



Quem se prepara, não para.

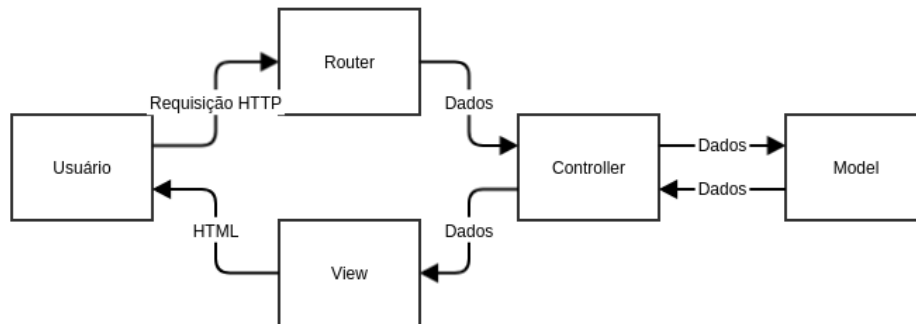
# Arquitetura de Sistemas

7º período

Professora: Michelle Hanne

# MVC em Node.JS

- O padrão MVC é simples na utilização, mas sua implementação no código pode ser um tanto complexa.
- Segue um exemplo simples e básico que utiliza do conceito MVC em Node.JS.



# MVC Vantagens

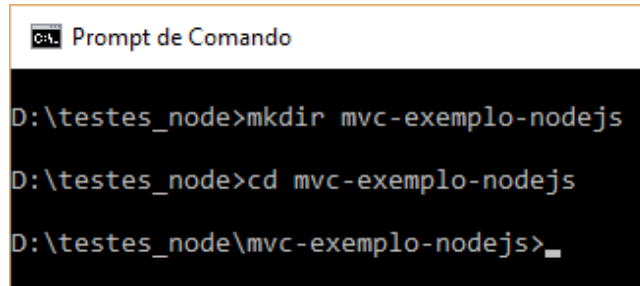
- **Reusabilidade de código:** uma funcionalidade já implementada em uma parte do sistema, pode ser novamente utilizada em outra parte, sem ser necessário o seu re-desenvolvimento.
  - **Exemplo:** em um sistema toda vez que um usuário altera sua senha, ele recebe um e-mail informando a alteração. Neste mesmo sistema quando ocorre uma venda, também é enviado um e-mail informando este usuário.
- **Desenvolvimento rápido:** neste modelo é possível que desenvolvedores e designers trabalhem simultaneamente na aplicação. Cada um trabalharia em uma camada diferente da aplicação, tornando o desenvolvimento produtivo.
- **Fácil manutenção:** como o código é dividido em partes distintas, se torna fácil adicionar novas funcionalidades e alterar características antigas. O código também fica mais fácil de ser compreendido por outros desenvolvedores.

# Exemplo - MVC em Node.JS

- A aplicação para exemplificação do MVC em Node.JS, consiste em uma tela onde serão listadas as últimas notícias fictícias.
- Será utilizado um arquivo JSON onde estarão salvas as notícias que serão exibidas.
- O código da aplicação está disponível no [GitHub: https://github.com/andersonirias/mvc-exemplo-nodejs](https://github.com/andersonirias/mvc-exemplo-nodejs).

# Exemplo - MVC em Node.JS

- **Passo 1:** Criar e acessar a pasta do Projeto (**mvc-exemplo-nodejs**)



```

C:\> Prompt de Comando

D:\testes_node>mkdir mvc-exemplo-nodejs

D:\testes_node>cd mvc-exemplo-nodejs

D:\testes_node\mvc-exemplo-nodejs>_

```

- **Passo 2:** Criar a estrutura de Arquivos e pastas da raiz do Projeto. **Pastas:** data e src. **Arquivos:** app.js e package.json

# Exemplo - MVC em Node.JS

```
D:\testes_node\mvc-exemplo-nodejs>mkdir data
D:\testes_node\mvc-exemplo-nodejs>mkdir src
D:\testes_node\mvc-exemplo-nodejs>type nul >app.js
D:\testes_node\mvc-exemplo-nodejs>npm init -y
Wrote to D:\testes_node\mvc-exemplo-nodejs\package.json:

{
  "name": "mvc-exemplo-nodejs",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

D:\testes_node\mvc-exemplo-nodejs>
```

# Exemplo - MVC em Node.JS

**Passo 3:** Acrescentar no arquivo **package.json** os metadados/pacotes necessários para o projeto

```

C:\> Prompt de Comando

D:\testes_node\mvc-exemplo-nodejs>npm i express consign ejs fs --save

> ejs@3.0.2 postinstall D:\testes_node\mvc-exemplo-nodejs\node_modules\ejs
> node --harmony ./postinstall.js

Thank you for installing EJS: built with the Jake JavaScript build tool (https://jakejs.com/)

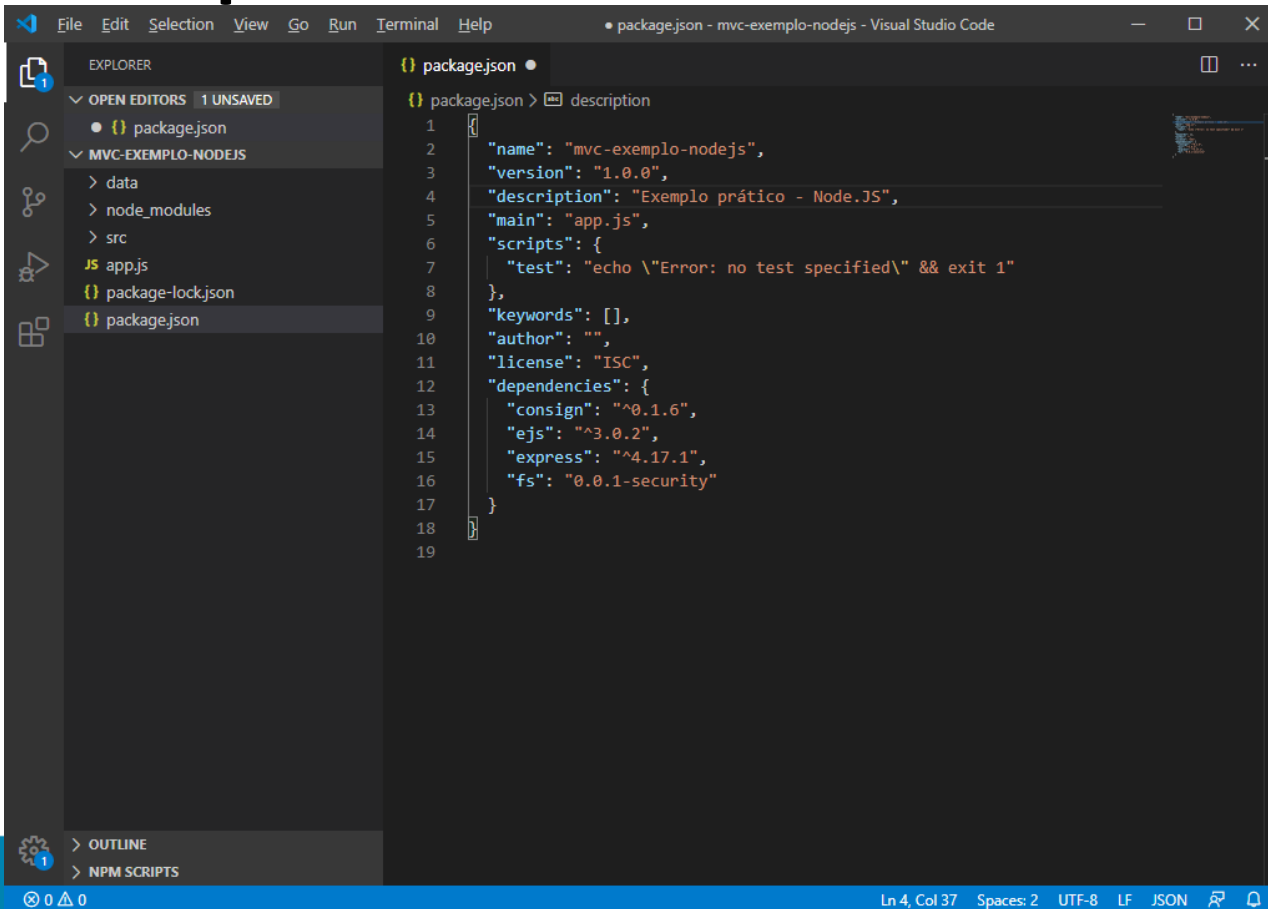
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN mvc-exemplo-nodejs@1.0.0 No description
npm WARN mvc-exemplo-nodejs@1.0.0 No repository field.

+ consign@0.1.6
+ ejs@3.0.2
+ fs@0.0.1-security
+ express@4.17.1
added 53 packages from 39 contributors and audited 129 packages in 7.997s
found 0 vulnerabilities

D:\testes_node\mvc-exemplo-nodejs>
```



# Exemplo - MVC em Node.JS



The screenshot shows the Visual Studio Code interface with the `package.json` file open. The Explorer on the left shows the project structure with folders `data`, `node_modules`, and `src`, and files `app.js`, `package-lock.json`, and `package.json`. The `package.json` file contains the following content:

```
{
  "name": "mvc-exemplo-nodejs",
  "version": "1.0.0",
  "description": "Exemplo prático - Node.JS",
  "main": "app.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "consign": "^0.1.6",
    "ejs": "^3.0.2",
    "express": "^4.17.1",
    "fs": "0.0.1-security"
  }
}
```

## Passo 4: Abrir o Visual Studio Code:

Conferir se o arquivo `package.json`, possui os metadados relevantes para o projeto.

# MVC em Node.JS

**Passo 5:** O arquivo **app.js** é o arquivo principal do projeto. Primeiramente incluímos as dependências do projeto. Depois configuramos o módulo **ejs**, que é para trabalharmos com as views. Após isso configuramos o módulo **consign**, que faz o carregamento automático dos scripts do nosso projeto. E por fim configuramos o servidor web do framework **express** para escutar requisições **http** na porta 3000.

```
JS app.js x
1  var express = require('express');
2  var consign = require('consign');
3
4  var app = express();
5  app.set('view engine', 'ejs');
6  app.set('views', './src/views');
7
8  consign()
9    .include('src/routes')
10   .then('src/models')
11   .then('src/controllers')
12   .into(app);
13
14  app.listen(3000, function(){
15    console.log('APP rodando na porta 3000');
16  });
17
```

# Exemplo - MVC em Node.JS

```
var express = require('express');
var consign = require('consign');

var app = express();
app.set('view engine', 'ejs');
app.set('views', './src/views');

consign()
  .include('src/routes')
  .then('src/models')
  .then('src/controllers')
  .into(app);

app.listen(3000, function(){
  console.log('APP rodando na porta 3000');
});
```

Abrir no **Visual Studio Code**

Copiar e colar o conteúdo do arquivo **app.js**.

# Exemplo - MVC em Node.JS

## Passo 6: Acessar a pasta **data** e criar o arquivo **noticias.json**

cmd Prompt de Comando

```
D:\testes_node\mvc-exemplo-nodejs>cd data
D:\testes_node\mvc-exemplo-nodejs\data>type nul > noticias.json
D:\testes_node\mvc-exemplo-nodejs\data>_
```

## Passo 7: Abrir no **Visual Studio Code** o arquivo **noticias.json**, copiar e colar o conteúdo.

```
{
  "noticias": [
    {
      "titulo": "Covid-19 29-03",
      "noticia": "O Ministério da Saúde confirmou 4.256 infecções e 136 mortes por coronavírus (Covid-19) no Brasil. Foram 353 novos casos nas últimas 24 horas, com uma taxa de letalidade de 3,2%."
    },
    {
      "titulo": "Coronavírus: fim do isolamento social não impede recessão econômica",
      "noticia": "Os presidentes do Brasil, Jair Bolsonaro, e dos Estados Unidos, Donald Trump, recentemente defenderam o fim do isolamento generalizado das populações e a retomada da vida normal para evitar uma crise maior."
    }
  ]
}
```

# Exemplo - MVC em Node.JS

**Passo 8:** Criar o arquivo **src/routes/news.js**, responsável por realizar o roteamento da aplicação. Nele falamos que ao se acessar a raiz da aplicação, com o método **GET** “<http://localhost:3000/>”. Será chamado o **controller news**.

**Siga os passos abaixo:**

```
C:\> Prompt de Comando

D:\testes_node\mvc-exemplo-nodejs>cd src

D:\testes_node\mvc-exemplo-nodejs\src>mkdir routes

D:\testes_node\mvc-exemplo-nodejs\src>cd routes

D:\testes_node\mvc-exemplo-nodejs\src\routes>type nul > news.js

D:\testes_node\mvc-exemplo-nodejs\src\routes>
```

OBS: para voltar ao diretório anterior o comando é **cd..**

# Exemplo - MVC em Node.JS

Abrir o **Visual Studio Code**, editar o arquivo **src/routes/news.js**, copiar e colar o conteúdo.

```
module.exports = function(application){  
  application.get('/', function(req, res){  
    application.src.controllers.news.index(application, req, res);  
  });  
}
```

# Exemplo - MVC em Node.JS

**Passo 9:** O arquivo **src/controllers/news.js** é o **controller** da aplicação. Primeiramente estamos criando um objeto do model news. Depois estamos utilizando a função **getLastNews** para recuperar as últimas 5 notícias do arquivo **noticias.json**. E por fim passamos as notícias para a view **news/index**. **Siga os passos:**

```

C:\> Prompt de Comando

D:\testes_node\mvc-exemplo-nodejs>cd src

D:\testes_node\mvc-exemplo-nodejs\src>mkdir controllers

D:\testes_node\mvc-exemplo-nodejs\src>cd controllers

D:\testes_node\mvc-exemplo-nodejs\src\controllers>type nul > news.js

D:\testes_node\mvc-exemplo-nodejs\src\controllers>_
```

# Exemplo - MVC em Node.JS

Abrir o **Visual Studio Code**, editar o arquivo `src/controllers/news.js`, copiar e colar o conteúdo.

```
module.exports.index = function(application, req, res) {  
    var newsModel = new application.src.models.news();  
  
    newsModel.getLastNews(function(err, result) {  
        res.render("news/index", {news : result});  
    });  
}
```



# Exemplo - MVC em Node.JS

**Passo 10:** O arquivo `src/models/news.js` é o model da aplicação. Primeiramente incluímos o módulo `fs` que auxiliará na leitura do arquivo JSON. Depois criamos a função `news` que vai funcionar como uma classe `news`. Após isto temos a função **`getLastNews`**, que realiza a leitura das notícias contidas no arquivo `noticias.json` e recupera somente as 5 ultimas, conforme a regra de negócio fictícia que foi estabelecida.

**Siga os passos:**

```

Prompt de Comando

D:\testes_node\mvc-exemplo-nodejs>cd src
D:\testes_node\mvc-exemplo-nodejs\src>mkdir models
D:\testes_node\mvc-exemplo-nodejs\src>cd models
D:\testes_node\mvc-exemplo-nodejs\src\models>type nul > news.js
D:\testes_node\mvc-exemplo-nodejs\src\models>_

```

# Exemplo - MVC em Node.JS

Abrir o **Visual Studio Code**, editar o arquivo **src/models/news.js**, copiar e colar o conteúdo.

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

function news() {}

news.prototype.getLastNews = function(callback) {
  fs.readFile('./data/noticias.json', 'utf8', function(err, result) {
    var data = [];

    if (!err) {
      var obj = JSON.parse(result);


      if (obj.noticias.length > 4) {
        var i = 4;
      } else {
        var i = (obj.noticias.length - 1);
      }

      obj.noticias.forEach(function(noticia) {
        if (i >= 0) {
          data[i] = noticia;
          i--;
        }
      });
    }
    callback(err, data);
  });
};

module.exports = function(){
  return news;
}
```

# Exemplo - MVC em Node.JS

**Passo 11:** O arquivo `src/views/news/index.ejs` é a view da aplicação. Ela é um arquivo **HTML** simples, onde pegamos as notícias enviadas pelo controller e realizamos uma iteração sobre elas, para exibir todas na tela inicial.

 Prompt de Comando

```
D:\testes_node\mvc-exemplo-nodejs>cd src
D:\testes_node\mvc-exemplo-nodejs\src>mkdir views
D:\testes_node\mvc-exemplo-nodejs\src>cd views
D:\testes_node\mvc-exemplo-nodejs\src\views>mkdir news
D:\testes_node\mvc-exemplo-nodejs\src\views>cd news
D:\testes_node\mvc-exemplo-nodejs\src\views\news>type nul > index.ejs
D:\testes_node\mvc-exemplo-nodejs\src\views\news>
```


# Exemplo - MVC em Node.JS

Abrir o **Visual Studio Code**, editar o arquivo `src/views/news/index.ejs`, copiar e colar o conteúdo.

```
<!doctype html>
<html lang="pt">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>MVC Exemplo Node.JS</title>
    <link href="https://getbootstrap.com/docs/4.0/dist/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <br/><br/><br/>
      <h2>MVC Exemplo Node.JS</h2>
      <br/>
      <% if (news.length > 0) { %>
        <div class="row">
          <% for(var i = 0; i < news.length; i++) { %>
            <div class="col-md-12 jumbotron">
              <h4><%= news[i].titulo %></h4>
              <p><%= news[i].noticia %></p>
            </div>
          <% } %>
        </div>
      <% } %>
    </div>
  </body>
</html>
```

# Exemplo - MVC em Node.JS

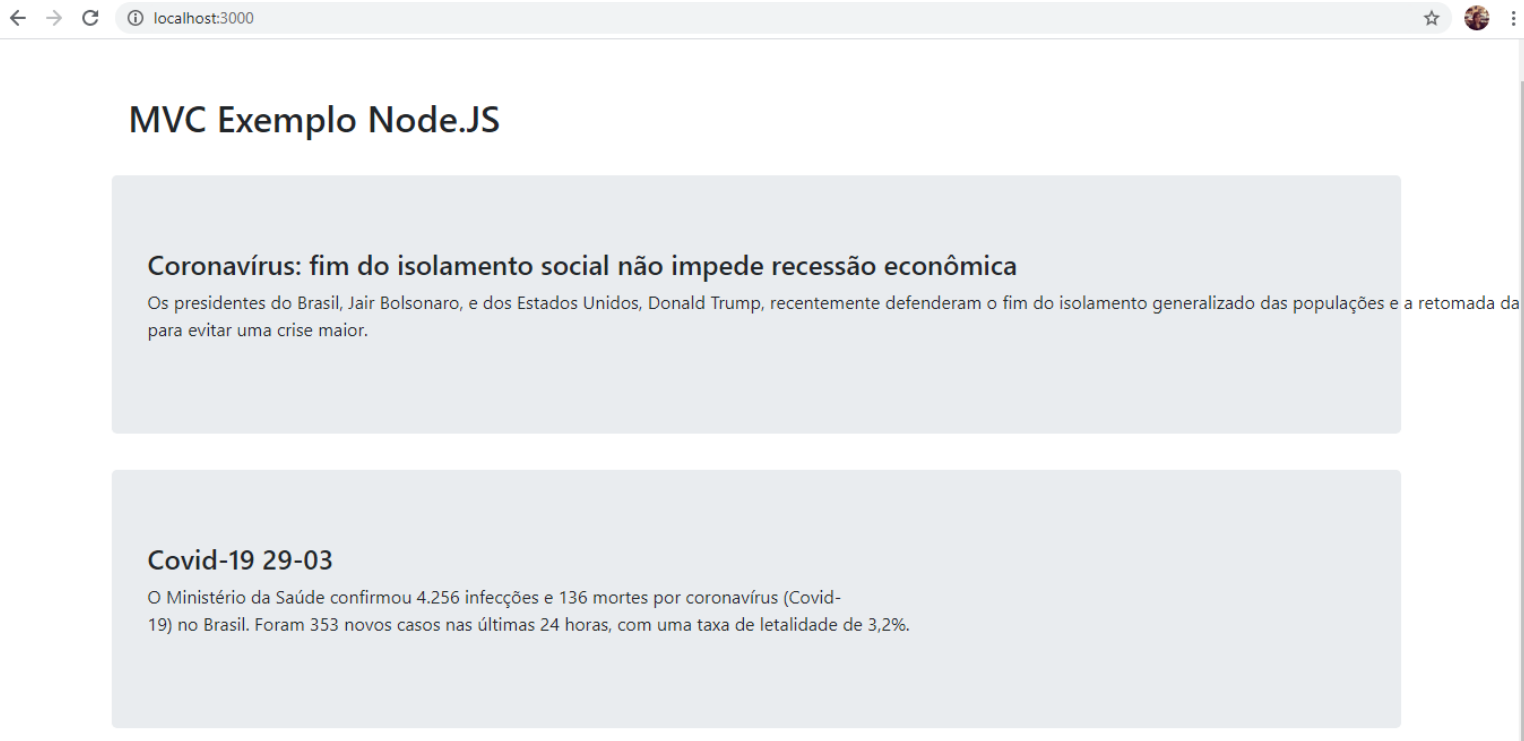
**Passo 12:** Testando a aplicação. Siga os passos:

 Prompt de Comando - node app.js

```
D:\testes_node\mvc-exemplo-nodejs>node app.js
consign v0.1.6 Initialized in D:\testes_node\mvc-exemplo-nodejs
+ .\src\routes\news.js
+ .\src\models\news.js
+ .\src\controllers\news.js
APP rodando na porta 3000
```

**Acesse o navegador para testar: localhost:3000**

# Exemplo - MVC em Node.JS



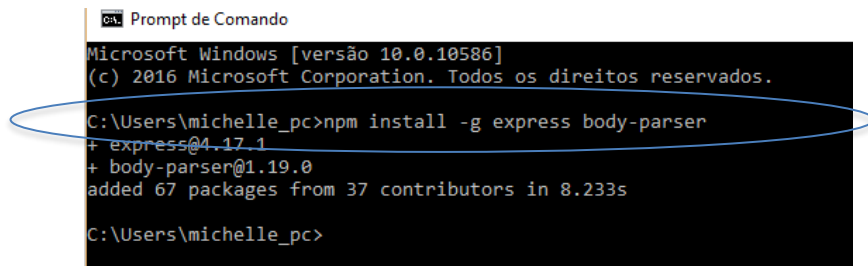
# Criando uma API Node com Express.js

O Express permite criar aplicações web simples e outras aplicações.

**Passo 1:** Para instalar acesse o terminal e digite:

**npm install -g express body-parser**

O comando **npm install -g** instala o Express globalmente no seu sistema, acrescentar dependências com o comando **body-parser**.



```
cmd Prompt de Comando
Microsoft Windows [versão 10.0.10586]
(c) 2016 Microsoft Corporation. Todos os direitos reservados.

C:\Users\michelle_pc>npm install -g express body-parser
+ express@4.17.1
+ body-parser@1.19.0
added 67 packages from 37 contributors in 8.233s

C:\Users\michelle_pc>
```

# Criando uma API Node com Express.js

## Passo 2: Criando a pasta da API

Ainda no terminal, crie um diretório com o nome da sua aplicação e vá até o diretório criado.

```
C:\> Prompt de Comando

D:\testes_node>mkdir api-node

D:\testes_node>cd api-node

D:\testes_node\api-node>_
```



# Criando uma API Node com Express.js

## Passo 2: Criando a pasta da API

Ainda no terminal, crie um diretório com o nome da sua aplicação e vá até o diretório criado.

```
C:\> Prompt de Comando

D:\testes_node>mkdir api-node

D:\testes_node>cd api-node

D:\testes_node\api-node>_
```

# Criando uma API Node com Express.js

**Passo 3:** Baixe o módulo do node na sua pasta. Cria uma estrutura de diretórios dentro de node\_modules.

```
npm install express
```

**Passo 4:** Criar o arquivo de configurações package.json

```
D:\testes_node\api-node>npm init -y
Wrote to D:\testes_node\api-node\package.json:

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

# Criando uma API Node com Express.js

## Passo 5: Criando o arquivo **app.js**

Agora, dentro do diretório **api-node**, abra o seu editor de textos preferido (ex. Visual Studio e crie um arquivo chamado **app.js**. Vamos precisar usar o pacote do Express que foi instalado anteriormente para poder criar a nossa API. Vejamos como vai ficar o nosso arquivo **app.js**:

# Criando uma API Node com Express.js

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

app.use(bodyParser.json());

app.get('/maps', function(req, res) {
  var dados = [
    {
      lat: -25.470991,
      lon: -49.271036
    },
    {
      lat: -0.935586,
      lon: -49.635540
    },
    {
      lat: -2.485874,
      lon: -43.128493
    }
  ];

  res.send(JSON.stringify(dados));
});

app.listen(8000, function() {
  console.log('Servidor rodando na porta 8000.');
```

Importa os pacotes instalados.

body-parser será o responsável por parsear as requisições via JSON.

configura o Express e atribui à variável app a ele

configura a forma como iremos tratar nossas requisições no Express.

rotear a URL /maps via método GET:

Quando o usuário navegar até a rota /maps, nosso servidor vai retornar um array de objetos contendo informações com latitude e longitude.

configurar uma porta onde nosso servidor web ficará escutando as chamadas HTTP.

# Criando uma API Node com Express.js

## Passo 6: Executar o arquivo app.js e testar no navegador

```
D:\testes_node\api-node>node app.js  
Servidor rodando na porta 8000.
```

<http://localhost:8000/maps>

← → ↻ ⓘ localhost:8000/maps

[{"lat":-25.470991,"lon":-49.271036},{"lat":-0.935586,"lon":-49.63554},{"lat":-2.485874,"lon":-43.128493}]