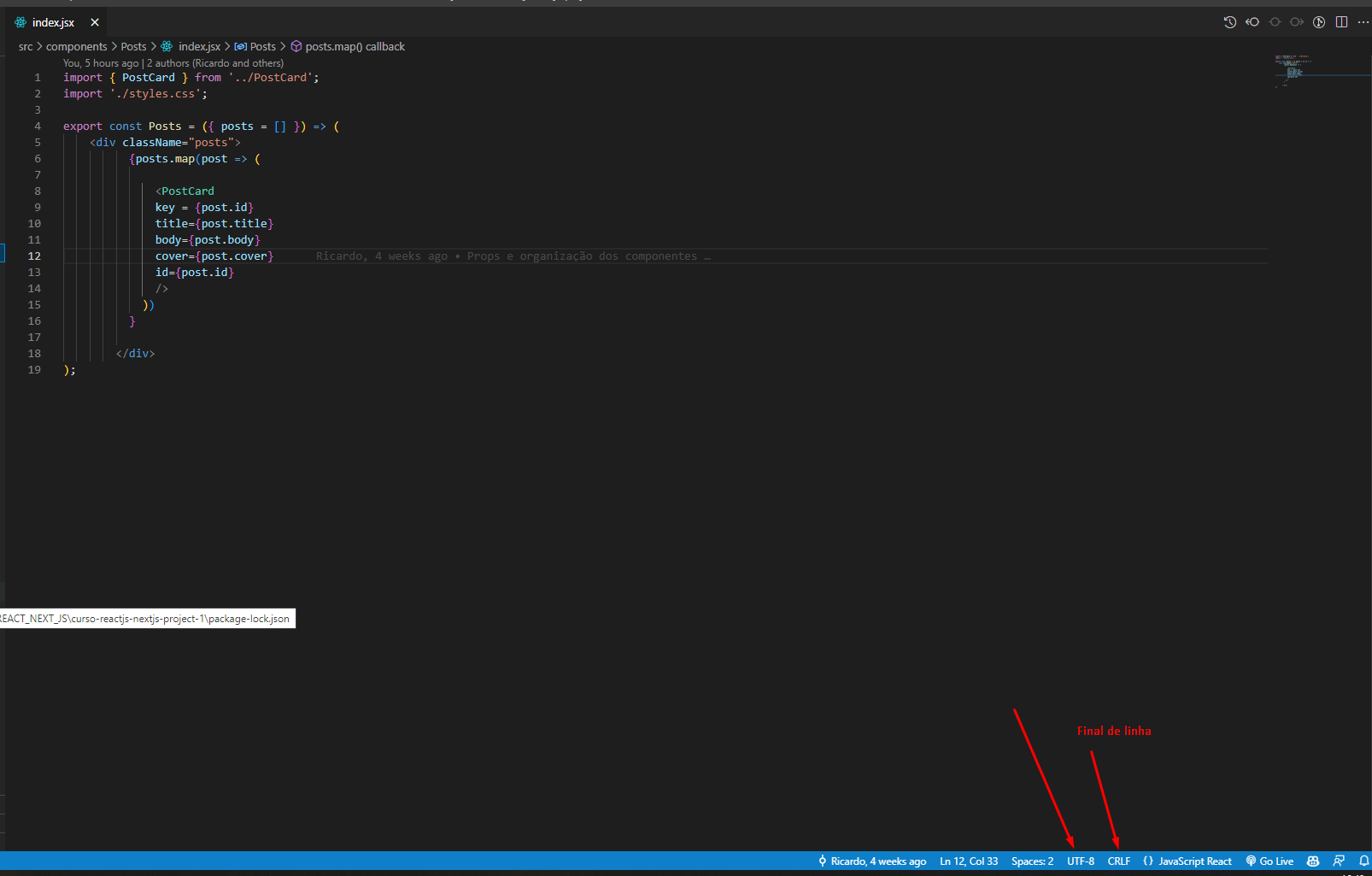
Aula 37. Eslint, prettier, editorconfig e prop-types do curso

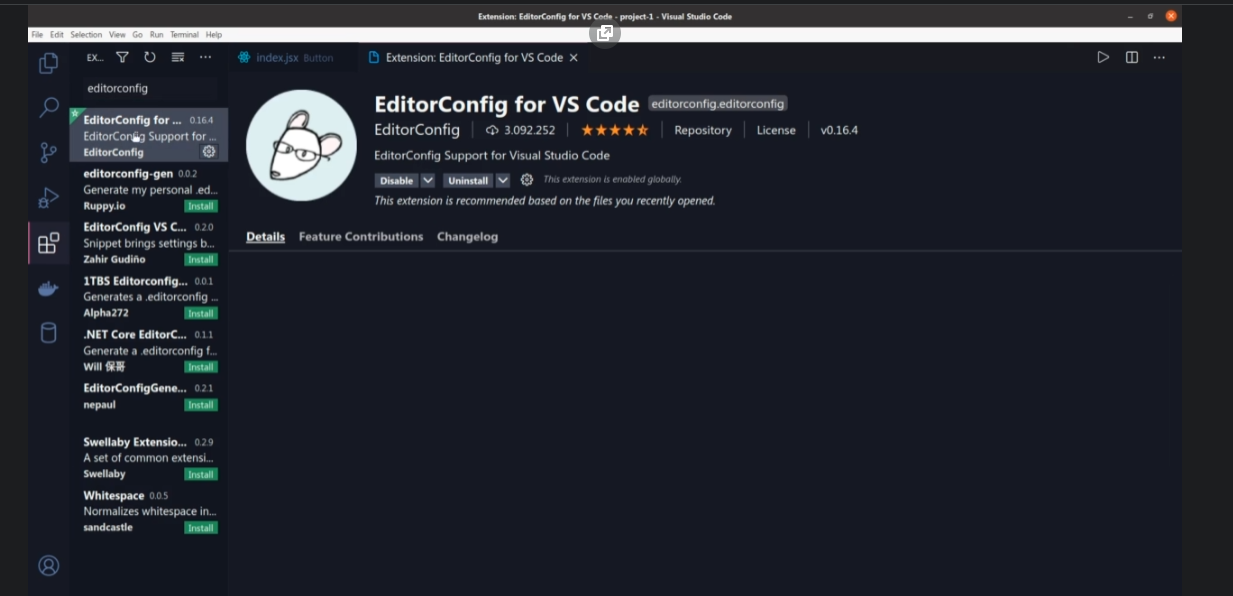
https://www.udemy.com/course/curso-de-reactjs-nextjs-completo-do-basico-ao-avancado/

Eslint, prettier, editorconfig e prop-types

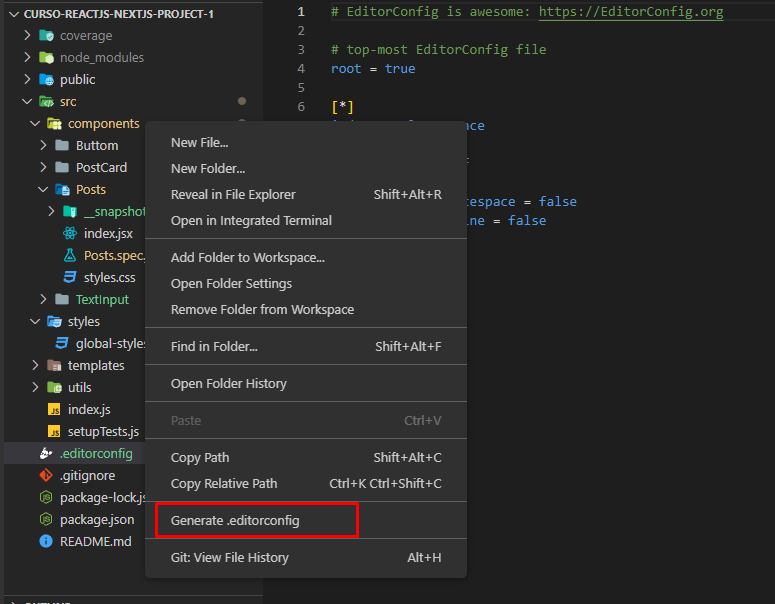
Durante um projeto, algumas coisas devem-se manter consistentes como final de linha, codificação e etc:



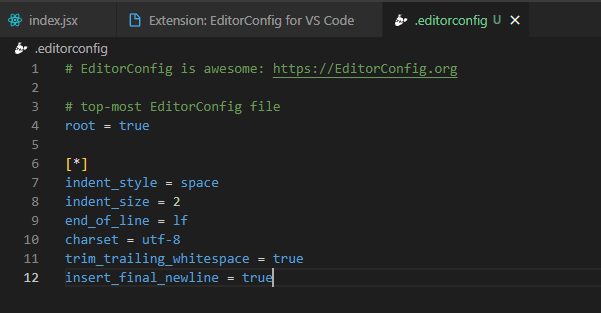
Podemos usar o editorconfig para VSCode:



Depois, reiniciamos o vscode e clicamos com o botão direito numa área vazia no explorer de arquivos e clicamos em generator editconfig:

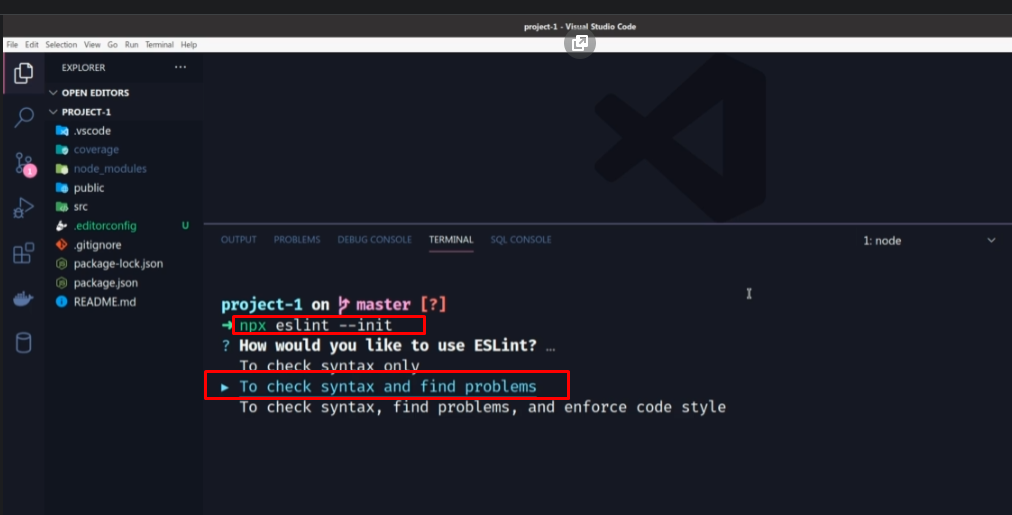


No curso, utilizamos as seguintes configurações:

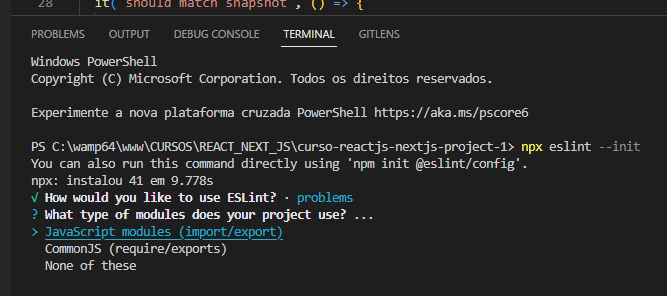


Para padronizar a utilização ou não de ponto e virgulas, utilizamos o Eslint e o prettier.

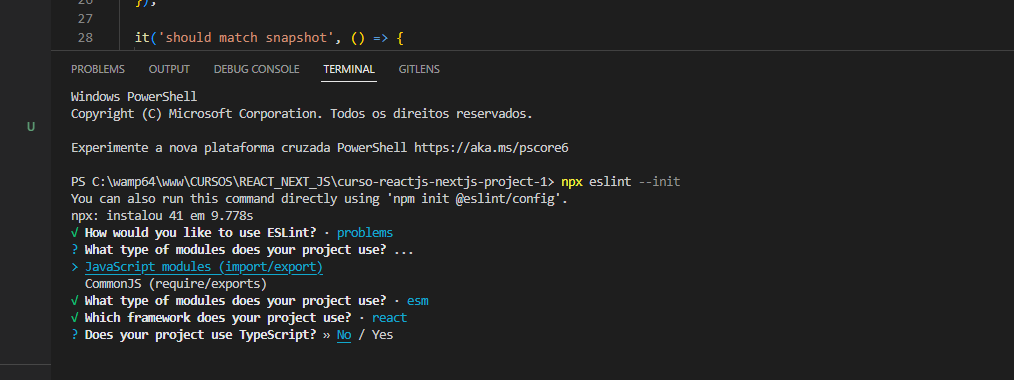
Usamos o comando npx eslint --init



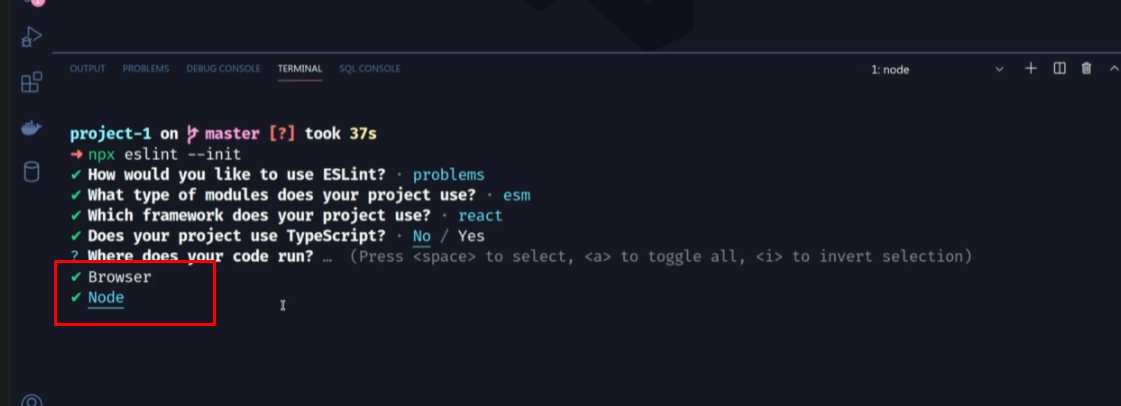
depois:



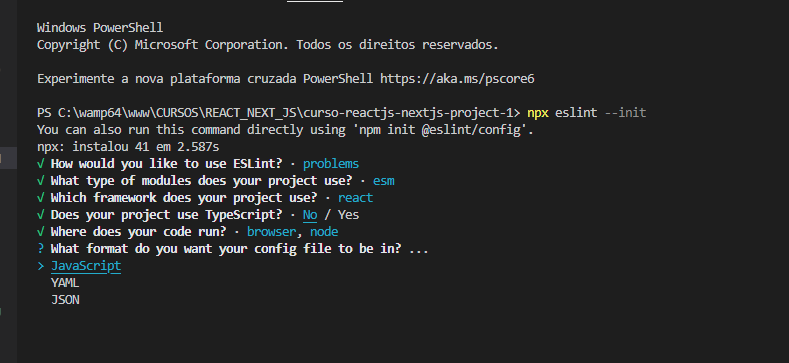
Marcamos que estamos utilizando react e depois que não utilizamos typescript, pois ainda não usaremos.



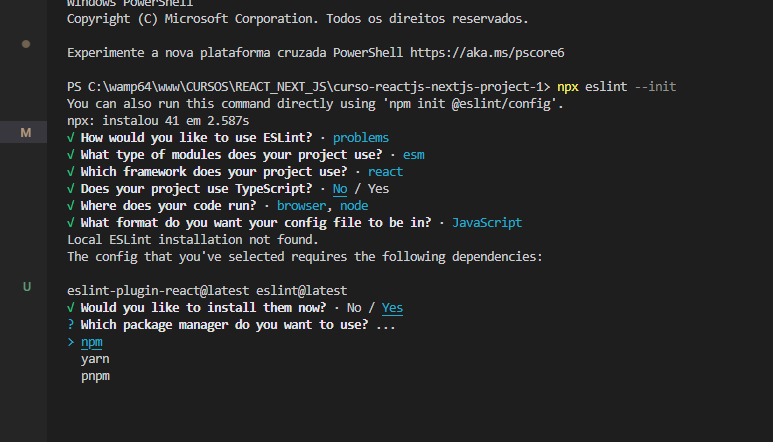
Com a barra de espaço, marcamos os Browser e o node



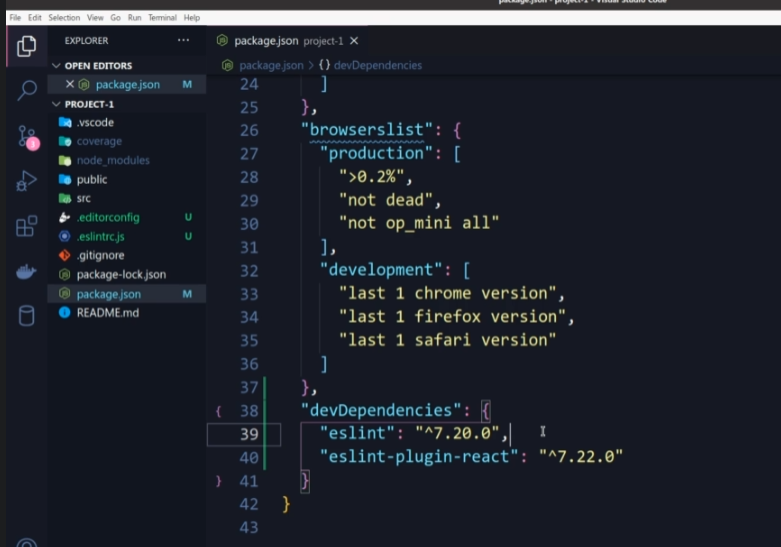
Depois usamos o formato JS para o arquivo de configuração



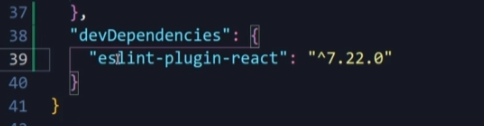
Após, colocamos yes para instalar os plug-ins



Serve para verificar a linguagem que utilizamos e etc daí a partir disto, podemos instalar algum app e etc:



Vamos ter que retirar o plugin do eslint (para evitar conflitos e etc), temos que deletar a linha (39 na img acima)



deletar o package-lock.json e a node modules para depois usar o npm i para reinstalar tudo novamente.

É criado um arquivo chamado .eslintrc.js onde ficam as configurações.

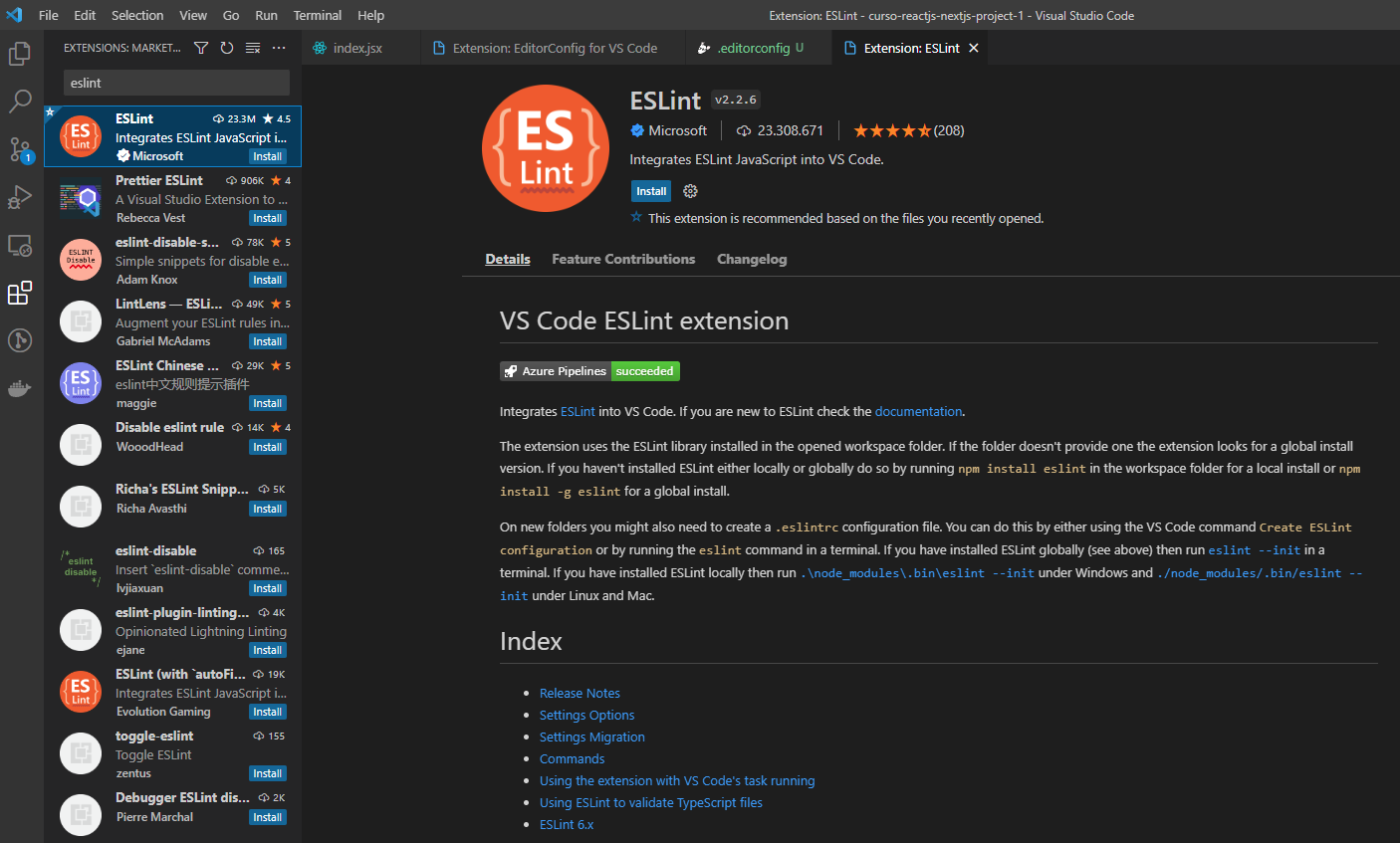
\*\* poderíamos simplesmente criar este arquivo vazio e incluir as configurações.

seguem as utilizadas pelo professor em [*https://github.com/luizomf/curso-reactjs-nextjs-project-1/blob/dafe58e9fca6f61b448af33248d7cb6d2aa12893/.eslintrc.js*](https://github.com/luizomf/curso-reactjs-nextjs-project-1/blob/dafe58e9fca6f61b448af33248d7cb6d2aa12893/.eslintrc.js)

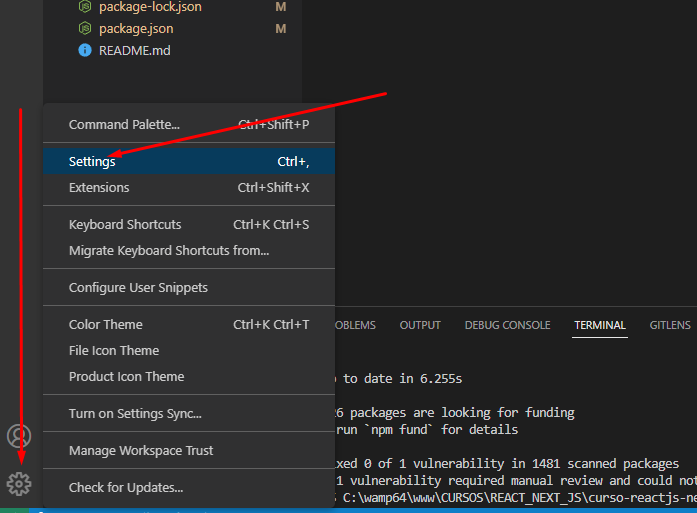
Depois vamos utilizar o seguinte comando para instalar as alterações:

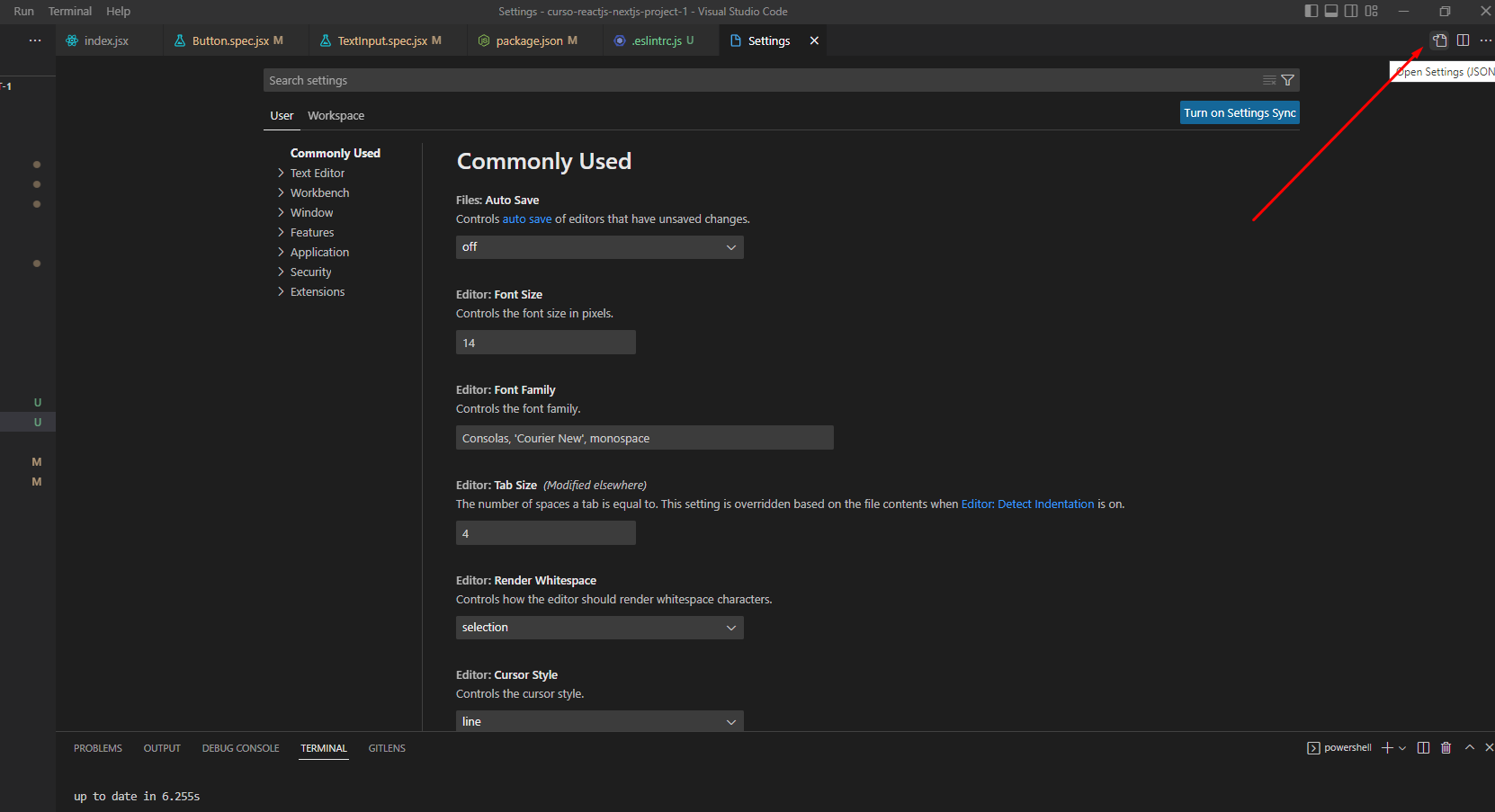
npm install -D prettier eslint-config-prettier eslint-plugin-prettier

Precisamos da extensão eslint também



Após, vamos clicar na engrenagem do vs code e em settings:



depois :

As configurações do professor estão todas em:

<https://gist.github.com/luizomf/88af1fde8d28b103540b74ec05224279>

Usei só alguma coisinha (achei interessante a questão de elimiar algumas pastas da busca como anode-modules) e ficou assim:

{

    // Workbench

    "workbench.startupEditor": "none",

    "workbench.editor.enablePreview": false,

    "workbench.editor.limit.value": 50,

    "workbench.iconTheme": "material-icon-theme",

    "javascript.updateImportsOnFileMove.enabled": "always",

    "editor.inlineSuggest.enabled": true,

      // Editor and breadcrumbs

  "editor.codeActionsOnSave": {

    "source.fixAll.eslint": true,

    "source.fixAll": true

  },

      // Git

  "git.enableSmartCommit": true,

  "diffEditor.ignoreTrimWhitespace": true,

      // Files and search

  "files.defaultLanguage": "{activeEditorLanguage}",

  "files.exclude": {

    "USE\_GITIGNORE": true

  },

  "search.exclude": {

    "\*\*/node\_modules": true,

    "\*\*/bower\_components": true,

    "\*\*/coverage": true,

    "\*\*/dist": true,

    "\*\*/build": true,

    "\*\*/.build": true,

    "\*\*/.gh-pages": true

  },

}

Vamos criar o arquivo de configuiração do prettier (.prettierrc.js)

que vai conter o seguinte:

module.exports = {

  semi: true,

  trailingComma: 'all',

  singleQuote: true,

  printWidth: 120,

  tabWidth: 2

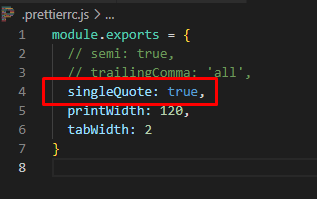
}

O semi obriga a por ponto e virgula em tudo.

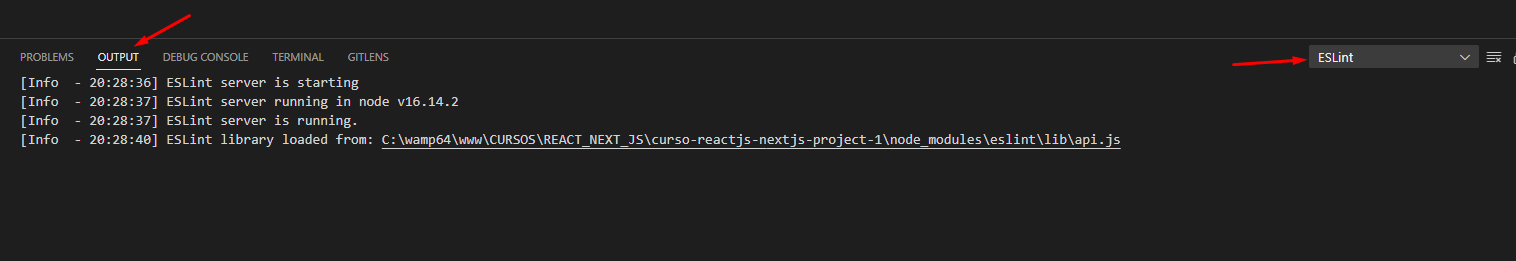
O trailingComma é uma virgula sobrando no final de tudo

Eu tirei praticamente tudo para não limitar o código.

Só esta linha é obrigatória, senão da erro de aspas em todo projeto:

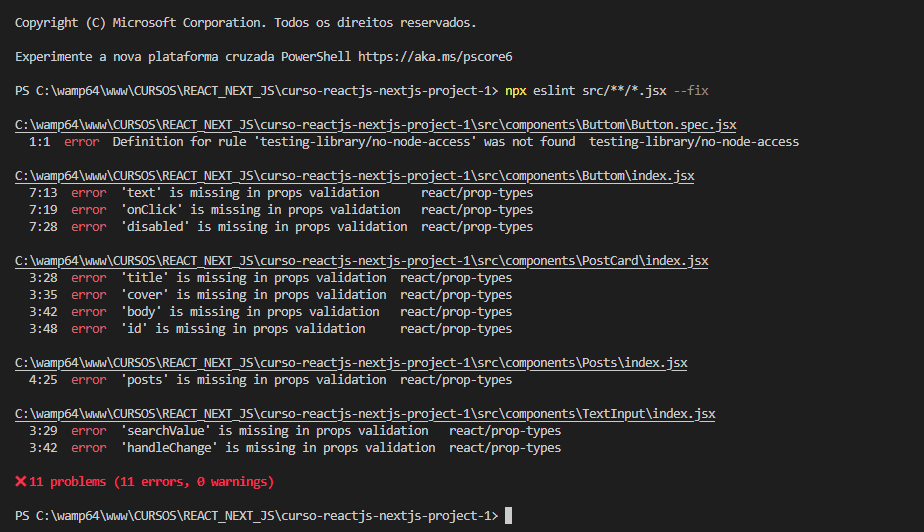


Depois, verificamos se existem erros no eslint, no output do terminal:



\*\* Vamos encontrar diversos erros, então vamos corrigir todos automaticamente com o seguinte comando no terminal: npx eslint src/\*\*/\*\*.jsx –fix que vai dizer onde estão os arquivos a serem corrigidos.

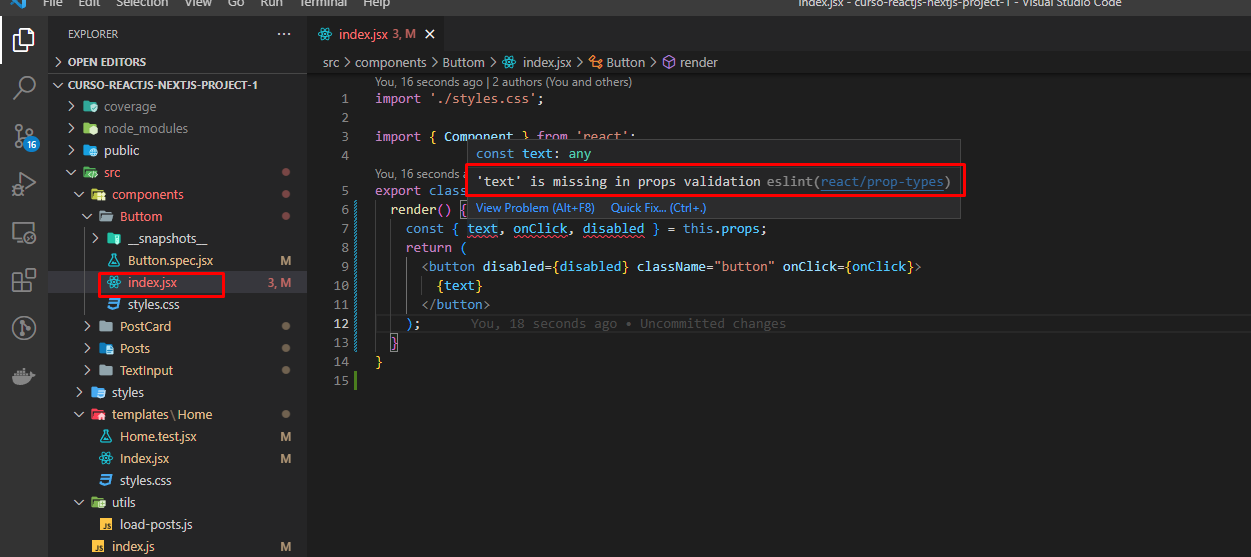
encontrados diversos erros:



Para corrigir os problemas, vamos utilizar os prop-types

no terminal: npm i prop-types

Iniciamos pelo componente do botão



Com os prop-types podemos validar as props como o eslint está exigindo