

Beans Composer: uma ferramenta para manipular componentes de processo

**Andréa Magalhães Magdaleno
Vanessa Tavares Nunes**

Projeto Final de Curso submetido ao Departamento de Ciência da Computação do Instituto de Matemática da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Bacharel em Informática.

Apresentado por:

Andréa Magalhães Magdaleno

Vanessa Tavares Nunes

Aprovado por:

Profa. Renata Mendes Araujo (UNIRIO e NCE/UFRJ), D. Sc.

Profa. Fernanda Araújo Baião Amorim (COPPE/UFRJ), D. Sc.

Prof. Alessandro de Almeida Castro Cerqueira (UNIGRANRIO), MSc

RIO DE JANEIRO, RJ - BRASIL
JANEIRO DE 2004

Agradecimentos

Às nossas famílias por todo amor, respeito e carinho que nos dedicaram, nos orientando ao longo das nossas vidas e contribuindo para esta realização.

À amiga e orientadora Renata Araujo, pela disponibilidade, atenção e paciência dedicadas a este trabalho, sendo uma constante fonte de equilíbrio.

Aos professores Marcos Borges, Fernanda Baião e Alessandro Cerqueira, que aceitaram prontamente o convite para participar da banca avaliadora deste trabalho.

Ao Marcos, que soube entender e aceitar todas as vezes que deixava de estar com ele para me dedicar um pouco mais a este trabalho, além de sempre se mostrar interessado em discutir e disponível para contribuir com sugestões.

Ao Leandro, pela compreensão e carinho, em todas as vezes que tive que deixá-lo para me dedicar a este projeto.

Aos nossos amigos, que colaboraram com informações, incentivo ou simplesmente boa conversa para que conseguíssemos concluir este trabalho.

RESUMO

***Beans Composer*: uma ferramenta para manipular componentes de processo**

**Andréa Magalhães Magdaleno
Vanessa Tavares Nunes**

Orientadora: Renata Mendes Araujo

Co-Orientador: Marcos Roberto da Silva Borges

As organizações estão cada vez mais interessadas em melhorar os seus processos de negócio. Para isso, algumas delas têm investido na modelagem e posterior automação de seus processos utilizando como estratégia a adoção da tecnologia de *workflow*. Neste cenário, uma das dificuldades encontradas é a adaptação do processo definido na organização para as diferentes necessidades que surgem em seu cotidiano. Este trabalho parte do princípio de que é possível definir componentes de processos implementados em sistemas de *workflow* que auxiliarão a definição e uso automatizado de processos em organizações. Para atender este objetivo foi desenvolvida uma ferramenta para manipulação destes componentes denominada *Beans Composer*.

ABSTRACT

Beans Composer: a tool to manipulate process beans

**Andréa Magalhães Magdaleno
Vanessa Tavares Nunes**

Supervisor: Renata Mendes Araujo

Co-Supervisor: Marcos Roberto da Silva Borges

Organizations are increasingly interested in improving their business processes. In order to do this, some of them have invested in the modeling and posterior automation of its processes using as a strategy the adoption of the workflow technology. One of the difficulties in this approach is how organizations adapt their defined processes for each different use. This work argues that it is possible to define components implemented in workflow systems that will assist the process definition, adaptation and use in organizations. A tool called Beans Composer was implemented in order to achieve this objective.

Sumário

1. Introdução.....	1
2. Modelagem de Processos em Organizações.....	3
2.1. Motivação	3
2.2. Conceitos	4
2.3. Objetivos.....	6
2.4. Processo de <i>Software</i>	7
2.5. Automação de Processos.....	10
3. Workflow.....	12
3.1. Histórico	12
3.2. Definição	15
3.3. Conceitos	16
3.4. Funcionalidades.....	19
3.5. <i>Lotus Workflow</i>	24
3.6. Benefícios	33
3.7. Desafios	36
4. Process Beans	37
4.1. Motivação	37
4.2. Definição	38
5. Ferramenta	43
5.1. Objetivos.....	43
5.2. Funcionalidades.....	44
5.3. Exemplo	48
5.4. Aspectos de Implementação	53
6. Conclusão.....	57
Referências	59
Anexos.....	63

Anexo I – Diagrama de Casos de Uso	64
Anexo II – Caso de Uso Manter Informações de Classificação de <i>Bean</i>	65
Anexo III – Caso de Uso Manter Informações de <i>Beans</i>	68
Anexo IV – Caso de Uso Manter Informações de Projeto.....	72
Anexo V – Caso de Uso Manter Informações de Etapa.....	76
Anexo VI – Caso de Uso Consultar Projeto	81
Anexo VII – Preencher Atividade Aberta.....	82
Anexo VIII – Caso de Uso Consultar Etapa	84
Anexo IX – Ativar Projeto.....	85
Anexo X – Visualizar Processo no <i>Architect</i>	87
Anexo XI – Caso de Uso Manter Informações de Atividade Aberta.....	88
Anexo XII – Manual do Usuário.....	90
Anexo XIII – Exemplo Código XML	97
Anexo XIV – <i>Bean</i> de Ciclo de Vida em Cascata.....	102
Anexo XV – <i>Bean</i> de Teste	104

Índice de Figuras

Figura 2.1 – Elementos de Modelo de Negócio da Organização (PROFORMA, 2002)	5
Figura 2.2 – Atividades do ciclo de vida de um Processo de Software Típico (PRESSMAN, 1997)	7
Figura 2.3 – Processo de Requisitos (RUP, 2002).....	8
Figura 2.4 – Atividade Analisar o Problema (RUP, 2002)	9
Figura 2.5 - Processos de Negócio Automatizados por WfMS (ARAUJO e BORGES, 2001)	10
Figura 2.6 – Exemplo da Integração Promovida por Sistemas de Workflow (SILVA, 2001)	11
Figura 3.1 – Evolução da Gerência de Documentos em Organizações	13
Figura 3.2 – Roteamento de Informações e Tarefas entre Membros de uma Organização (ARAUJO e BORGES, 2001)	14
Figura 3.3 - Elementos em um Sistema de Gerenciamento de Workflow (ARAUJO e BORGES, 2001).....	16
Figura 3.4 – Tipos de Rotas (CRUZ, 2000).....	18
Figura 3.5 – Exemplo de Definição de Processo – Processo de Solicitação de Compras	19
Figura 3.6 – Definição de processo e suas instâncias em execução (ARAUJO e BORGES, 2001).....	20
Figura 3.7 – Arquitetura do Lotus Workflow	25
Figura 3.8 – Exemplo de Mapa de Processo no Lotus Workflow Architect (Lotus Workflow)	26
Figura 3.9 – Owner de Atividade no Lotus Workflow Architect (Lotus Workflow).....	27
Figura 3.10 – Pacote de Trabalho no Lotus Workflow Architect (Lotus Workflow).....	28
Figura 3.11 – Associação de Recursos a Tarefas no Lotus Workflow Architect (Lotus Workflow).....	28
Figura 3.12 – Visão My Activities da Base de Aplicação (Lotus Workflow).....	29

Figura 3.13 – Execução da Tarefa de Criar Projeto no RequisitePro (Lotus Workflow).....	30
Figura 3.14 – Cadastro de Participante na Base de Organização (Lotus Workflow).....	31
Figura 3.15 – Grupo de Usuários na Base de Organização (Lotus Workflow).....	32
Figura 3.16 – Interface do Lotus Workflow Viewer destacando as atividades já executadas e o trâmite dos documentos (Lotus Workflow).....	33
Figura 4.1 – Ciclo de Vida em Cascata (PRESSMAN, 1997).....	39
Figura 4.2 – Exemplo de Combinação de Beans.....	40
Figura 4.3 – Exemplo de Combinação de Beans na Engenharia de Software.....	41
Figura 5.1 – Biblioteca Geral de Processos – Lista dos Processos Modelados.....	45
Figura 5.2 – Biblioteca de Beans.....	46
Figura 5.3 – Biblioteca de Projetos.....	47
Figura 5.4 – Framework Bean Ciclo de Vida em Cascata com Blackbox Bean na Atividade de Teste.....	49
Figura 5.5 – Criação do Projeto Verão.....	50
Figura 5.6 – Etapa do Ciclo de Vida em Cascata.....	51
Figura 5.7 – Atividade Aberta de Teste Preenchida com Blackbox Bean.....	52
Figura 5.8 – Execução do Projeto Verão na Base de Aplicação.....	52
Figura 5.9 – Arquitetura Beans Composer.....	54
Figura 5.10 – Diagrama de Classes da Beans Composer.....	56
Anexo XII - Figura 1 – Classificação de Bean.....	90
Anexo XII - Figura 2 – Bean.....	92
Anexo XII - Figura 3 – Projeto.....	93

Anexo XII - Figura 4 – Criando Etapa.....	94
Anexo XII - Figura 5 – Preencher Atividade Aberta.....	95

1. Introdução

Está acontecendo uma revolução na forma de negociar e competir nos mercados de hoje. As empresas têm modificado as suas formas de atuar nos mercados, utilizando-se de estratégias de marketing mais agressivas, reduzindo o ciclo de vida dos seus produtos e aumentando a sua velocidade de produção e se esforçando para atender da melhor maneira possível os seus clientes (SILVA, 2001). No final dos anos 90, destaca-se nas organizações a preocupação em fidelizar os seus clientes, que se tornam cada vez mais exigentes, oferecendo mais opções de produtos e serviços que atendam a suas necessidades (BARBIERI, 2001).

Nesta luta para vencerem os desafios e manterem-se competitivas, muitas organizações descobriram que a forma como conduziam os seus processos era, habitualmente, cara, lenta e ineficiente (SANTOS, 2001). Por isso, as organizações estão cada vez mais interessadas em melhorar os seus processos de negócio com a finalidade de alcançar reduções de custo e tempo. Porém, primeiramente é importante formalizar e conhecer melhor o negócio que está sendo executado, investindo na definição dos processos de negócio.

Neste cenário, a modelagem de processos se apresenta como uma opção para apoiar as organizações na sua busca por auto-conhecimento. A modelagem de processos é um conjunto de métodos e técnicas que auxiliam a organização na formalização do seu negócio. Ela auxilia nessa tarefa pois ajuda a organização a responder às questões críticas sobre o seu negócio, como: o que está sendo feito, por que está sendo feito, onde, por quem, quando e de que forma é feito. Essa formalização também ajuda a organização a encontrar anomalias e inconsistências criando até mesmo oportunidades de melhoria nos processos.

Uma vez realizado o trabalho de modelagem de processos, a organização já possui os seus processos de negócio definidos em termos de atividades e regras de execução. Esta definição pode ser automatizada com a aplicação da tecnologia da informação. Automatizar o fluxo de informações significa a utilização de sistemas para apoiar a transferência das mesmas entre as atividades que compõem um determinado processo (SILVA, 2001). Para isso, algumas organizações têm investido na automação de seus processos utilizando como estratégia a adoção de uma ferramenta de *workflow*.

Neste cenário, uma das dificuldades encontradas é a adaptação do processo definido na organização para as diferentes necessidades que surgem em seu cotidiano. Torna-se necessário definir uma única forma de execução que contemple as variações aceitas pela organização, de modo que atenda ao nível de habilidade e detalhamento que cada equipe tem e deseja. A definição de um procedimento padrão deve ser adaptável e configurável, o que não é uma tarefa trivial.

Tendo esta motivação em vista, surgiu a idéia da utilização de componentes de processo denominados *process beans* (SOBRINHO, 2000) que tem como objetivo agilizar a tarefa de modelagem de processos, de forma que os mesmos possam ser definidos a partir de processos pré-existentes. Este trabalho parte do principio de que é possível definir componentes de processos implementados em sistemas de *workflow* que auxiliarão a definição e uso automatizado de processos em organizações. Para atender este objetivo foi desenvolvida uma ferramenta para a manipulação destes componentes denominada *Beans Composer*.

No próximo capítulo, apresentaremos os conceitos de modelagem de processos. No capítulo 3 falaremos sobre *workflow* e a ferramenta *Lotus Workflow* a partir da qual foi desenvolvida a ferramenta *Beans Composer*. No capítulo 4 definiremos *process beans*, detalhando os conceitos através de um exemplo. No capítulo 5 detalharemos a ferramenta *Beans Composer* através de suas funcionalidades e implementação. No capítulo 6 mostraremos um exemplo da utilização do *Beans Composer* e, por último, no capítulo 7, apresentaremos as conclusões deste trabalho.

2. Modelagem de Processos em Organizações

Este capítulo apresenta os principais fatores motivadores bem como os conceitos e objetivos da modelagem de processos. A seguir, é apresentado um exemplo relacionado a processos de *software*. Ao final, são apresentadas algumas estratégias de automação para os processos modelados.

2.1. Motivação

As organizações estão cada vez mais interessadas em conhecer mais profunda e detalhadamente o negócio que executam com a finalidade de alcançar reduções de custo e tempo. Para tanto, é importante formalizar o seu negócio e o conhecimento sobre ele. A modelagem de processos é um conjunto de métodos e técnicas que auxiliam a organização na formalização e posterior melhoria do negócio (SILVA, 2001). Diversas razões podem ser indicadas como motivadoras para a modelagem de processos. DAVENPORT (1994, *apud* SILVA 2001) apresenta quatro principais razões que a justificam:

- O entendimento dos processos existentes facilita a comunicação entre os profissionais da empresa, além de contribuir para a capacitação daqueles que participam das atividades de modelagem. Os profissionais envolvidos com esta atividade tornam-se habilitados para a participação na implementação das melhorias, uma vez que se tornam os maiores conhecedores das características dos processos que serão melhorados.
- A compreensão dos processos existentes é um pré-requisito para o planejamento e execução das atividades de melhoria do processo, pois é de suma importância conhecer a organização através dos seus processos de negócio.
- Durante a modelagem dos processos os problemas existentes são identificados naturalmente. Conhecer os problemas do processo atual é importante para que se possa evitar a repetição desses problemas na elaboração dos novos processos.
- O conhecimento dos processos existentes permite que se faça uma comparação com os processos otimizados, de forma que se possa avaliar a efetividade da melhoria implementada. Esta avaliação é imprescindível para facilitar o aprendizado com os problemas identificados.

Dessa forma, a modelagem de processos acabará oferecendo o melhor entendimento, uma representação uniforme da empresa e um modelo que pode ser usado para controlar e monitorar as operações da empresa.

2.2. Conceitos

De maneira geral, um *processo* é um caminho para uma empresa organizar o trabalho e os recursos (pessoas, equipamento, informação e esforço) para atingir seus objetivos. Um processo é composto por um conjunto de atividades que tem por finalidade transformar, manipular ou processar insumos para produzir bens ou serviços que irão satisfazer a demanda gerada pelos consumidores ou clientes. Processos utilizam métodos que provêm as regras técnicas de “ como fazer” determinadas tarefas. A execução do processo envolve não só os membros de uma organização como também pode envolver seus fornecedores e clientes numa cadeia de atividades onde informações são geradas e manipuladas (MACHADO, 2000, FALBO, 1998, PFLEEGER, S.L., 1998, VASCONCELOS JR, 1997 *apud* ARAUJO, 2000).

Um *processo de negócio* é uma coleção de atividades realizadas para o objetivo do negócio. Essas atividades são partes bem definidas do trabalho, realizadas num certo momento por participantes do processo de acordo com um conjunto de regras definidas. Processos de negócio podem ser representados através de modelos que especificam: as atividades que compõem o processo, a ordem e as condições que as atividades devem ser executadas, os executores de cada atividade, as ferramentas a serem utilizadas e os documentos manipulados durante sua execução. O modelo de negócio reflete os processos de negócio executados pela organização. Ele mostra o relacionamento entre organizações e uma visão detalhada de como as atividades acontecem em um processo em resposta a um evento do negócio.

A *modelagem de negócio* é um conjunto de métodos e técnicas que auxiliam a organização na formalização dos seus processos de negócio. As abordagens para modelagem de negócio sugerem um conjunto de passos que, quando executados, auxiliam a organização na construção do seu modelo de negócio que fornecerá, através de uma representação, um conjunto de informações que respondem às perguntas abaixo conforme ilustrado na figura 2.1 (PROFORMA, 2002):

- A estrutura organizacional (*quem?*) - documenta de uma forma hierárquica a organização incluindo os papéis desempenhados pelos trabalhadores e os departamentos.
- As localizações geográficas de seu interesse (*onde?*) – geralmente as várias localizações geográficas de interesse da organização são organizadas de forma hierárquica.
- O momento de disparo dos seus serviços (*quando?*) - mostra como os eventos do negócio afetam o estado dos objetos de negócio.
- Os seus processos de negócio (*como?*) - mostra a decomposição do domínio do negócio em processos de negócio e atividades.
- Os recursos manipulados pelos seus processos (*o quê?*) - pessoas, lugares, coisas ou conceitos que retêm informações e provêem funcionalidades para um processo de negócio. Todo processo de negócio usa, produz ou adiciona valor a um conjunto de objetos de negócio.
- Os seus objetivos e metas (*porquê?*) – apresenta hierarquicamente os objetivos a serem alcançados na organização durante a execução dos seus processos.

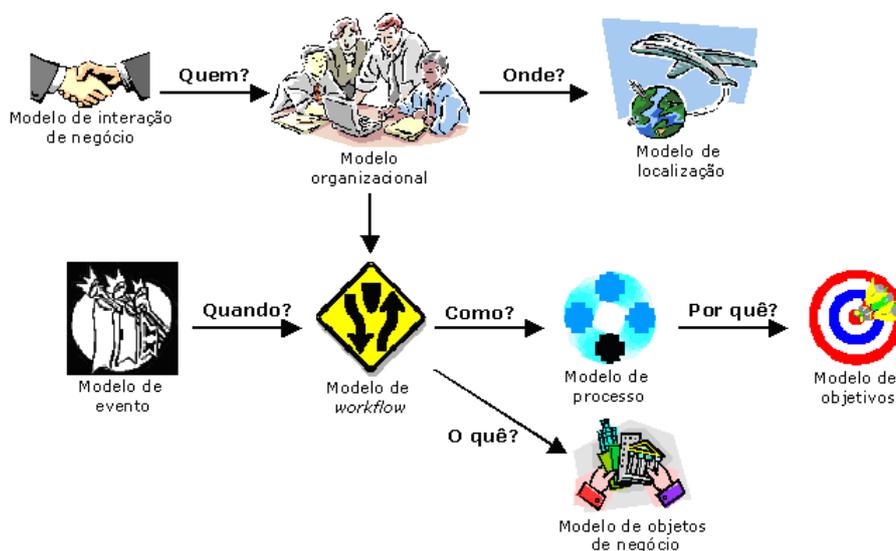


Figura 2.1 – Elementos de Modelo de Negócio da Organização (PROFORMA, 2002)

2.3. Objetivos

A modelagem de processos é realizada com o objetivo de explicitar os processos de uma organização, promovendo o entendimento uniforme. Ferramentas de suporte podem auxiliar na geração de uma descrição que represente inequivocamente as etapas do processo. A elaboração de tais modelos permite que a organização visualize, analise e otimize os seus negócios além de permitir o estudo e coleta de dados para ajudar no planejamento e melhoria do mesmo (SANTOS, 2001).

Quando definido, um processo pode ser considerado também uma estrutura para aprendizado, reuso e estabelecimento de uma cultura de práticas e metodologias padronizadas de trabalho na organização. Depois de construir o modelo de negócio, a organização pode mais facilmente identificar os seus problemas e as soluções tecnológicas mais apropriadas para apoiar o seu negócio.

As principais finalidades da Modelagem de Processos são:

- Uniformização do entendimento da forma de trabalho gerando integração;
- Análise e melhoria do fluxo de informações;
- Explicitação do conhecimento sobre os processos, armazenando o *know how* da organização;
- Realização de análises organizacionais e de indicadores;
- Realização de simulações apoiando a tomada de decisões.

Uma vez realizada a modelagem de processos, as informações disponibilizadas podem ser utilizadas de diversas formas (IENDRIKE, 2003):

- Realizar a re-engenharia do negócio - o próprio exercício de criação do modelo de negócio ajudará a organização, revelando anomalias, inconsistências, ineficiências e oportunidades de melhoria;
- Documentar os processos para uma certificação para se atingir um padrão esperado de qualidade;
- Contratar ou desenvolver sistemas de informação;
- Automatizar os processos de negócio - o modelo de negócio já disponibiliza a maioria das informações necessárias para a automação dos processos através das soluções tecnológicas mais apropriadas.

2.4. Processo de *Software*

A modelagem de processos pode ser utilizada por organizações com diferentes domínios de negócio, visando aumentar o conhecimento e o controle sobre todas as etapas do seu negócio. Neste contexto, uma das áreas aonde a modelagem de processos vem sendo aplicada é a engenharia de *software*, onde a qualidade e a produtividade se tornaram fatores críticos para a competitividade das organizações. Ao mesmo tempo, a complexidade dos produtos a serem desenvolvidos também cresceu. Assim, a qualidade do produto se tornou fortemente dependente da qualidade do processo de desenvolvimento utilizado.

Segundo PRESSMAN (1997), *processo de software* é o elemento que une os vários níveis tecnológicos da engenharia de *software* e permite o desenvolvimento de forma racional e programada visando à melhoria de sua qualidade (ARAUJO, 2000). A definição do processo de *software* inclui a escolha dos métodos e ferramentas para a sua execução. Métodos provêm as regras técnicas de “ como fazer” determinadas tarefas do processo. Ferramentas oferecem apoio automatizado tanto para os métodos como para o processo. Estes elementos estão ilustrados na Figura 2.2.



Figura 2.2 – Atividades do ciclo de vida de um Processo de *Software* Típico (PRESSMAN, 1997)

A definição de um processo de *software* compreende, de uma maneira geral: a definição de um ciclo de vida para o processo, o detalhamento de cada fase do ciclo de vida em atividades, a definição de como as atividades devem ser realizadas, a definição dos artefatos requeridos e produzidos por cada atividade e a definição dos recursos (pessoas e ferramentas) para a realização de cada atividade definida (DERNIAME, KABA e WASTELL, 1999 *apud*: PENADÉS, 1999, FALBO, 1999 *apud* ARAUJO, 2000).

Por exemplo, no *Rational Unified Process (RUP)*, *framework* de processo de desenvolvimento de *software* da *Rational*, o ciclo de vida adotado é o iterativo e incremental. Um dos modelos de processo que compõem o RUP é ilustrado pela figura 2.3. Neste exemplo, o processo escolhido foi o de requisitos, que tem como finalidades: estabelecer e manter concordância com os clientes e outros envolvidos sobre o que o sistema deve fazer; oferecer aos desenvolvedores do sistema uma compreensão melhor dos requisitos; fornecer uma base para estimar o custo e o tempo de desenvolvimento; definir uma interface de usuário para o sistema, focando nas necessidades e metas dos usuários. Ainda na figura 2.3 é ilustrada a seqüência de execução das atividades que compõem o processo de requisitos.

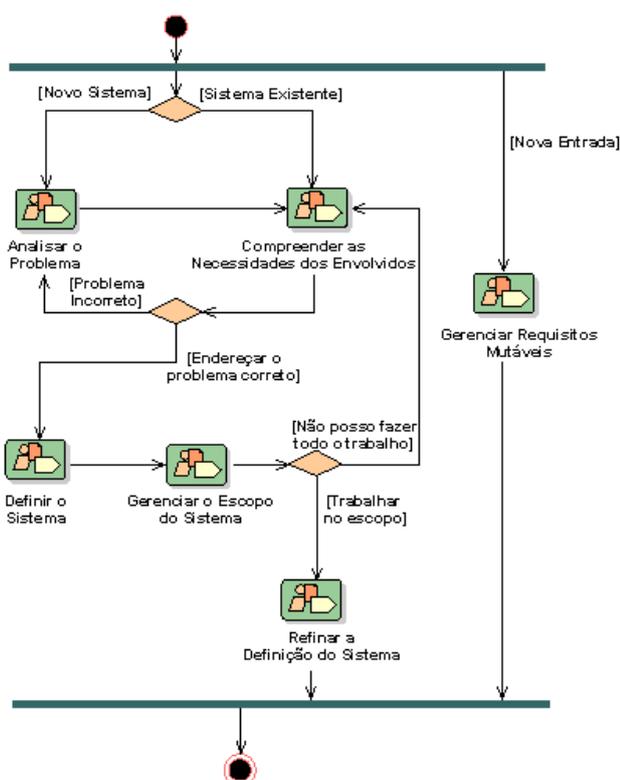


Figura 2.3 – Processo de Requisitos (RUP, 2002)

Como exemplo dos artefatos e recursos necessários para a definição de uma atividade, a figura 2.4. detalha a atividade de Analisar o Problema do processo de Requisitos. Esta atividade está expandida em sub-tarefas. Os artefatos (documentos, modelos e etc...) que servem de insumo ou que são produzidos por cada sub-tarefa, estão representados graficamente. A direção das setas indica em qual sub-tarefa o artefato é produzido e onde ele é consumido. Os recursos (pessoas que desempenham papéis) também estão ilustrados pelo Analista de Sistemas, Cliente e Usuário.

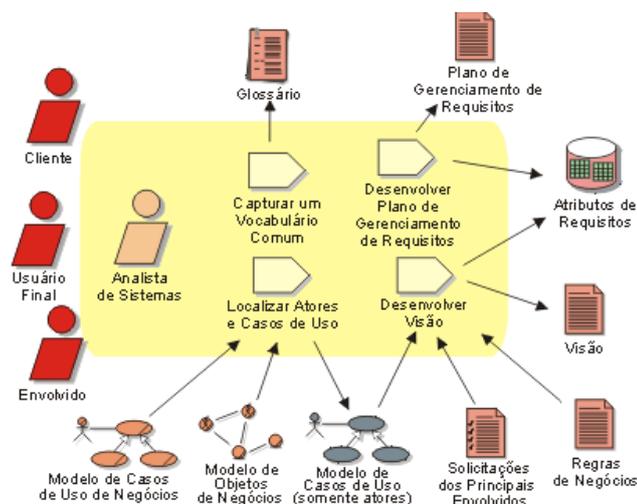


Figura 2.4 – Atividade Analisar o Problema (RUP, 2002)

Uma vez definido o processo de *software*, a estratégia que era utilizada pelas organizações para a sua implantação, era a definição de um procedimento padrão com uma definição operacional das práticas e atividades essenciais que devem guiar todos os projetos de desenvolvimento conduzidos dentro de uma organização (KALTIO e KINNULA, 2000, MACHADO, 2000, MAIDANTCHIK, ROCHA e XEXEO, 1999, FIORINI, ALO e BASTOS, 1999, EL EMAM, DROUIN e MELO, 1998, PAULK, 1993 *apud* ARAUJO, 2000). A idéia do processo padrão parte do pressuposto de que há um conjunto de elementos ou práticas fundamentais que se deseja incorporar em todos os projetos de desenvolvimento executados em uma organização.

Com o tempo foi constatado que um único modelo não satisfazia as variações do desenvolvimento, pois para ser eficaz e conduzir à construção de produtos de boa qualidade, um processo deve ser adequado ao domínio da aplicação e ao projeto específico. Desse modo, processos devem ser adaptados caso a caso considerando-se as especificidades da aplicação, a tecnologia a ser adotada na sua construção, o grupo e a metodologia de desenvolvimento. Sendo assim, a adaptação do processo padrão para cada projeto passa a ser uma tarefa crucial que, infelizmente, não é simples. Além disso, oferecer apoio automatizado a sua realização é um importante desafio (BORGES e FALBO 2001).

2.5. Automação de Processos

Uma vez realizado o trabalho de modelagem de processos, a organização já possui os seus processos de negócios definidos em termos de atividades e regras de execução e representados através de um fluxo de trabalho. Esta definição pode ser automatizada com a aplicação da tecnologia da informação. Automatizar o fluxo de trabalho significa a utilização de sistemas para apoiar a transferência das informações entre as atividades que compõem um determinado processo (SILVA, 2001). O relacionamento entre estes elementos está ilustrado na figura 2.5.

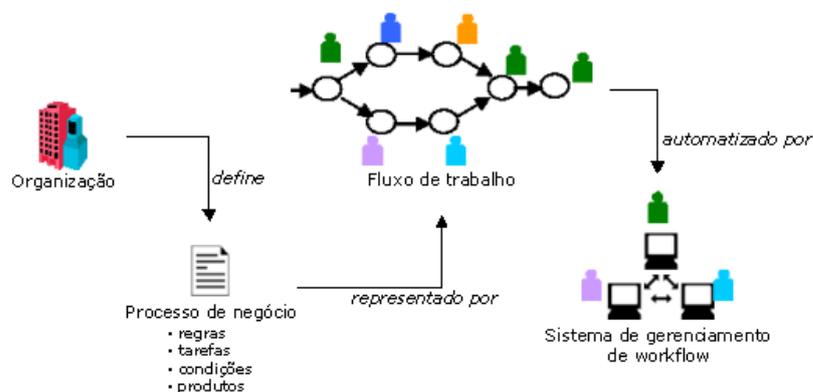


Figura 2.5 - Processos de Negócio Automatizados por WfMS (ARAUJO e BORGES, 2001)

Para a automação dos processos de negócio podem ser adotadas três estratégias: o desenvolvimento, aquisição ou implantação de novos sistemas de *software*; a adoção de um sistema de gerenciamento de *workflow* para a implementação dos processos automatizados e o acompanhamento e gerenciamento da sua execução; uma combinação das duas estratégias anteriores. Os sistemas de informação servirão para apoiar determinadas partes de negócio automatizando atividades cujo fluxo de execução está sendo acompanhado pelo sistema de gerenciamento de *workflow*.

Automatizar um processo através de um sistema de *workflow* permite a integração entre todas as atividades que compõem um processo, facilitando e controlando o fluxo de informações entre elas e contribuindo para um aprimoramento global de todo o processo (SILVA, 2001).

Entretanto, sistemas de *workflow* não substituem as tradicionais ferramentas de automação de escritório, como processadores de texto e planilhas eletrônicas, nem tão pouco os sistemas corporativos da empresa. Ao invés disso permitem uma utilização integrada dessas diversas ferramentas para a execução otimizada de todo o processo, conforme visto na figura 2.6.

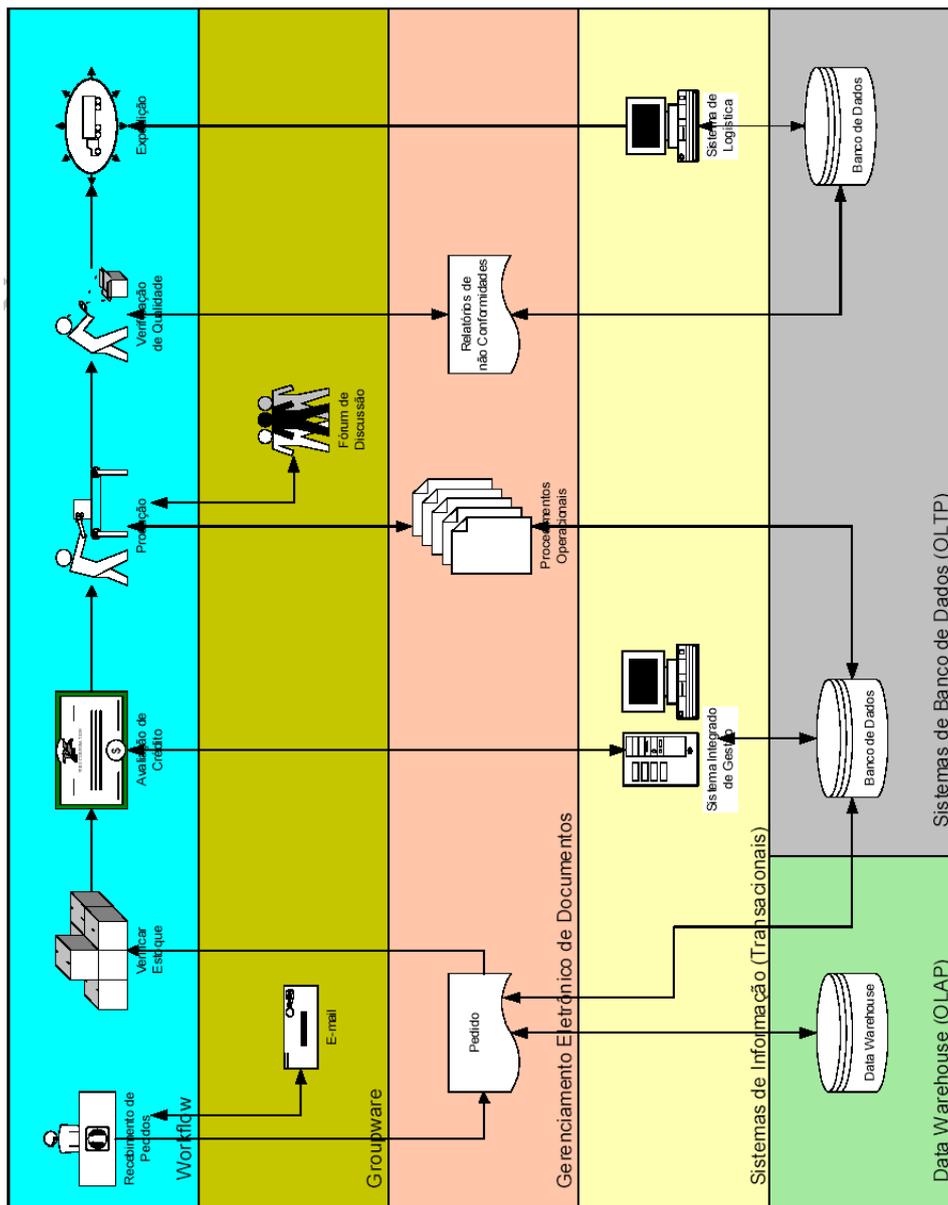


Figura 2.6 – Exemplo da Integração Promovida por Sistemas de *Workflow* (SILVA, 2001)

3. Workflow

Este capítulo caracteriza e define os sistemas de *workflow*, apresentando: sua evolução, os principais objetivos do uso desta tecnologia, alguns conceitos e funcionalidades e os principais benefícios e desafios que se apresentam com a sua utilização. Apresenta também a ferramenta *Lotus Workflow*, sistema de gerenciamento de *workflow*, a partir do qual a solução deste trabalho foi desenvolvida.

3.1. Histórico

Até o final da década de 60, a visão do funcionamento de uma organização era a de supervisores nos escritórios que distribuíam as tarefas para os diversos profissionais, baseados nos seus conhecimentos ou experiência. Os *office-boys* passavam de mesa em mesa entregando o trabalho ou recolhendo o que tinha sido feito. O gerenciamento do trabalho era de responsabilidade do supervisor que monitorava o andamento da tarefa designada. Existiam pessoas responsáveis por localizar e corrigir os erros de roteamento (PLESUMS, 2002). Em geral, os controles e registros eram feitos em papel o que acarretava uma série de desvantagens, entre as quais podemos citar: dificuldade de garantir consistência e regras, dificuldade de armazenamento, pesquisa extremamente lenta, utilização “ mono-usuário” ou necessidade de reprodução e a perda ou extravios freqüentes.

Neste contexto, iniciam-se pesquisas de automação de escritórios focadas em oferecer soluções para geração, armazenamento, manipulação, roteamento e gerência de documentos nas organizações, visando diminuir a manipulação física de documentos em papel, conforme demonstra a figura 3.1. Como resultado destas pesquisas, começam a surgir na década de 70, as primeiras ferramentas com características de *workflow*. Estas ferramentas começaram a entrar nas organizações para desempenhar o papel dos atuais sistemas de GED (Gerência Eletrônica de Documentos). Os documentos eram digitalizados e um sistema de *workflow* era utilizado para fazer seus roteamentos entre as pessoas que precisavam recebê-los (KHOSHAFIAN, 1995).

Ao longo dos anos, muitos outros tipos de produtos oferecidos no mercado possuíam características de *workflow*. Entretanto, o controle rígido no roteamento de informações, a deficiência no tratamento de exceções de execução ao processo, e ainda a ausência de recursos tecnológicos para o suporte à distribuição prejudicaram seu uso e fortaleceram as pesquisas na busca de soluções mais flexíveis e apropriadas.

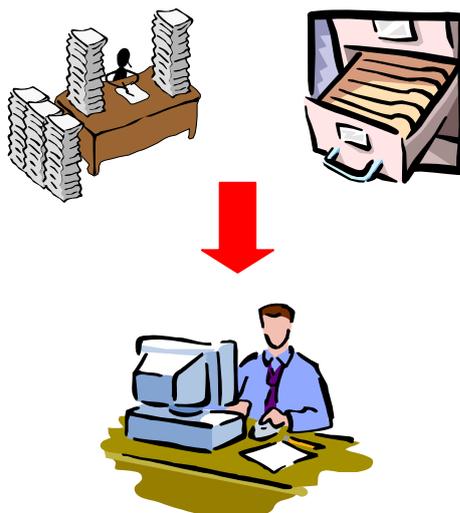


Figura 3.1 – Evolução da Gerência de Documentos em Organizações

A partir da década de 80, na área de tecnologia da informação, dois grandes marcos podem ser considerados fundamentais para o advento dos sistemas de *workflow*: o progresso das tecnologias de *software* e *hardware*, e a necessidade de sistemas que integrassem programas, processos, dados, documentos e organizações. Com o rápido progresso do *hardware*, presenciou-se uma revolução nos escritórios: os computadores pessoais, disponíveis a custos reduzidos, ganhavam lugar em cada área de trabalho. Além disso, começava a ser possível interconectar estas máquinas estabelecendo as redes de computadores. Como consequência da evolução do *hardware*, são desenvolvidas melhorias nos *softwares* de bancos de dados, editores de texto, planilhas eletrônicas e aplicativos para manipulação de imagens.

Ainda neste período, o desenvolvimento da área de pesquisa em CSCW (*Computer Software Cooperative Work*) e *groupware*, também impulsionam o desenvolvimento dos sistemas de *workflow*. O *workflow* começa a ser utilizado como ferramenta para coordenação do trabalho em equipe, ficando responsável pelo roteamento do trabalho entre as pessoas. Seu principal objetivo

era unir as chamadas “ ilhas de trabalho e informação” individuais, formadas por cada membro da organização, através do roteamento de trabalho (ARAUJO, 1999). Um exemplo de roteamento do trabalho é apresentado na figura 3.2.

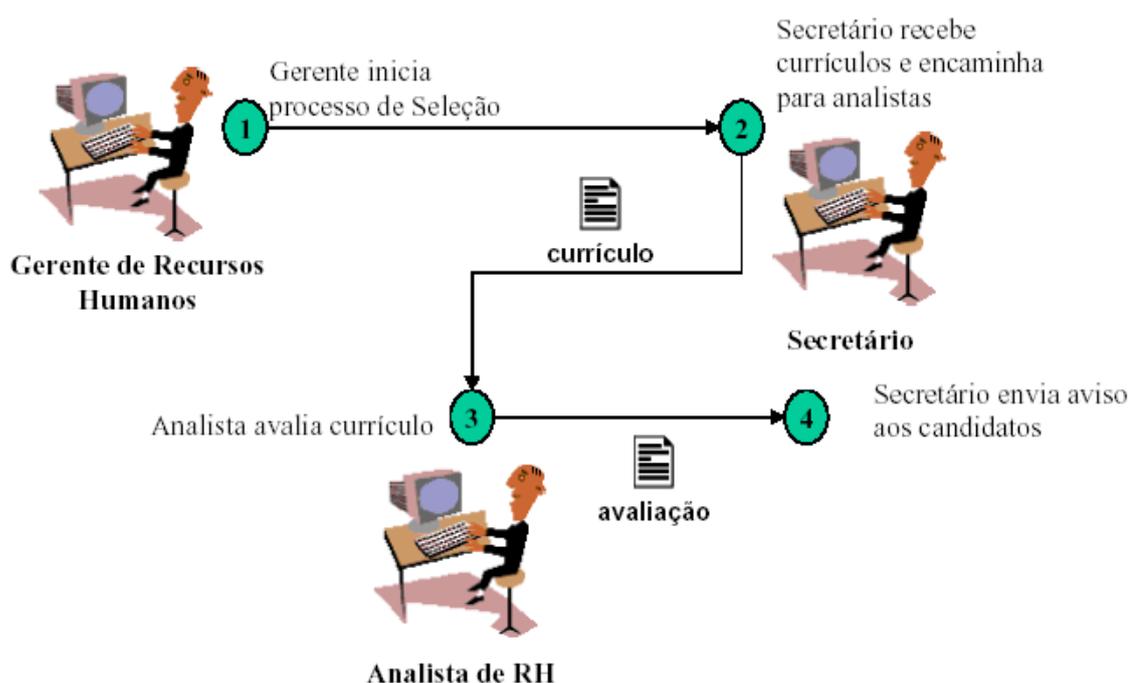


Figura 3.2 – Roteamento de Informações e Tarefas entre Membros de uma Organização (ARAUJO e BORGES, 2001)

As pesquisas voltam-se para a especificação de arquiteturas e a construção de sistemas de *workflow* que ampliem a utilização desta tecnologia. O *workflow* vai ganhando mercado como ferramenta coadjuvante com o crescimento das iniciativas de reengenharia e melhoria de processos, por oferecer soluções para atender à demanda das organizações por maiores índices de qualidade e eficiência no gerenciamento de seus processos de negócio.

Com a chegada dos anos 90, o mundo assistiu a evolução das infra-estruturas de redes, ao avanço da Internet, da intranet e da globalização. A tecnologia de *workflow* também acompanhou esta nova realidade, através de pesquisas de arquiteturas distribuídas de execução de processos.

3.2. Definição

O *Workflow Management Coalition* (WfMC), entidade sem fins lucrativos criada em agosto de 1993 por cerca de 90 empresas, entre elas: Coca-Cola, HP, IBM, Microsoft, Novell, Oracle, SAP e Xerox, tem como principal objetivo o desenvolvimento de padrões e terminologia para a tecnologia de *workflow*. Segundo esta entidade, a definição de *workflow* é “ a automação de um processo de negócios, no todo ou em parte, durante o qual documentos, informações ou tarefas são transferidos entre participantes do processo, de acordo com um conjunto definido de regras, para alcançar um determinado objetivo de negócio” (WfMC, 2003).

Ainda de acordo com o WfMC, um Sistema de Gerenciamento de *Workflow* (WFMS) é definido da seguinte forma: “ sistema para definição, criação e gerência da execução de fluxos de trabalho através do uso de *software*, capaz de interpretar a definição de processos, interagir com seus participantes e, quando necessário, invocar ferramentas e aplicações” . De acordo com a *Lotus Development Corporation*, o principal objetivo dos *softwares* de *workflow* é aumentar a eficiência de processos de negócio, e a efetividade das pessoas que trabalham em conjunto para executá-los.

Outra definição, dentre muitas encontradas na literatura, propõe que *workflow* é a tecnologia que possibilita automatizar processos, racionalizando-os e potencializando-os por meio de dois componentes implícitos: organização e tecnologia (CRUZ, 1998). A tecnologia e a organização se complementam, pois a tecnologia por si só não resolve problemas se não forem considerados aspectos da cultura organizacional. *Workflow* é uma tecnologia que transforma radicalmente a maneira como são executados processos, atividades, tarefas, políticas e procedimentos numa organização porque automatiza procedimentos e controla a informação em um processo de negócio.

3.3. Conceitos

Um sistema de gerenciamento de *workflow* objetiva a automação e a gerência de processos. Dentro destes sistemas, um processo é considerado um conjunto de atividades com *regras de execução* bem definidas. As *atividades* são realizadas por papéis. Aos *papéis* são associados *executores* que podem ser indivíduos, grupos ou agentes automatizados. Cada atividade possui um *pacote de trabalho* composto de dados, documentos ou formulários, que são passados entre os participantes de acordo com um conjunto de regras definidas pela organização. A relação entre estes elementos é ilustrada pela Figura 3.3.

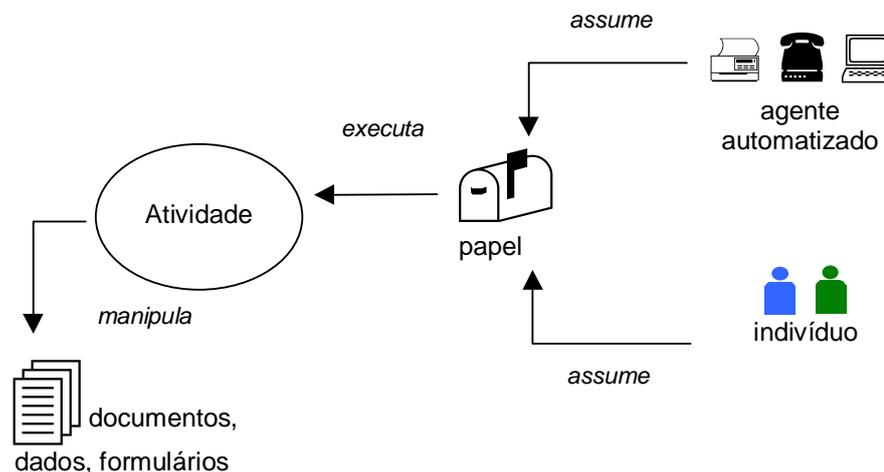


Figura 3.3 - Elementos em um Sistema de Gerenciamento de *Workflow* (ARAUJO e BORGES, 2001).

➤ *Atividade*

Uma atividade consiste em uma unidade individual de trabalho. Uma atividade é uma tarefa a ser desempenhada em um processo e para isso ela manipula documentos, dados e formulários. As atividades estão relacionadas através de regras de negócio que estabelecem a seqüência em que as atividades devem ser executadas, a sua obrigatoriedade ou não, e a dependência de outras atividades. A cada atividade, durante a sua definição, são associados os papéis responsáveis por sua realização.

➤ *Papel*

Um papel é responsável por executar uma ou mais atividades de um processo. O papel pode ser desempenhado por agentes automatizados, grupos ou indivíduos da organização que tenham acesso ao sistema de *workflow*. O mesmo indivíduo pode estar associado a mais de um papel e cada papel pode ser desempenhado por mais de um profissional.

As atividades automatizadas podem ser executadas tanto por agentes computacionais, por exemplo, um envio automático de e-mail, quanto por outras aplicações ou dispositivos eletrônicos como impressoras e faxes.

➤ *Participante*

Usuário cadastrado e com acesso a um sistema de *workflow*, ou seja, o indivíduo que interage com o sistema de *workflow*. O participante desempenha um ou mais papéis. Um participante pode fazer parte de um ou mais grupos.

➤ *Pacote de Trabalho*

Os sistemas de *workflow* costumam permitir a definição de pacotes de trabalho para cada atividade, onde poderão ser armazenados os documentos, dados e formulários que forem manipulados ao longo da execução do processo.

➤ *Lista de Trabalho*

Cada participante de um sistema de *workflow* possui a sua própria lista de trabalho. Cada lista contém as atividades a serem realizadas por este usuário, podendo incluir atividades de instâncias de processos distintos. Nestas listas, os participantes podem selecionar as atividades para serem executadas. O encaminhamento de novas atividades aos seus respectivos executores provoca a inclusão de itens nas listas de trabalho.

➤ *Rota*

A definição de um processo também inclui a explicitação do tipo de encadeamento das atividades: seqüencial, em paralelo ou sob determinadas condições ou qualquer.

No roteamento seqüencial, logo que uma atividade é concluída a sua sucessora é roteada para o executor responsável. Atualmente, devido à complexidade que os processos de negócio

das organizações atingiram, são raros os processos puramente seqüenciais. A regra geral é a existência de múltiplas rotas, dentro de um único processo, percorridas dependendo de cada ocorrência ou documento processado. (CRUZ, 2001).

O roteamento em paralelo permite que mais de uma atividade seja colocada em execução simultaneamente. Isso possibilita o tratamento das múltiplas rotas de um processo que ao final voltarão a se juntar em um único fluxo de trabalho.

Em um processo é possível ainda definir os caminhos condicionais que implicam na tomada de alguma decisão baseada nas informações de execução do processo disponíveis (ARAUJO e BORGES, 2001). A figura 3.4 caracteriza os três possíveis tipos de rotas.

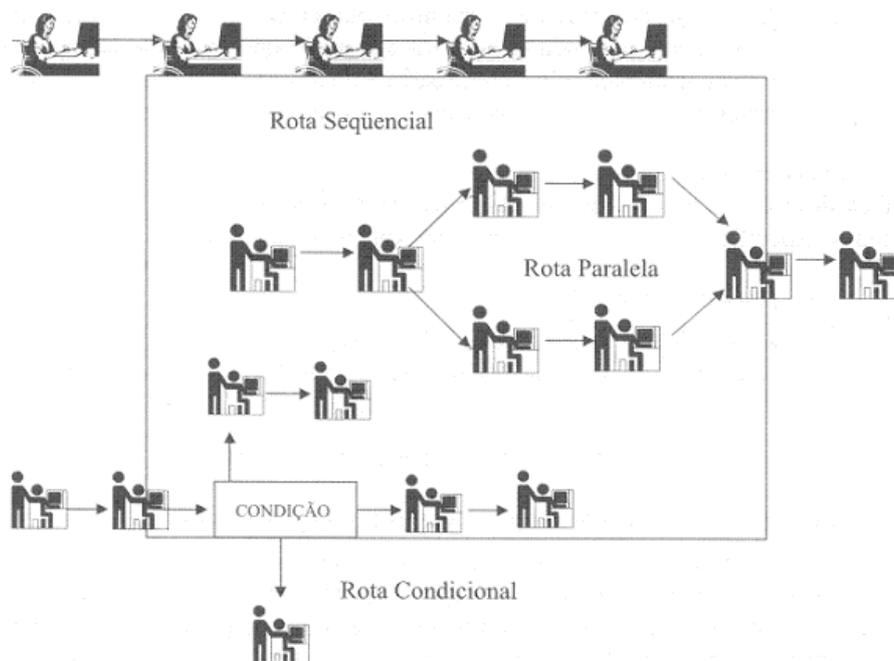


Figura 3.4 – Tipos de Rotas (CRUZ, 2000).

3.4. Funcionalidades

Nesta seção serão descritas as principais funcionalidades oferecidas por sistemas de *workflow* que, basicamente, se dividem em quatro áreas funcionais: definição de processos, execução dos processos definidos, acompanhamento das interações do usuário durante sua execução e gerência das instâncias em execução.

3.4.1. Definição dos Processos

A definição de um processo é a etapa de implementação na ferramenta de *workflow*. Conforme mencionado anteriormente, a modelagem de processos auxilia a organização na formalização do processo representado através de um modelo computacional e processável. Um modelo de processo deve conter todos os dados necessários sobre o processo que precisará ser executado pelo sistema de *workflow* tais como: atividades automáticas ou paralelas, condição de inicialização e finalização, regras de negócio, pontos de decisão, rotas seqüenciais ou paralelas, mapa do processo, executores, documentos manipulados em cada atividade e aplicações necessárias (PRIOR, 2003). Um exemplo de modelo de processo está ilustrado na figura 3.5.

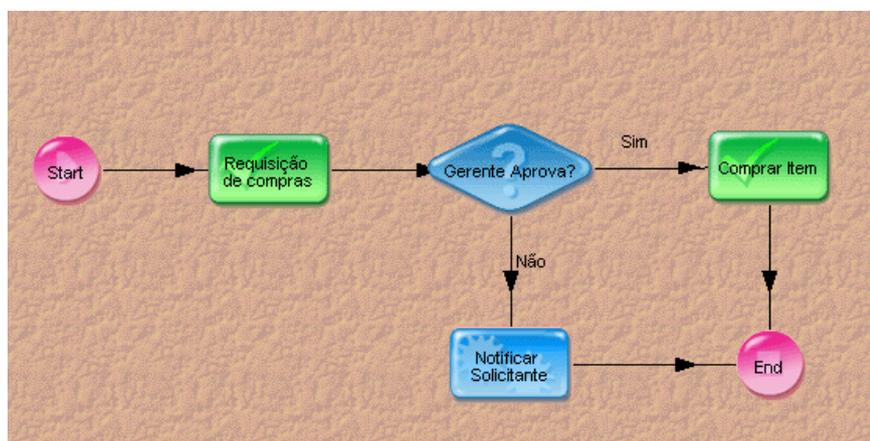


Figura 3.5 – Exemplo de Definição de Processo – Processo de Solicitação de Compras

3.4.2. Execução dos Processos

Depois de implementado, um processo pode ser executado através da sua interpretação pelo sistema de *workflow* que acompanha e coordena a execução das instâncias do processo. O sistema de gerenciamento de *workflow* é que se encarrega do roteamento das atividades aos seus executores no momento apropriado, convocando os recursos computacionais para a sua realização, quando necessário. Desta forma, as atividades vão sendo incluídas nas listas de trabalho dos executores. Cada participante do processo necessita interagir com o sistema para realizar as suas atividades e participar do processo.

A execução de um processo corresponde à ativação de instâncias deste processo. Várias instâncias de um mesmo processo ou de processos distintos podem estar em execução simultaneamente em uma mesma máquina de *workflow*. Estas possibilidades são exemplificadas pela figura 3.6.

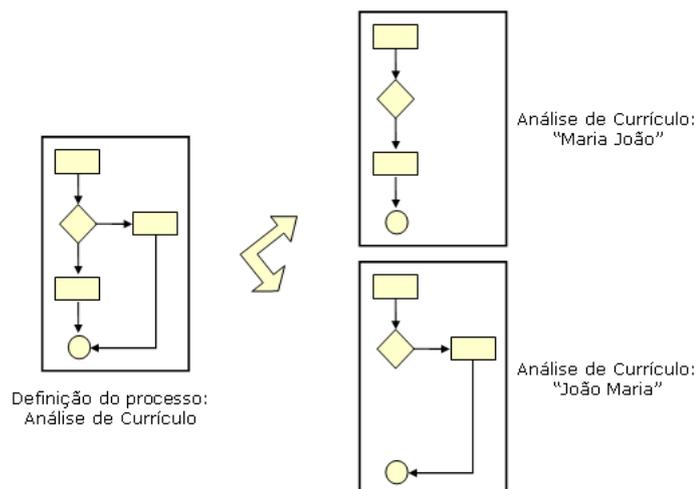


Figura 3.6 – Definição de processo e suas instâncias em execução (ARAUJO e BORGES, 2001).

A máquina de execução do *workflow* pode se encontrar distribuída por diversas plataformas computacionais. Além disso, várias máquinas de *workflow* podem cooperar para a encenação de um ou mais processos.

3.4.3. Interação com usuários

Os usuários interagem com o sistema de *workflow* através de suas respectivas listas de trabalho, onde são incluídas as novas atividades sob a sua responsabilidade para realização. Muitas vezes, o usuário não precisa nem ter conhecimento das atividades de mais alto nível de onde partiu a necessidade de realização das suas atividades particulares.

As atividades são executadas nos ambientes de trabalho dos executores através de aplicações ou ferramentas específicas. A realização de uma tarefa envolve a manipulação dos documentos estipulados para análise das informações, tomada de decisões ou preenchimento de dados. A conclusão da atividade dispara novas atividades de acordo com os resultados gerados.

3.4.4. Acompanhamento e Gerência de processos em execução

Os sistemas de *workflow* registram o histórico do processo e permitem que os usuários incluam comentários durante a execução de cada atividade. Este histórico inclui: data, hora, executor de cada atividade e o andamento de cada passo do processo. Por exemplo, podem-se saber quais atividades foram suspensas e as suas respectivas justificativas. Além disso, outras explicações sobre acontecimentos ou circunstâncias especiais também são capturadas. Desta forma estas ferramentas disponibilizam um *log* automático de quem fez o quê, quando e como (PLESUMS, 2002).

Algumas ferramentas de *workflow* oferecem relatórios de análise, tais como: o total de trabalho realizado em termos de volume e de tempo de resposta para o cliente; produtividades individuais ou das equipes; número e tipo de erros encontrados e corrigidos por cada tipo de processo ou usuário. Os sistemas de *workflow* também permitem que os gerentes acompanhem o *backlog* de trabalho ao longo do dia, de forma que eles possam realmente gerenciar as suas equipes para atender os prazos. Alguns permitem ainda que os gerentes configurem alarmes que lhe notifiquem a respeito de determinadas circunstâncias que eles considerem críticas. Obviamente, todas estas facilidades são muito mais adequadas do que uma contagem manual do trabalho pendente ou do acompanhamento do crescimento das pilhas de papéis sob cada mesa.

Alguns sistemas de *workflow* oferecem ainda a possibilidade de visualizar o *status* de execução de uma instância de processo. Esta visualização geralmente é apresentada graficamente, destacando as atividades já realizadas, as atividades em execução e as atividades que ainda devem ser executadas.

3.4.5. Administração

Diferenciação entre usuários executores e administradores de processos que possuem permissões para suspensão, cancelamento e restabelecimento da execução de processos.

3.4.6. Acesso integrado à informação

O acesso à informação necessária para completar a atividade deve ser fácil e rápido. Não adianta automatizar o roteamento de atividades e o participante que receber uma atividade precisar gastar horas para localizar os dados associados. Por isso, os sistemas de *workflow* costumam realizar interfaces com bancos de dados ou outros sistemas existentes; estabelecer *links* com imagens, servidores de *fax*, *e-mails* ou outros conjuntos de dados externos; extrair as informações principais para que sigam junto com o *workflow*. Entretanto, é importante perceber que o papel do sistema de *workflow* não é substituir o repositório de onde estão sendo extraídos os dados (PLESUMS, 2002). Ao invés disso, permitem uma utilização integrada dessas diversas ferramentas para a execução otimizada de todo o processo.

3.4.7. Funcionalidades Adicionais

Os sistemas de gerenciamento de *workflow* disponíveis comercialmente costumam variar bastante em relação à oferta de funcionalidades avançadas. Nesta seção vamos descrever algumas delas (ARAUJO e BORGES, 2001, CRUZ, 2001).

Funcionalidade	Descrição
<i>Definição de Pesos para Grupos</i>	Possibilidade de definir pesos dentro de um grupo de forma que uma atividade possa ser designada a um membro do grupo baseado nos pesos definido, facilitando a gerência da carga de trabalho.
<i>Grupos Seqüenciais</i>	Possibilidade de definição de grupos onde uma atividade é designada ao primeiro membro do grupo. Se este não for capaz de realizá-la no tempo requerido ela é automaticamente roteada para o próximo membro e assim sucessivamente.
<i>Monitoramento de Atividades</i>	Possibilidade de determinar a carga de trabalho de cada participante, visualizando-se quanto e que tipo de atividades estão pendentes para cada participante. Assim podem-se redistribuir algumas ou todas elas para outra pessoa.
<i>Sub-processos</i>	Possibilidade de definir sub-processos em um processo. Isto permite o projeto de implementação de processos aninhados.

<i>Disparos Periódicos</i>	Possibilidade de instanciações de processos de forma automática e periódica.
<i>Rejeição de um Passo de Trabalho</i>	Possibilidade de um participante rejeitar a execução de um passo de trabalho, fazendo com que o fluxo retorne ao passo anterior de execução.
<i>Grupos Dinâmicos</i>	Possibilidade de definir grupos responsáveis por uma atividade no momento de sua execução.
<i>Gerenciamento de Atrasos</i>	Quando uma atividade atrasa pode comprometer o custo, a eficiência e a produtividade do processo como um todo. Por isso, o responsável pelo processo deve dispor de ferramentas que lhe permitam tomar providências para acabar com tais gargalos ou disfunções. Sem o auxílio de uma ferramenta de <i>workflow</i> , os atrasos só são conhecidos depois que ocorreram e, como consequência, as pessoas estão sempre agindo de forma reativa aos atrasos (CRUZ, 2001). Com o suporte de uma ferramenta de <i>workflow</i> , que permita a configuração de notificações sobre ocorrência de eventos ao longo da execução do processo, os atrasos podem ser divulgados antes que ocorram efetivamente, permitindo a organização agir de forma pró-ativa para não deixar que eles degradem o processo.
<i>Auditoria Automática</i>	Possibilidade de manter versões dos documentos ao longo do processo.
<i>Agente de Correio</i>	Possibilidade de executar ações baseadas no recebimento de mensagens.
<i>Priorização Automática de Atividades</i>	Possibilidade de priorizar automaticamente as atividades de acordo com os prazos estimados, ou seja, as tarefas mais importantes são encaminhadas em primeiro lugar. Assim, os participantes não perdem tempo escolhendo qual atividade realizar primeiro.
<i>Processamento em Paralelo</i>	Possibilidade de que duas ou mais atividades sejam executadas concorrentemente.
<i>Seleção do Participante Mais Indicado</i>	Alguns sistemas de <i>workflow</i> mais simples distribuem o trabalho entre todos os recursos disponíveis, ou encaminham o trabalho que está aguardando para o recurso com uma menor fila de trabalho. Entretanto, alguns sistemas já conseguem encaminhar as atividades que vão chegando para o participante com o perfil mais indicado para atendê-la (PLESUMS, 2002).

3.5. Lotus Workflow

3.5.1. Arquitetura

O *Lotus Workflow* é um sistema de gerenciamento de *workflow* desenvolvido pela empresa IBM/Lotus na plataforma *Lotus Notes* (LOTUS, 2003). Ele funciona através de uma arquitetura cliente-servidor. Nas estações clientes funcionam a definição de processos, a interação, o acompanhamento e a administração do sistema. No servidor encontra-se o motor ou a máquina de *workflow*, responsável pela interpretação e roteamento do processo para os diversos executores.

O *Lotus Workflow* é composto de: seis bases de dados (Organização, Repositório de Processos, Definição de Processos, Aplicação, Arquivamento e Auditoria) geradas a partir de *templates* fornecidos com o produto; um módulo de modelagem gráfica de processo denominado *Lotus Workflow Architect*, um módulo que permite a visualização do andamento das instâncias de processos denominado *Lotus Workflow Viewer*. Esta arquitetura é ilustrada na figura 3.7.

Na base de Organização é definida a estrutura organizacional composta pelos participantes, departamentos, grupos de trabalho e papéis. Nesta base também são feitas as configurações dos dias úteis, horário de expediente, feriados e definições de ausência do escritório. Nesta base é possível ainda definir os aplicativos que poderão ser acessados diretamente na execução das tarefas.

As definições de processos criadas no *Lotus Workflow Architect*, têm os seus códigos fontes armazenadas na base de Repositório de Processos. Uma vez que uma determinada versão de uma definição de processo é verificada, compilada e ativada, ela fica disponível também na base de Definição de Processos.

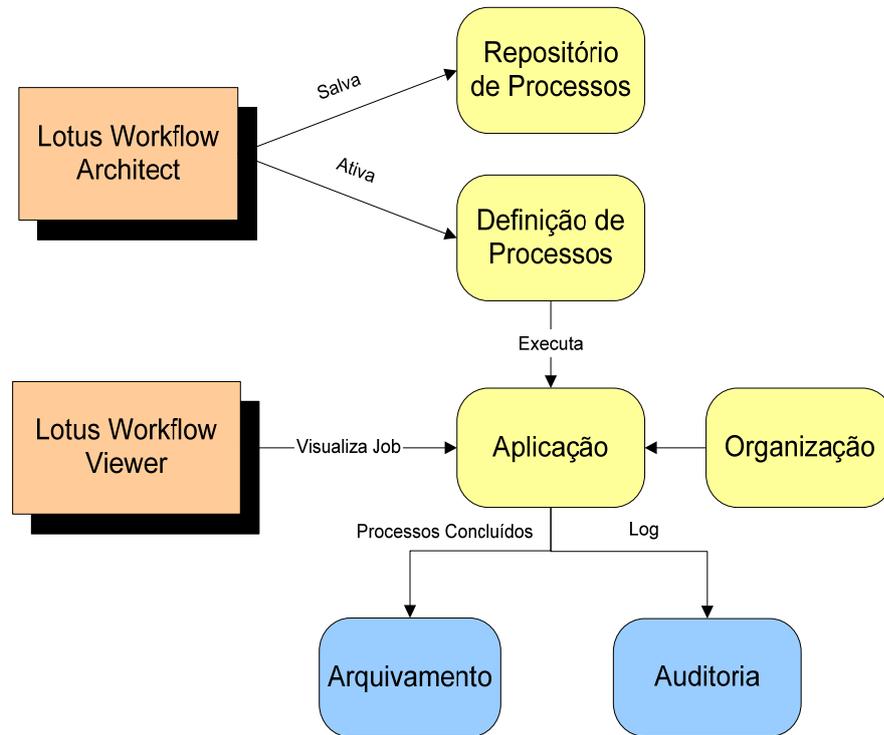


Figura 3.7 – Arquitetura do *Lotus Workflow*

Na base de Aplicação funciona a máquina de *workflow* efetivamente. Esta base possui um documento de configuração onde são definidos: os acessos às demais bases; a disponibilidade do *Lotus Workflow Viewer*; as regras para arquivamento e auditoria, entre outros. Uma vez ajustadas todas as configurações desejadas para a operação, é possível instanciar processos, conhecidos como *jobs*, e acompanhar a sua execução. A base de Aplicação é a única que costuma ser conhecida e utilizada pelos participantes do sistema de *workflow*. As demais bases, geralmente, ficam restritas aos administradores.

Depois de concluídos, os processos podem ser, opcionalmente, armazenados em uma base de Arquivamento, de acordo com as regras definidas pelos Administradores. Isto permite um melhor desempenho da base de Aplicação, sem que o histórico dos processos sejam perdidos.

A última base disponível no *Lotus Workflow* é a base de Auditoria, onde são registrados os *logs* das operações da base de Aplicação, se a organização assim o desejar.

3.5.2. Lotus Workflow Architect

- **Mapa do Processo**

O mapa de um processo é o modelo gráfico do processo. O *Lotus Workflow Architect* possui uma notação própria que é parcialmente apresentada na figura 3.8. No desenho, os retângulos verdes representam atividades manuais que compõem o processo. As setas orientadas indicam a seqüência de execução das atividades. Os dois círculos cor de rosa simbolizam os pontos de início e fim do processo em questão.

O *Lotus Workflow Architect* também possui simbologias específicas para atividades do tipo ponto de decisão (representadas através de losangos azuis) e para atividades executadas por agentes automatizados (representadas por retângulos azuis).

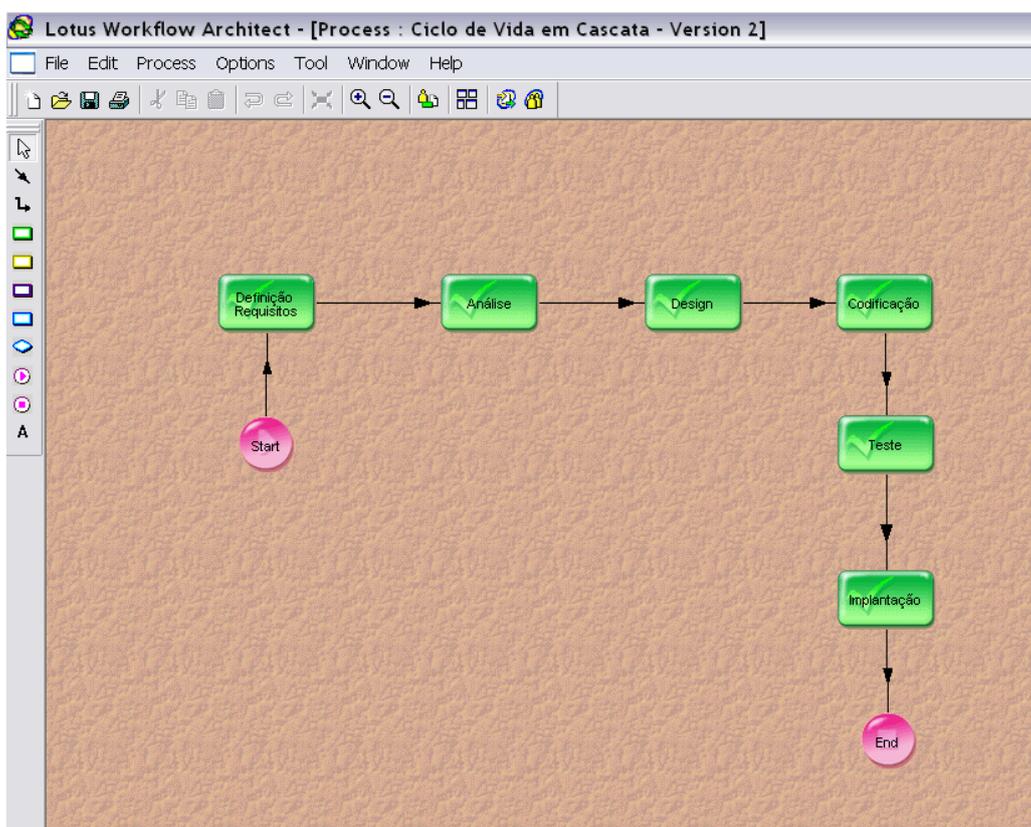


Figura 3.8 – Exemplo de Mapa de Processo no *Lotus Workflow Architect* (*Lotus Workflow*)

- **Papéis**

No *Lotus Workflow Architect* os papéis podem ser atribuídos tanto ao processo quanto a cada atividade. Os papéis podem ser desempenhados por usuários, grupos, departamentos ou papéis definidos na base de Organização. Na figura 3.9, vemos um exemplo da seção *owner* de uma atividade.

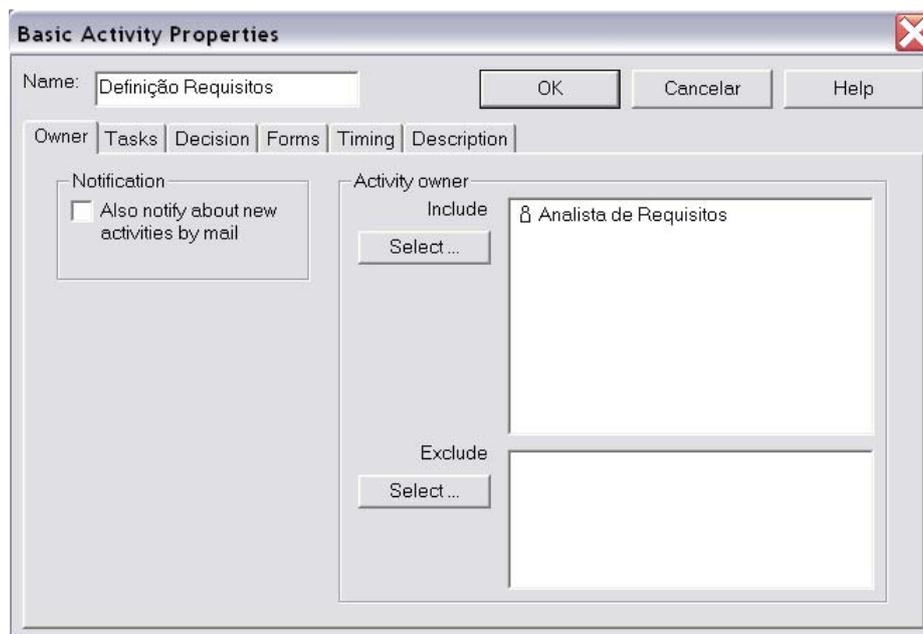


Figura 3.9 – *Owner* de Atividade no *Lotus Workflow Architect* (*Lotus Workflow*)

- **Pacote de Trabalho**

No *Lotus Workflow Architect* o pacote de trabalho de cada processo e de cada atividade é definido através das propriedades básicas na seção *Forms*. A figura 3.10 exemplifica as possibilidades de pacotes de trabalho na ferramenta, onde podem ser definidos o formulário principal e todos os outros documentos que poderão ser criados para a manipulação durante a execução.

- **Recursos utilizados na execução de tarefas**

Ao definir as tarefas que podem ser executadas em uma atividade, o *Lotus Workflow Architect* permite que se associem recursos a cada uma delas. Os recursos são aplicativos cadastrados pelos administradores na base de Organização. No exemplo ilustrado na figura 3.11, foi selecionado o *Rational RequisitePro* para apoiar a tarefa de criação de projeto.

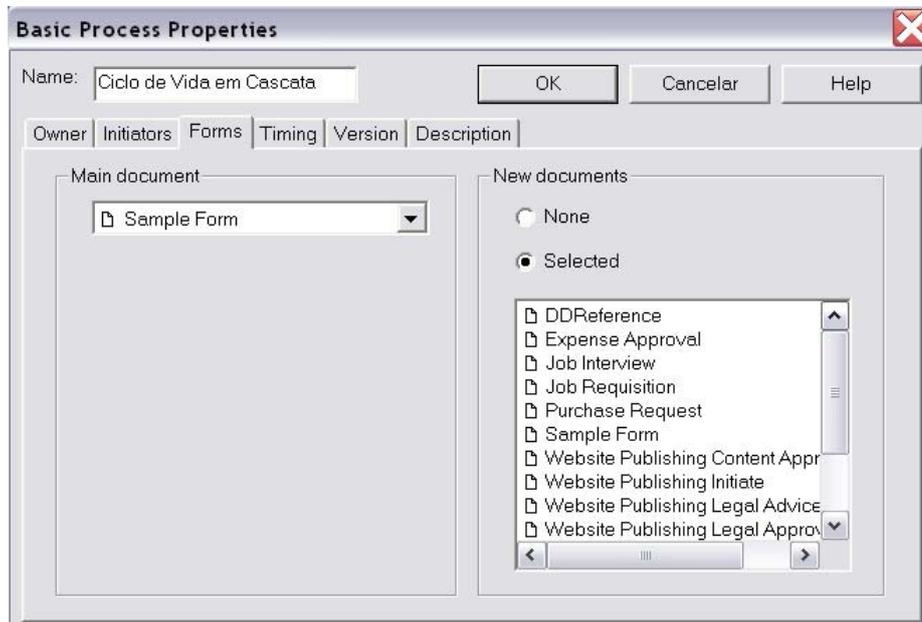


Figura 3.10 – Pacote de Trabalho no *Lotus Workflow Architect* (*Lotus Workflow*)

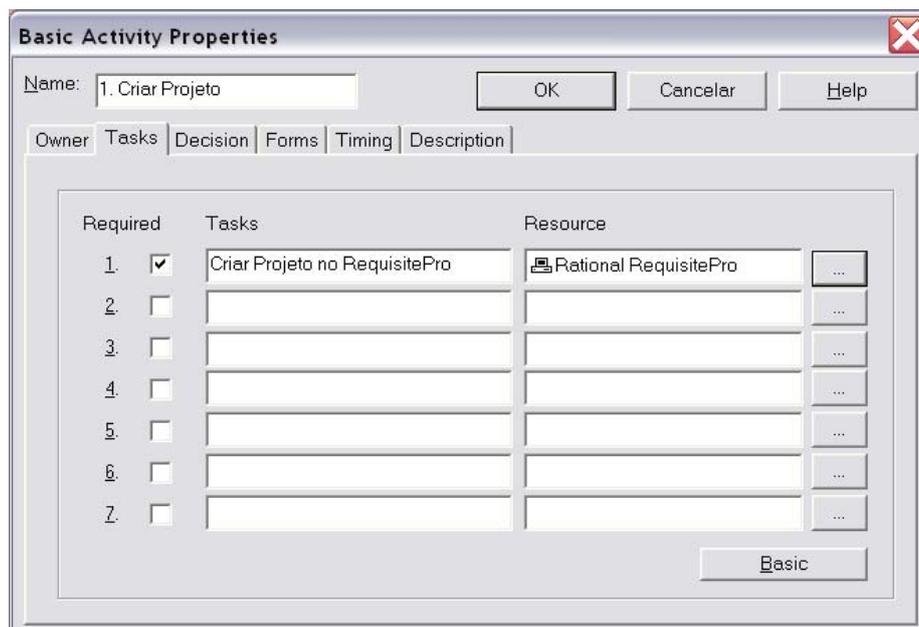


Figura 3.11 – Associação de Recursos a Tarefas no *Lotus Workflow Architect* (*Lotus Workflow*)

3.5.3. Base de Aplicação

▪ Lista de Trabalho

No *Lotus Workflow* a lista de trabalho é implementada através da visão *My Activities* disponível na base de Aplicação. Esta visão mostra todas as atividades de um participante, informando o *job* (instância de processo) do qual elas fazem parte e a prioridade de cada uma. Esta visão é ilustrada através da figura 3.12.

Job	Activity	Participants	Claim by	Priority
Controle de Estoque	Definição Requisitos (In Progress) - no subject (Sample Form)	▲ Andrea Magalhaes Magdaleno (AO)		2. Medium
Sistema de Pessoal	Definição Requisitos (In Progress) - no subject (Sample Form)	▲ Andrea Magalhaes Magdaleno (AO)		2. Medium

Figura 3.12 – Visão *My Activities* da Base de Aplicação (*Lotus Workflow*)

▪ Execução de *Jobs*

Um *job* é uma instância de um processo que consiste de atividades que devem ser completadas numa ordem pré-definida. À medida que uma atividade é concluída, a atividade seguinte é automaticamente instanciada, até que todo o *job* seja terminado. A base de Aplicação oferece visões que permitem aos participantes localizar os *jobs* de acordo com critérios como nome, prazo, status e participantes.

▪ Execução de Atividades

Durante a execução de cada atividade, deverão ser criados os documentos definidos e realizadas as tarefas indicadas pelo *checklist* que aparece no formulário principal. O executor deve assinalar as tarefas conforme elas vão sendo concluídas. O *Lotus Workflow* não permite que se conclua uma atividade até que todas as suas tarefas sejam marcadas como terminadas.

No caso das tarefas que foram associadas a algum recurso, aparece ao seu lado um ícone

que permite o acesso direto ao aplicativo escolhido. A figura 3.13 ilustra esta funcionalidade. Ao lado da tarefa “ Criar Projeto” aparece um botão que quando clicado abre automaticamente o *Rational RequisitePro* para que o usuário possa utilizar o aplicativo necessário sem sair da ferramenta de *workflow*.

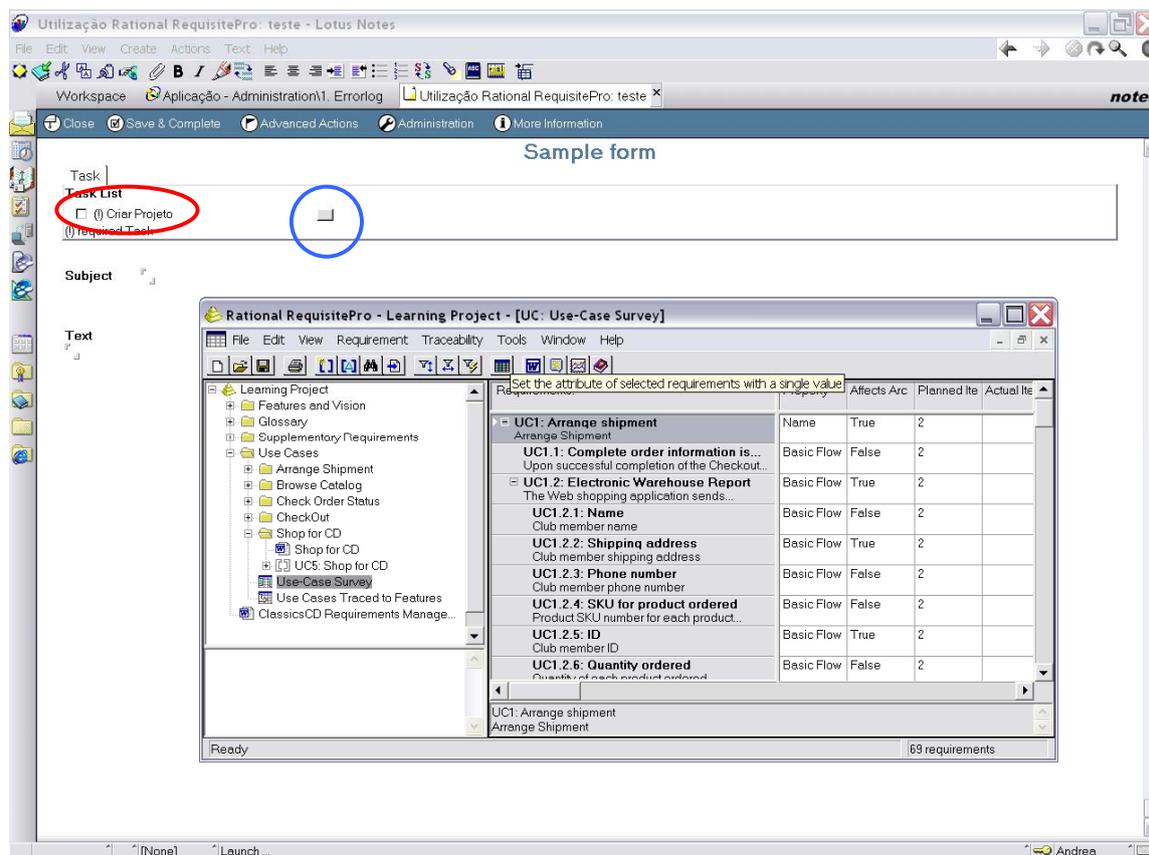


Figura 3.13 – Execução da Tarefa de Criar Projeto no *RequisitePro* (*Lotus Workflow*)

3.5.4. Base de Organização

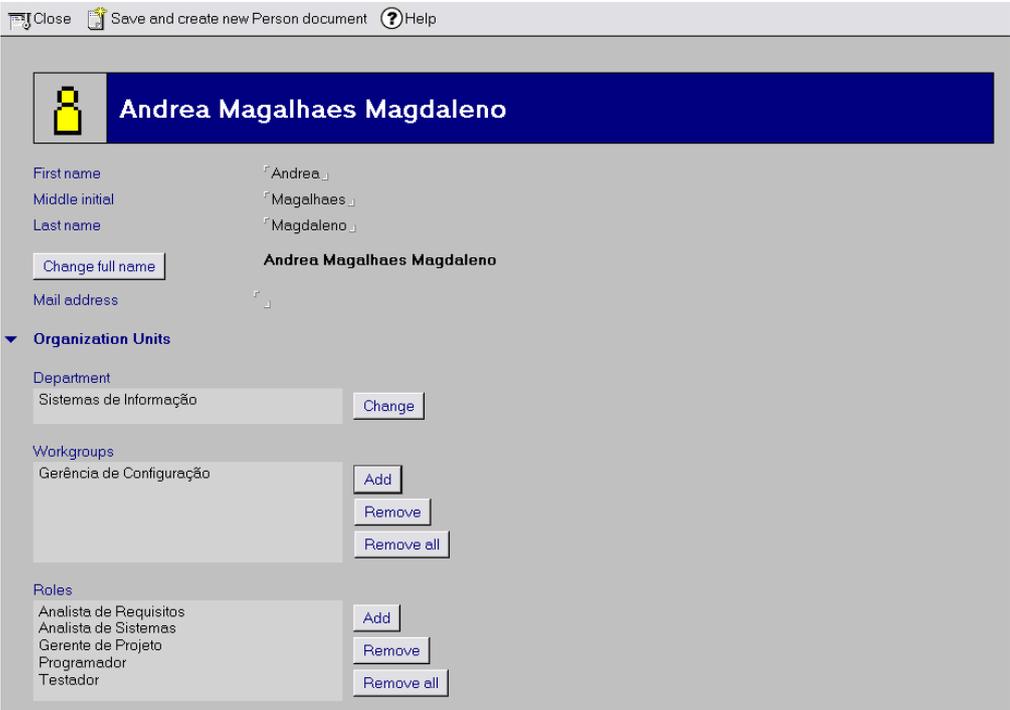
- **Cadastro de Participantes**

A figura 3.14 ilustra o cadastro de um usuário no *Lotus Workflow* através da base da Organização: são informados o nome completo, o e-mail, o departamento no qual o usuário trabalha, os grupos de trabalho dos quais ele participa e os papéis que desempenha.

- **Cadastro de Grupo de Usuários**

Os sistemas de *workflow* permitem que sejam definidos grupos de usuários que serão definidos como executores de atividades em processos.

Um exemplo de definição de um grupo de trabalho no *Lotus Workflow* pode ser visualizado na figura 3.15, onde foi criado o grupo Gerência de Configuração através de base de Organização. Cada grupo de usuário possui um gerente responsável e a sua lista de usuários cadastrados. Um grupo também pode fazer parte de outros grupos maiores.



The screenshot displays the user profile interface for Andrea Magalhaes Magdaleno. At the top, there is a header with the user's name and a yellow icon. Below the header, the user's details are listed: First name (Andrea), Middle initial (Magalhaes), Last name (Magdaleno), and Mail address. A 'Change full name' button is present. The 'Organization Units' section shows the Department as 'Sistemas de Informação' with a 'Change' button. The 'Workgroups' section lists 'Gerência de Configuração' with 'Add', 'Remove', and 'Remove all' buttons. The 'Roles' section lists 'Analista de Requisitos', 'Analista de Sistemas', 'Gerente de Projeto', 'Programador', and 'Testador', each with 'Add', 'Remove', and 'Remove all' buttons.

Figura 3.14 – Cadastro de Participante na Base de Organização (*Lotus Workflow*)



Figura 3.15 – Grupo de Usuários na Base de Organização (*Lotus Workflow*)

3.5.5. Lotus Workflow Viewer

Esta ferramenta permite que se acompanhe graficamente a execução dos *jobs*. Também é possível visualizar informações detalhadas do processo sob o qual está baseado este *job*. A figura 3.16 ilustra a interface desta ferramenta, que possibilita também a animação da execução do *job*.

A figura 3.16 ilustra como o *Lotus Workflow Viewer* utiliza cores distintas para informar o status da atividade e a rota seguida pelos documentos. A posição no momento, na execução de um *job*, é caracterizada pelo desenho de um documento, como visto logo após a primeira atividade (Definição de Requisitos). Todas as atividades terminadas, para esta interface, aparecem coloridas com um verde mais escuro. Através da opção *Options* é possível customizar a interface apresentada. O usuário também pode consultar as definições de cada atividade ou do processo.

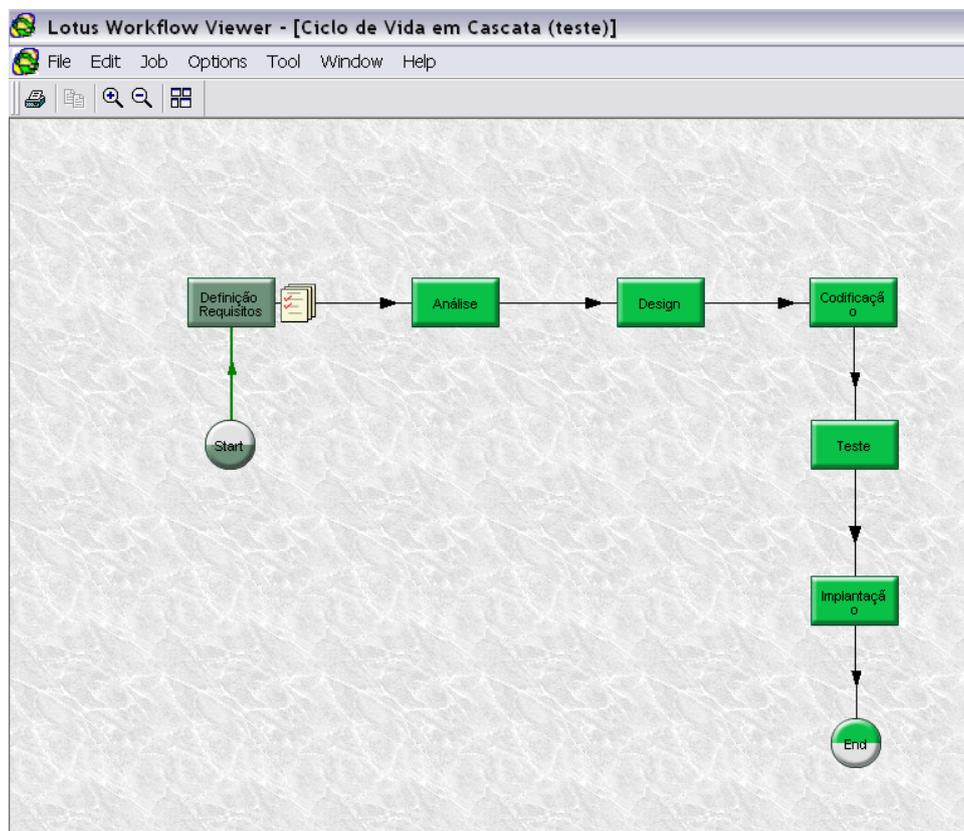


Figura 3.16 – Interface do *Lotus Workflow Viewer* destacando as atividades já executadas e o trâmite dos documentos (*Lotus Workflow*)

3.6. Benefícios

Workflow, embora seja uma tecnologia relativamente nova, já é suficientemente madura para que possa ser adotada por algumas empresas, desde que sejam cumpridas certas regras e que a metodologia utilizada tenha sido adequada para resolver os pontos relevantes para a sua utilização. Não há limites para a utilização desta tecnologia, que pode ser adotada sozinha ou em conjunto com outras tecnologias: na automatização de processos; em processos com grande número de usuários; localmente e globalmente; via Internet, Intranet e Extranet (CRUZ, 2000). Entretanto, como em qualquer tecnologia, existem parâmetros que devem ser obedecidos para garantir o sucesso da sua implantação. A correta modelagem dos processos de uma organização e a sua correspondente implementação através de sistemas de *workflow* pode trazer diversas vantagens:

➤ *Redução de custos*

Apesar dos altos investimentos iniciais necessários, a implantação de uma ferramenta de *workflow* permite um melhor aproveitamento da equipe, já que proporciona uma nova forma de executar as atividades na organização. Neste sentido, a solução de *workflow* pode ter um menor custo de operação se comparado com outros investimentos da área de desenvolvimento de *software*. Por isso, este é um dos benefícios mais rapidamente percebido nas organizações.

➤ *Redução da circulação de documentos em papel*

A designação automática de tarefas acompanhadas de todas as informações e documentos relevantes para a sua execução permite a manipulação eletrônica de documentos e ajuda a reduzir a circulação de documentos em papel, mas não necessariamente a reprodução.

➤ *Controle dos processos de trabalho*

Uma boa solução de *workflow* oferece maior visibilidade e controle dos processos de trabalho, seja por parte de quem executa, de quem gerencia ou de quem depende destas informações para prestar um melhor serviço. Uma ferramenta de *workflow* permite monitorar sobrecargas de trabalho ou ociosidade de grupos de trabalho. A possibilidade de controle, aliada a uma visão clara do funcionamento do processo, facilita a localização e correção de erros e gargalos, ponto inicial para atingir uma maior eficiência na execução do processo.

➤ *Garantia da Integridade do processo*

A garantia da integridade do processo evita que as normas e procedimentos estabelecidos pela organização deixem de ser praticados, seja por esquecimento, desconhecimento ou tentativa de fraude. Os procedimentos documentados são seguidos pelo sistema de *workflow*, garantindo que o trabalho seja realizado da forma planejada pela gerência e atendendo os requisitos de negócio ou regulatórios.

➤ *Segurança e privacidade das informações*

Uma ferramenta de *workflow* permite que somente usuários autorizados possam acessar uma tarefa e as informações associadas a ela.

➤ *Priorização de tarefas*

O envio de tarefas em ordem decrescente de prioridade ajuda a garantir que o trabalho

mais importante será feito primeiro, o que aumenta a produtividade.

➤ *Automação dos processos*

A automação dos processos de negócio ajuda a garantir que a pessoa mais capacitada será a designada para realizar uma determinada tarefa e melhora a qualidade do serviço por permitir que seja oferecida uma abordagem padrão para cada cliente.

Um sistema de *workflow* livra as pessoas da rotina de extensos trabalhos administrativos, ao automatizar a operação, proporcionando a elas um relevante conhecimento do fluxo de trabalho. Assim, os gerentes podem focar nos problemas do negócio ao invés de se ocupar da rotina de encaminhar e monitorar tarefas, pois eles possuem maior controle sobre o processo.

➤ *Interoperabilidade e distribuição*

Estas ferramentas de *workflow* atualmente existentes, de forma geral, estão aptas a operar de forma integrada em ambientes modernos, ou seja, geograficamente distribuídos, heterogêneos e com múltiplas aplicações. A utilização de sistemas de *workflow* permite ainda que as atividades desenvolvidas por setores diferentes da organização possam ser conectadas.

➤ *Separação da definição do processo da sua execução*

A separação formal da definição do processo de sua execução permite uma maior flexibilidade em face de mudanças em sua definição, dado que processos sofrem modificações com o tempo (ARAUJO, 2000).

➤ *Melhoria do atendimento ao cliente*

A melhor execução dos processos reflete automaticamente no relacionamento da organização com o cliente. A utilização de um sistema de gerenciamento de *workflow* acoplado com iniciativas de CRM ou de call-center traz como benefícios: a melhora a qualidade do serviço, redução do número de chamadas e diminuição da perda de clientes provocadas por atritos de relacionamento e falta de informações.

Resumindo: tendo o profissional mais capacitado realizando o trabalho mais prioritário, seguindo corretamente os procedimentos da organização conseguimos respostas mais rápidas que geram custos mais baixos e serviços de melhor qualidade para o cliente. Com o trabalho distribuído igualmente e com a confiança de que eles estão trabalhando na coisa certa, os

indivíduos se sentem mais satisfeitos. Desta forma o *workflow* é bom para a organização, para os clientes e para os participantes.

3.7. Desafios

Similar a outras tecnologias, a automação dos processos através de uma ferramenta de *workflow* envolve uma mudança cultural dentro da organização, pois as pessoas precisam modificar sua forma de trabalhar para que a tecnologia consiga prover seus benefícios. Assim, torna-se essencial vencer a resistência natural das pessoas que temem perder as suas funções e acabam boicotando o projeto.

Por estes motivos, a introdução da tecnologia de *workflow* precisa ser cuidadosamente planejada e acompanhada de perto para que possa ser adaptada às circunstâncias geradas pela própria introdução da tecnologia. Uma forma de amenizar as chances de insucesso é a implantação gradativa dos objetivos, sincronizando-os com as mudanças culturais promovidas (ARAUJO, 2000).

Um dos maiores desafios para a adoção de uma solução de *workflow* é o alto investimento necessário. Para que este investimento valha a pena, a alta gerência precisa estar convencida dos problemas na execução dos processos e deve patrocinar a solução para ajudar a vencer as resistências dentro da equipe.

Além disso, a área tecnológica deve ser capaz de absorver a nova tecnologia e de mantê-la operacional. Ainda no setor tecnológico, a baixa interoperabilidade de algumas ferramentas com os sistemas legados se tornou um obstáculo para algumas organizações.

4. *Process Beans*

Neste capítulo apresentaremos o conceito de *process beans* detalhando os motivos que levaram ao surgimento desta idéia, bem como os benefícios da sua utilização. Em seguida, serão discutidos os dois tipos de *process beans* existentes (*framework* e *blackbox*), bem como exemplos para ilustrá-los.

4.1. Motivação

Algumas organizações ainda não possuem uma definição dos seus processos. Aquelas que já possuem esta definição, geralmente disponibilizam-na de forma dispersa em documentos extensos, não automatizados, inconsistentes com a execução prática, imprecisos, ambíguos ou incompreensíveis. Isto pode se dar pelo fato delas não possuírem o conhecimento ou o ferramental necessário para apoiar esta atividade. Na maioria dos casos, os processos não seguem regras claras e bem definidas, variando de acordo com os envolvidos.

Entretanto, será necessário definir um único procedimento de trabalho que contemple as variações aceitas pela organização. Este procedimento deve servir como guia para o estabelecimento de um processo comum na organização. Algumas razões que podem ser indicadas como fatores motivadores para esta padronização são:

- Redução dos problemas relacionados a treinamento, revisões e suporte a ferramentas;
- As experiências adquiridas durante as execuções são incorporadas ao procedimento de trabalho e contribuem para melhorias em todos os processos definidos;
- Economia de tempo e esforço na definição de novos processos adequados a casos de execução.

Entretanto, quando os padrões definidos são automatizados através de uma ferramenta de *workflow* ou através de outros sistemas de *software*, torna-se mais claro que ele não atende perfeitamente às especificidades de cada caso de execução, devido ao seu nível de detalhamento. Por exemplo, as equipes que não têm muitas habilidades ou treinamento numa tarefa específica tendem a precisar de maiores detalhes para guiar seu trabalho. Entretanto, quando a equipe se

mostra mais hábil e treinada, as pessoas tendem a se sentir limitadas pelo elevado nível de controle sobre as suas atividades. Desta forma, ser adaptável e configurável torna-se um importante requisito a ser atingido na definição de um padrão de procedimento de trabalho. Porém, esta não é uma tarefa trivial.

Tendo esta motivação em vista, surgiu a idéia da utilização de componentes de processo denominados *process beans* que têm como objetivo agilizar a tarefa de modelagem de processos, de forma que os mesmos possam ser definidos a partir de processos pré-existentes (MENDES, 2003). Esta idéia foi apresentada em 1999 por PARRINI, CAMPOS e BORGES focando em componentes para o ambiente analítico. Em 2000, SOBRINHO também discutiu a utilização de componentes para processos com o intuito de incentivar e facilitar o uso de processos de *software* nas organizações. Mais recentemente, em 2003, MENDES utilizou a idéia de componentes de processo para apoiar as atividades de pós-reunião. Neste trabalho, estaremos propondo a utilização dos *process beans* independente do domínio de aplicação.

4.2. Definição

Process beans são definidos como componentes de processo reutilizáveis, para uso em sistemas de *workflow* (MENDES, 2003). Em sua essência, esses componentes são pré-definições de práticas/passos de processos implementados em uma ferramenta de *workflow* (SOBRINHO, 2000) e que podem ser combinados ou adaptados a cada novo caso de execução. Eles também poderão ser conectados aos padrões da organização tanto para refinar as atividades quanto para modelar novos processos através da composição com outros *beans*.

Os *process beans* podem ser de dois tipos: *framework beans* e *blackbox beans*. Os componentes do tipo *framework beans* são *containers* de atividades comuns para um grande número de processos. Eles são úteis para prover uma arquitetura de processo que guie a definição do fluxo de trabalho baseado nos padrões da organização. No contexto de projetos de desenvolvimento de *software*, um exemplo de *framework beans* seria a definição do ciclo de vida de um projeto, conforme visto na figura 4.1. Por se tratar de um processo de alto nível, permite que cada atividade seja realizada de forma customizada pelas equipes, respeitando-se as definições da organização.

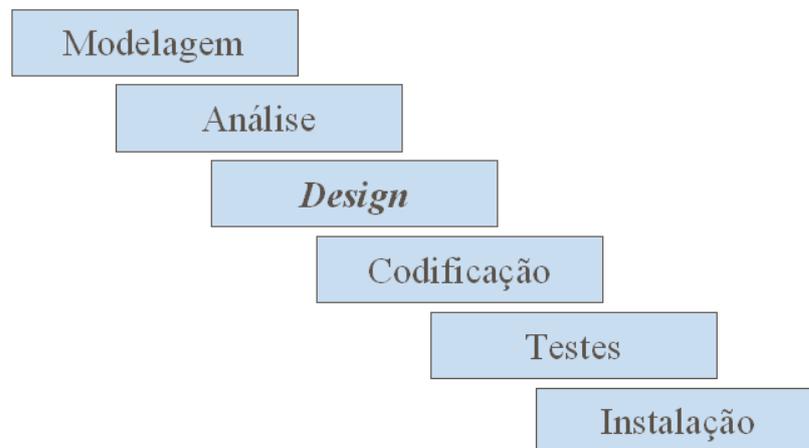


Figura 4.1 – Ciclo de Vida em Cascata (PRESSMAN, 1997)

Os *blackbox beans* seriam componentes menores e de uso mais específico. Este tipo de *process bean* é oferecido para ser utilizado inteiramente da forma como foi definido. Na Engenharia de *Software*, um exemplo deste tipo de *bean* poderia ser a implementação de uma classe ou método dentro da atividade de codificação.

A principal diferença entre *framework beans* e *blackbox beans* é o fato do *framework* possuir espaços em aberto que podem, opcionalmente, ser preenchidos com outros *beans*. Desta forma, os *framework beans* permitem que pontos pré-determinados da sua definição sejam expandidos por outros *beans*. Um *framework bean* pode possuir em cada uma das suas atividades, um ponto de acesso para a inserção de outros *beans*. Já os *blackbox beans*, não podem ser complementados desta forma. Tanto um *blackbox bean* quanto um *framework bean* podem ser utilizados para ocupar um dos espaços disponíveis em outro *framework bean*. Algumas possibilidades de combinação destes tipos de *beans* estão ilustradas na figura 4.2.

Nesta figura, percebe-se pelos dois primeiros exemplos que tanto *blackbox* quanto *framework beans* podem ser conectados em uma seqüência linear. Este cenário ocorre, caso opte-se por não preencher nenhum dos espaços disponíveis no *framework bean*. A figura também mostra a capacidade de um *framework bean* de ser complementado por *beans* de ambos os tipos. Além disso, fica claro que a combinação de *beans* pode ter um número ilimitado de níveis, de acordo com o grau de detalhe desejado na definição do processo.

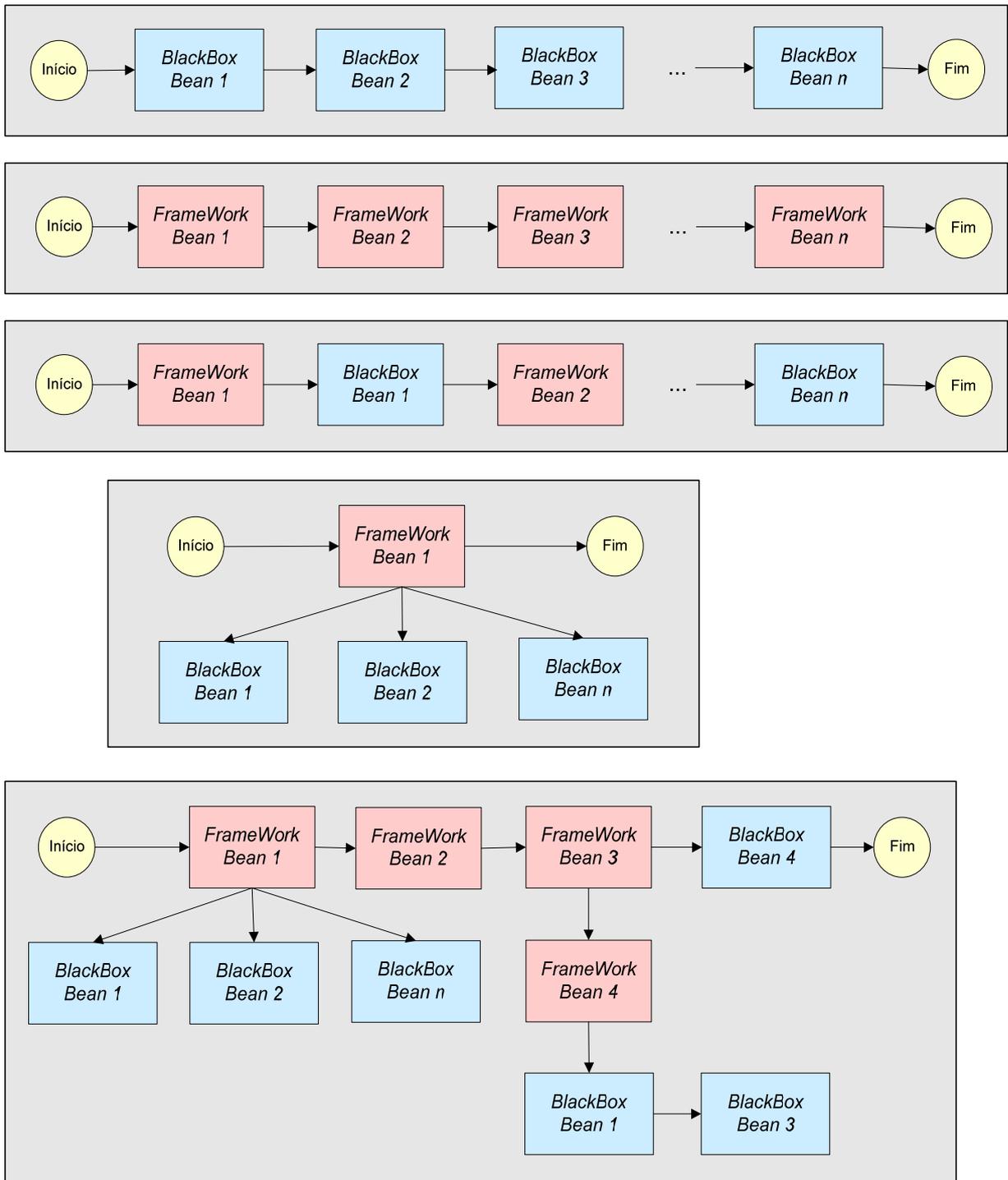


Figura 4.2 – Exemplo de Combinação de Beans

Mais uma vez retornando à Engenharia de *Software*, podemos ilustrar um exemplo do conjunto de *beans* oferecidos por uma organização que adotou o ciclo de vida em cascata. Neste exemplo, cada uma das atividades do ciclo de vida em cascata, é considerada um *bean* com um maior nível de detalhe. Os *framework beans* foram representados em rosa e os *blackbox beans* foram representados em azul na ilustração da figura 4.3.

Apesar do exemplo mostrar apenas as atividades de Análise e Testes como *framework beans*, é importante destacar que qualquer uma das outras atividades também poderia ser um *framework bean*, de acordo com o nível de detalhe que se deseje empregar. Além disso, as próprias atividades de Análise e Testes poderiam ser *blackbox beans* se a organização assim o desejasse.

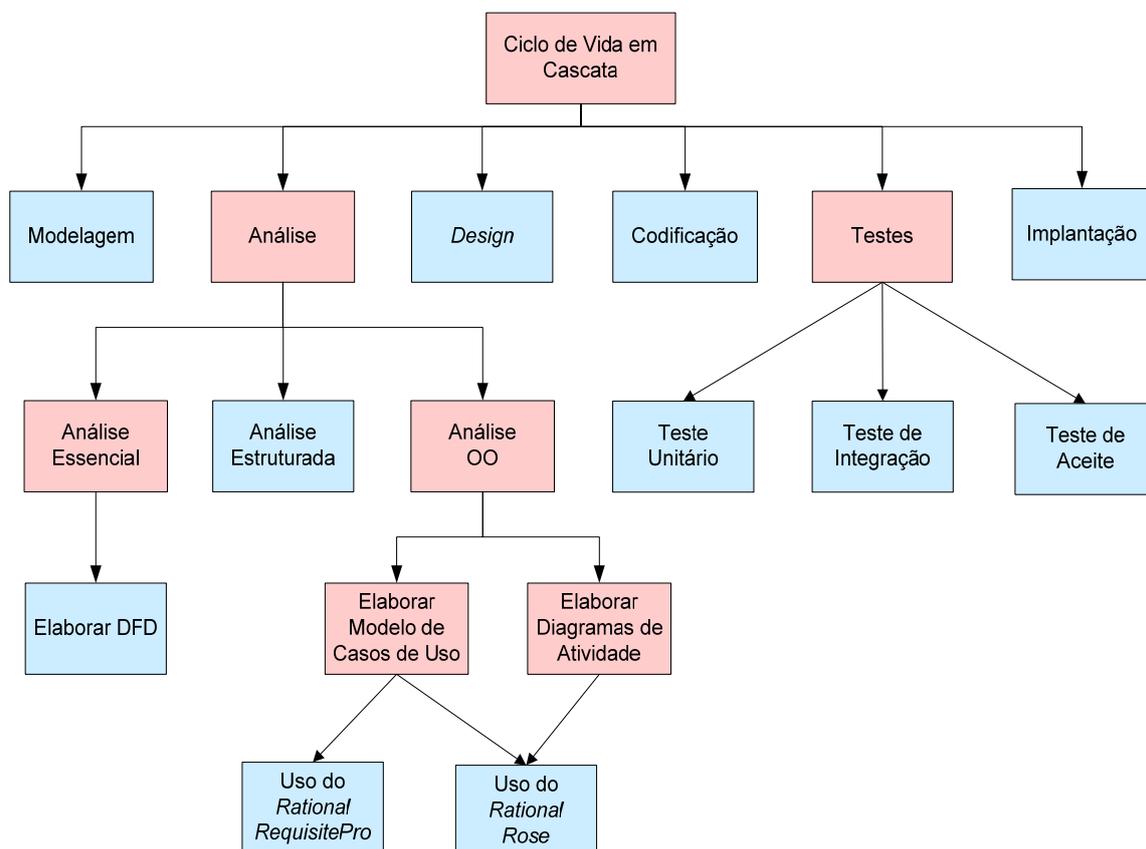


Figura 4.3 – Exemplo de Combinação de *Beans* na Engenharia de *Software*

Para compor a atividade de Análise, existiriam três opções de *beans* a serem escolhidos de acordo com a metodologia utilizada no desenvolvimento. Para cada uma das metodologias

escolhidas (Análise Estruturada, Análise Essencial ou Análise OO) existiria um conjunto diferente de *beans*. Desta forma, para cada novo projeto de desenvolvimento, o processo que será seguido pode ser montado combinando-se *beans* relacionados às características da metodologia.

Também é importante perceber que o mesmo *blackbox bean* (Uso do *Rational Rose*), pode ser usado para preencher mais de um *framework bean*. Esta é uma das mais importantes características dos *beans*: a possibilidade de serem reutilizados em diferentes casos de execução, sem que seja necessário repetir toda a sua definição. Desta forma, espera-se diminuir o tempo gasto na definição dos processos nas organizações e aproveitar as experiências adquiridas em casos de sucesso anteriores.

Resumindo: para usar os *process beans*, a organização deverá definir os *beans* que serão disponibilizados, permitir que cada equipe acrescente o detalhamento desejado aos *beans* para que os requisitos do projeto sejam cumpridos, construir um processo que co-relacione os *beans* selecionados, e, utilizando-se um sistema de *workflow*, instanciar este processo para que a equipe possa executá-lo.

5. Ferramenta

Este capítulo apresenta a ferramenta *Beans Composer*, proposta como solução para este trabalho, demonstrando as suas principais funcionalidades e detalhando a forma como ela foi implementada.

5.1. Objetivos

Uma vez que a organização já possui os seus procedimentos definidos e que se preocupou com a componentização dos seus processos, a cada novo caso de execução cabe ao responsável ou líder da equipe, montar o processo que será utilizado tendo como base os componentes disponíveis. Entretanto, como visto anteriormente, esta não é uma tarefa trivial. Para apoiá-la, é oferecido um repositório ou biblioteca de componentes de processo, comuns a um dado domínio. Estes componentes, já modelados e disponibilizados para reuso, são apresentados de forma clara e organizada na biblioteca.

Mesmo assim, muitas vezes, os *beans* disponibilizados na biblioteca, não poderão ser utilizados inteiramente da forma como foram definidos, pois as equipes são heterogêneas e, portanto, necessitam de diferentes níveis de detalhamento para o processo que será seguido. Além disso, os *beans* também devem ser capazes de se adaptar a cada necessidade de uso do processo.

Tendo em mente esta motivação foi criada a *Beans Composer*. Esta ferramenta tem como objetivos oferecer os recursos para a criação, manipulação, adaptação e combinação dos *beans* disponibilizando-os através de uma biblioteca. Uma vez combinados, os *beans* darão origem a um processo que será executado na ferramenta de *workflow*.

5.2. Funcionalidades

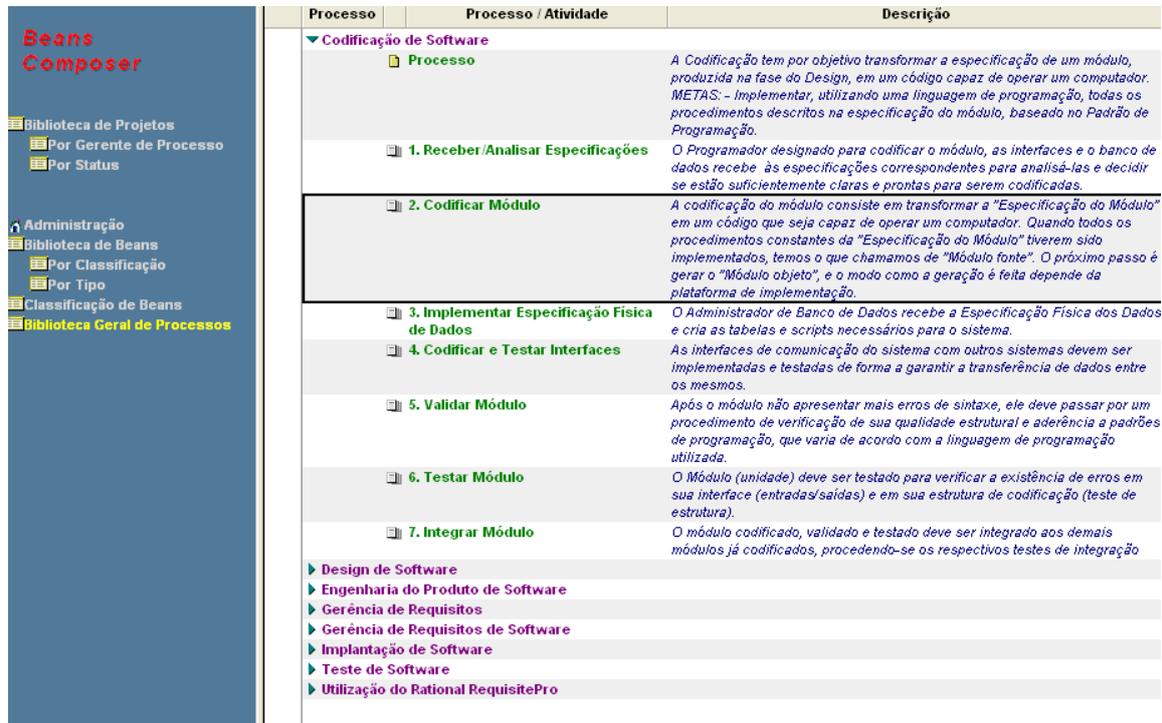
Nesta seção iremos apresentar as principais funcionalidades oferecidas pela *Beans Composer*, de acordo com os conceitos de *beans* já apresentados anteriormente. Todas as funcionalidades desenvolvidas foram documentadas através de casos de uso que se encontram detalhados na seção de Anexos. O diagrama que mostra a interação entre estes casos de uso pode ser encontrado no Anexo I.

Inicialmente, a ferramenta oferece dois tipos de perfis de acesso: Administrador de *Beans* e Gerente de Processo. O Administrador de *Beans* é o responsável por configurar os possíveis domínios de aplicação (aqui denominados de Classificação de *Beans*) e por manter a Biblioteca de *Beans*. O Gerente de Processo é o responsável por montar um novo projeto, de acordo com as particularidades do seu caso de execução, a partir da combinação dos *beans* oferecidos.

Uma vez que a aplicação desenvolvida se propõe a atender mais de um domínio de aplicação, a primeira tarefa do Administrador de *Beans* seria cadastrar as possíveis classificações de *beans*. As orientações para executar esta tarefa estão detalhadas no Caso de Uso Manter Informações de Classificação de *Bean* (Anexo II) e no Manual do Usuário (Anexo XII). Alguns exemplos de classificação de *beans* seriam: Engenharia de *Software*, Reunião, Recursos Humanos etc...

O próximo passo seria a criação dos *beans*. Entretanto, é necessário que, anteriormente, os processos já tenham sido modelados no *Lotus Workflow Architect*. A lista dos processos e atividades modelados, bem como as suas descrições, pode ser consultada através da Biblioteca Geral de Processos existente na *Beans Composer* e ilustrada na figura 5.1.

Uma vez realizada esta etapa de modelagem, o Administrador de *Beans* teria a tarefa de criar os *beans* que serão oferecidos como opções para os Gerentes de Processo. O passo-a-passo para a criação de um *bean* está descrito no Manual do Usuário (Anexo XII). Os detalhes sobre como Criar, Alterar e Excluir um *bean* também estão documentados no Caso de Uso Manter Informações de *Beans* (Anexo III).



Processo	Processo / Atividade	Descrição
▼ Codificação de Software		
	Processo	A Codificação tem por objetivo transformar a especificação de um módulo, produzida na fase do Design, em um código capaz de operar um computador. METAS: - Implementar, utilizando uma linguagem de programação, todas os procedimentos descritos na especificação do módulo, baseado no Padrão de Programação.
	1. Receber/Analisar Especificações	O Programador designado para codificar o módulo, as interfaces e o banco de dados recebe às especificações correspondentes para analisá-las e decidir se estão suficientemente claras e prontas para serem codificadas.
	2. Codificar Módulo	A codificação do módulo consiste em transformar a "Especificação do Módulo" em um código que seja capaz de operar um computador. Quando todos os procedimentos constantes da "Especificação do Módulo" tiverem sido implementados, temos o que chamamos de "Módulo fonte". O próximo passo é gerar o "Módulo objeto", e o modo como a geração é feita depende da plataforma de implementação.
	3. Implementar Especificação Física de Dados	O Administrador de Banco de Dados recebe a Especificação Física dos Dados e cria as tabelas e scripts necessários para o sistema.
	4. Codificar e Testar Interfaces	As interfaces de comunicação do sistema com outros sistemas devem ser implementadas e testadas de forma a garantir a transferência de dados entre os mesmos.
	5. Validar Módulo	Após o módulo não apresentar mais erros de sintaxe, ele deve passar por um procedimento de verificação de sua qualidade estrutural e aderência a padrões de programação, que varia de acordo com a linguagem de programação utilizada.
	6. Testar Módulo	O Módulo (unidade) deve ser testado para verificar a existência de erros em sua interface (entradas/saídas) e em sua estrutura de codificação (teste de estrutura).
	7. Integrar Módulo	O módulo codificado, validado e testado deve ser integrado aos demais módulos já codificados, procedendo-se os respectivos testes de integração
	Design de Software	
	Engenharia do Produto de Software	
	Gerência de Requisitos	
	Gerência de Requisitos de Software	
	Implantação de Software	
	Teste de Software	
	Utilização do Rational RequisitePro	

Figura 5.1 – Biblioteca Geral de Processos – Lista dos Processos Modelados

Ao criar um *bean*, o Administrador de *Beans* deve ter alguns cuidados:

- Escolher um nome semântico que ajude a identificar o *bean* de forma clara. Isso facilita a tarefa dos Gerentes de Processo ao definir os *beans* que serão utilizados no seu projeto.
- Se preocupar com as possibilidades de combinação ao definir se o *bean* será *framework* ou *blackbox*. Caso opte por um *framework bean*, ele também precisará definir em quais atividades poderão existir um maior nível de detalhe.
- Analisar os possíveis domínios de aplicação dos quais o *bean* poderá fazer parte.
- Documentar de forma clara e rica os objetivos, contexto e solução do *bean* que será oferecido para suportar o entendimento dos demais usuários.

Todos os *beans* cadastrados são disponibilizados na Biblioteca de *Beans*, onde podem ser consultados por todos os usuários da aplicação. A Biblioteca de Beans é apresentada na figura 5.2.

Bean	Tipo	Classificação	Objetivos
Codificação de Software	BlackBox Bean	ENGENHARIA DE SOFTWARE	A Codificação tem por objetivo transformar a especificação de um módulo, produzida na fase do Design, em um código capaz de operar um computador. METAS: - Implementar, utilizando uma linguagem de programação, todos os procedimentos descritos na especificação do módulo, baseado no Padrão de Programação.
Design de Software	Framework Bean	ENGENHARIA DE SOFTWARE	Produzir uma solução implementável e eficiente para atender aos requisitos de software, a partir de um modelo de abstração lógica. METAS: - Garantir que todos os requisitos alocados ao sistema sejam especificados, estabelecendo a estrutura de codificação.
Engenharia do Produto de Software	Framework Bean	ENGENHARIA DE SOFTWARE	Executar de forma organizada e consistente as atividades necessárias ao Processo de Engenharia do Produto ao se produzir um novo sistema ou evoluir um sistema já existente. METAS: - Garantir a consistência entre os produtos gerados ao longo do Ciclo de Vida do Desenvolvimento do Software; - Aumento da qualidade do software implementado em produção; - Facilidade de manutenção, contribuindo para o aumento do controle e diminuição de riscos; - Aumento da produtividade no ciclo de vida desenvolvimento do software, acarretando em
Gerência de Requisitos de Software	Framework Bean	ENGENHARIA DE SOFTWARE	Estabelecer e manter um entendimento comum entre as unidades de negócio e a equipe do projeto com relação aos requisitos a serem atendidos pelo(s) sistema(s). Os requisitos alocados serão usados como base para o planejamento e demais atividades do projeto.
Implantação de Software	Framework Bean	ENGENHARIA DE SOFTWARE	Este processo de Implantação tem por objetivo alcançar a aceitação da solução pelo cliente e a implantação desta solução em produção. METAS: - Garantir que o produto de software estará disponível para os usuários; - Garantir que os usuários foram adequadamente treinados para que estejam capacitados a utilizar o sistema com sucesso; - Garantir que toda a documentação necessária para a instalação e utilização do sistema encontra-se disponível e completa; - Implantar o sistema na data prevista no Plano de Implantação; - Garantir
Teste de Software	Framework Bean	ENGENHARIA DE SOFTWARE	O Teste de Software tem por objetivo validar de forma completa e sistemática os requisitos do sistema e verificar seus componentes de acordo com o projeto, visando identificar o maior número de defeitos antes da entrega, diminuindo custos de re-trabalho e tempo de entrega do

Figura 5.2 – Biblioteca de Beans

Uma vez que o Administrador de Beans já efetuou os cadastros dos beans, a Beans Composer já se encontra preparada para começar a ser utilizada pelos Gerentes de Processo que serão os responsáveis por montar os seus projetos através da combinação dos beans. Para isso, a sua primeira tarefa é criar um projeto. Esta operação encontra-se descrita no Manual do Usuário (Anexo XII) e no caso de uso Manter Informações de Projeto (Anexo IV).

Um projeto é o ponto de partida para a combinação dos beans. Cada projeto possui um nome único e só poderá ser alterado pelo Gerente de Processo que o criou. Um projeto possui dois status: “ Em Elaboração” e “ Ativo” . Um projeto em elaboração é aquele que ainda encontra-se em fase de definição pelo Gerente de Processo. Um projeto ativo é aquele que já teve todos os seus beans combinados e foi considerado pronto pelo Gerente de Processo. Todos os projetos criados estarão listados na Biblioteca de Projetos, apresentada na figura 5.3. Esta biblioteca também permite a localização de um determinado projeto a partir do nome do Gerente de Processo ou através do status atual do projeto.

Projeto	Etapas	Classificação	Tipo
▼ Sistema de Abastecimento de Navios			
Projeto			
✓1)	Gerência de Requisitos de Software	EINGENHARIA DE SOFTWARE	Framework
✓2)	Design de Software	EINGENHARIA DE SOFTWARE	Framework
✓3)	Codificação de Software	EINGENHARIA DE SOFTWARE	BlackBox
✓4)	Teste de Software	EINGENHARIA DE SOFTWARE	Framework
▼ Sistema de Bombas			
Projeto			
✓1)	Engenharia do Produto de Software	EINGENHARIA DE SOFTWARE	Framework

Figura 5.3 – Biblioteca de Projetos

Um projeto é composto por um número ilimitado de etapas. As etapas são os passos que serão realizados durante o projeto. Cada etapa possui um *bean* associado e uma classificação. O Gerente de Processo pode associar qualquer *blackbox* ou *framework bean* com uma determinada etapa.

A partir de um *framework bean*, o Gerente do Processo tem duas opções: utilizar a definição do processo exatamente como ela está definida em alto nível. Esta opção é útil para equipes que já possuam conhecimento nas tarefas envolvidas e que não necessitem de muitos detalhes ou controle sobre a sua execução; utilizar outro *framework* ou *blackbox bean* como subprocesso de uma etapa. Assim, ele consegue customizar uma tarefa específica, mas continua sendo guiado pelos padrões da organização.

Para oferecer esta segunda opção foi criado o conceito de atividade aberta, disponível apenas nos *framework beans*. Uma atividade aberta representa o ponto onde o Gerente de Processo tem a possibilidade de plugar um outro *bean* caso deseje. Ao completar este espaço, o Gerente de Processo pode optar por um *bean* do tipo *framework* que também possuirá as suas próprias atividades abertas passíveis de serem preenchidas e assim sucessivamente, de forma ilimitada.

Depois que o Gerente de Processo concluir a definição do novo projeto, ele poderá ativá-lo. Um projeto ativo torna-se disponível para ser executado na base de Aplicação e, por isso, não pode mais ser alterado. Esta operação encontra-se descrita no Manual do Usuário (Anexo XII) e no caso de uso Ativar Projeto (Anexo IX).

5.3. Exemplo

Para exemplificar a utilização da ferramenta *Beans Composer*, bem como a combinação dos *framework* e *blackbox beans*, construímos um projeto fictício no domínio da Engenharia de *Software* – Projeto Verão. Neste projeto, estaremos supondo que a organização já havia se preocupado em organizar os seus processos de forma componentizada. Assim, do conjunto de *beans* disponibilizados pela organização, o nosso Gerente de Processo selecionou aqueles que mais se adequavam às particularidades do seu projeto e ao perfil da sua equipe. Neste exemplo, escolhemos um *framework bean* do Ciclo de Vida em Cascata, cuja definição completa encontra-se disponível no Anexo XIV. Este *framework bean* possui todas as suas atividades em aberto, porém optou-se por preencher somente a atividade de Teste com um *blackbox bean* de Teste, detalhado no Anexo XV. O mapa do processo resultante pode ser visualizado na Figura 5.4.

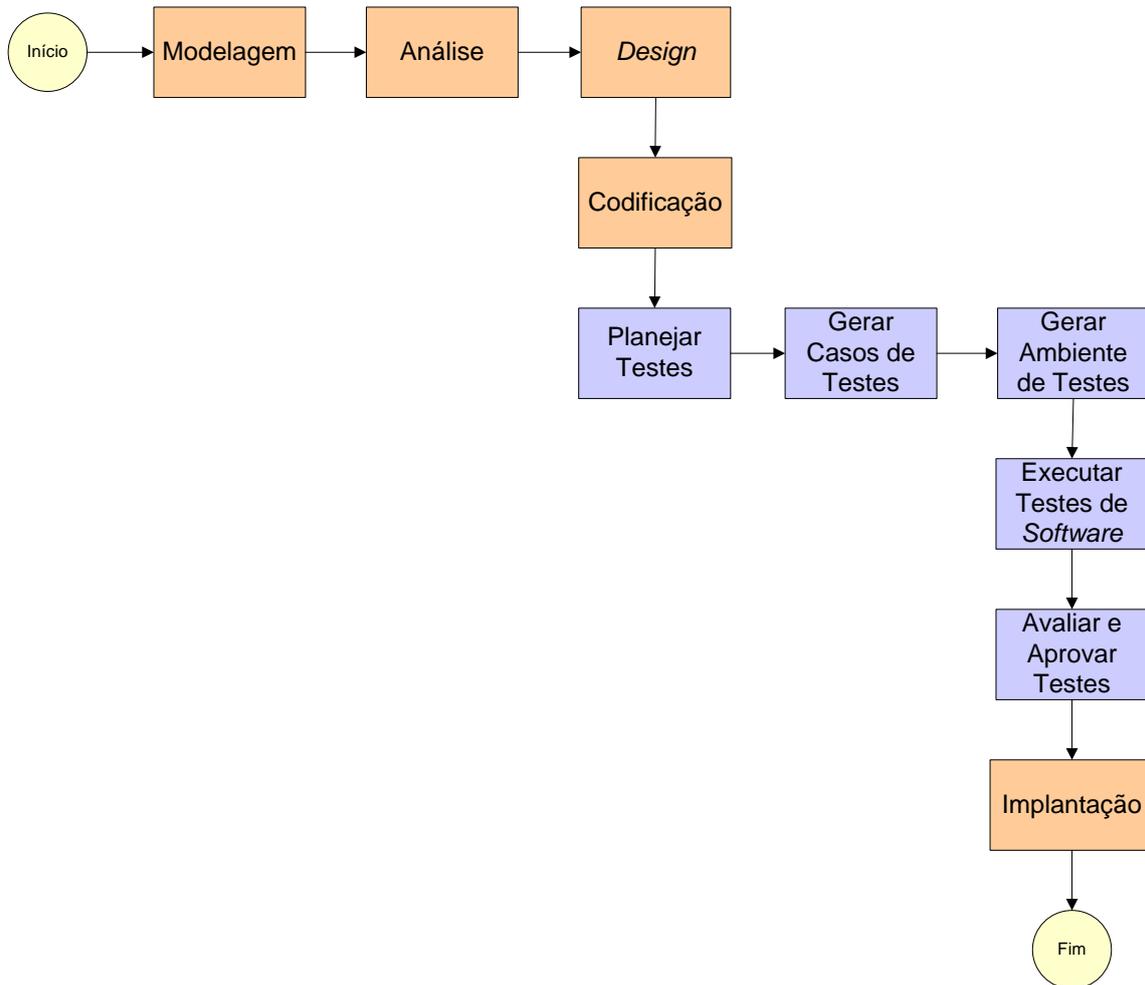


Figura 5.4 – *Framework Bean* Ciclo de Vida em Cascata com *Blackbox Bean* na Atividade de Teste

Uma vez escolhidos os *beans*, o Gerente de Processo deverá criar o Projeto Verão na *Beans Composer*. A criação deste projeto é demonstrada na figura 5.5. Em seguida será necessário criar a primeira e única etapa inserindo o *framework bean* de Ciclo de Vida em Cascata, conforme ilustrado na figura 5.6. Dentre as atividades abertas disponibilizadas, a atividade de Testes foi selecionada para ser preenchida com um *blackbox bean*. Este preenchimento está apresentado na figura 5.7.

Sair Ativar Projeto Etapa

Projeto

* Nome:	Projeto Verão
Status:	Em Elaboração
Gerente de Processo:	Vanessa T Nunes
Data:	09/01/2004
* Descrição:	O objetivo deste projeto é apresentar um exemplo de utilização dos beans na área de Engenharia de Software

* Campos Obrigatórios

Etapas do Projeto:

Etapas	Tipo	Classificação	Situação
→ 1) Ciclo de Vida em Cascata	Framework	ENGENHARIA DE SOFTWARE	Faltando Associar Beans

Figura 5.5 – Criação do Projeto Verão

Sair Consultar Etapa Preencher Atividades Abertas

Etapa

Projeto:	
Nome do Projeto:	Projeto Verão
Status do Projeto:	Em Elaboração
Gerente de Processo:	Vanessa T Nunes

Etapa:	
Número da Etapa:	1
* Bean:	Ciclo de Vida em Cascata
Classificação:	ENGENHARIA DE SOFTWARE
Tipo do Bean Selecionado:	Framework Bean
Atividades Abertas:	Análise Codificação Design Implantação Modelagem Teste

* Campos Obrigatórios

Atividades Abertas da Etapa

Atividades Abertas	Beans Associados	Situação
→ Análise		Faltando Associar Beans
→ Codificação		Faltando Associar Beans
→ Design		Faltando Associar Beans
→ Implantação		Faltando Associar Beans
→ Modelagem		Faltando Associar Beans
→ Teste		Faltando Associar Beans

Figura 5.6 – Etapa do Ciclo de Vida em Cascata

Projeto:	
Nome do Projeto:	Projeto Verão
Status do Projeto:	Em Elaboração
Gerente do Processo:	Vanessa T Nunes

Etapa:	
Número da Etapa:	1
Bean Associado:	Ciclo de Vida em Cascata

Atividade Aberta Atual:	
Atividade Aberta Atual:	Teste
* Bean Associado:	Teste
Classificação do Bean:	ENGENHARIA DE SOFTWARE
Tipo do Bean Seleccionado:	BlackBox Bean

* Campo Obrigatório

Figura 5.7 – Atividade Aberta de Teste Preenchida com *Blackbox Bean*

Uma vez combinados os *beans* desejados, o projeto foi ativado na *Beans Composer* e executado na base de Aplicação. A instância do Projeto Verão em execução está ilustrada pela figura 5.8.

Job	Activity	Participants	Job prio.	Job due	- postponed until
▼ Projeto Verão					
☐	Análise (New) - no subject (Sample Form)	Andrea Magalhaes Magdaleno, Andrea Magalhaes, Vanessa	2. Medium		

Figura 5.8 – Execução do Projeto Verão na Base de Aplicação

5.4. Aspectos de Implementação

Entre os diversos sistemas de gerenciamento de workflow disponíveis no mercado, optamos em trabalhar no Lotus Workflow, por ele dispor de todas as funcionalidades necessárias (definição e instanciação de processos, bem como o acompanhamento da execução das atividades); ser reconhecido comercialmente; e por estar disponível gratuitamente na página da empresa.

A ferramenta desenvolvida neste trabalho foi construída a partir da base de Definição de Processos do *Lotus Workflow*. Nesta base já eram disponibilizadas as informações referentes aos processos e atividades criados no *Lotus Workflow Architect*. O nosso trabalho estendeu o *Lotus Workflow* para permitir o desenvolvimento das funcionalidades de criação, adaptação e conexão de *beans* com o objetivo de gerar processos customizados que poderão ser executados na ferramenta de *workflow* e reutilizados, se necessário. A figura 5.9. ilustra esta arquitetura, destacando os itens que foram desenvolvidos. Para a criação dos recursos oferecidos, foi utilizada uma linguagem de programação nativa do *Lotus Notes* chamada *Lotus Script*.

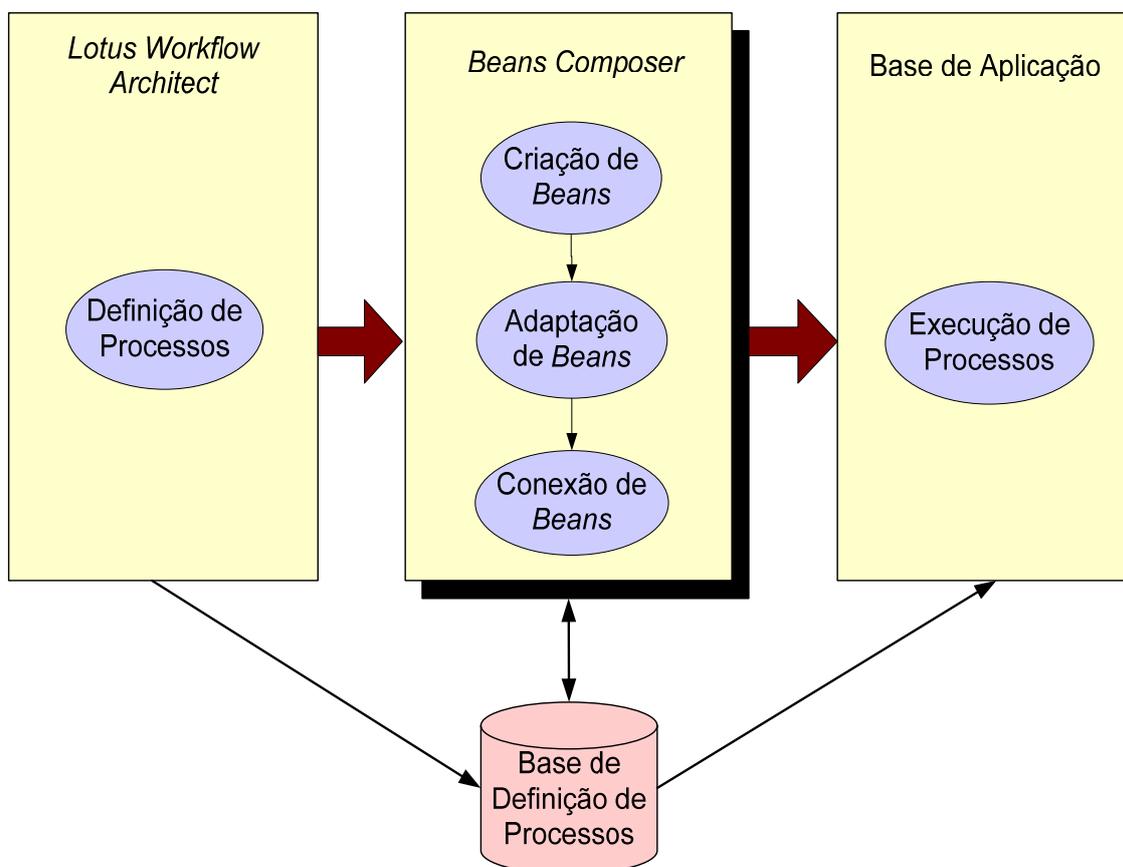


Figura 5.9 – Arquitetura *Beans Composer*

O modelo de dados ilustrado na figura 5.10 apresenta o modelo de informações final da aplicação desenvolvida. Neste diagrama, as classes de Processos e Atividades (cujos atributos estão parcialmente ilustrados) já existiam, pois faziam parte do *Lotus Workflow*. As demais classes foram criadas com o objetivo de suportar os *process beans*.

Dentre as funcionalidades descritas na seção anterior, duas foram mais complexas de serem implementadas: a ativação do projeto para que possa ser executado na máquina de *workflow* e a visualização gráfica do processo.

A ativação do projeto foi trabalhosa, pois era necessário transformar o projeto definido pelo Gerente de Processo (com todas as suas etapas e atividades abertas) em um formato possível de ser interpretado pela máquina do *Lotus Workflow*, que roda na base de aplicação. Neste ponto, o grande obstáculo encontrado era descobrir como o *Lotus Workflow* lia as informações dos

processos, criados a partir do *Lotus Workflow Architect*, e armazenados nas bases de Definição de Processos e de Repositório de Processos, para que nós pudéssemos simular o mesmo comportamento.

Esta etapa foi especialmente complexa porque o *Lotus Workflow* não oferece uma documentação referente ao seu meta-modelo. Assim, foi necessário investigar o significado de cada campo para aprendermos o momento certo de utilizá-lo. Esta estratégia provocou um forte acoplamento da ferramenta desenvolvida com a versão do *Lotus Workflow* utilizada, sem que se possa garantir de compatibilidade com versões futuras.

Além disso, a máquina de *workflow* do *Lotus Workflow* é composta por códigos pré-compilados em fase de desenvolvimento e armazenados em agentes de caixa-preta. Desta forma, ao tentar debugar a execução de um destes códigos, somente é possível observar as suas entradas e saídas, sem que se consiga visualizar o que ocorre ali dentro.

A outra dificuldade encontrada foi a visualização gráfica do mapa do processo. Para oferecer esta funcionalidade, inicialmente imaginávamos utilizar o *Lotus Workflow Viewer*. Entretanto, esta ferramenta encontra-se disponível apenas na base de Aplicação, porque as informações por ela apresentadas estão relacionadas à instância em execução de um determinado processo. Por isso, o *Lotus Workflow Viewer* precisa de informações sobre a atividade que está sendo executada atualmente, quais atividades já foram concluídas e os documentos que já foram criados. Dentro da *Beans Composer*, não existem estas informações pois os processos ainda não foram instanciados, uma vez que se encontram em fase de definição. Sendo assim, foi necessário abandonar esta idéia e procurar alternativas de solução.

Finalmente, a solução encontrada foi utilizar o *Lotus Workflow Architect* através da manipulação de um arquivo XML (*Extensible Markup Language*). Esta solução foi conseguida ao descobirmos que esta ferramenta permite importações e exportações das definições dos processos para o formato XML. Um exemplo do formato do arquivo XML utilizado pelo *Lotus Workflow Architect* foi disponibilizado no Anexo XIII.

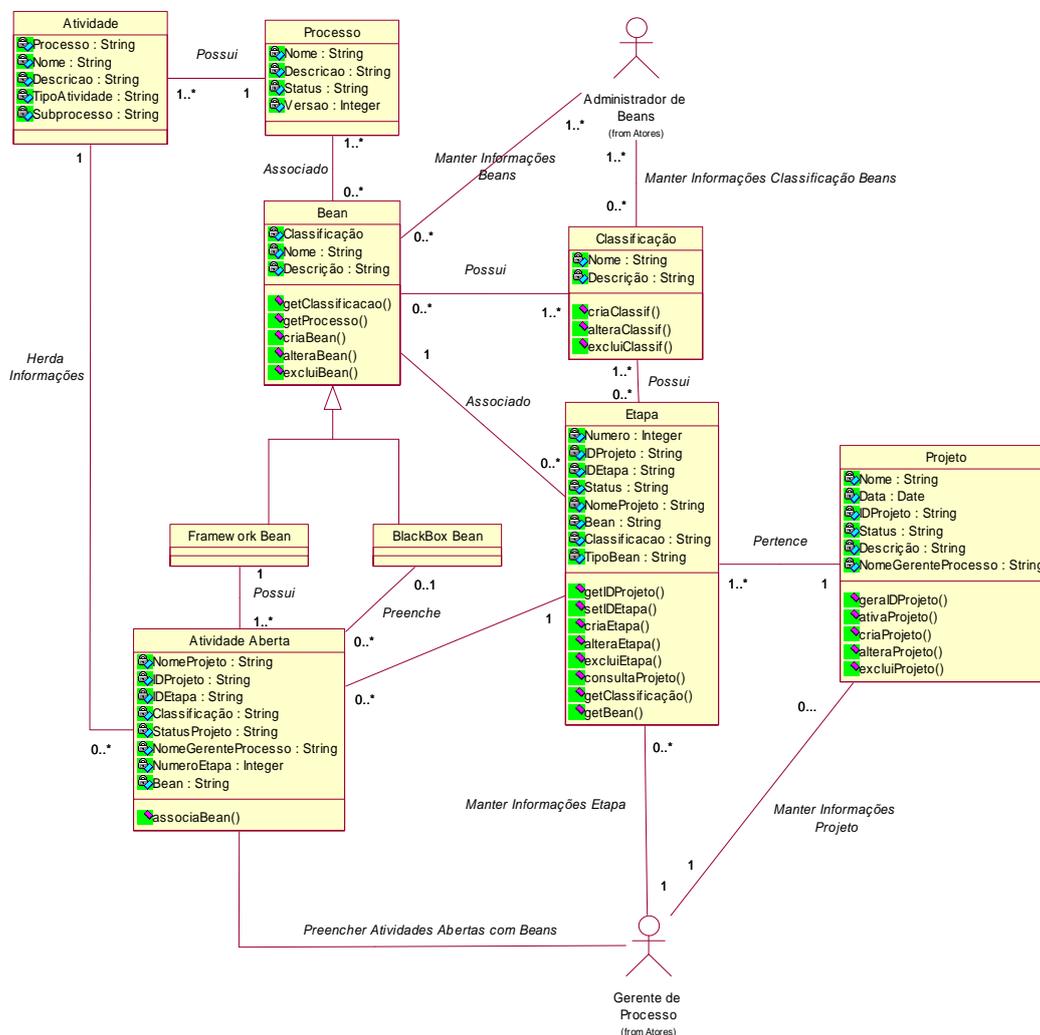


Figura 5.10 – Diagrama de Classes da *Beans Composer*

A partir deste ponto, nós estudamos as tags utilizadas neste padrão de código XML para interpretar o seu significado. Com este aprendizado fomos capazes de gerar um arquivo XML no mesmo formato. Este arquivo é gerado ao acionar a opção “ Visualizar Processo no *Architect*” , a partir daí o *Lotus Workflow Architect* é inicializado abrindo o arquivo recém-criado. Para tanto é necessário que o Gerente de Processo tenha instalado em sua máquina o *Lotus Workflow Architect*.

6. Conclusão

Neste contexto onde as organizações estão interessadas em modelar os seus processos, o principal objetivo deste trabalho foi especificar e implementar uma ferramenta que oferecesse a possibilidade de manipular *process beans* referentes a qualquer domínio de aplicação de forma a adaptar os processos da organização, complementando os trabalhos anteriores que apresentavam a idéia dos *process beans*. Para isso, foi criada a *Beans Composer*, a partir da ferramenta *Lotus Workflow*.

Esta ferramenta será particularmente benéfica em organizações onde já exista a iniciativa de modelagem de processos. Acreditamos que os *process beans* serão de grande auxílio na definição dos processos, uma vez que evitam o trabalho de redefinir processos já definidos anteriormente, além de permitir a adaptação e flexibilização de processos.

Podemos destacar como benefícios alcançados pela realização desse trabalho: a aquisição do conhecimento na tecnologia XML (*Extensible Markup Language*) necessário neste trabalho para a visualização gráfica dos processos, o aprendizado de uma ferramenta comercial de *workflow* (*Lotus Workflow*) e o aprofundamento da experiência de desenvolvimento na plataforma *Lotus Notes*.

Como trabalhos futuros poderiam ser tratadas as seguintes questões:

- Para auxiliar as organizações na definição da sua biblioteca de beans, poderia ser oferecido um conjunto de diretrizes para orientar o Administrador de *Beans* na definição e classificação dos *process beans*;
- Oferecer recursos para localização e recuperação dos *process beans* armazenados na biblioteca;
- Desenvolver uma funcionalidade que permita o aproveitamento de um projeto já definido na elaboração de novos *process beans*;
- Permitir que o Gerente de Processo possa visualizar graficamente todo o fluxo das atividades que compõem as etapas do projeto;

- Incluir na ferramenta recursos para o registro da avaliação do projeto. O Gerente de Processo deveria poder destacar aspectos positivos ou negativos detectados durante a execução do projeto de forma a criar uma base de conhecimento que auxilie na definição de novos projetos.

Além disso, poderiam ser oferecidos recursos, utilizando a tecnologia de *groupware*, para a divulgação da publicação de novos projetos;

- Realização de estudos de caso utilizando a ferramenta desenvolvida - como não pôde ser realizado durante a elaboração deste trabalho, sugerimos que seja realizada a experimentação da *Beans Composer* com o objetivo de avaliar o seu uso prático.

Referências

AMBRIOLA, V.; CONRADI R.; FUGGETTA A. **Assessing Process-Centered Software Engineering Environments**, ACM Transactions on Software Engineering and Methodology, v.6, n.3, 1997.

ARAUJO, R.; BORGES.M. **Sistemas de Workflow**. In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA (JAI), Congresso da SBC, 2001, Fortaleza, Ceará. p. 1 - 17.

ARAUJO, R.; BORGES.M. **Sobre a Aplicabilidade de Sistemas Workflow no Suporte a Processos de Software**. In: 2º WORKSHOP IBEROAMERICANO DE INGENIERÍA DE REQUISITOS Y AMBIENTES SOFTWARE, 1999, San Jose, Costa Rica.

ARAUJO, R.M. **Ampliando a cultura de Processos de Software – Um enfoque Baseado em Groupware e Workflow**. 2000. Tese de D.Sc., COPPE/UFRJ, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

CHAFFEY, D. **Groupware, Workflow and Intranets: Reengineering the Enterprise with Collaborative Software**. 1ed. Butterworth-Heinemann, EUA. 1998.

BARBIERI, C. **BI – Business Intelligence – Modelagem e Tecnologia**. 1 ed. Axcel Books. Rio de Janeiro, Brasil. 2001.

BORGES, M.R.S.; ARAUJO, R.M.; Sobrinho, B.C. **Bridging the gap between organizations and their software processes – An approach based on patterns and workflow systems**. In: CONGRESSO DO APSEC. 2000.

BOSATO, A.; REZENDE, E.; BITTES, J.M. **Sistemas de Workflow Apoiando Processos de Engenharia de Software**. 2002. Monografia da disciplina de Tópicos Especiais em Sistemas de Informação do Programa de Mestrado NCE/IM, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

CHORD – Disponível em: <http://chord.nce.ufrj.br> – Acesso em: 26 out 2003

CHRISTIE, A.M. **Software Process Automation**. 1995. 1 ed. Berlin Heidelberg, Springer-Verlag.

CRUZ, T., **e-Workflow – Como Implantar e Aumentar a Produtividade de Qualquer Processo**. Cenadem. São Paulo, Brasil. 2001.

CRUZ, T., **Workflow – A Tecnologia que vai Revolucionar Processos**. Atlas. São Paulo, Brasil. 2000.

COUTINHO, R.S.; GOMES, T.A.C. – **Facilitando a Melhoria de Processos de Negócio através da utilização de ferramentas de Workflow**. 2002. Monografia da disciplina de Tópicos Especiais em Sistemas de Informação do Programa de Mestrado NCE/IM, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

DICATERINO, A.; LARSEN, K.; TANG, M. *et al.* **An Introduction to Workflow Management Systems**. 1997.

E-WORKFLOW – **The Workflow Portal**. Disponível em: <http://www.e-workflow.org>. Acesso em: 10 mar 2003.

GEORGAKOPOULOS, D.; TSALGATIDOU, A. **Technology and Tools for Comprehensive Business Process Lifecycle Management**. In: Proceedings of the NATO Advanced Study Institute on Workflow. 1997. Vol 164. p. 356-395. Istambul, Turquia.

HUMPHREY, W.S. **Managing the Software Process**. Massachussets: Addison-Wesley. 1990.

IENDRIKE, H.S. **Método para Projeto de Workflow a partir do Modelo de Negócio das Organizações**. 2003. Projeto Final – Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

KHOSHAFIAN, S.; BUCKIEWICZ, M. **Introduction to Groupware, Workflow and Workgroup Computing**. 1ed. John Wiley & Sons. EUA. 1995.

KNIGHT, D.M.; ROSA, M.G.P.; LEAL, R.P. **Abordagem de Workflow no Suporte a Processos de**

Software. 2002. Monografia da disciplina de Tópicos Especiais em Sistemas de Informação do Programa de Mestrado NCE/IM, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

KOBIELUS, J. **Workflow Strategies.** IDG Books. Nova York, EUA. 1997.

LOTUS – **IBM/Lotus Developer** - Disponível em: <http://www.notes.net> - Acesso em: 15 ago 2003.

MEDEIROS, M.A.; LEITE, P.H.P.M.; JUNIOR, R.F. **Ampliando a Percepção de Processos em Sistemas de Workflow.** 2001. Projeto Final – Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

MENDES, C.D.L. **Suporte às Atividades de Pós Reunião Através da Utilização da Tecnologia de Workflow.** 2003. Tese de Mestrado – NCE/Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

PARRINI, E.; CAMPOS, M.L.; BORGES, M.R.S. - **AMPA: Knowledge Management in Analytical-based Decision Processes.** 1999. Tese de M.Sc., NCE/Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

PLESUMS, C. **Workflow Handbook 2002.** Computer Sciences Corporation, Financial Services Groups. Ed: Layna Fischer. Cap.1.

PRESSMAN, R.S. **Software Engineering – A Practitioner' s Approach.** 5 ed. McGraw Hill. Nova York, EUA. 1997.

PRIOR, C. **Workflow Handbook 2003.** Australia: Layna Fischer. Cap.1. Disponível em: <http://www.wfmc.org/information/handbook2003.htm> - Acesso em: 18 set 2003.

RUP, **Rational Unified Process 5.0,** Rational Software Corporation, 1998.

SANTOS, R.P.C. **Engenharia de Processos – Conceitos e Práticas.** 2001. Projeto Final – Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

SILVA, A.V., **Modelagem de Processos para Implementação de Workflow: Uma Avaliação Crítica**. 2001. Tese de M.Sc., COPPE/UFRJ, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

SOBRINHO, B.C. **Process Beans: definindo processos de software em um sistema de gerenciamento de workflow**. 2000. Projeto Final – Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

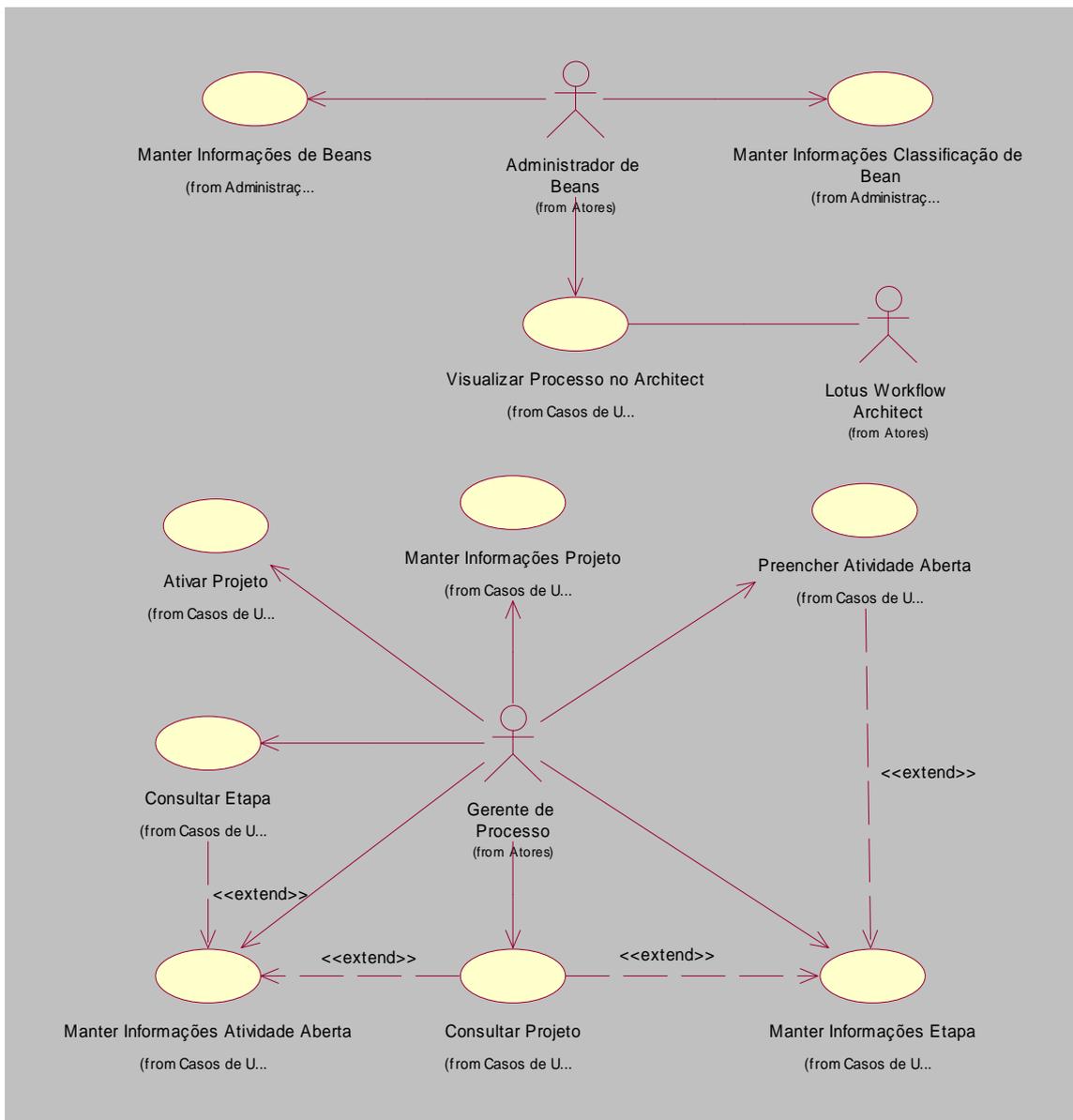
VILLAS, M.; FLEURY, N. **O Workflow como Suporte ao Trabalho Cooperativo** – Developer's Magazine. Ano II. n. 20. Abril 1998.

WARIA – **Workflow and Reengineering International Association**. Disponível em: <http://www.waria.com>. Acesso em: 20 fev 2003.

WFMC – **Workflow Management Coalition**. Disponível em: <http://www.wfmc.org>. Acesso em: 20 fev 2003.

Anexos

Anexo I – Diagrama de Casos de Uso



Anexo II – Caso de Uso Manter Informações de Classificação de *Bean*

Descrição Sumária

Este caso de uso descreve o processo de manutenção de informações das classificações de *beans*.

Atores

Administrador de *Beans*

Fluxo Principal

1. *Este caso de uso se inicia quando o Administrador de Beans entra na visão de Classificação de Beans.*
2. *Caso o Administrador de Beans clique no botão Criar Classificação de Bean, inicia-se o sub-fluxo Criar Classificação de Bean.(SF1)*
3. *Caso o Administrador de Beans clique na opção Alterar Classificação de Bean, inicia-se o sub-fluxo Alterar Classificação de Bean. (SF2)*
4. *Caso o Administrador de Beans clique na opção Excluir Classificação de Bean, inicia-se o sub-fluxo Excluir Classificação de Bean. (SF3)*
5. *Quando o Administrador de Beans seleciona outra visão ou fecha o sistema, este caso de uso termina.*

Sub-Fluxos (SF)

SF1. Criar Classificação de Bean

1. *O sistema exibe uma tela com o formulário de classificação de bean e os campos Classificação e Descrição em branco.*
2. *O Administrador de Beans informa a classificação e a descrição.*
3. *O Administrador de Beans clica no botão Salvar. (FA1, FA4)*
4. *O sistema grava a nova classificação de bean.*
5. *O Administrador de Beans clica no botão Sair.*
6. *O sistema fecha a tela de classificação de bean.*
7. *O caso de uso retorna ao fluxo principal no passo 1.*

SF2. Alterar Classificação de *Bean*

1. *O Administrador de Beans escolhe a classificação de bean que deseja alterar. (FA2, FA3, FA5)*
2. *O sistema exibe uma tela com a classificação do bean escolhida aberta em modo de edição, para que o Administrador de Beans possa realizar as alterações desejadas.*
3. *O Administrador de Beans informa a nova classificação ou a nova descrição.*
4. *O Administrador de Beans clica na opção Salvar. (FA1, FA4)*
5. *O sistema salva as alterações na classificação de bean.*
6. *O Administrador de Beans clica no botão Sair.*
7. *O sistema fecha a tela de classificação de bean.*
8. *O caso de uso retorna ao fluxo principal no passo 1*

SF3. Excluir Classificação de *Bean*

1. *O Administrador de Beans seleciona a(s) classificação(ões) que deseja excluir. (FA2)*
2. *O sistema exibe uma mensagem confirmando a ação de exclusão.*
3. *O Administrador de Beans confirma a ação.(FA5)*
4. *O sistema exclui a(s) classificação(ões) selecionada(s).*
5. *O sistema exibe uma mensagem confirmando o sucesso da operação.*
6. *O caso de uso retorna ao fluxo principal no passo 1*

Fluxos Alternativos (FA)

FA1. Campo Obrigatório Não Preenchido

1. *O Administrador de Beans não preenche o campo Classificação e/ou o campo Descrição.*
2. *O sistema exibe uma mensagem solicitando o preenchimento do campo para que possa salvar o documento.*
3. *O caso de uso retorna ao sub-fluxo Criar Classificação de Bean no passo 1 ou ao sub-fluxo Alterar Classificação de Bean no passo 2.*

FA2. Nenhuma Classificação Selecionada

1. *O Administrador de Beans não seleciona nenhuma classificação na visão*
2. *O sistema exibe uma mensagem solicitando a seleção de pelo menos uma classificação para que possa realizar a operação.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

FA3. Mais de 1 Classificação Seleccionada

1. *O Administrador de Beans seleciona mais de uma classificação para ser alterada.*
2. *O sistema exibe uma mensagem solicitando a seleção de apenas uma classificação.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

FA4. Classificação de Bean Já Cadastrada

1. *O Administrador de Beans digita uma classificação de bean que já se encontra cadastrada.*
2. *O sistema exibe uma mensagem de alerta e não o inclui novamente na lista.*
3. *O caso de uso retorna ao sub-fluxo Criar Classificação de Bean no passo 1 ou ao sub-fluxo Alterar Classificação de Bean no passo 3.*

FA5. Classificação com Beans Já Associados

1. *O Administrador de Beans escolhe alterar/excluir uma classificação que já se encontra associada com um ou mais beans.*
2. *O sistema detecta a associação e informa que não poderá realizar a operação.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

Pré-Condições

Nenhuma

Pós-Condições

1. *Lista de classificação de beans atualizada*

Anexo III – Caso de Uso Manter Informações de *Beans*

Descrição Sumária

Este caso de uso descreve o processo de manutenção de informações de *beans*.

Atores

Administrador de *Beans*

Fluxo Principal

1. *Este caso de uso se inicia quando o Administrador de Beans entra em uma das visões da Biblioteca de Beans.*
2. *Caso o Administrador de Beans clique no botão Criar Bean, inicia-se o sub-fluxo Criar Bean.(SF1)*
3. *Caso o Administrador de Beans clique no botão Alterar Bean, inicia-se o sub-fluxo Alterar Bean. (SF2)*
4. *Caso o Administrador de Beans clique no botão Excluir Bean, inicia-se o sub-fluxo Excluir Bean. (SF3)*
5. *Quando o Administrador de Beans seleciona outra visão ou fecha o sistema, este caso de uso termina.*

Sub-Fluxos (SF)

SF1. Criar *Bean*

1. *O sistema exibe uma tela com o formulário de bean e os campos Nome, Processo Associado, Tipo, Classificação, Objetivos, Contexto e Solução em branco.*
2. *O Administrador de Beans informa o nome do bean que está sendo criado.*
3. *O Administrador de Beans seleciona um processo na lista dos processos existentes na Biblioteca de Processos.*
4. *O sistema recupera a lista das atividades que compõem o processo selecionado a partir da biblioteca de processos.*
5. *O sistema recupera o ID do processo da base de repositório.*
6. *O Administrador de Beans determina o tipo do bean: BlackBox Bean ou Framework Bean.*
7. *O Administrador de Beans seleciona uma ou mais classificações na lista.*

8. *Caso o tipo do bean selecionado tenha sido Framework Bean:*
 - a. *O sistema exibe o campo “ Atividades Abertas” .*
 - b. *O Administrador de Beans informa qual(is) atividade(s) ficará(ão) aberta(s) no novo bean.*
9. *O Administrador de Beans informa os objetivos, contexto e solução para o bean que está sendo criado.*
10. *O Administrador de Beans clica no botão Salvar. (FA1, FA2, FA6)*
11. *O sistema grava as informações relativas ao novo bean que foi criado.*
12. *O Administrador de Beans clica no botão Sair*
13. *O sistema fecha a tela de bean.*
14. *O caso de uso retorna ao fluxo principal no passo 1.*

SF2. Alterar Bean

1. *O Administrador de Beans escolhe o bean que deseja alterar. (FA3, FA4, FA5)*
2. *O sistema exibe uma tela com o bean escolhido aberto em modo de edição.*
3. *O Administrador de Beans realiza as alterações desejadas.*
4. *O Administrador de Beans clica no botão Salvar. (FA1, FA2, FA6)*
5. *O sistema salva as alterações no bean.*
6. *O Administrador de Beans clica no botão Sair*
7. *O sistema fecha a tela de bean.*
8. *O caso de uso retorna ao fluxo principal no passo 1.*

SF3. Excluir Bean

1. *O Administrador de Beans seleciona o(s) bean(s) que deseja excluir. (FA3, FA5)*
2. *O sistema exibe uma mensagem confirmando a ação de exclusão.*
3. *O Administrador de Beans confirma a ação. (FA7)*
4. *O sistema exclui o(s) bean(s) selecionado(s).*
5. *O sistema exibe uma mensagem confirmando o sucesso da operação.*
6. *O caso de uso retorna ao fluxo principal no passo 1.*

Fluxos Alternativos (FA)

FA1. Campo Obrigatório Não Preenchido

1. *O sistema exibe uma mensagem solicitando o preenchimento do(s) campo(s) que foi(ram)*

deixado(s) em branco.

- 2. O sistema não grava o novo bean.*
- 3. O caso de uso retorna ao sub-fluxo Criar Bean no passo 2 ou ao sub-fluxo Alterar Bean no passo 3.*

FA2. Nome de *Bean* Já Cadastrado

- 1. O sistema exibe uma mensagem informando que já existe outro bean cadastrado com o mesmo nome e solicitando um novo nome para o bean que está sendo criado.*
- 2. O sistema não grava o novo bean.*
- 3. O caso de uso retorna ao sub-fluxo Criar Bean no passo 2 ou ao sub-fluxo Alterar Bean no passo 3.*

FA3. Nenhum *Bean* Selecionado

- 1. O Administrador de Beans não seleciona nenhum bean na visão.*
- 2. O sistema exibe uma mensagem solicitando a seleção de pelo menos um bean para que possa realizar a operação.*
- 3. O caso de uso retorna ao fluxo principal no passo 1.*

FA4. Mais de 1 *Bean* Selecionado

- 1. O Administrador de Beans seleciona mais de um bean para ser alterado.*
- 2. O sistema exibe uma mensagem solicitando a seleção de apenas um bean.*
- 3. O caso de uso retorna ao fluxo principal no passo 1.*

FA5. *Bean* com Etapa Já Associada

- 1. O Administrador de Beans seleciona um bean que já está associado com alguma etapa.*
- 2. O sistema detecta a associação e exibe uma mensagem alertando que não foi possível realizar a operação.*
- 3. O caso de uso retorna ao fluxo principal no passo 1.*

FA6. *Framework Bean* sem Atividade Aberta

- 1. O Administrador de Beans cadastra um framework bean sem informar as suas atividades abertas.*
- 2. O sistema exibe uma mensagem solicitando a lista das atividades abertas daquele bean.*
- 3. O caso de uso retorna ao fluxo principal no passo 1.*

FA7. Cancelamento de Operação

1. *O Administrador de Beans cancela a operação de exclusão.*
2. *O sistema não exclui os documentos.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

Pré-Condições

1. *Deve existir pelo menos uma classificação de bean cadastrada para que o bean que está sendo criado seja classificado.*
2. *Devem existir processos na Biblioteca de Processos.*
3. *Não pode existir nenhum outro bean com o mesmo nome.*

Pós-Condições

1. *Biblioteca de Beans atualizada.*

Anexo IV – Caso de Uso Manter Informações de Projeto

Descrição Sumária

Este caso de uso descreve o processo de manutenção de informações de projetos.

Atores

Gerente de Processo

Fluxo Principal

1. *Este caso de uso se inicia quando o Gerente de Processo entra em alguma das visões da Biblioteca de Projetos.*
2. *Caso o Gerente de Processo clique no botão Criar Projeto, inicia-se o sub-fluxo Criar Projeto. (SF1)*
3. *Caso o Gerente de Processo clique no botão Alterar Projeto, inicia-se o sub-fluxo Alterar Projeto. (SF2)*
4. *Caso o Gerente de Processo clique no botão Excluir Projeto, inicia-se o sub-fluxo Excluir Projeto. (SF3)*
5. *Quando o Gerente de Processo seleciona outra visão ou fecha o sistema, este caso de uso termina.*

Sub-Fluxos (SF)

SF1. Criar Projeto

1. *O sistema exibe uma tela com o formulário de projeto e os campos Nome, Status, Gerente de Processo, Data e Descrição. Os campos Status, Gerente de Processo e Data já são preenchidos automaticamente pelo sistema e apenas os campos Nome e Descrição encontram-se em branco para serem digitados.*
2. *O Gerente de Processo informa o nome e a descrição do projeto que está sendo criado.*
3. *O Gerente de Processo aciona a opção “ Salvar” . (FA1, FA6).*
4. *O sistema grava as informações relativas ao novo projeto que foi criado.*
5. *O sistema habilita o botão Criar Etapa.*
6. *O sistema desabilita a edição dos campos.*
7. *Caso o Gerente de Processo acione a opção Criar Etapa:*
 - a. *O sistema executa o caso de uso Manter Informações de Etapa.*

8. *O Gerente de Processo aciona a opção Sair.*
9. *O sistema fecha o formulário de projeto.*
10. *O caso de uso retorna ao fluxo principal no passo 1.*

SF2. Alterar Projeto

1. *O Gerente de Processo escolhe o projeto que deseja alterar. (FA2, FA3, FA5, FA8).*
2. *O sistema apresenta uma caixa de diálogo com o nome e a descrição atual do projeto para que o Gerente de Processo possa fazer as alterações desejadas.*
3. *O Gerente de Processo informa os novos dados e clica na opção OK. (FA1, FA4, FA6, FA7)*
4. *O sistema fecha a caixa de diálogo.*
5. *Caso o nome do projeto tenha sido alterado, o sistema atualiza o nome do projeto na(s) etapa(s) e atividade(s) aberta(s) associada(s).*
6. *O sistema salva as alterações no projeto.*
7. *O caso de uso retorna ao fluxo principal no passo 1.*

SF3. Excluir Projeto

1. *O Gerente de Processo seleciona o(s) projeto(s) que se deseja excluir. (FA2, FA5, FA8).*
2. *Para o(s) projeto(s) que possui(em) etapa(s) ou atividade(s) aberta(s) associada(s), o sistema pergunta se o Gerente de Processo deseja cancelar a operação ou excluir todos os documentos associados.*
3. *O Gerente de Processo confirma a exclusão de todos os documentos associados. (FA4)*
4. *O sistema exclui o(s) projeto(s) selecionado(s) e a(s) sua(s) etapa(s) e atividade(s) aberta(s).*
5. *O sistema exibe uma mensagem confirmando o sucesso da operação.*
6. *O caso de uso retorna ao fluxo principal no passo 1.*

Fluxos Alternativos (FA)

FA1. Campo Obrigatório Não Preenchido

1. *O sistema exibe uma mensagem solicitando o preenchimento do campo que foi deixado em branco.*
2. *O sistema não grava o novo projeto até que o nome seja informado.*
3. *O caso de uso retorna ao sub-fluxo Criar Projeto no passo 2 ou ao sub-fluxo Alterar Projeto no passo 2.*

FA2. Nenhum Projeto Selecionado

1. *O Gerente de Processo não seleciona nenhum projeto na visão.*
2. *O sistema exibe uma mensagem solicitando que pelo menos um projeto seja selecionado.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

FA3. Mais de 1 Projeto Selecionado

1. *O Gerente de Processo seleciona mais de um projeto para ser alterado.*
2. *O sistema exibe uma mensagem solicitando a seleção de apenas um projeto.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

FA4. Cancelamento de Operação

1. *O Gerente de Processo escolhe cancelar a operação para não excluir/alterar os documentos.*
2. *O sistema não exclui/altera o projeto e nem as etapas ou atividades abertas associadas.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

FA5. Alteração/Exclusão por Outro Gerente de Processo

1. *O Gerente de Processo que está tentando alterar ou excluir não é o mesmo que criou o projeto.*
2. *O sistema exibe uma mensagem informando que o Gerente de Processo atual não está autorizado a alterar ou excluir aquele projeto.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

FA6. Nome do Projeto Já existente

1. *O sistema exibe uma mensagem alertando que já existe outro projeto cadastrado com o mesmo nome e solicitando outro nome.*
2. *O sistema não cria outro projeto.*
3. *O caso de uso retorna ao sub-fluxo Criar Projeto no passo 2 ou ao sub-fluxo Alterar Projeto no passo 2.*

FA7. Nome do Projeto Igual ao Anterior

1. *O sistema exibe uma mensagem alertando que o nome informado era exatamente igual ao anterior e solicitando o novo nome.*
2. *O sistema não realiza a alteração.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

FA8. Projeto Ativo

1. *O sistema exibe uma mensagem alertando que o projeto que se deseja alterar ou excluir já está ativo e não poderá ser alterado.*
2. *O sistema não realiza a operação.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

Pré-Condições

1. *Apenas o Gerente de Processo que criou o projeto pode alterá-lo ou excluí-lo.*
2. *Para alterar ou excluir um projeto ele não pode estar ativo.*

Pós-Condições

1. *Informações de projetos atualizadas.*
2. *Informações de etapas atualizadas.*
3. *Informações de atividades abertas atualizadas.*

Anexo V – Caso de Uso Manter Informações de Etapa

Descrição Sumária

Este caso de uso descreve o processo de manutenção de informações de etapas nos projetos.

Atores

Gerente de Processo

Fluxo Principal

1. *Este caso de uso se inicia quando o Gerente de Processo entra em um dos seus projetos ou em uma das suas etapas.*
2. *O sistema exibe os botões Criar Etapa, Alterar Etapa e Excluir Etapa em cascata apenas para o Gerente de Processo que criou aquele projeto.*
3. *Caso o Gerente de Processo selecione a opção Criar Etapa, inicia-se o sub-fluxo Criar Etapa.(SF1)*
4. *Caso o Gerente de Processo selecione a opção Alterar Etapa dentro do formulário de Projeto, inicia-se o sub-fluxo Alterar Etapa pelo Projeto.(SF2)*
5. *Caso o Gerente de Processo selecione a opção Excluir Etapa dentro do formulário de Projeto, inicia-se o sub-fluxo Excluir Etapa pelo Projeto.(SF3)*
6. *Caso o Gerente de Processo selecione a opção Alterar Etapa dentro do formulário de Etapa, inicia-se o sub-fluxo Alterar Etapa.(SF4)*
7. *Caso o Gerente de Processo selecione a opção Excluir Etapa dentro do formulário de Etapa, inicia-se o sub-fluxo Excluir Etapa.(SF5)*
8. *O Gerente de Processo clica no botão Sair*
9. *O sistema fecha o formulário.*
10. *Este caso de uso termina.*

Sub-Fluxos (SF)

SF1. Criar Etapa

1. *O sistema exibe uma tela composta de: informações relativas ao projeto (Nome do Projeto, Status do Projeto e Gerente de Processo) já preenchidas pelo sistema e disponíveis apenas para leitura; o Número da Etapa atual também já preenchido; o campo Bean em branco e disponível para edição; as opções Salvar, Sair ou Consultar Projeto.*

2. *O Gerente de Processo seleciona um bean na lista de opções disponíveis.*
3. *O sistema exibe os campos Classificação e Tipo Bean já preenchidos de acordo com a escolha realizada no passo anterior.*
4. *Caso o Tipo do Bean selecionado seja Framework Bean:*
 - a. *O sistema exibe a lista das atividades que se encontram abertas naquele bean.*
5. *O Gerente de Processo aciona a opção Salvar. (FA1)*
6. *O sistema grava as informações relativas à nova etapa.*
7. *O sistema habilita as opções Criar Etapa, Alterar Etapa e Excluir Etapa.*
8. *Caso seja um framework bean:*
 - a. *O sistema gera as atividades abertas e habilita a opção Preencher Atividades Abertas.*
 - b. *Caso o Gerente de Processo selecione a opção Preencher Atividades Abertas:*
 - i. *O sistema executa o caso de uso “ Preencher Atividades Abertas” .*
9. *Caso o Gerente de Processo selecione a opção Consultar Projeto:*
 - a. *O sistema executa o caso de uso “ Consultar Projeto” .*
10. *O Gerente de Processo seleciona a opção sair.*
11. *O sistema fecha o formulário de etapa.*
12. *O caso de uso retorna ao fluxo principal no passo 2.*

SF2. Alterar Etapa pelo Projeto

1. *O sistema exibe a lista das etapas do projeto.*
2. *O Gerente de Processo seleciona a etapa que deseja alterar e clica na opção OK (FA2).*
3. *O sistema exibe uma caixa de diálogo com as informações da etapa atual (Bean Associado, Classificação do Bean e Tipo do Bean) e solicitando a escolha do novo bean.*
4. *O Gerente de Processo seleciona o bean desejado e clica na opção OK (FA2).*
5. *O sistema preenche os campos Classificação e Tipo do Bean, de acordo com a escolha no passo anterior.*
6. *O Gerente de Processo confere as informações clica na opção OK (FA1, FA2, FA3).*
7. *Se o bean anterior era Blackbox Bean e o novo bean for Framework Bean:*
 - a. *O sistema gera as atividades abertas para o novo bean.*
 - b. *O sistema exibe uma mensagem alertando ao Gerente de Processo que posteriormente será necessário preencher as atividades abertas.*
8. *Se o bean anterior era Framework Bean e o novo bean for Blackbox Bean:*
 - a. *O sistema exclui as atividades abertas existentes anteriormente.*
9. *Se o bean anterior era Framework Bean e o novo bean for Framework Bean:*

- a. *O sistema exclui as atividades abertas existentes anteriormente.*
 - b. *O sistema gera as atividades abertas para o novo bean.*
 - c. *O sistema exibe uma mensagem alertando ao Gerente de Processo que posteriormente será necessário preencher as atividades abertas.*
10. *O sistema atualiza as informações na etapa que está sendo alterada e fecha a tela.*
 11. *O caso de uso retorna ao fluxo principal no passo 2.*

SF3. Excluir Etapa pelo Projeto

1. *O sistema exibe a lista das etapas do projeto*
2. *O Gerente de Processo seleciona uma ou mais etapas para excluir e clica na opção OK (FA2).*
3. *O sistema exibe uma mensagem alertando que as atividades associadas com as etapas também serão excluídas e perguntando se o Gerente de Processo realmente deseja continuar.*
4. *O Gerente de Processo clica na opção OK. (FA2).*
5. *O sistema exclui todas as atividades abertas associadas às etapas.*
6. *O sistema corrige a numeração das etapas que restaram de acordo com os números das etapas excluídas.*
7. *O sistema exibe uma mensagem avisando que a operação foi realizada com sucesso.*
8. *O Gerente de Processo clica na opção OK.*
9. *O caso de uso retorna ao fluxo principal no passo 2.*

SF4. Alterar Etapa

1. *O sistema exibe uma caixa de diálogo solicitando a escolha do novo bean.*
2. *O Gerente de Processo seleciona o bean desejado e clica na opção OK (FA2).*
3. *O sistema preenche os campos Classificação e Tipo do Bean, de acordo com a escolha no passo anterior.*
4. *O Gerente de Processo confere as informações clica na opção OK (FA1, FA2, FA3).*
5. *Se o bean anterior era Blackbox Bean e o novo bean for Framework Bean:*
 - a. *O sistema gera as atividades abertas para o novo bean.*
 - b. *O sistema exibe uma mensagem alertando ao Gerente de Processo que posteriormente será necessário preencher as atividades abertas.*
6. *Se o bean anterior era Framework Bean e o novo bean for Blackbox Bean:*
 - a. *O sistema exclui as atividades abertas existentes anteriormente.*
7. *Se o bean anterior era Framework Bean e o novo bean for Framework Bean:*
 - a. *O sistema exclui as atividades abertas existentes anteriormente.*

- b. O sistema gera as atividades abertas para o novo bean.*
 - c. O sistema exibe uma mensagem alertando ao Gerente de Processo que posteriormente será necessário preencher as atividades abertas.*
- 8. O sistema atualiza as informações na etapa.*
 - 9. O caso de uso retorna ao fluxo principal no passo 2.*

SF5. Excluir Etapa

- 1. O sistema exibe uma mensagem alertando que as atividades associadas com as etapas também serão excluídas e perguntando se o Gerente de Processo realmente deseja continuar.*
- 2. O Gerente de Processo clica na opção OK. (FA2).*
- 3. O sistema exclui todas as atividades abertas associadas às etapas.*
- 4. O sistema corrige a numeração das etapas que restaram.*
- 5. O sistema exibe uma mensagem avisando que a operação foi realizada com sucesso.*
- 6. O Gerente de Processo clica na opção OK.*
- 7. O caso de uso retorna ao fluxo principal no passo 2.*

Fluxos Alternativos (FA)

FA1. Bean Não Preenchido

- 1. O sistema exibe uma mensagem solicitando o preenchimento do campo Bean.*
- 2. O sistema não grava os dados da etapa até que a informação necessária seja fornecida.*
- 3. O caso de uso retorna ao sub-fluxo Alterar Etapa pelo Projeto no passo 3 ou ao sub-fluxo Alterar Etapa no passo 1.*

FA2. Opção Cancelar

- 1. O Gerente de Processo aciona a opção Cancelar.*
- 2. O sistema não grava as informações e fecha a tela.*
- 3. O caso de uso retorna ao fluxo principal no passo 2.*

FA3. Bean Igual ao Anterior

- 1. O Gerente de Processo ao alterar uma etapa seleciona o bean que já estava anteriormente.*
- 2. O sistema exibe uma mensagem de alerta.*
- 3. O caso de uso retorna ao sub-fluxo Alterar Etapa pelo Projeto no passo 3 ou ao sub-fluxo Alterar Etapa no passo 1.*

Pré-Condições

1. *Apenas o Gerente de Processo que criou o projeto pode criar, alterar ou excluir etapas nele.*
2. *Deve existir pelo menos uma classificação cadastrada.*
3. *Pelo menos um bean deve ter sido criado.*

Pós-Condições

1. *Projeto com nova etapa ou*
2. *Projeto com etapa alterada ou*
3. *Projeto com etapa(s) excluída(s)*

Anexo VI – Caso de Uso Consultar Projeto

Descrição Sumária

Este caso de uso descreve o processo de consulta aos projetos a partir de outros documentos.

Atores

Gerente de Processo

Fluxo Principal

1. *Este caso de uso se inicia quando o Gerente de Processo entra em alguma etapa ou atividade aberta..*
2. *O Gerente de Processo clica no botão Consultar Projeto.*
3. *O sistema exibe a tela do projeto ao qual o documento atual está associado.*
4. *O documento atual é fechado.*
5. *Este caso de uso termina.*

Pré-Condições

Nenhuma

Pós-Condições

1. *Informações de projeto exibidas.*

Anexo VII – Preencher Atividade Aberta

Descrição Sumária

Este caso de uso descreve o processo de associação de *beans* com as atividades abertas das etapas de projetos.

Atores

Gerente de Processo

Fluxo Principal

1. *Este caso de uso se inicia quando o Gerente de Processo aciona a opção Preencher Atividades Abertas dentro de uma etapa.*
2. *O sistema exibe uma caixa de diálogo com a lista da(s) atividade(s) aberta(s) ainda não associada(s) com nenhum bean para o Gerente de Processo selecionar qual ele deseja preencher.*
3. *O Gerente de Processo seleciona a atividade aberta desejada e clica na opção OK.(FA2, FA3).*
4. *O sistema exibe uma tela preenchida com as informações do projeto, etapa e atividade aberta preenchidas, o campo bean em branco e disponível para edição e as opções Sair, Consultar Projeto e Consultar Etapa.*
5. *O Gerente de Processo escolhe o bean desejado.(FA2).*
6. *O sistema exibe a classificação e o tipo do bean selecionado.*
7. *Caso seja um framework bean, o sistema exibe também a lista das atividades abertas.*
8. *O sistema habilita a opção Salvar.*
9. *O Gerente de Processo clica na opção Salvar.(FA1, FA2).*
10. *O sistema grava as informações da atividade aberta atual.*
11. *Caso seja um framework bean:*
 - a. *O sistema gera as atividades abertas.*
12. *O sistema deixa o campo bean desabilitado para edição.*
13. *O sistema habilita as opções Atividade Aberta Limpar e Atividade Aberta Alterar.*
14. *Caso o Gerente de Processo selecione a opção Consultar Projeto:*
 - a. *O sistema executa o caso de uso “ Consultar Projeto” .*
15. *Caso o Gerente de Processo selecione a opção Consultar Etapa:*
 - a. *O sistema executa o caso de uso “ Consultar Etapa” .*
16. *Caso o Gerente de Processo selecione a opção Atividade Aberta Alterar:*

- a. *O sistema executa o caso de uso “ Manter Informações de Atividade Aberta” .*
17. *Caso o Gerente de Processo selecione a opção Atividade Aberta Limpar:*
 - a. *O sistema executa o caso de uso “ Manter Informações de Atividade Aberta” .*
18. *O Gerente de Processo clica na opção Sair.*
19. *O sistema fecha a tela de atividade aberta.*
20. *Este caso de uso termina.*

Fluxos Alternativos (FA)

FA1. Bean Não Preenchido

1. *O sistema exibe uma mensagem solicitando o preenchimento do campo Bean que foi deixado em branco e a opção OK.*
2. *O Gerente de Processo clica na opção OK.*
3. *O caso de uso retorna ao fluxo principal no passo 4.*

FA2. Opção Cancelar

1. *O sistema cancela a operação e retorna ao formulário original.*
2. *O caso de uso retorna ao fluxo principal no passo 1.*

FA3. Nenhuma Atividade Aberta Seleccionada

1. *O sistema detecta que nenhuma atividade aberta foi seleccionada.*
2. *O sistema exibe uma mensagem solicitando a escolha de uma atividade aberta.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

Pré-Condições

1. *A etapa deve ter um bean do tipo framework.*
2. *Apenas o Gerente de Processo que criou o projeto pode preencher atividades abertas.*
3. *Deve existir pelo menos uma atividade aberta ainda não preenchida.*
4. *Pelo menos um bean deve ter sido criado.*

Pós-Condições

1. *Etapa(s) com atividade(s) aberta(s) preenchida(s)*

Anexo VIII – Caso de Uso Consultar Etapa

Descrição Sumária

Este caso de uso descreve o processo de consulta as etapas a partir das atividades abertas.

Atores

Gerente de Processo

Fluxo Principal

1. *Este caso de uso se inicia quando o Gerente de Processo entra em alguma atividade aberta.*
2. *O Gerente de Processo clica no botão Consultar Etapa.*
3. *O sistema exibe a tela da etapa a qual a atividade aberta está associada.*
4. *A tela da atividade aberta é fechada.*
5. *Este caso de uso termina.*

Pré-Condições

Nenhuma

Pós-Condições

1. *Informações de etapa exibidas*

Anexo IX – Ativar Projeto

Descrição Sumária

Este caso de uso descreve o processo de ativação de um projeto na base de *Beans Composer*.

Atores

Gerente de Processo

Fluxo Principal

1. *Este caso de uso se inicia quando o Gerente de Processo entra em um dos seus projetos com status “ Em Elaboração” .*
2. *O sistema exibe a tela de projeto com os campos Nome, Status, Gerente de Processo, Data e Descrição já devidamente preenchidos e desabilitados para edição; as opções Sair e Ativar Projeto; e uma lista das etapas daquele projeto.*
3. *O Gerente de Processo clica na opção Ativar Projeto. (FA1, FA3).*
4. *O sistema exibe uma mensagem alertando que depois de ativo um projeto não poderá mais ser alterado e perguntando se o Gerente de Processo deseja continuar. (FA2)*
5. *O sistema cria um processo na Biblioteca de Processos.*
6. *Para cada etapa do projeto, o sistema verifica quais são os beans associados a elas e cria todas as atividades correspondentes na Biblioteca de Processos.*
7. *Caso existam etapas associadas com framework beans, o sistema verifica os beans associados com cada atividade aberta e cria as atividades correspondentes na Biblioteca de Processos.*
8. *O sistema coloca o status “ Ativo” em todas as etapas e atividades abertas do projeto.*
9. *O sistema coloca o status “ Ativo” no projeto.*
10. *O sistema fecha o formulário de projeto e retorna para a visão de Projetos.*
11. *Este caso de uso termina.*

Fluxos Alternativos (FA)

FA1. Nenhuma Etapa Criada

1. *O sistema exibe uma mensagem avisando que não é possível realizar a operação, pois ainda não foi criada nenhuma etapa.*
2. *O caso de uso retorna ao fluxo principal no passo 2.*

FA2. Cancelamento de Operação

1. *O Gerente de Processo escolhe cancelar a operação.*
2. *O sistema não realiza nenhuma alteração no projeto ou nas suas etapas.*
3. *O caso de uso retorna ao fluxo principal no passo 2.*

FA3. Framework Bean sem Atividade Aberta Associada

1. *O sistema detecta que alguma etapa encontra-se associada com um bean do tipo framework e que não foram associados beans com todas as suas atividades abertas.*
2. *O sistema exibe uma mensagem informando ao Gerente de Processo que nem todas as atividades abertas foram preenchidas e perguntando se ele deseja continuar assim mesmo.*
3. *O Gerente de Processo confirma que deseja continuar com a ativação do projeto.*
4. *O caso de uso retorna ao fluxo principal no passo 4.*

Pré-Condições

1. *Apenas o Gerente de Processo que criou o projeto pode ativá-lo.*
2. *O projeto não pode já estar ativo.*
3. *O projeto já deve possuir pelo menos uma etapa criada.*

Pós-Condições

1. *Projeto, etapas e atividades abertas com status de ativo, o que impede qualquer tipo de atualização nas suas informações.*
2. *Novo processo e atividades disponíveis na Biblioteca de Processos, prontos para serem executados na base de aplicação.*

Anexo X – Visualizar Processo no *Architect*

Descrição Sumária

Este caso de uso descreve o processo de abrir o *Lotus Workflow Architect* a partir da definição de um processo na *Beans Composer*.

Atores

Administrador de *Beans* e *Lotus Workflow Architect*

Fluxo Principal

1. *Este caso de uso se inicia quando o Administrador de Beans entra na definição de um processo ou em um bean e aciona a opção Visualizar Processo no Architect.*
2. *O sistema gera um arquivo XML com as definições do processo selecionado.*
3. *O sistema inicializa o Lotus Workflow Architect exibindo as definições do arquivo XML recém criado.*
4. *Este caso de uso termina.*

Pré-Condições

1. *O Administrador de Beans deve possuir o Lotus Workflow Architect instalado na máquina.*

Pós-Condições

1. *Lotus Workflow Architect com o processo especificado aberto para consultas.*

Anexo XI – Caso de Uso Manter Informações de Atividade Aberta

Descrição Sumária

Este caso de uso descreve o processo de manutenção de informações de atividades abertas.

Atores

Gerente de Processo

Fluxo Principal

1. *Este caso de uso se inicia quando o Gerente de Processo entra em alguma das suas atividades abertas.*
2. *Caso o Gerente de Processo clique no botão Alterar Atividade Aberta, inicia-se o sub-fluxo Alterar Atividade Aberta. (SF1)*
3. *Caso o Gerente de Processo clique no botão Limpar Atividade Aberta, inicia-se o sub-fluxo Excluir Atividade Aberta. (SF2).*
4. *O Gerente de Processo clica na opção Sair.*
5. *O sistema fecha o formulário.*
6. *Este caso de uso termina.*

Sub-Fluxos (SF)

SF1. Alterar Atividade Aberta

1. *O sistema exibe uma caixa de diálogo solicitando a escolha do novo bean.*
2. *O Gerente de Processo seleciona o bean desejado e clica na opção OK (FA2).*
3. *O sistema preenche os campos Classificação e Tipo do Bean, de acordo com a escolha no passo anterior.*
4. *O Gerente de Processo confere as informações clica na opção OK (FA1, FA2, FA3).*
5. *Se o bean anterior era Blackbox Bean e o novo bean for Framework Bean:*
 - a) *O sistema gera as atividades abertas para o novo bean.*
6. *Se o bean anterior era Framework Bean e o novo bean for Blackbox Bean:*
 - a) *O sistema exclui as atividades abertas existentes anteriormente.*
7. *Se o bean anterior era Framework Bean e o novo bean for Framework Bean:*
 - a) *O sistema exclui as atividades abertas existentes anteriormente.*
 - b) *O sistema gera as atividades abertas para o novo bean.*

8. *O sistema atualiza as informações na atividade aberta.*
9. *O caso de uso retorna ao fluxo principal no passo 1.*

SF2. Limpar Atividade Aberta

1. *O sistema limpa os campos Bean, Tipo de Bean, Classificação de Bean e Atividade Aberta.*
2. *O sistema fecha o formulário de atividade aberta.*
3. *O sistema exibe o formulário de etapa.*
4. *O caso de uso retorna ao fluxo principal no passo 1.*

Fluxos Alternativos (FA)

FA1. Campo Obrigatório Não Preenchido

1. *O sistema exibe uma mensagem solicitando o preenchimento do campo bean que foi deixado em branco.*
2. *O caso de uso retorna ao sub-fluxo Alterar Atividade Aberta no passo 1.*

FA2. Cancelamento de Operação

1. *O Gerente de Processo escolhe cancelar a operação.*
2. *O sistema não executa a operação.*
3. *O caso de uso retorna ao fluxo principal no passo 1.*

FA3. Bean Igual ao Anterior

1. *O Gerente de Processo ao alterar uma atividade aberta seleciona o mesmo bean que já estava anteriormente.*
2. *O sistema exibe uma mensagem de alerta.*
3. *O caso de uso retorna ao sub-fluxo Alterar Atividade Aberta no passo 1.*

Pré-Condições

2. *Apenas o Gerente de Processo que criou o projeto pode alterá-lo ou excluí-lo.*
3. *Para alterar ou excluir um projeto ele não pode estar ativo.*

Pós-Condições

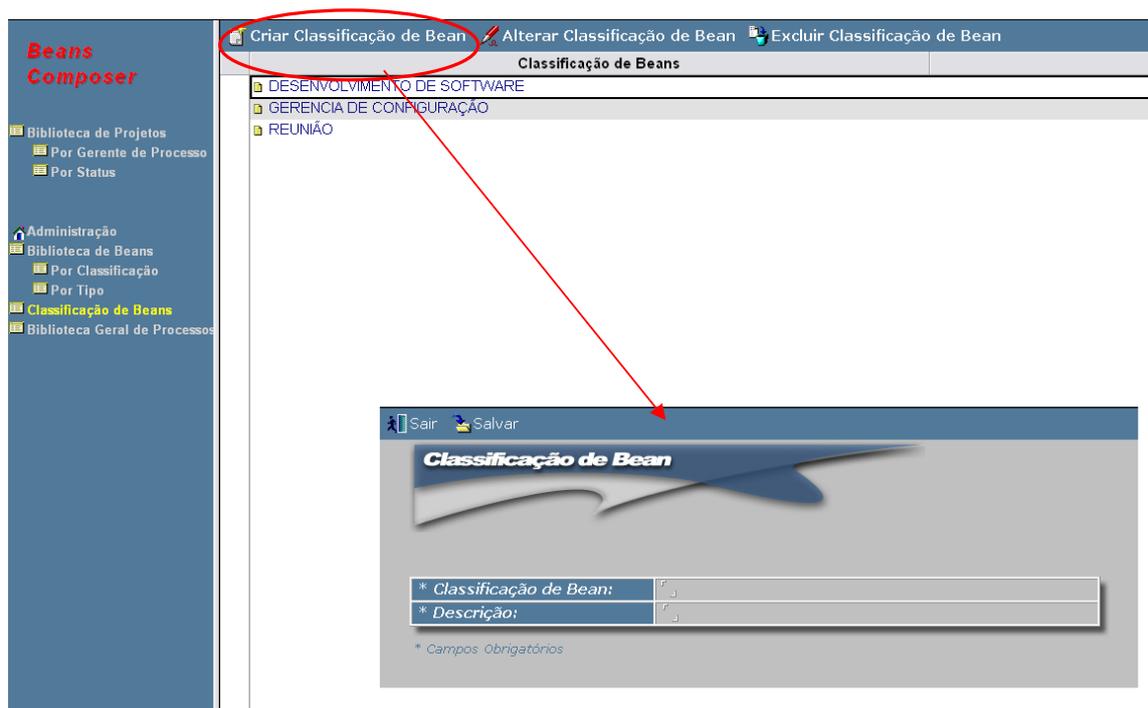
1. *Informações de atividades abertas atualizadas*

Anexo XII – Manual do Usuário

1. Cadastrando Classificações de Bean

Para que seja possível distinguir os *beans* de acordo com as suas áreas de aplicação, é necessário cadastrar uma lista de classificações de *bean*. Alguns exemplos de classificação de *bean* são: reunião, desenvolvimento de software e gerência de configuração.

Uma classificação de *bean* deve conter um nome e uma descrição que ajude a entender os seus objetivos. Na figura 1 é mostrado como o Administrador de *Beans* pode cadastrar uma nova classificação. As instruções detalhadas sobre como cadastrar, alterar ou excluir uma classificação de *bean* estão descritas no caso de uso Manter Informações de Classificação de *Bean*, disponível no Anexo III.



Anexo XII - Figura 1 – Classificação de *Bean*

2. Criando um Bean

Ao criar um *bean*, o Administrador de *Beans* deverá informar o seu nome, o processo associado, o tipo de *bean* (*framework* ou *blackbox bean*), a classificação do *bean*, os seus objetivos, seu contexto de utilização e a descrição da solução que ele oferece.

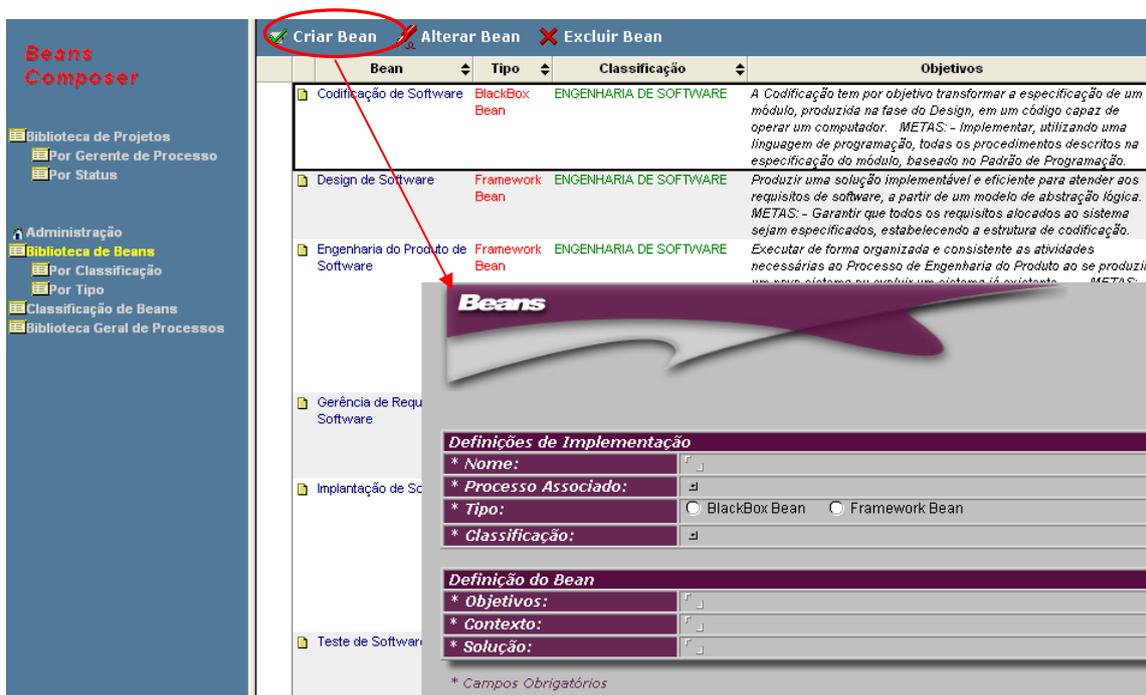
O nome do *bean* deve ser único para que cada *bean* possa ser facilmente identificado. O processo deverá ser escolhido de acordo com as definições de processos criados no *Lotus Workflow Architect*. Cada *bean* deverá ser associado com um único processo, do qual herdará a lista de atividades e as suas características principais.

A escolha do tipo de *bean* deve ser baseada nas necessidades de conexão de outros *beans*. Caso o Administrador de *Beans* opte pelo tipo *blackbox bean*, o processo associado será seguido em sua forma original. No caso do tipo *framework bean*, o sistema permitirá que sejam escolhidas atividades onde poderão ser conectados outros *beans*.

A classificação do *bean* deverá ser escolhida de acordo com as opções previamente cadastradas. Um *bean* poderá possuir mais de uma classificação, caso ele se aplique a mais de um domínio de aplicação.

Todos os *beans* criados ficam disponíveis para consulta na Biblioteca de *Beans*.

Na figura 2 é mostrado como o Administrador de *Beans* pode cadastrar um novo *bean*. As instruções detalhadas sobre como cadastrar, alterar ou excluir um *bean* estão descritas no caso de uso Manter Informações de *Beans*, disponível no Anexo IV.



Anexo XII - Figura 2 – Bean

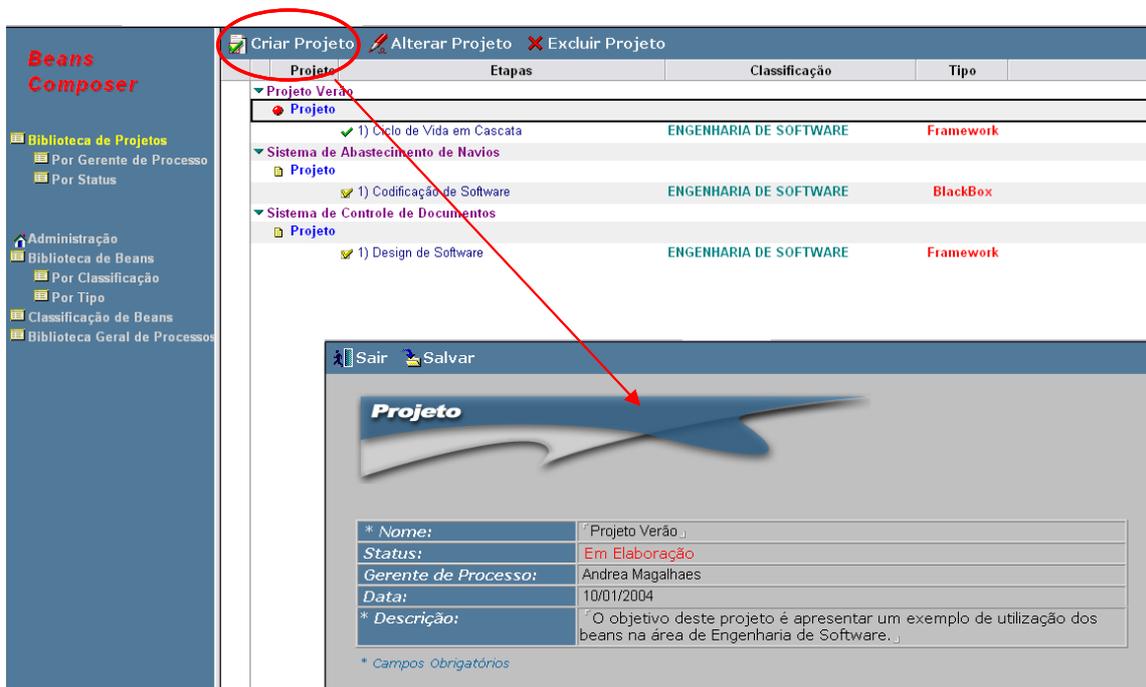
3. Criando um Projeto

Definir um projeto na *Beans Composer* corresponde a criar as diversas etapas que posteriormente serão executadas na base de aplicação do *Lotus Workflow*.

Ao criar um projeto, o Gerente de Processo deverá informar um nome único e uma descrição que ajude a entender a aplicabilidade do projeto que foi criado. Na criação de um projeto, o sistema registra automaticamente, o nome do Gerente de Processo responsável pelo novo projeto, a data de criação e o status (inicialmente Em Elaboração).

Depois de criado o projeto, ele fica disponível na Biblioteca de Projetos. A partir de então, é possível começar a criar etapas dentro dele. A criação das etapas é abordada no próximo item.

Na figura 3 é mostrado como o Gerente de Processo pode criar um projeto. As instruções detalhadas sobre como criar, alterar ou excluir um projeto estão descritas no caso de uso Manter Informações de Projeto, disponível no Anexo V.



Anexo XII - Figura 3 – Projeto

4. Criando Etapas

Uma etapa é um passo de trabalho dentro de um projeto. Cada etapa estará associada com um único *bean*. As etapas são numeradas seqüencialmente pelo sistema. Definir uma etapa em um projeto, consiste em especificar a sua classificação e o *bean* associado.

Uma etapa pode possuir mais de uma classificação, escolhida na lista previamente cadastrada pelo Administrador de *Beans*. De acordo com a(s) classificação(ões) escolhida(s) será apresentada a lista de *beans* possíveis. Caso o Gerente de Processo opte por um *framework bean*, serão exibidas as atividades abertas para que seja possível associá-las com novos *beans* através do botão Preencher Atividades Abertas que será detalhado no próximo item.

Na figura 4 é mostrado como o Gerente de Processo pode criar uma etapa. As instruções detalhadas sobre como criar, alterar ou excluir uma etapa estão descritas no caso de uso Manter Informações de Etapa, disponível no Anexo VI.

The screenshot shows a web application interface for managing projects. At the top, there is a navigation bar with 'Sair', 'Ativar Projeto', and 'Etapa'. Below this, a 'Projeto' header is visible. A dropdown menu for 'Etapa' is open, showing options: 'Criar', 'Alterar', and 'Excluir'. A red arrow points from the 'Criar' option to the 'Criar' button in the 'Etapa' form below.

Projeto

* Nome:	Projeto Verão
Status:	Em Elaboração
Gerente de Processo:	Andrea Magalhaes
Data:	10/01/2004
* Descrição:	O objetivo deste projeto é apresentar um exemplo de utilização dos beans na área de Engenharia de Software

* Campos Obrigatórios

Etapas do Projeto:

Etapas	Tipo	Classificação	Situação
→ 1) Ciclo de Vida em Cascata	Framework	ENGENHARIA DE SOFTWARE	Faltando Associar Beans

Etapa

Projeto:

Nome do Projeto:	Projeto Verão
Status do Projeto:	Em Elaboração
Gerente de Processo:	Andrea Magalhaes

Etapa:

Número da Etapa:	2
* Bean:	<input type="checkbox"/>

* Campos Obrigatórios

Anexo XII - Figura 4 – Criando Etapa

5. Preencher Atividades Abertas

Uma atividade aberta permite que sejam conectados outros *beans* ao projeto de forma que este se adapte às necessidades do Gerente de Processo. Uma atividade aberta só faz parte de *beans* do tipo *framework*.

O preenchimento de uma atividade aberta poderá ser feito durante a criação de uma etapa ou a qualquer momento antes da ativação do projeto.

Na figura 5 é mostrado como o Gerente de Processo pode preencher uma atividade aberta. As instruções detalhadas sobre realizar este procedimento estão descritas no caso de uso Preencher Atividade Aberta, disponível no Anexo VII.

Sair Consultar Etapa Preencher Atividades Abertas

Etapa

Projeto:

Nome do Projeto:	Projeto Verão
Status do Projeto:	Em Elaboração
Gerente de Processo:	Andrea Magalhaes

Etapa:

Número da Etapa:	1
* Bean:	Ciclo de Vida em Cascata
Classificação:	ENGENHARIA DE SOFTWARE
Tipo do Bean Selecionado:	Framework Bean
Atividades Abertas:	Análise Codificação Design Implantação Modelagem Teste

* Campos Obrigatórios

Atividade Aberta

Projeto:

Nome do Projeto:	Projeto Verão
Status do Projeto:	Em Elaboração
Gerente do Processo:	Andrea Magalhaes

Etapa:

Número da Etapa:	1
Bean Associado:	Ciclo de Vida em Cascata

Atividade Aberta Atual: Análise

* Bean Associado:

* Campo Obrigatório

Atividades Abertas da Etapa

Atividades Abertas	Beans Associados	Situação
→ Análise		Faltando Associar Beans
→ Codificação		Faltando Associar Beans
→ Design		Faltando Associar Beans
→ Implantação		Faltando Associar Beans
→ Modelagem		Faltando Associar Beans
→ Teste		Faltando Associar Beans

Anexo XII - Figura 5 – Preencher Atividade Aberta

6. Ativar Projeto

Um projeto possui dois status: Em Elaboração ou Ativo. No momento de sua criação ele recebe o status Em Elaboração. Ao executar a operação Ativar Projeto, o projeto e as suas etapas recebem o status Ativo, que impede qualquer alteração ou exclusão no projeto.

Para que seja possível ativar um projeto, todas as suas etapas já devem estar completas. Por completo entende-se que todas as etapas associadas com *framework beans* devem ter as suas atividades abertas preenchidas.

Ao ativar um projeto, o Gerente de Processo torna-o disponível para ser executado na base de aplicação do *Lotus Workflow*.

Na figura 6 é mostrado como o Gerente de Processo pode ativar um projeto. As instruções detalhadas sobre como ativar um projeto estão descritas no caso de uso Ativar Projeto, disponível

no Anexo IX.

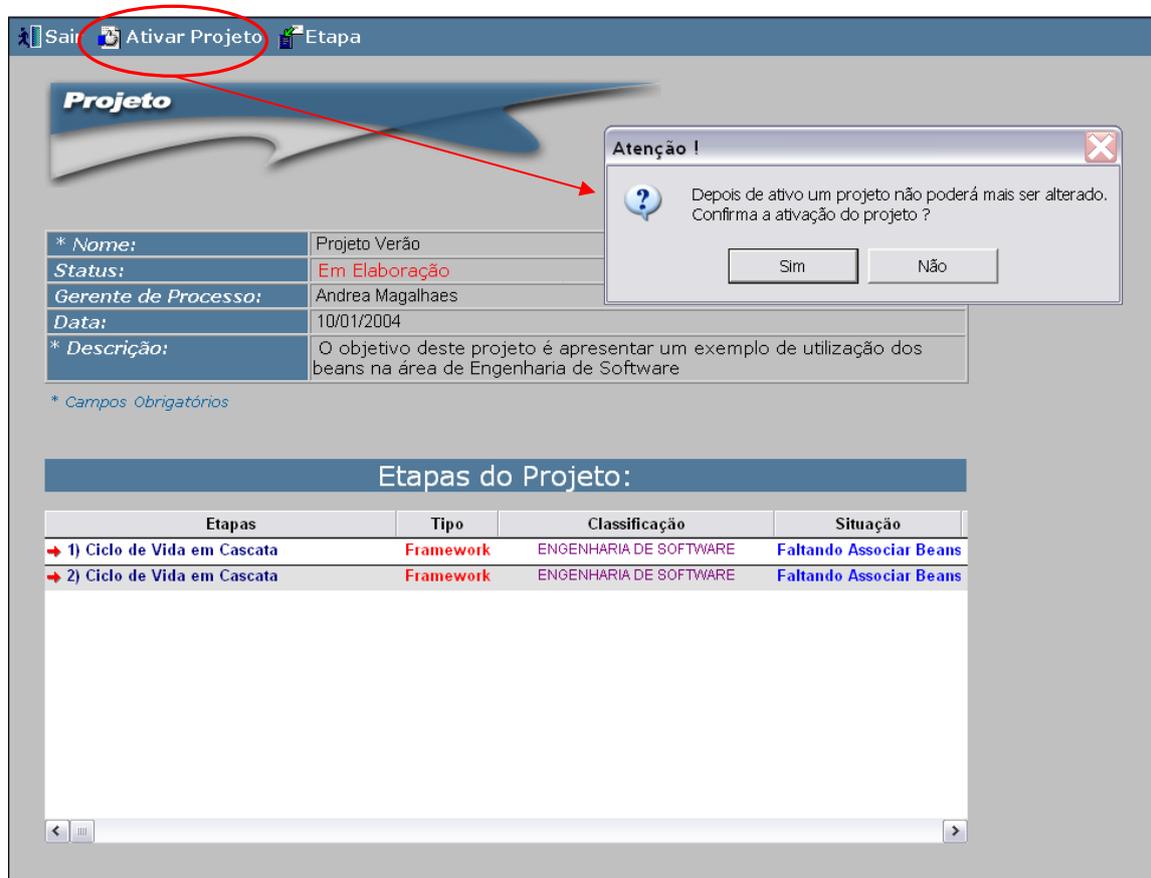


Figura 6 – Ativar Projeto

Anexo XIII – Exemplo Código XML

```
<?xml version="1.0" encoding="UTF-16" ?>
<Model Author="Andrea Magalhaes/Casa" CreateDate="24/11/2003" CreateTime="23:17:58">
<OrgUnits>
<OrgUnit Type="Workgroup" Id="OrgUnit1">
  <Name>Gerência de Configuração</Name>
</OrgUnit>
<OrgUnit Type="Role" Id="OrgUnit2">
  <Name>Gerente de Processo</Name>
</OrgUnit>
<OrgUnit Type="Workgroup" Id="OrgUnit3">
  <Name>Inspeção</Name>
</OrgUnit>
<OrgUnit Type="Department" Id="OrgUnit4">
  <Name>Sistemas de Informação</Name>
</OrgUnit>
<OrgUnit Type="Role" Id="OrgUnit5">
  <Name>Analista de Requisitos</Name>
</OrgUnit>
<OrgUnit Type="Role" Id="OrgUnit6">
  <Name>Analista de Sistemas</Name>
</OrgUnit>
<OrgUnit Type="Role" Id="OrgUnit7">
  <Name>Programador</Name>
</OrgUnit>
<OrgUnit Type="Role" Id="OrgUnit8">
  <Name>Testador</Name>
</OrgUnit>
<OrgUnit Type="Role" Id="OrgUnit9">
  <Name>Gerente de Projeto</Name>
</OrgUnit>
</OrgUnits>
<Process Id="Process1">
```

```

<Name>Engenharia do Produto de Software 1</Name>
<Owner>
<Includes>
  <OrgUnitLink Id="OrgUnit1" />
  <OrgUnitLink Id="OrgUnit2" />
  <OrgUnitLink Id="OrgUnit3" />
  <OrgUnitLink Id="OrgUnit4" />
  </Includes>
</Owner>
<Initiator InitiatedBy="All" />
<Forms>
<MainDocForm>
  <Form>Sample Form</Form>
  </MainDocForm>
<NewDocForm Scope="None" />
</Forms>
<Routing Mechanism="Immediate" />
<JoinSetting BinderStrategy="KeepNewestOnly" MainDocStrategy="RandomSelect" />
<Reader Scope="All" />
<ProcessStart Id="Node5" />
<ProcessEnd Id="Node6" />
<Activity Id="Node1">
  <Name>1. Gerência de Requisitos</Name>
  <Owner>
  <Includes>
    <OrgUnitLink Id="OrgUnit5" />
    </Includes>
  </Owner>
  <Reassignment Allowance="AnyPerson" />
  <JoinSetting MainDocStrategy="RandomSelect" />
  </Activity>
<Activity Id="Node2">
  <Name>2. Design de Software</Name>
  <Owner>

```

```

<Includes>
  <OrgUnitLink Id="OrgUnit6" />
  </Includes>
</Owner>
<Reassignment Allowance="AnyPerson" />
<JoinSetting MainDocStrategy="RandomSelect" />
</Activity>
<Activity Id="Node3">
  <Name>3. Codificação de Software</Name>
<Owner>
<Includes>
  <OrgUnitLink Id="OrgUnit7" />
  </Includes>
</Owner>
<Reassignment Allowance="AnyPerson" />
<JoinSetting MainDocStrategy="RandomSelect" />
</Activity>
<Activity Id="Node4">
  <Name>4. Testes de Software</Name>
<Owner>
<Includes>
  <OrgUnitLink Id="OrgUnit8" />
  </Includes>
</Owner>
<Reassignment Allowance="AnyPerson" />
<JoinSetting MainDocStrategy="RandomSelect" />
</Activity>
<Activity Id="Node7">
  <Name>5. Implantação de Software</Name>
<Owner>
<Includes>
  <OrgUnitLink Id="OrgUnit5" />
  <OrgUnitLink Id="OrgUnit6" />
  <OrgUnitLink Id="OrgUnit2" />

```

```

<OrgUnitLink Id="OrgUnit9" />
<OrgUnitLink Id="OrgUnit7" />
<OrgUnitLink Id="OrgUnit8" />
  </Includes>
  </Owner>
<Reassignment Allowance="AnyPerson" />
<JoinSetting MainDocStrategy="RandomSelect" />
  </Activity>
<RoutingRelation Id="Edge1" SourceId="Node1" TargetId="Node2">
  <Name>abc</Name>
  </RoutingRelation>
<RoutingRelation Id="Edge2" SourceId="Node2" TargetId="Node3">
  <Name>def</Name>
  </RoutingRelation>
<RoutingRelation Id="Edge3" SourceId="Node3" TargetId="Node4">
  <Name>ghi</Name>
  </RoutingRelation>
<RoutingRelation Id="Edge4" SourceId="Node4" TargetId="Node7">
  <Name>jkl</Name>
  </RoutingRelation>
<RoutingRelation Id="Edge5" SourceId="Node5" TargetId="Node1">
  <Name />
  </RoutingRelation>
<RoutingRelation Id="Edge6" SourceId="Node7" TargetId="Node6">
  <Name>mno</Name>
  </RoutingRelation>
<Layout>
<Node Id="Node1">
  <Position X="255" Y="117" />
  </Node>
<Node Id="Node2">
  <Position X="422" Y="117" />
  </Node>
<Node Id="Node3">

```

```
<Position X="422" Y="247" />
</Node>
<Node Id="Node4">
  <Position X="423" Y="364" />
  </Node>
<Node Id="Node5">
  <Position X="126" Y="117" />
  </Node>
<Node Id="Node6">
  <Position X="712" Y="364" />
  </Node>
<Node Id="Node7">
  <Position X="579" Y="364" />
  </Node>
<Edge Id="Edge1" OrthogonalMinimalLength="20" />
<Edge Id="Edge2" OrthogonalMinimalLength="20" />
<Edge Id="Edge3" OrthogonalMinimalLength="20" />
<Edge Id="Edge4" OrthogonalMinimalLength="20" />
<Edge Id="Edge5" OrthogonalMinimalLength="20" />
<Edge Id="Edge6" OrthogonalMinimalLength="20" />
</Layout>
</Process>
</Model>
```

Anexo XIV – *Bean* de Ciclo de Vida em Cascata

Nome:
Ciclo de Vida em Cascata
Objetivos:
Executar de forma organizada e consistente as atividades necessárias ao Ciclo de Vida em Cascata ao se produzir um novo sistema ou evoluir um sistema já existente.
Contexto:
Este <i>bean</i> se aplica à Área de Desenvolvimento de Sistemas.
Solução:
Os objetivos principais de cada atividade são os seguintes:
<p>1. Modelagem</p> <p>As finalidades da atividade de modelagem são:</p> <ul style="list-style-type: none"> • Entender a estrutura e a dinâmica da organização na qual um sistema deve ser implantado • Entender os problemas atuais da organização e identificar as possibilidades de melhoria • Assegurar que os clientes, usuários e desenvolvedores tenham um entendimento comum da organização • Derivar os requisitos de sistema necessários para sustentar a organização. <p>Para atingir essas metas, a atividade de modelagem descreve como desenvolver uma visão da nova organização e, com base nesta visão, definir os processos, os papéis e as responsabilidades dessa organização.</p>
<p>2. Análise</p> <p>A finalidade da atividade de análise é:</p> <ul style="list-style-type: none"> • Estabelecer e manter concordância com os clientes e outros envolvidos sobre o que o sistema deve fazer. • Oferecer aos desenvolvedores do sistema uma compreensão melhor dos requisitos do sistema. • Definir as fronteiras do sistema • Fornecer uma base para estimar o custo e o tempo de desenvolvimento do sistema. • Definir uma interface de usuário para o sistema, focando nas necessidades e metas dos usuários. <p>Para atingir essas metas, é importante, antes de tudo, compreender a definição e o escopo do problema que tentamos resolver com este sistema.</p>

3. *Design*

O *design* tem por objetivo produzir uma solução implementável e eficiente para atender aos requisitos de *software*, a partir de um modelo de abstração lógica.

4. Codificação

A codificação tem por objetivo transformar a especificação de um módulo, produzida na fase do *design*, em um código capaz de operar um computador.

5. Testes

O teste tem por objetivo validar de forma completa e sistemática os requisitos do sistema e verificar seus componentes de acordo com o projeto, visando identificar o maior número de defeitos antes da entrega, diminuindo custos de retrabalho e tempo de entrega do sistema, avaliando a qualidade do *software* produzido.

6. Implantação

A Implantação tem por objetivo garantir que o produto de *software* estará disponível para os usuários e que estes estarão capacitados a utilizá-lo com sucesso. Para isso deve-se garantir que os usuários foram adequadamente treinados e que toda a documentação necessária para a instalação e utilização do sistema encontra-se disponível e completa.

Executores:

Cliente/Usuários, Analistas de Negócios, Analista/Programador, Administração de Dados, Garantia da Qualidade de *Software*, Gerência de Configuração.

Documentos:

Especificação Física de Dados, Especificação de Interface, Especificação de Módulos, Módulo Fonte/Compilado, Manual do Usuário, Material de Treinamento, Aprovação de Teste de Aceite, Plano de Teste de *Software*, Casos de Teste, Relatório de Teste, Documento de Aprovação do Teste de Sistema, Documento de Requisitos de *Software*, Especificação Funcional, Modelo de Dados, Especificação Lógica de Dados.

Anexo XV – *Bean* de Teste

Nome:
Teste
Objetivos:
<p>O Teste tem por objetivo validar de forma completa e sistemática os requisitos do sistema e verificar seus componentes de acordo com o projeto, visando identificar o maior número de defeitos antes da entrega, diminuindo custos de re-trabalho e tempo de entrega do sistema, avaliando a qualidade do <i>software</i> produzido.</p> <p>METAS:</p> <ul style="list-style-type: none"> • Diminuir o índice de re-trabalho em relação ao esforço total do projeto; • Reduzir o número de manutenções corretivas em produção; • Diminuir o tempo de teste dos projetos; • Identificar erros ao final da fase de construção; • Diminuir o número de defeitos identificados no ambiente de Aceite.
Contexto:
Este <i>bean</i> se aplica à Área de Desenvolvimento de Sistemas.
Solução:
<p>Os objetivos principais de cada atividade são os seguintes:</p> <p>1. Planejar Testes</p> <p>Construir o Plano de Teste de <i>Software</i> – contemplando os testes de Unidade, Integração, Sistema e Aceite.</p> <ul style="list-style-type: none"> • Planejar Teste de Unidade: <p>O Planejamento de Teste de Unidade visa descrever os testes necessários à verificação dos módulos especificados na fase de <i>Design</i>. Planeja a ordem das unidades a serem testadas, os casos de testes associados e os resultados esperados.</p> <ul style="list-style-type: none"> • Planejar Teste de Integração: <p>O Planejamento de Teste de Integração visa a integração funcional através da junção das unidades a fim de cumprir as funções projetadas. Testa as interfaces internas e externas ao sistema.</p> <ul style="list-style-type: none"> • Planejar Teste de Sistema:

O Planejamento de Teste de Sistema visa testar a funcionalidade geral do sistema (inclusive o inter-relacionamento com outros sistemas), usabilidade, recuperação (*restart e recovery*), *stress*, desempenho e instalação.

- Planejar Teste De Aceite:

O Plano de Teste de Aceite visa testar os requisitos do sistema bem como a performance e o impacto junto aos sistemas inter-relacionados. Testa também a conversão para a situação de produção.

2. Gerar Casos de Testes

Gerar os casos de testes necessários à verificação estrutural das unidades produzindo os testes necessários à completa verificação dos módulos, funções etc.

3. Gerar Ambiente de Testes

- Definir Ambiente de Teste:

Definir as necessidades de alterações no ambiente de processamento para suportar o teste de *software* planejado: planejar necessidades de espaço de armazenamento especial; definir arquivos existentes ou novos necessários ao teste; enumerar data/hora específica para teste e pessoas envolvidas, caso necessário.

- Montar Ambiente de Teste:

Em conjunto com a área de suporte, deve ser montado o ambiente de teste definido.

4. Executar Testes

- Executar Teste de Unidade:

A execução do Teste de Unidade deverá seguir o Plano de Teste de *Software* de acordo com os casos de uso definidos para cada unidade. O resultado de cada teste deverá ser registrado de acordo com o modelo Relatório de Testes.

- Executar Teste de Integração:

A execução de teste de Integração deverá seguir o Plano de Teste correspondente e os scripts associados. Também deverão ser registrados os resultados ocorridos através do modelo Relatório de Testes.

- Executar Teste de Sistema:

A partir dos resultados do teste de integração serão efetuados os testes de sistema, baseados no Plano de Teste de *Software*, visando validar a funcionalidade geral do sistema (inclusive o inter-relacionamento com outros sistemas), usabilidade, recuperação (*restart e recovery*), *stress*, performance e instalação. Deverão também ser registrados todos os resultados através do modelo Relatório de Testes.

- Executar Teste de Aceite:

Executar os casos de teste definidos para o teste de aceite com o usuário. Este teste visa validar os requisitos do sistema bem como o impacto junto aos sistemas inter-relacionados. Testa também a conversão para a situação de produção.

Conduzir juntamente com o cliente os testes de funcionalidades baseados em situações com dados reais. Estes testes poderão ser conduzidos no ambiente de homologação ou, nos casos em que a solução será implantada nas instalações do cliente, no ambiente do mesmo.

5. Avaliar e Aprovar Testes

Baseado nos resultados dos testes de *software* (Unidade, Integração, Sistema e Aceite) e na avaliação dos resultados, será feita uma aprovação do referido teste. Avaliação do teste significa gerar uma posição consolidada do teste informando os principais eventos ocorridos e os problemas e/ou dificuldades encontradas. Esta avaliação deverá ser formalmente registrada através do Relatório de Testes.

Executores:

Cliente/Usuários, Analistas de Teste, Analista/Programador e Garantia da Qualidade de *Software*.

Documentos:

Aprovação de Teste de Aceite, Plano de Teste de *Software*, Casos de Teste, Relatório de Teste, Documento de Aprovação do Teste de Sistema.