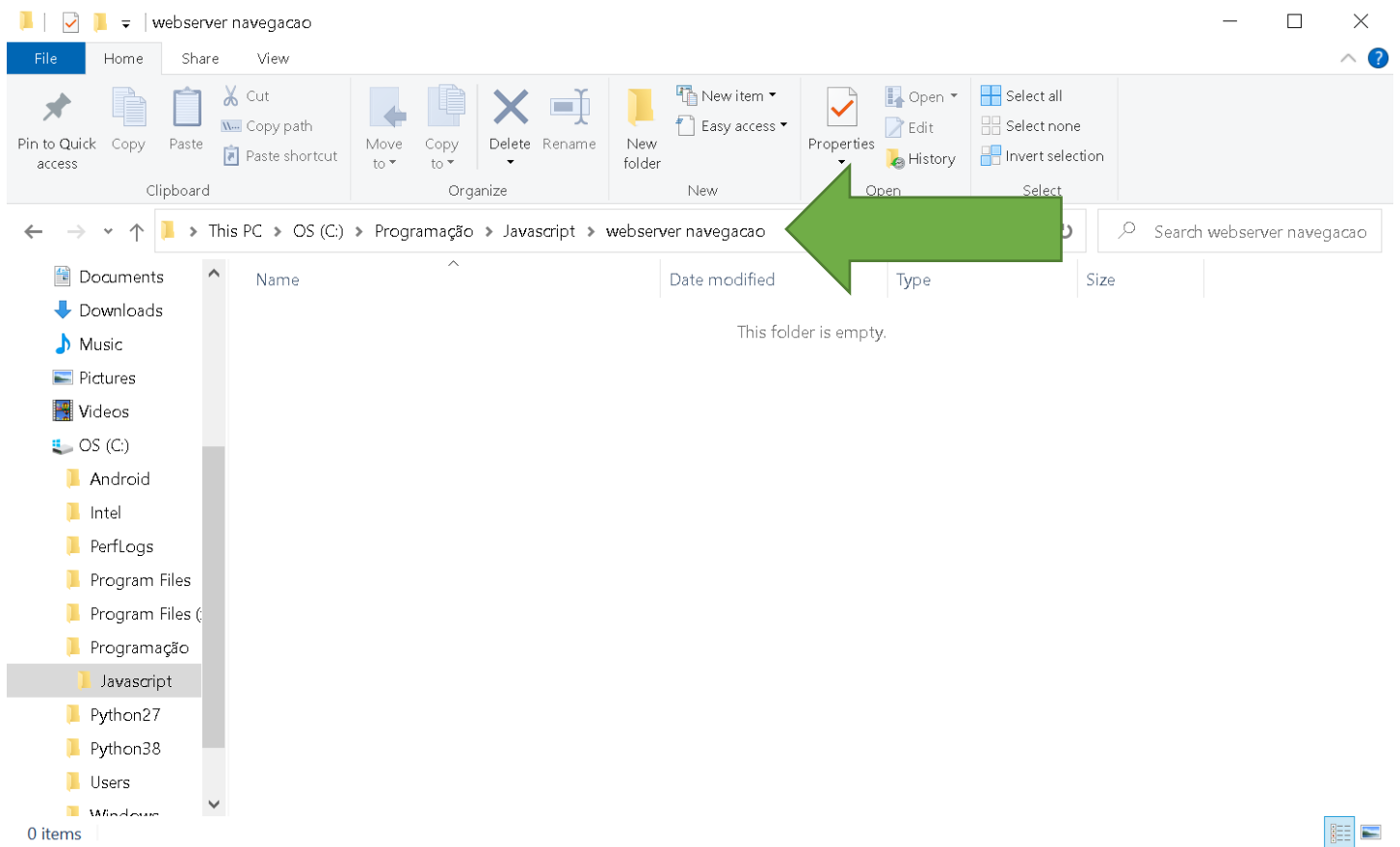
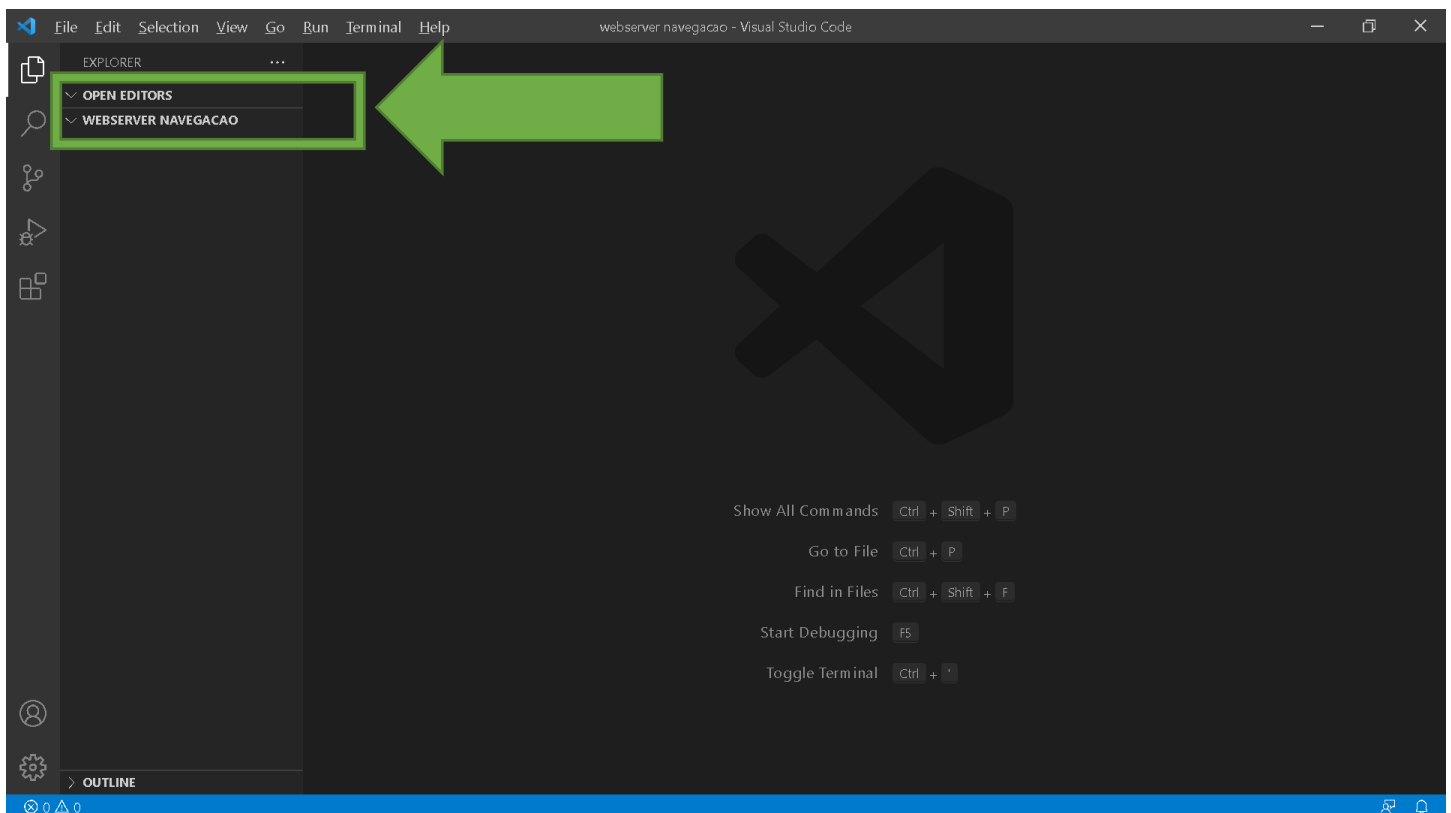


PLEASE, TRANSLATE FROM PORTUGUESE ALL INSTRUCTIONS WITHIN THIS DOCUMENT.

⇒ Crie uma pasta onde os arquivos de seu projeto serão criados:



⇒ Abra a pasta do projeto no VS Code:





⇒ Abra o terminal em modo administrador e prossiga com os comandos:

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Julio>cd C:\Programação\Javascript\webserver navegacao

C:\Programação\Javascript\webserver navegacao>npm install express
npm WARN saveError ENOENT: no such file or directory, open 'C:\Programação\Javascript\webserver navegacao\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:\Programação\Javascript\webserver navegacao\package.json'
npm WARN webserver navegacao No description
npm WARN webserver navegacao No repository field.
npm WARN webserver navegacao No README data
npm WARN webserver navegacao No license field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 2.388s
found 0 vulnerabilities

C:\Programação\Javascript\webserver navegacao>
```

⇒ Inicie o projeto:

```
npm
(c) Microsoft Corporation. All rights reserved.

C:\Users\Julio>cd C:\Programação\Javascript\webserver navegacao

C:\Programação\Javascript\webserver navegacao>npm install express
npm WARN saveError ENOENT: no such file or directory, open 'C:\Programação\Javascript\webserver navegacao\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:\Programação\Javascript\webserver navegacao\package.json'
npm WARN webserver navegacao No description
npm WARN webserver navegacao No repository field.
npm WARN webserver navegacao No README data
npm WARN webserver navegacao No license field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 2.388s
found 0 vulnerabilities

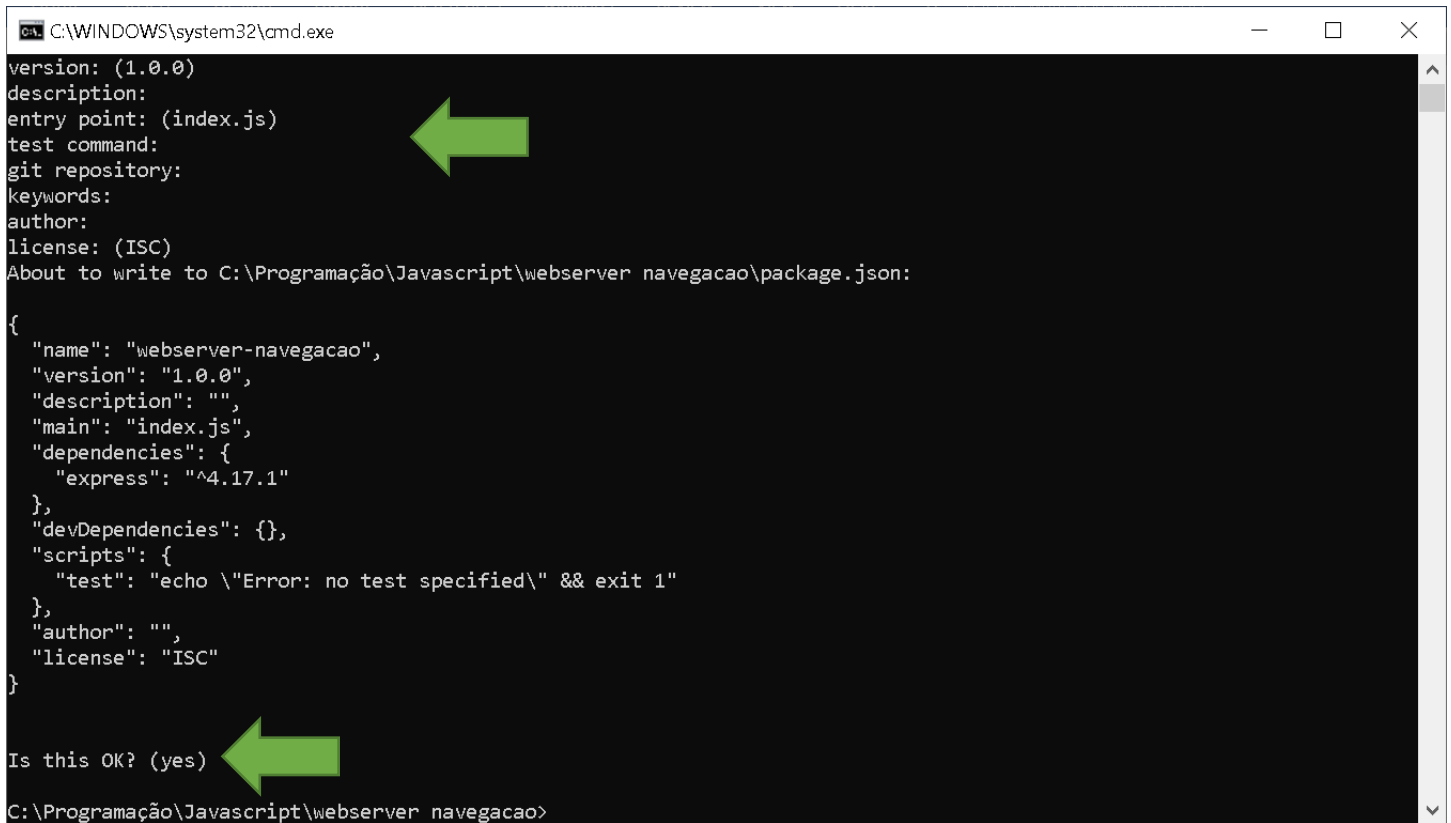
C:\Programação\Javascript\webserver navegacao>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (webserver-navegacao)
```

⇒ Pressione “enter” em todas as confirmações:

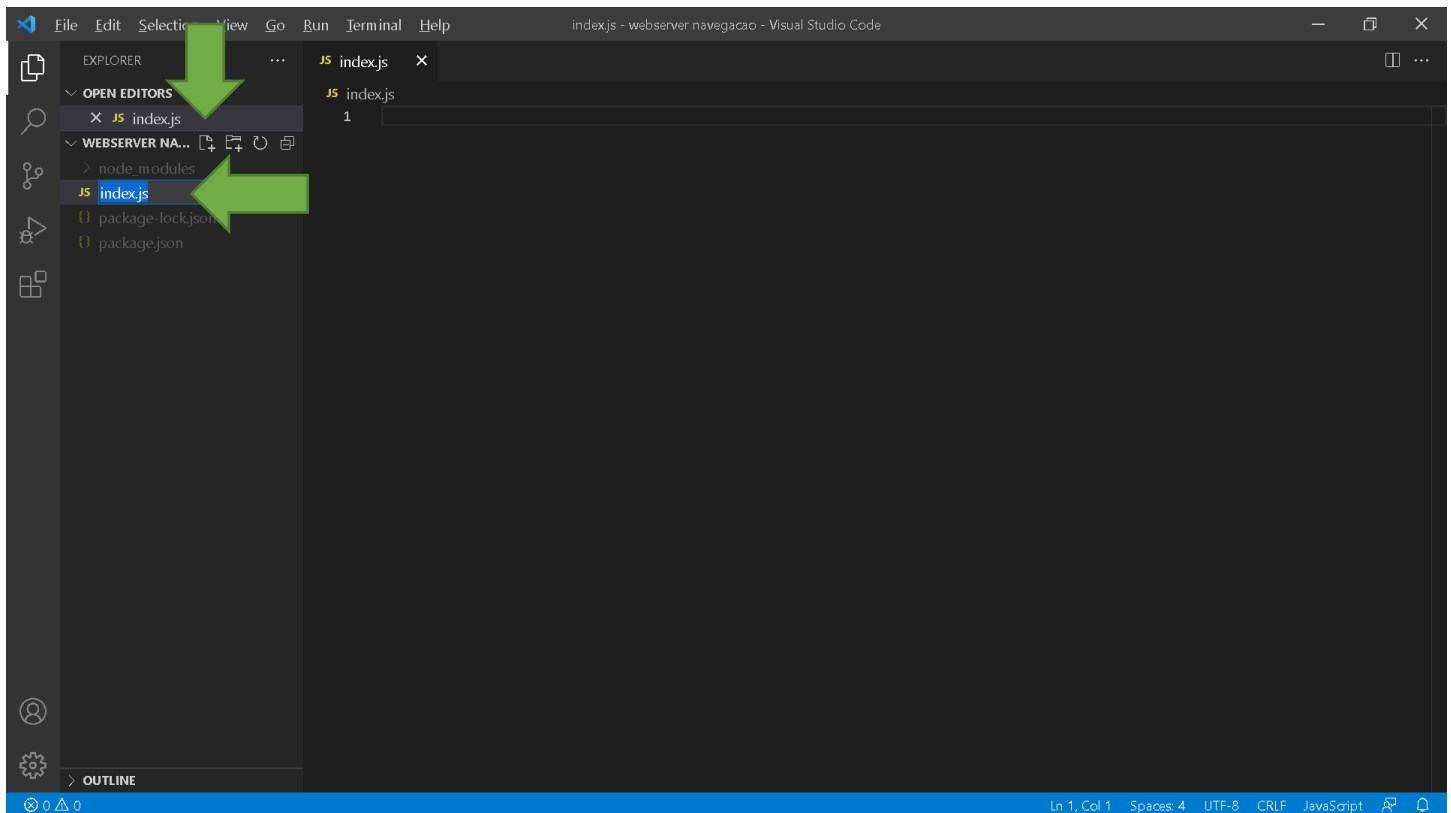


```
C:\WINDOWS\system32\cmd.exe
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Programação\Javascript\webserver navegacao\package.json:

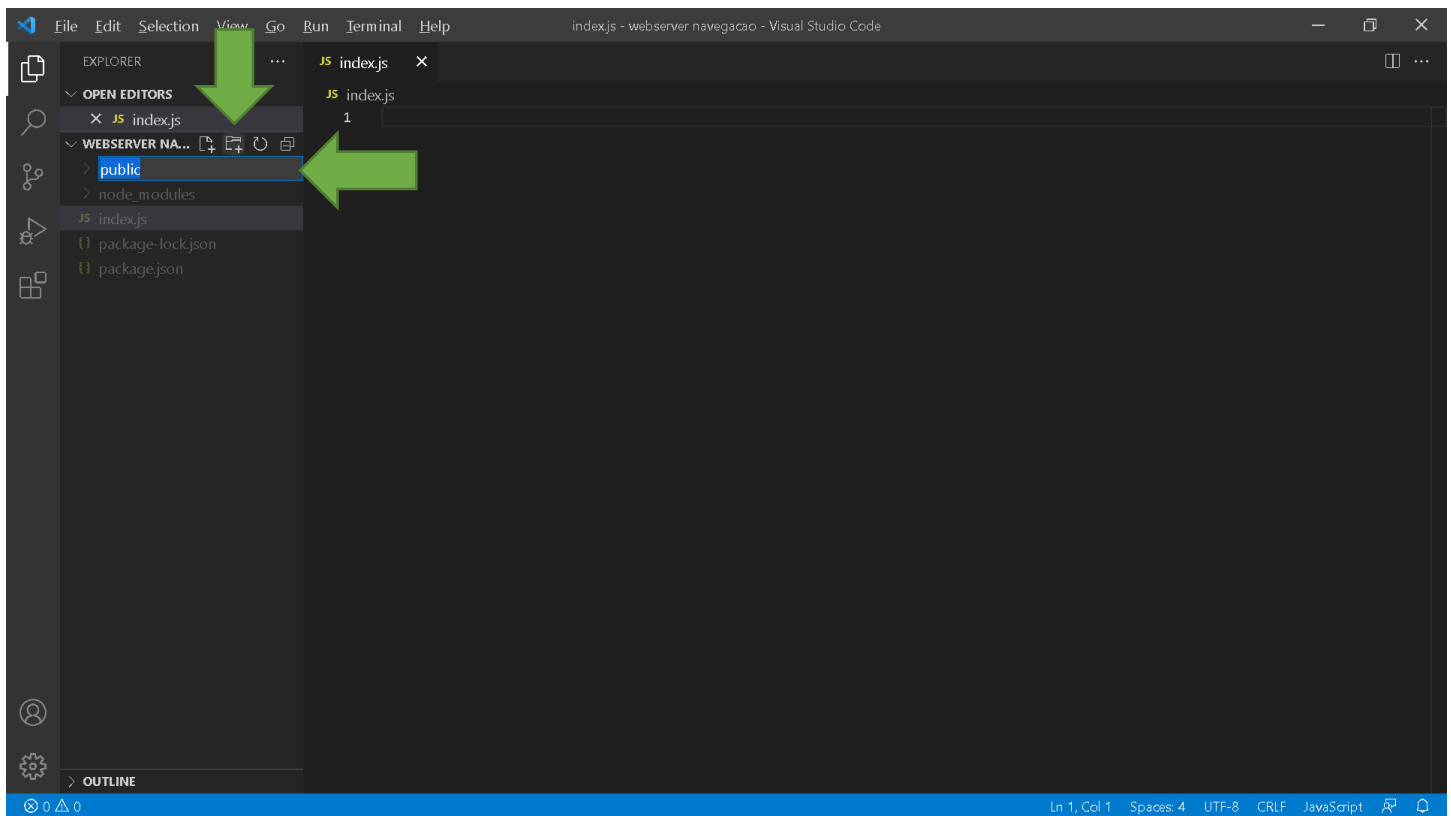
{
  "name": "webserver-navegacao",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "dependencies": {
    "express": "^4.17.1"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)
C:\Programação\Javascript\webserver navegacao>
```

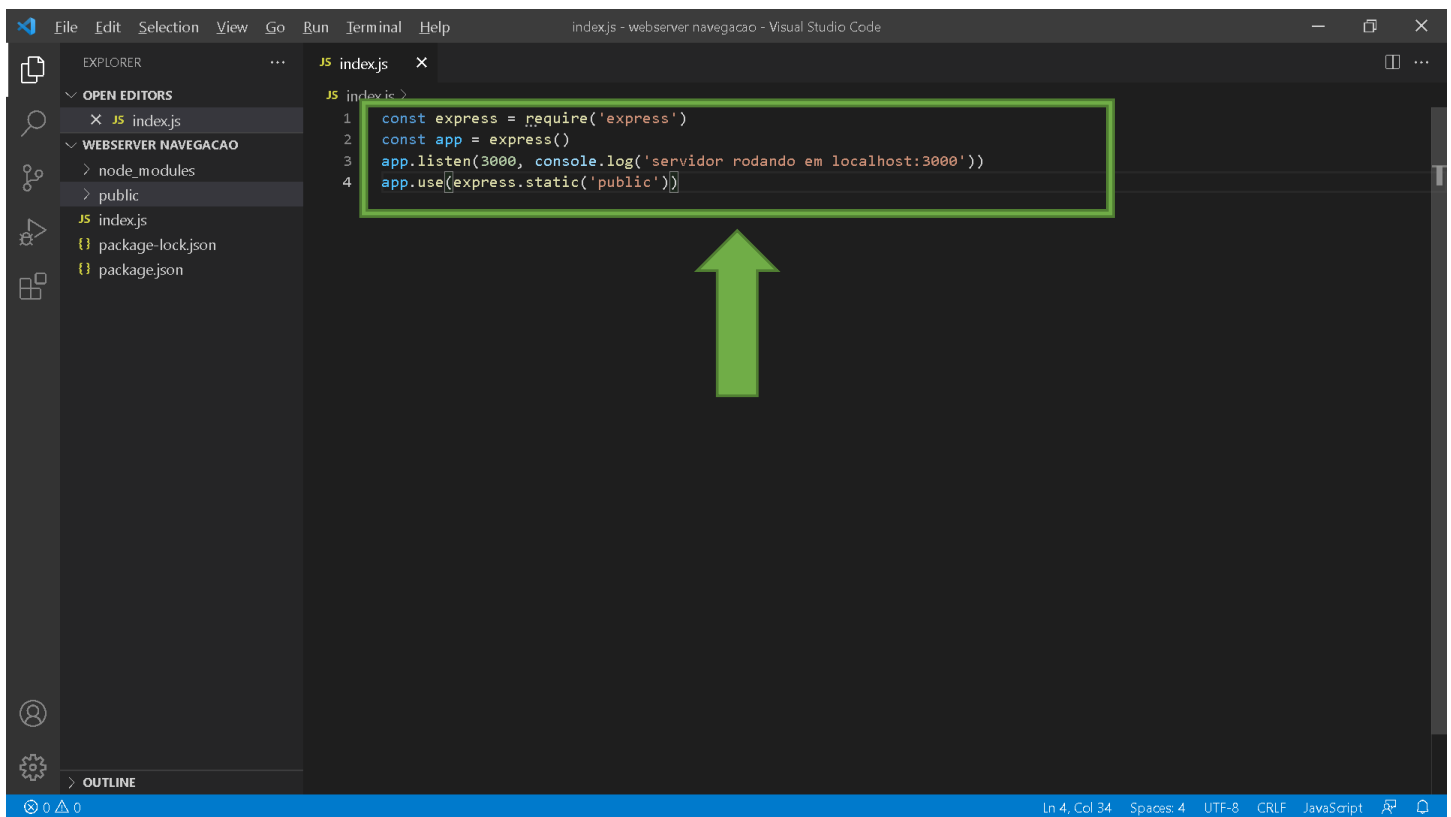
⇒ Crie o arquivo index.js:



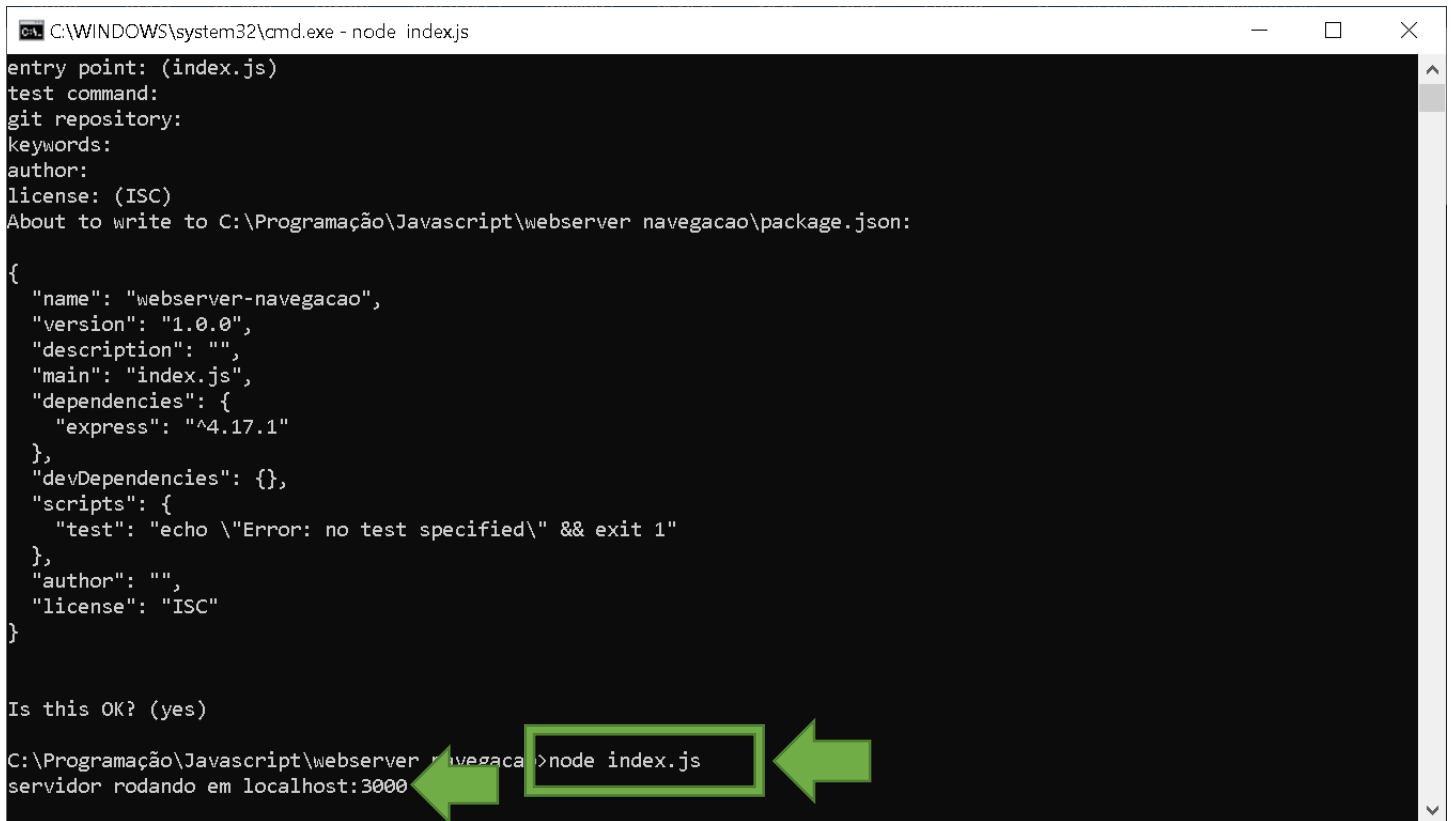
⇒ Crie a pasta public:



⇒ Digite o código mínimo para um servidor em Node express:



⇒ Rode o servidor para confirmar se está tudo ok:



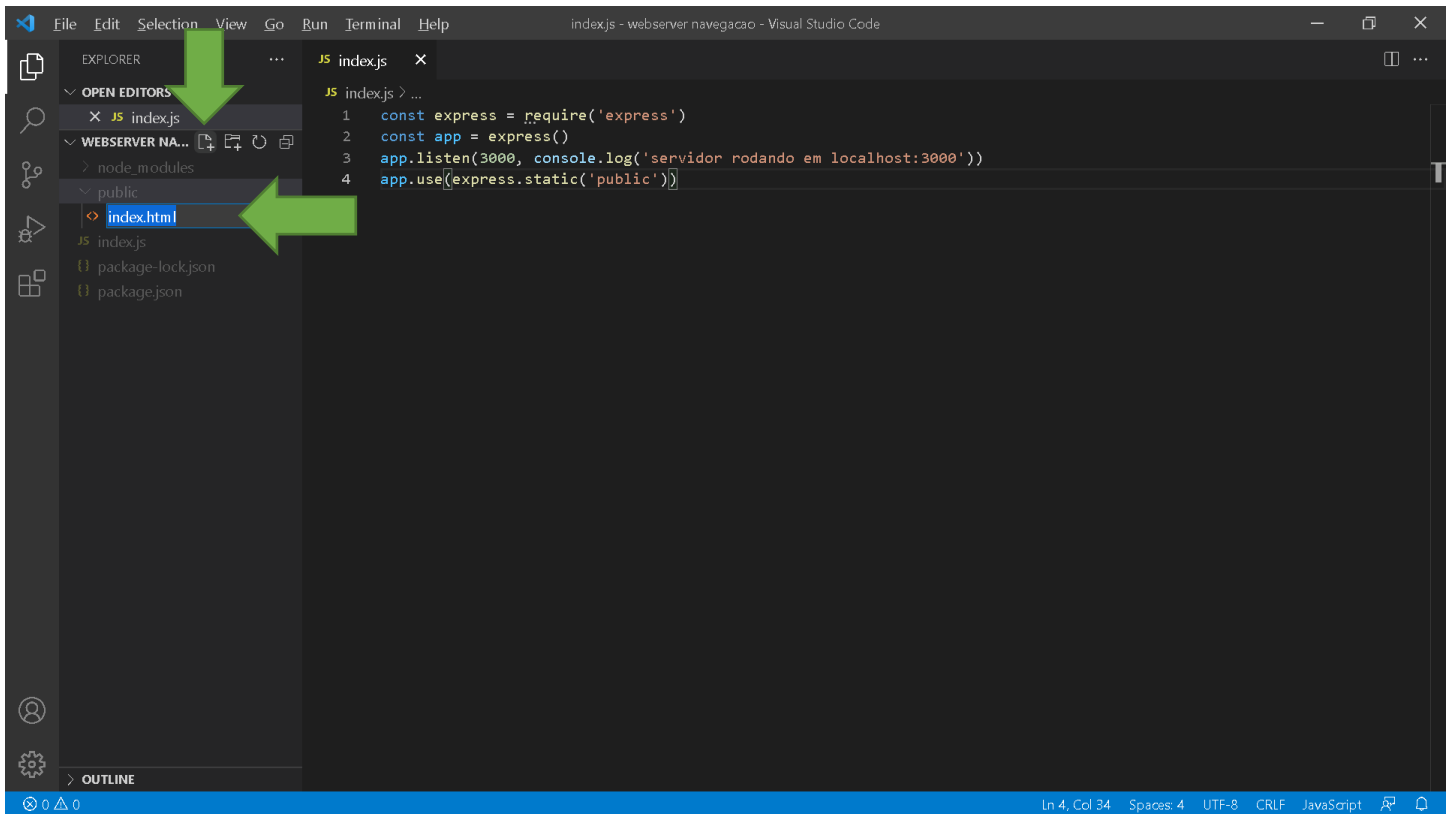
```
C:\WINDOWS\system32\cmd.exe - node index.js
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Programação\Javascript\webserver navegacao\package.json:

{
  "name": "webserver-navegacao",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "dependencies": {
    "express": "^4.17.1"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)
C:\Programação\Javascript\webserver navegacao>node index.js
servidor rodando em localhost:3000
```

The screenshot shows a Windows command prompt window with a black background and white text. The title bar reads 'C:\WINDOWS\system32\cmd.exe - node index.js'. The terminal output shows the creation of a package.json file for a project named 'webserver-navegacao'. The file includes fields for name, version (1.0.0), description, main (index.js), dependencies (express ^4.17.1), devDependencies, scripts (test: echo \"Error: no test specified\" && exit 1), author, and license (ISC). A prompt 'Is this OK? (yes)' is shown. Below, the command 'node index.js' is executed, and the output 'servidor rodando em localhost:3000' is displayed. Three green arrows are overlaid on the image: one points to the 'node index.js' command, another points to the output 'servidor rodando em localhost:3000', and a third points to the 'node' part of the command.

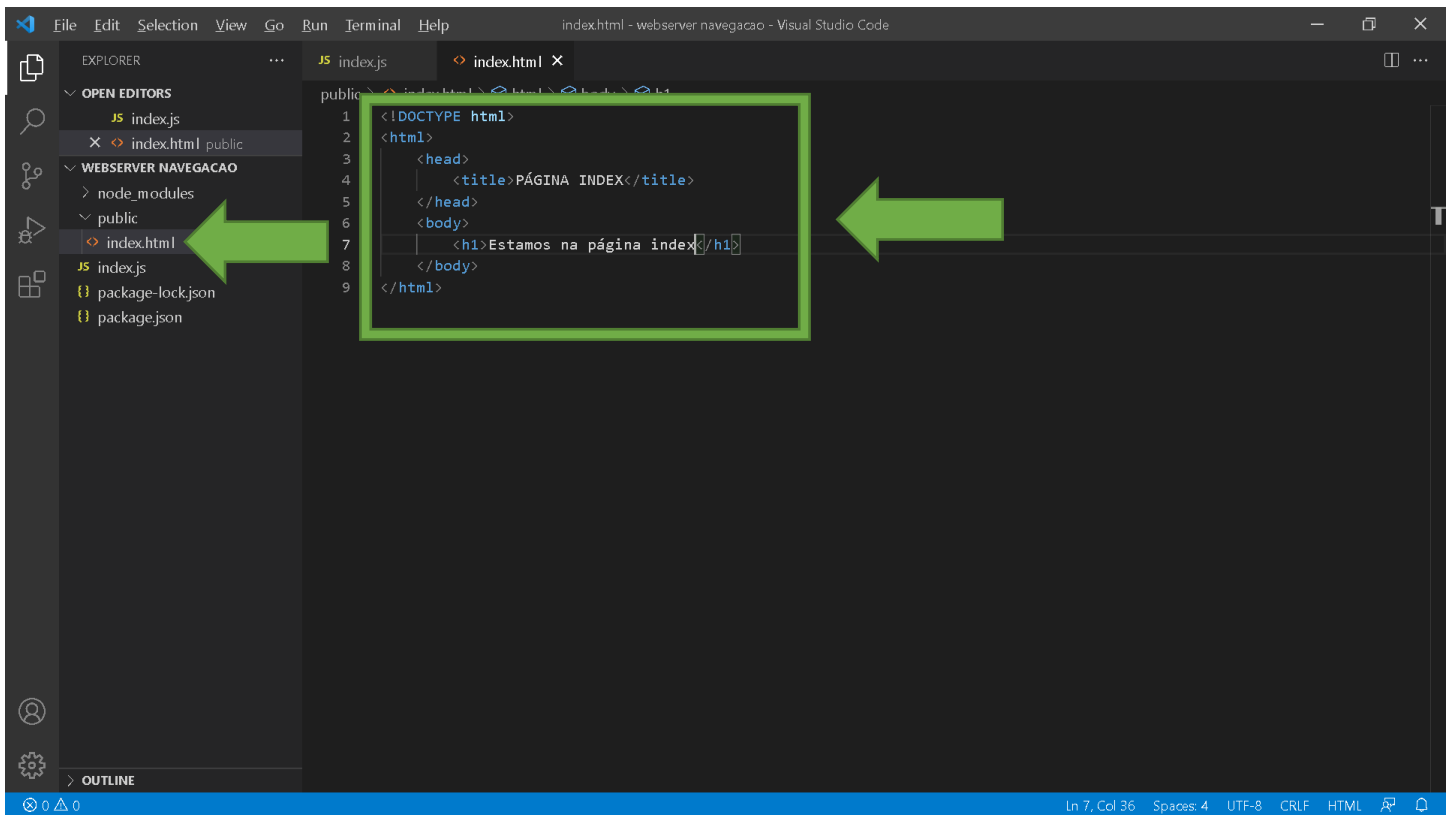
⇒ Crie o arquivo index.html na pasta public:



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar is expanded to show the 'public' folder, and the 'index.html' file is highlighted. A green arrow points to the 'index.html' file in the Explorer. Another green arrow points to the 'index.js' file in the Explorer. The main editor area shows the content of 'index.js'.

```
1 const express = require('express')
2 const app = express()
3 app.listen(3000, console.log('servidor rodando em localhost:3000'))
4 app.use(express.static('public'))
```

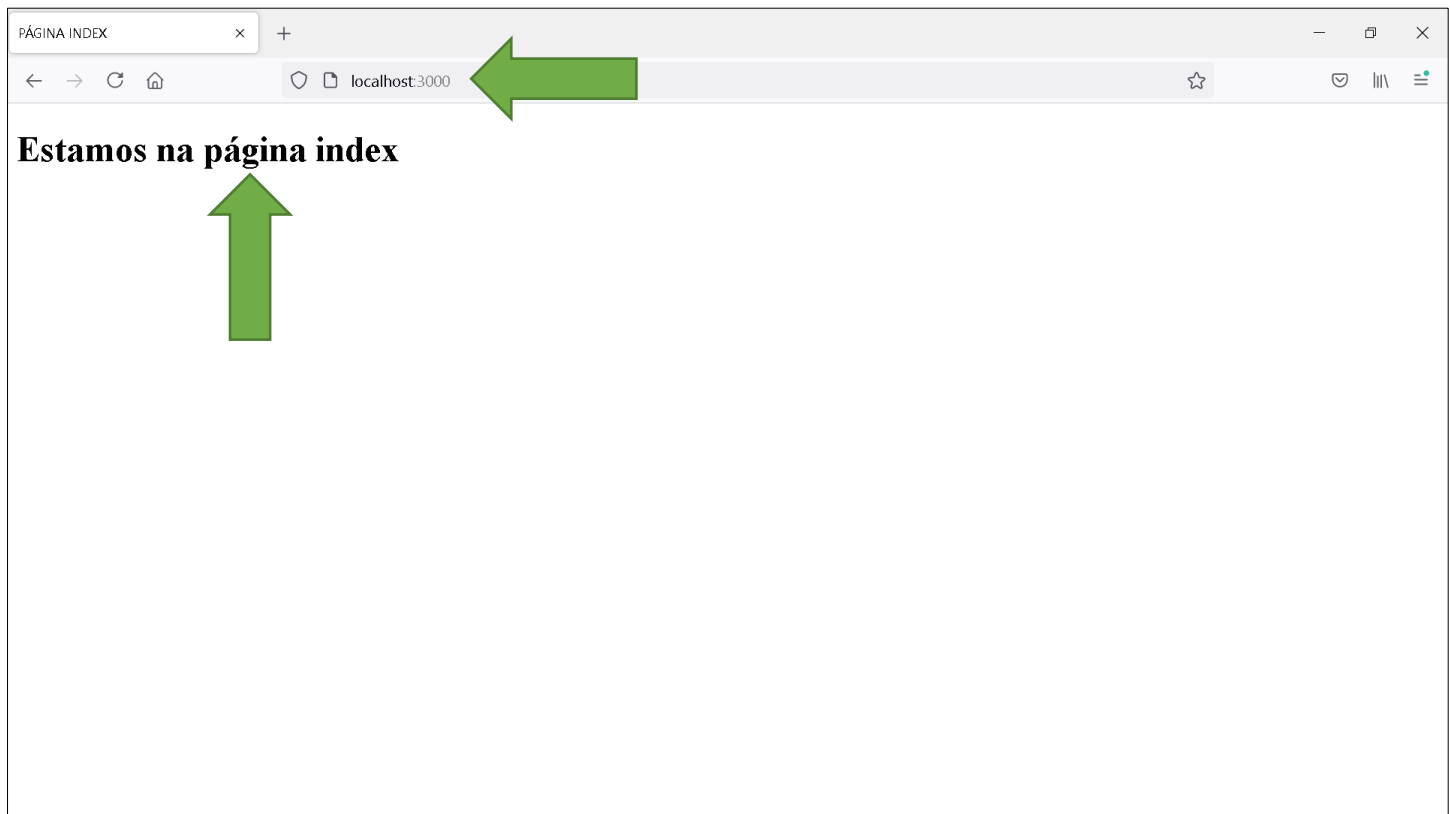
⇒ Edite o arquivo index.html com o código abaixo:



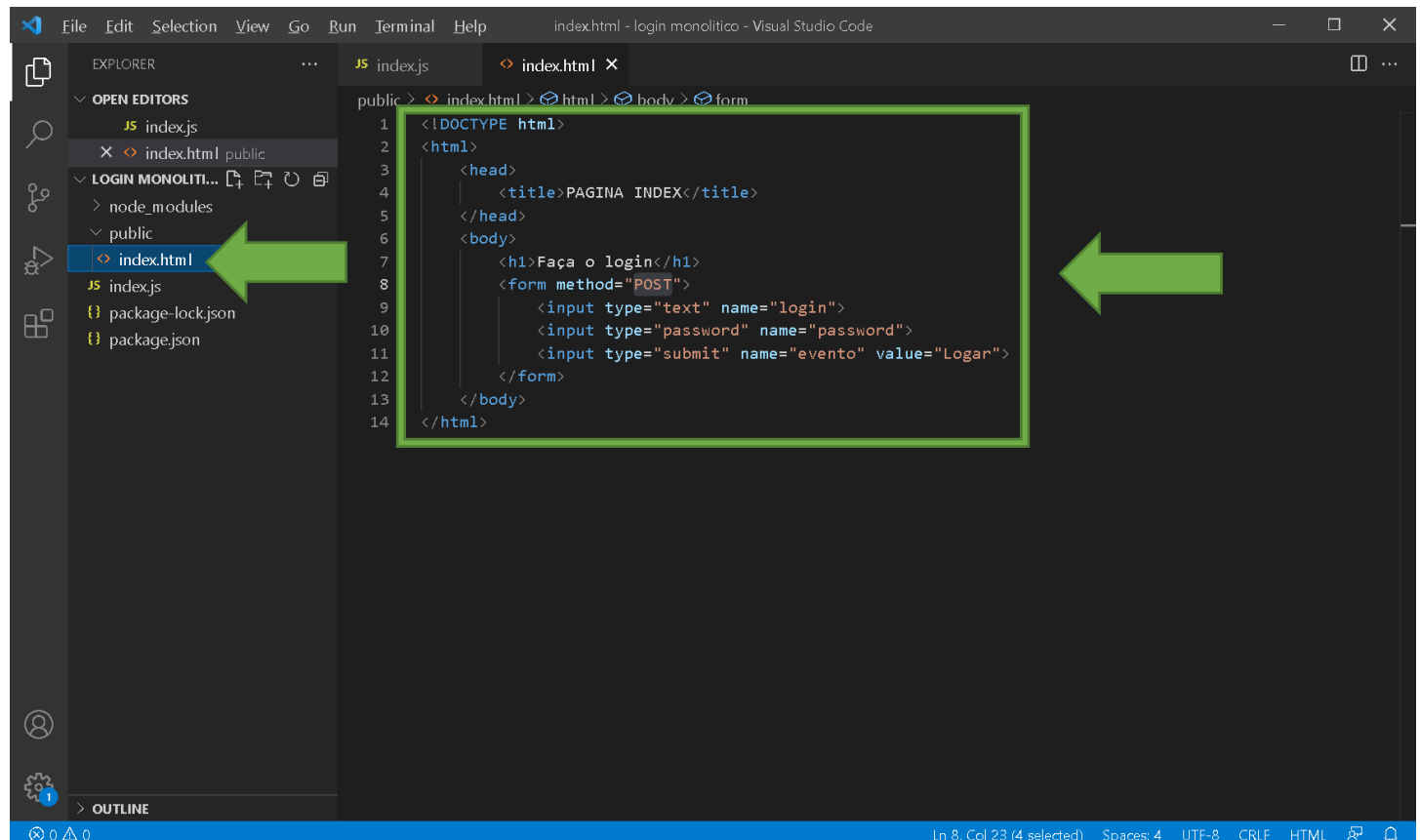
The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar is expanded to show the 'public' folder, and the 'index.html' file is highlighted. A green arrow points to the 'index.html' file in the Explorer. The main editor area shows the content of 'index.html'.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>PÁGINA INDEX</title>
5   </head>
6   <body>
7     <h1>Estamos na página index</h1>
8   </body>
9 </html>
```

⇒ Digite no endereço do navegador “http://localhost:3000/”:

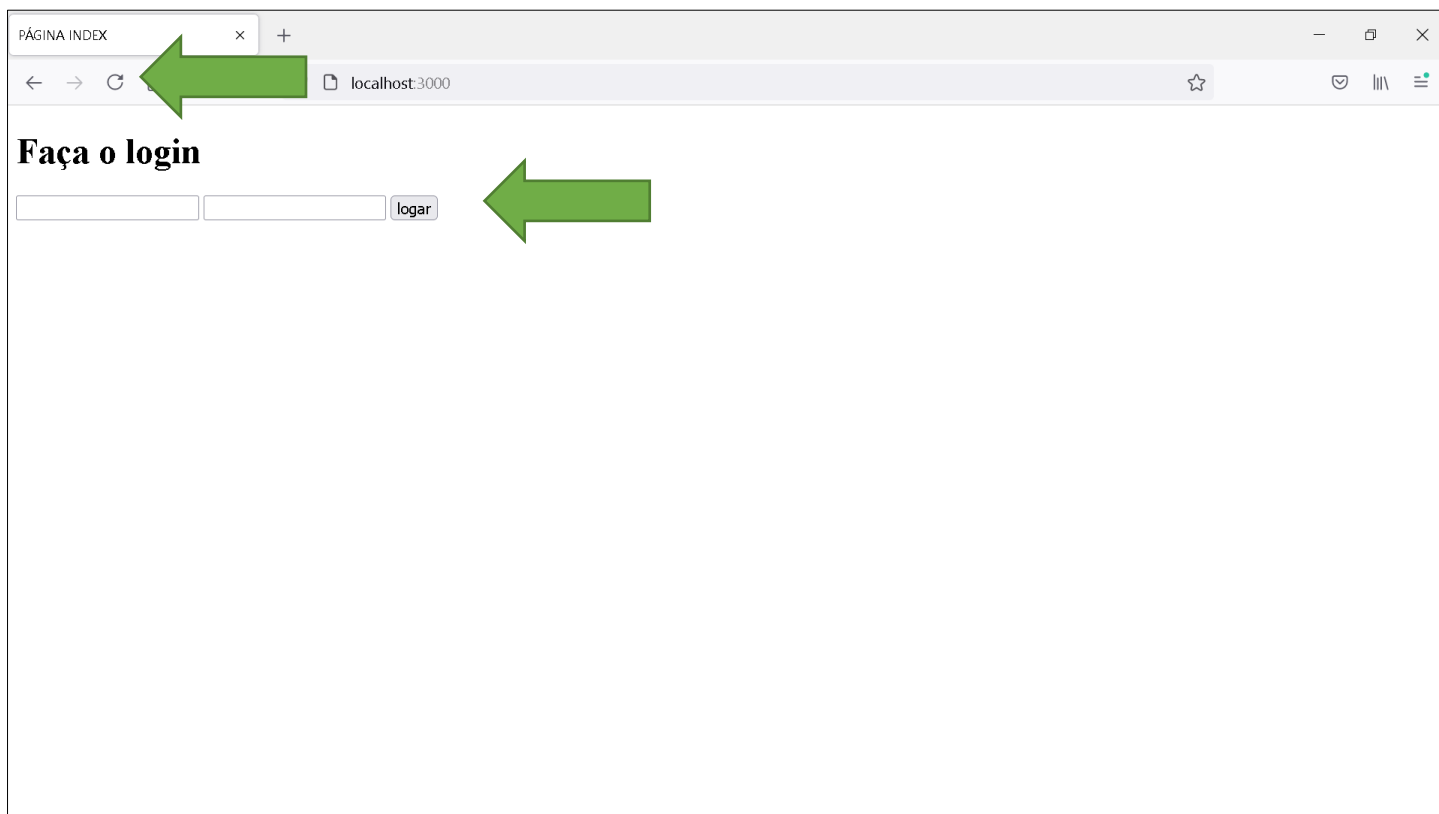


⇒ Ajuste o código da página index.html e salve:





⇒ Atualize o navegador:



⇒ Pare o servidor e instale o pacote ejs (<https://ejs.co/>), que auxilia na renderização das páginas:

```
C:\WINDOWS\system32\cmd.exe

"version": "1.0.0",
"description": "",
"main": "index.js",
"dependencies": {
  "express": "^4.17.1"
},
"devDependencies": {},
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"author": "",
"license": "ISC"
}

Is this OK? (yes)

C:\Programação\Javascript\webserver navegacao>node index.js
servidor rodando em localhost:3000
^C
C:\Programação\Javascript\webserver navegacao>npm install ejs --save
npm WARN webserver-navegacao@1.0.0 No description
npm WARN webserver-navegacao@1.0.0 No repository field.

+ ejs@3.1.6
added 15 packages from 8 contributors and audited 65 packages in 2.669s
found 0 vulnerabilities

C:\Programação\Javascript\webserver navegacao>
```

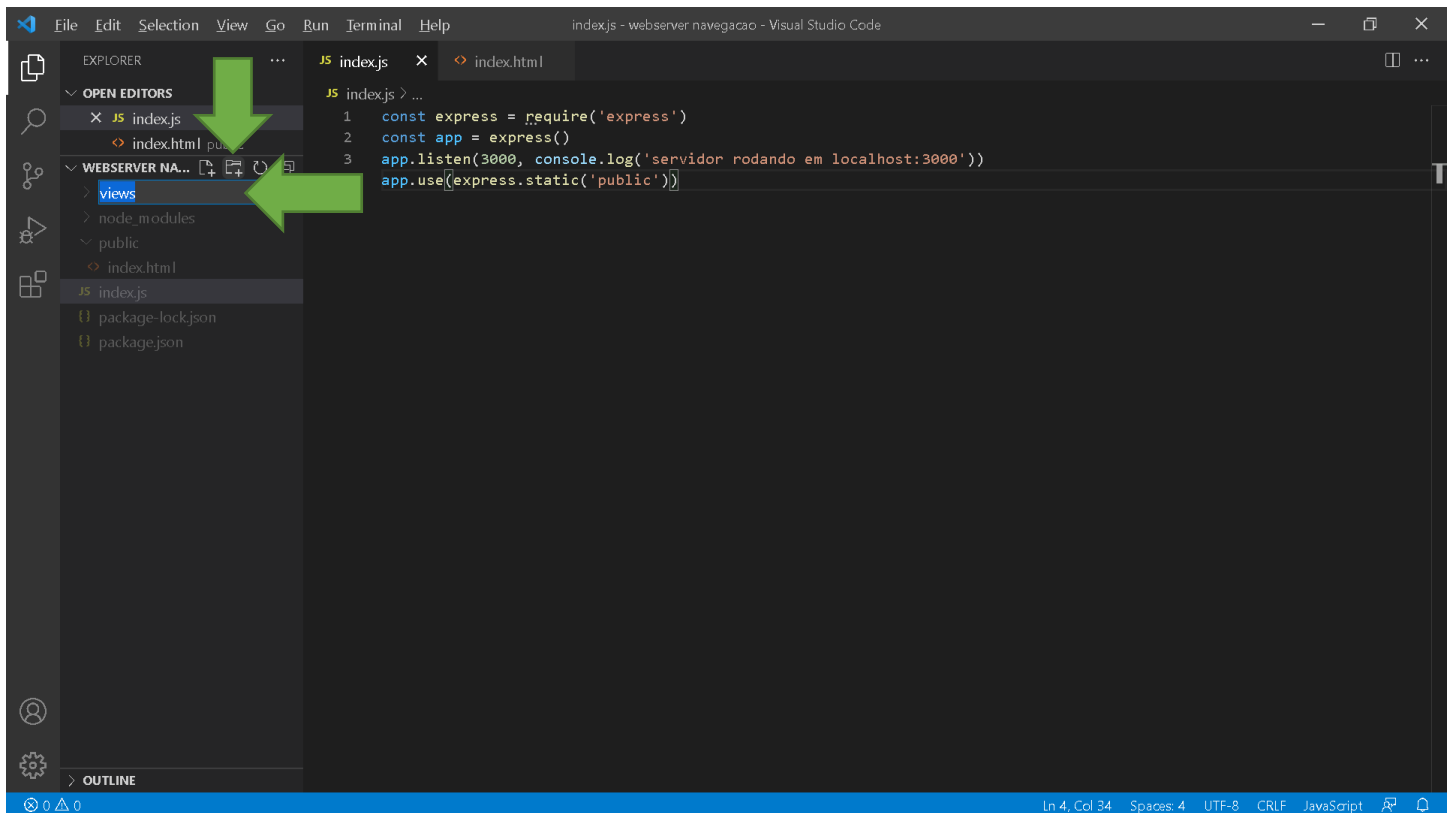
⇒ Instale o pacote path para redefinir a pasta de visualização das páginas:

```
C:\WINDOWS\system32\cmd.exe
ayer.js:95:5)
  at C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:281:22
  at Function.process_params (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:3
35:12)
  at next (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:275:10)
Error: Failed to lookup view "index" in views directory "C:\Programação\Javascript\webserver navegacao\views"
  at Function.render (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\application.js:580:17)
  at ServerResponse.render (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\response.js:1012:7)
  at C:\Programação\Javascript\webserver navegacao\index.js:6:9
  at Layer.handle [as handle_request] (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\l
ayer.js:95:5)
  at next (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\route.js:137:13)
  at Route.dispatch (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\route.js:112:3)
  at Layer.handle [as handle_request] (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\l
ayer.js:95:5)
  at C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:281:22
  at Function.process_params (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:3
35:12)
  at next (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:275:10)
^C
C:\Programação\Javascript\webserver navegacao>npm install path --save
npm WARN webserver-navegacao@1.0.0 No description
npm WARN webserver-navegacao@1.0.0 No repository field.

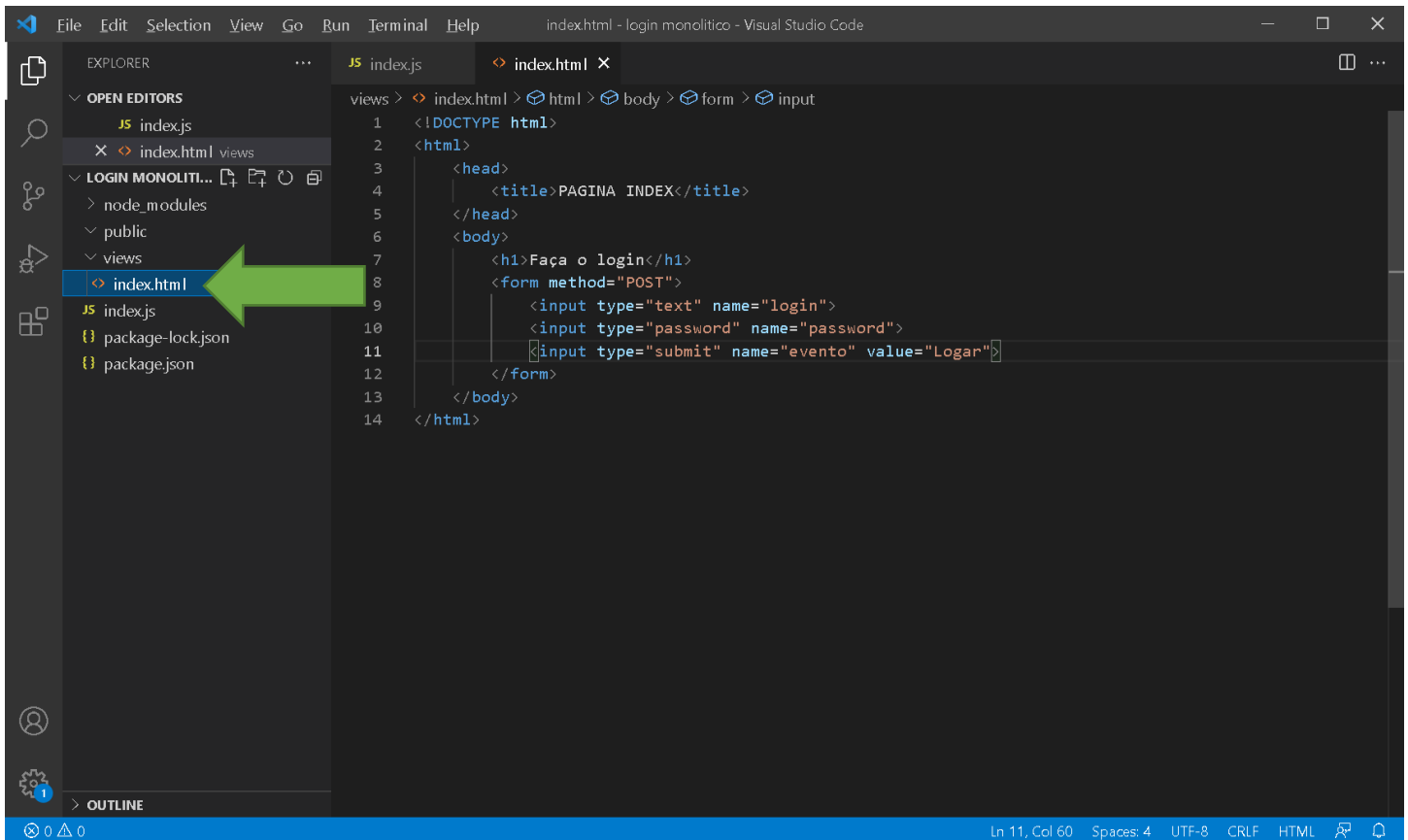
+ path@0.12.7
added 3 packages from 2 contributors and audited 68 packages in 1.146s
found 0 vulnerabilities

C:\Programação\Javascript\webserver navegacao>
```

⇒ Crie a pasta “views” na pasta raiz do projeto por que o EJS padroniza apontar para essa pasta:



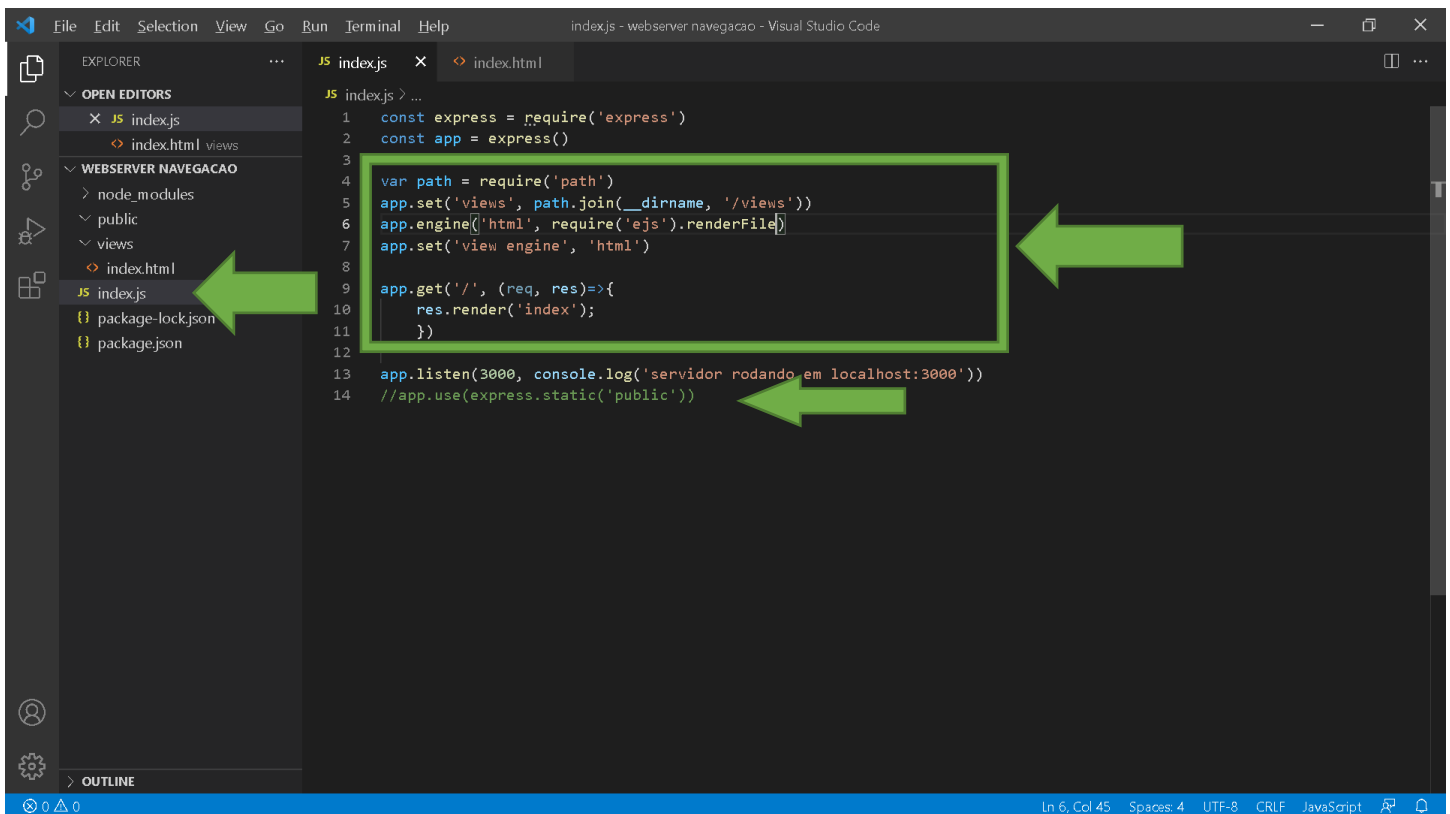
⇒ Mova o arquivo index.html da pasta “public” para a pasta “views”:



index.html - login monolitico - Visual Studio Code

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>PAGINA INDEX</title>
5   </head>
6   <body>
7     <h1>Faça o login</h1>
8     <form method="POST">
9       <input type="text" name="login">
10      <input type="password" name="password">
11      <input type="submit" name="evento" value="Logar">
12    </form>
13  </body>
14 </html>
```

⇒ Ajuste o código do arquivo index.js:



index.js - webserver navegacao - Visual Studio Code

```
1 const express = require('express')
2 const app = express()
3
4 var path = require('path')
5 app.set('views', path.join(__dirname, '/views'))
6 app.engine('html', require('ejs').renderFile)
7 app.set('view engine', 'html')
8
9 app.get('/', (req, res)=>{
10   res.render('index');
11 })
12
13 app.listen(3000, console.log('servidor rodando em localhost:3000'))
14 //app.use(express.static('public'))
```

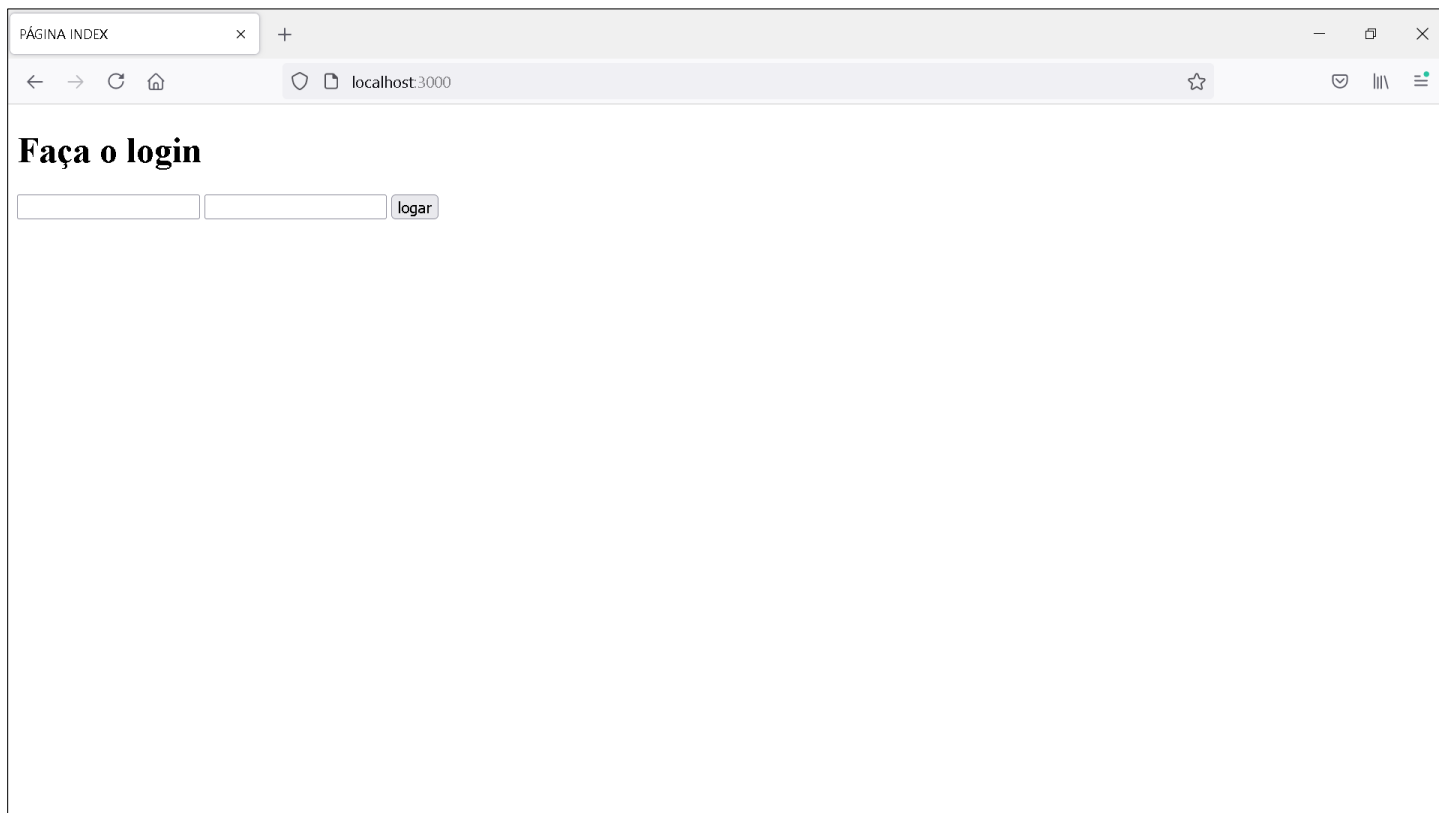
⇒ Rode o servidor novamente:

```
C:\WINDOWS\system32\cmd.exe - node index.js
    at Function.process_params (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:35:12)
    at next (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:275:10)
Error: Failed to lookup view "index" in views directory "C:\Programação\Javascript\webserver navegacao\views"
    at Function.render (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\application.js:580:17)
    at ServerResponse.render (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\response.js:1012:7)
    at C:\Programação\Javascript\webserver navegacao\index.js:6:9
    at Layer.handle [as handle_request] (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\layer.js:95:5)
    at next (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\route.js:137:13)
    at Route.dispatch (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\route.js:112:3)
    at Layer.handle [as handle_request] (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\layer.js:95:5)
    at C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:281:22
    at Function.process_params (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:35:12)
    at next (C:\Programação\Javascript\webserver navegacao\node_modules\express\lib\router\index.js:275:10)
^C
C:\Programação\Javascript\webserver navegacao>npm install path --save
npm WARN webserver-navegacao@1.0.0 No description
npm WARN webserver-navegacao@1.0.0 No repository field.

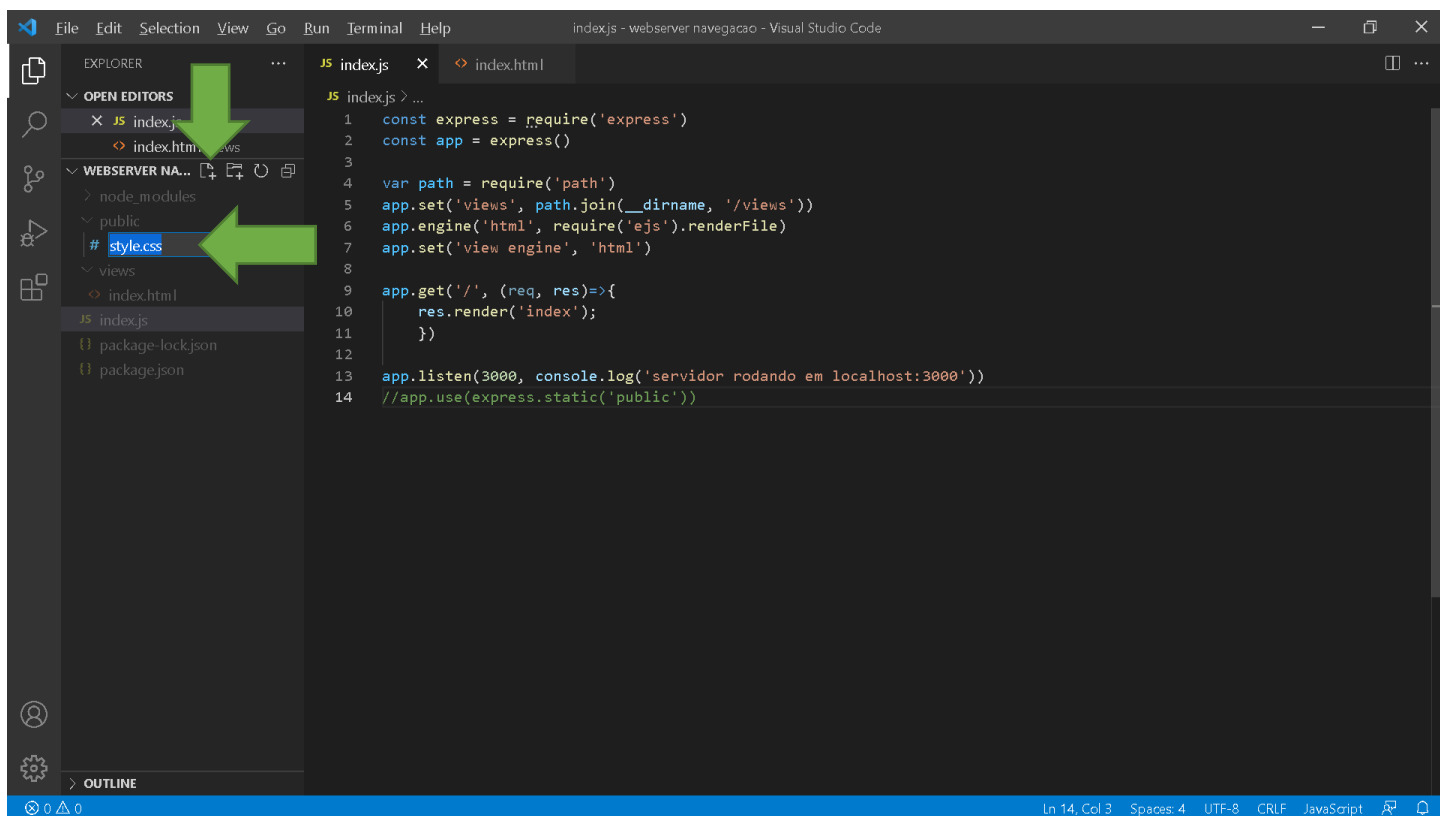
+ path@0.12.7
added 3 packages from 2 contributors and audited 68 packages in 1.146s
found 0 vulnerabilities

C:\Programação\Javascript\webserver navegacao>node index.js
servidor rodando em localhost:3000
```

⇒ Visualize a página novamente:



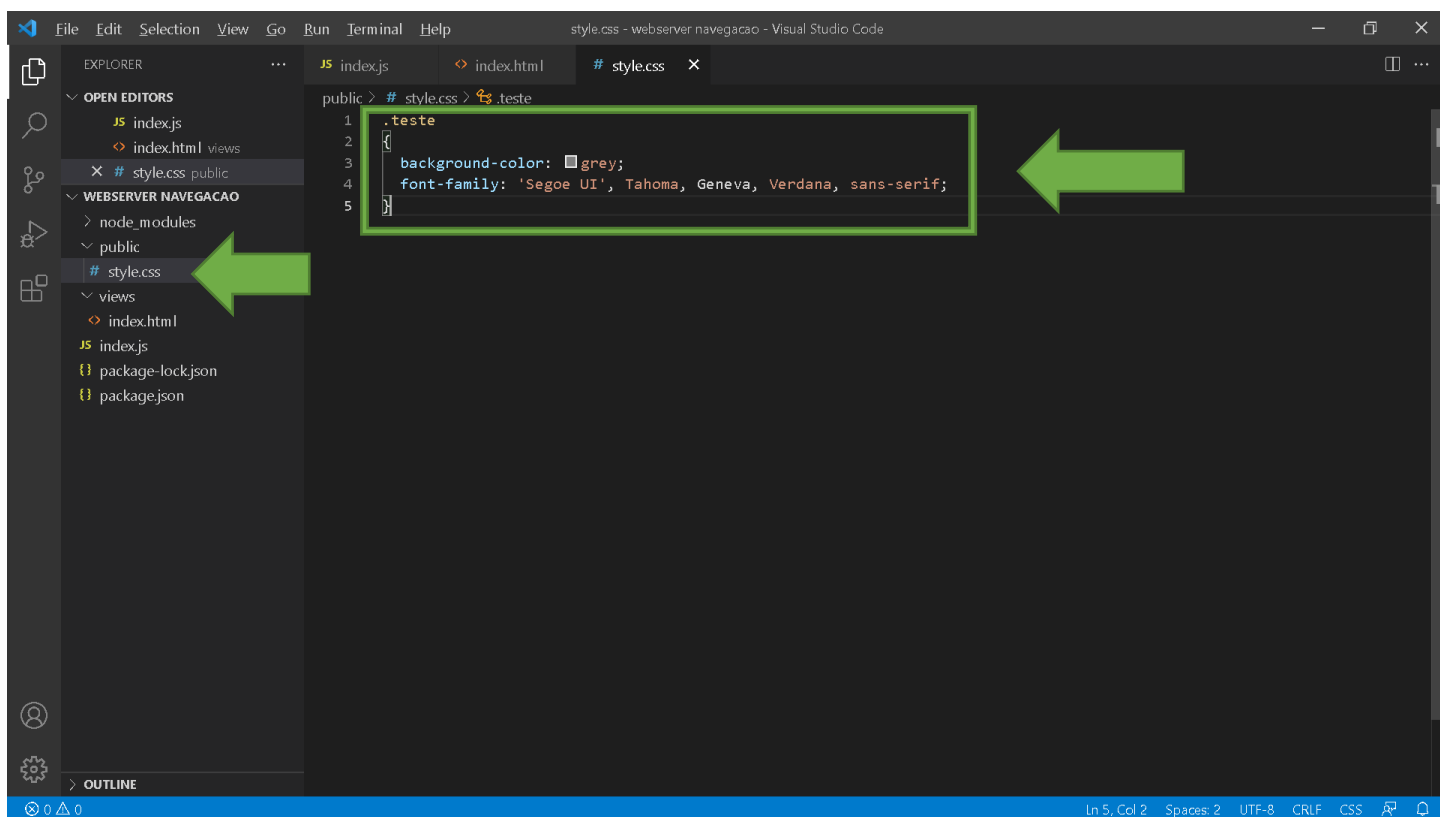
⇒ Crie o arquivo “style.css” na pasta public:



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar is expanded to show the 'public' folder, and a new file named 'style.css' has been created. A green arrow points to the 'style.css' file in the Explorer. The main editor area shows the 'index.js' file with the following code:

```
1 const express = require('express')
2 const app = express()
3
4 var path = require('path')
5 app.set('views', path.join(__dirname, '/views'))
6 app.engine('html', require('ejs').renderFile)
7 app.set('view engine', 'html')
8
9 app.get('/', (req, res) => {
10   res.render('index')
11 })
12
13 app.listen(3000, console.log('servidor rodando em localhost:3000'))
14 //app.use(express.static('public'))
```

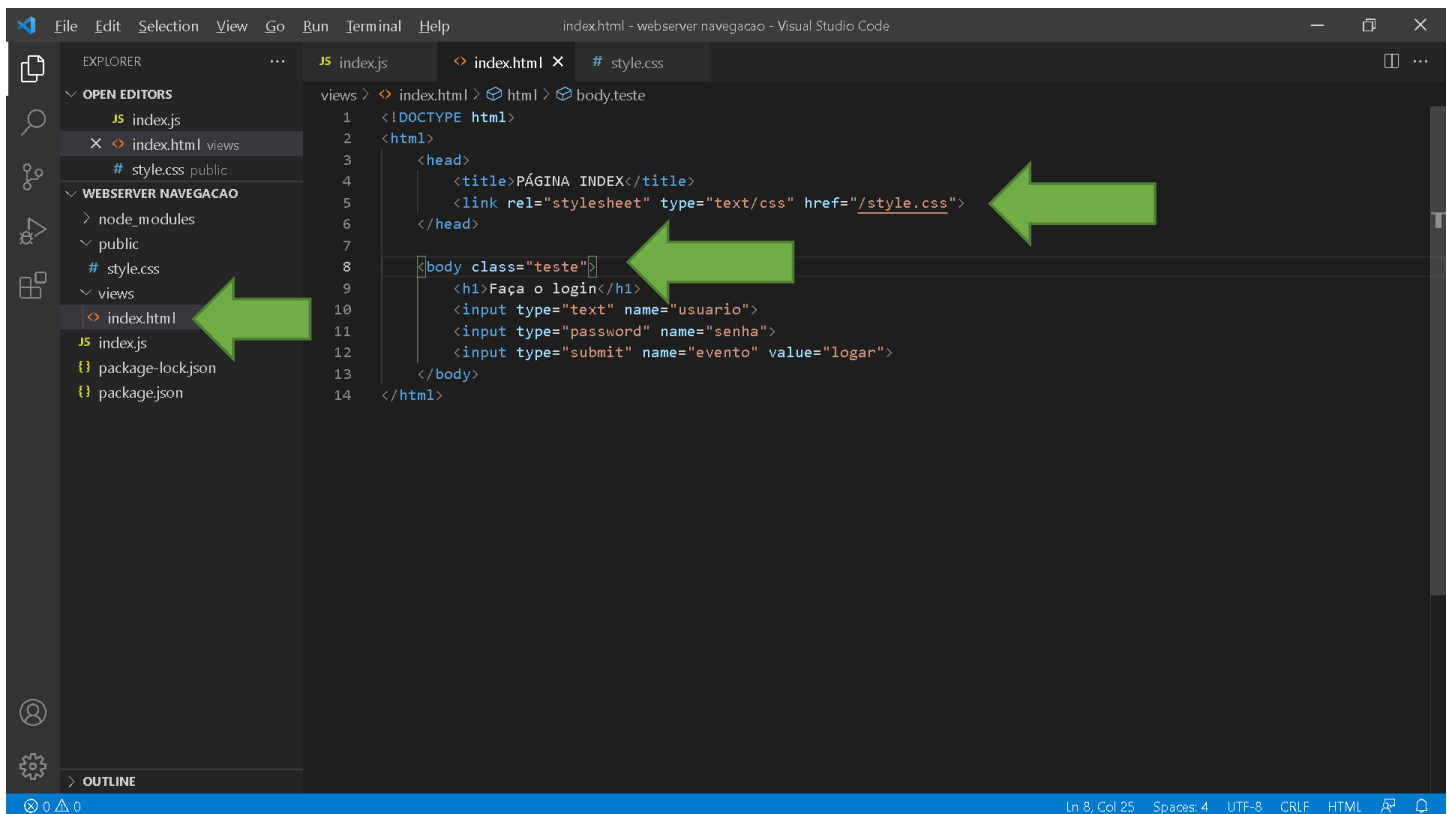
⇒ Tente alguns ajustes no arquivo “style.css”:



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer sidebar is expanded to show the 'public' folder, and the 'style.css' file is selected. A green arrow points to the 'style.css' file in the Explorer. The main editor area shows the 'style.css' file with the following code:

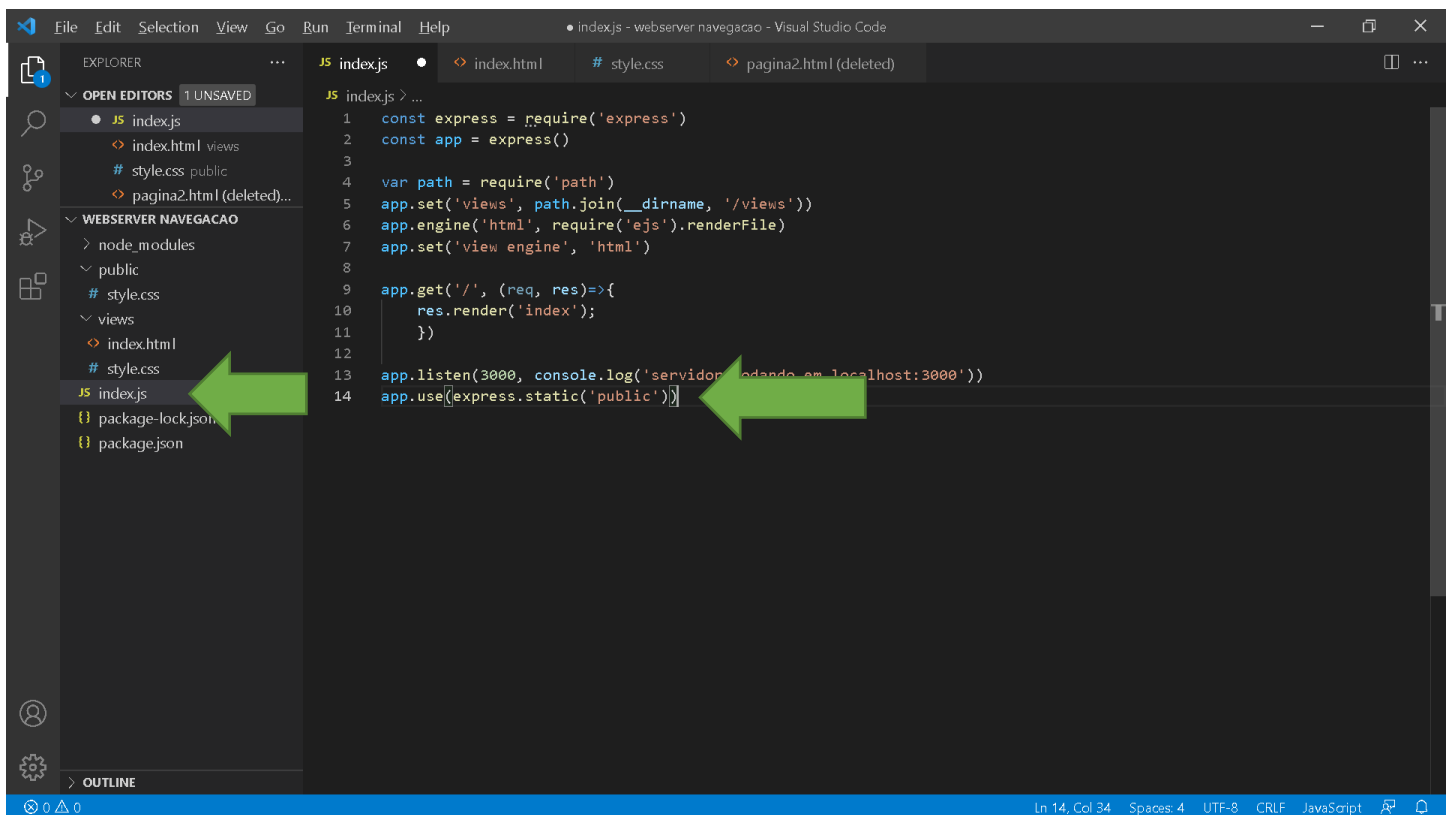
```
public > # style.css > .teste
1 .teste
2 {
3   background-color: grey;
4   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
5 }
```

⇒ Realize os seguintes ajustes no arquivo “index.html”:



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>PÁGINA INDEX</title>
5     <link rel="stylesheet" type="text/css" href="/style.css">
6   </head>
7
8   <body class="teste">
9     <h1>Faça o login</h1>
10    <input type="text" name="usuario">
11    <input type="password" name="senha">
12    <input type="submit" name="evento" value="logar">
13  </body>
14 </html>
```

⇒ É necessário liberar/descomentar a última linha do arquivo “index.js” para que o arquivo CSS tenha efeito na página:



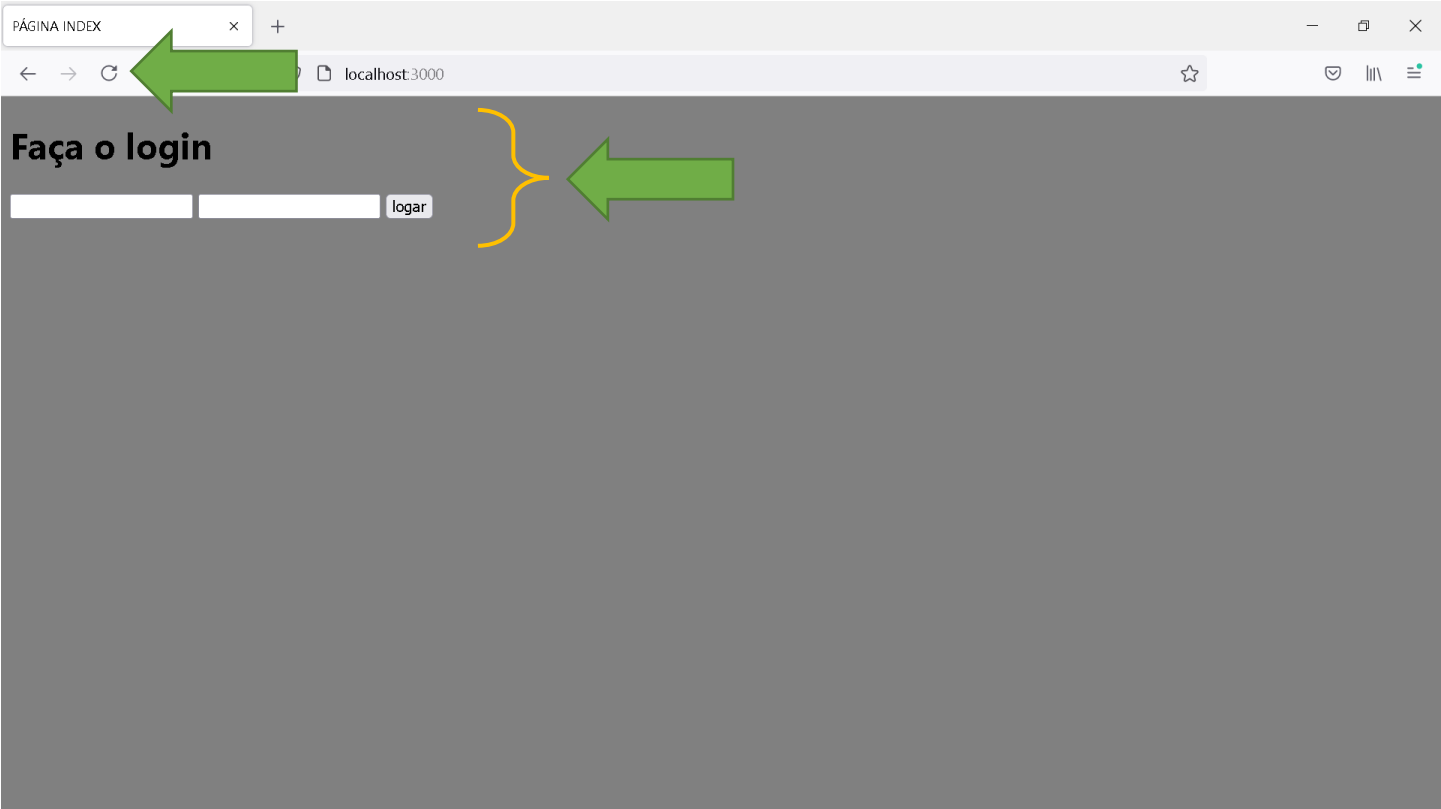
```
1 const express = require('express')
2 const app = express()
3
4 var path = require('path')
5 app.set('views', path.join(__dirname, '/views'))
6 app.engine('html', require('ejs').renderFile)
7 app.set('view engine', 'html')
8
9 app.get('/', (req, res)=>{
10   res.render('index');
11 })
12
13 app.listen(3000, console.log('servidor rodando em localhost:3000'))
14 app.use(express.static('public'))
```

\*\*\* O AJUSTE ACIMA LIBERA OS ARQUIVOS DA PASTA “public” PARA SEREM LIDOS E UTILIZADOS NO PROJETO PARA ENTENDER MELHOR, CONSULTE O LINK <https://expressjs.com/en/starter/static-files.html> SOBRE A LIBERAÇÃO DE ARQUIVOS LOCAIS PARA USO (CSS, IMAGENS, ETC.).

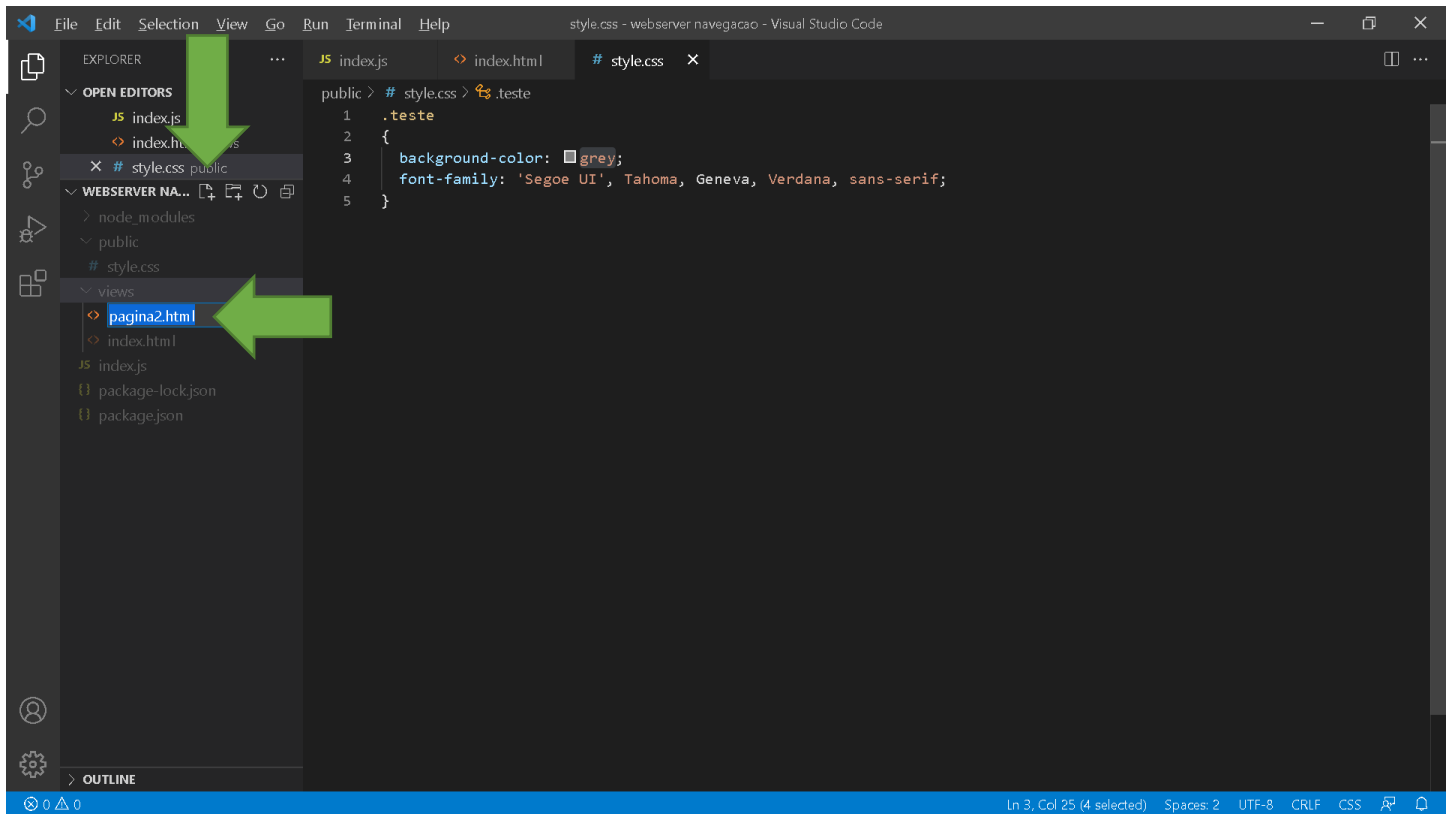
⇒ Pare o servidor e reinicie para que os ajustes no “index.js” surtam efeito:

[illegible]

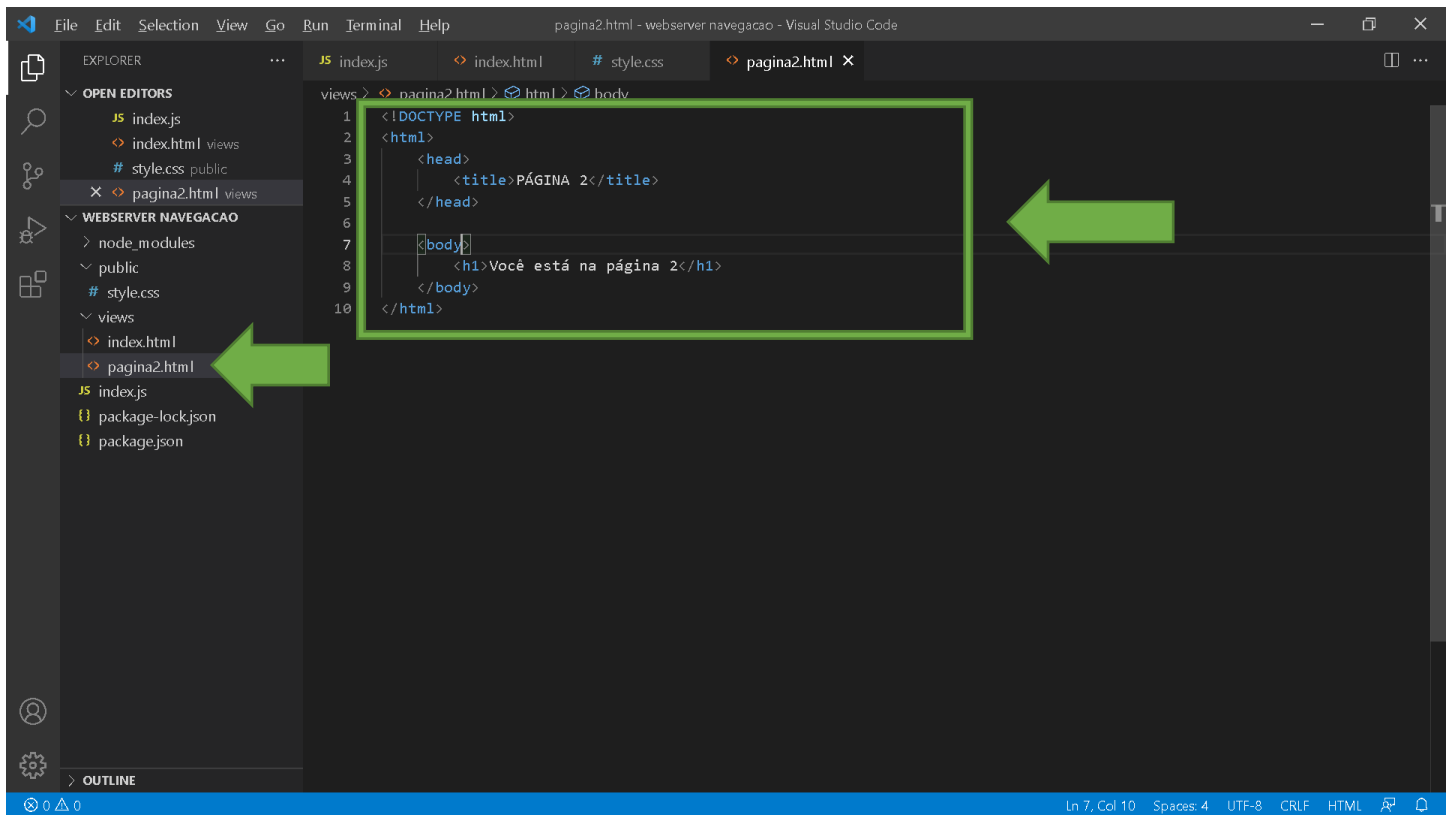
⇒ Atualize e visualize a página novamente:



⇒ Criar outra página HTML na pasta views:

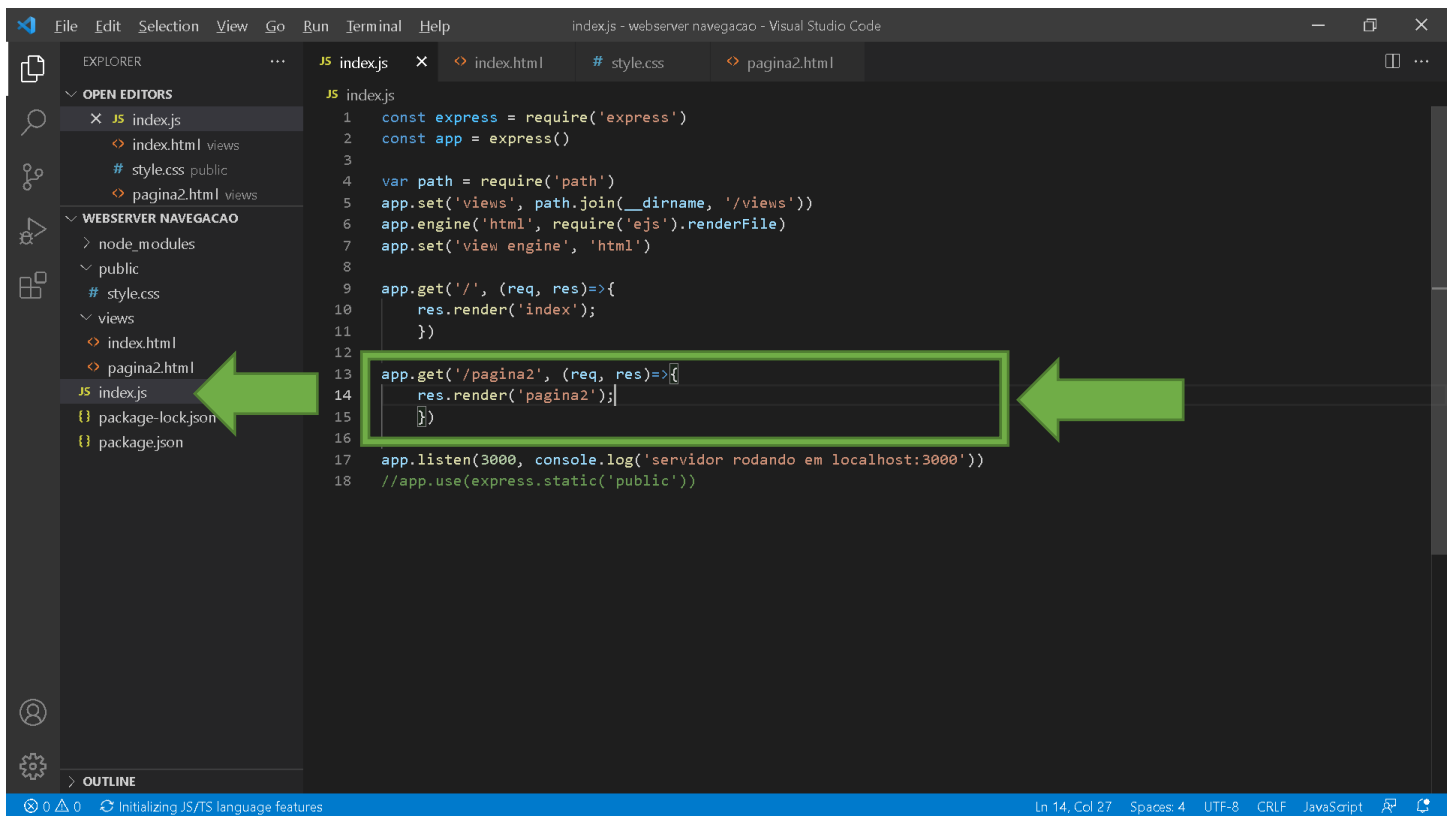


⇒ Realize os ajustes abaixo no arquivo “pagina2.html”:



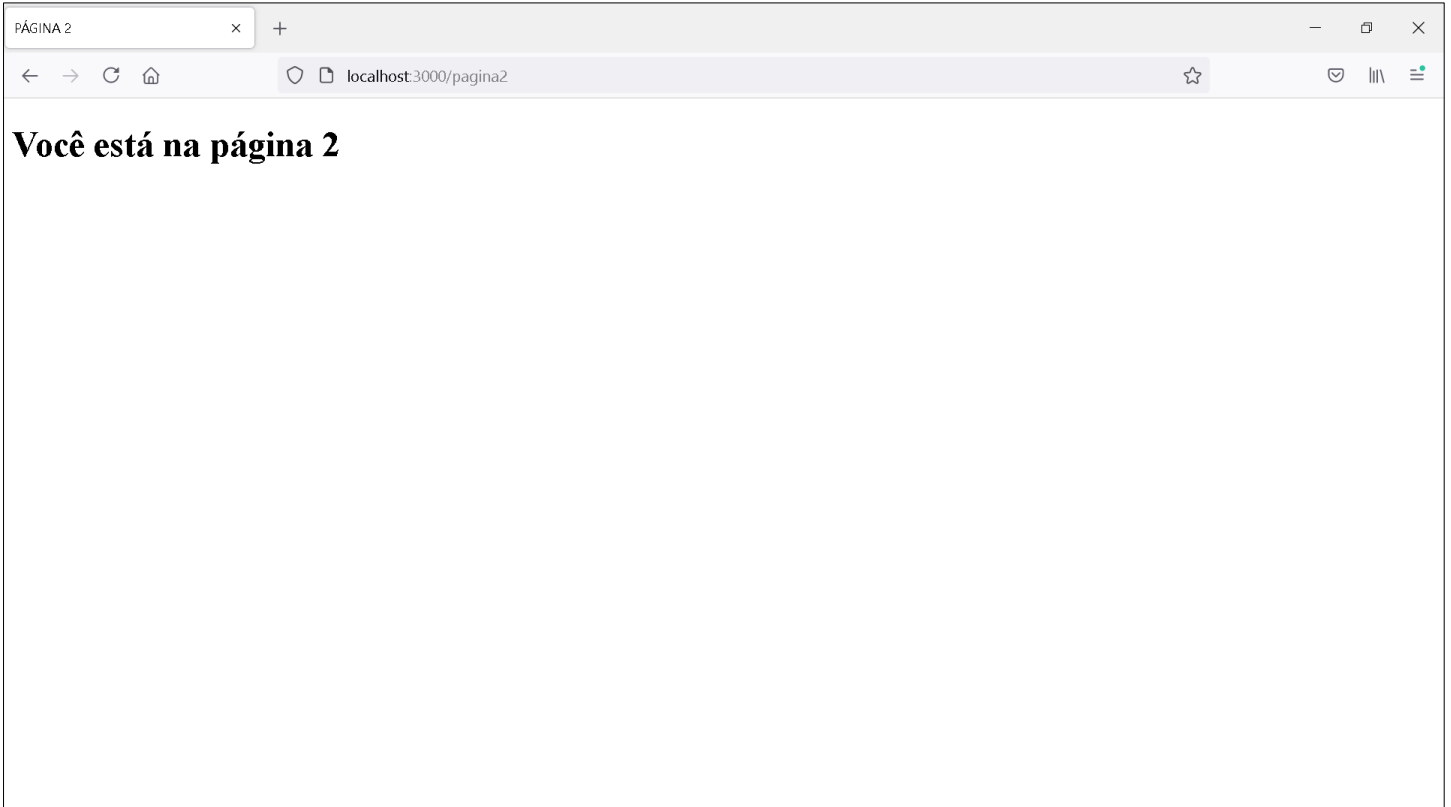


⇒ Ajuste o código do arquivo index.js:

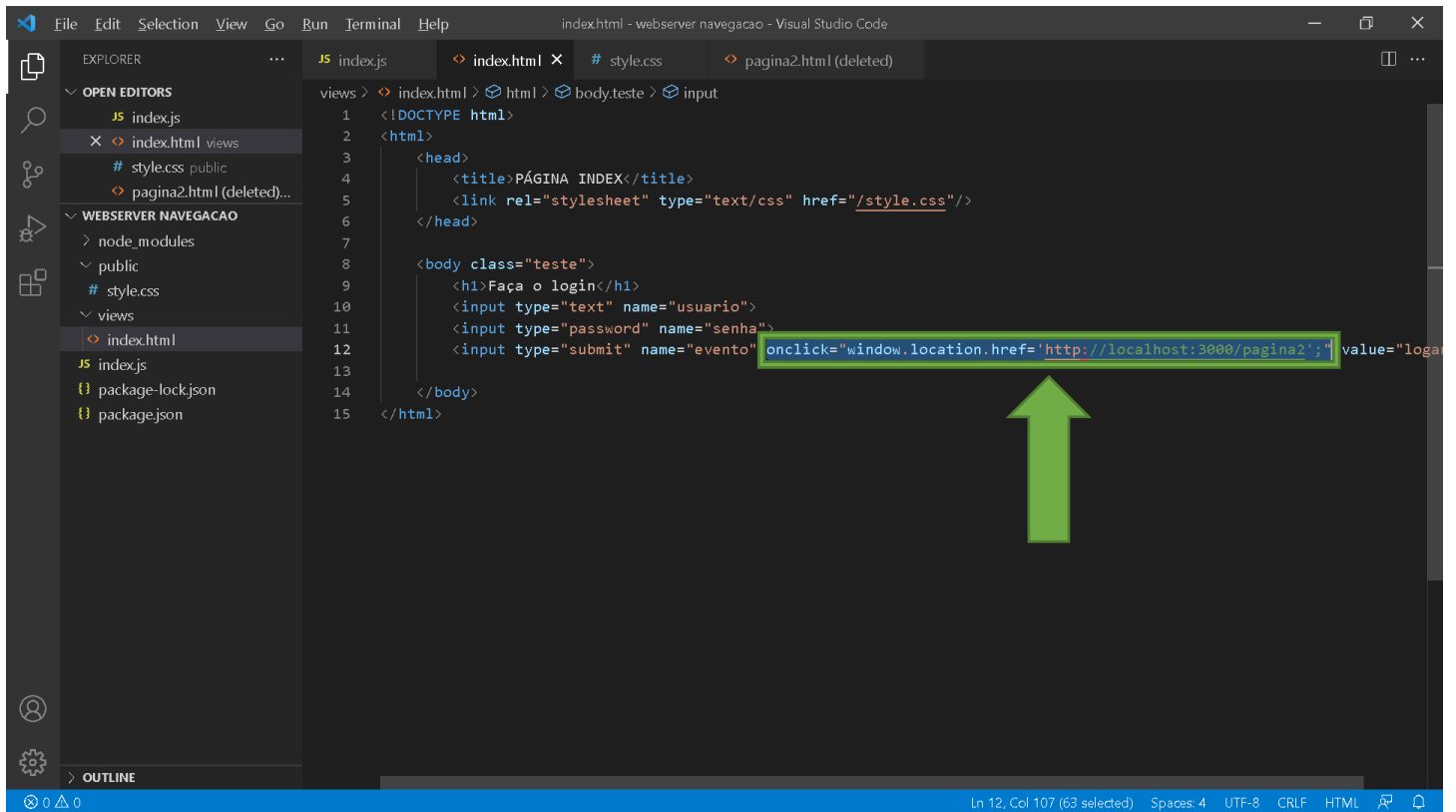


```
1 const express = require('express')
2 const app = express()
3
4 var path = require('path')
5 app.set('views', path.join(__dirname, '/views'))
6 app.engine('html', require('ejs').renderFile)
7 app.set('view engine', 'html')
8
9 app.get('/', (req, res) => {
10   res.render('index');
11 })
12
13 app.get('/pagina2', (req, res) => {
14   res.render('pagina2');
15 })
16
17 app.listen(3000, console.log('servidor rodando em localhost:3000'))
18 //app.use(express.static('public'))
```

⇒ Digite o endereço da página2 <http://localhost:3000/pagina2> e visualize a página pelo navegador:

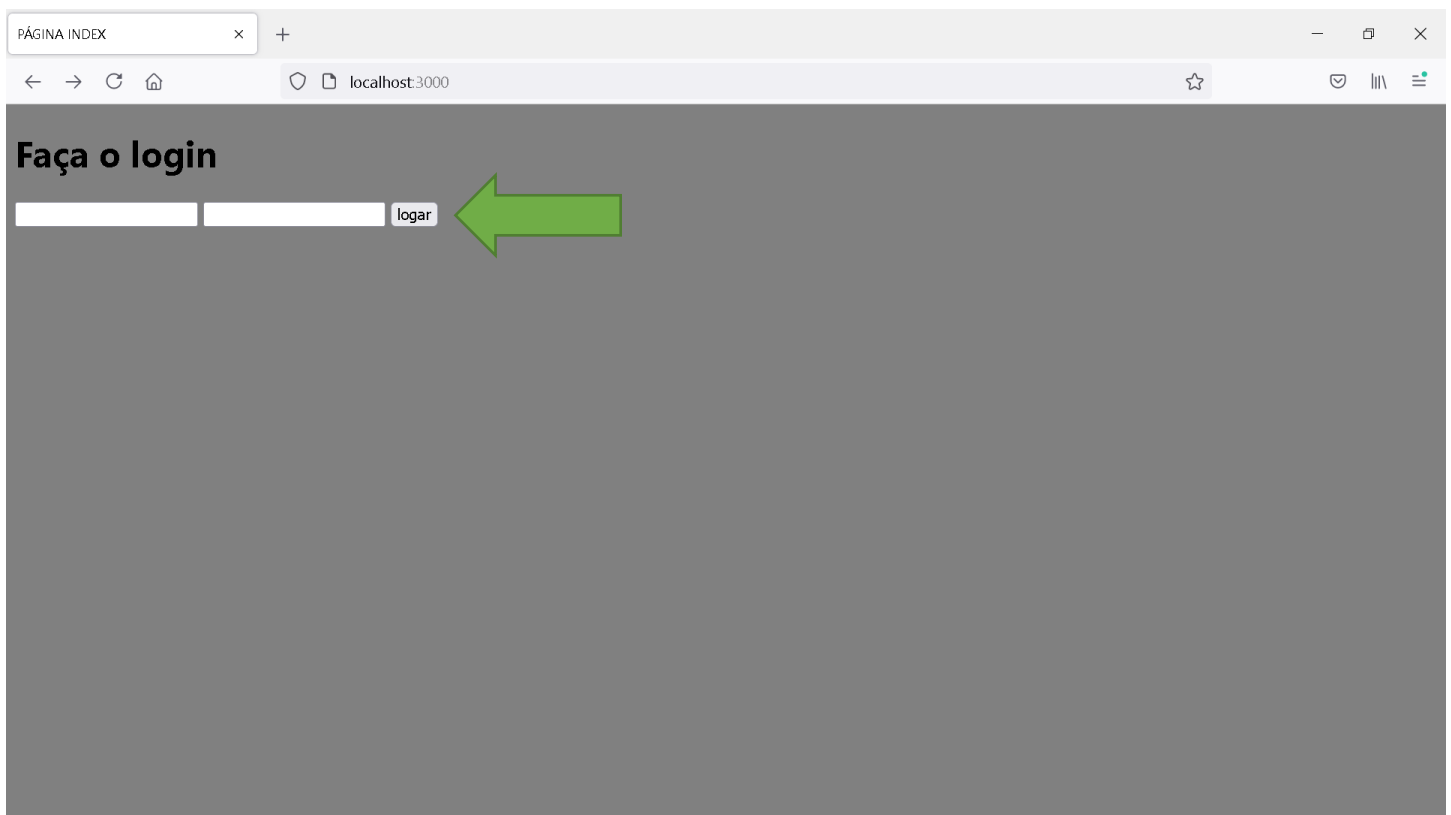


⇒ Adicione um evento ao botão “logar” para navegação:

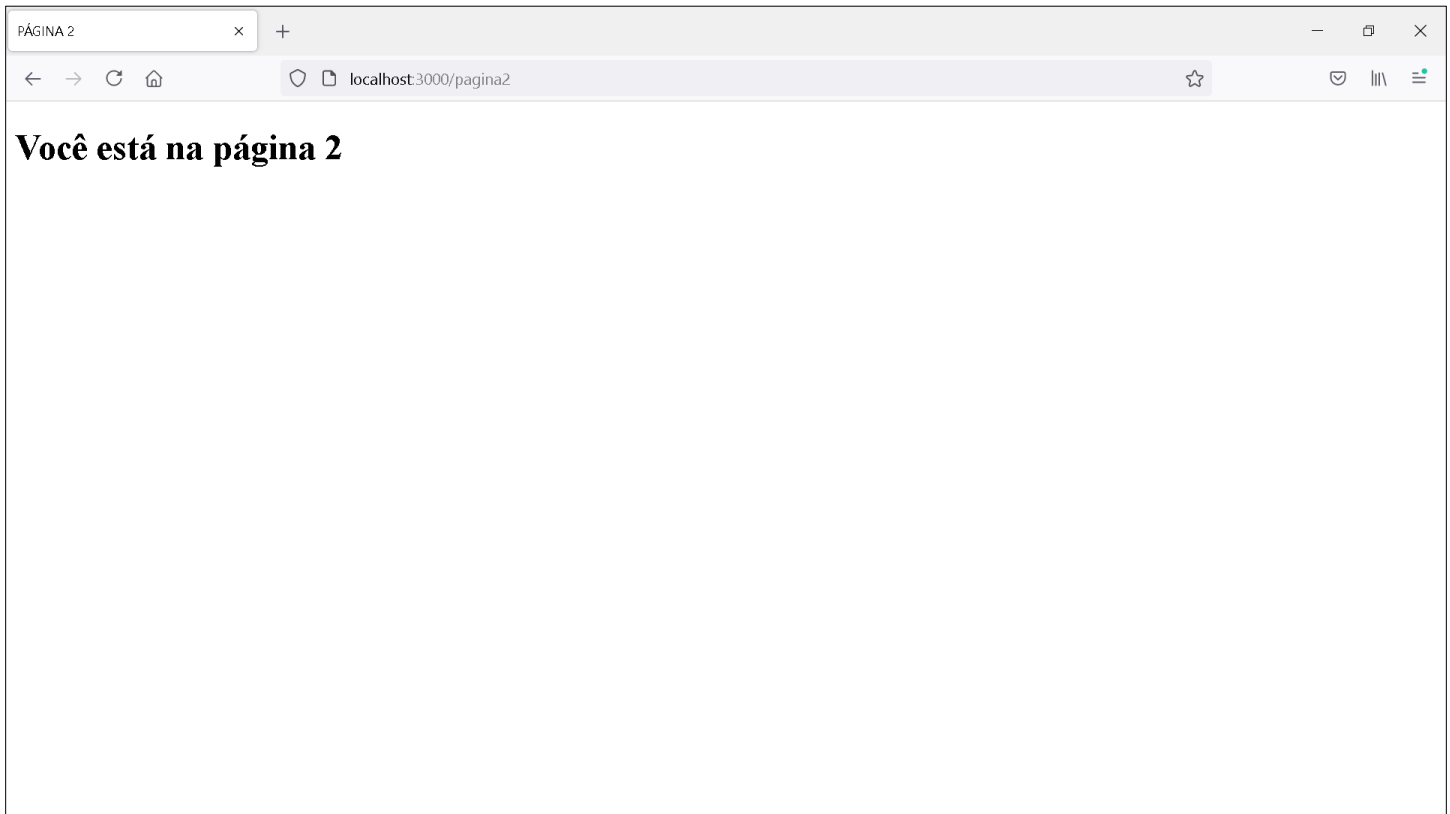


```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>PÁGINA INDEX</title>
5     <link rel="stylesheet" type="text/css" href="/style.css"/>
6   </head>
7
8   <body class="teste">
9     <h1>Faça o login</h1>
10    <input type="text" name="usuario">
11    <input type="password" name="senha">
12    <input type="submit" name="evento" onclick="window.location.href='http://localhost:3000/pagina2';" value="logar" />
13  </body>
14 </html>
```

⇒ Volte para a página inicial (<http://localhost:3000/>) e clique no botão “logar”:



⇒ Você visualizará a “pagina2.html”:



- ⇒ Através da aplicação dos conhecimentos dessa aula, você já é capaz de criar sites básicos com eventos, navegação e recursos visuais (CSS) interessantes em uma “estrutura monolítica”. Esse mesmo site pode ser alocado em qualquer serviço de cloud (AWS, Google Cloud, Hiroku, Hostgator, etc.) que funcionará normalmente.
- ⇒ **Para o desenvolvimento em backend com Node JS, é fortemente sugerido que você utilize a ferramenta Nodemon (<https://www.npmjs.com/package/nodemon>), por que ela identifica qualquer alteração nos arquivos “.js” e já executa as suas atualizações automaticamente, sem a necessidade de ficar e reiniciando o servidor.**