



SGB_REL_RelatorioSonar

Versão 1.1

Histórico de Revisões

Nome	Alterações	Data	Versão
Gabriel Vieira	Criação do Documento	14/11/2012	1.0
Bruno Marquete	Formatação do Documento e ajustes de conteúdo	27/11/2012	1.1

Sumário

[1 Introdução](#)

[2 Objetivo do Experimento](#)

[3 O Experimento](#)

[4 Resultados](#)

[5 Descrição dos Resultados](#)

[5.1 Apontamento de Reuso](#)

[5.2 Uso de Bibliotecas](#)

[5.3 Reuso por meio de herança](#)

[6 Considerações Finais](#)

1. Introdução

Este documento apresenta os resultados da experimentação da ferramenta Sonar no quesito reuso de código.

2. Objetivo do Experimento

Testar a ferramenta Sonar a fim de verificar se a mesma oferece suporte para acompanhar a reutilização de código em projetos de software.

3. O Experimento

Para a elaboração do experimento foram adotadas as seguintes etapas:

- a) Instalação do Sonar;
- b) Instalação do Maven;
- c) Criação de um projeto teste;
- d) Execução do projeto teste;
- e) Análise do projeto teste;
- f) Levantamento do resultado.

Obs.: As etapas de instalação de ferramentas fogem do escopo deste relatório. Para orientar na instalação recomenda-se os *links*:

Instalação do Sonar

http://franciscofonseca.com.br/artigos/qos/gerenciando_qos_com_sonar.php

Instalação do Maven

http://maven3tutorial.blogspot.com.br/2011/02/instalando-o-maven-no-seu-computador.html#.UKP0mcWJv_E

Criação de um projeto Maven

http://maven3tutorial.blogspot.com.br/2011/05/maven-em-5-minutos.html#.UKPrhcWJv_E

Para mais informações sobre o Sonar consultar <http://www.sonarsource.org/>.

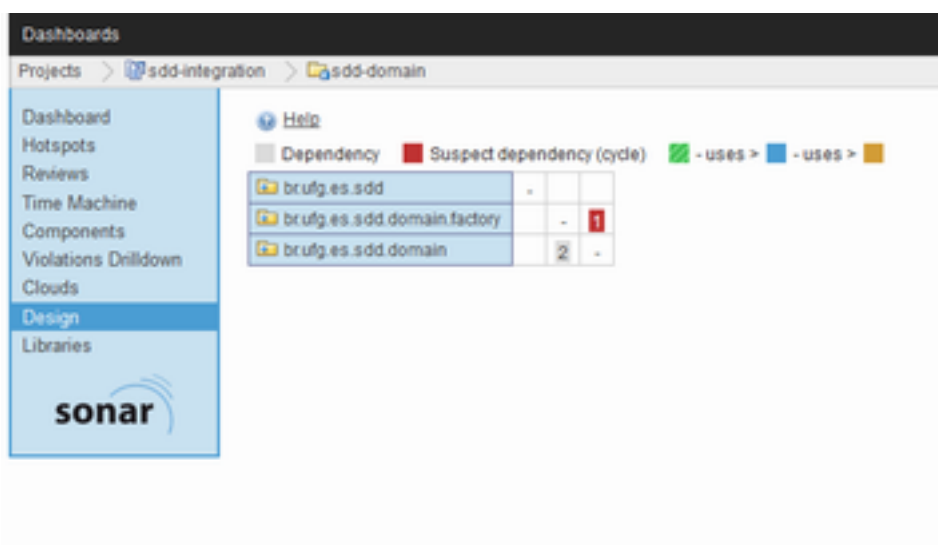
4 Resultados

- a) O Sonar permite o acompanhamento de reutilização de código por meio da análise de Arquitetura e Design. O uso pode ser acompanhado por meio da métrica de quantificação de dependências;
- b) O Sonar verifica dependências entre projetos;
- c) O Sonar verifica dependências de bibliotecas;
- d) O Sonar não verifica reuso de código por meio de herança. Entretanto, se for feita por meio de projetos distintos será apontada.

5 Descrição dos Resultados

5.1 Apontamento de Reuso

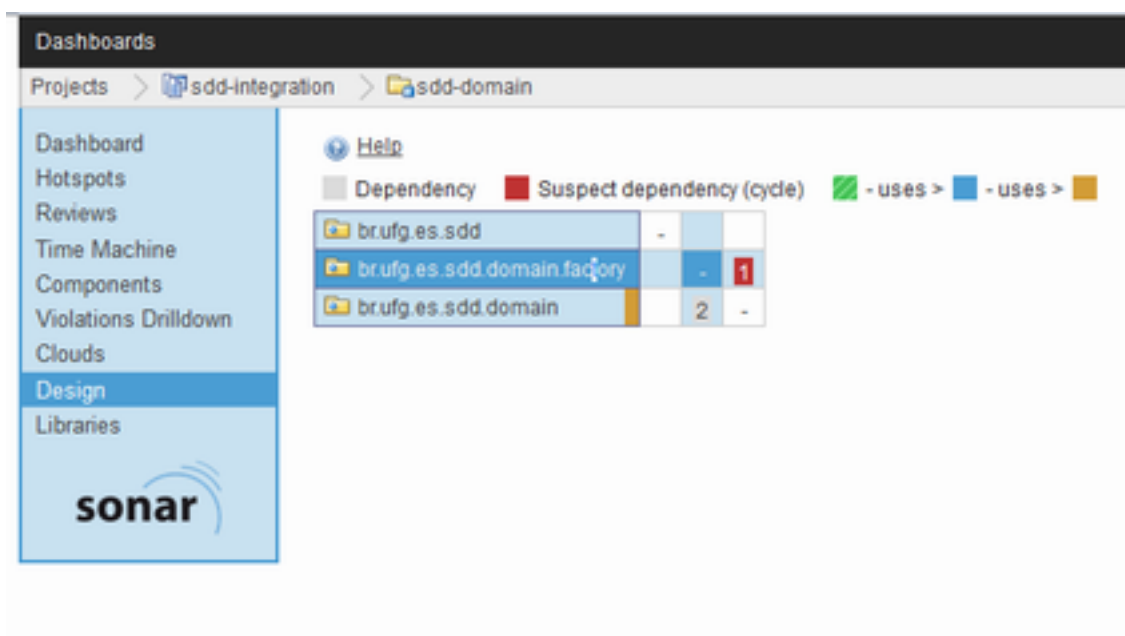
A figura abaixo ilustra um projeto no qual existe dependências entre pacotes. A aba *Design* do Sonar mostra que o pacote `br.ufg.es.sdd.domain.factory` possui 01 dependência. Enquanto o pacote `br.ufg.es.sdd.domain` possui 02 dependências.



Um clique sobre o pacote aponta detalhes da dependência. A figura abaixo exemplifica esta análise. Observa-se que o pacote `br.ufg.es.sdd.domain.factory` possui

uma classe que depende do pacote br.ufg.es.sdd.domain. Essa observação é feita por meio da listra amarela no final de br.ufg.es.sdd.domain. Para maiores esclarecimentos consultar

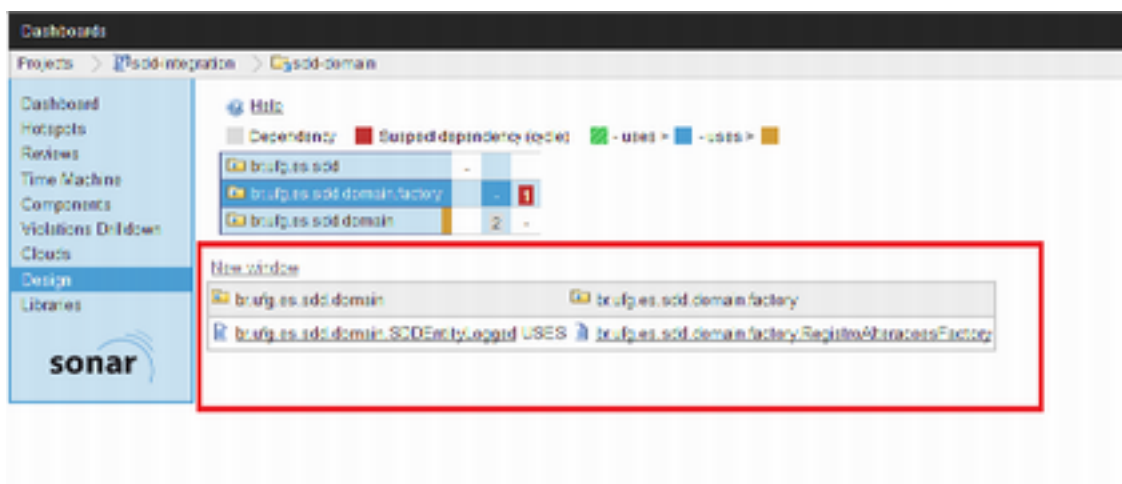
<http://docs.codehaus.org/display/SONAR/Cycles++Dependency+Structure+Matrix>



The screenshot shows the SonarQube interface with the 'Design' tab selected. A sidebar on the left contains navigation links: Dashboard, Hotspots, Reviews, Time Machine, Components, Violations Drilldown, Clouds, Design (selected), and Libraries. The main content area displays a dependency matrix for the 'sdd-domain' project. The matrix has three rows representing packages: 'br.ufg.es.sdd', 'br.ufg.es.sdd.domain.factory', and 'br.ufg.es.sdd.domain'. The columns represent dependencies. A legend indicates that a green square means 'Dependency', a red square means 'Suspect dependency (cycle)', a green square with a checkmark means '- uses >', a blue square with a checkmark means '- uses >', and a yellow square means '- uses >'. The matrix shows that 'br.ufg.es.sdd.domain' has a dependency on 'br.ufg.es.sdd.domain.factory' (indicated by a red square with the number 1) and a dependency on 'br.ufg.es.sdd' (indicated by a yellow square with the number 2).

Package	Dependency	Suspect dependency (cycle)	- uses >	- uses >	- uses >
br.ufg.es.sdd	-	-	-	-	-
br.ufg.es.sdd.domain.factory	-	1	-	-	-
br.ufg.es.sdd.domain	-	-	2	-	-

Um duplo clique sobre o número na tabela apresenta maiores detalhes da dependência, no seguinte formato *Classe 1 USES Classe 2*.



The screenshot shows the SonarQube interface with the 'Design' tab selected. A sidebar on the left contains navigation links: Dashboard, Hotspots, Reviews, Time Machine, Components, Violations Drilldown, Clouds, Design (selected), and Libraries. The main content area displays a dependency matrix for the 'sdd-domain' project. The matrix has three rows representing packages: 'br.ufg.es.sdd', 'br.ufg.es.sdd.domain.factory', and 'br.ufg.es.sdd.domain'. The columns represent dependencies. A legend indicates that a green square means 'Dependency', a red square means 'Suspect dependency (cycle)', a green square with a checkmark means '- uses >', a blue square with a checkmark means '- uses >', and a yellow square means '- uses >'. The matrix shows that 'br.ufg.es.sdd.domain' has a dependency on 'br.ufg.es.sdd.domain.factory' (indicated by a red square with the number 1) and a dependency on 'br.ufg.es.sdd' (indicated by a yellow square with the number 2). A red box highlights the detailed view of the dependency between 'br.ufg.es.sdd.domain' and 'br.ufg.es.sdd.domain.factory', showing the text 'br.ufg.es.sdd.domain.SDDEntityLogged USES br.ufg.es.sdd.domain.factory.RegistroProcessFactory'.

Package	Dependency	Suspect dependency (cycle)	- uses >	- uses >	- uses >
br.ufg.es.sdd	-	-	-	-	-
br.ufg.es.sdd.domain.factory	-	1	-	-	-
br.ufg.es.sdd.domain	-	-	2	-	-

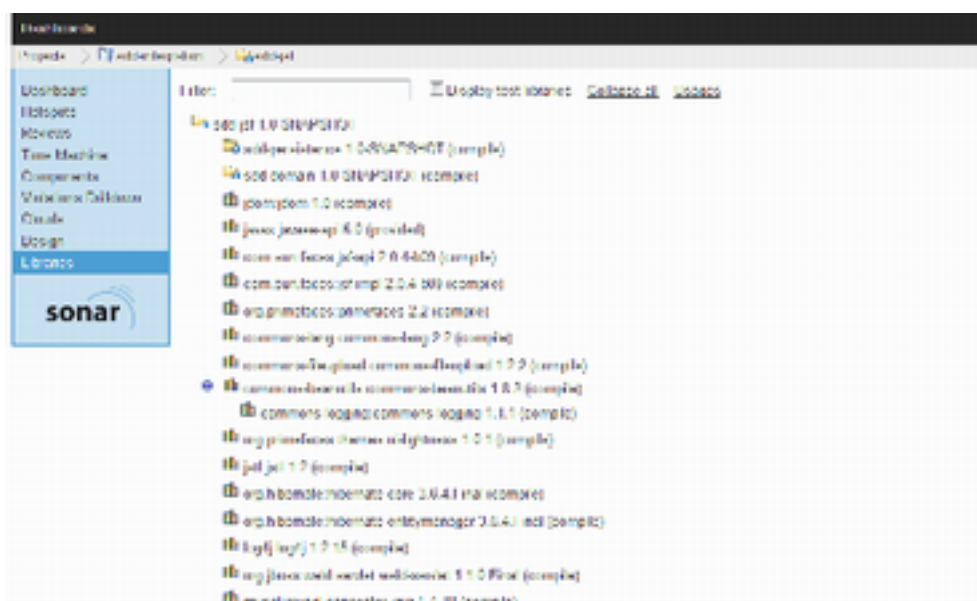
New window

br.ufg.es.sdd.domain br.ufg.es.sdd.domain.factory

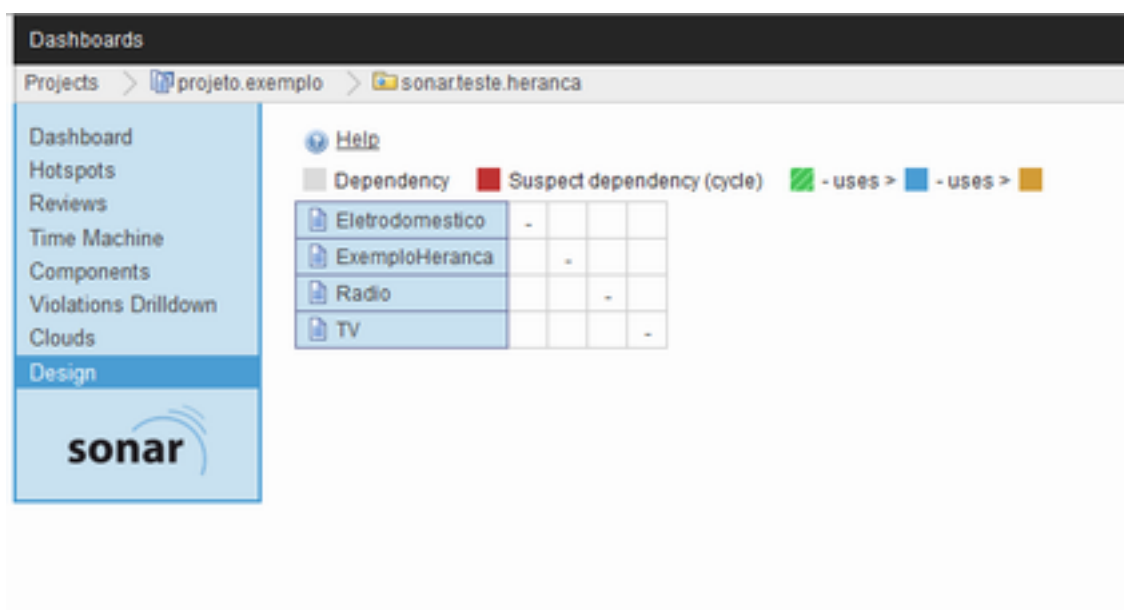
br.ufg.es.sdd.domain.SDDEntityLogged USES br.ufg.es.sdd.domain.factory.RegistroProcessFactory

[illegible]

Por meio da aba *Libraries* é possível verificar as dependências de um projeto com relação a bibliotecas (APIs externas). A figura abaixo mostra as bibliotecas utilizadas no projeto analisado.



Para verificar está possível funcionalidade foi elaborado para este experimento um projeto com uso de herança entre objetos. O projeto possuía 01 classe pai denominada *Eletrdoméstico*, e duas classes filhas *Radio* e *TV*. Ao executar a ferramenta Sonar não houve nenhuma métrica referente ao uso da classe pai pelas classes filhas. A figura abaixo ilustra essa parte do experimento.



Todavia classes de entidade, mapeadas para um determinado banco de dados e herdadas foram apontadas pela ferramenta como dependências. Um código como o abaixo que mostra que um objeto *Docente* estende de uma classe *Usuario* é capturado pela ferramenta como uma dependência entre classes.

```
@Entity
@Table(name = "tb_docentes")
@PrimaryKeyJoinColumn(name = "id")
public class Docente extends Usuario {}
```

6 Considerações Finais

Embora a análise de código oferecida pela ferramenta Sonar não apresente uma métrica de reuso de código para uma política de gestão de reutilização, ainda assim, é possível utilizá-la para tal finalidade por meio da verificação de dependências. A indicação de uma dependência mostra que um determinado código está sendo utilizado em algum outro arquivo. Isto implica que a observância dessa métrica permite rastrear a reutilização de códigos, inclusive com a sua quantificação.

Para uso da ferramenta Sonar faz necessário o uso de uma ferramenta para gerenciamento e automação de projetos em Java, como Maven ou Ant. Observa-se, que na abordagem de uma política de reutilização será necessária a criação de um projeto específico de reuso para futuramente dar-se início a análise de dependências para com este projeto.

Por fim, é importante ressaltar que a ferramenta Sonar oferece diversos recursos que aqui não foram apresentados. Recomendações futuras para teste da ferramenta pode incluir:

- a) Verificar a possibilidade de integrar via plug-in o Sonar com o NetBeans – um bom começo seria: <http://kenai.com/projects/netbeans-sonar-plg/pages/Home>
- b) Verificar os recursos do plug-in Sonar para o eclipse – tutorial de instalação: <http://www.youtube.com/watch?v=H2irgIPE4hw>