

# Eleições 2018

Ciência da Computação  
UCS

Maurício Mussatto Scopel



# Hipóteses

Qual dos candidatos à presidência para o segundo turno da eleição brasileira do ano de 2018 possui mais popularidade?

Qual dos candidatos à presidência para o segundo turno da eleição brasileira do ano de 2018 representam um sentimento mais positivo?

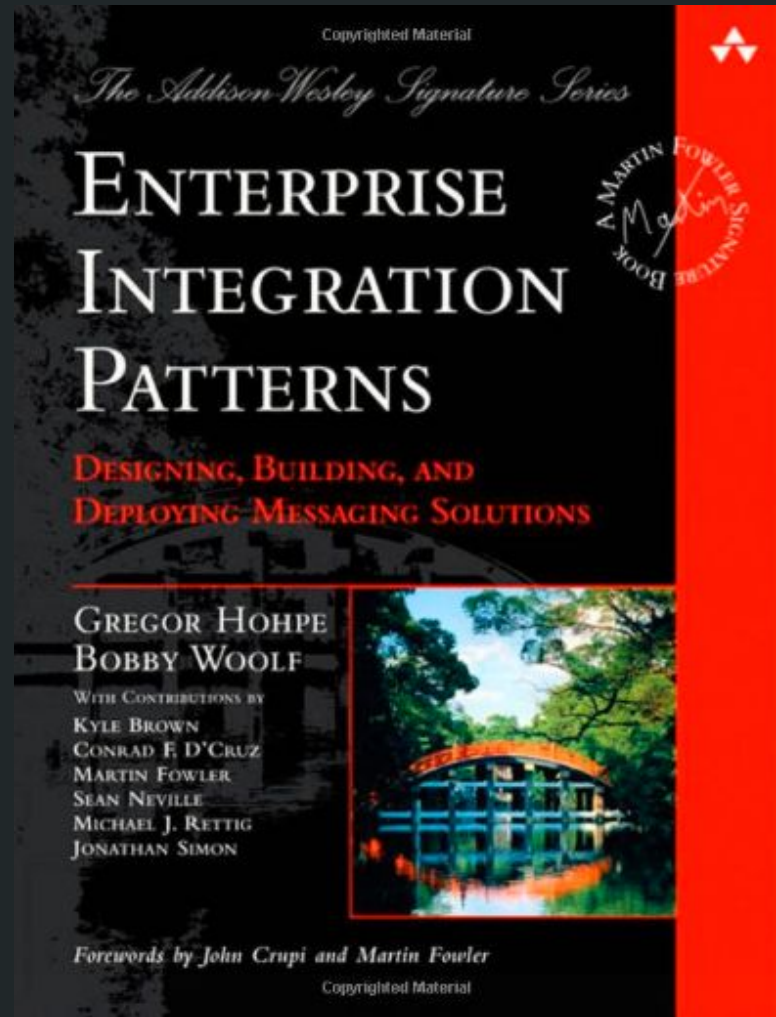
# Construção da base de dados

Pontos a considerar:

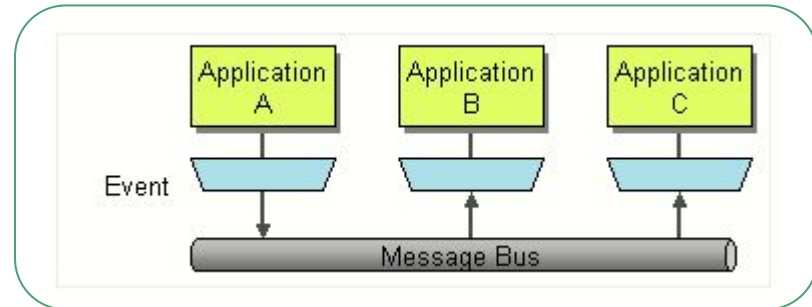
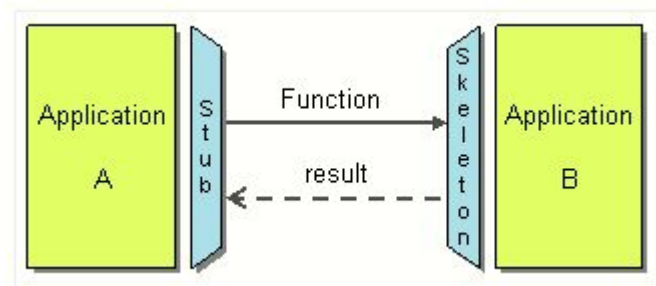
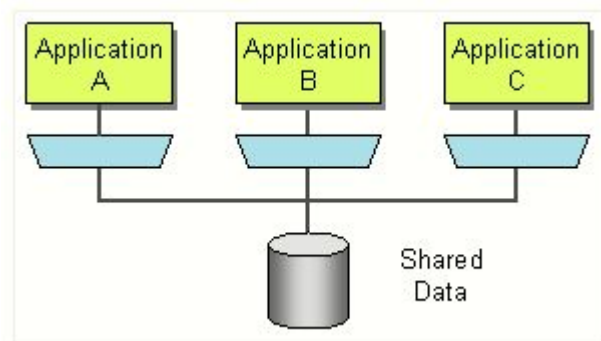
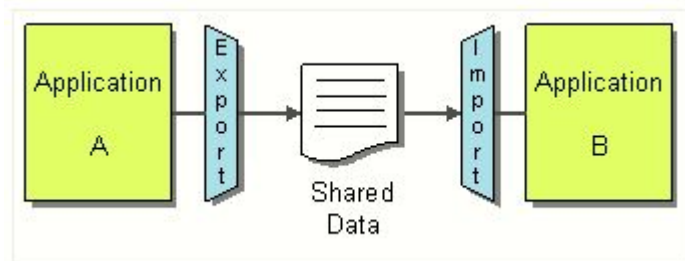
- Comunicação de duas fontes de dados distintas
  - API do Twitter
  - Banco de dados NOSQL

# Padrões de Integração

- Diferentes linguagens
- Diferentes tecnologias
- Diferentes formatos
- Diferentes protocolos
- Soluções similares

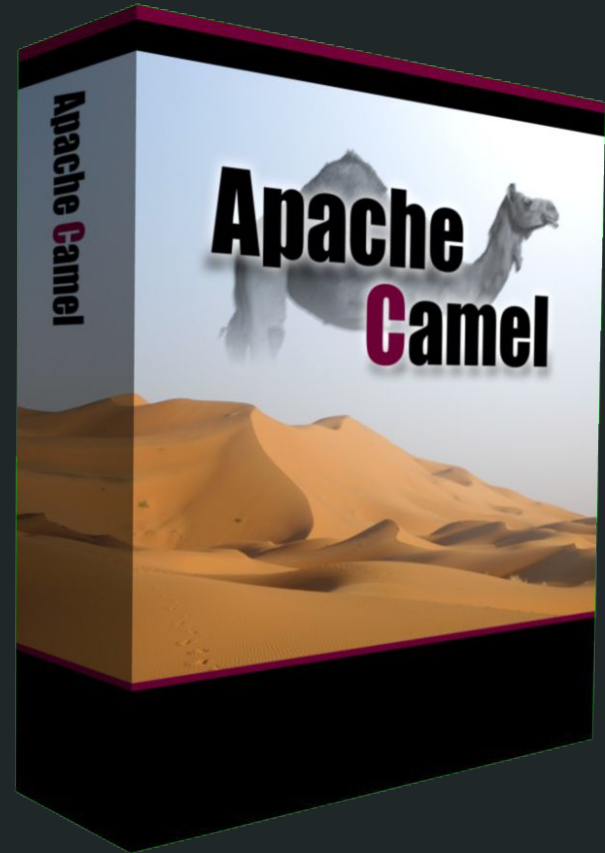


# Estilos de integração

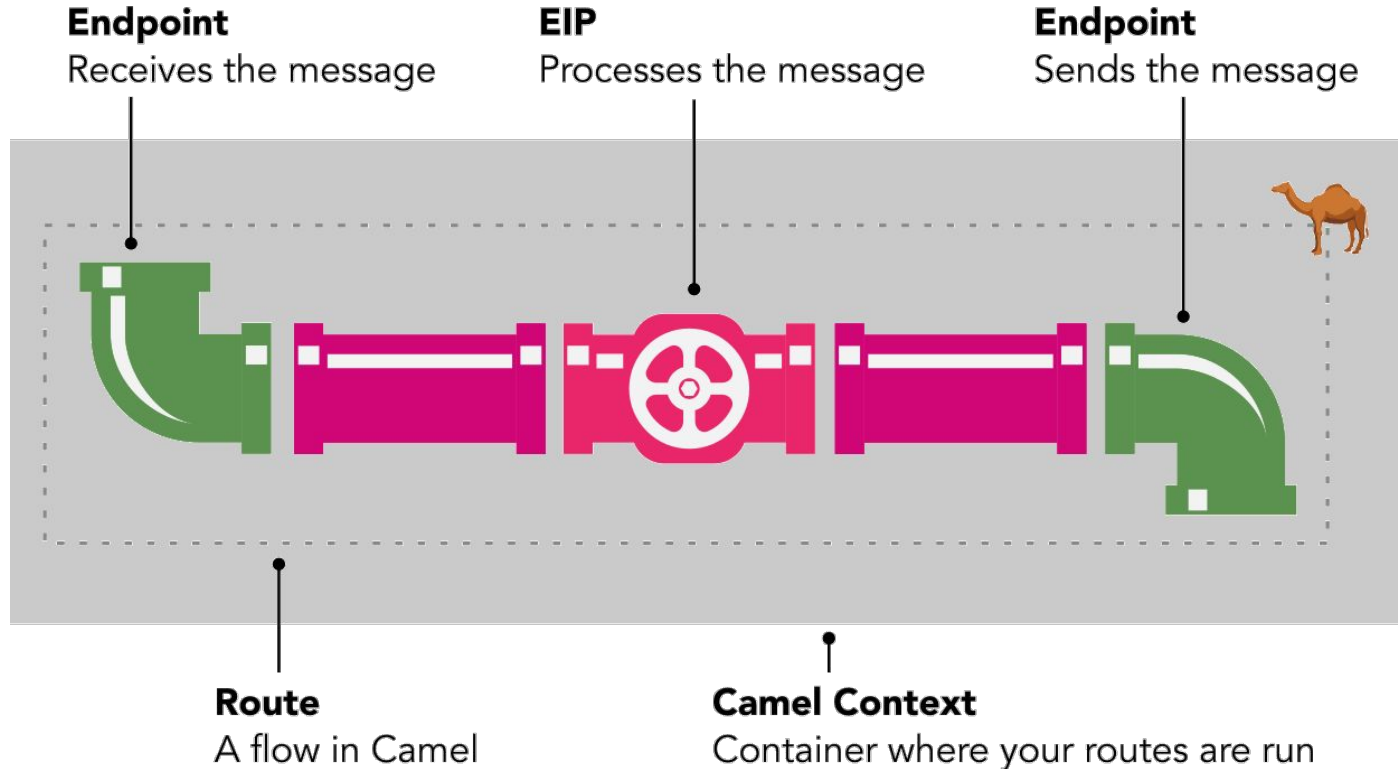


# Apache Camel

- Framework de integração
- DSL para inúmeras linguagens
- Inúmeros componentes
- Criação de componentes
- OpenSource
  - <https://github.com/apache/camel> □



# Como o Apache Camel funciona?



# Polling

```
@Override
public void configure() throws Exception {
    configureDelayWhenRateLimitExceeded();

    fromF( uri: "twitter-search://%s?" +
            "type=polling&delay=10&extendedMode=true&bridgeErrorHandler=true&count=450" +
            "&consumerKey=%s&consumerSecret=%s&accessToken=%s&accessTokenSecret=%s",
        keywords, twitterConfig.getKey(),
        twitterConfig.getSecret(), twitterConfig.getAccessToken(),
        twitterConfig.getAccessTokenSecret() )
    .routeId("twitter-search-route")
    .process(exchange -> {
        LOGGER.info("Reading tweet {}", tweetRead());
    })
    .toF( uri: "mongodb3:mongoBean?database=%s&collection=%s&operation=insert",
        mongoDbConfig.getDbName(),
        mongoDbConfig.getCollectionName() );
}
```



# Varredura desde uma data

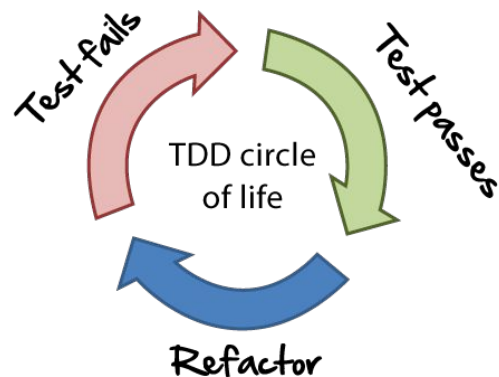
```
@Override
public void configure() throws Exception {
    configureDelayWhenRateLimitExceeded();
    from( uri: "timer://timerSearch?period=2000") RouteDefinition
        .setBody() ExpressionClause<ProcessorDefinition<RouteDefinition>>
        .simple( text: "Timer search fired at ${header.firedTime}") ProcessorDefinition<RouteDefinition>
        .toF( uri: "bean:searchOldTweetsBean?method=search", SearchOldTweetsBean.NAME) |
    .to("mongodb3:mongoBean?database=%s&collection=%s&operation=insert",
        mongoDbConfig.getDbName(),
        mongoDbConfig.getCollectionName());
}
```

# Configuração da Rota

```
protected void configureDelayWhenRateLimitExceeded() {  
    onException(Exception.class)  
        .process(exchange -> {  
            LOGGER.info("{} tweets read", tweetsRead.get());  
            tweetsRead.set(0);  
        })  
        .log("Error delivering message")  
        .delay(60000 * 15);  
}
```

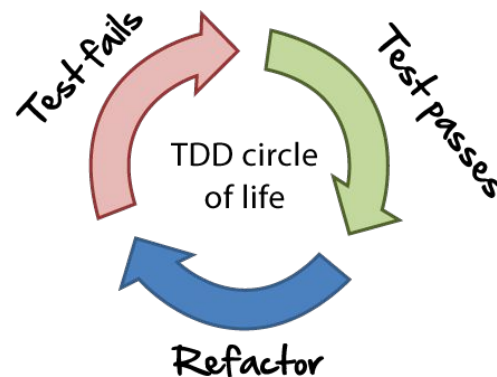
# Testes Unitários

```
@Test
public void datePlusDaysAssert() {
    final String datePlusDays = dateHelper.getDatePlusDays(
        date: "2018-10-31", days: 1);
    Assert.assertEquals( expected: "2018-11-01", datePlusDays);
}
```



# Testes Unitários

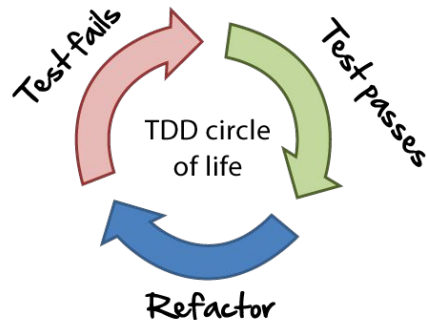
```
@Test
public void getQueryAsOrAssert() {
    final String queryAsOr = twitterHelper.
        getQueryAsOr( text: "java,scala");
    Assert.assertEquals( expected: "java OR scala", queryAsOr);
}
```



# Testes Unitários

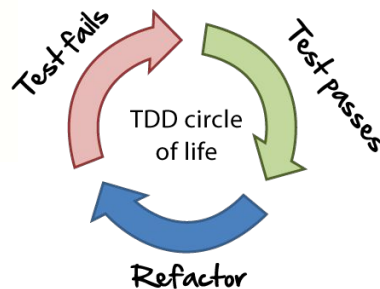
```
@Test
public void assertThatPrivateKeyExistsAfterGenerationKeys()
    throws NoSuchAlgorithmException, IOException, URISyntaxException {
    rsaKeyPairGenerator.generate();
    final boolean exists = Files.exists(new PathsHelper()
        .getResource( name: "security/private"));
    Assert.assertTrue(exists);
}
```

```
@Test
public void assertThatPublicKeyExistsAfterGenerationKeys()
    throws NoSuchAlgorithmException, IOException, URISyntaxException {
    rsaKeyPairGenerator.generate();
    final boolean exists = Files.exists(new PathsHelper()
        .getResource( name: "security/public"));
    Assert.assertTrue(exists);
}
```



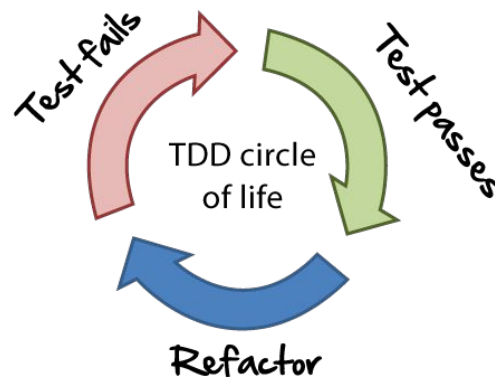
# Testes Unitários

```
@Before
public void setUp() throws URISyntaxException, IOException {
    rsaKeyPairGenerator = new RSAKeyPairGenerator(pathsHelper);
    try {
        Files.deleteIfExists(new PathsHelper()
            .getResource(name: "security/private"));
        Files.deleteIfExists(new PathsHelper()
            .getResource(name: "security/public"));
    } catch (Exception e) {
        LOGGER.error("Directory empty", e);
    }
    Mockito.when(pathsHelper.getResource(name: "security"))
        .thenReturn(new PathsHelper().getResource(name: "security"));
}
```



# Testes Unitários

```
@Test
public void passwordAfterDecryptShouldBeTheSameAsBefore() throws Exception {
    final String password = "somePasswordToUnitTest";
    final String encrypted = Base64.getEncoder()
        .encodeToString(rsaHelper.encrypt(password));
    final String decrypted = rsaHelper.decrypt(encrypted);
    Assert.assertEquals(password, decrypted);
}
```





# Criptografia

*# Api Twitter Url's*

twitter.config.oauth2.token.url=https://api.twitter.com/oauth2/token

twitter.config.search.tweets.url=https://api.twitter.com/1.1/search/tweets.json

*# Credentials*

twitter.config.key=

twitter.config.secret=

twitter.config.access.token=

twitter.config.access.token.secret=

*# Keywords to search, should be separated by ,*

twitter.search.keywords=bolsonaro,haddad,psl,pt,eleicoes

*# Date of start to search older tweets*

twitter.search.until=2018-10-31

*# Mongo properties*

mongo.db.name=tweetsDb

mongo.db.collection.name=tweets

mongo.db.host=127.0.0.1

mongo.db.port=27017



# Geração das chaves

```
public void generate() throws URISyntaxException,  
    IOException, NoSuchAlgorithmException {  
    final KeyPairGenerator keyGen = KeyPairGenerator.getInstance("RSA");  
    keyGen.initialize( keysize: 1024);  
    final KeyPair pair = keyGen.generateKeyPair();  
    final PrivateKey privateKey = pair.getPrivate();  
    final PublicKey publicKey = pair.getPublic();  
    final String securityPath = String.valueOf(pathsHelper  
        .getResource( name: "security"));  
    writeToFile(String.format("%s/public", securityPath),  
        publicKey.getEncoded());  
    writeToFile(String.format("%s/private", securityPath),  
        privateKey.getEncoded());  
}  
  
private void writeToFile(final String path,  
    final byte[] key) throws IOException {  
    Files.write(Paths.get(path), key);  
}
```

# RSA

```
@Autowired
public TwitterConfig(final RSAHelper rsaHelper,
    @Value("${twitter.config.oauth2.token.url}")
    final String tokenUrl,
    @Value("${twitter.config.search.tweets.url}")
    final String searchTweetsUrl,
    @Value("${twitter.config.key}")
    final String key,
    @Value("${twitter.config.secret}")
    final String secret,
    @Value("${twitter.config.access.token}")
    final String accessToken,
    @Value("${twitter.config.access.token.secret}")
    final String accessTokenSecret)
    throws InvalidKeyException, BadPaddingException, NoSuchAlgorithmException, Ill
this.tokenUrl = tokenUrl;
this.searchTweetsUrl = searchTweetsUrl;
this.key = rsaHelper.decrypt(key);
this.secret = rsaHelper.decrypt(secret);
this.accessToken = rsaHelper.decrypt(accessToken);
this.accessTokenSecret = rsaHelper.decrypt(accessTokenSecret);
}
```

# Dados obtidos

+ de 1.700.000 tweets

Relacionados à: bolsonaro, haddad, psol, pt, eleicoes

# Como responder as hipóteses?

- Identificação de frequência de candidato
- Análise de sentimentos de cada tweet
- Atribuição de 1 tweet para 1 candidato
- Maiores problemas:
  - Como calcular o sentimento para 1 Tweet?
  - Como determinar 1 candidato para 1 Tweet?

# Léxico de sentimentos

- Como léxico de sentimentos foi utilizado o SentiLex-PT01

```
gerador, gerador.PoS=Adj; GN=ms; TG=HUM; POL=0; ANOT=JALC  
corrupto, corrupto.PoS=Adj; GN=ms; TG=HUM; POL=-1; ANOT=MAN  
generoso, generoso.PoS=Adj; GN=ms; TG=HUM; POL=1; ANOT=MAN
```

# Cálculo de Sentimento

**-0.944911182523**

@GuilhermeBoulos @Haddad\_Fernando

Terrorista filho da puta do caralho ooooo vai invadir o cu da sua mãe com esses terroristas sem terra vão para puta q te pariu

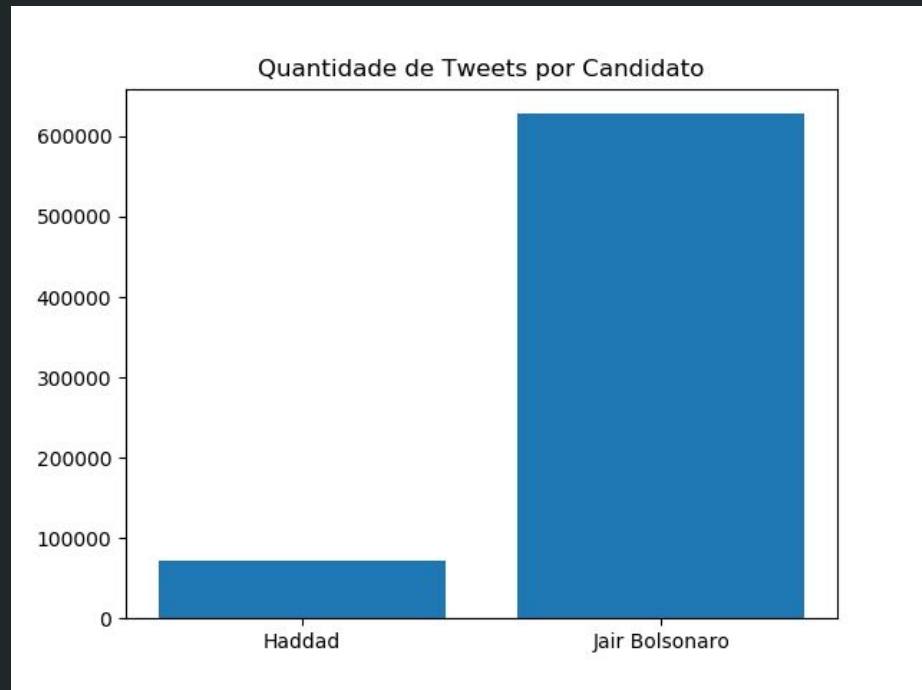
**0.784464540553**

RT @FernandoHortaOf: Todas as minhas aulas são contra Bolsonaro. Se filmarem verão. Minhas aulas são a favor dos direitos humanos, de igual...

$$sentimento(F) = \sum_{i=1}^m \frac{sentimento(P_i)}{\sqrt{comprimento(F)}}$$

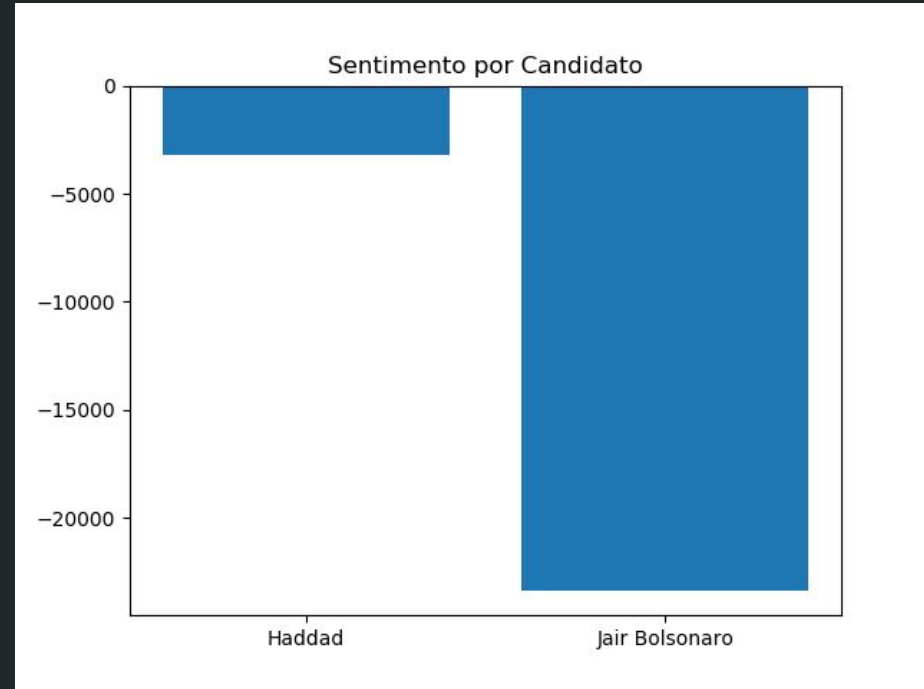
# Hipótese 1

Qual dos candidatos à presidência para o segundo turno da eleição brasileira do ano de 2018 possui mais popularidade?



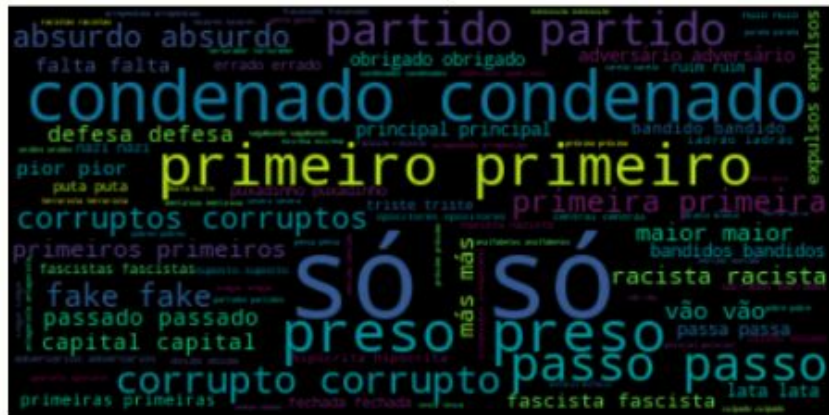
# Hipótese 2

Qual dos candidatos à presidência para o segundo turno da eleição brasileira do ano de 2018 representam um sentimento mais positivo?





## Piores Palavras - Jair Bolsonaro



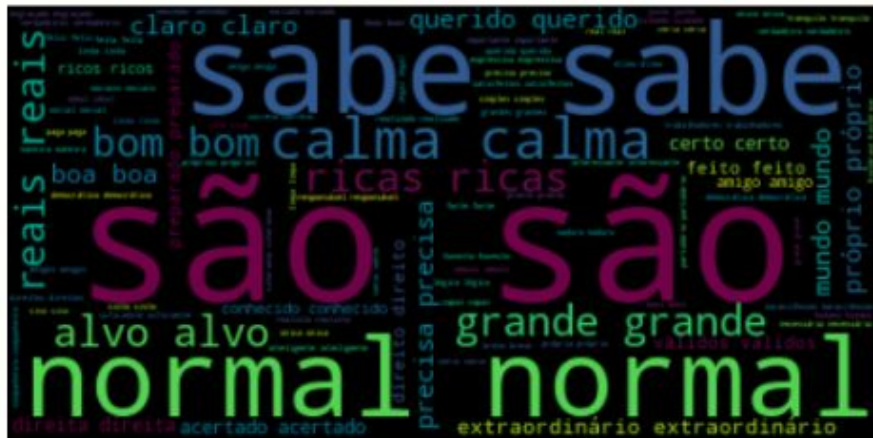
## Piores Palavras - Haddad



## Melhores Palavras - Jair Bolsonaro



## Melhores Palavras - Haddad



# Hipóteses Futuras

- Candidato mais popular por estado?
- Candidato mais popular no exterior?
  - França
  - Estados Unidos

# Melhorias

- Desenvolvimento de novo roteamento para limpeza da base de dados
  - Remoção de aspas (“ ‘)
  - Remoção de links
  - Remoção de #, @, entre outros
- Novos métodos de classificação

# Conclusão

O desenvolvimento deste trabalho permitiu uma experiência em lidar com grandes volumes de dados, podendo experienciar as dificuldades neste tipo de aplicação. Tive a oportunidade de implementar uma (mesmo que simples) análise de sentimentos, a qual posso facilmente testar e evoluir a cada eleição presidencial.

# Referências

<https://fernandofranzini.wordpress.com/2017/06/19/padroles-de-integracao/>

<https://www.devmedia.com.br/apache-camel-entenda-o-que-e/28747>

<https://cleverbuilder.com/articles/camel-tutorial/>

<http://xldb.fc.ul.pt/wiki/SentiLex-PT01>

<https://www.devmedia.com.br/criptografia-assimetrica-criptografando-e-descriptografando-dados-em-java/31213>

Lopes Rosa, RENATA. Análise de sentimentos e afetividade de textos extraídos das redes sociais. Universidade de São Paulo. São Paulo, 2015.



Muito Obrigado!

Talkei!

Maurício Mussatto Scopel



#JornalDaManhãETal