CORRECCIÓN DE ERRORES: ESLINT

YESI ESTEBAN PANTOJA CUELLAR

DOCENTE

BRAYAN IGNACIO ARCOS BURBANO

INSTITUTO TECNOLÓGICO DEL PUTUMAYO

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

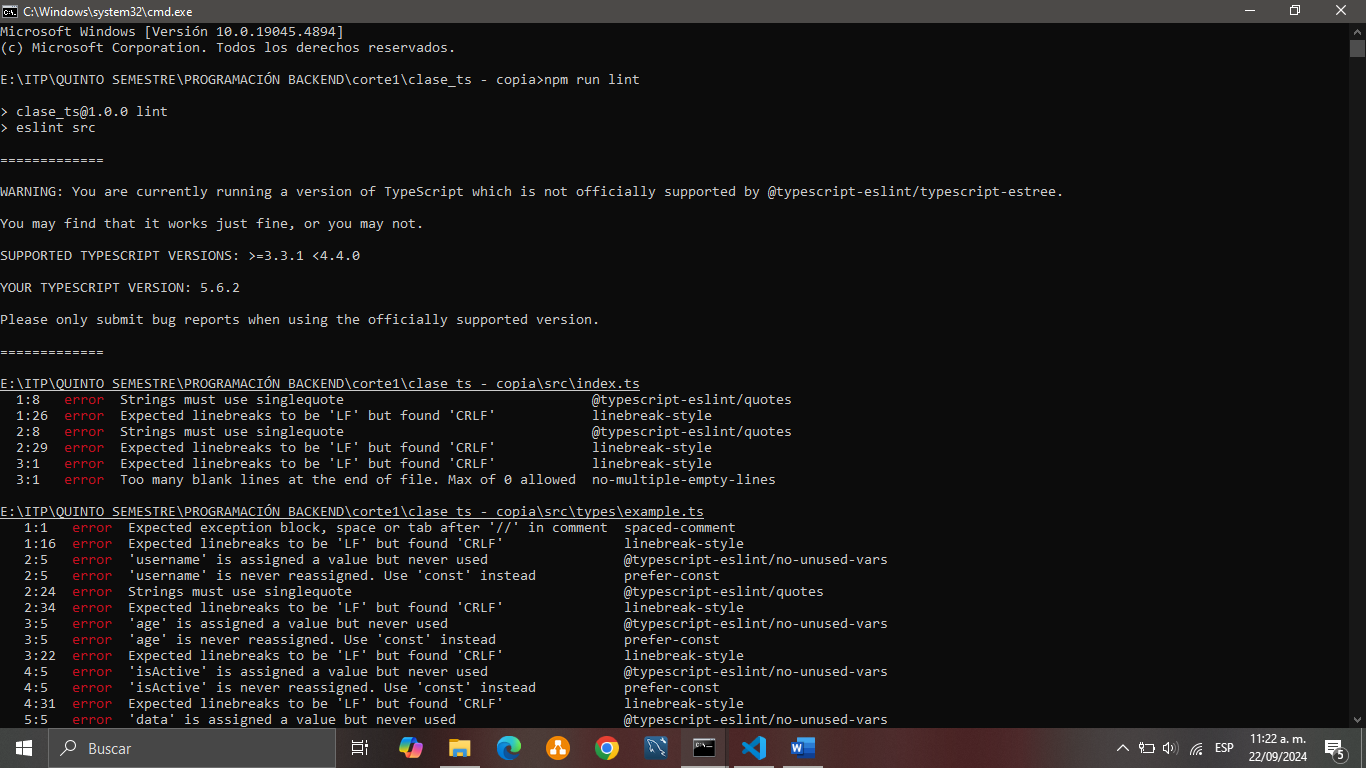
QUINTO SEMESTRE

PROGRAMACIÓN BACKEND

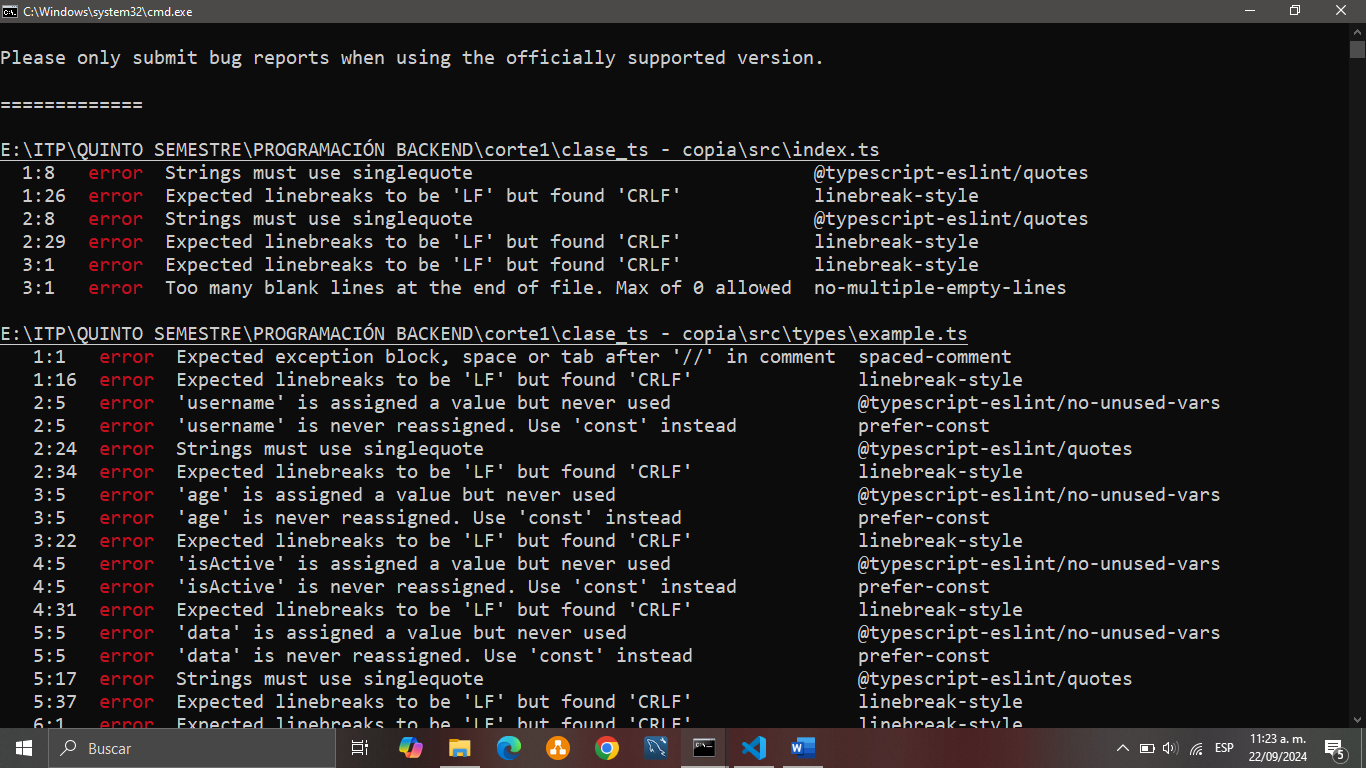
MOCOA - PUTUMAYO

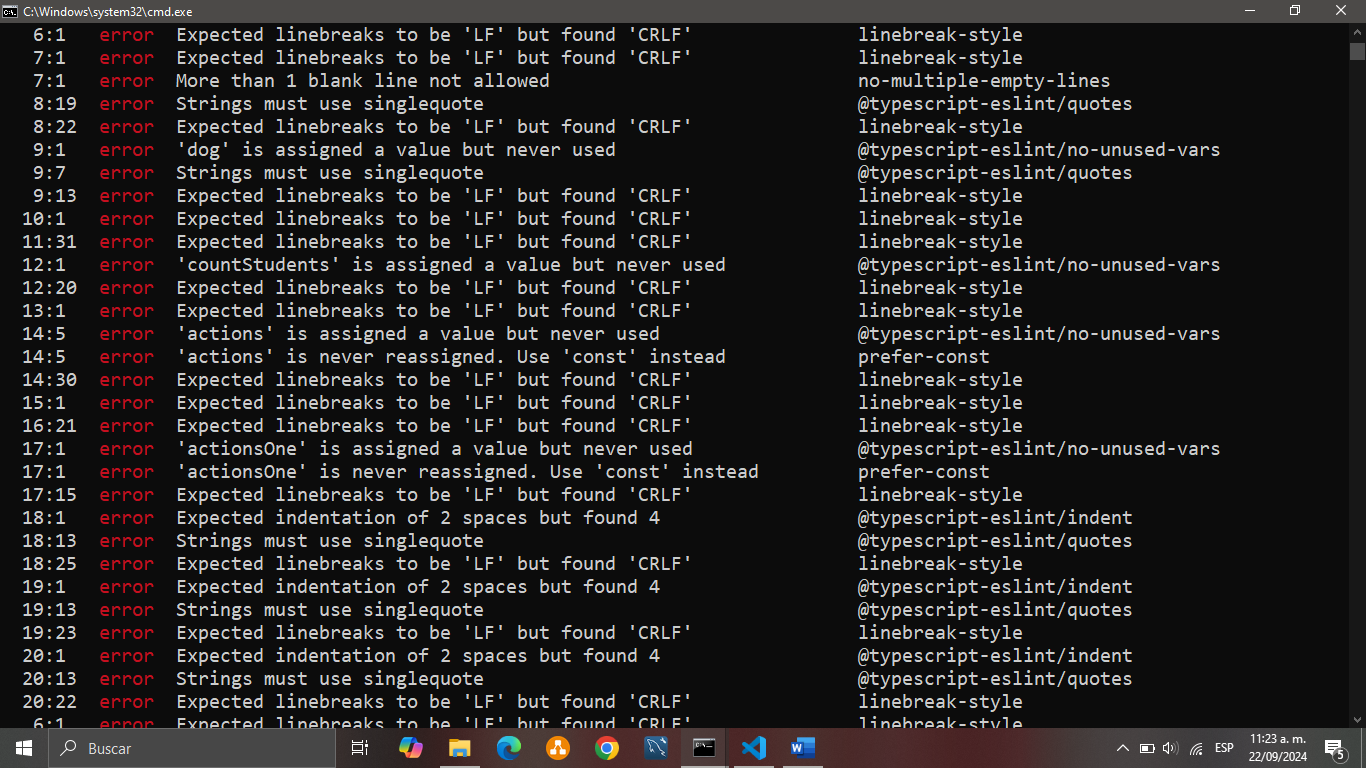
2023

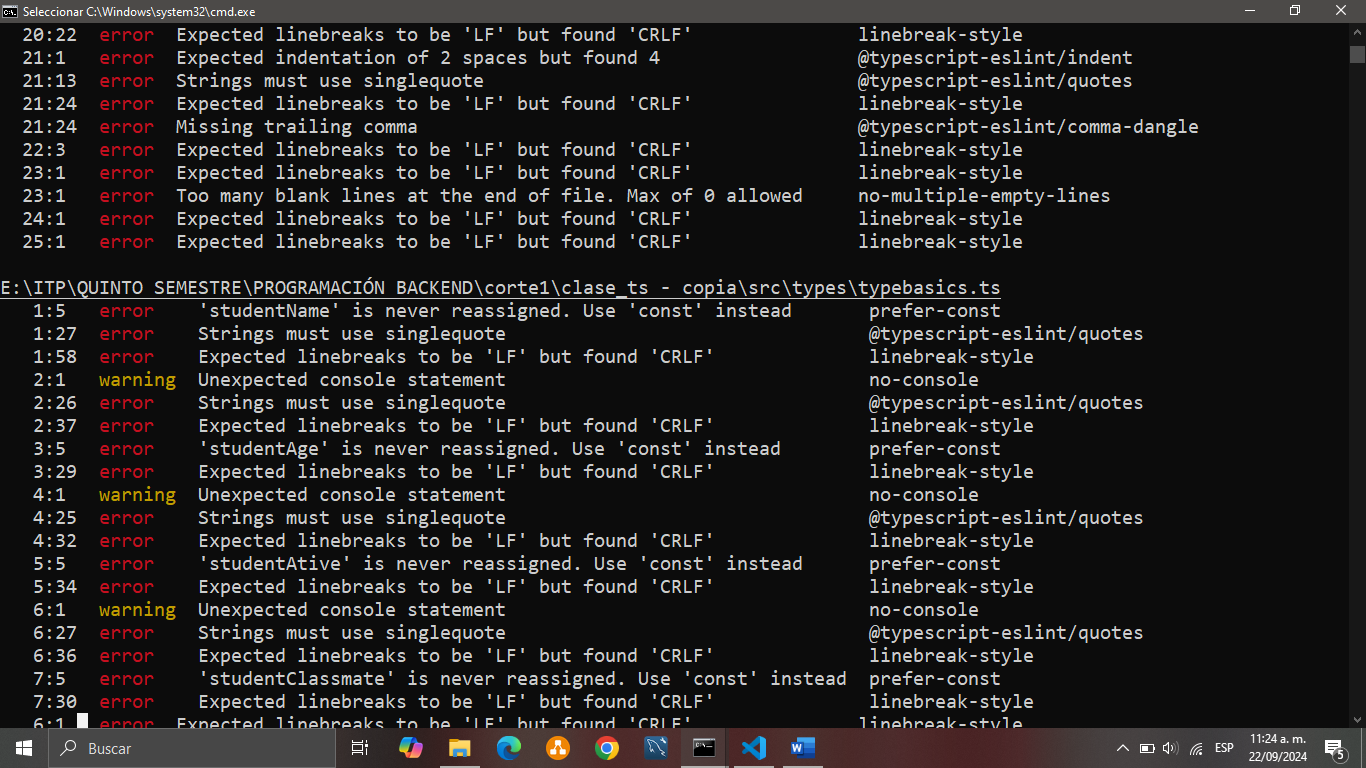
Para empezar con la corrección de los errores propuestos por eslint, primero debemos utilizar el comando “npm run lint” en nuestra terminal, de la siguiente manera:

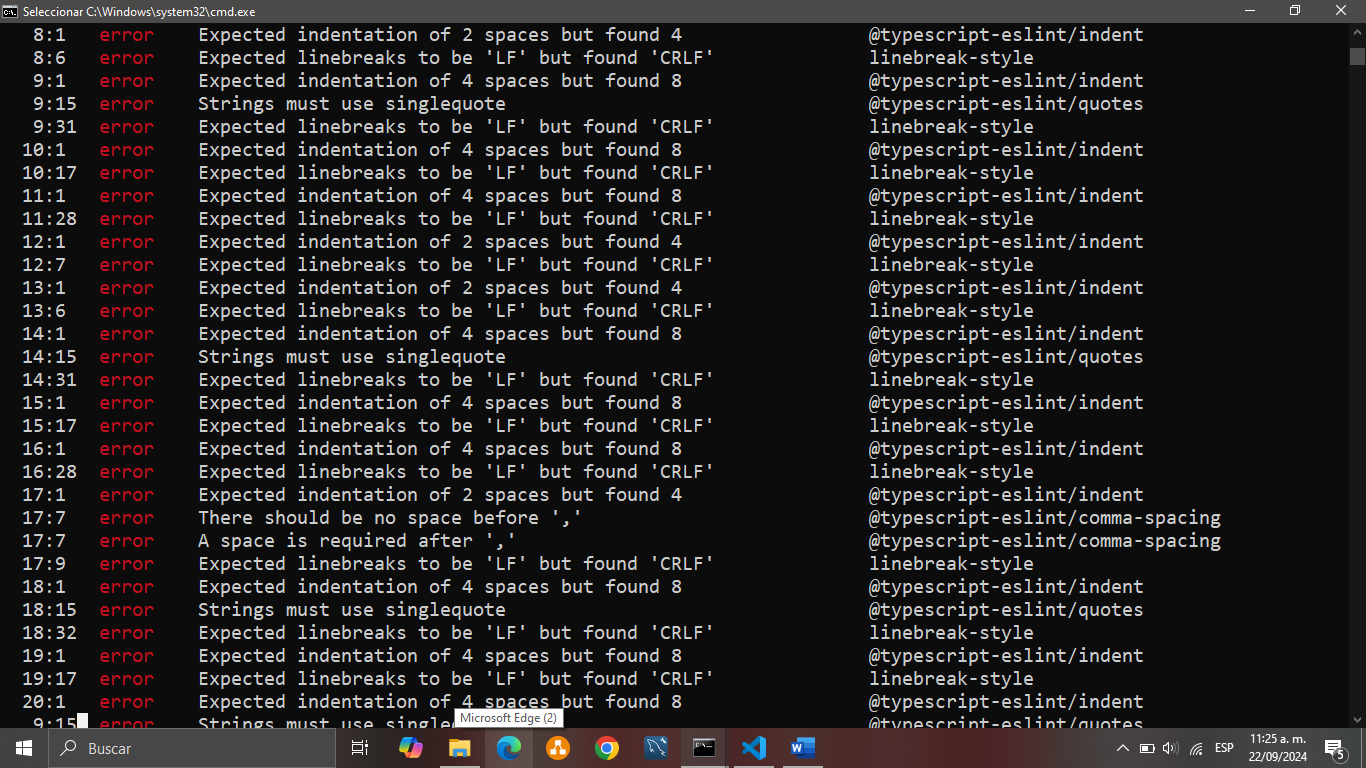


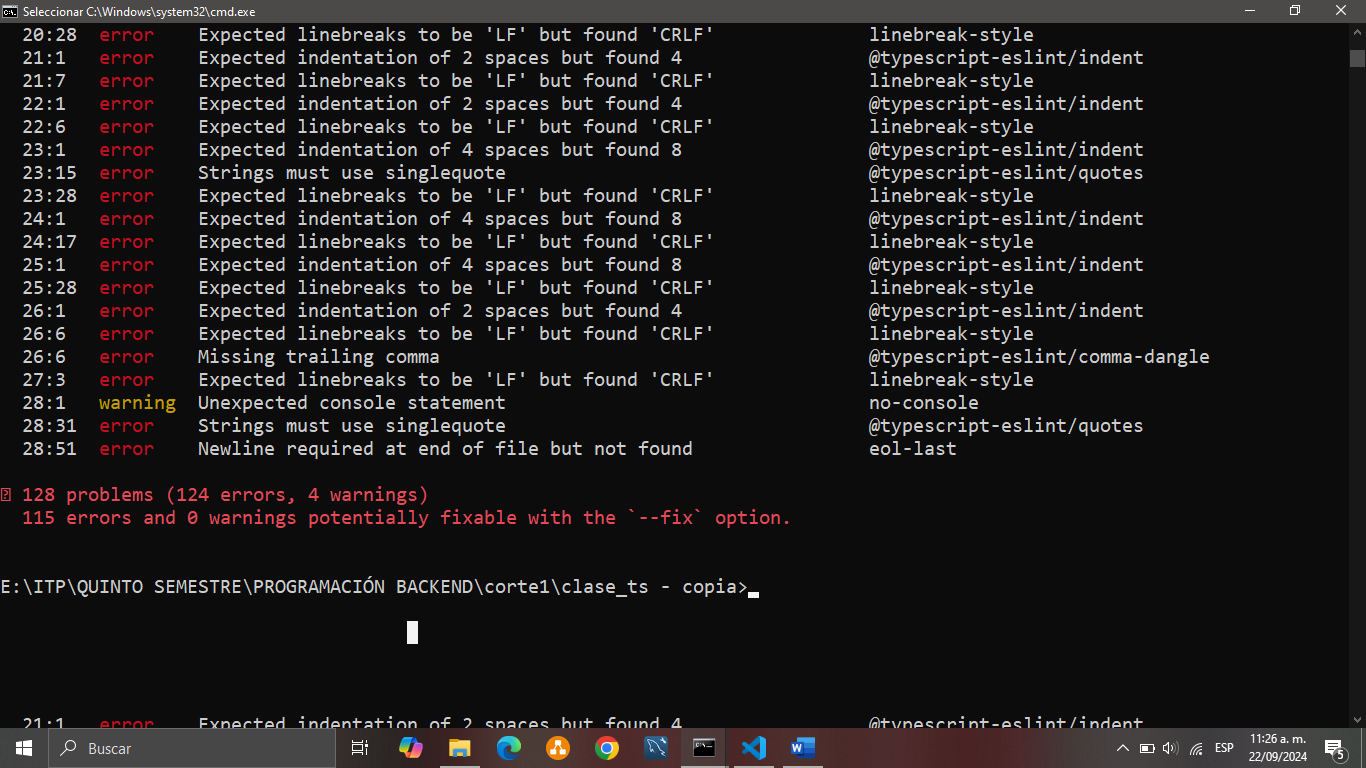
con este podremos visualizar que errores podremos tener en nuestro proyecto TypeScript.











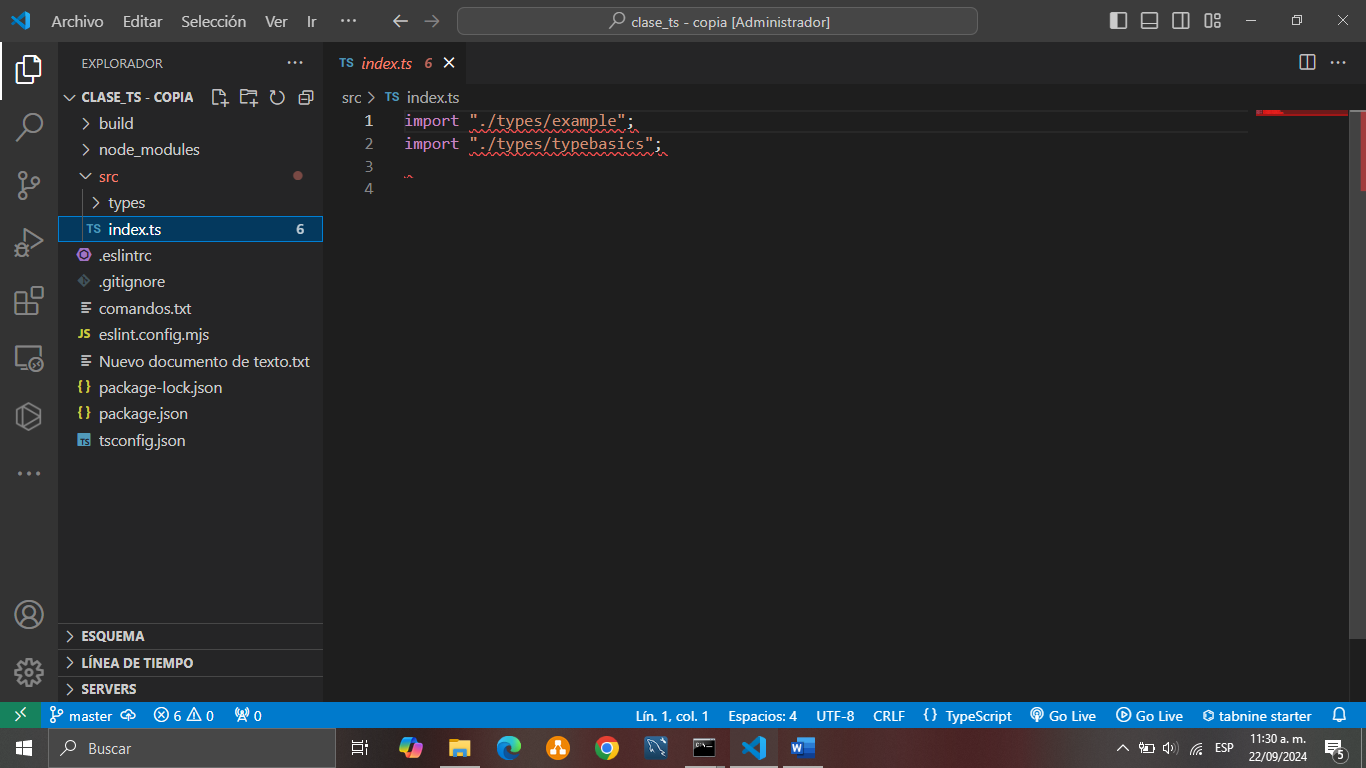
Estos son los errores que corregiremos en este documente.

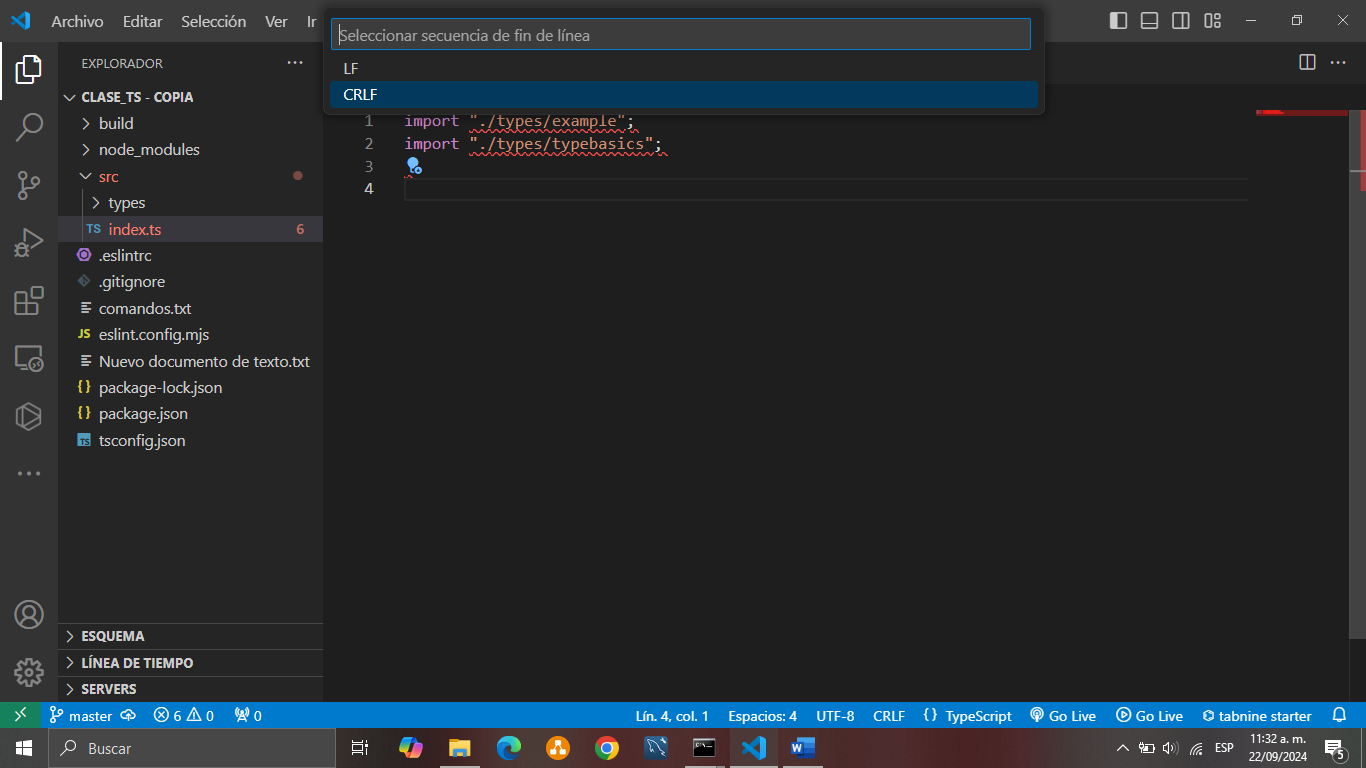
Pasos.

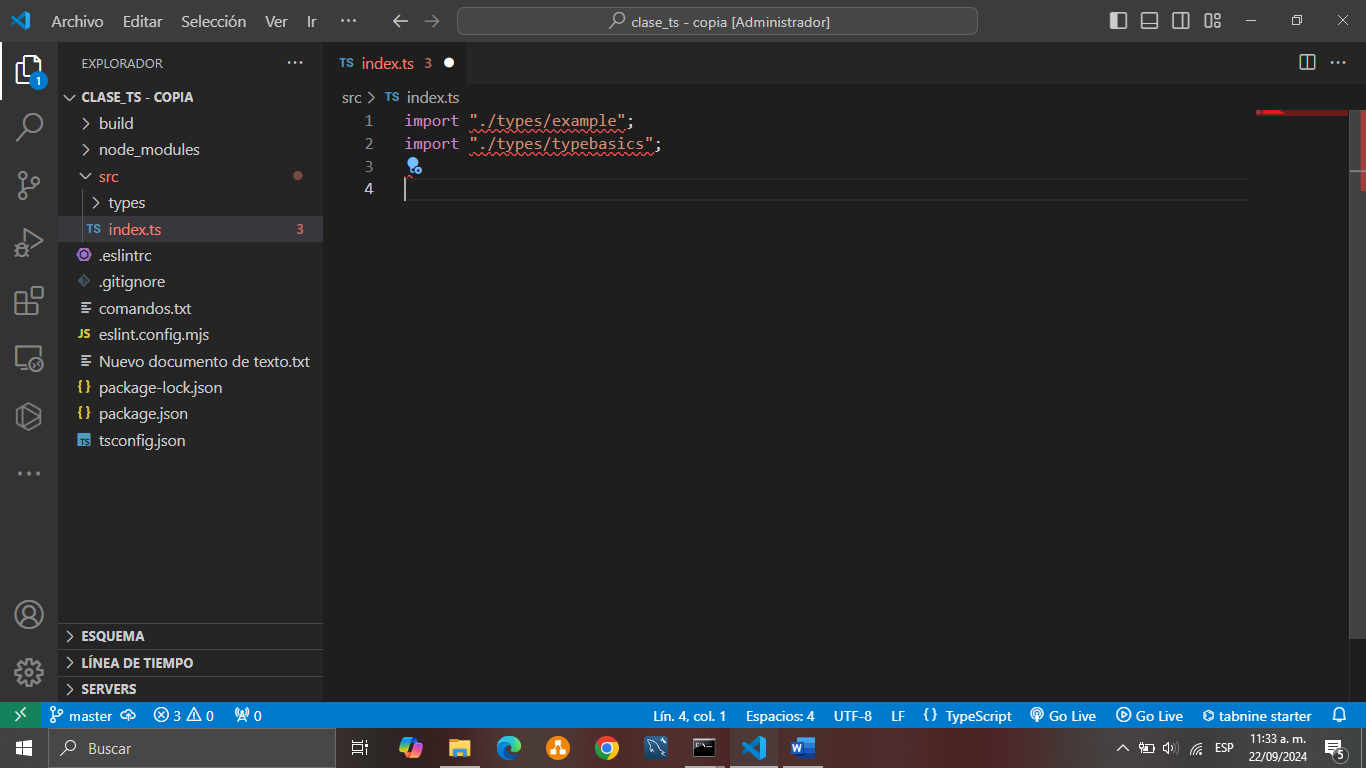
1. Visualizar el error más común

error Expected linebreaks to be 'LF' but found 'CRLF'

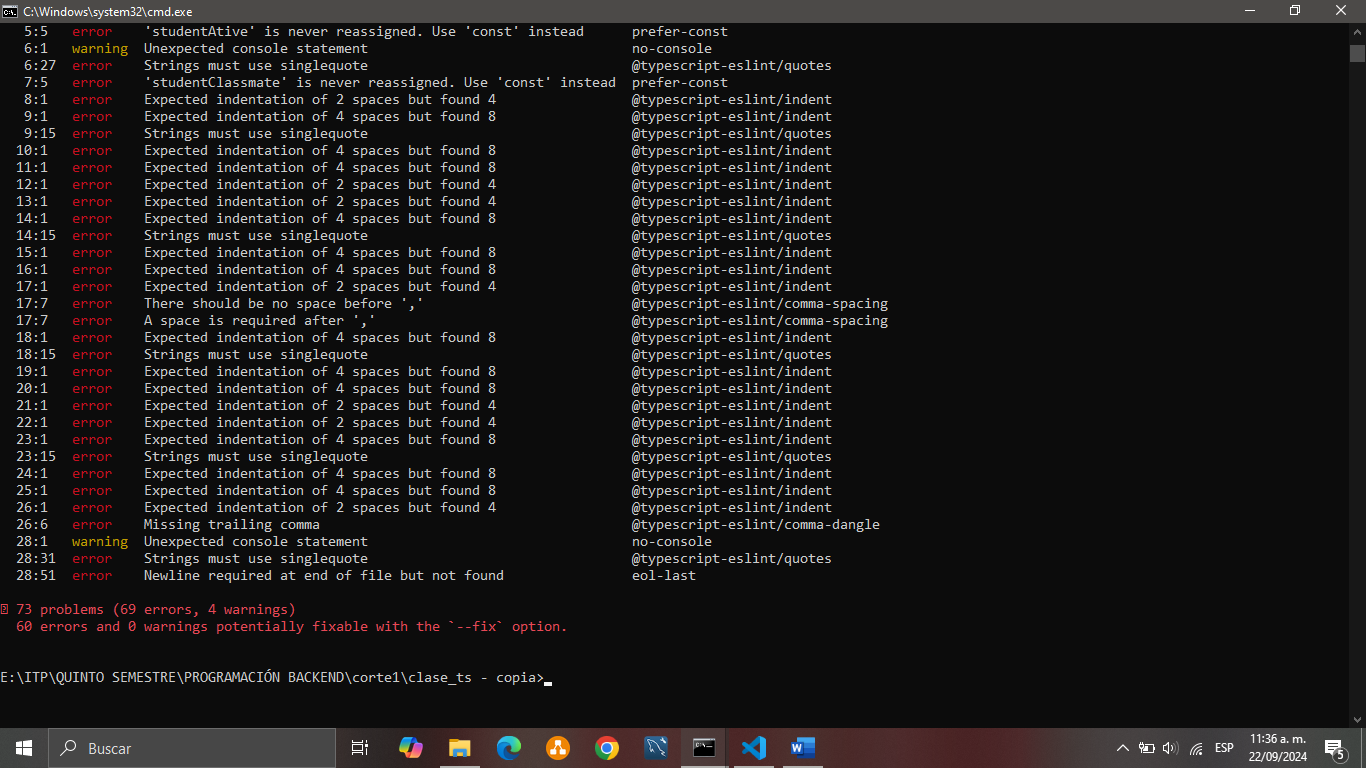
Empezaremos corrigiendo este error debido a que se encuentra en varios campos repetidos, para eso vamos a nuestro Visual Studio Code, abrimos nuestros archivos TS y nos dirigimos a la parte inferior dónde se encuentra nuestra barra, allí debemos ubicar la palabra “CRLF” damos click y cambiamos a LF, de la siguiente manera:







Cómo podremos ver ya hemos cambiado esa parte, esto tenemos que hacerlo en cada archivo .TS.



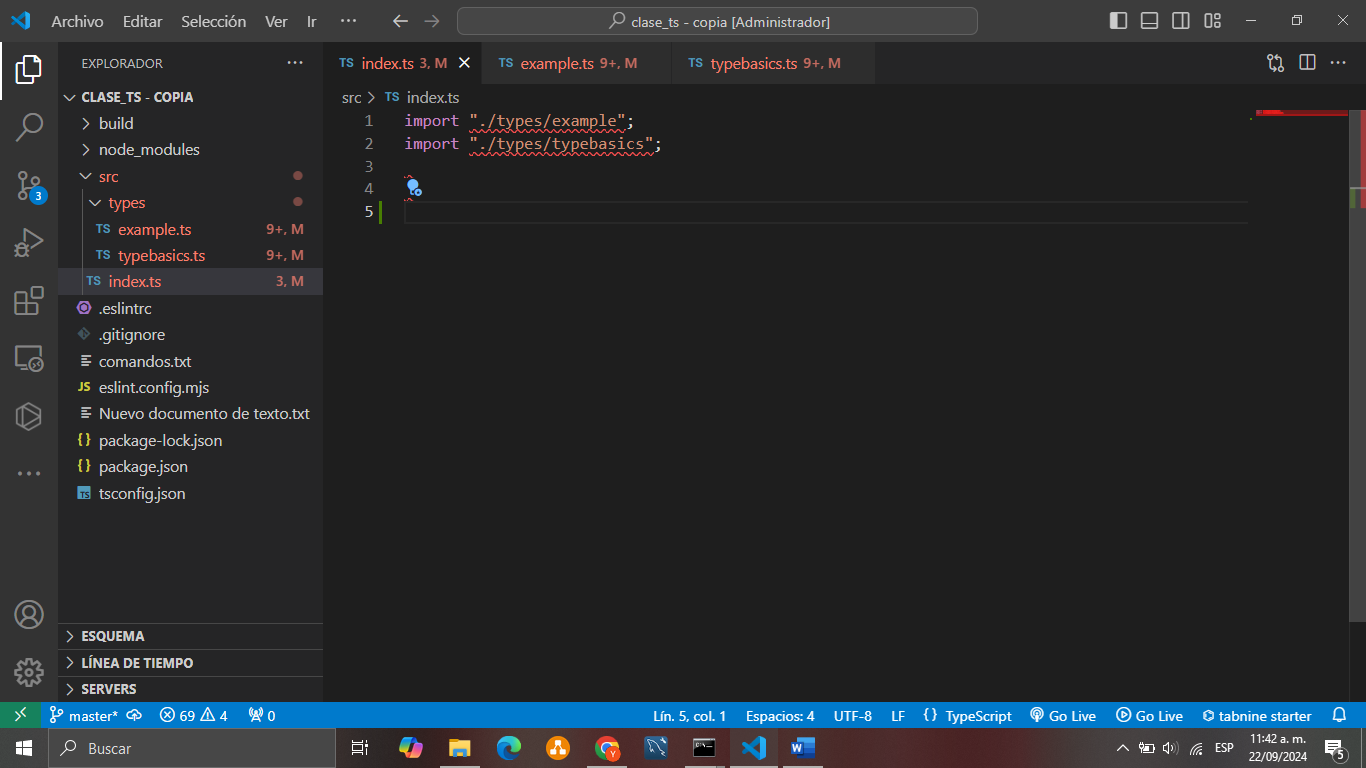
Verificamos y ya hemos solucionado casi la mitad de los errores.

1. Continuamos por esa línea de encontrar errores comunes

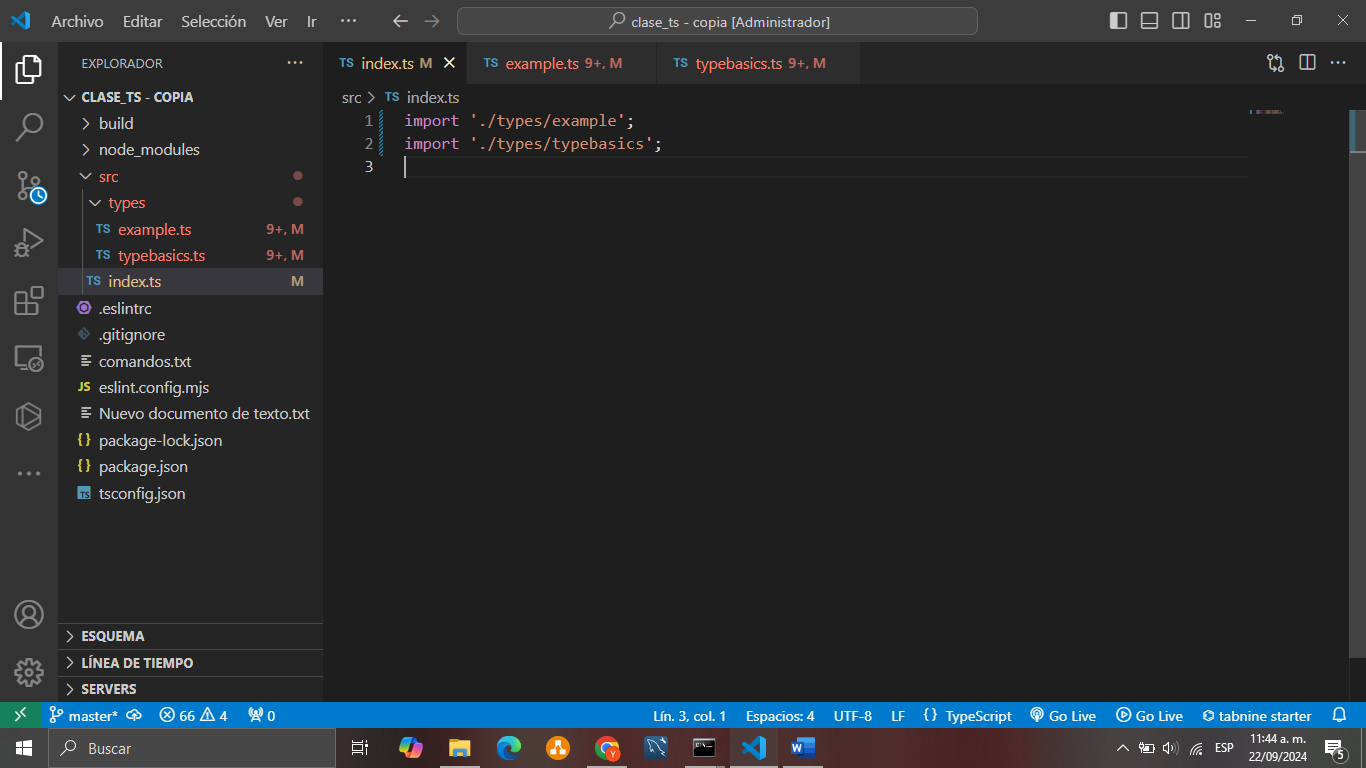
error Strings must use singlequote

En este error lo que debemos es cambiar las comillas dobles (“ “) en nuestros archivos .TS por comillas simples (‘ ’).

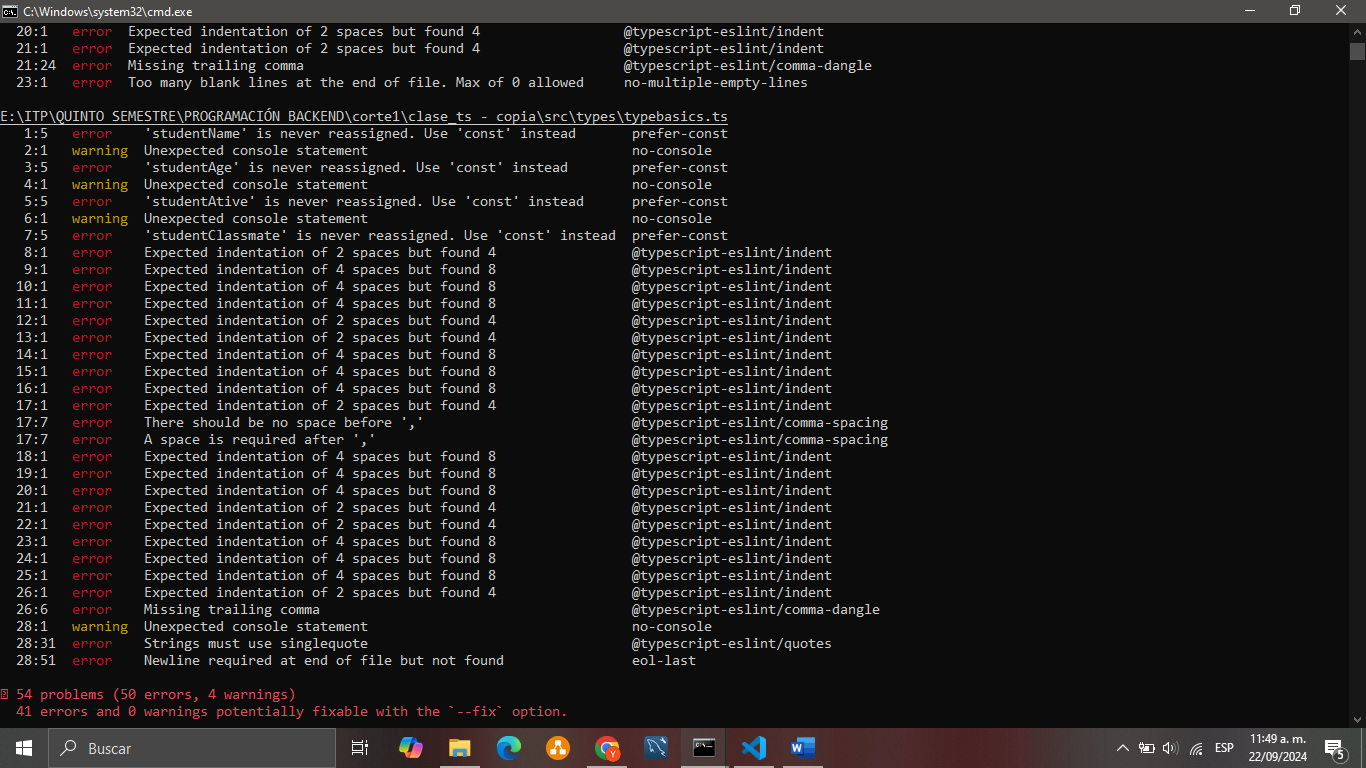
1. Tenemos este caso, en el que usamos comillas dobles (“ “) y nos marca error



1. Entonces hacemos el cambio y queda de esta manera



Cómo podemos ver ya no tenemos subrayado en el rojo nuestro archivo, eso quiere decir que hemos hechos los cambios requeridos por nuestro eslint. Hay que hacer esto para cada archivo .TS en los que tengamos este error de comillas.

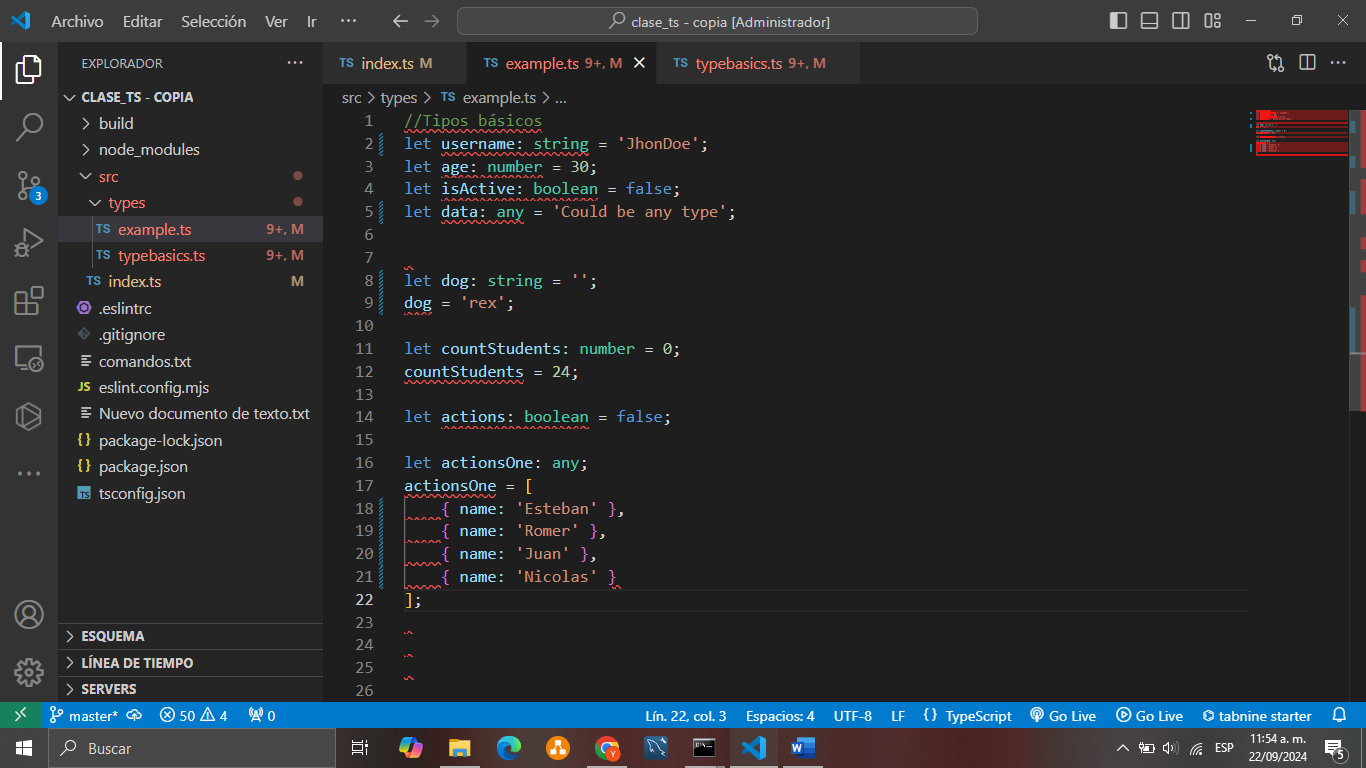


Y bueno, seguimos disminuyendo nuestros errores.

1. Ahora vamos a corregir errores específicos

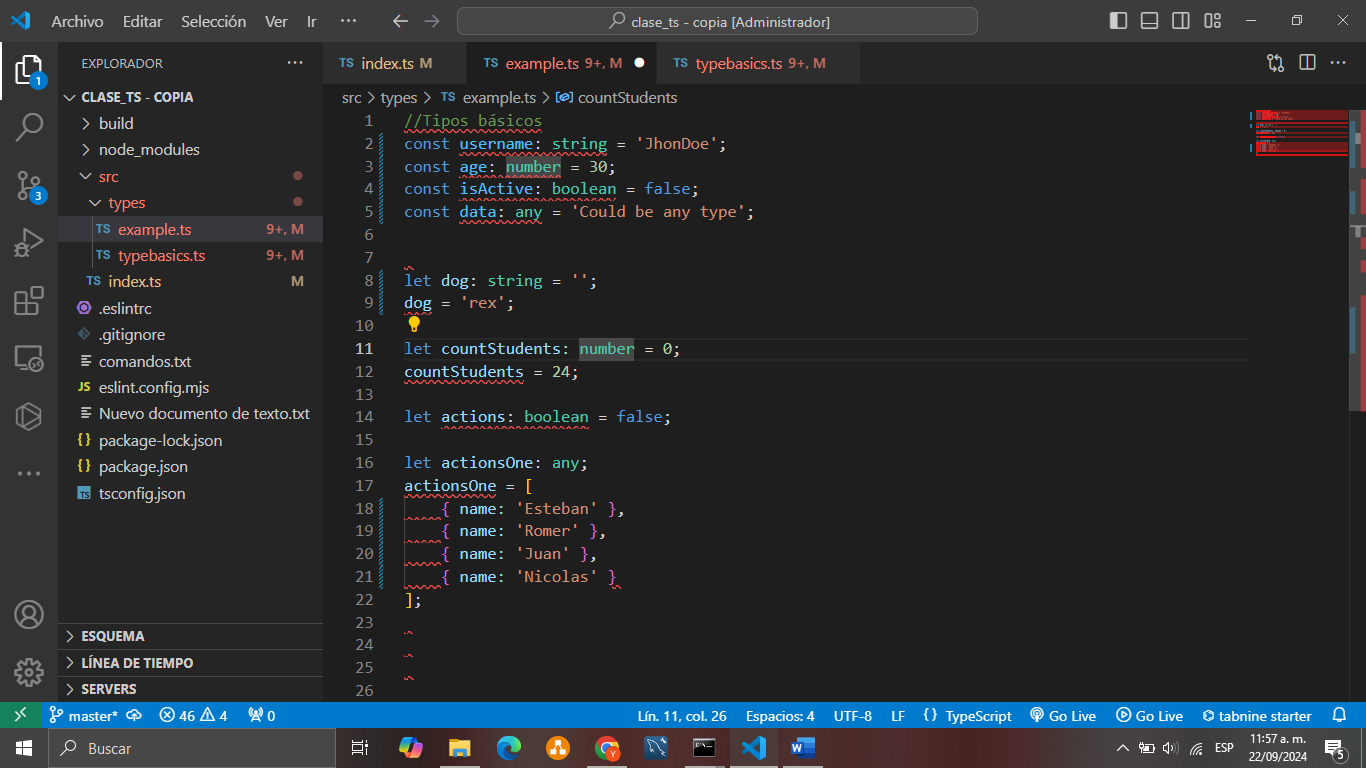
error 'username' is never reassigned. Use 'const' instead

Bueno este error nos dice qué debemos reasignar nuestra declaración de la variable por const debido a qué no estamos haciendo un llamado de esa variable en otra parte, esto significa que es una variable constante y siempre va a mantener su valor.

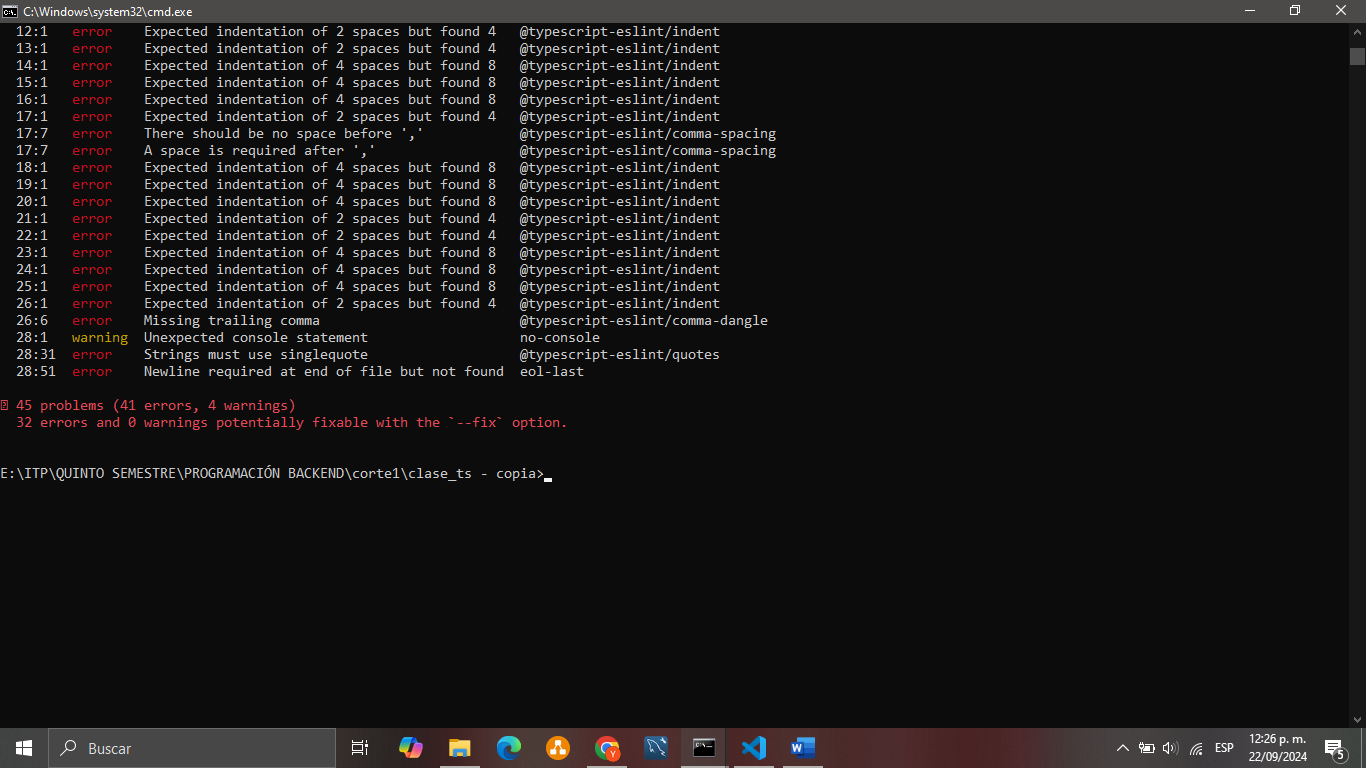


Tenemos estas variables que son declaradas por “let” pero que su valor asignado es un valor constante, entonces lo que debemos hacer es cambiar esa declaración por una declaración “const” de la siguiente manera





Aquí hemos cambiado “let” por “const”, ahora eso deberíamos cambiarlo en cada varible a la cuál tenga un valor ya asignado y que no sea cambiado por otro valor.

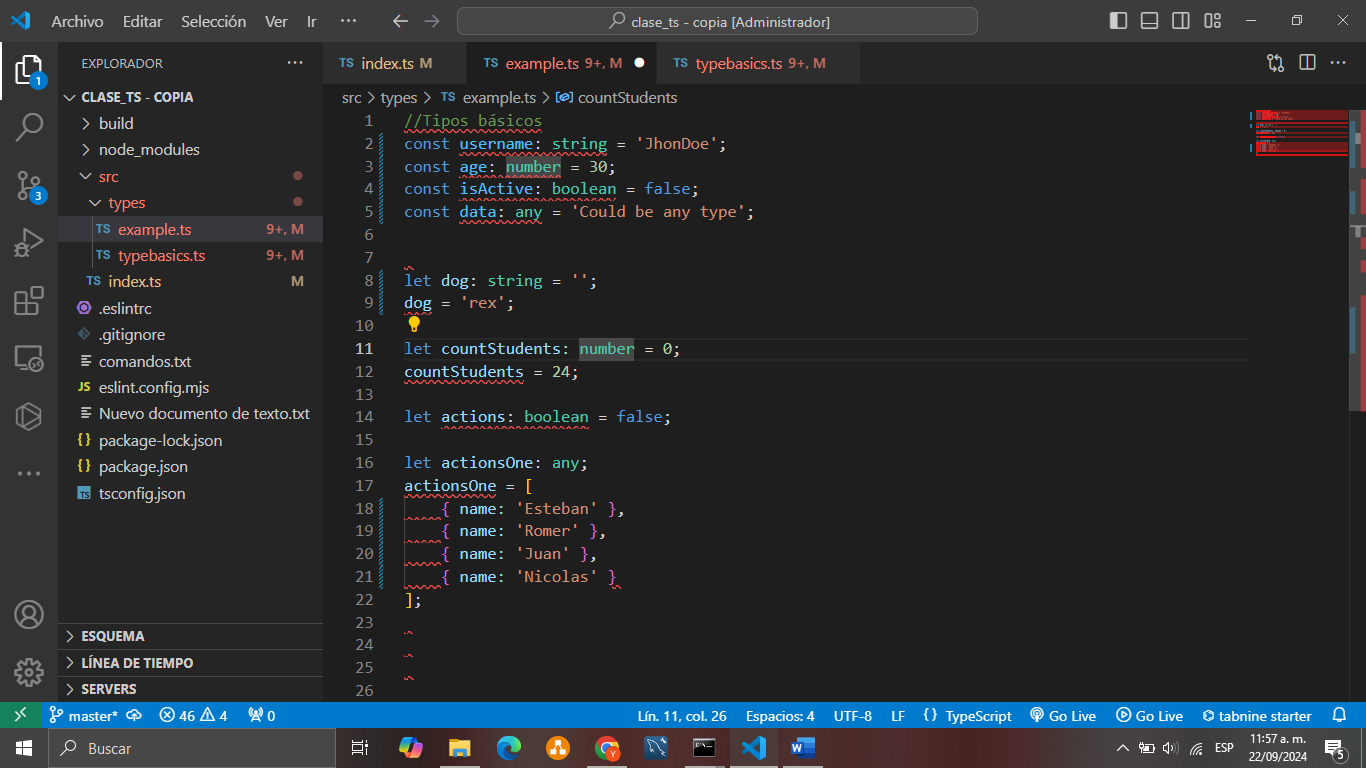


Y cómo podemos ver seguimos disminuyendo los errores

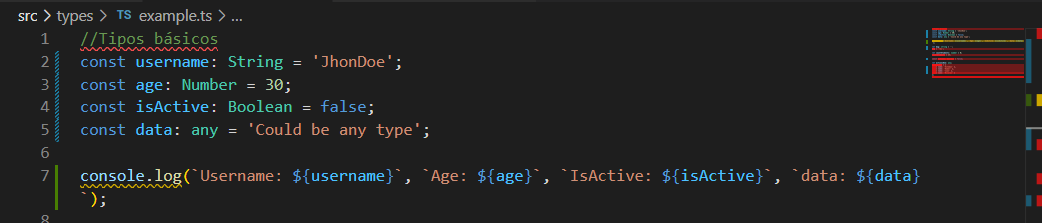
1. Errores de llamado

error 'username' is assigned a value but never used

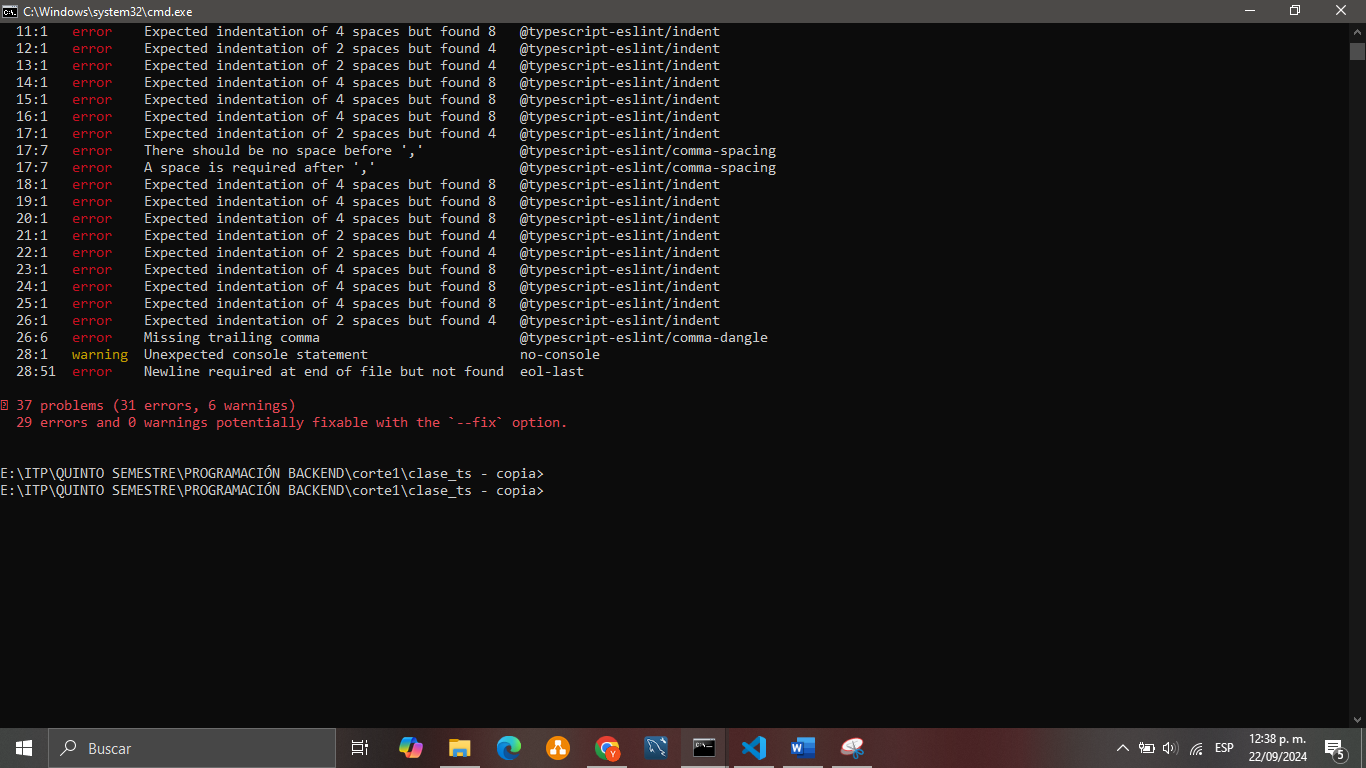
Este error simplemente nos dice que tenemos una variable asignada pero que no está siendo llamada, en pocas palabras que no la estamos utilizando, hay 2 formas de corregir esto, la primera es eliminando esa variable o la segundo es llamándola desde otro lado, en este caso los llamaré en un console.log



Tenemos las siguientes variables, a las cuáles ya le hemos declarado mediante un const, lo siguiente que haremos será llamarlas en cualquier punto, en este caso utilizaremos un console.log cómo lo mencioné anteriormente.



Cómo podemos ver ya no nos marca error, esto lo haremos con cada variable que no haya sido llamada en algún otro punto de nuestro proyecto.



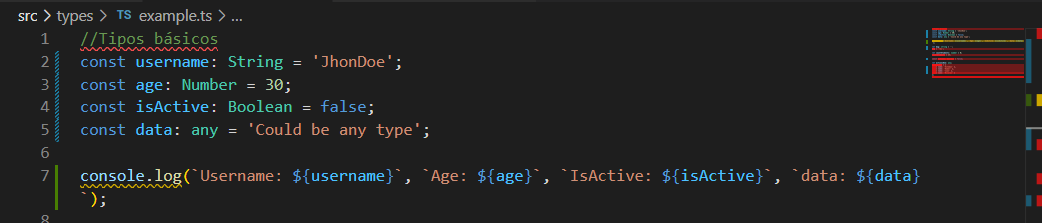
Cómo podemos ver ya casi acabamos

1. Errores de espacio

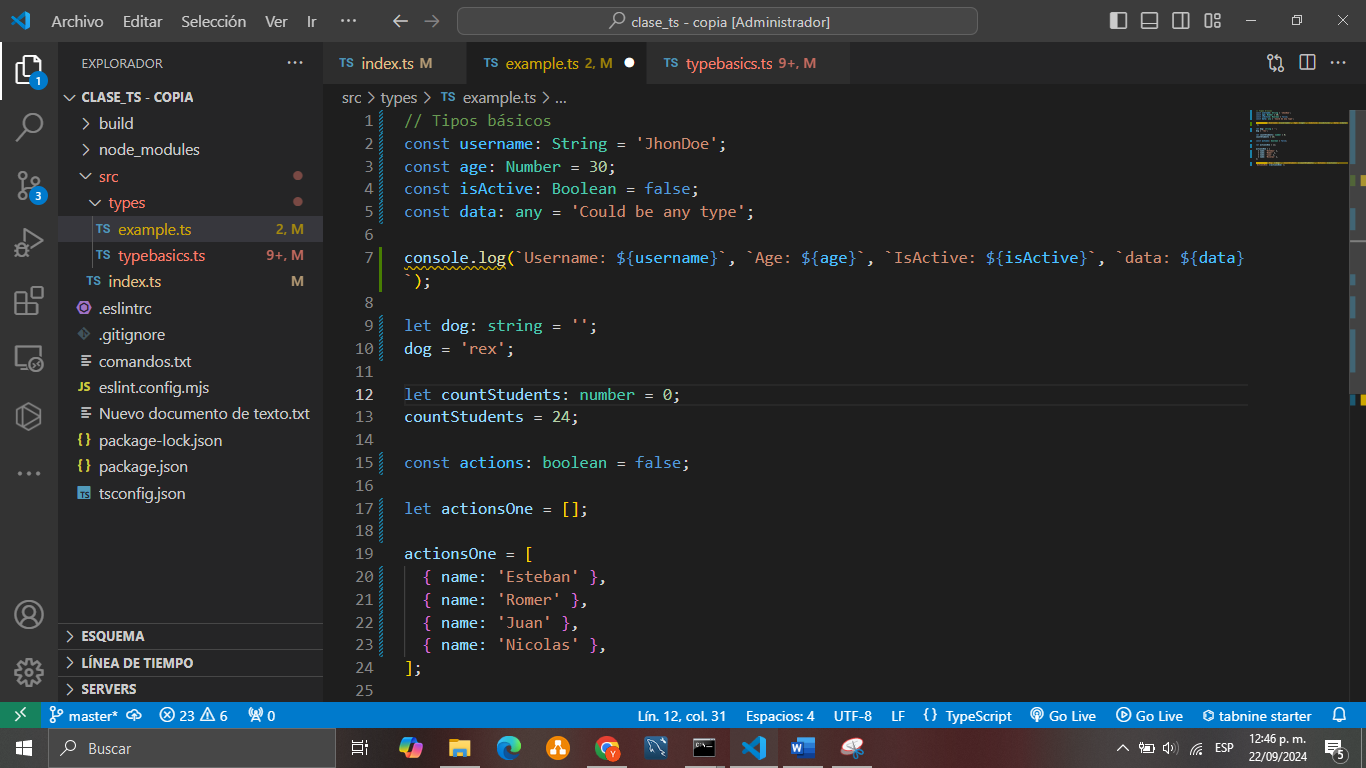
Estos errores son comunes, pero tienen sus reglas

error Expected exception block, space or tab after '//' in comment

Este error nos indica que a la hora de hacer un comentario después de ‘//’ debe haber un espacio de la siguiente manera:



Tenemos el siguiente comentario sin el espacio detrás de ‘//’ por lo que nos marca error entonces lo que debemos hacer es lo siguiente:

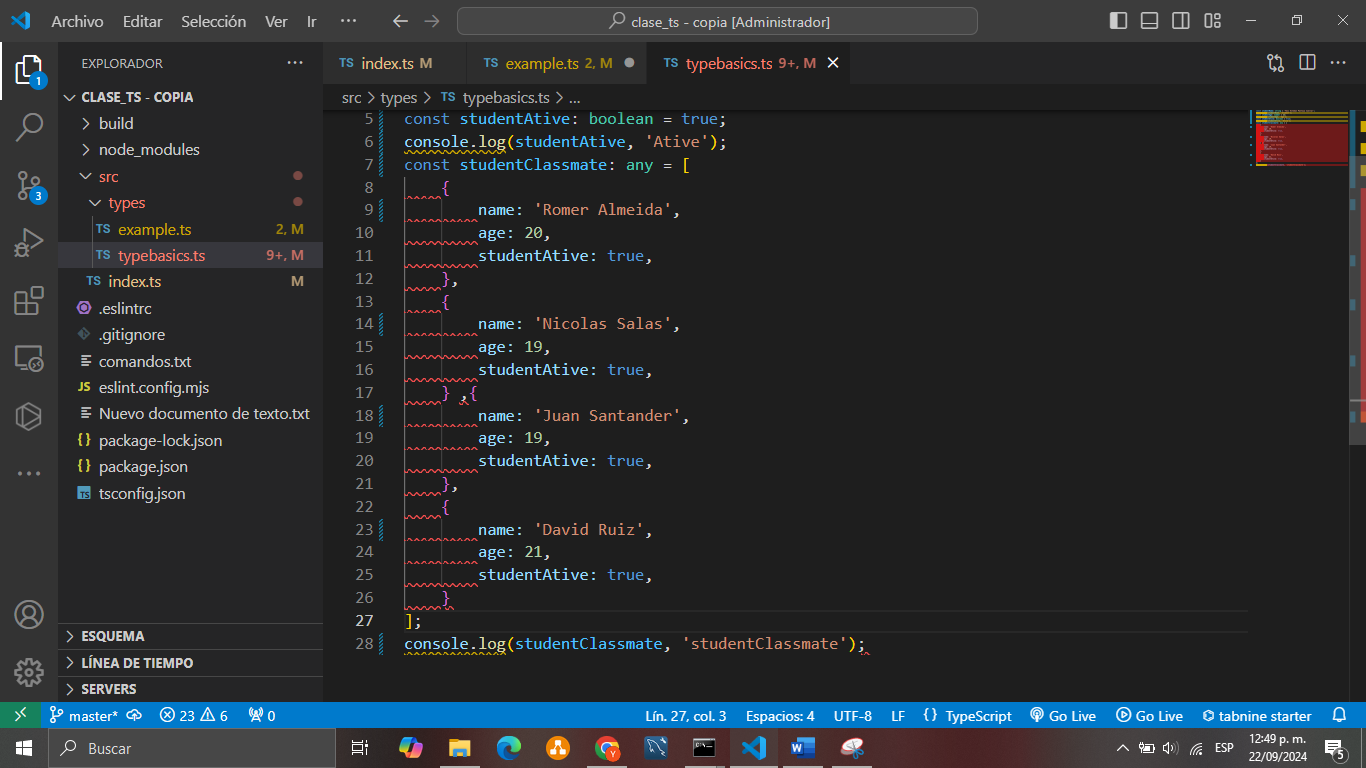


Aquí ya hemos solucionado el primer error por espacio. Ahora vamos por el siguiente error de espacios.

error Expected indentation of 2 spaces but found 4

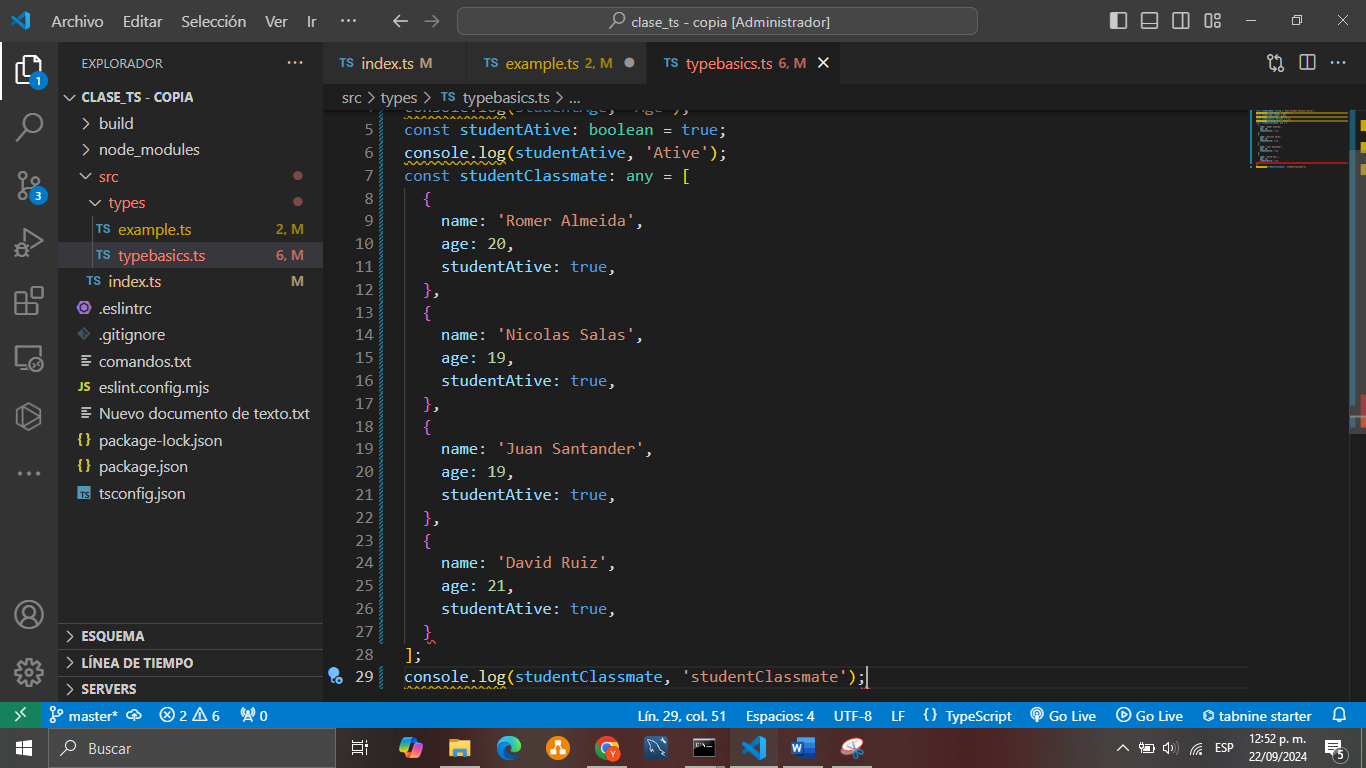
error Expected indentation of 4 spaces but found 8

Estos errores nos quieren decir que tenemos más espacios de los que deberíamos tener



Las líneas rojas son los espacios dónde tenemos que corregir para así solucionar nuestro error

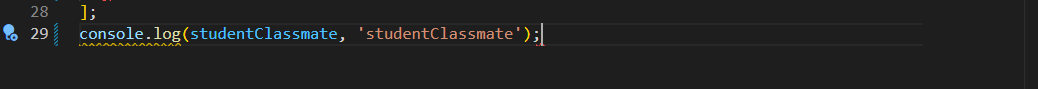




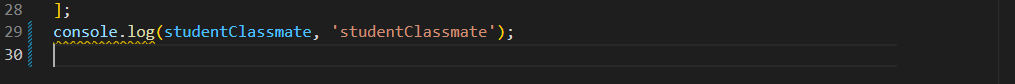
Cómo podemos ver ya hemos solucionado esa parte de los espacios en los cuáles hay más de los que debería.

error Newline required at end of file but not found

Este error nos indica que al finalizar nuestra línea de código debemos aumentar una línea más y dejarla sin nada escrito

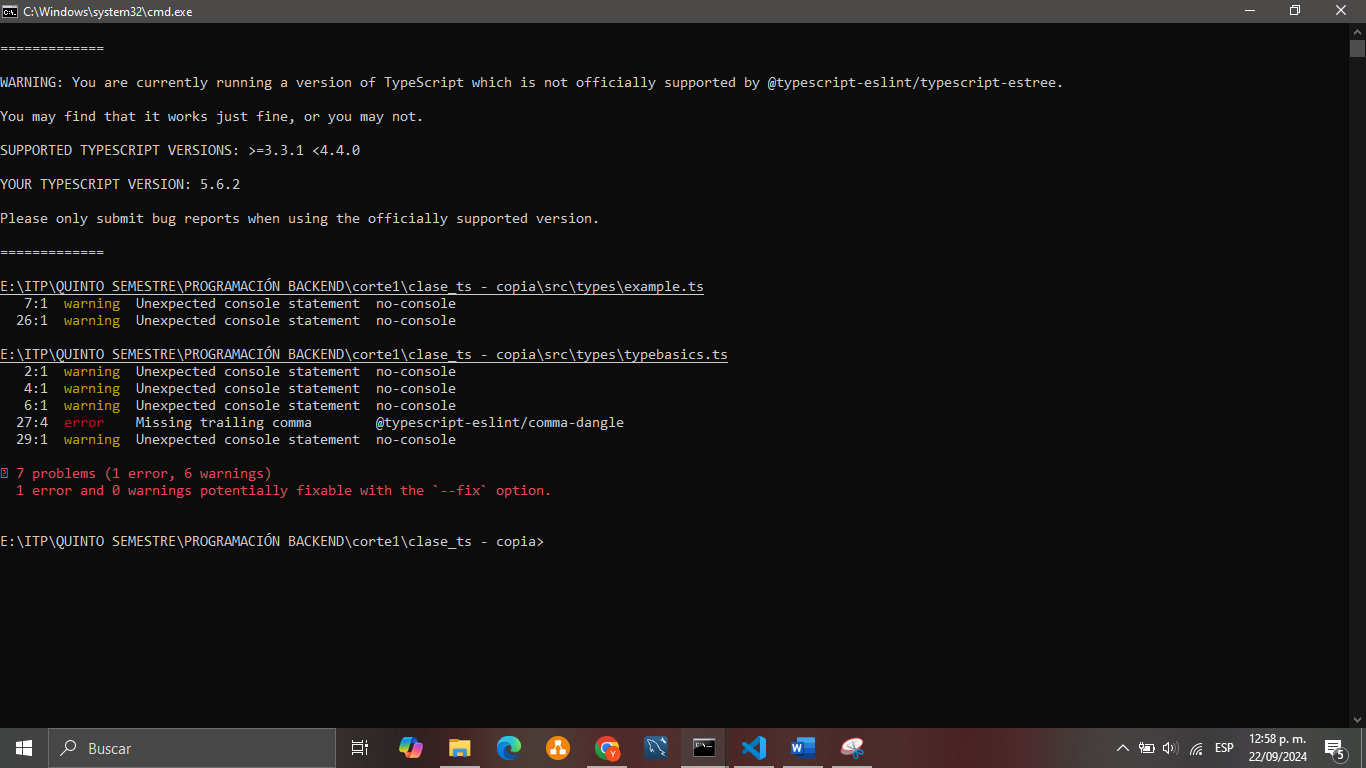


Tenemos esta última línea de código el cuál nos marca error al finalizar, (en los “;”), lo que debemos hacer es aumentar una línea más y dejarla en blanco



Cómo podemos ver ya no tenemos el error marcado.

Y así es cómo logramos arreglar nuestros problemas de espaciado.

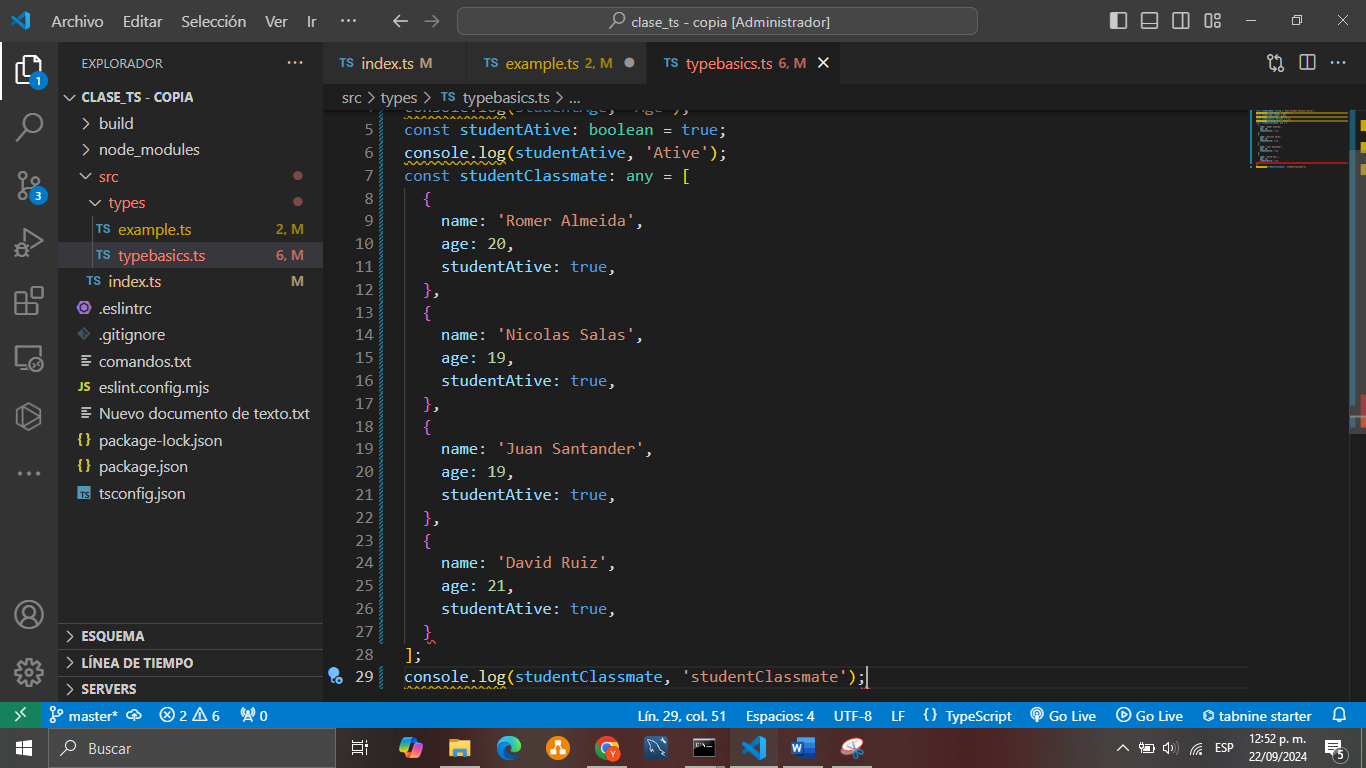


Ya casi finalizamos nuestra corrección de errores.

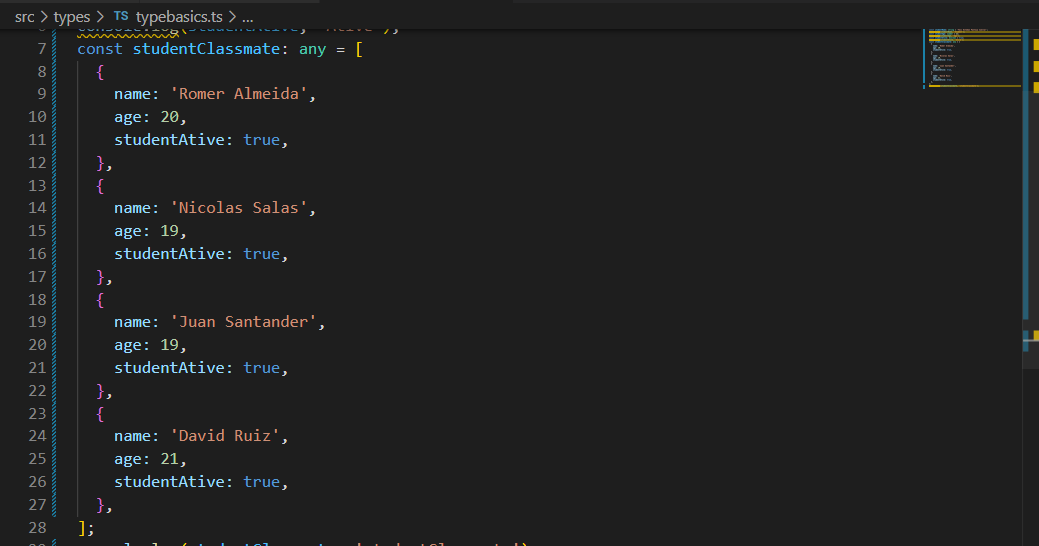
1. Errores de coma “,”

error Missing trailing comma

Este error significa que en alguna parte de nuestro proyecto nos hace falta una coma.

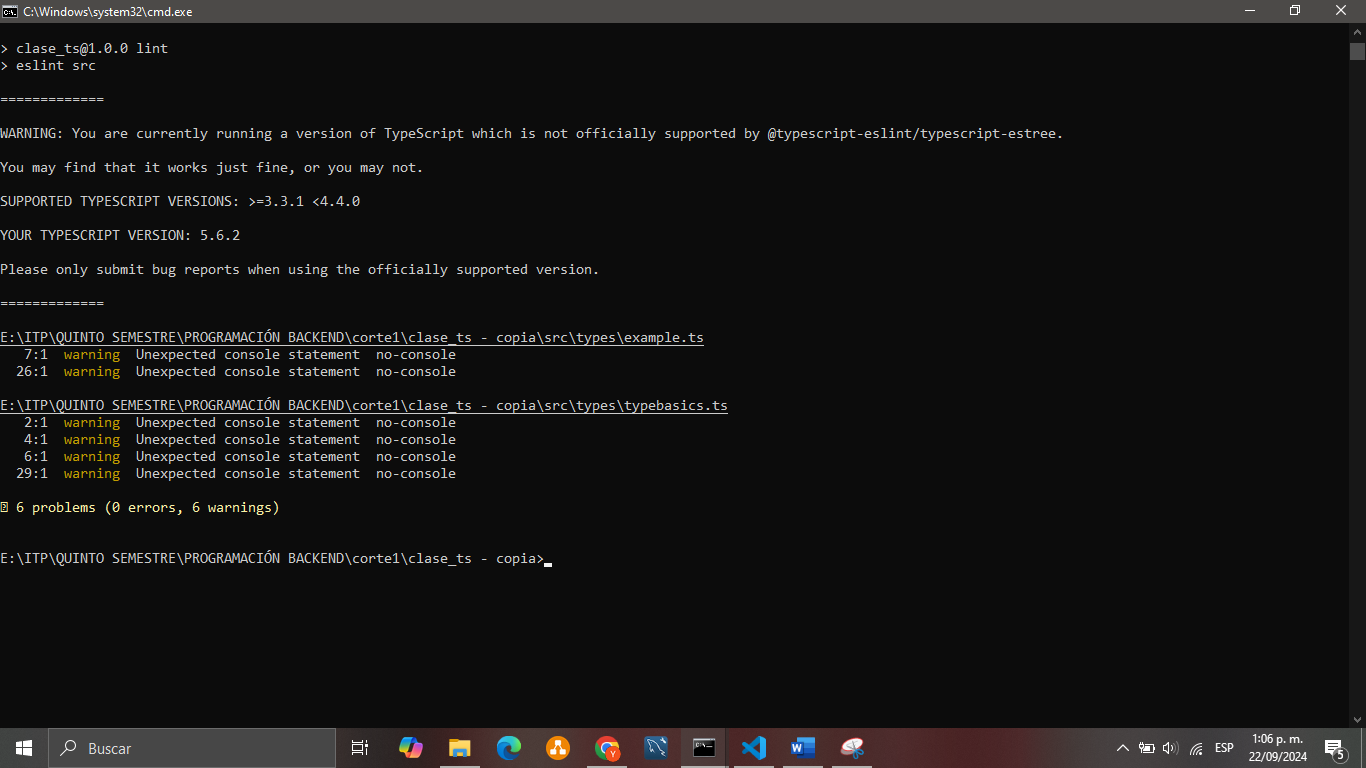


Cómo podemos ver en la línea 27 tenemos el error de la coma faltante, esto lo marca eslint debido a que en futuras actualizaciones nos permite agregar más objetos a nuestro variable.



Y listo, así es cómo hemos solucionado nuestros errores marcados por eslint.





Link de vídeo: <https://youtu.be/zROG6NQVsJY>