

UNIVERSIDADE DE BRASÍLIA–UNB  
INSTITUTO DE CIÊNCIAS EXATAS  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

**Felipe Evangelista dos Santos**  
**Fundação Universidade de Brasília**

**Implementação do SDK para  
monitoramento dos serviços da  
Universidade de Brasília**  
**Linha de Pesquisa: Engenharia de  
Software**

Brasília  
maio, 2017

**Felipe Evangelista dos Santos**  
**Fundação Universidade de Brasília**

**Implementação do SDK para  
monitoramento dos serviços da  
Universidade de Brasília**  
**Linha de Pesquisa: Engenharia de  
Software**

Pré-Projeto de Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação Aplicada do Departamento de Ciência da Computação da Universidade de Brasília.

Área de Concentração: Engenharia de Software.

Brasília  
maio, 2017

---

# Sumário

<b>1</b>	<b>Implementação do SDK para monitoramento dos serviços da Universidade de Brasília</b>	<b>3</b>
1.1	Introdução . . . . .	3
1.2	Justificativa . . . . .	4
1.3	Objetivo Geral . . . . .	4
1.3.1	Objetivos Específicos . . . . .	4
1.4	Revisão da Literatura . . . . .	5
1.5	Metodologia . . . . .	5
1.6	Plano de Trabalho e Cronograma . . . . .	5
	<b>Referências</b>	<b>6</b>

---

# Implementação do SDK para monitoramento dos serviços da Universidade de Brasília

## 1.1 Introdução

A manutenção de *software* é um processo complexo e requer uma análise detalhada do problema quando se necessita realizar modificações, sejam elas corretivas ou evolutivas [1] [2] [3]. Assim, para a modificação em um *software*, além do entendimento das regras de negócio e da documentação, é necessário ter o conhecimento do seu funcionamento interno. Atualmente, o Centro de Informática (CPD) da Universidade de Brasília (UnB), não possui documentação da arquitetura da maioria dos seus sistemas legados, ponto que promove a dificuldade na manutenção desses sistemas.

O entendimento arquitetural de um *software* sem documentação é uma tarefa árdua e dependendo da estratégia utilizada pode gerar um esforço que será desperdiçado por não identificar os ajustes realmente necessários para a recuperação da arquitetura do *software* que está sendo analisado. Dessa forma, é necessário o uso de métodos e ferramentas para apoiar a análise estática para a refatoração dos sistemas legados.

As técnicas de análise estática podem ser usadas para verificar os modelos de especificação - e de projeto de um sistema - para encontrar erros antes que uma versão executável do sistema esteja disponível e ainda tem a vantagem de que a presença desses erros não interrompa a verificação do sistema [1]. Uma técnica de análise estática é a visualização de *software*, que é usada para a criação de imagens, diagramas ou animações para relacionar informações e situações que seriam difíceis de descrever e compreender em outros formatos [4].

Segundo [5], a área de visualização busca representar dados e informações graficamente, por meio de técnicas e abstrações, de forma que a capacidade cognitiva do ser humano (derivada da sua memória, percepção e raciocínio) seja estimulada para facilitar

a compreensão de um determinado assunto, e que existem duas grandes disciplinas dentro da visualização, a visualização científica, que processa dados físicos, e a visualização da informação, que processa dados abstratos, baseando-se em alguns princípios: a exploração interativa que é a visualização interativa que permite a o usuário, ter uma visão geral e, depois, aplique filtros e *zoom* para obter detalhes sobre demanda, e o foco + contexto que trata-se de uma visualização detalhada de uma parte da informação - o foco - é incorporada dentro de uma visualização do contexto, isto é, uma informação mais refinada sobre as partes relacionadas ao foco. Assim, técnicas de foco + contexto proveem detalhamento e uma visão geral ao mesmo tempo.

Nos sistemas mantidos pela unidade de Serviços de Sistemas de Informação (SSI) do CPD da UnB, existem situações bastante diferentes em relação aos padrões de desenvolvimento de *software* atuais. No período que antecedeu o *bug* do milênio, vários sistemas eram hospedados em *mainframes*. Dessa forma, para evitar que esses *softwares* interpretassem a data do ano 2000 como 1900, esses sistemas precisaram ser migrados para novas tecnologias. Nessa época, conforme sugestão de uma empresa de consultoria contratada especificamente para essa migração, as funções e sub-rotinas foram implementadas em uma camada de *scripts Transact-SQL*. Com isso, as regras de negócio, inserções, deleções e alterações foram unificadas em uma mesma camada, aumentando o acoplamento entre diversos sistemas e dificultando o entendimento.

Diante desse cenário, é necessária uma abordagem de apoio à análise estática para a refatoração dos sistemas legados, que envolve o estudo, com foco no entendimento dos *scripts Transact-SQL* e dos modelos de dados, para a melhoria de desempenho e da manutenção. Essa análise exige o mapeamento dos processos desses sistemas e o entendimento das ferramentas de apoio que serão utilizadas, e deve ser feita baseando-se em pesquisas, teses, artigos publicados e projetos realizados sobre o assunto.

## 1.2 Justificativa

## 1.3 Objetivo Geral

### 1.3.1 Objetivos Específicos

- ❑
- ❑
- ❑
- ❑
- ❑

## 1.4 Revisão da Literatura

## 1.5 Metodologia

## 1.6 Plano de Trabalho e Cronograma

Tabela 1 – Cronograma de Atividades do Mestrado

Tarefa	2017/2	2018/1	2018/2	2019/1
Disciplinas do Núcleo Básico	X			
Disciplinas de Engenharia de <i>Software</i>		X		
Tarefa 1	X	X	X	
Tarefa 2		X	X	
Tarefa 3			X	
Tarefa 4		X	X	X
Tarefa 5			X	X
Tarefa 6				X
Tarefa 7		X	X	X
Tarefa 8				X

---

## Referências

- [1] SOMMERVILLE, I. **Software engineering**. Décima edição. New York City, USA: Pearson Publisher, 2016.
- [2] PRESSMAN, R. S. **Engenharia de Software - Uma abordagem profissional**. Sétima edição. São Paulo: Editora Makron Books, 2011.
- [3] CAGNIN, M. I. **PARFAIT: uma contribuição para a reengenharia de software baseada em linguagens de padrões e frameworks**. Tese (Doutorado) — Universidade de São Paulo (USP). Instituto de Ciências Matemáticas e de Computação de São Carlos, 2005.
- [4] SPENCE, R. **Information Visualization**. Third edition. Springer International Publishing, 2014. ISBN 978-3-319-07341-5. Disponível em: <<http://link.springer.com/book/10.1007/978-3-319-07341-5>>.
- [5] DIEHL, S. **Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software**. Secaucus, NJ, USA: Springer-Verlag New York Publisher, 2007. ISBN 3540465049.
- [6] VASCONCELOS, A. P. V. de. **Uma Abordagem de apoio à Criação de Arquiteturas de Referência de Domínio baseada na Análise de Sistemas Legados**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro., 2007. Disponível em: <[http://reuse.cos.ufrj.br/files/publicacoes/doutorado/Dou\\_Aline.pdf](http://reuse.cos.ufrj.br/files/publicacoes/doutorado/Dou_Aline.pdf)>.
- [7] FERREIRA, R. de A. M. **Modularização de tratamento de exceções usando programação orientada a aspectos**. Dissertação (Mestrado) — Universidade Estadual de Campinas, 2006. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?code=vtls000414124>>.
- [8] SANTOS, G.; VALENTE, M. T.; ANQUETIL, N. Remodularization analysis using semantic clustering. **IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE)**, p. 224–233, 2014.
- [9] FARRER, H. et al. **Programação estruturada de Computadores: Algoritmos estruturados**. Terceira edição. Rio de Janeiro: Editora LTC, 2008.