

Informe Análisis de Seguridad del Código Fuente con SonarCloud – Actividad 2

Arnulfo Monroy Ortiz

Id 654265

Jose Murcia

Id 1052782

Corporación Universitaria Minuto de Dios

Edwin Albeiro Ramos Villamil

Desarrollo de Software Seguro

NRC 67848

Octubre de 2025

1. Marco Teórico

La calidad del software es un aspecto fundamental en el desarrollo de sistemas seguros y eficientes. Herramientas como SonarCloud permiten evaluar automáticamente aspectos críticos del código fuente, tales como la seguridad, mantenibilidad, duplicación, fiabilidad y cobertura. El análisis estático del código facilita la detección de errores, vulnerabilidades y malas prácticas de programación antes de que el software entre en producción. SonarCloud es una plataforma en la nube que realiza este tipo de análisis de manera automatizada y visualmente comprensible, lo que facilita la toma de decisiones en entornos DevSecOps.

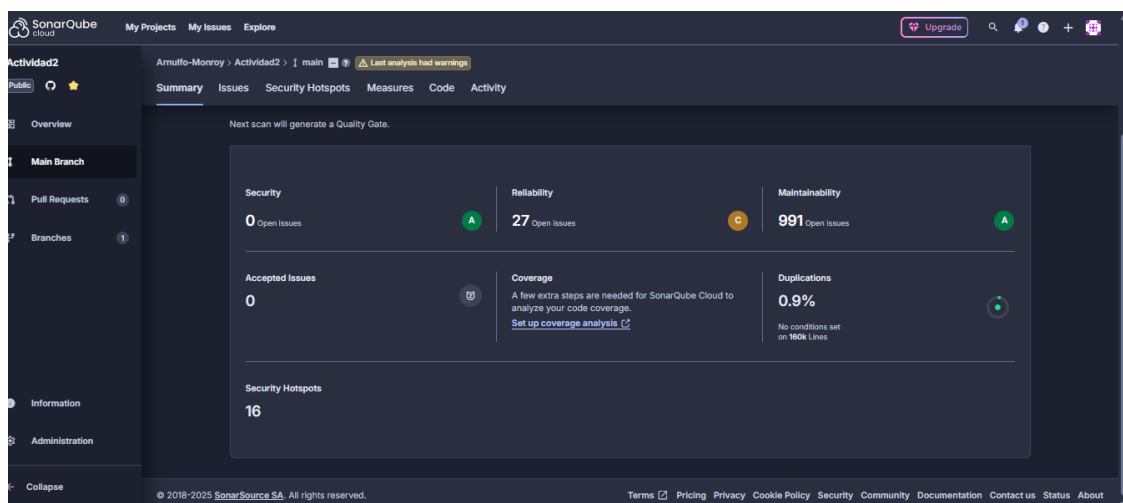
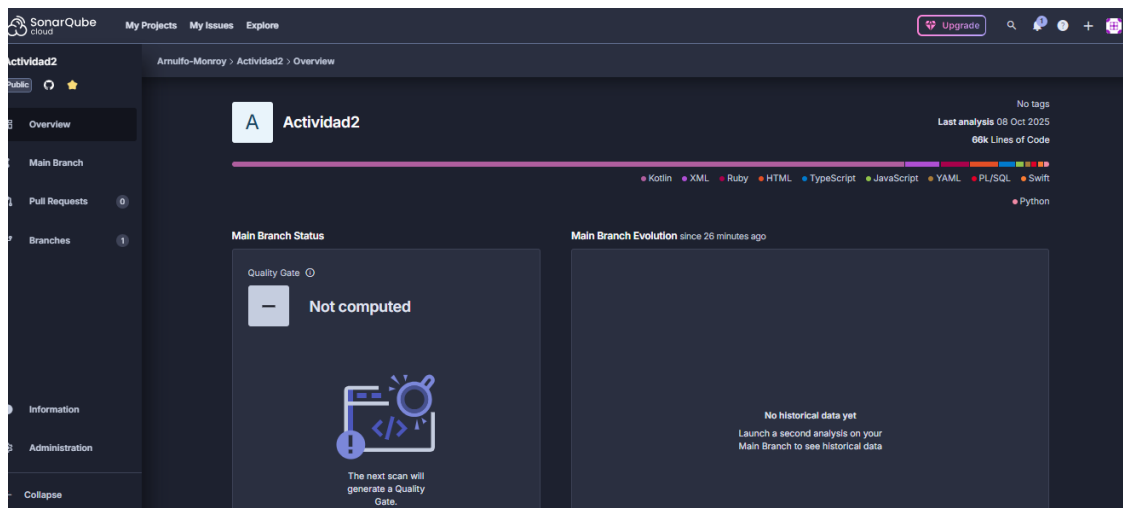
2. Introducción

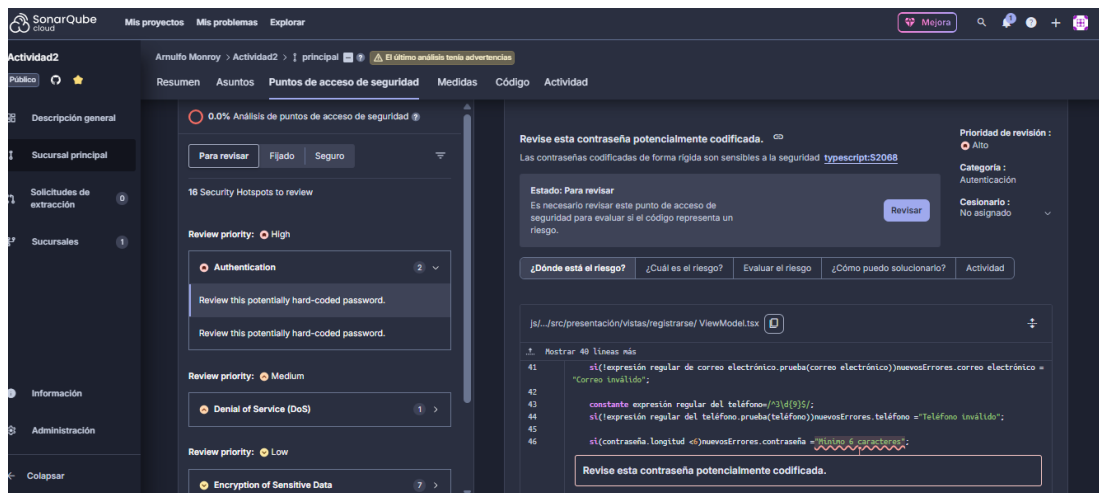
SonarCloud es una plataforma en línea de análisis estático de código que permite auditar proyectos en múltiples lenguajes de programación. Está diseñada para integrarse fácilmente con repositorios como GitHub, permitiendo identificar automáticamente vulnerabilidades, errores, duplicaciones y problemas de mantenibilidad. En el contexto de esta auditoría académica, SonarCloud fue seleccionada por su enfoque en calidad y seguridad de software, y por su accesibilidad como herramienta de evaluación gratuita para proyectos open source.

3. Evidencia del Análisis

Se ejecutó un análisis estático del código fuente, que corresponde a una revisión automatizada sin necesidad de ejecutar el sistema. Esta prueba permite detectar malas prácticas, errores potenciales, duplicación de código y posibles vulnerabilidades. No se realizaron pruebas dinámicas, de rendimiento ni de penetración debido al enfoque de esta actividad en la inspección de calidad y seguridad a través del análisis estático.

A continuación, se presentan los principales resultados detectados por SonarCloud:

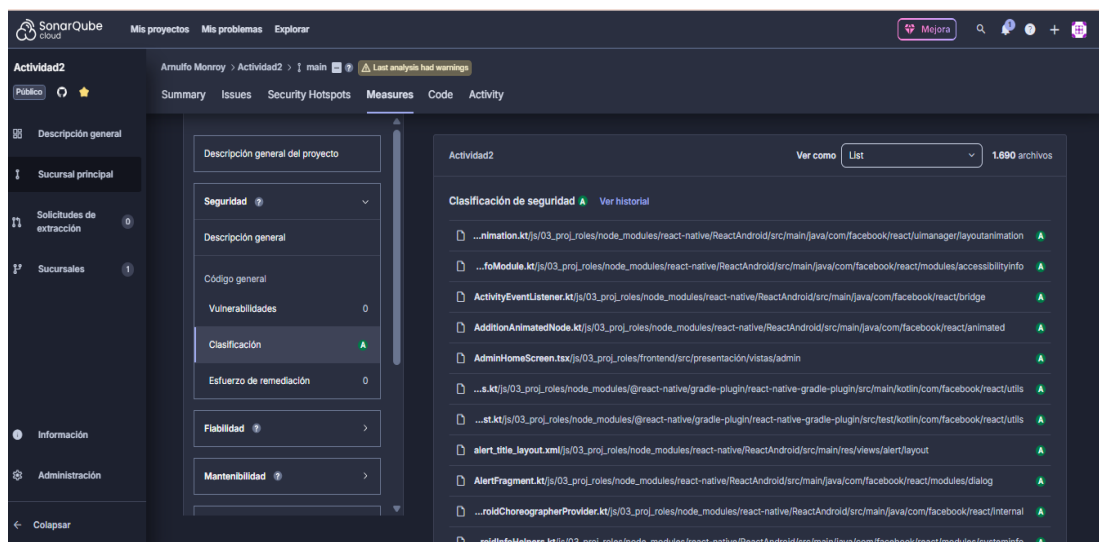




3.1 Seguridad

El análisis muestra 0 problemas de seguridad. Esto indica que el código no presenta vulnerabilidades críticas.

- Vulnerabilidades detectadas: 0
- Rating de seguridad: A



3.2 Fiabilidad

Se encontraron 27 errores que pueden comprometer la estabilidad del sistema. Es importante revisar estos puntos.

- Errores encontrados: 27
- Rating: C
- Tiempo estimado de remediación: 2h 26min

The screenshot shows the SonarQube Cloud interface for project 'Actividad2'. The 'Measures' tab is selected, displaying a table of code quality metrics for 1,690 files. The table includes columns for file names and their corresponding quality scores. The overall project rating is 'C' and the estimated remediation effort is '2h 26min'.

File Name	Quality Score
angular.html/js/03_proj_roles/node_modules/sprint-js/demo	C
API.html/js/03_proj_roles/backend/node_modules/bignumber.js/doc	A
Integration_test_runner.html/js/03_proj_roles/node_modules/@react-native/debugger-frontend/dist/third-partyfront_end	A
JsonUtilsTest.kt/js/03_proj_roles/node_modules/@react-native/gradle-plugin/shared/src/test/kotlin/com/facebook/react/utlis	A
props.html/js/03_proj_roles/node_modules/terser/tools	D
ReadableNativeArray.kt/js/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/bridge	D
UserEditScreen.tsx/js/03_proj_roles/frontend/src/presentación/Vistas/admin	D
UsersListScreen.tsx/js/03_proj_roles/frontend/src/presentación/Vistas/admin	D
...s.kt/js/03_proj_roles/node_modules/@react-native/gradle-plugin/react-native-gradle-plugin/src/main/kotlin/com/facebook/react/utlis	A
...ettingsExtension.kt/js/03_proj_roles/node_modules/@react-native/gradle-plugin/settings-plugin/src/main/kotlin/com/facebook/react	B

3.3 Mantenibilidad

El sistema contiene 991 'code smells', lo cual sugiere que hay muchas mejoras posibles en la estructura del código.

- Code Smells: 991
- Rating: A
- Deuda técnica estimada: 14 días

The screenshot shows the SonarQube Cloud interface for project 'Actividad2'. The 'Measures' tab is selected, displaying a table of code quality metrics for 1,690 files. The table includes columns for file names and their corresponding quality scores. The overall project rating is 'A' and the estimated remediation effort is '14d'.

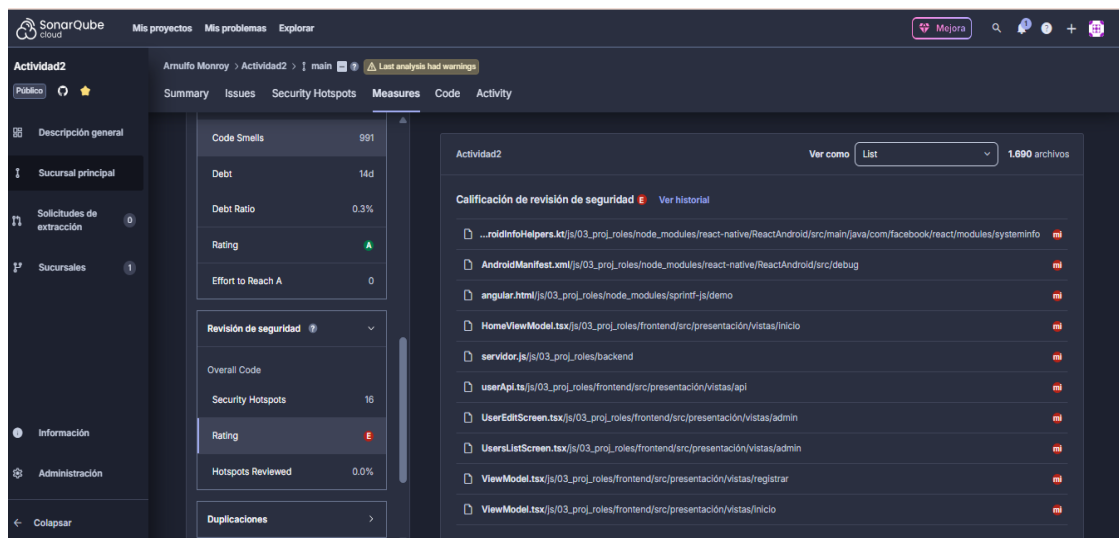
File Name	Quality Score
RCTEventEmitter.kt/js/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/uimanager/eventos	B
...oduleWithSpec.kt/js/03_roles_del_proyecto/módulos_de_nodo/react-native/ReactAndroid/src/main/java/com/facebook/react/bridge	B
TurboReactPackage.kt/js/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react	B
UIBlockViewResolver.kt/js/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/fabric/interop	B
UIManagerModuleListener.kt/js/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/uimanager	B

Hay 1.680 componentes ocultos con una puntuación de A. [Muéstralos](#)

3.4 Hotspots de Seguridad

Se detectaron 16 hotspots de seguridad. Estas son zonas críticas que deben ser evaluadas manualmente.

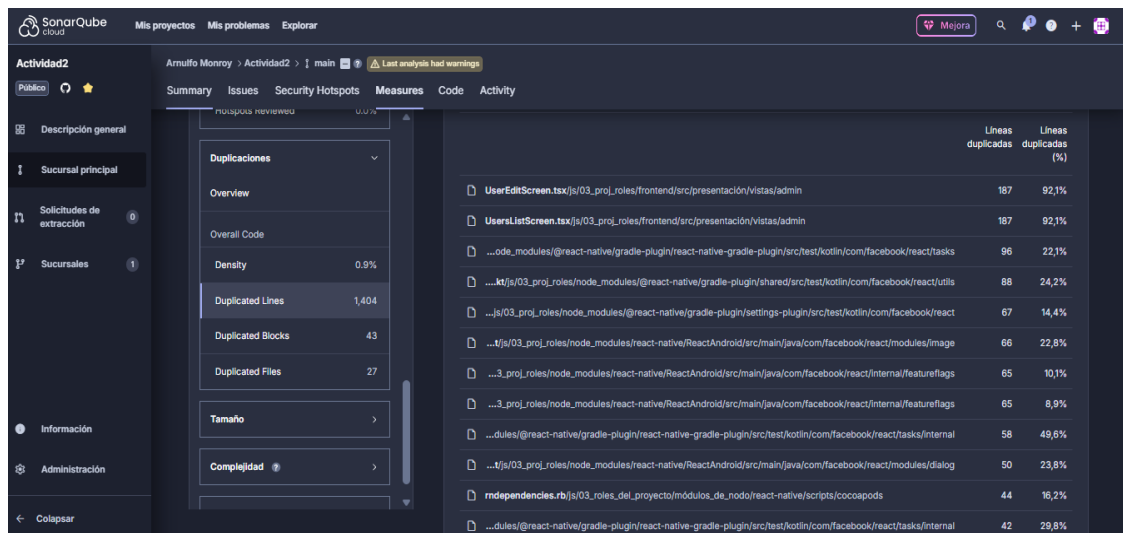
- Hotspots identificados: 16
- Rating general: E
- Requiere revisión manual detallada



3.5 Duplicación de Código

Se observó un 0.9% de duplicación. Aunque es un valor bajo, se recomienda mejorar este aspecto.

- Duplicación: 0.9%
- Líneas duplicadas: 1.404
- Archivos duplicados: 27

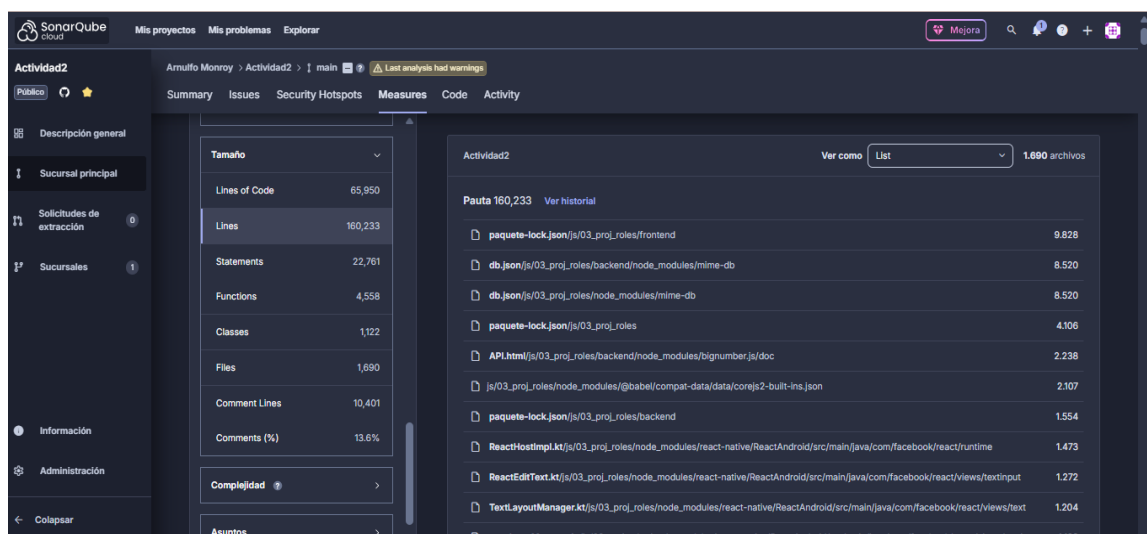


File Path	Lines	Duplicated Lines (%)
UserEditScreen.tsx/js/03_proj_roles/frontend/src/presentación/visas/admin	187	92,1%
UserListScreen.tsx/js/03_proj_roles/frontend/src/presentación/visas/admin	187	92,1%
...ode_modules/@react-native/gradle-plugin/react-native-gradle-plugin/src/test/kotlin/com/facebook/react/tasks	96	22,1%
...kt/js/03_proj_roles/node_modules/@react-native/gradle-plugin/shared/src/test/kotlin/com/facebook/react/utlis	88	24,2%
...js/03_proj_roles/node_modules/@react-native/gradle-plugin/settings-plugin/src/test/kotlin/com/facebook/react	67	14,4%
...Vjs/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/modules/image	66	22,8%
...3_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/internal/featureflags	65	10,1%
...3_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/internal/featureflags	65	8,9%
...dules/@react-native/gradle-plugin/react-native-gradle-plugin/src/test/kotlin/com/facebook/react/tasks/internal	58	49,6%
...Vjs/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/modules/dialog	50	23,8%
rndependencies.rb/js/03_roles_del_proyecto/módulos_de_nodo/react-native/scripts/cocoapods	44	16,2%
...dules/@react-native/gradle-plugin/react-native-gradle-plugin/src/test/kotlin/com/facebook/react/tasks/internal	42	29,8%

3.6 Líneas de Código

El proyecto contiene aproximadamente 160k líneas de código, lo cual es considerable y requiere herramientas automáticas para su control.

- Líneas totales: 160.233
- Líneas de código: 65.950
- Clases: 1.122
- Funciones: 4.558



Metric	Value	Percentage
Lines of Code	65,950	
Lines	160,233	
Statements	22,761	
Functions	4,558	
Classes	1,122	
Files	1,690	
Comment Lines	10,401	
Comments (%)	13,6%	

File Path	Lines
paquete-lock.json/js/03_proj_roles/frontend	9.828
db.json/js/03_proj_roles/backend/node_modules/mime-db	8.520
db.json/js/03_proj_roles/node_modules/mime-db	8.520
paquete-lock.json/js/03_proj_roles	4.306
API.html/js/03_proj_roles/backend/node_modules/bignumber.js/doc	2.238
js/03_proj_roles/node_modules/@babel/compat-data/data/corejs2-built-ins.json	2.107
paquete-lock.json/js/03_proj_roles/backend	1.554
ReactHostImpl.kt/js/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/runtime	1.473
ReactEditText.kt/js/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/views/textinput	1.272
TextLayoutManager.kt/js/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/views/text	1.204
TextLayoutManager.kt/js/03_proj_roles/node_modules/react-native/ReactAndroid/src/main/java/com/facebook/react/views/textinput	1.133

4. Evaluación de Seguridad del Proyecto

A partir de los resultados, se concluye que el proyecto no presenta vulnerabilidades graves, pero tiene oportunidades de mejora en cuanto a fiabilidad y mantenibilidad. Los hotspots encontrados deben ser evaluados manualmente por el equipo de desarrollo. El proyecto puede considerarse moderadamente seguro, siempre que se implementen mejoras en el corto plazo.

5. Reflexión sobre la Importancia de Auditar el Código

La auditoría automatizada del código fuente es esencial para garantizar la entrega de software seguro y confiable. Herramientas como SonarCloud permiten identificar problemas antes de que se conviertan en fallos en producción. En un mundo donde las aplicaciones están constantemente expuestas en entornos conectados, integrar auditoría como parte del ciclo de vida del software es una práctica crítica que fortalece la seguridad, reduce costos de mantenimiento y protege la privacidad de los usuarios finales.

6. Conclusiones

El análisis realizado con SonarCloud permitió identificar múltiples aspectos relevantes del código fuente. Aunque no se detectaron problemas de seguridad graves, sí se hallaron problemas de mantenibilidad y errores menores. El uso de herramientas de análisis estático como SonarCloud fortalece los procesos de desarrollo seguro y mejora la calidad del software entregado.

Referencias

- Andrews, G. R. (2000). *Foundations of multithreaded, parallel, and distributed programming*. Addison-Wesley.
- Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2011). *Distributed systems: concepts and design* (5th ed.). Pearson Education.
- Fox, A., & Brewer, E. A. (1999). Harvest, yield, and scalable tolerant systems. In *Proceedings of the ACM Symposium on Operating Systems Principles* (SOSP).
- Liu, Y., Jin, H., & Zhang, Y. (2015). Analysis of system reliability in distributed environments. *Journal of Computer and System Sciences*, 81*(2), 208–220.
- Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed systems: principles and paradigms*. Prentice Hall.
- SonarCloud. (2025). *Clean Code and Security Rating Documentation*. <https://docs.sonarcloud.io>