

Desenvolvimento web com Nodejs

Carolina Junqueira Ferreira



Nodejs

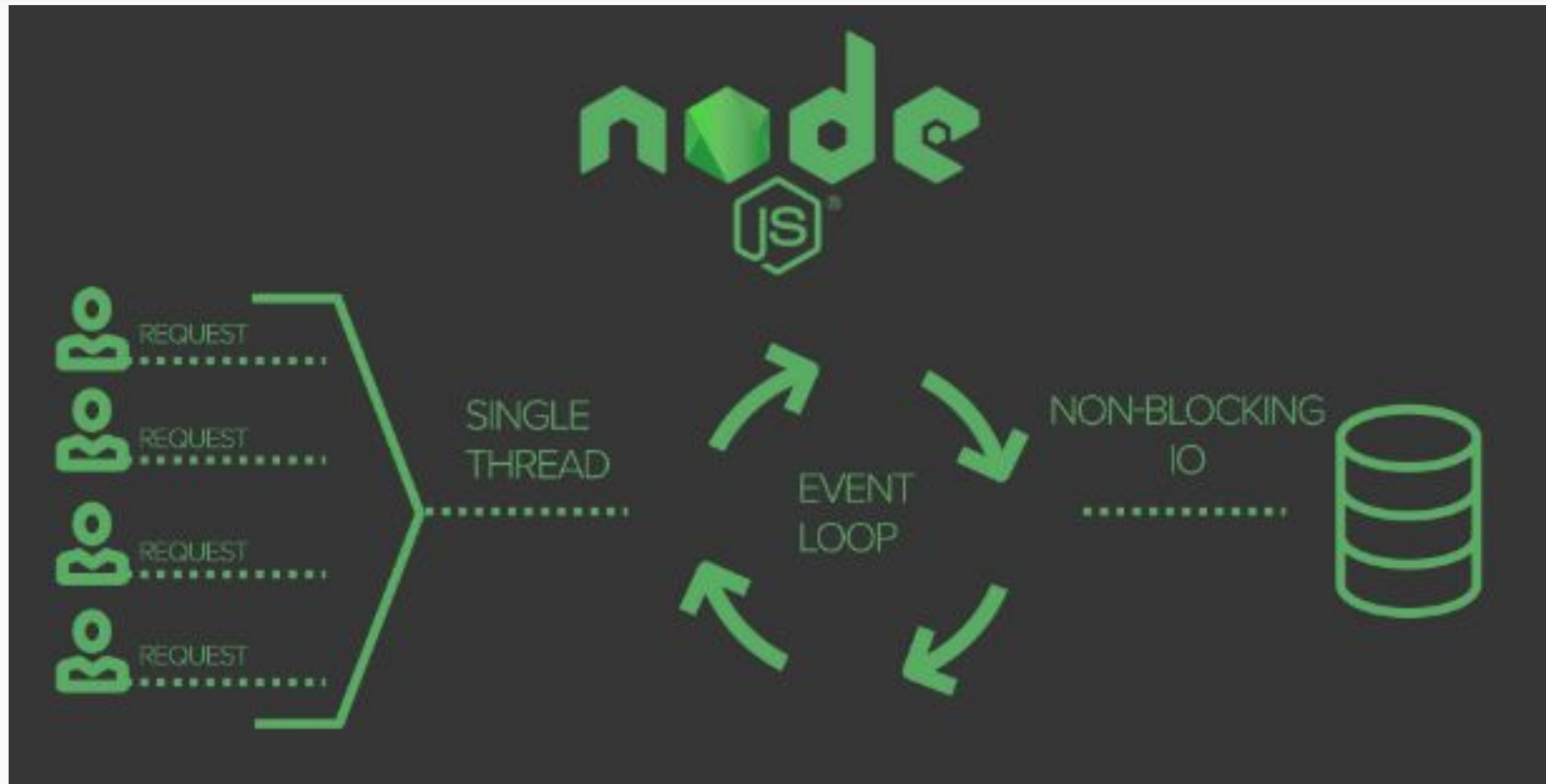
- Runtime Environment (v8 engine + módulos)
- Pacotes open source (npm)
- Event Driven
- Assíncrono - non-blocking I/O
- Escalabilidade
- Single Threaded - event queue



Arquitetura tradicional



Arquitetura do Nodejs



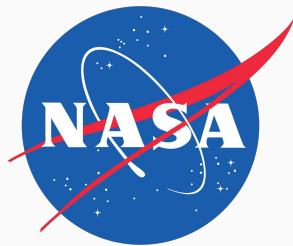
Recomendações

- Recomendado em aplicações tempo real e que tem intenso I/O (arquivos, rede, etc.)
- Não recomendado para aplicações com uso intenso de CPU (ex: manipulação de imagens ou vídeos)

Quem usa?

NETFLIX

 **PayPal**



ebay



Walmart 

NPM - Gerenciador de Pacotes

- Orientado a comunidade
- <https://www.npmjs.com/7>
- Exemplos:
- <https://www.npmjs.com/package/karaoke>
- <https://www.npmjs.com/package/pokedex>
- <https://www.npmjs.com/package/express>



Minicurso

- Mapa colaborativo de São Carlos
- MongoDB
- Leaflet - Open Street Maps
- Nodejs
- Express js
- EJS

Conferindo instalações

- Node instalado

```
→ node -v
```

- Npm

```
→ npm -v
```

- Editor de texto

Começando o projeto

- **Crie** uma pasta **mapa-sanca** em qualquer lugar que você **LEMBRE**
- Abra o terminal nessa pasta

```
→ cd path/dapasta/mapa-sanca
```

- Dentro dessa pasta dê o comando para criar um projeto node.

```
→ npm init
```

```
version: (1.0.0)  
description:  
entry point: (index.js) app.js
```

-
- Abrir pasta **mapa-sanca** no **editor** de texto

AVISOS

- A bolinha branca no canto significa que está disponível uma tag no github
- git clone <https://github.com/caroljung/mapa-sanca.git>
- npm install
- git tag 1-algum-nome, só olhar no git
- **Se der RUIM, me chama!!! Não tenha vergonha, é um minicurso, estamos aqui pra aprender**



Rodando scripts

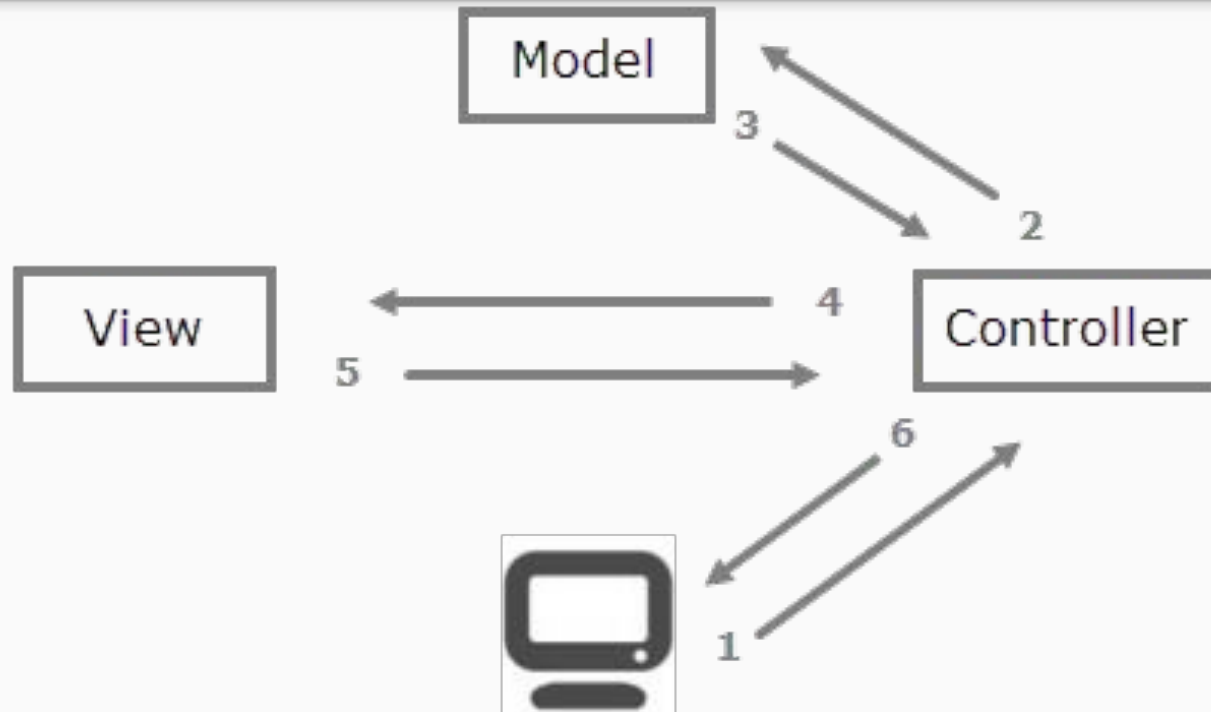
- **Crie** o arquivo **app.js** nessa **pasta**
- No arquivo escreva:

```
console.log("alguma coisa");
```

- No terminal:

```
→ node app.js
```

Arquitetura MVC



MVC no projeto

- **Criar** em mapa-sanca **pastas controllers, models, public, views**
- Apagar o conteúdo de app.js e salvar

Restful API

- Métodos do HTTP (**GET**, **POST**, DELETE, PUT, etc.)
- REST - Transferência de Estado Representacional
- Conjunto de princípios → métodos padronizados
- Mudar o estado de recursos
- URIs = ID Ex: <http://www.exemplo.com/empresas/123>
- Restful = ser capaz de aplicar o Rest

Expressjs

- Pacote NPM
- Recursos HTTP
- Instalar Expressjs no projeto:

→ `npm install --save express`

- Copiar código e testar

→ `node app.js`

- Entrar no browser **localhost:3000**

```
1  const express = require('express');
2  const app = express();
3
4  app.get('/', function(req, res){
5    res.send("Maoee Silvio!");
6  })
7
8  app.listen(3000, function(){
9    console.log("Ouvindo a porta 3000!");
10 })
```


EJS

- → npm install --save ejs
- Ferramenta didática: mantém HTML acrescenta JS
- Na pasta **views** criar arquivos:
 - map.ejs
 - register.ejs
- Copiar exemplo dentro dos arquivos e mudar texto

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="utf-8">
  <title>Mapa Sanca</title>
</head>
<body>
  MAPAMAPA
</body>
</html>
```

Novas rotas

- Abrir o **app.js**
- Modificar para código ao lado
- Dizer ao express que está usando a engine EJS
- Novas rotas
 - /register
- → node app.js
- **localhost:3000** e **localhost:3000/register**

```
1  const express = require('express');
2  const app = express();
3
4  app.set('view engine','ejs');
5
6  app.get('/', function(req,res){
7    res.render('map');
8  })
9
10 app.get('/register', function(req,res){
11   res.render('register');
12 })
13
14 app.listen(3000,function(){
15   console.log("Ouvindo a porta 3000!");
16 })
```

- Abra o arquivo **register.ejs** no editor de texto
- No **<body>**, coloque o código
- `→ nodejs app.js` e **localhost:3000/register**

```
<h2>Registro de novas localizações</h2>
<form action="/post-register" method="post">
  Nome do ponto: <input type="text" name="name"><br>
  Descrição: <input type="text" name="description"><br>
  Seu nome: <input type="text" name="author"><br>
  Latitude: <input type="text" name="lat"><br>
  Longitude: <input type="text" name="lng"><br>
  <input type="submit" value="Registrar">
</form>
```

- → npm install --save body-parser
- Em **app.js**
- Acrescentar mais código
- **localhost:3000/register**
- Preencha o formulário
- Clique em registrar
- **Observe terminal**

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

app.set('view engine', 'ejs');

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

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

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

app.post('/post-register', function(req, res){
  res.render('map');
  console.log(req.body);
})

app.listen(3000, function(){
  console.log("Ouvindo a porta 3000!");
})
```

- No arquivo **app.js**, vamos modificar as seguintes rotas:

```
app.get('/', function(req,res){  
  let obj = {  
    name: "teste",  
    description: "descricao",  
    author: "alguem",  
    lat: "123",  
    lng: "44"  
  };  
  res.render('map',obj);  
})
```

```
app.post('/post-register', function(req,res){  
  res.render('map',req.body);  
})
```

- Em **map.ejs**, no **<body>** colocar:

```
<div>Nome: <%= name %></div>
<div>Descrição: <%= description %></div>
<div>Autor: <%= author %></div>
<div>Latitude: <%= lat %></div>
<div>Longitude: <%= lng %></div>
```

- Testando:

- → node app.js

-

- localhost:3000/register ou localhost:3000

MongoDB

RDBMS		MongoDB
Database	→	Database
Table	→	Collection
Index	→	Index
Row	→	Document
Column	→	Field
Join	→	Embedding & Linking

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

Collection

- npm install --save mongoose
- Na pasta **raiz do projeto**, criar o arquivo **db-connection.js**

```
1  const mongoose = require("mongoose");
2
3  mongoose.Promise = global.Promise;
4
5  mongoose.connect("mongodb://aluno:secomp123@ds233212.mlab.com:33212/mapa-sanca", { useNewUrlParser: true }, function(err){
6    if(err){
7      console.log(err);
8    }
9  });
```

- No **topo** do arquivo **app.js**

const db = require('./db-connection.js');

```
const db = require("./db-connection.js");
const express = require('express');
const app = express();
const bodyParser = require('body-parser');
```


- Na pasta **models**, criar arquivo **coordinate.js** com o código

```
const mongoose = require("mongoose");

let coordinateSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  description: {
    type: String,
    required: true
  },
  author: {
    type: String,
    required: true
  },

```

```
    lat: {
      type: Number,
      required: true
    },
    lng: {
      type: Number,
      required: true
    }
  }, { collection: 'coordinates' });

module.exports = mongoose.model('CoordinateSchema', coordinateSchema)
```

- Na pasta **controllers**, criar arquivo **map-controller.js**.

```
1  const Coordinate = require('../models/coordinate');
2
3  function all(){
4    return Coordinate.find();
5  }
6
7  function save(coordinate){
8    let new_coordinate = new Coordinate(coordinate);
9    return new_coordinate.save();
10 }
11
12 module.exports.all = all;
13 module.exports.save = save;
```

- No arquivo **app.js**, acrescente:

const mapCtrl = require("../controllers/map-controller");

- Arrumar a **rota "/" de get** com o seguinte código → → → →

- Testando:

→ node app.js

- Conecte em **localhost:3000**
- Observe o **terminal**

```
app.get('/', function(req,res){
  let obj = {
    name: "teste",
    description: "descricao",
    author: "alguem",
    lat: "123",
    lng: "44"
  };
  mapCtrl.all()
    .then((coordinates) => {
      console.log(coordinates)
    })
    .catch((err) =>{
      res.send("Banco de dados deu problema");
    });
  res.render('map',obj);
})
```

- Abra o **map.ejs**
- **Entre** em <https://leafletjs.com/examples/quick-start/>
- Copiar dentro da tag **<head>**

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.4/dist/leaflet.css"
  integrity="sha512-puBpdR07980ZvTTbP4A8Ix/l+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWAKuyeQUyT07+7N4QKrDh+drA=="
  crossorigin="" />
```

- **Apagar o que tem dentro do body** e colocar no lugar:
<div id="mapid" style="height: 500px; width: 100%"></div>
- No final do **<body>**, copiar:

```
<script src="https://unpkg.com/leaflet@1.3.4/dist/leaflet.js"
  integrity="sha512-nMMmRyTVoLYqjP9hrbed9S+FzjZHW5gY1TWCHA5ckwXZBadntCNS8kEqAWdrb907rxbCaA4lKTIWjDXZxf10cA=="
  crossorigin=""></script>
```

- Abra o **map.ejs**
- No **<body>**, depois da **última** tag **<script>**, acrescente:

```
<script>
  let mymap = L.map("mapid").setView([-22.0087, -47.8909], 13);
  L.tileLayer("https://tile.openstreetmap.org/{z}/{x}/{y}.png",
  {
    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a>',
  }).addTo(mymap)
</script>
```

- Testando:

→ node app.js

- **localhost:3000**

- Abra o arquivo **app.js**
- Na rota **get "/"**, a mudança fica:

```
app.get('/', function(req,res){
  mapCtrl.all()
    .then((coordinates) => {
      res.render('map',{results: coordinates});
    })
    .catch((err) =>{
      res.send("Banco de dados deu problema");
    });
})
```

- No arquivo **map.ejs**
- Acrescentar na última tag **<script>**

```
let coordinates = <%-JSON.stringify(results)%>;
for(let i = 0; i < coordinates.length; i++){
  let text = "<h3>" + coordinates[i].name + "</h3><p>Descrição: " + coordinates[i].description +
  "</p><p>Autor: " + coordinates[i].author+ "</p>";
  L.marker([coordinates[i].lat,coordinates[i].lng]).bindTooltip(text,{direction: 'top'}).addTo(mymap);
}
```

- → node app.js
- **localhost:3000**

- Em **app.js**, na rota **/post-register**

```
26  app.post('/post-register', function(req,res){
27    mapCtrl.save(req.body)
28    .then(() => {
29      res.redirect("/");
30    })
31    .catch((err) =>{
32      res.send("Deu ruim ao registrar!!");
33    });
34  })
```

- → node app.js
- **localhost:3000/register** e cadastre um novo ponto DIFERENTE ex: -15.23 e -20.99

- Abra o **register.ejs**
- **Copie** esses códigos de **map.ejs**
- Copiar dentro da tag **<head>**

```
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.4/dist/leaflet.css"
  integrity="sha512-puBpdR07980ZvTTbP4A8Ix/l+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWAKuyeQUyT07+7N4QKrDh+drA=="
  crossorigin="" />
```

- **Depois** da tag **</form>**

```
<div id="mapid2" style="height: 500px; width: 500px"></div>
```

- No final do **<body>**, copiar:

```
<script src="https://unpkg.com/leaflet@1.3.4/dist/leaflet.js"
  integrity="sha512-nMMmRyTVoLYqjP9hrbed9S+FzjZHW5gY1TWCHA5ckwXZBadntCNS8kEqAWdrb907rxbCaA4lKTIWjDXZxf10cA=="
  crossorigin=""></script>
```

No arquivo **register.ejs**

Adicione o

código **depois**

do último **<script>**

```
<script>
  let coord = {
    lat: 0,
    lng: 0
  }

  function copyCoord(){
    document.getElementById('latitude').value = coord.lat;
    document.getElementById('longitude').value = coord.lng;
  }

  let mymap2 = L.map("mapid2").setView([-22.0087,-47.8909],13);
  L.tileLayer("https://tile.openstreetmap.org/{z}/{x}/{y}.png", {
    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a>',
  }).addTo(mymap2)
  L.marker([-22.0087,-47.8909],{draggable: true}).addTo(mymap2).bindTooltip(
    'Arraste-me',{permanent: true, interactive: true, direction: 'top'})
    .on('move',function(event){
      let coordenadas = "<div>Lat: " + event.latlng.lat + "</div><div>Lng: " +
        event.latlng.lng + "</div><button onclick='copyCoord()'>Copiar</button>";
      coord.lat = event.latlng.lat;
      coord.lng = event.latlng.lng;
      this.bindTooltip(coordenadas);
    });
</script>
```

- No arquivo **register.ejs**.
- **Modifique** essas tags, colocando o **id="latitude"** e **id="longitude"**

```
Latitude: <input id="latitude" type="text" name="lat"><br>  
Longitude: <input id="longitude" type="text" name="lng"><br>
```

- Teste toda a aplicação agora que já sabe usar o node!

Projeto final

- github.com/caroljunq/mapa-sanca
- Controle de tags (1-10)
 - `git clone`
 - `npm install`
 - `git checkout nome-da-tag-que-vc-quer`
 - `node app`

Referências

- <https://nodejs.org/api/>
- <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>
- https://www.youtube.com/watch?v=TIB_eWDSMt4
- <https://www.npmjs.com/>
- <https://www.netguru.co/blog/top-companies-used-nodejs-production>
- <https://colorlib.com/wp/npm-packages-node-js/>
- <https://codeburst.io/25-node-js-tutorials-1db3b1da0260>
- <https://codeburst.io/build-a-weather-website-in-30-minutes-with-node-js-express-openweather-a317f904897b>
- <https://leafletjs.com/reference-1.3.4.html>
- <https://medium.com/dev-bits/writing-neat-asynchronous-node-js-code-with-promises-32ed3a4fd098>
- <https://howtofirebase.com/save-and-query-firebase-data-ed73fb8c6e3a>