

Documentação

# Observatório UFSJ

São João del Rei - Minas Gerais,  
08 de Novembro de 2018

Versão 1.0.0

# Ficha Técnica

**Equipe:**

Higor Henrique Alves

**Público Alvo:**

Este manual destina-se a estudantes e pesquisadores que estão interessados em minerar dados para avaliar informações sobre os candidatos presidenciais ou mesmo de outros políticos na rede social Twitter.

---

Versão 1.0.0 - São João del Rei, Outubro de 2018

---

# Introdução

O projeto apresentado neste documento, refere-se às primeiras etapas para a implementação de uma aplicação web denominada “Observatório UFSJ”. O objetivo deste projeto é coletar dados dos candidatos à presidência da república do Brasil no ano de 2018, e realizar análises, como, definir a personalidade dos candidatos e quais são os assuntos mais abordados pelos mesmos.

O local escolhido para coletar os dados mencionados foi a rede social Twitter, pois nela os candidatos expõe de forma resumida suas opiniões.

Usando a API do Twitter que é fornecida para qualquer pessoa que justifique o seu cadastro no site de desenvolvedores, o uso da mesma tornou-se essencial para poder coletar os dados de forma rápida e estruturada já que ela nos retorna um JSON onde pode-se trabalhar com as informações.

Também foram usadas as API's do Watson, Personality Insights para avaliar a personalidade do candidato, Translate para traduzir os tweets dos candidatos, e Natural Language Understanding para verificar os sentimentos de cada tweet feito.

Todas essas API's trabalham juntas com um banco de dados não relacional (MongoDB) para armazenagem e consultas dos dados. A linguagem de programação utilizada foi o JavaScript baseando-se na arquitetura Flux que em conjunto com a API do Twitter realizou a coleta de informações.

Os candidatos a presidência do ano de 2018 abordados pelo projeto são: Jair Messias Bolsonaro e Fernando Haddad, tendo como vices respectivamente, General Hamilton Mourão e Manuela D'Ávila.

Os programas que serão empregados no processo do projeto irão viabilizar a mineração dos dados provenientes da web relacionados aos candidatos por meio de tweets ( Postagem em formato de texto

no site do Twitter). Até o presente momento já foram coletados 30.000 tweets e a previsão até a entrega do final do projeto é de 100.000 tweets.

## Visão geral deste Documento

Este documento fornece as informações sobre os objetivos e desenvolvimento do projeto, abrangendo as fases de coleta, armazenamento, tratamento e exibição dos dados. Etapas essas que serão descritas e detalhadas em seções posteriores. Fornece detalhes e referências sobre recursos e ferramentas utilizadas como APIs e Bibliotecas. E ainda, as propostas de análises a serem realizadas na etapa final do projeto.

## Referências

Nesta seção estão relatadas algumas nas bibliotecas, API`s e suas respectivas documentações, visto que o sistema faz uso de uma vasta gama de documentações de aplicações de terceiros, as quais algumas precisam de cadastro e pagamento.

Documentos relacionados ao Observatório UFSJ e/ou mencionados nas seções posteriores estão descritos a seguir:

### 1. **Título:** Twitter API

#### a. **Descrição:**

Para a coleta de dados no twitter fez-se necessário o uso de uma API. Para melhor compreensão, a API (interface de Programação de Aplicativos) é um conjunto de rotinas e padrões de programação que permite acesso a um aplicativo de software ou plataforma

baseada na web. Logo, a API permite o acesso aos dados do Twitter de forma estruturada.

Para ter acesso a essa API, foi preciso dispor de um token (chave) cedido pelo próprio Twitter para realizar a autenticação. Essa API retorna códigos de status HTTP, além de mensagens e códigos de erro baseados em JSON.

Dentre os Tokens criados pelo Twitter utilizou-se dois tipos de chaves de autenticação e dois tokens:

- Consumer Key: é a chave de identificação da aplicação para coletar dados do twitter;
- Consumer Secret: é a chave de autenticação da aplicação para liberar o processo de coleta de dados;
- Access Token key: é o Token de identificação da aplicação para acessar dados do Twitter;
- Access Token Secret: é o Token de autenticação para acessar as informações do Twitter.

Usando códigos na linguagem de programação JavaScript foi implementada uma aplicação para realizar a chamada da API do Twitter.

- b. Documentação: <https://developer.twitter.com/en.html>

## 2. **Título:** Twitter for Node Js

- a. **Descrição:** Biblioteca assíncrona para consumir a API do Twitter.
- b. **Documentacao:** <https://www.npmjs.com/package/twitter>

## 3. **Título:** Mongoose ODM

- a. **Descrição:** Mongoose ODM ((Object Data-Modeling) é uma biblioteca feita em JavaScript para facilitar a comunicação com o servidor do MongoDB. Essa biblioteca ODM para MongoDB gerência relacionamento entre dados e fornece

validação de esquema. No projeto, seu uso se deu na tradução dos objetos coletados para o MongoDB.

- b. **Documentação:** <https://mongoosejs.com/docs/guide.html>

#### 4. **Título:** Watson - Language Translator

- a. **Descrição:** O Neural Machine Translation é fornecido como padrão para cada par de idiomas. A customização de Corpus permite criar seus próprios modelos de tradução que representam termos específicos de uma região ou indústria. Traduz o conteúdo de maneira instantânea para diversos idiomas. É possível traduzir documentos, aplicativos e websites, além de criar chatbots que falam diversos idiomas.

- b. **Documentação:**

<https://console.bluemix.net/catalog/services/language-translator>

#### 5. **Título:** Watson - Natural Language Understanding.

- a. **Descrição:** Análise textos para extrair metadados do conteúdo, tais como conceitos, entidades, palavras-chave, categorias, impressões, emoções, relações e funções semânticas, usando o entendimento de língua natural. Com os modelos de anotação customizada desenvolvidos usando o Watson Knowledge Studio, identifique entidades e relações específicas de domínio/segmento de mercado em texto não estruturado.

- b. **Documentação:**

<https://console.bluemix.net/catalog/services/natural-language-understanding>

#### 6. **Título:** Watson - Tone Analyzer

- a. **Descrição:** As pessoas mostram vários sinais, como alegria, tristeza, raiva e amabilidade, nas comunicações diárias. Tais sinais podem ter impacto na efetividade da comunicação em

diferentes contextos. O Tone Analyzer alavanca a análise linguística cognitiva para identificar uma variedade de sinais nos níveis de sentença e de documento. Este insight pode, então, ser usado para refinar e melhorar as comunicações. Ele detecta três tipos de sinais, incluindo emoção (raiva, repugnância, medo, alegria e tristeza), propensões sociais (sinceridade, escrupulosidade, extroversão, amabilidade e faixa emocional) e estilos de texto (analítico, confiante e hesitante).

b. **Documentação:**

<https://console.bluemix.net/catalog/services/tone-analyzer>

7. **Título:** Watson - Personality Insights

- a. **Descrição:** O Personality Insights fornece insights de dados de mídia social e transacionais para identificar traços psicológicos que determinam decisões, intenção e traços comportamentais; utilizados para melhorar as taxas de conversão.

b. **Documentação:**

<https://console.bluemix.net/docs/services/personality-insights/getting-started.html>

## Descrição geral do sistema

O sistema irá gerar uma plataforma de consultas pré-configuradas para que usuários comuns possam realizar buscas relacionadas aos tweets dos candidatos e de palavras-chaves que fazem alusão aos mesmos. Disponibilizará uma opção para visualizar Tweets cadastrados dos candidatos e por meio das palavras-chaves as publicações de outros usuários que participam da rede social.

Para os tweets coletados e já armazenados na base de dados, o sistema irá fornecer variados relatórios que apresentaram

informações associadas aos candidatos como: quais os assuntos mais abordados por eles e outros usuários do Twitter, os sentimentos mais usados na hora das postagem dos tweets, personalidade dos candidatos, dentre outros ainda a serem testados.

O sistema não se caracteriza como independente e totalmente contido, ou seja, há a comunicação entre suas partes e com as API's mencionadas nesse documento. Essa interação facilita o processamento da informações para melhor utilização do usuário final.

## Coleta dos Dados

A primeira parte do projeto abrange a coleta de dados, ou seja, extração das informações da web realizadas por meio da API disponibilizada pelo site do Twitter. Com relação à aplicação a ser desenvolvida, esta irá recolher as informações de tweets publicados nos perfis dos quatro candidatos: Jair Bolsonaro, Fernando Haddad, General Hamilton Mourão e Manuela D'Ávila, bem como fazer uma busca por determinadas palavras-chaves que fazem referência aos mesmos.

As palavras-chaves a serem utilizadas estão correlacionadas aos candidatos das eleições, sendo algumas delas: Bolsonaro, Haddad, #bolsonaropresidente, #jairbolsonaro, #haddad, #haddadpresidente; Dentre os diversos elementos fornecidos pela documentação da API, foram selecionadas apenas as informações que seriam relevantes para serem apresentadas por meio de relatórios no site "Observatório UFSJ".

Para recolher as informações de tweets publicados nos perfis dos quatro candidatos foi utilizado a "API Timelines". E para as



palavras-chaves foi utilizado a “Api Search” que retornar uma coleção de [Tweets](#) correspondente a uma consulta específica.

Ao fazer uma solicitação referente ao perfil de determinado candidato, coletamos as informações sobre a conta do candidato e os tweets publicados por ele, utilizando a API Timelines que permite o retorno de dados do twitter. Este por sua vez devolve as informações em um documento no formato JSON (dados brutos). Por conseguinte, uma função chamada .map (da linguagem usada) trata de percorrer todo o documento JSON. Assim é feito um pré-processamento onde os dados são filtrados, buscando por meio dessa varredura somente os valores de alguns campos já pré-estabelecidos e configurados na aplicação. Os dados resultantes dessa filtragem são armazenados diretamente na coleção ‘Candidatos’ no Banco de Dados MongoDB. Os campos pré-estabelecidos mencionados acima são: Nome do usuário, Screen name, Localização, Verificador, Contador de favoritos, Url imagem de perfil e Url imagem de painel, Texto Completo do tweet, Coordenadas geográfica, Contador de retweets, Contagem de favoritos do tweet, Entidade e Localidade. Esses campos ou atributos, estão descritos no quadro abaixo:

CAMPOS PRÉ-ESTABELECIDOS PARA COLETA POR PERFIL DO CANDIDATO		
Atributo	Tipo	Descrição
user_name	String	Nome do usuário (candidato). Definido pelo próprio usuário no Twitter. Normalmente limitado a 20 caracteres.
screen_name	String	Nome identificador com o que usuário (o candidato) do Twitter se identifica. (Máximo 15 caracteres. É um valor único dentro do Twitter, por isso serve como identificador.

location	string	Local definido pelo usuário para o perfil de sua conta (cidade)
verified	boolean	Indicado por true caso o usuário tenha uma conta verificada. Contas verificadas no Twitter são aquelas que são determinadas como de interesse público.
followers_count	int	O número de seguidores que a conta do candidato tem.
profile_image_url_https	string	Um URL baseado em HTTP apontando para a imagem do perfil do usuário (candidato).
profile_banner_url	string	O URL com base em HTTPS que aponta para a representação da web padrão do banner do perfil enviado pelo usuário.
id	Int	Identificador do tweet, já estabelecido pelo Twitter.
coordinates	object	O objeto fornecerá a longitude e latitude (um array [long, lat]) com as coordenadas geográficas do tweet.
full_text	string	O texto do tweet.
retweet_count	int	Número de vezes que o tweet foi retweetado.
favorite_count	int	Indica aproximadamente quantas vezes o Tweet foi <a href="#">apreciado</a> pelos usuários do Twitter.
entities	object	Entidades que foram analisadas fora do texto do Tweet
description	string	Descrição HTML para o link do Tweet
display_text_range	object	Uma matriz de dois índices de ponto de código unicode, identificando o início inclusivo e o fim exclusivo do conteúdo exibível do Tweet.

Já para realizar a busca por palavras-chave é usada a API Search que retorna uma coleção de [Tweets](#) relevantes que correspondem a palavra pesquisada. Logo, o processo se assemelha a solicitação por candidato, ou seja, o Twitter retorna um JSON (dados brutos) com todos os tweets, de qualquer usuário da rede social, que contenha a determinada palavra-chave. Dessa forma a função map é chamada e percorre todo o documento JSON para procurar somente os dados dos campos pré-estabelecidos e já configurados na aplicação. Assim é feito o processamento e os dados são filtrados. O armazenamento dos dados resultantes vão direto para a coleção 'PalavraChave' no Banco de Dados MongoDB.

Os campos pré-estabelecidos para a busca por palavra-chave são: Palavra-chave; Identificador; Screen name, Texto Completo; Entidades; Contador de retweets; Localidade; Nome do usuário; Contagem de seguidores, Url imagem de perfil; Url imagem de painel.

CAMPOS PRÉ-ESTABELECIDOS PARA COLETA POR PALAVRA-CHAVE		
Atributo	Tipo	Descrição
palavra_chave	string	A palavra-chave pesquisada. Campo adicionado na aplicação.
id	int	Identificador do tweet, gerado pelo Twitter.
screen_name	string	Nome identificador com o que usuário do Twitter se identifica. O nome do usuário que mencionou a palavra-chave.
full_text	string	O texto do tweet.
retweet_count	int	Número de vezes que o tweet foi retweetado.

followers_count	int	O número de seguidores que a conta tem.
name	string	Nome do usuário. Definido pelo próprio usuário no Twitter.
entities		Entidades que foram analisadas fora do texto do Tweet.
location	string	Localização
profile_image_url_https	string	URL baseado em HTTP apontando para a imagem do perfil do usuário.
profile_banner_url	string	O URL com base em HTTPS que aponta para a representação da web padrão do banner do perfil enviado pelo usuário.

Para a realização da coleta alguns aspectos importantes foram considerados, como o limite de taxa da API determinado pelo Twitter. Esse limite configura o número de solicitações que um usuário pode fazer em um determinado período de tempo para evitar uma demanda incontrolável nos servidores da rede social.

O limite de taxa permitido pelo Twitter configura 180 solicitações dentro de um intervalo de 15 minutos para consumo da API Timelines e 100 solicitações a cada 15 minutos para a API Search. Dessa forma as requisições algumas vezes precisaram seguir tais recomendações.

# Armazenamento de Dados

Devido a estrutura e a quantidade dos dados coletados, o banco de dados selecionado foi o MongoDB. Este é um banco de dados de código aberto, gratuito, de alta performance, sem esquemas e orientado à documentos. Outro fator que justifica o uso do mongoDB é que a coleta dos tweets geram um grande número de dados, e como a quantidade de tweets a serem coletados é relativamente grande, o MongoDB se torna a ferramenta mais viável. Para modelar os dados da aplicação, foi utilizada a biblioteca mongoose do node.js. O mongoose é uma biblioteca ODM (Object Data-Modeling) para MongoDB que gerencia o relacionamento entre os dados e fornece validação de esquema. No projeto, seu uso se deu na tradução dos objetos coletados para o MongoDB.

No processo de armazenamento, a API do twitter extrai dos tweets dos candidatos os dados, como o texto postado, localidade, quantidade de retweets entre outros. Ao inserir a biblioteca mongoose é criado um schema no banco e os dados extraídos são alocados em formato de documentos nas coleções criadas. Dessa extração realizada foram criadas as seguintes coleções para armazenar os dados: Candidatos, Candidatos\_personalidade, PalavraChave, TextoCandidatos.

Coleções usadas no MongoDB	
Coleções	Descrição
PalavraChave	Registra todos os tweets coletados quando se usa a API Search que procura por palavra-chave.
Candidatos	Registra todos os tweets coletados quando se usa a API Timeline que retorna o perfil do candidato.

Candidatos_person alidade	Registra a personalidade dos candidatos, por meio da análise dos textos de seus tweets.
TextoCandidatos.	Registra um conjunto de textos retirados dos tweets para serem usados em análises pelo Watson. O objetivo principal dessa coleção é guardar os textos agrupados que precisam ser traduzidos de português para inglês. Quando traduzidos os dados da coleção são alterados para a os textos em inglês.

Devido a necessidade de armazenamento de informações resultantes das análises do Watson - NLU referentes a cada tweet, foram adicionados campos na coleção 'Candidatos' que estão descritos na tabela abaixo.

<b>Campos inseridos na coleção Candidatos</b>		
sentiment:	object	recebe o resultado na análise da API NLU
keywords	object	recebe o resultado na análise da API NLU
entitiesNLU	object	recebe o resultado na análise da API NLU
categories	object	recebe o resultado na análise da API NLU

# Watson - API's

O "IBM Watson Platform" é uma Plataforma de Inteligência Artificial composta por diversas API's que auxiliam a construção de sistemas cognitivos. No projeto os recursos do Watson foram utilizados para realizar uma análise de sentimento, ou seja, extrair informações a partir dos textos dos tweets publicados para analisar a personalidade dos candidatos, bem como o sentimento de cada tweet individualmente.

Durante o desenvolvimento do projeto usamos API'S para processamento de linguagem natural, para tradução de textos e para análise de personalidade. Todas serão descritas abaixo.

- **NLU**

Do inglês Natural Language Understanding, essa API é utilizada para extrair metadados de determinado conteúdo, tais como conceitos, entidades, palavras-chave, categorias, impressões, emoção, relações e funções semânticas, usando o entendimento de língua natural.

Após a coleta e armazenamento dos tweets na coleção Candidatos, os textos dos tweeters serão selecionados e agrupados por meio do recurso aggregate do MongoDB e armazenados na coleção TextoCandidatos que possui os seguintes atributos: screen\_name, texto. A partir das informações dessa coleção, a API NLU busca identificar o sentimento do candidato (se é positivo ou negativo), as palavras-chaves, a categoria (se é um texto falando de política, economia, mídia, etc) , a entidade, que busca reconhecer empresas, estados nacionais entre outros dentro do texto. Os resultados da

avaliação da API são armazenados na coleção Candidatos (nos atributos sentiment, keywords, entitiesNLU, categories).

- **Watson Translator**

API utilizada para fazer traduções entre idiomas. No contexto dessa aplicação foi utilizada no processo de tradução dos textos dos tweets dos candidatos para o inglês para que a API de análise do sentimento faça a análise, visto que as APIs do Watson utilizadas não realizam análise em Português.

- **Personality Insights**

O Personality Insights fornece insights que podem identificar traços psicológicos que determinam decisões, intenção e traços comportamentais. Nessa aplicação os dados extraídos pela API vão para a coleção CandidatosPersonalidade.

Os tweets para serem analisados pelo Watson no contexto desta API passará pelo seguinte processo. Dos tweets já coletados e armazenados na coleção Candidatos serão selecionados os textos de determinado candidato, agrupados por meio do recurso aggregate do MongoDB e armazenados na coleção TextoCandidatos (que possui os seguintes atributos: screen\_name, texto). A API Translator irá traduzir esse texto de português para inglês (pois o Watson só faz a análise em inglês). Por conseguinte, a API Personality Insights é executada e realiza uma análise sobre todo o texto já em inglês, retornando os sentimentos relativo ao mesmo. O sentimento retornado refere-se então a personalidade do candidato estabelecido. Essas informações serão armazenados na coleção CandidatoPersonalidade que possui os atributos relacionados abaixo. Ou seja, esses atributos vão registrar as informações retornadas pela API Personality Insights.



<b>Atributos da coleção CandidatoPersonalidade</b>	
<b>Coleções</b>	<b>Descrição</b>
screen_name	O campo screen_name com o nome do candidato; o campo Personality com os aspectos, fatores da personalidade.
personality	o campo PErsonality retorna os aspectos, fatores da personalidade. Uma matriz de objetos que fornece resultados detalhados para os cinco grandes características de personalidade (dimensões e facetas dentro da psicologia ) inferidas a partir do texto de entrada.
needs	O campo needs com informações sobre os níveis de compatibilidade que os textos dos candidatos tem com algumas virtudes ou emoções como liberdade, excitação entre outras;
values	O campo values descreve fatores motivadores que influenciam a tomada de decisão de uma pessoa. O modelo inclui cinco valores: Autotranscendência / Ajudar os outros, Conservação / Tradição, Hedonismo / Sentir prazer na vida, Auto-aperfeiçoamento / Alcançar o sucesso, e Aberto à mudança / Excitação;
consumption_preferences	o campo consumption_preferences se refere as preferência de consumo ou de assunto abordado a partir dos textos recebidos como entrada.

<b>Atributos da coleção TextoCandidato</b>
--

screen_name	O campo screen_name com o nome do candidato.
texto	texto a ser analisado pelo API Watson.

# Funcionalidades

Nessa seção serão descritos algumas funcionalidades relevantes sobre o sistema.

- **Cadastrar Tweet**

O cadastro dos tweets no Banco de Dados é realizado no momento da Coleta dos dados, onde os dados retornados pelas API's do Twitter são filtrados e armazenados nas Coleções do banco.

- **Total de Tweets**

Na aplicação é implementada uma função para contar o total de Tweets já armazenado por cada candidato. Para realizar essa contagem é usada primeiramente um recurso do MongoDB denominado aggregate. A agregação é uma operação que agrupa os dados dos documentos armazenados no banco MongoDB. Ele permite realizar várias operações nos dados agrupados para retornar um único resultado. Dessa forma foi possível agrupar dentro da coleção Candidatos somente os documentos referentes a um único candidato e realizar a contagem já armazenada de tweets dele.

- **Último Tweet**

O id do Tweet (fornecido pela API do Twitter) foi utilizado com o objetivo de evitar duplicidade de dados no MongoDB, ou seja, não permitir que Tweets já cadastrados fossem novamente inseridos no banco de dados.

Uma vez que o MongoDB não oferece uma função que retorne o id do último Tweet armazenado, utilizamos uma lógica de programação para obtê-lo. Ordenamos os registros no MongoDB de forma decrescente a fim de que o último registro se torne o primeiro. Dessa maneira conseguimos o id do último Tweet coletado.

O id é, então, utilizado no próximo processo de coleta dos Tweets. Serão coletados apenas os Tweets a partir do id fornecido. Isso quer dizer que o último Tweet cadastrado no MongoDB será novamente coletado. No entanto, antes da inserção deste Tweet no MongoDB, o mesmo será descartado para impedir a dualidade de registros.

- **Busca Palavra**

Função para realizar a busca de tweets baseado na palavra chave escolhida do dropdown da aplicação realizando a verificação do último tweet inserido com base no ID e buscando somente tweets a partir daquele evitando que seja gravado dados repetidos no banco de dados.

- **Tweets para Texto**

Cria um texto de todos os tweets do candidato selecionado e salva no banco de dados.

- **Quantidade Seguidores**

Pega a quantidade de seguidores que o candidato escolhido tem no momento atual e os retorna para a aplicação.

- **Analisar NLU / Analisar NLU Palavra Chave**

Analisa o tweet do candidato ou da palavra chave pedidos pelo usuário, a plataforma Watson analisa os sentimentos e retorna os dados atualizando a lista de tweets.

- **Traduzir Texto**

Traduz o texto do candidato escolhido para inglês usando a plataforma Watson.

- **Analisar Personalidade**

Envia um texto em inglês para a plataforma Watson o mesmo o analisa e retorna um json com as características da personalidade do candidato e salva numa nova coleção estes dados.

- **Pegar todos Tweets**

Retorna o dado de todos os tweets dentro de um limite inserido pelo usuário assim é possível ver a quantidade de tweets sem os repetir

- **Update Tweet**

Verifica o \_id do tweet e o atualiza no banco de dados com os novos dados.

- **Qt Sentimento**

Ou quantidade de sentimento é uma função para contar no banco de dados quais foram os sentimentos sentido pelo candidato e retorna a quantidade de vezes nos tweets que ele foi identificado.

- **Qt Retweets**

Ou quantidade de retweets é uma função para somar a quantidade de retweets dos tweets cadastrados no banco de dados.

# Propostas de análises

Em concordância com os objetivos desse projeto, algumas propostas para análise e exibição final dos dados estão sendo estudadas, bem como já implementadas.

Como já descrito nesse documento, já se encontra em fase de elaboração a análise de sentimento dos dados coletados. Após os dados passarem pelo processamento do Watson o sistema irá apresentar a personalidade de cada candidato, de acordo com as características estabelecidas pela API Personality Insights. Já para análise de sentimento de cada tweet o retorno será exibido em uma tela do sistema e conterà o resultado das características de retorno da API Natural Language Understanding.

No que concerne aos dados armazenados sobre os candidatos e palavras-chave pretende-se exibir, possivelmente no formato de gráfico, as seguintes considerações:

- relação de cada candidato quanto ao uso do Twitter através de uma consulta que retorne a quantidade de tweets de cada um;
- candidato que obteve maior apreço em suas postagens por meio de uma consulta que informe qual candidato teve mais retuítes;
- dentro de uma amostra, exibir qual região brasileira mencionou com maior frequência determinada palavra-chave referente a determinado(s) candidato(s);
- o favoritismo dos candidatos baseado na contagem de seus seguidores;
- a(s) postagem(s) dos candidatos que tiveram maior repercussão no Twitter, levando em consideração quantas vezes foi retuitada.

# Resultados

Os resultados obtidos com a implementação deste projeto são bastante claras ao se analisar os gráficos disponibilizados na aplicação, dentre os presidenciáveis o candidato Jair Messias Bolsonaro apresenta o maior rendimento de tráfego orgânico na rede social seguido por Fernando Haddad, Manuela Davila e General Mourão, também pode se notar que durante o tempo eleitoral foram vistos postagens com mais sentimentos positivos ou neutros, mostrando uma mudança de padrão nas atitudes em suas postagens, usuários da rede social também tiveram seus dados analisados com sucesso vendo que em suma maioria os tweets estão ligados ao candidato Bolsonaro tanto quanto tweets com sentimentos negativos quanto sentimentos positivos, em suma maioria todos incluindo os candidatos estão falando de assuntos como política já previsto por ser o assunto central e também uma grande quantidade sobre finanças e racismo mostrando a preocupação e alerta dos usuários.

Analisando a personalidade de cada candidato é possível notar que para o candidato Mourão a análise foi classificada como fraca por existir apenas 114 tweets em sua conta, já dos demais foi possível analisar com maior exatidão, mostrando que devido a época eleitoral os candidatos se diferenciam nas características apenas alguns pontos, não passando de 10.

Todas as funções elaboradas dentro da primeira versão deste documento foram completadas com sucesso, algumas com maiores dificuldades do que outras pela aplicação está usando um sistema assíncrono e com o CRUD clusterizado para ter uma maior eficiência e ganhos extraordinários nas consultas.

# Conclusão

Durante a implementação do projeto proposto, diversas mudanças foram realizadas na estrutura do código fonte, quanto do banco de dados, os dados disponibilizados pela rede social são de fácil manipulação o grande desafio se tornou analisar o que seria útil e o que não seria para obter os resultados que agregaria maior valor. Devido Javascript ser uma linguagem assíncrona foram enfrentadas diversas barreiras para se inserir os dados e renderiza-los na tela, e também para realizar a pesquisa no banco de dados.

O uso da ferramenta de versionamento Git foi essencial para manter a ordem e a clareza no código, nem todas as funcionalidades pensadas que não estavam neste documento foram implementadas devido ao tempo para elaborar e executar o projeto.

Concluimos que em suma é possível usar de ferramentas bem simples e gratuitas para fazer uma análise de uma pessoa somente com o que ela posta no Twitter, durante o trabalho foi percebido várias modificações nas quantidades de tweets do candidato Haddad que tinha 5 mil tweets e depois das eleições foi para 3 mil analisando estes dados percebeu-se que vários tweets sobre sua campanha foram excluídos, também se nota como o twitter é usado pelos presidentiáveis para gerar uma relação mais próxima com seus eleitores. Podendo modificar o pensamento de grandes massas com apenas alguns caracteres.

Diante de tantas opções e meios de comunicação o twitter é onde se gera maior conteúdo para se minerar mas nem sempre este conteúdo é de qualidade, é preciso grande trabalho para conseguir analisar os dados para saber o que é ouro e o que é lixo.