

ADRIANO DINIZ CASTANHEIRA
EDUARDO CAVALCANTE SALOMÃO
GUILHERME STUMPF OLIVEIRA DA SILVA
RODRIGO MACIEL RODRIGUES

AMBIENTE DE MODELAGEM E EXECUÇÃO DE PROCESSOS BASEADO EM BPM

Projeto de Formatura apresentado à disciplina
PCS 2502 – Laboratório de Projeto de
Formatura II, da Escola Politécnica da
Universidade de São Paulo

Área de Concentração:
Engenharia de Computação

São Paulo
2006

ADRIANO DINIZ CASTANHEIRA
EDUARDO CAVALCANTE SALOMÃO
GUILHERME STUMPF OLIVEIRA DA SILVA
RODRIGO MACIEL RODRIGUES

AMBIENTE DE MODELAGEM E EXECUÇÃO DE PROCESSOS BASEADO EM BPM

Projeto de Formatura apresentado à disciplina
PCS 2502 – Laboratório de Projeto de
Formatura II, da Escola Politécnica da
Universidade de São Paulo

Área de Concentração:
Engenharia de Computação

Orientadora: Prof^a, Dr^a.Selma Shin Shimizu
Melnikoff
Co-Orientador: Eng^o. Leonardo Dominguez
Dias

São Paulo
2006

Resumo

O objetivo deste trabalho é aplicar os conceitos do gerenciamento de processos de negócios (BPM), através do desenvolvimento de um ambiente de modelagem, execução e monitoramento desses processos.

Após a revisão da literatura, analisa-se a notação BPMN como ferramenta para modelagem de processos em diferentes áreas e é desenvolvida uma ferramenta para aplicação dos conceitos num ambiente operacional onde se conclui que a ferramenta pode gerar ganhos em todas as dimensões consideradas (rastreabilidade, produtividade, qualidade e segurança).

Palavras-chave: BPM, *workflow*, modelagem de processos, execução de processos.

Abstract

The aim of this project is to apply Business Process Management (BPM) concepts, by developing a process modeling, execution and monitoring framework.

After reviewing the literature, the BPMN notation is analyzed as a process modeling tool targeting different business areas. After that, an application is proposed and developed to test the concepts in an operational environment. Therefore, the conclusion is that this application can achieve the proposed goals in all considered dimensions (productivity, quality and security).

Keywords: BPM, workflow, process modeling, process execution

Índice de Figuras

Figura 1 - Visão vertical da empresa.....	13
Figura 2 - Representação de processos em estruturas verticais.....	13
Figura 3 - Estrutura com processos de negócios horizontais.....	15
Figura 4 - Ciclo de otimização de processos	17
Figura 5 - Modelagem de um processo de um consultório médico (WHITE, 2005)	25
Figura 6 - Modelo de camadas do BPMS típico.....	30
Figura 7 - Componentes de BPMS (MIERS e HARMON, 2005).....	33
Figura 8 - Ciclo de desenvolvimento das atividades.....	39
Figura 9 – Interfaces com o desenvolvedor e com o usuário	44
Figura 10 - Fluxo do Modelo MVC	49
Figura 11 - Relacionamento entre as classes utilizadas.....	51
Figura 12 - Tela de compilação da ferramenta Eclipse	56
Figura 13 - Processo de compilação	57
Figura 14 - Modelo de arquitetura de integração	60
Figura 15 - Modelo de processo de atendimento.....	66
Figura 16 - Processo na ferramenta de modelagem.....	67
Figura 17 - Código resultante da modelagem do processo.....	68

Índice de Tabelas

Tabela 1 - Detalhamento das atividades do ciclo de otimização de processos.....	18
Tabela 2 - Principais elementos da notação BPMN (WHITE, 2005).....	23
Tabela 3 - Capacidades de ferramentas BPM (DUTRA, 2006)	28
Tabela 4 - Versões da ferramenta	41
Tabela 5 - Configurações para implementação de calendário de negócios.....	52
Tabela 6 - Pacotes de suporte para a interface.....	54
Tabela 7 - Testes de integração realizados	63

Sumário

1	Introdução.....	9
1.1	Objetivo	9
1.2	Justificativa e Motivação.....	9
1.3	Metodologia (Contexto do Trabalho)	10
1.4	Organização do documento	11
2	Modelos de Gestão de Empresas.....	12
2.1	Processos de Negócios	12
2.2	Modelo Funcional.....	12
2.3	BPM (Business Process Management).....	14
2.4	Vantagens e desvantagens	19
2.5	Casos de Sucesso	19
2.6	BPM e a Tecnologia da Informação	20
3	Modelagem de Processos de Negócios	21
3.1	Padrões e Notações usados na modelagem.....	21
3.2	BPMN (Business Process Modeling Notation)	22
3.3	Melhores práticas para modelagem	26
4	Tecnologias BPM	28
4.1	Comparação com outras ferramentas	31
4.2	Principais ferramentas e fornecedores.....	32
5	Metodologia do Projeto de Formatura.....	37
5.1	Visão Geral	37
5.2	Aplicação da Metodologia.....	38
5.3	Cronograma	42
6	Desenvolvimento de Ferramenta.....	43
6.1	Requisitos da Ferramenta	43
6.2	Arquitetura.....	46
6.3	Implantação e documentação.....	62
6.4	Testes	62
6.5	Avaliação dos Resultados	64
7	Aplicação no PECE (Fase de testes da ferramenta)	65

7.1	Detalhamento das atividades	65
7.2	Modelagem dos processos	66
7.3	Implantação dos processos e serviços	67
7.4	Testes de aceitação	68
8	Considerações Finais	70
8.1	Conclusão	70
8.2	Contribuição do projeto	70
8.3	Trabalhos Futuros	71
	REFERÊNCIAS	72
	Anexo 1 – Descrição dos Casos de Uso do Sistema.....	74
	Anexo 2 – Relatório de Revisão do Plano de Aceitação e Teste.....	78
	Anexo 3 – Especificações da Interface de Hardware e Software aplicados ao desenvolvimento.....	85
	Anexo 4 – Especificações da Interface de Hardware e Software da aplicação no PECE	89
	Anexo 5 – Pacotes do Projeto.....	90
	Anexo 6 – Manual do Usuário e de Instalação da Ferramenta.....	92
	Anexo 7 – WBS do Projeto	102
	Anexo 8 – Cronograma de Atividades	103

1 Introdução

1.1 *Objetivo*

O objetivo deste trabalho é aplicar os conceitos do gerenciamento de processos de negócios (BPM), através do desenvolvimento de um ambiente de modelagem, execução e monitoramento desses processos.

Através da pesquisa de padrões existentes, da análise de ferramentas de mercado, de comparações com outros modelos de gerência de empresas e da implementação da ferramenta desenvolvida, pretende-se apresentar as vantagens que o BPM pode trazer ao ser adotado pelas empresas, como a aproximação dos usuários no desenvolvimento dos processos, a flexibilidade obtida pela empresa e a melhoria nos resultados através do constante aperfeiçoamento dos processos.

1.2 *Justificativa e Motivação*

Dado o grande número de fracassos nas implementações de ERP e outras tecnologias que visam automatizar os processos de negócios, surgiram novas técnicas e ferramentas que visam melhorar estas estatísticas. Isto é feito pela aproximação da linguagem de desenvolvimento de sistemas à linguagem de modelagem de negócios, facilitando a tarefa do desenvolvedor de traduzir as regras do negócio em código-fonte e diminuindo as barreiras de comunicação entre os analistas de negócio e os desenvolvedores técnicos, o que torna os resultados mais previsíveis.

Desta forma, melhora-se a qualidade do desenvolvimento de software e encurta-se o tempo necessário para tal, além de reduzir-se o número de problemas devido a erros na interpretação das regras de negócio.

1.3 Metodologia (Contexto do Trabalho)

A metodologia adotada para o desenvolvimento das atividades do grupo para o projeto de formatura pode ser descrita em cinco fases:

- Estudo do tema
- Levantamento das necessidades do cliente
- Levantamento dos requisitos da ferramenta
- Desenvolvimento de ferramenta
- Teste da ferramenta

A primeira fase inclui um profundo estudo da BPM, com análise de especificações técnicas, materiais bibliográficos, documentos e casos bem sucedidos de implantação da metodologia. Essa etapa visa o conhecimento da tecnologia, suas limitações, suas capacidades e, principalmente, as melhores práticas que devem ser adotadas em projetos que envolvem a BPM. Além disso, servirá como base para o desenvolvimento nas outras etapas do projeto.

Na segunda fase foram levantados os requisitos reais em conjunto com o cliente, o PECE. Essa entidade tem o interesse em documentar os seus processos para obter um maior controle das atividades realizadas. O grupo participou da modelagem de alguns desses processos buscando identificar e tornar clara as regras de negócios envolvidas em cada atividade. Os resultados desta fase são fundamentais para o desenvolvimento do sistema, de forma a garantir uma aplicação autêntica.

A terceira fase destina-se à consolidação e fixação dos conceitos estudados na primeira etapa com os da segunda a fim de definir os requisitos necessários da ferramenta. Com base nestes dados, será possível a definição de uma ferramenta para desenvolvimento da solução.

Na quarta fase, de desenvolvimento, optou-se por utilizar uma metodologia de desenvolvimento baseada em componentes de forma a tornar a solução customizável. Para isto utilizou-se a metodologia RUP que atende essas características, além de tornar possível a gerência dos requisitos.

Por fim a última fase, de testes, visa à realização de testes de integração e também de aceitação da ferramenta no ambiente do cliente.

1.4 Organização do documento

Este documento segue a seguinte estrutura:

Os capítulos 2 a 4 se dedicam à revisão da literatura, em que são definidos os conceitos e a teoria que servirão de base para o desenvolvimento das demais atividades do trabalho.

No capítulo 2, é estudado o contexto em que o trabalho se insere e será desenvolvido através da apresentação de conceitos relacionados a modelos de gestão de empresas, com maior ênfase para o modelo de gestão por processos.

Já no capítulo 3, a etapa de modelagem de processos de negócio é detalhada, bem como são descritas algumas notações usadas para esse fim. A ênfase é dada na notação BPMN.

Por fim, no capítulo 4, os aspectos de tecnologia relacionados aos conceitos dos capítulos anteriores são estudados e analisados.

O capítulo 5 descreve a metodologia utilizada para a realização do projeto de formatura, descrevendo a seqüência de atividades realizadas, bem como os documentos de gerenciamento do projeto.

Os capítulos 6 e 7 descrevem a metodologia utilizada no desenvolvimento da ferramenta e sua aplicação prática no PECE, apresentando o processo de desenvolvimento da ferramenta, com maior aprofundamento na parte técnica do projeto.

O capítulo 8 apresenta as conclusões obtidas com o trabalho e propõe futuras aplicações

2 Modelos de Gestão de Empresas

2.1 Processos de Negócios

Segundo Davenport apud Usirono (2003), “processo é um conjunto de atividades estruturadas e definidas para produzir uma saída específica para um cliente ou mercado em particular. Além disso, constitui-se numa série de atividades inter-relacionadas que cruzam fronteiras entre área funcionais com entradas e saídas próprias. Ele define também dimensões de desempenho que podem ser medidas e melhoradas como custo, tempo, qualidade e grau de satisfação do cliente”.

Essa é uma definição considerada bem completa sobre um processo de negócio e, seguindo a argumentação de Usirono (2003), as demais definições encontradas na literatura não diferem muito desta.

Dutra (2006) ainda destaca a importância de se acrescentar a essa definição o conceito de coordenação dessas atividades para alcançar os resultados determinados. McGregor (2006a) também destaca dois aspectos nessa definição:

- Processos de negócio envolvem clientes
- Processos de negócio permeiam as barreiras das organizações

O consenso em torno de uma definição e a correta compreensão do conceito de processo de negócio são fundamentais, já que todo o trabalho desenvolvido está baseado nesse conceito.

2.2 Modelo Funcional

Tradicionalmente, as empresas foram projetadas em função da sua própria realidade interna, sendo centradas em si mesmas. Nesse modelo, os esforços de melhoria estão voltados para as atividades internas e funções departamentais, em detrimento dos processos.

Como na Figura 1, a visão que se tem da empresa é vertical, com as áreas funcionais e departamentos muito bem definidos e separados, justamente pelo foco da melhoria estar nas atividades internas de cada uma dessas áreas verticais.

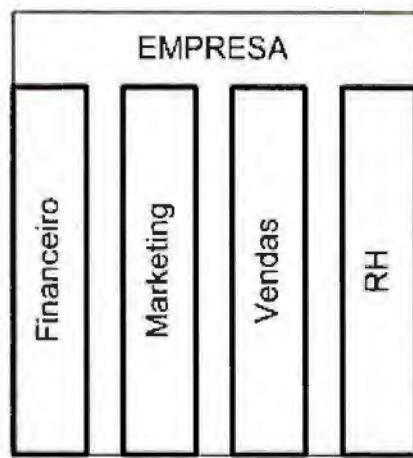


Figura 1 - Visão vertical da empresa

Muitas vezes, empresas que adotam esse modelo não têm sequer a visão do processo, devido à alta especialização das áreas funcionais e também pelo conhecimento das atividades estar restrito às pessoas que as executam.

Numa estrutura verticalizada, os processos são vistos como conjuntos de atividades espalhadas pelas diversas áreas funcionais. Numa representação de processos, essa característica pode ser vista como atividades, representadas por retângulos, espalhadas pelos departamentos e com diversas ligações representando o fluxo de informações e produtos.

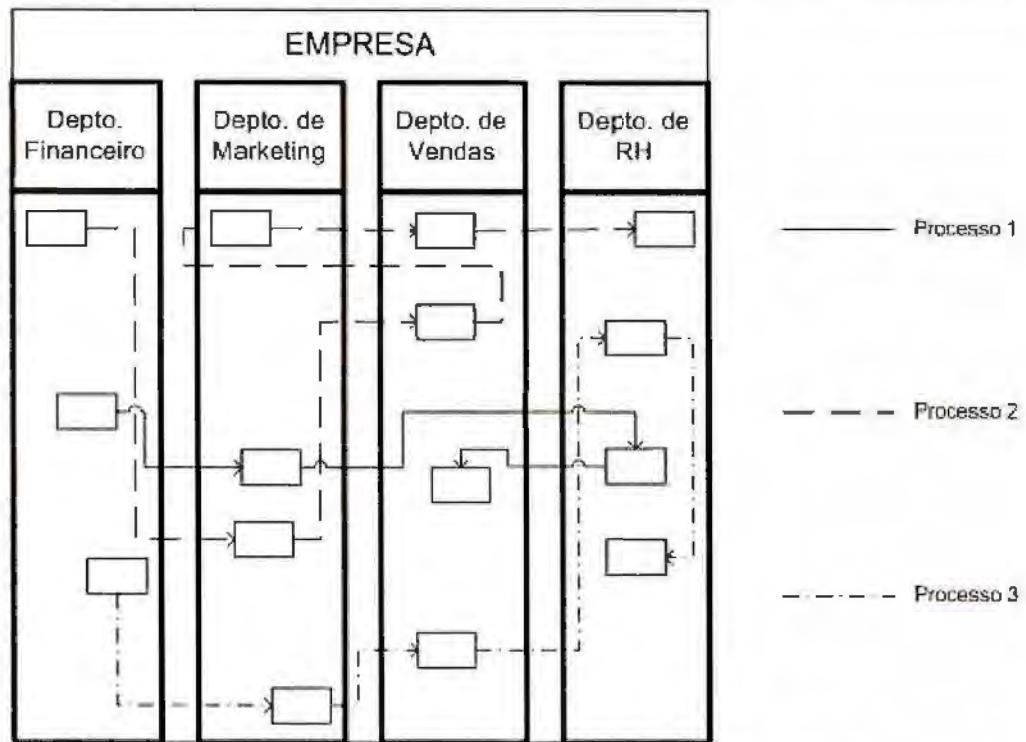


Figura 2 - Representação de processos em estruturas verticais

Como se pode ver na Figura 2, a própria representação do processo numa estrutura vertical torna-se complexa, devido ao cruzamento do fluxo de informações e à falta de organização dessas atividades.

2.3 BPM (*Business Process Management*)

A gestão por processos de negócios, ou BPM, refere-se à compreensão, projeto, execução e otimização das atividades espalhadas por toda a organização e que incorporam pessoas, processos, sistemas e estratégia, segundo Lyneham-Brown apud McGregor (2006a).

Ampliando o conceito, Netto (2004) define o objetivo do BPM é fornecer um modelo de operações que alinhe decisões sobre o processo global de fornecimento de produtos e serviços e que permita avaliar o grau de atendimento às necessidades do cliente.

Nesse ponto faz-se necessário retornar à definição de processos de negócios e aos destaques dados à participação do cliente e à característica de transpor as barreiras da organização, que compõem as principais características do BPM e vistos como fatores de diferenciação em relação ao modelo funcional.

As organizações estão imersas em ambientes com ciclos de negócio cada vez menores e com uma concorrência crescente. Um ambiente com tais características exige que as empresas sejam mais flexíveis e ágeis para que possam responder às demandas do mercado, e para que possam melhorar seu desempenho e alcançar seus resultados. Internamente, a organização ainda deve levar em consideração o aumento de sua complexidade, que a deixa menos dinâmica.(MCGREGOR, 2006a)

Com o foco no cliente, o BPM permite que a empresa comprehenda e atenda às necessidades do mercado, já que a qualidade dos produtos e serviços prestados a esse cliente se reflete diretamente em sua satisfação. (NETTO, 2004) É a mesma linha defendida por McGregor (2006b) que define tal característica como fundamental para o sucesso da organização.

Como os produtos e serviços são produzidos através de processos de negócio, e não através de funções organizacionais, as empresas tentam se adaptar a essa realidade de alta competitividade com projetos de aperfeiçoamento de processos, já que a otimização das funções organizacionais não necessariamente leva a melhoria dos itens produzidos. (JENSEN, 2003) , (NETTO, 2004)

Além disso, Netto (2004) ainda argumenta que, para que as funções organizacionais funcionem de forma eficaz, é necessário identificar e gerenciar suas inter-relações, que estão

presentes nos processos horizontais da empresa. A horizontalização é mostrada na Figura 3, que representa a separação entre os processos voltados para o cliente e as demais atividades internas da organização.

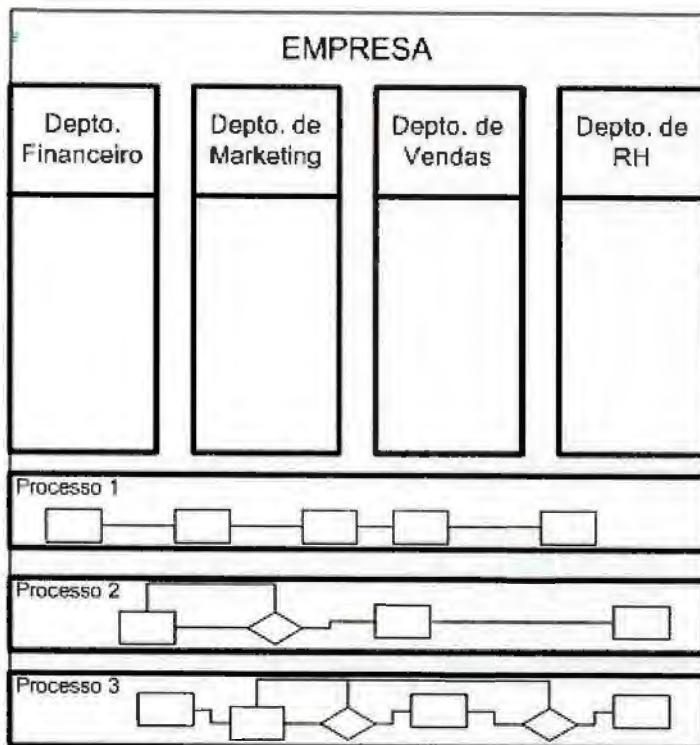


Figura 3 - Estrutura com processos de negócios horizontais

A administração e a gerência dos processos por um dono ou grupo de pessoas, focados em melhorar suas atividades e responder às demandas existentes, bem como a destinação de recursos da organização para esses processos, representam pontos importantes nessa separação em relação às áreas funcionais. Com tais iniciativas, a empresa poderá melhorar seu tempo de resposta às demandas do mercado e aumentar a eficiência de suas operações.

Como conclusão, para McGregor (2006a), o BPM deve ter controle sobre todos os processos da empresa que envolvem clientes de alguma maneira, deve envolver as atividades da empresa em todos os seus níveis hierárquicos e também deve ter a participação de todas as pessoas da empresa para que possa gerar todos os seus benefícios.

2.3.1 Histórico

Smith e Fingars (2003) dividem a evolução da gestão por processos em três etapas, que também levam o nome de ondas, dentre as quais o BPM seria a terceira.

A primeira remete à teoria de Fredrick Taylor, da década de 1920, que pregava que os processos estavam implícitos nas atividades do trabalho e guardados em manuais. Esse primeiro modelo recebeu o nome de Análise de Métodos e Procedimentos e também é considerado por Usirono (2003) como a primeira utilização do conceito de gestão por processos.

A segunda onda caracteriza-se pela presença dos processos dentro de aplicações rígidas, com os ERP's, que, apesar de permitirem um maior controle pelos gerentes de negócios, não os dava o controle sobre todo o ciclo de processo. Essa onda é mais recente que a primeira, situando-se na década de 1990.

Já na terceira onda foi introduzido o conceito de processo de negócio, e é caracterizada pela possibilidade de criação e alteração constante. É baseada na mudança como fator para criação de vantagem competitiva.

Um levantamento presente em Netto (2004) demonstra que entre as épocas da segunda e terceira ondas houve uma grande quantidade de estudos e modelos propostos na área de gestão, o que reforça ainda mais a importância que esse tema tem assumido, tanto na área acadêmica como na área corporativa.

2.3.2 Comparação entre o conceito de processo nos modelos

Miers (2004) compara a visão de processo no modelo funcional e no BPM. No caso do modelo funcional, em que se tem uma empresa com estrutura hierarquizada, um processo é normalmente representado com uma seqüência linear de atividades, agrupadas em uma forma departamental.

Nesse caso, afirma o autor, a ênfase é na velocidade de execução das tarefas e não se exige dos funcionários que conheçam todo o processo, eles são apenas governados por procedimentos que, nesse contexto, é sinônimo de processo.

Já para o BPM, que possui uma visão ampla sobre a empresa, os processos são enxergados como interações em que a comunicação e a sincronização das atividades são tão importantes quanto a ordem em que são executadas.

O foco nesse caso está na satisfação das necessidades do cliente e na qualidade. Esse tipo de organização permitiria aos funcionários enxergarem todo o processo e, com isso, melhorarem seus resultados e sua participação. O conceito de procedimentos é deixado de lado e os funcionários passariam a ser governados por práticas, que evoluem com o passar do tempo.

2.3.3 Implantação do BPM

A mudança do modelo funcional para o BPM está diretamente associada a uma mudança de filosofia da organização e de seus colaboradores e também a uma mudança das regras da empresa, sendo necessário um projeto de implantação para sua adoção. McGregor (2006a)

É fundamental que todos os colaboradores estejam envolvidos com tal projeto, visto que as mudanças podem envolver desde a ligação entre as atividades de cada função organizacional até a reorganização da empresa, e assim podem provocar desconfortos caso os objetivos não tenham sido bem traçados e comunicados claramente às pessoas envolvidas. Netto (2004)

O BPM também possui a característica da evolução contínua do processo para se adaptar aos ciclos de negócios menores. A evolução contínua é tratada através de ciclos de otimização incrementais. O ciclo pode ser visto na Figura 4 e o detalhamento de cada uma das atividades é exibido na Tabela 1. (MIERS, 2006), (DUTRA, 2006).

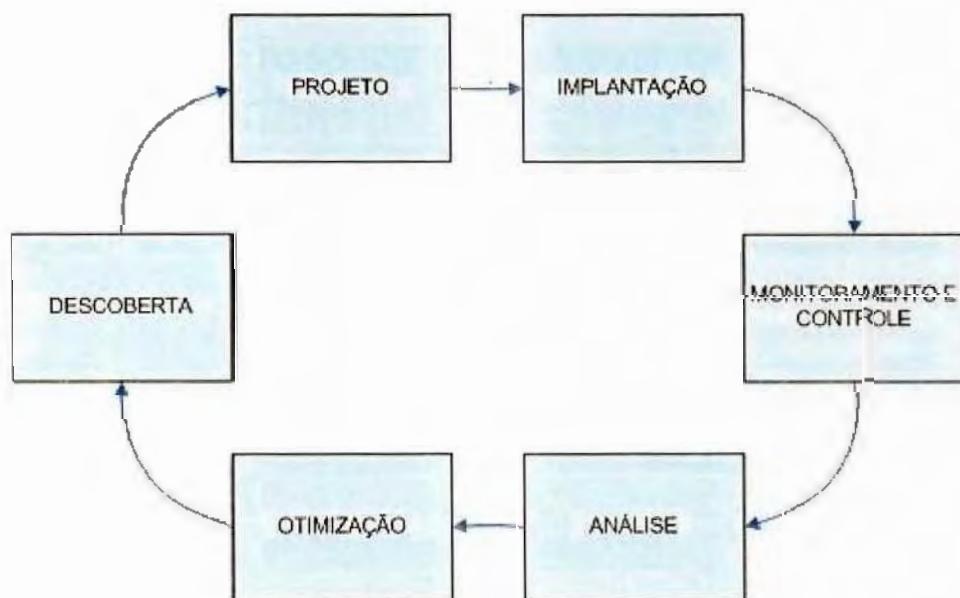


Figura 4 - Ciclo de otimização de processos

Tabela 1 - Detalhamento das atividades do ciclo de otimização de processos

Atividade	Descrição
Descoberta	Inclui o levantamento dos processos e leva ao entendimento de seu fluxo de trabalho, entradas, saídas e participantes. O nível de detalhe dos resultados e os processos em que a atividade é aplicada podem variar conforme o grau de maturidade da empresa em aplicar o gerenciamento por processos e também conforme a perspectiva com a qual se quer avaliar os processos.
Projeto	Inclui modelagem, manipulação e reprojeto dos processos, transformando-os para que atendam às necessidades do negócio e definindo métricas para que possam ser monitorados e analisados posteriormente.
Implantação	Consiste na aplicação e disponibilização dos novos processos criados na etapa de projeto para seus usuários e participantes.
Monitoramento e Controle	Atividades para manter a operação correta dos processos, alocando recursos e tratando exceções.
Análise	Está relacionado à tomada de medidas para analisar o desempenho do processo, fornecendo métricas, análises e oportunidades de melhoria para inovações no processo.
Otimização	É a atividade que relaciona as oportunidades resultantes da etapa de análise com a etapa de descoberta. É responsável pelo reinício do ciclo, caracterizando sua continuidade.

2.4 Vantagens e desvantagens

A adoção da gestão por processos pode trazer diversas vantagens, algumas já citadas anteriormente. A primeira é o fato de analisar o fornecimento de produtos e serviços sob o ponto de vista do consumidor, levando-se a uma análise da cadeia de produção por completo e de como alterações em suas etapas podem alterar a percepção do cliente.

Essa análise da cadeia de produção permite à empresa se conhecer e passar a entender como agrupa valor aos seus clientes. Já no modelo funcional, essa noção de processo é nebulosa e não costuma estar documentada, ficando restrita aos participantes do processo, que ainda possuem uma visão limitada do processo como um todo.

Com a documentação e disseminação dos processos, a equipe ganha em qualidade, que passa a ter uma visão macro de suas atividades e como elas afetam o resultado entregue ao final do processo. O profissional também tem ganhos de produtividade pela sua maior capacitação e conhecimento das atividades.

Outra vantagem inerente ao BPM é o constante aperfeiçoamento dos processos e de suas atividades, de forma incremental. McGregor (2006a) Porém essa é uma característica que pode limitar a adoção do BPM por algumas empresas.

Todas as empresas podem se beneficiar de conhecer melhor seus processos e poder analisá-los. Porém, em empresas em que mudanças culturais e organizacionais não sejam bem aceitas, a adoção de um regime constante de aperfeiçoamento pode trazer desconforto e, no caso mais crítico, não trazer os resultados esperados. Jensen (2003)

2.5 Casos de Sucesso

Em artigo pesquisado (CBR, 2005), uma pesquisa do Gartner Research mostra que 80% das empresas que estão usando BPM podem esperar uma taxa interna de retorno maior que 15%. O trabalho também sugere que houve uma grande taxa de projetos bem sucedidos e implantados no tempo prometido. Das companhias pesquisadas, 95% obtiveram 90% de taxa de sucesso em seus projetos de BPM. Dos projetos implantados, 67% foram terminados em menos de seis meses, com 50% finalizados em quatro meses ou menos.

De acordo com a empresa americana Lockheed Martin, através da automação da consolidação dos dados vindos de centenas de sistemas e entidades legais, a empresa do setor aeroespacial

conseguiu reduzir mais de mil posições de trabalho, o que economiza U\$ 40 milhões por ano. Em contrapartida, o investimento inicial foi aproximadamente U\$ 10 milhões, o que garantiu um retorno sobre o investimento (ROI) significativo.

2.6 BPM e a Tecnologia da Informação

O termo *Business Process Management*, e seu acrônimo BPM, podem ter significados diferentes quando analisados sob as ópticas da área de negócios e da área de tecnologia.

Para a área de negócios, como explicitado anteriormente, o termo significa um conjunto de práticas, que permeiam todas as camadas da empresa. É um modelo de condução das atividades da organização. (SCHURTER, 2006)

Para a área técnica, o BPM é um conjunto de ferramentas e tecnologias contidas em uma arquitetura de desenvolvimento de aplicações que permitam a interligação dos diversos sistemas de informação da empresa e melhorem o desempenho de seus negócios. Esse conjunto leva o nome de BPMS (*Business Process Management System*).

Apesar de usarem o mesmo nome, esses dois pontos de vista não necessariamente precisam coexistir. O modelo de gestão BPM não obriga o uso de ferramentas e tecnologias para suas práticas serem executadas. Seu foco principal está em voltar as atividades da empresa para o atendimento das necessidades do cliente.

Porém, segundo Giaglis (2004), os processos de negócios e a tecnologia de informação são parceiros naturais, já que a escolha de um modelo de condução das operações da empresa influenciará na escolha dos sistemas a serem implementados, assim como o desenvolvimento de tecnologias cria novas oportunidades para as empresas.

No modelo funcional, o papel de TI é resolver os problemas de um determinado departamento ou grupo de pessoas que executam a mesma função organizacional, reduzindo os custos daquele grupo em específico, sem se preocupar com as demais aplicações e sistemas da organização. Já no BPM, o foco de tecnologia é na organização como um todo. (ENIX, 2003) Assim, o alinhamento do desenvolvimento dos processos de negócio com o de sistemas de informação é fundamental para que as vantagens da parceria possam ser alcançadas.

3 Modelagem de Processos de Negócios

A modelagem de processos envolve todas as atividades relacionadas à transformação do conhecimento sobre o negócio em modelos que descrevem os processos executados por ela. (GIAGLIS, 2004) Essas atividades consistem em identificar os processos organizacionais com suas entradas e saídas, interligá-los mediante uma representação de processos, identificar os recursos necessários para sua operação e estabelecer indicadores de desempenho para seu monitoramento. (NETTO, 2004)

Para Miers (2004) esses modelos servirão como base para outras atividades, desde o simples conhecimento do negócio, passando por projetos de desenvolvimento de sistemas, até os projetos de remodelagem dos processos, e essa diversidade faz com que seja necessário definir os fins da modelagem antes de ela ser iniciada.

O autor ainda afirma que os modelos são uma forma de abstração da realidade e permitem filtrar complexidades irrelevantes para quem ele se destina. (GIAGLIS, 2004) Assim, sem a definição da finalidade que os modelos terão, podem-se construir modelos extremamente detalhados e complexos ou com nível de detalhamento insuficiente.

Esse problema é recorrente quando os modelos de processos se destinam ao desenvolvimento de sistemas de informação que os forneçam suporte, em que a modelagem serve como ferramenta de comunicação entre os profissionais de TI e analistas de negócio. Os artefatos utilizados por um grupo não são suficientemente detalhados para o outro ou contêm uma linguagem extremamente técnica que os torua incompreensíveis para os outros profissionais.

3.1 Padrões e Notações usados na modelagem

As notações e modelos usados num processo de modelagem decorrem diretamente da metodologia escolhida para orientar esse processo. As notações estão associadas às técnicas que são escolhidas para guiar os trabalhos, enquanto que as metodologias se ligam aos paradigmas escolhidos para a modelagem, como a orientação a objetos ou orientação a dados. (GIAGLIS, 2004)

Os objetivos que os modelos terão, definirão quais ferramentas e metodologias serão usadas na modelagem de processos já que, dependendo da escolha, algumas perspectivas são

privilegiadas em relação a outras. As possíveis perspectivas consideradas numa modelagem são:

- Funcional: representa quais atividades estão sendo executadas
- Comportamento: representa a seqüência das atividades e outras características relativas ao seu fluxo, como laços de execução, iterações, condições e critérios de entrada e saída.
- Organização: informa quem executa as atividades do processo, os meios de comunicação para troca de entidades entre as atividades e os locais de armazenamento de entidades.
- Informação: representa os dados produzidos e manipulados por um processo e seus relacionamentos.

Dentre as ferramentas mais utilizadas na modelagem de processos, destacam-se o fluxograma, o diagrama de fluxo de dados e o modelo entidade-relacionamento.

O fluxograma é muito difundido pela familiaridade com que a maioria dos profissionais, independente da área de atuação, tem com essa ferramenta. As outras duas ferramentas são muito utilizadas na área de tecnologia da informação. (NETTO, 2004)

Alguns autores, como Giaglis (2004) e McGregor (2006a) pregam o uso de mais de uma ferramenta nos processos de modelagem e a criação de modelos com níveis de detalhamento diferentes. Essa abordagem mais heterogênea permitiria uma comunicação mais fácil entre as diversas partes envolvidas na modelagem de processos.

3.2 BPMN (*Business Process Modeling Notation*)

O BPMN é um padrão de modelagem criado para se tornar uma ferramenta que permita a comunicação entre analistas de negócio e técnicos no levantamento de processos. (WHITE, 2005)

O padrão é mantido pelo *Object Management Group* (OMG), que argumenta que os demais padrões de modelagem são sempre focados numa ponta da linha negócio X técnico, o que dificulta a compreensão pelos que estão na outra ponta ou contém informações insuficientes para uma correta análise.

O OMG o mantém como um padrão aberto e seu grupo de trabalho, responsável pela sua definição e manutenção, é formado por membros de diversas empresas e entidades relacionadas ao tema, o que facilita sua adoção e sua aceitação.

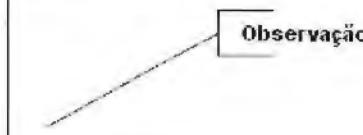
Para alcançar seu objetivo de facilitar a comunicação, a notação utiliza elementos baseados em modelos de fluxogramas, por ser um artefato largamente difundido e compreendido. Os elementos da notação prevêem diversas situações de negócio, como recebimento de mensagens, troca de informações e controle de tempo de tarefas.

A Tabela 2 apresenta e descreve os elementos fundamentais da notação, exibindo também suas representações. Esses elementos estão agrupados em quatro categorias: elementos de fluxo; elementos de conexão; contêineres; artefatos. A Figura 5 mostra um modelo desenvolvido usando alguns dos elementos da BPMN.

Tabela 2 - Principais elementos da notação BPMN (WHITE, 2005)

ELEMENTOS DE FLUXO	
Evento 	Representa acontecimentos no processo e, geralmente, possui um iniciador e um resultado. Os casos básicos são denotados por círculos. Há variações, cujas representações são determinadas por marcadores dentro desses círculos.
Atividade 	Atividade é um termo genérico para um trabalho realizado no processo, podendo ser uma única tarefa ou um conjunto de tarefas.
Gateway 	É um ponto de convergência ou divergência no fluxo do processo, em que uma decisão é tomada ou em que os diversos caminhos do processo são reunidos.

ELEMENTOS DE CONEXÃO	
Seqüência 	Representa a ordem em que as atividades são executadas no fluxo do processo.
Mensagem 	Representa as mensagens trocadas entre os participantes do processo.
Associação 	Permite associar dados, textos e outros artefatos do modelo aos elementos de fluxo.
CONTÊINERES	
Pool 	Representa um participante do processo e permite agrupar as atividades para que não se misturem às atividades pertencentes a outro participante.
Raias 	É uma subdivisão do <i>pool</i> e serve para agrupar logicamente as atividades de um mesmo participante.
ARTEFATOS	
Objeto de dados Name	Não possui qualquer efeito sobre o fluxo das atividades, porém fornece informações sobre requisitos das atividades e em que elas resultam.
Grupo 	Agrupamento lógico de atividades. Não altera o fluxo das atividades.

Observação	Usada para fornecer informações adicionais para quem lê o diagrama.
	

A representação gráfica desta notação é armazenada em formato XML, padrão para troca de dados entre sistemas, que é usado na maioria dos produtos do mercado de BPM. Isso também permite que o padrão de modelagem tenha correspondência direta com linguagens de execução de processos, usadas nos ambientes de execução das ferramentas de BPM apresentadas mais a seguir.

Algumas referências argumentam que a riqueza de detalhes do BPMN e sua tentativa de prever muitas situações de negócio criam um nível de detalhamento que dificulta a aceitação da notação, visto que essa característica eleva a complexidade das ferramentas de modelagem. Essa situação se evidencia no estudo das ferramentas de BPM realizado e exposto mais a seguir, em que muitas ferramentas não implementam a notação por completo, ficando restrita apenas a algumas funcionalidades.

Harrison-Broninski (2006) argumenta ainda que a notação BPMN só serviria para processos em que o envolvimento do usuário só estaria limitado à entrada de dados e a pontos de decisão, não sendo adaptável a processos que envolvam comportamentos criativos e adaptativos.

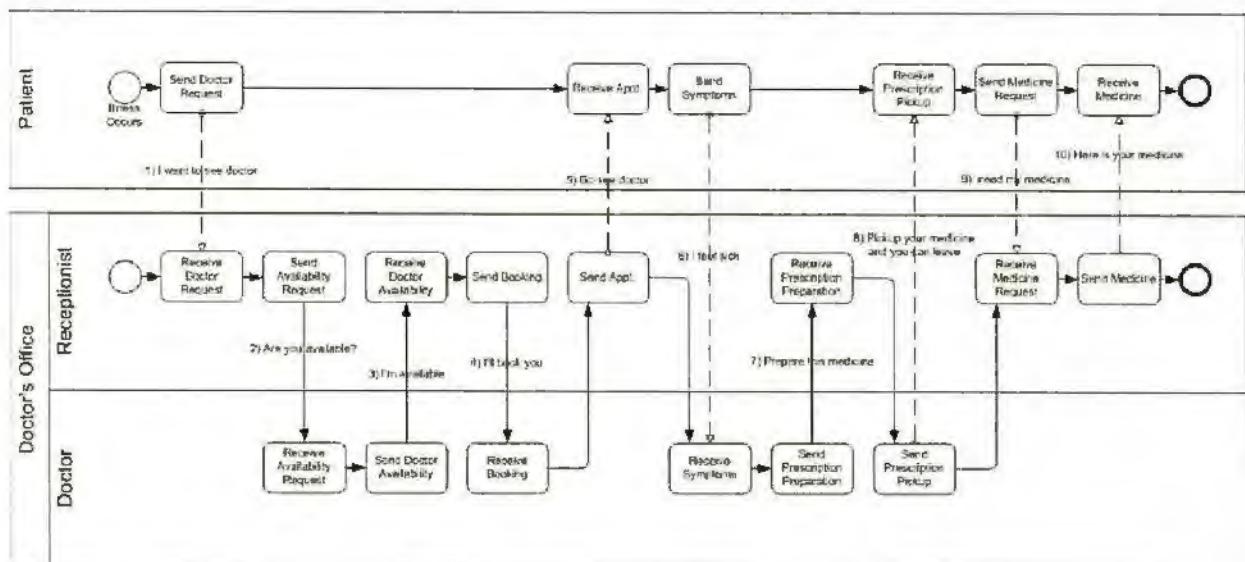


Figura 5 - Modelagem de um processo de um consultório médico (WHITE, 2005)

3.3 Melhores práticas para modelagem

O uso de uma notação completa e que consiga facilitar a comunicação entre usuários de negócio e técnicos auxilia as organizações a alcançarem seus objetivos com relação a seus processos de negócios. (JENSEN, 2003)

Porém essa não é uma condição suficiente para o sucesso dos projetos. É necessário que um método seja seguido para que os processos sejam corretamente mapeados e atendam às expectativas.

O primeiro passo para que o plano de modelagem dos processos de uma empresa seja alcançado é definir os objetivos do modelo que se quer criar e o nível de maturidade da empresa com relação a seus processos (MIERS, 2004). Com um objetivo traçado, é possível definir exatamente quais elementos do processo são fundamentais para a compreensão do modelo, podendo desprezar outros pontos que não agregariam valor ao modelo final. O nível de detalhes contido no modelo também se relaciona com a maturidade da empresa, já que um modelo extremamente detalhado não se aplicaria a uma empresa imatura, assim como um modelo superficial frustraria as expectativas de uma empresa mais amadurecida.

O passo seguinte, que antecede a coleta de dados, é conhecer a estrutura da empresa e saber quem são as pessoas que podem fornecer detalhes sobre os processos. Em organizações com estruturas não voltadas a processos, esses profissionais incluem gerentes, supervisores e especialistas, sendo os primeiros os que possuem uma visão mais ampla das atividades da empresa, enquanto que os últimos possuem informações mais específicas sobre as atividades. Caso a empresa já possua uma estrutura de processos, surge a figura do dono do processo (*process owner*, em inglês), que é o profissional responsável pelo processo e tudo relacionado a ele.

Sabendo quais informações coletar e onde coletá-las, o próximo passo para o levantamento dos processos é a coleta de dados. O meio tradicional de fazer a coleta é através de entrevistas com os profissionais envolvidos com o processo. Outro modo, caso o processo esteja implementado em alguma ferramenta ou software, é análise desse aplicativo, tanto do seu modo de operação como de seu código.

Segundo Miers (2004), só após a passagem por essas etapas é que se pode planejar e implementar alterações no processo analisado, já que informações suficientes já teriam sido coletadas.

A passagem por todas essas etapas não garante que o projeto será bem-sucedido, porém oferece condições para que resultados melhores sejam alcançados. Sem o conhecimento do escopo e das pessoas e sistemas envolvidos no projeto, as atividades ficam prejudicadas e o tempo necessário para executá-las é maior.

4 Tecnologias BPM

A aplicação da tecnologia ao BPM pretende trazer as vantagens da dupla CAD/CAM ao projeto de processos de negócios. O uso de tais ferramentas deve dar suporte a oito capacidades: descoberta, projeto, implantação, execução, monitoramento, interação, controle e análise, que devem ser disponibilizadas em todos os níveis de trabalho. (SMITH e FINGARS, 2003) Essas capacidades dão suporte às etapas do ciclo de otimização de processos apresentado anteriormente, por isso os nomes são semelhantes em alguns casos. Elas são apresentadas com suas descrições na Tabela 3.

Tabela 3 - Capacidades de ferramentas BPM (DUTRA, 2006)

Descoberta	A ferramenta deve dar suporte à descoberta dos processos. Novas tecnologias permitirão também que padrões de trabalho sejam detectados e mapeados.
Projeto	É a capacidade de manipular e reprojetar os processos à medida que a organização evolui, com o uso de repositórios, ferramentas de simulação e gerenciamento.
Implantação	A implantação é a disponibilização dos processos para os participantes (pessoas, sistemas e outros processos), realizando a conexão com os sistemas, e distribuindo e gerenciando as pessoas que têm permissão para executá-lo.
Execução	A execução é a garantia de que um novo processo é recebido por todos os seus participantes. É de responsabilidade do sistema de gerenciamento, que controla o estado em que se encontra conforme é executado. É ele quem implementa as transações, controla o fluxo de dados e correlaciona as diferentes atividades.

Interação	É o uso de interfaces em computadores pessoais ou portais que permitem às pessoas interagirem com o processo. Essa interação varia desde a entrada de dados em formulários, alocação de trabalho e gerenciamento de tarefas.
Monitoramento e controle	Compreendem as atividades focadas no negócio e nas intervenções técnicas necessárias para manter a integridade dos processos e do ambiente. Algumas delas são: identificação e gerenciamento de exceções; <i>upgrade</i> de processos; alteração de participantes.
Análise	É a mensuração de desempenho do processo para fornecer as métricas, análises e inteligência de negócios necessários para derivar estratégias de melhorias, e descoberta de oportunidades para inovação.

A idéia de analisar e modelar uma organização por seus processos não é nova. Iniciativas como o BPR (*Business Process Re-engineering*) já pregavam mudanças nas empresas com base em seus processos, e incluíam ferramentas que davam suporte a essa filosofia, como *workflows* e ERP's.

A inovação do BPM em relação às outras iniciativas de processos de negócio está em sua filosofia de aperfeiçoamento contínuo e cíclico, e suas ferramentas devem lhe dar suporte, aproximando os processos e os recursos relacionados com TI. (LACHAL, 2005)

Além disso, as ferramentas também devem atender a outros dois pontos da metodologia: facilidade de adaptação a novos cenários de negócio, permitindo a rápida alteração dos processos; facilidade de gerenciamento, para que as empresas possam medir seu desempenho em tempo real.

E para atender estes dois pontos, é fundamental que exista a separação do processo e da tecnologia que o implementa e executa. Para isto, os sistemas BPMS geralmente são divididos em algumas camadas, como exibido na Figura 6. Essas camadas são:



Figura 6 - Modelo de camadas do BPMS típico

- **Camada de conectividade baseada em padrões abertos**

Interliga serviços e sistemas já existentes, geralmente através de protocolos de mensagens abertos e conectores dependentes das tecnologias.

- **Camada de aceleração de negócios**

Descreve as regras de negócios e processos de forma independente da tecnologia e sem duplicidade, garantindo o controle e rastreamento da execução dos processos de negócio. Esta camada também contribui para diminuir o esforço para alterar os processos, contribuindo para gerar novos cenários de negócio, maior rapidez na alteração dos processos.

- **Camada de interface operacional**

Separa tudo o que o usuário vê da informação e da lógica que é fornecida pelas outras camadas, minimizando riscos e reduzindo esforços de desenvolvimento.

4.1 Comparação com outras ferramentas

As principais ferramentas de BPM presentes hoje no mercado de tecnologia são propriedades de grandes fornecedores de antigos produtos de *workflow*, EAI e ERP. A ênfase da ferramenta de cada fornecedor acaba dependendo do produto em que a empresa era especializada o que, por vezes, limita o escopo da sua ferramenta e gera falta de clareza na definição do que um sistema BPM deve conter.

O conjunto de funcionalidades que um BPMS apresenta já foi esclarecido anteriormente. A seguir esses sistemas são comparados com os três tipos de ferramentas citados anteriormente: *workflows*, EAI e ERP.

Sistemas de Workflow

Os sistemas de *workflow* são um conjunto de produtos e ferramentas que visam à automação de processos. Originalmente, foram projetados para operar internamente em departamentos, a fim de agilizar o trabalho e garantir sua qualidade, através do mapeamento das atividades de cada pessoa envolvida e facilitando sua execução com o uso de formulários. (DUTRA, 2006)

Os BPMS também incluem essa funcionalidade, porém seu escopo e sua abrangência são maiores. Tal ferramenta conta com um módulo de automatização das atividades de processos, porém seu principal fim é a gerência desses processos. Os BPMS também não se restringem aos departamentos, mas abrangem a organização como um todo, interligando e gerenciando as atividades de diversos setores e sistemas.

Os BPMS ainda têm por característica o uso de padrões para modelagem e execução de seus processos, enquanto que nos *workflows* utilizam tecnologias proprietárias. Assim, teoricamente, um processo desenvolvido numa ferramenta BPM poderia ser executado em outra. Essa flexibilidade ainda é limitada pela aderência BPMS's aos padrões, porém essa é uma tendência devido à participação de grandes empresas na elaboração desses padrões.

ERP

Os sistemas ERP surgiram como proposta para a automação e integração de processos. Esses sistemas costumam contar com processos pré-modelados, comuns à maioria das categorias das empresas, envolvendo e integrando áreas de finanças, vendas, cadeia de suprimentos, recursos humanos, entre outras. (DUTRA, 2006)

A grande vantagem trazida pelos ERP's foi a padronização dos dados por toda empresa, já que todo o fluxo é controlado por um único sistema que garante a coerência. Porém, junto com a padronização dos processos, os ERP's trazem com sua implementação a necessidade de adaptar os processos da empresa aos seus modelos.

Além disso, seus modelos são pouco flexíveis o que, por vezes, limita a capacidade de adaptação das empresas a novos cenários e necessidades do mercado.

Os BPMS's não exigem que a empresa altere seus processos para que seja implementado e ainda permitem que a organização altere seus processos conforme suas necessidades. Porém nem todos os processos são suportados pelos BPMS.

Khan (2006) sustenta a tese de que há processos que devem ser mantidos dentro dos ERP's, que seriam mais rígidos e que não necessitam de mudanças, enquanto que os processos com mais chances de serem alterados, sejam suportados pelos BPMS.

EAI (*Enterprise Application Integration*)

As ferramentas EAI, acrônimo em inglês de Integração de Aplicações Corporativas, surgiram com o intuito de integrar os diversos departamentos e sistemas de uma empresa, através da criação de uma camada intermediária.

Acredita-se que com tal camada os custos de manutenção dos sistemas legados possam ser reduzidos e a necessidade de criação de novos sistemas seja diminuída, já que os sistemas passariam a ser interoperáveis.

Devido ao conjunto de funcionalidades de um BPMS, há a tendência de definir os sistemas para BPM como a integração de sistemas de *workflow* e EAI, visto que a união das características de cada uma dessas ferramentas resultaria nas principais funcionalidades de um BPMS, ou seja, a capacidade de execução e administração de processos interligando os diversos componentes envolvidos.

4.2 Principais ferramentas e fornecedores

Antes de relacionar as ferramentas existentes no mercado, devem-se esclarecer algumas diferenças entre os conceitos de linguagens, servidores, ferramentas e utilitários, suítes BPM, plataformas e aplicações completas, que podem ser vistos na Figura 7, em que também é possível enxergar como esses itens se relacionam.

As linguagens são os artefatos mais flexíveis e multi-propósito que existem, servem para a criação de qualquer solução, porém exigem alto investimento em tempo de projeto. Qualquer empresa poderia criar seus aplicativos com o uso de uma linguagem, mas isto se mostra arriscado, caro e pouco eficiente. No outro extremo há as soluções completas de BPM, que já trazem melhores práticas, funções maduras e conhecimento do fabricante. Porém, possuem muitas limitações para que seus modelos funcionem de forma integrada e em todos os casos, dificultando customizações e geralmente têm custos proibitivos para a maioria das empresas.



Figura 7 - Componentes de BPMS (MIERS e HARMON, 2005)

Neste intervalo, estão posicionadas as principais ferramentas de mercado, que buscam um compromisso entre a flexibilidade para customizações e a facilidade de componentes configuráveis. Geralmente pode-se escolher entre linguagens de programação, usar diversos servidores de aplicações e buscar modelos já criados para processos padrões das diversas indústrias.

Geralmente, as suítes BPM buscam atender um conjunto de funcionalidades-chave. São elas:

- **Modelagem de processos de forma gráfica**

A modelagem de processos deve ser baseada em notações padrão, como o BPMN, e fornecer suporte para a simulação como etapa do desenvolvimento de um processo. Também é importante a possibilidade de reuso de sub-processos, criação de formulários para coleta de dados e facilidades para documentação.

- **Servidor de aplicação para execução dos processos**

O servidor é onde os processos são executados. É necessário conhecer sua arquitetura para prever seu comportamento com um pequeno ou grande número de usuários. Também devem ser avaliadas características como acesso a dados e integração com outras soluções.

- **Regras de negócios**

Uma das vantagens do BPM é o gerenciamento facilitado das regras de negócios usadas nos processos. Logo, esta funcionalidade é de extrema importância para que os objetivos de flexibilidade e controle sejam atingidos.

- **Integração**

A capacidade de buscar informações em aplicações externas já é fundamental em sistemas de informação empresariais, o BPM deve ir além e conseguir orquestrar diversos sistemas para que eles respondam de forma integrada e sem barreiras ou duplicações desnecessárias.

- **Estrutura organizacional**

No momento em que processos são documentados e executados em uma ferramenta, é necessário deixar claros os papéis das pessoas envolvidas da organização. O gerenciamento de permissões fica vulnerável, se não existir uma ferramenta sofisticada de modelagem de estrutura organizacional para suportar o acesso das pessoas corretas à informação necessária, e bloquear atentados à segurança da informação.

- **Evolução e flexibilidade de processos**

A possibilidade de alterar os processos para suportar suas evoluções é o que diferencia fundamentalmente os sistemas de BPM dos baseados em *workflow*. À medida que os gargalos são identificados ou novas regulamentações são impostas, os processos precisam ser rapidamente alterados para suportar as mudanças e, ao mesmo tempo, manter compatibilidade e rastreabilidade com versões anteriores.

- **Monitoração**

Muitas suítes estão incorporando soluções de monitoramento de atividades do negócio (*Business Activity Monitoring – BAM*), permitindo ao gestor acompanhar indicadores do negócio *on-line*, para tomar decisões operacionais e estratégicas de forma mais rápida.

- **Modelos**

Quando um cliente compra uma ferramenta para modelar processos, é fundamental que já existam modelos implementados com processos fundamentais de ramos da indústria ou domínios específicos de conhecimento, como contabilidade ou logística. Modelos ajudam a diminuir o prazo e risco da primeira implantação da suíte.

4.2.1 Escolha da ferramenta

Com o objetivo de selecionar uma ferramenta para ser usada como base para o desenvolvimento do ambiente, fez-se um levantamento amplo baseado em critérios fundamentais, seguido da eliminação a partir das dificuldades enfrentadas durante a instalação e execução dos exemplos que as acompanham. Os critérios fundamentais para a seleção foram:

- Ter o código fonte aberto e ser um software livre.
- Ser um ambiente completo com interface com o usuário e persistência implementados de forma nativa.
- Possuir servidor de aplicação e possuir serviço de orquestração de processos já implementado.
- Possibilitar expansão e customizações.

A partir de pesquisas na Internet, em particular nos sites de repositórios de código aberto, tais como SourceForge.net e ObjectWeb.org e em artigos específicos (Miers; Harmon, 2005) localizou-se uma lista de ferramentas, com um resumo da descrição dada pelo próprio fornecedor, que atendiam aos critérios fundamentais:

- **jBoss jBPM**

Ferramenta para *workflow*, BPM e orquestração de serviços que habilita a criação de processos de negócios que coordenam pessoas, aplicações e serviços. Faz uso de uma linguagem de *workflow* própria, o jBPM Process Definition Language (JPDL) que pode ser customizada, e oferece suporte a *Business Process Execution Language* (BPEL).

Possui um bom editor visual integrado ao Eclipse e ao servidor de aplicações JBossAS, o que oferece agilidade no momento à implantação de processos. Possui uma comunidade muito atuante. Oferece suporte à criação de formulários customizados que faça uso de Java Server Faces.

- **Intalio Designer**

Muito semelhante ao jBPM, mas ainda conta com suporte à BPMN 1.0 e BPEL 2.0 de forma nativa. Possui ótima interface gráfica e pouco código é necessário para criar *web services*. No entanto, ainda é muito instável e conta com vários defeitos em sua primeira versão.

- **Fuego 5.5**

Ferramenta vendida pela BEA que não possui código-fonte aberto e custa mais de R\$ 100.000 para uma instalação empresarial. A versão para testes de 30 dias foi usada para fazer *benchmark* de suas funcionalidades.

- **WfMOpen**

Implementação de um mecanismo de *workflow* baseado em Java 2 Enterprise Edition (J2EE) conforme proposto pela WfMC e OMG. É bem documentada e aderente aos padrões XPDL e Xforms. No entanto, não oferece opção de customização da linguagem e a expansão se mostrou complicada.

5 Metodologia do Projeto de Formatura

5.1 Visão Geral

A metodologia adotada para o desenvolvimento das atividades do grupo para o projeto de formatura pode ser descrita em cinco fases:

- Estudo do tema
- Levantamento das necessidades do cliente
- Levantamento dos requisitos da ferramenta
- Desenvolvimento de ferramenta
- Teste da ferramenta

A primeira fase inclui um profundo estudo da BPM, com análise de especificações técnicas, materiais bibliográficos, documentos e casos bem sucedidos de implantação da metodologia. Essa etapa visa o conhecimento da tecnologia, suas limitações, suas capacidades e, principalmente, as melhores práticas que devem ser adotadas em projetos que envolvem a BPM. Além disso, servirá como base para o desenvolvimento nas outras etapas do projeto.

Na segunda fase foram levantados os requisitos reais em conjunto com o cliente, o PECE. Essa entidade tem o interesse em documentar os seus processos para obter um maior controle das atividades realizadas. O grupo participou da modelagem de alguns desses processos buscando identificar e tornar clara as regras de negócios envolvidas em cada atividade. Os resultados desta fase são fundamentais para o desenvolvimento do sistema, de forma a garantir uma aplicação autêntica.

A terceira fase destina-se à consolidação e fixação dos conceitos estudados na primeira etapa com os da segunda a fim de definir os requisitos necessários da ferramenta. Com base nestes dados, será possível a definição de uma ferramenta para desenvolvimento da solução.

Na quarta fase, de desenvolvimento, optou-se por utilizar uma metodologia de desenvolvimento baseada em componentes de forma a tornar customizável a solução. Para isto utilizou-se a metodologia RUP que atende essas características, além de tornar possível a gerência dos requisitos.

Por fim a última fase, de testes, visa à realização de testes de integração e também de aceitação da ferramenta no ambiente do cliente.

5.2 Aplicação da Metodologia

5.2.1 Estudo do tema

Esta etapa do projeto consistiu no estudo do BPM e mais exatamente no estudo da sua viabilidade. O período mais crítico desta fase foi o que antecedeu estes estudos. A avaliação dos possíveis temas para o escopo, bem como sua própria definição, foram fundamentais para que o grupo pudesse dar início aos estudos básicos e assim, continuar o projeto. Reuniões e pesquisas foram determinantes para a boa conclusão desta etapa.

Apesar de terem sido concentradas nesta fase, as atividades de pesquisa e estudo do tema e de seus assuntos relacionados não se limitaram apenas ao início do projeto, tendo prosseguido paralelamente com as fases posteriores.

Uma vez bem definido o escopo e bastante discutido entre os integrantes do grupo, deu-se início a fase de pesquisa dirigida. Com o foco definido, buscou-se o máximo de informações em livros, internet e artigos publicados que pudessem dar uma base teórica para a compreensão do tema e o desenvolvimento das demais atividades.

Dentre as atividades dessa fase, também houve a participação no evento Seminário Executivo BPM, organizado pela empresa Hypermedia Technologies em conjunto com o BPMGroup, entidade britânica que se propõe a difundir o BPM. Esse evento contou com a participação de representantes de empresas brasileiras que implementam ou tem interesse em implementar o BPM de alguma maneira.

Outro ponto importante neste desenvolvimento inicial foram as reuniões internas do grupo e as reuniões com os orientadores. Nesses encontros foi possível apresentar e discutir os resultados e conceitos obtidos, o que contribuiu para fixá-los e, eventualmente, corrigi-los. Como o ciclo de atividades é em espiral, como exibido na Figura 8, foi possível, após uma melhor compreensão do tema, uma volta à fase de definição do escopo, visando o seu aperfeiçoamento.



Figura 8 - Ciclo de desenvolvimento das atividades

Nesta primeira fase, de estudo do tema, também fizeram parte a definição dos requisitos iniciais da ferramenta e o estudo aprofundado das tecnologias. Com base nos requisitos iniciais, foi possível desenvolver uma pesquisa detalhada sobre as ferramentas disponíveis e, deste modo, rever o escopo.

5.2.2 Levantamento das necessidades do cliente

Nesta segunda etapa, foram desenvolvidas atividades com o PECE (cliente) a fim de definir os requisitos que a ferramenta deveria suprir para sua devida aplicação. Houve um estudo das atividades diárias do PECE para que fosse possível a identificação das regras de negócios e fosse possível identificar os processos e assim defini-los. Os pontos críticos e relevantes desta etapa foram a modelagem em si e a identificação das atividades em que se poderia aplicar uma automação. Vale ressaltar que houve bastante ajuda e colaboração por parte do cliente nesta etapa fundamental à fase seguinte de levantamento de requisitos da ferramenta.

5.2.3 Levantamento dos requisitos da ferramenta

Nesta etapa, foram estabelecidos os requisitos referentes à ferramenta. Nesta categoria, incluem-se os requisitos funcionais e não funcionais, assim como os requisitos levantados com os usuários. Os pontos críticos desta etapa foram o próprio levantamento dos requisitos e, principalmente, a definição da tecnologia principal do produto.

No processo de escolha da ferramenta, após a instalação e execução com sucessos dos testes que as acompanham, todas as ferramentas ofereceram condições de implementação de fluxos variados de trabalho, inclusive os pretendidos nesse projeto. Mas o jBPM foi o que atendeu a todos os critérios fundamentais de seleção da ferramenta citados anteriormente e ainda implementava dois requisitos funcionais do projeto. Optou-se então pelo jBPM.

Assim, foram definidas a arquitetura da ferramenta, sua linguagem e sua plataforma, com base nos requisitos e também se utilizando os conceitos adquiridos nos estudos prévios.

5.2.4 Desenvolvimento da ferramenta

Nesta etapa verificou-se o amadurecimento dos requisitos através da definição das camadas e funcionalidades da ferramenta assim como na melhor divisão para o desenvolvimento da mesma. Priorizaram-se alguns requisitos frente a outros e definiu-se um cronograma, paralelo ao do projeto, para o desenvolvimento e prototipação da interface.

Optou-se por entregar duas versões parciais do ambiente, para que o trabalho de modelagem e a primeira implementação dos processos pudessem ser realizados paralelamente ao desenvolvimento da segunda etapa do software. Pode-se analisar este resultado na Tabela 4.

Tabela 4 - Versões da ferramenta

Versão 1.0		
ID	Tarefa	Fim aprox
101	Autenticação	21/8/2006
102	Esqueci minha senha	16/10/2006
103	Autorização	25/10/2006
104	Mostrar somente os processos que o perfil pode iniciar / Pool	8/10/2006
105	Cadastro, alteração e exclusão de usuários para o administrador.	25/10/2006
106	Controle de grupos/perfis de usuários	25/10/2006
107	Envio de e-mail (componente, código padrão e manual)	25/10/2006
108	Documentação do funcionamento de autorizações, grupos e perfis	20/10/2006

Entrega prevista desta versão: 25/10/2006

Versão 2.0		
ID	Tarefa	Fim aprox
201	Código para conexão com web services e exemplo	5/12/2006
202	Interface para consulta de desempenho dos processos com filtros (BAM V1.0)	25/11/2006
203	Ferramenta para auxílio à criação de formulários	-
204	Design gráfico do portal e estilos css	21/8/2006
205	Documentação V1 (Formato padrão, versionamento e índice)	31/11/2006
206	Alterações básicas na interface do IDE (Plug-in do eclipse)	Cancelada

Entrega prevista desta versão: 5/12/2006

Desta forma, foi possível ajustar a segunda versão em relação às necessidades e problemas identificados durante a utilização da primeira versão do ambiente. Estes problemas foram os pontos críticos desta fase, pois limitaram o escopo da ferramenta final.

5.2.5 Testes da ferramenta

Nesta última etapa, o foco principal foi o teste final e a aceitação da ferramenta no ambiente do cliente. Para isso, foi necessário o levantamento da infra-estrutura necessária à instalação do software, assim como a programação dos módulos e os testes unitários e de integração. Como resultado dessa fase, foi gerado um manual de usuário e de instalação da ferramenta final, presentes no Anexo 6.

5.3 Cronograma

As atividades de estudo, desenvolvimento, pesquisa, documentação seguiram um cronograma, definido no início do projeto e aprovado junto aos *stakeholders*. Foi produzido um cronograma, presente no Anexo 8 – Cronograma de Atividades com as atividades distribuídas ao longo do projeto, assim como um WBS, presente no Anexo 7 contendo todas as atividades melhor detalhadas e divididas conforme o tema referente.

6 Desenvolvimento de Ferramenta

6.1 Requisitos da Ferramenta

6.1.1 Escopo

A ferramenta visa fornecer um ambiente que disponibilize as funções de modelagem, desenvolvimento, execução e análise de desempenho de processos de negócios orientados fortemente para interação com pessoas, mas também provendo a integração com aplicações e serviços externos ao ambiente. Para isto, utiliza-se uma notação de modelagem própria, o *jBPM Process Definition Language* (jPDL) que pode ser customizada e também oferece suporte a *Business Process Execution Language* (BPEL). Para que isso fosse possível, também foi necessária a configuração do ambiente em um servidor de aplicações com acesso a algum banco de dados.

Grande parte dos benefícios que esta ferramenta traz está ligada ao fato de ela ser baseada em código-fonte aberto e em *software* livre, além da implementação de sua interface ser um *plug-in* do Eclipse, que é um padrão de mercado para desenvolvimento Java.

6.1.2 Interface com o Sistema e Usuário

O sistema dispõe de duas interfaces com os usuários distintas: *plug-in* do Eclipse para modelagem e desenvolvimento de processos; e um portal para execução e análise de desempenho dos processos. A primeira é desenhada para o desenvolvedor de TI e para o analista de negócios, que juntos devem modelar o processo e posteriormente o técnico inclui detalhes como formulários e interfaces com outros sistemas.

Já a segunda é voltada para os usuários e para os donos dos processos, por isso é baseada na tecnologia de portal para integrar diversos serviços de forma simples e intuitiva.

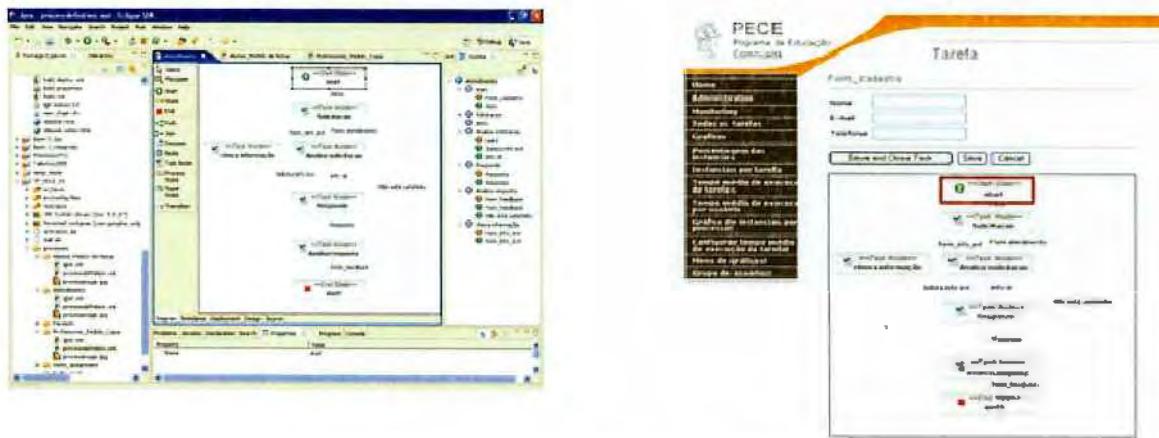


Figura 9 – Interfaces com o desenvolvedor e com o usuário

6.1.3 Interfaces de Hardware e Software

As interfaces de hardware e software consistem basicamente nas configurações que o sistema deve ter e nas ferramentas auxiliares. Esta interface pode ser dividida em duas. A primeira, relativa à fase de desenvolvimento da ferramenta, onde o hardware deve ter configurações tais que permitam o desenvolvimento assim como os softwares necessários a este desenvolvimento. A segunda, relativa ao usuário final, especifica quais configurações de hardware e software são necessárias para que o sistema opere normalmente.

Foram gerados dois documentos para essas duas divisões da interface hardware/software, sendo que para a primeira, foi apresentada uma listagem dos softwares necessários ao desenvolvimento assim como uma breve explicação sobre eles. Além disso, há no Anexo 3 uma listagem das configurações de hardwares necessárias. Para a segunda ferramenta, há no Anexo 4 uma listagem e detalhamento das ferramentas auxiliares que rodam em *background* assim como um manual de usuário contendo todas as especificações de hardware.

6.1.4 Requisitos Funcionais

RF 1 - Desenho de processos em ferramenta gráfica e via XML

Uma das etapas mais importantes do BPM é a modelagem do processo, de forma que este seja compatível com a realidade de negócios e com a infra-estrutura de TI. Para isso, a ferramenta deve disponibilizar interfaces gráficas do tipo arrastar-e-soltar e acesso ao código criado.

RF 2 - Execução de processos

Esta etapa é quando o usuário interage com o processo em seu dia-a-dia de trabalho, é muito importante que a ferramenta tenha um tempo de resposta padrão *web*, ou seja, menos de 5 segundos para carregar uma página. Também é importante suportar o usuário em caso de falhas, indisponibilidade do sistema ou dúvidas. E por último o fundamental, executar os processos de forma correta e eficiente.

RF 3 - Criação facilitada de formulários para coleta de dados dos processos

O esforço para a diminuição do tempo de implementação médio de novos processos resultou na criação de uma ferramenta para automatizar a criação de formulários simples. Esta ferramenta deve receber a descrição dos campos do formulário e suas variáveis associadas, e gerar o formulário automaticamente.

RF 4 - Envio de *e-mail*, de acordo com eventos do processo

O envio de *e-mail* durante o processo tem o objetivo de avisar os usuários que existe uma tarefa pendente no sistema, ou também pode ser usado para avisar aos supervisores que uma meta de execução de um processo terá seu limite ultrapassado.

RF 5 - Acompanhamento do desempenho dos processos - relatórios e gráficos

A otimização dos processos só acontece porque existe uma análise dos índices que determinam a situação atual dos processos e fornece dados para a evolução destes.

RF 6 - Interface com outros aplicativos via interface padrão *web services*

Opção de tecnologia para integração de aplicações, possibilitando interoperabilidade.

6.1.5 Requisitos Não Funcionais

RNF 1 - Segurança: Autenticação e Autorização

O sistema deve autenticar seus usuários para o acesso a qualquer página do *site*.

O sistema deve verificar se a permissão do usuário é suficiente para acessar o conteúdo requisitado.

RNF 2 - Manutenção: Qualidade da documentação

O sistema deve disponibilizar os seguintes documentos para sua manutenção futura:

- Manual do Usuário
- Manual de Instalação
- Manual do Desenvolvedor
- Descrição de casos de uso
- Descrição da Arquitetura
- Detalhamento de testes de integração e aceitação

6.2 Arquitetura

6.2.1 Introdução

O desenvolvimento de sistemas para o mundo corporativo requer a utilização de plataformas e ambientes de execução que forneçam segurança, disponibilidade e escalabilidade. Tais requisitos são essenciais para corporações que necessitam oferecer serviços de acesso rápido a grandes comunidades de clientes, fornecedores e colaboradores, utilizando a *Internet / Intranet* como canais de comunicação. Portais B2C e B2B de comércio eletrônico, além de diversos sistemas corporativos, são exemplos destas *Enterprise-class Applications*.

Tais plataformas devem seguir um paradigma de construção de aplicações baseadas em arquiteturas distribuídas e multicamadas. Em outras palavras, envolve a migração de arquiteturas cliente/servidor tradicionais para arquiteturas que forneçam serviços transparentes de distribuição, persistência, transação, e que estejam estruturadas em três ou mais camadas (*3-tier / N-tier*). Neste paradigma, entra em cena a modelagem de componentes de negócio, geralmente presentes na camada intermediária (*middle tier*).

6.2.2 Tecnologia: Java & Componentes

A linguagem Java possui uma série de características importantes que foram desenvolvidas ao longo de sua evolução. Entre elas, pode-se citar sua grande robustez, desempenho adequado

para aplicações *server-side* (aplicações que executam em um servidor, geralmente realizando processamento crítico) e segurança. Além disso, possui um conjunto bastante completo de interfaces de programação, com ênfase no suporte a comunicação em rede e distribuição.

Além das características e vantagens técnicas nativas, Java torna-se poderosa para a modelagem de objetos de negócio a partir dos recursos fornecidos pelo paradigma da orientação a objetos. De fato, o encapsulamento cria a oportunidade de existirem objetos que acompanham a evolução dos processos e objetivos de negócio de forma mais transparente, facilitando a manutenção. E este comportamento atualmente tem sido uma das metas da indústria de software.

Desta forma, torna-se fácil perceber que o encapsulamento da lógica e processos, entre outros exemplos, em objetos, tende a se concretizar em um contexto maior: o desenvolvimento baseado em componentes (*CBD - Component-Based Development*). Assim, tais componentes formam peças de montagem de aplicações corporativas complexas, executando tarefas específicas e sendo customizáveis sem alterações internas.

6.2.3 Camadas do Sistema

Camada Cliente

A camada cliente é o acesso de onde o usuário poderá realizar o acesso ao sistema. Basicamente, a camada cliente é formada pelo *browser* do cliente. Faz parte também dessa camada as aplicações *applets* que são rodadas nos clientes e das aplicações *desktop*, que é outra opção para se conectar no sistema. Há a possibilidade de ter também nessa camada outras aplicações que podem se conectar e utilizar componentes do sistema, no estilo *B2B (Business To Business)*.

Camada Apresentação

Como a aplicação é multicamada, o objetivo da camada de apresentação é de expor a lógica de negócios ao usuário e possibilitar a interação do usuário com o sistema. Essa camada é popularmente chamada de camada *web*.

Existem alguns problemas com relação à aplicação em camadas, como problemas relacionados com a alteração da regra de negócio, que necessita que a camada de apresentação inteira precise ser alterada.

Contudo, apenas a divisão em camadas não basta para solucionar estes desafios. Para que isto ocorra, é preciso também que a camada de apresentação seja limpa, isto é, o fluxo de controle e a chamada dos métodos de negócios são separados da visão, ou seja, não deve possuir mais código do que o necessário para chamar a camada de negócios. Para garantir que a camada seja limpa, foi utilizado o padrão de arquitetura *Model View Controller* (MVC).

O modelo MVC divide os elementos da camada de apresentação em três tipos:

- **Controlador (Controller)** - encarregada de receber a entrada do usuário, acessar a camada de negócios para manipular o modelo e selecionar a visão;
- **Modelo (Model)** - um objeto representando uma parte do domínio e que provê os dados que a visão vai exibir. É o contrato entre o controlador e a visão;
- **Visão (View)** - encarregada de exibir os dados do modelo para o usuário.

A Figura 10 mostra o fluxo do MVC que é utilizado na aplicação com algumas modificações:

- A visão notifica o controlador: Ocorre quando um formulário em uma aplicação web é submetido;
- O controlador acessa a lógica de negócios e atualiza o modelo. Corresponde a chamada de um método em um JavaBean;
- O controlador seleciona a visão mais adequada, passando para ela o estado do modelo que ela deve exibir. O controlador pode, por exemplo, selecionar uma visão caso a operação tenha sido bem sucedida ou outra visão em caso de erro.
- A visão consulta o estado do modelo e exibe as informações ao usuário. Por exemplo, a visão exibe as mensagens postadas de um processo em execução;
- O estado do modelo é alterado. O modelo dispara um evento e a visão se atualiza. Por exemplo, quando uma tarefa de um processo é realizado, o sistema envia uma mensagem ao próximo executor da tarefa.

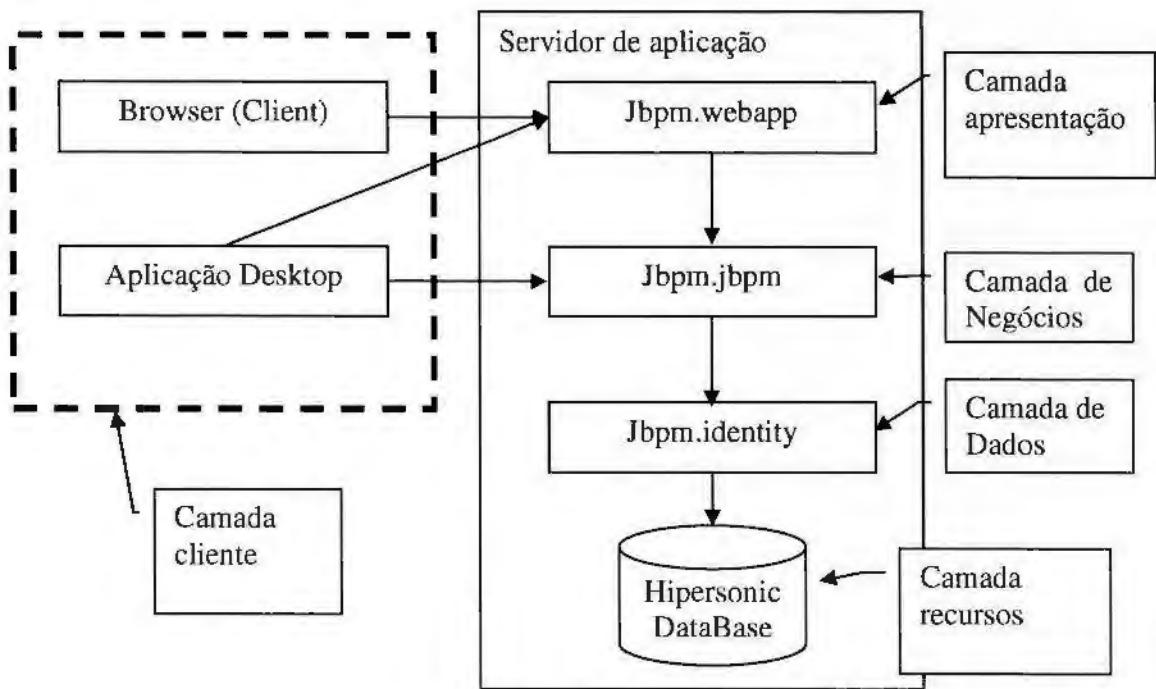


Figura 10 - Fluxo do Modelo MVC

Desta forma, para poder realizar a implementação do MVC, foram utilizados alguns *frameworks web*.

Atualmente, existem cerca de 50 *frameworks web* gratuitos disponíveis no mercado. Analisando as características técnicas dos mais recomendados e mais utilizados no mercado, foram selecionados alguns *frameworks web*.

Assim, para poder utilizar essa modelo de arquitetura nessa camada, foram utilizados os *frameworks JSF (Java Server Faces)* e Struts.

Camada de Negócio

O desenvolvimento da camada de negócio está baseado em componentes (CBD - *Component-Based Development*) que formam peças de montagem de aplicações corporativas para execução de tarefas específicas sendo customizáveis sem muitas alterações internas. Da mesma forma, o sistema precisa ser eficaz e portável - plataformas de execução que dêem suporte a modelos de componentes padronizados e que seja de ampla aceitação na indústria de software.

Desta forma, na arquitetura do sistema, foi utilizado o modelo de componentes de Java por apresentar todas as características de um objeto, tais como: encapsulamento, Alta Coesão, Baixo acoplamento, Polimorfismo, Responsabilidade, Persistência e abstração.

Assim, dentre os modelos de componentes de Java, poder-se-ia utilizar um dos dois modelos existentes nessas plataformas: JavaBean, e Enterprise JavaBean.

Como a aplicação é distribuída e não apresenta grandes volumes de transações e/ou de missão crítica, o JavaBean foi selecionado para poder ser o componente da nossa aplicação.

Da mesma maneira, JavaBean foi escolhida também pelo fato de ser mais leve e mais simples de trabalhar. Como o sistema apresenta interface pela *web* e apresenta componentes gráficos, o JavaBean é ideal para a criação destes componentes gráficos (botões, caixa de texto, etc.). E como o sistema não é crítico com relação à regra de negócio e transações, a aplicação utiliza também o JavaBean para poder realizar algumas regras de negócios.

Mas caso ocorra alguma necessidade de se utilizar os tradicionais EJB (Enterprise JavaBean), o sistema já está pronto para poder realizar essas alterações.

Camada de Dados

Para realizar a camada de integração, foi utilizado o Hibernate e uma coleção de classes de apoio presentes no pacote Jbpm.db para fazer a integração com o banco de dados. Nessa camada apresenta também ligação com o JMS para realizar a comunicação com um servidor de filas. É nessa camada que poderia se programar funcionalidades para que se possa conectar *web services*, sistemas legados e até funcionalidade para se possa conectar sistemas ERP (*Enterprise Resource Planning*). No mercado existem, por exemplo, o JavaConnector para poder realizar a integração do sistema com um SAP. Também para a indústria, pode-se realizar a integração com sistemas de laboratórios LIMS (*Laboratory Information Management System*) ou sistemas de informações de processos (*Process Information Management System*).

Camada de Recursos

Na camada de recursos foi utilizado o banco de dados relacional Hypersonic SQL que foi totalmente desenvolvido em Java e já está presente junto com o servidor de aplicação jBOSS. Apesar de se utilizar o HSQLDB (Hypersonic SQL Database), outros bancos de dados mais conhecidos no mercado poderiam ter sido utilizados. Como a aplicação não demandava uma carga crítica de informações, o banco de dados poderia ser de pequeno porte e como o HSQLDB já estava integrado no servidor, foi a melhor opção para banco de dados.

6.2.4 Projeto

Esse item apresenta uma descrição das principais classes utilizadas no desenvolvimento da ferramenta. Uma lista de outras classes que também foram utilizadas, porém com menos freqüência, está presente no **Erro! Fonte de referência não encontrada.**,

java.identity

Pacote que estabelece e define as entidades, grupos, relações entre grupos e suas hierarquias. Além disso, estabelece e cria usuários do sistema definindo o valor de seus atributos, a qual grupo ele irá pertencer entre outros. As principais classes presentes neste pacote são a Entity.java, Goup.java, Membership.java, User.java, todas do pacote java.org.identity; e, a classe Permission.java (do pacote java.security). A classe Entity.java é responsável atribuição das permissões de cada usuário de acordo com cada grupo de usuários. A Group.java é responsável por criar os tipos de grupos de acordo com os grupos de usuários existentes e por definir relações de parentesco dentro de cada grupo. A classe Menmbership.java por sua vez, dado um usuário, define o papel que ele desempenhará e em qual grupo de usuários ele pertencerá. Por fim, a classe User.java, define os atributos dos usuários como senha e email.

O relacionamento entre essas diferentes classes pode ser mais bem analisado no diagrama a seguir.

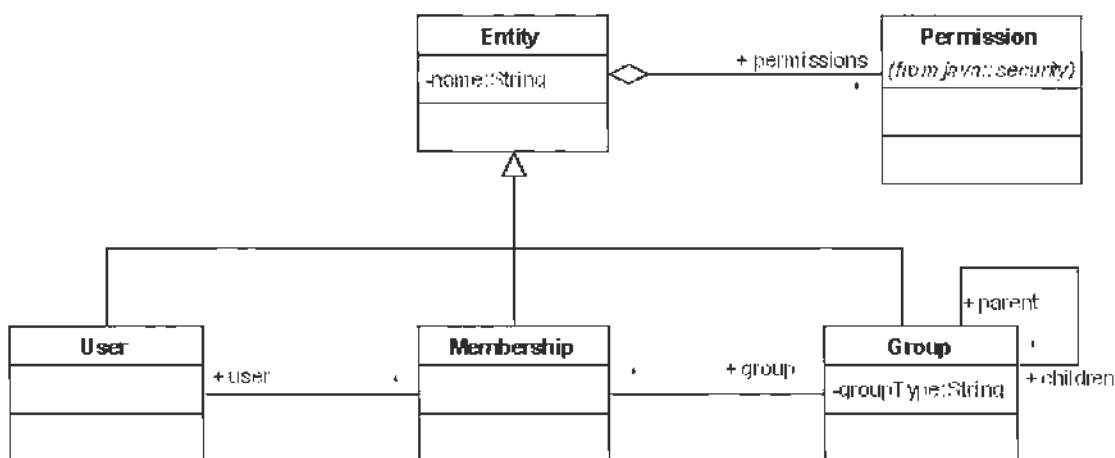


Figura 11 - Relacionamento entre as classes utilizadas

O pacote `java.jbpm` é o pacote principal do jbpm. Através desse pacote que se realizam praticamente as funcionalidades principais da aplicação. É nesse pacote que se encontra o “core” da solução. Esse pacote concentra também a central de acesso ao banco de dados jbpm e outros serviços.

Java.jbpm.security

Pacote responsável por realizar a autenticação e autorização do sistema. É através desse pacote que pode-se desenvolver módulos para realizar a autenticação e autorização com base em banco de dados, arquivos, ou até mesmo no LDAP (Lightweight Directory Access Protocol).

Java.jbpm.calendar

Pacote responsável pelo calendário de negócio do sistema, ou seja, é esse pacote que sabe a hora de negócio do sistema, ou o “horário comercial” que será considerado no sistema. É através deste pacote que poderá configurar a quantidade de dias, horas, meses do “calendário de negócios”. Através do arquivo `jbpm.business.calendar.properties` é possível alterar as configurações da aplicação, desde feriados, horas e minutos dos dias válidos: Assim, a Tabela 5 exibe as possíveis configurações e seus comentários.

Tabela 5 - Configurações para implementação de calendário de negócios

Linha do código	Descrição detalhada
<code>Hour.format=HH:mm</code>	Formato de data
<code>#weekday ::= [<daypart> [& <daypart>]*]</code>	Representa o formato de representação do dia
<code>#daypart ::= <start-hour>-<to-hour></code>	Representa o formato de representação da hora
<code>#start-hour and to-hour must be in the hour.format</code>	
<code>#dayparts have to be ordered</code>	
<code>weekday.monday= 9:00-12:00 & 12:30-17:00</code>	Horário comercial dia
<code>weekday.tuesday= 9:00-12:00 & 12:30-17:00</code>	Horário comercial dia
<code>weekday.wednesday= 9:00-12:00 & 12:30-17:00</code>	Horário comercial dia
<code>weekday.thursday= 9:00-12:00 & 12:30-17:00</code>	Horário comercial dia
<code>weekday.friday= 9:00-12:00 & 12:30-17:00</code>	Horário comercial dia
<code>weekday.saturday=</code>	Horário comercial dia
<code>weekday.sunday=</code>	Horário comercial dia
<code>day.format=dd/MM/yyyy</code>	Formato do dia
<code># holiday syntax: <holiday></code>	Formato dos feriados
<code># holiday period syntax: <start-day>-<end-day></code>	
<code># below are the belgian official holidays</code>	
<code>holiday.1= 01/01/2005 # nieuwjaar</code>	Feriado
<code>holiday.2= 27/3/2005 # pasen</code>	Feriado

<code>holiday.3= 28/3/2005 # paasmaandag</code>	Feriado
<code>holiday.4= 1/5/2005 # feest van de arbeid</code>	Feriado
<code>holiday.5= 5/5/2005 # hemelvaart</code>	Feriado
<code>holiday.6= 15/5/2005 # pinksteren</code>	Feriado
<code>holiday.7= 16/5/2005 # pinkstermaandag</code>	Feriado
<code>holiday.8= 21/7/2005 # my birthday</code>	Feriado
<code>holiday.9= 15/8/2005 # moederkesdag</code>	Feriado
<code>holiday.10= 1/11/2005 # allerheiligen</code>	Feriado
<code>holiday.11= 11/11/2005 # wapenstilstand</code>	Feriado
<code>holiday.12= 25/12/2005 # kerstmis</code>	Feriado
<code>business.day.expressed.in.hours= 8</code>	Qnt de horas por dia
<code>business.week.expressed.in.hours= 40</code>	Qnt de horas por semana
<code>business.month.expressed.in.business.days= 21</code>	Dias do mês
<code>business.year.expressed.in.business.days= 220</code>	Dias no ano

JbpmJava.jbpm.db

Esse pacote serve para realizar as principais tarefas de persistência ao banco de dados, utilizando transações, querys . Através do arquivo hibernate.queries.hbm.xml, é possível implementar querys EJB-SQL para realizar a busca avançada de dados do banco de dados.

java.jbpm.jpdl

Através do JPDL (jBPM Process Definition Language), que é definido a linguagem de processo para o jBPM. Ou seja, especifica um schema XML e um mecanismo para guardar todas as definições de processo em um arquivo de processo.

O arquivo de processo é um arquivo ZIP. Neste arquivo, o arquivo principal é o “processdefinition.xml”. A principal informação no arquivo é o gráfico de processo. Ele contém também informações sobre as ações e tarefas. Detalhes do deploying dos arquivos de processos serão apresentados no item xxx.

Assim, existem basicamente alguns pacotes dando suporte a essa funcionalidade e estão detalhados na Tabela 6.

Tabela 6 – Pacotes da linguagem de modelagem

Pacote	Descrição
<code>org.jbpm.jpdl.convert</code>	Conversão de jpdl 2.0 para jpdl 3.0.
<code>org.jbpm.jpdl.el</code>	Elemento primário de processo
<code>org.jbpm.jpdl.el.impl</code>	Implementação dos elementos
<code>org.jbpm.jpdl.el.parser</code>	Traduz em elementos em componentes

org.jbpm.jpdl.exe	Estrutura de dados jpdl para execução
org.jbpm.jpdl.par	Empacotando os arquivos de processos
org.jbpm.jpdl.xml	Traduz os arquivos XML em elementos

Java.jbpm.web

É nesse pacote que se inicia a execução do sistema. Através do Servlet **JbpmThreadsServlet** ocorre a chamada das três principais classes que irão executar a aplicação.

JbpmConfiguration – Realiza a configuração geral da aplicação. Através dessa classe é possível realizar

CommandExecutorThread – Thread para esperar as mensagens e executar os comandos da aplicação jbpm. Faz parte do pacote org.jbpm.msg.command e é onde o servidor de aplicação fica esperando mensagens da aplicação. Nessa classe apresenta o comando executeCommand() que fica verificando constantemente se chegou mensgen

SchednlerThread – Thread para “controlar” o calendário e suas execuções. Faz parte do pacote org.jbpm.scheduler.impl e serve para gerenciar o horário comercial da aplicação e sempre fica verificando as tarefas que estão pendentes e controlando as tarefas dos processos que apresentam tempo limite para a execução.

Java.jbpm.context

Contém a classe ContextInstance, a classe de interface central para trabalhar com as variáveis de processo.

Assim, existem basicamente alguns pacotes dando suporte a essa funcionalidade e estão detalhados na Tabela 7.

Tabela 7 - Pacotes de suporte para a interface

Pacote	Descrição
org.jbpm.context.def	
org.jbpm.context.exe	
org.jbpm.context.exe.converter	Implementações de Converter.
org.jbpm.context.exe.matcher	
org.jbpm.context.exe.variableinstance	Implementações das instâncias das variáveis

org.jbpm.context.log	Loga as informações das variáveis dos processos.
org.jbpm.context.log.variableinstance	variable instance update logs.

O pacote representa a execução de um comando das tarefas. Através dele é que são persistidos os objetos referentes a execução de cada tarefa do processo.

Java.webapp

Nesse pacote se encontra praticamente as regras de negócios para a camada de apresentação, constando-se basicamente dos JavaBeans que, através do JSF, servem para realizar a lógica de negócio com o core da aplicação.

6.2.4.1 Configuração

Para a configuração do sistema para realização do deploy, são necessários alguns procedimentos básicos. Na pasta src/config-files, ficam os arquivos de configuração para o hibernate, log4j, jbpm e da JVM (Java Virtual Machine).

Log4j.properties – Serve para realizar a configuração do componente log4j. É através desse arquivo que se pode alterar o nível de informações que deverão ser logadas no sistema. É através dele que podemos, também, configurar quais as informações e como deverão aparecer. Caso seja necessário logar todas as informações em um banco de dados relacional, poderemos realizar a configuração diretamente no arquivo.

Hibernate.cfg.xml – Arquivo principal de configuração das características do hibernate. Fazemos nele a configuração de qual banco de dados será utilizado e quais serão as classes que serão mapeadas em banco de dados. Através de tag *mapping-resource* (exemplo: `<mapping resource = "org/jbpm/graph/def/ProcessDefinition.hbm.xml" />`) devemos dizer quais são os arquivos que configuraram todos os elementos que deverão ser persistidos.

Ehcache.xml – Como poderosa característica nos frameworks de persistência é a habilidade para cachear dados e evitar constantes viagens ao banco de dados. Assim, o arquivo serve para configurar o Caching por classe ou coleção-por-coleção na JVM ou ainda em nível de cluster. Isso é feito porque o objeto Session do Hibernate é um cache em nível de transação de persistência de dados e não controla aquelas características.

Jbpm.cfg.xml – Serve para realizar a configuração da aplicação jbpm. A configuração padrão fica na pasta org/jbpm/default.jbpm.cfg.xml e as configurações podem ser sobreescritas por colocar esse arquivo no diretório raiz do classpath e customizar os valores desejados.

6.2.4.2 Deploy

Para realizar o deploy da aplicação é necessário apenas abrir o arquivo jbpm.deploy.xml pela ferramenta de ANT e executar o script correspondente. A Figura 12 exibe a tela do Eclipse correspondente.

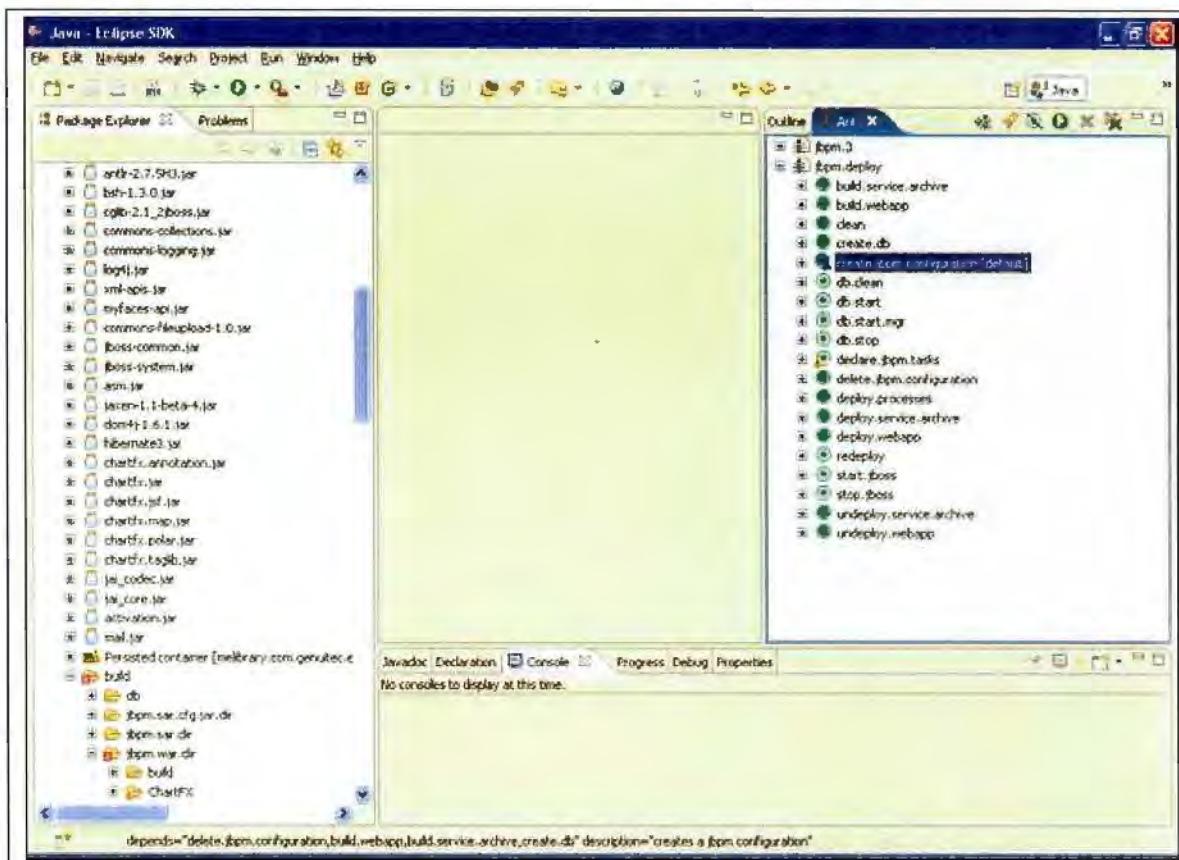


Figura 12 - Tela de compilação da ferramenta Eclipse

Após realizar o deploy da aplicação, é gerado o pacote jbpm.WAR, que basta ser colocado no servidor de aplicação, na pasta deploy, que a aplicação já estará instalada.

Para o deploy, ocorre um série de atividades que o script Ant realizará, exibidas na Figura 13

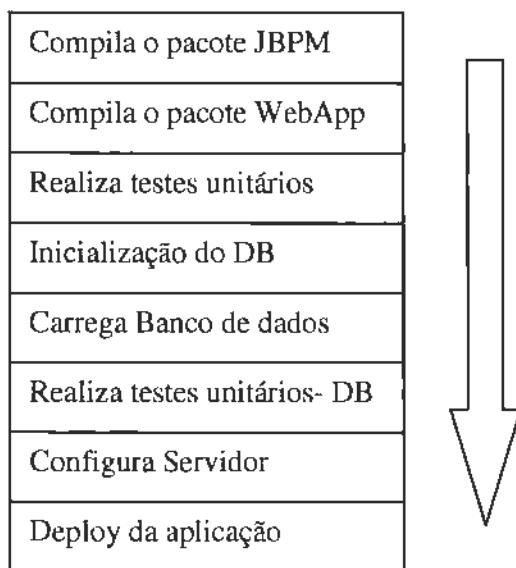


Figura 13 - Processo de compilação

Assim, caso ocorra algum problema em uma das tarefas de nível superior, o script para de executar.

6.2.4.3 Persistindo novos elementos no banco

Caso se precise alterar atributos, ou ainda, criar novos elementos no banco de dados devemos seguir alguns passos:

- 1) Criar a classe em um pacote;
- 2) Estruturar a tabela que corresponderá o objeto;
- 3) Criar o arquivo .cfg.xml e, de acordo com os padrões do hibernate, configurar corretamente a classe java com o banco de dados;
- 4) Alterar o script inicial de preenchimento do banco. Deve-se preencher o arquivo jbpm.deploy.xml.

Adicionar no arquivo **Hibernate.cfg.xml** mais uma tag mapping-resource referente ao arquivo recém criado, para que o hibernate possa configurar e carregar os arquivos de mapeamentos corretamente.

6.2.4.4 Dependências - Hibernate

O Hibernate possui algumas bibliotecas de terceiros da qual ele depende. Todas elas estão disponíveis como partes do download do Hibernate. Abaixo estão os JARs incluindo no download do Hibernate 2.1.6. Todas elas são requeridas, exceto as marcadas como opcional. O Spring requer o Hibernate 2.1 ou superior.

- **hibernate2.jar** : core do Hibernate
 - **c3p0-0.8.4.5.jar** : Pool de conexão básico para executar testes unitários
 - **cglib-full-2.0.2.jar** : Biblioteca geradora de código para criação de proxies para classes persistentes.
 - **dom4j-1.4.jar** : biblioteca XML para parse de arquivos de configuração e mapeamento.
 - **ehcache-0.9.jar** : Cache puro em Java, cache padrão do Hibernate.
 - **jta.jar** : Java Transaction API.
 - **odmg-2.0.1.jar** : Base para o mapeamento de produtos objeto-relacional.
- (opcional) **oscache-2.0.1.jar** e (cluster-aware) **swarmcache-1.0rc2.jar** :

Implementações de caches alternativos.

6.2.4.5 Dependências – ChartFx

Da mesma forma que o Hibernate, o ChartFx é um componente utilizado para a geração de gráficos e possui muitas bibliotecas que estão incluídas junto ao projeto. Assim, para a correta configuração do banco de dados, o ChartFx precisa de uma hierarquia correta na aplicação servidor, e aonde estarão as informações para a aplicação cliente.

Assim, a hierarquia é demonstrada na Figura 14.

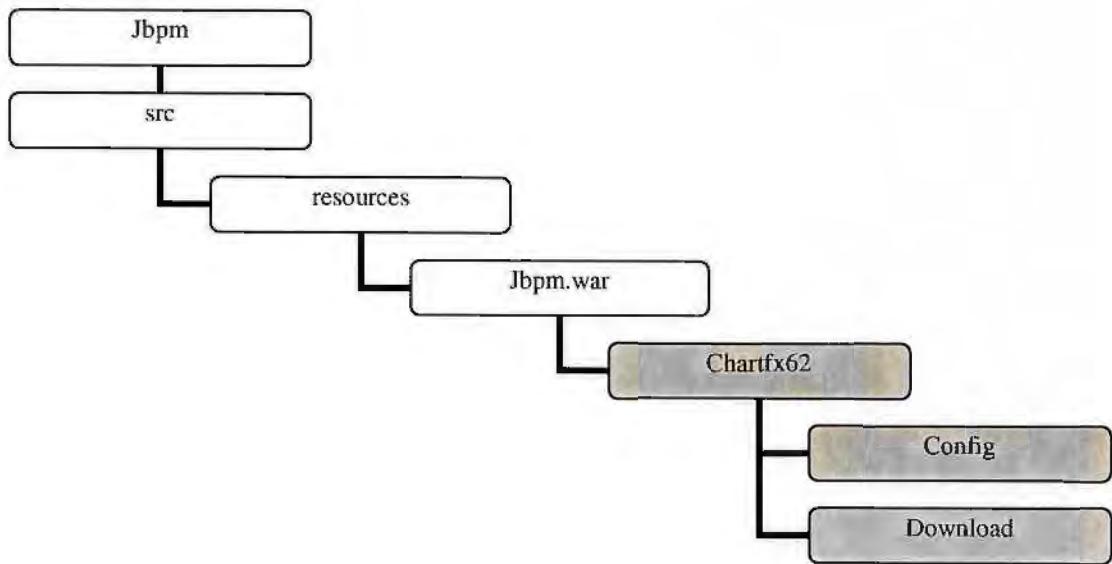


Figura 14 - Hierarquia de configurações do ChartFX

Basicamente, dentro da pasta Config se localizam os arquivos de configuração e licença do software. Esses arquivos são:

- ChartFX.Internet.config
- cfxjava.lic

Já dentro da pasta Download do Chartf, deve-se ter as bibliotecas que geraram os gráficos e que são necessários para a correta execução da aplicação.

- ChartFX.ComIEClient.Annotation.cab
- ChartFX.ComIEClient.Core.cab
- ChartFX.Internet.Annotation.DLL
- ChartFX.Internet.Base.DLL
- ChartFX.Internet.Client.DLL
- ChartFX.Internet.Polar.DLL
- ChartFX.MainClient.DLL
- ChartFX.Map.Client.DLL
- ChartFX.Statistical.Client.DLL

6.2.5 Aplicação na indústria

Através da arquitetura utilizada, pode-se ampliar a utilização do sistema para a utilização também na indústria de processos. O sistema pode se integrar a sistemas de: supervisão e aquisição de dados SCADA (*Supervisory Control And Data Acquisition*), sistema de informação de processos PIMS (*Process Information Management System*), sistemas de análises laboratoriais LIMS (*Laboratory Information Management System*), planejamento e programação da produção APS (*Advanced Production Planning and Scheduling System*) e sistemas de gestão corporativa ERP (*Enterprise Resources Planning*).

Uma proposta de arquitetura que o sistema integrado em uma indústria pode ter está presente na Figura 15.

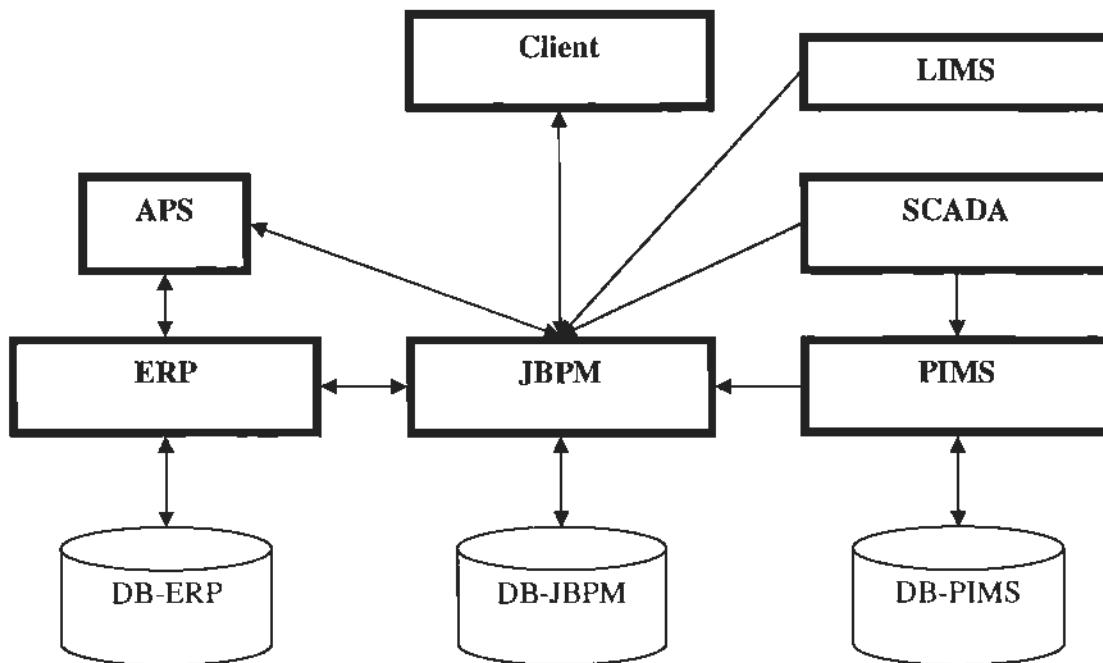


Figura 15 - Modelo de arquitetura de integração

Os sistemas SCADA são sistemas que utilizam software para monitorar e supervisionar as variáveis e os dispositivos do sistema de controle conectados através de *drivers* específicos. Estes sistemas podem assumir topologias mono-posto, cliente-servidor ou múltiplos servidores-clientes. Como a maioria dos sistemas SCADA utiliza arquitetura cliente-servidor pelo protocolo OPC (*OLE for Process Control*), o sistema pode ser alterado para poder realizar e integrar esses sistemas. A tendência dos sistemas de supervisão será a supervisão

remota, usando sistemas de telemetria através de tecnologias sem fios, como celular, rádio ou satélite. Através destas tecnologias, o sistema integrado ao sistema SCADA terão a capacidade de controlar e analisar processos industriais numa planta de trabalho local, ou geograficamente espalhada. Aplicações de logística e controle de frotas são bons exemplos práticos, assim como controle de distribuição de serviços públicos, como água, esgoto, gás ou energia. Um sistema SCADA muito utilizado no mercado é o Spectrum PowerCC da Siemens.

Já através das ferramentas PIMS, pode-se realizar busca de dados históricos diretamente no banco de dados relacional do sistema, ou ainda utilizar conectores especializados nestas funções. No mercado é muito utilizado como ferramenta PIMS o sistema Simatic IT da Siemens e o InfoPlus 21 da Aspentech.

Basicamente o sistema LIMS é muito próximo a sistemas PIMS, com características claras e definidas para as análises laboratoriais.

O sistema JBPM pode ser utilizado também integrado aos sistemas de APS para poder ajudar a empresa a alcançar seus objetivos de modo muito mais eficaz, reduzindo os custos de produção com a melhor utilização de matérias-primas e energia, por exemplo.

Utilizando o sistema JBPM integrado a um sistema ERP, pode-se garantir um gerenciamento muito mais eficiente por possibilitar a tomada de decisões com base em informações úteis, atuais e confiáveis da realidade de diversos setores da empresa.

6.3 *Implantação e documentação*

6.3.1 Casos de uso

A lista exibe os casos de uso elaborados para serem desenvolvidos na ferramenta. A documentação completa desses casos de uso pode ser encontrada no Anexo 1 – Descrição dos Casos de Uso do Sistema

- 101. Autenticação
- 102. Esqueci minha senha
- 103. Autorização
- 104. Mostrar processos por perfil
- 105. Cadastro, alteração e exclusão de usuários para o administrador
- 106. Controle de grupos/perfis de usuários
- 107. Envio de e-mail
- 201. Conexão com *web services*

6.4 *Testes*

Neste item são apresentados os testes a serem executados no sistema durante sua fase de desenvolvimento. Os testes descritos vão desde os de integração, testando a integração das funcionalidades do sistema, até os testes de unidade, testando funções e algoritmos menores. No Plano de Aceitação, presente no Anexo 2, serão apresentados os testes finais do sistema para que seja considerado pronto.

6.4.1 Testes de integração

Os testes de integração visam verificar se as unidades já validadas estão interagindo corretamente entre si após a integração. E esta é dada de forma gradual, visando alcançar o caso de uso objetivado.

Os casos de teste, agrupados por casos de uso, são exibidos na Tabela 8.

Tabela 8 - Testes de integração realizados

Status	Descrição do Teste
101. Autenticação	
OK	- Testar o acesso a uma página sem fazer a autenticação direto pelo navegador.
OK	- Repetir o teste anterior após se autenticar e apertar o link Sair , para testar o <i>logoff</i> .
OK	- Testar a autenticação para uma amostra de usuários e senhas.
102. Esqueci minha senha	
OK	- Testar o reenvio de senha para uma amostra de usuários, que tenham o e-mail.
103. Autorização	
OK	- Testar se um usuário não administrador pode acessar as páginas de monitoração de desempenho e alteração de instâncias de processos.
104. Mostrar somente os processos que o perfil pode iniciar e Grupo de distribuição	
OK	- Testar a autenticação com um usuário de cada perfil e se para cada uma são mostrados os processos correspondentes.
OK	- Testar o envio de uma tarefa para a lista de tarefas não alocadas de um grupo.
OK	- Testar se um usuário pode executar uma atividade da lista de tarefas não alocadas.
105. Cadastro, alteração e exclusão de usuários para o administrador	
OK	- Testar cadastro, alteração e exclusão de usuários.
106. Controle de grupos/perfis de usuários	
OK	- Testar cadastro de perfis pelo administrador.
107. Envio de e-mail	
OK	- Testar envio de e-mail em um processo.
201. Código para conexão com <i>web services</i> e exemplo	
OK	- Testar conexão com <i>web service</i> de exemplo.

6.4.2 Testes de Aceitação

Os testes de aceitação visam avaliar a conformidade do sistema com os requisitos funcionais, não funcionais e de qualidade percebida pelo usuário (Qualidade em uso, segundo a ISO 12207). Critérios avaliados:

- RF1 - Desenho de processos em ferramenta gráfica e via XML
- RF2 - Execução de processos
- RF3 - Criação facilitada de formulários para coleta de dados dos processos
- RF4 - Envio de *e-mail*, de acordo com eventos do processo
- RF5 - Acompanhamento do desempenho dos processos - relatórios e gráficos
- RF6 - Interface com outros aplicativos via interface padrão *web services*
- RNF1 - Segurança: Autenticação e Autorização
- RNF2 - Manutenção: Qualidade da documentação

O Relatório de Aceitação pode ser encontrado no Anexo 2 – Relatório de Revisão do Plano de Aceitação e Teste.

6.5 Avaliação dos Resultados

A plataforma se mostrou útil para a rápida implantação de processos de *workflow* e tomada de decisão com base em regras bem definidas. Foram implantados todos os requisitos previstos, apesar da interface com *web services* não ter sido usada nos processos implementados, por falta de necessidade de integração com outros aplicativos.

No entanto, houve a necessidade de fazer algumas concessões, logo no início do projeto, devido ao tempo disponível. O plano original era bem abrangente contemplando: a alteração da interface do *plug-in* de modelagem para o padrão BPMN, criação de novas funções no núcleo do sistema e evoluções no portal. Durante o projeto, concluiu-se que a adequação da linguagem de modelagem de processos ao padrão BPMN seria inviável para o cronograma de projeto acordado, e logo esta modificação não foi realizada neste projeto.

7 Aplicação no PECE (Fase de testes da ferramenta)

7.1 Detalhamento das atividades

O projeto foi aplicado a título de teste no PECE a fim de verificar o comportamento dos sistemas e suas funcionalidades num ambiente real. Para isso foram definidas as seguintes atividades, em duas etapas:

1^a Etapa: Modelagem do processo de atendimento

Modelagem dos processos do PECE	45 days	Fri 30/6/06	Fri 11/8/06
Estudo das atividades do PECE	30 days	Fri 30/6/06	Fri 28/7/06
Modelagem de processos	10 days	Fri 28/7/06	Sun 6/8/06
Validação da modelagem	5 days	Sun 6/8/06	Fri 11/8/06

2^a Etapa: Desenho na ferramenta e implantação do processo no portal

7.2 Modelagem dos processos

Após uma breve explicação sobre os departamentos do PECE e seus principais processos, partiu-se para a escolha de processos em que o ganho proveniente da aplicação da ferramenta fosse máximo. Nesta primeira etapa também se buscou minimizar o risco de uma implementação mal sucedida impactar o serviço da instituição.

Atenderam aos critérios já explicados o processo de atendimento, cuja descrição encontra-se no Quadro 1 e seu modelo na Figura 16.

Este processo suporta o trabalho da atendente que deve registrar a solicitação do cliente e busca a informação necessária com as pessoas responsáveis, dependendo do tipo de solicitação.

Quadro 1 - Descrição do processo de atendimento

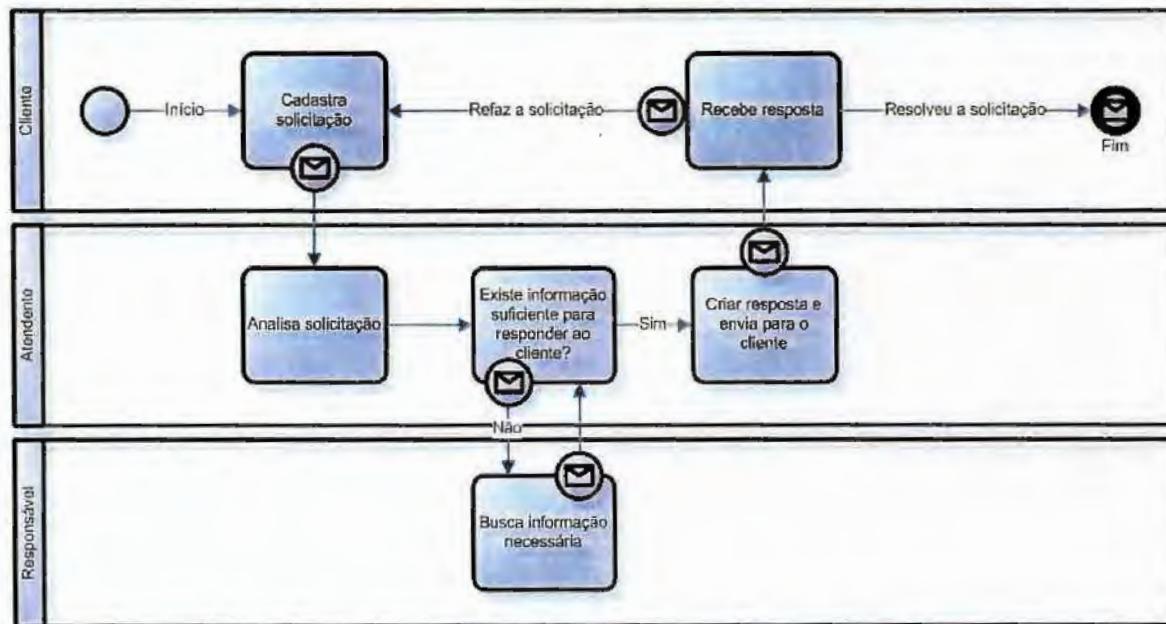


Figura 16 - Modelo de processo de atendimento

Exemplos de execução:

- Solicitação de reenvio de boleto de cobrança da mensalidade. Responsável: Financeiro.
- Solicitação de revisão de prova. Responsável: Professor.

7.3 Implantação dos processos e serviços

Após a modelagem, é preciso desenvolver os formulários, eventos e ações, que compõe o processo em sua implementação tecnológica, mas que não aparecem na modelagem de negócios. Para isso utilizou-se o módulo de modelagem da ferramenta, gerando o seguinte processo:

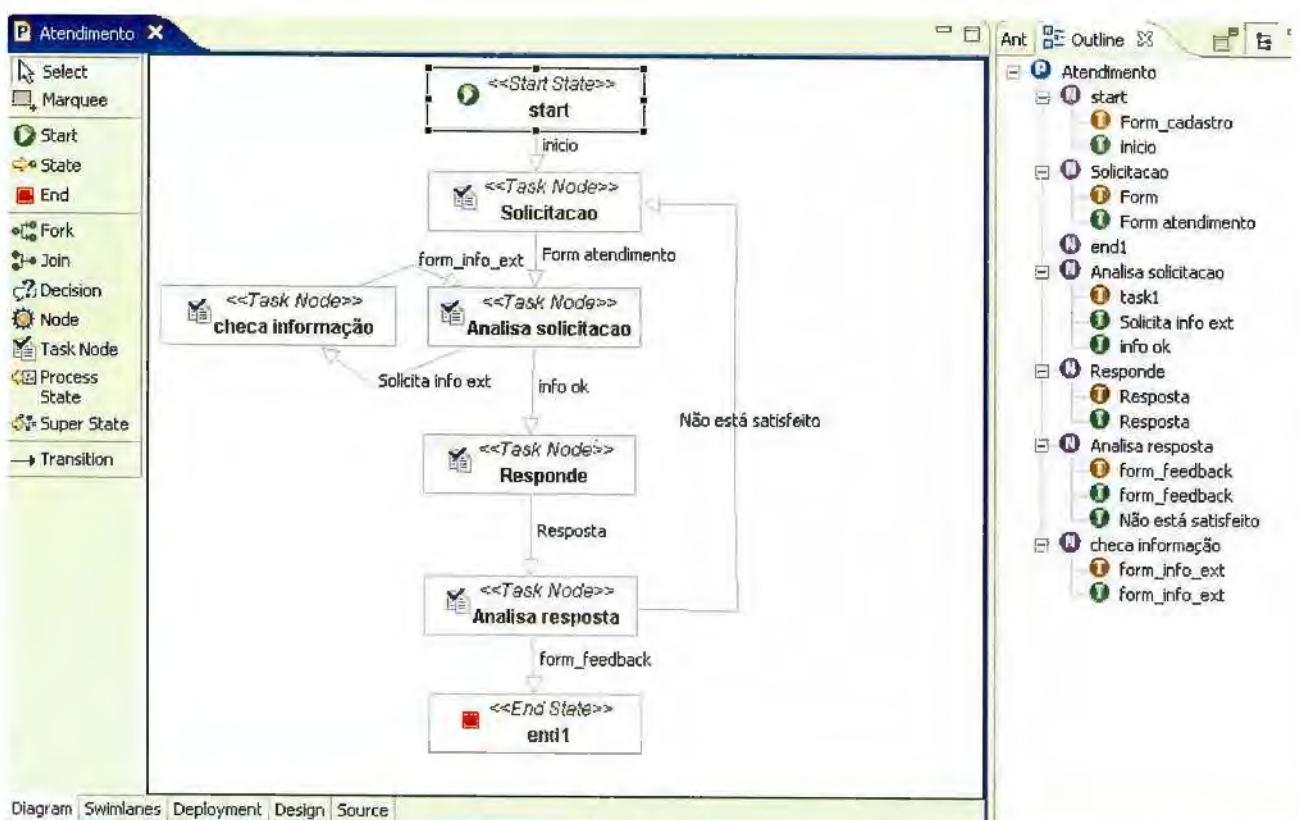


Figura 17 - Processo na ferramenta de modelagem

O processo é basicamente um fluxograma com nós especializados e funções que são disparadas em eventos de entrada ou saída de tarefas (nós) do processo. Também poderiam ser incorporados nós de acesso a sistemas externos via *web services*, decisão automatizada ou customizada. No entanto, não foi verificada esta necessidade para a implementação deste processo, já que o a versão atual do processo acontece via *e-mails* e logo o sistema substituirá a versão atual sem ter que inter-operar com outros sistemas legados.

Como pode ser visto a diante, a modelagem do processo é armazenada no formato XML para envio e instalação no servidor, que criará o modelo de dados e execução.

```

<?xml version="1.0" encoding="UTF-8" ?>
<process-definition xmlns="urn:jbpm.org:jpd1-3.1" name="Atendimento">

<swimlane name="Cliente">
<assignment expression="initiator"/>
</swimlane>
<swimlane name="Atendente">
<assignment expression="user(Atendente1)" />
</swimlane>
<swimlane name="Professor">
<assignment expression="user(Professor1)" />
</swimlane>

<start-state name="start">
<task name="Form_cadastro" swimlane="Cliente">
<controller>
<variable name="Nome" access="read,write,required" />
<variable name="E-mail" access="read,write,required" />
<variable name="Telefone" />
</controller>
</task>
<transition name="inicio" to="Solicitacao" />
</start-state>
<task-node name="Solicitacao">
<task name="Form" swimlane="Cliente">
<controller>
<variable name="Solicitacao" access="read,write,required" />
</controller>
</task>
<transition name="Form atendimento" to="Analisa solicitacao" />
</task-node>
<end-state name="end1" />
<task-node name="Analisa solicitacao">
<task name="task1" swimlane="Atendente">
</task>
</task-node>

```

Diagram | Swimlanes | Deployment | Design | Source

Figura 18 - Código resultante da modelagem do processo

Assim, finalizadas as duas etapas de modelagem e desenvolvimento, o próximo passo foi utilizar o processo no portal e analisar seu histórico para futuras melhorias e descrição de casos não previstos inicialmente.

7.4 Testes de aceitação

Os testes de aceitação visam avaliar a satisfação do usuário em relação à forma como o processo foi implementado, ou seja, determina a aceitação da ferramenta para seu uso na empresa. Pode-se identificar dois fatores determinantes nesta etapa:

- **Adequação do processo implantado ao processo real**

Devido ao processo de modelagem e às características tecnológicas, o processo final pode apresentar problemas que não existiam anteriormente. Por exemplo: não seriam previstas situações de exceção onde um atalho é necessário para pular algumas etapas devido à exigência de agilidade dos negócios ou situação não prevista pelo próprio processo de negócio (legislação, falhas em outros sistemas relacionados, etc.).

Este tipo de exceção pode gerar a não entrada de dados no sistema, e consequentemente a falta de histórico, que pode levar a problemas de auditoria ou até ao descrédito no sistema pelos usuários.

- **Adequação do usuário ao ambiente (usabilidade)**

Quando são criados procedimentos e normas sempre existe resistência dos usuários, assim como acontece na adoção de novos sistemas de TI. Quando estes dois são unidos, cria-se uma potencial barreira de adoção da solução que deve ser transposta através do correto planejamento em conjunto com os usuários e através de um plano de comunicação abrangente e planejado.

No projeto, optou-se por fazer reuniões periódicas com os usuários para avaliar, a cada etapa do desenvolvimento, se o resultado estava como esperado. Neste aspecto, o conceito do BPM de evolução contínua dos processos foi muito útil, pois, na sua essência, ele já prevê sucessivas etapas de modelagem, desenvolvimento e teste. Porém, é importante notar que o fato do processo ser cíclico, ou em espiral, não gera esforço desnecessário para remodelamento, pois a cada etapa são agregados novos atributos e funções até que o resultado seja próximo do desejado e pouco ou nenhum trabalho precisa ser refeito.

8 Considerações Finais

8.1 Conclusão

Com este trabalho foi possível aplicar os conceitos do gerenciamento de processos de negócios (BPM), através do desenvolvimento de um ambiente de modelagem, execução e monitoramento desses processos.

O processo de pesquisa sobre o tema e as tecnologias relacionadas foi de fundamental importância para a correta compreensão do contexto em que se situa o trabalho e o seu resultado.

Nesse processo, descobriu-se que o tema do projeto não estava restrito apenas à área de tecnologia, mas também envolve conceitos relacionados ao modo de trabalho das pessoas envolvidas e ao ambiente de negócios, que podem ser fatores de risco para o sucesso de um projeto de automação de processos.

Já durante as atividades de desenvolvimento da ferramenta, houve a possibilidade de ter contato e desenvolver um sistema baseado em troca de mensagens e envolvendo conceitos de arquitetura orientada a serviços, em que a complexidade do sistema trouxe um nível de dificuldade além do esperado no início do projeto.

Além disso, também foram encontrados problemas no desenvolvimento do componente gráfico de modelagem que, devido ao seu alto nível de detalhes e ser baseado em outra tecnologia, ficou fora do escopo a que o projeto se propunha.

Mesmo com tais dificuldades, foi possível desenvolver um ambiente funcional e que atingiu os objetivos propostos pelos casos de uso e requisitos, além de permitir um aprendizado prático em que os conceitos estudados foram aplicados e testados.

8.2 Contribuição do projeto

Esse projeto foi fundamental para a compreensão das aplicações e limitações do BPM, em que se descobriu logo no início que se tratava de um modelo de gestão de empresas e que não necessariamente determina uma tecnologia, mas sim que pode ser suportado por um conjunto de ferramentas que implementam sua filosofia.

Esses estudos também foram importantes para a descoberta de que o padrão BPMN, apesar de ter como objetivo ser uma notação independente da área de negócios a que vai ser aplicada, não consegue atender a esse objetivo por não incorporar elementos específicos e necessários para aplicação em diferentes áreas, resultando em modelos simples que, por vezes, não representam todos os elementos para uma automação bem sucedida.

Também foi possível observar as limitações que as tecnologias utilizadas impõem à automação de processos, reduzindo sua aplicação a processos baseados em troca de documentos e informações entre usuários (*workflow*), não sendo ideal para processos que exijam processamento de grande volume de dados, que podem ser tratados por outras ferramentas.

8.3 Trabalhos Futuros

A partir dos resultados desse trabalho, há a possibilidade de estudo do impacto da implementação do BPM em diversas empresas a fim de mensurar a influência dos componentes humano e de tecnologia nas implementações de sistemas empresariais.

Ficou claro que a ferramenta atende às necessidades e é uma ótima opção para o suporte aos processos empresariais, o que justifica uma expansão futura com o objetivo de possibilitar sua utilização de forma mais intuitiva e com maior poder para modelagem e execução de processos. Para isso sugeriu-se incorporar os seguintes requisitos adicionais:

- Utilizar arquitetura orientada a serviços
- Reutilização de processos já existentes de forma mais eficiente
- Ajuda sobre os processos em cada página de execução de tarefas
- Compartilhamento de arquivos e mensagens entre usuários
- Ferramenta para gerenciamento de Regras de Negócio separada dos processos

Dessa forma, ter-se-ia um ambiente completo e comparável com outras opções de mercado.

REFERÊNCIAS

CBR. Making a Business Case for BPM. 2005. Disponível em <http://www.cbronline.com/article_cbr.asp?guid=78090714-E59A-49A0-B403-D42CE3AEF6CF> Acesso em 24/11/2006

DUTRA, E. P. BPM – Uma nova arquitetura de tecnologia da informação na gestão de processos de negócio. 2006. 142p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

ENIX. A BPM Primer. 2003. Disponível em <<http://www.enix.co.uk/Documents/BPM%20-%20A%20Primer.pdf>> Acesso em 24/11/2006.

GIAGLIS, G. M. A Taxonomy of Business Process Modelling and Information Systems Modelling Techniques. 2004. Brunel University, Reino Unido.

HARRISON-BRONINSKI, K. BPM, Anyone? 2006. Disponível em <<http://www.bpmg.org/Zpost1450.php>> Acesso em 24/11/2006.

JENSEN, C. J. An Introduction to Business Process Management. 2003. Disponível em <<http://www.dpsmagazine.com/content/ContentCT.asp?P=38>> Acesso em 24/11/2006.

KHAN, R. N. Outside SAP. 2006. Disponível em <http://www.itobserver.com/articles/1170/outside_sap/> Acesso em 24/11/2006.

LACHAL, L. What is business process management. 2005. Disponível em <http://download.microsoft.com/documents/uk/windowsserversystem/bpm/ovum_what_is_bp_m.pdf> Acesso em 24/11/2006.

MCGREGOR, M. How to Apply Best Practice Priuciples Of BPM in The Real World. Apresentado a Seminário Executivo de BPM 2006, Rio de Janeiro. Não publicado.

_____. **Living in the Age of the Customer.** 2006. Disponível em <<http://www.bpmg.org/Zpost1465.php>> Acesso em 12/10/2006.

MIERS, D. Best Practice BPM. 2006. Disponível em <<http://www.acmqueue.org/modules.php?name=Content&pa=showpage&pid=374>> Acesso em 24/11/2006.

_____. **The Split Personality of BPM.** 2004. Disponível em <<http://www.enix.co.uk/Documents/The%20Split%20Personality%20of%20BPM.pdf>> Acesso em 24/11/2006.

MIERS, D; HARMON, P. **The 2005 BPM Suites Report.** 2005. BUSINESS PROCESS TRENDS

NETTO, C. A. A. **Proposta de modelo de mapeamento e gestão por macroprocessos.** 2004. 146p. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2004.

SCHURTER, T. **Technologists Lost In Technology.** 2006. Disponível em <<http://www.bpmg.org/Zpost1503.php>> Acesso em 24/11/2006.

SMITH, H.; FINGAR, P. **BPM's Third Wave.** 2003. Disponível em <http://www.bptrends.com/deliver_file.cfm?fileType=publication&fileName=BPM%20Third%20Wave%20Smith%20Fingar%20Apr2003%2Epdf> Acesso em 24/11/2006.

USIRONO, C. H. **Tecnologia Workflow: O impacto de sua utilização nos processos de negócio. Um estudo de casos múltiplos.** 2003. 120p. Dissertação (Mestrado) – Faculdade de Economia e Administração, Universidade de São Paulo, São Paulo, 2003.

WHITE, S. A. **Introduction to BPM.** 2005. Disponível em <<http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf>>. Acesso em 24/11/2006.

Anexo 1 – Descrição dos Casos de Uso do Sistema

Caso de Uso
101. Autenticação
Descrição
Este caso de uso descreve como é realizada a autenticação do usuário no sistema.
Evento iniciador
Inicialização do sistema
Atores
Usuário do sistema
Pré-condição
Usuário deve ter um nome de usuário e senha.
Seqüência de eventos
<ol style="list-style-type: none"> 1. Usuário preenche os campos com nome de usuário através de um menu com opções e insere sua senha, 2. Confirma a operação clicando no botão “Entra”.
Pós-condição
Usuário é autenticado no sistema e é redirecionado para sua página inicial.
Extensões
<ol style="list-style-type: none"> 1. Usuário não cadastrado deve consultar o administrador do sistema. 2. Usuário não lembra a senha e solicita o envio de senha por e-mail através do link “Esqueceu sua senha?” (caso 102).
Inclusões

Caso de Uso
102. Esqueci minha senha
Descrição
Este caso de uso descreve como é efetuado o envio de senha ao usuário que esqueceu a sua.
Evento iniciador
Clique no link “Esqueceu sua senha?”.
Atores
Usuário
Pré-condição
Ter iniciado o sistema.
Seqüência de eventos
<ol style="list-style-type: none"> 1. O usuário clica no link “Esqueceu sua senha?”, 2. O sistema analisa que usuário está solicitando a senha, consulta sua base de dados e envia a senha por e-mail ao usuário que solicitou a senha.
Pós-condição
Usuário deve receber em seu e-mail a senha.
Extensões
Inclusões

Caso de Uso
103. Autorização
Descrição
Este caso de uso descreve como é efetuada a autorização de um dado usuário.
Evento iniciador
Usuário solicita uma funcionalidade através da interface.
Atores
Sistema
Pré-condição
Usuário estar autenticado
Seqüência de eventos
<ol style="list-style-type: none"> 1. Usuário solicita uma dada funcionalidade, 2. Sistema consulta a base de dados e o perfil deste usuário para liberar o acesso a tal funcionalidade.
Pós-condição
Usuário utiliza a funcionalidade da ferramenta.
Extensões
<ol style="list-style-type: none"> 1. Se usuário não tiver acesso a tal funcionalidade é exibida uma tela de autenticação (caso 101).
Inclusões

Caso de Uso
104. Mostrar processos por perfil
Descrição
Este caso de uso demonstra como são exibidos os processos de acordo com cada perfil registrado na página inicial após a autenticação.
Evento iniciador
Visita à página inicial.
Atores
Usuário
Pré-condição
Estar autenticado no sistema.
Seqüência de eventos
<ol style="list-style-type: none"> 1. Sistema verifica na base de dados as atividades atribuídas e seus parâmetros ao usuário solicitante e processos que ele pode iniciar. 2. Sistema exibe a interface de execução de processos.
Pós-condição
Usuário pode verificar o andamento das atividades alocadas a ele, inserir novas atividades, entre outras funcionalidades da interface.
Extensões
Inclusões

Caso de Uso
105. Cadastro, alteração e exclusão de usuários para o administrador.
Descrição
Este caso de uso descreve as atividades de cadastro, alteração e exclusão de usuários.
Evento iniciador
Solicitação da interface de monitoramento
Atores
Administrador
Pré-condição
Estar autenticado como Administrador do sistema.
Seqüência de eventos
<ol style="list-style-type: none"> 1. O Administrador solicita a exibição do menu de monitoramento, 2. Nesta interface o Administrador pode inserir novo cadastro de usuários 3. Confirma dados apertando o botão enviar
Pós-condição
Cadastros, alterações e exclusões efetuadas.
Extensões
Inclusões

Caso de Uso
106. Controle de grupos/perfis de usuários
Descrição
Este caso de uso descreve como é efetuado o controle de grupos e perfis de usuários.
Evento iniciador
1. Administrador solicita a interface de controle de grupo e perfis de usuários.
Atores
Administrador
Pré-condição
Estar autenticado no sistema como administrador.
Seqüência de eventos
<ol style="list-style-type: none"> 1. Seleção de um usuário na lista de usuários 2. Escolha de um grupo e confirmar a associação.
Pós-condição
Extensões
Cadastro de um novo grupo
Inclusões

Caso de Uso
107. Envio de e-mail
Descrição
Este caso de uso descreve como funciona o envio de e-mail para um dado processo.
Evento iniciador
Atribuição de uma tarefa a um usuário.
Atores
Sistema
Pré-condição
Ter efetuado todas as transações envolvidas no processo.
Seqüência de eventos
<ol style="list-style-type: none"> 1. Sistema verifica que existe uma solicitação de envio de e-mail nas regras de negócios do processo em questão 2. Sistema verifica se as pré-condições de envio estão estabelecidas 3. Sistema envia e-mail ao destinatário descrito durante o processo.
Pós-condição
Usuário recebe um e-mail avisando que existe uma atividade pendente no sistema.
Extensões
Inclusões

Caso de Uso
201. Interface para consulta de dados agregados sobre as instâncias de um processo
Descrição
Este caso de uso descreve o funcionamento das interfaces que fornecem dados para análise da eficiência dos processos.
Evento iniciador
Solicitação da interface de gráficos de desempenhos.
Atores
Administrador
Pré-condição
Estar autenticado.
Seqüência de eventos
<ol style="list-style-type: none"> 1. Escolher o gráfico ou relatório desejado na lista disponível. 2. Escolher processo desejado.
Pós-condição
Gráfico ou relatório gerado.
Extensões
Configurar tempos padrão para processos.
Inclusões

Anexo 2 – Relatório de Revisão do Plano de Aceitação e Teste

Produto avaliado: Sistema PECE BPM

<i>Data: 12/12/06</i>	
-----------------------	--

Participantes

Líder: Guilherme

Relator: Guilherme

- 1.
- 2.
- 3.

Resultado da avaliação

ACEITO

NÃO ACEITO (Nova revisão é necessária)

Como está

Revisões maiores

Com revisões menores

Reconstruir

Revisão incompleta

Materiais suplementares produzidos

Itens da revisão

<i>Teste no. 1</i>	<i>Requisito testado</i> Desenho de processos em ferramenta gráfica e via XML	
<i>Funcional:</i> <input checked="" type="checkbox"/>	<i>Observações:</i>	
<i>Não funcional:</i> <input type="checkbox"/>		
<i>Pré-condição:</i> 1) Abrir o Eclipse, já configurado.		
<i>Seqüência para verificação do caso comum</i>	<i>Comportamento esperado do software</i>	<i>Testado</i>
1) Criar um novo projeto	Cria diretórios e arquivos padrão	Ok
2) Criar um novo processo	Cria pasta do processo e abre área de desenho	Ok
3) Desenhar um processo simples de teste	Arrastar e soltar os ícones de tarefas e ligações	Ok
4) Fazer a instalação do processo no servidor	Processo aparece na lista de processos do website e pode ser executado.	Ok
<i>Seqüência para verificação de exceções</i>	<i>Comportamento esperado do software</i>	
<i>Comentários</i>		
<input checked="" type="checkbox"/> Aprovado		
<input type="checkbox"/> Reprovado		
<i>Cientes:</i> PECE <i>Avaliadores:</i> Equipe	<i>Apresentador:</i> Guilherme	

Teste no. 2		<i>Requisito testado</i> Execução de processos e Envio de e-mail,
<i>Funcional:</i>	<input checked="" type="checkbox"/>	<i>Observações:</i>
<i>Não funcional:</i>	<input type="checkbox"/>	
<i>Pré-condição:</i> I) Estar autenticado no portal		
<i>Seqüência para verificação do caso comum</i>	<i>Comportamento esperado do software</i>	<i>Testado</i>
1) clicar no link de iniciar o processo	Aparece um formulário e o fluxograma do processo	Ok
2) Envie o formulário	Envia uma tarefa para algum usuário o finaliza do processo	Ok
3) Verifique se foi enviado um e-mail com os dados do processo para o próximo responsável.	Responsável pela tarefa recebe um e-mail convocando para entrar no site.	Ok
<i>Seqüência para verificação de exceções</i>	<i>Comportamento esperado do software</i>	
Não preencher os campos obrigatórios	Mostrar erro e pedir correção	ok
<i>Comentários</i>		
<input checked="" type="checkbox"/> Aprovado		
<input type="checkbox"/> Reprovado		
<i>Cientes:</i> PECE <i>Avaliadores:</i> Equipe		<i>Apresentador:</i> Guilherme

Teste no. 3		Requisito testado Criação de formulários para coleta de dados dos processos
Funcional:	<input checked="" type="checkbox"/>	Observações:
Não funcional: <input type="checkbox"/>		
Pré-condição: 1) Abrir um processo do Eclipse.		
Seqüência para verificação do caso comum	Comportamento esperado do software	Testado
1) Crie uma nova tarefa para algum processo		Ok
2) Clique 2 vezes na tarefa e cadastre os campos na aba: controler	Interface para cadastro de campos e validação do formulário	Ok
3) Executar o processo e verificar o formulário funcionando	Campos são apresentados na ordem correta e são salvos no banco de dados ao enviar o formulário	Ok
Seqüência para verificação de exceções	Comportamento esperado do software	
<input checked="" type="checkbox"/> Aprovado	Comentários	
<input type="checkbox"/> Reprovado		
Cientes: PECE		
Avaliadores: Equipe		
Apresentador: Guilherme		

Teste no. 4	<i>Requisito testado</i> Acompanhamento do desempenho dos processos	
<i>Funcional:</i> <input checked="" type="checkbox"/>	<i>Observações:</i>	
<i>Não funcional:</i> <input type="checkbox"/>		
<i>Pré-condição:</i> 1) Estar autenticado no portal		
<i>Seqüência para verificação do caso comum</i>	<i>Comportamento esperado do software</i>	<i>Testado</i>
1) Clicar no link desempenho	Mostra listas de gráficos disponíveis	Ok
2) Acessar os gráficos	Verifique se eles funcionam para todos os processos e mostram dados corretos	Ok
<i>Seqüência para verificação de exceções</i>	<i>Comportamento esperado do software</i>	
Processos não finalizados(ativos) não são contabilizados	Gráficos não mostram dados de processo ativos	
<i>Comentários</i>		
<input checked="" type="checkbox"/> Aprovado		
<input type="checkbox"/> Reprovado		
Cientes: PECE Avaliadores: Equipe	Apresentador: Guilherme	

Teste no. 5		<i>Requisito testado</i> Interface com outros aplicativos via interface padrão web services
<i>Funcional:</i>	<input checked="" type="checkbox"/>	<i>Observações:</i>
<i>Não funcional:</i>	<input type="checkbox"/>	
<i>Pré-condição:</i>		
1) Abrir o eclipse e criar um processo de decisão		
<i>Seqüência para verificação do caso comum</i>	<i>Comportamento esperado do software</i>	<i>Testado</i>
1) Criar um nó de decisão que acessa dados externos via webservice		Ok
2) Executar processo e verificar se os dados são acessados com sucess	Decisão é tomada corretamente	Ok
<i>Seqüência para verificação de exceções</i>	<i>Comportamento esperado do software</i>	
Webservice fora do ar	Erro	
<i>Comentários</i>		
<input checked="" type="checkbox"/>	Aprovado	
<input type="checkbox"/>	Reprovado	
<i>Cientes:</i> PECE <i>Avaliadores:</i> Equipe		<i>Apresentador:</i> Guilherme

Teste no. 6	<i>Requisito testado</i> Autenticação e Autorização	
<i>Funcional:</i> <input checked="" type="checkbox"/>	<i>Observações:</i>	
<i>Não funcional:</i> <input type="checkbox"/>		
<i>Pré-condição:</i>		
<i>Seqüência para verificação do caso comum</i>	<i>Comportamento esperado do software</i>	<i>Testado</i>
1) Tentar acessar páginas do site sem estar logado	Mostrar tela de login	Ok
2) Tentar acessar páginas de administrador com login de usuário comum	Mostrar tela de login	Ok
<i>Seqüência para verificação de exceções</i>	<i>Comportamento esperado do software</i>	
<input checked="" type="checkbox"/> Aprovado	<i>Comentários</i>	
<input type="checkbox"/> Reprovado		
<i>Cientes: PECE</i> <i>Avaliadores: Equipe</i>	<i>Apresentador: Guilherme</i>	

Anexo 3 – Especificações da Interface de Hardware e Software aplicados ao desenvolvimento

Especificações de Hardware/Software:

Operating system – Microsoft Windows XP

Processor architecture/Window system –Intel x86 / Win32

Java 2 Platform – Sun Java 2 Standard Edition 1.4.2_10 for Microsoft Windows

Especificações de Software/Ferramentas:

J2EE

O J2EE (Java Enterprise Edition) é uma plataforma de desenvolvimento de aplicações distribuídas, ou seja, através de componentes que são executados em um servidor de aplicação. O projeto utiliza alguns componentes da especificação J2EE. Desta forma, o J2EE apresenta uma série de especificações para tratamento de banco de dados, segurança, transações, ligações a aplicações legadas, EJB (Enterprise JavaBeans – componentes de software, focado na regra de negócio da aplicação) Desta forma, o servidor de aplicação se responsabiliza pela infra-estrutura, segurança, disponibilidade e escalabilidade da aplicação

Jboss jBPM

JBoss é um sistema que visa a administração de workflows. Com essa ferramenta é possível descrever processos de negócios na forma gráfica através de tarefas, estados de espera, ações, nós e transições. Pode ser usado como uma biblioteca Java, utilizar qualquer tipo de sistema de banco de dados e servidor de aplicação. Atualmente suporta duas linguagens de processos a jPDL, para implementar BPM e workflow em Java combinando gerenciamento de tarefas com workflow; e, também BPEL.

Eclipse

O Eclipse foi utilizado no projeto para ser a plataforma de desenvolvimento da solução. Utilizando a versão mais recente da aplicação (3.2) e com a utilização de testes unitários, log4j (framework de logging) e do MyEclipse, o processo de desenvolvimento foi mais eficaz.

O Eclipse é uma plataforma de desenvolvimento *Open Source*, ou, baseada em Java que inclui uma série de *plug-ins*.

MyEclipse 5.0

O MyEclipse foi utilizado para poder realizar uma maior produtividade, integração e desenvolvimento de componentes. Com a ferramenta, é possível realizar a integração do desenvolvimento J2EE, junto com o Eclipse, e todo o desenvolvimento Web. Tanto para JSP, XML, Struts, Jsf, Hibernate e servidores de aplicação já que o MyEclipse é um plug-in que trás muitas funcionalidades ao Eclipse.

Hibernate 3.0

O Hibernate foi utilizado para realizar o mapeamento dos objetos do sistema em um banco de dados relacional. Assim, praticamente todos os fluxos de dados, assim como os processos e seus relacionamentos e iterações, que são mapeadas em objetos, são persistidos no banco de dados. Desta forma, a utilização de códigos de acesso a banco de dados e SQL foi simplesmente reduzida a configurações e há algumas querys para a busca de dados.

Com a utilização da ferramenta, foi possível diminuir a complexidade resultante da convivência de modelos diferentes: modelo orientado a objetos (linguagem Java) e o relacional (da maioria dos SGBDs). Além disso, o Hibernate é uma ferramenta interessante a nossa aplicação, pois não foram utilizados muitos *stored procedures* nem *triggers*, que são pontos fracos deste *framework*.

Struts

O Struts é um *framework* de desenvolvimento da camada responsável pela camada de apresentação. Basicamente, o Struts envolve e engloba todos os controles e manuseio das páginas que serão enviadas para o usuário do sistema.

O Struts foi utilizado devido aos grandes avanços de mercado e a ampla utilização e apoio pela comunidade de desenvolvimento Java-Web, por se tornar um padrão de mercado. Ele apresenta, desta forma, integração com a maioria das IDEs de mercado, sendo que no Eclipse, o Struts é apoiado pelo MyEclipse. É através do Struts que se pode manter a aplicação padronizada, facilitando a manutenção, e aplicações internacionalizadas. Além, é claro, de aumentar a produtividade do desenvolvimento.

JSP

O JSP (Java Server Pages) foi utilizado para o desenvolvimento da camada de apresentação, que será enviada ao usuário (camada cliente). Baseada na linguagem de programação Java, ela apresenta portabilidade de plataforma, que permite a sua execução em vários sistemas operacionais. O JSP é instalado no servidor de aplicação JBOSS-BPM, que é compatível com a tecnologia J2EE.

JMX

JMX (Java Management Extension) é uma especificação que define uma arquitetura de gerencia, pois permite o gerenciamento local e remoto de aplicações, objetos de sistema, drives e mais. O nível de instrumentação, agente e gerente. O primeiro nível fornece a gerencia em si. O segundo, os agentes para a gerencia e o terceiro, os componentes utilizados na gerencia.

HSQLDB

O HSQLDB (Hypersonic SQL Database) é um banco de dados, baseado em Java projetado para arquitetura cliente serviror ou standalone. É um banco multiplataforma, que ocupa pouco espaço em disco, tem poucos recursos, mas um bom desempenho. Não tem a mesma robustez e segurança de um Oracle. Esta base foi utilizada pois algumas das aplicações *desktops* interagem com a persistência através de uma query SQL.

JSF

O JSF (JavaServer Faces) é um *framework* para o desenvolvimento de aplicações Web de forma fácil, rápido e dinâmica, pois permite o desenvolvedor criar UIs (*User Interfaces* – Interfaces do usuário) através de um conjunto de componentes UIs pré-definidos. O JSF apóia o desenvolvimento de JSP.

Log4j

É um pacote (API) projetado por vários autores para controlar o comportamento de *logging* de uma aplicação. Pode-se, por exemplo, ativar o *log* em tempo de execução sem modificar os binários, mudando um arquivo de configuração. Desta forma, mantém-se o código conciso e legível diferentemente de quando se insere *statements* de *log* no código. Além disso, sua saída pode ser escrita em arquivos específicos.

ChartFX

É um componente JAVA que usa JSP e servidores de aplicação J2EE para produzir gráficos em diversos formatos (JPEG, FLASH, PNG). É uma ferramenta que roda nos mais populares servidores de aplicações, está disponível como server-side bean, tem vários recursos visuais como bordas, renderização, transparência. Além disso, pode escrever todos atributos visuais em um arquivo XML.

Dependências

O Hibernate possui algumas bibliotecas de terceiros da qual ele depende. Todas elas estão disponíveis como parte do download do Hibernate. A seguir estão os JARs incluindo no download do Hibernate 2.1.6. Todas elas são requeridas, exceto as marcadas como opcional. O Spring requer o Hibernate 2.1 ou superior.

hibernate2.jar : core do Hibernate

c3p0-0.8.4.5.jar : Pool de conexão básico para executar testes unitários

cglib-full-2.0.2.jar : Biblioteca geradora de código para criação de proxies para classes persistentes.

dom4j-1.4.jar : biblioteca XML para parse de arquivos de configuração e mapeamento.

ehcache-0.9.jar : Cache puro em Java, cache padrão do Hibernate.

jta.jar : Java Transaction API.

odmg-2.0.1.jar : Base para o mapeamento de produtos objeto-relacional.

(opcional) **oscache-2.0.1.jar** e (cluster-aware) **swarmcache-1.0rc2.jar** : Implementações de caches alternativos.

Anexo 4 – Especificações da Interface de Hardware e Software da aplicação no PECE

Especificações de Hardware/Software:

Operating system – Microsoft Windows XP

Processor architecture/Window system –Intel x86 / Win32

Java 2 Platform – Sun Java 2 Standard Edition 1.4.2_10 for Microsoft Windows

Especificações de Software/Ferramentas:

Ver Anexo 6 – Manual do Usuário e de Instalação da Ferramenta.

Anexo 5 – Pacotes do Projeto

São descritos brevemente os demais pacotes que são utilizados no sistema. Basicamente, são pacotes de suporte que auxiliam direta ou indiretamente seu funcionamento.

Pacote	Descrição
org.jbpm.ant	Configuração ANT
org.jbpm.bytes	Auxílio para configuração de diversos DB
org.jbpm.configuration	Cria os objetos gráficos baseados na descrição XML para a configuração do JBPM.
org.jbpm.db.compatibility	Auxílio para configuração de diversos DB
org.jbpm.db.hibernate	Auxílio para configuração de diversos DB
org.jbpm.db.jmx	Auxílio para configuração de diversos DB
org.jbpm.file.def	Armazena as informações de informações binárias
org.jbpm.graph.action	Contém as implementações básicas das ações dos processos
org.jbpm.graph.def	Contém todos os elementos estruturais para um processo gráfico.
org.jbpm.graph.exe	Contém a estrutura de tempo de execução para a execução de um processo.
org.jbpm.graph.log	Log dos gráficos de processos
org.jbpm.graph.node	Contém estrutura básica para os nós
org.jbpm.graph.node.advanced	Contém estrutura básica para os nós avançados
org.jbpm.instantiation	Gerencia as instâncias das classes.
org.jbpm.jpd़l.convert	Conversão de jpd़l 2.0 para jpd़l 3.0.
org.jbpm.jpd़l.el	Elementos de jpd़l
org.jbpm.jpd़l.el.impl	Elementos de jpd़l
org.jbpm.jpd़l.el.parser	Elementos de jpd़l
org.jbpm.jpd़l.exe	Estruturas de dados de execução de jpd़l.
org.jbpm.jpd़l.par	Empacota os arquivos do sistema.
org.jbpm.jpd़l.xml	Transforma xml em jpd़l.
org.jbpm.logging	Apoio para log do sistema
org.jbpm.logging.db	Apoio para log do sistema
org.jbpm.logging.exe	Apoio para log do sistema
org.jbpm.logging.log	Apoio para log do sistema

org.jbpm.module.def	Apoio para definição dos módulos
org.jbpm.module.exe	Apoio para execução dos módulos
org.jbpm.msg	Interface abstrata para tecnologia de mensagens assíncronas.
org.jbpm.msg.command	Apoio para tecnologia de mensagens
org.jbpm.msg.db	Apoio para tecnologia de mensagens
org.jbpm.msg.jms	Apoio para tecnologia de mensagens
org.jbpm.persistence	Apoio para tecnologia de persistências
org.jbpm.persistence.db	Apoio para tecnologia de persistência
org.jbpm.scheduler	Provê o tempo para os processos
org.jbpm.scheduler.db	Provê o tempo para os processos-Banco
org.jbpm.scheduler.def	Provê o tempo para os processos – Definição
org.jbpm.scheduler.exe	Provê o tempo para os processos – Execução
org.jbpm.scheduler.impl	Provê o tempo para os processos
org.jbpm.security.authentication	Apoio segurança - Autenticação
org.jbpm.security.authorization	Apoio segurança - Autorização
org.jbpm.security.permission	Apoio segurança – Permissão
org.jbpm.svc	Apoio serviços
org.jbpm.svc.save	Apoio serviços
org.jbpm.taskmgmt	Interface de gerenciamento de tarefas.
org.jbpm.taskmgmt.def	Interface de gerenciamento de tarefas – definição
org.jbpm.taskmgmt.exe	Interface de gerenciamento de tarefas – Execução
org.jbpm.taskmgmt.impl	Interface de gerenciamento de tarefas – Impl.
org.jbpm.taskmgmt.log	Interface de gerenciamento de tarefas – Log
org.jbpm.util	Utilidades de ajuda para jbpm
org.jbpm.web	Utilidades de ajuda para WEB

Anexo 6 – Manual do Usuário e de Instalação da Ferramenta

1 Introdução ao Manual do software PECE BPM

1.1 O software PECE BPM

O objetivo deste trabalho é estudar os conceitos do gerenciamento de processos de negócios (BPM) e desenvolver um ambiente de modelagem, execução e monitoramento desses processos.

Através da pesquisa de padrões existentes, da análise de ferramentas de mercado, de comparações com outros modelos de gerência de empresas e da implementação da ferramenta desenvolvida, pretende-se comprovar as vantagens que o BPM pode trazer ao ser adotado pelas empresas, como a aproximação dos usuários no desenvolvimento dos processos, a flexibilidade obtida pela empresa e a melhoria nos resultados através do constante aperfeiçoamento dos processos.

2 Instalação do ambiente

Faça o download dos pacotes necessários:

JDK 5 update 7 ou superior – <http://java.sun.com/javase/downloads/index.jsp>

Eclipse 3.1 ou superior – www.eclipse.org

PECE_BPM_v2.0x - PECE_BPM_v2.0x.zip

JBoss jBPM Process Designer Plugin 3.0.9.2 - [jbpm-gpd-site-3.0.9.2.zip](http://jboss.org/jbossmiddleware/jbossjbpm-processdesignerplugin)

Instalação dos pacotes necessários:

1. Instalar o JDK
2. Descompactar o eclipse em “c:/eclipse”
3. Descompactar o PECE_BPM_v2.0x.zip na pasta “c:/” e verificar se no arquivo “C:\jbpm-starters-kit-3.1.2\jbpm\build.properties” o nome do caminho está correto para todas as variáveis e se usar barras “\” ex:
`jbpm.home=C:\PECE_BPM_v2.0x\jbpm`.
4. Abra o Eclipse e clique em:
 - a. Manu Help > Software updates > Find and Install
 - b. Escolher a opção: search new for features to install
 - c. New arquived site: jbpm-gpd-site-3.0.9.2.zip > Ok.
 - d. Siga a sequência de next e finish.

Teste de instalação bem sucedida:

1º - Teste o servidor:

Passo 1:	Clique em: C:\jbpm-starters-kit-3.1.2\jbpm-server\start.bat
Passo 2:	Acesse o servidor em http://localhost:8080/jbpm
Resultado:	O portal jBPM deve estar funcionando.

2º - Teste o ambiente de desenvolvimento:

Pré-condição: Supomos que o teste do servidor foi feito com sucesso e o servidor está ativo.

Passo 1:	Abra o Eclipse
Passo 2:	Clique em File > new > project > Process Project.
Passo 3:	Após criado um novo projeto, abra o arquivo processdefinition.xml que está na pasta “/processes/simple/” do projeto criado.
Passo 4:	Para instalar o processo no servidor, clique na guia inferior “deployment” e clique em “Deploy Process Archive”.
Resultado:	Agora é só acessar o servidor em http://localhost:8080/jbpm , logar e verificar que o processo aparece na lista de processos e pode ser executado.

Manual original da plataforma JBoss jBPM (em inglês):

<http://wiki.jboss.org/wiki/Wiki.jsp?page=JbpmGettingStarted>

3 Guia de início rápido

Neste guia será demonstrado como fazer seu primeiro processo:

- Como iniciar o servidor;
- Interação com o processo de teste;
- Gerenciar um processo que está sendo executado;
- Criar seu primeiro processo de negócio;
- Instalar e usar seu processo.

Como iniciar o servidor

Passo 1:	Clique em: C:\jbpm-starters-kit-3.1.2\jbpm-server\start.bat
Passo 2:	Acesse o servidor em http://localhost:8080/jbpm
Resultado:	O portal jBPM deve estar funcionando.

Interação com o processo de teste

Utilizaremos como exemplo o processo: “Alunos_Pedido de Notas”

Passo 1:	Entre com os dados de usuário e senha do Aluno	
Passo 2:	Visualize e clique no processo Alunos_Pedido de Notas na lista de processos	
Passo 3:	Preencha os dados solicitados no formulário de requisição	

Requisição							
Nome	<input type="text" value="Guilherme"/>						
Matéria	<input type="text" value="Cálculo 3"/>						
Prova	<input type="text" value="P1"/>						
<input type="button" value="Save and Close Task"/> <input type="button" value="Save"/> <input type="button" value="Cancel"/>							
Passo 4:	A tarefa e um email foram enviados para o Professor 1. Faça o logout clicando em “Sair” e refaça o login com o usuário do Professor1 um para simular uma resposta. Basta clicar na tarefa que aparece na lista de tarefas pendentes e responder o formulário.						
Tarefas não alocadas <table border="1"> <thead> <tr> <th>Nome da tarefa</th> <th>Processo</th> <th>Versão</th> </tr> </thead> <tbody> <tr> <td>Resposta</td> <td>Alunos_Pedido de Notas</td> <td>1</td> </tr> </tbody> </table>		Nome da tarefa	Processo	Versão	Resposta	Alunos_Pedido de Notas	1
Nome da tarefa	Processo	Versão					
Resposta	Alunos_Pedido de Notas	1					
Iniciar nova execução de um Processo <table border="1"> <thead> <tr> <th>Iniciar processo</th> <th>Processo</th> <th>Versão</th> </tr> </thead> </table>		Iniciar processo	Processo	Versão			
Iniciar processo	Processo	Versão					
Resultado:	Agora é só acessar novamente com o usuário Aluno1 e clicar na tarefa correspondente para ver a resposta. Clique em “Save and Close Task” para finalizar o processo.						
Resposta <table border="1"> <thead> <tr> <th>Matéria</th> <th>Cálculo 3</th> </tr> </thead> <tbody> <tr> <td>Prova</td> <td><input type="text" value="P1"/></td> </tr> <tr> <td>Nota</td> <td><input type="text" value="10"/></td> </tr> </tbody> </table>		Matéria	Cálculo 3	Prova	<input type="text" value="P1"/>	Nota	<input type="text" value="10"/>
Matéria	Cálculo 3						
Prova	<input type="text" value="P1"/>						
Nota	<input type="text" value="10"/>						
<input type="button" value="Save and Close Task"/> <input type="button" value="Save"/> <input type="button" value="Cancel"/>							

Gerenciar nm processo que está sendo executado

Autentique-se com um usuário com perfil de administrador e clique no link de monitoração e em seguida Definições de processos.

Nesta tela devemos escolher qual processo e versão queremos buscar instâncias.

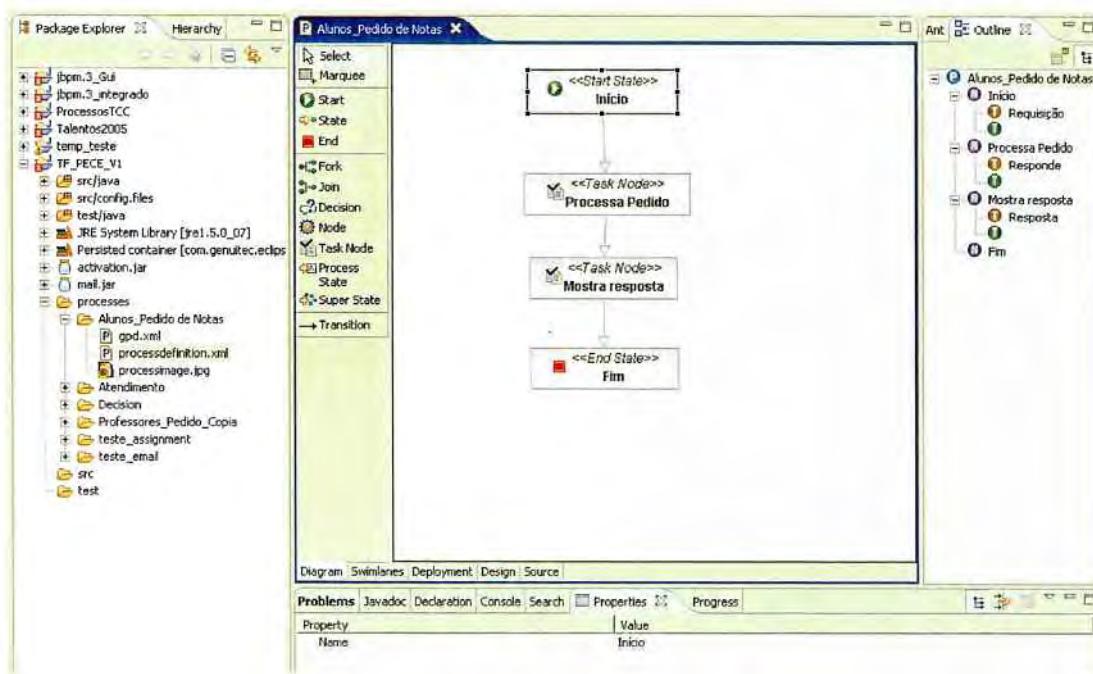
Ao selecionar um processo, serão exibidas todas as instâncias em execução ou finalizadas do processo. Clique em uma, para ver seu histórico, conteúdo das variáveis e alterar o valor de uma variável.

Criar seu primeiro processo de negócio

Abra o Eclipse e crie um novo projeto: clique em File > new > project > JBoss jBPM > Process Project. Escolha um nome e clique em Finish.

Após criado um novo projeto, para criar um processo clique em File > new > other > JBoss jBPM > Process Definition, escolha um nome. Será criada uma pasta para o processo na pasta Processes do projeto.

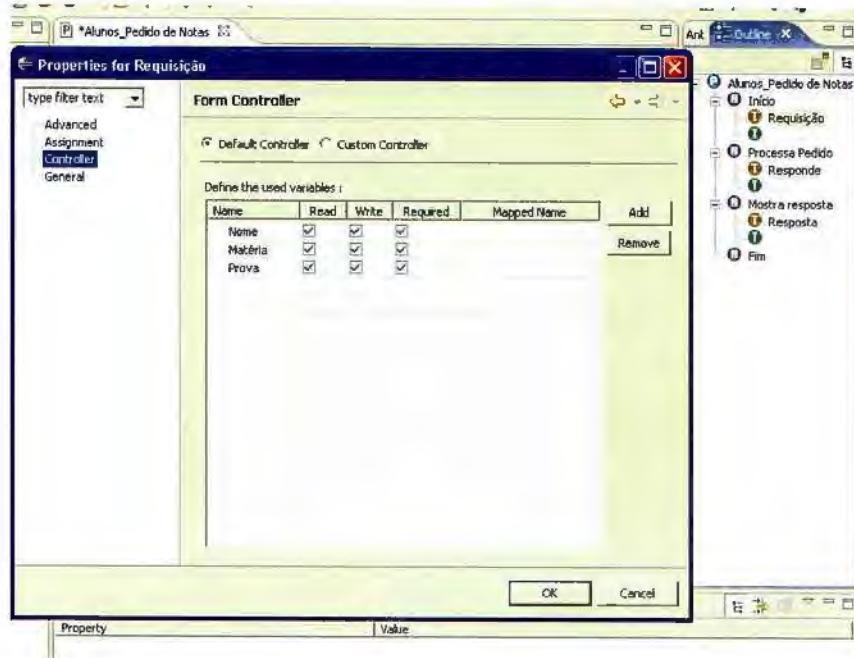
Clique no arquivo processdefinition.xml que é o arquivo que armazena a descrição do projeto em XML.



Clique nos nós disponíveis na barra de ferramentas e clique na área de desenho para adicionar um novo nó. Para alterar suas propriedades basta clicar em cima que aparecerá uma barra de propriedades ou duas vezes para mais detalhes. Nomeie os nós como na figura acima.

Ligue os nós com a ferramenta de transição, clique na guia “Transition” e em cada um dos dois nós em sequência.

Agora precisamos criar tarefas para as transições: clique em window > show view > outline, que mostra a estrutura do processo. Clique com o botão direito no primeiro nó e selecione Add Task. Será criada uma tarefa relacionada com a transição, que é um formulário configurável clicando-se duas vezes na tarefa. Como mostra a figura abaixo:



É necessário configurar seu nome, o Controller que é o formulário e a lógica de atribuição (Assignment).

General > Name: Requisição

Assignment > Type = Expression, Expression = “initiator”

Controller: Adicione os campos para escrita e leitura: Nome, Matéria e Prova.

E assim sucessivamente com as outras transições. Note que desta forma a atribuição é para o usuário que iniciou o processo, também podemos usar user(Nome do usuário) para atribuir a um usuário fixo e nomeado. No formulário, podemos colocar variáveis de tarefas anteriores do processo apenas escrevendo seu nome e selecionando somente leitura e novas variáveis com o nome único e a propriedade *write*.

Instalar e usar seu processo.

Salve seu processo e clique na aba Deployment. Basta selecionar os arquivos e pacotes necessários e clicar em “*Deploy Process Archive*”.

O processo irá aparecer na lista de processos dos usuários obedecendo a seguinte regra:

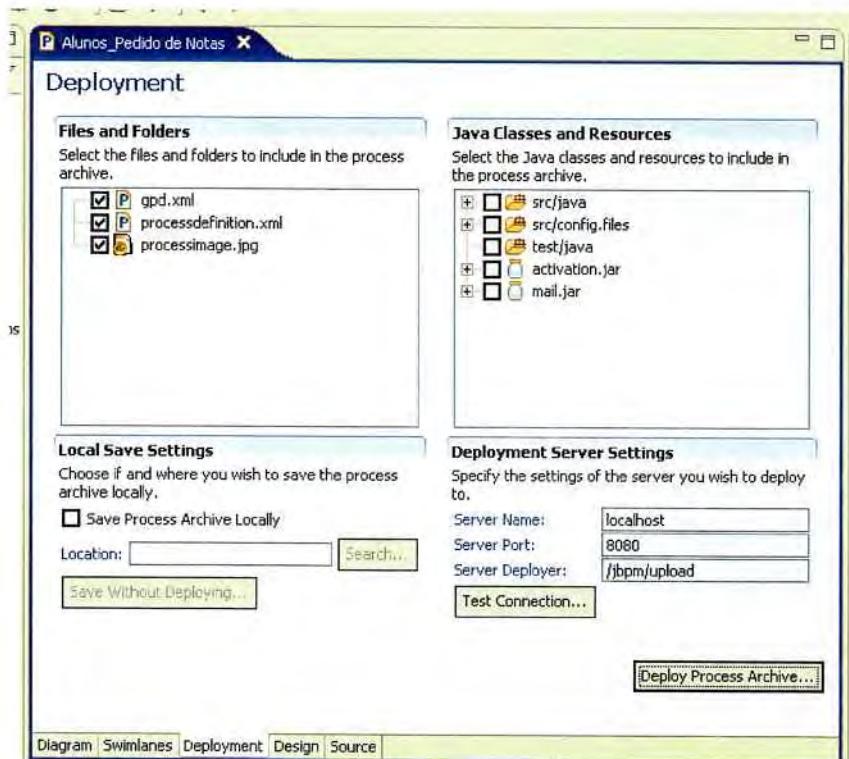
Cada grupo tem permissão para iniciar processos com nomes iniciados pelo nome de seu grupo, conforme explicado à seguir:

Exemplo: **Aluno1** consegue iniciar os processos:

- **Alunos_Requisição de Boletos**
- **Alunos_Revisão de notas**

Mas não aos processos: **Professores_Frequencia** e **Atendentes_Chamado**.

Logo você deve obedecer à padronização e nomear o processo colocando o nome do grupo no início do nome do processo, para que só este grupo tenha permissão para iniciá-lo.



4 Entendendo como criar processos

Os processos são modelados na linguagem jPDL, *x Process Definition Language*, que é uma linguagem expansível e customizável criada pelos desenvolvedores da ferramenta jBPM. Para isso é usado um plug-in para Eclipse, que garante uma interface amigável e padrão.

Para cada processo criado, existe uma interface gráfica que permite ao usuário desenhar o fluxo de atividades arrastando nós da barra de ferramentas, via visão hierárquica dos elementos do processo ou via XML.

Um processo, neste contexto, pode ser definido como uma seqüência de nós que provêm funcionalidades úteis ao processo de negócio. Estas podem ser ligadas sequencialmente ou com em paralelo, e ainda podemos criar eventos que serão disparados ao entrar ou sair de uma tarefa.

Os nós pode ter as seguintes características:

Nó de início (Start-node): Este é o nó que inicia o processo, e por isso sempre deve existir, ele pode ser iniciado por um usuário ou por um timer.

Nó de término (End-node): Este é o nó que finaliza o processo, e por isso sempre deve existir.

Tarefa (task-node): Este é o nó que realiza a interface com o usuário, ele representa uma tarefa que aparece nas listas de tarefas dos usuários e deve ser realizada por um ser humano. É através dele que são criados formulários de entrada de dados, respostas a solicitações e compartilhamento de arquivos, ou seja, o que chamamos de *workflow*.

Bifurcação (Fork): este nó possibilita a utilização de paralelismo, criando dois ou mais fluxos concorrentes, sem que o projetista tenha que se preocupar com a sincronização das tarefas.

Junção (Joint): este nó executa a sincronização de fluxos concorrentes, criando um único fluxo de saída, sem que o projetista tenha que se preocupar com a sincronização de tarefas paralelas.

Decisão (Decision): este nó serve para quando o sistema deve tomar uma decisão, através de uma lógica de decisão dependente das variáveis do processo.

Estado (State): este nó serve como um estado de espera para ser utilizado em sincronizações com sistemas externos.

Nó padrão (Node): este nó é um artefato programável, que pode ser utilizado para expandir as funcionalidades do sistema e fazer customizações.

Além dos nós, também é importante descrever os eventos, que são momentos na execução do processo em que se pode executar listas de ações. Estes momentos são: antes de processar um nó e após seu processamento.

Esta ferramenta serve para incluir ações que não precisam aparecer no desenho do processo, mas são fundamentais para ele funcionar. Por exemplo, podemos atualizar uma base de dados com os dados de um nó após sua execução, mas isto não precisa aparecer para quem modela o processo, pois é um detalhe técnico.

Ações podem ser classes Java do tipo *ActionHandler*, *Scripts* ou um formulário gerado pelo controlador de formulários (chamado de *Task* no programa).

5 FAQ e BUGs (Para o desenvolvedor):

1. Tive alguns problemas ao alterar as classes e não conseguia compilar, recebia o seguinte erro:

```
compile.webapp:
[javac] Compiling 1 source file to C:\jbpm-starters-kit-3.1.2\jbpm\build\classes.webapp
BUILD FAILED
C:\jbpm-starters-kit-3.1.2\jbpm\build.deploy.xml:115: The following error occurred while executing this
line:
C:\jbpm-starters-kit-3.1.2\jbpm\build.xml:88: Error running javac.exe compiler
```

Corrigi este erro alterando o arquivo build.xml:88 na propriedade fork="yes" para fork="no". Não sei exatamente por que, mas desta forma o erro é eliminado e se tiver algum erro de programação ele é mostrado no Tomcat ao acessar o site, com uma descrição mais completa. Pode-se usar isso para diversos pacotes além do webapp.

2. Procedimento para alterar o código e botar a nova versão para rodar no servidor:

- Altere os arquivos necessários(bean, jsp, classes..).
- Arraste o arquivo build.deploy.xml da pasta raiz do projeto para a barra do ANT.
- clique rodar(seta verde) no menu da janela do ANT, verifique a compilação no console.
- Na pasta C:\jbpm-starters-kit-3.1.2\jbpm-server\server\jbpm você deve adicionar arquivos da compilação original que são apagados a cada compilação. Copie a pasta deploy do diretório JBPM original(que veio no zip do starterskit(3.x)) para a pasta deploy gerada no build C:\jbpm-starters-kit-3.1.2\jbpm-server\server\jbpm\deploy, SEM SOBRESCREVER NENHUM ARQUIVO!
- Agora basta rodar o servidor em C:\jbpm-starters-kit-3.1.2\jbpm-server\start.bat

2.1 Procedimento para alterar o código sem parar o servidor:

- Altere os arquivos necessários(bean, jsp, classes..).
- Arraste o arquivo build.deploy.xml da pasta raiz do projeto para a barra do ANT e selecione o pacote build.webapp.
- clique rodar(seta verde) no menu da janela do ANT, verifique a compilação no console.
- copie o arquivo jbpm.war da pasta "C:\jbpm-starters-kit-3.1.2\jbpm\build" para "C:\jbpm-starters-kit-3.1.2\jbpm-server\server\jbpm\deploy".
- Espere alguns segundos para o servidor instalar o pacote e pronto.

6 Outros manuais:

<http://wiki.jboss.org/wiki/Wiki.jsp?page=JbpmGettingStarted>

http://www.jboss.com/services/online_education#integration

<http://www.jboss.com/products/jbpm/docs>

Anexo 7 – WBS do Projeto

WBS	1. Entrega do cronograma e escopo	1.1 Reunião de definição do cronograma 1.2 Avaliação de temas possíveis para o escopo 1.3 Reunião de definição do escopo
	2. Pesquisa de material de BPM e modelagem de processos	2.1 Pesquisa de material na Internet e em livros 2.2 Apresentação dos artigos estudados
	3. Análise de viabilidade	3.1 Definição inicial dos requisitos 3.2 Análise de ferramentas disponíveis 3.3 Detalhamento da abrangência do escopo
	4. Estudo das tecnologias workflow	4.1 Estudo e comparações dos padrões disponíveis 4.2 Teste de ferramentas para desenho e implementação de workflow 4.3 Exemplos de processos de workflow
	5. Modelagem dos processos do PECE	5.1 Estudo das atividades do PECE 5.2 Modelagem de processos 5.3 Validação da modelagem 5.4 Verificação de tarefas sujeitas à automação
	6. Levantamento dos requisitos	6.1 Levantamento dos requisitos funcionais 6.2 Levantamento dos requisitos não funcionais 6.3 Levantamento de requisitos com usuários 6.4 Validação com stakeholders 6.5 Confronto de requisitos escolhidos com padrões do mercado
	7. Definição da tecnologia	7.1 Definição da arquitetura 7.2 Definição da linguagem e plataforma 7.3 Vantagens e desvantagens
	8. Desenvolvimento da ferramenta	8.1 Definição das camadas e funcionalidades 8.2 Priorização da implementação das funcionalidades 8.3 Definição do cronograma de implementação e entregas dos módulos 8.4 Prototipação de interface
	9. Implementação	9.1 Levantamento de infra-estrutura necessária 9.2 Programação dos módulos 9.3 Testes unitários e de integração 9.4 Instalação 9.5 Documentação

Anexo 8 – Cronograma de Atividades