Prof. Esp. Jeferson Roberto de Lima

DESENVOLVIMENTO WEB II

DOCENTE

- Pós-graduado em Desenvolvimento de Aplicações Corporativas em Java pela Universidade Cidade de São Paulo (2014), Licenciado em Formação Pedagógica de Docentes pela Faculdade de Tecnologia do Estado de São Paulo (2014), Graduado em Sistema de Informação pela Universidade Bandeirante de São Paulo (2010).

EMENTA

- Relação entre arquitetura de dados, arquitetura de informação em um sistema web e arquitetura de sistema. Persistência de dados em sistemas web. Ambientes virtuais e sistemas de construção de software aplicados ao desenvolvimento web no que tange ao isolamento do ambiente de desenvolvimento, obtenção de dependências e automação de diferentes tarefas presentes no ciclo de desenvolvimento.

EMENTA

- Tecnologias de persistência de dados incluindo frameworks para mapeamento objeto-relacional aplicadas ao desenvolvimento de sistemas web. Sistemas web com persistência de dados e chamadas assíncronas. Páginas feitas pelo Back-end. Coockies. Escopos de Memória (Aplicação, Sessão). Criação de aplicações Web que consumam APIs públicas e abertas. Hospedagem do sistema. Controle de versionamento.

INSTALAÇÃO

Node;

Vs Studio Code;

INSTALAÇÃO DOS COMPONENTES

- Para iniciarmos os nosso projetos em Node JS, será necessário realizar o download do mesmo no site oficial da tecnologia:
- https://nodejs.org/en/download/
- https://code.visualstudio.com/download

O QUE É O NODE JS?

- Node.js é um ambiente de execução JavaScript que permite executar aplicações desenvolvidas com a linguagem de forma autônoma, sem depender de um navegador.

JAVASCRIPT E O NODE JS

- JavaScript é uma linguagem de programação que originalmente foi desenvolvida para trazer maior interatividade aos websites através da manipulação do *DOM* (Document Object Model).
- O JavaScript nasceu para atender demandas voltadas ao Front e como as necessidades aumentam de acordo com o crescimento tecnológico, surgiu a ideia de utilizar uma mesma linguagem no lado do cliente e do servidor para otimizar processos e serviços.

NODE JS

- De acordo com sua definição oficial, o Node é um runtime, que nada mais é do que um conjunto de códigos, API's, ou seja, são bibliotecas responsáveis pelo tempo de execução (é o que faz o seu programa rodar) que funciona como um interpretador de JavaScript fora do ambiente do navegador web.
- o Node.JS é um ambiente de execução assíncrono, isto é, ele trabalha de modo a não bloquear no momento da execução da aplicação, delegando os processos demorados a um segundo plano.

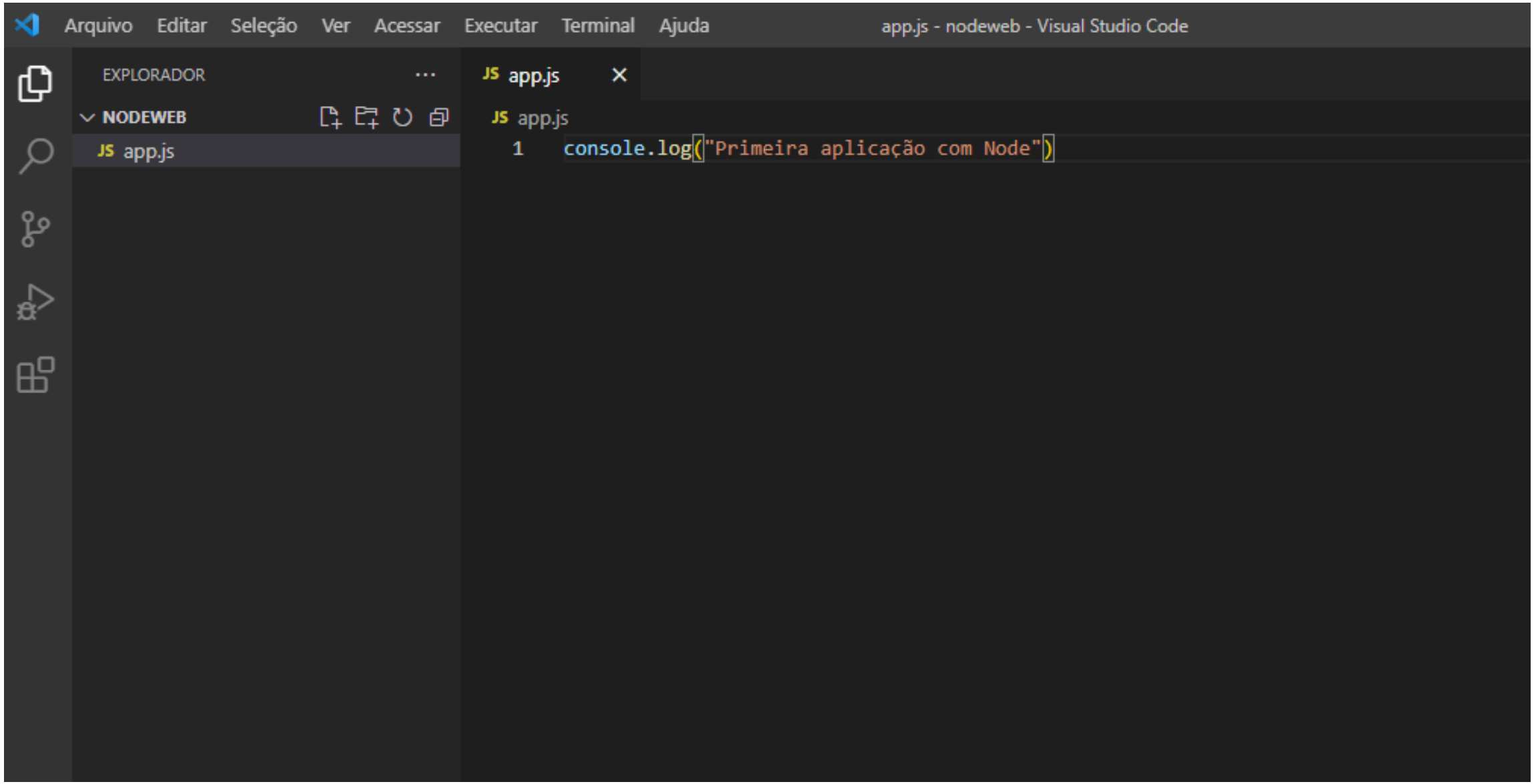
CRIANDO A PRIMEIRA APLICAÇÃO

MENSAGEM NO TERMINAL

- Podemos exibir mensagem no terminal para analisarmos o comportamento da aplicação.
- Para isso utilizamos o comando:

```
console.log("Mensagem a ser exibida");
```

- Se você reparar é o mesmo comando utilizado no JavaScript para trabalhar com log do lado do browser.


Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
app.js - nodeweb - Visual Studio Code
EXPLORADOR
NODEWEB
app.js
app.js
app.js
1 console.log("Primeira aplicação com Node")

VARIÁVEIS E OPERAÇÕES MATEMÁTICAS

- No Node JS podemos criar variáveis e realizar as respectivas operações matemáticas convencionais como:

  Adição
  Subtração
  Multiplicação
  Divisão

- Exemplo de utilização de variáveis:

  ```
  var n1 = 10;
  ```

- Nesse caso estamos criando uma variável n1 com o valor igual a 10.

VARIÁVEIS E OPERAÇÕES MATEMÁTICAS

- A operações básicas da matemática são representadas pelos seguintes sinais:

  Adição +

  Subtração -

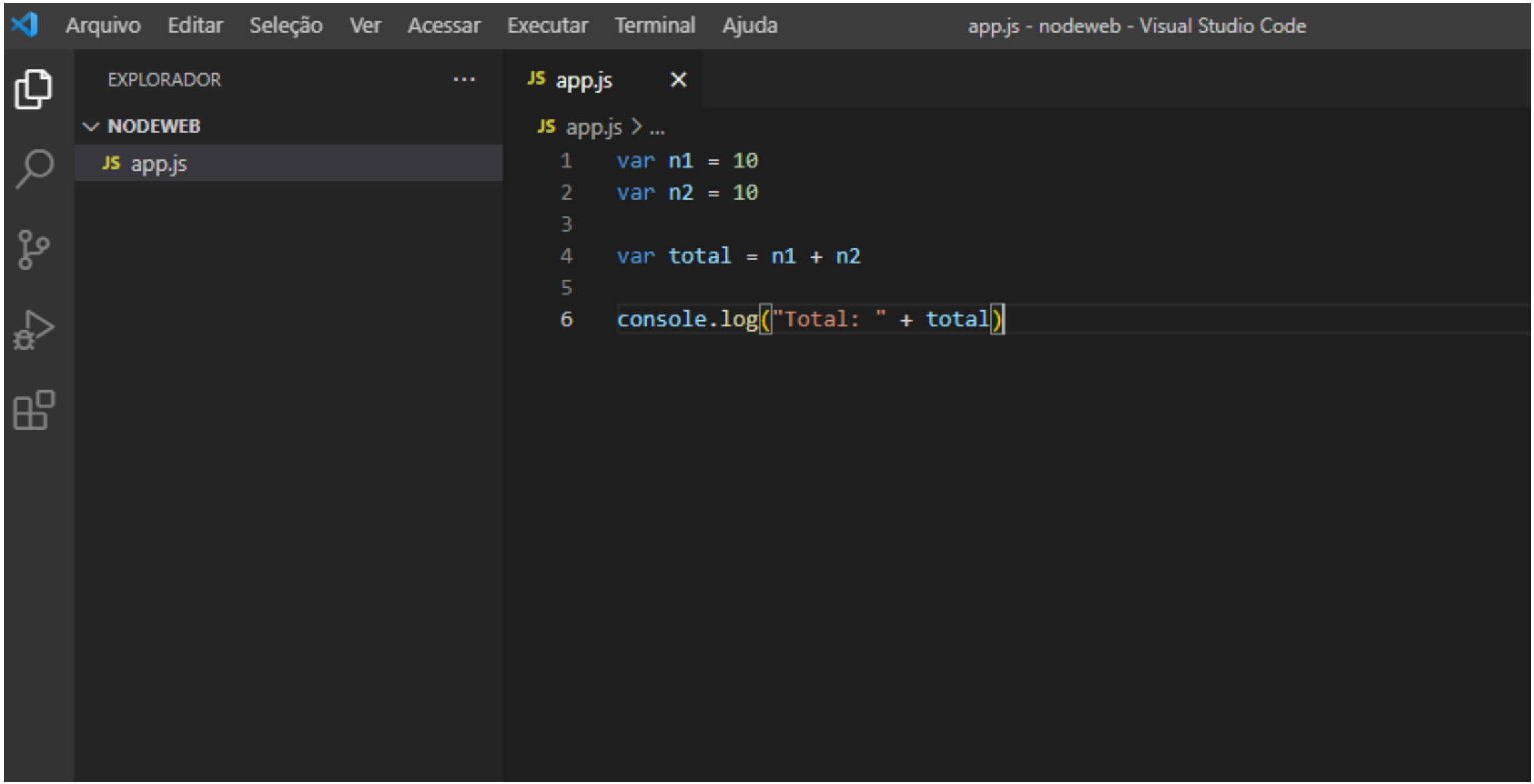
  Multiplicação *

  Divisão /

- Exemplo de operação matemática:

  total = n1 + n2

- Nesse exemplo temos anteriormente a criação das três variáveis com os respectivos valores e depois a realização da operação de adição.

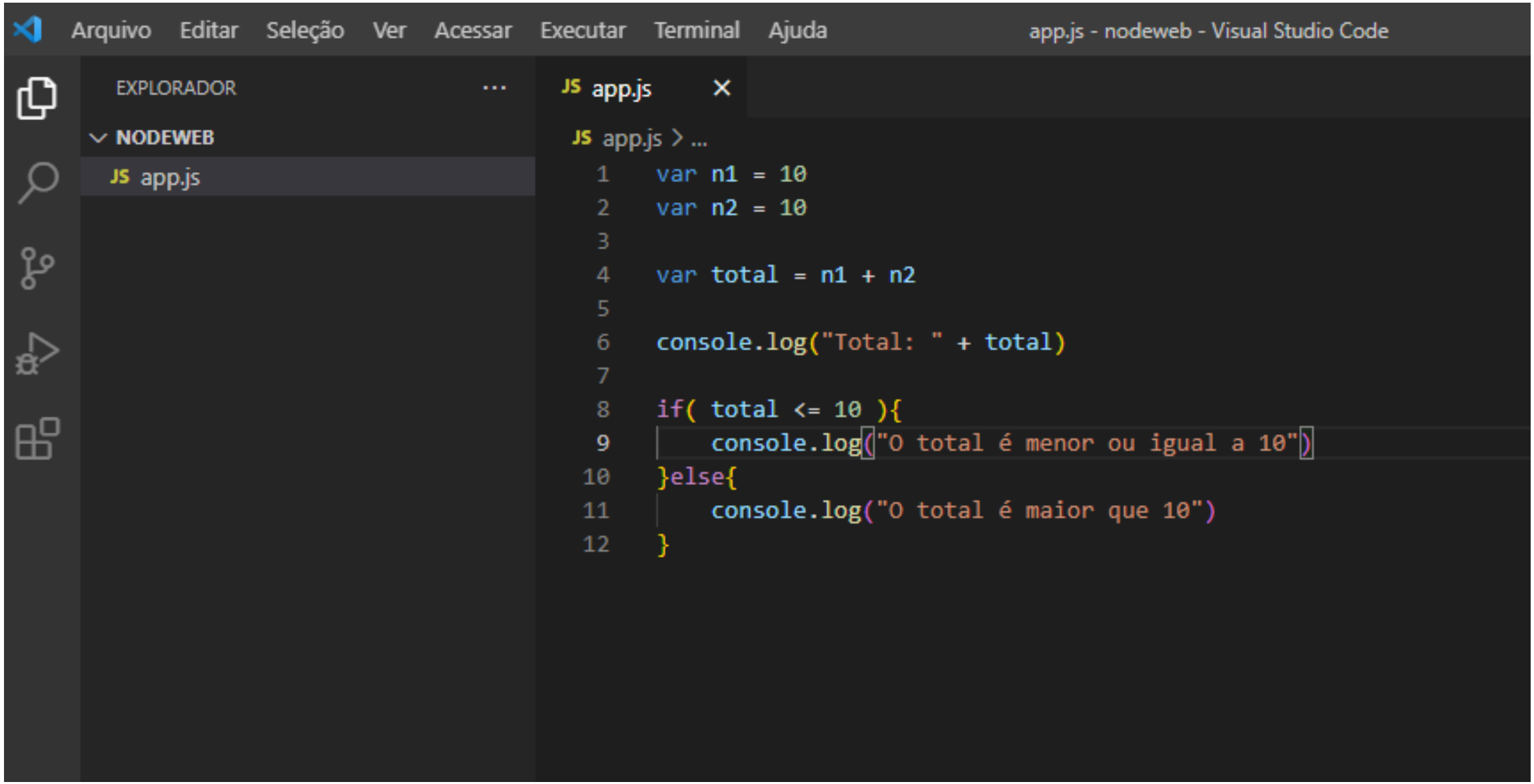

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
app.js - nodeweb - Visual Studio Code
EXPLORADOR
NODEWEB
app.js
app.js
app.js > ...
1 var n1 = 10
2 var n2 = 10
3
4 var total = n1 + n2
5
6 console.log("Total: " + total)

SISTEMA CONDICIONAL

- Podemos criar condições para exibição de determinados resultados na nossa programação, no exemplo abaixo utilizaremos o sistema de condicional (if/else) para exibição de uma mensagem no terminal.

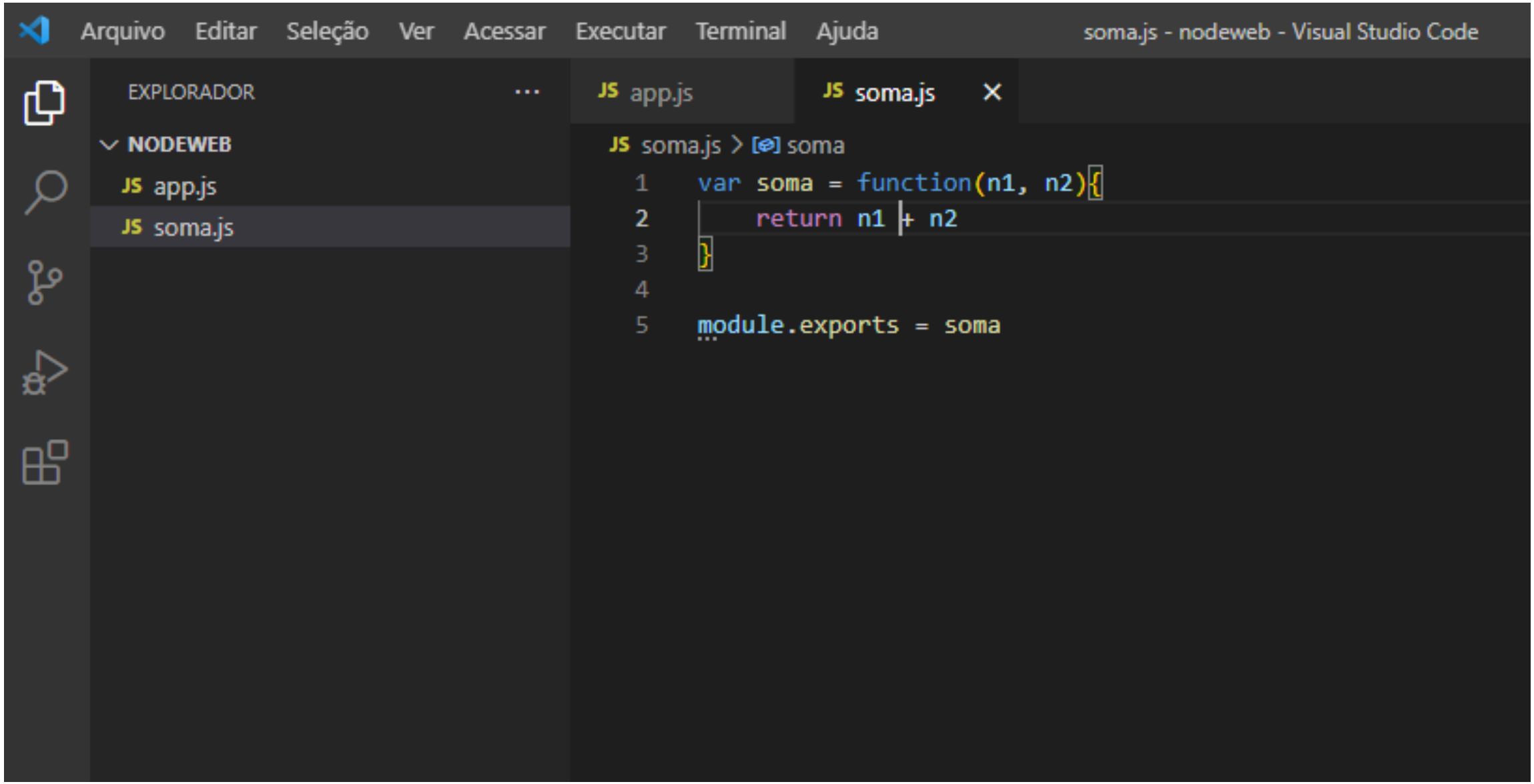
```
if(total <= 10){ console.log("True") }else{ console.log("False") }
```

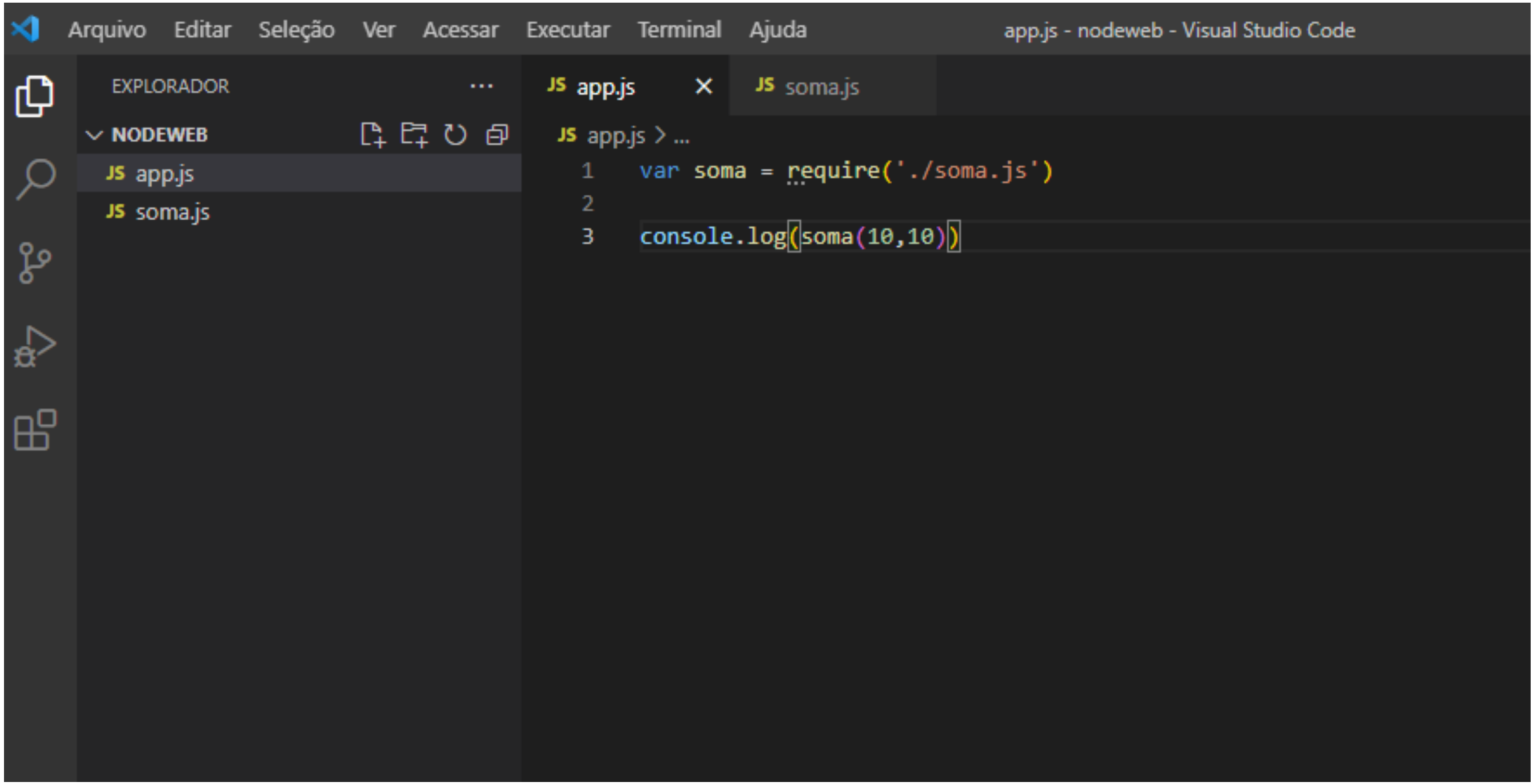
- Nesse exemplo estamos testando a variável total para verificar se o valor da mesma é maior ou igual a 10.


Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
app.js - nodeweb - Visual Studio Code
EXPLORADOR
NODEWEB
app.js
app.js
app.js > ...
1 var n1 = 10
2 var n2 = 10
3
4 var total = n1 + n2
5
6 console.log("Total: " + total)
7
8 if(total <= 10){
9 console.log("O total é menor ou igual a 10")
10 }else{
11 console.log("O total é maior que 10")
12 }

EXPORTANDO OS MÓDULOS

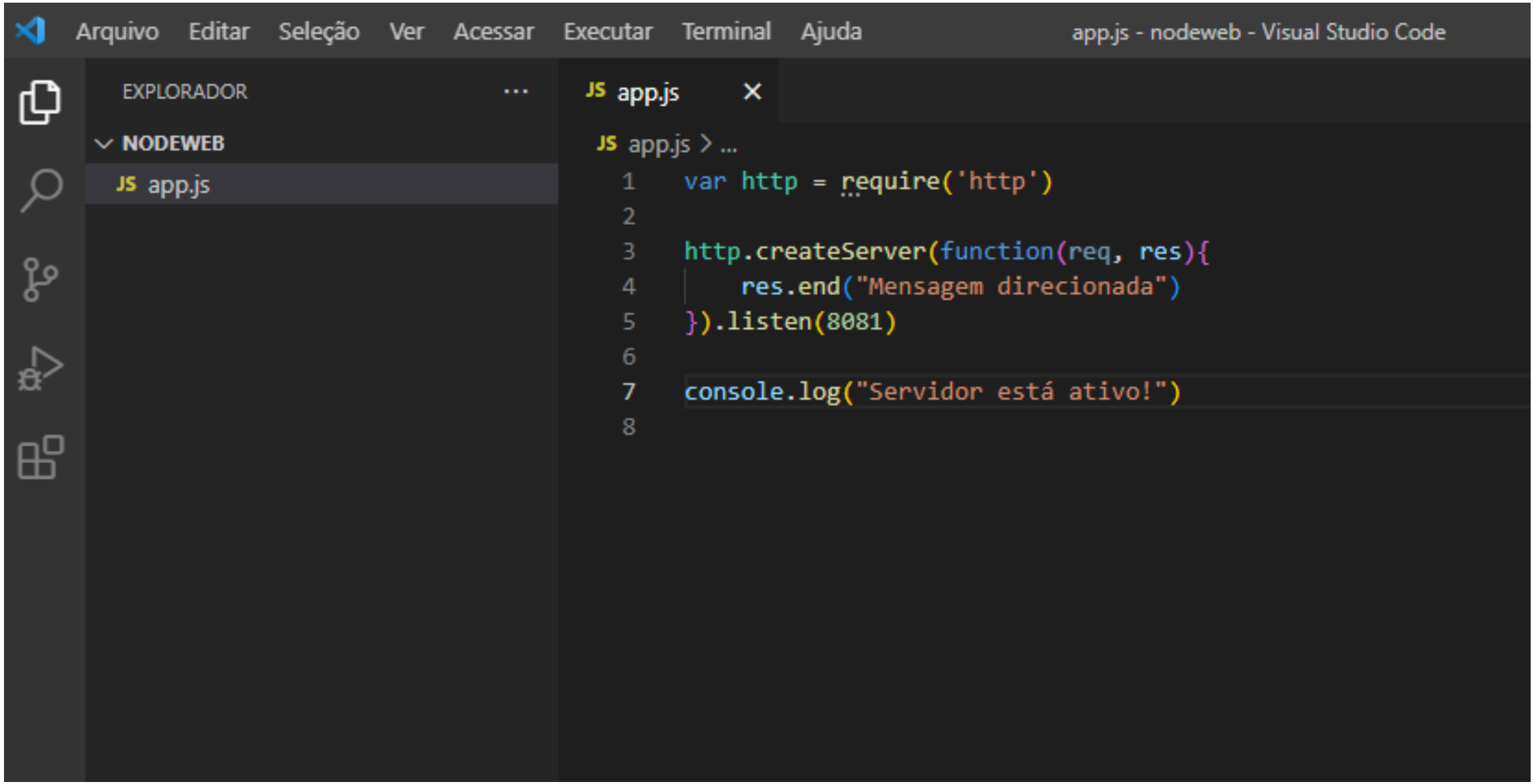
- No Node JS podemos exportar as funções criadas para reutilizamos em outros ambientes, esse processo é exemplificado no próximo slide.

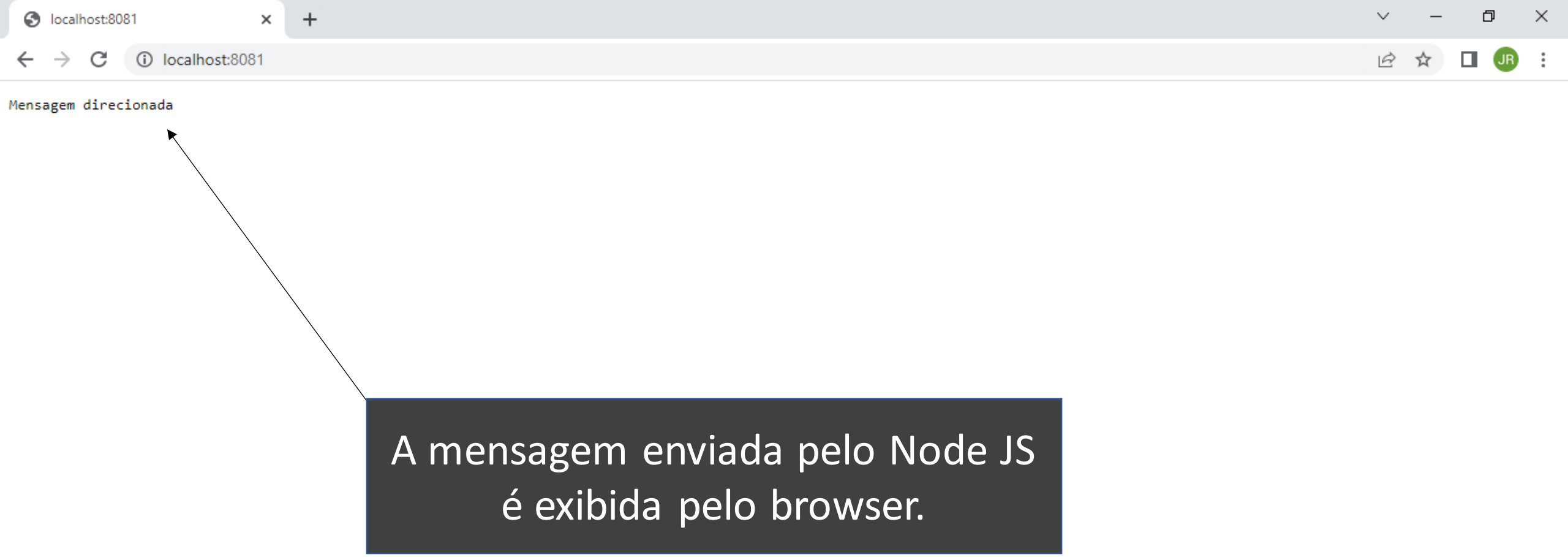

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
soma.js - nodeweb - Visual Studio Code
EXPLORADOR
NODEWEB
app.js
soma.js
app.js
soma.js
soma.js > soma
1 var soma = function(n1, n2){
2 return n1 + n2
3 }
4
5 module.exports = soma


Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
app.js - nodeweb - Visual Studio Code
EXPLORADOR
NODEWEB
app.js
soma.js
app.js
soma.js
app.js > ...
1 var soma = require('./soma.js')
2
3 console.log(soma(10,10))

CRIAÇÃO DE UM SERVIDOR WEB

- Para iniciarmos um conteúdo em uma página web primeiramente devemos criar um servidor HTTP e direcionar uma porta para testarmos a aplicação em um respectivo browser, segue a construção do exemplo no próximo slide.


Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda
app.js - nodeweb - Visual Studio Code
EXPLORADOR
NODEWEB
app.js
app.js
app.js > ...
1 var http = require('http')
2
3 http.createServer(function(req, res){
4 res.end("Mensagem direcionada")
5 }).listen(8081)
6
7 console.log("Servidor está ativo!")
8


localhost:8081
localhost:8081
JR
Mensagem direcionada
A mensagem enviada pelo Node JS é exibida pelo browser.