Relatório Final

Matheus Carvalho Raimundo[1]

[1]Instituto De Ciências Matemáticas e de Computação – Universidade de São Paulo
São Carlos – SP – Brasil

mcarvalhor@usp.br

Abstract. *This article defines the project for the class of Cloud Computing and Service-Oriented Architecture, the technologies used and its architecture.*

Resumo. *Este artigo define o projeto da disciplina de Computação em Nuvem e Arquitetura Orientadas a Serviços, as tecnologias usadas e sua arquitetura.*

1. Projeto

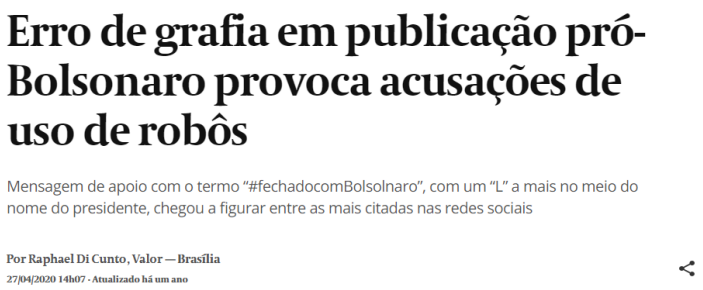
Nosso projeto consistirá de um sistema para analisar mensagens (chamados '*tweets*') publicadas na rede social Twitter.

É fornecido para o projeto uma *query* ou critério de busca. Tal critério de busca pode consistir de diversos fatores: uma frase, uma palavra, um tópico (chamado de *hashtag*, e sempre começa com '#' no Twitter) ou um usuário (sempre começa com '@' no Twitter). O sistema irá baixar todos os tweets que satisfazem tal busca usando a API do Twitter, e criar uma cópia local no banco de dados. Tal cópia consistirá apenas dos metadados - como conta que tweetou, geolocalização, data, entre outros. Após isso, é possível fazer análises destes metadados para determinar algum padrão entre eles.

Como exemplo específico, imagine que a *query* seja "#CPIdaCovid", um assunto muito comentado nas últimas semanas. O sistema irá primeiramente salvar todos os metadados de todos os tweets que satisfazem esta query. Após isso, nós vamos iniciar a análise para determinar, por exemplo, se a maioria dos usuários que tweetou sobre "#CPIdaCovid" são usuários que se cadastraram recentemente na rede social ou não, etc. Com isso podemos determinar se há um padrão entre todos estes tweets.

1.1 Motivação e Objetivo

A motivação deste projeto vem de uma reportagem publicada com a seguinte manchete:



Erro de grafia em publicação pró-Bolsonaro provoca acusações de uso de robôs

Mensagem de apoio com o termo "#fechadocomBolsolnaro", com um "L" a mais no meio do nome do presidente, chegou a figurar entre as mais citadas nas redes sociais

Por Raphael Di Cunto, Valor — Brasília

27/04/2020 14h07 · Atualizado há um ano

Figura 1. Manchete publicada no jornal "Valor Econômico - Globo".

O objetivo é descobrir se nosso sistema será capaz de reunir informações o suficiente para ser possível determinar se uma "hashtag" é autêntica (impulsionada por usuários autênticos do Twitter) ou não (impulsionada através do uso de robôs), tendo como fonte apenas os metadados dos tweets. Não saberemos se este objetivo será cumprido - pois isto só será possível saber ao final do semestre quando todo o projeto estiver pronto e funcional. Mas o sistema se enquadra nos requisitos do projeto da disciplina, pois será necessário um servidor de banco de dados, um programa que se comunique com a API do Twitter e vários programas rodando de maneira paralela que fazem o processamento dos dados. Além disso, será usado o Apache Kafka, para que os dados provenientes da API do Twitter sejam enfileirados e, posteriormente, processados.

2. Tecnologias Selecionadas

Nesta seção será detalhada as tecnologias que serão usadas no desenvolvimento do projeto.

2.1 Tweepy

A biblioteca do Python Tweepy será usada para se comunicar com a API do Twitter. Esta biblioteca é *open-source* e contém todas as chamadas que serão necessárias para o projeto. A biblioteca foi escolhida por ter uma extensa documentação e ser de simples implementação.

2.2 Apache Kafka

A plataforma *open-source* do Apache Kafka será usada para evitar atrasos por conta de congestionamentos do banco de dados. Durante a análise dos *tweets*, o produtor será o Gerenciador de Processamento, que enviará os *tweets* ao Kafka. Os consumidores serão os nós Processadores, que irão analisar os dados.

2.3 MySQL Server

O servidor de banco de dados MySQL Server será usado para armazenar todos os dados e as análises geradas por eles. O MySQL foi escolhido por ser um servidor de banco de dados relacional com potencial, eficiente e de rápida instalação e inicialização.

2.4 Flask

Será criada uma interface gráfica simples *web* para visualizar e trabalhar com os dados. O *framework* utilizado para esta interface *web* será o Flask, e o servidor *backend* utilizará do protocolo REST. O Flask será usado por ser um *framework* que simplifica a implementação do *backend* do servidor *web*, ao mesmo tempo em que tem alta escalabilidade.

2.5 Docker

Todos os serviços utilizados pela aplicação, e a própria aplicação em si, serão disponibilizados em contêineres do Docker, visando simplificação na implantação e configuração destes serviços.

3. Arquitetura do Projeto

Primeiramente, será requisitado uma busca por todos os tweets que satisfazem uma determinada *query*. O usuário acessa a interface *web* - que se comunica com o servidor *web* - e cadastra uma nova base de dados. O servidor *web* então vai criar a base no servidor de banco de dados e notificar o nosso programa responsável por se comunicar com a API do Twitter para que ele comece a coletar os *tweets* e salvar no banco de dados. Veja o diagrama de comunicação abaixo:

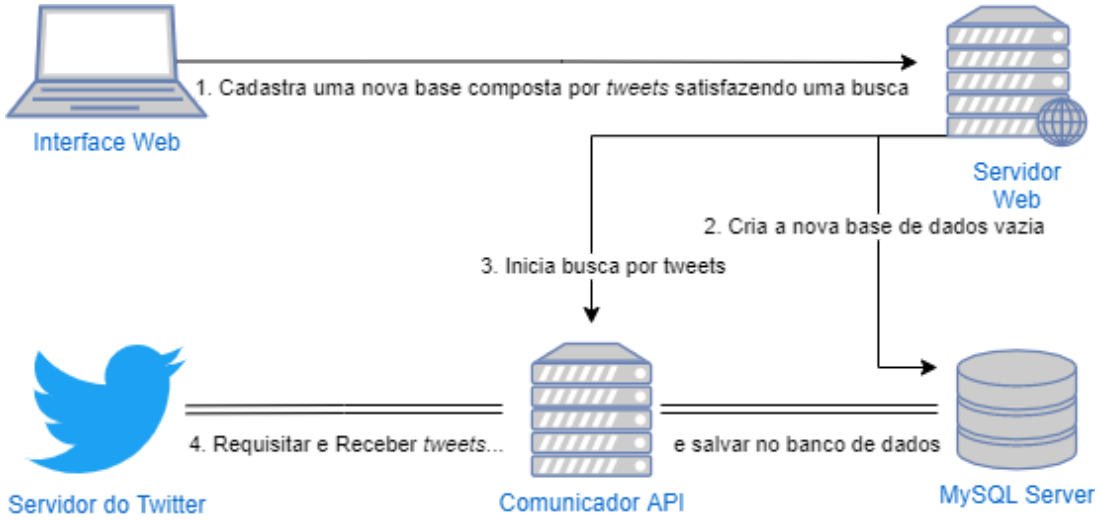


Figura 2. Diagrama de comunicação na criação de uma base de dados.

Os dados serão coletados até que o usuário deseje parar - ou seja, o Comunicador API vai continuar coletando *tweets* até o usuário ficar satisfeito, acessar a interface *web* e parar a coleta. Neste momento o Comunicador API vai parar de trabalhar.

Agora que temos uma base cheia de *tweets*, podemos iniciar o processamento deles. Neste caso, o usuário acessa a interface *web* e cadastra uma nova análise sob os dados coletados. O servidor *web* vai então criar a análise no banco de dados e notificar o Gerenciador de Processamento para que ele comece a distribuir o processamento dos dados entre cada um dos nós - chamados Processadores. Essa distribuição dos *tweets* entre os nós é dada pelo Apache Kafka. Veja o diagrama de comunicação abaixo:

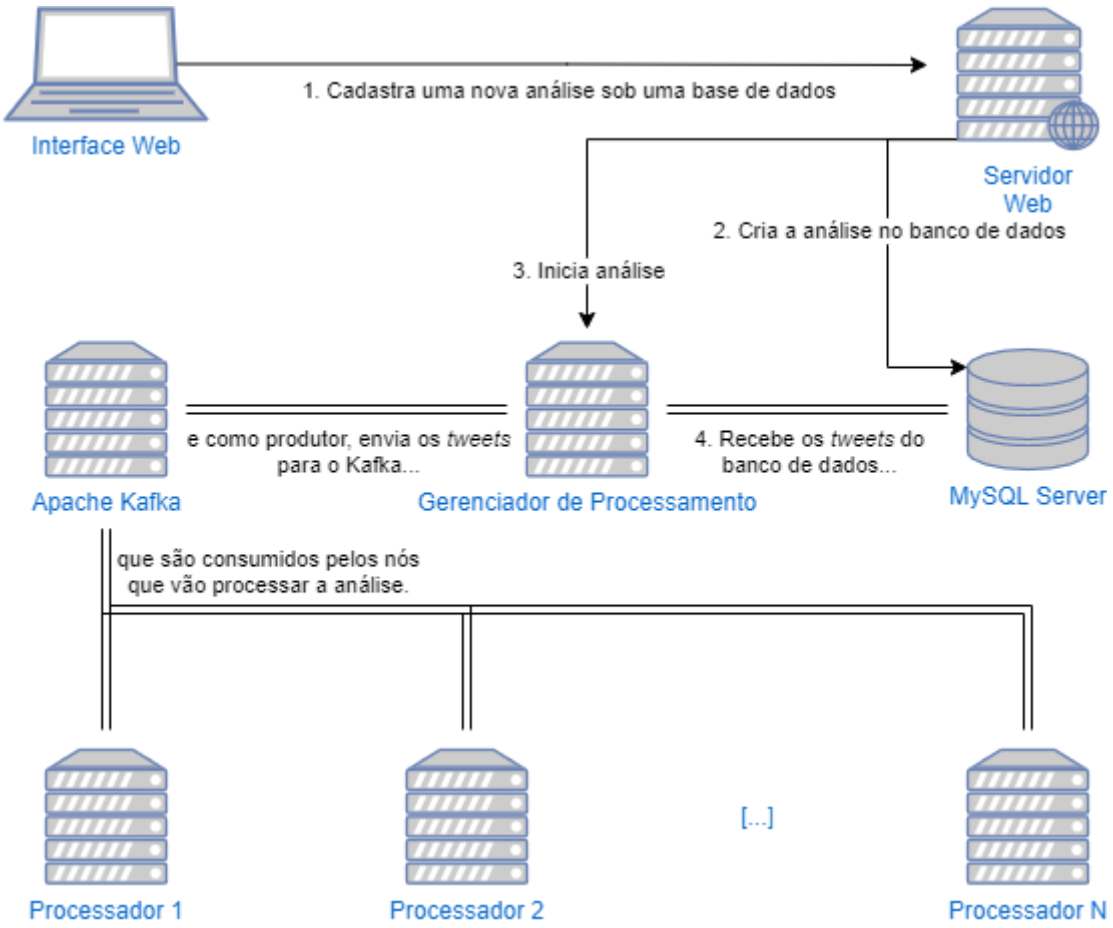


Figura 3. Diagrama de comunicação na análise dos *tweets*.

Quando a análise termina, o Gerenciador de Processamento apenas salva o resultado final no banco de dados MySQL.

4. Decisões de Projeto

Nesta seção serão abordados decisões de projeto, como detalhes de implementação, configuração e instalação (implantação) e execução em um cluster dedicado. O projeto será chamado de Twitter Analyser.

4.1 Implementação

Toda a implementação do projeto foi realizada utilizando a linguagem de programação Python. Todos os serviços oferecidos por esse código Python também foram conteinerizados, utilizando da tecnologia do Docker. Isso permite fácil implantação da aplicação em qualquer ambiente, seja ele distribuído ou não. Para simplificar ainda mais, também foi disponibilizado um *script bash* para lidar com a implantação, que considera a possibilidade de até 3 máquinas distintas - caso em que será abordado neste projeto nas seções seguintes.

O projeto se encontra hospedado no GitLab:
https://gitlab.com/icmc-ssc0158-2021/2021/gcloud09/

4.2 O Cluster

Com a finalidade de realizar testes e configurar a aplicação, foi disponibilizado um cluster da Universidade de São Paulo (ICMC/USP) para o projeto. Há à disposição 3 máquinas distintas (virtualizadas) que podem ser utilizadas para colocar a aplicação em funcionamento. A seguir, seguem as especificações destas máquinas e o propósito definido para cada uma delas:

Tabela 1. Máquinas do cluster dedicadas ao projeto.

Máquina	Acesso SSH	CPU	RAM	Disco	Propósito definido
and09-vm1	ssh gcloud09@andromeda.lasdpc.icmc.usp.br -p 2191	2	2GB	20GB	Aplicação
and09-vm2	ssh gcloud09@andromeda.lasdpc.icmc.usp.br -p 2192	2	2GB	20GB	Apache Kafka
and09-vm3	ssh gcloud09@andromeda.lasdpc.icmc.usp.br -p 2193	2	2GB	20GB	MySQL

Exatamente por esse motivo é que foi criado o *script bash* para implantação da aplicação: ele foi criado para facilitar a implantação neste ambiente em específico.

4.3 Implantação

Note que esta seção do relatório já foi realizada. Se você deseja testar a aplicação, cuja qual já se encontra implantada, pule para a seção “Execução” abaixo.

Para implantar a aplicação, primeiramente é necessário instalar o Docker em todas as 3 máquinas. Para isso, basta executar o seguinte comando em cada uma delas:

```
sudo apt-get install docker.io docker-compose
```

Agora, precisamos clonar o repositório do Git em que o projeto se encontra hospedado. Isso é feito em apenas uma das máquinas, e através do comando abaixo - substituindo “GIT_URL” pelo endereço apropriado do repositório *git* (https://gitlab.com/icmc-ssc0158-2021/2021/gcloud09.git):

```
git clone "GIT_URL"
```

Agora, basta executar o *script bash* “cluster-install.sh”:

```
bash cluster-install.sh
```

O *script* consegue sozinho e automaticamente lidar com a implantação de todos os serviços, mas é necessário seguir à risca todas as instruções mostradas na tela. Por exemplo, o *script* vai informar logo no início que precisará acessar todos os hosts como usuário “root” (e isso é algo que deve ser alterado manualmente nas configurações do servidor SSH, pois o *script* não consegue fazer isso sozinho).

4.4 Execução

Quando a aplicação é instalada utilizando o *script bash*, ela automaticamente já se torna online e pode ser acessada utilizando um navegador web através do endereço externo do cluster e a porta externa aberta no Firewall interno da USP (http://andromeda.lasdpc.icmc.usp.br:9180/). Se, porventura, a aplicação cair ou uma das máquinas em questão forem reiniciadas, é necessário iniciar a aplicação manualmente.

Para **iniciar** os serviços e a aplicação manualmente, basta acessar cada uma das máquinas (na ordem de MySQL, depois Apache Kafka e por último a Aplicação) e executar o seguinte comando:

```
docker stop $(docker ps -q); docker start TA_app TA_mysql TA_zookeeper; sleep 10; docker start TA_kafka
```

E caso seja necessário **parar** os serviços e a aplicação, basta acessar cada uma das máquinas (em qualquer ordem) e executar o seguinte comando:

```
docker stop $(docker ps -q); docker start TA_offline
```

5. Experimento

Será realizado um experimento utilizando a aplicação. Para isso, primeiramente será selecionado um tópico de interesse no Twitter. Acessando a página “Trending” na rede social (Explore / Twitter), temos alguns tópicos em destaque para o dia 20 de julho de 2021:

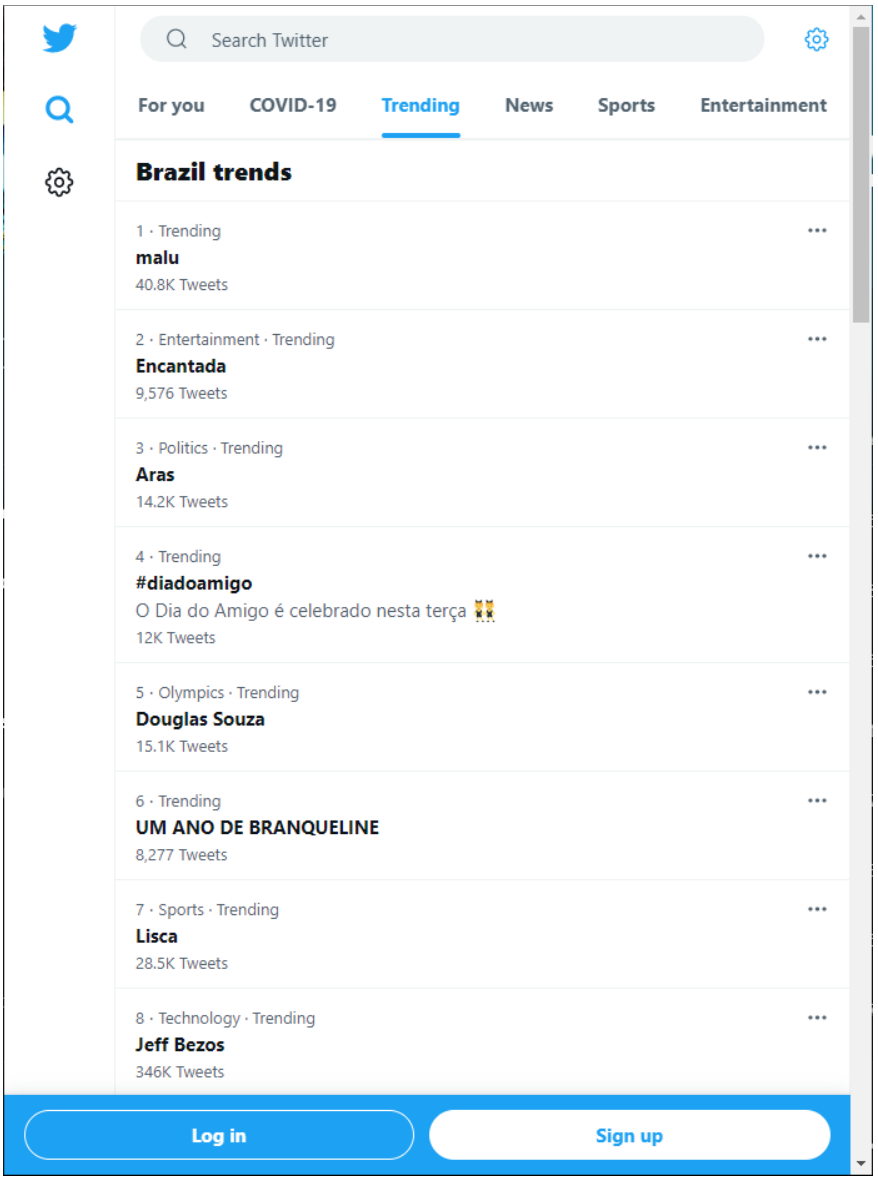


Figura 4. Tópicos em destaque no Twitter no dia 20 de julho de 2021.

Para este experimento, vamos utilizar o tópico “#diadoamigo”.

Acesse em seu navegador o endereço da aplicação (http://andromeda.lasdpc.icmc.usp.br:9180/):

Figura 5. Página inicial do Twitter Analyser.

Agora clique em “Bases de Dados” no menu superior, e depois clique em “Criar Base de Dados”. Insira o nome “Tweets sobre o Dia do Amigo”, o critério de busca "#diadoamigo”, o idioma “Português” e a data mínima “20 de julho de 2021, às 9:00AM”.

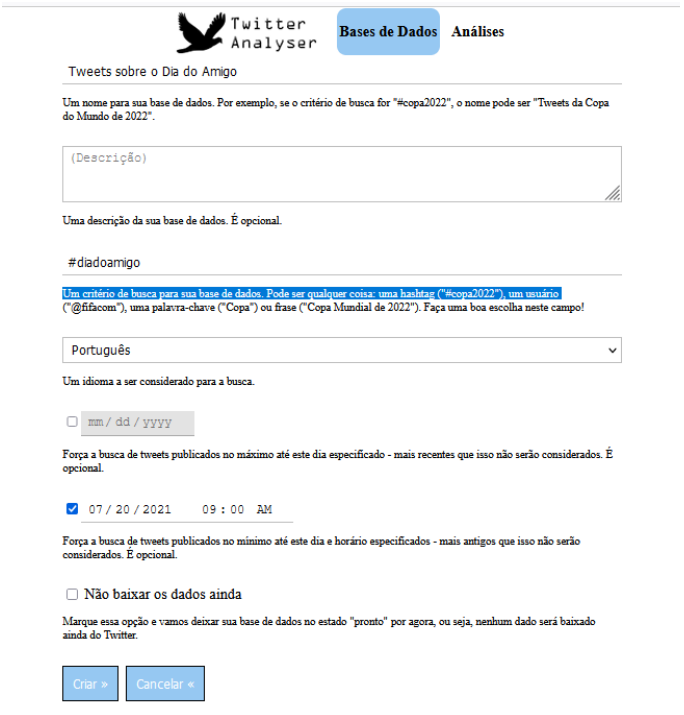


Figura 6. Página de criação de uma nova Base de Dados.

Clique em “Criar” e aguarde a busca pelos tweets. Quando a busca se encerrar, a mensagem “Busca completa” será exibida:

Figura 7. Página de Bases de Dados.

Agora clique em “Análises” e clique em “Criar Análise”. Insira o nome “Análise sobre o Dia do Amigo” e selecione “Tweets sobre o Dia do Amigo” como Base de Dados. Clique em “Criar” e aguarde a análise ser processada.

Figura 8. Página de Análises (mostrando nossa Análise em processamento).

Quando a análise terminar de ser processada, os resultados são exibidos:

Figura 9. Página de Análises (mostrando nossa Análise completa).

6. Resultados

Para o experimento realizado na seção anterior, obtém-se os seguintes resultados:

Tabela 2. Resultados obtidos no experimento.

Número de tweets	8897
Número de tweets com poucas interações	3128 (35.16%)
Número de contas	7754
Número de contas recentes	1802 (23.24%)
Número de contas com perfil padrão	5072 (65.41%)
Número de contas com foto de perfil padrão	59 (0.76%)
Número de contas com poucos 'seguindo'	1484 (19.14%)
Número de contas com poucos seguidores	1059 (13.66%)
Número de contas com poucos tweets publicados	351 (4.53%)
Número de contas com poucos tweets favoritados	655 (8.45%)

Referências Bibliográficas

Universidade de São Paulo (2021). “Aulas de SSC0158 Computação em Nuvem e Arquitetura Orientadas a Serviços”, maio.

Raphael Di Cunto (2020) “Erro de grafia em publicação pró-Bolsonaro provoca acusações de uso de robôs”, https://valor.globo.com/politica/noticia/2020/04/27/erro-de-grafia-em-publicacao-pro-bolsonaro-levanta-acusacoes-de-uso-de-robos.ghtml, maio.

Twitter API v1.1 (2021) “Standard search API”, https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets, maio.