

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

Análise Estática para a Refatoração dos
Sistemas Legados da Universidade de
Brasília
Linha de Pesquisa: Engenharia de
Software

Felipe Evangelista dos Santos
Fundação Universidade de Brasília

Análise Estática para a Refatoração dos
Sistemas Legados 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.

Orientador: Prof. Dr. Eduardo Henrique F.M. Teixeira

Sumário

1	Análise Estática para a Refatoração dos Sistemas Legados da Universidade de Brasília	3
1.1	Introdução	3
1.2	Justificativa	4
1.3	Objetivo Geral	5
1.3.1	Objetivos Específicos	5
1.4	Revisão da Literatura	5
1.5	Metodologia	6
1.6	Plano de Trabalho e Cronograma	6
	Referências	8

Análise Estática para a Refatoração dos Sistemas Legados 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

Os módulos implementados nos sistemas legados da UnB, apesar de facilitarem a portabilidade de *front-end* em caso de descontinuidade de uma tecnologia utilizada, ainda possuem alto acoplamento entre as regras de negócio e a persistência de dados, que são implementadas em uma única camada por meio de *scripts Transact-SQL*. Além disso, a situação atual desses sistemas é desfavorável em relação aos custos referentes à capacitação dos colaboradores para a tecnologia que foi utilizada no desenvolvimento desses sistemas. Ou seja, a maioria desses sistemas foi desenvolvida com linguagens que atualmente ou são obsoletas ou possuem arquiteturas que, apesar de possuírem baixo acoplamento com o núcleo da camada de regras de negócio, são de difícil entendimento e manutenção. Por conta dessas características, diversos sistemas legados da UnB possuem problemas de desempenho devido ao aumento da demanda de requisições de acesso e ao não acompanhamento

tecnológico dessas arquiteturas.

Dessa forma, com essa pesquisa, espera-se prover meios para facilitar o entendimento das funcionalidades para a refatoração visando otimizar o desempenho de funcionamento dos sistemas legados da UnB. Além disso pretende-se também que a análise estática para a refatoração dos sistemas possa trazer grandes benefícios como o entendimento das informações a partir do código fonte, criar e especificar processos e métodos de migração e *refactoring*, realizar estudos e utilizar métricas para estatísticas dos sistemas legados.

1.3 Objetivo Geral

O objetivo geral da pesquisa proposta é realizar a análise estática para a refatoração dos sistemas legados da UnB. Para isso serão realizadas pesquisas e utilizados projetos de arquitetura de *software*, técnicas e ferramentas de apoio, visando à melhoria de desempenho e manutenibilidade.

1.3.1 Objetivos Específicos

Os objetivos específicos são:

- ❑ Levantamento do estado da arte por meio da análise crítica dos artigos científicos recentes referentes ao tema análise estática para a refatoração dos sistemas legados.
- ❑ Definir um processo para a realização da refatoração dos sistemas legados da UnB baseando-se nas abordagens mais relevantes estudadas na literatura;
- ❑ Utilizar ferramentas de apoio para a análise estática para a refatoração dos sistemas legados da UnB;
- ❑ Realizar um estudo de caso aplicando o método desenvolvido para promover adequações e melhorias em um ou mais sistemas legados da UnB;
- ❑ Publicação de artigos científicos contendo descobertas relevantes e os resultados do estudo de caso realizado.

1.4 Revisão da Literatura

Em [6] é apresentada a importância dos *softwares* legados e a dificuldade para mantê-los, e que devem ser definidas técnicas e metodologias para a migração desses sistemas para uma nova tecnologia. Em [1] é descrita a modularização de programas agrupando as partes relacionadas e removendo as redundâncias de funcionalidades. Em alguns casos, esta fase pode envolver reconstrução arquitetural por meio de um processo manual. Entretanto, em [2] é explicado que as arquiteturas de *software* incorporam a modularidade dividindo as

sub-rotinas e funções em componentes separados, nomeados, endereçáveis e integrados para atender aos requisitos do problema.

Uma abordagem interessante para modularizar o comportamento excepcional de um sistema é descrita em [7]. A importância desse trabalho consiste em estudos de caso para a refatoração de dois sistemas orientados a objetos e de um sistema orientado a aspectos.

Em [8] são descritas abordagens que se baseiam em técnicas para recuperar informações necessárias para identificar classes distintas que efetuam as mesmas funcionalidades. Dessa forma, essas classes são selecionadas para serem remodularizadas. Uma outra abordagem também aplica a modularização, que é uma técnica altamente recomendável tanto pela eficiência no projeto e desenvolvimento dos mesmos quanto pela quantidade, confiabilidade e manutenção do produto elaborado por meio de ferramentas [9]. Nesse estudo partes comuns a vários programas ou que se repetem dentro de um mesmo programa são modularizadas em uma sub-rotina ou função, e conseqüentemente, são programadas e testadas uma só vez. Entretanto, deve haver uma preocupação com o desempenho, devido a construção de módulos genéricos e comuns.

1.5 Metodologia

O estudo da análise estática para a refatoração dos sistemas legados será realizado no CPD da UnB, e a execução do projeto ocorrerá na área de desenvolvimento de sistemas do setor SSI.

Serão coletados dados sobre os sistemas legados, a fim de buscar informações relativas às funcionalidades, regras de negócio e principais tecnologias que foram utilizadas. Assim haverá uma análise prévia das funcionalidades implementadas dos sistemas legados e como as regras de negócio podem ser codificadas em módulos e sub-rotinas de forma mais eficiente para atender às situações específicas de cada sistema.

Faz parte ainda do escopo metodológico do projeto a realização de pesquisas para o entendimento dos módulos, funções e sub-rotinas dos sistemas legados em que o escopo e a abrangência de funcionamento podem apresentar ambientes e configurações específicos. Para a melhoria desse entendimento serão utilizados processos e ferramentas de engenharia reversa e de reengenharia para análise e levantamento de problemas com modularidade e arquitetura. Um ambiente computacional será disponibilizado para a realização de testes dos sistemas, verificação de consistência da refatoração e a aceitação dos envolvidos no projeto.

1.6 Plano de Trabalho e Cronograma

O candidato à vaga no Mestrado em Computação Aplicada cursou como aluno especial a disciplina Mineração de Dados e Textos (MDT) e obteve aprovação. Além disso,

encontra-se cursando no período letivo 2016/1 as disciplinas Arquitetura e Desenho de *Software* (ADS), Engenharia de Requisitos (ER) e Tópicos Avançados em Engenharia de *Software* (TAES) . Considerando-se o aproveitamento desses 8 (oito) créditos, espera-se cursar 16 (dezesesseis) créditos em disciplinas obrigatórias e defender dissertação para a obtenção do título de Mestre em Computação Aplicada. Para esse fim, serão cursadas 8 (oito) disciplinas de 2 (dois) créditos, sendo 5 (cinco) do Núcleo Básico e 3 (três) da linha de Engenharia de Software, distribuídas conforme mostrado na Tabela 1.

Como o candidato a discente é servidor público da UnB, lotado no CPD, será dedicado o regime de tempo parcial. Dessa forma, uma vez que a referida instituição pública disponibiliza liberação de horário para os seus servidores de quadro efetivo cursarem disciplinas de cursos de pós-graduação, serão necessárias 12 (doze) horas semanais para a realização das disciplinas presenciais, mais 20 (vinte) horas de estudo, totalizando 32 (trinta e duas) horas semanais de dedicação. Após cursadas as disciplinas necessárias para completar os 24 (vinte e quatro) créditos exigidos para o título de Mestre, este trabalho será desenvolvido por meio de oito tarefas listadas a seguir, e por meio do cronograma apresentado na Tabela 1.

1. Levantamento do estado da arte sobre análise estática para a refatoração dos sistemas legados;
2. Escolha das abordagens que serão utilizadas por meio de características não exploradas nas soluções atuais de análise estática para a refatoração dos sistemas legados;
3. Exame de Qualificação de Mestrado;
4. Desenvolvimento de novas abordagens que serão utilizadas na análise estática para a refatoração dos sistemas legados da UnB;
5. Prova de conceito das abordagens desenvolvidas;
6. Submissão de artigos científicos em congressos e periódicos;
7. Escrita da dissertação de mestrado;
8. Defesa do mestrado.

Tabela 1 – Cronograma de Atividades do Mestrado

Tarefa	2016/2	2017/1	2017/2	2018/1
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>.
- [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.