Análise de Mídia Social: termos mais frequentes no Twitter acerca da tragédia com a Chapecoense

1. Motivação

Conforme solicitado pelo professor Gabriel Coutinho, o trabalho deveria consistir na coleta e análise de um grande volume de dados. Por infeliz coincidência, durante a escolha do tema para esta análise, acontecia o que foi dado como a maior tragédia do esporte: a queda do avião que vitimou a delegação da Chapecoense que seguia para Medellín, na véspera da disputa da final da Copa Sul-Americana, além de integrantes da imprensa e tripulação.

Obviamente, tal evento gerou uma comoção mundial que logo foi refletida nas Redes Sociais. De acordo com vários veículos de comunicação, depois do acidente que resultou na morte de mais de 70 pessoas, a hashtag **#ForçaChape** se tornou a mais usada no Twitter em todo o mundo, em solidariedade aos parentes e amigos das vítimas.

Portanto, apesar do assunto de extrema tristeza, o cenário seria uma oportunidade para uma coleta de posts em tempo real com um tema bem específico e com grande volume de publicações em pouco tempo. Não restou dúvida: o assunto deveria ser, de fato, o acidente com o avião da equipe da Chapecoense e diversos jornalistas esportivos que aconteceu na madrugada do dia 29 de novembro de 2016 próximo a Medellín, na Colômbia.

2. Desenvolvimento

2.1 Estratégia de Coleta

O trabalho consiste, então, na coleta de aproximadamente 1 milhão de posts do Twitter sobre o acidente aéreo para posterior análise. Foi definida como chaves de busca os termos:

- **forçachape**
- **forcachape**
- **chapecoense**

Outros termos foram levantados: *acidente*, *tragédia*, *Chapecó*, *chape* etc., mas, para que a filtragem fosse bem específica e sem muitas ruídos, não foram considerados.

A coleta iniciou-se durante os primeiros momentos da chegada dos corpos das vítimas ao Brasil - às **12h38 do dia 03/12/16** - e encerrou-se logo após o anúncio pela Conmebol de que a Chapecoense poderia receber o título de campeã da Copa Sul-Americana - às **17h00 do dia 05/12/16**.

2.2 Ferramenta de Captura

2.2.1 O Twitter Listener Plus

Foi desenvolvida uma aplicação para escuta do Tweeter batizada de "Twitter Listener Plus" e codificada em linguagem C# com o framework Microsoft .Net versão 4.5.2 e banco de dados MongoDB. A IDE (do inglês Integrated Development Environment ou Ambiente de Desenvolvimento Integrado) utilizada foi o Microsoft Visual Studio 14. Para a conexão com o banco de dados inclui-se na aplicação o driver MongoDB Driver versão 2.3.0 e para leitura do Twitter foi adicionada à aplicação a biblioteca Tweetinvi versão 1.1.1.

A ferramenta consiste de um Programa Principal (classe Program) e uma classe que representasse a entidade Tweet.

O programa é executado em Console Windows e ao ser iniciado, começa a capturar todos os tuítes que no seu texto constasse as palavras definidas na etapa anterior, mostra na tela o texto e armazena em um banco de dados NoSQL não-relacional cada captura. Propositalmente, foram persistidos apenas a ***Data de Publicação***, o ***Identificador***, o ***Texto*** e o ***Idioma***do tuíte - dados básicos para a análise proposta.

> **O código-fonte está disponível aqui.**

Classe Tweet:

```
[Serializable]
public class Tweet
{
    public Tweet() { }

    public Tweet(DateTime created_at, string idStr, string text, string lang)
    {
        this.created_at = created_at;
        this.idStr = idStr;
        this.text = text;
        this.lang = lang;
    }

    [BsonElement("created_at")]
    public DateTime created_at { get; set; }

    [BsonElement("idStr")]
    public string idStr { get; set; }

    [BsonElement("text")]
    public string text { get; set; }

    [BsonElement("lang")]
    public string lang { get; set; }
}
```

2.2.2 API Twitter

A API do Twitter permite que diversos aplicativos conectem-se a ele para os mais variados fins. O funcionamento desta API é baseado em algumas tecnologias e conceitos como OAuth e REST que não serão detalhados neste trabalho. Para a captura do fluxo do Twitter em tempo real, é necessária uma autenticação que é basicamente o fornecimento das informações credenciais à API:

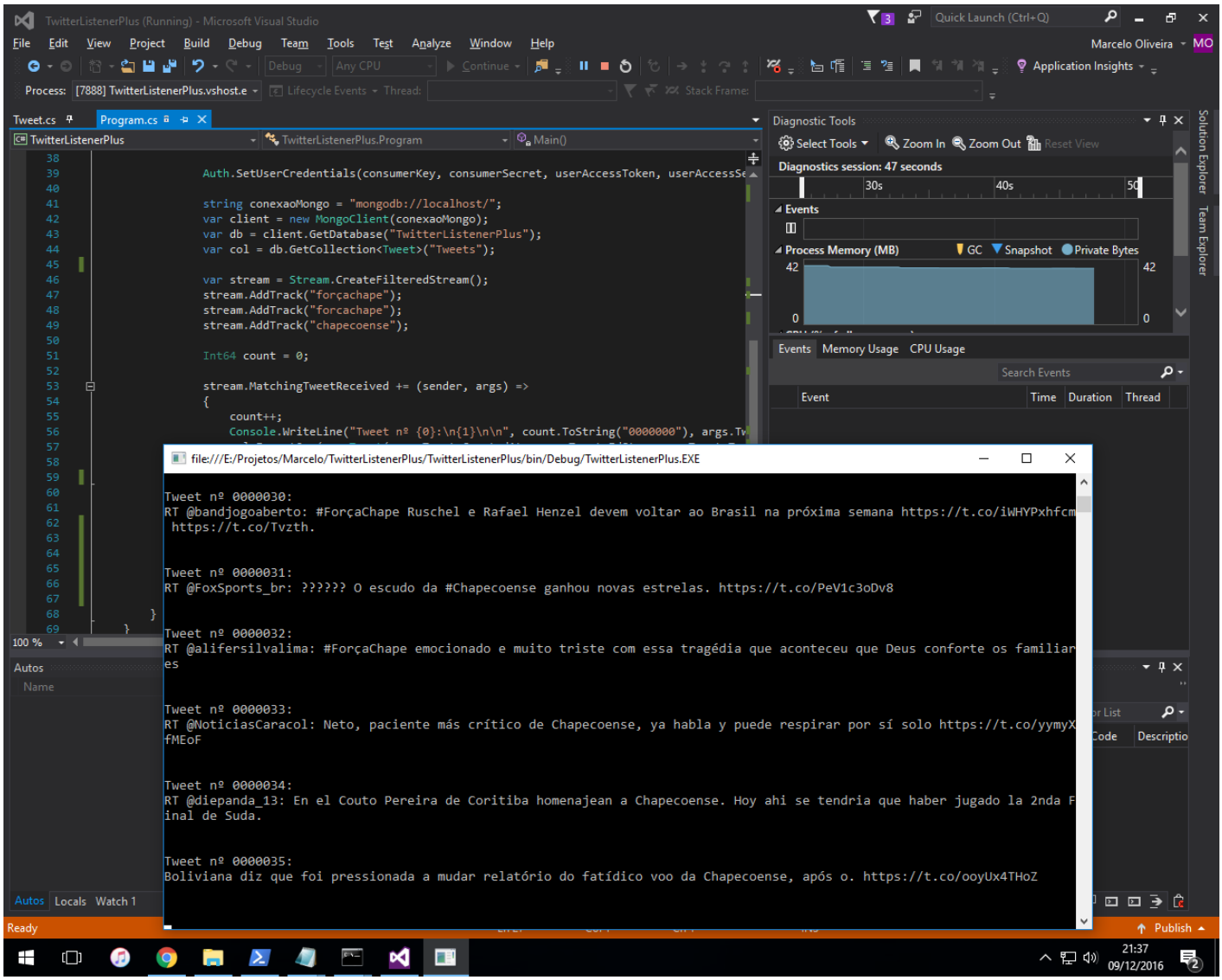
- ***consumerKey*** - Identificador do usuário consumidor dos dados;
- ***consumerSecret*** - Senha do usuário consumidor dos dados;
- ***userAccessToken*** - Token de acesso da aplicação;
- ***userAccessSecret*** - Senha de acesso da aplicação.

2.2.3 Execução do Programa

A máquina utilizada (um Desktop Windows 10 32 Bits, Pentium D 2,8GHZ com 2GB de RAM) não apresentou problemas de desempenho, incidentes ou falhas durante o coleta, armazenado e análise.

A ferramenta cumpriu bem seu objetivo e coletou **1.078.141 tuítes** com apenas 3 interrupções de no máximo 1 hora devido a problemas de rede (Internet) e queda rápida de energia. As interrupções não prejudicaram o trabalho por terem ocorrido apenas 3 vezes com tempo mínimo de duração e perda insignificante de tweets (menos de 2% do montante total).

> Screenshot da Execução em modo de Depuração:



2.3 Os Dados Coletados

2.3.1 O Armazenamento dos Dados

Como já citado, os dados dos tuítes foram devidamente armazenados em um banco de dados NoSQL MongoDB.

Foi criado um *Database* para a aplicação chamado *"**TwitterListenerPlus**"* e uma *Collection* chamado *"**Tweets**"* que guarda em documento JSON os tuítes coletados.

> Um arquivo com amostra dos Tuítes está disponível em JSON aqui.

Estatísticas da Coleção Tweets:

```
> db.Tweets.stats()
{
        "ns" : "TwitterListenerPlus.Tweets",
        "count" : 1078141,
        "size" : 344096944,
        "avgObjSize" : 319,
        "numExtents" : 15,
        "storageSize" : 460894208,
        "lastExtentSize" : 124993536,
        "paddingFactor" : 1,
        "paddingFactorNote" : "paddingFactor is unused and unmaintained in 3.0. It remains hard coded to 1.0 for com
        "userFlags" : 1,
        "capped" : false,
```

```
        "nindexes" : 1,
        "totalIndexSize" : 34993280,
        "indexSizes" : {
                "_id_" : 34993280
        },
        "ok" : 1
}
```

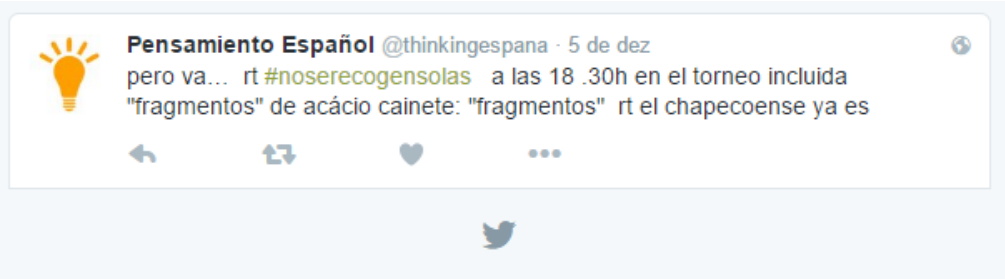
2.3.2 Primeiro Tuíte Coletado (Retweet de @Glaysson):

Link do Twitter: https://twitter.com/ChapecoenseReal/status/805002970559553536

Documento do Tuíte no MongoDB:

```
> db.Tweets.find().sort({created_at : 1}).limit(1).pretty()
{
        "_id" : ObjectId("5842d8fc2c8a4968008306ae"),
        "created_at" : ISODate("2016-12-03T14:38:50Z"),
        "idStr" : "805058720921161728",
        "text" : "RT @ChapecoenseReal: Procuramos uma palavra para agradecer tanto carinho e encontramos várias.\n#F
        "lang" : "Portuguese"
}
```

2.3.3 Último Tuíte Coletado (Publicação de @thinkingespana):



Link do Twitter: https://twitter.com/thinkingespana/status/805849259891363840

Documento do Tuíte no MongoDB:

```
> db.Tweets.find().sort({created_at : -1}).limit(-1).pretty()
{
        "_id" : ObjectId("5845b94e2c8a496800937a04"),
        "created_at" : ISODate("2016-12-05T19:00:10Z"),
        "idStr" : "805849259891363840",
        "text" : "pero va…  rt #noserecogensolas   a las 18 .30h en el torneo incluida  \"fragmentos\" de acácio cai
        "lang" : "Spanish"
}
```

3. Análise dos Dados

A análise foi divida em duas etapas a saber:

- A primeira com o objetivo de reduzir a quantidade grande de dados a um volume trabalhável e verificar a frequência por dia e por hora.
- E a segunda objetivando ir a um nível de maior detalhamento, tratamentos finos e preparação da apresentação dos resultados.

3.1 Análise Primária: Ttratamentos Iniciais

3.1.1 Iniciar o serviço para o Mongo DB

No Terminal Windows (CMD):

```
mkdir "C:\Program Files\MongoDB\Data\"
cd "C:\Program Files\MongoDB\Server\3.2\bin"
mongod --dbpath="C:/Program Files/MongoDB/Data"
```

3.1.2 Exportar os Tuítes em arquivo JSON (apenas backup)

No Terminal Windows (CMD):

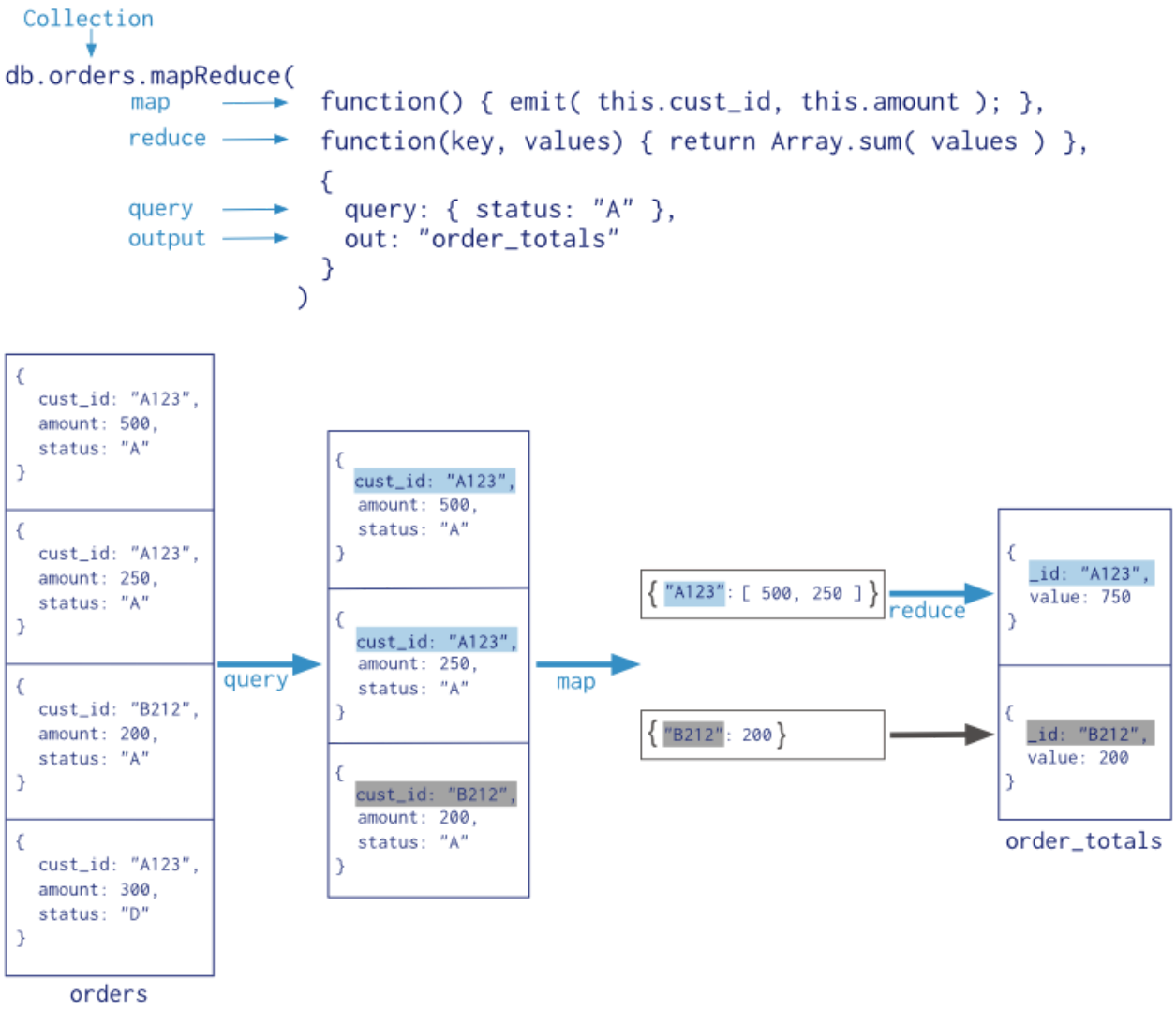
```
cd “C:\Program Files\MongoDB\Server\3.2\bin”
mongoexport --db TwitterListenerPlus --collection Tweets --out "Tweets.json"
```

3.1.3 Implementação do MapReduce

MapReduce é um modelo de programação e framework introduzido pelo Google para suportar computações paralelas em grandes coleções de dados em clusters de computadores. Aqui, foi realizado em apenas um cluster: o desktop Windows.

O MongoDB disponibiliza uma função nativa de MapReduce.

A imagem abaixo exibe um exemplo fictício para demonstrar o seu funcionamento:



Para essa análise primária, foram escritas as seguintes funções Map e Reduce:

map_fn:

1. Se o texto é indefinido, retorna;
2. Cria uma lista de palavras a partir do texto do Tuíte:
   - Mantém apenas consoantes e vogais (acentuadas ou não);
   - Exclui caracteres de quebra de linha e tabulação;
   - Exclui espaços repetidos;
   - Exclui a acentuação e cedilha;
   - Torna texto em letras maiúsculas;
3. Para cada palavra, criação de um registro novo.

reduce_fn:

1. Recebe chave e valor;
2. Retorna a palavra e sua frequência.

Implementação das Funções Map e Reduce:

```
var map_fn = function ()
{
    if (this.text == undefined)
        return;
```

```
    var regExp = new RegExp("[^a-záàâãäéèêëíìîïóòôõöúùûüçñ ]", "gi");

    var words = this.text.replace(regExp, "").replace(/\n/gi, " ").replace(/\t/gi, " ").replace(/\r/gi, " ").replace

    for (var i = 0; i < words.length; i++)
    {
        if (words[i] != "")
            emit(words[i], 1);
    }
};

var reduce_fn = function (key, value)
{
    return Array.sum(value);
};
```

3.1.4 Execução do MapReduce

Chamada da função MapReduce para a collection "Tweets":

O resultado é armazenado numa nova collection chamada "Terms".

```
var result = db.Tweets.mapReduce(
                        map_fn,
                        reduce_fn,
                        {
                            query: {},
                            out: "Terms"
                        });
```

Código para exibição da Lista dos 20 Termos mais frequentes - SEM TRATAMENTO FINO

```
var cursor_freq_terms = db.Terms.find().limit(20).sort({ value: -1 });

while (cursor_freq_terms.hasNext())
{
    printjson(cursor_freq_terms.next())
}

cursor_freq_terms.close();
```

Código para exibição do Volume de Tuítes por Dia

```
var cursor_tweets_day = db.Tweets.aggregate({ $group: { _id: { $dateToString: { format: "%Y-%m-%d", date: "$created_
                       , { $sort: { _id: 1 } });

while (cursor_tweets_day.hasNext())
{
    printjson(cursor_tweets_day.next())
}

cursor_tweets_day.close();
```

Código para exibição do Volume de Tuítes por Hora

```
var cursor_tweets_hour = db.Tweets.aggregate({ $group: { _id: { $dateToString: { format: "%Y-%m-%d %H:00", date: "$c
                        , { $sort: { _id: 1 } });

while (cursor_tweets_hour.hasNext())
{
    printjson(cursor_tweets_hour.next())
}

cursor_tweets_hour.close();
```

*No Terminal Windows (CMD), carregamento e execução do arquivo *.js* (Java Script) com os códigos acima.*

```
cd "C:\Program Files\MongoDB\Server\3.2\bin"
mongo

use TwitterListenerPlus
load("mapReduce.js")
```

Saída:

```
---------------------------------------------------------------------
                      Resultado do MapReduce
---------------------------------------------------------------------
{
        "result" : "Terms",
        "timeMillis" : 381572,
        "counts" : {
                "input" : 1078141,
                "emit" : 15154957,
                "reduce" : 1552369,
                "output" : 277616
        },
        "ok" : 1
}


---------------------------------------------------------------------
     Lista dos 20 Termos mais frequentes - SEM TRATAMENTO
---------------------------------------------------------------------
{ "_id" : "RT", "value" : 858866 }
{ "_id" : "CHAPECOENSE", "value" : 635933 }
{ "_id" : "DE", "value" : 506829 }
{ "_id" : "A", "value" : 402300 }
{ "_id" : "FORCACHAPE", "value" : 395384 }
{ "_id" : "LA", "value" : 240517 }
{ "_id" : "QUE", "value" : 208457 }
{ "_id" : "EL", "value" : 206111 }
{ "_id" : "E", "value" : 181867 }
{ "_id" : "O", "value" : 141667 }
{ "_id" : "DEL", "value" : 140140 }
{ "_id" : "EN", "value" : 127950 }
{ "_id" : "DA", "value" : 119390 }
{ "_id" : "CONMEBOL", "value" : 108332 }
{ "_id" : "FOR", "value" : 108287 }
{ "_id" : "CAMPEON", "value" : 90270 }
{ "_id" : "DO", "value" : 87816 }
{ "_id" : "RONALDINHO", "value" : 83519 }
{ "_id" : "AL", "value" : 83342 }
{ "_id" : "COPA", "value" : 78782 }
```

```
------------------------------------------------------------------------
                     Volume de Tweets por Dia
------------------------------------------------------------------------
{ "_id" : "2016-12-03", "total" : 472703 }
{ "_id" : "2016-12-04", "total" : 353588 }
{ "_id" : "2016-12-05", "total" : 251850 }


------------------------------------------------------------------------
                     Volume de Tweets por Hora
------------------------------------------------------------------------
{ "_id" : "2016-12-03 14:00", "total" : 49566 }
{ "_id" : "2016-12-03 15:00", "total" : 121715 }
{ "_id" : "2016-12-03 16:00", "total" : 84145 }
{ "_id" : "2016-12-03 17:00", "total" : 43317 }
{ "_id" : "2016-12-03 18:00", "total" : 29911 }
{ "_id" : "2016-12-03 19:00", "total" : 27771 }
{ "_id" : "2016-12-03 20:00", "total" : 32800 }
{ "_id" : "2016-12-03 21:00", "total" : 26519 }
{ "_id" : "2016-12-03 22:00", "total" : 25503 }
{ "_id" : "2016-12-03 23:00", "total" : 31456 }
{ "_id" : "2016-12-04 00:00", "total" : 24143 }
{ "_id" : "2016-12-04 01:00", "total" : 21240 }
{ "_id" : "2016-12-04 02:00", "total" : 18384 }
{ "_id" : "2016-12-04 03:00", "total" : 14358 }
{ "_id" : "2016-12-04 04:00", "total" : 9958 }
{ "_id" : "2016-12-04 05:00", "total" : 5950 }
{ "_id" : "2016-12-04 06:00", "total" : 4667 }
{ "_id" : "2016-12-04 07:00", "total" : 3633 }
{ "_id" : "2016-12-04 08:00", "total" : 4413 }
{ "_id" : "2016-12-04 09:00", "total" : 5646 }
{ "_id" : "2016-12-04 10:00", "total" : 10027 }
{ "_id" : "2016-12-04 11:00", "total" : 14156 }
{ "_id" : "2016-12-04 12:00", "total" : 15685 }
{ "_id" : "2016-12-04 13:00", "total" : 22881 }
{ "_id" : "2016-12-04 14:00", "total" : 22013 }
{ "_id" : "2016-12-04 16:00", "total" : 20699 }
{ "_id" : "2016-12-04 17:00", "total" : 22144 }
{ "_id" : "2016-12-04 18:00", "total" : 20010 }
{ "_id" : "2016-12-04 19:00", "total" : 19971 }
{ "_id" : "2016-12-04 20:00", "total" : 18577 }
{ "_id" : "2016-12-04 21:00", "total" : 20533 }
{ "_id" : "2016-12-04 22:00", "total" : 16100 }
{ "_id" : "2016-12-04 23:00", "total" : 18400 }
{ "_id" : "2016-12-05 00:00", "total" : 13425 }
{ "_id" : "2016-12-05 01:00", "total" : 16802 }
{ "_id" : "2016-12-05 02:00", "total" : 11930 }
{ "_id" : "2016-12-05 03:00", "total" : 8535 }
{ "_id" : "2016-12-05 04:00", "total" : 6620 }
{ "_id" : "2016-12-05 05:00", "total" : 4509 }
{ "_id" : "2016-12-05 06:00", "total" : 3602 }
{ "_id" : "2016-12-05 07:00", "total" : 3589 }
{ "_id" : "2016-12-05 08:00", "total" : 3753 }
{ "_id" : "2016-12-05 09:00", "total" : 4583 }
{ "_id" : "2016-12-05 10:00", "total" : 5516 }
{ "_id" : "2016-12-05 11:00", "total" : 6515 }
{ "_id" : "2016-12-05 12:00", "total" : 7390 }
{ "_id" : "2016-12-05 13:00", "total" : 9016 }
{ "_id" : "2016-12-05 14:00", "total" : 9050 }
{ "_id" : "2016-12-05 15:00", "total" : 9704 }
{ "_id" : "2016-12-05 16:00", "total" : 13078 }
{ "_id" : "2016-12-05 17:00", "total" : 63944 }
{ "_id" : "2016-12-05 18:00", "total" : 50145 }
```

```
{ "_id" : "2016-12-05 19:00", "total" : 144 }
true
```

3.2 Análise Secundária: Tratamentos Finos

Após a execução do Map-Reduce, foi necessário o refinamento da análise. A etapa anterior gerou uma coleção no banco de dados com todos os termos encontrados nos dados totais (~1M de tuítes). Isso gerou uma coleção com 277.616 documentos. Cada documento se refere a um termo (palavra) e sua frequência em relação a todos os tuítes.

> Um arquivo com amostra dos Termos está disponível em JSON **aqui**.

3.2.1 Stop Words

Essa segunda parte da análise consiste na remoção das palavras palavras que podem ser consideradas irrelevantes - as chamadas Stop Words.

> Um arquivo as Stop Words está disponível **aqui**.

Foram desconsideradas, também, as seguintes ocorrências:

Termo	Frequência	Observação
RT	858866	Retweet (abreviatura)
HTTPSTCOUTGUKSZNZ	30823	Link quebrado
HTTPSTCOREUCORU	30822	Link quebrado
HTTPS	26981	Link quebrado

A remoção foi feita utilizando um planilha eletrônica do Microsoft Excel 2016, onde foram filtradas apenas as palavra com significado e gerados os gráficos finais:

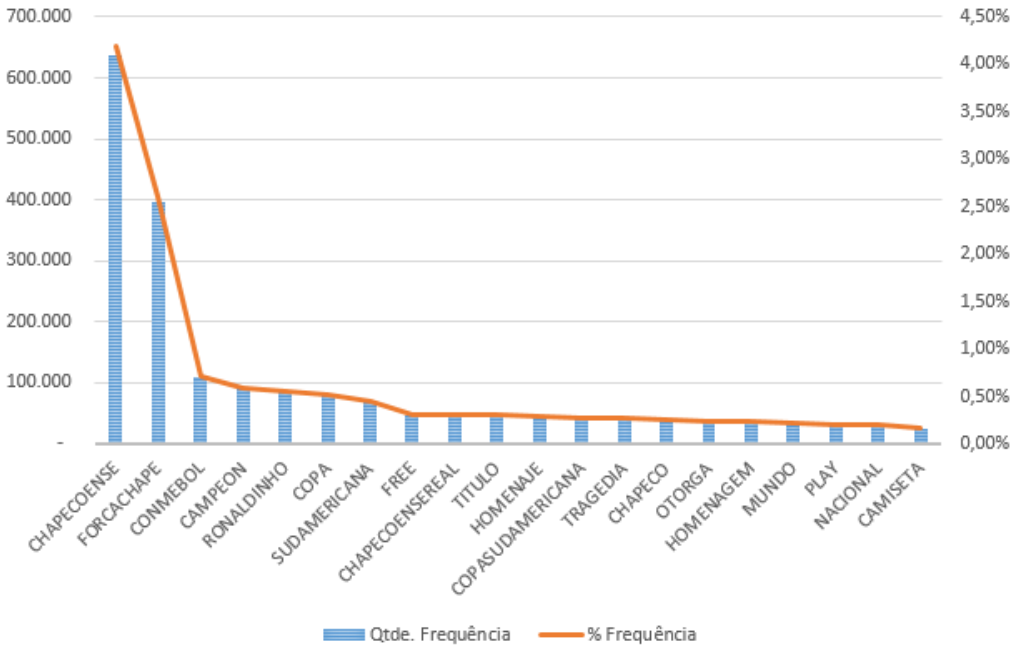
- Top 20 dos Termos mais Frequentes
- Quantidade de Tuítes por Dia
- Quantidade de Tuítes por Hora

3.2.2 Gráficos

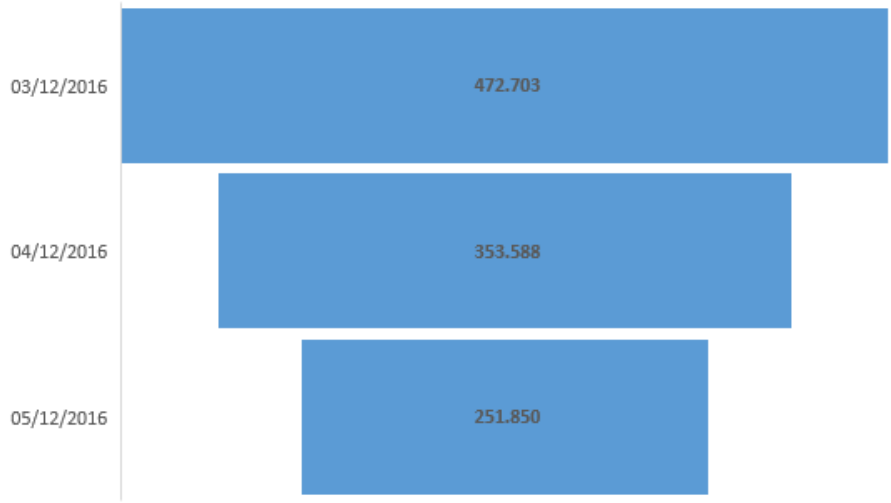
Um gráfico é uma representação dos dados, na forma de figuras geométricas - diagramas, desenhos, ou imagens - que permite ao leitor uma interpretação rápida e objetiva sobre o dados.

Portanto, um gráfico resume o que já se sabe sobre os dados e, também, revela o que não é evidente, transmitindo ideias e fenômenos que dificilmente seriam visíveis de outra forma - **Poder de Síntese**.

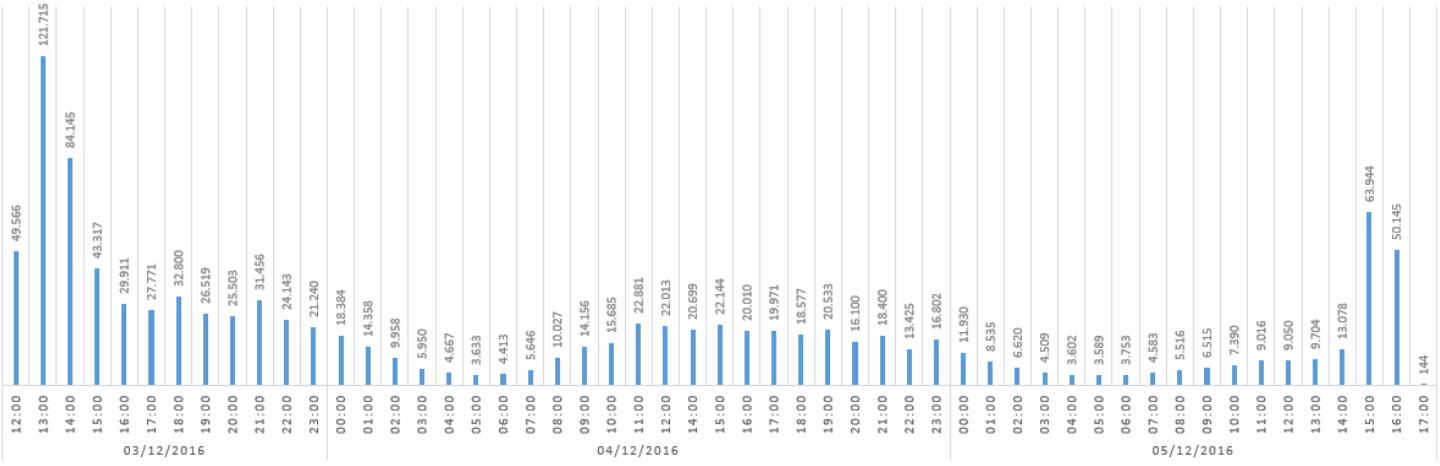
A. Top 20 dos Termos mais Frequentes



B. Quantidade de Tuítes por Dia



C. Quantidade de Tuítes por Hora



4. Conclusão

Levando-se em conta o que foi observado nesta análise, os termos mais frequentes representam mais um sentimento relativo ao Campeonato de Futebol cuja a final seria desputada pela equipe da Chapecoense do que sentimentos de consternação devido ao acidente. Lembrando que a captura de tuítes acontecia durante os ritos funerais das vítimas, esperava-se que termos relacionados à fé, apoio, solidariedade etc. fossem aparecer mais que assuntos sobre a competição esportiva. Contudo, não se observou isso por meio deste trabalho.

Para auxiliar na visualização destes resultados, segue a Word Cloud que representa a frequência dos termos que mais apareceram. Quanto mais vezes uma palavra foi mencionada, maior ela será na imagem. Evidencia-se, portanto, as palavras relativas a Copa Sul-Americana que à tragédia aérea: CONMEBOL, CAMPEON, RONALDINHO, COPA, SUDAMERICANA, TITULO, COPASUDAMERICANA.

Para o autor deste trabalho, a experiência obtida foi enriquecedora para seu currículo acadêmico e profissional. Seus objetivos de aprendizagem foram alcançados, pois o desenvolvimento do trabalho permitiu o estudo de novas ferramentas e a prática de conceitos aprendidos durante a disciplina de Bancos de Dados Não Relacionais do curso de Ciência de Dados e Big Data.

5. Bibliografia

1. **Análise de mídia social: análise de sentimento do Twitter em tempo real na Stream Analytics do Azure**. Disponível em https://docs.microsoft.com/pt-br/azure/stream-analytics/stream-analytics-twitter-sentiment-analysis-trends - Acesso em 09 de dez. de 2016.

2. **Twitter Developer Documentation - API Overview**. Disponível em https://dev.twitter.com/overview/api - Acesso em 09 de dez. de 2016.

3. **Utilizando a API do Twitter no desenvolvimento de aplicações web com PHP e cURL**. Disponível em http://www.linhadecodigo.com.br/artigo/3471/utilizando-a-api-do-twitter-no-desenvolvimento-de-aplicacoes-web-

com-php-e-curl.aspx - Acesso em 09 de dez. de 2016.

4. **MongoDB Manual - Map-Reduce**. Disponível em https://docs.mongodb.com/v3.0/core/map-reduce/ - Acesso em 09 de dez. de 2016.
5. **Brincando com dados: Ganhadores do Oscar**. Disponível em http://developers.hekima.com/data-science/brincando-com-dados/2016/02/11/importing_oscar_runners/ - Acesso em 09 de dez. de 2016.
6. **Listas de stopwords - stoplist (portugues, ingles, espanhol)**. Disponível em http://miningtext.blogspot.com.br/2008/11/listas-de-stopwords-stoplist-portugues.html - Acesso em 09 de dez. de 2016.
7. **Ferramenta Word Tagul Clouds**. Disponível em https://tagul.com/create - Acesso em 09 de dez. de 2016.
8. **Stop Words – Como Funcionam Palavras de Parada?**. Disponível em http://www.agenciamestre.com/seo/stop-words-como-funcionam-palavras-de-parada/ - Acesso em 09 de dez. de 2016.
9. **O que se esconde por trás de uma nuvem de palavras?**. Disponível em http://tarciziosilva.com.br/blog/o-que-se-esconde-por-tras-de-uma-nuvem-de-palavras/ - Acesso em 09 de dez. de 2016.
10. **Métodos para Análise de Sentimentos em mídias sociais - Dcc\UFMG**. Disponível em http://homepages.dcc.ufmg.br/~fabricio/download/webmedia-short-course.pdf - Acesso em 09 de dez. de 2016.