



Punto 7 Clean Code

Ingeniería de software I

Kevin David Rodriguez Riveros

Frank Sebastian Pardo Amaya

Jorge Andrés Torres Leal

Departamento de Ingeniería de Sistemas e Industrial

Universidad Nacional de Colombia - Sede Bogotá

Facultad de Ingeniería

31 de Enero de 2025

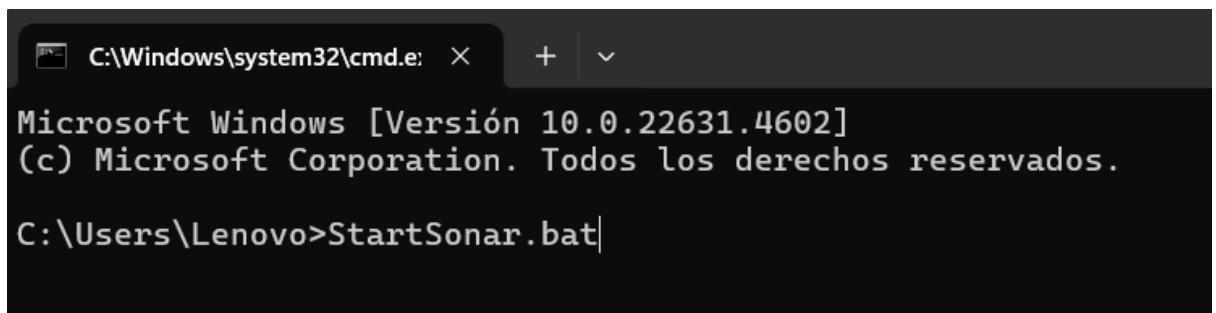
7. Clean Code

En esta sección se documenta el proceso de análisis de código limpio para el proyecto StockEase. Se utilizó SonarQube como herramienta principal de análisis estático, lo que permitió detectar errores, vulnerabilidades y mejorar la mantenibilidad del software.

Herramienta utilizada: SonarQube y Checkstyle: Se eligió SonarQube por su capacidad de integrarse con Maven y su funcionalidad de detección de código duplicado, vulnerabilidades de seguridad y problemas de mantenibilidad. Adicionalmente, se configuró Checkstyle en NetBeans para verificar si el código sigue las convenciones de Java.

Instalación de SonarQube

1. Se descargó **SonarQube Community Edition** desde SonarQube Downloads.
2. Se extrajo y se ejecutó el servidor (en windows) con: `StartSonar.bat`



```
C:\Windows\system32\cmd.e: X + v
Microsoft Windows [Versión 10.0.22631.4602]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\Lenovo>StartSonar.bat|
```

3. Se accedió a la interfaz en `http://localhost:9000`.

Instalación de SonarScanner

1. Se descargó SonarScanner.
2. Se configuró la variable de entorno SONAR_SCANNER_HOME.
3. Se agregó al PATH para su ejecución global.

Configuración en Maven

1. Se agregó el siguiente plugin en `pom.xml`:

```
<properties>
  <sonar.host.url>http://localhost:9000</sonar.host.url>
</properties>

<build>
  <plugins>
    <plugin>
      <groupId>org.sonarsource.scanner.maven</groupId>
      <artifactId>sonar-maven-plugin</artifactId>
      <version>3.9.1.2184</version>
    </plugin>
  </plugins>
</build>
```

2. Se ejecutó el análisis con el comando:

mvn clean verify sonar:sonar

3. Se revisaron los resultados en **http://localhost:9000**.

Configuración de Checkstyle en NetBeans

1. Se instaló el plugin de Checkstyle en NetBeans desde la sección de complementos.
2. Se configuró para que siga las reglas de Google Java Style Guide.
3. Se ejecutó el análisis y se revisaron las advertencias generadas.

Evaluación de los integrantes

Cada integrante realizó una breve evaluación sobre la calidad del código:

- **Kevin Rodríguez:** El código es claro en su mayoría, pero hay fragmentos con demasiada lógica en una sola función.
- **Frank Pardo:** La organización del código facilita la comprensión, aunque algunas variables podrían ser más descriptivas.
- **Jorge Torres:** Hay buena documentación, pero noté que ciertos métodos largos podrían dividirse en funciones más pequeñas.