

# Relatório de Iniciação Científica

ENGENHARIA DE SOFTWARE

Flávio Airjan Madureira de Alvarenga | Ciência da Computação | August 16, 2013  
Professor Orientador: Mariza Andrade da Silva Bigonha

## Sumário

<b>Índice de Ilustrações.....</b>	<b>3</b>
<b>Introdução .....</b>	<b>4</b>
Fase inicial .....	4
Fase de desenvolvimento .....	4
Fase de análises .....	4
<b>Referências Relacionadas .....</b>	<b>4</b>
Java.....	4
UML.....	4
Padrão de projeto.....	4
Bad Smells .....	4
<b>Objetivos .....</b>	<b>5</b>
<b>Metodologia .....</b>	<b>5</b>
<b>Implementação.....</b>	<b>5</b>
Linguagem .....	5
API .....	5
Funcionalidades.....	5
Janela Inicial .....	5
Menu File .....	6
Menu Edit.....	7
Menu Statistics.....	7
Menu Help .....	8
Janela Edit Metrics .....	8
Janela Edit Metrics 2 .....	9
Janela View Metrics.....	10
Janela View Metrics Table .....	11
Janela View Bad Smells .....	12
Janela Edit Bad Smells Table.....	13
<b>Código .....</b>	<b>14</b>
Pacote Controler.....	14
Pacote Metrics.....	14
Pacote Bad Smell .....	14
Pacote Model .....	15
Pacote Project.....	15
Pacote Export .....	15
<b>Conclusão .....</b>	<b>15</b>

## Índice de Ilustrações:

Figura 1: Janela Inicial .....	1
Figura 2: Menu File .....	1
Figura 3: Menu Edit.....	1
Figura 4: Menu Statistics.....	8
Figura 5: Menu Help.....	1
Figura 6: Janela Edit Metrics .....	9
Figura 7: Edit Metrics 2. ....	1
Figura 8: Janela View Metrics.....	1
Figura 9: View Metric Table .....	1
Figura 10: Janela View Bad Smells.....	1
Figura 11: Janela View Bad Smells Table .....	1

**Resumo:** “Este projeto de iniciação científica tem como propósito desenvolver uma ferramenta de engenharia de software, projeto do mestrando Henrique Gomes Nunes, com o fim de auxiliar os engenheiros de software a detectar falhas de projeto. A partir de uma análise UML é possível prever possíveis problemas de construção de software que podem futuramente impactar o custo do mesmo.”

## 1. Introdução:

### 1.1 Fase inicial

Os dois primeiros meses de iniciação científica foram utilizados como base para o desenvolvimento posterior da ferramenta. Inicialmente foram utilizados como estudo na parte de engenharia de software o artigo da professora Kécia, aonde é desenvolvido a ferramenta Connecta, que tem o mesmo propósito desta ferramenta, porém os resultados são gerados a partir de código fonte.

Para começar a desenvolver a ferramenta foi necessário o estudo da linguagem java e da estrutura de arquivos UML (xmi e xml).

### 1.2 Fase de desenvolvimento

Inicialmente foi desenvolvido um parser para arquivos uml com o objetivo de extrair informações necessárias para o cálculo das métricas para o processamento de Bad Smells. Em seguida foram desenvolvidas as classes para o cálculo das métricas, com essas classes prontas o programa já pode ser testado para análise dos resultados do parser e das métricas. Por fim, foi desenvolvido as telas para que o usuário final tenha facilidade de obter esses dados.

### 1.3 Fase de Análises

//parte ainda em desenvolvimento//

## 2. Referências Relacionadas //não sei se precisa

**Linguagem Java:** “Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é executado por uma máquina virtual.”, [http://pt.wikipedia.org/wiki/Java\\_%28linguagem\\_de\\_programa%C3%A7%C3%A3o%29](http://pt.wikipedia.org/wiki/Java_%28linguagem_de_programa%C3%A7%C3%A3o%29)

**UML:** “UML- Unified Modeling Language é uma linguagem gráfica padrão para a elaboração da estrutura de projetos complexos de software. A UML pode ser empregada para visualizar, especificar, construir e documentar os artefatos de sistema de software.”, <http://www.iweb.com.br/iweb/pdfs/20031008-uml-01.pdf>

**Padrão de Projeto:** “Um padrão descreve um problema que ocorre inúmeras\par vezes em determinado contexto, e descreve ainda a solução para esse problema, de modo que essa solução possa ser utilizada sistematicamente em distintas situações.”, Christopher Alexander, A Pattern Language, Oxford Press, Oxford, R. Unido, 1978.

**Bad Smells:** “Um Bad Smell é uma indicação superficial que geralmente corresponde a um problema mais profundo no sistema”, Martin Fowler.

### **3. Objetivos**

O objetivo deste projeto é desenvolver uma ferramenta que seja capaz de encontrar possíveis problemas de projeto de software durante sua fase de desenvolvimento em diagrama UML. Com a análise do arquivo UML é possível cogitar falhas no projeto que devem ser observadas pelos especialistas, antes mesmo do projeto começar a ser desenvolvido em código fonte.

### **4. Metodologia**

A implementação do software seguiu o padrão de projeto Strategy, em sua grande maioria, com o objetivo de definir várias operações sobre um mesmo objeto. Esta estratégia foi escolhida com o objetivo de deixar os algoritmos de desenvolvimento das métricas e Bad Smells encapsulados, permitindo a sua variação independente da solicitação do cliente.

Para a conexão entre as janelas e os algoritmos das métricas e Bad Smells, foi utilizado um padrão de projeto inspirado no padrão proxy. Esta decisão foi feita para que a parte de implementação das métricas e análise de arquivos uml sejam feitas independente do layout utilizado.

### **5. Implementação**

#### **5.1 Linguagem**

A implementação deste projeto foi feita em linguagem java utilizando o IDE NetBeans.

#### **5.2 API**

Foi utilizado a API JExcelApi para exportar as tabelas com os resultado das análises para o formato Excel. Para efetuar o parser dos arquivos UML foi utilizado a API SAXParser. As duas APIs foram escolhidas por serem leves e de alto desempenho.

#### **5.3 Funcionalidades**

##### **5.3.1 Janela Inicial**

Ao executar o programa a janela inicial vai ser exibida com o título UML Smells. Nesta janela é possível ver os últimos projetos utilizados no programa e os últimos arquivos UML (formato XMI) analisados. Como é visto na imagem a baixo no topo da tela existe 4 menus: File, Edit, Statistics e Help. Estes menus serão mostrados nos próximos tópicos.

Para que o programa saiba quais foram os últimos arquivos utilizados é lido um arquivo (.txt) padrão deste projeto que armazena o endereços dos últimos 10 projetos utilizados e os últimos 10 arquivos UML utilizados.

Ao selecionar um projeto na lista "Latest designs used" e clicar em Load Project os dados do projeto selecionado são carregados no programa. Ao selecionar um arquivo XMI na lista "Latest XMI Files used" e clicar em Load XMI este arquivo UML é carregado no projeto atualmente selecionado.

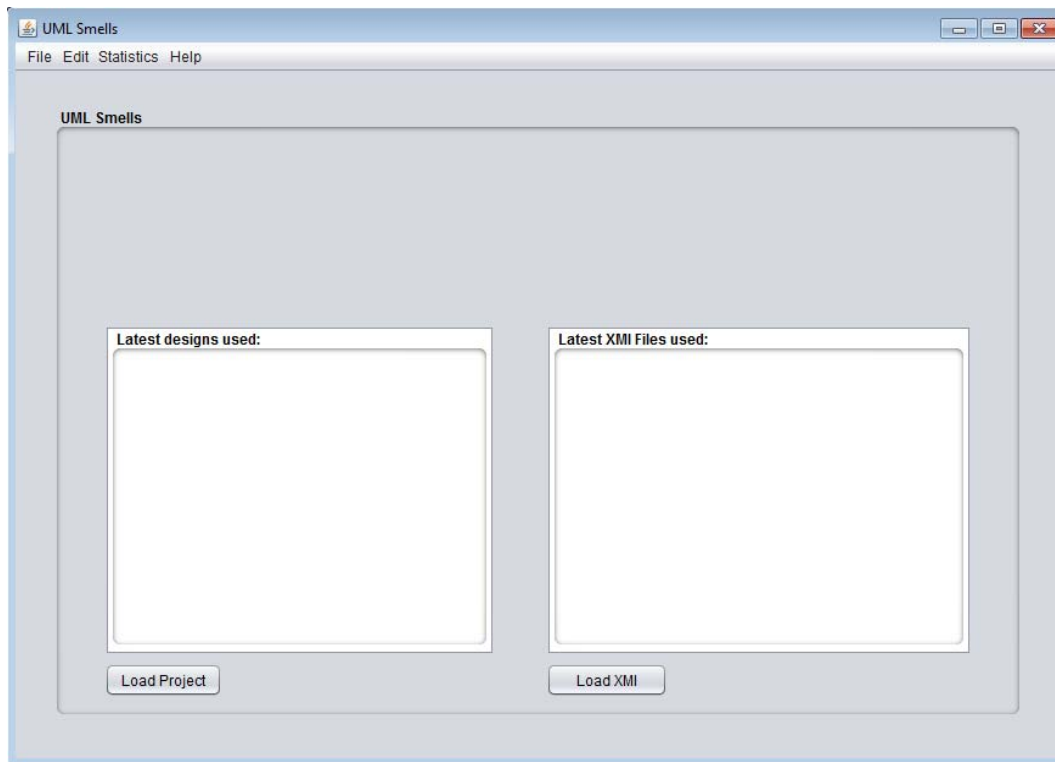


Figura 1: Janela Inicial sem nenhum projeto utilizado anteriormente e nenhum arquivo XMI analisado

### 5.3.2 Menu File

No Menu file se encontram as funcionalidades relativas à arquivos.

A primeira opção deste menu é "New Project". Sua função é criar um novo projeto, criando um arquivo (.txt) que o usuário escolhe o local e nome. Neste arquivo é salvo todos os dados das métricas e dos Bad Smells o nome do projeto (nome do projeto é o mesmo do arquivo) e o último arquivo UML utilizado neste projeto.

A segunda opção é "Open Project". Sua função é abrir projetos já criados pelo usuário e carregar todos os dados armazenados neste arquivo no sistema.

A terceira opção é "Save Project". Sua função é salvar o projeto atualmente utilizado no sistema.

A quarta opção é "Open XMI". Sua função é carregar um arquivo UML no atual projeto do sistema.

A última opção é "Quit". Sua função é finalizar o programa.

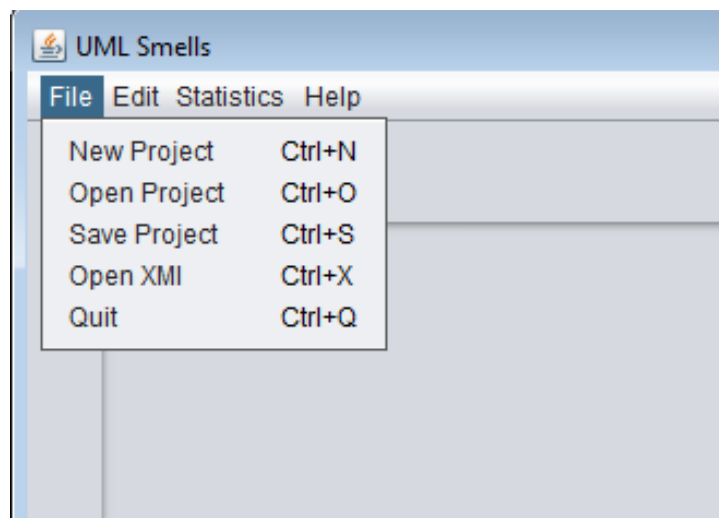


Figura 2: Menu File

### 5.3.3 Menu Edit

No menu Edit se encontram as funcionalidades de editar as configurações das métricas e dos Bad Smells(edição de Bad Smell em construção).

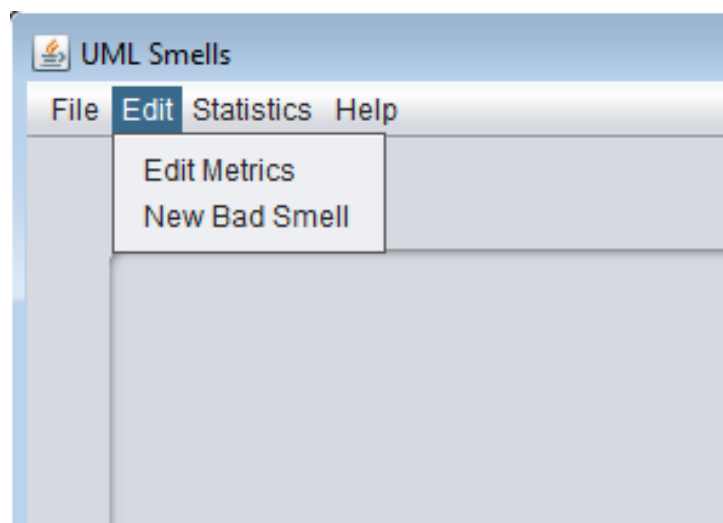
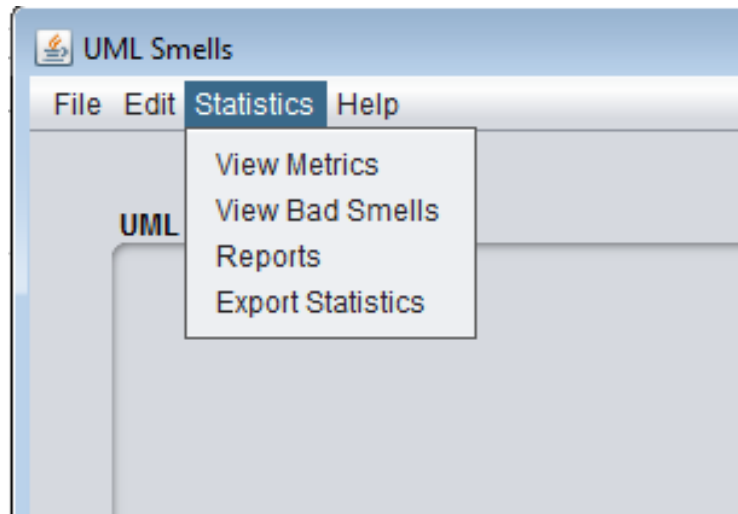


Figura 3: Menu Edit

### 5.3.4 Menu Statistics

No menu Statistics se encontram as funcionalidades de gerar os resultados das análises feitas a



partir do arquivo UML.

A primeira opção deste menu é “View Metric”. Sua função é abrir uma janela de seleção de métricas para que seja feita a análise e gerar uma tabela com os resultados.

A segunda opção é “View Bad Smells”. Sua função é abrir uma janela de seleção de Bad Smells para que seja feita a análise e gerar uma tabela com os resultados.

A terceira opção é “Reports”. Sua função ainda está em construção.

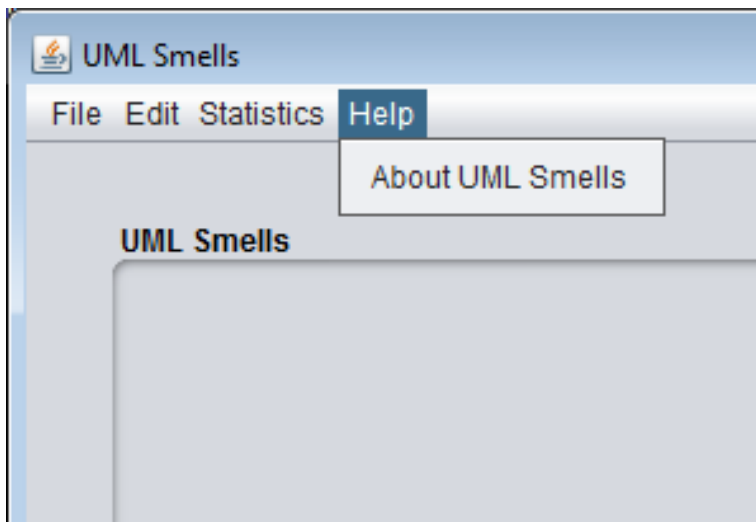
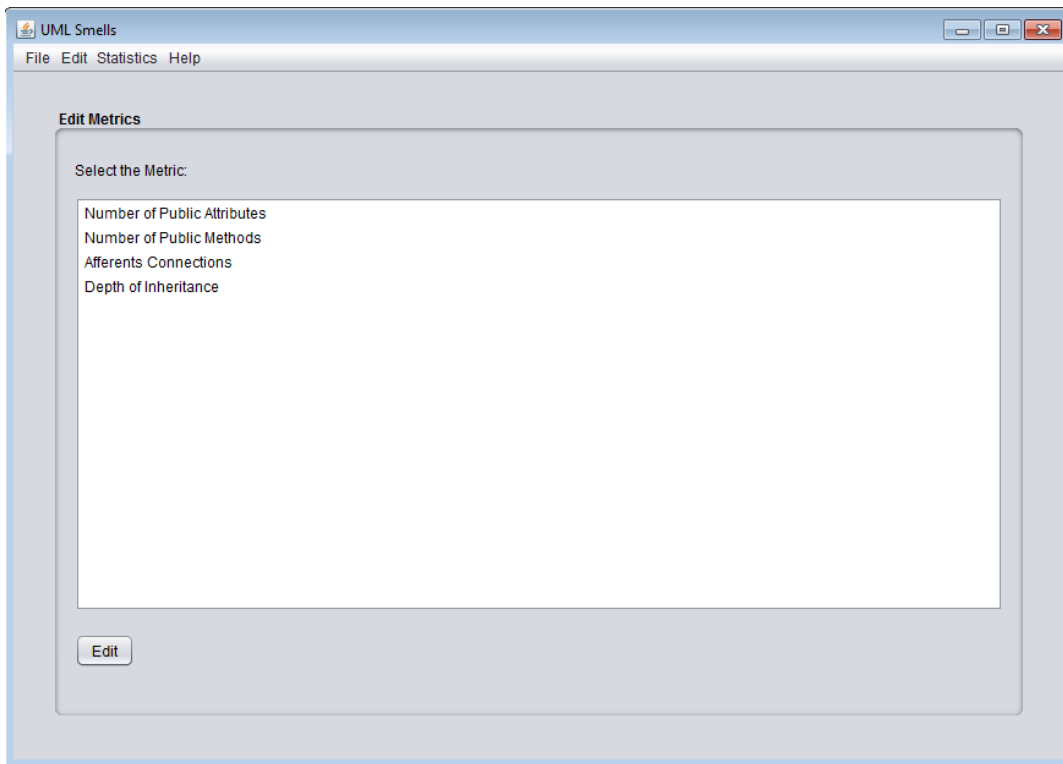
A última opção é “Export Statistics”. Sua função ainda está em construção.

*Figura 4: Menu Statistics*

### 5.3.5 Menu Help

No menu Help se encontra a funcionalidade “About UML Smells” que ainda está em processo de construção.





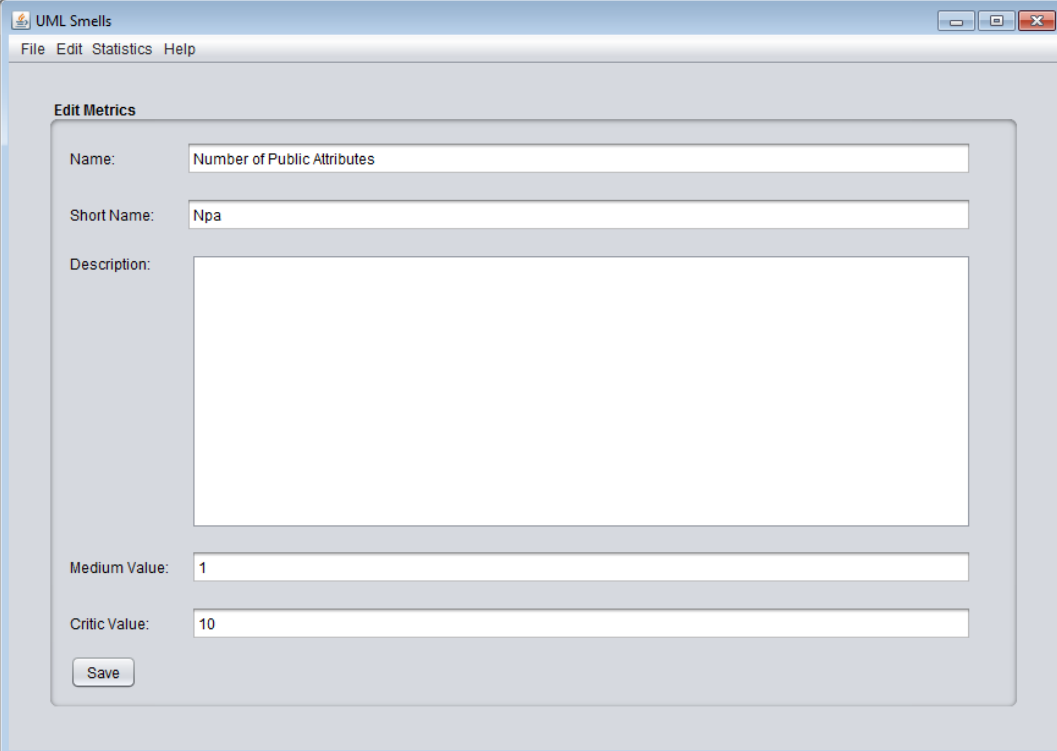
### 5.3.6 Janela Edit Metrics

Para que essa janela seja visualizada é necessário clicar no menu “Edit”, na opção “Edit Metrics”. Esta janela tem a função de dar liberdade ao usuário de editar as Métricas do projeto atual. Para alterar uma métrica basta selecionar a métrica desejada e clicar no botão “Edit”.

Figura 6: Janela Edit Metrics

### 5.3.7 Janela Edit Metrics 2

Ao clicar no botão edit na janela Edit Metrics esta janela será visualizada. Esta janela tem a função de alterar os dados de uma métrica previamente escolhida. Os campos que podem ser alterados são: Name, Short Name, Description, Medium Value e Critic Value.



The screenshot shows a window titled "UML Smells" with a menu bar containing "File", "Edit", "Statistics", and "Help". The main content area is titled "Edit Metrics" and contains the following fields:

- Name: Number of Public Attributes
- Short Name: Npa
- Description: (empty text area)
- Medium Value: 1
- Critic Value: 10

A "Save" button is located at the bottom left of the form.

Figura 7: Edit Metrics 2 com a métrica Number of Public Attributes selecionada.

### 5.3.8 Janela View Metrics

Para que esta janela seja visualizada é necessário clicar no menu "Statistics" na opção "View Metrics". Esta janela tem a função de deixar o usuário escolher quais métricas serão extraídas do arquivo UML e o resultado escrito na tabela da Janela View Metrics Table.

Para extrair os resultados e visualizar a tabela é preciso selecionar as métricas desejadas e clicar no botão "Generate". Abaixo do botão "Generate" encontra-se um barra de load com o propósito de informar a evolução da extração de dados do arquivo UML.

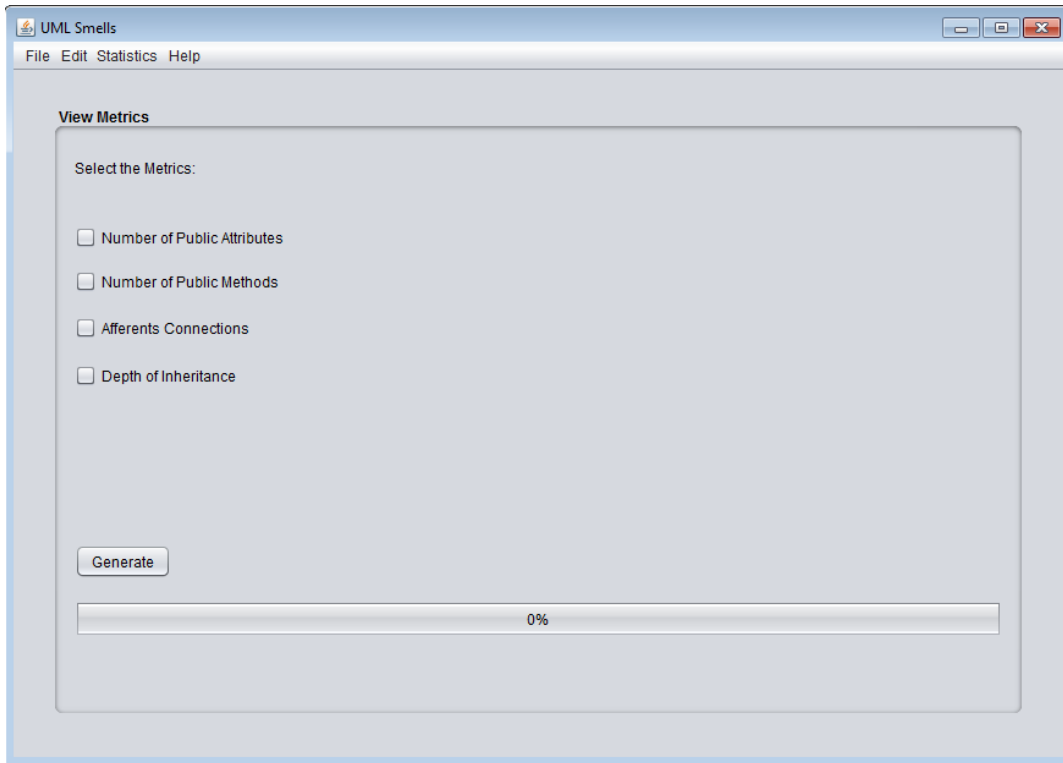


Figura 8: Janela View Metrics

### 5.3.9 Janela View Metrics Table

Ao clicar em “Generate” na janela View Metric esta janela será visualizada. Esta janela tem a função de mostrar os resultados das métricas em uma tabela. Esta tabela é composta de uma classe por linha e uma métrica por coluna.

Para facilitar a visualização dos resultados a tabela recebe uma coloração diferenciada de acordo com o resultado da métrica. A coloração da tabela segue a seguinte ordem:

- Cor vermelha: resultado acima do valor máximo da métrica. Classe com resultado ruim e deve ser analisada pelo especialista.
- Cor amarela: resultado acima do valor médio e abaixo do valor máximo da métrica. Classe com valor regular pode ser analisada pelo especialista se o mesmo achar necessário.
- Cor branca ou cinza: resultado dentro do valor esperado para a métrica analisada. Classe não precisa ser analisada pelo especialista.

The screenshot shows a window titled 'UML Smells' with a menu bar (File, Edit, Statistics, Help) and a 'View Metrics Table' section. The table contains the following data:

Class	Npa	Npm	Cof	Dit
BaseMessaging	0.0	3.0	2.0	0.0
BaseThread	0.0	2.0	0.0	0.0
MainUIMidlet	0.0	4.0	16.0	0.0
AbstractController	0.0	13.0	25.0	0.0
AlbumController	0.0	2.0	1.0	0.0
BaseController	0.0	3.0	2.0	0.0
MediaController	0.0	6.0	1.0	0.0
PasswordHandlerForL...	0.0	2.0	0.0	0.0
PasswordHandlerForVi...	0.0	2.0	0.0	0.0
MediaListController	0.0	4.0	3.0	0.0
MusicPlayController	0.0	4.0	1.0	0.0
PhotoViewController	0.0	4.0	2.0	0.0
PlayVideoController	0.0	4.0	1.0	0.0
ScreenSingleton	0.0	7.0	11.0	0.0
SelectMediaController	0.0	14.0	1.0	0.0
VideoCaptureController	0.0	2.0	1.0	0.0
AlbumData	0.0	14.0	24.0	0.0
ImageAlbumData	0.0	2.0	3.0	0.0
ImageMediaAccessor	0.0	4.0	2.0	0.0
MediaAccessor	0.0	15.0	9.0	0.0
MediaData	0.0	15.0	16.0	0.0

An 'Export' button is located at the bottom right of the table area.

Figura 9: View Metric Table com um exemplo de resultado de análise UML

### 5.3.10 Janela View Bad Smells

Para que esta janela seja visualizada é necessário clicar no menu “Statistics” na opção “View Bad Smells”. Esta janela tem a função de deixar o usuário escolher quais Bad Smells serão extraídas do arquivo UML e o resultado escrito na tabela da Janela View Bad Smells Table.

Para extrair os resultados e visualizar a tabela é preciso selecionar os Bad Smells desejados e clicar no botão “Generate”. Abaixo do botão “Generate” encontra-se um barra de load com o propósito de informar a evolução da extração de dados do arquivo UML.

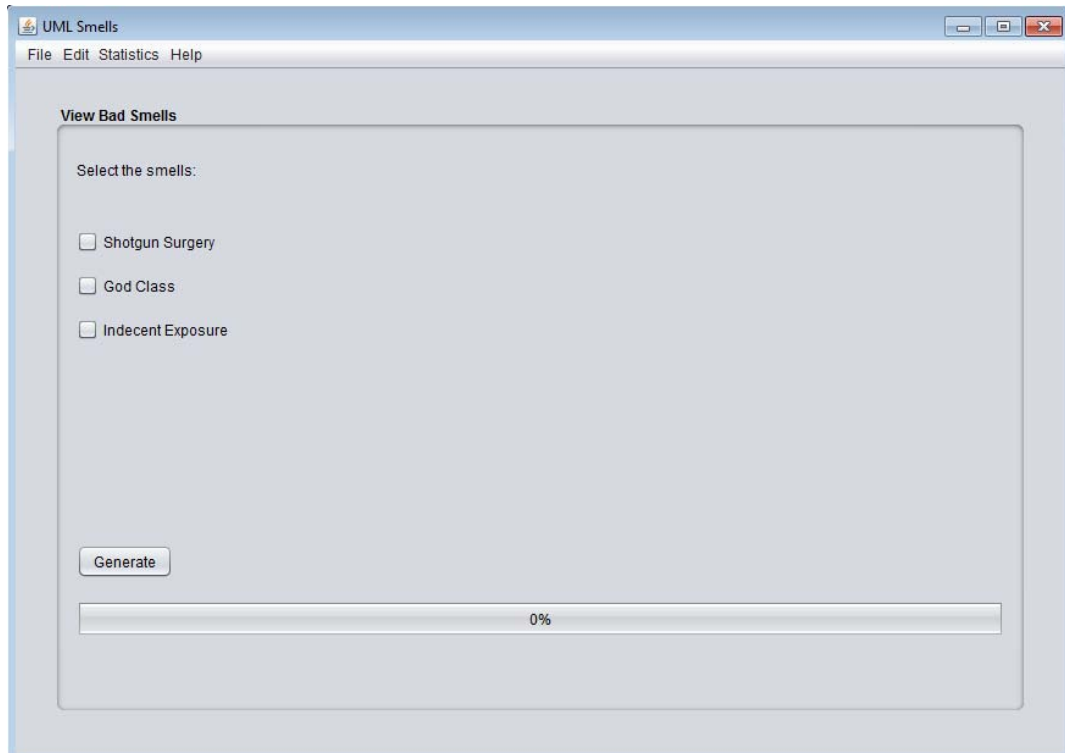


Figura 10: Janela View Bad Smells

### 5.3.11 Janela View Bad Smells Table

Ao clicar em “Generate” na janela View Bad Smells esta janela será visualizada. Esta janela tem a função de mostrar os resultados dos Bad Smells em uma tabela. Esta tabela é composta de uma classe por linha e um Bad Smells por coluna.

Para facilitar a visualização dos resultados a tabela recebe uma coloração diferenciada de acordo com o resultado do Bad Smell. A coloração da tabela segue a seguinte ordem:

- Cor vermelha: resultado está ruim conforme o Bad Smell. Classe com resultado ruim e deve ser analisada pelo especialista.
- Cor amarela: resultado com o valor regular para o Bad Smell. Classe com valor regular pode ser analisada pelo especialista se o mesmo achar necessário.
- Cor branca ou cinza: resultado com o valor bom para o Bad Smell. Classe não precisa ser analisada pelo especialista.

The screenshot shows a window titled 'UML Smells' with a menu bar (File, Edit, Statistics, Help). The main content area is titled 'View Bad Smells Table' and contains a table with the following data:

Class	Shotgun Surgery	God Class	Indecent Exposure
BaseMessaging	Good	Good	Good
BaseThread	Good	Good	Good
MainUIMidlet	Average	Good	Good
AbstractController	Bad	Bad	Average
AlbumController	Good	Good	Good
BaseController	Good	Good	Good
MediaController	Good	Good	Good
PasswordHandlerForLock	Good	Good	Good
PasswordHandlerForView	Good	Good	Good
MediaListController	Average	Good	Good
MusicPlayController	Good	Good	Good
PhotoViewController	Good	Good	Good
PlayVideoController	Good	Good	Good
ScreenSingleton	Average	Good	Good
SelectMediaController	Good	Good	Average
VideoCaptureController	Good	Good	Good
AlbumData	Bad	Bad	Average
ImageAlbumData	Average	Good	Good
ImageMediaAccessor	Good	Good	Good
MediaAccessor	Average	Average	Average
MediaData	Average	Average	Average

An 'Export' button is located at the bottom right of the table area.

Figura 11: Janela View Bad Smells Table com um exemplo de resultado de análise UML

## 5.4 Código

Todo código foi escrito em linguagem Java utilizando o IDE NetBeans. Os pacotes desenvolvidos para o funcionamento deste projeto estão listados a seguir.

### 5.4.1 Pacote controller

Pacote controller tem a função de controlar e extrair todos os dados necessários do arquivo UML. Este pacote faz o uso da API SaxParser para percorrer o arquivo UML e extrair os dados necessários.

### 5.4.2 Pacote Metrics

Pacote Metrics tem a função de implementar todas as métricas utilizadas no projeto. Este pacote contém uma interface Metric e as classes que implementam esta interface.

### 5.4.3 Pacote BadSmells

Pacote BadSmells tem a função de implementar todos os Bad Smells utilizados pelo projeto. Este pacote contém uma interface BadSmells e as classes que implementam esta interface.

#### **5.4.4 Pacote model**

Pacote model tem a função de armazenar os dados extraídos do arquivo UML. O pacote controller faz o uso deste pacote para armazenar os dados extraídos.

#### **5.4.5 Pacote Project**

Pacote Project tem a função de armazenar os dados do projeto atual e fazer a comunicação entre as janelas e a parte de análise e extração das métricas e Bad Smells.

#### **5.4.6 Pacote view**

Pacote view tem a função de gerar as telas de interação com o usuário.

#### **5.4.7 Pacote Export**

Pacote Export tem a função de exportar os resultados obtidos na análise para um formato externo.

### **6. Conclusão**

Neste projeto foi desenvolvido uma ferramenta de engenharia de software com o objetivo de localizar possíveis erros de projeto durante a fase de construção em diagrama UML. Segundo os resultados obtidos pelas análises, a ferramenta funciona de forma a ajudar o especialista a identificar os pontos aonde possíveis problemas de projeto se encontram.

Dentro das limitações das métricas UML, a ferramenta responde de forma simples e eficaz para que o especialista tenha uma maior facilidade de encontrar e solucionar os problemas de um projeto de software.

#### **Referência bibliográfica**

- Harvey M. Deitel. Como Programar Java, 6ª edição. Pearson Education – Br. Brasil, 2005.
- Artigo: Ferreira, Kecia. Avaliação de conectividade em sistemas orientados por objetos. UFMG. Belo Horizonte – MG, 22-06-2006.
- *Wilson de Pádua Paula Filho. Engenharia de Software: Fundamentos, Métodos e Padrões.* LTC Editora. Rio de Janeiro - RJ, 2001.