

Curso Node.js e MongoDB

LuizTools

<https://www.youtube.com/watch?v=kP99QO1gp8g>

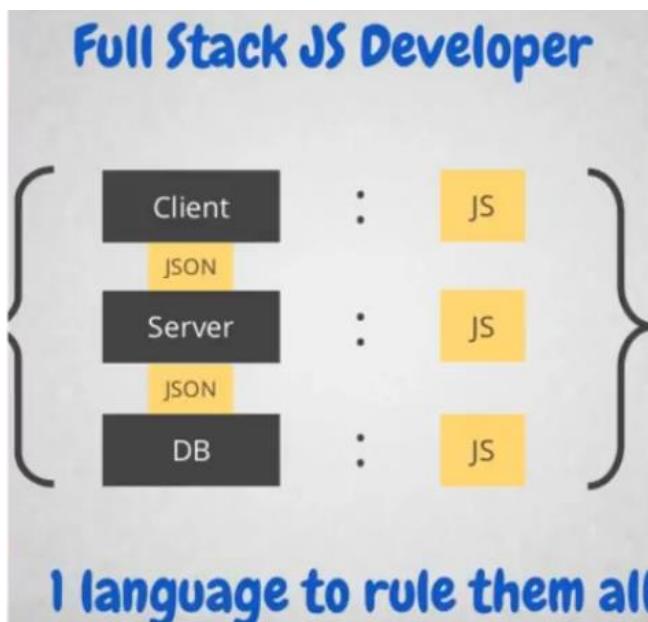
Resumo do curso feito por Roberto Pinheiro

Aula 01 - Por que Node.js?

JavaScript

- Linguagem web mais popular do mundo
- Baixa curva de aprendizado
- Forte comunidade

Full Stack



- JavaScript client-side;
- JavaScript server-side (Node.js);
- JavaScript database (MongoDB);
- dados JSON em todas camadas.

Leve e multiplataforma

- baixo custo de hardware;
- baixo custo de SO;
- sem licença de uso (Open Source e gratuito);

Corporativa

Tecnologia utilizada largamente em escala corporativa

- Netflix e Microsoft;
- GoDaddy e Groupon;
- PayPal e Rakuten;
- SAP e Walmart;
- Yahoo e Ebay;

Ambiente

O Node é um Runtime JS (um interpretador JavaScript).

O Node em si não é uma linguagem de aplicação. O Node é uma plataforma para aplicações. É composta por:

- Runtime JS (V8): motor de javascript. É o mesmo interpretador do Google Chrome.
- IO (libuv): biblioteca de IO (biblioteca de C++) - Para acessar recursos do sistema operacional (escrever no disco da máquina do usuário, fazer persistência de dados, etc).
- package manager (NPM): gerenciador de pacotes do Node.
- REPL (node): ferramenta linha de comando que permite usar o Node no console.
- version manager (NVM): gerenciador de versões. Permite usar mais de uma versão do Node na máquina.

MongoDB

- Não relacional (NoSQL)
- BSON (JSON binário)
- mongodb.org
- Versão: baixar a versão Community Server

JavaScript

- Padrão (especificação e norma formal): ECMAScript(1996)
- Linguagem imperativa e estruturada
- Tipagem dinâmica (os tipos de dados não são declarados)
- Baseada em objetos (em JavaScript tudo é um objeto, porém não é orientada à classes e sim a protótipos)
- Interpretada (é executada linha a linha)

Sintaxe

- variáveis: const, let, var
- logical, relational e arithmetic operators
- functions e anonymous/arrow functions
- estruturas de controle e repetição
- Tipos de dados: Number, String, Boolean, Array, Object e Function
- Comentários // Comentário ou /* Comentário */

Variáveis

- var é a declaração padrão de uma variável
- const é uma constante. Se tentar atribuir novamente uma variável com o mesmo nome, gera um erro.
- let é uma variável que só existe no escopo em que foi criada. O escopo é o código existente entre chaves.

Funções de String

- length, indexOf e toString
- toUpperCase e toLowerCase
- endsWith e startsWith
- replace e trim
- parseFloat e parseInt
- split
- charAt
- slice

O replace do JavaScript funciona com expressão regular.

Funções de Array

- length
- push
- forEach
- splice
- indexOf

Testes iniciais

Crie uma pasta chamada **NodeProjects** e dentro dela, crie um arquivo chamado **index.js**

aula1/index.js

```
var nome = "Roberto";
let numero = 58;
var isAccepted = true;
var objeto = {};
var colecao = [];

objeto.nome = nome;
objeto.idade = numero;

colecao[0] = 1;
colecao[1] = "teste";

console.log(objeto.nome);
console.log(objeto.idade);

console.log(colecao[0]);
console.log(colecao[1]);

function somar(n1, n2){
    return n1 + n2;
}

objeto.somar = somar;

if (somar(1,2) === 3){
    console.log("Funcionou!");
}

for(var x=0; x < 5; x++){
    console.log(x);
}
```

```
if(nome.indexOf('o') !== -1){
    nome = nome.replace('t', '7');
    nome = nome.replace('e', '3');
    console.log(nome.indexOf('o'));
    console.log(nome.toUpperCase());
}

colecao.push(6);

console.log ("Tamanho do array: " + colecao.length);

colecao.forEach((item) => {
    console.log(item);
})
```

Resultado:

node index.js

```
C:\laragon\www\NodeProjects>node aula1/index.js
Roberto
58
1
teste
Funcionou!
0
1
2
3
4
1
ROB3R70
Tamanho do array: 3
1
teste
6

C:\laragon\www\NodeProjects>
```

Aula 02 - Node.js e Event Loop

Node.js

Características

- Interpretador (V8) + libuv
- Single Thread
- Assíncrono
- Plataforma != linguagem != framework
- bom para API's e scripts
- backend de games
- IoT
- Mensageria

No Node cada arquivo js é um módulo.

Exemplos:

- require
- module.exports
- assíncrono

Exportando dados

aula1/modulo.js

```
function somar(num1, num2){  
    return num1 + num2;  
}  
  
module.exports = {somar};
```

aula1/index.js

```
var modulo = require("./modulo");

var nome = "Roberto";
let numero = 58;
var isAccepted = true;
var objeto = {};
var colecao = [];

objeto.nome = nome;
objeto.idade = numero;

colecao[0] = 1;
colecao[1] = "teste";

console.log(objeto.nome);
console.log(objeto.idade);

console.log(colecao[0]);
console.log(colecao[1]);

// function somar(n1, n2){
//   return n1 + n2;
// }

// objeto.somar = somar;

if (modulo.somar(1,2) === 3){
  console.log("Funcionou!");
}

for(var x=0; x < 5; x++){
  console.log(x);
}

if(nome.indexOf('o') !== -1){
  nome = nome.replace('t', '7');
  nome = nome.replace('e', '3');
  console.log(nome.indexOf('o'));
  console.log(nome.toUpperCase());
}

colecao.push(6);

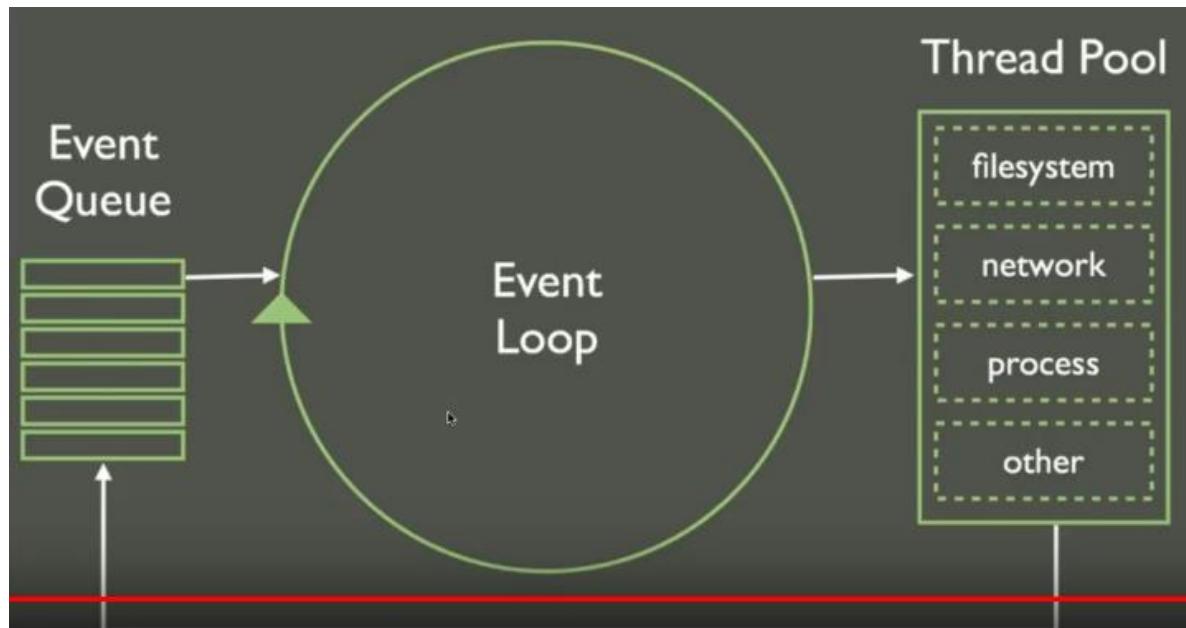
console.log ("Tamanho do array: " + colecao.length);

colecao.forEach((item) => {
  console.log(item);
})
```

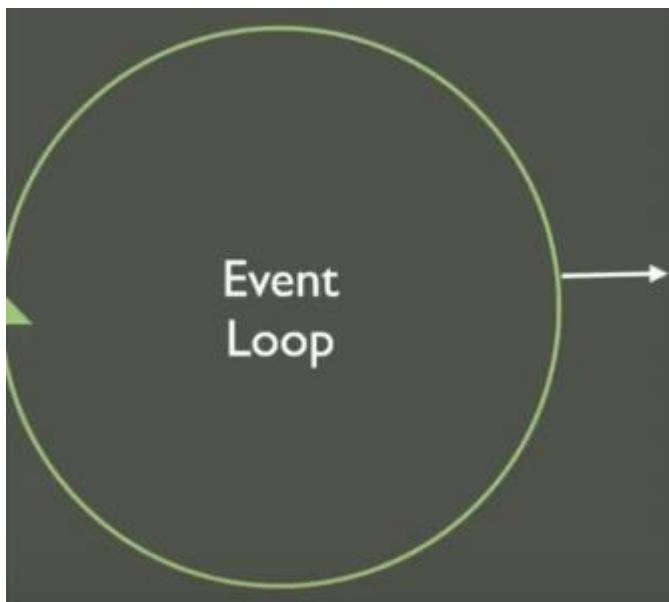
```
Roberto
58
1
teste
Funcionou!
0
1
2
3
4
1
ROB3R70
Tamanho do array: 3
1
teste
6

C:\laragon\www\NodeProjects>
```

Event Loop



Características



- V8 Single Thread
- libuv (C++) Multi Thread
- 1 ping = 88M instruções CPU
- O que fazer enquanto espera?

Aula 03 - ExpressJS e EJS

O ExpressJS é um web framework

Instalação do express-generator

npm install -g express-generator

```
C:\laragon\www\NodeProjects
λ npm install -g express-generator
C:\Users\Beto\AppData\Roaming\npm\express -> C:\Users\Beto\AppData\Roaming\npm\node_modules\express-generator\bin\express
+ express-generator@4.16.1
added 10 packages from 13 contributors in 17.421s
C:\laragon\www\NodeProjects
λ |
```

Instalando express-generator de maneira global é habilitado o comando express no console.

Na pasta NodeProjects entre com o comando: `express -e testeaula1`

```
C:\laragon\www\NodeProjects
λ express -e testeaula1

warning: option `--ejs' has been renamed to `--view=ejs'

create : testeaula1\
create : testeaula1\public\
create : testeaula1\public\javascripts\
create : testeaula1\public\images\
create : testeaula1\public\stylesheets\
create : testeaula1\public\stylesheets\style.css
create : testeaula1\routes\
create : testeaula1\routes\index.js
create : testeaula1\routes\users.js
create : testeaula1\views\
create : testeaula1\views\error.ejs
create : testeaula1\views\index.ejs
create : testeaula1\app.js
create : testeaula1\package.json
create : testeaula1\bin\
create : testeaula1\bin\www

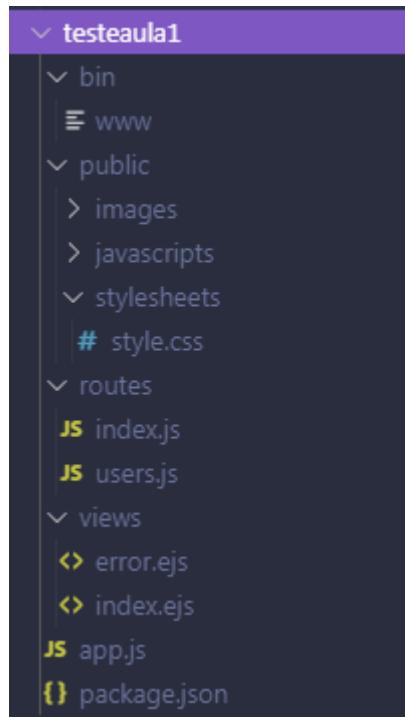
  testeaula1\bin\www]
change directory:
  > cd testeaula1

install dependencies:
  > npm install

run the app:
  > SET DEBUG=testeaula1:* & npm start
ode.js e MongoDB - Aula 03
```

C:\laragon\www\NodeProjects

É criada a seguinte estrutura:



testeaula1/package.json

```
{
  "name": "testeaula1",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "node ./bin/www"
  },
  "dependencies": {
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "ejs": "~2.6.1",
    "express": "~4.16.1",
    "http-errors": "~1.6.3",
    "morgan": "~1.9.1"
  }
}
```

Instalando as dependências

Vá para a pasta testeaula1 e entre com o comando:

```
npm install
```

```
C:\laragon\www\NodeProjects\testeaula1>npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
added 53 packages from 38 contributors and audited 141 packages in 15.311s
found 0 vulnerabilities
```

```
C:\laragon\www\NodeProjects\testeaula1>
```

Serão instaladas as dependências listadas no arquivo `package.json` que ficarão dentro de uma pasta criada chamada `node_modules`.

OBS.: Ao fazer deploy de sua aplicação, nunca suba esta pasta (`node_modules`) para o servidor.

Habilitando o servidor

Na pasta do projeto entre com o comando:

```
npm start
```

```
C:\laragon\www\NodeProjects\testeaula1
λ npm start -esquisa

> testeaula1@0.0.0 start C:\laragon\www\NodeProjects\testeaula1
> node ./bin/www

GET / 200 42.105 ms - 207
GET /stylesheets/style.css 200 16.890 ms - 111
GET /favicon.ico 404 7.078 ms - 1083
GET / 200 2.199 ms - 207
GET /stylesheets/style.css 200 1.837 ms - 111
GET /favicon.ico 404 5.909 ms - 1083
| Macbook-Pro-de-Luiz:testeaula1 Luizduartejr$ nano package.json
```

A porta padrão é a 3000.

No browser entre com:

```
localhost:3000
```



Express

Welcome to Express

CTRL-C derruba o servidor.

testeaula1/bin/www

```
#!/usr/bin/env node

/**
 * Module dependencies.
 */

var app = require('../app');
var debug = require('debug')('testeaula1:server');
var http = require('http');

/**
 * Get port from environment and store in Express.
 */
var port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

/**
 * Create HTTP server.
 */
var server = http.createServer(app);

/**
 * Listen on provided port, on all network interfaces.
 */

server.listen(port);
server.on('error', onError);
server.on('listening', onListening);

/**
 * Normalize a port into a number, string, or false.
 */
```

```
function normalizePort(val) {
  var port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }

  if (port >= 0) {
    // port number
    return port;
  }

  return false;
}

/***
 * Event listener for HTTP server "error" event.
 */

function onError(error) {
  if (error.syscall !== 'listen') {
    throw error;
  }

  var bind = typeof port === 'string'
    ? 'Pipe ' + port
    : 'Port ' + port;

  // handle specific listen errors with friendly messages
  switch (error.code) {
    case 'EACCES':
      console.error(bind + ' requires elevated privileges');
      process.exit(1);
      break;
    case 'EADDRINUSE':
      console.error(bind + ' is already in use');
      process.exit(1);
      break;
    default:
      throw error;
  }
}

/***
 * Event listener for HTTP server "listening" event.
 */
```

```
function onListening() {
  var addr = server.address();
  var bind = typeof addr === 'string'
    ? 'pipe ' + addr
    : 'port ' + addr.port;
  debug('Listening on ' + bind);
}
```

A nossa aplicação é [app.js](#)

testeaula1/app.js

```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use('/', indexRouter);
app.use('/users', usersRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};
```

```
// render the error page
res.status(err.status || 500);
res.render('error');
});

module.exports = app;
```

testeaula1/routes/index.js

```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('index', { title: 'Express' });
});

module.exports = router;
```

testeaula1/views/index.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <h1><%= title %></h1>
    <p>Welcome to <%= title %></p>
  </body>
</html>
```

Exemplo

- novo model
- nova view
- nova rota
- rota com parâmetro

testeaula1/app.js

```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
var testeRouter = require('./routes/teste');

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/teste', testeRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});
```

```
// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};
  // render the error page
  res.status(err.status || 500);
  res.render('error');
});

module.exports = app;
```

testeaula1/routes/index.js

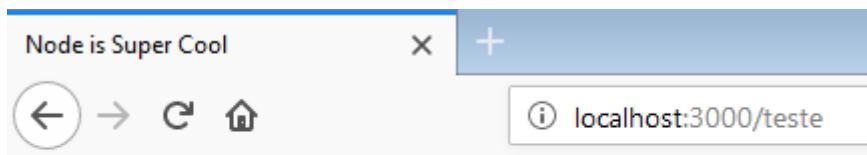
```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res, next) {
  res.render('teste', { title: 'Node is Super Cool', numero: 10 });
});

module.exports = router;
```

testeaula1/views/teste.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <h1><%= title %></h1>
    <p>Welcome to <%= title %></p>
    <p>
      Apenas um teste de Node.js! O número do model é <%= numero
    %>
    </p>
  </body>
</html>
```



Node is Super Cool

Welcome to Node is Super Cool

Apenas um teste de Node.js! O número do model é 10



Características

- HTML + JS Server Side (Node.js)
- Renderiza HTML
- Server Tag <%= %>
- Model
- forEach

Aula 04 - MongoDB e CRUD Command Line

MongoDB

Características

- free e open-source
- 10gen (2009)
- escrito em C++
- orientado a documentos (NoSQL)
- BSON
- sem JOINs e FKs
- documentos independentes
- schemaless
- usa JavaScript para consultas

Cenários

- Big Data
- dados desestruturados ou com estrutura dinâmica
- JavaScript Full-Stack
- Cache

Command Line

Uso

- mongod --dbpath
- show databases
- show collections
- use database
- db.collection.find

Entre na pasta do mongo:

C:\mongodb\bin

No terminal, digite:

mongod --storageEngine=mmapv1 --dbpath c:\mongodb\data\db --port 27017

A porta 27017 é a porta padrão do MongoDB.

```
C:\mongodb\bin
λ mongod --storageEngine=mmapv1 --dbpath c:\mongodb\data\db --port 27017
2019-08-15T20:31:09.451-0300 I CONTROL [main]
2019-08-15T20:31:09.457-0300 W CONTROL [main] 32-bit servers don't have journaling enabled by default. Please use --journal if you want durability.
2019-08-15T20:31:09.458-0300 I CONTROL [main]
2019-08-15T20:31:09.498-0300 I CONTROL [main] Hotfix KB2731284 or later update is not installed, will zero-out data files
2019-08-15T20:31:09.554-0300 I CONTROL [initandlisten] MongoDB starting : pid=13224 port=27017 dbpath=c:\mongodb\data\db 32-bit host=Beto-PC
2019-08-15T20:31:09.555-0300 I CONTROL [initandlisten] targetMinOS: Windows Vista/Windows Server 2008
2019-08-15T20:31:09.555-0300 I CONTROL [initandlisten] db version v3.2.22
2019-08-15T20:31:09.555-0300 I CONTROL [initandlisten] git version: 105acca0d443f9a47c1a5bd608fd7133840a58dd
2019-08-15T20:31:09.555-0300 I CONTROL [initandlisten] allocator: tcmalloc
2019-08-15T20:31:09.556-0300 I CONTROL [initandlisten] modules: none
2019-08-15T20:31:09.556-0300 I CONTROL [initandlisten] build environment:
2019-08-15T20:31:09.556-0300 I CONTROL [initandlisten]   distarch: i386
2019-08-15T20:31:09.556-0300 I CONTROL [initandlisten]   target_arch: i386
2019-08-15T20:31:09.557-0300 I CONTROL [initandlisten] options: { net: { port: 27017 }, storage: { dbPath: "c:\mongodb\data\db", engine: "mmapv1" } }
2019-08-15T20:31:09.658-0300 I CONTROL [initandlisten]
2019-08-15T20:31:09.660-0300 I CONTROL [initandlisten] ** WARNING: This 32-bit MongoDB binary is deprecated
2019-08-15T20:31:09.661-0300 I CONTROL [initandlisten]
2019-08-15T20:31:09.695-0300 I CONTROL [initandlisten]
2019-08-15T20:31:09.711-0300 I CONTROL [initandlisten] ** NOTE: This is a 32 bit MongoDB binary.
2019-08-15T20:31:09.712-0300 I CONTROL [initandlisten] ** 32 bit builds are limited to less than 2GB of data (or less with --journal).
2019-08-15T20:31:09.713-0300 I CONTROL [initandlisten] ** Note that journaling defaults to off for 32 bit and is currently off
2019-08-15T20:31:09.713-0300 I CONTROL [initandlisten] ** See http://dochub.mongodb.org/core/32bit
2019-08-15T20:31:09.713-0300 I CONTROL [initandlisten]
2019-08-15T20:31:09.832-0300 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2019-08-15T20:31:10.368-0300 W FTDC [initandlisten] Failed to initialize Performance Counters for FTDC: WindowsPdhError: PdhExpandCounterPathW failed with 'O objeto especificado não foi encontrado no computador.' for counter '\Processor(_Total)\% Idle Time'
2019-08-15T20:31:10.533-0300 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'c:/mongodb/data/db/diagnostic.data'
2019-08-15T20:31:10.575-0300 I NETWORK [initandlisten] waiting for connections on port 27017
2019-08-15T20:31:40.227-0300 I NETWORK [initandlisten] connection accepted from 127.0.0.1:51839 #1 (1 connection now open)
```

Abra um novo terminal e na pasta C:\mongodb\bin entre com o comando:

mongo

```
C:\mongodb\bin
λ mongo
2019-08-15T20:31:40.113-0300 I CONTROL [main] Hotfix KB2731284 or later update is not installed, will zero-out data files
MongoDB shell version: 3.2.22
connecting to: test
Server has startup warnings:
2019-08-15T20:31:09.658-0300 I CONTROL [initandlisten]
2019-08-15T20:31:09.660-0300 I CONTROL [initandlisten] ** WARNING: This 32-bit MongoDB binary is deprecated
2019-08-15T20:31:09.661-0300 I CONTROL [initandlisten]
2019-08-15T20:31:09.695-0300 I CONTROL [initandlisten]
2019-08-15T20:31:09.711-0300 I CONTROL [initandlisten] ** NOTE: This is a 32 bit MongoDB binary.
2019-08-15T20:31:09.712-0300 I CONTROL [initandlisten] ** 32 bit builds are limited to less than 2GB of data (or less with --journal).
2019-08-15T20:31:09.713-0300 I CONTROL [initandlisten] ** Note that journaling defaults to off for 32 bit and is currently off.
2019-08-15T20:31:09.713-0300 I CONTROL [initandlisten] ** See http://dochub.mongodb.org/core/32bit
2019-08-15T20:31:09.713-0300 I CONTROL [initandlisten]
```

show databases;

```
> show databases;
blogapp  0.078GB
local    0.078GB
> |
```

```
use aula2;
```

Com o comando `use <nome-banco>` se o banco não existir, ele será criado.

```
db.clientes.insert({nome: "Luiz", idade: 29, cidade: "Gravataí"});
```

```
> db.clientes.insert({nome: "Luiz", idade: 29, cidade: "Gravataí"});  
WriteResult({ "nInserted" : 1 })
```

```
show databases;
```

```
> show databases;  
aula2      0.078GB  
blogapp    0.078GB  
local      0.078GB
```

```
db.clientes.count();
```

```
> db.clientes.count();  
1
```

```
db.clientes.find();
```

```
> db.clientes.find();  
{ "_id" : ObjectId("5d5610fd528595682af3b275"), "nome" : "Luiz", "idade" : 29, "cidade" : "Gravataí" }
```

```
db.clientes.find().pretty();
```

```
> db.clientes.find().pretty();  
{ "_id" : ObjectId("5d5610fd528595682af3b275"),  
  "nome" : "Luiz",  
  "idade" : 29,  
  "cidade" : "Gravataí"  
}
```

```
db.clientes.insert( [ {nome: "Fernando", idade: 31, cidade: "Porto Alegre"}, {nome: "Osmar", UF: "RS"}]);
```

```
> db.clientes.insert([{"nome": "Fernando", "idade": 31, "cidade": "Porto Alegre"}, {"nome": "Osmar", "UF": "RS"}]);  
BulkWriteResult({  
  "writeErrors" : [ ],  
  "writeConcernErrors" : [ ],  
  "nInserted" : 2,  
  "nUpserted" : 0,  
  "nMatched" : 0,  
  "nModified" : 0,  
  "nRemoved" : 0,  
  "upserted" : [ ]  
})
```

```
db.clientes.find().pretty();

> db.clientes.find().pretty();

    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}

    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}

    "_id" : ObjectId("5d561390528595682af3b277"),
    "nome" : "Osmar",
    "UF" : "RS"
}
```

```
db.clientes.find({idade: 29}).pretty();
```

```
> db.clientes.find({idade: 29}).pretty();

    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}
```

```
// atualização de documentos inteiros
```

```
db.clientes.update({UF: "RS"}, {nome: "Francisco"});
```

```
> db.clientes.update({UF: "RS"}, {nome: "Francisco"});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
db.clientes.find().pretty();
```

```
> db.clientes.find().pretty();

    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}

    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}

{ "_id" : ObjectId("5d561390528595682af3b277"), "nome" : "Francisco" }
```

```
db.clientes.remove({_id: ObjectId("5d561390528595682af3b277")});
```

```
> db.clientes.remove({_id: ObjectId("5d561390528595682af3b277")});
WriteResult({ "nRemoved" : 1 })
```

```
db.clientes.find().pretty();
```

```
> db.clientes.find().pretty();
{
  "_id": ObjectId("5d5610fd528595682af3b275"),
  "nome": "Luiz",
  "idade": 29,
  "cidade": "Gravataí"
}
{
  "_id": ObjectId("5d561390528595682af3b276"),
  "nome": "Fernando",
  "idade": 31,
  "cidade": "Porto Alegre"
}
```

Aula 05 - Command Line (Part 2) e Studio 3T

Procura usando expressão regular

Buscando por clientes que possuem letra "a" no nome:

```
db.clientes.find({nome: {$regex: /a/}}).pretty();
```

```
> db.clientes.find({nome: {$regex: /a/}}).pretty();
    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}
```

Buscando por clientes que possuem letra "u" no nome:

```
db.clientes.find({nome: {$regex: /u/}}).pretty();
```

```
> db.clientes.find({nome: {$regex: /u/}}).pretty();
    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
} .
```

Buscando por clientes que possuem letra "a" ou "u" no nome:

```
db.clientes.find({nome: {$regex: /[au]/}}).pretty();
```

```
> db.clientes.find({nome: {$regex: /[au]/}}).pretty();
    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}
    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}
```

Uso

- db.collection.find({field: {\$regex: /a/i}})
- \$gte, \$gt, \$lt, \$lte
- \$eq, \$ne
- \$in, \$all

```
db.clientes.find({idade: {$lte: 30}}).pretty();
```

```
> db.clientes.find({idade: {$lte: 30}}).pretty();  
  
        "_id" : ObjectId("5d5610fd528595682af3b275"),  
        "nome" : "Luiz",  
        "idade" : 29,  
        "cidade" : "Gravataí"  
    },
```

```
db.clientes.find({idade: {$ne: 29}}).pretty();
```

```
> db.clientes.find({idade: {$ne: 29}}).pretty();  
  
        "_id" : ObjectId("5d561390528595682af3b276"),  
        "nome" : "Fernando",  
        "idade" : 31,  
        "cidade" : "Porto Alegre"  
    },
```

```
db.clientes.find({idade: {$in: [27, 28, 29]}}).pretty();
```

```
> db.clientes.find({idade: {$in: [27, 28, 29]}}).pretty();  
  
        "_id" : ObjectId("5d5610fd528595682af3b275"),  
        "nome" : "Luiz",  
        "idade" : 29,  
        "cidade" : "Gravataí"  
    },
```

```
db.clientes.insert({nome: "Gilberto", idade: 23, cidade: "Porto Alegre",  
tags: ["teste", "teste2"]});
```

```
> db.clientes.insert({nome: "Gilberto", idade: 23, cidade: "Porto Alegre", tags: ["teste", "teste2"]});  
WriteResult({ "nInserted" : 1 })
```

```
db.clientes.find({tags: {$all: ["teste", "teste2"]}}).pretty();
```

```
> db.clientes.find({tags: {$all: ["teste", "teste2"]}}).pretty();
Fonte: Parâmetro
    "_id" : ObjectId("5d5630c0528595682af3b278"),
    "nome" : "Gilberto",
    "idade" : 23,
    "cidade" : "Porto Alegre",
    "tags" : [
        "teste",
        "teste2"
    ]
}
```

Uso

- skip e limit
- sort
- db.collection.updateOne(filtro, updateOperator)
- \$set, \$unset

```
db.clientes.find().pretty();
```

```
> db.clientes.find().pretty();
Fonte: Parâmetro
    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}
    • sort
    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}

    "_id" : ObjectId("5d5630c0528595682af3b278"),
    "nome" : "Gilberto",
    "idade" : 23,
    "cidade" : "Porto Alegre",
    "tags" : [ db.clientes.find().skip(1).pretty(),
        "teste",
        "teste2"
    ]
}
```

```
db.clientes.find().skip(1).pretty();
```

```
> db.clientes.find().skip(1).pretty();

    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}

    "_id" : ObjectId("5d5630c0528595682af3b278"),
    "nome" : "Gilberto",
    "idade" : 23,
    "cidade" : "Porto Alegre",
    "tags" : [
        "teste",
        "teste2"
    ]
}
```

```
db.clientes.find().limit(1).pretty();
```

```
> db.clientes.find().limit(1).pretty();

    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}
```

```
db.clientes.find().skip(1).limit(1).pretty();
```

```
> db.clientes.find().skip(1).limit(1).pretty();

    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}
```

```
// ordena por ordem crescente
```

```
db.clientes.find().sort({idade: 1}).pretty();
```

```
> db.clientes.find().sort({idade: 1}).pretty();

    "_id" : ObjectId("5d5630c0528595682af3b278"),
    "nome" : "Gilberto",
    "idade" : 23,
    "cidade" : "Gravataí"
    "tags" : [
        "teste",
        "teste2"
    ]
}

    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}

    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}

```

// ordena por ordem decrescente

```
db.clientes.find().sort({idade: -1}).pretty();
```

```
> db.clientes.find().sort({idade: -1}).pretty();

    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}

    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}

    "_id" : ObjectId("5d5630c0528595682af3b278"),
    "nome" : "Gilberto",
    "idade" : 23,
    "cidade" : "Porto Alegre",
    "tags" : [
        "teste",
        "teste2"
    ]
}
```

```
db.clientes.updateOne({_id: ObjectId("5d5630c0528595682af3b278")},  
{$set: {nome: "Ricardo"}});
```

```
> db.clientes.updateOne({_id: ObjectId("5d5630c0528595682af3b278")}, {$set: {nome: "Ricardo"}})  
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }  
> |
```

```
db.clientes.find().pretty();
```

```
> db.clientes.find().pretty();  
  
        "_id" : ObjectId("5d5610fd528595682af3b275"),  
        "nome" : "Luiz",  
        "idade" : 29,  
        "cidade" : "Gravataí"  
    }  
  
        "_id" : ObjectId("5d561390528595682af3b276"),  
        "nome" : "Fernando",  
        "idade" : 31,  
        "cidade" : "Porto Alegre"  
    }  
    db.clientes.updateOne({_id: ObjectId("5d5630c0528595682af3b278")},  
    {$set: {nome: "Ricardo"}})  
  
        "_id" : ObjectId("5d5630c0528595682af3b278"),  
        "nome" : "Ricardo",  
        "idade" : 23,  
        "cidade" : "Porto Alegre",  
        "tags" : [ db.clientes.find().pretty();  
            "teste",  
            "teste2"  
        ]  
    }  
> |
```

```
// Para limpar um campo
```

```
db.clientes.updateOne({_id: ObjectId("5d5630c0528595682af3b278")},  
{$unset: {tags: null}});
```

```
> db.clientes.updateOne({_id: ObjectId("5d5630c0528595682af3b278")}, {$unset: {tags: null}})  
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }  
> |
```

```

db.clientes.find().pretty();

> db.clientes.find().pretty();
{
    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}
{
    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}
{
    "_id" : ObjectId("5d5630c0528595682af3b278"),
    "nome" : "Ricardo",
    "idade" : 23,
    "cidade" : "Porto Alegre"
}
> db.clientes.updateOne({_id: ObjectId("5d561390528595682af3b276"),
    $set: {tags: null}});
{
    "_id" : ObjectId("5d5630c0528595682af3b278"),
    "nome" : "Ricardo",
    "idade" : 23,
    "cidade" : "Porto Alegre"
}
> db.clientes.find().pretty();

```

Uso

- \$inc, \$mul
- \$rename
- upsert

```

db.clientes.update({nome: "José"}, {nome: "José", idade: 33, cidade: "São Paulo"}, {upsert: true});

```

```

> db.clientes.update({nome: "José"}, {nome: "José", idade: 33, cidade: "São Paulo"}, {upsert: true});
WriteResult({
    "nMatched" : 0,
    "nUpserted" : 1,
    "nModified" : 0,
    "_id" : ObjectId("5d563fad08758374fbad8c7b")
})
> db.clientes.find({nome: "José", idade: 33, cidade: "São Paulo"}, {upsert: true});

```

```
db.clientes.find().pretty();
```

```
> db.clientes.find().pretty();
{
    "_id" : ObjectId("5d5610fd528595682af3b275"),
    "nome" : "Luiz",
    "idade" : 29,
    "cidade" : "Gravataí"
}

    "_id" : ObjectId("5d561390528595682af3b276"),
    "nome" : "Fernando",
    "idade" : 31,
    "cidade" : "Porto Alegre"
}

    "_id" : ObjectId("5d5630c0528595682af3b278"),
    "nome" : "Ricardo",
    "idade" : 23,
    "cidade" : "Porto Alegre"
}

    "_id" : ObjectId("5d563fad08758374fbad8c7b"),
    "nome" : "José",
    "idade" : 33,
    "cidade" : "São Paulo"
}
> |
```

Studio 3T

É uma ferramenta gráfica para o Mongo.

Faça o download em:

<https://studio3t.com/download/>

The screenshot shows the official website for Studio 3T. At the top, there's a navigation bar with links for FEATURES, SOLUTIONS, RESOURCES, CONTACT US, PRICING, and DOWNLOAD. Below the navigation, there's a prominent green button labeled "Register here". A banner at the top of the page reads: "NEW: Studio 3T webinars. Kickstart your onboarding and sign up for our upcoming webinar, Getting Started with Studio 3T. Don't miss your spot!"

Download Studio 3T for MongoDB

Whether you're new to NoSQL or a MongoDB pro,

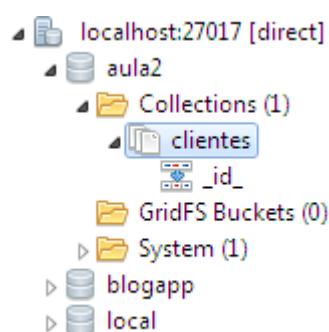


Studio 3T's website uses cookies to ensure you get the best experience on our website, to measure ad performance, and to analyse the way visitors use our website in general. [Learn more about how we use cookies and how to opt-out](#)

OK

Características

- ex-Mongo Chef
 - free for non-commercial
 - MongoDB GUI
- Crie uma nova conexão.



clientes

localhost:27017 [direct] aula2 clientes

Query: {}

Projection: {}

Skip: 0

Sort: {}

Limit: 100

Result | Query Code | Explain

Documents 1 to 4 | 50 | Tree View |

clientes > _id

_id	nome	idade	cidade
5d5610fd528595682af3b275	Luiz	29.0	Gravataí
5d561390528595682af3b276	Fernando	31.0	Porto Alegre
5d5630c0528595682af3b278	Ricardo	23.0	Porto Alegre
5d563fad08758374fbad8c7b	José	33.0	São Paulo

Result | Query Code | Explain

Documents 1 to 4 | 50 | Tree View |

Key | Value | Type

Key	Value	Type
1 { _id : 5d5610fd528595682af3b275 }	{ 4 fields } _id: 5d5610fd528595682af3b275 nome: Luiz idade: 29.0 cidade: Gravataí	Document
2 { _id : 5d561390528595682af3b276 }	{ 4 fields } _id: 5d561390528595682af3b276 nome: Fernando idade: 31.0 cidade: Porto Alegre	Document
3 { _id : 5d5630c0528595682af3b278 }	{ 4 fields } _id: 5d5630c0528595682af3b278 nome: Ricardo idade: 23.0 cidade: Porto Alegre	Document
4 { _id : 5d563fad08758374fbad8c7b }	{ 4 fields } _id: 5d563fad08758374fbad8c7b nome: José idade: 33.0 cidade: São Paulo	Document

Aula 06 - NodeJS + MongoDB

Na pasta **NodeProjects** entre com o comando:

```
express -e --git aula2
```

The screenshot shows a terminal window with the following output:

```
C:\laragon\www\NodeProjects
λ express -e --git aula2
Colar
warning: option `--ejs` has been renamed to `--view=ejjs'
Área de Transferência
C:\laragon\www\NodeProjects\aula2
create : aula2\
create : aula2\public\
create : aula2\public\javascripts\
create : aula2\public\images\
create : aula2\public\stylesheets\
create : aula2\public\stylesheets\style.css
create : aula2\routes\
create : aula2\routes\index.js
create : aula2\routes\users.js
create : aula2\views\
create : aula2\views\error.ejs
create : aula2\views\index.ejs
create : aula2\.gitignore
create : aula2\app.js
create : aula2\package.json
create : aula2\bin\
create : aula2\bin\www

change directory:
> cd aula2

install dependencies:
> npm install

run the app:
> SET DEBUG=aula2:* & npm start

C:\laragon\www\NodeProjects\aula2
λ
```

- Entre na pasta do projeto

```
cd aula2
```

- Instale as dependências:

```
npm install
```

```
C:\laragon\www\NodeProjects\aula2
λ npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
added 53 packages from 38 contributors and audited 141 packages in 24.184s
found 0 vulnerabilities
```

- Instale a dependência do MongoDB:

```
npm install mongodb
```

```
C:\laragon\www\NodeProjects\aula2
λ npm install mongodb
+ mongodb@3.3.0
added 5 packages from 3 contributors and audited 147 packages in 5.713s
found 0 vulnerabilities
```

Conexão db.js

```
const mongoClient = require("mongodb").MongoClient
mongoClient.connect("mongodb://localhost:27017/workshop")
  .then(conn => global.conn = conn.db("workshop"))
  .catch(err => console.log(err))

module.exports = {}
```

Funções db.js

```
function findAll(callback){

  global.conn.collection("customers").find({}).toArray(callback);
}

module.exports = { findAll }
```

aula2/db.js

```
var mongoClient = require("mongodb").MongoClient;

mongoClient.connect("mongodb://localhost:27017/aula2")
.then(conn => global.connection = conn.db("aula2"))
.catch(err => console.log(err));

function findAll(callback){
    global.connection.collection("clientes").find({}).toArray(callback);
}

module.exports = { findAll }
```

routes/index.js

```
var express = require('express');
var router = express.Router();

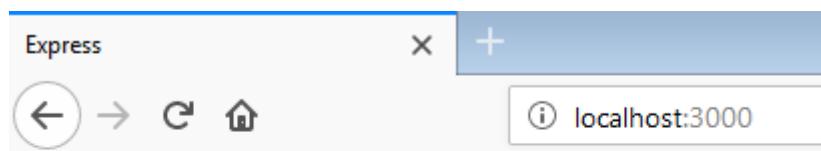
/* GET home page. */
router.get('/', function(req, res, next) {
  var db = require("../db");
  db.findAll(function(err, docs){
    if(err)
      res.render('index', { title: 'ERRO', clientes: {} });
    else
      res.render('index', { title: 'Express', clientes: docs });
  });
});

module.exports = router;
```

views/index.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <h1><%= title %></h1>
    <p>Welcome to <%= title %></p>
    <ul>
      <%
        clientes.forEach(function(item){
      %>
        <li><%= item.nome %></li>
      <%
        })
      %>
    </ul>

  </body>
</html>
```



Express

Welcome to Express

- Luiz
- Fernando
- Ricardo
- José