

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

Tractor 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 No tags L18 • 5min effort • 16 days ago • @ Code Smell • Major

app/Exceptions/Handler.php

- ☐ Remove the unused function parameter "Se". Intentionality unused 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 brain-overload L27 • 14min effort • 1 month ago • @ Code Smell • Critical

Embedded database should be used for evaluation purposes only

sonarqube

Projects

Issues

Rules

Quality Profiles

Quality Gates

Administration

More

Tractor Braille a Texto

main

Overview

Issues

Security Hotspots

Measures

Code

Activity

Project Settings

Project Information

Software Quality

Security

1

Reliability

0

Maintainability

15

Severity

High

10

Medium

5

Low

1

Type

Bug

0

Vulnerability

1

Code Smell

15

Scope

Main code

16

Test code

0

app/Http/Middleware/HandleInertiaRequests.php

Remove this method "version" to simply inherit it.

Maintainability

Open

Not assigned

L20 • 2min effort • 2 months ago • Code Smell • Minor

app/Providers/BroadcastServiceProvider.php

Replace "require_once" with namespace import mechanism through the "use" keyword.

Maintainability

Open

Not assigned

L17 • 5min effort • 16 days ago • Code Smell • Major

config/database.php

Define a constant instead of duplicating this literal "127.0.0.1" 4 times.

Maintainability

Open

Not assigned

L49 • 10min effort • 2 months ago • Code Smell • Critical

config/logging.php

Define a constant instead of duplicating this literal "logs/laravel.log" 3 times.

Maintainability

Open

Not assigned

L63 • 8min effort • 2 months ago • Code Smell • Critical

database/seeds/TranslationsTableSeeder.php

This function "run" has 240 lines, which is greater than the 150 lines authorized. Split it into smaller functions.

Maintainability

Open

Not assigned

L10 • 20min effort • 2 months ago • Code Smell • Major

sonar-project.properties

Make sure this SonarQube token gets revoked, changed, and removed from the code.

Security

Open

Not assigned

L19 • 30min effort • 28 minutes ago • Vulnerability • Blocker

tests/CreatesApplication.php

Replace "require_once" with namespace import mechanism through the "use" keyword.

Maintainability

Open

Not assigned

L10 • 20min effort • 2 months ago • Code Smell • Major

sonarqube

Projects

Issues

Rules

Quality Profiles

Quality Gates

Administration

More

Tractor Braille a Texto

main

Overview

Issues

Security Hotspots

Measures

Code

Activity

Project Settings

Project Information

Status

Open

16

Accepted

0

False Positive

0

Confirmed

0

Fixed

0

Security Category

Creation Date

Language

Rule

Tag

Directory

Define a constant instead of duplicating this literal "logs/laravel.log" 3 times.

Maintainability

Open

Not assigned

L63 • 8min effort • 2 months ago • Code Smell • Critical

database/seeds/TranslationsTableSeeder.php

This function "run" has 240 lines, which is greater than the 150 lines authorized. Split it into smaller functions.

Maintainability

Open

Not assigned

L10 • 20min effort • 2 months ago • Code Smell • Major

sonar-project.properties

Make sure this SonarQube token gets revoked, changed, and removed from the code.

Security

Open

Not assigned

L19 • 30min effort • 28 minutes ago • Vulnerability • Blocker

tests/CreatesApplication.php

Replace "require_once" with namespace import mechanism through the "use" keyword.

Maintainability

Open

Not assigned

L10 • 20min effort • 2 months ago • Code Smell • Major

Historia De Usuario

5

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More ? A

Tractor Braille a Texto / main ?

Overview **Issues** Security Hotspots Measures Code Activity Project Settings Project Information

Security Category

SonarSource

Authentication 1

OWASP Top 10 2021

OWASP Top 10 2017

CWE

Creation Date

11

Start Date to End Date

All Last week Last month Last year

Language

Search for languages...

PHP 15

Secrets 1

☐ Define a constant instead of duplicating this literal "/login" 3 times. Adaptability design

Maintainability ?

Open Not assigned L16 • 8min effort • 2 months ago • ? Code Smell • ? Critical

tests/Feature/Auth/PasswordConfirmationTest.php

☐ Define a constant instead of duplicating this literal "/confirm-password" 3 times. Adaptability design

Maintainability ?

Open Not assigned L17 • 8min effort • 2 months ago • ? Code Smell • ? Critical

tests/Feature/Auth/PasswordResetTest.php

☐ Define a constant instead of duplicating this literal "/forgot-password" 4 times. Adaptability design

Maintainability ?

Open Not assigned L17 • 10min effort • 2 months ago • ? Code Smell • ? Critical

tests/Feature/Auth/PasswordUpdateTest.php

☐ Define a constant instead of duplicating this literal "/profile" 4 times. Adaptability design

Maintainability ?

Embedded database should be used for evaluation purposes only
 The embedded database will not scale, it will not support migration to newer versions of SonarQube, and there is no support for migration your data out of it into a different database engine.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More ? A

Tractor Braille a Texto / main ?

Overview **Issues** Security Hotspots Measures Code Activity Project Settings Project Information

All Last week Last month Last year

Language

Search for languages...

PHP 15

Secrets 1

2 shown

Rule

Tag

Directory

File

Assignee

Author

Open Not assigned L17 • 10min effort • 2 months ago • ? Code Smell • ? Critical

tests/Feature/Auth/PasswordUpdateTest.php

☐ Define a constant instead of duplicating this literal "/profile" 4 times. Adaptability design

Maintainability ?

Open Not assigned L20 • 10min effort • 2 months ago • ? Code Smell • ? Critical

tests/Feature/ProfileTest.php

☐ Define a constant instead of duplicating this literal "/profile" 9 times. Adaptability design

Maintainability ?

Open Not assigned L19 • 20min effort • 2 months ago • ? Code Smell • ? Critical

☐ Define a constant instead of duplicating this literal "Test User" 3 times. Adaptability design

Maintainability ?

Open Not assigned L31 • 8min effort • 2 months ago • ? Code Smell • ? Critical

16 of 16 shown

Embedded database should be used for evaluation purposes only
 The embedded database will not scale, it will not support migration to newer versions of SonarQube, and there is no support for migration your data out of it into a different database engine.

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMore

Tractor Braille a Texto / main

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

0.0% Security Hotspots Reviewed

To reviewAcknowledgedFixedSafe

3 Security Hotspots

Review priority: Low

Insecure Configuration1

Make sure this permissive CORS policy is safe here.

Others2

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

Where is the risk?What's the risk?Assess the riskHow can I fix it?Activity

config/cors.php

Open in IDE

```
12  // in web browsers. You are free to adjust these settings as needed.
13  //
14  // To learn more: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
15  //
16  //
17  'paths' => ['api/*', 'sanctum/csrf-cookie'],
18  //
19  'allowed_methods' => ['*'],
20  //
21  'allowed_origins' => ['*'],
22  //
23  'allowed_origins_patterns' => [],
24  //
25  'allowed_headers' => ['*'],
26  //
27  'exposed_headers' => [],
28  //
```

Make sure this permissive CORS policy is safe here.

Review priority: Low

Category: Insecure Configuration

Assignee: Not assigned

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMore

Tractor Braille a Texto / main

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

Project Overview

Security

Reliability

Maintainability

Security Review

Coverage

Duplications

Size

Complexity

Issues

Tractor 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%

Technical Debt (min)	Coverage (%)	Size (Lines of Code)
0	0.0	Large
2	10.0	Small
5	10.0	Small
8	10.0	Small
10	10.0	Small
15	10.0	Medium
20	20.0	Large
25	10.0	Small