

{reprograma}

Introdução a Banco de Dados

{reprograma}

- Amo gatos;
- Pisciana com Ascendente em Câncer;
- Gosto de estudar sobre coisas relacionadas e não relacionadas a área de programação;
- Fã de Rupaul's Drag Race;
- Aprendendo a lidar com a vida financeira;
- <http://devas.herokuapp.com>

Nas redes sociais:
mariliamessias



{reprograma}



Apresentação

- Nome;
- Área atual;
- O que fez querer participar do curso;
- O que mais está tendo dificuldades;
- De onde você é?
- Se você fosse um site/app, qual você seria?

{reprograma}

Relembrar é viver!

The Kahoot! logo is displayed on a dark purple rectangular background. The word "Kahoot!" is written in a bold, white, sans-serif font. A lighter purple triangle is positioned behind the text, pointing towards the top right corner of the rectangle.

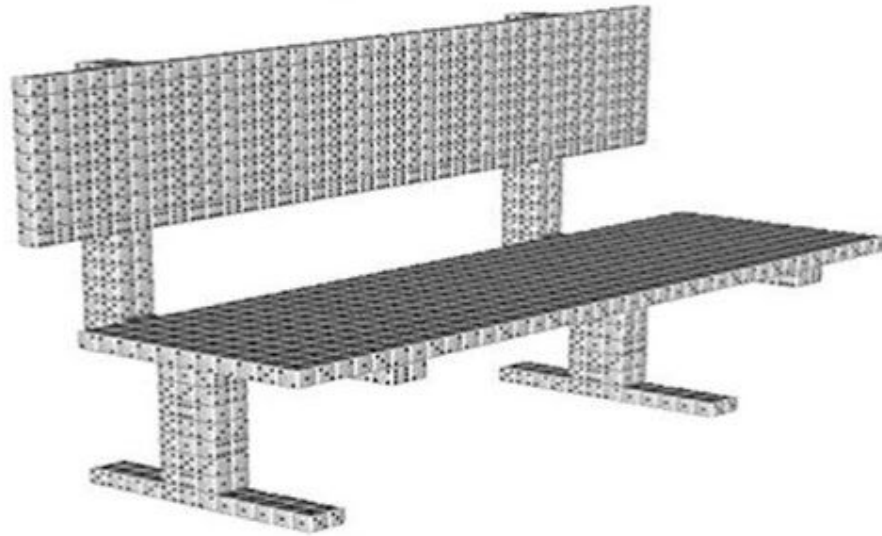
Kahoot!

PIN: ###

Link: <https://kahoot.com/>

{reprograma}

O que é banco de
Dados?

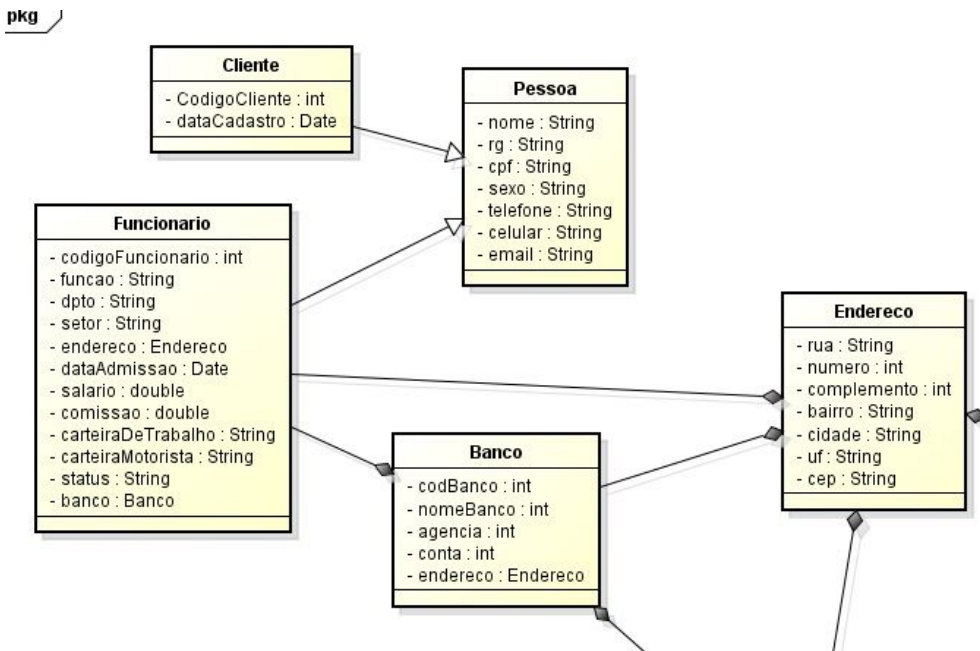


{reprograma}

O que é banco de Dados?

Coleção organizada de dados;

Abstração de algum aspecto da vida real com alguma finalidade específica;



Fonte: <https://www.filepicker.io/api/file/yPEOaImpSZ2gAKXsORnk>

{reprograma}

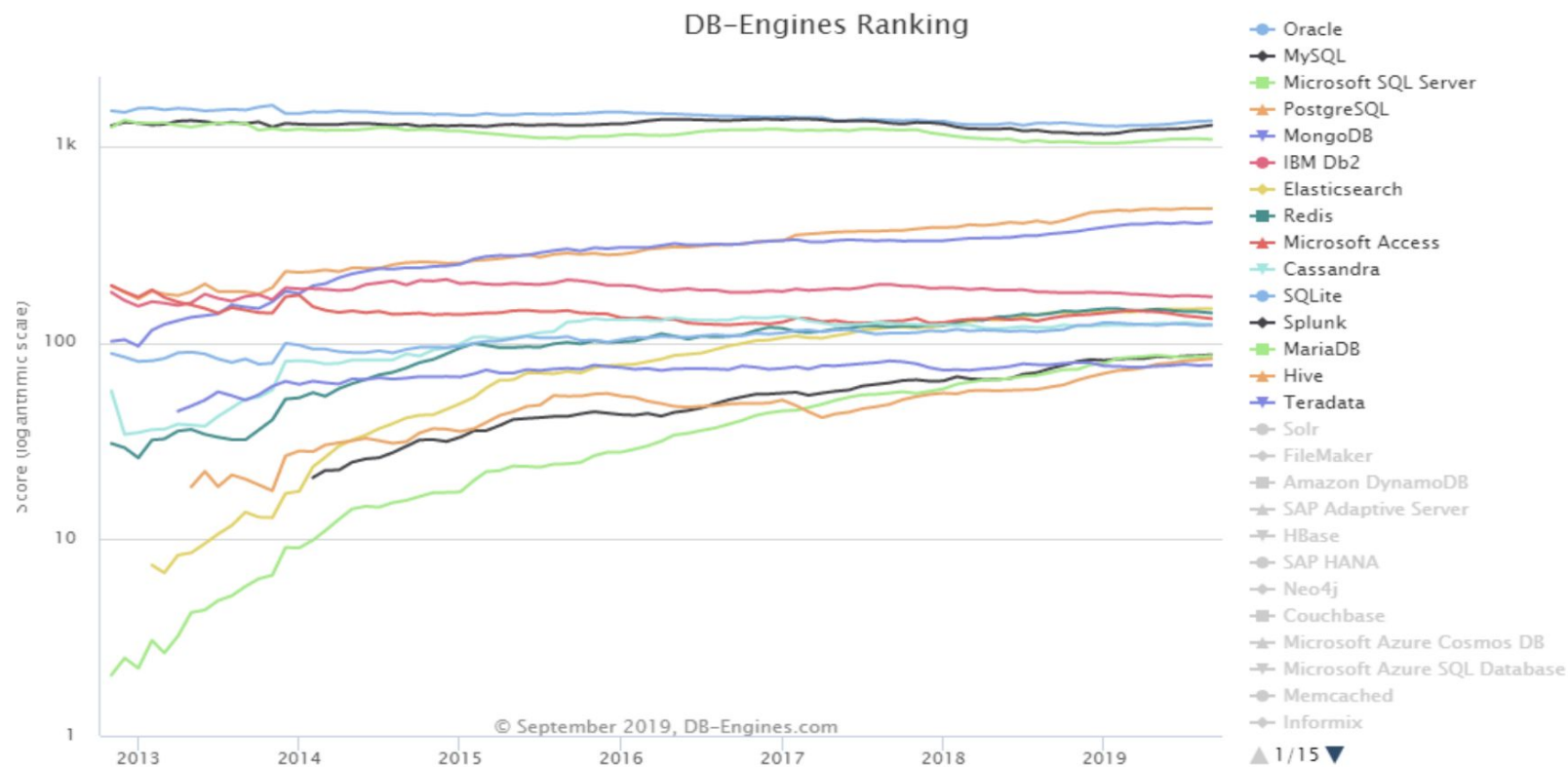
Entidades & Atributos

Quais entidades e atributos você identifica em um **caixa eletrônico** de um banco?

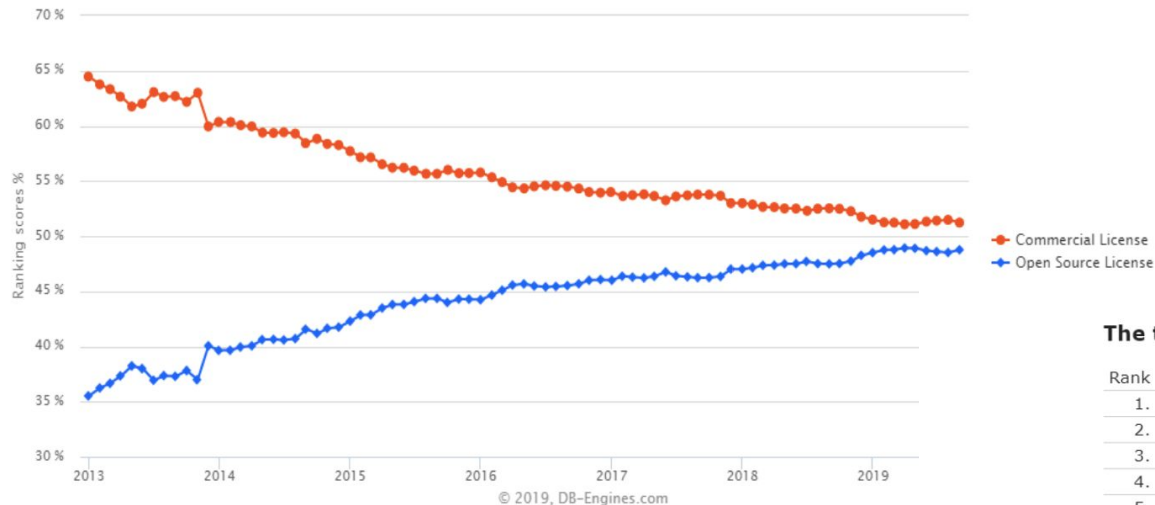


Os Sistemas de Gerenciamento de Banco de Dados são softwares utilizados para gerir as estruturas de armazenamento dos dados, permitem realizar manipulações, bem como controlar as permissões de utilização dos bancos de dados.





Popularity trend



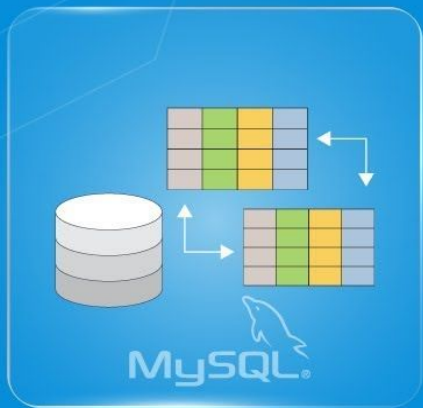
The top 5 commercial systems, September 2019

Rank	System	Score	Overall Rank
1.	Oracle	1347	1.
2.	Microsoft SQL Server	1085	3.
3.	IBM Db2	172	6.
4.	Microsoft Access	133	9.
5.	Splunk	87	12.

The top 5 open source systems, September 2019

Rank	System	Score	Overall Rank
1.	MySQL	1279	2.
2.	PostgreSQL	482	4.
3.	MongoDB	410	5.
4.	Elasticsearch	149	7.
5.	Redis	142	8.

{reprograma}



SQL

VS

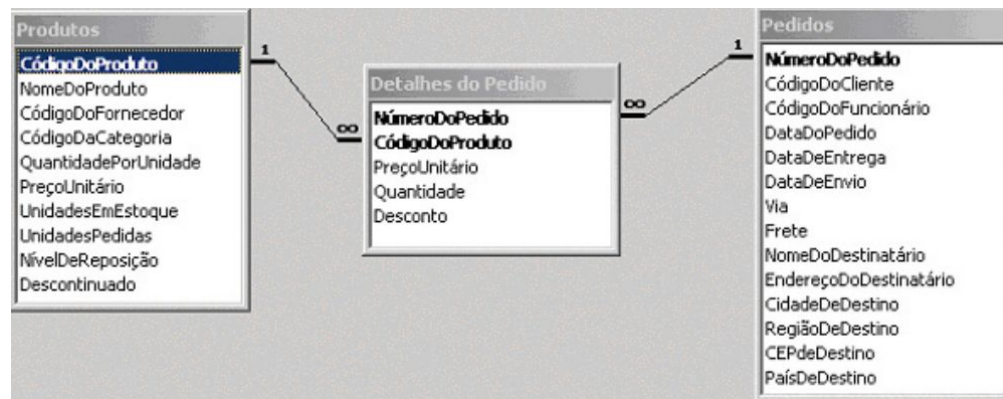


NoSQL

{reprograma}

SQL

Baseia-se em um modelo relacional em tabelas (linhas e colunas).
Através desse modelo, pode-se manipular dados (atualizar, adicionar, remover, etc.), realizar transações, bem como gerenciar o banco de dados.



	ID	First name	Last name
1	101	Danny	Young
2	102	Christine	Holand
3	103	Lars	Gordon
4	104	Roberto	Robitaille
5	105	Maria	Papadopoulos

	ID	First name	Last name
1	101	Danny	Young
2	102	Christine	Holand
3	103	Lars	Gordon
4	104	Roberto	Robitaille
5	105	Maria	Papadopoulo

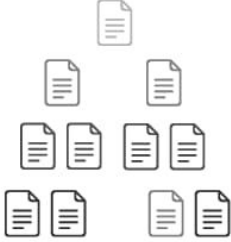
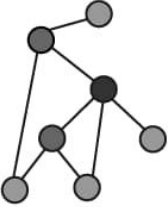
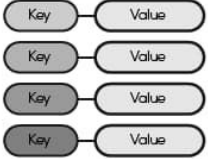
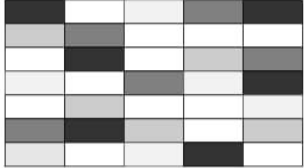
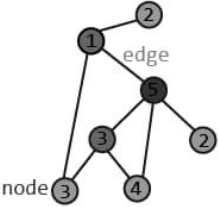
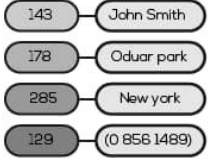
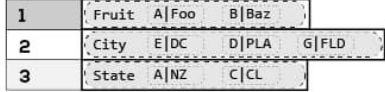




São bancos de dados dinâmicos, recomendados para utilização de dados não estruturados, além disso, o dado pode ser armazenado em estruturas independentes.

A partir desse princípio, permite que sejam criados documentos sem a necessidade de criação de sua estrutura previamente, cada documento pode conter características únicas, a sua sintaxe varia de uma base de dados para outra.

Key	Document
1001	<pre>{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }</pre>
1002	<pre>{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }</pre>

{reprograma}

Tipos de Banco NoSQL

Document	Graph	Key-Value	Wide-Column
			
<pre>{ "user": { "id": "143", "name": "improgrammer", "city": "New York" } }</pre>			
			

{reprograma}

Revisão



- O que é banco de Dados;
- Entidades;
- Atributos;
- SGDBs;
- SQL;
- NoSQL;

1 - Uma empresa de seguros precisa criar um sistema novo que permita com que os **funcionários** possam se cadastrar para receber a **cesta de natal**. Os funcionários utilizam como identificação um **número de matrícula, seu nome, sobrenome, email e cargo**. Além disso, nesse sistema será necessário cadastrar também as **cestas de natal**, que tem como características o **tipo (basic, gold, e diamond)**, a **data de vencimento**, a **data de entrega** e o **custo**. Identifique as entidades e os atributos.

2 - Com suas palavras explique o que é um banco de dados NoSQL.

{reprograma}

Introdução ao MongoDB

- NoSQL;
- Orientado a objetos (documentos);
- Open-source;
- BSON.

What Is MongoDB?

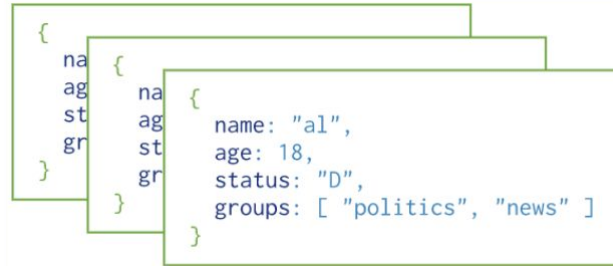
MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need

{reprograma}

MongoDB

```
{  
  name: "al",  
  age: 18,  
  status: "D",  
  groups: [ "politics", "news" ]  
}
```

Documento



Collections (Coleções)



Banco de Dados




<https://www.mongodb.com/try/download/community>

MongoDB Community Server

MongoDB offers both an Enterprise and Community version of its powerful distributed document database. The community version offers the flexible document model along with ad hoc queries, indexing, and real time aggregation to provide powerful ways to access and analyze your data. As a distributed system you get high availability through built-in replication and failover along with horizontal scalability with native sharding.

The MongoDB Enterprise Server gives you all of this and more. Review the Enterprise Server tab to learn what else is available.

Available Downloads

Version	4.4.1 (current) 
Platform	Windows 
Package	msi 



Download

Copy Link

Current releases & packages

Development releases

Archived releases

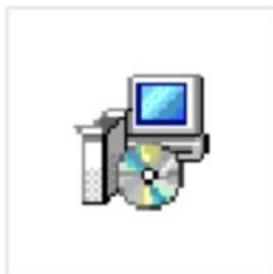
Changelog

Release Notes

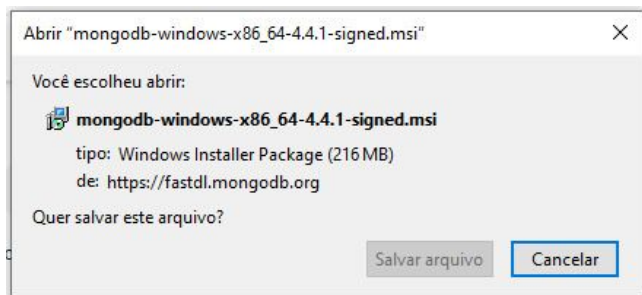
{reprograma}

Instalação - Windows

Arquivo:



mongodb-win32
-x86_64-2012plus
-4.2.0-signed.msi

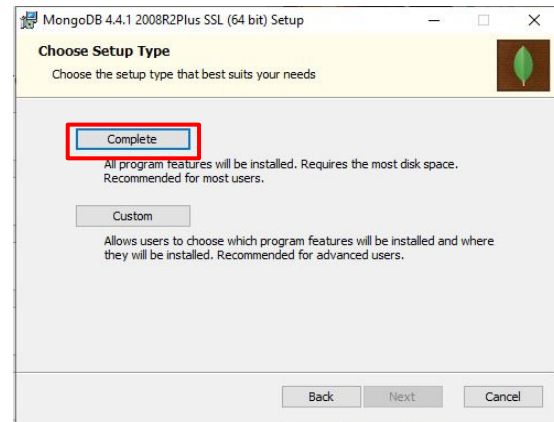
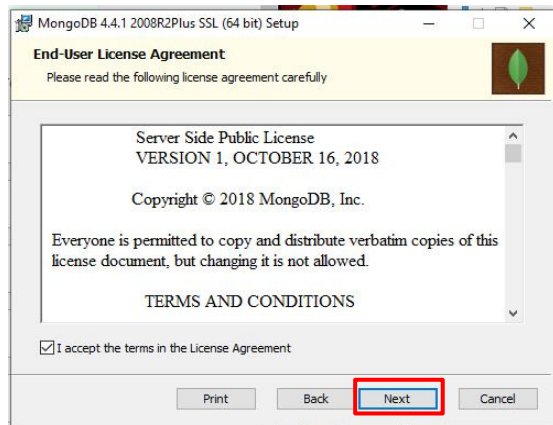
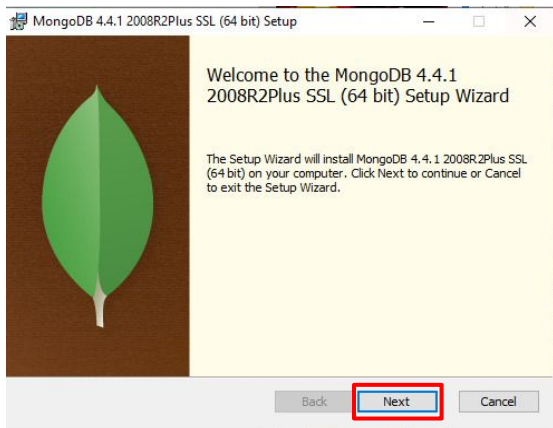


O diretório padrão de instalação
é: C:\Program
Files\MongoDB\Server\{versão
instalada}

Obs.: esse diretório default é
automaticamente
criado durante a instalação.

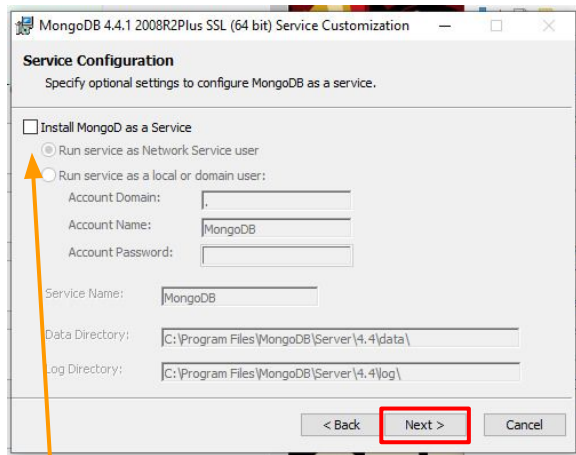
{reprograma}

Instalação - Windows

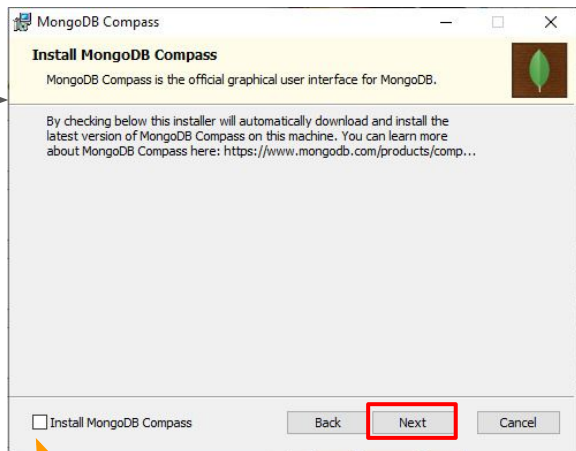


{reprograma}

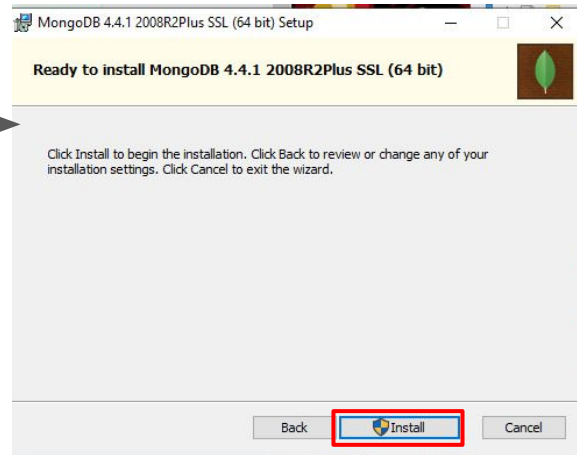
Instalação - Windows



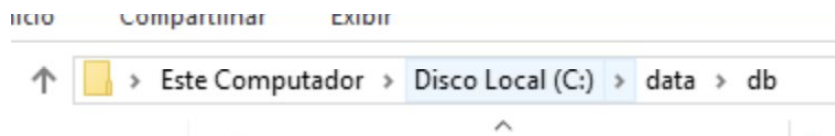
Desmarcar a opção: Install MongoDB as a Service, pois não temos nenhum servidor existente, iremos utilizar a nossa máquina como servidor por enquanto.



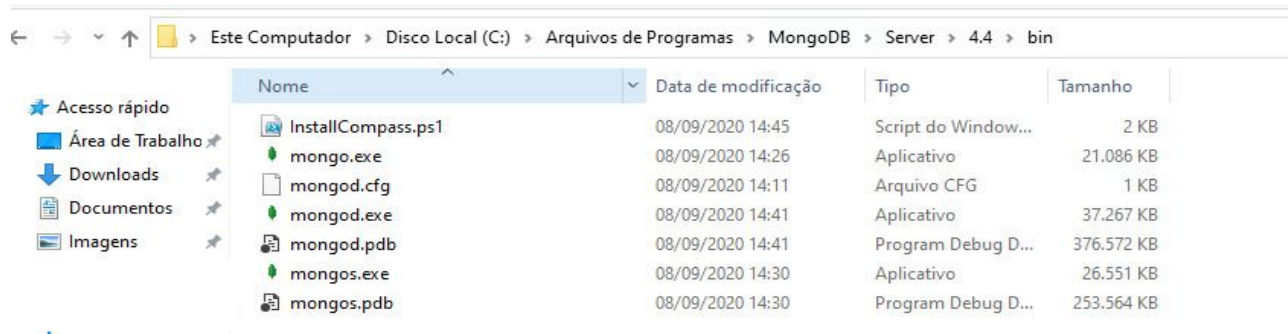
Desmarcar a opção Install MongoDB Compass.



O diretório padrão para armazenamento dos bancos de dados é "C:\data\db". **É necessário criá-lo.**



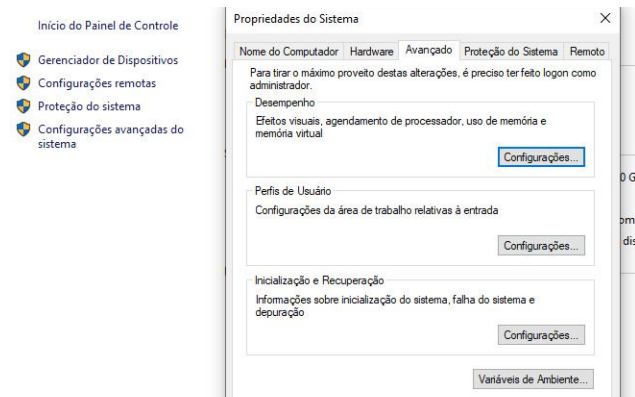
Repositório onde foi instalado o MongoDB:



{reprograma}

Instalação - Windows

Acessar as variáveis de ambiente para configurar o mongo instalado.



Necessário configurar a variável **PATH** com o mongo para reconhecimento no terminal do Windows.

C:\Program Files\MongoDB\Server\4.2\bin

Informar de acordo com a versão instalada

Comando para inicializar o server: mongod

```
CA: Prompt de Comando - mongod
Microsoft Windows [versão 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Lontrosos>mongod
2019-09-15T22:24:27.464-0300 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabl
2019-09-15T22:24:27.991-0300 I CONTROL [initandlisten] MongoDB starting : pid=4756 port=27017 dbpath=C:\data\db\ 64-bit ho
2019-09-15T22:24:27.991-0300 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2019-09-15T22:24:27.991-0300 I CONTROL [initandlisten] db version v4.2.0
2019-09-15T22:24:27.992-0300 I CONTROL [initandlisten] git version: a4b751dcf51dd249c5865812b390cfd1c0129c30
2019-09-15T22:24:27.992-0300 I CONTROL [initandlisten] allocator: tcmalloc
2019-09-15T22:24:27.992-0300 I CONTROL [initandlisten] modules: none
2019-09-15T22:24:27.992-0300 I CONTROL [initandlisten] build environment:
```

MacOS:

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/>

Linux:

<https://docs.mongodb.com/manual/administration/install-on-linux/>

{reprograma}

Comando para criar um banco de dados novo:

use reprograma

```
---  
> use reprograma  
switched to db reprograma  
> █
```

Comandos

Comando para listar todos os bancos de dados:

show databases

```
> show databases  
admin          0.000GB  
config         0.000GB  
local          0.000GB  
reprograma     0.000GB  
█
```

{reprograma}

Comandos

Remover banco de dados atual:

db.dropDatabase()

```
> db.dropDatabase()
{ "dropped" : "reprograma", "ok" : 1 }
> █
```

Criar uma Collection:

db.createCollection(<Nome da Collection>)

```
> db.createCollection("collectionReprogramaAlunas")
{ "ok" : 1 }
```

Listar todas as collections:

show collections

```
> show collections
collectionReprograma
collectionReprogramaAlunas
```

Comando para buscar todos os registros de uma collection:

db.<NomeDaCollection>.find()

```
> db.collectionReprograma.find()
{ "_id" : ObjectId("5f88eb2275db2480861c8502"),
  "nome" : "Aluna1", "turma" : "BackEnd" }
```

{reprograma}

Comando para trazer o retorno de uma forma mais amigável:

db.<NomeDaCollection>.find().pretty()

```
> db.collectionReprograma.find().pretty()
{
  "_id" : ObjectId("5f88eb2275db2480861c8502")
,
  "nome" : "Aluna1",
  "turma" : "BackEnd"
}
```

Comandos

Comando que retorna apenas um único registro:

db.<NomeDaCollection>.findOne()

```
> db.collectionReprograma.findOne()
{
  "_id" : ObjectId("5f88eb2275db2480861c8502")
,
  "nome" : "Aluna1",
  "turma" : "BackEnd"
}
```

{reprograma}

Comandos

Comando para incluir um registro dentro de uma collection:

db.<NomeDaCollection>.insertOne({<Atributos>})

```
> db.collectionReprograma.insertOne({"nome": "Aluna1", "turma": "BackEnd"})
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5f88e5a175db2480861c8501")
}
```

Comando para incluir vários registros de uma única vez:

db.<NomeDaCollection>.insertMany([{ // objetos a serem inseridos // }])

```
> db.collectionReprograma.insertMany([ {"nome": "Aluna1", "turma": "BackEnd"}, {
"nome": "Aluna2", "turma": "Backend"}, { "nome": "Aluna3", "turma": "Fullstack"}
])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5f8901e9c2bb53233659edc9"),
    ObjectId("5f8901e9c2bb53233659edca"),
    ObjectId("5f8901e9c2bb53233659edcb")
  ]
}
```

{reprograma}

Filtros

Um filtro é um atributo que iremos utilizar como limitador no resultado da busca, por exemplo:

```
[
  {
    "nome": "Ana",
    "turma": "FullStack",
    "signo": peixes
  },
  {
    "nome": "Carla",
    "turma": "BackEnd",
    "signo": peixes
  }
]
```

```
> db.collectionReprograma.find({"turma": "BackEnd"}).pretty()
{
  "_id" : ObjectId("5f924d402684d1e890f129c6"),
  "nome" : "Carla",
  "turma" : "BackEnd",
  "signo" : "peixes"
}
```

Como ficaria o filtro para retornar as alunas que estudam na turma de Backend:

```
db.collectionReprograma.find({"turma": "BackEnd"}).pretty()
```


{reprograma}

Mais de Um Filtro (Operador AND)

Para este mesmo cenário, gostaria de filtrar mais um atributo ao mesmo tempo e o retorno deve respeitar os filtros informados ao mesmo tempo.

```
[
  {
    "nome": "Ana",
    "turma": "FullStack",
    "signo": peixes
  },
  {
    "nome": "Carla",
    "turma": "BackEnd"
    "signo": peixes
  }
]
```

```
> db.collectionReprograma.find({"turma": "BackEnd", "signo": "peixes"}).pretty()
{
  "_id" : ObjectId("5f924d402684d1e890f129c6"),
  "nome" : "Carla",
  "turma" : "BackEnd",
  "signo" : "peixes"
}
```

Como ficaria o filtro para retornar as alunas que estudam na turma de Backend:

```
db.<NomeDaCollection>.find({"turma": "BackEnd", "signo": "peixes"}).pretty()
```

{reprograma}

Mais de Um Filtro (Operador OR)

Para este mesmo cenário, gostaria de filtrar mais um atributo ao mesmo tempo e respeitar ao menos um dos filtros.

```
[
  {
    "nome": "Ana",
    "turma": "FullStack",
    "signo": peixes
  },
  {
    "nome": "Carla",
    "turma": "BackEnd"
    "signo": peixes
  }
]
```

```
> db.collectionReprograma.find({ $or: [{"turma": "BackEnd"}, {"signo": "peixes"}]
}).pretty()
{
  "_id" : ObjectId("5f924d402684d1e890f129c5"),
  "nome" : "Ana",
  "turma" : "FullStack",
  "signo" : "peixes"
}
{
  "_id" : ObjectId("5f924d402684d1e890f129c6"),
  "nome" : "Carla",
  "turma" : "BackEnd",
  "signo" : "peixes"
}
```

Como ficaria o filtro para retornar as alunas que estudam na turma de Backend:

```
db.<NomeDaCollection>.find({ $or : [ { "turma": "BackEnd" }, { "signo": "peixes" } ]})
```

{reprograma}

Limit

Uso do limit para limitar a quantidade de registros retornados na consulta:

```
[  
  {  
    "nome": "Ana",  
    "turma": "FullStack",  
    "signo": peixes  
  },  
  {  
    "nome": "Carla",  
    "turma": "BackEnd",  
    "signo": peixes  
  }  
]
```

```
> db.collectionReprograma.find().limit(1).pretty()  
{  
  "_id" : ObjectId("5f924d402684d1e890f129c5"),  
  "nome" : "Ana",  
  "turma" : "FullStack",  
  "signo" : "peixes"  
}
```

Uso do Limit:

```
db.<NomeDaCollection>.find().limit(1).pretty()
```

{reprograma}

Revisão



- use database;
- show databases;
- db.dropDatabase;
- createCollection;
- show collections;
- find;
- findOne;
- insertMany
- insertOne;