

Universidad de Sevilla
Escuela Técnica Superior de Ingeniería Informática

Refactoring



Grado en Ingeniería Informática – Ingeniería del Software
Diseño y Pruebas II
Curso 2019 – 2020

Miembros del equipo

Jorge Andrea Molina

Juan Carlos Cortés Muñoz

María Elena Molino Peña

Alejandro Muñoz Aranda

Mario Ruano Fernández

Fernando Ruíz Robles

<https://github.com/dp2-g3-7/petclinic.git>

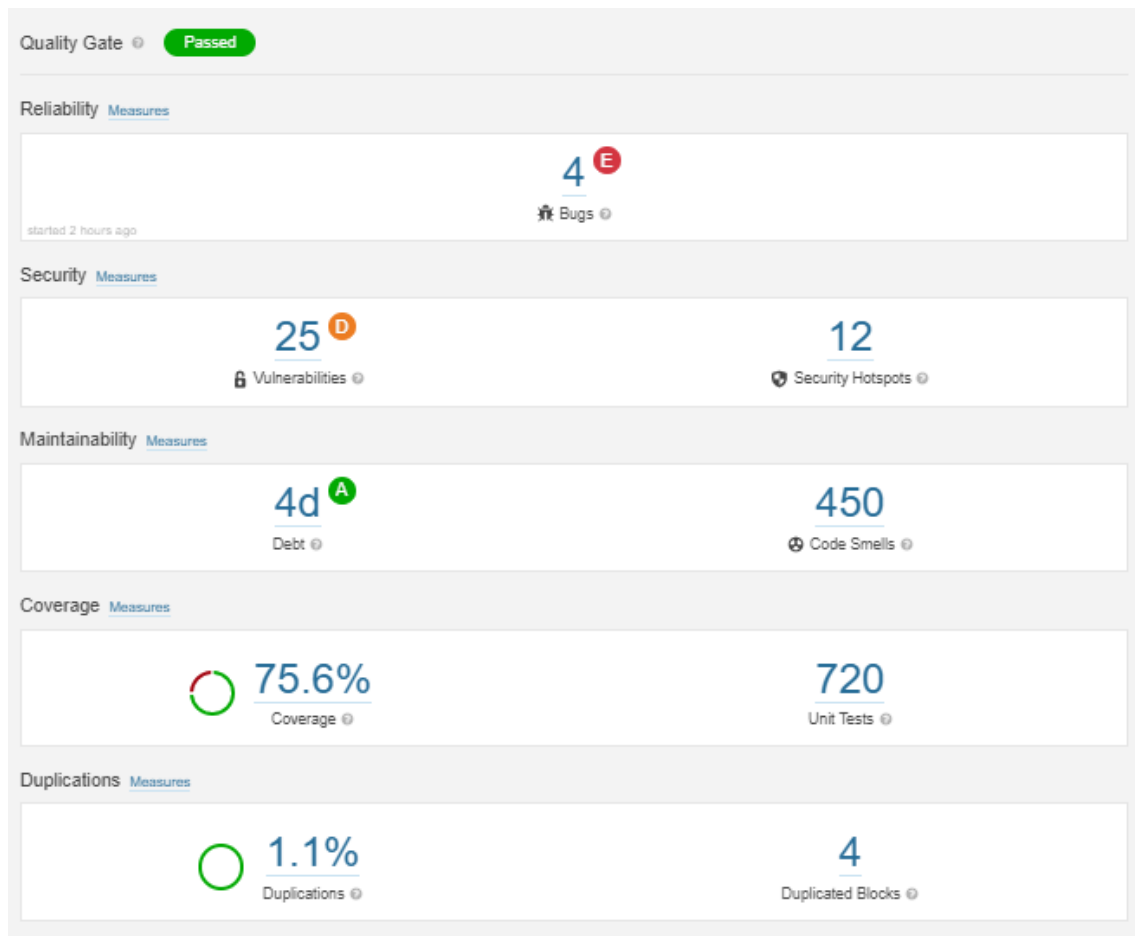
Índice

Refactoring	3
Bugs	4
Coverage.....	4
Maintainability – Technical Debt	4
Dispensables.....	5
Change preventers	6
Tests	7
Resultados	8

Refactoring

SonarCloud es un servicio web que realiza un análisis de la calidad y la seguridad del código de un proyecto. Presenta información muy importante previo al lanzamiento de una aplicación a producción, como pueden ser los bugs existentes, las vulnerabilidades, la deuda técnica, la cobertura de las pruebas, etc.

Se obtuvo el siguiente resultado:



Se han registrado algunos errores que se comentarán más específicamente a continuación y una deuda técnica de, aproximadamente, 4 días. Además, han surgido vulnerabilidades en la seguridad de la aplicación. Por otro lado, el porcentaje de líneas de código cubiertas es alto y apenas existe duplicidad de código. Por todo ello, según se indica con el valor Passed de Quality Gate, el proyecto está listo para ponerse en producción. A continuación, se profundiza más en cada uno de esos puntos, presentando ejemplos y su solución.

Bugs

El número de errores presentes en la aplicación es pequeño pero suficiente para ser calificado como suspenso. Los errores se deben al mal uso de los objetos de tipo Optional en algunas clases de la aplicación y a otros conceptos.

The screenshot shows a bug tracking interface. On the left is a sidebar with filters for Type, Severity, Resolution, Status, Security Category, Creation Date, Language, Rule, Tag, Directory, File, and Assignee. The 'Type' filter is set to 'BUG' with a 'Clear' button. The 'Severity' filter shows 1 Blocker, 0 Critical, 0 Info, and 3 Major. The main area displays three bugs:

- Bug 1:** "Use the 'equals' method if value comparison was intended. Why is this an issue?" (28 days ago, L78, cert, cwe). Severity: Major. Status: Open. Not assigned. 5min effort.
- Bug 2:** "'typeNameDontExists's' @Transactional requirement is incompatible with the one for this method. Why is this an issue?" (2 months ago, L23, spring). Severity: Blocker. Status: Open. Not assigned. 20min effort.
- Bug 3:** "Call 'Optional#isPresent()' before accessing the value. Why is this an issue?" (29 days ago, L44, cwe). Severity: Major. Status: Open. Not assigned. 10min effort.

At the bottom, it says "4 of 4 shown".

Estos problemas no han supuesto mucha dificultad y, con su resolución, se ha obtenido un código libre de errores.

Coverage

La cobertura de las pruebas presenta un buen porcentaje. Se han realizado un total de 720 tests al código de la aplicación por lo que, teniendo en cuenta las instrucciones de no realizar pruebas a métodos simples, se puede asegurar un buen funcionamiento de la aplicación.

Maintainability – Technical Debt

Se ha obtenido una valoración de A en este apartado. Pese a ello, la deuda técnica asciende a los 4 días y hay un total de 450 malos olores. Varios de ellos corresponden al mismo patrón y tienen una fácil solución. Se van a resolver varios de ellos con el fin de reducir la deuda técnica, en especial, aquellos cuya prioridad es Blocker y Critical.

Blocker	17	Minor	93
Critical	35	Info	114
Major	191		

En los próximos epígrafes, se muestran algunos tipos de code smells identificados y la solución que se le ha dado.

Dispensables

Son fragmentos inútiles cuya ausencia haría que el código fuese más legible y limpio. Hemos encontrado dos tipos distintos:

- **Dead code.** Se han registrado comentarios en los métodos que corresponden a antiguas implementaciones pero que no se han borrado.

src/.../samples/petclinic/service/StayService.java

This block of commented-out lines of code should be removed. Why is this an issue? 29 days ago L64 unused

Code Smell Major Open Not assigned 5min effort

This block of commented-out lines of code should be removed. Why is this an issue? 29 days ago L76 unused

Code Smell Major Open Not assigned 5min effort

This block of commented-out lines of code should be removed. Why is this an issue? last month L94 unused

Code Smell Major Open Not assigned 5min effort

La solución es sencilla: eliminar dichas líneas de código.

```
else {  
    // stayToUpdate.setRegisterDate(stay.getRegisterDate());  
    // stayToUpdate.setReleaseDate(stay.getReleaseDate());  
    // stay.setRegisterDate(newDate);  
    // stay.setReleaseDate(newDate2);  
    this.stayRepository.save(stay);  
}
```

- **Duplicate code.** Las partes de un método eran prácticamente iguales ya que pertenecían a roles distintos pero la funcionalidad es idéntica. Se ha creado un nuevo método formado por las partes que tenían en común, simplificando la dificultad de lectura.

src/.../samples/petclinic/model/Owner.java

1 duplicated blocks of code must be removed. Why is this an issue? 26 days ago pitfall

Code Smell Major Open Not assigned 20min effort

src/.../samples/petclinic/model/Vet.java

1 duplicated blocks of code must be removed. Why is this an issue? 2 months ago pitfall

Code Smell Major Open Not assigned 20min effort

src/.../samples/petclinic/model/Vets.java

Rename field "vets" Why is this an issue? 3 months ago L33 brain-overload

Code Smell Major Open Not assigned 10min effort

```

@PostMapping(value = "/adoptions/pet")
public String processAdoptForm(@RequestParam String name, @RequestParam String type, @RequestParam String age, ModelMap
String username = SecurityContextHolder.getContext().getAuthentication().getName();
Owner owner = this.ownerService.findOwnerByUsername(username);
this.initPet(name, type, age, owner);
return "redirect:/owner/requests";
}

private void initPet(String name, String type, String age, Owner owner) throws DataAccessException, DuplicatedPetNameEx
//Iniciación pet
Pet petToAdopt = new Pet();
petToAdopt.setType(this.petType(type));
petToAdopt.setBirthDate(this.birtdate(age));
petToAdopt.setName(name);
petToAdopt.setStatus(PENDING);
petToAdopt.setJustification("");
petToAdopt.setActive(true);
owner.addPet(petToAdopt);
//guardado mascota
try {
    this.petService.savePet(petToAdopt);
} catch (Exception e) {
    ThreadLocalRandom random = ThreadLocalRandom.current();
    Integer entero= random.nextInt(0,100);
    petToAdopt.setName(name+"-"+entero);
    this.petService.savePet(petToAdopt);
}
}

@PostMapping(value = "/adoptions/owner/{ownerId}/pet")
public String processAdoptAdminForm(@PathVariable("ownerId") int ownerId, @RequestParam String name, @RequestParam Stri
Owner owner = this.ownerService.findOwnerById(ownerId);
this.initPet(name, type, age, owner);
return "redirect:/owner/"+ownerId+"/requests";
}

```

Change preventers

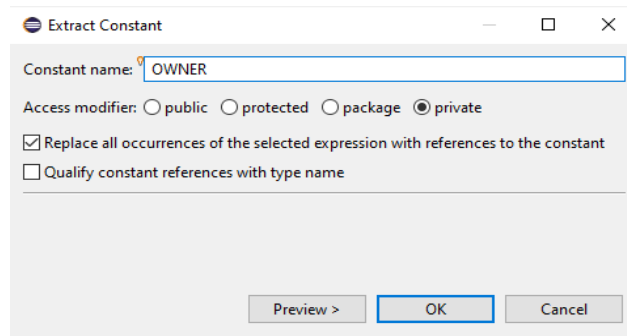
Para realizar un cambio en alguna parte de una clase o paquete, es necesario cambiar varias partes en otros lugares. Se ha dado, al menos, un tipo de este problema.

- **Divergent change.** Ha aparecido en un gran número de clases de la aplicación en las que se repetían continuamente valores de variables. Un cambio hubiera supuesto tener que modificar todos los métodos de la clase.

src/.../petclinic/configuration/SecurityConfiguration.java

Define a constant instead of duplicating this literal "admin" 14 times. Why is this an issue?	7 days ago ▾ L37		design
Code Smell Critical Not assigned 30min effort			
Define a constant instead of duplicating this literal "owner" 5 times. Why is this an issue?	10 days ago ▾ L38		design
Code Smell Critical Not assigned 12min effort			
Define a constant instead of duplicating this literal "veterinarian" 4 times. Why is this an issue?	10 days ago ▾ L43		design
Code Smell Critical Not assigned 10min effort			

Se ha optado por crear constantes con esos valores. De esta forma, si, por ejemplo, se pretende cambiar la página a la que redirigen varios métodos de un controlador, bastaría con sustituir en la constante. Además, Eclipse ofrece una funcionalidad por la que podemos implementar esta solución de una manera muy sencilla.



Tests

Se han realizado refactorizaciones en algunas pruebas de interfaz de usuario. SonarCloud reportaba la necesidad de completar las aserciones a estas ya que, en realidad, no se estaba probando nada.

src/.../samples/petclinic/service/MedicineServiceTests.java

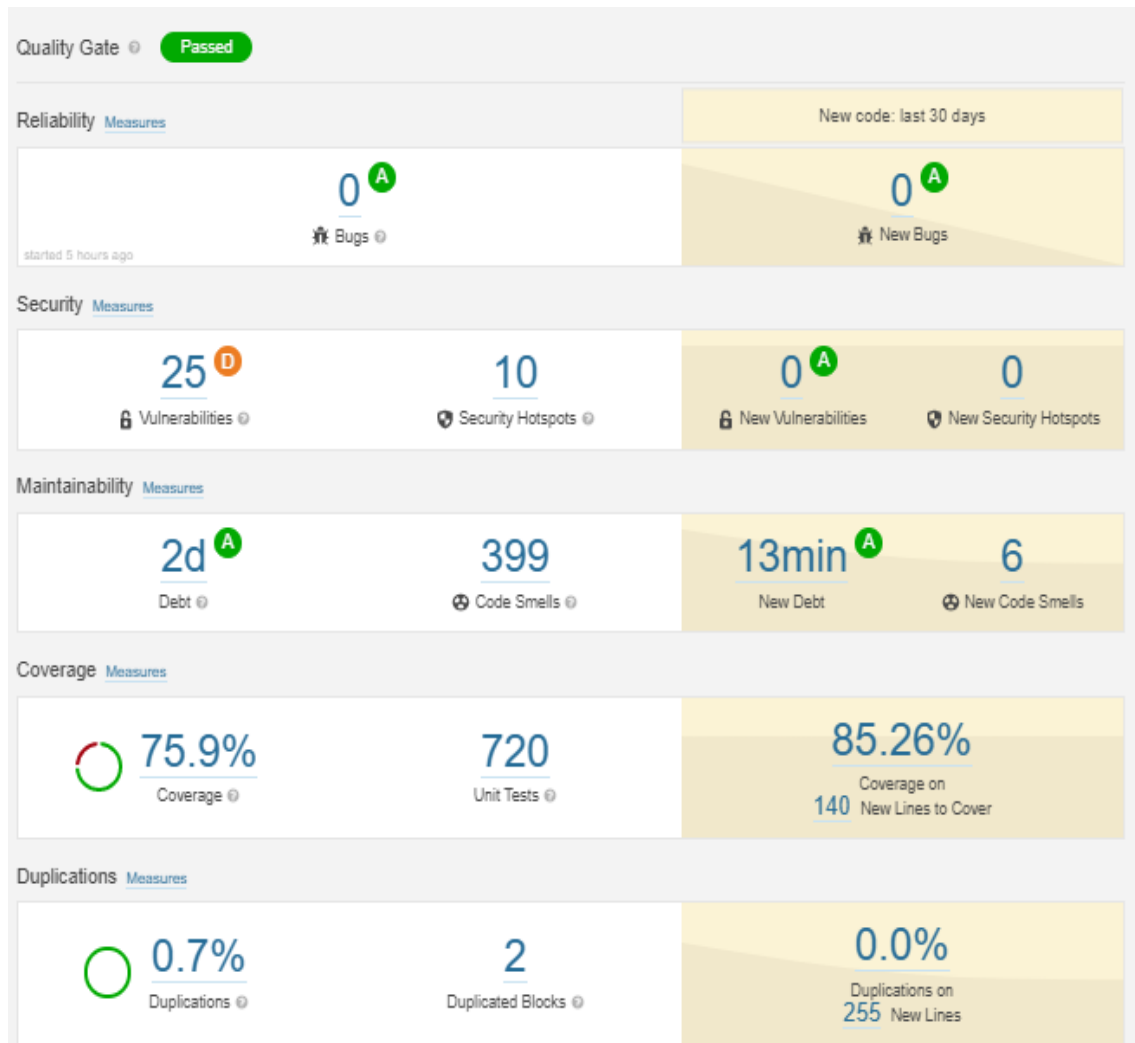
Complete the assertion. Why is this an issue?	2 months ago	L82	🔗	🔍	📄
🔧 Code Smell 🔴 Blocker 🔵 Open Not assigned 5min effort					tests
Complete the assertion. Why is this an issue?	2 months ago	L159	🔗	🔍	📄
🔧 Code Smell 🔴 Blocker 🔵 Open Not assigned 5min effort					tests
Complete the assertion. Why is this an issue?	2 months ago	L166	🔗	🔍	📄
🔧 Code Smell 🔴 Blocker 🔵 Open Not assigned 5min effort					tests

Simplemente se han añadido algunas líneas para que dichos métodos comprobasen cambios en la interfaz.

```
@Test
void shouldFindMedicineWithCorrectId() {
    Medicine med2 = this.medicineService.findMedicineById(2);
    assertThat(med2.getName()).contains("Pet Dalsy");
}
```

Resultados

Se ha optado por dar solución a los errores y resolver aquellos code smells que más se repetían para reducir la deuda técnica. Una vez refactorizadas todas las líneas de código involucradas, se han obtenido los siguientes resultados.



Los errores se han solucionado satisfactoriamente y se ha eliminado un total de 51 code smells. Esto ha supuesto una reducción de la deuda técnica a la mitad, así como una bajada del porcentaje de código duplicado. Por otra parte, la cobertura de pruebas asciende en décimas y se mantienen las vulnerabilidades de código.