



PROJETO PDSI2

VIAJEI

**Daniel Augusto**  
**Tiago Henrique**  
**Vanessa Fernandes**  
**William Nóbrega**

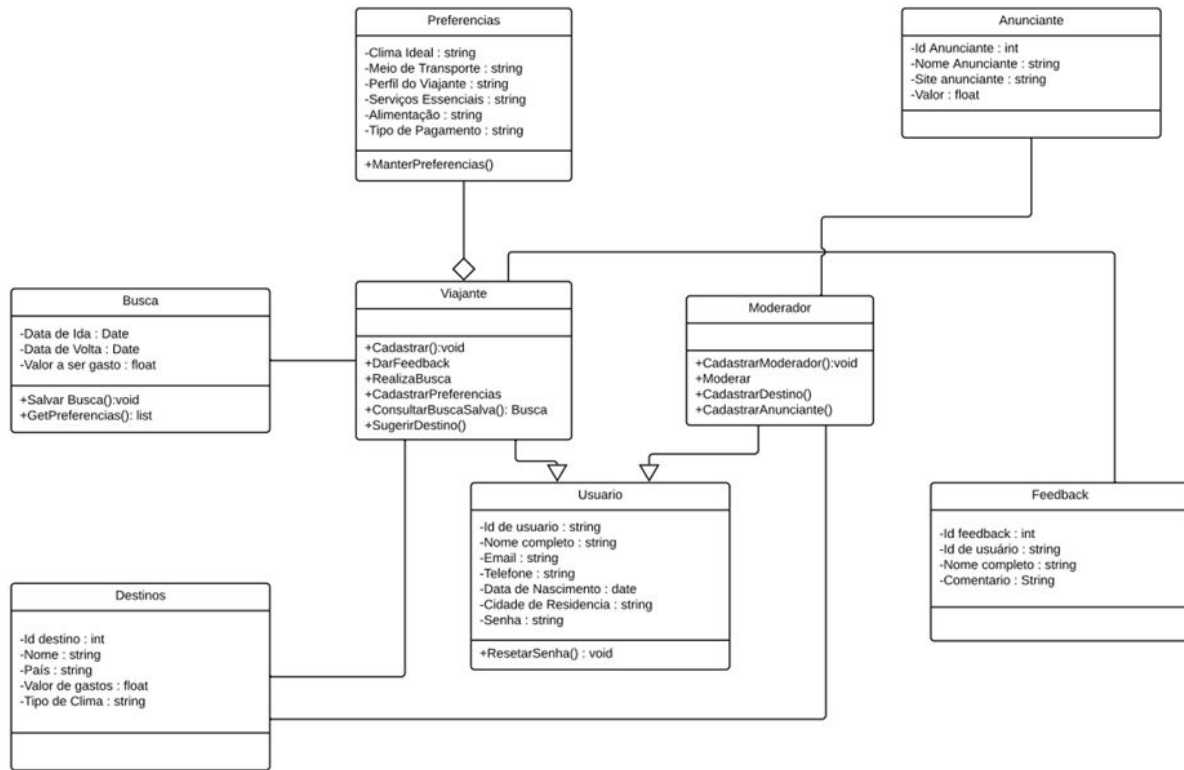
# O PROJETO

A ideia do projeto é desenvolver um sistema agente de viagens, que sugere ao usuário destinos para viajar com base em suas preferências pessoais e em quanto dinheiro o usuário tem disponível para gastar.

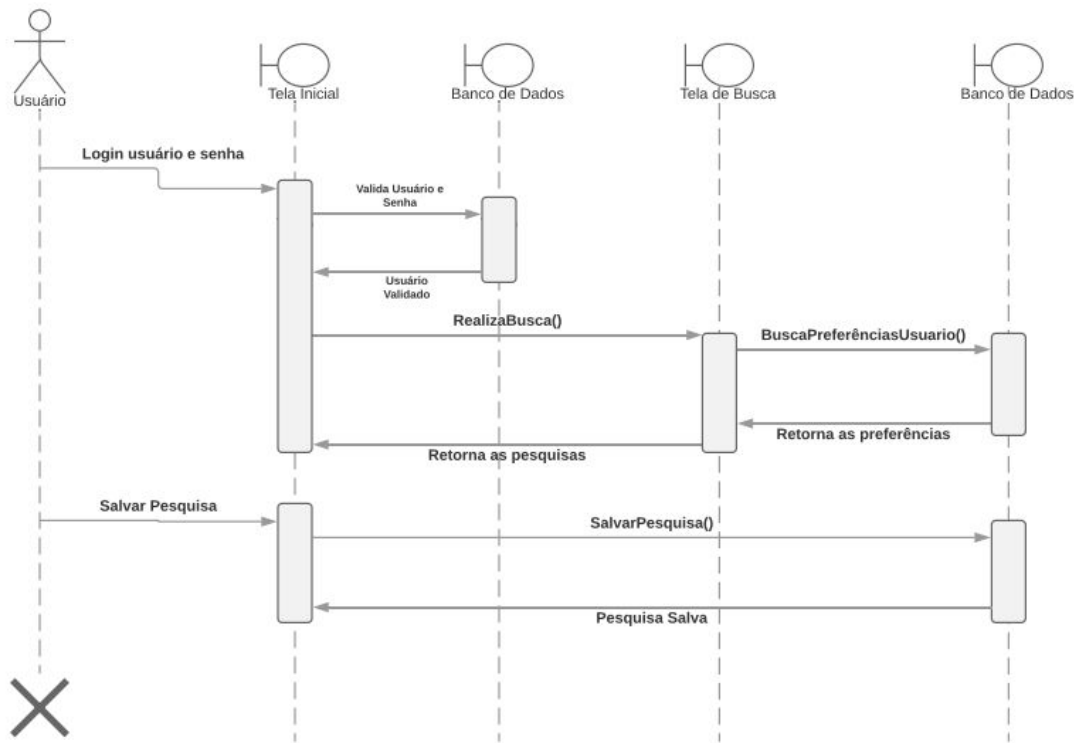
Além da sugestão de destinos, os usuários podem contribuir com suas experiências nos destinos, enriquecendo assim o mecanismo de busca da aplicação. Os usuários podem indicar quanto gastaram em cada destino, dar dicas e anexar imagens do local. Toda contribuição feita pelo usuário passa por uma moderação para garantir a veracidade e fidelidade das buscas.

# VISÃO LÓGICA

A visão lógica tem como foco a representação das classes e objetos principais da aplicação, aqui representada pelo diagrama de classes e diagrama de sequência

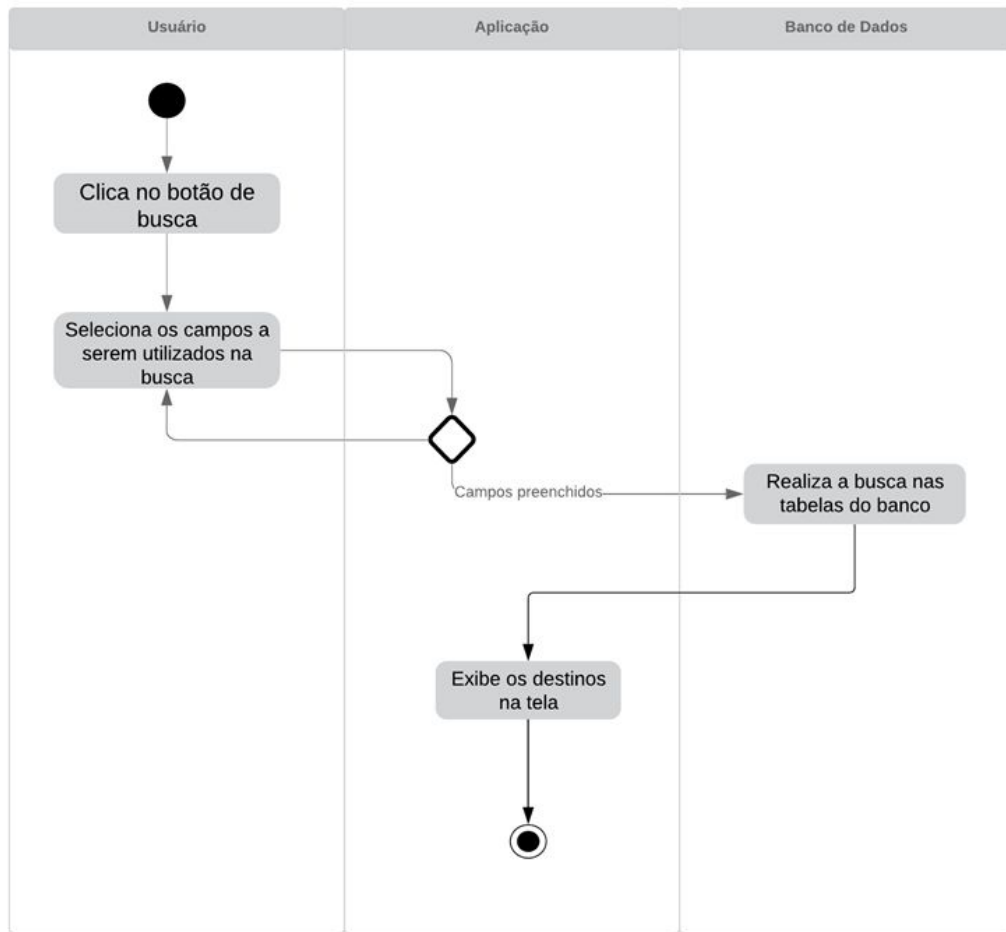


# VISÃO LÓGICA - DIAGRAMA DE SEQUÊNCIA



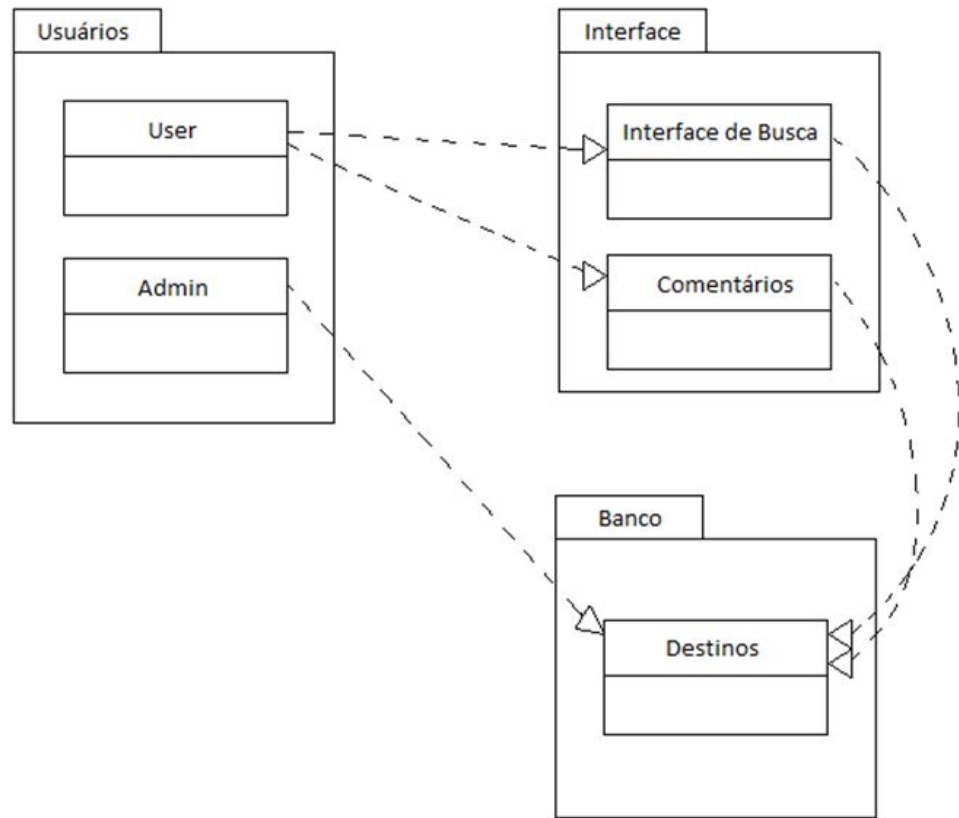
# VISÃO DE PROCESSOS

A visão de processos foca na representação dos processos de um sistema e como eles se comunicam. Ao lado, um diagrama de atividades de um caso de uso da aplicação



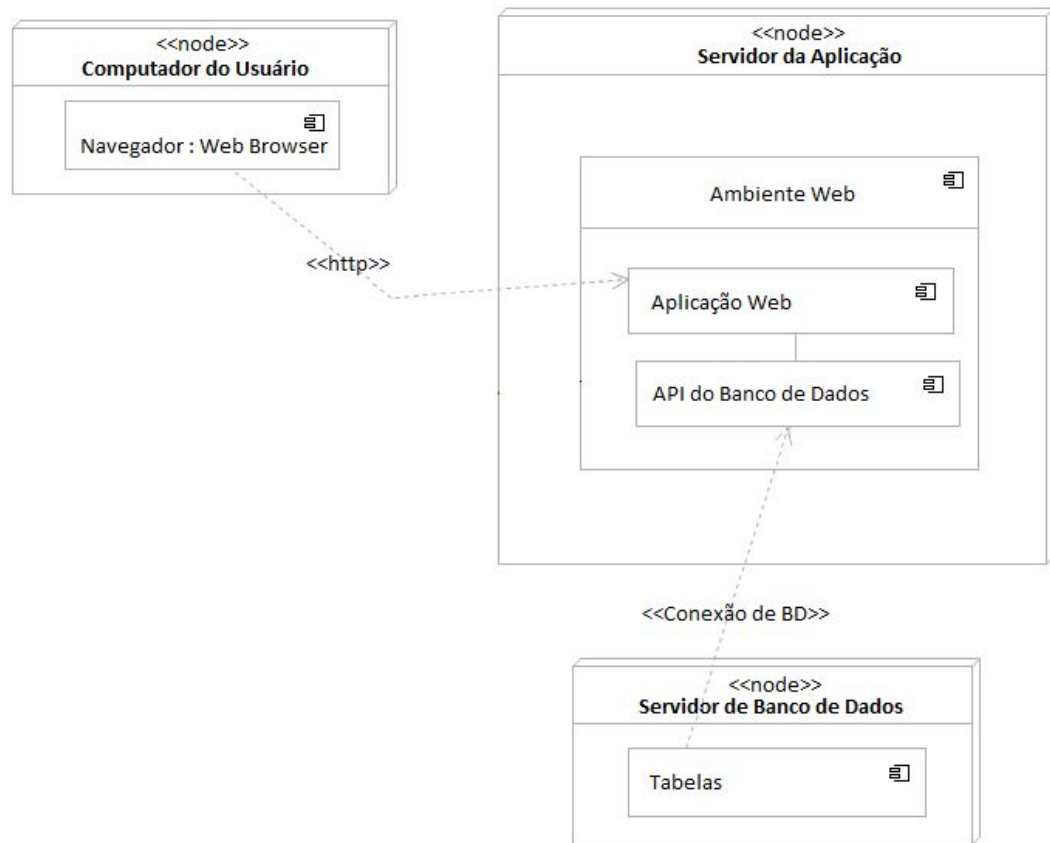
# VISÃO DE IMPLEMENTAÇÃO

A visão de implementação é voltada à descrição dos módulos do sistema, de uma forma mais abstrata, mostrando itens estruturais e comportamentais. O diagrama de pacotes é um dos exemplos de diagrama que demonstra visualmente a visão de implementação.



# VISÃO DE IMPLANTAÇÃO

Nessa visão são descritos os componentes para que a aplicação funcione em rede. Ressalta a escalabilidade, desempenho e confiabilidade da aplicação.



# DIAGRAMA DE CASO DE USO





# CARACTERÍSTICAS DE UM BOM CÓDIGO

- manutenabilidade
- baixa curva de aprendizagem
- segurança
- confiabilidade

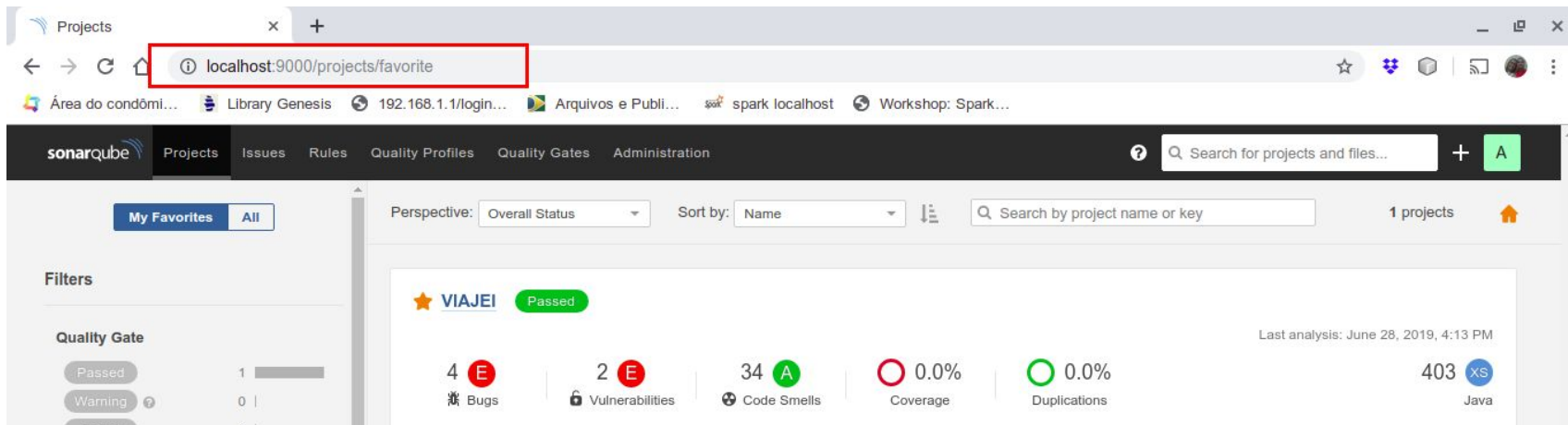
# TESTE DE QUALIDADE - SONARQUBE

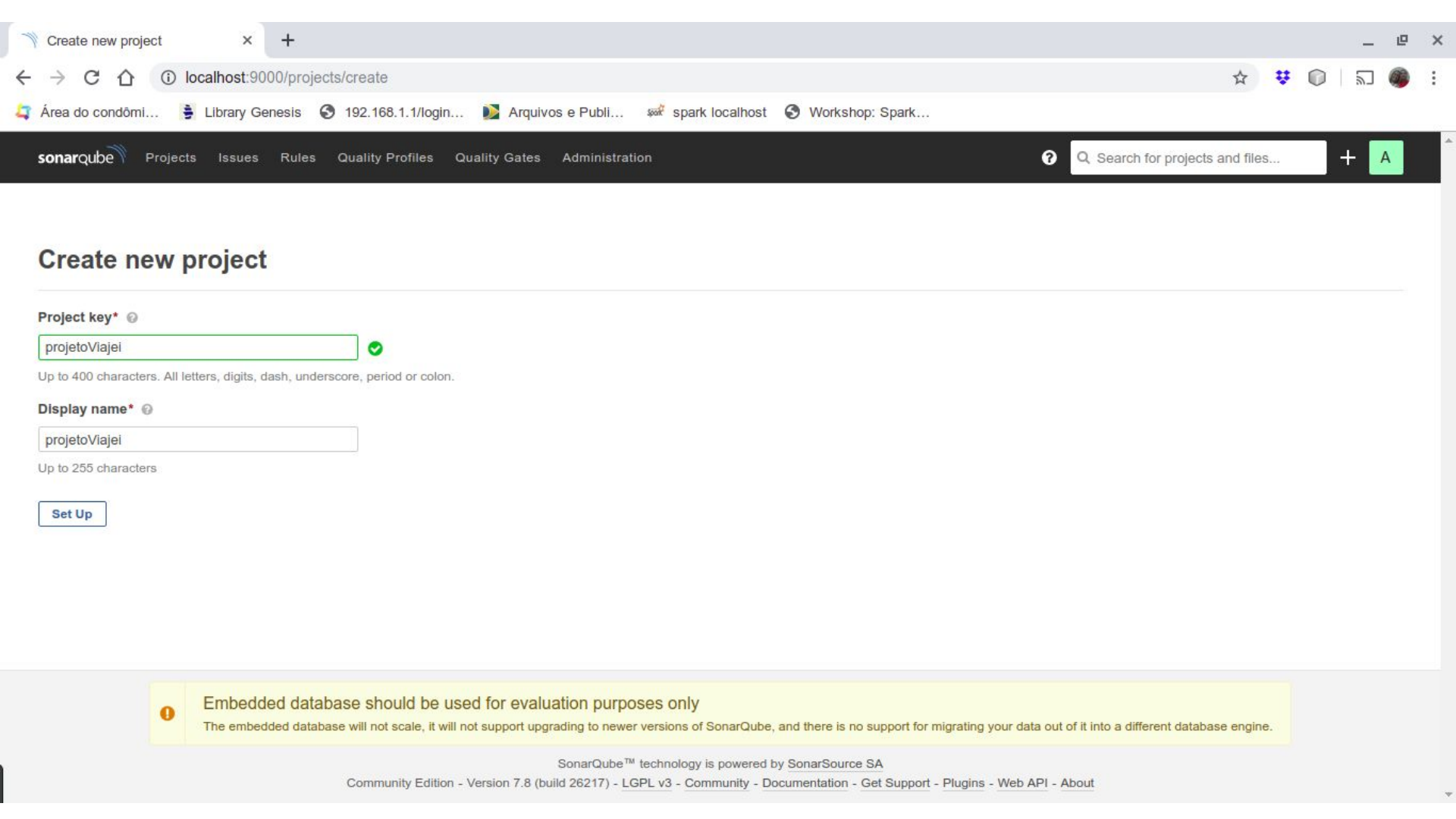
- Projeto open-source para avaliar a qualidade do código
- Arquitetura e design, comentários, regras de linguagem, complexidade, código duplicado, etc



# EXECUTANDO O SONARQUBE

- `http://www.sonarsource.org/`
- `sh sonar-2.14/bin/linux-x86-32/sonar.sh start`
- O sonar será executado, por padrão, na porta :9000





## Create new project

Project key\* ?



Up to 400 characters. All letters, digits, dash, underscore, period or colon.

Display name\* ?

Up to 255 characters

Set Up



### Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

Community Edition - Version 7.8 (build 26217) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About

projetoViajei

localhost:9000/dashboard?id=projetoViajei

Área do condômi...Library Genesis192.168.1.1/login...Arquivos e Publi...spark localhostWorkshop: Spark...

sonarqubeProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

Search for projects and files...

+

A

projetoViajei

master

OverviewIssuesMeasuresCodeActivityAdministration

We initialized your project on SonarQube, now it's up to you to launch analyses!

1

Provide a token

Analyze "projetoViajei":1c11f4af57bccb00100829f914cab0aa981edea2

2

Run analysis on your project

What is your project's main language?

JavaC# or VB.NETOther (JS, TS, Go, Python, PHP, ...)

You are developing primarily in Java: what is your build technology?

MavenGradle

Execute the Scanner for Maven from your computer

Running a SonarQube analysis with Maven is straightforward. You just need to run the following command in your project's folder.

mvn sonar:sonar \-Dsonar.projectKey=projetoViajei \-Dsonar.host.url=http://localhost:9000 \-Dsonar.login=1c11f4af57bccb00100829f914cab0aa981edea2

Copy

No tags

Quality Gate

(Default) Sonar way

Project Key

projetoViajei

Copy

Project Existence | SonarQube x

Analyzing with SonarQube Sc x

Plugin Library - Plugins - Doc x

Analyzing Source Code - Scar x

+

← → ↺ 🏠

https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner+for+Ant

☆ 🍷 📦 | 🖨️ 🌐

🌐 Área do condômi...

📖 Library Genesis

🌐 192.168.1.1/login...


📁 Arquivos e Publi...

🔥 spark localhost

🌐 Workshop: Spark...

☰ sonarqube Espaços ▾

Pesquisar 🔍 ? Autenticação



Scanners

SonarQube

Scanners

- Analyzing with SonarQube Scanner
- Analyzing with SonarQube Scanner for MSBuild
- Analyzing with SonarQube Scanner for Maven
- Analyzing with SonarQube Scanner for Gradle
- **Analyzing with SonarQube Scanner for Ant**
- Analyzing with SonarQube Scanner for Jenkins
- Analyzing with SonarQube Extension for VSTS-TFS

Plugins

Extension Guide

⚙️ Ferramentas do espaço

»

Páginas / Analyzing Source Code

...

# Analyzing with SonarQube Scanner for Ant

Criado por Freddy Mallet, última alteração por Julien Henry em abr 24, 2019

By [SonarSource](#) – GNU LGPL 3 – [Issue Tracker](#) – [Sources](#)

**SonarQube Scanner for Ant 2.6.0.1426** – Compatible with SonarQube 6.7+ (LTS)  
[Download](#)

Table of Contents

- [Features](#)
- [Compatibility](#)
- [Installation](#)
- [Usage](#)
  - [Simple Project](#)
  - [Sample Project](#)
  - [Multi-Module Project](#)
  - [Enable Debug Logs](#)
  - [Advanced Usage](#)

## Features

The SonarQube Scanner for Ant provides a task to allow integration of SonarQube analysis into an Apache Ant build script.

Projetos Arquivos x Serviços

- ▼ Viajei
  - build
  - dist
  - lib
  - nbproject
    - src
    - test
    - web
  - build.xml
- ▼ viajei3
  - .scannerwork
  - build
  - dist
  - lib
  - nbproject
    - src
    - web
  - build.xml

Navegador x

Destinos do Ant

- check-clean
- check-credentials
- clean
- clean-ear
- clean-webinf-lib
- compile
- compile-jsp

build.xml [viajei3] x build.xml [Viajei] x

Código-Fonte

```
1 / <!-- the Compile on Save feature is turned off for the project. -->
2 <!-- You can turn off the Compile on Save (or Deploy on Save) setting -->
3 <!-- in the project's Project Properties dialog box.-->
4
5 <project name="Viajei" default="default" basedir="." xmlns:sonar="antlib:org.sonar.ant">
6   <description>Builds, tests, and runs the project Viajei.</description>
7   <import file="nbproject/build-impl.xml"/>
8
9   <!-- Define the SonarQube global properties (the most usual way is to pass these properties via the command line) -->
10  <property name="sonar.host.url" value="http://localhost:9000" />
11
12  <!-- Define the SonarQube project properties -->
13  <property name="sonar.projectKey" value="org.sonarqube:sonarqube-scanner-ant" />
14  <property name="sonar.projectName" value="VIAJEI" />
15  <property name="sonar.projectVersion" value="1.0" />
16  <property name="sonar.sources" value="src" />
17  <property name="sonar.java.binaries" value="build" />
18  <property name="sonar.java.libraries" value="lib/*.jar" />
19
20  <!-- Define SonarQube Scanner for Ant Target -->
21  <target name="sonar">
22    <taskdef uri="antlib:org.sonar.ant" resource="org/sonar/ant/antlib.xml">
23      <!-- Update the following line, or put the "sonarqube-ant-task-*.jar" file in your "$HOME/.ant/lib" folder -->
24      <classpath path="/home/dibranco/NetBeansProjects/sonar_ant/sonarqube-ant-task-2.6.0.1426.jar" />
25    </taskdef>
26
27    <!-- Execute SonarQube Scanner for Ant Analysis -->
28    <sonar:sonar />
29  </target>
```

project



```
dibranco@Daniel:~/NetBeansProjects/Viajei$ ant sonar
Buildfile: /home/dibranco/NetBeansProjects/Viajei/build.xml
```

```
sonar:
```

```
[sonar:sonar] Apache Ant(TM) version 1.9.9 compiled on July 22 2018
[sonar:sonar] SonarQube Ant Task version: 2.6.0.1426
[sonar:sonar] Loaded from: file:/home/dibranco/NetBeansProjects/sonar_ant/sonarqube-ant-task-2.6.0.1426.jar
[sonar:sonar] User cache: /home/dibranco/.sonar/cache
[sonar:sonar] Default locale: "pt_BR", source code encoding: "UTF-8" (analysis is platform dependent)
[sonar:sonar] SonarQube scanners will require Java 11+ starting on next version
[sonar:sonar] Load global settings
[sonar:sonar] Load global settings (done) | time=55ms
[sonar:sonar] Server id: BF41A1F2-AWua-a5tb_-vFFHwP7VD
[sonar:sonar] User cache: /home/dibranco/.sonar/cache
[sonar:sonar] Load/download plugins
[sonar:sonar] Load plugins index
[sonar:sonar] Load plugins index (done) | time=37ms
[sonar:sonar] Load/download plugins (done) | time=60ms
[sonar:sonar] Process project properties
[sonar:sonar] Execute project builders
[sonar:sonar] Execute project builders (done) | time=4ms
[sonar:sonar] Project key: org.sonarqube:sonarqube-scanner-ant
[sonar:sonar] Base dir: /home/dibranco/NetBeansProjects/Viajei
[sonar:sonar] Working dir: /home/dibranco/NetBeansProjects/Viajei/.scannerwork
[sonar:sonar] Load project settings for component key: 'org.sonarqube:sonarqube-scanner-ant'
[sonar:sonar] Load quality profiles
[sonar:sonar] Load quality profiles (done) | time=66ms
[sonar:sonar] Load active rules
[sonar:sonar] Load active rules (done) | time=3404ms
[sonar:sonar] SCM provider autodetection failed. Please use "sonar.scm.provider" to define SCM of your project, or disable the SCM Sensor in the project settings.
[sonar:sonar] Indexing files...
[sonar:sonar] Project configuration:
[sonar:sonar] 10 files indexed
[sonar:sonar] Quality profile for java: Sonar way
[sonar:sonar] ----- Run sensors on module Projeto VIAJEI1
[sonar:sonar] Load metrics repository
[sonar:sonar] Load metrics repository (done) | time=110ms
[sonar:sonar] Sensor JavaSquidSensor [java]
[sonar:sonar] Configured Java source version (sonar.java.source): none
[sonar:sonar] JavaClasspath initialization
[sonar:sonar] JavaClasspath initialization (done) | time=15ms
[sonar:sonar] JavaTestClasspath initialization
[sonar:sonar] JavaTestClasspath initialization (done) | time=1ms
[sonar:sonar] Java Main Files AST scan
[sonar:sonar] -----
```



Projects

localhost:9000/projects?sort=reliability

Área do condômi...Library Genesis192.168.1.1/login...Arquivos e Publi...spark localhostWorkshop: Spark...

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

Search for projects and files...

+A

My FavoritesAll

Perspective: Overall StatusSort by: ReliabilitySearch by project name or key1 projects

Filters

Quality Gate

Passed1Warning0Failed0

Reliability ( Bugs )

A0B0C0D0E1

Security ( Vulnerabilities )

A0B0C0D0E1

☆ Projeto VIAJE1Passed

Last analysis: June 30, 2019, 9:37 PM

4 Bugs

2 Vulnerabilities

34 Code Smells

0.0% Coverage

0.0% Duplications

398 XSJava

1 of 1 shown

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

Community Edition - Version 7.8 (build 26217) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About

# HTTPS://DOCS.SONARQUBE.ORG/LATEST/



Docs **7.8** ▾

Architecture and Integration

Requirements ▾

Setup and Upgrade ▾

Analyzing Source Code ▾


User Guide ▾


Project Administration ▾

Instance Administration ▾

Extension Guide ▾

Frequently Asked Questions

Scanners 

Plugins 

## Documentation

### Welcome to the new SonarQube documentation

If you already have a SonarQube instance, you should be aware that we've made this documentation available within SonarQube itself (from v7.4). Just look under the "?" in the top menu to be sure you're always looking at the right documentation for *your* version!

SonarQube® is an automatic code review tool to detect bugs, vulnerabilities and code smells in your code. It can integrate with your existing workflow to enable continuous code inspection across your project branches and pull requests.

Below are a few key pointers, otherwise head over to the left pane for full documentation content and search capabilities.

### I'm new to SonarQube

- Installing a [quick demo version](#).
- Installing a [production instance](#).

### I need to set up analysis

- [Analyzing Source Code](#)
- [Branches](#)
- [Pull Requests](#)

# 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

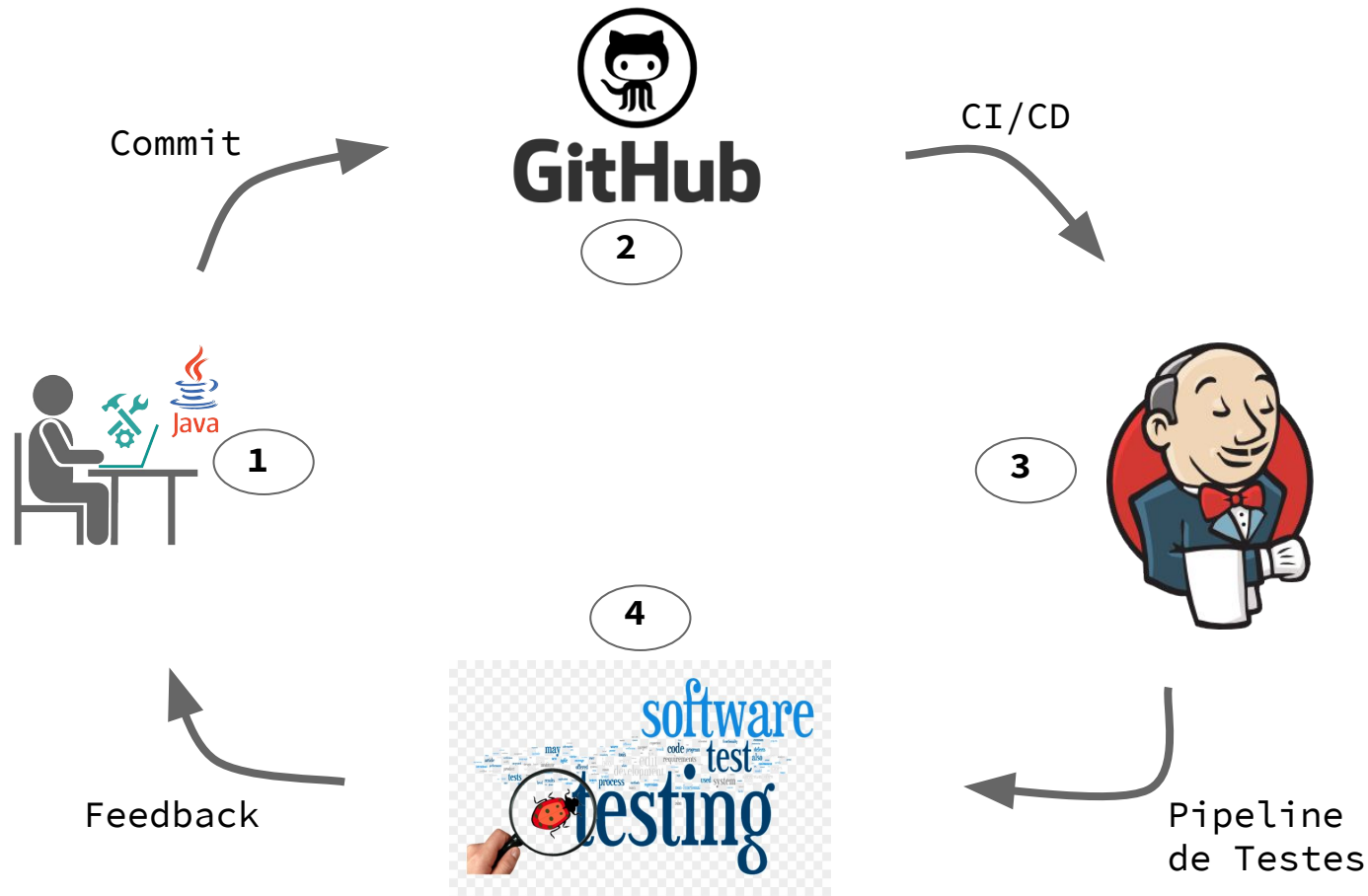
☒ Sonar

## Jenkinsfile:

```
pipeline {
  agent any

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

# FLUXO





OBRIGADO!