SOLUCIÓN DE ERRORES EN TYPE SCRIPT REGLAS ESLINT

OSCAR DAVID RUIZ SAEZ

Brayan Arcos

Programación Backend

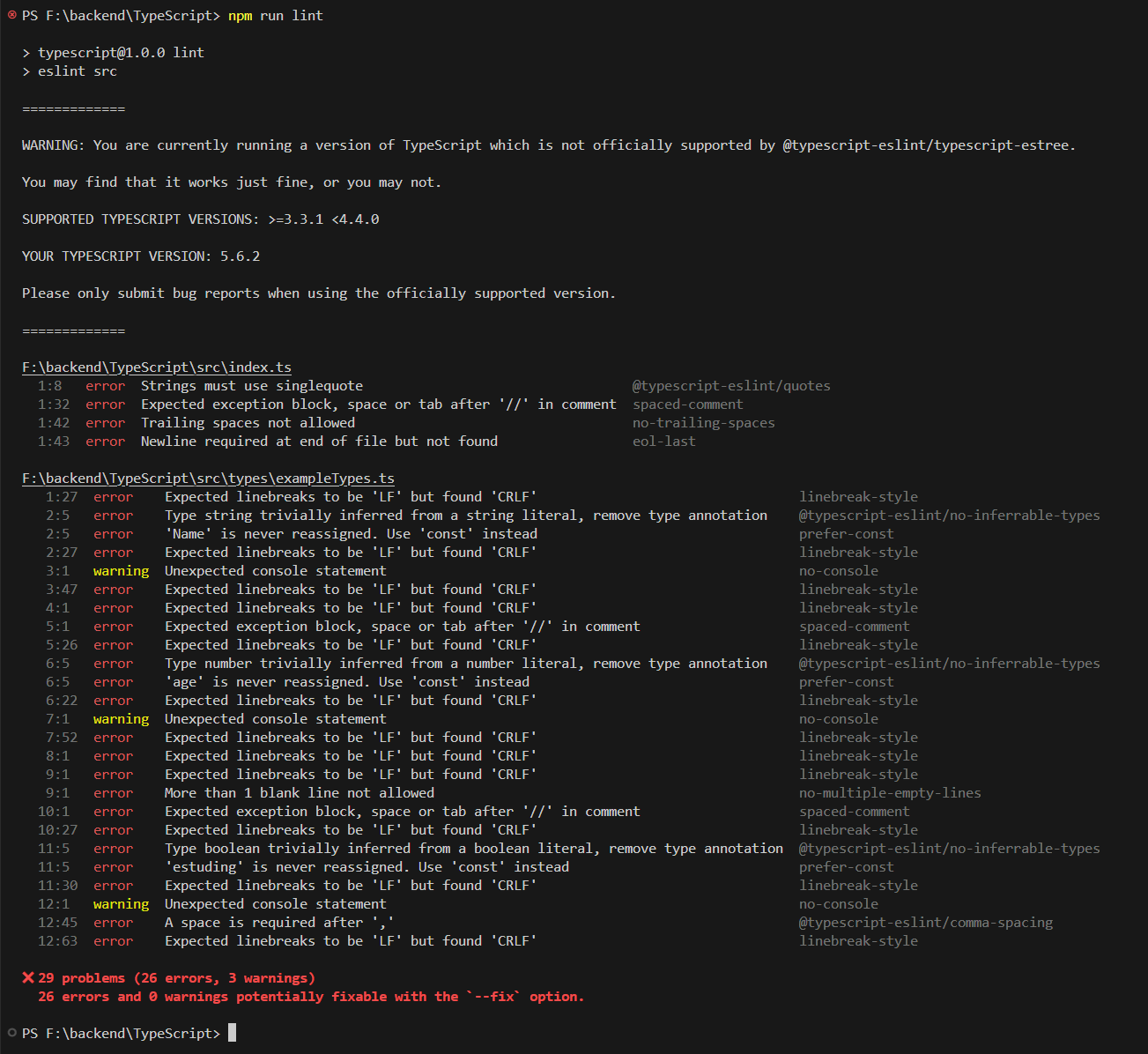
Instituto Tecnológico del Putumayo

Tecnología en Desarrollo de Software

Mocoa-Putumayo

2024

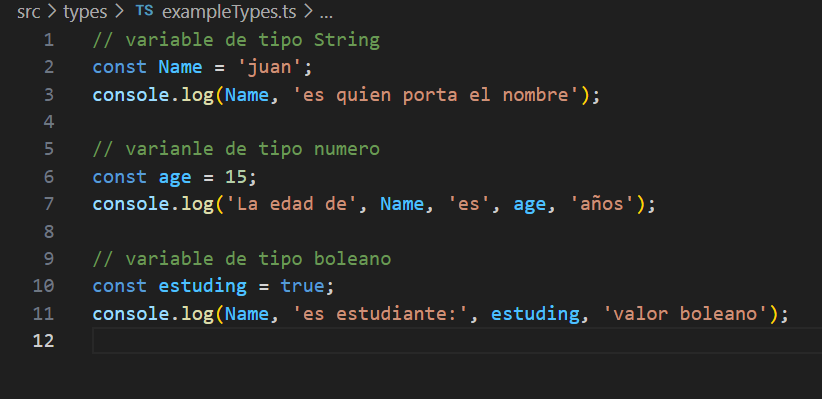
Para empezar nuestra corrección de errores primero debemos identificar los errores que tenemos, para esto nos vamos a la terminal y escribimos npm run lint y obtendremos los errores



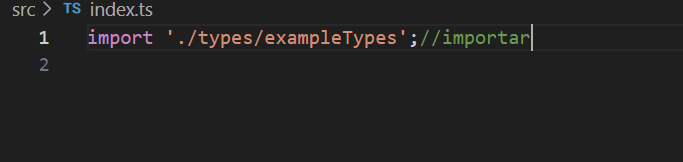
Aquí identificamos 26 errores de los cuales son 10 los cuales se suelen repetir lo cuales son:

1. **Strings must use singlequote**:
   * **Qué significa**: Debes usar comillas simples (') en lugar de comillas dobles (").
   * **Por qué importa**: Mantener un estilo consistente en tu código hace que sea más fácil de leer y mantener.
2. **Expected exception block, space or tab after ‘//’ in comment :**
   * **Qué significa**: Necesitas poner un espacio después de // en los comentarios.
   * **Por qué importa**: Esto mejora la legibilidad del código, haciendo que los comentarios sean más fáciles de leer.
3. **Trailing spaces not allowed :**
   * **Qué significa**: No se permiten espacios al final de las líneas.
   * **Por qué importa**: Los espacios finales son innecesarios y pueden causar problemas en el control de versiones y en la limpieza del código.
4. **Newline required at end of file but not found:**
   * **Qué significa**: Cada archivo debe terminar con una nueva línea.
   * **Por qué importa**: Algunos sistemas de control de versiones y herramientas de desarrollo requieren una nueva línea al final del archivo para funcionar correctamente.
5. **Expected linebreaks to be ‘LF’ but found ‘CRLF’ :**
   * **Qué significa**: Se espera que los saltos de línea sean LF (Unix) en lugar de CRLF (Windows).
   * **Por qué importa**: Mantener un estilo de salto de línea consistente es importante para la colaboración en proyectos, especialmente cuando se trabaja en diferentes sistemas operativos.
6. **Type string trivially inferred from a string literal, remove type**:
   * **Qué significa**: No es necesario especificar el tipo cuando se puede inferir del valor.
   * **Por qué importa**: Las anotaciones de tipo triviales son redundantes y hacen que el código sea más largo y menos legible.
7. **‘Name’ is never reassigned. Use ‘const’ instead :**
   * **Qué significa**: Usa const en lugar de let cuando la variable no se reasigna.
   * **Por qué importa**: Usar const para variables que no cambian mejora la seguridad y claridad del código, indicando que la variable es inmutable.
8. **A space is required after ‘,’** 
   * **Qué significa**: Debe haber un espacio después de cada coma.
   * **Por qué importa**: Esto mejora la legibilidad del código, asegurando que los elementos de una lista o argumentos de una función sean fáciles de leer.
9. **More than 1 blank line not allowed:**
   * **Qué significa**: No se permiten más de una línea en blanco consecutiva.
   * **Por qué importa**: Las múltiples líneas en blanco pueden hacer que el código sea más difícil de leer y navegar, y no aportan valor al código.

Así lucia nuestro código antes de ser reparado:

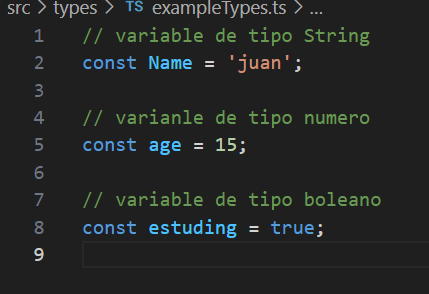
exampleTypes.ts  


index.ts

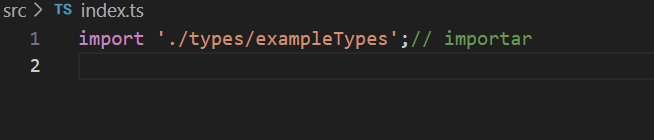


Y asi luce después de ser reparado:

exampleTypes.ts



Index.ts



Para verificar nuestra terminal usamos nuevamente npm run lint:

