

[Open in app](#)

Search Medium



E

# Como Configurar ESLint e Prettier Para Seus Projetos Em ReactJS, NodeJS e TypeScript

Matheus Teixeira · [Follow](#)

10 min read · Aug 16, 2021

[Share](#)[More](#)

Hello, Dev!

Neste artigo você irá aprender o passo a passo de como configurar o ESLint e o Prettier para seus projetos ReactJS e NodeJS que utilizam TypeScript.

- [Introdução](#)
- [Instalação](#)
  - [ESLint](#)
  - [Prettier](#)

- Problemas no Windows

Caso você queira acompanhar todo esse passo a passo em forma de vídeo, publiquei um vídeo no YouTube onde utilizei esse artigo como guia:

<https://www.youtube.com/watch?v=MPGmsRgqUVQ>

• • •

## INTRODUÇÃO

ESLint é uma ótima ferramenta — se não a melhor — para nos ajudar a automatizar a **padronização do código** em nosso projeto.

Com ela conseguimos **definir regras de padronização, achar códigos fora do padrão e consertá-los automaticamente**.

Por exemplo, no JavaScript, diferentemente de outras linguagens, o uso do ponto e vírgula no final de uma linha é opcional.

O uso de aspas duplas ou aspas simples também é facultativo.

Já quando criamos um objeto, o uso da vírgula no último item também é opcional, como podemos ver no exemplo abaixo:

```
const funcionário = {  
    nome: "Joana",  
    idade: 34,  
};  
  
const funcionário = {  
    nome: "Roberto",  
    idade: 19  
};
```

Neste exemplo, no primeiro objeto nós utilizamos a vírgula após o valor do último atributo, enquanto que no segundo objeto nós deixamos sem a vírgula.

Deixar sem ou com a vírgula no final do valor do último atributo dentro de um objeto não interfere na execução do código, mas o deixa sem uma padronização.

Agora, imagine em um projeto onde existam outros desenvolvedores: se cada um formata seu código de uma maneira individual e única, o resultado disso será uma verdadeira bagunça no seu projeto.

E é aí que o ESLint entra em ação.

Nós vamos **definir regras de formatação**, como por exemplo usar ou não ponto e vírgula ou ter ou não vírgula após o valor do último atributo dentro de um objeto, além de quebra de linhas e etc.

O ESLint é uma ferramenta para você **definir as regras** e te mostrar eventualmente códigos que estão fora das regras de padronização já definidas pela sua equipe.

Neste projeto também vamos utilizar o Prettier, ferramenta que vai **automaticamente** fazer as correções de padrões estabelecidas no ESLint.

• • •

## INSTALAÇÃO

Antes da gente começar a instalação e a configuração do ESLint no nosso projeto, primeiro vamos adicionar a extensão do ESLint no VSCode.

Ela vai interpretar as regras que definimos e mostrar com um risco vermelho em baixo do código que está formatado fora das regras previamente definidas — como se fosse uma palavra estivesse escrita errada no Word.

**ESLint - Visual Studio Marketplace**

Integrates ESLint into VS Code. If you are new to ESLint check the documentation. The extension uses the ESLint library...

[marketplace.visualstudio.com](https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint)



Nosso segundo passo será dizer para o VSCode sempre formatar nosso código após salvarmos o arquivo.

Para isso, aperte CTRL-SHIFT-P (Windows e Linux) ou CMD-SHIFT-P no Mac, e digite >Preferences: Open Settings (JSON).

Neste arquivo JSON, vamos editar a opção "CodeActionsOnSave".

Deixe esta opção como no exemplo abaixo:

```
"editor.codeActionsOnSave": {  
    "source.fixAll.eslint": true  
}
```

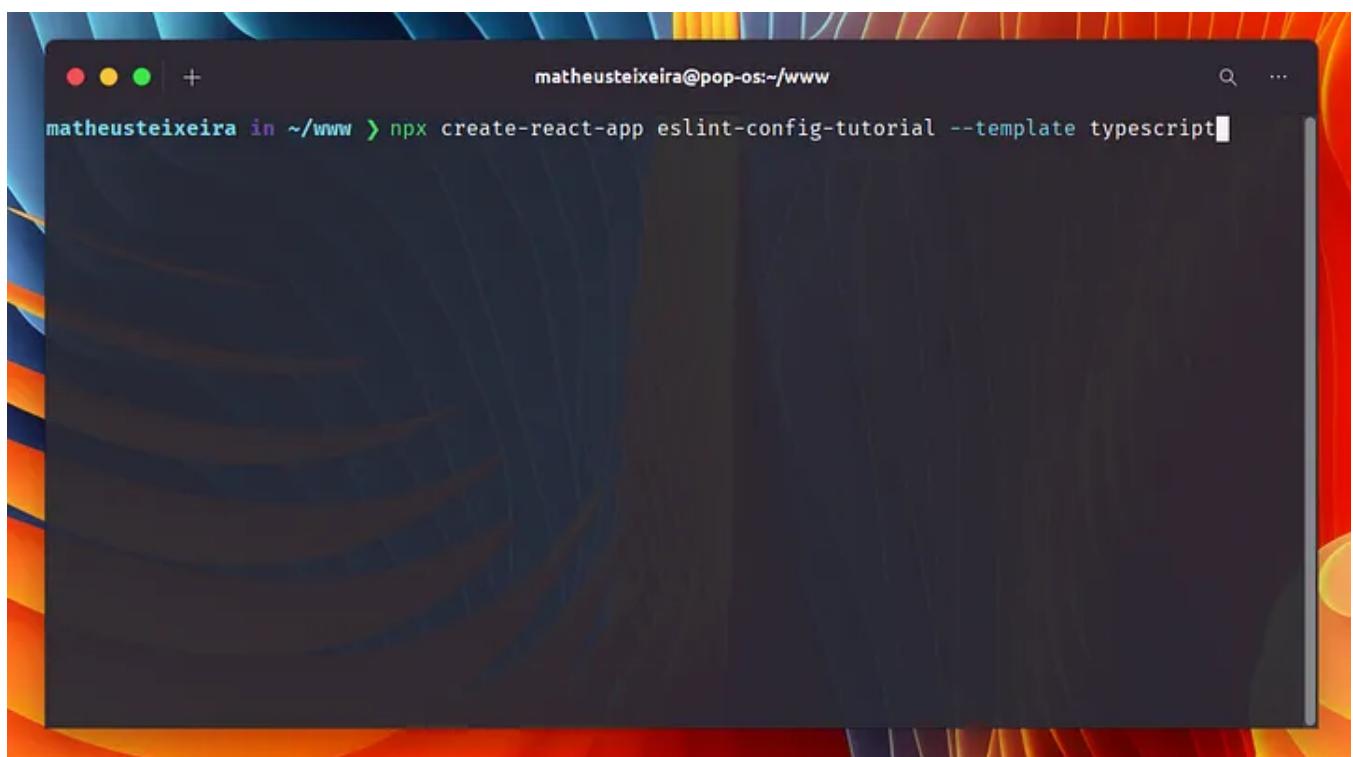
Caso você não tenha esta opção no seu arquivo JSON, sinta-se livre para adicioná-la.

• • •

## ESLINT

Para começarmos, vou criar um novo projeto utilizando React, mas sinta-se livre para incluir estas configurações no seu projeto já existente.

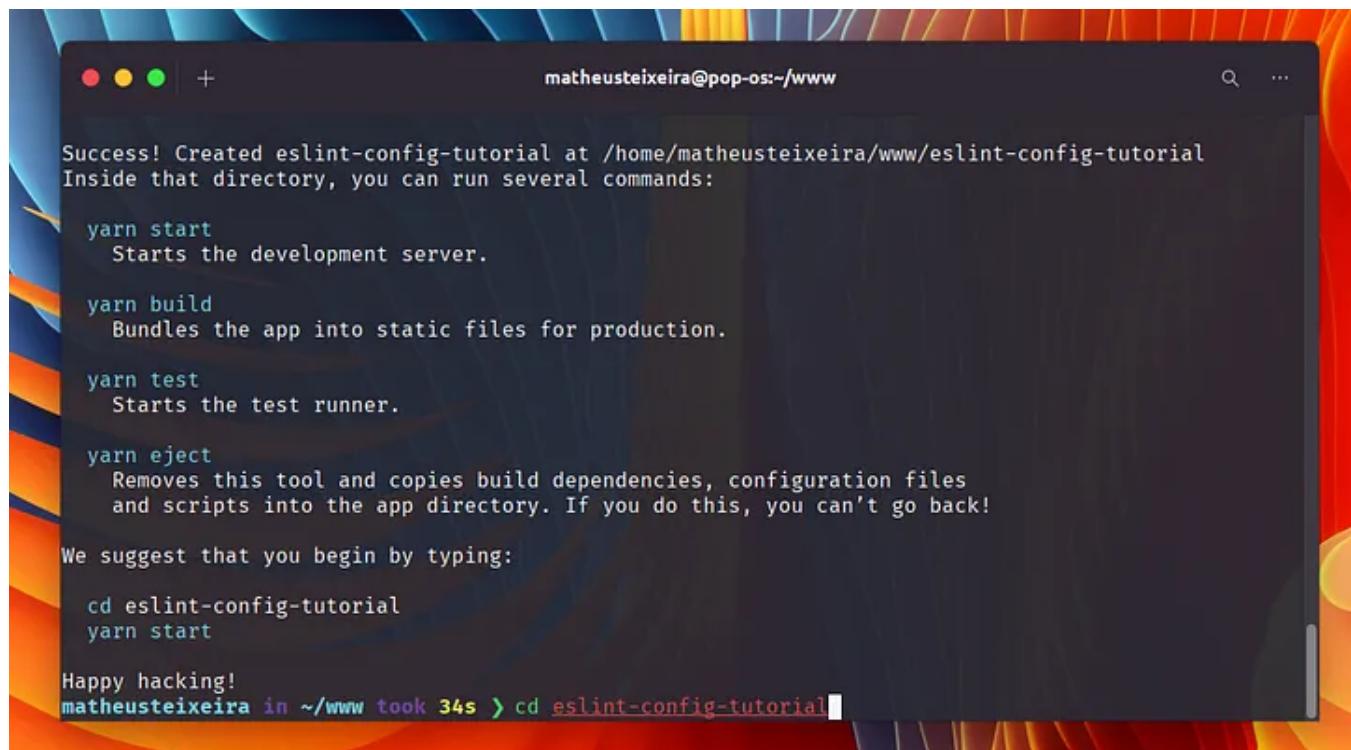
Para isso, vou criar um novo projeto React com o comando:



```
npx create-react-app eslint-config-tutorial --template typescript
```

• • •

Em seguida, vou entrar na pasta do projeto:



```
matheusteixeira@pop-os:~/www
Success! Created eslint-config-tutorial at /home/matheusteixeira/www/eslint-config-tutorial
Inside that directory, you can run several commands:

yarn start
  Starts the development server.

yarn build
  Bundles the app into static files for production.

yarn test
  Starts the test runner.

yarn eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

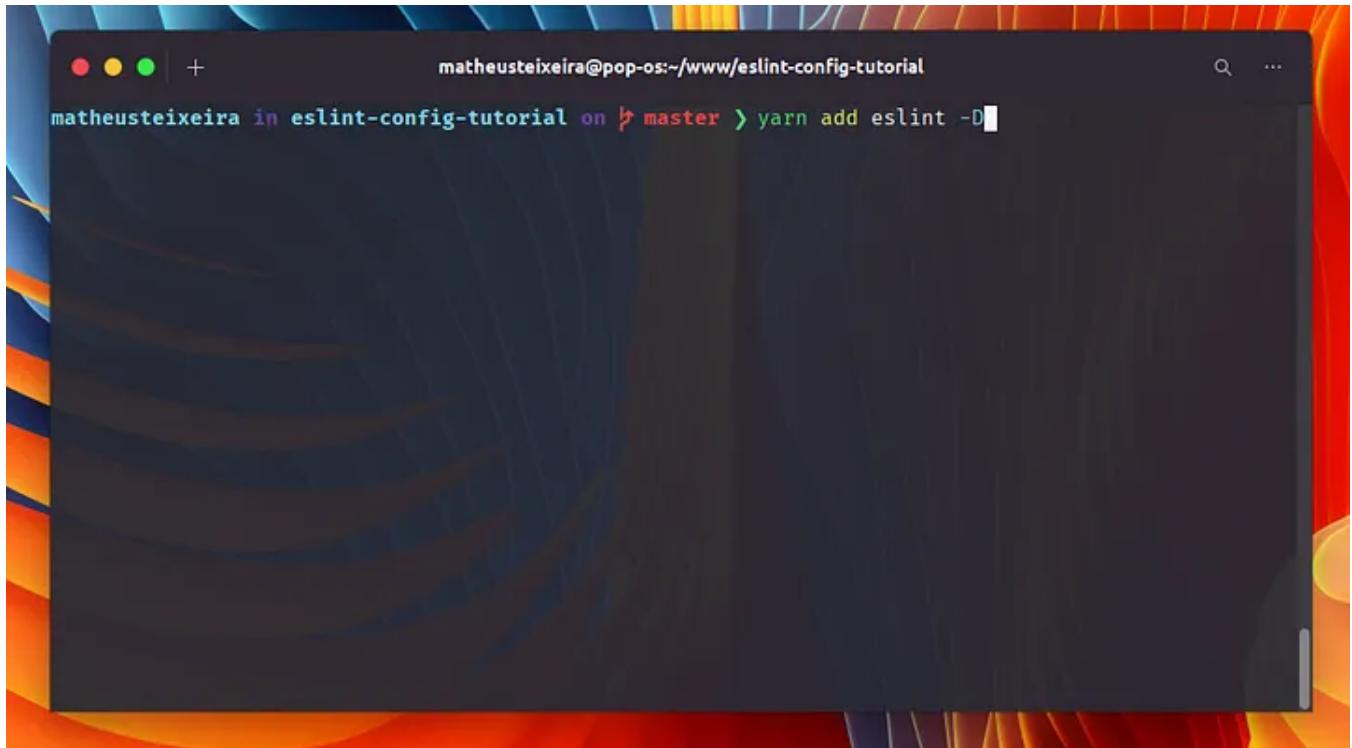
  cd eslint-config-tutorial
  yarn start

Happy hacking!
matheusteixeira in ~/www took 34s > cd eslint-config-tutorial
```

```
cd eslint-config-tutorial
```

• • •

Agora, para começarmos a instalar e configurar o ESLint em nosso projeto, vou instalar o ESLint no projeto como dependência de desenvolvimento utilizando o Yarn:

A screenshot of a terminal window titled "matheusteixeira@pop-os:~/www/eslint-config-tutorial". The window shows the command "yarn add eslint -D" being typed. The background of the terminal is dark, and the text is in a light color.

```
yarn add eslint -D
```

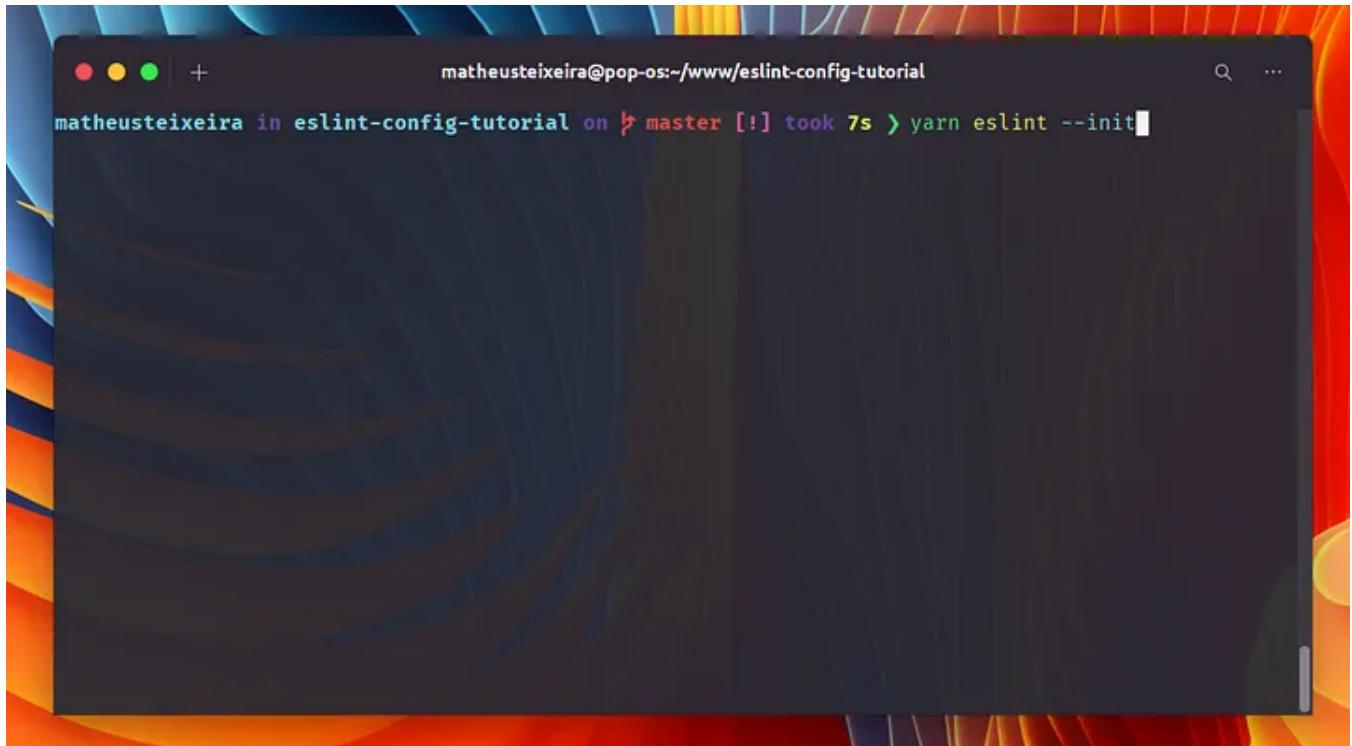
Caso prefira npm:

```
npm install eslint --save-dev
```

• • •

Com o ESLint instalado como dependência de desenvolvimento, vamos iniciá-lo no nosso projeto para iniciarmos a configuração.

Para iniciar, basta dar o seguinte comando:



A screenshot of a terminal window titled "matheusteixeira@pop-os:~/www/eslint-config-tutorial". The command "yarn eslint --init" is being typed at the prompt. The terminal has a dark background with colorful horizontal stripes on the left and right sides.

```
yarn eslint --init
```

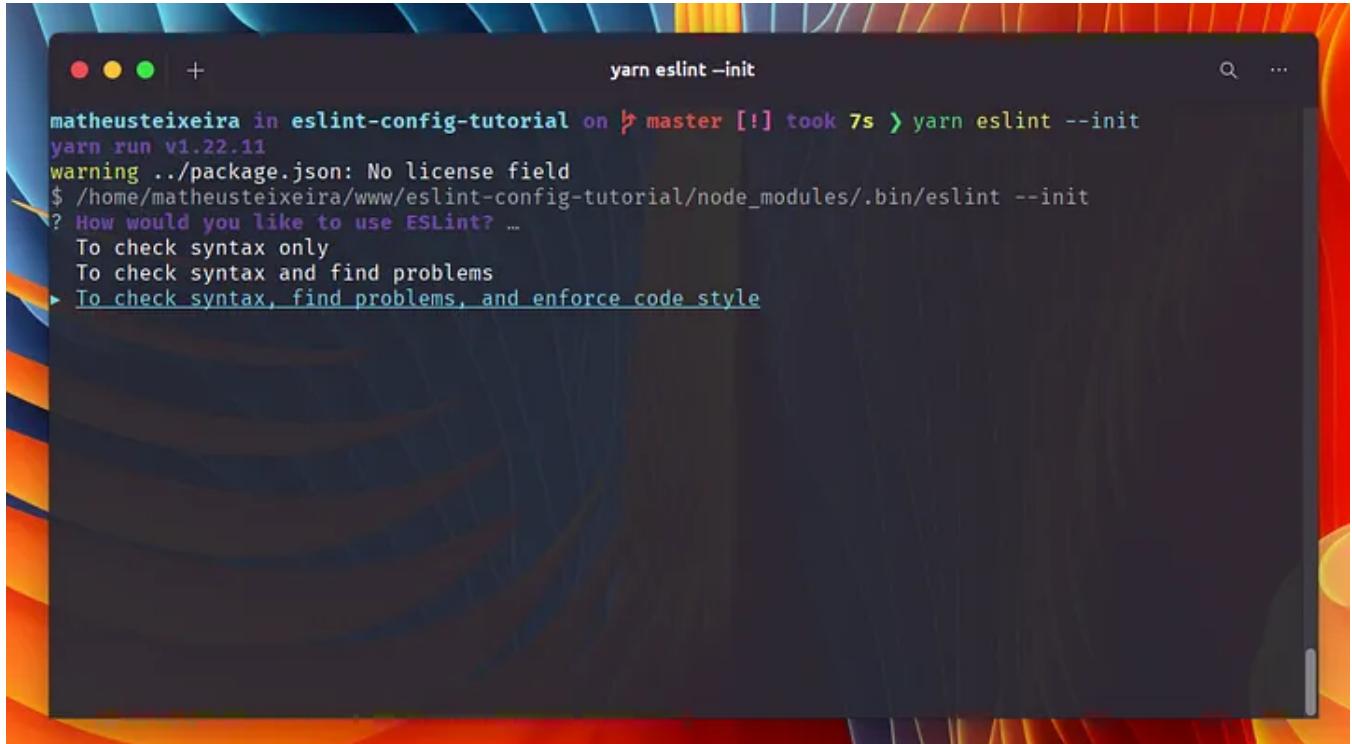
Caso prefira npm:

```
npx eslint --init
```

• • •

Assim que o comando acima for digitado, o ESLint irá te fazer algumas perguntas para a configuração inicial do seu projeto.

*Como você gostaria de usar o ESLint?*

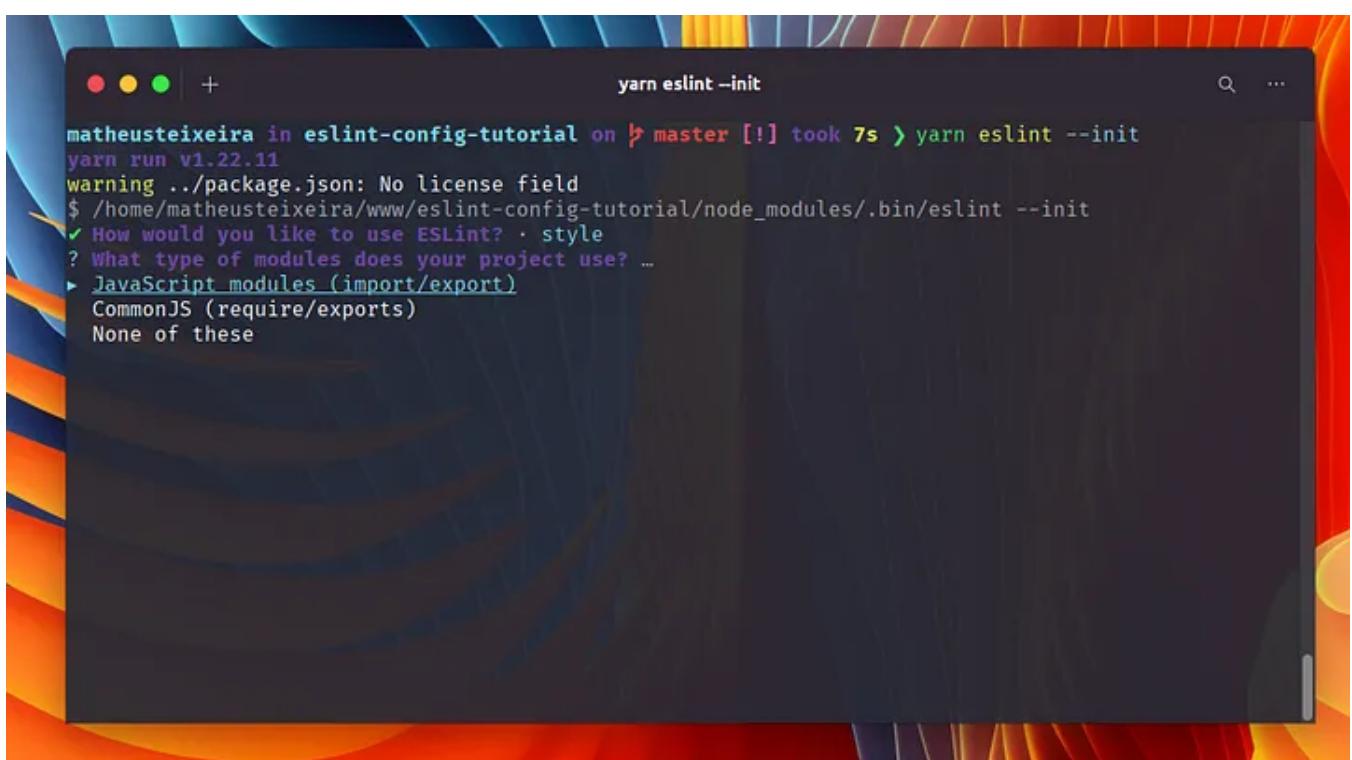


```
yarn eslint --init  
matheusteixeira in eslint-config-tutorial on ↵ master [!] took 7s > yarn eslint --init  
yarn run v1.22.11  
warning .../package.json: No license field  
$ /home/matheusteixeira/www/eslint-config-tutorial/node_modules/.bin/eslint --init  
? How would you like to use ESLint? ...  
  To check syntax only  
  To check syntax and find problems  
▶ To check syntax, find problems, and enforce code style
```

Nós iremos escolher a última opção: "*To check syntax, find problems and enforce code style*".

• • •

Qual tipo de módulo seu projeto usa?

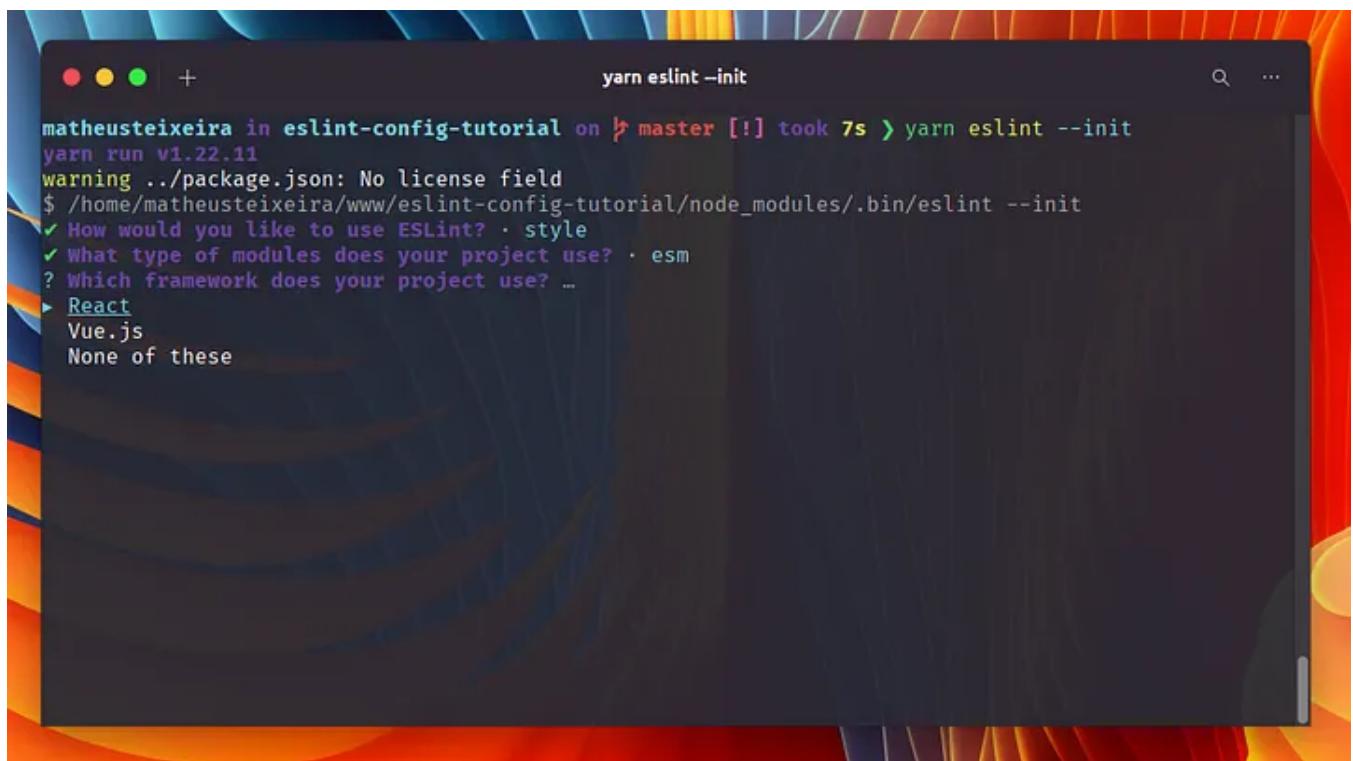


```
yarn eslint --init  
matheusteixeira in eslint-config-tutorial on ↵ master [!] took 7s > yarn eslint --init  
yarn run v1.22.11  
warning .../package.json: No license field  
$ /home/matheusteixeira/www/eslint-config-tutorial/node_modules/.bin/eslint --init  
✓ How would you like to use ESLint? · style  
? what type of modules does your project use? ...  
▶ JavaScript modules (import/export)  
CommonJS (require(exports))  
None of these
```

Vamos selecionar a primeira opção: "*JavaScript modules (import/export)*".

• • •

*Qual framework seu projeto usa?*



A screenshot of a terminal window titled "yarn eslint --init". The window shows the following text:

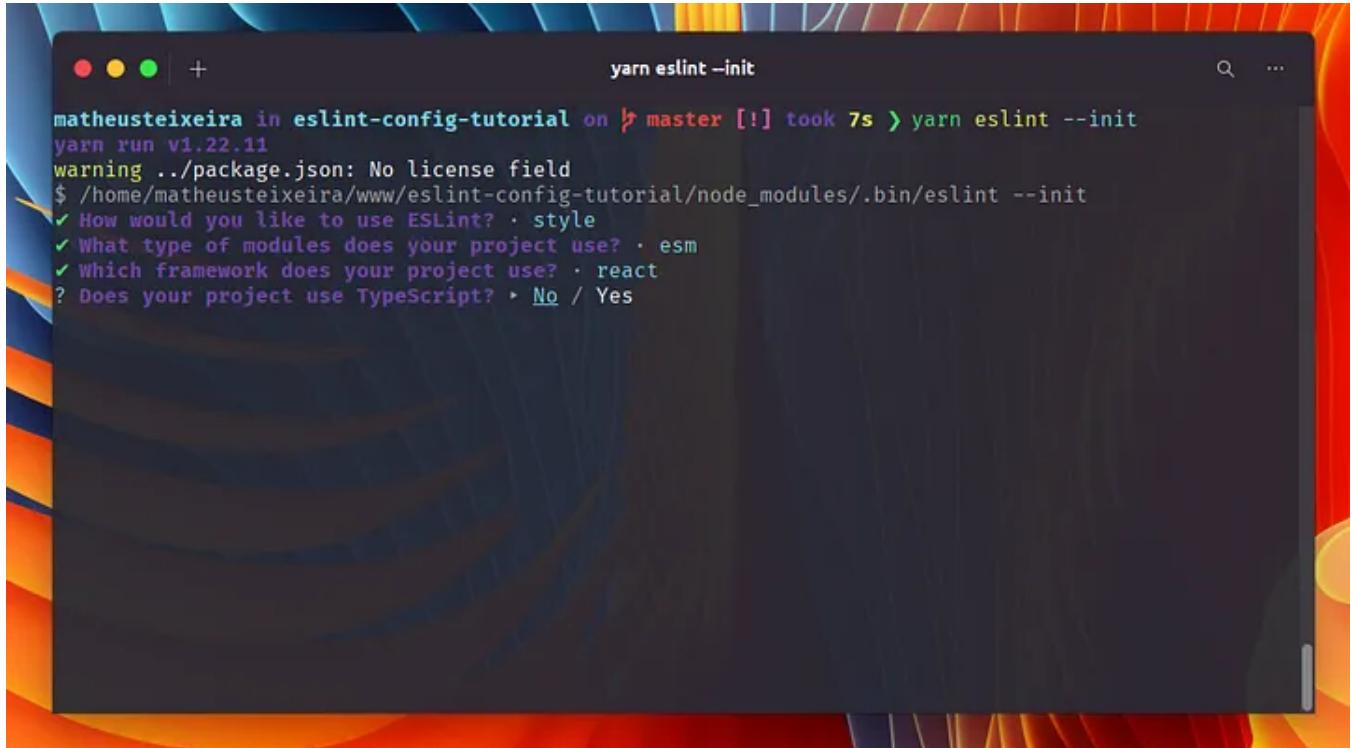
```
matheusteixeira in eslint-config-tutorial on ✪ master [!] took 7s > yarn eslint --init
yarn run v1.22.11
warning .../package.json: No license field
$ /home/matheusteixeira/www/eslint-config-tutorial/node_modules/.bin/eslint --init
✓ How would you like to use ESLint? · style
✓ What type of modules does your project use? · esm
? Which framework does your project use? ...
▶ React
Vue.js
None of these
```

Caso você esteja configurando um projeto NodeJS, você pode escolher a opção "None of these".

Como estou utilizando um projeto React para exemplificar, vou selecionar a opção "React".

• • •

*Seu projeto usa Typescript?*



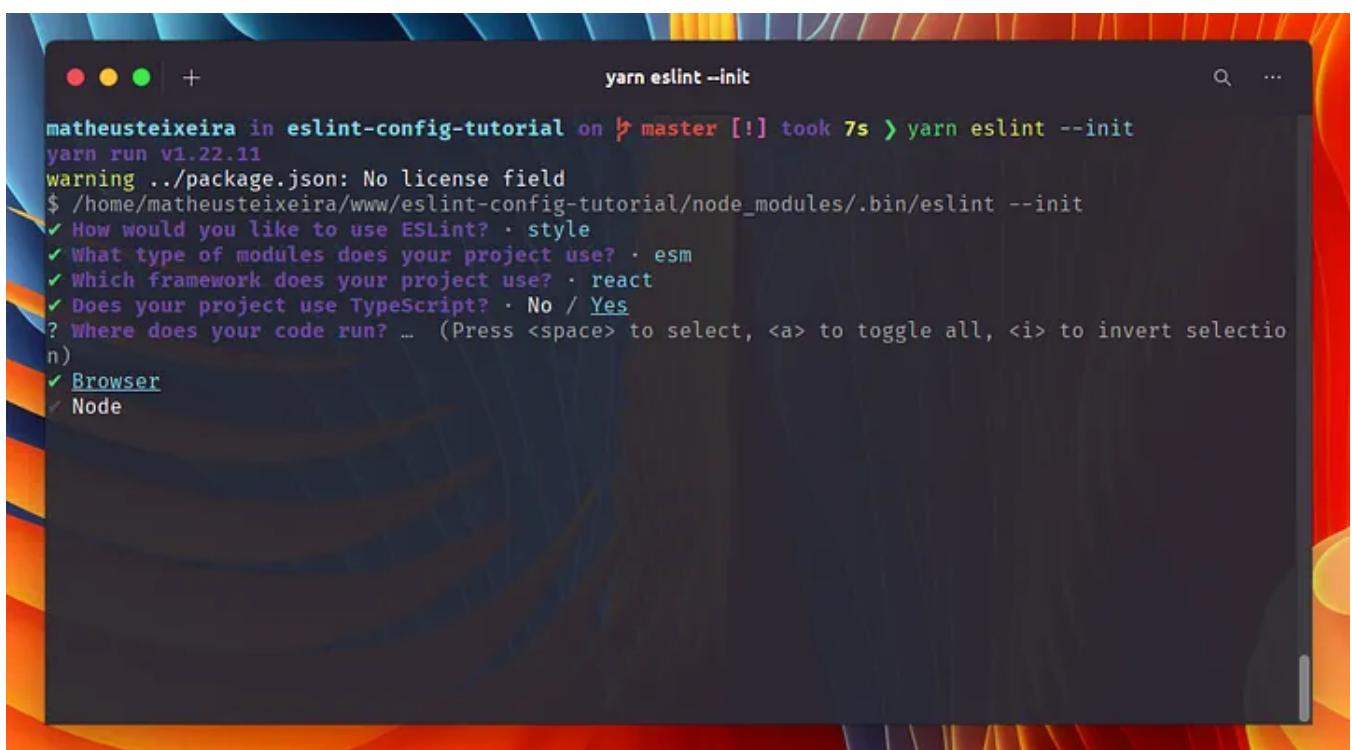
A screenshot of a terminal window titled "yarn eslint --init". The window shows the following text:

```
matheusteixeira in eslint-config-tutorial on ↵ master [!] took 7s > yarn eslint --init
yarn run v1.22.11
warning .../package.json: No license field
$ /home/matheusteixeira/www/eslint-config-tutorial/node_modules/.bin/eslint --init
✓ How would you like to use ESLint? · style
✓ What type of modules does your project use? · esm
✓ Which framework does your project use? · react
? Does your project use TypeScript? · No / Yes
```

Eu sempre utilizo TypeScript nos meus projetos e como proposta desse tutorial, vamos utilizá-lo aqui também.

Vamos escolher a opção "Yes".

Onde seu código está rodando?



A screenshot of a terminal window titled "yarn eslint --init". The window shows the following text:

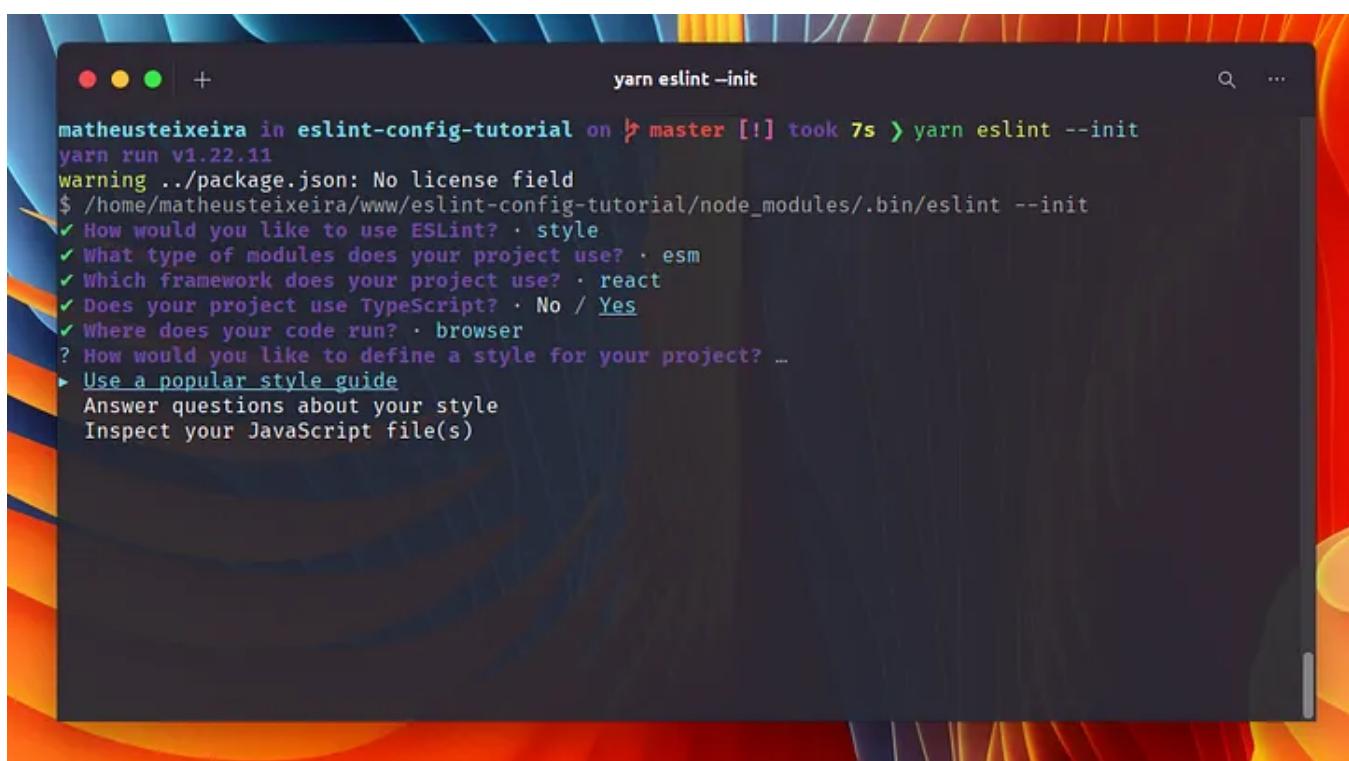
```
matheusteixeira in eslint-config-tutorial on ↵ master [!] took 7s > yarn eslint --init
yarn run v1.22.11
warning .../package.json: No license field
$ /home/matheusteixeira/www/eslint-config-tutorial/node_modules/.bin/eslint --init
✓ How would you like to use ESLint? · style
✓ What type of modules does your project use? · esm
✓ Which framework does your project use? · react
✓ Does your project use TypeScript? · No / Yes
? Where does your code run? ... (Press <space> to select, <a> to toggle all, <i> to invert selection)
✓ Browser
✓ Node
```

Caso você esteja configurando um projeto NodeJS, utilize a tecla "espaço" e marque a opção **Node** e desmarque a opção **Browser**.

Como neste exemplo estamos configurando um projeto React, vamos deixar ativo somente a opção "*Browser*".

• • •

*Qual guia de estilo queremos utilizar?*



```
yarn eslint --init
matheusteixeira in eslint-config-tutorial on ↵ master [!] took 7s > yarn eslint --init
yarn run v1.22.11
warning .../package.json: No license field
$ /home/matheusteixeira/www/eslint-config-tutorial/node_modules/.bin/eslint --init
✓ How would you like to use ESLint? · style
✓ What type of modules does your project use? · esm
✓ Which framework does your project use? · react
✓ Does your project use TypeScript? · No / Yes
✓ Where does your code run? · browser
? How would you like to define a style for your project? ...
▶ Use a popular style guide
Answer questions about your style
Inspect your JavaScript file(s)
```

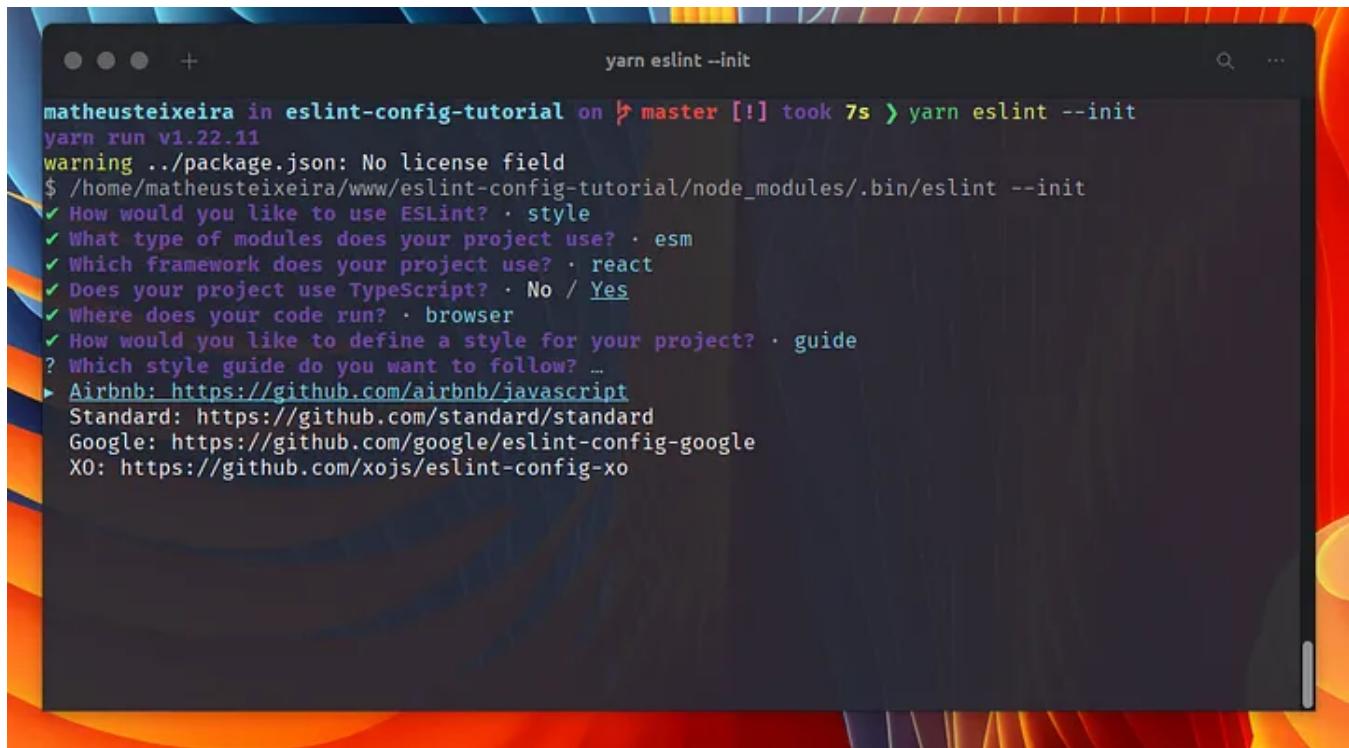
Com a opção "**Use a popular style guide**", você criará seu projeto com padrões criados por outras empresas, como Google e AirBnB.

Na opção "**Answer questions about your style**", você criará seu próprio padrão.

Nós vamos utilizar a opção "*Use a popular style guide*".

• • •

*Qual guia de estilo você deseja seguir?*



A screenshot of a terminal window titled "yarn eslint --init". The terminal is running on a Mac OS X desktop, indicated by the Dock icons at the bottom. The command "yarn eslint --init" is being run, and the output shows a series of configuration questions and their answers:

```
matheusteixeira in eslint-config-tutorial on ✘ master [!] took 7s > yarn eslint --init
yarn run v1.22.11
warning .../package.json: No license field
$ /home/matheusteixeira/www/eslint-config-tutorial/node_modules/.bin/eslint --init
✓ How would you like to use ESLint? · style
✓ What type of modules does your project use? · esm
✓ Which framework does your project use? · react
✓ Does your project use TypeScript? · No / Yes
✓ Where does your code run? · browser
✓ How would you like to define a style for your project? · guide
? Which style guide do you want to follow? ...
▶ Airbnb: https://github.com/airbnb/javascript
Standard: https://github.com/standard/standard
Google: https://github.com/google/eslint-config-google
XO: https://github.com/xojs/eslint-config-xo
```

Cada empresa desta tem seu próprio padrão e elas compartilham com a comunidade seus padrões.

A do Google por exemplo tem por regra não ter ponto e vírgula, deixando o código mais enxuto e simples, enquanto a AirBnB tem por padrão utilizar ponto e vírgula, dentre dezenas de outras regras diferentes.

Particularmente eu prefiro o estilo AirBnB, porém como já dito, não existe padrão certo.

Neste artigo nós iremos utilizar o padrão "AirBnB".

• • •

---

*Em qual formato você quer que o arquivo de configuração venha?*

A screenshot of a terminal window titled "yarn eslint --init". The window shows the following text:

```
matheusteixeira in eslint-config-tutorial on ↵ master [!] took 7s > yarn eslint --init
yarn run v1.22.11
warning .../package.json: No license field
$ /home/matheusteixeira/www/eslint-config-tutorial/node_modules/.bin/eslint --init
✓ How would you like to use ESLint? · style
✓ What type of modules does your project use? · esm
✓ Which framework does your project use? · react
✓ Does your project use TypeScript? · No / Yes
✓ Where does your code run? · browser
✓ How would you like to define a style for your project? · guide
✓ Which style guide do you want to follow? · airbnb
? What format do you want your config file to be in? ...
  JavaScript
  YAML
> JSON
```

Vamos selecionar o formato "**JSON**".

• • •

O ESLint irá verificar quais dependências você precisa para rodar o projeto a partir das configurações que você definiu acima e irá perguntar se você quer instalar elas com o NPM.

A screenshot of a terminal window titled "yarn eslint --init". The window shows the following text:

```
matheusteixeira in eslint-config-tutorial on ↵ master [!] took 7s > yarn eslint --init
yarn run v1.22.11
warning .../package.json: No license field
$ /home/matheusteixeira/www/eslint-config-tutorial/node_modules/.bin/eslint --init
✓ How would you like to use ESLint? · style
✓ What type of modules does your project use? · esm
✓ Which framework does your project use? · react
✓ Does your project use TypeScript? · No / Yes
✓ Where does your code run? · browser
✓ How would you like to define a style for your project? · guide
✓ Which style guide do you want to follow? · airbnb
✓ What format do you want your config file to be in? · JSON
Checking peerDependencies of eslint-config-airbnb@latest
The config that you've selected requires the following dependencies:

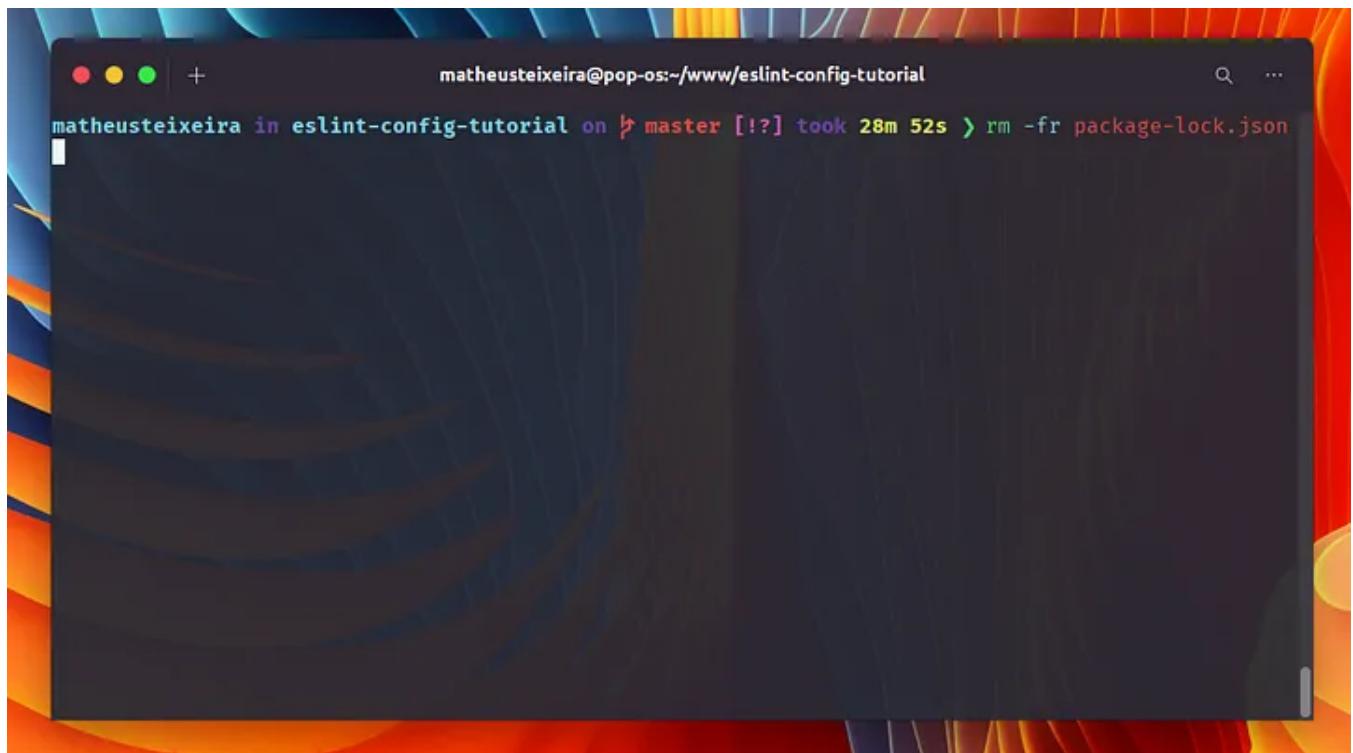
eslint-plugin-react@^7.21.5 @typescript-eslint/eslint-plugin@latest eslint-config-airbnb@latest eslint@^5.16.0 || ^6.8.0 || ^7.2.0 eslint-plugin-import@^2.22.1 eslint-plugin-jsx-a11y@^6.4.1 eslint-plugin-react-hooks@^4 || ^3 || ^2.3.0 || ^1.7.0 @typescript-eslint/parser@latest
? Would you like to install them now with npm? > No / Yes
```

Esta parte modifica um pouco para quem usa **Yarn** e para quem usa **NPM**.

Para ambos os gerenciadores de pacote, selecione a opção "**Yes**".

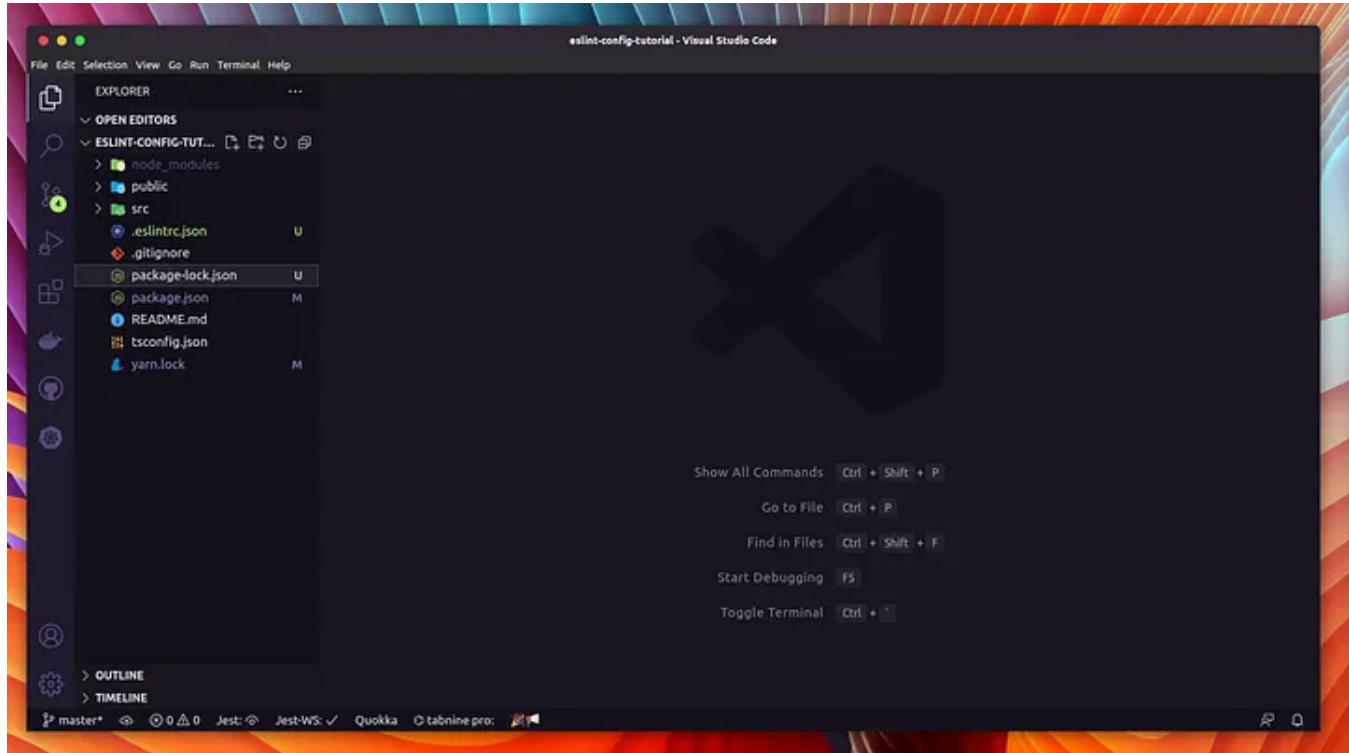
Porém, após os pacotes serem instalados, caso você esteja usando **Yarn**, exclua o arquivo "**package-lock.json**"

*Você pode excluir este arquivo pelo terminal executando o comando abaixo:*



```
rm -fr package-lock.json
```

*Ou então abrindo a pasta do projeto com o seu VSCode, e clicando com o botão direito em cima do arquivo, selecionar a opção "**Delete**".*



Depois de excluir o arquivo `package-lock.json`, execute o comando abaixo:

```
matheusteixeira@pop-os:~/www/eslint-config-tutorial
yarn install v1.22.11
warning .../package.json: No license field
[1/4] ⚡ Resolving packages...
[2/4] 🏛 Fetching packages...
info fsevents@2.3.2: The platform "linux" is incompatible with this module.
info "fsevents@2.3.2" is an optional dependency and failed compatibility check. Excluding it from installation.
info fsevents@1.2.13: The platform "linux" is incompatible with this module.
info "fsevents@1.2.13" is an optional dependency and failed compatibility check. Excluding it from installation.
[3/4] ⚡ Linking dependencies...
warning " > @testing-library/user-event@12.8.3" has unmet peer dependency "@testing-library/dom@>=7.21.4".
[4/4] 🏛 Building fresh packages...
success Saved lockfile.
⚡ Done in 17.00s.
```

yarn

Dessa forma nós iremos atualizar o arquivo `yarn.lock` com a versão das dependências que acabaram de serem adicionadas.

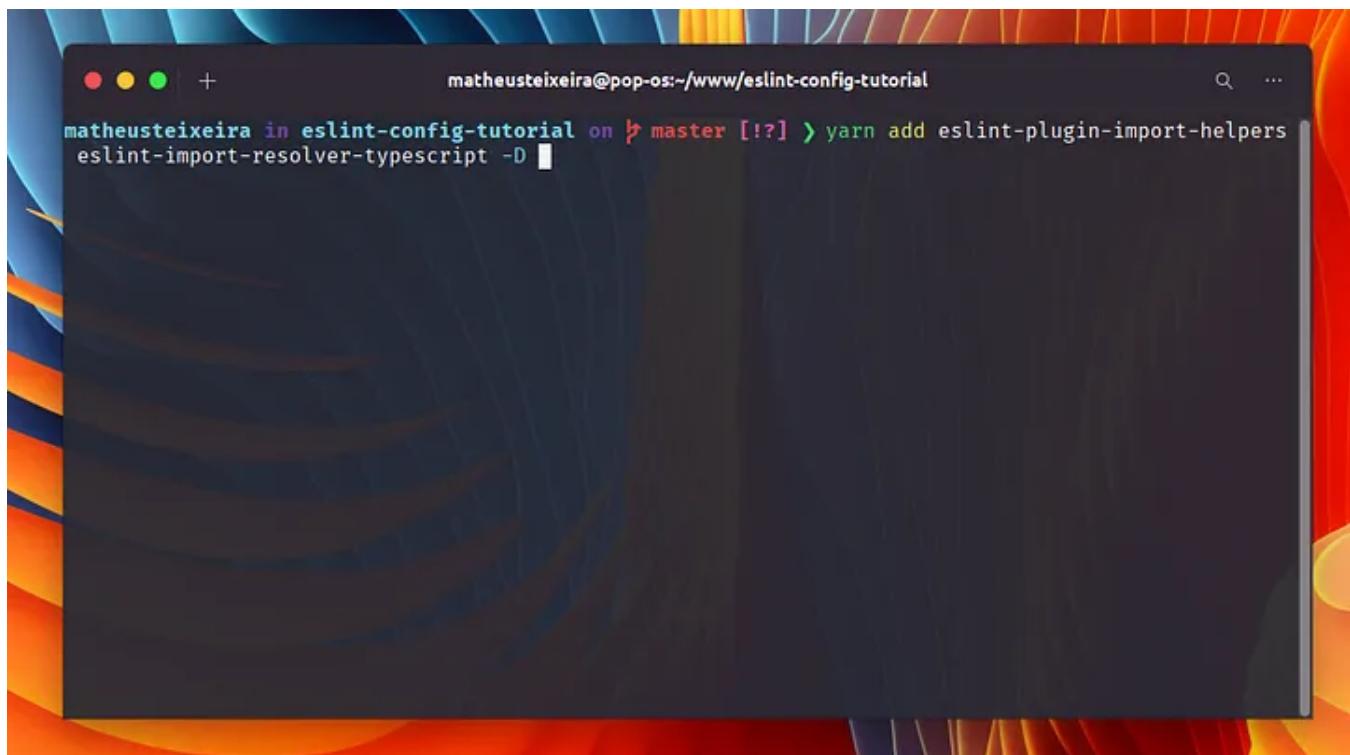
Tanto o arquivo `package-lock.json` quanto o arquivo `yarn.lock` são arquivos que guardam em seus registros de versionamento das dependências instaladas.

O que nós fizemos foi fazer com que o versionamento das dependências que acabaram de ser instaladas fossem manejadas pelo Yarn e não mais pelo NPM.

Certifique-se que você fez esta parte de forma correta. Caso contrário seu ESLint não funcionará.

• • •

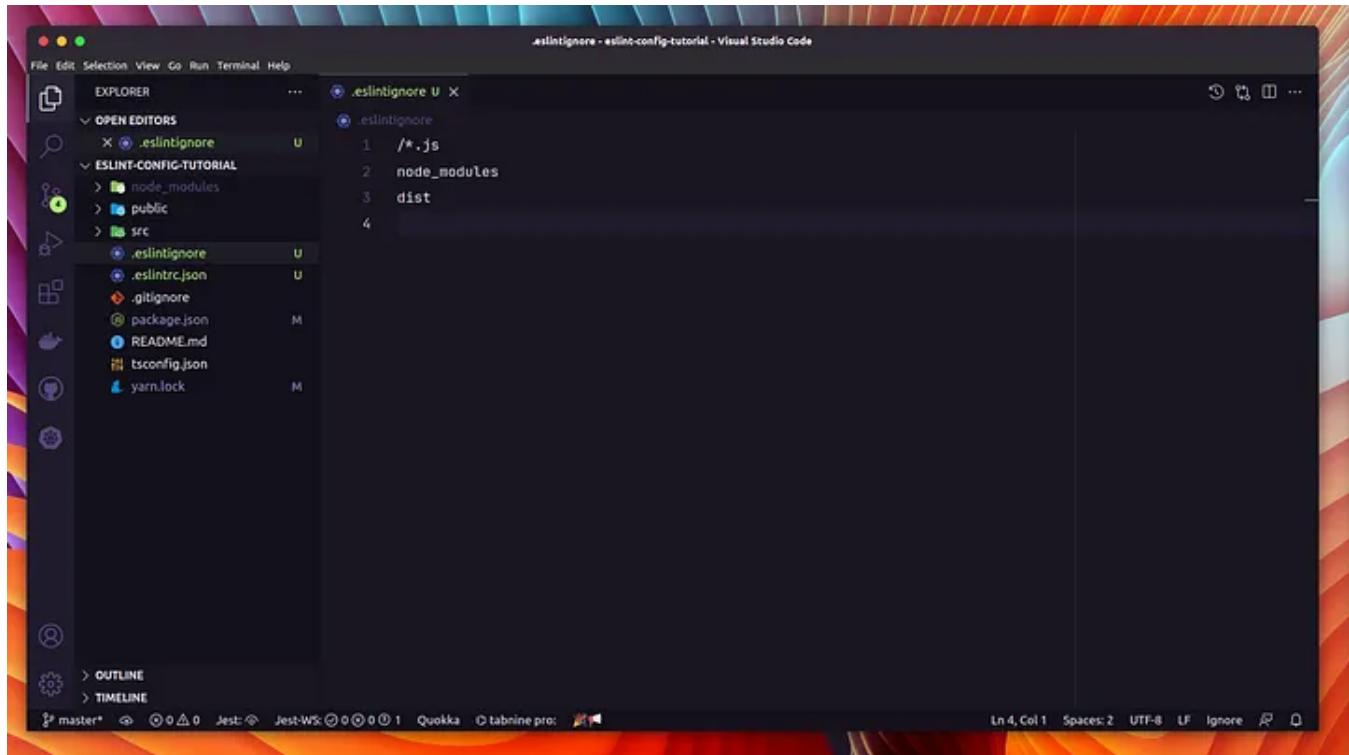
Agora vamos instalar um plugin para que organize a ordem dos imports dentro dos arquivos e outro plugin que irá permitir para a gente importar arquivos TypeScript sem que precisemos passar a extensão do arquivo.



```
matheusteixeira@pop-os:~/www/eslint-config-tutorial
matheusteixeira in eslint-config-tutorial on master [!?] > yarn add eslint-plugin-import-helpers
eslint-import-resolver-typescript -D
```

```
yarn add eslint-plugin-import-helpers eslint-import-resolver-
typescript -D
```

Com esses plugins instalados, vamos no VSCode e vamos criar um arquivo chamado `.eslintignore` na raiz do nosso projeto para ignorar o linting em alguns arquivos e pastas. Vamos adicionar o seguinte conteúdo:



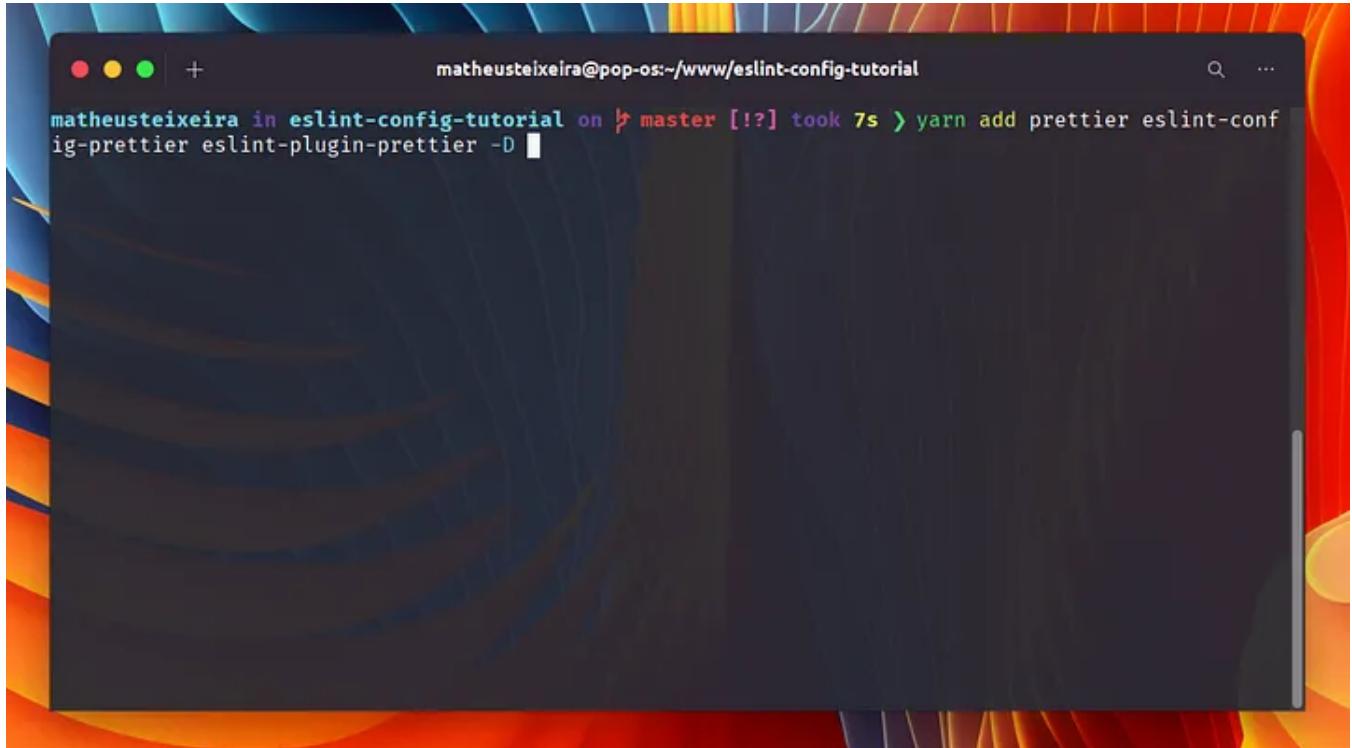
```
/*.js
node_modules
dist
```

• • •

## PRETTIER

Importante ressaltar a **necessidade de desinstalarmos a extensão Prettier – Code Formatter** no nosso VSCode antes de continuarmos com a configuração do Prettier, pois a extensão **pode causar incompatibilidades** com as configurações que iremos realizar a seguir.

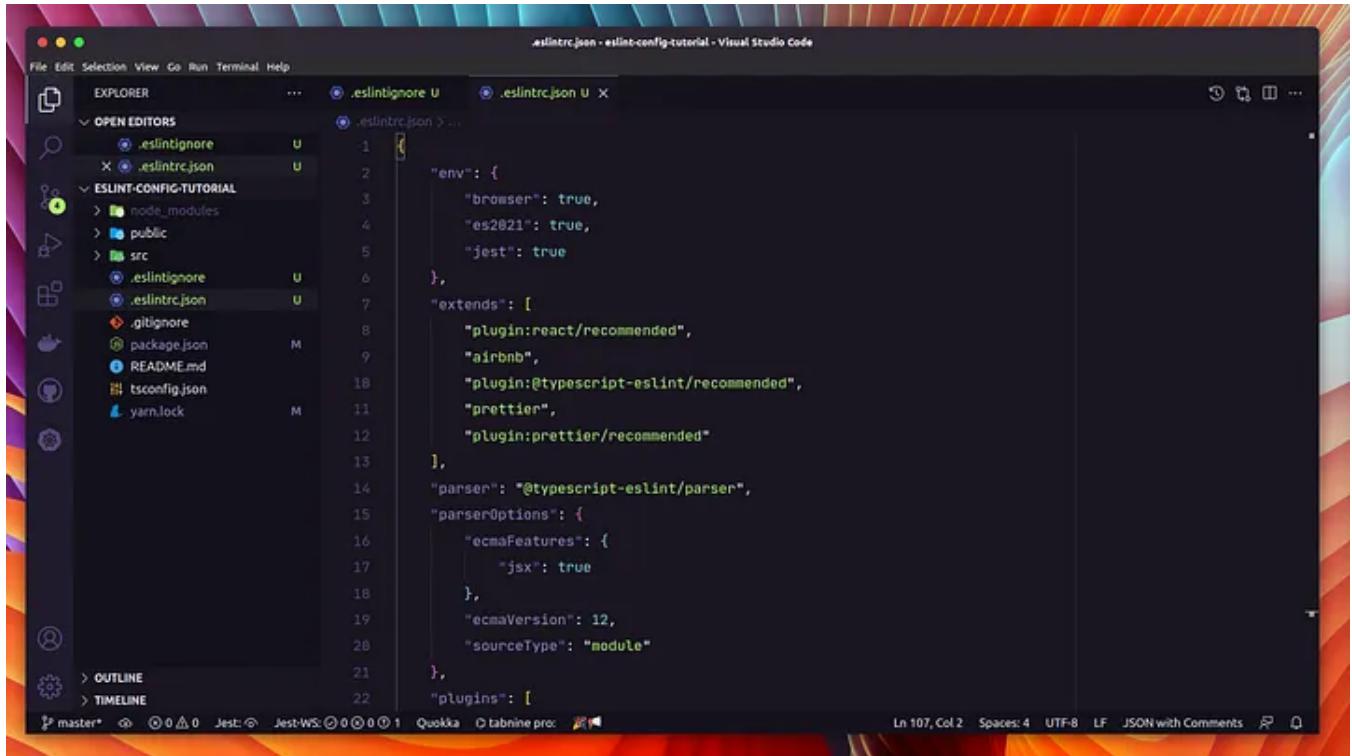
O que vamos fazer agora é instalar o Prettier e outras duas dependências que irão formatar o código baseado nas nossas regras do ESLint e também integrar o Prettier com o ESLint.



```
matheusteixeira@pop-os:~/www/eslint-config-tutorial matheusteixeira in eslint-config-tutorial on master [!?] took 7s > yarn add prettier eslint-config-prettier eslint-plugin-prettier -D
```

```
yarn add prettier eslint-config-prettier eslint-plugin-prettier -D
```

Com todas essas instalações prontas, vamos no nosso arquivo **eslintrc.json** e vamos substituir pelo código abaixo:



```
1  {
2     "env": {
3         "browser": true,
4         "es2021": true,
5         "jst": true
6     },
7     "extends": [
8         "plugin:react/recommended",
9         "airbnb",
10        "plugin:@typescript-eslint/recommended",
11        "prettier",
12        "plugin:prettier/recommended"
13    ],
14    "parser": "@typescript-eslint/parser",
15    "parserOptions": {
16        "ecmaFeatures": {
17            "jsx": true
18        },
19        "ecmaVersion": 12,
20        "sourceType": "module"
21    },
22    "plugins": [

```

{

```
"env": {  
  "browser": true,  
  "es2021": true,  
  "jest": true  
},  
"extends": [  
  "plugin:react/recommended",  
  "airbnb",  
  "plugin:@typescript-eslint/recommended",  
  "prettier",  
  "plugin:prettier/recommended"  
],  
"parser": "@typescript-eslint/parser",  
"parserOptions": {  
  "ecmaFeatures": {  
    "jsx": true  
  },  
  "ecmaVersion": 12,  
  "sourceType": "module"  
},  
"plugins": [  
  "react",  
  "@typescript-eslint",  
  "eslint-plugin-import-helpers",  
  "prettier"  
,  
  "rules": {  
    "camelcase": "off",  
    "import/no-unresolved": "error",  
    "@typescript-eslint/naming-convention": [  
      "error",  
      {  
        "selector": "interface",  
        "name": "PascalCase"  
      }  
    ]  
  }  
}
```

```
"format": [  
  "PascalCase"  
,  
  "custom": {  
    "regex": "^I[A-Z]",  
    "match": true  
  }  
}  
,  
  "class-methods-use-this": "off",  
  "import/prefer-default-export": "off",  
  "no-shadow": "off",  
  "no-console": "off",  
  "no-useless-constructor": "off",  
  "no-empty-function": "off",  
  "lines-between-class-members": "off",  
  "import/extensions": [  
    "error",  
    "ignorePackages",  
    {  
      "ts": "never",  
      "tsx": "never"  
    }  
,  
    "import-helpers/order-imports": [  
      "warn",  
      {  
        "newlinesBetween": "always",  
        "groups": [  
          "module",  
          "/^@shared/",  
          [  
            "parent",  
            "parent"

```

```
"sibling",
"index"
]
],
"alphabetize": {
"order": "asc",
"ignoreCase": true
}
}
],
"import/no-extraneous-dependencies": [
"error",
{
"devDependencies": [
"**/*.spec.js"
]
}
],
"prettier/prettier": "error",
"react/jsx-filename-extension": [
1,
{
"extensions": [
".jsx",
".tsx"
]
}
],
"react/react-in-jsx-scope": "off",
"no-use-before-define": "off",
"@typescript-eslint/no-use-before-define": [
"error"
]
```

```
},  
  "settings": {  
    "import/resolver": {  
      "typescript": {}  
    }  
  }  
}
```

Caso você esteja configurando um projeto NodeJS, dentro de "env", **mude** o atributo "**Browser**" para "**Node**".

Esse arquivo contém as regras que utilizo nos meus projetos ReactJS e NodeJS.

Em "**extends**" a gente habilita que estamos usando as regras utilizadas na AirBnB.

Em "**plugins**" temos ferramentas adicionais que irão integrar o ESLint com o TypeScript e nosso outro plugin que irá nos auxiliar na ordem que os imports devem ter dentro dos nossos arquivos.

Dentro de "**rules**" nós temos regras adicionais que queremos dar ao nosso projeto.

Essas são as regras que eu uso, mas caso você queira remover ou adicionar outras, fique à vontade.

Como já disse anteriormente, **não existe padrão correto**.

Existe o padrão que você e sua equipe definiram que irão utilizar no projeto.

• • •

Agora para a gente ver nosso ESLint funcionando, vamos **fechar** o nosso VSCode e abri-lo novamente.

Vamos perceber que agora nossos arquivos .ts e .tsx já estarão indicando erros de formatação.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files like .eslintignore, .eslintrc.json, App.tsx, index.css, logo.svg, etc.
- Editor:** The code editor has several red squiggly underlines under words like "import", "React", "from", "header", "img", "p", "a", "className", "App-link", "href", "target", "rel", "Learn React", and "header".
- Status Bar:** Shows "master+", "Jest: 0", "Quokka", "Tabnine pro", "You, an hour ago", "Ln 15, Col 37", "Spaces: 2", "UTF-8", "LF", "TypeScript React", "4.3.5".

Para corrigi-los automaticamente, é só você salvar o arquivo.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Same as the previous screenshot.
- Editor:** The code editor now has fewer red squiggly underlines, indicating some corrections have been made.
- Status Bar:** Shows "master+", "Jest: 0", "Quokka", "Tabnine pro", "You, an hour ago", "Ln 3, Col 1", "Spaces: 2", "UTF-8", "LF", "TypeScript React", "4.3.5".

## PROBLEMAS NO WINDOWS

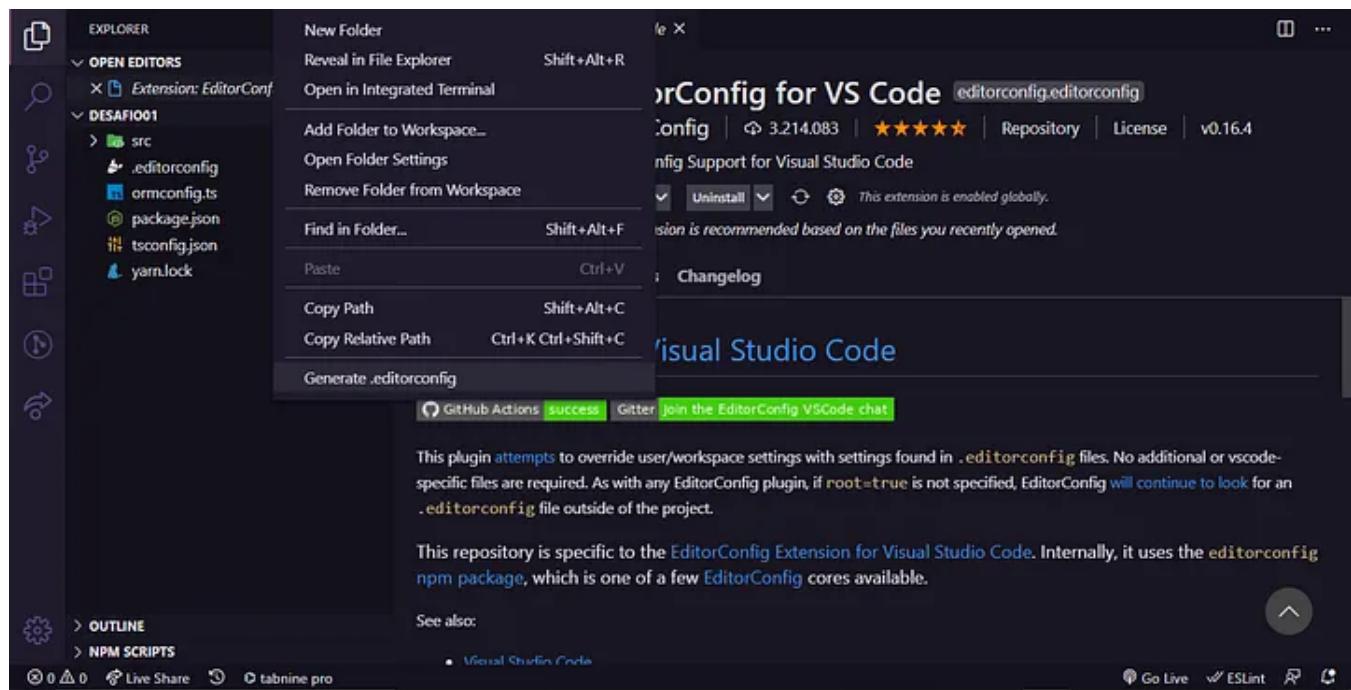
É provável que você enfrente alguns problemas de conflito entre o ESLint e o tipo de quebra de linha no Windows.

Isso acontece porque quando usamos a quebra de linha no Windows, ela é interpretada como `\r\n` enquanto em sistemas Unix é `\n`.

Como o ESLint tenta sempre corrigir para `\n`, esse conflito acaba acontecendo.

Para padronizar o tipo de quebra de linha usada pelo VS Code no Windows, iremos instalar uma extensão chamada [EditorConfig for VS Code](#).

Com ela instalada, na pasta raiz dos nossos projetos podemos clicar com o botão direito do mouse e escolher a opção `Generate .editorconfig`:



Com o arquivo criado você já está pronto para continuar. Todas as quebras de linha estarão no formato esperado pelo ESLint.

## CONCLUSÃO

Então é isso.

Espero ter contribuído para que você e sua equipe possam ter uma padronização no código do projeto de vocês e que possam agilizar o desenvolvimento do mesmo.

E você, tem alguma regra que gosta de utilizar nos seus projetos que não estão nas que eu passei?

Deixe aqui nos comentários quais regras você prefere e qual guia é a sua preferida.

Caso tenha tipo algum problema durante a instalação e/ou o ESLint atualizou e este guia não funcione mais, me mande mensagem que irei atualizar o mais rápido possível =)

Caso você tenha tido sucesso com este artigo, lembre-se de compartilhá-lo com a comunidade, para assim como você pode obter valor com este artigo, outros também possam.

Vou ficando por aqui, abraços.

Eslint

Prettier

Reactjs

Nodejs

JavaScript



Follow



## Written by Matheus Teixeira

18 Followers

Full Stack Developer

## Recommended from Medium



Muhammad Nuh Ibnu Aimar Tamir

## Efficient React Project Setup with Vite, ESLint, Prettier, and Husky

Setting up a React project with the right tools and configurations can greatly improve the development experience and code quality. In this...

3 min read · Jun 26



25



...



Leandro A. Siqueira

# Setting Up ESLint, Prettier, and Husky in a React 18 Project: A Comprehensive Guide—The update...

As the React ecosystem evolves, it's crucial to maintain code quality and consistency in your projects. Linting and code formatting tools...

3 min read · Jul 18

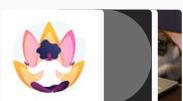


1



...

## Lists



### Stories to Help You Grow as a Software Developer

19 stories · 451 saves



### It's never too late or early to start something

15 stories · 161 saves



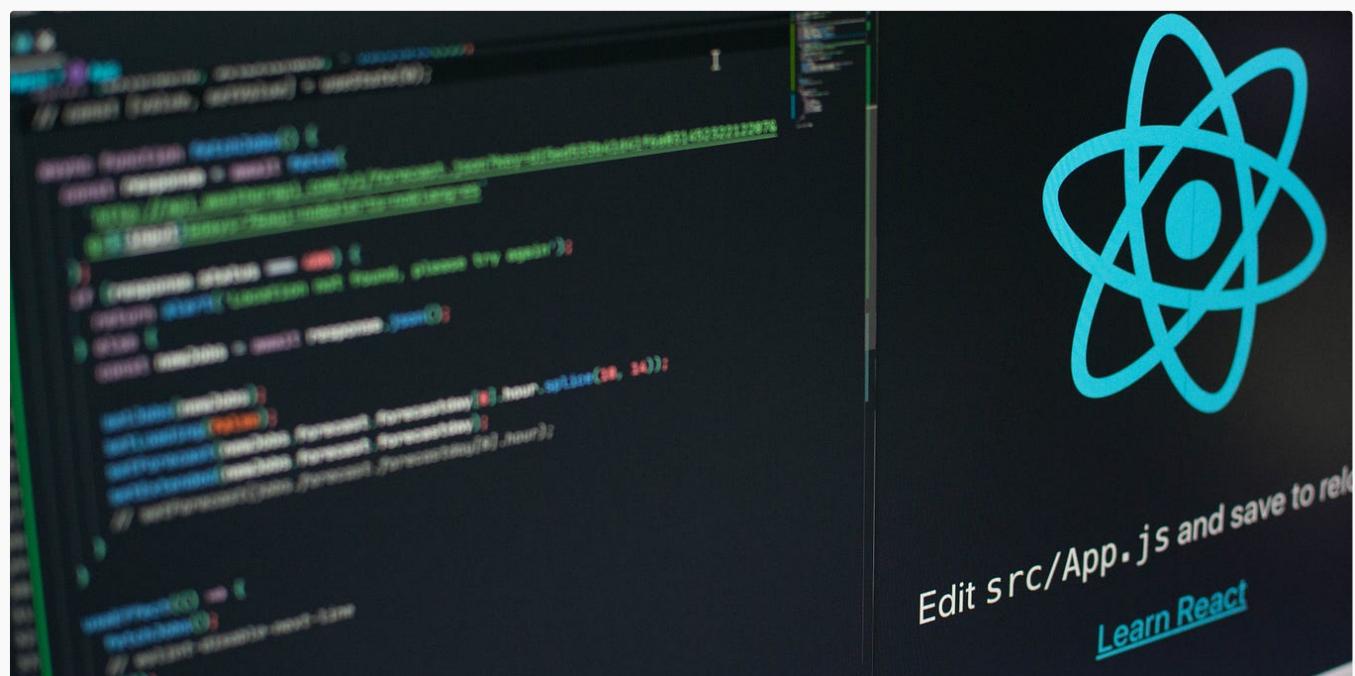
### General Coding Knowledge

20 stories · 427 saves



### Modern Marketing

34 stories · 182 saves



Tomisin Balogun in JavaScript in Plain English

# Creating a Next.js Project with ESLint, Airbnb Style Guide, TypeScript, and Husky

Introduction

3 min read · Aug 11

👏 66

💬 1



...



Cameron Adams

## Setting Up a Next.js 13 Project with ESLint and Prettier

In this tutorial, we will walk through the process of setting up a fresh Next.js 13 project with ESLint and Prettier. We'll also configure...

3 min read · Jul 23

👏 21

💬 2



...

 Bob Junior

## Maximizing TypeScript Project Quality: The Best Configuration Settings for tsconfig.json

tsconfig.json is a configuration file used by TypeScript to specify compiler options and project settings. It's a crucial part of any...

2 min read · Apr 17

 3 

...

```
PS C:\Users\rimaz\source\repos\playground> node -v
v18.13.0
PS C:\Users\rimaz\source\repos\playground> yarn -v
1.22.19
PS C:\Users\rimaz\source\repos\playground> yarn create react-app my-react-app
yarn create v1.22.19
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Installed "create-react-app@5.0.1" with binaries:
- create-react-app

Creating a new React app in C:\Users\rimaz\source\repos\playground\my-react-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

yarn add v1.22.19
info No lockfile found
```

 Rimaz Mohommed

## React 102 : Configuring code formatting using Prettier, ESLint & Husky

In this article I'll be diving into setting up prettier, eslint and husky for configuting the code formatting rules for your react js...

8 min read · Jul 13



...

See more recommendations