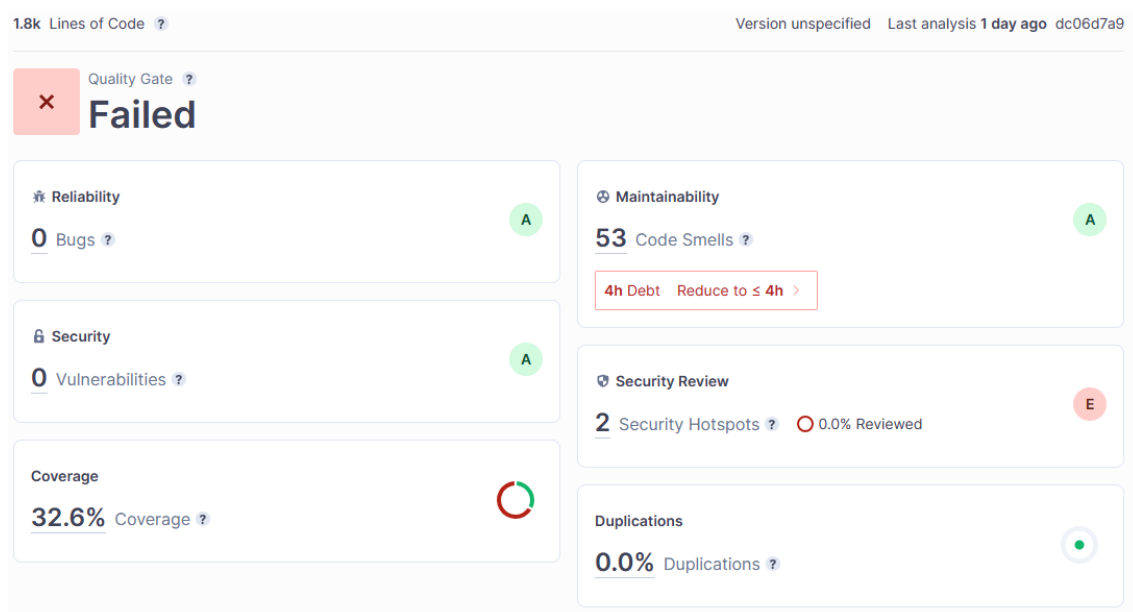


## Informe de Calidad Sprint 2: 27/10/2023



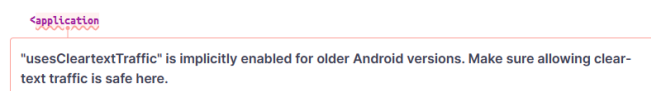
El quality gate del proyecto no pasa debido a la deuda de tiempo mayor de 4 horas, debido a los 53 code smells, de los cuales principalmente se concentran en estas clases del proyecto:

Technical Debt 4h 19min [See history](#)

src/main/java/es/unican/carchargers/repository/service/FakeCall.java	55min
src/main/java/es/unican/carchargers/activities/main/ChargersArrayAdapter.java	46min
src/main/java/es/unican/carchargers/activities/main/MainView.java	25min
src/main/java/es/unican/carchargers/activities/details/DetailsView.java	20min
src/main/java/es/unican/carchargers/constants/EConnectionType.java	14min
src/main/java/es/unican/carchargers/model/Charger.java	13min
src/test/java/es/unican/carchargers/activities/main/MainPresenterTest.java	12min
src/main/java/es/unican/carchargers/repository/service/APIFactory.java	10min
src/main/java/es/unican/carchargers/activities/main/MainPresenter.java	10min
src/main/java/es/unican/carchargers/common/Utils.java	10min
src/main/java/es/unican/carchargers/repository/service/OpenChargeMapService.java	9min
src/test/java/es/unican/carchargers/model/ChargerTest.java	6min

Por otra parte, la cobertura es de un 32.6%, no del todo completa debido a que no se probarán todos los métodos desarrollados según la definición de completado del proyecto integrado.

Finalmente, cabe destacar que los 2 fallos de seguridad se deben al desarrollo previamente entregado al grupo de trabajo. Estos se deben a la declaración de un atributo con un nombre declarativo muy comprometido como es “API\_KEY” en la clase OpenChargeMapService y en el fichero AndroidManifest.xml:



Centrándonos en el principal gasto de tiempo, tendremos en cuenta varios aspectos:

- Los fallos de mantenibilidad que se refieran a cambios de nombres de atributos serán descartados por simplicidad.
- Si detectamos fallos que se refieren al parámetros, funciones, TODOs, etc. Que no cumplan la normativa de calidad analizada por SonarCloud, serán obviados ya que forman parte del desarrollo de futuras funcionalidades que no contendrán fallos de ningún tipo, como por ejemplo parámetros de funciones sin utilizar(por el momento).

Los code smells a tratar son los siguientes:

Clase	Severidad	Tipo	Deuda Técnica	Descripción
/model/StatusType.java	Alta	Mantenibilidad (Intencional)	5min	Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.
/repository/service/FakeCall.java	Alta	Mantenibilidad (Intencional)	5min	Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.
/model/StatusType.java	Alta	Mantenibilidad (Intencional)	5min	Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.
/constants/EConnectionType.java	Alta	Mantenibilidad (Intencional)	6x2min (12min)	Suponen 6 cambios de nombres según el convenio de nombramiento de las variables estáticas.
/main/MainView.java	Media	Fiabilidad, Mantenibilidad (Consistencia)	5 min	Remove this field injection and use constructor injection instead.
/common/AndoridUtils.java	Media	Mantenibilidad (Intencional)	5min	Add a private constructor to hide the implicit public one.
/common/RepositoriesModule.java	Media	Mantenibilidad (Intencional)	5min	Add a private constructor to hide the implicit public one.
/common/Utils.java	Media	Mantenibilidad (Intencional)	5min	Add a private constructor to hide the implicit public one.
/details/DetailsView.java	Baja	Mantenibilidad (Consistencia)	15min	Remove this use of "getParcelable"; it is deprecated.
/repository/service/FakeCall.java	Baja	Mantenibilidad (Intencional)	20min	Use super.clone() to create and seed the cloned instance to be returned.

En resumen, la mayoría de las notificaciones recibidas por parte de SonarCloud son de mantenibilidad, de los cuales muchos son peticiones de cambio de nombre de atributos que, en nuestro contexto de trabajo actual, no tienen ninguna relevancia.

Las acciones a tomar para tratar de reducir la deuda técnica del proyecto serán principalmente los fallos que puedan llegar a tener un impacto mayor a lo largo del desarrollo de este, como creación de instancias de clases que deberían estar privadas, convenios de utilización de métodos, uso de métodos actualizados...

Al no ir mal de tiempo según lo expuesto en el burndown chart, abarcaremos la mayoría de las notificaciones proporcionadas por sonar, de tal manera que:

➔ La solución de los fallos propuestos nos tomará 82 minutos de trabajo.

#### Lista de tareas a añadir en Scrum:

Al ser notificaciones presentes en el desarrollo de ambas historias de usuario del segundo sprint, se declararán las tareas por defecto en una sola historia de usuario, haciendo efecto a ambas.

- Desarrollar plan de calidad. (Ya realizada)
- Solucionar fallos de nivel Alto/Medio. -> 47min.
- Solucionar fallos de nivel Bajo. -> 35min.