

Curso de Node.JS

Marcos Rivas (vidamrr)

Vídeos:

https://www.youtube.com/watch?v=APbb-z8A3Zc&list=PLvRPaExkZHF_n_EGzNOXsWqtAAGl8ZCbUo

Repositório Github:

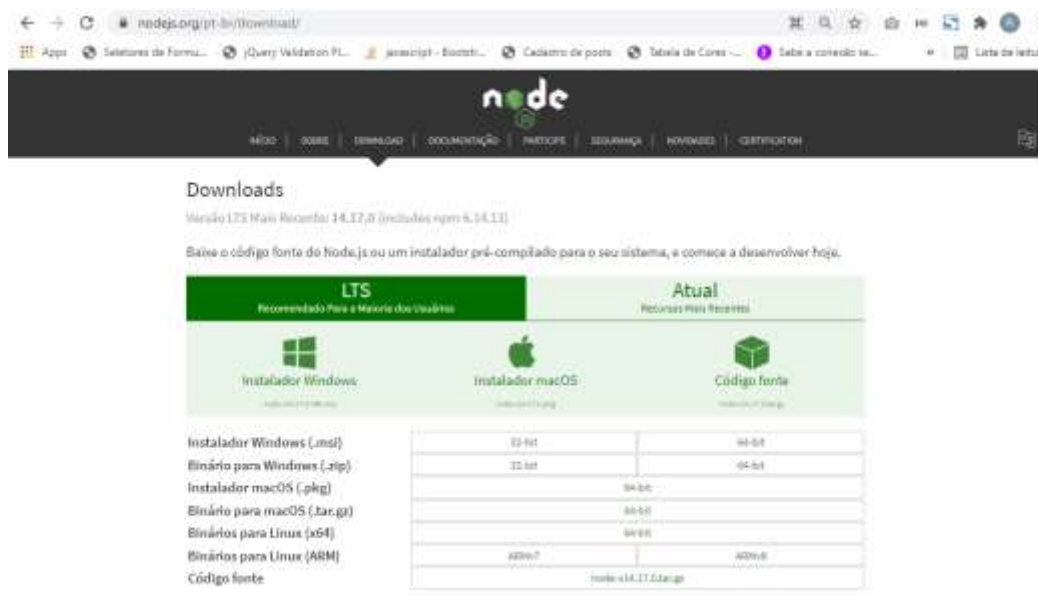
<https://github.com/marcosrivasr/Curso-de-NodeJS>

Resumo do curso feito por Roberto Pinheiro

Aula 01 - Introdução

- Baixe e instale o Node

<https://nodejs.org/pt-br/download/>



```
C:\marcos_rivas>node
Welcome to Node.js v14.15.3.
Type ".help" for more information.
> a = 20
20
> if(a > 18){
... console.log('Maior de idade')
... } else {
... console.log('Menor de idade')
... }
Maior de idade
undefined
>
```

md basico

cd basico

code .

```
C:\marcos_rivas>md basico
C:\marcos_rivas>cd basico
C:\marcos_rivas\basico>code .
```

01-instalacao\app.js

```
process.stdout.write("Olá\n");

var nome = "Marcos Rivas";

function saudar(){
    console.log("Seja bem-vindo, " + nome + "!");
};

saudar();
```

node app.js

```
C:\marcos_rivas\basico\01-instalacao>node app.js
Olá
Seja bem-vindo, Marcos Rivas!
```

Aula 02 - Exercícios básicos

node -v

```
C:\marcos_rivas>node -v
v14.15.3
```

02-primeiros_passos\basicos.js

```
var nome = 'Roberto'
var path = require('path')

console.log(nome)

var a = 5;
var b = 10;

soma = a + b;
produto = a * b;

console.log('Soma: ' + soma);
console.log(`Produto: ${produto}`);

// Variáveis globais

console.log(__dirname);
console.log(__filename);
console.log(path.basename(__filename));
```

node basicos.js

```
C:\marcos_rivas\basico\02-primeiros_passos>node basicos
Roberto
Soma: 15
Produto: 50
C:\marcos_rivas\basico\02-primeiros_passos
C:\marcos_rivas\basico\02-primeiros_passos\basicos.js
basicos.js
```

Aula 03 - Processos

03-processos\processos.js

```
console.log(process.argv);
```

```
node processos --nome "Marcos" --idade 25
```

```
C:\marcos_rivas\basico\03-processos>node processos --nome "Marcos" --idade 25
[
  'C:\\Program Files\\nodejs\\node.exe',
  'C:\\marcos_rivas\\basico\\03-processos\\processos',
  '--nome',
  'Marcos',
  '--idade',
  '25'
]
```

03-processos\processos.js

```
function param(p){
  var index = process.argv.indexOf(p);
  return process.argv[index + 1];
};
```

```
var nome = param('--nome');
var idade = param('--idade');
```

```
console.log(`Teu nome é ${nome}. Você tem ${idade} anos de idade.`);
```

```
node processos.js --nome "Marcos" --idade 25
```

```
C:\marcos_rivas\basico\03-processos>node processos.js --nome "Marcos" --idade 25
Teu nome é Marcos. Você tem 25 anos de idade.
```

Aula 04 - Ler e escrever no console

Processos stdout e stdin

04-stdin_stdout\app.js

```
process.stdout.write("Olá mundo!\n")
```

node app.js

```
C:\marcos_rivas\basico\04-stdin_stdout>node app.js
Olá mundo!
```

04-stdin_stdout\app.js

```
// Processos stdout e stdin
```

```
var perguntas = ['Qual é o seu nome? ',
                 'Qual é a sua idade? ',
                 'Qual é a sua linguagem de programação favorita? '];
```

```
var respostas = [];
```

```
function pergunta(i){
  process.stdout.write(perguntas[i]);
};
```

```
process.stdin.on('data', function(data){
  respostas.push(data.toString().trim());
```

```
  if(respostas.length < perguntas.length){
    pergunta(respostas.length);
  } else {
    process.exit();
  }
});
```

```
pergunta(0);
```

node app.js

```
C:\marcos_rivas\basico\04-stdin_stdout>node app.js
Qual é o seu nome? Roberto
Qual é a sua idade? 60
Qual é a sua linguagem de programação favorita? php
```

Aula 05 - Como funcionam os módulos

Documentação do Node.JS

<https://nodejs.org/api/>



Módulo path

05-modulos\path.js

```
// modulo path
```

```
var path = require('path');
```

```
console.log(path.basename(__filename));
```

```
console.log(__filename);
```

```
console.log(path.join(__dirname, 'www', 'img', 'home', 'uploads'));
```

```
console.log(path.basename('/foo/bar/baz/asdf/quux.html')); // Returns: 'quux.html'
```

node path.js

```
C:\marcos_rivas\basico\05-modulos>node path.js
path.js
C:\marcos_rivas\basico\05-modulos\path.js
C:\marcos_rivas\basico\05-modulos\www\img\home\uploads
quux.html
```

Módulo util

05-modulos\util.js

```
// modulo util

var util = require('util');

util.log("Hello world!");

var nome = "Marcos";
var idade = 25;

var texto = util.format("Olá %s! Sua idade é %d anos", nome, idade);
console.log(texto);
```

node util.js

```
C:\marcos_rivas\basico\05-modulos>node util.js
8 Jun 11:02:07 - Hello world!
Olá Marcos! Sua idade é 25 anos
```

Módulo v8

05-modulos\v8.js

```
var v8 = require('v8');
console.log(v8.getHeapStatistics());
```

node v8.js

```
C:\marcos_rivas\basico\05-modulos>node v8.js
{
  total_heap_size: 4468736,
  total_heap_size_executable: 524288,
  total_physical_size: 4468736,
  total_available_size: 2098335608,
  used_heap_size: 2855056,
  heap_size_limit: 2101608448,
  malloced_memory: 8192,
  peak_malloced_memory: 123168,
  does_zap_garbage: 0,
  number_of_native_contexts: 1,
  number_of_detached_contexts: 0
}
```

Aula 06 - Uso do módulo Readline

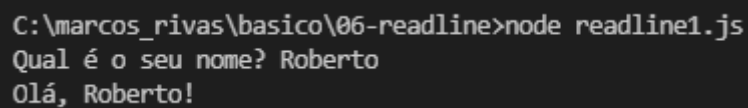
06-readline\readline1.js

```
var readline = require('readline');

var rl = readline.createInterface(process.stdin, process.stdout);

rl.question('Qual é o seu nome? ', (resposta) => {
  console.log(`Olá, ${resposta}!`);
  process.exit();
});
```

node readline1.js



```
C:\marcos_rivas\basico\06-readline>node readline1.js
Qual é o seu nome? Roberto
Olá, Roberto!
```

06-readline\readline2.js

```
var readline = require('readline');
var util = require('util');

var rl = readline.createInterface(process.stdin, process.stdout);

var pessoa = {
  nome: "",
  comentarios: []
};

rl.question('Qual é o seu nome? ', (resposta) => {
  pessoa.nome = resposta;

  rl.setPrompt('Deixe um comentário: ');
  rl.prompt();
});

rl.on('line', (input) =>{
  if(input.trim() === 'sair'){
    var message = util.format("Seu nome é %s. Seus comentários: %j", pessoa.nome, pessoa.comentarios);
    console.log(message);
    process.exit();
  }

  pessoa.comentarios.push(input.trim());

  rl.setPrompt('Deixe um comentário: ');
  rl.prompt();
});
```


node readline2.js

```
C:\marcos_rivas\basico\06-readline>node readline2.js
Qual é o seu nome? Roberto Pinheiro
Deixe um comentário: Gostei do curso!
Deixe um comentário: Muito bom!
Deixe um comentário: Parabéns!
Deixe um comentário: sair
Seu nome é Roberto Pinheiro. Seus comentários: ["Gostei do curso!","Muito bom!","Parabéns!"]
```

Aula 07 - Como exportar módulos

07-eventos\libs\pessoa.js

```
const EventEmitter = require('events').EventEmitter;
const util = require('util');

var Pessoa = function(nome){
  this.nome = nome;
}

util.inherits(Pessoa, EventEmitter);

module.exports = Pessoa;
```

07-eventos\evento.js

```
const pessoa = require('./libs/pessoa');

let pablo = new pessoa('Pablo');

pablo.on('fala', (message) => {
  console.log(`${pablo.nome}: ${message}`);
});

pablo.emit('fala', 'Hoje será um grande dia!');
```

node evento.js

```
C:\marcos_rivas\basico\07-eventos>node evento.js
Pablo: Hoje será um grande dia!
```

Aula 08 - Manipulação de eventos

08-manipulando_eventos\evento1.js

```
const events = require('events');
const emitter = new events.EventEmitter();

emitter.on('eventoCustom', (message, status) => {
  console.log(`{status}: ${message}`);
});

emitter.emit('eventoCustom', 'Mensagem carregada com sucesso!', 200);
```

node evento1.js

```
C:\marcos_rivas\basico\08-manipulando_eventos>node evento1.js
200: Mensagem carregada com sucesso!
```

08-manipulando_eventos\evento2.js

```
const EventEmitter = require('events').EventEmitter

var Pessoa = function (nome){
  this.nome = nome;
};

let pessoa = new Pessoa('Bob');
console.log(`Me chamo ${pessoa.nome}`);
```

node evento2.js

```
C:\marcos_rivas\basico\08-manipulando_eventos>node evento2.js
Me chamo Bob
```

08-manipulando_eventos\evento3.js

```
const EventEmitter = require('events').EventEmitter
const util = require('util');

var Pessoa = function (nome){
  this.nome = nome;
};

util.inherits(Pessoa, EventEmitter);

let pessoa = new Pessoa('Bob');

// console.log(`Me chamo ${pessoa.nome}`);

pessoa.on('falar', (message) => {
  console.log(`${pessoa.nome}: ${message}`);
});

pessoa.emit('falar', 'Hoje é um grande dia!');
```

node evento3.js

```
C:\marcos_rivas\basico\08-manipulando_eventos>node evento3.js
Bob: Hoje é um grande dia!
```

Aula 09 - Executar processos em NodeJS

09-executar_processos\processo1.js

```
const exec = require('child_process').exec

exec('dir /b', (err, stdout) =>{
  if(err){
    throw err;
  }

  console.log('Comando executado!');
  console.log(stdout);
});
```

node processo1.js

```
C:\marcos_rivas\basico\09-executar_processos>node processo1.js
Comando executado!
processo1.js
processo2.js
script.sh
```

09-executar_processos\processo2.js

```
const exec = require('child_process').exec

exec('ls', (err, stdout) =>{
  if(err){
    throw err;
  }

  console.log('Comando executado!');
  console.log(stdout);
});
```

node processo2.js

```
betoi@DESKTOP-NUIKR9E MINGW64 /c/marcos_rivas/basico/09-executar_processos
$ node processo2.js
Comando executado!
processo1.js
processo2.js
processo3.js
script.sh
```

09-executar_processos\processo3.js

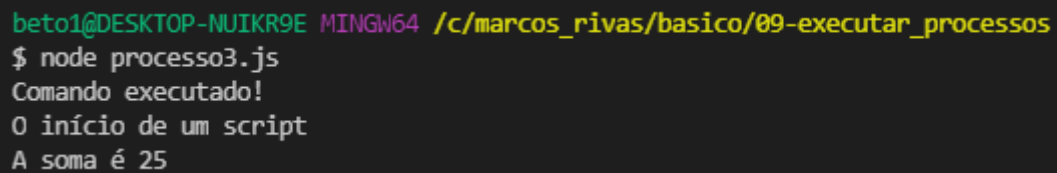
```
const exec = require('child_process').exec

const x = 10;
const y = 15;

exec('sh script.sh ' + x + ' ' + y, (err, stdout) =>{
  if(err){
    throw err;
  }

  console.log('Comando executado!');
  console.log(stdout);
});
```

node processo3.js

A terminal window with a dark background. The prompt is 'beto1@DESKTOP-NUIKR9E MINGW64 /c/marcos_rivas/basico/09-executar_processos'. The user enters '\$ node processo3.js'. The output shows 'Comando executado!', 'O início de um script', and 'A soma é 25'.

```
beto1@DESKTOP-NUIKR9E MINGW64 /c/marcos_rivas/basico/09-executar_processos
$ node processo3.js
Comando executado!
O início de um script
A soma é 25
```

Aula 10 - Ler arquivos com NodeJS

10-ler_arquivos\arquivo.txt

Este é um arquivo de texto plano. Se pode ler, pode abri-lo.

10-ler_arquivos\read1.js

```
// leitura de forma síncrona
```

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

```
var files = fs.readdirSync('./');  
console.log(files);
```

```
var arquivo = fs.readFileSync('./arquivo.txt', 'utf8');  
console.log('Conteúdo do arquivo...');  
console.log(arquivo);
```

node read1.js

```
C:\marcos_rivas\basico\10-ler_arquivos>node read1.js  
[ 'arquivo.txt', 'read1.js', 'read2.js', 'read3.js', 'read4.js' ]  
Conteúdo do arquivo...  
Este é um arquivo de texto plano. Se o pode ler, pode abri-lo.
```

10-ler_arquivos\read2.js

// leitura de forma assíncrona

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

```
fs.readdir('./', (error, files) => {  
  if(error){  
    throw(error);  
  }  
  console.log(files);  
})
```

```
fs.readFile('./arquivo.txt', 'utf8', (error, arquivo) => {  
  if(error){  
    throw(error);  
  }  
  console.log(arquivo);  
}  
console.log('Conteúdo do arquivo...');  
});
```

node read2.js

```
C:\marcos_rivas\basico\10-ler_arquivos>node read2.js  
[ 'arquivo.txt', 'read1.js', 'read2.js' ]  
Conteúdo do arquivo...  
Este é um arquivo de texto plano. Se pode ler, pode abri-lo.
```


Aula 11 - Escrever arquivos com NodeJS

11-escrever_arquivos\write1.js

```
// validar, criar e/ou escrever em um arquivo (de forma síncrona)

const fs = require('fs');

const arquivo = 'prova.txt';

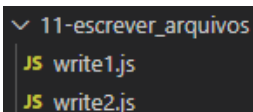
const conteudo = 'Este é o conteúdo de um texto';
const textoAdicional = "\nAqui foi adicionada uma nova linha de código!";

escrever();

function escrever(){
  if(fs.existsSync(arquivo)){
    console.log('O arquivo existe!');

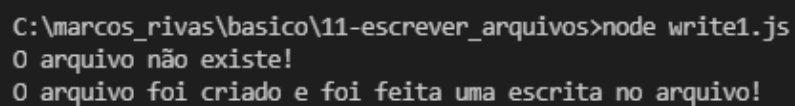
    // escrever uma linha no final do arquivo (de forma síncrona)
    fs.appendFileSync(arquivo, textoAdicional);
    console.log('Foi acrescentada uma nova linha no final do arquivo!');
  } else {
    console.log('O arquivo não existe!');

    // criar e escrever em um arquivo (de forma síncrona)
    fs.writeFileSync(arquivo, conteudo);
    console.log('O arquivo foi criado e foi feita uma escrita no arquivo!');
  }
}
```

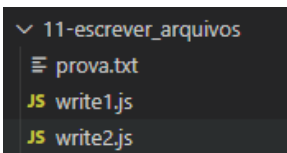


```
11-escrever_arquivos
├── write1.js
└── write2.js
```

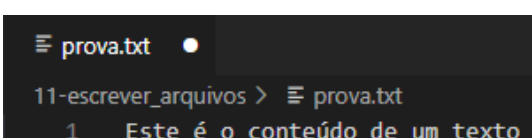
node write1.js



```
C:\marcos_rivas\basico\11-escrever_arquivos>node write1.js
O arquivo não existe!
O arquivo foi criado e foi feita uma escrita no arquivo!
```



```
11-escrever_arquivos
├── prova.txt
├── write1.js
└── write2.js
```



```
prova.txt
11-escrever_arquivos > prova.txt
1 Este é o conteúdo de um texto
```

node write1.js

node write1.js

```
C:\marcos_rivas\basico\11-escrever_arquivos>node write1.js
0 arquivo existe!
Foi acrescentada uma nova linha no final do arquivo!

C:\marcos_rivas\basico\11-escrever_arquivos>node write1.js
0 arquivo existe!
Foi acrescentada uma nova linha no final do arquivo!
```

≡ *prova.txt* ✕

11-escrever_arquivos > ≡ *prova.txt*

```
1 Este é o conteúdo de um texto
2 Aqui foi adicionada uma nova linha de código!
3 Aqui foi adicionada uma nova linha de código!
```

11-escrever_arquivos\write2.js

// validar, criar e/ou escrever em um arquivo (de forma assíncrona)

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

```
const arquivo = 'prova.txt';
```

```
const conteudo = 'Este é o conteúdo de um texto';
```

```
const textoAdicional = "\nAqui foi adicionada uma nova linha de código!";
```

```
escrever();
```

```
function escrever(){
```

```
  fs.access(arquivo, fs.constants.F_OK, (err) => {
```

```
    if(err){
```

```
      console.log('O arquivo não existe!');
```

```
      // criar e escrever em um arquivo (de forma assíncrona)
```

```
      fs.writeFile(arquivo, conteudo, (err) =>{
```

```
        if(err){
```

```
          throw('Ocorreu um erro ao escrever no arquivo!');
```

```
        } else {
```

```
          console.log('O arquivo foi criado e foi feita uma escrita no arquivo!');
```

```
        }
```

```
      });
```

```
    } else {
```

```
      console.log('O arquivo existe!');
```

```
      // escrever uma linha no final do arquivo (de forma assíncrona)
```

```
      fs.appendFile(arquivo, textoAdicional, (err) => {
```

```
        console.log('Foi acrescentada uma nova linha no final do arquivo!');
```

```
      });
```

```
    }
```

```
  })
```

```
}
```

✓ 11-escrever_arquivos

JS write1.js

JS write2.js

node write2.js

```
C:\marcos_rivas\basico\11-escrever_arquivos>node write2.js
O arquivo não existe!
O arquivo foi criado e foi feita uma escrita no arquivo!
```

```
▼ 11-escrever_arquivos
  ≡ prova.txt
  JS write1.js
  JS write2.js
```

```
≡ prova.txt X
11-escrever_arquivos > ≡ prova.txt
  1 Este é o conteúdo de um texto
```

node write2.js

node write2.js

```
C:\marcos_rivas\basico\11-escrever_arquivos>node write2.js
O arquivo existe!
Foi acrescentada uma nova linha no final do arquivo!

C:\marcos_rivas\basico\11-escrever_arquivos>node write2.js
O arquivo existe!
Foi acrescentada uma nova linha no final do arquivo!
```

```
≡ prova.txt X
11-escrever_arquivos > ≡ prova.txt
  1 Este é o conteúdo de um texto
  2 Aqui foi adicionada uma nova linha de código!
  3 Aqui foi adicionada uma nova linha de código!
```

Aula 12 - Usar diretórios em NodeJS

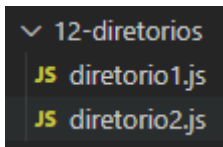
12-diretorios\diretorio1.js

```
// criar diretório de forma síncrona

const fs = require('fs');

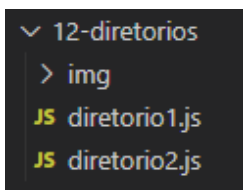
// validar se existe um diretório

if(fs.existsSync('img')){
  console.log('O diretório já existe!');
} else {
  console.log('O diretório não existe!');
  // criando o diretório
  fs.mkdirSync('img');
  console.log('Diretório criado com sucesso');
}
```



node diretorio1.js

```
C:\marcos_rivas\basico\12-diretorios>node diretorio1.js
O diretório não existe!
Diretório criado com sucesso
```



node diretorio1.js

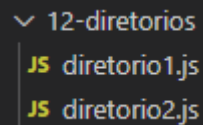
```
C:\marcos_rivas\basico\12-diretorios>node diretorio1.js
O diretório já existe!
```

12-diretorios\diretorio2.js

// criando diretório de forma assíncrona

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

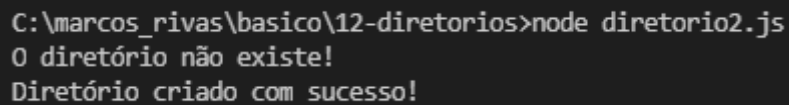
if (fs.existsSync('css')) {
  console.log('O diretório já existe!');
} else {
  console.log('O diretório não existe!')
  // criando o diretório
  fs.mkdir('css', function (err) {
    if(err){
      throw(err);
    }
    console.log('Diretório criado com sucesso!');
  });
}
```



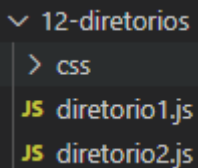
12-diretorios

- JS diretorio1.js
- JS diretorio2.js

node diretorio2.js



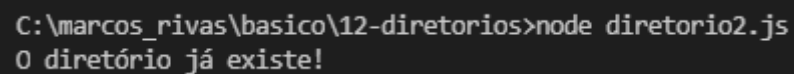
```
C:\marcos_rivas\basico\12-diretorios>node diretorio2.js
O diretório não existe!
Diretório criado com sucesso!
```



12-diretorios

- > css
- JS diretorio1.js
- JS diretorio2.js

node diretorio2.js



```
C:\marcos_rivas\basico\12-diretorios>node diretorio2.js
O diretório já existe!
```

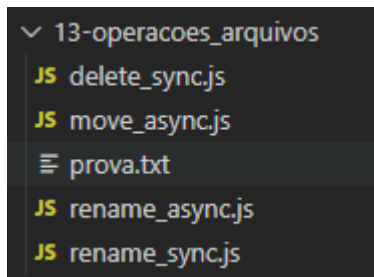
Aula 13 - Operações básicas com arquivos em NodeJS

Renomeando arquivo de forma síncrona

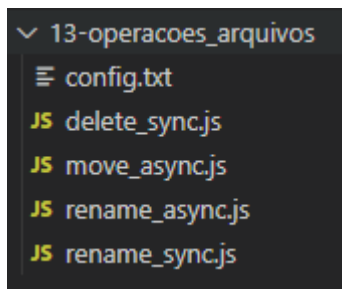
13-operacoes_arquivos\rename_sync.js

```
// renomear arquivo de forma síncrona

const fs = require('fs');
fs.renameSync('./prova.txt', './config.txt');
```



node rename_sync.js



Renomeando arquivo de forma assíncrona

13-operacoes_arquivos\rename_async.js

```
// renomear arquivos de forma assíncrona

const fs = require('fs');

fs.rename('./config.txt', './prova.txt', (err) => {
  if(err){
    throw(err);
  }
  console.log('O arquivo foi renomeado com sucesso!');
})
```

node rename_async.js

```
C:\marcos_rivas\basico\13-operacoes_arquivos>node rename_async.js  
0 arquivo foi renomeado com sucesso!
```

▼ 13-operacoes_arquivos

JS delete_sync.js

JS move_async.js

≡ prova.txt

JS rename_async.js

JS rename_sync.js

Movendo arquivo de forma assíncrona

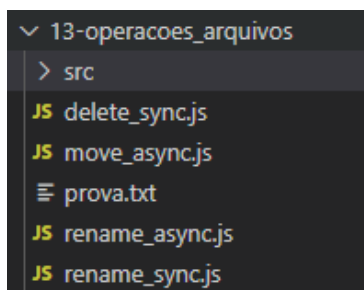
13-operacoes_arquivos\move_async.js

```
// mover arquivos de forma assíncrona

const fs = require('fs');

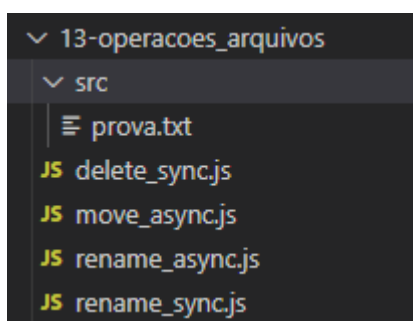
fs.rename('./prova.txt', './src/prova.txt', (err) => {
  if(err){
    throw(err);
  }

  console.log('O arquivo foi movido com sucesso!');
})
```



node move_async.js

```
C:\marcos_rivas\basico\13-operacoes_arquivos>node move_async.js
O arquivo foi movido com sucesso!
```



Excluindo arquivo de forma síncrona

13-operacoes_arquivos\delete_sync.js

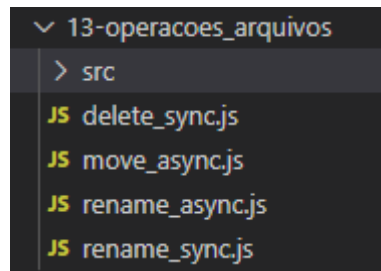
```
// deletando um arquivo de forma síncrona
```

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

```
fs.unlinkSync('./src/prova.txt');  
console.log('O arquivo foi excluído!');
```

```
node delete_sync.js
```

```
C:\marcos_rivas\basico\13-operacoes_arquivos>node delete_sync.js  
O arquivo foi excluído!
```



Aula 14 - Ler streams com NodeJS

14-ler_streams\ler_stream.js

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

// let conteudo = fs.readFileSync('./logs.log');
// console.log(`tamanho: ${conteudo.length}`);

let stream = fs.createReadStream('./logs.log', 'utf-8');

let data = '';

stream.once('data', () => {
  console.log('Iniciando o stream...');
});

stream.on('data', (chunk) => {
  console.log(`${chunk.length} | `);
  data += chunk;
});

stream.on('end', () => {
  console.log('Fim de stream!\n');
  console.log(data.length);
})
```

node ler_stream.js

```
C:\marcos_rivas\basico\14-ler_streams>node ler_stream.js
Iniciando o stream...
65536 |
65536 |
65536 |
65536 |
65536 |
.
.
.
65536 |
65536 |
65536 |
65536 |
65536 |
65536 |
46650 |
Fim de stream!

12105274
```

Aula 15 - Escrever streams com NodeJS

15-escrever_streams\escrever_stream.js

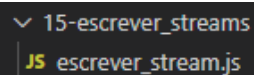
```
const fs = require('fs');
const readline = require('readline');

let i = readline.createInterface(process.stdin, process.stdout);

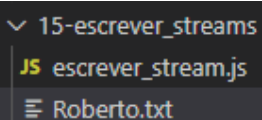
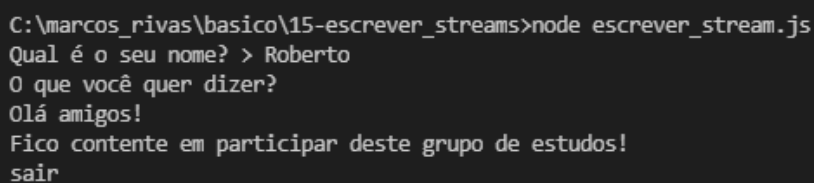
i.question('Qual é o seu nome? > ', (nome)=>{

    let stream = fs.createWriteStream(`./${nome}.txt`);
    stream.write(`Dito isto: ${nome} \n`);
    //fs.writeFileSync(`./${nome}.txt`, `Dito isto ${nome} \n`);
    process.stdout.write('O que você quer dizer? \n');

    i.on('line', (dito)=>{
        if(dito.trim() == 'sair'){
            stream.close();
            i.close();
        }else{
            stream.write(dito.trim() + '\n');
            //fs.appendFileSync(`./${nome}.txt`, dito.trim() + '\n');
        }
    });
});
```



node escrever_stream.js



É criado arquivo de texto com o nome de Roberto.txt e com o seguinte conteúdo:

Roberto.txt

Dito isto: Roberto
Olá amigos!
Fico contente em participar deste grupo de estudos!

Aula 16 - Primeira aplicação completa em NodeJS

16-exercicio01\tutorial\document.js

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

class Document{

  constructor(dir){
    this._content = "";
    this._isSaved = false;
    this._filename = "";
    this._dir = dir;
  }

  exists(name){
    return fs.existsSync(`${this._dir}/${name}`);
  }

  append(text){
    this._content += os.EOL + text;
    this._isSaved = false;
  }

  saveas(name){
    fs.writeFileSync(`${this._dir}/${name}`, this._content);
    this._filename = name;
    this._isSaved = true;
  }

  save(){
    fs.writeFileSync(`${this._dir}/${this._filename}`, this._content);
    this._isSaved = true;
    this._filename = this._filename;
  }

  getContent(){
    return this._content;
  }

  hasName(){
    if(this._filename != ""){
      return true;
    }else{
      return false;
    }
  }

  getName(){
    return this._filename;
  }

  isSaved(){
    return this._isSaved;
  }
}
```

```
open(name){
  this._content = fs.readFileSync(`${this._dir}/${name}`, 'UTF-8');
  this._filename = name;
  this._isSaved = true;
  return this._content;
}

}

module.exports = Document;
```

16-exercicio01\tutorial\messages.js

```
const messages = {
  fileSaved: 'Arquivo gravado, pode seguir editando...',
  fileNotSaved: 'Arquivo não gravado, pode seguir editando...',
  requestFileName: '* Nome do arquivo: ',
  replaceFile: 'Deseja substituir o arquivo?(y/n): ',
  fileNotFound: 'O arquivo não existe!',
  fileExists: '*** Já existe esse arquivo! ***'
};

module.exports = messages;
```

16-exercicio01\tutorial\directory.js

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

class Directory{

  constructor(){
    this._path = __dirname;
    this.createDocsDir();
  }

  createDocsDir(){
    this._path = path.join(this._path, 'docs', '');
    if(!fs.existsSync('./docs')){
      fs.mkdirSync('./docs');
    }
  }

  getPath(){
    return this._path;
  }

  getShortPath(){
    const paths = path.parse(this._path);
    let delimiter = "/";

    if(paths.dir.indexOf(delimiter) < 0){
      delimiter = '\\';
    }
    //const dirs = paths.dir.split(delimiter);
    return `${paths.root}...${delimiter}${paths.name}`;
  }

  getFilesInDir(){
    const files = fs.readdirSync(this._path);
    let n = 0;
    console.log(`
=====
Localização: ${this.getShortPath()}
=====`);
    files.forEach(file => {
      if(file !== '.DS_Store'){
        console.log(` ${file}`);
        n++;
      }
    });
  }
}

module.exports = Directory;
```

16-exercicio01\tutorial\app.js

```
const readline = require('readline');
const Messages = require('./messages');
const Document = require('./document');
const Directory = require('./directory');

const dir = new Directory();

let interface = readline.createInterface(process.stdin, process.stdout);

const tools = `Comandos: :q = sair, :sa = gravar como, :s = salvar
-----`

const pantalla = `
=====
Editor de texto
=====
Escolha uma opção:
1 Criar novo documento
2 Abrir documento
3 Fechar editor
> `;

mainScreen();
function mainScreen(){
  process.stdout.write('\033c');
  interface.question(pantalla, (res) =>{
    switch(res.trim()){
      case '1':
        createFile();
        break;

      case '2':
        openFileInterface();
        break;

      case '3':
        interface.close();
        break;

      default:
        mainScreen();
    }
  });
}

function readCommands(file){
  interface.on('line', (input) =>{
    switch(input.trim()){
      case ':sa':
        saveAs(file);
        break;
    }
  });
}
```



```

        case 'q':
            interface.removeAllListeners('line');
            mainScreen();
            break;

        case 's':
            save(file);
            break;

        default:
            file.append(input.trim());
    }
})
}

function createFile(){
    let file = new Document(dir.getPath());

    renderInterface(file);
    readCommands(file);
}

function save(file){
    if(file.hasName()){
        file.save()
        renderInterface(file, `${Messages.fileSaved}\n`);
    }else{
        saveAs(file);
    }
}

function saveAs(file){
    interface.question(Messages.requestFileName, (name) =>{
        if(file.exists(name)){
            console.log(Messages.fileExists);
            interface.question(Messages.replaceFile, (confirm) =>{
                if(confirm == 'y'){
                    file.saveas(name);
                    renderInterface(file, `${Messages.fileSaved}\n`);
                }else{
                    renderInterface(file, `${Messages.fileNotSaved}\n`);
                }
            });
        }else{
            file.saveas(name);
            renderInterface(file, `${Messages.fileSaved}\n`);
        }
    });
}

function openFile(file, name){
    content = file.open(name);

```

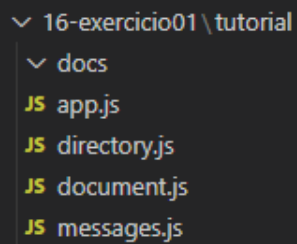
```

    renderInterface(file);
    readCommands(file);
}

function renderInterface(file, mensaje){
    process.stdout.write('\033c');
    (file.getName() == '') ? console.log(` | Untitled | `) : console.log(` | ${file.getName()} | `);
    console.log(tools);
    if(mensaje != null) console.log(mensaje);
    console.log(file.getContent());
}

function openFileInterface(){
    let file = new Document(dir.getPath());
    dir.GetFilesInDir();
    interface.question(Messages.requestFileName, (name) =>{
        if(file.exists(name)){
            openFile(file, name);
        }else{
            console.log(Messages.fileNotFound);
            interface.removeAllListeners('line');
            mainScreen();
        }
    });
}

```



```

v 16-exercicio01 \tutorial
  v docs
  JS app.js
  JS directory.js
  JS document.js
  JS messages.js

```

node app.js

```
=====
Editor de texto
=====

Escolha uma opção:

1 Criar novo documento
2 Abrir documento
3 Fechar editor

> 1
```

```
| Untitled |
Comandos: :q = sair, :sa = gravar como, :s = salvar
-----

> Olá amigos!
Meu nome é Roberto Pinheiro.
> Sejam todos bem-vindos a este grupo de estudos!
Qualquer dúvida, informação ou comentário, por favor, poste aqui neste aplicativo.
> :sa
```

```
| Untitled |
Comandos: :q = sair, :sa = gravar como, :s = salvar
-----

> Olá amigos!
Meu nome é Roberto Pinheiro.
> Sejam todos bem-vindos a este grupo de estudos!
Qualquer dúvida, informação ou comentário, por favor, poste aqui neste aplicativo.
> :sa
* Nome do arquivo: welcome.txt
```

```
✓ 16-exercicio01\tutorial
  ✓ docs
    ≡ welcome.txt
  JS app.js
  JS directory.js
  JS document.js
  JS messages.js
```

```
| welcome.txt |
Comandos: :q = sair, :sa = gravar como, :s = salvar
-----
Arquivo gravado, pode seguir editando...

Olá amigos!
Meu nome é Roberto Pinheiro.
Sejam todos bem-vindos a este grupo de estudos!
Qualquer dúvida, informação ou comentário, por favor, poste aqui neste aplicativo.
:q
```

```
=====
Editor de texto
=====

Escolha uma opção:

1 Criar novo documento
2 Abrir documento
3 Fechar editor

> |
```

```
=====
Editor de texto
=====

Escolha uma opção:

1 Criar novo documento
2 Abrir documento
3 Fechar editor

> 2|
```

```
=====
Localização: C:\...\docs
=====

welcome.txt
* Nome do arquivo: welcome.txt|
```

```
| welcome.txt |
Comandos: :q = sair, :sa = gravar como, :s = salvar
-----

Olá amigos!
Meu nome é Roberto Pinheiro.
Sejam todos bem-vindos a este grupo de estudos!
Qualquer dúvida, informação ou comentário, por favor, poste aqui neste aplicativo.
Conheçam o novo curso de Node.js e participem!
:s|
```

```
| welcome.txt |
Comandos: :q = sair, :sa = gravar como, :s = salvar
-----

Arquivo gravado, pode seguir editando...

Olá amigos!
Meu nome é Roberto Pinheiro.
Sejam todos bem-vindos a este grupo de estudos!
Qualquer dúvida, informação ou comentário, por favor, poste aqui neste aplicativo.
Conheçam o novo curso de Node.js e participem!
:q|
```

```
=====
Editor de texto
=====

Escolha uma opção:

1 Criar novo documento
2 Abrir documento
3 Fechar editor

> 3|
```

Aula 17 - Fazer requisições HTTP com NodeJS

17-requisicao_http\requisicao.js

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

const options = {
  hostname: 'localhost',
  port: 80,
  path: '/curso/49.%20carrito/tutorial/api/productos/api-productos.php?categoria=juguetes',
  method: 'GET'
};

const req = http.request(options, res => {
  console.log(`status code: ${res.statusCode}`);
  console.log('Headers: %j', res.headers);

  let body = '';
  res.on('data', chunk => {
    body += chunk;
  });

  res.on('end', () => {
    console.log("\n\nResultados:")
    console.log(body);
  });
});

req.on('error', err => {});
req.end();
```

Aula 18 - Criar um servidor web em 6 linhas de código

18-servidor_web\server.js

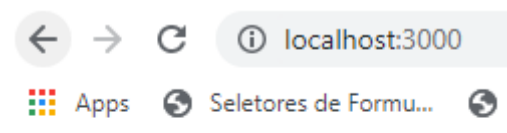
```
const http = require('http');
http.createServer((req, res) =>{
  res.writeHead(200, {'content-type': 'text/html'});
  res.end(`
    <!DOCTYPE html>
    <html lang="pt-br">
    <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <meta http-equiv="X-UA-Compatible" content="ie=edge">
      <title>Documento HTML</title>
    </head>
    <body>
      <h1>Olá Mundo!</h1>
      <p>
        Bem-vindo a meu servidor!
      </p>
    </body>
    </html>
  `);
}).listen(3000);
console.log('Servidor iniciado...');
```

node server.js

```
C:\marcos_rivas\basico\18-servidor_web>node server.js
Servidor iniciado...
█
```

- No browser:

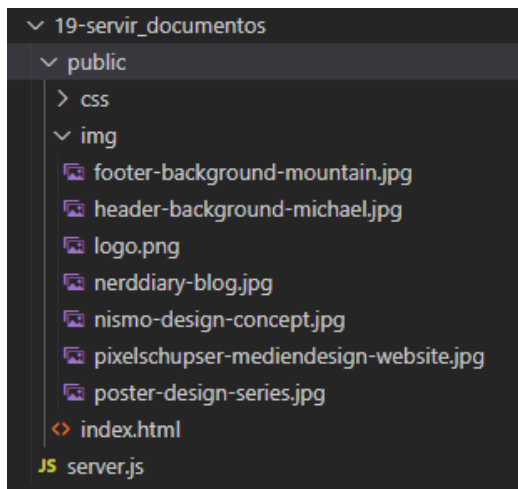
localhost:3000



Olá Mundo!

Bem-vindo a meu servidor!

Aula 19 - Carregar arquivos de um website



19-servir_documentos\server.js

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

http.createServer((req, res) =>{

  console.log(`${req.method} solicita ${req.url}`);

  if(req.url == '/'){
    fs.readFile('./public/index.html', 'UTF-8', (err, html) =>{
      res.writeHead(200, {'Content-Type': 'text/html'});
      res.end(html);
    });
  }else if(req.url.match(/.css$/)){
    const reqPath = path.join(__dirname, 'public', req.url);
    const fileStream = fs.createReadStream(reqPath, 'UTF-8');

    res.writeHead(200, {'Content-Type': 'text/css'});
    fileStream.pipe(res);
  }else if(req.url.match(/.jpg$/)){
    const reqPath = path.join(__dirname, 'public', req.url);
    const fileStream = fs.createReadStream(reqPath);

    res.writeHead(200, {'Content-Type': 'image/jpeg'});
    fileStream.pipe(res);
  }else{
    res.writeHead(404, {'Content-Type': 'text/plain'});
    res.end('404 ERROR');
  }

}).listen(3000);

console.log('Servidor iniciado...');
```

19-servir_documentos\public\index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Mi primer sitio web</title>
  <link rel="stylesheet" href="css/main.css">
</head>
<body>
  <div id="main-container">
    <section id="intro-container">
      <div class="center-align">
        <nav>
          <div id="logo"></div>
          <div id="menu-lista">
            <ul>
              <li><a href="#about-section">Acerca de..</a></li>
              <li><a href="#work">Proyectos</a></li>
              <li><a href="#footer">Contacto</a></li>
            </ul>
          </div>
        </nav>

        <div class="section-columns">
          <div>
            <h2 class="white">Mi primer sitio web en la vida</h2>
            <div class="btn-section">
              <a href="#about-section" class="btn btn-white">Conóceme más</a>
            </div>
          </div>

          <div id="image-bg">
          </div>
        </div>
      </div>
    </section>

    <div class="rectangle bg-white skew-left"></div>

    <div class="section" id="about-section">
      <div class="center-align">
        <div class="section-columns">
          <div>
            <h2>Diseño.<br />Desarrollo.<br/>Inspiración.</h2>
          </div>

          <div id="bio">
            <h3>Mi nombre es <span>Marcos Rivas</span> , soy un ingeniero en Sistemas
            Computacionales, <span>trabajo</span> <span>en</span> <span>Microsoft</span>, y me dedico al
            <span>diseño web</span><span> como</span><span> hobbie.</span></h3>

            <p>Tengo mi <span>propio canal de Youtube</span> que trata sobre tutoriales y
            cursos para aprender <span>diseño y programación.</span></p>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```


<p>Gasto mucho de mi tiempo en dibujar, que es otro de mis pasatiempos favoritos, así como seguir aprendiendo de las tecnologías actuales.</p>

<p>Puedes seguirme en Twitter o Facebook, o directamente en el canal de Youtube</p>

```
<div class="btn-section">
  <a href="#work" class="btn btn-black">Ver proyectos sobresalientes</a>
</div>
</div>
</div>
</div>
</div>
```

```
<div class="rectangle bg-gray skew-right"></div>
```

```
<div class="section bg-gray" id="work">
  <div class="center-align">
    <div class="section-columns" >
      <div>
        <h2 class="white">Posters.<br />Sitios Web.<br />Tutoriales.</h2>
        <div class="btn-section">
          <a href="" class="btn btn-white">Ver proyectos sobresalientes</a>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
<div class="section bg-gray" id="work02">
  <div class="center-align">
    <div class="section-columns" >
      <div>
        <h2 class="white">Opinión.<br />Impactante.<br />Blog.</h2>
        <div class="btn-section">
          <a href="" class="btn btn-white">Ver Proyecto</a>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
<div class="section bg-gray" id="work03">
  <div class="center-align">
    <div class="section-columns" >
      <div>
        <h2 class="white">Patrón<br />Diseño.<br />Móvil.</h2>
        <div class="btn-section">
          <a href="" class="btn btn-white">Ver Proyecto</a>
        </div>
      </div>
    </div>
  </div>
</div>
```

```

<div class="section bg-gray" id="work04">
  <div class="center-align">
    <div class="section-columns" >
      <div>
        <h2 class="white">Innovación.<br />Fresco.<br/>Color.</h2>
        <div class="btn-section">
          <a href="" class="btn btn-white">Ver Proyecto</a>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="rectangle bg-black skew-left"></div>
<div class="section bg-black" id="footer">
  <div class="center-align">
    <div class="section-columns" >
      <div>
        <h2 class="white">Sigamos en Contacto.</h2>
        <div class="btn-section">
          <a href="mailto: contacto@misitio.com" class="btn btn-white">Mándame un
mensaje</a>
        </div>
      </div>
      <div>
        <h3>Contáctame si quieres conocerme y saber cómo podemos colaborar en tu
siguiente proyecto.</h3>
        <p>También puedes estar conmigo por medio de las redes sociales, enterarte de mi
más reciente trabajo y conocerme un poco más.</p>
        <div class="btn-section">
          <a href="http://www.facebook.com/misitio/" class="btn btn-
white">Facebook</a>
          <a href="http://www.twitter.com/misitio/" class="btn btn-white">Twitter</a>
          <a href="http://www.instagram.com/misitio/" class="btn btn-
white">Instagram</a>
        </div>
      </div>
    </div>
  </div>
</div>
</body>
</html>

```

19-servir_documentos\public\css

```
body{
  padding: 0;
  margin: 0;
  font-family: Arial;
}
span{
  position: relative;
}
span::after{ /* #ffcd6 */
  background-color: #ffcd6;
  content: "";
  width: 100%;
  height: 50%;
  position: absolute;
  bottom: -4px;
  left: 0;
  z-index: -1;
}
p a{ /* #184DA8 */
  color: #184DA8;
  text-decoration: none;
}
p a:hover{ /* #184DA8 */
  text-decoration: underline;
}
.center-align{
  margin: 0 auto;
  width: 1100px;
}
.black{color: black;}
.white{color: white;}
.section-columns{
}
.section-columns > div{
  display: inline-block;
  vertical-align: top;
  width: 49%;
}
.section-columns h2{
  font-size: 72px;
}
#intro-container .section-columns h2{
  padding-top: 120px;
  padding-bottom: 30px;
}
.section-columns h3{
  font-size: 26px;
  font-weight: lighter;
  line-height: 40px;
}
.section-columns p{
  font-weight: lighter;
  font-size: 16px;
  line-height: 30px;
  padding-bottom: 10px;
}
```

```

}
.section-columns #image-bg{
    background-image: url('../img/header-background-michael.jpg');
    background-repeat: no-repeat;
    background-size: 750px;
    background-position: -140px -80px;
    height: 600px;
}
.section-columns #image-bg-02{
}
.section-columns #bio{
    height: 600px;
}
#work, #work02, #work03, #work04{
    background-repeat: no-repeat;
    background-size: 750px;
    background-position: right center;
    height: 600px;
    padding-top: 150px;
    margin-top: -50px;
}
#work {background-image: url('../img/poster-design-series.jpg');}
#work02{background-image: url('../img/nerddiary-blog.jpg');}
#work03{background-image: url('../img/nismo-design-concept.jpg');}
#work04{background-image: url('../img/pixelschupser-mediendesign-website.jpg');}
#footer{
    margin-top: -50px;
    padding-top: 50px;
    height: 550px;
    background-image: url('../img/footer-background-mountain.jpg');
    background-size: 100%;
    background-position: center bottom;
    background-repeat: no-repeat;
}
#footer h3{color: white; padding-top: 70px; margin: 0;}
#footer p {color: gray; padding-bottom: 30px;}
.rectangle{
    height: 100px;
    margin-top: -50px;
}
.bg-white{
    background-color: white;
}
.bg-gray{
    background-color: #666;
}
.bg-black{
    background-color: #000;
}
.skew-left{
    transform: skewY(-4deg);
}
.skew-right{
    transform: skewY(4deg);
}

```

```
.btn{
    border: solid 1px;
    padding: 20px 40px;
    text-decoration: none;
    margin-right: 10px;
}
.btn-black{
    border-color: black;
    color: black;
}
.btn-black:hover{
    background-color: black;
    border-color: white;
    color: white;
}
.btn-white{
    border-color: white;
    color: white;
}
.btn-white:hover{
    background-color: white;
    border-color: black;
    color: black;
}
#intro-container{
    background-color: black;
    padding-top: 100px;
}
#intro-container nav{
    margin: 0;
    padding: 0;
}
#intro-container nav div{
    display: inline-block;
    vertical-align: top;
}
#intro-container nav #logo{
    width: 9%;
}
#intro-container nav #menu-lista{
    width: 89%;
    text-align: right;
}
#intro-container nav #menu-lista ul{
    margin: 0;
    padding: 0;
}
#intro-container nav #menu-lista ul li{
    display: inline-block;
    padding: 0;
    margin: 0;
    list-style: none;
}
#intro-container nav #menu-lista ul li a{
    color: #777;
    text-decoration: none;
}
```

```
#intro-container nav #menu-lista ul li a: hover{  
  color: white;  
  text-decoration: white;  
}
```

node server.js

```
C:\marcos_rivas\basico\19-servir_documentos>node server.js  
Servidor iniciado...
```

localhost:3000

```
C:\marcos_rivas\basico\19-servir_documentos>node server.js  
Servidor iniciado...  
GET solicita /  
GET solicita /css/main.css  
GET solicita /img/logo.png  
GET solicita /img/header-background-michael.jpg  
GET solicita /img/poster-design-series.jpg  
GET solicita /img/nerddiary-blog.jpg  
GET solicita /img/nismo-design-concept.jpg  
GET solicita /img/pixelschupser-mediendesign-website.jpg  
GET solicita /img/footer-background-mountain.jpg
```

- No browser:

Mi primer sitio web en la vida

Ver portfolio



Diseño. Desarrollo. Inspiración.

Me llamo Marco Flores, soy un Ingeniero en Sistemas Computacionales, trabajo en Microsoft, y me dedico al diseño web como hobby.

Me gusta jugar video de Football con mis amigos, también me gusta leer libros y jugar video de Fútbol.

Quiero estar siempre en el mundo del diseño web y la tecnología.

Quiero estar siempre en el mundo del diseño web y la tecnología.

Ver portfolio

Posters. Sitios Web. Tutoriales.

Ver portfolio



Opinión. Impactante. Blog.

Ver portfolio



Patrón Diseño. Móvil.

Ver portfolio



Innovación. Fresco. Color.

Ver portfolio



Sigamos en Contacto.

Enviar mensaje

Contáctame si quieres conocerme y saber cómo podemos colaborar en tu siguiente proyecto.

Quiero estar siempre en el mundo del diseño web y la tecnología.

Ver portfolio

Ver portfolio

Ver portfolio

Ver portfolio

Aula 20 - Criar uma API estática em NodeJS

20-api_estatica\server.js

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

const pedidos = [
  { 'nome': 'sopa', 'quantidade': 2, 'status': 'servido', 'mesa': 5},
  { 'nome': 'corte de carne', 'quantidade': 2, 'status': 'processo', 'mesa': 4},
  { 'nome': 'sopa', 'quantidade': 1, 'status': 'pago', 'mesa': 3},
  { 'nome': 'salada', 'quantidade': 1, 'status': 'pago', 'mesa': 5},
  { 'nome': 'massa', 'quantidade': 4, 'status': 'em processo', 'mesa': 1},
];

http.createServer((req, res) =>{
  if(req.url == '/'){
    res.writeHead(200, {'Content-Type': 'text/json'});
    res.end(JSON.stringify(pedidos));
  }else if(req.url == '/pedidos-processo'){
    pedidosEmProcesso(res);
  }else if(req.url == '/pedidos-sopa'){
    pedidosPorNome('sopa', res);
  }else{
    res.writeHead(404, {'Content-Type': 'text/plain'});
    res.end('Não encontrada a requisição requerida','latin1');
  }
}).listen(3000);

console.log('servidor iniciado...');

const pedidosEmProcesso = (res) =>{
  const arrayRes = pedidos.filter(item =>{
    return item.status == 'em processo';
  });

  res.writeHead(200, {'Content-Type': 'text/json'});
  res.end(JSON.stringify(arrayRes));
}

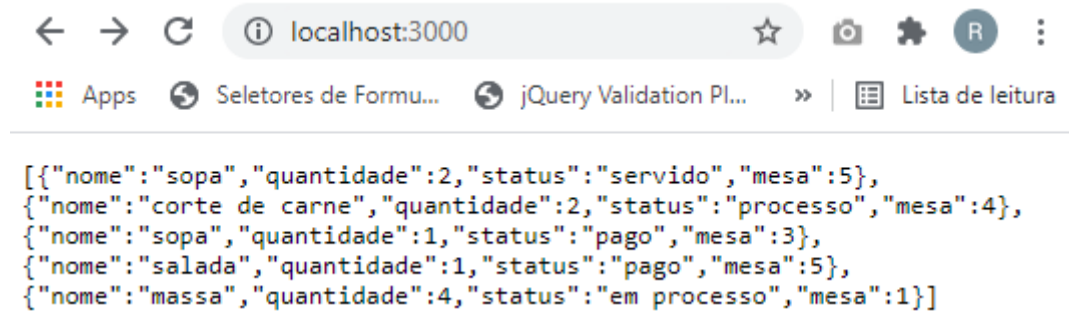
const pedidosPorNome = (nome, res) =>{
  const arrayRes = pedidos.filter(item =>{
    return item.nome == nome;
  });

  res.writeHead(200, {'Content-Type': 'text/json'});
  res.end(JSON.stringify(arrayRes));
}
```

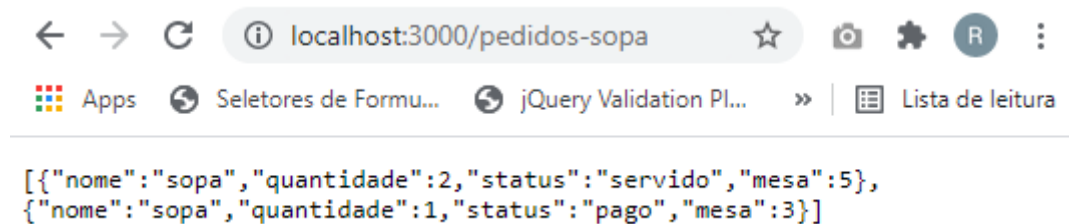

node server.js

```
C:\marcos_rivas\basico\20-api_estatica>node server.js
servidor iniciado...
[]
```

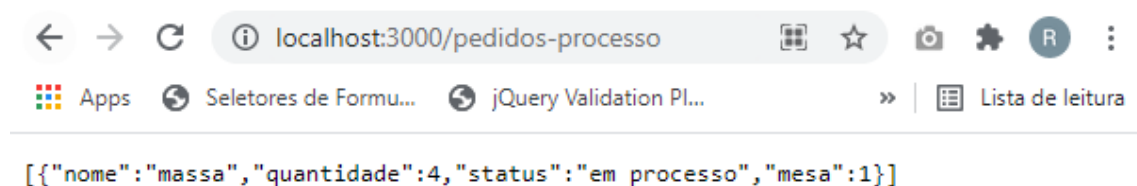
<http://localhost:3000>



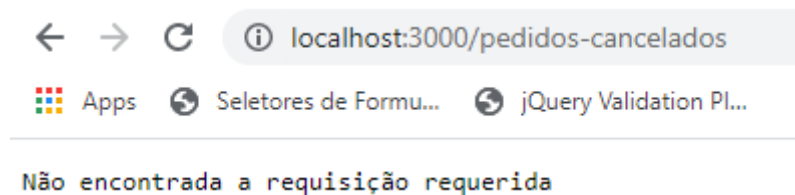
<http://localhost:3000/pedidos-sopa>



<http://localhost:3000/pedidos-processo>



<http://localhost:3000/pedidos-cancelados>



Aula 21 - Manipulação de requisições POST

21-requisicao_post\server.js

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

http.createServer((req, res) =>{

  if(req.method == 'GET'){
    res.writeHead(200, {'Content-Type': 'text/html'});
    fs.createReadStream('./form.html', 'UTF-8').pipe(res);
  }else if(req.method == 'POST'){

    let body = "";

    req.on('data', chunk =>{body+= chunk;});

    req.on('end', () =>{
      res.writeHead(200, {'Content-Type': 'text/html'});
      res.end(`
        <!DOCTYPE html>
        <html lang="pt-br">
        <head>
          <meta charset="UTF-8">
          <meta name="viewport" content="width=device-width, initial-scale=1.0">
          <meta http-equiv="X-UA-Compatible" content="ie=edge">
          <title>Resultados</title>
        </head>
        <body>
          <h1>Dados recebidos do formulário</h1>
          <p>${body}</p>
        </body>
        </html>
      `);
    });
  }

}).listen(3000);

console.log('Servidor iniciado...');
```

21-requisicao_post\form.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Formulario</title>
</head>
<body>
  <form action="/" method="POST">
    <h1>Formulário de contato</h1>
    <p>
      Nome: <br/>
      <input type="text" name="nome">
    </p>

    <p>
      Email: <br/>
      <input type="text" name="email">
    </p>

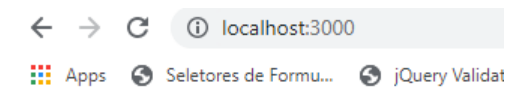
    <p>
      País de origem: <br/>
      <input type="text" name="pais">
    </p>
    <p>
      <input type="submit" value="Enviar formulário">
    </p>
  </form>
</body>
</html>
```

node server.js

```
C:\marcos_rivas\basico\21-requisicao_post>node server.js
Servidor iniciado...
█
```

- No browser:

localhost:3000

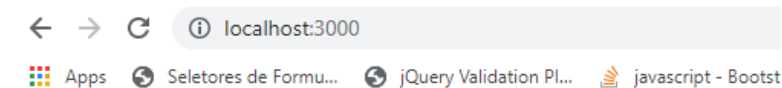


Formulário de contato

Nome:

Email:

País de origem:



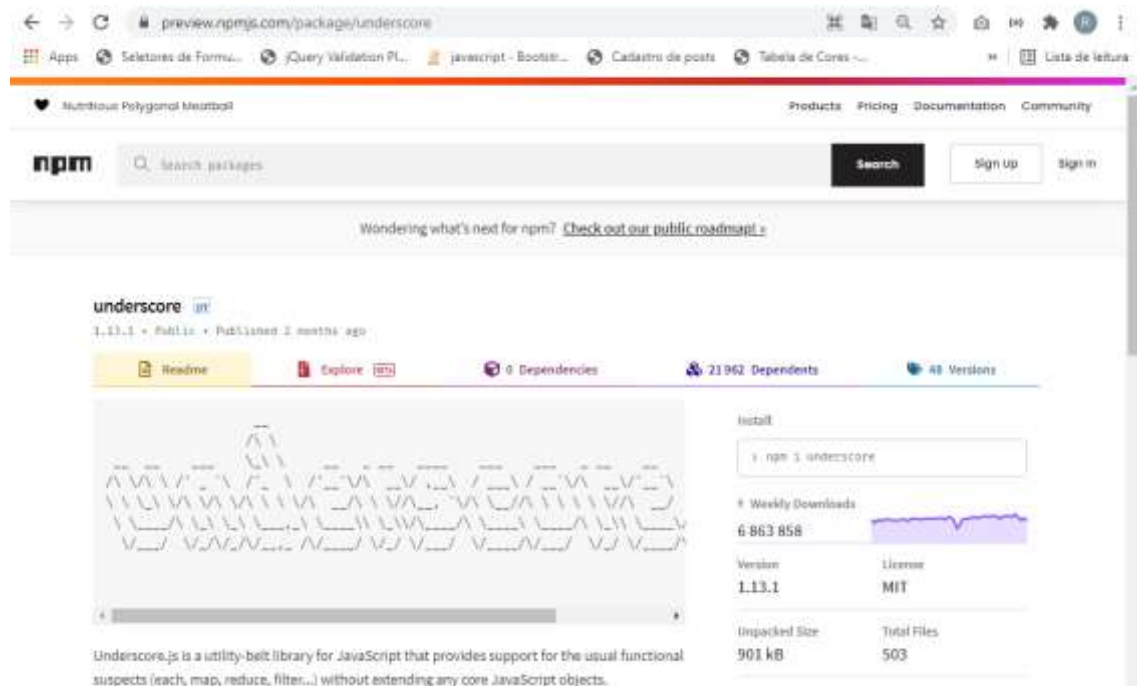
Dados recebidos do formulário

nome=Roberto+Pinheiro&email=beto10561%40gmail.com&pais=Brasil

Aula 22 - Tutorial básico de NPM

Instalando o pacote underscore

<https://preview.npmjs.com/package/underscore>



`npm install underscore`

```
C:\marcos_rivas\basico\22-npm>npm install underscore
npm WARN saveError ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\22-npm\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\22-npm\package.json'
npm WARN 22-npm No description
npm WARN 22-npm No repository field.
npm WARN 22-npm No README data
npm WARN 22-npm No license field.

+ underscore@1.13.1
added 1 package from 1 contributor and audited 1 package in 4.74s
found 0 vulnerabilities
```

```
▼ 22-npm
  > node_modules\underscore
  JS app.js
  {} package-lock.json
```

22-npm\app.js

```
const _ = require('underscore');

const lista = [
  {'id': 1, 'nome': 'Marcos', 'idade': 27},
  {'id': 2, 'nome': 'Regina', 'idade': 26},
  {'id': 3, 'nome': 'Bruno', 'idade': 22}
];

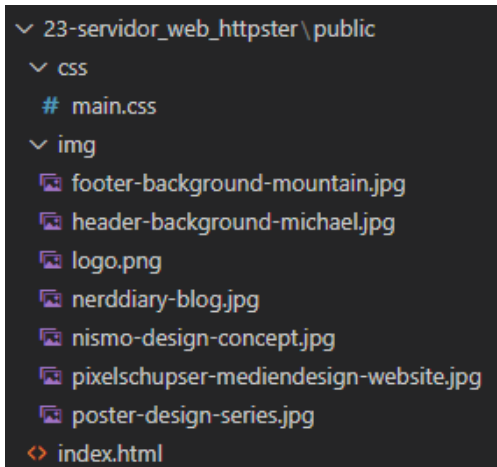
const res = _.findWhere(lista, {'idade': 22});

console.log(res);
```

node app.js

```
C:\marcos_rivas\basico\22-npm>node app.js
{ id: 3, nome: 'Bruno', idade: 22 }
```

Aula 23 - Servidor web básico con httpster



23-servidor_web_httpster\public\index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Mi primer sitio web</title>
  <link rel="stylesheet" href="css/main.css">
</head>
<body>
  <div id="main-container">
    <section id="intro-container">
      <div class="center-align">
        <nav>
          <div id="logo"></div>
          <div id="menu-lista">
            <ul>
              <li><a href="#about-section">Acerca de..</a></li>
              <li><a href="#work">Proyectos</a></li>
              <li><a href="#footer">Contacto</a></li>
            </ul>
          </div>
        </nav>

        <div class="section-columns">
          <div>
            <h2 class="white">Mi primer sitio web en la vida</h2>
            <div class="btn-section">
              <a href="#about-section" class="btn btn-white">Conóceme más</a>
            </div>
          </div>
          <div id="image-bg">
          </div>
        </div>
      </div>
    </section>
  </div>
</body>
</html>
```

```
<div class="rectangle bg-white skew-left"></div>
```

```
<div class="section" id="about-section">
```

```
<div class="center-align">
```

```
<div class="section-columns">
```

```
<div>
```

```
<h2>Diseño.<br />Desarrollo.<br/>Inspiración.</h2>
```

```
</div>
```

```
<div id="bio">
```

```
<h3>Mi nombre es <span>Marcos Rivas</span> , soy un ingeniero en Sistemas  
Computacionales, <span>trabajo</span> <span>en</span> <span>Microsoft</span>, y me dedico al  
<span>diseño web</span><span> como</span><span> hobbie.</span></h3>
```

```
<p>Tengo mi <span>propio canal de Youtube</span> que trata sobre tutoriales y  
cursos para aprender <span>diseño y programación.</span></p>
```

```
<p><span>Gasto mucho de mi tiempo en dibujar</span>, que es otro de mis  
pasatiempos favoritos, así como seguir aprendiendo de las <span>tecnologías actuales.</span></p>
```

```
<p><span>Puedes seguirme</span> en <a href="#">Twitter</a> o <a  
href="#">Facebook</a>, o directamente en el <a href="#">canal de Youtube</a></p>
```

```
<div class="btn-section">
```

```
<a href="#work" class="btn btn-black">Ver proyectos sobresalientes</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="rectangle bg-gray skew-right"></div>
```

```
<div class="section bg-gray" id="work">
```

```
<div class="center-align">
```

```
<div class="section-columns" >
```

```
<div>
```

```
<h2 class="white">Posters.<br />Sitios Web.<br/>Tutoriales.</h2>
```

```
<div class="btn-section">
```

```
<a href="" class="btn btn-white">Ver proyectos sobresalientes</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="section bg-gray" id="work02">
```

```
<div class="center-align">
```

```
<div class="section-columns" >
```

```
<div>
```

```
<h2 class="white">Opinión.<br />Impactante.<br/>Blog.</h2>
```

```
<div class="btn-section">
```

```
<a href="" class="btn btn-white">Ver Proyecto</a>
```

```
</div>
```

```
</div>
```

```
</div>
```



```

    </div>
</div>

<div class="section bg-gray" id="work03">
  <div class="center-align">
    <div class="section-columns" >
      <div>
        <h2 class="white">Patrón<br />Diseño.<br/>Móvil.</h2>
        <div class="btn-section">
          <a href="" class="btn btn-white">Ver Proyecto</a>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="section bg-gray" id="work04">
  <div class="center-align">
    <div class="section-columns" >
      <div>
        <h2 class="white">Innovación.<br />Fresco.<br/>Color.</h2>
        <div class="btn-section">
          <a href="" class="btn btn-white">Ver Proyecto</a>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="rectangle bg-black skew-left"></div>
<div class="section bg-black" id="footer">
  <div class="center-align">
    <div class="section-columns" >
      <div>
        <h2 class="white">Sigamos en Contacto.</h2>
        <div class="btn-section">
          <a href="mailto: contacto@misitio.com" class="btn btn-white">Mándame un
mensaje</a>
        </div>
      </div>
    </div>

    <div>
      <h3>Contáctame si quieres conocerme y saber cómo podemos colaborar en tu
siguiente proyecto.</h3>
      <p>También puedes estar conmigo por medio de las redes sociales, enterarte de mi
más reciente trabajo y conocerme un poco más.</p>

      <div class="btn-section">
        <a href="http://www.facebook.com/misitio/" class="btn btn-
white">Facebook</a>
        <a href="http://www.twitter.com/misitio/" class="btn btn-white">Twitter</a>
        <a href="http://www.instagram.com/misitio/" class="btn btn-
white">Instagram</a>
      </div>
    </div>
  </div>
</div>

```

```
        </div>

    </div>
</body>
</html>
```

23-servidor_web_httpster\public\css\main.css

```
body{
    padding: 0;
    margin: 0;
    font-family: Arial;
}
span{
    position: relative;
}
span::after{ /* #ffcd6 */
    background-color: #ffcd6;
    content: "";
    width: 100%;
    height: 50%;
    position: absolute;
    bottom: -4px;
    left: 0;
    z-index: -1;
}
p a{ /* #184DA8 */
    color: #184DA8;
    text-decoration: none;
}
p a:hover{ /* #184DA8 */
    text-decoration: underline;
}
.center-align{
    margin: 0 auto;
    width: 1100px;
}
.black{color: black;}
.white{color: white;}
.section-columns{
}
.section-columns > div{
    display: inline-block;
    vertical-align: top;
    width: 49%;
}
.section-columns h2{
    font-size: 72px;
}
#intro-container .section-columns h2{
    padding-top: 120px;
    padding-bottom: 30px;
}
.section-columns h3{
    font-size: 26px;
```

```

    font-weight: lighter;
    line-height: 40px;
}
.section-columns p{
    font-weight: lighter;
    font-size: 16px;
    line-height: 30px;
    padding-bottom: 10px;
}
.section-columns #image-bg{
    background-image: url('../img/header-background-michael.jpg');
    background-repeat: no-repeat;
    background-size: 750px;
    background-position: -140px -80px;
    height: 600px;
}
.section-columns #image-bg-02{
}
.section-columns #bio{
    height: 600px;
}
#work, #work02, #work03, #work04{
    background-repeat: no-repeat;
    background-size: 750px;
    background-position: right center;
    height: 600px;
    padding-top: 150px;
    margin-top: -50px;
}
#work {background-image: url('../img/poster-design-series.jpg');}
#work02{background-image: url('../img/nerddiary-blog.jpg');}
#work03{background-image: url('../img/nismo-design-concept.jpg');}
#work04{background-image: url('../img/pixelschupser-mediendesign-website.jpg');}
#footer{
    margin-top: -50px;
    padding-top: 50px;
    height: 550px;
    background-image: url('../img/footer-background-mountain.jpg');
    background-size: 100%;
    background-position: center bottom;
    background-repeat: no-repeat;
}
#footer h3{color: white; padding-top: 70px; margin: 0;}
#footer p {color: gray; padding-bottom: 30px;}
.rectangle{
    height: 100px;
    margin-top: -50px;
}
.bg-white{
    background-color: white;
}
.bg-gray{
    background-color: #666;
}
.bg-black{
    background-color: #000;
}

```

```
.skew-left{
    transform: skewY(-4deg);
}
.skew-right{
    transform: skewY(4deg);
}
.btn{
    border: solid 1px;
    padding: 20px 40px;
    text-decoration: none;
    margin-right: 10px;
}
.btn-black{
    border-color: black;
    color: black;
}
.btn-black:hover{
    background-color: black;
    border-color: white;
    color: white;
}
.btn-white{
    border-color: white;
    color: white;
}
.btn-white:hover{
    background-color: white;
    border-color: black;
    color: black;
}
#intro-container{
    background-color: black;
    padding-top: 100px;
}
#intro-container nav{
    margin: 0;
    padding: 0;
}
#intro-container nav div{
    display: inline-block;
    vertical-align: top;
}
#intro-container nav #logo{
    width: 9%;
}
#intro-container nav #menu-lista{
    width: 89%;
    text-align: right;
}
#intro-container nav #menu-lista ul{
    margin: 0;
    padding: 0;
}
#intro-container nav #menu-lista ul li{
    display: inline-block;
    padding: 0;
    margin: 0;
```

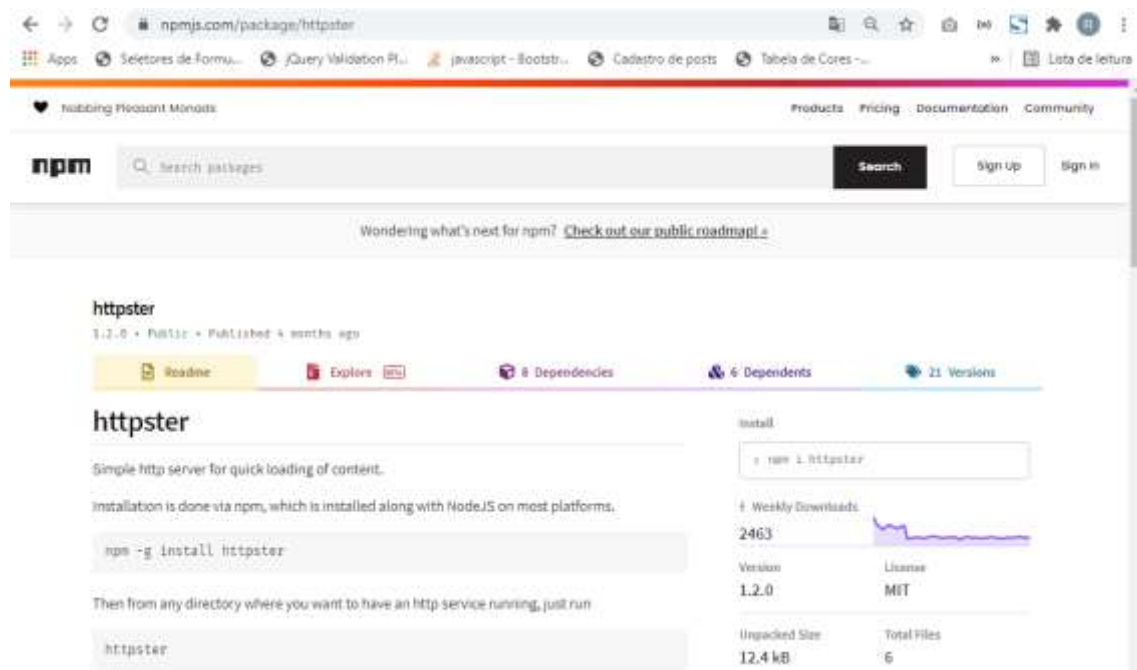
```

    list-style: none;
}
#intro-container nav #menu-lista ul li a{
    color: #777;
    text-decoration: none;
}
#intro-container nav #menu-lista ul li a:hover{
    color: white;
    text-decoration: white;
}

```

Instalando o pacote httpster

<https://www.npmjs.com/package/httpster>



`npm -g install httpster`

```

C:\marcos_rivas\basico\23-servidor_web_httpster>npm -g install httpster
C:\Users\betoi\AppData\Roaming\npm\httpster -> C:\Users\betoi\AppData\Roaming\npm\node_modules\httpster\bin\httpster
+ httpster@1.2.0
added 61 packages from 40 contributors in 12.706s

```

`httpster -p 8081 -d ./public`

```

C:\marcos_rivas\basico\23-servidor_web_httpster>httpster -p 8081 -d ./public
Starting HTTPster v1.2.0 on port "8081" from ./public

```

- No browser:

localhost:8081



Aula 24 - Reinício automático de servidor

24-reinicio\index.js

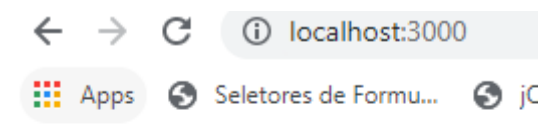
```
const http = require('http');

http.createServer((req, res) =>{
  res.writeHead(200, {'content-type': 'text/html'});

  res.end(`
    <!DOCTYPE html>
    <html lang="pt-br">
    <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <meta http-equiv="X-UA-Compatible" content="ie=edge">
      <title>Document</title>
    </head>
    <body>
      <h1>Website de tutoriais</h1>
      <h3>Bem-vindo ao meu site!</h3>
    </body>
    </html>
  `);
}).listen(3000);

console.log('Servidor iniciado...');
```

node index.js

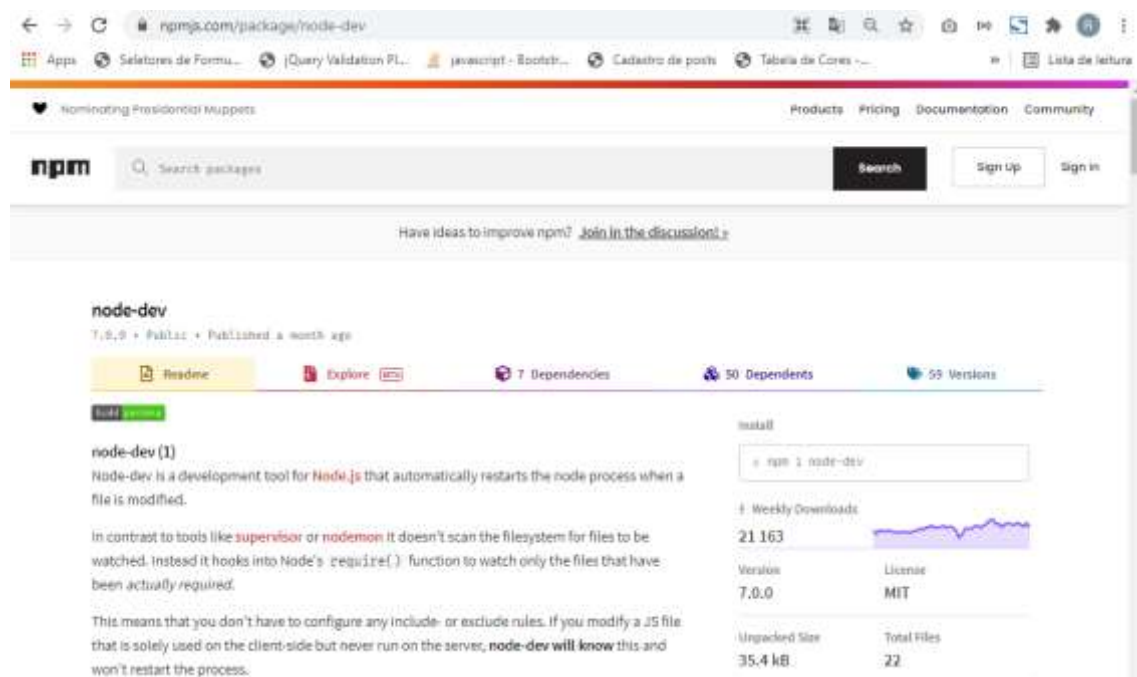


Website de tutoriais

Bem-vindo ao meu site!

Instalando o pacote node-dev

<https://www.npmjs.com/package/node-dev>



`npm install -g node-dev`

```
C:\marcos_rivas\basico\24-reinicio>npm install -g node-dev
C:\Users\beto1\AppData\Roaming\npm\node-dev -> C:\Users\beto1\AppData\Roaming\npm\node_modules\node-dev\bin\node-dev
+ node-dev@7.0.0
added 23 packages from 25 contributors in 7.565s
```

`node-dev index.js`

```
C:\marcos_rivas\basico\24-reinicio>node-dev index.js
Servidor iniciado...
```


24-reinicio\index.js

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

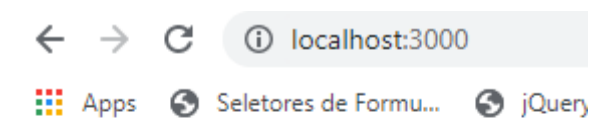
http.createServer((req, res) =>{
  res.writeHead(200, {'content-type': 'text/html'});

  res.end(`
    <!DOCTYPE html>
    <html lang="pt-br">
    <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <meta http-equiv="X-UA-Compatible" content="ie=edge">
      <title>Document</title>
    </head>
    <body>
      <h1>Website de tutoriais</h1>
      <h3>Meu nome é Roberto Pinheiro</h3>
      <h3>Bem-vindo ao meu site!</h3>
    </body>
    </html>
  `);
}).listen(3000);

console.log('Servidor iniciado...');
```

- No browser:

localhost:3000



Website de tutoriais

Meu nome é Roberto Pinheiro

Bem-vindo ao meu site!

Instalando o pacote nodemon

npm install -g nodemon

```
C:\marcos_rivas\basico\24-reinicio>npm install -g nodemon
C:\Users\betol\AppData\Roaming\npm\nodemon -> C:\Users\betol\AppData\Roaming\npm\node_modules\nodemon\bin\nodemon.js
> nodemon@2.0.7 postinstall C:\Users\betol\AppData\Roaming\npm\node_modules\nodemon
> node bin/postinstall || exit 0
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.1 (node_modules\nodemon\node_modules\chokidar\node_modules\fsevents):
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
+ nodemon@2.0.7
updated 1 package in 14.681s
```

nodemon index.js

```
C:\marcos_rivas\basico\24-reinicio>nodemon index.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Servidor iniciado...
█
```

24-reinicio\index.js

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

```
http.createServer((req, res) =>{
  res.writeHead(200, {'content-type': 'text/html'});

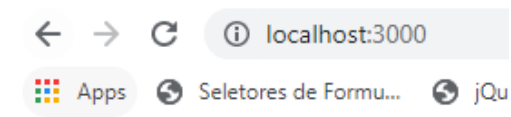
  res.end(`
    <!DOCTYPE html>
    <html lang="pt-br">
    <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <meta http-equiv="X-UA-Compatible" content="ie=edge">
      <title>Document</title>
    </head>
    <body>
      <h1>Website de tutoriais</h1>
      <h3>Tutoriais de desenvolvimento web</h3>
      <h3>Bem-vindo ao meu site!</h3>
    </body>
    </html>
  `);
}).listen(3000);
```

```
console.log('Servidor iniciado...');
```

```
C:\marcos_rivas\basico\24-reinicio>nodemon index.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Servidor iniciado...
[nodemon] restarting due to changes...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Servidor iniciado...
█
```

- No browser:

localhost:3000

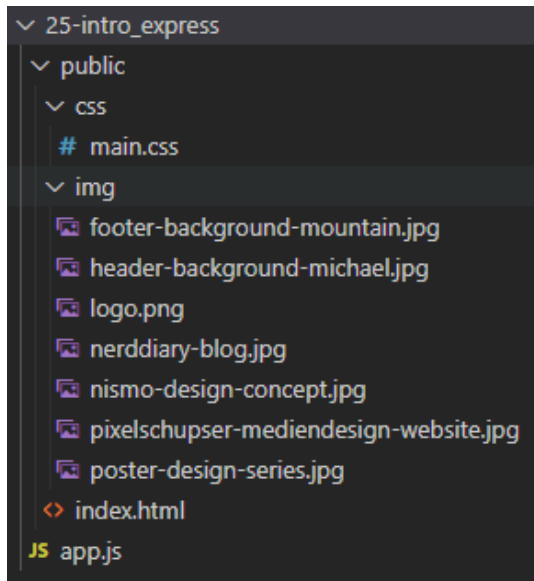


Website de tutoriais

Tutoriais de desenvolvimento web

Bem-vindo ao meu site!

Aula 25 - Introdução a Express



Instalando o pacote Express

`npm install express`

```
C:\marcos_rivas\basico\25-intro_express>npm install express
npm WARN saveError ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\25-intro_express\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\25-intro_express\package.json'
npm WARN 25-intro_express No description
npm WARN 25-intro_express No repository field.
npm WARN 25-intro_express No README data
npm WARN 25-intro_express No license field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 9.907s
found 0 vulnerabilities
```

25-intro_express\app.js

```
const express = require('express');
var app = express();

app.use(express.static('./public'));
app.listen(3000);

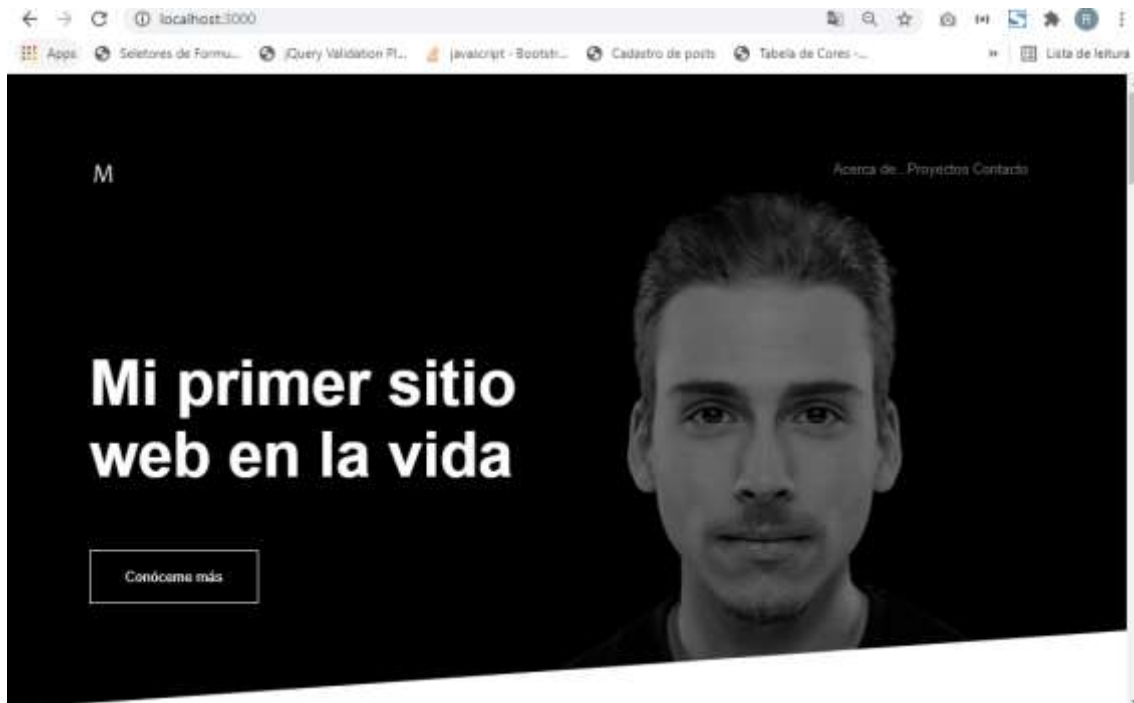
console.log('Express iniciado...');
```

nodemon app.js

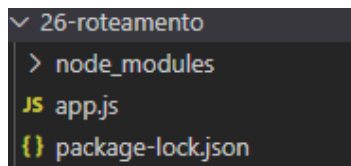
```
C:\marcos_rivas\basico\25-intro_express>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Express iniciado...
```

- No browser:

localhost:3000



Aula 26 - Roteamento em Express



26-rotaamento\app.js

```
const express = require('express');
const port = 3000;
const app = express();
const pedidos = [
  { 'nome': 'sopa', 'quantidade': 2, 'status': 'servido', 'mesa': 5},
  { 'nome': 'corte de carne', 'quantidade': 2, 'status': 'processo', 'mesa': 4},
  { 'nome': 'sopa', 'quantidade': 1, 'status': 'pago', 'mesa': 3},
  { 'nome': 'salada', 'quantidade': 1, 'status': 'pago', 'mesa': 5},
  { 'nome': 'massa', 'quantidade': 4, 'status': 'em processo', 'mesa': 1},
];

app.get('/', (req, res) =>{
  res.send('Olá a todos!');
});

app.get('/home', (req, res) =>{
  res.send('Página inicial');
});

app.get('/pedidos-em-processo', (req, res) =>{
  pedidosEmProcesso(res);
});

app.get('/pedidos-por-nome/:nome', (req, res) =>{
  pedidosPorNome(req.params.nome,res);
});

app.listen(port, () =>{
  console.log(`Servidor iniciado na porta ${port}`);
});

const pedidosEmProcesso = (res) =>{
  const arrayRes = pedidos.filter(item =>{
    return item.status == 'em processo';
  });
  res.json(arrayRes);
}

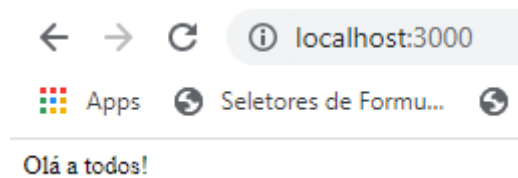
const pedidosPorNome = (nome, res) =>{
  const arrayRes = pedidos.filter(item =>{
    return item.nome == nome;
  });
  res.json(arrayRes);
}
```

nodemon app.js

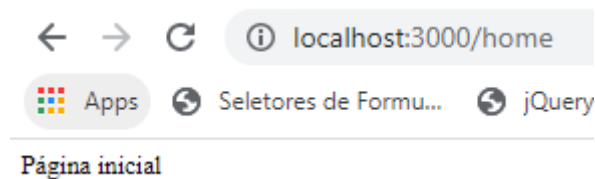
```
C:\marcos_rivas\basico\26-roteamento>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Servidor iniciado na porta 3000
```

- No browser:

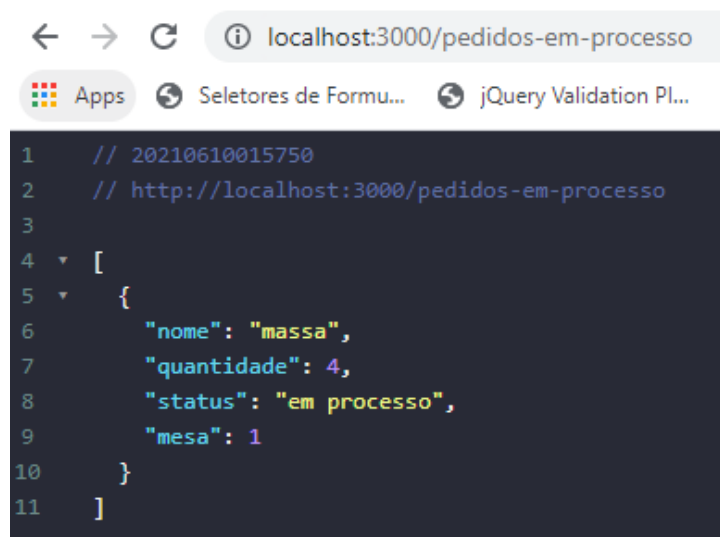
<http://localhost:3000/>



<http://localhost:3000/home>



<http://localhost:3000/pedidos-em-processo>



http://localhost:3000/pedidos-por-nome/sopa

← → ↻ ⓘ localhost:3000/pedidos-por-nome/sopa

Apps Seletores de Formu... jQuery Validation Pl... javas

```
1 // 20210610020102
2 // http://localhost:3000/pedidos-por-nome/sopa
3
4 [
5   {
6     "nome": "sopa",
7     "quantidade": 2,
8     "status": "servido",
9     "mesa": 5
10  },
11  {
12    "nome": "sopa",
13    "quantidade": 1,
14    "status": "pago",
15    "mesa": 3
16  }
17 ]
```


Aula 27 - Exemplo de middleware em Express

Instalando body-parser

`npm install body-parser`

```
C:\marcos_rivas\basico\27-middleware>npm install body-parser
npm WARN saveError ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\27-middleware\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\27-middleware\package.json'
npm WARN 27-middleware No description
npm WARN 27-middleware No repository field.
npm WARN 27-middleware No README data
npm WARN 27-middleware No license field.

+ body-parser@1.19.0
updated 1 package and audited 88 packages in 4.027s
found 0 vulnerabilities
```

27-middleware\app.js

```
const express = require('express');
const bodyParser = require('body-parser');

const app = express();

const urlencodedParser = bodyParser.urlencoded({extended: false});

app.get('/', urlencodedParser, (req, res) =>{
  res.send(`<h1>Teus dados são: <br/>
  Nome: ${req.query.nome} <br/>
  Sobrenome: ${req.query.sobrenome} </h1>`);
  console.log(req.query);
});

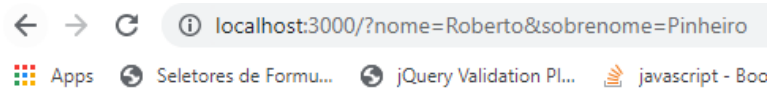
app.listen(3000, () =>{
  console.log('Servidor iniciado...');
});
```

`nodemon app.js`

```
C:\marcos_rivas\basico\27-middleware>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Servidor iniciado...
```

- No browser:

<http://localhost:3000/?nome=Roberto&sobrenome=Pinheiro>



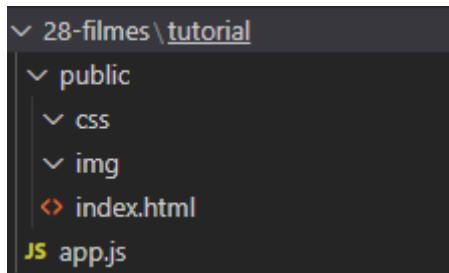
Teus dados são:

Nome: Roberto

Sobrenome: Pinheiro

```
C:\marcos_rivas\basico\27-middleware>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Servidor iniciado...
{ nome: 'Roberto', sobrenome: 'Pinheiro' }
```

Aula 28 - Exemplo de Express JS com AJAX - Parte 1/2

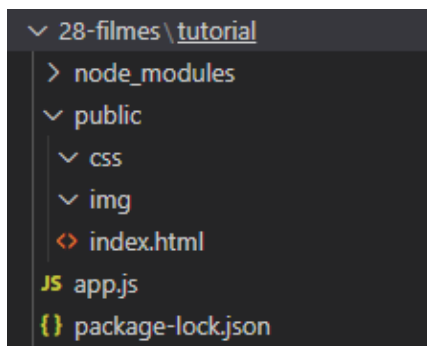


Instalando Express

npm install express

```
C:\marcos_rivas\basico\28-filmes\tutorial>npm install express
npm WARN saveError ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\28-filmes\tutorial\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\28-filmes\tutorial\package.json'
npm WARN tutorial No description
npm WARN tutorial No repository field.
npm WARN tutorial No README data
npm WARN tutorial No license field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 10.96s
found 0 vulnerabilities
```



Instalando body-parser

npm install body-parser

```
C:\marcos_rivas\basico\28-filmes\tutorial>npm install body-parser
npm WARN saveError ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\28-filmes\tutorial\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\28-filmes\tutorial\package.json'
npm WARN tutorial No description
npm WARN tutorial No repository field.
npm WARN tutorial No README data
npm WARN tutorial No license field.

+ body-parser@1.19.0
updated 1 package and audited 88 packages in 2.457s
found 0 vulnerabilities
```

Obs.: Nodemon já havia sido instalado de forma global.

28-filmes\tutorial\app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const path = require('path');
const fs = require('fs');

const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
app.use(express.static('./public'));

app.get('/', (req, res) =>{

  res.setHeader('Content-type', 'text/html');
  res.sendFile('./public/index.html');
})

app.get('/get-filmes', (req, res) =>{
  const file = fs.readFileSync('./filmes.json', 'UTF-8');
  console.log(file);

  res.setHeader('Content-type', 'text/json');
  res.send(file);
});

app.post('/new', (req, res) =>{
  res.setHeader('Content-type', 'text/plain');
  const nome = req.body.nome;
  const rating = req.body.rating;

  // abrir arquivo
  let file = fs.readFileSync('./filmes.json', 'UTF-8');

  // converte-lo em um array
  const json = JSON.parse(file);

  // inserir um novo elemento
  json.filmes.push({"nome": nome, "rating": parseInt(rating)});

  // gravar json no arquivo
  file = fs.writeFileSync('./filmes.json', JSON.stringify(json));

  res.send('Dados gravados com sucesso!');
});

app.listen(3001, () =>{
  console.log('Servidor iniciado...');
});
```

28-filmes\tutorial\filmes.json

```
{"filmes": [{"nome": "Harry Potter 4", "rating": 9}]}
```

28-filmes\tutorial\public\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Filmes</title>
  <link rel="stylesheet" href="css/main.css">
</head>
<body>
  <div id="main-container">
    <h1>Registro de filmes</h1>
    <form action="/new" method="POST">
      <p>
        <label for="nome">Nome do filme</label><br/>
        <input type="text" name="nome" id="nome">
      </p>
      <p>
        <label for="rating">Qualificação</label><br/>
        <input type="range" min="0" max="10" name="rating" id="rating">
      </p>

      <input type="submit" value="Registrar novo filme" id="bSubmit">
    </form>

    <div id="filmes-container">

      </div>
    </div>
  </body>
</html>
```

28-filmes\tutorial\public\css\main.css

```
body{
  font-family: Arial, Helvetica, sans-serif;
  padding: 0;
  margin: 0;
}

form{
  padding-bottom: 20px;
}

#main-container{
  margin: 0 auto;
  width: 400px;
}
```

```
input[type="text"]{
  padding: 10px;
  width: 350px;
}

input[type="button"]{
  color: white;
  background-color: rgba(18, 69, 163, 255);
  padding: 10px;
  border: 0;
}


#filmes-container div{
  padding: 10px 0px;
  border-bottom: solid 1px #ccc;
}
```

nodemon app.js

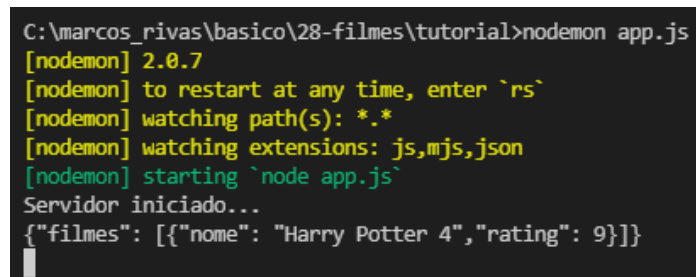
```
C:\marcos_rivas\basico\28-filmes\tutorial>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Servidor iniciado...
```

- No browser:

<http://localhost:3001/get-filmes>

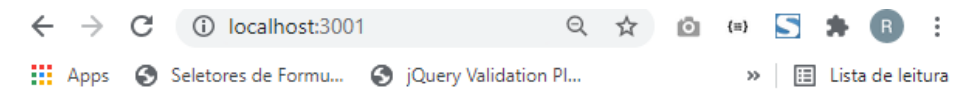


```
1 // 20210610051041
2 // http://localhost:3001/get-filmes
3
4 {
5   "filmes": [
6     {
7       "nome": "Harry Potter 4",
8       "rating": 9
9     }
10  ]
11 }
```



```
C:\marcos_rivas\basico\28-filmes\tutorial>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Servidor iniciado...
{"filmes": [{"nome": "Harry Potter 4","rating": 9}]}
```

localhost:3001



Registro de filmes

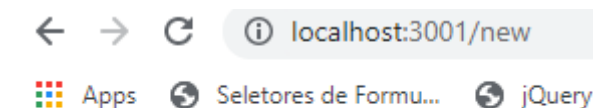
Nome do filme

O salvavidas

Qualificação



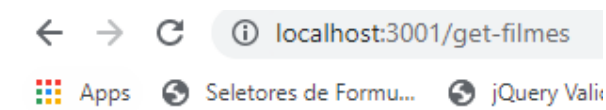
Registrar novo filme



Dados gravados com sucesso!

- No browser:

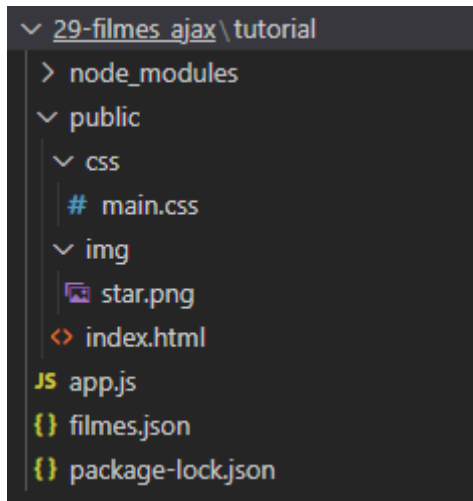
<http://localhost:3001/get-filmes>



```
1 // 20210610052018
2 // http://localhost:3001/get-filmes
3
4 {
5   "filmes": [
6     {
7       "nome": "Harry Potter 4",
8       "rating": 9
9     },
10    {
11      "nome": "O salvavidas",
12      "rating": 10
13    }
14  ]
15 }
```

```
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
Servidor iniciado...
{"filmes":[{"nome":"Harry Potter 4","rating":9},{"nome":"O salvavidas","rating":10}]}
```


Aula 29 - Exemplo de Express JS com AJAX - Parte 2/2



29-filmes_ajax\tutorial\app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const path = require('path');
const fs = require('fs');

const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
app.use(express.static('./public'));

app.get('/', (req, res) =>{

  res.setHeader('Content-type', 'text/html');
  res.sendFile('./public/index.html');
})

app.get('/get-filmes', (req, res) =>{
  const file = fs.readFileSync('./filmes.json', 'UTF-8');
  console.log(file);

  res.setHeader('Content-type', 'text/json');
  res.send(file);
});

app.post('/new', (req, res) =>{
  res.setHeader('Content-type', 'text/plain');
  const nome = req.body.nome;
  const rating = req.body.rating;

  // abrir arquivo
  let file = fs.readFileSync('./filmes.json', 'UTF-8');
```

```

// converte-lo em um array
const json = JSON.parse(file);

// inserir um novo elemento
json.filmes.push({"nome": nome, "rating": parseInt(rating)});

// gravar json no arquivo
file = fs.writeFileSync('./filmes.json', JSON.stringify(json));

res.send('Dados gravados com sucesso!');

});

app.listen(3001, () =>{
  console.log('Servidor iniciado...');
});

```

29-filmes_ajax\tutorial\public\index.html

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Filmes</title>
  <link rel="stylesheet" href="css/main.css">
</head>
<body>
  <div id="main-container">
    <h1>Registro de filmes</h1>
    <form action="/new" method="POST">
      <p>
        <label for="nome">Nome do filme</label><br/>
        <input type="text" name="nome" id="nome">
      </p>
      <p>
        <label for="rating">Qualificação</label><br/>
        <input type="range" min="0" max="10" name="rating" id="rating">
      </p>

      <input type="submit" value="Registrar novo filme" id="bSubmit">
    </form>

    <div id="filmes-container">

    </div>
  </div>
</body>
</html>

```

```
<script>
const bSubmit = document.querySelector('#bSubmit');

bSubmit.addEventListener('click', () =>{
  //validar campos
  const nome = document.querySelector('#nome').value;
  const rating = document.querySelector('#rating').value;

  if(nome.trim() === '' || rating.trim() === '') return false;

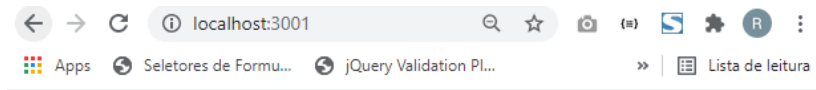
  // enviar requisição POST a /new
  fetch('/new', {
    method: 'POST',
    headers: {'Content-type': 'application/json'},
    body: JSON.stringify({nome: nome, rating: rating})
  })
  .then(res => res.text())
  .then(data =>{
    // mostrar mensagem de erro/sucesso
    alert(data);

    //atualizar lista de filmes
    loadFilmes();
  });
});

function loadFilmes(){
  fetch('/get-filmes', {method: 'GET'})
  .then(res => res.json())
  .then(data =>{
    const filmes = document.querySelector('#filmes-container');
    let html = '';
    data.filmes.forEach(filme =>{
      html += `<div>${filme.nome} ${filme.rating}</div>`
    });
    filmes.innerHTML = html;
  });
}
</script>
```

- No browser:

localhost:3001



Registro de filmes

Nome do filme

Gladiador

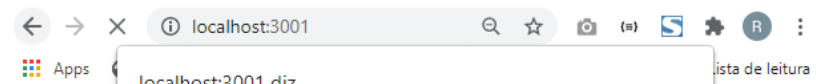
Qualificação



Registrar novo filme

Harry Potter 4 ★9

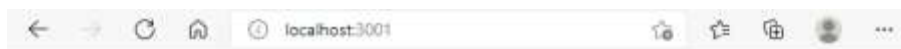
O salvavidas ★10



localhost:3001 diz

Dados gravados com sucesso!

OK



Registro de filmes

Nome do filme

Hunger Games

Qualificação

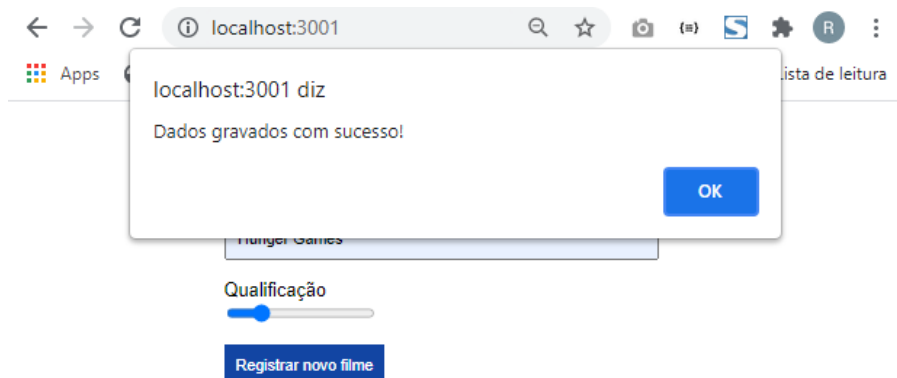
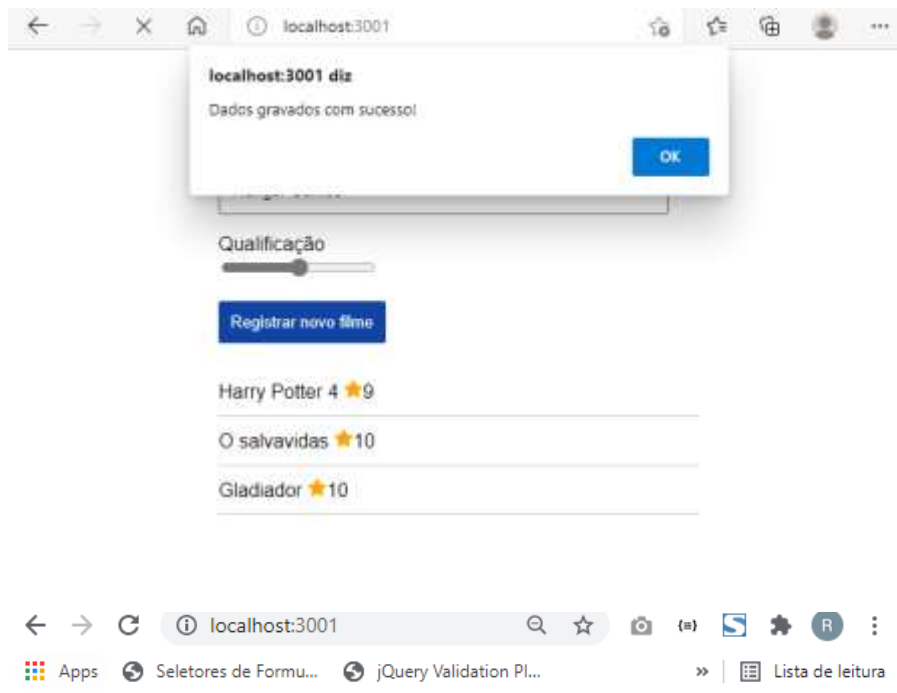


Registrar novo filme

Harry Potter 4 ★9

O salvavidas ★10

Gladiador ★10

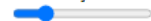


Registro de filmes

Nome do filme

Hunger Games

Qualificação



Registrar novo filme

Harry Potter 4 ★9

O salvavidas ★10

Gladiador ★10

Hunger Games ★2

Aula 30 - Tutorial de Pug para fazer templates dinâmicos

Instalando o Express

npm install --save Express

```
C:\marcos_rivas\basico\30-templates_pug>npm install --save express
npm WARN saveError ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\30-templates_pug\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\30-templates_pug\package.json'
npm WARN 30-templates_pug No description
npm WARN 30-templates_pug No repository field.
npm WARN 30-templates_pug No README data
npm WARN 30-templates_pug No license field.

+ express@4.17.1
added 28 packages from 25 contributors and audited 72 packages in 7.208s
found 0 vulnerabilities
```

Instalando o body-parser

npm install body-parser

```
C:\marcos_rivas\basico\30-templates_pug>npm install body-parser
npm WARN saveError ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\30-templates_pug\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\30-templates_pug\package.json'
npm WARN 30-templates_pug No description
npm WARN 30-templates_pug No repository field.
npm WARN 30-templates_pug No README data
npm WARN 30-templates_pug No license field.

+ body-parser@1.19.0
added 22 packages from 17 contributors and audited 23 packages in 6.191s
found 0 vulnerabilities
```

Instalando pug

npm install pug

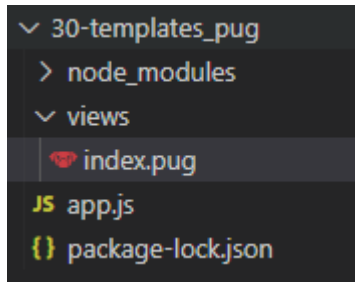
```
C:\marcos_rivas\basico\30-templates_pug>npm install pug
npm WARN saveError ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\30-templates_pug\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\30-templates_pug\package.json'
npm WARN 30-templates_pug No description
npm WARN 30-templates_pug No repository field.
npm WARN 30-templates_pug No README data
npm WARN 30-templates_pug No license field.

+ pug@3.0.2
added 41 packages from 16 contributors and audited 129 packages in 14.211s

6 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- No diretório raiz da aplicação crie uma pasta chamada "**views**" e dentro dela um arquivo chamado "**index.pug**":



30-templates_pug\app.js

```
const express = require('express');
const app = express();
const bodyParser = require('body-parser');

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
app.set('view engine', 'pug');

app.get('/', (req, res) =>{
  res.render('index', {titulo: "Minha primeira app", nome: "Roberto Pinheiro"});
});

app.listen(3002, () =>{
  console.log('Servidor iniciado...');
});
```

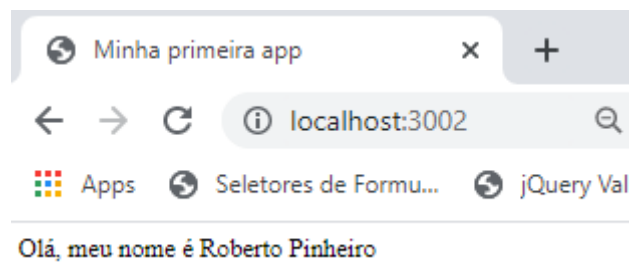
30-templates_pug\views\index.pug

```
html
  head
    title= titulo
  body
    div Olá, meu nome é #{nome}
```

node app.js

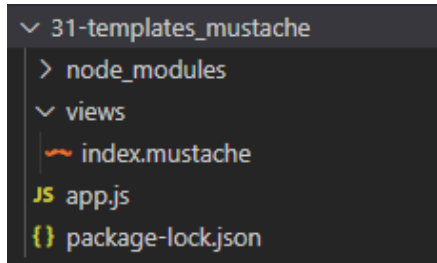
```
C:\marcos_rivas\basico\30-templates_pug>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Servidor iniciado...
█
```


localhost:3002



Aula 31 - Tutorial de Mustache para fazer templates dinâmicas

Instalando mustache-express



npm install mustache-express

```
C:\marcos_rivas\basico\31-templates_mustache>npm install mustache-express
npm WARN saveError ENOENT: no such file or directory, open 'C:\marcos_rivas\basico\31-templates_mustache\package.json'
npm WARN EBADENGINE EBADENGINE: no such file or directory, open 'C:\marcos_rivas\basico\31-templates_mustache\package.json'
npm WARN 31-templates_mustache No description
npm WARN 31-templates_mustache No repository field.
npm WARN 31-templates_mustache No README data
npm WARN 31-templates_mustache No license field.

+ mustache-express@1.3.1
added 5 packages from 11 contributors and audited 157 packages in 8.484s

6 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

31-templates_mustache\app.js

```
const express = require('express');
const app = express();
const bodyParser = require('body-parser');
const mustacheExpress = require('mustache-express');
```

```
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
```

```
app.engine('.mustache', mustacheExpress());
app.set('view engine', 'mustache');
```

```
app.get('/', (req, res) =>{
  const dados = [
    {nome: 'Helena', idade: 10},
    {nome: 'Luis', idade: 40}
  ];
```

```
  res.render('index',{
    titulo: 'Minha primeira app',
    nome: 'Juan Perez',
    dados: dados
  });
});
```

```
app.listen(3002, () =>{
  console.log('Servidor iniciado...');
});
```

31-templates_mustache\views\index.mustache

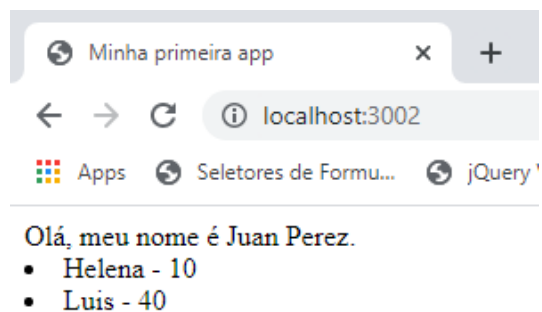
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>{{titulo}}</title>
</head>
<body>
  Olá, meu nome é {{nome}}.

  {{#dados}}
    <li>{{nome}} - {{idade}}</li>
  {{/dados}}
</body>
</html>
```

node app.js

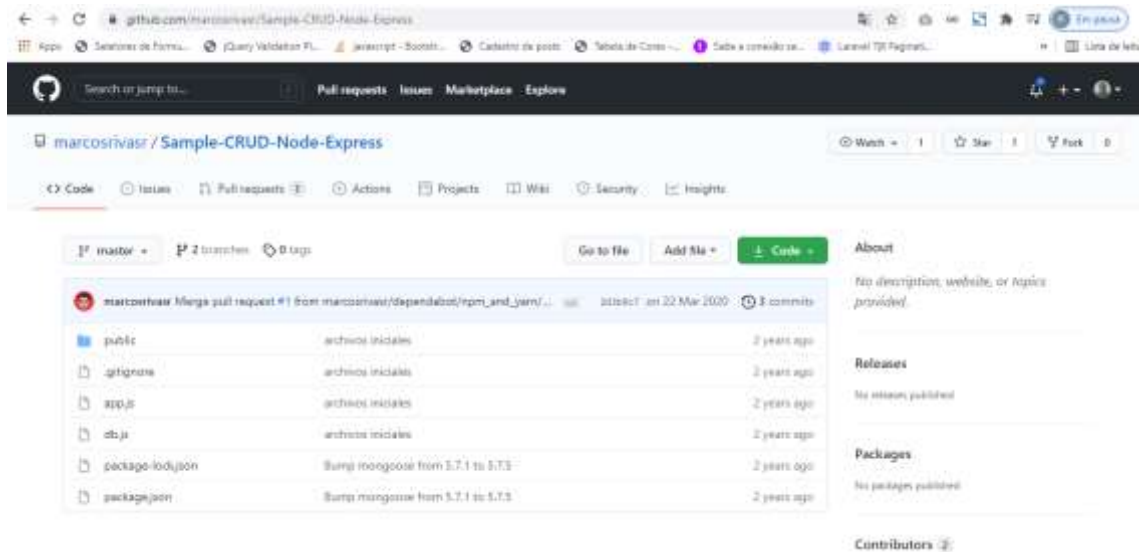
```
C:\marcos_rivas\basico\31-templates_mustache>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Servidor iniciado...
```

localhost:3002



Aula 32 - CRUD - Conexão a base de dados

<https://github.com/marcosrivasr/Sample-CRUD-Node-Express>



Instalando e configurando o Mongo Express

O Mongo Express é uma interface para gerenciamento de banco de dados.

- Instale o Mongo Express de forma global em seu computador:

```
npm install -g mongo-express
```

- Acesse a pasta:

```
C:\Users\beto1\AppData\Roaming\npm\node_modules\mongo-express
```

- Crie um arquivo chamado "**config.js**" e copie o conteúdo do arquivo "**config.default.js**" para dentro dele.

- Dentro deste arquivo há um trecho que permite alterar o nome de usuário e a senha para acessar o aplicativo:

```
basicAuth: {
  username: getEnv(basicAuthUsername) || 'admin',
  password: getEnv(basicAuthPassword) || 'pass',
},
```

- Por segurança é recomendado alterá-lo, porém se isso não for feito, os valores padrões são:

```
user: admin
password: pass
```

Rodando o Mongo Express

- Inicialmente rode o servidor do Mongo. No terminal entre com o comando:

mongod

```
C:\marcos_rivas\basico\32-conexao>mongod
{"t":{"$date":"2021-06-11T09:18:10.308-03:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"main","msg":"Automatically
disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2021-06-11T09:18:11.653-03:00"},"s":"W", "c":"ASIO", "id":22681, "ctx":"main","msg":"No Transport
layer configured during NetworkInterface startup"}
{"t":{"$date":"2021-06-11T09:18:11.654-03:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"main","msg":"Implicit TCP
FastOpen in use."}
{"t":{"$date":"2021-06-11T09:18:11.657-03:00"},"s":"I", "c":"STORAGE", "id":4615611, "ctx":"initandlisten","msg":"Mong
OOD starting","attr":{"pid":9776,"port":27017,"dbPath":"C:/data/db/","architecture":"64-bit","host":"DESKTOP-NUIKR9E"}}
{"t":{"$date":"2021-06-11T09:18:11.658-03:00"},"s":"I", "c":"CONTROL", "id":23308, "ctx":"initandlisten","msg":"Targ
et operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2021-06-11T09:18:11.658-03:00"},"s":"I", "c":"CONTROL", "id":23483, "ctx":"initandlisten","msg":"Buil
d Info","attr":{"buildInfo":{"version":"4.4.6","gitVersion":"72e66213c2c3eab37d9358d5e78ad7f5cid0d0d7","modules":[],"all
ocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2021-06-11T09:18:11.658-03:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten","msg":"Oper
ating System","attr":{"os":{"name":"Microsoft Windows 10","version":"18.0 (build 16299)}}}}
{"t":{"$date":"2021-06-11T09:18:11.658-03:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten","msg":"Opti
ons set by command line","attr":{"options":{}}}
{"t":{"$date":"2021-06-11T09:18:11.669-03:00"},"s":"I", "c":"STORAGE", "id":22270, "ctx":"initandlisten","msg":"Stor
age engine to use detected by data files","attr":{"dbpath":"C:/data/db/","storageEngine":"wiredtiger"}}
{"t":{"$date":"2021-06-11T09:18:11.671-03:00"},"s":"I", "c":"STORAGE", "id":22315, "ctx":"initandlisten","msg":"Open
ing WiredTiger","attr":{"config":{"create,cache_size=144M,session_max=33000,eviction=(threads_min=4,threads_max=4),confi
g_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_t
ime=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose=[recovery_progress,checkpoi
nt_progress,compact_progress]}}}}
{"t":{"$date":"2021-06-11T09:18:12.498-03:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten","msg":"Wired
tiger message","attr":{"message":{"[1623413892:498201][9776:140721844210512]. txn-recover: [WT_VERB_RECOVERY_PROGRESS] R
ecovering log 6 through 7"}}}}
```

- Para rodar o Mongo Express, no terminal, entre com o comando:

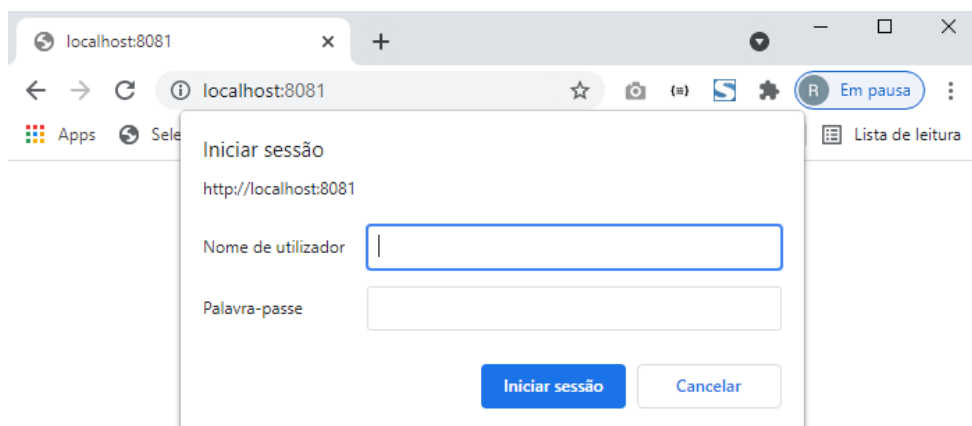
mongo-express --admin --url mongodb://127.0.0.1:27017

```
C:\Users\beto1>mongo-express --admin --url mongodb://127.0.0.1:27017
Welcome to mongo-express

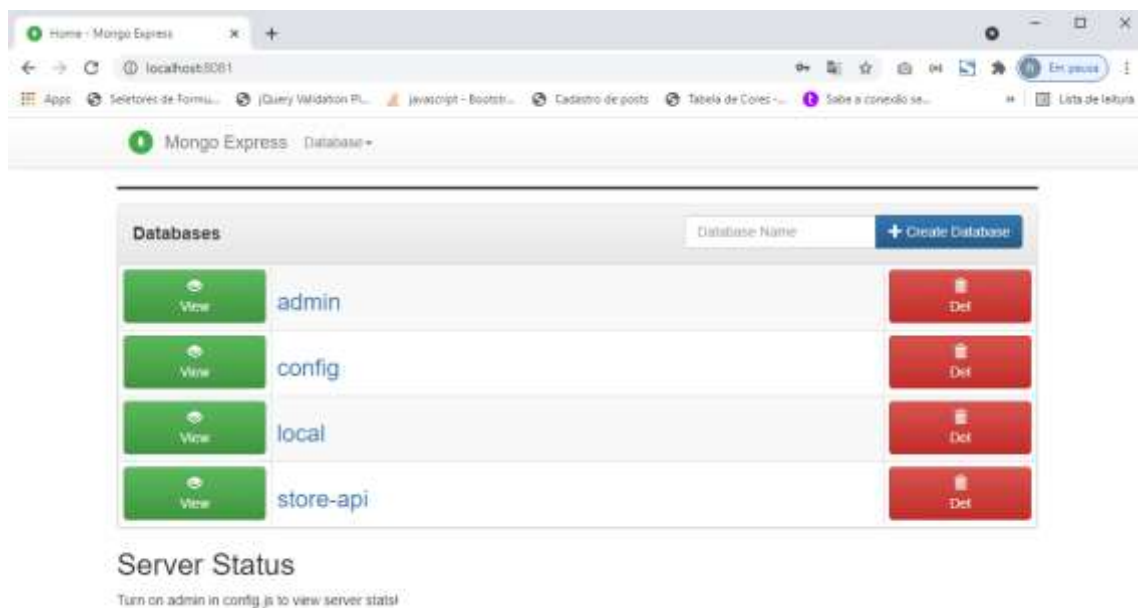
(node:4320) [MONGODB DRIVER] Warning: Current Server Discovery and Monitoring engine is deprecated, and will be removed in a future ver
sion. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
(Use "node --trace-warnings ..." to show where the warning was created)
Mongo Express server listening at http://localhost:8081
```

- No browser:

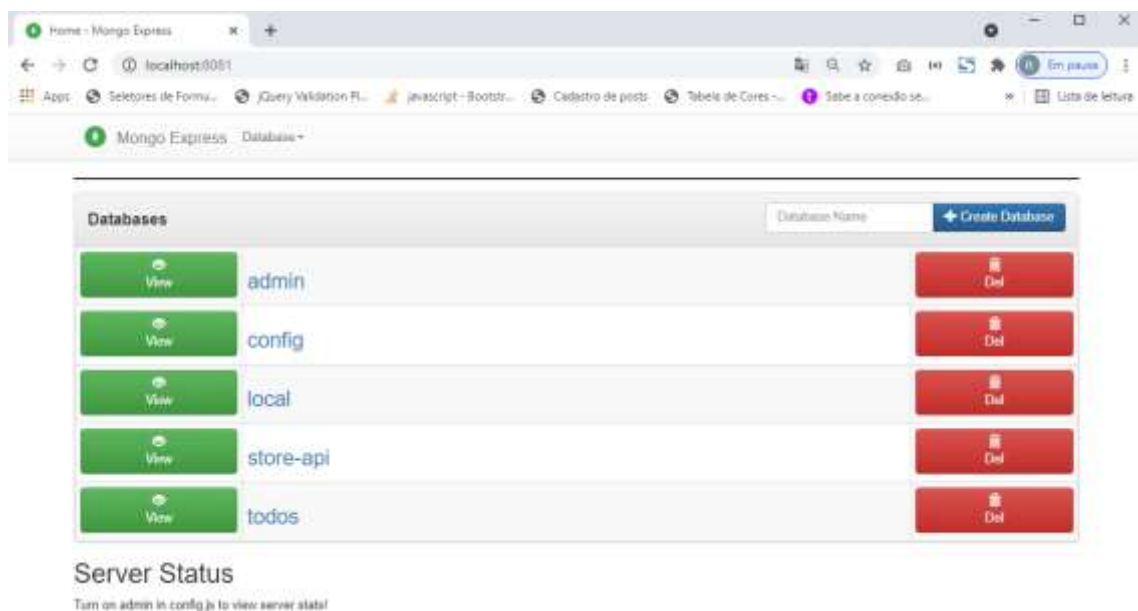
localhost:8081



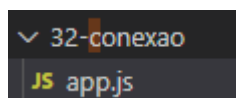
- Entre com os valores configurados e clique no botão "Iniciar sessão"



- Crie um database com o nome "todos"



- No Visual Studio Code, na pasta raiz do projeto, adicione um arquivo chamado "app.js":

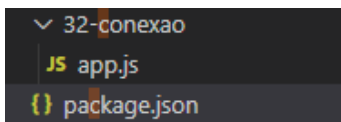


- No terminal, entre com o comando:

npm init -y

```
C:\marcos_rivas\basico>npm init -y
Wrote to C:\marcos_rivas\basico\package.json:

{
  "name": "basico",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```



Instalando os pacotes Express e Mongoose

npm install express mongoose

```
C:\marcos_rivas\basico>npm install express mongoose
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN basico@1.0.0 No description
npm WARN basico@1.0.0 No repository field.

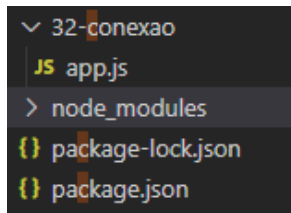
+ express@4.17.1
+ mongoose@5.12.13
added 79 packages from 123 contributors and audited 79 packages in 33.675s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

package.json

```
{
  "name": "basico",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.1",
    "mongoose": "^5.12.13"
  }
}
```



Instalando o body-parser de forma global

`npm install -g body-parser`

```
C:\marcos_rivas\basico>npm install -g body-parser
+ body-parser@1.19.0
added 22 packages from 17 contributors in 6.759s
```

Criando um usuário para o BD "todos" no Mongo Express

- No terminal, entre com o comando:

`mongo`

`show databases;`

```
C:\Users\batulimongo
MongoDB shell version v4.4.0
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id": "UUID('a979600f-e84c-40fe-810a-a76d900a50e4') " }
MongoDB server version: 4.4.0
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
  https://community.mongodb.com
***
The server generated these startup warnings when booting:
  2021-06-11T09:18:18.161-03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2021-06-11T09:18:18.161-03:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip addr
  ess 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
***
  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()
***
> show databases;
admin            0.000008
config           0.000008
local            0.000008
store-api        0.000008
todos            0.000008
>
```

use todos;

```
> use todos;
switched to db todos
```

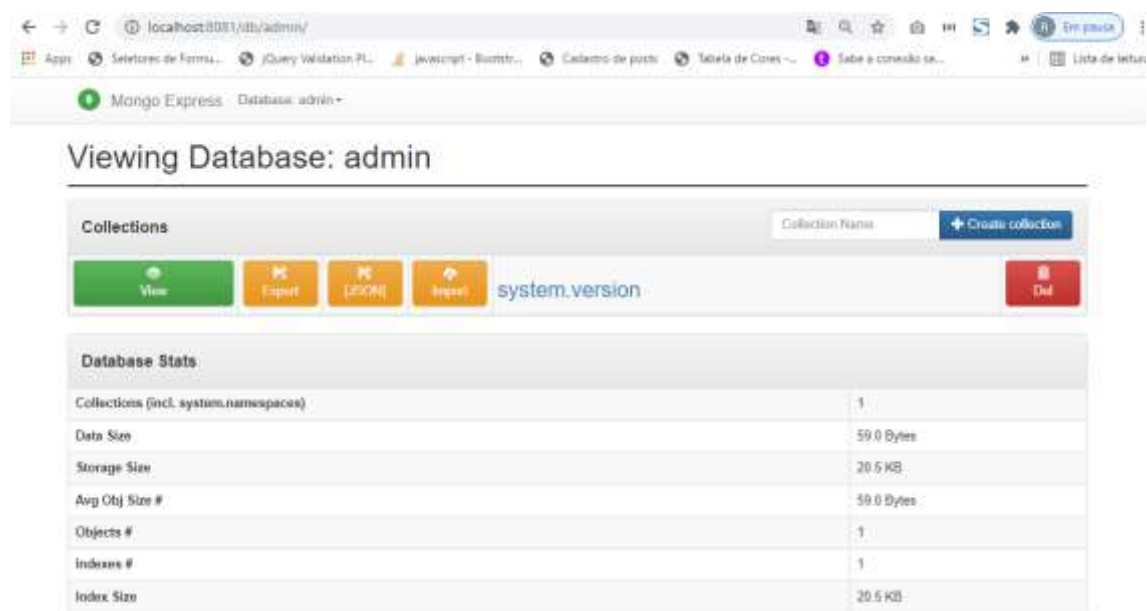


```
db.createUser({
  user: 'dev',
  pwd: 'dev',
  roles: [
    { role: 'readWrite', db: 'todos' }
  ]
});
```

```
> db.createUser({
... user: 'dev',
... pwd: 'dev',
... roles: [
... { role: 'readWrite', db: 'todos' }
... ]
... });
Successfully added user: {
  "user" : "dev",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "todos"
    }
  ]
}
```

- No Mongo Express:

- Clique no link "**admin**":



- Clique no link "**system.users**":

localhost:8081/db/admin/system.users

Mongo Express Database: admin Collection: system.users

Viewing Collection: system.users

New Document New Index

Simple Advanced

Key Value String Find

Delete all 1 documents selected

| _id | username | user | db | credentials | roles |
|-----------|-------------------------|------|-------|---|-------------------------|
| todos dev | reTouqjRfmmiQLF7hwEHq== | dev | todos | <pre>{ "screen-size-1": @{...}, "screen-size-2": @{...} }</pre> | <pre>{ @{...} }</pre> |

Criando a conexão com a base de dados

app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const app = express();

// conexão com o banco de dados
mongoose.connect('mongodb://dev:dev@localhost:27017/todos', {useNewUrlParser: true,
useUnifiedTopology: true});
const connection = mongoose.connection;

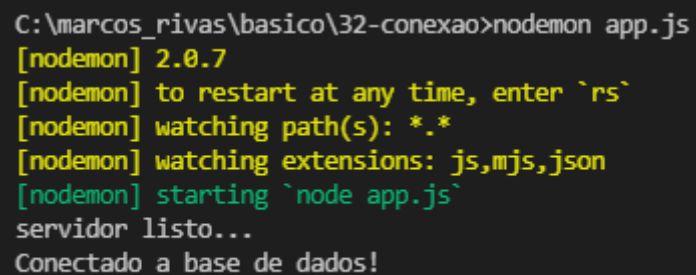
connection.once('open', () =>{
  console.log('Conectado a base de dados!');
});

connection.on('error', (err) =>{
  console.log('Erro ao se conectar a base de dados: ', err);
});

app.get('/', (req, res) => {
  res.json({response: 'success'});
});

app.listen(3000, () =>{
  console.log('servidor listo...');
});
```

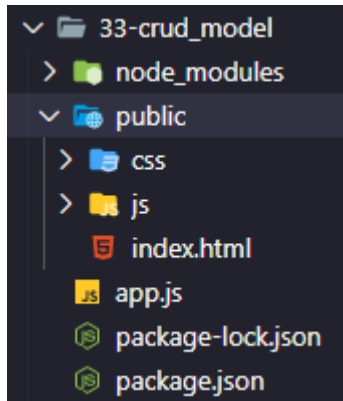
node app.js



```
C:\marcos_rivas\basico\32-conexao>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
servidor listo...
Conectado a base de dados!
```

Aula 33 - CRUD - Criando o modelo

- Dentro da pasta raiz do projeto adicione uma subpasta chamada "public" e dentro desta pasta adicione um arquivo chamado "index.html" e duas subpastas: "css" e "js":



app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
app.use(express.static(__dirname + '/public'));

// conexão com o banco de dados
mongoose.connect('mongodb://dev:dev@localhost:27017/todos', {useNewUrlParser: true,
useUnifiedTopology: true});
const connection = mongoose.connection;

connection.once('open', () =>{
  console.log('Conectado a base de dados!');
});

connection.on('error', (err) =>{
  console.log('Erro ao se conectar a base de dados: ', err);
});

//modelo
const Todo = mongoose.model('Todo', {text: String, completed: Boolean});

app.get('/', (req, res) => {
  res.json({response: 'success'});
});

app.listen(3000, () =>{
  console.log('servidor listo...');
});
```

33-crud_model\public\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>CRUD Node.js + Express.js + MongoDB</title>
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <div id="main-container">
      <form action="/add" method="post">
        <input
          type="text"
          name="text"
          id=""
          autocomplete="off"
          placeholder="Insira um novo todo"
        />
      </form>

      <div id="todos"></div>

      <script src="js/main.js"></script>
    </body>
  </html>
```

33-crud_model\public\css\style.css

```
body {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 24px;
}
#main-container {
  margin: 0 auto;
  width: 600px;
}

#main-container form input {
  font-size: 24px;
  padding: 20px;
  width: 558px;
}

#todos .todo {
  width: 100%;
  margin: 10px 0;
}

#todos .todo div {
  display: inline-block;
  padding: 10px 5px;
}
#todos .todo .checkbox-container {
  width: 25px;
}

#todos .todo .checkbox-container input {
  transform: scale(2);
}

#todos .todo .text-container {
  color: black;
  width: 495px;
}

#todos .todo .completed {
  color: #ccc;
  text-decoration: line-through;
}

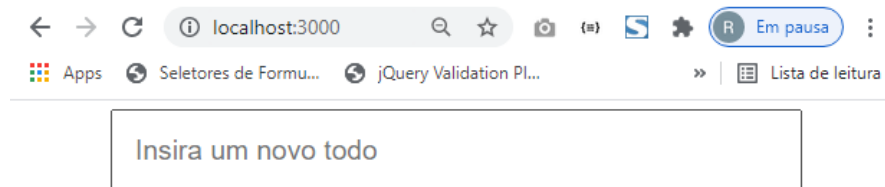
#todos .todo .actions-container a {
  color: #ccc;
  font-weight: bolder;
  text-decoration: none;
  padding: 10px;
}
#todos .todo .actions-container a:hover {
  color: rgb(30, 71, 136);
}
```

nodemon app.js

```
C:\marcos_rivas\basico\33-crud_model>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
servidor listo...
Conectado a base de dados!
█
```

- No browser:

localhost:3000



Aula 34 - CRUD - create

34-crud_create\app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

app.use(express.static(__dirname + '/public'));

// conexão com o banco de dados
mongoose.connect('mongodb://dev:dev@localhost:27017/todos', {useNewUrlParser: true,
useUnifiedTopology: true});
const connection = mongoose.connection;

connection.once('open', () =>{
  console.log('Conectado a base de dados!');
});

connection.on('error', (err) =>{
  console.log('Erro ao se conectar a base de dados: ', err);
});

//modelo
const Todo = mongoose.model('Todo', {text: String, completed: Boolean});

app.post('/add', (req, res) =>{
  const todo = new Todo({text: req.body.text, completed: false});

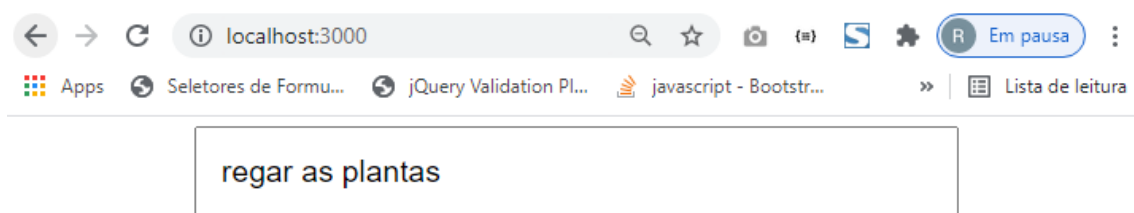
  todo.save()
    .then(doc =>{
      console.log('Dado inserido com sucesso!', doc);
      //res.redirect('/');
      res.json({response: 'success'});
    })
    .catch(err =>{
      console.log('Erro ao inserir ', err.message);
      res.status(400).json({response: 'failed'});
    });
});

app.listen(3000, () =>{
  console.log('servidor listo...');
});
```

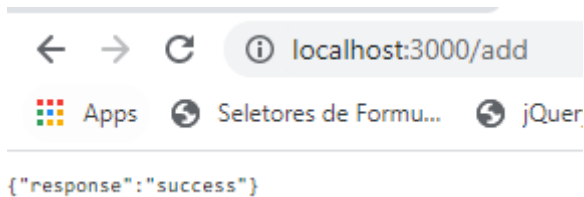

nodemon app.js

```
C:\marcos_rivas\basico\34-crud_create>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
servidor listo...
Conectado a base de dados!
```

localhost:3000



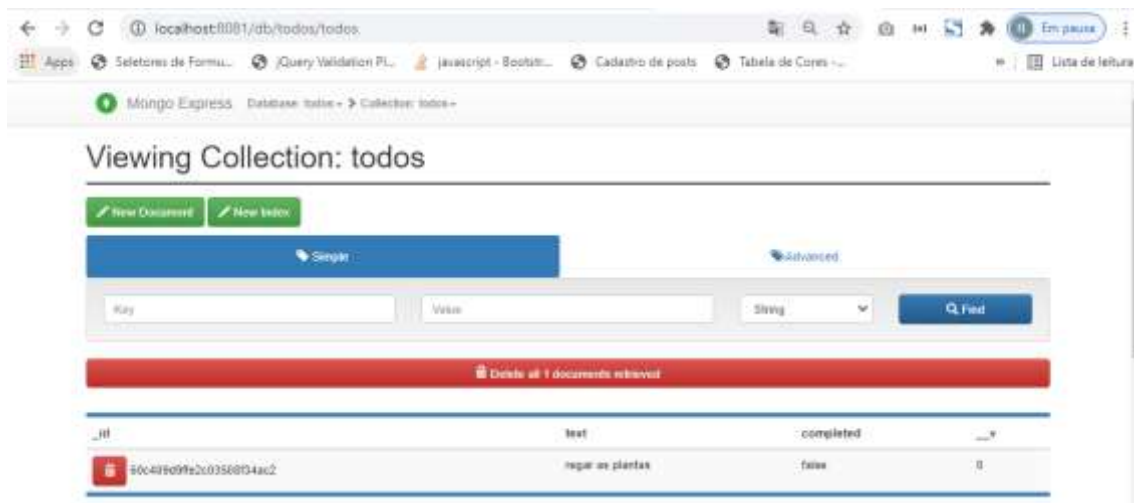
http://localhost:3000/add



```
Dado inserido com sucesso! {
  _id: 60c409d9ffe2c03508f34ac2,
  text: 'regar as plantas',
  completed: false,
  __v: 0
}
```

- No Mongo Express:

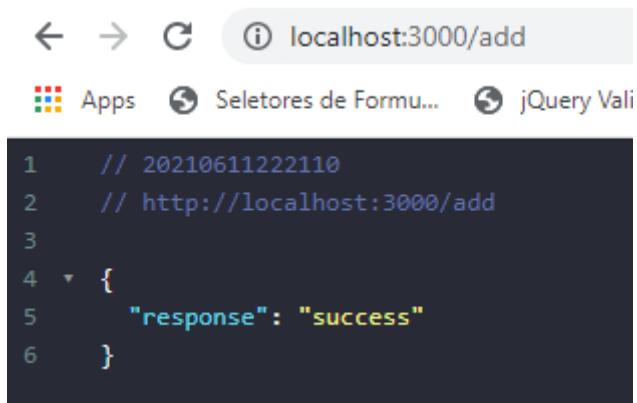
mongo-express --admin --url mongodb://127.0.0.1:27017



localhost:3000



localhost:3000/add



```
Dado inserido com sucesso! {
  _id: 60c40c05ffe2c03508f34ac3,
  text: 'Terminar minha tarefa',
  completed: false,
  __v: 0
}
```

- No Mongo Express:

The screenshot shows the Mongo Express web interface in a browser. The address bar indicates the URL is `localhost:8081/db/todos/todos`. The interface title is "Mongo Express" with subtext "Database: todos -> Collection: todos".

The main heading is "Viewing Collection: todos". Below this, there are two green buttons: "New Document" and "New Index".

There are two tabs: "Simple" (selected) and "Advanced".

Below the tabs is a search bar with fields for "Key" and "Value", a dropdown menu set to "String", and a "Find" button.

A red banner message states: "Delete all 2 documents retrieved".

Below the banner is a table with the following data:

| _id | text | completed | _v |
|-------------------------|-----------------------|-----------|----|
| 60c409d9ff2c0350ff54ac2 | regar as plantas | false | 0 |
| 60c40c05ff2c0350ff54ac3 | Terminar minha tarefa | false | 0 |

Aula 35 - CRUD - consulta

app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

app.use(express.static(__dirname + '/public'));

// conexão com o banco de dados
mongoose.connect('mongodb://dev:dev@localhost:27017/todos', {useNewUrlParser: true,
useUnifiedTopology: true});
const connection = mongoose.connection;

connection.once('open', () =>{
  console.log('Conectado a base de dados!');
});

connection.on('error', (err) =>{
  console.log('Erro ao se conectar a base de dados: ', err);
});

//modelo
const Todo = mongoose.model('Todo', {text: String, completed: Boolean});

app.post('/add', (req, res) =>{
  const todo = new Todo({text: req.body.text, completed: false});

  todo.save()
  .then(doc =>{
    console.log('Dado inserido com sucesso!', doc);
    //res.redirect('/');
    res.json({response: 'success'});
  })
  .catch(err =>{
    console.log('Erro ao inserir ', err.message);
    res.status(400).json({response: 'failed'});
  });
});

app.get('/getall', (req, res) =>{
  Todo.find({}, 'text completed')
  .then(doc =>{
    res.json({response: 'success', data: doc});
  })
  .catch(err =>{
    console.log('Erro ao consultar elementos!', err.message);
    res.status(400).json({response: 'failed'});
  });
});
```

```
app.listen(3000, () =>{
  console.log('servidor listo...');
});
```

<http://localhost:3000/getall>



The screenshot shows a web browser window with the address bar displaying `localhost:3000/getall`. Below the address bar, there are tabs for 'Apps', 'Seletores de Formu...', and 'jQuery Validation'. The main content area of the browser displays a JSON response from the server, which is a list of two tasks. The first task has an ID of `60c409d9ffe2c03508f34ac2`, the text `regar as plantas`, and is not completed. The second task has an ID of `60c40c05ffe2c03508f34ac3`, the text `Terminar minha tarefa`, and is also not completed.

```
1 // 20210611223751
2 // http://localhost:3000/getall
3
4 {
5   "response": "success",
6   "data": [
7     {
8       "_id": "60c409d9ffe2c03508f34ac2",
9       "text": "regar as plantas",
10      "completed": false
11    },
12    {
13      "_id": "60c40c05ffe2c03508f34ac3",
14      "text": "Terminar minha tarefa",
15      "completed": false
16    }
17  ]
18 }
```

Aula 36 - CRUD - update

app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

app.use(express.static(__dirname + '/public'));

// conexão com o banco de dados
mongoose.connect('mongodb://dev:dev@localhost:27017/todos', {useNewUrlParser: true,
useUnifiedTopology: true});
const connection = mongoose.connection;

connection.once('open', () =>{
  console.log('Conectado a base de dados!');
});

connection.on('error', (err) =>{
  console.log('Erro ao se conectar a base de dados: ', err);
});

//modelo
const Todo = mongoose.model('Todo', {text: String, completed: Boolean});

app.post('/add', (req, res) =>{
  const todo = new Todo({text: req.body.text, completed: false});

  todo.save()
  .then(doc =>{
    console.log('Dado inserido com sucesso!', doc);
    //res.redirect('/');
    res.json({response: 'success'});
  })
  .catch(err =>{
    console.log('Erro ao inserir ', err.message);
    res.status(400).json({response: 'failed'});
  });
});

app.get('/getall', (req, res) =>{
  Todo.find({}, 'text completed')
  .then(doc =>{
    res.json({response: 'success', data: doc});
  })
  .catch(err =>{
    console.log('Erro ao consultar elementos!', err.message);
    res.status(400).json({response: 'failed'});
  });
});
```

```

app.get('/complete/:id/:status', (req, res) =>{

  const id = req.params.id;
  const status = req.params.status == 'true'; // converter para booleano

  Todo.findByIdAndUpdate({_id: id},{ $set: {completed: status}})
  .then(doc =>{
    res.json({response: 'success'});
  })
  .catch(err =>{
    console.log('Error ao atualizar dado ', err.message);
    res.status(400).json({response: 'failed'});
  });

});

app.listen(3000, () =>{
  console.log('servidor listo...');
});

```

nodemon app.js

```

C:\marcos_rivas\basico\36-crud_update>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
servidor listo...
Conectado a base de dados!

```

- No browser:

<http://localhost:3000/complete/60c409d9ffe2c03508f34ac2/true>

← → ↻ ⓘ localhost:3000/complete/60c409d9ffe2c03508f34ac2/true

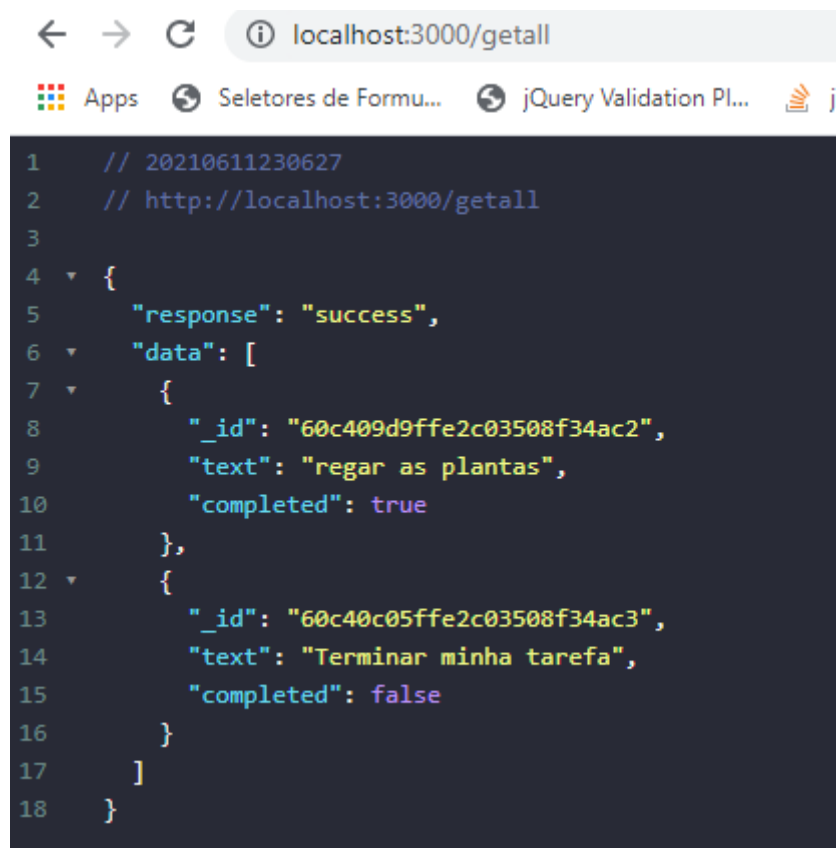
Apps Seletores de Formu... jQuery Validation Pl... javascript - Bootstr...

```

1 // 20210611230524
2 // http://localhost:3000/complete/60c409d9ffe2c03508f34ac2/true
3
4 {
5   "response": "success"
6 }

```

http://localhost:3000/getall



```
1 // 20210611230627
2 // http://localhost:3000/getall
3
4 {
5   "response": "success",
6   "data": [
7     {
8       "_id": "60c409d9ffe2c03508f34ac2",
9       "text": "regar as plantas",
10      "completed": true
11    },
12    {
13      "_id": "60c40c05ffe2c03508f34ac3",
14      "text": "Terminar minha tarefa",
15      "completed": false
16    }
17  ]
18 }
```


Aula 37 - CRUD - delete

app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

app.use(express.static(__dirname + '/public'));

// conexão com o banco de dados
mongoose.connect('mongodb://dev:dev@localhost:27017/todos', {useNewUrlParser: true,
useUnifiedTopology: true});
const connection = mongoose.connection;

connection.once('open', () =>{
  console.log('Conectado a base de dados!');
});

connection.on('error', (err) =>{
  console.log('Erro ao se conectar a base de dados: ', err);
});

//modelo
const Todo = mongoose.model('Todo', {text: String, completed: Boolean});

app.post('/add', (req, res) =>{
  const todo = new Todo({text: req.body.text, completed: false});

  todo.save()
  .then(doc =>{
    console.log('Dado inserido com sucesso!', doc);
    //res.redirect('/');
    res.json({response: 'success'});
  })
  .catch(err =>{
    console.log('Erro ao inserir ', err.message);
    res.status(400).json({response: 'failed'});
  });
});

app.get('/getall', (req, res) =>{
  Todo.find({}, 'text completed')
  .then(doc =>{
    res.json({response: 'success', data: doc});
  })
  .catch(err =>{
    console.log('Erro ao consultar elementos!', err.message);
    res.status(400).json({response: 'failed'});
  });
});
```

```

app.get('/complete/:id/:status', (req, res) =>{

  const id = req.params.id;
  const status = req.params.status == 'true'; // converter para booleano

  Todo.findByIdAndUpdate({_id: id},{ $set: {completed: status}})
  .then(doc =>{
    res.json({response: 'success'});
  })
  .catch(err =>{
    console.log('Erro ao atualizar dado ', err.message);
    res.status(400).json({response: 'failed'});
  });

});

app.get('/delete/:id', (req, res) =>{
  const id = req.params.id;

  Todo.findByIdAndDelete({_id: id})
  .then(doc =>{
    //res.json({response: 'success'});
    res.redirect('/');
  })
  .catch(err =>{
    console.log('Erro ao excluir dado ', err.message);
    res.status(400).json({response: 'failed'});
  });
});

app.listen(3000, () =>{
  console.log('servidor listo...');
});

```

nodemon app.js

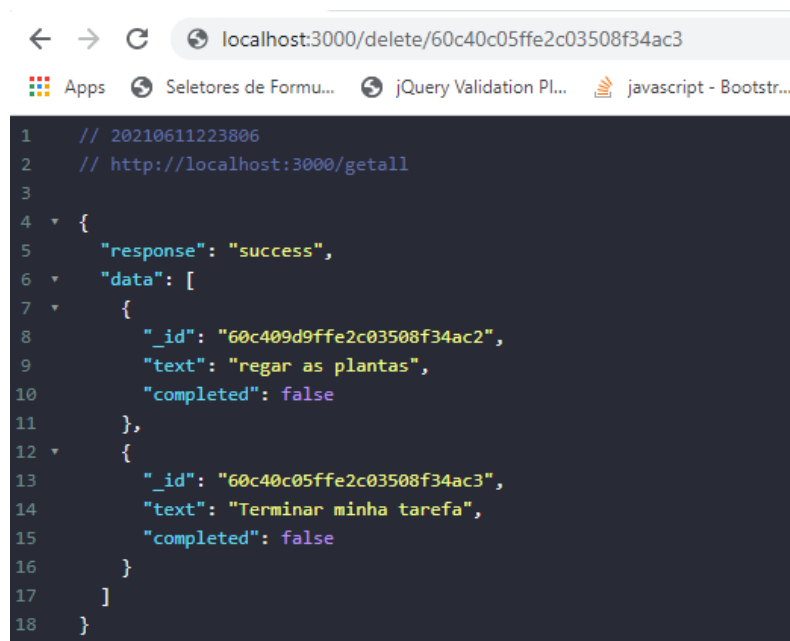
```

C:\marcos_rivas\basico\37-crud_delete>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
servidor listo...
Conectado a base de dados!

```

- No browser:

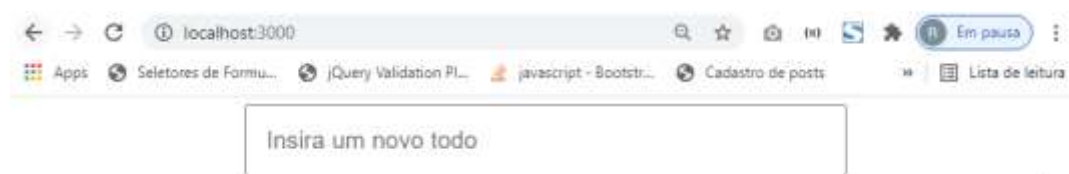
<http://localhost:3000/delete/60c40c05ffe2c03508f34ac3>



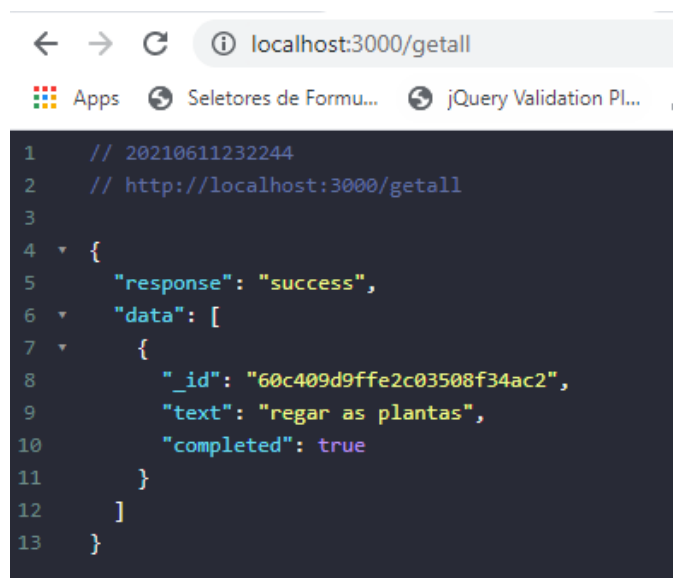
The screenshot shows a web browser with the address bar displaying `localhost:3000/delete/60c40c05ffe2c03508f34ac3`. Below the address bar, there are several tabs: "Apps", "Seletores de Formu...", "jQuery Validation Pl...", and "javascript - Bootstr...". The main content area displays a REST client response for a DELETE request. The response is as follows:

```
1 // 20210611223806
2 // http://localhost:3000/getall
3
4 {
5   "response": "success",
6   "data": [
7     {
8       "_id": "60c409d9ffe2c03508f34ac2",
9       "text": "regar as plantas",
10      "completed": false
11    },
12    {
13      "_id": "60c40c05ffe2c03508f34ac3",
14      "text": "Terminar minha tarefa",
15      "completed": false
16    }
17  ]
18 }
```

localhost:3000



<http://localhost:3000/getall>



The screenshot shows a web browser with the address bar displaying `localhost:3000/getall`. Below the address bar, there are several tabs: "Apps", "Seletores de Formu...", "jQuery Validation Pl...", and "javascript - Bootstr...". The main content area displays a REST client response for a GET request. The response is as follows:

```
1 // 20210611232244
2 // http://localhost:3000/getall
3
4 {
5   "response": "success",
6   "data": [
7     {
8       "_id": "60c409d9ffe2c03508f34ac2",
9       "text": "regar as plantas",
10      "completed": true
11    }
12  ]
13 }
```

Aula 38 - CRUD - completo

- Dentro da pasta "public/js" adicione um arquivo chamado "main.js":

38-crud_completo\public\js\main.js

```
window.addEventListener('DOMContentLoaded', (e) => {
  updateTodos();
});

const mapCheckboxes = () => {
  document.querySelectorAll('.complete-checkbox').forEach(item => {
    item.addEventListener('click', async e => {
      const id = e.target.parentNode.parentNode.id;
      let classes = e.target.parentNode.parentNode.childNodes[3].className.replace('completed',
        '').trim();

      const completed = e.target.checked;
      const res = await updateTodo(id, completed);

      if(res.response === 'success'){
        if(completed){
          e.target.parentNode.parentNode.childNodes[3].className += 'completed';
        } else {
          e.target.parentNode.parentNode.childNodes[3].className += 'classes';
        }
      }
    });
  });
};

const updateTodos = () => {
  fetch('http://localhost:3000/getall')
    .then(res => res.json())
    .then(data => {
      if(data.response === 'success'){
        const todos = data.data;

        document.querySelector('#todos').innerHTML = '';
        todos.forEach(todo => {
          document.querySelector('#todos').innerHTML += `
          <div class='todo' id='${todo._id}'>
            <div class="checkbox-container"><input type="checkbox" class="complete-checkbox"
name="" id="" ${((todo.completed === true)? 'checked': '')}</div>
            <div class="text-container ${((todo.completed === true)? 'completed': '')}>${todo.text}</div>
            <div class="actions-container"><a href="/delete/${todo._id}">X</a></div>
          </div>
          `;
        });

        mapCheckboxes();
      }
    })
    .catch(err => console.error(err));
}
```

```

const updateTodo = async (id, completed) =>{
  const res = await fetch('http://localhost:3000/complete/' + id + '/' + completed)
    .then(res => res.json());
  return res;
}

document.querySelector('#formulario').addEventListener('submit', e => {
  e.preventDefault();

  const text = document.querySelector('#text').value;

  if(text === ''){
    return false;
  }

  fetch('http://localhost:3000/add', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({text: text})
  })
  .then(res => res.json())
  .then(data => {
    if(data.response === 'success'){
      updateTodos();
      document.querySelector('#text').value = "";
    }
  });
});

```

38-crud_model\public\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>CRUD Node.js + Express.js + MongoDB</title>
    <link rel="stylesheet" href="css/style.css" />
  </head>
  <body>
    <div id="main-container">
      <form id="formulario" action="/add" method="post">
        <input
          type="text"
          name="text"
          id="text"
          autocomplete="off"
          placeholder="Insira um novo todo"
        />
      </form>

      <div id="todos"></div>
    </div>

    <script src="js/main.js"></script>
  </body>
</html>
```

38-crud_completo\app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));

app.use(express.static(__dirname + '/public'));

// conexão com o banco de dados
mongoose.connect('mongodb://dev:dev@localhost:27017/todos', {useNewUrlParser: true,
useUnifiedTopology: true});
const connection = mongoose.connection;

connection.once('open', () =>{
  console.log('Conectado a base de dados!');
});

connection.on('error', (err) =>{
  console.log('Erro ao se conectar a base de dados: ', err);
});

//modelo
const Todo = mongoose.model('Todo', {text: String, completed: Boolean});

app.post('/add', (req, res) =>{
  const todo = new Todo({text: req.body.text, completed: false});

  todo.save()
  .then(doc =>{
    console.log('Dado inserido com sucesso!', doc);
    // res.redirect('/');
    res.json({response: 'success'});
  })
  .catch(err =>{
    console.log('Erro ao inserir ', err.message);
    res.status(400).json({response: 'failed'});
  });
});

app.get('/getall', (req, res) =>{
  Todo.find({}, 'text completed')
  .then(doc =>{
    res.json({response: 'success', data: doc});
  })
  .catch(err =>{
    console.log('Erro ao consultar elementos!', err.message);
    res.status(400).json({response: 'failed'});
  });
});
```

```

app.get('/complete/:id/:status', (req, res) =>{
  const id = req.params.id;
  const status = req.params.status == 'true'; // converter para booleano

  Todo.findByIdAndUpdate({_id: id},{ $set: {completed: status}})
  .then(doc =>{
    res.json({response: 'success'});
  })
  .catch(err =>{
    console.log('Erro ao atualizar dado ', err.message);
    res.status(400).json({response: 'failed'});
  });
});

app.get('/delete/:id', (req, res) =>{
  const id = req.params.id;
  Todo.findByIdAndDelete({_id: id})
  .then(doc =>{
    //res.json({response: 'success'});
    res.redirect('/');
  })
  .catch(err =>{
    console.log('Erro ao excluir dado ', err.message);
    res.status(400).json({response: 'failed'});
  });
});

app.listen(3000, () =>{
  console.log('servidor listo...');
});

```

nodemon app.js

```

C:\marcos_rivas\basico\38-crud_completo>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
servidor listo...
Conectado a base de dados!

```


localhost:3000

localhost:3000

AppsSeletores de Formu...jQuery Validation Pl...javascript - Bootstr...Cadastro de posts»Lista de leitura

Insira um novo todo

☒

regar-as-plantasX

localhost:3000

AppsSeletores de Formu...jQuery Validation Pl...javascript - Bootstr...Cadastro de posts»Lista de leitura

Ir ao curso de manhã

☒

regar-as-plantasX

☐Ir caminharX

☐Passear com o cachorroX

☐Terminar minha tarefaX

☐AlmoçarX

localhost:3000

AppsSeletores de Formu...jQuery Validation Pl...javascript - Bootstr...Cadastro de posts»Lista de leitura

Insira um novo todo

☒

regar-as-plantasX

☐Ir caminharX

☐Passear com o cachorroX

☐Terminar minha tarefaX

☐AlmoçarX

☐Ir ao curso de manhãX

Insira um novo todo

| | | |
|-------------------------------------|------------------------|---|
| <input checked="" type="checkbox"/> | regar-as plantas | X |
| <input type="checkbox"/> | Ir caminhar | X |
| <input type="checkbox"/> | Passear com o cachorro | X |
| <input type="checkbox"/> | Terminar minha tarefa | X |
| <input type="checkbox"/> | Almoçar | X |
| <input checked="" type="checkbox"/> | Ir ao curso de manhã | X |

Insira um novo todo

| | | |
|-------------------------------------|------------------------|---|
| <input type="checkbox"/> | Ir caminhar | X |
| <input type="checkbox"/> | Passear com o cachorro | X |
| <input type="checkbox"/> | Terminar minha tarefa | X |
| <input type="checkbox"/> | Almoçar | X |
| <input checked="" type="checkbox"/> | Ir ao curso de manhã | X |

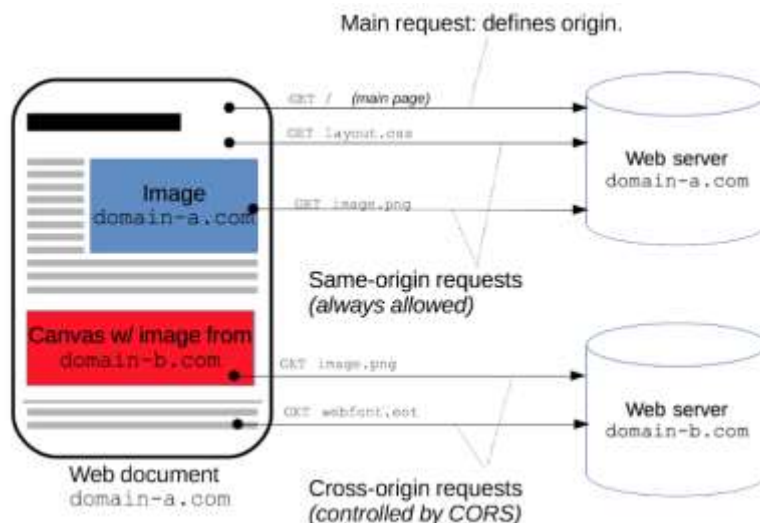
Aula 39 - O que é CORS e como configurá-lo?

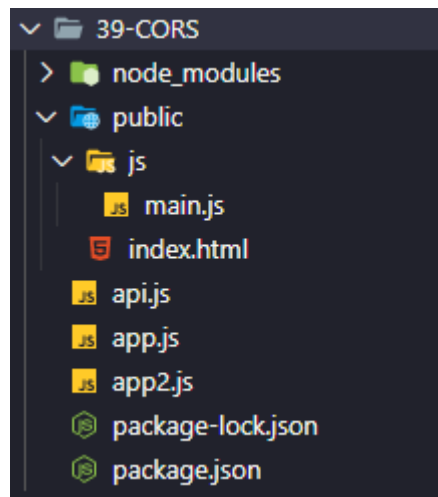
<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/CORS>



Cross-Origin Resource Sharing (Compartilhamento de recursos com origens diferentes) é um mecanismo que usa cabeçalhos adicionais HTTP para informar a um navegador que permita que um aplicativo Web seja executado em uma origem (domínio) com permissão para acessar recursos selecionados de um servidor em uma origem distinta. Um aplicativo Web executa uma requisição cross-origin HTTP ao solicitar um recurso que tenha uma origem diferente (domínio, protocolo e porta) da sua própria origem.

Por motivos de segurança, navegadores restringem requisições cross-origin HTTP iniciadas por scripts. Por exemplo, XMLHttpRequest e Fetch API seguem a política de mesma origem (same-origin policy). Isso significa que um aplicativo web que faz uso dessas APIs só poderá fazer solicitações para recursos de mesma origem da qual o aplicativo foi carregado, a menos que a resposta da outra origem inclua os cabeçalhos CORS corretos.





39-CORS\app.js

```
const express = require('express');
const bodyParser = require('body-parser');

const app = express();

app.use(express.static('./public/'));

app.listen(3000, () =>{
  console.log('Servidor 1 iniciado...');
});
```

39-CORS\public\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>CORS</title>
  </head>
  <body>
    <h1>Olá Mundo!</h1>

    <script>
      fetch('http://localhost:3001')
        .then((res) => res.json())
        .then((data) => {
          console.log(data);
        });
    </script>
  </body>
</html>
```

node app.js

```
C:\marcos_rivas\basico\39-CORS>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Servidor 1 iniciado...
█
```

39-CORS\app2.js

```
const express = require('express');
const bodyParser = require('body-parser');

const app = express();

app.use(express.static('./public/'));

app.listen(3002, () =>{
  console.log('Servidor 2 iniciado...');
});
```

node app2.js

```
C:\marcos_rivas\basico\39-CORS>nodemon app2.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app2.js`
Servidor 2 iniciado...
```

39-CORS\api.js

```
const express = require('express');
const bodyParser = require('body-parser');

const app = express();

app.get('/', (req, res) => {
  res.json({message: 'ok'});
});

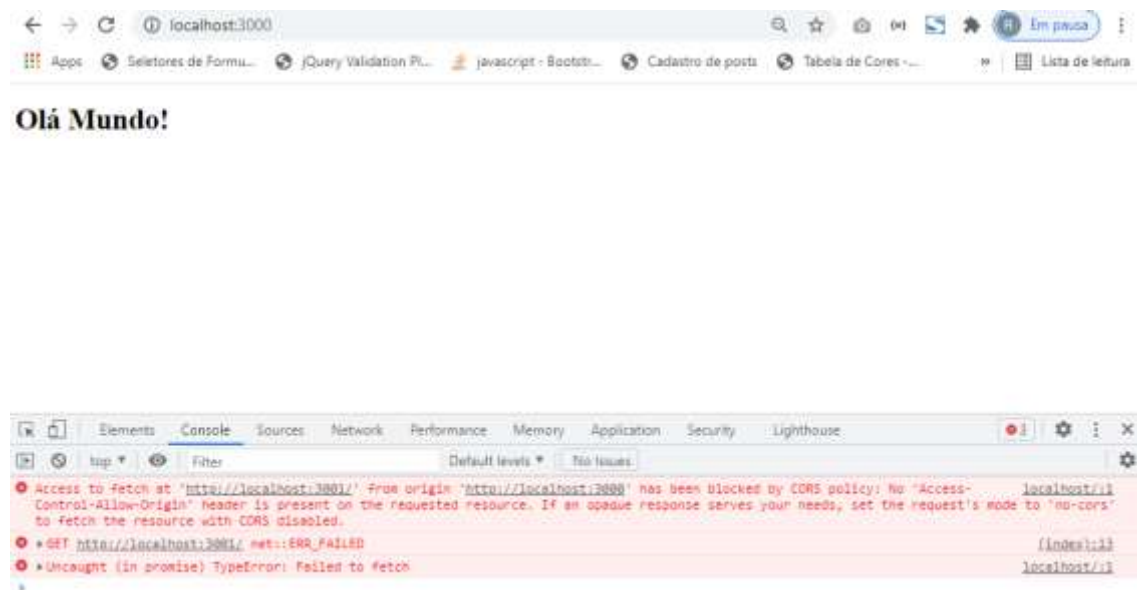
app.listen(3001, () =>{
  console.log('API iniciada...');
});
```

nodemon api.js

```
C:\marcos_rivas\basico\39-CORS>nodemon api.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node api.js`
API iniciada...
█
```

- No browser:

localhost:3000



Instalando CORS

npm install cors

```
C:\marcos_rivas\basico\39-CORS>npm install cors
npm WARN basic@1.0.0 No description
npm WARN basic@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os"
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os"
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os"
: "darwin", "arch": "any"} (current: {"os": "win32", "arch": "x64"})
+ cors@2.8.5
added 2 packages from 2 contributors and audited 199 packages in 9.347s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

39-CORS\api.js

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');

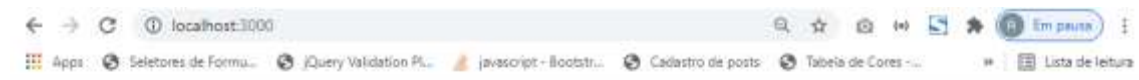
const app = express();
app.use(cors());

app.get('/', (req, res) =>{
  res.json({message: 'ok'});
});

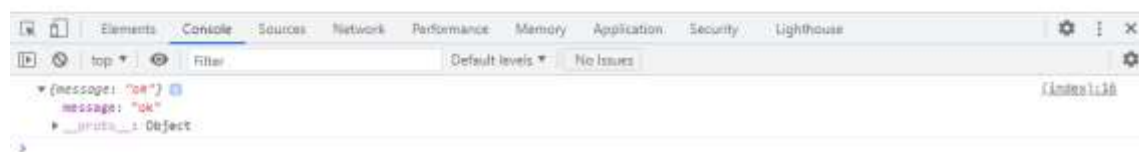
app.listen(3001, () =>{
  console.log('API iniciada...');
});
```

- No browser:

localhost:3000



Olá Mundo!



39-CORS\api.js

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');

const app = express();
app.use(cors());

var whitelist = ['http://localhost:3000']
var corsOptions = {
  origin: function (origin, callback) {
    if (whitelist.indexOf(origin) !== -1) {
      callback(null, true)
    } else {
      callback(new Error('Not allowed by CORS'))
    }
  }
}

app.get('/', cors(corsOptions), (req, res) =>{
  res.json({message: 'ok'});
});

app.listen(3001, () =>{
  console.log('API iniciada...');
});
```

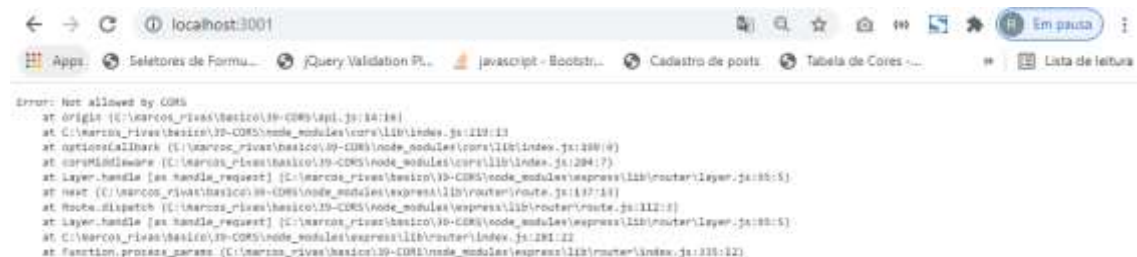
- No browser:

localhost:3002



```
C:\marcos_rivas\basico\39-CORS>nodemon api.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node api.js`
API iniciada...
Error: Not allowed by CORS
    at origin (C:\marcos_rivas\basico\39-CORS\api.js:14:16)
    at C:\marcos_rivas\basico\39-CORS\node_modules\cors\lib\index.js:219:13
    at optionsCallback (C:\marcos_rivas\basico\39-CORS\node_modules\cors\lib\index.js:199:9)
    at corsMiddleware (C:\marcos_rivas\basico\39-CORS\node_modules\cors\lib\index.js:204:7)
    at Layer.handle [as handle_request] (C:\marcos_rivas\basico\39-CORS\node_modules\express\lib\router\layer.js:95:5)
    at next (C:\marcos_rivas\basico\39-CORS\node_modules\express\lib\router\route.js:137:13)
    at Route.dispatch (C:\marcos_rivas\basico\39-CORS\node_modules\express\lib\router\route.js:112:3)
    at Layer.handle [as handle_request] (C:\marcos_rivas\basico\39-CORS\node_modules\express\lib\router\layer.js:95:5)
    at C:\marcos_rivas\basico\39-CORS\node_modules\express\lib\router\index.js:281:22
    at Function.process_params (C:\marcos_rivas\basico\39-CORS\node_modules\express\lib\router\index.js:335:12)
```

localhost:3001



Aula 40 - Registro e login completo com NodeJS e MongoDB

npm init -y

```
C:\marcos_rivas\basico\40-login>npm init -y
wrote to C:\marcos_rivas\basico\40-login\package.json:

{
  "name": "40-login",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Instalando express, bcrypt e mongoose

- No terminal entre com o seguinte comando:

npm install express bcrypt mongoose

```
C:\marcos_rivas\basico\40-login>npm install express bcrypt mongoose

> bcrypt@5.0.1 install C:\marcos_rivas\basico\40-login\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build

[bcrypt] Success: "C:\marcos_rivas\basico\40-login\node_modules\bcrypt\lib\binding\napi-v3\bcrypt_lib.node" is ins
talled via remote
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN 40-login@1.0.0 No description
npm WARN 40-login@1.0.0 No repository field.

+ express@4.17.1
+ mongoose@5.12.13
+ bcrypt@5.0.1
added 136 packages from 229 contributors and audited 136 packages in 36.878s

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
40-login
├── node_modules
│   ├── package-lock.json
│   └── package.json
```

- Na pasta raiz do projeto, adicione um arquivo chamado "app.js":

```
40-login
├── node_modules
│   ├── package-lock.json
│   └── package.json
└── app.js
```

40-login\app.js

```
const express = require('express');
const path = require('path');
const bodyParser = require('body-parser');
const app = express();

const bcrypt = require('bcrypt');
const mongoose = require('mongoose');

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));

app.use(express.static(path.join(__dirname, 'public')));

const mongo_uri = 'mongodb://dev:dev@localhost/todos';

mongoose.connect(mongo_uri, function(err) {
  if(err) {
    throw err;
  } else {
    console.log(`Successfully connected to ${mongo_uri}`);
  }
})

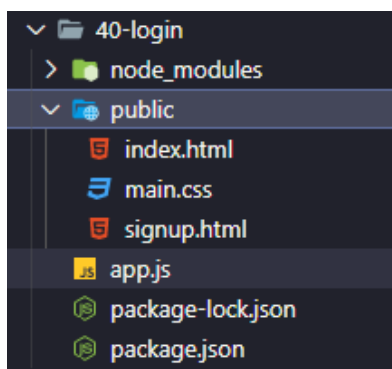
app.get('/', (req, res) => {

});

app.listen(3000, () => {
  console.log('Server started...');
})

module.exports = app;
```

- Na pasta raiz do projeto, adicione uma pasta chamada "public" com três arquivos: "index.html, main.css, signup.html":



40-login\public\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Login</title>
  <link rel="stylesheet" href="main.css" />
</head>
<body>
  <form action="/authenticate" method="POST">
    <h2>Login</h2>
    <div class="section">
      <div class="title">Username</div>
      <div class="field">
        <input type="text" id="username" name="username" />
      </div>
    </div>
    <div class="section">
      <div class="title">Password</div>
      <div class="field">
        <input type="password" id="password" name="password" />
      </div>
    </div>
    <div class="section">
      <div class="button"><input type="submit" value="Login" /></div>
      <div class="button"><a href="signup.html">Registrar-se</a></div>
    </div>
  </form>
</body>
</html>
```

40-login\public\main.css

```
* {
  box-sizing: border-box;
}

body {
  font-family: Arial, Helvetica, sans-serif;
}

form {
  width: 400px;
  margin: 0 auto;
  border: solid 1px #ccc;
  padding: 20px;
}

input[type='text'],
input[type='password'] {
  border: solid 1px #ccc;
  border-radius: 3px;
  font-size: 18px;
  outline: none;
  padding: 10px;
  width: 100%;
}

input[type='submit'] {
  border: 0;
  background-color: #003366;
  border-radius: 3px;
  font-size: 18px;
  color: white;
  outline: 0;
  padding: 10px 30px;
}

.section {
  padding: 15px 0;
}

.section .title {
  padding: 10px 0;
}

.button {
  padding: 10px 0;
  text-align: center;
}

.button a {
  color: #1a72cb;
}
```

40-login\public\signup.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Registro</title>
  <link rel="stylesheet" href="main.css" />
</head>
<body>
  <form action="/register" method="POST">
    <h2>Registro</h2>
    <div class="section">
      <div class="title">Username</div>
      <div class="field">
        <input type="text" id="username" name="username" />
      </div>
    </div>
    <div class="section">
      <div class="title">Password</div>
      <div class="field">
        <input type="password" id="password" name="password" />
      </div>
    </div>
    <div class="section">
      <div class="button">
        <input type="submit" value="Registrar usuário" />
      </div>
      <div class="button">
        <a href="index.html">Login</a>
      </div>
    </div>
  </form>
</body>
</html>
```

- Rode o servidor do Mongo DB:

`mongod`

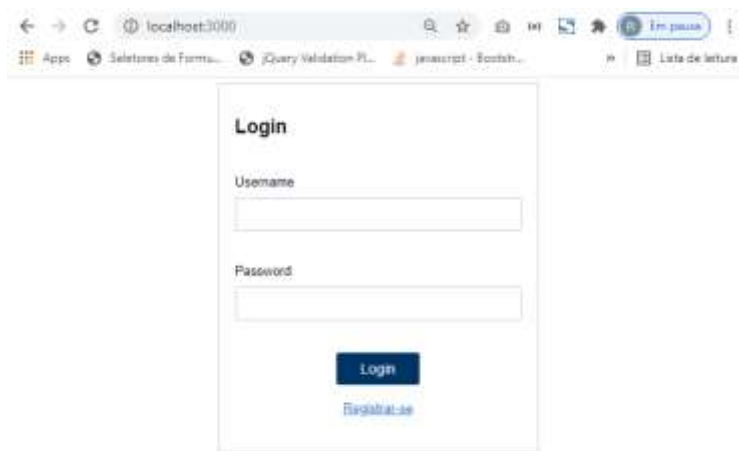
- No terminal do Visual Studio Code, entre com:

`nodemon app.js`

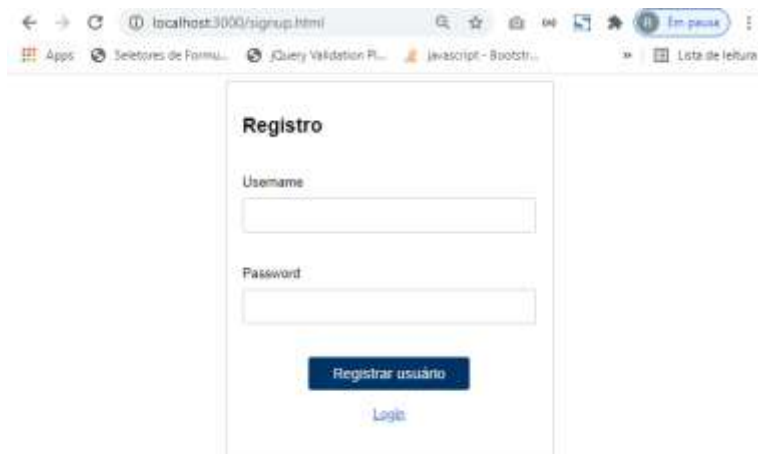
```
C:\marcos_rivas\basico\40-login>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node app.js'
(node:14368) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version.
To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:14368) [MONGODB DRIVER] Warning: Current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Server started...
Successfully connected to mongodb://dev:dev@localhost/todos
```

- No browser:

localhost:3000

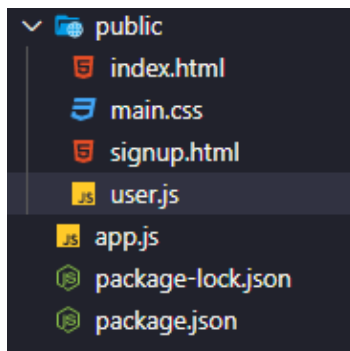


localhost:3000/signup.html



Criando o Model de usuários

- Adicione a pasta "public" um arquivo chamado "user.js"



40-login\public\user.js

```
const mongoose = require('mongoose');
const bcrypt = require('bcrypt');

const saltRounds = 10;

const UserSchema = new mongoose.Schema({
  username: {
    type: String,
    require: true,
    unique: true
  },
  password: {
    type: String,
    require: true
  }
});

UserSchema.pre('save', function(next){
  if(this.isNew || this.isModified('password')){
    const document = this;

    bcrypt.hash(document.password, saltRounds, (err, hashedPassword) => {
      if(err) {
        next(err);
      } else {
        document.password = hashedPassword;
        next();
      }
    });
  } else {
    next();
  }
});

UserSchema.methods.isCorrectPassword = function(password, callback) {
  bcrypt.compare(password, this.password, function(err, same){
    if(err){
      callback(err);
    } else {
      callback(err, same);
    }
  });
}

module.exports = mongoose.model('User', UserSchema);
```

40-login\app.js

```
const express = require('express');
const path = require('path');
const bodyParser = require('body-parser');
const app = express();
```

```

const mongoose = require('mongoose');
const User = require('./public/user');

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));

app.use(express.static( path.join(__dirname, 'public')));

const mongo_uri = 'mongodb://dev:dev@localhost/todos';

mongoose.connect(mongo_uri, function(err) {
  if(err) {
    throw err;
  } else {
    console.log(`Successfully connected to ${mongo_uri}`);
  }
})

app.post('/register', (req, res) => {
  const {username, password} = req.body;
  const user = new User({username, password});
  user.save(err => {
    if(err){
      res.status(500).send('Erro ao registrar o usuário');
    } else {
      res.status(200).send('Usuário registrado com sucesso!');
    }
  });
});

app.post('/authenticate', (req, res) => {
  const {username, password} = req.body;

  User.findOne({username}, (err, user) => {
    if(err){
      res.status(500).send('Erro ao autenticar o usuário!');
    } else if(!user) {
      res.status(500).send('O usuário não existe!');
    } else {
      user.isCorrectPassword(password, (err, result) => {
        if(err){
          res.status(500).send('Erro ao autenticar o usuário!');
        } else if(result) {
          res.status(200).send('Usuário autenticado com sucesso!');
        } else {
          res.status(500).send('Usuário e/ou senha incorreta!');
        }
      });
    }
  });
});

```

```
app.listen(3000, () => {  
  console.log('Server started...');  
})
```

```
module.exports = app;
```

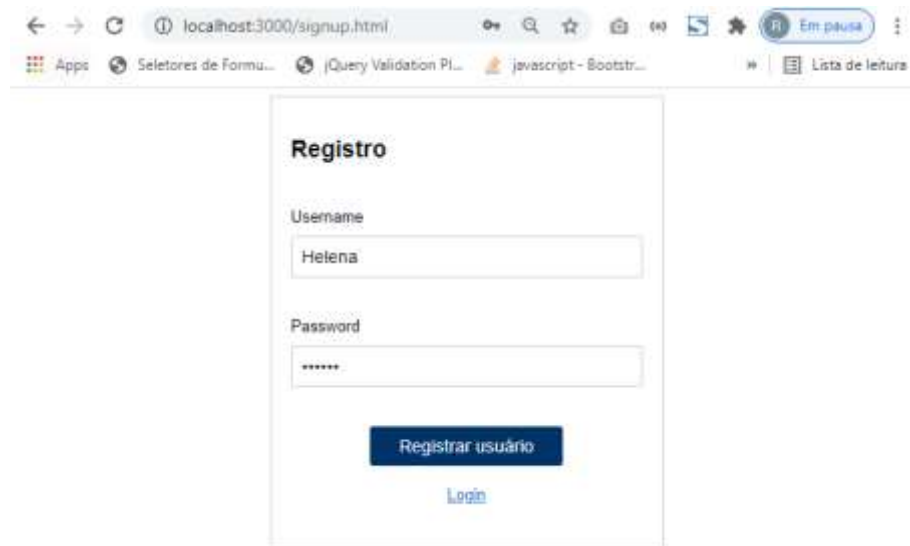
nodemon app.js

```
C:\marcos_rivas\basico\40-login>nodemon app.js  
[nodemon] 2.0.7  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: js,mjs,json  
[nodemon] starting `node app.js`  
(node:7432) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version.  
To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.  
(Use `node --trace-deprecation ...` to show where the warning was created)  
(node:7432) [MONGODB DRIVER] Warning: Current Server Discovery and Monitoring engine is deprecated, and will be re  
moved in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology:  
true } to the MongoClient constructor.  
Server started...  
(node:7432) DeprecationWarning: collection.ensureIndex is deprecated. Use createIndexes instead.  
Successfully connected to mongodb://dev:dev@localhost/todos  
█
```

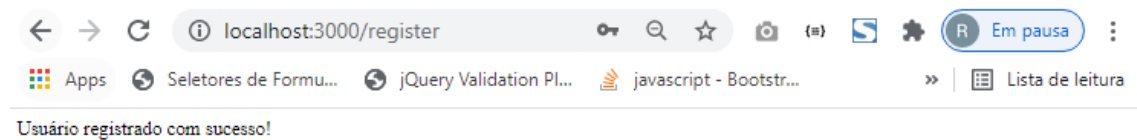
- No browser:

Senha: 123456

<http://localhost:3000/signup.html>



The screenshot shows a web browser window with the address bar displaying `localhost:3000/signup.html`. The page contains a registration form titled "Registro". The form has two input fields: "Username" with the value "Helena" and "Password" with masked characters "*****". Below the password field is a blue button labeled "Registrar usuário". At the bottom of the form, there is a blue link labeled "Login".



The screenshot shows a web browser window with the address bar displaying `localhost:3000/register`. The page displays a success message: "Usuário registrado com sucesso!".

`mongo-express --admin --url mongodb://127.0.0.1:27017`



The screenshot shows the Mongo Express web interface in a browser window. The address bar displays `localhost:3081`. The interface has a header with "Mongo Express" and "Database -". Below the header, there is a section titled "Databases" with a table listing databases: "admin", "config", "local", "store-api", and "todos". Each row has a "View" button (green) and a "Del" button (red). At the bottom, there is a "Server Status" section with the text "Turn on admin in config.js to view server stats!".

| Databases | |
|----------------------|-----------|
| View | admin |
| View | config |
| View | local |
| View | store-api |
| View | todos |

Server Status
Turn on admin in config.js to view server stats!

← → ↻ localhost:8081/db/todos/ 📄 🔍 ☆ 🖨️ 📱 🌐 Em pausa ⋮

Apps Seletores de Formu... jQuery Validation Pl... javascript - Bootstr... » Lista de leitura

Mongo Express Database: todos +

Viewing Database: todos

Collections + Create collection

View

Export

JSON

Import

delete_me

Del

View

Export

JSON

Import

todos

Del

View

Export

JSON

Import

users

Del

Database Stats

| | |
|---------------------------------------|------------|
| Collections (incl. system.namespaces) | 3 |
| Data Size | 480 Bytes |
| Storage Size | 51.4 KB |
| Avg Obj Size # | 80.0 Bytes |
| Objects # | 6 |
| Indexes # | 4 |
| Index Size | 51.9 KB |

← → ↻ localhost:8081/db/todos/users 📄 🔍 ☆ 🖨️ 📱 🌐 Em pausa ⋮

Apps Seletores de Formu... jQuery Validation Pl... javascript - Bootstr... » Lista de leitura

Mongo Express Database: todos + Collection: users +

Viewing Collection: users

New Document

New Index

Simple

Advanced

Key

Value

String

Find

Delete all 1 documents retrieved

| _id | username | password |
|---|----------|--|
| <div><div></div><div>60c55379f8276a31241fa491</div></div> | Helena | \$2b\$10SL9MfS2oEmYalH178qOK8OmbUo2nVuzEXhPZfYkSQ... |

- No browser:

Usuário não existe:

localhost:3000

← → ↻ ⓘ localhost:3000 🔑 🔍 ☆ 📷 ⓘ S 🛠 R Em pausa ⋮

Apps Seletores de Formu... jQuery Validation PL... javascript - Bootstr... » Lista de leitura

Login

Username

Password

Login

[Registrar-se](#)

← → ↻ ⓘ localhost:3000/authenticate 🔑 🔍 ☆ 📷 ⓘ S 🛠 R Em pausa ⋮

Apps Seletores de Formu... jQuery Validation PL... javascript - Bootstr... » Lista de leitura

O usuário não existe!

Senha errada:

← → ↻ ⓘ localhost:3000 🔑 🔍 ☆ 📷 ⓘ S 🛠 R Em pausa ⋮

Apps Seletores de Formu... jQuery Validation PL... javascript - Bootstr... » Lista de leitura

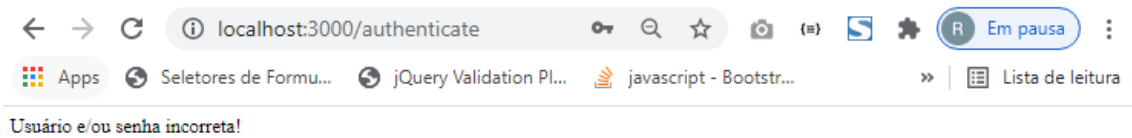
Login

Username

Password

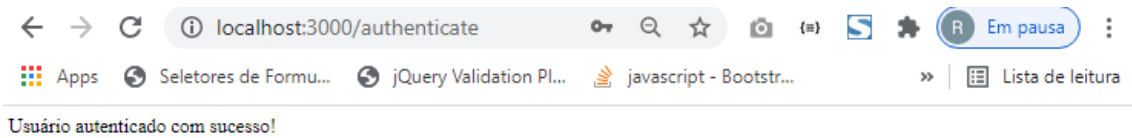
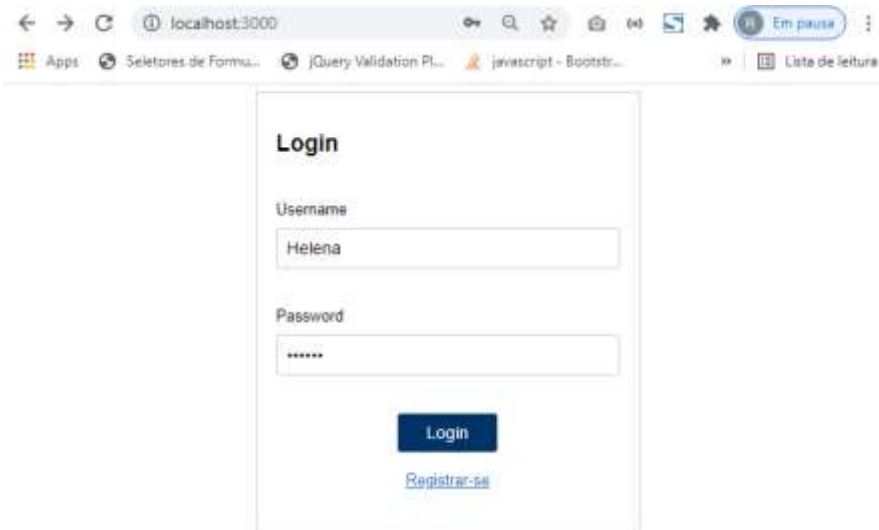
Login

[Registrar-se](#)



username e password corretos:

Senha: 123456



Aula 41 - Como usar variáveis de ambiente com dotenv

No terminal, entre com o comando:

node

process.env

```
C:\marcos_rivas\basico\41-env>node
Welcome to Node.js v14.15.3.
Type ".help" for more information.
> process.env
{
  ALLUSERSPROFILE: 'C:\\ProgramData',
  APPDATA: 'C:\\Users\\betol\\AppData\\Roaming',
  CHROME_CRASHPAD_PIPE_NAME: '\\\\.\\pipe\\crashpad_12028_FXMRGKNOQEQRFLCQ',
  CommonProgramFiles: 'C:\\Program Files\\Common Files',
  'CommonProgramFiles(x86)': 'C:\\Program Files (x86)\\Common Files',
  CommonProgramW6432: 'C:\\Program Files\\Common Files',
  COMPUTERNAME: 'DESKTOP-NUIKR9E',
  ComSpec: 'C:\\Windows\\system32\\cmd.exe',
  HOMEDRIVE: 'C:',
  HOMEPATH: '\\\\Users\\betol',
  LOCALAPPDATA: 'C:\\Users\\betol\\AppData\\Local',
  LOGONSERVER: '\\\\DESKTOP-NUIKR9E',
  NUMBER_OF_PROCESSORS: '2',
  OneDrive: 'C:\\Users\\betol\\OneDrive',
  ORIGINAL_XDG_CURRENT_DESKTOP: 'undefined',
  OS: 'Windows NT',
  Path: 'C:\\Windows\\system32;C:\\Windows;C:\\Windows\\System32\\Wbem;C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\;C:\\Program Files\\nodejs\\;C:\\Program Files\\Git\\cmd;C:\\Users\\betol\\AppData\\Local\\Microsoft\\WindowsApps;C:\\Program Files\\heroku\\bin;C:\\Program Files\\JetBrains\\PhpStorm 2020.3.1\\bin;C:\\Users\\betol\\AppData\\Roaming\\npm;C:\\Program Files\\MongoDB\\Server\\4.4\\bin;C:\\Users\\betol\\AppData\\Local\\Programs\\Microsoft VS Code\\bin',
  PATHEXT: '.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC',
  PhpStorm: 'C:\\Program Files\\JetBrains\\PhpStorm 2020.3.1\\bin;',
  PROCESSOR_ARCHITECTURE: 'AMD64',
  PROCESSOR_IDENTIFIER: 'Intel64 Family 6 Model 42 Stepping 7, GenuineIntel',
  PROCESSOR_LEVEL: '6',
  PROCESSOR_REVISION: '2a07',
  ProgramData: 'C:\\ProgramData',
  ProgramFiles: 'C:\\Program Files',
  'ProgramFiles(x86)': 'C:\\Program Files (x86)',
  ProgramW6432: 'C:\\Program Files',
  PROMPT: '$PS$G',
  PSMODULEPATH: 'C:\\Program Files\\WindowsPowerShell\\Modules;C:\\Windows\\system32\\WindowsPowerShell\\v1.0\\Modules',
  PUBLIC: 'C:\\Users\\Public',
  SESSIONNAME: 'Console',
  SystemDrive: 'C:',
  SystemRoot: 'C:\\Windows',
  TEMP: 'C:\\Users\\betol\\AppData\\Local\\Temp',
  TMP: 'C:\\Users\\betol\\AppData\\Local\\Temp',
  USERDOMAIN: 'DESKTOP-NUIKR9E',
  USERDOMAIN_ROAMINGPROFILE: 'DESKTOP-NUIKR9E',
  USERNAME: 'betol',
  USERPROFILE: 'C:\\Users\\betol',
  windir: 'C:\\Windows',
  TERM_PROGRAM: 'vscode',
  TERM_PROGRAM_VERSION: '1.57.0',
  LANG: 'pt_BR.UTF-8',
  COLORTERM: 'truecolor',
  VSCODE_GIT_IPC_HANDLE: '\\\\.\\pipe\\vscode-git-5a0c41d603-sock',
  GIT_ASKPASS: 'c:\\Users\\betol\\AppData\\Local\\Programs\\Microsoft VS Code\\resources\\app\\extensions\\git\\dist\\askpass.sh',
  VSCODE_GIT_ASKPASS_NODE: 'C:\\Users\\betol\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe',
  VSCODE_GIT_ASKPASS_MAIN: 'c:\\Users\\betol\\AppData\\Local\\Programs\\Microsoft VS Code\\resources\\app\\extensions\\git\\dist\\askpass-main.js'
}
```


Usos:

- Informação sensível (exemplo: senhas);
- Informação de configuração;
- Tokens, conexões e ports

npm init -y

```
C:\marcos_rivas\basico\41-env>npm init -y
Wrote to C:\marcos_rivas\basico\41-env\package.json:

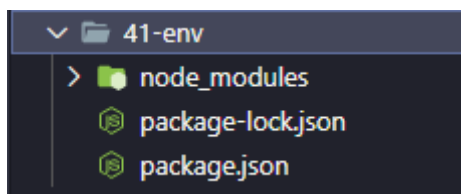
{
  "name": "41-env",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Instalando dotenv

npm install dotenv -D

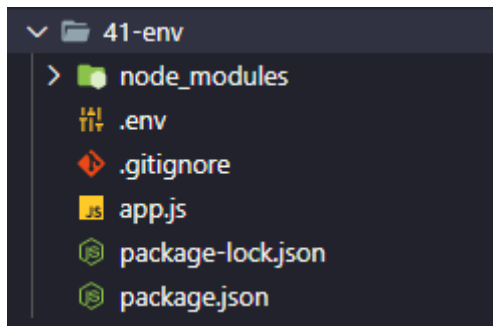
```
C:\marcos_rivas\basico\41-env>npm install dotenv -D
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN 41-env@1.0.0 No description
npm WARN 41-env@1.0.0 No repository field.

+ dotenv@10.0.0
added 1 package and audited 1 package in 3.254s
found 0 vulnerabilities
```



41-env\package.json

```
{
  "name": "41-env",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "dotenv": "^10.0.0"
  }
}
```



41-env\app.js

```
if(process.env.NODE_ENV !== 'production'){
  require('dotenv').config();
}

const app = require('http').createServer((req, res) => res.send('Hello World!'));
const PORT = process.env.PORT || 3000;

console.log(process.env.TOKEN);

app.listen(PORT, () => {
  console.log(`Server is listening on port ${PORT}`);
});
```

41-env\.env

```
PASSWORD=esteehmeupassword
TOKEN=23232VB567M5N3N452M3J43
PORT=6700
```

nodemon app.js

```
C:\marcos_rivas\basico\41-env>nodemon app.js
[nodemon] 2.0.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
23232VB567M5N3N452M3J43
Server is listening on port 6700
█
```

41-env\.gitignore

.env

Aula 42 - Autenticação com JWT (JSON Web Token)

npm init -y

```
C:\marcos_rivas\basico\42-jwt>npm init -y
Wrote to C:\marcos_rivas\basico\42-jwt\package.json:

{
  "name": "42-jwt",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Instalando as dependências

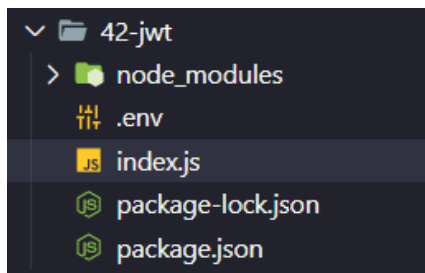
npm install express dotenv jsonwebtoken

```
C:\marcos_rivas\basico\42-jwt>npm install express dotenv jsonwebtoken
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN 42-jwt@1.0.0 No description
npm WARN 42-jwt@1.0.0 No repository field.

+ dotenv@10.0.0
+ express@4.17.1
+ jsonwebtoken@8.5.1
added 65 packages from 46 contributors and audited 65 packages in 16.693s
found 0 vulnerabilities
```

42-jwt\package.json

```
{
  "name": "42-jwt",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "dotenv": "^10.0.0",
    "express": "^4.17.1",
    "jsonwebtoken": "^8.5.1"
  }
}
```



42-jwt\index.js

```
const express = require('express');
const app = express();
const jwt = require('jsonwebtoken');
require('dotenv').config();

app.use(express.urlencoded({extended: false}));
app.use(express.json());

app.get('/', (req, res) => {
  res.send('Hello World!');
})

app.get('/login', (req, res) => {
  res.send(`<html>
    <head>
      <title>Login</title>
    </head>
    <body>
      <form method="POST" action="/auth">
        Nome de usuário: <input type="text" name="text"><br />
        Senha: <input type="password" name="password"><br />
        <input type="submit" value="Iniciar sessão"/>
      </form>
    </body>
  </html>`);
});

app.get('/api', validateToken, (req, res) => {
  res.json({
    tuits: [
      {
        id: 1,
        text: 'Este é meu primeiro tuit',
        username: 'vidamrr'
      },
      {
        id: 2,
        text: 'A melhor linguagem é HTML',
        username: 'patinho_feliz'
      }
    ]
  });
});
```

```

app.post('/auth', (req, res) => {
  const {username, password} = req.body;

  // consultar BD e validar que existem tanto username como password
  const user = {username: username};
  const accessToken = generateAccessToken(user);

  res.header('authorization', accessToken).json({
    message: 'Usuário autenticado!',
    token: accessToken
  });
});

function generateAccessToken(user){
  return jwt.sign(user, process.env.SECRET, {expiresIn: '50m'});
}

function validateToken(req, res, next){
  const accessToken = req.headers['authorization'];

  if(!accessToken) res.send('Access denied');

  jwt.verify(accessToken, process.env.SECRET, (err, user) => {
    if(err){
      res.send('Access denied, token expired or incorrect');
    }else{
      next();
    }
  });
}

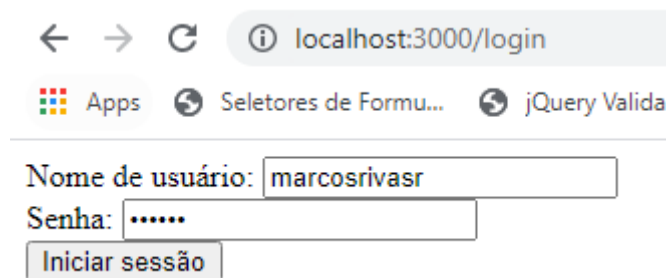
app.listen(3000, () => {
  console.log('Servidor iniciado...');
});

```

42-jwt\.env

SECRET=vidamrr

Senha: marcos



← → ↻ ⓘ localhost:3000/login

Apps Seletores de Formu... jQuery Valida

Nome de usuário:

Senha:

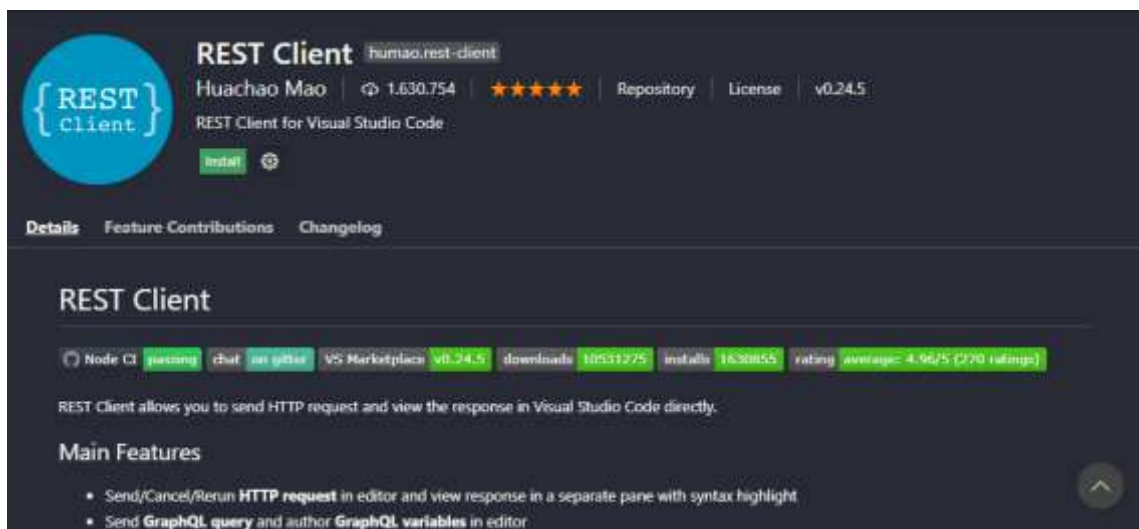
localhost:3000/auth



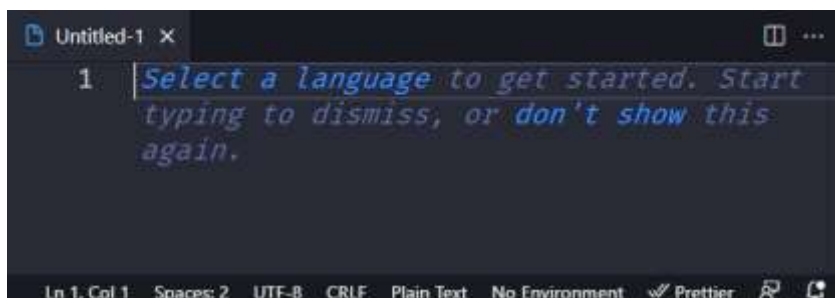
```
1 // 20210613164348
2 // http://localhost:3000/auth
3
4 {
5   "message": "Usuário autenticado!",
6   "token":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im1hcmNvc3JpdmFzciIsIm1hdCI6MTYyMzYwODU5MywiZXhwIjoxNjIzNjIyOTkzfkQ.RrSiYlkVfnpOXab9AbW9TLOSeTjHZoniyaoaqtSx010"
7 }
```

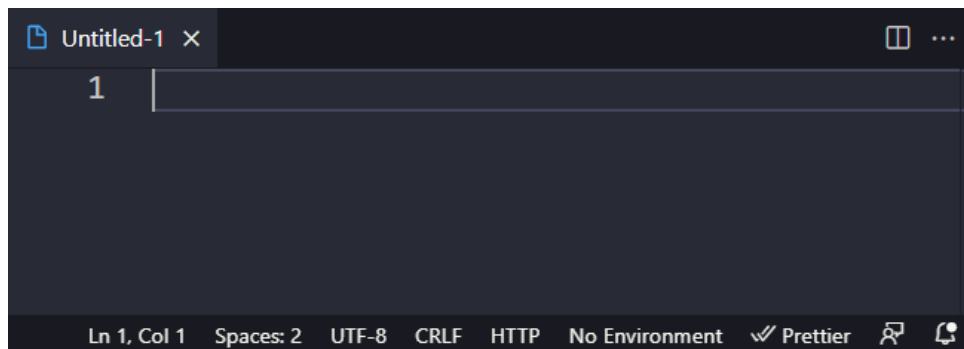
```
{
  "message": "Usuário autenticado!",
  "token":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im1hcmNvc3JpdmFzciIsIm1hdCI6MTYyMzYwODU5MywiZXhwIjoxNjIzNjIyOTkzfkQ.RrSiYlkVfnpOXab9AbW9TLOSeTjHZoniyaoaqtSx010"
}
```

- No Visual Studio Code, instale uma extensão chamada "REST Client":

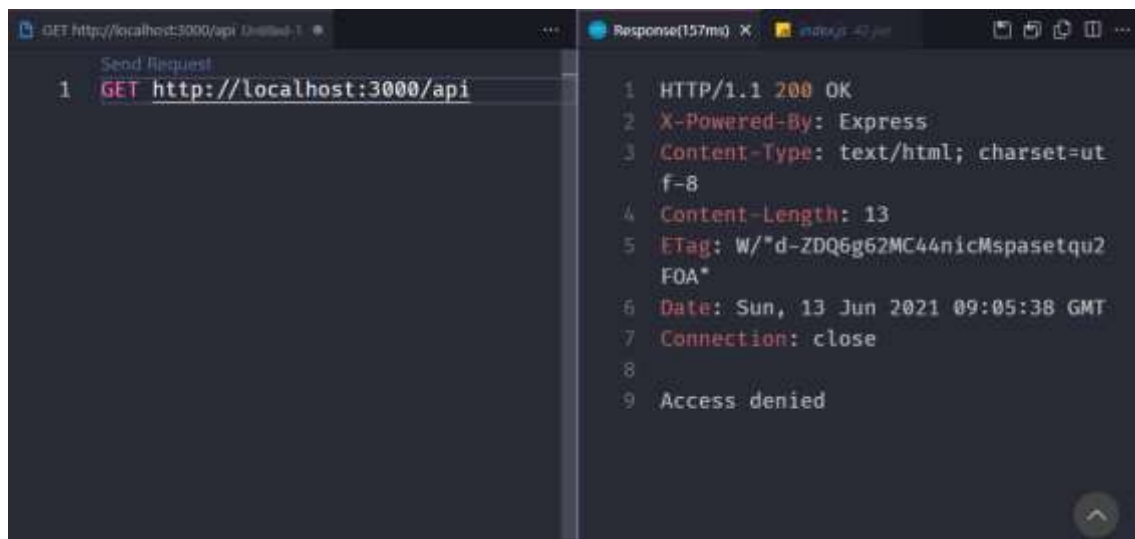
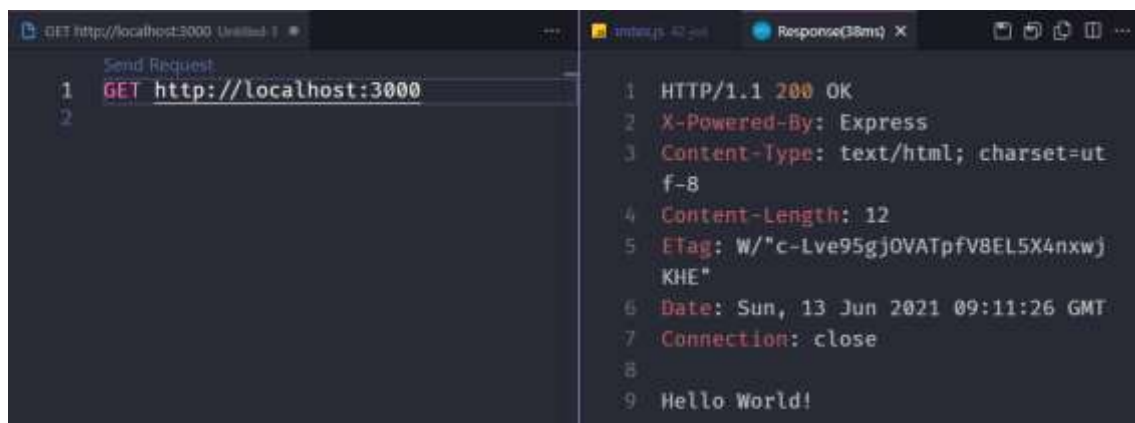


- No Visual Studio Code crie um novo arquivo e o defina como tipo "http":





- Entre com:



- Passando um token correto e não expirado:

```
Send Request
1 GET http://localhost:3000/api
2 Content-Type: application/json
3 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImthcmVudCJpdmFrciIsImlhdCI6MTYyMzYwODU5MywiZXN1IjoxNjIzNjIyOTkzfkQ.RrS1vLkVfnpOXab9Abw9TL0SeTjH2oniyaqaqtsXB10

Response (http)
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 148
5 ETag: W/"94-BZr88McwCent1JLRb15M7EJkhw"
6 Date: Sun, 13 Jun 2021 19:28:26 GMT
7 Connection: close
8
9 {
10   "tweets": [
11     {
12       "id": 1,
13       "text": "Este é meu primeiro tweet",
14       "username": "vidaeerr"
15     },
16     {
17       "id": 2,
18       "text": "A melhor linguagem é HTML",
19       "username": "patinho_feliz"
20     }
21   ]
22 }
```

Passando um token incorreto ou inválido:

```
Send Request
1 GET http://localhost:3000/api
2 Content-Type: application/json
3 Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImthcmVudCJpdmFrciIsImlhdCI6MTYyMzYwODU5MywiZXN1IjoxNjIzNjIyOTkzfkQ.RrS1vLkVfnpOXab9Abw9TL0SeTjH2oniyaqaqtsXB10

Response (http)
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 41
5 ETag: W/"29-UKw6rtgz89y4PbyNAjRoyIAJ0ig"
6 Date: Sun, 13 Jun 2021 19:23:55 GMT
7 Connection: close
8
9 Access denied, token expired or incorrect
```

42-jwt\index.js

```
const express = require('express');
const app = express();
const jwt = require('jsonwebtoken');
require('dotenv').config();
```

```
app.use(express.urlencoded({extended: false}));
app.use(express.json());
```

```
app.get('/', (req, res) => {
  res.send('Hello World!');
})
```

```
app.get('/login', (req, res) => {
  res.send(`<html>
    <head>
      <title>Login</title>
    </head>
    <body>
      <form method="POST" action="/auth">
        Nome de usuário: <input type="text" name="text"><br />
        Senha: <input type="password" name="password"><br />
        <input type="submit" value="Iniciar sessão"/>
      </form>
```

```

        </body>
    </html>`);
});

app.get('/api', validateToken, (req, res) => {
    res.json({
        tuits: [
            {
                id: 1,
                text: 'Este é meu primeiro tuit',
                username: 'vidamrr'
            },
            {
                id: 2,
                text: 'A melhor linguagem é HTML',
                username: 'patinho_feliz'
            }
        ]
    });
});

app.post('/auth', (req, res) => {
    const {username, password} = req.body;

    // consultar BD e validar que existem tanto username como password
    const user = {username: username};
    const accessToken = generateAccessToken(user);

    res.header('authorization', accessToken).json({
        message: 'Usuário autenticado!',
        token: accessToken
    });
});

function generateAccessToken(user){
    return jwt.sign(user, process.env.SECRET, {expiresIn: '50m'});
}

function validateToken(req, res, next){
    const accessToken = req.headers['authorization'] || req.query.accesstoken;

    if(!accessToken) res.send('Access denied');

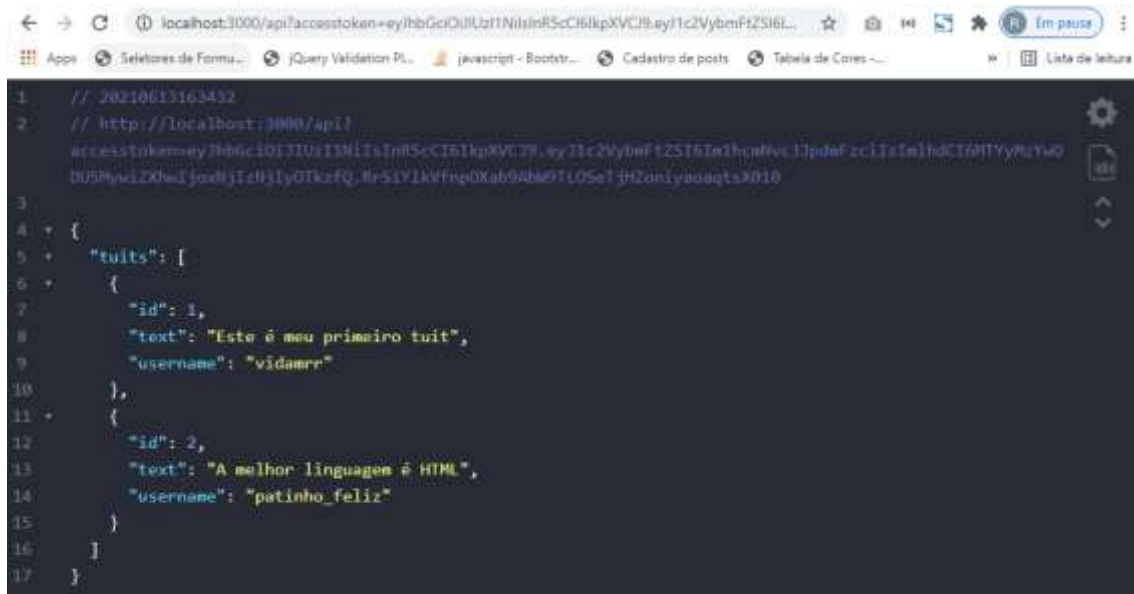
    jwt.verify(accessToken, process.env.SECRET, (err, user) => {
        if(err){
            res.send('Access denied, token expired or incorrect');
        }else{
            next();
        }
    });
}

app.listen(3000, () => {
    console.log('Servidor iniciado...');
});

```

- No browser:

<http://localhost:3000/api?accesstoken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im1hcmNvc3JpdmFzciIsImh0bCI6MTYyMzYwODU5MywiZXhwIjoxNjIzNjIyOTkzfkQ.RrSiYlkVfnpOXab9AbW9TLOSeTjHZoniyaoaqtSx010>



42-jwt\index.js

```
const express = require('express');
const app = express();
const jwt = require('jsonwebtoken');
require('dotenv').config();

app.use(express.urlencoded({extended: false}));
app.use(express.json());

app.get('/', (req, res) => {
  res.send('Hello World!');
})

app.get('/login', (req, res) => {
  res.send(`<html>
    <head>
      <title>Login</title>
    </head>
    <body>
      <form method="POST" action="/auth">
        Nome de usuário: <input type="text" name="text"><br />
        Senha: <input type="password" name="password"><br />
        <input type="submit" value="Iniciar sessão"/>
      </form>
    </body>
  </html>`);
});
```

```

app.get('/api', validateToken, (req, res) => {
  res.json({
    username: req.user,
    tuits: [
      {
        id: 1,
        text: 'Este é meu primeiro tuit',
        username: 'vidamrr'
      },
      {
        id: 2,
        text: 'A melhor linguagem é HTML',
        username: 'patinho_feliz'
      }
    ]
  });
});

app.post('/auth', (req, res) => {
  const {username, password} = req.body;

  // consultar BD e validar que existem tanto username como password
  const user = {username: username};
  const accessToken = generateAccessToken(user);

  res.header('authorization', accessToken).json({
    message: 'Usuário autenticado!',
    token: accessToken
  });
});

function generateAccessToken(user){
  return jwt.sign(user, process.env.SECRET, {expiresIn: '50m'});
}

function validateToken(req, res, next){
  const accessToken = req.headers['authorization'] || req.query.accesstoken;

  if(!accessToken) res.send('Access denied');

  jwt.verify(accessToken, process.env.SECRET, (err, user) => {
    if(err){
      res.send('Access denied, token expired or incorrect');
    }else{
      req.user = user;
      next();
    }
  });
}

app.listen(3000, () => {
  console.log('Servidor iniciado...');
});

```

The screenshot shows two panels in VS Code. The left panel displays a REST client request for a GET endpoint. The right panel shows the corresponding HTTP response, which includes headers and a JSON body containing user information and tweets.

```

GET http://localhost:3000/api
Content-Type: application/json
Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im1hcmludmVjLjpdnfzc1IiwiaWF0IjE6MTYyMzYwODU5MywiZXhwIjoxNjIzNjIyOTkzfQ.RrSjYLkvfnpDxah9AbW9TL0SeIJH7oniyaoaqtsXB1B

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 221
ETag: W/"dd-IF5RWXTAItoOduvE/Psd7NhxuJw"
Date: Sun, 13 Jun 2021 19:39:36 GMT
Connection: close

{
  "username": {
    "username": "marcosrivast",
    "iat": 1623688593,
    "exp": 1623622993
  },
  "tweets": [
    {
      "id": 1,
      "text": "Este é meu primeiro tuit.",
      "username": "vidamrr"
    },
    {
      "id": 2,
      "text": "A melhor linguagem é HTML",
      "username": "patinho_feliz"
    }
  ]
}

```