# SonarQube

Este livro contém informações e instruções de utilização do SonarQube em projetos.

# Apresentação

**Autoria e identificação**

Este material foi elaborado por **Guilherme Rocha Leite**, Analista de Tecnologia da Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM), e disponibilizado no Sistema de Documentação de Serviços da UFVJM.

**Data de criação do livro:** 16 de setembro de 2026.
**Sistema de publicação:** Documentação de Serviços – UFVJM.

# Introdução e instruções de execução

O **SonarQube** é uma ferramenta gratuita para análise de código que visa auxiliar o desenvolvedor a descobrir blocos de código que podem ser reescritos de maneira mais eficiente e padronizada semântica e sintaticamente, a fim de maximizar a qualidade de escrita do código. Além disso, ele também consegue analisar a cobertura de testes unitários de um projeto, ajudando a identificar se os testes estão bem desenvolvidos e com ampla cobertura.

## Inicialização

Existe um projeto no workspace da DDS que pode ser utilizado e é com ele que a presente documentação foi criada. Para executá-lo:

```
cd ~/ && git clone git@git.dds.ufvjm.edu.br:dds/sonarqube.git
cd ~/sonarqube
docker compose up -d
```

Aguarde 1-2 minutos para o SonarQube inicializar completamente.

## Acessando o SonarQube

Abra no navegador: http://localhost:9000

**Login padrão:**

- Usuário: `admin`
- Senha: `admin`

⚠ **Importante:** No primeiro acesso, você será solicitado a alterar a senha padrão.

Em seguida, crie um novo projeto, definindo um nome e uma chave que serão utilizados em passos posteriores. Também será solicitada a geração de um token, ao gerá-lo, copie e salve em algum local pois ele também será utilizado posteriormente.

## □□ Configuração Avançada - sonar-project.properties

Crie um arquivo `sonar-project.properties` na raiz do projeto que deseja escanear:

```
# Identificação do projeto
sonar.token=token-projeto # gerado no passo anterior
sonar.projectKey=meu-projeto # definido no passo anterior
sonar.projectName=Meu Projeto # definido no passo anterior
sonar.projectVersion=1.0

# Diretórios
sonar.sources=.
sonar.sourceEncoding=UTF-8

# Exclusões
sonar.exclusions=**/vendor/**,**/node_modules/**,**/tests/**,**/tpls/compiled/**

# Para PHP
sonar.language=php
sonar.php.coverage.reportPaths=coverage.xml
sonar.php.tests.reportPath=tests/_output/report.xml

# Para JavaScript
# sonar.javascript.lcov.reportPaths=coverage/lcov.info
```

## □□ Escanear um Projeto

### Opção 1: Usando o script (Recomendado)

```
cd ~/sonarqube
./scan-project.sh /caminho/do/projeto nome-do-projeto
```

Exemplo: Escanear o projeto Void:

`./scan-project.sh ~/void pressiga`

### Opção 2: Manualmente com Docker

```
docker run --rm \
    --network sonarqube_sonarqube-network \
    -e SONAR_HOST_URL="http://sonarqube:9000" \
    -e SONAR_LOGIN="admin" \
    -e SONAR_PASSWORD="sua-nova-senha" \
    -v "/caminho/do/projeto:/usr/src" \
```

```
sonarsource/sonar-scanner-cli \
-Dsonar.projectKey="meu-projeto" \
-Dsonar.projectName="Meu Projeto" \
-Dsonar.sources=.
```

## □□ Quality Gates e Badges

É possível obter o selo de qualidade de código de acordo com critérios pré-estabelecidos, desde que o projeto atenda a esses requisitos mínimos. É possível defini-los manualmente da seguinte maneira:

1. Acesse: http://localhost:9000/quality_gates
2. Crie ou edite um Quality Gate
3. Defina os critérios mínimos:
   - **Coverage**: mínimo 80%
   - **Duplications**: máximo 3%
   - **Maintainability Rating**: A ou B
   - **Reliability Rating**: A
   - **Security Rating**: A
   - **Vulnerabilities**: 0
   - **Bugs**: 0

**Gerar Badge (Selo):**

Após escanear um projeto, você pode gerar badges em:

**Dashboard do projeto → Project Information → Badges**

Exemplos de URLs de badges:

```
http://localhost:9000/api/project_badges/measure?project=pressiga&metric=alert_status
http://localhost:9000/api/project_badges/measure?project=pressiga&metric=coverage
http://localhost:9000/api/project_badges/measure?project=pressiga&metric=bugs
http://localhost:9000/api/project_badges/measure?project=pressiga&metric=vulnerabilities
```

## □□ Comandos Úteis

**Ver logs do SonarQube**

`docker logs sonarqube -f`

**Parar o SonarQube**

```
cd ~/sonarqube
docker compose down
```

**Reiniciar o SonarQube**

```
cd ~/sonarqube
docker compose restart
```

**Remover tudo (incluindo dados)**

```
cd ~/sonarqube
docker compose down -v
```

# Gerando relatórios

Para gerar relatórios de cobertura para o SonarQube:

1. Instale as dependências:

```
composer require --dev phpunit/php-code-coverage
```

2. Configure o `codeception.yml`:

```
coverage:
    enabled: true
    include:
        - apps/*
        - core/*
    exclude:
        - vendor/*
        - tests/*
```

3. Execute os testes com cobertura:

**Opção 1 - Apenas testes unitários (recomendado para cobertura):**

`XDEBUG_MODE=coverage ./vendor/bin/codecept run unit --coverage --coverage-xml`

**Opção 2 - Se precisar rodar acceptance com cobertura:**

```
# Limpar snapshots antes
rm -rf tests/_data/snapshots/*

# Executar sem usar snapshots
XDEBUG_MODE=coverage ./vendor/bin/codecept run acceptance --coverage --coverage-xml
```

**Opção 3 - Usar script customizado do projeto:**

`./tests.sh --coverage`

⚠ **Nota:** O Xdebug pode causar conflitos com cookies no WebDriver durante testes de aceitação. Se encontrar erro `invalid cookie domain`, use apenas testes unitários para gerar cobertura ou limpe os snapshots antes de executar.

Ou se preferir, exporte a variável antes:

```
export XDEBUG_MODE=coverage
./vendor/bin/codecept run unit --coverage --coverage-xml
```

4. Escaneie com o SonarQube apontando para o relatório gerado.

## □□ Métricas Importantes

- **Bugs**: Problemas que podem causar erros
- **Vulnerabilities**: Problemas de segurança
- **Code Smells**: Código que pode ser melhorado
- **Coverage**: Porcentagem de código coberto por testes
- **Duplications**: Código duplicado
- **Technical Debt**: Tempo estimado para corrigir todos os problemas

# Extras

Para mais informações sobre a ferramenta, seguem alguns links da documentação oficial do SonarQube e do Codeception:

- Documentação oficial SonarQube: https://docs.sonarqube.org/
- Quality Gates: https://docs.sonarqube.org/latest/user-guide/quality-gates/
- SonarScanner: https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/
- Codeception: https://codeception.com/