

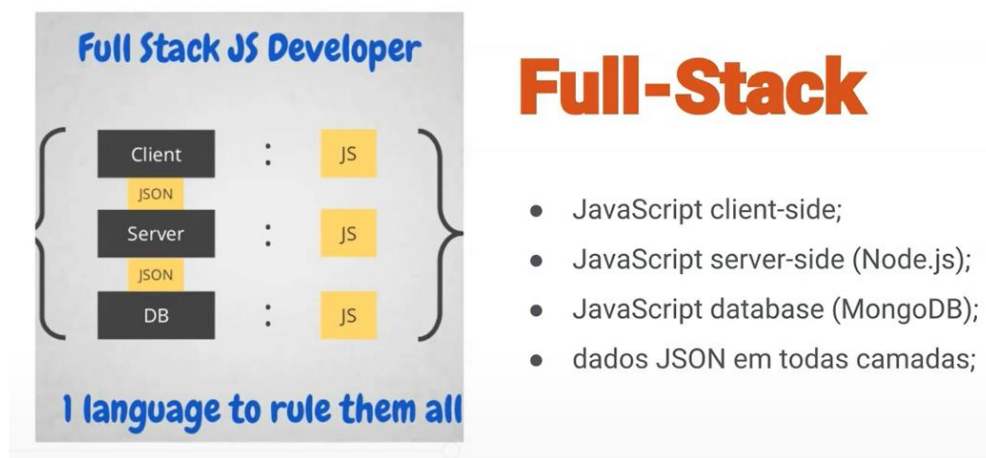
Curso básico Node.JS

Prof. Luiz Duarte (Luiz Tools)

Aula 01 - O que é Nodejs

JavaScript:

- Linguagem web mais popular do mundo;
- baixa curva de aprendizado;
- forte comunidade;



Leve e Multiplataforma:

- baixo custo de hardware;
- baixo custo de SO;
- sem licença;

- É utilizada por várias instituições corporativas:



Aula 02 - Ambiente Nodejs e MongoDB

Ambiente

Node.js

- nodejs.org
- Runtime JS (V8)
- IO (libuv)
- package manager (NPM)
- REPL (node)
- version manager (NVM)

- O Node, em si, não é uma linguagem de programação. Ele é uma plataforma para aplicações composta por:

- Interpretador V8 (o mesmo do Google Chrome): Motor de JavaScript;
- Biblioteca IO (libuv): biblioteca C++ para leitura e escrita;
- package manager (NPM): gerenciador de pacotes;
- ferramenta REPL: linha de comando onde se pode usar o Node diretamente do console;
- version manager (NVM): gerenciador de versões que permite usar mais de uma versão do Node na máquina

- Acesse nodejs.org, baixe e instale o **Node**.

Ferramenta

- Acesse code.visualstudio.com, baixe e instale o **Visual Studio Code**.

VS Code

- Microsoft
- Free
- Multiplataforma
- code.visualstudio.com
- Code Complete

- Acesse mongodb.org, baixe e instale o **MongoDB** (Versão Community Server)

MongoDB

- Não-relacional (NoSQL)
- BSON (JSON binário)
- mongodb.org
- Community Server

Aula 03 - O que é JavaScript

Características:

- Criado por Brendan Eich;
- Usado inicialmente no Netscape (1995);
- Especificação formal (norma): ECMAScript (1996);
- Linguagem imperativa e estruturada;
- Tipagem dinâmica (não se declara tipos nos dados);
- Baseada em objetos (em JavaScript tudo é um objeto);
- Interpretada (É executado linha a linha. Não retorna erros de compilação);

- JavaScript não é orientado a classes. É orientado a protótipos.

- **Typescript:** é um subset criado encima do JavaScript para que ele forneça recursos de tipagem, de objetos principalmente, e alguns outros recursos ligados a orientação a objetos. Adiciona uma nova camada.

- Neste curso não será usado typescript.

Aula 04 - Revisão de JavaScript - Parte 1

Sintaxe

- `const`, `let`, `var`
- logical, relational e arithmetic operators
- functions e anonymous/arrow functions

Variáveis em JavaScript

- `var` cria uma variável que pode ser alterada.
- `const` cria uma variável que não pode ser alterada.
- `let` cria uma variável que só existe dentro do escopo em que foi criado (bloco entre chaves).

- Crie uma pasta chamada `aula1` e dentro dela adicione um arquivo chamado `index.js`:

```
C:\luiztools>md aula1
```

`cd aula1`

`index.js`

```
var nome = 'Roberto';
console.log("Nome: " + nome);

let numero1 = 3;
console.log("Número 1: " + numero1);

let numero2 = 6.9;
console.log("Número 2: " + numero2);

var isAccepted = true; // ou false
console.log("É aceito: " + isAccepted);

// objeto
var user = {};
user.nome = 'Gilberto';
user.idade = 17;

console.log("Nome do usuário: " + user.nome);
console.log("Idade do usuário: " + user.idade);

// array
var colecao = [];
colecao[0] = 1;
colecao[1] = 'abacate';

console.log("Primeiro item da coleção: " + colecao[0]);
console.log("Segundo item da coleção: " + colecao[1]);
```

- O ponto e vírgula no final de um comando é opcional na maioria das versões do JavaScript (exceto nas versões mais antigas)

node index.js

```
C:\luiztools\aula1>node index.js
Nome: Roberto
Número 1: 3
Número 2: 6.9
É aceito: true
Nome do usuário: Gilberto
Idade do usuário: 17
Primeiro item da coleção: 1
Segundo item da coleção: abacate
```

Aula 05 - Revisão de JavaScript - Parte 2

Operadores

Operadores de comparação

operadores_comparacao.js

```
if(3 == '3'){
  console.log("3 == '3': true");
}

if(3 === '3'){
  console.log("3 === '3': true");
} else {
  console.log("3 === '3': false");
}

if(3 != '3'){
  console.log("3 != '3': true");
} else {
  console.log("3 != '3': false");
}

if(3 !== '3'){
  console.log("3 !== '3': true");
} else {
  console.log("3 !== '3': false");
}
```

```
C:\luiztools\aula1>node operadores_comparacao.js
3 == '3': true
3 === '3': false
3 != '3': false
3 !== '3': true
```

Operadores lógicos

&&	and
	or
!	not

Operadores relacionais

>	maior
<	menor
>=	maior ou igual
<=	menor ou igual

Operadores aritméticos

+	soma
-	subtração
*	multiplicação
/	divisão
%	resto

Funções

somar.js

```
function somar(num1, num2){  
  return num1 + num2;  
}
```

```
var result = somar(3, 4.5);
```

```
console.log(result);
```

```
node somar.js
```

```
C:\luiztools\aula1>node somar.js  
7.5
```

Anonymous Function

anonymous_function.js

```
const somar = function (num1, num2) {  
  return num1 + num2;  
}
```

```
var result = somar(3, 61);
```

```
console.log(result);
```

```
node anonymous_function.js
```

```
C:\luiztools\aula1>node anonymous_function.js  
64
```


Arrow Function

arrow_function.js

```
const somar = (num1, num2) => {  
  return num1 + num2;  
}
```

```
var result = somar(6, 8);
```

```
console.log(result);
```

node arrow_function.js

```
C:\luiztools\aula1>node arrow_function.js  
14
```

Aula 06 - Revisão de JavaScript - Parte 3

Estruturas de controle e repetição

estruturas.js

```
// Estruturas de controle e repetição
```

```
alunos = [{nome: "Carlos", media: 7.5},{nome: "Renato", media: 4.75},{nome:"Francisco",media: 3.5}];
```

```
for (let i = 0; i < alunos.length; i++) {  
  if(alunos[i].media > 5){  
    console.log(alunos[i].nome + " - Média: " + alunos[i].media + " - Aprovado!" );  
  } else if(alunos[i].media > 4 && alunos[i].media < 5) {  
    console.log(alunos[i].nome + " - Média: " + alunos[i].media + " - Em recuperação!" );  
  } else {  
    console.log(alunos[i].nome + " - Média: " + alunos[i].media + " - Reprovado!" );  
  }  
}
```

```
var j = 0;
```

```
while(j < 5){  
  console.log(j);  
  j++;  
}
```

```
var option = 3;
```

```
switch(option){  
  case 1:  
    console.log("Escolheu a opção 1");  
    break;  
  case 2:  
    console.log("Escolheu a opção 2");  
    break;  
  case 3:  
    console.log("Escolheu a opção 3");  
    break;  
  case 4:  
    console.log("Escolheu a opção 4");  
    break;  
  default:  
    console.log("Não escolheu nenhuma das quatro opções!");  
}
```

node estruturas.js

```
C:\luiztools\aula1>node estruturas.js
Carlos - Média: 7.5 - Aprovado!
Renato - Média: 4.75 - Em recuperação!
Francisco - Média: 3.5 - Reprovado!
0
1
2
3
4
Escolheu a opção 3
```

Tipos de dados

- Number;
- String;
- Boolean;
- Array;
- Object;
- Function;

Funções de String

- length, indexOf e toString
- toUpperCase e toLowerCase
- endsWith e startsWith
- replace e trim
- parseFloat e parseInt
- split: transforma uma string em array;
- charAt: informa qual é a letra passada por parâmetro;
- slice: para pegar um pedaço de uma string (uma substring);

strings.js

```
var nome1 = "Roberto";
var nome2 = "Lucas";
var nome3 = " Felipe ";
var nome4 = "Roberto Pinheiro";
var num1 = "5";
var num2 = "57.31";

if(nome1.indexOf("o") != -1){
  console.log(nome1.toUpperCase());
} else {
  console.log(nome1.toLocaleLowerCase());
}

if(nome2.indexOf("o") != -1){
  console.log(nome2.toUpperCase());
} else {
  console.log(nome2.toLocaleLowerCase());
}

if(nome1.indexOf("o") != -1){
  console.log(nome1.replace("o", "3"));
}

console.log(nome3.trim());
console.log(parseInt(num1));
console.log(parseFloat(num2));
console.log(nome4.split(" "));
console.log(nome1.charAt(3));
console.log(nome1.slice(0,3));
```

```
C:\luiztools\aula1>node strings.js
ROBERTO
lucas
R3berto
Felipe
5
57.31
[ 'Roberto', 'Pinheiro' ]
e
Rob
```

Arrays

- length
- push
- forEach
- splice: para remover pedaços de um array
- indexOf: para saber em qual posição do array está um determinado texto

arrays.js

```
var nomes = ["Geraldo", "Cássio", "Ademir", "José", "Valdir"];
```

```
nomes.push("Valter");
```

```
var total_nomes = nomes.length;  
console.log("Total de nomes: " + total_nomes);
```

```
nomes.forEach(function(item){  
    console.log(item);  
});
```

```
console.log(nomes.splice(0, 3));
```

```
console.log(nomes.indexOf("Valter"));
```

```
C:\luiztools\aula1>node arrays.js  
Total de nomes: 6  
Geraldo  
Cássio  
Ademir  
José  
Valdir  
Valter  
[ 'Geraldo', 'Cássio', 'Ademir' ]  
2
```

▼ AULA1

- .js anonymous_function.js
- .js arrays.js
- .js arrow_function.js
- .js estruturas.js
- .js index.js
- .js operadores_comparacao.js
- .js somar.js
- .js strings.js

Aula 07 - O que é Nodejs - Parte 2

Características:

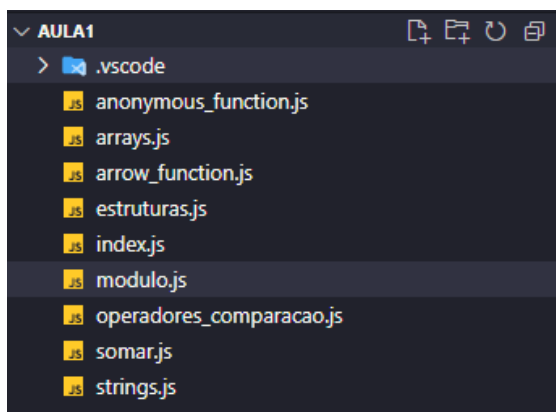
- Interpretador (V8) + libuv
- Single Thread
- Assíncrono
- Plataforma != linguagem != framework
- Bom para API's e scripts
- backend de games
- IoT
- mensageria

O Node é uma plataforma. É um hospedeiro de aplicações.

Aula 09 - Nodejs na Prática - Parte 2

- No Node cada arquivo js é um módulo.

Módulos



modulo.js

```
function somar(valor1, valor2){  
  return valor1 + valor2;  
}
```

```
module.exports = {somar}
```

index.js

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

```
var nome = 'Roberto';  
console.log("Nome: " + nome);
```

```
let numero1 = 3;  
console.log("Número 1: " + numero1);
```

```
let numero2 = 6.9;  
console.log("Número 2: " + numero2);
```

```
var isAccepted = true; // ou false  
console.log("É aceito: " + isAccepted);
```

```
// objeto  
var user = {};  
user.nome = 'Gilberto';  
user.idade = 17;
```

```
console.log("Nome do usuário: " + user.nome);  
console.log("Idade do usuário: " + user.idade);
```

```
// array  
var colecao = [];
```

```
colecacao[0] = 1;
colecacao[1] = 'abacate';
```

```
console.log("Primeiro item da coleção: " + colecacao[0]);
console.log("Segundo item da coleção: " + colecacao[1]);
```

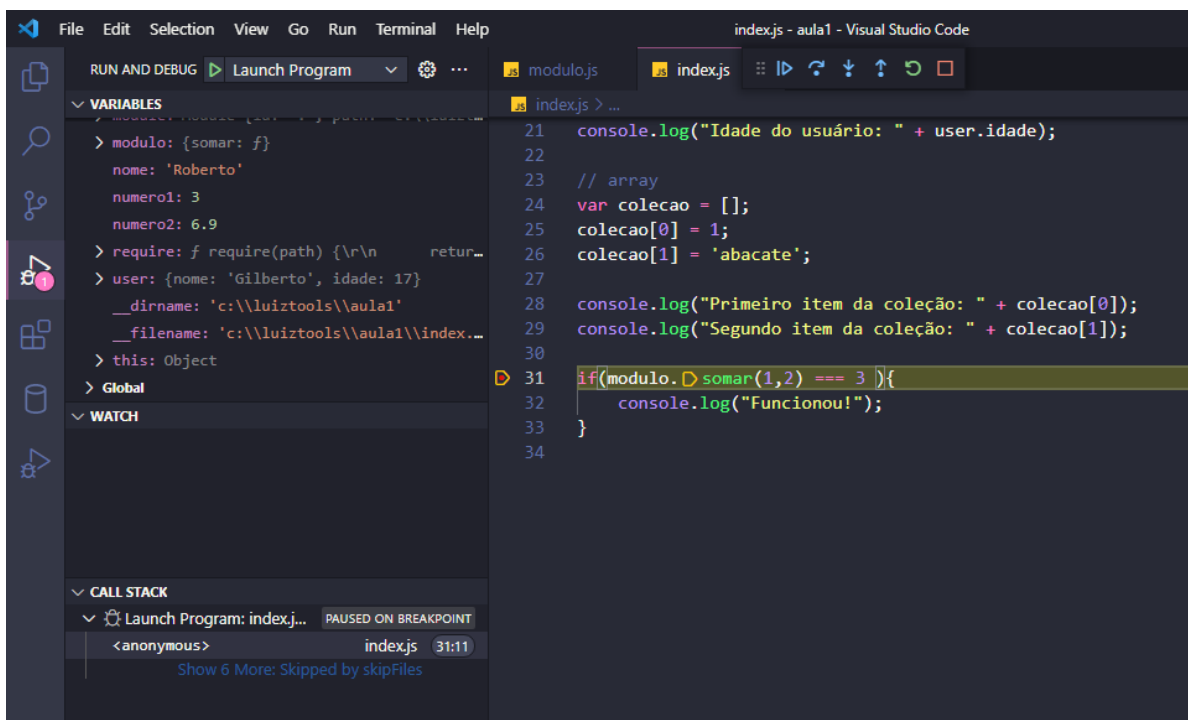
```
if(modulo.somar(1,2) === 3 ){
  console.log("Funcionou!");
}
```

node index.js

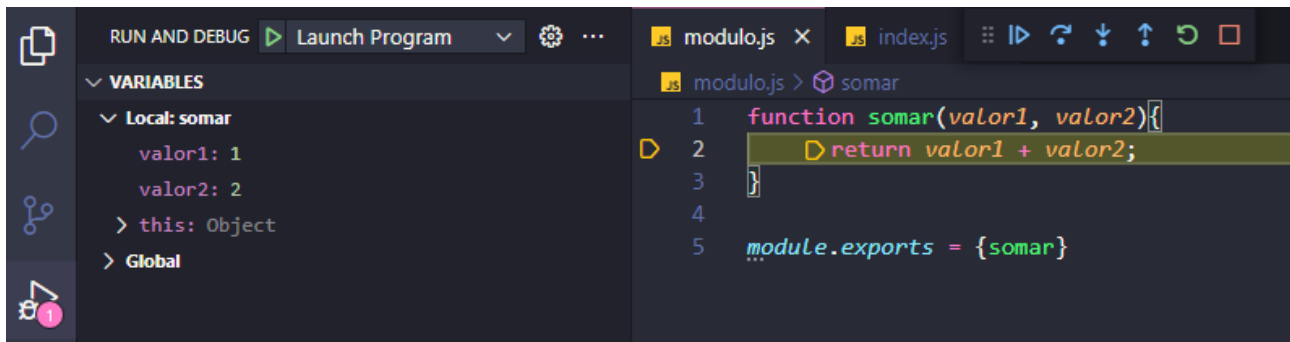
```
C:\luiztools\aula1>node index.js
Nome: Roberto
Número 1: 3
Número 2: 6.9
É aceito: true
Nome do usuário: Gilberto
Idade do usuário: 17
Primeiro item da coleção: 1
Segundo item da coleção: abacate
Funcionou!
```

Debug no Visual Studio Code

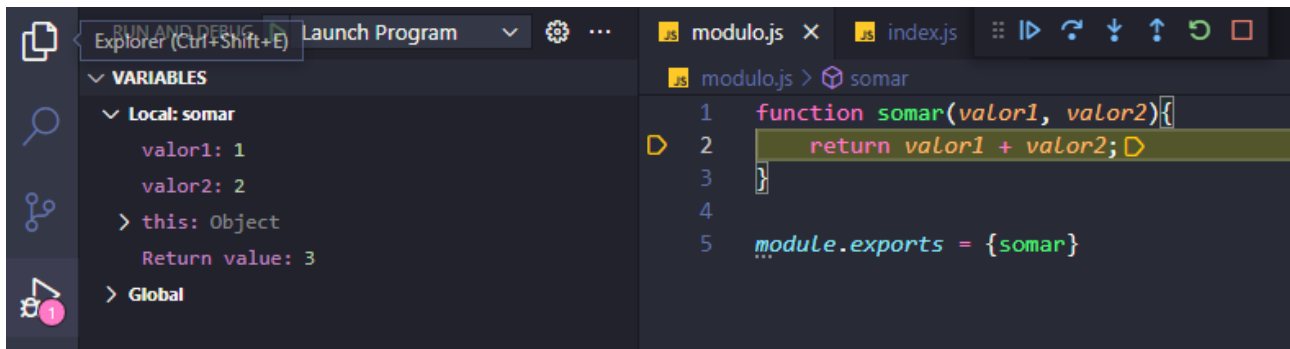
- Faça um breakpoint na linha 31.



- Clique no botão **Step into**:



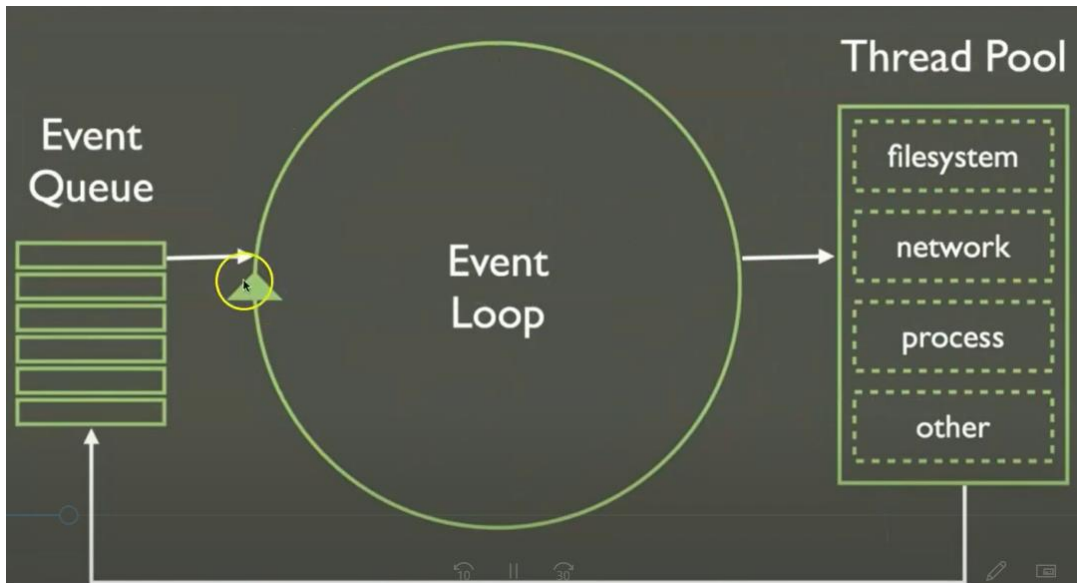
- Clique novamente no botão **Step into**:



Aula 10 - Event Loop do Nodejs

- O comportamento assíncrono no JS se deve em parte a sua arquitetura centrada em event loop.
- Event loop é uma thread única, uma single thread principal, onde todas as requisições chegam inicialmente nele, são processadas imediatamente e depois retornam a sua execução (resultado da sua computação) no fluxo principal e consequentemente para o usuário.

Event loop



- Ao chegar uma requisição, ela vai para uma fila de eventos.
- Se ela for a primeira da fila, ela vai entrar no Event Loop. Se é rápido de processar, ele já processa e devolve para o usuário. Depois vai para o próximo evento da fila. Se não for rápido de processar, vai para um pool de threads. O Node tem uma thread principal (single thread), mas mantém um pool de threads (quatro). Por ele não abrir uma thread por requisição, mas sim compartilhar um pool de threads, ele se torna mais eficiente em consumo de recursos.
- O Node envia para o pool de threads tudo o que é de execução demorada e retorna para atender a próxima requisição da fila.

Características:

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

Aula 11 - O que é ExpressJS

- O Express é um web framework.
- Express Generator é o scaffold de Express.

Instalando o pacote express-generator

No terminal do DOS, entre com o seguinte comando:

```
npm install -g express-generator
```

```
C:\Users\beto1>npm install -g express-generator
npm WARN deprecated mkdirp@0.5.1: Legacy versions of mkdirp are no longer supported. Please update to mkdirp 1.x. (Note that the API surface has changed to use Promises in 1.x.)
C:\Users\beto1\AppData\Roaming\npm\express -> C:\Users\beto1\AppData\Roaming\npm\node_modules\express-generator\bin\express-cli.js
+ express-generator@4.16.1
added 10 packages from 13 contributors in 6.37s
```

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

Criando o scaffold

```
express -e testeaula01
```

```
C:\luiztools>express -e testeaula01

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

create : testeaula01\
create : testeaula01\public\
create : testeaula01\public\javascripts\
create : testeaula01\public\images\
create : testeaula01\public\stylesheets\
create : testeaula01\public\stylesheets\style.css
create : testeaula01\routes\
create : testeaula01\routes\index.js
create : testeaula01\routes\users.js
create : testeaula01\views\
create : testeaula01\views\error.ejs
create : testeaula01\views\index.ejs
create : testeaula01\app.js
create : testeaula01\package.json
create : testeaula01\bin\
create : testeaula01\bin\www

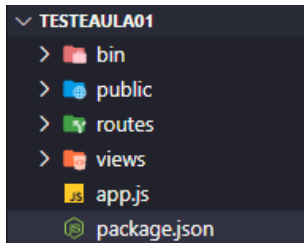
change directory:
> cd testeaula01

install dependencies:
> npm install

run the app:
> SET DEBUG=testeaula01:* & npm start
```

cd testeaula01
code .

```
C:\luiztools>cd testeaula01  
C:\luiztools\testeaula01>code .
```



package.json

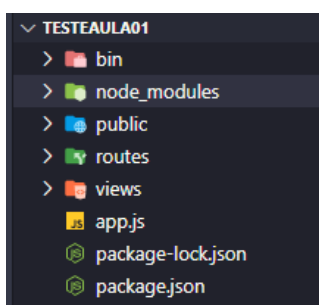
```
{  
  "name": "testeaula01",  
  "version": "0.0.0",  
  "private": true,  
  "scripts": {  
    "start": "node ./bin/www"  
  },  
  "dependencies": {  
    "cookie-parser": "~1.4.4",  
    "debug": "~2.6.9",  
    "ejs": "~2.6.1",  
    "express": "~4.16.1",  
    "http-errors": "~1.6.3",  
    "morgan": "~1.9.1"  
  }  
}
```

- No terminal do Visual Studio Code, entre com o comando:

npm install

```
C:\luiztools\testeaula01>npm install  
npm notice created a lockfile as package-lock.json. You should commit this file.  
added 54 packages from 38 contributors and audited 55 packages in 18.595s  
found 0 vulnerabilities
```

- É criada uma pasta chamada "**node_modules**" com todas as dependências e não deve ser subida ao servidor.



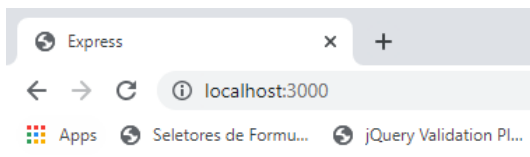
Rodando o projeto

npm start

```
C:\luiztools\testeaula01>npm start  
  
> testeaula01@0.0.0 start C:\luiztools\testeaula01  
> node ./bin/www
```

- No browser:

localhost:3000



Express

Welcome to Express

```
C:\luiztools\testeaula01>npm start  
  
> testeaula01@0.0.0 start C:\luiztools\testeaula01  
> node ./bin/www  
  
GET / 200 37.831 ms - 207  
GET /stylesheets/style.css 200 7.600 ms - 111  
GET /favicon.ico 404 6.338 ms - 943
```

Aula 12 - ExpressJS na Prática - Parte 1

bin\www

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

/**
 * Module dependencies.
 */

var app = require('../app');
var debug = require('debug')('testeaula01: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);
}

```

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;
```


Aula 13 - ExpressJS na Prática - Parte 2

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;
```

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>
```

public\stylesheets\style.css

```
body {
  padding: 50px;
  font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;
}

a {
  color: #00B7FF;
}
```

Aula 14 - ExpressJS na Prática - Parte 3

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>Teste de Node.js! O número do model é <%= numero %></p>
</body>

</html>
```

routes\teste.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;
```

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;

```

npm start

```

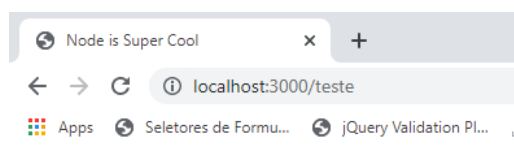
C:\luiztools\testeaula01>npm start

> testeaula01@0.0.0 start C:\luiztools\testeaula01
> node ./bin/www


```

- No browser:

localhost:3000/teste



Node is Super Cool

Welcome to Node is Super Cool

Teste de Node.js! O número do model é 10

Aula 15 - O que é MongoDB

- MongoDB é orientado a documentos.
- É um banco de dados não-relacional.

Características:

- BSON
- sem JOIN's e FK's
- documentos independentes
- schemaless
- usa JavaScript para consulta

ACID:

- Atomicidade
- Consistência
- Isolamento
- Durabilidade

- Em vez de tabelas, no MongoDB temos coleções de documentos.

Aula 16 - Quando usar MongoDB

Cenários de uso do MongoDB:


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

- É muito usado em ecommerce (em catálogo de produtos).

Aula 17 - Como rodar servidor MongoDB

Instalação

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

 [Cloud](#) [Software](#) [Pricing](#) [Learn](#) [Solutions](#) [Docs](#)

[Contact](#) [Sign In](#) [Try Free](#)

MongoDB Enterprise Server

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
4.4.6 (current)

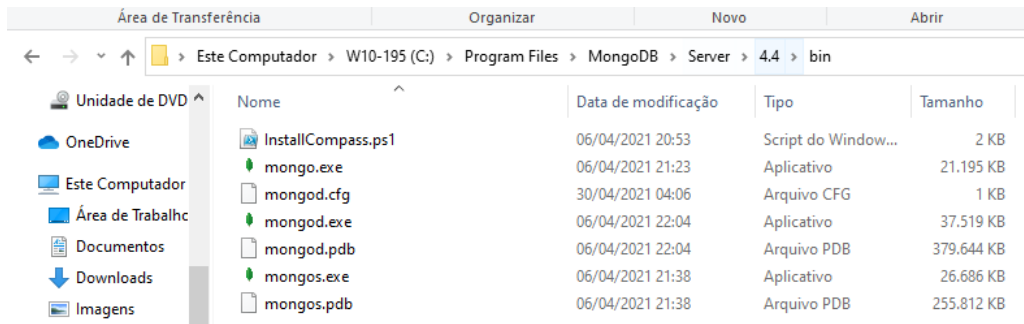
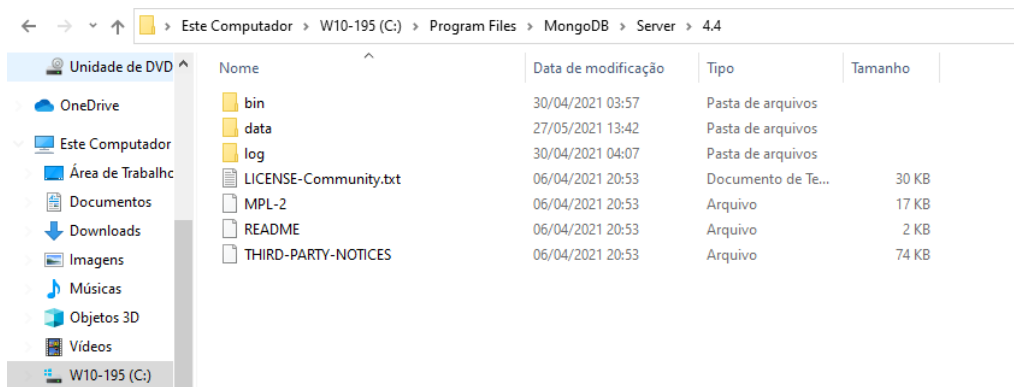
Platform
Windows

Package
msi

 [Download](#) [Copy Link](#)



```
C:\>mongo -h
MongoDB shell version v4.4.5
usage: mongo [options] [db address] [file names (ending in .js)]
db address can be:
    foo                foo database on local machine
    192.168.0.5/foo     foo database on 192.168.0.5 machine
    192.168.0.5:9999/foo foo database on 192.168.0.5 machine on port 9999
    mongodb://192.168.0.5:9999/foo connection string URI can also be used
Options:
  --ipv6                enable IPv6 support (disabled by default)
  --host arg            server to connect to
  --port arg            port to connect to
  -h [ --help ]        show this usage information
  --version             show version information
  --verbose             increase verbosity
  --shell              run the shell after executing files
  --nodb                don't connect to mongod on startup - no 'db address' arg expected
  --norc                will not run the ".mongorc.js" file on start up
  --quiet              be less chatty
  --eval arg            evaluate javascript
  --disableJavaScriptJIT disable the Javascript Just In Time compiler
  --enableJavaScriptJIT enable the Javascript Just In Time compiler
  --disableJavaScriptProtection allow automatic JavaScript function marshalling
  --retryWrites         automatically retry write operations upon transient network errors
  --disableImplicitSessions do not automatically create and use implicit sessions
  --jsHeapLimitMB arg   set the js scope's heap size limit
  --idleSessionTimeout arg (=0) Terminate the Shell session if it's been idle for this many seconds
```



```
C:\Users\beto1>mongo --version
MongoDB shell version v4.4.5
Build Info: {
  "version": "4.4.5",
  "gitVersion": "ff5cb77101b052fa02da43b8538093486cf9b3f7",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
```

- Em **C:/** crie uma pasta chamada "**data**":

Subindo o servidor do MongoDB

- No terminal do DOS, entre com o comando:

mongod

```
C:\>mongod
{"t":{"$date":"2021-05-27T13:58:41.207-03:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 spec
ify --sslDisabledProtocols 'none'"
{"t":{"$date":"2021-05-27T13:58:41.220-03:00"},"s":"W", "c":"ASIO", "id":22601, "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"
}
{"t":{"$date":"2021-05-27T13:58:41.221-03:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2021-05-27T13:58:41.224-03:00"},"s":"I", "c":"STORAGE", "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":8,"port":27017,"dbPa
th":"C:/data/db/","architecture":"64-bit","host":"DESKTOP-85HCHH0"}}
{"t":{"$date":"2021-05-27T13:58:41.224-03:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"ta
rgetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2021-05-27T13:58:41.224-03:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"4.4.5","
gitVersion":"ff5cb77101b052fa02da43b8538093486cf9b3f7","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"
}}}}
{"t":{"$date":"2021-05-27T13:58:41.224-03:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Wi
ndows 10","version":"10.0 (build 19042)}}}
{"t":{"$date":"2021-05-27T13:58:41.224-03:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{}}}
{"t":{"$date":"2021-05-27T13:58:41.230-03:00"},"s":"I", "c":"STORAGE", "id":22270, "ctx":"initandlisten","msg":"Storage engine to use detected by data files","attr
":{"dbpath":"C:/data/db/","storageEngine":"wiredTiger"}}
{"t":{"$date":"2021-05-27T13:58:41.232-03:00"},"s":"I", "c":"STORAGE", "id":22315, "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":{"create,cache_s
ize=1444M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),fi
le_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoint_progress,compact_progr
ess]}}}}
{"t":{"$date":"2021-05-27T13:58:41.490-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1622134721:4
90052][8:140717625333072], txn-recover: [WT_VERB_RECOVERY_PROGRESS] Recovering log 7 through 8"}}
{"t":{"$date":"2021-05-27T13:58:41.747-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1622134721:7
46124][8:140717625333072], txn-recover: [WT_VERB_RECOVERY_PROGRESS] Recovering log 8 through 8"}}
{"t":{"$date":"2021-05-27T13:58:42.004-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1622134722:3
960][8:140717625333072], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Main recovery loop: starting at 7/12160 to 8/256"}}
{"t":{"$date":"2021-05-27T13:58:42.505-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1622134722:5
05649][8:140717625333072], txn-recover: [WT_VERB_RECOVERY_PROGRESS] Recovering log 7 through 8"}}
{"t":{"$date":"2021-05-27T13:58:42.860-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1622134722:8
60837][8:140717625333072], txn-recover: [WT_VERB_RECOVERY_PROGRESS] Recovering log 8 through 8"}}
{"t":{"$date":"2021-05-27T13:58:43.108-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1622134723:1
07698][8:140717625333072], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)}}
{"t":{"$date":"2021-05-27T13:58:43.108-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1622134723:1
07698][8:140717625333072], txn-recover: [WT_VERB_RECOVERY | WT_VERB_RECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)}}
{"t":{"$date":"2021-05-27T13:58:43.144-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"WiredTiger message","attr":{"message":"[1622134723:1
44661][8:140717625333072], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 1, snapshot max: 1 snapshot count: 0, oldest timestamp:
(0, 0), meta checkpoint timestamp: (0, 0)}}
{"t":{"$date":"2021-05-27T13:58:43.643-03:00"},"s":"I", "c":"STORAGE", "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMillis":2410}}
{"t":{"$date":"2021-05-27T13:58:43.644-03:00"},"s":"I", "c":"RECOVERY", "id":23987, "ctx":"initandlisten","msg":"WiredTiger recoveryTimestamp","attr":{"recoveryTimes
tamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2021-05-27T13:58:43.649-03:00"},"s":"I", "c":"STORAGE", "id":4366408, "ctx":"initandlisten","msg":"No table logging settings modifications are required
for existing WiredTiger tables","attr":{"loggingEnabled":true}}
{"t":{"$date":"2021-05-27T13:58:43.653-03:00"},"s":"I", "c":"STORAGE", "id":22262, "ctx":"initandlisten","msg":"Timestamp monitor starting"}}
{"t":{"$date":"2021-05-27T13:58:43.741-03:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read
and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2021-05-27T13:58:43.742-03:00"},"s":"W", "c":"CONTROL", "id":22140, "ctx":"initandlisten","msg":"This server is bound to localhost. Remote systems wi
ll be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to
bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["startupWarnings"]}
{"t":{"$date":"2021-05-27T13:58:43.751-03:00"},"s":"I", "c":"STORAGE", "id":20536, "ctx":"initandlisten","msg":"Flow Control is enabled on this deployment"}}
{"t":{"$date":"2021-05-27T13:58:45.928-03:00"},"s":"W", "c":"FTDC", "id":23718, "ctx":"initandlisten","msg":"Failed to initialize Performance Counters for FTDC",
"attr":{"error":{"code":179,"codeName":"WindowsPdhError","errmsg":"PdhExpandCounterPathW failed with 'O objeto especificado não foi encontrado no computador.' for count
er '\\Processor(_Total)\\% Idle Time'"}}}
{"t":{"$date":"2021-05-27T13:58:45.928-03:00"},"s":"I", "c":"FTDC", "id":20625, "ctx":"initandlisten","msg":"Initializing full-time diagnostic data capture","att
r":{"dataDirectory":"C:/data/db/diagnostic.data"}}
{"t":{"$date":"2021-05-27T13:58:45.940-03:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"127.0.0.1"}}
{"t":{"$date":"2021-05-27T13:58:45.940-03:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting for connections","attr":{"port":27017,"ssl":"off"
}}
{"t":{"$date":"2021-05-27T13:59:43.658-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"WTCheckpointThread","msg":"WiredTiger message","attr":{"message":"[1622134
783:657528][8:140717625333072], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 3, snapshot max: 3 snapshot count: 0, oldest timest
amp: (0, 0), meta checkpoint timestamp: (0, 0)}}

```

- Indica que foi feita a conexão:

"Waiting for connections","attr":{"port": 27017, "ssl":"off"}

Aula 18 - CRUD MongoDB no Terminal - Parte 1

- Em um outro terminal do DOS, entre com o client do MongoDB com o comando:

mongo

```
C:\Users\beto1>mongo
MongoDB shell version v4.4.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongod
Implicit session: session { "id" : UUID("e59e3337-5b18-474a-af92-7ab16aa7160e") }
MongoDB server version: 4.4.5
---
The server generated these startup warnings when booting:
  2021-05-27T10:40:40.882-03:00: Access control is not enabled for the database. Read and write access to data and
  configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
  metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
  and anyone you share the URL with. MongoDB may use this information to make product
  improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

Comandos básicos

show databases;

```
> show databases;
admin            0.000GB
blogapp          0.000GB
celke            0.000GB
config           0.000GB
local            0.000GB
testes_mongo     0.000GB
>
```

use aula02;

```
> use aula02;
switched to db aula02
```

- Como o database **aula02** não existe, ele será criado, porém só será exibido quando conter dados.

- No MongoDB ao invés de tabelas, existem as collections.

- Inserindo um documento no database aula02:

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;
admin          0.000GB
aula02         0.000GB
blogapp        0.000GB
celke          0.000GB
config         0.000GB
local          0.000GB
testes_mongo   0.000GB
>
```

db.clientes.count();

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

db.clientes.find();

```
> db.clientes.find();
{ "_id" : ObjectId("60afee9861f2185e0cb8b2aa"), "nome" : "Luiz", "idade" : 29, "cidade" : "Gravataí" }
>
```

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

```
> db.clientes.find().pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
>
```

Aula 19 - CRUD MongoDB no Terminal - Parte 2

Adicionando clientes

`db.clientes.insert([{nome: "Fernando", idade: 25, cidade: "Porto Alegre"}, {nome: "Alberto", idade: 29, cidade: "Campinas"}, {nome: "Regina", idade: 32, uf: "SP"}]);`

```
> db.clientes.insert([{nome: "Fernando", idade: 25, cidade: "Porto Alegre"}, {nome: "Alberto", idade: 29, cidade: "Campinas"}, {nome: "Regina", idade: 32, uf: "SP"}]);
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 3,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
>
```

```
> db.clientes.find().pretty();
{
  "_id" : ObjectId("60afef9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Alberto",
  "idade" : 29,
  "cidade" : "Campinas"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ad"),
  "nome" : "Regina",
  "idade" : 32,
  "uf" : "SP"
}
>
```

Consultas

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

```
> db.clientes.find({idade:29}).pretty();
{
  "_id" : ObjectId("60afef9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Alberto",
  "idade" : 29,
  "cidade" : "Campinas"
}
>
```

```
db.clientes.find({nome:"Fernando"}).pretty();
```

```
> db.clientes.find({nome:"Fernando"}).pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
```

- As strings do MongoDB são case-sensitive.

```
db.clientes.find({nome:"fernando"}).pretty();
```

```
> db.clientes.find({nome:"fernando"}).pretty();
>
```

Busca por string com expressão regular

```
db.clientes.find({nome:/fernando/i}).pretty();
```

```
> db.clientes.find({nome:/fernando/i}).pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
```

Update

```
db.clientes.update({cidade:"Campinas"}, {nome:"Ricardo"});
```

```
> db.clientes.update({cidade:"Campinas"}, {nome:"Ricardo"});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

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

```
> db.clientes.find().pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"), "nome" : "Ricardo" }
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ad"),
  "nome" : "Regina",
  "idade" : 32,
  "uf" : "SP"
}
```

- Faz alterações no documento inteiro.

```
db.clientes.update({nome:"Ricardo"}, {nome:"Ricardo", idade: 29, cidade:"Campinas",uf:"SP"});
```

```
> db.clientes.update({nome:"Ricardo"}, {nome:"Ricardo", idade: 29, cidade:"Campinas",uf:"SP"});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.clientes.update({nome:"Ricardo"}, {nome:"Ricardo", idade: 29, cidade:"Campinas",uf:"SP"});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.clientes.find().pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ad"),
  "nome" : "Regina",
  "idade" : 32,
  "uf" : "SP"
}
>
```

Removendo um documento

```
db.clientes.remove({_id: ObjectId("60afef6d61f2185e0cb8b2ad")});
```

```
> db.clientes.remove({_id: ObjectId("60afef6d61f2185e0cb8b2ad")});
WriteResult({ "nRemoved" : 1 })
```

```
> db.clientes.find().pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
>
```

Aula 20 - Consultas com filtros em MongoDB

Busca por mais de um parâmetro

```
db.clientes.find({nome:"Fernando", idade:25}).pretty();
```

```
> db.clientes.find({nome:"Fernando", idade:25}).pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
```

Busca usando expressões regulares

Busca por nomes que contenham a letra "a"

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

```
> db.clientes.find({nome: {$regex: /a/ }}).pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
```

Busca por nomes que contenham a letra "i" ou "u"

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

```
> db.clientes.find({nome: {$regex: /[iu]/ }}).pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
```

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

```
> db.clientes.find({nome: /[iu]/ }).pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
>
```

Busca por idades menores ou iguais a 25

```
db.clientes.find({idade: {$lte: 25} }).pretty();
```

```
> db.clientes.find({idade: {$lte: 25} }).pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
>
```

Busca por idades maiores ou iguais a 29

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

```
> db.clientes.find({idade: {$gte: 29} }).pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
>
```

Busca por idades maiores que 29

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

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

Busca por idades iguais a 29

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

```
> db.clientes.find({idade: 29}).pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
>
```

OU:

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

```
> db.clientes.find({idade: {$eq: 29}}).pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
>
```

Busca por idades diferentes de 29

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

```
> db.clientes.find({idade: {$ne: 29}}).pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
>
```


Busca por idades com valores especificados em um array

```
db.clientes.find({idade: {$in: [23, 25, 29]} }).pretty();
```

```
> db.clientes.find({idade: {$in: [23, 25, 29]} }).pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
>
```

tags

- Insira os seguintes documentos:

```
db.clientes.insert([
  {nome: "Jorge", idade: 41, cidade: "Belo Horizonte", tags: ["python", "javascript", "ruby", "php"]},
  {nome: "Maria", idade: 33, cidade: "Florianópolis", tags: ["java", "php", "c"]},
  {nome: "Helena", idade: 48, cidade: "São Paulo", tags: ["javascript", "c++", "python", "java"]}
]);
```

```
> db.clientes.insert([
  {nome: "Jorge", idade: 41, cidade: "Belo Horizonte", tags: ["python", "javascript", "ruby", "php"]},
  {nome: "Maria", idade: 33, cidade: "Florianópolis", tags: ["java", "php", "c"]},
  {nome: "Helena", idade: 48, cidade: "São Paulo", tags: ["javascript", "c++", "python", "java"]}
]);
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 3,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
>
```

```
> db.clientes.find().pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
>
```

```
{
  "_id" : ObjectId("60affe6bb61f2185e0cb8b2ae"),
  "nome" : "Jorge",
  "idade" : 41,
  "cidade" : "Belo Horizonte",
  "tags" : [
    "python",
    "javascript",
    "ruby",
    "php"
  ]
}
{
  "_id" : ObjectId("60affe6bb61f2185e0cb8b2af"),
  "nome" : "Maria",
  "idade" : 33,
  "cidade" : "Florianópolis",
  "tags" : [
    "java",
    "php",
    "c"
  ]
}
{
  "_id" : ObjectId("60affe6bb61f2185e0cb8b2b0"),
  "nome" : "Helena",
  "idade" : 48,
  "cidade" : "São Paulo",
  "tags" : [
    "javascript",
    "c++",
    "python",
    "java"
  ]
}
>
```

Busca por clientes que contenham pelo menos uma das tags especificadas

```
db.clientes.find({tags: {$in: ["java", "php"]}}).pretty();
```

```
> db.clientes.find({tags: {$in: ["java", "php"]}}).pretty();
{
  "_id" : ObjectId("60affe6bb61f2185e0cb8b2ae"),
  "nome" : "Jorge",
  "idade" : 41,
  "cidade" : "Belo Horizonte",
  "tags" : [
    "python",
    "javascript",
    "ruby",
    "php"
  ]
}
{
  "_id" : ObjectId("60affe6bb61f2185e0cb8b2af"),
  "nome" : "Maria",
  "idade" : 33,
  "cidade" : "Florianópolis",
  "tags" : [
    "java",
    "php",
    "c"
  ]
}
{
  "_id" : ObjectId("60affe6bb61f2185e0cb8b2b0"),
  "nome" : "Helena",
  "idade" : 48,
  "cidade" : "São Paulo",
  "tags" : [
    "javascript",
    "c++",
    "python",
    "java"
  ]
}
>
```

Busca por clientes que contenham todas as tags especificadas

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

```
> db.clientes.find({tags: {$all: ["javascript", "python"]}}).pretty();
{
  "_id" : ObjectId("60affebb61f2185e0cb8b2ae"),
  "nome" : "Jorge",
  "idade" : 41,
  "cidade" : "Belo Horizonte",
  "tags" : [
    "python",
    "javascript",
    "ruby",
    "php"
  ]
}
{
  "_id" : ObjectId("60affebb61f2185e0cb8b2b0"),
  "nome" : "Helena",
  "idade" : 48,
  "cidade" : "São Paulo",
  "tags" : [
    "javascript",
    "c++",
    "python",
    "java"
  ]
}
>
```

Consulta de clientes ignorando os quatro primeiros documentos

```
db.clientes.find().pretty().skip(4);
```

```
> db.clientes.find().pretty().skip(4);
{
  "_id" : ObjectId("60affebb61f2185e0cb8b2af"),
  "nome" : "Maria",
  "idade" : 33,
  "cidade" : "Florianópolis",
  "tags" : [
    "java",
    "php",
    "c"
  ]
}
{
  "_id" : ObjectId("60affebb61f2185e0cb8b2b0"),
  "nome" : "Helena",
  "idade" : 48,
  "cidade" : "São Paulo",
  "tags" : [
    "javascript",
    "c++",
    "python",
    "java"
  ]
}
>
```

Consulta de apenas um documento (o primeiro)

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

```
> db.clientes.find().pretty().limit(1);
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
>
```

Exibindo apenas o segundo e terceiro documentos

```
db.clientes.find().pretty().skip(1).limit(2);
```

```
> db.clientes.find().pretty().skip(1).limit(2);
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
>
```

Exibindo clientes por ordem crescente de idade

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

```
> db.clientes.find().sort({idade: 1}).pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
{
  "_id" : ObjectId("60affe6b61f2185e0cb8b2af"),
  "nome" : "Maria",
  "idade" : 33,
  "cidade" : "Florianópolis",
  "tags" : [
    "java",
    "php",
    "c"
  ]
}
{
  "_id" : ObjectId("60affe6b61f2185e0cb8b2ae"),
  "nome" : "Jorge",
  "idade" : 41,
  "cidade" : "Belo Horizonte",
  "tags" : [
    "python",
    "javascript",
    "ruby",
    "php"
  ]
}
{
  "_id" : ObjectId("60affe6b61f2185e0cb8b2b0"),
  "nome" : "Helena",
  "idade" : 48,
  "cidade" : "São Paulo",
  "tags" : [
    "javascript",
    "c++",
    "python",
    "java"
  ]
}
>
```

Exibindo clientes por ordem decrescente de idade

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

```
> db.clientes.find().sort({idade: -1}).pretty();
{
  "_id" : ObjectId("60affebb61f2185e0cb8b2b0"),
  "nome" : "Helena",
  "idade" : 48,
  "cidade" : "São Paulo",
  "tags" : [
    "javascript",
    "c++",
    "phyton",
    "java"
  ]
}
{
  "_id" : ObjectId("60affebb61f2185e0cb8b2ae"),
  "nome" : "Jorge",
  "idade" : 41,
  "cidade" : "Belo Horizonte",
  "tags" : [
    "phyton",
    "javascript",
    "ruby",
    "php"
  ]
}
{
  "_id" : ObjectId("60affebb61f2185e0cb8b2af"),
  "nome" : "Maria",
  "idade" : 33,
  "cidade" : "Florianópolis",
  "tags" : [
    "java",
    "php",
    "c"
  ]
}
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 29,
  "cidade" : "Gravataí"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "uf" : "SP"
}
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ab"),
  "nome" : "Fernando",
  "idade" : 25,
  "cidade" : "Porto Alegre"
}
>
```

Aula 21 - Update com operadores no MongoDB

Atualizando um campo (\$set)

```
db.clientes.updateOne( {_id: ObjectId("60affe6b61f2185e0cb8b2af")}, {$set:{nome:"Márcia"}});
```

```
> db.clientes.updateOne( {_id: ObjectId("60affe6b61f2185e0cb8b2af")}, {$set:{nome:"Márcia"}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
```

```
db.clientes.find({_id: ObjectId("60affe6b61f2185e0cb8b2af")}).pretty();
```

```
> db.clientes.find({_id: ObjectId("60affe6b61f2185e0cb8b2af")}).pretty();
{
  "_id" : ObjectId("60affe6b61f2185e0cb8b2af"),
  "nome" : "Márcia",
  "idade" : 33,
  "cidade" : "Florianópolis",
  "tags" : [
    "java",
    "php",
    "c"
  ]
}
```

Eliminando um campo de um documento (\$unset)

```
db.clientes.updateOne( {_id: ObjectId("60affe6b61f2185e0cb8b2af")}, {$unset:{idade:null}});
```

```
> db.clientes.updateOne( {_id: ObjectId("60affe6b61f2185e0cb8b2af")}, {$unset:{idade:null}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
```

```
db.clientes.find({_id: ObjectId("60affe6b61f2185e0cb8b2af")}).pretty();
```

```
> db.clientes.updateOne( {_id: ObjectId("60affe6b61f2185e0cb8b2af")}, {$unset:{idade:null}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.clientes.find({_id: ObjectId("60affe6b61f2185e0cb8b2af")}).pretty();
{
  "_id" : ObjectId("60affe6b61f2185e0cb8b2af"),
  "nome" : "Márcia",
  "cidade" : "Florianópolis",
  "tags" : [
    "java",
    "php",
    "c"
  ]
}
```

Renomeando um campo (\$rename)

```
db.clientes.updateOne( {_id: ObjectId("60afef6d61f2185e0cb8b2ac")}, {$rename:{uf:"estado"}});
```

```
> db.clientes.updateOne( {_id: ObjectId("60afef6d61f2185e0cb8b2ac")}, {$rename:{uf:"estado"}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
```

```
db.clientes.find({_id: ObjectId("60afef6d61f2185e0cb8b2ac"))}.pretty();
```

```
> db.clientes.updateOne( {_id: ObjectId("60afef6d61f2185e0cb8b2ac")}, {$rename:{uf:"estado"}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.clientes.find({_id: ObjectId("60afef6d61f2185e0cb8b2ac"))}.pretty();
{
  "_id" : ObjectId("60afef6d61f2185e0cb8b2ac"),
  "nome" : "Ricardo",
  "idade" : 29,
  "cidade" : "Campinas",
  "estado" : "SP"
}
>
```

Incrementando a idade

```
db.clientes.updateOne( {_id: ObjectId("60afee9861f2185e0cb8b2aa")}, {$inc:{idade:1}} );
```

```
> db.clientes.updateOne( {_id: ObjectId("60afee9861f2185e0cb8b2aa")}, {$inc:{idade:1}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
```

```
db.clientes.find({_id: ObjectId("60afee9861f2185e0cb8b2aa"))}.pretty();
```

```
> db.clientes.find({_id: ObjectId("60afee9861f2185e0cb8b2aa"))}.pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 30,
  "cidade" : "Gravataí"
}
>
```

Decrementando a idade

```
db.clientes.updateOne( {_id: ObjectId("60afee9861f2185e0cb8b2aa")}, {$inc:{idade:-2}} );
```

```
> db.clientes.updateOne( {_id: ObjectId("60afee9861f2185e0cb8b2aa")}, {$inc:{idade:-2}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
```

```
db.clientes.find({_id: ObjectId("60afee9861f2185e0cb8b2aa"))}.pretty();
```

```
> db.clientes.find({_id: ObjectId("60afee9861f2185e0cb8b2aa"))}.pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 28,
  "cidade" : "Gravataí"
}
>
```

Multiplicando a idade por 2

```
db.clientes.updateOne( {_id: ObjectId("60afee9861f2185e0cb8b2aa")}, {$mul:{idade:2}} );
```

```
> db.clientes.updateOne( {_id: ObjectId("60afee9861f2185e0cb8b2aa")}, {$mul:{idade:2}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
```

```
db.clientes.find({_id: ObjectId("60afee9861f2185e0cb8b2aa"))}.pretty();
```

```
> db.clientes.find({_id: ObjectId("60afee9861f2185e0cb8b2aa"))}.pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 56,
  "cidade" : "Gravataí"
}
```

Dividindo a idade por 2

```
db.clientes.updateOne( {_id: ObjectId("60afee9861f2185e0cb8b2aa")}, {$mul:{idade:1/2}} );
```

```
> db.clientes.updateOne( {_id: ObjectId("60afee9861f2185e0cb8b2aa")}, {$mul:{idade:1/2}} );
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

```
db.clientes.find({_id: ObjectId("60afee9861f2185e0cb8b2aa"))}.pretty();
```

```
> db.clientes.find({_id: ObjectId("60afee9861f2185e0cb8b2aa"))}.pretty();
{
  "_id" : ObjectId("60afee9861f2185e0cb8b2aa"),
  "nome" : "Luiz",
  "idade" : 28,
  "cidade" : "Gravataí"
}
```

Update com Insert (upsert)

Atualiza um campo se o mesmo existir. Caso o campo não exista, ele será criado.

```
db.clientes.update( {nome:"Carlos"}, {cidade:"Curitiba"}, {upsert:true} );
```

```
> db.clientes.update( {nome:"Carlos"}, {cidade:"Curitiba"}, {$upsert:true} );
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
```

```
> db.clientes.update( {nome:"Carlos"}, {cidade:"Curitiba"}, {upsert:true} );
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 1,
  "nModified" : 0,
  "_id" : ObjectId("60b03cc0bc8995bb0a229c1b")
})
```

```
db.clientes.find({cidade:"Curitiba"}).pretty();
```

```
> db.clientes.find({cidade:"Curitiba"}).pretty();
{ "_id" : ObjectId("60b03cc0bc8995bb0a229c1b"), "cidade" : "Curitiba" }
```


Inserindo um documento e retornando o id (insertOne)

```
db.clientes.insertOne({nome:"Fábio",idade:36,cidade:"Natal"});
```

```
> db.clientes.insertOne({nome:"Fábio",idade:36,cidade:"Natal"});
{
  "acknowledged" : true,
  "insertedId" : ObjectId("60b03fc661f2185e0cb8b2b1")
}
```

```
db.clientes.find({cidade:"Natal"}).pretty();
```

```
> db.clientes.find({cidade:"Natal"}).pretty();
{
  "_id" : ObjectId("60b03fc661f2185e0cb8b2b1"),
  "nome" : "Fábio",
  "idade" : 36,
  "cidade" : "Natal"
}
```

Aula 22 - Nodejs conectando MongoDB

- Inicialmente, crie um novo projeto chamado "aula2". No terminal do DOS, entre com o comando:

`express -e --git aula2`

```
C:\luiztools>express -e --git aula2

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

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
```

`cd aula2`

`npm install`

```
C:\luiztools\aula2>npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
added 54 packages from 38 contributors and audited 55 packages in 13.27s
found 0 vulnerabilities
```

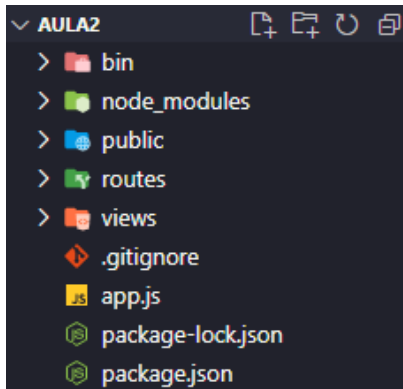
Instalando o driver mongodb

- Entre com o comando:

`npm install mongodb`

```
C:\luiztools\aula2>npm install mongodb
+ mongodb@3.6.9
added 14 packages from 9 contributors and audited 69 packages in 6.649s
found 0 vulnerabilities
```

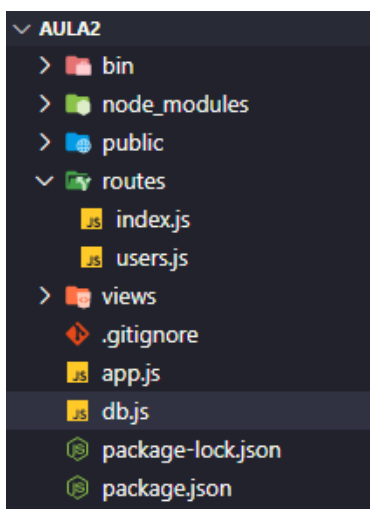
code .



package.json

```
{
  "name": "aula2",
  "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",
    "mongodb": "^3.6.9",
    "morgan": "~1.9.1"
  }
}
```

- No diretório raiz do projeto crie um arquivo chamado "db.js":



db.js

```
const MongoClient = require("mongodb").MongoClient;

mongoClient.connect("mongodb://localhost:27017",
{useUnifiedTopology: true},
(error, connection) => {
  if(error){
    return console.log(error);
  } else {
    global.connection = connection.db("aula2");
    console.log("Connected!");
  }
});

module.exports = {};
```

routes\index.js

```
const express = require('express');
const router = express.Router();
const db = require("../db");

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

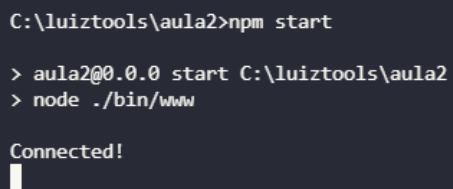
module.exports = router;
```

- Suba o servidor:

`mongod`

- No terminal do Visual Studio Code, entre com o comando:

`npm start`



```
C:\luiztools\aula2>npm start

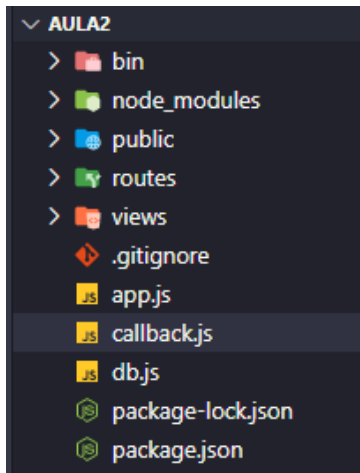
> aula2@0.0.0 start C:\luiztools\aula2
> node ./bin/www

Connected!
█
```

Callbacks, Promises e Async-Await no Node.js

- O Node trabalha de forma assíncrona.

Callbacks



- fs (filesystem) é o módulo para sistema de arquivos.
- O sistema de arquivo executa leituras e escritas em disco. Trata-se de uma operação lenta (especialmente escrever em disco).
- Enviar pacotes pela rede é outra operação lenta.

callback.js

```
const fs = require('fs');
```

```
var texto = 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vel mauris diam. In hac habitasse platea dictumst. Nam at lacinia eros. Proin ut facilisis tellus. Phasellus eget scelerisque ipsum. Donec hendrerit neque id consequat pretium. Aenean vel enim dignissim, venenatis risus auctor, semper lectus. Aenean massa elit, congue quis enim nec, fermentum aliquet risus. Mauris at enim et felis lobortis sollicitudin.';
```

```
// call back
```

```
console.log("\nPasso 1\n");
```

```
function funcAssinc(){  
  fs.writeFile('teste.txt', `${texto}`, () => {  
    console.log(texto);  
  });  
}
```

```
funcAssinc();
```

```
console.log("Passo 2\n");
```

```
console.log("Passo 3\n");
```

- O Node envia esse fluxo de execução para ser executado em background e continua a executar as demais linhas de programação do código.

node index.js

```
C:\luiztools\aula2>node callback.js

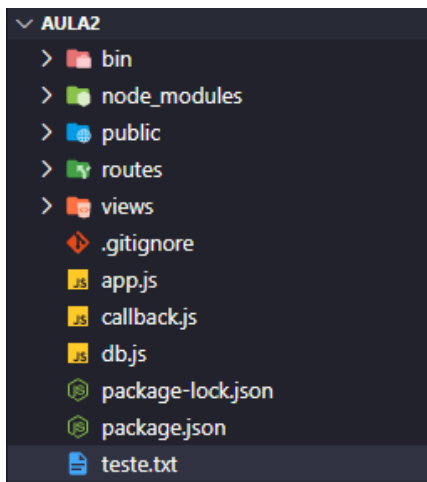
Passo 1

Passo 2

Passo 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vel mauris diam. In hac habitasse platea dictumst. Nam at lac inia eros. Proin ut facilisis tellus. Phasellus eget scelerisque ipsum. Donec hendrerit neque id consequat pretium. Aenean vel enim dignissim, venenatis risus auctor, semper lectus. Aenean massa elit, congue quis enim nec, fermentum aliquet risus. Mauris at enim et felis lobortis sollicitudin.
```

- O conteúdo da variável `texto` é gravado em um arquivo chamado `teste.txt`:



Callback hell

- O problema (em termos de estruturação de código) é quando dentro do callback existem outras funções lentas. Colocando um callback dentro de outro e assim por diante, o código fica longo e confuso.

callback.js

```
const fs = require('fs');
```

```
var texto = 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vel mauris diam. In hac habitasse platea dictumst. Nam at lacinia eros. Proin ut facilisis tellus. Phasellus eget scelerisque ipsum. Donec hendrerit neque id consequat pretium. Aenean vel enim dignissim, venenatis risus auctor, semper lectus. Aenean massa elit, congue quis enim nec, fermentum aliquet risus. Mauris at enim et felis lobortis sollicitudin.';
```

```
var texto2 = 'Aenean non est arcu. Curabitur volutpat pellentesque dolor eu euismod. Etiam quis vulputate nunc. Quisque ac commodo tortor, vel fermentum metus. Aliquam auctor, quam sit amet rutrum feugiat, massa elit rhoncus felis, quis dictum ante mi in velit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Ut consectetur augue ullamcorper felis consequat vulputate. Phasellus vehicula, massa et hendrerit congue, ligula lacus consequat augue, bibendum maximus ipsum felis vel ex. Aliquam erat volutpat. Curabitur pulvinar sagittis ligula, a finibus tellus tempor et. Maecenas leo mi, iaculis vitae porttitor vel, tincidunt et erat. Cras non scelerisque dui.';
```

```
var texto3 = 'In et consequat justo. Nulla vel porta eros, non consectetur ligula. Pellentesque arcu massa, interdum a elementum et, ultricies in felis. Duis laoreet mi turpis, eu luctus urna viverra sed. Donec quis cursus nibh. Suspendisse potenti. Vivamus laoreet nisl sed augue mattis, efficitur semper diam tincidunt. Nullam iaculis, metus quis varius aliquam, velit eros rutrum erat, et finibus nunc quam at ex. Suspendisse cursus ullamcorper erat eget rhoncus. Ut eu sollicitudin dui. Donec auctor eleifend dolor pharetra tempus. Phasellus ac commodo ipsum.';
```

```
// call back
```

```
console.log("\nPasso 1\n");
```

```
function funcaoAssinc(){
  fs.writeFile('teste.txt', `${texto}`, () => {
    console.log(texto);
    console.log("");
    fs.writeFile('teste2.txt', `${texto2}`, () => {
      console.log(texto2);
      console.log("");
      fs.writeFile('teste3.txt', `${texto3}`, () => {
        console.log(texto3);
        console.log("");
      })
    });
  });
}
```

```
funcaoAssinc();
```

```
console.log("Passo 2\n");
```

```
console.log("Passo 3\n");
```

node callback.js

```
C:\luiztools\aula2>node callback.js
```

Passo 1

Passo 2

Passo 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vel mauris diam. In hac habitasse platea dictumst. Nam at lac inia eros. Proin ut facilisis tellus. Phasellus eget scelerisque ipsum. Donec hendrerit neque id consequat pretium. Aenean vel enim dignissim, venenatis risus auctor, semper lectus. Aenean massa elit, congue quis enim nec, fermentum aliquet risus. Mauris at enim et felis lobortis sollicitudin.

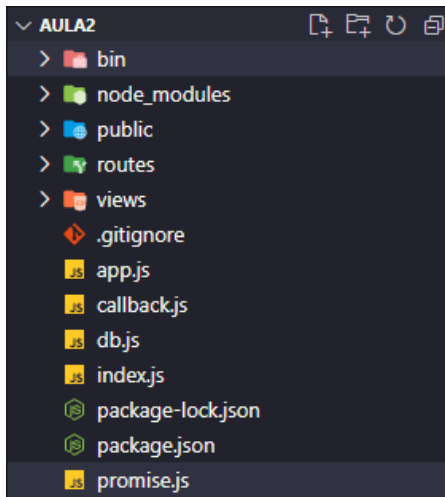
Aenean non est arcu. Curabitur volutpat pellentesque dolor eu euismod. Etiam quis vulputate nunc. Quisque ac commodo tortor, vel fermentum metus. Aliquam auctor, quam sit amet rutrum feugiat, massa elit rhoncus felis, quis dictum ante mi in velit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Ut consectetur augue ullamcorper felis consequat vulputate. Phasellus vehicula, massa et hendrerit congue, ligula lacus consequat augue, bibendum maximus ipsum felis vel ex. Aliquam erat volutpat. Curabitur pulvinar sagittis ligula, a finibus tellus tempor et. Maecenas leo mi, iaculis vitae porttitor vel, tincidunt et erat. Cras non scelerisque dui.

In et consequat justo. Nulla vel porta eros, non consectetur ligula. Pellentesque arcu massa, interdum a elementum et, ultricies in felis. Duis laoreet mi turpis, eu luctus urna viverra sed. Donec quis cursus nibh. Suspendisse potenti. Vivamus laoreet nisl sed augue mattis, efficitur semper diam tincidunt. Nullam iaculis, metus quis varius aliquam, velit eros rutrum erat, et finibus nunc quam at ex. Suspendisse cursus ullamcorper erat eget rhoncus. Ut eu sollicitudin dui. Donec auctor eleifend dolor pharetra tempus. Phasellus ac commodo ipsum.

▼ AULA2

- > bin
- > node_modules
- > public
- > routes
- > views
- .gitignore
- app.js
- callback.js
- db.js
- package-lock.json
- package.json
- teste.txt
- teste2.txt
- teste3.txt

Promise



- É uma forma mais elegante de se escrever um callback;

promise.js

```
// promise
```

```
console.log("\nPasso 1\n");
```

```
function fazRequisicao(resolver = true) {  
  return new Promise((resolve, reject) => {  
    setTimeout(() => {  
      if (!resolver) {  
        reject("Deu erro!");  
      }  
      resolve("Promise resolvida!");  
    }, 5000);  
  });  
}
```

```
fazRequisicao(false)  
  .then(console.log)  
  .catch(console.error);
```

```
console.log("Passo 2\n");  
console.log("Passo 3\n");
```

node promise.js

```
C:\luiztools\aula2>node promise.js  
  
Passo 1  
  
Passo 2  
  
Passo 3  
  
Deu erro!
```

promise.js

```
// promise

console.log("\nPasso 1\n");

function fazRequisicao(resolver = true) {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (!resolver) {
        reject("Deu erro!");
      }
      resolve("Promise resolvida!");
    }, 5000);
  });
}

fazRequisicao()
  .then(console.log)
  .catch(console.error);

console.log("Passo 2\n");
console.log("Passo 3\n");
```

node promise.js

```
C:\luiztools\aula2>node promise.js

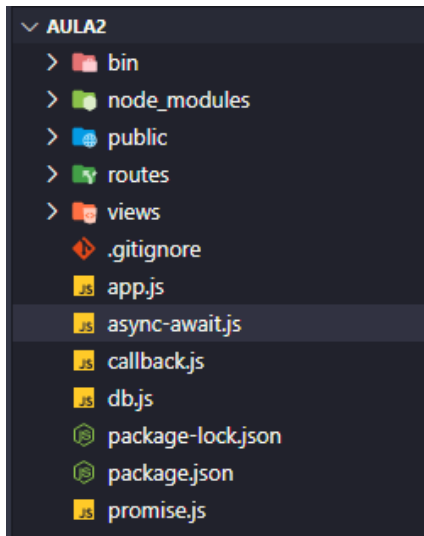
Passo 1

Passo 2

Passo 3

Promise resolvida!
```

Async/Await



async-await.js

```
const fs = require('fs');
```

```
var texto = 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vel mauris diam. In hac habitasse platea dictumst. Nam at lacinia eros. Proin ut facilisis tellus. Phasellus eget scelerisque ipsum. Donec hendrerit neque id consequat pretium. Aenean vel enim dignissim, venenatis risus auctor, semper lectus. Aenean massa elit, congue quis enim nec, fermentum aliquet risus. Mauris at enim et felis lobortis sollicitudin.';
```

```
var texto2 = 'Aenean non est arcu. Curabitur volutpat pellentesque dolor eu euismod. Etiam quis vulputate nunc. Quisque ac commodo tortor, vel fermentum metus. Aliquam auctor, quam sit amet rutrum feugiat, massa elit rhoncus felis, quis dictum ante mi in velit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Ut consectetur augue ullamcorper felis consequat vulputate. Phasellus vehicula, massa et hendrerit congue, ligula lacus consequat augue, bibendum maximus ipsum felis vel ex. Aliquam erat volutpat. Curabitur pulvinar sagittis ligula, a finibus tellus tempor et. Maecenas leo mi, iaculis vitae porttitor vel, tincidunt et erat. Cras non scelerisque dui.';
```

```
var texto3 = 'In et consequat justo. Nulla vel porta eros, non consectetur ligula. Pellentesque arcu massa, interdum a elementum et, ultricies in felis. Duis laoreet mi turpis, eu luctus urna viverra sed. Donec quis cursus nibh. Suspendisse potenti. Vivamus laoreet nisl sed augue mattis, efficitur semper diam tincidunt. Nullam iaculis, metus quis varius aliquam, velit eros rutrum erat, et finibus nunc quam at ex. Suspendisse cursus ullamcorper erat eget rhoncus. Ut eu sollicitudin dui. Donec auctor eleifend dolor pharetra tempus. Phasellus ac commodo ipsum.';
```

```
console.log("\nPasso 1\n");
```

```
async function funcaoAssinc(){
  try {
    await fs.writeFile('teste.txt', `${texto}`, () => {
      console.log(texto);
      console.log("");
    });
    await fs.writeFile('teste2.txt', `${texto2}`, () => {
      console.log(texto2);
      console.log("");
    });
    await fs.writeFile('teste3.txt', `${texto3}`, () => {
      console.log(texto3);
      console.log("");
    });
  }
}
```

```
    } catch (error) {  
        console.log(error);  
    }  
}
```

```
funcaoAssinc();
```

```
console.log("Passo 2\n");
```

```
console.log("Passo 3\n");
```

node async-await.js

```
C:\luiztools\aula2>node async-await.js
```

```
Passo 1
```

```
Passo 2
```

```
Passo 3
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vel mauris diam. In hac habitasse platea dictumst. Nam at lac inia eros. Proin ut facilisis tellus. Phasellus eget scelerisque ipsum. Donec hendrerit neque id consequat pretium. Aenean vel enim dignissim, venenatis risus auctor, semper lectus. Aenean massa elit, congue quis enim nec, fermentum aliquet risus. Mauris at enim et felis lobortis sollicitudin.

Aenean non est arcu. Curabitur volutpat pellentesque dolor eu euismod. Etiam quis vulputate nunc. Quisque ac commodo tortor, vel fermentum metus. Aliquam auctor, quam sit amet rutrum feugiat, massa elit rhoncus felis, quis dictum ante mi in velit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Ut consectetur augue ullamcorper felis consequat vulputate. Phasellus vehicula, massa et hendrerit congue, ligula lacus consequat augue, bibendum maximus ipsum felis vel ex. Aliquam erat volutpat. Curabitur pulvinar sagittis ligula, a finibus tellus tempor et. Maecenas leo mi, iaculis vitae porttitor vel, tincidunt et erat. Cras non scelerisque dui.

In et consequat justo. Nulla vel porta eros, non consectetur ligula. Pellentesque arcu massa, interdum a elementum et, ultricies in felis. Duis laoreet mi turpis, eu luctus urna viverra sed. Donec quis cursus nibh. Suspendisse potenti. Vivamus laoreet nisl sed augue mattis, efficitur semper diam tincidunt. Nullam iaculis, metus quis varius aliquam, velit eros rutrum erat, et finibus nunc quam at ex. Suspendisse cursus ullamcorper erat eget rhoncus. Ut eu sollicitudin dui. Donec auctor eleifend dolor pharetra tempus. Phasellus ac commodo ipsum.