**MongoDB com mongoose no node**

**INSTALANDO O MONGO NO WINDOWS**

1. Primeiramente vamos realizar a instalação do mongodb.
2. Na página [www.mongodb.com](http://www.mongodb.com) vá em Community server e localize a versão do Windows e baixe o instalador msi.
3. Execute o instalador clique em :
4. Next
5. Marque a opção I accept the terms.....
6. Clique em next
7. Clique em complete
8. Marque a opção Install MongoD as a servisse
9. Clique em next
10. Desmarque a opção Install MongoDB compass
11. Clique em next
12. Clique em Install
13. Após a instalação vamos adicionar o executável no path do Windows
14. Para isso vamos copiar o caminho da pasta bin da instalação do mongodb.
15. Ex: C:\Program Files\MongoDB\Server\5.0\bin
16. Ir em variáveis de ambiente e adicionar o caminho copiado em path.
17. Vá no cmd do Windows e verifique se o mongo está rodando.

**COMANDOS DO MONGODB**

1. Show dbs – para visualizar os banco de dados.
2. use e nome do banco de dados – criar um banco de dados
3. db – mostra o banco de dados selecionado
4. vamos criar o banco de dados aprendendomongodb
5. digite: use aprendendomongodb.

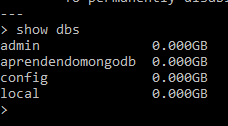
**INSTALANDO A BIBLIOTECA MONGOOSE**

1. o mongoose pode ser utilizado da mesma forma do sequealize.
2. Crie uma pasta para o projeto e inicie o npm.
3. Digite npm init para um novo projeto.
4. Instale o mongoose digite:
5. Npm install mongoose –save
6. Crie um arquivo index.js
7. Utilize o mongoose do index.js
8. Const mongoose = require(‘mongoose’);
9. Depois disso vamos realizar a conexão.
10. Digie: mongoose.connect(“mongodb://localhost:27017/aprendendomongodb”)

**CRIAÇÃO DO ESQUEMA MODEL E O MODEL**

1. Vamos criar a pasta model e um model chamado article.js
2. Importamos o moongose e criamos uma variável articleModel,
3. Que terá o esquema da tabela article que será criada.
4. *const* mongoose = require('mongoose');
5. *const* articleModel = new mongoose.Schema({
6. title: string,
7. author : string,
8. body: string,
9. date: {type: *Date*, Date.now},
10. especial: *Boolean*,
11. resume: {
12. content: string,
13. author: string
14. }
15. });
16. module.exports = articleModel;
17. Para salvar dados em uma tabela mongo, não é preciso criar a tabela primeiro, quando formos inserir o primeiro registro a tabela será criada automaticamente.
18. Vamos voltar no arquivo index.js e vamos importar o esquema do model
19. *const* mongoose = require('mongoose');
20. *const* articleModel = require('../model/article');
21. mongoose.connect("mongodb://localhost:27017/aprendendomongodb",{useNewUrlParser: true},{useUnifieldTopology:true});
22. Temos também que preparar o model para serem utilizados
23. Criamos uma constante Article que criará um model(‘tabela’) chamado Article, utilizando o esquema montado no articleMode.
24. *const* Article = mongoose.model('Article',articleModel);
25. c

**INSERINDO UM ARTIGO**

1. Criamos uma variável artigo que irá receber uma instânica da tabela Article e será passado para os parâmetros os dados do primeiro artigo.
2. //criando o primeiro artigo
3. *const* artigo = new Article({title: "Primeiro Artigo", author: "Eu mesmo", body:"Não sei mais..."});
4. //salvando o primeiro artigo no banco de dados.
5. artigo.save().then(()*=>*{
6. console.log("Artigo salvo");
7. }).catch(*err* *=>*{
8. console.log(*err*);
9. });
10. Agora no index.js vamos rodar a aplicação e o artigo será salvo
11. Para confirmar se o artigo foi salvo, vamos entrar no cmd do Windows e digitar mongo
12. Será aberto o mongo no prompt
13. Digite show dbs para exibir os bancos de dados criados.
14. 
15. Digite use aprendendomongodb para selecionar o banco.
16. 
17. Digite show collections para mostrar as tabelas
18. 
19. Digite db.articles.find() e será exibido os artigos salvos
20. 
21. C
22. O mongoose trabalha de forma assíncrona e por isso podemos utilizar as promisses then e catch
23. C

**BUSCANDO OS DADOS**

**BUSCANDO TODOS OS REGISTROS**

1. Vamos cadastrar mais alguns artigos.
2. Após isso vamos buscar todos os artigos da tabela article.
3. Utilizando o model Article e o método find vamos montar uma promisse que irá retornar um json com todos os artigos do model articles. Ou um erro caso exista.
4. //buscando todo os artigos
5. Article.find({}).then(*articles* *=>* {
6. console.log(*articles*);
7. }).catch(*err* *=>* {
8. console.log(*err*);
9. })
10. Isso ira carregar os registros

**BUSCANDO REGISTROS ESPECÍFICOS**

1. Para buscar registros específicos temos que incluir uma chave e valor no json do find e a busca irá retornar os valores.
2. //buscando dados espedificos
3. //buscando pelo id
4. Article.find({'\_id': '61f061a16af645ed3acc63bc'}).then(*article* *=>* {
5. console.log(*article*);
6. }).catch(*err* *=>* {
7. console.log(*err*);
8. })
9. //buscando pelo title
10. Article.find({'title': 'Primeiro Artigo'}).then(*article* *=>* {
11. console.log(*article*);
12. }).catch(*err* *=>* {
13. console.log(*err*);
14. })

**BUSCANDO DADOS DE CAMPOS DENTRO DE CAMPOS**

1. //buscando dados de campos que estão dentro de campos
2. Article.find({'resume.author': 'Teste !'}).then(*article* *=>* {
3. console.log(*article*);
4. }).catch(*err* *=>* {
5. console.log(*err*);
6. })
7. C

**BUSCANDO O PRIMEIRO REGISTRO DE UMA CONDIÇÃO**

1. //mostrando o primeiro resultado de uma busca que corresponde a um critério
2. Article.findOne({'author': 'Eu sei 3'}).then(*article* *=>* {
3. console.log(*article*);
4. }).catch(*err* *=>* {
5. console.log(*err*);
6. })
7. Caso o um registro não for encontrado, a resposta será null.