Nome: Paulo Roberto Xavier da Silva Data: 18/11/2024 Turma: 231N

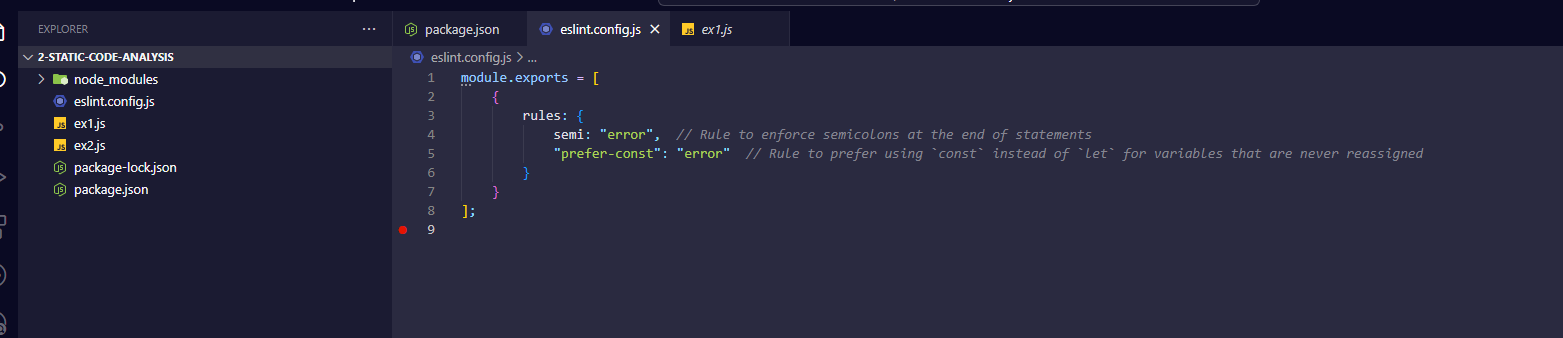
**Relatório - Análise Estática**

Este documento contém as etapas desenvolvidas durante a realização da atividade proposta de Análise Estática (Static Analysis).

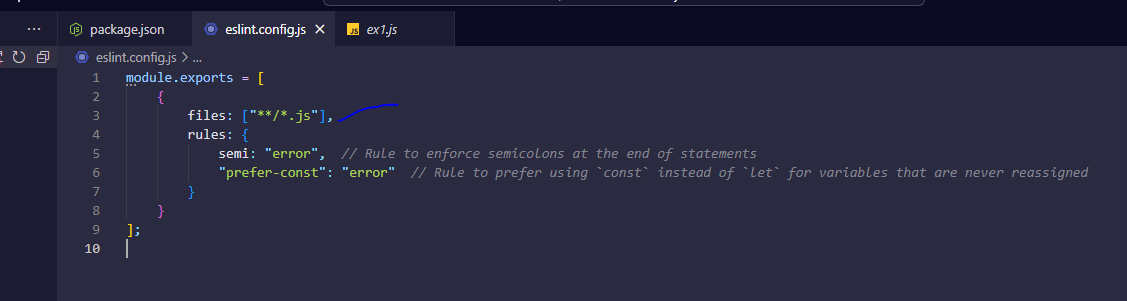
1. Configurar o arquivo “ ***eslint.config.js*** ” para encontrar esses problemas e executar o comando “ ***npm run lint*** ”.

De primeiro momento, acessei a documentação do ESLint, disponível no site “[***https://eslint.org/docs/latest/use/configure/configuration-files***](https://eslint.org/docs/latest/use/configure/configuration-files)”.

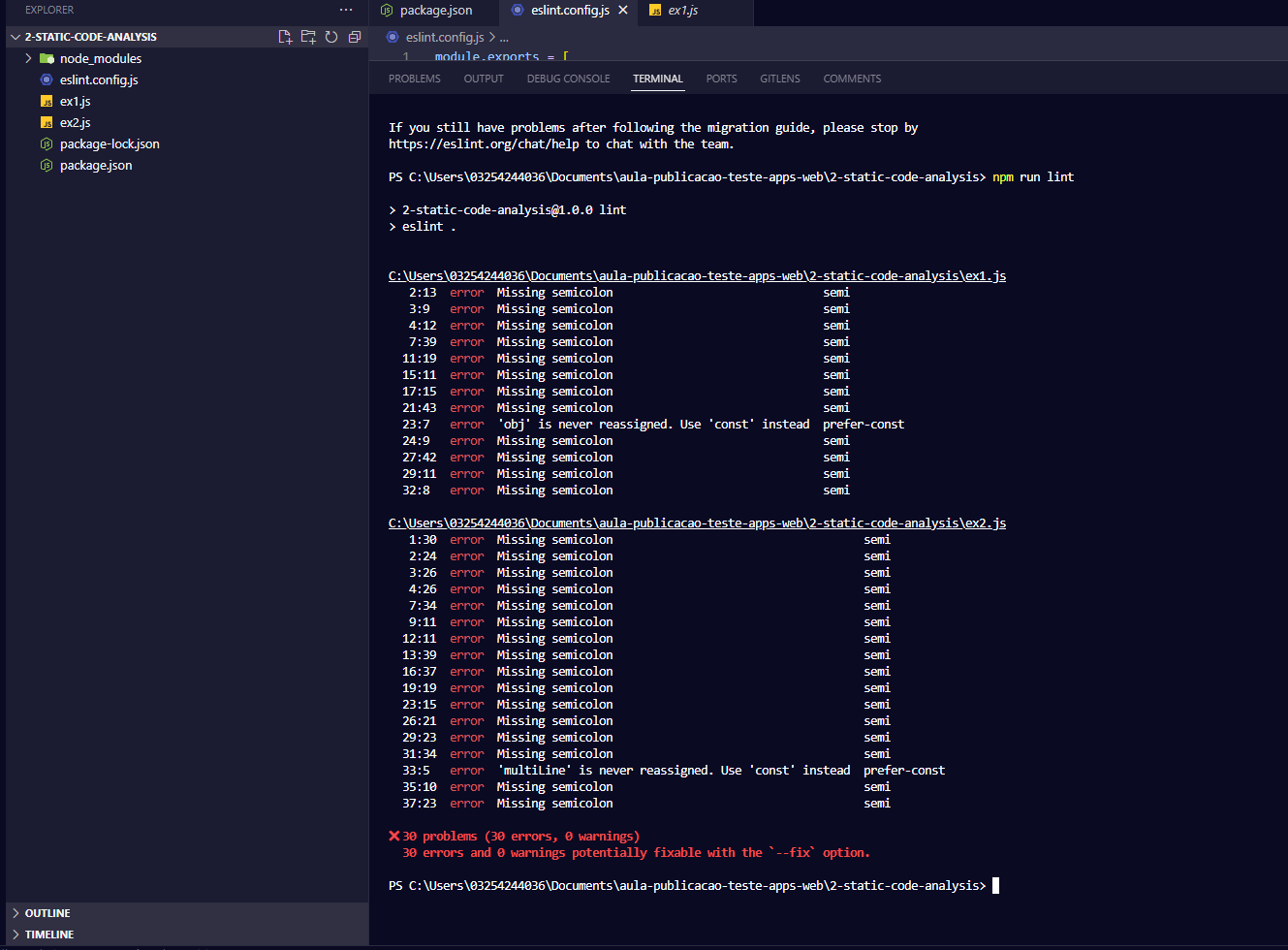
Criei o arquivo “ **eslint.config.js** ” e inseri o seguinte trecho:



Também inseri a especificação das extensões dos arquivos que o **ESLint** irá validar:

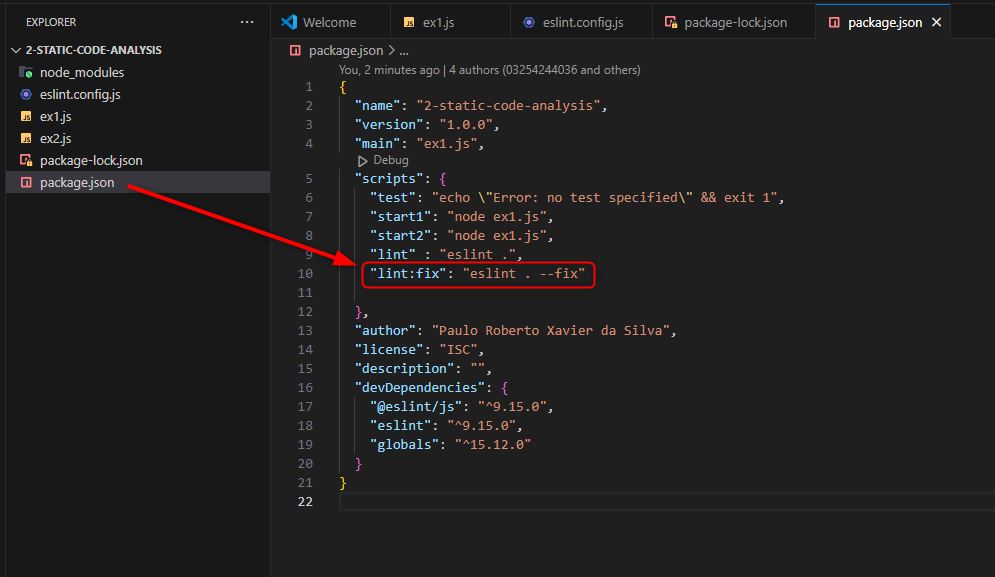


Após a inserção dessas configurações, executei pela primeira vez o comando “ ***npm run lint***  “ e obtive o seguinte resultado:

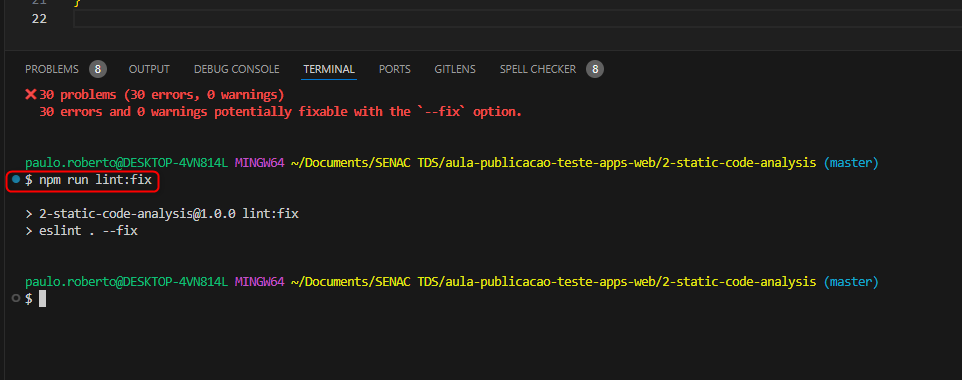


1. Criar script “ **lint:fix** ” no package.json para corrigir automaticamente

Para gerarmos esse script de correção, através da leitura na documentação, identifiquei que basta inserir a seguinte linha no script no arquivo “ **package.json** “:

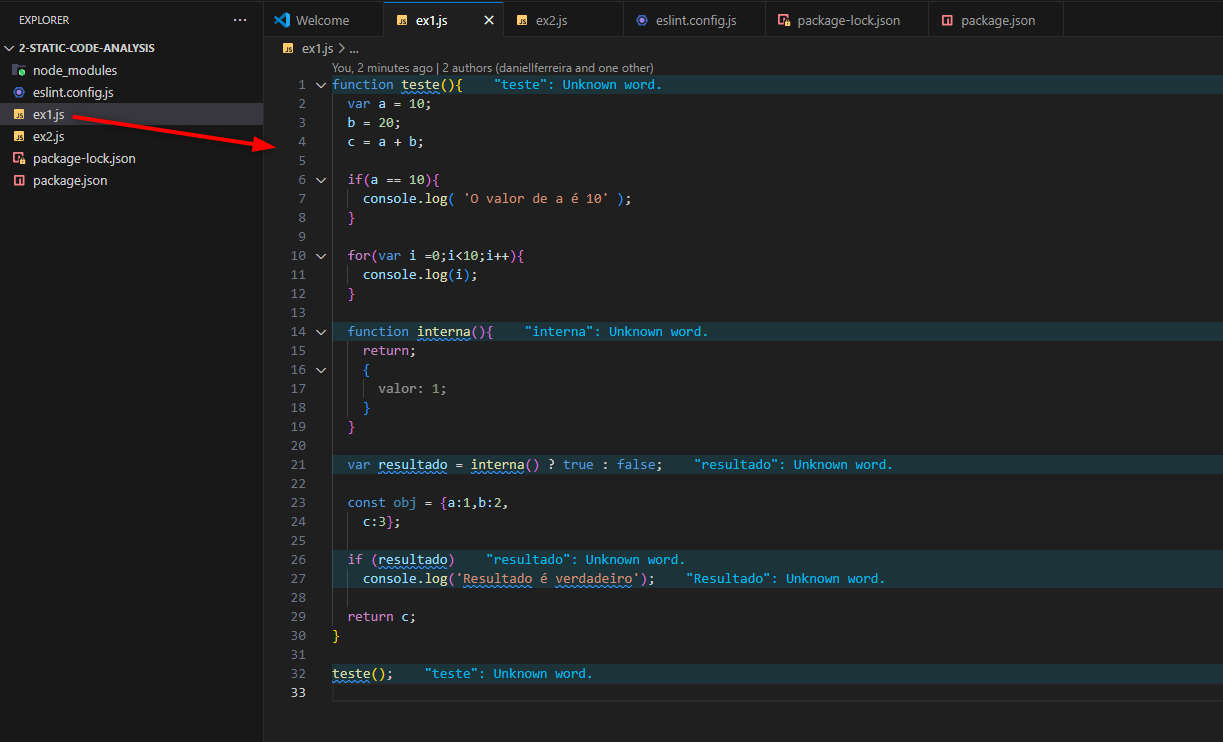


Feito esse ajuste, abri o terminal no VSCode e apliquei o comando “ **npm run lint:fix** “

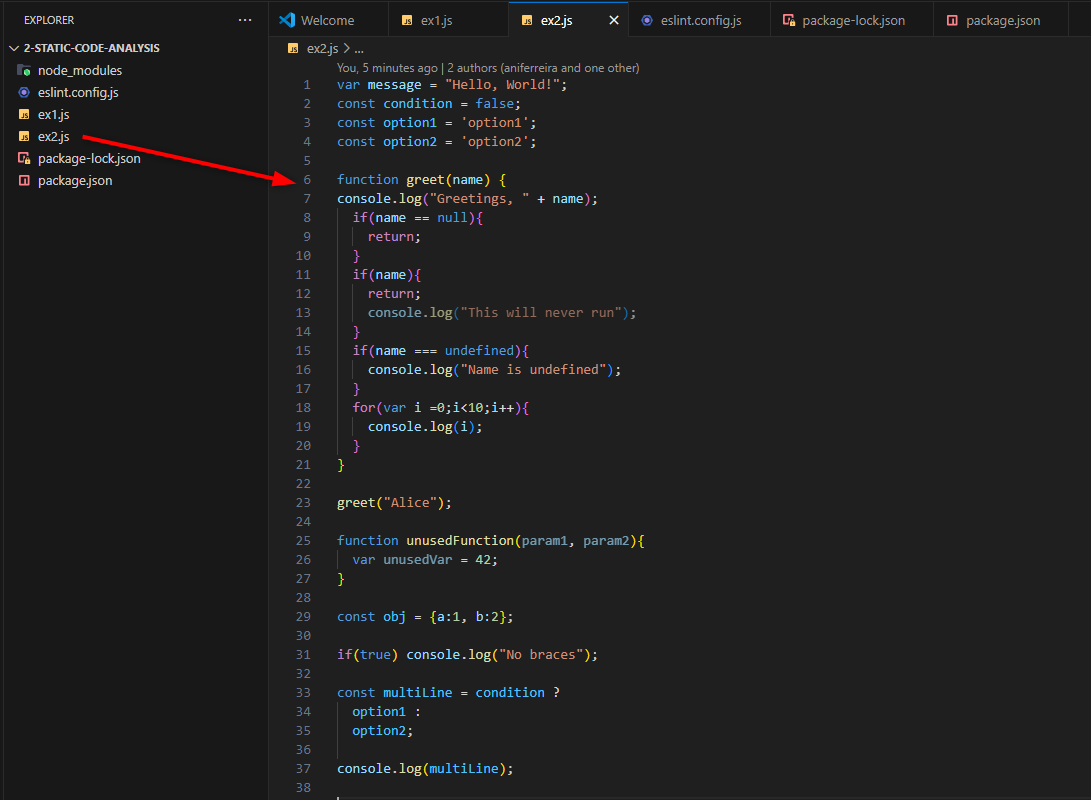


Após executar o comando, observei que os arquivos “ **ex1.js** ” e “ **ex2.js** ” foram modificados, conforme as regras que eu inseri no arquivo “ **eslint.config.js** “.

**ex1.js**

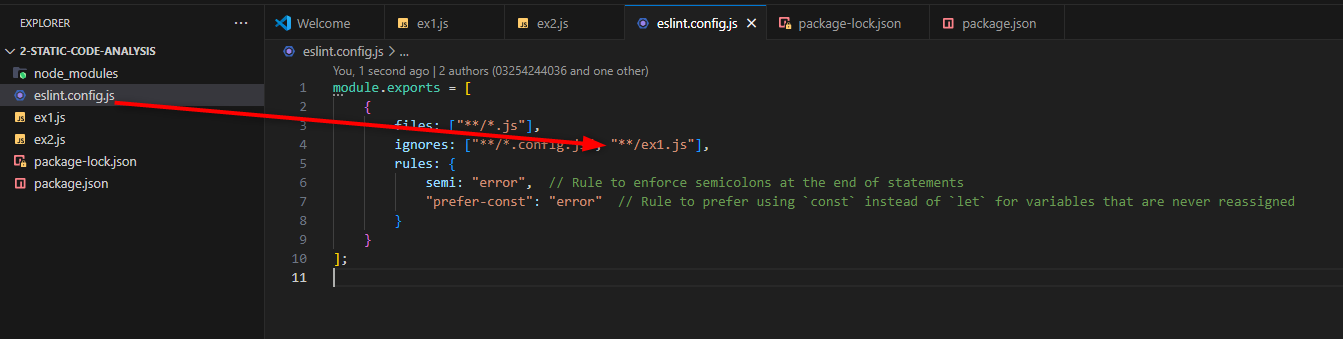


**ex2.js**

****

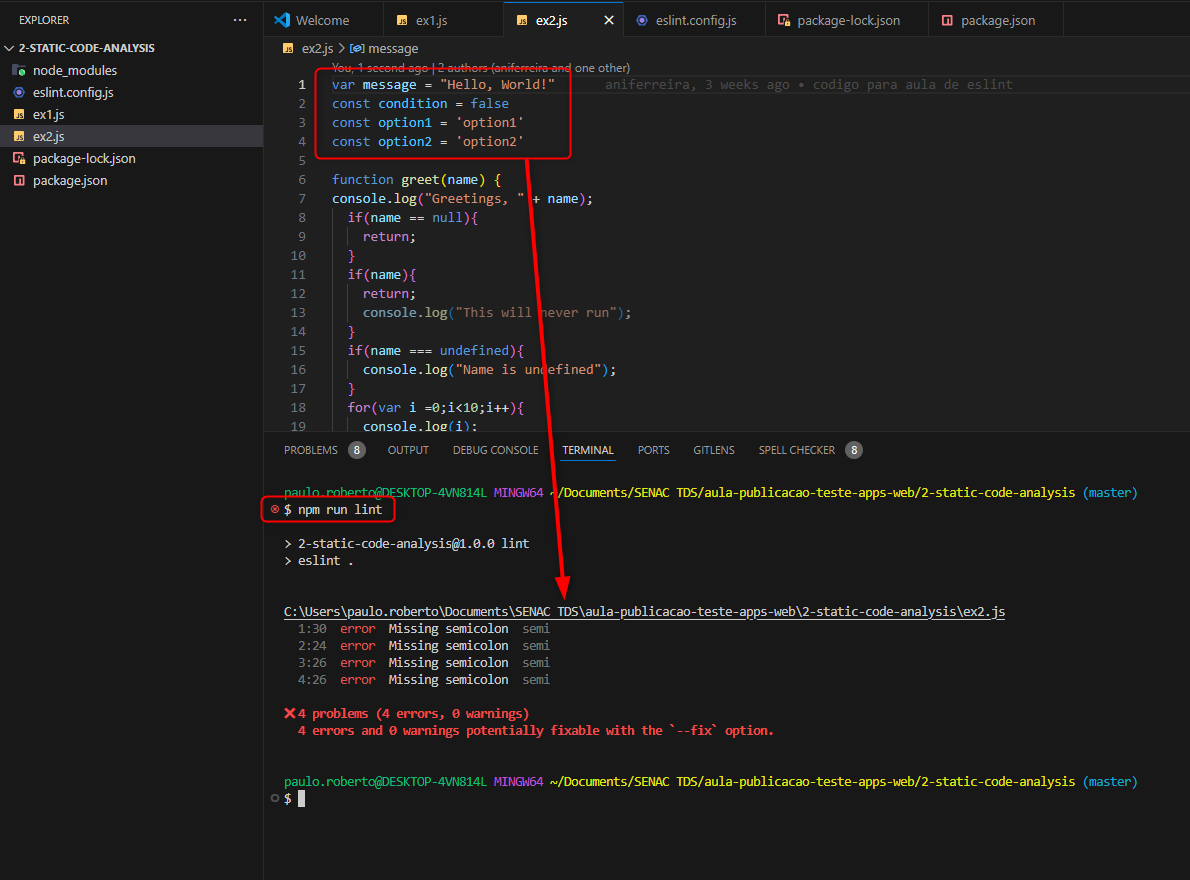
1. Desabilitar o linter um arquivo inteiro de cada vez para ter os resultados por arquivo.

Para ignorarmos determinados arquivos durante a execução do linter, basta inserirmos a seguinte linha no arquivo “ **eslint.config.js** “, juntamente com o arquivo que desejamos ser ignorado:

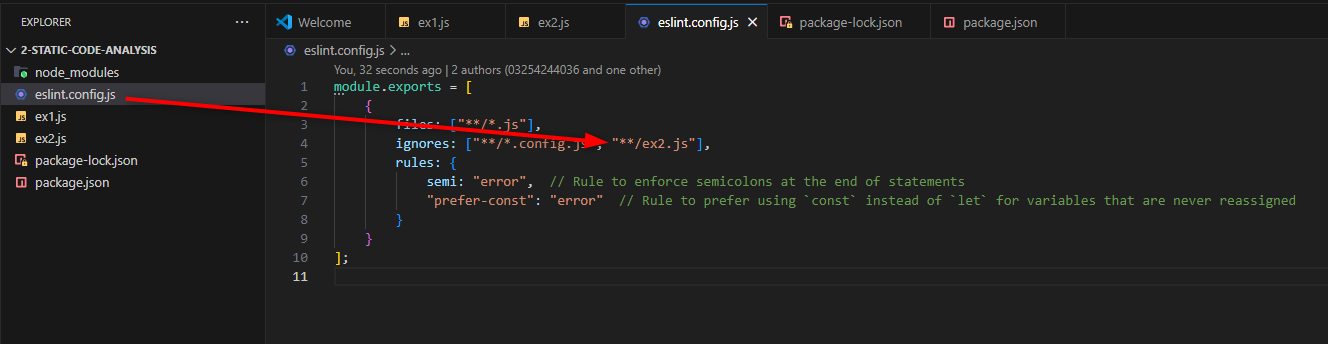


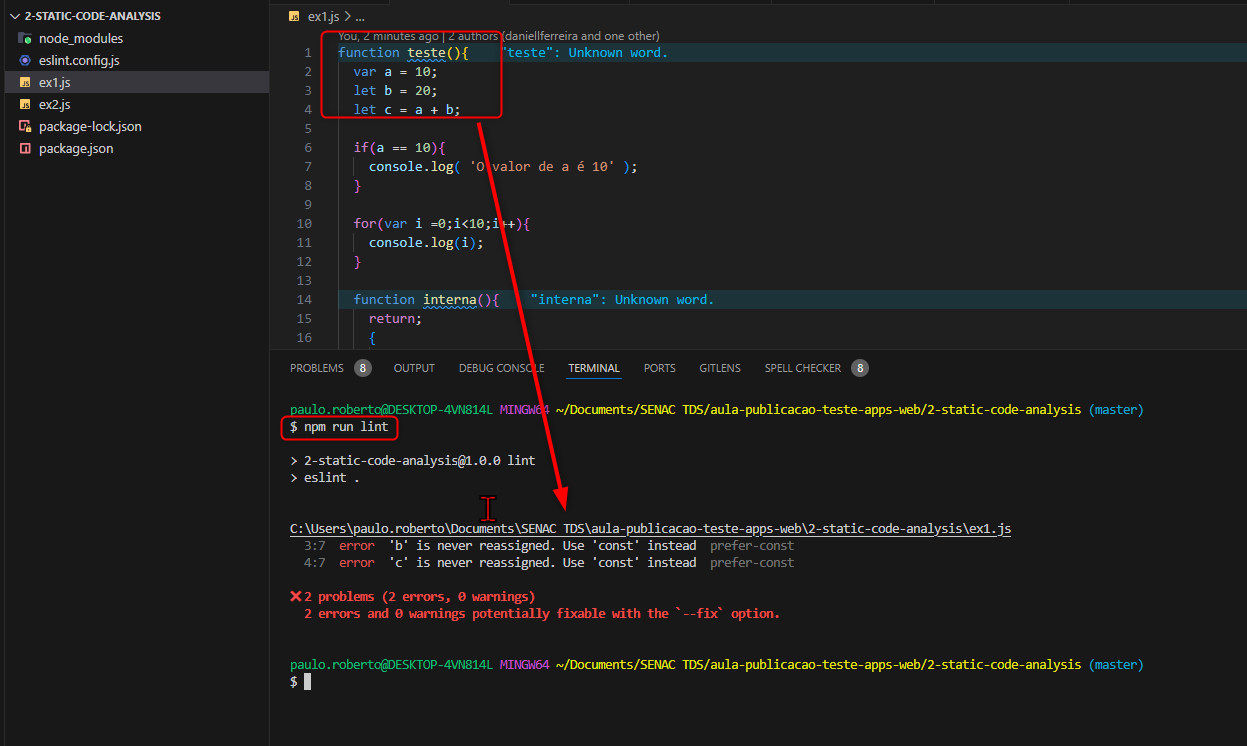
Para fazer o teste, visto que com o enunciado anterior já foram corrigidos os erros, acessei o arquivo “ **ex2.js** ” e removi removi alguns pontos e vírgulas para que sejam exibidos erros.

Abri o terminal e novamente inseri o comando “ **npm run lint** “.



Agora, refiz o mesmo procedimento, porém para que seja ignorado o arquivo “ **ex2.js** “:

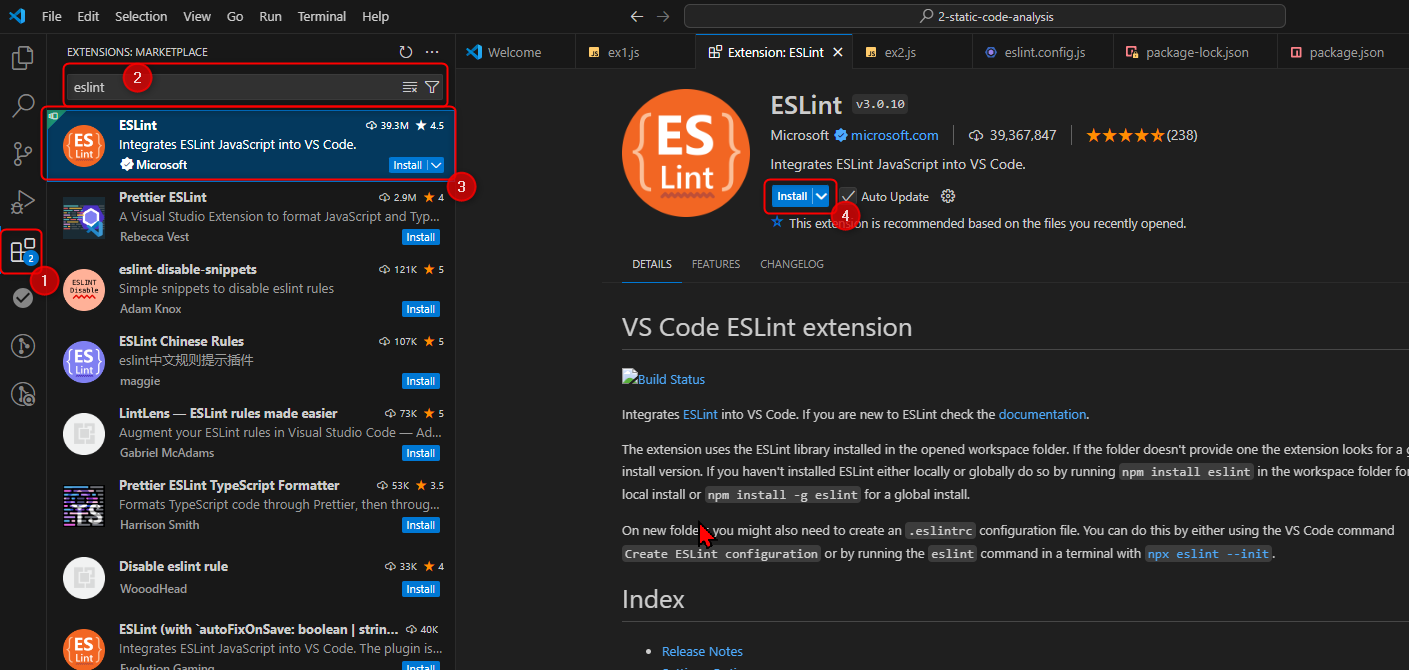




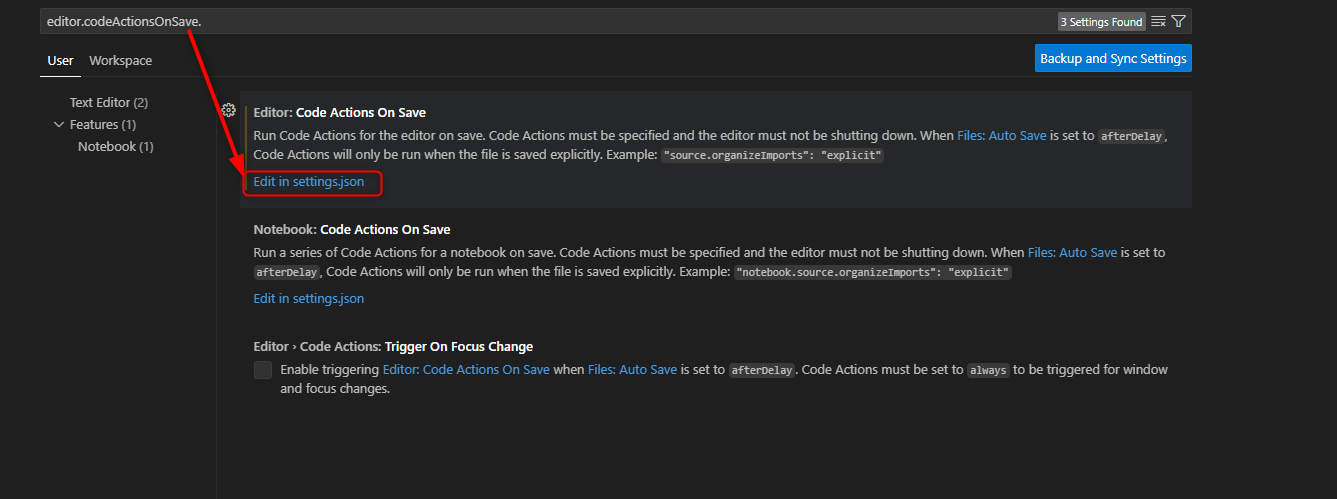
1. Configurar o editor para corrigir automaticamente ao salvar

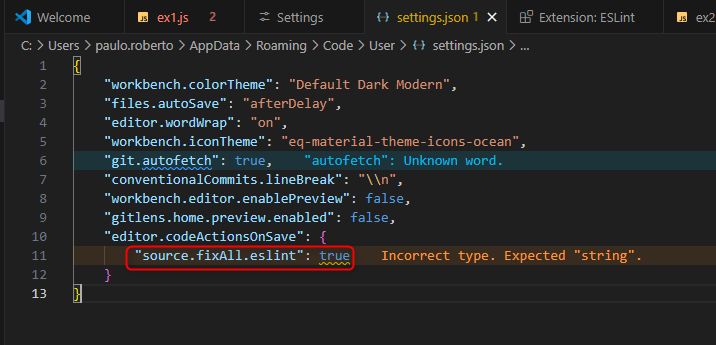
Para fazer essas configurações, apliquei a seguinte sequência:

* Abri o ícone de “**Extensões**” do VSCode e instalei a extensão oficial da Microsoft chamada “**ESLint**”;

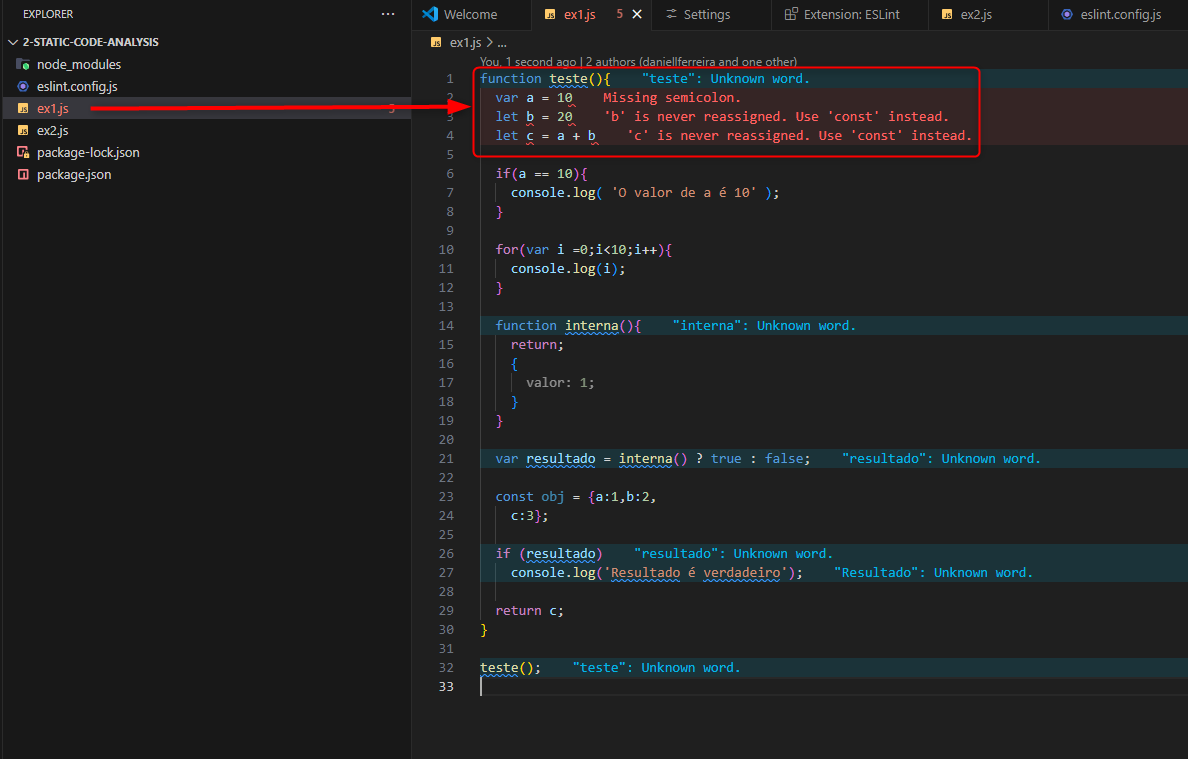


* Posteriormente, acessei as configurações do VSCode através do atalho “**Ctrl+,**”
* Pesquisei por “ **editor.codeActionsOnSave** “ e selecionei a opção “ **Edit in settings.json** “

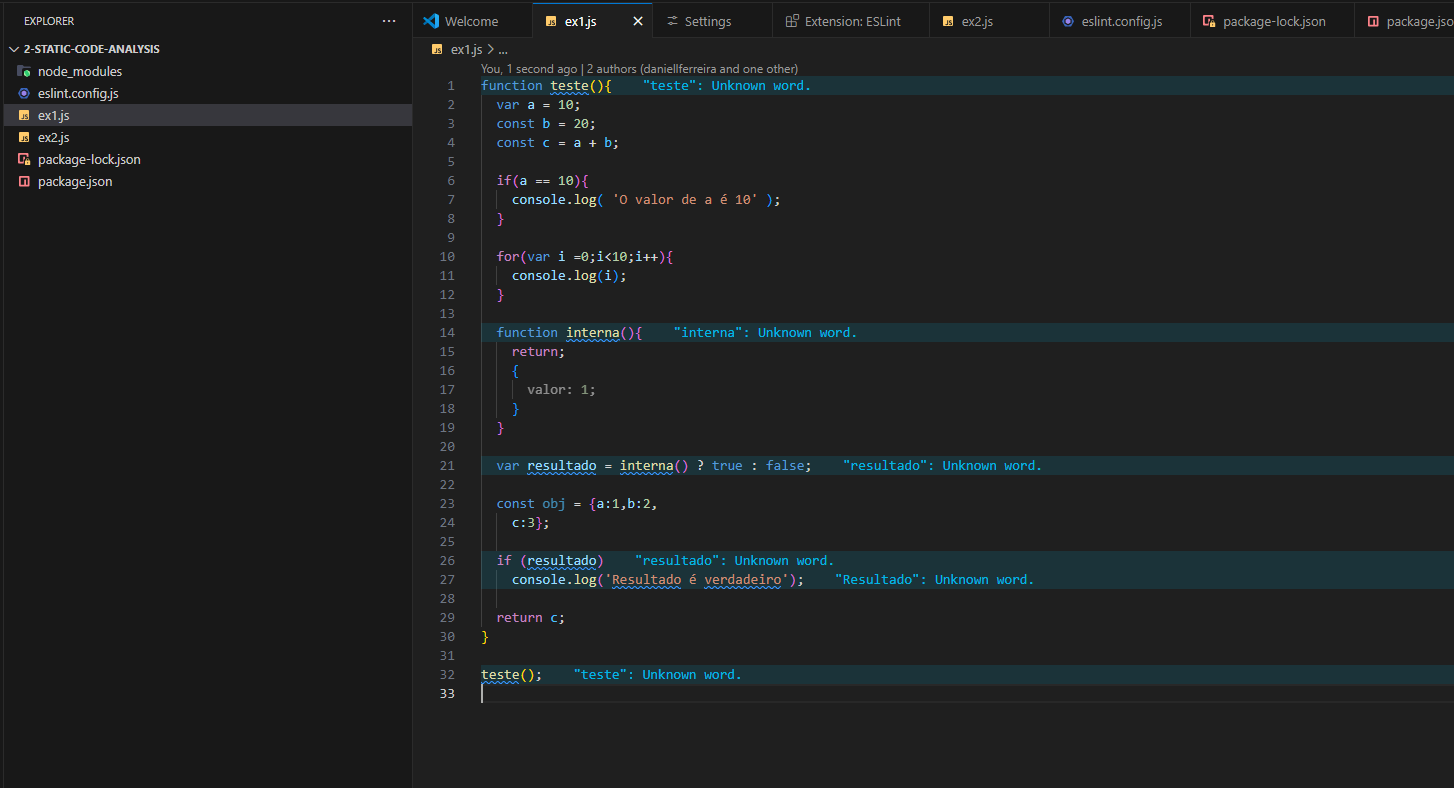


* Por fim, inseri a seguinte linha: **“source.fixAll.eslint” : true**
* 

Para fazer o teste e validar, acessei o arquivo “ **ex1.js** “, que estava constando alguns erros, conforme a imagem:

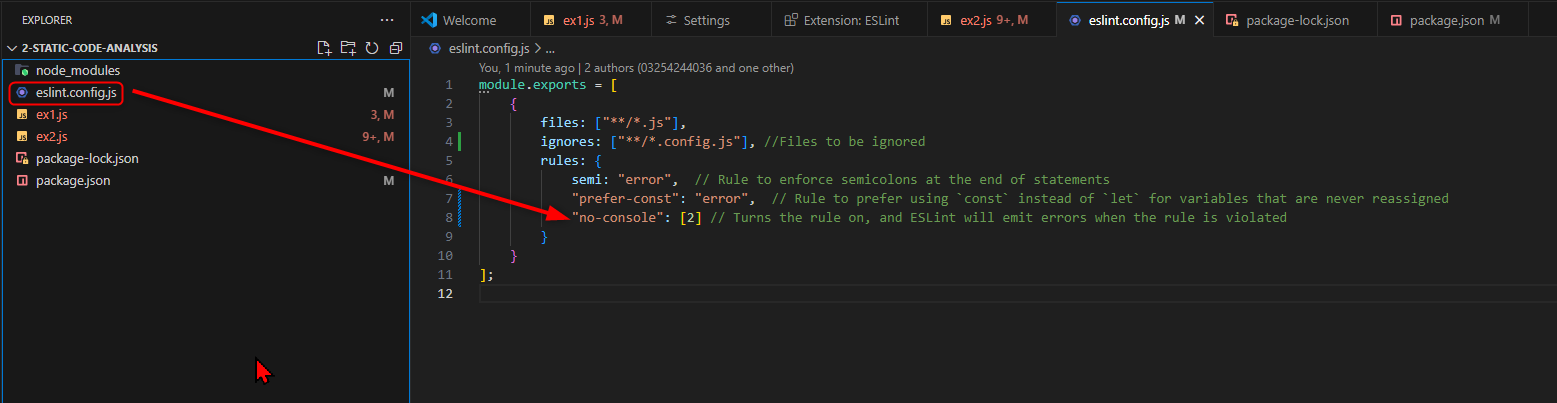


Pressionei “ **Ctrl + s** “ e automaticamente foi corrigido o código:

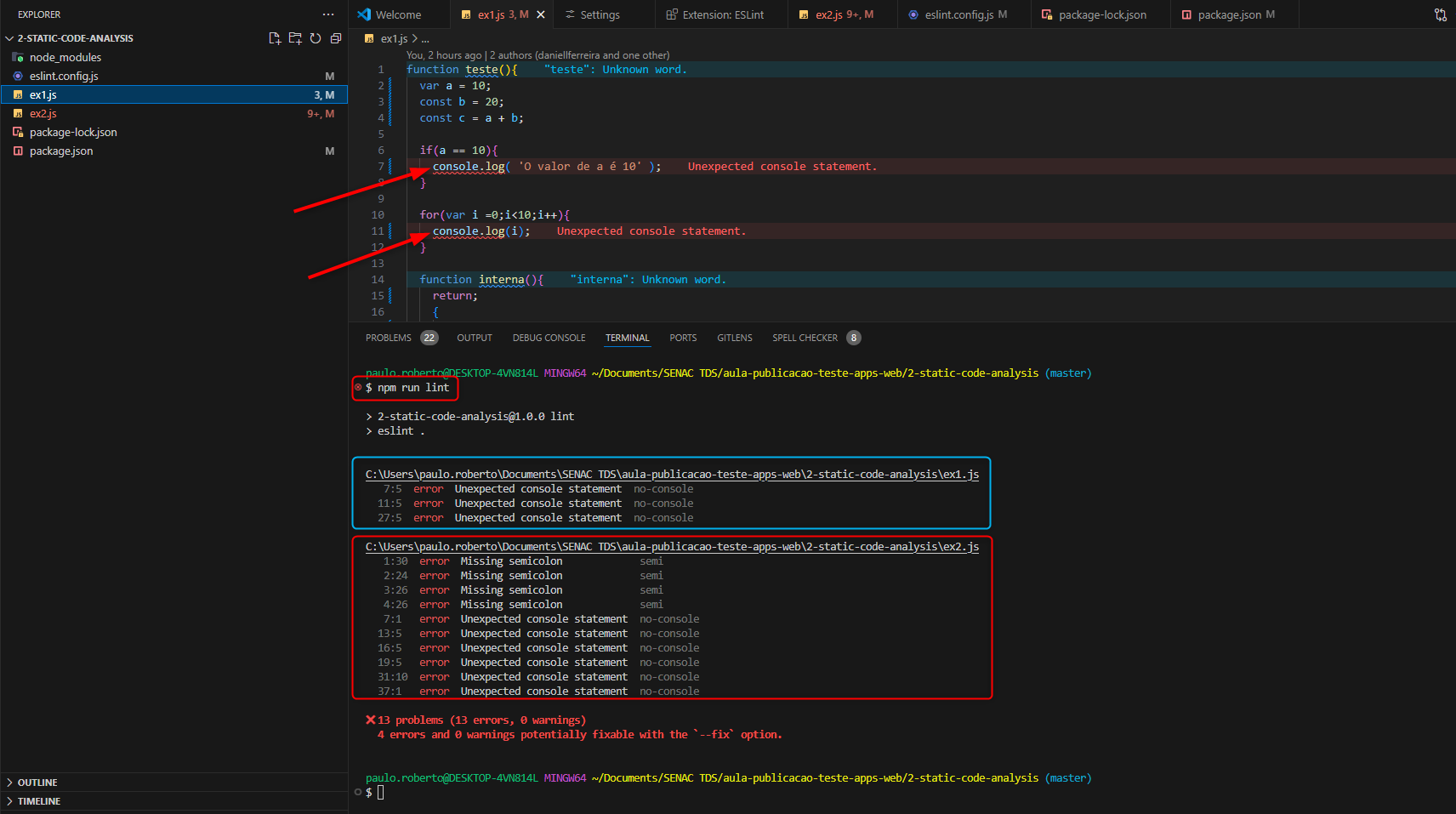


1. Desabilitar por linha os comandos “ **console.log** “

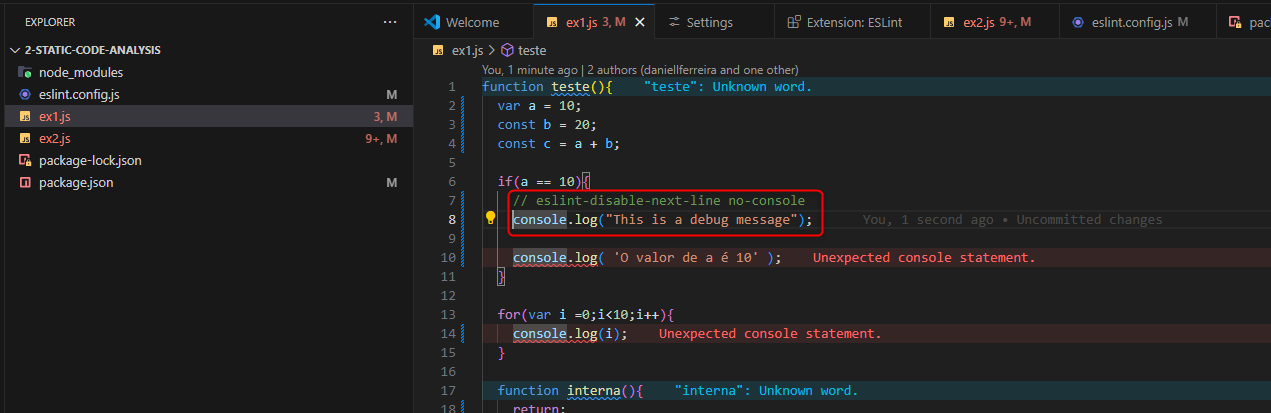
Para poder desabilitar os comandos de console, de primeiro momento, criei uma regra no arquivo “ **eslint.config.js** ” que bloqueie e emita erros ao utilizar o comando “ **console.log** “



Após aplicada essa regra, executei novamente o comando “ **npm run lint** “ e obtive os seguintes erros, incluindo os erros de “ **console** ”:



Para fazer com que o ESLint ignore determinada linha contendo console, inseri o seguinte comentário antes da linha contendo o console: “ **// eslint-disable-next-line no-console** “



Isso fez com que a mensagem de erro não fosse exibida, mesmo havendo uma regra aplicada dentro do arquivo “ **eslint.config.js** “.

Também é possível configurar essa regra por bloco, conforme a imagem:

