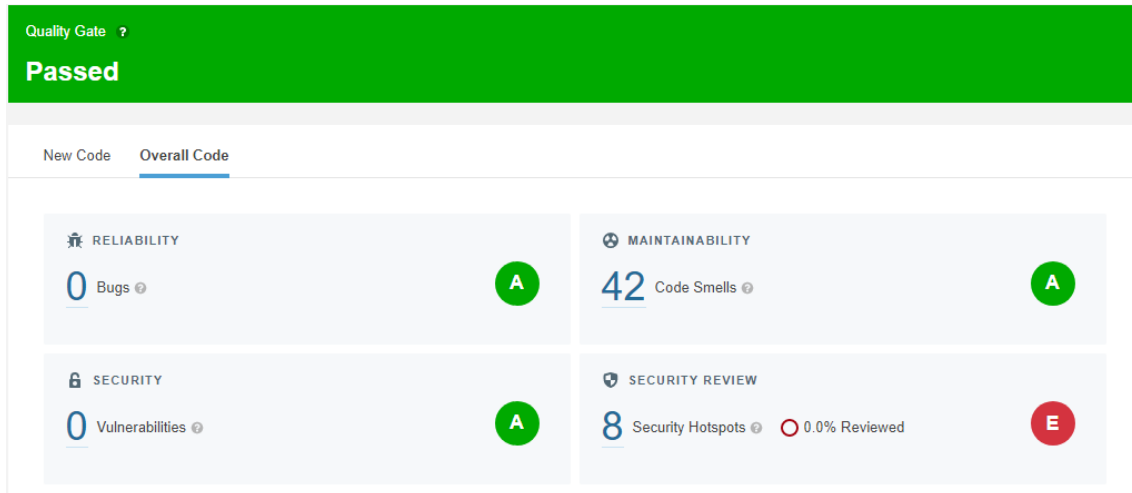


Informe de Calidad (Sprint 2)

Autores: Pedro Monje y Pablo Almohalla

ANÁLISIS 30 OCTUBRE 2021 - Pedro Monje

CAPTURA



INCIDENCIAS

El análisis pasa correctamente los criterios de calidad de la organización.

A pesar de esto, encontramos **42 issues de mantenibilidad** (code smells): **8 issues** en la clase **EventsDetailPresenter** que suman **31 minutos** a la deuda técnica, **8 issues** en la clase **EventsActivity** que suman **56 minutos**, **7 issues** en la clase **EventsPresenter** que suman **23 minutos**, **5 issues** en la clase **LocalEvents** que suman **24 minutos**, **4 issues** en **EventsDetailActivity** que suman **35 minutos**, **3 issues** en **FavouriteEventsActivity** que suman **6 minutos**, **2 issues** en **FavouriteEventsArrayAdapter** que suman **22 minutos**, **1 issue** en **EventsActivityUITest** que suma **20 minutos**, **1 issue** en **Event** que suma **2 minutos**, **1 issue** en **EventsAPIService** que suma **5 minutos**, **1 issue** en **EventArrayAdapter** que suma **20 minutos** y por último **1 issue** en **EventsDetailContract** que suma **2 minutos**.

En conclusión, tenemos 42 issues de mantenibilidad que generan **4 horas y 6 minutos de deuda técnica**.

PLAN DE ACCIÓN

- 1) Eliminar **Imports** que no se utilizan en las clases **Event**, **EventsPresenter**, **EventsDetailPresenter**, **LocalEvents**, **EventsActivity**, **EventsDetailContract**, **FavouriteEventArrayAdapter** y **FavouriteEventsActivity**.
- 2) Eliminar **campos privados** que no se utilizan en las clases **EventsDetailPresenter** y **LocalEvents**.
- 3) Eliminar el método no utilizado **creaFavoritos** así como sus respectivas llamadas dentro de la clase **EventsPresenter**.

Informe de Calidad (Sprint 2)

Autores: Pedro Monje y Pablo Almohalla

ANÁLISIS 03 NOVIEMBRE 2021 - Pablo Almohalla

CAPTURA



INCIDENCIAS

El análisis no pasa correctamente los criterios de calidad de la organización debido a que la deuda técnica debe de ser menor o igual a 4h y actualmente nos encontramos con 5h.

Es por esto que encontramos **46 issues de mantenibilidad** (code smells): **13** issues en la clase **CategoryFilterActivity.java** que suman **38 minutos** a la deuda técnica, **7** issues en la clase **EventsActivity.java** que suman **54 minutos**, **6** issues en la clase **EventsPresenter.java** que suman **15 minutos**, **4** issues en la clase **CategoryPresenter.java** que suman **11 minutos**, **4** issues en **EventsDetailActivity.java** que suman **32 minutos**, **3** issues en **AccessSharedPrefs.java** que suman **32 minutos**, **3** issues en **LocalEvents.java** que suman **35 minutos**, **1** issue en **EventsActivityUITest.java** que suma **20 minutos**, **1** issue en **EventsAPIService.java** que suma **5 minutos**, **1** issue en **EventArrayAdapter.java** que suma **20 minutos**, **1** issue en **ICategoryContract.java** que suma **2 minutos** y por último **1** issue en **FavouriteEventArrayAdapter.java** que suma 20 minutos. También hay **1** code smell en la clase **EventsRepository.java** el cuál no es capaz de sumar deuda técnica debido a que es un *TODO*, es decir, una tarea sin completar.

En conclusión, tenemos 46 issues de mantenibilidad que generan **4 horas y 44 minutos de deuda técnica**.

Cabe destacar que la clase **LocalEvents.java** queda obsoleta siendo sustituida por **AccessSharedPrefs.java**, por lo que en el siguiente plan de acción no se tendrá en cuenta.

PLAN DE ACCIÓN

- 1) Eliminar la clase obsoleta **LocalEvents.java**
- 2) Completar la tarea pendiente *TODO* de la clase **EventsRepository.java**, ya que a pesar de que no resuelve el problema de exceso de deuda técnica es una tarea importante a finalizar para resolver los posibles problemas que puedan surgir con GitHub Actions.

Informe de Calidad (Sprint 2)

Autores: Pedro Monje y Pablo Almohalla

- 3) Resolver los code smells determinados con una gravedad alta según Sonar, eliminando primero los códigos de bloques duplicados de la clase **AccessSharedPrefs.java**.
- 4) Eliminar los *import* que no se utilizan en las clases **EventsPresenter.java**, **CategoryPresenter.java**, **CategoryFilterActivity.java** y **EventsDetailActivity.java**.
- 4) Finalmente se resolverán el resto de code smells marcados como menos prioritarios en cuanto a gravedad según Sonar como sustituir variables como locales de **EventsAPIService.java** y renombrado de distintas variables de la clase **CategoryFilterActivity.java**, entre otros problemas menores.