



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Testing Linter

Integrantes:

David Camilo Cortes Salazar

Juan Manuel Espitia Pizza

Andrew Nicolay Prieto Mendoza

Sebastian Steeven Rodriguez Ortiz

Yeswah Gonzalez Tapia

Docente

Oscar Eduardo Alvarez Rodriguez

Universidad Nacional de Colombia
Sede Bogotá

Bogotá, Colombia
16 de Noviembre 2025

Herramienta Utilizada: ESLint

Se optó por escoger ESLint debido a que es una de las herramientas más sólidas y extendidas en el ecosistema JavaScript. Esta tiene una gran flexibilidad, ya que permite adaptarlo a las necesidades específicas de una aplicación como la nuestra que combina Electron y React, permitiendo que todo el código siga un mismo estándar de calidad. Además, la disponibilidad de configuraciones predefinidas facilitan mucho su implementación, ayudando a detectar errores rápidamente y mantener un estilo de código consistente en todo el proyecto.

Configuración Aplicada:

ESLint permite configurar diferentes reglas desde un archivo [“eslint.config.js”](#) ubicado en la raíz del proyecto. A continuación las explicamos a detalle:

- Imports:

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

Estos nos ayudan a traer de manera predefinida ciertas configuraciones recomendadas, variables globales y reglas para frameworks específicos como lo es React en nuestro caso.

- Directorios ignorados:

```
JavaScript
{ ignores: ['dist', 'dist-react'] }
```

Estas carpetas contienen archivos compilados que no son parte del código desarrollado por nosotros, sino de la configuración de las dependencias que se están utilizando.

- Archivos Objetivo:

```
JavaScript
{ files: ['**/*.{js,jsx}'],
```

Podemos especificar que tipo de archivos deben revisarse. En nuestro caso, serán los js y jsx, pues estamos trabajando con React y Vite.

- Opciones del lenguaje:

```
JavaScript
languageOptions: {
  ecmaVersion: 2020,
  globals: {
    ...globals.browser,
    ...globals.node
  },
  parserOptions: {
    ecmaVersion: 'latest',
    ecmaFeatures: { jsx: true },
    sourceType: 'module',
  },
},
```

En este caso, la mayoría son configuraciones para que ESLint trabaje de manera estándar. Es importante resaltar que aquí estamos diciéndole que las variables globales tanto del navegador como de Node no deben ser evaluadas.

- Plugins:

```
JavaScript
plugins: {
  'react-hooks': reactHooks,
  'react-refresh': reactRefresh,
},
```

Los plugins nos ayudan a que ESLint entienda los frameworks utilizados en el desarrollo del proyecto, ayudando a que no realice revisión sobre las variables relacionadas con los hooks de React ni al código relacionado con el fast refresh de Vite.

- Reglas Personalizadas:

```
JavaScript
rules: {
  ...js.configs.recommended.rules,
  ...reactHooks.configs.recommended.rules,
  'no-unused-vars': ['error', {
```

```
varsIgnorePattern: '^[A-Z_]'\n  }],\n\n  'react-refresh/only-export-components': [\n    'warn',\n    { allowConstantExport: true }\n  ],\n}\n}
```

Las reglas son parte vital del funcionamiento de ESLint. Para nuestro caso, al no tener el desarrollo completo, muchas veces tenemos secciones de código que, aunque no se han implementado del todo, existen para permitir probar su integridad con otras funcionalidades. Aquí es donde las reglas personalizadas nos ayudan, pues podemos asignar patrones en nombres para que se detecte que estas aún no deben analizarse.

Resultados Obtenidos:

En un principio se determinó corregir los errores detectados por ESLint que estaban dentro de la carpeta “./src.” Esto ya que es donde más código se había realizado. De aquí el linter determinó 12 errores constituidos por los siguientes tipos:

- Variables no definidas
- Variables definidas y no usadas

Luego se revisó error por error para arreglar estos problemas, arreglando algunos errores en código y señalando a ESLinter que algunas variables no se usaban aún porque no se había empezado el desarrollo del componente que las utiliza.

Al terminar esta tarea, se inició con el análisis completo del proyecto. Se detectaron 117 errores, todos de los mismos tipos que en el análisis anterior. Esto se debía a una mala configuración del “[eslint.config.js](#)”, pues no se especificó el ignorar la carpeta “./dist-react” ni el uso de las variables globales de Node. Cabe resaltar que esta carpeta generaba 98 de los errores detectados.

En total, el proyecto corregido solo quedó con una advertencia de ESLint referente a Vite. Como evidencia de lo anterior, se dejan los reportes generados por ESLint en cada uno de los pasos descritos.

- **Evidencia de su Ejecución:**

Análisis de ./src:

```

USUARIO@DESKTOP-DCV4M61 MINGW64 /d/GitHub/no_informado/Project (ui)
$ npx eslint src

D:\GitHub\no_informado\Project\src\backend\repositories\CajaRepository.js
   6:11  error  'movement' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u
        no-unused-vars
  38:11  error  'closing' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u
        no-unused-vars
  77:11  error  'paymentMethod' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u
        no-unused-vars
  88:11  error  'paymentMethod' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u
        no-unused-vars

D:\GitHub\no_informado\Project\src\backend\services\ClosingService.js
  83:10  error  Parsing error: Export 'isCajero' is not defined

D:\GitHub\no_informado\Project\src\backend\services\MovementService.js
  63:69  error  Parsing error: Export 'movement' is not defined

D:\GitHub\no_informado\Project\src\renderer\components\auth\AuthContext.jsx
  20:18  error  'error' is defined but never used
        no-unused-vars
  61:18  error  'error' is defined but never used
        no-unused-vars
  86:14  warning  Fast refresh only works when a file only exports components. Use a new file to share constants or functions between components

D:\GitHub\no_informado\Project\src\renderer\components\auth\Login.jsx
  23:18  error  'err' is defined but never used
        no-unused-vars

D:\GitHub\no_informado\Project\src\renderer\components\auth\UserRegister.jsx
  84:18  error  'err' is defined but never used
        no-unused-vars

D:\GitHub\no_informado\Project\src\renderer\components\layout\Dashboard.jsx
   1:10  error  'useState' is defined but never used. Allowed unused vars must match /^[A-Z_]/u
        no-unused-vars

✖ 12 problems (11 errors, 1 warning)

```

Análisis del proyecto completo:

```

USUARIO@DESKTOP-DCVAM65: /d/GitHub/no_informado/Project (u1)
$ npx eslint .
49:37226 error '..._REACT_DEVTOOLS_GLOBAL_HOOK...' is not defined no-undef
49:37283 error '..._REACT_DEVTOOLS_GLOBAL_HOOK...' is not defined no-undef
49:37614 error Empty block statement no-empty
174:8562 error 'k' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
174:8646 error 'cl' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
174:8723 error 'Dl' is defined but never used no-unused-vars

D:\GitHub\no_informado\Project\electron\ipc\MovimientosHandler.js
71:9 error 'crea' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
76:17 error 'payload' is not defined no-undef

D:\GitHub\no_informado\Project\electron\main.js
53:7 error 'process' is not defined no-undef

D:\GitHub\no_informado\Project\prisma\seed.js
10:9 error 'defaultRoles' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
72:3 error 'process' is not defined no-undef

D:\GitHub\no_informado\Project\src\renderer\components\auth\AuthContext.jsx
86:14 warning Fast refresh only works when a file only exports components. Use a new file to share constants or functions between components react-refresh/only-export-components
10:9 error 'defaultRoles' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
72:3 error 'process' is not defined no-undef

D:\GitHub\no_informado\Project\src\renderer\components\auth\AuthContext.jsx
86:14 warning Fast refresh only works when a file only exports components. Use a new file to share constants or functions between componen 10:9 error 'defaultRoles' is assigned a v
value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
72:3 error 'process' is not defined no-undef
10:9 error 'defaultRoles' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
72:3 error 'process' is not defined no-undef
10:9 error 'defaultRoles' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
10:9 error 'defaultRoles' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
72:3 error 'process' is not defined no-undef
10:9 error 'defaultRoles' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
10:9 error 'defaultRoles' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
10:9 error 'defaultRoles' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars
72:3 error 'process' is not defined no-undef

D:\GitHub\no_informado\Project\src\renderer\components\auth\AuthContext.jsx
86:14 warning Fast refresh only works when a file only exports components. Use a new file to share constants or functions between components react-refresh/only-export-components

D:\GitHub\no_informado\Project\tests\TestBasededatos.js
8:9 error 'crea' is assigned a value but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars

D:\GitHub\no_informado\Project\tests\movementsHandler.test.js
2:19 error 'doPaymentMethodExist' is defined but never used. Allowed unused vars must match /^[A-Z_]/u no-unused-vars

X118 problems (117 errors, 1 warning)

```

- Evidencia de su Corrección:

```
USUARIO@DESKTOP-DCVAM61 MINGW64 /d/GitHub/no_informado/Project (ui)
$ npx eslint .

D:\GitHub\no_informado\Project\src\renderer\components\auth\AuthContext.jsx
  86:14  warning  Fast refresh only works when a file only exports components. Use a new file to share c
onstants or functions between components  react-refresh/only-export-components

✖ 1 problem (0 errors, 1 warning)
```

USUARIO@DESKTOP-DCVAM61 MINGW64 /d/GitHub/no_informado/Project (ui)