



{ name: mongo, type: DB }

Raul Sena Ferreira

Sobre mim

Mestrando em Engenharia de Sistemas e Computação (PESC-UFRJ)

Bacharel em Ciência da Computação (UFRRJ)

Bolsista de TICs pelo CNPQ - Nível E (MNRJ-UFRJ)

Desenvolvedor Web (COPPE-UFRJ)

Data Science, Science for Good, Web Development e Network science

<http://lattes.cnpq.br/7007150957758256>

<http://www.raulferreira.com.br/>

Sobre mim

Um dos vencedores - 1º Desafio em Big Data (EMC²/UFRJ)

1º Lugar - Assistente de Pesquisa II do IPEA

Menção Honrosa - XIX Maratona Programação

3º Lugar - Primeira etapa do concurso Samsung Launching People

Um dos 40 selecionados - II Hackathon nacional de Dados Educacionais (INEP, Brasília-DF)

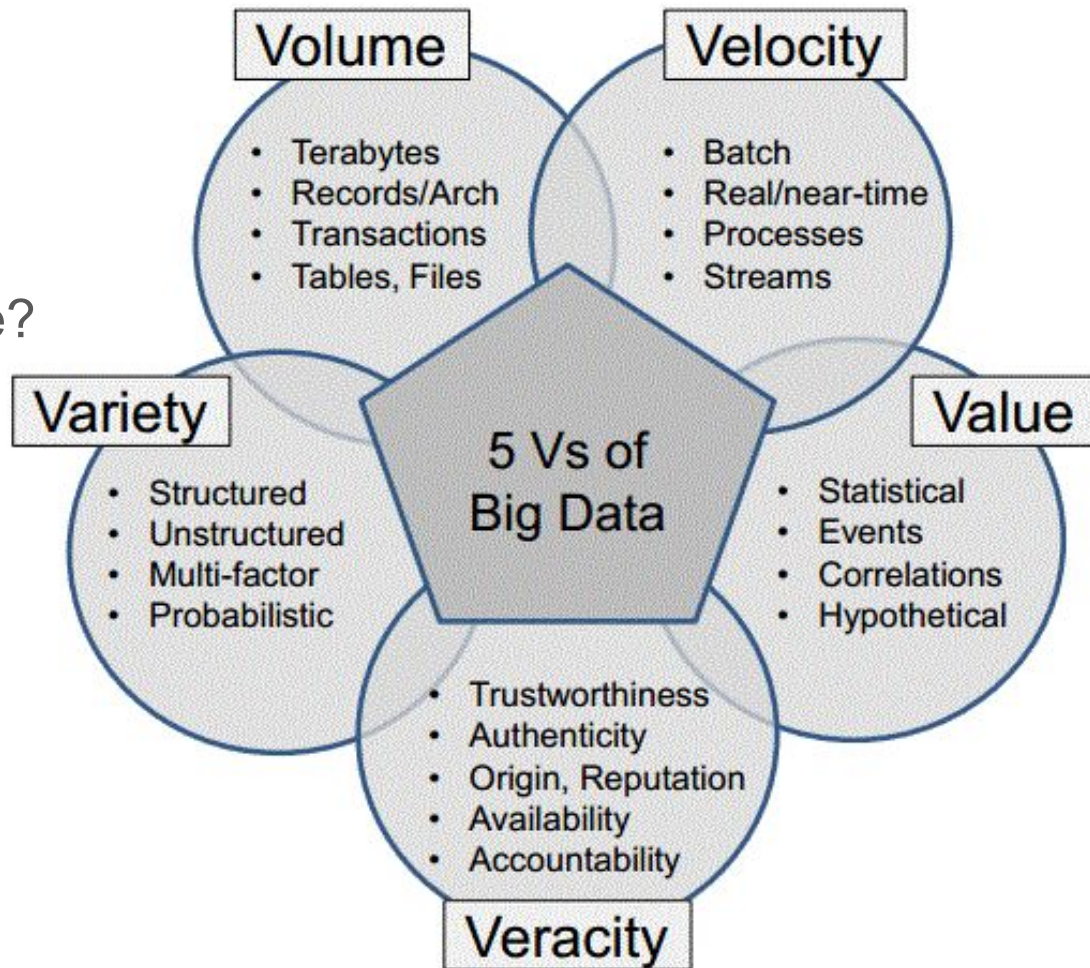
Big Data

Armazenar em SGBDs relacionais é o suficiente?

SQL puro continuará dando conta do recado?

NoSQL é a solução dos problemas?

Novos tipos de BD substituirão os BDs tradicionais?



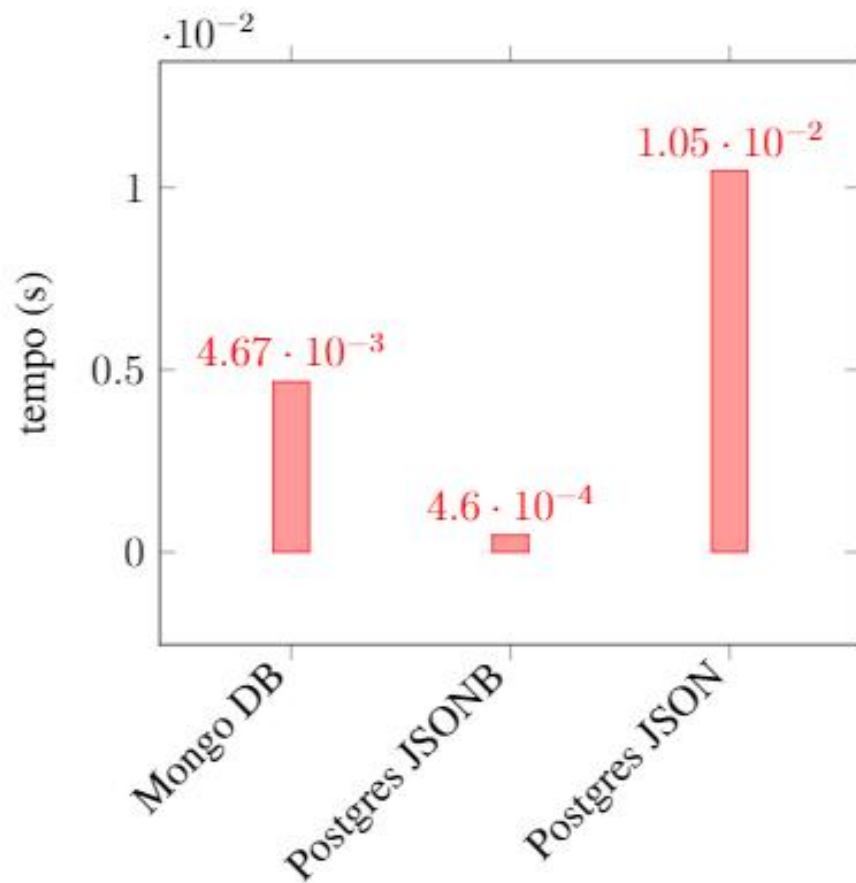
Postgres x MongoDB

Estudo experimental realizado por dois amigos do mestrado

Consulta simples, na base de dados do herbário nacional

Quase 2 milhões de registros

Resultados obtidos da média aritmética de 30 execuções



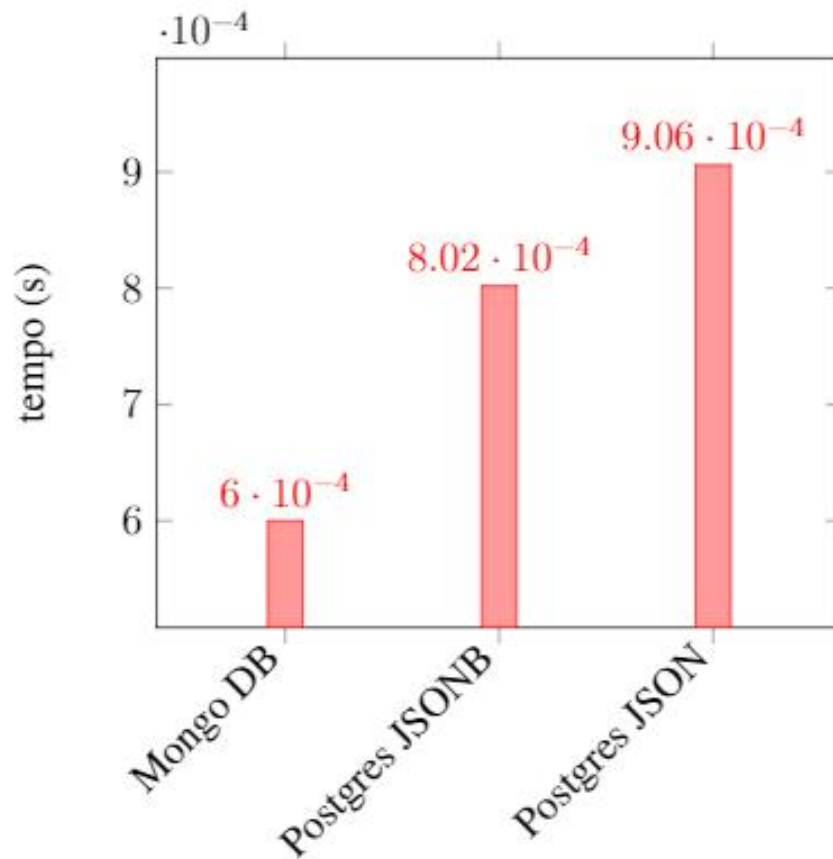
(f) Consulta C6

Postgres x MongoDB

Consulta acessando informação contida no segundo nível de um objeto JSON

Todos os objetos estavam devidamente indexados

JSONB, presente no Postgres a partir da versão mais nova (9.4)



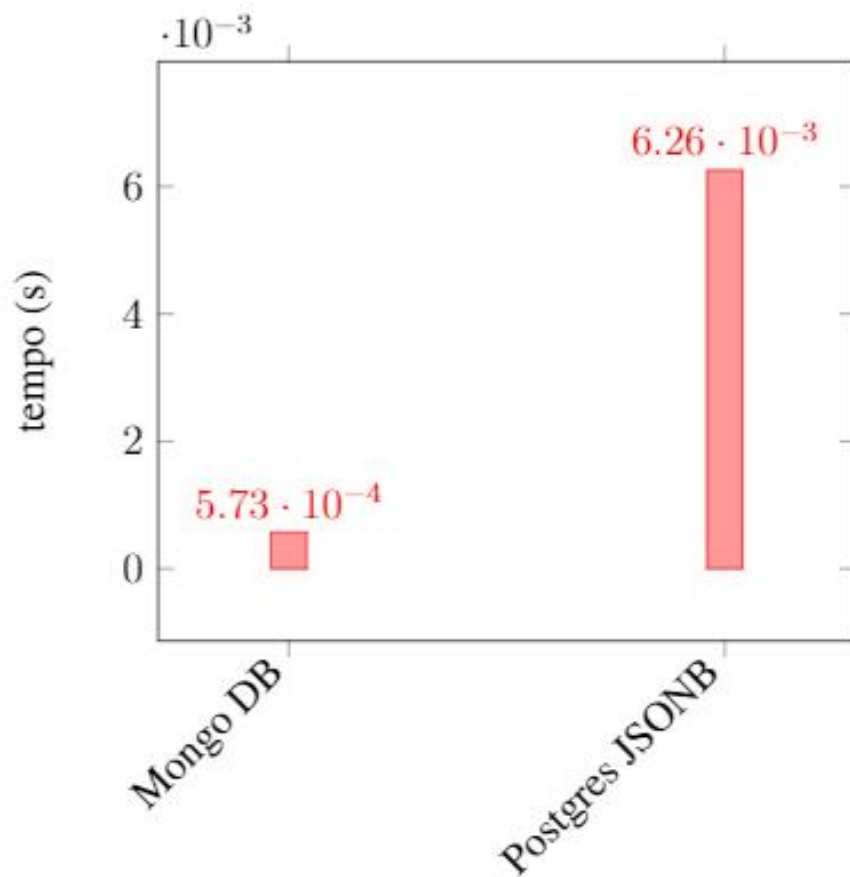
(a) Consulta C1

Postgres x MongoDB

Consulta acessando terceiro nível de um objeto JSON

Diferença chega a atingir uma ordem de grandeza!

MongoDB se destaca em consultas com mais de um nível de profundidade!



(b) Consulta C2

Postgres x MongoDB

O operador "contém" do Postgres JSONB se limita à verificações de igualdade, não sendo possível realizar testes do tipo ilike ou >= . Ex:

- `historico_determinacoes.taxon_name.genus_or_uninomial = Macha*`

MongoDB oferece suporte e ainda realizou a consulta em 0,007s

Para trabalhar com JSON, MongoDB ainda é a melhor escolha!

- Melhor desempenho
- Maior capacidade semântica

MongoDB

Do inglês humongous, "gigantesco", é um Banco de dados NoSQL

Código aberto e de alta performance, escrito em C++, C e Javascript

Schemaless (sem esquemas) e orientado a documentos

Armazenamento em BSON (JSON Binário)

Lançado em 2009

Quem usa?



Vantagens

Leitura rápida

Flexibilidade de esquema

Fácil integração com Node.js

Comunicação via JSON

Implementações nativas de Map Reduce, GIS, GRID, indexação e etc

Desvantagens

Não garante integridade

Tendência a ocupar muito espaço

Tendência de consumir memória

“Não possui join”

Quando usar?

Dados com estruturas heterogêneas

Sistema c/ uso intensivo de leitura de dados

Sistemas javascript like (Angular.js, Node.js)

Estrutura de dados com objetos auto contidos

Objetos com hierarquias

Quando não usar?

Aplicações transacionais

Aplicações voltadas ao modelo relacional

Aplicações que requerem alta consistência

Sistemas que necessitam fazer updates constantes

Um pouco de teoria

Tabela ~ Coleção

Tupla / Linha ~ Documento

Coluna ~ Campo

Chave primária (serial) ~ _id gerado pelo MongoDB (hexadecimal)

Joins ~ Documentos dentro de documentos

Mais um pouco de teoria

Coleção = {documento, documento 2, ..., documento N}

Documento = {[“chave” : “valor”], [“chave” : “valor”], ..., [“chave” : “valor”]}

Coleção **não** precisa ter schema pré-definido, ou seja,

Documentos de uma mesma coleção **não** precisam ter mesma estrutura

Mais um pouco, ainda, de teoria

Modelo BASE (Contrário ao ACID)

Banco inicializado como serviço (mongod)

Acesso simples pelo shell (mongo)

Banco pode ser inicializado como “replica set”

Possui sharding (escalável horizontalmente)

Alguns comandos

show dbs ---> Mostra todos os bancos contidos no sistema

show collections ---> Mostra todas as coleções contidas em um banco

use “db” ---> Cria ou acessa um banco já existente

db.stats() ---> Traz algumas estatísticas sobre as coleções

db.colecao.find() ---> Busca os registros, aceita cláusulas, regex e etc

\$explain ---> Serve para analisar *queries*

\$hint ---> Serve para forçar o otimizador a usar um índice específico

Estudo de caso 1 -



TransReport

TransReport - Fiscalizador colaborativo do transporte público

Objetivo:

- Gerar dados e indicadores mais precisos da qualidade do transporte público tanto para os órgãos fiscalizadores quanto para os usuários desse transporte
- Ferramenta de pressão popular pela qualidade do transporte
- Base de dados para futuras pesquisas e BI



TransReport - Mídia

4 COTIDIANO



Jornal Atual | Quinta-feira, 12 de fevereiro de 2015

TECNO-ATUAL

POR GIAN CORNACHINI
colunas@jornalatual.com.br



Governo promete Internet mais barata e democrática



Um novo plano de internet banda larga deve ser lançado neste semestre. É o que afirma Ricardo Berzoini, ministro das comunicações, em anúncio feito na terça-feira (10), em Brasília, durante um seminário de políticas públicas de comunicação. A ideia do projeto é oferecer internet a todos com preço acessível — apesar de os valores não terem sido informados.

O ministro ressaltou que o projeto só será amplo se houver parceria com os setores público e privado, mas não revelou quais empresas (Oi, Vivo, Net, GVT, entre outras) demonstraram interesse em participar da proposta.

As declarações de Berzoini foram feitas em discurso favorável à regulação socioeconômica das comunicações. O ministro disse acreditar que a mão do governo no setor contribui para fazer da internet um instrumento de democratização. Para isso, o Ministério das Comunicações irá estimular o investimento das empresas em áreas mais pobres e afastadas, pois hoje as grandes operadoras de telefonia concentram investimentos em áreas urbanas e regiões mais ricas.

APLICATIVO DA SEMANA

Mais atuantes pela qualidade no transporte público

Quem utiliza transporte público já passou por maus bocados com ônibus que não param, e veículos sucatados. Mas é certo que a cada ônibus que a gente insiste, tiramos 50 carros do engarrafamento. Claro que muitos preferem ter o próprio automóvel a pegar uma lotação sem ar-condicionado. Porém, a luta por um serviço de respeito à população não para, e uma nova ideia surgiu para somar forças à batalha: o aplicativo "TransReport" para Android. Com ele, é possível relatar problemas no transporte público do Rio de Janeiro e região de forma colaborativa. Basta se registrar no app, escolher o tipo de reclamação, fazer um comentário e anexar uma foto para ilustrar (se tiver). A mensagem é enviada para a

Fetranspor — federação responsável por gerir empresas de ônibus no Rio de Janeiro — e a ação se consolida em uma pressão popular. Divulgue essa ideia!



Quer mais 2 GB na sua conta do Google? É fácil, fácil..

Você utiliza serviços do Google como Gmail e Drive? A gigante da tecnologia está oferecendo 2 GB para usuários que fizeram apenas uma tarefa: checar as configurações de segurança da conta. Até a próxima terá-feira (17), ao completar a "Verificação de segurança", o internauta irá garantir o espaço extra nos servidores da empresa.

Para isso, acesse o seguinte link: <http://goo.gl/o1Wlzs>.

Feito o login no site, é só conferir e fazer o cadastro em sua conta, e e-mail para receber a confirmação de senha, as atividades recentes e a lista de aplicativos permitidos a acessar seus dados. Depois de tudo pronto, basta clicar em "Verificação de segurança", o internauta irá garantir o espaço extra nos servidores da empresa.

Fiscalize o transporte público do Rio de Janeiro com o TransReport

por RAQUEL FREIRE em 06/11/2015 19h43

TransReport é um app que permite aos moradores do Rio de Janeiro fiscalizarem e avaliarem a qualidade do transporte público. Ele pode ser usado para fazer queixas sobre atraso nas linhas de ônibus, superlotação, acidentes, assaltos e má conservação do veículo, entre outras situações. A plataforma é gratuita e está disponível para **Android** e em versão web.

Nossa opinião

A simplicidade de uso do aplicativo é seu ponto forte. O formulário de reclamação é preenchido em uma só tela, onde constam os dados mais importantes para o registro da ocorrência. Leva menos de um minuto para enviar a queixa, tempo ideal considerando que a denúncia provavelmente será feita de dentro de um transporte público.

O visual colabora com a usabilidade, trazendo um plano de fundo branco e fontes de tamanho razoável. Além de não exigir esforço da vista, o design é agradável e compatível com o de um app atualizado. Outra coisa boa é que o TransReport conta com várias possibilidades de reclamação, de forma que praticamente qualquer situação está contemplada.

A crítica principal é que o aplicativo só funciona para a cidade do Rio de Janeiro, embora reivindique servir para todo o estado. A lista de localidades só contempla os bairros da capital, deixando os outros municípios de fora. Mesmo aí há uma restrição grande: só são aceitas reclamações para ônibus. Os desenvolvedores explicam que estão trabalhando na expansão para barcas, metrô e BRT — o que exclui os trens.

Outro aspecto negativo está na impossibilidade de enviar reclamação sem informar dados pessoais. Como não fica claro como serão usadas essas informações, há sempre uma desconfiança sobre se as empresas de ônibus terão ou não acesso ao e-mail do usuário.

Prós

- Gratuito
- Fácil de usar
- Útil
- Muitas opções de situações para reclamar

Contras

- Só para a cidade do Rio de Janeiro, embora o app alegue servir para todo o estado
- Não permite registrar ocorrência sem informar dados pessoais

FOTOS



ACESSAR

Web | mais info

NOSSA OPINIÃO

8.5
NOTA 11

A simplicidade de uso do aplicativo é seu ponto forte. O formulário de reclamação é preenchido em uma só tela, onde constam os dados mais importantes para o registro da ocorrência. Leva menos de um minuto para enviar a queixa, tempo ideal considerando que a denúncia provavelmente será feita de dentro de um transporte público.

INTERFACE	9
USABILIDADE	10
DESEMPENHO	10
FUNCIONALIDADES	5

VEJA A ANÁLISE COMPLETA →

OPINIÃO DA GALERIA

★★★★★

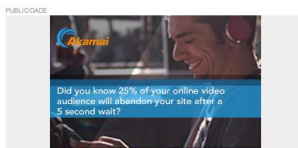
Michel Dias de Arruda ★★★★★
Como aplicativo. Simples e útil. Utilizo sempre!
QUAL A SUA OPINIÃO? →



FÓRUM

Você ficou com alguma dúvida sobre Produtividade? Visite o Fórum TechTudo e peça ajuda para a comunidade.

FAÇA UMA PERGUNTA





Primeiro protótipo em Android e o Site foram criados por:

Raul Sena Ferreira (WebService e Banco de dados)

Renan Prata (Android)

Vinícius Mororó (Testes)

Ana Figueiredo (Design)



TransReport - Tecnologias utilizadas

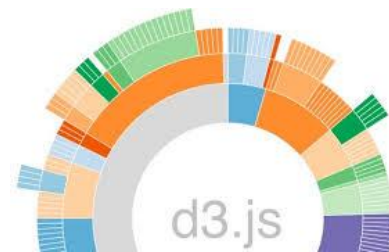


mongoDB

{ name: mongo, type: DB }



ANDROID





TransReport - Fluxo da ferramenta

O usuário acessa o aplicativo e escolhe o nº da linha

Em seguida, escolhe o identificador do ônibus

Escolhe o tipo de problema encontrado

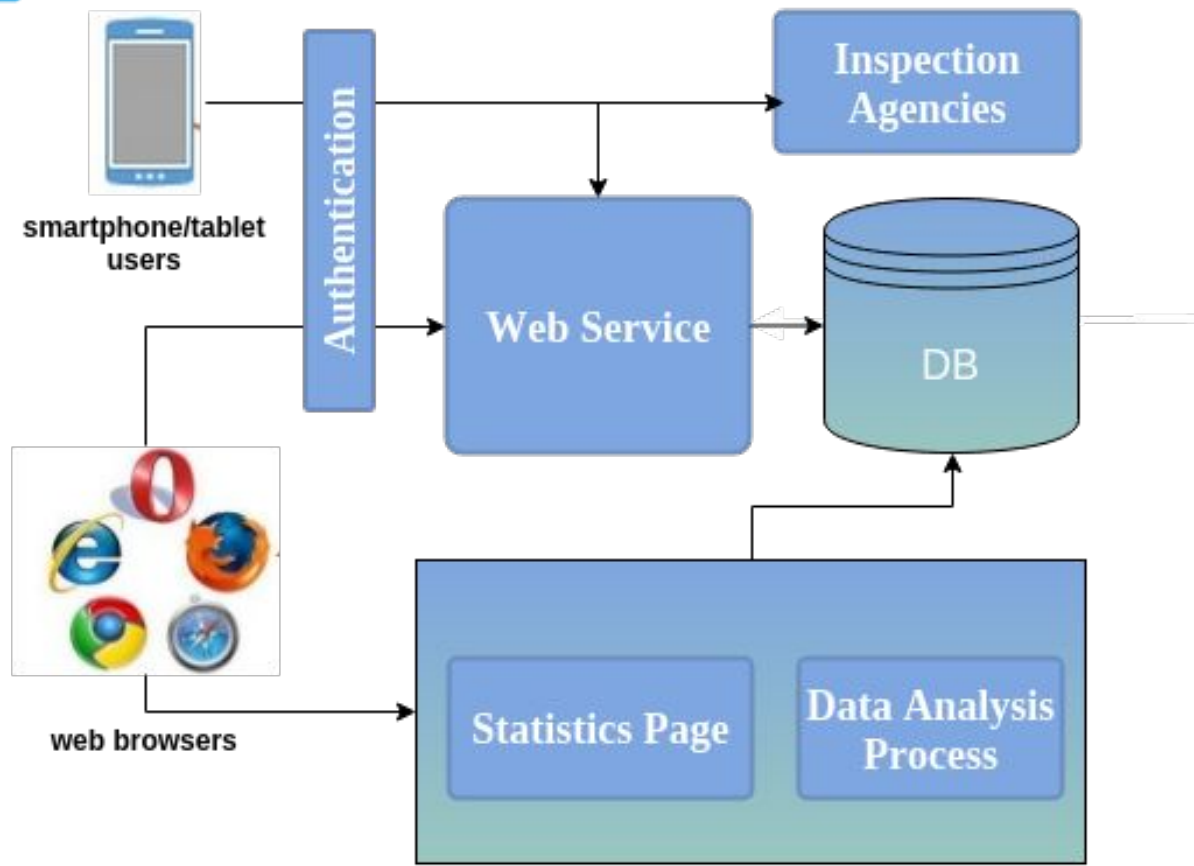
Tira e envia uma foto do problema (opcional);

Digita um comentário de até 100 caracteres(opcional);

Ao apertar o botão “enviar” a mensagem é enviada por SMS para o Twitter do Faleônibus e para nossa base



TransReport - Arquitetura





TransReport - Alguns detalhes (MongoDB)

Webservice (Node.js) fica “escutando” através de uma URL (configurada com express), juntamente com uma chave secreta, as requisições do APP (Android) e em seguida, são salvas no MongoDB

Imagens enviadas (Android) e guardadas (MongoDB) em formato Base64

Algumas agregações utilizam MapReduce do próprio MongoDB

Joins são desnecessários, documento já contém tudo o que se precisa



TransReport - Alguns detalhes (MongoDB)

Cada funcionalidade é uma coleção (reclamação e avaliação de qualidade), cada reclamação/avaliação é um documento

5 instâncias do MongoDB são inicializadas como serviço simultaneamente (Maior disponibilidade x Maior consumo de memória)

Serviço é restartado semanalmente (Evita-se Memory Leak)

Mais fácil e mais rápido em buscas em níveis mais profundos do conteúdo armazenado (Processo de BI simplificado)



TransReport - Alguns detalhes (Aplicação)

Webservice restarta sozinho quando algum arquivo é atualizado ou acontece alguma falha (módulo PM2)

Arquivo package.json permite o download de todas as dependências facilmente

Todos os dados são salvos como JSON

Todos os dados são recuperados como JSON

Toda a aplicação é escrita com apenas uma linguagem, Javascript!



TransReport - Links

Site: www.transreport.com.br

Estatísticas: <http://www.transreport.com.br:4000/stats>

Feed de reclamações: <http://www.transreport.com.br:4000/news>

Google Play: <https://play.google.com/store/apps/details?id=com.transreport>

Estudo de caso 2 - Projeto Myosotis

Myosotis - Uma abordagem NoSQL no auxílio a unificação e estudo dos registros de pessoas desaparecidas

Objetivo:

- Modelo de unificação das diferentes bases de dados existentes na internet
- Modelo de interface de consulta (espacial e textual) e análise dos dados já coletados
- Disponibilização dos dados coletados através de API

Myosotis - Tecnologias utilizadas (Bancos de dados)

ETL:



{ name: mongo, type: DB }

Consulta:



PostgreSQL




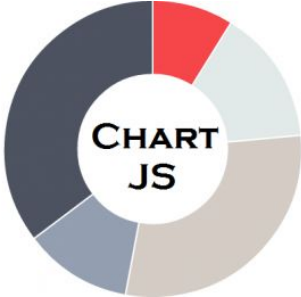
Cache:



redis

Myosotis - Tecnologias utilizadas (Aplicação)

Servidor:  python  

Interface:   



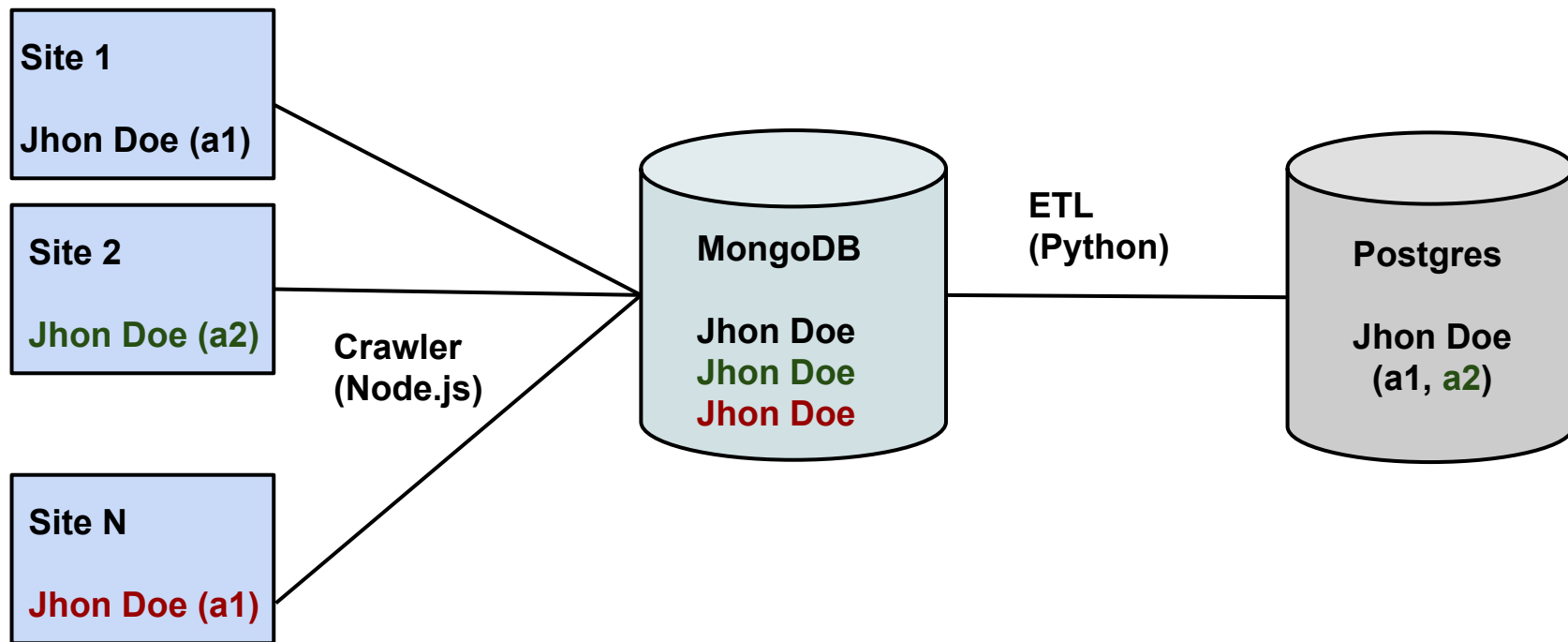
Myosotis

Modelo de unificação das bases:

- O crawler feito em Node.js varre os sites que possuem registros de pessoas desaparecidas
- As informações são guardadas no MongoDB onde cada site é tratado como uma coleção de documentos
- Em seguida, um processo de ETL, feito em python, extrai, limpa, unifica e salva os dados de forma estruturada no Postgres

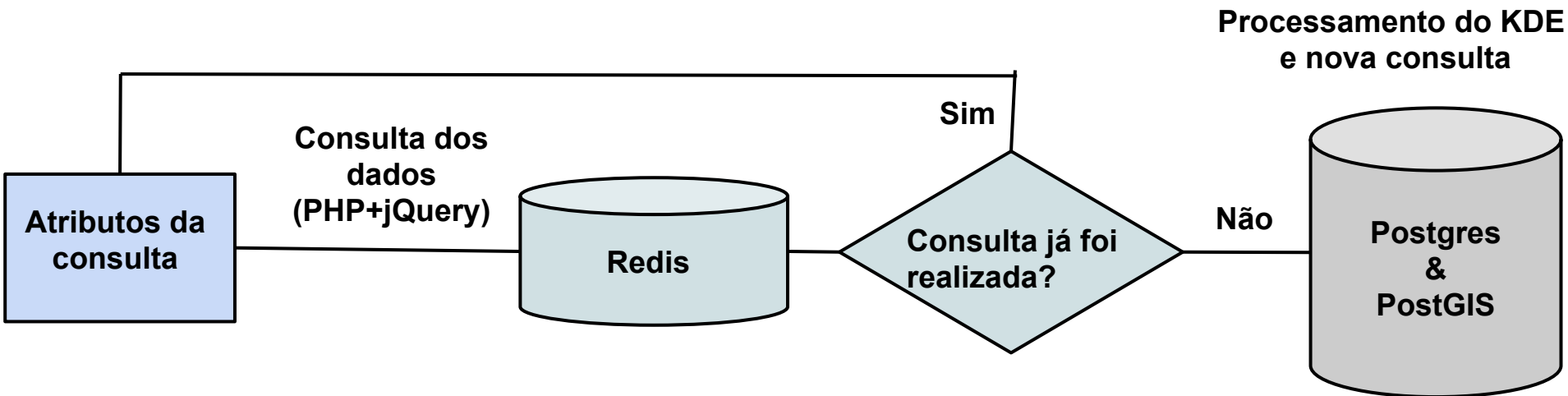
Myosotis - Fluxo da aplicação

Modelo de unificação das bases:

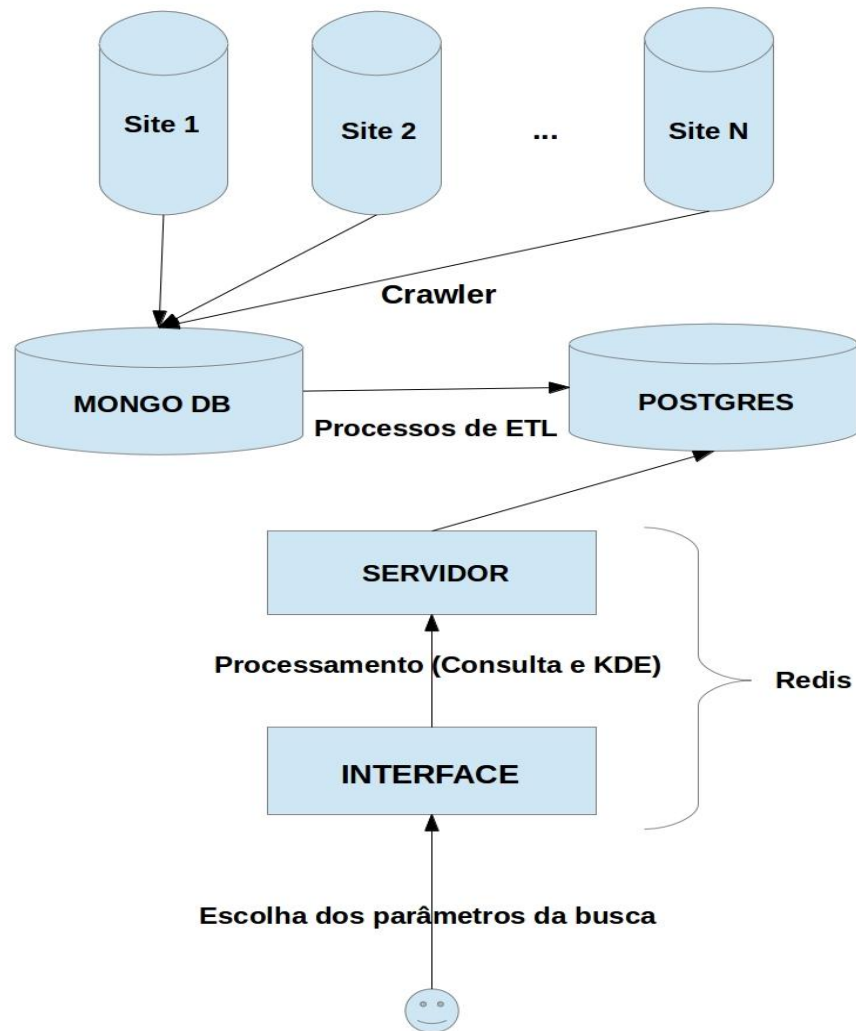


Myosotis - Fluxo da aplicação

Modelo de interface de consulta:



Myosotis - Fluxo



Myosotis - Alguns detalhes (Banco de dados)

Mongoose garante, de forma organizada, como e quais os dados que serão guardados pelo MongoDB

Redis é usado como cache mas suporta qualquer operação, relacionados a conjunto, sobre os dados

Consultas geográficas são feitas com o PostGIS integrado ao PostGres

Myosotis - Alguns detalhes (Aplicação)

Módulo Async, garante que as buscas do crawler sejam feitas de forma paralela

Módulo Request, faz com que uma url seja facilmente acessada pelo crawler

Módulo Cheerio, faz com que o html da página seja acessado via jquery

Myosotis - Links

A versão beta do sistema encontra-se funcional e pode ser acessado através do link:

- <http://www.projeto-myosotis.com.br>

Código do sistema está disponível sob a licença APACHE no link:

- <https://github.com/raulsenaferreira/Projeto-Myosotis>

Contribuições são bem vindas!

Estudo de caso 3 - Aplicações offline

Redes complexas aplicada ao estudo do fenômeno do desaparecimento de pessoas no Brasil

Objetivo:

- Utilizar teoria dos grafos e network science para modelar e investigar tal fenômeno
- Tentar encontrar subgrafos com perfis específicos dentro da rede

MongoDB + Python

MongoDB foi usado como repositório de dados

Conectado ao Python, para alimentar o grafo

Operações de agregação feitas no próprio banco

Cada coleção representa uma época da coleta dos dados

Estatísticas temporais fáceis de serem trabalhadas e modeladas

Buscas mais profundas no JSON de forma eficiente

Estudo de caso 4 - DESHIN

Sumarizador automático de texto

Desenvolvido por dois colegas de classe e eu durante uma matéria do mestrado

Objetivo:

- Extrair e criar sumários mais atrativos ao leitor de notícias em português
- Extrair e resumir as notícias levando em conta 3 tipos de apelos cognitivos

DESHIN

MongoDB armazena dados extraídos de um grande corpus de notícias em português

MongoDB integrado ao Python: Simples, prático e rápido em leitura

Link:

- <https://github.com/vbursztyn/DESHIN>

Conclusão

MongoDB é um banco poderoso e flexível para lidar com problemas de big data

Fácil integração com tecnologias/linguagens boas para se trabalhar com bigdata (python, node.js)

Possui grande praticidade para soluções em real time (online) e para soluções offline (BI e analytics)

O trio MongoDB + Node.js + Angular.js (ou algum framework MV*) pode ser poderoso

Links úteis

Livro em pdf da O'Reilly (50 dicas e truques para MongoDB):

- <http://m.friendfeed-media.com/ec44db47ced6bb8bb2d97c32746ccd449d6bdf21>

Interfaces administrativas para o MongoDB:

- <https://docs.mongodb.org/ecosystem/tools/administration-interfaces/>

MEAN (MongoDB+Express+Angular+Node.js):

- <http://mean.io/#!>

Links úteis

Tutorial do mong shell:

- <https://docs.mongodb.org/getting-started/shell/>

Tutorial sobre MongoDB:

- <https://docs.mongodb.org/manual/tutorial/>

Plataformas na nuvem (pra começar a desenvolver de graça!):

- <https://mongolab.com/>
- <https://www.heroku.com/>

Referências

<https://www.emaze.com/@AOTTTQLO/Big-data-Analytics-for-Security-Intelligence>

<https://www.mongodb.org/>

<https://www.mongodb.com/big-data-explained>

<https://nodejs.org/en/>

<http://d3js.org/>

http://s3.amazonaws.com/info-mongodb-com/MongoDB_BI_Analytics.pdf

Agradecimentos

Eduardo Morelli

Ygor Canalli & Kleyton Pontes - Postgres x MongoDB

Victor Bursztyn - DESHIN

Dúvidas e sugestões?

raulsf@cos.ufrj.br

raulsenaferreira@gmail.com

<https://www.linkedin.com/in/raulsenaferreira>

<https://github.com/raulsenaferreira>