UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
Programa de Pós-Graduação em Educação
Hércules Batista de Oliveira

FRAMEWORK ORÁCULO:
Camada de Coleta e Mineração de Textos para o Twitter

Diamantina
2019

Hércules Batista de Oliveira

FRAMEWORK ORÁCULO:
Camada de Coleta e Mineração de Textos para o Twitter

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Educação, como parte dos requisitos exigidos para a obtenção do título de Mestre em Educação.

Orientador: Marcus Vinícius Carvalho Guelpeli

Diamantina
2019

Elaborado com os dados fornecidos pelo(a) autor(a).

O48f Oliveira, Hércules Batista de.
Framework Oráculo: camada de coleta e mineração de textos para o Twitter /Hércules Batista de Oliveira, 2019.
73 p. : il.

Orientador: Marcus Vinícius Carvalho Guelpeli

Dissertação (Mestrado – Programa de Pós-Graduação em Educação) - Universidade Federal dos Vales do Jequitinhonha e Mucuri, Diamantina, 2019.

1. Coleta de textos. 2. Mineração de textos. 3. Twitter. I. Guelpeli, Marcus Vinícius Carvalho. II. Título. III. Universidade Federal dos Vales do Jequitinhonha e Mucuri.

CDD 006.312

Ficha Catalográfica – Sistema de Bibliotecas/UFVJM
Bibliotecária: Jullyele Hubner Costa – CRB6/2972

Hércules Batista de Oliveira

FRAMEWORK ORÁCULO:
Camada de Coleta e Mineração de Textos para o Twitter

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Educação, como parte dos requisitos exigidos para a obtenção do título de Mestre em Educação.

Orientador: Prof. Dr. Marcus Vinícius Carvalho Guelpeli

Data de aprovação 08/11/19.

Prof. Dr. Marcus Vinícius Carvalho Guelpeli
Departamento de Computação – UFVJM

Profa. Dra. Alexandre Ramos Fonseca
Instituto de Ciência e Tecnologia – UFVJM

Prof. Dr. Maria Lúcia Bento Villela
Departamento de Computação – UFVJM

Renato Dourado Maia
Departamento de Ciências da Computação – UNIMONTES

Diamantina

MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

DIAMANTINA – MINAS GERAIS

PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

ATESTADO DE DEFESA POR VIDEOCONFERÊNCIA

Atesto para os devidos fins que no dia 08 de novembro de 2019, às 10h, nas dependências da UFVJM – em Diamantina , foi realizada a defesa de dissertação do discente Hércules Batista de Oliveira com o trabalho intitulado ***"FRAMEWORK ORÁCULO: Camada de Coleta e Mineração de Textos para o Twitter"***, no Programa de Pós-graduação em Educação.

Na qualidade de presidente da banca, atesto que o Prof. Dr. Renato Dourado Maia docente da Universidade Estadual de Montes Claros - Unimontes, participou através de videoconferência.

Em virtude da participação remota do membro da banca acima indicado, eu, Prof. Dr. Marcus Vinicius Carvalho Guelpeli, enquanto servidor público, no gozo de fé pública, assino no lugar desse na Ata de Defesa e na Folha de Aprovação da referida defesa.

Por ser verdade, dou fé e assino o presente atestado.

Diamantina, 08 de novembro de 2018.

Prof. Dr. Marcus Vinicius Carvalho Guelpeli

Presidente da Banca

Dedico este trabalho a minha Dinha, Júlia Volponi (*In Memoriam*), que de muitas formas contribuiu para minha formação e educação.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela oportunidade e por me abençoar com saúde e foco para realizar este estudo.

À minha esposa, pelo suporte, amor e incentivo diário.

Aos meus familiares e amigos pelo apoio, compreensão e paciência durante o desenvolvimento desta pesquisa.

Aos professores e colegas do PPGED, em especial os membros do grupo de pesquisa MTPLNAM, pelo companheirismo.

“O que sabemos é uma gota; o que ignoramos é um oceano.” (Isaac Newton)

RESUMO

As redes sociais *online* constituem um importante espaço de convivência para a população, com aplicações em comunicação, diversão, propaganda, mobilização social e comunitária. Os dados compartilhados em tais redes constituem fonte de pesquisa de diversos trabalhos que buscam analisar as interações dos seus usuários. Para que se possam analisar os dados coletados de maneira eficiente, devido ao grande volume produzido por essas redes, faz-se necessária a utilização de técnicas de mineração de textos. Nesse processo de mineração de texto apresenta-se o desafio da falta de acesso direto aos dados das redes sociais *online*, o que torna necessário utilizar ferramentas especializadas para realizar a coleta de dados. O *framework* Oráculo, em desenvolvimento pelo grupo de pesquisa MTPLNAM, é formado por diferentes camadas. Nesta pesquisa foi desenvolvida a camada de coleta e mineração de textos, que aplica diferentes técnicas e algoritmos para coletar texto do Twitter, buscando contornar as limitações impostas pela API disponibilizada por ele, e integra um minerador de textos para analisar as coletas realizadas. Essa camada do *framework* dispõe de interface web, permitindo a utilização por pesquisadores não familiarizados com a área de computação. Foram realizados testes comparativos de desempenho entre o *framework* Oráculo e outra ferramenta semelhante de coleta e mineração de textos, o DMI-TCAT. Os resultados desses testes apontam que o *framework* Oráculo teve desempenho superior ao DMI-TCAT em número de *tweets* coletados nos cenários analisados. Testes estatísticos foram executados e validaram os resultados dos testes de desempenho.

Palavras-chave: Coleta de textos. Mineração de textos. Twitter.

ABSTRACT

Online social networks are an important social space for the population, with applications in communication, entertainment, advertising, social and community mobilization. The data shared in such networks is a source of research for several works that seek to analyze the interactions of their users. In order to analyze the collected data efficiently, due to the large volume produced by these networks, it is necessary to use text mining techniques. This text mining process presents the challenge of the lack of direct access to data from online social networks, which makes it necessary to use specialized tools to perform data collection. The Oracle framework, under development by the MTPLNAM research group, is made up of different layers. This research developed the text collection and mining layer, which applies different techniques and algorithms to collect text from Twitter, seeking to circumvent the limitations imposed by the API provided by Twitter, and integrates a text miner to analyze the collections made. This layer of the framework has web interface, allowing the use by researchers unfamiliar with the area of computing. Comparative performance tests were performed between the Oracle framework and another similar text collection and mining tool, DMI-TCAT. The results of these tests indicate that the Oracle framework outperformed the DMI-TCAT in the number of tweets collected in the analyzed scenarios. Statistical tests were performed and validated the results of the performance tests.

Keywords: Text collect. Text mining. Twitter.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BSON	Binary JSON
CSV	Comma-separated values
Decom	Departamento de Computação
DMI-TCAT	Digital Methods Initiative Twitter Capture and Analysis Toolset
EUA	Estados Unidos da América
GNU	General Public License
HTTP	Hypertext Transfer Protocol
IHC	Design de Interação Humano-Computador
JSON	JavaScript Object Notation
Mbps	Megabits por segundo
MTPLNAM	Grupo de Pesquisa em Mineração de Textos, Processamento de Linguagem Natural e Aprendizagem de Máquina
UFVJM	Universidade Federal dos Vales do Jequitinhonha e Mucuri
yTk	yourTwapperKeeper

SUMÁRIO

1 INTRODUÇÃO

As redes sociais *online* ocupam relevante espaço na vida dos seus usuários. Com aplicações em comunicação, diversão, propaganda, mobilização social e comunitária, seus bilhões de usuários distribuídos pelo mundo dedicam cada vez mais tempo à sua utilização (SILVIUS; KAVALIAUSKAITE, 2014). Segundo Recuero (2014), redes sociais *online* representam um novo e complexo universo de fenômenos comunicativos, sociais e discursivos. Por meio do registro dessas dinâmicas sociais e do seu acesso pelos pesquisadores, é possível que interações e conversações sejam mapeadas e estudadas em larga escala.

Esses sistemas, em geral, permitem a produção descentralizada de conteúdo, a interação e a mobilização de pessoas, criando grupos com interesses em comum, facilitando assim sua estruturação e organização (FRANÇA; OLIVEIRA, 2014). Essas características, aliadas ao grande número de usuários, fazem com que o volume de dados e a velocidade de criação sejam expressivos.

São necessárias técnicas computacionais apuradas que permitam a análise desse imenso volume de dados, para que se possa, a partir deles, obter conhecimento útil. Dentre as técnicas existentes, destaca-se a mineração de textos ou *text mining*, que pode ser definida como o uso de técnicas, baseadas em modelos, para encontrar padrões, sumarizar dados ou realizar predições, e assim extrair conhecimento de textos (TAN *et al.*, 1999). A mineração de textos abrange as etapas de coleta, pré-processamento, indexação, mineração (extração do conhecimento), visualização e análise (MATHIAK; ECKSTEIN, 2004). Como não há acesso direto aos dados das redes sociais *online*, são necessárias ferramentas especializadas para realizar o processo de coleta.

O Twitter foi escolhido como fonte dos dados desta pesquisa, uma vez que possui como objetivo primário o compartilhamento de textos, os *tweets*. Segundo Ausserhofer e Maireder (2013), o Twitter é uma rede social que rompeu as barreiras para a participação e o debate devido à natureza pública da comunicação desenvolvida em seu espaço. Sua relevância social levou a estudos em diferentes áreas do conhecimento, como educação (TANG; HEW, 2017)(KIM; HWANG; RHO, 2018), movimentos sociais (MARADEI, 2018) (FRANÇA; OLIVEIRA, 2014)(BRUNS; BURGESS, 2011), eleições e política (RECUERO; ZAGO; SOARES, 2017)(GABARDO *et al.*, 2019), saúde (PRUSS *et al.*, 2019)(STEFANIDIS *et al.*, 2017), entre outros.

Embora existam ferramentas de coleta e análise de *tweets* licenciadas como *software* livre, como o *Digital Methods Initiative Twitter Capture and Analysis Toolset* - DMI-TCAT (BRUNS *et al.*, 2014) e *yourTwapperKeeper* - yTk (BRUNS; LIANG, 2012), em geral, essas ferramentas utilizam da A*pplication Programming Interface* - API do Twitter para realizar as suas coletas. Para evitar abusos essa API impõe limites de acesso e restrições aos dados que podem ser recuperados (TWITTER, 2019c); assim, as coletas de textos realizadas com estas ferramentas estão restritas aos limites impostos por esta rede social. Estes limites estão relacionados ao

número de textos que podem ser retornados em uma requisição, número de requisições dentro de um determinado espaço de tempo e a capacidade de recuperar textos históricos.

1.1 Problema

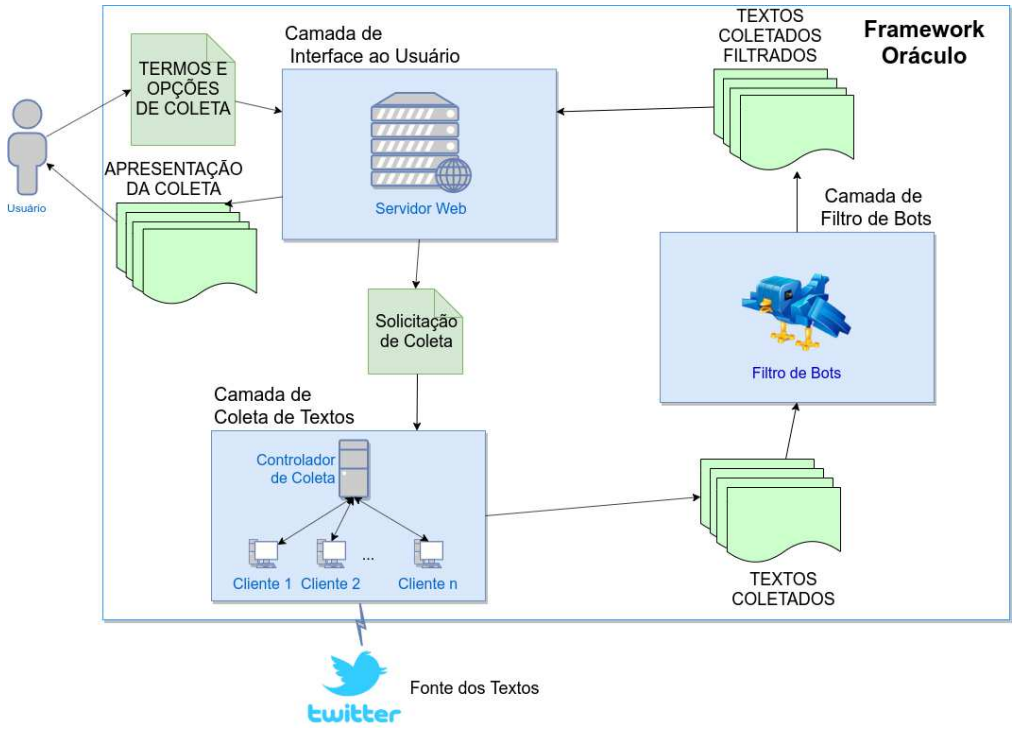
Para que seja possível analisar e extrair conhecimento do grande volume de textos obtidos de uma rede social *online* é necessário utilizar de sistemas de mineração de textos. Como não há acesso direto aos dados, gerados por essas redes, consequentemente há a necessidade de coletar os textos com a utilização de sistemas especializados em coletas.

Este trabalho elenca a seguinte questão de pesquisa: De que forma o desenvolvimento de uma ferramenta de coleta e mineração de textos no Twitter, que disponibilize diferentes algoritmos e técnicas de coleta, impacta no processo de coleta e análise dos textos?

Esta pesquisa desenvolveu essa ferramenta na forma da Camada de Coleta e Mineração Textos do *framework* Oráculo, que encontra-se em construção pelo Grupo de Pesquisa em Mineração de Textos, Processamento de Linguagem Natural e Aprendizagem de Máquina - MTPLNAM, do Departamento de Computação (Decom) da Universidade Federal dos Vales do Jequitinhonha e Mucuri - UFVJM. As demais camadas do *framework*, Filtro de *Bots* e Interface do Usuário, são objeto de trabalho de outros pesquisadores do MTPLNAM.

A modelagem arquitetura de funcionamento do *framework* Oráculo, apresentando os fluxos, funcionalidades e interações entre as camadas que o compõem é representada pela Figura 1.

Figura 1: Arquitetura do *Framework* Oráculo



Fonte: Próprio autor

A interação entre as diferentes camadas formam o *framework* Oráculo. A finalidade e as funcionalidades de cada camada são:

- **Camada de Coleta e Mineração de Textos**: Realiza a recuperação de textos junto ao Twitter, ao aplicar diferentes técnicas e algoritmos de coleta, objetivando uma coleta mais eficiente e eficaz, ao contornar limitações comuns à outras ferramentas com a mesma finalidade. Nesta camada também estão presentes funcionalidades de mineração de textos sobre os dados coletados.
- **Camada de Filtro de *Bots***: Camada responsável por realizar a filtragem nos dados coletados, a partir de um determinado conjunto de regras, e identificar quais as postagens são oriundas de *bots* e quais não são, possibilitando isolar as primeiras de forma que o usuário final possa desconsiderá-las de suas pesquisas.
- **Camada da Interface ao Usuário**: Realiza a comunicação entre Usuário e *framework*, através de uma plataforma *web* interativa. Esta possui usabilidade desenvolvida a partir de propostas de Design de Interação Humano-Computador - IHC e possui o objetivo de se preocupar com a experiência do usuário e lhe conferir alta qualidade de uso, focando no modo e na forma que se dará a comunicação entre o usuário e a interface do sistema.

1.2 Objetivos

Nesse contexto, o objetivo geral deste trabalho é desenvolver a camada de coleta e mineração de textos no Twitter do *framework* Oráculo, de forma que disponibilize diferentes algoritmos e técnicas de coleta para apoiar pesquisas nessa rede social *online*.

Além do objetivo geral apresentado são elencados os seguintes objetivos específicos:

- Analisar quantitativamente o desempenho em coletas no Twitter de cada algoritmo e técnica disponível da ferramenta desenvolvida;
- Comparar o desempenho dos algoritmos e técnicas de coleta disponíveis nessa ferramenta com outra ferramenta semelhante, o DMI-TCAT;
- Analisar os pontos positivos e negativos de cada algoritmo e técnica disponível nessa ferramenta;
- Disponibilizar uma interface para a camada de coleta e mineração de textos do *framework* Oráculo, como forma de permitir a sua utilização, independente das outras camadas do *framework*, tornando assim a ferramenta funcional e disponível aos pesquisadores;
- Analisar a interferência da localização geográfica e da conexão de acesso à internet quanto ao desempenho da coleta no Twitter;
- Realizar testes estatísticos para validação dos resultados comparativos das coletas efetuadas com o *framework* Oráculo e com o DMI-TCAT;

1.3 Justificativa

Na literatura é possível identificar diversas pesquisas recentes e relevantes que utilizam um número de *tweets* restrito como amostra em suas discussões: 900 *tweets* (MARADEI,

2018), 5000 *tweets* por dia (GABARDO *et al.*, 2019), 54611 *tweets* (RECUERO; ZAGO; SOARES, 2017). Em contraponto, pesquisas que utilizaram de ferramentas proprietárias e comerciais, indisponíveis a todos os pesquisadores, para coletar seus *tweets* analisaram um número consideravelmente superior de *tweets*: 6 milhões de *tweets* (STEFANIDIS *et al.*, 2017) e 15 milhões de *tweets* (PRUSS *et al.*, 2019).

Assim, sugere-se que uma ferramenta de coleta que dispõe de recursos para contornar as limitações impostas pela API do Twitter possibilitará que pesquisas sejam realizadas utilizando amostras mais abrangentes. Essa ferramenta, ao possuir uma interface simples, disponível em uma plataforma *web*, pode auxiliar e potencializar essas e outras pesquisas ao permitir que usuários não familiarizados com a área de computação realizem suas coletas e análises de forma mais prática.

1.4 Contribuições

A principal contribuição desta pesquisa está relacionada a disponibilizar a camada de coleta e mineração de textos do *framework* Oráculo, na forma de um *software* livre para que pesquisadores de diferentes áreas do conhecimento possam utilizar e adaptar essa ferramenta ao seu domínio de pesquisa. Essa ferramenta apresenta uma interface *web* simples e funcional, o que auxilia pesquisadores que não sejam familiarizados com a área de ciência da computação a realizarem coletas e mineração de textos no Twitter.

Aliada à camada do *framework*, este trabalho também apresenta uma análise do desempenho em coletas dessa ferramenta. Ao disponibilizar uma arquitetura de coleta distribuída e diferentes algoritmos e técnicas, esta ferramenta busca realizar a coleta no Twitter de forma inovadora para contornar as limitações de número de requisições imposta pela API do Twitter.

Os resultados dos testes comparativos entre os diferentes algoritmos de coleta disponíveis no *framework* Oráculo poderão embasar o planejamento de pesquisadores na utilização dessa ferramenta. Essa pesquisa apresenta ainda um estudo da relação entre a localização geográfica do coletor, velocidade de conexão de acesso à internet do nó coletor e os resultados obtidos no processo de coleta.

1.5 Organização deste trabalho

Este trabalho está organizado de forma que no capítulo 2 são apresentadas as referências bibliográficas que embasam esta pesquisa, com foco nas redes sociais *online*, principalmente o Twitter, fonte desta pesquisa, e as formas e ferramentas de acesso utilizadas. No capítulo 3 são apresentados os procedimentos metodológicos empregados no desenvolvimento da camada de coleta e mineração de textos do *framework* Oráculo e a forma como dados foram analisados. O capítulo 4 apresenta os resultados e as discussões traçadas. E, por fim, o capítulo 5 apresenta as considerações finais.

2 REFERENCIAL TEÓRICO

Neste capítulo, são abordados os pressupostos teóricos que alicerçam esta pesquisa, com foco na mineração de textos, rede social *online*, Twitter, sua API, os algoritmos e as ferramentas para coleta utilizadas neste estudo.

2.1 Framework

Frameworks podem ser definidos como um arcabouço de software (MINETTO, 2007), o que pode ser detalhado como uma aplicação reusável e semi-completa capaz de ser especializada para produzir outras aplicações personalizadas. Ao utilizar-se das técnicas usuais de orientação a objetos, organizam-se as classes em bibliotecas. Já em *frameworks*, essa organização é feita em unidades particulares de negócios e domínios de aplicações. Destacam-se como benefícios do desenvolvimento utilizando a abordagem de *frameworks*: modularidade; reusabilidade; extensibilidade e inversão de controle (FAYAD; SCHMIDT, 1997).

A modularidade é implementada por intermédio da definição de interfaces e o encapsulamento de métodos. A reusabilidade é percebida no fato de possibilitar a criação de novas aplicações através de componentes genéricos, o que confere um ganho de produtividade no desenvolvimento, e incremento da qualidade, desempenho e confiabilidade do software. A extensibilidade está relacionada à capacidade de extensão de funcionalidades a partir de uma estrutura inicialmente projetada, sem causar impactos negativos nessa estrutura. A inversão de controle refere-se à capacidade do *framework* em responder a eventos externos, mantendo o controle da execução da aplicação (FAYAD; SCHMIDT, 1997).

A utilização de *framework* no desenvolvimento de *softwares* científicos livres é bastante aplicável, uma vez que o desenvolvimento cooperativo requer a incorporação de novos requisitos, a reusabilidade de código e a programação baseada em métodos de interfaces bem definidos, o que garante a integração entre os diferentes módulos do *software* e sua extensibilidade de forma a agregar novas funcionalidades. Essas características permitem que diversos desenvolvedores possam utilizar um *framework* no seu domínio de pesquisa (BITTENCOURT; OSÓRIO, 2001).

2.2 Mineração de Textos

A Mineração de Textos (*text mining*), Mineração de Dados Textuais ou Descoberta de Conhecimento em Textos é uma abordagem ao processamento de grandes bases de dados textuais com o objetivo de extrair informação relevante e obter conhecimento implícito e útil (SOARES, 2013).

Na literatura é possível encontrar outras definições de mineração de textos, dentre as quais cabem ser destacadas:

- Processo de extração de padrões de interesse não triviais ou conhecimento, a partir de textos não estruturados, ou ainda, como um conjunto de técnicas e processos que se prestam a descobrir conhecimento inovador nos textos (LOPES, 2004);
- Consiste na utilização de técnicas de reconhecimento de padrões e heurística, extração de informações a partir de textos livres, com base nos elementos nele contidos, por exemplo, as palavras-chave (WIVES, 2002);
- Processo capaz de fazer emergir de grande massa documental de texto verbal informações úteis de modo que sejam exploradas automaticamente, ou seja, extrair conhecimentos de documentos semanticamente próximos, bem como pesquisar a relação entre entidades textuais (termos) ou entre documentos e descobrir tendências de conceitos que se espalham nos documentos (FELDMAN; SANGER, 2007).
- Extração de informação de textos usando os princípios da linguística computacional (SULLIVAN, 2000);
- Aplicação de algoritmos computacionais que processam textos e identificam informações úteis e implícitas, as quais não poderiam ser recuperadas utilizando outros métodos de consulta, devido à forma não estruturada que estão armazenadas (MORAIS; AMBRÓSIO, 2007).

Embora existam essas diversas definições, é possível concluir que Mineração de Textos, de maneira análoga à Mineração de Dados, busca extrair informação útil de bases de dados através da identificação e exploração de padrões interessantes (SOARES, 2013).

No entanto, é preciso destacar a diferença entre a Mineração de Dados e a de Textos. Mineração de Textos é um processo de obtenção de conhecimento a partir de bases de dados textuais, ou seja, documentos em linguagem natural, e que, portanto, são não estruturados. Em Mineração de Dados, a obtenção de conhecimento ocorre em bases de dados estruturadas, geralmente armazenadas em Sistemas Gerenciadores de Bancos de Dados (SOARES, 2013).

O processo de mineração de textos abrange as etapas de coleta, pré-processamento, indexação, mineração (extração do conhecimento), visualização e análise (MATHIAK; ECKSTEIN, 2004). A Figura 2 representa esse processo e relaciona cada uma das etapas com as ações desenvolvidas (ARANHA, 2007).

Segundo Soares (2013), a definição de cada etapa de mineração de textos é:

- **Coleta**: primeira etapa a ser realizada, tem como objetivo a compor a base textual de trabalho, em geral organizada em documentos, elemento básico de qualquer processo de Mineração de Textos.
- **Pré-processamento**: tem por objetivo preparar os documentos coletados de maneira a obter uma forma de representá-los estruturadamente.
- **Indexação**: é responsável por criar índices com o objetivo de dar maior rapidez e agilidade à recuperação dos documentos e seus termos.

Figura 2: Etapas do processo de Mineração de Texto

Fonte: (ARANHA, 2007) (Adaptado)

- **Mineração**: objetiva a extração de conhecimento ao aplicar algoritmos de Aprendizagem de Máquina e de Estatística, com a finalidade de descobrir padrões úteis e desconhecidos, presentes nos documentos.
- **Análise**: etapa em que o usuário realiza a avaliação e interpretação de todo o conhecimento obtido pelo processo.

Como o objeto de pesquisa deste trabalho é a mineração de textos em uma rede social *online*, o Twitter, há a necessidade de especial atenção à fase de coleta de dados, visto que não há acesso direto aos dados desses sistemas. Nesse cenário, esta pesquisa buscou desenvolver uma camada para o *framework* Oráculo dedicada a coleta no Twitter e que integrasse as demais etapas do processo de mineração de textos em uma única ferramenta.

2.3 Redes Sociais *Online*

Benevenuto, Almeida e Silva (2011) definem redes sociais *online* como um serviço web que permite indivíduos construam perfis públicos ou semipúblicos dentro de um sistema, de forma a articular uma lista de outros usuários (amigos, seguidores) com os quais compartilham conexões, para visualizar ou interagir com suas listas de conexões e outras listas feitas por outros usuários no sistema.

A aplicação prática dessa definição é detalhada por Torres (2018), para quem redes sociais *online* visam reunir pessoas, membros, que uma vez inscritos, podem expor seu perfil com dados, como fotos, textos, mensagens e vídeos, além de interagir com outros membros, criando listas de amigos, comunidades, grupos e fóruns, ou até escrevendo um *blog* ou *microblog* (textos curtos).

Verifica-se a existência de diversas redes sociais *online* disponíveis na web, sendo que cada uma tem seu objetivo primário, como o compartilhamento de fotos (Instagram),

compartilhamento de vídeos (Youtube), compartilhamento de textos curtos (Twitter), redes profissionais (Linkedin), rede de amigos (Facebook), entre outras (BENEVENUTO; ALMEIDA; SILVA, 2011).

Dentre as redes sociais disponíveis, o Twitter foi escolhido como fonte dos dados para esta pesquisa devido a ter como objeto primário o compartilhamento os textos, os quais podem ser minerados para extração de conhecimento. O fato do Twitter disponibilizar uma API para que desenvolvedores tenham o acesso, mesmo que limitado, a seus textos e usuários. Aliada a essa característica, o fato de que a maior parte das contas serem públicas e as informações circularem de forma aberta, com uma menor incidência de algoritmos que possam restringir o conteúdo exibido para os demais usuários (RECUERO; ZAGO; SOARES, 2017), também justifica a escolha.

2.4 Twitter

O Twitter é uma rede social *online* de textos curtos, caracterizando um microblog. Nessa rede, os usuários compartilham mensagens de até 280 caracteres que podem conter elementos alfanuméricos, fotos e vídeos, chamadas *tweets* (MURTHY, 2018). Esta rede foi lançada em 2006 e conta atualmente com cerca de 330 milhões de usuários ativos mensalmente, dos quais, 136 milhões a acessam em média diariamente (TWITTER, 2019a) e produzem em média 500 milhões de *tweets* por dia (SAEED *et al.*, 2019).

O Twitter é público, *multicast*, interativo e em rede, ou seja, a maioria das contas são abertas para consulta pública e o conteúdo é criado de forma descentralizada pelos seus usuários que interagem entre si, em suas postagens, através de marcações de usuário e *hash-tags*[1] (BRUNS; BURGESS, 2011). Assim, essa rede promove a democratização da mídia, autoprodução, determinismo tecnológico e interacionismo.

A proposta do Twitter, no tocante aos seus valores e seus princípios de segurança, aliada aos seus números, faz esta plataforma desempenhar importante papel na comunicação social (TWITTER, 2019b). Os exemplos de como mudou as práticas jornalísticas, ao dar voz ao jornalista cidadão, ou ao possibilitar um canal para circular informações sobre desastres naturais, são prova da relevância desse papel (MURTHY, 2018).

2.4.1 Twitter na Educação

As redes sociais *online* podem ser compreendidas como uma das formas de representação dos relacionamentos afetivos ou profissionais dos seus usuários, a qual é utilizada para compartilhamento em rede ou em comunidade de ideias, informações e interesses. Estas redes ou comunidades podem ser utilizadas para a Educação, contribuindo assim para o processo de ensino e aprendizagem. (LORENZO, 2013).

[1] Termo precedido por #, sem espaços, utilizado para marcar um determinado tópico de discussão (KWAK *et al.*, 2010)

As características do Twitter com relação à publicidade de suas informações e interação entre usuários e descentralização de produção de conteúdo possibilitam que toda e qualquer pessoa pode ser importante influenciadora da cultura globalizada (SANTAELLA; MORAIS, 2010). Esta visibilidade proporcionada pelo Twitter pode ser explorada para ações educativas com reflexos em outras esferas sociais (SANTANA; COUTO *et al.*, 2017).

Na literatura recente diversas pesquisas envolvem o Twitter e a Educação. Dentre essas, merecem destaque pesquisas relacionadas aos seguintes temas:

- Formas e benefícios do uso do Twitter na Educação (TANG; HEW, 2017) (DENKER *et al.*, 2018) (GUTIÉRREZ-MARTÍN; TORREGO-GONZÁLEZ, 2018);
- Twitter na identidade acadêmica formal (JORDAN, 2019);
- Twitter como uma plataforma de comunicação com alunos, professores, outras instituições e o público (JEONG; JALALI, 2019)(KIMMONS; VELETSIANOS; WOODWARD, 2017);
- Comunicação social e colaborativa para aprendizagem (KIM; HWANG; RHO, 2018);
- Acompanhamento de alunos no ensino em saúde (BOOTH, 2015).

Estas e outras pesquisas podem ser potencializadas por uma ferramenta como o *framework* Oráculo, que agrega a coleta de diversas formas e a mineração de textos no Twitter de forma integrada.

2.4.2 API do Twitter

Para que seja possível acessar a massa de textos produzidos pelo Twitter, a rede social disponibiliza a desenvolvedores um conjunto de procedimentos e padrões, ou *Application Programming Interface* - API, para que seja possível acessar os *tweets*, status e dados dos usuários (TWITTER, 2019c).

Visando coibir abusos e sobrecarga dos seus servidores, o Twitter impõe limitações ao acesso aos dados através da sua API. São disponibilizados três níveis de acesso diferentes nessa API: *Standard, Premium* e *Enterprise*. O Quadro 1 representa o comparativo entre as funcionalidades de cada versão (TWITTER, 2019c).

Quadro 1: Comparativo dos níveis de acesso da API

Funcionalidade	Standard	Premium	Enterprise
Publicar e Engajar	x		
Procurar Tweets: 7 dias	x		
Procurar Tweets: 30 dias		x	x
Procurar Tweets: Histórico Completo		x	x
Filtrar Tweets	x		x
Tweets de Exemplo	x		x
Tweetar em Lote			x
Mensagens Diretas	x		
Contas e usuários	x	x	x
Métricas			x
API de Anúncios	x		
Ferramentas de Publicação e SDKs	x		

Fonte: (TWITTER, 2019c)(Adaptado)

Verifica-se que as principais diferenças entre esses níveis de acesso estão relacionadas à quantidade de requisições mensais que podem ser realizadas e à capacidade de retroagir uma pesquisa. A versão *Standard* alcança *tweets* postados em até sete dias, a versão *Premium* e *Enterprise* variam entre 30 dias e histórico completo, dependendo do pacote selecionado. Além dessa restrição, também são limitados o número de requisições por segundo, por minuto e numa janela de 15 minutos (TWITTER, 2019c).

O nível *Standard* da API é gratuito e está disponível a todos os usuários da rede social. Os custos para acesso às versões pagas do Twitter iniciam em 99,00 dólares por mês e vão até milhares de dólares, dependendo do número de requisições feitos à API. O custo de acesso através da versão *Enterprise* é fornecido somente em contato direto com os vendedores do Twitter. A Tabela 1 apresenta o custo por requisição da API *Premium* do Twitter (TWITTER, 2019c):

Tabela 1: Quantidade *tweets* coletados nos testes para algoritmos retroativos

Número de Requisições de Histórico Completo	Valor Mensal
Até 100	US$ 99,00
Até 250	US$ 224,00
Até 500	US$ 399,00
Até 1.000	US$ 774,00
Até 2.500	US$ 1.899,00

Fonte: (TWITTER, 2019c) (Adaptado)

Por entender que os custos relativos à aquisição de acesso privilegiado à API podem inviabilizar muitas pesquisas, o *framework* Oráculo busca, por meio de diferentes algoritmos e técnicas, contornar as limitações da API na versão *Standard*, e, assim, potencializar diversas pesquisas que tenham os textos do Twitter como fonte.

A API do Twitter disponibiliza dois meios principais pelos quais os textos podem ser recuperados: a API *Search* e a API *Streaming*. Utilizando os recursos da API *Search*, é possível acessar os dados históricos dos *tweets* postados, também chamados de retroativos, relacionados a um conjunto de palavras pesquisadas. A API *Streaming* retorna os *tweets* em tempo real, ou seja, que foram postados enquanto a coleta está sendo executada, também chamada de *realtime*, realizando assim a filtragem da massa de textos da rede social a partir das palavras chaves selecionadas (TWITTER, 2019c).

2.4.3 Desafios da Coleta no Twitter

Bruns e Liang (2012) apontam que os desafios enfrentados no desenvolvimento de ferramentas para realização de coleta e análise do Twitter estão relacionados a três pontos: limitações da API do Twitter, escalabilidade e pontualidade.

No tocante à limitação da API do Twitter, destaca-se o controle exercido no acesso aos dados, com limites restritos de recuperação da informação por acesso e por conexão com a in-

ternet, originadas de um determinado endereço de IP ou chave de acesso. Solicitações recorrentes de um mesmo requisitante enfrentam limitações do número de consultas por determinado espaço de tempo. Outro desafio referente à API do Twitter configura-se com a limitação da versão gratuita a recuperar dados históricos limitados a sete dias (TWITTER, 2019c). Ao restringir o período e, por consequência, a amostra pesquisas podem ser inviabilizadas, por demandarem uma amostra mais abrangente. (BRUNS; LIANG, 2012).

O Twitter tem em média 500 milhões de *tweets* por dia, cerca de 5800 *tweets* por segundo (SAEED *et al.*, 2019). Quanto ao desafio referente à escalabilidade, os números do Twitter evidenciam a necessidade de alocar considerável quantidade de recursos computacionais, seja para armazenamento ou poder computacional para processamento de uma coleta nessa massa de textos.

A pontualidade está relacionada ao tempo de resposta a um determinado evento, que esteja ocorrendo no Twitter, ser coletado e analisado. Esse tratamento temporal requer, além de poder computacional, que as ferramentas estejam prontas para coletar os textos da rede social quando forem demandadas.

2.4.4 Ferramentas de Coleta no Twitter

O papel do Twitter como meio de comunicação social o torna interessante para pesquisadores de diversas áreas do conhecimento realizarem seus estudos. Para que se possa acessar seu conteúdo e recuperar informação, os pesquisadores utilizam ferramentas, muitas vezes personalizadas, que são desenvolvidas apenas para uma pesquisa ou grupo de pesquisas limitadas e depois seguem indisponíveis para outros pesquisadores. Tal indisponibilidade contribui negativamente para a replicabilidade de pesquisas e extrapolação para outros contextos semelhantes (BRUNS; LIANG, 2012).

Existem, no entanto, iniciativas e ferramentas de código aberto disponíveis para coleta de dados no Twitter, que permitem que pesquisadores as apliquem em suas próprias áreas de pesquisa, de forma a gerar conjuntos de dados comparáveis e estudos replicáveis. São exemplos o yourTwapperKeeper e DMI-TCAT.

O DMI-TCAT foi escolhido como ferramenta de referência para comparação do desempenho do *framework* Oráculo, por se tratar de uma ferramenta disponível e utilizada pela comunidade de pesquisa desde 2014.

Cabe destacar também o Scrapy, que apesar de não ser uma ferramenta de coleta de textos no Twitter, pode ser adaptado para tal finalidade. O Scrapy é um *framework* para *web crawling* que foi aplicado em um dos algoritmos disponíveis do Oráculo.

2.4.4.1 yourTwapperKeeper

O yourTwapperKeeper - yTk (BRUNS; LIANG, 2012) é uma ferramenta de código aberto, licenciada através da GNU - *General Public License* (Licença Pública Geral GNU). Esta ferramenta utiliza a API do Twitter para realizar a coleta dos *tweets*, unindo as funcionalidades

da API *Search* e Streaming, estando assim sujeita às limitações impostas pela API. Desenvolvida em PHP e MySQL, essa ferramenta exporta os *tweets* coletados em formato *Comma-separated values* - CSV, permitindo, portanto, que os dados sejam acessados através de editores de textos e de planilhas eletrônicas.

2.4.4.2 DMI-TCAT

O *Digital Methods Initiative Twitter Capture and Analysis Toolset* - DMI-TCAT é uma ferramenta desenvolvida para coletar e analisar *tweets* com finalidade acadêmica, inicialmente, em ciências humanas e sociais (BRUNS *et al.*, 2014).

Além dos textos presentes nos *tweets*, o DMI-TCAT consegue recuperar os metadados correspondentes, como menções, respostas, localização e urls, além de dados relacionais aos usuários dos *tweets* coletados, como, seguidores e *tweets* publicados.

No seu desenvolvimento, foi utilizada a linguagem de programação PHP e o banco de dados MySQL, sendo executado em um servidor web local. Para acessar os textos no Twitter, utiliza-se da API Search, para *tweets* retroativos, e Streaming, para *tweets realtime*, dependendo da opção do usuário. Ao utilizar-se da API disponibilizada pelo Twitter, essa ferramenta fica restrita aos limites por ela estabelecidos.

2.4.4.3 Scrapy

Web crawlers são sistemas que navegam e baixam automaticamente páginas da *Web* seguindo os *hiperlinks* de maneira metódica e automatizada. Os *web crawlers* geralmente são conhecido por coletar páginas da *Web*, mas quando um *crawler* também pode realizar a extração de dados durante o rastreamento, ele é chamado de *web scraper* (KHALIL; FAKIR, 2017). *Web scraping* pode ser definido como o processo de extrair informações de uma página web ao processar as *tags* de hipertexto e recuperar informação estruturada. O processo de recuperar informações de um determinado domínio ou *website* por intermédio de um *web crawler* é denominada *web crawling*.

O *framework* Scrapy trata-se de um robusto conjunto de funcionalidades para *web scraping*. Desenvolvido em Python e de código fonte aberto, este *framework* pode ser utilizado para recuperar informações através do protocolo HTTP de diversas fontes de dados (KOUZIS-LOUKAS, 2016).

A utilização do Scrapy para coleta de textos no twitter é descrita por Hernandez-Suarez *et al.* (2018) como uma opção viável para contornar as limitações da API do Twitter, com resultados superiores quando comparado com outras aplicações que utilizam a API.

2.4.4.4 Framework Oráculo

O *framework* Oráculo ao unir a possibilidade de coleta por meio da API do Twitter e *crawling* em uma única ferramenta contribui de forma inovadora para esse processo. Soma-se a esse diferencial a coleta de forma distribuída, ao aplicar técnicas de sistemas distribuídos.

2.5 Sistemas Distribuídos

Segundo Coulouris *et al.* (2013), sistemas distribuídos podem ser definidos como aqueles no qual os componentes de hardware ou software, localizados em computadores autônomos interligados em rede, comunicam-se e coordenam suas ações apenas trocando mensagens entre si. Estes computadores podem estar distribuídos geograficamente distantes, bastando apenas estarem conectados em rede para executar os sistemas distribuídos. A motivação para desenvolver sistemas distribuídos consiste no desejo de compartilhar recursos, sejam esses de hardware ou software.

As seguintes características são comuns aos sistemas distribuídos (FERNANDES, 2001):

- **Compartilhamento de Recursos**: A capacidade de compartilhar um conjunto de objetos em um ambiente colaborativo. Tais objetos podem ser componentes de hardware, como discos rígidos e memória, ou de software, como arquivos, bancos de dados ou outros objetos de informação;
- **Abertura**: A capacidade do sistema ser estendido, agregando novas funcionalidades a partir de suas interfaces;
- **Escalabilidade**: A capacidade do sistema crescer, incrementando a sua capacidade de processamento, memória e armazenamento de forma a atender uma demanda, a qual seria limitada quando comparada com um sistema centralizado;
- **Concorrência**: Necessidade de sincronização decorrente da requisição acesso de diferentes computadores (nós) a um determinado recurso compartilhado;
- **Tolerância a falhas**: A capacidade do sistema continuar operando em caso de falhas de hardware ou software, mediante redundância ou recuperação;
- **Transparência**: A capacidade do Sistemas Distribuído ser entendido como um sistema único pelo usuário e não como uma coleção de componentes independentes.

Baeza-Yates e Ribeiro-Neto (2013) elencam a coleta distribuída, ou seja, a partir de pontos geograficamente distribuídos, como uma importante tendência, seja para contornar possíveis gargalos de rede ou para dar maior celeridade ao processo de coleta. No entanto, estes autores alertam para as variáveis de custo de comunicações entre os diferentes nós e de múltiplos acessos, que devem ser considerados.

Após avaliar as características dos sistemas distribuídos, decidiu-se por aplicar técnicas de sistemas distribuídos no desenvolvimento do *framework* Oráculo visando contornar

as limitações comumente encontradas nas ferramentas de coleta no Twitter e derivadas da API disponibilizada por esta rede social *online*.

2.6 Trabalhos Correlatos

Os trabalhos correlatos na área de coleta e mineração de textos no Twitter, comparados neste trabalho, estão relacionados no Quadro 2. As linhas deste Quadro correspondem aos trabalhos em ordem cronológica e as colunas os atributos nos quais esses trabalhos foram analisados.

Quadro 2: Trabalhos correlatos a coleta e mineração de textos no Twitter

Autor / Ferramenta	Coleta API	Coleta Crawler	Interface de Texto	Interface Web	Coleta Distribuída
Bruns e Liang(2012) yourTwapperKeeper	Sim	Não	Sim	Não	Não
Borra e Rieder(2014) DMI-TCAT	Sim	Não	Sim	Sim	Não
Hernandez-suarez(2018) Scrapy	Não	Sim	Sim	Não	Não
SANTOS (2019) LTWEET	Não	Sim	Não	Sim	Não
Framework Oráculo	Sim	Sim	Sim	Sim	Sim

Fonte: Próprio Autor.

3 METODOLOGIA

A metodologia de desenvolvimento desta pesquisa pode ser classificada quanto a sua natureza como aplicada, visto que busca gerar conhecimento para aplicação prática a solução de problemas específicos (GIL, 2008).

Segundo Gil (2008), a classificação quanto aos objetivos desta pesquisa é exploratória, por buscar proporcionar maior familiaridade com os fenômenos aqui investigados. Em geral, esse tipo de pesquisa envolve o levantamento bibliográfico e análise de exemplos que estimulem a compreensão. Cabe também ressaltar o objetivo descritivo desta pesquisa, por exigir do pesquisador observar e descrever os fatos e fenômenos de determinada realidade (TRIVIÑOS, 1987).

A classificação quanto aos procedimentos deste trabalho é bibliográfica, pois é realizada por meio de um levantamento bibliográfico com a finalidade de compreender, analisar e avaliar as contribuições teóricas existentes sobre técnicas de mineração de textos, redes sociais *online*, Twitter e ferramentas disponíveis para coleta nesta rede (GIL, 2008). A partir da camada do *framework* desenvolvida, foram realizados experimentos com a finalidade de verificar se as diferentes técnicas e algoritmos disponíveis nesta ferramenta se configuram de modo a influenciar nos resultados da coleta de textos, assim, este trabalho também é classificado quanto aos procedimentos como experimental (GIL, 2008).

3.1 Arquitetura da camada de coleta e mineração de textos

Para contribuir para a compreensão da metodologia desta pesquisa, foi elaborado um modelo da camada de Coleta e Mineração de Textos do *framework* Oráculo . Esse modelo foi então desenvolvido e aplicado neste trabalho. A Figura 3 representa essa modelagem.

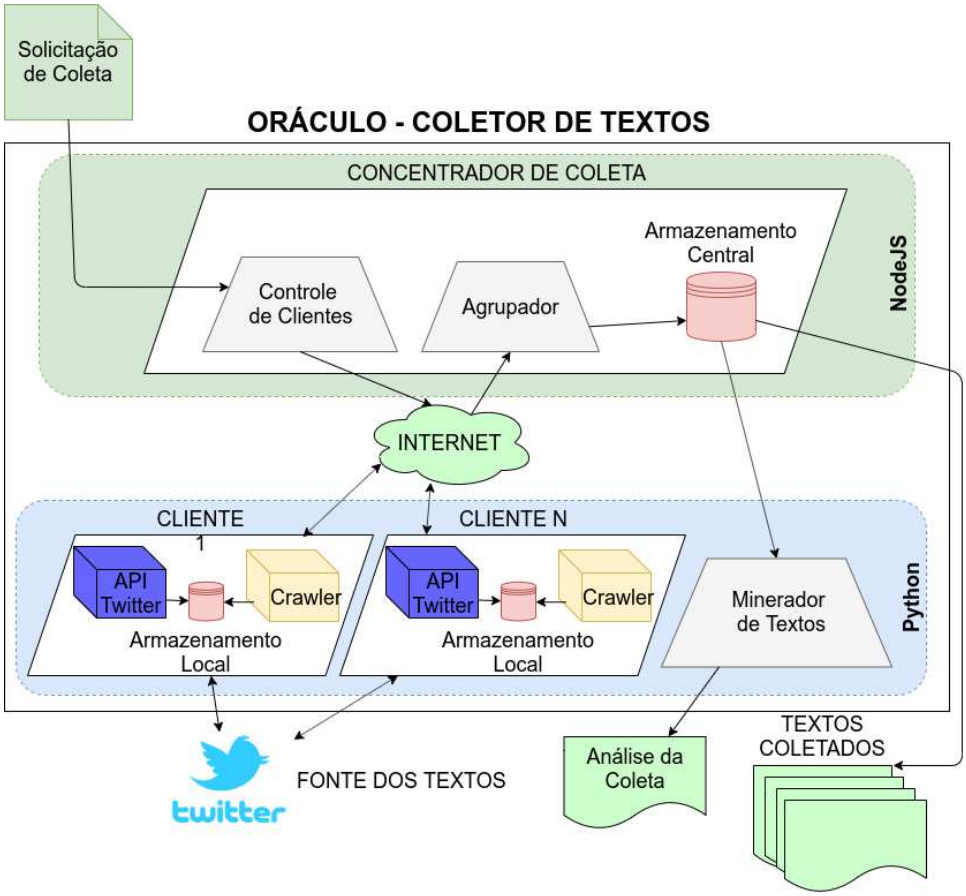
A camada de coleta e mineração de textos é composta pelo módulo concentrador e instâncias do módulo cliente:

- **Concentrador**: é responsável por receber a solicitação de coleta em sua interface *web* e realizar as tratativas correspondentes para executá-la. Este módulo então utiliza a classe controle de clientes para agendar uma requisição de coleta para as instâncias do módulo clientes configuradas.
- **Cliente**: este módulo reúne os diferentes algoritmos de coleta, chaves de acesso ao Twitter e o armazenamento local. Ao receber a solicitação de coleta este módulo a processa e retorna os dados coletados para o módulo concentrador.

A comunicação entre os módulos é realizada através do protocolo HTTP - *Hypertext Transfer Protocol*, padrão na internet para comunicação multimídia entre aplicações distribuídas e colaborativas (FIELDING *et al.*, 1999).

A Figura 4 representa o fluxo da classe clientes e seus diferentes algoritmos de coleta.

Figura 3: Modelo do funcionamento da camada de coleta e mineração de textos do *Framework* Oráculo



Fonte: Próprio autor

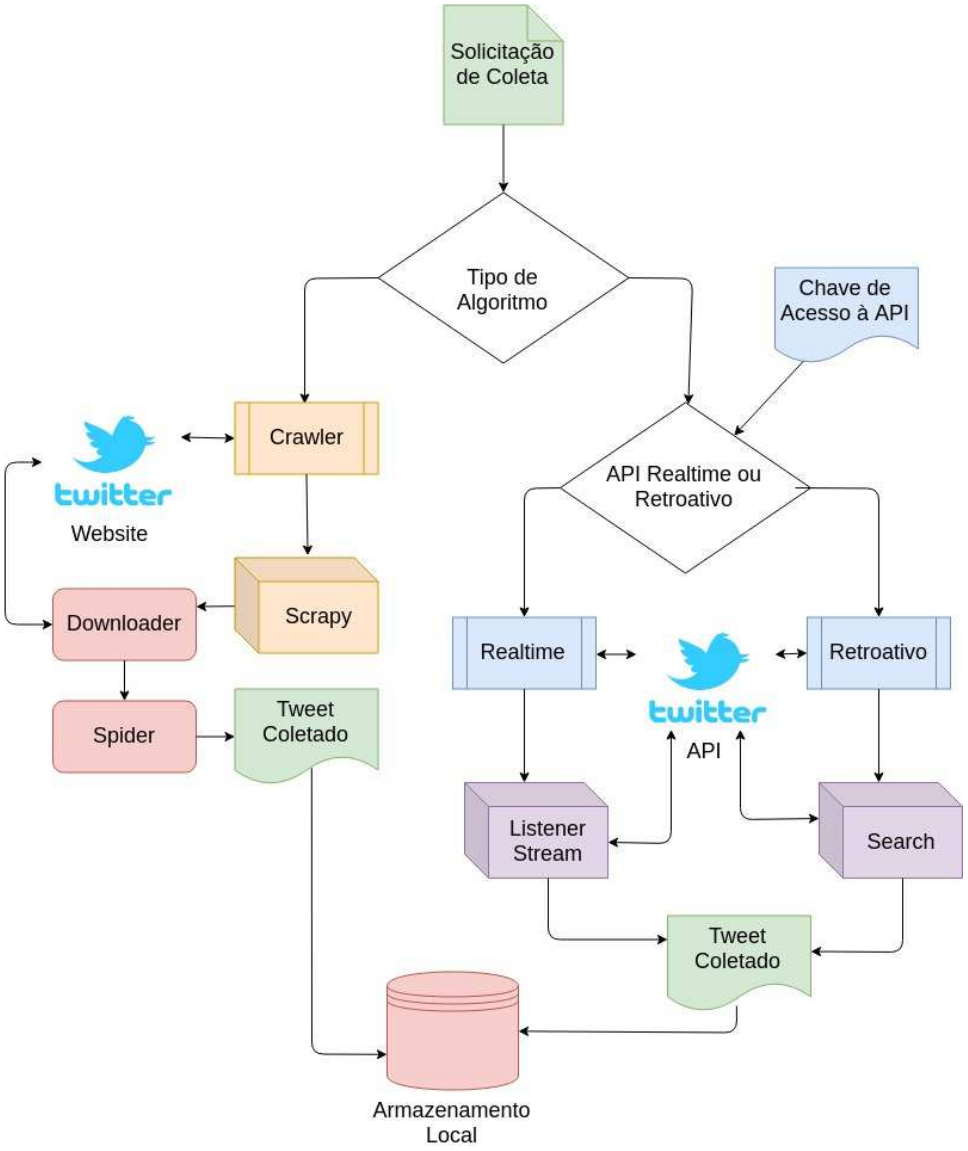
Inicialmente é enviada à classe cliente uma requisição de coleta, na qual devem estar presentes as seguintes informações: termo de consulta, algoritmo utilizado na coleta, datas de início e fim de coleta, tipo de saída de dados e número de nós utilizados na coleta.

A coleta solicitada pode se referir a textos do tipo *realtime*, para *tweets* postados posteriores ao início da coleta, ou retroativa, para *tweets* postados antes do início da coleta. Para coletas *realtime*, a única opção disponível é a coleta através da API do Twitter. Em coletas retroativas o usuário pode selecionar a coleta através de um Crawler ou da API do Twitter.

Para saída de dados, o usuário pode selecionar arquivos individuais do tipo *JavaScript Object Notation* - JSON ou em uma coleção do banco de dados MongoDB[1]. Estas opções foram escolhidas por permitir uma rápida visualização dos dados e conversão para outros formatos, além de permitir o intercâmbio das coletas entre estes dois tipos de saída. Esse intercâmbio é possível, visto que os arquivos do MongoDB são gravados como *Binary* JSON - BSON (JSON binários). Para grandes volumes de arquivos gravados, este formato demonstra um desempenho superior aos bancos de dados relacionais MySQL (MARQUES; MEDEIROS; PEREIRA, 2018).

[1] MongoDB é um sistema de banco de dados NoSQL de código livre é orientado a documentos (MONGODB, 2019)

Figura 4: Diagrama do Coletor de Textos do *framework* Oráculo



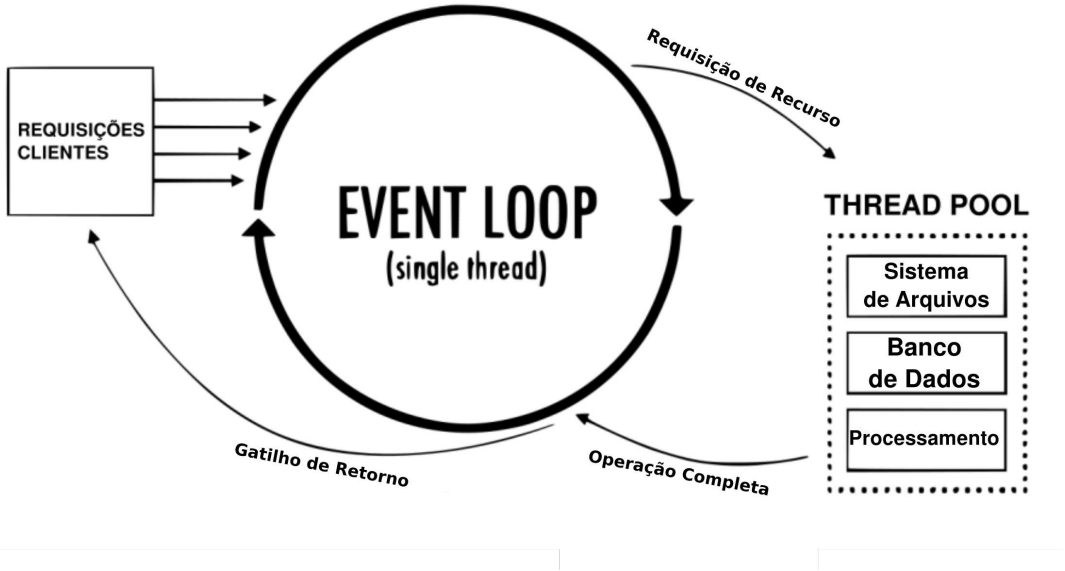
Fonte: Próprio autor

O número de nós clientes que podem ser usados na coleta varia de acordo o número de instâncias do módulo cliente configuradas. Nos testes desta pesquisa foram utilizados dois nós, um no Brasil e outro nos Estados Unidos. O usuário então pode selecionar a coleta local, utilizando apenas um nó, a coleta distribuída entre dois nós de forma idêntica ou ainda a coleta distribuída de forma complementar. Esta última opção particiona a coleta em dois períodos distintos que são coletados e depois unidos para formar uma única coleta.

O módulo concentrador de coleta foi desenvolvido com o Node JS, que se trata de um interpretador de linguagem JavaScript, que executa no servidor seu código. Esse interpretador tem por característica possuir recursos para construção de interface web, ser orientado a eventos, *single thread*, assíncrono e não bloqueante, ou seja, apesar de trabalhar apenas sobre uma *thread*, o processamento de requisições não fica bloqueado aguardando uma requisição terminar para

continuar o processamento das demais requisições do sistema (PEREIRA, 2014). A Figura 5 representa a arquitetura de funcionamento do Node JS.

Figura 5: Arquitetura de funcionamento do Node JS



Fonte: (DAYLEY, 2014) (Adaptada)

Estas características conferem escalabilidade e possibilidade real de paralelismo em aplicações desenvolvidas com o Node JS (MORAES, 2018). Assim, o Node JS foi escolhido para o desenvolvimento da classe concentrador de coleta visando permitir o gerenciamento de requisições de múltiplos clientes, acesso ao disco e ao banco de dados sem afetar desempenho do sistema.

O módulo cliente é constituído pela classe de coleta através da API do Twitter, classe de coleta através de um Crawler, utilizando o *framework* Scrapy, e pelo armazenamento local, que pode ser feito através de arquivos JSON ou banco de dados MongoDB. Ao receber uma requisição de coleta, o módulo cliente identifica os termos a serem coletados, o tipo de coleta, o período e o algoritmo a ser utilizado. A depender do algoritmo selecionado, o cliente utiliza a classe correspondente, seja a relacionada à API do Twitter ou ao Crawler, e realiza a coleta, salvando os dados correspondentes no Armazenamento Local.

Finalizada a coleta, o cliente retorna ao módulo Concentrador de Coleta a informação coletada, que, por sua vez, deve aguardar que todos os clientes terminem esse procedimento para então processar o agrupamento e retornar os dados ao usuário.

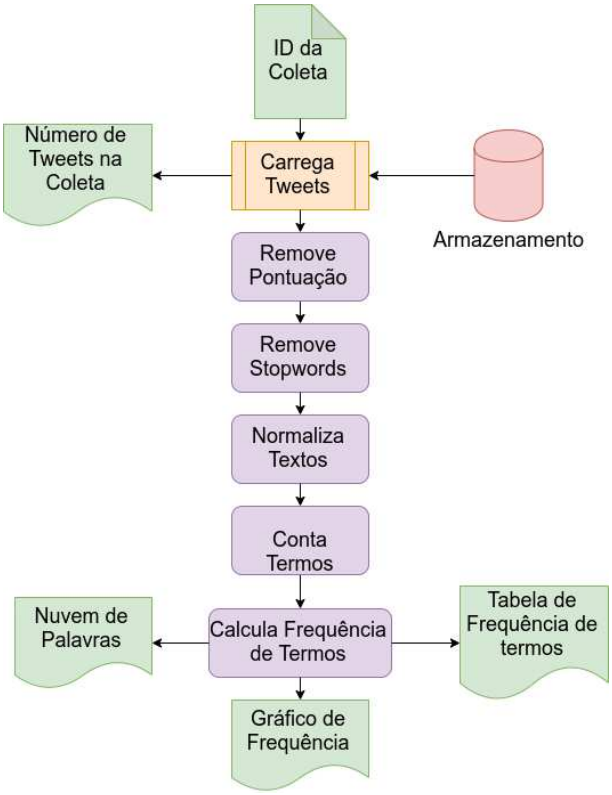
O módulo cliente pode ser instalado e configurado em múltiplos computadores, geograficamente distribuídos, criando assim várias instâncias de coleta. Essas múltiplas instâncias visam ampliar o número de *tweets* coletados, contornar as limitações de acesso através da API do Twitter, visto que serão feitos acessos distintos através de conexões diferentes à internet e chaves de acesso ao Twitter, de forma que o Twitter não correlacione esses acessos, e pode prevenir que a variação na qualidade do *link* de acesso à internet e possíveis falhas na rede interfiram na

capacidade de coleta naquele momento, visto que a redundância nos acessos torna o sistema mais tolerante a falhas.

O módulo cliente foi desenvolvido utilizando a linguagem de programação Python. A escolha dessa linguagem se deve à vasta documentação e bibliotecas para acesso ao Twitter. Outra razão é a existência de suporte a essa linguagem em diversas hospedagens distribuídas pelo mundo, fato que facilita instanciar clientes geograficamente distribuídos.

Aliado ao coletor de textos, foi desenvolvido um minerador de textos para proporcionar análises das coletas realizadas. A Figura 6 apresenta o diagrama de funcionamento deste Minerador.

Figura 6: Diagrama do Minerador de Textos do *framework* Oráculo



Fonte: Próprio autor

Esse minerador fornece informações sobre o número de *tweets* coletados, número de termos presentes na coleta, frequência dos termos na coleta, com geração de gráficos desta frequência e nuvem de palavras dos termos presentes. O minerador de textos foi desenvolvido em linguagem Python, mais um vez devido ao número de bibliotecas disponíveis nesta linguagem, e para que pudesse ser instanciado junto ao módulo cliente nos diferentes nós onde o Oráculo estiver presente.

3.2 Ambiente de testes

As coletas de textos foram realizadas em computadores diferentes com configuração de hardware semelhante: 4 (quatro) núcleos de processamento, 2 (duas) *threads* por núcleo e com no mínimo 8 (oito) Gigabytes de RAM. Desta maneira, buscou-se dar condições de execução justas a cada algoritmo. Embora, durante os testes, observou-se que os algoritmos de coleta utilizam apenas uma *thread* e consomem recursos computacionais mínimos em termos de processamento e memória.

Os nós que realizaram os testes possuem acesso à internet de pontos distintos e geograficamente distribuídos, cada um de 100 (cem) Megabits por segundo - Mbps, para que o tráfego de rede gerado por uma coleta não interferisse nas demais. Estes nós encontram-se distribuídos entre o Brasil e os Estados Unidos da América - EUA e foram identificados assim nos resultados dos testes.

Nas coletas realizadas por meio da API do Twitter foram utilizadas diferentes chaves de acesso e de identificação da aplicação, mais uma vez, para que não houvesse interferência nos resultados de diferentes coletas devido às limitações impostas pela API.

3.3 Testes da ferramenta

Para verificar o desempenho em coletas do Oráculo, foram realizados testes de desempenho para cada algoritmo nele disponível: API Retroativo, em nós distintos e com coletas complementares; Crawler, em nós distintos e com coletas complementares; e API *Realtime* em cada nó disponível no *framework*.

Estes testes do *framework* Oráculo foram acompanhados de testes do DMI-TCAT, ferramenta de coleta no twitter com interface *web*. Desta forma, objetivou-se comparar o desempenho com uma ferramenta reconhecida e utilizada por diversos pesquisadores em todo o mundo.

Os testes foram divididos em dois momentos: testes de desempenho e testes de validação. Inicialmente foram realizados testes de desempenho em diferentes intervalos de tempo, buscando analisar de que forma os algoritmos se comportam com o incremento do tempo de coleta. Nestes testes foram coletados pouco mais de 1.700.000 *tweets* em 44 coletas.

Os testes de desempenho foram realizados em coletas com tempo determinado, ocorrendo de 5 (cinco), 15 (quinze), 30 (trinta) e 60 (sessenta) minutos, de forma a verificar a evolução no número de *tweets* coletados em cada ferramenta e algoritmo com o decorrer do tempo. As coletas se iniciaram e finalizaram em todos os algoritmos e nós clientes ao mesmo momento, através da sincronização prévia dos relógios dos computadores utilizados como nós.

Das coletas realizadas, foram analisados o número de *tweets* coletados no espaço de tempo determinado, número de palavras em cada coleta e a frequência do termo de consulta dentro do conjunto da coleta. Para que fosse possível analisar o número de palavras e a frequência

do termo de consulta no conjunto de textos advindos de uma coleta, o Oráculo conta com um minerador de textos, capaz de gerar gráficos e nuvem de palavras dos textos analisados.

Devido à limitação imposta pela API do Twitter a 180 (cento e oitenta) requisições dentro de um período de 15 (quinze) minutos, esse foi o intervalo selecionado para que suas coletas fossem mineradas. Dessa forma, buscou-se equalizar a comparação entre os algoritmos disponíveis e ferramentas utilizadas.

Buscando um respaldo científico dos resultados dos testes de desempenho foram realizados os testes de validação. Esses testes consistiram em realizar uma nova série de coletas para os algoritmos do Oráculo e o DMI-TCAT, reunindo assim uma nova massa de textos, e sobre essa massa de coletas foram executados testes estatísticos.

Segundo Guelpeli (2012), testes estatísticos são utilizados em pesquisas com o objetivo comparar condições experimentais, visando auxiliar e fornecer respaldo científico às pesquisas que tenham validade e aceitabilidade no meio científico. Estes testes podem ser divididos em paramétricos e não paramétricos.

Os testes paramétricos são aqueles em que os valores da variável estudada possuem uma distribuição específica, como normal, uniforme ou exponencial. Já os não paramétricos são aqueles em que não há conhecimento da distribuição da variável na população. Em pesquisas, por não se conhecer bem a distribuição da população e seus parâmetros, a estatística não paramétrica é mais apropriada, o que reforça a necessidade destes estudos e a importância da análise destas pesquisas por meio desses testes (CALLEGARI-JACQUES, 2009).

Nesta pesquisa, a variável para análise através dos testes estatísticos é o número de *tweets* coletados por cada algoritmo para cada amostra. Para escolha dos testes estatísticos a serem utilizados foi utilizado o diagrama proposto por Callegari-Jacques (2009), disponível no ANEXO A. Foram selecionados o teste de Friedman e o coeficiente de concordância de Kendall. Segundo Guelpeli (2012), estes testes são os mais adequados para verificar se existe diferença significativa na distribuição em todas as amostras analisadas nos experimentos.

O teste de Friedman é útil para testar a hipótese de que existe diferença entre mais de dois tratamentos, com base em amostra de grupos dependentes. É, portanto, uma alternativa não paramétrica para a análise de variância - ANOVA com dois critérios de classificação. Por essa razão é referido também como análise de variância de Friedman, ANOVA de Friedman e ANOVA não paramétrica com dois critérios (VIEIRA, 2018).

O Coeficiente de Concordância de Kendall - Kendall W é teste não paramétrico, utilizado para normalizar o teste de Friedman. Este teste mede a concordância entre diferentes avaliadores de uma amostra, sendo 0 (zero) nenhuma concordância e 1 (um) a completa concordância (DELGADO; VIANNA; GUELPELI, 2010) (GUELPELI, 2012).

Para a realização dos testes de validação foram realizadas coletas com intervalo de tempo fixo de 15 minutos, mais uma vez devido este ser período que a API do Twitter disponibiliza para requisições. Desta forma, buscou-se equalizar a comparação entre os métodos que utilizam a API do Twitter dos que não a utilizam. As coletas realizadas neste momento

foram analisadas quanto ao número de *tweets* coletados. Foram analisadas 110 amostras para os 5 algoritmos, totalizando 550 coletas e pouco mais de 13 milhões de tweets. Estes dados foram analisados com os testes estatísticos de Friedman e Coeficiente de Concordância de Kendall.

O termo de consulta utilizado em todos os testes desta pesquisa foi “bolsonaro”, sobrenome do atual presidente da república brasileira. Este termo foi escolhido para garantir que os *tweets* estivessem disponíveis durante todo o período de coleta, visto que há engajamento considerável à figura do Presidente nesta plataforma, desde a eleição presidencial de 2018 (GABARDO *et al.*, 2019).

4 RESULTADOS E DISCUSSÕES

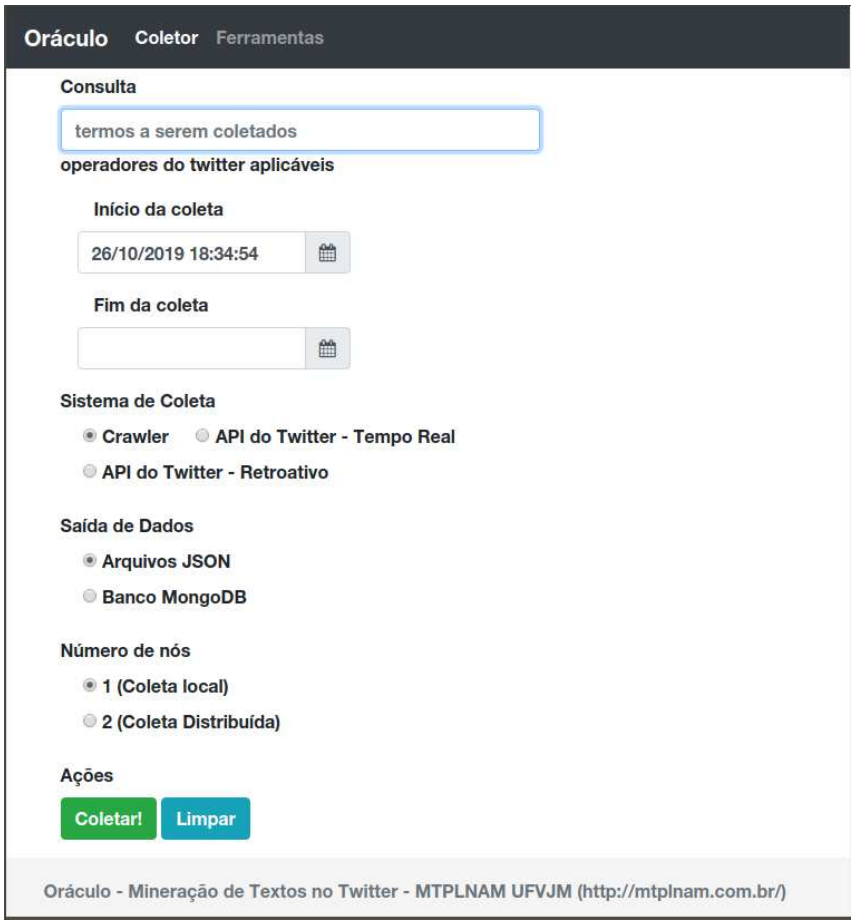
Neste capítulo, são apresentados os resultados do desenvolvimento da camada de coleta e mineração de textos do *framework* Oráculo, segundo a metodologia desta pesquisa, e os resultados provenientes dos testes realizados com a ferramenta.

4.1 Interface da ferramenta

Ao analisar outras ferramentas de coleta de textos no Twitter, observou-se que muitas encontram-se na forma de *scripts*, e que requerem conhecimento avançado na área de Ciências da Computação para a sua correta configuração e execução. Embora os nós da camada de coleta e mineração de textos do Oráculo permitam a execução por meio de *scripts* e comandos, houve a preocupação de disponibilizar também uma interface *web* para essa ferramenta.

Assim, por meio de uma interface web, simples e funcional, usuários de diferentes áreas do conhecimento podem utilizar de maneira satisfatória a camada de coleta e mineração de textos do *framework* Oráculo, um dos objetivos específicos desta pesquisa. A Figura 7 apresenta interface desenvolvida.

Figura 7: Interface *web* da camada de Coleta e Mineração de Textos



Fonte: Próprio autor

Nessa interface o usuário do Oráculo deve informar o termo de consulta para que seja realizada a sua coleta no Twitter. Aliados ao termo de consulta, o usuário pode utilizar operadores para ampliar ou reduzir o seu espaço de coleta no twitter, como especificar um determinado usuário ou uma determinada data. Os operadores de consulta disponíveis no Oráculo são compatíveis com os disponibilizados na página de consulta do twitter e estão listados no Quadro 3:

Quadro 3: Operadores disponíveis para consulta ao Twitter através do Oráculo

Operador	Descrição	Exemplo
(espaço)	Retorna tweets contendo as duas ou mais palavras separadas por espaços (operador padrão)	enem mec
“”	Retorna tweets contendo exatamente a expressão entre “” (aspas)	“educação infantil”
OR	Retorna tweets contendo uma ou ambas as palavras	sisu OR prouni
-	Retorna tweets que não contém a palavra precedida de -	sisu -prouni
#	Retorna tweets com a hashtag	#enem
from:	Retorna tweets enviados do usuário indicado	from:MEC_Comunicacao
to:	Retorna tweets enviados para o usuário indicado	to:MEC_Comunicacao
@	Retorna tweets que tenham o usuário marcado	@MEC_Comunicacao
near:	Retorna tweets enviados da proximidade da cidade indicada	near:belo horizonte
within:	Utilizado junto com o operador near, define a distância da proximidade da cidade indicada	near:belo horizonte within:10km
since:	Retorna tweets enviados desde a data definida	since:2019-06-01
until:	Retorna tweets enviados até a data definida	until:2019-06-01
?	Retorna tweets que tenham perguntas sobre o termo	enem ?
filter:links	Retorna tweets que tenham links aliados ao termo	prouni filter:links

Fonte: Próprio Autor.

A camada do *framework* Oráculo realiza as coletas de *tweets* mediante agendamento, com data e hora para início e fim determinados (estas opções estão disponíveis na interface *web* logo abaixo do termo de consulta).

A próxima opção, da interface da ferramenta, está relacionada ao sistema de coleta que será utilizado. Estão disponíveis as funcionalidades relacionadas a API do twitter: retroativo, para coletar *tweets* que tenham sido postados antes do início da coleta; e *realtime*, para *tweets* que sejam postados depois do início da coleta. A outra opção relacionada ao sistema de coleta é o Crawler, que utiliza o retorno textual da interface *web* do twitter para coletar os *tweets*.

As opções oferecidas pelo Oráculo para saída dos dados são: JSON, na qual os *tweets* são salvos em arquivos individuais de texto neste formato; e MongoDB, na qual os *tweets* são organizados numa coleção neste banco de dados orientado a documentos, semelhante ao JSON.

A última opção que o usuário da camada do Oráculo pode selecionar antes de iniciar a sua coleta é relacionada ao número de nós que serão utilizados. A coleta pode ser realizada em apenas 1 (um) nó - coleta local, 2 (dois) nós de forma idêntica, ou 2 (dois) nós de forma complementar. Nesta última, a ferramenta divide a coleta em dois períodos distintos e os nós coletam *tweets* únicos que depois são agregados para formar uma única coleta.

4.2 Análise Quantitativa

As coletas para análise do desempenho da camada de coleta e mineração de textos do *framework* foram realizadas entre 20 e 30 de junho de 2019, sincronizadas conforme a

metodologia desta pesquisa. A Tabela 2 apresenta os resultados para a coleta retroativa, realizada com cada um dos algoritmos disponíveis na camada do *framework* Oráculo, e com a ferramenta DMI-TCAT:

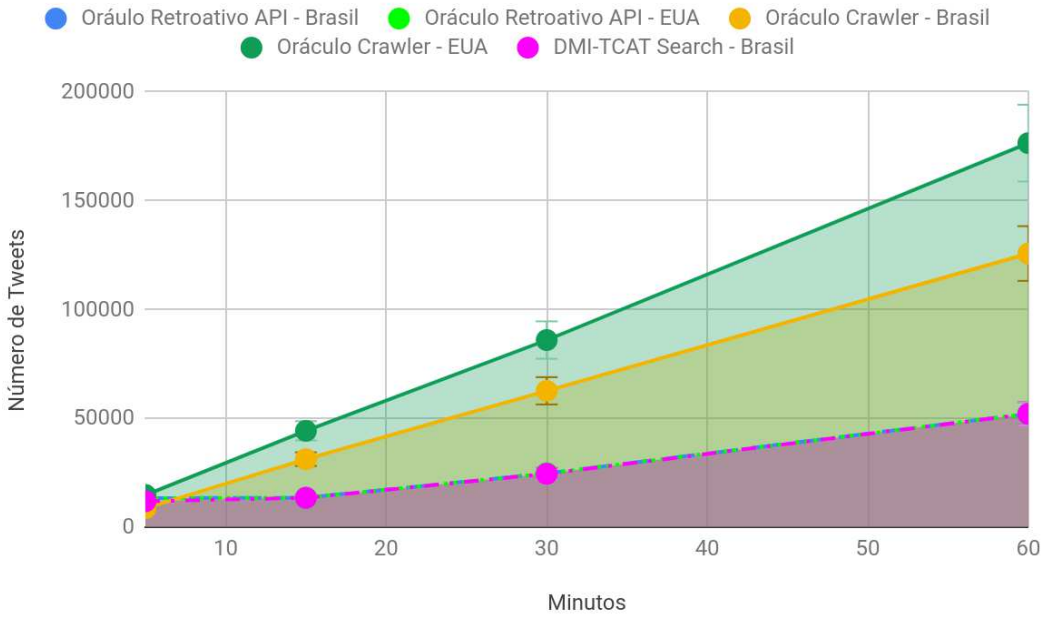
Tabela 2: Quantidade tweets coletados nos testes para algoritmos retroativos

Algoritmo	5 min	15 min	30 min	60 min	Média Tweets/Minuto
Oráculo API Retroativo - Brasil	13.304	13.494	24.685	52.091	941,58
Oráculo API Retroativo - EUA	13.304	13.493	24.684	52.092	941,57
Oráculo Crawler - Brasil	8.703	31.238	62.634	125.734	2.075,54
Oráculo Crawler - EUA	14.767	44.234	85.956	176.474	2.922,1
DMI-TCAT Search - Brasil	11.861	13.457	24.494	52.012	925,67

Fonte: Próprio autor.

As coletas foram feitas para cada intervalo, e não possuem relação com o intervalo anterior. Dessa forma, foi possível verificar uma tendência de evolução estável do número de tweets coletados de acordo com o incremento do intervalo de tempo. A Figura 8 apresenta essa tendência na forma de um gráfico.

Figura 8: Gráfico com a quantidade de *tweets* coletados nos testes para algoritmos retroativos



Fonte: Próprio autor

Nota-se que os nós que utilizam o algoritmo baseado na API do Twitter obtiveram desempenho semelhante, tanto no Oráculo quanto no DMI-TCAT. Devido à limitação do número de requisições do twitter a cada 15 minutos, os resultados para 5 e 15 minutos são próximos,

visto que os algoritmos consomem essas requisições rapidamente e ficam aguardando a próxima janela de 15 minutos para fazer novas requisições.

Embora utilizem o mesmo algoritmo, os nós no Brasil e nos Estados Unidos executando na opção Crawler obtiveram resultados consideravelmente diferentes, sendo o nó nos Estados Unidos 40% mais eficiente em média que o do Brasil. Como esse algoritmo realiza acesso através do protocolo HTTP à página de pesquisa do Twitter, a limitação para retorno dos dados encontra-se apenas na comunicação de rede.

Como ambos os *links* de acesso a internet dos nós eram idênticos, restou analisar a latência média de resposta do servidor do Twitter, momento em que observou-se que a latência média do nó brasileiro era de 122 (cento e vinte e dois) milissegundos e a latência média do nó localizado nos Estados Unidos era de 5 (cinco) milissegundos. Essa diferença no tempo de resposta explica a diferença de desempenho neste tipo de coleta.

As coletas retroativas, realizadas no intervalo de 15 minutos, foram analisadas com o minerador de textos do Oráculo e do DMI-TCAT, quanto ao número de termos presentes, frequência do termo de consulta utilizado, e a porcentagem de incidência que essa frequência representa no número total de termos. A Tabela 3 apresenta estes resultados:

Tabela 3: Análise quantitativa dos *Tweets* coletados de forma Retroativa em 15 minutos

Algoritmo	Número de Termos	Frequência Termo Consulta	% Frequência
Oráculo API Retroativo - Brasil	249.438	14.169	5,68
Oráculo API Retroativo - EUA	249.427	14.168	5,68
Oráculo Crawler - Brasil	561.501	28.322	5,04
Oráculo Crawler - EUA	808.468	40.341	4,99
DMI-TCAT Search - Brasil	326.392	13.945	4,27

Fonte: Próprio autor.

Analisando os dados da Tabela 3, nota-se que, embora o quantitativo de *tweets* retornados entre os algoritmos seja diferente, a incidência do termo de consulta dentro da coleta realizada se mantém próxima. A diferença de método, coleção de *stopwords* e algoritmo de mineração de textos do DMI-TCAT pode explicar a vantagem do Oráculo.

Para coletas realizadas com o algoritmo *realtime*, foram testados novamente nós com o Oráculo e com o DMI-TCAT. A Tabela 4 apresenta os resultados desta coleta.

Verificou-se nessa situação um desempenho muito diferente entre os nós do Oráculo para o DMI-TCAT: como em todos estes nós a API do Twitter é a base dessas coletas, restam apenas as bibliotecas e métodos utilizados no desenvolvimento desta ferramenta para explicar essa diferença. Salienta-se que a linguagem de programação adotada pelo DMI-TCAT difere do *framework* Oráculo, possuindo assim diferentes bibliotecas e recursos.

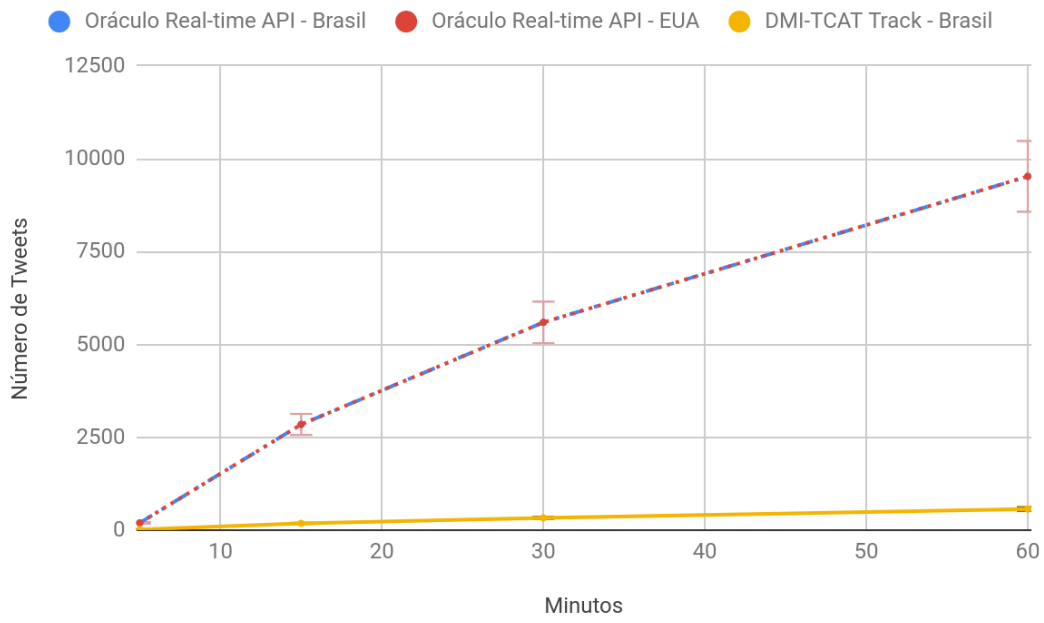
Tabela 4: Quantidade tweets coletados nos testes para algoritmos *realtime*

Algoritmo	5 min	15 min	30 min	60 min	Média Tweets/Minuto
Oráculo API - Realtime - Brasil	195	2.846	5.594	9.524	165,0818182
Oráculo API - Realtime - EUA	195	2.845	5.593	9.525	165,0727273
DMI-TCAT - Track - Brasil	17	181	329	567	9,945454545

Fonte: Próprio autor.

A Figura 9 apresenta o gráfico com os resultados para *tweets* coletados em *realtime* e a tendência de evolução com incremento de minutos no tempo de coleta, em que o nó coletor do Oráculo no Brasil e Estados Unidos obtiveram o mesmo resultado.

Figura 9: Gráfico com a quantidade de *tweets* coletados nos testes para algoritmos *realtime*



Fonte: Próprio autor

As coletas *realtime*, realizadas no intervalo de 15 minutos, foram também analisadas utilizando o minerador de dados do Oráculo e DMI-TCAT. A Tabela 5 apresenta os resultados dessa análise.

Ao analisar os resultados apresentados na Tabela 5, verificou-se que o desempenho superior do Oráculo no número de *tweets* coletados implicou também em uma maior incidência do termo de consulta no conjunto analisado, embora essa diferença possa ter relação com a diferença do algoritmo de mineração de textos. Devido a esses algoritmos serem próprios de cada ferramenta e possuírem uma diferença de formato de dados utilizados, não foi possível comparar as coletas com um único algoritmo.

Tabela 5: Análise quantitativa dos *Tweets* coletados de forma *Realtime* em 15 minutos

Algoritmo	Número de Termos	Frequência Termo Consulta	% Frequência
Oráculo API - Realtime - Brasil	53.615	3.100	5,78
Oráculo API - Realtime - USA	53.584	3.098	5,78
DMI-TCAT - Track	4.213	195	4,63

Fonte: Próprio autor.

O Oráculo possibilita coletas distribuídas em diferentes nós geograficamente dispersos. Ao utilizar esse recurso, o Oráculo divide a coleta em diferentes partições de acordo com a data de coleta, e cada um dos nós assume uma partição para coletar. Finda a fase de coleta, o sistema reúne todos os *tweets* coletados formando uma coleta maior e assim mais abrangente. Apesar desse recurso não estar disponível no DMI-TCAT, ele foi comparado para servir de referência ao desempenho do Oráculo. A Tabela 6 apresenta os resultados desse teste para 2 (dois) nós.

Tabela 6: Quantidade tweets coletados nos testes para algoritmos retroativos distribuídos

Algoritmo	5 min	15 min	30 min	60 min	Média Tweets/Minuto
Oráculo - API Retroativo Nós Complementares	28.281	27.805	54.361	108.506	1.990,481818
Oráculo - Crawler Nós Complementares	18.870	63.885	134.005	265.085	4.380,409091
DMI-TCAT Search	13.881	14.568	28.935	46.224	941,8909091

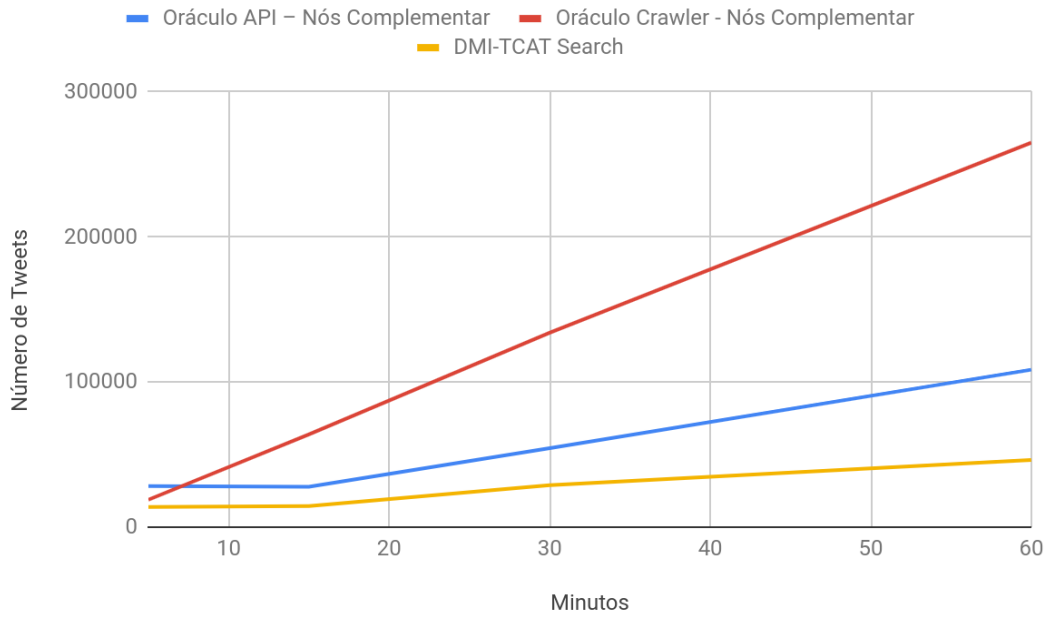
Fonte: Próprio autor.

A Figura 10 apresenta o gráfico com os resultados para coletas retroativas realizadas de forma distribuída entre dois nós. É possível observar a tendência de evolução do Oráculo utilizando o algoritmo Crawler inicia-se com um desempenho inferior ao algoritmo executando a API mas com incremento de minutos no tempo de coleta o seu desempenho supera o API por não ter limitação de janelas de coleta.

Observa-se que, apesar de utilizar o dobro do número nós, o resultado da coleta distribuída não é necessariamente o dobro da coleta feita por apenas um nó. Como a coleta distribuída é realizada utilizando períodos diferentes de *tweets* postados, o número de *tweets* disponíveis para coleta varia.

Cabe destacar o desempenho superior do algoritmo utilizando o Crawler em nós distribuídos em relação ao algoritmo do Oráculo utilizando da API também de forma distribuída, cenário no qual o Crawler obteve um desempenho 120% superior.

Figura 10: Gráfico com a quantidade de *tweets* coletados nos testes para algoritmos retroativos distribuídos



Fonte: Próprio autor

As coletas realizadas de forma distribuída em 15 minutos com o Oráculo e com o DMI-TCAT como referência foram também analisadas com seus respectivos mineradores de texto. A Tabela 7 apresenta os resultados dessa análise.

Tabela 7: Análise quantitativa dos *Tweets* coletados de forma retroativa distribuída em 15 minutos

Algoritmo	Número de Termos	Frequência Termo Consulta	% Frequência
Oráculo - API Retroativo Nós Complementares	505.337	29.374	5,81
Oráculo - Crawler Nós Complementares	1.200.650	56.780	4,73
DMI-TCAT Search - Brasil	395.595	15.268	3,86

Fonte: Próprio autor.

Analisado os resultados da Tabela 7, identifica-se um desempenho superior ao se utilizar o Oráculo. Como a coleta complementar divide a coleta em partições de tempo distintas, coletando assim *tweets* diferentes entre os algoritmos do Oráculo, pode-se explicar, assim, a diferença de incidência. Como o DMI-TCAT ficou limitado a um único nó, assim não particionou sua coleta entre diferentes datas, e seu resultado refletiu essa limitação.

4.3 Pontos Positivos e Negativos

Embora possa aparentar que o Crawler é sempre a melhor opção de coleta e que os algoritmos baseados na API não são interessantes, é necessário analisar o retorno obtido com ambos os algoritmos. Apesar de ambos retornarem os mesmos textos dos *tweets*, e a saída do Oráculo padroniza a sua saída em JSON ou no MongoDB, a principal diferença entre esses está nos metadados aliados aos *tweets*.

Nos algoritmos que se utilizam da API do Twitter, o retorno de metadados aliados aos textos é composto de 34 dados, alguns deles compostos. São dados como a localização geográfica do usuário que postou, língua e informações sobre o usuário que realizou a postagem. Embora muitas dessas informações não estejam sempre presentes, uma vez que são opcionais, podem ser relevantes em pesquisas diversas.

O conjunto de dados retornados pelo algoritmo baseado em Crawler é menor, uma vez que estes resultados estão limitados às informações visuais disponíveis na página de pesquisa do Twitter. São coletadas pelo Crawler referente aos *tweets*: número de *retweets*, id do usuário, url, texto, nome do usuário, data e hora de postagem, mídias, se é uma resposta, se é um *retweet*, se possui mídias, id do *tweet*, número de respostas e número de vezes que este foi favoritado. Quanto aos usuários, o Crawler coleta: nome, usuário, avatar e id. Apesar de relevantes e sempre presentes, estes dados podem ser insuficientes para a realização de determinadas pesquisas.

Quanto à possibilidade de coletar *tweets* de forma retroativa, o algoritmo de coleta baseado no Crawler consegue retornar *tweets* de toda a base histórica do Twitter, desde a sua fundação em 21 de março de 2006. Assim, por não estar limitado aos últimos sete dias como os algoritmos baseados na API do Twitter, oferece aos pesquisadores o acesso uma base de pesquisa muito maior.

Desta forma, não há como uma única técnica atender às necessidades de todos os pesquisadores que utilizam o Twitter como fonte de dados para suas pesquisas. É necessário que o pesquisador, ao utilizar o Oráculo, conheça as limitações e os benefícios de cada algoritmo e selecione o que melhor atende às suas necessidades.

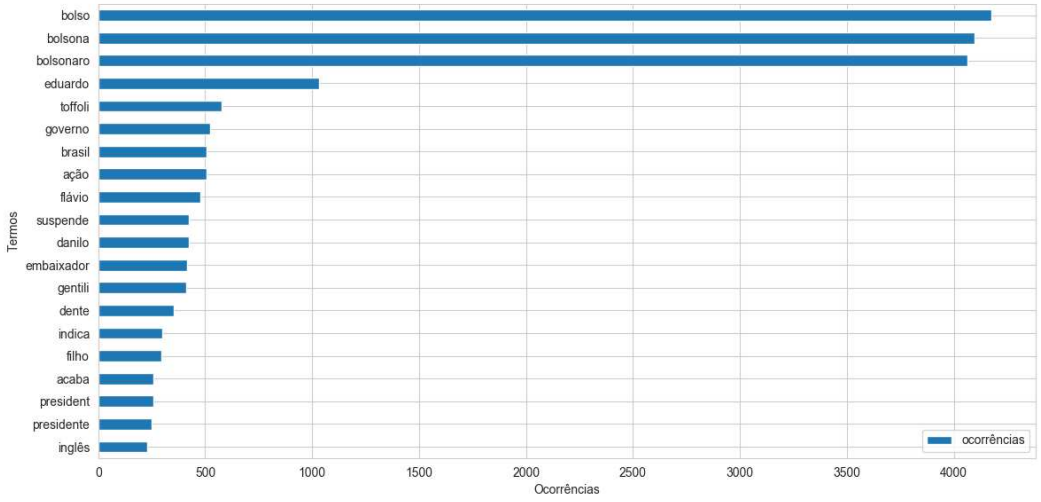
4.4 Minerador de Textos

Ao lidar com dados textuais, é preciso manter em mente que, para que se possa compreender de uma forma rápida e simplificada o conhecimento implícito, é necessário encontrar formas que possam representá-lo para transmitir algum conhecimento (SARGIANI *et al.*, 2018). Histogramas e nuvens de palavras podem ser utilizados no processo de avaliação de documentos contendo textos não estruturados, para obtenção de conhecimento oculto nos textos (BRUNO, 2016).

O *framework* Oráculo conta com ferramentas de mineração de textos para análise dos textos coletados. Essas ferramentas compreendem a remoção de *stopwords*, contagem de termos na coleta, levantamento de frequência de termos, gráficos de frequência e nuvem de

palavras acerca da coleta. Essas funções de mineração de texto podem ser configuradas pelo usuário, de forma a personalizar estes produtos, ajustando-os a demanda de cada domínio de pesquisa. A Figura 11 apresenta um exemplo de gráfico gerado pelo Oráculo com uma coleta de textos realizada para o desenvolvimento desta pesquisa.

Figura 11: Gráfico de frequência dos termos



Fonte: Próprio autor

As nuvens de palavras são comumente utilizadas para análise qualitativa de dados. A construção destas nuvens consiste em usar tamanhos e fontes de letras diferentes de acordo com a frequência das ocorrências das palavras no texto analisado (VILELA; RIBEIRO; BATISTA, 2018). A Figura 12 apresenta o exemplo de uma nuvem de palavras gerada pelo Oráculo da mesma coleta de textos.

4.5 Testes de Validação

Foram realizadas 139 (cento e trinta e nove) séries de coletas, entre os dias 17 e 23 de julho de 2019, com cinco nós coletores, sendo que um utilizou a ferramenta DMI-TCAT e os outros quatro do *framework* Oráculo. No entanto, em cerca 20% (vinte por cento) dessas coletas, o DMI-TCAT apresentou algum erro e não conseguiu concluir o processo, realizando assim um número inferior de requisições à API do Twitter. Assim, para que fosse realizada uma análise justa entre as ferramentas de coleta, as séries em que o DMI-TCAT apresentou algum erro não foram analisadas.

Dessa forma, foram analisadas 110 (cento e dez) séries de coletas para os cinco métodos, totalizando 550 (quinhentas e cinquenta) coletas. Os dados completos com os resultados destas coletas estão disponíveis no Apêndice A.

Figura 12: Nuvem de Palavras gerada pelo *framework* Oráculo

Fonte: Próprio autor

Essa massa de coletas foi submetida ao Teste de Friedman e Coeficiente de Concordância de Kendall (Kendall W). A Tabela 8 apresenta os resultados desses testes.

Tabela 8: Resultados dos Testes Estatísticos

			DMI-TCAT		Oráculo API BR		Oráculo API EUA		Oráculo Crawler BR		Oráculo Crawler EUA	
Coletor	Média dos Ranks	Kendall W	FRIEDMAN									
DMI-TCAT	1,0818	0,9268	-	-	156,50	<0,05	146,50	<0,05	321,00	<0,05	431,00	<0,05
Oráculo API – BR	2,5045		156,50	<0,05	-	-	10,00	ns	164,50	<0,05	274,50	<0,05
Oráculo API EUA	2,4136		146,50	<0,05	10,00	ns	-	-	174,50	<0,05	284,50	<0,05
Oráculo Crawler BR	4		321,00	<0,05	164,50	<0,05	174,50	<0,05	-	-	110,00	<0,05
Oráculo Crawler EUA	5		431,00	<0,05	274,50	<0,05	284,50	<0,05	110,00	<0,05	-	-

Fonte: Próprio autor.

Ao analisar os resultados dos testes estatísticos apresentados na Tabela 8 percebe-se que:

- O teste de Friedman apontou que há diferença significativa no desempenho de coleta entre os algoritmos do Oráculo e o DMI-TCAT. Ao comparar os algoritmos do Oráculo apenas entre os nós do Oráculo que coletaram por meio da API no Brasil e Estados Unidos não há uma diferença significativa.
- O teste de Friedman produziu também o ranking de desempenho de cada métodos e nó, tendo o Crawler localizados nos Estados Unidos o melhor desempenho, seguido do Crawler localizado no Brasil, em seguida o nó executando a API no Brasil, depois o nó executando a API nos Estados Unidos e por último o DMI-TCAT. Os dados completos do rank de cada coleta estão disponíveis no Apêndice B.
- O Coeficiente de Concordância de Kendall (Kendall W) apontou que as 110 amostras analisadas têm um grau de concordância de 92%, ou seja, uma concordância alta entre as coletas realizadas e por consequência no ranking produzido entre os diferentes algoritmos.

Em todas as amostras dos testes de validação, as coletas utilizando o método Crawler do *framework* Oráculo se mostraram mais eficientes, seja no nó localizado no Brasil ou nos Estados Unidos, em comparação com as coletas utilizando o método API. Esse resultado demonstra uma superioridade do método Crawler, embora seja necessário ressaltar mais uma vez que os metadados aliados a esse método são mais restritos quando comparados com os metadados coletados com o método API.

O nó coletor utilizando o Crawler localizado nos Estados Unidos obteve, assim como nos primeiros testes de desempenho, um resultado superior ao nó utilizando esse método no Brasil, devido à uma menor latência para a rede do Twitter do nó na América do norte.

Dentre os nós do *framework* Oráculo que utilizando a API, o desempenho obtido em coletas foi semelhante, estivessem eles no Brasil ou Estados Unidos. Entre esses nós, demonstrou-se não haver diferença significativa nos resultados, com os *rankings* de desempenho praticamente empatados, com um diferença média de 0,0909 pontos. Para o DMI-TCAT, que também utiliza a API do Twitter, o desempenho foi significativamente inferior, obtendo um *ranking* distante dos resultados do Oráculo, um diferença média de 1,4227 pontos.

Esses resultados validam os resultados anteriormente obtidos na fase de testes de desempenho da ferramenta, e permitem afirmar que o *framework* Oráculo tem um desempenho superior ao DMI-TCAT para os cenários testados.

5 CONCLUSÃO

O grande volume de usuários, informações trafegadas e aplicações na vida cotidiana fazem das redes sociais *online* importante fonte para pesquisadores de diversas áreas do conhecimento. Para que se possa extrair conhecimento útil destas redes é necessário que utilizem de técnicas de mineração de textos. Devido a falta de acesso direto aos dados dessas redes, a fase de coleta de textos para mineração de textos demanda que sejam utilizadas ferramentas especializadas na coleta de textos.

Esta pesquisa dedicou-se a desenvolver a Camada de Coleta e Mineração Textos no Twitter do *framework* Oráculo, com o objetivo de que essa ferramenta apoia-se outros pesquisadores em seus trabalhos em diferentes domínios de pesquisa. Investigou-se então, de que forma o desenvolvimento desta ferramenta, que disponibilizou diferentes algoritmos e técnicas de coleta, impactou no processo de coleta e análise dos textos.

O *framework* Oráculo realizou o processo de coleta e mineração de textos do Twitter coletando cerca de 15 milhões de *tweets* nos testes realizados, obtendo um desempenho superior a ferramenta DMI-TCAT. As técnicas e algoritmos de distribuição da coleta e de *web crawling* possibilitaram contornar as limitações de número de requisições imposta pela API do Twitter. Dessa forma, demonstrou-se que este *framework* é uma ferramenta útil para que pesquisadores possam ampliar, de forma gratuita, a amostra de suas pesquisas ao coletar mais *tweets*.

Os testes de validação por meio de testes estatísticos corroboram com os resultados dos testes de desempenho e demonstraram que as diferenças de desempenho dos algoritmos do Oráculo são significantes quando comparadas ao DMI-TCAT. Estes mesmos testes apontaram que o desempenho dos algoritmos do Oráculo para coleta utilizando o método Crawler são significativamente superiores aos algoritmos do Oráculo utilizando a API do Twitter.

De posse dos resultados comparativos do desempenho entre a coleta realizada em apenas um nó (não distribuída) e a realizada em múltiplos nós (distribuída), juntamente com as informações de pontos positivos e negativos de cada um dos algoritmos disponíveis, futuros usuários do *framework* Oráculo podem planejar de forma mais eficiente suas coletas.

Destacam-se como informações importantes para planejamento de futuras coletas as relacionadas a diferença de *metadados* retornados nas coletas realizadas com a API e o Crawler. A depender do objetivo e dos dados necessários ao pesquisador usuário do Oráculo, o conjunto reduzido de metadados retornado pelo Crawler pode inviabilizar o seu uso. Restará ao pesquisador utilizar a coleta com a API, podendo ainda assim, contornar as limitações desta API através da distribuição da coleta em distintos nós.

Com os resultados desta pesquisa, evidenciou-se que o Twitter não faz diferenciação da localização geográfica do cliente para coleta dos *tweets*, visto que as coletas realizadas por meio da API do Twitter nos Estados Unidos obtiveram resultados idênticos das coletas realizada no Brasil.

No entanto, percebeu-se que o *link* de acesso a internet utilizado pode produzir impacto no desempenho de coleta quando da utilização do Crawler. Esse impacto está relacionado à latência do nó coletor para o servidor do Twitter. Quanto menor a latência mais rápida é a resposta do servidor, o que se traduz em um número maior de *tweets* coletados em um determinado espaço de tempo.

Assim, retomando a questão central desta pesquisa, é possível concluir que o *framework* Oráculo teve impacto positivo no processo de coleta e mineração de textos no Twitter. Essa ferramenta, ao reunir diferentes algoritmos e técnicas, com o objetivo de contornar as limitações da API do Twitter, alcançou um desempenho superior em comparação a outra ferramenta. Esse desempenho se traduz em coletas mais abrangentes, realizadas em um sistema distribuído e com interface *web* integrada ao minerador de textos.

Destaca-se como contribuição desta pesquisa o fato da camada de coleta e mineração de textos do *framework* Oráculo estar disponível, na forma de *software* livre, para que pesquisadores de diferentes áreas do conhecimento possam utilizá-lo em suas pesquisas. Com interface *web* mínima, porém funcional, pesquisadores que não estejam familiarizados com a área de ciência da computação possam realizar coletas e mineração de textos no Twitter de forma mais simples e eficiente.

Esta pesquisa, limitou-se à análise quantitativa dos dados, ao comparar o desempenho de número de *tweets* coletados dentro de determinado espaço de tempo entre diferentes algoritmos disponíveis no Oráculo e outra ferramenta. Elenca-se para trabalhos futuros:

- Dar continuidade ao desenvolvimento do *framework* Oráculo, melhorando a camada de coleta e mineração de textos e acrescentando outras camadas
- Realizar a análise qualitativa dos dados coletados, por meio de métricas computacionais e avaliadores humanos;
- Agregar novas ferramentas de mineração de textos, uma vez disponibilizadas de forma integrada, pesquisadores não precisarão lançar mão de outras ferramentas e executar conversões de saída de dados para realizar suas análises sobre os textos coletados;
- Adaptar o uso deste *framework* para outras redes sociais *online*, atendendo assim a necessidade de uma outra gama de pesquisadores que realizam suas pesquisas nestas redes;
- Documentar o processo de engenharia e desenvolvimento do *software* de todo o *framework* Oráculo, em conjunto com os demais pesquisadores do MTPLNAM que trabalham as outras camadas deste *framework*;
- Comparar os resultados de coleta no Twitter do *framework* Oráculo utilizando a API do Twitter em sua versão gratuita (*standard*) com as versões pagas (*premium* e *enterprise*);
- Comparar o desempenho da ferramenta produzida nesta pesquisa com outras ferramentas de coleta, sejam essas gratuitas ou pagas, como o LTWEET.

REFERÊNCIAS

ARANHA, C. N. **Processamento Automático para Mineração de Textos em Português: Sob o Enfoque da Inteligência Computacional**. Tese (Doutorado) — PUC-Rio, 2007.

AUSSERHOFER, J.; MAIREDER, A. National politics on twitter: Structures and topics of a networked public sphere. **Information, Communication & Society**, Taylor & Francis, v. 16, n. 3, p. 291–314, 2013.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Recuperação de Informação-: Conceitos e Tecnologia das Máquinas de Busca**. [S.l.]: Bookman Editora, 2013.

BENEVENUTO, F.; ALMEIDA, J. M.; SILVA, A. S. Explorando redes sociais online: Da coleta e análise de grandes bases de dados às aplicações. **Porto Alegre: Sociedade Brasileira de Computação**, 2011.

BITTENCOURT, J. R.; OSÓRIO, F. Annef–artificial neural networks framework: Uma solução software livre para o desenvolvimento, ensino e pesquisa de aplicações de inteligência artificial multiplataforma. In: **Anais do II Workshop sobre Software Livre. Porto Alegre: SBC**. [S.l.: s.n.], 2001. p. 13–16.

BOOTH, R. G. Happiness, stress, a bit of vulgarity, and lots of discursive conversation: A pilot study examining nursing students' tweets about nursing education posted to twitter. **Nurse education today**, Elsevier, v. 35, n. 2, p. 322–327, 2015.

BRUNO, G. Text mining and sentiment extraction in central bank documents. In: IEEE. **2016 IEEE International Conference on Big Data (Big Data)**. [S.l.], 2016. p. 1700–1708.

BRUNS, A.; BURGESS, J. E. The use of twitter hashtags in the formation of ad hoc publics. In: **Proceedings of the 6th European Consortium for Political Research (ECPR) General Conference 2011**. [S.l.: s.n.], 2011.

BRUNS, A.; LIANG, Y. E. Tools and methods for capturing twitter data during natural disasters. **First Monday**, University of Chicago, v. 17, n. 4, p. 1–8, 2012.

BRUNS, A.; WELLER, K.; BORRA, E.; RIEDER, B. Programmed method: Developing a toolset for capturing and analyzing tweets. **Aslib Journal of Information Management**, Emerald Group Publishing Limited, 2014.

CALLEGARI-JACQUES, S. **Bioestatística: Princípios e aplicações**. Artmed Editora, 2009. ISBN 9788536311449. Disponível em: ⟨https://books.google.com.br/books?id=nuaVLSCiAgsC⟩.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. **Sistemas Distribuídos-: Conceitos e Projeto**. [S.l.]: Bookman Editora, 2013.

DAYLEY, B. **Node.js, MongoDB, and AngularJS Web Development**. [S.l.]: Pearson Education, 2014. (Developer's Library). ISBN 9780133844344.

DELGADO, C. H.; VIANNA, C. E.; GUELPELI, M. V. C. Comparando sumários de referência humano com extratos ideais no processo de avaliação de sumários extrativos. **IADIS Ibero-Americana WWW/Internet**, v. 1, n. 1, p. 293–300, 2010.

DENKER, K. J.; MANNING, J.; HEUETT, K. B.; SUMMERS, M. E. Twitter in the classroom: Modeling online communication attitudes and student motivations to connect. **Computers in Human Behavior**, Elsevier, v. 79, p. 1–8, 2018.

FAYAD, M.; SCHMIDT, D. C. Object-oriented application frameworks. **Communications of the ACM**, ACM, v. 40, n. 10, p. 32–38, 1997.

FELDMAN, R.; SANGER, J. **The text mining handbook: advanced approaches in analyzing unstructured data**. [S.l.]: Cambridge university press, 2007.

FERNANDES, M. R. **Engenhos de Busca Distribuídos: Uma abordagem visando escalabilidade para Crawling e Indexação**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2001.

FIELDING, R.; GETTYS, J.; MOGUL, J.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. **Hypertext transfer protocol–HTTP/1.1**. [S.l.]: RFC 2616, june, 1999.

FRANÇA, T. C.; OLIVEIRA, J. Análise de sentimento de tweets relacionados aos protestos que ocorreram no brasil entre junho e agosto de 2013. In: **Proceedings of the III Brazilian Workshop on Social Network Analysis and Mining (BRASNAN)**. [S.l.: s.n.], 2014. p. 128–139.

GABARDO, A. C.; HATTORI, L. T.; BERNO, B. C. S.; GUTOSKI, M.; AGOSTINHO, W. R. U.; LOPES, H. S. Como mensurar a importância, inflência e a relevância de usuários do twitter? uma análise da interação dos candidatos à presidência do brasil nas eleições de 2018. **arXiv preprint arXiv:1902.11197**, 2019.

GIL, A. C. **Métodos e técnicas de pesquisa social**. [S.l.]: 6. ed. Ediitora Atlas SA, 2008.

GUELPELI, M. V. C. Cassiopeia: Um modelo de agrupamento de textos baseado em sumarização. **Niterói: Tese (Doutorado em Computação)-Univerisade Federal Fluminense**, 2012.

GUTIÉRREZ-MARTÍN, A.; TORREGO-GONZÁLEZ, A. The twitter games: media education, popular culture and multiscreen viewing in virtual concourses. **Information, Communication & Society**, Taylor & Francis, v. 21, n. 3, p. 434–447, 2018.

HERNANDEZ-SUAREZ, A.; SANCHEZ-PEREZ, G.; TOSCANO-MEDINA, K.; MARTINEZ-HERNANDEZ, V.; SANCHEZ, V.; PEREZ-MEANA, H. A web scraping methodology for bypassing twitter api restrictions. **arXiv preprint arXiv:1803.09875**, 2018.

JEONG, D.; JALALI, A. Who tweets in academia? an overview of twitter use in higher education. 2019.

JORDAN, K. From finding a niche to circumventing institutional constraints: Examining the links between academics' online networking, institutional roles, and identity-trajectory. **International Review of Research in Open and Distributed Learning**, Athabasca University, v. 20, n. 2, 2019.

KHALIL, S.; FAKIR, M. Rcrawler: An r package for parallel web crawling and scraping. **SoftwareX**, Elsevier, v. 6, p. 98–106, 2017.

KIM, Y.; HWANG, E.; RHO, S. Twitter news-in-education platform for social, collaborative, and flipped learning. **The Journal of Supercomputing**, Springer, v. 74, n. 8, p. 3564–3582, 2018.

KIMMONS, R.; VELETSIANOS, G.; WOODWARD, S. Institutional uses of twitter in us higher education. **Innovative Higher Education**, Springer, v. 42, n. 2, p. 97–111, 2017.

KOUZIS-LOUKAS, D. **Learning Scrapy**. Packt Publishing, 2016. ISBN 9781784390914. Disponível em: ⟨https://books.google.com.br/books?id=EF8dDAAAQBAJ⟩.

KWAK, H.; LEE, C.; PARK, H.; MOON, S. What is twitter, a social network or a news media? In: ACM. **Proceedings of the 19th international conference on World wide web**. [S.l.], 2010. p. 591–600.

LOPES, M. C. S. Mineração de dados textuais utilizando técnicas de clustering para o idioma português. **Rio de Janeiro: sn**, 2004.

LORENZO, E. M. **A utilização das redes sociais na educação**. [S.l.]: Clube de Autores (managed), 2013.

MARADEI, A. Twitter como esfera pública em momentos de protesto: Estudo da comunicação pela rede social nos movimentos de 2013, 2015 e 2016 no brasil. Universidade Metodista de Sao Paulo, 2018.

MARQUES, G. V.; MEDEIROS, C. R.; PEREIRA, J. Q. Análise comparativa de desempenho de aplicação java com persistência em banco de dados mysql e mongodb. **Anais do Encontro de Computação do Oeste Potiguar ECOP/UFERSA (ISSN 2526-7574)**, n. 2, 2018.

MATHIAK, B.; ECKSTEIN, S. Five steps to text mining in biomedical literature. In: **Proceedings of the second European workshop on data mining and text mining in bioinformatics**. [S.l.: s.n.], 2004. v. 24.

MINETTO, E. L. Frameworks para desenvolvimento em php. **São Paulo: Novatec**, 2007.

MONGODB. **What is MongoDB**. 2019. Acesso: 01 jul 2019. Disponível em: ⟨https://www.mongodb.com/what-is-mongodb⟩.

MORAES, W. B. **Construindo aplicações com NodeJS**. [S.l.]: Novatec Editora, 2018.

MORAIS, E. A. M.; AMBRÓSIO, A. P. L. Mineração de textos. **Relatório Técnico–Instituto de Informática (UFG)**, 2007.

MURTHY, D. **Twitter**. Wiley, 2018. (Digital Media and Society). ISBN 9781509512539. Disponível em: ⟨https://books.google.com.br/books?id=LJ1PDwAAQBAJ⟩.

PEREIRA, C. **Aplicações web real-time com Node.js**. [S.l.]: Casa do Código, 2014. ISBN 9788566250930.

PRUSS, D.; FUJINUMA, Y.; DAUGHTON, A. R.; PAUL, M. J.; ARNOT, B.; SZAFIR, D. A.; BOYD-GRABER, J. Zika discourse in the americas: A multilingual topic analysis of twitter. **PloS one**, Public Library of Science, v. 14, n. 5, p. e0216922, 2019.

RECUERO, R. Contribuições da análise de redes sociais para o estudo das redes sociais na internet: o caso da hashtag# tamojuntodilma e# calaabocadilma. **Fronteiras-estudos midiáticos**, v. 16, n. 2, p. 60–77, 2014.

RECUERO, R. d. C.; ZAGO, G. d. S.; SOARES, F. B. Midia social e filtros-bolha nas conversações políticas no twitter. **Associação Nacional de Programas de Pós-Graduação em Comunicação. Encontro Anual (COMPÓS).(26.: 2017 jun. 06-09: São Paulo, SP).[Anais]. São Paulo: Faculdade Cásper Líbero, 2017.**, 2017.

SAEED, Z.; ABBASI, R. A.; MAQBOOL, O.; SADAF, A.; RAZZAK, I.; DAUD, A.; ALJOHANI, N. R.; XU, G. What's happening around the world? a survey and framework on event detection techniques on twitter. **Journal of Grid Computing**, Springer, p. 1–34, 2019.

SANTAELLA, L.; MORAIS, R. L. Redes sociais digitais: a cognição conectiva do twitter. PAULUS Publications, 2010.

SANTANA, C. L.; COUTO, E. S. *et al.* Estratégias de visibilidade e ações docentes no twitter. **Educação**, Universidade Federal de Santa Maria, v. 42, n. 2, p. 435–450, 2017.

SANTOS, M. C. **LTWEET: Ferramenta de extração de dados do Twitter. Versão beta. Labcom Digital**. 2019. Acesso em: 01 out 2019. Disponível em: ⟨https://www.labcomdata.com.br/⟩.

SARGIANI, V. *et al.* Identificação de padrões em textos de mídias sociais utilizando redes neurais e visualização de dados. Universidade Presbiteriana Mackenzie, 2018.

SILVIUS, A.; KAVALIAUSKAITE, R. Value of online social networks from the perspective of the user. **Journal of International Technology and Information Management**, v. 23, n. 2, p. 1, 2014.

SOARES, F. d. A. **Categorização Automática de Textos Baseada em Mineração de Textos**. Tese (Doutorado) — PUC-Rio, 2013.

STEFANIDIS, A.; VRAGA, E.; LAMPRIANIDIS, G.; RADZIKOWSKI, J.; DELAMATER, P. L.; JACOBSEN, K. H.; PFOSER, D.; CROITORU, A.; CROOKS, A. Zika in twitter: temporal variations of locations, actors, and concepts. **JMIR public health and surveillance**, JMIR Publications Inc., Toronto, Canada, v. 3, n. 2, p. e22, 2017.

SULLIVAN, D. The need for text mining in business intelligence. **DM REVIEW**, POWELL PUBLISHING INC, v. 10, p. 12–16, 2000.

TAN, A.-H. *et al.* Text mining: The state of the art and the challenges. In: SN. **Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases**. [S.l.], 1999. v. 8, p. 65–70.

TANG, Y.; HEW, K. F. Using twitter for education: Beneficial or simply a waste of time? **Computers & Education**, Elsevier, v. 106, p. 97–118, 2017.

TORRES, C. **A bíblia do marketing digital: tudo o que você queria saber sobre marketing e publicidade na internet e não tinha a quem perguntar**. [S.l.]: Novatec Editora, 2018.

TRIVIÑOS, A. N. S. Pesquisa qualitativa. **Introdução à pesquisa em ciências sociais: a pesquisa qualitativa em educação. São Paulo: Atlas**, p. 116–173, 1987.

TWITTER. **Q1 2019 Shareholder Letter**. 2019a. Acesso em: 01 jul 2019a. Disponível em: ⟨https://s22.q4cdn.com/826641620/files/doc_financials/2019/q1/Q1-2019-Shareholder-Letter.pdf⟩.

TWITTER. **Sobre**. 2019b. Acesso em: 01 jul 2019b. Disponível em: ⟨https://about.twitter.com/pt/⟩.

TWITTER. **Docs**. 2019c. Acesso em: 01 jul 2019c. Disponível em: ⟨https://developer.twitter.com/en/docs⟩.

VILELA, R. B.; RIBEIRO, A.; BATISTA, N. A. Os desafios do mestrado profissional em ensino na saúde: uso da nuvem de palavras no apoio à pesquisa qualitative. **CIAIQ2018**, v. 2, 2018.

WIVES, L. K. Tecnologias de descoberta de conhecimento em textos aplicadas à inteligência competitiva. **Exame de Qualificação EQ-069, PPGC-UFRGS**, 2002.

APÊNDICE A – COLETAS PARA REALIZAÇÃO DE TESTES DE VALIDAÇÃO

Tabela 9: Resultados de coleta para realização dos testes de validação.

IDENTIFICAÇÃO DA COLETA	DMI-TCAT	API - BR	API – EUA	Crawler – BR	Crawler – EUA
170114	16083	16102	16103	26958	44737
170214	15780	15856	15857	26312	41463
181214	16083	16111	16115	26107	40277
181244	16083	16105	16107	29248	40915
181314	16087	16178	16180	25221	40494
181344	16183	16229	16234	25268	39996
181414	16187	16194	16173	25556	39643
181444	16083	16100	16100	26093	39705
181514	16084	16090	16090	26060	39077
181544	15780	15849	15847	25946	38754
181614	15681	15783	15785	25848	39665
181644	15678	15694	15695	26157	39105
181714	15580	15660	15659	25724	39227
181744	15578	15595	15594	29566	39355
181914	15076	15182	15182	26026	40330
182114	15579	15595	15584	26997	43525
182144	15581	15627	15606	27096	44049
182244	15680	15713	15711	27325	42846
182344	15782	15805	15803	27302	43664
191044	16085	16162	16172	25171	40399
191144	16689	16713	16707	24470	40709
191244	16891	16911	16908	25956	41156
191344	16790	16834	16829	26336	41102
191414	16789	16826	16832	26593	39987
191444	16792	16856	16884	30131	40333
191544	16992	16993	16994	26650	39456
191614	16790	16802	16810	26437	40003
191644	16588	16616	16620	26198	40634
191714	16386	16404	16419	29772	39680
191744	16084	16180	16185	25897	39957
191914	16489	16492	16486	26088	40326
192014	16285	16279	16279	26894	42661
192044	16487	16583	16587	26912	43398
192314	16387	16452	16445	27420	44876
200014	16388	16470	16461	27839	45359

IDENTIFICAÇÃO DA COLETA	DMI-TCAT	API - BR	API – EUA	Crawler – BR	Crawler – EUA
200044	16286	16385	16383	28050	40296
200114	16286	16365	16363	27801	45417
200144	16386	16394	16394	27767	46341
200214	16388	16404	16404	26602	46792
200244	16286	16326	16326	25498	46686
200314	16184	16231	16230	24736	46405
200344	16085	16120	16124	24442	46542
200414	15982	16062	16061	24240	46736
200544	15578	15624	15624	32752	46281
200614	15578	15634	15634	23925	46301
200644	15680	15731	15718	32784	46044
200714	15782	15832	15831	23963	45392
200744	16085	16129	16145	24097	45551
200814	16285	16333	16354	24789	44320
200914	16489	16501	16489	26023	43115
201014	16389	16440	16436	26644	42605
201044	16590	16678	16668	26837	42470
201144	16792	16811	16809	26535	42441
201444	16385	16387	16401	26893	42911
201514	16286	16355	16353	26772	42731
201544	16386	16386	16383	27077	43031
201614	16490	16529	16533	31067	43578
201644	16487	16550	16549	27236	43581
201714	16591	16620	16621	26938	43689
201744	16487	16557	16561	27214	38462
201844	16591	16653	16645	30956	43798
210244	15580	15613	15614	25941	47337
210314	15477	15512	15512	25324	47097
210344	15476	15542	15542	24752	47454
210414	15479	15534	15533	24361	47319
220044	15680	15729	15729	27961	45176
220114	15681	15742	15735	27898	46108
220144	15580	15673	15673	27955	46367
220214	15681	15726	15724	26906	46427
220244	15579	15650	15649	32553	46610
220314	15479	15561	15562	24721	46612
220344	15478	15574	15573	24598	47007
220414	15477	15510	15509	32829	46866
220444	15478	15489	15491	24117	46808
220514	15477	15535	15535	24058	46650

IDENTIFICAÇÃO DA COLETA	DMI-TCAT	API - BR	API – EUA	Crawler – BR	Crawler – EUA
220544	15579	15585	15586	32952	46314
220614	15578	15654	15654	24112	46131
220644	15580	15689	15689	24172	46048
220714	15781	15784	15780	24648	44994
220744	15881	15925	15928	24968	44146
220844	16083	16093	16092	26066	41798
220914	16185	16189	16189	26553	40830
220944	16186	16274	16274	26907	40179
221014	16183	16211	16206	26826	40956
221044	16183	16246	16249	26609	40184
221114	16386	16482	16492	26657	40544
221144	16588	16606	16605	25719	40401
221214	16487	16534	16535	26103	40904
221244	16386	16428	16431	26289	41276
230114	16084	16103	16089	28178	44358
230144	15982	16048	16048	28276	46207
230214	15982	15999	16000	27620	46883
230244	15983	15988	15993	26392	46880
230314	15981	15982	15983	32961	47334
230344	15882	15973	15971	24919	47414
230414	15982	16006	16006	24560	47786
230444	15882	15977	15978	24479	47643
230514	15982	16005	16004	24264	47044
230544	15981	15978	15977	24248	47101
230614	15882	15973	15970	24366	47079
230744	16287	16318	16317	25838	44705
230814	16386	16474	16473	26623	43010
230844	16588	16611	16613	27066	41548
230914	16690	16734	16739	26875	41113
230944	16589	16632	16633	27110	41642
231014	16588	16677	16676	26381	41495
231114	16589	16619	16623	26440	40218
231144	16285	16339	16337	26008	40948
231214	15984	15993	15987	26176	41702
231244	15379	15388	15392	30566	41149

Fonte: Próprio autor.

APÊNDICE B – *RANKING* DE COLETAS

Tabela 10: *Ranking* das coletas para realização dos testes de validação.

IDENTIFICAÇÃO DA COLETA	DMI-TCAT	API - BR	API – EUA	Crawler – BR	Crawler – EUA
170114	1	2	3	4	5
170214	1	2	3	4	5
181214	1	2	3	4	5
181244	1	2	3	4	5
181314	1	2	3	4	5
181344	1	2	3	4	5
181414	2	3	1	4	5
181444	1	2,5	2,5	4	5
181514	1	2,5	2,5	4	5
181544	1	3	2	4	5
181614	1	2	3	4	5
181644	1	2	3	4	5
181714	1	3	2	4	5
181744	1	3	2	4	5
181914	1	2,5	2,5	4	5
182114	1	3	2	4	5
182144	1	3	2	4	5
182244	1	3	2	4	5
182344	1	3	2	4	5
191044	1	2	3	4	5
191144	1	3	2	4	5
191244	1	3	2	4	5
191344	1	3	2	4	5
191414	1	2	3	4	5
191444	1	2	3	4	5
191544	1	2	3	4	5
191614	1	2	3	4	5
191644	1	2	3	4	5
191714	1	2	3	4	5
191744	1	2	3	4	5
191914	2	3	1	4	5
192014	3	1,5	1,5	4	5
192044	1	2	3	4	5
192314	1	3	2	4	5
200014	1	3	2	4	5
200044	1	3	2	4	5

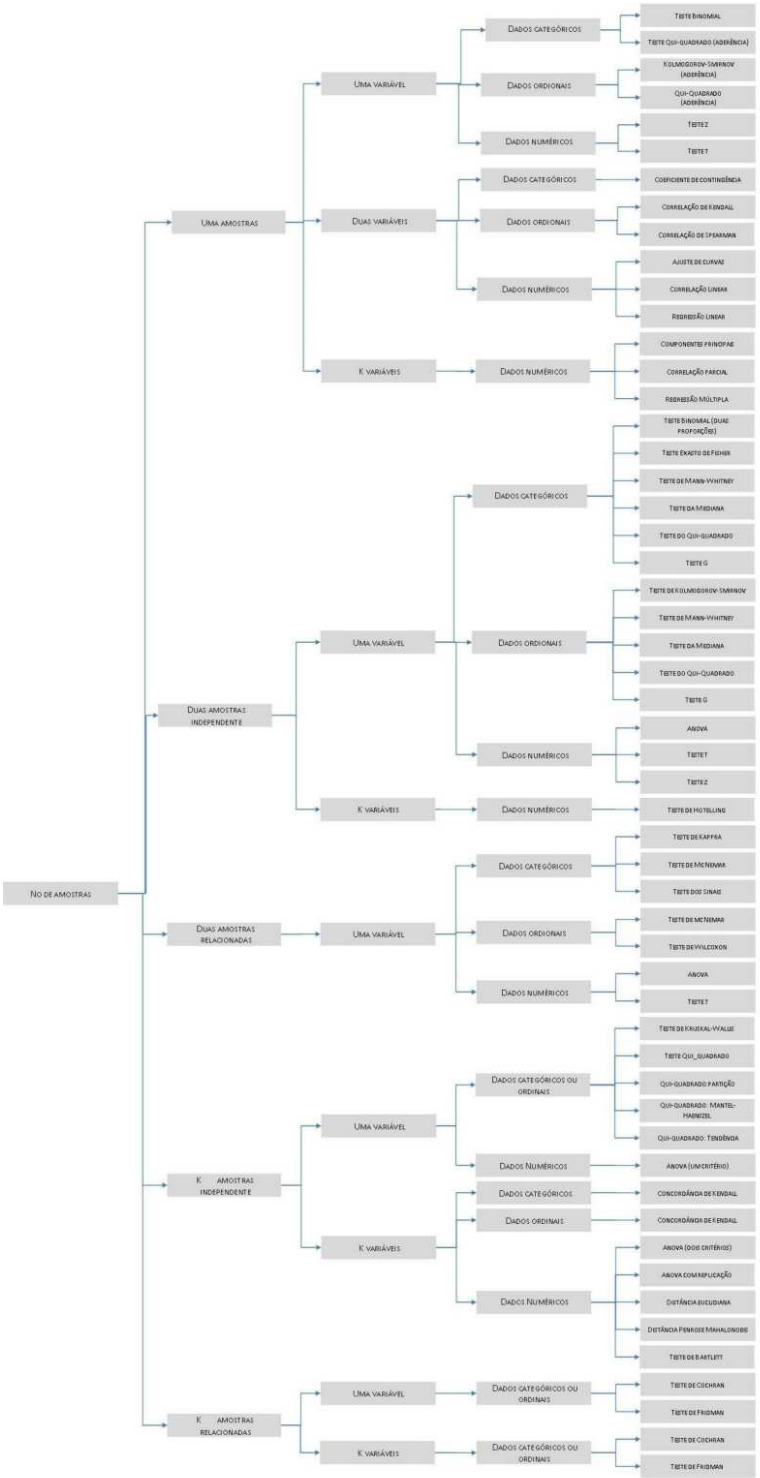
IDENTIFICAÇÃO DA COLETA	DMI-TCAT	API - BR	API – EUA	Crawler – BR	Crawler – EUA
200114	1	3	2	4	5
200144	1	2,5	2,5	4	5
200214	1	2,5	2,5	4	5
200244	1	2,5	2,5	4	5
200314	1	3	2	4	5
200344	1	2	3	4	5
200414	1	3	2	4	5
200544	1	2,5	2,5	4	5
200614	1	2,5	2,5	4	5
200644	1	3	2	4	5
200714	1	3	2	4	5
200744	1	2	3	4	5
200814	1	2	3	4	5
200914	1,5	3	1,5	4	5
201014	1	3	2	4	5
201044	1	3	2	4	5
201144	1	3	2	4	5
201444	1	2	3	4	5
201514	1	3	2	4	5
201544	2,5	2,5	1	4	5
201614	1	2	3	4	5
201644	1	3	2	4	5
201714	1	2	3	4	5
201744	1	2	3	4	5
201844	1	3	2	4	5
210244	1	2	3	4	5
210314	1	2,5	2,5	4	5
210344	1	2,5	2,5	4	5
210414	1	3	2	4	5
220044	1	2,5	2,5	4	5
220114	1	3	2	4	5
220144	1	2,5	2,5	4	5
220214	1	3	2	4	5
220244	1	3	2	4	5
220314	1	2	3	4	5
220344	1	3	2	4	5
220414	1	3	2	4	5
220444	1	2	3	4	5
220514	1	2,5	2,5	4	5
220544	1	2	3	4	5

IDENTIFICAÇÃO DA COLETA	DMI-TCAT	API - BR	API – EUA	Crawler – BR	Crawler – EUA
220614	1	2,5	2,5	4	5
220644	1	2,5	2,5	4	5
220714	2	3	1	4	5
220744	1	2	3	4	5
220844	1	3	2	4	5
220914	1	2,5	2,5	4	5
220944	1	2,5	2,5	4	5
221014	1	3	2	4	5
221044	1	2	3	4	5
221114	1	2	3	4	5
221144	1	3	2	4	5
221214	1	2	3	4	5
221244	1	2	3	4	5
230114	1	3	2	4	5
230144	1	2,5	2,5	4	5
230214	1	2	3	4	5
230244	1	2	3	4	5
230314	1	2	3	4	5
230344	1	3	2	4	5
230414	1	2,5	2,5	4	5
230444	1	2	3	4	5
230514	1	3	2	4	5
230544	3	2	1	4	5
230614	1	3	2	4	5
230744	1	3	2	4	5
230814	1	3	2	4	5
230844	1	2	3	4	5
230914	1	2	3	4	5
230944	1	2	3	4	5
231014	1	3	2	4	5
231114	1	2	3	4	5
231144	1	3	2	4	5
231214	1	3	2	4	5
231244	1	2	3	4	5

Fonte: Próprio autor.

ANEXO A – ESCOLHA DA TÉCNICA DE TESTE ESTATÍSTICO A PARTIR DO NÚMERO DE AMOSTRAS

Figura 13: Fluxograma para escolha de Teste Estatístico



Fonte: (CALLEGARI-JACQUES, 2009)