

Identificação dos Pontos de Estabilidade e Flexibilidade no Modelo de Requisitos dos Métodos de Preço de Venda

Vinicius Camargo Andrade¹, Paulo E. Capeller¹, Simone Nasser Matos¹

¹Coordenação de Informática – Universidade Tecnológica Federal do Paraná (UTFPR) 84.016-210 – Ponta Grossa – PR – Brasil

Abstract. *Actually there is not in the market free software that implements several pricing methods. Due about this problem, the Software Engineering Research Group from Federal Technological University of Paraná, Campus Ponta Grossa, has been developing a framework that implements several pricing methods. This article compared some methodologies for framework developing and used responsibilities-driven approach to create the requirements model of the framework in the sales price domain. This model contains the identification of the frozen e hot spots of pricing methods, core of the frameworks development.*

Keywords: *Framework Development Methodologies, Subframework, Domain framework, Sales Price.*

Resumo. *Atualmente não há no mercado um software que implemente diversos métodos de formação de preço de venda e que seja gratuito. Conscientes deste problema, o Grupo de Pesquisa em Engenharia de Software da Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa, está desenvolvendo um framework que implemente diversos métodos de formação de preço de venda. Este artigo comparou algumas metodologias de desenvolvimento de framework e usou a abordagem dirigida a responsabilidades para criar o modelo de requisitos do framework no domínio de preço de venda. O modelo contém a identificação dos pontos de estabilidade e flexibilidade dos métodos de preço de venda, núcleo para o desenvolvimento de frameworks.*

Palavras-chave: *Metodologias de Desenvolvimento de Frameworks, Subframework, Framework de Domínio, Preço de Venda.*

1. Introdução

Para se manterem no mercado, as empresas atualmente enfrentam diversos problemas, o principal deles e a formação de preço de venda.

Durante anos, diversos autores desenvolveram métodos para calcular o preço de venda de um produto ou serviço, em que cada método é composto por suas próprias características.

A princípio estes cálculos eram feitos manualmente, porém entender qual método era o ideal para naquele instante, demandava tempo. Atualmente existem

softwares que implementam métodos de formação de preço de venda auxiliando o gestor, porém estes softwares são pagos e implementam poucos métodos.

O presente artigo apresenta um estudo sobre os métodos de formação de preço de venda, juntamente com uma comparação sobre as metodologias para o desenvolvimento de *framework*. Após a análise desta comparação foi aplicada a metodologia de Matos e Fernandes (2008) para o desenvolvimento do *subframework* Controlar Atributos. Este *subframework* contém as características comuns e específicas de cada método de formação de preço de venda, possibilitando a criação de um aplicativo capaz de conter vários métodos.

A seção 2, aborda uma breve introdução sobre preço de venda. A seção 3 descreve um estudo analítico sobre como as abordagens da literatura tratam o projeto arquitetural de *framework* e *subframework*. A seção 4 descreve a aplicação da abordagem dirigida por responsabilidades e sua aplicação no modelo de requisitos do *subframework* Controlar Atributos. E, por fim, a seção 5 conclui o trabalho, discutindo os resultados obtidos.

2. Preço de Venda

Segundo Bornia (2002), preço de venda é o valor atribuído a um produto ou serviço, cujo valor deve ser suficiente para cobrir todas as despesas (fixas e variáveis) e o custo que do que está sendo vendido, gerando ainda assim, uma margem de lucro para investimentos posteriores.

Uma empresa possui despesas fixas e variáveis [Bornia 2002]. Despesas fixas, ou também chamadas de custos fixos, são aqueles custos que independentemente da quantidade produzida ou vendida serão sempre as mesmas, como por exemplo, salários administrativos, aluguel, IPTU.

Despesas variáveis ou custos variáveis são obtidos em função das vendas, ou seja, quanto maior o nível de produção e de trabalho em uma empresa, maior será a despesa. Normalmente, as despesas variáveis se caracterizam por um percentual sobre o valor das vendas efetivadas, tal como: comissões.

Para uma empresa obter sucesso, vários aspectos importantes devem ser levados em consideração, entre eles está o estabelecimento do preço de venda, uma vez que o mercado está cada vez mais competitivo devido às tarifações e impostos, da necessidade de novos investimentos e do surgimento de novas despesas [Silva 2001].

Com o crescimento da concorrência e do mercado consumidor, as empresas se obrigaram a procurar novas formas de analisar e tomar decisões que atenda as suas necessidades, além dessas decisões serem satisfatórias, elas precisam ser tomadas com rapidez, pois o mercado é variável e o preço de venda precisa ser revisto a todo o momento.

O preço estabelecido deve cumprir a seguinte meta: cobrir todas as despesas fixas e variáveis, somadas com os próprios custos e acrescentada a necessidade de se obter lucro com a venda, além de ser um preço competitivo.

Por isso, a importância de conhecer os métodos de preço de venda e também definir o sistema de rateio de custos que mais de adaptem a sua realidade [Silva 2001].

Dentre os métodos de preço de venda, foram estudados os descritos a seguir.

2.1. Método de Custeio Baseado em Atividades (Activity-Based Costing)

Segundo Nakagawa (1995), o Custeio Baseado em Atividades trata-se de um método desenvolvido para facilitar a análise estratégica de custos relacionados com as atividades que mais impactam o consumo de recursos de uma empresa.

Os recursos são classificados como diretos e indiretos. Os recursos diretos são os custos relacionados diretamente com a produção de um produto. Por exemplo, custo de mão-de-obra direta, horas de mão-de-obra direta, matéria-prima e a quantidade de horas de utilização de máquinas na produção.

Já os recursos indiretos, também chamado de custos indiretos de fabricação (CIF) ou *overhead*, não estão relacionados diretamente aos custos dos produtos, por esse motivo, o método ABC leva grande vantagem em relação aos sistemas tradicionais de custeio [Cogan 1995].

Diferentemente dos demais métodos, o custeio baseado em atividades leva em consideração as atividades envolvidas na produção, como por exemplo: preparação de máquinas, inspeções de qualidade, recebimento e movimentação de materiais, entre outros.

O custeio baseado em atividades pressupõe que estas consomem recursos, gerando custos, e que os produtos utilizam atividades, absorvendo seus custos. Assim, os procedimentos do ABC consistem em dividir a organização em atividades relacionadas à produção, determinada pelo gestor de finanças. Após a compreensão e comportamento destas atividades, identificam-se as causas dos custos, em seguida, aplica-se o procedimento da alocação dos custos aos produtos de acordo com as intensidades de uso [Bornia 2002].

2.2. Método SEBRAE-PR

Primeiramente é necessário retirar o valor do ICMS do custo de Aquisição do produto. Com isso, é feito um levantamento do percentual das despesas fixas e variáveis da empresa [Crazuski, Feitosa and Cordeiro 2008].

É necessário o estabelecimento da porcentagem de lucro desejável, isso se dá pela obtenção do faturamento previsto pela empresa e seu investimento total.

A precisão do valor final de venda do produto depende da precisão do levantamento efetuado para o cálculo. Depois disso, chega-se a um valor que ainda deve ser analisado, pois se deve ao fim do processo verificar se este preço calculado esta compatível com os processos que o mercado pratica [Sebrae 2009].

2.3. Método Baseado no Custo Pleno

Segundo Nascimento e Vartaniam (1999:34), “ método de custeio pleno é aquele em que todos os custos e despesas de uma entidade são levados aos objetos de custeio, normalmente unidades de produtos e/ou ordens de serviços”

O método de custo pleno leva em consideração todos os gastos de uma organização, sem exceção. Porém, não faz distinção entre custos fixos e variáveis, o que pode levar a uma empresa a vender um produto e não cobrir as despesas do mesmo.

O seu funcionamento consiste na soma de todos os custos e despesas do produto ou serviço, envolvendo matérias-primas, mão-de-obra direta, custos indiretos de

produção, custos de transformação, custos de produção e despesas de vendas e administração.

Após realizada esta soma, é acrescentada uma margem de lucro que é definida de acordo com a necessidade de venda.

3. Metodologias de Desenvolvimento de *Framework*

Para a realização deste estudo, levantaram-se na literatura as abordagens que contemplavam um processo de definição arquitetural para o desenvolvimento de *frameworks* e *subframeworks*. Dentre elas, destacam-se as seguintes: Johnson (1993), Taligent (1994), Landin e Niklasson (1995), Pree (1999), Fayad et al. (1999), Mattson (2000), Butler e Xu (2001), Braga (2002), e por fim, Matos e Fernandes (2008).

Após o estudo das abordagens, realizou-se uma análise qualitativa entre elas, considerando-se os seguintes critérios: Criação de subsistemas (SU), *subframeworks* (SF) e componentes (C) – Possibilita a definição de uma arquitetura que facilita o entendimento do funcionamento do sistema, bem como seu reuso; Aplicação de padrões de projeto e metapadrões – Permite aumentar a reusabilidade na fase de projeto; Análise arquitetural – É um processo iterativo que permite que sejam feitos refinamentos no modelo arquitetural, que inicialmente foi escolhido de forma a contribuir para a criação de uma arquitetura que contemple algumas características julgadas interessantes, tais como: integridade, reusabilidade e flexibilidade. Para Butler e Xu (2001), esse processo é denominado de refatoração arquitetural; Estilo Arquitetural – Reduz o esforço para compreender o sistema desenvolvido por outra pessoa e, portanto, diminui a quantidade de informações que serão assimiladas num novo projeto permitindo, desta forma, facilitar o reuso em um novo [Bass et al. 2003]; Reuso Arquitetural – Traz benefícios que incluem redução de custo de construção e do tempo de produção; Definição dos *Hot Spots* nos *subframeworks* – Representam os pontos que são específicos para uma aplicação-exemplo e constituem a parte central do projeto de *framework* [Pree 1999].

Na Tabela 1, relacionam-se as abordagens citadas e os critérios. As linhas onde aparecem o “X” significa a presença do critério, caso contrário, a sua ausência. Observou-se que todas as abordagens contemplam a utilização de padrões de projeto e metapadrões durante a definição da arquitetura de *framework*. Com relação à Criação de Subsistemas, *Subframeworks* e Componentes verificou-se que somente a abordagem de Matos e Fernandes (2008) abrange mais detalhadamente seu processo de identificação e construção. Além disso, essa abordagem explica como a definição dos hot spots deve ser feita durante a construção dos *subframeworks* na fase inicial do seu processo de construção. Isso, de acordo com Tang et al. (2004), pode contribuir para a confecção de um sistema com um maior grau de flexibilidade e portabilidade, impactando sob os seus aspectos de desempenho, custo e escalabilidade.

Tabela 1 - Critérios contemplados pelas abordagens da literatura.

Critérios	Criação de:	Aplicação	Análise	Estilo	Reuso	Definição
-----------	-------------	-----------	---------	--------	-------	-----------

	SU	SF	C	de Padrões de Projeto e Metapa- drões	Arquite- tural	Arquite- tural	Arquite- tural	dos <i>hot spots</i> no <i>subframe- work</i>
Johnson (1993)				X				
Taligent (1994)	X	X		X	X		X	
Landin e Niklasson (1995)	X			X	X			
Pree (1995)				X				
Mattson (1996)	X	X		X	X	X		
Fayad et al. (1999)				X	X	X	X	
Butler e Xu (2001)	X		X	X	X		X	
Braga (2003)				X	X	X	X	
Matos e Fernandes (2008)	X	X	X	X	X	X	X	X

Constatou-se também que o critério de Análise Arquitetural é contemplado na maioria das abordagens e é considerado um fator importante no processo de definição da arquitetura do *framework*. Menos da metade das abordagens não contemplam a definição de um estilo arquitetural e que, segundo Paris (2004), a sua escolha implica em um aumento de flexibilidade do sistema, sendo assim um critério importante na definição da arquitetura. Por fim, o Reúso Arquitetural é contemplado por mais da metade das abordagens possibilitando o reúso de requisitos, projeto, métodos e de componentes de software.

Após a análise das abordagens, verificou-se que a de Matos e Fernandes (2008) era a ideal para ser aplicada no desenvolvimento do *subframework* para o controle de atributos, pois traz o processo detalhado para sua construção. Além disso, contempla também os benefícios trazidos pelas outras abordagens, dentre eles, podem-se citar os seguintes, entre outros: estilo arquitetural, aplicação de padrões de projeto e metapadrões.

4. Identificação dos Pontos de Estabilidade e Flexibilidade

O processo de desenvolvimento de *frameworks* de domínio corresponde a uma evolução iterativa de sua estrutura de classes, que envolve atividades como identificação de classes, modelagem de cenários e identificação de estados de objetos, entre outros.

Uma abordagem que pode ser utilizada para o desenvolvimento de *frameworks* é a abordagem PDR-DFD (Processo Dirigido a Responsabilidade aplicado ao Desenvolvimento de *Framework* de Domínio). Na PDR-DFD as responsabilidades são declarações gerais sobre as ações que um objeto executa e mantém, e sobre as decisões principais produzidas por um objeto que afeta os outros [Matos and Fernandes 2006].

Essas responsabilidades são encontradas a partir das ações que um sistema deverá executar e nas informações que necessitam ser mantidas ou criadas [Wirfs-Brock and McKeane 2003]. Além disso, a PDR-DFD foi construída com o objetivo de dar um contexto adequado de aplicação do método de identificação antecipada de pontos de estabilidade e de flexibilidade proposto por Matos e Fernandes (2008).

Para aplicação da abordagem dirigida por responsabilidades, selecionaram-se métodos no domínio de formação de preço de venda, tais como os seguintes: ABC [Cogan 1999 and Silvestre 2002], Custo Pleno [Martins 2003] e SEBRAE [Sebrae 2009]. Esses métodos representam as aplicações-exemplo para o desenvolvimento de *framework* e todos possuem um cadastro de atributos.

Depois de realizada a análise de cada método descrita em Crazuski et al. (2008), do subsistema “Controlar Atributos”, e a partir de cada Modelo e Requisitos e de Classes gerado para cada aplicação-exemplo analisada, pode-se levantar os *subframeworks* executando os passos descritos a seguir [Matos and Fernandes 2006]. Neste artigo será exibido o processo para o *subframework* Controlar Atributos.

O processo de criação obedeceu quatro passos descritos a seguir.

4.1. PASSO 1 – Análise do Modelo de Requisitos dos Métodos de Formação de Preço de Venda

Primeiramente foi analisado cada aplicação-exemplo descritas no trabalho de Crazuski et al. (2008) e verificado seu modelo de requisitos no formato UML, elaborou-se uma planilha com os seguintes dados: *Actor*, *Perspective*, *UseCase*, *Responsability* e *Exemplename*, onde:

- *Actor* – corresponde ao usuário ou sistema que interage com a aplicação;
- *Perspective* – equivale ao subsistema de onde o use case pertence, tais como: conexão, gerenciar produto, cadastro de atributos, validação, etc.;
- *UseCase* – ações a qual o ator pode executar;
- *Responsability* – provém a descrição textual do caso de uso. Por exemplo: A tabela 2 representa a descrição textual do caso de uso “Abrir Tela de Cálculo”.

Tabela 2. Descrição textual do Caso de Uso “Abrir Tela de Cálculo” [Crazuski, Feitosa and Cordeiro 2008]

Ator	Sistema
1. Usuário solicita abertura da Tela de Cálculo	2. Exibir Tela de Cálculo

- *Example name* – nome da aplicação-exemplo o qual a perspectiva ou subsistema pertence, por exemplo, para o domínio de formação de preço de venda se tem as seguintes aplicações-exemplo: ABC, SEBRAE e CUSTO PLENO.

A Figura 1 mostra apenas um trecho da planilha gerada para os métodos ABC, SEBRAE e CUSTO PLENO.

Actor	Perspective	UseCase	Responsability	Exemplename
Usuário	Gerenciar Linha de Produção	Consultar Linha	Exibir Tela de Consulta	ABC
Usuário	Gerenciar Linha de Produção	Consultar Linha	Exibir todas as linhas	ABC
Usuário	Gerenciar Linha de Produção	Consultar Linha	Verificar opção selecionada	ABC
Usuário	Gerenciar Linha de Produção	Consultar Linha	Exibir mensagem de erro	ABC
Usuário	Gerenciar Linha de Produção	Editar Linha	Exibe Tela de Cadastro	ABC
Usuário	Gerenciar Linha de Produção	Editar Linha	Exibe uma linha	ABC
Usuário	Gerenciar Linha de Produção	Editar Linha	Verificar opção selecionada	ABC
Usuário	Gerenciar Linha de Produção	Editar Linha	Exibir mensagem de erro	ABC
Usuário	Gerenciar Linha de Produção	Excluir Linha	Exibe mensagem de confirmação	ABC
Usuário	Gerenciar Linha de Produção	Excluir Linha	Verificar opção selecionada	ABC
Usuário	Gerenciar Linha de Produção	Excluir Linha	Excluir Linha	ABC
Usuário	Gerenciar Linha de Produção	Excluir Linha	Exibir mensagem de erro	ABC
Usuário	Gerenciar Linha de Produção	Cadastrar Linha	Exibir Tela de Cadastro	ABC
Usuário	Gerenciar Linha de Produção	Cadastrar Linha	Verificar opção selecionada	ABC
Usuário	Gerenciar Linha de Produção	Cadastrar Linha	Inserir Linha	ABC
Usuário	Gerenciar Linha de Produção	Cadastrar Linha	Exibir mensagem de erro	ABC
Usuário	Gerenciar Linha de Produção	Cadastrar Linha	Limpar campos	ABC
Usuário	Gerenciar Produto	Consultar Produto	Exibir Tela de Consulta	ABC
Usuário	Gerenciar Produto	Consultar Produto	Exibir Todos os Produtos	ABC
Usuário	Gerenciar Produto	Consultar Produto	Verificar opção selecionada	ABC
Usuário	Gerenciar Produto	Consultar Produto	Exibir mensagem de erro	ABC

Figura 1. Planilha contendo os dados retirados do caso de uso UML de cada Aplicação-Exemplo

4.2. PASSO 2 – Análise Semântica dos Nomes das Responsabilidades

Após a elaboração da planilha 1, foi realizada uma análise semântica dos nomes das responsabilidades, pois as mesmas poderiam conter nomes diferentes mas que representavam a mesma funcionalidade e vice-versa. Das 201 responsabilidades dos modelos de requisitos, houve alteração de nomes em 21 delas, representando um total de aproximadamente 10,5% de problema semântico. A figura 2 ilustra essa planilha em que consta o nome da responsabilidade antiga (coluna Responsabilidade), seu novo nome é baseado em qual método de formação de preço de venda foi estabelecido esse nome.

Quantidade	Método	Responsabilidade	Novo Nome	Baseado no Método
1	ABC	Exibir todos dados dos atributos	Exibir todos os atributos	ABC
2	ABC	Exibir Tela de Cadastro	Exibir Tela de Cadastro de Atributos	SEBRAE
3	ABC	Exibir o preço de venda	Exibir resultado na Tela de Cálculo	SEBRAE
4	ABC	Consultar valores	Buscar valores dos atributos cadastrados no Banco de Dados	SEBRAE
5	ABC	Verificar opção selecionada	Verificar ação escolhida	PLENO
6	ABC	Exibir todos os atributos	Mostrar os atributos cadastrados no Banco de Dados	PLENO
7	ABC	Exibir um atributo	Mostrar os dados do Atributo selecionado	PLENO
8	SEBRAE	Listar os valores já persistidos anteriormente no Banco de Dados	Exibir todos dados dos atributos	ABC
9	SEBRAE	Exibir Tela de Alimentação de Valores	Exibir Tela de Cadastro	ABC
10	SEBRAE	Emitir mensagem de confirmação de exclusão	Exibe mensagem de confirmação	ABC
11	SEBRAE	Efetuar Cálculo do Preço de Venda	Calcular o preço de venda baseado nos valores informados	PLENO
12	SEBRAE	Verificar preenchimento de valores	Validar dados	PLENO
13	SEBRAE	Verificar qual ação executar	Verificar ação escolhida	PLENO
14	SEBRAE	Listar os valores já persistidos anteriormente no Banco de Dados	Mostrar os atributos cadastrados no Banco de Dados	PLENO
15	SEBRAE	Buscar dados do registro no Banco de Dados	Mostrar os dados do Atributo selecionado	PLENO
16	PLENO	Mostrar a tela de Consulta de Atributos	Exibir Tela de Consulta	ABC
17	PLENO	Mostrar a tela de Consulta de Atributos	Exibir Tela de Consulta	ABC
18	PLENO	Confirmar com o Usuário se ele realmente deseja excluir o atributo selecionado	Exibe mensagem de confirmação	ABC
19	PLENO	Calcular o preço de venda baseado nos valores informados	Efetuar Cálculo do Preço de Venda	SEBRAE
20	PLENO	Mostrar o preço de venda calculado	Exibir resultado na Tela de Cálculo	SEBRAE
21	PLENO	Mostrar a Tela de Cadastro de Atributos	Exibir Tela de Cadastro de Atributos	SEBRAE

Figura 2. Responsabilidades que sofreram alterações em seus nomes

4.3. PASSO 3 – Identificação dos *subframework*

Após a análise do modelo de requisitos dos métodos de preço de venda, o próximo passo foi realizar a identificação dos possíveis *subframework*. Dentre eles, pode-se citar: Analisador Sintático, Analisador Semântico, Controlar de Atributos, entre outros. Neste trabalho, será descrito o modelo de requisitos do Controlar Atributos.

A lista de requisitos deste *subframework* é:

- Cadastrar Atributo;
- Consultar Atributo;
- Editar Atributo;
- Excluir Atributo;
- Conectar ao Banco de Dados;
- Desconectar do Banco de Dados.

Com os requisitos já definidos, foi elaborado um diagrama de caso de uso no formato UML-F [Fontoura et al. 2000] para o *subframework* “Controlar Atributo”, figura 3, baseando-se nos modelos já definidos por Crazuski, Feitosa e Cordeiro (2008) e pela análise citada anteriormente.

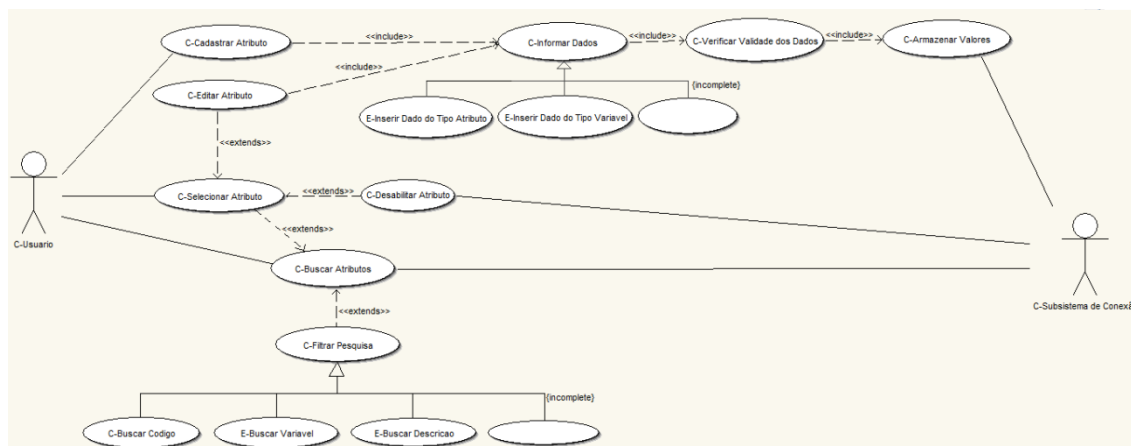


Figura 3. Diagrama de Caso de Uso UML-F do subsistema Cadastrar Atributo

As notações utilizadas na figura 3 são: caso de uso base (frozen spot), caso de uso específico ou flexível (hot spot), ator base e uma herança entre casos de uso em uma aplicação reusando o *framework*.

Os casos de uso base estão todos representados com a letra “C” antecedendo o nome do caso de uso, por serem aspectos comuns entre as aplicações concretas do domínio, os casos de uso específicos seguem a mesma regra, porém são representados pela letra “E” antecedendo o nome do caso de uso.

Ambos os atores são atores bases, ou seja, atores comuns entre as aplicações concretas do domínio, e são representados com a letra “C” antes do seu nome (exemplo: C-Usuario).

Outra notação utilizada neste diagrama é o conceito de herança entre casos de uso, representado por dois casos de uso interligados por um “*{incomplete}*”, o que significa que poderá futuramente ter novas heranças, deixando assim o *framework* mais adaptável.

5. Conclusões

Neste trabalho apresentou-se a utilização da abordagem dirigida por responsabilidades durante a modelagem do *subframework* “Controlar Atributos” a partir dos casos de uso definidos como *hot spots* e obtidos na etapa de análise do processo de desenvolvimento de *framework*.

Como parte do resultado do processo proposto, criou-se a modelagem de um *subframework* que pode ser utilizado em outra aplicação, pois seus casos de uso *frozen* e *hot spots* já estão identificadas. Contudo, a implementação do *subframework* está sendo concluída.

7. Referências

- Bass, L. and Clements, P. and Kazman, R. (2003) “Software Architecture in Practice, Addison Wesley”.
- Bornia, A. C. (2002) “Análise gerencial de custos – aplicação em empresas modernas”. Porto Alegre: Bookman.
- Braga, R. T. V. and Masiero, P. C. (2002) “A Process for Framework Construction Based on a Pattern Language”. In: Annual International CSAC, 26., 2002. Oxford: IEEE. p. 615-620.
- Butler, G. and Xu, L. (2001) “Cascaded Refactoring for Framework Evolution”, In: Proc. SSR’01, May, Toronto, Ontario, Canada. ACM 1-58113-358-8/01/0005, p. 51-57.
- Cogan, S. (1995) “Activity-Based Costing (ABC)” – a ponderosa estratégia empresarial. São Paulo: Pioneira.
- Crazuski, A., Feitosa, L. B. and Cordeiro, T, L. (2008) “Identificação dos pontos de estabilidade e de flexibilidade dos métodos para o estabelecimento de preço de venda”. Ponta Grossa, PR. 66 f. Trabalho de Conclusão de Curso (Graduação) - UTFPR. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Ponta Grossa.
- Fayad, M., Schmidt, D. and Johnson, R. (1999) “Building Application Frameworks – Object-Oriented Foundations of Framework Design”, Wiley, p.688.
- Fontoura, M., Pree, W. and Rumpe, B. (2000) “UML-F:A modeling language for object-oriented frameworks”, In ECOOP. LNCS, 1850. Springer-Verlag, 63-82.
- Johnson, R. E. (1993) “How to design frameworks”, In: Object-Oriented Programming Systems, Languages and Applications Conference – OOPSLA, Washington proceedings.
- Kirk, D., Roper, M. and Wood, M. (2005) “Identifying and Addressing Problems in Framework Reuse”, In: Proceedings of the 13th IWPC’05.

- Land, R. (2002) “How can frameworks facilitate component reuse?”, Extended Report for I. Crnkovic and M. Larsson (editors), “Building Reliable Component-Based Systems”, Artech House.
- Landin, N. and Niklasson, A., (1995) “Development of Object-Oriented Frameworks”, Department of Communication System.Lund Institute of Technology, Lund University. Lund, Sweden.
- Martins, E. (2003) “Contabilidade de Custo”. 6ª Ed., São Paulo: Atlas. P.220
- Matos, S. N. and Fernandes, C. T. (2006) “Defining the architectural design of frameworks through a group of *subframeworks* created from frozen and hot spots”. In: International Conference on Software Engineering Advances, 2006, Tahiti: IEEE Computer Society Press.
- Matos, S. N. and Fernandes, C. T. (2008) “Using Responsibilities for Early Identification of Hot Spot Reused in Framework Modeling”. In: IEEE International Workshop on Security, Trust, and Privacy for Software Applications. 3. Turku: IEEE Computer Society Press.
- Mattsson, M. (2000) “Evolution and composition of object-oriented frameworks”. 224f. PhD Thesis– University of Karlskrona/Ronneby, Sweden.
- Nakagawa, M. (1995) “ABC – Custeio baseado em atividades”. São Paulo: Atlas.
- Nascimento, D. T. do and Vartanian, G. H. (1999) “O método pleno” – uma abordagem conceitual. Revista CRC-SP.
- Paris, M. (2004) “Reuse-based Layering: a Strategy for Architectural Framework for Learning Technologies”, In: IEEE Conferece on Advanced Learning Technologies (ICALT’04).
- Pree, W. (1999) “Hot-spot-driven development”. In: Fayad, M., Johnson, R. and Schmidt, D. “Building application frameworks: object-oriented foundations of framework design”. NY: John Wiley and Sons. p. 379-393.
- Sebrae. (2009) “Cálculo do preço de venda”. Disponível em <http://www.sebraepr.com.br>, October.
- Silva, C. L. da. (2001) “Competitividade: mais que um objetivo, uma necessidade”. Revista FAE BUSINESS, Blumenau, nº1.
- Silvestre, W. C. (2002) “Sistema de custos ABC: uma visão avançada para tecnologia de informação e avaliação de desempenho”. São Paulo: Atlas.
- Taligent, (1994) “Building object-oriented frameworks”, Taligent Inc.white paper.
- Tang, A., Han, J. and Chen, P. (2004) “A Comparative Analysis of Architecture Frameworks”, In: Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC’04).
- Wirfs-brock, R. and Mckean, A. (2003) “Object Design: Roles, esponsabilities, and Collaborations”, Addison Wesley.