

UNIVERSIDADE FEDERAL DE OURO PRETO – UFOP

CIÊNCIA DA COMPUTAÇÃO



METODOLOGIA CIENTÍFICA

ATIVIDADE I

Marcus Vinícius Souza Fernandes

Ouro Preto

2021

Artigo 1:

- **Contexto:** Ocorrência de falhas durante o desenvolvimento de software e o alto gasto de tempo/energia para identificar a origem destas.
- **Objetivo:** Análise de um mecanismo para auxiliar no processo de identificação de falhas e seus respectivos trechos de código relacionados.
- **Revisão da literatura:** Há diversas citações neste arquivo, dentre elas podemos ressaltar “Brown, A., and Patterson, D. A. (2001)”, “Akhtar, N. and Missen, M. M. S. (2014)” e “Araújo, T., Wanderley, C. and von Staa, A. (2012)”.
- **Metodologia:** As configurações do experimento estão um pouco confusas mas segundo meu entendimento, se resumem ao modelo e demonstração do uso de assertivas executáveis.
- **Resultados:** Os resultados obtidos foram dispostos em uma tabela.

Estrutura de dados	Método	Classe	Total	Equivalentes	Morto	Vivo	EM
AA Tree	133	56	189	2	189	0	1,0
AVL Tree	139	16	155	6	155	0	1,0
Binary Heap	191	2	193	1	193	0	1,0
Binary Search Tree	50	5	55	1	55	0	1,0
Binomial Queue	225	7	232	0	232	0	1,0
Black Red Tree	88	88	176	5	176	0	1,0
BTree	1582	30	1612	16	1612	0	1,0
Deterministic Skip List	32	40	72	0	72	0	1,0
Fibonacci Heap	167	39	206	0	206	0	1,0
Leftist Heap	32	6	38	0	38	0	1,0
Linked List	173	87	260	12	260	0	1,0
Pair Heap	203	87	290	1	290	0	1,0
Spaly Tree	54	142	196	3	196	0	1,0
Treap	72	23	95	1	95	0	1,0

Artigo 2:

- **Contexto:** O impacto positivo de conhecer ferramentas/tecnologias de boa e fácil manutenção .
- **Objetivo:** Avaliar a manutenibilidade de três LPSs equivalentes implementadas com Programação Orientada a Características utilizando AHEAD.
- **Revisão da literatura:** Há diversas citações neste arquivo, dentre elas podemos ressaltar “Abílio, R.; Teles, P.; Costa, H.; Figueiredo, E. (2012)”, “Aldekoa, G.; Trujillo, S.; Sagardui, G.; Diaz, O. (2008)” e “Ali, M.; Babar, M. A.; Chen, L.; Stol, K.-J. (2010)”.
- **Metodologia:** Foram utilizadas seis medidas para avaliar quatro atributos (tamanho, coesão, acoplamento e complexidade).

- **Resultados:** Houve uma diferença significativa na comparação, apenas para uma das seis medidas utilizadas no qual a manutenibilidade das LPSs desenvolvidas com AspectJ e DeltaJ são consideradas a mesma, no entanto, melhores que a LPS em AHEAD.

Artigo 3:

- **Contexto:** A depreciação de API's e a ausência de mensagens de substituição para informar os usuários/clientes.
- **Objetivo:** Propor uma maneira para a recomendação de mensagens ausentes em API's depreciadas.
- **Revisão da literatura:** Há diversas citações neste arquivo, dentre elas podemos ressaltar "Agarwal, R. and Srikant, R. (1994)", "Bogart, C., Kastner, C., Herbsleb, J., and Thung, F. (2016)" e "Brito, A., Xavier, L., Hora, A., and Valente, M. T. (2018)".
- **Metodologia:** Analisar o histórico de versões de uma centena de sistemas clients e o framework Android.
- **Resultados:** Foi notório que algumas mensagens são visíveis porém são rasas, diante disso é reforçado que melhorias ocorram neste contexto.

Artigo 4:

- **Contexto:** Visando o reúso sistemático e em larga escala, a capacidade de manutenção da LPS deve ser elevada, pois a mudança em um módulo pode impactar em vários outros produtos.
- **Objetivo:** Avaliar a manutenibilidade de duas Linhas de Produtos de Software equivalentes implementadas com Programação Orientada a Características utilizando AHEAD e com Programação Orientada a Aspectos utilizando AspectJ
- **Revisão da literatura:** Há diversas citações neste arquivo, dentre elas podemos ressaltar "Aldekoa, G.; Trujillo, S.; Sagardui, G.; Diaz, O. (2008)", "Apel, S.; Beyer, D. (2011)" e "Batory, D. (2003)".
- **Metodologia:** Não consegui identificar de forma clara.
- **Resultados:** A LPS implementada em AspectJ obteve melhores resultados de manutenibilidade e se destacou no estudo.

Artigo 5:

- **Contexto:** Devido o avanço das tecnologias, o mercado de desenvolvimento de software tornou-se mais competitivo e exigente para atender às novas demandas e abordagens no desenvolvimento de software. Pode-se destacar a Linha de Produtos de Software (LPS).
- **Objetivo:** Identificar alguns critérios de caracterização através de técnicas de extração de LPS.
- **Revisão da literatura:** Há diversas citações neste arquivo, dentre elas podemos ressaltar “Ahmed, F.; Campbell, P.; Lagharid, M. S. (2009)”, “Apel, S.; Leich, T.; Saake, G. (2008)” e “Batory, D. (2004)”.
- **Metodologia:** Fazer o uso de dois tipos de técnicas de extração: baseadas em anotações e baseadas em composição.
- **Resultados:** Os resultados obtidos foram dispostos em uma tabela de comparações.

Técnicas Critérios	Baseadas em Anotação		Baseadas em Composição		
	Compilação Condicional	Coloração de Código	Programação Orientada a Características	Programação Orientada a Aspectos	Módulos de Características Aspectuais
Baseia-se em atividades	<i>Não atende</i>	<i>Atende parcialmente</i>	<i>Não atende</i>	<i>Não atende</i>	<i>Não atende</i>
	Essa técnica está baseada principalmente em anotação de código	A técnica possui algumas atividades, porém está baseada principalmente em coloração de código.	A técnica está baseada principalmente em extração de módulos de características.	A técnica está baseada principalmente em extração de módulos de interesses transversais.	A técnica está baseada principalmente em extração de módulos de características.
Permite coloração em Código	<i>Não atende</i>	<i>Atende</i>	<i>Não atende</i>	<i>Não atende</i>	<i>Não atende</i>
	As ferramentas que auxiliam na extração não suportam coloração.	---	Como são extraídos módulos, não se faz necessária coloração em código.	Como são extraídos módulos, não se faz necessária coloração em código.	Módulos são extraídos utilizando ferramentas que não suportam coloração.
Suporta criação de módulos	<i>Não atende</i>	<i>Não atende</i>	<i>Atende</i>	<i>Atende</i>	<i>Atende</i>
	A anotação é feita no próprio código fonte.	As características do código fonte são coloridas sem que haja modularização.	---	---	---
Utiliza diretivas de pré-processamento	<i>Atende</i>	<i>Atende parcialmente</i>	<i>Não atende</i>	<i>Não atende</i>	<i>Não atende</i>
	---	---	Como módulos são extraídos, não se faz necessária a utilização de diretivas de pré-processamento.	Como módulos são extraídos, não se faz necessária a utilização de diretivas de pré-processamento.	Como módulos são extraídos, não se faz necessária a utilização de diretivas de pré-processamento.

Artigo 6:

- **Contexto:** A configuração de LPS, dentre as milhares de combinações de características possíveis, tem se mostrado inviável mesmo para pequenos projetos.
- **Objetivo:** Propor um modelo de LPS baseado em algoritmos de busca e otimização.
- **Revisão da literatura:** Há diversas citações neste arquivo, dentre elas podemos ressaltar “Batory, D. (2005)”, “Batory, D., Sarvela, J., Rauschmayer. (2004)” e “Benavides, D. et al. (2006)”.
- **Metodologia:** O modelo citado foi implementado utilizando duas soluções: algoritmos de enumeração exaustiva com pré processamento e backtracking e heurística gulosa.
- **Resultados:** Temos que os experimentos realizados por enumeração exaustiva resultaram que ela é inviável para grandes instâncias do problema. Por outro lado, a implementação da heurística gulosa resolve o problema em tempo polinomial com uma taxa de acerto de 92%.

Artigo 7:

- **Contexto:** O CADI (Curso Aberto Disponível na Internet) é um método emergente de ensino remoto. A implantação bem sucedida dele exige mudanças conceituais na forma como professores e alunos se comportam em um ambiente aberto de ensino.
- **Objetivo:** Apresentar um CADI para um curso introdutório de Engenharia de Software proposto a partir de um curso presencial.
- **Revisão da literatura:** Há diversas citações neste arquivo, dentre elas podemos ressaltar “Derwin, E. (2009)”, “A. Fox and D. Patterson. (2012)” e “K. Masters. (2011)”.
- **Metodologia:** Comparar o desempenho do aluno do CADI a um aluno do ensino presencial por meio de questionários.
- **Resultados:** Os resultados dos alunos no curso presencial com apoio do CADI são estatisticamente maiores do que os resultados de alunos cursando a mesma disciplina puramente presencial.

Artigo 8:

- **Contexto:** A notoriedade da escolha de linguagens estaticamente ou dinamicamente tipadas para determinados exercícios.

- **Objetivo:** Realizar uma análise de quais fatores mais influenciam a escolha de um programador por um sistema de tipos.
- **Revisão da literatura:** Há diversas citações neste arquivo, dentre elas podemos ressaltar “Bruce, K. (2002)”, “Bruch, M., Monperrus, M., and Mezini, M. (2009)” e “Cardelli, L. (1996)”.
- **Metodologia:** Analisar uma bateria de códigos fonte de projetos e identificar padrões.
- **Resultados:** A bagagem de experiência prévia do programador, extensão do projeto, complexidade dos componentes, escopo e visibilidade são alguns dos fatores mais importantes para essa decisão.