

Esclarecimentos e anotações:

*ESLint:

ESLint é um projeto open source que ajuda a encontrar e corrigir problemas no código JavaScript e JSX. Não importa se está escrevendo código no navegador ou no servidor, com ou sem framework, o ESLint pode ajudar no código a ser mais consistente e robusto.

Analisa estaticamente o código para encontrar problemas rapidamente. É implementado na maioria dos editores de texto e pode usá-lo como parte do processo de integração contínua.

Vários problemas que o ESLint encontra podem ser automaticamente corrigidos. As correções do ESLint são conscientes da sintaxe então não terá problemas típicos introduzidos pelos tradicionais algoritmos de encontrar e substituir.

É possível pré-processar o código, usando parseadores customizados e escrevendo suas próprias regras que funcionem junto às regras integradas do ESLint.

*Rules:

<https://eslint.org/docs/latest/rules/>

*Relatórios:

#1 Relatório: Entendendo a estruturação de ESLint.config.js e definição do ambiente & versão.

```
module.exports = {  
  env: {  
    es2024: true, // Define o ambiente como ECMAScript 2024  
    node: true, // Indica que o código será executado no ambiente Node.js  
  },  
  parserOptions: {  
    ecmaVersion: 2024, // Define a versão do ECMAScript como 2024  
    sourceType: 'module', // Configura o código como módulos ES6  
  },  
  rules: {  
    // ..  
  },  
};
```

Anotações & Observações:

- "env" (abreviação de "environment") define os ambientes nos quais seu código será executado. Ajuda o ESLint a entender quais variáveis globais estão disponíveis e a ajustar as regras de acordo.
- "es2024: true": Define que o código está utilizando as funcionalidades mais recentes do ECMAScript 2024. "Obs: ECMAScript é o nome oficial da linguagem".
- node: true: Indica que o código será executado em um ambiente Node.js
- "parserOptions" ajusta como o ESLint interpreta o código. Ela define a versão do ECMAScript que está utilizando e como o código deve ser tratado.
- "ecmaVersion: 2024": Define a versão do ECMAScript como 2024, permitindo o uso das funcionalidades mais recentes do JavaScript.
- "sourceType: 'module'": Configura o tipo de código como módulos. Isso significa que o ESLint entende que o código usa as palavras-chave import e export para modularidade.

#2 Relatório: Entendendo um pouco mais sobre suas diferentes rules

```
module.exports = {  
  env: {  
    es2024: true, // define o ambiente como ECMAScript 2024  
    node: true, // indica que o código será executado no ambiente Node.js  
  },  
  parserOptions: {  
    ecmaVersion: 2024, // define a versão do ECMAScript como 2024  
    sourceType: 'module', // configura o código como módulos ES6  
  },  
  rules: {  
    // indentação  
    indent: ['error', 2], // Força a indentação de 2 espaços para manter a consistência no estilo  
    // de código.  
  
    // estilo de Quebra de Linha  
    'linebreak-style': ['error', 'unix'], // Garante quebras de linha no estilo Unix (LF), úteis para  
    // compatibilidade entre diferentes sistemas operacionais.  
  
    // aspas  
    quotes: ['error', 'double'], // Enforce the use of double quotes for strings for consistency.  
  
    // ponto e vírgula  
    semi: ['error', 'always'], // exige o uso de ponto e vírgula no final das instruções, evitando  
    // erros de sintaxe.  
  
    // console Logs
```

'no-console': 'warn', // emite um aviso ao usar console.log para evitar a presença de logs em produção.

// uso Antes da Definição

'no-use-before-define': 'error', // gera erro se variáveis forem usadas antes de serem declaradas.

// variáveis não utilizadas

'no-unused-vars': ['error', {

 'vars': 'all', // verifica todas as variáveis declaradas.

 'args': 'after-used', // verifica parâmetros de função após os utilizados.

 'ignoreRestSiblings': false // não ignora variáveis não utilizadas em declarações com rest properties.

}],

// Var ao invés de Let/Const

'no-var': 'error', // proíbe o uso de var, encorajando let e const para declarar variáveis.

// espaços ao Redor dos Operadores

'space-infix-ops': 'error',

// número Máximo de Linhas

'max-lines': ['error', {

 max: 300, // limita o número máximo de linhas em um arquivo a 300.

 skipBlankLines: true, // ignora linhas em branco na contagem.

 skipComments: true // ignora comentários na contagem.

}]

},

};

#3 Relatório: Colocando o código em prática. Remoção do Env & ParserOptions por não serem necessários no projeto. Implementando a opção de revisar os arquivos separadamente

module.exports = [

{

 // Configuração de regras específicas para o arquivo ex1.js

 files: ['ex1.js'],

 rules: {

 indent: ['error', 2],

 'linebreak-style': ['error', 'unix'],

 quotes: ['error', 'double'],

 semi: ['error', 'always'],

 'no-console': 'warn',

 'no-use-before-define': 'error',

 'no-unused-vars': ['error', {

 'vars': 'all',

```

    'args': 'after-used',
    'ignoreRestSiblings': false
  ]],
  'no-var': 'error',
  'space-infix-ops': 'error',
  'max-lines': ['error', {
    max: 300,
    skipBlankLines: true,
    skipComments: true
  }]
}
},
{
  // Configuração de regras específicas para o arquivo ex2.js
  files: ['ex2.js'],
  rules: {
    indent: ['error', 2],
    'linebreak-style': ['error', 'unix'],
    quotes: ['error', 'double'],
    semi: ['error', 'always'],
    'no-console': 'warn',
    'no-use-before-define': 'error',
    'no-unused-vars': ['error', {
      'vars': 'all',
      'args': 'after-used',
      'ignoreRestSiblings': false
    }],
    'no-var': 'error',
    'space-infix-ops': 'error',
    'max-lines': ['error', {
      max: 300,
      skipBlankLines: true,
      skipComments: true
    }]
  }
}
];

```

package.json: Adaptando “scripts” para que seja possível usar o lint especificamente em determinados arquivos, rodando através de “npm run lint:ex1” & “npm run lint:ex2”

```

...
"start2": "node ex2.js",
"lint": "eslint .",
"lint:ex1": "eslint ex1.js",
"lint:ex2": "eslint ex2.js"
},
...

```

Resultados:

ex1.js:

The screenshot shows the VS Code editor interface with a file named `ex1.js` open. The editor displays a JSON object with the following content:

```
12  },
13  "author": "",
14  "license": "ISC",
15  "description": "",
16  "devDependencies": {
17    "@eslint/js": "^9.15.0",
18    "eslint": "^9.15.0",
19    "globals": "^15.12.0"
20  }
```

The bottom panel of the editor shows the **PROBLEMS** view with the following errors:

- 28:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
- 29:11 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
- 29:11 error Missing semicolon semi
- 30:2 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
- 31:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
- 32:8 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
- 32:8 error Missing semicolon semi

A summary message states: **55 problems (52 errors, 3 warnings)**. 51 errors and 0 warnings potentially fixable with the `--fix` option.

The terminal shows the command prompt: `PS C:\Users\notch0 real\OneDrive\Área de Trabalho\atividade para fazer hoje\UC-14\2-static-c is>`

ex2.js:

The screenshot shows the VS Code editor interface with a file named `ex2.js` open. The editor displays a JSON object with the following content:

```
12  },
13  "author": "",
14  "license": "ISC",
15  "description": "",
16  "devDependencies": {
17    "@eslint/js": "^9.15.0",
18    "eslint": "^9.15.0",
19    "globals": "^15.12.0"
20  }
```

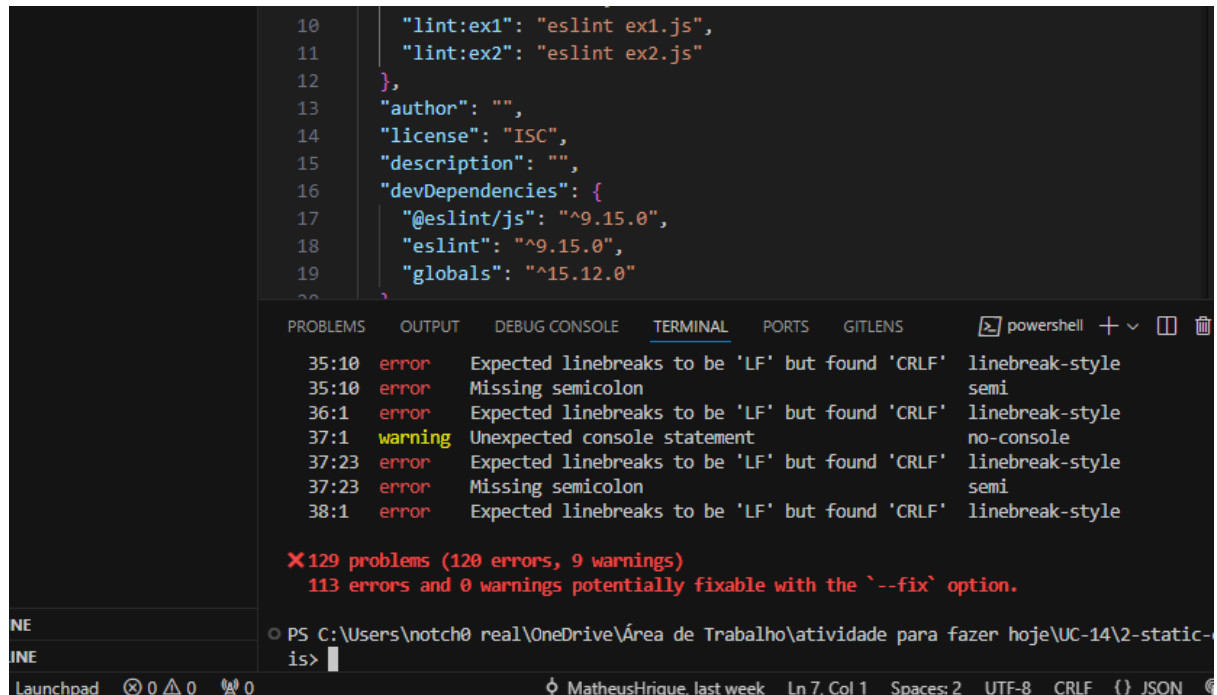
The bottom panel of the editor shows the **PROBLEMS** view with the following errors:

- 35:10 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
- 35:10 error Missing semicolon semi
- 36:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
- 37:1 warning Unexpected console statement no-console
- 37:23 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
- 37:23 error Missing semicolon semi
- 38:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style

A summary message states: **74 problems (68 errors, 6 warnings)**. 62 errors and 0 warnings potentially fixable with the `--fix` option.

The terminal shows the command prompt: `PS C:\Users\notch0 real\OneDrive\Área de Trabalho\atividade para fazer hoje\UC-14\2-static-code is>`

total:



```
10  "lint:ex1": "eslint ex1.js",
11  "lint:ex2": "eslint ex2.js"
12  },
13  "author": "",
14  "license": "ISC",
15  "description": "",
16  "devDependencies": {
17    "@eslint/js": "^9.15.0",
18    "eslint": "^9.15.0",
19    "globals": "^15.12.0"
20  }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS powershell + -

35:10 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
35:10 error Missing semicolon semi
36:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
37:1 warning Unexpected console statement no-console
37:23 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style
37:23 error Missing semicolon semi
38:1 error Expected linebreaks to be 'LF' but found 'CRLF' linebreak-style

✖ 129 problems (120 errors, 9 warnings)
113 errors and 0 warnings potentially fixable with the `--fix` option.

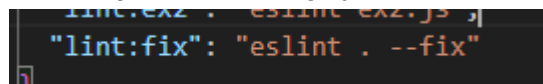
PS C:\Users\notch0 real\OneDrive\Área de Trabalho\atividade para fazer hoje\UC-14\2-static-
is>

Launchpad 0 0 0 MatheusHrique, last week Ln 7, Col 1 Spaces: 2 UTF-8 CRLF {} JSON

#4 relatório: Terminando de aplicar configurações necessárias e adaptando o código para a finalização dos exercícios

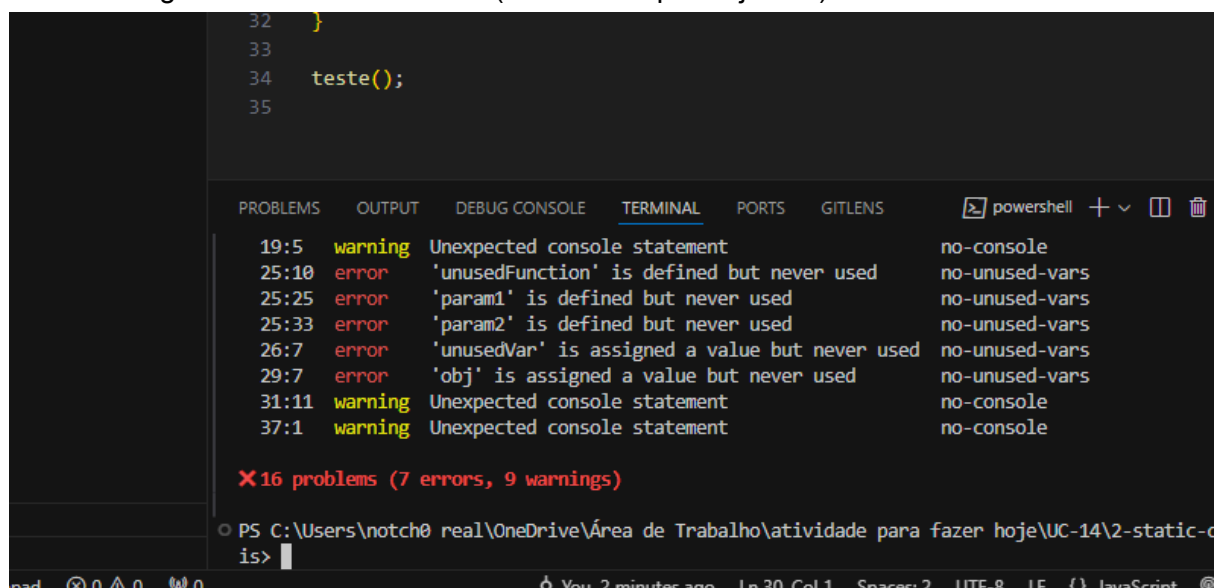
Para corrigir automaticamente, usando “npm run lint:fix”.

Modificações no package.json em “scripts”:



```
"lint:fix": "eslint . --fix"
```

Resolveu a grande maioria dos erros (dos dois arquivos juntos):



```
32  }
33
34  teste();
35

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS powershell + -

19:5 warning Unexpected console statement no-console
25:10 error 'unusedFunction' is defined but never used no-unused-vars
25:25 error 'param1' is defined but never used no-unused-vars
25:33 error 'param2' is defined but never used no-unused-vars
26:7 error 'unusedVar' is assigned a value but never used no-unused-vars
29:7 error 'obj' is assigned a value but never used no-unused-vars
31:11 warning Unexpected console statement no-console
37:1 warning Unexpected console statement no-console

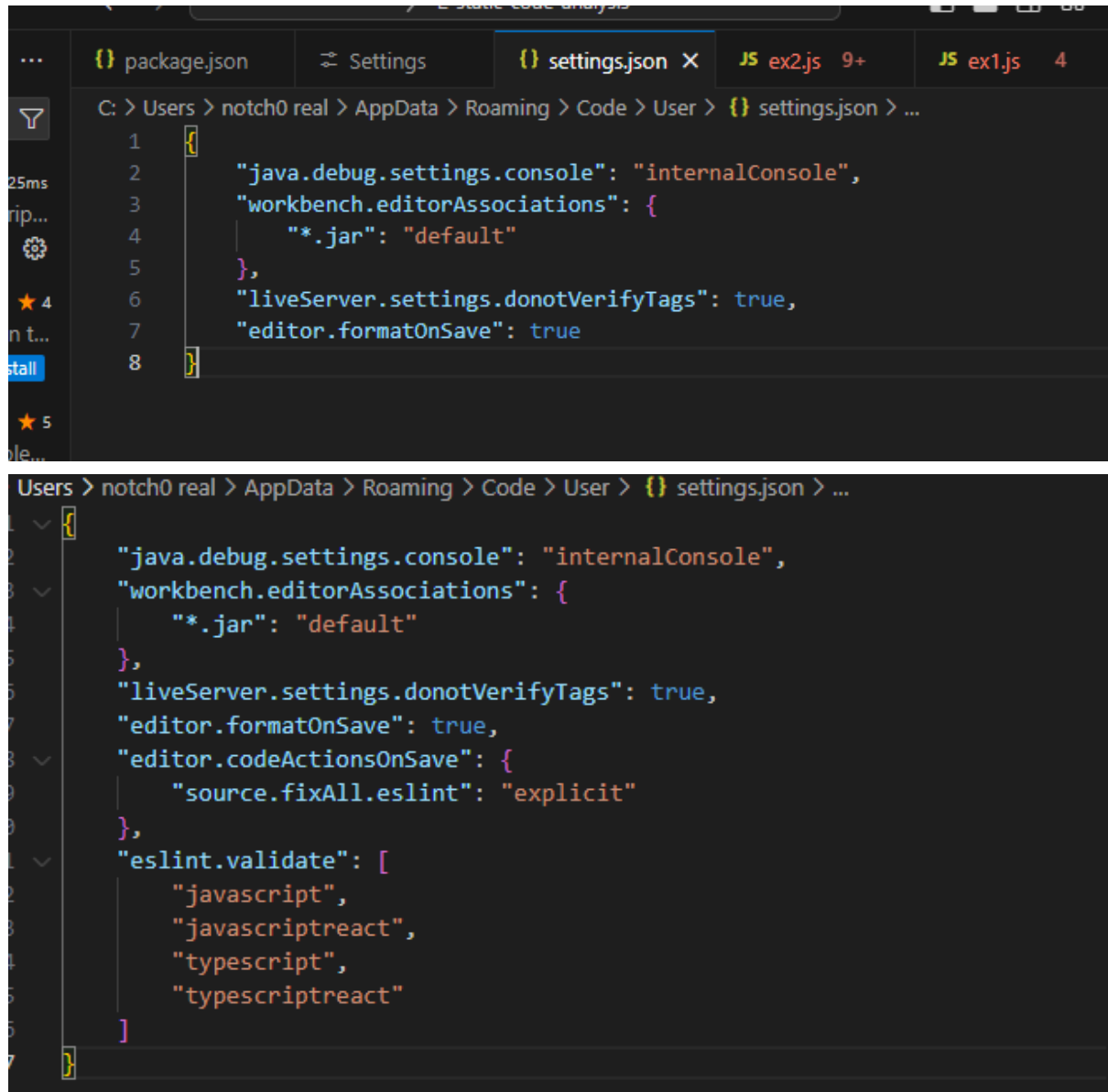
✖ 16 problems (7 errors, 9 warnings)

PS C:\Users\notch0 real\OneDrive\Área de Trabalho\atividade para fazer hoje\UC-14\2-static-
is>

pad 0 0 0 You 2 minutes ago Ln 30, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

[illegible]

Adaptando o settings.json para fazer a correção no momento que o arquivo for salvo:

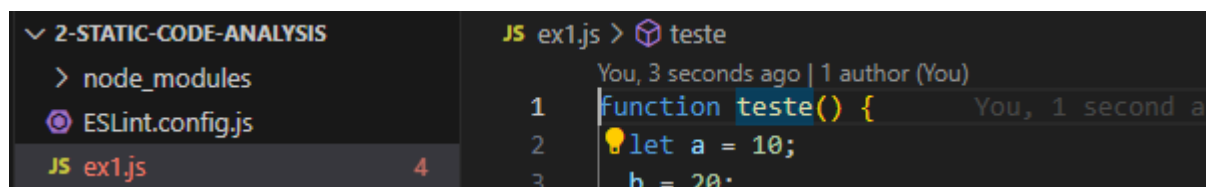


```
1 {
2   "java.debug.settings.console": "internalConsole",
3   "workbench.editorAssociations": {
4     "*.jar": "default"
5   },
6   "liveServer.settings.donotVerifyTags": true,
7   "editor.formatOnSave": true
8 }
```

```
1 {
2   "java.debug.settings.console": "internalConsole",
3   "workbench.editorAssociations": {
4     "*.jar": "default"
5   },
6   "liveServer.settings.donotVerifyTags": true,
7   "editor.formatOnSave": true,
8   "editor.codeActionsOnSave": {
9     "source.fixAll.eslint": "explicit"
10  },
11  "eslint.validate": [
12    "javascript",
13    "javascriptreact",
14    "typescript",
15    "typescriptreact"
16  ]
17 }
```

Para desabilitar o linter em um arquivo inteiro, apenas é necessário adicionar um comentário no topo do arquivo, colocando: `/* eslint-disable */`
Por exemplo:

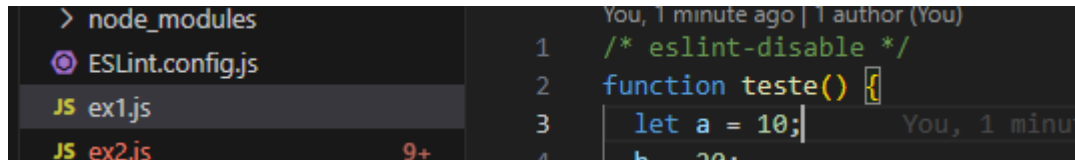
Sem `/* eslint-disable */`



```
2-STATIC-CODE-ANALYSIS
> node_modules
ESLint.config.js
JS ex1.js 4
```

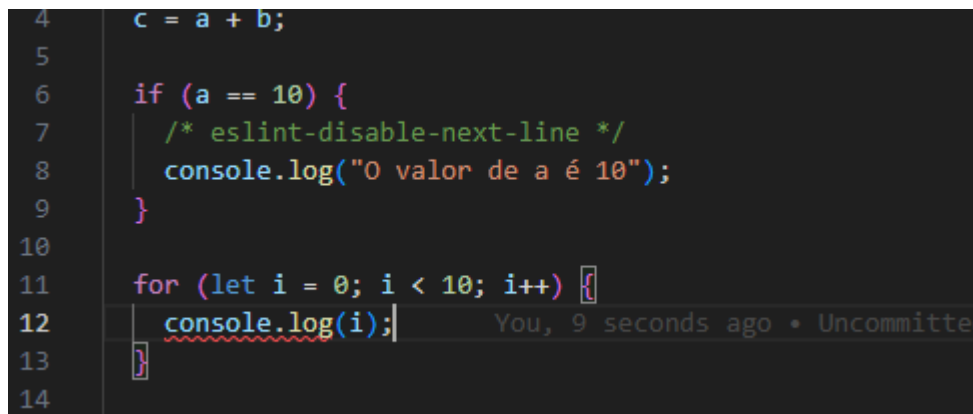
```
JS ex1.js > teste
You, 3 seconds ago | 1 author (You)
1 function teste() {
2   let a = 10;
3   b = 20;
```


Com `/* eslint-disable */`



```
1  /* eslint-disable */
2  function teste() {
3      let a = 10;
4      b = 20;
```

Para desabilitar o linter em linhas específicas com `console.log`, apenas é necessário colocar um comentário `/* eslint-disable-next-line */` antes da linha:



```
4  c = a + b;
5
6  if (a == 10) {
7      /* eslint-disable-next-line */
8      console.log("O valor de a é 10");
9  }
10
11  for (let i = 0; i < 10; i++) {
12      console.log(i);
13  }
```

Versão final do código & arquivos no repositório!