

**UMA FERRAMENTA PARA  
RECOMENDAÇÃO DE  
DEMANDAS SIMILARES**



VAGNER CLEMENTINO DOS SANTOS

**UMA FERRAMENTA PARA  
RECOMENDAÇÃO DE  
DEMANDAS SIMILARES**

Proposta de dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: RODOLFO F. RESENDE

Belo Horizonte  
Novembro de 2015



# Lista de Figuras



# Sumário

<b>Lista de Figuras</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Justificativa</b>	<b>3</b>
<b>3 Revisão da Literatura</b>	<b>5</b>
<b>4 Metodologia</b>	<b>7</b>
4.1 Revisão Sistemática da Literatura . . . . .	7
4.2 Construção da Ferramenta . . . . .	8
4.3 Avaliação . . . . .	8
<b>5 Conclusão e Trabalhos Futuros</b>	<b>9</b>
<b>Referências Bibliográficas</b>	<b>11</b>





# Capítulo 1

## Introdução

Dentro do ciclo de vida do produto de software o processo de manutenção tem papel fundamental. Apesar de não ter merecido tanta atenção quanto a parte de projeto e desenvolvimento de software, no últimos anos o processo de manter o software vem ganhando relevância devido, primordialmente, à seu custo associado.

No Capítulo 2 . O Capítulo 3 . No Capítulo 4 é discutida a metodologia a ser aplicada. No Capítulo 5, especialmente na Tabela 5.1 é exibido o cronograma do trabalho.



# Capítulo 2

## Justificativa

- Redução da troca de contexto (context switch)
- Aumento da produtividade
- Trabalhos de recomendação de novas demandas utilizam análise textual através de técnicas de Recuperação da Informação (Information Retrieve - IR)

Dois problemas podem afetar a produtividade do desenvolvedor, e por conseguinte toda a equipe de manutenção, são as demandas duplicadas e a troca de contexto. No primeiro caso, temos um mesmo problema sendo reportado por diversos usuários em demandas distintas. No segundo, a dificuldade analisar duas demandas totalmente distintas uma da outra. Diversos trabalhos vêm sendo realizados no intuito de remover bug duplicados da base de dados dos Sistemas de Controle de Demandas (Issue Tracking System - ITS). Com relação ao problema da troca de contexto, existem na literatura trabalhos com objetivo de recomendar demandas similares, evitando a repetição de algumas tarefas e por consequência a produtividade do desenvolvedor. O problema de recomendar demandas similares pode ser definido formalmente da seguinte forma:

Seja  $I$  o conjunto de demandas de manutenção para um sistema  $S$  qualquer de cardinalidade  $n$ . Seja  $i$  um demanda pertencente ao conjunto  $I$  e que foi atribuída para um desenvolver  $d$ . Pede-se que seja encontrado um subconjunto  $J$  de  $I$ , de tamanho  $k \ll n$ , tal que para todo  $j$  no subconjunto  $J$  seja similar a  $i$  em um grau maior ou igual a  $s$ .



## Capítulo 3

# Revisão da Literatura

No trabalho de Junio et al. [Junio et al., 2011] é proposto um processo denominado PASM (Process for Arranging Software Maintenance Requests) que propõe lidar com tarefas de manutenção como projetos de software. Para tanto, utilizou-se técnicas de análise de agrupamento (clustering) a fim de melhor compreender e comparar as demandas de manutenção. Os resultados demonstraram que depois de adotar PASM os desenvolvedores têm dedicado mais tempo para análise e validação e menos tempo para as tarefas de execução e de codificação.

*NextBug* [Rocha et al., 2015] é uma extensão (plugin) para a ferramenta de Controle de Demanda - Issue Tracking System (ITS) Bugzilla<sup>1</sup> que recomenda novos bugs para um desenvolvedor baseado no bug que ele esteja tratando atualmente. O objetivo da extensão é sugerir bugs com base em técnicas de Recuperação de Informação [Baeza-Yates et al., 1999].

---

<sup>1</sup><https://www.bugzilla.org/>



# Capítulo 4

## Metodologia

O processo de desenvolvimento deste trabalho pode ser dividido nas seguintes etapas *I - Revisão Sistemática da Literatura*; *II - Construção da Ferramenta (Prova de Conceito)*; *IV - Avaliação*. Cada uma das etapas é detalhada nas próximas seções.

### 4.1 Revisão Sistemática da Literatura

Uma *Revisão Sistemática da Literatura* - SLR (do inglês Systematic Literature Review) é uma metodologia científica cujo objetivo é identificar, avaliar e interpretar *toda* pesquisa *relevante* sobre uma questão de pesquisa, área ou fenômeno de interesse [Keele, 2007, Wohlin et al., 2012]. Neste trabalho será utilizada a diretriz proposta [Keele, 2007] no qual uma Revisão Sistemática deve seguir os seguintes passos:

#### 1. Planejamento

- a) *Identificar a necessidade da Revisão*
- b) *Especificar questões de pesquisa*
- c) *Desenvolver o Protocolo da Revisão*

#### 2. Condução/Execução

- a) *Seleção dos Estudos Primários*
- b) *Análise da qualidade dos Estudos Primários*
- c) *Extração dos Dados*
- d) *Sintetização dos Dados*

#### 3. Escrita/Publicação

- a) *Redigir documento com os resultados da Revisão*
- b) *Redigir documento com lições aprendidas*

A SRL proposta segue as diretrizes por [Keele, 2007] e visa responder as seguintes questões de pesquisa:

- Q1: Quais são as metodologias/técnicas utilizadas para recomendação de novas demandas?
- Q2: Quais são as metodologias/técnicas utilizadas para remoção de bugs duplicados?
- Q3: Qual procedimento utilizado para avaliar a técnica/metodologia proposta?
- Q4: Em qual tipo de projeto (comercial ou código-aberto) a metodologia/técnica proposto foi aplicada?

## 4.2 Construção da Ferramenta

Como prova de conceito da técnica proposta será construída uma ferramenta que possibilite a recomendação de novas demandas a serem tratadas baseada em uma “demanda-semente”. Com intuito de reduzir o problema da troca de contexto e ao mesmo aumentar a produtividade de equipes de manutenção de software é proposto uma ferramenta para recomendação de demandas similares utilizando Redes Neurais.

## 4.3 Avaliação

Com o objetivo de avaliar a ferramentas proposto nes trebablho será realizado um *Quasi-Experimento* [Wohlin et al., 2012] utilizando a base de dados de demandas de manutenção de uma empresa de software real. Dado que uma demanda  $i$  foi atribuída a um desenvolvedor  $d$ , serão geradas 03 listas de sugestões: uma lista proposta pelo gerente imediato de  $d$ ; outra proposta por um desenvolver do mesmo setor de  $d$ ; e a terceira gerada por nossa ferramenta. Naturalmente o desenvolvedor não saberá a origem de nenhum das listas. De posse das três listas pediremos ao desenvolvedor  $d$  que informe qual delas pode reduzir a troca de contexto e aumentar sua produtividade.



## Capítulo 5

# Conclusão e Trabalhos Futuros

Para tanto, a tabela 5.1 descreve as atividades que serão realizadas para atingir este objetivo.

#	Atividade	Início (MM/AAAA)	Término (MM/AAAA)
01	Revisão da Literatura	10/2015	11/2015
02	Ponto de Controle 01 – Reunião com orientador sobre Revisão da Literatura	12/2016	12/2016
03	Avaliação da Técnica de Rede Neural	01/2016	01/2016
04	Ponto de Controle 02 – Reunião com orientador sobre a Técnica de Rede Neural	02/2016	02/2016
05	Implementação da Ferramenta	02/2016	04/2016
06	Ponto de Controle 03 – Avaliação da Ferramenta Avaliada	05/2016	05/2016
07	Experimento de Avaliação da Ferramenta	05/2016	05/2016
08	Ponto de Controle 04 – Avaliação do Experimento junto com o orientador	05/2016	05/2016
09	Finalização do texto da dissertação	06/2016	07/2016
10	Ponto de Controle 05 – Avaliação do texto da dissertação com o orientador	07/2016	07/2016
11	Defesa da dissertação	07/2016	07/2016

**Tabela 5.1.** Cronograma de execução do trabalho



# Referências Bibliográficas

- [Baeza-Yates et al., 1999] Baeza-Yates, R.; Ribeiro-Neto, B. et al. (1999). *Modern information retrieval*, volume 463. ACM press New York.
- [Junio et al., 2011] Junio, G.; Malta, M.; de Almeida Mossri, H.; Marques-Neto, H. & Valente, M. (2011). On the benefits of planning and grouping software maintenance requests. Em *Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on*, pp. 55–64. ISSN 1534-5351.
- [Keele, 2007] Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Em *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*.
- [Rocha et al., 2015] Rocha, H.; Oliveira, G.; Marques-Neto, H. & Valente, M. (2015). Nextbug: a bugzilla extension for recommending similar bugs. *Journal of Software Engineering Research and Development*, 3(1).
- [Wohlin et al., 2012] Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M. C.; Regnell, B. & Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.