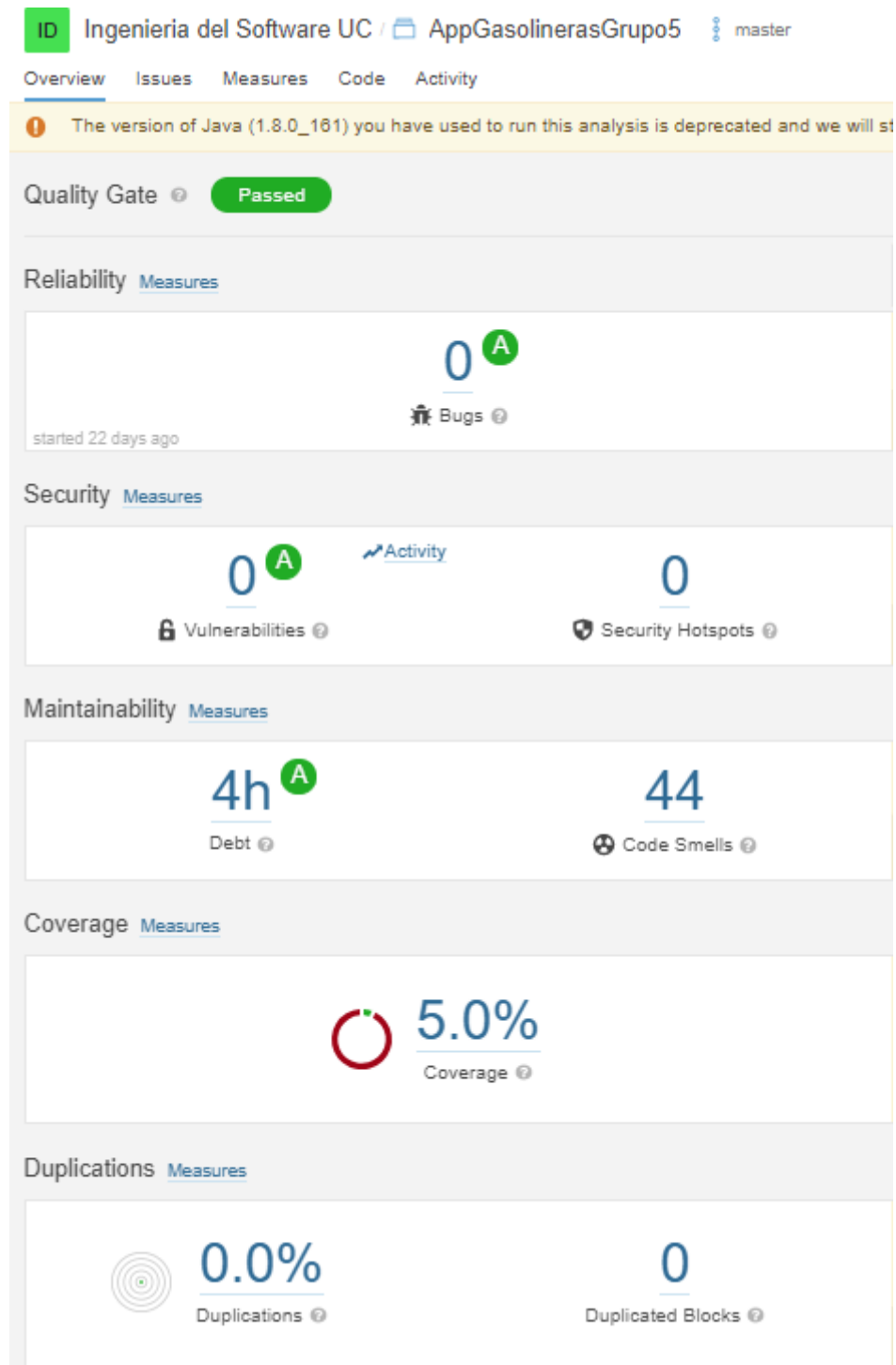


# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

## ANÁLISIS 12 NOVIEMBRE 2020 (Pablo Villegas Fernández)

### CAPTURA



# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

## INCIDENCIAS

Se ha modificado la estructura del proyecto, por lo que se han añadido muchas líneas de código y, al ser el comienzo del sprint, se ha considerado necesario realizar un análisis de calidad del producto.

El análisis pasa los criterios de calidad de la organización debido a que la calificación en confiabilidad (reliability) es A. Sin embargo, la deuda técnica es bastante alta por lo que este análisis se va a centrar en bajar sus horas corrigiendo algunos de los code smells.

En primer lugar, como muestran las imágenes 1 y 2, se ha accedido a la pestaña de “Measures”, en ella encontramos que hay al menos un par de clases que concentran el mayor porcentaje de deuda técnica.

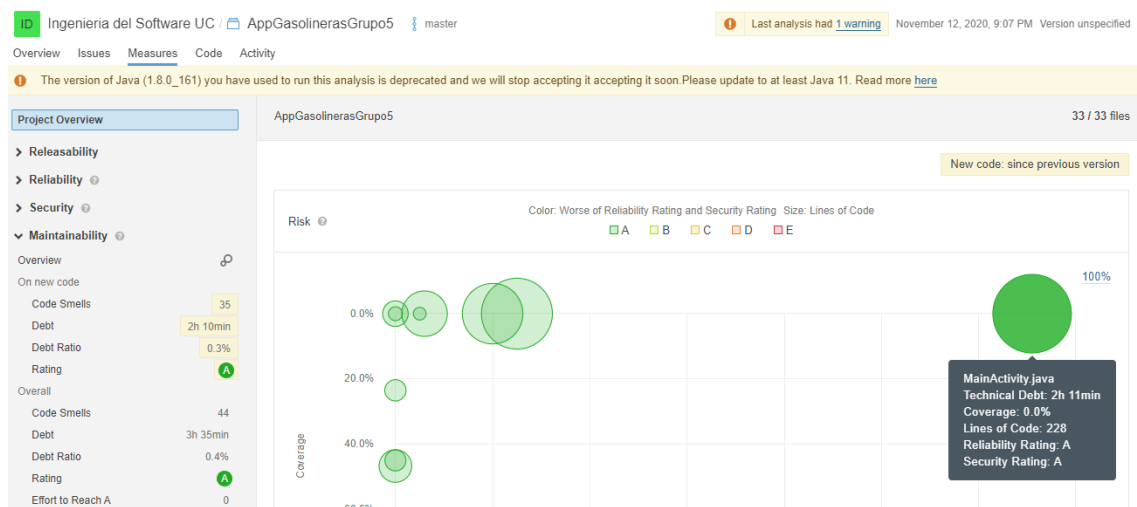


Imagen 1

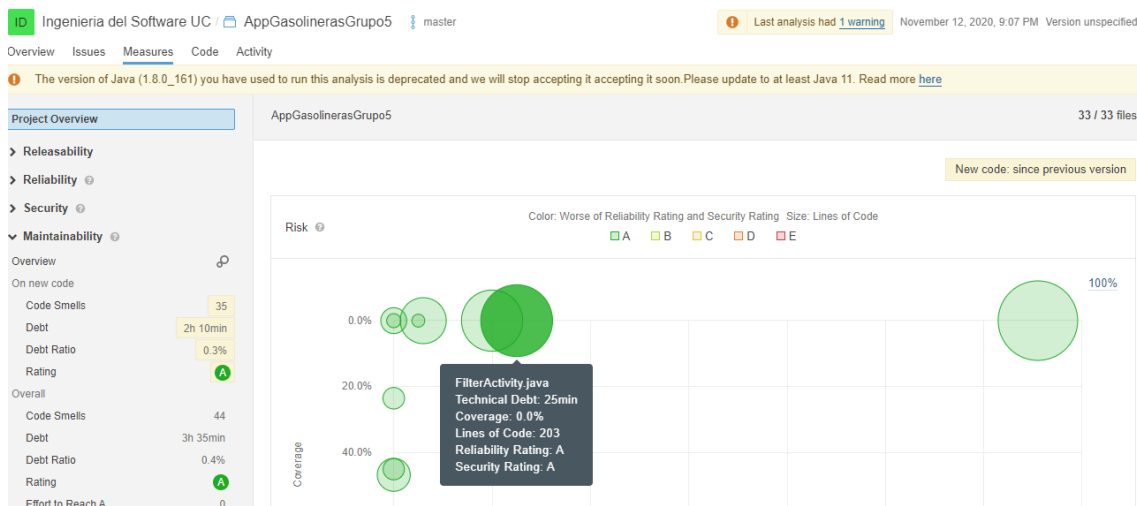


Imagen 2

# Informe de Calidad (Sprint 2)

**Autores:** Roberto González Jiménez, Pablo Villegas Fernández

En la imagen 3, se ha revisado la pestaña Issues y se ha aplicado el filtro “Tipo”, de esta manera comprobamos que hay un error crítico y otros 3 bloqueantes, por lo que nos centraremos en resolverlos.

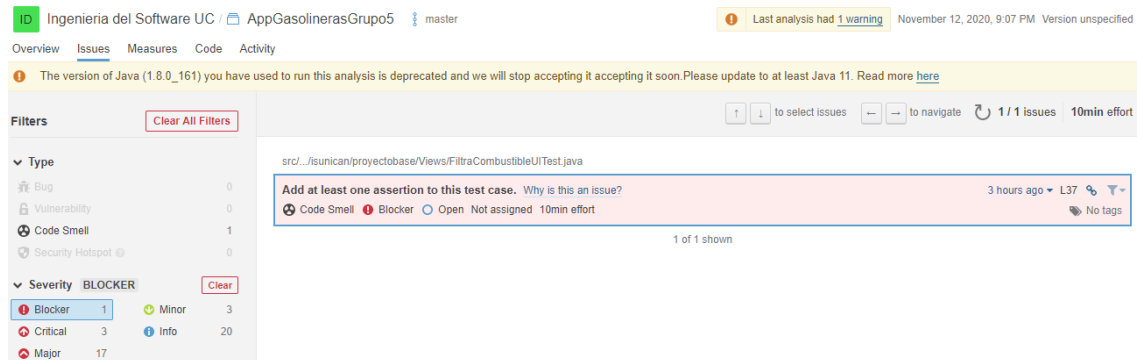


Imagen 3

Por último, en la imagen 4 se aprecia un code smell con una deuda técnica de 20 minutos. Esta corrección se dejará para posteriores análisis de calidad debido a la dificultad.

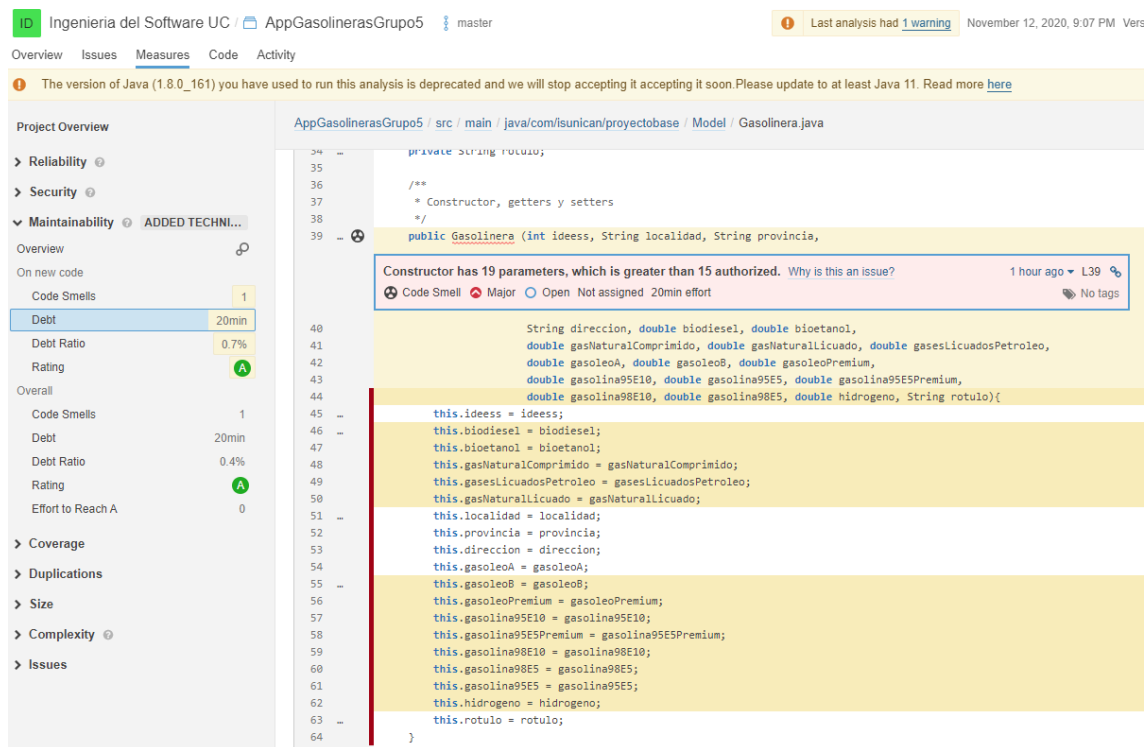


Imagen 4

## PLAN DE ACCIÓN

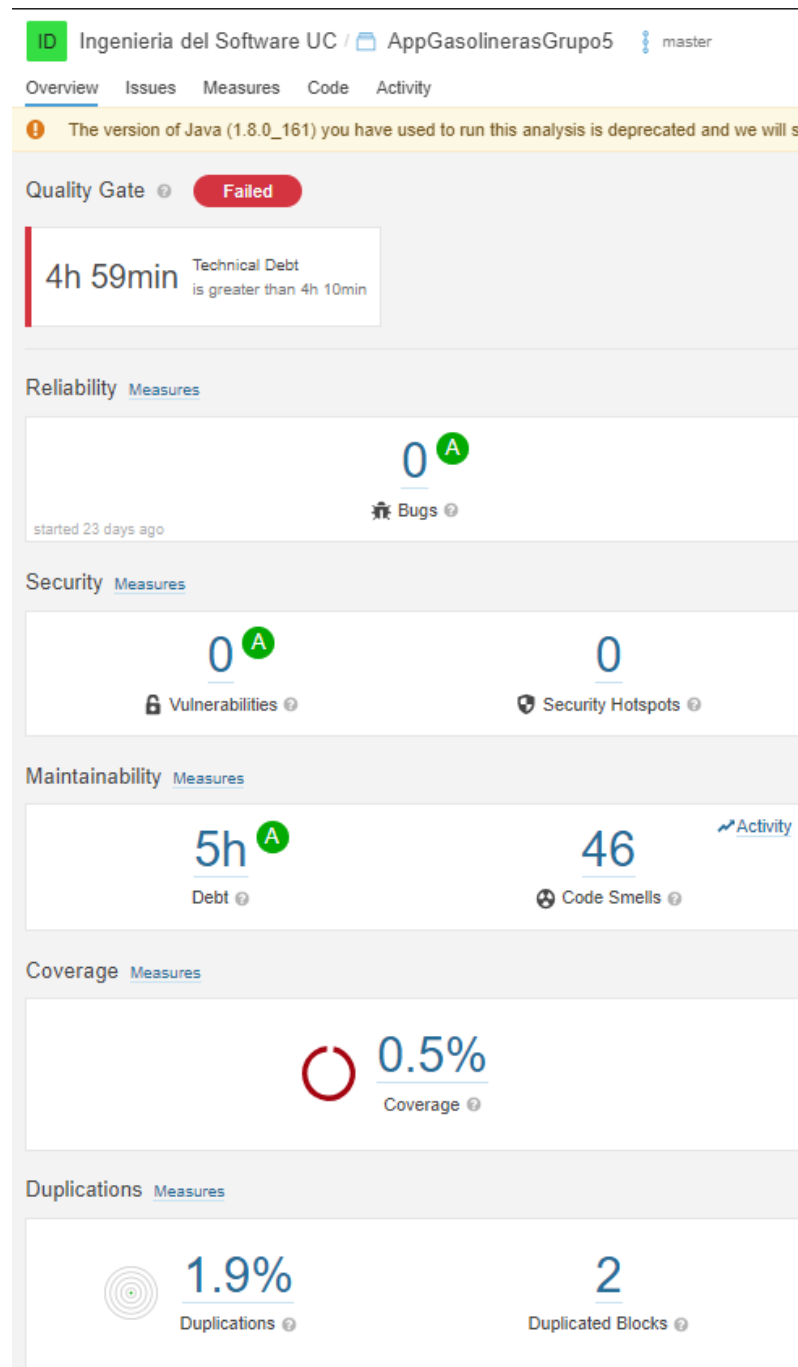
- 1) Arreglar code smells menores encontrados en la pestaña “Measures” para las clases MainActivity y FilterActivity.
- 2) Corregir el issue bloqueante debido a la carencia de asserts en un test de integración.
- 3) Los issue críticos se dan por falta de documentación en métodos vacíos. Se ha documentado por qué lo están.

# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

ANÁLISIS 13 NOVIEMBRE 2020 (Roberto González Jiménez)

## CAPTURA



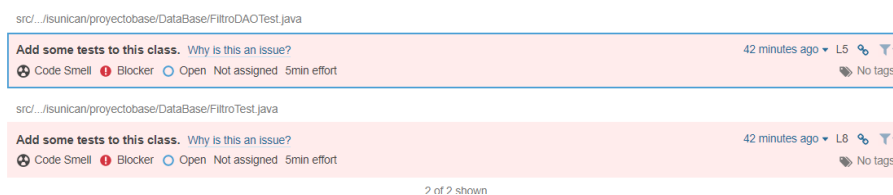
# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

## INCIDENCIAS

El análisis no pasa los criterios de calidad de la organización ya que la deuda técnica es superior a la mínima aceptada, que es 4h 10min, en nuestro caso supone una deuda técnica de 4h 59 min. Esta deuda ha aumentado en 59 minutos respecto al análisis anterior, esto se debe principalmente al aumento de code smells presentes en nuestro código, que ha aumentado de 37 a 46, un total de 9 code smells más respecto al análisis anterior.

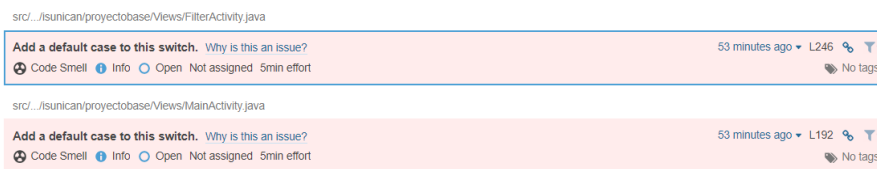
Por otro lado, ha disminuido el parámetro de "Coverage", ya que se han comentado unos cuantos tests para que funcione correctamente el travis, estos test se volverán a comentar al finalizar la implementación y se configurarán para que se adapten a los nuevos casos de la aplicación, esta también es una de las causas de algunas deudas técnicas



En cuanto a los demás aspectos tipo bugs y vulnerabilidades, por ahora nuestra aplicación se conserva limpia, con 0 y una calificación en confiabilidad(reliability) de A en todos los aspectos, solamente habría que reducir los code smells para que pase la quality gate establecida.

## PLAN DE ACCIÓN

- 1) Añadir en los switches de las clases de la actividad del Filtro y la principal un caso default, que no es necesario que haga nada (10 min en total)



- 2) Respecto a las declaraciones de las clases Converters.java y Filtro.java se referencian en algunas ocasiones a ArrayList, que correctamente se deberían referenciar a la clase List, se tiene que añadir también el import a List, también hay unos cuantos errores de este tipo en la clase FilterActivity.java lo cual realizamos los mismos pasos (10x6, 60 minutos en total)



## Informe de Calidad (Sprint 2)

**Autores:** Roberto González Jiménez, Pablo Villegas Fernández

**Comentarios:**

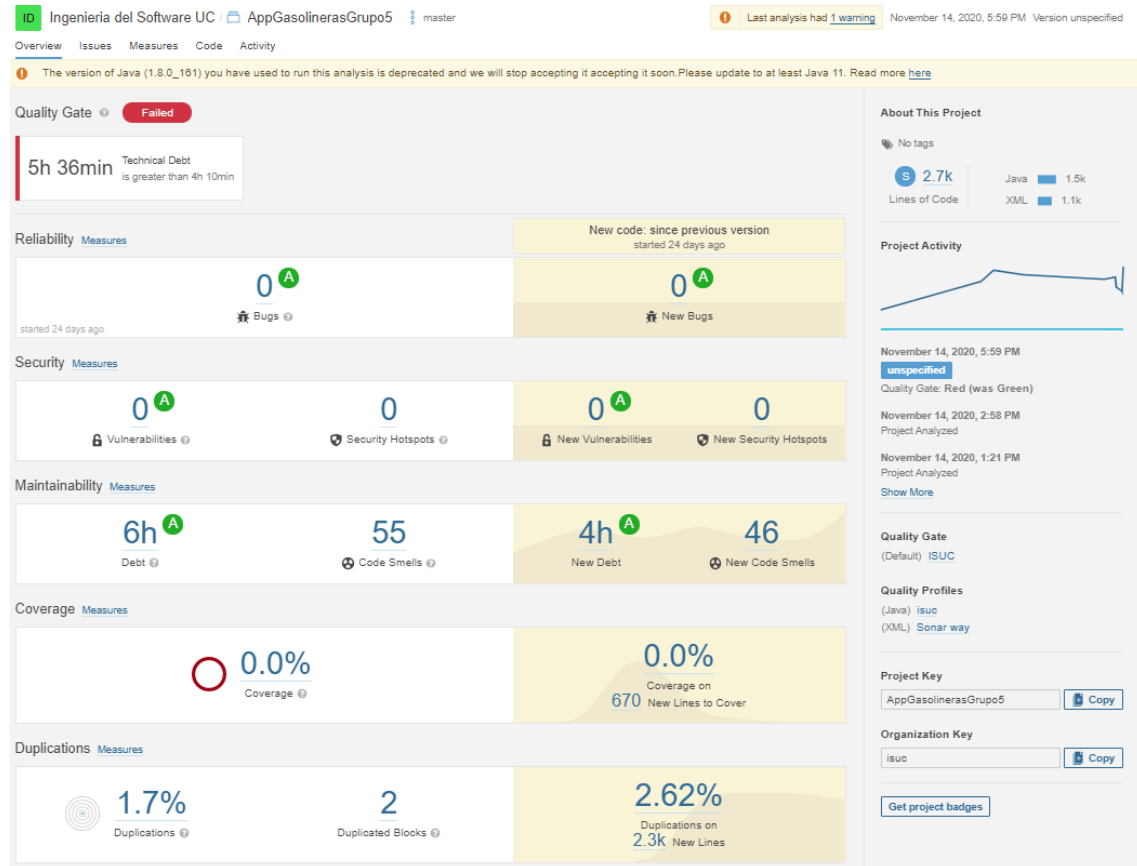
-Con estos 2 tipos de code smells arregladas, ya habríamos cubierto la deuda mínima necesaria para que pase la quality gate puesta, hemos reducido 70 minutos y hacían falta reducir un total de 49 minutos.

# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

ANÁLISIS 14 NOVIEMBRE 2020 (Pablo Villegas Fernández)

## CAPTURA



1. Vista principal de SonarCloud

## INCIDENCIAS

El análisis no pasa los criterios de calidad de la organización debido a que la deuda técnica supera el valor máximo 4h 10min, tal y como se puede observar en la imagen 1.

# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

Al acceder a la pestaña de Issues, nos encontramos 17 code smells críticos, por lo cuál en este análisis de calidad de sonar nos centraremos en depurarlas.

The screenshot shows the SonarQube interface for the project 'Ingeniería del Software UC / AppGasolinerasGrupo5'. The 'Issues' tab is active, displaying 17 critical code smells. The left sidebar shows filters for Type (Bug, Vulnerability, Code Smell, Security Hotspot) and Severity (Critical, Minor, Major, Info). The main area lists 17 issues, all of which are 'Define a constant instead of duplicating this literal' with a severity of 'Critical' and an effort of '8min'.

Type	Severity	Issue Description	Effort	Tags
Code Smell	Critical	Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation. Why is this an issue?	5min	No tags
Code Smell	Critical	Define a constant instead of duplicating this literal "TODOS" 3 times. Why is this an issue?	8min	No tags
Code Smell	Critical	Define a constant instead of duplicating this literal "GASOLEO A" 3 times. Why is this an issue?	8min	No tags
Code Smell	Critical	Define a constant instead of duplicating this literal "GASOLEO B" 3 times. Why is this an issue?	8min	No tags
Code Smell	Critical	Define a constant instead of duplicating this literal "GASOLEO PREMIUM" 3 times. Why is this an issue?	8min	No tags
Code Smell	Critical	Define a constant instead of duplicating this literal "GASOLINA 95 E10" 3 times. Why is this an issue?	8min	No tags
Code Smell	Critical	Define a constant instead of duplicating this literal "GASOLINA 95 E5" 3 times. Why is this an issue?	8min	No tags
Code Smell	Critical	Define a constant instead of duplicating this literal "GASOLINA 95 E5 PREMIUM" 3 times. Why is this an issue?	8min	No tags
Code Smell	Critical	Define a constant instead of duplicating this literal "GASOLINA 98 E10" 3 times. Why is this an issue?	8min	No tags
Code Smell	Critical	Define a constant instead of duplicating this literal "GASOLINA 98 E5" 3 times. Why is this an issue?	8min	No tags

## 2 Pestaña Issues

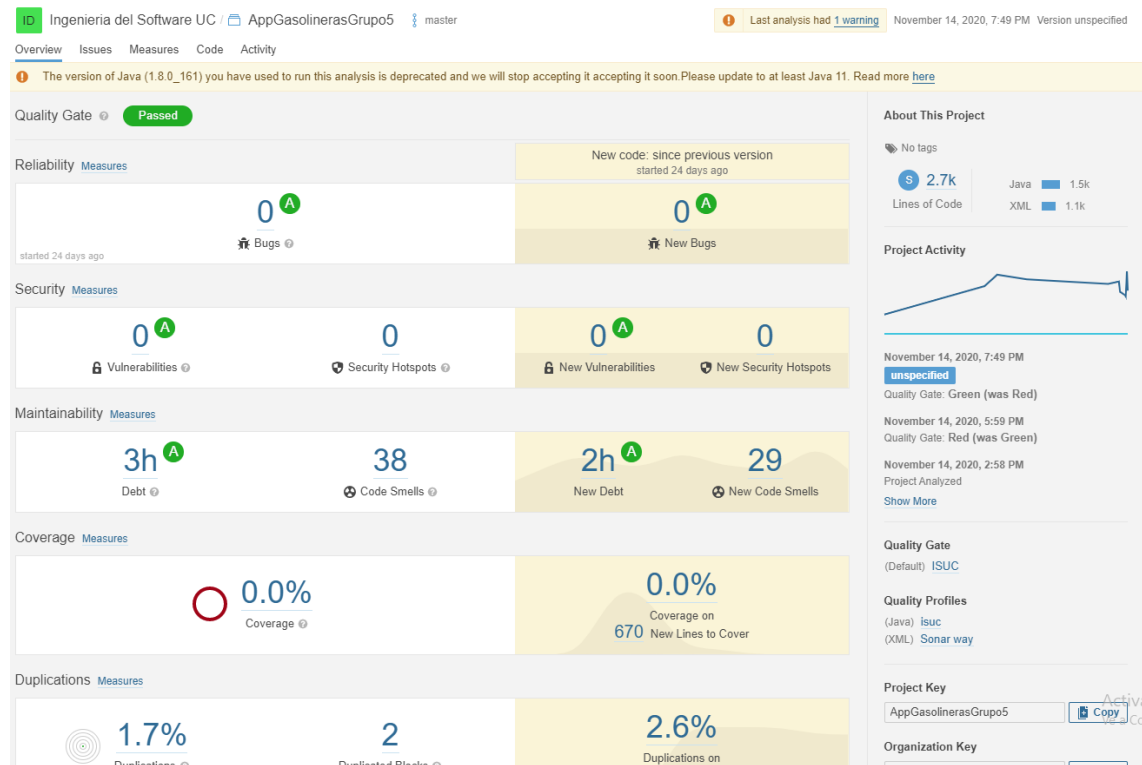
### PLAN DE ACCIÓN

- 1) Arreglar los 17 code smells críticos. Como se puede ver en la imagen 2, los code smells tienen su origen en la creación triplicada de Strings para hacer referencia al mismo nombre. A pesar de que los Strings tienen el mismo contenido, Java los considera objetos diferentes, por lo que existe un desperdicio de memoria al guardarlos y tiempo en crearlos. Como solución aportada, se ha creado una constante de solo lectura por cada String, de tal manera que en las tres llamadas se acceda al mismo String, ahorrando tiempo y espacio.



# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández



3 Resultado tras el análisis

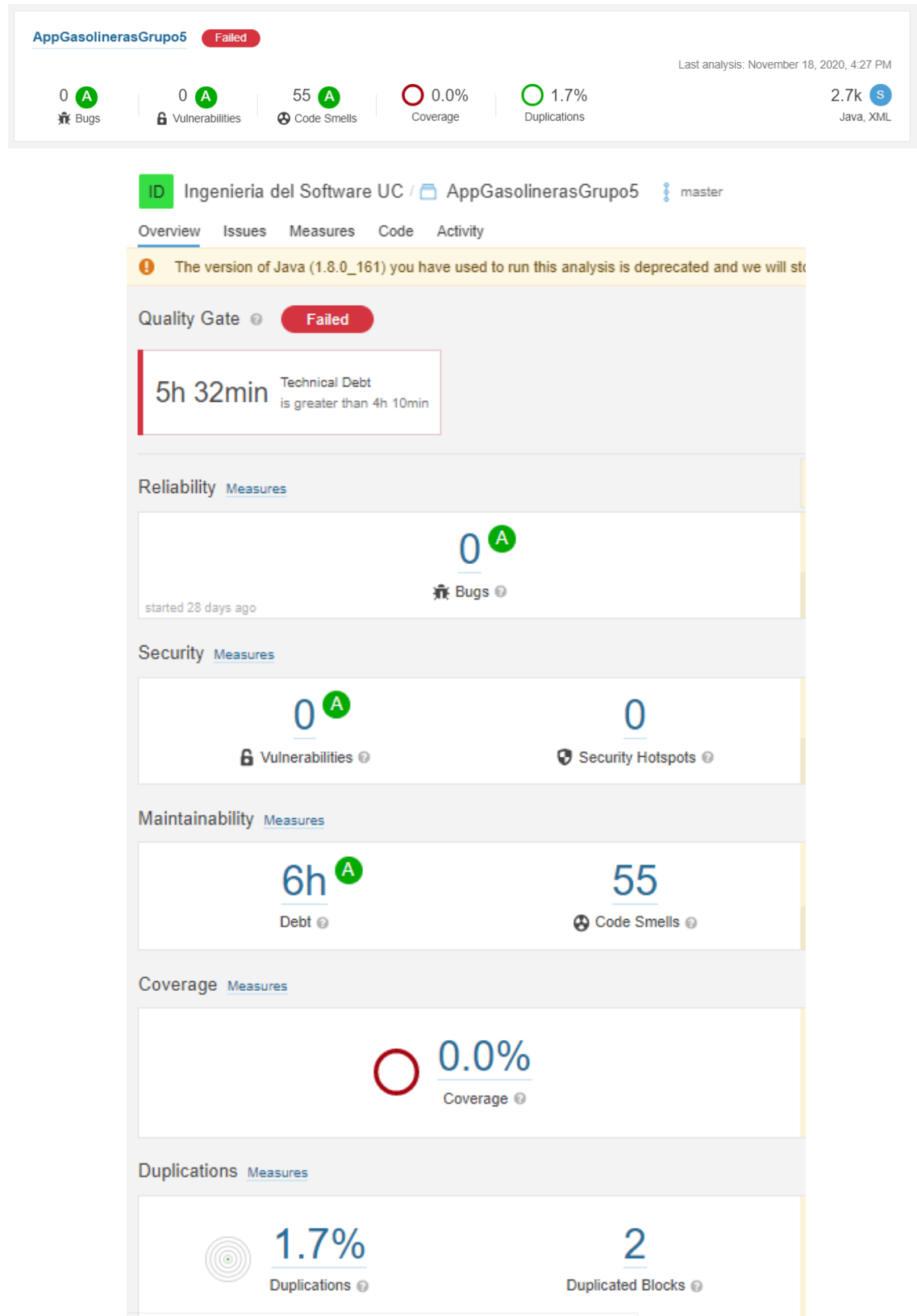
Tras realizar las modificaciones, se ha comprobado que la deuda técnica vuelve a tener un valor por debajo del límite, habiendo quitado alrededor de dos horas de deuda técnica, por lo que se ha considerado como terminado este análisis.

# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

ANÁLISIS 18 NOVIEMBRE 2020 (Roberto González Jiménez)

## CAPTURA



# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

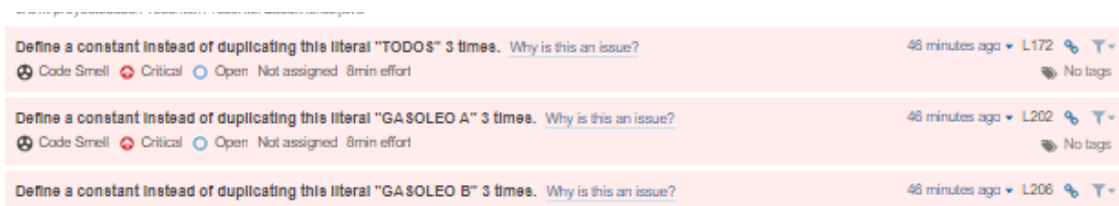
## INCIDENCIAS

Este análisis de calidad corresponde a una fase del proyecto la cual la implementación de las historias de usuarios está prácticamente acabada, menos por un pequeño detalle, al general casi todo el código nuevo, se han encontrado bastantes nuevos code smells por lo tanto no pasa los criterios de calidad de la organización ya que la deuda técnica es superior a la mínima aceptada, que es de 4h 10min, en nuestro caso supone una deuda técnica de 5h 32min. Esta deuda es bastante parecida a la del anterior análisis, ya que curiosamente tienen el mismo número de code smells, 55.

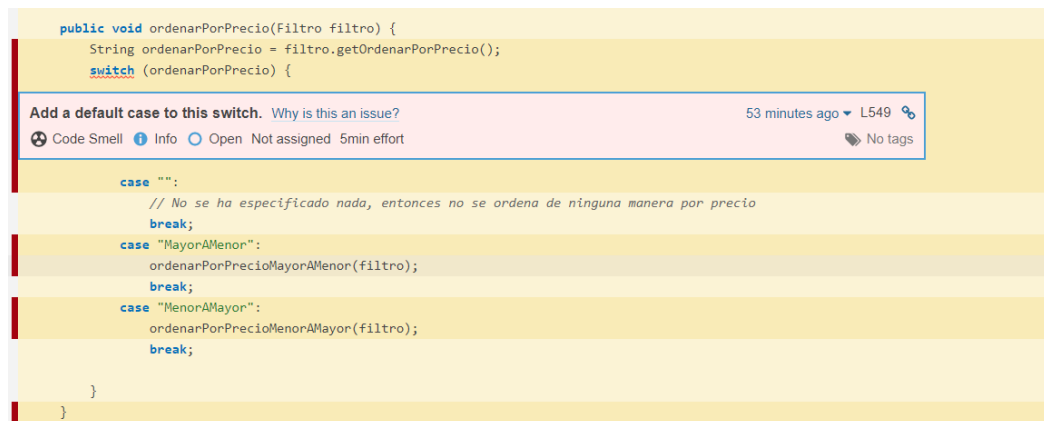
En cuanto a los demás parámetros del análisis Sonar, se conservan igual, en los próximos análisis ya incluiremos las clases test que se añadirán y que se han desactivado parcialmente.

## PLAN DE ACCIÓN

- 1) Cambiar los Strings de todos los tipos de combustible a una constante, ya que se repite 3 veces, mismo fallo que en el análisis anterior pero como se han refactorizado los métodos a la clase presenter vuelve a ocurrir en mismo fallo (15x8, 120 min en total).



- 2) Añadir un default a un switch del método de ordenar por precio (5 min).

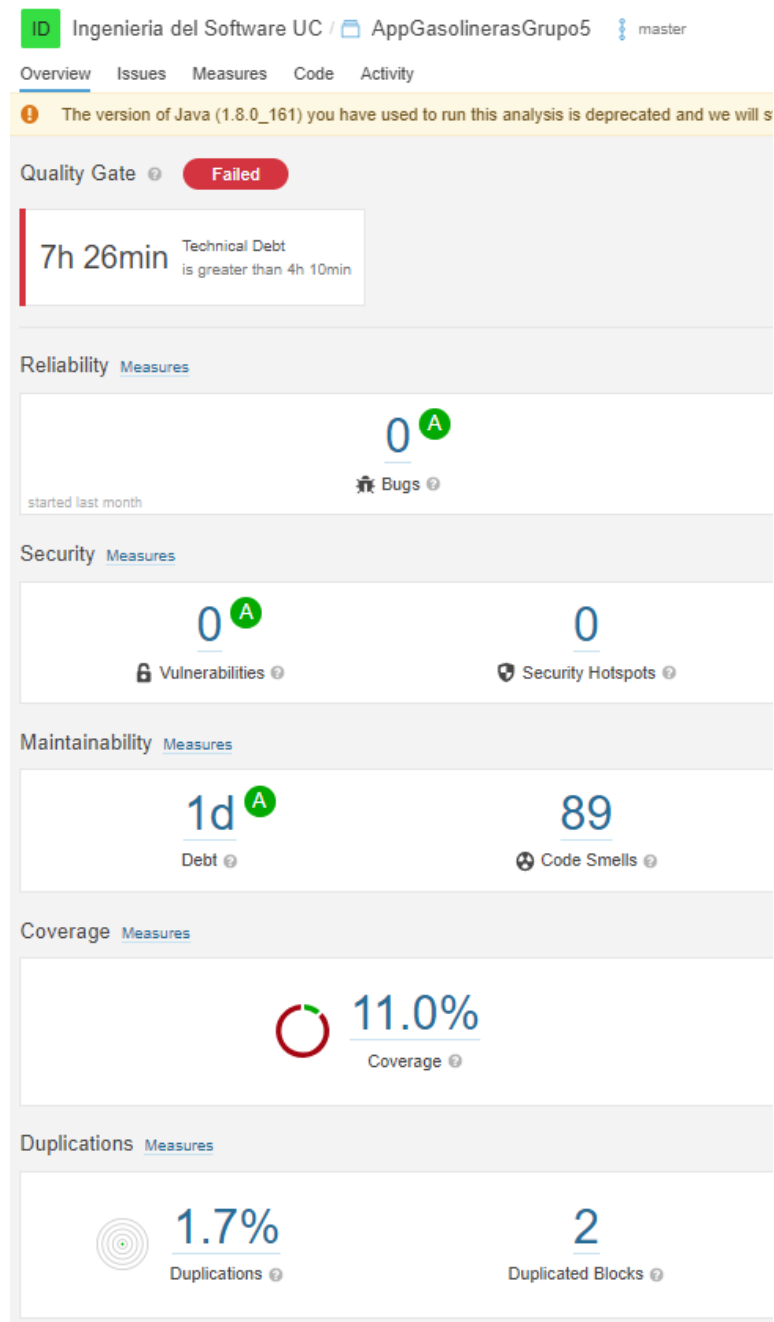
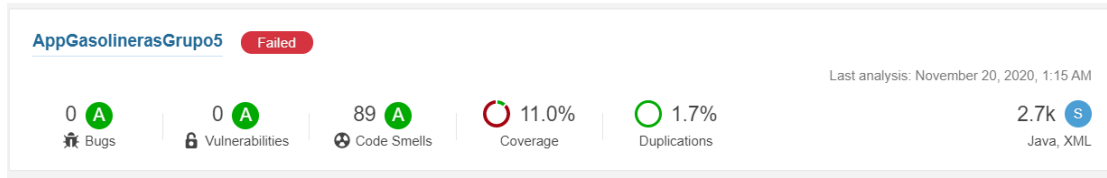


# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

ANÁLISIS 20 NOVIEMBRE 2020 (Roberto González Jiménez)

## CAPTURA



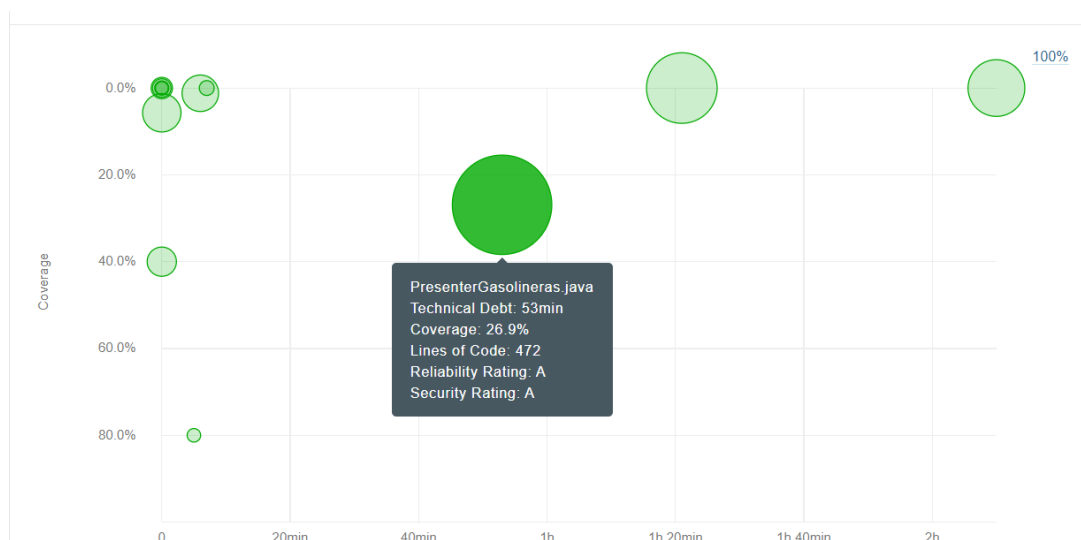
# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

## INCIDENCIAS

Este análisis de calidad corresponde a una fase del proyecto la cual está casi prácticamente finalizada, se han realizado todos los tests menos alguno que da fallo que se corregirá más adelante, se han encontrado bastantes nuevos code smells por lo tanto no pasa los criterios de calidad de la organización ya que la deuda técnica es superior a la mínima aceptada, que es de 4h 10min, en nuestro caso supone una deuda técnica de 7h 26min. Esta deuda es bastante más alta que la del análisis anterior, esto se debe a que se han implementado los tests y contienen bastantes errores que tendremos que corregir para reducir la deuda técnica.

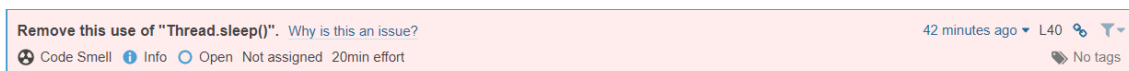
En cuanto a los demás parámetros del análisis Sonar, se conservan igual que siempre, todo en A, a excepción del coverage, ha subido al 11% debido a los tests, antes como no estaban implementados estaba a 0.



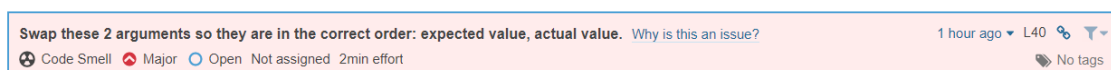
Podemos ver que se ha distribuido bastante la deuda técnica, ahora en el presenter hay más líneas de códigos a ser revisadas

## PLAN DE ACCIÓN

- 1) Cambiar el `Thread.sleep()`, ya que sonar dice que puede dar problemas, por `SystemClock.sleep()` (6x20, 120 min en total).



- 2) Cambiar el orden del `assertEquals` para que se muestre el valor que se espera true, no el actual (2x9, 18 min en total).



- 3) Añadir al menos una assertion a las clases de prueba (2x10, 20 min en total).

# Informe de Calidad (Sprint 2)

Autores: Roberto González Jiménez, Pablo Villegas Fernández

Add at least one assertion to this test case. [Why is this an issue?](#)

1 hour ago ▾ L31 🔗

🚫 Code Smell 🛑 Blocker 🔵 Open Not assigned 10min effort

🏷 No tags

- 4) Especificar que devuelva una lista de gasolineras (5 min).

```
^ /  
public static List readArrayGasolineras (JsonReader reader) throws IOException {
```

Provide the parametrized type for this generic. [Why is this an issue?](#)

last month ▾ L64 🔗

🚫 Code Smell 🛑 Major 🔵 Open Not assigned 5min effort

🏷 No tags

- 5) Quitar imports sin usar (2x5, 10 min en total).
- 6) Quitar returns que había sin hacer ninguna función(1x28, 28 min en total).