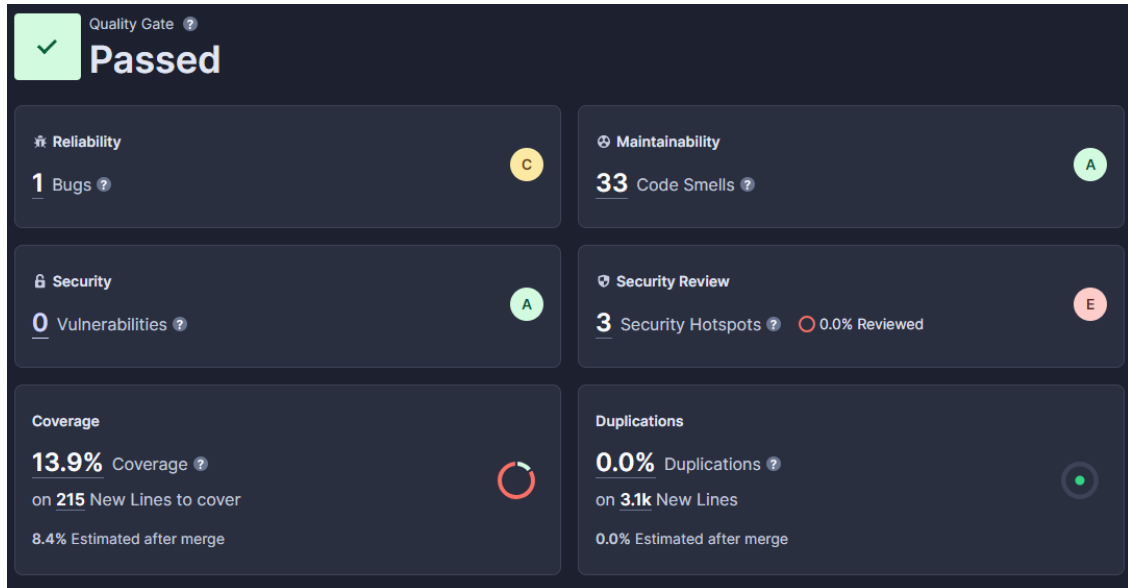


Informe de calidad

fecha de realización: 17 de Octubre de 2023

ANÁLISIS GENERAL DEL CÓDIGO



BUGS: 1

```
} catch(NullPointerException e){
    if (charger.address.province == null && charger.address.province == null){
        // Bug: Red squiggly line under the second 'charger.address.province == null'
    }
    informacion = informacion + "Ubicación: Provincia y ciudad no disponible\n";
} else if (charger.address.town == null){
    informacion = informacion + "Ubicación: " + charger.address.province + "\n";
} else if (charger.address.province == null){
    informacion = informacion + "Ubicación: " + charger.address.town + "\n";
}
```

Correct one of the identical sub-expressions on both sides of operator "&&"

Solución: sustituir una de las condiciones que forman el &&, ya que comprueban la misma condición (provincia). Esta condición se sustituye por: "charger.address.town == null"

SECURITY: 0

No hay nada que comprobar ya que no existen problemas de seguridad.

CODE SMELLS: 33

Dentro de los code smells, se comprueba cómo resolver los problemas de severidad alta, ya que son los que más pueden causar problemas en la funcionalidad del código al ampliarlo.

Primer problema

```
try {
    if ((charger.address.town.isBlank()) && (charger.address.province.isBlank())) {
        informacion = informacion + "Ubicación: Provincia y ciudad no disponible\n";
    } else if (charger.address.town.isBlank()) {
        informacion = informacion + "Ubicación: " + charger.address.province + "\n";
    }
}
```

Define a constant instead of duplicating this literal "Ubicación: " 5 times.

Solución: Se genera una constante llamada STR_UBI = "Ubicación: ", para solucionar la repetición de strings escritos de forma literal al ser de uso frecuente.

Segundo problema

```
Log.d("[DEBUG EN PRESENTER]", "Los filtros son:");
```

Define a constant instead of duplicating this literal "[DEBUG EN PRESENTER]" 5 times.

Solución: Se genera una constante llamada DEBUG_PRESENTER = "[DEBUG EN PRESENTER]", para solucionar la repetición de strings escritos de forma literal al ser de uso frecuente.

SECURITY REVIEW: 3

Primer Problema

```
//Web que muestra el mapa
WebView webView = findViewById(R.id.web);
webView.getSettings().setJavaScriptEnabled(true);
```

Make sure that enabling JavaScript support is safe here.

Solución: se establece la condición de javascript como "false", ya que esto provoca un problema en la seguridad del WebView debido a que, si se utiliza incorrectamente, podría ser una forma de ataque a la aplicación.

Segundo problema

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Make sure the use of "ACCESS_FINE_LOCATION" permission is necessary.

Solución: Esto no supone un problema para la aplicación, ya que es necesario acceder a la ubicación del usuario en base al GPS para poder utilizar la aplicación.

Tercer problema

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Make sure the use of "ACCESS_COARSE_LOCATION" permission is necessary.

Solución: Esto no supone un problema para la aplicación, ya que es necesario acceder a la ubicación del usuario en base al WIFI y torres de telefonía para poder utilizar la aplicación como alternativa al GPS en caso de que no funcione.

COVERAGE: 13.9 %

No hay nada que comprobar en el *coverage*, ya que según se vayan realizando los sprints el código se verá con una mejor cobertura al realizar más pruebas

DUPLICATIONS: 0.0%

No hay nada que comprobar, ya que no existe ninguna parte del código que se encuentre duplicada.