Disciplina: Tópicos de Bando de Dados

Twitter Sentimental Analysis

With Python + Docker + MongoDB

Alunos:

* Lucca Pessoa da Silva Matos
* Paulo Henrique Alves

Brasília - DF, dia 08 de julho de 2020

**Descrição do problema**

**Por que dados do Twitter?**

O Twitter hoje em dia é uma mina de ouro de dados. Ao contrário de outras plataformas e rede sociais, quase todos os tweets dos usuários são completamente públicos e acessíveis. Essa é uma enorme vantagem se você estiver tentando obter uma considerável massa de dados para executar análises. Os dados do Twitter também são bastante específicos.

**Problema**

Sabemos que em tempos de crise, a melhor ferramenta que temos são os dados. Tento isso como consideração, o problema central que temos aqui é a coleta de dados do Twitter que tenham a palavra "covid-19" em seu tweet para que passamos tirar análises e tentar prever que tipo de sentimento uma determinada quantidade de pessoas transmite quando utilizam essa palavra.

**Proposta**

Tendo em vista a pandemia ocorrida recentemente, temos como proposta a criação de um analisador sentimental para tweets com a palavra "covid-19" no Twitter e que utiliza o bando de dados NoSQL MondoDB para armazenar as informações coletadas e tratadas.

**Banco de dados**

**Caracterização do banco utilizado**

* Possui uma orientação a documentos - é formado por um conjunto de documentos JSON
* Não utiliza transações nem joins - Sem junções
* Dinâmico - sem esquema rígido.
* Alta performance
* Alta disponibilidade - Replicação
* Alta escalabilidade - Sharding – Distribuído

**Vantagem**

O NoSQL é mais indicado para aqueles sistemas que tenham necessidades maiores de armazenamento e desempenho.

Existem vários tipos de banco NoSql, o MongoDb é um banco Nosql orientado a documento. Esse tipo de banco tem uma performance incrível para escrita e não tão boa para leitura.

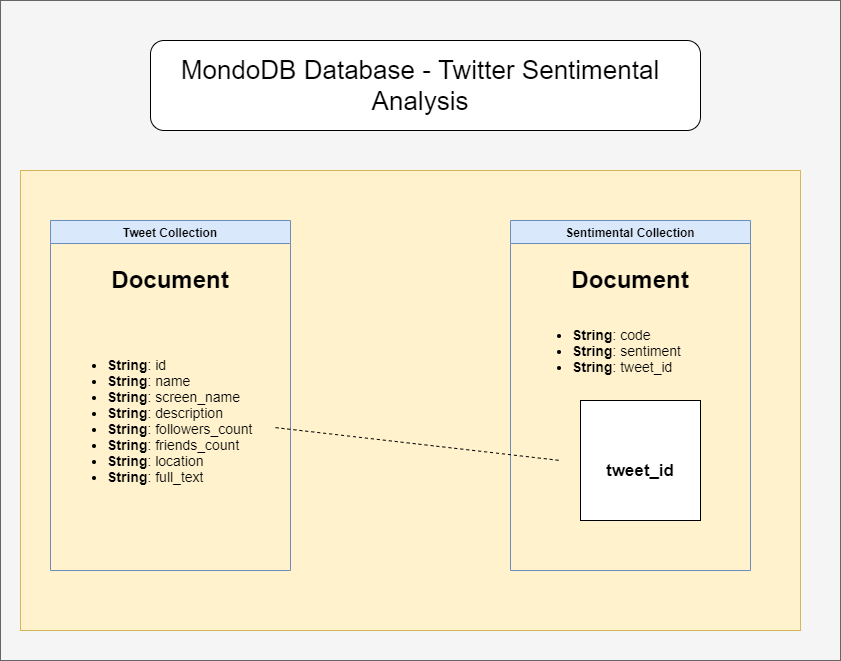
**Bom para**

* Catálogo de produtos de comércio eletrônico.
* Blogs e gerenciamento de conteúdo.
* Análise em tempo real e registro em alta velocidade, armazenamento em cache e alta escalabilidade.
* Objetivos pouco acoplados - o design pode mudar com o tempo.

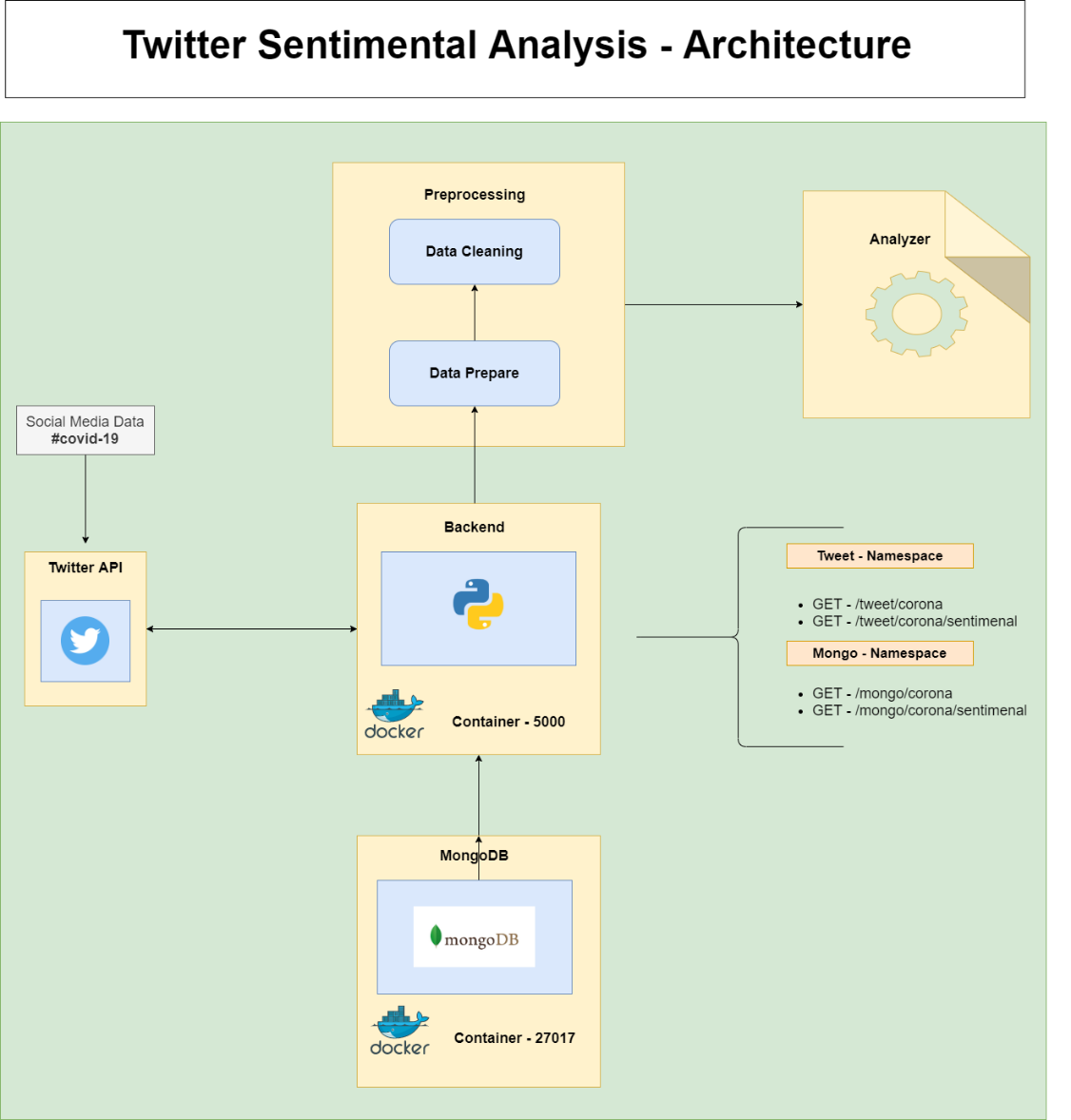
**Não é tão bom para**

* Sistemas altamente transacionais ou onde o modelo de dados é projetado com antecedência.
* Sistemas firmemente acoplados.

**Modelagem dos dados**

****

**Arquitetura com Dockers**



**Referências**

* <https://docs.mongodb.com/>
* <https://docs.docker.com/>
* <http://db4beginners.com/blog/consultas-no-mongodb/>
* <https://stackoverflow.com/questions/25589113/how-to-select-a-single-field-for-all-documents-in-a-mongodb-collection>
* <https://www.youtube.com/watch?v=_zxwlrYUHr4&t=932s>
* <https://bsi.uniriotec.br/tcc/textos/201612CarolinaTozzi.pdf>