

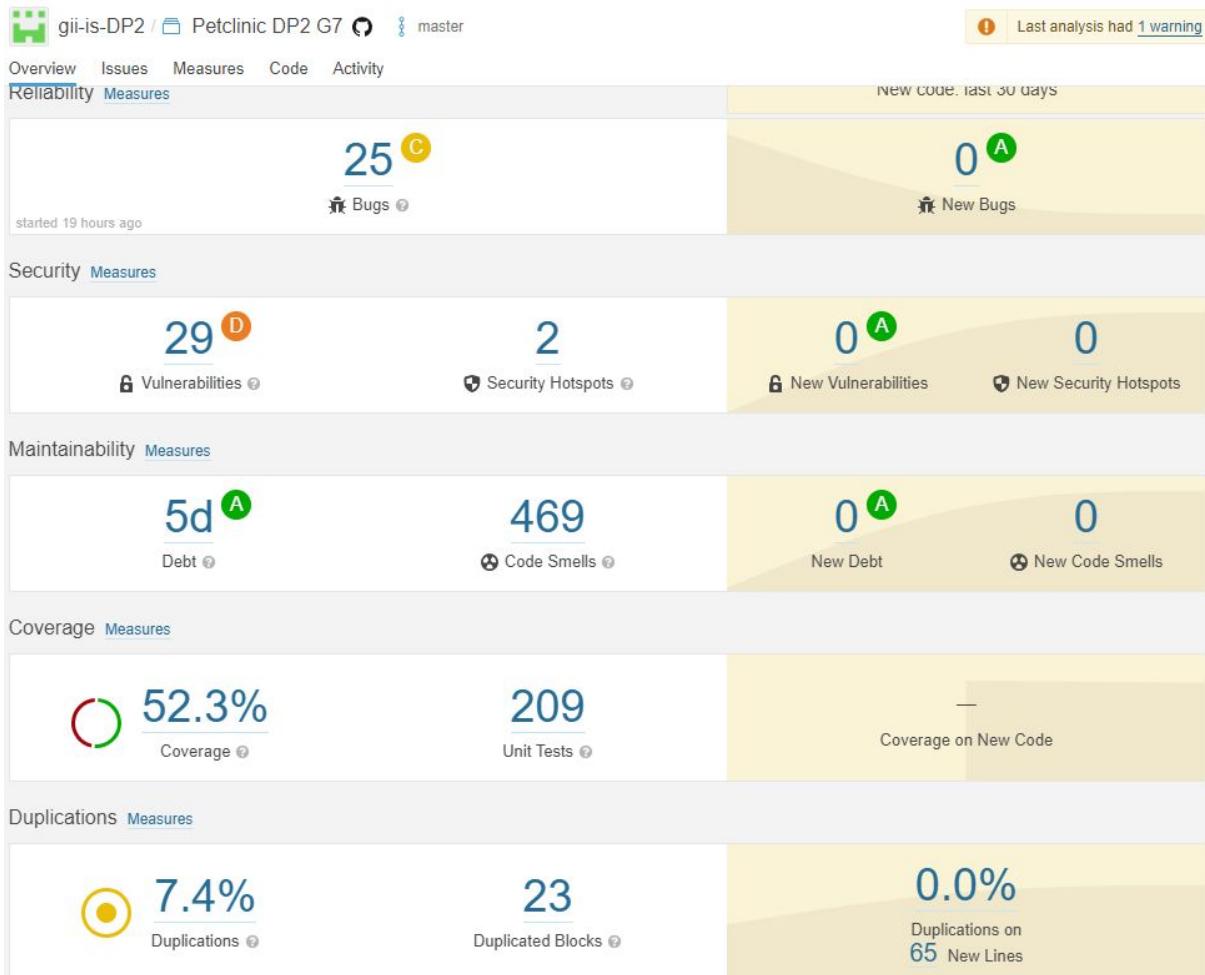
# REFACTORING 1 AMINE:

<b>INTRODUCTION</b>	<b>2</b>
<b>BAD SMELLS</b>	<b>3</b>
Code Smell on package org.springframework.samples.petclinic.web.GoogleOauthController:	3
Code Smell on package org.springframework.samples.petclinic.web.ReservationController:	7
Code Smell on package org.springframework.samples.petclinic.web.ReservationValidator:	9
Code Smell on package org.springframework.samples.petclinic.web.RoomController: 10	
Code Smell on package org.springframework.samples.petclinic.web.RoomValidator: 12	
Code Smell on package org.springframework.samples.petclinic.model.Reservation:	13
Code Smell on package org.springframework.samples.petclinic.model.Room:	13
Code Smell on package org.springframework.samples.petclinic.service.RoomServiceTest:	14
Code Smell on package org.springframework.samples.petclinic.service.VisitServiceTest:	15
<b>BUGS</b>	<b>16</b>
Bugs on package org.springframework.samples.petclinic.web.GoogleOauthController 16	
Bugs on package org.springframework.samples.petclinic.web.ReservationController: 17	
Bugs on package org.springframework.samples.petclinic.web.RoomController.	18
<b>CODE CHANGED:</b>	<b>19</b>
<b>RESULT FACTORING 1 By AMINE:</b>	<b>20</b>
Refactoring 2. Sergio.	22
<b>INTRODUCTION</b>	<b>23</b>
<b>BAD SMELLS</b>	<b>24</b>
<b>BUGS</b>	<b>31</b>
<b>RESULTS</b>	<b>32</b>

# INTRODUCTION

El objetivo de esta parte de la asignatura es llevar a cabo una refactorización del código, buscando obtener una mejora en las métricas proporcionadas por Sonar Cloud.

Nos encargaremos de solucionar ciertos malos olores, así como bugs o vulnerabilidades de los que nos ha indicado Sonar Cloud tras un primer análisis que se muestra en la siguiente captura:



Según el análisis tenemos una deuda técnica de 5 días, la cual va a ser difícil reducir demasiado, debido a las fechas del curso en que nos encontramos, pero sí intentaremos reducir lo máximo posible los malos olores (Code Smells), los Bugs, etc.

# BAD SMELLS

## 1. Code Smell on package

org.springframework.samples.petclinic.web.GoogleOauthController:

Para esta clase tenemos varios malos olores, la gran mayoría relacionados con imports que no están siendo usados.

<input type="checkbox"/> Remove this unused import 'java.io.BufferedReader'. <a href="#">Why is this an issue?</a>	last month ▾ L3				
Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 2min effort Comment					
<input type="checkbox"/> Remove this unused import 'java.io.IOException'. <a href="#">Why is this an issue?</a>	last month ▾ L4				
Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 2min effort Comment					
<input type="checkbox"/> Remove this unused import 'java.io.IOException'. <a href="#">Why is this an issue?</a>	last month ▾ L5				
Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 2min effort Comment					
<input type="checkbox"/> Remove this unused import 'java.security.GeneralSecurityException'. <a href="#">Why is this an issue?</a>	last month ▾ L6				
Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 2min effort Comment					
<input type="checkbox"/> Remove this unused import 'java.util.Date'. <a href="#">Why is this an issue?</a>	last month ▾ L8				
Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 2min effort Comment					

<input type="checkbox"/> Remove this unused import 'java.util.HashSet'. Why is this an issue?	last month ▾ L9	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'java.util.Set'. Why is this an issue?	last month ▾ L10	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'javax.servlet.ServletException'. Why is this an issue?	last month ▾ L13	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.beans.factory.annotation.Autowired'. Why is this an issue?	last month ▾ L18	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.samples.petclinic.model.Visit'. Why is this an issue?	last month ▾ L20	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.samples.petclinic.model.googleCalendar.End'. Why is this an issue?	last month ▾ L21	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.samples.petclinic.model.googleCalendar.Start'. Why is this an issue?	last month ▾ L23	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.samples.petclinic.resources.GoogleCalendarResources'. Why is this an issue?	last month ▾ L24	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.samples.petclinic.service.VisitService'. Why is this an issue?	last month ▾ L25	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.web.bind.annotation.PathVariable'. Why is this an issue?	last month ▾ L29	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.web.bind.annotation.PostMapping'. Why is this an issue?	last month ▾ L30	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove this unused import 'com.google.api.services.calendar.Calendar'. Why is this an issue?	last month ▾ L43	unused ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment		unused ▾
<input type="checkbox"/> Remove the "httptransport" field and declare it as a local variable in the relevant methods. Why is this an issue?	last month ▾ L57	pitfall ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 5min effort Comment		pitfall ▾
<input type="checkbox"/> Replace this use of System.out or System.err by a logger. Why is this an issue?	19 days ago ▾ L73	bad-practice, cert ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟥 Major ▾ ⚡ Open ▾ 🌐 achaghirc ▾ 10min effort Comment		bad-practice, cert ▾
<input type="checkbox"/> Define and throw a dedicated exception instead of using a generic one. Why is this an issue?	last month ▾ L78	cert, cue, error-handling ▾
<input type="checkbox"/> ⚠ Code Smell ▾ 🟥 Major ▾ ⚡ Open ▾ Not assigned ▾ 20min effort Comment		cert, cue, error-handling ▾

Todos los imports han sido eliminados correctamente. (17 eliminados)

## - Conjunto de malos olores solucionados:

<code>private static HttpTransport httptransport;</code>	Remove the "httptransport" field and declare it as a local variable in the relevant methods. Why is this an issue?	last month ▾ L57
<input type="checkbox"/> ⚠ Code Smell ▾ 🟢 Minor ▾ ⚡ Open ▾ Not assigned ▾ 5min effort Comment		pitfall ▾

Se está usando la variable httptransport como una variable estática fuera del único método que la usa.

Por tanto podemos declararla directamente dentro del método que la necesita:

```
private String authorize(HttpServletRequest request) throws Exception {
    HttpTransport httptransport;
    AuthorizationCodeRequestUrl auRequestUrl;
    if(flow == null) {
        Details web = new Details();
        web.setClientId(clientId);
```

## 1.1 REPLACE

```
@GetMapping(value="/singIn/google")
public RedirectView googleConnectionToOauth2(HttpServletRequest request, HttpServletResponse response) throws Exception {
    System.out.println(request);

Replace this use of System.out or System.err by a logger. Why is this an issue? 19 days ago ▾ L73 ⓘ
⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ 🚧 achaghirc ▾ 10min effort Comment
⌚ bad-practice, cert ▾

    return new RedirectView(authorize(request));
}
```

Esta línea de código la eliminamos directamente ya que no afecta a la funcionalidad de nuestro código.

## 1.2 DEDICATED EXCEPTION AND REMOVE UNUSED PARAMETERS

```
private String authorize(HttpServletRequest request) throws Exception {
    Define and throw a dedicated exception instead of using a generic one. Why is this an issue? last month ▾ L78 ⓘ
⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ Not assigned ▾ 20min effort Comment
⌚ cert, cwe, error-handling ▾

    Remove this unused method parameter "request". Why is this an issue? last month ▾ L78 [1] ⓘ
⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ Not assigned ▾ 5min effort Comment
⌚ cert, unused ▾

    AuthorizationCodeRequestUrl auRequestUrl;
    if(flow == null) {
        Details web = new Details();
        web.setClientId(clientId);
        web.setClientSecret(clientSecret);
        clientSecrets = new GoogleClientSecrets().setWeb(web);
        httptransport = GoogleNetHttpTransport.newTrustedTransport();

Make the enclosing method "static" or remove this set. Why is this an issue? last month ▾ L85 ⓘ
⌚ Code Smell ▾ ⚡ Critical ▾ ⚡ Open ▾ Not assigned ▾ 20min effort Comment
⌚ multi-threading ▾

        flow = new GoogleAuthorizationCodeFlow.Builder(httptransport, JSON_FACTORY, clientId, clientSecret, Collecti
    }
    auRequestUrl = flow.newAuthorizationUrl().setRedirectUri(redirectUrl);
    System.out.println("cal authorizationUrl->" + auRequestUrl);

Replace this use of System.out or System.err by a logger. Why is this an issue? last month ▾ L89 ⓘ
⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ Not assigned ▾ 10min effort Comment
⌚ bad-practice, cert ▾
```

El parámetro request no es usado en ninguno lado por lo tanto sobra, por otro lado la exception se puede especificar exactamente cual es ya que la emite el metodo newTrustedTrasnport(). El code smell de httptransport ya está solucionado en (1). Y el system.out se cambia por un log de Logger.

```
private String authorize() throws IOException, GeneralSecurityException {
    HttpTransport httptransport;
    AuthorizationCodeRequestUrl auRequestUrl;
    if(flow == null) {
        Details web = new Details();
        web.setClientId(clientId);
        web.setClientSecret(clientSecret);
        clientSecrets = new GoogleClientSecrets().setWeb(web);
        httptransport = GoogleNetHttpTransport.newTrustedTransport();
        flow = new GoogleAuthorizationCodeFlow.Builder(httptransport, JSON_FACTORY, clientId, clientSecret, Collecti
    }
    auRequestUrl = flow.newAuthorizationUrl().setRedirectUri(redirectUrl);
    Log.fine("cal authorizationUrl->" + auRequestUrl);
    return auRequestUrl.build();
```

## 1.3 COMMENTS

```
// TODO: handle exception
```

Complete the task associated to this TODO comment. Why is this an issue?

last month ▾ L101 🔍

Code Smell ▾  Info ▾  Open ▾ Not assigned ▾ Comment

🔗 cwe ▾

Esa línea de código se elimina ya que es un comentario.

## 1.4 UNUSED IMPORTS

Remove this unused import 'java.util.Arrays'. Why is this an issue?

3 months ago ▾ L5 🔍

Code Smell ▾  Minor ▾  Open ▾ Not assigned ▾ 2min effort Comment

🔗 unused ▾

Remove this unused import 'java.util.List'. Why is this an issue?

3 months ago ▾ L7 🔍

Code Smell ▾  Minor ▾  Open ▾ Not assigned ▾ 2min effort Comment

🔗 unused ▾

Remove this unused import 'java.util.Set'. Why is this an issue?

last month ▾ L9 🔍

Code Smell ▾  Minor ▾  Open ▾ Not assigned ▾ 2min effort Comment

🔗 unused ▾

Remove this unused import 'java.util.stream.Collectors'. Why is this an issue?

last month ▾ L10 🔍

Code Smell ▾  Minor ▾  Open ▾ Not assigned ▾ 2min effort Comment

🔗 unused ▾

Remove this unused import 'javax.websocket.serverPathParam'. Why is this an issue?

3 months ago ▾ L14 🔍

Code Smell ▾  Minor ▾  Open ▾ Not assigned ▾ 2min effort Comment

🔗 unused ▾

Remove this unused import 'org.springframework.samples.petclinic.repository.PetRepository'. Why is this an issue?

3 months ago ▾ L22 🔍

Code Smell ▾  Minor ▾  Open ▾ Not assigned ▾ 2min effort Comment

🔗 unused ▾

Remove this unused import 'org.springframework.security.core.Authentication'. Why is this an issue?

3 months ago ▾ L27 🔍

Code Smell ▾  Minor ▾  Open ▾ Not assigned ▾ 2min effort Comment

🔗 unused ▾

Remove this unused import 'org.springframework.validation.ObjectError'. Why is this an issue?

2 months ago ▾ L33 🔍

Code Smell ▾  Minor ▾  Open ▾ Not assigned ▾ 2min effort Comment

🔗 unused ▾

## 2. Code Smell on package org.springframework.samples.petclinic.web.ReservationController:

La gran mayoría de malos olores son imports que no están siendo usados:

<input type="checkbox"/> Remove this unused import 'java.util.Arrays'. Why is this an issue?	3 months ago	L5	%	▼
<input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment				unused ▾
<input type="checkbox"/> Remove this unused import 'java.util.List'. Why is this an issue?	3 months ago	L7	%	▼
<input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment				unused ▾
<input type="checkbox"/> Remove this unused import 'java.util.Set'. Why is this an issue?	last month	L9	%	▼
<input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment				unused ▾
<input type="checkbox"/> Remove this unused import 'java.util.stream.Collectors'. Why is this an issue?	last month	L10	%	▼
<input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment				unused ▾
<input type="checkbox"/> Remove this unused import 'javax.websocket.serverPathParam'. Why is this an issue?	3 months ago	L14	%	▼
<input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment				unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.samples.petclinic.repository.PetRepository'. Why is this an issue?	3 months ago	L22	%	▼
<input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment				unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.security.core.Authentication'. Why is this an issue?	3 months ago	L27	%	▼
<input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment				unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.validation.ObjectError'. Why is this an issue?	2 months ago	L33	%	▼
<input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment				unused ▾

La solución aplicada es eliminar todos estos imports.

### 2.1. RETURN INMEDIATO

```
public Collection<Pet> findAllPetsByOwnerId(HttpServletRequest request) {  
    Principal principal = request.getUserPrincipal();  
    String name = principal.getName();  
    Collection<Pet> pets = petService.findPetsByOwnerName(name);
```

Immediately return this expression instead of assigning it to the temporary variable "pets". Why is this an issue? 3 months ago L90 %

Code Smell ▾  Minor ▾  Open ▾ Not assigned ▾ 2min effort Comment

clumsy ▾

```
return pets;
```

Podemos devolver inmediatamente la expresión sin almacenarla en una variable.

```
@ModelAttribute("petsOfOwner")  
public Collection<Pet> findAllPetsByOwnerId(HttpServletRequest request) {  
    Principal principal = request.getUserPrincipal();  
    String name = principal.getName();  
    return petService.findPetsByOwnerName(name);
```

## 2.2 LITERALES DUPLICADOS

```
model.addAttribute( 1 "status", s);
```

Define a constant instead of duplicating this literal "status" 3 times. Why is this an issue?

2 months ago ▾ L107 %

⌚ Code Smell ▾ ⚡ Critical ▾ ⚡ Open ▾ Not assigned ▾ 8min effort Comment

⌚ design ▾

```
model.addAttribute( 2 "status", s);
```

```
modelMap.addAttribute("completedRoom", false);
modelMap.addAttribute( 3 "status", this.reservationService.findAllStatus());
return "reservations/updateReservationForm";
```

La solución es declarar una propiedad estática al principio de la clase y así ahorrarnos el tener tres declaraciones de "status", que significan lo mismo.

```
30 @Controller
31 public class ReservationController {
32
33     private static final String STATUS = "status";
```

y sustituyendo **STATUS** en todos los lugares de la clase donde tengamos "status".

Y lo mismo sucede con:

```
return "reservations/createReservationForm";
```

Define a constant instead of duplicating this literal "reservations/createReservationForm" 3 times. Why is this an issue?

2 months ago ▾ L109 %

⌚ Code Smell ▾ ⚡ Critical ▾ ⚡ Open ▾ Not assigned ▾ 8min effort Comment

⌚ design ▾

Y por lo tanto se soluciona de la misma forma, creando otra propiedad estática:

```
@Controller
public class ReservationController {
    private static final String STATUS = "status";
    private static final String CREATEROOMFORM = "reservations/createReservationForm";
```

Y sustituimos **CREATEROOMFORM** en todos los lugares de la clase donde está la sentencia "**reservations/createReservationForm**"

## 2.3 COMENTARIOS

```
// public Boolean validaciones(Optional<Room> r,Optional<Pet> p,LocalDate d,BindingResult result) {
```

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

3 months ago ▾ L112 %

⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ Not assigned ▾ 5min effort Comment

⌚ unused ▾

```
//     Boolean res = false;
//     Boolean differentTypes;
//     Boolean isValidDate;
//
//     differentTypes = r.get().getType() != p.get().getType();
```

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

3 months ago ▾ L117 %

⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ Not assigned ▾ 5min effort Comment

⌚ unused ▾

```
//     isValidDate = d.isBefore(LocalDate.now());
//     if(differentTypes) {
//         FieldError errorType = new FieldError("reservation", "pet", "Esta habitacion no puede acoger mascotas de tipo "+p.get());
//         result.addError(errorType);
//     }
//     if(isValidDate) {
//         FieldError errorType = new FieldError("reservation", "entryDate", "La fecha no puede ser superior a la fecha actual.");
//     }
//     return true;
// }
```

Con este código lo que hacemos es eliminarlo ya que no es necesario en nuestro proyecto.

## 2.4 Primitive Boolean

```
if(completedRoom) {
```

Use the primitive boolean expression here. [Why is this an issue?](#)

3 months ago ▾ L170 🔍

Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 5min effort Comment

pitfall ▾

Este mal olor se debe a que “ When boxed type `java.lang.Boolean` is used as an expression it will throw `NullPointerException` if the value is `null`. ”  
es decir, que al usar Boolean como expresión corremos el riesgo de que nos lance una excepción de `NullPointerException` si recibe null, esto se soluciona imponiendo que debe ser igual a true para que en caso de que sea null se vaya al else.

```
141     if(completedRoom.equals(true)) {
142         Status accepted = this.reservationService.findStatusById(2);
143         s.remove(accepted);
144         model.addAttribute("statusWithoutAccepted", s);
145     }else {
146         model.addAttribute(STATUS, s);
147     }
```

## 3. Code Smell on package org.springframework.samples.petclinic.web.ReservationValidator:

### 3.1 UNUSED IMPORTS

Remove this unused import 'org.springframework.web.bind.annotation.RequestParam'. Why is this an issue?

3 months ago ▾ L4 🔍

Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 2min effort Comment

unused ▾

Remove this unused import 'org.springframework.samples.petclinic.model.Room'. Why is this an issue?

3 months ago ▾ L11 🔍

Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 2min effort Comment

unused ▾

Remove this unused import 'org.springframework.samples.petclinic.service.ReservationService'. Why is this an issue?

3 months ago ▾ L13 🔍

Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 2min effort Comment

unused ▾

Remove this unused import 'org.springframework.util.StringUtils'. Why is this an issue?

3 months ago ▾ L15 🔍

Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 2min effort Comment

unused ▾

La solución es eliminar todos los imports no usados.

### 3.2 PRIMITIVE BOOLEAN

```
if(!ambasNull) {
```

Use the primitive boolean expression here. [Why is this an issue?](#)

3 months ago ▾ L49 🔍

Code Smell ▾  Minor ▾  Open ▾  Not assigned ▾ 5min effort Comment

pitfall ▾

Sucede lo mismo que en el apartado anterior (2.4) y la solución es la misma, obligar a que entre en ese if solo si **ambasNull** es igual a **false**.

```

if(jambasNull.equals(false)) {
    if(entryDate.isBefore(now) ) {
        errors.rejectValue("entryDate", "The entry d
}

```

### 3.3 Expression always null

```
if(exitDate.isBefore(now) || exitDate.isBefore(entryDate) || exitDate.equals(null)) {
```

Remove this expression which always evaluates to "false". Why is this an issue?

3 months ago ▾ L54 2 %

Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 10min effort Comment

cert, cwe, redundant, suspicious ▾

La sentencia exitDate.equals(null) es siempre falsa en nuestro código por lo tanto no tiene sentido tenerla ahí, así que la eliminamos.

## 4. Code Smell on package org.springframework.samples.petclinic.web.RoomController:

### 4.1 IMMEDIATELY RETURN

```

@ModelAttribute("types")
public Collection<PetType> allPetTypes() {
    Collection<PetType> petType = this.petService.findPetTypes();

```

Immediately return this expression instead of assigning it to the temporary variable "petType". Why is this last month ▾ L112 %  
an issue?

Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment

clumsy ▾

```
return petType;
```

En este método podemos devolver directamente la consulta sin tener que guardarla en una variable.

```

@ModelAttribute("types")
public Collection<PetType> allPetTypes() {
    return this.petService.findPetTypes();
}

```

### 4.2 RENAME

```

@POSTMapping(value = "/rooms/new")
public String processSaveRoom(@Valid final Room room, final BindingResult result, final ModelMap model) {
    RoomValidator roomValidator = new RoomValidator(this.roomService);

```

Rename "roomValidator" which hides the field declared at line 52. Why is this an issue?

last month ▾ L145 %

Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment

cert, pitfall, suspicious ▾

En otra línea tenemos otra declaración con el mismo nombre de roomValidator, por lo que debemos cambiar el nombre de ésta. En esta clase se da dos veces este mismo Code Smell, en "processSaveRoom" y en "processUpdateRoom"

```

142 @PostMapping(value = "/rooms/new")
143 public String processSaveRoom(@Valid final Room room, final BindingResult
144     RoomValidator roomVal = new RoomValidator(this.roomService);
145     roomVal.validate(room, result);

```

```

83@     @PostMapping(value = "/rooms/{roomId}/edit")
84     public String processUpdateRoom(@Valid final Room room, @PathVariable("room")
85             RoomValidator roomVal = new RoomValidator(this.roomService, roomId);
86             roomVal.validate(room, result);
87

```

#### 4.3 DUPLICATED LITERAL

```

if (result.hasErrors()) {
    model.addAttribute("message", "Room not created");

Define a constant instead of duplicating this literal "message" 3 times. Why is this an issue? 3 months ago L148 3 %
🔗 Code Smell 🔗 Critical 🔗 Open 🔗 Not assigned 8min effort Comment 🔍 design

```

Para solucionarlo nos creamos una constante estática al principio de la clase.

```

@Controller
public class RoomController {

    private static final String VIEWS_ROOM_CREATE_OR_UPDATE_F
    private static final String MESSAGE = "message";
```

Y sustituyendo esa variable estática donde anteriormente teníamos la sentencia "message."

```

if (result.hasErrors()) {
    model.addAttribute(MESSAGE, "Room not created");
    model.addAttribute("room", room);
    return "rooms/createOrUpdateRoomForm";
} else {
    this.roomService.saveRoom(room);
    model.addAttribute(MESSAGE, "Room successfully created");
}
```

## 5. Code Smell on package org.springframework.samples.petclinic.web.RoomValidator:

### 5.1 UNUSED IMPORTS AND UNUSED FIELD

<input type="checkbox"/> Remove this unused import 'java.time.LocalDate'. Why is this an issue?  <input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment	last month ▾ L3 %   unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.samples.petclinic.model.Pet'. Why is this an issue?  <input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment	last month ▾ L7 %   unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.samples.petclinic.model.Reservation'. Why is this an issue?  <input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment	last month ▾ L8 %   unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.samples.petclinic.service.PetService'. Why is this an issue?  <input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment	last month ▾ L10 %   unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.validation.FieldError'. Why is this an issue?  <input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Minor ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 2min effort Comment	last month ▾ L13 %   unused ▾
<input type="checkbox"/> Remove this unused "REQUIRED" private field. Why is this an issue?  <input checked="" type="checkbox"/> Code Smell ▾ <input checked="" type="checkbox"/> Major ▾ <input type="checkbox"/> Open ▾ Not assigned ▾ 5min effort Comment	last month ▾ L18 %   unused ▾

Arreglamos este error eliminando los import no usados e igual con el campo REQUIRED que no es usado.

## 6. Code Smell on package org.springframework.samples.petclinic.model.Reservation:

### 6.1 Unused Imports

```
import javax.persistence.OneToOne;
```

Remove this unused import 'javax.persistence.OneToOne'. Why is this an issue?

3 months ago ▾ L9 %

⌚ Code Smell ▾ ⚡ Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment

⌚ unused ▾

El problema se soluciona eliminando el import.

## 7. Code Smell on package org.springframework.samples.petclinic.model.Room:

### 7.1 UNUSED AND DUPLICATED IMPORTS

```
import java.util.stream.Collectors;
```

Remove this unused import 'java.util.stream.Collectors'. Why is this an issue?

last month ▾ L8 %

⌚ Code Smell ▾ ⚡ Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment

⌚ unused ▾

```
import java.util.List;  
import java.util.Set;
```

Remove this duplicated import. Why is this an issue?

last month ▾ L11 %

⌚ Code Smell ▾ ⚡ Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment

⌚ unused ▾

Para arreglar esto simplemente eliminamos los imports.

### 7.2 REPLACE THE TYPE SPECIFICATION WITH THE OPERATOR (<>).

```
public Set<Pet> getPetsInternal() {  
    if (this.pets == null) {  
        this.pets = new HashSet<Pet>();
```

Replace the type specification in this constructor call with the diamond operator ("<>"). Why is this an issue?

3 months ago ▾ L83 %

⌚ Code Smell ▾ ⚡ Minor ▾ ⚡ Open ▾ Not assigned ▾ 1min effort Comment

⌚ clumsy ▾

```
}  
return this.pets;  
}
```

En HashSet<Pet>() lo cambiamos por HashSet<>().

```
public Set<Pet> getPetsInternal() {  
    if (this.pets == null) {  
        this.pets = new HashSet<>();  
    }  
    return this.pets;  
}
```

8. Code Smell on package  
org.springframework.samples.petclinic.service.RoomServiceTest:

### 8.1 COMPLETE THE ASSERT.

```
@Test  
void shoudFindRoomByCorrectId() {  
    Room room = this.roomService.findRoomById(1);  
    assertThat(room.getName()).equals("Room1"));
```

Complete the assertion. [Why is this an issue?](#)

2 months ago ▾ L30 %

 Code Smell ▾  Blocker ▾  Open ▾  Not assigned ▾ 5min effort Comment

 tests ▾

El assertThat no esta comprobando nada ya que el equals se encuentra dentro del paréntesis del assertThat() en vez de estar fuera  
assertThat().isEqualTo();

```
@Test  
void shoudFindRoomByCorrectId() {  
    Room room = this.roomService.findRoomById(1);  
    assertThat(room.getName()).isEqualTo("Room1"));  
}
```

9. Code Smell on package  
org.springframework.samples.petclinic.service.VisitServiceTest:

## 9.1 DELETE PUBLIC ON VOID AND SWAP ATTRIBUTES IS ASSERT

The screenshot shows a SonarQube interface with a code review comment for a Java test method. The code is as follows:

```
@Test  
public void testMyVisits() {  
  
    Boolean visits = visitService.visitFind().toString().contains("Visit");  
  
    assertEquals(visits, true);  
  
    Swap these 2 arguments so they are in the correct order: expected value, actual value. Why is this an issue?  
    assertNotEquals(visits, false);  
  
    Swap these 2 arguments so they are in the correct order: expected value, actual value. Why is this an issue?  
}  
}
```

Comments from SonarQube:

- Remove this 'public' modifier. Why is this an issue? (2 months ago, L23, 1%, Code Smell, Info, Open, Not assigned, 2min effort, Comment, junit, tests)
- Swap these 2 arguments so they are in the correct order: expected value, actual value. Why is this an issue? (2 months ago, L27, 1%, Code Smell, Major, Open, Not assigned, 2min effort, Comment, junit, suspicious, tests)
- Swap these 2 arguments so they are in the correct order: expected value, actual value. Why is this an issue? (2 months ago, L28, 1%, Code Smell, Major, Open, Not assigned, 2min effort, Comment, junit, suspicious, tests)

Tenemos que eliminar el public en la declaración del método e intercambiar los parámetros dentro de los assertEquals ya que assertEquals(expected value, actual value) por lo que nosotros el valor que esperamos es true y el que tenemos actual es visits e igual con assertNotEquals.

```
@Test  
void testMyVisits() {  
  
    Boolean visits = visitService.visitFind().toString().contains("Visit");  
  
    assertEquals(true, visits);  
    assertNotEquals(false, visits);  
}
```

# BUGS

10. Bugs on package  
org.springframework.samples.petclinic.web.GoogleOAuthController

## 10.1 REMOVE MISLEADING MUTABLE OR MAKE IT STATIC OR FINAL

```
GoogleClientSecrets clientSecrets;
```

Remove this misleading mutable servlet instance field or make it "static" and/or "final" Why is this an issue?

last month ▾ L60 %

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 30min effort Comment

cert, multi-threading, struts ▾

```
GoogleAuthorizationCodeFlow flow;
```

Remove this misleading mutable servlet instance field or make it "static" and/or "final" Why is this an issue?

last month ▾ L61 %

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 30min effort Comment

cert, multi-threading, struts ▾

```
Credential credential;
```

Remove this misleading mutable servlet instance field or make it "static" and/or "final" Why is this an issue?

last month ▾ L62 %

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 30min effort Comment

cert, multi-threading, struts ▾

```
@Value("${google.client.client-id}")
private String clientId;
```

Remove this misleading mutable servlet instance field or make it "static" and/or "final" Why is this an issue?

last month ▾ L65 %

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 30min effort Comment

cert, multi-threading, struts ▾

```
@Value("${google.client.client-secret}")
private String clientSecret;
```

Remove this misleading mutable servlet instance field or make it "static" and/or "final" Why is this an issue?

last month ▾ L67 %

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 30min effort Comment

cert, multi-threading, struts ▾

Este bug no se puede resolver, al menos no fácilmente ya que en nuestro caso si lo declaramos como static el flujo de autenticación con Google falla.

## 11. Bugs on package

org.springframework.samples.petclinic.web.ReservationController:

### 11.1 CHANGE == BY EQUALS.

```
@GetMapping(value = "/rooms/{roomId}/{ownerId}/reservation/{reservationId}/edit")
public String processInitUpdateForm(@PathVariable("roomId") int roomId, @PathVariable("ownerId") int ownerId,
    @PathVariable("reservationId") int reservationId, Model model) {
    Boolean completedRoom = false;
    Room r = this.roomService.findRoomById(roomId);
    if(r.getReservations()!= null) {
        Integer numReservationsAccepted = (int) r.getReservations().stream().filter(x->x.getStatus().getName().equals("ACCEPTED")).c
        completedRoom = numReservationsAccepted == r.getCapacity();
```

Use the "equals" method if value comparison was intended. Why is this an issue?

2 months ago ▾ L165 %

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment

cert, cwe ▾

```
model.addAttribute("completedRoom", numReservationsAccepted == r.getCapacity());
```

Use the "equals" method if value comparison was intended. Why is this an issue?

3 months ago ▾ L166 %

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment

cert, cwe ▾

Siempre que se vaya a comprobar que dos elementos son iguales es recomendable hacerlo con equals() siempre que no sea un valor numérico. (int, double, float)

```
134     if(r.getReservations()!= null) {
135         Integer numReservationsAccepted = (int) r.getReservations().stream().filter(x->x.getStatus()
136         completedRoom = numReservationsAccepted.equals(r.getCapacity()));
137         model.addAttribute("completedRoom", numReservationsAccepted.equals(r.getCapacity()));
138     }
```

## 12. Bugs on package org.springframework.samples.petclinic.web.RoomController.

### 12.1 NULLABLE ROOM

```
if ( room != null && !room.getReservations().isEmpty() ) {
    for (Reservation r : room.getReservations()) {
        if (!r.getStatus().getName().equals("ACCEPTED")) {
            this.roomService.delete(room);
            model.addAttribute("message", "Event Successfully deleted");
        }
    }
} else if (room != null && room.getReservations().isEmpty() ) {
    this.roomService.delete(room);
} else {

    model.addAttribute("roomNotDeleted", "The room named " + room.getName() + " cannot be removed because it has");
}
```

A "NullPointerException" could be thrown; "room" is nullable here. Why is this an issue?

last month ▾ L172 🔍

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 10min effort Comment

🔗 cert, cwe ▾

```
}
```

Si observamos la traza podemos observar que en el último else el atributo room.getName() nos puede dar null, por lo tanto no tiene mucho sentido ponerlo ahí. Por lo que la solución más sencilla es eliminarlo y dejar la frase como “The room cannot be removed ...”

### 12.2 SWAP == BY EQUALS

```
if (r.getReservations() != null) {
    Integer numReservationsAccepted = (int) r.getReservations().stream().filter(x -> x.getStatus().getName().equals("A
Boolean completedRoom = numReservationsAccepted == r.getCapacity();
```

Use the "equals" method if value comparison was intended. Why is this an issue?

3 months ago ▾ L205 🔍

Bug ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment

🔗 cert, cwe ▾

Resolvemos el error cambiando == por equals() por el mismo motivo que en el punto (11.1)

```
202     if (r.getReservations() != null) {
203         Integer numReservationsAccepted = (int) r.getReservations().stream().filter(x -> x.get
204         Boolean completedRoom = numReservationsAccepted.equals(r.getCapacity());
205         model.addAttribute("completedRoom", completedRoom);
```

## CODE CHANGED:

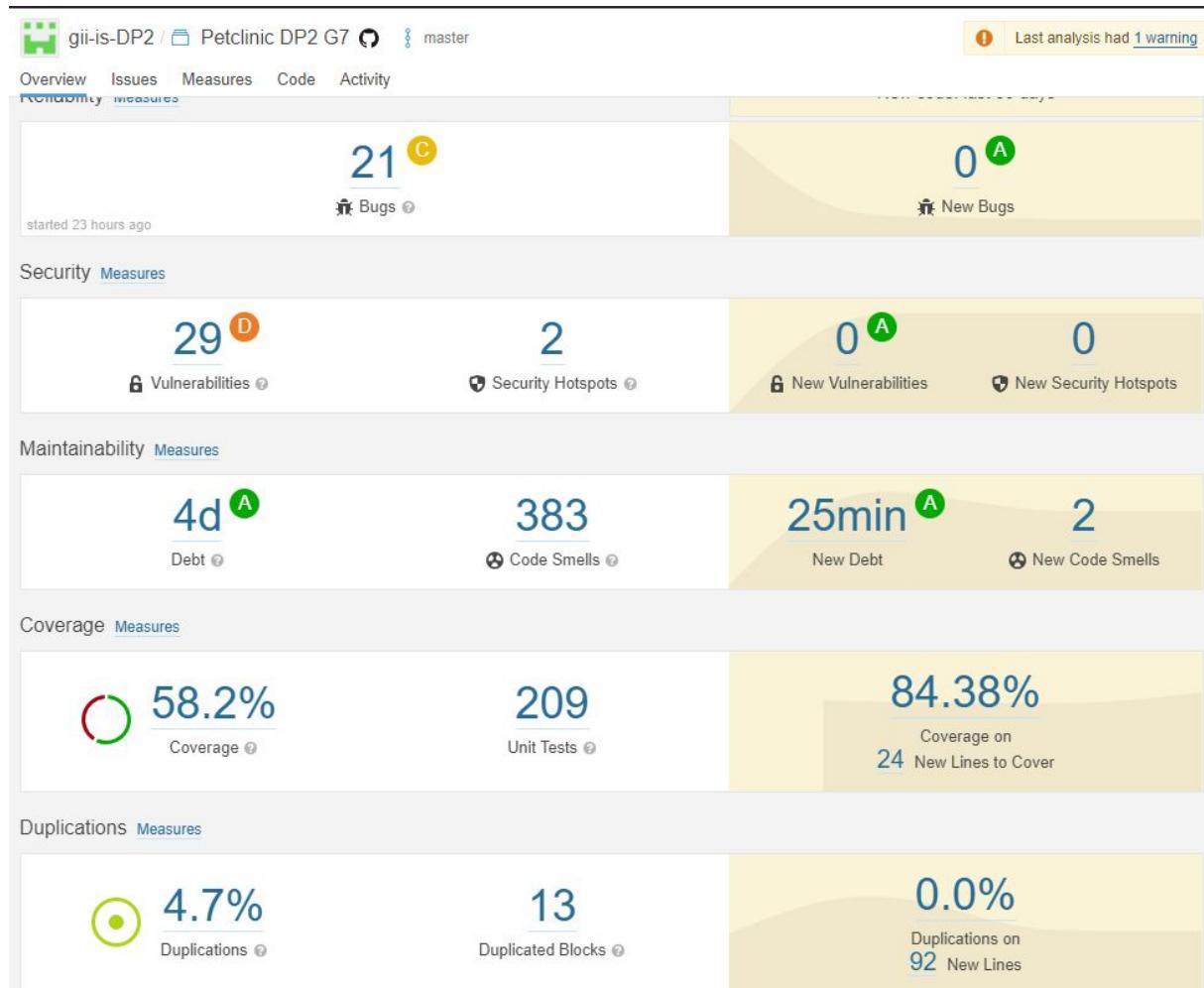
Aquí se indicarán las clases que has sido modificadas.

REFACTORING 1 (AMINE):

- +Reservation.java
- +Room.java
- +GoogleOauthController.java
- +ReservationController.java
- +RoomController.java
- +VisitService.java
- +RoomServiceTest.java
- +ReservationValidator.java
- +RoomValidator.java

# RESULT FACTORING 1 By AMINE:

Tras realizar la refactorización, vemos como hemos reducido el número de bugs de 25 a 21, además hemos reducido el número de code smells de 469 a 383, hemos reducido la deuda de 5 a 4 días.



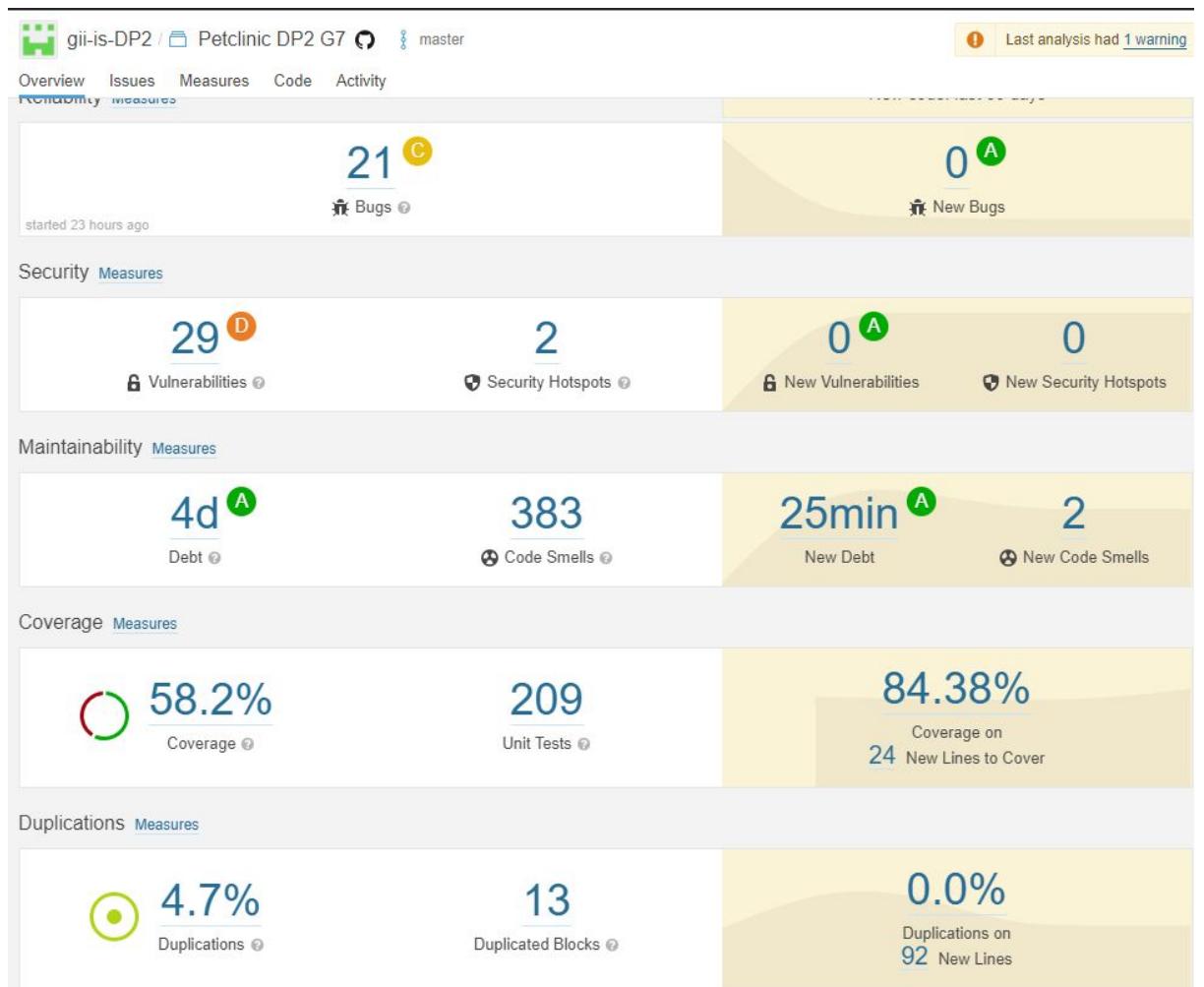


## **Refactoring 2. Sergio.**

# INTRODUCTION

Continuaremos con una nueva refactorización del código para seguir mejorando en las métricas proporcionadas por Sonar Cloud.

El estado del proyecto luego de la primera refactorización es el siguiente:



# BAD SMELLS

En esta refactorización procederemos a eliminar los malos olores por orden de “peligrosidad” según sonar.

## Categoría INFO:

<input type="checkbox"/> Complete the task associated to this TODO comment. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ Not assigned ▾ Comment	last month ▾ L39 % T- cwe ▾
src/.../samples/petclinic/service/VisitService.java  <input type="checkbox"/> Complete the task associated to this TODO comment. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ Not assigned ▾ Comment	2 months ago ▾ L23 % T- cwe ▾
src/.../samples/petclinic/web/GoogleOAuthController.java  <input type="checkbox"/> Complete the task associated to this TODO comment. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ Not assigned ▾ Comment	last month ▾ L84 % T- cwe ▾
src/.../petclinic/integrationApi/GoogleCalendarAPITest.java  <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ achaghirc ▾ 2min effort Comment	19 days ago ▾ L38 % T- junit, tests ▾
<input type="checkbox"/> Remove this 'public' modifier. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ achaghirc ▾ 2min effort Comment	19 days ago ▾ L59 % T- junit, tests ▾
<input type="checkbox"/> Remove this 'public' modifier. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ achaghirc ▾ 2min effort Comment	19 days ago ▾ L78 % T- junit, tests ▾
src/.../samples/petclinic/model/CauseValidatorTests.java  <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ Not assigned ▾ 2min effort Comment	2 months ago ▾ L16 % T- junit, tests ▾
src/.../samples/petclinic/service/CauseServiceTests.java  <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ Not assigned ▾ 2min effort Comment	2 months ago ▾ L23 % T- junit, tests ▾
src/.../samples/petclinic/service/DiagnosisServiceTests.java  <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ Not assigned ▾ 2min effort Comment	2 months ago ▾ L22 % T- junit, tests ▾
src/.../samples/petclinic/service/DonationServiceTests.java  <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ Not assigned ▾ 2min effort Comment	2 months ago ▾ L16 % T- junit, tests ▾
<input type="checkbox"/> Remove this 'public' modifier. Why is this an issue?  Code Smell ▾ Info ▾ Open ▾ Not assigned ▾ 2min effort Comment	2 months ago ▾ L22 % T- junit, tests ▾

Como podemos observar los primeros errores vienen dado a comentarios TO DO los cuales se han escrito automáticamente y se han olvidado de eliminar, a parte, también quitaremos todos los public los cuales no tienen ninguna función en el código y sonar los detecta como malos olores.

**-Clases mejoradas con malos olores de tipo INFO:**

VisitService.java  
GoogleOauthController.java  
GoogleCalendarApiTest.java  
CauseValidatorTest.java  
CauseServiceTest.java  
DiagnosisServiceTest.java  
DonationServiceTest.java

## Categoría MINOR:

src/.../petclinic/configuration/ExceptionHandlerConfiguration.java

<input type="checkbox"/> Remove this unused import 'org.springframework.http.ResponseEntity'. Why is this an issue?	3 months ago ▾ L6 % T▼
<input checked="" type="checkbox"/> Code Smell ▾ ⚡ Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment	unused ▾
<input type="checkbox"/> Remove this unused import 'org.springframework.validation.BindException'. Why is this an issue?	3 months ago ▾ L7 % T▼

src/.../samples/petclinic/model/Room.java

<input type="checkbox"/> Remove this unused import 'java.util.stream.Collectors'. Why is this an issue?	last month ▾ L8 % T▼
<input checked="" type="checkbox"/> Code Smell ▾ ⚡ Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment	unused ▾
<input type="checkbox"/> Remove this duplicated import. Why is this an issue?	last month ▾ L11 % T▼

src/.../repository/springdatajpa/SpringDataRoomRepository.java

<input type="checkbox"/> Remove this unused import 'org.springframework.dao.DataAccessException'. Why is this an issue?	last month ▾ L6 % T▼
<input checked="" type="checkbox"/> Code Smell ▾ ⚡ Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment	unused ▾

src/.../repository/springdatajpa/SpringDataUserRepository.java

<input type="checkbox"/> Remove this unused import 'org.springframework.data.repository.query.Param'. Why is this an issue?	3 months ago ▾ L6 % T▼
<input checked="" type="checkbox"/> Code Smell ▾ ⚡ Minor ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment	unused ▾

Los fallos de menor son igual de triviales que los de INFO, los cuales vamos a poder solucionar simplemente eliminando las importaciones duplicadas y las no usadas.

### -Clases mejoradas con malos olores de tipo Minor:

ExceptionHandlerConfiguration.java

Room.java

SpringDataRoomRepository.java

SpringDataUserRepository.java

## Categoría MAJOR:

src/.../petclinic/configuration/SecurityConfiguration.java	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	last month ▾ L35 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
src/.../samples/petclinic/repository/DiagnosisRepository.java	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	2 months ago ▾ L17 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
src/.../samples/petclinic/repository/UserRepository.java	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	3 months ago ▾ L11 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	3 months ago ▾ L15 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
src/.../repository/springdatajpa/SpringDataOwnerRepository.java	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	3 months ago ▾ L43 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	3 months ago ▾ L50 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
src/.../repository/springdatajpa/SpringDataUserRepository.java	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	3 months ago ▾ L14 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
src/.../repository/springdatajpa/SpringDataVetRepository.java	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	3 months ago ▾ L46 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
src/.../samples/petclinic/service/DiagnosisService.java	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	2 months ago ▾ L26 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
src/.../samples/petclinic/service/RoomService.java	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	3 months ago ▾ L23 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused
src/.../samples/petclinic/service/UserService.java	<input type="checkbox"/> This block of commented-out lines of code should be removed. Why is this an issue?	3 months ago ▾ L51 % T- Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment unused

Los errores de tipo major, al igual que los otros vistos anteriormente, se resuelven simplemente eliminando bloques de código repetido. Como podemos ir observando los malos olores del proyecto pueden parecer y son triviales, pero no por eso debemos dejarlos de lado ya que en su conjunto todos ellos al principio sumaban 5 días de deuda y poco a poco vamos mejorando ese número.

**-Clases mejoradas con malos olores de tipo Major:**

SecurityConfiguration.java  
DiagnosisRepository.java  
UserRepository.java  
SpringDataOwnerRepository.java  
SpringDataUserRepository.java  
SpringDataVetRepository.java  
DiagnosisService.java  
RoomService.java  
UserService.java

## Categoría CRITICAL:

src/.../samples/petclinic/resources/GoogleCalendarResources.java
<p>Rename this constant name to match the regular expression "[A-Z][A-Z0-9]*_[A-Z0-9]+*\$". Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 2min effort <input type="radio"/> Comment</p> <p>last month ▾ L15 % ⚡ convention ▾</p>
src/.../samples/petclinic/web/CauseController.java
<p>Define a constant instead of duplicating this literal "cause" 4 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 10min effort <input type="radio"/> Comment</p> <p>last month ▾ L88 % ⚡ design ▾</p>
src/.../samples/petclinic/web/DiagnosisController.java
<p>Define a constant instead of duplicating this literal "diagnosis" 3 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 8min effort <input type="radio"/> Comment</p> <p>2 months ago ▾ L46 % ⚡ design ▾</p>
src/.../samples/petclinic/web/DonationController.java
<p>Define a constant instead of duplicating this literal "donation" 3 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 8min effort <input type="radio"/> Comment</p> <p>2 months ago ▾ L128 % ⚡ design ▾</p>
<p>Define a constant instead of duplicating this literal "message" 3 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 8min effort <input type="radio"/> Comment</p> <p>3 months ago ▾ L180 % ⚡ design ▾</p>
src/.../samples/petclinic/web/GoogleCalendarEventController.java
<p>Define a constant instead of duplicating this literal "event" 3 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 8min effort <input type="radio"/> Comment</p> <p>last month ▾ L62 % ⚡ design ▾</p>
src/.../samples/petclinic/web/PetsController.java
<p>Define a constant instead of duplicating this literal "redirect:/owner/pets" 3 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 8min effort <input type="radio"/> Comment</p> <p>3 months ago ▾ L86 % ⚡ design ▾</p>
src/.../samples/petclinic/web/ReservationController.java
<p>Define a constant instead of duplicating this literal "redirect:/rooms/{roomId}" 3 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 8min effort <input type="radio"/> Comment</p> <p>2 months ago ▾ L125 % ⚡ design ▾</p>
src/.../samples/petclinic/web/RoomController.java
<p>Define a constant instead of duplicating this literal "redirect:/rooms/" 4 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 10min effort <input type="radio"/> Comment</p> <p>last month ▾ L131 % ⚡ design ▾</p>
src/.../samples/petclinic/web/RoomValidator.java
<p>Define a constant instead of duplicating this literal "This field is required" 4 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 10min effort <input type="radio"/> Comment</p> <p>last month ▾ L35 % ⚡ design ▾</p>
10 of 10 shown
<p>Define a constant instead of duplicating this literal "admin" 12 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 28min effort <input type="radio"/> Comment</p> <p>last month ▾ L40 % ⚡ design ▾</p>
<p>Define a constant instead of duplicating this literal "owner" 10 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 22min effort <input type="radio"/> Comment</p> <p>last month ▾ L41 % ⚡ design ▾</p>
<p>Define a constant instead of duplicating this literal "veterinarian" 5 times. Why is this an issue?</p> <p><input type="checkbox"/> Code Smell <input checked="" type="radio"/> Critical <input type="radio"/> Open <input type="radio"/> Not assigned <input type="radio"/> 12min effort <input type="radio"/> Comment</p> <p>last month ▾ L48 % ⚡ design ▾</p>

Estos son todos los malos olores de tipo critical que tiene el proyecto después de la primera refactorización, los cuales los dividimos en dos tipos, el primero es por un fallo al nombrar las constantes estáticas en minúsculas, por lo cual las pondremos en mayúsculas. El segundo tipo de fallos puede parecer igual de trivial pero no lo es, ya que consiste en crear constantes estáticas en los métodos para evitar que se repita el código en ocasiones hasta doce veces.

```

private static final String ADMIN = "admin";
private static final String OWNER = "owner";
private static final String VETERINARIAN = "veterinarian";
@Autowired

```

Este es un ejemplo de como se solucionaría el problema, simplemente cambiando la el nombre de la variable por su valor a lo largo del código, pero, al hacer estos cambios y volver a pedirle a sonar que analice el código nos muestra nuevos fallos en estas mismas clases los cuales no habían aparecido antes.

The screenshot shows three separate audit findings from the OWASP A6 tool:

- Reorder the URL patterns from most to less specific, the pattern "/owner/pets/\*\*" should occurs before "/owner/\*\*". Why is this an issue?** (3 minutes ago, L42)
- Reorder the URL patterns from most to less specific, the pattern "/vets/create" should occurs before "/vets/\*\*". Why is this an issue?** (3 minutes ago, L55)
- Reorder the URL patterns from most to less specific, the pattern "/vets/visit" should occurs before "/vets/\*\*". Why is this an issue?** (3 minutes ago, L56)

Each finding includes a checkbox for reporting, a severity level (Critical), and a status (Open). The last two also show the reporter's name (Álvaro Frias) and effort (10min).

Estos fallos son de tipo Vulnerability y estaban por todas las clases con fallos de tipo critical, para solucionarlo lo que debemos hacer es reordenar los patterns con sentido ya que al hacerlos al principio no les prestamos la atención necesaria. Arreglando estos nuevos fallos conseguiremos mejorar el código de forma notable.

#### -Clases mejoradas con malos olores de tipo Critical:

GoogleCalendarResources.java

CauseController.java

DiagnosisController.java

DonationController.java

GoogleCalendarEventController.java

PetsController.java

ReservationController.java

RoomController.java

RoomValidator.java

# BUGS

The screenshot shows a bug tracking interface with two entries:

**Issue 1:** A blue-bordered box containing code and a message. The code is:

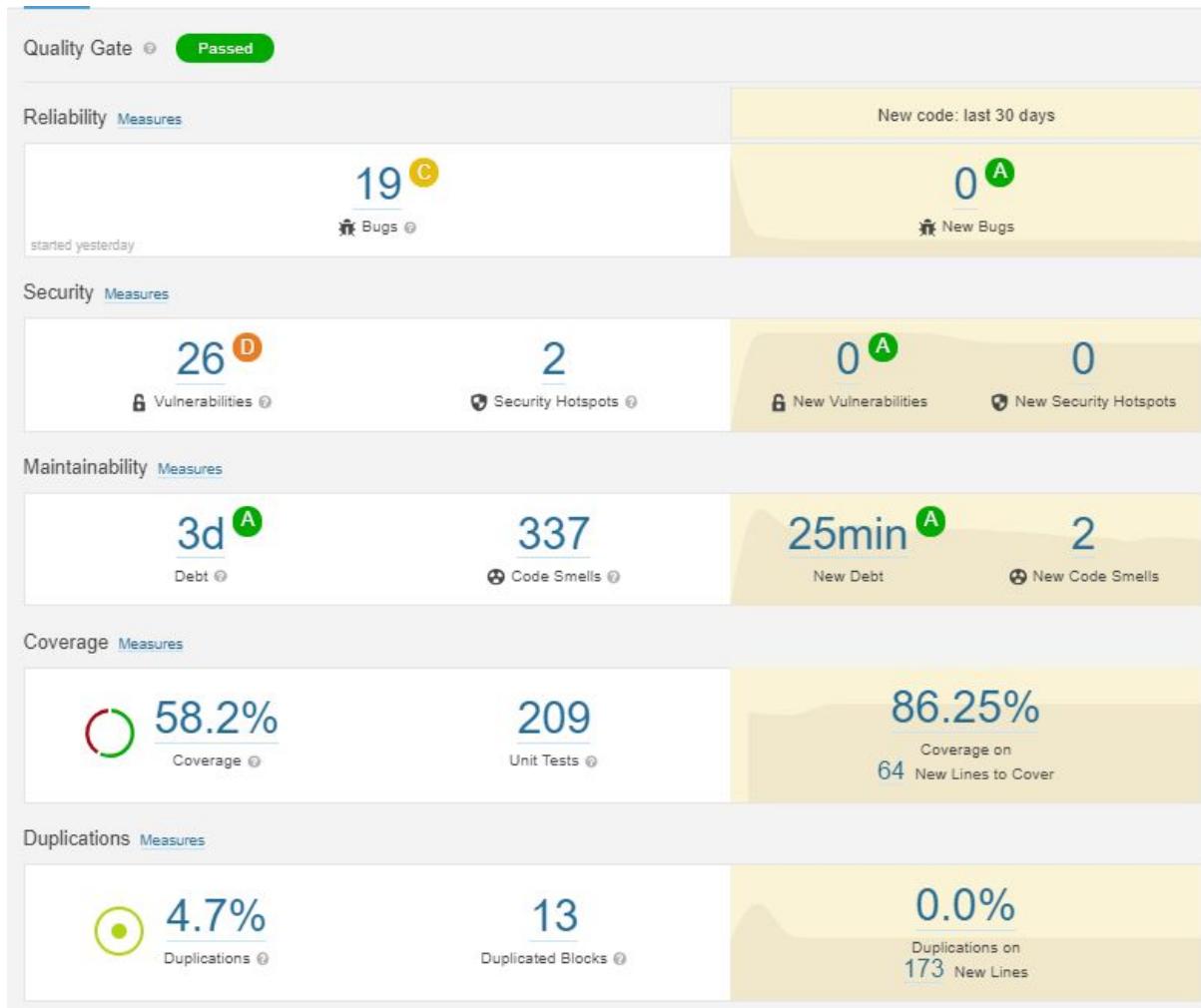
```
if (title == null || title == "") {  
    errors.rejectValue("title", "The title is empty", "The title is empty");  
}  
  
if (description == null || description == "") {
```

The message is: "Use the "equals" method if value comparison was intended. Why is this an issue?" Below the message are status indicators: Bug (red), Major (orange), Open (blue), Not assigned (grey), 5min effort, and Comment. To the right are "3 months ago L21" and a link icon.

**Issue 2:** A pink-bordered box containing the same code and message. It has the same status indicators and timestamp ("3 months ago L25") with its own link icon.

Los bugs que trataremos son estos dos, los cuales se solucionan cambiando el comparador `==` por `equals()` siempre y cuando el valor a comparar, como es en este caso, no sea de tipo numérico.

# RESULTS



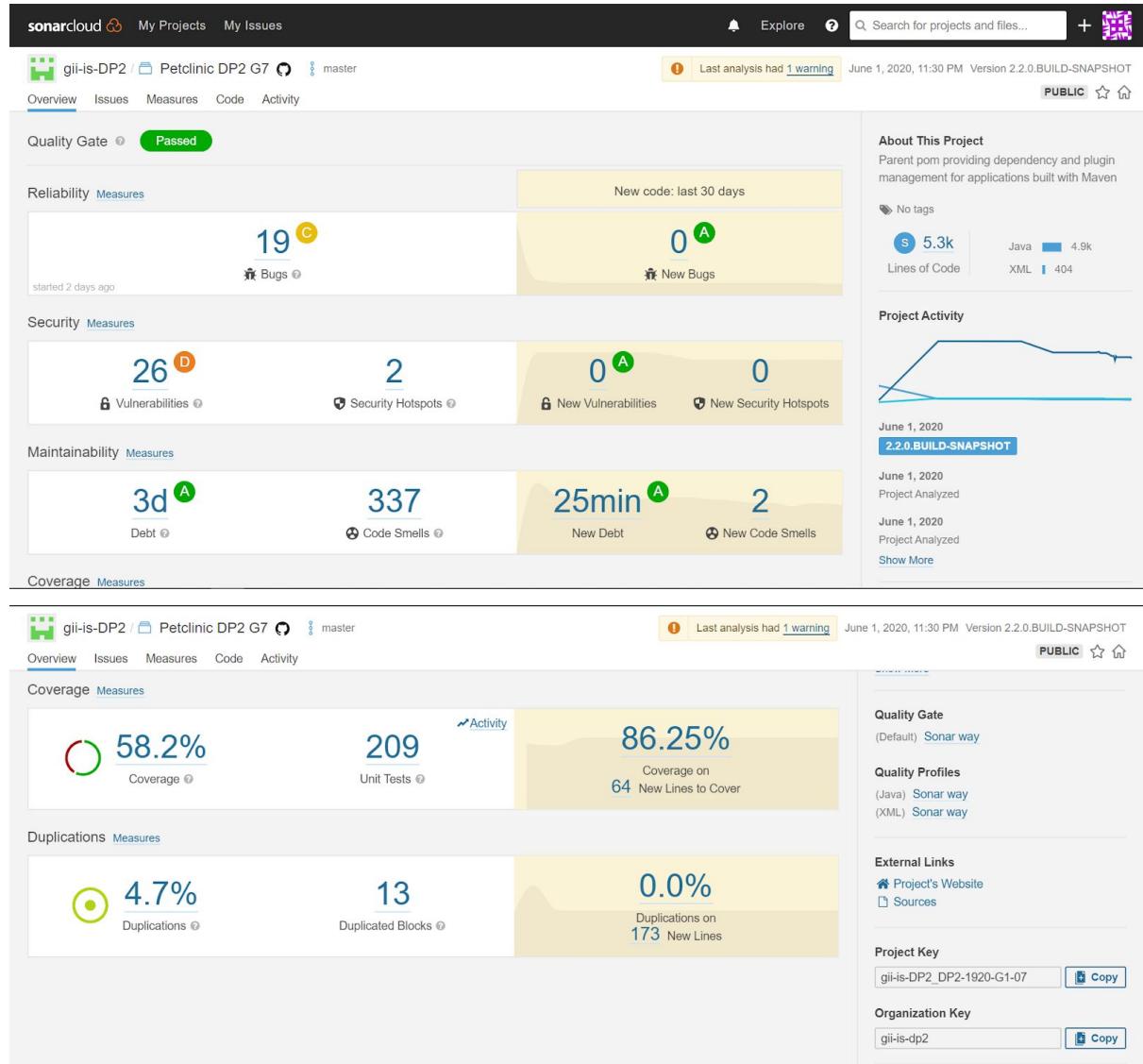
Como vemos en la nueva captura de pantalla los resultados en esta segunda refactorización son satisfactorios ya que hemos conseguido eliminar 2 bugs, 3 vulnerabilidades, 50 malos olores y un día de deuda.

# Refactoring 3 Alvaro:

## 1. Introducción:

Con la realización de esta parte de la asignatura intentaremos realizar una refactorización del código que mejore los errores mostrados por SonarCloud de nuestro sistema.

Inicialmente en SonarCloud nos encontramos con una pantalla en la que se nos muestra la cantidad de bugs, code smells, vulnerabilidades, etc. La vista es la siguiente:



Una vez visto el análisis, vemos que tenemos una deuda técnica de 3 días que por desgracia, debido a la altura de curso a la que estamos, va a ser difícil de reducir aunque intentaremos mejorar en lo que podamos los otros aspectos.

## 2. Code Smells:

Vamos a empezar a arreglar distintos errores de CodeSmells señalados por SonarCloud. Los arreglos han sido los siguientes:

### 2.1. Arreglos en CauseController:

SonarCloud nos muestra este BadSmell en la clase de CauseController.

```
@GetMapping(path = "/cause")
public String showCausesList(final ModelMap model, final HttpServletRequest request) {
    Principal principal = request.getUserPrincipal();
    String userName = principal.getName();
    model.put("userName", userName);

    Collection<Cause> causes = this.causeService.findAcceptedCauses();
    Collection<Cause> causasValidas = new ArrayList<Cause>();
```

Replace the type specification in this constructor call with the diamond operator ("<>"). Why is this an issue?

2 months ago ▾ L62 🔍

Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 1min effort Comment

clumsy ▾

```
Localdate now = LocalDate.now();
for (Cause c : causes) {
    if (c.getDeadline().isAfter(now)) {
        causasValidas.add(c);
    }
}
model.addAttribute(CAUSE, causasValidas);
return "causes/causesList";
}
```

Como podemos ver el error es simple, solamente hay que cambiar ese <Cause> por <>. Lo arreglamos y quedaría así:

```
@GetMapping(path = "/cause")
public String showCausesList(final ModelMap model, final HttpServletRequest request) {
    Principal principal = request.getUserPrincipal();
    String userName = principal.getName();
    model.put("userName", userName);

    Collection<Cause> causes = this.causeService.findAcceptedCauses();
    Collection<Cause> causasValidas = new ArrayList<>();
    LocalDate now = LocalDate.now();
    for (Cause c : causes) {
```

### 2.2. Arreglos en los validadores

Agrupo todos los validadores (CauseValidator, VisitValidator y DiagnosisValidator) en este apartado dado que se ha repetido el mismo bad smell en todos los que he mejorado. El fallo más repetido es el siguiente:

```

if (deadline != null) {
    Boolean posterior = deadline.isBefore(now);
    if (posterior) {

```

Use the primitive boolean expression here. Why is this an issue?  
Code Smell Minor Open Álvaro Fries 5min effort Comment

3 months ago ▾ L37 🔍

pitfall ▾

```

        errors.rejectValue("deadline", "The deadline is before the current date", "The deadline is before the cur
    }
}
```

Hemos repetido este error en los tres validator y para solucionarlo deberíamos, en vez de primero inicializar y luego usar ese parámetro en el if(), añadir ese "deadline.isBefore(now)" dentro del if y así nos ahorraríamos una línea de código innecesaria. Quedaría tal que así:

```

if (deadline != null) {
    if (deadline.isBefore(now)) {
        errors.rejectValue("deadline", "The deadline is before the current date", "The deadline is before the cur
    }
} else {
    errors.rejectValue("deadline", "The deadline is empty", "The deadline is empty");
}

```

Nuevo Modo ▾ Aj  
Selecciona el modo de recorte mediante el botón Nuevo.

Además, en el DonationValidator tenemos imports que no son usados. SonarCloud nos lo muestra de la siguiente manera:

Petclinic DP2 G7 src/.../samples/petclinic/web/DonationValidator.java See all issues i

```

1 serpe... package org.springframework.samples.petclinic.web;
2
3 import org.springframework.beans.factory.annotation.Autowired;

```

Remove this unused import 'org.springframework.beans.factory.annotation.Autowired'. Why is this an issue?  
Code Smell Minor Open Not assigned 2min effort Comment unused ▾

```

4 import org.springframework.samples.petclinic.model.Cause;
5
6 Remove this unused import 'org.springframework.samples.petclinic.model.Cause'. Why is this an issue?  

Code Smell Minor Open Not assigned 2min effort Comment unused ▾
7
8 import org.springframework.validation.Errors;
9 import org.springframework.validation.Validator;

```

La solución sería quitar estos imports de nuestro código dado que son innecesarios. El código quedaría así una vez eliminados estos imports:

```

1
2 package org.springframework.samples.petclinic.web;
3
4 import org.springframework.samples.petclinic.model.Donation;
5 import org.springframework.validation.Errors;
6 import org.springframework.validation.Validator;
7
8 public class DonationValidator implements Validator {
9
10    @Override

```

## 2.2. Quitar comentarios en varias clases

Volvemos a reunir varios cambios en un sólo apartado dado que es uno de los errores más comunes en los Code Smells y se repite en numerosas clases de nuestro proyecto. He realizado modificaciones en estas clases: DonationController, DiagnosisController, OwnerController, PetController, StatusFormatter y VetController. Estos code smells tienen una fácil solución, simplemente hay que quitarlos de nuestros sistemas ya que son líneas de código sin utilidad y solo hace que nuestro sistema sea menos legible. Expondré cada uno a continuación con lo mostrado en SonarCloud y como quedaría nuestro código una vez arreglado:

### DonationController

-Antes

```
// @ModelAttribute("causes")
// public Cause findCauseById(HttpServletRequest request){
```

This block of commented-out lines of code should be removed. [Why is this an issue?](#)

3 months ago ▾ L46 🔍

[Code Smell](#) ▾ [Major](#) ▾ [Open](#) ▾ [Not assigned](#) ▾ [5min effort](#) [Comment](#)

[unused](#) ▾

```
// String causeId = request.getAttribute("causeId").toString();
// return this.causeService.findCauseById(Integer.parseInt(causeId));
//
```

This block of commented-out lines of code should be removed. [Why is this an issue?](#)

3 months ago ▾ L50 🔍

[Code Smell](#) ▾ [Major](#) ▾ [Open](#) ▾ [Not assigned](#) ▾ [5min effort](#) [Comment](#)

[unused](#) ▾

-Después

```

@RequestMapping("/cause/{causeId}/donations")
public class DonationController {

    private static final String DONATION      = "donation";
    private static final String MESSAGE       = "message";

    @Autowired
    private DonationService     donationService;
    @Autowired
    private CauseService        causeService;

    @InitBinder(DonationController.DONATION)
    public void initDonationBinder(final WebDataBinder dataBinder) {
        dataBinder.setValidator(new DonationValidator());
    }
}

```

## DiagnosisController

-Antes

```

@PostMapping(value = "/vet/{vetId}/diagnosis")
public String processCreateDiagnosis(@Valid final Diagnosis diagnosis, final BindingResult result, final ModelMap m
    DiagnosisValidator diagnosisValidator = new DiagnosisValidator(this.diagnosisService);
    diagnosisValidator.validate(diagnosis, result);
    //      if (diagnosis.getDate() == null) {

This block of commented-out lines of code should be removed. Why is this an issue? last month ▾ L56 🔍
⌚ Code Smell 🔴 Major 🟡 Open 👤 Álvaro Frías 5min effort Comment 🏷️ unused
//          result.rejectValue("date", "Date must not be null");
//      }

```

-Después

```

@PostMapping(value = "/vet/{vetId}/diagnosis")
public String processCreateDiagnosis(@Valid final Diagnosis diagnosis, final BindingResult result) {
    DiagnosisValidator diagnosisValidator = new DiagnosisValidator(this.diagnosisService);
    diagnosisValidator.validate(diagnosis, result);

    if (result.hasErrors()) {
        model.put("message", "Diagnosis not created");
        model.put(DiagnosisController.DIAGNOSIS, diagnosis);
        return DiagnosisController.VIEWS_DIAGNOSIS_CREATE_FORM;
    } else {

```

## PetController

-Antes

```

    ...
    /*
     * @ModelAttribute("pet")
     * public Pet findPet(@PathVariable("petId") Integer petId) {

```

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

3 months ago ▾ L76 ⚙

Code Smell ▾ Major ▾ Open ▾ Álvaro Frías ▾ 5min effort Comment

unused ▾

```

        * Pet result=null;
        * if(petId!=null)
        * result=this.clinicService.findPetById(petId);
        * else
        * result=new Pet();
        * return result;
        *
    */

```

!6...

-Después

```

    ...
    return this.ownerService.findOwnerByOwnerId(ownerId),
72 }
73
74@ | @InitBinder("owner")
75 |     public void initOwnerBinder(final WebDataBinder dataBinder)
76 |         dataBinder.setDisallowedFields("id");
77 |
78
79@ | @InitBinder("pet")
80 |     public void initPetBinder(final WebDataBinder dataBinder) {

```

## StatusFormatter

-Antes

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.format.Formatter;
import org.springframework.samples.petclinic.model.Status;
import org.springframework.samples.petclinic.service.CauseService;

```

Remove this unused import 'org.springframework.samples.petclinic.service.CauseService'. Why is this an issue?

Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment

unused ▾

```

import org.springframework.stereotype.Component;

```

-Después

```
1 PetController.java  2 PetTypeFormatter.java  3 StatusFormatter.java
4 package org.springframework.samples.petclinic.web;
5
6 import java.text.ParseException;
7 import java.util.Collection;
8 import java.util.Locale;
9
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.format.Formatter;
12 import org.springframework.samples.petclinic.model.Status;
13 import org.springframework.samples.petclinic.service.ReservationService;
14 import org.springframework.stereotype.Component;
15
16 @Component
17 public class StatusFormatter implements Formatter<Status> {
```

# VetController

-Antes

```
57 56026...           // private VetValidator      vetValidator;
58 56026...           vetValidator;
59
60
61
62
This block of commented-out lines of code should be removed. Why is this an issue? 12 days ago ▾ L58 %
63
64
65
66
67
@Code Smell ▾ ⚠ Major ▾ ○ Open ▾ Not assigned ▾ 5min effort Comment
🔗 unused ▾
68
69
70 56026...
This block of commented-out lines of code should be removed. Why is this an issue? 12 days ago ▾ L62 %
@Code Smell ▾ ⚠ Major ▾ ○ Open ▾ Not assigned ▾ 5min effort Comment
🔗 unused ▾
This block of commented-out lines of code should be removed. Why is this an issue? 12 days ago ▾ L67 %
@Code Smell ▾ ⚠ Major ▾ ○ Open ▾ Not assigned ▾ 5min effort Comment
🔗 unused ▾
@PostMapping(value = "/vets/create")
public String processCreationForm(@Valid final Vet vet, final BindingResult result, @RequestParam(required = false) f
    VetValidator vetValidator = new VetValidator(this.vetService, specialties);
    vetValidator.validate(vet, result);
    if (result.hasErrors()) {
        return VetController.VIEWS_VET_CREATE_FORM;
    } else {
        for (int i : specialties) {
            Specialty s = this.vetService.findSpecialtiesById(i);
            vet.addSpecialty(s);
        }
        // this.authService.saveAuthorities(vet.getUser().getUsername(), "veterinarian");
    }
}

This block of commented-out lines of code should be removed. Why is this an issue? 3 months ago ▾ L96 %
@Code Smell ▾ ⚠ Major ▾ ○ Open ▾ Not assigned ▾ 5min effort Comment
🔗 unused ▾
if (vet.getUser() != null) {
    vet.getUser().setEnabled(true);
}

```

-Después

```
47  */
48 @Controller
49 public class VetController {
50
51     private static final String VIEWS_VET_CREATE_FORM      = "vets/createVet";
52     private static final String VIEWS_VET_LIST             = "vets/visitList";
53     private static final String VIEWS_VET_ADMIN_LIST        = "vets/vetListAdmin";
54
55     private final VetService    vetService;
56
57
58@Autowired
59     public VetController(final VetService clinicService) {
60         this.vetService = clinicService;
61     }
62
63
64
65     @PostMapping(value = "/vets/create")
66     public String processCreationForm(@Valid final Vet vet, final BindingResult result, @RequestParam(required = false)
67         VetValidator vetValidator = new VetValidator(this.vetService, specialties);
68         vetValidator.validate(vet, result);
69         if (result.hasErrors()) {
70             return VetController.VIEWS_VET_CREATE_FORM;
71         } else {
72
73             for (int i : specialties) {
74                 Specialty s = this.vetService.findSpecialtiesById(i);
75                 vet.addSpecialty(s);
76             }
77             if (vet.getUser() != null) {
78                 vet.getUser().setEnabled(true);
79             }
80             this.vetService.saveVet(vet);
81         }
82
83         return "redirect:/vets/admin";
84     }
85 }
```

## OwnerController

-Antes

```

// private static final String VIEWS_OWNER_DETAILS = "owners/ownerDetails";

This block of commented-out lines of code should be removed. Why is this an issue? 3 months ago ▾ L52 %
⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ Not assigned ▾ 5min effort Comment ⚡ unused ▾

149
150
151 56026... /////////////////
152 // @GetMapping(value = "/owners/profile")
// public String processFindProfile(final Owner owner, final BindingResult result, final Map<String, Object> model) {
This block of commented-out lines of code should be removed. Why is this an issue? 3 months ago ▾ L152 %
⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ Not assigned ▾ 5min effort Comment ⚡ unused ▾

153 //     Integer idOwner = owner.getId();
154 //     return "redirect:/owners/profile/" + idOwner;
155 //
156 //

This block of commented-out lines of code should be removed. Why is this an issue? 3 months ago ▾ L156 %
⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ Not assigned ▾ 5min effort Comment ⚡ unused ▾

157 //
158 // @GetMapping("/owners/profile/{ownerId}")
159 // public ModelAndView showProfile(@PathVariable("ownerId") final int ownerId) {
This block of commented-out lines of code should be removed. Why is this an issue? 3 months ago ▾ L159 %
⌚ Code Smell ▾ ⚡ Major ▾ ⚡ Open ▾ Not assigned ▾ 5min effort Comment ⚡ unused ▾

160 //     ModelAndView mav = new ModelAndView("owners/ownerDetails");
161 //     mav.addObject(this.ownerService.findOwnerById(ownerId));
162 //     return mav;
163 //
164 56026...
165 }


```

-Después

```

; @Controller
; public class OwnerController {
;

    private static final String VIEWS_OWNER_CREATE_OR_UPDATE_FORM = "owners/createOrUpdateOwnerForm";
}

| private final OwnerService ownerService;

;

138@    /**
139@     * @GetMapping("/owners/{ownerId}")
140@     * public ModelAndView showOwner(@PathVariable("ownerId") final int ownerId)
141@     *     ModelAndView mav = new ModelAndView("owners/ownerDetails");
142@     *     mav.addObject(this.ownerService.findOwnerById(ownerId));
143@     *     return mav;
144@ }
145}
146


```

## 2.3. Quitar imports no usados

Otro de los bad smells más comunes son los imports en las distintas clases que no son usados en ningún momento por lo que no son necesarios mantenerlos. La solución es simple, sólo habría que quitar los imports de nuestras clases y así ahorraríamos esas líneas de código. Hemos mejorado las siguientes clases y

mostraremos el antes (lo mostrado en SonarCloud) y el después (cómo quedaría nuestro código una vez realizado la solución):

## OwnerController

-Antes

The screenshot shows a SonarCloud interface for a file named 'OwnerController.java'. A specific line of code, 'import org.springframework.validation.FieldError;', is highlighted with a red underline, indicating it is unused. A tooltip box appears over this line with the text: 'Remove this unused import 'org.springframework.validation.FieldError''. Below the code, there are several status indicators: 'Code Smell' (red), 'Minor' (yellow), 'Open' (blue), 'Not assigned' (grey), '2min effort', and 'Comment'. At the top right, the date '2 months ago' and line number 'L33' are visible. A small icon for 'unused' is also present.

-Después

The screenshot shows the same file 'OwnerController.java' after refactoring. The previously unused import statement 'import org.springframework.validation.FieldError;' has been removed. The code now includes imports for javax.validation.Valid, Autowired, Owner, AuthoritiesService, OwnerService, UserService, Controller, Model, BindingResult, WebDataBinder, GetMapping, InitBinder, PathVariable, PostMapping, and ModelAndView. The code also contains two Javadoc comments at the bottom:

```
39 /**
40 * @author Juergen Hoeller
41 * @author Ken Krebs
```

## PetFormatter

-Antes

The screenshot shows a portion of the file 'PetFormatter.java'. Line 74 contains a code snippet with annotations: '@InitBinder("owner")' and 'public void initOwnerBinder(final WebDataBinder dataBinder)'. Line 79 contains another annotation: '@InitBinder("pet")'. The code is part of a larger class definition with other methods and annotations.

-Después

```
1 PetController.java  2 PetTypeFormatter.java
2+ * Copyright 2002-2013 the original author or authors.
3
4 package org.springframework.samples.petclinic.web;
5
6
7 import java.text.ParseException;
8 import java.util.Collection;
9 import java.util.Locale;
10
11
12 import org.springframework.beans.factory.annotation.Autowired;
13 import org.springframework.format.Formatter;
14 import org.springframework.samples.petclinic.model.PetType;
15 import org.springframework.samples.petclinic.service.PetService;
16 import org.springframework.stereotype.Component;
17
18
19 /**
20
21
22
23
24
25
26
27
28
29 */
```

## UserController

-Antes

```
import org.springframework.samples.petclinic.model.Owner;
import org.springframework.samples.petclinic.service.AuthoritiesService;

Remove this unused import 'org.springframework.samples.petclinic.service.AuthoritiesService'. Why is this an issue? 3 months ago L24
Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment
unused ▾

import org.springframework.samples.petclinic.service.OwnerService;
import org.springframework.samples.petclinic.service.VetService;

Remove this unused import 'org.springframework.samples.petclinic.service.VetService'. Why is this an issue? 3 months ago L26
Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment
unused ▾

import org.springframework.samples.petclinic.service.UserService;

Remove this unused import 'org.springframework.samples.petclinic.service.UserService'. Why is this an issue? 3 months ago L27
Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment
unused ▾

import org.springframework.stereotype.Controller;
```

-Después

```

2* * Copyright 2002-2013 the original author or authors.
3 package org.springframework.samples.petclinic.web;
4
5 import java.util.Map;
6
7 import javax.validation.Valid;
8
9 import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.samples.petclinic.model.Owner;
11 import org.springframework.samples.petclinic.service.OwnerService;
12 import org.springframework.stereotype.Controller;
13 import org.springframework.validation.BindingResult;
14 import org.springframework.web.bind.WebDataBinder;
15 import org.springframework.web.bind.annotation.GetMapping;
16 import org.springframework.web.bind.annotation.InitBinder;
17 import org.springframework.web.bind.annotation.PostMapping;
18
19 /**
20  * @author Juergen Hoeller
21  * @author Ken Krebs
22  * @author Arjen Poutsma
23  * @author Michael Isvy
24  */
25 @Controller
26 public class UserController {
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

```

## WelcomeController

-Antes

Petclinic DP2 G7 src/.../samples/petclinic/web/WelcomeController.java

See all issues in this file

```

1  56026... package org.springframework.samples.petclinic.web;
2
3  import java.util.Map;
4
5  import org.springframework.beans.factory.annotation.Autowired;
6
7  Remove this unused import 'org.springframework.beans.factory.annotation.Autowired'. Why is this an issue? 3 months ago ▾ L5 %
8  Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment
8  unused ▾
9
10 import org.springframework.samples.petclinic.repository.UserRepository;
11
12 Remove this unused import 'org.springframework.samples.petclinic.repository.UserRepository'. Why is this an issue? 3 months ago ▾ L6 %
13  Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment
13  unused ▾
14
15 import org.springframework.security.core.userdetails.UserDetails;
16
17 Remove this unused import 'org.springframework.security.core.userdetails.UserDetails'. Why is this an issue? 3 months ago ▾ L7 %
18  Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment
18  unused ▾
19
20 import org.springframework.security.core.userdetails.UserDetailsService;
21
22 Remove this unused import 'org.springframework.security.core.userdetails.UserDetailsService'. Why is this an issue? 3 months ago ▾ L8 %
23  Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment
23  unused ▾
24
25 import org.springframework.samples.petclinic.model.User;
26
27 Remove this unused import 'org.springframework.samples.petclinic.model.User'. Why is this an issue? 3 months ago ▾ L9 %
28  Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment
28  unused ▾
29
30 import org.springframework.stereotype.Controller;
31 import org.springframework.web.bind.annotation.GetMapping;
32 import java.util.List;
33
34 Remove this unused import 'java.util.List'. Why is this an issue? 3 months ago ▾ L12 %
35  Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 2min effort Comment
35  unused ▾
36
37 @Controller
38 public class WelcomeController {

```

-Después

```

package org.springframework.samples.petclinic.web;

import java.util.Map;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
|
@Controller
public class WelcomeController {

    @GetMapping({
        "/", "/welcome"
    })
    public String welcome(final Map<String, Object> model) {

        return "welcome";
    }
}

```

## 2.4. Retoque en la clase DonationController

Al usar SonarCloud nos hemos dado cuenta de que en la clase DonationController, concretamente en el método saveDonation, había en una parte 2 if que podían ser unidos en un solo if si juntásemos ambas condiciones, dado que así haría el mismo funcionamiento y nos ahorrariamos mirar las 2 condiciones por separado teniendo peor rendimiento de esta manera. El código inicialmente era de esta manera:

The screenshot shows a code editor with SonarCloud's analysis overlaid. The code is as follows:

```

142     1 if (donation.getMoney() != null) {
143         ...
144             if (donation.getMoney() > moneyRest) {
145                 ...
146                     FieldError err = new FieldError(DONATION, "money", "Your donation must not exceed the remaining mon
147                     result.addError(err);
148             }
149     }

```

A tooltip from SonarCloud suggests merging the two inner if statements:

Merge this if statement with the enclosing one. Why is this an issue? 3 months ago L144 %

Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment by clumsy ▾

Al refactorizar el código y hacer lo que hemos comentado anteriormente, quedaría de la siguiente forma:

```

129
130     if (donation.getMoney() != null && donation.getMoney() > moneyRest) {
131         FieldError err = new FieldError(DonationController.DONATION, "mone
132         result.addError(err);
133     }
134

```

## 2.5. Eliminación de System.out innecesario

Durante la implementación del sistema, es habitual hacer uso de System.out para comprobar el valor de ciertos parámetros en la ejecución del método que estamos creando. Por ello, es normal que en varias ocasiones se queden estos System.out en nuestro código y no tienen una utilidad más allá de la ayuda anteriormente nombrada, por lo que hay que eliminarlos por su nula importancia. Esto nos ha ocurrido en la clase DonationController y este es el antes:

```
model.put(DONATION, donation);

System.out.println(donation.getUser().getUsername());
```

Replace this use of System.out or System.err by a logger. Why is this an issue?

3 months ago ▾ L107 🔍

Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 10min effort Comment

bad-practice, cert ▾

Después de eliminar el System.out innecesario:

```
81     if (!ds.isEmpty()) {
82
83         moneyD = ds.stream().mapToDouble(x -> x.getMoney()).sum();
84
85     }
86     Double moneyRest = moneyF - moneyD;
87     donation.setMoneyRest(moneyRest);
88
89     donation.setUser(u);
90     List<String> x = Arrays.asList("true", "false");
91     model.addAttribute("anonymous", x);
92     String nombre = causes.getTitle();
93     donation.setCauses(causes);
94     donation.getCauses().setTitle(nombre);
95
96     model.put(DonationController.DONATION, donation);
97
98     return view;
99
100 }
```

## 3.Bugs:

Con SonarCloud nos hemos dado cuenta además de que tenemos varios bugs dentro de nuestro sistema. Debido al poco tiempo de margen, va a ser complicado arreglar todos ellos pero intentaremos al menos que sean los menos posibles. Por mi parte, estos han sido los bugs que he corregido para mejorar nuestro código:

### 3.1. Bug en RoomController

Gracias a SonarCloud, nos hemos dado cuenta de que un método de esta clase podría presentar un error de NullPointerException. El error era este:

```

156 ...
157     ...
158     @GetMapping(value = "/rooms/delete/{roomId}")
159     ...
160     public String processDeleteRoom(@PathVariable("roomId") final int roomId, final Model model) {
161         ...
162         String view = REDIRECT_ROOM;
163         Room room = this.roomService.findRoomById(roomId);
164         if (room != null && !room.getReservations().isEmpty()) {
165             for (Reservation r : room.getReservations()) {
166                 if (!r.getStatus().getName().equals("ACCEPTED")) {
167                     this.roomService.delete(room);
168                     model.addAttribute(MESSAGE, "Event Successfully deleted");
169                 }
170             }
171         } else if (room != null && room.getReservations().isEmpty()) {
172             this.roomService.delete(room);
173         } else {
174             model.addAttribute("roomNotDeleted", "The room named " + room.getName() + " cannot be removed because it has reservations");
175         }
176     }

```

A "NullPointerException" could be thrown; "room" is nullable here. Why is this an issue?  
last month ▾ L172 🔍

Bug Major Open Not assigned 10min effort Comment  
cert, cwe

Dn 2-ENTREGA VIERNES

Cómo podemos ver, en el caso de que room sea igual a null, entraría en el else. Cómo en el else utilizamos room.getName() y la condición para que entre dentro de este punto es que room sea null, ocasionaría el error que hemos comentado anteriormente. La solución más rápida y eficaz que he encontrado ha sido eliminar este room.getName(), que tampoco tiene una funcionalidad importante, evitando así la aparición del error. El código quedaría de la siguiente manera:

```

@GetMapping(value = "/rooms/delete/{roomId}")
public String processDeleteRoom(@PathVariable("roomId") final int roomId, final Model model) {
    String view = RoomController.REDIRECT_ROOM;
    Room room = this.roomService.findRoomById(roomId);
    if (room != null) {
        if (!room.getReservations().isEmpty()) {
            for (Reservation r : room.getReservations()) {
                if (!r.getStatus().getName().equals("ACCEPTED")) {
                    this.roomService.delete(room);
                    model.addAttribute(RoomController.MESSAGE, "Event Successfully deleted");
                }
            }
        } else {
            this.roomService.delete(room);
        }
    } else {
        model.addAttribute("roomNotDeleted", "The room cannot be removed because it has reservations");
    }
    return view;
}

```

### 3.2. Bug en los Validadores

Hago este apartado más genérico que el anterior dado que este bug estaba tanto en VisitValidator, VetValidator y DiagnosisValidator. El bug ha sido simple pero que había que arreglar dado que no estaba hecho correctamente. El fallo ha sido el siguiente (le expongo una foto de sólo uno de los casos dado que es el mismo):

```

@Override
public void validate(final Object obj, final Errors errors) {
    Diagnosis diagnosis = (Diagnosis) obj;
    String description = diagnosis.getDescription();
    LocalDate date = diagnosis.getDate();
    LocalDate now = LocalDate.now();

    if (description == null || description == "") {

        Use the "equals" method if value comparison was intended. Why is this an issue?
        last month ▾ L30 0%
        Bug ▾ Major ▾ Open ▾ Alvaro Frias ▾ 5min effort Comment cert, cwe ▾
    }

    errors.rejectValue("description", "The description is empty", "The description is empty");
}

if (date != null) {
    Boolean posterior = date.isBefore(now);
    if (posterior) {
        errors.rejectValue("date", "The date is before the current date", "The date is before the current date");
    }
}

```

Cómo podemos ver, en la parte de descripción hemos usado el comparador `==` en vez de `equals` que sería realmente el comparador correcto para la comparación de Strings. Por ello, realizamos estos cambios en los validadores antes comentados y quedarían de esta forma:

## VisitValidator

```

@Override
public void validate(final Object obj, final Errors errors) {
    Visit visit = (Visit) obj;
    String description = visit.getDescription();
    LocalDate date = visit.getDate();
    LocalDate now = LocalDate.now();

    if (description == null || description.equals("")) {
        errors.rejectValue("description", "The description is empty", "The description is empty");
    }

    if (date != null) {
        if (date.isBefore(now)) {
            errors.rejectValue("date", "The date is before the current date", "The date is before the current date");
        }
    } else {
        errors.rejectValue("date", "The date is empty", "The date is empty");
    }
}

```

## VetValidator

```

@Override
public void validate(final Object obj, final Errors errors) {
    Vet vet = (Vet) obj;
    int[] specialties = this.specialties;
    String username = vet.getUser().getUsername();
    String password = vet.getUser().getPassword();

    if (specialties == null) {
        errors.rejectValue("specialties", "The specialties is empty", "The specialties is empty");
    }

    if (username == null || username.equals("")) {
        errors.rejectValue("user.username", "The username is empty", "The username is empty");
    }

    if (password == null || password.equals("")) {
        errors.rejectValue("user.password", "The password is empty", "The password is empty");
    }
}

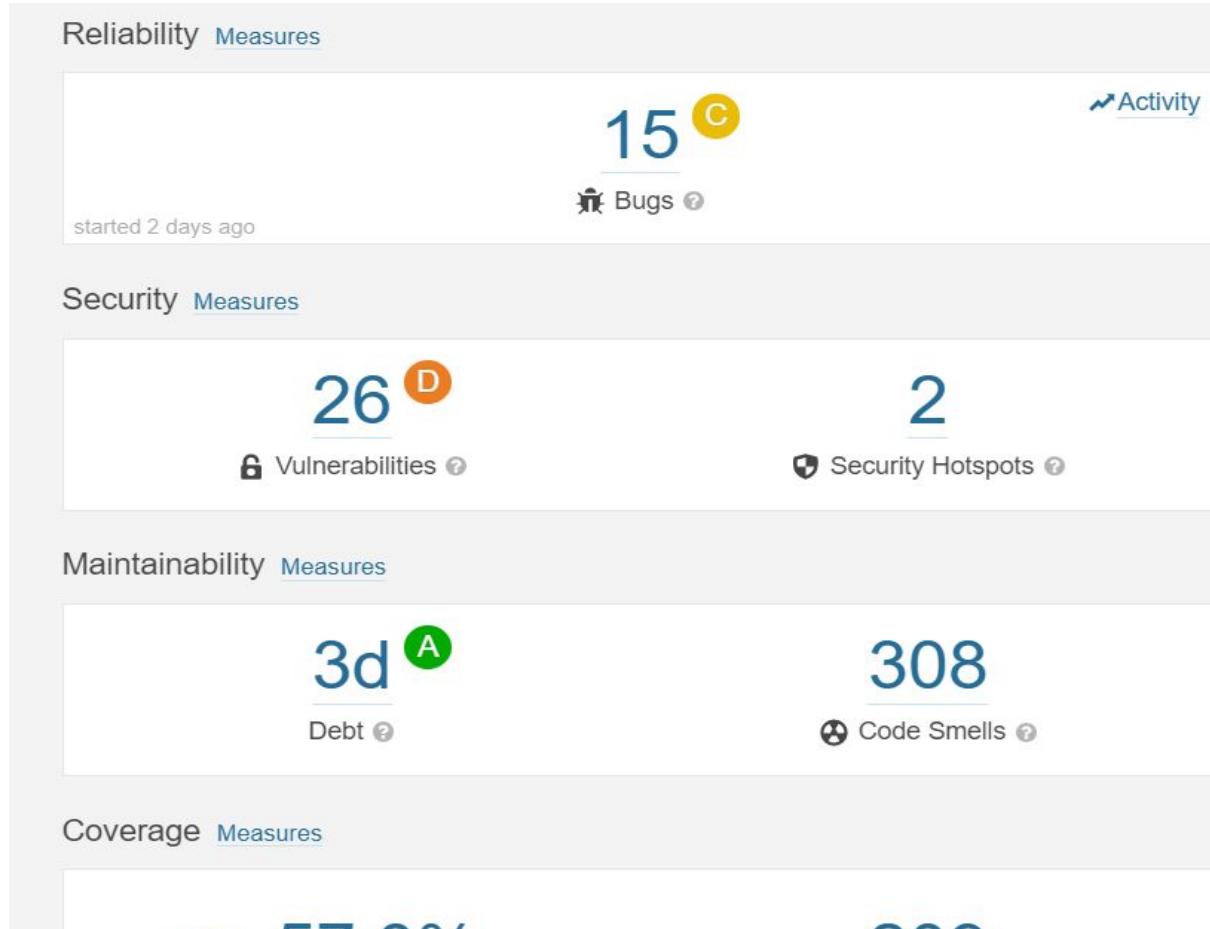
```

## DiagnosisValidator

```
@Override  
public void validate(final Object obj, final Errors errors) {  
    Diagnosis diagnosis = (Diagnosis) obj;  
    String description = diagnosis.getDescription();  
    LocalDate date = diagnosis.getDate();  
    LocalDate now = LocalDate.now();  
  
    if (description == null || description.equals("")) {  
        errors.rejectValue("description", "The description is empty", "The description is empty");  
    }  
  
    if (date != null) {  
        if (date.isBefore(now)) {  
            errors.rejectValue("date", "The date is before the current date", "The date is before the  
            current date");  
        }  
    }  
}
```

## 4. Resultados

Una vez realizada la refactorización en ciertas partes, analizamos de nuevo nuestro sistema y obtenemos el siguiente resultado:



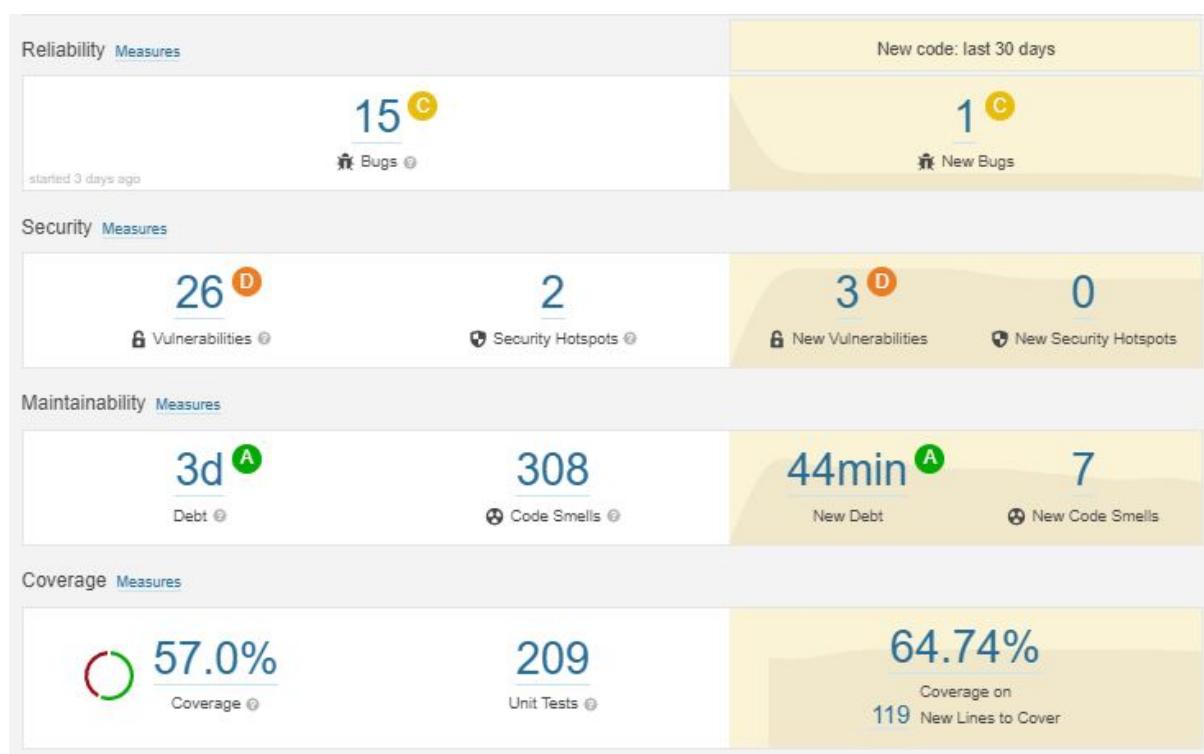
Cómo podemos ver, hemos reducido varios Code Smells y bugs pero sigue sin ser lo óptimo. Nos habría gustado tener más tiempo para poder refactorizar mucho mejor nuestro sistema y quede mucho más perfecto pero dado que el tiempo a estas

alturas ya escasea pues en este momento es lo más óptimo que hemos conseguido refactorizar.

## Refactoring 4 Adrián:

### 1. Introducción:

Vamos a ver en esta parte el conjunto de bugs, vulnerabilidades y bad smells que podemos encontrar dentro de nuestra aplicación para posteriormente realizar una refactorización solucionandolos. Los resultados del análisis de sonar cloud son:



Con estos resultados vamos a proceder a corregir algunos bad smells y vulnerabilidades.

### 2. Code smells

Vamos a ver varios bad smells, viendo de qué se tratan y su solución:

- VetValidator: Existe una variable `specialities` que está dos veces nombrada, esto puede ser confuso:

```

@Override
public void validate(final Object obj, final Errors errors) {
    Vet vet = (Vet) obj;
    int[] specialties = this.specialties;

    Rename "specialties" which hides the field declared at line 16. Why is this an issue?
    13 days ago ▾ L31 %
    ⚡ Code Smell ▾ 🔴 Major ▾ ⚡ Open ▾ 🌐 achaghirc ▾ 5min effort Comment
    🔗 cert, pitfall, suspicious ▾

```

Solución: renombrar la variable con otro nombre

```

@Override
public void validate(final Object obj, final Errors errors) {
    Vet vet = (Vet) obj;
    int[] specialties = this.specialties;
    String username = vet.getUser().getUsername();
    String password = vet.getUser().getPassword();

    if (specialties == null) {
        errors.rejectValue("specialties", "The specialties is empty", "Specialties cannot be null");
    }
}

```

- DonationServiceTest: Orden del asset incorrecto

```

assertEquals(1, count);

Swap these 2 arguments so they are in the correct order: expected value, actual value. Why is this an issue?
2 months ago ▾ L27 %
    ⚡ Code Smell ▾ 🔴 Major ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment
    🔗 junit, suspicious, tests ▾

```

Solución: cambiar el orden del assert.

```

public void testMyDonations() {
    int count = donationService.findMyDonations("admin1").size();

    assertEquals(3, count);
    assertNotEquals(0, count);
}

```

- Todos los test: Para Junit5 añadir public a una clase Test es irrelevante, sin embargo eliminarlo mejoraría el código.

```

@WebMvcTest/controllers=ReservationController.class,
excludeFilters = @ComponentScan.Filter(type = FilterType.ASSIGNABLE_TYPE, classes = WebSecurityConfigurer.class),
excludeAutoConfiguration= SecurityConfiguration.class)
public class ReservationControllerTest {

Remove this 'public' modifier. Why is this an issue?
2 months ago ▾ L43 %
    ⚡ Code Smell ▾ ⓘ Info ▾ ⚡ Open ▾ Not assigned ▾ 2min effort Comment
    🔗 junit, tests ▾

```

Solución: Eliminar public de la clase.

```

@DataJpaTest(includeFilters = @ComponentScan.Filter(Service.class))
@AutoConfigureTestDatabase(replace = Replace.NONE)
class DonationServiceTests {

    @Autowired
    private DonationService donationService;
}

```

## 3. Bugs

Ahora procederemos a corregir los bugs en nuestro sistema:

- AuthoritiesService: La variable Optional puede no estar presente:

```
public Authorities findAuthorityByName(String username) throws DataAccessException {
    return authoritiesRepository.findById(username).get();
```

Solución: Comprobar antes si la variable está presente.

```
public Authorities findAuthorityByName(String username) throws DataAccessException {
    Authorities res = null;
    if(authoritiesRepository.findById(username).isPresent()) {
        res = authoritiesRepository.findById(username).get();
    }
    return res;
}
```

- PetService: No se está usando el equals:

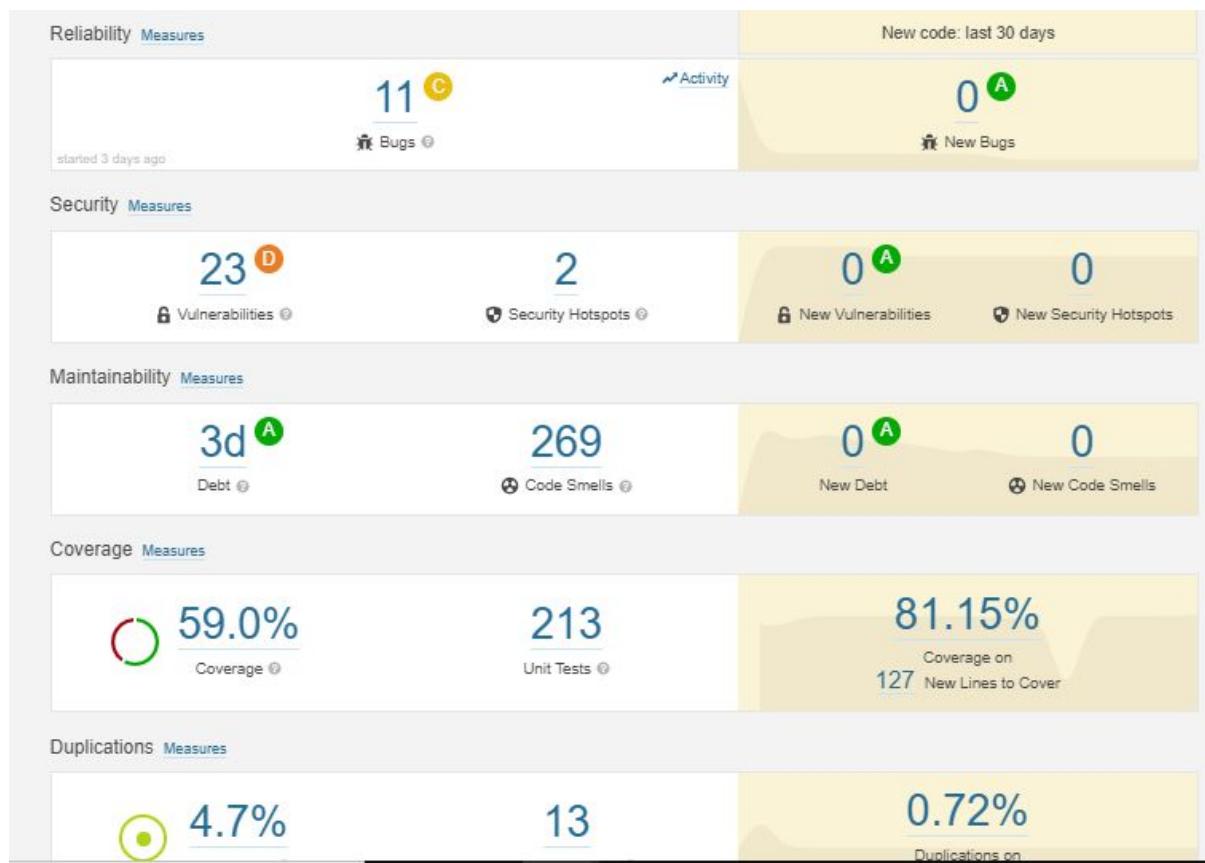
```
PET OTHERPET = pet.getOwner().getPetWithIdDifferent(pet.getId(), pet.getName(), pet.getIu());
if (StringUtils.hasLength(pet.getName()) && otherPet != null && otherPet.getId() != pet.getId()) {
```

Solución: Cambiar la comparación por un equals.

```
if (StringUtils.hasLength(pet.getName()) && otherPet != null && !otherPet.getId().equals(pet.getId())) {
    throw new DuplicatedPetNameException();
} else {
    this.petRepository.save(pet);
}
```

## 4. Resultados

Después de solucionar todos estos bugs y bad smells vamos a ver los resultados, cabe destacar que aunque solo hayamos nombrado los problemas anteriores una vez estos se repetían varias veces en toda la aplicación pero para no repetir varias veces dicho problema solo se ha puesto una vez.



Como podemos observar hemos reducido los bugs de 15 a 11, los code smells de 308 a 269 y las vulnerabilidades de 26 a 23. Además de esto se han añadido nuevos test para las líneas de código añadidas ya que anteriores análisis la cobertura de estas líneas era muy inferior a 80%.