

CENTRO DE ENSINO SUPERIOR DE JUIZ DE FORA

NOME DO ALUNO

TÍTULO DO PROJETO

Projeto de Pesquisa

Juiz de Fora

2008

NOME DO ALUNO

TÍTULO DO PROJETO

Projeto de pesquisa apresentado como pré-requisito para a conclusão do curso de XXXXXXXXXXXXXXXX do Centro de Ensino Superior de Juiz de Fora, tendo como professor orientador YYYYYYYYYYYYYY.

Juiz de Fora

1º. sem. 2008

SUMÁRIO

INTRODUÇÃO.....	4
1- LEVANTAMENTO DE LITERATURA	7
2- PROBLEMA.....	8
3- HIPÓTESE	9
4- OBJETIVOS	10
5- JUSTIFICATIVA.....	11
6-METODOLOGIA	12
7-ETAPAS DE DESENVOLVIMENTO DA MONOGRAFIA.....	13
8-CRONOGRAMA.....	14
9-REFERÊNCIAS BIBLIOGRÁFICAS.....	15

INTRODUÇÃO

O termo refatoração já é amplamente difundido entre os grupos de desenvolvimento de *software*, principalmente quando entramos no setor dos desenvolvedores de *frameworks*, em que estão as origens da refatoração onde havia a competição pela intensa redução das linhas de código existentes. Vale destacar que a redução das linhas de código não é, de forma alguma, refatoração na concepção atual, pois não existia uma política de teste eficiente, catálogo de refatoração e muito menos um processo bem mapeado para aplicá-la. Não podemos, de forma alguma, medir a qualidade do desenvolvimento de uma aplicação pelo número de linhas da mesma e sim pela qualidade da modelagem, estruturação, organização, documentação dos códigos e testes e, principalmente, se os requisitos funcionais ou não funcionais estão sendo prontamente atendidos pela solicitação do cliente.

Sabemos que, mesmo tendo bons profissionais projetando e desenvolvendo uma aplicação, sempre são implementados códigos que não se enquadram nas boas práticas do desenvolvimento e sabemos que os desenvolvedores já possuem essa cultura entre eles, onde sempre fazem refatoração do código já desenvolvido paralelamente com a implementação de novas funcionalidades sem alterar as que já estão implementadas, quando corrigem alguns erros ou falhas, quando revisão código, realizando, assim, pequenas modificações, à medida que estas vão sendo implementadas no decorrer do desenvolvimento do projeto.

Estas modificações irão ter um impacto direto na qualidade do código, e propicia que os desenvolvedores, apliquem metodologias ágeis que, levem a uma melhora contínua a cada ciclo da implementação. Porém, para evidenciarmos esta postura, teremos que adotar certos cuidados e certos padrões para realmente melhorar o código do projeto existente. Sem isso, em vez de melhorá-lo, erros serão inseridos e faremos a refatoração de uma maneira descontrolada, sem uma gama de testes bem elaborados, sem sabermos o que é, realmente, refatorar ou como refatorar e sem o controle do que foi alterado desse determinado trecho de código, atacando, desse modo, os vícios sempre existentes no processo de desenvolvimento de *software*, utilizando livros de autores de extrema qualidade como Martin Fowler, Craig Lerman, Kent Benk.

Até este momento podemos nos perguntar onde os bancos de dados relacionais e os bancos de dados orientados a objetos se enquadram neste tema “refatoração” é onde tudo começa, pois as equipes de desenvolvimento utilizam, constantemente, a refatoração juntamente

com metodologia de desenvolvimento ágil em seu código sem nunca se preocuparem com banco de dados que a sua aplicação utiliza.

Sabemos que o relacionamento entre desenvolvedores e *DBAs* é extremamente conturbado, já que nenhum dos lados assumem responsabilidades por erros e lentidões nas aplicações e nos bancos. É mais fácil atribuir a responsabilidade de determinados erros a uma gama de desenvolvedores ou administradores de dados do que juntos analisarem o problema em causa e juntos corrigirem-no, seja na aplicação ou no banco de dados.

A cultura de refatoração ainda é vista com desconfiança pela grande maioria dos *DBAs* já que é uma cultura criada, executada e aprimorada pelos "rivais" desenvolvedores que vivem no mesmo mundo, porém com posturas completamente distintas, pois os desenvolvedores "OO" já a adotam de maneira mais natural. Os *DBAs* são profissionais extremamente conservadores, bem arredios às mudanças, já que os bancos relacionais são confiáveis, robustos porém, estão sendo impulsionados pelos desenvolvedores e metodologias de desenvolvimento de *software* ágeis como *Scrum*, *Extreme Program (XP)* que visam o processo iterativo e incremental a aderirem ao tema e através de fóruns e sites especializados e literaturas, com foco em banco de dados. Os *DBAs* vêm discutindo alternativas para melhoria nos catálogos de refatoração dos bancos de dados e as estão introduzindo de maneira correta entre eles, que as utilizam para refatorar, e estão percebendo que realmente estão melhorando, a estrutura, a qualidade dos dados, a integridade referencial, a arquitetura e os métodos em bancos de dados.

Como as mudanças são constantes em todos campos do desenvolvimento, suas etapas tornam-se extremamente voláteis a todo momento e isto possibilitará refatorar tanto o código da aplicação quanto o banco de dados, fazendo com que os *DBAs* realmente adotem a cultura dos desenvolvedores "OO", seguindo a mesma linha de pensamento, criando uma perspectiva coesa entre os mesmos.

Para a refatoração de código, existem catálogos abordando inúmeros aspectos que indicam o que refatorar, como refatorar e suas práticas.

Os desenvolvedores são apoiados diretamente por ferramentas que propiciam controle de alterações e metodologias bem definidas para a realização de testes e *IDEs*, com integração à ferramentas de controle de versões, otimização e outras funcionalidades que detectam pontos que não se enquadram nas boas práticas de desenvolvimento. Contudo, não vemos a integração e o suporte tão eficazes entre as *IDEs* que existem e os banco de dados, estas não são, de maneira alguma, tão eficientes e abrangentes quanto as de desenvolvimento. Existem bem menos pontos mapeados para fazermos boas práticas na refatoração dos bancos de dados.

Assim, o objetivo deste projeto de pesquisa, é mencionar os principais pontos que estão sendo abordados por estes profissionais, suas ferramentas de testes, controle de versões; o processo para executar de forma correta e eficaz a refatoração, o catálogo de refatoração, mencionar os problemas. Serão estes os tópicos principais, nos quais devemos nos ater, assim como sugestões para novos projetos de pesquisa, novos itens que não estão presentes nos catálogos de refatoração, dentre outros.

1. LEVANTAMENTO DE LITERATURA

O autor FOWLER (2004) será minha grande fonte de inspiração para iniciar esta pesquisa, pois ele foi o primeiro autor a falar sobre o assunto, e como se iniciou o processo da refatoração no mundo do desenvolvimento de *software*. Ele a descreve como sendo um processo que atenta a práticas técnicas de uma maneira bem ordenada, disciplinada e documentada, atendendo a regras básicas e ao catálogo de refatoração onde ele sugere que os desenvolvedores façam pequenas alterações no código existente para melhorar sua estrutura, organização, encontrar erros com mais facilidade, e a desenvolver com mais agilidade. Sugere também que os desenvolvedores façam essas pequenas alterações no código em paralelo com o desenvolvimento sempre que forem consertar um determinado erro, quando revisar o código ou acrescentar funcionalidade, com isso permite que os desenvolvedores possam fazer a refatoração de uma maneira incremental e evolutiva. Também em seu livro ele faz uma sucinta menção sobre os problemas da refatoração nos bancos de dados orientados a objetos e nos bancos relacionais.

Outra fonte de pesquisa utilizada neste projeto, será a do autor SCOTT W. AMBER, que traz uma literatura mais específica sobre o tema mencionado, onde realmente é feita uma abordagem muito mais profunda do tema, trazendo todos os tópicos necessários para, de fato, se fazer a refatoração de banco de dados de maneira correta, mencionando todo o processo de refatoração e o catálogo com os tipos de refatorações existentes.

Será ainda utilizada outra literatura - a dos autores ELMASRI e NAVATHE. Eles abordam tópicos sobre banco de dados “OO” e Relacionais.

Além desses livros, será a Internet uma grande fonte de consulta para aquisição de conhecimento e para conclusão de todo o projeto de pesquisa. Outras linhas de pesquisa referentes ao tema poderão ser abordadas à medida que este projeto for avançando.

2. PROBLEMA

Como realizar pequenas alterações nos bancos de dados melhorando sua estrutura e performance sem prejudicar os acessos das inúmeras aplicações que o utilizam, preservando a semântica dos dados, como colocar estas alterações em produção em curto espaço de tempo, com total confiança de seu funcionamento e com todo o processo bem mapeado e documentado.

3. HIPÓTESE

Utilizando conceitos específicos de refatoração de banco de dados, podemos fazer vários tipos de pequenas alterações nos bancos, preservando toda integridade e semântica de seus dados e as colocando em produção rapidamente, com toda a segurança e confiabilidade, sem prejudicar de forma alguma, o acesso das aplicações que utiliza, preservando a semântica dos dados e documentando todo o processo.

4. OBJETIVOS

- Aprofundar meus conhecimentos sobre as técnicas de refatoração de banco de dados relacionais e de bancos de dados orientados a objetos;
- Aprofundar meus conhecimentos em arquitetura de *software*;
- Aprofundar meus conhecimentos na arquitetura de bancos dados relacionais;
- Aprofundar meus conhecimentos em banco de dados orientados a objetos;
- Conhecer e indicar as pesquisas que estão sendo realizadas na área de refatoração de banco de dados;
- Conhecer ferramentas que auxiliam os profissionais, para fazerem o processo de refatoração de uma maneira mais controlada e correta;
- Observar a aceitação deste tema por profissionais que utilizam principalmente os bancos de dados relacionais.

5. JUSTIFICATIVA

Sabemos que inúmeros *DBAs* e desenvolvedores generalistas realizam pequenas modificações no banco de dados afim de melhorar desempenho, estrutura, organização de dados, alterações e em *views, triggers, procedures* dentre outras. Mudanças das formas que são realizadas, prejudicam o desempenho e a estrutura do banco de dados e o acesso das aplicações pois, não utilizam nenhum tipo de padrão, metodologia ou processos para realizar estas pequenas alterações com total segurança e rastreabilidade do que foi realizado.

6. METODOLOGIA

Será utilizado o que há de melhor em literatura para elaboração deste projeto de pesquisa. Primeiramente, será feita uma pesquisa sobre o que esta disponível em literatura sobre o tema, inclusive em sites e fóruns que tratem especificamente deste assunto.

Em seguida, será pesquisado quais as categorias de refatoração que estão sendo utilizadas em bancos de dados.

Também será pesquisado os testes necessários para garantir uma boa refatoração, além de como se realiza todas as etapas do processo de refatoração.

7. ETAPAS DE DESENVOLVIMENTO DA MONOGRAFIA

Segue um esquema da divisão do trabalho:

1.O que é refatoração?

1.1 Refatoração de banco de dados

2.Principais diferenças entre refatoração e banco de dados e de *software*

3. Categorias de Refatoração de banco de dados

3.1 Estruturais

3.2 Qualidade de dados

3.3 Integridade referencial

3.4 Arquiteturais

3.5 Método

4. Testes

5. *SandBoxes*

6.Processo de refatoração de banco de dados

7.O que não é considerado refatoração em banco de dados

8.Quando Refatorar

9. *LiquiBase* uma ferramenta de auxílio a Refatoração de banco de dados

10.Conclusão

8. CRONOGRAMA

<i>Mês/ Atividades</i>	<i>Janeiro 2008</i>	<i>Fevereiro 2008</i>	<i>Março 2008</i>	<i>Abril 2008</i>	<i>Maió 2008</i>	<i>Junho 2008</i>	<i>Julho 2008</i>	<i>Agosto 2008</i>	<i>Setembro 2008</i>	<i>Outubro 2008</i>	<i>Novembro 2008</i>	<i>Dezembro 2008</i>
Escolha do tema	X											
Pesquisa bibliográfica	X											
Pesquisa Sobre as fontes achadas	X	X										
Criação do pré-projeto		X	X	X								
Ajustes e Aprovação do pré-projeto				X	X							
Redação da monografia						X	X	X	X	X	X	
Revisões e acertos						X	X	X	X	X	X	
Entrega da monografia												X

9. REFERÊNCIAS BIBLIOGRÁFICAS:

ELMASRI, Ramez; NAVATHE, Shamkant B. Fundamentals Of Database System, 4ª.edição. Local: São Paulo: Addison Wesley, 2005.

FOWLER, MARTIN; Refatoração Aperfeiçoando o Projeto de Código Existente / Martin Fowler Com contribuições de Kent Beck, Jonh Brant, Wilian Opdyke e Don Roberts; tradução Acauan Fernandes. Local: Porto Alegre: Bookman, 2004.

Refactoring Databases - Evolutionary Database Design. Disponível em : <http://databaserefactoring.com/index.html>. Acesso em : 07 dez. 2007

Scott W. Ambler ,Pramodkumar J. Sadalage; Refactoring Databases: Evolutionary Database Design, 3ª edição, edição revisada MAKRON Books, 2006.

The Process of Database Refactoring. Disponível em: <http://www.agiledata.org/essays/databaseRefactoring.html>>. Acesso em : 07 dez. 2007.

Sato, Danilo; Ferreira, João Eduardo. Banco de Dados Ágeis e Refatoração. Disponível em : <http://www.agilcoop.org.br>>. Acesso em : 07 dez. 2007.

LiquiBase Database Refactoring Tool 1.0. Disponível em: http://www.theserverside.com/news/thread.tss?thread_id=45919>. Acesso em : 08 maio 2008.

Database Refactoring With LiquiBase. Disponível em: <http://www.liquibase.org/>> Acesso em : 08 maio. 2008.

Universidade Federal de Santa Catarina Departamento de Informática e Estatística Ciências da Computação INE5322 - Engenharia de software. REFATORAÇÃO equipe George Tavares Herbert Mattei de Borba Professor Ricardo Pereira e Silva, D. Sc. Disponível em: http://www.inf.ufsc.br/~herb/disc/engenharia_de_software/>. Acesso em: 18 de maio de 2008.

Database Refatoração. Disponível em: <http://www.agiledata.org/essays/databaseRefactoring.html>>. Acesso em: 18 de maio de 2008.

Refatora. Disponível em: <http://pt.wikipedia.org/wiki/Refatora%C3%A7%C3%A3o>>. Acesso em: 18 de maio de 2008.