Sistema Inteligente de Recolha, Armazenamento e Visualização de Informação proveniente do Twitter

A Smart System for Twitter Corpus Collection, Management and Visualization

Gaspar Bogueira, ISCTE-IUL – Instituto Universitário de Lisboa & INESC-ID Lisboa, Portugal gmrba@iscte.pt

Fernando Batista, ISCTE-IUL – Instituto Universitário de Lisboa & INESC-ID Lisboa, Portugal fernando.batista@inesc-id.pt

Joao Paulo Carvalho, Instituto Superior Técnico - Universidade de Lisboa & INESC-ID Lisboa, Portugal joao.carvalho@inesc-id.pt

Resumo

As redes sociais têm ganho bastante popularidade para a partilha de informação sobre os mais diversos tópicos desde a política, desporto ou mesmo aspetos do quotidiano. As mensagens partilhadas no Twitter[1] (tweets) são essencialmente públicas constituindo uma fonte de informação que por ser difundida em tempo real pode revelar-se útil em domínios como o turismo, marketing, saúde ou segurança. Este artigo descreve o desenvolvimento de um Sistema de Informação envolvendo a criação de um repositório de tweets (*corpus*) escritos em Português Europeu e publicados em Portugal. O sistema envolve também uma REST API que permite o acesso à informação armazenada e um Dashboard Web para análise e visualização de indicadores sobre os dados.

Palavras-chave: Sistema de Informação, Twitter, MongoDB, Geolocalização, Redes Sociais

Abstract

Social networks have become popular and are now becoming an alternate mean of communication, used to share information on various topics, ranging from politics or sports to simple aspects of everyday life. Twitter messages (tweets) are shared in real time and are essentially public, making them a useful source of information for areas such as tourism, marketing, health, and safety. This paper describes an information system that involves the creation and storage of a ***corpus*** *of tweets, written in European Portuguese and published within the Portuguese territory. The system also involves a REST API that allows access to the stored information, and a web-based dashboard that makes it possible to analyze and visualize indicators concerning the stored data.*

Keywords: *Information System, Twitter, MongoDB, Geolocation, Social Networks*

[1] https://twitter.com/

ISSN 2183-489X
DOI http://dx.doi.org/10.18803/capsi.v15.418-439

1. INTRODUÇÃO

Uma grande parte da sociedade atual vive hiperconectada. Apesar da distância, a todo o momento estamos próximos de amigos ou de outras pessoas cujas vidas queremos acompanhar através de fotografias colocadas no Instagram, conteúdos que são partilhados no Twitter ou no Facebook, vídeos que se colocam no Youtube, etc.. 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 pelas comunidades em que participa no Orkut, faz com que os gostos interesses e demais características sejam expostas ao mundo através das redes sociais. A rede social tal como é definida por Boyl *et. al.*[2007] é um serviço baseado na *web* que permite construir um perfil público ou semipúblico num sistema delimitado e gerir uma lista de outros utilizadores com os quais se mantem uma conexão e se acede quer à sua lista de conexões quer à lista de conexões de outros utilizadores, visualizando as suas atualizações de estado. As redes sociais tendem a transformar-se não só num ambiente para seguirmos a vida de amigos, mas também numa plataforma de interação com empresas, marcas, aplicações ou serviços.

A análise dos conteúdos e informações partilhadas nas redes sociais tem-se revelado bastante útil em diversas áreas, incluindo a política, o turismo, a saúde pública ou a segurança. No caso particular do Twitter, a rede social em que incidirá o estudo apresentado neste trabalho, são publicados em média cerca de 500 milhões de novos tweets por dia [Twitter 2015]. O Twitter disponibiliza o livre acesso a alguma da informação produzida pelos seus utilizadores através de APIs (*Application Programming Interface*) públicas. A popularidade do Twitter como fonte de informação tem conduzido ao desenvolvimento de inúmeras aplicações e investigações em diversos domínios. Com cerca de um milhão e meio de tweets relacionados com a saúde, onde constavam menções a diversas doenças incluindo alergias, obesidade e insónias, Paul *et al.* [2011] desenvolveu um método para rastreamento de doenças medindo fatores de risco ao nível comportamental, localizando doenças por regiões geográficas ao analisar os sintomas e medicação aplicada. Igualmente, com informação obtida do Twitter, Santos *et al.* [2013, 2014] utilizou um conjunto de aproximadamente 2700 tweets produzidos em Portugal para prever a taxa de incidência e disseminação do vírus *influenza* na população Portuguesa. Widener *et al.* [2014] utilizou uma *framework* de Data Mining para, através da aplicação de métodos de extração de informação e análise de sentimentos, tentar compreender como tweets geolocalizados podem ser utilizados na pesquisa da prevalência de alimentos saudáveis e não saudáveis em regiões contíguas dos Estados Unidos. Outros estudos relacionados com a saúde pública foram também desenvolvidos por Culotta [2010] e Scanfeld *et al.* [2010].

O Twitter foi também usado como fonte de informação para ajudar a identificar ou localizar a ocorrência de sismos [Sakaki *et al.* 2010] dado que "ao ocorrer um sismo as pessoas produzem muitos *posts* no Twitter relacionados com o acontecimento o que permite a identificação de sismos simplesmente pela observação do aumento do volume de tweets". No estudo desenvolvido por Kumar *et al.* [2013b] é proposta uma abordagem 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 crise. Por exemplo, no caso de um sismo, os tweets produzidos num local próximo ao sismo são provavelmente mais pertinentes para avaliação do estado da situação do que tweets produzidos num local mais distante. Outros estudos foram produzidos tendo por base tópicos semelhantes [Mendoza *et al.* 2010, Qu *et al.* 2011, Lachlan *et al.* 2014].

Gerber [2014] refere que o Twitter é uma fonte de dados ideal para problemas de suporte à decisão. 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.

O desenho de uma arquitetura de software para a captura e extração de informação contida em tweets configura-se como um grande desafio considerando as limitações no acesso à informação impostas pela Twitter API [Oussalah *et al.* 2013]. Perera *et al.* [2010] descreve uma arquitetura de software baseada na Twitter API que, recorrendo a Python e MySQL, recolhe os tweets enviados para utilizadores específicos. Na 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 do Twitter, nomeadamente o Presidente Barack Obama. Anderson *et al.* [2011] apresenta uma arquitetura com elevado grau de concorrência de modo a permitir a recolha de um grande volume de dados com um máximo teórico de 500 milhões de tweets por dia. O código desenvolvido é *multithreaded* e está adaptado para executar em máquinas com diversos processadores. Foram utilizadas as *frameworks* Spring, MVC, Hibernate e JPA e os componentes de infraestrutura Tomcat, MySQL e Lucene. Por outro lado, Marcus *et al.* [2011] desenvolveu o TwitInfo, uma plataforma para recolha e processamento de tweets em tempo real para efeitos de Análise de Sentimento. A arquitetura proposta por Oussalah *et al.* [2013] tem como objetivo

a análise semântica e espacial de dados recolhidos do Twitter. Na 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). A implementação do software é baseada em Django (*framework* para desenvolvimento de aplicações web em Python), Apache Lucene (para indexação dos tweets) e MySQL para armazenamento dos dados recolhidos. Esta arquitetura permite a pesquisa de tweets por texto, nome de utilizadores ou localização.

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 consequentemente um dos principais formadores da perceção da sociedade e do mundo que nos rodeia. Hoje em dia os eventos importantes são muitas vezes comentados nas redes sociais, mesmo antes de se tornarem "notícia pública" e até mesmo as agências noticiosas tiveram de se adaptar e começar a usar as redes sociais como fonte de informação. Apesar da sua importância, muitas questões sobre o efeito das redes sociais na sociedade estão ainda por analisar devidamente. Para que possam ser analisadas questões como quais os temas, tópicos ou eventos importantes partilhados nas redes sociais, quais os atores principais dentro dos temas, qual a origem e propagação dos temas (*timeline*), o que faz determinado evento tornar-se importante nas redes sociais, quanto tempo é necessário para que um evento tenha impacto nas redes sociais e na sociedade, qual o papel dos "principais atores" na propagação dos eventos, é necessária a recolha, processamento e análise de informações partilhadas nas redes sociais.

O objetivo principal deste trabalho está relacionado com o desenvolvimento de uma arquitetura para recolha, processamento e armazenamento da informação partilhada no Twitter em Portugal e escrita em Português Europeu, resultando num Sistema de Informação que agrega quatro módulos para recolha, armazenamento, acesso e visualização dos dados. Este artigo encontra-se estruturado nas seguintes secções: a Secção 2 introduz de forma resumida o Sistema de Informação resultante deste trabalho; nas Secções 3 e 4 são descritos os processos de recolha, armazenamento e processamento de informação extraída do Twitter; as Secções 5 e 6 apresentam as ferramentas desenvolvidas para acesso e visualização dos dados e por fim, na Secção 7, são mencionadas as principais conclusões e as propostas para continuação do trabalho desenvolvido.

2. SISTEMA DE INFORMAÇÃO

A metodologia para recolha, processamento, armazenamento, acesso e visualização de dados do Twitter apresentada neste artigo culminou no desenvolvimento de um Sistema de Informação

que poderá permitir a realização de diversos tipos de estudos e aplicações baseadas nos dados recolhidos. A Figura 1 apresenta o fluxo da informação entre os quatro módulos que constituem o Sistema de Informação: no módulo *Recolha* são capturados tweets geolocalizados em Portugal e destes são descartados os tweets cujo texto não se encontra escrito em Português Europeu; no módulo *Expansão* é recolhida a timeline de cada um dos utilizadores portugueses identificados no módulo *Recolha*, conduzindo à expansão da informação capturada sobre cada utilizar; o módulo *Acesso* possibilita o acesso ao *corpus* através de uma REST API que abstrai o acesso à base de dados; por último, o módulo *Visualização* apresenta na forma de um Dashboard Web algumas métricas e indicadores sobre os dados recolhidos. Nas Secções seguintes descrevem detalhadamente cada um dos módulos.

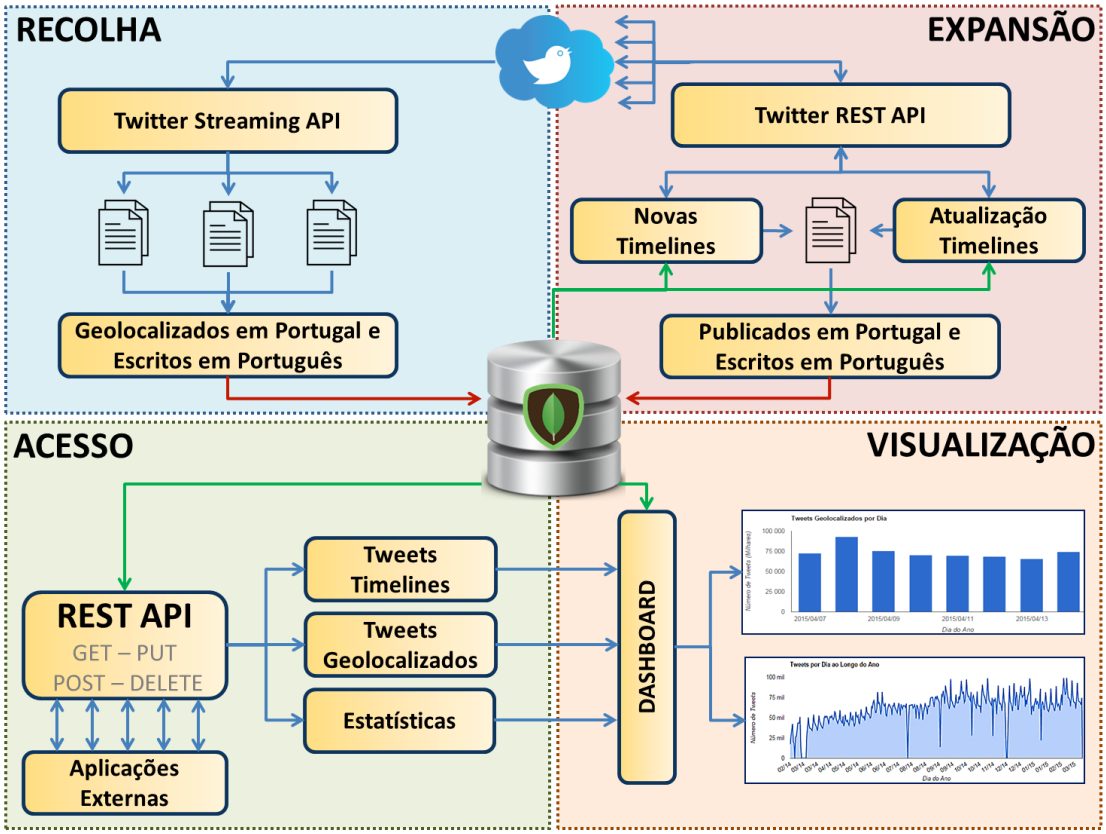

Figura 1: Sistema de informação para recolha e visualização de dados do Twitter.

3. Recolha e Armazenamento de Dados Geolocalizados

A localização geográfica de um tweet pode ser obtida seguindo uma de duas abordagens:

A. Geolocalização: os utilizadores podem optar por tornar pública a informação da sua localização no momento da publicação de um novo tweet. Esta informação pode ser

bastante precisa se o tweet for publicado com recurso, por exemplo, a um *smartphone* com GPS.

B. Perfil do Utilizador: a localização de determinado utilizador pode ser extraída ou inferida pela informação contida no campo *location* do perfil de utilizador. Esta informação é igualmente disponibilizada na API do Twitter.

Tendo como objetivo a recolha de tweets geolocalizados em Portugal foi explorada a primeira abordagem, recorrendo à informação disponibilizada pela Twitter API, nomeadamente a Streaming API *statuses/filter*[2]. O acesso aos dados disponibilizados pela Twitter API só é permitido após autenticação pelo protocolo aberto OAuth[3]. Após autenticação e envio do primeiro pedido HTTP para a Streaming API *statuses/filter* é disponibilizado o acesso ao fluxo de tweets produzidos em tempo real, permanecendo ativo o acesso ao fluxo sem necessidade de posterior interação por parte da aplicação cliente que iniciou a conexão à referida API.

O armazenamento dos tweets provenientes da *statuses/filter* API é efetuado continuamente em ficheiros comprimidos em disco. A estrutura de ficheiros utilizada para armazenar os tweets é criada dinamicamente sendo os ficheiros organizados em diretórios cuja nomenclatura é constituída pela referência ao ano, mês e ao dia em que os dados são recolhidos. A cada hora é criado um novo ficheiro guardado no diretório correspondente ao ano, mês e dia em que os tweets são capturados.

O fluxo de tweets retornado pela *statuses/filter* API está de acordo com um ou mais filtros. Esta API permite a pesquisa por palavras-chave, *hashtags*, *id* de utilizador ou regiões delimitadas geograficamente [Kumar *et al.* 2013a]. Porém, genericamente, o Twitter impõe limitações à utilização da sua API tanto no número de parâmetros como na quantidade de resultados devolvidos. Atualmente, cada pedido à *statuses/filter* API admite como parâmetros no máximo 400 palavras-chave, 25 delimitações de regiões geográficas ou 5000 *id* de utilizador. Como resultado são retornados todos os tweets que satisfaçam as restrições do pedido até um certo limite, que no caso da *statuses/filter* API é de 1% do volume total de tweets produzidos no Twitter em determinado instante [Kumar *et al.* 2013a]. Considerando que diariamente são produzidos a nível mundial aproximadamente 500 milhões de tweets [Twitter 2015] poderão teoricamente ser recolhidos da s*tatuses/filter* API cerca de 5 milhões de tweets por dia.

Em resultado de estudos prévios a este artigo concluiu-se que em Portugal são produzidos diariamente em média cerca de 60 mil tweets geolocalizados e escritos em Português Europeu

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

[Brogueira *et al.* 2015], pelo que o volume de tweets que diariamente é possível recolher da s*tatuses/filter* API é bastante superior à quantidade de tweets geolocalizados que efetivamente são publicados em Portugal.

Na arquitetura proposta, o fluxo de tweets geolocalizados após ser guardado na estrutura de ficheiros comprimidos em disco referida anteriormente, é processado de modo a filtrar os tweets que não estejam escritos em Português Europeu ou que não tenham sido produzidos em Portugal.

Esta filtragem é necessária visto que embora na invocação da *statuses/filter* API sejam indicadas as delimitações geográficas de Portugal, a dificuldade de representação exata dos limites geográficos de Portugal tem como consequência a recolha de alguns tweets de regiões geograficamente próximas de Portugal, tal como Espanha ou Marrocos. Estes tweets são filtrados pela validação do campo *lang*[4] que deverá conter o valor igual a "pt" e o campo *place.country*[5] que deverá conter o valor "Portugal".

A verificação dos campos *lang* e *place.country* permite também detetar os casos em que os algoritmos de deteção automática de linguagem utilizados pelo Twitter [Twitter 2013] não identificam de forma correta a língua em que a mensagem de determinado tweet foi escrita. Por vezes verifica-se a ocorrência de tweets cujo campo lang tem erradamente o valor "pt", visto que o campo text não se encontra escrito em Português embora o valor do campo place.country esteja aparentemente de acordo com a língua em que o tweet foi efetivamente escrito. A Figura 2 apresenta um excerto de um tweet cujo valor do campo lang não corresponde à língua em que o texto do tweet foi escrito.

```
{
   ”text”: “a mi o me aportas o te apartas.”,
   ”lang”: “pt”,
   ”created_at”: “2014-02-20 16:23:59”,
   ”user”:{
      ”geo_enabled”: true
   },
   ”place”:{
      ”full_name”: “Lucena del Puerto, Huelva”,
      ”country”: “España”,
      ”country_code”: “ES”
   }
}
```

Figura 2: Excerto de um tweet cujo campo *lang* foi incorretamente classificado pelo Twitter.

[4] *Lang* – Campo que identifica a língua em que o texto do tweet foi escrito, através de algoritmos de reconhecimento de língua do Twitter.
[5] *Place.Country* – Campo que identifica o país em que o tweet foi publicado.

Como resultado do processo de filtragem são considerados "válidos" todos os tweets cujo valor do campo lang é igual a "pt" e o campo place.country é igual a "Portugal". Estes tweets serão armazenados numa base de dados MongoDB que conterá apenas os tweets geolocalizados em Portugal e escritos em Português Europeu capturados da statuses/filter API.

Aos tweets validados no contexto deste trabalho antes de serem inseridos no MongoDB é-lhes adicionado um novo campo que se convencionou designar por created_at_object que conterá o mesmo valor que o campo created_at (que indica a data de publicação do tweet), mas convertido para um objeto do tipo *Date* que facilita a posterior pesquisa e manipulação dos tweets em base de dados. A Figura 3 apresenta um exemplo da alteração efetuada a cada tweet.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 3: Documento JSON que guarda a informação sobre um utilizador.

Paralelamente ao armazenamento dos tweets geolocalizados em MongoDB é registado numa outra coleção MongoDB alguma informação relativa aos autores dos tweets. Para cada autor é construído um objeto JSON com a estrutura apresentada na Figura 4.

```
{
   "_id": Number,
   "timeline_status": String,
   "last_timeline_status_date": ISODate,
   "user_date": ISODate
}
```

Figura 4: Documento JSON que guarda a informação sobre um utilizador.

O campo _id corresponde ao identificador único do utilizador que se encontra no campo user.id de cada tweet, associando o tweet ao respetivo autor. Os campos timeline_status e last_timeline_status_date permitem controlar o estado do utilizador no âmbito da arquitetura apresentada neste artigo cujo significado será detalhado na Secção 4. No campo user_date é guardada a data em que o utilizador foi identificado pela primeira vez como tendo publicado um tweet geolocalizado capturado pelo procedimento apresentado neste artigo.

4. EXPANSÃO DA BASE DE DADOS

O conjunto de utilizadores cuja informação é armazenada pelo método apresentado na Secção 3 irá permitir a expansão da base de dados de tweets pela recolha da timeline de cada um dos utilizadores, isto é, serão lidos os tweets do histórico da atividade recente de todos os utilizadores. A timeline está acessível através da REST API *statuses/user_timeline*[6] que dado um *id* de utilizador retorna os últimos 3200 tweets produzidos pelo respetivo utilizador. O procedimento de recolha da timeline de todos os utilizadores não é um processo trivial, considerando as restrições impostas pelo Twitter na utilização da API *statuses/user_timeline*. De uma forma geral o número de acessos permitidos à API do Twitter é contabilizado em períodos de 15 minutos sendo autorizados 180 pedidos à API em cada período utilizando autenticação do nível de utilizador ou 300 pedidos no caso de autenticação com nível de aplicação [Kumar *et al.* 2013a]. No presente trabalho foram utilizadas somente contas de acesso com autenticação do nível de utilizador. Em cada invocação à API *statuses/user_timeline* são retornados no máximo 200 tweets pelo que a leitura dos 3200 tweets da timeline necessita de 16 pedidos à API. Considerando que com uma conta de autenticação de nível de utilizador é possível efetuar 180 invocações à API *statuses/user_timeline* em cada período de 15 minutos e assumindo o caso limite em que todos os utilizadores têm na sua timeline 3200 tweets, é recolhida a timeline completa de cerca de 11 utilizadores a cada intervalo de 15 minutos. Neste cenário serão recolhidas 45 timelines por hora e 1080 timelines completas por dia. Com esta limitação é imperativo que o método de leitura das timelines seja otimizado de modo a minimizar o desperdício de tempo entre invocações, evitando não só a repetição de pedidos como a falha de invocações devido ao excesso de solicitações num dado período. O método de expansão do *corpus* apresentado nesta Secção é executado continuamente procedendo à recolha, processamento e armazenamento da timeline completa de cada novo utilizador integrado na base de dados e à pesquisa dos novos tweets publicados desde a última atualização das restantes timelines.

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

De acordo com o referido na Secção 6, são incluídos na base de dados diariamente em média 232 novos utilizadores. Considerando que o universo de utilizadores apresenta um crescimento constante e contando já com mais de 100 mil utilizadores, à data da escrita deste artigo, atualizar continuamente todas as *timelines* utilizando apenas uma conta para acesso à Twitter API, facilmente se depreende que o intervalo de tempo necessário para atualizar todas as timelines é bastante elevado e tendencialmente crescente. Com o intuito de minimizar o período entre duas iterações completas para atualização de todas as timelines, foram criadas e registadas diversas contas para acesso à Twitter API. Sobre cada conta é guardada a informação apresentada na Figura 5 onde os campos *ConsumerKey* e *ComsumerSecret* identificam univocamente a aplicação junto do Twitter e os campos *AccessToken* e *AccessTokenSecret* contêm a chave de autenticação no Twitter.

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

Figura 5: Documento JSON com as credenciais de acesso à API do Twitter.

O campo *timeline_status* pode assumir diversos valores estando diretamente relacionado com o campo *timeline_status* da informação associada a cada utilizador (*cf.* Figura 4), uma vez que cada conta recolhe a timeline dos utilizadores cujo valor do campo *timeline_status* é igual ao valor do campo com o mesmo nome na informação do utilizador. Deste modo de entre o conjunto de contas para processamento das timelines existem contas dedicadas à recolha da timeline dos novos utilizadores e contas dedicadas à atualização das timelines dos utilizadores já incluídos na base de dados. Dado que o número de novos utilizadores integrados por dia é teoricamente inferior ao máximo de timelines que é possível processar com apenas um cliente, é portanto suficiente que apenas uma conta seja dedicada à recolha da timeline dos novos utilizadores, ficando os restantes clientes alocados à tarefa de atualização das timelines.

O procedimento de leitura das timelines tem em consideração o estado indicado no campo *timeline_status*, associado à informação de cada utilizador permitindo categorizar cada utilizador

relativamente ao estado da recolha da respetiva timeline. Este campo pode assumir os seguintes estados: "Leitura Integral", "Atualizar", "Em atualização", "Bloqueado", "Casual" ou "Erro" cujo significado é apresentado na Tabela 1.

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
Bloqueado	O utilizador não permite o acesso à *timeline*
Casual	Os novos tweets não foram publicados em Portugal
Erro	O processo de recolha terminou em erro

Tabela 1: Estados da recolha da *timeline* de um utilizador.

Quando um novo utilizador é adicionado à base de dados de utilizadores o campo *timeline_status* é colocado por defeito no estado "Leitura Integral", indicando que o utilizador foi introduzido no sistema e que se deverá proceder à recolha da totalidade da sua timeline. Concluída a recolha ou atualização da timeline, o campo *last_timeline_status_date* é atualizado com a data correspondente ao dia em que o processamento foi realizado. Este facto é bastante relevante uma vez que o processo de atualização das timelines dá prioridade aos utilizadores cujo campo *last_timeline_status_date* contém a data mais antiga.

O diagrama de transição de estados do processo de recolha da timeline é apresentado na Figura 6. Quando determinado utilizador é selecionado para que seja recolhida ou atualizada a respetiva timeline o seu estado é alterado para "Em Atualização". Deste modo evita-se que um mesmo utilizador possa estar a ser processado por mais do que um cliente em simultâneo. No caso do processamento da timeline terminar como esperado o campo *timeline_status* é colocado no estado "Atualizar" fazendo com que na próxima iteração do procedimento, sejam apenas pesquisados os novos tweets publicados por determinado utilizador desde o seu tweet mais recente, guardado na base de dados. O campo *last_timeline_status_date* é igualmente atualizado com a data do dia do processamento. No entanto, o estado "Em Atualização" pode originar a passagem a outros estados, nomeadamente, ao estado de "Erro" caso ocorra alguma situação inesperada durante a execução do processo; ao estado "Bloqueado" quando o utilizador não permite o acesso público à timeline ou ao estado "Casual" para o caso de utilizadores que já publicaram tweets em Portugal e escritos em português mas cujas posteriores mensagens não foram publicadas em Portugal ou escritas numa outra língua que não o português. A passagem a um destes estados não implica a atualização do campo *last_timeline_status_date*. Também para os estados de erro existe uma conta dedicada que periodicamente testa a possibilidade de "recuperação" dos utilizadores

nestes estados. Quando o campo *user_date* permanece igual ao campo *last_timeline_status_date* significa que o utilizador nunca passou ao estado "Atualizar" pelo que a tentativa de "recuperação" pressupõe a passagem para o estado "Leitura Integral". Por outro lado, quando o campo *user_date* de determinado utilizador difere do campo *last_timeline_status_date* significa que em dado momento o processamento integral da sua timeline foi efetuado com sucesso pelo que a tentativa de "recuperação" irá recolocar o utilizador no estado "Atualizar".

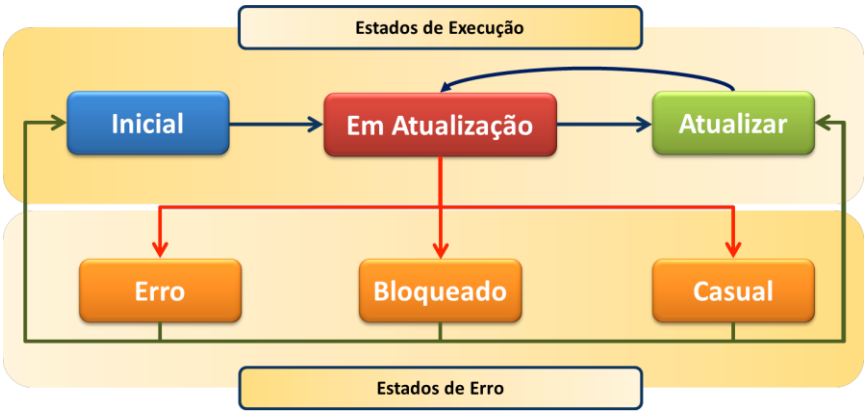

Figura 6: Diagrama de transição de estados na recolha da *timeline*.

À imagem do procedimento de recolha dos tweets geolocalizados os tweets recolhidos das timelines são armazenados em ficheiros comprimidos em disco. Neste caso os ficheiros são guardados numa estrutura organizada de forma diferente, onde o nome de cada diretório é composto pela sequência de caracteres 000, 001, 002 … 999. Por cada invocação da API statuses/user_timeline é gerado um ficheiro com os tweets devolvidos na resposta da API, sendo este guardado no diretório correspondente aos três últimos dígitos do id de utilizador processado. Esta nomenclatura permite uma distribuição uniforme dos ficheiros por cada diretório, visto que com a elevada quantidade de invocações à API e consequente número de ficheiros criados, rapidamente se atingiria o limite de ficheiros que é possível armazenar num só diretório caso todos eles fossem guardados em apenas um diretório.

Na Figura 7 é apresentado o fluxograma que resume a arquitetura proposta para recolha e expansão de uma base de dados de tweets produzidos em Portugal e escritos em português europeu. A parte esquerda representa a recolha dos tweets geolocalizados e a parte direita corresponde ao procedimento de leitura das timelines dos utilizadores portugueses que publicam tweets com recurso a dispositivos que permitem partilhar a sua geolocalização.

5. ACESSO AOS DADOS

Os tweets recolhidos pelo mecanismo apresentado neste artigo não constituirão uma mais-valia considerando a informação que deles se pode retirar, se não forem processados e analisados. Por vezes este tipo de análises é efetuado por especialistas de áreas cujo conhecimento de conceitos como a representação dos tweets em objetos JSON ou a sintaxe da linguagem para pesquisa de informação numa base de dados MongoDB não é muito acentuado, constituindo uma dificuldade acrescida. Nesse sentido foi implementada uma REST API que pode ser vista como uma forma de abstrair todos estes conceitos, facilitando o acesso aos dados.

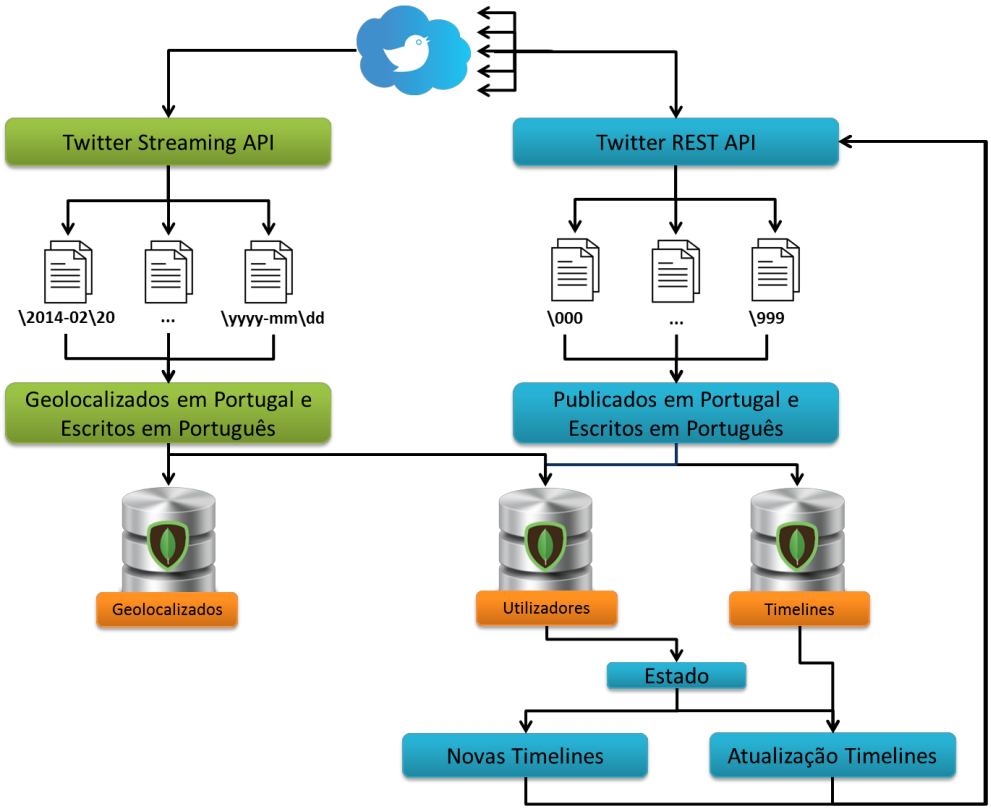

Figura 7: Arquitetura de recolha e expansão da base de dados de Tweets.

O conceito de REST (*Representational State Transfer*) apresentado por Fielding [2000] refere-se a um conjunto de princípios de arquitetura de interfaces web. De um modo geral uma API expõe um conjunto de dados e funções para facilitar a interação e a troca de informações entre aplicações e *web services*. Uma web API que é desenhada em conformidade com os princípios de arquitetura REST é designada por REST API.No contexto deste Sistema de Informação foi implementada uma REST API que disponibiliza um conjunto de serviços (*endpoints*) através dos quais é permitido o acesso à informação guardada na base de dados MongoDB. Os serviços implementados podem ser vistos como uma camada de abstração em relação à base de dados

MongoDB permitindo o desenvolvimento de aplicações baseadas na informação nela armazenada, sem a necessidade de conhecimento do modelo de dados ou da sintaxe para pesquisa e extração de dados. A REST API foi implementada com recurso à microframework Flask[7] que permite o desenvolvimento de aplicações web em Python. Os serviços disponibilizados com a REST API permitem apenas a consulta e leitura de dados, podendo ser divididos em três grupos: acesso à coleção de tweets, acesso a informações sobre os utilizadores e acesso a estatísticas e indicadores gerais sobre os dados. Para acesso à coleção de tweets os serviços mais relevantes são indicados na Tabela 2.

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 pela ordem de publicação decrescentemente
/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"

Tabela 2: Serviços disponíveis na REST API para acesso aos tweets.

Dado que os tweets geolocalizados e os tweets lidos das timelines são armazenados em coleções de dados distintas, a definição do nome dos serviços da Tabela 2 é composto pelo nome da coleção (<collection>) que poderá assumir os valores "geolocated" ou "timeline", respetivamente. O acesso a informações relativas aos utilizadores é disponibilizado pelos serviços da Tabela 3.

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

[7] http://flask.pocoo.org/

/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 permitem inferir a idade e o género do utilizador indicado em @screen_name

Tabela 3: Serviços da REST API para acesso à informação de cada utilizador.

Devido ao elevado volume de dados armazenados determinadas pesquisas podem tornar-se bastante demoradas. Para tal, o resultado de pesquisas mais complexas é pré-processado e guardado também em MongoDB permitindo a visualização de estatísticas sobre os dados de forma consideravelmente mais rápida. Algumas das pesquisas pré-processadas são disponibilizadas pelos serviços da Tabela 4.

Tabela 4: Serviços da REST API para acesso a estatísticas pré-processadas.

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

Na Figura 8 é esquematizada a forma como através da REST API se realiza a ligação e extração de dados do 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 mediante a realização de pedidos HTTP a cada *endpoint* sendo devolvida a informação solicitada, tal como é demonstrado na Figura 9. Na resposta às invocações é utilizada a biblioteca Python Jinja2[8] que

[8] http://jinja.pocoo.org/docs/dev/

auxilia na criação de templates HTML, disponibilizando os dados num formato adequado à visualização e interpretação dos mesmos.

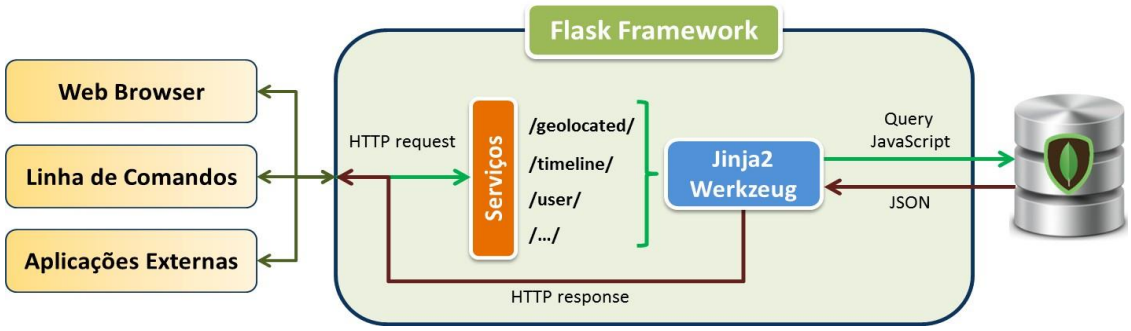

Figura 8: Interação entre aplicações externas e o MongoDB via REST API.

O exemplo da Figura 9 apresenta o resultado da invocação do *endpoint* api/geolocated/tweet/ passando como parâmetro o *id* de tweet 551528493021147136, obtendo-se como resposta o objeto JSON correspondente ao tweet.

```
> 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",
    (...)

}
```

Figura 9: Exemplo de utilização da REST API.

6. VISUALIZAÇÃO E MÉTRICAS SOBRE OS DADOS

A execução contínua do procedimento de recolha de tweets descrito neste artigo resulta na captação de um grande volume de informação, conduzindo à necessidade de agregar parte dessa informação num formato que permita uma visão geral de alguns indicadores de desempenho e performance de todo o Sistema de Informação. Justamente por esse motivo foi desenvolvido um Dashboard Web que resume parte da informação recolhida de uma forma clara, concisa e acima

de tudo, visual. Recorrendo à API do Google para criação de gráficos Google Charts[9] e com a invocação dos serviços da REST API, descritos na Secção 5, relativos às estatísticas sobre os dados armazenados ou consultando diretamente a base de dados MongoDB foi desenvolvido um Dashboard. Um dos cinco quadros do Dashboard é resumidamente apresentado na Figura 10, onde se podem observar algumas estatísticas relativamente aos tweets geolocalizados. No gráfico 1 são indicados os tweets recolhidos por dia, no gráfico 2 é indicado o número de tweets recolhidos por hora e no gráfico 3 é indicado o número de tweets recolhidos para cada um dos dias da semana. No gráfico 4 observa-se o número de tweets recolhidos por dia ao longo do último mês de recolha de dados, que neste caso concreto corresponde a maio de 2015. O gráfico 5 apresenta a soma de tweets geolocalizados recolhidos por mês ao longo do período durante o qual foi efetuada a recolha de dados.

[9] https://developers.google.com/chart/

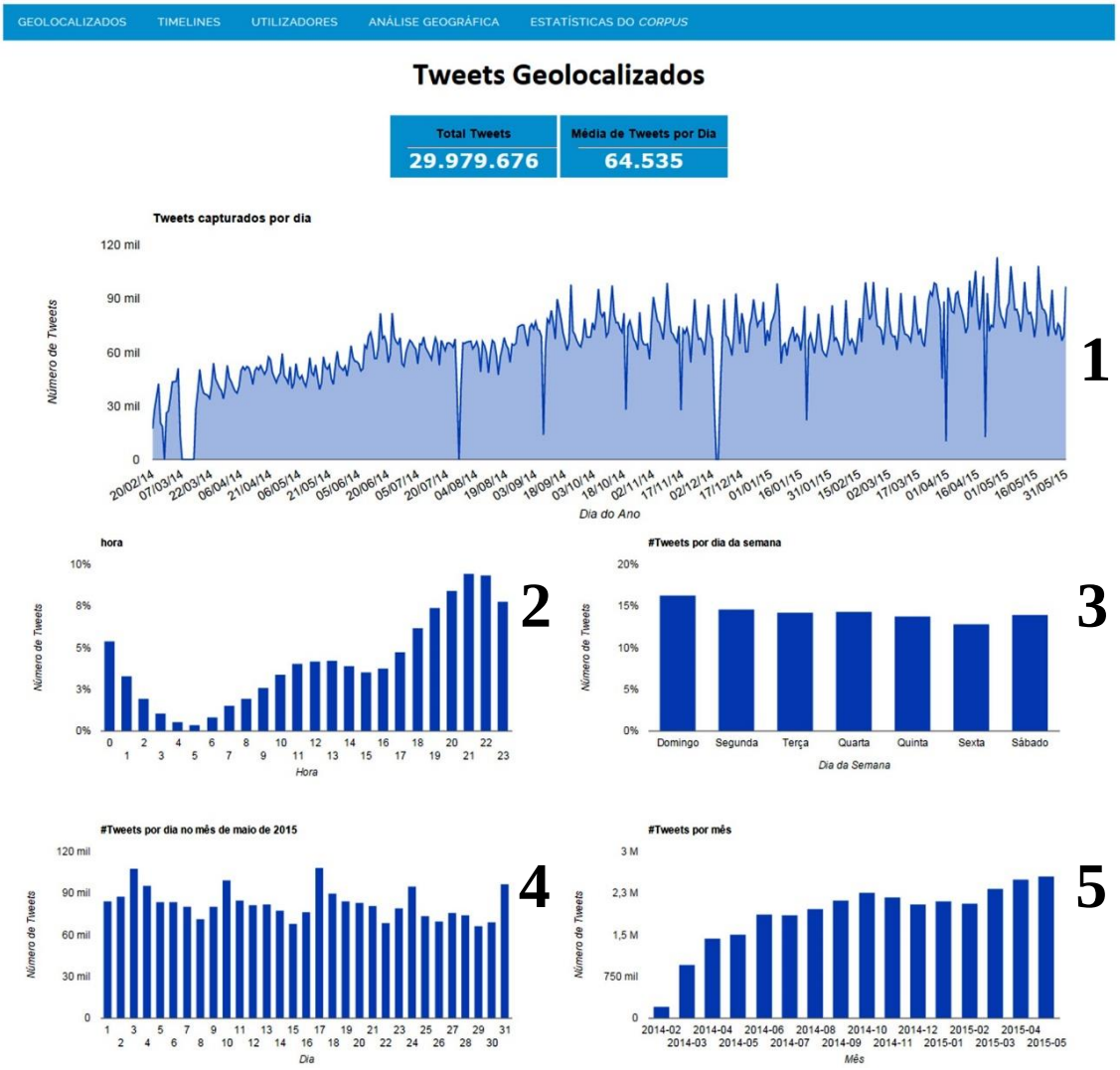

Figura 10: Dashboard para visualização de indicadores relativos aos dados recolhidos.

O processo de recolha de tweets foi iniciado a 20 de fevereiro de 2014 e terminou a 31 de maio de 2015. Na Figura 11 é apresentado o número de tweets geolocalizados recolhido por dia no período de fevereiro de 2014 a maio de 2015 observando-se um aumento do número de tweets capturados ao longo desse período. Os dias com cor branca deveram-se a problemas pontuais na recolha dos tweets. Neste período foram recolhidos aproximadamente 30 milhões de tweets geolocalizados com uma média diária de aproximadamente 66 mil tweets.

Pelo procedimento descrito na Secção 3 foram integrados em média 232 novos utilizadores por dia num total aproximado de 105 mil utilizadores. Analisando o gráfico da Figura 12 observa-se que o número de novos utilizadores por dia manteve-se constante ao longo de todo o período.

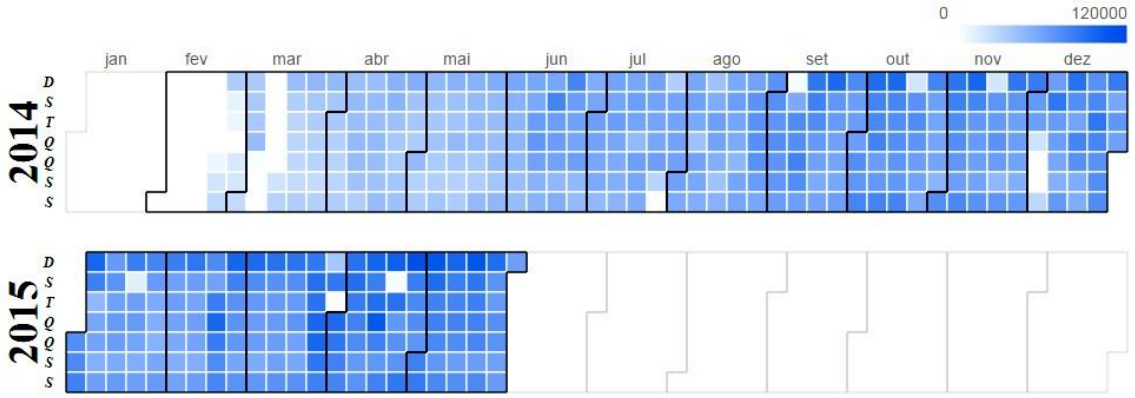

Figura 11: Distribuição da recolha de tweets geolocalizados por dia.

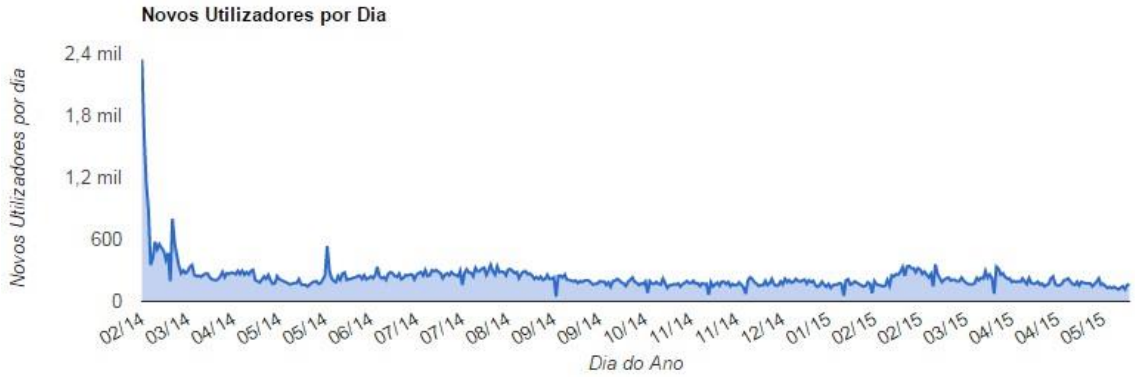

Figura 12: Novos utilizadores por dia.

Da execução do procedimento de expansão da base de dados descrito na Secção 4, resultou a recolha de cerca de 260 milhões de tweets. De entre os tweets armazenados em resultado da leitura das timelines cerca de 246 milhões correspondem ao intervalo de fevereiro de 2014 a maio de 2015, o que representa um acréscimo médio de cerca de 8 vezes relativamente ao número de tweets recolhidos do fluxo da Streaming API. Na Figura 13 apresenta-se a relação entre a quantidade de tweets geolocalizados e os tweets lidos das timelines. É notório o acréscimo de tweets recolhidos pelo mecanismo apresentado neste artigo, por comparação com volume de tweets disponibilizados pelo Twitter na Streaming API. A linha "paralela" ao eixo das abcissas representa o número de tweets geolocalizados recolhidos por dia. As restantes linhas representam a evolução do volume de tweets lidos das timelines em determinado instante, indicado pela data de cada uma das linhas na

legenda da Figura 13. Cada uma das linhas resulta de uma iteração completa para a recolha da totalidade ou atualização da timeline dos utilizadores registados na base de dados.

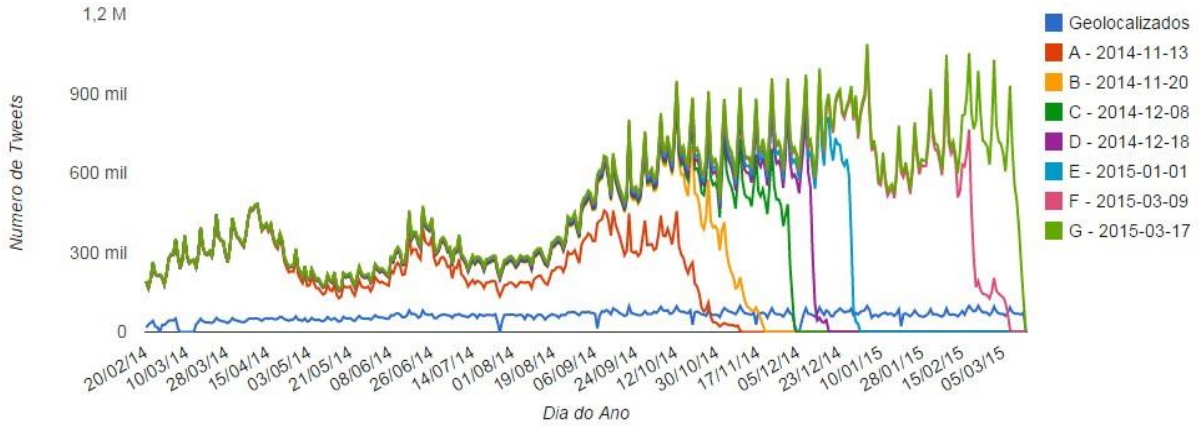

Figura 13: Comparação do volume de tweets obtidos da Streaming API e da REST API do Twitter.

Ainda relativamente à Figura 13 verifica-se que os máximos locais observados na oscilação da linha G da parte direita do gráfico correspondem aos primeiros dias da semana, nomeadamente aos dias de Domingo e Segunda-Feira e, por conseguinte, os mínimos locais correspondem aos dias do final da semana mais concretamente a Sexta-Feira. O dia 4 de janeiro de 2015 foi o dia para o qual se recolheram mais tweets pelo processo apresentado, com o total de 1.069.851 tweets enquanto que no mesmo dia foram recolhidos somente 98.366 tweets da Streaming API, representando um acréscimo de sensivelmente 11 vezes em relação ao fluxo de tweets geolocalizados.

7. CONCLUSÕES E TRABALHO FUTURO

Neste artigo é apresentado um Sistema de Informação que permite o desenvolvimento de uma base de dados de tweets escritos em Português Europeu. O mecanismo apresentado tenderá a armazenar grande parte dos tweets produzidos pela comunidade Portuguesa do Twitter dada a capacidade de identificar os utilizadores portugueses e de recolher continuamente as mensagens mais recentes produzidas pelos mesmos, tendo a capacidade de integrar continuamente novos utilizadores. A arquitetura desenvolvida permite a recolha de cerca de 8 vezes mais informação do que a disponibilizada em tempo real pelo Twitter no fluxo da Streaming API. A REST API implementada para acesso aos dados e visualização de estatísticas sobre os mesmos permite que a informação contida na base de dados possa ser partilhada e analisada em estudos das mais diversas áreas do conhecimento.

Na continuação deste trabalho tem-se como principal objetivo permitir a utilização do Sistema de Informação pelo método desenvolvido por Rosa *et al.* [2014] como fonte de dados para identificação e análise de tópicos, assim como dos utilizadores mais importantes relativamente a esses tópicos. Além da informação que este método poderá obter do conjunto de dados recolhido irá permitir a validação da arquitetura proposta. Também está previsto o enriquecimento da informação obtida de cada utilizador pela adição da idade e género (masculino ou feminino) correspondente a cada utilizador, através do método proposto por Vicente *et al.* [2015]. Para tal, será necessária a implementação de um *endpoint* na REST API que possibilite a alteração e adição de dados na base de dados MongoDB.

8. AGRADECIMENTOS

Este trabalho foi financiado com fundos nacionais através da FCT – Fundação para a Ciência e a Tecnologia pelo projeto PTDC/IVC-ESCT/4919/2012 (MISNIS) e fundos com referência UID/CEC/50021/2013.

REFERÊNCIAS

Anderson, K., e Schram, A., "Design and implementation of a data analytics infrastructure in support of crisis informatics research.", ICSE, ACM, (2011), 844-847.

Boyd, D. M. e Ellison, N. B., "Social network sites: definition, history, and scholarship". Journal of Computer-Mediated Communication, (2007), 210–230.

Brogueira, G., Batista, F., e Carvalho, J. P., Arquitetura e Desenvolvimento de um Repositório de Tweets em Português Europeu, 5ª Jornadas de Informática da Universidade de Évora, JIUE'15, (2015), Universidade de Évora.

Culotta, A., "Towards Detecting Influenza Epidemics by Analyzing Twitter Messages", in Proceedings of the First Workshop on Social Media Analytics ACM, (2010), 115-122.

Fielding, R., Architectural styles and the design of network-based software architectures, PhD thesis, (2000), Dept. of Information and Computer Science, University of California.

Gerber , M., "Predicting crime using Twitter and kernel density estimation", Decision Support Systems, (2014), 115 – 125.

Kumar, S., Morstatter, F., e Liu, H., "Twitter Data Analytics". Springer, (2013a).

Kumar, S., Morstatter, F., Zafarani, R., e Liu, H., "Whom Should I Follow?: Identifying Relevant Users During Crises", Proceedings of the 24th ACM Conference on Hypertext and Social Media, ACM, (2013b), 139-147.

Lachlan, K., Spence, P., Lin, X., "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 (2014), 554-559.

Laudon, K. C. e Laudon, J. P., "Sistemas de informação: com Internet", LTC Editora, (1999).

Marcus, A., Bernstein, M., Badar, O., Karger, D., Madden, S., e Miller, R., "Twitinfo: Aggregating and Visualizing Microblogs for Event Exploration", in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, (2011), 227-236.

Mendoza, M., Poblete, B. e Castillo, C., “Twitter Under Crisis: Can We Trust What We RT? “Proceedings of the First Workshop on Social Media Analytics, ACM, (2010), 71-79.

Oussalah, M., Bhat, F., Challis, K., e Schnier , T., “A software architecture for Twitter collection, search and geolocation services”, Knowledge-Based Systems (2013), 105-120.

Paul, Michael J. e Dredze, Mark, “You Are What You Tweet: Analyzing Twitter for Public Health”. ICWSM, The AAAI Press, (2011).

Perera, R.D.W., Anand, S., Subbalakshmi, K.P., e Chandramouli, R., “Twitter analytics: Architecture, tools and analysis”, in MILITARY COMMUNICATIONS CONFERENCE 2010 - MILCOM 2010 (2010), 2186-2191.

Qu, Y., Huang, C., Zhang, P., e Zhang, J., “Microblogging After a Major Disaster in China: A Case Study of the 2010 Yushu Earthquake”, Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work , ACM, (2011), 25-34.

Rosa, H., Carvalho, J. P., e Batista, F., “Detecting a Tweet’s Topic within a Large Number of Portuguese Twitter Trends”. In Pereira, M. J. V., Leal, J. P., and Simões, A., editors, 3rd Symposium on Languages, Applications and Technologies, volume 38 of OpenAccess Series in Informatics (OASIcs), Dagstuhl, Germany, (2104), 185–199.

Sakaki, T., Okazaki, M. e Matsuo, Y., “Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors”, Proceedings of the 19th International Conference on World Wide Web, ACM (2010), 851-860.

Santos, J. C. e Matos, S., “Predicting flu incidence from Portuguese tweets”, In Proceedings of the International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO 2013, Granada, Spain, (2013), 11-18.

Santos, J. C. e Matos, S., “Analysing twitter and web queries for flu trend prediction”, Theoretical Biology and Medical Modelling, (2014), 11(1).

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

Twitter, Inc., https://about.twitter.com/company, (2015) (acedido em 14-05-2015).

Twitter, Blog, “Introducing new metadata for Twitter”, https://blog.twitter.com/2013/introducing-new-metadata-for-tweets, (2013) (acedido em 14-05-2015).

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

Widener, M., e Wenwen, L., “Using geolocated Twitter data to monitor the prevalence of healthy and unhealthy food references across the {US}”, Applied Geography 54, (2014), 189-197.