



ANÁLISIS DEL CÓDIGO FUENTE Y MÉTRICAS ASOCIADAS

PSG2 G6-62

Índice de contenidos

1. SPRINT L2 – CONFIGURATION MANAGEMENT	2
1.1. SONAR CLOUD: PANEL DE ESTADÍSTICAS, DESCRIPCIÓN Y VALORES	2
1.2. DESCRIPCIÓN Y ANÁLISIS DE LOS BUGS POTENCIALES	4
1.3. DESCRIPCIÓN Y ANÁLISIS DE LOS DIFERENTES “TIPOS DE MALOS OLORES” EN EL CÓDIGO .	5
1.4. CONCLUSIÓN SOBRE LOS RESULTADOS DEL ANÁLISIS.....	11
2. SPRINT L3 – MEASUREMENT	12
2.1. SONARCLOUD: PANEL DE ANÁLISIS Y DESCRIPCIÓN DE MÉTRICAS	12
2.2. DESCRIPCIÓN Y ANÁLISIS DE LOS BUGS POTENCIALES	15
2.3. DESCRIPCIÓN Y ANÁLISIS DE LOS DIFERENTES “TIPOS MALOS DE OLORES” EN EL CÓDIGO	15
2.4. CONCLUSIÓN SOBRE LOS RESULTADOS DEL ANÁLISIS.....	16
2.5. COMPARATIVA CON SPRINT L2.....	15
3. E3.2 – EXTENSIÓN	12
3.1. INTRODUCCIÓN.....	12
3.1. VULNERABILIDADES	12
3.1. CODE SMELLS	12
3.1. CONCLUSIÓN	24

1. SPRINT L2 - CONFIGURATION MANAGEMENT

1.1. SONAR CLOUD: PANEL DE ESTADÍSTICAS, DESCRIPCIÓN Y VALORES

En este documento vamos a representar un análisis global de la calidad del código tanto del Sprint L2 como del L3.

Para comenzar, vamos a mostrar un panel de estadísticas del L2 dividido en dos partes:

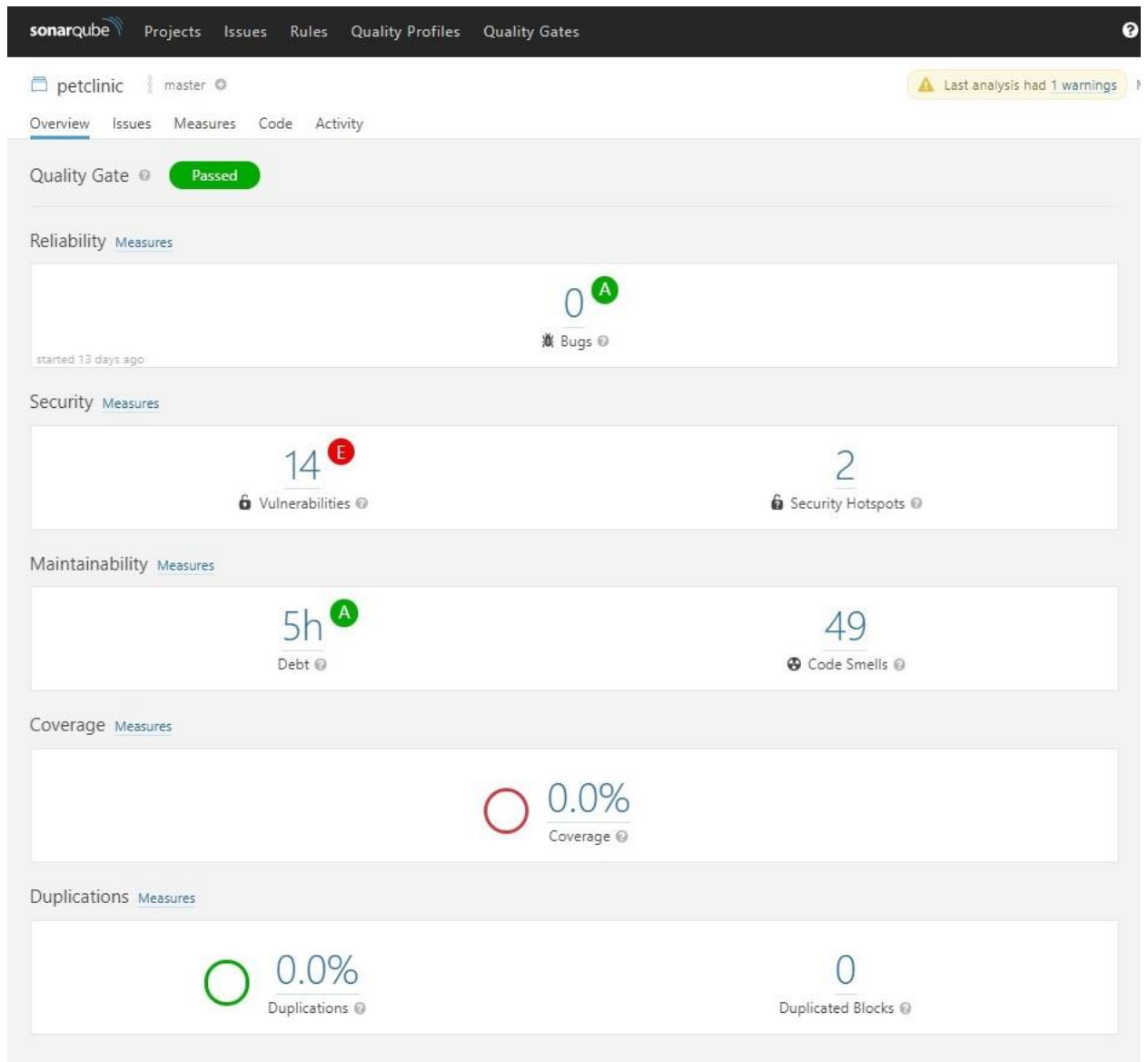


Imagen 1: Panel de estadísticas 1

En esta 1ª parte, podemos observar que no hay bugs (A) ni duplicaciones de código (A). En cuanto a la calidad del análisis, es pasable, a pesar de que tenemos 16 vulnerabilidades (E) y 49 “malos olores” de código.

En la 2º parte podemos apreciar la cantidad y el tipo de código usado para este Sprint:



Imagen 2: Panel de estadísticas 2

Un total de, aproximadamente, 1.5k líneas de código. 222 son de XML y unas 1300 son de Java. A continuación, se muestra una gráfica de posibles riesgos en la cobertura abierta del proyecto:

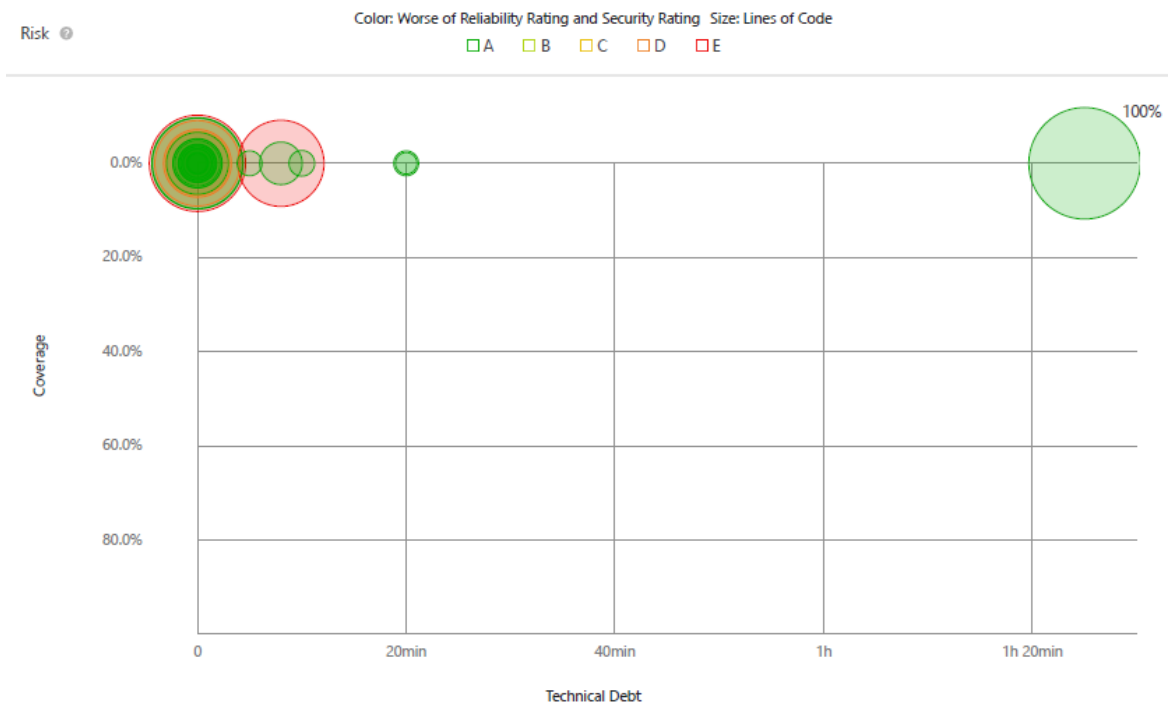


Imagen 3: Cobertura abierta

Los círculos rojos hacen referencia a las vulnerabilidades de seguridad. Representan un 0%.

1.2. DESCRIPCIÓN Y ANÁLISIS DE LOS BUGS POTENCIALES

Para el L2, no tenemos ningún tipo de bugs, lo cual es algo bastante positivo.

Si tenemos una serie de vulnerabilidades:

src/.../samples/petclinic/web/OwnerController.java	
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L68 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L85 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L119 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
Add a "method" parameter to this "@RequestMapping" annotation. See Rule	19 days ago • L131 🔍 ⚡
🔒 Vulnerability 🚫 Blocker 🔓 Open Not assigned 5min effort	🔍 cwe, owasp-a5, lars-top25-insecure, sp...
src/.../samples/petclinic/web/PetController.java	
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L73 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L81 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L81 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L104 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L104 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
Add a "method" parameter to this "@RequestMapping" annotation. See Rule	19 days ago • L116 🔍 ⚡
🔒 Vulnerability 🚫 Blocker 🔓 Open Not assigned 5min effort	🔍 cwe, owasp-a5, lars-top25-insecure, sp...
src/.../samples/petclinic/web/ResidenceController.java	
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L55 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
src/.../samples/petclinic/web/VetController.java	
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L96 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L113 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring
src/.../samples/petclinic/web/VisitController.java	
Replace this persistent entity with a simple POJO or DTO object. See Rule	19 days ago • L80 🔍 ⚡
🔒 Vulnerability 🚫 Critical 🔓 Open Not assigned 10min effort	🔍 cwe, owasp-a5, spring

Imagen 4: Vulnerabilidades

Podemos clasificarlas en dos: 12 críticas y 2 bloqueantes.

Las críticas hacen referencia al posible riesgo de la posibilidad de inyectar código en la web. En cambio, los bloqueantes a faltas de anotaciones en los controladores.

El tiempo de esfuerzo en solucionar estas vulnerabilidades serían, aproximadas, 2h 10'.

1.3. DESCRIPCIÓN Y ANÁLISIS DE LOS DIFERENTES "TIPOS DE MALOS OLORES" EN EL CÓDIGO

Para este Sprint tenemos 49 "malos olores", clasificados en menores, mayores y críticos, según su orden de importancia. A continuación, haremos un repaso sobre ellos:

Menores:

src/.../samples/petclinic/repository/OwnerRepository.java	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L44	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L52	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L59	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L61	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
src/.../samples/petclinic/repository/PetRepository.java	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L42	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L50	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L57	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L59	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
src/.../samples/petclinic/repository/ResidenceRepository.java	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L10	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L14	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule 20 days ago L16	Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...

Imagen 5: Menores 1

src/.../samples/petclinic/repository/VetRepository.java	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L42
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L44
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L46
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L50
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L52
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L55
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
src/.../samples/petclinic/repository/VisitRepository.java	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L43
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L45
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
Rename this local variable to match the regular expression '^ [a-z][a-zA-Z0-9]*\$'. See Rule	20 days ago ▾ L47
Code Smell Minor Open Not assigned 2min effort convention	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L47
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L49
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	
src/.../repository/springdatajpa/SpringDataPetRepository.java	
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. See Rule	20 days ago ▾ L37
Code Smell Minor Open Not assigned 5min effort clumsy, error-handling, redundant, unu...	

Imagen 6: Menores 2

src/_/repository/springdatajpa/SpringDataVetRepository.java

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L40 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

src/_/samples/petclinic/service/ClinicService.java

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L74 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L79 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L84 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L89 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L94 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L99 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L113 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L118 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L123 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception. [See Rule](#) 20 days ago ▾ L129 🔗 ⚙️
🔧 Code Smell 🟡 Minor 🔵 Open Not assigned 5min effort 🗑️ clumsy, error-handling, redundant, unu...

Imagen 7: Menores 3

See Rule	Code Smell Minor Open Not assigned 5min effort	clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.	20 days ago ▾ L134	
See Rule	Code Smell Minor Open Not assigned 5min effort	clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.	20 days ago ▾ L139	
See Rule	Code Smell Minor Open Not assigned 5min effort	clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.	20 days ago ▾ L144	
See Rule	Code Smell Minor Open Not assigned 5min effort	clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.	20 days ago ▾ L149	
See Rule	Code Smell Minor Open Not assigned 5min effort	clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.	20 days ago ▾ L158	
See Rule	Code Smell Minor Open Not assigned 5min effort	clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.	20 days ago ▾ L167	
See Rule	Code Smell Minor Open Not assigned 5min effort	clumsy, error-handling, redundant, unu...
Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.	20 days ago ▾ L172	
See Rule	Code Smell Minor Open Not assigned 5min effort	clumsy, error-handling, redundant, unu...
src/.../samples/petclinic/util/EntityUtils.java		
Convert the abstract class "EntityUtils" into an interface. See Rule	20 days ago ▾ L33	java8
Code Smell Minor Open Not assigned 10min effort		
Remove the declaration of thrown exception 'org.springframework.orm.ObjectRetrievalFailureException' which is a runtime exception.	20 days ago ▾ L44	
See Rule	Code Smell Minor Open Not assigned 5min effort	clumsy, error-handling, redundant, unu...

Imagen 8: Menores 4

Mayores:

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

This block of commented-out lines of code should be removed. [See Rule](#)

20 days ago ▾ L17 🔗 ⚙️ ▾

🔗 Code Smell 🚨 Major 🔵 Open Not assigned 5min effort

🔗 based-on-misra, unused

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

Rename field "vets" [See Rule](#)

20 days ago ▾ L33 🔗 ⚙️ ▾

🔗 Code Smell 🚨 Major 🔵 Open Not assigned 10min effort

🔗 brain-overload

src/.../samples/petclinic/util/EntityUtils.java

Add a private constructor to hide the implicit public one. [See Rule](#)

20 days ago ▾ L33 🔗 ⚙️ ▾

🔗 Code Smell 🚨 Major 🔵 Open Not assigned 5min effort

🔗 design

src/.../samples/petclinic/web/CrashController.java

Define and throw a dedicated exception instead of using a generic one. [See Rule](#)

20 days ago ▾ L34 🔗 ⚙️ ▾

🔗 Code Smell 🚨 Major 🔵 Open Not assigned 20min effort

🔗 cert, cwe, error-handling

Imagen 9: Menores 4

Críticos:

src/.../samples/petclinic/web/PetController.java

Define a constant instead of duplicating this literal "redirect:/owners/{ownerId}" 3 times. [See Rule](#)

20 days ago ▾ L92 🔗 ⚙️ ▾

🔗 Code Smell 🚨 Critical 🔵 Open Not assigned 8min effort

🔗 design

src/.../samples/petclinic/web/ResidenceValidator.java

Define a constant instead of duplicating this literal "releaseDate" 3 times. [See Rule](#)

20 days ago ▾ L27 🔗 ⚙️ ▾

🔗 Code Smell 🚨 Critical 🔵 Open Not assigned 8min effort

🔗 design

Imagen 10: Críticos

1.4. CONCLUSIÓN SOBRE LOS RESULTADOS DEL ANÁLISIS

En conclusión final de este sprint: se puede mejorar en el concepto de las vulnerabilidades, aunque tampoco son muchas. En general, estamos muy contentos, porque aparte de que el análisis global ha sido pasable, hemos acabado sin bugs y sin duplicaciones de código, lo cual son resultados satisfactorios. Esperamos una mejora para el Sprint L3. En el caso de que no sea así, nos propondremos hacer el documento extra para solucionar los errores que se reflejan en el código.

2. Sprint L3 – Measurement

2.1. SONARCLOUD: PANEL DE ANÁLISIS Y DESCRIPCIÓN DE MÉTRICAS

En la siguiente imagen mostramos el panel de control obtenido en el análisis realizado por 'SonarCloud' tras la realización de las implementaciones especificadas en el 'Product Backlog' para este 'Sprint':

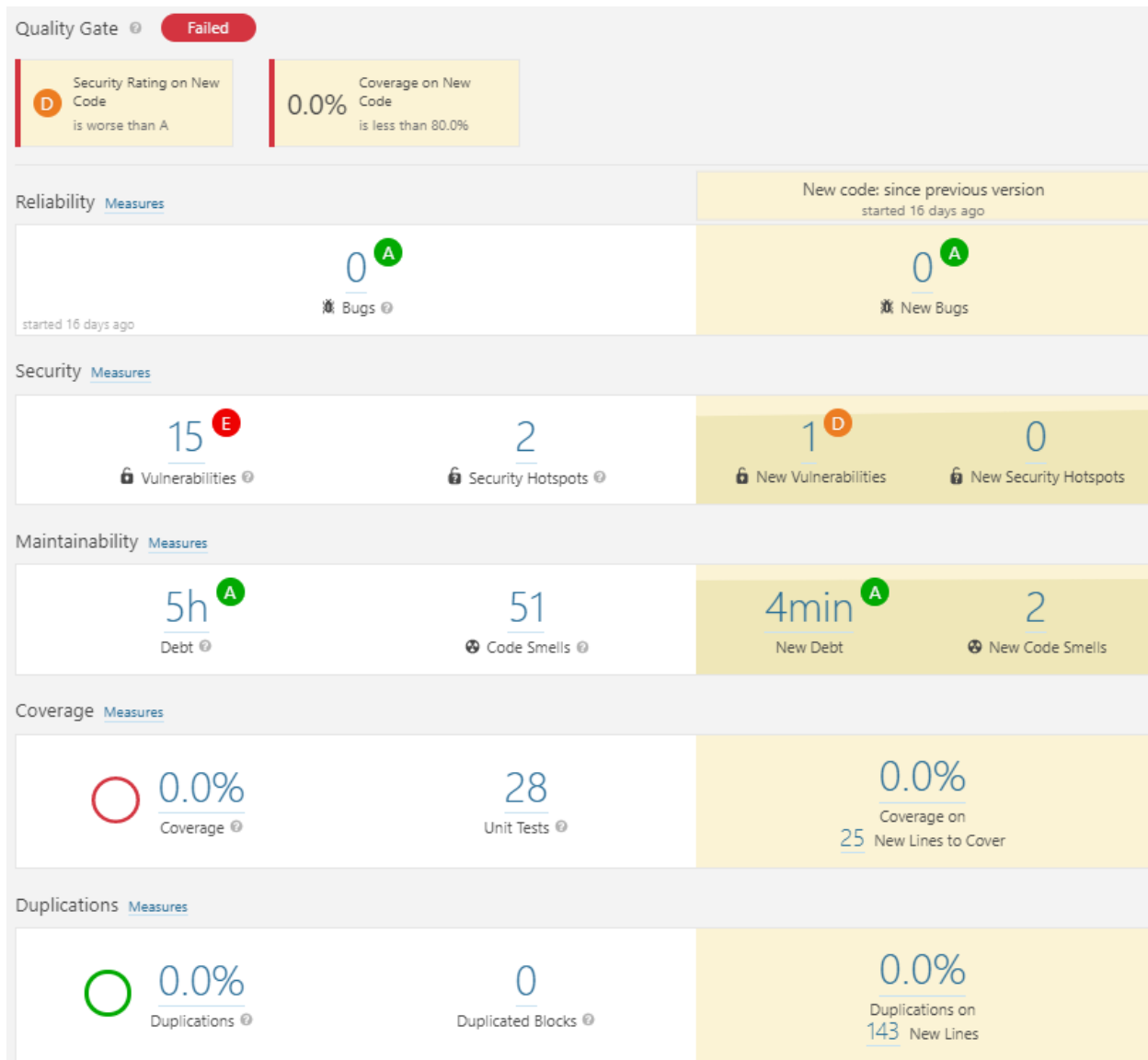


Imagen 1: Panel de análisis de SonarCloud

Observamos que la calidad del análisis es 'failed', al contrario que el anterior, en el que se obtuvo 'passed'. Esto se debe a la ampliación de las funcionalidades en el código y las nuevas vulnerabilidades y 'code smells'. Otra de las razones por la que se encuentra en dicho estado es la ausencia de nuevas pruebas del código. Todo esto se planea corregir en el caso de hacer la actividad extra relacionada con este tema. En resumen, nos encontramos con 0 'bugs', 15

vulnerabilidades, 51 *'code smells'* y un 0% del código cubierto con pruebas. Esto se mejorará en la actividad extra de este sprint si se hiciese.

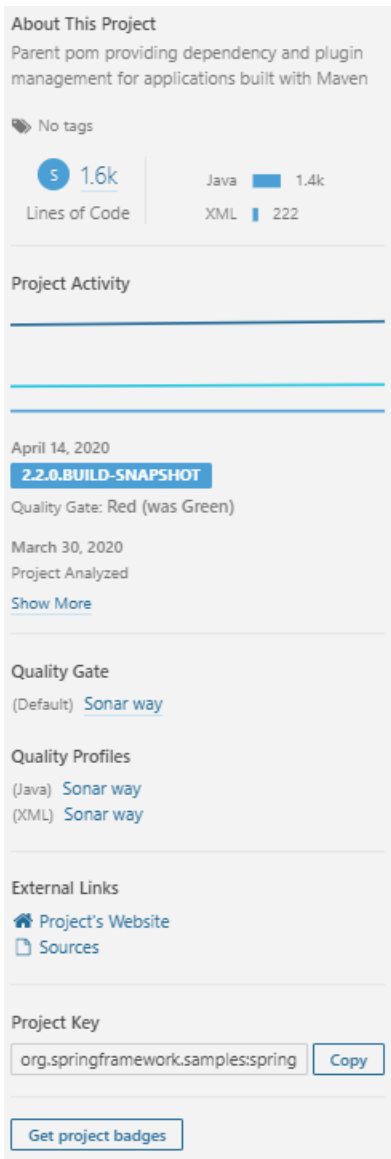


Imagen 2: Líneas de código

Respecto a las líneas de código, contamos con alrededor de 1600, de las cuales más o menos 1400 son de Java y unas 222 son de xml.

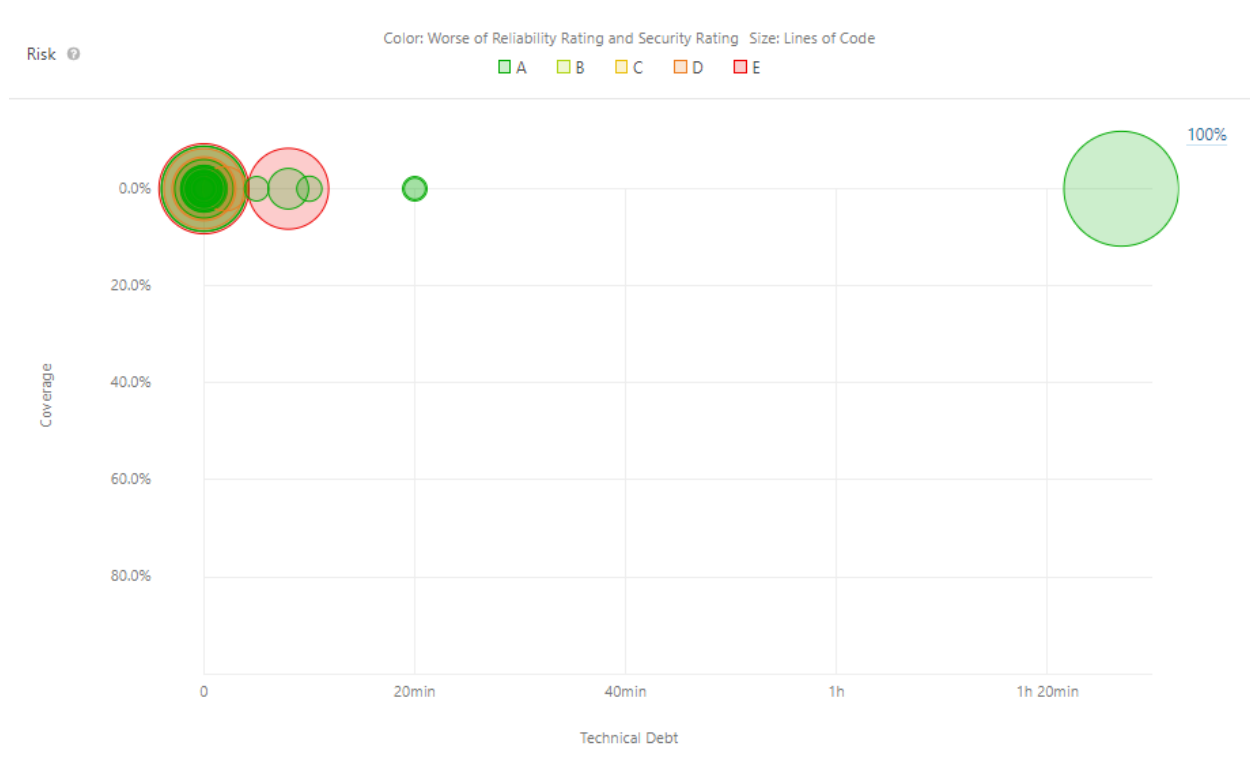


Imagen 3: Covertura vs Deuda técnica

En la tabla observamos como no se ha cubierto ninguna parte del proyecto con pruebas. Además, encontramos círculos rojos y naranjas, que indican mala calificación en la seguridad, es decir, vulnerabilidades.

2.2. DESCRIPCIÓN Y ANÁLISIS DE LOS BUGS POTENCIALES

En el proyecto no se han encontrado bugs, por lo cual la tabla dada por SonarCloud queda así:

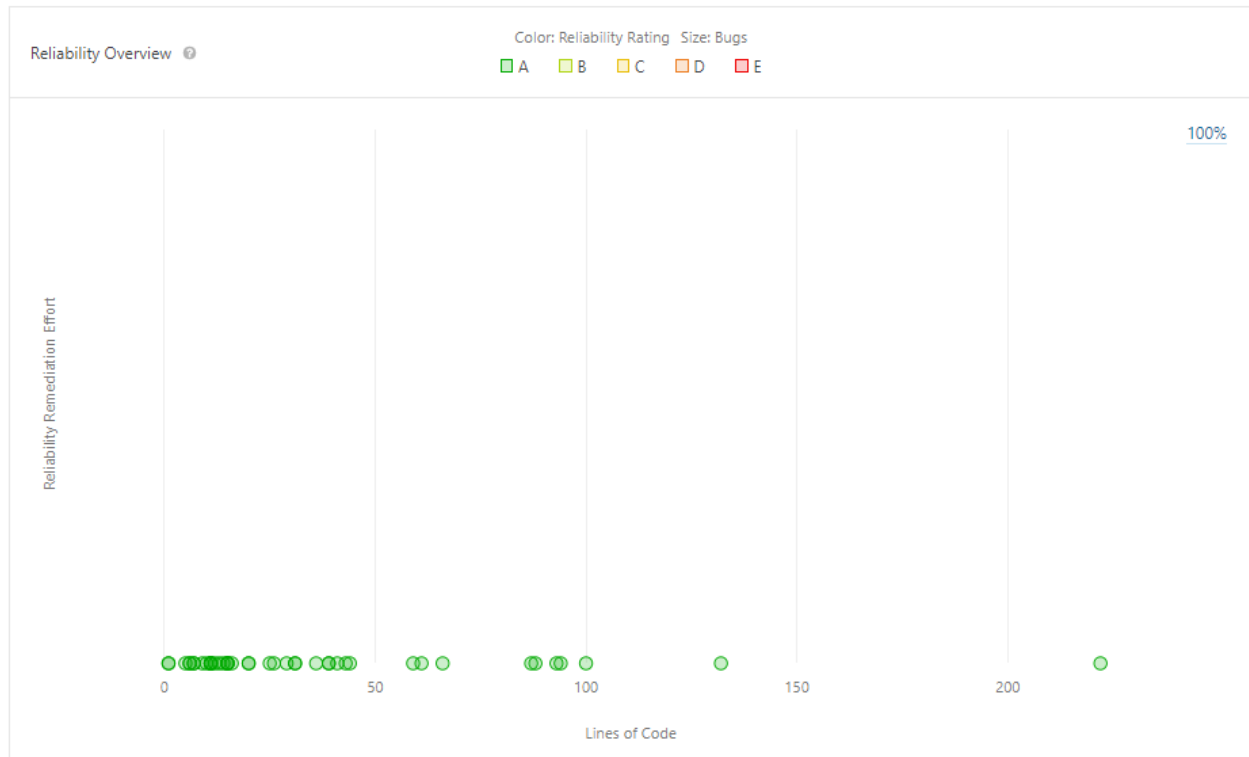


Imagen 4: Esfuerzo de remedio vs Líneas de código

Como se puede observar todas las burbujas aparecidas son de color verde, indicando nivel A en 'Reliability Rating', y pequeñas, indicando que no hay bugs.

2.3. DESCRIPCIÓN Y ANÁLISIS DE LOS DIFERENTES "TIPOS MALOS DE OLORES" EN EL CÓDIGO

Respecto a los malos olores encontramos 51 en todo el proyecto, es decir, se han encontrado dos nuevos en el código subido para el 'sprint'3. Ambos malos olores son de gravedad menor, puesto que, el primero es referido a que el nombre de una clase java no cumple con la convención, y el segundo es debido a una importación que no está siendo utilizada, es decir, ambos poco importantes:

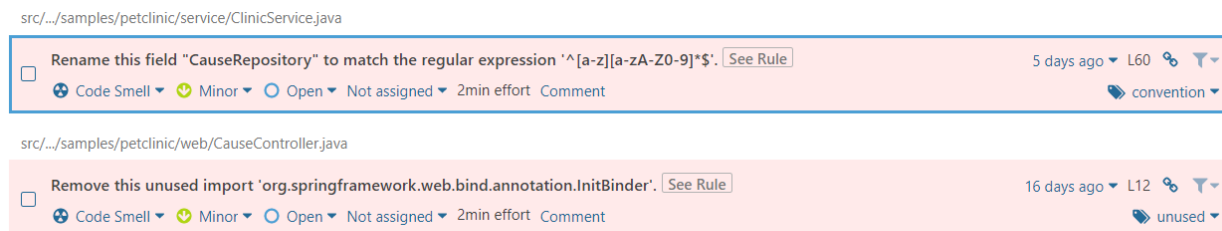


Imagen 5: Malos olores

Ambos pueden ser corregidos fácilmente con el cambio del nombre, en el primer caso; y la eliminación del 'import' en el segundo.

2.4. CONCLUSIÓN SOBRE LOS RESULTADOS DEL ANÁLISIS

No estamos contentos con los resultados obtenidos en el análisis, por lo que en el tiempo restante del 'sprint' se va a realizar su corrección, para intentar conseguir eliminar todos los malos olores y vulnerabilidades. También se intentará cubrir el código con pruebas para intentar que el resultado del análisis sea favorable. Como puntos favorables, podemos observar que no se ha encontrado ningún bug en el proyecto, lo que nos da buena sintonía sobre nuestra forma de desarrollar código, al igual que la complejidad ciclomática, siendo esta de 213, aunque nos esforzaremos e intentaremos bajar ese valor si es posible. Con todo ello, en resumen, pensamos que a la hora de desarrollo de código cumplimos con lo requerido, aunque debemos tener más cuidado para intentar reducir las vulnerabilidades y los malos olores al mínimo.

2.5. COMPARATIVA CON SPRINT L2

Vamos a hacer un contraste con el sprint anterior para analizar la causa de los resultados y así tener una idea más clara de los posibles cambios:

- En primer lugar, en ambos sprints tenemos 0 bugs y 0 bloques duplicados (*"imagen1 (L1yL2)"*) lo cual asociamos a una buena planificación antes del desarrollo del código y a una correcta implementación de este.

- Por otro lado, en L1 tuvimos 14 vulnerabilidades y 49 "malos olores" mientras que en L2 solo obtuvimos 1 nueva vulnerabilidad y 2 "malos olores" (*"imagen1 (L1yL2)"*). Esta gran desproporción se debe a que en L1 tuvimos que crear el proyecto e implementar ciertas funcionalidades lo cual supuso 1500 líneas de código (*"imagen 2(L1)"*). Mientras que en L2 solo se le añadieron alrededor de 100 líneas de código, sumando así aproximadamente 1600 (*"imagen 2 (L2)"*). Por lo tanto pensamos que la comparativa será mucho mas útil entre el sprint actual y el próximo.

- Por último, hay que comentar la diferencia entre la etiqueta "Passed" (*"Imagen1 (L1)"*) y "Failed" (*"Imagen1 (L2)"*) en el apartado de calidad. Este cambio se debe a la aparición de un riesgo de seguridad en las vulnerabilidades y a la no implementación de pruebas en L2.

3. E3.2 Extensión del report

3.1. INTRODUCCIÓN

En este apartado, se pretende ahondar más en el uso de sonarqube, solucionando posibles errores o fallos cometidos en el sprint. Para ello esta herramienta, analiza estos errores y los clasifica en vulnerabilidades o code smells. Cada uno de los anteriores así mismo, puede ser un error bloqueante, crítico, mayor o menor.

A continuación, se pretende dar a conocer la metodología usada para afrontar estos errores y como se ha llevado a cabo el tratamiento de los mismos.

METODOLOGÍA

- 1) El grupo en su totalidad se ha reunido una vez que todas las tareas han sido realizadas para poder solucionar esta última tarea.
- 2) Se ha analizado grupalmente el código de la aplicación que resultaba en errores para sonarqube para poder tener una idea superficial de estos errores.
- 3) Se ha realizado una breve sesión de brainstorming en la que cada compañero ha dado ideas sobre como podría llevarse a cabo la corrección de estos errores.
- 4) Se ha asignado a una persona (Fernando Ruiz Robles) para implementar las opciones más viables obtenidas del brainstorming anterior.
- 5) Finalmente se han realizado la corrección de los errores y se realiza un nuevo análisis por parte de sonarqube para poder detectar cuan eficaz han sido estas correcciones.

3.1. VULNERABILIDADES

En este apartado se pretende analizar las vulnerabilidades detectadas por sonarqube y como se han solucionado, para ello se mostrarán capturas de las vulnerabilidades que se han presentado (no se mostrarán dos errores iguales a no ser que su corrección cambie ya que la solución de los mismos es similar).

1- Añadir parámetro method a los @RequestMapping.

(Categoría: Bloqueante, Número de errores: 2)

☐ Add a "method" parameter to this "@RequestMapping" annotation. [See Rule](#) 24 days ago L131

Vulnerability Blocker Open Not assigned 5min effort [Comment](#) cwe, owasp-a6, sans-top25-insecure, sp...

-Esta vulnerabilidad consistía en que los @RequestMapping precisaban de un parámetro method, la solución de la misma se da añadiendo este parámetro y su valor.

```
@RequestMapping(value = "/owners/{ownerId}/delete", method = RequestMethod.GET)
public String processDeleteOwner(@PathVariable("ownerId") int ownerId) {
    Owner owner = this.clinicService.findOwnerById(ownerId);
    this.clinicService.deleteOwner(owner);
    return "redirect:/owners/find";
}
```

-También se debe tener en cuenta que otra solución implementada es cambiar los @requestMapping por @GetMapping

```
@GetMapping(value = "/owners/{ownerId}/delete")
public String processDeleteOwner(@PathVariable("ownerId") int ownerId) {
    Owner owner = this.clinicService.findOwnerById(ownerId);
    this.clinicService.deleteOwner(owner);
    return "redirect:/owners/find";
}
```

2- Cambiar los @entity de las entidades.

(Categoría: Crítico, Número de errores: 14)

☐ Replace this persistent entity with a simple POJO or DTO object. [See Rule](#) 24 days ago L76

Vulnerability Critical Open Not assigned 10min effort [Comment](#) cwe, owasp-a6, sans-top25-insecure cwe, owasp-a5, spring

-Esta vulnerabilidad nos requería eliminar los @entity de las entidades. Este error no se ha solucionado ya que pensamos que eliminar los @entity haría que no funcionasen nuestras entidades, una posible solución planteada sería eliminarlos y en los controladores crear un objeto de la clase que se necesite y crear otro que fuese el persistente y simplemente llevar a cabo un mapping entre ambos objetos.

3.1. CODE SMELLS

-En este apartado se pretende analizar los code smells detectados por sonarqube y como se han solucionado, para ello se mostrarán capturas de los errores que se han presentado (no se mostrarán dos errores iguales a no ser que su corrección cambie ya que la solución de los mismos es similar).

1- Definir una constante en vez de duplicar un literal.

(Categoría: Crítico, Número de errores: 3)

Define a constant instead of duplicating this literal "redirect:/owners/{ownerId}" 3 times. [See Rule](#)

24 days ago L92

Code Smell Critical Open Not assigned 8min effort [Comment](#)

design

-Este error se ha solucionado simplemente declarando variables que se usarán en vez de duplicar literales, un ejemplo se muestra a continuación:

```
20 * Copyright 2002-2013 the original author or authors.
16 package org.springframework.samples.petclinic.web;
17
18 import org.springframework.beans.factory.annotation.Autowired;
33
34 /**
35  * @author Juergen Hoeller
36  * @author Ken Krebs
37  * @author Arjen Poutsma
38  */
39 @Controller
40 @RequestMapping("/owners/{ownerId}")
41 public class PetController {
42
43     private static final String VIEWS_PETS_CREATE_OR_UPDATE_FORM = "pets/createOrUpdatePetForm";
44
45     private final ClinicService clinicService;
46
47     private static final String VIEWS_REDIRECT_OWNERS = "redirect:/owners/{ownerId}";
48 }
```

```

@PostMapping(value = "/pets/new")
public String processCreationForm(Owner owner, @Valid Pet pet, BindingResult result, ModelMap model) {
    if (StringUtils.hasLength(pet.getName()) && pet.isNew() && owner.getPet(pet.getName(), true) != null) {
        result.rejectValue("name", "duplicate", "already exists");
    }
    if (result.hasErrors()) {
        model.put("pet", pet);
        return VIEWS_PETS_CREATE_OR_UPDATE_FORM;
    } else {
        owner.addPet(pet);
        this.clinicService.savePet(pet);
        return VIEWS_REDIRECT_OWNERS;
    }
}

@GetMapping(value = "/pets/{petId}/edit")
public String initUpdateForm(@PathVariable("petId") int petId, ModelMap model) {
    Pet pet = this.clinicService.findPetById(petId);
    model.put("pet", pet);
    return VIEWS_PETS_CREATE_OR_UPDATE_FORM;
}

@PostMapping(value = "/pets/{petId}/edit")
public String processUpdateForm(@Valid Pet pet, BindingResult result, Owner owner, ModelMap model) {
    if (result.hasErrors()) {
        model.put("pet", pet);
        return VIEWS_PETS_CREATE_OR_UPDATE_FORM;
    } else {
        owner.addPet(pet);
        this.clinicService.savePet(pet);
        return VIEWS_REDIRECT_OWNERS;
    }
}

```

2- Refactorizar para disminuir la complejidad.

(Categoría: Crítico, Número de errores: 1)

☐ Refactor this method to reduce its Cognitive Complexity from 33 to the 15 allowed. [See Rule](#)
13 minutes ago L29

Code Smell
Critical
Open
Not assigned
23min effort
Comment
brain-overload

-Este error ha sido uno de los que más nos ha costado solucionar ya que no se nos ocurría ninguna forma. Al final se han realizado una serie de cambios para solucionar dicho error. A continuación, se dejan los cambios realizados.

-Básicamente se está realizando una llamada a la base de datos para buscar una residencia que incumpla lo que se quiere validar. Para ello se cambió el validator y se creó una query count que buscase lo nombrado anteriormente en la base de datos

```

32
33     if (residence.getRegisterDate() == null) {
34         errors.rejectValue(registerDateS, REQUIRED, REQUIRED);
35     }
36
37     if (residence.getReleaseDate() == null) {
38         errors.rejectValue(releaseDateS, REQUIRED, REQUIRED);
39     }
40
41     if (residence.getRegisterDate() != null && residence.getReleaseDate() != null) {
42
43         LocalDate registerDate = residence.getRegisterDate();
44         LocalDate releaseDate = residence.getReleaseDate();
45         LocalDate now = LocalDate.now();
46
47         if (registerDate.isBefore(now)) {
48             errors.rejectValue(registerDateS, REQUIRED,
49                 "La fecha de registro tiene que ser igual o posterior a la actual");
50         }
51         if (releaseDate.isBefore(now)) {
52             errors.rejectValue(releaseDateS, REQUIRED,
53                 "La fecha de salida tiene que ser igual o posterior a la actual");
54         }
55         if (releaseDate.isBefore(registerDate)) {
56             errors.rejectValue(releaseDateS, REQUIRED,
57                 "La fecha de salida tiene que ser igual o posterior a la de registro");
58         }
59
60         Integer count = this.clinicService.compruebaResidence(registerDate, releaseDate,
61             residence.getPet().getId());
62         if (count != 0 && count != null) {
63             errors.rejectValue(registerDateS, REQUIRED, "El registro no puede coincidir con uno anterior");
64         }
65     }
66 }

```

-Aquí está la query mencionada anteriormente declarada en los repositorios.

```

package org.springframework.samples.petclinic.repository.springdatajpa;

import java.time.LocalDate;

public interface SpringDataResidenceRepository extends ResidenceRepository, Repository<Residence, Integer>{

    @Query("SELECT count(r) FROM Residence r WHERE ((r.registerDate < :register AND r.releaseDate > :register) OR "
        + "(r.registerDate < :release AND r.releaseDate > :release) OR "
        + "(r.registerDate > :register AND r.releaseDate < :release) OR "
        + "(r.registerDate < :register AND r.releaseDate > :release) OR "
        + "r.registerDate = :register OR r.releaseDate = :release) AND r.pet.id = :petId")
    Integer count(@Param("register") LocalDate register, @Param("release") LocalDate release, @Param("petId") int petId);
}

public interface ResidenceRepository {

    void save(Residence residence);

    List<Residence> findByPetId(Integer petId);

    void delete(Residence residence);

    Residence findById(int id);

    Integer count(LocalDate register, LocalDate release, int petId);
}

```

-Y el método en el servicio.

```

public Integer compruebaResidence(LocalDate register, LocalDate release, int petId) {
    return this.residenceRepository.count(register, release, petId);
}

```

-Y por último aplicamos estos cambios en el validator

```
if (residence.getRegisterDate() != null && residence.getReleaseDate() != null) {  
    LocalDate registerDate = residence.getRegisterDate();  
    LocalDate releaseDate = residence.getReleaseDate();  
    LocalDate now = LocalDate.now();  
  
    if (registerDate.isBefore(now)) {  
        errors.rejectValue(registerDateS, REQUIRED,  
            "La fecha de registro tiene que ser igual o posterior a la actual");  
    }  
    if (releaseDate.isBefore(now)) {  
        errors.rejectValue(releaseDateS, REQUIRED,  
            "La fecha de salida tiene que ser igual o posterior a la actual");  
    }  
    if (releaseDate.isBefore(registerDate)) {  
        errors.rejectValue(releaseDateS, REQUIRED,  
            "La fecha de salida tiene que ser igual o posterior a la de registro");  
    }  
    Integer count = this.clinicService.compruebaResidence(registerDate, releaseDate,  
        residence.getPet().getId());  
    if (count != 0 && count != null) {  
        errors.rejectValue(registerDateS, REQUIRED, "El registro no puede coincidir con uno anterior");  
    }  
}
```

3- Borrar bloque de código comentado.

(Categoría: Mayor, Número de errores: 1)



This block of commented-out lines of code should be removed. [See Rule](#)

24 days ago L17

Code Smell Major Open Not assigned 5min effort Comment

based-on-misra, unused

-Este error ha consistido en simplemente borrar un bloque de código que estaba comentado. No cabe destacar nada más.

```
//  
@ControllerAdvice  
public class ExceptionHandlerConfiguration  
{  
    // @Autowired  
    // private BasicErrorController errorController;  
    // // add any exceptions/validations/binding problems  
  
    @ExceptionHandler(Exception.class)  
    public String defaultErrorHandler(HttpServletRequest request, Exception ex) {  
        request.setAttribute("javax.servlet.error.request_uri", request.getPathInfo());  
        request.setAttribute("javax.servlet.error.status_code", 400);  
        request.setAttribute("exception", ex);  
        return "exception";  
    }  
}
```

4- Renombrar campo vets.

(Categoría: Mayor, Número de errores: 1)

☐ Rename field "vets" [See Rule](#)

24 days ago ▾ L33

Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 10min effort [Comment](#)

brain-overload ▾

-Este error consistía en que un campo de la entidad vets, se llamaba vets también para solucionarlo hemos usado ALT + SHIFT + R y renombrado vets por veterinarians

```
@XmlRootElement
public class Vets {

    private List<Vet> vets;

    @XmlElement
    public List<Vet> getVetList() {
        if (vets == null) {
            vets = new ArrayList<>();
        }
        return vets;
    }
}
```

5- Añadir un constructor.

(Categoría: Mayor, Número de errores: 1)

Add a private constructor to hide the implicit public one. [See Rule](#)

24 days ago ▾ L33

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

design ▾

-Este error simplemente se ha solucionado añadiendo un constructor vacío a la entidad que lo requería. Nada más que comentar.

6- Definir un throw en vez de usar el Throw genérico.

(Categoría: Mayor, Número de errores: 1)

Define and throw a dedicated exception instead of using a generic one. [See Rule](#)

24 days ago ▾ L34 🔗 🔍 ▾

🔧 Code Smell ▾ 🚨 Major ▾ 🔓 Open ▾ Not assigned ▾ 20min effort [Comment](#)

🔗 cert, cwe, error-handling ▾

-Este error simplemente se ha solucionado creando una clase que extienda de Exception y usandola en vez del throw genérico.

```
package org.springframework.samples.petclinic.service;

public class TimeEndException extends Exception {

    public TimeEndException(String string) {
    }

    /**
     *
     */
    private static final long serialVersionUID = 1L;
}
```

Los siguientes code smells son menores y se explicará su resolución de forma más simple ya que requieren de modificaciones muy cortas.

- 1) Cambiar el nombre de una variable en visitRepository y CauseController (empezaban por mayúscula en vez de minúscula)
- 2) Cambiar la clase EntityUtils a una interfaz.
- 3) Cambiar una expresión lambda(`d -> d.getAmount()` por `Donation::getAmount`)

3.3. CONCLUSIÓN

Como conclusión, cabe destacar que hemos solucionado todos los code smells (56) y quedan pendiente 14 vulnerabilidades que no hemos podido solucionar ya que como grupo pensamos que cambiarían el funcionamiento de nuestra aplicación.

Por lo general estamos bastante satisfechos con las correcciones realizadas y hemos trabajado conjuntamente para poder solucionarlas de manera rápida y efectiva. Aun que hemos tenido problemas ya mencionados anteriormente o dificultades a la hora de realizar algunas correcciones (Refactorizar para disminuir la complejidad) pensamos que el trabajo realizado está a la medida de las expectativas del grupo.

Nuestra primera toma de contacto usando la herramienta sonarqube ha sido muy positiva aun que hemos tenido problemas a la hora de poder mostrar los errores que habían sido ya corregidos. Por lo general, la herramienta nos parece de gran utilidad y nos ha permitido detectar pequeños errores que, como programadores en formación, podemos llegar a cometer y pueden significar una vulnerabilidad para nuestra aplicación o crear un bloqueo en la misma.