

Actividades de Transferencia del Conocimiento

David S. Madrid Cardozo, Laura V. Hidalgo Melo, Johan S. Zapata Talero & Javier M. Diaz

Sanabria.

Servicio Nacional de Aprendizaje (SENA)

Técnica

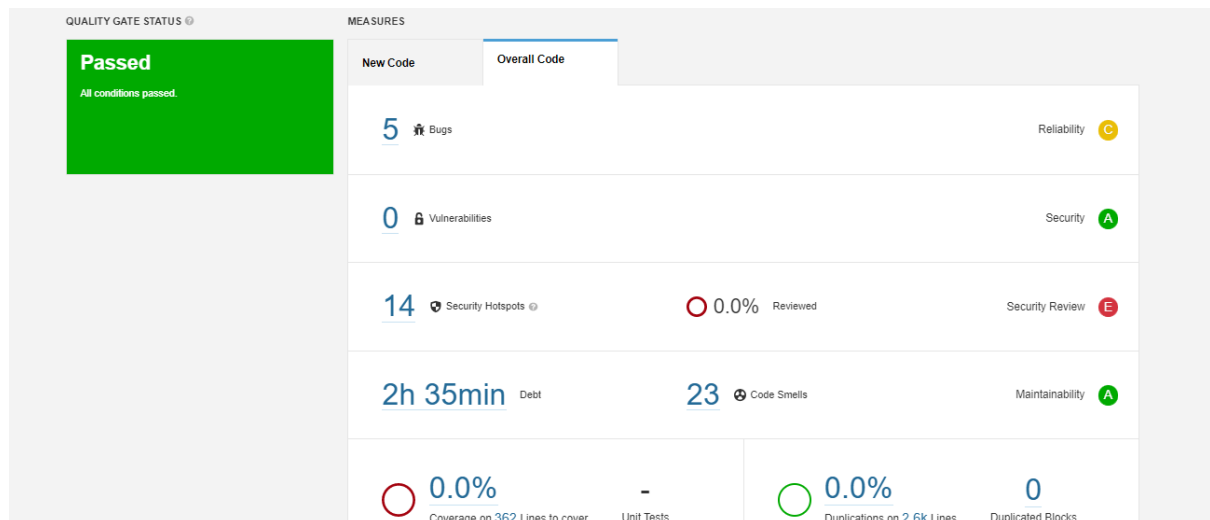
Análisis y Desarrollo de Sistemas de Información

Ing. Miguel Angel López Cacho

28/03/2022

Informe SonarQube Proyecto Webanimal

Es muy importante entregar productos con calidad, ya que esto puede disminuir costos y permite aumentar la rentabilidad del proyecto, para esto se utilizará la herramienta SonarQube es una plataforma de código abierto para la inspección de calidad del código, brinda información sobre código duplicado, estándares de codificación, errores potenciales de código y seguridad, etc. En la siguiente imagen se muestra un informe general de la evaluación al proyecto y sus soluciones.



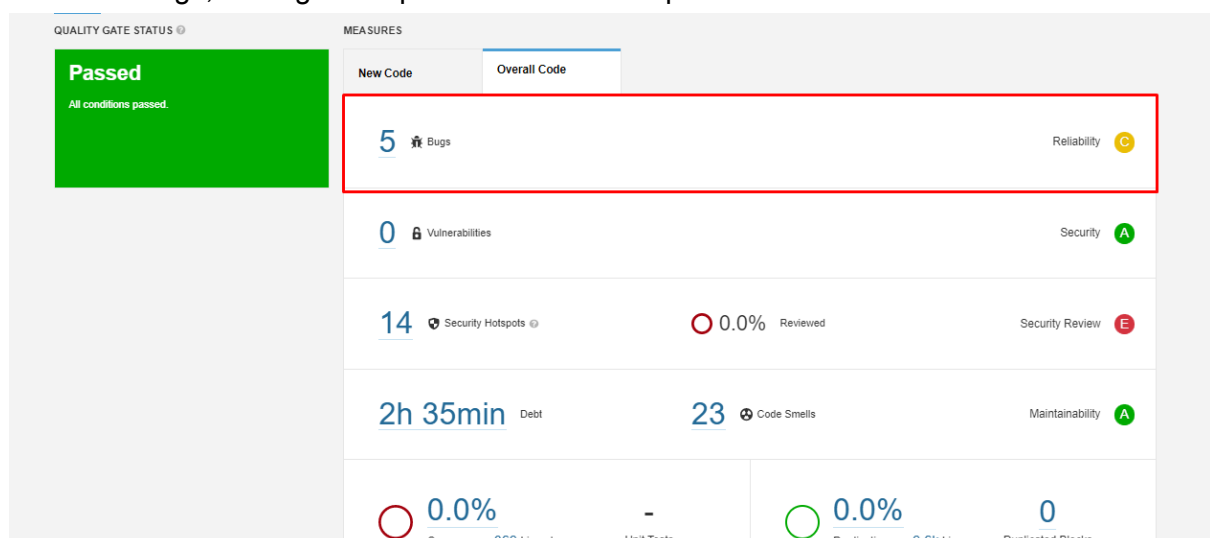
Quality Gate Status

Según el análisis realizado por SonarQube, en general, el proyecto cuenta con la mayoría de condiciones en un estado favorable ya que se ha podido disminuir los bugs para que se cumpla y se realice la evaluación antes de ser enviados a producción.

Fiabilidad

Bugs

Segun los reportes en el aplicativo web SonarQube analizó el software y dio a conocer 5 bugs, un bug es un punto de fallo real o potencial en el software



Bugs Presentados

A continuación, veremos los bugs que nos dio a conocer SonarQube para poderlos identificar y corregir:

Su solución y sus bugs a conocer es que faltan etiquetas de html como <title>, entre otros. Debido a que estamos trabajando con bootstrap y el backend de Django, no se implementó.

Seguridad Vulnerabilidades

En el proyecto podemos encontrar las vulnerabilidades que son un agujero de seguridad que puede usarse para atacar el software, en este caso, nuestro proyecto Webanimal en la etapa de vulnerabilidad tiene una calificación A, ya que no hay un problema que afecte la seguridad de la aplicación.

Revisión de Seguridad

Puntos de acceso de Seguridad

Filters

Assigned to meAll

Status

To review

Overall code

Security Hotspots Reviewed0.0%

14 Security Hotspots to review

Review priority:HIGH

Authentication3

"password" detected here, review this potentially hard-coded credential.
TO REVIEW

"password" detected here, review this potentially hard-coded credential.
TO REVIEW

"password" detected here, review this potentially hard-coded credential.
TO REVIEW

Cross-Site Request Forgery (CSRF)10

Review priority:LOW

Insecure Configuration1

"password" detected here, review this potentially hard-coded credential.
Hard-coded credentials are security-sensitive python:S2068

CategoryAuthentication

Review priorityHIGH

AssigneeNot assigned

Status: To review
This Security Hotspot needs to be reviewed to assess whether the code poses a risk.
Change status

mysite/db.py

```
11 POSTGRESQL = {
12     'default': {
13         'ENGINE': 'django.db.backends.postgresql_psycopg2',
14         'NAME': 'personal',
15         'USER': 'postgres',
16         'PASSWORD': 'postgres',
17         'HOST': 'localhost',
18         'PORT': '5432'
19     }
20 }
21
```

Filters

Assigned to meAll

Status

To review

Overall code

Security Hotspots Reviewed0.0%

Authentication3

"password" detected here, review this potentially hard-coded credential.
TO REVIEW

"password" detected here, review this potentially hard-coded credential.
TO REVIEW

"password" detected here, review this potentially hard-coded credential.
TO REVIEW

Cross-Site Request Forgery (CSRF)10

Make sure allowing safe and unsafe HTTP methods is safe here.
TO REVIEW

Make sure allowing safe and unsafe HTTP methods is safe here.
TO REVIEW

Make sure allowing safe and unsafe HTTP methods is safe here.
TO REVIEW

Make sure allowing safe and unsafe HTTP methods is safe here.
Allowing both safe and unsafe HTTP methods is security-sensitive python:S3752

CategoryCross-Site Request Forgery (CSRF)

Review priorityHIGH

AssigneeNot assigned

Status: To review
This Security Hotspot needs to be reviewed to assess whether the code poses a risk.
Change status

Adoption/views.py

```
10
11
12
13
14 # Create your views here.
15 def frmAdoption(request):
16     form = AdoptionForm(request.POST or None)
17
18     if request.method == 'POST' and form.is_valid():
19         user = form.save()
20
```

Filters

Assigned to meAll

Status

To review

Overall code

Security Hotspots Reviewed0.0%

TO REVIEW

Make sure allowing safe and unsafe HTTP methods is safe here.
TO REVIEW

Make sure allowing safe and unsafe HTTP methods is safe here.
TO REVIEW

Make sure allowing safe and unsafe HTTP methods is safe here.
TO REVIEW

Make sure allowing safe and unsafe HTTP methods is safe here.
TO REVIEW

Review priority:LOW

Insecure Configuration1

Make sure this debug feature is deactivated before delivering the code in production.
TO REVIEW

Make sure this debug feature is deactivated before delivering the code in production.
Delivering code in production with debug features activated is security-sensitive python:S4507

CategoryInsecure Configuration

Review priorityLOW

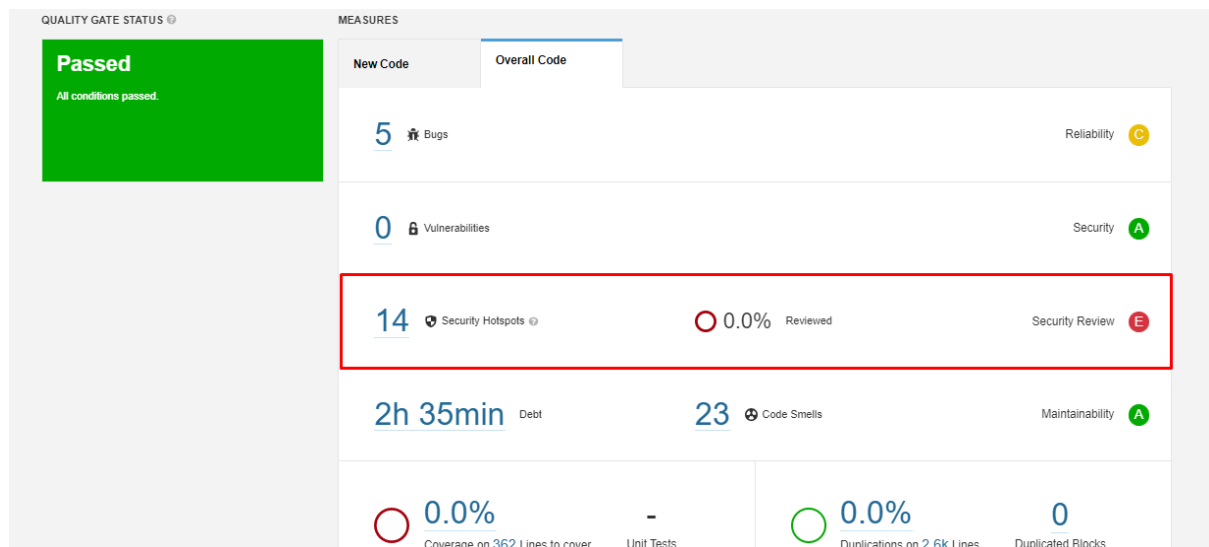
AssigneeNot assigned

Status: To review
This Security Hotspot needs to be reviewed to assess whether the code poses a risk.
Change status

mysite/settings.py

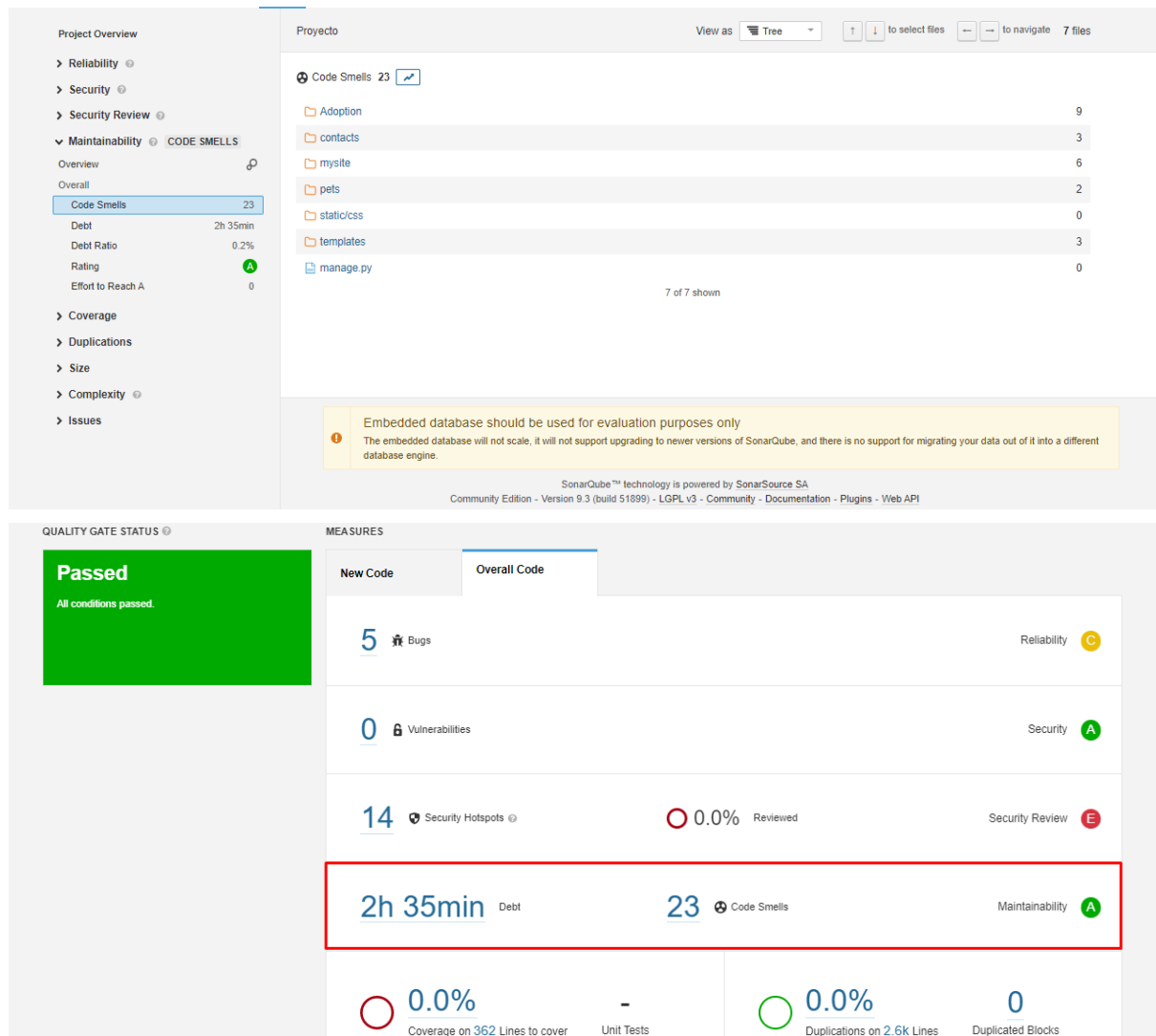
```
22
23 # SECURITY WARNING: keep the secret key used in production secret!
24 SECRET_KEY = 'django-insecure-cqc9z-d5t2a1%z%zso_1@zu72ex6d&x&e=(xazr9=tw1ocee'
25
26 # SECURITY WARNING: don't run with debug turned on in production!
27 DEBUG = True
28
29 ALLOWED_HOSTS = []
30
31
32 # Application definition
```

El Security Hotspot en el proyecto hay fragmento de código sensible a la seguridad en el password para la conexión a la base de datos, sin embargo, se pondrá opcionalmente la contraseña en una variable de entorno virtual, por ende no existe un riesgo alto de que afecte la seguridad en general de ampliación, en este caso, no necesaria una corrección para asegurar el código.



Mantenibilidad

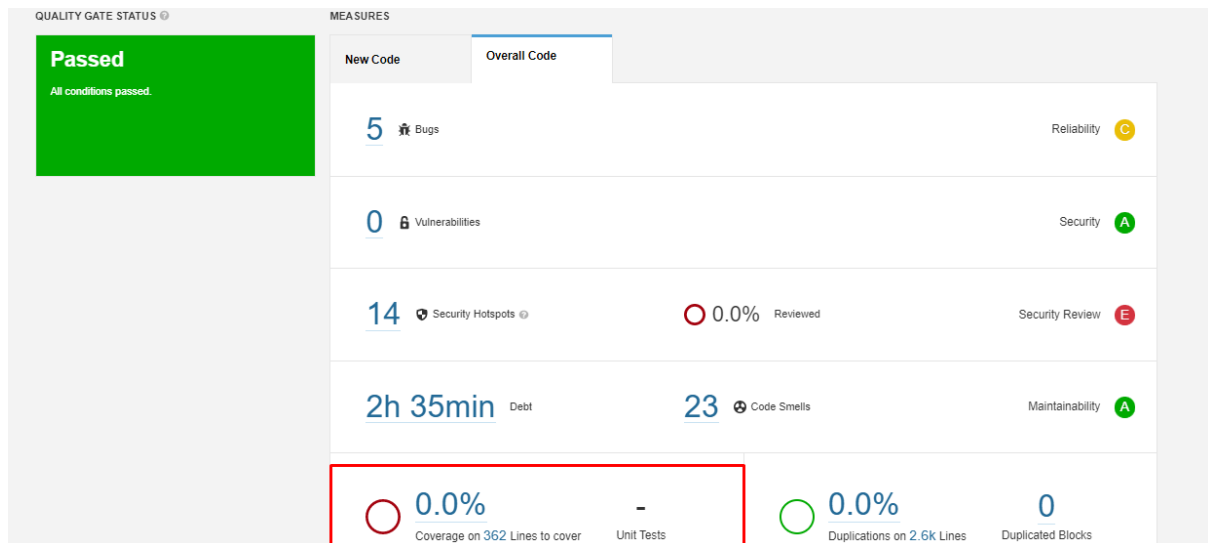
Según SonarCube, Code Smells es un problema relacionado con la mantenibilidad del código, dicho esto podemos encontrar 7 valores hallados, a continuación, se mostrarán los resultados:



Cobertura

En este apartado, la métrica de cobertura, refleja gráficamente cuantas líneas de código cuentan con pruebas unitarias. En la siguiente imagen muestra que el porcentaje es equivalente a cero, puesto que para realizar dichas pruebas se deben implementar otros

tipos de herramientas, tales como JaCoCo o Junit.



Código Repetido

Gracias a las librerías del framework usado, Django, podemos hacer muy bien la revisión del código repetido usando la herencia entre templates, de esta manera no tendremos ni una sola línea de código repetida en todo el proyecto, haciéndolo menos pesado y más accesible.

