PRACTICA 2: Testing en E2E y Sonar

Ampliación de Ingeniería del Software
Ingeniería Informática (3º año)

Por Edgar Gutiérrez Pleite y Fernando

Índice

Parte 1: Bug en la aplicación

1.1Localizar el bug

El bug consiste en el fallo al volver a la página de una película tras cancelar la operación de editar de la misma.
Se localiza en el botón de la clase “ui cancel” en el último párrafo del html editFilmPage.html,en concreto en el atributo onclick

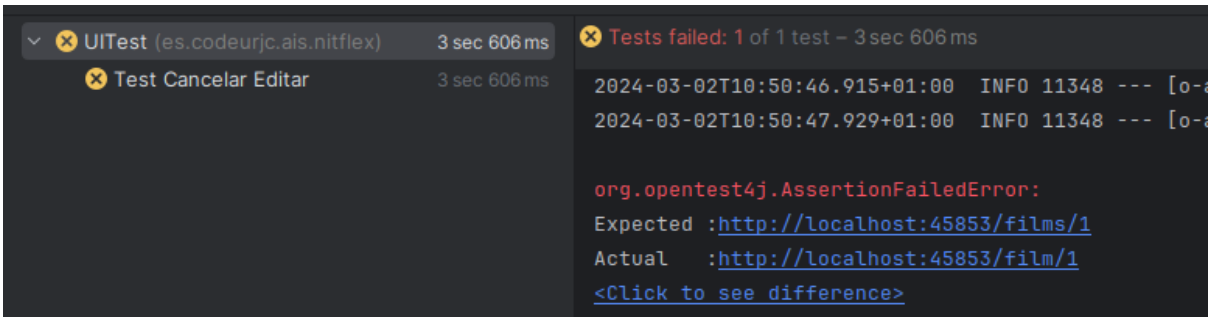
```
onclick="location.href='/film/{{film.id}}';return false”
```

1.2 Escribir un test que detecte el bug

```
@Test
@DisplayName("Test Cancelar Editar")
public void cancelEditFilmTest(){
    driver.get("http://localhost:"+this.port+"/");
    int num=driver.findElements(By.className("film-details")).size();
    driver.findElement(By.id("create-film")).click();
    driver.findElement(By.name("title")).sendKeys("Casino Royale");
    driver.findElement(By.name("releaseYear")).sendKeys("2006");
driver.findElement(By.name("url")).sendKeys("https://www.zonanegativa.com/
imagenes/2016/04/cubierta_james_bond_casino_royal.jpg");
    driver.findElement(By.name("synopsis")).sendKeys("Esta película
trata sobre un espía . Esta to guapa");
    driver.findElement(By.id("Save")).click();
    String filmUrl= driver.getCurrentUrl();
    driver.findElement(By.id("edit-film")).click();
    driver.findElement(By.id("cancel-edit")).click();
    assertEquals(filmUrl,driver.getCurrentUrl())
};
```

Hemos preferido usar un formato de código en vez de captura porque parece verse mejor en general,aunque un par de líneas eran algo largas y han quedado un poco regular,que son la de la url de la película y la sinopsis.

1.3 Ejecutar el test (debe fallar)



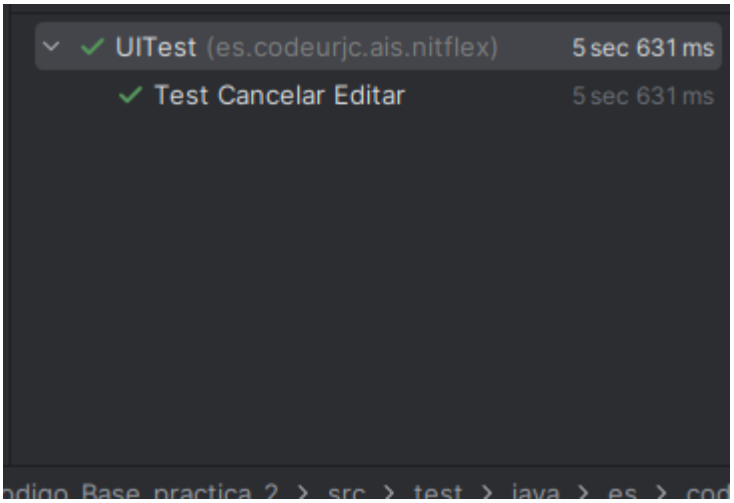
1.4 Corrección del bug

En concreto en la url de href,que debería ser:

```
onclick="location.href='/films/{{film.id}}';return false"
```

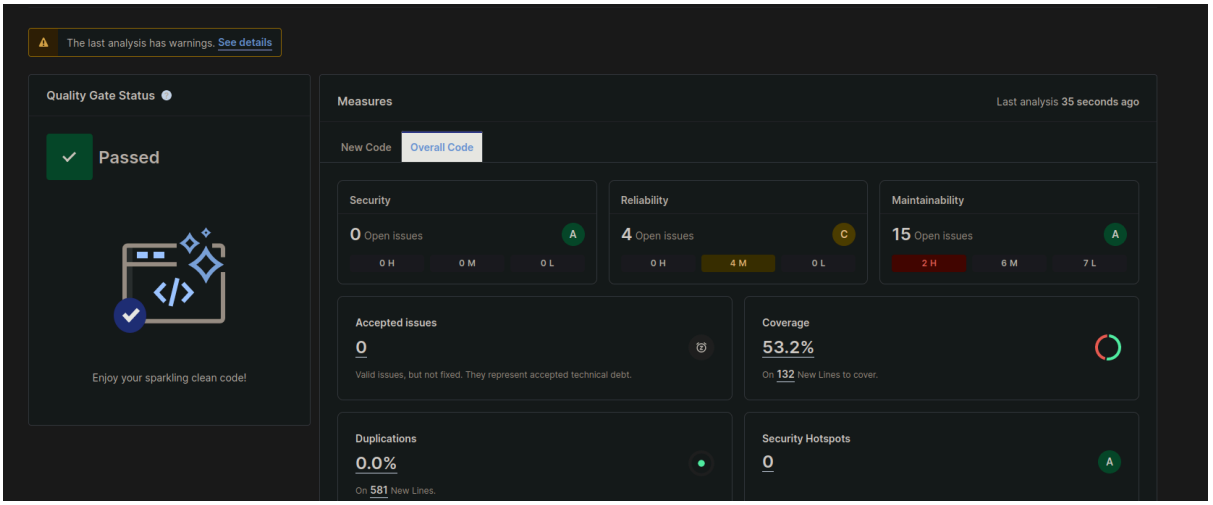
En vez de /film es /films.Ya que olvidar la “s”. llevaría a una url incorrecta.

1.5 Ejecución del test de nuevo



Parte 2: Análisis estático con sonar

2.0 Captura inicial



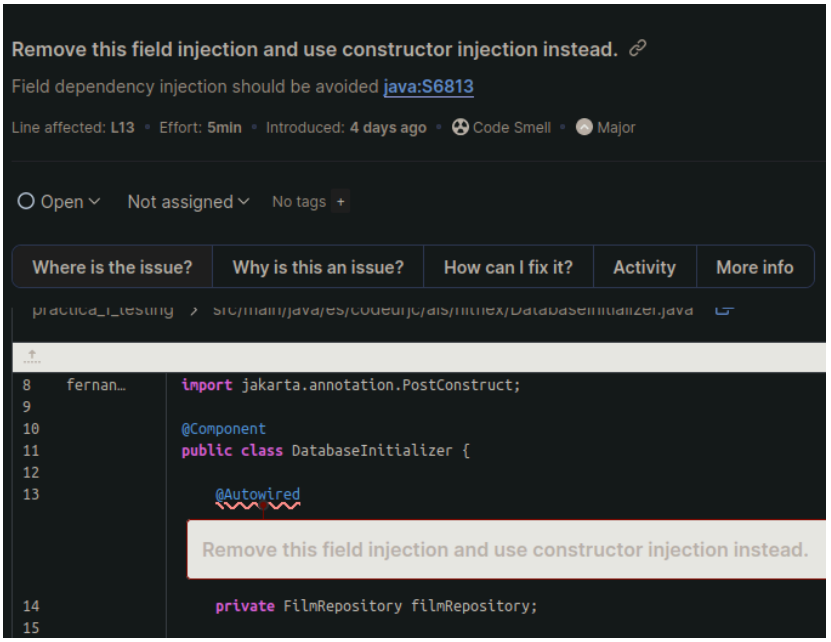
Como se ve en la captura, Sonar nos reporta 19 errores en el código, 4 de confianza y 15 de mantenibilidad.

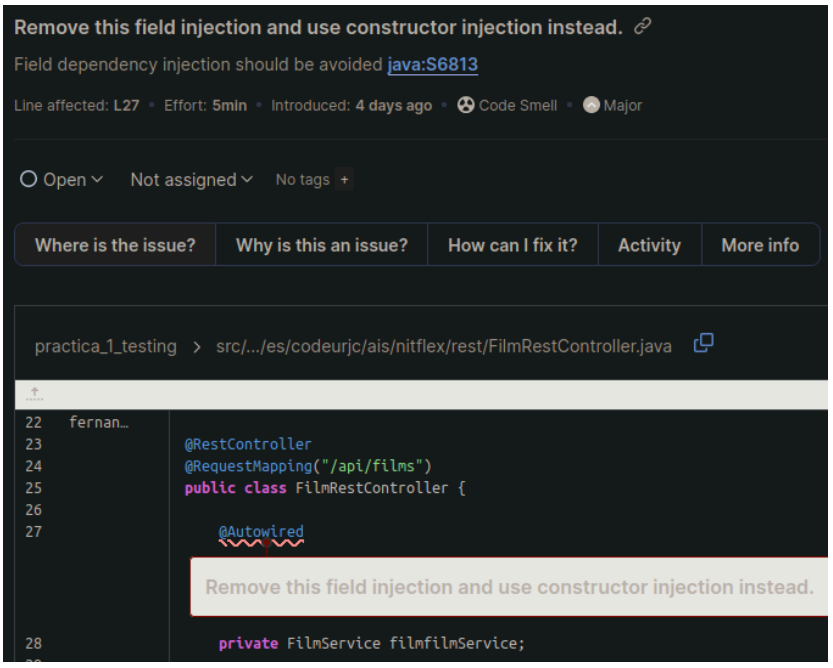
2.1 Positivos

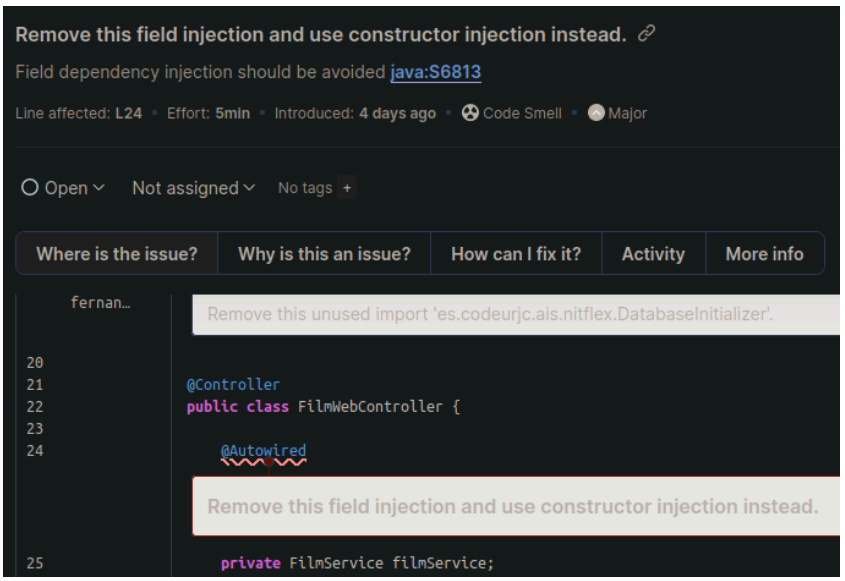
2.1.1 Code smell

2.1.1.1 Cambiar inyección de campo por inyección en constructor

2.1.1.1.1 Muestras



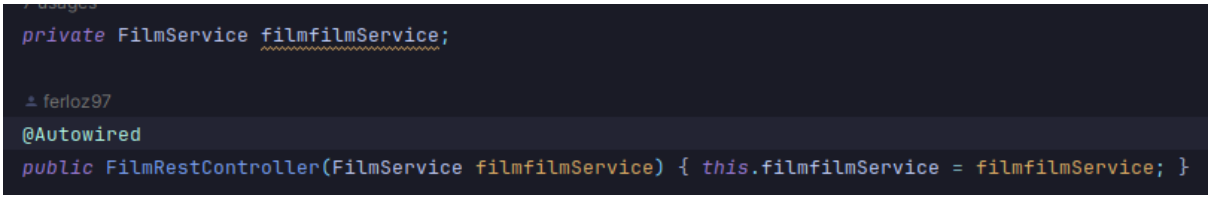


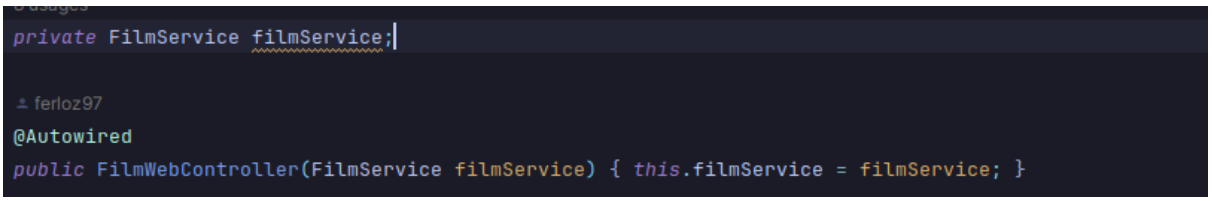


2.1.1.1.2 Explicación

Se desaconseja la inyección de dependencias a través de campos. Esto permite la creación de objetos en un estado no válido y dificulta las pruebas.

2.1.1.1.3 Corrección





2.1.1.2 Uso de la clase System.out en vez de la clase Logger

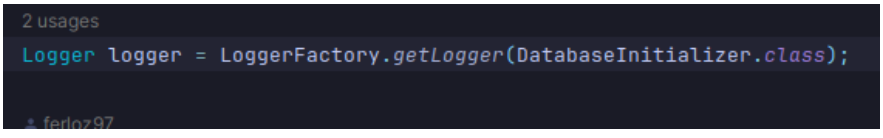
2.1.1.2.1 Muestras

-

2.1.1.2.2 Explicación

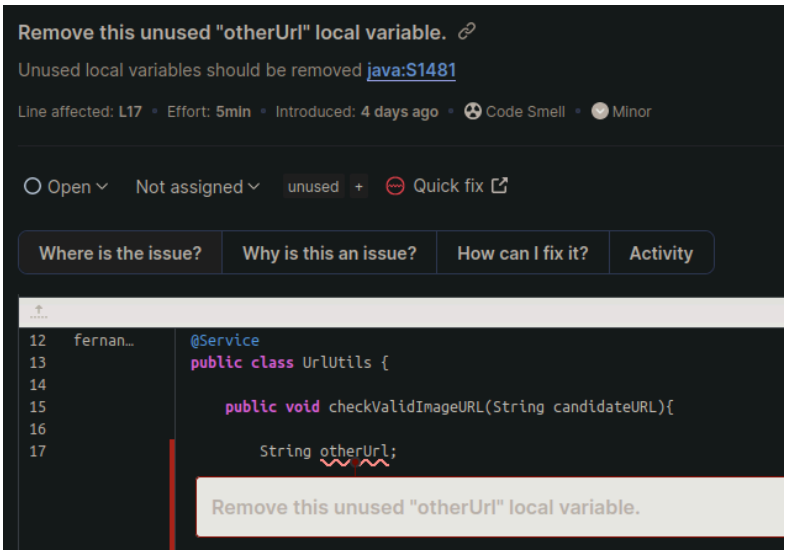
Uso de logs que sean de fácil acceso, uniformemente formateados para facilitar la legibilidad, registrados adecuadamente y registrados de manera segura al tratar datos sensibles. Usando **System.out** no se logra eso, hay que usar **loggers.**

2.1.1.2.3 Corrección



2.1.1.3 Variable no usada

2.1.1.3.1 Muestra



2.1.1.3.2 Explicación

Es código muerto, no tiene ninguna función

2.1.1.3.3. Corrección

Quitar esa línea de código

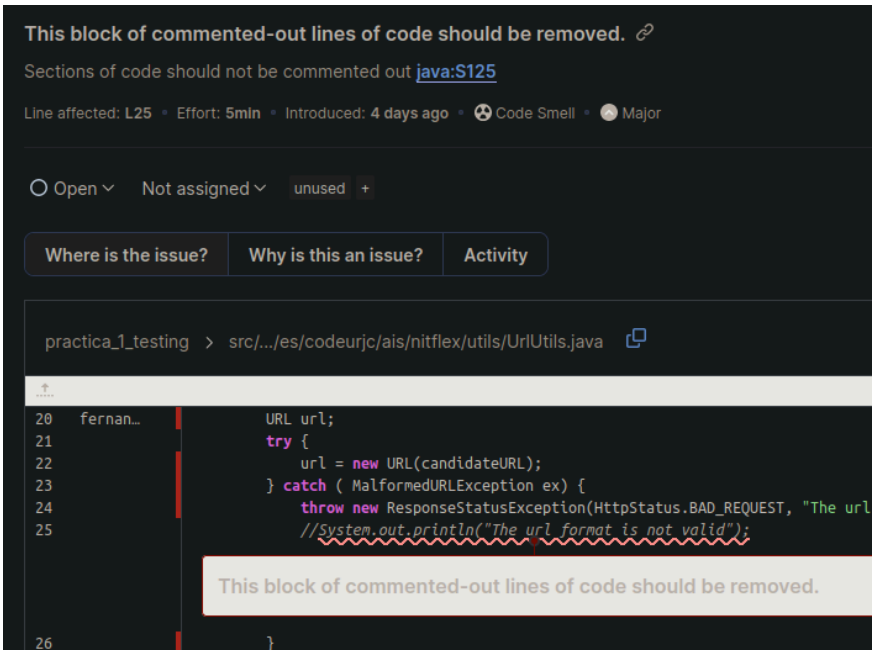
```
public class UrlUtils {

    1 usage   ferloz97
    public void checkValidImageURL(String candidateURL){


        // CHECK THAT URL HAS VALID FORMAT
        URL url;
        try {
            url = new URL(candidateURL);
        } catch ( MalformedURLException ex) {
            throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "The url format is not valid");
        }
```

2.1.1.4 Lineas de codigo comentadas

2.1.1.4.1 Muestra



2.1.1.4.2 Explicación

Es código muerto, no tiene ninguna función y afecta negativamente a la mantenibilidad del código.

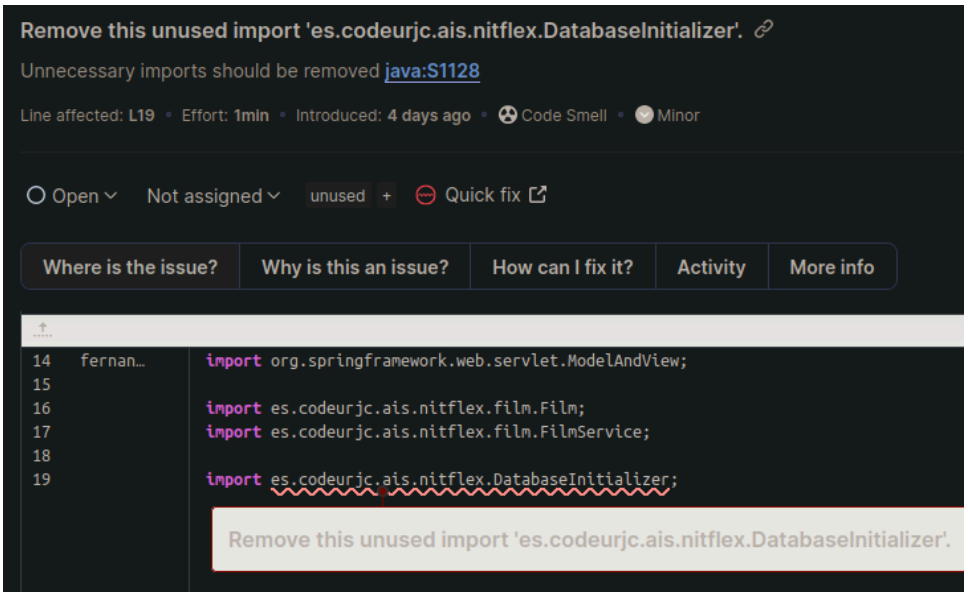
2.1.1.4.3. Corrección

Quitar esa línea de código

```
    url = new URL(candidateURL);
} catch ( MalformedURLException ex) {
    throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "The url format is not valid");
}
```

2.1.1.5 Import no usado

2.1.1.5.1 Muestra

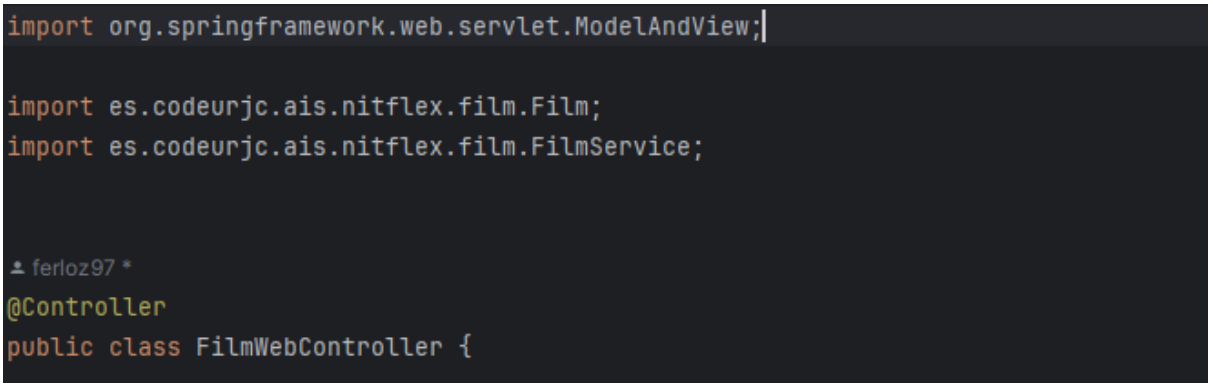


2.1.1.5.2 Explicación

Es código muerto, no tiene ninguna función. Además de que probablemente este import es incorrecto de por sí,ya que quien trabaja con la base de datos es alguna clase repositorio y no el controlador web.

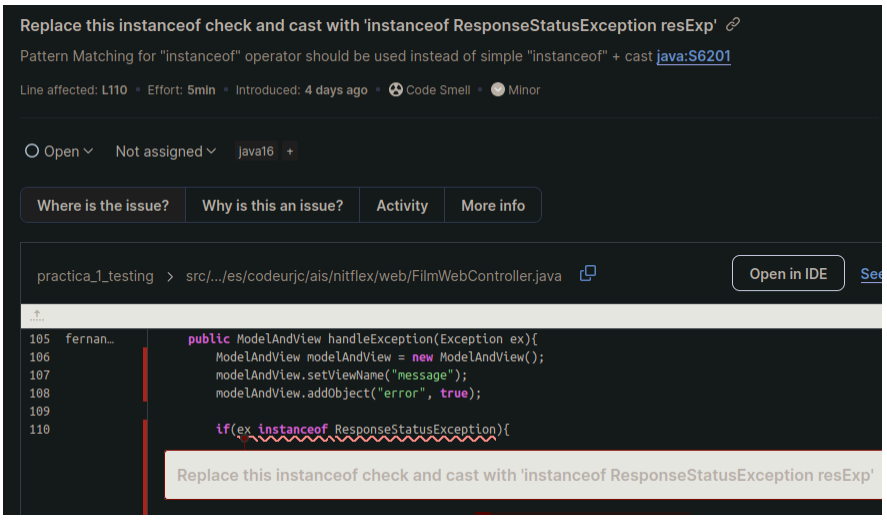
2.1.1.5.3. Corrección

Quitar esa línea de código



2.1.1.6 Uso de instanceof no recomendado

2.1.1.6.1 Muestra



2.1.1.6.2 Explicación

No es la forma más clara de programar

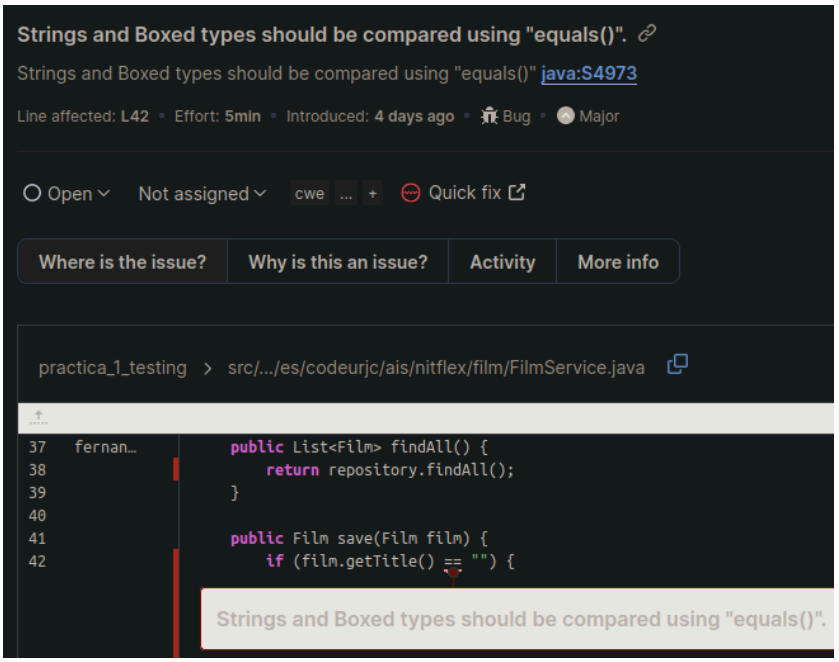
2.1.1.6.3 Corrección

```
if(ex instanceof ResponseStatusException resExp){
    modelAndView.addObject(MESSAGE_STRING, resExp.getReason());
}else if(ex instanceof BindException){
```

2.1.2 Bug

2.1.2.1 Uso de == en vez de equals para comparar instancias de clases

2.1.2.1.1 Muestra



2.1.2.1.2 Explicación

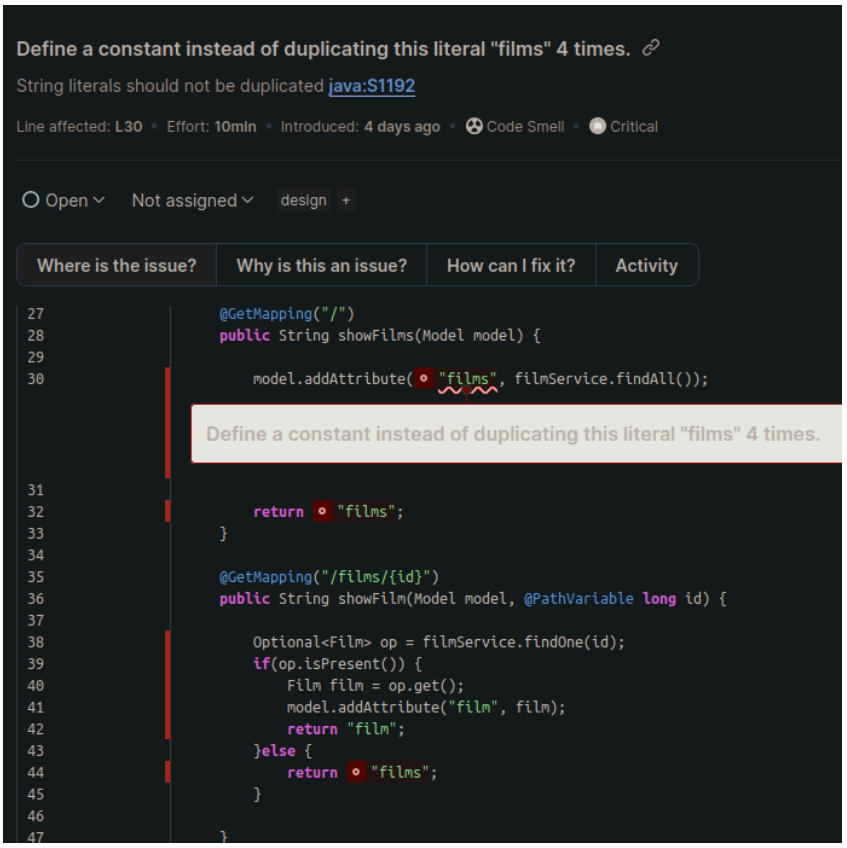
De esta manera se está comparando las direcciones de memoria de las instancias en vez del contenido de estas

2.1.2.1.3 Correción

```
if (film.getTitle().equals("")) {
    throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "The title is empty");
}
```

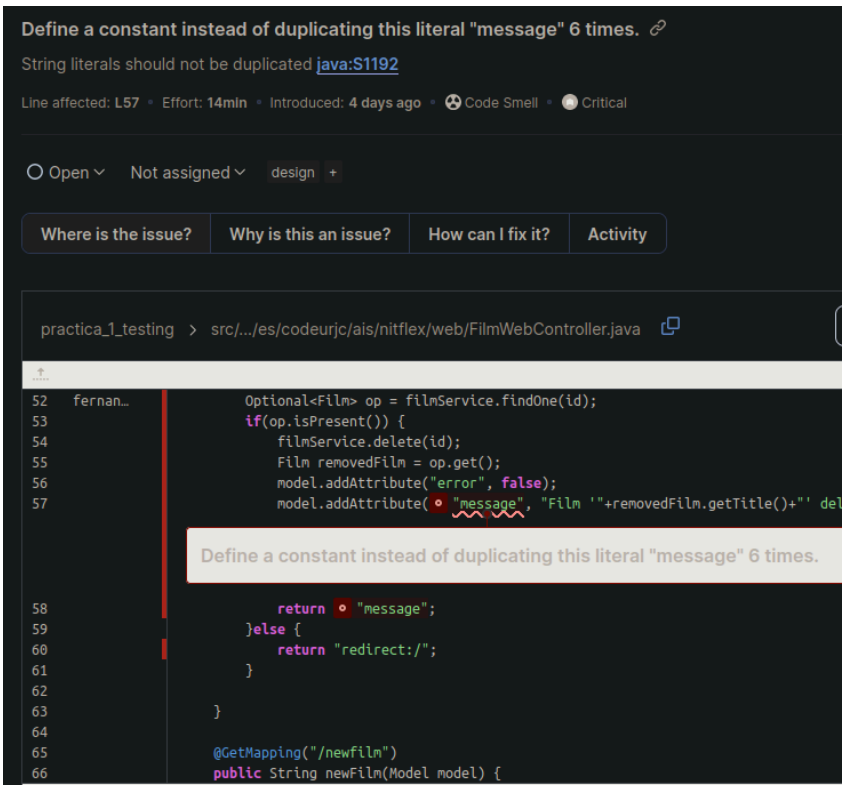
2.2 Falsos positivos

2.2.1 Código duplicado

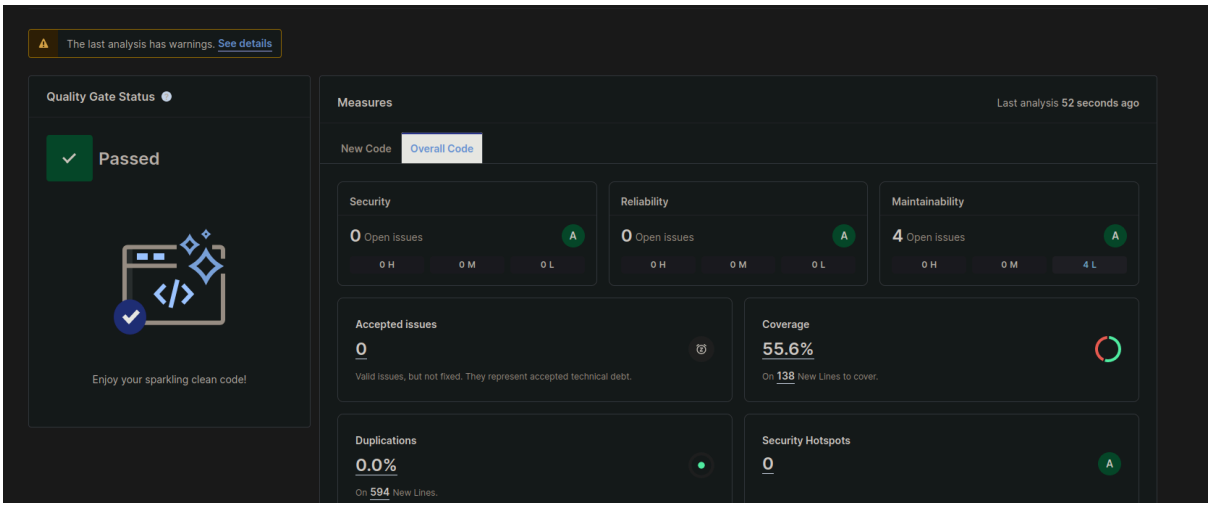


Explicación: Sonar detecta que debería usarse una constante para ambos string, pero esto supondría un problema de mantenibilidad,ya que el primer films se refiera al nombre del atributo en el html(p.e objetos de la clase que pasa al html),mientras que el otro es el nombre del html que se devuelve,es decir, este cambio podría producir que en caso de que se cambie el nombre del html en un futuro la constante se vea comprometida. Nuestra recomendación es dar nombres cortos más significativos a los html para que no se confundan con los atributos, por ejemplo “showFilms” en este caso(habría que cambiar el nombre de los html y los strings en los return de los distintos métodos del controlador web para que sean más apropiados).

Lo mismo ocurre con los string que son “message” en el controlador web.Recomendamos al atributo llamarlo errorMessage y al html errorMessagePage o algo por estilo.


Define a constant instead of duplicating this literal "message" 6 times.
String literals should not be duplicated java:S1192
Line affected: L57 • Effort: 14min • Introduced: 4 days ago • Code Smell • Critical
Open
Not assigned
design +
Where is the issue?
Why is this an issue?
How can I fix it?
Activity
practica_1_testing > src/.../es/codeurjc/ais/nitflex/web/FilmWebController.java
52 fernan… Optional<Film> op = filmService.findOne(id);
53 if(op.isPresent()) {
54 filmService.delete(id);
55 Film removedFilm = op.get();
56 model.addAttribute("error", false);
57 model.addAttribute("message", "Film '"+removedFilm.getTitle()+"' del
Define a constant instead of duplicating this literal "message" 6 times.
58 return "message";
59 }else {
60 return "redirect:/";
61 }
62
63 }
64
65 @GetMapping("/newfilm")
66 public String newFilm(Model model) {

2.3 Overview. New Code



2.4 ¿Falso positivo?

El propio IDE nos avisa indicando que el parámetro de entrada no se usa en el método, pero por alguna razón Sonar no ha reportado nada. La solución es quitar el parámetro de entrada del método tal que: