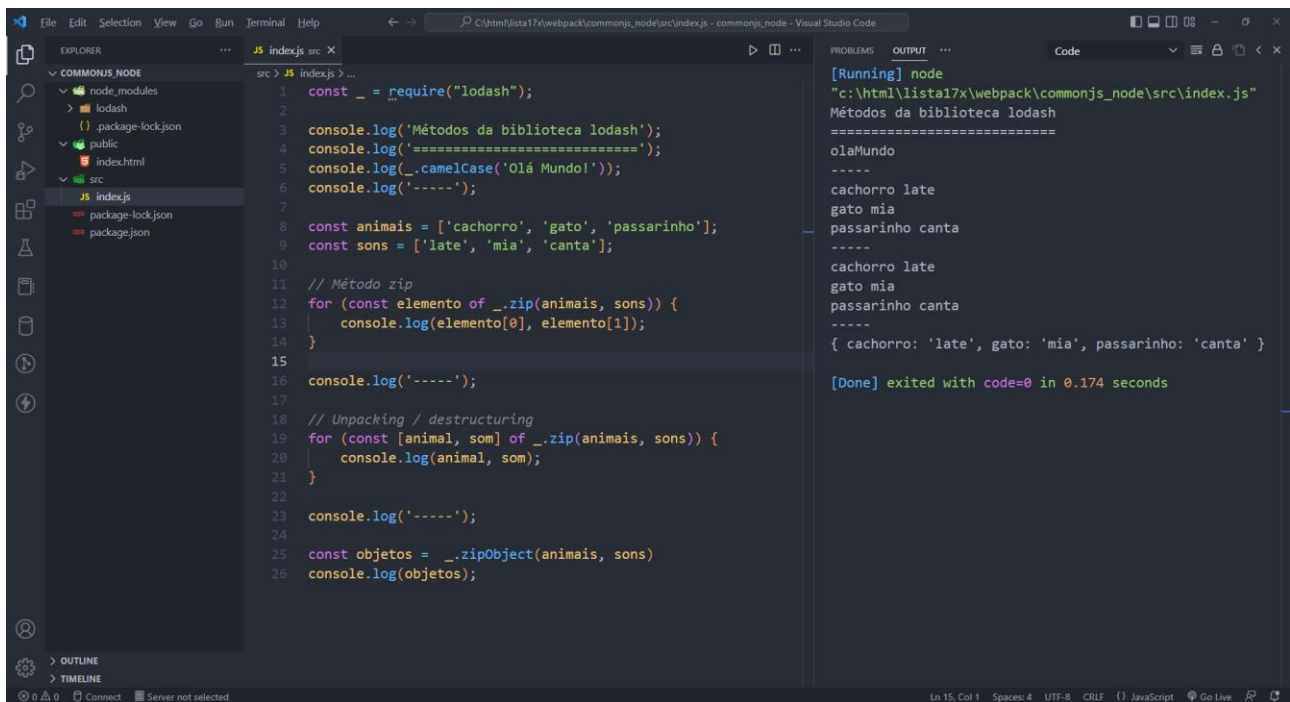


Lista 17d – Webpack

Sistema de módulos CommonJS

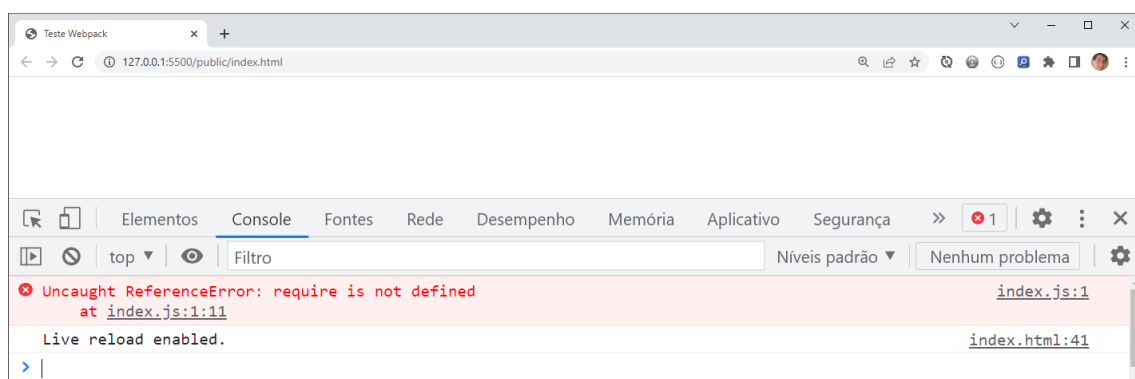
1. Crie uma pasta chamada *webpack*, dentro dela a pasta *commonjs_node* e dentro dela as pastas *public* e *src*.
2. Dentro da pasta *public* crie um arquivo chamado *index.html* e dentro da pasta *src* o arquivo *index.js*.
3. Abra a pasta *commonjs_node* no VS Code e, no terminal, crie um projeto node e instale a biblioteca [lodash](#).
 - `npm init -y`
 - `npm i lodash`
4. Digite o código a seguir e execute o arquivo (*F5* ou *Ctrl + Alt + N*). O código irá funcionar sem problemas.



5. No arquivo *index.html* digite o código abaixo e abra-o no navegador.



6. Ele dará problema pois o sistema de módulos do ES6 (ESM) não entende a função `require()`.

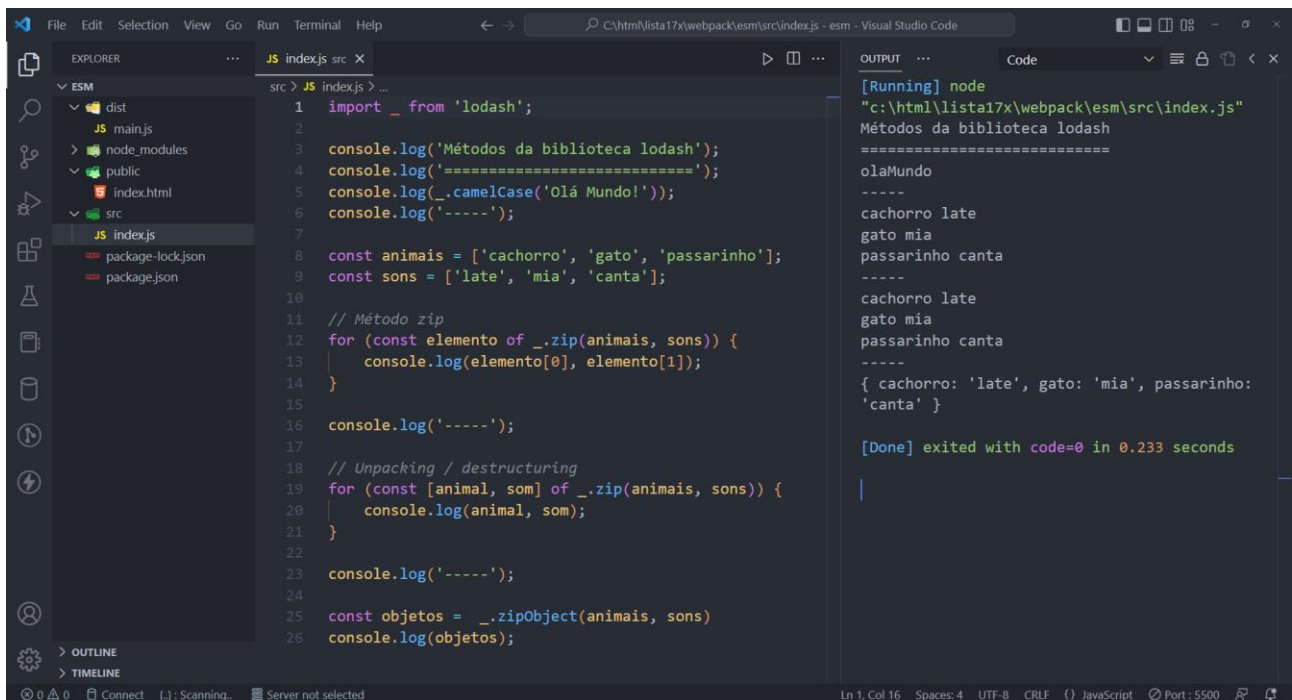


Sistema de módulos ESM

1. Dentro da pasta *webpack*, crie a pasta *esm* e dentro dela as pastas *public* e *src*.
2. Dentro da pasta *public* crie um arquivo chamado *index.html* e dentro da pasta *src* o arquivo *index.js*.
3. Abra a pasta *commonjs_node* no VS Code e, no terminal, crie um projeto node e instale a biblioteca [lodash](#).
 - `npm init -y`
 - `npm i lodash`
4. Altere o arquivo *package.json* como abaixo. A linha 6 indica que o arquivo é um módulo. Outra forma de fazer isso seria retirar a linha 6 e alterar a extensão do arquivo de *index.js* para *index.mjs*; a extensão *mjs* indica ao node.js que o arquivo é um módulo ES6 e, portanto, compatível com o sistema de módulos ESM.

```
esm - package.json
1 {
2   "name": "esm",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "type": "module",
7   "scripts": {
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "dependencies": {
14    "lodash": "^4.17.21"
15  }
16 }
```

5. O código do arquivo *index.js* é igual ao do projeto anterior, exceto a primeira linha. Digite o código e o execute.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows the project structure with folders 'dist', 'node_modules', 'public', and 'src'. The 'src' folder is expanded, showing 'index.js'. The main editor displays the content of 'index.js' with the following code:

```
1 import _ from 'lodash';
2
3 console.log('Métodos da biblioteca lodash');
4 console.log('=====');
5 console.log(_.camelCase('Olá Mundo!'));
6 console.log('-----');
7
8 const animais = ['cachorro', 'gato', 'passarinho'];
9 const sons = ['late', 'mia', 'cantar'];
10
11 // Método zip
12 for (const elemento of _.zip(animais, sons)) {
13   console.log(elemento[0], elemento[1]);
14 }
15
16 console.log('-----');
17
18 // Unpacking / destructuring
19 for (const [animal, som] of _.zip(animais, sons)) {
20   console.log(animal, som);
21 }
22
23 console.log('-----');
24
25 const objetos = _.zipObject(animais, sons)
26 console.log(objetos);
```

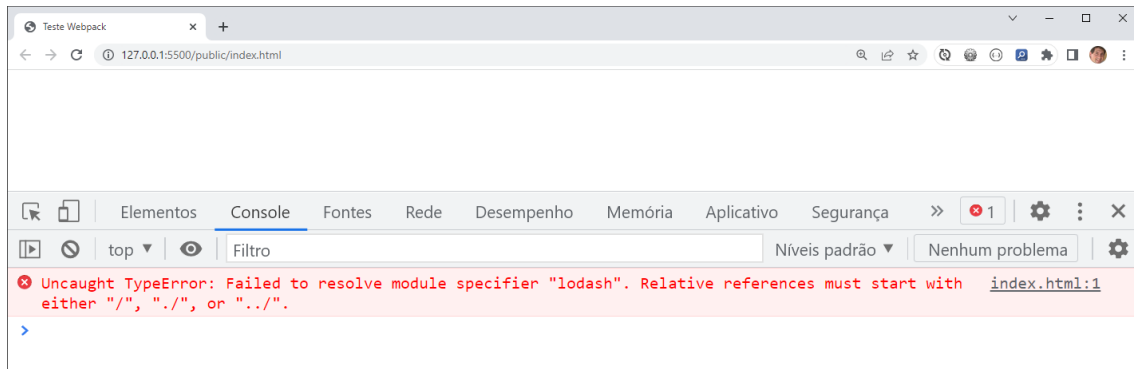
On the right, the Output pane shows the execution results of the command `node "c:\html\lista17x\webpack\esm\src\index.js"`. The output is as follows:

```
[Running] node
"c:\html\lista17x\webpack\esm\src\index.js"
Métodos da biblioteca lodash
=====
olaMundo
-----
cachorro late
gato mia
passarinho cantar
-----
cachorro late
gato mia
passarinho cantar
-----
{ cachorro: 'late', gato: 'mia', passarinho: 'cantar' }
[Done] exited with code=0 in 0.233 seconds
```

6. O arquivo *index.html* é idêntico ao do sistema anterior exceto a linha 9. Digite o código e abra-o no Live Server.

```
esm - index.html
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Teste Webpack</title>
8
9   <script type="module" src="../src/index.js"></script>
10 </head>
11 <body>
12
13 </body>
14 </html>
```

7. Mesmo com o sistema de módulos do ES6 ele não irá funcionar, pois não encontrará a biblioteca.

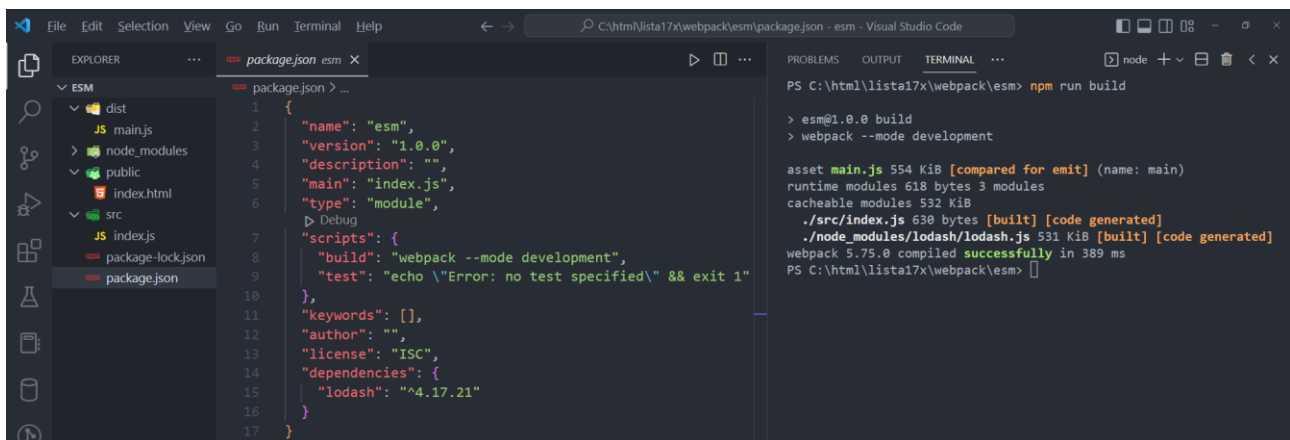


8. Digite o comando abaixo no terminal para incluir a dependência do webpack.

- `npm i -D webpack-cli webpack`

9. Modifique o arquivo `package.json` (linha 10) para incluir um `script` de `build` e no terminal execute o comando:

- `npm run build`



Dentro da pasta `dist` será gerado o pacote (*bundle*) `main.js`.

10. Modifique agora o arquivo `index.html` de forma a carregar o `bundle`. Agora deve funcionar no navegador.

