Escuela Politécnica Nacional

- Code Cat Creations -

Reporte SonarQube

Introducción

SonarQube es una plataforma que permite inspeccionar la calidad del código fuente en diversos lenguajes de programación, proporcionando análisis automáticos de calidad. La herramienta se basa en detectar "code smells" (malos olores en el código), vulnerabilidades, errores de mantenimiento y otros problemas que puedan afectar la eficiencia y seguridad del software. En el marco del análisis de calidad del código para el proyecto "Traductor Braille a Texto", se realizó un escaneo exhaustivo utilizando SonarQube, cuyos resultados se detallan en este informe.
El análisis de código es esencial para garantizar que el software desarrollado cumpla con estándares de calidad, reduciendo la probabilidad de errores en producción y mejorando la mantenibilidad del código. SonarQube permite medir y asegurar la calidad del código, con métricas clave como seguridad, mantenibilidad, y cobertura de pruebas.

Desarrollo

Resultados del Análisis con SonarQube:

- **Resumen General:**
  - **Quality Gate:** Pasado.
  - **Líneas de código analizadas:** 2.9K.
  - **Último análisis:** Hace 27 minutos.

- **Métricas Clave:**
  - **Seguridad:**
    - **Issues abiertos:** 1
    - **Nivel de severidad:** Bloqueante
  - **Fiabilidad:**
    - **Issues abiertos:** 0
  - **Mantenibilidad:**
    - **Issues abiertos:** 15
    - **Esfuerzo técnico requerido:** 9 horas
  - **Cobertura de pruebas:** 0%
  - **Duplicaciones:** 0%

- **Detalles de los Issues Identificados:**
  - **Seguridad:**
    - Se detectó un issue relacionado con la correcta gestión del token de SonarQube. Es crucial asegurarse de que este token sea revocado, cambiado y eliminado del código para evitar vulnerabilidades.
  - **Mantenibilidad:**
    - Se detectaron varios "code smells" relacionados con la complejidad cognitiva y la duplicación de literales. Ejemplos incluyen:
    - Refactorización de una función que excede la complejidad cognitiva permitida.
    - Duplicación de literales en varios lugares del código que podrían ser manejados con constantes.
    - Reemplazo de `require_once` con un mecanismo de importación de espacios de nombres más eficiente.
  - **Ejemplos de Issues en Archivos Específicos:**
    - `app/Http/Controllers/Braille/Exped/TranslationsController.php`: Refactorizar la función para reducir su complejidad cognitiva de 24 a 15.
    - `config/database.php`: Definir una constante en lugar de duplicar el literal "127.0.0.1" cuatro veces.

Conclusiones y Recomendaciones

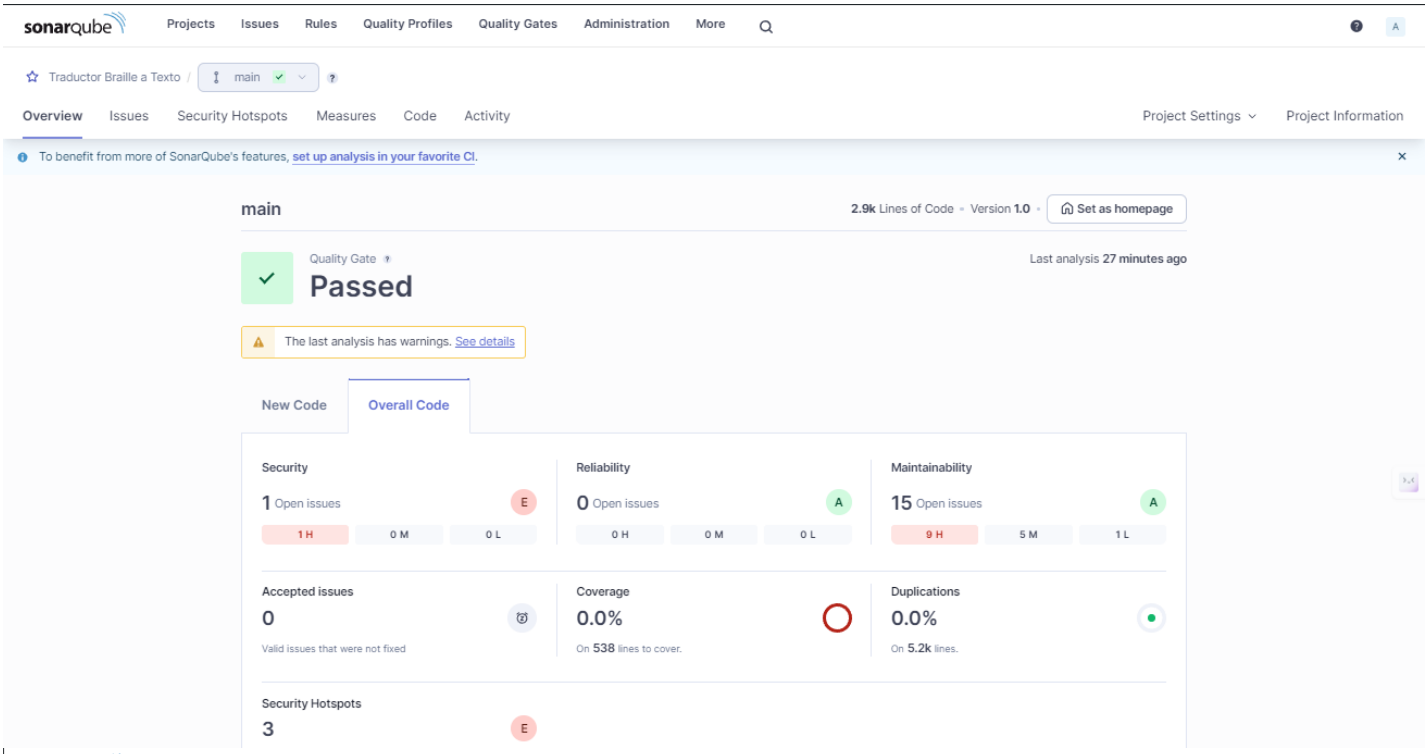
Conclusiones:
El análisis realizado con SonarQube reveló que, aunque el código pasó las verificaciones generales de calidad (Quality Gate), existen áreas específicas que requieren atención, particularmente en términos de mantenibilidad y seguridad. La complejidad del código y la duplicación de literales son problemas recurrentes que podrían impactar negativamente la capacidad de mantener y escalar el software en el futuro.

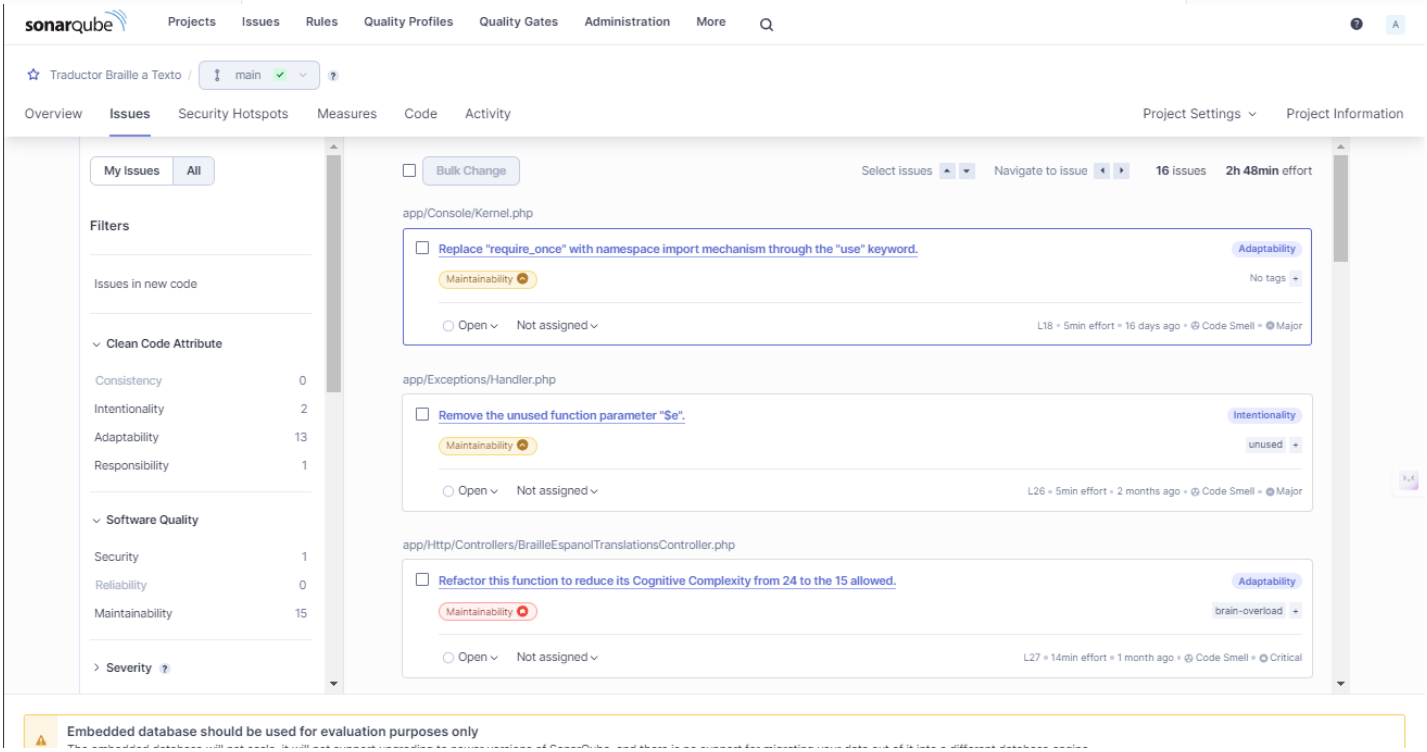
Recomendaciones:

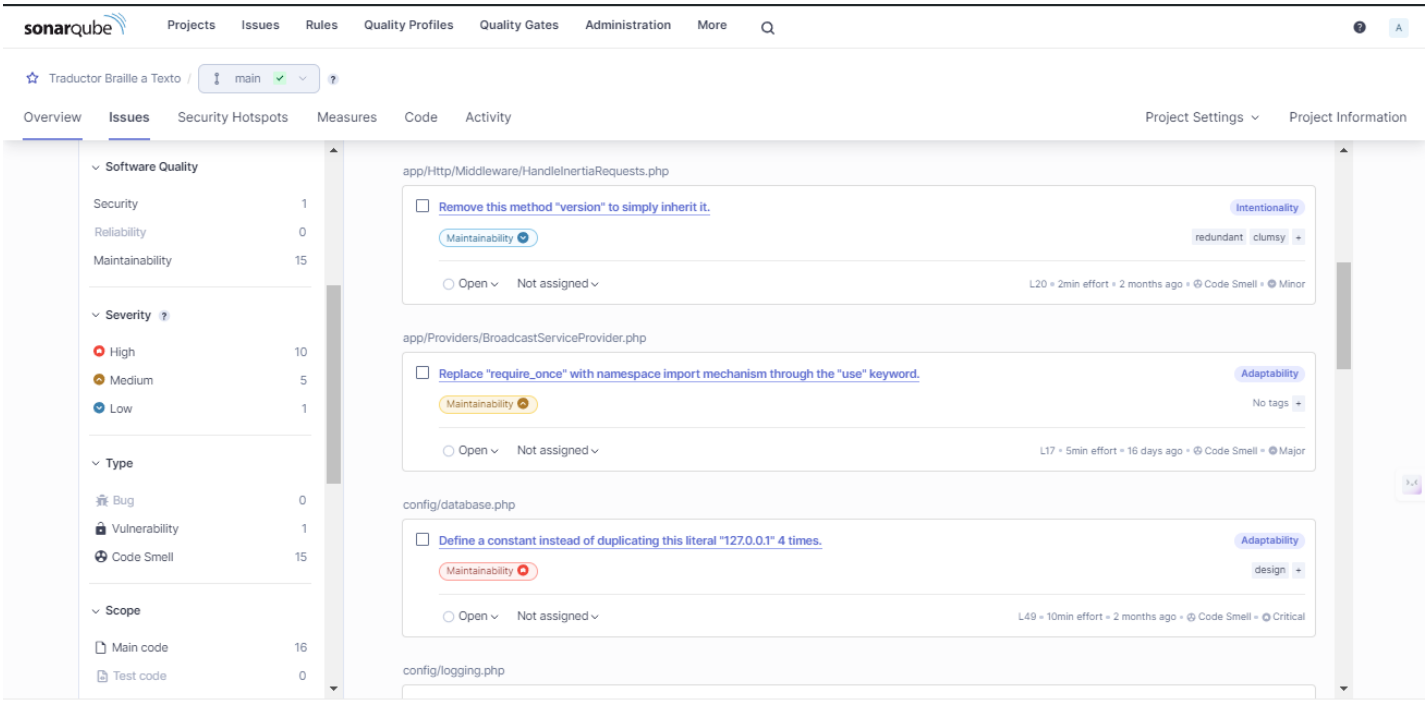
- Se recomienda realizar una refactorización de las funciones que presentan alta complejidad cognitiva para reducir la probabilidad de errores y facilitar su mantenimiento.
- Implementar constantes para literales que se repiten en el código, lo que no solo mejorará la mantenibilidad, sino también la claridad del código.
- Se debe revisar y corregir el issue de seguridad identificado relacionado con la gestión del token de SonarQube, garantizando que no quede expuesto en el código fuente.

Anexo

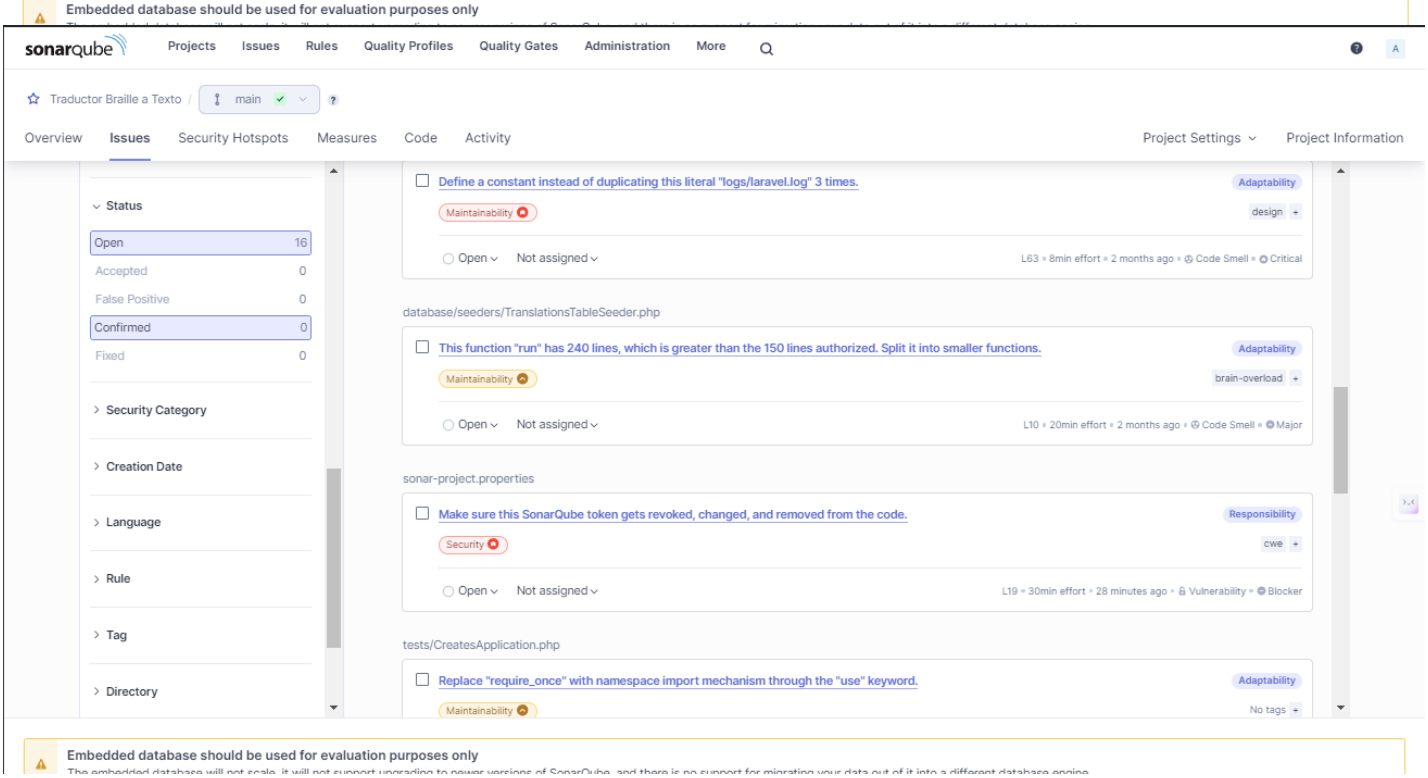
Como anexo, se adjunta las capturas tomadas de la aplicación SonarQube


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Traductor Braille a Texto
main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
To benefit from more of SonarQube's features, set up analysis in your favorite CI.
main
2.9k Lines of Code · Version 1.0 · Set as homepage
Quality Gate
Passed
Last analysis 27 minutes ago
The last analysis has warnings. See details
New Code
Overall Code
Security
1 Open issues
E
1 H
0 M
0 L
Reliability
0 Open issues
A
0 H
0 M
0 L
Maintainability
15 Open issues
A
9 H
5 M
1 L
Accepted issues
0
Valid issues that were not fixed
Coverage
0.0%
On 538 lines to cover.
Duplications
0.0%
On 5.2k lines.
Security Hotspots
3
E

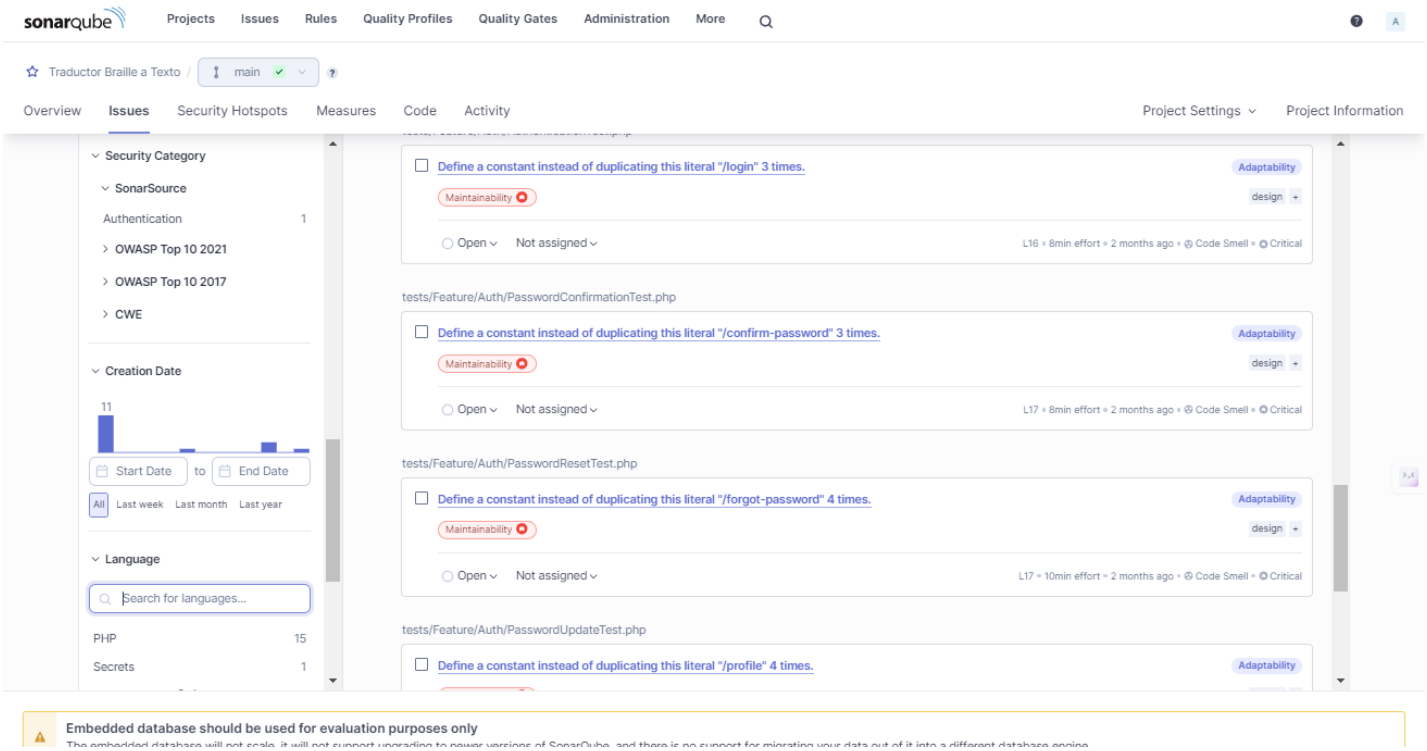

sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Traductor Braille a Texto
main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
My Issues
All
Filters
Issues in new code
Clean Code Attribute
Consistency 0
Intentionality 2
Adaptability 13
Responsibility 1
Software Quality
Security 1
Reliability 0
Maintainability 15
Severity
Bulk Change
Select issues
Navigate to issue
16 issues
2h 48min effort
app/Console/Kernel.php
Replace "require_once" with namespace import mechanism through the "use" keyword.
Adaptability
Maintainability
No tags
Open
Not assigned
L18 · 5min effort · 16 days ago · Code Smell · Major
app/Exceptions/Handler.php
Remove the unused function parameter "$e".
Intentionality
Maintainability
unused
Open
Not assigned
L26 · 5min effort · 2 months ago · Code Smell · Major
app/Http/Controllers/BrailleEspanolTranslationsController.php
Refactor this function to reduce its Cognitive Complexity from 24 to the 15 allowed.
Adaptability
Maintainability
brain-overload
Open
Not assigned
L27 · 14min effort · 1 month ago · Code Smell · Critical
Embedded database should be used for evaluation purposes only



Embedded database should be used for evaluation purposes only

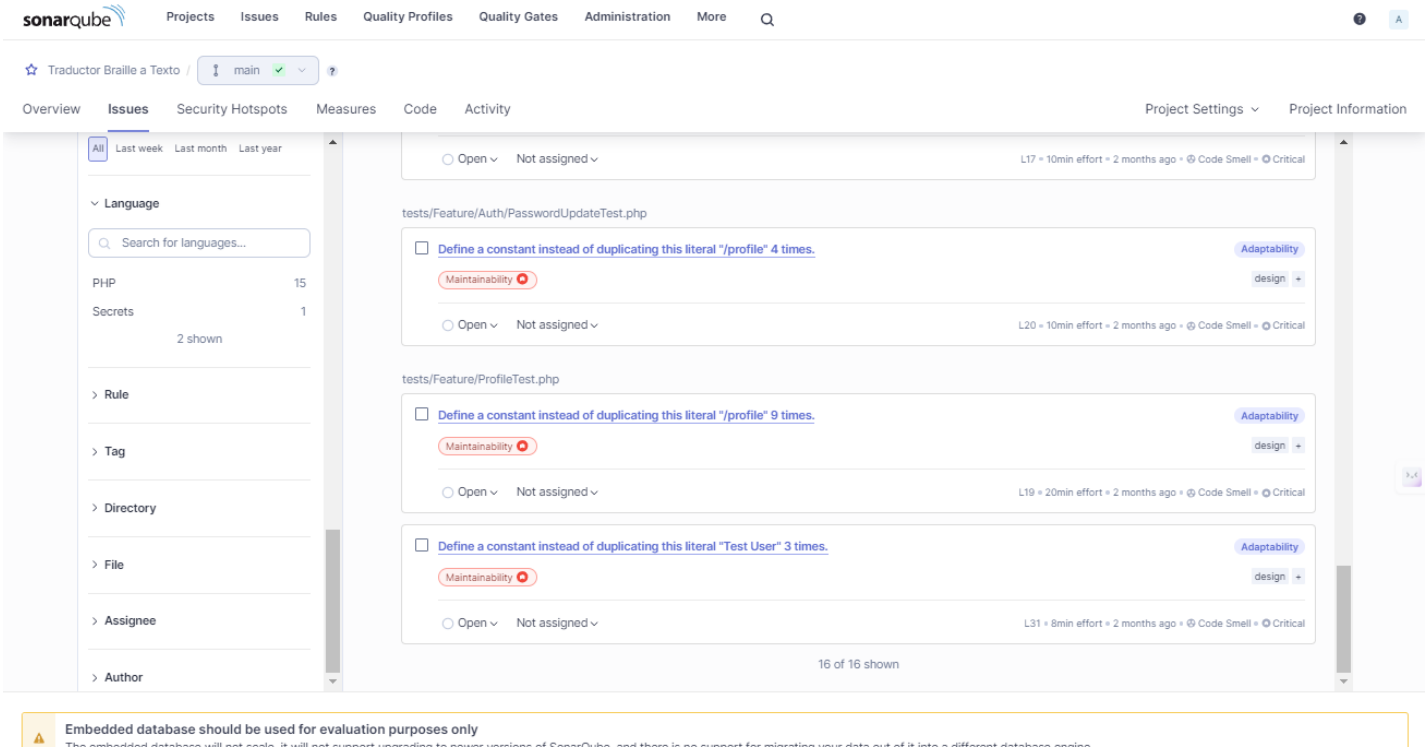


Embedded database should be used for evaluation purposes only



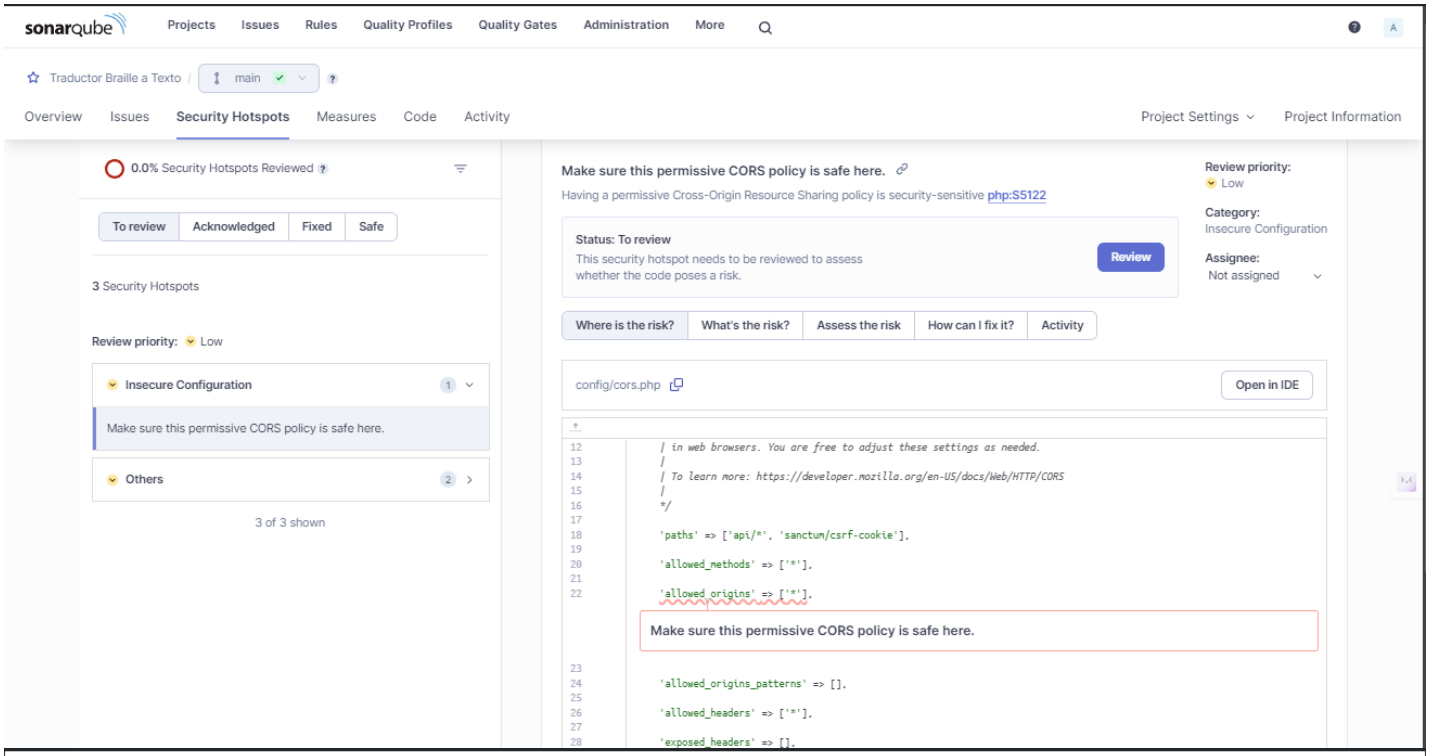
Embedded database should be used for evaluation purposes only

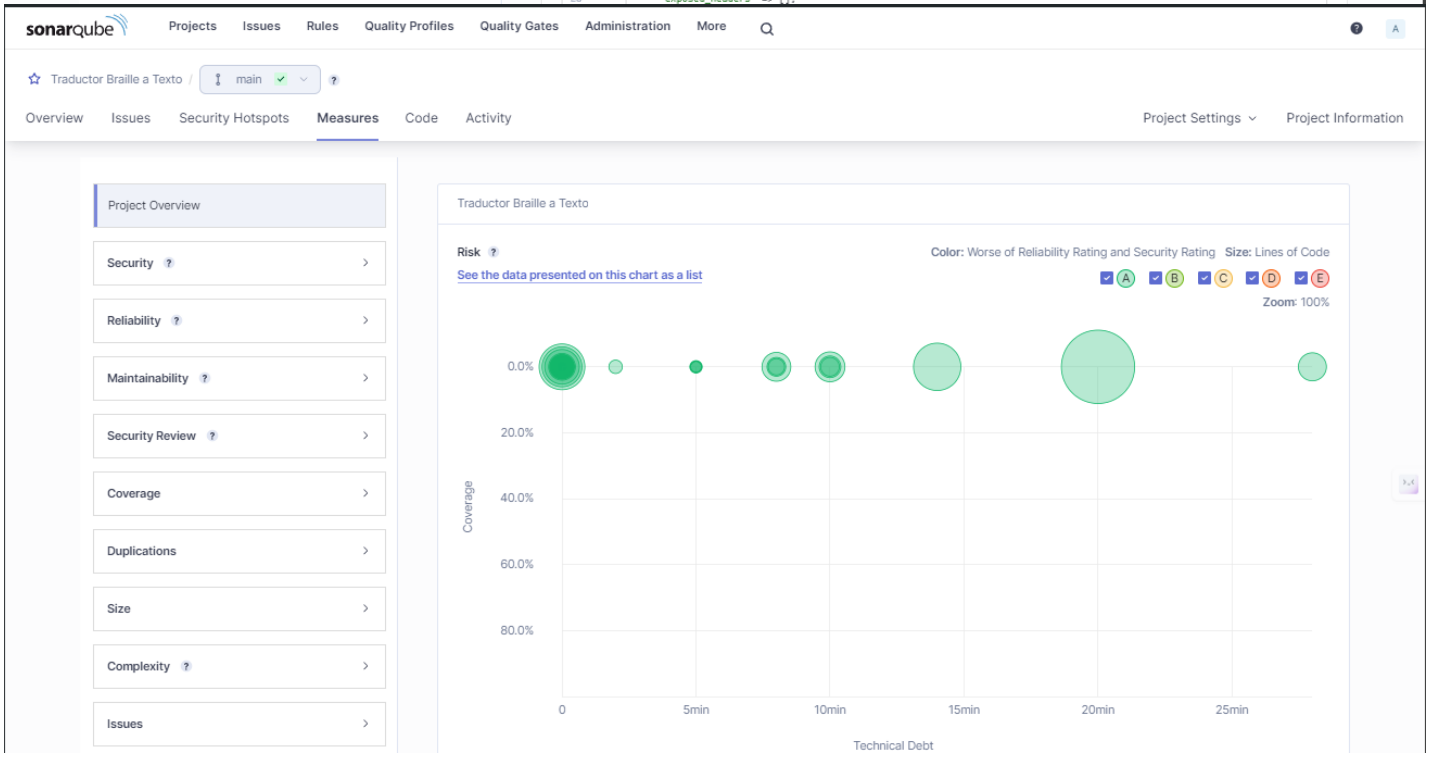
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine



Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Traductor Braille a Texto
main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
0.0% Security Hotspots Reviewed
To review
Acknowledged
Fixed
Safe
3 Security Hotspots
Review priority: Low
Insecure Configuration
Make sure this permissive CORS policy is safe here.
Others
3 of 3 shown
Make sure this permissive CORS policy is safe here.
Having a permissive Cross-Origin Resource Sharing policy is security-sensitive php:S5122
Status: To review
This security hotspot needs to be reviewed to assess whether the code poses a risk.
Review
Review priority: Low
Category: Insecure Configuration
Assignee: Not assigned
Where is the risk?
What's the risk?
Assess the risk
How can I fix it?
Activity
config/cors.php
Open in IDE
'allowed_origins' => ['*'],
Make sure this permissive CORS policy is safe here.


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
Traductor Braille a Texto
main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
Project Overview
Security
Reliability
Maintainability
Security Review
Coverage
Duplications
Size
Complexity
Issues
Traductor Braille a Texto
Risk
See the data presented on this chart as a list
Color: Worse of Reliability Rating and Security Rating
Size: Lines of Code
Zoom: 100%
0.0%
20.0%
40.0%
60.0%
80.0%
Coverage
0
5min
10min
15min
20min
25min
Technical Debt