Diagnóstico de la Calidad del Software

Fabián Guillermo Salazar Sarmiento

Facultad de Ingeniería, Universidad UNIMINUTO

NRC: 40-61831: Desarrollo de Software Seguro

Edwin Albeiro Ramos Villamil

21 de febrero de 2026

1. Descripción del Proyecto de Desarrollo de Software

El proyecto seleccionado, "StockPro", es un sistema de información para la gestión de inventario, diseñado para optimizar el control de stock y los movimientos de productos en un pequeño comercio. La gestión del proyecto se realiza bajo la metodología ágil Scrum, priorizando la entrega de software funcional en ciclos cortos llamados Sprints.

Objetivo General

Desarrollar un prototipo funcional de una aplicación web para la gestión de inventario de un pequeño comercio, aplicando la metodología Scrum en todas las fases del ciclo de vida del desarrollo.

Requerimientos Funcionales (Historias de Usuario)

Las funcionalidades clave (Product Backlog) del sistema se centran en la gestión integral del inventario:

ID	Nombre de la HU	Prioridad	Criterio de Validación Clave
HU01	Gestión de Productos	Alta	Permitir crear, leer y eliminar productos (CRUD).
HU02	Registro de Entradas	Alta	Seleccionar un producto y añadir cantidad a su stock actual.
HU03	Registro de Salidas	Alta	Restar cantidad del stock actual y
HU04	Alertas de Stock Bajo	Media	Mostrar una alerta visual si el stock cae por debajo de un umbral mínimo.

HU05	Gestión de Proveedores	Media	Permitir crear, leer y eliminar proveedores.
HU06	Reporte de Inventario	Baja	Generar un reporte del estado actual en formato imprimible.

Actores Implicados (Stakeholders)

Los principales actores del proyecto son:

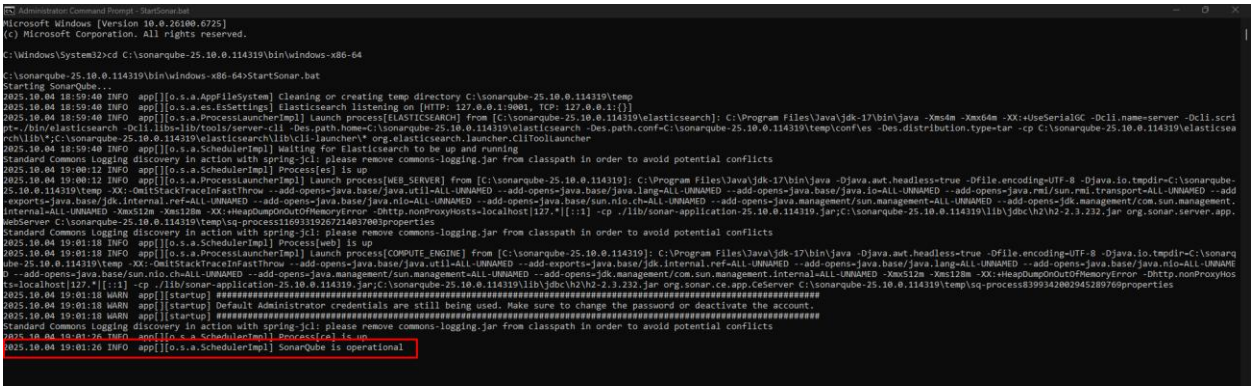
Actor	Rol en el Proyecto
Administrador	Gestiona productos, proveedores, configura umbrales de stock y genera reportes.
Operario	Registra los movimientos de inventario (entradas y salidas de stock).
Desarrollador	Ejecuta la metodología Scrum y construye el software (asume roles de

2. Instalación, Configuración y Ejecución de SonarQube

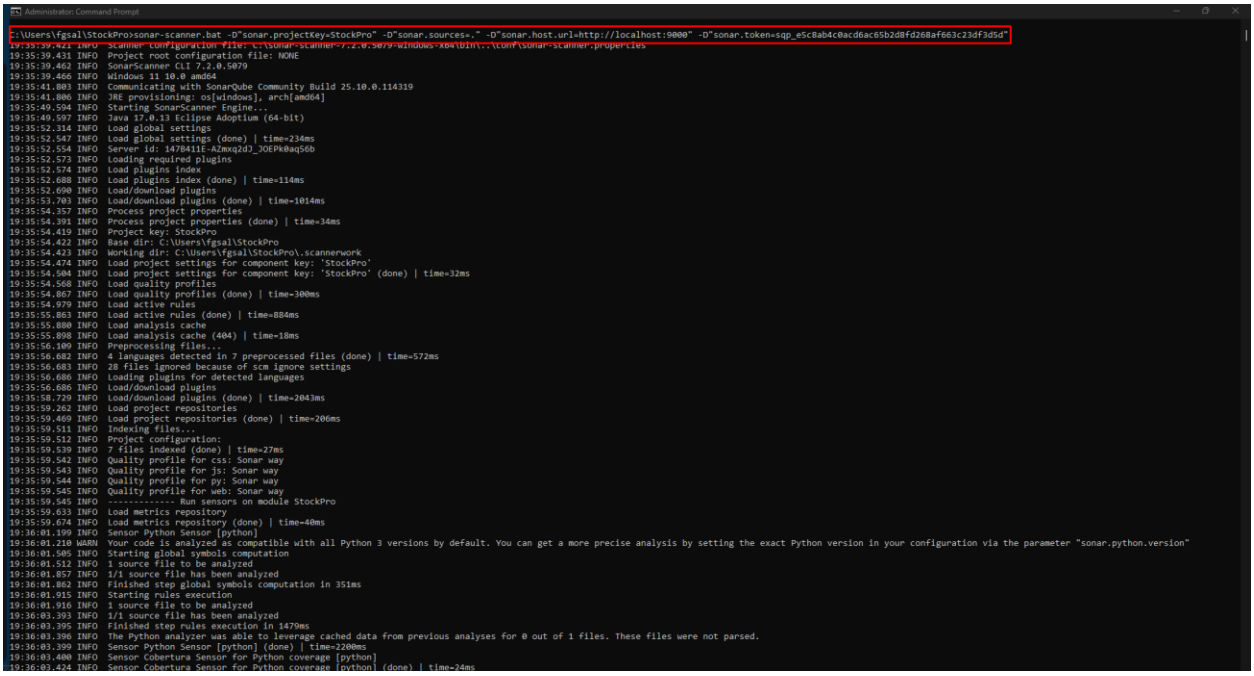
Se completó la instalación de SonarQube Community Edition y la configuración de SonarScanner en el entorno local. El aplicativo se configuró para analizar el código fuente de StockPro, que está desarrollado en Python (Flask) para el backend y HTML/CSS/JavaScript para el frontend.

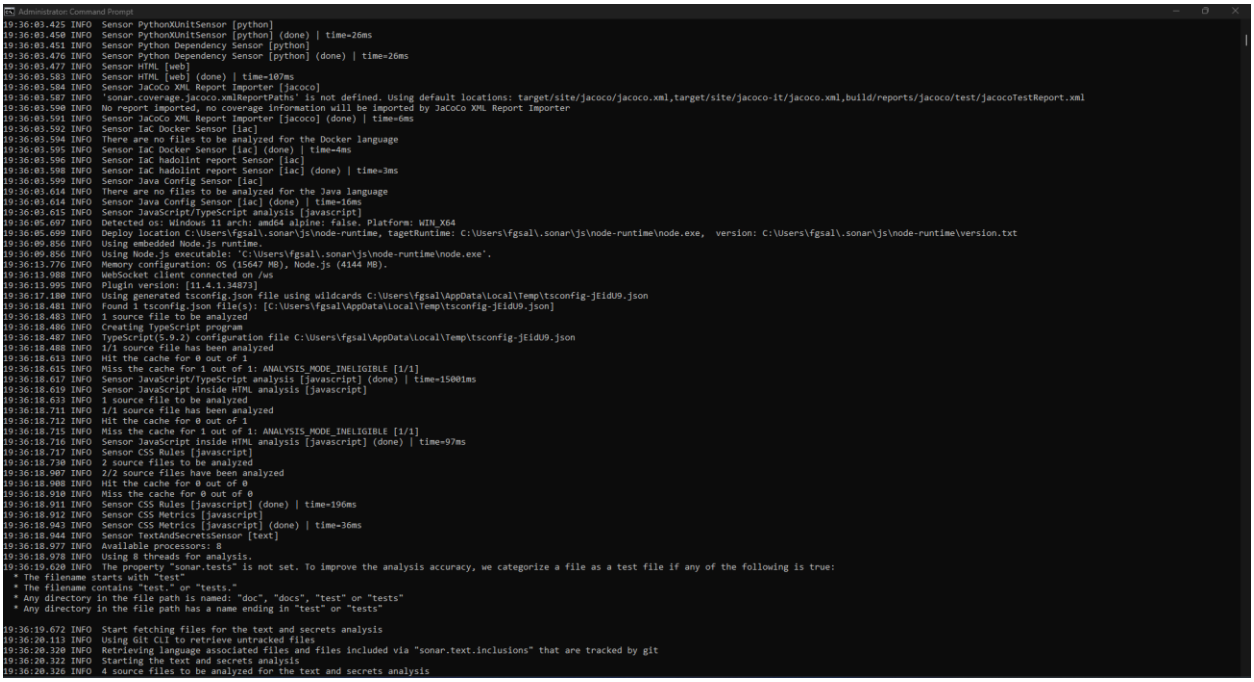
La ejecución del análisis se realizó mediante el comando del SonarScanner, integrando la herramienta en el proceso de calidad antes de la entrega del incremento de Sprint.

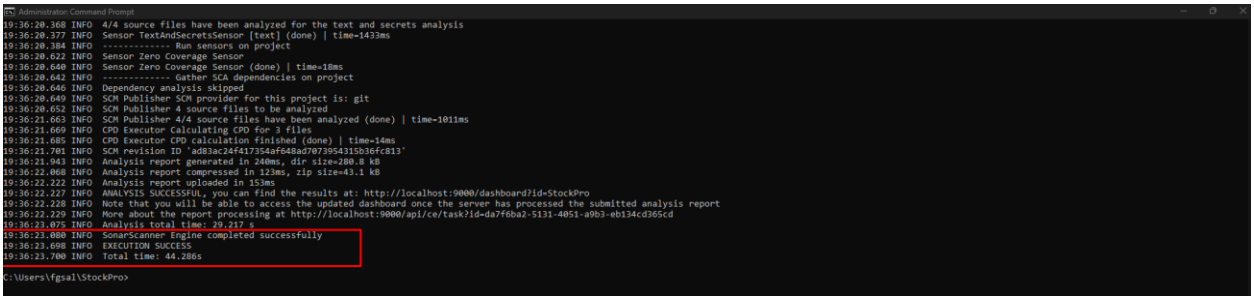
Inicio de SonarQube:



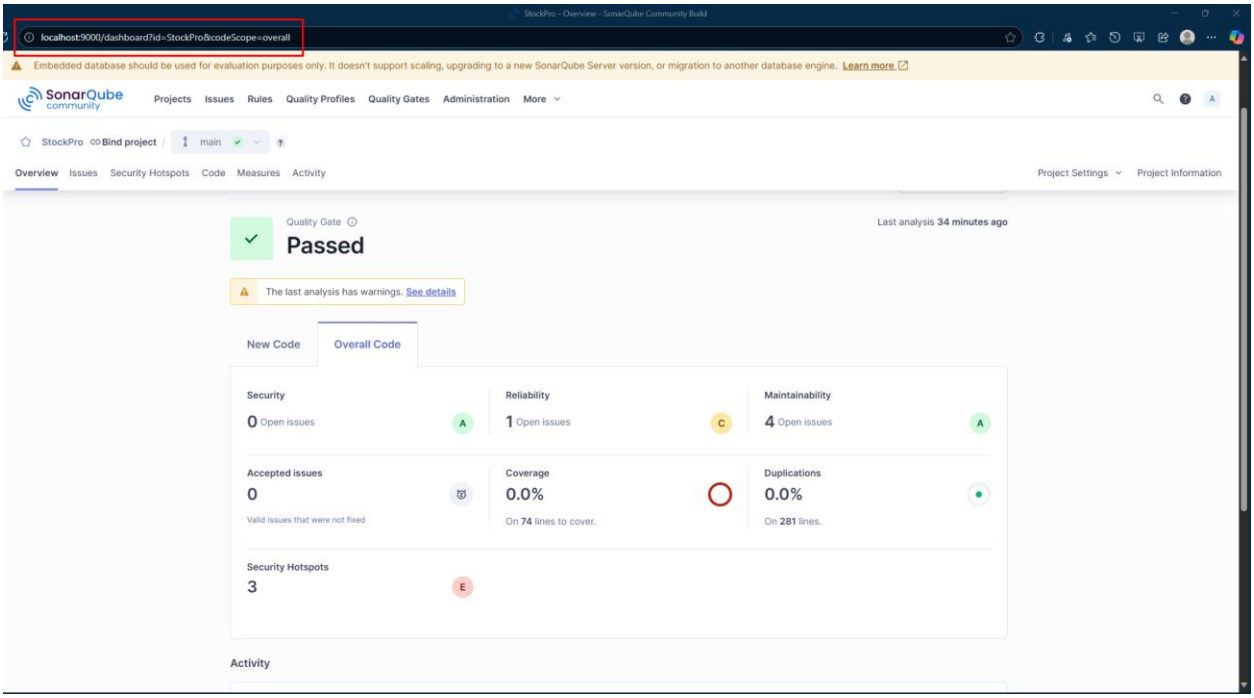
Ejecución de análisis con la plataforma:




Administrator: Command Prompt
19:36:03.425 INFO Sensor PythonXUnitSensor [python]
19:36:03.450 INFO Sensor PythonXUnitSensor [python] (done) | time=26ms
19:36:03.451 INFO Sensor Python Dependency Sensor [python]
19:36:03.476 INFO Sensor Python Dependency Sensor [python] (done) | time=26ms
19:36:03.477 INFO Sensor HTML [web]
19:36:03.583 INFO Sensor HTML [web] (done) | time=107ms
19:36:03.584 INFO Sensor JaCoCo XML Report Importer [jacoco]
19:36:03.587 INFO 'sonar.coverage.jacoco.xmlReportPaths' is not defined. Using default locations: target/site/jacoco/jacoco.xml,target/site/jacoco-it/jacoco.xml,build/reports/jacoco/test/jacocoTestReport.xml
19:36:03.590 INFO No report imported, no coverage information will be imported by JaCoCo XML Report Importer
19:36:03.591 INFO Sensor JaCoCo XML Report Importer [jacoco] (done) | time=6ms
19:36:03.592 INFO Sensor IaC Docker Sensor [iac]
19:36:03.594 INFO There are no files to be analyzed for the Docker language
19:36:03.595 INFO Sensor IaC Docker Sensor [iac] (done) | time=4ms
19:36:03.596 INFO Sensor IaC hadolint report Sensor [iac]
19:36:03.598 INFO Sensor IaC hadolint report Sensor [iac] (done) | time=3ms
19:36:03.599 INFO Sensor Java Config Sensor [iac]
19:36:03.614 INFO There are no files to be analyzed for the Java language
19:36:03.614 INFO Sensor Java Config Sensor [iac] (done) | time=16ms
19:36:03.615 INFO Sensor JavaScript/TypeScript analysis [javascript]
19:36:05.697 INFO Detected os: Windows 11 arch: amd64 alpine: false. Platform: WIN_X64
19:36:05.699 INFO Deploy location C:\Users\fgsal\.sonar\js\node-runtime, tagetRuntime: C:\Users\fgsal\.sonar\js\node-runtime\node.exe, version: C:\Users\fgsal\.sonar\js\node-runtime\version.txt
19:36:09.856 INFO Using embedded Node.js runtime.
19:36:09.856 INFO Using Node.js executable: 'C:\Users\fgsal\.sonar\js\node-runtime\node.exe'.
19:36:13.776 INFO Memory configuration: OS (15647 MB), Node.js (4144 MB).
19:36:13.988 INFO WebSocket client connected on /ws
19:36:13.995 INFO Plugin version: [11.4.1.34873]
19:36:17.180 INFO Using generated tsconfig.json file using wildcards C:\Users\fgsal\AppData\Local\Temp\tsconfig-jEidU9.json
19:36:18.481 INFO Found 1 tsconfig.json file(s): [C:\Users\fgsal\AppData\Local\Temp\tsconfig-jEidU9.json]
19:36:18.483 INFO 1 source file to be analyzed
19:36:18.486 INFO Creating TypeScript program
19:36:18.487 INFO TypeScript(5.9.2) configuration file C:\Users\fgsal\AppData\Local\Temp\tsconfig-jEidU9.json
19:36:18.488 INFO 1/1 source file has been analyzed
19:36:18.613 INFO Hit the cache for 0 out of 1
19:36:18.615 INFO Miss the cache for 1 out of 1: ANALYSIS_MODE_INELIGIBLE [1/1]
19:36:18.617 INFO Sensor JavaScript/TypeScript analysis [javascript] (done) | time=15001ms
19:36:18.619 INFO Sensor JavaScript inside HTML analysis [javascript]
19:36:18.633 INFO 1 source file to be analyzed
19:36:18.711 INFO 1/1 source file has been analyzed
19:36:18.712 INFO Hit the cache for 0 out of 1
19:36:18.715 INFO Miss the cache for 1 out of 1: ANALYSIS_MODE_INELIGIBLE [1/1]
19:36:18.716 INFO Sensor JavaScript inside HTML analysis [javascript] (done) | time=97ms
19:36:18.717 INFO Sensor CSS Rules [javascript]
19:36:18.730 INFO 2 source files to be analyzed
19:36:18.907 INFO 2/2 source files have been analyzed
19:36:18.908 INFO Hit the cache for 0 out of 0
19:36:18.910 INFO Miss the cache for 0 out of 0
19:36:18.911 INFO Sensor CSS Rules [javascript] (done) | time=196ms
19:36:18.912 INFO Sensor CSS Metrics [javascript]
19:36:18.943 INFO Sensor CSS Metrics [javascript] (done) | time=36ms
19:36:18.944 INFO Sensor TextAndSecretsSensor [text]
19:36:18.977 INFO Available processors: 8
19:36:18.978 INFO Using 8 threads for analysis.
19:36:19.620 INFO The property "sonar.tests" is not set. To improve the analysis accuracy, we categorize a file as a test file if any of the following is true:
  * The filename starts with "test"
  * The filename contains "test." or "tests."
  * Any directory in the file path is named: "doc", "docs", "test" or "tests"
  * Any directory in the file path has a name ending in "test" or "tests"
19:36:19.672 INFO Start fetching files for the text and secrets analysis
19:36:20.113 INFO Using Git CLI to retrieve untracked files
19:36:20.320 INFO Retrieving language associated files and files included via "sonar.text.inclusions" that are tracked by git
19:36:20.322 INFO Starting the text and secrets analysis
19:36:20.326 INFO 4 source files to be analyzed for the text and secrets analysis


Administrator: Command Prompt
19:36:20.368 INFO 4/4 source files have been analyzed for the text and secrets analysis
19:36:20.377 INFO Sensor TextAndSecretsSensor [text] (done) | time=1433ms
19:36:20.384 INFO ------------ Run sensors on project
19:36:20.622 INFO Sensor Zero Coverage Sensor
19:36:20.640 INFO Sensor Zero Coverage Sensor (done) | time=18ms
19:36:20.642 INFO ------------ Gather SCA dependencies on project
19:36:20.646 INFO Dependency analysis skipped
19:36:20.649 INFO SCM Publisher SCM provider for this project is: git
19:36:20.652 INFO SCM Publisher 4 source files to be analyzed
19:36:21.663 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=1011ms
19:36:21.669 INFO CPD Executor Calculating CPD for 3 files
19:36:21.685 INFO CPD Executor CPD calculation finished (done) | time=14ms
19:36:21.701 INFO SCM revision ID 'ad83ac24f417354af648ad707395431Sb36fc813'
19:36:21.943 INFO Analysis report generated in 240ms, dir size=280.8 kB
19:36:22.068 INFO Analysis report compressed in 123ms, zip size=43.1 kB
19:36:22.222 INFO Analysis report uploaded in 153ms
19:36:22.227 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=StockPro
19:36:22.228 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
19:36:22.229 INFO More about the report processing at http://localhost:9000/api/ce/task?id=da7f6ba2-5131-4051-a9b3-eb134cd365cd
19:36:23.075 INFO Analysis total time: 29.217 s
19:36:23.080 INFO SonarScanner Engine completed successfully
19:36:23.698 INFO EXECUTION SUCCESS
19:36:23.700 INFO Total time: 44.286s
C:\Users\fgsal\StockPro>

Visión del resumen del análisis en la consola web de SonarQube:



3. Análisis de Resultados y Áreas Problemáticas (Bajo Normas ISO)

El análisis del código fuente del Sprint 1 revela áreas de riesgo que impactan las características de Calidad del Producto de la norma ISO 25000 (SQuaRE), a pesar de que la funcionalidad inicial (HU01) es operativa.

A. Problemas de Fiabilidad y Consistencia (ISO 25000: Fiabilidad y Mantenibilidad)

- Issues Abiertos (1 Bug/Medium):
    - Área Problemática Específica: Uso de la función global parseInt() en el frontend (script.js:L56).
    - Implicación ISO: Afecta la Mantenibilidad (Consistencia y Organización) al no usar el estándar moderno (Number.parseInt()), lo que puede llevar a inconsistencias y errores de tipo en la manipulación de datos.

B. Problemas de Mantenibilidad (ISO 25000: Mantenibilidad)

Se reportan Code Smells que incrementan la Deuda Técnica, afectando la Facilidad de Análisis y Modificación (ISO 25000).

ID	Issue Específico	Ubicación	Implicación en la Mantenibilidad
CS01	app.py: La función delete_producto no maneja adecuadamente las excepciones de base de datos.	Backend	Riesgo de fallos no controlados (errores 500) que detienen la aplicación. Afecta la Facilidad de Análisis y el Manejo de Errores del sistema.
CS02	script.js: La función cargarProductos es demasiado larga, con múltiples responsabilidades (obtener datos, limpiar DOM, construir filas, adjuntar manejadores de eventos).	Frontend	Violan el principio de Responsabilidad Única. Afecta la Facilidad de Modificación, pues un cambio en la lógica de datos requiere revisar la lógica de la UI.
CS03	script.js: eliminarProducto no maneja correctamente las excepciones de red (catch vacío o incompleto).	Frontend	Afecta la Facilidad de Pruebas y el diagnóstico de errores. El usuario no sería notificado si la operación falla.
CS04	script.js: No se manejan excepciones específicas en cargarProductos.	Frontend	Similar a CS03, afecta la robustez de la aplicación y la experiencia

			del usuario si la conexión a la API falla.

C. Problemas de Seguridad (ISO 25000: Seguridad)

Se detectan Security Hotspots que requieren revisión manual para garantizar la Seguridad (Integridad de Datos) del sistema.

ID	Issue Específico	Ubicación	Implicación en la Seguridad
SH01	app.py: La función get_productos usa una consulta SQL que puede ser vulnerable a inyección de código si la consulta se construyera dinámicamente con entradas externas (riesgo de inyección SQL).	Backend	Aunque la consulta es estática, la regla advierte sobre el uso inseguro de sqlite3.connect en el contexto de una función web.
SH02	app.py: La función add_producto utiliza request.get_json() sin validar ni sanear las entradas, lo que es un potencial vector para ataques.	Backend	Riesgo de Inyección SQL si data contuviera código malicioso, comprometiendo la Integridad de la base de datos (clave para el inventario).
SH03	app.py: La función delete_producto tiene el mismo problema de conexión y	Backend	El endpoint que maneja la eliminación es una función crítica y debe estar protegida

	manejo de peticiones de usuario sin validación estricta.		contra manipulaciones externas no deseadas.

4. Objetivos para Mejorar la Calidad del Software

Los objetivos se centran en reducir la Deuda Técnica y reforzar la Seguridad, en línea con la ISO 25000.

ID	Objetivo Específico	Métrica a Mejorar	Alineación con ISO 25000
OQ01	Eliminar la Deuda Técnica de Mantenibilidad: Resolver los 4 Code Smells (CS01-CS04) y el 1 Bug (Consistencia).	Bugs (0), Code Smells (0)	Mantenibilidad y Fiabilidad (Robustez del código).
OQ02	Aumentar la Cobertura de Pruebas de la lógica de negocio del Modelo al 60%.	Coverage (≥60%)	Fiabilidad (Madurez y Testabilidad).
OQ03	Mitigar los 3 Security Hotspots mediante validación de entradas y conexiones seguras.	Security Hotspots (0/Revisados)	Seguridad (Integridad y Confidencialidad).

OQ04	Implementar manejo centralizado de conexiones a la DB para mejorar la Modularidad y evitar fallos de conexión aislados.	Deuda Técnica (Diseño)	Mantenibilidad (Facilidad de Modificación y Modularidad).

5. Tareas Específicas y Responsabilidades

El plan de acción consolida la resolución de todos los Bugs, Code Smells y Security Hotspots identificados, además de la implementación de pruebas unitarias. Se asigna un plazo de dos semanas (06 al 19 de octubre de 2025) para estas tareas de calidad intensivas, cumpliendo con los objetivos (OQ01-OQ04) antes del cierre del proyecto.

Tarea de Calidad	Objetivo Asociado	Detalles de la Tarea	Responsabilidad
TC01	OQ01, OQ04	Refactorización de Backend y DB: Solucionar CS01 (Manejo de excepciones en delete_producto) y OQ04 (Conexión centralizada a la DB).	Desarrollador
TC02	OQ01	Refactorización de Frontend: Solucionar el Bug de parseInt y los Code Smells CS02, CS03 y CS04 (Refactorizar	Desarrollador

		cargarProductos y manejar errores de red en JS).	
TC03	OQ03	Mitigación de Seguridad - Backend Crítico: Implementar consultas parametrizadas y validación estricta de entradas en add_producto (SH02) y delete_producto (SH03).	Desarrollador
TC04	OQ02	Implementación de Pruebas: Escribir pruebas unitarias para la lógica del Modelo, enfocándose en la gestión del stock (requerimiento HU03) para alcanzar la cobertura mínima del 60%.	Desarrollador
TC05	OQ03	Revisión Final de Seguridad: Documentar y revisar el Hotspot SH01 y realizar un último análisis de SonarQube para validar el cumplimiento de los objetivos.	Desarrollador

6. Reflexión sobre la Importancia de las Normas de Calidad

Conocer e implementar las normas de calidad de software es fundamental porque traslada la calidad de un concepto abstracto a un conjunto de métricas medibles y objetivas.

1. **Rigor y Estructura:** La familia de normas ISO 25000 (SQuaRE) proporciona un modelo de calidad que va más allá de la simple funcionalidad, cubriendo aspectos como Fiabilidad, Mantenibilidad y Seguridad. En un proyecto ágil como StockPro, la disciplina de medir la calidad con estas métricas estandarizadas evita que la flexibilidad de Scrum se traduzca en código de baja calidad (Pérez A., 2011).

2. **Mitigación de Riesgos y Costos:** La aplicación de herramientas como SonarQube, al aplicar principios de las normas de medición (ISO/IEC 25023), cuantifican la Deuda Técnica. Abordar los Security Hotspots (OQ03) o el 0.0% de Cobertura (OQ02) de manera temprana es más eficiente que corregir fallos en producción. Este enfoque en la calidad como una inversión es clave, pues los métodos y métricas rigurosas permiten un control estricto sobre el proyecto (Heras del Dedo & Álvarez García, s.f.).

3. **Integración Ágil-Estándar:** El proceso de SonarQube transforma la Deuda Técnica en objetivos concretos (OQ01-OQ04), que son inmediatamente incorporados al Product Backlog para ser trabajados. Esto demuestra que las normas de calidad no son un obstáculo burocrático, sino una herramienta de mejora continua que garantiza la sostenibilidad del producto a largo plazo, validando la adaptabilidad de la metodología (Monte Galiano, 2016).

Referencias

- **ISO/IEC 25000 (SQuaRE):** *Systems and software engineering - Systems and software Quality Requirements and Evaluation*.
- **ISO/IEC 25020, 25021, 25022, 25023, 25024:** Normas de medición de calidad de software, incluyendo métricas de calidad de proceso, producto y calidad en uso.
- Pérez A., O. A. (2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM. *INVENTUM, 6*(10), 64-78. https://doi.org/10.26620/uniminuto.inventum.6.10.2011.64-78
- Monte Galiano, J. (2016). *Implantar scrum con éxito: (ed.).* Editorial UOC. https://elibro.net/es/lc/uniminuto/titulos/58575
- Heras del Dedo, R. D. L. & Álvarez García, A. (2017). *Métodos ágiles: Scrum, Kanban, Lean:* (ed.). Difusora Larousse - Anaya Multimedia. https://elibro.net/es/lc/uniminuto/titulos/122933