

**Artigo escrito por:**



**Elves Sampaio Moura**

(elvessampaio@live.com)

**Cleartech – Turma 2023 (Squad Corvo)**  
**Trilha QA**

## **Análise Estática de Código e Qualidade do Software: Uma Exploração do SonarQube**

Resumo: A qualidade do código é um aspecto fundamental no desenvolvimento de software, pois afeta diretamente a manutenibilidade, confiabilidade e eficiência do sistema. Uma das abordagens cruciais para avaliar e melhorar a qualidade do código é a análise estática, que verifica o código-fonte sem executá-lo. Este artigo explora o conceito de análise estática de código e como a ferramenta SonarQube desempenha um papel vital na identificação de melhorias e problemas de qualidade do código. Será discutido como o SonarQube é configurado para reconhecer a qualidade do código, e serão fornecidos exemplos práticos do seu uso.

### **1. Introdução**

A qualidade do código é essencial para garantir que um software seja robusto, seguro e de fácil manutenção. Uma maneira de avaliar e melhorar a qualidade do código é por meio da análise estática, um processo que examina o código-fonte sem executá-lo. Entre as ferramentas mais renomadas para análise estática de código, o SonarQube se destaca como uma solução poderosa que ajuda as equipes de desenvolvimento a identificarem problemas e melhorias na qualidade do código. Neste artigo, vamos explorar como o SonarQube funciona e como ele é configurado para reconhecer a qualidade do código.

### **2. Análise Estática de Código**

A análise estática de código é um processo que examina o código-fonte sem executar o programa. Ela verifica o código em busca de erros de sintaxe, más práticas de programação, vulnerabilidades de segurança, e conformidade com padrões de codificação. Esse processo é essencial para identificar problemas antes que eles causem falhas no sistema em produção.

A análise estática pode ser realizada manualmente, mas ferramentas automatizadas, como o SonarQube, tornam esse processo mais eficiente e abrangente. Ela ajuda a detectar uma ampla gama de problemas de código, desde problemas simples, como variáveis não utilizadas, até questões mais complexas, como vulnerabilidades de segurança.

### **3. Como o SonarQube funciona**

O SonarQube é uma plataforma de código aberto que oferece uma variedade de recursos para análise estática de código e avaliação da qualidade do software. Ele funciona da seguinte maneira:

- **Coleta de Dados**

O SonarQube coleta dados do código-fonte, que podem incluir arquivos de origem, bibliotecas externas e configurações do projeto. Ele suporta várias linguagens de programação, tornando-o versátil para equipes de desenvolvimento que trabalham em projetos diversos.

- **Análise Estática**  
Após a coleta de dados, o SonarQube realiza uma análise estática detalhada do código-fonte. Ele verifica cada arquivo em busca de problemas de qualidade, como código duplicado, complexidade excessiva, violações de padrões de codificação e vulnerabilidades conhecidas.
- **Geração de Relatórios**  
Depois de concluir a análise, o SonarQube gera relatórios detalhados que destacam os problemas encontrados no código. Esses relatórios são acessíveis por meio de uma interface da web amigável, permitindo que os desenvolvedores visualizem facilmente os problemas e tomem medidas para corrigi-los.

#### 4. Configuração do SonarQube para Reconhecer a Qualidade do Código

Uma das características mais poderosas do SonarQube é sua capacidade de ser configurado para atender às necessidades específicas de um projeto. A configuração envolve a definição de regras e padrões de qualidade que a ferramenta usará para avaliar o código. Alguns aspectos-chave da configuração incluem:

- **Definição de Perfil de Qualidade**  
Os perfis de qualidade no SonarQube permitem que você selecione um conjunto de regras e padrões que se aplicam ao seu projeto. Você pode escolher entre os perfis predefinidos para linguagens populares ou criar seu próprio perfil personalizado.
- **Configuração de Métricas e Limiares**  
Você pode configurar métricas específicas que deseja rastrear, como a complexidade ciclomática ou a quantidade de código duplicado permitida. Além disso, é possível definir limites para essas métricas, indicando quando o código deve ser considerado de baixa qualidade.
- **Personalização de Regras**  
O SonarQube permite personalizar as regras de qualidade, adicionando ou removendo regras específicas do seu projeto. Isso garante que o SonarQube se alinhe com os padrões de codificação da equipe.

#### 5. Exemplos Práticos

Vamos considerar alguns exemplos práticos de como o SonarQube pode identificar problemas de qualidade do código:

- **Detecção de Código Duplicado**  
O SonarQube pode destacar trechos de código duplicados, o que ajuda a reduzir a duplicação e melhorar a manutenibilidade.
- **Avaliação da Complexidade Ciclométrica**  
A ferramenta pode calcular a complexidade ciclométrica de um método e alertar quando ela excede um limite predefinido.

### Detecção de Código Duplicado e Avaliação da Complexidade Ciclométrica no SonarQube: Melhorando a Qualidade do Código de Software

A qualidade do código é um dos pilares fundamentais para o desenvolvimento de software confiável e de fácil manutenção. A detecção de código duplicado e a avaliação da complexidade ciclométrica são duas práticas essenciais nesse contexto. O SonarQube, uma ferramenta de análise estática de código amplamente utilizada, desempenha um papel vital ao identificar e relatar problemas relacionados a esses aspectos críticos do código-fonte. Neste texto, exploraremos como o SonarQube realiza a detecção de código duplicado e avaliação da complexidade ciclométrica e como essas funcionalidades podem contribuir para melhorar a qualidade do código de software.

## Detecção de Código Duplicado no SonarQube

A detecção de código duplicado é uma técnica que visa encontrar partes do código que são semanticamente idênticas ou semelhantes. Essa duplicação pode ocorrer em diferentes níveis, desde pequenos fragmentos de código até módulos inteiros. A presença de código duplicado não apenas torna o código difícil de manter, mas também aumenta o risco de erros, uma vez que as alterações feitas em uma instância duplicada podem ser esquecidas na outra.

O SonarQube aborda a detecção de código duplicado de forma eficiente. Ele analisa o código-fonte em busca de padrões semelhantes e relata as ocorrências encontradas, permitindo que os desenvolvedores identifiquem e resolvam problemas de duplicação. A interface do SonarQube fornece informações detalhadas sobre a duplicação, incluindo o trecho de código duplicado e onde ele está presente, simplificando o processo de correção.

A eliminação de código duplicado não apenas melhora a manutenibilidade, mas também reduz a probabilidade de erros e torna o código mais limpo e legível.

## Avaliação da Complexidade Ciclômática no SonarQube

A complexidade ciclômática é uma métrica que avalia a complexidade de um método ou função em um programa. Ela mede a quantidade de caminhos independentes que podem ser percorridos em um método, o que está diretamente relacionado à sua complexidade. Quanto maior a complexidade ciclômática, maior a probabilidade de o método ser difícil de entender, testar e manter.

O SonarQube calcula a complexidade ciclômática para métodos e funções no código-fonte e fornece informações sobre os casos em que a complexidade atinge níveis preocupantes. Isso permite que os desenvolvedores identifiquem áreas críticas de código que podem requerer refatoração ou simplificação.

Além de fornecer uma pontuação de complexidade ciclômática, o SonarQube também destaca as partes específicas do código que contribuem para essa métrica, facilitando a identificação das áreas que requerem atenção.

## Conclusão

A análise estática de código é uma prática fundamental para garantir a qualidade do software. O SonarQube é uma ferramenta poderosa que simplifica esse processo, identificando problemas e melhorias de qualidade do código de forma eficiente. Ao configurar o SonarQube de acordo com as necessidades do projeto, as equipes de desenvolvimento podem manter um alto padrão de qualidade, resultando em software mais confiável e de fácil manutenção.

A detecção de código duplicado e a avaliação da complexidade ciclômática são práticas fundamentais para melhorar a qualidade do código de software. O SonarQube oferece uma abordagem eficaz para ambas as áreas, permitindo que as equipes de desenvolvimento identifiquem problemas de duplicação e complexidade em seus projetos.

Ao utilizar o SonarQube como parte do ciclo de desenvolvimento de software, as equipes podem manter um código mais limpo, reduzir a probabilidade de erros e melhorar a manutenibilidade geral do sistema. Isso contribui para a criação de software mais robusto e confiável, atendendo às expectativas dos usuários e mantendo a eficiência no desenvolvimento e na manutenção a longo prazo.

Em resumo, o SonarQube desempenha um papel crucial na avaliação da qualidade do código, e sua configuração personalizável o torna uma ferramenta versátil para uma ampla variedade de projetos de desenvolvimento de software. A detecção de código duplicado e a avaliação da complexidade ciclômática no SonarQube são ferramentas valiosas para alcançar a qualidade do código desejada em projetos de desenvolvimento de software.

## Referências:

1. Site oficial do SonarQube Docs. - (<https://www.sonarqube.org/>);
2. (7Masters Qualidade de Código - Complexidade Ciclométrica com Jakeliny Gracielly) - 7Masters Qualidade de Código - Complexidade Ciclométrica com Jakeliny Gracielly;
3. Melhoria da qualidade e padrões de código fonte utilizando ferramentas de análise estática - Leandro Ugioni ([https://www.slideshare.net/LeandroUgioni/melhoria-da-qualidade-e-padres-de-codigo-fonte-utilizando-ferramentas-de-anlise-esttica](https://www.slideshare.net/LeandroUgioni/melhoria-da-qualidade-e-padres-de-codigo-fonte-utilizando-ferramentas-de-analise-esttica));
4. Análise qualitativa de código com SonarQube - Orlando Burli Júnior (<https://blog.impulso.team/analise-qualitativa-de-codigo-com-sonarqube/>);
5. Qualidade de código com SonarQube - Full Cycle ([https://www.youtube.com/watch?v=VBjAf6KUhzY&ab\\_channel=FullCycle](https://www.youtube.com/watch?v=VBjAf6KUhzY&ab_channel=FullCycle)).