|  |  |
| --- | --- |
|  | *Code Inspection Report*  *Anti-Spam Configuration Software Development Project*  BSc in LEI  3º Year 2017/2018 - 1º Semester  Software Engineering I  Group 33  73457, Bruno Eugénio, EIC1  72792, Mikie Ribeiro, EIC2  73326, Nuno Carapinha, EIC1  73399, Paulo Franco, EIC1  ISCTE-IUL, Instituto Universitário de Lisboa  1649-026 Lisbon  Portugal  December 15th 2017 |

**Table of Contents**

[Introduction 3](#_Toc498465002)

[Code inspection – Name of the component being inspected 3](#_Toc498465003)

[Code inspection checklist 3](#_Toc498465004)

[Found defects 3](#_Toc498465005)

[Corrective measures 3](#_Toc498465006)

[Conclusions of the inspection process 3](#_Toc498465007)

# Introduction

O software desenvolvido é uma configuração de um filtro Anti-Spam. Neste trabalho, dado uma lista de Spam’s e de Ham’s, o software vai avaliar manualmente ou automaticamente (recorrendo a um algoritmo) qual a configuração ideal para as regras do filtro.

# Code inspection – Name of the component being inspected

*Description of the software component being inspected*

|  |  |
| --- | --- |
| *Meeting date:*  *Meeting duration:*  *Moderator:*  *Producer:*  *Inspector:*  *Recorder:* | *11/12/2017*  *90 minutes*  *Nuno Carapinha*  *Mikie Ribeiro*  *Paulo Franco*  *Bruno Eugénio* |
| *Component name (Package/Class/Method):* | *antiSpamFilter* |
| *Component was compiled:* | *Yes* |
| *Component was executed:* | *Yes* |
| *Component was tested without errors:* | *No* |
| *Testing coverage achieved:* | *82,4%* |

# 

# Code inspection checklist

The checklist for Java code inspection used in this project is available at <http://www.cs.toronto.edu/~sme/CSC444F/handouts/java_checklist.pdf> and in blackboard ES1 page.

# Found defects

Identify and describe found defects, opinions and suggestions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Found defect Id** | **Package, Class, Method, Line** | **Defect category** | **Description** |
| 1 | ActionPerformedEvent methods, AvaliarManual, GravarManual | Implementação | Os métodos das classes que depois eras chamados na classe Gui\_Cliente, quando era executando um botão, tinham atributos desnecessários. |
| 2 | Gui\_Cliente | Código | O JList apenas mostrava 334 regras. |
| 3 | Avaliar\_Manual | Código | Ao executar a avaliação uma segunda vez, os falsos positivos e negativos não voltavam a zero, e assim, eram somados os da primeira com os da segunda avaliação. |

# Corrective measures

1. Criar o método ActionPerformedEvent na classe Gui\_Cliente e chamar o método das classes GravarManual e AvaliarManual dentro desse método.

2. Retirar o “-1” do array.size() no for que adicionava os elementos ao model do JList.

3. Cada vez que voltar a usar a função, inicializar a 0 os Falsos Positivos e Negativos.

# Conclusions of the inspection process

Tentámos inspecionar o melhor possível o software desenvolvido, encontrámos algumas dificuldades, mas foram resolvidas.