Modelagem de uma Aplicação Web a partir de um *Framework* de Agenda de Tarefas

Sergio A. Tanaka[1], Patrícia M. Rodrigues[1] ,Rodolfo M. de Barros[2], Ruy. T. Nishimura[1]

[1]UNIFIL – Centro Universitário Filadélfia – Depto. de Informática
Av. JK 1626, Londrina, PR, Brasil, 86020-000
[2]Departamento de Computação, Universidade Estadual de Londrina
Londrina, PR, CEP 86051-990, Brasil

{tanaka, ruy}@dc.unifil.br, rodolfo@uel.br, ter-patriciar@weg.com.br,

Abstract. This paper presents the modeling of a Web application from one framework of agenda of tasks for process manager. The framework was developed on the basis of the Catalysis method and was validated by means of a prototype implemented in the Java language. Beyond the Web modeling, is presented a component architecture of framework and the modeling of data base generated from the project. The standard of modeling used was UML with the tool CASE IBM Rational Rose. This work contributes for definition and construction of architectures for development based on components and the construction of a component since the phase of requirements until the generation of the codification.

Resumo. Este artigo apresenta a modelagem de uma aplicação Web a partir de um *framework* de agenda de tarefas para gerenciadores de processos. O *framework* foi desenvolvido com base no método Catalysis e foi validado por meio de um protótipo implementado na linguagem Java. Além da modelagem Web, é apresentado uma arquitetura componentizada do *framework* e a modelagem de banco de dados gerada a partir do projeto. O padrão de modelagem utilizada foi a UML com a ferramenta CASE IBM Rational Rose. Este trabalho contribui para definição e construção de arquiteturas para desenvolvimento baseado em componentes desde a fase de requisitos até a geração do banco de dados e codificação.

Palavras Chaves: *workflow*, arquitetura de software, *framework*, modelagem de aplicação Web com UML.

1. Introdução

A engenharia de software tem por objetivo a qualidade dos sistemas de software, isso significa, maior confiabilidade, redução dos riscos de processo, conformidade com padrões, extensibilidade e flexibilidade. Ao mesmo tempo, a engenharia de software busca reduzir os custos e esforços, aumentando a produtividade no desenvolvimento dos produtos.

O ponto chave para obtenção dos objetivos é o reuso de componentes, que permite a construção de software utilizando *frameworks* e modelos bem definidos e testados.

A técnica de desenvolvimento baseado em componentes é uma abordagem baseada no reuso que emprega técnicas de *frameworks*, baseadas em padrões. O uso de padrões de software permite disseminar soluções de desenvolvimento de software já existentes.

O método Catalysis incorpora conceitos relevantes que apóiam o desenvolvimento de software para sistemas baseados em objetos e componentes, baseia-se na linguagem de modelagem UML e alinha-se aos padrões de sistemas de objetos distribuídos.

O *framework* utilizado neste trabalho, foi o de agenda tarefas desenvolvido por Tanaka [TAN 2000] e tomou como base o padrão de arquitetura *Process Manager* desenvolvido no projeto ExPSEE[1] [GIM 1999] e a análise de aplicações de gerenciamento de *workflows*. O padrão de modelagem utilizada foi a UML (*Unified Modeling Language*) com a ferramenta CASE IBM Rational Rose. Este trabalho contribui para definição e construção de arquiteturas para desenvolvimento baseado em componentes no ambiente Web e a construção de um componente desde a fase de requisitos até a geração do banco de dados e codificação utilizando uma ferramenta CASE do início ao término do projeto.

A seção 2 apresenta a análise do domínio do *framework* de agenda de tarefas. A seção 3 apresenta a arquitetura de software do projeto. A seção 4 apresenta a modelagem do banco de dados, a modelagem Web do *framework* de agenda de tarefas e o diagrama de classes. Finalmente, na seção 5 são apresentadas as conclusões e trabalhos futuros.

2. Análise do Domínio do *Framework* da Agenda de Tarefas

O estudo de caso utilizado foi um *framework* de agenda de tarefas para gerenciadores de processos. Este *framework* pode ser utilizado em aplicações que envolvem o gerenciamento de tarefas tais como: aplicações de *workflow*, ambientes de engenharia de software e gerenciadores de projeto [TAN 2000], usualmente presentes em PSEEs, WfMSs (*Workflow Manager System*) ou em domínios correlatos, tais como: agendamento de obras para construção civil e agendamento do conteúdo de disciplinas em cursos.

Segundo [TAN 2000], pode-se desenvolver *frameworks* pequenos para cada função importante em uma aplicação ou domínio, com base em um *framework* mais genérico. O diagrama de caso de uso mostrado na figura 1 exibe o comportamento da agenda de tarefas, cujo cenário tem como ator principal o gerente de projeto interagindo diretamente com o *framework* de agenda de tarefas.

[1] *ExPSEE – Experimental Process-Centred Software Engineering Environment*, é um projeto desenvolvido no Laboratório de Engenharia de Software do Departamento de Informática da Universidade Estadual de Maringá, financiado pelo CNPq de 1994 a 2002

Figura 1 – Diagrama de Caso de Uso de Agenda de Tarefas

3. Arquitetura de Software

A arquitetura de um sistema de software engloba a definição de suas estruturas gerais, descrevendo os elementos que compõem os sistemas e as interações entre estes. Além de descrever esses elementos e suas interações, uma arquitetura de software apóia também questões importantes de projeto, tais como: a organização do sistema como composição de componentes, as estruturas de controle globais, os protocolos de comunicação, a composição dos elementos do projeto e a designação da funcionalidade dos componentes do projeto [GIM 1999].

O sistema em camadas divide as funcionalidades e regras de negócio em componentes manejáveis que estão inseridos em diferentes pacotes. A figura 2 apresenta o diagrama da arquitetura do sistema, onde os componentes do sistema estão separados em três camadas de serviço: o *User Services*, *Business Services* e *Data Services,* que são camadas lógicas que colaboram na aplicação.

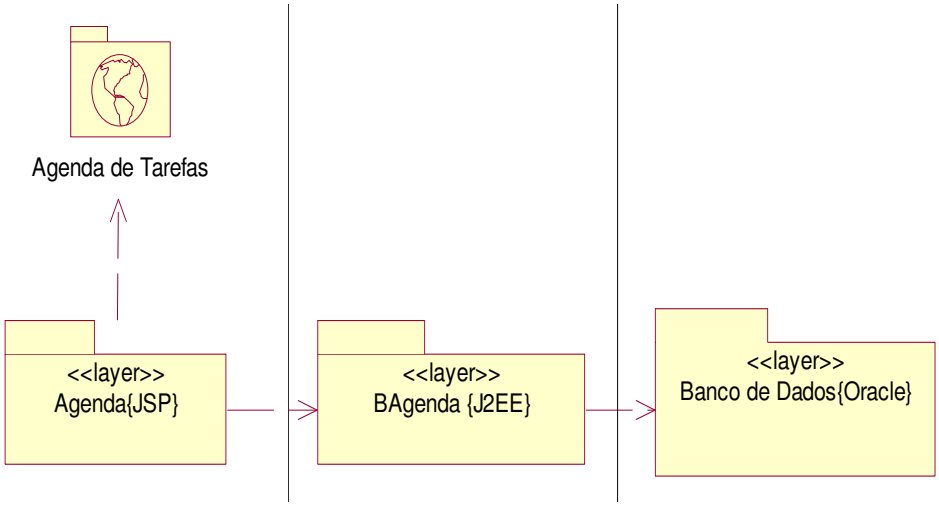

Figura 2 – Estilo Arquitetural do *Framework* de Agenda de Tarefas.

Uma parte importante de qualquer processo de projeto é a especificação das *interfaces* entre os diferentes componentes no projeto [SOM 2003]. Um componente se

comunica com outros componentes através de suas *interfaces* e estas não podem ser modificadas por usuários.

4. Modelagem do *Framework*

A modelagem do *framework* foi feita utilizando o padrão UML. Nesta seção são apresentados os diagramas de Classes, Banco de Dados e modelagem Web.

A aplicação WEB executa num ambiente distribuído, onde cada parte que compõe o programa está localizada em uma máquina diferente, uma arquitetura multicamadas onde as funções executadas pelas aplicações podem estar distribuídas [SOM 2003].

Na UML, cada página Web é uma classe do modelo *Design View* (*Logical View*) e os seus relacionamentos com outras páginas (associações) representam *hiperlinks*. Mas esta abstração falha se considerarmos que qualquer página Web pode potencialmente representar uma série de funções e colaborações que existem no lado servidor e outras completamente diferentes que existem somente no lado cliente [CON 1999].

Pode-se então, modelar os aspectos do lado servidor de uma página Web como uma classe, e o lado cliente como outra. Os mecanismos de extensão da UML permite representa-las através dos seus estereótipos como ícones customizados ou com as *tags* simples, o nome do estereótipo entre *guillemets* (<<>>), quando os atributos e operações das classes são expostos (Figura 3).

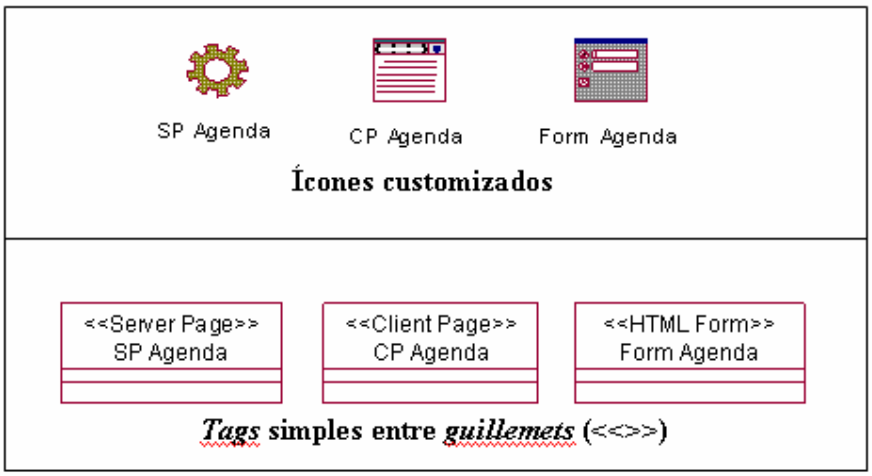

Figura 3 – Mecanismos de extensão da UML

As classes estereotipadas podem ser representadas em um diagrama UML ao demonstrar a visão geral conforme é exibido na figura 4. O padrão de nomenclatura adotado para as classes neste diagrama, prevê o uso de prefixos que identificam o tipo da classe, assim, tem-se: SP para Server Page; CP para Client Page e Form para formulários HTML.

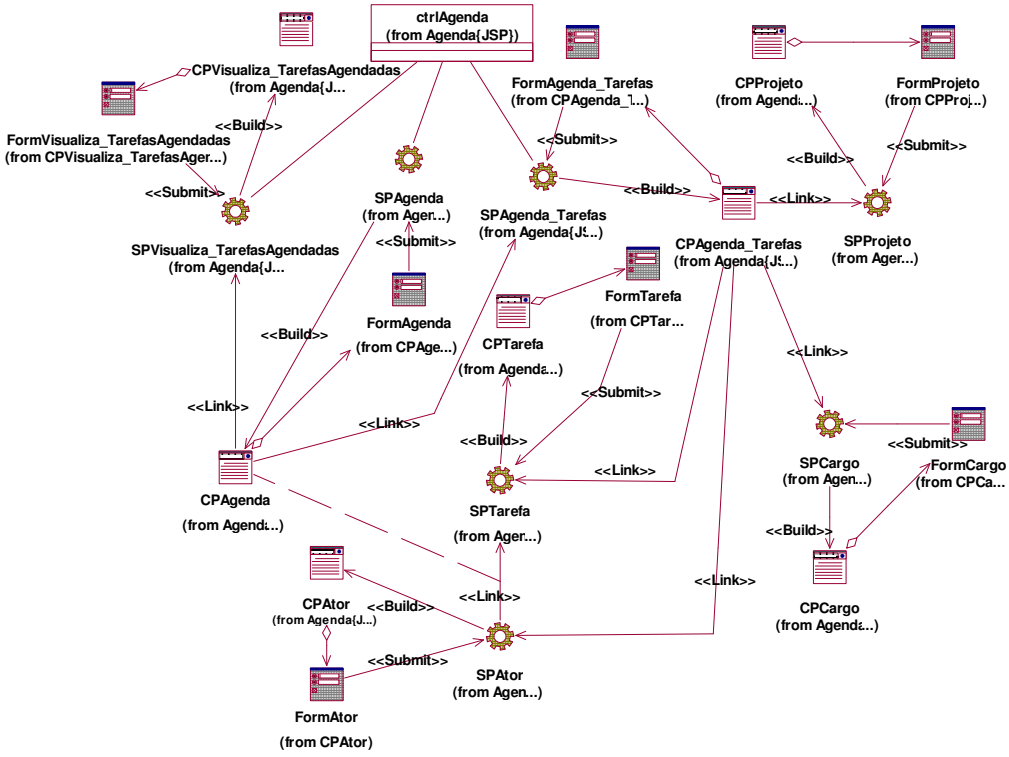

Figura 4 – Diagrama de Classes Web do Framework de Agenda de Tarefas.

O processo de execução do formulário da agenda inicia chamando o objeto SPAgenda que possui o estereótipo <<server page>>, este por sua vez, é responsável pela construção do <<client page>> CPAgenda. A associação entre estes objetos é demonstrada através do estereótipo <<build>>. A *Client Page* contém um formulário (<<form>>) agregado chamado FormAgenda. Resumindo, o SPAgenda constrói a página Web CPAgenda com um formulário FormAgenda, que mostra a agenda de determinado usuário. Quando os dados são enviados do formulário para o S*erver Page* utiliza-se uma associação estereotipada <<submit>> e o uso do estereotipo <<link>> representa o relacionamento entre páginas. Esta associação geralmente é originada de uma *Client Page* para outra C*lient Page* ou para um S*erver Page*.

A seguir, tem-se a modelagem das classes do *framework* de agenda de tarefas exibido pela figura 5.

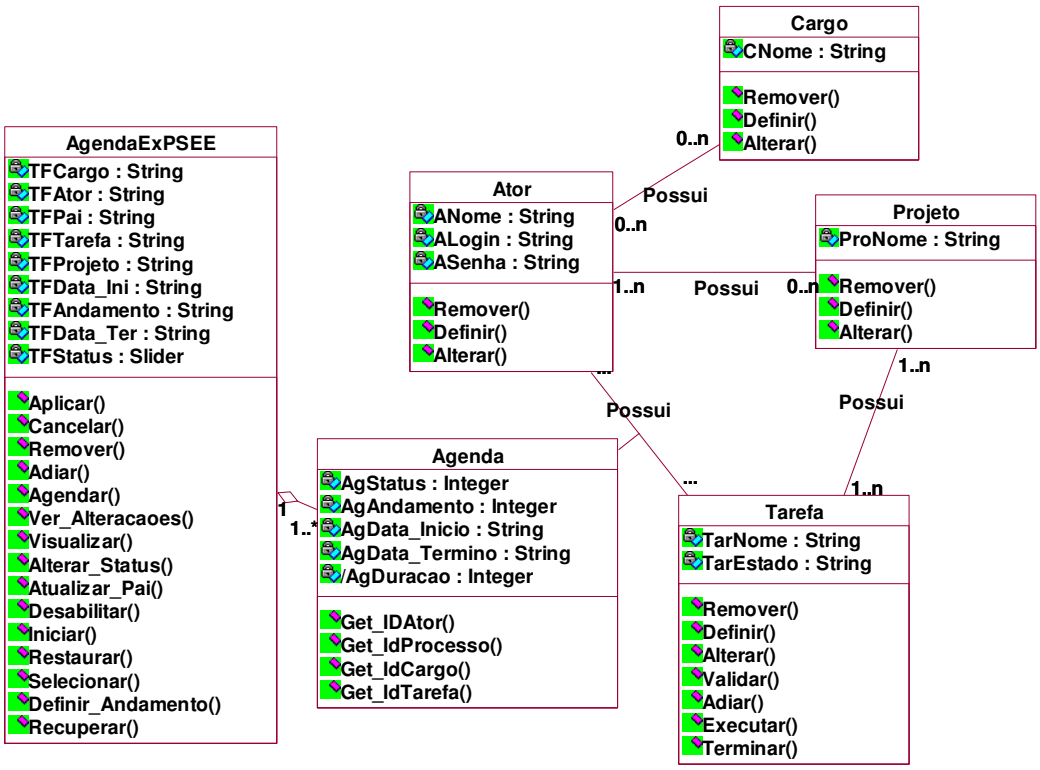

Figura 5 – Diagrama de Classes do *Framework* de Agenda de Tarefas.

Para consolidar a modelagem do *framework*, proposto através de uma extensão da UML para aplicações Web, foi também adotada esta linguagem para fazer a modelagem de banco de dados.

Como o banco de dados armazena as informações na estrutura relacional, as classes persistentes definidas na UML podem ser traduzidas no modelo relacional como tabelas do SGBDR (Sistema Gerenciador de Banco de Dados Relacional).

A figura 6 ilustra o diagrama de banco de dados gerado através da ferramenta Rational Rose.

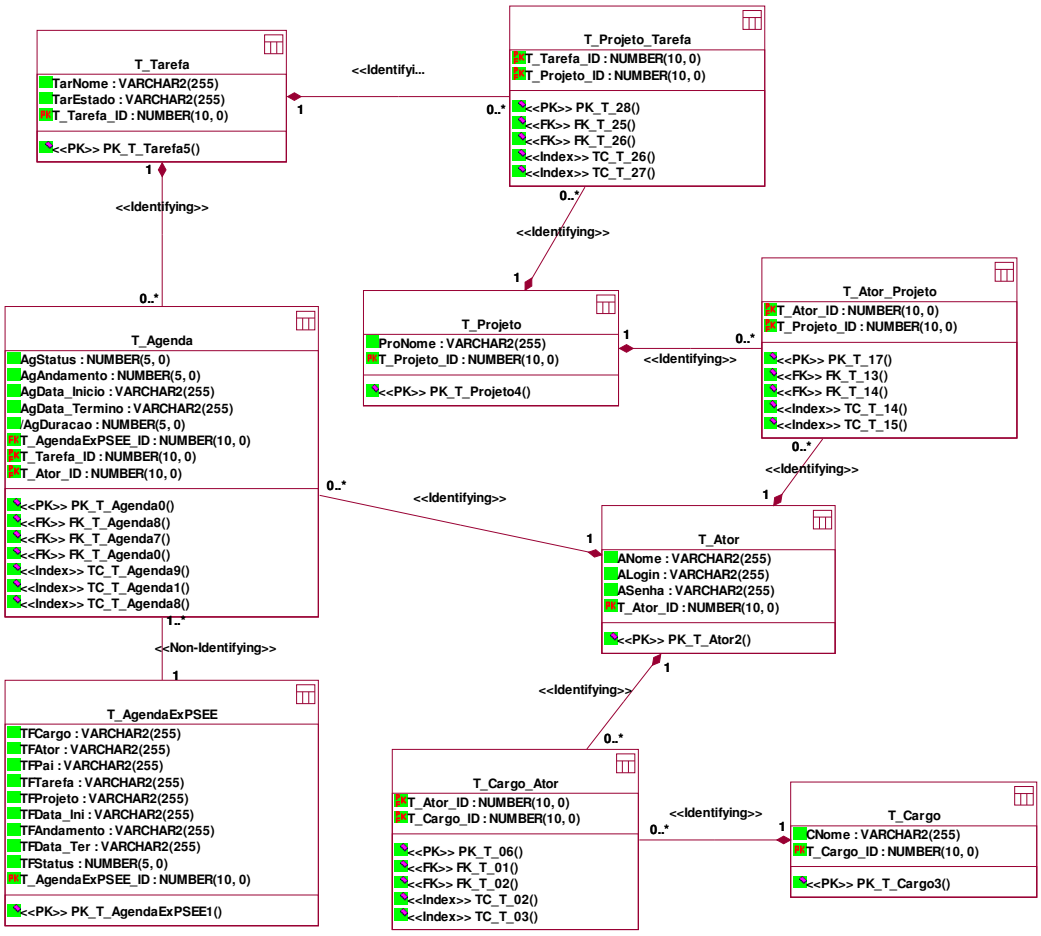

Figura 6 – Diagrama de Banco de Dados

5. Protótipo

A agenda de tarefas possui interfaces visuais, páginas Web, das quais os atores interagem para visualizar as tarefas agendadas, agendar tarefas, incluir um novo ator, cargo, projeto e tarefa.

Para validar o estudo, um protótipo foi desenvolvido em ASP.NET e banco de dados Oracle 8. A figura 7 exibe a interface da tela Visualizar Tarefas Agendadas que mostra atributos como: data início e término da execução das tarefas, a tarefa pai, sendo esta a raiz ou o nó principal, o andamento da tarefa e seus estados. A interface apresenta também funcionalidades através de botões, tais como: aplicar, cancelar, agendar, remover, adiar, mostrar estados, visualizar alterações, visualizar agenda individual das tarefas e sair.

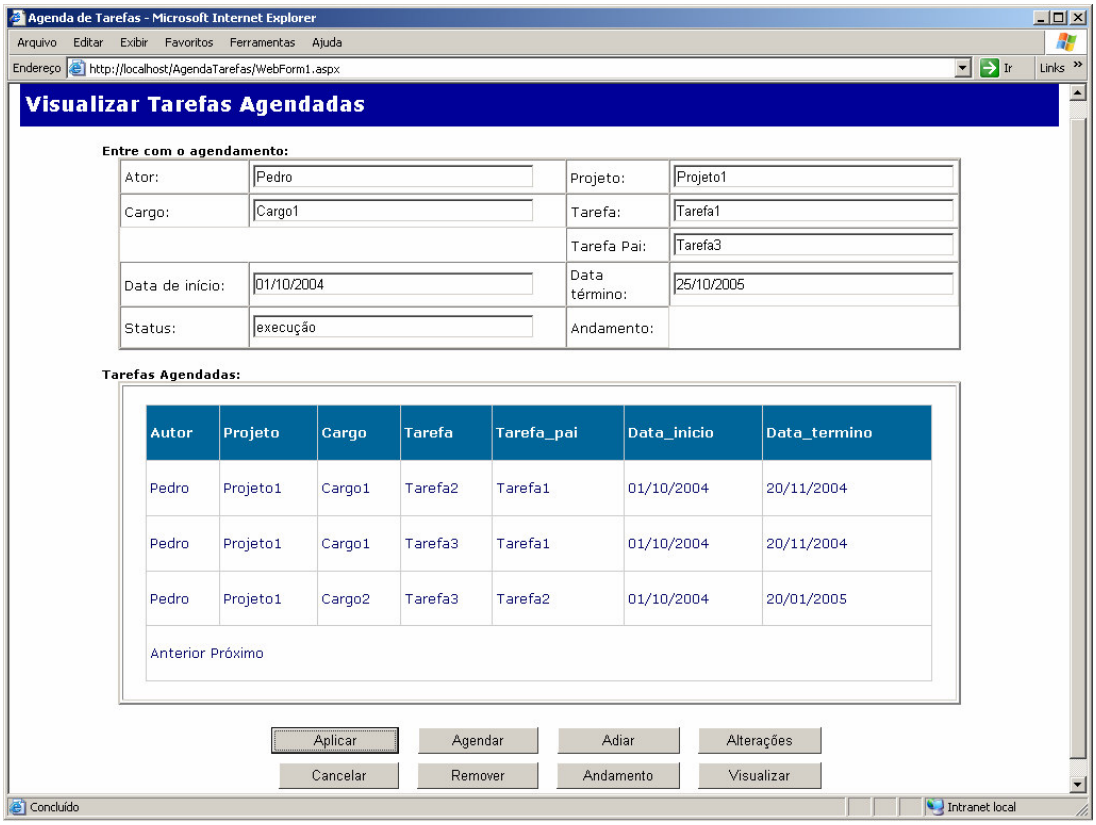

Figura 7 - Interface de a tela Visualizar Tarefas Agendadas

6. Conclusões e Trabalhos Futuros

As aplicações Web necessitam de um planejamento para modelagem de sistemas através de abordagens, tais como a UML, que permitem maior consistência e clareza na definição da arquitetura. Este artigo teve como finalidade apresentar a modelagem de uma aplicação Web, utilizando o desenvolvimento baseado em componentes (DBC) e a modelagem de banco de dados em um estudo de caso de um *framework* de agenda de tarefas.

Na extensão para a Web, as páginas clientes ou servidoras são tratadas igualmente como classes. Sua diferenciação é feita por meio de ícones (forma visual) e estereótipos (forma descritiva), que descrevem o comportamento lógico destas páginas.

O desenvolvimento baseado em componentes tem como principal objetivo a reutilização em diferentes aplicações para obter maior produtividade no desenvolvimento, redução de custos e maior confiabilidade, uma vez que são construídos a partir de partes bem especificadas e testadas.

O uso de ferramenta CASE no processo de desenvolvimento é essencial. Toda a modelagem foi feita utilizando a ferramenta CASE IBM Rational Rose que facilmente atualiza a modelagem, código fonte e define as tabelas de banco de dados a partir do diagrama de classes e gera o script completo que ao ser executado no banco de dados, gera todas as tabelas com as respectivas chaves primárias e secundárias,

constraints, *triggers* e relacionamentos, sem nenhum esforço do administrador de banco de dados.

6. Referências Bibliográficas

[CON 2004] CONALLEN, J. **Modeling Web application architectures with UML**. Disponível em http://www.rational.com/media/whitepapers/webapps.pdf. Acessado em 13 de Setembro de 2004.

[D'SO 1999] D'SOUZA, D.; WILLS, A. **Objects, Components and Frameworks with UML - The Catalysis Approach.** USA: Addison-Wesley, 1999.

[GIM 1999] GIMENES, I. M. S. et al. **ExPSEE – An Experimental Process Centred Software Engineering Environment.** Maringá: UEM/CTC/ DIN, 1999.

[SOM 2003] SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Addison Wesley, 2003.

[TAN 2000] TANAKA, S. **Um *framework* de Agenda de Tarefas para Gerenciadores de Processos**. Porto Alegre, 2000. 124p. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul.

[TAN 2001] Tanaka, Sergio Akio; Nishimura, Ruy T; Olguin, Carlos José M. The Use of the Catalysis Approach in the Construction of a Framework in an N-tier Architecture. In: the XXVIII Latin-American Conference on Informatics (CLEI 2002). Montevideo-Uruguay, 2002.