradas algumas estatísticas referentes aos utilizadores integrados na base de dados. Na parte superior deste quadro pode observar-se a distribuição dos utilizadores relativamente ao estado de processamento da respetiva timeline, que em conformidade com os indicadores da Figura 5.5 refletem em tempo real o panorama geral do processamento do conjunto das timelines



	User ID	#Tweets Geolocalizados	#Tweets Timeline	#Statuses Total	#Friends	#Followers
1	1361619691	51604	19920	102466	2	2762
2	410149632	49761	23056	161389	611	3866
3	1180278619	33872	17813	85579	592	4025
4	1320600997	29682	15295	66092	666	1419
5	1673716939	28781	19661	95119	306	1072
6	1912155739	28262	22790	113093	832	1340
7	1566973176	25865	17526	119351	774	2145
8	551242269	25043	10844	74029	1964	2580
9	1143587701	24316	20081	93983	438	1577
10	1867458996	24072	18390	63739	934	5302

Figura 5.6: Visualização de indicadores relativos aos utilizadores integrados na base de dados.

de todos os utilizadores. Na tabela visível na parte inferior da Figura 5.6 é expressa a quantidade de tweets recolhidos para cada um dos 10 utilizadores com maior nível de atividade de produção de tweets geolocalizados. É igualmente mostrado o número de friends e followers de cada um destes utilizadores, relativamente ao último tweet capturado de cada utilizador.

5.4.4 Análise Geográfica

Considerando que a constituição do *corpus* tem por base tweets geolocalizados, no quadro da Figura 5.7 são apresentados 4 mapas do território continental Português, nos quais são indicados os tweets que estão a ser recolhidos no dia atual, nos períodos entre as 0h e as 6h, as 6h e as 12h, as 12h e as 18h e por último, entre as 18h as 24h. Nesta Figura são apenas representados os tweets geolocalizados cujas coordenadas geográficas estão contidas no campo *geo*.

Nos dois mapas da parte inferior da Figura 5.7 são retratadas de forma dinâmica as estatísticas referentes ao número de tweets recolhidos e aos utilizadores identificados em cada um dos distritos de Portugal Continental.

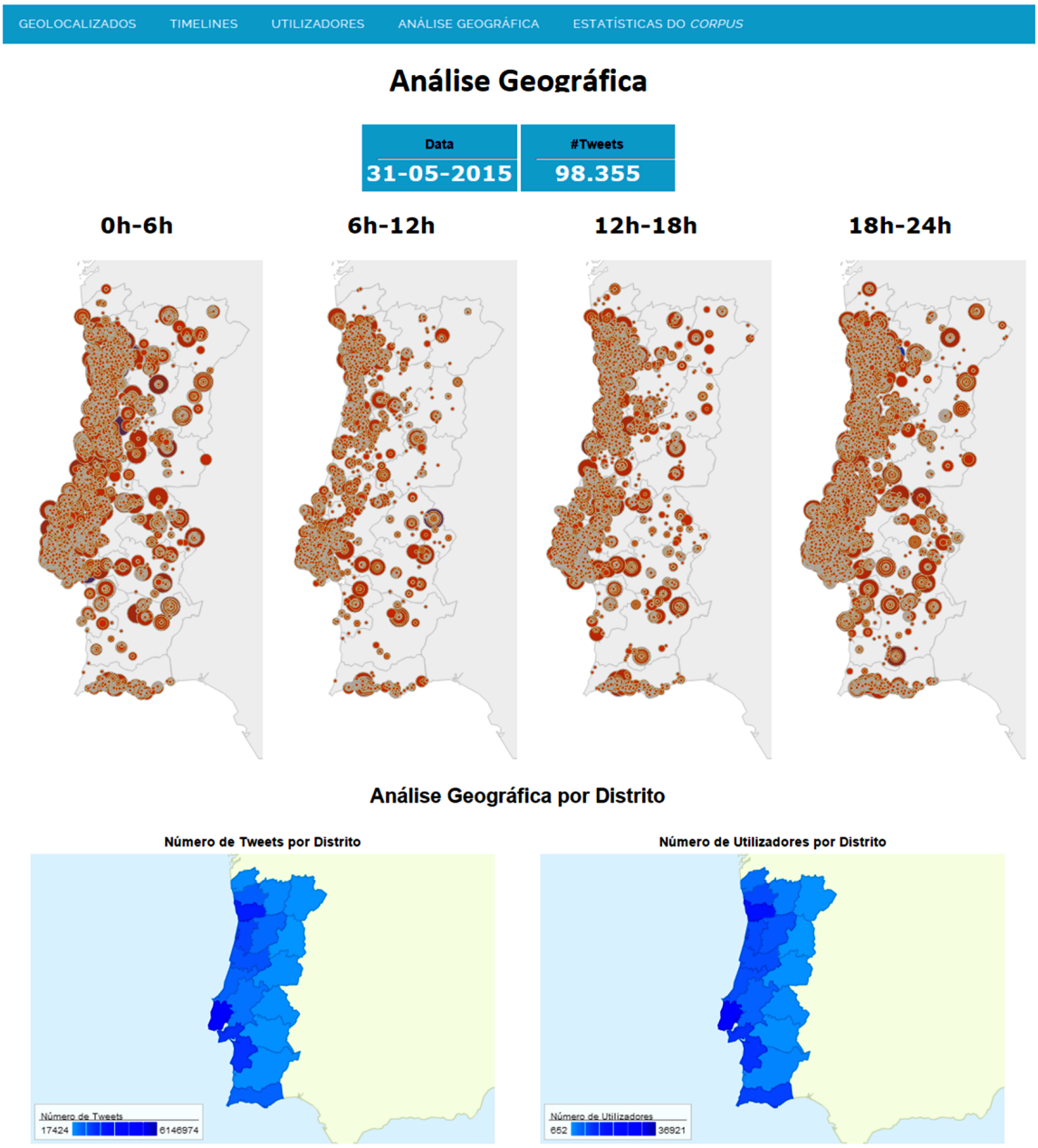


Figura 5.7: Estatísticas da distribuição geográfica dos tweets recolhidos.

5.4.5 Estatísticas do *corpus*

Relativamente a certos aspectos do conteúdo das mensagens dos tweets, no quadro da Figura 5.8 é indicado a contabilização das hashtags mais utilizadas, os utilizadores mais mensionados e as fontes de publicação de tweets mais utilizadas.

	Hashtag	Quantidade
1	It	73432
2	LT	55229
3	CarregaBenfica	41052
4	factorxsic	32374
5	LRT	23960
6	Leão	17591
7	Gémeos	17342
8	Touro	17226
9	Escorpião	16786
10	Peixes	16537
11	Câncer	16309
12	KCA	15457
13	Virgem	15204
14	TwitterOff	14760
15	Capricórnio	14723
16	Irt	14702
17	Aquário	14537
18	Sagitário	13928
19	RaparigaQueÉRapariga	13093
20	Libra	13067

	Screen_name	Menções
1	YouTube	213576
2	SignosFodas	88920
3	SigaSOJA	68301
4	VoceNaoSabiaQ	37736
5	SignosDeHoje	27540
6	instagranzin	26822
7	_Filosofei	22578
8	monalisafudida	22572
9	filipejsilvaf	19742
10	brunapereira696	18384
11	Nelson_Andradee	17027
12	pfvrchato	15904
13	freire_98	14799
14	omalestafeito	14632
15	ShareThis	13462
16	Publico	12925
17	brunzselfie	11706
18	UmFilosofoCitou	11613
19	Pedromvfranco	11121
20	miguelmartins04	10675

	Source	Quantidade
1	web	10909031
2	Twitter for Android	5517177
3	Twitter Web Client	5082769
4	Twitter for iPhone	4847210
5	Twitter for Android	1769737
6	Instagram	755507
7	Mobile Web (M2)	688839
8	Twitter for iPad	687921
9	TweetDeck	687089
10	Facebook	647622
11	Twitter for BlackBerry®	645166
12	ask.fm/ Ask.fm	223651
13	Google	196648
14	Twitter for Windows Phone	171693
15	Tumblr	166026
16	Mobile Web (M5)	152833
17	foursquare	150336
18	Twitter for Websites	140907
19	Tweet Button	135240
20	Twitter for Nokia S40	124754

Figura 5.8: Análise da distribuição geográfica dos tweets recolhidos.

5.5 Sumário

Neste Capítulo foram discutidos alguns aspetos de desenho e implementação de REST APIs, assim como os detalhes de implementação de uma REST API que possibilita o acesso ao *corpus* de tweets, abstraindo a sua estrutura e a forma como estes se encontram armazenados. A REST API para além de permitir o acesso aos tweets, poderá ser utilizada como *input* de dados para o Dashboard Web. Este Dashboard permite resumir de forma visual e por vezes em tempo real, alguns indicadores relativos não só ao desempenho e eficiência do Sistema de Informação, como à extração de estatísticas sobre o *corpus*.

Descrição, Análise e Interpretação de Dados

6

> Not everything that can be counted counts, and not everything that counts can be counted.
>
> *Albert Einstein*

Neste Capítulo é efetuada uma análise detalhada de todos os dados recolhidos sob diversas perspetivas. São analisadas na Secção 6.1 as estatísticas gerais relativas ao volume de tweets recolhidos e na Secção 6.2 as estatísticas relativas aos autores desses mesmos tweets. Na Secção 6.3 são quantificadas algumas das características das mensagens publicadas no twitter, nomeadamente as fontes de publicação, as hashtags mais utilizadas e os utilizadores mais mencionados. Na Secção 6.4 efetua-se uma breve discussão dos resultados obtidos por comparação com o volume de dados capturados em trabalhos de outros autores. O Capítulo termina com a Secção 6.5 onde são apresentados dois trabalhos desenvolvidos ao longo desta tese, cujo objetivo se centrou na análise aprofundada do subconjunto de dados obtido à data da realização dos mesmos.

6.1 Volume de Dados do *corpus*

A recolha de tweets geolocalizados foi iniciada a 20 de fevereiro de 2014 e, para efeitos de análise neste Capítulo, considera-se que terminou a 31 de maio de 2015. Neste período foram capturados da Streaming API aproximadamente 30 milhões de tweets geolocalizados escritos em português, correspondendo a uma média diária de aproximadamente 64.5 mil tweets. A Figura 6.1 indica que o número

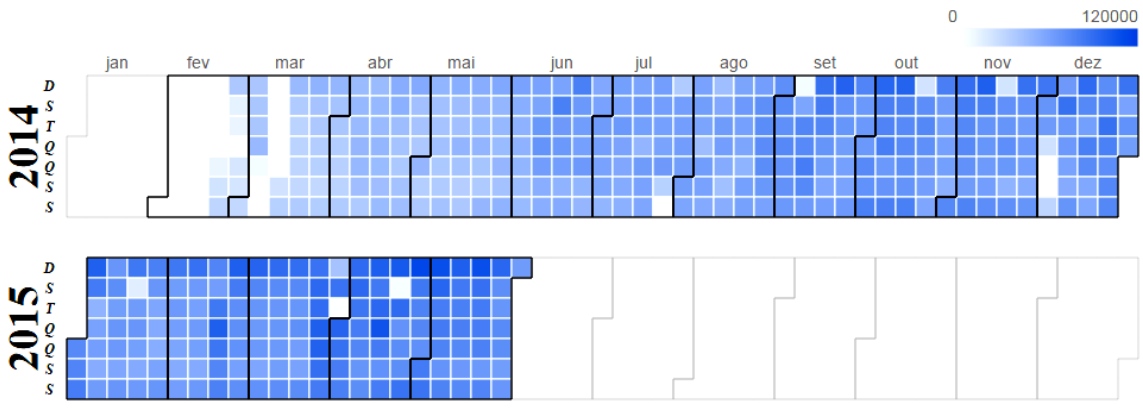


Figura 6.1: Distribuição diária da recolha dos tweets geolocalizados.

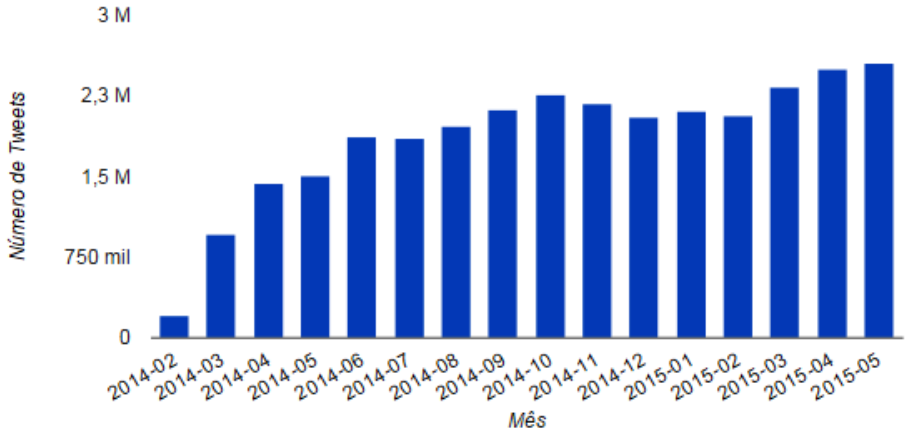


Figura 6.2: Número de tweets geolocalizados recolhidos por mês.

de tweets geolocalizados capturados por dia aumentou ao longo do tempo, onde os dias com cor branca estão relacionados com problemas pontuais nas máquinas que procedem à recolha dos tweets.

A Figura 6.2 confirma o crescente aumento da recolha de tweets geolocalizados ao longo de 2014, embora com um ligeiro decréscimo nos meses de novembro e dezembro de 2014, voltando a observar-se a tendência de crescimento nos primeiros meses de 2015.

Embora os tweets geolocalizados escritos numa língua diferente do português não sejam integrados no MongoDB, estes permanecem guardados nos ficheiros gerados em resultado das respostas da invocação da Twitter API. Numa sucinta análise verificou-se a recolha de 17 milhões de tweets em que o campo *lang* difere de *pt,* dos quais 8.4 milhões foram produzidos em Portugal e os restantes 8.6 milhões foram produzidos fora de Portugal. Dos tweets produzidos em Portugal, aproximadamente 40% foram escritos em inglês, 15% em espanhol e 18% têm o campo *lang* com valor indefinido. Com percentagem bastante mais reduzida aparece a linguagem francesa com 4.6% e o italiano com 4%. As restantes línguas têm ocorrem com uma expressão abaixo dos 3%. No estudo publicado pelo Turismo de Portugal (2014) relativo aos dados do turismo em Portugal no ano de 2014, é revelado que o Reino Unido, Espanha e França lideraram o *ranking* dos países de origem dos turistas que visitaram Portugal e que por cá pernoitaram, correspondendo pela ordem exata do top 3 de línguas estrangeiras com maior ocorrência nos tweets geolocalizados em Portugal, tal como é possível observar na Tabela 6.1.

Em resultado do procedimento exposto na Secção 4.4 cuja execução foi iniciada em meados de março de 2014, obtiveram-se aproximadamente 260 milhões de tweets das timelines de cerca de 96 mil utilizadores identificados pelo método descrito na Secção 4.3, dado que cerca de 9 mil utilizadores não permitem o acesso à sua timeline. A elevada atividade da comunidade portuguesa do Twitter pode ser evidenciada pela reduzida quantidade de tweets recolhidos das timelines para datas anteriores a fevereiro de 2014, visto que da totalidade de tweets das timelines 90.5% (~246 milhões) correspondem ao intervalo de fevereiro de 2014 a maio de 2015. Os restantes 9.5% foram publicados em datas anteriores a fevereiro 2014, conseguindo-se recuperar tweets de outubro de 2010. Nas análises efetuadas na continuação desta Secção, para simplificação na representação gráfica, serão apenas considerados os tweets das timelines produzidos entre fevereiro de 2014 a maio de 2015 tal como indicado na Figura 6.3.

A timeline de cada utilizador inclui tweets geolocalizados e não geolocalizados, logo era esperado que grande parte dos tweets geolocalizados recolhidos da Streaming API estivessem no conjunto de

Tabela 6.1: Línguas estrangeiras com maior ocorrência em tweets geolocalizados em Portugal.

Língua	*lang*	Ocorrência (%)
Inglês	en	41,490%
Espanhol	es	15,766%
Francês	fr	4,622%
Italiano	it	4,052%
Tagalo	tl	3,080%
Indico	in	2,229%
Haitian	ht	1,409%
Estoniano	et	0,903%
Holandês	nl	0,794%
Esloveno	sl	0,757%
Turco	tr	0,680%
Vietnamita	vi	0,621%
Húngaro	hu	0,603%
Eslovaco	sk	0,566%

Língua	*lang*	Ocorrência (%)
Alemão	de	0,534%
Romeno	ro	0,461%
Letão	lv	0,455%
Polaco	pl	0,421%
Finlandês	fi	0,411%
Lituano	lt	0,366%
Sueco	sv	0,269%
Galês	cy	0,230%
Dinamarquês	da	0,170%
Islandês	is	0,161%
Norueguês	no	0,155%
Bósnio	bs	0,142%
Croata	hr	0,112%
Grego	el	0,004%

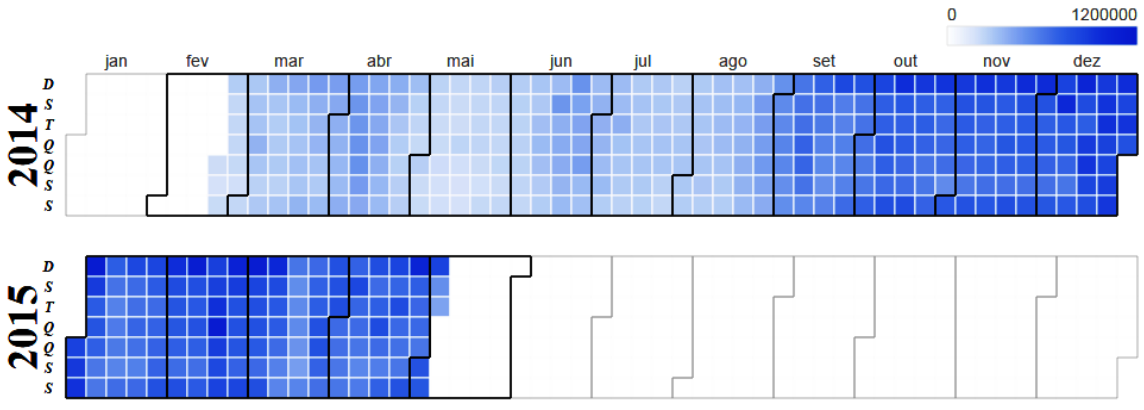


Figura 6.3: Distribuição diária do processamento de timelines.

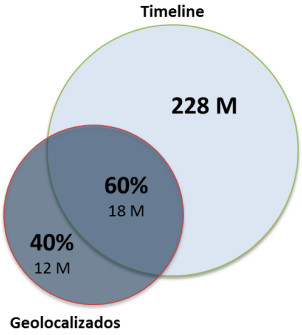


Figura 6.4: Tweets geolocalizados em comum nos dois conjuntos de dados.

tweets lidos das timelines. Tal facto é validado pela Figura 6.4 onde se verifica que 60% da coleção de tweets geolocalizados (~18 milhões) é comum à coleção de tweets da timeline, pelo que entre fevereiro de 2014 e maio de 2015 foram capturados 258 milhões de tweets distintos. Com este volume de dados verifica-se a recuperação em média de 553 mil tweets por dia, tendo este valor aumentado ao longo do tempo como consequência da inclusão diária de novos utilizadores e da contínua leitura dos novos tweets produzidos por todos os utilizadores. Caso seja considerado o volume total de tweets únicos capturados, o *corpus* é constituído por 272 milhões de tweets (260 milhões de tweets das timelines + 12 milhões de tweets geolocalizados não recolhidos novamente pelo processamento das timelines).

Os tweets provenientes das timelines representam um aumento na quantidade de tweets armazenada em cerca de 8 vezes relativamente ao volume de tweets recolhidos em tempo real do fluxo da Streaming API. Pontualmente, para alguns dias foi possível recuperar mais de 1 milhão de tweets pelo método da leitura de timelines, perfazendo um acréscimo de sensivelmente 10 vezes relativamente ao volume de dados disponibilizado na Streaming API nesses mesmos dias. O gráfico da Figura 6.5 mostra a comparação entre o volume de tweets geolocalizados recolhidos diariamente com o volume de tweets obtidos das timelines. A linha a azul, legendada por ”Geolocalizados”, representa o número de tweets geolocalizados capturados por dia ao longo de todo o período e as restantes linhas representam sucessivas medições da quantidade de tweets das timelines, nos instantes de tempo indicados na respetiva legenda, de onde se podem retirar diversas ilações. A linha H sobrepõe-se a todas a restantes linhas de A a G derivado à introdução diária de novos utilizadores, possibilitando a recuperação de tweets transversalmente a todo o período em análise. Outro ponto relevante está relacionado com o facto de que mesmo em dias onde ocorram problemas na recolha de tweets geolocalizados, a atualização iterativa da timeline permite obter tweets para esses mesmos dias, inclusivamente tweets geolocalizados por consequência de estes também estarem incluídos na timeline. Ainda outra observação interessante, deve-se à acentuada oscilação verificada sensivelmente a partir de outubro de 2014, quanto ao volume de tweets recuperados por dia. A oscilação está correlacionada com o dia da semana, sendo que os máximos locais correspondem a uma maior atividade nos primeiros dias da semana (domingo e segunda-feira), notando-se o decréscimo da atividade ao longo da semana, sendo este mais acentuado nos dias de sexta-feira e sábado, correspondendo aos mínimos locais. Tal oscilação tornou-se mais pronunciada a partir de outubro de 2014 devido ao facto da execução do algoritmo ter estabilizado, sendo possível a sua execução de forma contínua e sem grandes interrupções, e também devido ao crescimento do número de utilizadores permitindo recolher um maior volume de tweets.

O padrão de atividade ao longo da semana é confirmado pelo gráfico da Figura 6.6 onde está

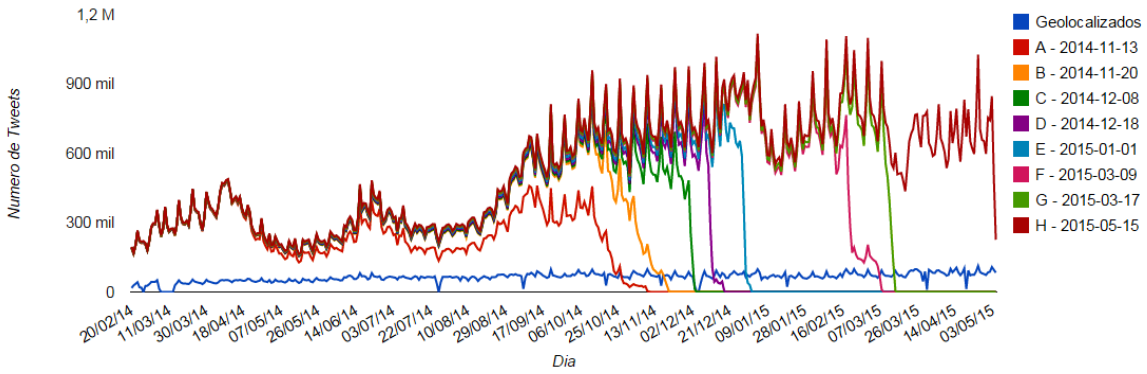


Figura 6.5: Volume de tweets recolhidos por ambos os métodos.

bem patente uma maior atividade de publicação de tweets nos primeiros dias da semana. Este comportamento poderá justificar-se em parte, pelo facto da comunidade Portuguesa presente no Twitter, ser essencialmente composta por adolescentes ou jovens adultos que aproveitam as sextas-feiras e sábados para saírem com os amigos diminuindo a sua atividade no Twitter (Brogueira et al., 2014a).

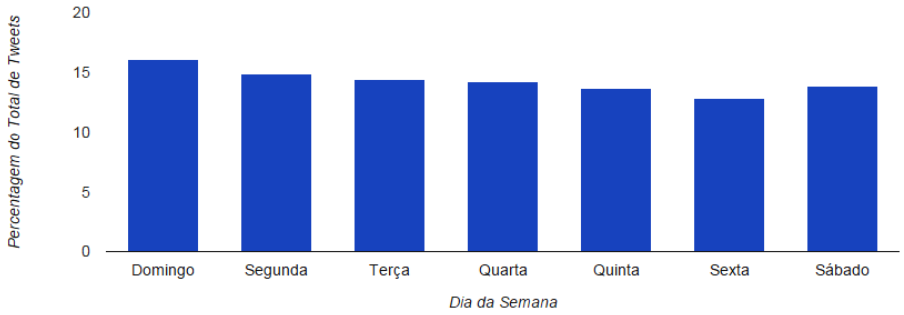


Figura 6.6: Distribuição dos tweets recolhidos por dia da semana.

Outra evidência do padrão de atividade durante a semana está bem patente no gráfico da Figura 6.7 onde é analisado a quantidade de tweets geolocalizados recolhidos por dia, durante o mês de maio de 2015. Os cinco máximos locais correspondem ao dia de domingo, observando-se a diminuição da atividade no decorrer da semana.

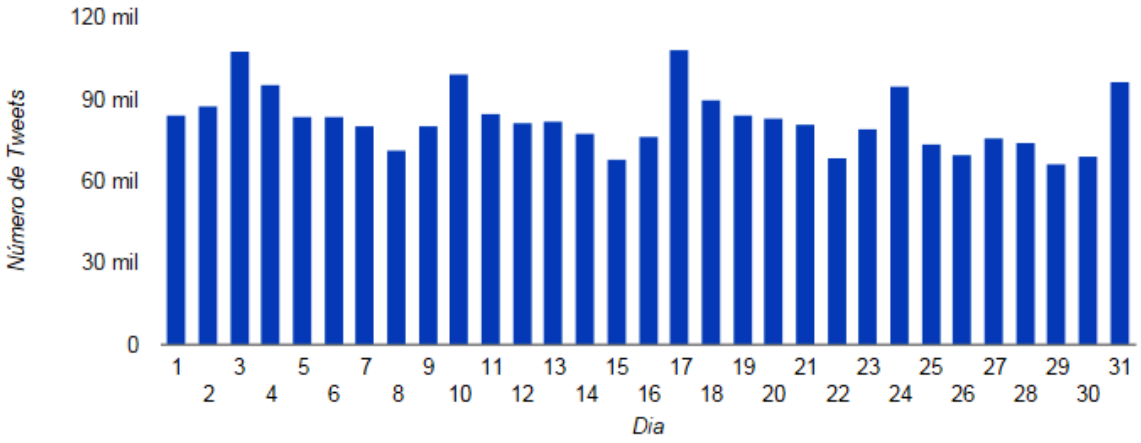


Figura 6.7: Tweets armazenados por dia no mês de maio de 2015.

No gráfico da Figura 6.8 analisa-se a variação da publicação de tweets ao longo do dia, verificando-se dois períodos em que a atividade é mais intensa: o primeiro situa-se por volta da hora de almoço, crescendo progressivamente a partir do final da tarde atingindo-se o pico de atividade entre as 21h e as 22h.

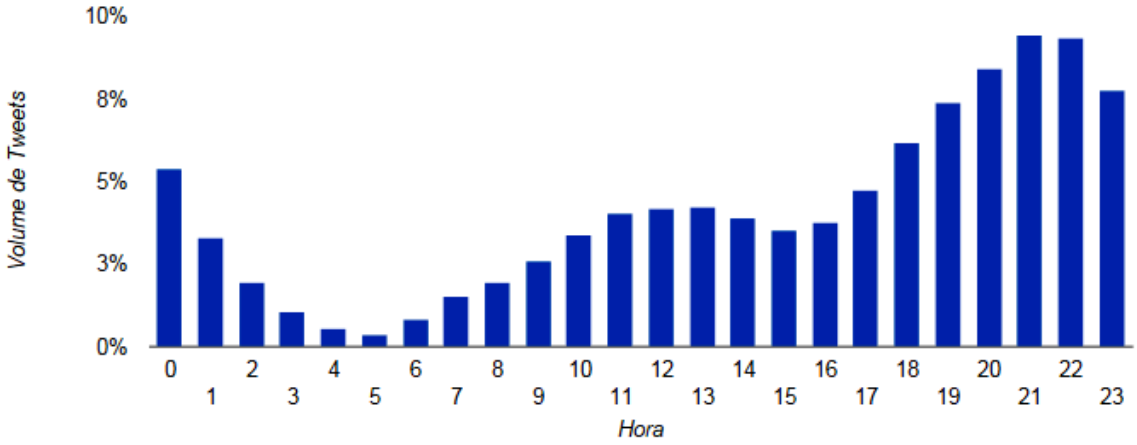


Figura 6.8: Distribuição do volume de tweets produzidos por hora do dia.

6.2 Utilizadores Portugueses do Twitter

A execução do procedimento descrito na Secção 4.3.3 resultou na identificação de aproximadamente 105 mil utilizadores, verificando-se que este número cresce de forma linear ao longo do tempo, sendo integrados cerca de 232 novos utilizadores por dia, tal como demonstra a Figura 6.9. A execução contínua deste procedimento poderá permitir a identificação de grande parte da comunidade portuguesa presente no Twitter desde que cada utilizador efetue uma publicação em que permita a transmissão da sua geolocalização.

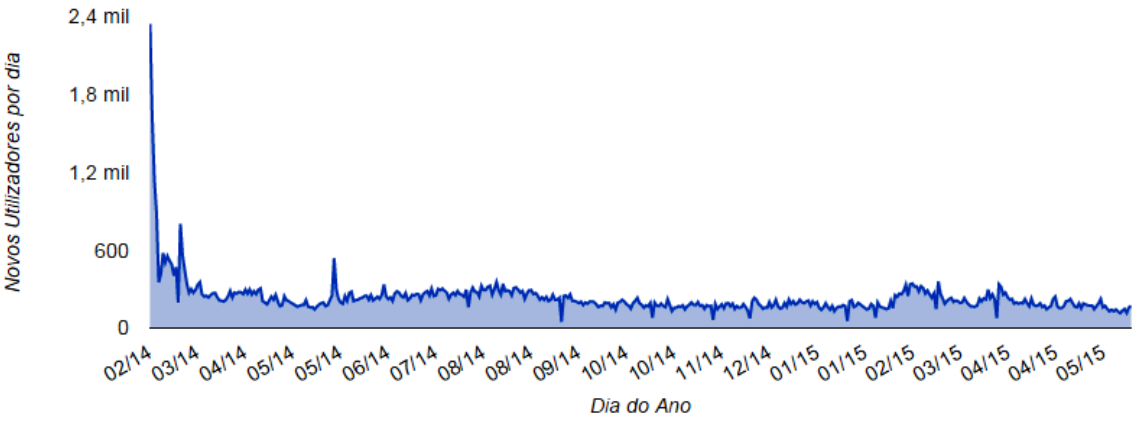


Figura 6.9: Novos utilizadores por dia.

Quanto ao número de tweets geolocalizados capturados de cada utilizador, observa-se na Figura 6.10 que cerca de dois terços dos utilizadores tem uma reduzia atividade geolocalizada, mas por outro

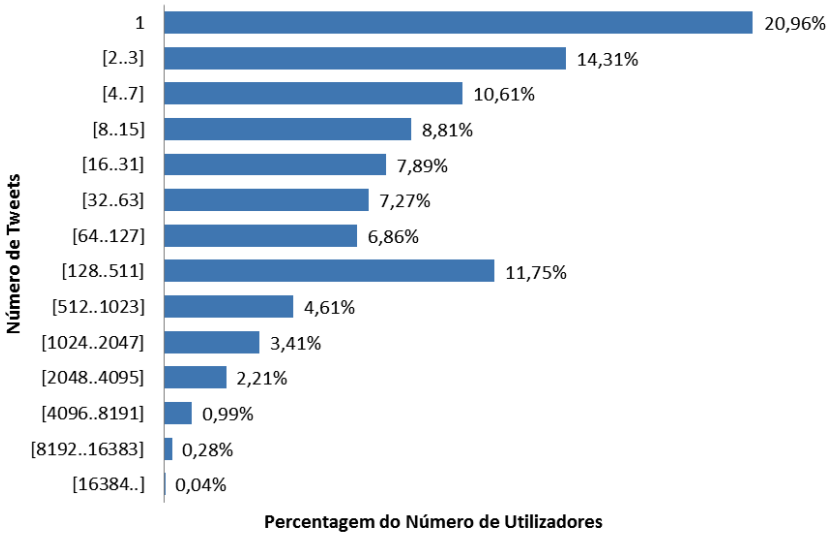


Figura 6.10: Número de tweets geolocalizados por utilizador.

lado cerca de 7% tem uma atividade acima das 1000 mensagens geolocalizadas e além disso são estes os utilizadores responsáveis pela produção de mais de 70% do total de tweets geolocalizados.

No que diz respeito ao número de tweets recolhidos do histórico de mensagens de cada utilizador a Figura 6.11 demonstra que para 51% dos utilizadores recuperaram-se mais de 1000 tweets e apenas cerca de 13% têm uma timeline reduzida com menos de 15 tweets. De referir que dos 105 mil utilizadores identificados, 7% têm deliberadamente bloqueado o acesso à sua timeline. Estes dados sugerem que a atividade da comunidade portuguesa do Twitter é maioritariamente não geolocalizada.

Todo o tweet contém no seu conteúdo diversa informação relativa ao seu autor, permitindo a caracterização deste segundo diversos parâmetros ,dos quais se destacam: i) *followers_count*: número de seguidores; ii) *friends_count*: número de utilizadores seguidos; iii) *statuses_count*: total de mensagens publicadas. Esta informação é variável uma vez que o número de *followers* ou *friends* pode ser diferente entre duas publicações consecutivas, dependendo da dinâmica das interações do utilizador com os seus pares no Twitter. A cada publicação o valor de *statuses_count* é incrementado em uma unidade. Tendo por base este conjunto de valores, efetuou-se uma análise da sua variação entre o primeiro e último tweet presente no *corpus*, relativamente a cada um dos utilizadores. A Figura 6.12 apresenta o total de mensagens publicadas tendo em conta a diferença dos valores do campo *statuses_count* entre o primeiro e último tweet. Este valor é condicionado pela data em que determinado utilizador foi integrado na base de dados, logo é possível que a variação de tweets produzidos pelos utilizadores integrados à mais tempo possa ser superior. Importa salientar que 18% de utilizadores produziram mais de 16 mil tweets, ao passo que para apenas 1% de utilizadores foi recolhido mais de 16 mil tweets de histórico (ver Figura 6.11). Esta diferença pode estar correlacionada com diversos fatores: i) frequência de publicação de tweets bastante superior à limitação de extração de informação da Twitter API; ii) utilizadores portugueses que publicam muitas mensagens em outras línguas que não o Português Europeu ou fora de Portugal; iii) intervalo de tempo relativamente elevado entre a atualização da timeline para utilizadores bastante ativos, provocando a perda de parte da informação. Estes dados indicam que o processo apresentado neste trabalho pode ser melhorado, tendo em vista a recuperação de um maior volume de informação sobre os utilizadores mais ativos.

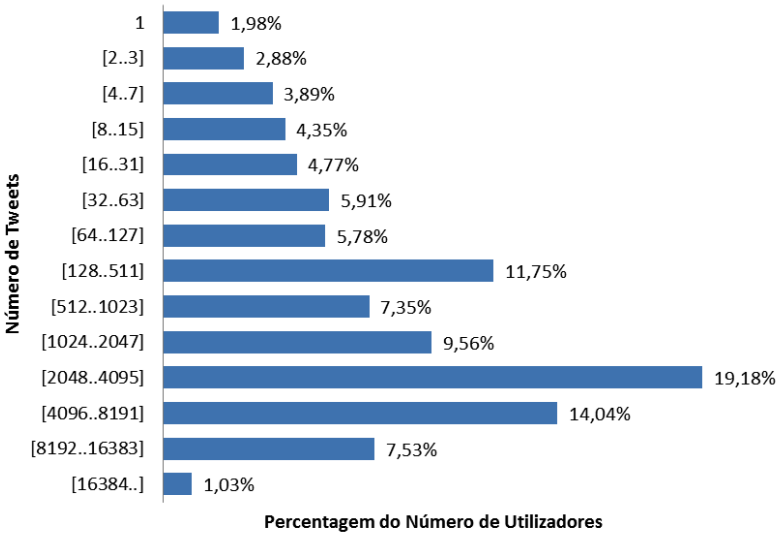


Figura 6.11: Número de tweets lidos da timeline de cada utilizador.

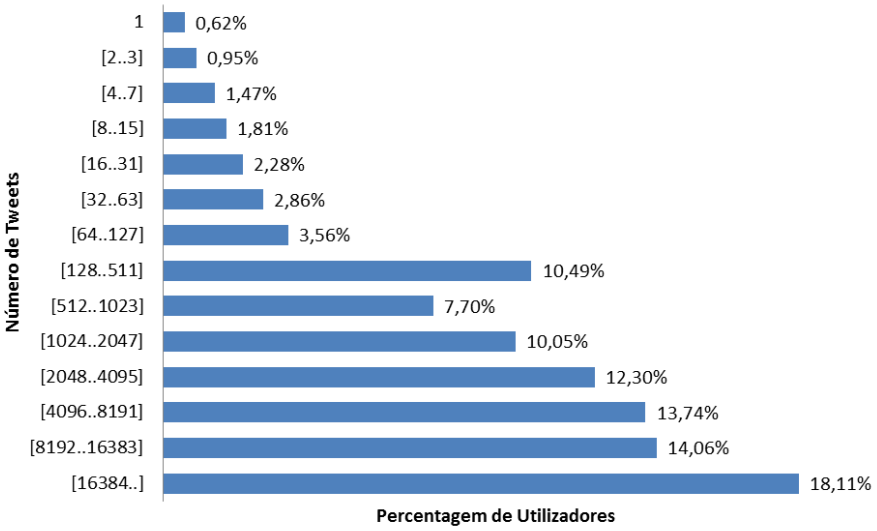


Figura 6.12: Número de tweets produzidos por utilizador.

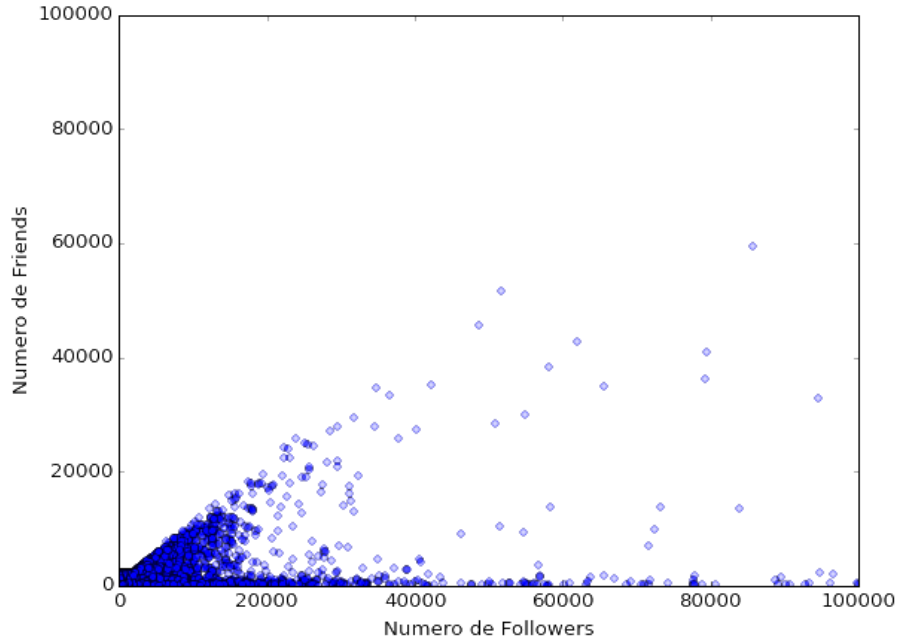


Figura 6.13: Relação entre o número de *followers* e o número de *friends*.

Relativamente à análise da variação do número de *followers* e *friends*, o gráfico da Figura 6.13 revela-nos que na topologia da comunidade portuguesa do Twitter, utilizadores com um número elevado de *friends* têm um número elevado número de *followers,* embora o inverso não seja propriamente verdade, visto que bastantes utilizadores têm um grande número de *followers* sem no entanto terem um volume considerável de *friends*.

A quantidade de *followers* de determinado utilizador pode ter bastante relevância, podendo ser um indicador da sua popularidade e prestígio tal como refere Hutto et al. (2013). O mesmo autor analisou as variações no número de *followers* e *friends* de 507 utilizadores por um período de 15 meses, entre agosto de 2010 e outubro de 2011. Sobre este conjunto de utilizadores verificou que a maioria dos utilizadores tinham entre 176 a 949 *followers* e entre 135 e 661 *friends*. Em termos de mediana os valores apurados por Hutto et al. (2013) foram 391.5 e 289.5, para os *followers* e *friends* respetivamente. No caso dos 105 mil utilizadores analisados neste trabalho, a maioria tem entre 128 e 1023 *followers* e *friends* e mediana 224 e 216 respetivamente, como é possível confirmar na Figura 6.14. Dado que os valores da mediana são bastante inferiores aos obtidos por Hutto et al. (2013), poderá concluir-se que a densidade de ligações de *followers* e *friends* é menor na comunidade portuguesa do Twitter por comparação com os 507 utilizadores analisados por Hutto et al. (2013). Tal como é referido também por Hutto et al. (2013), construir uma audiência de *followers* pode criar o acesso a uma rede de laços sociais, recursos e influência. No entanto, pouco se sabe sobre como fazer crescer tal audiência. Diversos fatores podem afetar a formação dos laços sociais e sua dissolução ao longo do tempo, como por exemplo, o comportamento social e o conteúdo das mensagens ou aspetos relacionados com a estrutura da rede social, podendo ajudar na previsão do aumento das ligações entre utilizadores. O *corpus* resultante deste trabalho pode constituir uma importante fonte de informação para este tipo de estudos, tendo em conta o número de utilizadores e o volume de informação guardada sobre cada utilizador.

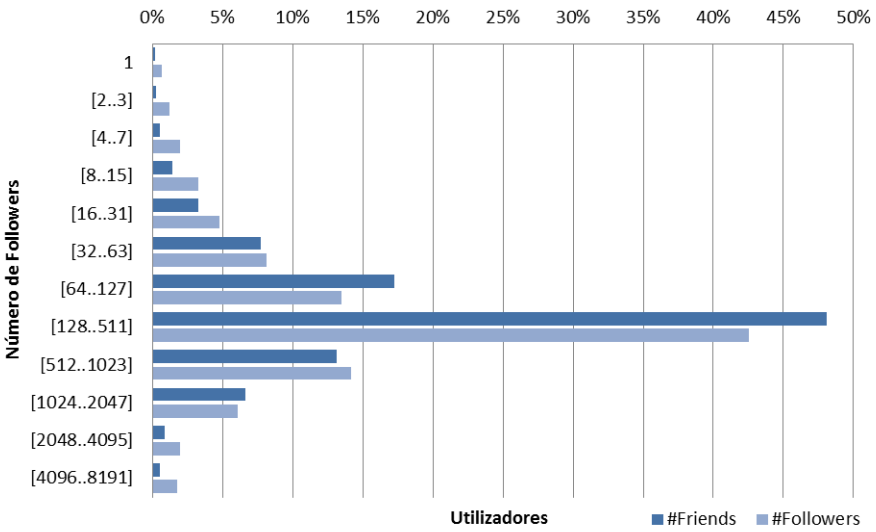


Figura 6.14: Número de *followers* e *friends* por utilizador.

6.3 Estatísticas sobre o *corpus*

Considerando um subconjunto aleatório de 35.5 milhões de tweets do *corpus*, extraíram-se diversas métricas relativamente ao conteúdo da mensagem dos tweets nomeadamente o número de *retweets*, utilizadores e autores, links, *hashtags* mais populares, utilizadores mais citados e as fontes de produção de tweets mais usadas, de modo a tentar caracterizar a utilização do Twitter em Portugal. Ao longo de todo o texto, têm sido utilizados os conceitos de ”utilizador” e ”autor” com o mesmo significado, ou seja, indicando o indivíduo que produziu determinado tweet. Apenas nesta Secção os dois conceitos terão significados distintos. ”Autor” significará quem produziu determinado tweet e ”utilizador” corresponderá a todos os indivíduos mencionados no texto de um tweet, independentemente de terem sido autores de tweets incluídos no *corpus*.

Verificou-se na Secção 6.2 que o total de 258 milhões de tweets do *corpus* foi produzido por 105 mil autores. Nos 35.5 milhões de tweets analisados em 37.2% é efetuada a menção a pelo menos um utilizador, tendo sido mencionados cerca de 624 mil utilizadores distintos. De notar que esta informação poderá revelar-se bastante útil na melhoria do processo de expansão do *corpus* visto aumentar consideravelmente o número de utilizadores sobre os quais se poderá efetuar a leitura da timeline, incrementando em muito a base de dados de utilizadores sobre os quais é possível analisar o seu comportamento online, tendo em conta não só o conteúdo, localização e frequência das mensagens que publica como as informações que partilha do seu perfil.

No subconjunto de tweets analisados aproximadamente 19.6% são *retweets*. No contexto do Twitter a ação de *retweet* é um método muito utilizado na disseminação de informação. Embora este método seja utilizado com alguma expressão pela comunidade Portuguesa no *corpus* recolhido por McMinn et al. (2013), cerca de 30% das mensagens são *retweets*. De notar igualmente que uma percentagem elevada de *retweets* pode estar muitas vezes relacionado com a ação de propagação de *spam* (Boyd et al., 2010), sendo os *retweets* muitas vezes descartados para efeitos de outro tipos de estudos dada a repetição de mensagens.

Tabela 6.2: Hashtags mais utilizadas.

Hashtag	#Ocorrências	Hashtag	#Ocorrências
#lt	128661	#KCA	15457
#CarregaBenfica	41052	#Virgem	15204
#LRT	38662	#TwitterOff	14760
#factorxsic	32374	#Capricórnio	14723
#Leão	17591	#Aquário	14537
#Gêmeos	17342	#Sagitário	13928
#Touro	17226	#RaparigaQueÉRapariga	13093
#Escorpião	16786	#Libra	13067
#Peixes	16537	#webcamtoy	12479
#Câncer	16309	#np	12097

Tabela 6.3: Utilizadores mais citados.

Autor	#Citações	Autor	#Citações
@YouTube	213576	@Nelson_Andradee	17027
@SignosFodas	88920	@pfvrchato	15904
@SigaSOJA	68301	@freire_98	14799
@VoceNaoSabiaQ	37736	@omalestafeito	14632
@SignosDeHoje	27540	@ShareThis	13462
@instagranzin	26822	@Publico	12925
@_Filosofei	22578	@brunzselfie	11706
@monalisafudida	22572	@UmFilosofoCitou	11613
@filipejsilvaf	19742	@Pedromvfranco	11121
@brunapereira696	18384	@miguelmartins04	10675

Verificou-se também que 10.6% dos tweets contêm no texto a referência a pelo menos um URL, permitindo a difusão de informação de forma simples e rápida. Relativamente aos tweets em que é utilizado pelo menos uma hashtag para efetuar a referência a tópicos ou destacar algum assunto, tal ocorre em apenas 7.9% dos tweets. Na Tabela 6.2são apresentadas as 20 hashtags mais utilizadas no subconjunto de tweets analisado. Nas hashtags mais utilizadas tem predominância a utilização de palavras relacionadas com os signos do zodíaco, remetendo-nos para assuntos relacionados com a astrologia. Hashtags que fazem referência ao último tweet (#lt - last tweet) ou ao útimo retweet (#LRT - last retweet) estão no top das mais utilizadas, de acordo com o verificado na Secção 6.5.1.

A Tabela 6.3 contém a lista dos 20 utilizadores mais citados. A influência de determinado utilizador poderá ser quantificada em função do número de citações relativas a esse mesmo utilizador. De facto, a avaliar pelo *screen_name* dos utilizadores mais citados verifica-se algum relacionamento com as hashtags mais usadas (@SignosDeHoje).

Relativamente às fontes utilizadas na publicação dos tweets, a Web é o meio mais utilizado confirmando a mesma tendência verificada por Petrović et al. (2010). No entanto, no top 10 das fontes mais utilizadas identificadas por Petrović et al. (2010) não predominam as aplicações para dispositivos móveis, tal como é observado na Figura 6.15. Esta alteração de paradigma estará certamente relacionada com a massificação da utilização de dispositivos móveis com acesso à Internet, permitindo a interação e partilha nas redes sociais em qualquer lugar e a qualquer instante. De acordo com esta afirmação verifica-se que atualmente as aplicações para publicação de tweets em dispositivos móveis com tecnologia Android ou Apple estão entre o top de preferências dos utilizadores do Twitter, seguindo os indicadores apresentados por Framingham (2015) relativamente ao mercado dos sistemas operativos

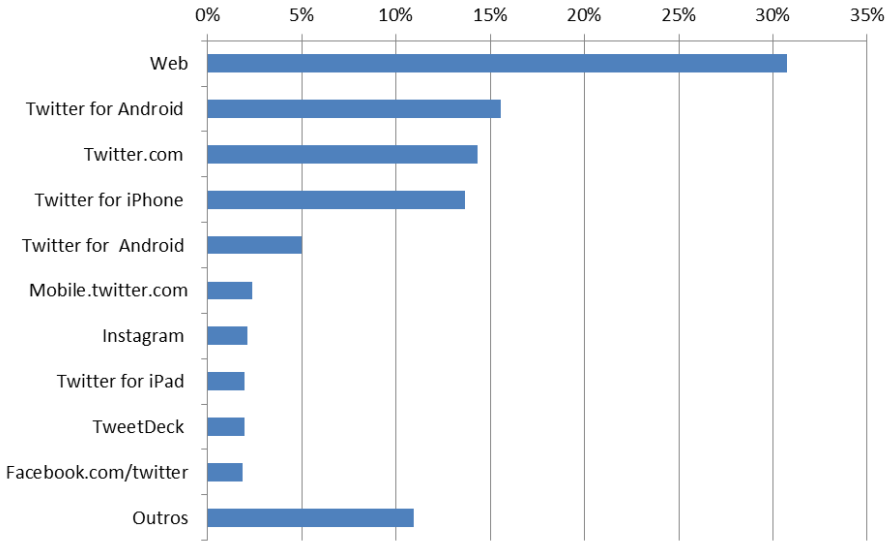


Figura 6.15: Diferentes fontes de produção de tweets.

utilizados em *smartphones*. A publicação de tweets a partir de outras redes sociais, como é o caso do Facebook e Instragram é outra tendência igualmente verificada.

6.4 Discussão dos Resultados

A Figura 6.16 apresenta a relação entre a produção estimada de tweets e a quantidade efetivamente recolhida. Embora considerando a volatilidade dos valores do campo *statuses_count*, dado que cada utilizador pode apagar tweets publicados no passado, o *corpus* contém um valor muito próximo dos 70% do volume total estimado de tweets produzidos entre fevereiro de 2014 e maio de 2015. Para esta análise, foi considerada a diferença do valor do campo *statuses_count* entre o primeiro e o último tweet registado para cada utilizador, dando uma estimativa do número de tweets produzidos. Os utilizadores cuja diferença resultou em valores negativos, não foram considerados. Foram também considerados somente os utilizadores cuja timeline tinha sido processada pelo menos uma vez. A percentagem de cobertura relativamente baixa, pode estar correlacionada com diversos fatores, nomeadamente:

- Utilizadores que publicam mensagens que não são escritas em Português Europeu ou que publicam fora de Portugal. Os tweets produzidos que satisfaçam uma destas duas condições são contabilizados no número de tweets produzidos pelo utilizador, mas mesmo que sejam lidos da timeline não serão considerados ”válidos” para o âmbito do *corpus* que se pretende construir neste trabalho, fazendo baixar a percentagem de cobertura dos dados recolhidos;
- A dificuldade em manter a execução contínua do processo de leitura das timelines, fez com que bastantes tweets não tenham sido recuperados. O volume de tweets produzidos por determinados utilizadores é bastante elevado e muitas vezes superior aos 3200 que é possível recuperar em cada iteração do processo de leitura e atualização das timelines. Em concreto, registaram-se 7 utilizadores que pelos valores contidos no campo *statuses_count* é expectável que tenham produzido mais de 150 mil tweets, que por motivos de maior clareza do gráfico da Figura 6.16 não se encontram incluídos no mesmo.

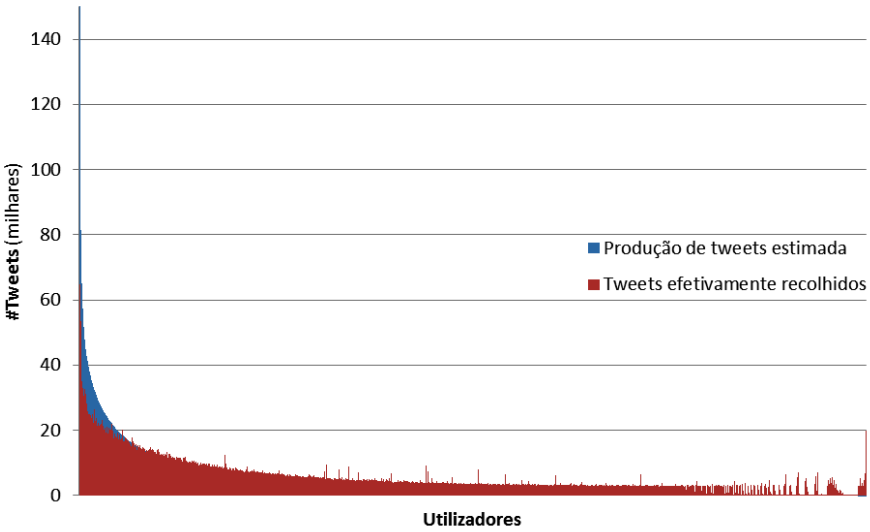


Figura 6.16: Cobertura dos dados recolhidos.

- A hipótese da existência de algum erro no processo de leitura das timelines também é totalmente descartado, podendo contribuir para a não recuperação de alguns tweets.

De realçar que para alguns utilizadores o número de tweets recolhidos é superior ao número de tweets que foi estimado terem sido produzidos, tal como se observa à direita da Figura 6.16. Esta situação é devida à possibilidade, já referida, de ao longo do tempo ser permitido apagar tweets produzidos no passado, mas essa ação não é replicada no *corpus* resultante deste trabalho, pelo que os tweets permanecerão na base de dados fazendo com que, nestes casos, o número de tweets recolhidos seja superior ao valor indicado no campo *statuses_count* de determinado utilizador, associado a cada um dos seus tweets.

Considerando a multiplicidade de aplicações e análises que podem ser efetuadas sobre cada uma das informações que constituem um tweet, optou-se por guardar o objeto JSON completo retornado pelo Twitter ocupando em média 3 a 4 Kb por tweet, pelo que o armazenamento dos cerca de 290 milhões de tweets ocupou à volta de 1 TB. Não foi explorada a característica de escalabilidade horizontal proporcionada pelo MongoDB utilizando-o apenas numa única máquina, pelo que houve a necessidade de gerir cuidadosamente o espaço disponível em disco. Notou-se alguma ineficiência na gestão dos recursos por parte do MongoDB 2.4, nomeadamente no excessivo consumo da memória e espaço ocupado pelos indices que foram criados por motivos de otimização na pesquisa de informação. Tais constrangimentos fizeram com que o processo de leitura e atualização das timelines, por vezes, tivesse de ser suspenso por alguns períodos de tempo, de modo a efetuar operações de otimização do espaço ocupado. Não obstante, com o método apresentado obteve-se um volume de dados bastante superior quando comparado com trabalhos de outros autores. Para efeitos de comparação com os métodos de outros autores, serão considerados todos os tweets únicos armazenados no *corpus* totalizando 272 milhões. A Tabela 6.4 resume os valores obtidos em trabalhos de outros autores encontrando-se as colunas Utilizadores, Tweets e Tweets/Dia em milhares.

Na comparação entre os diversos *corpus* resultantes dos trabalhos representados na Tabela 6.4 deverá ter-se em consideração que a recolha dos tweets foi efetuada em momentos bastante distintos.

Tabela 6.4: Comparação do volume de dados armazenados com trabalhos de outros autores.

Autores	Tempo de Recolha	#dias	#utilizadores	#Tweets	#Tweets/dia
Java et al. (2007)	01-04-2007 a 30-05-2007	59	76 k	1300 k	22 k
Wang (2010)	03-01-2010 a 24-01-2010	22	25 k	500 k	23 k
Ghiassi et al. (2013)	06-05-2013 a 08-06-2013	33	–	10300 k	31 k
Bošnjak et al. (2012)	11 meses	330	81 k	14500 k	44 k
Brogueira et al.	*20-02-2014 a 31-05-2015*	*466*	*105 k*	*272000 k*	*583 k*
Petrović et al. (2010)	11-11-2009 a 01-02-2010	83	–	97000 k	1017 k

Nota-se uma natural menor atividade em 2007 dado que o Twitter foi lançado apenas em julho de 2006 e a recolha efetuada por Java et al. (2007) foi realizada ainda antes do primeiro aniversário do Twitter. Por outro lado, utilizando a Streaming API Petrović et al. (2010) conseguiu obter mais de 1 milhão de tweets por dia em 2010, quando eram produzidos cerca de 35 milhões de tweets por dia[1] e na versão 1.0 da Twitter API em vigor até 2014 eram disponibilizados mais do que 1% do volume total de tweets produzidos em determinado momento. De notar que os autores não referem a aplicação de qualquer filtragem aos dados recolhidos, pelo que colecionaram todos os tweets disponibilizados no fluxo da Streaming API, sem qualquer restrição.

Considerando as particularidades associadas ao período durante o qual os dados são recolhidos, as limitações impostas pela Twitter API, o tipo de análise pretendida e a filtragem imposta aos tweets são factores que contribuem para a limitação do volume de dados colecionados, não sendo possível efetuar a comparação entre os diversos métodos com base nos mesmos pressupostos. De entre os trabalhos apresentados na Tabela 6.4, a arquitetura proposta nesta tese apresenta o segundo melhor desempenho relativamente ao número de tweets capturados para cada dia. Ainda assim e dada a necessidade de formação de *corpus* para posteriores análises sobre diversas perspetivas (Petrović et al., 2010), a arquitetura apresentada tem as seguintes mais-valias:

- Os tweets são armazenados integralmente no seu formato original (JSON), não inviabilizando qualquer tipo de estudo posterior;
- Permite a análise de perfis e comportamentos online visto efetuar o ”seguimento” dos utilizadores ao recolher permanentemente a quase totalidade da sua atividade no Twitter;
- Com a continuidade da execução deste procedimento, os dados obtidos tenderão para a criação de uma base de dados bastante completa sobre determinada comunidade no Twitter, que neste caso é a comunidade portuguesa, mas que poderá ser qualquer outra devido à facilidade de parameterização do Sistema de Informação.

6.5 Utilização do Twitter em Portugal

No decorrer deste trabalho foram realizados dois estudos considerando o subconjunto de dados armazenado até à data de realização de cada uma das análises. Os resultados e conclusões que se observaram em ambos os estudos são apresentados nas duas Subsecções seguintes: a Subsecção 6.5.1 remete para uma análise preliminar da comunidade Portuguesa presente no Twitter (Brogueira et al., 2014a,b) e na Subsecção 6.5.2 é abordada uma breve caracterização dos Distritos Portugueses

[1] http://www.internetlivestats.com/twitter-statistics/

tendo em consideração o volume de tweets geolocalidados recolhidos ao longo de 2014 (Brogueira et al., 2015c).

6.5.1 Caracterização da Comunidade Portuguesa no Twitter

Considerando um período de oito dias consecutivos (14 a 21 de março de 2014) em que não foi interrompida a recolha de tweets geolocalizados, totalizando cerca de 307 mil tweets, produzidos por aproximadamente 11.391 utilizadores distintos, efetuou-se uma análise detalhada sobre o conteúdo das mensagens. Verificou-se uma predominância de autores jovens que usam o Twitter como uma forma de interagir com os seus seguidores, partilhando os seus sentimentos, ideias e comentários. O total de 307 mil de tweets recolhidos corresponde a uma média diária de cerca de 48 mil tweets. A Figura 6.17 representa o número de utilizadores que produziram Tweets num determinado intervalo. A maioria dos tweets são, de facto, produzida por utilizadores distintos. No entanto, mais de metade dos utilizadores não produziu mais do que um tweet durante o período em análise. O número de tweets varia consideravelmente por cada utilizador, entre 1 e cerca de 1100 tweets no período de 8 dias. Este comportamento justifica a abordagem descrita na Secção 4.4 para expandir a base de dados, ou seja, obter um conhecimento mais profundo de cada utilizador permitindo uma análise mais adequada dos traços de cada utilizador.

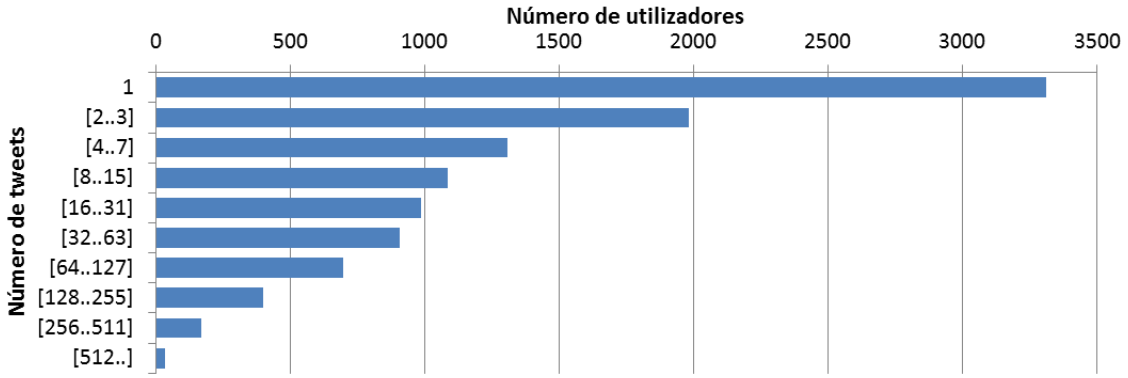


Figura 6.17: Conjunto de utilizadores que produziram certa quantidade de tweets ao longo de 8 dias consecutivos.

Sabe-se que o número de tweets produzidos não é linear no tempo. A Figura 6.18 apresenta a atividade por hora, descrevendo que durante as noites de sexta-feira o número de tweets é menor do que durante o resto das noites, sugerindo a utilização do Twitter de forma principalmente doméstica durante a noite, ou seja, os utilizadores normalmente não utilizam a rede social do Twitter quando saem com amigos, nas noites do fim de semana. Na verdade, os utilizadores são menos ativos durante o dia. No entanto, 43% de toda a atividade é realizada durante esse período o que representa uma proporção considerável. Levando isso em linha de conta, tentou-se caracterizar esta comunidade de utilizadores de outras perspetivas.

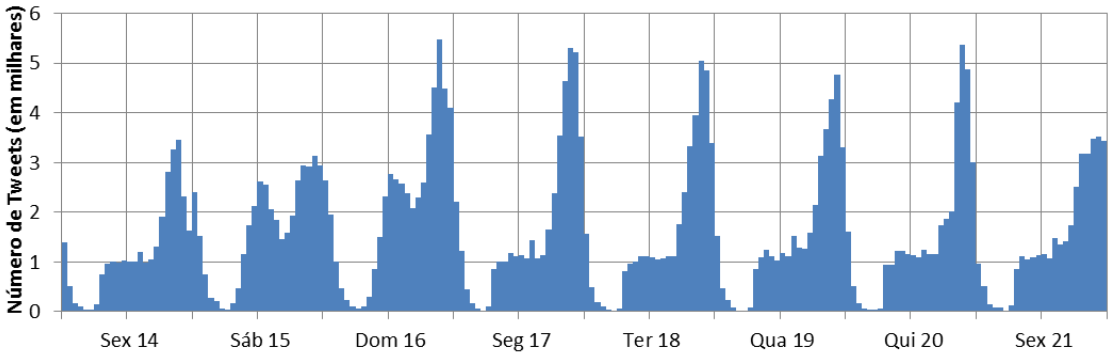


Figura 6.18: Distribuição da produção de tweets por hora.

O conteúdo da mensagem dos tweets sugere que estes foram produzidos principalmente por adolescentes. Caracterizar a comunidade envolvida em termos de idade, não é uma tarefa trivial, porque essa informação não está identificada de forma explícita dentro do conteúdo do tweet. Verifica-se que a descrição associada a cada utilizador, por vezes, contém informações que podem permitir a indução da idade aproximada do utilizador. A Figura 6.19apresenta alguns exemplos.

```
1          Paredes | 13 anos | Eminem
2          Ola tenho 14 anos e só sei dormir.
3          metro e meio de gente / 20 anos / ESTSP
```

Figura 6.19: Descrições partilhadas pelos utilizadores no seu perfil que permitem inferir a sua idade.

Para se ter uma ideia aproximada da idade dos utilizadores foram marcados manualmente cerca de 265 utilizadores com a sua idade potencial, com base na própria descrição. A informação resultante é mostrada na Figura 6.20, mostrando claramente uma predominância de adolescentes e jovens adultos, confirmando a análise realizada sobre o conteúdo dos tweets.

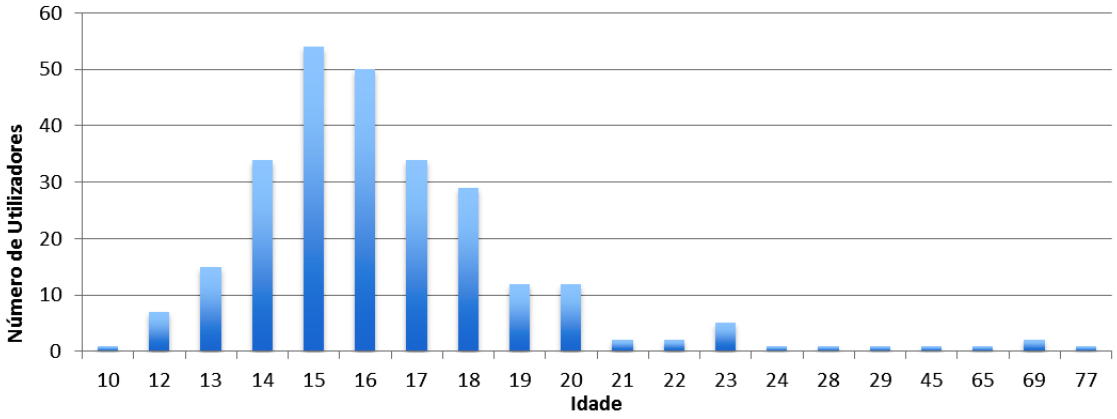


Figura 6.20: Distribuição da idade dos utilizadores.

Uma das particularidades mais interessantes do Twitter está relacionado com o limite do tamanho da mensagem em 140 caracteres, dando origem a que as mensagens contenham um reduzido número

Tabela 6.5: Trigramas mais frequentes no texto dos tweets.

Trigrama	Freq.	Trigrama	Freq.	Trigrama	Freq.
a minha mãe	1395	fim de semana	527	acho que vou	394
o meu pai	903	o que eu	459	que é que	386
sei o que	746	todos os dias	450	a dizer que	382
o que é	741	a minha vida	444	dia do pai	376
com a minha	704	que a minha	431	é que eu	373
tudo o que	634	como é que	428	ir para a	368
toda a gente	621	porque é que	424	como Assunto do	353
com o meu	617	que o meu	415	e a minha	350
A minha mãe	601	tenho de ir	412	o meu irmão	345

de palavras. Os dados revelam uma tendência para o aumento do número de palavras por mensagem ao longo do dia. O valor máximo é atingido em torno da hora de maior atividade no Twitter. Quanto ao conteúdo dos tweets, a Tabela 6.5 mostra os trigramas mais frequentes ocorridos nos 8 dias analisados. A frequência dos trigramas é um fator-chave para a compreensão da seleção lexical utilizada pelos autores dos tweets. Pode-se dizer que os tweets portugueses são na sua maioria incidentes sobre mensagens pessoais baseadas em laços familiares, como ilustrado na seleção de palavras a partir do mesmo campo semântico - “mãe“, “pai“, “irmão“, “irmã“. Além disso, há também um campo semântico associado com a escola, englobando vocabulário como “teste“, “A Turma minha“, “Aula“ sugerindo uma forte atividade de adolescentes ou jovens adultos.

Outra pista lexical para a caracterização da componente pessoal da mensagem publicada nos tweets é o uso de pronomes na primeira pessoa (sujeito “eu“, objeto “me“ e possessivos ”minha”, ”meu”). No que diz respeito às formas verbais, grande parte encontra-se na primeira pessoa, verificando-se uma seleção de verbos epistémicos (”sei”, ”acho”) o que proporciona um indicador crucial da forma como a comunidade do Twitter comunica as suas dúvidas e certezas, geralmente associando valores de apreciação ou antipatia. Expressões (“e que“) são outra estrutura muito frequente nas mensagens dos tweets, utilizadas para dar mais ênfase a uma frase. A partir do conjunto de opções léxico-sintáticas enunciadas é possível caracterizar os 8 dias de tweets, sendo que as mensagens partilham essencialmente dados muito pessoais, escritos na primeira pessoa, comunicando emoções. Em linha com a publicação de mensagens de cariz muito pessoal, está o recurso aos *emoticons*, representações pictóricas associadas a emoções distintas, que permitem a expressão de sentimentos em *e-contextos*. Apesar de alguns autores afirmarem que *emoticons* são usados principalmente por adolescentes e jovens adultos (Rao et al., 2014), atualmente o recurso aos *emoticons* é generalizado entre os grupos de todas as idades onde nenhuma conclusão pode ser tomada a partir deste facto.

Emoticons como :), :-), :3, ;), e :)) expressam alegria, enquanto *emoticons* como :(, :$, :/, :((e :-(expressam tristeza. O conjunto de *emoticons* encontrados nos tweets produzidos na semana em análise, são semelhantes aos relatados por Schnoebelen (2012) para tweets em inglês e estão praticamente todos incluídos no conjunto de *emoticons* utilizados por Ghiassi et al. (2013) para modelar a expressão e alteração de sentimentos por parte do consumidor sobre determinada marca. Os *emoticons* encontram-se ordenados segundo a sua frequência na Tabela 6.6.

Apenas um pequeno número de tweets incluem *hashtags* (cerca de 4,3%). Este é um número

Tabela 6.6: *Emoticons* mais frequentes.

Emoticon	**Frequência**	**Significado**
:)	2240	Sorriso
:c	1345	Atónito
:(	1255	Triste
:3	1017	Sorriso ou cara feliz
:-)	1016	Sorriso
:o	871	Surpreendido
;)	674	Piscar o olho
:D	526	Grande sorriso
(@	493	Ondulação
:p	351	Tirar a língua

Tabela 6.7: Hashtags mais frequentes ordenadas pelo número de utilizadores que as referem.

Hashtag	**Utilizadores**	**Freq.**	**Hashtag**	**Utilizadores**	**Freq.**
#lt	631	1555	#sun	79	97
#lrt	616	1403	#somosporto	69	171
#np	529	1882	#me	61	103
#twitteroff	238	439	#night	61	67
#portugal	185	377	#porto	61	105
#carregabenfica	175	617	#beach	56	66
#lisbon	125	249	#happy	56	68
#lisboa	110	231	#valetudo	55	89
#love	102	132	#sunset	54	57
#friends	94	132	#benfica	50	146
#selfie	90	98	#excluidadasociedade	50	58
#nw	83	116	#sad	48	57

bastante baixo quando comparado com outras recolhas de dados do Twitter. Weerkamp et al. (2011b) relata que cerca de 11% dos tweets Portugueses contêm *hashtags*, incluindo não só o Português Europeu como o Português do Brasil. Além disso, o Português é uma das línguas em que se usa menos *hashtags* das 8 línguas analisadas. Finalmente, numa base de dados semelhante com cerca de 1.5 milhões de tweets escritos em Português (incluindo todas as variedades de Português) recolhidos durante um período de tempo com igual número de dias, sem restringir o local e onde a maioria dos tweets são escritos em Português do Brasil, esse valor corresponde a 10,2%.

O top de *hashtags* mais frequentes são expressos na Tabela 6.7 e incluem #lt (último ou tweet mais recente), #lrt (último ou mais recente *retweet*), #NP ("*Now Playing*") usado sempre que alguém está a ouvir uma música e quer compartilhá-lo e #twitteroff (tweets suficientes por hoje). No entanto, observou-se que a frequência de tais *hashtags* foi relativamente baixa na base de dados de 1.5 milhões de tweets, mencionada no parágrafo anterior: #tl (posição 69), #lrt (posição 146), #NP (posição 109), #twitterof (posição 643).

Outro facto interessante diz respeito ao número de *retweets* (RT) que é muito baixo nos tweets recolhidos na semana em análise, o que corresponde a cerca de 0,1% dos tweets. No entanto, cerca de 26% dos tweets dos 1.5 milhões os tweets mencionados anteriormente são *retweets*. Duas possibilidades podem estar correlacionadas com este facto. A primeira possibilidade é que os tweets geolocalizados não costumam incluir *retweets*. Outra hipótese está relacionada com o facto dos tweets recolhidos serem na sua maioria de cariz pessoal e por isso não são normalmente reenviados e partilhados por

outros utilizadores.

Com este estudo foi possível determinar algumas das características da comunidade portuguesa do Twitter, nomeadamente a faixa etária dos utilizadores, o conteúdo das mensagens que partilham e a forma como utilizam o Twitter.

6.5.2 Caracterização dos Distritos Portugueses

Um outro estudo foi efetuado tendo por base a informação capturada relativamente aos tweets geolocalizados com o objetivo de obter uma análise dos distritos de Portugal. Para tal análise foi considerado o volume de tweets produzidos em cada um dos distritos, no período de fevereiro a dezembro de 2014 num total aproximado de 18.4 milhões de tweets geolocalizados. Por questões de simplificação foram considerados apenas os distritos de Portugal continental, que se encontra dividido em 18 distritos: Aveiro, Beja, Braga, Bragança, Castelo Branco, Coimbra, Évora, Faro, Guarda, Leiria, Lisboa, Portalegre, Porto, Santarém, Setúbal, Viana do Castelo, Vila Real, Viseu. Para a análise de cada uma das regiões pressupõe-se a aferição do distrito em que determinado tweet foi publicado. No entanto, o nome do distrito na grande maioria dos casos não se encontra explicitamente associado à informação da localização do tweet, nomeadamente nos campos *place* e *geo,* embora a informação contida nestes campos possa permitir inferir o distrito correspondente à localidade onde o tweet foi produzido.

6.5.2.1 Inferir o nome do Distrito através do nome da Localidade

Foram testadas diversas abordagens para obter o distrito correspondente a cada localidade. Por vezes no campo *place.full_name* é indicado o nome da localidade seguido do nome do distrito correspondente, tal como se pode verificar pelos exemplos da Figura 6.21. No primeiro exemplo, sendo o valor do campo “Lisboa, Lisboa“ a primeira referência a “Lisboa” corresponde ao nome da localidade Lisboa e a segunda referência a “Lisboa” corresponde ao nome do distrito a que a localidade de Lisboa pertence, que neste caso é ao distrito de Lisboa. Nestes casos o problema teria uma solução trivial, no entanto, em apenas 8.37% dos tweets o nome do distrito estava associado ao nome da localidade.

```
1 { _id : ___, place : { full_name : Lisboa, Lisboa } }
2 { _id : ___, place : { full_name : Gondomar, Porto } }
3 { _id : ___, place : { full_name : Tavira, Faro } }
4 { _id : ___, place : { full_name : Faro, Faro } }
5 { _id : ___, place : { full_name : Vila Nova de Gaia, Porto } }
6 { _id : ___, place : { full_name : Setúbal, Setúbal } }
7 { _id : ___, place : { full_name : Alcacér do Sal, Setúbal } }
```

Figura 6.21: Tweets em que o campo *place.full_name* além do nome da localidade contém o nome do respetivo distrito.

Na Figura 6.22 observam-se exemplos de tweets cujo campo *place.full_name* não contém qualquer informação que indique explicitamente o nome do distrito, tendo somente a indicação do nome da localidade.

```
{ _id : ___, place : { full_name : Vila Real } }
{ _id : ___, place : { full_name : Sintra } }
{ _id : ___, place : { full_name : Vila Viçosa } }
```

Figura 6.22: Exemplo de tweets em que o campo *full_name* não contém explicitamente o nome do distrito.

Dado o reduzido número de tweets que contêm o nome do distrito no campo *place.full_name,* foi testada a possibilidade de inferir o nome do distrito pelas coordenadas geográficas presentes no campo *geo*. No entanto, também através deste campo não é possível obter a informação pretendida para a quase totalidade dos tweets, uma vez que neste caso apenas cerca de 66.6% dos tweets analisados têm o campo *geo* devidamente preenchido.

Portanto, foi considerada uma abordagem alternativa. A determinação do nome do distrito correspondente à localidade referida no campo *place.name* é possível pela consulta da ligação entre ambos na lista de códigos postais disponibilizada pelos CTT - Correios de Portugal S. A.[2]. Nesta lista, atualizada todos os dias de madrugada, constam além de outras informações a associação entre localidades e respetivos distritos, para todas as localidades do território continental português e para os arquipélagos dos Açores e da Madeira.

A lista de códigos postais disponibilizada pelos CTT contém diversos ficheiros CSV, num dos quais é efetuado o mapeamento entre determinada localidade e o respetivo distrito. Cada linha deste ficheiro é composta por 17 campos correspondentes a diferentes tipos de dados, separados por ponto e vírgula, contendo entre outros os seguintes valores: código do distrito, código do concelho, código da localidade, nome da localidade, número do código postal e extensão do código postal, tal como é exemplificado da Figura 6.23.

```
01;04;69893;Picoto;;;;;;;;;;;4540;205;AROUCA
02;11;63284;Cabanas;;;;;;;;;;;7630;039;ODEMIRA
14;12;18815;Golegã;1251214;Rua;de;São;;Martinho;;;;;2150;153;GOLEGÃ
```

Figura 6.23: Informação relativa a cada localidade.

O valor 01 da Figura 6.23 corresponde ao código distrito (Aveiro), o valor 04 é o código do concelho (Arouca), 69893 o código da localidade (Picoto), 4540 o valor do código Postal, 205 corresponde à extensão do código postal e Arouca é a designação postal. O mapeamento entre o código do distrito e a respetiva designação é disponibilizado num outro ficheiro CSV, cujo conteúdo se apresenta na Figura 6.24[3].

[2]Foi efetuado o download da informação relativa às Localidades, Concelhos e Distritos a partir do endereço https://www.ctt.pt/feapl_2/app/restricted/postalCodeSearch/postalCodeDownloadFiles.jspx em 21-01-2015.

[3]As localidades e respetivos distritos dos Arquipélagos dos Açores e da Madeira estão tamém incluídos no ficheiro CSV dos CTT, no entanto por questões de simplicidade não foram incluídos neste estudo, pelo que não se encontram referenciados na Figura 6.24

```
01;Aveiro
02;Beja
03;Braga
04;Bragança
05;Castelo Branco
06;Coimbra
07;Évora
08;Faro
09;Guarda
10;Leiria
11;Lisboa
12;Portalegre
13;Porto
14;Santarém
15;Setúbal
16;Viana do Castelo
17;Vila Real
18;Viseu
```

Figura 6.24: Mapeamento entre o código do distrito e o respetivo nome.

Desta forma foi possível obter o distrito para 99.69% dos tweets. Além dos mapeamentos efetuados por via da pesquisa na base de dados dos CTT, foram necessárias algumas correções de forma manual, ou seja, para os casos em que o campo *place.name* tem o nome da localidade escrito numa língua diferente de Português (p.ex.: "Lisbona" ou "Oporto"), ou o nome da localidade apresenta algum erro ortográfico (p.ex.: "Sétubal" ou "Guimaraes") foi definido um dicionário que relaciona os nomes das localidades com erro ou em língua diferente de Português, com o respetivo nome do distrito.

Dos 0.31% de tweets em que o campo *place.name* contém um valor do qual não é possível inferir o distrito, a sua maioria apresenta somente a palavra "Portugal" através da qual é impossível obter o distrito onde o tweet foi publicado. Uma quantidade quase insignificante de tweets contém no campo *place.name* valores que não correspondem ao nome de nenhuma localidade portuguesa referindo-se, por exemplo, a estabelecimentos comerciais tal como é o caso de "Restaurante Zé Pinto" ou "Casa dos Presuntos".

Existem diversos casos de localidades portuguesas com nomes iguais pertencentes a distritos diferentes. Um dos exemplos é a "Covilhã", cidade do distrito de Castelo Branco, cujo nome é igualmente associado a uma localidade no distrito do Porto e outra localidade no distrito de Braga. Outro exemplo é o nome da localidade "Seixal", que sendo uma cidade do distrito de Setúbal, tem uma localidade homónima no distrito de Aveiro. Ambos os exemplos são apresentados na Figura 6.25. Para desambiguação destes casos foi considerada a área de cada localidade, sendo o nome da localidade com maior área associado ao respetivo distrito, ou seja, assumiu-se como muito pouco provável a produção de tweets nos locais designados por Covilhã nos distritos do Porto e Braga, tendo sido todos os tweets que indicam terem sido produzidos na Covilhã associados ao distrito de Castelo Branco.

6.5.2.2 Estatísticas por Distrito e por Regiões

Considerando a divisão de Portugal continental em 18 distritos e através da contagem do número de tweets produzidos em cada distrito, obtiveram-se diversas métricas relativas à variação da atividade no Twitter em cada distrito. O mapa esquerdo da Figura 6.26 mostra a distribuição de utilizadores por

```
03;13;53346;Covilhã;;;;;;;;;;;4730;490;SANTIAGO CARREIRAS
05;03;14718;Covilhã;1030305;Rua;do;;;Botoreu;;;;;6200;058;COVILHÃ
13;01;4000;Covilhã;;;;;;;;;;;4600;757;TELÕES AMT

01;04;60744;Seixal;;;;;;;;;;;4540;497;ROSSAS ARC
15;10;43887;Seixal;200101015;Rua;;;;Silvana Alves Cunha;;;;;2840;471;SEIXAL
```

Figura 6.25: Localidades com o mesmo nome em diferentes distritos.

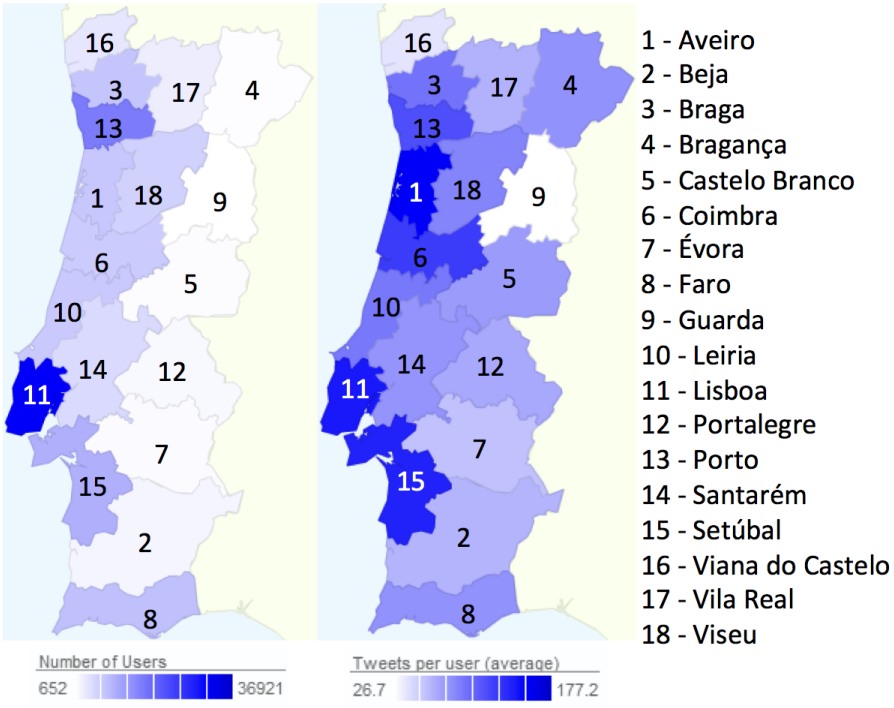


Figura 6.26: Distribuição do volume de tweets e respetivos autores em cada distrito.

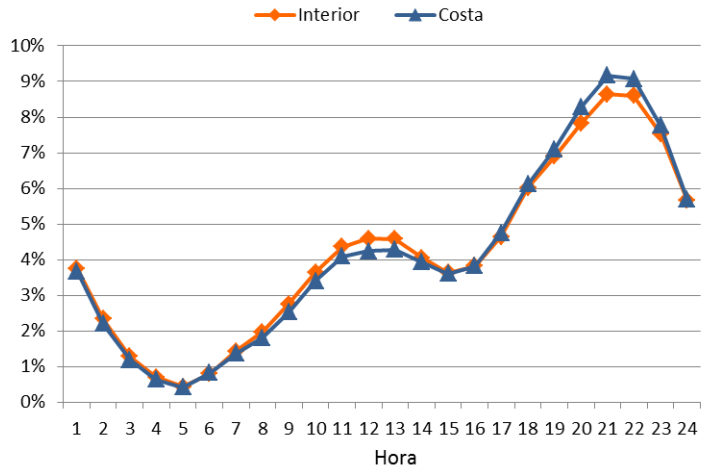


Figura 6.27: Atividade durante o dia nos distritos da costa e do interior de Portugal.

cada um dos distritos, verificando-se um maior volume de utilizadores na costa de Portugal, nomeadamente em Lisboa (37 mil), Porto (20 mil) e Faro (10 mil) com valores aproximados. Faro tem um valor elevado de utilizadores, devido ao afluxo de população no período das férias de verão. O mapa da direita na Figura 6.26 relaciona o número de utilizadores com a sua atividade no Twitter, verificando-se que os utilizadores mais ativos concentram-se nos distritos de Aveiro, Lisboa e Setúbal.

De acordo com os ”Censos 2011” (Estatística, 2011) a distribuição da população Portuguesa não é equalitária em todos os distritos do território Português. De facto, verifica-se uma elevada desertificação das regiões do interior de Portugal, em contraste com as regiões metropolitanas de Lisboa e do Porto. Nos ”Censos 2011” é igualmente referido que a distribuição da população jovem e idosa de Portugal contém algumas assimetrias, verificando-se nas regiões do interior uma predominância de população idosa e o cenário oposto nos grandes centros urbanos do litoral de Portugal. No seguimento da análise efetuada na Secção 6.5, onde se concluiu que a comunidade portuguesa no Twitter é composta essencialmente por adolescentes ou jovens adultos, tal ocorrência pode estar relacionada com a observação de uma maior atividade dos utilizadores nos distritos do litoral de Portugal, nomeadamente em Lisboa (~6.1 milhões de tweets), seguido do Porto (~2.5 milhões) a Setúbal (~1.9 milhões), que é consistente com a distribuição da população Portuguesa (Estatística, 2011).

Ainda relativamente aos ”Censos 2011” é referido neste estudo, que os distritos do litoral contêm uma maior percentagem de população ativa, com um horário de trabalho superior a 45 horas semanais, particularmente nas regiões metropolitanas de Lisboa e do Porto, o que pode estar correlacionado com a menor atividade no Twitter durante o dia nos distritos do litoral embora com um ligeiro crescimento na hora de almoço (12h-14h) nos distritos do interior. Para o final do dia, ocorre o inverso e é nos distritos do litoral onde se verifica maior atividade, principalmente entre as 18h e as 24h como se confirma pela análise da Figura 6.27 A principal atividade económica do interior de Portugal está ligada ao setor primário, devido ao menor desenvolvimento desta região do país o que poderá estar relacionado com o facto da população desta região inicie o período de repouso diário um pouco mais cedo, havendo portanto menor atividade no Twitter no período da noite nesta região. Outra possível explicação para as diferenças de atividade no Twitter entre as regiões do litoral e interior do país, está relacionado com as diferenças de hábitos entre as grandes e as pequenas cidades. As pessoas que vivem nas grandes cidades têm maior oferta de atividades noturnas e porque as grandes cidades estão essencialmente

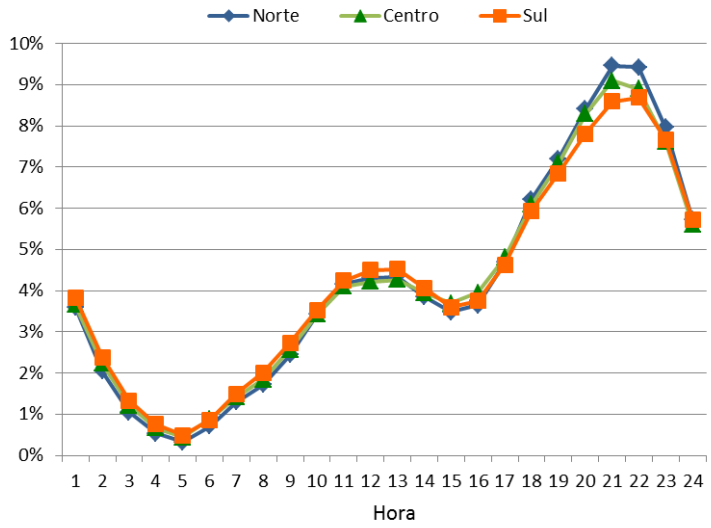


Figura 6.28: Atividade diária nos distritos do Norte, Centro e Sul.

localizadas no litoral de Portugal, verifica-se que a atividade no Twitter nestas zonas é mais proeminente durante a noite.

Outro agrupamento dos distritos de Portugal que é comum observar consiste na divisão em Norte, Centro e Sul. A região Norte inclui os distritos de Aveiro, Braga, Bragança, Guarda, Porto, Viana do Castelo, Vila Real, e Viseu; a região Centro contém os distritos de Castelo Branco, Coimbra, Leiria, Lisboa, Portalegre e Santarém; e a região Sul contém os distritos de Beja, Évora, Faro, e Setúbal. O gráfico da Figura 6.28 mostra a utilização do Twitter em cada hora do dia nestas três regiões, revelando que a produção de tweets no período da noite (18h às 24h) é superior na região norte, enquanto que a região sul tem menor atividade durante o mesmo período, em contraste com o período a meio do dia onde a região sul tem uma atividade mais elevada na publicação de tweets. Este tipo de informação poderá revelar-se bastante útil para o lançamento de campanhas publicitárias ou para a propagação de notícias, direcionado para os interesses da população em cada região. Ao conhecer a hora do dia em que a audiência alvo numa determinada região é mais ativa no Twitter a informação pode ser propagada de forma mais eficaz e visualizada por um maior número de potenciais clientes.

O seguinte conjunto de análises consideram o período do ano em que os tweets foram produzidos e relacionam a atividade dos utilizadores com os períodos de trabalho e períodos de férias. Os dados foram divididos em quatro períodos de trabalho e de férias, considerando como períodos de trabalho os intervalos entre 20 de fevereiro e 14 de junho e de 15 de setembro a 19 de dezembro. O período de férias de verão foi considerado a partir de 15 de junho a 14 de setembro e o período de férias de Natal e Ano Novo de 20 de dezembro a 31 de dezembro. A Figura 6.29 resume a atividade nesses períodos, revelando o generalizado aumento da atividade durante as férias.

O período de férias de verão é especialmente ativo em Faro, local eleito por um número considerável de portugueses para gozar as suas férias, devido às boas praias características desta região e ao bom tempo durante os meses de verão. Os distritos de Bragança e Guarda são também particularmente ativos durante este período, principalmente por causa dos muitos imigrantes que regressam a Portugal nesta época do ano para se reunir com suas famílias. O período de férias de Natal é o mais ativo na maioria das regiões, sendo particularmente ativo em Portalegre, Castelo Branco, Porto e

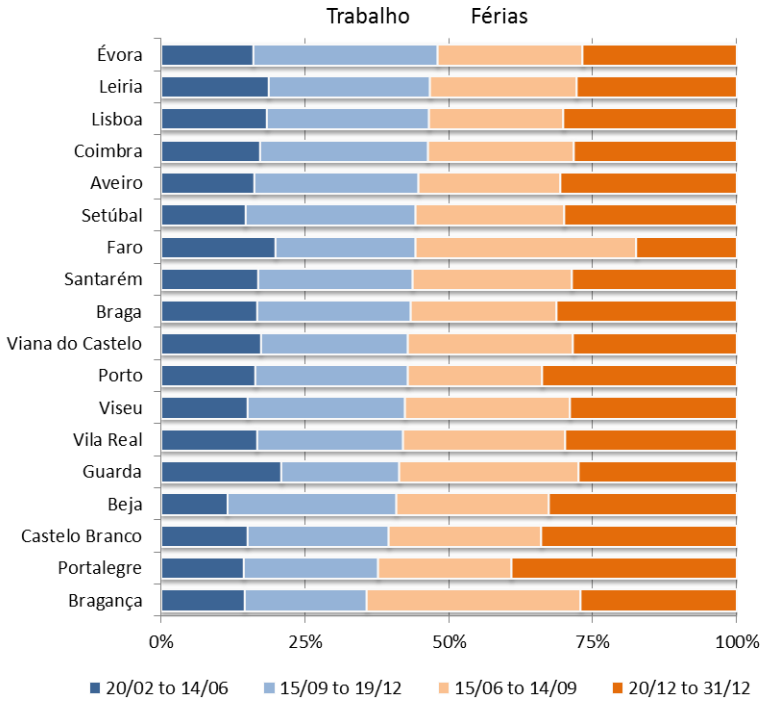


Figura 6.29: Atividade por dia em períodos de trabalho e períodos de férias.

Beja. A única exceção é o distrito de Faro que em contraste com o período de férias de verão é aquele com menor produção de tweets durante as férias de Natal. Finalmente, a Figura 6.29 mostra que o período de trabalho mais ativo vai de setembro a dezembro, o qual corresponde ao início da escola. Este comportamento poderá estar correlacionado com o facto da maioria dos utilizadores identificados e integrados na base de dados serem maioritariamente jovens, que usam o Twitter como uma forma de ocupação e troca de mensagens com os colegas de escola.

A última análise refere-se à atividade diária nos diferentes períodos de trabalho e de férias, sendo ilustrada nas Figuras 6.30 e 6.31. A Figura 6.30 mostra a atividade diária correspondente a períodos de trabalho, enquanto que a Figura 6.31 mostra a atividade correspondente nos períodos de férias. Durante os períodos de trabalho a distribuição da atividade é bastante semelhante em ambos os períodos. Os picos de atividade são em torno de 21h e entre 11h e as 13h. A menor atividade corresponde ao intervalo entre as 4h e 5h. No que diz respeito aos períodos de férias também seguem um formato semelhante, mas as diferenças são mais notórias. Durante o dia o pico de atividade é antes do meio-dia durante o verão e depois do meio-dia durante a época do Natal. A atividade sobe durante a noite e atinge o seu limite inferior em torno das 6h da madrugada.

As Figuras 6.30 e 6.31 revelam que os períodos de trabalho e períodos de férias são de facto bastante distintos em termos de atividade dos utilizadores. Uma das diferenças mais notórias é a atividade prolongada durante a noite em períodos de férias, em oposição ao auge da atividade que pode ser encontrado em torno das 21h durante os períodos de trabalho.

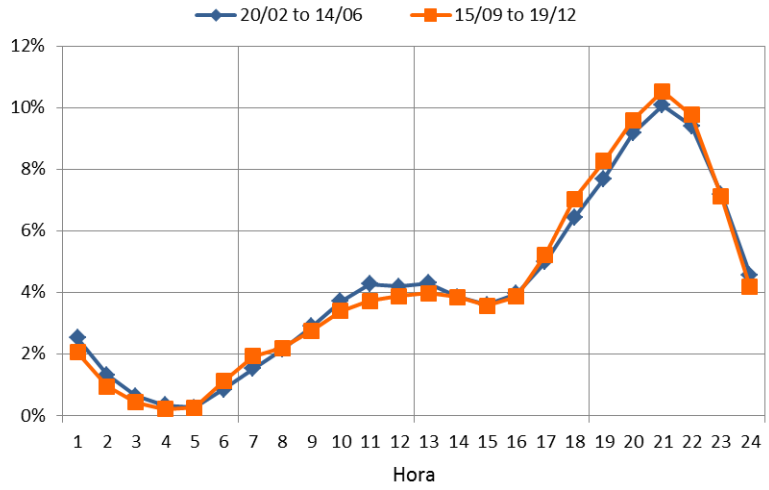


Figura 6.30: Atividade por hora nos períodos de trabalho.

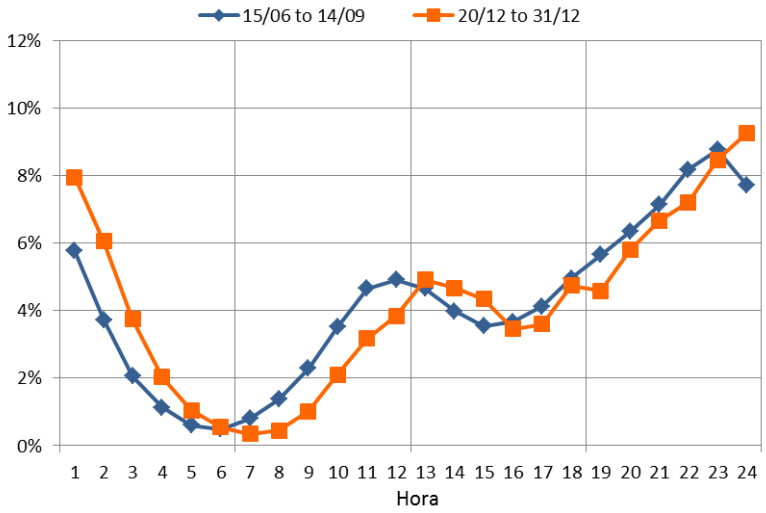


Figura 6.31: Atividade por hora nos períodos de férias.

6.6 Sumário

Este Capítulo resume e analisa as principais estatísticas obtidas sobre os dados recolhidos em resultado do procedimento apresentado neste trabalho. São igualmente apresentados dois estudos preliminares baseados em subconjuntos de dados do *corpus* obtidos em momentos distintos ao longo do trabalho. Efetua-se também uma breve discussão dos dados recolhidos por comparação com o volume de dados obtidos por trabalhos de outros autores, demonstrando que a arquitetura proposta permite a recolha de uma quantidade assinalável de informação sobre os utilizadores portugueses do Twitter.

7 Conclusões

> Facebook wants to know "What's on your mind?" Twitter asks "What's happening?" But that's getting old already. The burning question for the next wave of social networking is "Where are you?"
>
> *Daniel Ionescu, PCWorld*

Dada a relevância que o conteúdo das mensagens trocadas nas redes sociais pode assumir em diversas áreas o *corpus* resultante deste trabalho pode permitir o desenvolvimento de diversos estudos sociológicos, socioeconómicos, análise de tendências, previsões de resultados eleitorais ou a análise de sentimentos focada em marcas, produtos ou serviços. Nos estudos baseados em tweets escritos em Português tem sido utilizado um conjunto reduzido de tweets não ultrapassando os milhares, o que porventura poderá condicionar alguns tipos de análises. Com o presente estudo, apresenta-se uma arquitetura para recolha, armazenamento, partilha e processamento eficiente de um volume considerável de mensagens publicadas no Twitter. O processo apresentado obtém os tweets geolocalizados produzidos em tempo real e disponibilizados na Twitter Streaming API. Para os utilizadores cujos tweets foram identificados como tendo sido produzidos em Portugal e escritos em português europeu, é lida a respetiva timeline possibilitando em dado momento recolher para cada utilizador as suas últimas 3200 publicações. Desta forma efetua-se uma expansão da base dados por recuperação de tweets produzidos no passado, tendo em conta informação que é recolhida no presente e em tempo real. O método proposto para recuperar os tweets adicionais produzidos pelo conjunto de autores que partilham a sua geolocalização tem como objetivo ampliar o conhecimento sobre cada utilizador desta comunidade, tornando-se possível caracterizar cada utilizador automaticamente pelo conteúdo que publica, os grupos da comunidade e abrir outras possibilidades no âmbito da análise social.

O período de trabalho desta tese decorreu entre janeiro de 2014 e agosto de 2015, tendo a recolha de tweets ocorrido entre fevereiro de 2014 e maio de 2015. Neste período foram armazenados certa de 30 milhões de tweets geolocalizados produzidos por aproximadamente 105 mil utilizadores distintos, recolhendo-se em média 64.5 mil tweets geolocalizados por dia. Da leitura da timeline de todos os utilizadores, excetuando os que bloquearam o acesso à respetiva timeline, recuperaram-se aproximadamente 242 milhões de tweets adicionais, constituindo-se *um corpus* com um total aproximado de 272 milhões de tweets, resultando num incremento de 8 vezes relativamente ao número de tweets obtidos do fluxo da Streaming API. Para cada um dos 105 mil utilizadores foi possível obter em média 2653 tweets do seu histórico de mensagens, com a garantia de terem sido produzidos em Portugal e escritos em português europeu, representando um valor bastante significativo permitindo efetuar uma análise de perfis e estudar o comportamento online da comunidade portuguesa do Twitter.

Os *media* sociais desempenham um papel crucial no quotidiano dos adolescentes em rede, proporcionando um espaço onde podem passar o tempo e interagir com os amigos. As interações entre adolescentes mediadas pelas redes sociais complementam ou suplementam, em muitas circunstâncias, os encontros face a face. Verificou-se que a comunidade portuguesa do Twitter é predominantemente constituída por jovens adolescentes, utilizando esta rede social para partilhar e falar sobre aspetos do seu dia a dia, como por exemplo, assuntos relacionados com a escola.

As tecnologias e os conceitos abordados nesta tese têm um âmbito bastante abrangente, focando aspetos relacionados com diversos domínios tais como Big Data, NoSQL, Web APIs e Data Visualization (visualização de dados). A participação em diversos workshops, conferências, *Summer Schools* (Universidades de verão) e a realização de alguns MOOCs (*Massive Open Online Courses*) em muito contribuíram para a familiarização com algumas das tecnologias utilizadas no decorrer deste trabalho. Seguem-se alguns dos eventos mais relevantes nos quais se participou:

- 2º DataStorm Big Data Summer School, julho de 2015, Instituto Superior Técnico, Lisboa
- SQLSaturday, março de 2015, Microsoft Portugal, Lisboa
- Oracle Day 2014, Digital Disruption - Big Data & Analytics, novembro de 2014, Lisboa
- 4ª Lisbon Machine Learning School 2014 - Learning with Big Data, julho de 2014, Instituto Superior Técnico, Lisboa
- 1º DataStorm Big Data Summer School 2014, julho de 2014, Instituto Superior Técnico, Lisboa
- *Massive Open Online Course*s:
  - MongoDB for Python Developers, MongoDB University, março de 2015
  - Introduction to Computer Science and Programming Using Python, edX, dezembro de 2014

7.1 Trabalho Futuro

As propostas sugeridas para a continuação deste trabalho, enquadram-se em três domínios:

1. **Software utilizado:**

   (a) As dificuldades na gestão da informação armazenada no MongoDB 2.4, podem de alguma forma ser minimizadas pela utilização da mais recente versão do MongoDB, que à data da escrita desta tese está na versão 3.0.4. São de realçar dois aspetos integrados na versão 3.0 do MongoDB que decerto serão uma mais-valia em problemas idênticos aos apresentados nesta tese: i) melhorias significativas ao nível do desempenho e escalabilidade, nomeadamente na camada de armazenamento onde a compressão em disco reduz a necessidade de espaço de armazenamento até 80%; ii) incremento na performance de escrita, sendo 7 a 10 vezes mais rápido. Outra funcionalidade do MongoDB que não foi explorada e que poderá permitir melhorar o desempenho do Sistema de Informação, relaciona-se com a utilização do particionamento dos dados em blocos (*sharding*[1]) possibilitando o seu processamento de forma distribuída;

[1] *Sharding* é um processo de armazenamento de dados de forma distribuída em diversas máquinas. Quando o volume de dados aumenta, uma única máquina pode não ser suficiente para armazenar todos os dados, nem proporcionar uma performance de escrita e leitura aceitáveis, pelo que o *sharding* resolve o problema com a escalonamento horizontal.

(b) Na sequência do disposto na Secção 5.2, a biblioteca Bokeh para geração de gráficos dinâmicos, proporciona uma maior interatividade aos gráficos quando comparado com a biblioteca de gráficos Google Charts. A sua utilização permitiria uma maior uniformização de todo o Sistema de Informação, dado que deste modo ficaria baseado somente na linguagem Python e independente das bibliotecas online do Google Charts.

2. **Abrangência da base de dados:**

   (a) No sentido de aumentar o conjunto de utilizadores portugueses sobre os quais é recolhido o seu histórico de publicações, de modo a possibilitar o estudo de um maior número de perfis ou efetuar a análise de determinado evento com informação sobre um conjunto mais alargado de utilizadores, poderá ser considerada a inclusão dos utilizadores mencionados nos tweets recolhidos ressalvando as devidas confirmações para aferir se são realmente utilizadores portugueses;

   (b) Também com o objetivo de aumentar o conjunto de utilizadores ”seguidos” pelo Sistema de Informação, poderá optar-se pela inclusão dos *followers* e *friends* dos utilizadores portugueses considerados pela arquitetura apresentada, considerando-se apenas os que no seu histórico contenham tweets geolocalizados nas condições já referidas;

   (c) No processo de utilização do *corpus* no âmbito do algoritmo de deteção de tópicos de Rosa et al. (2014) apresentado na Subsecção 5.1.3, foi identificada a necessidade de recolher o histórico de mensagens dos utilizadores relacionados com o evento em análise mas que não constam da base de dados de utilizadores. Esta poderá ser uma nova *feature* a incluir futuramente no Sistema de Informação;

   (d) Do conjunto de informações associadas aos utilizadores não consta a sua idade nem o género. Tendo em vista a inclusão deste tipo de dados sobre cada um dos utilizadores, poderá ser utilizado o método desenvolvido por Vicente et al. (2015) através da implementação de um *endpoint* na REST API que permita a realização de tal operação.

3. **Desempenho do processo de expansão da base de dados de tweets:**

   (a) Embora seja utilizada uma quantidade assinalável de clientes no processo de atualização das timelines, verifica-se que determinados utilizadores têm uma atividade de publicação de tal modo elevada que entre duas iterações consecutivas deste processo o volume de mensagens é superior a 3200. Deste modo poderá considerar-se definir um conjunto de utilizadores mais ativos e, por conseguinte, associar alguns clientes em exclusivo ao processamento deste conjunto de utilizadores tentando minimizar o intervalo de tempo entre duas iterações consecutivas para atualização das respetivas timelines. Esta situação foi identificada na Secção 6.2.

Referências Bibliográficas

Abel, F., Hauff, C., Houben, G.-J., Stronkman, R., and Tao, K. (2012). Twitcident: Fighting fire with information from social web streams. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, pages 305–308, New York, NY, USA. ACM.

Abramova, V. and Bernardino, J. (2013). Nosql databases: Mongodb vs cassandra. In *Proceedings of the International C* Conference on Computer Science and Software Engineering*, C3S2E '13, pages 14–22, New York, NY, USA. ACM.

Alexa (2014). How popular is twitter.com? `http://www.alexa.com/siteinfo/twitter.com`. [Online; acedido em 15-Julho-2015].

Anderson, K. M. and Schram, A. (2011). Design and implementation of a data analytics infrastructure in support of crisis informatics research. In Taylor, R. N., Gall, H., and Medvidovic, N., editors, *ICSE*, pages 844–847. ACM.

Arpaci-Dusseau, R. and Arpaci-Dusseau, A. (2012). *Operating Systems: Three Easy Pieces*.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *Int. J. Semantic Web Information Systems*, 5(3).

Borkar, V. R., Carey, M. J., and Li, C. (2012). Big data platforms: What's next? *XRDS*, 19(1):44–49.

Bošnjak, M., Oliveira, E., Martins, J., Mendes Rodrigues, E., and Sarmento, L. (2012). TwitterEcho - A Distributed Focused Crawler to Support Open Research with Twitter Data. *Proceedings of the WWW 2012, the 21st International Conference Companion on World Wide Web*, pages 1233–1239.

Boyd, D. (2014). *It's Complicated: The Social Lives of Networked Teens*. Yale University Press.

Boyd, D., Golder, S., and Lotan, G. (2010). Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10.

Brewer, E. A. (2000). Towards robust distributed systems (abstract). In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '00, pages 7–, New York, NY, USA. ACM.

Brogueira, G., Batista, F., and Carvalho, J. P. (2015a). Arquitetura e desenvolvimento de um repositóorio de tweets em português europeu. In *5as Jornadas de Informática da Universidade de Évora - JIUE 2015*. Springer.

Brogueira, G., Batista, F., and Carvalho, J. P. (2015b). Sistema inteligente de recolha, armazenamento e visualização de informação proveniente do twitter. In *15ªConferência da Associação Portuguesa de Sistemas de Informação*, CAPSI 2015.

Brogueira, G., Batista, F., and Carvalho, J. P. (2015c). Using geolocated tweets for characterization of portuguese administrative regions. In *18th AGILE International Conference on Geographic Information Science*.

Brogueira, G., Batista, F., Carvalho, J. P., and Moniz, H. (2014a). Expanding a database of portuguese tweets. In *SLATE'14 3rd Symposium on Languages, Applications and Technologies*, volume 4569 of *OpenAccess Series in Informatics (OASIcs)*, pages 275–282. Schloss Dagstuhl.

Brogueira, G., Batista, F., Carvalho, J. P., and Moniz, H. (2014b). Portuguese geolocated tweets: an overview. In *ISDOC2014 - Proceedings of the International Conference on Information Systems and Design of Communication*, pages 178–179. ACM.

Buneman, P., Fernandez, M., and Suciu, D. (2000). Unql: A query language and algebra for semistructured data based on structural recursion. *The VLDB Journal*, 9(1):76–110.

Caldeira, J. (2010). *Dashboards - Comunicar Eficazmente a Informação de Gestão*. Almedina.

Carvalho, J. P., Pedro, V. C., and Batista, F. (2013). Towards intelligent mining of public social networks' influence in society. *Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting, IFSA/NAFIPS 2013*, pages 478–483.

Castells, M. (2000). *The Rise of The Network Society: The Information Age: Economy, Society and Culture*. Number vol. 1 in Information Age Series. Wiley.

Cattell, R. (2011). Scalable sql and nosql data stores. *SIGMOD Rec.*, 39(4):12–27.

Celko, J. (2013). *Joe Celko's Complete Guide to NoSQL*. Elsevier (Morgan Kaufmann), Amsterdam.

Cha, M., Haddai, H., Benevenuto, F., and Gummadi, K. P. (2010). Measuring User Influence in Twitter : The Million Follower Fallacy. *International AAAI Conference on Weblogs and Social Media*, pages 10–17.

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. E. (2006). Bigtable: A distributed storage system for structured data. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7*, OSDI '06, pages 15–15, Berkeley, CA, USA. USENIX Association.

Chasseur, C., Li, Y., and Patel, J. M. (2013). Enabling json document stores in relational systems. In Bonifati, A. and Yu, C., editors, *WebDB*, pages 1–6.

Chew, C. and Eysenbach, G. (2010). Pandemics in the age of twitter: Content analysis of tweets during the 2009 h1n1 outbreak. *PLoS ONE*, 5(11):e14118.

Claster, W., Cooper, M., and Sallis, P. (2010). Thailand – tourism and conflict: Modeling sentiment from twitter tweets using naïve bayes and unsupervised artificial neural nets. In *Computational Intelligence, Modelling and Simulation (CIMSiM), 2010 Second International Conference on*, pages 89–94.

Crockford, D. (2006). The application/json media type for javascript object notation (json). `https://tools.ietf.org/html/rfc4627`. [Online; acedido em 16-Julho-2015].

Culotta, A. (2010). Towards detecting influenza epidemics by analyzing twitter messages. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 115–122, New York, NY, USA. ACM.

Dehkharghani, R., Mercan, H., Javeed, A., and Saygin, Y. (2014). Sentimental causal rule discovery from twitter. *Expert Systems with Applications*, 41(10):4950 – 4958.

Demchenko, Y., Grosso, P., De Laat, C., and Membrey, P. (2013). Addressing big data issues in scientific data infrastructure. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 48–55.

Dharmasiri, H. and Goonetillake, M. (2013). A federated approach on heterogeneous nosql data stores. In *Advances in ICT for Emerging Regions (ICTer), 2013 International Conference on*, pages 234–239.

D.M. Boyd, N. E. (2007). Social network sites: definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230.

Duarte, E. S. (2013). Sentiment analysis on twitter for the portuguese language. Master's thesis, Faculdade de Ciencias da Universidade Nova de Lisboa, Lisboa.

Eriksson, B., Barford, P., Sommers, J., and Nowak, R. (2010). A learning-based approach for ip geolocation. 6032:171–180.

Estatística, I. N. (2011). Censos 2011 resultados definitivos - portugal. `http://censos.ine.pt/xportal/xmain?xpgid=censos2011_apresentacao&xpid=CENSOS`. [Online; acedido em 21-Julho-2015].

Fabio Pianese, Xueli An, F. K. and Ishizuka, H. (2013). Discovering and predicting user routines by differential analysis of social network traces. *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 1–9.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis. AAI9980887.

Framingham, M. (2015). Android and ios squeeze the competition.

Gerber, M. S. (2014). Predicting crime using twitter and kernel density estimation. *Decision Support Systems*, 61(0):115 – 125.

Ghiassi, M., Skinner, J., and Zimbra, D. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications*, 40(16):6266–6282.

Gilbert, S. and Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59.

Goonetilleke, O., Sellis, T., Zhang, X., and Sathe, S. (2014). Twitter analytics: A big data management perspective. *SIGKDD Explor. Newsl.*, 16(1):11–20.

Grinberg, M. (2014). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc., 1st edition.

Gruber, D. A., Smerek, R. E., Thomas-Hunt, M. C., and James, E. H. (2015). The real-time power of twitter: Crisis management and leadership in an age of social media. *Business Horizons*, 58(2):163–172.

Han, J., Haihong, E., Le, G., and Du, J. (2011). Survey on nosql database. In *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on*, pages 363–366.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., and Khan, S. U. (2015). The rise of big data on cloud computing: Review and open research issues. *Information Systems*, 47:98 – 115.

Hecht, R. and Jablonski, S. (2011). Nosql evaluation: A use case oriented survey. In *Cloud and Service Computing (CSC), 2011 International Conference on*, pages 336–341.

Heidemann, J., Klier, M., and Probst, F. (2012). Online social networks: A survey of a global phenomenon. *Computer Networks*, 56(18):3866 – 3878.

Hiruta, S., Yonezawa, T., Jurmu, M., and Tokuda, H. (2012). Detection, classification and visualization of place-triggered geotagged tweets. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 956–963, New York, NY, USA. ACM.

Huang, T., Lan, L., Fang, X., An, P., Min, J., and Wang, F. (2015). Promises and Challenges of Big Data Computing in Health Sciences. *Big Data Research*, 2(1):2–11.

Hutto, C., Yardi, S., and Gilbert, E. (2013). A longitudinal study of follow predictors on twitter. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, page 821.

Indrawan-Santiago, M. (2012). Database research: Are we at a crossroad? reflection on nosql. In *Network-Based Information Systems (NBiS), 2012 15th International Conference on*, pages 45–51.

Jansen, B. J., Zhang, M., Sobel, K., and Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 60(11):2169–2188.

Java, A., Song, X., Finin, T., and Tseng, B. (2007). Why We Twitter : Understanding Microblogging. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65.

Kaleel, S. B. and Abhari, A. (2015). Cluster-discovery of twitter messages for event detection and trending. *Journal of Computational Science*, 6(0):47 – 57.

Kaur, K. and Rani, R. (2013). Modeling and querying data in nosql databases. In *Big Data, 2013 IEEE International Conference on*, pages 1–7.

Kendall, L., Hartzler, A., Klasnja, P., and Pratt, W. (2011). Descriptive analysis of physical activity conversations on twitter. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 1555–1560, New York, NY, USA. ACM.

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., and Smith, N. a. (2014). A Dependency Parser for Tweets. *Proceedings of EMNLP 2014*.

Kontopoulos, E., Berberidis, C., Dergiades, T., and Bassiliades, N. (2013). Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications*, 40(10):4065 – 4074.

Krishnan, K. (2013). *Data Warehousing in the Age of Big Data*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition.

Kumar, S., Barbier, G., Ali Abbasi, M. A., and Liu, H. (2011). TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief. *Fifth International AAAI Conference on Weblogs and Social Media*, pages 661–662.

Kumar, S., Morstatter, F., and Liu, H. (2013a). *Twitter Data Analytics*. Springer, New York, NY, USA.

Kumar, S., Morstatter, F., Zafarani, R., and Liu, H. (2013b). Whom should i follow?: Identifying relevant users during crises. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 139–147, New York, NY, USA. ACM.

Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is Twitter , a Social Network or a News Media ? Categories and Subject Descriptors. *the 19th international conference on World Wide Web*, pages 591–600.

Lachlan, K. A., Spence, P. R., and Lin, X. (2014). Expressions of risk awareness and concern through twitter: On the utility of using the medium as an indication of audience needs. *Computers in Human Behavior*, 35(0):554 – 559.

Lampos, V. and Cristianini, N. (2010). Tracking the flu pandemic by monitoring the social web. In *Cognitive Information Processing (CIP), 2010 2nd International Workshop on*, pages 411–416.

Laney, D. (2001). 3d data management: Controlling data volume, velocity and variety. *Gartner*, 949(February 2001):4.

Lau, J., Collier, N., and Baldwin, T. (2012). On-line Trend Analysis with Topic Models: #twitter Trends Detection Topic Model Online. *International Conference on Computational Linguistics (COLING)*, 2(December):1519–1534.

Leonard, A. (2013). *Pro Hibernate and MongoDB*. Apress, Berkely, CA, USA, 1st edition.

Lin, J. and Cromley, R. G. (2015). Evaluating geo-located Twitter data as a control layer for areal interpolation of population. *Applied Geography*, 58:41–47.

Liu, Y., Huang, X., An, A., and Yu, X. (2007). Arsa: A sentiment-aware model for predicting sales performance using blogs.

López, V., del Río, S., Benítez, J. M., and Herrera, F. (2015). Cost-sensitive linguistic fuzzy rule based classification systems under the mapreduce framework for imbalanced big data. *Fuzzy Sets and Systems*, 258:5 – 38. Special issue: Uncertainty in Learning from Big Data.

Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S., and Miller, R. C. (2011). Tweets as data: Demonstration of tweeql and twitinfo. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 1259–1262, New York, NY, USA. ACM.

McCreary, D. and Kelly, A. (2013). *Making Sense of NoSQL: A Guide for Managers and the Rest of Us*. Manning.

McMinn, A. J., Moshfeghi, Y., and Jose, J. M. (2013). Building a large-scale corpus for evaluating event detection on twitter. *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management - CIKM '13*, pages 409–418.

Mendoza, M., Poblete, B., and Castillo, C. (2010). Twitter under crisis: Can we trust what we rt? In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 71–79, New York, NY, USA. ACM.

Microsoft (2013). The big bang: How the big data explosion is changing the world.

Minelli, M., Chambers, M., and Dhiraj, A. (2013). *Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses (Wiley CIO)*. Wiley Publishing, 1st edition.

Mohanty, S., Jagadeesh, M., and Srivatsa, H. (2013). *Big Data Imperatives: Enterprise Big Data Warehouse, BI Implementations and Analytics*. Apress, Berkely, CA, USA, 1st edition.

Moreira, S., Almeida, M., Martins, B., Filgueiras, J., and Silva, M. J. (2014). TUGAS: Exploiting unlabelled data for twitter sentiment analysis. In *The 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 673–677. ACL.

Morstatter, F., Kumar, S., Liu, H., and Maciejewski, R. (2013). Understanding twitter data with tweetxplorer. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1482–1485, New York, NY, USA. ACM.

Nayak, A., Poriya, A., and Poojary, D. (2013). Article: Type of nosql databases and its comparison with relational databases. *International Journal of Applied Information Systems*, 5(4):16–19. Published by Foundation of Computer Science, New York, USA.

Oussalah, M., Bhat, F., Challis, K., and Schnier, T. (2013). A software architecture for twitter collection, search and geolocation services. *Knowledge-Based Systems*, 37(0):105 – 120.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). 1 Introduction and Motivation 2 A Ranking for Every Page on the Web. *World Wide Web Internet And Web Information Systems*, 54(1999-66):1–17.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Paul, M. J. and Dredze, M. (2011). You are what you tweet: Analyzing twitter for public health. In Adamic, L. A., Baeza-Yates, R. A., and Counts, S., editors, *ICWSM*. The AAAI Press.

Perera, R., Anand, S., Subbalakshmi, K., and Chandramouli, R. (2010). Twitter analytics: Architecture, tools and analysis. In *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, pages 2186–2191.

Petrović, S., Osborne, M., and Lavrenko, V. (2010). The Edinburgh Twitter Corpus. *Computational Linguistics*, (June):25–26.

Poese, I., Uhlig, S., Kaafar, M. A., Donnet, B., and Gueye, B. (2011). Ip geolocation databases: Unreliable? *SIGCOMM Comput. Commun. Rev.*, 41(2):53–56.

Portugal, T. (2014). Os resultados do turismo 2014. `http://www.turismodeportugal.pt/Portugu%C3%AAs/ProTurismo/estat%C3%ADsticas/an%C3%A1lisesestat%C3%ADsticas/osresultadosdoturismo/Anexos/Os%20resultados%20do%20Turismo%20-%202014.pdf`. [Online; acedido em 18-Agosto-2015].

Prieto, V. M., Matos, S., Alvarez, M., Cacheda, F., and Oliveira, J. L. (2014). Twitter: A good place to detect health conditions. *PLoS ONE*, 9(1).

Qu, Y., Huang, C., Zhang, P., and Zhang, J. (2011). Microblogging after a major disaster in china: A case study of the 2010 yushu earthquake. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, pages 25–34, New York, NY, USA. ACM.

Rao, Y., Li, Q., Mao, X., and Wenyin, L. (2014). Sentiment topic models for social emotion mining. *Information Sciences*, 266(0):90 – 100.

Raymond, M. (2010). "how tweet it is! library acquires entire twitter archive", library of congress blog (14 april 2010). `http://blogs.loc.gov/loc/2010/04/how-tweet-it-is-library-acquires-entire-twitter-archive`. [Online; acedido em 13-Agosto-2015].

Rill, S., Reinel, D., Scheidt, J., and Zicari, R. V. (2014). PoliTwi: Early detection of emerging political topics on twitter and the impact on concept-level sentiment analysis. *Knowledge-Based Systems*, 69:24–33.

Rosa, H., Carvalho, J. P., and Batista, F. (2014). Detecting a tweet's topic within a large number of Portuguese Twitter trends. i:1–15.

Sagiroglu, S. and Sinanc, D. (2013). Big data: A review. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 42–47.

Saif, H., He, Y., and Alani, H. (2012). Semantic sentiment analysis of twitter. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7649 LNCS(PART 1):508–524.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, New York, NY, USA. ACM.

Santos, C. J. and Matos, S. (2013). Predicting flu incidence from portuguese tweets. In *International Work-Conference on Bioinformatics and Biomedical Engineering 2013. Proceedings*.

Santos, J. and Matos, S. (2014). Analysing twitter and web queries for flu trend prediction. *Theoretical Biology and Medical Modelling*, 11(Suppl 1):S6.

Scanfeld, D., Scanfeld, V., and Larson, E. L. (2010). Dissemination of health information through social networks: Twitter and antibiotics. *American Journal of Infection Control*, 38(3):182–188.

Schnoebelen, T. (2012). Do you smile with your nose? stylistic variation in twitter emoticons. *Working Papers in Linguistics, University of Pennsylvania*, 18.

Shim, S. S. (2012). Guest editor's introduction: The cap theorem's growing impact. *Computer*, 45(2):21–22.

Smith, T. (2013). Ted-ed : Exploration on the big data frontier.

Souza, M. and Vieira, R. (2012). Sentiment analysis on twitter data for portuguese language. In *10th International Conference, PROPOR 2012, Coimbra, Portugal, April 17-20, 2012. Proceedings*, pages 241–247.

Spencer, J. and Uchyigit, G. (2012). Sentimentor: Sentiment analysis of twitter data. *CEUR Workshop Proceedings*, 917:56–66.

Sundar, D. S. and Kumaresh (2013). Probing of Geospatial Stream Data To Report. pages 227–232.

Tiwari, S. (2011). *Professional NoSQL*. Wrox programmer to programmer. John Wiley, Hoboken, N.J. Wiley Chichester. Index.

Tornes, A. (2013). 4 things you need to know about migrating to version 1.1 of the twitter api. `https://blog.gnip.com/migrating-version1-1-twitter-api`. [Online; acedido em 25-Julho-2015].

Tudorica, B. and Bucur, C. (2011). A comparison between several nosql databases with comments and notes. In *Roedunet International Conference (RoEduNet), 2011 10th*, pages 1–5.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185.

Twitter (2015a). Introducing new metadata for tweets. disponível em https://blog.twitter.com/2013/introducing-new-metadata-for-tweets, à data de 30/06/2015.

Twitter (2015b). Working with timelines.

Van Kleek, M., Smith, D., Stranders, R., and schraefel, m. (2012). Twiage: A game for finding good advice on twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 889–898, New York, NY, USA. ACM.

Vaz, C. A. (2012). *Os 8 Ps do Marketing Digital: O Guia Estratégico de Marketing Digital*. NOVATEC.

Vicente, M., Batista, F., and Carvalho, J. P. (2015). Twitter gender classification using user unstructured information. In *FUZZ-IEEE 2015, IEEE International Conference on Fuzzy Systems*, IEEE Xplorer, Istanbul, Turkey.

Vosecky, J., Jiang, D., and Ng, W. (2013). Limosa: A system for geographic user interest analysis in twitter. In *Proceedings of the 16th International Conference on Extending Database Technology*, EDBT '13, pages 709–712, New York, NY, USA. ACM.

Wang, A. H. (2010). Don't follow me: Spam detection in Twitter. *2010 International Conference on Security and Cryptography (SECRYPT)*, pages 1–10.

Ward, J. S. and Barker, A. (2013). Undefined By Data: A Survey of Big Data Definitions. *arXiv.org*.

Weerkamp, W., Carter, S., and Tsagkias, M. (2011a). How people use twitter in different languages. In *Web Science 2011*, Koblenz. ACM, ACM.

Weerkamp, W., Carter, S., and Tsagkias, M. (2011b). How People use Twitter in Different Languages. *Proceedings of the ACM WebSci'11*, (1):1–2.

Widener, M. J. and Li, W. (2014). Using geolocated twitter data to monitor the prevalence of healthy and unhealthy food references across the {US}. *Applied Geography*, 54(0):189 – 197.

Yang, C., Zhang, X., Zhong, C., Liu, C., Pei, J., Ramamohanarao, K., and Chen, J. (2014). A spatiotemporal compression based approach for efficient big data processing on cloud. *Journal of Computer and System Sciences*, 80(8):1563 – 1583. Special Issue on Theory and Applications in Parallel and Distributed Computing Systems.

Yang, M.-C. and Rim, H.-C. (2014). Identifying interesting twitter contents using topical analysis. *Expert Systems with Applications*, 41(9):4330 – 4336.

Zagheni, E., Garimella, V., and Weber, I. (2014). Inferring international and internal migration patterns from Twitter data. *WWW'14 Companion*, pages 1–6.

Zikopoulos, I., Eaton, C., and Zikopoulos, P. (2011). *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. Mcgraw-hill.

Zimmer, M. (2015). "the twitter archive at the library of congress: Challenges for information practice and information policy",first monday, volume 20, number 7 - 6 july 2015. `http://firstmonday.org/ojs/index.php/fm/article/view/5619/4653`. [Online; acedido em 13-Agosto-2015].