

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
INSTITUTO DE COMPUTAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GABRIEL DE SAPIENZA LUNA  
FELIPE TOMAZELLI CRESPO  
RODRIGO PAIVA DAMASCENO

Framework baseado em big data e machine learning para identificação de ideação suicida  
de usuários do Twitter.

RIO DE JANEIRO  
2022

GABRIEL DE SAPIENZA LUNA  
FELIPE TOMAZELLI CRESPO  
RODRIGO PAIVA DAMASCENO

Framework baseado em big data e machine learning para identificação de ideação suicida de usuários do Twitter.

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. D.Sc. Jonice de Oliveira Sampaio

RIO DE JANEIRO

2022

L961f

Luna, Gabriel de Sapienza

Framework baseado em big data e machine learning para identificação de ideação suicida de usuários do Twitter / Gabriel de Sapienza Luna, Felipe Tomazelli Crespo e Rodrigo Paiva Damasceno. – Rio de Janeiro, 2022.

69 f.

Orientadora: Jonice de Oliveira Sampaio.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal do Rio de Janeiro, Instituto de Computação, Bacharel em Ciência da Computação, 2022.

1. Redes sociais. 2. Twitter. 3. Big Data. 4. Análise de sentimento. 5. Framework. 6. Inteligência artificial. 7. Twitter. I. Crespo, Felipe Tomazelli. II. Damasceno, Rodrigo Paiva. III. Sampaio, Jonice de Oliveira (Orient.). IV. Universidade Federal do Rio de Janeiro, Instituto de Computação. V. Título.


GABRIEL DE SAPIENZA LUNA  
FELIPE TOMAZELLI CRESPO  
RODRIGO PAIVA DAMASCENO

Framework baseado em big data e machine learning para identificação de ideação suicida de usuários do Twitter.

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.


Aprovado em 14 de Março de 2022

BANCA EXAMINADORA:

Documento assinado digitalmente  
 JONICE DE OLIVEIRA SAMPAIO  
Data: 16/03/2022 17:28:43-0300  
Verifique em <https://verificador.iti.br>


---

Jonice de Oliveira Sampaio  
D.Sc. (PPGI/UFRJ)

Documento assinado digitalmente  
 Monica Ferreira da Silva  
Data: 16/03/2022 18:00:45-0300  
Verifique em <https://verificador.iti.br>


---

Mônica Ferreira da Silva  
D.Sc. (PPGI/UFRJ)

Documento assinado digitalmente  
 JULIANA BAPTISTA DOS SANTOS FRANCA  
Data: 18/03/2022 10:08:24-0300  
Verifique em <https://verificador.iti.br>

---

Juliana Baptista dos Santos França  
D.Sc. (IC/UFRJ)

Documento assinado digitalmente  
 SILAS PEREIRA LIMA FILHO  
Data: 18/03/2022 10:19:08-0300  
Verifique em <https://verificador.iti.br>

---

Silas Pereira Lima Filho  
M.Sc. (PPGI/UFRJ)

Dedicamos este trabalho a Deus, que nos presenteia todos os dias com a energia da vida, que nos dá forças e coragem para atingir os nossos objetivos

## AGRADECIMENTOS

### AGRADECIMENTOS GABRIEL DE SAPIENZA LUNA

Agradeço a minha namorada Carina que me inspirou, apoiou em todos os momentos. Foi ela quem me incentivou acerca da temática do suicídio, que é tão atual e relevante para a sociedade.

Agradeço a todos os meus familiares por me incentivarem desde o início na minha jornada acadêmica. Meus pais, Carlos e Sheila, e meus irmãos Rodrigo e Rafael.

Aos professores Jonice Oliveira, Mônica Ferreira, Juliana Baptista e Silas Pereira por participarem da banca de avaliação.

Aos professores e técnicos da Universidade Federal do Rio de Janeiro, em especial do Instituto de Computação, por todo conhecimento compartilhado e que contribuíram imensamente ao longo da minha formação.

## **AGRADECIMENTOS**

### **AGRADECIMENTOS FELIPE TOMAZELLI CRESPO**

Agradeço a minha família por me incentivar e garantir que eu pudesse chegar onde cheguei. A professora Jonice por acreditar no potencial do nosso trabalho e nos dar a oportunidade de o efetivar e apresentar.

A UFRJ pelos anos de ensino e orientação, essenciais para que possamos realizar este e qualquer trabalho futuro.

E a todos que direta ou indiretamente fizeram parte de minha formação, o meu muito obrigado.

## **AGRADECIMENTOS**

### AGRADECIMENTOS RODRIGO PAIVA DAMASCENO

Agradeço a minha família por me apoiar até aqui.

E a toda UFRJ pelos anos de ensino e por proporcionar uma ótima orientação ao longo da minha jornada.



*“Either write something worth reading or do something worth writing.”*

**Benjamin Franklin**

## RESUMO

O pensamento suicida é um dos problemas de saúde mental mais graves enfrentados pela sociedade. São diversos fatores de riscos envolvidos que podem levar ao acometimento do suicídio. Alguns destes são: depressão, isolamento social e a ansiedade. A utilização de ferramentas para realizar a detecção precoce desses fatores auxiliam na prevenção ou redução desses números de suicídios. Com o advento das redes sociais, os usuários começaram a expressar seus pensamentos abertamente para outros indivíduos frequentadores dessas páginas. Este trabalho apresenta uma proposta de framework utilizando o Twitter como base, para analisar melhor a ideação suicida, auxiliando na identificação de pessoas em risco potencial de suicídio. Os dados são coletados em tempo real por meio da API do Twitter, tweepy. Com o auxílio de diversas ferramentas conhecidas no mercado de Big Data, pode-se realizar uma análise dos tweets em tempo real. O objetivo final da framework será distinguir os tweets, de forma automática, entre a ideação suicida ou não.

**Palavras-chave:** redes sociais; twitter; big data; analise de sentimento; framework; inteligência artificial; suicídio.

## ABSTRACT

Suicidal thinking is one of the most serious mental health problems faced by society. There are several risk factors involved that can lead to suicide. Some of these are: depression, social isolation and anxiety. The use of tools to perform early detection of these factors help to prevent or reduce these numbers of suicides. With the advent of social networks, users began to express their thoughts openly to other individuals who frequent these pages. This work presents a framework proposal using Twitter as a basis, to better analyze suicidal ideation, helping to identify people at potential risk of suicide. Data is collected in real-time via Twitter's tweepy API. With the help of several tools known in the Big Data market, it is possible to perform an analysis of tweets in real time. The final objective of the framework will be to distinguish the tweets, automatically, between suicidal ideation or not.

**Keywords:** social networks; twitter; big data; sentiment analysis; framework; artificial intelligence; suicide.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Taxa de suicídio por 100 mil habitantes . . . . .	19
Figura 2 – Taxa de suicídio por faixa etária entre 1985 até 2015 . . . . .	19
Figura 3 – Representação gráfica dos 3’vs de Big Data . . . . .	24
Figura 4 – Representação gráfica dos 7’vs de Big Data adaptada de (NICULESCU, 2020) . . . . .	25
Figura 5 – Arquitetura Produtor e Consumidor de um Tópico no Kafka (NANNONI, 2015) . . . . .	28
Figura 6 – Arquitetura geral com Zookeeper no Kafka (NANNONI, 2015) . . . . .	28
Figura 7 – Arquitetura Cluster Spark adaptado de (ZAHARIA et al., 2012) . . . . .	30
Figura 8 – (ZAHARIA et al., 2012) . . . . .	31
Figura 9 – Exemplificação de formato colunar . . . . .	32
Figura 10 – Armazenamento de dados em formato de linha . . . . .	32
Figura 11 – Armazenamento de dados em formato colunar . . . . .	33
Figura 12 – Comparativo entre arquivos no formato CSV e no formato Parquet (DATABRICKS, 2021) . . . . .	33
Figura 13 – Gráfico da função sigmoid . . . . .	37
Figura 14 – Matriz de Confusão . . . . .	40
Figura 15 – Resumo do funcionamento da arquitetura do Docker (DOCKER, 2022a) . . . . .	44
Figura 16 – Arquitetura do Sistema . . . . .	46
Figura 17 – Arquitetura do Sistema . . . . .	46
Figura 18 – ADAPTADA DE (NANNONI, 2015) . . . . .	48
Figura 19 – Pipeline de Machine Learning . . . . .	49
Figura 20 – Exemplificação de uma parte do Data set de treino. . . . .	50
Figura 21 – ADAPTADA DE (NANNONI, 2015) . . . . .	52
Figura 22 – Data set . . . . .	53
Figura 23 – Criação do tópico Analise-de-Twitter . . . . .	54
Figura 24 – Produtor de dados . . . . .	54
Figura 25 – Produtor de dados . . . . .	55
Figura 26 – Consumidor de dados . . . . .	55
Figura 27 – Script python que retorna os tweets com as palavras mencionadas no capítulo 3.1.2. . . . .	56
Figura 28 – Script python que retorna os tweets com as palavras mencionadas no capítulo 3.1.2. . . . .	56
Figura 29 – Consumidor do tópico Kafka Analise-de-Twitter . . . . .	57
Figura 30 – Spark Streaming classificando Tweets : Lote 38 . . . . .	57
Figura 31 – Spark Streaming classificando Tweets : Lote 42 . . . . .	58

Figura 32 – Spark Streaming classificando Tweets: Lote 55 . . . . .	58
Figura 33 – Spark Streaming classificando Tweets : Lote 305 . . . . .	58
Figura 34 – Spark Streaming classificando Tweets : Lote 470 . . . . .	58
Figura 35 – Spark Streaming classificando Tweets : Lote 550 . . . . .	59
Figura 36 – Spark Streaming classificando Tweets : Lote 666 . . . . .	59
Figura 37 – Spark Streaming classificando Tweets : Lote 924 . . . . .	59
Figura 38 – Spark Streaming classificando Tweets : Lote 1637 . . . . .	59
Figura 39 – Spark Streaming classificando Tweets : Lote 1641 . . . . .	60
Figura 40 – Dados armazenados na extensão parquet . . . . .	60
Figura 41 – foto ilustrativa do site <a href="http://www.twitterscan.com.br">www.twitterscan.com.br</a> . . . . .	61
Figura 42 – foto ilustrativa do site <a href="http://www.twitterscan.com.br">www.twitterscan.com.br</a> . . . . .	62
Figura 43 – Arquitetura do ambiente virtualizado na nuvem AWS . . . . .	63

## LISTA DE TABELAS

Tabela 1 – Definições de Big Data. . . . .	23
Tabela 2 – Fase de Tokenizacão. . . . .	51
Tabela 3 – Fase de StopWordRemover. . . . .	51
Tabela 4 – Fase de Hashing. . . . .	51
Tabela 5 – Fase de Regressão. . . . .	52
Tabela 6 – Separação de dados de treinamento e dados de teste. . . . .	54

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
I <sup>2</sup> C	Inter-Integrated Circuit
SRAM	Static Random-Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
LED	Light-Emitting Diode
MLP	Modulação por Largura de Pulso
PWM	Pulse-Width Modulation
PID	Proportional–Integral–Derivative
RAM	Random-Access Memory
API	Application Programming Interface
GPL	GNU General Public License
GNU	GNU's Not Unix
iid	Independente e identicamente distribuídas
OMS	Organização Mundial da Saúde
OPAS	Organização Pan-Americana da Saúde
EUA	Estados Unidos da América
IASP	International association for Suicide Prevention
I/O	Input/Output
RDD	Resilient Distributed Dataset
SQL	Structured Query Language
HDFS	Hadoop Distributed File System
IA	Inteligência Artificial
AWS	Amazon Web Services
CSV	Comma-separated values

JSON      JavaScript Object Notation

IOS      iPhone Operating System



## LISTA DE SÍMBOLOS

$\Gamma$	Letra grega Gama
$\Lambda$	Lambda
$\zeta$	Letra grega minúscula zeta
$\in$	Pertence
$\$$	subcampo

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>18</b>
1.1	JUSTIFICATIVA . . . . .	18
1.2	MOTIVAÇÃO . . . . .	20
1.3	OBJETIVOS . . . . .	21
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA . . . . .</b>	<b>22</b>
2.1	BIG DATA . . . . .	22
2.2	APACHE KAFKA . . . . .	26
<b>2.2.1</b>	<b>Arquitetura . . . . .</b>	<b>27</b>
<b>2.2.2</b>	<b>Aplicabilidade . . . . .</b>	<b>28</b>
2.3	APACHE SPARK . . . . .	29
<b>2.3.1</b>	<b>Arquitetura . . . . .</b>	<b>29</b>
<b>2.3.2</b>	<b>Tolerância a falhas . . . . .</b>	<b>30</b>
<b>2.3.3</b>	<b>Módulos Spark . . . . .</b>	<b>30</b>
2.4	APACHE PARQUET . . . . .	32
<b>2.4.1</b>	<b>Formato Colunar vs Formato de linha . . . . .</b>	<b>32</b>
<b>2.4.2</b>	<b>Eficiência . . . . .</b>	<b>33</b>
<b>2.4.3</b>	<b>Vantagens em armazenar os dados em formato colunar . . . . .</b>	<b>34</b>
2.5	MACHINE LEARNING . . . . .	34
<b>2.5.1</b>	<b>Aprendizado Supervisionado . . . . .</b>	<b>36</b>
<b>2.5.2</b>	<b>Regressão Logística . . . . .</b>	<b>36</b>
<b>2.5.3</b>	<b>Overfitting . . . . .</b>	<b>39</b>
<b>2.5.4</b>	<b>Metricas de desempenho . . . . .</b>	<b>39</b>
<b>2.5.5</b>	<b>Matriz de Confusão . . . . .</b>	<b>39</b>
<b>2.5.6</b>	<b>Análise Exploratória . . . . .</b>	<b>40</b>
2.6	COMPUTAÇÃO EM NUVEM . . . . .	41
2.7	DOCKER . . . . .	41
<b>2.7.1</b>	<b>Virtualização . . . . .</b>	<b>42</b>
<b>2.7.2</b>	<b>Contêineres . . . . .</b>	<b>42</b>
<b>2.7.3</b>	<b>Motores de contêiner . . . . .</b>	<b>42</b>
<b>2.7.4</b>	<b>Arquitetura Docker . . . . .</b>	<b>43</b>
2.7.4.1	Docker objects . . . . .	43
2.7.4.2	Docker Daemon . . . . .	43
2.7.4.3	Docker Client . . . . .	43
2.7.4.4	Docker Registries . . . . .	43

2.7.4.5	Docker Host . . . . .	43
<b>3</b>	<b>ARQUITETURA DO SISTEMA . . . . .</b>	<b>45</b>
3.1	INTRODUÇÃO . . . . .	45
3.1.1	Desenho da Arquitetura Lógica do sistema . . . . .	46
3.1.2	Coleta de dados . . . . .	47
3.1.3	Envio dos dados ao tópico Kafka . . . . .	48
3.1.4	Pipeline de Machine Learning Spark . . . . .	48
3.1.4.1	Exemplificação do estágios do pipeline de Machine Learning . . . . .	50
3.1.5	Consumo do tópico Kafka . . . . .	52
3.1.6	Armazenamento de dados em arquivos Parquet . . . . .	52
<b>4</b>	<b>PROVA DE CONCEITO DO FRAMEWORK PROPOSTO . . . . .</b>	<b>53</b>
4.1	EXECUÇÃO DA FRAMEWORK . . . . .	53
4.2	VISUALIZAÇÃO DOS DADOS . . . . .	61
4.3	EXECUÇÃO DA FERRAMENTA VIRTUALIZADA NO SERVIÇO DE NUVEM DA AWS . . . . .	62
4.3.1	Arquitetura da framework em nuvem . . . . .	63
<b>5</b>	<b>LIMITAÇÕES ENCONTRADAS . . . . .</b>	<b>64</b>
<b>6</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS . . . . .</b>	<b>65</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>67</b>

# 1 INTRODUÇÃO

Esse trabalho de conclusão de curso, tem como objetivo utilizar ferramentas de Big Data e Machine Learning para criar um framework com o intuito de trazer benefícios para a sociedade. Os alunos envolvidos, sempre tiveram um anseio por devolver à sociedade todo o investimento feito neles por utilizarem recursos públicos na sua formação profissional. Dessa forma, tiveram a ideia de desenvolver um framework, que tem como finalidade ajudar a reduzir a taxa de suicídio no Brasil, principalmente entre os jovens ativos nas redes sociais. Trabalhamos com o intuito de utilizar as melhores ferramentas existentes no mercado, e algoritmos de Machine Learning para desenvolver uma ferramenta que seja capaz de analisar redes sociais como Twitter em busca de possíveis pessoas com intenção suicida. Neste capítulo falaremos sobre:

- A justificativa, onde falaremos sobre o ambiente em que o nosso framework trabalhará e o espaço que tem para seu desenvolvimento.
- A motivação, onde falaremos sobre o que nos fez visualizar essa ferramenta, o que nos permitiu desenvolvê-la..
- E objetivos do trabalho, onde detalharemos sobre o que esperamos do resultado final do nosso trabalho.

## 1.1 JUSTIFICATIVA

Segundo a Organização Mundial Da Saúde, em torno de 800 mil pessoas no mundo ceifam suas próprias vidas por ano, e um número ainda maior tentam o suicídio. Em sua maioria, a concentração dessas pessoas é predominante em países de baixa e média renda. O suicídio é, atualmente, a segunda principal causa global de mortes entre os jovens de 15 a 29 anos (OPAS/OMS, 2021). Com isso, ele é visto como um problema de saúde pública, no entanto esse tipo de morte possui grandes chances de serem evitadas se identificados os sinais de risco em tempo oportuno. A utilização de bases em análise psicológica e comportamental do indivíduo é uma alternativa aplicável e facilitadora dessa identificação, tendo em vista que os jovens dessa faixa etária são usuários da internet e se expressam por meio dela. Dessa maneira, as redes sociais podem auxiliar na busca por sinais de risco desses usuários por meio de suas interações com o meio digital. É difícil achar um conjunto de dados públicos de qualidade e atualizados que traga informações precisas sobre o assunto. Mas após diversas pesquisas encontramos um dataset, disponível em (RUSTY, 2018), que é uma das maiores plataformas de competição de Data Science do mundo, o dataset possui informações a respeito do suicídio em diversos países a partir do ano de 1985 até o ano de 2016. Dessa forma, conseguimos analisar as ocorrências de

suicídio no mundo e compará-las com os dados do Brasil.

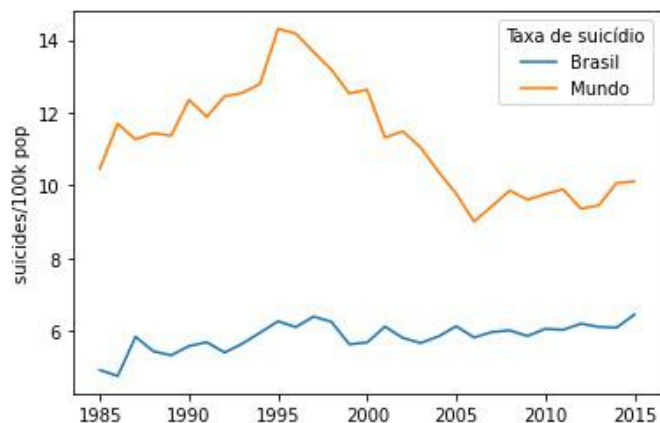


Figura 1 – Taxa de suicídio por 100 mil habitantes

A figura 1 traz uma comparação em números de suicídio por 100 mil habitantes entre a tendência mundial e no Brasil. A finalidade de se analisar a taxa por 100 mil habitantes tem como objetivo principal realizar a comparação entre os locais com diferentes tamanhos de população, permitindo então, uma comparação a médio e longo prazo. Abaixo temos a figura que contém a quantidade de mortes por suicídio no Brasil ao longo dos anos, dividida em cinco faixas etárias: dos 15 aos 24 anos, 25 aos 34 anos, 35 aos 54 anos, 55 aos 75 anos, e 75 anos ou mais. É notório que o Brasil se mantém abaixo da média mundial, mas há uma tendência de aumento ao longo dos anos. E com isso foi feita uma análise mais aprofundada para identificar as faixas etárias que mais cometem o suicídio no Brasil.

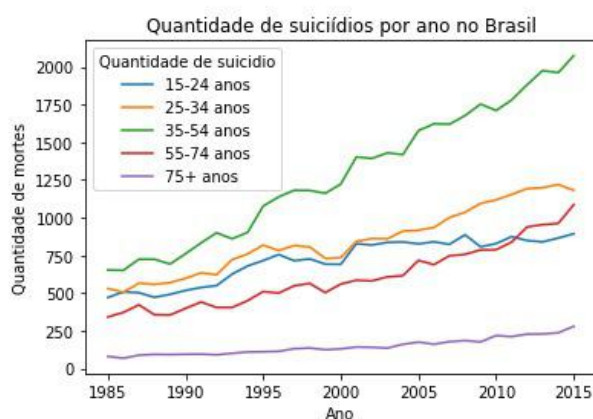


Figura 2 – Taxa de suicídio por faixa etária entre 1985 até 2015

Existe então, uma percepção de que em todas as faixas etárias há um crescimento na quantidade de mortes anual. Contudo os maiores valores de mortes se concentram

nas faixas etárias mais jovens, com uma predominância maior na faixa etária dos 35 aos 54 anos. Com esses dados a pesquisa foi norteada com o objetivo de tentar entender o suicídio dentro da rede global de computadores.

## 1.2 MOTIVAÇÃO

A internet teve seu início em 1969 com o nome ARPANET (OLIVEIRA, 2003), que tinha um simples objetivo central, fazer a conexão dos laboratórios nos EUA, facilitando o envio de e-mails entre essas instituições. Contudo, seu uso comercial se deu a partir da década de 90, com o surgimento de provedores de internet. Essas empresas trouxeram a proposta de facilitar a comunicação com outras pessoas através de compartilhamento e envio de vídeos, fotos e mensagens por e-mail. No Brasil, a internet começou a ser comercializada em 1996, com a liberação do serviço em todo o país. Com essa expansão acelerada, surgiram diversas redes sociais famosas como o Orkut, Facebook, Instagram e dentre outras gamas de redes de comunicação social. Com isso, pessoas começaram a expressar seus sentimentos no meio digital. Entretanto, essa maneira mais rápida e fácil de conectar as pessoas, tornou-se um dos colaboradores para o aumento do suicídio em massa. Segundo o (FAHEY; MATSUBAYASHI; UEDA, 2018) esse efeito se denomina *Werther* que se refere a um aumento drástico na quantidade de suicídios em uma determinada região após a divulgação de algum suicídio anterior. Podemos dizer então, que a utilização das redes sociais contribui significativamente para o aumento dos casos. Já que as pessoas se conectam umas com as outras expressando seus sentimentos. Porém existe um contraponto conhecido como o efeito *Papageno* que segundo (BLATT, 2019) refere-se à quando uma pessoa supera sua vontade de suicidar e essa superação é amplamente divulgada, servindo então de inspiração para outras pessoas desistirem de algum ato que coloque a sua vida em risco. Com isso, diversas organizações mundiais de saúde colocaram em foco a prevenção do suicídio no mundo, como por exemplo a Organização Mundial da Saúde (OMS) criou um órgão específico para tratar sobre o assunto, a Associação Internacional para a Prevenção do Suicídio (IASP) que tem o objetivo de promover campanhas em diversos países por meio de seus setores de saúde.

Das distintas redes sociais existentes, o Twitter se singulariza por ser uma plataforma onde os usuários costumam divulgar suas opiniões e pensamentos em mensagens curtas, o que facilita a análise computacional, pois quanto mais concisa a mensagem, menos interpretações podemos tirar dela, além desse cunho comunicativo pessoal, onde as mensagens quase sempre podem ser associadas a uma característica ou estado da psique do usuário. Diferentemente de outras plataformas como Instagram e Facebook, onde os usuários também se comunicam visualmente através de fotos e imagens, que necessitam uma análise mais profunda e complexa para encontrarmos padrões que possam ser avaliados e usados como base para análises futuras. O Software desenvolvido pode facilmente ser adaptado para analisar mensagens de outras redes sociais, mas escolhemos o Twitter para o desenvolvimento inicial pelos motivos citados.

Portanto, esse trabalho de conclusão de curso, se justifica através de utilizar ferramentas de computação para auxiliar na prevenção do suicídio. Por meio da identificação dos casos e regiões, auxiliando então o profissional da saúde e seus gestores a tomar decisões e medidas efetivas para uma prevenção mais robusta e eficaz.

### 1.3 OBJETIVOS

O objetivo final esperado desse trabalho é proporcionar um framework escalável e distribuído, com a utilização de diversas ferramentas no mercado de Big Data e algoritmos de Aprendizado de Máquina com o intuito de extrair , analisar e fazer uma classificação em relação à ideia suicida contido no texto de redes sociais.

## 2 REVISÃO BIBLIOGRÁFICA

Neste capítulo falaremos sobre conceitos e ferramentas utilizadas para o desenvolvimento do software. Os conceitos principais que escolhemos falar foram Big Data, Machine Learning e Computação em nuvem, pois consideramos como fundamental o entendimento desses para que se possa entender como funciona a ferramenta desenvolvida. Já as ferramentas que utilizamos e iremos detalhar neste capítulo são Apache Kafka, Apache Spark, Apache Parquet e Docker, tecnologias de ponta que são amplamente utilizadas e conceituadas no mercado.

Sobre a ordem das seções, temos Big Data como um conceito base utilizado pelas ferramentas Apache Kafka, Apache Spark e Apache Parquet, descritas nas seções subsequentes. Ao final destas ferramentas falaremos sobre Machine Learning, que é utilizado para extrair informações essenciais advindas do conceito de Big Data. Em seguida falamos sobre Computação em Nuvem, um conceito ligado a forma como utilizamos o Docker, ferramenta que é descrita em seguida e utilizamos para trabalhar a virtualização de recursos e aproveitar ao máximo as ferramentas descritas anteriormente.

### 2.1 BIG DATA

O termo "dados" pode ser empregado para definir diversas coisas, desde documentos criados, perfis em redes sociais, até mesmo simples cliques no navegador.

Ao projetar a escala destes dados para o âmbito global, obtemos um conceito chamado Big Data, porém não existe uma definição singular sobre essa terminologia, fazendo com que existam várias acepções traçadas por estudos e publicações, a conceituação se trata de uma parte importante do processo de compreensão, e para defini-lo melhor, a seguir é apresentada uma tabela com diversas abordagens de diferentes autores e seus respectivos conceitos:



Autores	Definição
(ORACLE, 2021)	Big Data é um conjunto de dados maior e mais complexo, especialmente de novas fontes de dados. Esses conjuntos de dados são tão volumosos que o software tradicional de processamento de dados simplesmente não consegue gerenciá-los. No entanto, esses grandes volumes de dados podem ser usados para resolver problemas de negócios que não era possível resolver antes.
(GARTNER, 2021) (tradução livre)	É um ativo de informação de alto volume, alta velocidade e/ou de alta variedade que exigem formas inovadoras e que tenham um ótimo custo-benefício para processamento de informações que permitem discernimentos aprimorados, tomada de decisões e automação de processos.
(OXFORD, 2021) (tradução livre)	Conjuntos de informações que são muito grandes ou muito complexas para lidar, analisar ou usar com métodos padrões.
(MCKINSEY, 2011) (tradução livre)	Refere-se a conjuntos de dados cujo tamanho está além da capacidade das ferramentas típicas de captura, armazenamento, gerenciamento e análise de bancos de dados. Esta definição é intencionalmente subjetiva e incorpora uma inconstância de quão grande um conjunto de dados precisa ser para ser considerado Big Data. Ou seja, não definimos Big Data em termos de ser maior do que determinado número de terabytes. Assumimos que, conforme a tecnologia avança com o tempo, o tamanho necessário para que os conjuntos de dados sejam qualificados como Big Data também aumenta. Note também que a definição pode variar de acordo com o setor, dependendo de que tipos de ferramentas de desenvolvimento são comumente utilizadas e que tamanho de conjuntos de dados são comuns para uma determinada indústria. Com essas ressalvas, hoje em dia Big Data pode variar em diversos setores entre algumas dezenas de terabytes a até mesmo múltiplos petabytes (milhares de terabytes).

Tabela 1 – Definições de Big Data.

Podemos observar que são diversas as abordagens a respeito do tema, todavia elas convergem ao concordarem que Big Data se trata de uma grande quantidade de dados, como pode ser visto nas abordagens dos diferentes autores. E de acordo com a constante e crescente produção de dados, o conceito de Big Data foi se tornando cada vez mais comum, em conformidade com (LYMAN; VARIAN, 2003) a quantidade de dados criados no ano 2000 foi de dois exabytes, sendo esse, o mesmo valor criado diariamente no ano de 2011. Já em 2020, o valor ultrapassou os 44 zettabytes segundo (ZWOLENSKI; WEATHERILL, 2014). Devemos observar também que além da quantidade de dados produzidos, outra característica primordial é seu tipo de dado ou sua forma.

Segundo (PARK; LEYDESDORFF, 2013), os primeiros indícios de Big Data surgiram na década de 70, dando como exemplo o livro “Concise Survey of Computer Methods” escrito por (NAUR, 1974) e publicado em 1974, mas foi na década de 90 que o campo de pesquisa em Big Data começou a se desenvolver de fato quando o conceito passou a ser associado a modelos computacionais e desenvolvimento de software para grandes conjuntos de dados. Eles descrevem que o propósito fundamental das pesquisas em Big Data é o de extrair dados fundamentais para descobrir padrões ocultos em grandes volumes de dados de uma determinada fonte, que pode ser tecnológica, sociológica, ou econômica. Descrevem que este também pode ser referenciado como “The Fourth Paradigm”, onde se referem a uma citação de Jim Gray, um cientista famoso da Microsoft (HEY; TANSLEY; TOLLE, 2009)

Segundo (ROUSSEAU, 2012), o termo Big Data pode se referir a diferentes quantidades de dados em diferentes cenários, para uma determinada empresa pode ser que algumas centenas de Gigabytes sejam suficientes para que eles repensem suas tecnologias de gestão e processamento de dados, enquanto para outras pode ser que esse valor chegue a Petabytes ou Exabytes.

Para (LANEY, 2001), existem três grandes importantes V’s para manipulação de grandes volumes de dados, sendo estes:



Figura 3 – Representação gráfica dos 3’vs de Big Data

- Volume: É necessário se preocupar com a quantidade de dados.
- Velocidade: Deve-se observar a velocidade com que esses dados são gerados.
- Variedade: É de grande importância considerar que esses dados podem vir de diversas fontes e com diferentes formatos.

Desde então, diferentes autores adotaram novos termos a esta lista. De acordo com (NICULESCU, 2020), por exemplo, temos 7 V's, sendo os três anteriores e mais quatro adicionais:

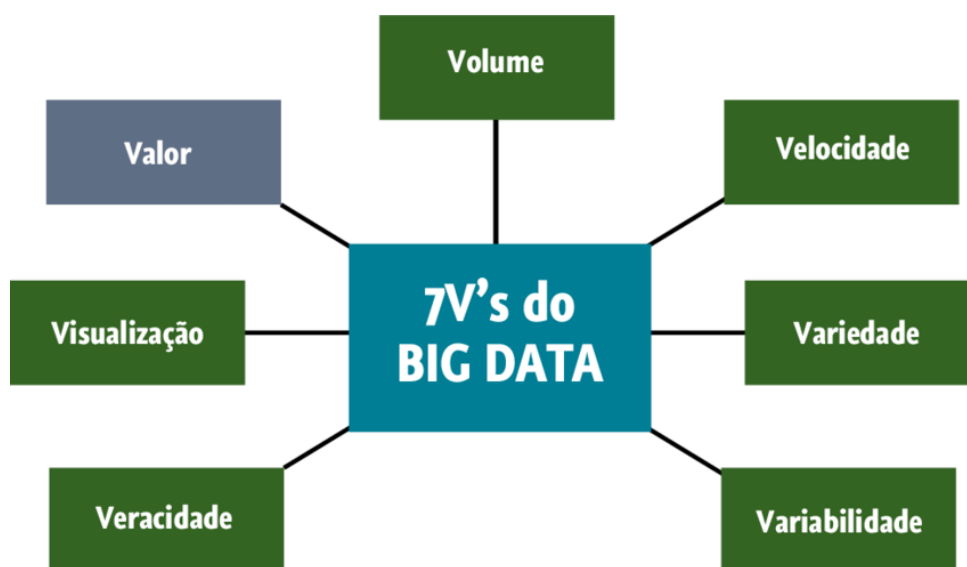


Figura 4 – Representação gráfica dos 7'vs de Big Data adaptada de (NICULESCU, 2020)

- Variabilidade: O aumento no alcance dos valores e a contínua mutabilidade dos dados.
- Valor: Que aborda a necessidade de avaliação dos dados da empresa e de informações adicionais que os dados podem trazer para gerar conhecimento.
- Veracidade: Que indica a confiabilidade e ambiguidades herdadas devido à incerteza e inconsistência dos dados.
- Visibilidade: O estado de ser visível e acessível. A visibilidade é importante porque dados de fontes distintas precisam ser analisados e dados críticos podem existir, mas se eles não forem visíveis para os processos de um sistema de Big Data, a análise completa se torna impossível; mas, ao mesmo tempo, a visibilidade não autorizada é um risco.

Para se trabalhar com Big Data precisamos de ferramentas especializadas, e para nosso trabalho escolhemos Apache Kafka, Apache Spark e Apache Parquet ferramentas atuais

que se complementam com excelente desempenho e fácil utilização.

## 2.2 APACHE KAFKA

O Apache Kafka (APACHE, 2021a) é uma plataforma de Streaming de dados que foi originalmente criada pela empresa LinkedIn Corporation (CORPORATION, 2021), teve seu desenvolvimento iniciado em 2010 e foi lançada em 2011 como um projeto open-source (CONFLUENT, 2021). Alguns anos depois, em 2014 alguns membros da equipe original que criaram a ferramenta no LinkedIn (CORPORATION, 2021) fundaram uma nova empresa, a Confluent (CONFLUENT, 2021), com o objetivo de prover soluções e serviços relacionados ao Apache Kafka (APACHE, 2021a). Em 2019 já estava presente em 60% das empresas listadas pela “Fortune 100” (FORTUNE, 2021), uma classificação das 100 maiores empresas públicas e privadas dos Estados Unidos com base na receita (RAO; CONFLUENT, 2021). A forma como manipulamos os dados é tão importante quanto os dados em questão. O Apache Kafka (APACHE, 2021a) funciona como uma ferramenta de comunicação entre dados e aplicação (Streaming) que possui uma estrutura desenvolvida para realizar esta tarefa de forma rápida, escalável, robusta e segura, sendo capaz de permitir, publicar e ler fluxos de dados, armazenar dados de forma que seja tolerante a falhas e processar dados conforme são gerados (APACHE, 2021a). Empresas como Walmart utilizam Kafka para processar mais de 10 bilhões de eventos diariamente (RAO; CONFLUENT, 2021).

Segundo (NANNONI, 2015), a plataforma permite publicação e consumo de mensagens de maneira que os produtores e consumidores realizam a tarefa independente uns dos outros. Além de possuir cópias de segurança dos tópicos de dados através de múltiplas partições em diferentes servidores, garantindo uma ótima segurança e tolerância a falhas.

O Apache Kafka (APACHE, 2021a) tem como padrão não utilizar memória do servidor, ele tem como regra forçar a persistência de mensagens para o disco, tendo otimizações para ser capaz de realizar um grande volume de operações de escrita em disco. Suas chamadas de I/O são feitas diretamente para o Kernel do sistema operacional. Os dados saem do produtor para o consumidor divididos em partes, organizadas em tópicos, visando a eficiência da compressão de dados e reduzindo a latência. Cada tópico pode ser dividido em centenas de partições, podendo um único processamento ser dividido em milhares de servidores, o que garante uma ótima escalabilidade. Sendo assim, ele permite centenas de partições em um simples tópico, dividindo o processamento entre centenas ou até milhares de servidores. Assim é possível manipular grandes quantidades de dados sob demanda. Contudo, para garantir sua performance, as escritas são feitas de forma imutável e sequencial, impedindo acessos randômicos ao disco (NANNONI, 2015).

### 2.2.1 Arquitetura

O padrão seguido pela arquitetura, onde há publicação e assinatura de mensagens, permite entregar e consumir estas de forma distribuída, através de múltiplas aplicações e de forma assíncrona, garantindo a geração de dados desacoplada (AMAZON.COM, 2021).

Segundo (NANNONI, 2015), a arquitetura do framework consiste em 5 partes essenciais: produtores, brokers, consumidores, tópicos e partições.

- Os tópicos podem ser comparados com uma arquitetura de filas, sendo a maior entidade de agrupamento de mensagens. As mensagens são enviadas por um produtor para um tópico e partição específicos e cada partição possui um offset de mensagens, podendo ocorrer de as mensagens não serem distribuídas igualmente através das partições, isto pode gerar complicações e a orientação é de que sejam de fato distribuídas igualmente. Esta definição fica à cargo do produtor (NANNONI, 2015).
- Produtores por sua vez são os responsáveis por produzir as mensagens que serão transmitidas. Dados podem ser produzidos por diversas fontes, como aplicações em Java, Dot Net, entre outras opções (APACHE, 2021a).
- Um Broker é uma instância do Apache Kafka (APACHE, 2021a) em execução dentro de um ou diversos servidores, ou máquinas virtuais, que recebe e redistribui mensagens, armazena partições e é responsável pela interação com o Apache Zookeeper (JOHANSSON, 2021).
- Por fim, os consumidores são os responsáveis pela leitura das mensagens enviadas. Consumidores podem ser organizados em grupos de consumidores quando mais de um consumidor está designado para realizar a mesma tarefa dentro do mesmo tópico e mesmas partições. Desta forma, as mensagens enviadas para um grupo de consumidores devem, cada uma, ser direcionadas apenas para um integrante do grupo que será responsável por interpretar aquela mensagem. Além disso, é possível que diferentes grupos de consumidores se inscrevam para receber as mesmas mensagens, como mostra a imagem abaixo (NANNONI, 2015).

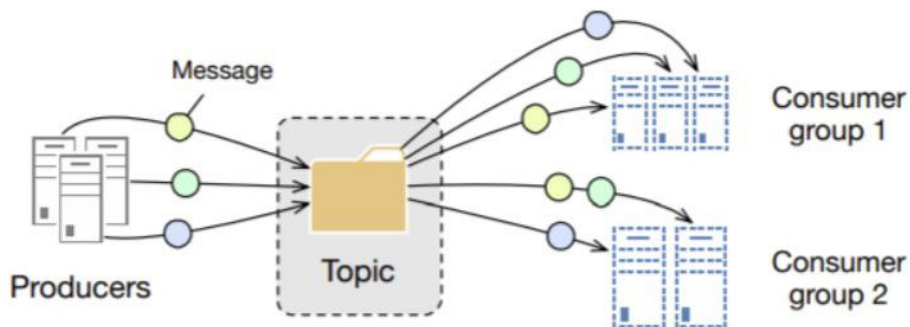


Figura 5 – Arquitetura Produtor e Consumidor de um Tópico no Kafka (NANNONI, 2015)

Além disso, possui uma dependência em um segundo framework, Apache Zookeeper (APACHE, 2021b), que funciona como um agendador, definindo quando as mensagens podem ser enviadas, podendo garantir que uma mensagem seja enviada apenas quando todas as suas réplicas tenham sido feitas. Também funciona como um banco de dados compartilhado (NANNONI, 2015).

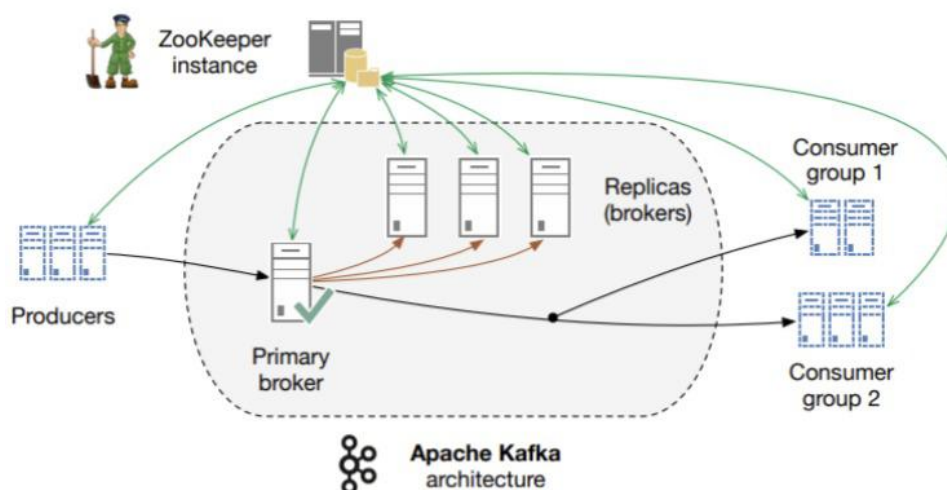


Figura 6 – Arquitetura geral com Zookeeper no Kafka (NANNONI, 2015)

### 2.2.2 Aplicabilidade

Como o Apache Kafka é capaz de armazenar grandes quantidades de dados redundantes, com baixa latência, boa disponibilidade, e alta performance, essa ferramenta é utilizada em diversos cenários de desenvolvimento de software como :

- Troca de mensagens entre sistemas.
- Coleta de dados geográficos. Sendo utilizado principalmente em serviços como Uber e Google Maps.

- Monitoramento e armazenamento de logs.
- Processamento de stream de dados.
- Data Warehouse para aplicação de Bussiness Intelligence.

## 2.3 APACHE SPARK

O Apache Spark foi originalmente desenvolvido no AMPLab da universidade de Berkeley em 2009 com o intuito de criar uma estrutura de processamento de dados poderosa, que fornece uma ferramenta de fácil utilização para a análises eficientes de dados heterogêneos (INOUBLI et al., 2018). Ele possui uma larga vantagem competitiva com as melhores ferramentas do mercado, como o Hadoop e Apache Storm (INOUBLI et al., 2018). Sendo utilizado em diversas Big Tech's como a Yahoo, a Baidu e a Tencent. Um conceito chave do Spark é a utilização de conjunto de dados distribuídos e resilientes (RDDs) que é uma abstração para o processamento de dados distribuídos, em memória e com tolerância a falhas. Segundo Zaharia (ZAHARIA et al., 2012), os RDDs são motivados por dois principais tipos de aplicações que as estruturas de computação atuais manipulam de uma certa maneira ineficientemente: Algoritmos iterativos e ferramentas de mineração de dados iterativas. Nesses dois casos, manter os dados em memória pode melhorar o desempenho significativamente. Para tolerar as falhas com eficiência, os RDDs fornecem uma forma restrita de memória compartilhada, com base em transformações *coarse-grained updates*, que tem como objetivo fazer com que cada transformação realizada dentro da execução do programa, seja salva em um log, podendo fazer a recriação desse RDD caso ele seja perdido, ao contrário de outras arquiteturas que para o processamento distribuído utilizam o *fine-grained updates*, isto é, todos os dados e atualizações realizadas durante o processamento são replicados através dos nós do cluster, causando uma grande utilização no uso de banda e no espaço de armazenamento.

### 2.3.1 Arquitetura

A arquitetura do Spark é composta por dois principais componentes, o Driver e os Workers. O Driver é o local onde é iniciada a execução da aplicação, e os Workers são os locais em que são realizadas as computações da aplicação.

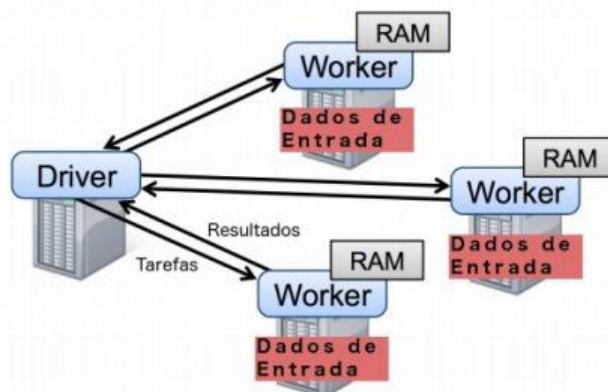


Figura 7 – Arquitetura Cluster Spark adaptado de (ZAHARIA et al., 2012)

A arquitetura do spark é baseada no master/slave, na figura acima vemos claramente uma comunicação bilateral entre os workers e o driver, mas não há comunicação entre os workers. Isso se deve por conta da característica dessa arquitetura, onde a comunicação do nó mestre é feita bilateralmente somente com os nós escravos, sem comunicação entre eles.

- Driver: É responsável por orquestrar a execução do processamento de dados
- Worker: São as máquinas onde são executados os programas. Caso o Spark seja executado em uma única máquina local, ela irá desempenhar o papel tanto de Driver quanto de Worker, conhecido como modo de execução standalone.

### 2.3.2 Tolerância a falhas

Spark utiliza um sistema para tolerar falhas no nível das tarefas. Sendo então possível reexecutar as tarefas a partir dos pontos que as falhas ocorreram (ZAHARIA et al., 2012). Esse método utiliza o método de linhagem de modificações do RDD, na qual ele identifica o ponto em que a tarefa parou sua execução. Então, caso exista uma falha, essa mesma tarefa é re-executada em outro nó desde que o pai do estágio atual ainda esteja disponível.

### 2.3.3 Módulos Spark

Além do núcleo do Spark, alguns projetos adicionais foram adicionados para complementar toda a funcionalidade fornecida pelo núcleo. A imagem a seguir mostra os núcleos do Spark.



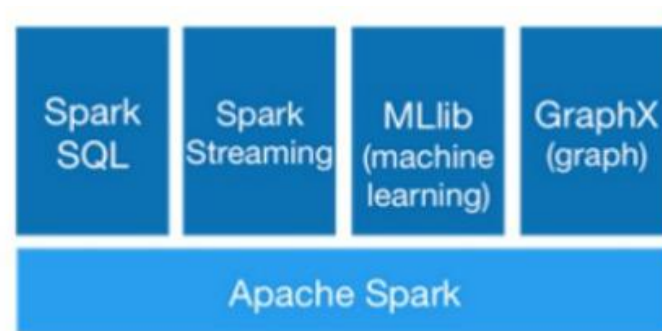


Figura 8 – (ZAHARIA et al., 2012)

- Spark SQL: Trás a ideia dos DataFrames, que é uma estrutura de dados para dados semi-estruturados e estruturados. Ele oferece a oportunidade de introduzir as consultas SQL nos códigos Spark fornecendo suporte para a linguagem SQL (GARCÍA-GIL et al., 2017).
- Spark Streaming: É voltada para o processamento de dados, com diversas propriedades, dentre elas a fácil escalabilidade e a tolerância a falhas. Funciona dividindo o stream de dados em lotes de alguns milissegundos e processando cada lote desse como um RDD no spark. Para prevenir falhas, o Spark Streaming por default persiste e replica todos os dados duas vezes. Como é facilmente escalável, ele funciona com diversas fontes de dados como: HDFS, Apache Flume ou Apache Kafka (GARCÍA-GIL et al., 2017).
- MLib (Machine Learning): Um dos grandes objetivos do spark foi proporcionar uma melhor performance para processos iterativos. O Mlib consiste em uma biblioteca de códigos de machine learning prontos e de fácil utilização, funcionando de forma muito parecida com os pacotes já disponíveis no python o numpy e o scikit-learn. Esta biblioteca foi especialmente projetada para simplificar pipelines de Machine learning em ambientes de grande escala.
- GraphX: Segundo (SPARK, 2021), é um componente do Spark para grafos e computação paralela destinadas a grafos. o GraphX estende o Spark RDD introduzindo uma nova abstração de Graph: um multigrafo direcionado com propriedades anexadas a cada vértice e aresta. Ele oferece um amplo suporte à computação de grafos, possuindo um grande conjunto de operados fundamentais como: subgraph, joinVertices e aggregateMessages.

## 2.4 APACHE PARQUET

O Apache Parquet é um popular formato de armazenamento colunar que possui código aberto e é bem estabelecido para aplicações que utilizam o Apache Spark, Ecosistema Hadoop, AWS, Azure e entre outros. Esse projeto se iniciou com uma parceria entre o Twitter e a Cloudera, tendo a sua primeira versão sendo liberada em 2013 (PARQUET, 2013). Diferente dos modelos tradicionais de armazenamento como o formato CSV e o JSON que utilizam uma abordagem orientada a linhas, os arquivos no Apache Parquet são colunares. Ele foi criado principalmente para suportar compressão e codificação de uma forma eficiente, sendo possível especificar a compressão por coluna, além de ser otimizado para trabalhar com estruturas de dados complexas. Esses benefícios vieram em virtude de um algoritmo *record shredding and assembly algorithm* disponibilizado em (MELNIK et al., 2010) e implementado no core do Parquet.

### 2.4.1 Formato Colunar vs Formato de linha

A figura ilustra de forma simples como é feito o armazenamento pelos modelos de colunas e de linhas.

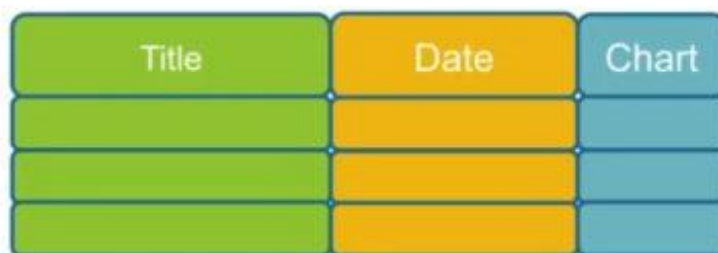


Figura 9 – Exemplificação de formato colunar

Na figura acima temos três campos, Título, Data e gráfico. No formato de linha os dados são armazenados de maneira adjacente com seu tipo da coluna como exemplo na imagem abaixo:



Figura 10 – Armazenamento de dados em formato de linha

Os dados são armazenados em conjunto com seus tipos um do lado do outro, diferentemente o formato colunar que armazena os valores da coluna de forma adjacente uma da outra como na imagem a seguir :



Figura 11 – Armazenamento de dados em formato colunar

No formato colunar acima, os dados são armazenados adjacentes aos valores de suas colunas.

### 2.4.2 Eficiência

Um estudo foi realizado pela (DATABRICKS, 2021) para medir a eficiência do armazenamento colunar e o por linhas demonstrou que o formato armazenado pelo Parquet em comparativo com o formato utilizado pelo CSV é muito mais eficiente. A tabela abaixo produzida pela Databricks detalha essa diferença.

Dataset	Size on Amazon S3	Query Run Time	Data Scanned	Cost
Data stored as CSV files	1 TB	236 seconds	1.15 TB	\$5.75
Data stored in Apache Parquet Format	130 GB	6.78 seconds	2.51 GB	\$0.01
Savings	87% less when using Parquet	34x faster	99% less data scanned	99.7% savings

Figura 12 – Comparativo entre arquivos no formato CSV e no formato Parquet (DATABRICKS, 2021)

Com essa demonstração ficou claro a eficiência da utilização do formato colunar pelo Apache Parquet. O espaço de armazenamento foi reduzido em 87%, escaneou 99% menos dados e executou 30 vezes mais rápidos em determinadas operações (DATABRICKS, 2021).

### 2.4.3 Vantagens em armazenar os dados em formato colunar

- O armazenamento colunar foi projetado para trazer mais eficiência em relação a arquivos baseados em linhas como JSON e CSV. A consulta dos dados no armazenamento colunar, facilita o pulo dos dados não relevantes na busca muito rapidamente. Tendo como resultado, resultados das consultas de agregação com menos tempo em comparação a bancos por linhas.
- Suporta compressão flexíveis e esquemas de codificações eficientes
- Funciona em diversas tecnologias como: AWS Athena, Amazon Redshift Spectrum, Google BigQuery, Google Dataproc, Apache Spark e entre muitos outros.

## 2.5 MACHINE LEARNING

De acordo com (LUGER, 2004), na década de 40 surgiam os primeiros estudos relativos a IA (Inteligência Artificial), e esta estava restrita à universidades, laboratórios e institutos de pesquisa. As aplicações da área possuíam um contexto muito limitado, específico e eram embarcadas com hardware.

Segundo Faceli et al.(FACELI et al., 2011), a IA vem sendo aplicada em diversas áreas da atividade econômica, por conta tanto do avanço dos estudos relativos à mesma quanto dos avanços tecnológicos, e atualmente se tornou uma das maiores áreas de pesquisa e estudo, desenvolvendo aplicações práticas com grande valor comercial e social.

Faceli et al.(FACELI et al., 2011) também dizem que, conforme cresce a complexidade dos problemas a serem tratados computacionalmente e o volume dados gerados, torna-se clara a necessidade de ferramentas computacionais mais robustas e mais independentes, para reduzir a necessidade de intervenção humana. Para atingir este objetivo, estas ferramentas devem ser capazes de utilizar alguma base (conjunto de treino) para gerar uma hipótese de como resolver o problema. Este processo recebe o nome de Aprendizado de Máquina. Nos algoritmos de Aprendizado de Máquina, deve-se induzir uma função ou hipótese capaz de resolver o problema a partir de dados que representam instâncias do problema. Este conjunto de dados recebe o nome de dataset, e instâncias de dado deste dataset, também chamadas de objeto, possuem valores característicos que as descrevem (atributos). Determinadas tarefas podem possuir um tipo de atributo especial chamado atributo de saída, ou atributo alvo, que possui a característica de poder ser estimado através dos valores dos demais atributos do dataset, que podem ser chamados de atributos previsores.

Russell e Norvig(RUSSELL; NORVIG, 2009) dividem os algoritmos de aprendizagem de máquina em três classificações:

- Supervisionado: Quando os dados são rotulados, ou seja a máquina sabe qual deverá ser a saída de cada entrada;
- Não supervisionado: Quando a máquina não sabe qual deverá ser a saída e identifica semelhanças nos dados, reagindo com base na presença ou ausência de tais semelhanças;
- Aprendizado por reforço: Quando o aprendizado é feito através de sistemas de recompensa, onde a máquina é capaz de acompanhar o impacto de suas decisões e trabalha para minimizar suas perdas;

Faceli et al.(FACELI et al., 2011) classificam as tarefas indutivas do aprendizado de máquina em duas categorias:

- Preditivas, onde o objetivo é encontrar uma função, a partir do conjunto de treino, que possa ser utilizada para prever valores do atributo alvo, esta função também pode ser chamada de hipótese ou modelo. Para isso se utiliza como base os demais atributos do dataset, que devem estar rotulados, e conseqüentemente, os algoritmos utilizados nesse tipo de tarefa se comportam como os de um aprendizado de máquina supervisionado.
- Descritivas, onde o objetivo é explorar ou descrever um dataset. Nas tarefas classificadas como descritivas, não se utiliza o atributo de saída, e por isso seguem o paradigma de aprendizado não supervisionado. Um bom exemplo de tarefa descritiva pode ser o agrupamento de dados, onde buscamos encontrar similaridades entre grupos de dados de um dataset.

Faceli et al.(FACELI et al., 2011) dão exemplos de aplicações que utilizam técnicas de aprendizado de máquina, como:

- Reconhecimento de palavras faladas;
- Predição de taxas de cura de pacientes com diferentes doenças;
- Detecção de Fraudes em cartões de créditos;
- Veículos Autônomos;
- Detecção de potenciais ameaças ao computador;
- Detecção de tendências de valorização de ações no mercado financeiro;

### 2.5.1 Aprendizado Supervisionado

Russell e Norvig (RUSSELL; NORVIG, 2009) definem que no aprendizado supervisionado o objetivo é encontrar uma função  $h(x)$  que aproxime uma função  $f(x)$ , de forma que dado um conjunto de treinamento com  $n$  pares de entrada e saída,  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \in X \times Y$ , cada saída  $y_i$  pode ser obtida através de uma função  $y_i = f(x_i)$ , onde o domínio  $X$  e o contra-domínio  $Y$  não precisam ser necessariamente numéricos.

O processo de aprendizagem é na verdade buscar a função que trará o melhor desempenho, tendo em mente que na aplicação prática surgirão exemplos que vão além do conjunto conhecido. Para isso é utilizado um conjunto conhecido de dados, chamado conjunto de testes, contendo apenas exemplos que não foram utilizados no conjunto de treinamento. A função  $h$  deve encontrar corretamente os valores  $y_i$  para seus pares  $x_i$  do conjunto de testes e do conjunto de treinamento.

Se a saída  $y$  pertencer a um conjunto discreto, o problema de aprendizagem é denominado como classificação, onde tenta-se encontrar o elemento que melhor representa  $y$ . No caso de  $y$  pertencer a um conjunto contínuo, o problema é denominado como regressão, e o objetivo é estimar o valor  $y'$  que melhor aproxime  $y$ . Neste tipo de problema, a probabilidade de se encontrar exatamente o valor real de  $y$  tende a zero. (RUSSELL; NORVIG, 2009)

### 2.5.2 Regressão Logística

De acordo com (CRAMER, 2003) a regressão logística foi descoberta no século XIX para descrever o crescimento das populações e as reações químicas no curso de autocatálise (reações em cadeia).

Segundo (HOSMER; LEMESBOW, 1980), a regressão logística possui a característica de que a variável resposta ( $Y$ ) é dicotômica, ou seja, seu valor está entre 1 ou 0, sim ou não, falha ou sucesso, a variável está entre dois possíveis valores ou categorias.

Com probabilidades :

- Para **sucesso** :

$$P_{sucesso} = \pi_i = P(Y = 1|X = x_i) \quad (2.1)$$

- Para **fracasso** :

$$P_{fracasso} = 1 - \pi_i = P(Y = 0|X = x_i) \quad (2.2)$$

Sendo a variável ( $\pi_i$ ) utilizada para descrever a média condicional de  $Y$  dado  $X$  com a distribuição logística.

O modelo de Regressão Logística é semelhante ao modelo de Regressão Linear, sendo este estabelece uma relação entre as variáveis explicativas e a probabilidade de ocorrer ou não o fenômeno estudado, permitindo então criar uma variável binária para estimar a probabilidade de classificação de sucesso ou fracasso. No caso do desenvolvimento da framework, foram utilizados as features que seriam o conjunto de palavras que determinam ou não a ocorrência de um sentimento suicida dentro do texto. É considerado então o problema de classificar os comentários em suicidas ou não suicidas. Como é um problema de classificação binária, a regressão logística funciona aplicando a função sigmoide (função logística), ao produto interno do vetor de features com o vetor de pesos, retornando um valor no intervalo de 0 a 1 que representa no caso a probabilidade daquele tweet possuir uma ideia suicida. A hipótese dada para um comentário qualquer  $x$  é dada pela equação :

$$h_{\theta(x)} = g(z) \quad (2.3)$$

onde  $z$  é dado por:

$$z = \theta^T * x \quad (2.4)$$

e  $g$  é a famosa função sigmoide da regressão logística e  $\theta$  é o vetor de pesos:

$$g(z) = \frac{1}{1 + e^{-z}} \quad (2.5)$$

A partir de um valor já bem utilizado nesse tipo de regressão, o 0.5, um comentário é não suicida, se o valor da função logística for menor que isso, caso contrário, é um comentário suicida.

$$h_{\theta(x)} \geq 0.5 \quad (2.6)$$

$$h_{\theta(x)} < 0.5 \quad (2.7)$$

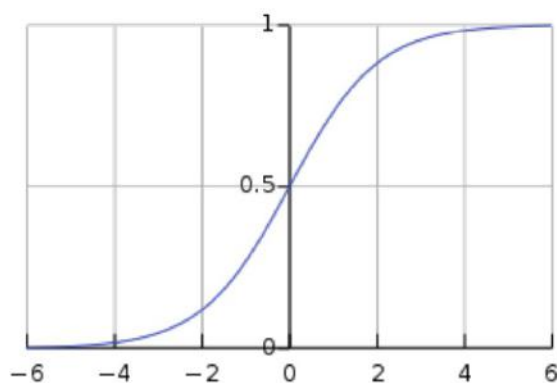


Figura 13 – Gráfico da função sigmoide

A parte do treino na regressão logística é basicamente estimar o vetor de pesos. O Algoritmo então define uma função de custo que para cada comentário classificado, diz o quão longe a hipótese atual chegou da classificação correta. Inicialmente há então um peso  $J(\theta)$  que é a média aritmética do custo de cada comentário da etapa de treino. É necessário então minimizar essa função custo  $J(\theta)$ , possuindo então os valores de  $\theta$  que preveem com o menor erro possível as classificações reais dos comentários.

$$J(\theta) = \frac{1}{m} * \sum_{n=1}^m \text{Custo}(h_{\theta}(x_i), y_i) \quad (2.8)$$

Temos então um problema que envolve minimização, é utilizado geralmente o método do gradiente descendente que tem como objetivo encontrar o mínimo de uma função. Se a função for convexa então esse mínimo será global. Esse método parte da ideia de que se movermos um passo em direção contrária ao gradiente da função em um determinado ponto, estaremos então indo para o mínimo dessa função, determinamos o tamanho desse passo de taxa de aprendizagem, precisamos ter uma certeza somente da função utilizada ser convexa, para que então encontrar um mínimo global.

Como  $J(\theta)$  é dada em uma função de cada par  $h_{\theta}(x_i, y_i)$ , onde  $x_i$  é o vetor de features que representa o comentário  $i$  e  $y_i$  é sua respectiva classificação. É necessário somente definir uma função Custo  $h_{\theta}(x_i, y_i)$  que garanta que  $J(\theta)$  seja uma função convexa e então utilizar o método do gradiente para encontrar um mínimo global e obter a certeza que são encontrados os valores ótimos de  $\theta$  para o modelo. A função Custo é definida como:

$$\text{Custo}(h_{\theta}(x), y) = -y \log(h_{\theta}(x)) - (1 - y) \log(1 - h_{\theta}(x)) \quad (2.9)$$

substituindo a equação acima (2.8) em (2.9) é obtido então a função  $J(\theta)$  que é uma função convexa, para que então utilizar o método do gradiente. A equação é transformada em vetorial para facilitação das contas:

$$J(\theta) = \frac{1}{m} * (-y^T \log(h) - (1 - y)^T \log(1 - h)) \quad (2.10)$$

onde:

$$h = (X\theta) \quad (2.11)$$

$$\theta = \theta - \frac{\alpha}{m} X^T (g(X * \theta) - y) \quad (2.12)$$

Como esse método pode demorar várias iterações para convergir a um mínimo global. Um  $\epsilon$  é definido para qual a diferença do valor da função custo de uma iteração para outra for menor que esse valor definido, o algoritmo encontrou então um mínimo global. A partir do dataset já classificado dos dados de treino, para cada comentário é obtido então seu respectivo vetor de features, e aplicado então a função logística a ele obtendo uma predição do classificador.



### 2.5.3 Overfitting

(FACELI et al., 2011) definem overfitting como um caso específico onde a hipótese se ajusta perfeitamente para o conjunto de dados utilizado durante o treino, mas não consegue prever resultados bons para novos exemplos. Neste caso a hipótese se especializou, ou memorizou, o conjunto de treinamento e não consegue bons resultados fora dele.

Uma técnica definida por eles (FACELI et al., 2011) para evitar o overfitting é o early stopping, onde o aprendizado é interrompido quando o conjunto de treino apresenta um decaimento desprezível após  $n$  iterações, impedindo que o modelo preditivo memorize o conjunto de treino.

### 2.5.4 Metricas de desempenho

Faceli et al. (FACELI et al., 2011) definem as métricas de desempenho como maneiras de medir a eficiência dos modelos criados. Sendo importantes indicadores da distância entre os resultados obtidos e a realidade.

### 2.5.5 Matriz de Confusão

Segundo Faceli et al. (FACELI et al., 2011), a matriz de confusão é um método utilizado para visualizar o desempenho de um determinado classificador definindo suas taxas de erro e de acerto. É montada uma matriz cujas linhas indicam a qual classe os objetos pertencem, enquanto as colunas indicam qual classe o algoritmo previu para os objetos.

Por simplicidade, eles (FACELI et al., 2011) definem que para um problema com duas classes, geralmente uma classe é denotada positiva (+) e a outra é denominada negativa (-),

Para essa matriz, colocando na primeira linha a classe positiva e na segunda linha a classe negativa, na primeira coluna a classe positiva e na segunda coluna a classe negativa, temos as seguintes definições:

- VP, correspondente ao número de verdadeiros positivos, onde a classificação foi feita corretamente para a classe positiva, e está na primeira linha e primeira coluna na configuração estabelecida.
- VN, correspondente ao número de verdadeiros negativos, onde a classificação foi feita corretamente para a classe negativa, e está na segunda linha e segunda coluna na configuração estabelecida.
- FP, correspondente ao número de falsos positivos, onde a classificação foi feita erroneamente para a classe positiva, e está na segunda linha e primeira coluna na configuração estabelecida.

- FN, correspondente ao número de falsos negativos, onde a classificação foi feita erroneamente para a classe negativa, e está na primeira linha e segunda coluna na configuração estabelecida.

		Classe predita	
		+	-
Classe verdadeira	+	VP	FN
	-	FP	VN

Figura 14 – Matriz de Confusão

Utilizando a matriz de confusão pode-se derivar diversas métricas de desempenho para problemas de classificação, como acurácia, taxa de erro, especificidade e sensibilidade.

### 2.5.6 Análise Exploratória

Faceli et al.(FACELI et al., 2011) definem a análise exploratória como uma análise detalhada das características que permite a descoberta de padrões e tendências dentro de um conjunto de dados. Possibilitando extrair um grande número de informações deste conjunto, adquirindo um melhor entendimento do problema e uma melhor e mais eficiente modelagem para a solução. Essas características podem ser obtidas através de estudos estatísticos do problema utilizando valores como média, moda e mediana, ou até mesmo induzidas observando o conjunto e gerando representações visuais como gráficos e histogramas.

## 2.6 COMPUTAÇÃO EM NUVEM

De acordo com (SILBERSCHATZ; BAER; GAGNE, 2015) a computação em nuvem ou *Cloud Computing* é um novo modelo de computação que permite a distribuição de recursos computacionais e até mesmo aplicativos via rede, tendo como base a virtualização, permitindo ao usuário final acessar uma grande quantidade de serviços e aplicações em diversos lugares e independente de plataforma. Deste modo, o usuário efetua o pagamento desses serviços sob demanda e os utiliza. Os ambientes de computação em nuvem podem ser classificados como:

- **Nuvem pública:** Infraestrutura de nuvem é disponibilizada para o público em geral, sendo possível ser acessado por qualquer usuário;
- **Nuvem privada:** Infraestrutura de nuvem é utilizada exclusivamente por uma organização;
- **Nuvem comunitária:** Infraestrutura de nuvem é exclusivo para o uso de uma comunidade específica, ou seja, organizações que possuem interesses em comum, como projetos, metas e objetivos;
- **IaaS:** Infraestrutura como Serviço, fornecendo servidores ou espaço de armazenamento via internet;
- **CaaS:** Contêiner como Serviço, fornecendo contêiners via internet;
- **PaaS:** Plataforma como Serviço, apresentando uma pilha de software pronta para uso via internet;
- **SaaS:** Software como Serviço, oferecendo aplicativos para uso via internet.

## 2.7 DOCKER

Para aproveitar ao máximo os recursos já apresentados precisamos de um sistema distribuído que seja fácil de manusear, para isso escolhemos utilizar o Docker, uma plataforma de virtualização que foi lançada em 2013 (DOCKER, 2022b) que utiliza um conceito de contêineres, onde os recursos necessários para uma aplicação se tornam menos vinculados a máquina específica que estão instalados.

Algumas vantagens que o Docker trás são as de controle sobre os contêineres, como:

- Limitar o uso de CPU
- Limitar o uso de I/O
- Limitar o uso de memória

- Limitar o uso de rede
- Sistema de arquivos isolados
- Controle de permissões

### 2.7.1 Virtualização

A virtualização não é nenhuma novidade na área tecnológica e já existe no mercado a décadas, porém com o avanço tecnológico ela se torna cada vez mais relevante. Segundo (GOLDEN; SCHEFFY, 2008), os data centers costumam utilizar apenas de 10 a 15% de suas capacidades totais de processamento, deixando a maior parte de sua capacidade ociosa e sem aproveitamento. Porém isso não significa que estejam consumindo menos energia, e acabam gerando os mesmos custos computacionais que gerariam se estivessem aproveitando 100% de suas capacidades. Uma das soluções para esse problema é a virtualização, segundo (VERAS, 2011) ela permite alterar a infraestrutura rapidamente utilizando instrumentos lógicos ao invés de físicos, estabilizando o ambiente e tornando as aplicações independentes do hardware. Com essa desvinculação das aplicações e o sistema operacional dos recursos físicos você pode criar diversos ambientes virtuais simultâneos em uma mesma máquina aproveitando ao máximo seu hardware.

### 2.7.2 Contêineres

Contêineres por sua vez são um tipo de virtualização específica, segundo (YU, 2007) a virtualização em um nível de sistema operacional se define pela existência de múltiplas instâncias isoladas de espaços de usuário, estes gerenciados pelo kernel do sistema. Essas instâncias representam todo um ambiente de execução, os contêineres, que também armazenam todas as dependências, bibliotecas e arquivos de configuração necessários para o funcionamento da aplicação. Este modelo torna possível criar diversas partições lógicas de forma isolada mas que compartilham o mesmo kernel, permitindo que o sistema operacional nativo possa acessar e controlar todas as instâncias.

A vantagem de se utilizar contêineres é que não é necessário criar uma máquina virtual completa, ou seja, não é necessário virtualizar um sistema operacional para virtualizar uma aplicação, com isso se tem uma otimização, economizando recursos computacionais.

### 2.7.3 Motores de contêiner

O uso de contêineres se tornou algo amplamente utilizado com o surgimento das chamadas container engines, ou motores de contêiner, de código aberto, o Docker é uma destas e é a que utilizaremos na nossa aplicação. Essas ferramentas trouxeram junto delas um conceito de imagem de contêiner, que por sua vez é um modelo somente leitura que

possui instruções para que a máquina possa criar um contêiner, no nosso caso do Docker são chamadas de Docker Images (DOCKER, 2022a).

## **2.7.4 Arquitetura Docker**

O Docker trás consigo alguns conceitos como Docker Daemon, Docker Client, Docker Registries, Docker Host e Docker Objects, sobre o qual falaremos brevemente. (DOCKER, 2022a)

### **2.7.4.1 Docker objects**

Quando se utiliza o Docker você está criando e utilizando imagens, contêineres, redes e outros objetos que são genericamente chamados de Docker Objects.

### **2.7.4.2 Docker Daemon**

É o responsável por ouvir aos pedidos da API do Docker e gerencia os Docker Objects, também pode se comunicar com outros Docker Daemons para gerenciar serviços do Docker.

### **2.7.4.3 Docker Client**

É o meio que o usuário do Docker tem de se comunicar com o Docker, uma interface onde você escreve comandos, que serão enviados para o Docker Daemon. O Docker Client pode se comunicar com diversos Docker Daemons simultaneamente.

### **2.7.4.4 Docker Registries**

É um local de armazenamento de Docker Images, existe um domínio público chamado Docker Hub que possui diversas imagens prontas para uso, o Docker já vem configurado para interagir com o Docker Hub, mas o usuário também pode criar registros privados de imagens.

### **2.7.4.5 Docker Host**

O Docker Host é o ambiente em que estão rodando o Docker Daemon e onde estão instaladas as imagens, não necessariamente é o mesmo ambiente em que o Docker Client está rodando, o que permite alta escalabilidade.

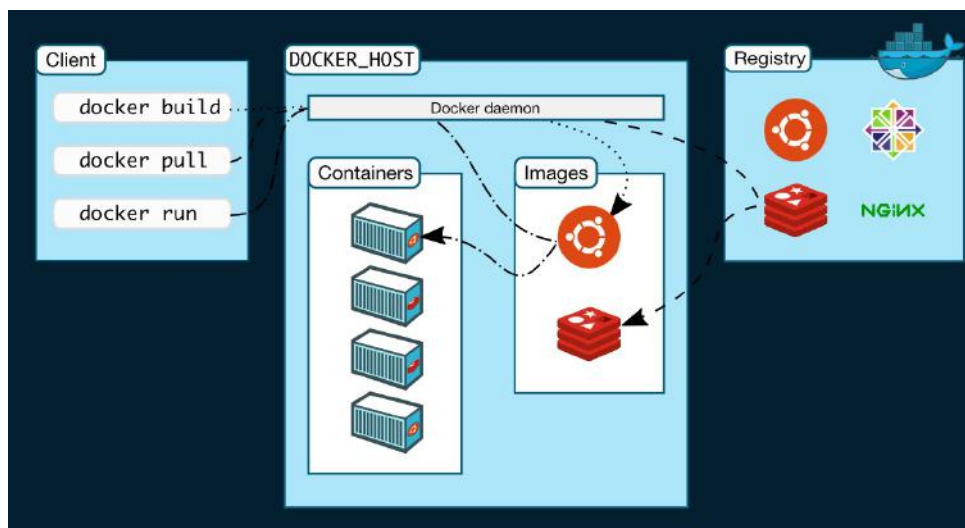


Figura 15 – Resumo do funcionamento da arquitetura do Docker (DOCKER, 2022a)

### 3 ARQUITETURA DO SISTEMA

#### 3.1 INTRODUÇÃO

Neste capítulo é explicado detalhadamente como as ferramentas de Big Data apresentadas na seção anterior foram utilizadas para implantar uma arquitetura distribuída de detecção de ideação suicida dentro da rede social Twitter. São abordados também os módulos e como eles se comunicam. Por fim, um panorama a respeito do seu funcionamento é mostrado. A arquitetura foi pensada de maneira que fosse facilmente escalável de modo que se pudesse utilizar em diversos sistemas operacionais diferentes. Embora, para a produção desse trabalho de conclusão foi utilizado um ambiente Linux, todas as ferramentas utilizadas funcionam de maneira simples e fácil em qualquer outro Sistema Operacional, tanto no Windows quanto no Mac IOS.

### 3.1.1 Desenho da Arquitetura Lógica do sistema

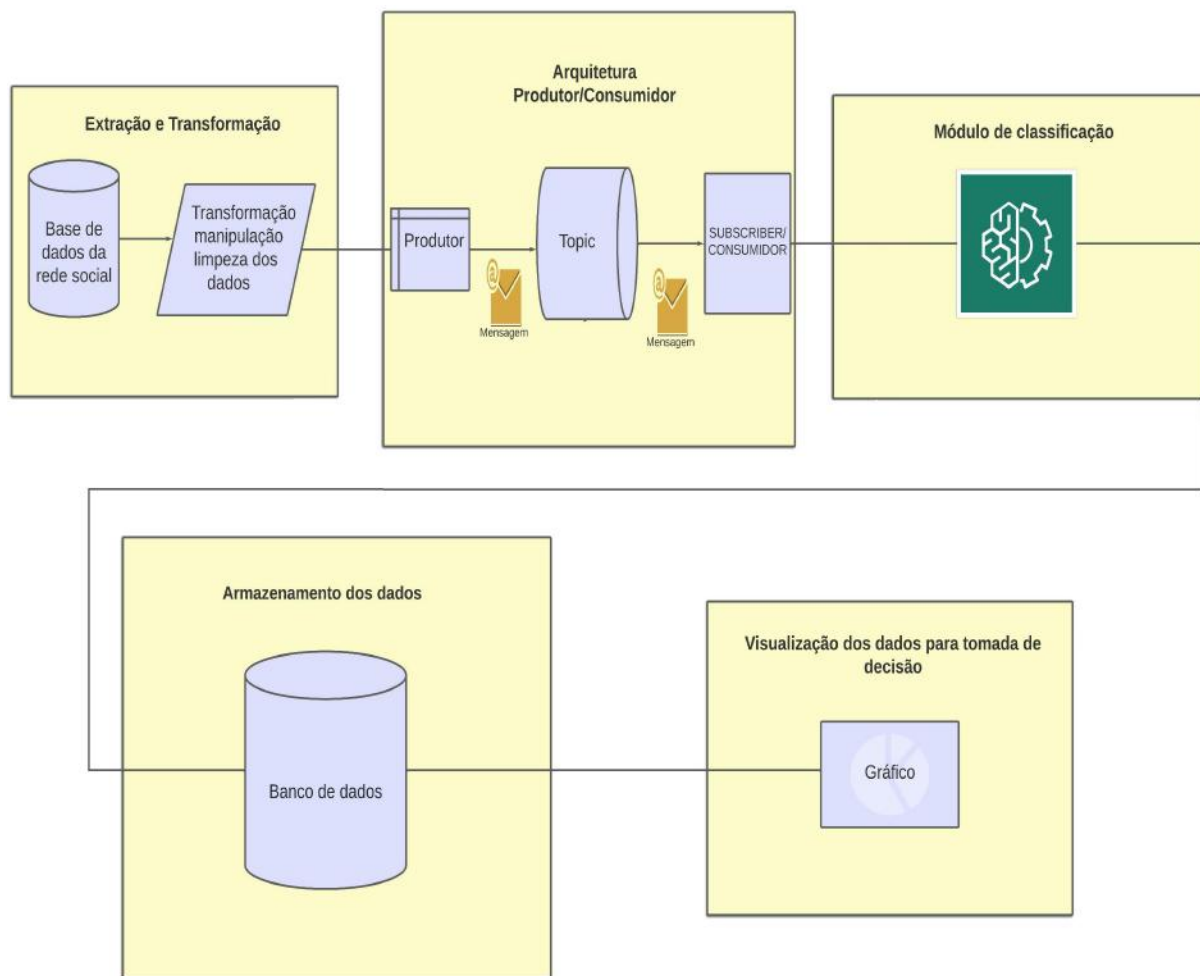


Figura 16 – Arquitetura do Sistema



Figura 17 – Arquitetura do Sistema



### 3.1.2 Coleta de dados

Como há uma dificuldade de encontrar uma base de dados com diversos textos com ideiação suicida escritas em português, foi utilizado uma base de dados com frases suicidas no idioma inglês para treinar o modelo de regressão. Base de dados disponível em (INTENTION, 2021).

A primeira parte da arquitetura é fazer a coleta de dados da rede social do Twitter. Para isso, foi utilizado um script em Python utilizando a biblioteca Tweepy que faz o acesso a API do twitter e coleta as informações de todos os tweets como localização, nome, data e hora da postagem, e as hashtags utilizadas. Com o intuito de pesquisar somente os tweets que possuem indicação de teor suicida, foram selecionadas dentro do dataset de treinamento as palavras mais significativas. Para isso foi utilizado um *Word-Count*(Contador de palavras) dentro do próprio PySpark para retornar essas palavras. São elas:

- "i want to die"
- "i dont want to live anymore"
- "i will kill myself"
- "suicidal"
- "pain"
- "die"
- "i dont care to my life"
- "death"

Com esse conjunto de termos mais relevantes podemos utilizar a API do tweepy pesquisar todos os tweets que possuem tais elementos, para que então seja possível classificar toda a sentença. É possível também fazer a utilização de localizadores geográficos e de data. Após isso, esses tweets são enviados para um tópico Kafka denominado "Análise-de-Twitter". É feito um pequeno tratamento na string como :

1. A retirada de acentos e de pontos gramaticais desnecessários.
2. Remoção de hyperlinks, urls, hashtags e menções, visto que não possuem valor semântico para classificação.

Um exemplo de tweet retirado utilizando a API após o tratamento :

- "see pain pills on the kitchen counter I hate to see it all hurt so bad"

### 3.1.3 Envio dos dados ao tópico Kafka

Com as frases retiradas do Twitter, é feito o envio para um Tópico dentro do Kafka. Para que então essas mensagens sejam consumidas e classificadas. Esse passo está representado na figura abaixo no Fluxo de 1 (Twitter.py) para 2 ( Analise-de-Twitter)

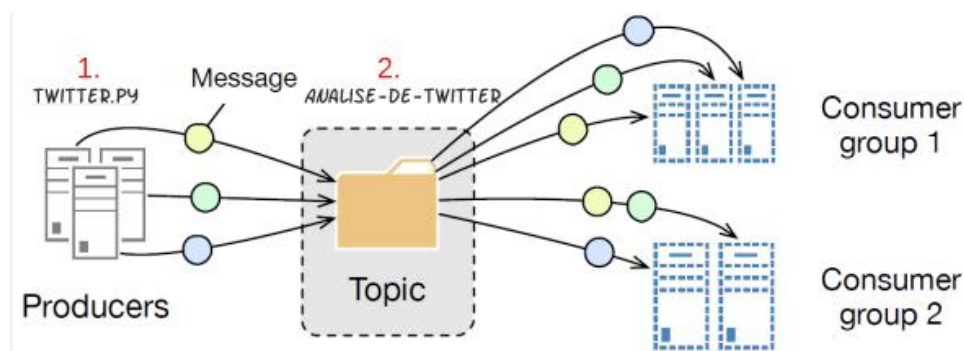


Figura 18 – ADAPTADA DE (NANNONI, 2015)

### 3.1.4 Pipeline de Machine Learning Spark

Um pipeline de Machine Learning é um fluxo completo que combina diversos algoritmos de machine learning em conjunto. Pode haver diversas etapas para processar e aprender com os dados, exigindo então uma sequência de algoritmos. O pipeline utilizado para a construção desse trabalho está representado na figura abaixo:



Figura 19 – Pipeline de Machine Learning

1. Tokenização (Análise Léxica): Consiste na separação e armazenamento de tokens, é nesse processo também que se retira acentos e coloca o texto em caixa baixa se necessário.
2. Stop Word Remover : É utilizado para a remoção de palavras sem valor semântico.
3. HashingTF : Mapeia uma sequência de termos para suas frequências de termo usando o truque de hashing.
4. Regressão Logística : É um modelo de análise de regressão, e é utilizado para prever a probabilidade de uma resposta binária (sim ou não) baseado em uma ou mais variáveis independentes.

O data set utilizado para criar o modelo de pipeline é o a seguir :

```

+-----+-----+
|          tweet|label|
+-----+-----+
|my life is meanin...|    1|
|muttering i wanna...|    1|
|work slave i real...|    1|
|i did something o...|    1|
|i feel like no on...|    1|
|i am great and wo...|    1|
|i ll be dead just...|    1|
|health anxiety pr...|    1|
|everything is oka...|    1|
|ptsd and alcohol ...|    1|
|i dont have long ...|    1|
|i almost attempte...|    1|
|i am 16 and hate ...|    1|
|goodbye everybody...|    1|
|i cant stop fucki...|    1|
|i ve got exactly ...|    1|
|i hate my parents...|    1|
|i cant live in th...|    1|
|why is mankind af...|    1|
|after failing onc...|    1|
+-----+-----+

```

Figura 20 – Exemplificação de uma parte do Data set de treino.

A primeira coluna são os tweets e a segunda coluna é seu valor. 1 se eles forem um tweet com ideiação suicida e 0 para tweet comum.

#### 3.1.4.1 Exemplificação do estágios do pipeline de Machine Learning

A seguir é feito um detalhamento a respeito de casa fase do estágio de pipeline com o intuito de simplificar o entendimento de como ocorre todo o funcionamento

- **Fase de tokenização** : Ao lado esquerdo temos o tweet e ao lado direito temos o tweet tokenizado. É um vetor cuja cada posição é uma palavra

Tweet	Tokenizer
a fear of fatness an idealization of whiteness a quashing of sexuality and a colonization of agency	[, a, fear, of, fatness, an, idealiza- tion, of, whiteness, a, quashing, of, sexuality, and, a, colonization, of, agency]
i bear the wounds of all the battles i avoided can anybody relate to thisquote by fernando pessoa	[, i, bear, the, wounds, of, all, the, battles, i, avoided, can, anybody, relate, to, thisquote, by, fernando, pessoa]
before i die i want to	[, before, i, die, i, want, to]
help aaahhh heelp i dont want to die	[, help, aaahhh, heelp, i, dont, want, to, die]

Tabela 2 – Fase de Tokenizacao.

- **Fase de retirada de palavras** : Nessa fase é feita a retirada de palavras que não possuem valor semântico para o texto.

Tweet	Stop Word Remover
a fear of fatness an idealization of whiteness a quashing of sexuality and a colonization of agency	[, fear, fatness, idealization, whi- teness, quashing, sexuality, colo- nization, agency]
i bear the wounds of all the battles i avoided can anybody relate to thisquote by fernando pessoa	[, bear, wounds, battles, avoided, anybody, relate, thisquote, fer- nando, pessoa]
before i die i want to	[, die, want]
help aaahhh heelp i dont want to die	[, help, aaahhh, heelp, i, dont, want, to, die]

Tabela 3 – Fase de StopWordRemover.

- **Fase de Hashing** : Transforma cada palavra em uma hash e verifica a sua frequên-  
cia na frase.

Tweet	HashingTF
before i die i want to	[262144,[163059,190256,249180],[1 1 1] ]
help aaahhh heelp i dont want to die	[262144 [74245 87273 163059 190256 239859 247272 249180][ 1 1 1 1 1 1]]

Tabela 4 – Fase de Hashing.

- **Fase da Regressão** Calcula a probabilidade e prediz a probabilidade de uma res-  
posta binária.

Tweet	Regressão
a fear of fatness an idealization of whiteness a quashing of sexuality and a colonization of agency	0
before i die i want to	1
had one of those fretful nights hoping today goes well sword of damocles is hanging above me for some reason	0
help aaahhh heelp i dont want to die	1
i bear the wounds of all the battles i avoided can anybody relate to thisquote by fernando pessoa	0

Tabela 5 – Fase de Regressão.

### 3.1.5 Consumo do tópico Kafka

Com o modelo de pipeline já salvo. É carregado então um outro arquivo de inicialização no Spark que irá consumir as mensagens no tópico Kafka e utilizar o modelo de pipeline previamente definido para fazer a classificação das mensagens.

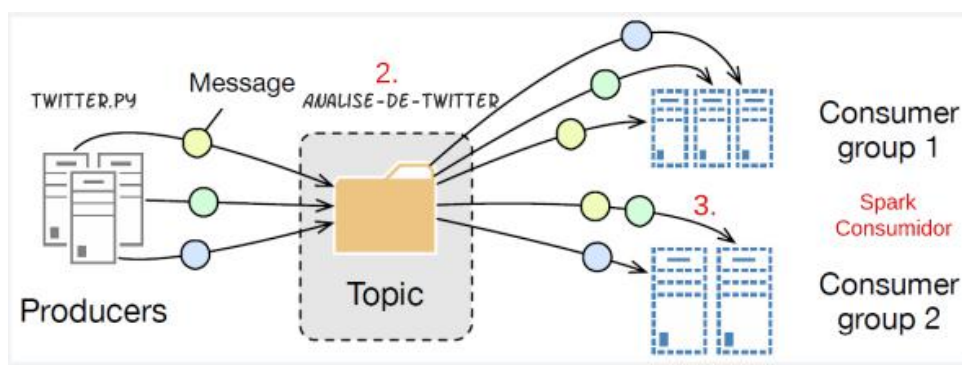


Figura 21 – ADAPTADA DE (NANNONI, 2015)

Essa parte é demonstrada na figura acima no fluxo 2 (Tópico Análise de Twitter) para o fluxo 3 (Spark consumidor)

### 3.1.6 Armazenamento de dados em arquivos Parquet

A partir desses resultados, são salvos em arquivos Parquet para serem analisados posteriormente de maneira simples e eficiente. O formato Parquet permite a utilização de linguagem SQL para retornar algum determinado de dados dentro do ecossistema do Spark com significativas vantagens explicadas em seções anteriores.

## 4 PROVA DE CONCEITO DO FRAMEWORK PROPOSTO

### 4.1 EXECUÇÃO DA FRAMEWORK

O primeiro passo para rodar o modelo se deve a criar um pipeline com o treinamento do dataset. Para isso é inicializado uma sessão no Pyspark e carregado a base de dados em uma variável.

```

+-----+-----+
|          tweet|intention|
+-----+-----+
|my life is meanin...|      1|
|muttering i wanna...|      1|
|work slave i real...|      1|
|i did something o...|      1|
|i feel like no on...|      1|
|i am great and wo...|      1|
|i ll be dead just...|      1|
|health anxiety pr...|      1|
|everything is oka...|      1|
|ptsd and alcohol ...|      1|
|i dont have long ...|      1|
|i almost attempte...|      1|
|i am 16 and hate ...|      1|
|goodbye everybody...|      1|
|i cant stop fucki...|      1|
|i ve got exactly ...|      1|
|i hate my parents...|      1|
|i cant live in th...|      1|
|why is mankind af...|      1|
|after failing onc...|      1|
+-----+-----+
only showing top 20 rows

```

Figura 22 – Data set

Os dados deste dataset são separados em dados de treinamento e dados de teste. De acordo com a tabela a seguir :

Dados	Quantidade
Treinamento	6374
Teste	2745
Total	9119

Tabela 6 – Separação de dados de treinamento e dados de teste.

Após a utilização dos dados de treinamento dentro do pipeline de Machine Learning, é utilizado então os dados de teste. Que são dados nunca antes vistos pelo modelo, com o intuito de verificar se o modelo de aprendizado de máquina está realmente fazendo a previsão dos dados e não somente decorando.

A Acurácia desse modelo é de 0.95 ou seja de 95%. Após criado o pipeline de Machine Learning ( Explicado no capítulo 3.1.4 ) para o recebimento dos dados é feita a inicialização do servidor Kafka.

O Zookeeper é então inicializado. Após a sua inicialização, um servidor Broker é aberto com as configurações locais da máquina.

```
gabriel@gabriel-A320M-S2H:~/Downloads/kafka_2.12-2.8.0/bin$  
./kafka-topics.sh --create --topic Analise-de-Twitter -zooke  
eper localhost:2181 --replication-factor 1 --partitions 1  
Created topic Analise-de-Twitter.
```

Figura 23 – Criação do tópico Analise-de-Twitter

Então um produtor é instanciado para teste, para verificar se eles está enviando a mensagem ao tópico Analise-de-Twitter

```
gabriel@gabriel-A320M-S2H:~/Downloads/kafka_2.12-2  
.8.0/bin$ ./kafka-console-producer.sh --broker-lis  
t localhost:9092 --topic Analise-de-Twitter  
>Teste  
>Produtor  
>Ola TCC
```

Figura 24 – Produtor de dados



```
gabriel@gabriel-A320M-S2H:~/Downloads/kafka_2.12-2.8.0/bin$ ./kafka-console-producer.sh --broker-list localhost:9092 --topic Analise-de-Twitter
>Teste
>Produtor
>Ola TCC
>Estou treinando o Produtor de Dados
>Alo Mundo !
>
```

Figura 25 – Produtor de dados

Após a inicialização de um produtor de mensagens do tópico Analise-de-Twitter, é inicializado então um consumidor para verificar se as mensagens estão sendo enviadas de maneira correta para o tópico Kafka.

```
gabriel@gabriel-A320M-S2H:~/Downloads/kafka_2.12-2.8.0/bin$ ./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic Analise-de-Twitter --from-beginning
Teste
Produtor
Ola TCC
Estou treinando o Produtor de Dados
Alo Mundo !
```

Figura 26 – Consumidor de dados

Com o produtor e o consumidor funcionando corretamente dentro do tópico Kafka. É inicializado então um script no python para fazer a pesquisa dentro da base de dados do Twitter e então enviar esses tweets para o tópico dentro do kafka para que posteriormente essas mensagens sejam consumidas e assim classificadas.

É feita então uma pesquisa dentro da API do twitter, pelas palavras mencionadas no capítulo 3.1.2 Coleta de Dados, esses tweets então são enviados para o tópico Analise-de-Twitter no Kafka. Não foi retornado o usuário que escreveu para a preservação da identidade do mesmo, somente o Twitter e sua localidade. Um exemplo de saída a seguir :

```

i want to die
['i think at this point it s pretty clear that they do indeed and yes actually they still', 'Rio de Janeiro, Brasil']
['I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes', 'Rio de Janeiro']
['cause you re a sky full of stars i want to die in your arms cause you get lighter the more it gets dark i m gonna give you my heart', 'Praia Grande, Brasil']
['I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes', 'Rio de Janeiro']
['I really need 2 say something very important some people want we be 2gether but that s s impossible because innoce', 'SAO PAULO BRASIL']
['Hey guys kind reminder that when I die I allow and expect you to use this fact to get out of work if you want to', 'Sao Bernardo do Campo, Brasil']
['I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes', 'Rio de Janeiro']
['But you can like my life doesn t matter in the literal sense if I die almost nothing in', 'Rio de Janeiro, Brasil']
['I see a rare looking bird out the car window Flying like a sign like he s saying I know This isn t my time to die', 'Rio de Janeiro, Brasil']
['if i lose it all outside the wall live to die another day i dont want anything im just here to', 'Rio de Janeiro']
['Not to be dramatic but I want to die', 'Nova Iguacu, Brasil']
['I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes', 'Rio de Janeiro']
['If I lose it all slip and fall I will never look away If I lose it all lose it all Lose it all If I lose it all', 'Sao Paulo, Brazil']

```

Figura 27 – Script python que retorna os tweets com as palavras mencionadas no capítulo 3.1.2.

```

i will kill myself
['I will kill myself when you least expect it', "J's B's C"]
['tonight i will cry myself to sleep reason im not living the euphoria teenage dream nor any teenage dream at all', 'Sao Jose dos Campos, Brasil']
['Dexter I dont know if I want to see this ep but I need to see how it ends and if dexter die i kill myself I', 'RJ']
live
['Live On Testanto super People e depois Tarkov', 'Rio de Janeiro, Brasil']
['Would you rather sweat mayonnaise or be constantly sexually attracted to fruit What Who said I m not already at', 'Brazil-Sao Paulo']
['5 Imagine how frantic it would be to live with the 80 s version of Axl and Slash', 'Sao Paulo']
['GodOfWar GoW PS4 Gameplay Live Youtube Kratos PS4live God of War live at', 'Lavras, Brasil']
['Ed sheeran Multiple Live in Dublin eu Feliz', 'Sao Paulo, Brasil']
['Live Laugh Love Little racos', 'Sao Paulo, Brasil']
['tmj pela live', 'Sao paulo - Brasil']

```

Figura 28 – Script python que retorna os tweets com as palavras mencionadas no capítulo 3.1.2.

Após o retorno pelo script, é verificado então se essas mensagens estão todas dentro do tópico Kafka, abrindo a aba do consumidor verificamos então quais são as mensagens dentro desse tópico.

```

gabriel@gabriel-A320M-S2H:~/Downloads/kafka_2.12-2.8.0/bin$ ./kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic Analise-de-Twitter --from-beginning
Teste
Produtor
Ola TCC
Estou treinando o Produtor de Dados
Alo Mundo !
["I always delete selfies because some of you lot follow an irl of mine and I d DIE if she found this account I want", "Localizacao25, she/her"]
["i want to die", "Localizacaoacami; il mio chiaro di luna"]
["I just want to play elden ring so I go die already", "Localizacao"]
["Spent 30 minutes trying to convince my hospitalized friend not to work on grades or lesson plans I want us all to be fre", "LocalizacaoIllinois, USA"]
["i m slowly getting to the point where i want to die because i hate school and i do nothing ever i hope i die", "Localizacao 18"]
["And it s my problem if I have no friends and feel I want to die", "Localizacaocreated by @rottwangel"]
["mo ran omg laoshi just hang in there I don t want you to die cwn what do you mean I m going to die do you hate me so m", "Localizacao"]
["Having to wear a mask so you don t give everyone around you covid is not tyranny I know conservat", "Localizacao"]
["I don t want to live the wrong life and then die Speaks to every unhappily partnered person amp to every c", "LocalizacaoNew York, NY"]
["If I was to ever have a sponsorship I d want it to be with I will die on the hill that they are the best", "LocalizacaoLouisville, KY"]
["i think at this point it s pretty clear that they do indeed and yes actually they still", "LocalizacaoRio de Janeiro, Brasil"]
["I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes", "LocalizacaoRio de janeiro"]
["cause you re a sky full of stars i want to die in your arms cause you get lighter the more it gets dark i m gonna give you my heart", "LocalizacaoPraia Grande, Brasil"]
["I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes", "LocalizacaoRio de janeiro"]
["I really need 2 say something very important some people want we be 2gether but that s s impossible because innoce", "LocalizacaoSAO PAULO BRASIL"]
["Hey guys kind reminder that when I die I allow and expect you to use this fact to get out of work if you want to", "LocalizacaoSao Bernardo do Campo, Brasil"]
["I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes", "LocalizacaoRio de janeiro"]
["But you can like my life doesn t matter in the literal sense if I die almost nothing in", "LocalizacaoRio de Janeiro, Brasil"]
["I see a rare looking bird out the car window Flying like a sign like he s saying I know This isn t my time to die", "LocalizacaoRio de Janeiro, Brasil"]
["if i lose it all outside the wall live to die another day i dont want anything im just here to", "LocalizacaoRio de Janeiro"]
["Not to be dramatic but I want to die", "LocalizacaoNova Iguacu, Brasil"]
["I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes", "LocalizacaoRio de janeiro"]
["If I lose it all slip and fall I will never look away If I lose it all lose it all lose it all", "LocalizacaoSao Paulo, Brazil"]
["wanna see nirvana but i dont want to die yet", "LocalizacaoGuaruja, Brasil"]
["I want to die without making everyone sad", "Localizacaoar araruma "]
["I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes", "LocalizacaoRio de janeiro"]
["Live to die another day I don t want anything", "LocalizacaoOsasco, Brasil"]
["I said kill me now I want to die I heard there s a chance at an afterlife I might not get let in but at least I won t be living", "LocalizacaoSao paulo, Brasil"]
["i think at this point it s pretty clear that they do indeed and yes actually they still", "Rio de Janeiro, Brasil"]
["i think at this point it s pretty clear that they do indeed and yes actually they still", "Rio de Janeiro, Brasil"]
["I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes", "Rio de janeiro"]
["cause you re a sky full of stars i want to die in your arms cause you get lighter the more it gets dark i m gonna give you my heart", "Praia Grande, Brasil"]
["I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes", "Rio de janeiro"]
["I really need 2 say something very important some people want we be 2gether but that s s impossible because innoce", "SAO PAULO BRASIL"]
["Hey guys kind reminder that when I die I allow and expect you to use this fact to get out of work if you want to", "Sao Bernardo do Campo, Brasil"]
["I m the one that s got to die when it s time for me to die so let me live my life the way I want to LifeQuotes", "Rio de janeiro"]

```

Figura 29 – Consumidor do tópico Kafka Analise-de-Twitter

Com todas as mensagens então sendo carregadas no tópico Kafka. É então inicializado o Spark Streaming que lê e classifica essas mensagens em tempo de execução, salvando essas mensagens posteriormente dentro do arquivo parquet.

Abaixo segue alguns exemplos de execução do Spark Streaming. Onde a primeira coluna é o Twitter coletado, a segunda coluna a Localização desse tweets e a terceira coluna é a predição do twitter. Se possui ideiação suicida então é classificado sempre como 1.

```

Batch: 38
-----
+-----+-----+-----+
|tweet|Localizacao|prediction|
+-----+-----+-----+
|["some days i want to live and some days i want to die I tried I feel trapped in my own mind I feel it s time", "|Sao Paulo, Brasil"]|1.0|
+-----+-----+-----+

```

Figura 30 – Spark Streaming classificando Tweets : Lote 38

```

Batch: 42
-----
+-----+-----+-----+
|tweet                                     |Localizacao|prediction|
+-----+-----+-----+
|["Fuck this shit I really want to die But I can t kill myself because there s someone I don t want to see cry", "|Mogi Mirim, Brasil"]|1.0|
+-----+-----+-----+

```

Figura 31 – Spark Streaming classificando Tweets : Lote 42

```

+-----+-----+-----+
|tweet                                     |Localizacao|prediction|
+-----+-----+-----+
|["My weeks have been so exhausting It s monday and I already want to die cries Burnout is real I don t even know", "|Sao Paulo"]|1.0|
+-----+-----+-----+

```

Figura 32 – Spark Streaming classificando Tweets: Lote 55

```

Batch: 305
-----
+-----+-----+-----+
|tweet                                     |Localizacao|prediction|
+-----+-----+-----+
|["My home is the same It s empty sad broken brings back the worst memories of my family The good thi", "|Rio de Janeiro, Brasil"]|1.0|
+-----+-----+-----+

```

Figura 33 – Spark Streaming classificando Tweets : Lote 305

```

Batch: 470
-----
+-----+-----+-----+
|tweet                                     |Localizacao|prediction|
+-----+-----+-----+
|["You were my everything and all that you did was make me fucking sad", "|Rio de Janeiro, Brasil"]|1.0|
+-----+-----+-----+

```

Figura 34 – Spark Streaming classificando Tweets : Lote 470

```

-----
Batch: 550
-----
+-----+-----+-----+
|tweet                                     |Localizacao   |prediction|
+-----+-----+-----+
|["JUST KILL ME ONE LAST TIME WHILE YOU SAY THAT LOVE ME DON T WAST ANOTHER FUCKING LIFE I STILL WANT YOU BY MY SIDE", "|Guaruja, Brasil"]|1.0   |
+-----+-----+-----+

```

Figura 35 – Spark Streaming classificando Tweets : Lote 550

```

-----
Batch: 666
-----
+-----+-----+-----+
|tweet                                     |Localizacao   |prediction|
+-----+-----+-----+
|["Gimme a fucking gun and some time and you will see what happens to you", "|Sao Paulo, Brasil"]|1.0   |
+-----+-----+-----+

```

Figura 36 – Spark Streaming classificando Tweets : Lote 666

```

-----
Batch: 924
-----
+-----+-----+-----+
|tweet                                     |Localizacao   |prediction|
+-----+-----+-----+
|["some days i want to live and some days i want to die I tried I feel trapped in my own mind I feel it s time", "|Sao Paulo, Brasil"]|1.0   |
+-----+-----+-----+

```

Figura 37 – Spark Streaming classificando Tweets : Lote 924

```

-----
Batch: 1637
-----
+-----+-----+-----+
|tweet                                     |Localizacao   |prediction|
+-----+-----+-----+
|["some days i want to live and some days i want to die I tried I feel trapped in my own mind I feel it s time", "|Sao Paulo, Brasil"]|1.0   |
+-----+-----+-----+

```

Figura 38 – Spark Streaming classificando Tweets : Lote 1637

```

-----
Batch: 1641
-----
+-----+-----+-----+
|tweet|                                     |Localizacao| |prediction|
+-----+-----+-----+
|["Fuck this shit I really want to die But I can t kill myself because there s someone I don t want to see cry", "Mogi Mirim, Brasil"]|1.0|
+-----+-----+-----+

```

Figura 39 – Spark Streaming classificando Tweets : Lote 1641

Após essa análise, esses Tweets são salvos em Formato Parquet para uma análise posterior desses dados. Segue abaixo um dos arquivos parquet gerados, e a visualização desses dados.

```

>>> df = spark.read.parquet("/home/gabriel/Downloads/spark/part-00000-93429e5b-2107-459e-aa85-aaf0c07a9e1e-c000.snappy.parquet")
>>> df.show()
+-----+-----+-----+
|          tweet|          Localizacao|prediction|
+-----+-----+-----+
|["I ve just tried...| Sao Paulo, Brazil"|          1.0|
|["FireMonkey Sadl...| Sao Paulo, Brasil"|          1.0|
|["tkotz This site...| Sao Paulo, Brasil"|          1.0|
|["All of it I alr...| Sao Paulo, Brazil"|          1.0|
|["Also I m relati...| Rio de Janeiro, B...|          1.0|
|["I tried to open...| Rio de Janeiro - ...|          1.0|
|["You can t deny ...| Sacoma, Sao Paulo"|          1.0|
|["So many times I...| Guarulhos, Brasil"|          1.0|
|["Got too many de...| Sao Paulo, Brasil"|          1.0|
|["dude I tried to...| Rio de Janeiro, B...|          1.0|
|["You can to know...| Rio de Janeiro"|          1.0|
|["Everyone s at F...| Tubingen - Germany"|          1.0|
|["you destroyed m...| Sao Caetano do Su...|          1.0|
|["Update after 3 ...| Sao Paulo, SP"|          1.0|
|["oh I m a wreck ...| Niteroi, RJ"|          1.0|
|["remember when d...| Sao Paulo"|          1.0|
|["I reported and ...| Sao Paulo, Brazil"|          1.0|
|["alienshe It wil...| Sao Paulo, Brasil"|          1.0|
|["The best part b...| Rio de Janeiro"|          1.0|
|["you ve decided ...| Santa Catarina, B...|          1.0|
+-----+-----+-----+
only showing top 20 rows

```

Figura 40 – Dados armazenados na extensão parquet

Observação a respeito do estudo : Ele foi executado no dia 15/01/2022 às 5 horas da tarde em uma máquina com ram: 16gb, SO: Linux, CPU: Ryzen 5 2600.

O repositório do código pode ser encontrado em (TOMAZELLI; LUNA; PAIVA, 2021), para a replicação de qualquer experimento ou alteração no código, o usuário deve clonar o projeto ou baixá-lo. Na página inicial há um breve manual de como fazer a inicialização de cada instância da framework, junto com um tutorial no Youtube descrevendo seu passo a passo de cada ferramenta.

## 4.2 VISUALIZAÇÃO DOS DADOS

Para facilitar a visualização dos dados e a tomada de decisão por parte dos usuários. Foi disponibilizado um site contendo algumas informações a respeito dos dados coletados e classificados pela framework. Nesse site há diversas abas para a visualização de determinadas informações coletadas. Os itens estão listados abaixo:

- Quantidade de Tweets;
- Quantidade de Tweets com ideação suicida;
- Quantidade por dia dos Tweets com ideação suicida;
- Rank de Países/Cidades analisadas que possuem a maior quantidade de tweets com ideação suicida;
- Últimos tweets classificados pela framework que possuem ideação suicida;
- Um mapa com atualização de quantidade dos Tweets que possuem ideação suicida.

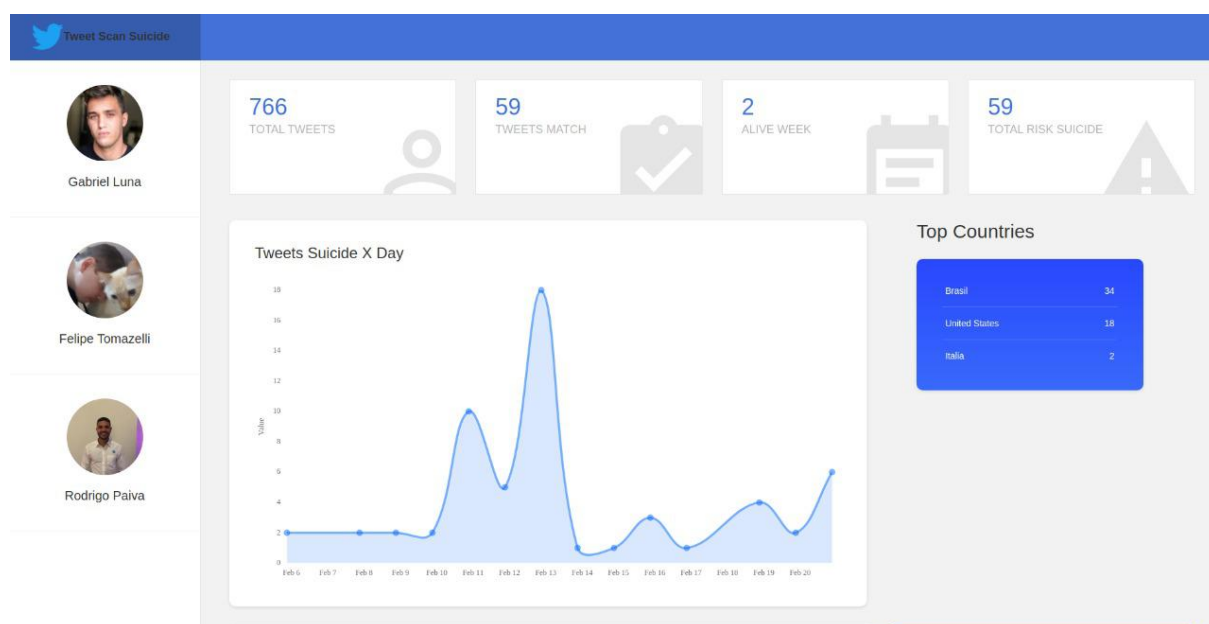


Figura 41 – foto ilustrativa do site [www.twitterscan.com.br](http://www.twitterscan.com.br)

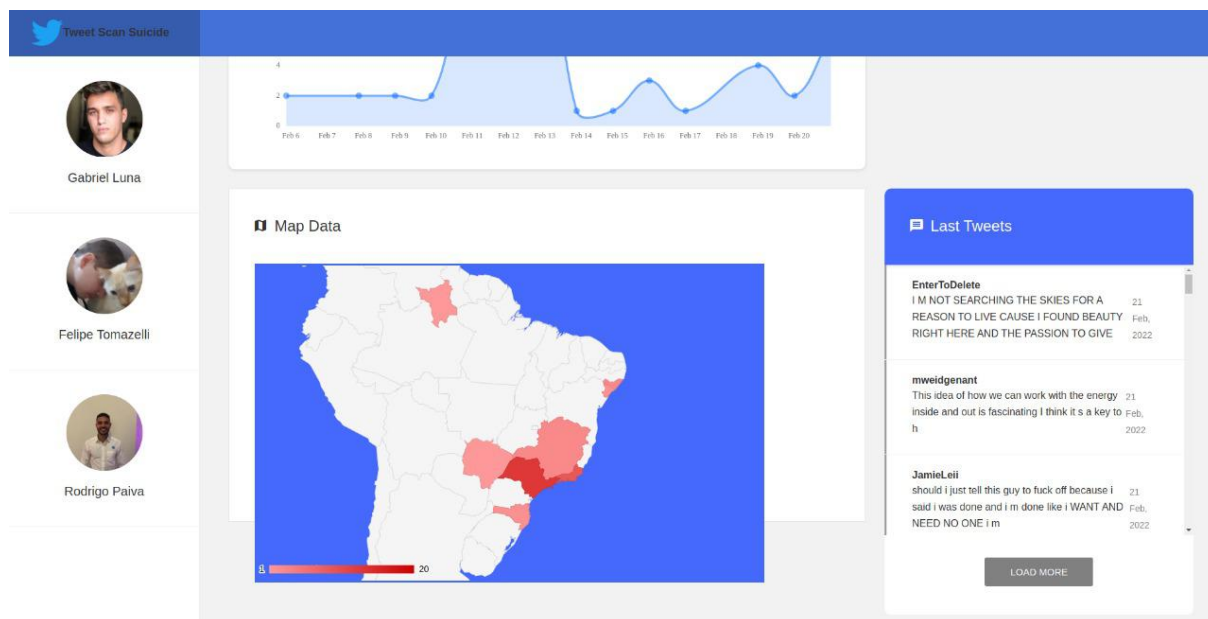


Figura 42 – foto ilustrativa do site [www.twitterscan.com.br](http://www.twitterscan.com.br)

#### 4.3 EXECUÇÃO DA FERRAMENTA VIRTUALIZADA NO SERVIÇO DE NUVEM DA AWS

Por conta da limitação do uso de máquina pessoal para a realização desse trabalho, toda a framework foi modificada para o uso de virtualização dos componentes apresentados. Para isso, foi utilizado o Docker para criar microsserviços de cada estrutura da ferramenta e assim disponibilizá-la em um serviço de nuvem. Com a finalidade de deixar o serviço da framework disponível por mais tempo.



### 4.3.1 Arquitetura da framework em nuvem

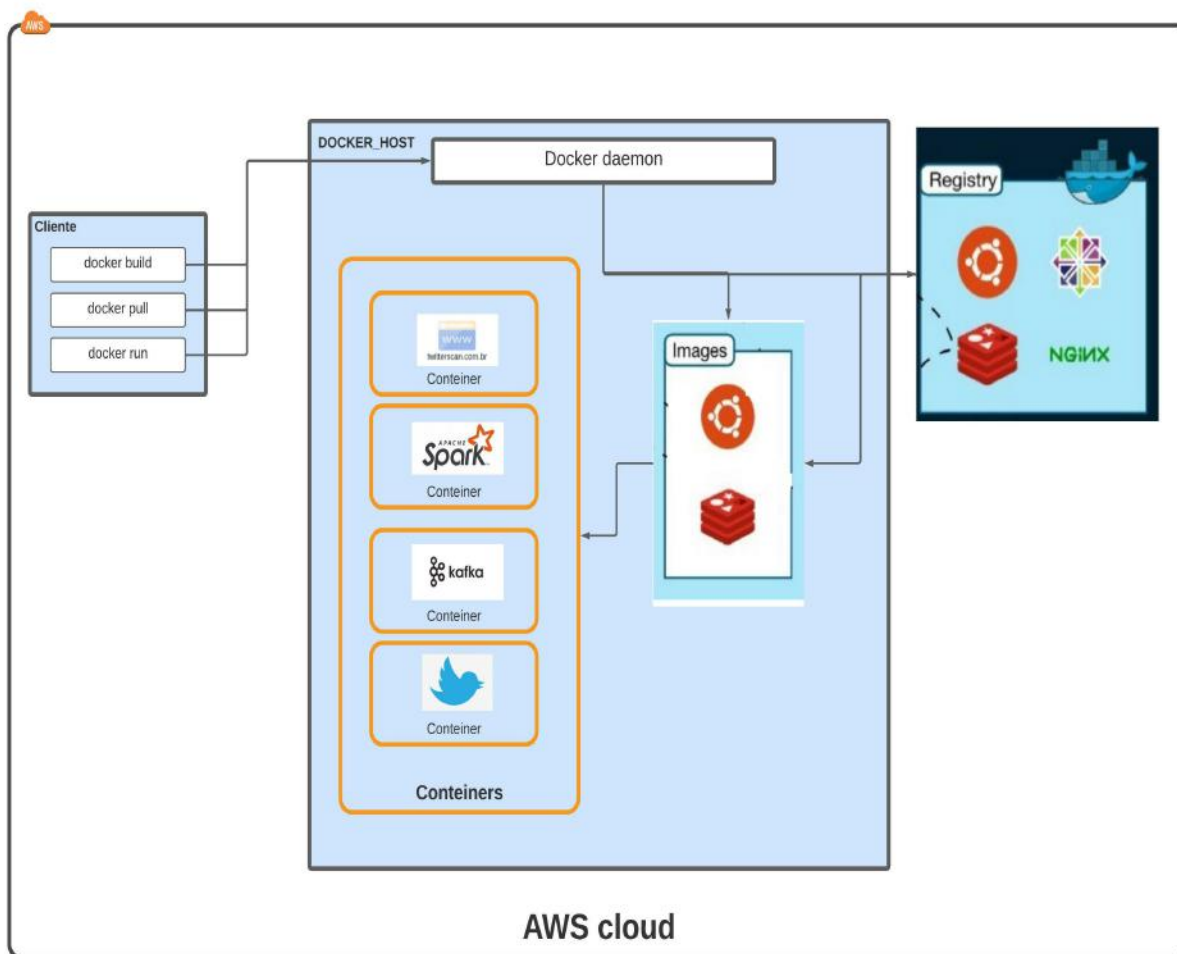


Figura 43 – Arquitetura do ambiente virtualizado na nuvem AWS

A figura acima representa a virtualização da framework por meio do docker, que tem seu funcionamento explicado no capítulo 2.7. O repositório a seguir contém todos os comandos para o funcionamento da framework no ambiente de virtualização (TOMAZELLI; LUNA; PAIVA, 2022).

## 5 LIMITAÇÕES ENCONTRADAS

Infelizmente pela falta de disponibilização de uma base de dados de possíveis frases com sentimentos suicidas no idioma em português, limitou o trabalho a pesquisar tweets somente no idioma inglês. A tradução dessa base não foi possível pelo fato do texto do twitter conter muitas gírias no idioma português, fazendo com que a taxa de acerto do algoritmo caísse bruscamente. É necessário se reunir junto com algum especialista da área da saúde mental para coletar alguns tweets e classificá-los, com o intuito de criar uma base robusta com as determinadas palavras que indicam se aquele texto possui uma ideia suicida ou não. Uma outra limitação foi o pagamento do serviço em nuvem, por necessitar de uma máquina robusta e disponível o tempo todo, a utilização da AWS foi essencial para a finalização do trabalho, infelizmente por conta do custo não foi possível realizar uma grande coleta desses Tweets, sendo limitado a poucos dias somente.

## 6 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho de conclusão se desenvolveu um framework com as melhores ferramentas de Big Data do mercado, que tem como objetivo analisar as redes sociais os sentimentos de seus usuários, utilizando algumas técnicas de *Machine Learning* para a classificação dessas mensagens.

No capítulo 2 revisa-se, sistematicamente, todo o conteúdo relacionado às ferramentas utilizadas, com o intuito de descrever e dar ao leitor uma facilidade de entendimento de todo o trabalho apresentado. Foram abordados uma introdução inicial a respeito do tema Big Data e como está seu desenvolvimento nos dias atuais. Uma abordagem das ferramentas e suas características como: Apache Kafka, Algoritmos de Aprendizados de máquina, Apache Spark e o Apache Parquet, Computação em Nuvem e Docker.

O Capítulo 3 descreve toda a arquitetura do sistema e como está sendo o seu funcionamento passo a passo. Essas características permitem que futuros desenvolvedores e interessados no tema possam adicionar funcionalidades ou melhorias, sem a necessidade de grandes alterações. Pelo fato de se utilizar ferramentas que são altamente escaláveis e distribuídas, trás uma dinamicidade maior para a utilização da ferramenta.

O Capítulo 4 detalha um exemplo de execução de toda a ferramenta, desde a configuração inicial dos servidores e apaches, até a saída das mensagens classificadas. Neste capítulo também é detalhado o tipo de armazenamento dessas mensagens, para caso necessário utilizar em trabalhos futuros que tragam uma análise aprofundada desses textos e regiões. Detalha também a acurácia do modelo de 95% em dados não vistos anteriormente. Além de demonstrar uma versão virtualizada da framework e a disponibilização de um web site [www.twitterscan.com.br](http://www.twitterscan.com.br) para a análise dos tweets.

A grande contribuição deste trabalho é o desenvolvimento de uma ferramenta capaz de ser altamente distribuída e escalável, que possa ficar fazendo análises de diversas redes sociais ao mesmo tempo.

Para o desenvolvimento dessa ferramenta, foi utilizada a linguagem Python com o objetivo de pesquisar dentro da base de dados do twitter utilizando sua Api Tweepy, esse script contém aproximadamente 1000 linhas, pois além de fazer essa busca, ele faz umas melhorias na string retornada e alguns refinamentos. Foi utilizado também um programa para rodar no Spark com o intuito de criar o pipeline de dados, esse programa possui aproximadamente 700 linhas de código. Fora toda a configuração dos Apaches utilizados, como o Kafka, o Parquet e o Spark, que juntos fizeram ser possível essa ferramenta de análise de Big Data.

Para trabalhos futuros podem ser agregados novas funcionalidades no web site [twitterscan.com.br](http://twitterscan.com.br) para tornar possível uma análise profunda a respeito dos tweets em relação às suas regiões, para que se possa fazer uma intervenção por meio de propagandas ou até

mesmo o contato com essas pessoas, para que um efeito *Papageno* (BLATT, 2019) seja mais relevante dentro da sociedade. É possível também utilizar novas ferramentas de *machine learning* para se obter um resultado melhor na classificação de mensagens como a utilização de redes convolucionais e *embeddings words*.

## REFERÊNCIAS

- AMAZON.COM, I. **What is Pub/Sub Messaging?** 2021. Disponível em: <https://aws.amazon.com/pt/pub-sub-messaging/>.
- APACHE, S. F. T. **Apache Kafka: A distributed streaming platform.** 2021. Disponível em: <https://kafka.apache.org/documentation/>.
- APACHE, S. F. T. **Apache Zookeeper.** 2021. Disponível em: <https://zookeeper.apache.org/doc/>.
- BLATT, M. R. A relevância das redes sociais na prevenção ao suicídio. **Revista da Saúde da AJES**, v. 5, n. 10, 2019.
- CONFLUENT. **Confluent: Data in Motion.** 2021. Disponível em: <https://www.confluent.io>.
- CORPORATION, L. **LinkedIn Corporation.** 2021. Disponível em: <https://www.linkedin.com>.
- CRAMER, J. S. **Logit models from economics and other fields.** Cambridge, UK New York: Cambridge University Press, 2003. ISBN 9780521815888.
- DATABRICKS. **The Data and AI Company.** 2021. Disponível em: <https://databricks.com/glossary/what-is-parquet>.
- DOCKER, I. **Docker architecture.** 2022. Disponível em: <https://docs.docker.com/engine/docker-overview/#docker-architecture>.
- DOCKER, I. **What is a Container?** 2022. Disponível em: <https://www.docker.com/resources/what-container>.
- FACELI, K. et al. **Inteligência artificial: uma abordagem de aprendizado de máquina.** [S.l.]: LTC - GRUPO GEN, 2011. ISBN 9788521618805.
- FAHEY, R. A.; MATSUBAYASHI, T.; UEDA, M. Tracking the werther effect on social media: Emotional responses to prominent suicide deaths on twitter and subsequent increases in suicide. **Social Science & Medicine**, Elsevier, v. 219, p. 19–29, 2018.
- FORTUNE. **Fortune.** 2021. Disponível em: <https://fortune.com/about-us/>.
- GARCÍA-GIL, D. et al. A comparison on scalability for batch big data processing on apache spark and apache flink. **Big Data Analytics**, BioMed Central, v. 2, n. 1, p. 1–11, 2017.
- GARTNER, I. **Definition of Big Data.** 2021. Disponível em: <https://www.gartner.com/en/information-technology/glossary/big-data>.
- GOLDEN, B.; SCHEFFY, C. **Virtualization for dummies.** [S.l.]: Wiley, 2008.
- HEY, T.; TANSLEY, S.; TOLLE, K. **The Fourth Paradigm: Data-Intensive Scientific Discovery.** [S.l.]: Microsoft Research, 2009. ISBN 9780982544204.

- HOSMER, D. W.; LEMESBOW, S. Goodness of fit tests for the multiple logistic regression model. **Communications in statistics-Theory and Methods**, Taylor & Francis, v. 9, n. 10, p. 1043–1069, 1980.
- INOUBLI, W. et al. An experimental survey on big data frameworks. **Future Generation Computer Systems**, Elsevier, v. 86, p. 546–564, 2018.
- INTENTION, T. S. **suicidio**. 2021. Disponível em: <https://github.com/laxmimerit/twitter-suicidal-intention-dataset>.
- JOHANSSON, L. **Apache Kafka for beginners**. 2021. Disponível em: <https://www.cloudkarafka.com/blog/2016-11-30-part1-kafka-for-beginners-what-is-apachekafka.html>.
- LANEY, D. 3D data management: Controlling data volume, velocity, and variety. **META Group**, v. 6, n. 70, 2001.
- LUGER, G. F. **Inteligência Artificial: Estruturas e estratégias para a solução de problemas complexos**. [S.l.]: Bookman, 2004. ISBN 9788536303963.
- LYMAN, P.; VARIAN, H. R. How much information. p. 100, 2003. Disponível em: <http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/index.htm>.
- MCKINSEY, G. I. **MGI<sub>bigdata</sub>execssummary.pdf**. 2011. *Disponvelem* : .
- MELNIK, S. et al. Dremel: Interactive analysis of web-scale datasets. In: **Proc. of the 36th Int'l Conf on Very Large Data Bases**. [s.n.], 2010. p. 330–339. Disponível em: <http://www.vldb2010.org/accept.htm>.
- NANNONI, N. **Message-oriented Middleware for Scalable Data Analytics Architectures**, Programa de Mestrado em sistemas de comunicação, KTH Escola de Tecnologia da Informação e Comunicação, Estocolmo. 2015. Disponível em: <https://www.diva-portal.org/smash/get/diva2:813137/FULLTEXT01.pdf>.
- NAUR, P. **Concise survey of computer methods**. [S.l.]: Petrocelli Books, 1974. ISBN 9780884053149.
- NICULESCU, V. On the impact of high performance computing in big data analytics for medicine. **Applied Medical Informatics.**, v. 42, n. 1, p. 9–18, Mar. 2020. Disponível em: <https://ami.info.umfcluj.ro/index.php/AMI/article/view/766>.
- OLIVEIRA, A. M. de. A webvertising ambiental: A publicidade na internet como ferramenta de conscientização e democratização dos problemas e soluções aplicáveis ao meio ambiente. 2003.
- OPAS/OMS, S. **suicidio**. 2021. Disponível em: <https://www.paho.org/pt/topicos/suicidio>.
- ORACLE, C. **O que é Big Data?** 2021. Disponível em: <https://www.oracle.com/br/big-data/what-is-big-data/>.

OXFORD, U. P. **big-data noun - Definition, pictures, pronunciation and usage notes**. 2021. Disponível em: <https://www.oxfordlearnersdictionaries.com/us/definition/english/big-data?q=big+data>.

PARK, H. W.; LEYDESDORFF, L. Decomposing social and semantic networks in emerging “big data” research. **Journal of Informetrics**, Elsevier, v. 7, n. 3, p. 756–765, 2013.

PARQUET. **Apache parquet**. 2013. Disponível em: <https://parquet.apache.org/documentation/latest/>.

RAO, J.; CONFLUENT. **Kafka Summit SF 2019 Keynote ft. Chris Kasten, Walmart Labs**. 2021. Disponível em: <https://fortune.com/about-us/>.

ROUSSEAU, R. A view on big data and its relation to informetrics. **Chinese Journal of Library and Information Science**, CJLIS, v. 5, n. 3, p. 12–26, 2012.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. [S.l.]: Prentice Hall Press, 2009. ISBN 9780136042594.

RUSTY. **Suicide Rates Overview 1985 to 2016**. 2018. Disponível em: <https://www.kaggle.com/russellyates88/suicide-rates-overview-1985-to-2016>.

SILBERSCHATZ, A.; BAER, P.; GAGNE, G. **Fundamentos de sistemas operativos**. [S.l.]: Mc Graw-Hill, 2015.

SPARK, A. **GraphX Programming Guide**. 2021. Disponível em: (<https://spark.apache.org/docs/latest/graphx-programming-guide.html>).

TOMAZELLI, F.; LUNA, G.; PAIVA, R. **Tomazelli/Spark**. 2021. Disponível em: <https://github.com/Tomazelli/Spark>.

TOMAZELLI, F.; LUNA, G.; PAIVA, R. **Docker/twitterscan**. 2022. Disponível em: <https://github.com/GabrielSLuna/Spark/tree/django-docker>.

VERAS, M. **Virtualização: tecnologia central do Datacenter Capa comum**. [S.l.]: Brasport Livros e Multimídia Ltda, 2011.

YU, Y. Os-level virtualization and its applications. 2007. Disponível em: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.76.4527&rep=rep1&type=pdf>.

ZAHARIA, M. et al. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: **9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)**. [S.l.: s.n.], 2012. p. 15–28.

ZWOLENSKI, M.; WEATHERILL, L. The digital universe: Rich data and the increasing value of the internet of things. **Journal of Telecommunications and the Digital Economy**, v. 2, n. 3, p. 47–1, 2014.