



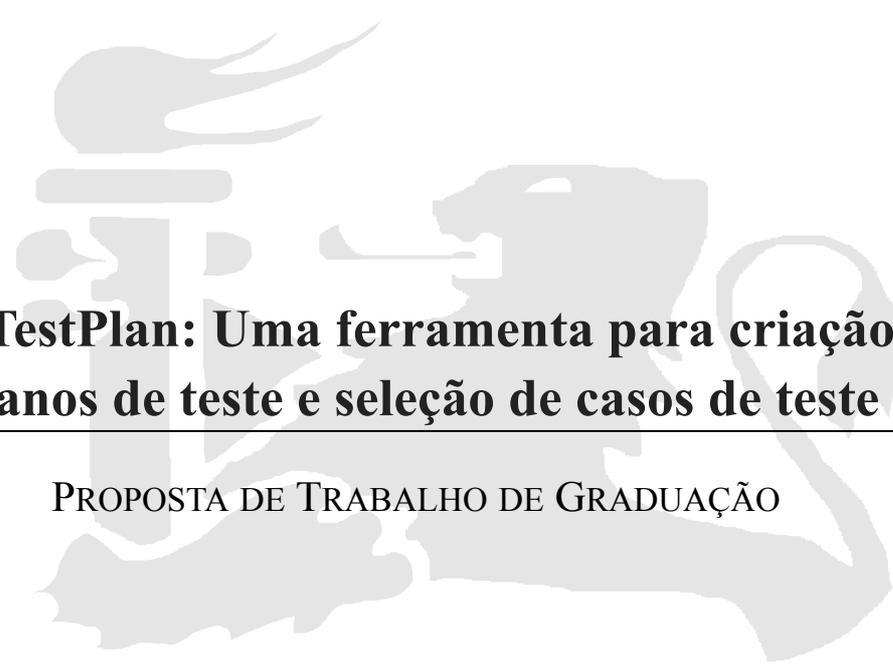
UNIVERSIDADE FEDERAL DE  
PERNAMBUCO

GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

CENTRO DE INFORMÁTICA

2016.1

---



**AutoTestPlan: Uma ferramenta para criação de  
planos de teste e seleção de casos de teste**

---

PROPOSTA DE TRABALHO DE GRADUAÇÃO

**Aluno:** Claudio José Antunes Salgueiro Magalhães (cjas@cin.ufpe.br)  
**Orientador:** Alexandre Cabral Mota (acm@cin.ufpe.br)

Recife, 04 de Abril de 2016.

## Sumário

1. Introdução .....	3
2. Objetivos.....	3
3. Cronograma.....	4
4. Possíveis avaliadores.....	4
5. Referências .....	5
6. Assinaturas.....	5

## 1. Introdução

Atualmente é comum ocorrer mudanças nos software, algumas mudanças envolvem correções de falhas enquanto outras melhorias de funcionalidade. Essas mudanças são documentadas através de *Change Requests* (ou simplesmente CR). Como sabemos nenhum software está livre de falhas, e as alterações feitas podem introduzir mais delas, por isso é importante testar o sistema periodicamente usando diferentes formas e abordagens. Uma delas relata as mudanças através de versões consecutivas do sistema, ela é o teste de regressão. Nele executamos certos testes para garantir que as mudanças não introduziram problemas.

Com o crescimento do software e suas funcionalidades, a quantidade de casos de testes também aumenta. Sabemos que uma opção pode ser reexecutar todos os testes, mas isto é muito custoso na prática. Para isso o teste de regressão tenta aproveitar a oportunidade de focar em mudanças que ocorreram em um certo período de tempo. Ou seja, podemos obter as CRs que correspondem a determinado período de tempo e reexecutar apenas os casos de testes mais apropriados.

Quando o código é disponibilizado, a seleção pode ser feita com precisão e segurança [1]. Porém no nosso contexto, os casos de testes são executados manualmente ( Este é o caso da nossa industria parceira: Motorola Mobility) Mesmo com os casos de teste escolhidos levando em consideração as CRs ainda podem ter muitos. Neste caso, consideraremos critérios de seleção baseado na recuperação de informação[2] para conseguir priorizar os casos de testes que devem ser executados.

## 2. Objetivos

Este trabalho tem como objetivo principal avaliar a viabilidade do uso do PyLucene como priorizador ou ordenador de casos de testes para a criação de um plano de testes . Para tal, iremos utilizar a AutoTestPlan, uma ferramenta web utilizada para criação de planos de teste e escolhas de casos de teste. Ela foi contruída em Django, um framework web de python, e integrada com JIRA, ferramenta para acompanhamento de planos e *issues*

O PyLucene é uma versão em python da ferramenta Lucene. O Lucene é uma biblioteca de engenho de busca. Ela consegue ser facilmente integrada com Django, principalmente por sua versão em python e é adequada para fazer buscas através de

textos. Neste trabalho visamos utilizar o processamento de textos de CRs(Solicitação de Mudança) para buscar casos de testes e prioriza-los.

Este trabalho propõe um proceso de seleção baseado em recuperação de informação, especificamente com o uso do Apache Lucene framework. Esta é implementada na ferramenta AutoTestPlan, que está sendo usada pelo nosso parceiro industrial. O desenvolvimento desta ferramenta foi dividido em duas fases principais: (i) Automatização de várias atividades manuais realizadas (representa 75% de melhoria do tempo de trabalho); (ii) Implementação do método de seleção (representando os 25% restantes). Nosso objetivo é reduzir o tempo gasto com o trabalho em (i) e diminuir o esforço do trabalho em (ii), pois o arquiteto de testes terá apenas que validar a seleção proposta.

### 3. Cronograma

Nessa seção, a tabela 1 descreve o cronograma de atividades previsto para desenvolvimento desse trabalho de graduação ao longo desse semestre.

Atividade	Março	Abril	Maió	Junho	Julho
Elaboração da Proposta	■				
	■	■			
		■	■		
Pesquisa bibliográfica		■	■	■	
Testes e experimentos			■	■	
Análise dos resultados			■	■	
Elaboração do relatório				■	■
Preparação para apresentação e defesa					■

Tabela 1- Cronograma de Trabalho de Graduação

### 4. Possíveis avaliadores

Os possíveis avaliadores para o resultado a ser obtido no final de todas as etapas descritas nesse documento são:

- Juliano Manabu Iyoda;
- Flávia de Almeida Barros

## 5. Referências

- [1] Greg Rothermel and Mary Jean Harrold. A safe, efficient regression test selection techniques. *ACM Trans. Softw. Eng. Methodol*, 6(2):173-210, April 1997.
- [2] Rippon K. Saha, Lingming Zhang, Sarfraz Khurshid, and Dewayne E. Perry. An information retrieval approach for regression test prioritization based on program changes. In *Proceedings of the 37<sup>th</sup> International Conference on Software Engineering – Volume 1, ICSE'15*, pages 268-279. IEEE Press, 2015
- [3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
- [4] Apache. **Lucene**. <https://lucene.apache.org/core/>. Acesso 27 mar. 2016.
- [5] Atlassian. **JIRA**. <https://www.atlassian.com/software/jira>. Acesso 27 mar. 2016.

## 6. Assinaturas

---

Claudio José Antunes Salgueiro Magalhães  
**Orientando**

---

Alexandre Cabral Mota  
**Orientador**