INFORME DE CALIDAD DE PRODUCTO

Autor	Alexandru Guzun
Número de Sprint	Sprint 001
Historia de Usuario	Vista Detalle
Fecha	26/10/16

Introducción.

El objetivo de la actividad es realizar un análisis de calidad interna de producto software mediante la herramienta SonarQube.

Este análisis será aplicado al proyecto “AppGasolineras”, sobre la historia de usuario “Vista Detalle”, en el cual se aplicarán las medidas convenientes para mejorar y/o maximizar la calidad del proyecto.

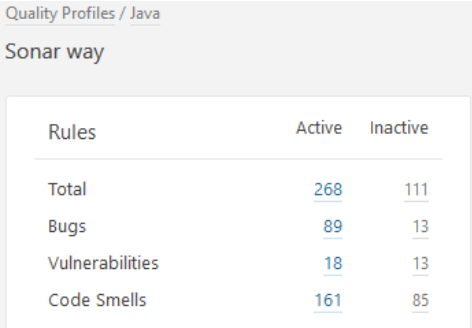
Análisis

Para el primer análisis se procederá con la configuración por defecto de SonarQube en cuanto a Quality Gate y Quality Profile.

Quality Gate:

Metric	Over Leak Period	Operator	Warning	Error
Coverage on New Code	Always	is less than		80
New Bugs	Always	is greater than		0
New Vulnerabilities	Always	is greater than		0
Technical Debt Ratio on New Code	Always	is greater than		5

Quality Profile:



Para evitar fallos no controlados sobre clases generadas automáticamente del proyecto, la ruta sobre la que se aplicará el escáner será solo en el contenido de la carpeta “src” del proyecto.

Fijándonos primero en la calificación SQALE, daremos prioridad a las medidas cuya calificación sea D o peor, ya que eso quiere decir que estamos ante un ratio de deuda técnica superior al 50%.

Una vez nos hemos fijado en la calificación SQALE, y es C o mejor, nos centraremos en otros aspectos.

Los aspectos a los cuales se les darán más importancia a la hora de analizar serán los bugs y vulnerabilidades con una severidad “Blocker”, “Critical” o “Major”, en ese orden de importancia.

Después procederemos con los “Code Smell”, dando el mismo orden de importancia que con los bugs según su severidad.

En caso de existir duplicaciones de código, se revisarán a continuación.

Una vez revisados estos apartados, si el proyecto dispone de tiempo sobrante se realizarán las correcciones a la complejidad del código, a la documentación y a los bugs, vulnerabilidades y “Code Smell” poco severos en ese orden.

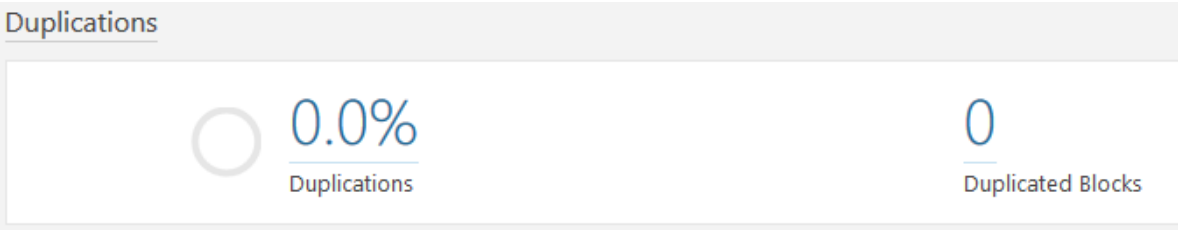
Resultados del análisis

El proyecto ha pasado el Quality Gate, pero en este momento, al ser la primera versión, no es algo que debamos tener muy en cuenta, ya que el Quality Gate aplicado se centra sobretodo en los nuevos fallos respecto a una versión anterior.

En cuanto a Bugs y Vulnerabilidades, pese a que parecen pocos, la calificación SQALE es muy mala, por lo que deberemos centrarnos en mejorarlo.
Si nos fijamos en el esfuerzo necesario, vemos que además podemos solucionarlos rápidamente.

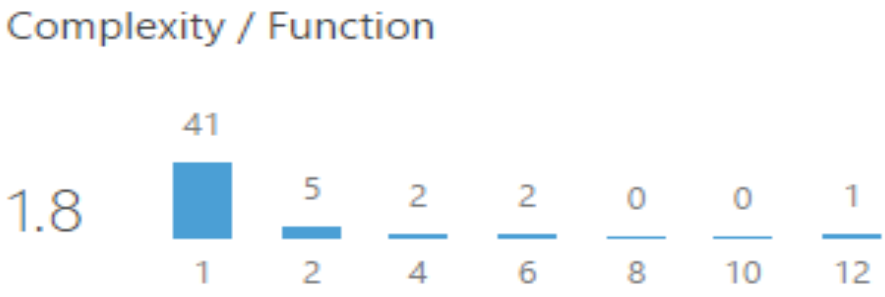
Las medidas de mantenimiento son muy buenas, recibiendo una calificación A.
Aún así, debido a que son muchos Code Smells, la deuda técnica sube a 1 dia, pero no llega a ser un 5%, por lo tanto será algo secundario a la hora de mejorarlo.

El código del proyecto está completamente libre de duplicaciones, no requiere medidas correctoras.

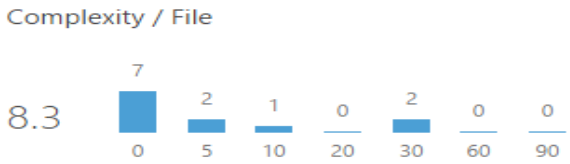


Fijándonos en el tamaño del proyecto, vemos que el número de ficheros y de clases es muy parecido, de forma que no estamos sobrecargando cada fichero.

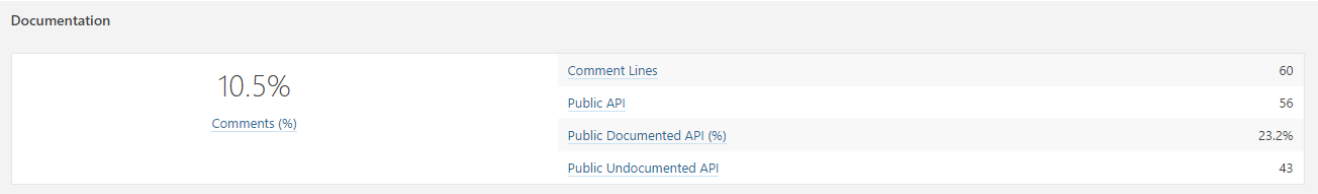
es buena, ya que la mayoría tienen una complejidad muy simple y solo hay uno que suba un poco, pero no de forma excesiva.



Si nos fijamos en la complejidad de los ficheros si que aumenta un poco, ya que hay dos ficheros con una complejidad superior, que son los que desvían la media. Pero son dos ficheros importantes que no podemos modificar demasiado sin entorpecer el proyecto, estos ficheros son la clase del objeto principal que manejamos, "Gasolinera", y la clase que maneja los datos recibidos del servicio web.



La complejidad en torno a las clases es prácticamente igual que la de los ficheros, ya que la relación clases por fichero es casi 1 a 1.
La documentación es muy baja, solo un 10.5% del proyecto esta comentado. No es algo prioritario, considerando que se cuentan las clases generadas automáticamente, pero convendría aumentar la cantidad de documentación, sobretodo en las clases cuya complejidad hemos visto que es mayor que las demás.



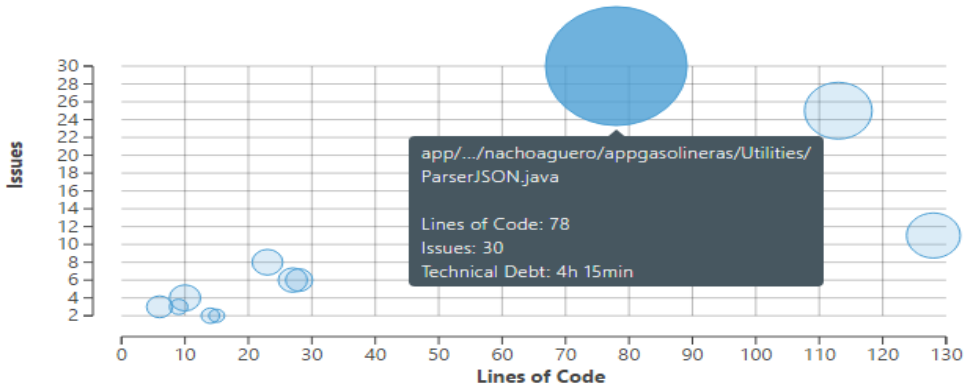
Plan de mejora

Para mejorar el proyecto software se seguirán los siguientes pasos, teniendo en cuenta la gestión previa al análisis realizada al principio del documento y observando los resultados obtenidos:

1. Se procederá a corregir los bugs y vulnerabilidades que han resultado en una calificación SQALE D o peor.

2. Se documentará de forma más completa las clases con alta complejidad.

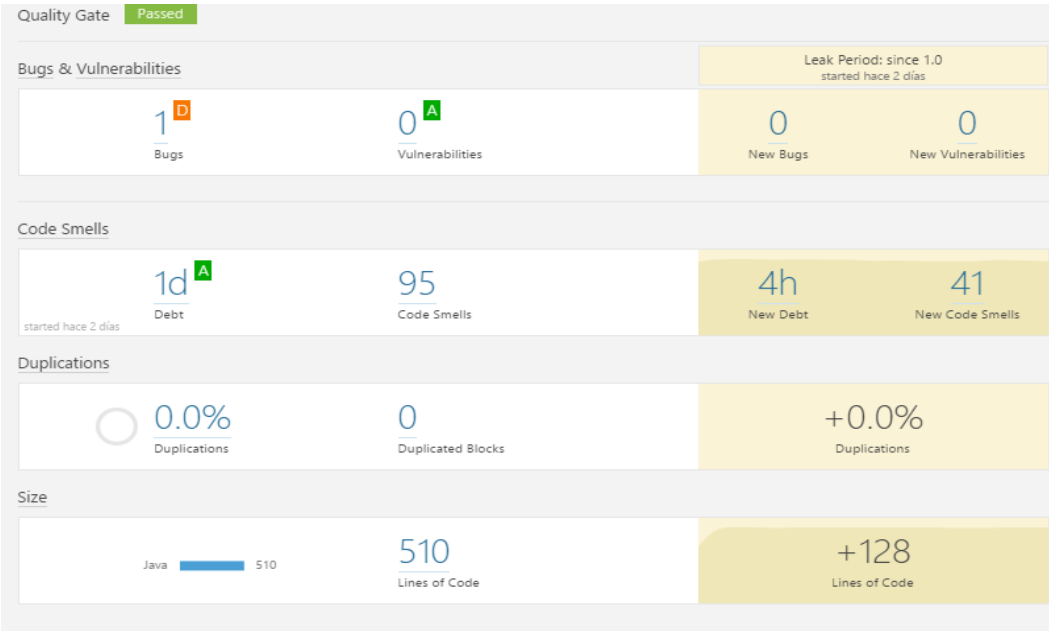
3. Se intentará reducir la deuda técnica atendiendo a mayor severidad de los fallos y después a mayor número de incidencias por fichero, como en el ejemplo de la imagen.



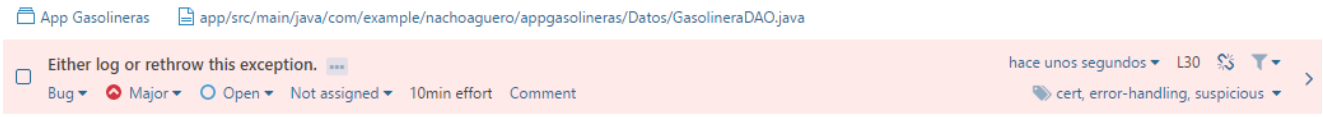
4. Se reducirá en lo posible la deuda técnica producida por el resto de “Code Smells”.

Evolución del proyecto

Los bugs han conseguido reducirse de 7 a 1 y las Vulnerabilidades de 5 a 0, obteniendo los siguientes datos:



Bugs corregidos :

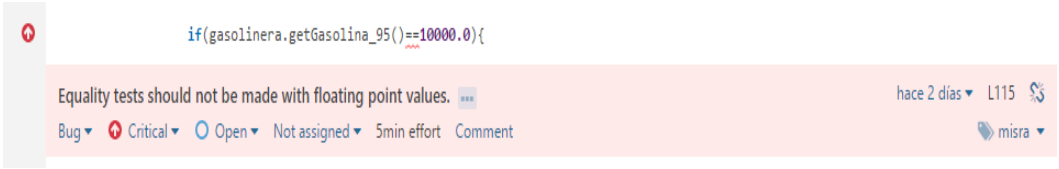


Se soluciona sustituyendo las siguientes líneas de código:

```
Log.e("ERROR","Error en la obtención de gasolineras: "+e.getMessage());

e.printStackTrace();
```

Por :

```
Log.e("ERROR","Error en la obtención de gasolineras: ",e);
```

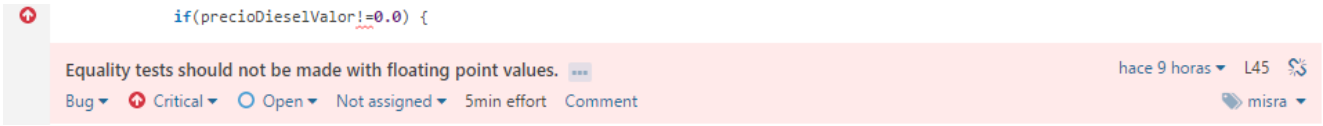


Se soluciona sustituyendo las siguientes líneas de código:

```
  if(gasolinera.getGasolina_95()==10000.0){

    gasolina.setText("Gasolina: "+"No disponible");

}
```

Por :

```
if(Double.compare(gasolinera.getGasolina_95(), 10000.0)==0){

    gasolina.setText("Gasolina: "+"No disponible");

}
```

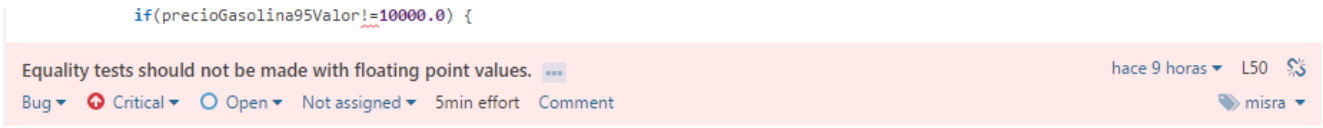


Se soluciona sustituyendo las siguientes líneas de código:

```
if(precioDieselValor!=0.0)
```

Por :

```
if(Double.compare(precioDieselValor,0.0)!=0)
```

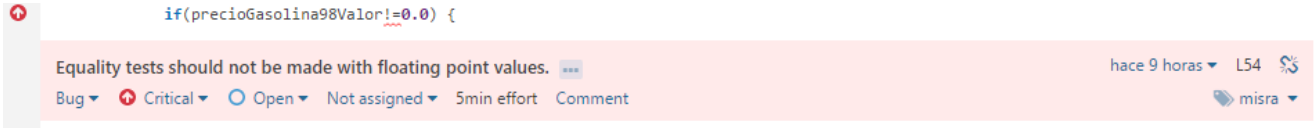


Se soluciona sustituyendo las siguientes líneas de código:

```
if(precioGasolina95Valor!=10000.0)
```

Por :

```
if(Double.compare(precioGasolina95Valor,0.0)!=0)
```

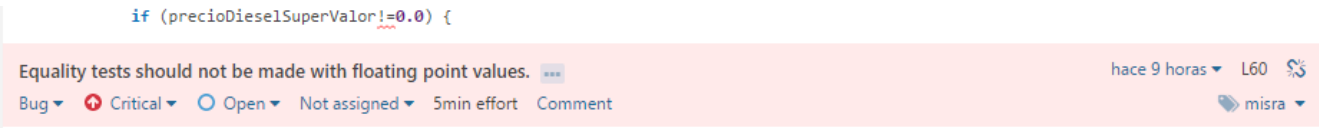


Se soluciona sustituyendo las siguientes líneas de código:

```
if(precioGasolina98Valor!=0.0)
```

Por :

```
if(Double.compare(precioGasolina98Valor,0.0)!=0)
```



Se soluciona sustituyendo las siguientes líneas de código:

```
if (precioDieselSuperValor!=0.0)
```

Por :

```
if(Double.compare(precioDieselSuperValor,0.0)!=0)
```