Informe Herramientas

Carlos Sousa González

Introducción:

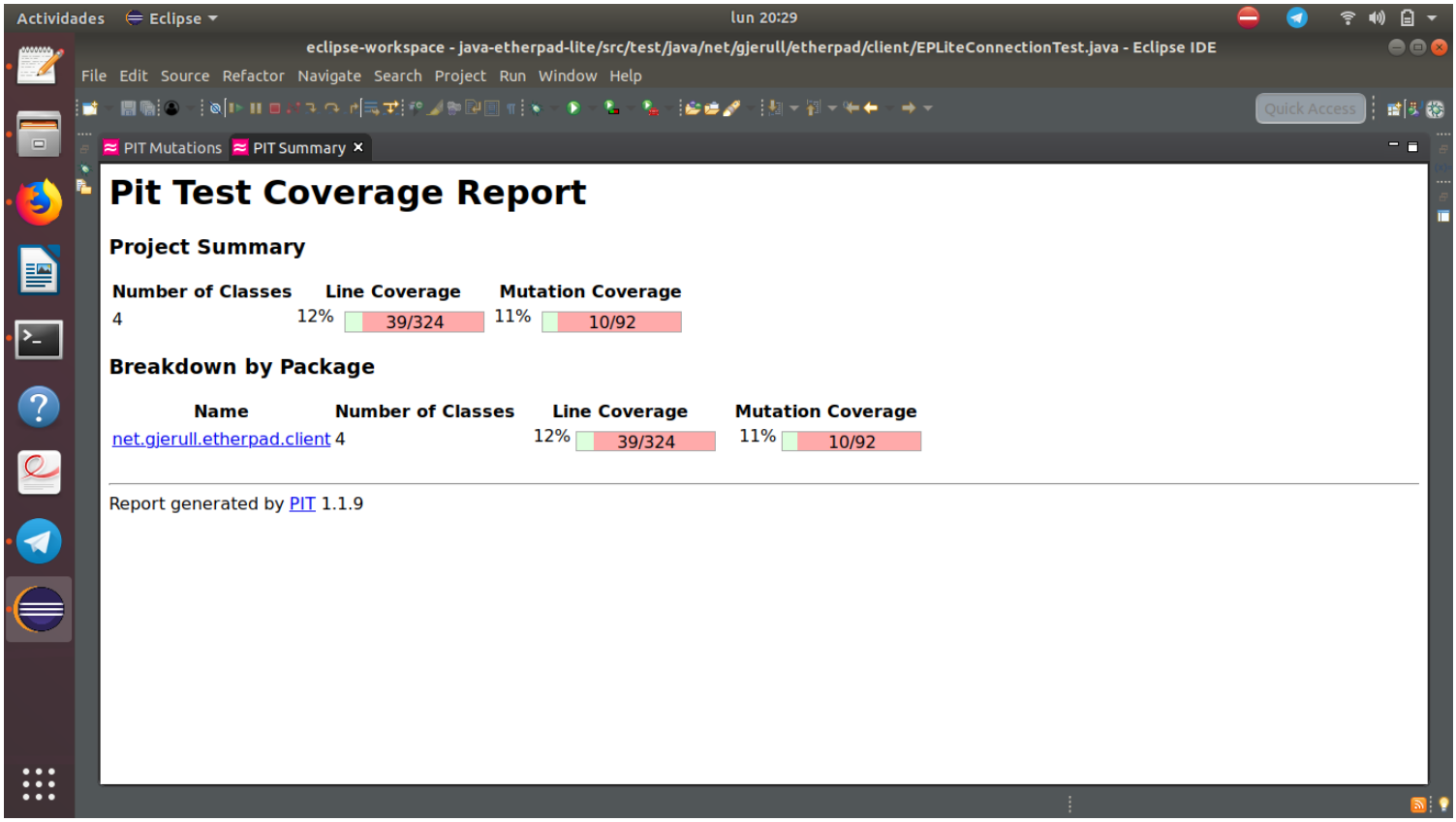
En esta práctica se pretende usar una serie de herramientas para mejorar las pruebas unitarias y de integración del proyecto java-etherpad-lite, he decido usar las siguientes herramientas para conseguir una visión global sobre la calidad de las pruebas proporcionadas:

- PIT (Conocer la cobertura de instrucciones y de condiciones)
- Mockito/MockServer (Falsear las respuestas de un servidor para probar los casos de uso que necesiten de este)
- CheckStyle (Verifica una serie de reglas de programación con respecto al estilo, siguiendo un stándar)
- SpotBugs (Recopila los errores y warnings en una lista)

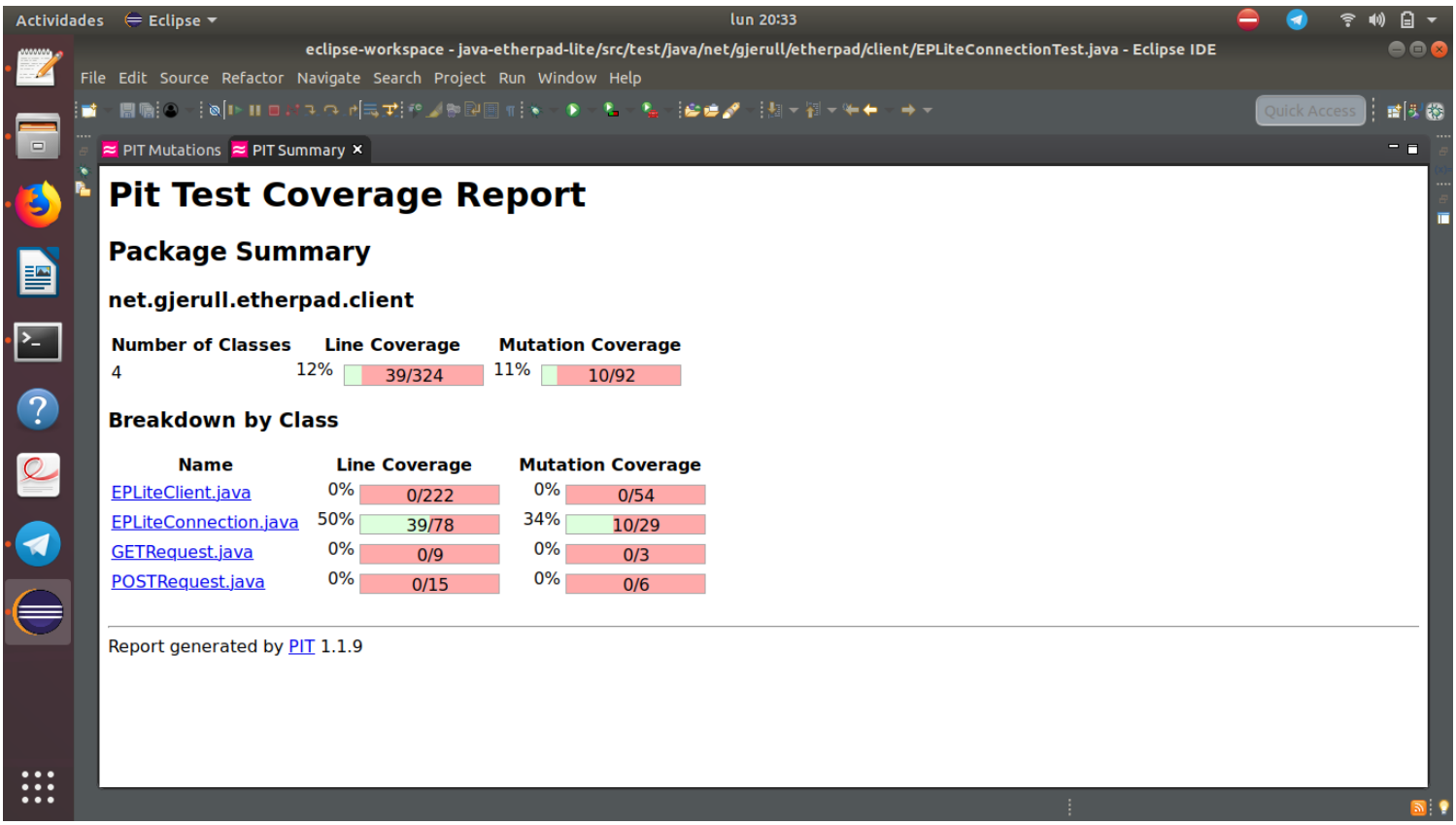
Herramientas:

PIT: Esta herramienta comprueba el porcentaje de cobertura tanto de instrucciones como de condiciones.

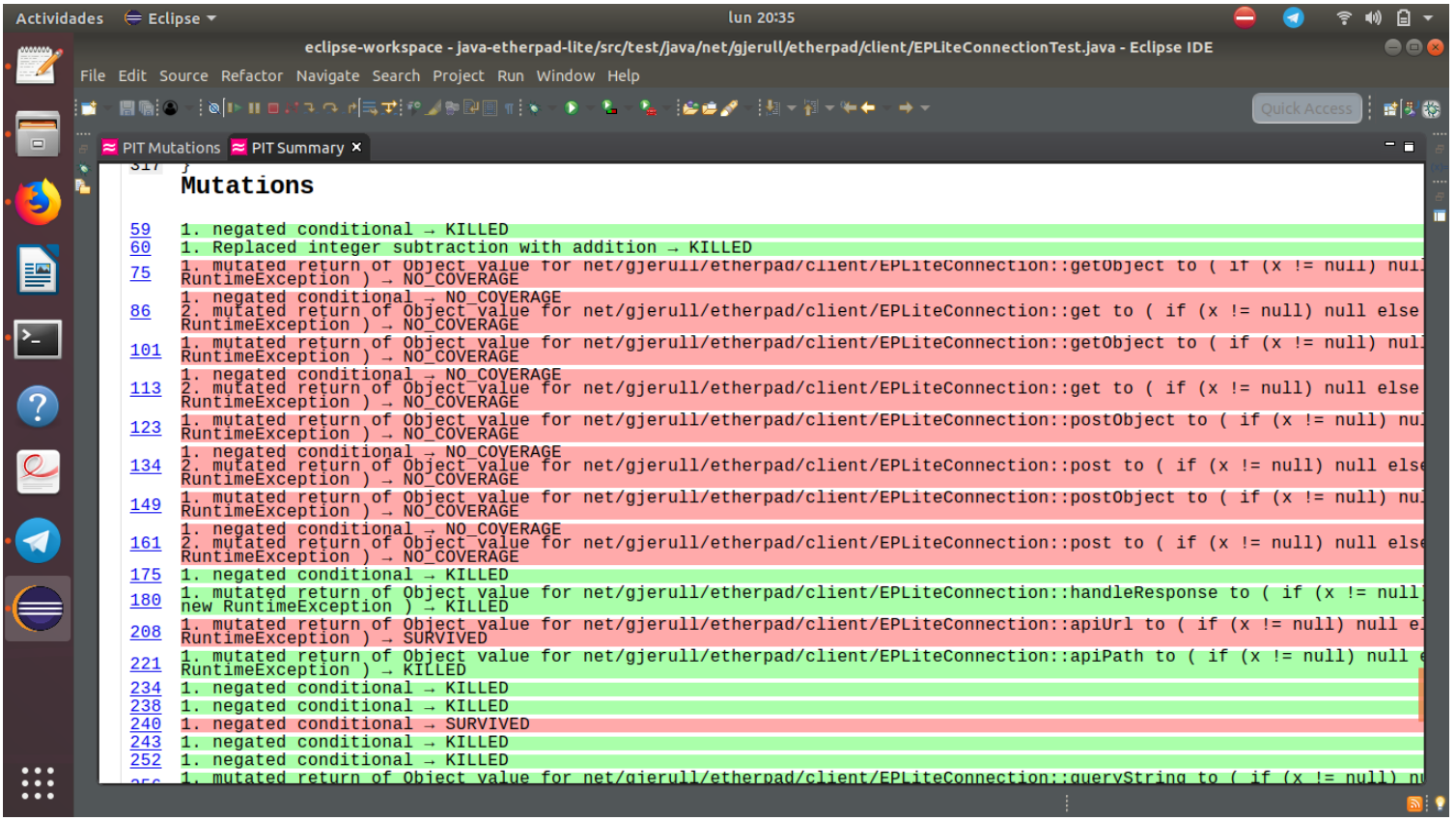
Como se puede apreciar en la siguiente imagen, la cobertura de lineas/instrucciones es muy baja (esto ya implica que la de condiciones también lo será), pero tener una cobertura de lineas alta, incluso de 100%, no implica que se esten probando todos los casos si la de condiciones no lo es.

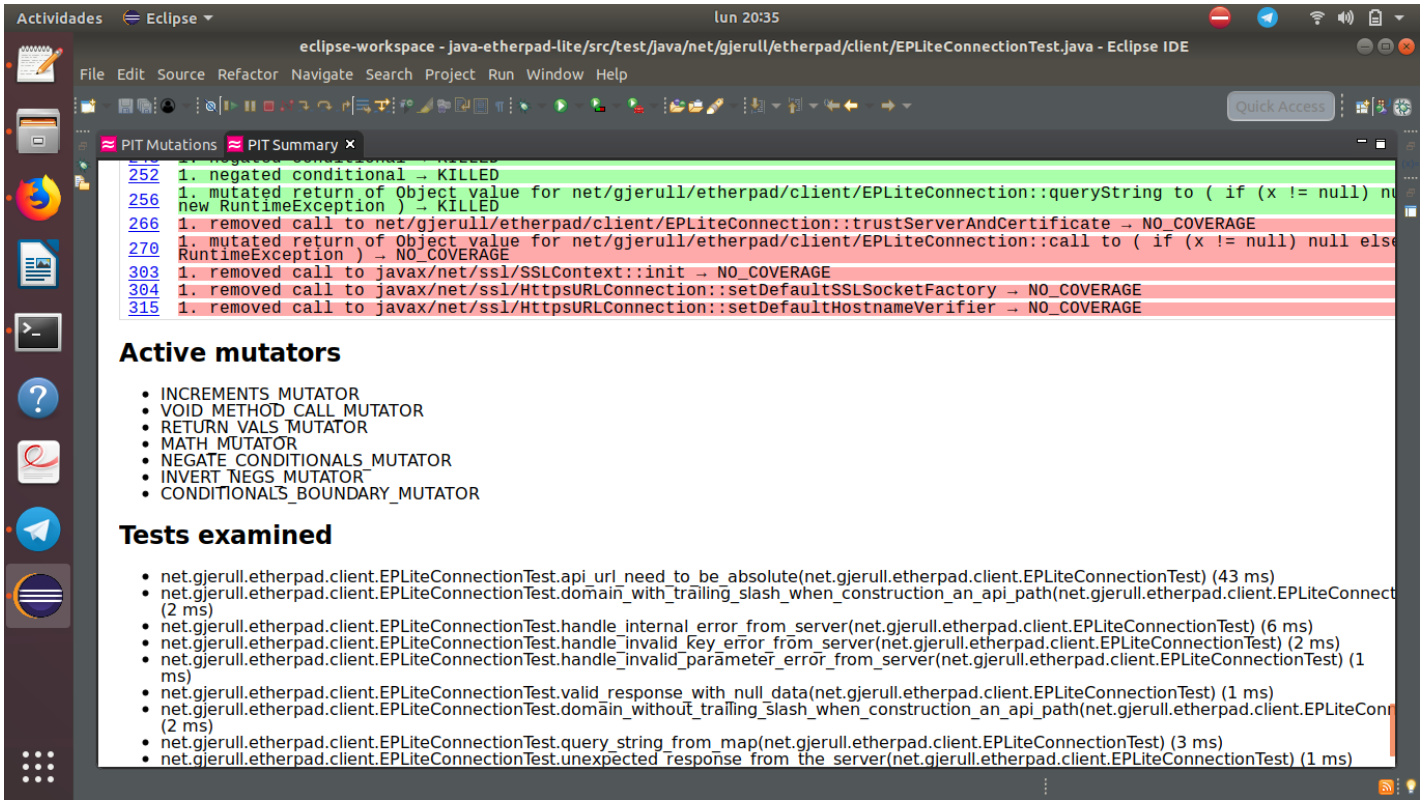


Entrando más en profundidad (clickando en el link proporcionado), se puede ver la cobertura de las clases individualmente, como se ve a continuación.



Si seguimos indagando podemos llegar al código fuente y ver las lineas coloreadas según si están cubiertas no.

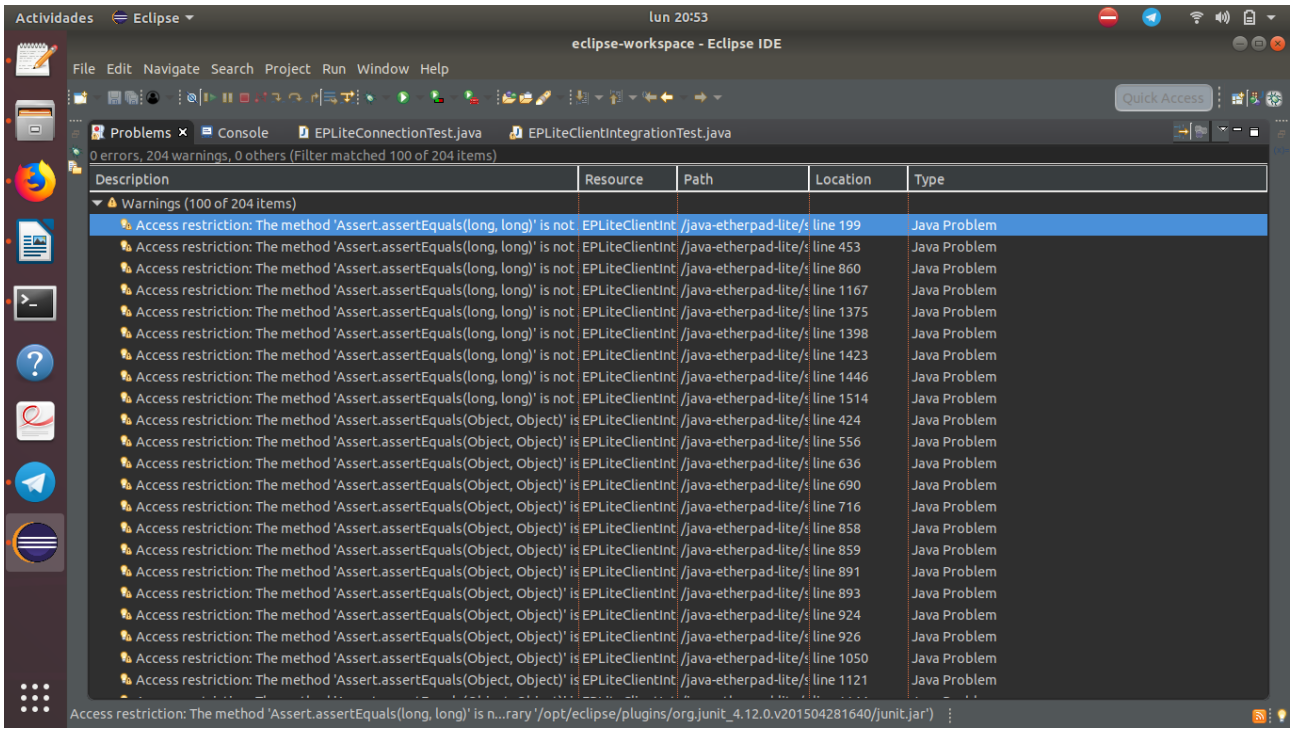




Información concluida con PIT:

Las coberturas son extremadamente bajas, ya que no se llega a probar ni un 25% del código, suponiendo que el 25% fuese código que sabemos que no tenemos que probar (debido a diversas razones), lo mínimo exigible sería un 75% en ambos campos.

SPOTBUGS: Esta herramienta resume todos los errores y warnings de los proyectos abiertos en el entorno de programación en una lista. Como podemos ver indica la descripción, el nombre del archivo, el path, la línea y por último el tipo. (Esto puede ser útil para filtrado de errores o para obtener una idea general sobre como se encuentra actualmente el proyecto).



Información concluida con SPOTBUGS: En mi opinión no es imprescindible el hecho de corregir todos los warnings, ya que se puede “convivir” con algunos en la época de desarrollo pero deberia ser preocupante tener 100.

CHECKSTYLE: Esta herramienta permite standarizar el estilo de un código, permitiendo que muchas personas usen el mismo estilo y mejorando las posibilidades de lectura por personas ajenas a su creación, trata desde como hacer los comentarios hasta como indentar, pasando por no pasarse de la capacidad de columnas de la pantalla y comprobando que no haya “TODOs”.

A continuación podemos ver algunos ejemplos:

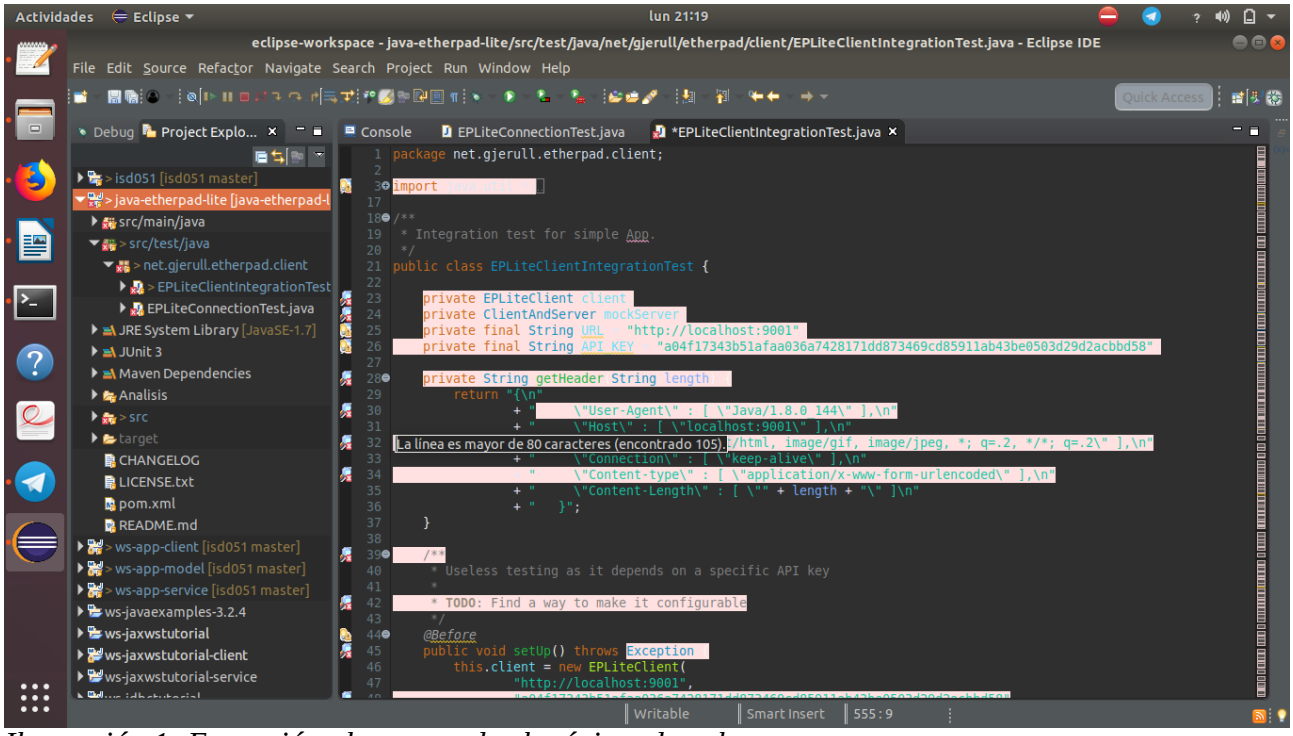


Ilustración 1: Excepción al pasarse de el máximo de columnas

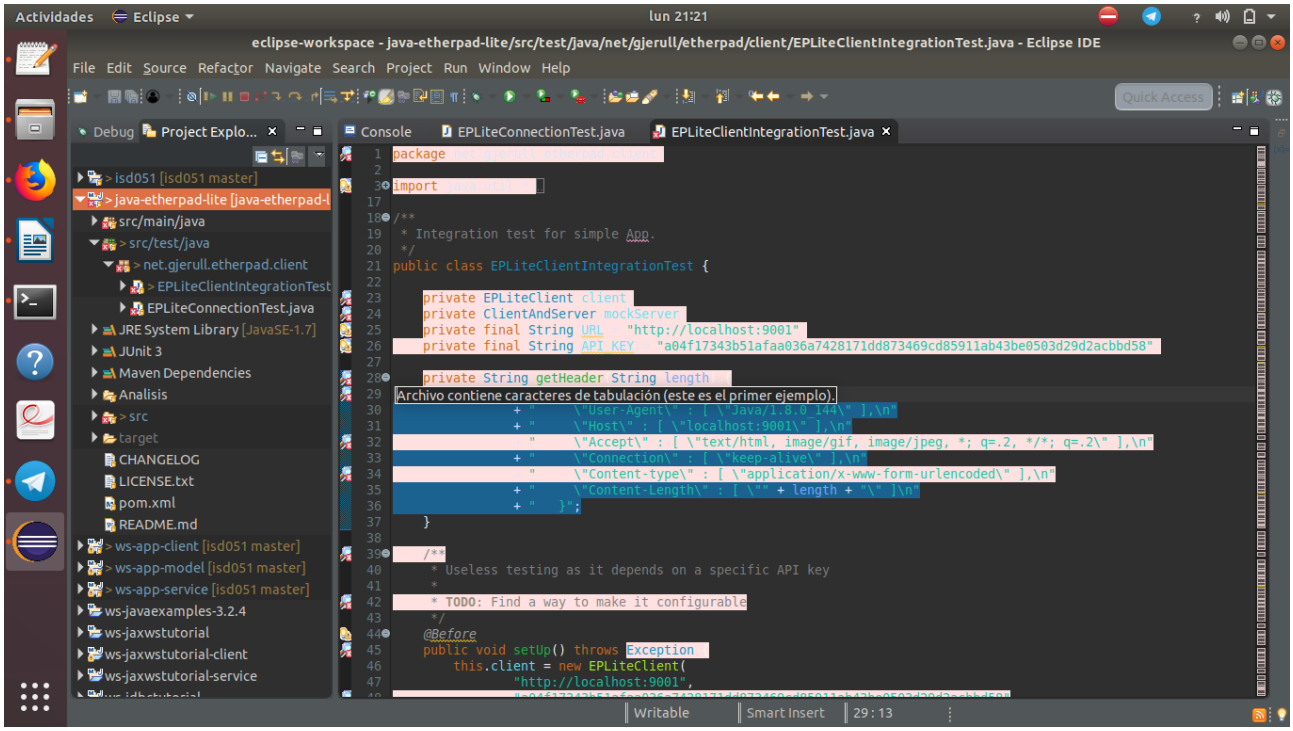


Ilustración 2: Excepción al errar en la tabulación

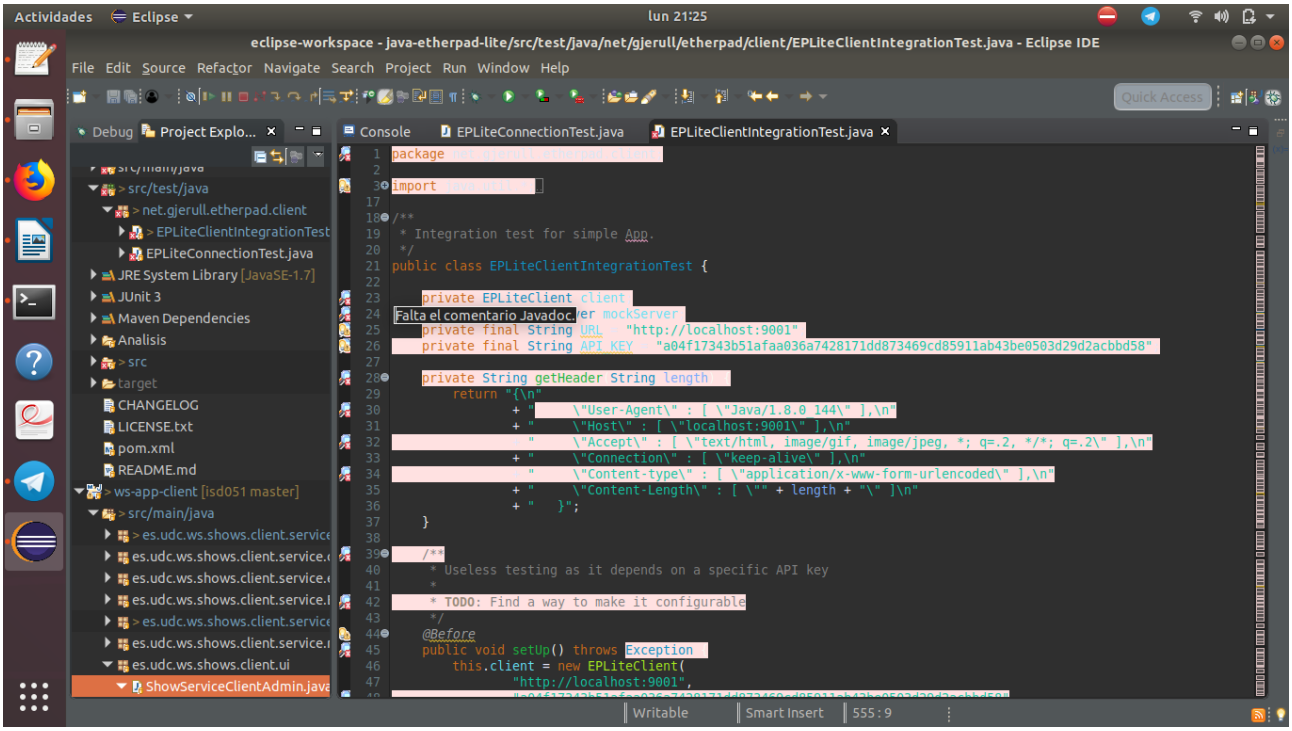


Ilustración 3: Falta de comentarios

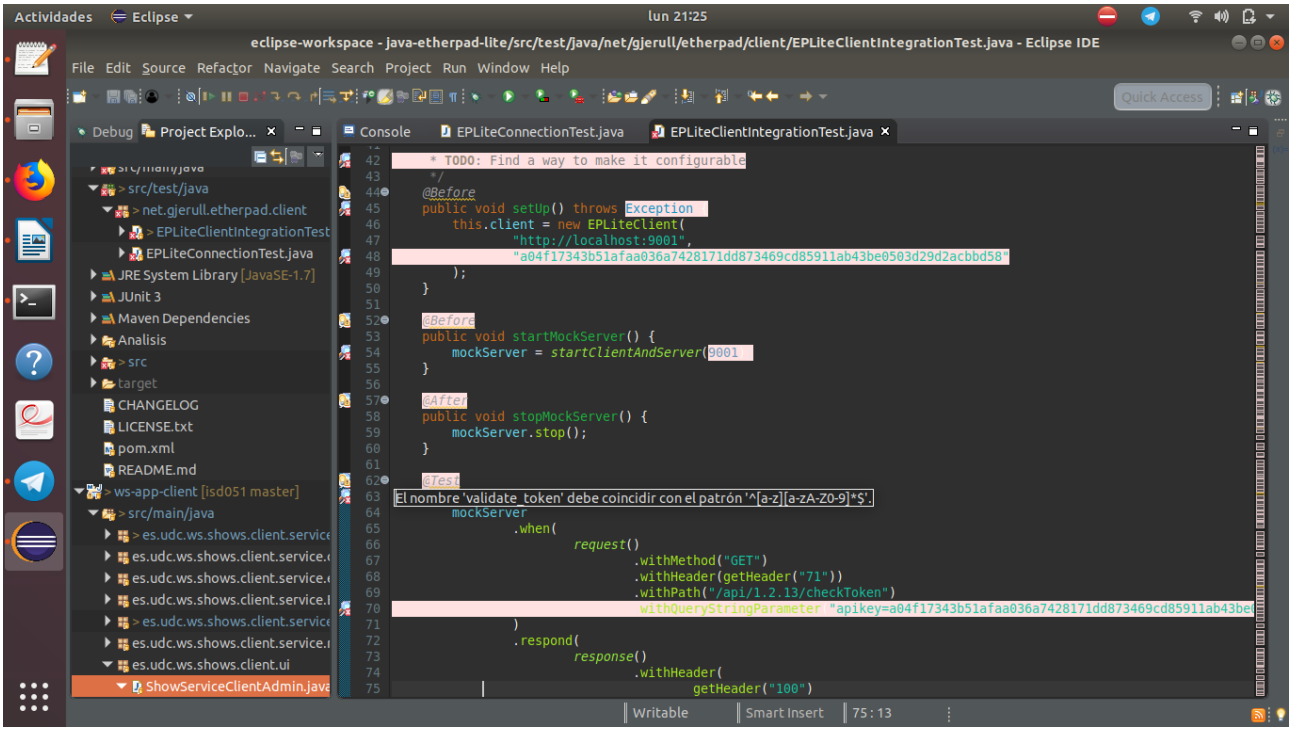


Ilustración 4: Excepción por no usar el CamelCase

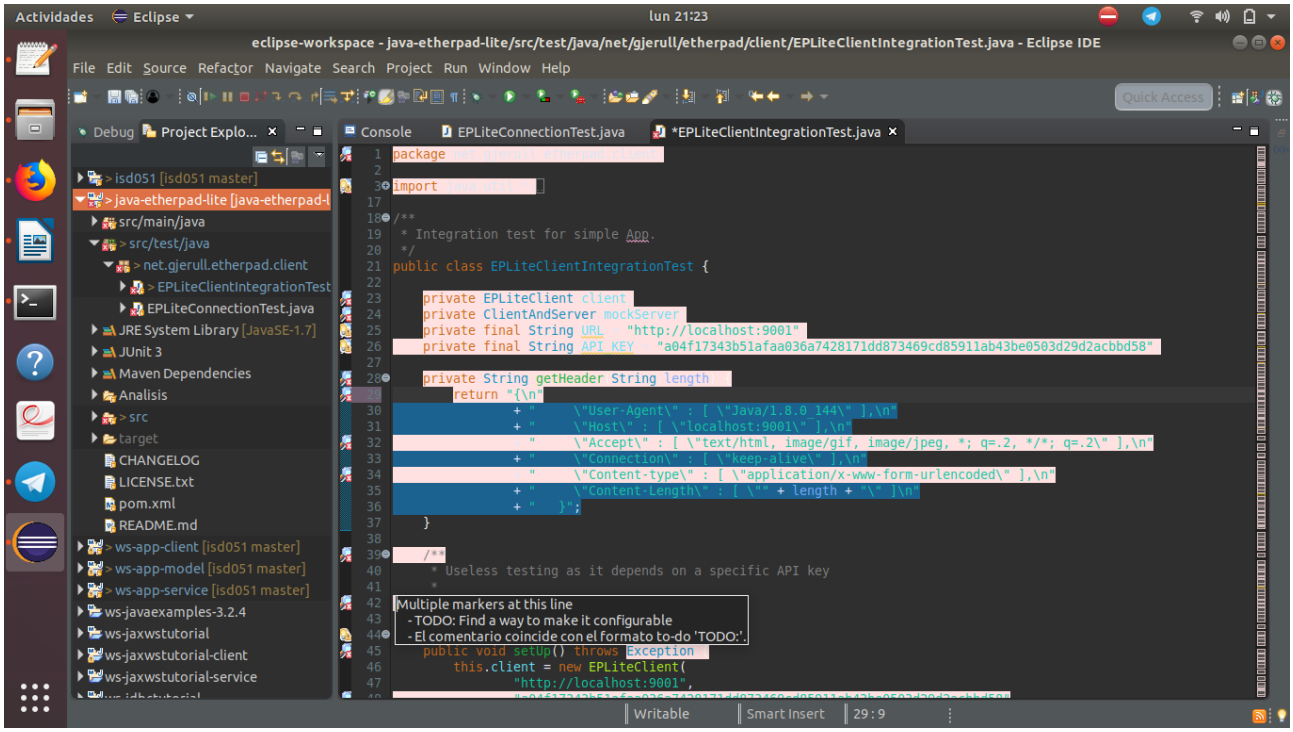


Ilustración 5: Excepción "TODO" (método por implementar)