

PROJETO PDSI2

VIAJEI

Daniel Augusto

Tiago Henrique

Vanessa Fernandes

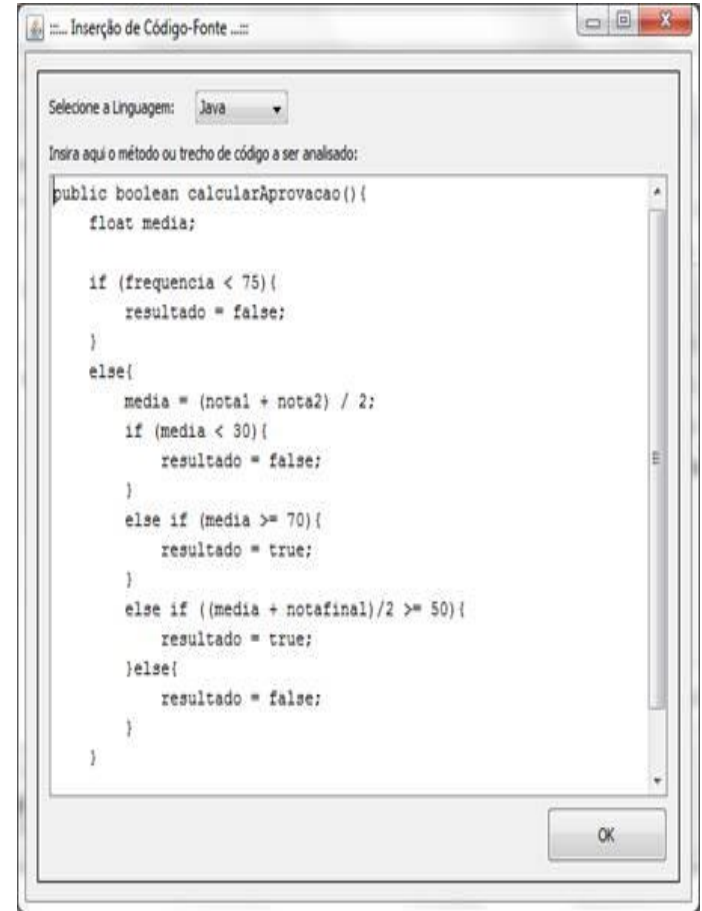
William Nóbrega

TESTES UNITÁRIOS

- ComplexGraph:

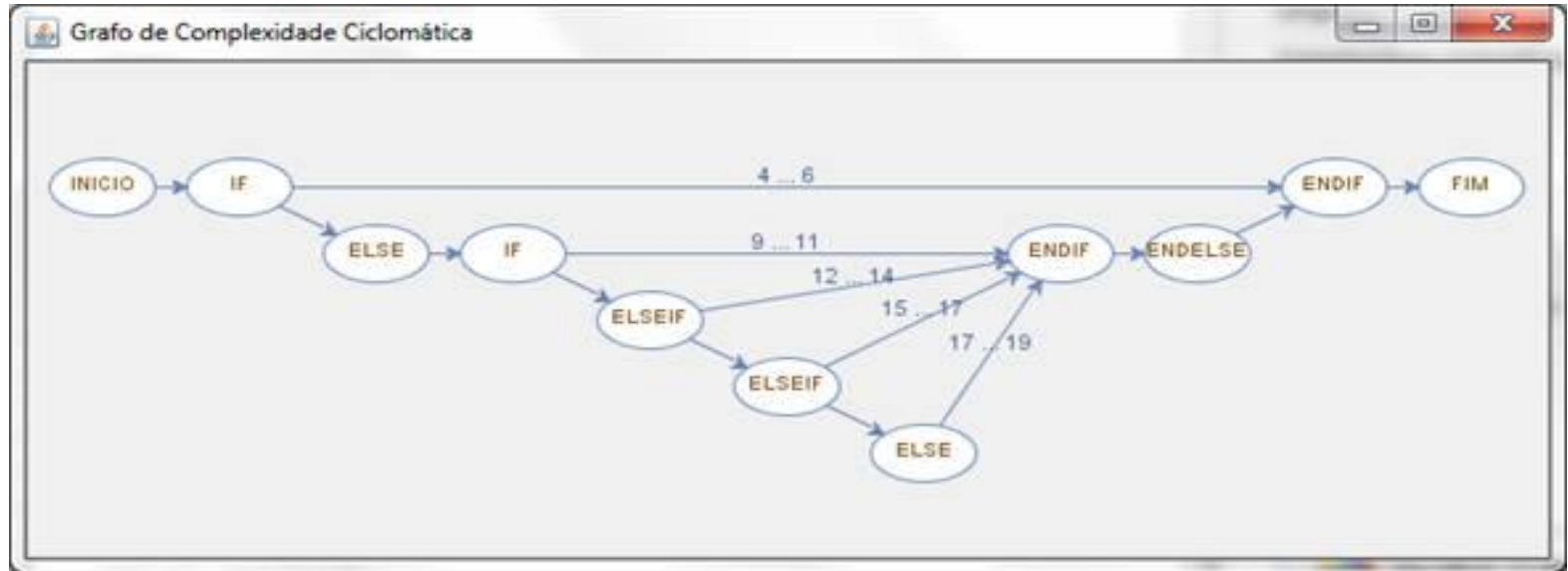
O objetivo é obter o número e grafo da complexidade ciclomática antes de iniciar o teste unitário.

Exemplo de entrada:



TESTES UNITÁRIOS

Exemplo de saída:



EXECUTANDO O TESTE UNITÁRIO

- JUnit

Será criado um caso de teste para cada um dos métodos mais importantes do projeto.

Para a execução automatizada usaremos o Jenkins.



TESTES DE DESEMPENHO

- Performance Plugin

O Jenkins irá gerar gráficos com o relatório de tendências de desempenho, além informar os status finais de build: bom, instável ou com falha.

- JMeter

A partir dos limites de falhas definidos para os builds, qualquer desvio apresentado nos testes irá gerar um erro.

TESTES DE DESEMPENHO

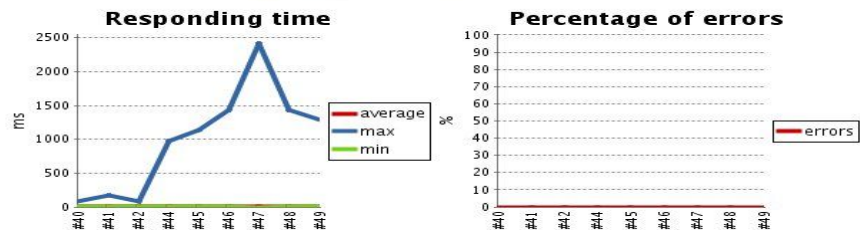
- [Retour au tableau de bord](#)
- [Statut](#)
- [Changements](#)
- [Espace de travail](#)
- [Lancer un build](#)
- [Supprimer ce Projet](#)
- [Configurer](#)
- [Modules](#)
- [Performance trend](#)**

Performance Trend

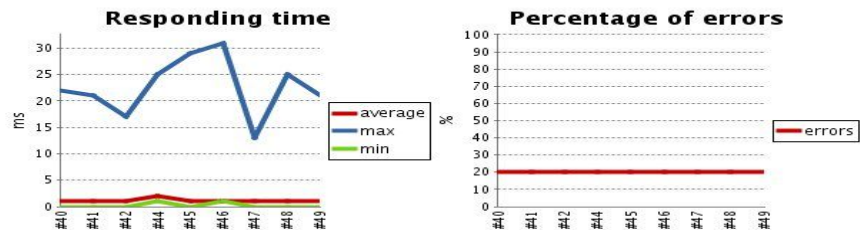
[Configure...](#)

Historique des builds (tendance)	
#49	4 janv. 2010 14:44:58
#48	4 janv. 2010 10:43:24
#47	31 déc. 2009 16:57:10
#46	30 déc. 2009 10:21:01
#45	24 déc. 2009 17:22:34
#44	24 déc. 2009 16:02:58
#43	24 déc. 2009 15:55:47
#42	24 déc. 2009 15:24:39
#41	24 déc. 2009 15:20:48
#40	24 déc. 2009 15:16:31
#39	24 déc. 2009 15:10:50
#38	24 déc. 2009 15:02:52
#37	24 déc. 2009 14:57:18
#36	24 déc. 2009 14:36:46
#35	24 déc. 2009 11:11:30
#34	23 déc. 2009 15:28:02
#33	23 déc. 2009 15:20:16
#32	23 déc. 2009 15:15:03
#31	11 déc. 2009 11:16:48
#30	11 déc. 2009 11:08:10

perf_continue_test1.xml



perf_continue_test2.xml

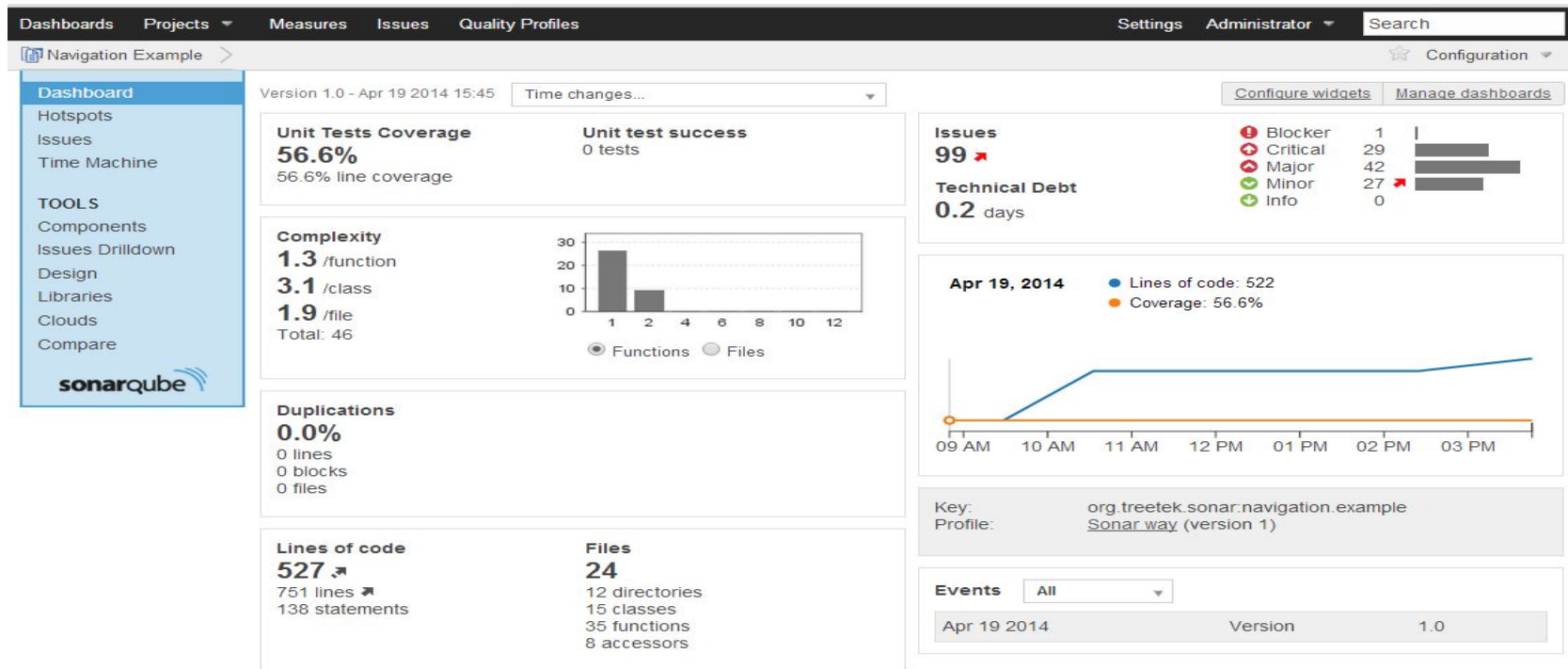


TESTES DE QUALIDADE

- SonarQube

Disponibiliza um conjunto de ferramentas que analisam a arquitetura e design, número de comentários, regras de linguagem, nível de complexidade, código duplicado, cobertura de testes no código, etc

TESTES DE QUALIDADE



TESTES DE FUNCIONALIDADE

- Selenium integrado com JUnit

Executar cada Caso de Uso, fluxo ou função, utilizando dados válidos e inválidos.

Os resultados esperados ocorrem quando dados válidos são usados;

As mensagens de erro correspondem ao esperado quando dados inválidos são utilizados;

ORQUESTRAÇÃO / PIPELINE



Sonar

Sonar installations

Name

Disable ☐

Check to quickly disable Sonar on all jobs.

List of Sonar installations

Ações pós-construção

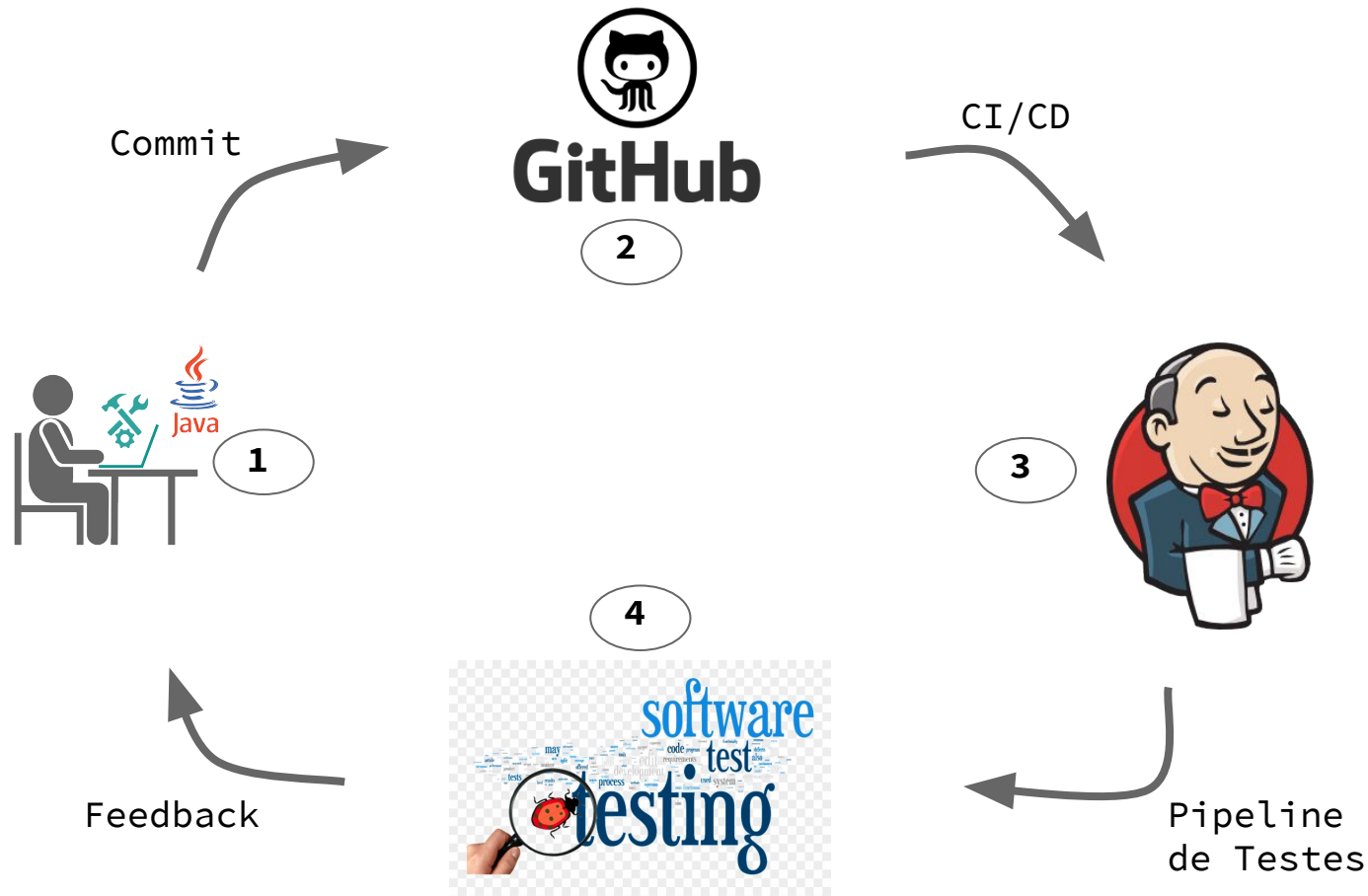
- ☐ Construir outros projetos
- ☐ Arquivar os artefatos
- ☐ Agrega o resultado dos testes com projeto filho
- ☐ Implantar os artefatos no repositório Maven
- ☒ Sonar

Jenkinsfile:

```
pipeline {
  agent any

  stages {
    stage('Build') {
      steps {
        echo 'Building..'
      }
    }
    stage('Test') {
      steps {
        echo 'Testing..'
      }
    }
    stage('Deploy') {
      steps {
        echo 'Deploying....'
      }
    }
  }
}
```

FLUXO



OUTROS TESTES

- **Teste de Estresse:** Submete o sistema a situações anormais, de forma que o programa demande recursos em quantidade, frequência ou volume anormais.
- **Teste de Integração:** Visa garantir que os componentes da aplicação desenvolvidos separadamente funcionem perfeitamente quando integrados.
- **Testes de Segurança:** Verifica se a aplicação se comporta adequadamente mediante as mais diversas tentativas ilegais de acesso.

REFERÊNCIAS

Guia Técnicas de Teste - Celepar Informática do Paraná

<https://jmeter.apache.org/>

<https://wiki.jenkins.io/display/JENKINS/Performance+Plugin>

<https://www.devmedia.com.br/teste-unitario-com-junit-e-complexgraph/31382>

<https://blog.db1.com.br/importancia-da-integracao-continua-utilizando-jenkins-e-sonarqube>

<http://www.universidadejava.com.br/materiais/ic-jenkins-maven-subversion-sonar>