

ANALISIS SONAR SCANNER

Enlace de repositorio: <https://github.com/training-practice-sofkau/taller-pruebas-junit>

Usuario de Github: Julián Lasso

Contexto: ¿Qué es SonnarQube?

Es una herramienta para realizar análisis estático de código, revisión automática de código para detectar bugs, vulnerabilidades y código apestoso (code smells).

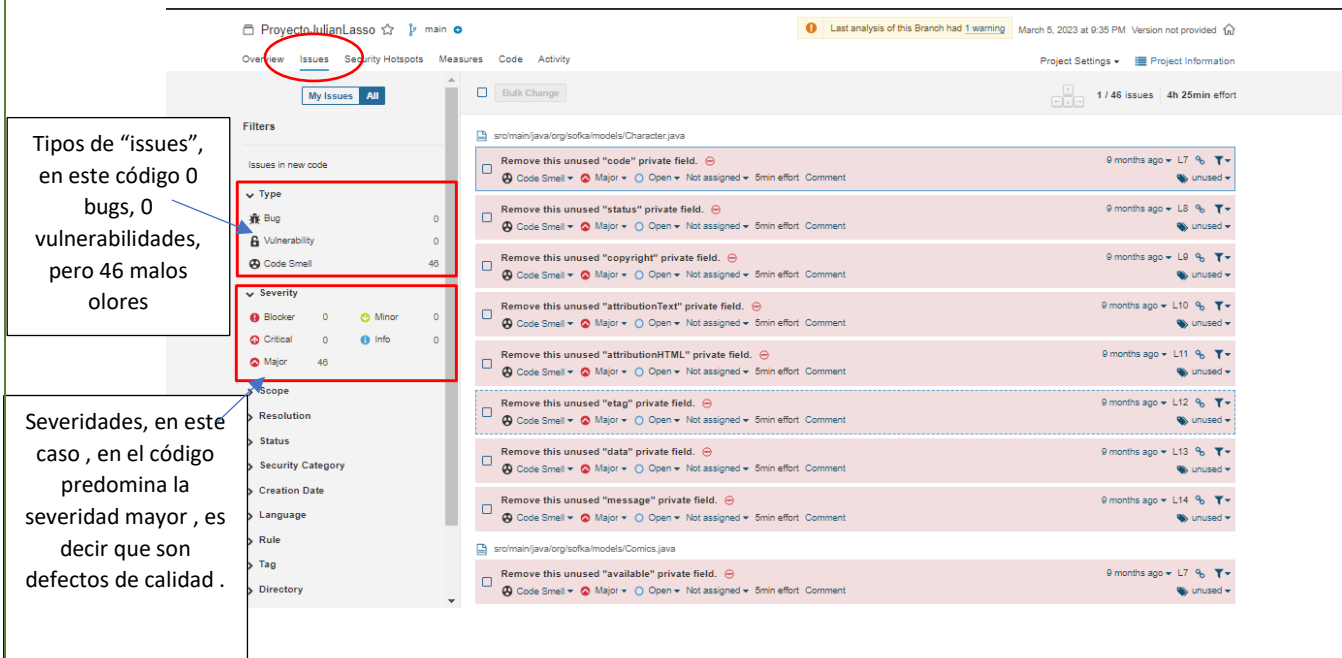
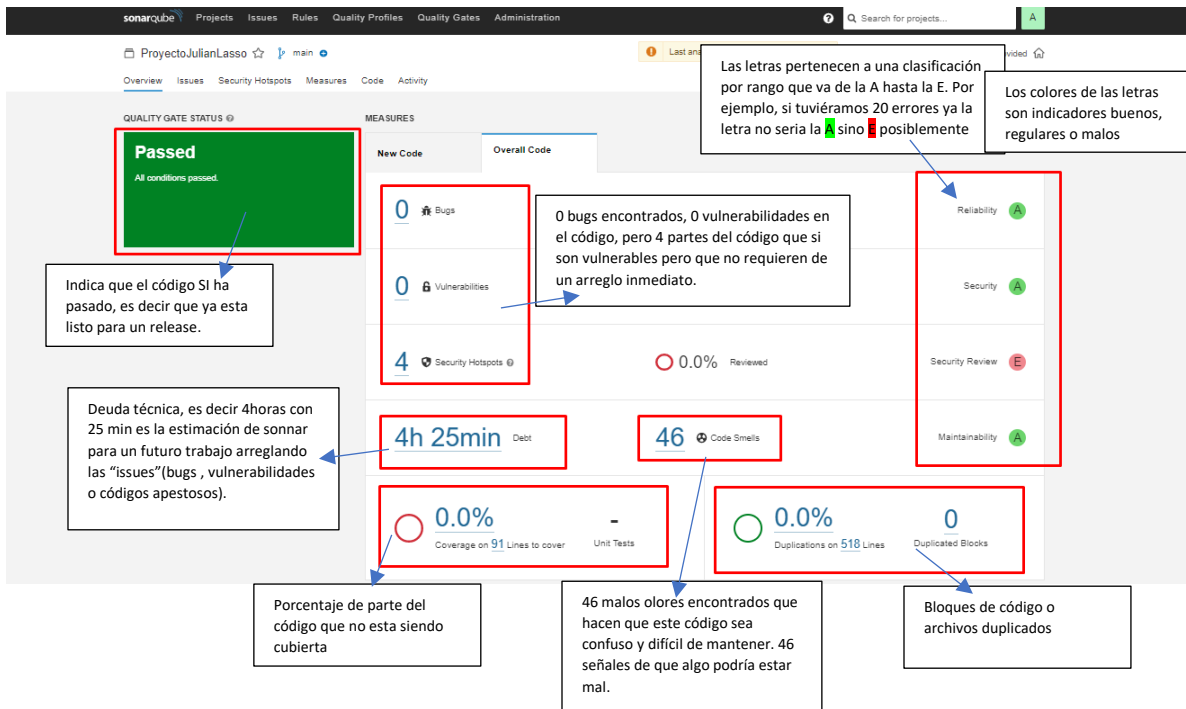
Análisis realizado:

Métricas que observaremos en el análisis:

- Bug: un error que romperá el código y necesita ser arreglado de inmediato
- Vulnerability: parte del código que hace que el software sea vulnerable a ataques y que tambien debe ser arreglado de inmediato
- Security Hotspot: parte del código vulnerable a ataques pero que, al no comprometer la seguridad de todo el software, no necesita un arreglo inmediato
- Code smells: no es un error, pero hace que el código sea confuso y difícil de mantener. Se pueden incluir aquí: patrones de código que son difíciles de entender o mantener, código duplicado, métodos o funciones demasiado largos o complejos, nombres de variables o funciones confusos o poco descriptivos, y otras características
- Cobertura: parte del código que no está cubierta, es decir, que no esta siendo probada por test unitarios. Sonarqube no detecta esto, pero es capaz de leer los reportes de herramientas que si lo hacen ej: Jacoco en java o Istanbul en JavaScript
- Duplicaciones: número de bloques de código o archivos duplicados. Tener estas duplicaciones dificulta la mantenibilidad y lectura del código
- Deuda técnica: es un concepto que refleja el futuro trabajo adicional que va a llevar algo al poder elegir una solución rápida pero limitada, en vez de una costosa

pero mejor. Sonarqube hace una estimación sobre el tiempo que se va a tardar en arreglar todas las “issues” que se han detectado en el análisis.

- Quality gate: Conjunto de métricas que tiene cumplir el código para ser considerado “listo para un reléase”. Sonarqube tiene su propio quality gates que se aplica por defecto, pero se pueden crear nuevos con cierto número de condiciones.
- Las severidades que podemos encontrar son: bloqueador (alta probabilidad de afectar el comportamiento de nuestra aplicación en producción), critico (afecta el comportamiento o es un problema que representa una falla de seguridad), mayor (defecto de calidad ejemplo bloques duplicados o parámetros no utilizados), menor (defecto de calidad) y de información (hace referencia a un hallazgo).
- Quality profile: conjunto de reglas y configuraciones que se aplican a un proyecto de software con el fin de evaluar la calidad del código y detectar posibles problemas. Puede incluir diferentes tipos de reglas, como reglas de seguridad, reglas de mantenibilidad, reglas de rendimiento, etc. Los desarrolladores pueden personalizar el Quality Profile para adaptarlo a las necesidades específicas de su proyecto y ajustar el nivel de severidad de las reglas según sus objetivos
- Rules: Estándares o prácticas de código que se deben seguir y determinan la severidad de los posibles bugs, vulnerabilidades, etc. Estas reglas vienen dadas por plugins asociados a cada lenguaje, que a su vez proporciona “quality profiles” por defecto.
Se ejecutan en el código fuente para generar los reportes



sonarqube Projects Issues **Rules** Quality Profiles Quality Gates Administration

Search for rules... 1 / 2,879 rules

Filters

Language

Search for languages...

Language	Count
Java	828
C#	405
JavaScript	312
TypeScript	307
PHP	257
Python	230
VB.NET	179
Go	114
Flex	82
HTML	65
Terraform	50
Ruby	48
Scala	47
Go	44
XML	36

15 shown [Show More](#)

Type

Bug 611

Vulnerability 167

Code Smell 1.7k

Reglas

- "important" should not be used on "keyframes"
- "\$this" should not be used in a static context
- "&&" and "||" should be used
- ".equals()" should not be used to test the values of "Atomic" classes
- "<!DOCTYPE>" declarations should appear before "<html>" tags
- "<>" should not be used to test inequality
- "<?php" and "<?=" tags should be used
- "<fieldset>" tags should contain a "<legend>"
- "<frames>" should have a "title" attribute
- "<html>" element should have a language attribute
- "" and "<div>" item tags should be in "", "" or "<div>" container tags
- "<object>" tags should provide an alternative content
- "" and "" tags should be used
- "<table>" tags should have a description
- "<th>" tags should have "id" or "scope" attributes

CSS Bug

PHP Bug

PHP Code Smell suspicious

Java Bug multi-threading

HTML Bug user-experience

Python Code Smell obsolete

PHP Code Smell convention, part 1

HTML Bug accessibility

HTML Bug accessibility

HTML Bug accessibility, wcag2-a

HTML Code Smell accessibility, wcag2-a

DEPRECATED HTML Bug

HTML Bug accessibility, wcag2-a

HTML Bug accessibility, wcag2-a

sonarqube Projects Issues **Quality Profiles** Quality Gates Administration

Quality Profiles

Quality profiles are collections of rules to apply during an analysis. For each language there is a default profile. All projects not explicitly assigned to some other profile will be analyzed with the default. Ideally, all projects will use the same profile for a language. [Learn More](#)

Filter profiles by: Select...

Conjunto de reglas que se aplican por defecto

Language	Profile(s)	Projects	Rules	Updated	Used
C#, 1 profile(s)	Sonar way BUILT-IN	DEFAULT	260	4 days ago	Never
CSS, 1 profile(s)	Sonar way BUILT-IN	DEFAULT	23	4 days ago	3 days ago
CloudFormation, 1 profile(s)	Sonar way BUILT-IN	DEFAULT	26	4 days ago	Never
Docker, 1 profile(s)	Sonar way BUILT-IN	DEFAULT	7	4 days ago	Never
Flex, 1 profile(s)	Sonar way BUILT-IN	DEFAULT	47	4 days ago	Never

Se pueden crear si se es administrador, acorde a los objetivos que se tengan

Create Restore

Recently Added Rules

- Failed unit tests should be OK, not yet activated
- Skipped unit tests should be OK, not yet activated
- Lines should have sufficient coverage by tests OK, not yet activated
- Source files should have OK, not yet activated
- Source files should not have any duplicated blocks OK, not yet activated
- Branches should have sufficient coverage by tests OK, not yet activated
- Branches should have sufficient coverage by tests Ruby, not yet activated
- Lines should have sufficient coverage by tests Ruby, not yet activated
- Source files should not have any duplicated blocks Ruby, not yet activated
- Skipped unit tests should be either removed or fixed Ruby, not yet activated

[See all 2.9k](#)

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

Search for projects...

A

Quality Gates ⓘ

Create

Sonar wayDEFAULTBUILT-IN

Sonar wayBUILT-IN

Copy

This quality gate complies with Clean as You Code

This quality gate complies with the [Clean as You Code](#) methodology, so that you benefit from the most efficient approach to delivering Clean Code. It ensures that:

✓

- No new bugs are introduced
- No new vulnerabilities are introduced
- All new security hotspots are reviewed
- New code has limited technical debt
- New code has limited duplication
- New code is properly covered by tests

Conditions ⓘ

Sonarqube tiene estas condiciones por defecto

Conditions on New Code

Metric	Operator	Value
Coverage	is less than	80.0%
Duplicated Lines (%)	is greater than	3.0%
Maintainability Rating	is worse than	A (Technical debt ratio is less than 5.0%)
Reliability Rating	is worse than	A (No bugs)
Security Hotspots Reviewed	is less than	100%
Security Rating	is worse than	A (No vulnerabilities)

Projects ⓘ

Every project not specifically associated to a quality gate will be associated to this one by default.

Se pueden crear, siendo el caso, las condiciones que se requieran

Condiciones predefinidas por Sonnar.