



Prof. Esp. Jeferson
Roberto de Lima

DESENVOLVIMENTO WEB II

TÓPICOS ANTERIORES



Variáveis e operações matemáticas;



Sistema condicional;



Módulos;



Criação de um servidor web;

EXPRESS

- O **Express.js** é um **Framework rápido** e um dos mais utilizados em conjunto com o **Node.js**, facilitando no **desenvolvimento de aplicações back-end** e até, em conjunto com sistemas de templates, **aplicações full-stack**.
- **Escrito em JavaScript**, o Express.js é utilizado por diversas empresas ao redor do mundo, dentre elas a Fox Sports, PayPal, IBM, Uber, entre outras.

The logo for Express.js, featuring the word "express" in a lowercase, white, sans-serif font, centered within a dark gray circle.

NPM E EXPRESS

- Para utilizarmos o express, precisamos instalar o **npm** em nossa máquina, para verificarmos a instalação podemos acessar o terminal e digitar o seguinte comando:
 - **npm -v**
- O terminal deve retornar a **versão instalada do npm** em seu computador.

```
C:\Users\f46337\projeto web>npm -v  
9.5.0
```

INSTALAÇÃO DO EXPRESS

- Para **instalarmos o framework express** devemos acessar o caminho do projeto no terminal e digitar o seguinte comando:
 - **npm install express --save**

```
C:\Users\f46337\projetoweb>npm install express --save
added 57 packages in 4s

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

CRIAR UM SERVIDOR WEB

Visual Studio Code interface showing the setup of a web server using Express.js.

EXPLORER

- PROJETOWEB
 - node_modules
 - app.js**
 - package-lock.json
 - package.json

app.js

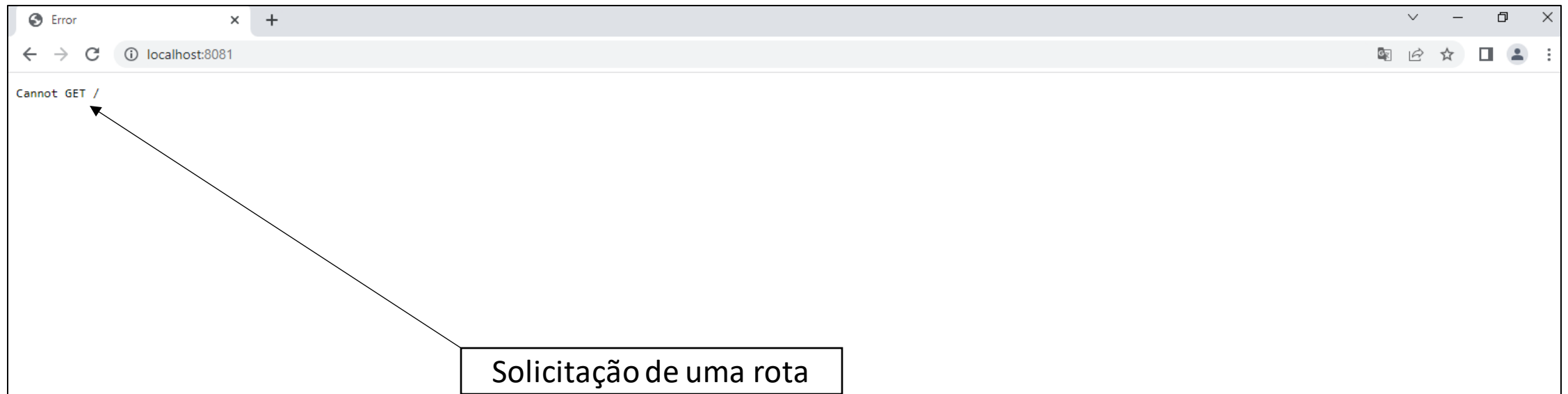
```
1 const express = require("express")
2 const app = express()
3
4 app.listen(8081)
5
```

Annotations:

- npm install express --save
- Importação do módulo do express
- Criando o servidor web com a porta 8081

RODAR UM SERVIDOR WEB

- Para rodar um **servidor web** devemos acessar o diretório do nosso projeto no **terminal** e digitar o seguinte comando:
 - **node app.js**
- Na sequência acesse o endereço **http://localhost:8081/** através do seu navegador.

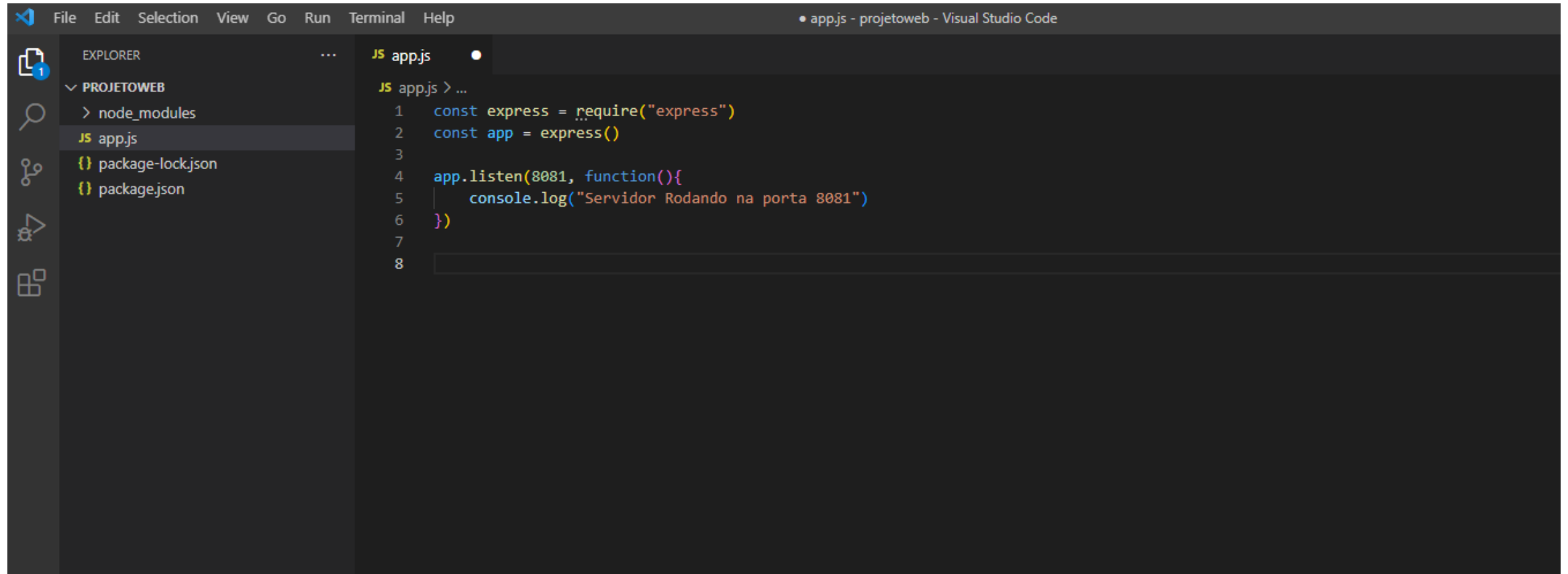


RESPOSTA DO NAVEGADOR

ROTAS

- A maioria dos frameworks utilizam o conceito de **rotas para redirecionar os conteúdos de uma página web**, uma rota define o nome e o endereço contendo as informações necessárias para a visualização, exemplo:
 - `http://localhost:8081/index`
 - `http://localhost:8081/produtos`
 - `http://localhost:8081/contato`
- Para criarmos as nossas rotas utilizaremos o conceito de **callback**.

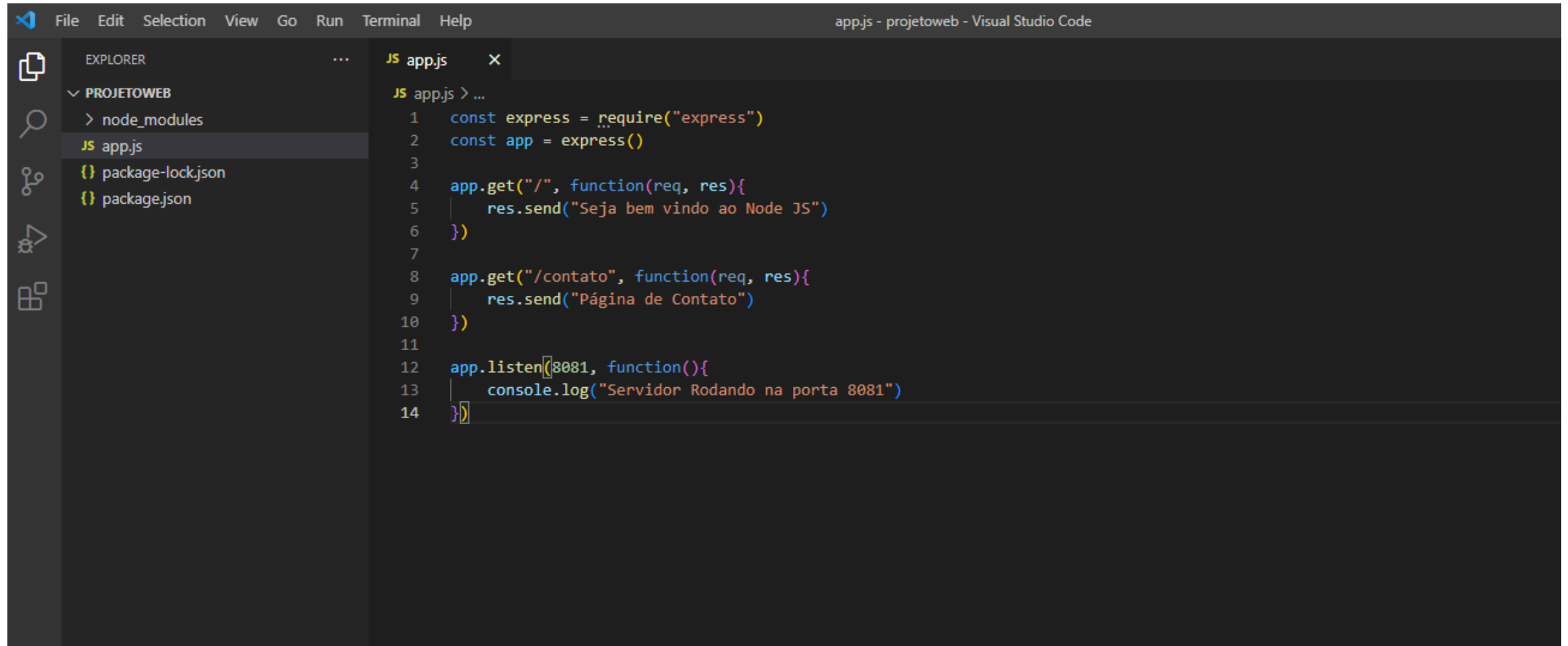
CALLBACK



The image shows a screenshot of the Visual Studio Code editor interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The Explorer sidebar on the left shows a project named 'PROJETOWEB' with a folder 'node_modules' and files 'app.js', 'package-lock.json', and 'package.json'. The main editor area displays the content of 'app.js', which is a JavaScript file. The code defines an Express application, listens on port 8081, and logs a message to the console when the server starts. The code is as follows:

```
JS app.js > ...
1  const express = require("express")
2  const app = express()
3
4  app.listen(8081, function(){
5    |   console.log("Servidor Rodando na porta 8081")
6  | })
7
8
```

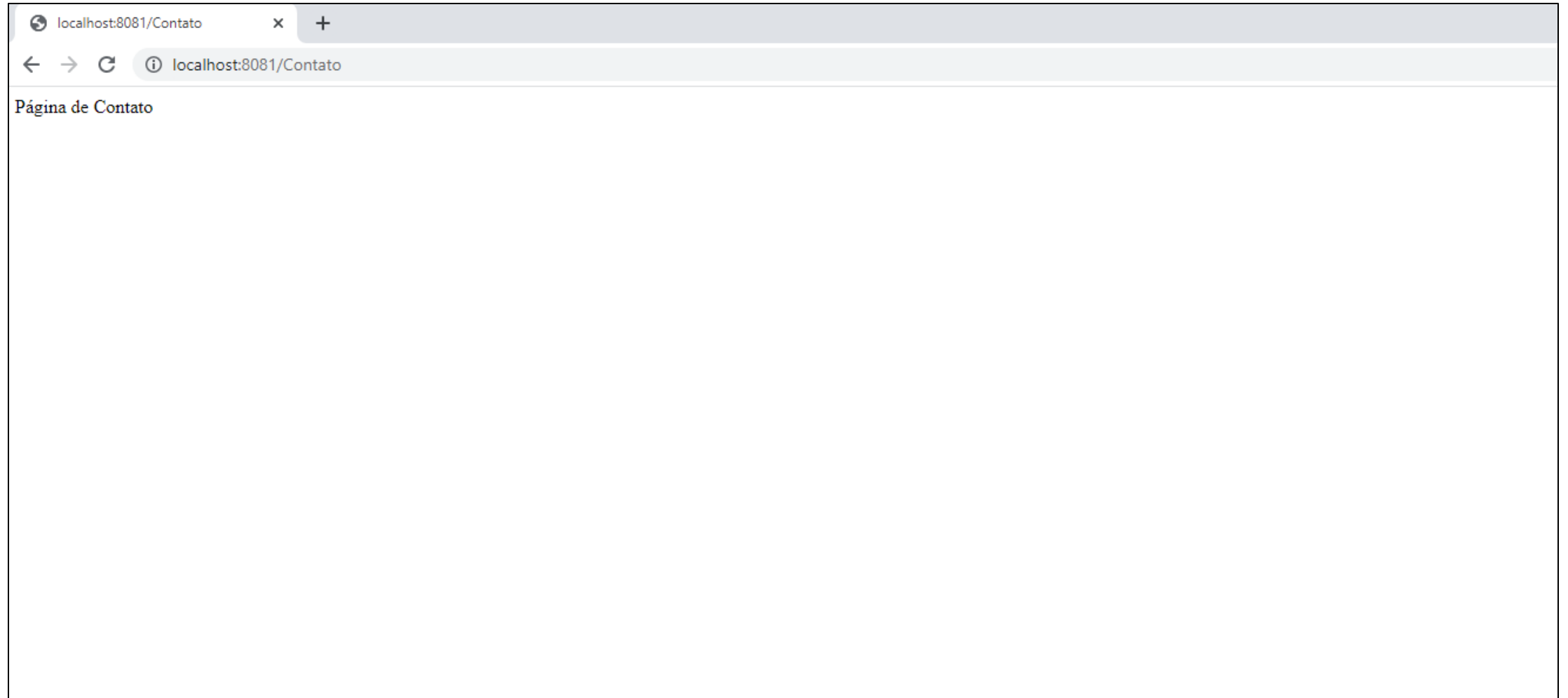
CRIANDO ROTAS



The image shows a screenshot of the Visual Studio Code editor interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The Explorer sidebar on the left shows a project named 'PROJETOWEB' with a subdirectory 'node_modules' and files 'package-lock.json' and 'package.json'. The main editor area displays the file 'app.js' with the following JavaScript code:

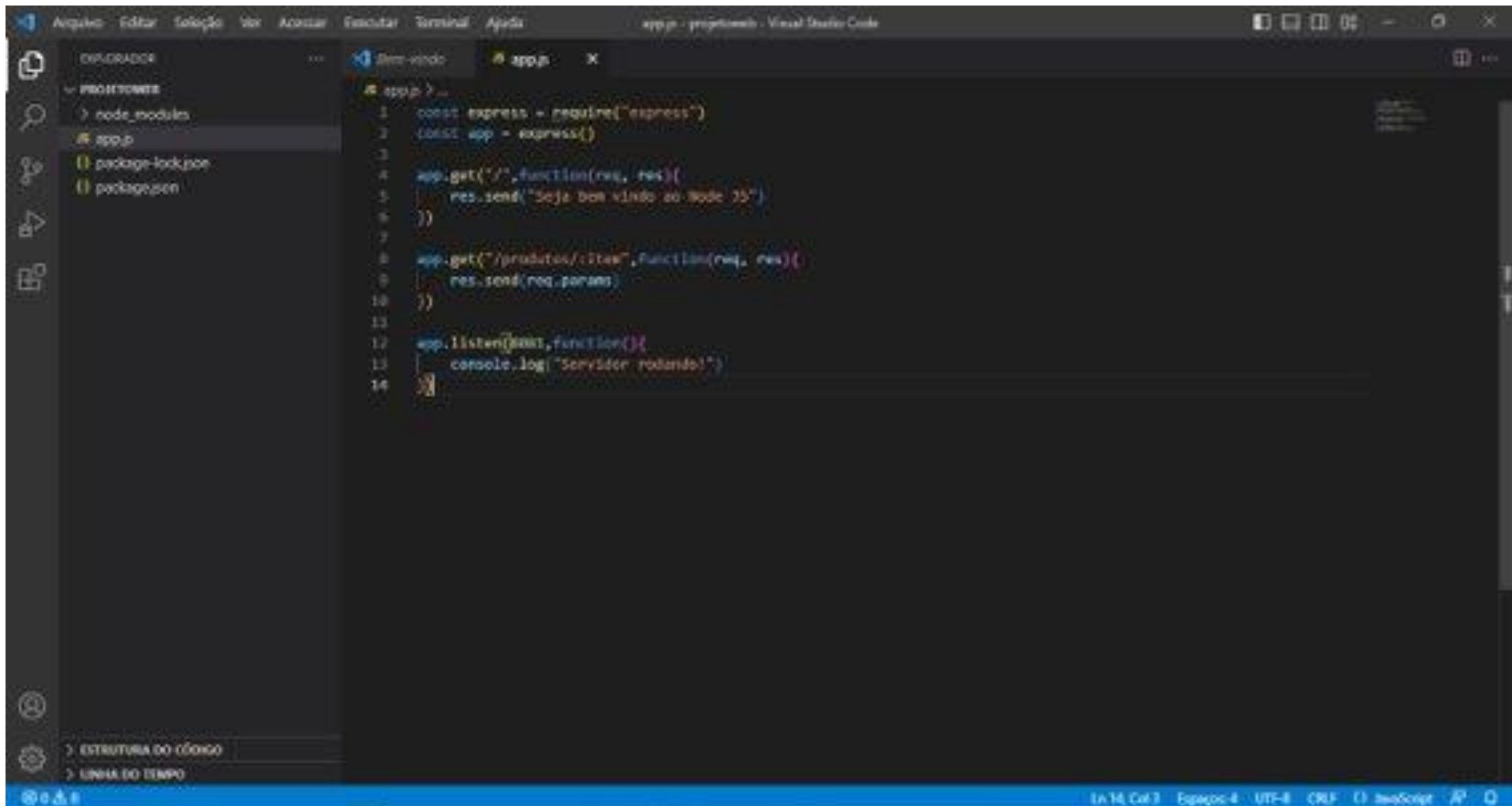
```
JS app.js > ...
1  const express = require("express")
2  const app = express()
3
4  app.get("/", function(req, res){
5    |   res.send("Seja bem vindo ao Node JS")
6  | })
7
8  app.get("/contato", function(req, res){
9    |   res.send("Página de Contato")
10 | })
11
12 app.listen(8081, function(){
13 |   console.log("Servidor Rodando na porta 8081")
14 | })
```

ROTAS PRONTAS



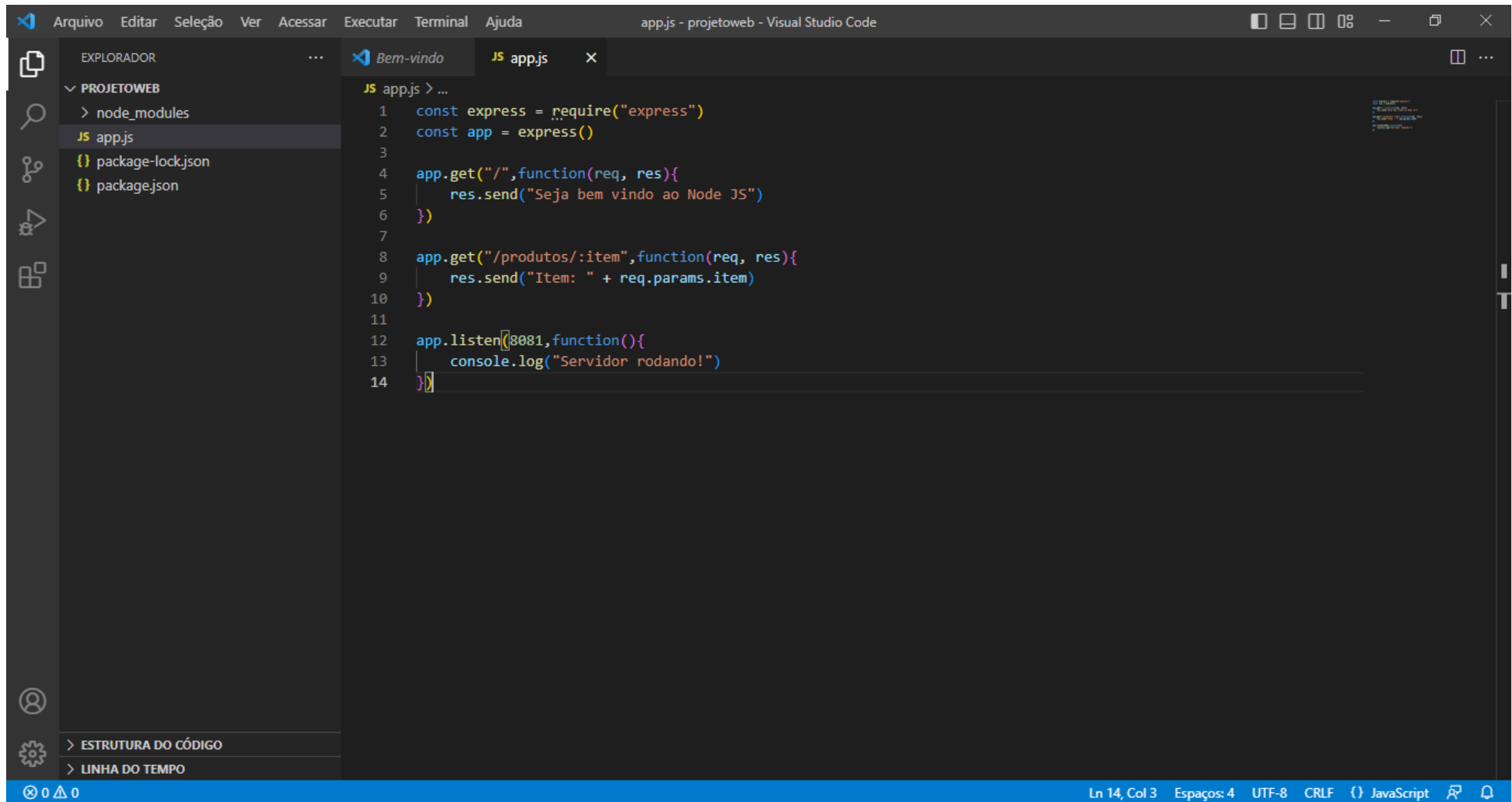
RECEBENDO PARAMETROS

- No node é possível passar parâmetros através da variável req que criamos na função de callback referente a cada rota.



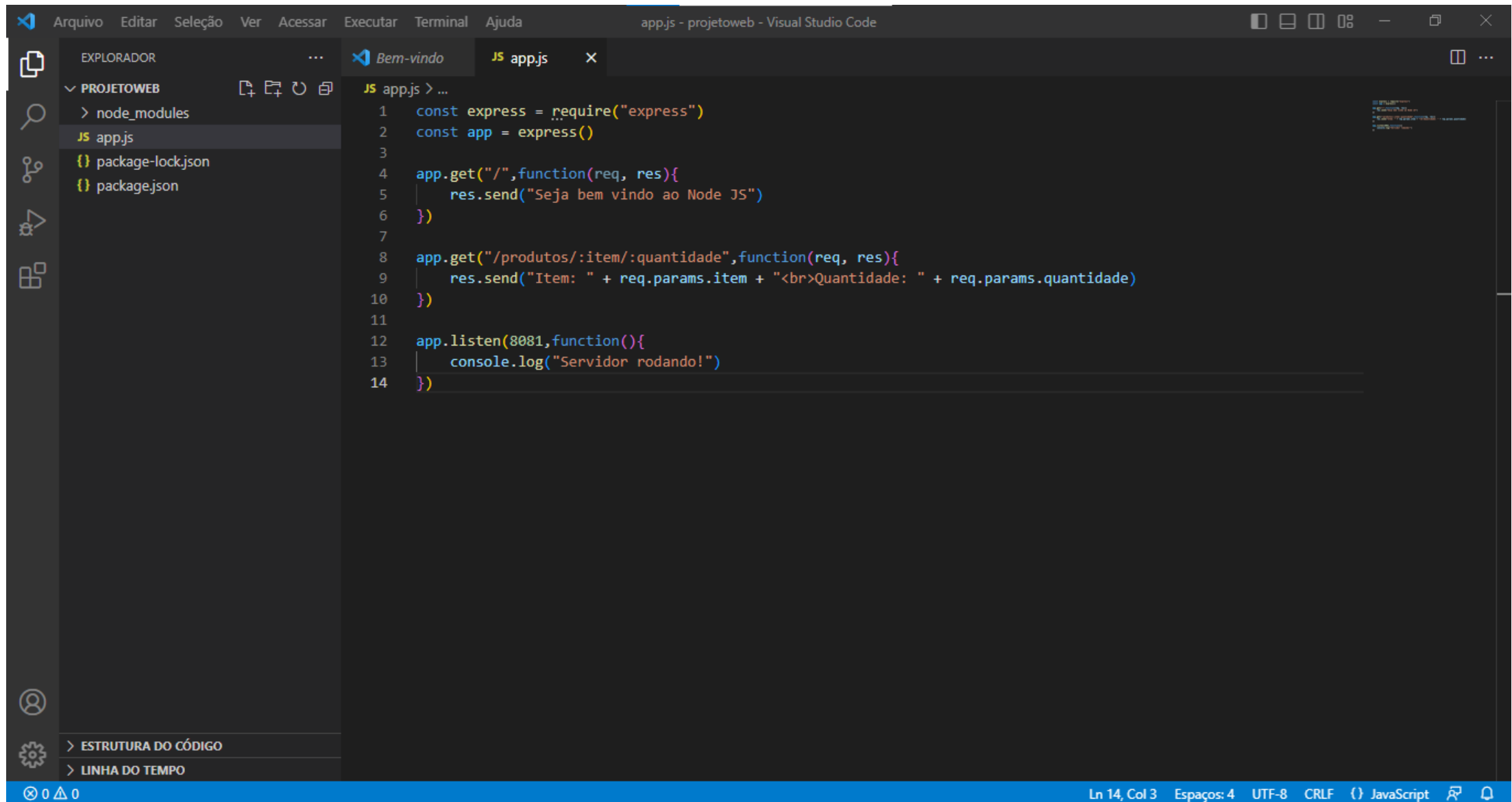
RECEBENDO PARAMETROS

- Através da variável req podemos passar um valor para ser exibido na respectiva página da rota.



RECEBENDO PARAMETROS

- Podemos receber diversos parâmetros através da url e esses valores podem ser utilizados na respectiva página da rota.



RECEBENDO PARAMETROS

- Para visualizarmos os resultados podemos utilizar o navegador criando dados na url:
 - localhost:8081/produtos/borracha/10
- Na url criamos os valores borracha e o número 10, o resultado pode ser analisado no próximo slide.



localhost:8081/produtos/borracha/10



Item: borracha
Quantidade: 10

UTILIZANDO O NODEMON

- Para não precisarmos reiniciar o terminal toda vez que fizemos uma alteração no nosso projeto node, podemos utilizar a ferramenta nodemon que fará esse serviço automaticamente, após acessar a sua aplicação no respectivo diretório do seu projeto digite o seguinte comando:
 - `npm install nodemon -g`
- O parâmetro `-g` permitirá que esse recurso seja implementado globalmente em seu projeto.

C:\Windows\system32\cmd.exe

```
C:\Users\Jeferson\Documents\projetoweb>npm install nodemon -g
```

```
changed 32 packages in 1s
```

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

```
C:\Users\Jeferson\Documents\projetoweb>
```

UTILIZANDO O NODEMON

- Para utilizarmos o nodemon precisamos digitar o seguinte comando:
 - `nodemon app.js`
- Esse recurso verificará o cada alteração no seu código node e realizará a reinicialização do servidor no terminal.

C:\Windows\system32\cmd.exe - "node" "C:\Users\Jeferson\AppData\Roaming\npm\node_modules\nodemon\bin\nodemon.js" app.js

```
C:\Users\Jeferson\Documents\projetoweb>npm install nodemon -g
```

```
changed 32 packages in 1s
```

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

```
C:\Users\Jeferson\Documents\projetoweb>nodemon app.js
```

```
[nodemon] 2.0.20
```

```
[nodemon] to restart at any time, enter `rs`
```

```
[nodemon] watching path(s): *.*
```

```
[nodemon] watching extensions: js,mjs,json
```

```
[nodemon] starting `node app.js`
```

```
Servidor rodando!
```


C:\Windows\system32\cmd.exe - "node" "C:\Users\Jeferson\AppData\Roaming\npm\node_modules\nodemon\bin\nodemon.js" app.js

```
C:\Users\Jeferson\Documents\projetoweb>npm install nodemon -g
```

```
changed 32 packages in 1s
```

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

```
C:\Users\Jeferson\Documents\projetoweb>nodemon app.js
```

```
[nodemon] 2.0.20
```

```
[nodemon] to restart at any time, enter `rs`
```

```
[nodemon] watching path(s): *.*
```

```
[nodemon] watching extensions: js,mjs,json
```

```
[nodemon] starting `node app.js`
```

```
Servidor rodando!
```

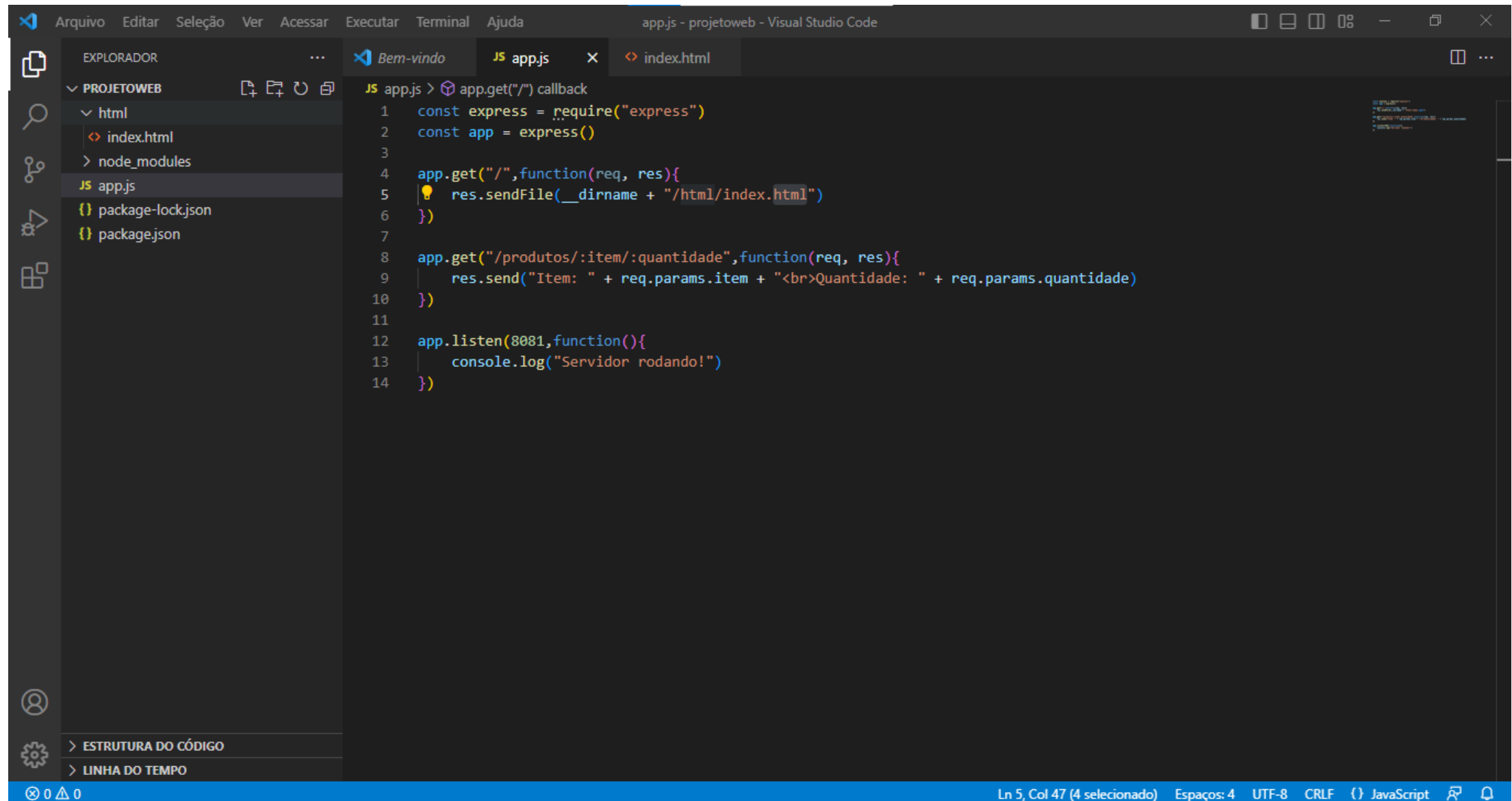
```
[nodemon] restarting due to changes...
```

```
[nodemon] starting `node app.js`
```

```
Servidor rodando!
```

UTILIZANDO PÁGINAS HTML

- Podemos direcionar um arquivo cada vez que uma rota é acessada através do seguinte comando:
 - `res.sendFile(__dirname + "/html")`
- O comando `__dirname` é utilizado para recuperarmos o caminho completo do nosso diretório na respectiva máquina.



Seja bem vindo ao Node Js com HTML