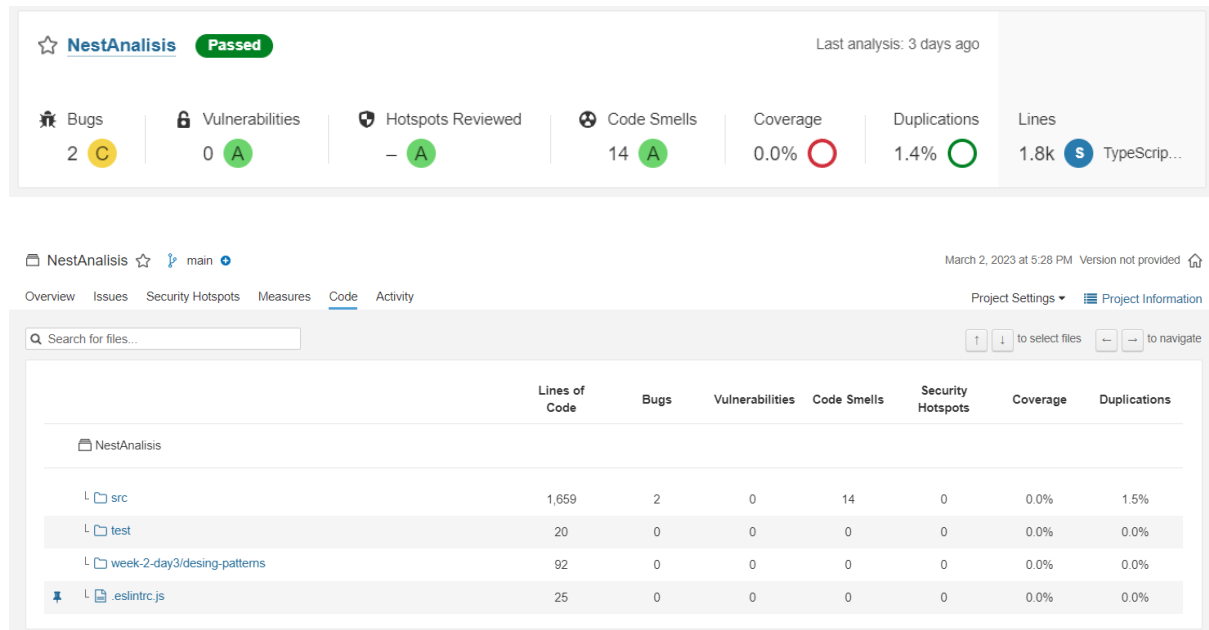


Reporte General análisis de código estático del repositorio

<https://github.com/codeJ30/C1-2023-Nest-backend-Colombia.git>

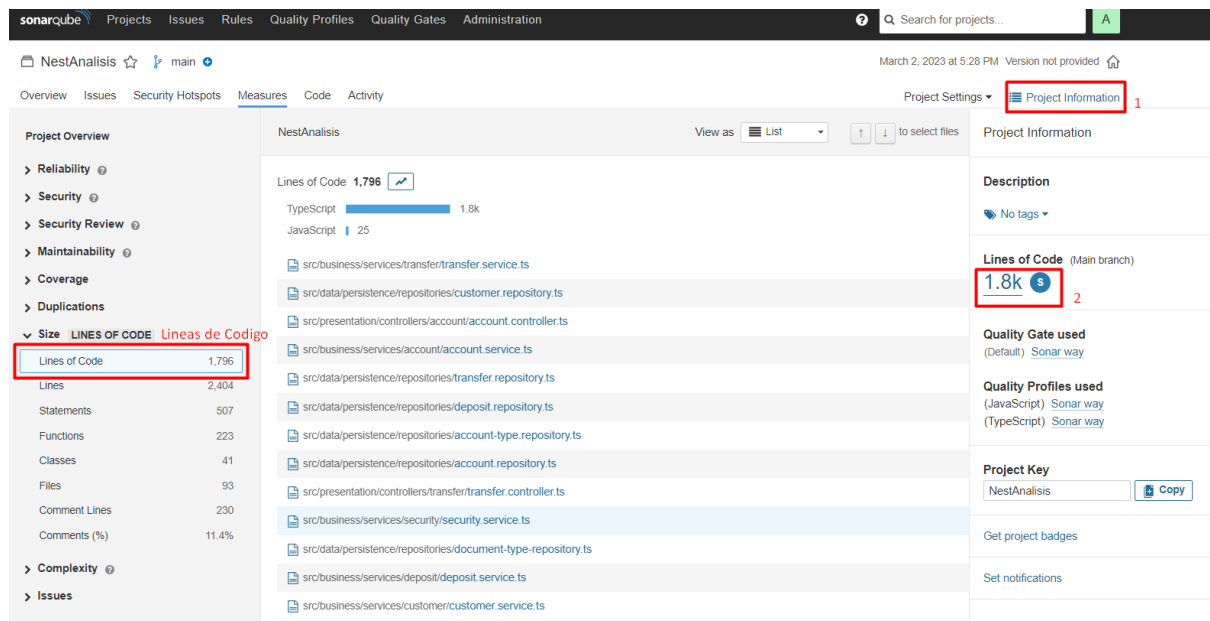
con SonarQube



¿Qué resultados se obtuvieron durante en el análisis?

A continuación se explica cada uno de los puntos obtenidos por el análisis:

Si accedemos a “Measures”, luego pulsamos sobre 'Project Information' y pulsamos sobre 'Lines of Code' del menú anterior, veremos dicho conteo segregado por Lenguajes:



Project Overview

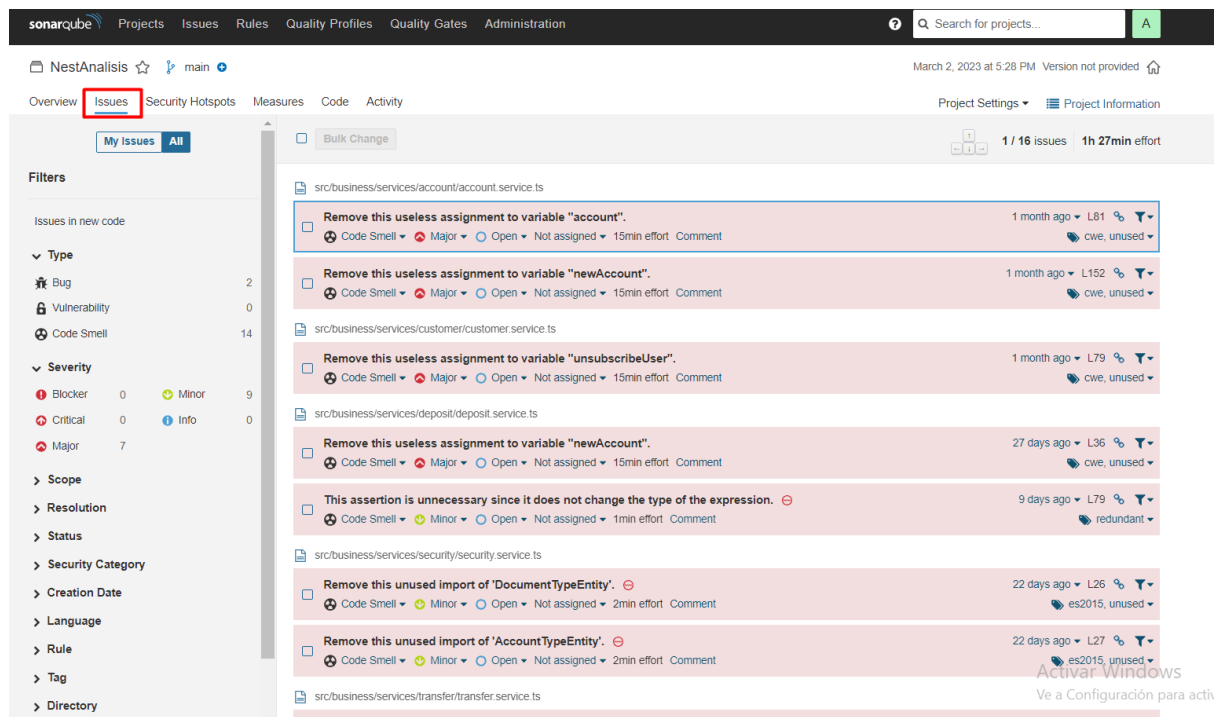
- Reliability
- Security
- Security Review
- Maintainability
- Coverage
- Duplications
- Size** **LINES OF CODE** **Líneas de Código**
 - Lines of Code** 1,796
 - Lines 2,404
 - Statements 507
 - Functions 223
 - Classes 41
 - Files 93
 - Comment Lines 230
 - Comments (%) 11.4%
- Complexity
- Issues

Project Information

- Description: No tags
- Lines of Code (Main branch): 1.8k
- Quality Gate used (Default): Sonar way
- Quality Profiles used (JavaScript): Sonar way (TypeScript): Sonar way
- Project Key: NestAnalysis

Este proyecto tiene 1796 líneas de código.

Issues



Issues

My Issues All

Filters

- Issues in new code
- Type
 - Bug 2
 - Vulnerability 0
 - Code Smell 14
- Severity
 - Blocker 0
 - Critical 0
 - Major 7
 - Minor 9
 - Info 0
- Scope
- Resolution
- Status
- Security Category
- Creation Date
- Language
- Rule
- Tag
- Directory




1 / 16 issues 1h 27min effort

- src/business/services/account/account.service.ts
 - Remove this useless assignment to variable "account". 1 month ago L81 Code Smell Major Open Not assigned 15min effort Comment cwe, unused
 - Remove this useless assignment to variable "newAccount". 1 month ago L152 Code Smell Major Open Not assigned 15min effort Comment cwe, unused
- src/business/services/customer/customer.service.ts
 - Remove this useless assignment to variable "unsubscribeUser". 1 month ago L79 Code Smell Major Open Not assigned 15min effort Comment cwe, unused
- src/business/services/deposit/deposit.service.ts
 - Remove this useless assignment to variable "newAccount". 27 days ago L36 Code Smell Major Open Not assigned 15min effort Comment cwe, unused
 - This assertion is unnecessary since it does not change the type of the expression. 9 days ago L79 Code Smell Minor Open Not assigned 1min effort Comment redundant
- src/business/services/security/security.service.ts
 - Remove this unused import of 'DocumentTypeEntity'. 22 days ago L26 Code Smell Minor Open Not assigned 2min effort Comment es2015, unused
 - Remove this unused import of 'AccountTypeEntity'. 22 days ago L27 Code Smell Minor Open Not assigned 2min effort Comment es2015, unused
- src/business/services/transfer/transfer.service.ts

issue: se atribuye a la unidad de trabajo para realizar una mejora.

El issue, es clasificado en **tipos** por SonarQube como:

- **(0) Vulnerability:** Indica que no se ha descubierto un problema que afecta la seguridad de la aplicación y que debe solucionarse de inmediato .
- **(2) Bug:** 2 errores.
- **Security Hotspot:** Se resalta un fragmento de código sensible a la seguridad.
- **(14) Code Smell:** Código que dificulta el mantenimiento (huele mal).

▼ Type	
 Bug	2
 Vulnerability	0
 Code Smell	14

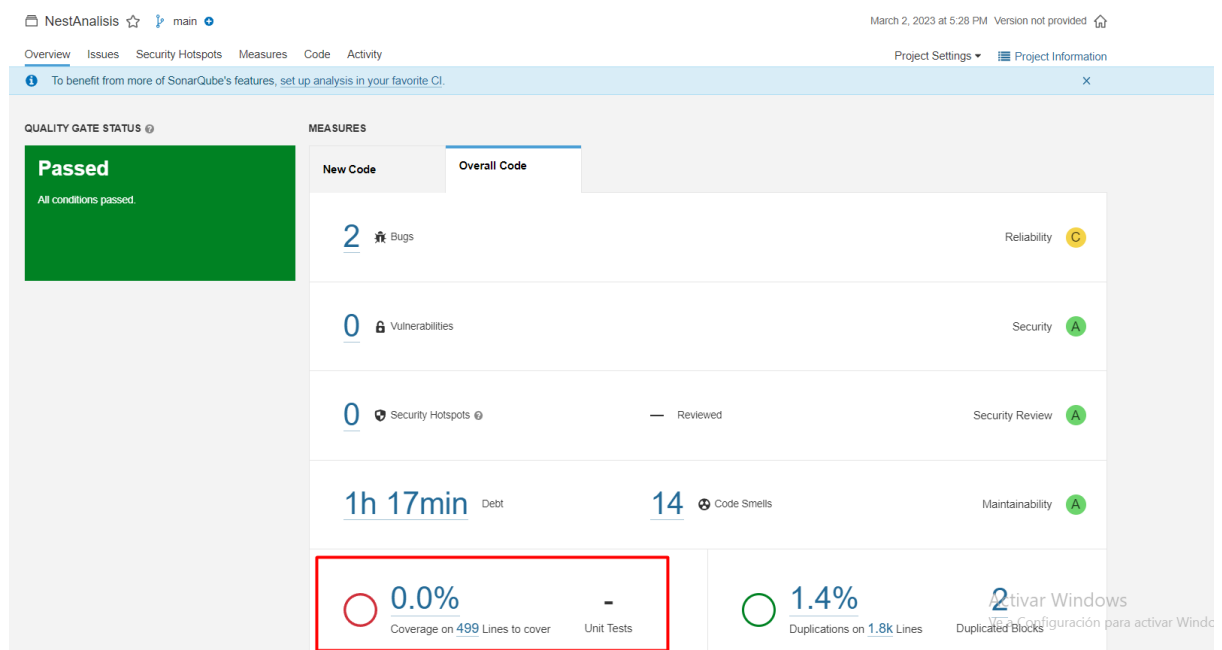
Finalmente, son segregados según la **severidad**:

- **(1) Blocker:** Error con una alta probabilidad de afectar el comportamiento de la aplicación en producción.
- **(0) Critical:** Urge su corrección aunque no suponga una afectación al comportamiento de la aplicación en producción o porque es una falla de seguridad de impacto alto.
- **(7) Major:** Defecto de calidad que puede afectar enormemente a la productividad del desarrollador.
- **(9) Minor:** Defecto de calidad que puede afectar ligeramente la productividad del desarrollador.
- **(0) Info:** Ni un error ni un defecto de calidad, solo un hallazgo o una mejora.

Severity			
! Blocker	0	Minor	9
↑ Critical	0	Info	0
Major	7		

Cobertura

En el Dashboard principal y en el submenú 'Overview' tendremos detalle sobre la cobertura:



Si pulsamos sobre la métrica nos da información detallada y los ficheros afectados:

Project Overview

- Reliability
- Security
- Security Review
- Maintainability
- Coverage UNCOVERED LINES**
 - Overview
 - Overall
 - Coverage: 0.0%
 - Lines to Cover: 499
 - Uncovered Lines: 499
 - Line Coverage: 0.0%
 - Duplications
 - Size
 - Complexity
 - Issues

Measures

NestAnalysis

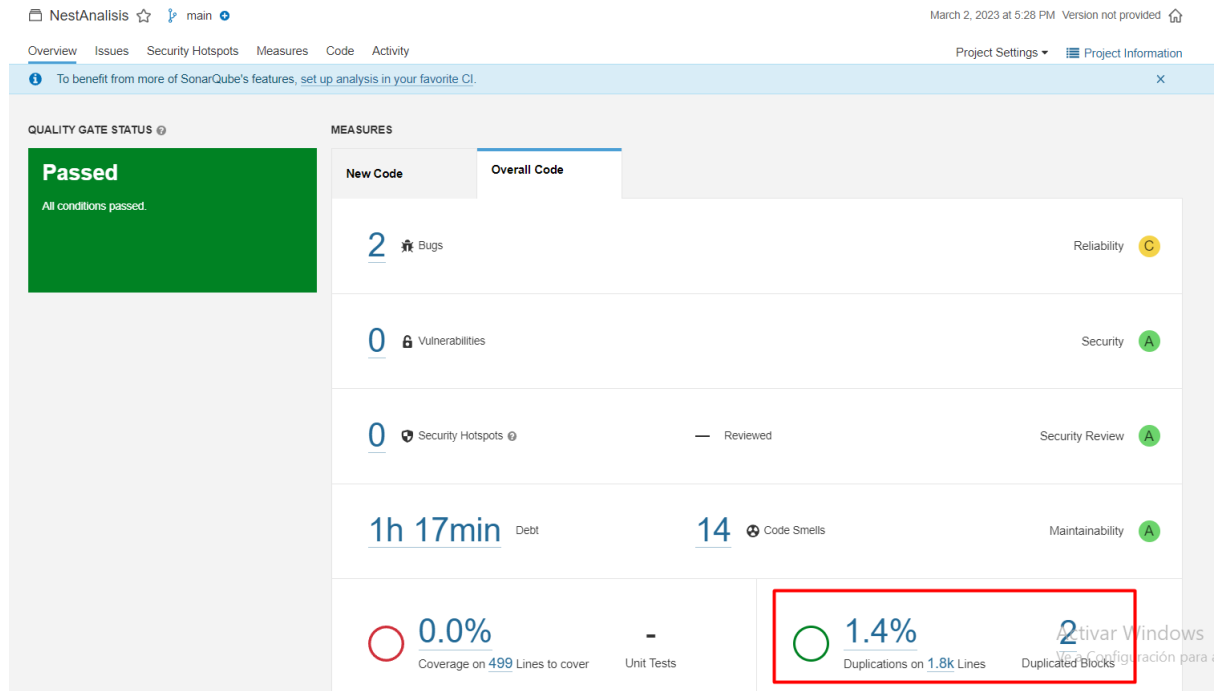
View as: List

File	Uncovered Lines	Line Coverage
src/business/services/transfer/transfer.service.ts	65	0.0%
src/business/services/account/account.service.ts	31	0.0%
src/data/persistence/repositories/customer.repository.ts	30	0.0%
src/data/persistence/repositories/deposit.repository.ts	30	0.0%
src/data/persistence/repositories/transfer.repository.ts	29	0.0%
src/data/persistence/repositories/account.repository.ts	28	0.0%
src/business/services/security/security.service.ts	28	0.0%
src/presentation/controllers/transfer/transfer.controller.ts	26	0.0%
src/business/services/customer/customer.service.ts	25	0.0%
src/data/persistence/repositories/account-type.repository.ts	23	0.0%
src/business/services/deposit/deposit.service.ts	21	0.0%
src/data/persistence/repositories/document-type.repository.ts	21	0.0%
src/presentation/controllers/account/account.controller.ts	13	0.0%
week-2-day3/desing-patterns/observer-pattern/index.ts	12	0.0%
test/app.e2e-spec.ts	8	0.0%
week-2-day3/desing-patterns/singleton-pattern/index.ts	8	0.0%
src/presentation/controllers/account/account.controller.spec.ts	7	0.0%

Porcentaje de código duplicado

Es una de las métricas que mejor indica qué grado de **malas prácticas** está instaurado en el proyecto.

SonarQube realiza este análisis con el objetivo de encontrar **líneas, bloques** o **ficheros** duplicados en el código fuente de los proyectos.



Overview	
Overall	
Density	1.4%
Duplicated Lines	33
Duplicated Blocks	2
Duplicated Files	2

Complejidad

El resultado obtenido en el cálculo de la complejidad ciclomática define el número de caminos independientes dentro de un fragmento de código y determina la cota superior del número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez.

The screenshot shows the NestAnalysis tool interface. On the left, the 'Project Overview' sidebar is expanded to 'Complexity', showing 'Cyclomatic Complexity' at 303 and 'Cognitive Complexity' at 111. The main panel displays a list of files with their respective complexity scores. The top of the main panel shows 'Cyclomatic Complexity 303' with a small chart icon. The file list includes various source files with their complexity scores on the right.

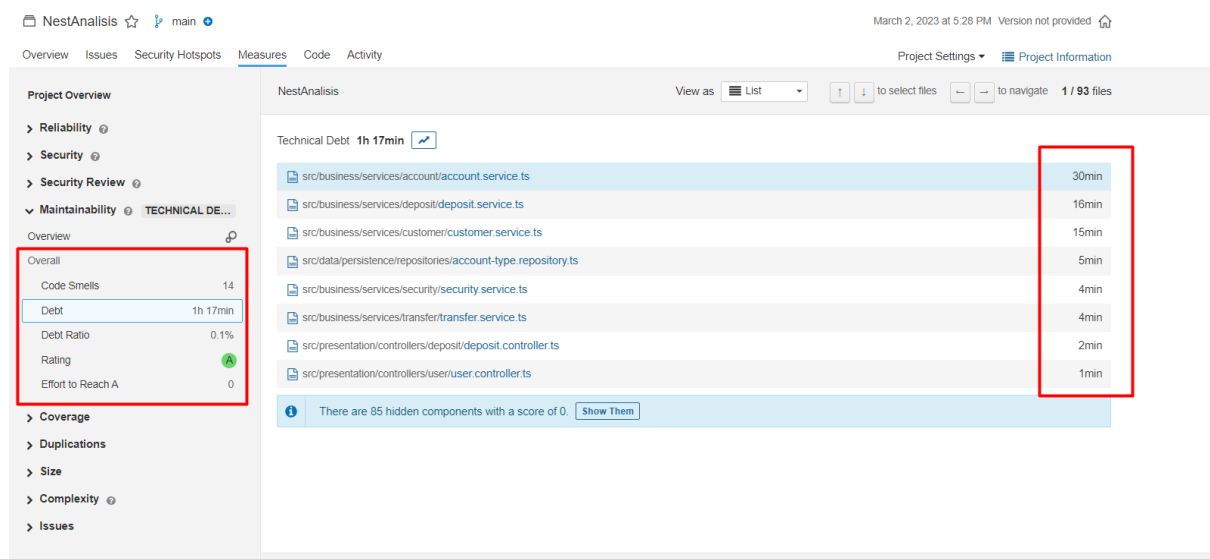
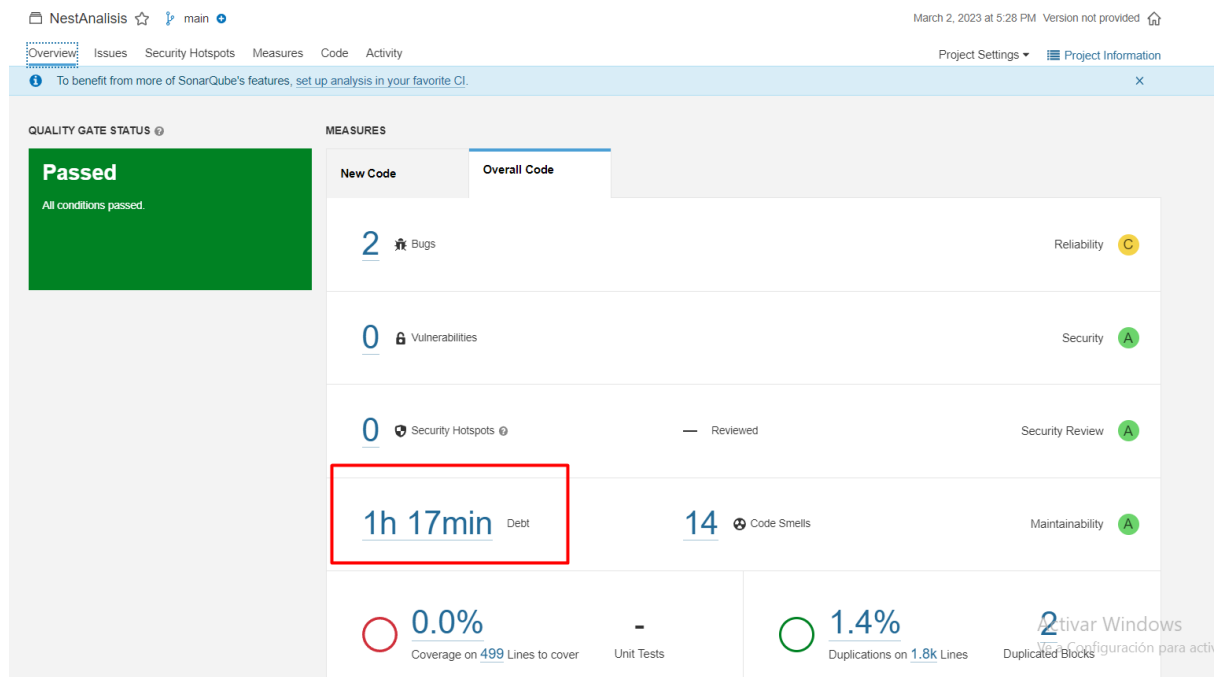
File Path	Complexity Score
src/data/persistence/repositories/customer.repository.ts	38
src/data/persistence/repositories/transfer.repository.ts	31
src/data/persistence/repositories/account.repository.ts	29
src/data/persistence/repositories/deposit.repository.ts	25
src/business/services/transfer/transfer.service.ts	20
src/business/services/account/account.service.ts	18
src/data/persistence/repositories/document-type.repository.ts	17
src/data/persistence/repositories/account-type.repository.ts	16
src/presentation/controllers/account/account.controller.ts	14
src/business/services/customer/customer.service.ts	8
src/business/services/security/security.service.ts	8
src/business/services/deposit/deposit.service.ts	7
src/presentation/controllers/transfer/transfer.controller.ts	7
src/presentation/controllers/user/user.controller.ts	6
src/presentation/controllers/deposit/deposit.controller.ts	5
week-2-day3/desing-patterns/observer-pattern/index.ts	5
week-2-day3/desing-patterns/factory-pattern/factory.ts	4
src/presentation/controllers/security/security.controller.ts	4

Existen los siguientes **Niveles de Complejidad** con un Riesgo asociado directamente al esfuerzo necesario para el mantenimiento:

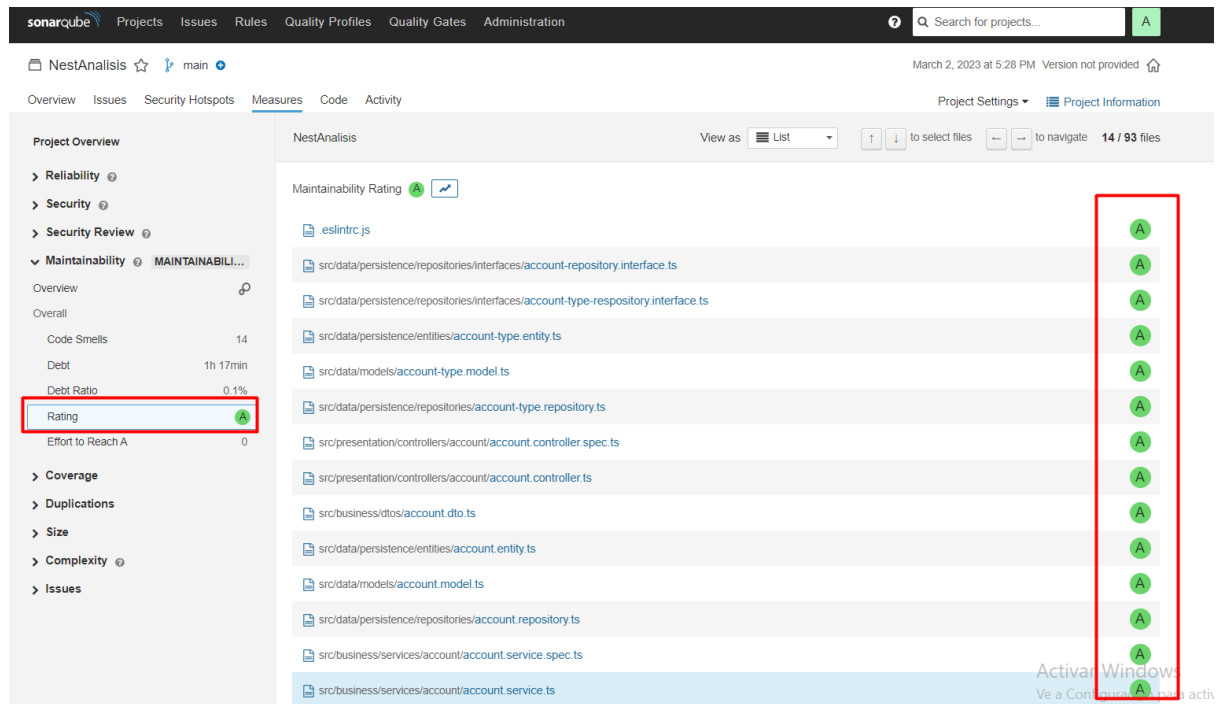
- 1 - 10: Sin riesgo
- 11 - 20: Riesgo Moderado
- 21 - 50: Riesgo Alto
- > 50: Es necesario/imprescindible refactorizar.

Deuda Técnica

Deuda técnica es el esfuerzo futuro que se debe realizar para arreglar defectos, vulnerabilidades y problemas de mantenimiento



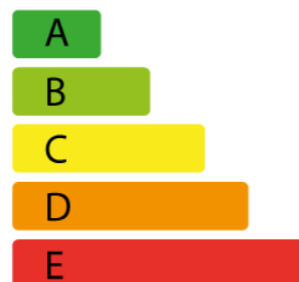
Otra métrica, el Rating de Deuda Técnica, lo que viene siendo la Calificación del código:



Rating --> SQALE

La calificación para una característica concreta y un artefacto concreto es la distancia que existe entre la calidad actual del proyecto vs la calidad deseada al final de desarrollo, o dicho de otra forma, la diferencia entre coste estimado de corregirla Vs la estimación del coste de desarrollo de ese artefacto.

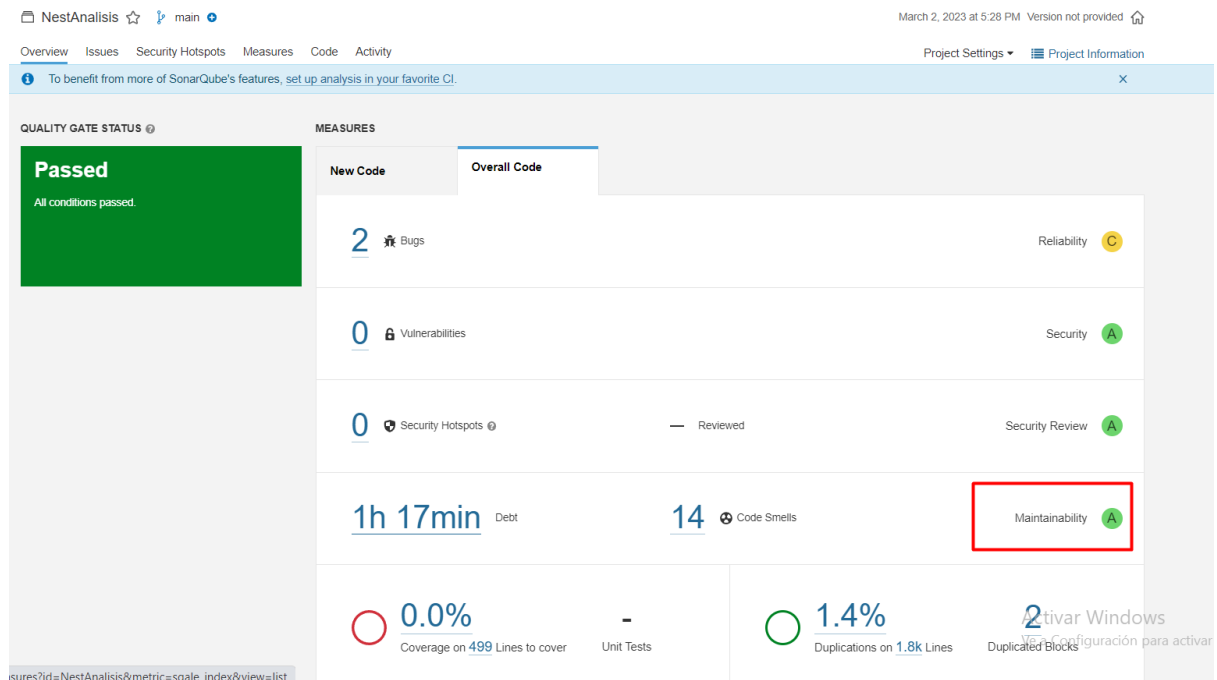
- A --> Calidad excelente
- B --> Calidad notable
- C --> Calidad insuficiente
- D --> Calidad baja
- E --> Calidad muy baja



Maintainability / Mantenibilidad: Capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

Se calcula de la siguiente manera:

- A: < 5%
- B: 6% - 10%
- C: 11% - 20%
- D: 21% - 50%
- E: > 50%



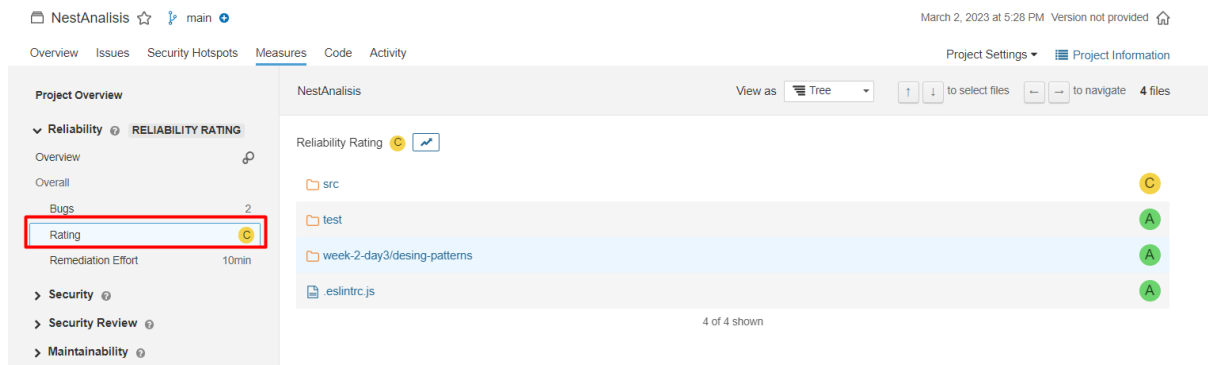
Reliability / Fiabilidad: Capacidad de un sistema o componente para desempeñar correctamente las funciones especificadas. Es directamente proporcional al coste de solucionar los issues que provocan que el sistema no sea confiable. Por ello, el rating se realiza en base a las Bugs encontrados.

Se calcula de la siguiente manera:

- A: Sin Bugs
- B: 1 Bug Minor

● C: 1 Bug Major

- D: 1 Bug Crítico
- E: 1 Bug Blocker

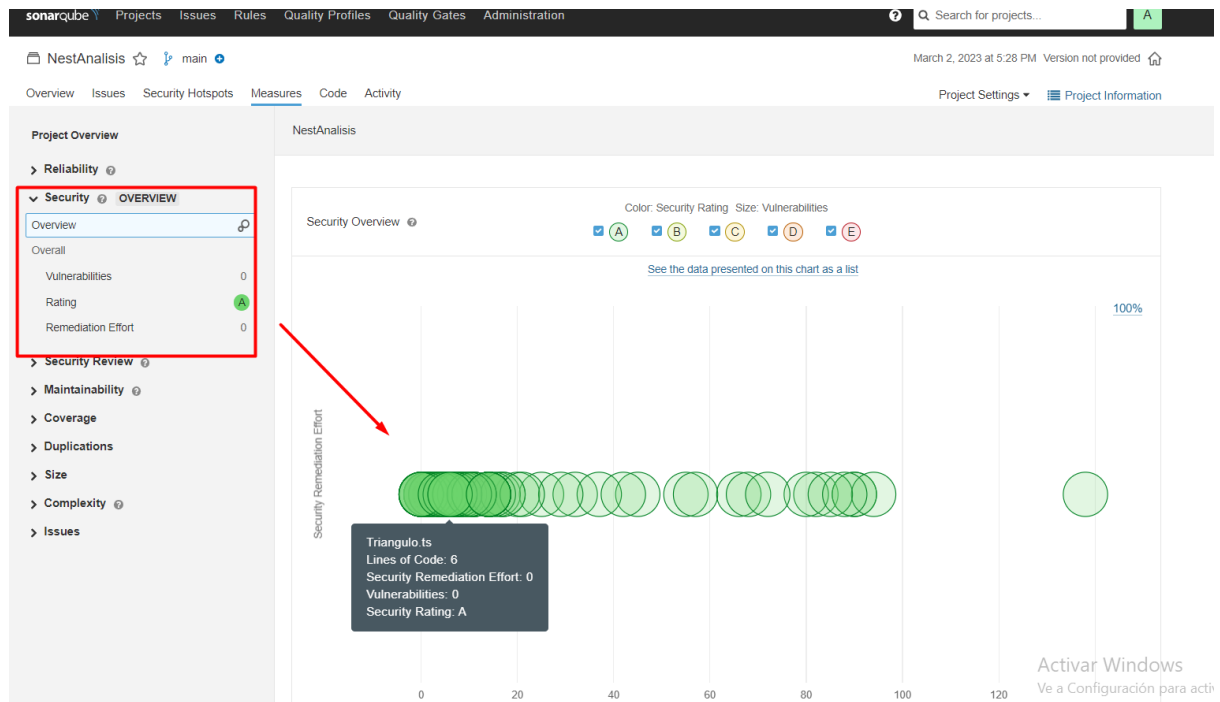


Security / Seguridad: Grado de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Directamente proporcional a la capacidad que tiene el sistema para reducir el número de los ataques. Por ello, el rating se realiza en base a las vulnerabilidades encontrados.

Se calcula de la siguiente manera:

● A: 0 Vulnerabilidades

- B: 1 Vulnerabilidad Minor
- C: 1 Vulnerabilidad Major
- D: 1 Vulnerabilidad Critical
- E: 1 Vulnerabilidad Blocker



Security Review / Revisión de Seguridad: El rating se realiza en base al número de hotspots revisados.

Se calcula de la siguiente manera:

● A: > 80%

● B: 70% - 80%

● C: 50% - 70%

● D: 30% - 50%

● E: < 30%

The screenshot shows the SonarQube interface for a project named 'NestAnalysis'. The left sidebar is expanded to 'Security Review', which is highlighted with a red box. The main area shows a tree view of files with their respective security hotspot counts:

File	Security Hotspots
src	0
test	0
week-2-day3/desing-patterns	0
eslinttr.js	0

Quality Gates

Es un conjunto predefinido de criterios de calidad de código definidos en SonarQube que un proyecto debe cumplir con el fin de pasar de una etapa de su ciclo de vida a la siguiente.

The screenshot shows the 'Quality Gates' configuration page in SonarQube. The 'BUILT-IN' quality gate is selected, and its conditions are listed in a table:

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)