

Proyecto final: Informe código limpio

Sebastian Andrade Cedano
Cristian Steven Motta Ojeda
Juan Esteban Santacruz Corredor

Ingeniería de software I

Universidad Nacional de Colombia

Bogotá D.C.
2025

Linters

La tecnología usada es *ESLint*.

```
import js from '@eslint/js'
import globals from 'globals'
import reactHooks from 'eslint-plugin-react-hooks'
import reactRefresh from 'eslint-plugin-react-refresh'
import tseslint from 'typescript-eslint'

export default tseslint.config(
  { ignores: ['dist'] },
  {
    extends: [js.configs.recommended, ...tseslint.configs.recommended],
    files: ['**/*.ts', '**/*.tsx'],
    languageOptions: {
      ecmaVersion: 2020,
      globals: globals.browser,
    },
    plugins: {
      'react-hooks': reactHooks,
      'react-refresh': reactRefresh,
    },
    rules: {
      ...reactHooks.configs.recommended.rules,
      'react-refresh/only-export-components': [
        'warn',
        { allowConstantExport: true },
      ],
    },
  },
)
```

Archivo eslint.config.js

En este archivo se especifican las configuraciones para nuestro linters en términos generales se especifica lo siguiente:

- Habilita reglas para TypeScript y JavaScript.
- Aplica reglas recomendadas para React y Hooks.
- Ignora la carpeta dist para evitar revisar archivos compilados.
- Define un entorno de navegador.
- Activa reglas para evitar problemas con Fast Refresh en React.

Adicionalmente muestra diferentes errores de nuestro código que no afectan su ejecución pero si están relacionados con el clean code.

```
import calendar from "
import lightbulb from "
function UserStats() {
  const [solvedProblem
  const [solvedLastMon
  const [submissions, setSubmissions] = useState<number>(245);
  return (
    'setSubmissions' is assigned a value but never used. eslint(@typescript-eslint/no-unused-vars)
    'setSubmissions' is declared but its value is never read. ts(6133)
    const setSubmissions: React.Dispatch<React.SetStateAction<number>>
    View Problem (Alt+F8) Quick Fix... (Ctrl+.)
```

Error en un estado que nunca es utilizado (React)

```
interface SecondLevelMenuProps {
  options: Arr
  labels: Arra
  selected: st
  select: (o: any) => any; // Setter function for change the option (useState setter funciton)
};
```

Problema con el tipado de una interface (Ts)

```
cons 'buffer' is never reassigned. Use 'const' instead. eslint(prefer-const)
}
  Let buffer: any[]
  View Problem (Alt+F8) Quick Fix... (Ctrl+.)
Let buffer = [];
```

Malas prácticas al declarar constantes como variables (Ts / Js)

Como se muestra en las imágenes, los errores no están relacionados con problemas de lógica o sintaxis, sino con malas prácticas detectadas por ESLint. Aunque no afectan la ejecución del código, pueden dificultar el desarrollo del proyecto.

Narraciones sobre el código del equipo

1. *Sebastian Andrade Cedano*: Hubo algunos inconvenientes con la base de datos; sin embargo, no representaron un problema mayor, ya que, gracias a la comunicación constante del equipo, se solucionaron fácilmente. Actualmente, cada miembro cumple con sus tareas asignadas, pero todos estamos involucrados en el desarrollo del proyecto, por lo que no hay problema si se tuviera que modificar el código.
2. *Cristian Steven Motta Ojeda*: El código ha sido fácil de trabajar y leer. Al inicio, distribuimos las tareas según las fortalezas y debilidades de cada miembro. Sin embargo, para salir de nuestra zona de confort, asumimos algunas tareas fuera de nuestras especialidades. Además, mantenemos una comunicación constante y sabemos exactamente en qué está trabajando cada integrante del equipo.
3. *Juan Esteban Santacruz Corredor*: Aunque faltan comentarios, el código sigue buenas prácticas para el nombramiento de variables y modularización de la lógica por lo que es intuitivo. Todos nos distribuimos tareas al inicio y sabemos en qué parte del avance está cada uno. Hasta el momento aunque se han presentado dificultades hemos podido lograr los plazos acordados.