



mongoDB®

# Dados no espaço-tempo: Trabalhando com dados georreferenciados no bancos de dados MongoDB

Ricardo Robaina

Tchelinux Pelotas

24 de Agosto de 2019



Tchelinux

# Sobre o palestrante



**Ricardo Robaina**

- 📍 Bagé
- 🎓 Engenheiro de Computação (Unipampa)
- 🎓 Mestrando em Computação Aplicada
- 🐧 Voluntário do Tchelinux

## Slides & Arquivos

⌚ <https://github.com/robainaricardo/talks>

# Sumário

1. Dados no espaço-tempo
2. MongoDB
3. Trabalhando com dado georreferenciados
4. Próximos Passos



# Dados no espaço-tempo

**Dados podem variar**

- Em relação ao tempo
- Em relação ao espaço
- Em relação ao espaço e ao tempo



# Dados no espaço-tempo

## Dados podem variar

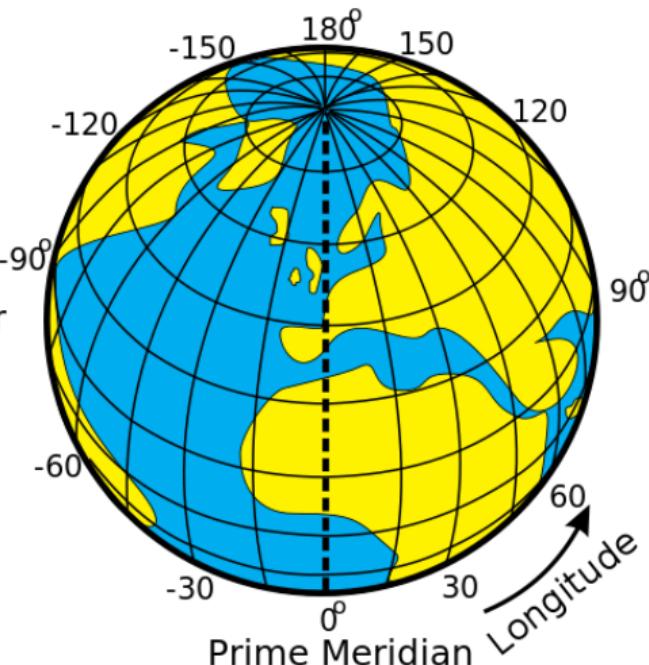
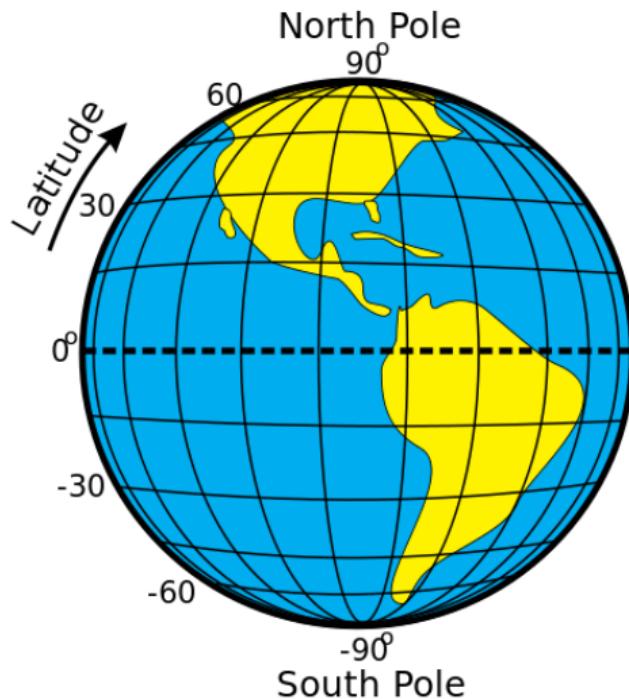
- Em relação ao tempo
- Em relação ao espaço
- Em relação ao espaço e ao tempo

## Formas de medir a variação

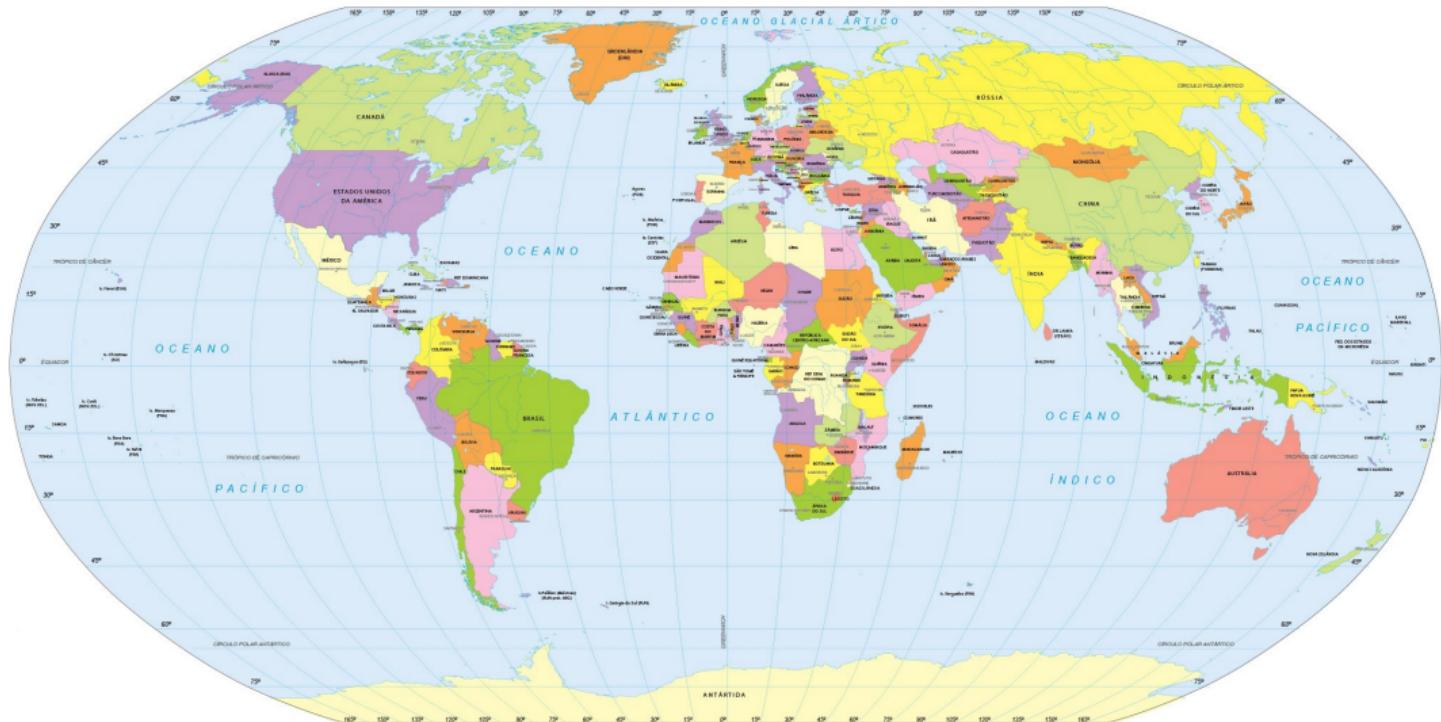
- Relógio (tempo)
- Sistema de navegação (espaço)



# Sistemas de referência



## Sistemas de referência



# Sistemas de navegação

## Localização

- Posição = (Latitude, Longitude)
- Mapa
- GPS



# GPS ou GNSS, eis a questão?

## Global Positioning System (GPS)

- Primeiro sistema de navegação por satélite
- Governo americano

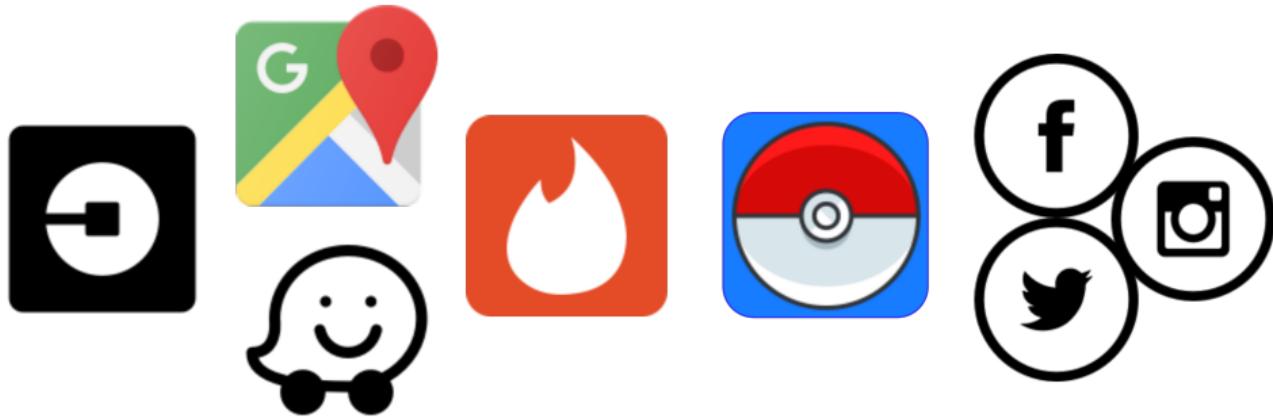
## Global Navigation Satellite System (GNSS)

Qualquer sistema de navegação por satélites. Existem vários:

- Glonass
- Galileo
- Beidou
- GPS

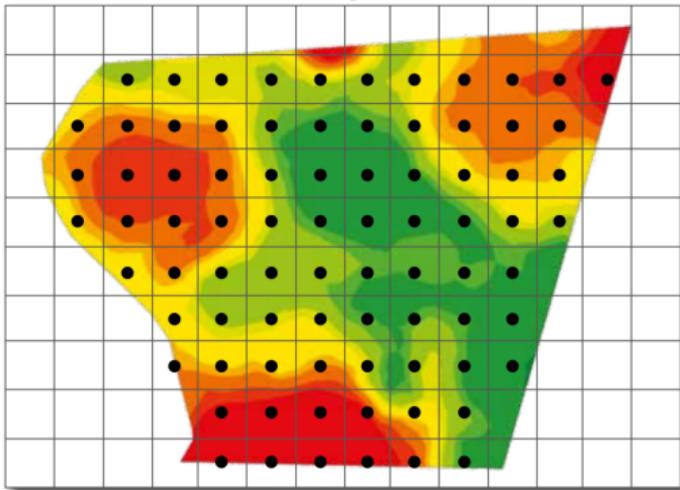


# Aplicações baseadas em localização

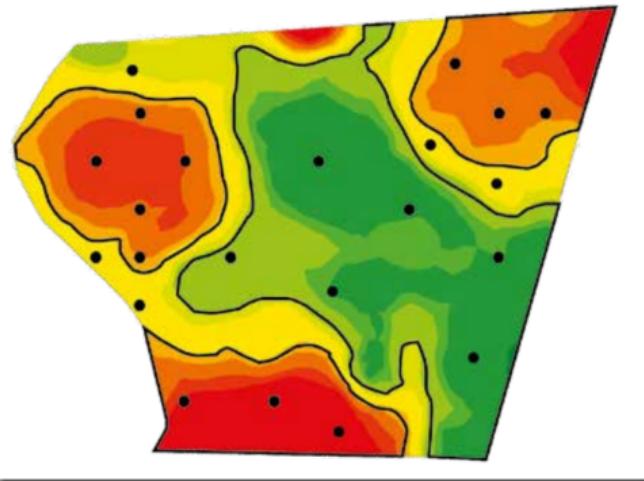


# Variabilidade espaço-temporal

Grid de amostragem tradicional



Zonas de manejo geradas a partir do mapa de condutividade elétrica



# Como usar isso tudo?

## Necessidade

- Coletar a informação espacial
- Armazenar esse dado
- Executar operações nesses dados



# Como usar isso tudo?

## Necessidade

- Coletar a informação espacial
- Armazenar esse dado
- Executar operações nesses dados

## Solução

- Sensor GNSS, metadados, etc..
- Banco de dados



# Bancos de dados

## Suporte a dados Georreferenciados

- PostgreSQL + PostGIS
- MongoDB



# Relacional vs NoSQL

## Relacional

*Structured Query Language*

- Baseado em tabelas e seus relacionamentos
- Estrutura estática
- Suporte a transações

## NoSQL

*Not Only SQL (Não Relacional)*

- Baseado em documentos, chave-valor
- Estrutura dinâmica
- Não possui suporte a transações





## The database for modern applications

MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era. No database is more productive to use.

# MongoDB



mongoDB®

## Características

- Open Source
- Orientado a documentos
- Suporte a dados e a operações georreferenciadas
- Drivers para várias linguagens de programação (C, Python, Java, Go...)

## Ferramentas

- Mongo Shell
- Mongo Compass



# Bancos de dados orientados à documentos: Documento

```
{  
    name: "sue",           ← field:value  
    age: 26,              ← field:value  
    status: "A",           ← field:value  
    groups: [ "news", "sports" ] ← field:value  
}
```



# Bancos de dados orientados à documentos: Coleção



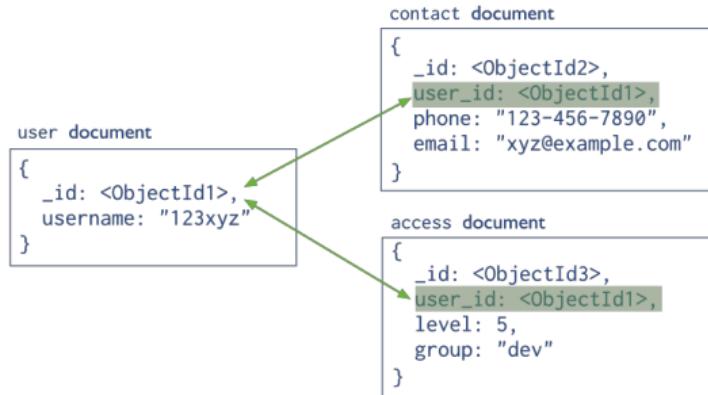
# Bancos de dados orientados à documentos: Designs

## Documentos aninhados

```
{  
  _id: <ObjectId1>,  
  username: "123xyz",  
  contact: {  
    phone: "123-456-7890",  
    email: "xyz@example.com"  
  },  
  access: {  
    level: 5,  
    group: "dev"  
  }  
}
```



## Documentos por referência



# Organização dos dados

## JSON

- JavaScript Object Notation
- `{“chave” : “valor”}`

## BSON

- Binray JSON
- Serialização binária de um objeto JSON

## GeoJSON

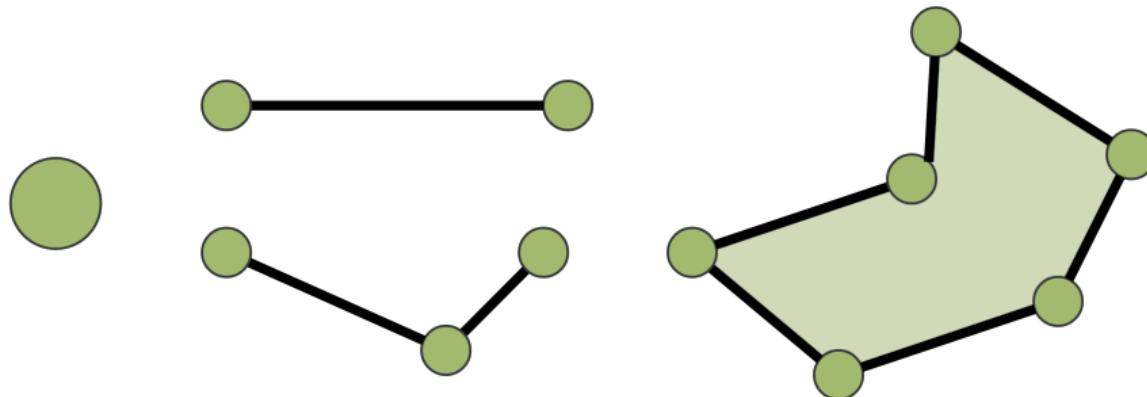
- Geospatial JSON (RFC 7946)
- Conjunto de tipos de dados estendido



# GeoJSON

## Tipos de dados georreferenciados

- Point
  - LineString
  - Polygon
- MultiPoint
  - MultiLineString
  - MultiPolygon



# Tipos de dados e índices

## Tipos de dados

- GeoJSON objects
  - Cálculos baseados em uma esfera, como a Terra<sup>1</sup>
- Legacy coordinate pair
  - Cálculos baseados em um Plano Euclidiano

## Índices

- 2dsphere
- 2d



---

<sup>1</sup>A Terra não é plana!

# Notação GeoJSON

## Organização dos dados

- type
- coordinates
  - longitude (-180, 180)
  - latitude (-90, 90)

```
<field>: { type: <GeoJSON type> , coordinates: <coordinates> }
```



# Notação GeoJSON

```
location: {  
    type: "Point",  
    coordinates: [-73.856077, 40.848447]  
}
```



# Operadores de consulta

## \$geoIntersects

- Seleciona objetos que intersectam com uma geometria GeoJSON

## \$geoWithin

- Seleciona objetos dentro de uma limite (geometria GeoJSON)

## \$near

- Retorna objetos próximos de um ponto

## \$nearSphere

- Retorna objetos próximos de um ponto (Earth-like sphere)



# Dados georreferenciados

- Dados das estações do INMET
- Inclusão de dados de normais climatológicas
- Inserção automática (pymongo)

```
_id: ObjectId("5d488e9af11cc49c6e269e8d")
numero: 346
codigo: 83980
nome: "BAGE"
UF: "RS"
localizacao: Object
  coordinates: Array
    0: -54.1
    1: -31.33333333
    type: "Point"
altitude: 242.31
inicio_operacao: 1912-01-01T00:00:00.000+00:00
fim_operacao: ""
situacao: NaN
temperatura: Array
pressao: Array
insolacao: Array
evaporacao: Array
nebulosidade: Array
umidade: Array
precipitacao: Array
vento: Array
evapotranspiracao: Array
```



# Visualização



# Consulta simples

```
$ use clima
$ db.estacoes.find({"nome": "BAGE"}).pretty()

{
    "_id" : ObjectId("5d488e9af11cc49c6e269e8d"),
    "numero" : 346,
    "codigo" : 83980,
    "nome" : "BAGE",
    "UF" : "RS",
    "localizacao" : {
        "coordinates" : [
            -54.1,
            -31.333333333
        ],
        "type" : "Point"
    },
    "altitude" : 242.31,
    ...
}
```



# Criação de índice

```
$ db.estacoes.createIndex({"localizacao": "2dsphere"})  
  
{  
    "createdCollectionAutomatically" : false,  
    "numIndexesBefore" : 1,  
    "numIndexesAfter" : 2,  
    "ok" : 1  
}
```



# Consulta \$nearSphere

```
$ db.estacoes.find({localizacao:{$nearSphere:{  
... $geometry:{type: "Point", coordinates: [-54.1, -31.3]},  
... $maxDistance: 300000}  
... }}, {"nome": 1, "_id": 0}).pretty()  
  
{ "nome" : "BAGE" }  
{ "nome" : "SANTANA DO LIVRAMENTO" }  
{ "nome" : "PELOTAS" }  
{ "nome" : "ENCRUZILHADA DO SUL" }  
{ "nome" : "SANTA MARIA" }  
{ "nome" : "RIO GRANDE" }  
{ "nome" : "BARRA DO RIO GRANDE" }  
{ "nome" : "CAMAQUA" }  
{ "nome" : "SANTIAGO" }  
{ "nome" : "SANTA VITORIA DO PALMAR" }
```



# MongoDB Compass

MongoDB Compass - localhost:27017/clima.estacoes

localhost:27017 STANDALONE MongoDB 3.8.3 Community

My Cluster

My Cluster

clima.estacoes

Documents Aggregations Schema Explain Plan Indexes Validation

DOCUMENTS 436 TOTAL SIZE 965.6KB AVG. SIZE 2.2KB INDEXES 2 TOTAL SIZE 36.0KB AVG. SIZE 18.0KB

filter

admin

clima

estacoes

config

golang-test

local

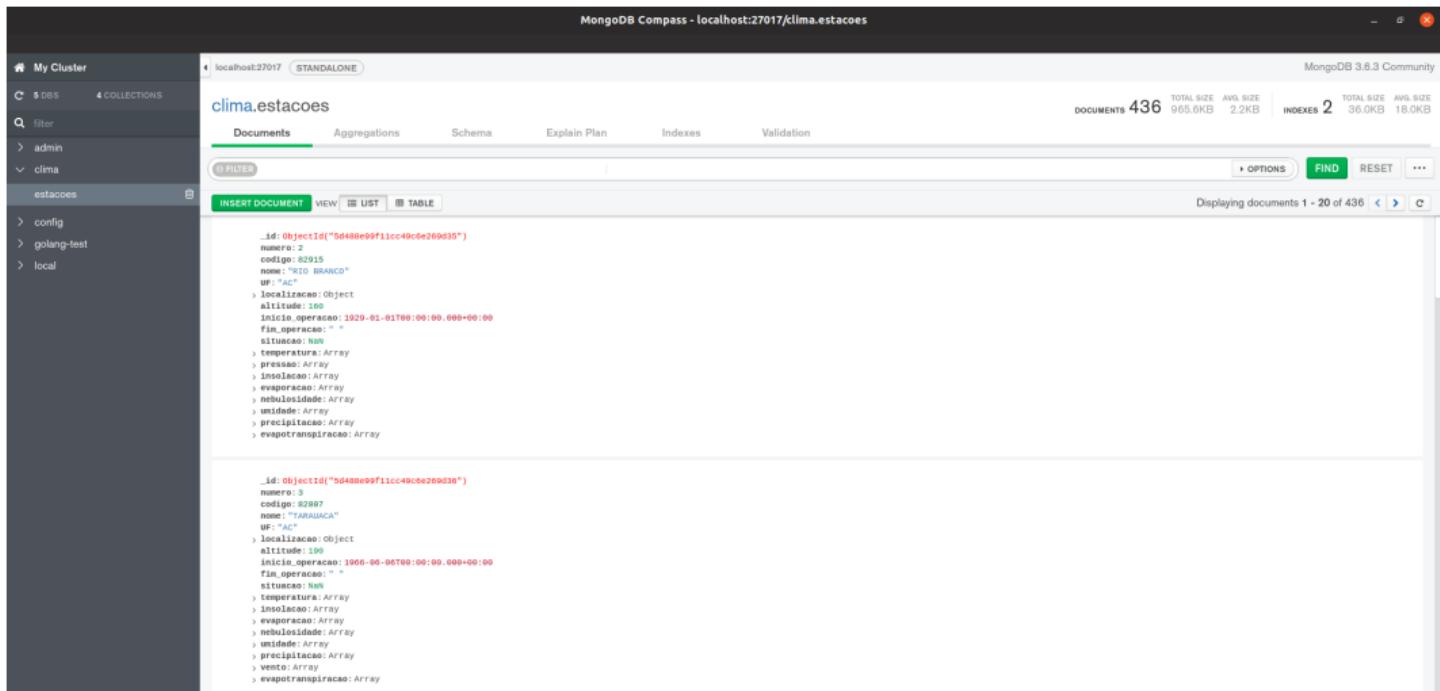
INSERT DOCUMENT VIEW LIST TABLE

FIND RESET ...

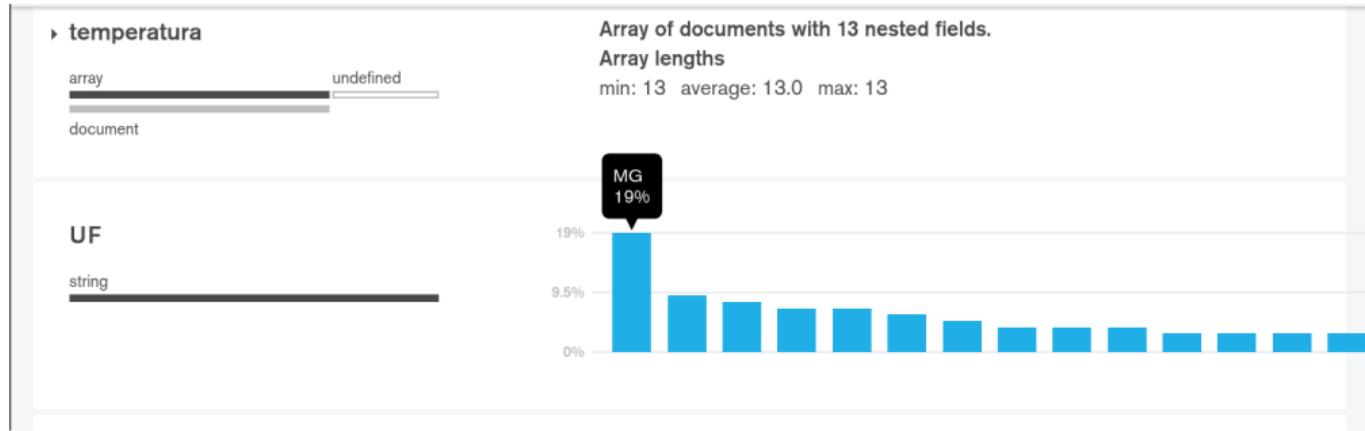
Displaying documents 1 - 20 of 436

`_id: ObjectId("5d488e99f11cc49c8e289d35")  
numero: 2  
codigo: 82915  
nome: "RIO BRANCO"  
UF: "AC"  
> localizacao: Object  
 altitude: 100  
 inicio_operacao: 1929-01-01T00:00:00.000+00:00  
 fim_operacao: "  
 situacao: NaN  
> temperatura: Array  
 > pressao: Array  
 > insolação: Array  
 > evaporação: Array  
 > nebulosidade: Array  
 > umidade: Array  
 > precipitação: Array  
 > evapotranspiração: Array`

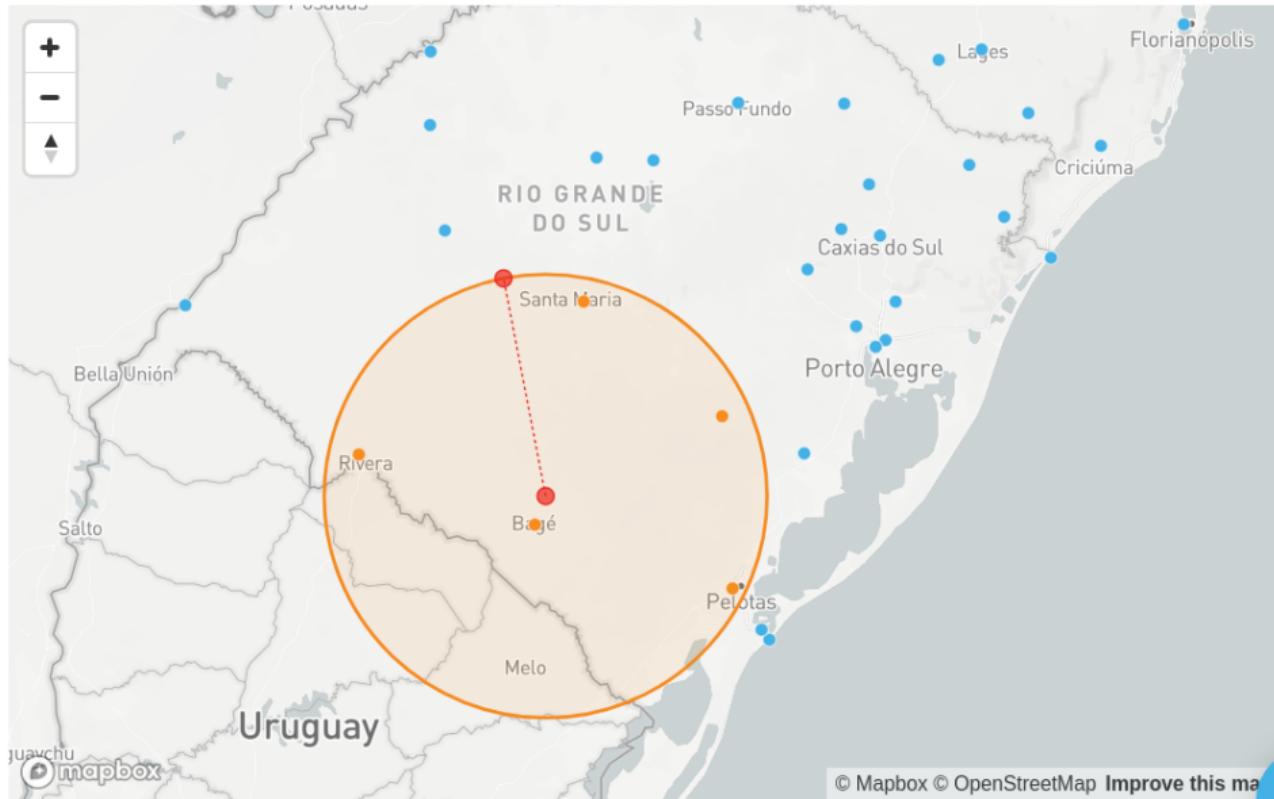
`_id: ObjectId("5d488e99f11cc49c8e289d36")  
numero: 3  
codigo: 82887  
nome: "PARAUÁCA"  
UF: "AC"  
> localizacao: Object  
 altitude: 190  
 inicio_operacao: 1966-06-05T00:00:00.000+00:00  
 fim_operacao: "  
 situacao: NaN  
> temperatura: Array  
 > pressao: Array  
 > evaporação: Array  
 > nebulosidade: Array  
 > umidade: Array  
 > precipitação: Array  
 > vento: Array  
 > evapotranspiração: Array`



# Análise de dados



# Consulta \$geoWithin no Compass



2

<sup>2</sup>Estações meteorológicas do INMET. (script python aqui.)

# Coisas legais

- Mongo Mobile
- Aggregation framework
- MongoDB University
- MongoDB Atlas
- Documentação e comunidade



# Próximos Passos

- Aprenda a notação do MongoDB
- Crie um CRUD usando o driver da sua linguagem favorita
- Dê uma olhada nessa base de dados e no código
- Aplicação Android utilizando GPS, MongoDB e a API Google Maps



# Próximos Passos

- Aprenda a notação do MongoDB
- Crie um CRUD usando o driver da sua linguagem favorita
- Dê uma olhada nessa base de dados e no código
- Aplicação Android utilizando GPS, MongoDB e a API Google Maps
- Junte-se ao **Tchelinux**
- **Compartilhe o seu conhecimento!**



# Onde me encontrar

## Obrigado pela sua atenção!



Ricardo Robaina

- ricardorobaina11@gmail.com
- <https://github.com/robainaricardo>
- <https://www.linkedin.com/robainaricardo>
- @robainaricardo
- @robainaricardo