

Hello World!,

Meu nome é Robson Gil Rocha é um prazer participar deste teste.

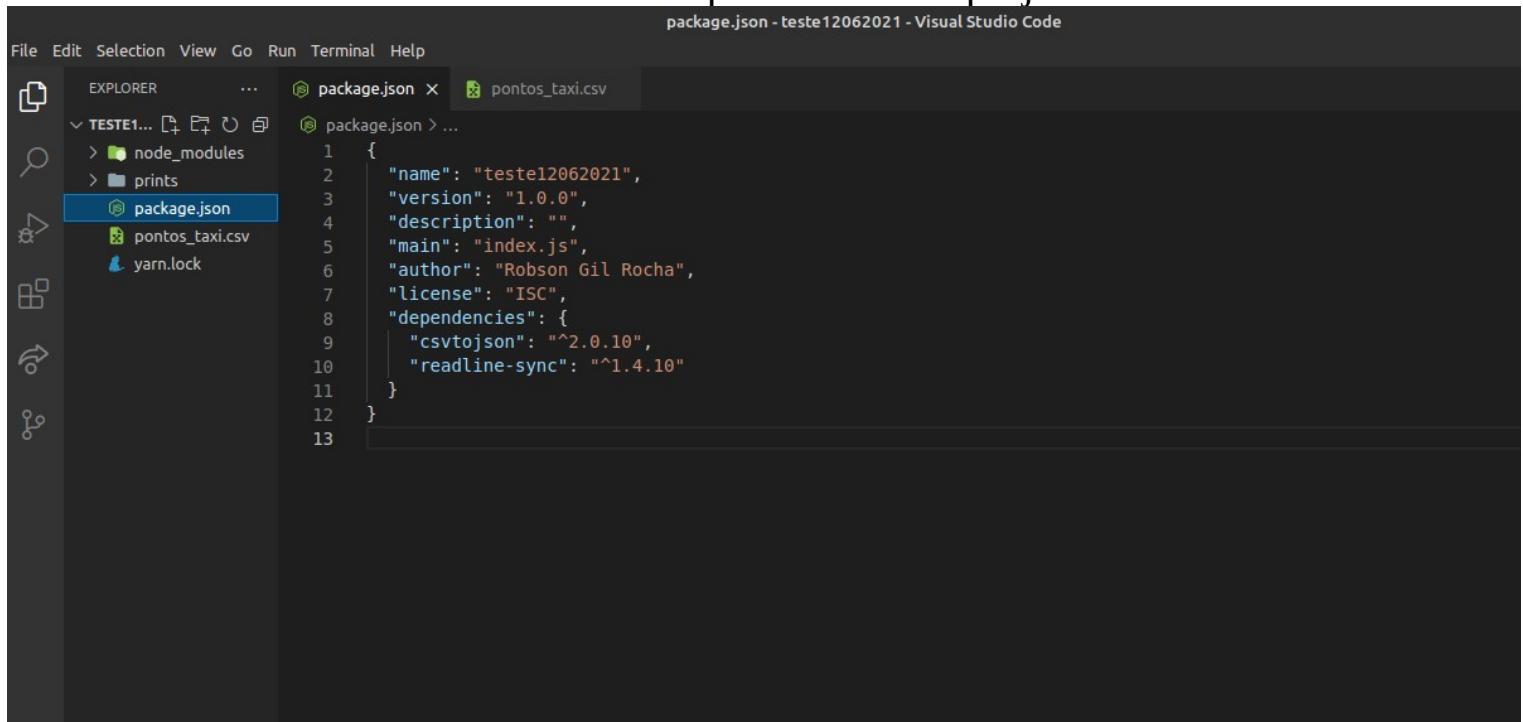
Projeto na linguagem javascript foi a escolhida por minha atual afinidade.

Usando o programa VScode no sistema operacional Linux Mint.

O link do GitHub : https://github.com/RobsonGilRocha/teste_12062021

npm init	Iniciando o projeto.
yarn add readline-sync	Adicionando biblioteca para ler os inputs do usuário via terminal.
yarnr add csvtojson	Adicionando biblioteca para ler arquivos CSV.

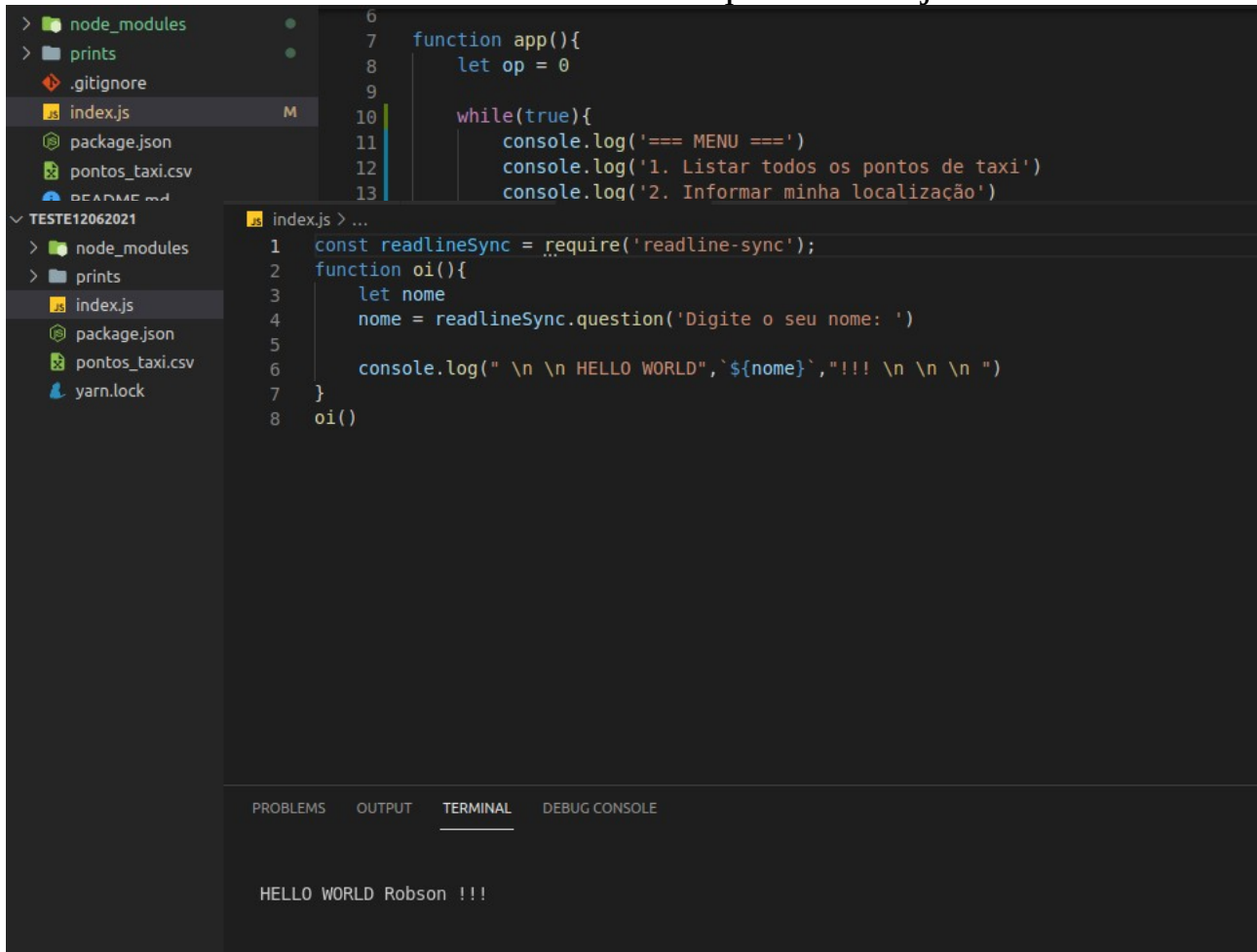
Adicionando arquivos CSV no projeto.

A screenshot of the Visual Studio Code editor interface. The title bar at the top reads "package.json - teste12062021 - Visual Studio Code". The Explorer sidebar on the left shows a project named "TESTE1..." with subfolders "node_modules" and "prints", and files "package.json", "pontos_taxi.csv", and "yarn.lock". The "package.json" file is selected and open in the main editor. The code in the editor is as follows:

```
1 {
2   "name": "teste12062021",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "author": "Robson Gil Rocha",
7   "license": "ISC",
8   "dependencies": {
9     "csvtojson": "^2.0.10",
10    "readline-sync": "^1.4.10"
11  }
12 }
13
```

Projeto é iniciado no terminal usando o comando “node index.js” testando o arquivo index.js , biblioteca “readline-sync” , funcionando; é iniciado o projeto no arquivo index.js.

Terminando menu no arquivo index.js



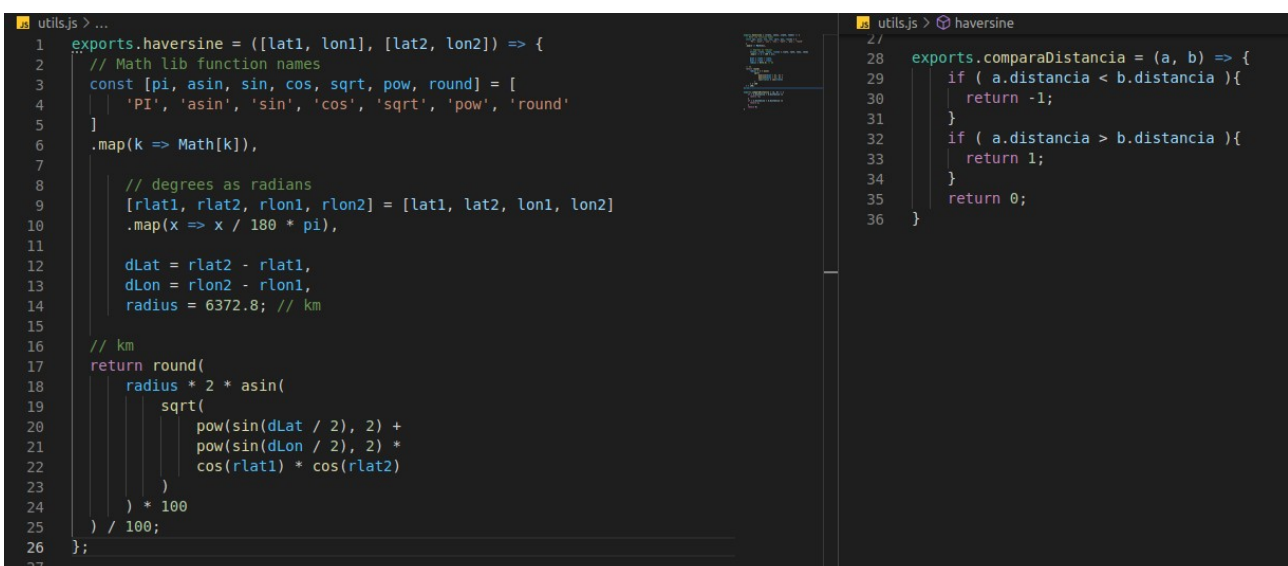
The screenshot shows a VS Code editor with a file explorer on the left and a code editor in the center. The file explorer shows a project structure with files like node_modules, prints, .gitignore, index.js, package.json, pontos_taxi.csv, and README.md. The code editor shows the index.js file with the following code:

```
6
7 function app(){
8   let op = 0
9
10  while(true){
11    console.log('=== MENU ===')
12    console.log('1. Listar todos os pontos de taxi')
13    console.log('2. Informar minha localização')
```

Below the code editor, there is a terminal window showing the output of the command "node index.js". The output is:

```
HELLO WORLD Robson !!!
```

Criando arquivo “utils.js” para adicionar fórmula de Haversine em forma de função, extraído pelo site http://rosettacode.org/wiki/Haversine_formula e a função de comparar distancias.



The screenshot shows a VS Code editor with two files open: utils.js and haversine.js. The utils.js file contains the following code:

```
1 exports.haversine = ([lat1, lon1], [lat2, lon2]) => {
2   // Math lib function names
3   const [pi, asin, sin, cos, sqrt, pow, round] = [
4     'PI', 'asin', 'sin', 'cos', 'sqrt', 'pow', 'round'
5   ]
6   .map(k => Math[k]),
7
8   // degrees as radians
9   [rLat1, rLat2, rLon1, rLon2] = [lat1, lat2, lon1, lon2]
10  .map(x => x / 180 * pi),
11
12  dLat = rLat2 - rLat1,
13  dLon = rLon2 - rLon1,
14  radius = 6372.8; // km
15
16  // km
17  return round(
18    radius * 2 * asin(
19      sqrt(
20        pow(sin(dLat / 2), 2) +
21        pow(sin(dLon / 2), 2) *
22        cos(rLat1) * cos(rLat2)
23      )
24    ) * 100
25  ) / 100;
26 };
27
```

The haversine.js file contains the following code:

```
28 exports.comparaDistancia = (a, b) => {
29   if ( a.distancia < b.distancia ){
30     return -1;
31   }
32   if ( a.distancia > b.distancia ){
33     return 1;
34   }
35   return 0;
36 }
```

Testando e finalizando a primeira opção.

```
11 while(true) {
12     console.log('=== MENU ===')
13     console.log('1. Listar todos os pontos de taxi')
14     console.log('2. Informar minha localização')
15     console.log('3. Encontrar pontos próximos')
16     console.log('4. Buscar pontos por logradouro')
17     console.log('5. Terminar o programa\n')
18
19     op = readlineSync.question('Escolha uma das opções: ');
20
21     if (op == 1) {
22         pontos.forEach(ponto => console.log(ponto.nome))
23     }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

2. Informar minha localização
3. Encontrar pontos próximos
4. Buscar pontos por logradouro
5. Terminar o programa

Escolha uma das opções: 1
RODOVIÁRIA
AV. DAS INDÚSTRIAS X AV. SEVERO DULLIUS
DOM PEDRO II X CORCOVADO
SHOPPING CENTER IGUATEMI II
RAMALHETE X TENENTE ALPOIN

Testando e finalizando a segunda opção.

```

23
24 } else if(op == 2) {
25     console.log('Informe sua localização:')
26     lat = readlineSync.question('Digite sua latitude: ')
27     long = readlineSync.question('Digite sua longitude: ')
28     console.log('Localização armazenada.')
29
30 } else if (op == 3) {
31     console.log(op, "foi selecionada")
32
33 } else if (op == 4) {
34     console.log(op, "foi selecionada")
35
36 } else if (op == 5) {
37     console.log(op, "foi selecionada")
38     }
39 break

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

Escolha uma das opções: 2
Informe sua localização:
Digite sua latitude: 0
Digite sua longitude: 0
Localização armazenada.

```

Testando e finalizando a terceira opção.

```

30 } else if (op == 3) {
31     if (lat === null || long === null) {
32         console.log('Informe sua localização primeiro')
33     } else {
34         console.log('Os pontos de taxi mais próximos são:')
35
36         const pontosProximos = []
37         pontos.forEach(ponto => {
38             pontosProximos.push({
39                 ...ponto,
40                 distancia: utils.haversine(
41                     [parseFloat(lat), parseFloat(long)],
42                     [parseFloat(ponto.latitude.replace(",", ".")), parseFloat(ponto.longitude.replace(",", "."))]
43                 )
44             })
45         })
46
47         pontosProximos.sort(utils.comparaDistancia).slice(0, 3).forEach(ponto => console.log(`${ponto.nome} (${ponto.distancia}km)`)
48     }
49

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

5: bash

5. Terminar o programa

```

Escolha uma das opções: 3
Os pontos de taxi mais próximos são:
RODOVIÁRIA (0.04km)
CONCEIÇÃO X PASSARELA DA RODOVIÁRIA (0.05km)
RODOVIÁRIA (ANEXO/NOTURNO) (0.05km)

```

Testando e finalizando a quarta opção.

```

} else if (op == 4) {
    let busca = readlineSync.question('Digite todo ou parte do nome do logradouro: ')
    console.log('Os pontos de taxi ao longo de ${busca.toUpperCase()} são:')
    const pontosBuscados = pontos.filter(ponto => ponto.logradouro.includes(busca.toUpperCase()))
    pontosBuscados.forEach(ponto => console.log(ponto.nome))

```

```
Escolha uma das opções: 4
Digite todo ou parte do nome do logradouro: ipiranga
Os pontos de taxi ao longo de IPIRANGA são:
PUC UNIVERSIDADE
HOSPITAL DA PUC
IPIRANGA X AZENHA
HOSPITAL ERNESTO DORNELLES
IPIRANGA (HOSPITAL PUC)
```

Testando e finalizando a quinta e ultima opção.

```
50     } else if (op == 4) {
51         let busca = readlineSync.question('Digite todo ou parte do nome do logradouro: ')
52         console.log(`Os pontos de taxi ao longo de ${busca.toUpperCase()} são:`)
53         const pontosBuscados = pontos.filter(ponto => ponto.logradouro.includes(busca.toUpperCase()))
54         pontosBuscados.forEach(ponto => console.log(ponto.nome))
55     }
56     } else if (op == 5) {
57         console.log(`\n Fim da aplicação \n \n `)
58         break
59     }
60 }
61
62 }
63
64 app()
65
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

5: bash

```
=== MENU ===
1. Listar todos os pontos de taxi
2. Informar minha localização
3. Encontrar pontos próximos
4. Buscar pontos por logradouro
5. Terminar o programa
```

Escolha uma das opções: 5

Fim da aplicação

Impedindo que o usuário coloque dados incorretos no MENU

```
else if (op !== 5) {
    console.log(`\n Digite uma opção valida! ',`${op}`, ' não é opção valida \n \n `)
```

```
console.log('Informe sua localização:');
    lat = readlineSync.question('Digite sua latitude: ');
    long = readlineSync.question('Digite sua longitude: ');
    if (lat < limitgeo[0] && lat > limitgeo[1] && long < limitgeo[2] && long > limitgeo[3])
        console.log(`\nSua localização é ,latitude ${lat}`, ' e longitude ${long}`);
        enter = readlineSync.question('Aperte qualquer tecla para continuar.\n');
    } else {
        console.log(`\nDados invalidos`);
        console.log('    Siga o exemplo para preencher os dados');
        console.log('    corretamente, latitude: \'-30.023927\\', ' ');
        console.log('    longitude \'-51.219871\\', ' qualquer outro valor não é valido');
        console.log('    qualquer outro valor não é valido');
        enter = readlineSync.question(`\nAperte qualquer tecla para continuar.\n`);
    }
}
```

Execução:

```
Escolha uma das opções: 9

  Digite uma opção valida!  9  não é opção valida

=== MENU ===
1. Listar todos os pontos de taxi
2. Informar minha localização
3. Encontrar pontos próximos
4. Buscar pontos por logradouro
5. Terminar o programa

Escolha uma das opções: qualquer string

  Digite uma opção valida!  qualquer string  não é opção valida
```

Comentando o código.

```
const readlineSync = require('readline-sync'); // lib para ler o input do usuário
const csv = require('csvtojson'); // lib para ler o arquivo CSV para JSON
const utils = require('./utils'); // acessando arquivo utils.js

async function app() { // aplicação em si. Função principal
  const pontos = await csv({delimiter: ";"}).fromFile('pontos_taxi.csv'); // lendo o arquivo CSV
  let op = 0 //Menu vareavel opção
  let enter = 0; //Menu vareavel
  let lat = 0; // -30.023927; rodoviaria de PoA
  let long = 0; // -51.219871; rodoviaria de PoA
  let limitgeo = [-29.855968, -30.285812, -50.981312, -51.367795]; //limit da latitude e longitude em
```

Finalizando PDF, enviando ZIP e Obrigado pela oportunidade.

gil1990gil@hotmail.com