LINTER

Comprobación de lenguajes de programación

21 DE NOVIEMBRE DE 2017

EMILIANO MONTESDEOCA DEL PUERTO
2DAW B – CIFP CESAR MANRIQUE

Índice

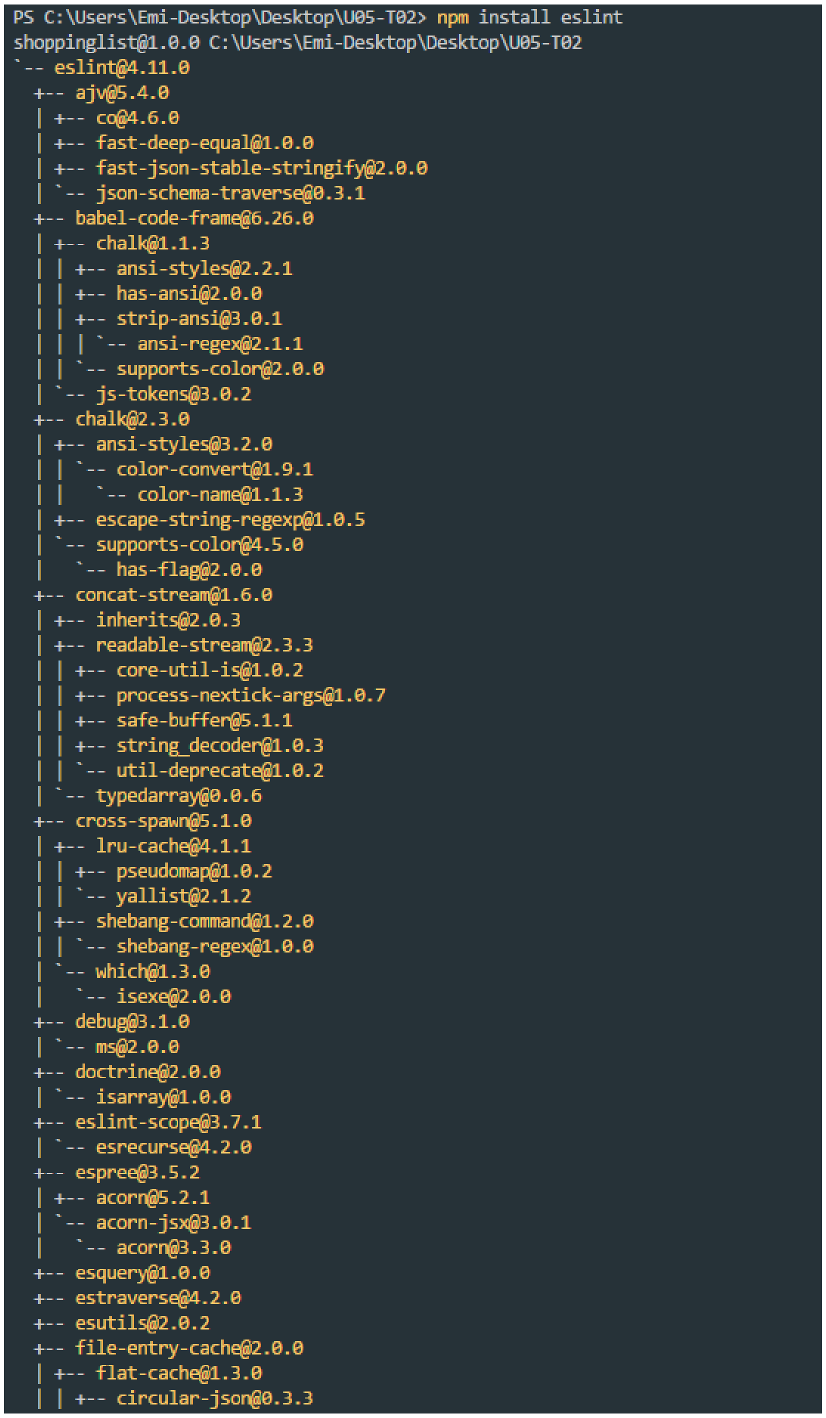
¿Linter?

Lint es una herramienta de programación; originalmente **lint** era el nombre de una herramienta de programación utilizada para detectar código sospechoso, confuso o incompatible entre distintas arquitecturas en programas escritos en C; es decir, errores de programación que escapan al habitual análisis sintáctico que hace el compilador.

En la actualidad, se utiliza este término para designar a herramientas que realizan estas tareas de comprobación en cualquier lenguaje de programación. Las herramientas de tipo *lint* generalmente funcionan realizando un análisis estático del código fuente.

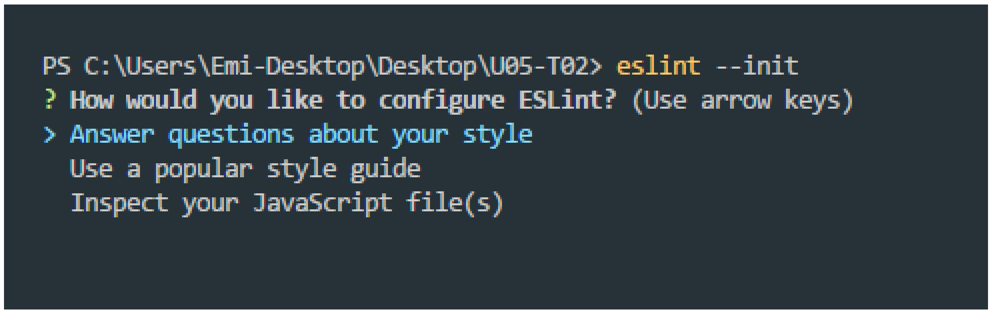
Instalación en proyecto

Primero hay que instalar **npm** e instalar **eslint** en el proyecto, esto se realiza con **npm install eslint**. Esto nos agregara dependencias a **node_modules**.



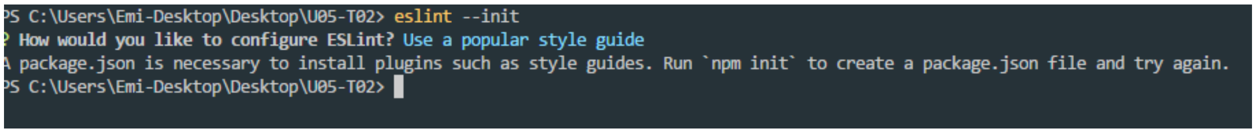
```
PS C:\Users\Emi-Desktop\Desktop\U05-T02> npm install eslint
shoppinglist@1.0.0 C:\Users\Emi-Desktop\Desktop\U05-T02
`-- eslint@4.11.0
  +-- ajv@5.4.0
  | +-- co@4.6.0
  | +-- fast-deep-equal@1.0.0
  | +-- fast-json-stable-stringify@2.0.0
  | `-- json-schema-traverse@0.3.1
  +-- babel-code-frame@6.26.0
  | +-- chalk@1.1.3
  | | +-- ansi-styles@2.2.1
  | | +-- has-ansi@2.0.0
  | | +-- strip-ansi@3.0.1
  | | | `-- ansi-regex@2.1.1
  | | `-- supports-color@2.0.0
  | `-- js-tokens@3.0.2
  +-- chalk@2.3.0
  | +-- ansi-styles@3.2.0
  | | `-- color-convert@1.9.1
  | |   `-- color-name@1.1.3
  | +-- escape-string-regexp@1.0.5
  | `-- supports-color@4.5.0
  |   `-- has-flag@2.0.0
  +-- concat-stream@1.6.0
  | +-- inherits@2.0.3
  | +-- readable-stream@2.3.3
  | | +-- core-util-is@1.0.2
  | | +-- process-nextick-args@1.0.7
  | | +-- safe-buffer@5.1.1
  | | +-- string_decoder@1.0.3
  | | `-- util-deprecate@1.0.2
  | `-- typedarray@0.0.6
  +-- cross-spawn@5.1.0
  | +-- lru-cache@4.1.1
  | | +-- pseudomap@1.0.2
  | | `-- yallist@2.1.2
  | +-- shebang-command@1.2.0
  | | `-- shebang-regex@1.0.0
  | `-- which@1.3.0
  |   `-- isexe@2.0.0
  +-- debug@3.1.0
  | `-- ms@2.0.0
  +-- doctrine@2.0.0
  | `-- isarray@1.0.0
  +-- eslint-scope@3.7.1
  | `-- esrecurse@4.2.0
  +-- espree@3.5.2
  | +-- acorn@5.2.1
  | `-- acorn-jsx@3.0.1
  |   `-- acorn@3.3.0
  +-- esquery@1.0.0
  +-- estraverse@4.2.0
  +-- esutils@2.0.2
  +-- file-entry-cache@2.0.0
  | +-- flat-cache@1.3.0
  | | +-- circular-json@0.3.3
```

Para agregarlo en un proyecto, en mi IDE, Visual Studio Code, hay que en consola realizar un eslint –init



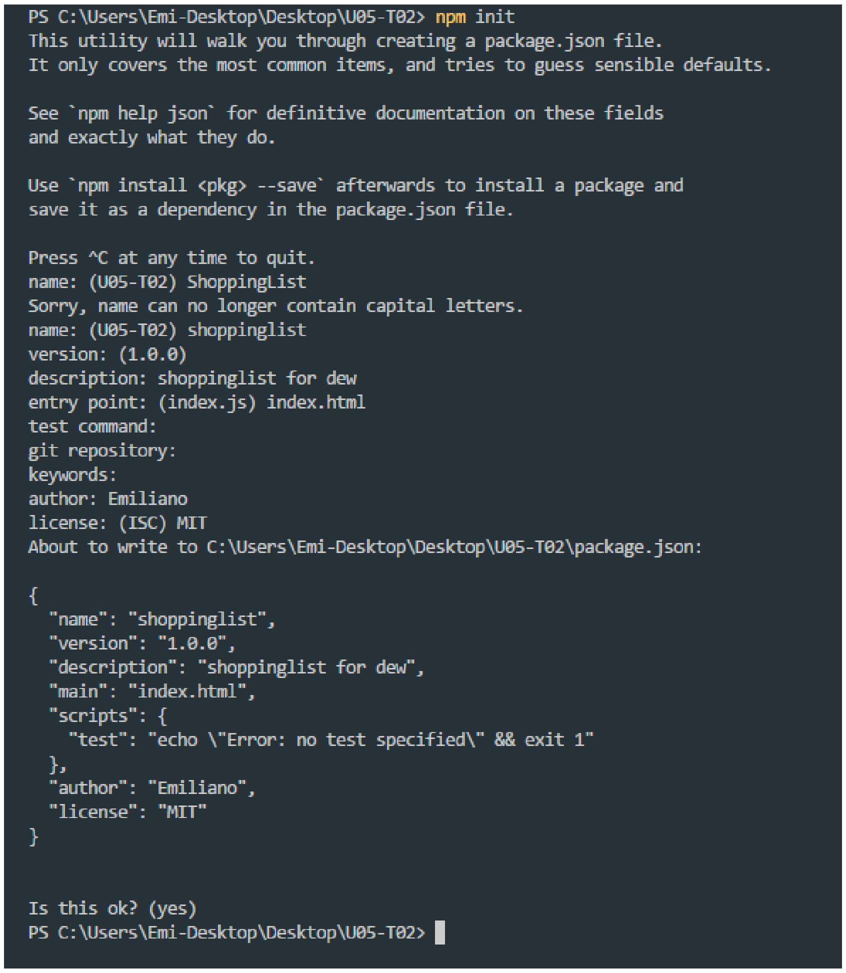
```
PS C:\Users\Emi-Desktop\Desktop\U05-T02> eslint --init
? How would you like to configure ESLint? (Use arrow keys)
> Answer questions about your style
  Use a popular style guide
  Inspect your JavaScript file(s)
```

Luego podemos elegir como quiere configurar **ESLint**, pero no nos va a dejar porque nuestro proyecto no es un proyecto npm, por lo que no tiene paquete de dependencias.



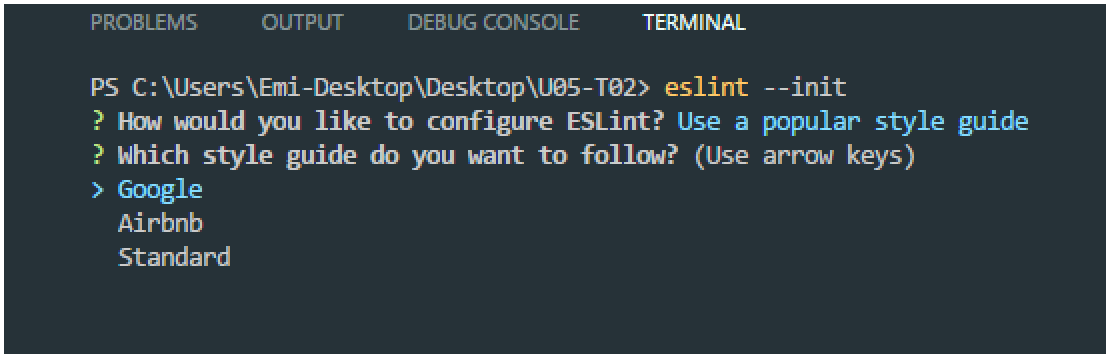
```
S C:\Users\Emi-Desktop\Desktop\U05-T02> eslint --init
 How would you like to configure ESLint? Use a popular style guide
 package.json is necessary to install plugins such as style guides. Run `npm init` to create a package.json file and try again.
S C:\Users\Emi-Desktop\Desktop\U05-T02>
```

Asi que inicializamos el proyecto para **npm** hay que rellenar unas preguntas y se pondrá a crear la estructura, como podremos ver en la siguiente imagen.



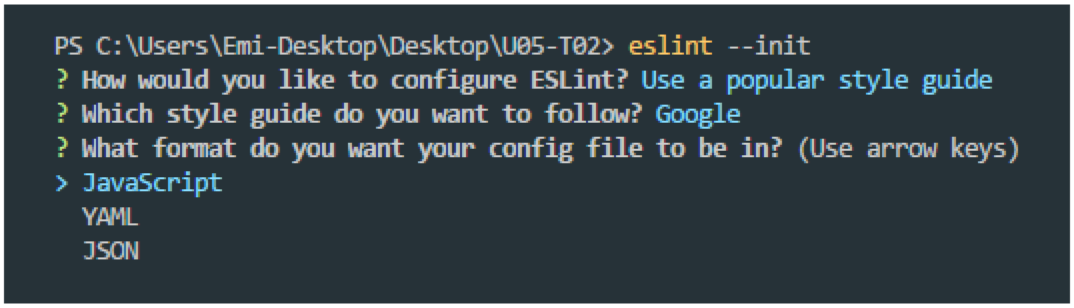
```
PS C:\Users\Emi-Desktop\Desktop\U05-T02> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

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

Press ^C at any time to quit.
name: (U05-T02) ShoppingList
Sorry, name can no longer contain capital letters.
name: (U05-T02) shoppinglist
version: (1.0.0)
description: shoppinglist for dew
entry point: (index.js) index.html
test command:
git repository:
keywords:
author: Emiliano
license: (ISC) MIT
About to write to C:\Users\Emi-Desktop\Desktop\U05-T02\package.json:

{
  "name": "shoppinglist",
  "version": "1.0.0",
  "description": "shoppinglist for dew",
  "main": "index.html",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Emiliano",
  "license": "MIT"
}


Is this ok? (yes)
PS C:\Users\Emi-Desktop\Desktop\U05-T02>
```

Ahora ejecutamos le comando **eslint –init** de nuevo, para asi agregar las dependencias de **eslint**.


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

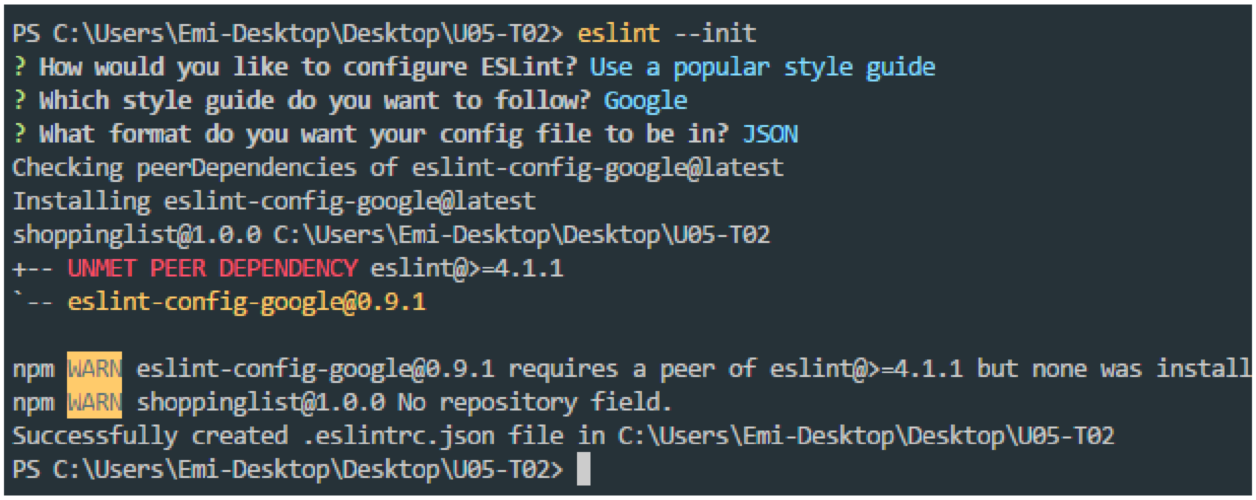
```
PS C:\Users\Emi-Desktop\Desktop\U05-T02> eslint --init
? How would you like to configure ESLint? Use a popular style guide
? Which style guide do you want to follow? (Use arrow keys)
> Google
  Airbnb
  Standard
```

Ahora como se puede ver, ya te dice para poder utilizar de los **linters** más populares, como son el de **Airbnb** o **Google**.



```
PS C:\Users\Emi-Desktop\Desktop\U05-T02> eslint --init
? How would you like to configure ESLint? Use a popular style guide
? Which style guide do you want to follow? Google
? What format do you want your config file to be in? (Use arrow keys)
> JavaScript
  YAML
  JSON
```

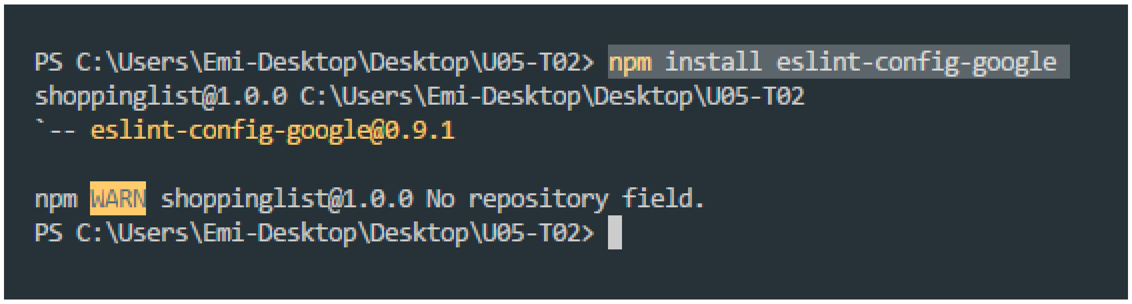
Luego nos pregunta cómo queremos guardar la configuración, lo mas lógico es utilizar un **JSON** para guardar la configuración.



```
PS C:\Users\Emi-Desktop\Desktop\U05-T02> eslint --init
? How would you like to configure ESLint? Use a popular style guide
? Which style guide do you want to follow? Google
? What format do you want your config file to be in? JSON
Checking peerDependencies of eslint-config-google@latest
Installing eslint-config-google@latest
shoppinglist@1.0.0 C:\Users\Emi-Desktop\Desktop\U05-T02
+-- UNMET PEER DEPENDENCY eslint@>=4.1.1
`-- eslint-config-google@0.9.1

npm WARN eslint-config-google@0.9.1 requires a peer of eslint@>=4.1.1 but none was install
npm WARN shoppinglist@1.0.0 No repository field.
Successfully created .eslintrc.json file in C:\Users\Emi-Desktop\Desktop\U05-T02
PS C:\Users\Emi-Desktop\Desktop\U05-T02>
```

Una vez terminado, se puede ver como se ha creado el archivo **.eslintrc.json**, donde se encuentran los parámetros para el **linter**.

En mi caso había una error porque no detectaba el **ESlint** instalado, se ha resuelto con **npm install eslint**, luego he ejecutado **npm install eslint-config-google**

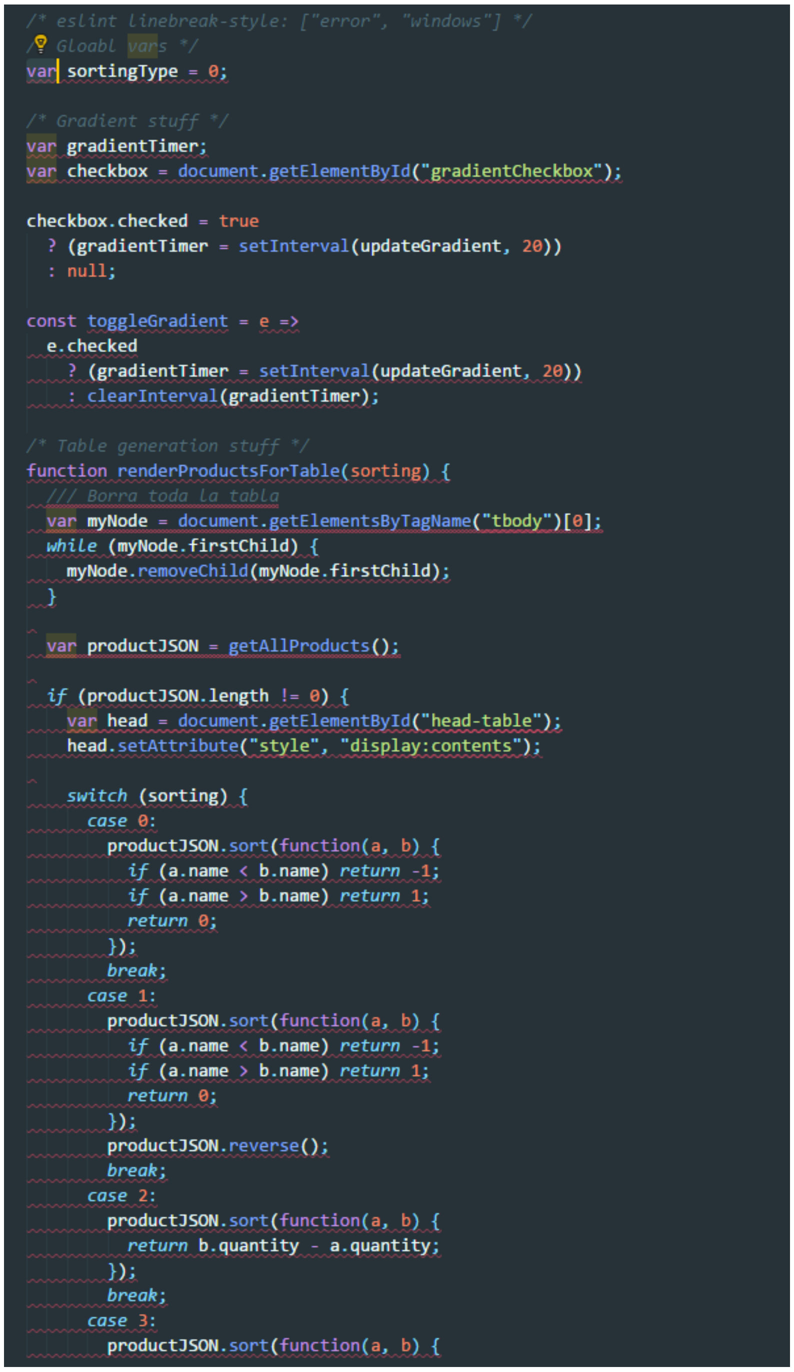


```
PS C:\Users\Emi-Desktop\Desktop\U05-T02> npm install eslint-config-google
shoppinglist@1.0.0 C:\Users\Emi-Desktop\Desktop\U05-T02
`-- eslint-config-google@0.9.1

npm WARN shoppinglist@1.0.0 No repository field.
PS C:\Users\Emi-Desktop\Desktop\U05-T02>
```

Utilización en proyecto

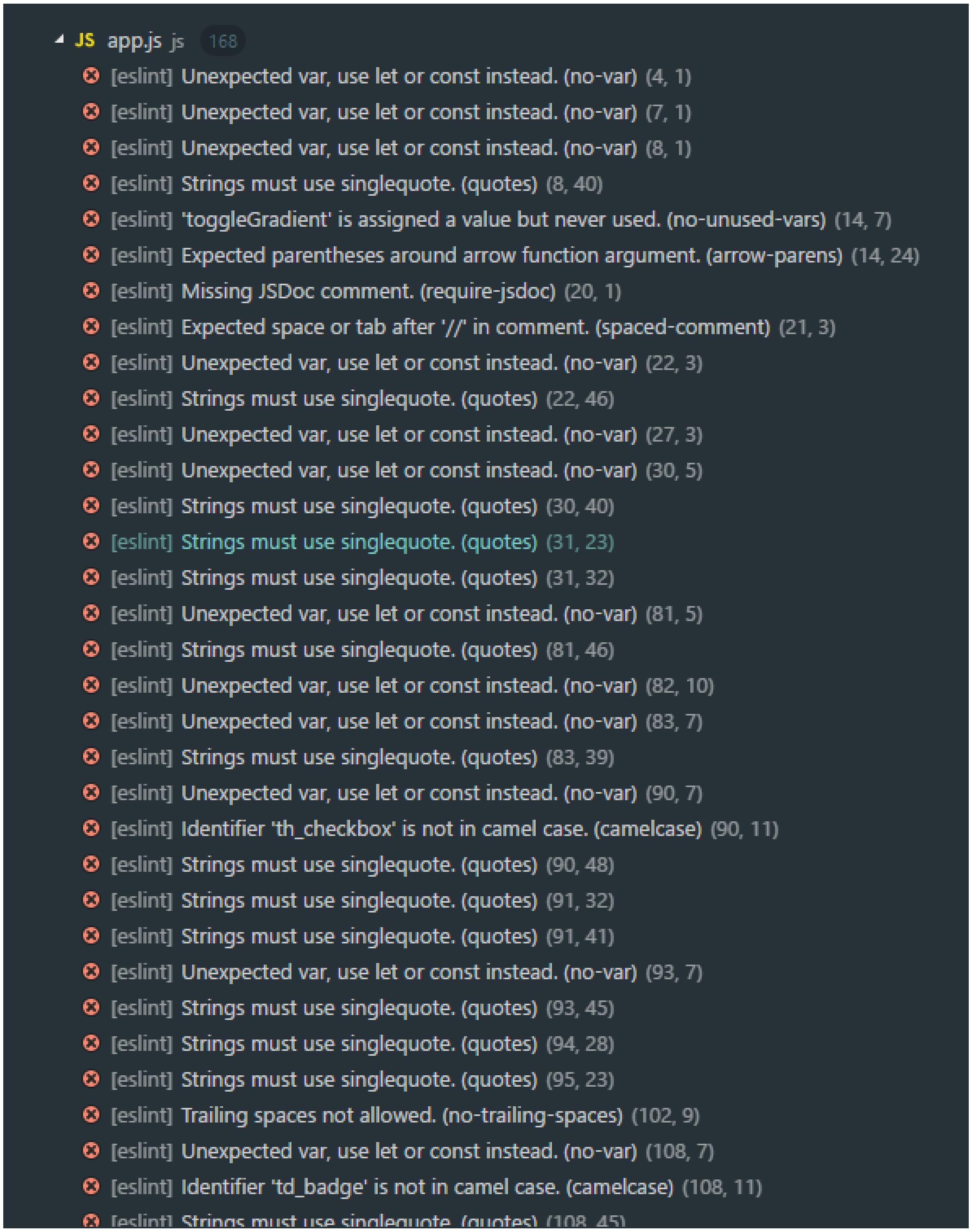
Una vez instalado todo, simplemente tenemos que abrir un archivo JavaScript situado en nuestro proyecto y el **linter** se pondrá a trabajar.

Utilizando el archivo app.js nos muestra bastantes errores.



```
/* eslint linebreak-style: ["error", "windows"] */
/* Gloabl vars */
var sortingType = 0;

/* Gradient stuff */
var gradientTimer;
var checkbox = document.getElementById("gradientCheckbox");

checkbox.checked = true
  ? (gradientTimer = setInterval(updateGradient, 20))
  : null;

const toggleGradient = e =>
  e.checked
    ? (gradientTimer = setInterval(updateGradient, 20))
    : clearInterval(gradientTimer);

/* Table generation stuff */
function renderProductsForTable(sorting) {
  /// Borra toda la tabla
  var myNode = document.getElementsByTagName("tbody")[0];
  while (myNode.firstChild) {
    myNode.removeChild(myNode.firstChild);
  }

  var productJSON = getAllProducts();

  if (productJSON.length != 0) {
    var head = document.getElementById("head-table");
    head.setAttribute("style", "display:contents");

    switch (sorting) {
      case 0:
        productJSON.sort(function(a, b) {
          if (a.name < b.name) return -1;
          if (a.name > b.name) return 1;
          return 0;
        });
        break;
      case 1:
        productJSON.sort(function(a, b) {
          if (a.name < b.name) return -1;
          if (a.name > b.name) return 1;
          return 0;
        });
        productJSON.reverse();
        break;
      case 2:
        productJSON.sort(function(a, b) {
          return b.quantity - a.quantity;
        });
        break;
      case 3:
        productJSON.sort(function(a, b) {
```

Tenemos suerte que, el **ESLint** también tiene documentada la solución:

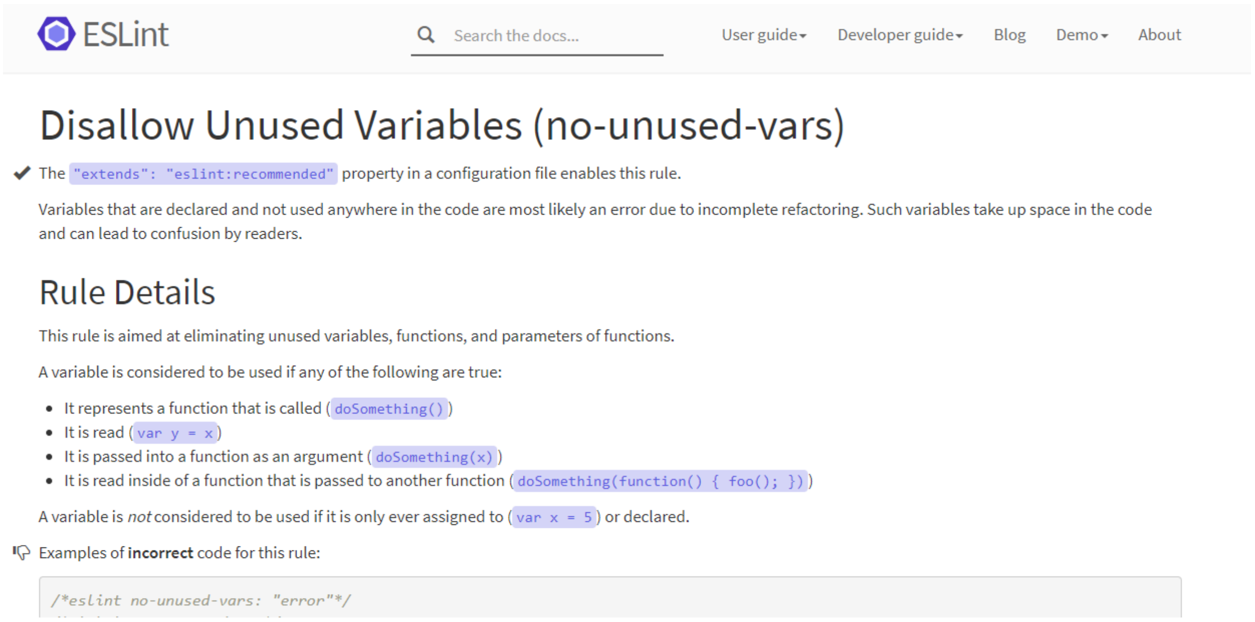


```
JS app.js js 168
  [eslint] Unexpected var, use let or const instead. (no-var) (4, 1)
  [eslint] Unexpected var, use let or const instead. (no-var) (7, 1)
  [eslint] Unexpected var, use let or const instead. (no-var) (8, 1)
  [eslint] Strings must use singlequote. (quotes) (8, 40)
  [eslint] 'toggleGradient' is assigned a value but never used. (no-unused-vars) (14, 7)
  [eslint] Expected parentheses around arrow function argument. (arrow-parens) (14, 24)
  [eslint] Missing JSDoc comment. (require-jsdoc) (20, 1)
  [eslint] Expected space or tab after '//' in comment. (spaced-comment) (21, 3)
  [eslint] Unexpected var, use let or const instead. (no-var) (22, 3)
  [eslint] Strings must use singlequote. (quotes) (22, 46)
  [eslint] Unexpected var, use let or const instead. (no-var) (27, 3)
  [eslint] Unexpected var, use let or const instead. (no-var) (30, 5)
  [eslint] Strings must use singlequote. (quotes) (30, 40)
  [eslint] Strings must use singlequote. (quotes) (31, 23)
  [eslint] Strings must use singlequote. (quotes) (31, 32)
  [eslint] Unexpected var, use let or const instead. (no-var) (81, 5)
  [eslint] Strings must use singlequote. (quotes) (81, 46)
  [eslint] Unexpected var, use let or const instead. (no-var) (82, 10)
  [eslint] Unexpected var, use let or const instead. (no-var) (83, 7)
  [eslint] Strings must use singlequote. (quotes) (83, 39)
  [eslint] Unexpected var, use let or const instead. (no-var) (90, 7)
  [eslint] Identifier 'th_checkbox' is not in camel case. (camelcase) (90, 11)
  [eslint] Strings must use singlequote. (quotes) (90, 48)
  [eslint] Strings must use singlequote. (quotes) (91, 32)
  [eslint] Strings must use singlequote. (quotes) (91, 41)
  [eslint] Unexpected var, use let or const instead. (no-var) (93, 7)
  [eslint] Strings must use singlequote. (quotes) (93, 45)
  [eslint] Strings must use singlequote. (quotes) (94, 28)
  [eslint] Strings must use singlequote. (quotes) (95, 23)
  [eslint] Trailing spaces not allowed. (no-trailing-spaces) (102, 9)
  [eslint] Unexpected var, use let or const instead. (no-var) (108, 7)
  [eslint] Identifier 'td_badge' is not in camel case. (camelcase) (108, 11)
  [eslint] Strings must use singlequote. (quotes) (108, 45)
```

Como podemos ver da una serie de errores que hay que corregir:

1. **String tiene que usar comas simples**
2. **No utilizar var**
3. **Identificadores tienen que estar en camel case**
4. **Los comentarios tienen que ser de 2 barras (//)**
5. **Funciones no utilizadas**
6. **Variables no utilizadas**
7. **Falta comentario JSDoc para funciones**

Pero es fácil de solucionar, ya que se puede buscar cómo hacerlo bien en la página de **ESlint**.



Disallow Unused Variables (no-unused-vars)

✓ The `"extends": "eslint:recommended"` property in a configuration file enables this rule.

Variables that are declared and not used anywhere in the code are most likely an error due to incomplete refactoring. Such variables take up space in the code and can lead to confusion by readers.

Rule Details

This rule is aimed at eliminating unused variables, functions, and parameters of functions.

A variable is considered to be used if any of the following are true:

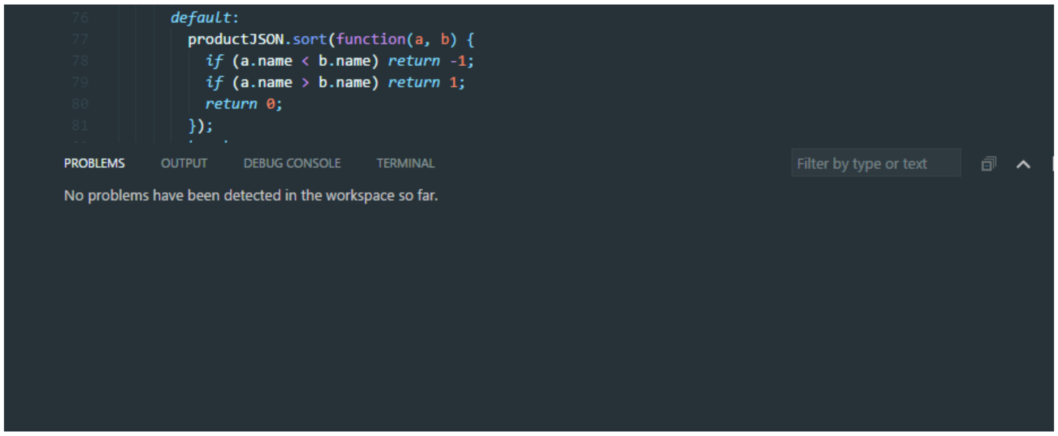
- It represents a function that is called (`doSomething()`)
- It is read (`var y = x`)
- It is passed into a function as an argument (`doSomething(x)`)
- It is read inside of a function that is passed to another function (`doSomething(function() { foo(); })`)

A variable is *not* considered to be used if it is only ever assigned to (`var x = 5`) or declared.

Examples of **incorrect** code for this rule:

```
/*eslint no-unused-vars: "error"*/
```

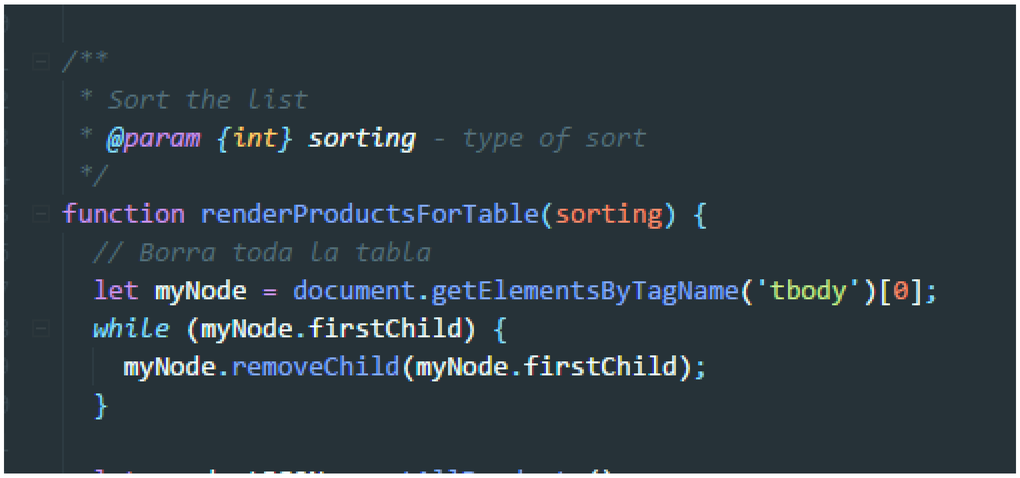
Una vez cambiado nuestro código siguiendo las normas que nos pone el linter, el código no da mas errores.



```
    default:
      productJSON.sort(function(a, b) {
        if (a.name < b.name) return -1;
        if (a.name > b.name) return 1;
        return 0;
      });
```

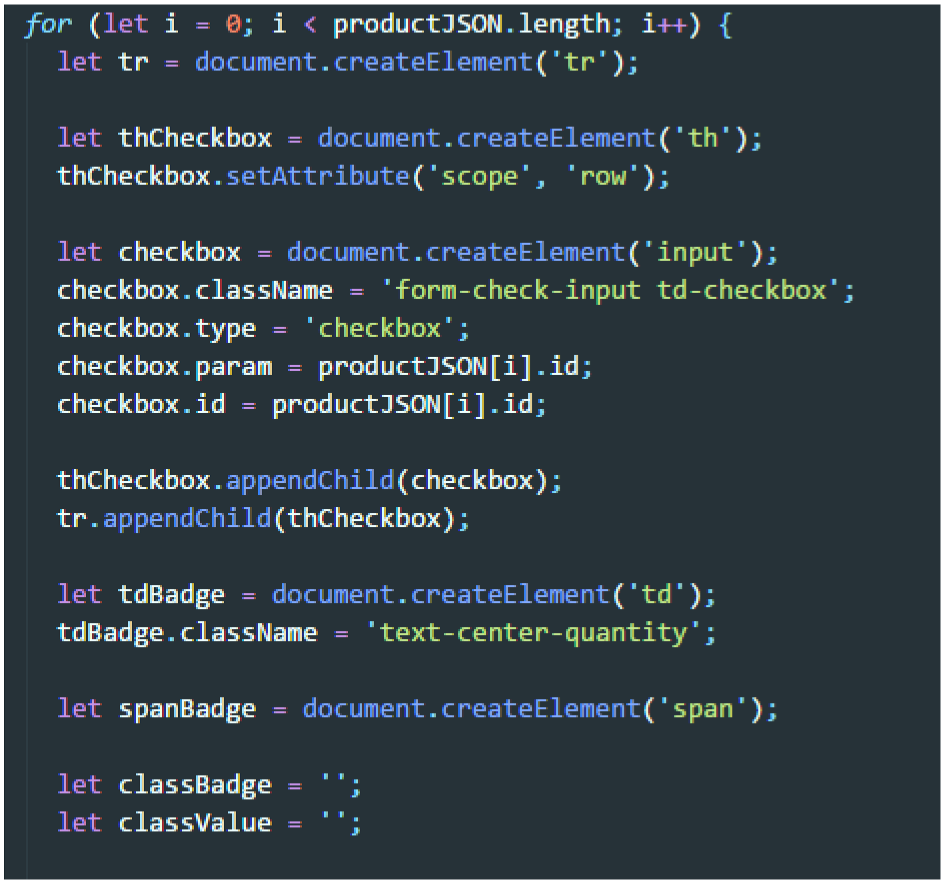
No problems have been detected in the workspace so far.

He agregado comentarios **JSDoc** a las funciones



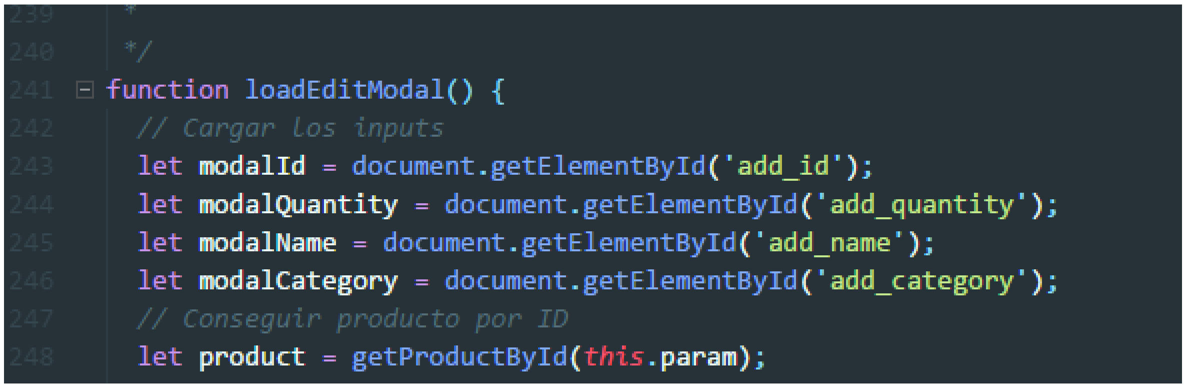
```
/**
 * Sort the list
 * @param {int} sorting - type of sort
 */
function renderProductsForTable(sorting) {
  // Borra toda la tabla
  let myNode = document.getElementsByTagName('tbody')[0];
  while (myNode.firstChild) {
    myNode.removeChild(myNode.firstChild);
  }
```

He cambiado a **camelCase** las variables



```
for (let i = 0; i < productJSON.length; i++) {
  let tr = document.createElement('tr');

  let thCheckbox = document.createElement('th');
  thCheckbox.setAttribute('scope', 'row');

  let checkbox = document.createElement('input');
  checkbox.className = 'form-check-input td-checkbox';
  checkbox.type = 'checkbox';
  checkbox.param = productJSON[i].id;
  checkbox.id = productJSON[i].id;

  thCheckbox.appendChild(checkbox);
  tr.appendChild(thCheckbox);

  let tdBadge = document.createElement('td');
  tdBadge.className = 'text-center-quantity';

  let spanBadge = document.createElement('span');

  let classBadge = '';
  let classValue = '';
```

He cambiado los **vars** por **lets**



```
 *
 */
function loadEditModal() {
  // Cargar los inputs
  let modalId = document.getElementById('add_id');
  let modalQuantity = document.getElementById('add_quantity');
  let modalName = document.getElementById('add_name');
  let modalCategory = document.getElementById('add_category');
  // Conseguir producto por ID
  let product = getProductById(this.param);
```

En definitiva

Las utilizaciones de estas herramientas sirven para no solo escribir **clean code** sino que sirve para mejorar la forma de programar siguiendo las pautas de profesionales como Google o Airbnb.

Vale la pena aprender a utilizar el lintel y aplicarlo diariamente a las prácticas de cada uno.

Bibliografía

1. https://es.wikipedia.org/wiki/Lint
2. https://eslint.org/