P5

Tipos de pruebas:

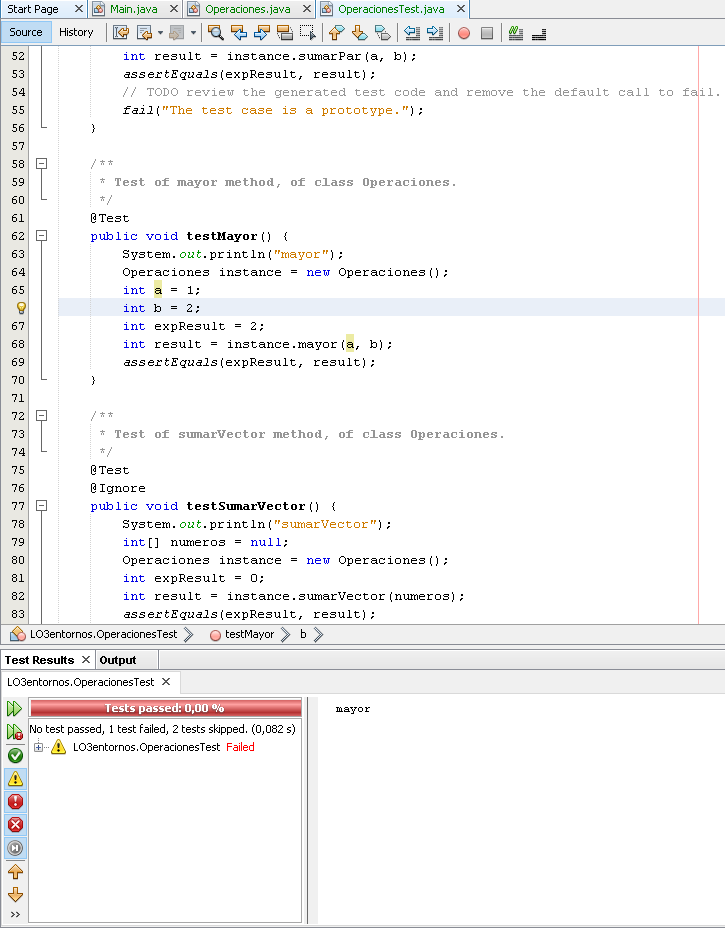
* Funcionales: Garantizan que las características y funcionalidades del software se comportan según lo esperado sin ningún problema.
* Estructurales: Son una aproximación al diseño de casos de prueba en donde las pruebas se derivan a partir del conocimiento de la estructura e implementación del software.
* Regresión: Son cualquier tipo de pruebas de software con el objeto de descubrir errores (bugs), carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, causados por la realización de un cambio en el programa.

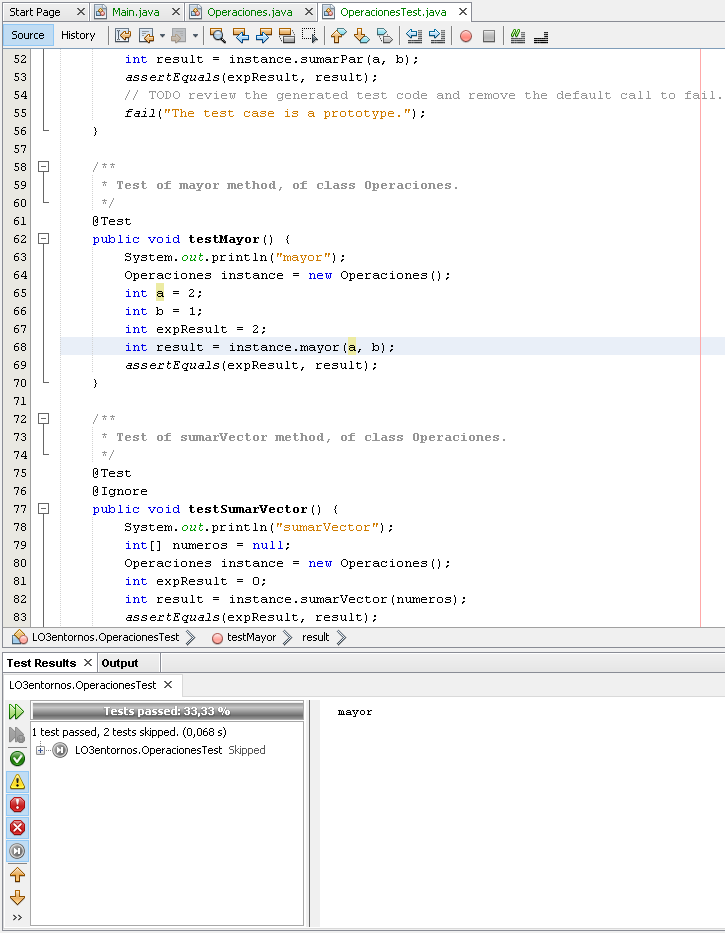
Pruebas de la caja blanca: Se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. El ingeniero de pruebas escoge distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados.

Pruebas de la caja negra: Es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

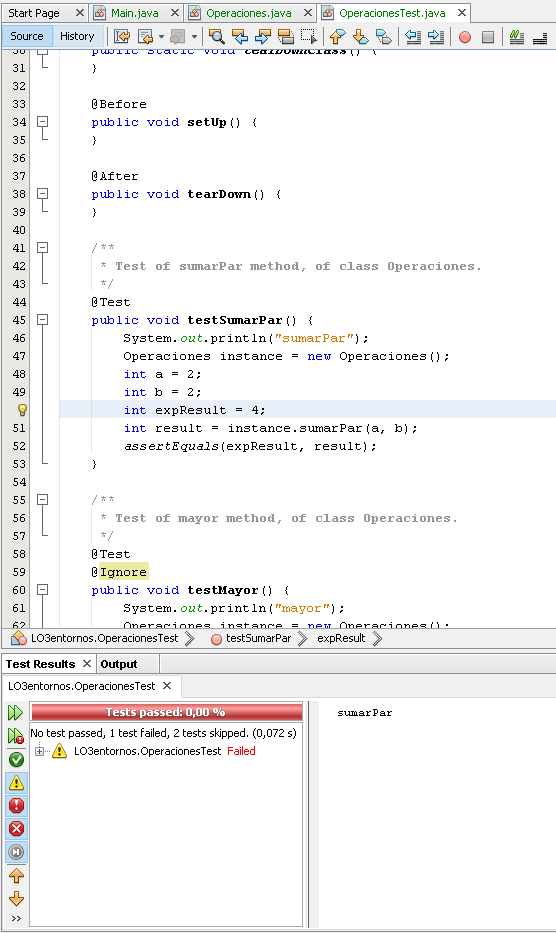
Métricas de calidad de software: Es un conjunto de medidas utilizadas para estimar la calidad de un proyecto a desarrollar, entre otros conceptos, y que permiten comparar o planificar estas aplicaciones.

P7

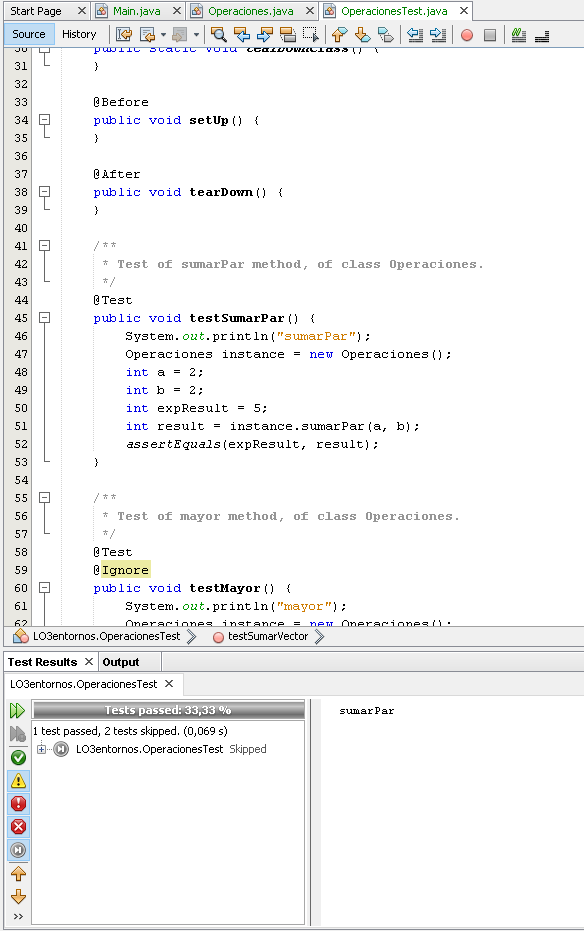
En este caso, teniendo variable a = 1 y b = 2 el resultado debería ser 2, pero en este caso da error, es debido a que da su resultado en base a la variable a.



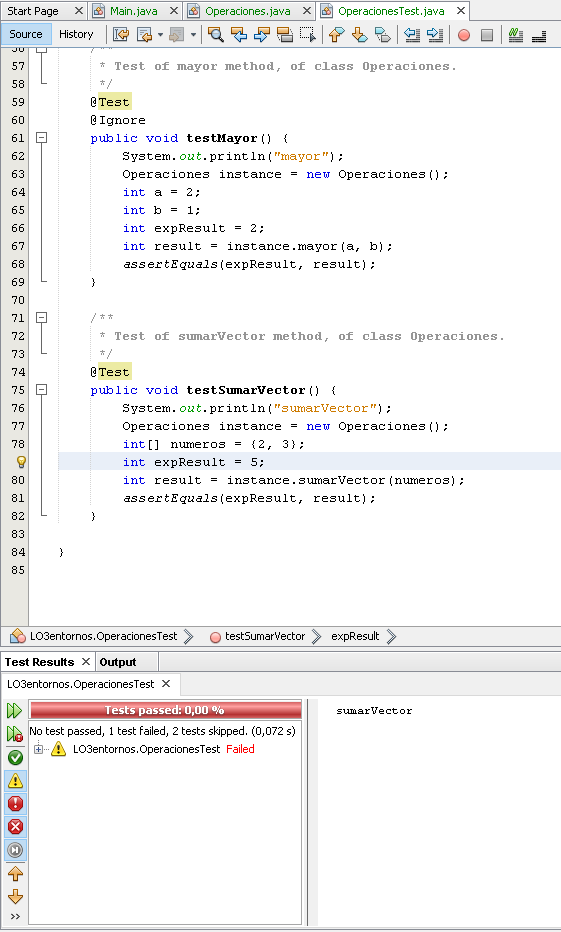
En este caso damos la vuelta a las variables, a = 2 y b = 1, en este caso el test no da error porque la variable a es la mayor.



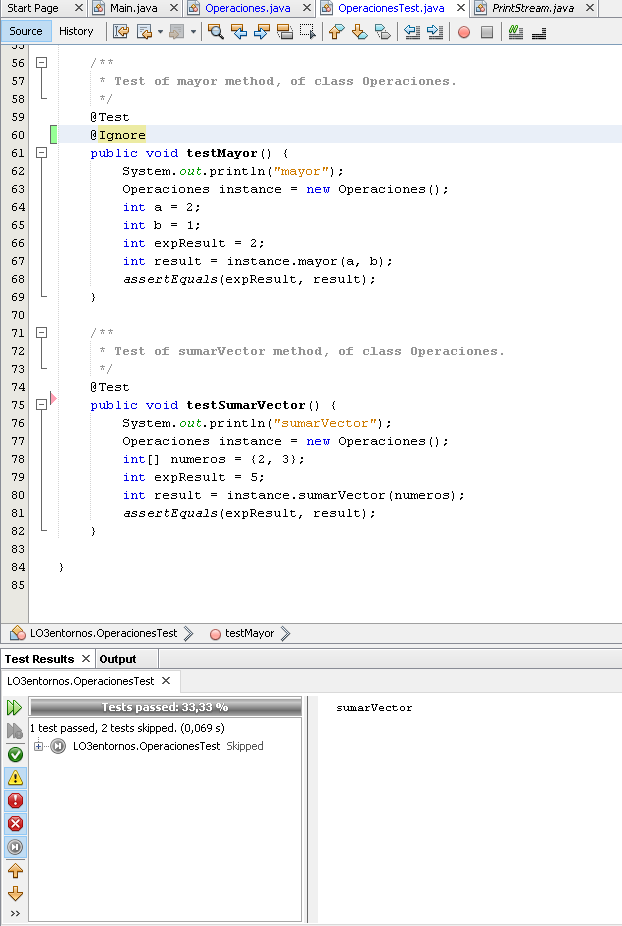
En este caso siendo las variables a y b valor 2 da error porque da 5 esperando como resultado 4.



El código tiene un fallo y da 5 en vez de dar 4, por eso antes nos daba error de la misma manera, porque le suma 1 al resultado en vez de dejarlo.

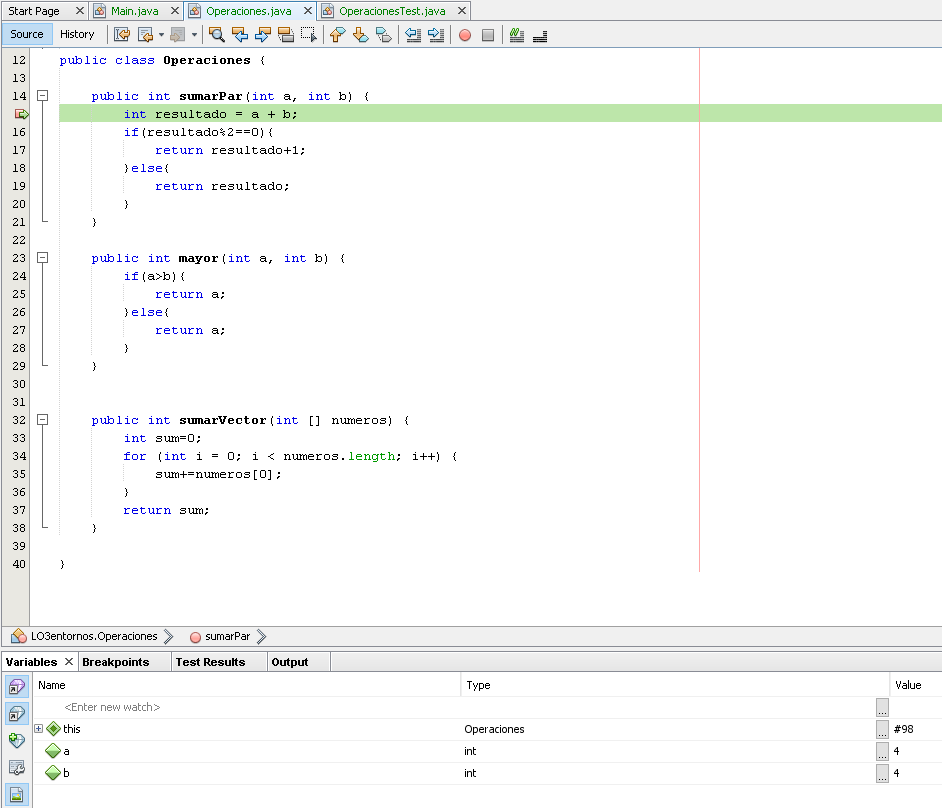


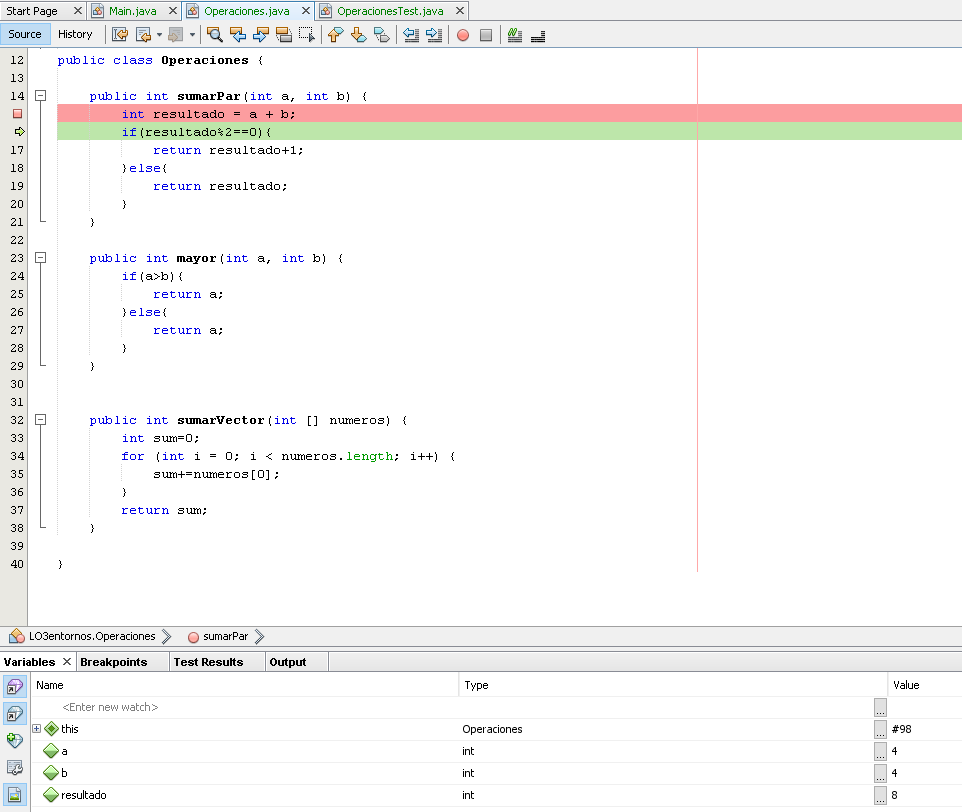
En este caso da error el código a pesar de que la suma es correcta, es un error de código.

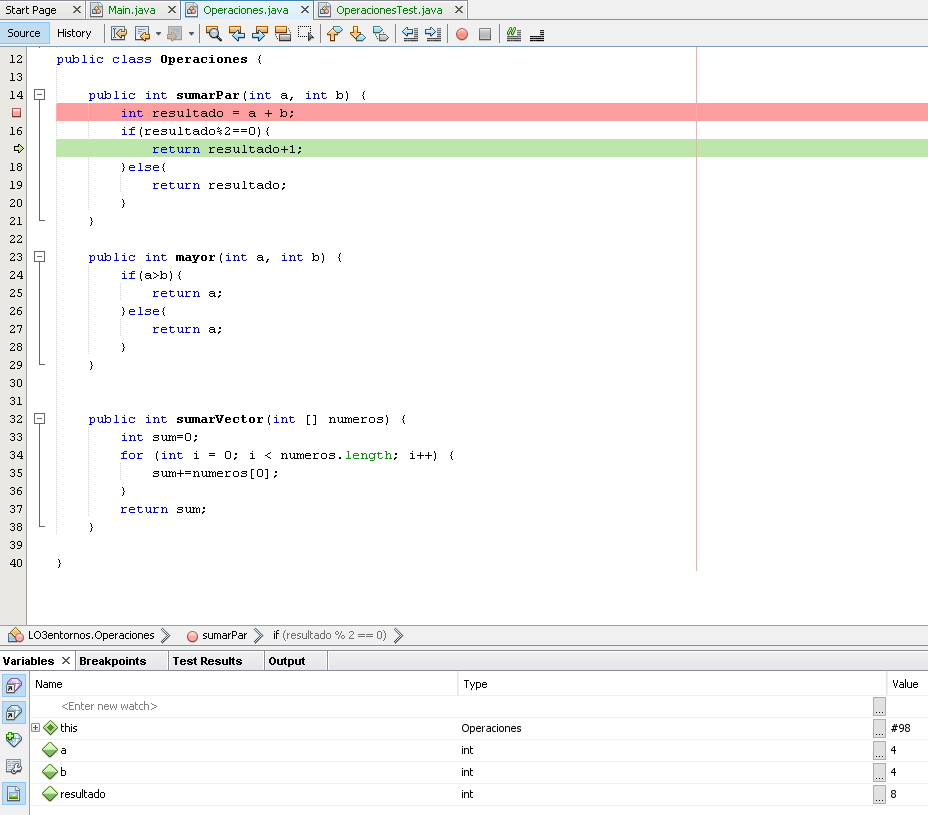


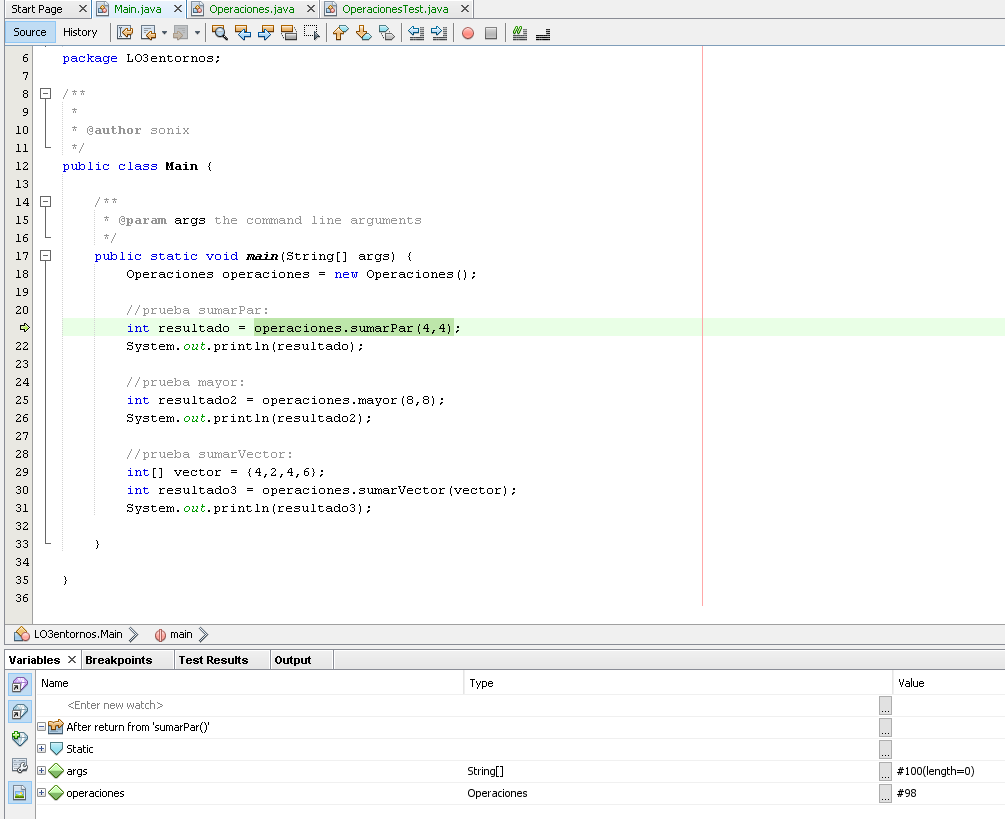
Después de arreglar el error de código la operación no da error.

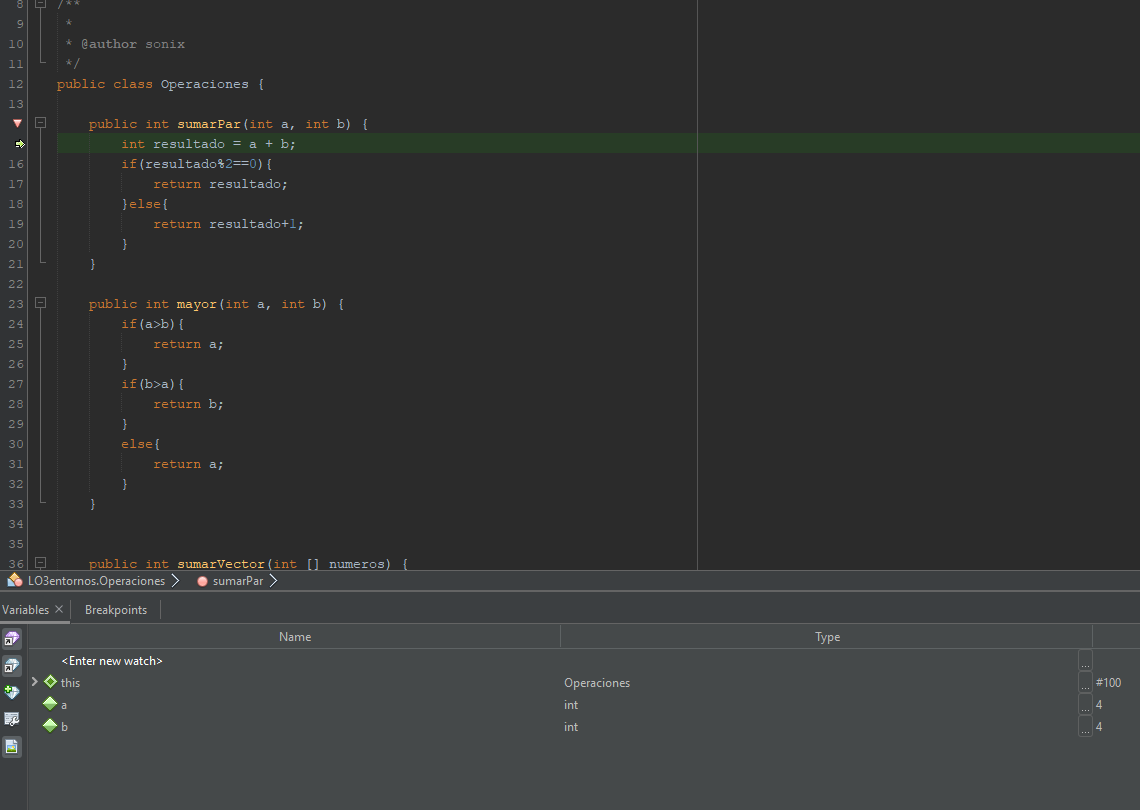
M6

En las siguientes imágenes podemos ver el proceso de depuración antes de realizar los cambios.

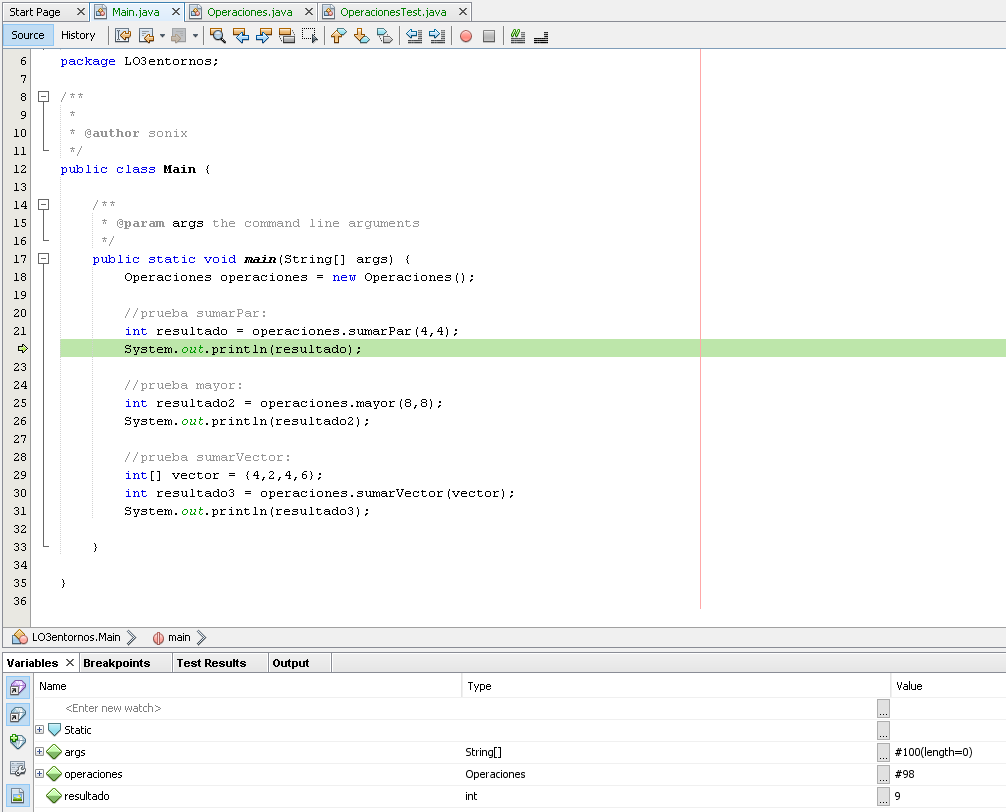


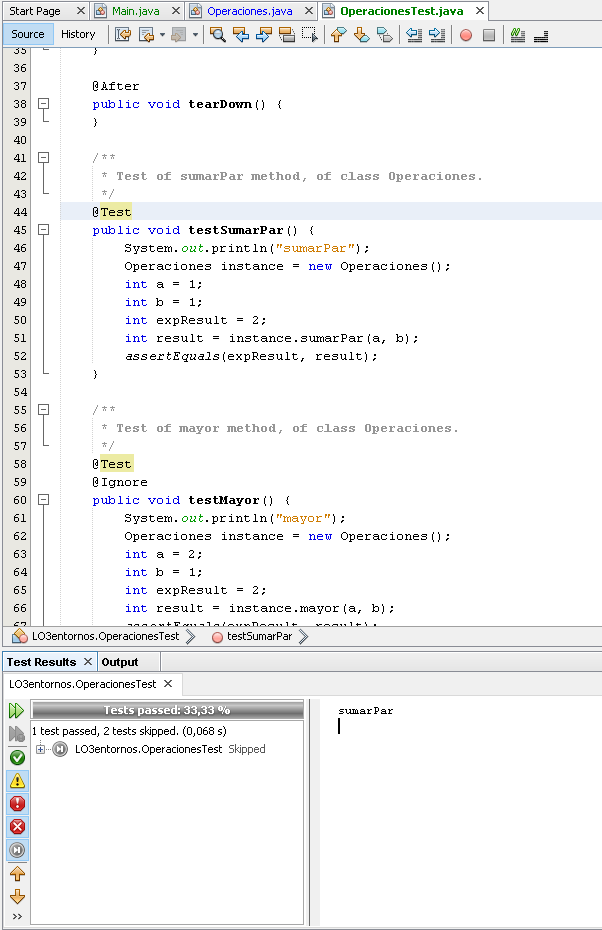


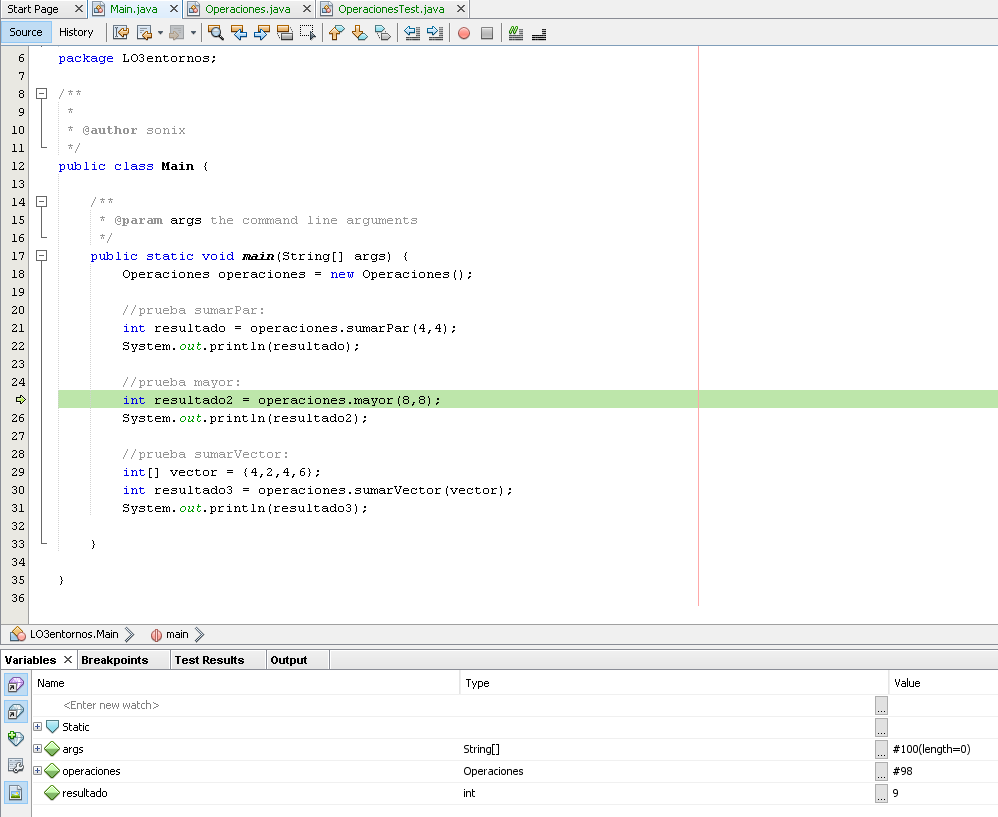


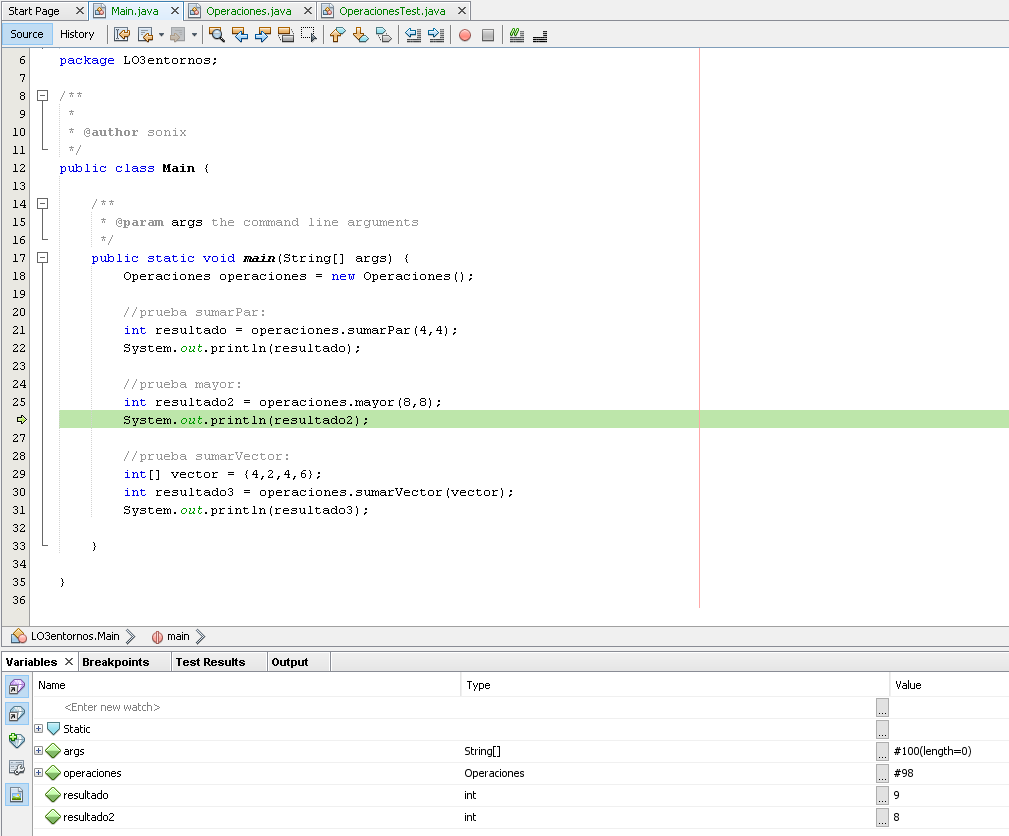


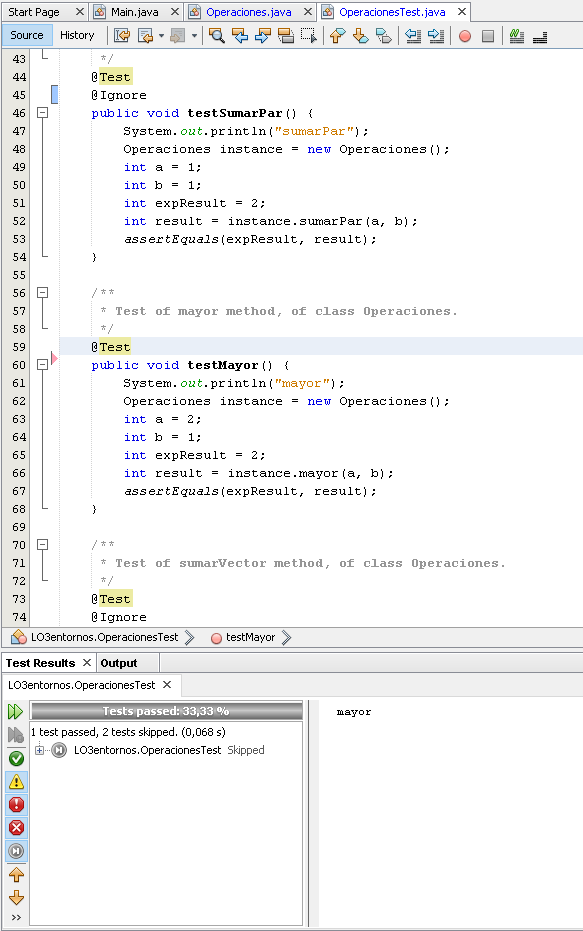
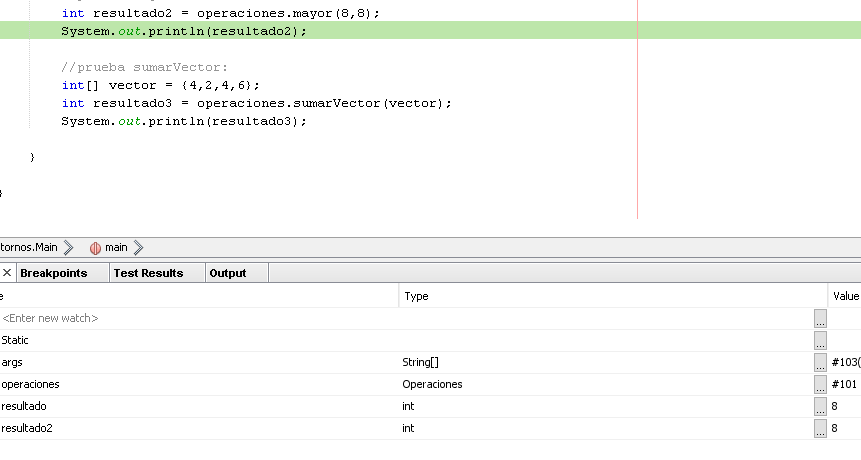
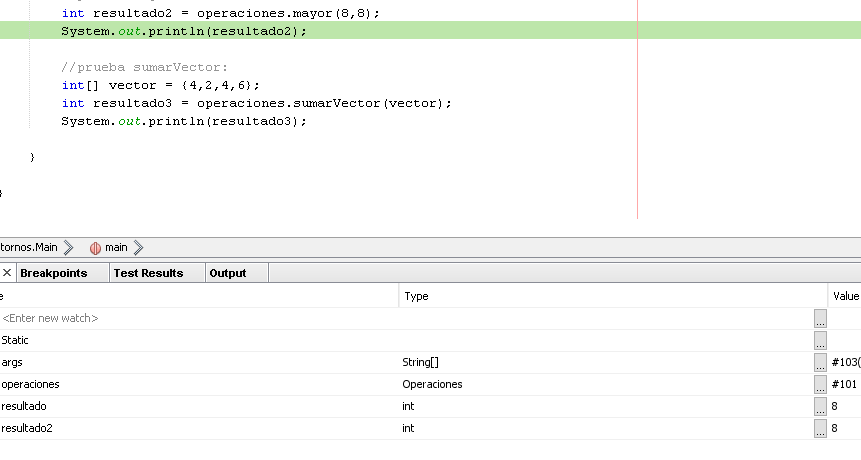
Esta captura fue hecha posteriormente en casa porque no la hice en su momento en clase, error humano, de ahí la temática oscura.

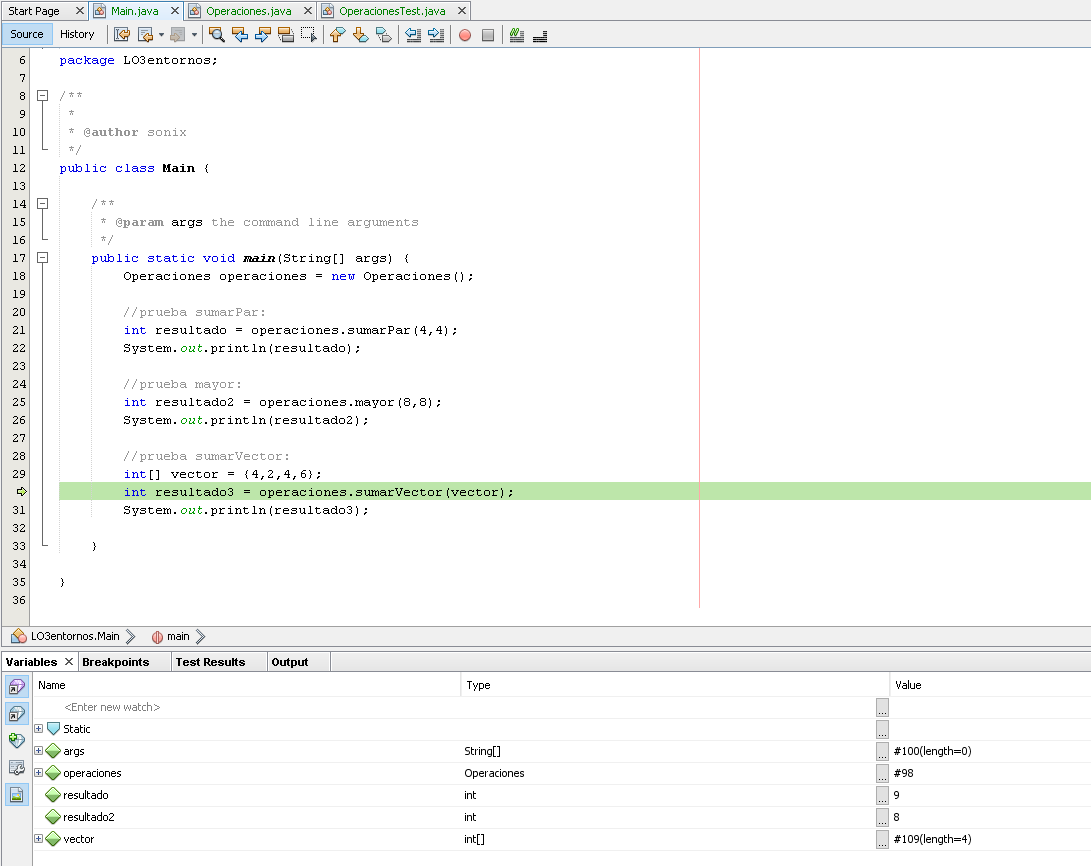


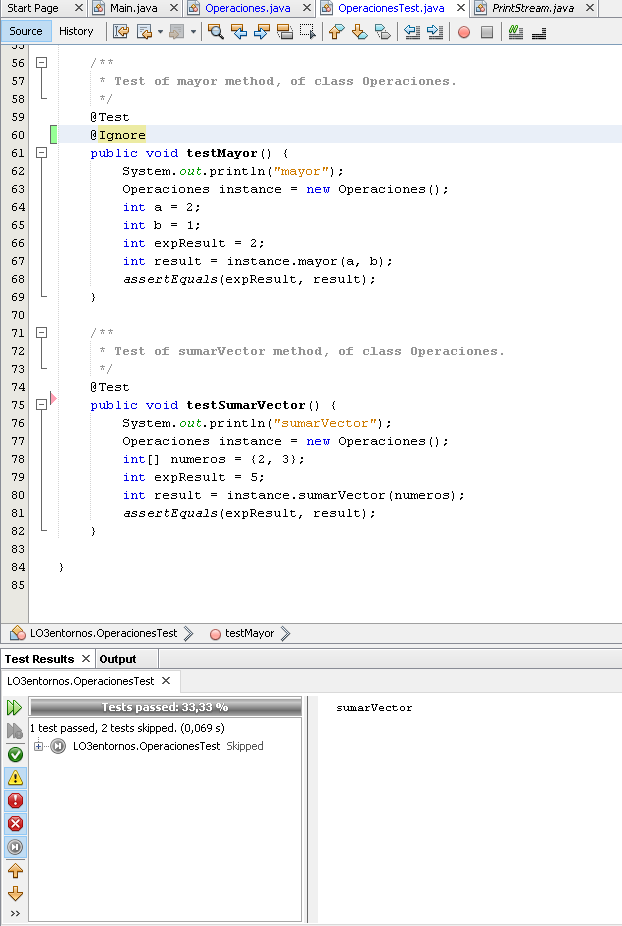


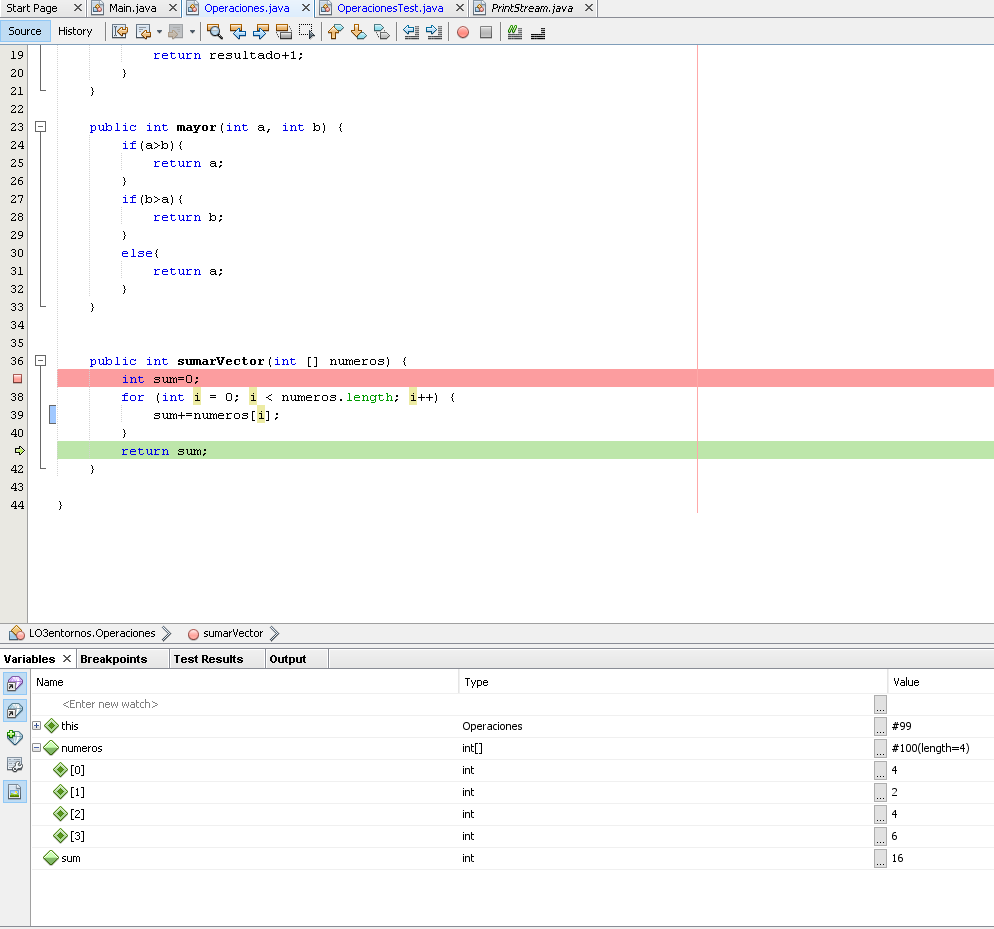












D3

Con los test unitarios, después de cambiar las variables para ver resultados posibles es cuando podemos confirmar que partes del código funcionan, así que considero que podemos decir que sí que podemos evaluar el funcionamiento de la aplicación.