

Be



MongoDb

5 horas

Objetivo

Criar um Instagram:

- Web
- Desktop
- Mobile

Utilizando as seguintes tecnologias:

- MongoDB
- Express
- Angular
- Node.js

Mas não só isso, pois também teremos esses bancos no módulo de NoSQL:

- ElasticSearch
- OrientDB
- Neo4J
- Redis

E na parte de Front:

- Atomic Design
- Mobile-first
- Offline-first
- Stylus

Fora os módulos de:

- ES6
- Deploy
- Desktop
- Segurança
- UX & Animations

SIM!

Tudo incluso!

**Além disso o mini-curso
de Geolocation com
MongoDB**

Também já está OPEN

**Pois todas vagas que
existiam foram
“compradas”**

Arquitetura



node express



NoSQL



Pense em cervejas, cada tipo de cerveja é diferente.

Pense no Relacional como as Pilsens que são as mais comuns achadas em qualquer boteco.



Agora as cervejas NoSQL são as diferentes, cada uma tem suas características marcantes e podem ser agrupadas pelo tipo. E apenas quem possui gosto apurado usa.



**FICA
& DICA!**

**Com bancos relacionais o que importa
são as respostas que você tem.**

**Com os bancos NoSQL
são as perguntas que importam.**

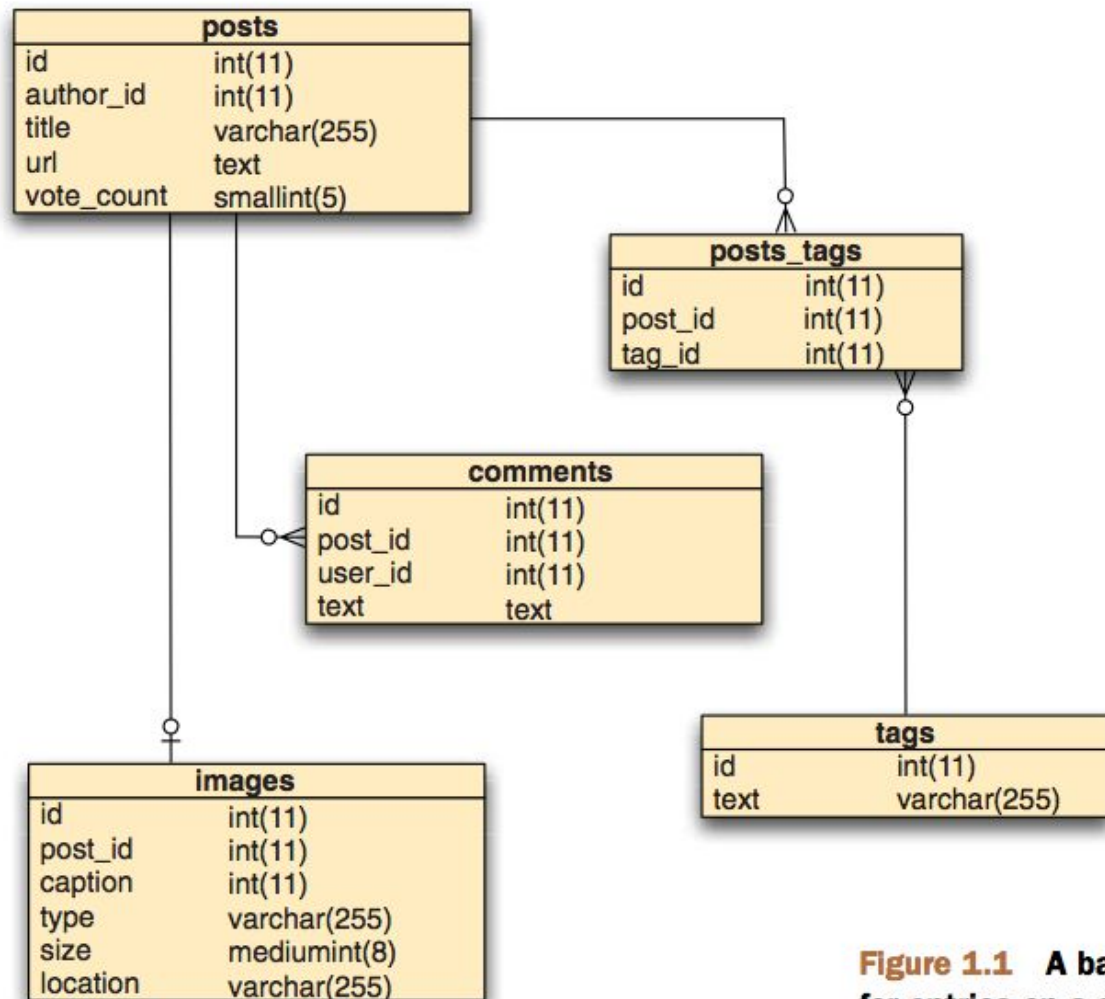


Figure 1.1 A basic relational data model for entries on a social news site

Principais grupos de NoSQL:

- chave/valor
- documento
- grafos
- colunas
- mistos



tetra



externa

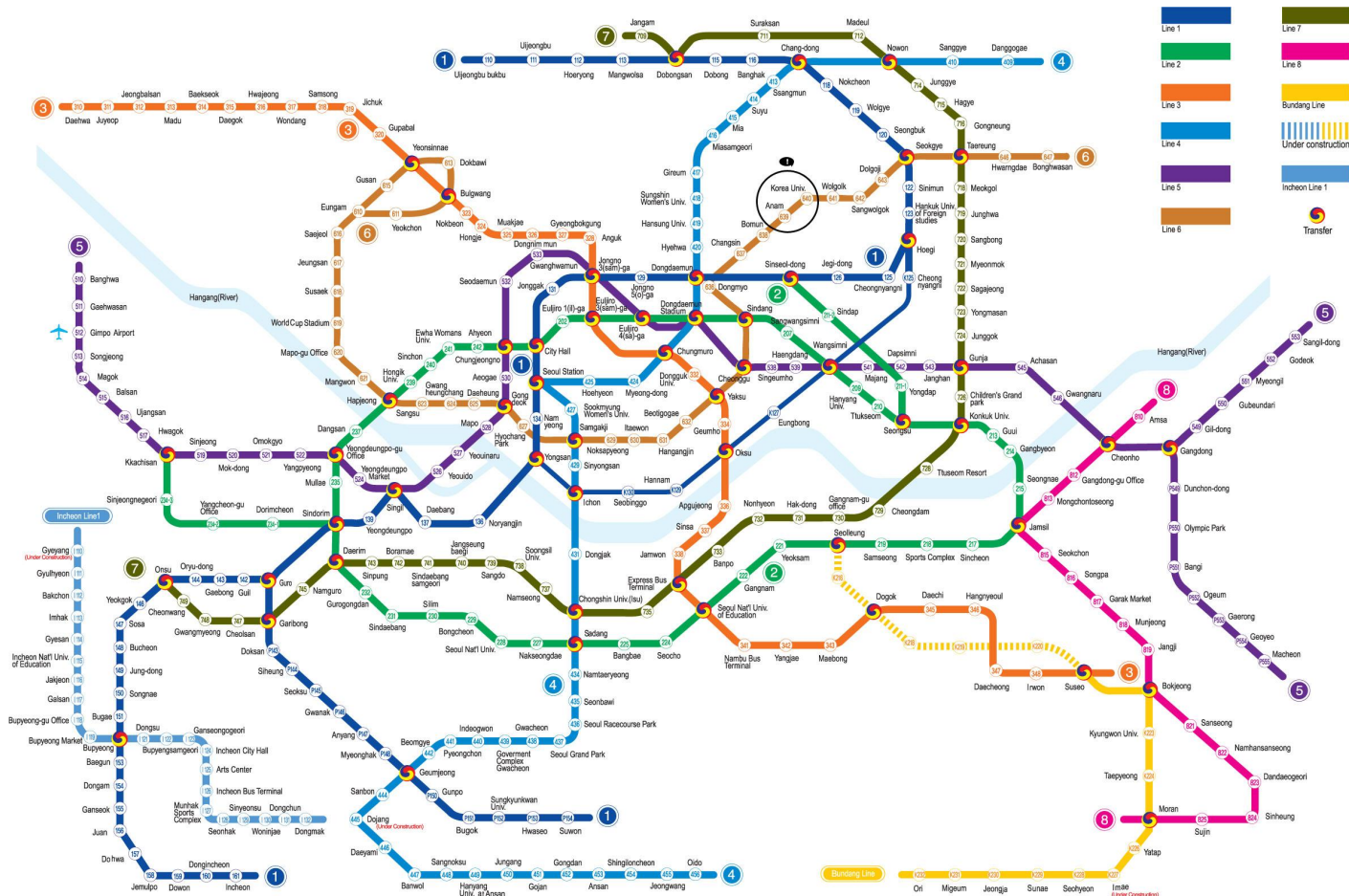


interna



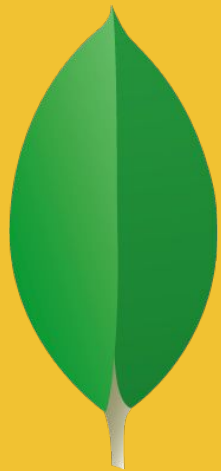
banheiro











mongoDB

Content
Repo

IoT Sensor
Data

Ad Service

Customer
Analytics

Archive

MongoDB Query Language

MongoDB Data Model

MMAP V1

WiredTiger

In-Memory

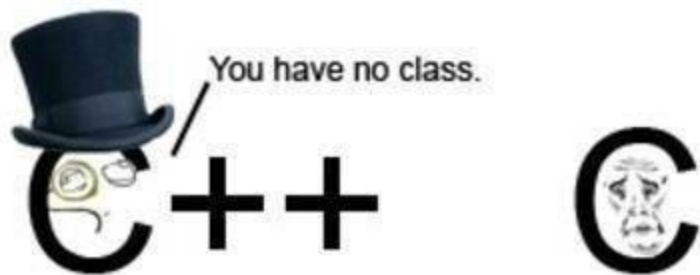
HDFS

Other Storage
Engine

Security

Management

C++



Schemaless

Shemale



JSON/BSON



Replica



Sharding



GridFs



Geolocation



Terminologia

SQL RDBMS MongoDB

DATABASE DATABASE

TABLE COLLECTION

ROWS DOCUMENT JSON

QUERY QUERY

INDEX INDEX

PARTITION SHARD

mongod

Para iniciarmos o MongoDB precisamos rodar o seu serviço, executando o comando:

```
mongod
```

Caso seja sua primeira vez com o MongoDB e ter dado erro e você localizar isso na mensagem de erro:

data/db

Significa que sua pasta data/db não está criada, então por favor crie e de as permissões necessárias.

Exemplo Sistemas Operacionais:

```
sudo mkdir /data
```

```
sudo mkdir /data/db
```

```
sudo chmod 777 -Rf /data
```

mongo

Para conectarmos no servidor do MongoDB usaremos seu cliente, o:

mongo

mongohacker

<https://github.com/TylerBrock/mongo-hacker>

mongoexport

Para exportarmos os dados de uma coleção no MongoDB usaremos o comando:

- mongoexport

```
mongoexport --db nome_do_database --collection  
nome_da_colecao --out minha_colecao.json
```

mongoimport

Para importarmos os dados de um arquivo para uma coleção no MongoDB usaremos o comando:

- mongoimport

```
mongoimport --db database --collection collection --drop --  
file data.json
```

Exercício

https://github.com/Webschool-io/be-mean-instagram/blob/master/apostila/mongodb/export_import.md#exercício

FIM AULA 01

INICIO AULA 02

Exercício RESOLVIDO

use nome_database

Caso você queira escolher uma database na hora de levantar o mongod, basta executá-lo assim:

```
mongo nome_database
```

Então agora é sua hora de mudar seu banco de dados, vamos todos usar a mesma base:

use be-mean-instagram

Perceba que logo abaixo você receberá uma mensagem assim:

switched to db be-mean-instagram

Isso significa que agora nossa variável `db` está aprontando para nosso database, para você verificar basta digitar `db` e apertar o ENTER:

```
suissacorp(mongod-3.0.6) be-mean-instagram> db  
be-mean-instagram
```

show dbs

O comando **show dbs** lista as databases criadas, porém perceba como está a sua listagem após o comando, provavelmente parecida com isso:

local	0.078GB
-------	---------



Isso acontece porque o MongoDB só irá criar sua database quando você tiver inserido algo nela, para inserirmos algo faremos da seguinte maneira:

```
db.teste.insert({nome: "Suissa", idade: 30})
```

```
Inserted 1 record(s) in 1854ms
```

```
WriteResult({  
  "nInserted": 1  
})
```

Agora quando você listar suas databases verá a que acabamos de criar.

```
show dbs
```

```
be-mean-instagram      0.078GB
```

```
local                  0.078GB
```

show collections

Esse comando é o responsável por listar as coleções da sua database especificada com use.

```
show collections  
system.indexes  
teste
```


Caso deseje criar uma coleção vazia você deverá usar o comando:

```
db.createCollection()
```

<https://docs.mongodb.org/v3.0/reference/method/db.createCollection/#db.createCollection>

Então como você pode ter notado o MongoDB criou uma coleção no momento em que foi inserido um JSON nela.

Você deve ter notado que teve uma demora significativa na primeira inserção, isso acontece porque o MongoDB está pré-alocando espaço no seu HD.

Para ver como agora qualquer inserção será veloz, insira mais o seguinte objeto:

```
var json = {escola: 'Webschool', active: true}  
db.teste.insert(json)
```

Inserted 1 record(s) in 1ms

```
WriteResult({  
  "nInserted": 1  
})
```

Antes demorou 1854ms e agora 1ms.

Bom né? :p

db.insert()
Create

Vamos iniciar nossa coleção de Pokemons inserindo o Pikachu como primeiro.

```
var pokemon = {'name':'Pikachu','description':'Rato elétrico bem fofinho','type':  
'electric', attack: 55, height: 0.4 }  
db.pokemons.insert(pokemon)
```

Inserted 1 record(s) in 3ms

```
WriteResult({  
  "nInserted": 1  
})
```

Para você conferir seu pokemon na sua coleção basta usar o comando find():

```
db.pokemons.find()
```

```
{  
  "_id": ObjectId("5642105678a516628da8d8d5"),  
  "name": "Pikachu",  
  "description": "Rato elétrico bem fofinho",  
  "type": "electric",  
  "attack": 55,  
  "height": 0.4  
}
```

Fetches 1 record(s) in 3ms

Agora insira mais esses 3 pokemons:

```
{'name':'Bulbassauro','description':'Chicote de trepadeira','type': 'grama', 'attack':  
49, height: 0.4 }
```

```
{'name':'Charmander','description':'Esse é o cão chupando manga de  
fofinho','type': 'fogo', 'attack': 52, height: 0.6 }
```

```
{'name':'Squirtle','description':'Ejeta água que passarinho não bebe','type':  
'água', 'attack': 48, height: 0.5 }
```


Dados dos Pokemons:

<http://www.pokemon.com/br/pokedex/pikachu>

<http://pokeapi.co/api/v1/pokemon/25/>

<http://www.pokemon.com/br/pokedex/bulbasaur>

<http://pokeapi.co/api/v1/pokemon/1/>

<http://www.pokemon.com/br/pokedex/charmander>

<http://pokeapi.co/api/v1/pokemon/4/>

<http://www.pokemon.com/br/pokedex/squirtle>

<http://pokeapi.co/api/v1/pokemon/7/>

Depois de inserido todos os pokemons agora basta listá-los:

```
db.pokemons.find()
```

```
mongod  mongod...  mongo  repo-
https://docs.google.com/presentation/d/1K0Xmcwa
Be MEAN Instagram - MongoDB
db.pokemons.find()
{
  "_id": ObjectId("564220f0613f89ac53a7b5d0"),
  "name": "Pikachu",
  "description": "Rato elétrico bem fofinho",
  "type": "electric",
  "attack": 100,
  "height": 0.4
}
{
  "_id": ObjectId("56422345613f89ac53a7b5d1"),
  "name": "Bulbassau",
  "description": "Chicote de trepadeira",
  "type": "grama",
  "attack": 49,
  "height": 0.4
}
{
  "_id": ObjectId("56422345613f89ac53a7b5d2"),
  "name": "Charmander",
  "description": "Esse é o cão chupando manga de fofinho",
  "type": "fogo",
  "attack": 52,
  "height": 0.6
}
{
  "_id": ObjectId("56422345613f89ac53a7b5d3"),
  "name": "Squirtle",
  "description": "Ejeta água que passarinho não bebe",
  "type": "água",
  "attack": 48,
  "height": 0.5
}
db.update()
Update
Fetches 4 record(s) in 2ms
suissacorp(mongod-3.0.6) be-mean-instagram>
```

Bonito meu find() né?

VIVA O MONGO-HACKER!!!!

db.save()

Inserire e Salva

```
var pokemon = {'name':'Caterpie','description':'Larva lutadora','type': 'inseto',  
attack: 30, height: 0.3 }
```

```
suissacorp(mongod-3.0.6) be-mean-instagram> db.pokemons.save(pokemon)
```

```
Inserted 1 record(s) in 1ms
```

```
WriteResult({  
  "nInserted": 1  
})
```

```
var query = {name: 'Caterpie'}
suissacorp(mongod-2.4.8) be-mean> var p = db.pokemons.find(query)
suissacorp(mongod-3.0.6) be-mean-instagram> p
{
  "_id": ObjectId("56422705613f89ac53a7b5d4"),
  "name": "Caterpie",
  "description": "Larva lutadora",
  "type": "inseto",
  "attack": 30,
  "height": 0.3
}
Fetched 1 record(s) in 1ms
suissacorp(mongod-3.0.6) be-mean-instagram> p.name
suissacorp(mongod-3.0.6) be-mean-instagram>
```

Não conseguimos acessar diretamente nosso objeto pois ele é retornado na forma de [Cursor](#), que possui métodos especiais para acessar seus valores, [como visto aqui](#).

Então precisamos utilizar o **findOne** pois ele retorna um objeto comum.

```
var p = db.pokemons.findOne(query)
suissacorp(mongod-3.0.6) be-mean-instagram> p
{
  "_id": ObjectId("56422705613f89ac53a7b5d4"),
  "name": "Caterpie",
  "description": "Larva lutadora",
  "type": "inseto",
  "attack": 30,
  "height": 0.3
}
```

```
suissacorp(mongod-3.0.6) be-mean-instagram> p.name
```

Caterpie

```
suissacorp(mongod-3.0.6) be-mean-instagram> p.defense = 35
```

35

```
suissacorp(mongod-3.0.6) be-mean-instagram> p
```

```
{
```

```
  "_id": ObjectId("56422705613f89ac53a7b5d4"),
```

```
  "name": "Caterpie",
```

```
  "description": "Larva lutadora",
```

```
  "type": "inseto",
```

```
  "attack": 30,
```

```
  "height": 0.3,
```

```
  "defense": 35
```

```
}
```

```
suissacorp(mongod-3.0.6) be-mean-instagram> db.pokemons.save(p)
```

```
Updated 1 existing record(s) in 2ms
```

```
WriteResult({  
  "nMatched": 1,  
  "nUpserted": 0,  
  "nModified": 1  
})
```

Cursor

```
var cur = db.pokemons.find();  
while( cur.hasNext() ) { print(tojson(cur.next()))};
```

Exercício

1. Crie uma database chamada be-mean-pokemons;
2. Liste quais databases você possui nesse servidor;
3. Liste quais coleções você possui nessa database;
4. Insira pelo menos 5 pokemons **A SUA ESCOLHA** utilizando o mesmo padrão de campos utilizado: name, description, attack, defense e height;
5. Liste os pokemons existentes na sua coleção;
6. Busque o pokemons a sua escolha, pelo nome, e armazene-o em uma variável chamada `poke`;
7. Modifique sua `description` e salvê-o

Como enviar

Veja na nossa Wiki.

FIM AULA 02

db.find()
Retrieve

db.update()
Update

db.remove()
Delete