

Linter

Comprobación de lenguajes de programación



21 de noviembre de 2017

Emiliano montesdeoca del puerto

2DAW B – CIFP CESAR MANRIQUE

Índice

1. ¿Linter?
2. Instalación en proyecto
3. Utilización en proyecto
4. En definitiva
5. Bibliografía

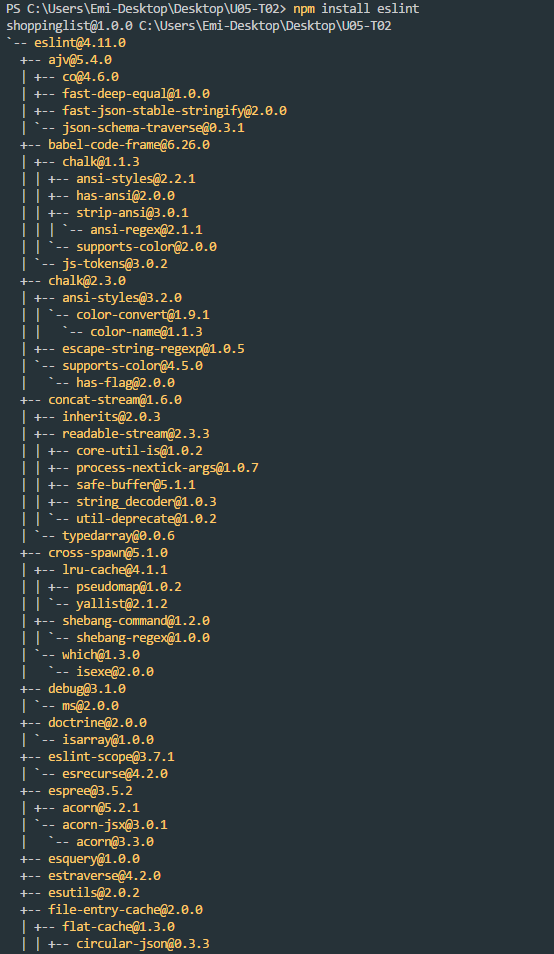
¿Linter?

**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.

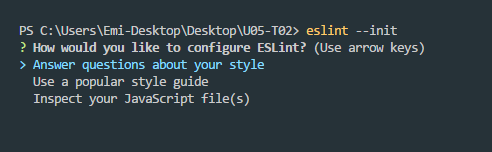
En la actualidad, se utiliza este término para designar a herramientas que realizan estas tareas de comprobación en cualquier lenguaje de programación. Las herramientas de tipo *lint* generalmente funcionan realizando un análisis estático del código fuente.

Instalación en proyecto

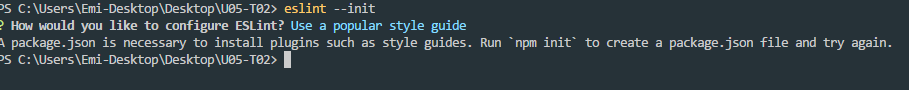
Primero hay que instalar **npm** e instalar **eslint** en el proyecto, esto se realiza con **npm install eslint**. Esto nos agregara dependencias a **node\_modules**.



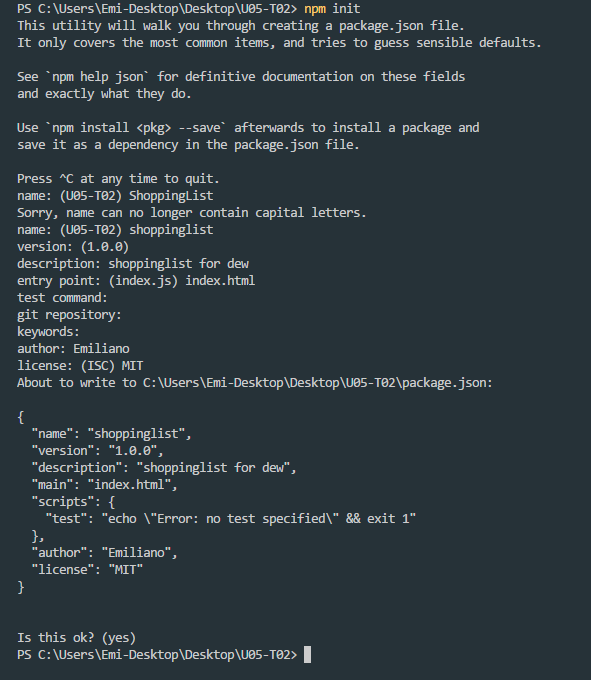
Para agregarlo en un proyecto, en mi IDE, Visual Studio Code, hay que en consola realizar un eslint –init



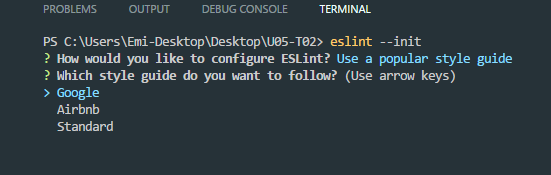
Luego podemos elegir como quiere configurar **ESLint**, pero no nos va a dejar porque nuestro proyecto no es un proyecto npm, por lo que no tiene paquete de dependencias.



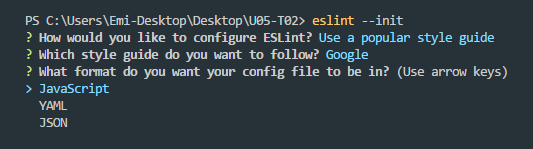
Asi que inicializamos el proyecto para **npm** hay que rellenar unas preguntas y se pondrá a crear la estructura, como podremos ver en la siguiente imagen.



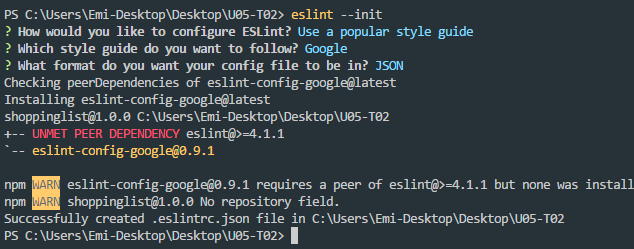
Ahora ejecutamos le comando **eslint –init** de nuevo, para asi agregar las dependencias de **eslint**.



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

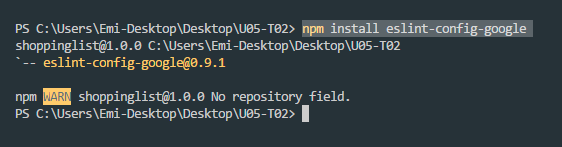


Luego nos pregunta cómo queremos guardar la configuración, lo mas lógico es utilizar un **JSON** para guardar la configuración.



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

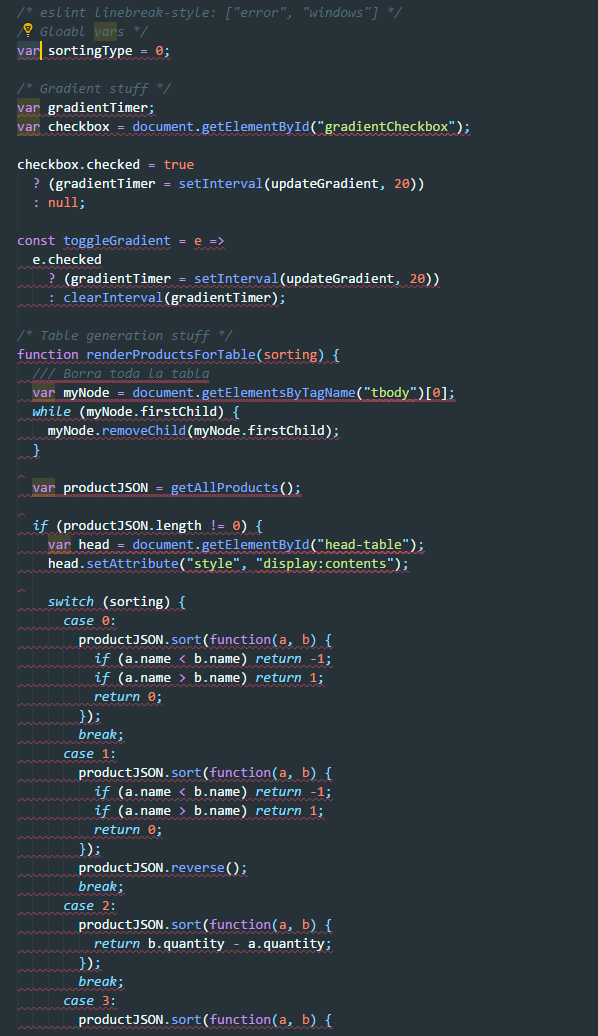
En mi caso había una error porque no detectaba el **ESlint** instalado, se ha resuelto con **npm install eslint**, luego he ejecutado **npm install eslint-config-google**



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.



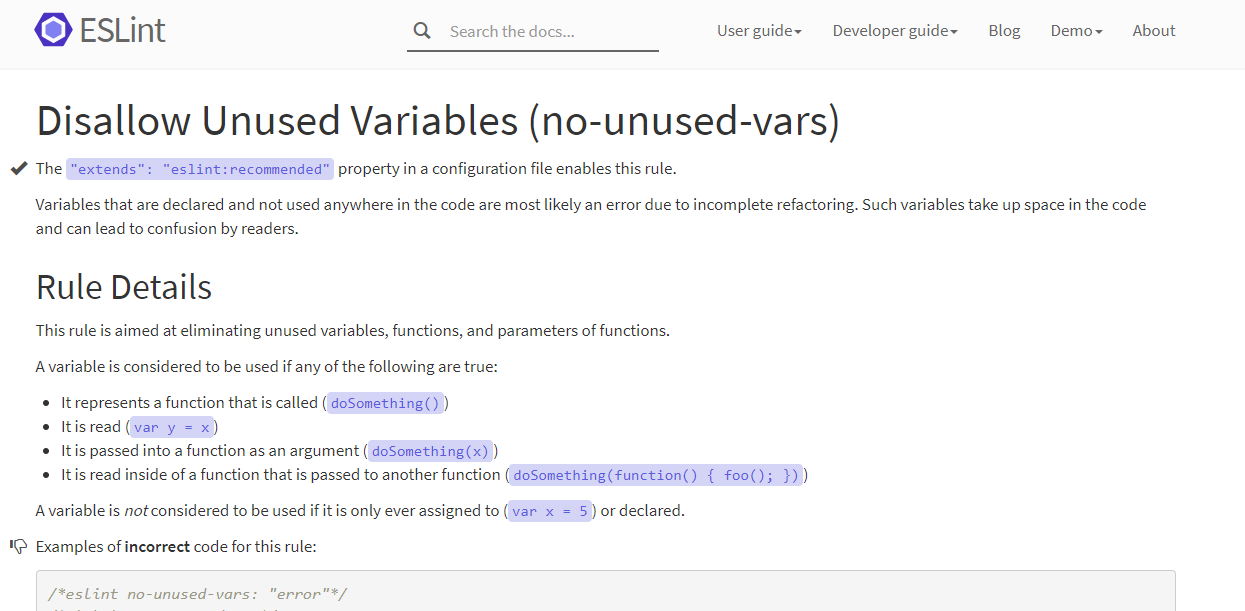
Tenemos suerte que, el **ESLint** también tiene documentada la solución:



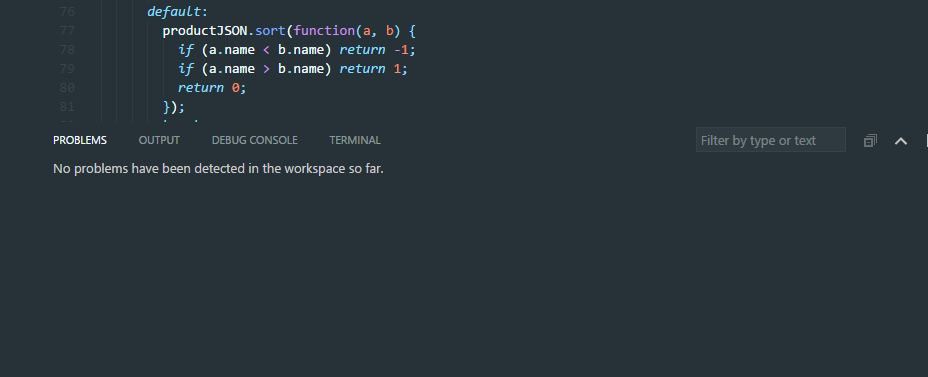
Como podemos ver da una serie de errores que hay que corregir:

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**

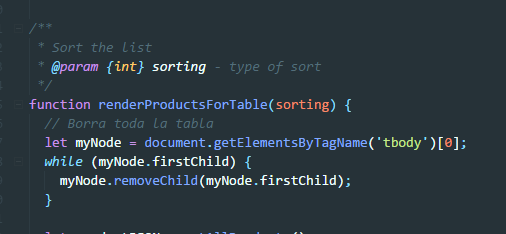
Pero es fácil de solucionar, ya que se puede buscar cómo hacerlo bien en la página de **ESlint**.



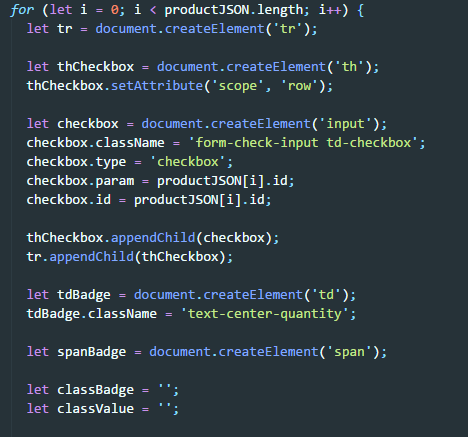
Una vez cambiado nuestro código siguiendo las normas que nos pone el linter, el código no da mas errores.



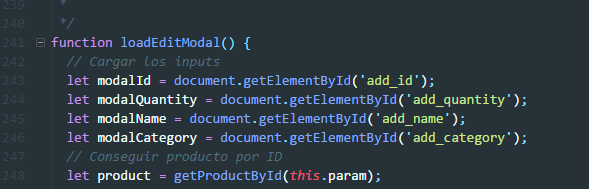
He agregado comentarios **JSDoc** a las funciones



He cambiado a **camelCase** las variables



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



**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/>