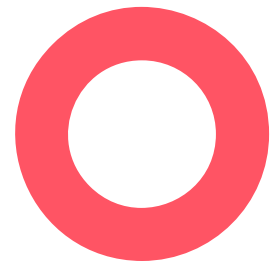


# DESENVOLVIMENTO DE SISTEMAS

## UC13

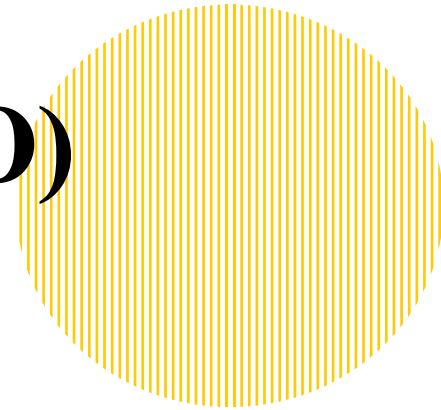
Prof. Viviane de Lima

[viviane.lfrancelino@sp.senac.br](mailto:viviane.lfrancelino@sp.senac.br)

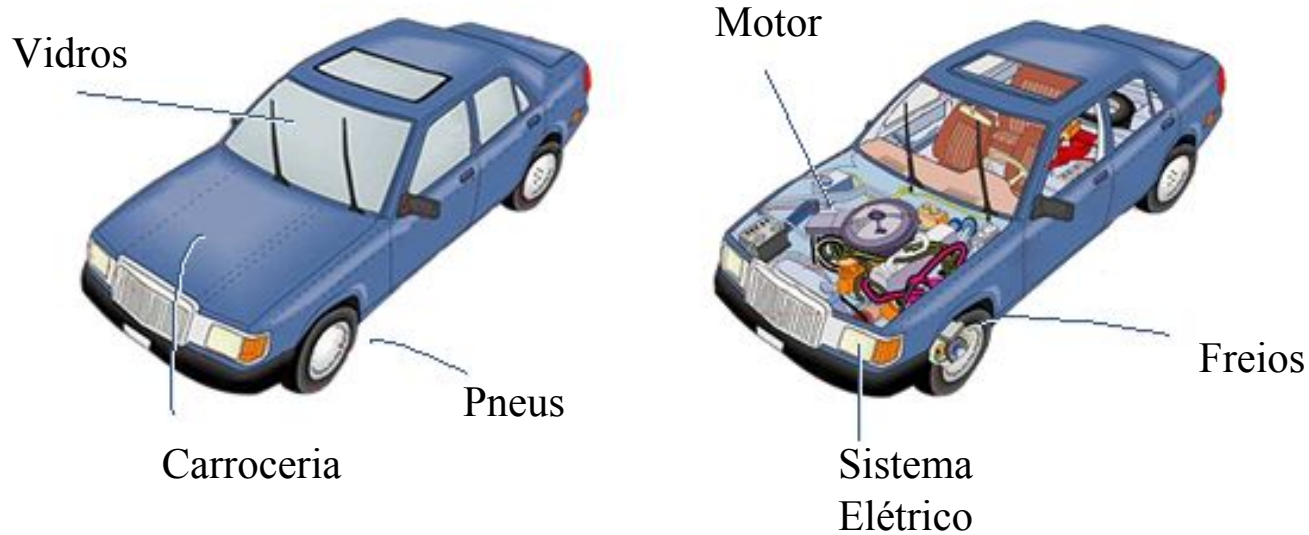


**AULA 01**

# **VISÃO GERAL FUNCIONAMENTO DO NODEJS (REVISÃO)**



# FRONT-END X BACK-END



O NODEJS É A MONTADORA DE CARROS , ELE QUE VAI SER RESPONSÁVEL POR ASSEGURAR QUE O FRONT-END E O BACK-END FUNCIONEM JUNTOS

# ANTES DO NODEJS



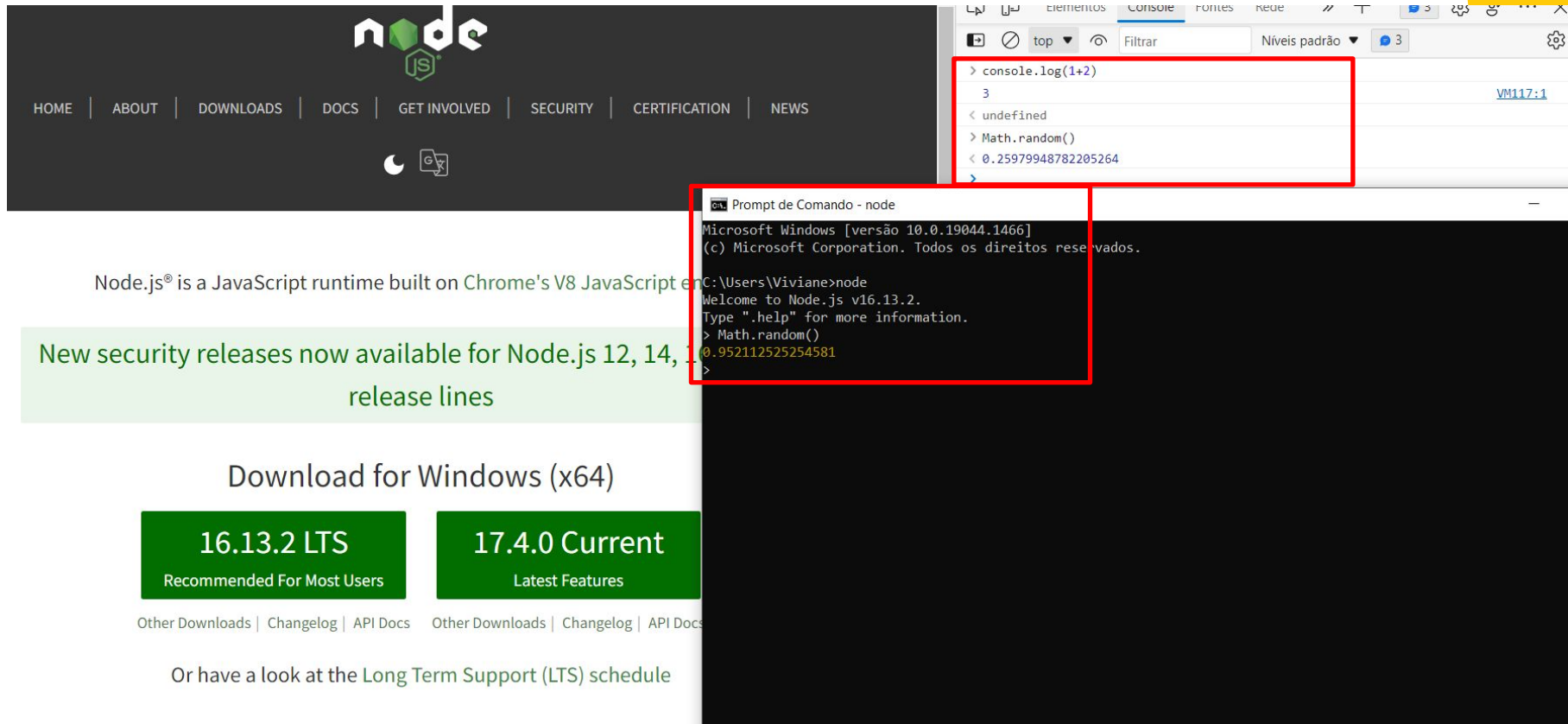
Antes do NodeJS o JavaScript funcionava apenas no navegador, então se você queria executar código JavaScript você criava um site e rodava esse site dentro do próprio navegador, basicamente foi o que aprendemos de JS na UC15, criamos eventos, manipulamos com DOM e rodamos a requisição no navegador.

# DEPOIS DO NODEJS

Com o surgimento do NodeJS o JavaScript passou a ser utilizado de outras formas, sem ser dentro do navegador. O surgimento do NodeJS abriu um leque de oportunidades, pois com ele podemos criar um servidor, um app e etc.



# ONDE PODEMOS EXECUTAR JS?



The image shows a composite of three elements: the Node.js website header, a browser console, and a Windows command prompt. The website header includes the Node.js logo and navigation links. The browser console shows JavaScript code being executed. The command prompt shows the Node.js runtime being started and a command being executed.

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine

New security releases now available for Node.js 12, 14, 16, and 18 release lines

Download for Windows (x64)

**16.13.2 LTS**  
Recommended For Most Users

**17.4.0 Current**  
Latest Features

Other Downloads | Changelog | API Docs

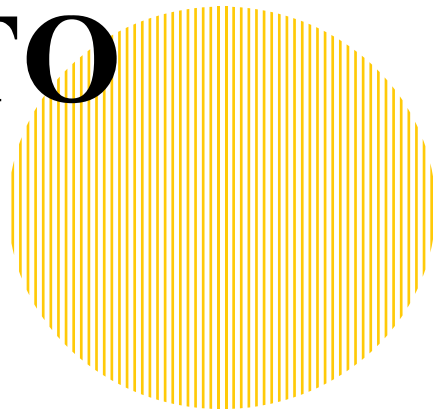
Or have a look at the [Long Term Support \(LTS\) schedule](#)

```
> console.log(1+2)
3
< undefined
> Math.random()
< 0.25979948782205264
>
```

```
Microsoft Windows [versão 10.0.19044.1466]
(c) Microsoft Corporation. Todos os direitos reservados.
C:\Users\Viviane>node
Welcome to Node.js v16.13.2.
Type ".help" for more information.
> Math.random()
0.952112525254581
>
```

**NO CONSOLE E NO CMD**

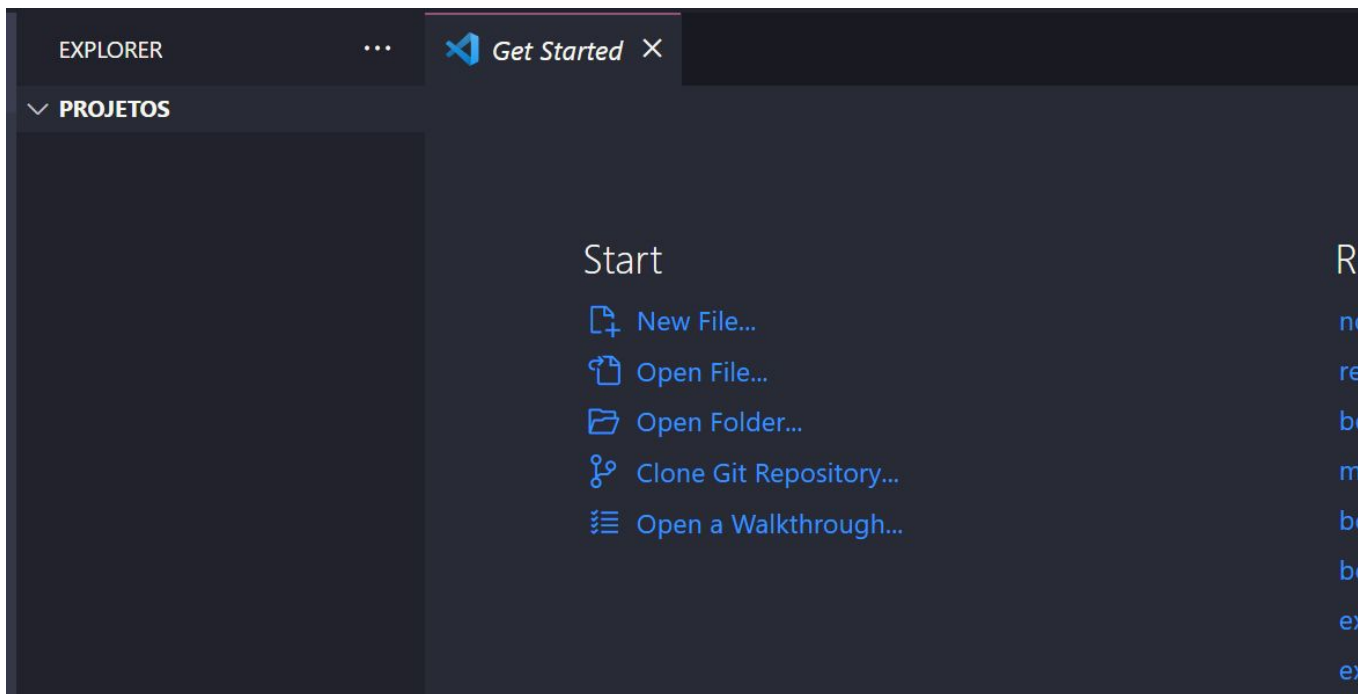
# **CRIANDO NOSSO PRIMEIRO PROJETO COM NODEJS**



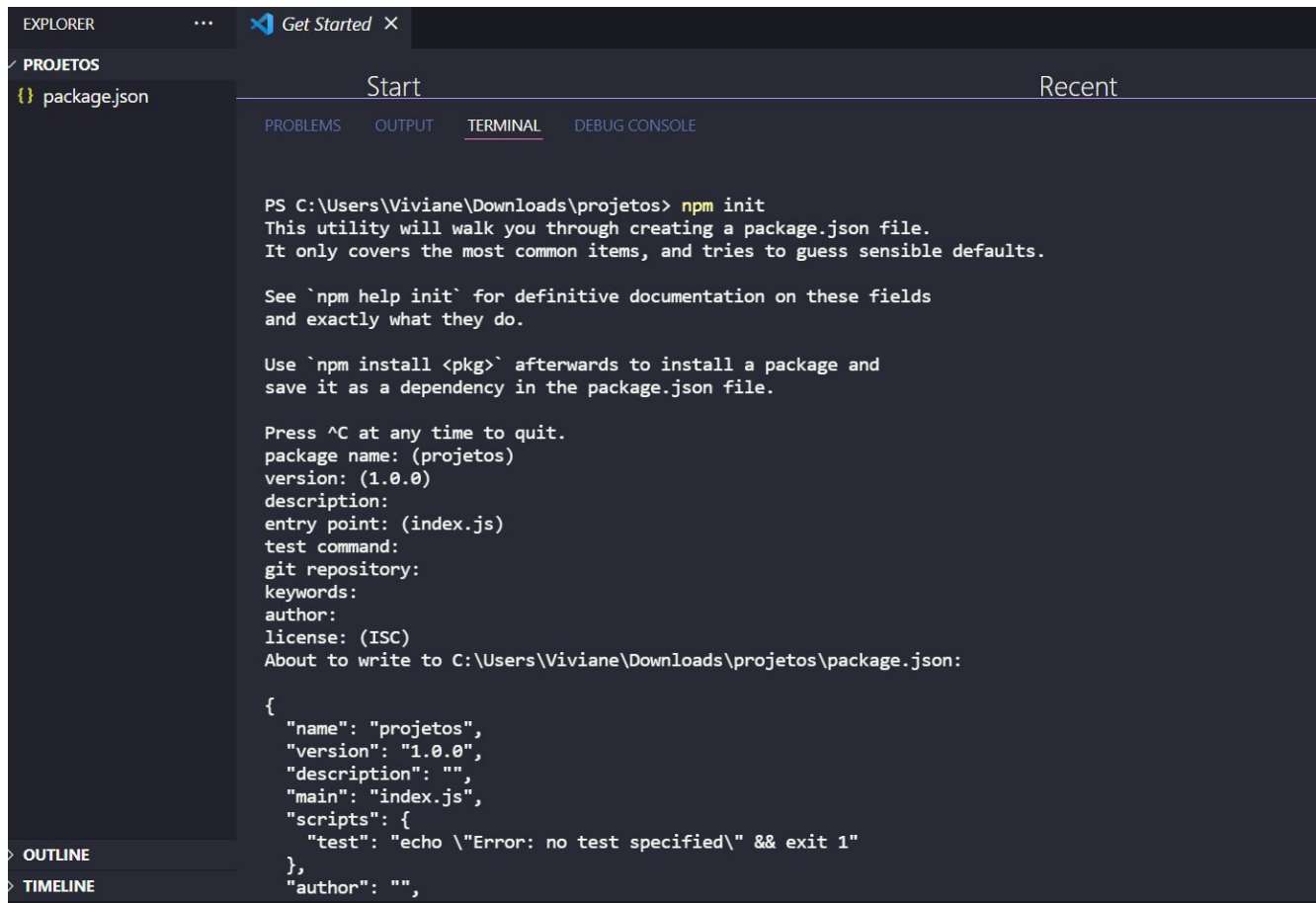
# O PRIMEIRO PROJETO

O NodeJS é diferente de outras linguagens back-end, se você está acostumado com PHP você sabe que é necessário instalar um servidor e depois rodar o servidor para que o PHP funcione. Entretanto no NodeJS conseguimos criar o nosso próprio servidor dentro do nosso projeto

## CRIE UM PASTA CHAMADA PROJETOS E ABRA NO VSCODE



COM O VSCODE ABERTO ABRA O TERMINAL E DIGITE O COMANDO **NPM INIT**, ELE VAI TE PERGUNTAR VÁRIAS COISAS, APENAS DÊ ENTER EM TODAS ELAS, QUANDO CHEGAR NO **IS THIS OK?** COLOQUE Y(YES)



```
EXPLORED ... Get Started X
PROJETOS
package.json

Start Recent
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\Viviane\Downloads\projetos> 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.

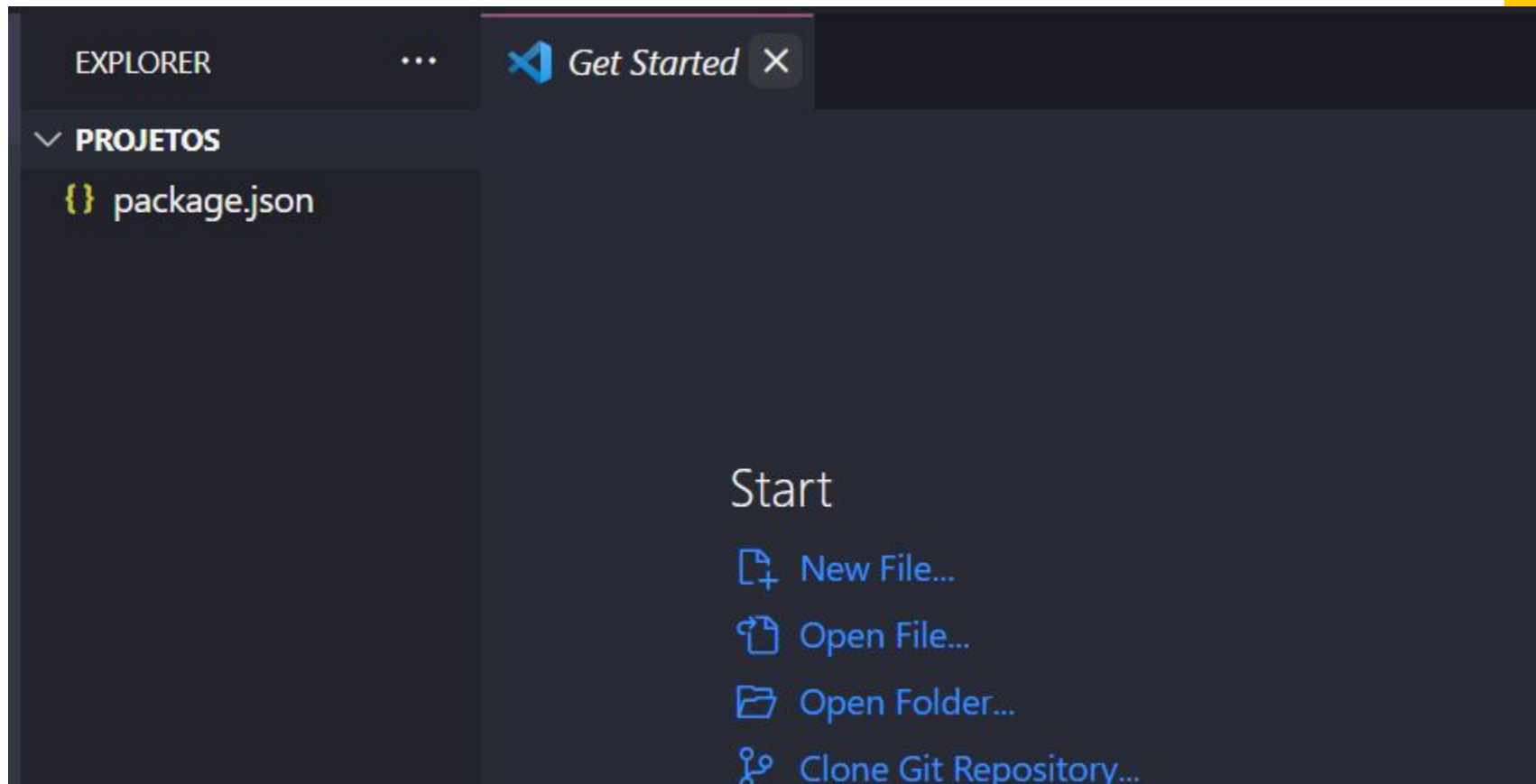
Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (projetos)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\Viviane\Downloads\projetos\package.json:

{
  "name": "projetos",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
```



OBSERVE QUE UM ARQUIVO FOI CRIADO NO SEU PROJETO



## CLIQUE EM PACKAGE.JSON E VEJA AS INFORMAÇÕES QUE APARECEM

EXPLORER

...

{ } package.json X

▼ PROJETOS

{ } package.json

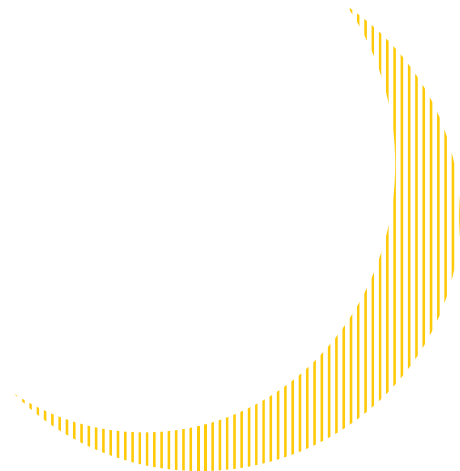
{ } package.json > ...

```
1  {
2    "name": "projetos",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC"
11  }
12
```

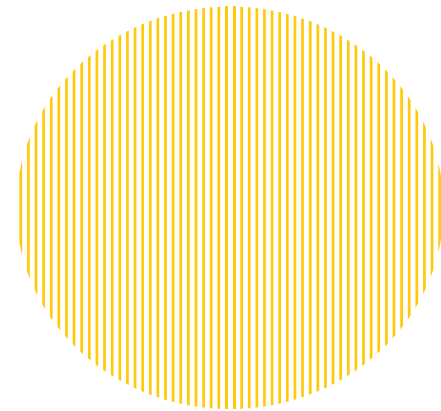
# O QUE É O PACKAGE.JSON?



No NodeJS nós vamos trabalhar com bibliotecas externas, ou seja, vamos aproveitar códigos da comunidade , então vamos trabalhar com muitas **dependências**. Para isso precisamos organizar em algum lugar do nosso código quais são as dependências, versões e todas as informações gerais. O package.json, é um dos arquivos mais importantes pois é ele que vai mostrar como nosso projeto está funcionando



# **INSTALANDO O TYPESCRIPT NO PROJETO**



# ABRA O CMD DO VSCODE E DIGITE **NPM INSTALL -G** **TYPESCRIPT**



C:\> Prompt de Comando

```
Microsoft Windows [versão 10.0.19044.1466]  
(c) Microsoft Corporation. Todos os direitos reservados.
```

```
C:\Users\Viviane>npm install -g typescript
```

```
added 1 package, and audited 2 packages in 4s
```

```
found 0 vulnerabilities
```

```
C:\Users\Viviane>
```



**AGORA VAMOS CONFIGURAR O TYPESCRIPT NO NOSSO PROJETO, COM O PROJETO ABERTO NO VSCODE, ABRA O TERMINAL E DIGITE O COMANDO: **TSC****



The screenshot shows the Visual Studio Code interface. In the Explorer panel on the left, under the 'PROJETOS' folder, there is a file named 'package.json' and a file named 'tsconfig.json' with a TypeScript icon and a red arrow pointing to it. The Editor panel in the center shows the content of 'package.json' with the following JSON structure:

```
1 {
2   "name": "projetos",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "author": "",
10  "license": "ISC"
11 }
12
```

The Terminal panel at the bottom shows the command prompt with the command `tsc --init` entered and highlighted by a red box. Below the command, it says 'Created a new tsconfig.json with:'.

Observe que foi Instalado um aquivo chamado tsconfig.json

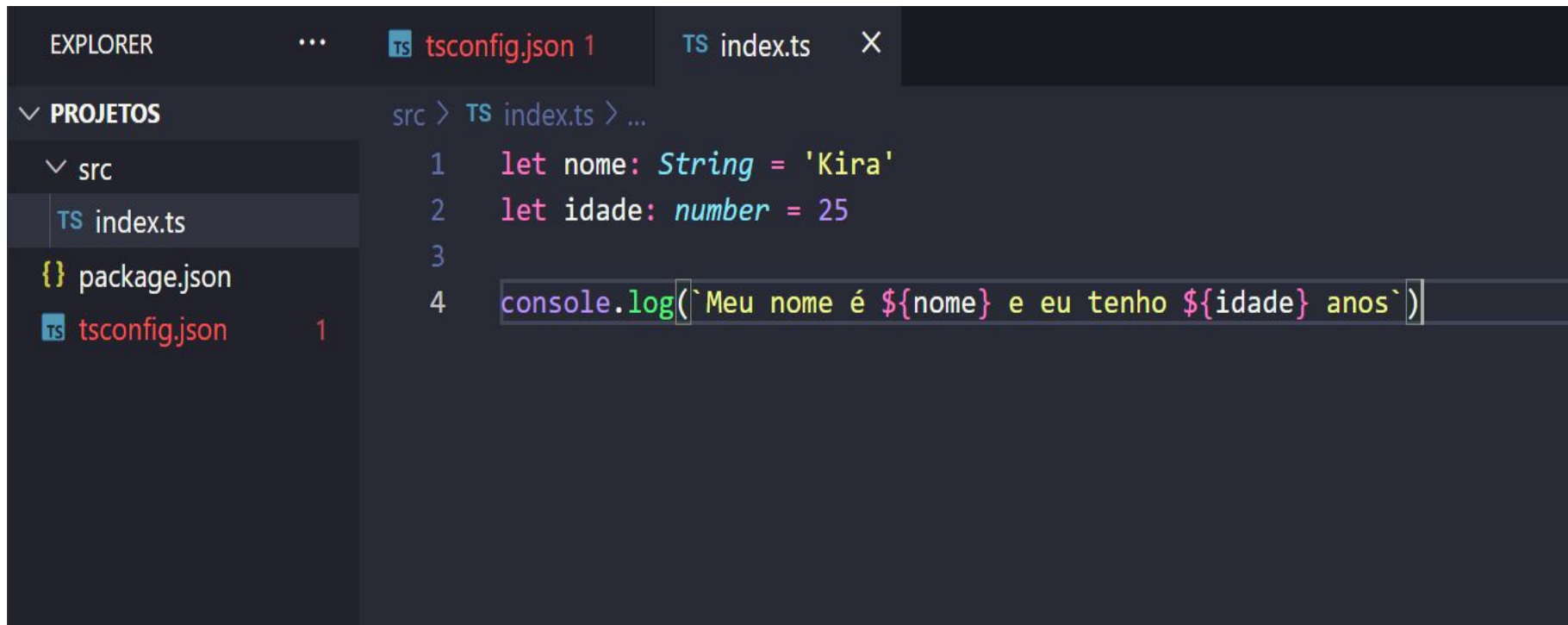


**SE DER ERRO NA MÁQUINA DOS ALUNOS  
TENTAR O COMANDO:**

**npx tsc --init**  
**npm install -D ts-node**



# CRIE UM PASTA CHAMADA SRC E DENTRO DA PASTA CRIE SEU ARQUIVO INDEX.TS



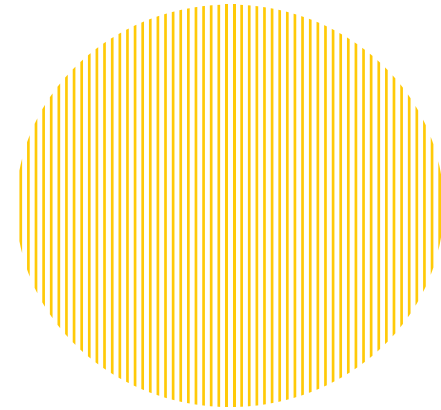
The image shows a screenshot of the Visual Studio Code editor interface. On the left, the Explorer sidebar is open, showing a project named 'PROJETOS'. Inside 'PROJETOS', there is a folder named 'src'. Within the 'src' folder, there is a file named 'index.ts'. Below the 'src' folder, there are two other files: 'package.json' and 'tsconfig.json'. The 'index.ts' file is selected, and its content is displayed in the main editor area. The code in 'index.ts' is as follows:

```
src > TS index.ts > ...  
1  let nome: String = 'Kira'  
2  let idade: number = 25  
3  
4  console.log(`Meu nome é ${nome} e eu tenho ${idade} anos`)
```

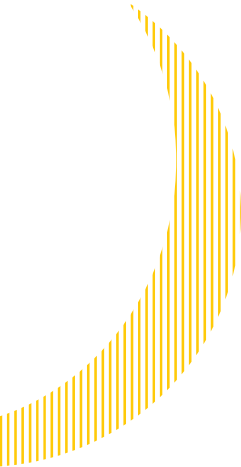





# **CONFIGURANDO O TYPESCRIPT**



## ACESSE TSConfig.JSON E FAÇA AS ALTERAÇÕES ABAIXO:



```
package.json  index.ts  tsconfig.json X
tsconfig.json > {} compilerOptions
8      // "tsBuildInfoFile": "./", /* Specify the folder
9      // "disableSourceOfProjectReferenceRedirect": true, /* Disable preferring
10     // "disableSolutionSearching": true, /* Opt a project out
11     // "disableReferencedProjectLoad": true, /* Reduce the number
12
13     /* Language and Environment */
14     "target": "es6", /* Set the JavaScript la
15     // "lib": [], /* Specify a set of b
16     // "jsx": "preserve", /* Specify what JSX c
17     // "experimentalDecorators": true, /* Enable experimenta
18     // "emitDecoratorMetadata": true, /* Emit design-type m
19     // "jsxFactory": "", /* Specify the JSX fa
20     // "jsxFragmentFactory": "", /* Specify the JSX Fr
21     // "jsxImportSource": "", /* Specify module spe
22     // "reactNamespace": "", /* Specify the object
23     // "noLib": true, /* Disable including
24     // "useDefineForClassFields": true, /* Emit ECMAScript-st
25
26     /* Modules */
27     "module": "commonjs", /* Specify what modul
28     "moduleResolution": "node",
29     // "rootDir": "./", /* Specify the root f
30     // "moduleResolution": "node", /* Specify how TypeSc
```

# OUTDIR E ROOTDIR

```
package.json index.ts tsconfig.json X
tsconfig.json > {} compilerOptions
13      /* Language and Environment */
14      "target": "es6",                                /* Set the JavaScript language
15      // "lib": [],                                    /* Specify a set of bundl
16      // "jsx": "preserve",                            /* Specify what JSX code
17      // "experimentalDecorators": true,               /* Enable experimental su
18      // "emitDecoratorMetadata": true,               /* Emit design-type metad
19      // "jsxFactory": "",                             /* Specify the JSX factory
20      // "jsxFragmentFactory": "",                    /* Specify the JSX Fragme
21      // "jsxImportSource": "",                       /* Specify module specifi
22      // "reactNamespace": "",                        /* Specify the object inv
23      // "noLib": true,                               /* Disable including any
24      // "useDefineForClassFields": true,              /* Emit ECMAScript-standa
25
26      /* Modules */
27      "module": "commonjs",                            /* Specify what module co
28      "moduleResolution": "node",
29      "outDir": "./dist",                                ← Pasta final do projeto
30      "rootDir": "./src",                                ← Pasta do nosso projeto
31      // "moduleResolution": "node",                  /* Specify the root folder
32      // "baseUrl": "./",                             /* Specify how TypeScript
33      // "paths": {},                                  /* Specify the base direc
34      // "paths": {},                                  /* Specify a set of entri
35      // "rootDirs": [],                               /* Allow multiple folders
36      // "rootDirs": [],                               /* Allow multiple folders
```



# DEPENDÊNCIA

É como se fosse uma casa.

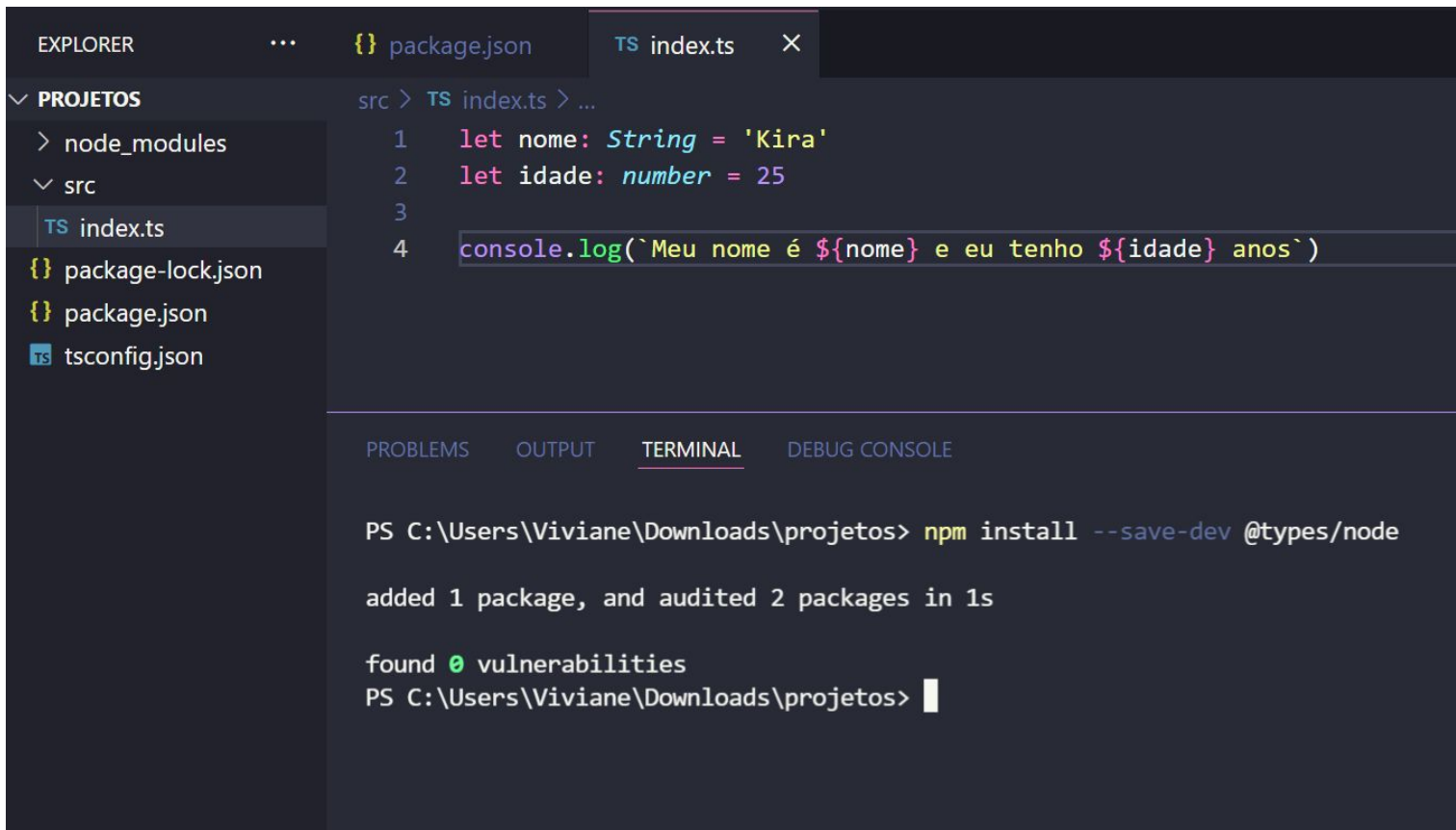
Para uma casa existir, ela depende de uma rua, e depende também de um lote.  
Depende porque sem a Rua ou sem o lote a casa **não consegue ser** uma casa e,  
por **depende** do lote ou da rua, **qualquer alteração** na estrutura desse lote ou dessa  
rua pode **gerar efeito colateral** na casa.



# INSTALANDO ALGUMAS DEPENDÊNCIAS

Vamos instalar uma dependência chamada TypesNode, essa dependência vai trazer para o TypeScript novas noções de código para deixarmos o TypeScript mais inteligente.

Abra o terminal e digite o comando: **install --save-dev @types/node**



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project named 'PROJETOS' with a 'src' folder containing 'index.ts'. The main editor displays the contents of 'index.ts':

```
src > TS index.ts > ...
1  let nome: String = 'Kira'
2  let idade: number = 25
3
4  console.log(`Meu nome é ${nome} e eu tenho ${idade} anos`)
```

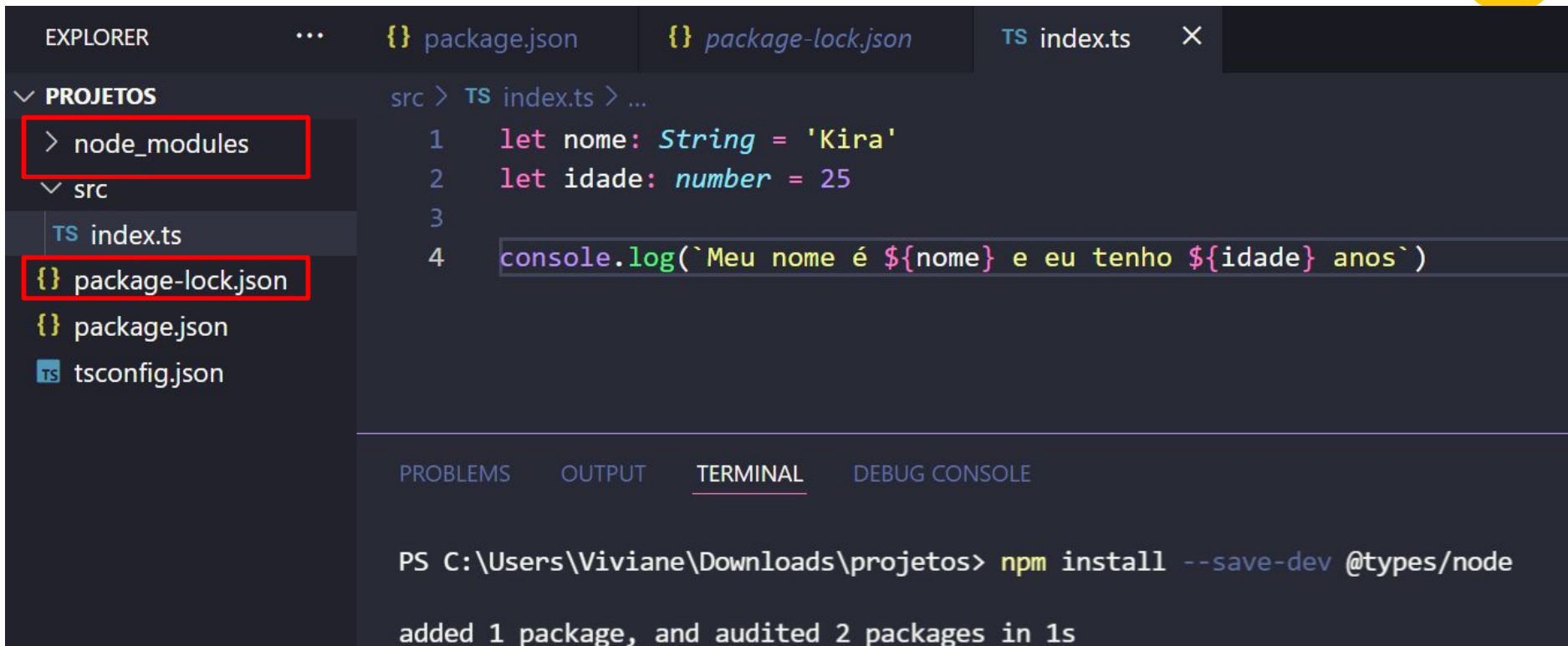
Below the editor, the TERMINAL tab is active, showing the command prompt output:

```
PS C:\Users\Viviane\Downloads\projetos> npm install --save-dev @types/node

added 1 package, and audited 2 packages in 1s

found 0 vulnerabilities
PS C:\Users\Viviane\Downloads\projetos> 
```

# OBSERVE QUE A PASTA NODE\_MODULES E PACKAGE-LOCK.JSON FORAM CRIADAS



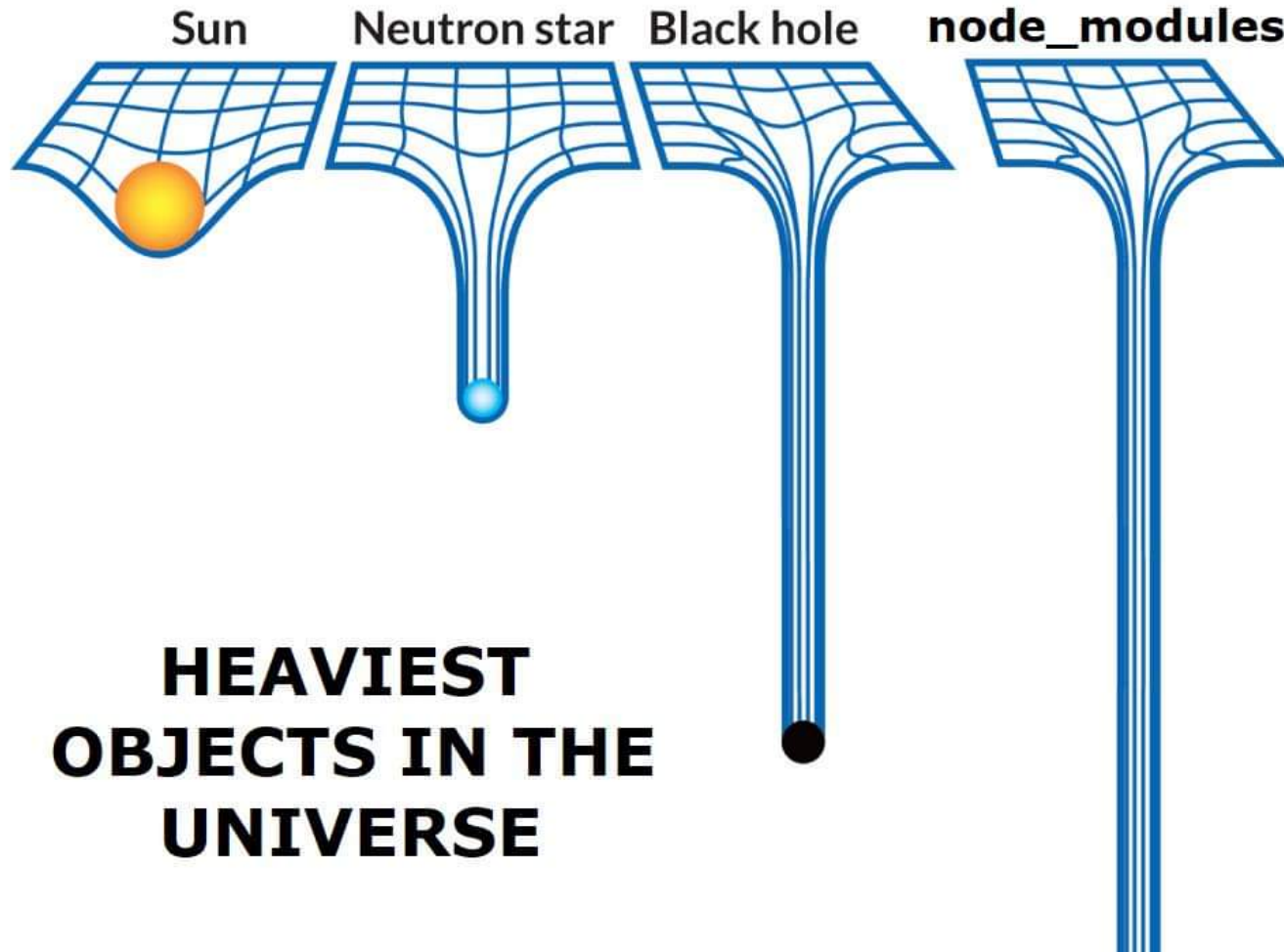
The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure under 'PROJETOS'. It includes a folder 'src' containing 'TS index.ts', and two files at the root: 'package-lock.json' and 'package.json'. Both 'node\_modules' and 'package-lock.json' are highlighted with red rectangles. The main editor shows 'TS index.ts' with the following TypeScript code:

```
src > TS index.ts > ...  
1 let nome: String = 'Kira'  
2 let idade: number = 25  
3  
4 console.log(`Meu nome é ${nome} e eu tenho ${idade} anos`)
```

At the bottom, the TERMINAL panel shows the command executed in the PowerShell prompt:

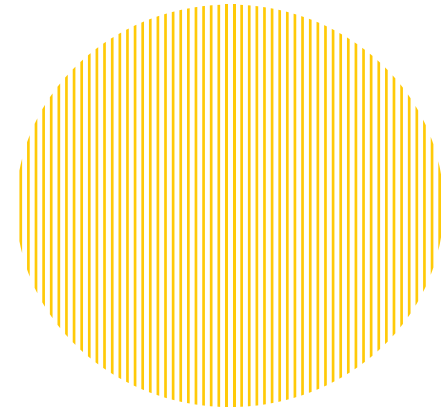
```
PS C:\Users\Viviane\Downloads\projetos> npm install --save-dev @types/node  
  
added 1 package, and audited 2 packages in 1s
```

# O QUE É O NODE\_MODULES?



**HEAVIEST  
OBJECTS IN THE  
UNIVERSE**

# CONHECENDO O NODEMON






# O NODEMON


Vamos aprender a instalar e usar a biblioteca chamada Nodemon, vamos usar essa biblioteca em nossos projetos que iremos desenvolver. O Nodemon vai servir para monitorar alterações dos nossos arquivos, dessa forma o Nodemon vai facilitar bastante nossa vida com desenvolvimento de projetos. Acesse o site:

[www.npmjs.com/package/nodemon](http://www.npmjs.com/package/nodemon)




 Need Package Maintenance

ProductsPricingDocumentationCc


 Search packages



Search


Sign Up


**nodemon** 


2.0.15 • Public • Published 3 months ago


 Readme

 Explore 

 10 Dependencies

 3.439 Dependents

 236 Versions




**nodemon**


Install


```
> npm i nodemon
```

Repository

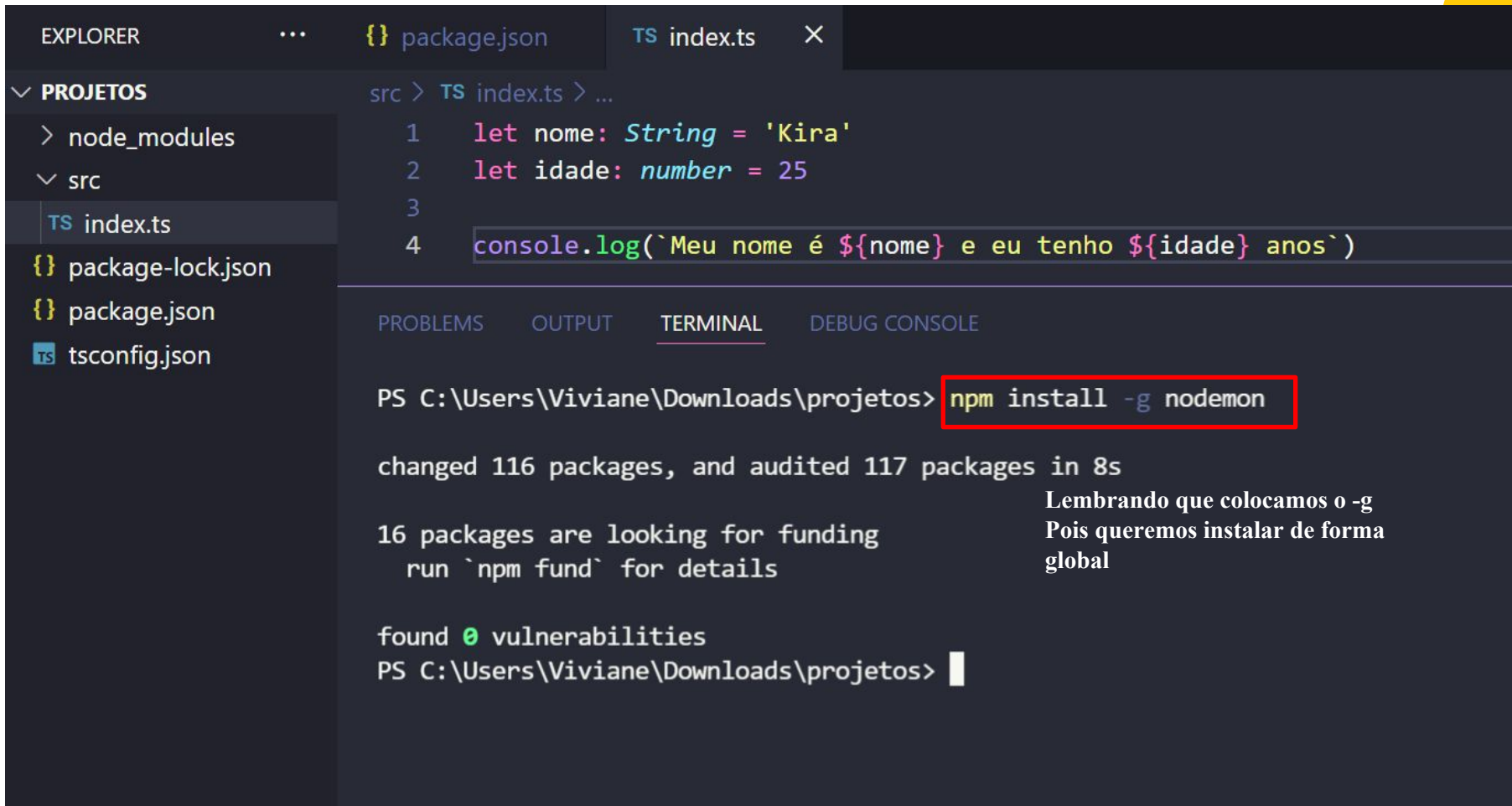
 [github.com/remy/nodemon](https://github.com/remy/nodemon)

Homepage

 [nodemon.io](http://nodemon.io)

 Fund this package

# VÁ NO TERMINAL DO SEU VSCODE E VAMOS INSTALAR O NODEMON



The image shows a screenshot of the Visual Studio Code (VS Code) interface. On the left, the Explorer sidebar is open, showing a project structure with folders 'node\_modules' and 'src', and files 'package-lock.json', 'package.json', and 'tsconfig.json'. The main editor area displays a TypeScript file named 'index.ts' with the following code:

```
src > TS index.ts > ...
1 let nome: String = 'Kira'
2 let idade: number = 25
3
4 console.log(`Meu nome é ${nome} e eu tenho ${idade} anos`)
```

Below the editor, the TERMINAL tab is active, showing the output of a command. The command `npm install -g nodemon` has been executed, and the output is as follows:

```
PS C:\Users\Viviane\Downloads\projetos> npm install -g nodemon

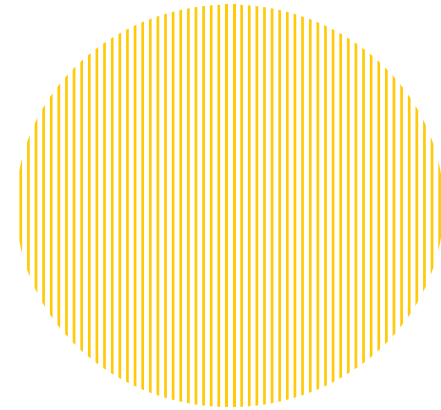
changed 116 packages, and audited 117 packages in 8s

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

found 0 vulnerabilities
PS C:\Users\Viviane\Downloads\projetos>
```

A red rectangular box highlights the command `npm install -g nodemon` in the terminal. To the right of the terminal output, there is a text annotation in Portuguese: "Lembrando que colocamos o -g Pois queremos instalar de forma global".

# **USANDO O NODEMON COM TYPESCRIPT**



# INSTALANDO A DEPENDÊNCIA TS NODE

Acesse o site do npmjs e busque pelo TSNODE



Neapolitan Pasta Maker

**npm**



Search packages

**ts-node** TS

10.4.0 • Public • Published 3 months ago



Readme



Explore

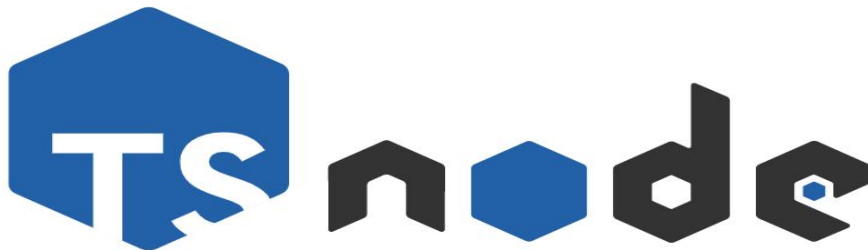
BETA



12 Dependencies



5.



npm

v10.4.0

downloads

69M/month

build

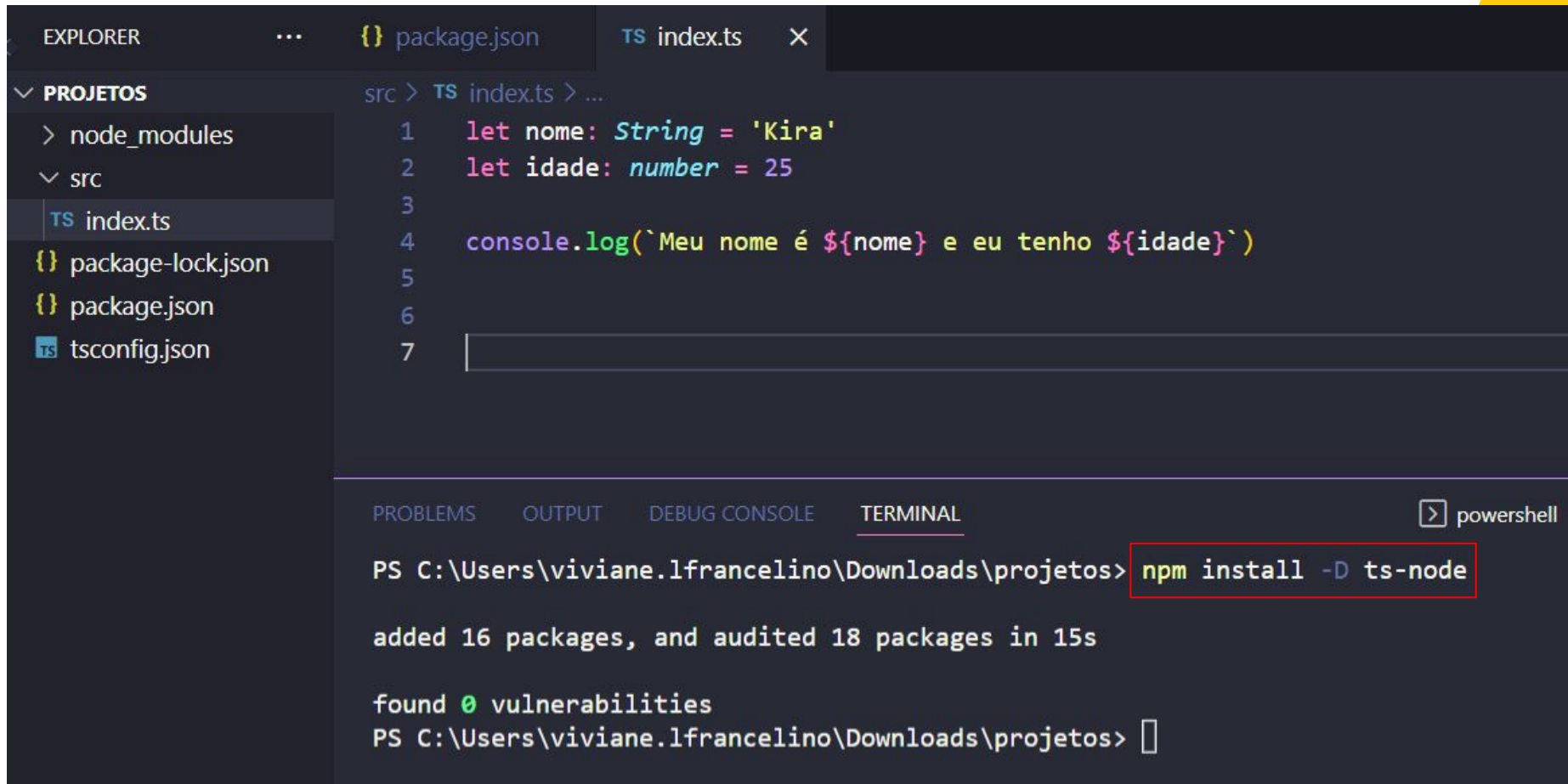
passing

codecov

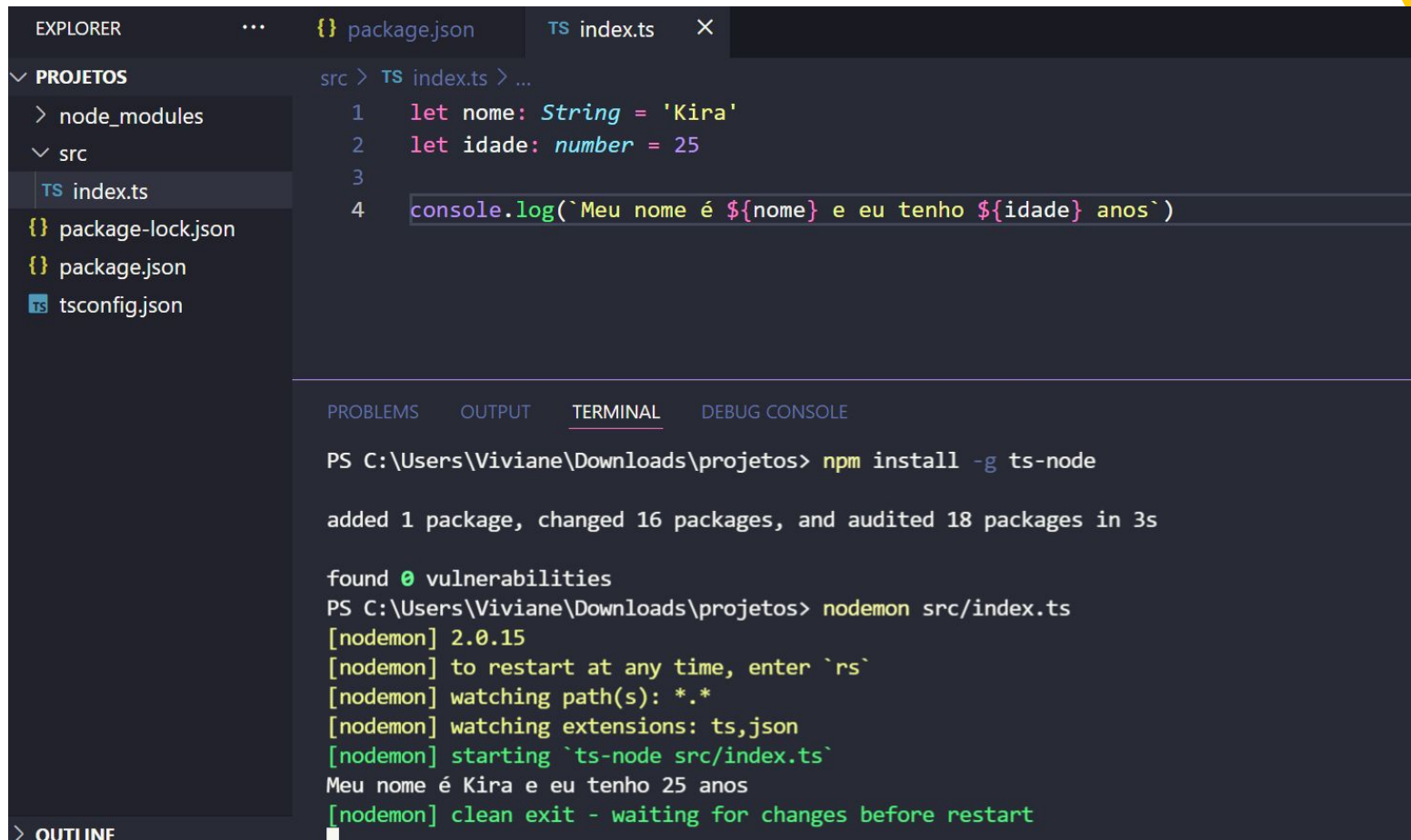
78%

TypeScript execution and REPL for node.js, with source map and native ESM support.

# NO SEU TERMINAL DO VS CODE INSTALE O TS NDOE



# EM SEU NOVO TERMINAL, DIGITE O COMANDO TS-NODE COM O NODEMON QUE É: **nodemon src/index.ts**



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project named 'PROJETOS' with a 'src' folder containing 'index.ts'. The main editor displays the content of 'src > TS index.ts' with the following code:

```
1 let nome: String = 'Kira'
2 let idade: number = 25
3
4 console.log(`Meu nome é ${nome} e eu tenho ${idade} anos`)
```

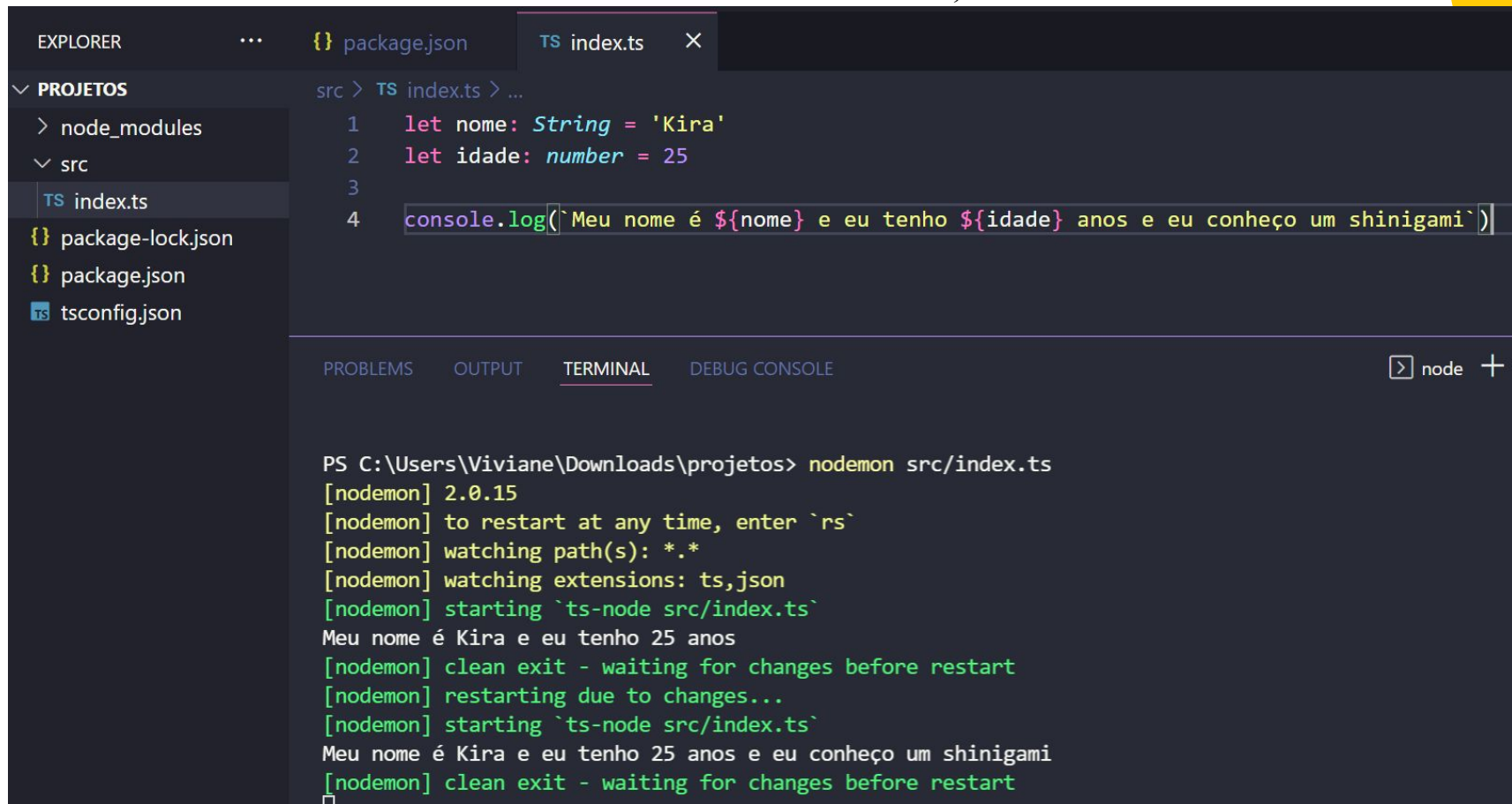
Below the editor, the TERMINAL tab is active, showing the following output:

```
PS C:\Users\Viviane\Downloads\projetos> npm install -g ts-node

added 1 package, changed 16 packages, and audited 18 packages in 3s

found 0 vulnerabilities
PS C:\Users\Viviane\Downloads\projetos> nodemon src/index.ts
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: ts,json
[nodemon] starting `ts-node src/index.ts`
Meu nome é Kira e eu tenho 25 anos
[nodemon] clean exit - waiting for changes before restart
```

# FAÇA UMA ALTERAÇÃO NO SEU CÓDIGO E VEJA QUE VOCÊ NÃO PRECISA DIGITAR O COMANDO NODEMON NOVAMENTE, POIS ELE ATUALIZA



```
EXPLORER    ...    {} package.json    TS index.ts    X
```

▼ PROJETOS

- > node\_modules
- ▼ src
  - TS index.ts
- {} package-lock.json
- {} package.json
- TS tsconfig.json

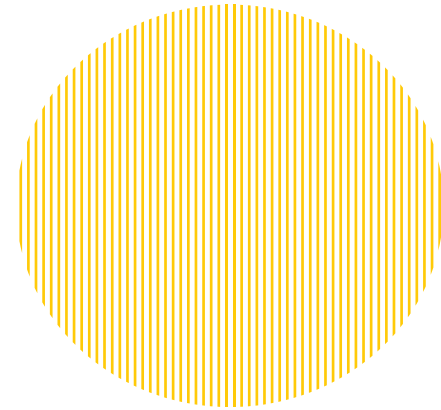
```
src > TS index.ts > ...
1  let nome: String = 'Kira'
2  let idade: number = 25
3
4  console.log(`Meu nome é ${nome} e eu tenho ${idade} anos e eu conheço um shinigami`)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

node +

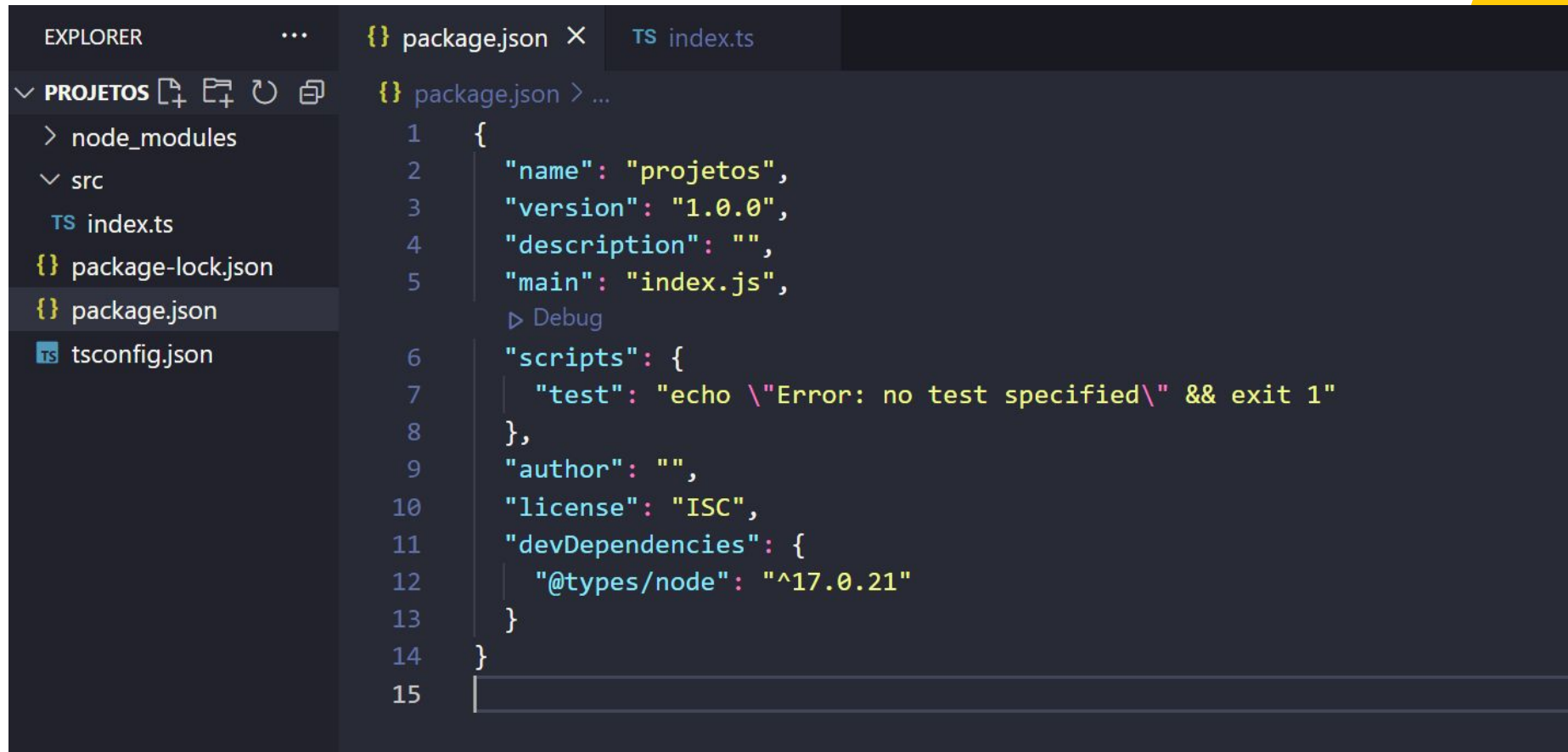
```
PS C:\Users\Viviane\Downloads\projetos> nodemon src/index.ts
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: ts,json
[nodemon] starting `ts-node src/index.ts`
Meu nome é Kira e eu tenho 25 anos
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `ts-node src/index.ts`
Meu nome é Kira e eu tenho 25 anos e eu conheço um shinigami
[nodemon] clean exit - waiting for changes before restart
```

# SCRIPTS DO PACKAGE.JSON





# VÁ ATÉ O ARQUIVO PACKAGE.JSON



EXPLORER

PROJETOS

- node\_modules
- src
- index.ts
- package-lock.json
- package.json
- tsconfig.json

package.json

```
{
  "name": "projetos",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "@types/node": "^17.0.21"
  }
}
```

EXPLORER

...

PROJETOS

node\_modules

src

index.ts

package-lock.json

package.json

tsconfig.json

package.json

TS index.ts

package.json > ...

1 {

2 "name": "projetos",

3 "version": "1.0.0",

4 "description": "",

5 "main": "index.js",

6 "scripts": {

7 | "test": "echo \"Error: no test specified\" && exit 1"

8 },

9 "author": "",

10 "license": "ISC",

11 "devDependencies": {

12 | "@types/node": "^17.0.21"

13 }

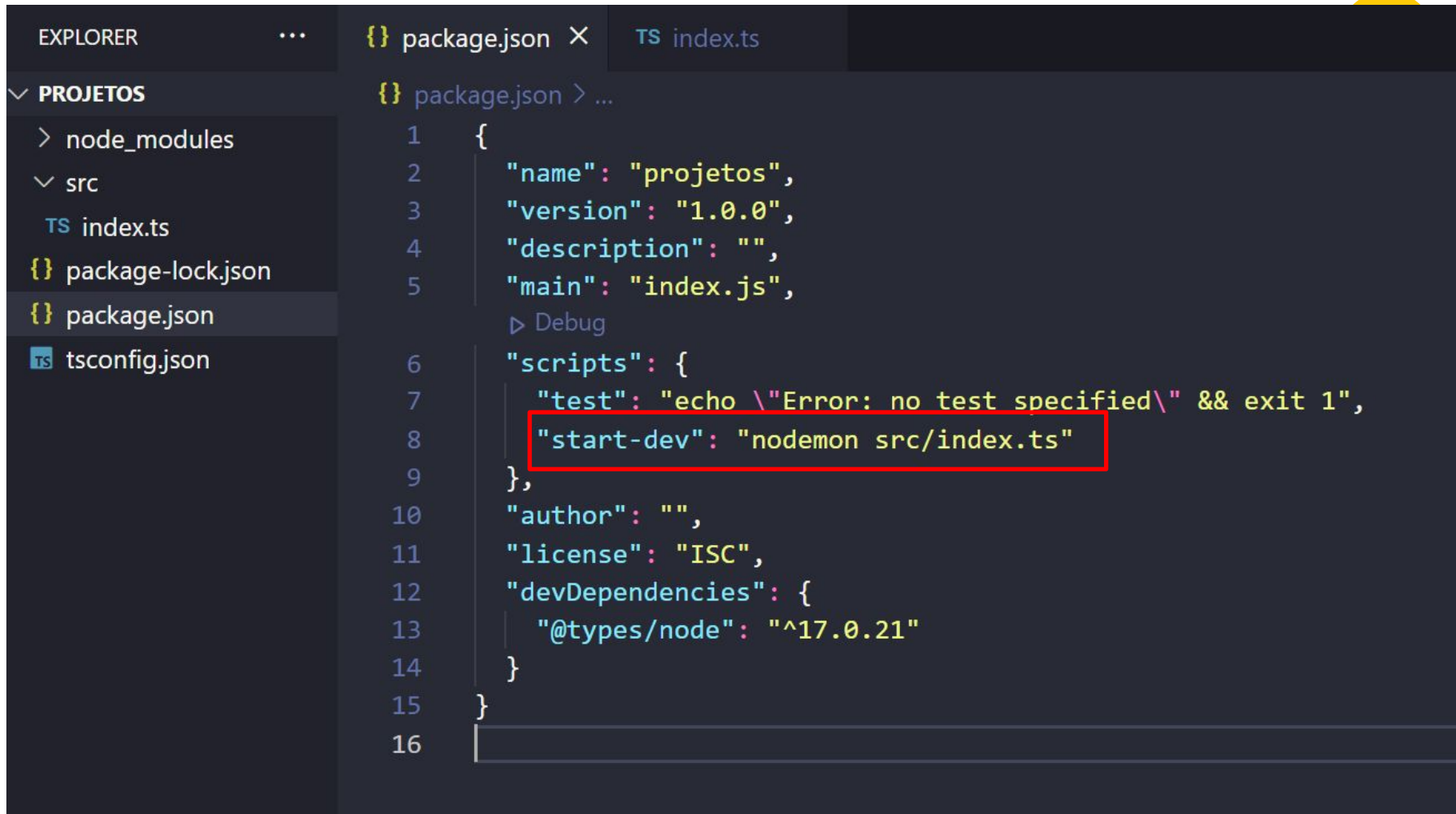
14 }

15 |

Debug

Em package.json temos os scripts, dentro desses “scripts” vamos colocar para rodar alguns atalhos para o nosso terminal

# CRIANDO COMANDO PARA DAR START NO PROJETO



The image shows the Visual Studio Code interface. On the left, the Explorer sidebar is open, showing a project named 'PROJETOS'. Inside the project, there is a 'src' folder containing 'index.ts'. The file explorer also shows 'package-lock.json', 'package.json', and 'tsconfig.json'. The main editor area displays the 'package.json' file. The file content is as follows:

```
1  {
2    "name": "projetos",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "start-dev": "nodemon src/index.ts"
9    },
10   "author": "",
11   "license": "ISC",
12   "devDependencies": {
13     "@types/node": "^17.0.21"
14   }
15 }
```

The line `"start-dev": "nodemon src/index.ts"` is highlighted with a red rectangle.

## TESTE O NOVO SCRIPT NO SERVIDOR DO VSCODE

{} package.json X TS index.ts

{} package.json > ...

```
1  {
2    "name": "projetos",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "start-dev": "nodemon src/index.ts"
9    }
10 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\Viviane\Downloads\projetos> npm run start-dev

> projetos@1.0.0 start-dev

> nodemon src/index.ts

[nodemon] 2.0.15

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

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

[nodemon] watching extensions: ts,json

[nodemon] starting `ts-node src/index.ts`

Meu nome é Kira e eu tenho 25 anos e eu conheço um shinigami

[nodemon] clean exit - waiting for changes before restart

# VAMOS PRATICAR



Crie um projeto novo com NPM, instale o TypeScript, os types do TypeScript, configure o TypeScript, instale o Nodemon e os types do Nodemon.

Documente o passo a passo e suba o projeto criado no Github

