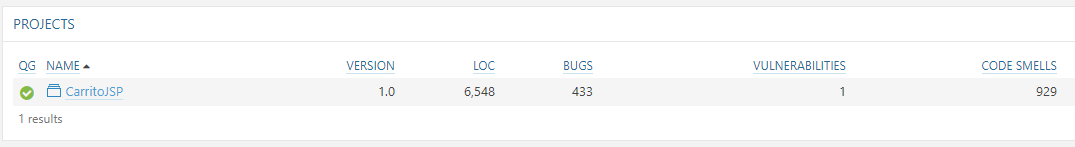
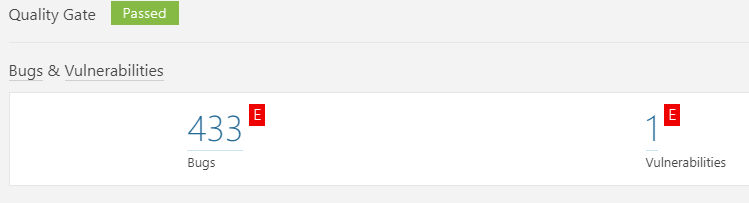
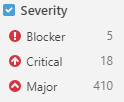
**Informe Prueba asociada a Modificabilidad**

Se evaluó el atributo de calidad de modificabilidad, del proyecto a través de la herramienta SonarQube. Del análisis inicial se obtuvieron los siguientes resultados:



Lo cual indica que, en el proyecto realizado hay 6584 líneas de código, se encontraron 433 bugs y una vulnerabilidad.



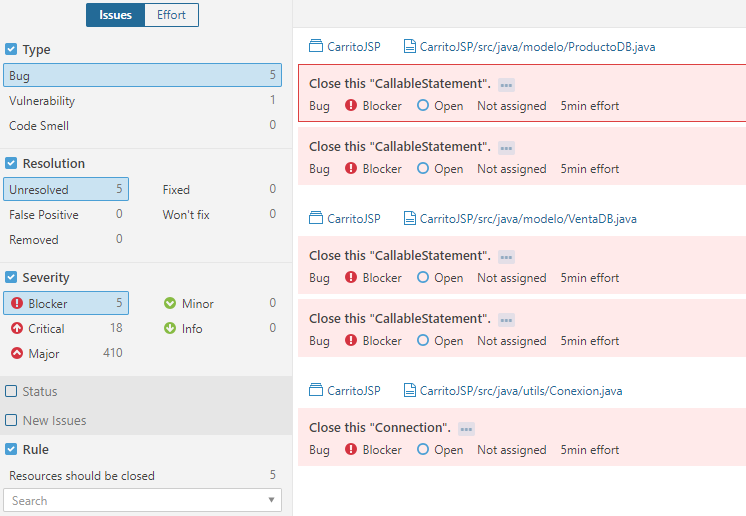
Se considera que el número de Bugs es muy alto, por tanto, es necesario analizar la razón, y cuáles son los bugs presentes. A continuación, se observó que 428 bugs corresponden a las plantillas de js y css utilizadas, y no se consideran bugs necesarios de arreglar inmediatamente, sin embargo 5 bugs si es necesario corregir rápidamente, que son los correspondientes al código java desarrollado.

Algunos de estos bugs no significativos se presentan a continuación.



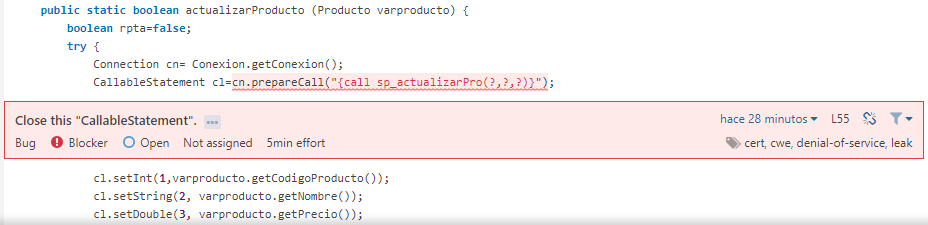


Los bugs que son corregirán son:



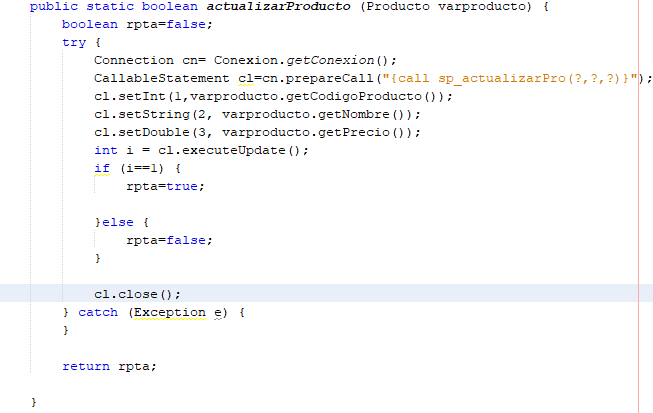
Los cuales corresponden a que los llamados de la base de datos no se cierran.

En el caso de ProductoDB se presenta:

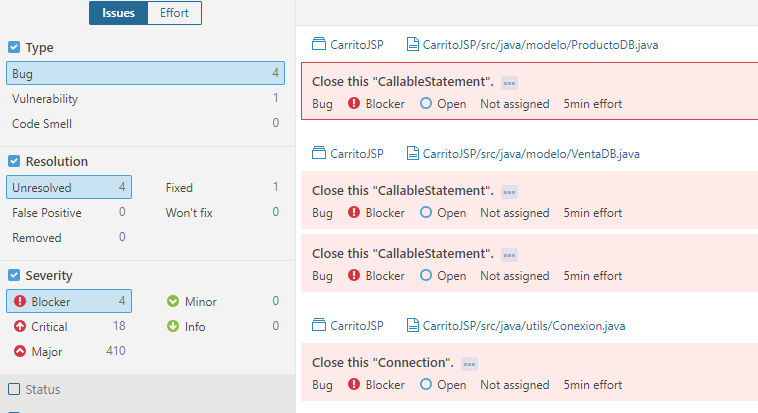


La herramienta informa que no se puede confiar en que las conexiones, secuencias, archivos y otras clases que implementan la interfaz de Closeable o su súper interfaz AutoCloseable, se cierren automáticamente, por lo que se deben cerrar manualmente después de la creación. De lo contrario, se producirá una fuga de recursos.

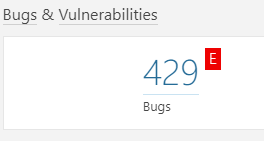
Para solucionar este bug basta con utilizar el método close() luego de realizar y utilizar la conexión.



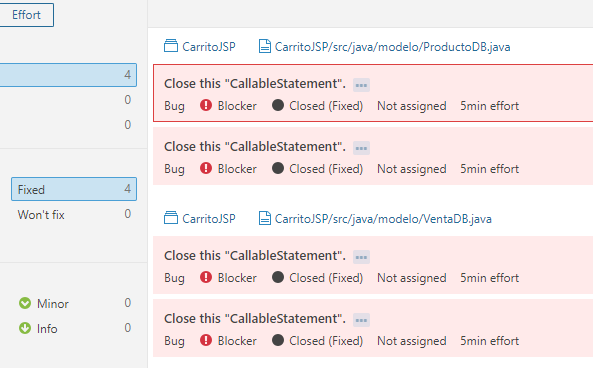
Al ejecutar el sonar-runner una vez modificado el código, se encuentra que si se solucionó el bug efectivamente.

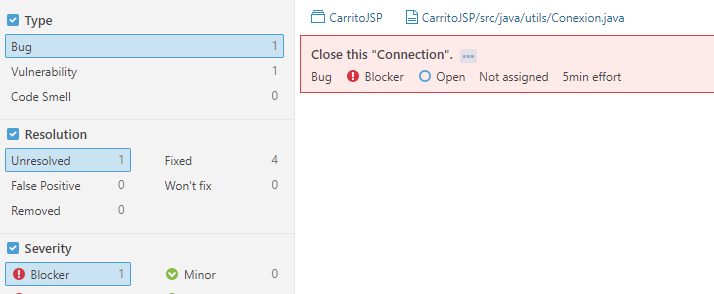


Se realiza el mismo procedimiento para los demás bug de “Close this CallableStatemen”.



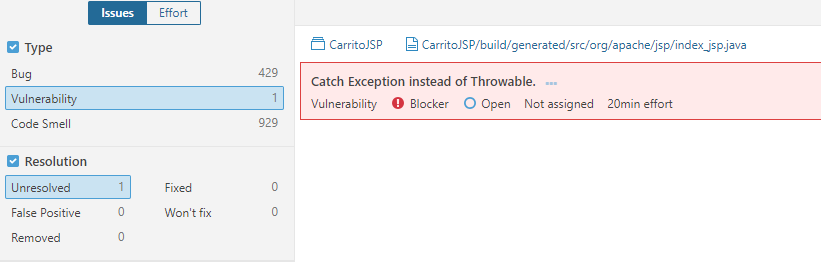
De esta forma se han disminuido el número de bugs de 433 a 429.



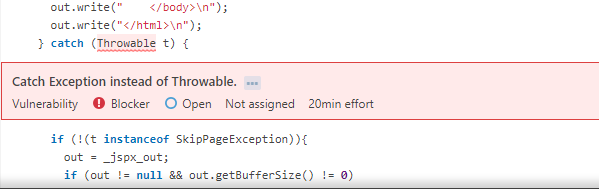
Sin embargo queda un bug, 

En este caso la herramienta muestra que sucede el mismo bug que los anteriormente corregidos, en donde es necesario cerrar la conexión, sin embargo, al ser la clase que permite la conexión no se puede agregar el método close(), sin embargo esto no representa una vulnerabilidad ya que se están cerrando las conexiones que implementan esta clase Conexión.

En cuanto a la vulnerabilidad encontrada, esta corresponde a archivos temporales que crea el servidor apache, tal y como se observa en la ubicación presentada.



La herramienta menciona que la vulnerabilidad encontrada corresponde a que en el catch se captura el error utilizando la superclase Throwable, la cual es la superclase de todos los errores y excepciones en Java, y esto representa una vulnerabilidad ya que también detectará OutOfMemoryError e InternalError, los cuales don errores que una aplicación no debería intentar recuperar.



En comparación con el desarrollo del ciclo I, en este se presentaban los mismos bugs que se corrigieron, no estaban los 428 bugs de las plantillas js y cs, ya que fue en el ciclo 2 donde se desarrolló la interfaz gráfica.

