
Preparación del espacio de trabajo

Gabriel Rodríguez Flores

November 2, 2021

- ESLint
- Nodemon
- Configuración VSCode

Contents

1	Teoría	3
1.1	Nodemon	3
1.2	ESLint	3
1.2.1	Instalación	3
1.2.2	Configuración	3
1.2.3	Plugins	3
1.2.4	Uso	4
1.3	Configuración del entorno VSCode	4
2	Ejemplos	4
2.1	Ejemplo de <code>.eslintrc</code> con reglas y excepciones personalizadas	4
2.2	Ejemplo de configuración de VSCode para auto-aplicar corrección del ESLint al guardar	5
2.3	Ejemplo de configuración con otros parámetros extra	5
3	Ejercicios	5
4	Entregables	6
4.1	En clase	6
4.2	Proyecto	6

1 Teoría

1.1 Nodemon

Conocido anteriormente en los módulos NPM. Nos permite reiniciar la ejecución del código de forma automática cada vez que se edite un fichero.

- Instalación `npm i -D nodemon`
- Uso `nodemon index.js`

1.2 ESLint

1.2.1 Instalación

```
1 npm i -D eslint
```

1.2.2 Configuración

Para poder ejecutar eslint, primero hay que inicializarlo con el comando: `npx eslint --init`

Tras una serie de preguntas, tendremos configurado ESLint listo para usarse.

Se habrá generado el fichero `.eslintrc` en la extensión seleccionada, y sobre él se pueden añadir nuevas reglas y excepciones.

1.2.3 Plugins

1.2.3.1 Configuración base de Airbnb

```
1 npm i -D eslint-config-airbnb-base
```

1.2.3.2 Plugin para módulos instalados

```
1 npm i -D eslint-plugin-import
```

1.2.3.3 Plugin para AVA

```
1 npm i -D eslint-plugin-ava
```

1.2.4 Uso

- Comandos

```
1 npx eslint # Validación y visualización de errores
2 npx eslint --fix # Arreglar automáticamente los errores detectados
```

- **package.json**

```
1 {
2   ...
3   "scripts": {
4     ...
5     "lint": "eslint --ignore-path .gitignore .",
6     "lint:fix": "eslint --fix --ignore-path .gitignore .",
7     ...
8   },
9   ...
10 }
```

1.3 Configuración del entorno VSCode

La configuración del entorno es muy personal y por eso no suele incluirse en el repositorio (se añade al `.gitignore`).

Para configurar VSCode, se puede hacer una configuración particular del usuario del equipo (global) o una configuración particular para el proyecto o espacio de trabajo (workspace).

Podemos encontrar la configuración por defecto aquí

2 Ejemplos

2.1 Ejemplo de `.eslintrc` con reglas y excepciones personalizadas

```
1 {
2   "extends": [
3     "airbnb-base"
4   ],
5   "rules": {
6     "arrow-parens": [
7       "error",
8       "as-needed"
9     ],
10    // "max-classes-per-file": ["error", 2],
```

```
11     "no-underscore-dangle": ["error", { "allow": ["_id"] } ] ,
12     "no-unused-vars": [ "warn", { "argsIgnorePattern": "next" } ]
13   }
14 }
```

2.2 Ejemplo de configuración de VSCode para auto-aplicar corrección del Eslint al guardar

- settings.json

```
1 {
2   ...
3   "editor.codeActionsOnSave": {
4     "source.fixAll": true,
5   },
6   ...
7 }
```

2.3 Ejemplo de configuración con otros parámetros extra

- settings.json

```
1 {
2   "editor.codeActionsOnSave": {
3     "source.fixAll": true,
4   },
5   "editor.rulers": [
6     99,
7     100,
8   ],
9   // "editor.minimap.enabled": false,
10  "terminal.integrated.tabs.title": "${cwdFolder}",
11 }
```

3 Ejercicios

1. Configurar un proyecto para que detecte los errores de estilo de código.
2. Detectar y arreglar malas prácticas y estilos en el código con Eslint.
3. Ver errores del siguiente código:

```
1 const chalk = require('chalk');
2
```

```
3 var youShouldNeverUseVar = "This is my very long line that eslint  
  should check as an error  
  .....";  
4  
5 function myFunction(used, nonUsed){  
6   if(used){  
7     console.log(used)  
8     return  
9   }  
10 }  
11  
12 module.exports = nonExistingVar;
```

4. Arreglar el código anterior según la guía de estilos de Airbnb

4 Entregables

4.1 En clase

Ejercicios 1, 2, 3 y 4

4.2 Proyecto

Aplicar la configuración de nodemon, ESLint y VSCode al proyecto de notas.