

# SonarQube

É uma ferramenta que consegue identificar as alterações que queremos fazer no nosso código (**pull requests**) e quando identifica ela analisa todo o código e verifica se não tem erros, faz testes, e também pode verificar se há algum tipo de gambiarra no código, verifica se tem as melhores práticas aplicadas, verifica se tem questões de vulnerabilidade de segurança, etc.

Para configurar a gente baixa o **sonar** como arquivo zip e coloca na pasta do **windows** e depois configura como **variável de ambiente path** e depois a gente roda o **docker** com o comando **docker run -d --name sonarqube -p 9000:9000 sonarqube:[Versão desejada]**

Depois disso iremos entrar no <http://localhost:9000> para fazer login (de primeiro acesso o usuário e senha ficam como 'admin') no **SonarQube** e criar um projeto e depois executar os comandos que ele passará de acordo.

Para rodar em projetos do **GitHub** a gente cria um arquivo **.yaml** com os comandos para usar **actions** do **GitHub** para rodar o OS desejado, baixar a linguagem, checar o código, criar o **.env** para projetos que tem variáveis de ambiente, para executar os teste unitários também e para funcionar o **Sonar Scanner**(para isso vamos usar 2 variáveis de ambiente que 1 é o **GitHub Token** que é um **Secret Token** do **GitHub** que todo mundo já tem e já é embutida outra é o **Sonar Token** que precisamos gerar no **SonarQube** e criar um **Secret** no nosso repositório do **GitHub**).

## EXEMPLO DE .YAML

32 lines (28 sloc) | 769 Bytes

```
1  on: [push, pull_request]
2  name: Test
3  jobs:
4    test:
5      strategy:
6        matrix:
7          go-version: [1.14.x]
8          platform: [ubuntu-latest]
9      runs-on: ${{ matrix.platform }}
10     steps:
11       - name: Install Go
12         uses: actions/setup-go@v1
13         with:
14           go-version: ${{ matrix.go-version }}
15
16       - name: Checkout code
17         uses: actions/checkout@v2
18
19       - name: Creating .env
20         uses: canastro/copy-file-action@master
21         with:
22           source: ".env.example"
23           target: ".env"
24
25       - name: Test
26         run: go test ./...
27
28       - name: SonarCloud Scan
29         uses: sonarsource/sonarcloud-github-action@master
30         env:
31           GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
32           SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
```

**AONDE CONFIGURAR A SECRET TOKEN DO SONAR.**

< Code 1 Issues 2 Pull requests 4 Actions Projects 0 Wiki Security 0 Insights Settings

Options  
Manage access  
Branches  
Webhooks  
Notifications  
Integrations  
Deploy keys  
**Secrets**  
Actions  
Dependabot alerts

Moderation  
Interaction limits  
Reported content

## Secrets

New secret

Secrets are environment variables that are **encrypted** and only exposed to selected actions. Anyone with **collaborator** access to this repository can use these secrets in a workflow.

Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more.](#)

### Repository secrets

**There are no secrets for this repository.**

Encrypted secrets allow you to store sensitive information, such as access tokens, in your repository.

Secrets can also be created at the organization level and authorized for use in this repository.

### Organization secrets

Manage organization secrets

SONAR\_TOKEN Updated 21 hours ago

Depois disso quando executarmos um **pull request** ou **push** irá rodar os **testes unitários** e o **SonarQube**, vemos o progresso na aba **Actions** e o resultado em **pull requests**.