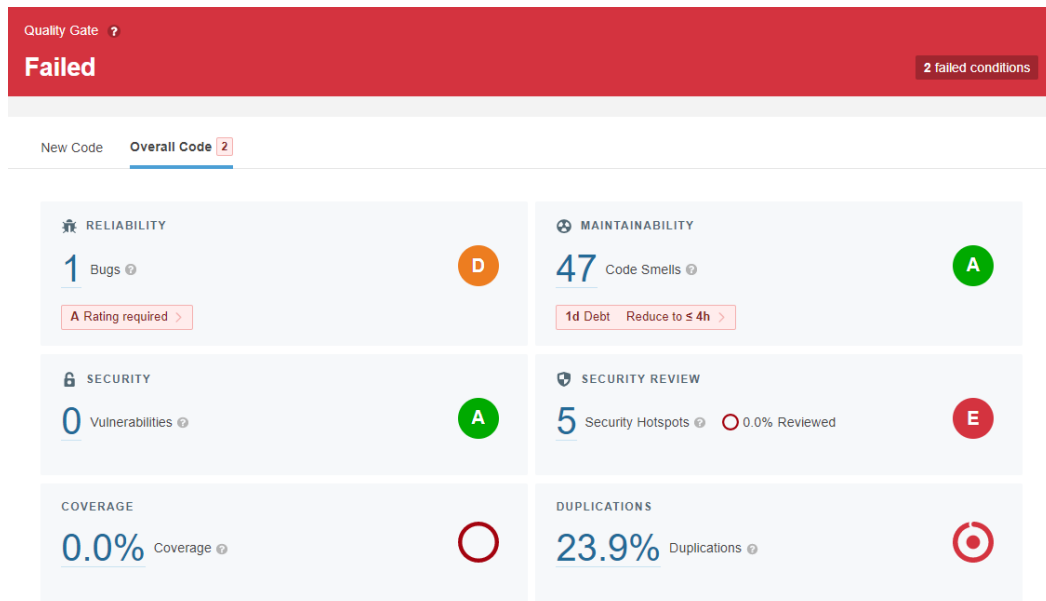


INFORME DE CALIDAD – SPRINT 3

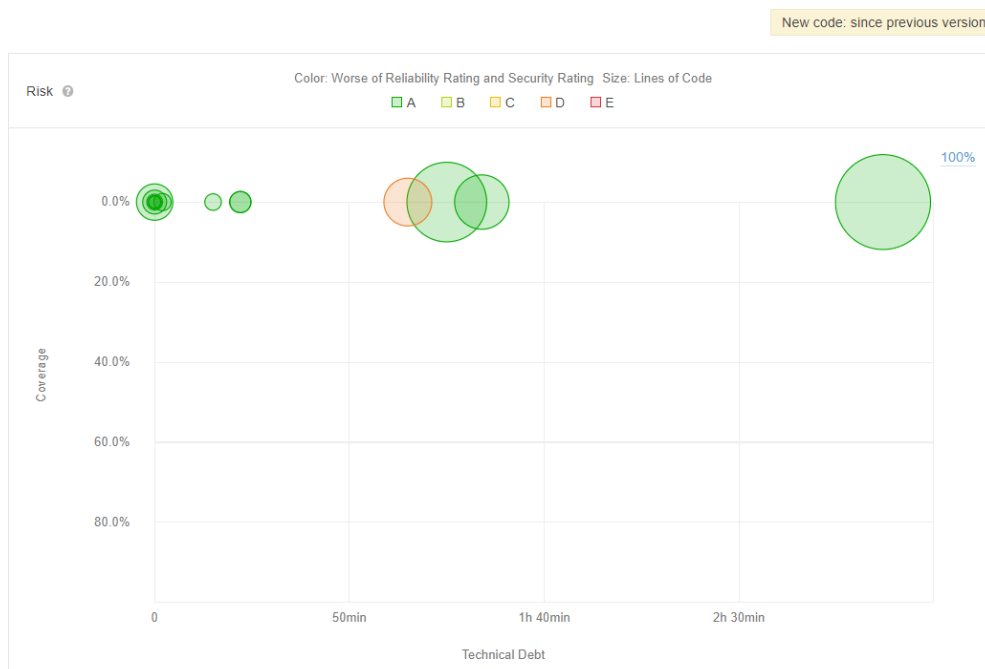
ANÁLISIS 15 NOVIEMBRE 2021 – Sergio Gallego Álvarez



Captura 1: vista general.

INCIDENCIAS

Como se puede observar en la captura 1, el análisis no pasa los criterios de calidad de nuestra organización. Contribuyendo a esto tenemos una deuda técnica de 8h y 57min y debería ser inferior a 4 horas. Pese a haberse reducido los code smells, sigue habiendo 47, más de lo ideal. Añadiendo a lo anteriormente comentado, hay un bug, dando una calificación de D en el apartado de fiabilidad. Por último, comentar que en el apartado de vulnerabilidad hay 5 vulnerabilidades, dejando este apartado con una calificación de E.



Captura 2: deuda técnica.

Entrando en más detalles, podemos observar que la mayoría de la deuda técnica (mostrada en la captura 2) viene de la clase “EventsActivity.java” con una deuda de 3h y 7min.

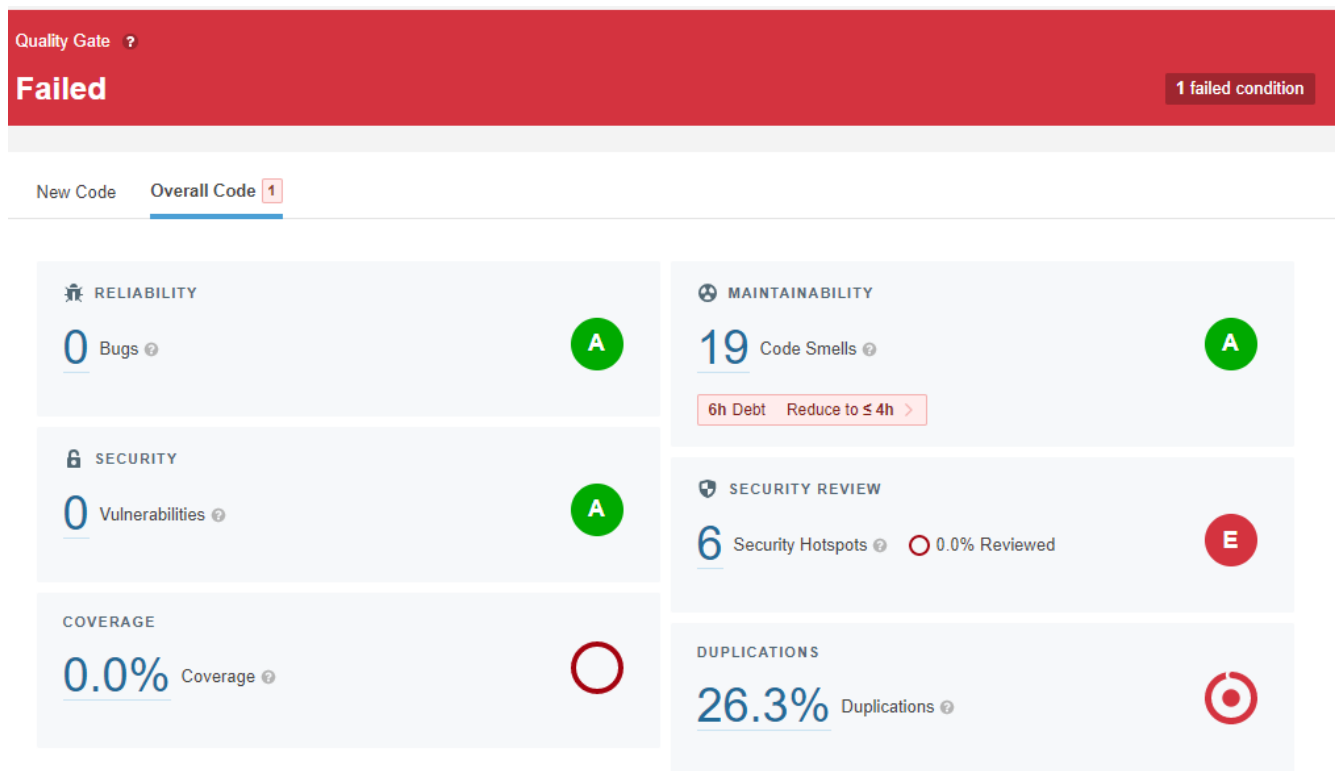


Captura 3: bug.

En cuanto al bug existente (mostrado en la captura 3), si se analiza con más detalle, se puede observar que proviene de un método getter de la clase “TodayEventsPresenter.java”, en concreto el nombre puede llegar a confusión al trabajar con él, dando lugar a problemas. A su vez, si se observa más generalmente, se puede observar como abajo hay otro método que realiza lo mismo, por lo que esto sería código redundante.

PLAN DE ACCIÓN

- 1) Reducir los code smells mayores y críticos que se han producido, localizados de la siguiente manera:
 - 23 code smells en “EventsActivity.java”.
 - 7 code smells en “EventsPresenter.java”.
 - 5 code smells en “TodayEventsActivity.java”.
 - 2 code smells en “TodayEventsPresenter.java”, “EventsArrayAdapter.java”, “TodayEventsArrayAdapter.java” y “EventsPresenterITest.java”.
 - 1 code smell en “FiltrarEventosPorFechaUITest.java”, “OrdenarSobreListaFiltradaUITest.java”, “EventsComparatorHora.java” y “EventsDetailActivity.java”.
- 2) Solucionar el bug de la clase “TodayEventsPresenter.java” para poder obtener una calificación de A en el apartado de fiabilidad.
- 3) Solucionar los 3 problemas de seguridad que se han producido en la clase “EventsActivity.java” y los 2 problemas producidos en la clase “TodayEventsActivity.java” para poder obtener una calificación de A en el apartado de seguridad.

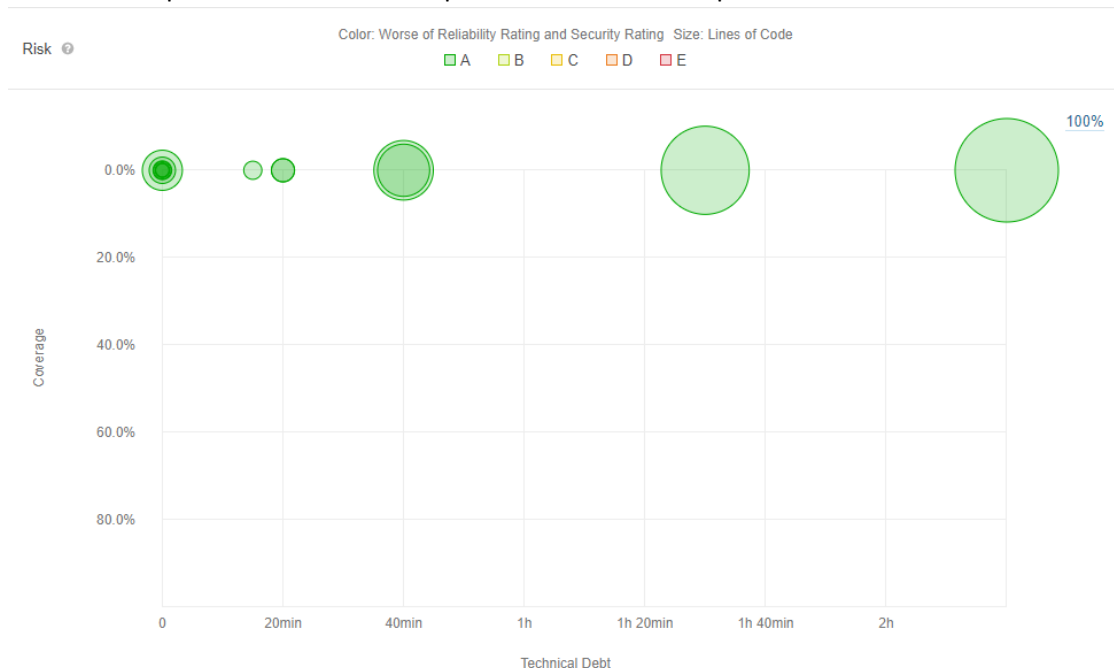


Vista general.

INCIDENCIAS

El análisis no pasa los criterios de calidad de la organización debido a que la deuda técnica es superior a la mínima aceptada, que es de 4 horas, y en nuestro caso tenemos 6 horas de deuda técnica. Y también debido a que tiene 6 brechas de seguridad.

Como puede observarse en la *Captura 1*, las clases donde mayor deuda técnica tenemos son TodayEventsActivity.java. Por tanto, nos centraremos en disminuir en la medida de lo posible los code smells presentes en esa clase para reducir la deuda técnica.



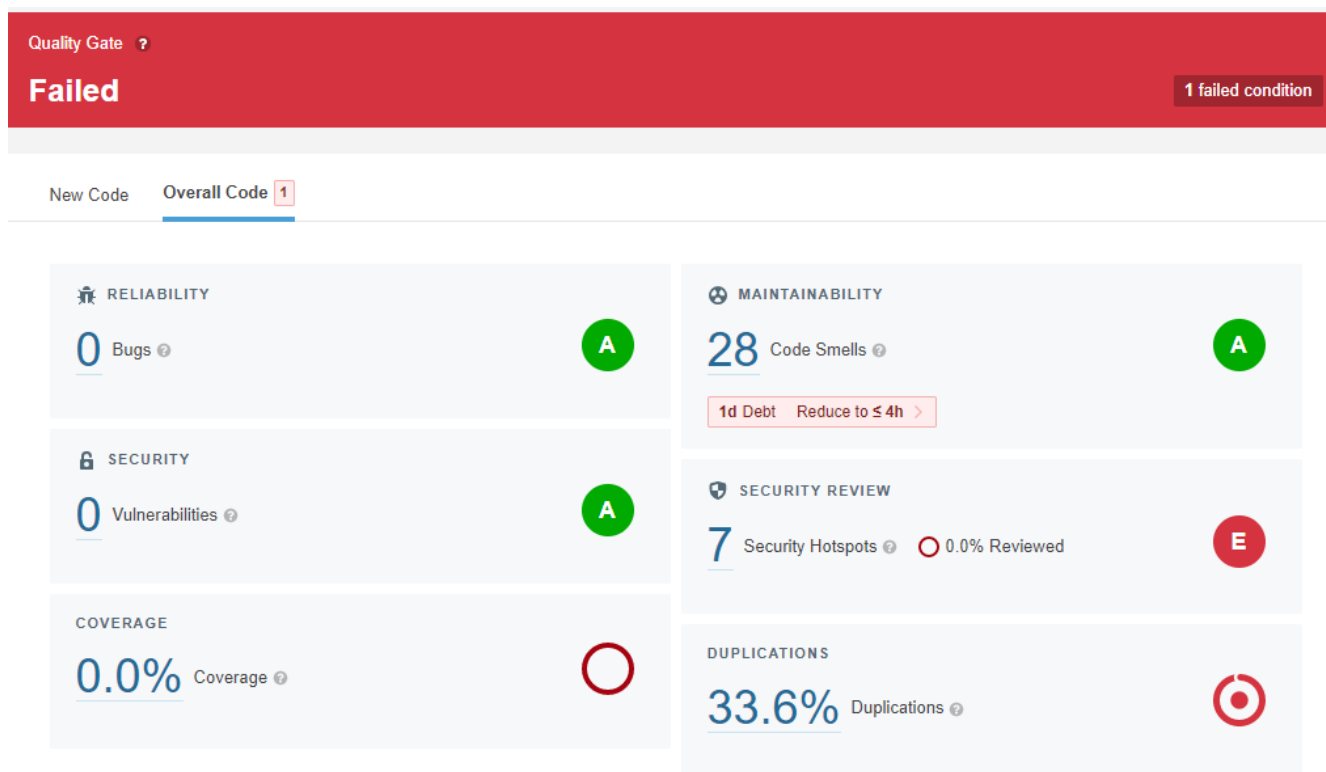
Captura 1.

Dentro de la pestaña issues, podemos observar que tenemos un fallo crítico, y nos centraremos en resolverlo.

PLAN DE ACCIÓN

- 1) Corregir el error crítico presente en la clase TodayEventsActivity.java.
- 2) Arreglar los code smells para las clases EventsActivity.java y EventsPresenterTest.java, especialmente los que tengan mayor gravedad.
- 3) Solucionar los 6 problemas de seguridad que se han producido para poder obtener una calificación de A en el apartado de seguridad.

ANÁLISIS 17 NOVIEMBRE 2021 – Adrián García Cubas



Vista general.

INCIDENCIAS

El análisis no pasa los criterios de calidad de la organización debido a que la deuda técnica es superior a la mínima aceptada, que es de 4 horas, y en nuestro caso tenemos 7.20 horas de deuda técnica (aunque sonar marque 1 día). Y también debido a que tiene 7 brechas de seguridad.

Como puede observarse en la *Vista general*, la mayor parte de la deuda técnica, es generada por el código duplicado, ya que este supone un 33.7 % del código de la aplicación. Como puede observarse en la *Captura 1*, de las 7 brechas de seguridad se corresponden al uso del método `getPreferences`; mientras que en la *Captura 2*, podemos observar que gran parte de los code smells y del tiempo de deuda técnica corresponde al código duplicado de las clases, que se considera como fallos mayores.

7 Security Hotspots to review

Review priority: **LOW**

Encryption of Sensitive Data 7

Make sure using an unencrypted database is safe here.
TO REVIEW

Make sure using an unencrypted database is safe here.
TO REVIEW

Make sure using an unencrypted database is safe here.
TO REVIEW

Make sure using an unencrypted database is safe here.
TO REVIEW

Make sure using an unencrypted database is safe here.
TO REVIEW

Make sure allowing clear-text traffic is safe here.
TO REVIEW

7 of 7 shown

Make sure using an unencrypted database is safe here.

Using unencrypted databases in mobile applications is security-sensitive [java:S6291](#)

Category: Encryption of Sensitive Data

Review priority: **LOW**

Assignee: Not assigned

Status: To review
This Security Hotspot needs to be reviewed to assess whether the code poses a risk.

src/.../isunican/eventossantander/view/events/EventsActivity.java

```

347  /**
348   * Crea un alertDialog personalizado que permite seleccionar una ordenación de la lista
349   * de eventos ascendentemente y descendentemente por el tipo de evento o por la hora de comienzo.
350   */
351  public void onOrdenarAlertDialog() {
352      SharedPreferences sharpref = getPreferences(Context.MODE_PRIVATE); // Sensitive
353
354      AlertDialog.Builder builder = new AlertDialog.Builder(this);
355      View view = LayoutInflater.from(this).inflate(
356          R.layout.alert_dialog_ordenar,
357          (ConstraintLayout) findViewById(R.id.layout_dialog_container)

```

What's the risk?

Are you at risk?

How can you fix it?

Storing data locally is a common task for mobile applications. Such data includes preferences or authentication tokens for external services, among other things. There are many convenient solutions that allow storing data persistently, for example SQLiteDatabase, SharedPreferences, and Realm. By default these systems store the data unencrypted, thus an attacker with physical access to the device can read them out easily. Access to sensitive data can be harmful for the user of the application, for example when the device gets stolen.

Captura 1

	Duplicated Lines (%)	Duplicated Lines
src/main/java/com/isunican/eventossantander/presenter/events/EventsPresenter.java	69.2%	202
src/main/java/com/isunican/eventossantander/presenter/today/TodayEventsPresenter.java	64.7%	174
src/main/java/com/isunican/eventossantander/view/today/TodayEventsActivity.java	63.0%	357
src/main/java/com/isunican/eventossantander/view/events/EventsActivity.java	59.9%	359
src/main/java/com/isunican/eventossantander/view/today/TodayEventArrayAdapter.java	47.6%	39
src/main/java/com/isunican/eventossantander/view/events/EventArrayAdapter.java	47.0%	39

There are 44 hidden components with a score of 0.0%. [Show Them](#)

Captura 2.

PLAN DE ACCIÓN

- 1) Eliminar la mayor parte posible del código duplicado de la aplicación
- 2) Solucionar los 7 problemas de seguridad que se han producido para poder obtener una calificación de A en el apartado de seguridad.

RESULTADO TRAS APLICAR EL PLAN DE ACCIÓN

Ingeniería del Software UC / AppEventosSantanderGrupo3 / Branches / develop

Summary Issues Security Hotspots Measures Code Activity

2.6k Lines of Code Version unspecified Last analysis 9 minutes ago 031b6eb5

Quality Gate

Passed

New Code Overall Code

RELIABILITY

0 Bugs

A

MAINTAINABILITY

19 Code Smells

A

SECURITY

0 Vulnerabilities

A

SECURITY REVIEW

0 Security Hotspots

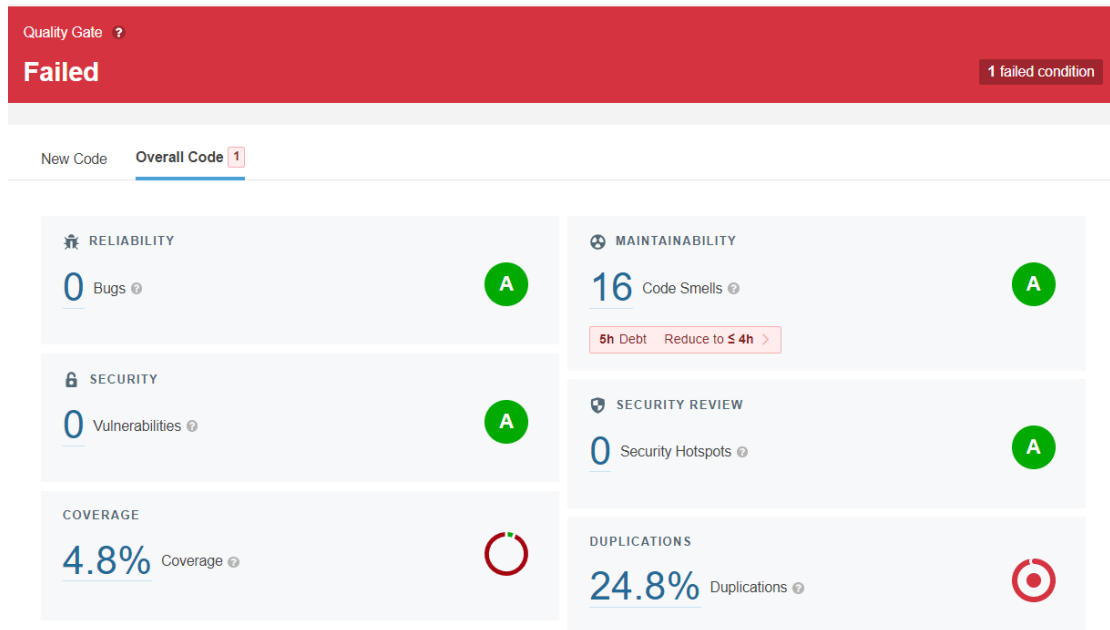
A

COVERAGE

3.2% Coverage

DUPLICATIONS

7.0% Duplications



Captura 1: vista general.

INCIDENCIAS

El análisis no pasa los criterios de calidad de la organización debido a que la deuda técnica es superior a la mínima aceptada, que es de 4 horas, en nuestro caso tenemos una deuda técnica de 5 horas y 19 minutos. Como puede observarse en la *Vista general*, la mayor parte de la deuda técnica, es generada por el código duplicado, ya que este supone un 24,8 % del código de la aplicación. Como podemos ver en las siguientes capturas las clases que más deuda general son EventsActivity y TodayEventsActivity



AppEventosSantanderGrupo3

View as

List

↑

↓

 to select files

←

→

 to navigate

59 files

Technical Debt 5h 19min

New code: since previous version

src/main/java/com/isunican/eventossantander/view/events/EventsActivity.java1h 37min

src/main/java/com/isunican/eventossantander/view/today/TodayEventsActivity.java1h 32min

src/main/java/com/isunican/eventossantander/presenter/today/TodayEventsPresenter.java35min

src/main/java/com/isunican/eventossantander/presenter/events/EventsPresenter.java30min

src/main/java/com/isunican/eventossantander/view/common/CommonArrayAdapter.java15min

src/main/java/com/isunican/eventossantander/view/eventsdetail/EventsDetailActivity.java15min

src/androidTest/java/com/isunican/eventossantander/view/events/AnhadirIconosDeTipoUITest.java10min

src/main/java/com/isunican/eventossantander/presenter/common/CommonPresenter.java10min

src/main/java/com/isunican/eventossantander/model/comparators/EventsComparatorHora.java10min

src/test/java/com/isunican/eventossantander/presenter/events/EventsPresenterTest.java5min

There are 49 hidden components with a score of 0.

Show Them

PLAN DE ACCIÓN

- 1) Eliminar la mayor parte posible del código duplicado de la aplicación.