



CENTRO UNIVERSITÁRIO UNIVATES

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

CURSO DE SISTEMAS DE INFORMAÇÃO

AMADEU WEIRICH

**PROMAP: UMA FERRAMENTA PARA APOIAR O  
MAPEAMENTO E DISSEMINAÇÃO DE PROCESSOS DE  
NEGÓCIO**

Lajeado

2012

AMADEU WEIRICH

## **PROMAP: UMA FERRAMENTA PARA APOIAR O MAPEAMENTO E DISSEMINAÇÃO DE PROCESSOS DE NEGÓCIO**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Tecnológicas do Centro Universitário UNIVATES, como requisito parcial para a obtenção do título de bacharel em Sistemas de Informação.

Área de concentração: CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

ORIENTADOR: PABLO DALL'OGGIO

Lajeado

2012

AMADEU WEIRICH

## **PROMAP: UMA FERRAMENTA PARA APOIAR O MAPEAMENTO E DISSEMINAÇÃO DE PROCESSOS DE NEGÓCIO**

Este trabalho foi julgado adequado para a obtenção do título de bacharel em Sistemas de Informação do CETEC e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Pablo Dall'Oglio, UNIVATES

Mestre pela Universidade do Vale do Rio dos

Sinos – São Leopoldo, Brasil

Banca Examinadora:

Prof. Evandro Franzen, UNIVATES

Mestre pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil.

Prof. Paulo Roberto Mallmann, UNIVATES

Mestre pela Universidade do Vale do Rio dos Sinos – São Leopoldo, Brasil.

Coordenador do curso de Sistemas de Informação : \_\_\_\_\_

Prof. Evandro Franzen

Lajeado, Julho 2012.

Dedico este trabalho aos meus pais, em especial pela dedicação e apoio em todos os momentos difíceis.

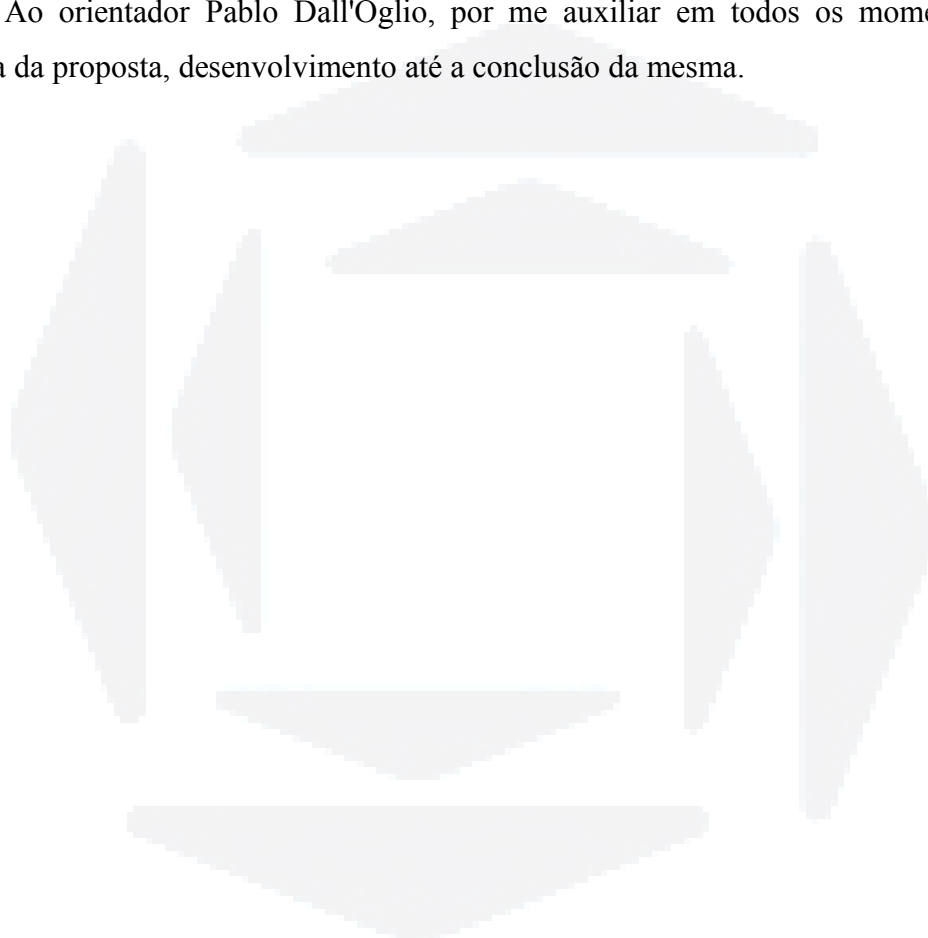


## **AGRADECIMENTOS**

Aos meus pais que sempre estiveram presentes apoiando-me no que fosse necessário.

À Univates, instituição em que trabalhei, universidade pela qual busco adquirir o título de Bacharel em Sistemas de Informação.

Ao orientador Pablo Dall'Oglio, por me auxiliar em todos os momentos, desde a escolha da proposta, desenvolvimento até a conclusão da mesma.



## RESUMO

A crescente quantidade e complexidade dos processos de negócio das organizações atuais, juntamente com a alta rotatividade dos quadros funcionais e a utilização de sistemas informatizados, tem levado as organizações a um cenário com problemas de identificação e organização de seus processos de negócio. Assim, surge a necessidade de organizar os processos de negócio para difundir seu conhecimento na organização, bem como permitir a melhoria dos processos existentes. Para apoiar a organização e disseminação dos processos de negócio, este trabalho propõe o desenvolvimento de um modelo para representação de processos, uma arquitetura para intercâmbio de processos de negócio entre diferentes aplicações, bem como uma ferramenta colaborativa que permitirá a disseminação dos processos e a sua documentação.

**Palavras-chaves:** BPM, BPMN, Mapeamento de Processos de Negócio, Sistemas de Informação.

## ABSTRACT

The growing quantity and complexity of business processes from the current organizations, along with the big turnover and the strong use of automated systems, has led the organizations to a scenario with problems like identification, organization and improvement of its business processes. This way, there are needs to organize the business process to disseminate this knowledge inside the organization, besides to allow the improvement of existing processes. To support the organization and dissemination of business processes, this work proposes the development of a model for process representation, an architecture for process interchange among different applications, besides a collaborative tool that will allow the process dissemination and its documentation.

**Keywords:** BPM, BPMN, Business Processes, Systems Information.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>14</b>
1.1	Objetivo.....	15
1.2	Metodologia.....	15
1.3	Roteiro do restante do trabalho.....	16
<b>2</b>	<b>REVISÃO DE LITERATURA.....</b>	<b>17</b>
2.1	Processo.....	17
2.2	Gestão por Processos.....	18
2.3	Mapeamento de Processos.....	20
2.3.1	Matriz RACI.....	21
2.4	BPM - Business Process Management.....	22
2.4.1	Processo de Planejamento e Estratégia.....	23
2.4.2	Análise de Processos de Negócio.....	23
2.4.3	Desenho de Modelagem de Processos de Negócio.....	23
2.4.4	Implementação de Processos.....	24
2.4.5	Monitoramento e Controle de Processos.....	24
2.4.6	Refinamento de Processos.....	24
2.5	BPMN - Business Process Modeling Notation.....	24
2.5.1	Swimlanes.....	27
2.5.2	Objetos de Fluxo.....	27
2.5.2.1	Evento.....	27
2.5.2.2	Atividade.....	30
2.5.2.3	Gateway.....	31
2.5.3	Objetos de Conexão.....	33
2.5.4	Artefatos.....	35
2.6	SOA e Serviços Web.....	36
2.6.1	Interoperabilidade.....	37
2.6.2	Desempenho.....	37
2.6.3	Segurança.....	38
2.6.4	Garantia.....	39
2.6.5	Disponibilidade.....	39
2.6.6	Modificabilidade.....	40
2.6.7	Testabilidade.....	40
2.6.8	Usabilidade.....	41
2.6.9	Escalabilidade.....	41
2.7	XML Process Definition Language.....	41
2.8	UML – Unified Modeling Language.....	42
2.8.1	Diagrama de Casos de Uso.....	43
2.8.2	Diagrama de Classes.....	44
2.8.3	Diagrama de Componentes.....	45
<b>3</b>	<b>TRABALHOS RELACIONADOS.....</b>	<b>47</b>
3.1	Modelos.....	47



3.1.1 Artigo: A Basing on Model-Driven Framework of Service-Oriented Software Production Line.....	47
3.1.2 Artigo: Modeling and Validating BPMN Diagrams.....	48
3.2 Ferramentas.....	48
3.2.1 BizAgi.....	49
3.2.2 Bonita Open Solution.....	50
3.2.3 Graphviz.....	50
<b>4 TRABALHO PROPOSTO.....</b>	<b>52</b>
4.1 Visão Geral.....	52
4.2 Arquitetura da Ferramenta.....	53
4.3 Papéis e Responsabilidades.....	54
4.4 Modelo de Domínio.....	56
4.5 Modelo de classes.....	57
4.5.1 Utilização das classes.....	62
4.6 Modelo Relacional.....	64
4.7 Especificação dos Casos de Uso.....	68
4.7.1 Importar Processo (XPDL).....	68
4.7.2 Visualizar processo.....	69
4.7.3 Documentar processo.....	71
4.7.4 Navegar nos processos.....	72
4.7.5 Exportar documento.....	73
4.8 Arquitetura dos serviços.....	75
4.8.1 Utilização dos webservices.....	77
4.9 Tecnologias utilizadas.....	80
4.10 Validação da ferramenta.....	82
4.11 Trabalhos futuros.....	85
<b>5 CONCLUSÃO.....</b>	<b>86</b>
<b>REFERÊNCIAS.....</b>	<b>87</b>
<b>APÊNDICES.....</b>	<b>89</b>

## LISTA DE FIGURAS

Figura 1 Pesquisa sobre investimento de organizações em Processos (INSADI, 2008)....	15
Figura 2 Organograma organizacional vertical.....	19
Figura 3 Organograma organizacional horizontal.....	19
Figura 4 Pesquisa sobre barreiras para implementação da visão funcional de Processos em organizações (INSADI, 2008).....	20
Figura 5 Exemplo de matriz de papéis e responsabilidade (SANTOS, 2009B).....	22
Figura 6 Ciclo de vida BPM (PIRES, 2011).....	23
Figura 7 Exemplo de modelagem de um processo interno.....	25
Figura 8 Exemplo de modelagem de um processo abstrato.....	26
Figura 9 Exemplo de modelagem de um processo de colaboração.....	26
Figura 10 Exemplo de Eventos, elementos de Objetos de Fluxo da BPMN.....	28
Figura 11 Exemplo de Atividades, elementos de Objetos de Fluxo da BPMN.....	30
Figura 12 Exemplo de Atividades com esteriótipos, elementos de Objetos de Fluxo da BPMN.....	30
Figura 13 Exemplo de atividade com subprocesso fechado, elementos de Objetos de Fluxo da BPMN.....	31
Figura 14 Exemplo de atividade com subprocesso aberto, elementos de Objetos de Fluxo da BPMN.....	31
Figura 15 Exemplo de Gateway Exclusive-or, elementos de Objetos de Fluxo da BPMN. ....	32
Figura 16 Exemplo de Gateway Inclusive-or, elementos de Objetos de Fluxo da BPMN. ....	33
Figura 17 Exemplo de Gateway Parallel, elementos de Objetos de Fluxo da BPMN.....	33
Figura 18 Exemplo da utilização de Objetos de Conexão da BPMN.....	35
Figura 19 Exemplo da utilização de Artefatos da BPMN.....	36
Figura 20 Exemplificação de utilização do método SOA (W3C, 2002).....	37
Figura 21 Exemplo de um código XPD.....	42
Figura 22 Elementos de um Diagrama de Caso de Uso.....	43
Figura 23 Exemplo de um Diagrama de Caso de Uso.....	43
Figura 24 Exemplo de Classe.....	44
Figura 25 Exemplo de um Diagrama de Classes.....	45
Figura 26 Elemento que representa um Componente em um Diagrama de Componentes .....	45
Figura 27 Exemplo de um Diagrama de Componentes.....	46
Figura 28 Meta modelo de BPMN de acordo com CHAO (2009). ....	47
Figura 29 Meta modelo de BPMN de acordo com CHINOSI e TROMBETTA (2009). .	48
Figura 30 Software BizAgi .....	49
Figura 31 Software Bonita Open Solution .....	50
Figura 32 Exemplo de saída do Graphviz.....	51
Figura 33 Visão geral.....	53
Figura 34 Diagrama de Modelo de Arquitetura (MVC).....	54
Figura 35 Diagrama de Casos de Uso.....	55
Figura 36 Proposta de Diagrama de Domínio.....	57

<b>Figura 37 Diagrama de Classes.....</b>	<b>58</b>
<b>Figura 38 Criação de processo utilizando as Classes.....</b>	<b>63</b>
<b>Figura 39 Carregamento de processo utilizando as Classes.....</b>	<b>63</b>
<b>Figura 40 Exclusão de processo utilizando as Classes.....</b>	<b>64</b>
<b>Figura 41 Diagrama E.R. do ProMap.....</b>	<b>64</b>
<b>Figura 42 Primeiro passo para importar um processo.....</b>	<b>68</b>
<b>Figura 43 Segundo passo para importar um processo.....</b>	<b>69</b>
<b>Figura 44 Primeiro passo para visualizar um processo.....</b>	<b>70</b>
<b>Figura 45 Segundo passo para visualizar um processo.....</b>	<b>70</b>
<b>Figura 46 Tela para inserção de anotações.....</b>	<b>72</b>
<b>Figura 47 Navegando nos processos.....</b>	<b>73</b>
<b>Figura 48 Documento em formato HTML.....</b>	<b>74</b>
<b>Figura 49 Documento em formato PDF.....</b>	<b>75</b>
<b>Figura 50 Diagrama de Componentes.....</b>	<b>76</b>
<b>Figura 51 Webservice chamando o método listProcess().....</b>	<b>78</b>
<b>Figura 52 Webservice chamando o método getProcess().....</b>	<b>79</b>
<b>Figura 53 Webservice chamando o método getProcessFlow().....</b>	<b>79</b>
<b>Figura 54 Webservice chamando o método getProcessImage().....</b>	<b>80</b>
<b>Figura 55 Respostas do formulário disponibilizado online.....</b>	<b>83</b>

## LISTA DE TABELAS

Tabela 1 Descrição dos elementos de maturidade de processos (PAIM; CARDOSO; CAULLIRAUX, 2009).....	18
Tabela 2 Descrição dos elementos Swimlanes da notação BPMN.....	27
Tabela 3 Descrição dos elementos de Objetos de Fluxo da notação BPMN.....	28
Tabela 4 Descrição dos objetos de Gateway da notação BPMN.....	31
Tabela 5 Descrição dos elementos de Objetos de Conexão da notação BPMN.....	34
Tabela 6 Descrição dos artefatos da notação BPMN.....	35
Tabela 7 Descrição dos Casos de Uso.....	55
Tabela 8 Descrição da classe PProcess.....	58
Tabela 9 Descrição da classe PPool.....	59
Tabela 10 Descrição da classe PSwimlane.....	60
Tabela 11 Descrição da classe PFlowObject.....	60
Tabela 12 Descrição da classe PArtifact.....	61
Tabela 13 Descrição da classe PConnectObject.....	62
Tabela 14 Descrição da tabela pro_usuarios.....	65
Tabela 15 Descrição da tabela pro_papeis.....	65
Tabela 16 Descrição da tabela pro_usuarios_papeis.....	65
Tabela 17 Descrição da tabela pro_process.....	65
Tabela 18 Descrição da tabela pro_pool.....	66
Tabela 19 Descrição da tabela pro_swimlane.....	66
Tabela 20 Descrição da tabela pro_flow_object.....	66
Tabela 21 Descrição da tabela pro_connect_object.....	67
Tabela 22 Descrição da tabela pro_artifact.....	67
Tabela 23 Descrição da tabela pro_flow_object_artifact.....	67
Tabela 24 Descrição do caso de uso Importar Processo (XPDL).....	68
Tabela 25 Descrição do caso de uso Visualizar processo.....	69
Tabela 26 Descrição do caso de uso Documentar processo.....	71
Tabela 27 Descrição do caso de uso Navegar nos processos.....	72
Tabela 28 Descrição do caso de uso Exportar documento.....	73
Tabela 29 Descrição dos serviços disponibilizados pelo Promap.....	77
Tabela 30 Descrição dos métodos da interface Graphvz.....	77
Tabela 31 Descrição dos métodos da interface DomPdf.....	77

## LISTA DE ABREVIATURAS

BPM: Business Process Management  
BPMS: Business Process Management Suite  
BPMN: Business Process Modeling Notation  
HTML: HyperText Markup Language  
MVC: Model, View e Controller  
OMG: Object Management Group  
SGBD: Sistema de Gerenciamento de Banco de Dados  
SLA: Service Level Agreement  
SOA: Service Oriented Architecture  
SOAP: Simple Object Access Protocol  
SSL: Secure Socket Layer  
UML: Unified Modeling Language  
WfMC: Workflow Management Coalition  
WS-I: Web Services-Interoperability Organization  
WSDL: Web Service Definition Language  
XML: Extensible Markup Language  
XPDL: XML Process Definition Language

## 1 INTRODUÇÃO

As organizações, independente de seu porte, são estruturadas por meio de processos de negócios que possuem o objetivo de agregar valor a seus clientes. No contexto atual, é possível localizar problemas comuns à maioria das organizações, como a crescente quantidade e complexidade de seus processos de negócios, a alta rotatividade de seus quadros funcionais e a falta de documentação de seus processos.

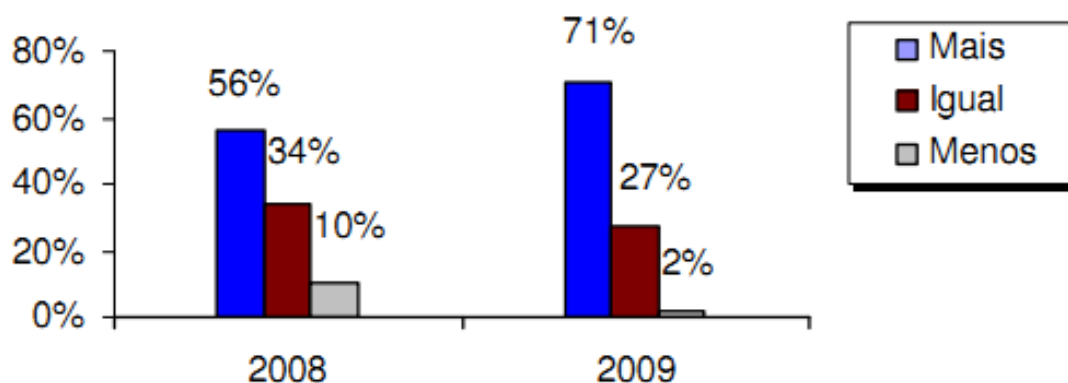
A alta complexidade e a falta de documentação dos processos de negócios, aliada à rotatividade do quadro funcional, acaba causando instabilidades operacionais às organizações. Isso se deve à falta de formalização do conhecimento acerca dos processos de negócio, ou seja, o conhecimento é tácito, não explícito.

Um dos principais agravantes para os problemas gerados pela troca do quadro funcional é a falta de documentação dos processos de negócios. Para ter-se uma documentação concisa e precisa, inicialmente a organização precisará adotar gestão por processos: trata-se de uma metodologia orientada a processos, cujo objeto é modificar o modo de trabalho da organização, saindo dos métodos setoriais e iniciando com o método de processos. O tema gestão por processos será melhor definido na sequência.

Baseado nas ideias de Santos (2009B), destacam-se alguns motivos pelos quais as organizações estão optando mais pela utilização de metodologias, como gestão por processos:

- Redução de custos;
- Vantagem competitiva;
- Aumento da satisfação do cliente;
- Busca por inovação;
- Controle de recursos;
- Alinhamento/integração entre as unidades de negócio.

Na Figura 1, é demonstrada a pesquisa realizada pela INSADI (2008), exibindo os investimentos feitos pelas organizações brasileiras na área de processos em 2008 e a tendência para 2009.



**Figura 1 Pesquisa sobre investimento de organizações em Processos (INSADI, 2008).**

Com o intuito de apoiar a implementação de metodologias de processos de negócios, as organizações estão utilizando a informática na automação destes. Entretanto, com a utilização da informática surgem problemas relativos à estruturação dos processos. O analista de sistemas produz o software solicitado, não reestrutura os processos, apenas faz o que é solicitado pelo cliente. Muitos casos resultarão em processos com atividades desnecessárias, mas que acabam sendo implementadas (gerando custos) desnecessários. Contudo, de acordo com a Figura 1, esse tipo de procedimento tende a melhorar, pois, quanto mais investimentos em processos por parte das organizações, melhores os resultados obtidos.

### 1.1 Objetivo

O objetivo da presente monografia é apoiar o mapeamento dos processos de negócio de uma organização, por meio de um modelo para representação de processos de negócio e uma ferramenta para gerir esses processos.

A ferramenta será desenvolvida em plataforma *web*, disponibilizará um local centralizado que irá conter todos os processos da organização, tornando simples e rápida a consulta e navegação de um determinado processo. Além disso, essa ferramenta irá dispor de interfaces que facilitarão sua conexão com outras ferramentas voltadas à gestão de processos de forma simplificada, permitindo, assim, estender facilmente suas funcionalidades.

Com a facilidade de acesso à ferramenta, esta irá auxiliar colaboradores da organização a conhecerem melhor suas atividades e os processos nos quais estão inseridos, bem como melhorar a especificação dos mesmos. Os resultados do mapeamento correto dos processos poderão ser vistos posteriormente, quando as organizações terão solucionado seus problemas de conhecimento de processos em relação à troca do quadro funcional.

Mesmo sabendo que já existem softwares com esse objetivo, o nosso diferencial será criar uma estrutura SOA com maior flexibilidade para interação com outros softwares.

## **1.2 Metodologia**

Para o desenvolvimento do presente trabalho, a seguinte metodologia será seguida:

- Identificar e analisar as principais características e técnicas sobre Gestão por Processos;
- Identificar e analisar as principais características e técnicas sobre Mapeamento de Processos;
- Analisar a metodologia BPM, aprofundando a análise em BPMN;
- Estudar técnicas relacionadas a Arquiteturas Orientadas a Serviços;
- Desenvolver uma ferramenta que apoie o mapeamento e a disseminação de processos de negócios organizacionais.

## **1.3 Roteiro do restante do trabalho**

Para a melhor compreensão da proposta, seus capítulos foram divididos na seguinte ordem de apresentação:

No capítulo 2, é feita a revisão de literatura que conta com conteúdos descritivos sobre Processos, Gestão por Processos, Mapeamento de Processos, BPM, BPMN, SOA, XPDL e UML.

O capítulo 3 irá apresentar trabalhos relacionados, demonstrando alguns modelos e ferramentas utilizadas para a definição da proposta.

No capítulo 4, são definidos diversos modelos que foram utilizados para o desenvolvimento da mesma ferramenta.



## 2 REVISÃO DE LITERATURA

Para alcançar os objetivos do trabalho, foi necessário envolver diversos conhecimentos e tecnologias que serão representados neste capítulo para um maior entendimento do trabalho proposto. Cada tópico é um fragmento importante para a compreensão de toda a proposta.

### 2.1 Processo

As diversas definições para processos: gestão de processos, gestão por processos, visão por processos, orientação por processo não se apresentam tão claras e confundem-se na literatura, sendo assim, não permitem uma definição singular. Dessa forma, a seguir, serão apresentadas algumas definições para processos:

Michaelis (2011) define processo como:

**1** Ato de proceder ou de andar. **2** Sucessão sistemática de mudanças numa direção definida. **4** Seguimento, decurso: *O processo dos tempos*. **5** Série de ações sistemáticas visando a certo resultado: *O processo de fazer vinho*. **14** Conjunto dos papéis relativos a um negócio.

Hammer e Champy (1994) apud Paim, Cardoso, Caulliraux (2009) definem processos como:

Um conjunto de atividades que juntas produzem um resultado de valor para um consumidor. Para esses autores, processos são os que as organizações fazem.

Nagel e Rosemann (1999) apud Paim, Cardoso, Caulliraux (2009) ressaltam que atividades têm responsáveis, mas processos, como um todo, não têm um responsável.

De acordo com as definições, depreende-se que tudo o que fazemos pode ser considerado processo, desde uma saída de casa para comprar pão, até a manutenção de um automóvel. Para definir corretamente cada parte de um processo, podemos separá-lo em elementos centrais:

- Ação: é um elemento que representa a atividade a ser realizada.
- Recursos: podem ser as pessoas que realizam atividades que, por sua vez, transformam ou movimentam o objeto em fluxo.
- Objeto em fluxo: é tudo aquilo que está sendo transferido de uma ação para outra, desde materiais, informações, capital, conhecimento, ideias.

Após a conceituação de processos e seus elementos, serão definidos, na Tabela 1, os níveis de maturidade para os processos.

**Tabela 1 Descrição dos elementos de maturidade de processos (PAIM; CARDOSO; CAULLIRAUX, 2009)**

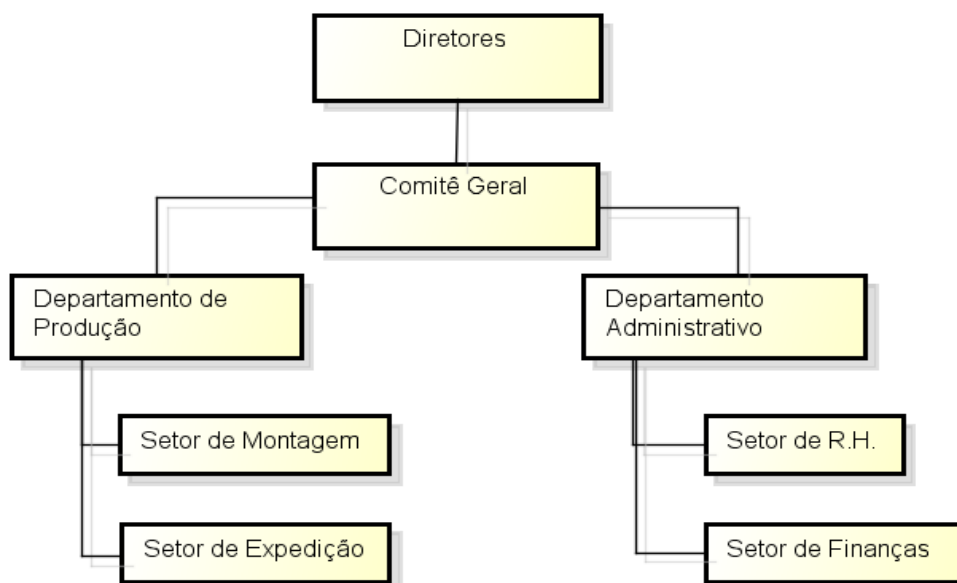
Elemento	Descrição
Ad hoc	Processos disparados por demandas não frequentes; não há uma estruturação do processo.
Repetitivos	Processos realizados com muita frequência, geralmente no dia a dia, não são documentados, estão internalizados na experiência das pessoas.
Normatizados	Processos realizados com frequência e cuja sequência de atividades está bem definida e já documentada.
Mensurados	Processos que além de documentados e normatizados já apresentam um conjunto de indicadores para medição de seu desempenho.
Geridos	Processos já mensurados e que, de acordo com seus indicadores de desempenho, já passaram e continuam passando por ciclos de melhoria e inovação.

## 2.2 Gestão por Processos

Muitas organizações possuem uma estrutura organizacional definida em setores, denominada visão funcional, ou também vertical. A proposta da gestão por processos é modificar essa visão para horizontal, chamada de visão por processos.

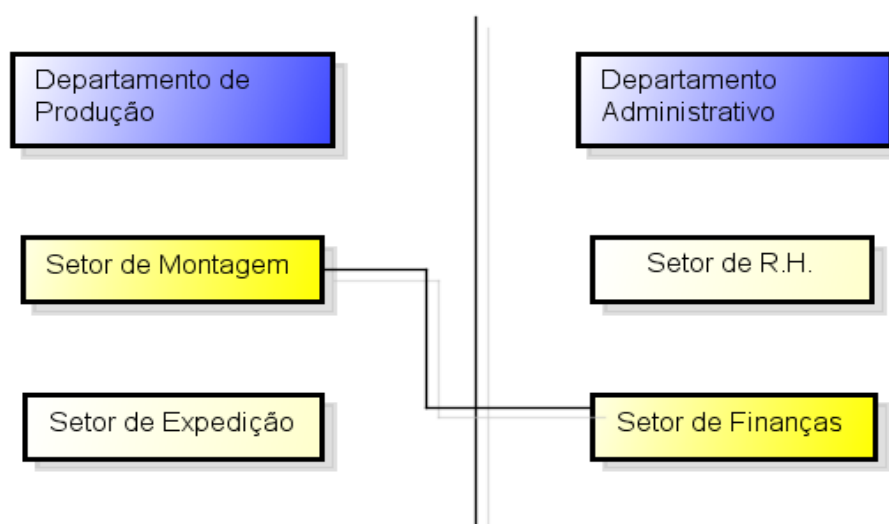
A estrutura vertical surgiu para que houvesse uma divisão da organização em grupos que realizavam atividades semelhantes, como em uma fábrica de carros, dividindo a linha de produção em vários setores, em que cada um consegue efetuar suas atividades de forma independente.

A Figura 2 exemplifica uma fábrica de carros em que o setor de montagem tem um processo que deve passar em sequência pelos setores: Departamento de produção, Comitê geral, Departamento administrativo para, finalmente, chegar ao Setor de finanças.



**Figura 2 Organograma organizacional vertical.**

De acordo com De Sordi (2009), a visão funcional trabalha com o foco em setores que necessitam desenvolver passos para se chegar ao próximo setor, reportando-se primeiramente ao chefe. Já a visão por processos utiliza a troca de informações entre setores de forma direta. Nesse caso cada pessoa exerce sua atividade e a encaminha para o responsável da atividade seguinte, como exemplificado na Figura 3. O Setor de montagem se comunica diretamente com o Setor de finanças, sendo esse o caminho direto do processo que está simulado na Figura 3.

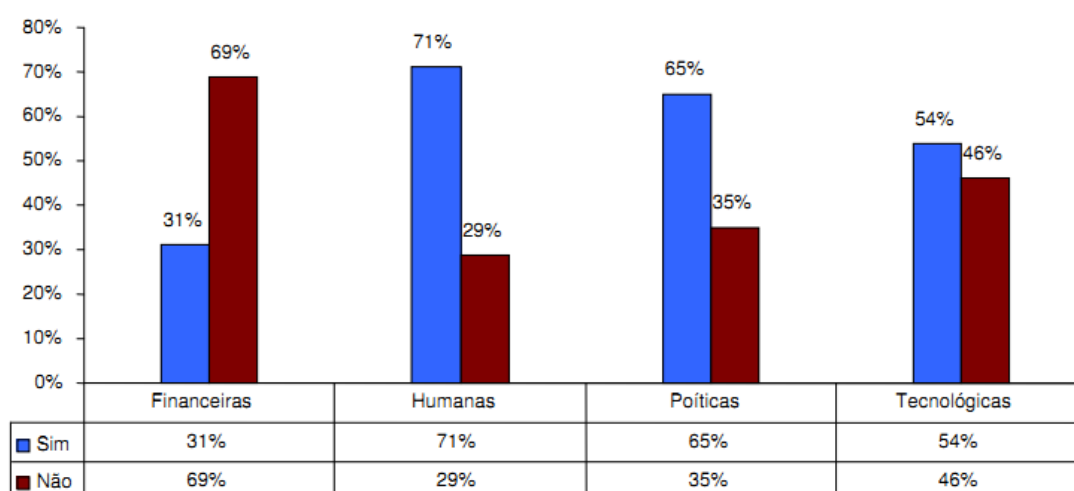


**Figura 3 Organograma organizacional horizontal.**

Com a adoção da visão por processos, a estrutura da organização continua a mesma, com cada setor tendo suas devidas funções, a diferença estará na autonomia que os

responsáveis por cada atividade terão. Uma atividade passará de um nível setorial para processual, sendo que todos os envolvidos saberão que estão fazendo parte de um processo e não apenas de uma atividade, entenderão mais amplamente os motivos da atividade estar sendo efetuada, assim como os caminhos pelos quais esse processo percorre (DE SORDI, 2009).

A Figura 4 apresenta uma pesquisa feita pela INSADI (2008) que busca representar as principais barreiras para a implementação da forma de gestão funcional para a visão por processos. Na Figura 4, podemos constatar que o investimento financeiro não é um impasse, porém barreiras como humanas e políticas têm grande peso para a adoção da gestão por processos. Muitas vezes, as pessoas criam essas barreiras, pois temem ser controladas quantitativamente, além de não verem vantagem em modificar a forma com que as coisas estão sendo feitas (DE SORDI, 2009).



**Figura 4 Pesquisa sobre barreiras para implementação da visão funcional de Processos em organizações (INSADI, 2008).**

Algumas vantagens que podem ser apresentadas a colaboradores para aderirem a esta nova visão, de acordo com De Sordi (2009):

- Facilidade para treinamento de colaboradores novos;
- Facilidade de entendimento do processo;
- Facilidade para encontrar responsáveis por atividades;
- Facilidade para encontrar informações sobre as atividades e processos;
- Facilidade para localizar o ponto em que um processo encontra-se parado.

## 2.3 Mapeamento de Processos

O mapeamento de processos é uma ferramenta utilizada em organizações que desejam migrar da visão funcional para a visão por processos.

É na etapa de mapeamento de processos que os analistas vão adquirir visibilidade e conhecimento sobre determinadas tarefas, definindo, assim, missão e objetivos, responsabilidades, entradas e saídas, fornecedores e clientes (PAIM; CARDOSO; CAULLIRAUX, 2009).

A realização de uma análise crítica poderá responder a diversas perguntas, tais como:

- Este processo é realmente necessário? Agrega valor?
- Qual é o impacto do processo para a organização?
- Como está seu desempenho? Como medir a performance?
- Pode ser melhorado? Atende aos objetivos definidos?

Com o mapeamento de processos tem-se, como resultado, uma visão atual, consegue-se efetuar melhorias, sempre com o intuito de simplificar as tarefas.

Nas definições de Business Process Management (BPM), veremos os modelos AS-IS e TO-BE, quando o assunto mapeamento de processos será mais detalhadamente explicado.

### 2.3.1 Matriz RACI

Em todos os processos existem papéis e responsabilidades, para tal, ambos são definidos da seguinte maneira:

- Papéis desempenham atividades que constituem um processo, podem ser pessoas com devidos cargos.
- Responsabilidade é aquilo que deve ser feito por um papel envolvido no processo.

A partir da definição de papéis e responsabilidades, na Figura 5, é demonstrada a utilização da matriz RACI (SANTOS, 2009B). Nesta Figura, cada letra, R, A, C, I tem um significado para o papel e processo onde está posicionada.

- R - Responsável pela execução da tarefa;
- A - *Accountable*, aquele que presta contas;
- C - Consultado, aquele é que é consultado para a efetivação da tarefa.
- I - Informado, aquele que é informado sobre a execução de determinada tarefa.

Processos/ Atividades	Papel					
	Vendedor	Analista Produção	Operador	Analista Qualidade	Entregador	Cliente
Efetivar Pedido	R/A	I				
Planeja Produção		R	I			
Produzir Produto			R	I		
Avaliar Qualidade	I	I	I	R		
Entregar Produto	C				R	I

Figura 5 Exemplo de matriz de papéis e responsabilidade (SANTOS, 2009B).

## 2.4 BPM - Business Process Management

BPM é traduzido como Gerenciamento de Processos de Negócios e, de acordo com a ABPMP (2009):

BPM é uma abordagem disciplinada para identificar, desenhar, executar, documentar, medir, monitorar, controlar e melhorar processos de negócio automatizados ou não para alcançar os resultados pretendidos consistentes e alinhados com as metas estratégicas de uma organização.

Entretanto, o BPM define alguns passos disciplinados para que os processos da empresa sejam remodelados a ponto de trazer melhores benefícios para a organização.

O BPM não necessita de tecnologia para ser implantada, pois se trata de uma metodologia, entretanto a tecnologia está cada vez mais presente como auxiliadora em todas as etapas que fazem parte do BPM. Com o passar do tempo, organizações foram desenvolvendo ferramentas auxiliares no processo de implantação de BPM. Elas começaram criando ferramentas que desenhavam processos, passando para implementação de ferramentas de *workflow*, seguindo para ferramentas de monitoramento de processos, enfim, criando toda a estrutura necessária para o ciclo BPM. Essas ferramentas, hoje, são conhecidas como Business Process Management Suite (BPMS) (ABPMP, 2009).

O BPM é dividido em 6 etapas para rodar o ciclo completo de sua implantação, também conhecido como ciclo de vida BPM, são elas: Processo de Planejamento e Estratégia, Análise de Processos de Negócio, Desenho de Modelagem de Processos de Negócio, Implementação de Processos, Monitoramento e Controle de Processos e Refinamento de Processos. Nos tópicos seguintes, serão apresentadas as descrições de cada uma das etapas.

A Figura 6 representa a ordem pela qual as atividades do ciclo de vida BPM são organizadas.



**Figura 6** Ciclo de vida BPM (PIRES, 2011).

#### **2.4.1 Processo de Planejamento e Estratégia**

Processo de Planejamento e Estratégia é a etapa inicial e é definida, principalmente, pelos gerentes da organização. Nesse processo são definidos os objetivos e metas centrais da organização. A conciliação entre os processos, os objetivos e as metas da organização é muito importante, pois, dessa forma, pode-se definir prioridades entre os processos, conseguindo definir os mais prioritários, fazendo com que a organização siga os objetivos estipulados.

#### **2.4.2 Análise de Processos de Negócio**

Esta etapa incorpora várias metodologias para entender os processos atuais no contexto das metas e objetivos desejados.

A análise busca todas as informações possíveis sobre os processos, a fim de compreendê-los melhor.

### **2.4.3 Desenho de Modelagem de Processos de Negócio**

O desenho de modelagem de processos de negócio é a etapa em que os processos organizacionais são especificados, seguindo uma notação. A Business Process Modeling Notation (BPMN) é a notação padrão de acordo com ABPMP (2009).

Durante a etapa de modelagem de processos, existem dois modelos a serem seguidos: o AS-IS e o TO-BE.

- O modelo AS-IS modela o processo da maneira como ele é feito atualmente, utilizando os mesmos envolvidos, rotas e documentos.
- O modelo TO-BE modela o processo da maneira como ele deveria ser feito, já pensando nas melhores práticas a serem adotadas.

Mesmo que o modelo TO-BE seja mais atrativo, algumas empresas não conseguem implementá-lo diretamente, assim, criando, um “meio termo” entre os dois. Dessa forma, conseguem adaptar seu modelo atual, simplificando e melhorando o processo.

### **2.4.4 Implementação de Processos**

A etapa de Implementação de Processos parte da ideia de que o processo foi mapeado e ajustado corretamente na etapa anterior. Em caso positivo, começa-se a implementá-lo, modificando fluxos antigos, adicionando novas tarefas e demais alterações que tenham de ser feitas. Somente pequenos ajustes devem ocorrer durante a implementação.

### **2.4.5 Monitoramento e Controle de Processos**

O Monitoramento e o Controle de Processos fornecem informações sobre o desempenho efetuado nas tarefas, sendo positivos ou negativos, dependendo das metas e objetivos elencados na primeira etapa. A análise dos desempenhos pode efetuar modificações de reengenharia nos processos.

### **2.4.6 Refinamento de Processos**

O monitoramento constante dos processos juntamente com a análise de desempenho são fatores que definirão se os processos devem ou não passar pela etapa de Refinamento de



Processos. Essa etapa é responsável por fazer pequenos ajustes e melhorias no processo. Caso seja constatado que o processo não está funcionando corretamente, o mesmo deverá voltar para a etapa de modelagem e começar o ciclo novamente.

## 2.5 BPMN - Business Process Modeling Notation

Michaelis (2011) define notação como:

1 Ato ou efeito de notar. 2 Sistema de representação ou designação convencional. 3 Conjunto de sinais com que se faz essa representação ou designação.

Notação de BPM, ou BPMN é uma representação gráfica, contendo elementos como atividades, responsabilidades e fluxos. Ela tem, como objetivo, utilizar esses elementos para modelar (desenhar) processos de negócio (SANTOS, 2010).

Com a utilização do BPMN podemos definir as seguintes características:

- Objetivo do processo
- Entradas
- Saídas
- Fluxo de atividades
- Eventos que conduzem o processo
- Padronização para modelagem de processo.

A notação foi desenvolvida para ser compreendida por analistas de negócio, técnicos, usuários e todos os envolvidos no processo. Ela pode ser utilizada para modelar processos internos e abstratos (OMG, 2009).

Processos internos são os mais comumente utilizados. Eles definem apenas as atividades exercidas dentro da organização, podendo passar atividades entre diversos atores envolvidos, mas nunca para outras organizações. Um exemplo é apresentado na Figura 7, na qual temos um processo de Solicitação de compras que não exemplifica consultas externas à fronteira da organização.

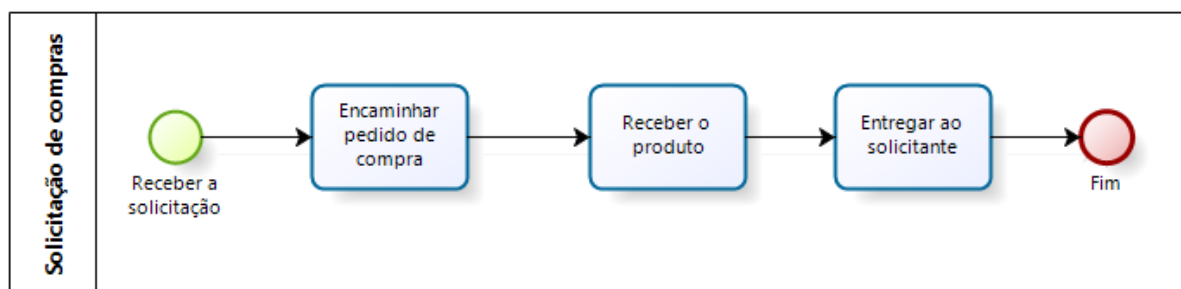
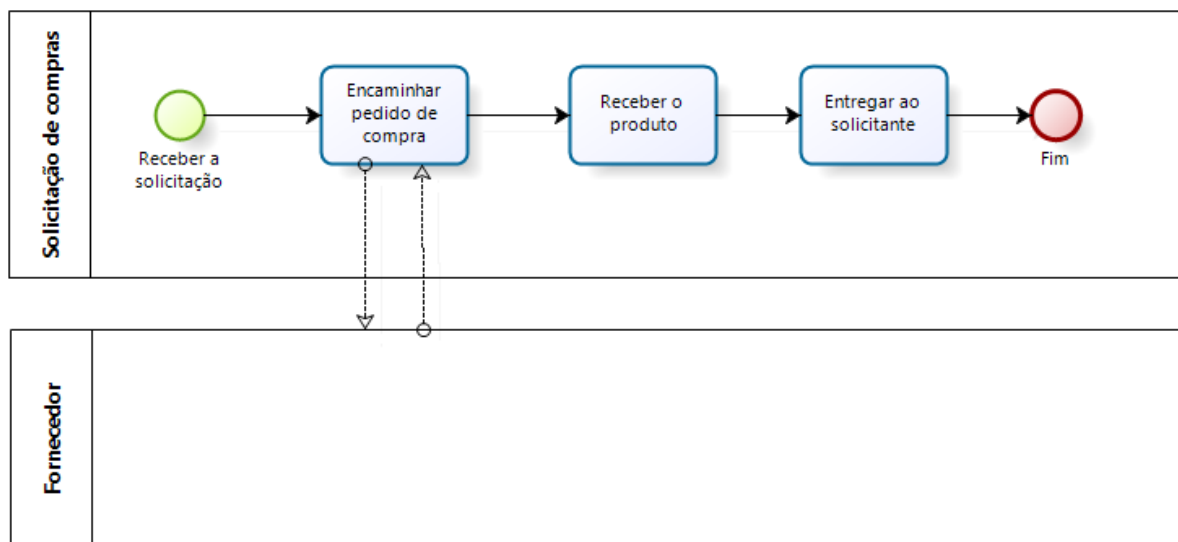


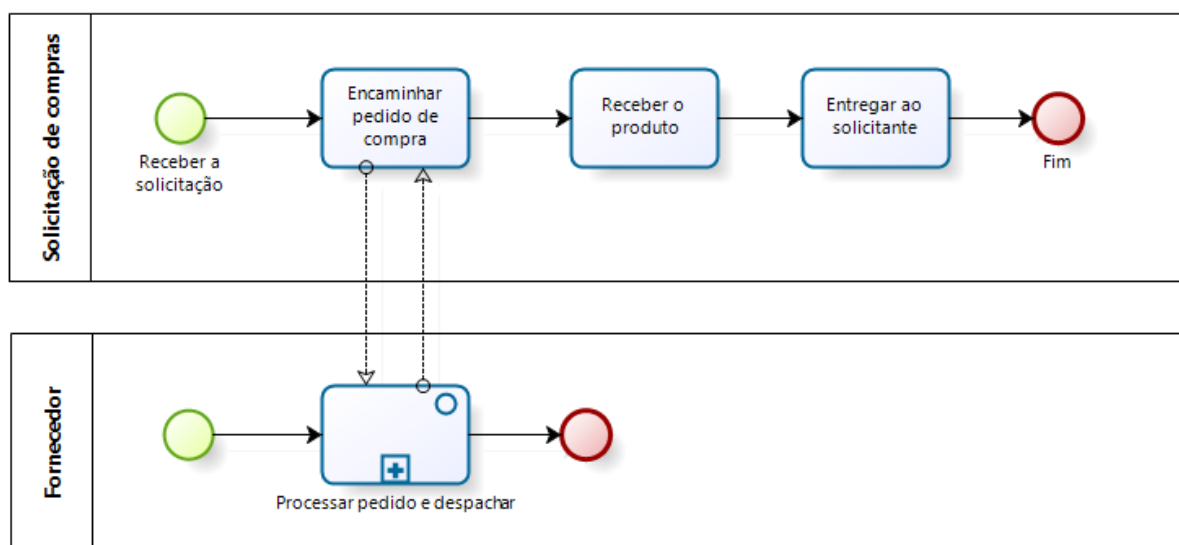
Figura 7 Exemplo de modelagem de um processo interno.

Processos abstratos são utilizados unicamente para demonstrar que uma atividade passará dos limites internos da organização para o exterior. Vale ressaltar que as atividades nesta organização externa não estão sob a gerência da organização solicitante. No exemplo da Figura 8, temos o processo de Solicitação de compras também modelando a interação com uma organização fornecedora.



**Figura 8 Exemplo de modelagem de um processo abstrato.**

Além de processos internos e abstratos, também existem processos de colaboração que são utilizados apenas no ponto de vista global, quando as interações são feitas entre duas ou mais entidades. Seguindo o mesmo processo de Solicitação de compras, a Figura 9 exibe uma atividade no Fornecedor.



**Figura 9 Exemplo de modelagem de um processo de colaboração.**

A notação é muito ampla e está designada a modelar todo e qualquer tipo de processo, sendo assim, possui muitos elementos. Os quais são divididos em dois grandes grupos (OMG, 2009):


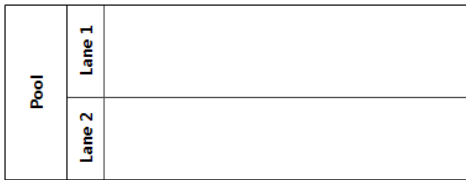
- *Core Elements*: conjunto de elementos simplificados capazes de modelar a maior parte dos processos organizacionais.
- *Full Elements*: todos elementos da notação, inclusive “*core elements*”, capazes de modelar qualquer negócio.

Nos próximos tópicos, serão representados os principais elementos utilizados no BPMN. Os mesmos estão organizados nas categorias *swimlanes*, Objetos de Fluxo, Objetos de Conexão e Artefatos.

### 2.5.1 Swimlanes

Na Tabela 2, são apresentados os dois elementos que representam os *swimlanes* no BPMN, conforme Santos (2009A) e OMG (2009).

**Tabela 2 Descrição dos elementos *Swimlanes* da notação BPMN.**

Objeto	Descrição	Figura	
<i>pool</i>	São utilizadas para exemplificar tanto organizações ou entidades externas como o nome do processo a ser mapeado.		
<i>lane</i>	As <i>lanes</i> sempre estão dentro de uma <i>pool</i> ; são utilizadas para organizar as atividades; geralmente servem para separar os papéis que classificam os participantes do processo, por exemplo os setores da organização.		

### 2.5.2 Objetos de Fluxo

São os principais elementos gráficos que definem o comportamento do processo de negócio. São sub categorizados em Eventos, Atividades e *Gateway*.

### 2.5.2.1 Evento

O objeto do tipo evento é algo que acontece durante o andamento de um processo de negócio, podendo ser: Início, Intermediário ou Fim. Um exemplo desse objeto é dado na Figura 10.



**Figura 10 Exemplo de Eventos, elementos de Objetos de Fluxo da BPMN.**


















Os três eventos listados na Figura 10 podem ser aprimorados com imagens dentro deles. Na Tabela a seguir, são apresentadas algumas destas imagens, conforme Santos (2009A) e OMG (2009).

Na Tabela 3, cada “Elemento” mapeado pode ter dois símbolos para “Capturar” e dois símbolos para “Lançamento”:

- O primeiro representa um evento do tipo Início;
- O segundo e terceiro representam elementos do tipo Intermediário;
- O último representa um elemento do tipo Fim.

A coluna denominada Lançamento (*throwing*) representa um disparo para outros eventos; a coluna denominada Capturar (*catching*) simboliza a espera de algum evento para que este, por sua vez, seja disparado. A coluna Lançamento difere-se da outra por ter seus símbolos em negrito (OMG, 2009).

Tabela 3 Descrição dos elementos de Objetos de Fluxo da notação BPMN

Elemento	Capturar		Lançamento	Descrição
<i>Message</i>				 Uma mensagem somente pode ser utilizada entre duas <i>pools</i> diferentes, nunca entre atividades da mesma <i>pool</i> .
<i>Timer</i>				Eventos nos quais se deve ter uma certa espera de tempo ou tempo limite para execução.
<i>Error</i>				Eventos para representar uma possibilidade de erro no fluxograma.
<i>Cancel</i>				Evento de cancelamento do fluxo.
<i>Compensation</i>				 Serve como um mecanismo para “desfazer” alguma atividade, é utilizado para representar uma circunstância que ocorre fora do fluxo normal.
<i>Conditional</i>				Retrata uma condição que deve ser cumprida para que o fluxo do processo continue.
<i>Link</i>				Serve para fazer a chamada entre atividades de forma não interligada por setas.
<i>Signal</i>				 É um evento no qual o <i>signal</i> de capturar espera o envio do <i>signal</i> de lançar para ser executado.
<i>Terminate</i>				Símbolo para finalização de um processo.
<i>Multiplex</i>				 Define que pode receber mais de uma atividade para fazer a continuação do fluxo, porém apenas uma é o suficiente.

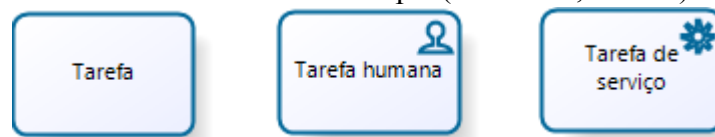
### 2.5.2.2 Atividade

O objeto Atividade é utilizado para definir um trabalho a ser executado. O mesmo pode estar representado de duas formas: como uma tarefa ou subprocesso. Um exemplo é dado na Figura 11.



**Figura 11 Exemplo de Atividades, elementos de Objetos de Fluxo da BPMN.**

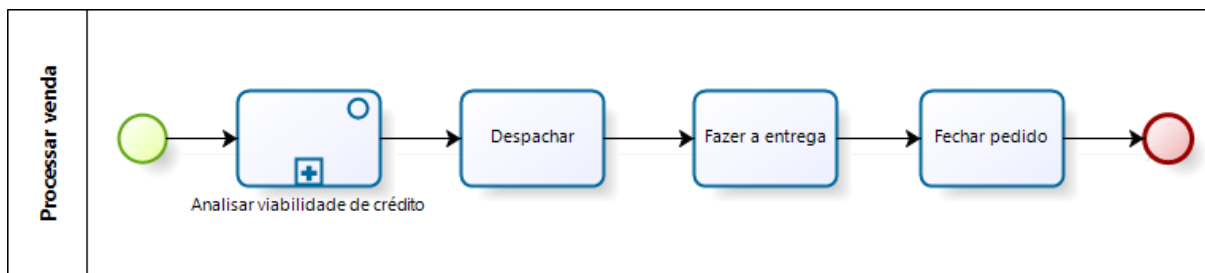
Atividade é o menor nível disponível na execução de um processo. Diversas ferramentas permitem a identificação das atividades com esteriótipos. Os principais são: automáticas/serviço (sistema) e humanas. Vale ressaltar que essa marcação não é obrigatória devido à *lane* já sinalizar qual tipo de atividade está contemplando. Não é possível, em uma mesma *lane*, ter tarefas humanas e automáticas. A seguir é apresentada a Figura 12 com uma exemplificação de atividades com e sem esteriótipos (SANTOS, 2009A).



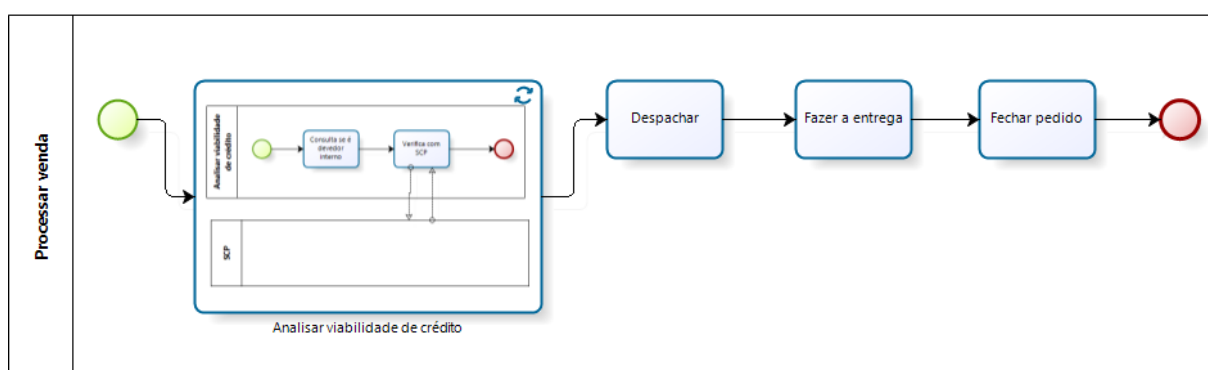
**Figura 12 Exemplo de Atividades com esteriótipos, elementos de Objetos de Fluxo da BPMN.**

Um subprocesso representa um conjunto de atividades com um propósito específico. Ele é útil para reunir atividades que podem ser repetidas em momentos distintos do processo, caracterizando reuso; pode ser definido como dependente (*Embedded Sub-process*), quando são dependentes do processo pai; ou como independente (*Reusable Sub-process*), quando podem ser reutilizados em processos diferentes (SANTOS, 2009A).

Os subprocessos podem ser representados de forma aberta, quando exibirão todas as atividades a serem efetuadas dentro do mesmo; ou de forma fechada. Seguem duas figuras, ambas fazem o mesmo processo. A Figura 13 exemplifica a representação fechada e a Figura 14 exemplifica a representação aberta.



**Figura 13** Exemplo de atividade com subprocesso fechado, elementos de Objetos de Fluxo da BPMN.



**Figura 14** Exemplo de atividade com subprocesso aberto, elementos de Objetos de Fluxo da BPMN.

### 2.5.2.3 Gateway


O objeto *Gateway* é definido em Michaelis (2011) como:

*n* 1 porta, abertura, passagem, portão. 2 caminho de entrada ou de saída.

Em outras palavras, é uma separação do fluxo em mais do que um caminho; é atrelada a uma condição para saber qual caminho correto do fluxo deverá seguir (OMG, 2009). Pode ser dividido em três tipos, demonstrados na Tabela 4:

**Tabela 4** Descrição dos objetos de *Gateway* da notação BPMN.

Objeto	Figura	Descrição
<i>Exclusive-or</i>		Este <i>Gateway</i> pode ter várias saídas definidas, porém apenas uma será utilizada. Ele também pode ser representado com um X na área interna.
<i>Inclusive-or</i>		Este <i>Gateway</i> poderá ter mais de uma opção selecionada.

Objeto	Figura	Descrição
<i>Parallel</i>		É utilizado principalmente para dividir ( <i>split</i> ) ou unir ( <i>merge</i> ) rotas paralelas em um processo.

Na Figura 15, segue exemplo da utilização de *gateway* do tipo *Exclusive-or*, em que apenas uma das opções pode ser seguida. No exemplo, o solicitante encaminha um pedido de aprovação de investimentos que pode seguir três fluxos:

- Caso o valor seja menor que R\$ 500,00, a aprovação não precisa passar pelos superiores;
- Caso o valor seja maior que R\$ 500,00 e menor que R\$ 2.000,00, a aprovação deve passar pelo diretor do centro de custo;
- Caso o valor seja maior que R\$ 2.000,00, a aprovação deve passar pelo Comitê Gestor.

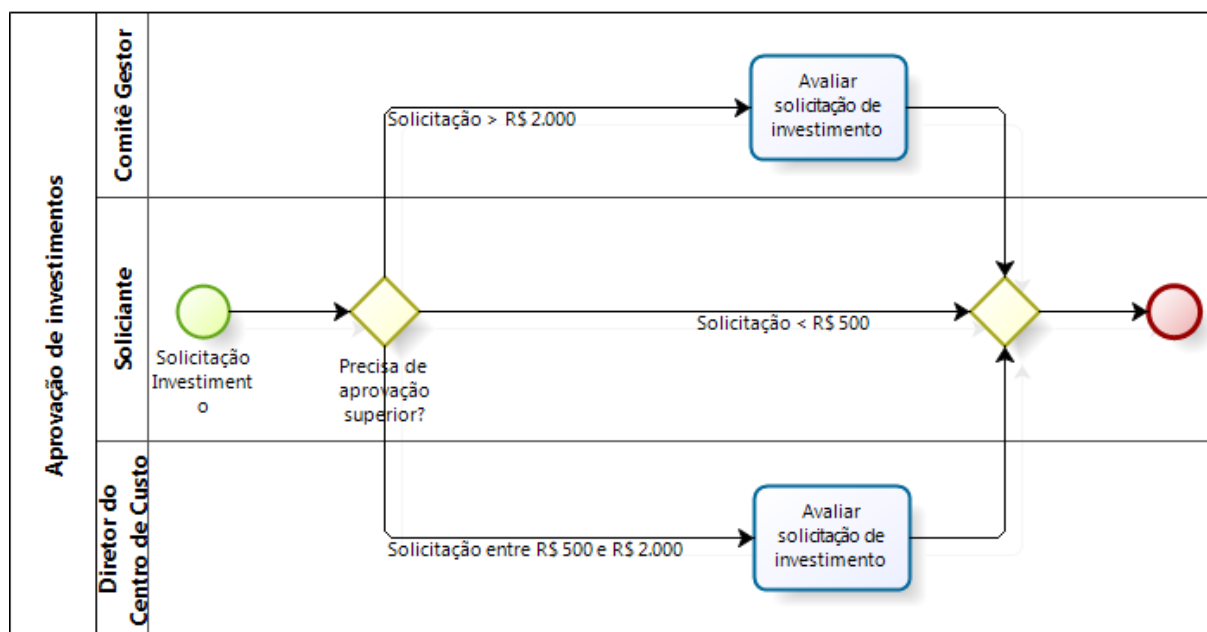
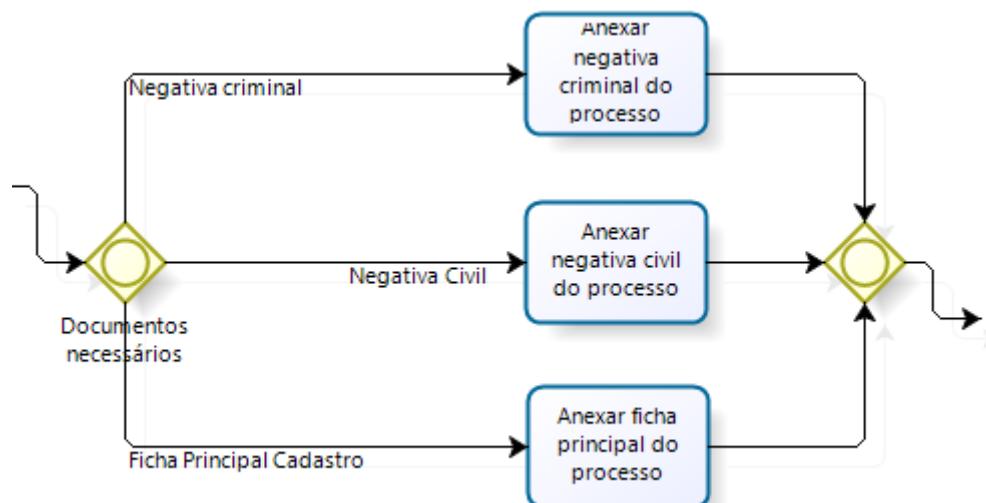


Figura 15 Exemplo de *Gateway Exclusive-or*, elementos de Objetos de Fluxo da BPMN.

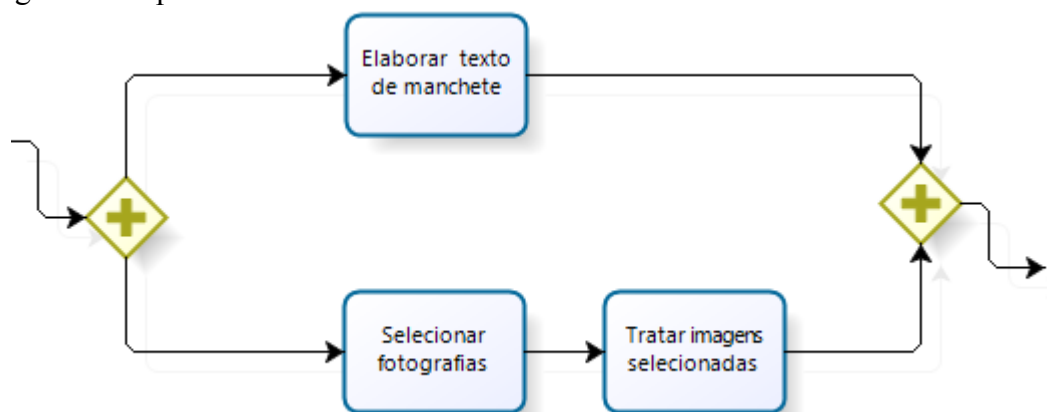
Na Figura 16, segue exemplo da utilização de *gateway* do tipo *Inclusive-or*, em que mais de uma opção pode ser seguida. No exemplo, o solicitante encaminha a requisição de um ou mais documentos que seguirão fluxos em paralelo; ao chegar ao final, o processo aguardará todos os fluxos iniciados para continuar o processo.





**Figura 16** Exemplo de *Gateway Inclusive-or*, elementos de Objetos de Fluxo da BPMN.

Na Figura 17, segue exemplo da utilização de *gateway* do tipo *Parallel*, em que as duas opções devem ser seguidas. O processo exemplifica a criação de uma notícia/manchete que passa por dois setores, ou pessoas diferentes; um dos setores é responsável pelo texto e o outro pela seleção e tratamento das imagens; chegando ao final, o fluxo junta novamente e prossegue com o processo.




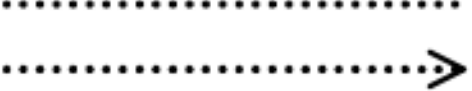


**Figura 17** Exemplo de *Gateway Parallel*, elementos de Objetos de Fluxo da BPMN.

### 2.5.3 Objetos de Conexão

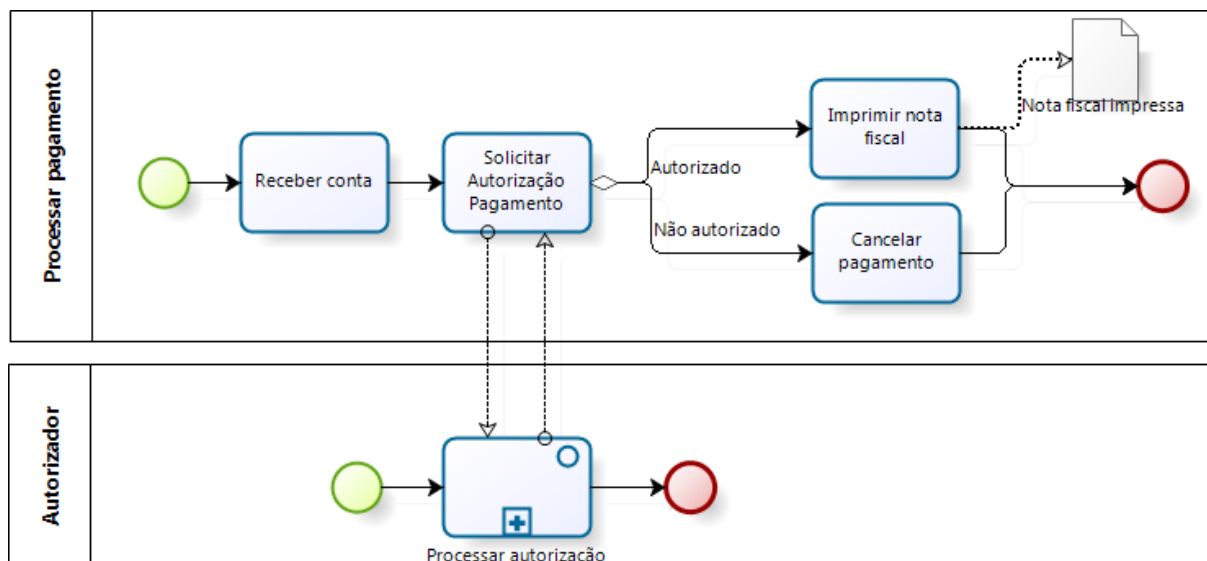
Os objetos de conexão têm o objetivo de interligar os objetos de fluxo, desenhando, assim, uma estrutura sequencial desde o início do processo até o final (SANTOS, 2009A). Segue Tabela 5 com tipos de objeto de conexão mais utilizados.

**Tabela 5 Descrição dos elementos de Objetos de Conexão da notação BPMN**

Objeto	Descrição	Figura
Fluxo de sequência	Apresenta a ordem de execuções das atividades de um processo	
Fluxo de mensagem	Representa o fluxo de mensagens entre participantes do processo e, da mesma forma, podem representar o fluxo de informações entre os processos.	
Fluxo condicional	Conecta duas atividades com a ideia de fazer um fluxo condicional, sem a necessidade de utilizar um <i>Gateway</i> .	
Associação	Utilizado para associar informações e objetos às atividades que não fazem parte do processo.	

Na Figura 18, segue exemplo de um processo de pagamento em que todos os objetos de conexão listados acima aparecem:

- Na passagem da atividade “Receber conta” para “Solicitar Autorização pagamento” é utilizada uma seta do tipo fluxo de sequência;
- Na passagem da atividade “Solicitar Autorização pagamento” para a atividade do Autorizador é utilizada uma seta do tipo fluxo de mensagem; utiliza-se essa seta, pois a atividade se encontra em outra *pool*.
- Na passagem da atividade “Solicitar Autorização pagamento” para as atividades de “Imprimir nota fiscal” e “Cancelar pagamento” é utilizada uma seta do tipo fluxo condicional; utiliza-se essa seta, pois se trata de um fluxo condicional que, caso seja autorizado, seguirá um rumo e, caso não seja, seguirá outro; esta mesma seta pode ser substituída por um *gateway* de tipo *exclusive-or*.
- Na atividade de “Imprimir nota fiscal” é apresentada a seta de associação, ligando a atividade a um artefato de objeto de dados.



**Figura 18 Exemplo da utilização de Objetos de Conexão da BPMN.**

#### 2.5.4 Artefatos

Artefatos são elementos que ajudam a prover mais informações sobre o processo, visando seu entendimento mais amplo, não interferindo no fluxo do processo (OMG, 2009).

Na Tabela 6, seguem os artefatos da BPMN.

**Tabela 6 Descrição dos artefatos da notação BPMN**

Objeto	Descrição	Figura
Anotações	São textos e informações genéricas sobre o processo ou um elemento do processo.	
Grupo	É uma maneira de agrupar atividades para melhor compreensão do processo, mas sem interferir no mesmo	
Objetos de dados	Pode indicar um documento ou dados de entrada ou saída de uma atividade.	

A Figura 19 exemplifica a utilização de cada um dos artefatos listados acima:

- A atividade “Avaliar solicitação de crédito” está informando que existe um objeto de dado com um formulário de solicitação de crédito;

- A atividade de “Preparar Documentação” está informando uma anotação através do artefato de anotação;
- As atividades “Avaliar histórico do cliente” e “Avaliar valor do crédito solicitado” estão agrupadas para representar mais visivelmente que ambas tratam a avaliação do crédito.

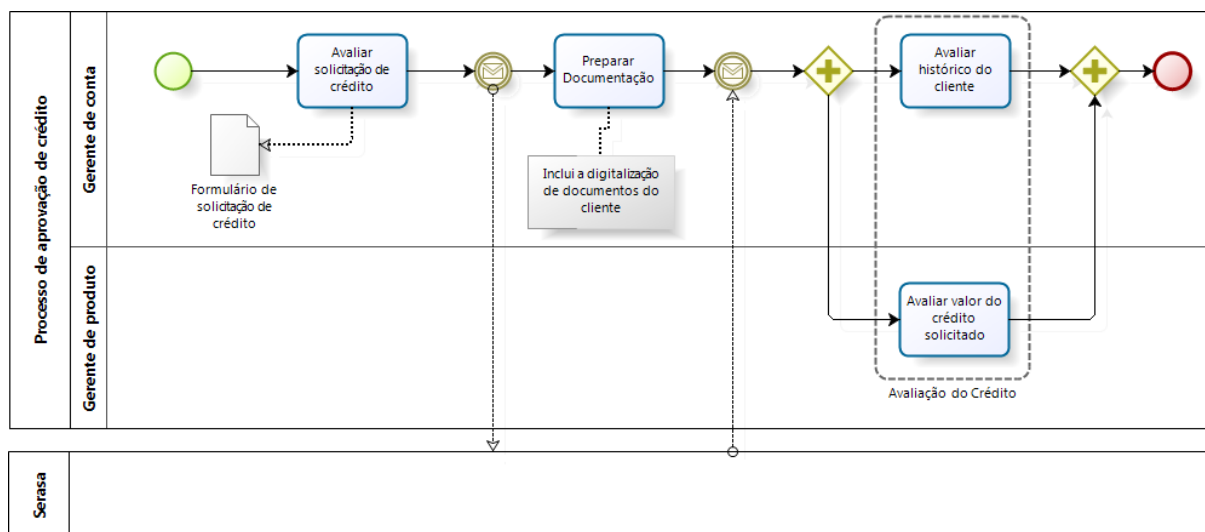


Figura 19 Exemplo da utilização de Artefatos da BPMN.

## 2.6 SOA e Serviços Web

Arquitetura Orientada a Serviços, mais conhecida como Service Oriented Architecture (SOA), é uma metodologia de programação criada para interligar diferentes softwares de uma forma simplificada e unificada; tem, como foco, a interoperabilidade, sendo adaptável em quase todas as situações. Ela já está em constante utilização no desenvolvimento de aplicações (BRIEN, MERSON, BASS, 2007).

Esses métodos usufruem da tecnologia de sistemas distribuídos, criando serviços que os acessem e tragam as informações necessárias. Esses métodos possuem as seguintes características, de acordo com Brien, Merson e Bass (2007):

- Uma interface publicada que abstrai a lógica subjacente;
- Sua localização é transparente;
- Pode ser implementada em diferentes linguagens ou plataformas e continuar sendo interoperável.

O SOA trabalha diretamente com *web services* e nada mais é do que a transformação da metodologia SOA em código fonte.

Nas subseções a seguir, serão exemplificados alguns atributos de qualidade e os seus prós e contras na utilização dessa metodologia.

Na Figura 20, é exemplificada uma estrutura SOA.

## Service Oriented Architecture

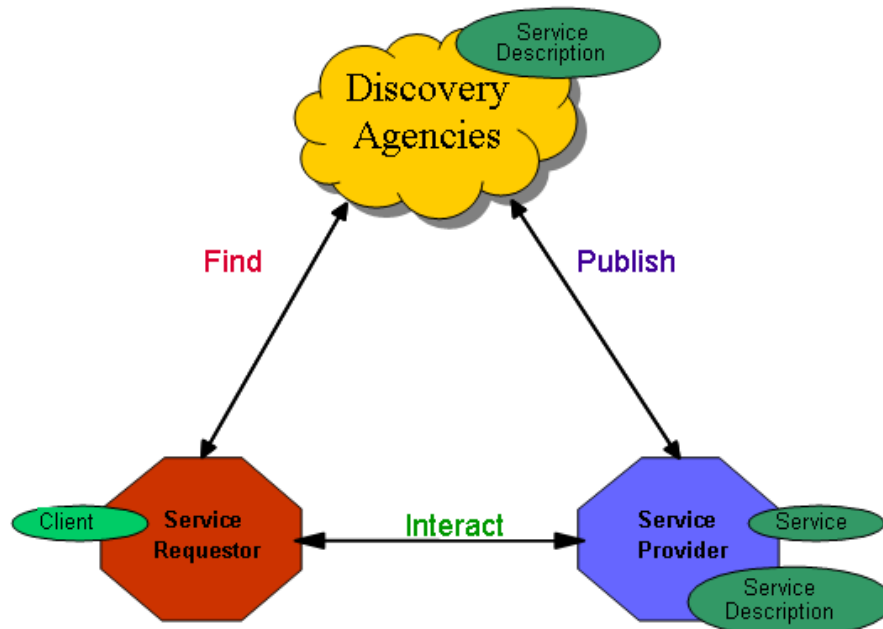


Figura 20 Exemplificação de utilização do método SOA (W3C, 2002).

### 2.6.1 Interoperabilidade

Na interoperabilidade os usuários do serviço e provedores do mesmo podem ser implementados em diferentes linguagens de maneira transparente através do mecanismo de envio e retorno. Isso é possível, pois os serviços *web* definem o formato da interface e os protocolos de comunicação, mas não restringem a linguagem implementada ou a plataforma.

Para garantir a interoperabilidade dos serviços *web* através das plataformas, os serviços *web* utilizam basicamente dois padrões: Web Service Definition Language (WSDL) e Simple Object Access Protocol (SOAP), que são padrões abertos e de ampla utilização.

### 2.6.2 Desempenho

O desempenho pode ser lido de diversas formas, mas em se tratando de serviços *web*, leva-se em conta o tempo de resposta entre a solicitação de um pedido até o recebimento do mesmo. Na maioria dos casos, o desempenho é negativamente afetado, utilizando SOA.

Para sistemas que devem ter uma resposta em tempo real, SOA não será a melhor opção a ser adotada.

O protocolo de interação, às vezes, tem de fazer uma consulta a uma base na qual estão listados os servidores para fazer a conexão, isso é um fator que amplia latência na requisição.

O uso do padrão Extensible Markup Language (XML) aumenta em muito a latência no servidor, pois o mesmo deve passar por três etapas a mais para cada requisição: análise, validação e transformação.

No lado positivo, SOA oferece transparência no quesito de localização dos serviços. Os usuários não necessariamente precisam saber a localização dos registros, isso faz com que nem percebam qualquer modificação na localização dos mesmos. Toda essa transparência pode ser utilizada também para balanceamento de carga, alternando entre servidores, tudo isso sem que o usuário requisitante perceba.

### **2.6.3 Segurança**

Embora o termo segurança seja muito amplo em se tratando de software, em geral, existem quatro princípios associados:

- Confidencialidade, garantia de que apenas o destinatário autorizado deve ler a mensagem;
- Autenticidade, está relacionada com a confiança de que a mensagem enviada é realmente do autor indicado;
- Integridade, garantia de que a mensagem esteja incorrupta;
- Disponibilidade, garantia de que o serviço estará disponível em tempo hábil.

A seguir há algumas características que são levadas em consideração para que a segurança atingida seja a esperada:

- As mensagens enviadas devem ser criptografadas, mesmo sabendo que isso aumentará o tamanho do pacote a ser enviado. Dessa forma, garantimos que informações como dados de um cartão de crédito não serão interceptadas.
- As informações transferidas devem ser criptografadas não apenas durante a troca de mensagens, mas também no local onde serão armazenadas, assim, garantindo que a informação não seja descoberta mesmo após já estar armazenada no local de destino.

Isso tudo, pois algumas informações poderão estar armazenadas em organizações de terceiros.

- Um sistema de autenticação deve ser implementado para que apenas pessoas autorizadas acessem os serviços.
- Uma solução SOA pode ser invocada, procurando os serviços em um diretório público, dessa forma, será necessário garantir que apenas pessoas autorizadas possam publicar informações.

Os servidores que irão armazenar os serviços web devem estar configurados para utilizar Secure Socket Layer (SSL) e certificados digitais para encriptar e garantir que a transmissão seja feita entre destinatário e remetente.

Mecanismos de segurança têm impacto negativo na performance, modificabilidade e interoperabilidade.

#### **2.6.4 Garantia**

A garantia de entrega dos pacotes SOA está diretamente ligada ao serviço de Internet contratado. Contudo, outro fator importante é a transação, sabe-se que ela é iniciada em um serviço e também é terminada no mesmo, para tanto, em casos de chamar diversos serviços que possuam transações, deve-se ter cuidado com a ordem de chamada e, se possível, possuir algum mecanismo que faça o processo inverso, caso alguma das chamadas tenha falhado.

#### **2.6.5 Disponibilidade**

Disponibilidade é o tempo em que o serviço ficará disponível para acesso. Pela perspectiva do usuário, quando um serviço não estiver disponível, não conseguirá finalizar corretamente todos os seus requerimentos. Na perspectiva do servidor do serviço, o mesmo somente deverá estar disponível, quando for utilizado pelo usuário. Em casos de não funcionamento correto dos serviços, o servidor acaba tendo sua reputação afetada.

Quando serviços externos são contratados, geralmente os mesmos vêm com um acordo de nível de segurança (Service Level Agreement (SLA)), que define em detalhes o tempo durante o qual o serviço estará disponível, o tempo máximo de espera até que o serviço seja reabilitado em caso de queda e, assim por diante os outros demais fatores.

### 2.6.6 Modificabilidade

Modificabilidade é a habilidade de se fazer alterações rápidas em um serviço. Serviços são autosuficientes, modulares e acessados por meio de interfaces coesas. Dessa maneira, os custos de manutenção são reduzidos e a modificabilidade aumenta. Por outro lado, caso a interface também tenha de ser modificada, fica difícil localizar quem está usando o serviço e quais os impactos que terá sobre estas interfaces.

Extensibilidade é um caso especial para a modificabilidade, são as formas como um serviço pode ser evoluído sem que afete outras partes do sistema. Isso é muito importante, pois as organizações estão sempre modificando suas regras de negócio. Para a extensão dos serviços SOA, existem três representações:

- Adição de novos serviços: não gera impacto algum em outros serviços que já estão em funcionamento.
- Estender um serviço sem modificação em interface: a utilização deste não interfere na visualização final do usuário, pois suas modificações ficam vedadas a regras e não a melhorias em interface.
- Estender um serviço com modificação em interface: quando utilizada esta forma, existirá um passo a mais na modificação, em que se deve atualizar também os arquivos de interface para que tudo funcione alinhadamente.

### 2.6.7 Testabilidade

Testabilidade é o grau em que um sistema ou serviço facilita o estabelecimento de critérios de teste e a realização de testes para determinar se esses critérios foram cumpridos (BRIEN, MERSON, BASS, 2007).

Fazer testes em sistemas que utilizam SOA é mais complexo por razões que incluem:

- Dificuldade de traçar a execução do teste, quando os elementos residem em diferentes partes na internet.
- O código fonte dos serviços externos podem não estar disponíveis durante a execução dos testes. Também é possível não se ter acesso aos *logs* gerados por estes serviços externos, o que dificulta a coleta dos resultados.
- Em alguns casos os testes devem ser feitos em todas plataformas disponíveis para garantir o total funcionamento do sistema, isso pode ser um desafio, vendo a quantidade de sistemas operacionais e de sistemas intermediários que existem e que são utilizados hoje em dia.



### 2.6.8 Usabilidade

A usabilidade é um fator que pode ser comprometido com a utilização de serviços web, uma vez que a demora de resposta pode variar por diversos motivos da rede, causando uma má impressão para o usuário que o utilizará. O protocolo SOAP não exibe uma notificação do progresso da chamada, além de não conseguir cancelar uma chamada ativa.

Um exemplo prático de demora no tempo de resposta de uma requisição pode ser demonstrado em um sistema de reserva de passagens aéreas, se o cliente solicita saber os voos que saem do aeroporto de Porto Alegre, terá de encaminhar consultas para todas as companhias aéreas que operam em Porto Alegre para saber quais os voos disponíveis. Uma forma de agilizar esse processo é ter uma base de dados pré carregada, o que fará com que sejam desnecessárias as consultas a todas as organizações aéreas. Que, por fim, fará com que o tempo de resposta seja menor, causando menos incômodo ao usuário.

### 2.6.9 Escalabilidade

Escalabilidade é a possibilidade de uma plataforma estar preparada para crescer seu número de requisições sem redução de performance.

A tecnologia de serviços web não oferece nenhuma forma de aumentar a escalabilidade, porém existem algumas estratégias que podem ser utilizadas para este fim. Seguem três exemplos:

- Escala horizontal: adicionar servidores com balanceamento de carga.
- Escala vertical: aumentar a capacidade do servidor.
- A performance do servidor tende a baixar a cada requisição, sendo que a cada mensagem enviada, o protocolo SOAP deve fazer todo o processo em cima do XML para tornar as informações usáveis no sistema. Deve ser feito um teste cuidadoso sobre esse processo, aumentando gradativamente a quantidade de requisições, pois a magnitude da escalabilidade deve ser testada.

## 2.7 XML Process Definition Language

A XML Process Definition Language (XPDL) foi criada com o objetivo de estabelecer um modelo de intercâmbio de processos de negócio entre diversas ferramentas que trabalham

com mapeamento de processos de negócio, sejam elas de modelagem, definição de processos ou controle de *workflow* de processos (WfMC, 2008). Ela foi proposta, inicialmente, pela Workflow Management Coalition (WfMC), entidade que define padrões para a comunidade que trabalha com *workflow*.

De acordo com WfMC, BPMN é o padrão ideal para a definição de processos e, por esse motivo, a linguagem XPDL é totalmente compatível com ele (WfMC, 2008). O XPDL é uma escrita em forma textual técnica dos processos visualizados pela notação BPMN, transcrevendo atividades, *gateways*, *pools* e demais elementos da notação em blocos de texto formatados em XML.

Por ser um padrão muito difundido e utilizado em diversos softwares de modelagem de processos de negócio, foi escolhido para estar presente na arquitetura dessa proposta.

Na Figura 21, é demonstrado um código XPDL. Cada elemento possui um atributo “Id” para a descoberta dos fluxos das atividades, das ligações existentes no processo.

```

▼<Pools>
  ▶<Pool Id="7cca3f99-e6dd-43b7-bf6a-48d988a0f614" Name="Main Process" Process="ad450c
  ▼<Pool Id="668a0ac8-dce8-4bd0-9449-e0c01fb27f59" Name="Piscina Processar pagamento"
    ▶<Lanes>...</Lanes>
    ▶<NodeGraphicsInfos>...</NodeGraphicsInfos>
  </Pool>
  ▶<Pool Id="89e66472-4107-443a-b9b8-73660027bcd" Name="Piscina autorizacao" Process=
</Pools>
▼<MessageFlows>
  ▶<MessageFlow Id="1e71c943-f2f5-442a-b646-a1dc8bb125e2" Source="09e63430-abc8-4f71-9
  ▶<MessageFlow Id="3b308de7-6208-4dc7-b859-84cb9ff7efe0" Source="be7737c6-fe25-4b0f-9
</MessageFlows>
▼<Associations>
  ▶<Association Id="43d2481d-41ee-4fb6-b3e8-39d438f874da" Source="562f3a57-2d6a-4e1e-9
  impressa">...</Association>
  ▶<Association Id="a3ccf627-4ba9-4abc-8ce0-b475eac878be" Source="432d28fb-759c-49a8-9
</Associations>
▶<Artifacts>...</Artifacts>
▼<WorkflowProcesses>
  ▶<WorkflowProcess Id="ad450d35-5255-47b4-92fd-20a4732ac3f3" Name="Main Process">...<
  ▶<WorkflowProcess Id="0b6ab10c-6eb2-4835-a7c6-3367d1a3d09a" Name="Piscina Processar
  ▶<WorkflowProcess Id="5125b8b4-c4bf-4466-8b6d-134271a18d1a" Name="Piscina autorizac
</WorkflowProcesses>

```

Figura 21 Exemplo de um código XPDL.

## 2.8 UML – Unified Modeling Language

UML é uma linguagem de modelagem unificada cujo objetivo é auxiliar no desenvolvimento de projetos de software. Através de sua notação consegue representar, mais claramente, processos que serão implementados nos softwares (JONES, 2002).

UML não é uma metodologia para o desenvolvimento de softwares, ela não definirá os passos que devem ser seguidos para o mesmo, pois serve somente como auxiliadora no entendimento de processos para o desenvolvimento de softwares (JONES, 2002).

Em 1997, a UML foi aceita como padrão pela Object Management Group (OMG). OMG é um consórcio internacional de empresas que define padrões na área de Orientação a Objetos (TONSIG, 2011).

Dentre os diversos diagramas que a UML define, será descrito um pouco mais sobre o Diagrama de Casos de Uso, Diagrama de Classes e Diagrama de Componentes.

### 2.8.1 Diagrama de Casos de Uso

Um caso de uso é a representação de uma ação dentro de um software. Essa ação é executada por um ator. Portanto, um diagrama de casos de uso é a representação de um cenário em que um ou mais atores podem executar casos de uso (SANTOS, 2011).

Na Figura 22, são demonstrados os elementos utilizados neste diagrama. São eles: o ator e o caso de uso.



Figura 22 Elementos de um Diagrama de Caso de Uso

Na Figura 23, está representado um diagrama de caso de uso em que é exemplificado um Vendedor podendo acessar os casos de uso de Consultar produtos e Consultar estoque, enquanto o Cliente possui apenas acesso ao caso de uso de Consultar produtos.

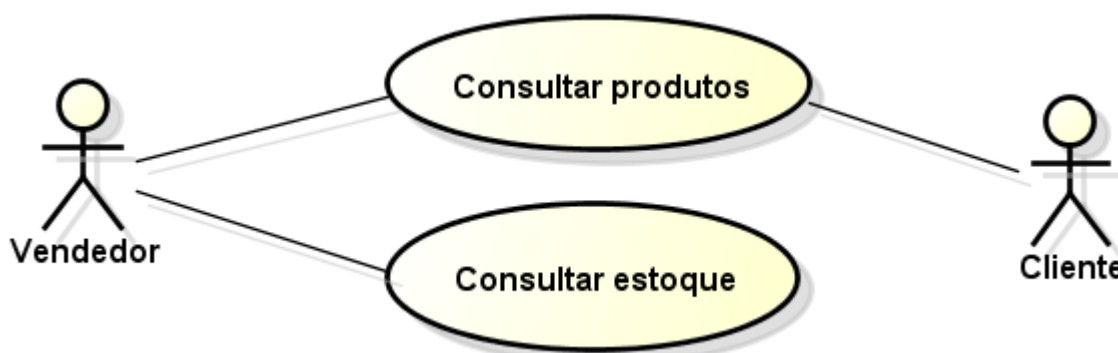


Figura 23 Exemplo de um Diagrama de Caso de Uso

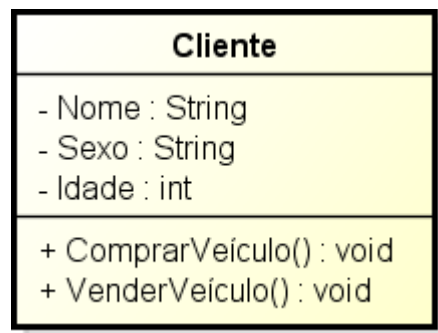
### 2.8.2 Diagrama de Classes

Uma classe é a descrição de um objeto. Um diagrama de classes é composto por diversas classes, cujo objetivo é demonstrar, em forma diagramada, todos os objetos que interagem em determinado processo ou ferramenta (TONSIG, 2011).

Para uma melhor compreensão sobre as classes, será descrita sua estrutura e a mesma poderá ser acompanhada na Figura 24. As classes são representadas com três retângulos, um abaixo do outro:

- O primeiro retângulo é designado para armazenar o nome da classe, por exemplo, “Cliente”.
- O segundo retângulo é utilizado para descrever os atributos da classe, tais como “Nome”, “Sexo” e “Idade”.
- O terceiro retângulo é utilizado para definir as operações que a classe pode fazer, tais como “ComprarVeículo()” e “VenderVeículo()”.

Não é necessário o preenchimento de todos os retângulos, pode-se criar um diagrama mais simples contendo apenas o nome das classes.



**Figura 24 Exemplo de Classe**

A Figura 25 contém um exemplo para a compreensão das ligações entre classes. As classes representadas são Cliente, Veículo, Carro e Moto. Segue a descrição das ligações:

- A ligação existente entre o Veículo, Carro e Moto é chamada de herança. Como a flecha está apontando para o veículo, significa que a classe Veículo poderá ser um Carro ou uma Moto.
- A ligação existente entre a classe Cliente e Veículo é uma ligação denominada associação que, neste caso, aponta que um Cliente pode ter zero ou vários (0 .. \*) Veículos, e, olhando pelo outro lado, podemos ver que um Veículo pode pertencer a apenas um (1) Cliente.

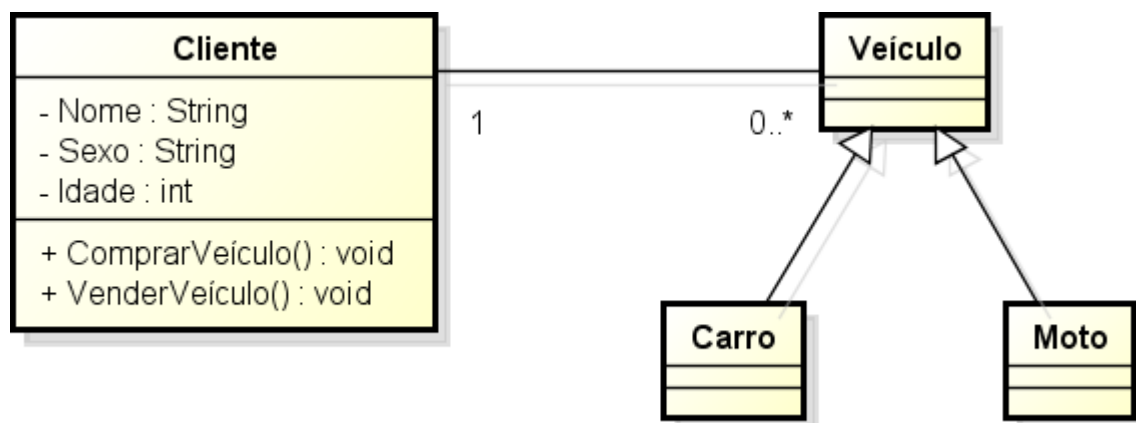


Figura 25 Exemplo de um Diagrama de Classes

### 2.8.3 Diagrama de Componentes

No contexto de desenvolvimento de softwares um componente pode ser considerado desde um código fonte, uma classe externa, uma biblioteca de funções, entre outros (SANTOS, 2011).

Na Figura 26, é demonstrado o elemento que representa um componente dentro de um diagrama de componentes.

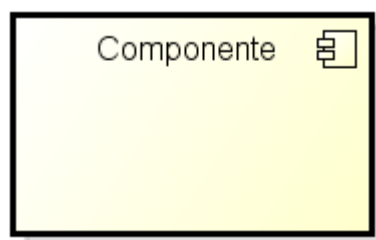
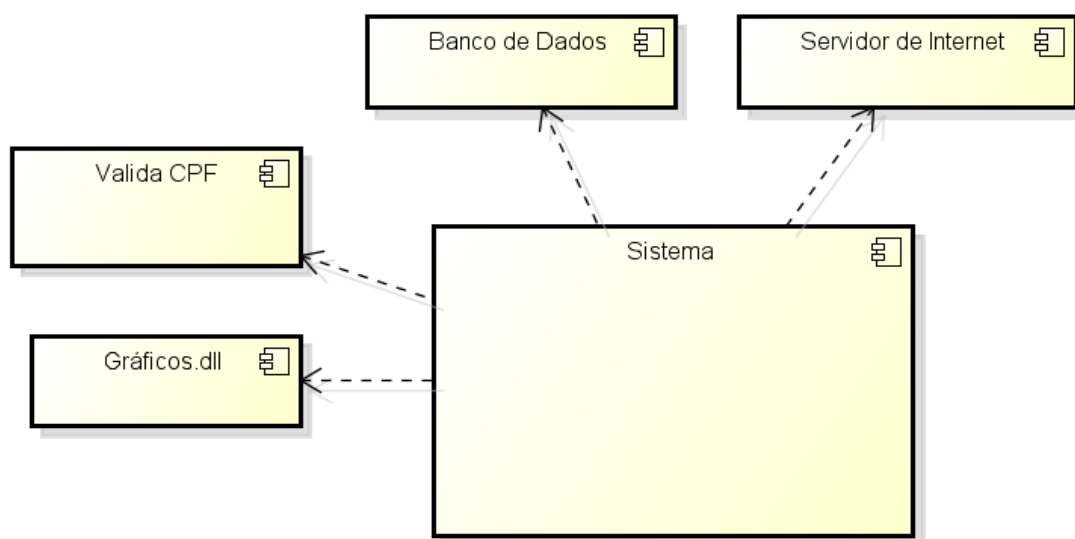


Figura 26 Elemento que representa um Componente em um Diagrama de Componentes

Na Figura 27, é exemplificado um diagrama de componentes em que o componente Sistema necessita de todos os demais componentes, sendo eles:

- Banco de Dados e Servidor de Internet estão representando sistemas externos.
- Valida CPF e Gráficos.dll estão representando componentes necessários para o funcionamento do sistema.



**Figura 27 Exemplo de um Diagrama de Componentes**

### 3 TRABALHOS RELACIONADOS

#### 3.1 Modelos

Existem diversos modelos que implementam estruturas BPM. A seguir exemplificaremos dois que foram utilizados como base para a criação do modelo da ferramenta que está sendo desenvolvida.

##### 3.1.1 Artigo: A Basing on Model-Driven Framework of Service-Oriented Software Production Line

O trabalho proposto por CHAO (2009), tem como objetivo criar um *framework* para modelar processos de linha de produção, utilizando características do BPMN no motor central do *framework*. Neste é apresentado um modelo simplificado para interações de atividades, utilizando elementos do BPMN. O modelo é representado na Figura 28, cujos elementos estão dispostos em classes como *BPMNElement* que herda características da *Graphical Element* que, por sua vez, herda informações das classes *Flow Object* ou *Connecting Object*.

A classe *Flow Object* pode ser dos tipos *Event*, *Gateway* ou *Activity*.

A classe *Connecting Object* possui os atributos *sourceRef[1]: flowObject* e *sourceRef[2]: flowObject* que são objetos do tipo *flowObject* e demonstram o fluxo de uma atividade para outra. A mesma classe ainda herda características das classes *Sequence Flow* ou *Message Flow* para definir qual tipo de conexão existe entre as atividades.

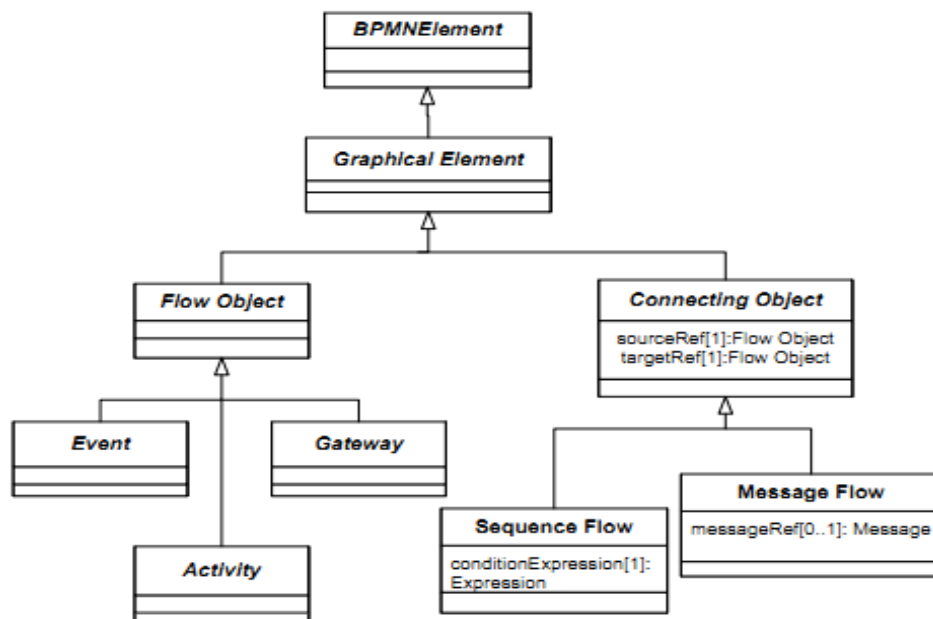


Figura 28 Meta modelo de BPMN de acordo com CHAO (2009).

### 3.1.2 Artigo: Modeling and Validating BPMN Diagrams

O trabalho proposto por CHINOSI e TROMBETTA (2009) tem, como objetivo, criar um novo modelo de classes simplificado para suportar a notação BPMN. Os autores atribuem a criação deste novo modelo devido à grande complexidade e à falta de clareza com que os modelos atuais estão sendo feitos. O modelo proposto é estruturado de forma simples o suficiente para suportar a notação. Sendo ele um meta modelo, não contém a descrição dos atributos e operações das classes, porém suficientemente entendível para que seja estendido.

Na Figura 29, é demonstrado o modelo criado no artigo. Pode-se observar que tal modelo possui estas classes principais: BPD, *Pool*, *Lane*. A ligação entre as classes ocorre desta forma:

- BPD pode ter um ou mais *Pool*;
- *Pool* pode ter apenas um BPD;
- *Pool* pode ter um ou mais *Lane*;
- *Lane* pode ter apenas uma *Pool*.

Desta forma, diz-se que um diagrama terá uma ou várias *Pools*, e cada *Pool* poderá ter uma ou várias *Lanes*. Dentro das *Lanes*, serão adicionados os elementos como *Activities* e *Events*.

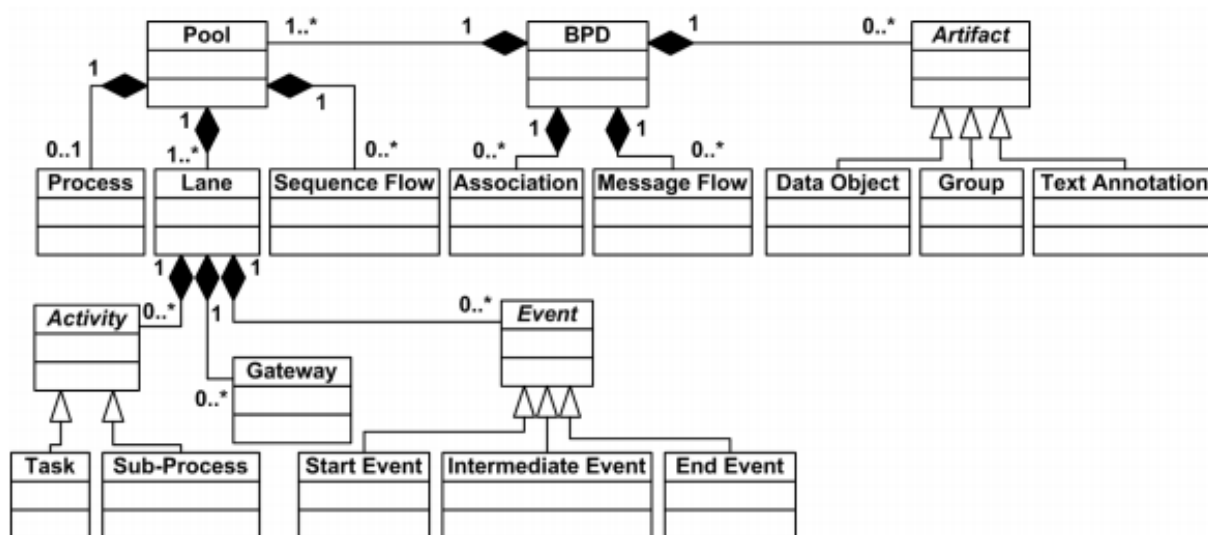


Figura 29 Meta modelo de BPMN de acordo com CHINOSI e TROMBETTA (2009).

### 3.2 Ferramentas

No mercado, existem várias ferramentas que desempenham tarefas de um BPMS, dentre estas, duas se destacam, sendo elas a BizAgi e Bonita Open Source, que serão brevemente apresentadas na sequência.



### 3.2.1 BizAgi

BizAgi é uma organização que está há mais de 20 anos no mercado e desenvolve uma ferramenta BPMS denominada BizAgi. A ferramenta é dividida em dois módulos. Um deles é mais completo, possuindo todas as funcionalidades exigidas para o completo funcionamento de um BPMS, não é uma ferramenta *freeware*. O outro módulo serve para a modelagem de processos de acordo com a notação BPMN, este é *open source*.

O módulo para modelagem de processos é um dos mais utilizados hoje em dia, possui suporte para exportação XPD, que é um formato padrão definido pela WfMC. A interface é bem completa, intuitiva e prática para a modelagem de processos.

Na Figura 30, é exemplificada a ferramenta, criando um processo dentro do módulo BPMN. Observando a Figura 30, à esquerda, existem os elementos que podem ser adicionados ao processo. O processo é desenhado no centro e para adicionar um elemento ao processo, basta arrastá-lo com o mouse.

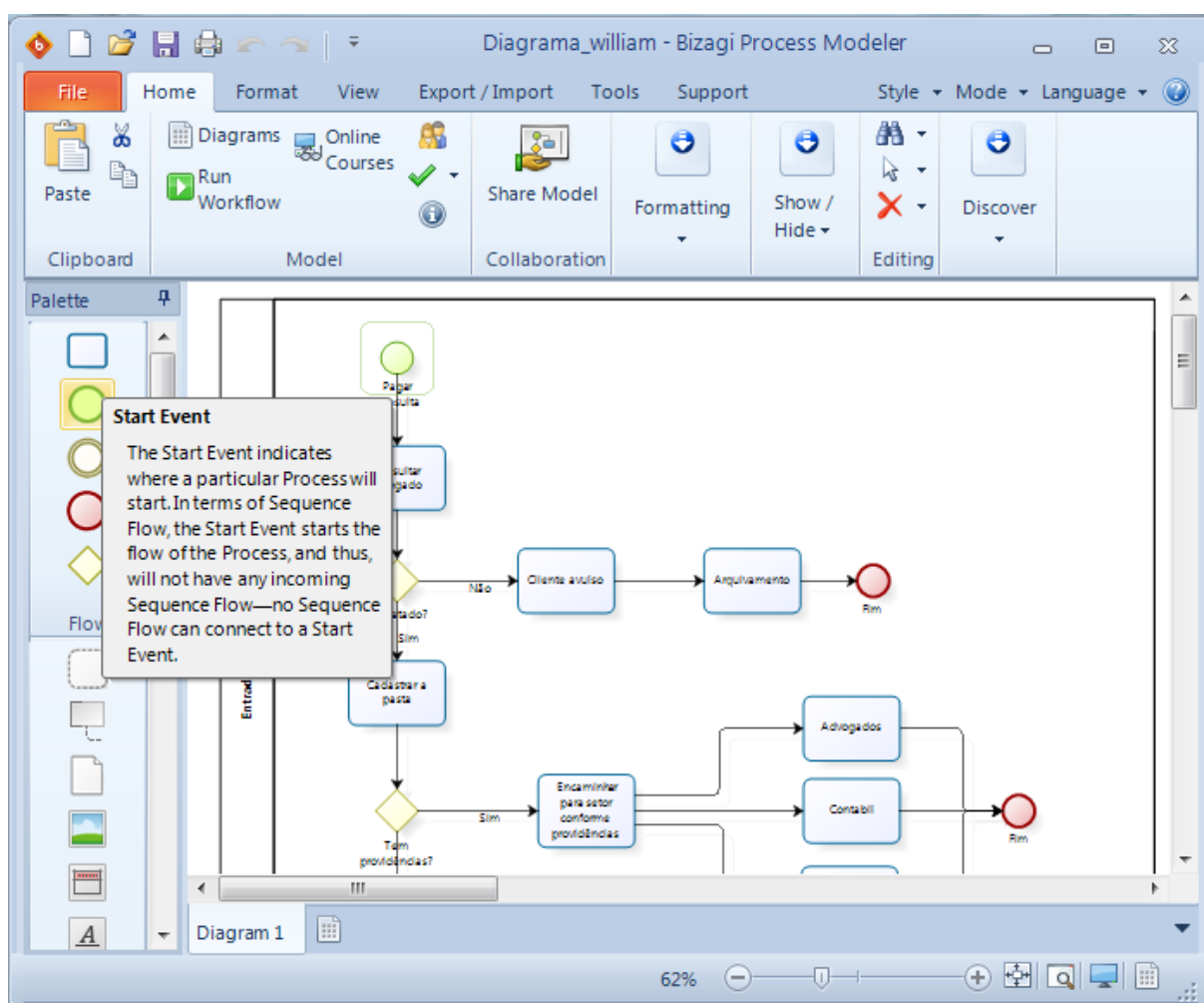


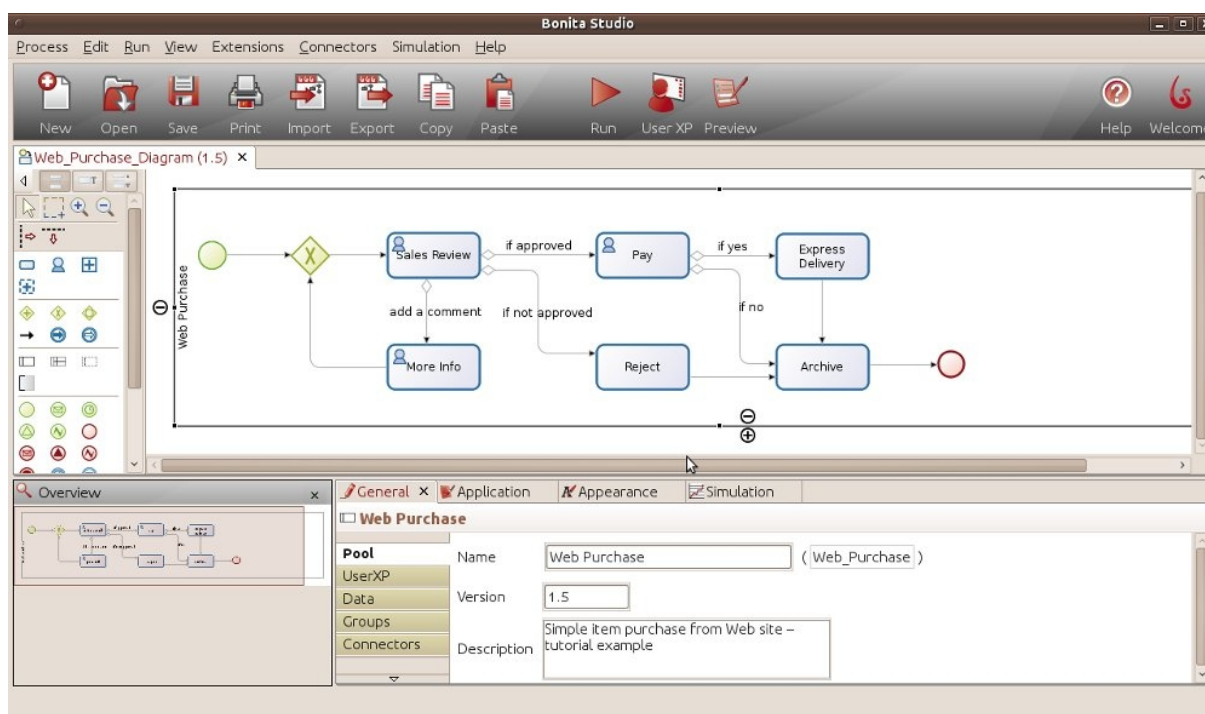
Figura 30 Software BizAgi

### 3.2.2 Bonita Open Solution

Bonita Open Solution é uma ferramenta desenvolvida a partir de 2001 e mantida pela organização BonitaSoft. Ela é uma ferramenta parecida com a BizAgí, porém tendo um diferencial, é totalmente open source, desde a modelagem do processo até os módulos que compreendem o fluxo de processos.

Bonita Open Solution também implementa funcionalidades de importação e exportação XPDŁ.

A Figura 31 demonstra a Bonita Studio em sua etapa de modelagem de processos. Percebe-se que sua interface é similar a do BizAgí, contendo, à esquerda, todos os elementos da notação BPMN e, ao centro, o desenho do processo.



**Figura 31 Software Bonita Open Solution**

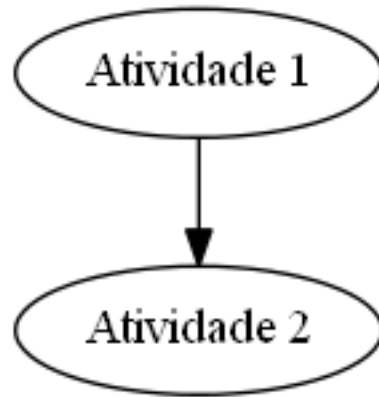
### 3.2.3 Graphviz

Graphviz<sup>1</sup> é um ferramenta para criar fluxos de diagramas. Seu diferencial está na criação desses fluxos, pois os mesmos não precisam ser desenhados, apenas referenciados textualmente, o próprio Graphviz se encarrega de criar o desenho.

Na Figura 32, é demonstrado o funcionamento do Graphviz. Para tanto foi criada uma estrutura simples com o seguinte código:

<sup>1</sup>Biblioteca disponível em <http://www.graphviz.org/>

digraph G {"Atividade 1"->"Atividade 2"}



**Figura 32** Exemplo de saída do Graphviz

## 4 TRABALHO PROPOSTO

### 4.1 Visão Geral

O propósito básico do trabalho é fornecer uma arquitetura em software para gerir modelos de processos de negócio. Para tal, essa arquitetura oferecerá interfaces para realizar a carga de modelos representados no formato XPDL. As interfaces seguirão a arquitetura SOA. A partir dessa carga, que poderá ser realizada a partir de ferramentas já estabelecidas como Bizagi ou Bonita, os processos serão armazenados no modelo interno da aplicação, que pode ser visto no diagrama de classes.

Além de permitir a carga e o armazenamento de modelos, a ferramenta facilitará a disseminação dos processos por meio de uma ferramenta *web*, bem como o refinamento do processo, por meio de anotações e comentários. Dessa forma, o processo poderá ser evoluído de maneira colaborativa por meio de um ambiente no qual todos têm acesso à informação e podem consultar as relações entre os processos.

O nome da ferramenta será Promap, em que “Pro” significa “Processo” e “Map” significa “Mapeamento”, ou seja uma ferramenta de mapeamento de processos.

O trabalho tem a premissa de que a ferramenta seja capaz de controlar os diversos processos desenvolvidos em organizações. O mesmo manterá os processos organizados para simplificar a localização destes, facilitando as tarefas dos colaboradores que poderão fazer uma consulta quando não recordarem corretamente das atividades a serem feitas. Além das definições das tarefas individuais estarem mais esclarecidas, os colaboradores começarão a ter uma ideia do todo, ou seja, saberão onde inicia o trabalho e onde ele termina.

Na Figura 33, é exibida a visão geral do projeto. Na mesma figura existe o ator Analista de Processos que trabalha diretamente com o software modelador e com o próprio ProMap. No software modelador, ele cria o processo e o exporta em formato XPDL, já, no ProMap, ele importa esse processo.

Também temos presente, na Figura 33, o ator Colaborador (podendo ser mais de um) que trabalha diretamente com o ProMap, podendo visualizar os processos dos quais faz parte, documentá-los textualmente, vinculá-los a outros processos, ou extrair um documento completo sobre o processo.

Dentro da estrutura do ProMap, podemos ver uma Interface Web conectando-se diretamente aos controladores do sistema (os casos de uso). Esses controladores, por sua vez,

trabalham diretamente com o modelo de classes, e este faz as consultas e as atualizações no banco de dados.

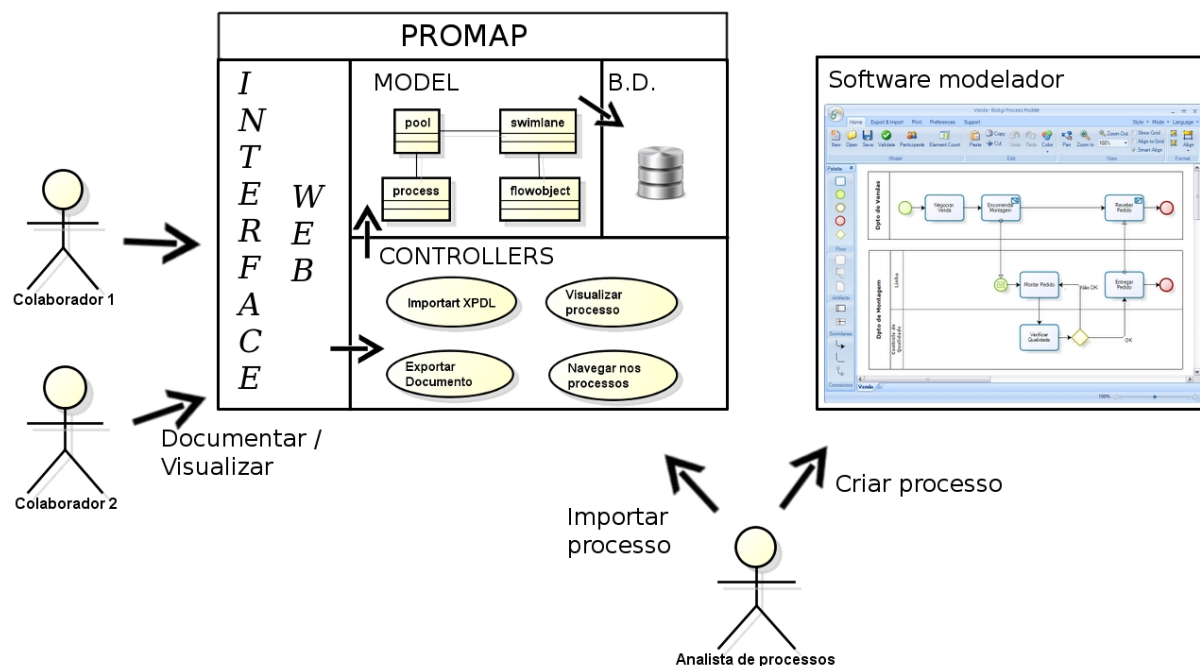
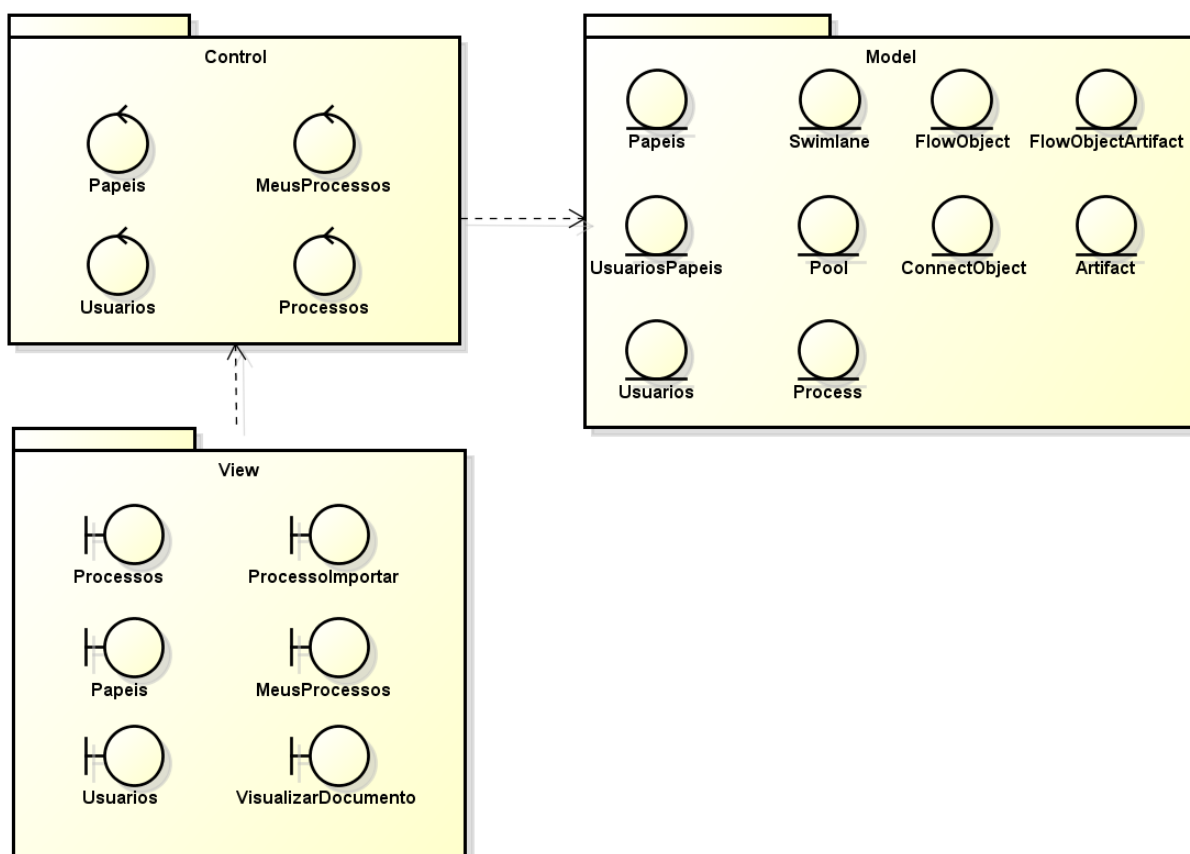


Figura 33 Visão geral.

## 4.2 Arquitetura da Ferramenta

A arquitetura da ferramenta demonstra como será feita a distribuição das classes. Utilizando o padrão Model, View e Controller (MVC), consegue-se dividir a programação, de forma a separar o trabalho em que o desenvolvedor que trabalhará nas classes de modelo, com as regras de negócio, poderá não ser o mesmo desenvolvedor que programará o *layout*. Com o padrão MVC cada desenvolvedor poderá trabalhar apenas nas suas classes, sem interferir no trabalho do outro.

O diagrama apresentado na Figura 34 demonstra uma estrutura dividida em Model, View e Controller. A View serve para formalizar uma apresentação direta com o usuário da ferramenta, a Model representa as informações gerenciadas pela aplicação e implementa as regras de negócio da mesma, e a Controller controla o fluxo das mensagens entre as camadas View e Model. Dentro do bloco Control, estão os controladores da ferramenta; dentro do bloco Model, estão os modelos que fazem conexão direta às tabelas do banco de dados; e dentro do bloco View, estão as classes utilizadas para gerar as telas.

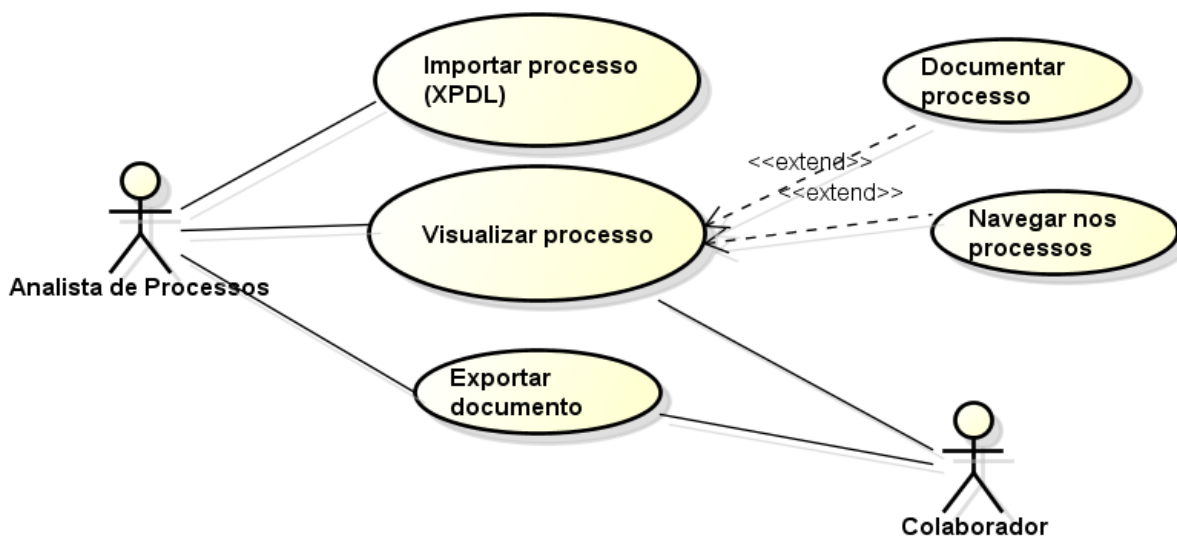


**Figura 34 Diagrama de Modelo de Arquitetura (MVC).**

### 4.3 Papéis e Responsabilidades

Para a definição dos papéis e responsabilidades será demonstrado um diagrama de casos de uso na Figura 35.

Na Figura 35, existem dois atores envolvendo-se com os casos de uso. Eles são o Analista de Processos e o Colaborador. O Analista de Processos se envolve em todos os casos de uso, diferente do Colaborador, que se envolve em todos, exceto no caso de uso de Importar processo (XPDL).



**Figura 35 Diagrama de Casos de Uso.**

Na Tabela 7, serão, brevemente, definidas as funcionalidades de cada caso de uso.

**Tabela 7 Descrição dos Casos de Uso.**

Caso de Uso	Descrição
Importar Processo (XPDL)	Essa rotina serve para importar um processo escrito em XPDL. As ferramentas BizAgi e Bonita permitem o desenho do processo e, na sequência, permitem exportá-lo para XPDL.
Documentar processo	Essa rotina é utilizada apenas pelo Analista de Processos. Nesse caso de uso, os usuários da ferramenta podem adicionar informações textuais às atividades e processos, também podem ser cadastradas informações como regras de negócio.
Visualizar processo	Ainda, é possível criar vínculos entre os processos, por exemplo, uma atividade chamada “verificar débito”, pode ser vinculada a um outro processo que serve apenas para fazer essas verificações, com isso, também, conseguimos um reaproveitamento de processos. Tela para visualização dos processos. Nessa tela, os usuários têm acesso às informações inseridas pelos colaboradores.
Navegar nos processos	Também está disponível uma visualização gráfica do processo, utilizando a ferramenta Graphviz. Para conseguir navegar de forma agradável entre os processos, será importante que os mesmos estejam vinculados uns aos outros, dessa forma podendo gerar uma visualização ampla com o

Caso de Uso	Descrição
	auxílio da ferramenta Graphviz.
	Após estarem visualizando todos os processos, podem clicar sobre eles para detalhá-los.
Exportar documento	A exportação é feita individualmente por processo, juntando as informações de suas atividades, criando um documento em que são listadas as atividades e suas descrições.  No final do documento, está incluso um fluxograma do processo gerado através da ferramenta Graphviz.

#### 4.4 Modelo de Domínio

Modelos de Domínio são utilizados para representar um problema ou definição de uma futura estrutura. Por ser apenas conceitual auxilia na definição do vocabulário a ser usado entre os envolvidos. O modelo representa relações entre entidades, assim como atributos e métodos, lembrando que, por ser conceitual, na prática, poderá sofrer modificações.

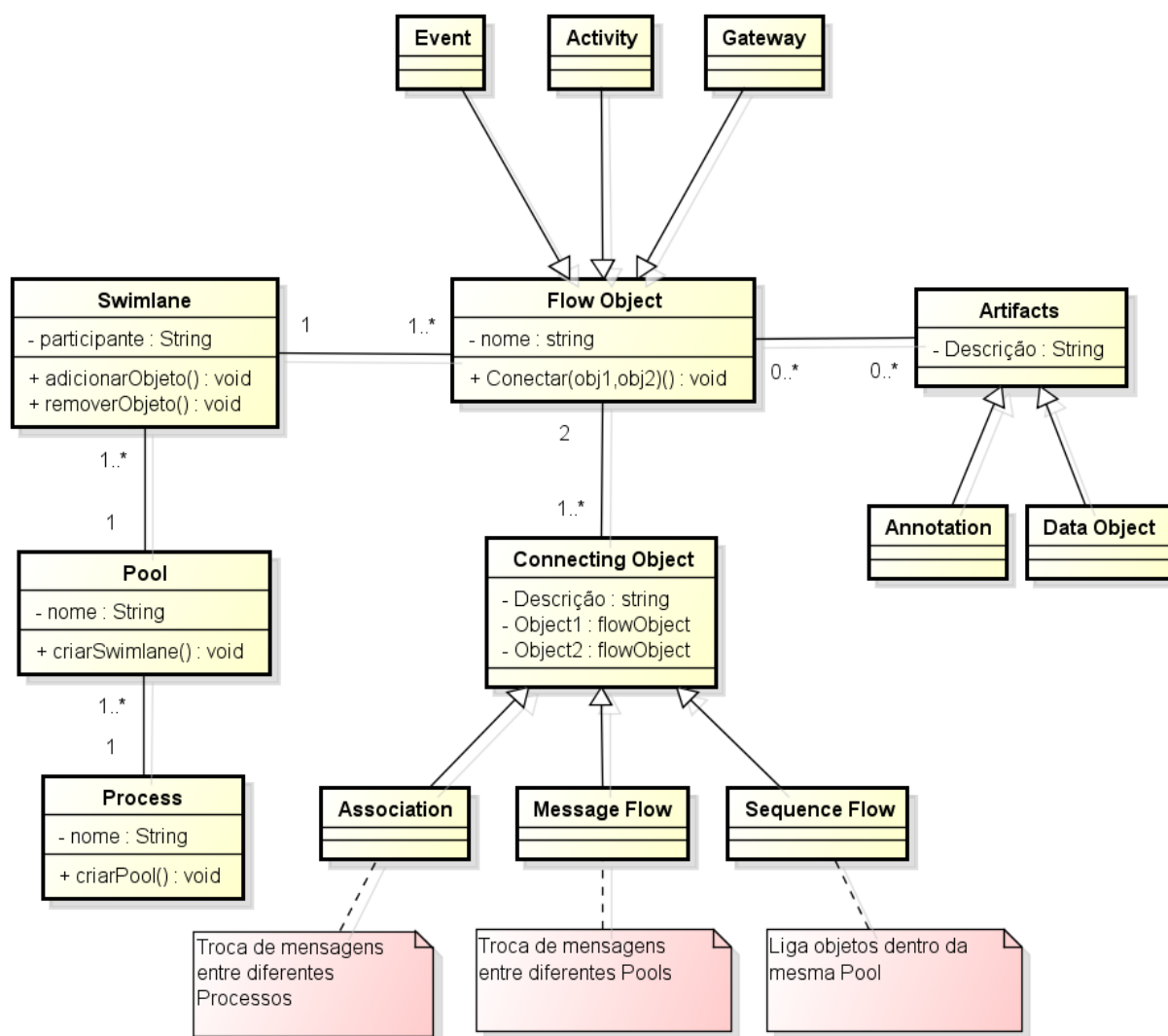
O modelo de domínio a seguir procura representar os conceitos envolvidos em um processo, bem como um conceito de como eles serão armazenados dentro da ferramenta proposta.

Para representarmos um processo em forma de classes, desenvolvemos o diagrama representado na Figura 36 e mapeamos o mesmo da seguinte maneira:

- Classe *Process* é a representação do processo em forma de classes. Tem, como objetivo, criar objetos do tipo *Pool*;
- Classe *Pool* é a representação de uma piscina dentro do processo. Tem, como objetivo, criar *Swimlanes*;
- Classe *Swimlane* é a representação de uma raia dentro da piscina que, por sua vez, também pode ser considerado um participante. Essa classe consegue criar os objetos de fluxo para o processo.
- Classe *Flow Object* é responsável por definir os objetos de fluxo que, por sua vez, poderá ser um Evento, uma Atividade ou um *Gateway*. Essa classe também pode ter artefatos relacionados, sendo estes de tipo *Annotation* ou *Data Object*.
- Classe *Connecting Object* é responsável por manter a conexão entre dois objetos de fluxo. A conexão poderá ser de três tipos:



- *Association*, que acontece quando ocorre a troca de mensagens entre processos diferentes.
- *Message Flow* é usada quando há troca de mensagens entre diferentes *Pools*.
- *Sequence Flow* liga objetos dentro de uma mesma *Pool*.
- Classe *Artifact* é utilizada para adicionar artefatos em um *Flow Object*. Os mesmos servem apenas como documentação do processo e podem ser de dois tipos:
  - *Annotation*: texto simples para adicionar alguma observação ao processo.
  - *Data Object*: adiciona uma imagem de objeto, utilizada para representar algum documento que seja necessário nessa atividade.



**Figura 36** Proposta de Diagrama de Domínio.

#### 4.5 Modelo de classes

O modelo de classes, exposto na Figura 37, representa os conceitos envolvidos em um processo, bem como a forma pela qual eles serão armazenados dentro da ferramenta.

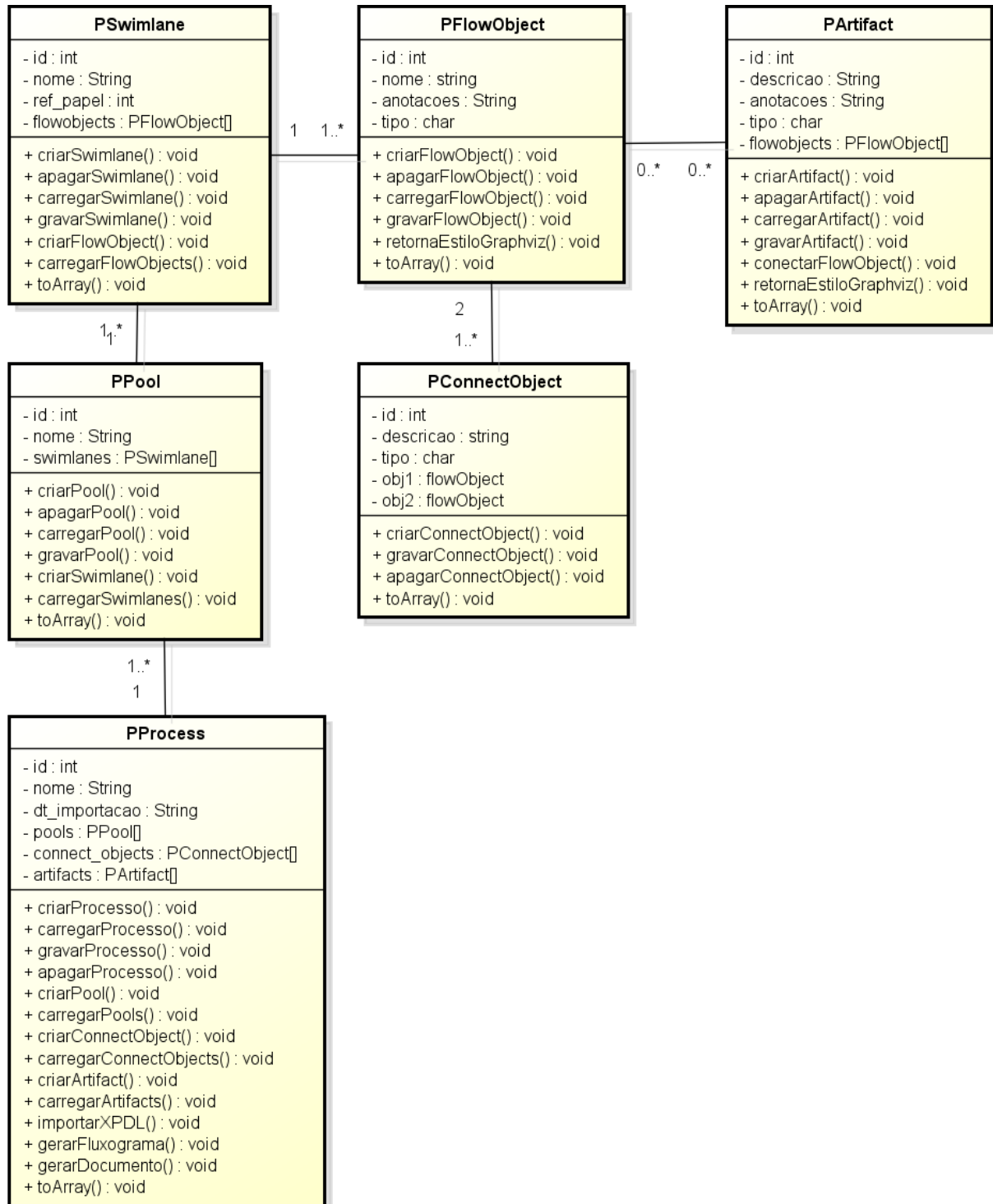


Figura 37 Diagrama de Classes.

As tabelas enumeradas de 8 até 13 dispõem uma breve descrição das classes, atributos e métodos existentes no ProMap.

**Tabela 8 Descrição da classe PProcess.**

Classe	PProcess
Representa o processo na estrutura BPM.	
<b>Atributos</b>	
id	Código no banco de dados.
nome	Nome.
dt_importacao	Data de importação.
pools	<i>Pools</i> que estão dentro deste processo.
connect_objects	Conexões do processo.
artifacts	Artefatos dentro do processo.
<b>Métodos</b>	
criarProcesso()	Cria um novo processo.
carregarProcesso()	Carrega um processo do banco de dados bem como suas <i>Pools</i> e toda estrutura do processo.
gravarProcesso()	Grava o processo no banco de dados.
apagarProcesso()	Apaga o processo do banco de dados.
criarPool()	Cria uma <i>pool</i> para este processo.
carregarPools()	Carrega as <i>Pools</i> deste processo.
criarConnectObject()	Cria uma nova conexão entre dois Flow Objects.
carregarConnectObjects()	Carrega as conexões.
criarArtifact()	Cria um novo artefato.
carregarArtifacts()	Carrega os artefatos.
importarXPDL()	Importa um arquivo XPDL.
gerarFluxograma()	Gera o desenho (png) a partir da classe Graphviz.
gerarDocumento()	Gera o documento pdf contendo todas informações sobre um processo.
toArray()	Retorna o objeto em formato de Array.

Tabela 9 Descrição da classe PPool.

Classe	PPool
Representa uma piscina na estrutura BPM.	
<b>Atributos</b>	
id	Código no banco de dados.
nome	Nome do processo ou subprocesso dentro do processo principal (ex: avaliação de crédito).
swimlanes	<i>Swimlanes</i> que estão dentro da <i>Pool</i> .
<b>Métodos</b>	
criarPool()	Cria uma nova <i>Pool</i> .
carregarPool()	Carrega dados da <i>Pool</i> a partir do banco de dados.
apagarPool()	Apaga a <i>Pool</i> do banco de dados.
gravarPool()	Grava a <i>Pool</i> no banco de dados.
criarSwimlane()	Cria uma <i>Swimlane</i> dentro desta <i>Pool</i> .
carregarSwimlanes()	Carrega as <i>Swimlanes</i> desta <i>Pool</i> .
toArray()	Retorna o objeto em formato de Array.

Tabela 10 Descrição da classe PSwimlane.

Classe	PSwimlane
Representa a raia na estrutura BPM.	
<b>Atributos</b>	
id	Código no banco de dados.
nome	Nome do participante, geralmente se trata de cargos que operam com o processo, também denominados papéis. (Ex: Gerente de Contas, Gerente de Marketing, Setor de Manutenção).
ref_papel	Código do papel armazenado no banco de dados.
flowobjects	Flow Objects que estão dentro da <i>Swimlane</i> .
<b>Métodos</b>	
criarSwimlane()	Cria uma nova <i>Swimlane</i>
apagarSwimlane()	Apaga a <i>Swimlane</i> .
carregarSwimlane()	Carrega a <i>Swimlane</i> do banco de dados.
gravarSwimlane()	Grava a <i>Swimlane</i> no banco de dados.
criarFlowObject()	Cria um <i>Flow Object</i> dentro desta <i>Swimlane</i> .
carregarFlowObjects()	Carrega os Flow Objects desta <i>Swimlane</i> .
toArray()	Retorna o objeto em formato de Array.

Tabela 11 Descrição da classe PFlowObject.

Classe	PFlowObject
Representa um objeto na estrutura BPM, podendo ser um evento, uma atividade, um subprocesso ou um <i>gateway</i> .	
<b>Atributos</b>	
id	Código no banco de dados.
nome	Nome do objeto (Ex: Pagamento efetuado?, Gerar senha de atendimento).
anotacoes	Anotações feitas internamente no Promap neste objeto.
tipo	Tipo de FlowObject, podendo ser: ES (Evento de início) EE (Evento de finalização) AA (Atividade normal) AS (Sub processo) GW ( <i>Gateway</i> )
<b>Métodos</b>	
criarFlowObject()	Cria um novo FlowObject. Neste método já é definido o tipo.
carregarFlowObject()	Carrega o FlowObject com dados do banco de dados.
apagarFlowObject()	Apaga o FlowObject do banco de dados.
gravarFlowObject()	Grava o FlowObject no banco de dados.
retornaEstiloGraphviz()	Retorna o estilo apropriado para este objeto, a diferença é baseada no atributo tipo.
toArray()	Retorna o objeto em formato de Array.

Tabela 12 Descrição da classe PArtifact.

Classe	PArtifact
Representa o artefato na estrutura BPM.	
<b>Atributos</b>	
id	Código no banco de dados.
descrição	Descrição do artifact (Ex: Formulário de solicitação de crédito).
anotacoes	Anotações feitas internamente no Promap neste Artifact.
tipo	Tipo de Artifact, podendo ser dos tipos: A (Anotation) - Uma breve anotação D (Data object) - Imagem que representa algum conteúdo como por ex. formulário
flowobjects	Flow Objects que estão vinculados a este artefato.
<b>Métodos</b>	
criarArtifact()	Cria um novo Artifact. Neste método já é definido o tipo.

apagarArtifact()	Apaga o Artifact.
carregarArtifact()	Carrega o Artifact do banco de dados.
gravarArtifact()	Grava o Artifact no banco de dados.
conectarFlowObject()	Conecta um FlowObject a este Artifact.
retornaEstiloGraphviz()	Retorna o estilo apropriado para este objeto, a diferença é baseada no atributo tipo.
toArray()	Retorna o objeto em formato de Array.

Tabela 13 Descrição da classe PConnectObject.

Classe	PConnectObject
Representa a conexão entre os objetos na estrutura BPM.	
<b>Atributos</b>	
id	Código no banco dados.
descrição	Descrição da conexão (Ex: Sim, Não, também pode não ter descrição).
tipo	Tipo da conexão, podendo ser dos tipos: A (Association) - Troca de mensagem entre diferentes processos M (Message Flow) - Troca de mensagem entre diferentes <i>Pools</i> S (Sequence Flow) - Troca de mensagem dentro da mesma <i>Pool</i>
obj1: flowObject	Objeto do tipo <i>Flow Object</i> que cria a ligação para o próximo.
obj2: flowObject	Objeto do tipo <i>Flow Object</i> que recebe a ligação do anterior.
<b>Métodos</b>	
criarConnectObject()	Cria um novo Connect Object. Neste momento já são passados os dois objetos que serão conectados.
gravarConnectObject()	Grava o Connect Object no banco de dados.
apagarConnectObject()	Apaga o Connect Object do banco de dados.
toArray()	Retorna o objeto em formato de Array.

#### 4.5.1 Utilização das classes

A seguir é apresentada a forma com que as classes são utilizadas para criar um novo processo e armazená-lo no banco de dados, assim como carregar um processo do banco de dados e para excluir um processo no mesmo.

A Figura 38 demonstra a criação de um novo processo e o armazenamento do mesmo no banco de dados.

```

<?
$process = new PProcess();
$process->criarProcesso('Processo');

$pool = $process->criarPool('Pool');

$swimlane = $pool->criarSwimlane('Swimlane',2); // Passando o código do papel no BD

### Criação e conexão de objetos ###

$flowobject1 = $swimlane->criarFlowObject('Evento inicial', 'ES');
$flowobject2 = $swimlane->criarFlowObject('Atividade 1', 'AA');
$flowobject3 = $swimlane->criarFlowObject('Atividade 2', 'AA');
$flowobject4 = $swimlane->criarFlowObject('Evento final', 'EE');

$conn1 = $process->criarConnectObject( null, $flowobject1, $flowobject2, 'S' );
$conn2 = $process->criarConnectObject( null, $flowobject2, $flowobject3, 'S' );
$conn3 = $process->criarConnectObject( null, $flowobject3, $flowobject4, 'S' );

### Tratamento de artefatos ###

$artifact1 = $process->criarArtifact('Anotação', 'A');
$artifact2 = $process->criarArtifact('Data Object', 'D');

$artifact1->conectarFlowObject( $flowobject2 );
$artifact1->conectarFlowObject( $flowobject3 );
$artifact2->conectarFlowObject( $flowobject2 );

### Gravação do processo ###

$process->gravarProcesso();
?>

```

**Figura 38 Criação de processo utilizando as Classes.**

A Figura 39 demonstra o carregamento de um processo do banco de dados. Após o carregamento, o código fonte exibe uma forma para capturar o nome dos papéis deste processo.

```

<?
$process = new PProcess();
$process->carregarProcesso('4');

foreach( $process->pools as $pool )
{
    foreach( $pool->swimlanes as $swimlane )
    {
        echo( $swimlane->nome );
    }
}
?>

```

**Figura 39 Carregamento de processo utilizando as Classes.**

A Figura 40 demonstra a exclusão de um processo do banco de dados.

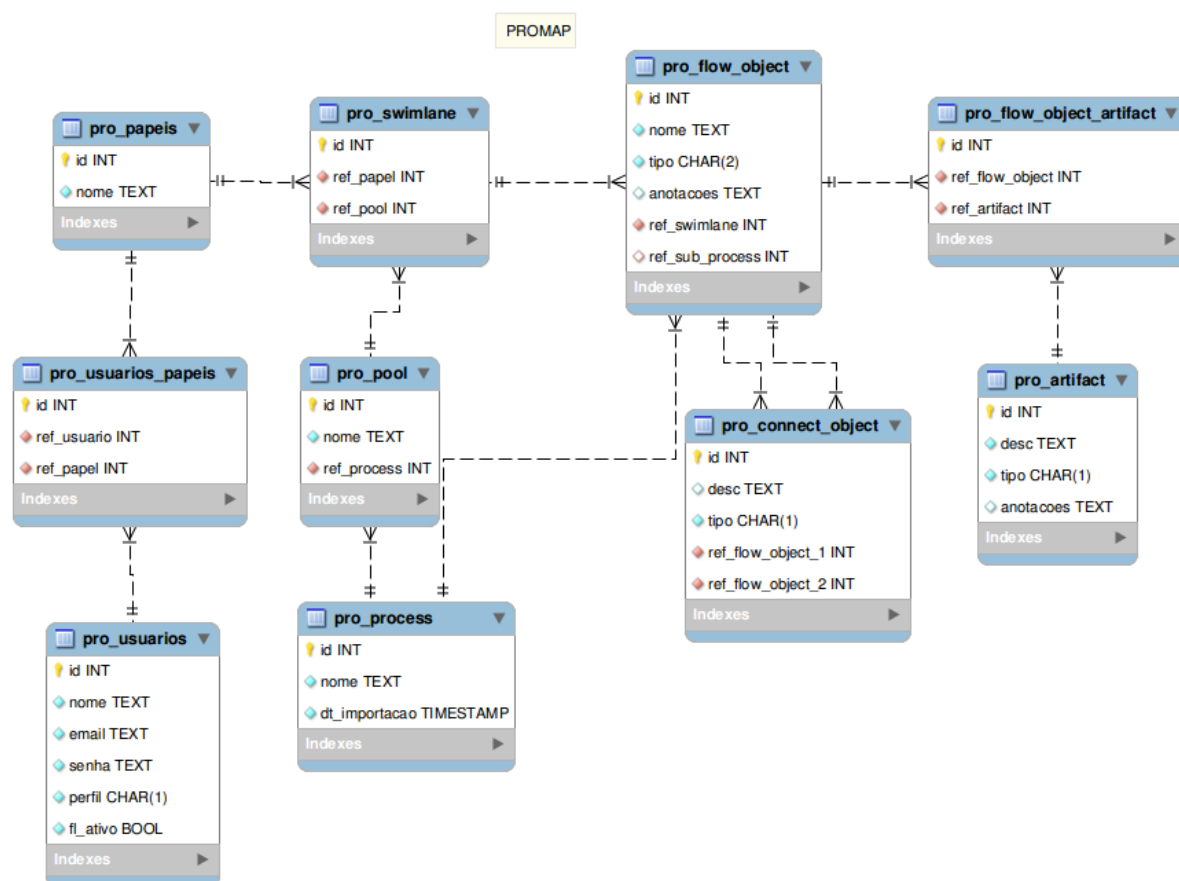
```
<?
$process = new PProcess();
$process->carregarProcesso('82');

$process->apagarProcesso();
?>
```

**Figura 40** Exclusão de processo utilizando as Classes.

#### 4.6 Modelo Relacional

O modelo Relacional, também conhecido como Diagrama E.R. (Entidade Relacionamento), foi desenvolvido baseado no diagrama de classes. O mesmo é demonstrado na Figura 41.



**Figura 41** Diagrama E.R. do ProMap.

As Tabelas enumeradas de 14 até 23 dispõem uma breve descrição das tabelas e dos campos do Diagrama E.R. do ProMap.



**Tabela 14 Descrição da tabela pro\_usuarios.**

<b>Tabela</b>	<b>pro_usuarios</b>
Tabela destinada a armazenar os usuários que terão acesso ao sistema.	
<b>Campos</b>	
id	Código único que cada usuário tem no sistema.
nome	Nome do usuário.
email	Email do usuário. Utilizado como Login para acessar o sistema.
senha	Senha do usuário para acessar o sistema.
perfil	Pode ser Analista de Processos (A) ou Colaborador (C).
fl_ativo	Define se o usuário poderá acessar o sistema.

**Tabela 15 Descrição da tabela pro\_papeis.**

<b>Tabela</b>	<b>pro_papeis</b>
Tabela destinada a armazenar os papéis no sistema. Eles podem ser cadastrados como setores ou cargos da empresa.	
<b>Campos</b>	
id	Código único do papel.
nome	Nome do papel.

**Tabela 16 Descrição da tabela pro\_usuarios\_papeis.**

<b>Tabela</b>	<b>pro_usuarios_papeis</b>
Tabela utilizada para relacionar os usuário a um determinado papel no sistema.	
<b>Campos</b>	
id	Código único para representar um usuário em um papel.
ref_usuario	Código único da tabela pro_usuarios.
ref_papel	Código único da tabela pro_papeis.

**Tabela 17 Descrição da tabela pro\_process.**

<b>Tabela</b>	<b>pro_process</b>
Tabela destinada ao armazenamento das informações básicas do processo.	
<b>Campos</b>	
id	Código único do processo.
nome	Nome do processo.
dt_importacao	Data em que o processo foi importado.

Tabela 18 Descrição da tabela pro\_pool.

Tabela	pro_pool
Tabela destinada ao armazenamento de informações das Pools do processo.	
Campos	
id	Código único da <i>Pool</i> .
nome	Nome da <i>Pool</i> .
ref_process	Código único da tabela pro_process.

Tabela 19 Descrição da tabela pro\_swimlane.

Tabela	pro_swimlane
Tabela destinada ao armazenamento de informações das <i>Swimlanes</i> .	
Campos	
id	Código único da <i>Swimlane</i> .
ref_papel	Código único da tabela de pro_papeis.
ref_pool	Código único da tabela pro_pool.

Tabela 20 Descrição da tabela pro\_flow\_object.

Tabela	pro_flow_object
Tabela destinada ao armazenamento dos objetos que estarão presentes no processo.	
Campos	
id	Código único do <i>Flow Object</i> .
nome	Nome do objeto, podendo não ser preenchido em casos como início ou fim do processo.
tipo	Tipo do objeto, podendo ser ES (Evento de início), EE (Evento de finalização), AA (Atividade normal), AS (Sub processo), GW ( <i>Gateway</i> ).
anotacoes	Aqui são armazenadas as informações textuais adicionadas pelos colaboradores.
ref_swimlane	Código único da tabela pro_swimlane.
ref_sub_process	Código único da tabela pro_process, esse campo é utilizado para representar que essa atividade possui um subprocesso.

Tabela 21 Descrição da tabela pro\_connect\_object.

Tabela	pro_connect_object
Tabela utilizada para fazer a ligação entre os Flow Objects.	
Campos	
id	Código único da conexão.
desc	Texto descritivo para a conexão, cujo preenchimento não é de obrigatório.
tipo	Tipo da conexão, podendo ser A (Association), M (Message Flow) ou S (Sequence Flow).
ref_flow_object_1	Código único do objeto originário da conexão da tabela pro_flow_object.
ref_flow_object_2	Código único do objeto destinatário da conexão da tabela pro_flow_object.

Tabela 22 Descrição da tabela pro\_artifact.

Tabela	pro_artifact
Tabela contendo informações dos artefatos que estão no sistema.	
Campos	
id	Código único do artefato.
desc	Descrição, nome para o artefato.
tipo	Tipo do artefato, podendo ser A (Anotation) ou D (Data object).
anotacoes	Aqui são armazenadas as informações textuais adicionadas pelos colaboradores.

Tabela 23 Descrição da tabela pro\_flow\_object\_artifact.

Tabela	pro_flow_object_artifact
Tabela utilizada para vincular um artefato a um <i>Flow Object</i> .	
Campos	
id	Código único para a ligação.
ref_flow_object	Código único da tabela pro_flow_object.
ref_artifact	Código único da tabela pro_artifact.

## 4.7 Especificação dos Casos de Uso

Segue, abaixo, a descrição completa para cada caso e uso desenvolvido.

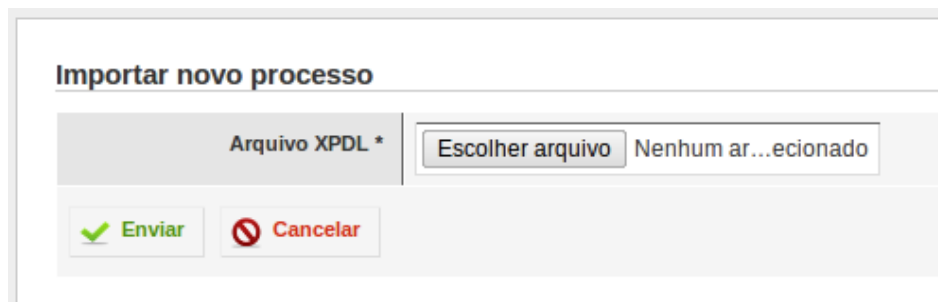
### 4.7.1 Importar Processo (XPDL)

A Tabela 24 dispõe de uma breve descrição sobre o Caso de Uso.

**Tabela 24 Descrição do caso de uso Importar Processo (XPDL).**

Caso de Uso	Importar Processo (XPDL)
	Permitir ao usuário importar um processo no formato XPDL para dentro da estrutura de classes proposta no sistema ProMap.
<b>Atores</b>	
	Analista de Processos
<b>Fluxo básico</b>	
	<ol style="list-style-type: none"> <li>1. Selecionar o arquivo a ser importado.</li> <li>2. Após a análise do arquivo, o sistema irá verificar quais são as <i>Swimlanes</i>.</li> <li>3. O sistema solicitará ao Analista vincular as <i>Swimlanes</i> com papéis previamente cadastrados no sistema.</li> <li>4. Analista pode alterar o nome do processo neste momento e continuar com o processo.</li> <li>5. Sistema exibe uma mensagem, dizendo que o processo foi importado com sucesso.</li> </ol>
<b>Fluxo de exceção</b>	
	<ol style="list-style-type: none"> <li>1. Caso o arquivo importado não seja reconhecido como um XML válido, emite uma mensagem de erro e aborta a importação.</li> </ol>

Na sequência, na Figura 42, o Analista de Processos seleciona o arquivo XPDL.



**Figura 42 Primeiro passo para importar um processo.**

Na sequência, na Figura 43, o Analista de Processos deve selecionar o papel adequado para cada *Swimlane*, também, neste momento, pode alterar o nome do processo.



**Importar novo processo - Seleção de papéis**

Arquivo:	Compra2.xpdl
Nome do processo a ser importado:	Compra
<b>Papéis:</b>	
Setor de estoque	Analista de Processos ▼
Setor financeiro	Analista de Processos ▼
Setor inerente	Analista de Processos ▼
<input type="button" value="✓ Enviar"/> <input type="button" value="✗ Cancelar"/>	

**Figura 43 Segundo passo para importar um processo.**

#### 4.7.2 Visualizar processo

A Tabela 25 dispõe de uma breve descrição sobre o Caso de Uso.

**Tabela 25 Descrição do caso de uso Visualizar processo.**

Caso de Uso	Visualizar Processo
Permitir ao usuário visualizar graficamente um processo importado.	
<b>Atores</b>	
Analista de Processos e Colaborador	
<b>Fluxo básico</b>	
<ol style="list-style-type: none"> <li>1. Usuário clica em Meus Processos.</li> <li>2. Sistema exibe uma tela com todos os processos de que ele participa.</li> <li>3. Usuário seleciona um dos processos.</li> <li>4. Sistema exibe uma tela contendo informações do processo, tais como nome, data de importação, participantes e processos relacionados. Além dessas informações, aparece um fluxograma do processo.</li> </ol>	
<b>Fluxo de exceção</b>	
<ol style="list-style-type: none"> <li>1. Caso o usuário não tenha processos, aparece uma mensagem informando que não possui processos cadastrados.</li> </ol>	

Na sequência, na Figura 44, o ator visualiza, no menu lateral, todos os processos de que ele faz parte.

### Meus Processos

**MEUS PROCESSOS**

- \* Autorização de Matrícula
- \* Compra
- \* Documentar e visualizar Processos
- \* Importar Processo (XPDL)
- \* Reingresso
- \* Todos Elementos Utilizáveis
- \* Transferência Interna

### Selecione um processo no menu lateral

Figura 44 Primeiro passo para visualizar um processo.

Na sequência, na Figura 45, após selecionar um processo, o ator consegue visualizar diversas informações do processo no menu lateral, já, no centro da tela, ele consegue visualizar o fluxograma do processo.

### Meus Processos

**PROCESSO ATUAL**

Nome: Autorização de Matrícula  
Importado em: 04-05-2012

**PARTICIPANTES:**

- \* Secretaria Geral

**PROCESSOS RELACIONADOS**

- \* Reingresso
- \* Transferência Interna

**VISUALIZAR DOCUMENTO**

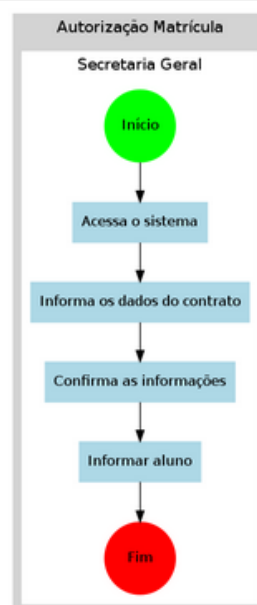
- \* Versão HTML
- \* Versão PDF

**MEUS PROCESSOS**

- \* Autorização de Matrícula
- \* Compra
- \* Documentar e visualizar Processos
- \* Importar Processo (XPDL)
- \* Reingresso
- \* Todos Elementos Utilizáveis
- \* Transferência Interna

### Processo atual: Autorização de Matrícula

Zoom: 50%



Processo: Autorização de Matrícula

Figura 45 Segundo passo para visualizar um processo.

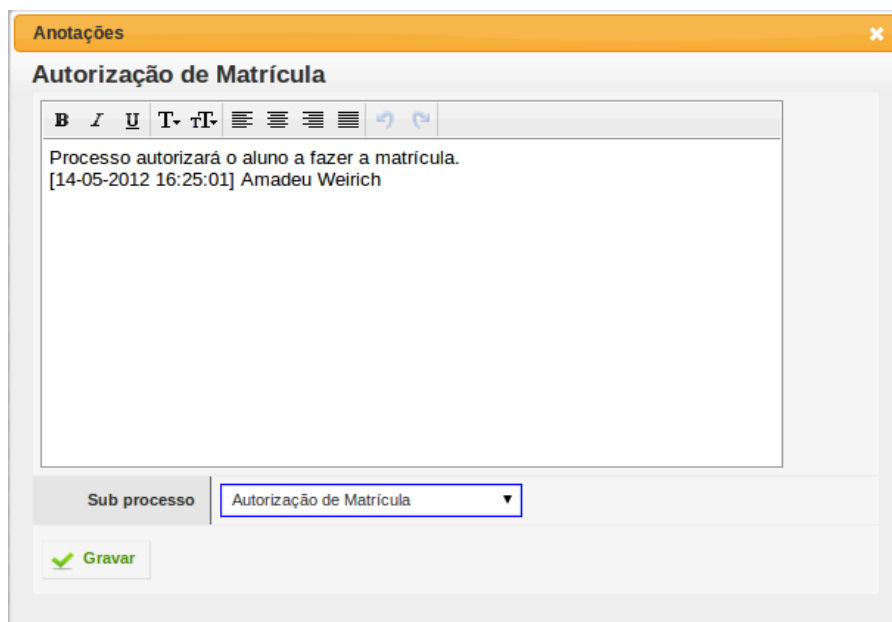
### 4.7.3 Documentar processo

A Tabela 26 dispõe de uma breve descrição sobre o Caso de Uso.

**Tabela 26 Descrição do caso de uso Documentar processo.**

Caso de Uso	Documentar Processo
<p>Permitir ao usuário descrever textualmente comentários sobre um determinado processo. Assim, auxiliando os leitores para uma melhor compreensão do processo. Descrições que não são facilmente explicadas com imagens, podem ser adicionadas aqui.</p> <p>Além de inserir descrições textuais, o usuário pode vincular outros processos a este. Por exemplo, definir qual ou quais os processos que são efetuados para que o processo atual seja processado, uma espécie de requisito para que o processo seja rodado.</p>	
<b>Atores</b>	
Analista de Processos e Colaborador	
<b>Fluxo básico</b>	
<ol style="list-style-type: none"> <li>1. Usuário clica em Meus Processos.</li> <li>2. Sistema exibe uma tela com todos os processos de que ele participa.</li> <li>3. Usuário seleciona um dos processos.</li> <li>4. É gerada a tela com o fluxograma do processo.</li> <li>5. Usuário clica em uma atividade ou artefato deste fluxograma.</li> <li>6. Sistema exibe uma tela com opções de adição de conteúdo textual e opção de vincular a atividade a algum outro processo.</li> <li>7. Usuário entra com as informações desejadas e clica em Finalizar.</li> <li>8. O conteúdo é gravado na atividade ou artefato.</li> </ol>	
<b>Fluxo de Exceção</b>	
<ol style="list-style-type: none"> <li>1. Pode ocorrer de não haver outro processo para ser vinculado. Então o usuário não terá a opção de vínculo de subprocesso.</li> </ol>	

Na sequência, na Figura 46, o ator visualizará este formulário, ao clicar em uma atividade, nele poderá entrar com informações textuais e, além disso, vincular um processo a outro.



**Figura 46 Tela para inserção de anotações.**

#### 4.7.4 Navegar nos processos

A Tabela 27 dispõe de uma breve descrição sobre o Caso de Uso.

**Tabela 27 Descrição do caso de uso Navegar nos processos.**

Caso de Uso	Navegar nos processos
Permitir ao usuário acessar um determinado processo e ter links diretos para os processos que têm relação com o mesmo.	
<b>Atores</b>	
Analista de Processos e Colaborador	
<b>Fluxo básico</b>	
<ol style="list-style-type: none"> <li>1. Usuário clica em Meus Processos.</li> <li>2. Sistema exibe uma tela com todos os processos de que ele participa.</li> <li>3. Usuário seleciona um dos processos.</li> <li>4. Nessa tela são listados os demais processos vinculados.</li> <li>5. Usuário clica em um desses processos para acessá-lo.</li> </ol>	
<b>Fluxo de Exceção</b>	
<ol style="list-style-type: none"> <li>1. Pode ocorrer de não ter processos vinculados, então não aparecerá opção para entrar em outros processos.</li> </ol>	



Na sequência, na Figura 47, o ator visualizará a tela contendo um fluxograma ao centro e, no menu lateral, diversas informações sobre o processo. Nesse menu lateral, existe um submenu sob o nome Processos relacionados, no qual é possível clicar sobre os processos e navegar neles.

## Meus Processos

**PROCESSO ATUAL**

**Nome:** Autorização de Matrícula  
**Importado em:** 04-05-2012

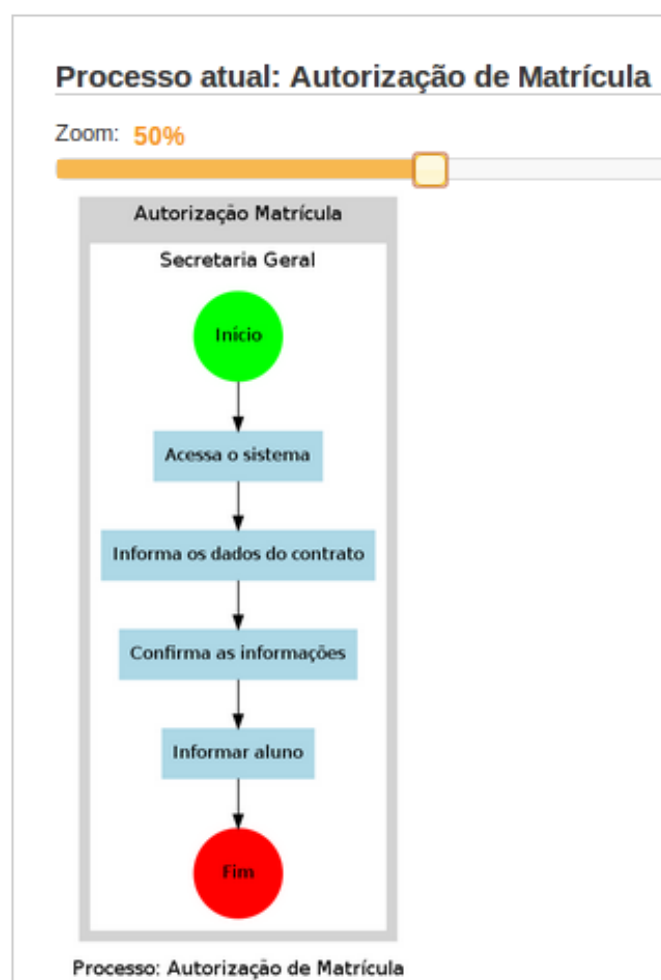
**PARTICIPANTES:**  
\* Secretaria Geral

**PROCESSOS RELACIONADOS**  
\* Reingresso  
\* Transferência Interna

**VISUALIZAR DOCUMENTO**  
\* Versão HTML  
\* Versão PDF

**MEUS PROCESSOS**

- \* Autorização de Matrícula
- \* Compra
- \* Documentar e visualizar Processos
- \* Importar Processo (XPD)
- \* Reingresso
- \* Todos Elementos Utilizáveis
- \* Transferência Interna



**Figura 47 Navegando nos processos.**

### 4.7.5 Exportar documento

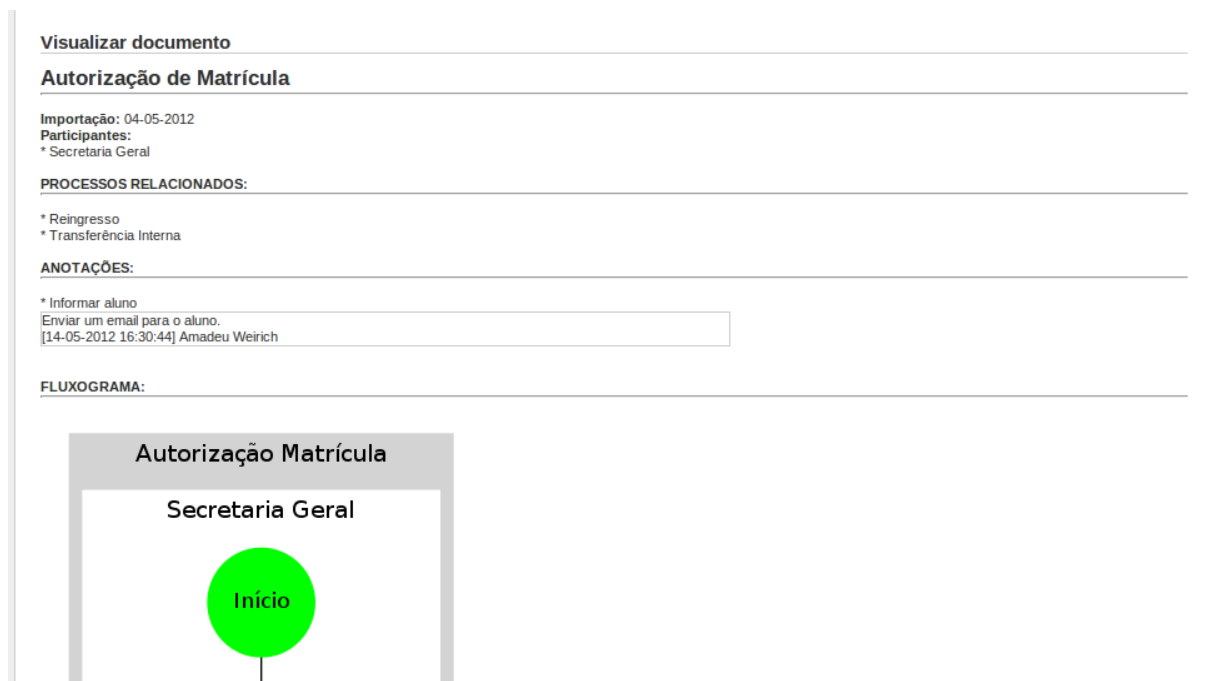
A Tabela 28 dispõe de uma breve descrição sobre o Caso de Uso.

**Tabela 28 Descrição do caso de uso Exportar documento.**

Caso de Uso	Exportar documento
	Permitir ao usuário gerar um documento com todas as características de determinado processo. O documento emitido terá a imagem do processo e toda a documentação adicionada pelos usuários no caso de uso Documentar processo.

Atores
Analista de Processos e Colaborador
Fluxo básico
<ol style="list-style-type: none"> <li>1. Usuário clica em Meus Processos.</li> <li>2. Sistema exibe uma tela com todos os processos de que ele participa.</li> <li>3. Usuário seleciona um dos processos.</li> <li>4. Na tela, irá aparecer a opção: Visualizar Documento.</li> <li>5. Usuário seleciona o formato adequado: <ol style="list-style-type: none"> <li>1. Clicando na opção visualizar HTML, abre uma tela contendo todas as informações do processo.</li> <li>2. Clicando na opção visualizar PDF, abre uma tela para fazer o download do PDF.</li> </ol> </li> </ol>

Na sequência, na Figura 48, o ator visualizará o documento em formato HTML. As informações contidas nesse documento são as mesmas visualizadas na tela de processos, mais as informações de anotações feitas nas atividades.



**Figura 48 Documento em formato HTML.**

Na sequência, na Figura 49, o ator visualizará o documento em formato PDF. As informações contidas nesse documento são as mesmas visualizadas na tela de processos, mais as informações de anotações feitas nas atividades.

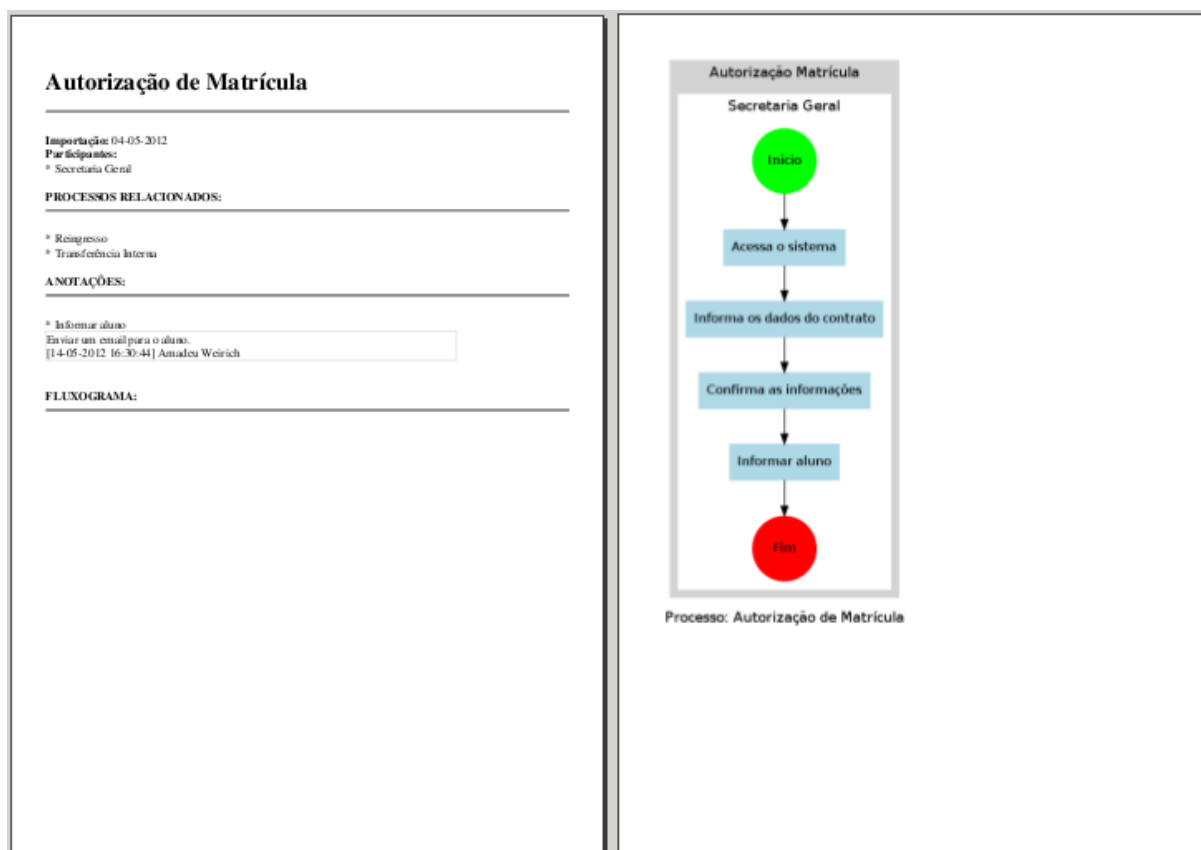


Figura 49 Documento em formato PDF.

#### 4.8 Arquitetura dos serviços

A arquitetura dos serviços representa os serviços que serão disponibilizados para que outras ferramentas possam usufruir das informações contidas dentro do Promap.

A Figura 50 demonstra, de forma ampla, todos os componentes que trabalham com o Promap. Dentro do componente Promap, temos três componentes que farão conexão com os componentes externos, são eles:

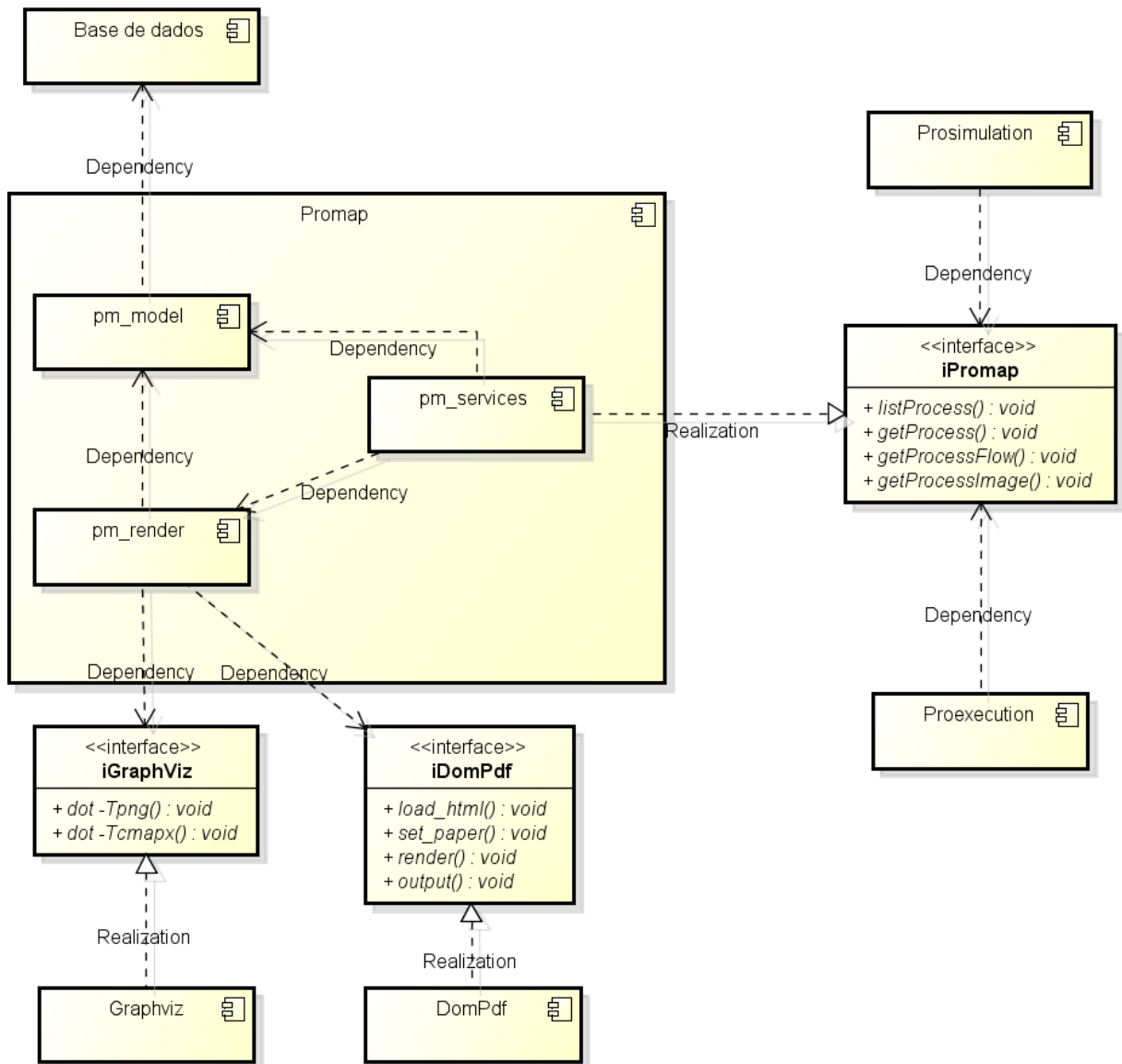
- pm\_model: Representa a camada *Model* da aplicação;
- pm\_render: Trata-se de uma camada de conexão com uma ferramenta externa;
- pm\_services: esse componente implementa uma interface chamada iPromap, que define os serviços disponíveis para que outras ferramentas possam se conectar ao Promap.

Tanto o pm\_services quanto o pm\_render são dependentes do pm\_model.

Existem dois componentes que são dependentes da Interface iProcess, são eles: Prosimulation e Proexecution. Ambos componentes poderão ser desenvolvidos em outros

trabalhos. Neste diagrama, servirão apenas como exemplos de outras ferramentas que poderão fazer consultas à interface.

A `pm_render` faz chamadas para os programas externos GraphViz e DomPdf<sup>2</sup>. Ambos programas disponibilizam funcionalidades que são utilizadas para geração de um fluxograma e de PDF para o ProMap. Abaixo, nas Tabelas 30 e 31, consta uma breve descrição sobre os métodos utilizados desses softwares.



**Figura 50 Diagrama de Componentes.**

A Tabela 29 descreve serviços que serão disponibilizados para outras ferramentas consultarem informações sobre os processos.

<sup>2</sup>Biblioteca disponível em <http://code.google.com/p/dompdf/>

**Tabela 29 Descrição dos serviços disponibilizados pelo Promap.**

Método	Descrição
listProcess()	Esse serviço retorna uma lista em formato Array com o código e o nome de todos os processos. Esse método aceita um parâmetro, o nome do Papel, dessa forma irá trazer apenas os processos relativos a determinado Papel no sistema.
getProcess()	Esse serviço retornará ao solicitante todas informações contidas em um processo, desde suas atividades até as descrições. As informações estarão dispostas em formato Array com os mesmos atributos que estão descritos no diagrama de classes.
getProcessFlow()	Esse serviço retornará os processos de origem e destino vinculados ao processo consultado. O resultado estará em formato de Array com duas posições, uma “origem” e outra “destino”.
getProcessImage()	Esse serviço retorna a imagem do fluxograma gerada pelo Graphviz.

**Tabela 30 Descrição dos métodos da interface Graphvz.**

Método	Descrição
dot -Tpng	Esse método gera a imagem png do fluxograma.
dot -Tcmapx	Esse método gera um “map” em estrutura HTML para que os pontos no fluxograma possam conter links referenciáveis.

**Tabela 31 Descrição dos métodos da interface DomPdf.**

Método	Descrição
load_html()	Carrega o HTML.
set_paper()	Marca o tipo de papel, A4 e paisagem.
render()	Renderiza o html em formato PDF.
output()	Joga o arquivo na tela ou em uma variável.

#### 4.8.1 Utilização dos webservice

A seguir é demonstrado, brevemente, como um cliente fará a conexão ao ProMap para consultar informações referentes aos processos.

Na Figura 51, é demonstrada a forma com que o cliente faz a conexão e utiliza o método `listProcess()`. Para a conexão, o cliente instancia a “SoapClient” com os parâmetros de conexão, em seguida faz a chamada do método e, para finalizar, chama a função “`is_soap_fault`” para verificar se houve erro na conexão. Nessa chamada, é passado o parâmetro “Secretaria Geral”, fazendo com que o método retorne apenas processos de que este Papel participa.

```
<?
// passando o endereço do servidor
$client = new SoapClient(null, array(
    'location' => 'http://www.amadeuw.net/promap/index.php/soap',
    'uri' => 'http://www.amadeuw.net/promap/',
    'trace' => 1));

// chamada do serviço SOAP
$result = $client->listProcess('Secretaria Geral');

// verifica erros na execução do serviço e exibe o resultado
if (is_soap_fault($result)){
    trigger_error("SOAP Fault: (faultcode: {$result->faultcode},
        faultstring: {$result->faultstring})", E_ERROR);
}else{
    echo "Resultado Encontrado : <br><br><pre>";
    print_r($result);
}
?>
```

**Figura 51 Webservice chamando o método `listProcess()`.**

Na Figura 52, é feito o mesmo procedimento de conexão, porém a chamada é feita para o método `getProcess()`. Passando como parâmetro “1”, o que fará com que o retorno seja toda a informação inerente ao processo em formato de Array. Nesse código apresentado, o programa exibe o nome de todas as *swimlanes*, leia-se todos os papéis participantes do processo. Dessa forma, também é possível navegar nos níveis do Array até chegar às informações das Atividades.

```

<?
// passando o endereço do servidor
$client = new SoapClient(null, array(
    'location' => 'http://www.amadeuw.net/promap/index.php/soap',
    'uri' => 'http://www.amadeuw.net/promap/',
    'trace' => 1));

// chamada do serviço SOAP
$result = $client->getProcess(1);

// verifica erros na execução do serviço e exibe o resultado
if (is_soap_fault($result)){
    trigger_error("SOAP Fault: (faultcode: {$result->faultcode},
        faultstring: {$result->faultstring})", E_ERROR);
}else{
    foreach( $result['pools'] as $pool )
    {
        foreach( $pool['swimlanes'] as $swimlane )
        {
            echo $swimlane['nome'];
        }
    }
}
?>

```

**Figura 52 Webservice chamando o método getProcess().**

Na Figura 53, é feito o mesmo procedimento de conexão, porém a chamada é feita para o método getProcessFlow(). O retorno será de um Array com as posições “origem” e “destino”.

```

<?
// passando o endereço do servidor
$client = new SoapClient(null, array(
    'location' => 'http://www.amadeuw.net/promap/index.php/soap',
    'uri' => 'http://www.amadeuw.net/promap/',
    'trace' => 1));

// chamada do serviço SOAP
$result = $client->getProcessFlow(1);

// verifica erros na execução do serviço e exibe o resultado
if (is_soap_fault($result)){
    trigger_error("SOAP Fault: (faultcode: {$result->faultcode},
        faultstring: {$result->faultstring})", E_ERROR);
}else{
    echo "Resultado Encontrado : <br><br><pre>";
    print_r($result);
}
?>

```

**Figura 53 Webservice chamando o método getProcessFlow().**

Na Figura 54, é feito o mesmo procedimento de conexão, porém a chamada é feita para o método getProcessImage(). Diferentemente das chamadas anteriores, esta requer um pouco mais de cuidado, pois, com o retorno do método, deve se utilizar a função “urldecode”,

e o resultado desse processo deve ser colocado dentro de um arquivo, assim como demonstrado na figura.

```
<?
// passando o endereço do servidor
$client = new SoapClient(null, array(
    'location' => 'http://www.amadeuw.net/promap/index.php/soap',
    'uri' => 'http://www.amadeuw.net/promap/',
    'trace' => 1));

// chamada do serviço SOAP
$result = $client->getProcessImage(1);
$fp = fopen("imagem.png", 'w');
fwrite($fp, urldecode($result));
fclose($fp);

// verifica erros na execução do serviço e exibe o resultado
if (is_soap_fault($result)){
    trigger_error("SOAP Fault: (faultcode: {$result->faultcode},
    faultstring: {$result->faultstring})", E_ERROR);
}else{
    echo "Resultado Encontrado : <br><br><pre>";
    print_r($result);
}
?>
```

Figura 54 Webservice chamando o método getProcessImage().

#### 4.9 Tecnologias utilizadas

Como apoio para o desenvolvimento dessa proposta, fez-se uso de várias ferramentas cujos propósitos são: armazenamento, interface e algoritmos. As tecnologias adotadas e que também serão mais bem detalhadas no decorrer deste capítulo, são: PostgreSQL<sup>3</sup>, como banco de dados de armazenamento; linguagem de programação PHP; JQuery<sup>4</sup>, como API Javascript auxiliadora; DomPdf, como classe utilizada para renderizar arquivos PDF; CodeIgniter<sup>5</sup>, para escrever o código na estrutura MVC e, como técnica, a Programação Orientada a Objetos.

Segundo a própria documentação do PostgreSQL, a ferramenta é considerada um sistema gerenciador de banco de dados, o que nada mais é do que um sistema computadorizado para manter registros de arquivos, fazendo com que o usuário da informação manipule-as com facilidade (DATE, 2000).

O SGBD (Sistema de Gerenciamento de Banco de Dados) PostgreSQL faz uso da metodologia objeto-relacional. Um banco de dados objeto-relacional, em sua estrutura básica, é uma coleção de Tabelas, todas com nomes únicos, que compõem a base de dados, podendo estar relacionada a uma ou mais Tabelas. Conceitos como integridade referencial de dados

<sup>3</sup>Software disponível em <http://www.postgresql.org/>

<sup>4</sup>Biblioteca disponível em <http://www.jquery.com/>

<sup>5</sup>Biblioteca disponível em <http://codeigniter.com/>



garantem que um dado referenciado em uma Tabela esteja presente na Tabela que está sendo referenciada. Chaves primárias estão presentes e garantem que um conjunto de informações possa ser representado de maneira consistente, independente da forma de acesso (BOSCARIOLI, 2006).

Também foi feito uso da linguagem PHP, que significa PHP Hypertext Preprocessor. Ela é uma linguagem de programação de ampla utilização, livre, interpretada, utilizada, em sua grande maioria, para o desenvolvimento *web* e que pode ser mesclada dentro do código HTML.

Segundo Behar (2003), a linguagem PHP possui ainda outras vantagens como: simplicidade, velocidade, segurança, facilidade de configuração e estabilidade, utilizando seu próprio sistema de administração, possuindo um método sofisticado para lidar com variáveis.

Através da linguagem PHP, foi aplicado o paradigma de Programação Orientada a Objetos, a qual faz uso de classes e objetos criados a partir dos modelos para representar e processar dados, usando programas de computadores. A modelagem dos dados e operações sobre os dados em um programa de computador permite o processamento de forma coesa, rápida e menos suscetível a erros (SANTOS, 2003).

De acordo com Sebesta (2000), a essência da programação orientada a objetos é resolver problemas, identificando os objetos do mundo real do problema e o seu processamento necessário, criando, então, simulação dos mesmos, seus processos e as comunicações necessárias entre eles.

O conceito definido por Booch (2007) para programação orientada a objetos é um método de implementação no qual programas são organizados como coleções cooperativas de objetos, cada um representando uma instância de alguma classe, e estas classes são todas membros de uma hierarquia de classes unidas via relacionamentos de herança.

O JQuery foi lançado em janeiro de 2006 por John Resgin como uma biblioteca JavaScript Open Source, para tornar mais fácil a interatividade em aplicações web. A ferramenta traz a promessa de trabalhar com os navegadores mais populares.

O DomPdf é um projeto em PHP com crescente utilização para renderizar arquivos PDF. Seu diferencial é a simplicidade para a geração dos documentos. Para gerar um documento PDF, escrevemos uma página HTML e rodamos o DomPdf sobre ela, após isso, o arquivo é renderizado e pode ser disponibilizado para download.

CodeIgniter é um framework web de código aberto amplamente utilizado em páginas e sistemas web. O CodeIgniter possui uma grande comunidade de utilizadores, um simples

manual de utilização e alta performance na renderização das páginas. Sua estrutura interna baseia-se em MVC e trabalha com estrutura orientada a objetos.

Graphviz foi a ferramenta utilizada para a geração dos fluxogramas. Ela é código aberto e teve uma fácil implementação no ProMap.

#### **4.10 Validação da ferramenta**

Para validação da ferramenta foi elaborado um questionário com 11 perguntas, sendo a maioria delas baseadas na ISO 9126 (norma ISO para qualidade de produto de software). As questões foram desenvolvidas para serem respondidas na escala Likert, em que as respostas são divididas em 5 pontos, sendo 1 ponto ruim e 5 pontos excelente.

A distribuição desse questionário foi por meio eletrônico, utilizando um formulário online (desenvolvido no Google Forms) para facilitar a coleta dos resultados. O público-alvo foram pessoas que trabalham na área de gestão de processos e graduandos na área de tecnologia da informação. O período disponibilizado para responder ao formulário foi de 23/05/2012 até 31/05/2012. Após a data final de disponibilização do formulário as respostas foram coletadas e tabuladas. O número de respondentes foi 12.

No apêndice A, está disponível o manual de usuário entregue para as pessoas que testaram a ferramenta.

No apêndice B, está representado o questionário utilizado.

Na Figura 55, está representado um gráfico com todas as respostas, de perguntas em escala Likert, recebidas pelo formulário online. Ainda, abaixo da figura, há a interpretação das respostas em forma textual.

### Perguntas e respostas

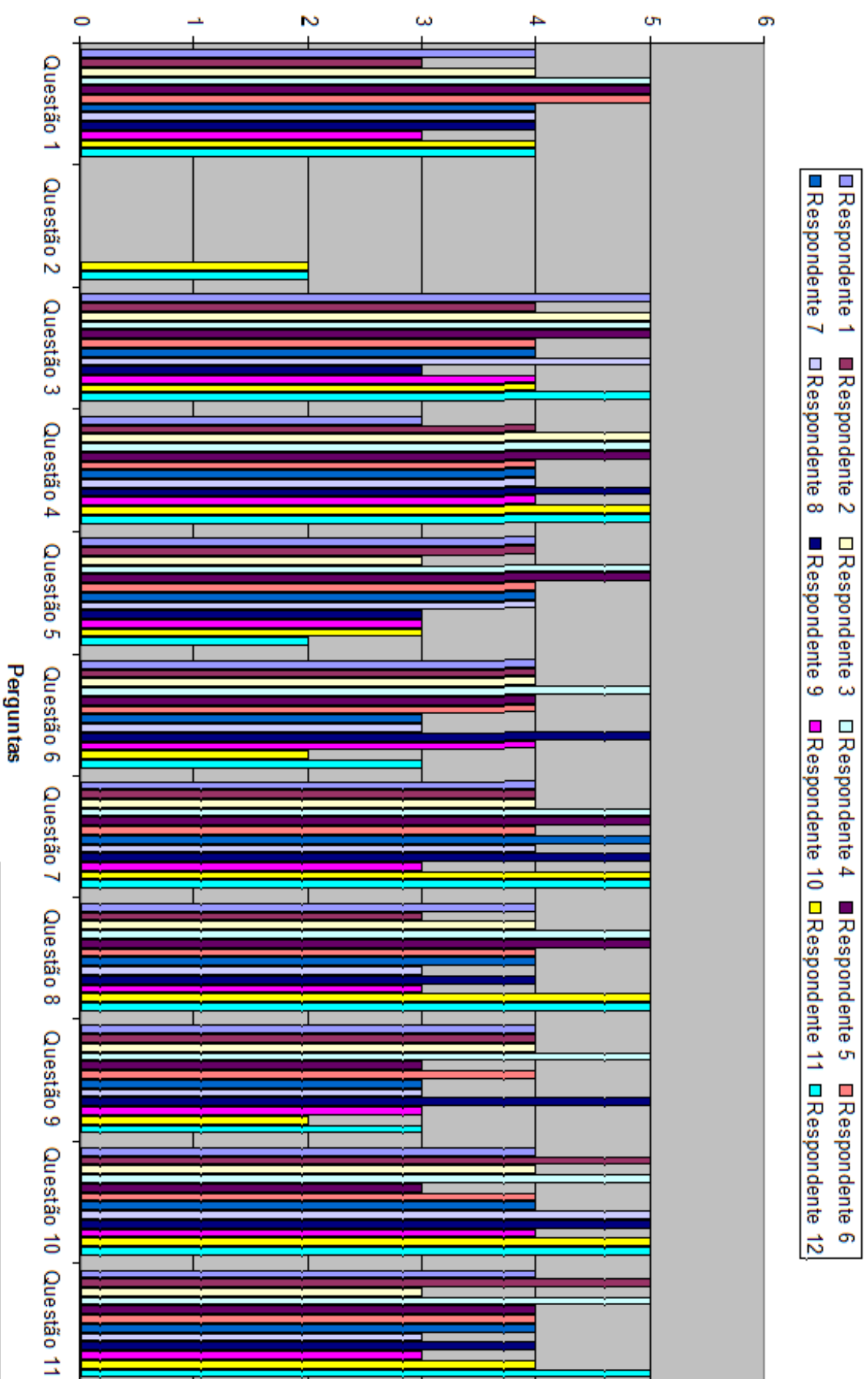


Figura 55 Respostas do formulário disponibilizado online.

Em relação ao software ser adequado para a gerência de projetos e maduro para utilização em um cenário real, o resultado ficou 77,5% positivo. Representando, dessa forma, maturidade suficiente para entrar em produção, porém dependendo de alguns ajustes para chegar a um cenário ideal. Um usuário comentou que a proposta do software está boa, porém existe a necessidade de sanar uma questão técnica na importação de processos. Este mesmo usuário utilizou uma versão mais recente do software BizAgi e encontrou problemas na importação de processos.

A utilização de recursos do computador, assim como a demora no carregamento das páginas do software agradou em 84,2% positivamente os usuários, representando, assim, uma leveza considerável para utilização do mesmo em um ambiente real.

A estrutura de navegação juntamente com o layout, termos utilizados e a facilidade na inserção de informações no software representaram 82,5% em pontos positivos quanto à utilização dos recursos da ferramenta. Um percentual alto suficiente para que a maioria dos usuários consiga interagir com o software.

A questão sobre fidelidade entre o processo criado no BizAgi e o processo importado no software teve apenas 2 respondentes, sendo que a avaliação alcançou apenas 40% de pontos positivos. Esse resultado está fortemente atrelado à dificuldade em importar processos de uma determinada versão do BizAgi, juntamente com a falta de suporte na importação de outros elementos da notação BPMN no ProMap. Outro ponto que pode não ter agradado ao usuário é o fluxograma gerado pelo sistema, algo que pode ser facilmente modificado, alterando o software de geração dos gráficos (Graphviz).

A segurança que cada usuário tem de apenas visualizar tarefas relacionadas ao seu papel na empresa gerou 88,3% de pontos positivos, porém teve um comentário importante. Um usuário escreveu que algumas empresas podem preferir deixar que todos os usuários visualizem todos os processos da empresa. Mediante esse comentário, sugere-se criar um outro link, contendo todos os processos importados no ProMap, pois o link Meus Processos, existente, hoje, no sistema, é muito útil quando um usuário deseja acessar rapidamente seus processos, sem haver a necessidade de visualizar processos dos quais não faz parte.

A qualidade do software perante erros e falhas obteve 75% de pontos positivos. Esse resultado é bom, visto que alguns usuários tiveram problemas para importar alguns processos.

#### 4.11 Trabalhos futuros

Como este projeto tem, por finalidade, organizar os processos de uma organização por meio de uma arquitetura, a seguir serão listadas algumas propostas para futuros trabalhos de conclusão que poderão facilmente se acoplar a este.

Criação de uma ferramenta de desenho de processos, sendo ela capaz de exportar o diagrama em formato XPDL. Contudo, tendo uma exportação padronizada em XPDL a mesma poderá considerar-se acoplada a este trabalho, que já possuirá importação XPDL. Este trabalho ocupará a etapa de Modelagem dentro de uma estrutura BPM Suite. Sugestão de nome para ferramenta: Prodesign.

Criação do motor de fluxo de trabalho que será responsável por rodar um processo automaticamente, enviando e-mails de alerta para os participantes do processo, quando estes tiverem uma tarefa pendente a ser entregue, além de criar os formulários para preenchimento das informações. Este trabalho ocupará as etapas de Simulação e Execução dentro de uma estrutura BPM Suite. Sugestão de nomes: Prosimulation e/ou Proexecution.

Criação de uma ferramenta para a otimização dos processos. Esta deverá medir o tempo levado em cada etapa do processo rodado no motor de fluxo, criando interfaces capazes de descobrir onde existem gargalos no processo. Sugestão de nome: Prooptimization.

Criação de uma ferramenta para avaliar impactos da alteração de um processo, podendo auxiliar o analista de processos na decisão de atualizar ou não o processo. Sugestão de nome: Proimpactor.

Com essas ferramentas desenvolvidas, tem-se uma estrutura completa de BPMS, desde a modelagem do processo até a otimização do mesmo.

## 5 CONCLUSÃO

Percebeu-se que a utilização da Informática, nas mais diversas áreas das organizações, está passando de um diferencial para um requisito. A partir deste fato, tem-se um aumento na demanda por tecnologias inovadoras, todas com o objetivo de reduzir custos, auxiliando os colaboradores em suas atividades.

Com uma perspectiva de aumento no investimento em gestão por processos por parte das empresas, entramos em um cenário em que cada vez mais, serão desenvolvidas ferramentas informatizadas para o auxílio das atividades realizadas no dia a dia. Nesse contexto, foi apresentado no presente trabalho uma ferramenta capaz de disponibilizar todas informações de seus processos, através de uma interface de serviços, independente da linguagem e plataforma utilizadas.

Com a estrutura orientada a serviços em funcionamento, consegue-se expandir as funcionalidades da ferramenta para outros softwares, tornando a mesma capaz de suprir demais necessidades que possam surgir futuramente. A interface de serviço foi desenvolvida neste trabalho com o intuito de aprimorar o software e na perspectiva de desenvolver uma estrutura BPMS completa, contendo as etapas de: desenho do processo, motor de fluxo e refinamento de processos.

Após a coleta dos testes com validações positivas praticamente em todos os quesitos, percebeu-se que a ferramenta poderá facilmente entrar em uso por empresas, necessitando apenas de pequenas adaptações conforme o ambiente ao qual será implantada.

No sentido de apoiar instituições com a organização e disseminação de seus processos de negócio, auxiliando os colaboradores em suas atividades e também buscando a minimização da perda de conhecimento com a alta rotatividade funcional, esse trabalho apresentou uma ferramenta para auxiliar organizações na gerência de seus processos de negócio.

## REFERÊNCIAS

- ABPMP. **Guia para o Gerenciamento de Processos de Negócio (BPM)**, 2009. Disponível em: [http://www.abpmp-br.org/CBOK/CBOK\\_v2.0\\_Portuguese\\_Edition\\_Thrid\\_Release\\_Look\\_Inside.pdf](http://www.abpmp-br.org/CBOK/CBOK_v2.0_Portuguese_Edition_Thrid_Release_Look_Inside.pdf). Acesso em: 15 de dezembro de 2012.
- BEHAR, A. P. **Etc: uma proposta de editor de texto coletivo na web**. 2003. 11p. Disponível em: <http://www.niee.ufrgs.br/eventos/RIBIE/2004/comunicacao/com363-373.pdf>. Acesso em: 24 out. 2011.
- BOOCH, G. **Object Oriented Design with Applications**. 3 ed. Benjamin/Cummings Pub. Co., 2007. 691p.
- BOSCARIOLI, C. **Uma reflexão sobre Banco de Dados Orientados a Objetos**. 2006. 12p. Disponível em: <http://conged.deinfo.uepg.br/artigo4.pdf>. Acesso em: 24 out. 2011.
- CHAO, X. X. S. **A Basing on Model-Driven Framework of Service-Oriented Software Production Line**. In: Second International Symposium on Computational Intelligence and Design, 2009.
- CHINOSI, M.; TROMBETTA, A. **Modeling and Validating BPMN Diagrams**. In: IEEE Conference on Commerce and Enterprise Computing, 2009.
- DATE, C. J. **An Introduction to Database Systems**. 6 ed. Wesley Pub Co: Ed Addison, 2000. 639p. ASIN: 020154329X
- DE SORDI, José Osvaldo. **Gestão por processos: uma abordagem da moderna administração**. São Paulo: Saraiva, 2009.
- INSADI. **O Mercado Nacional de Processos no seu estágio de gestão**. Business Process Day, SP, 2008. [http://www.businessprocessday.com.br/bpd2009.1/palestras\\_bpd2008/pdfs/Plenaria4\\_LAPie\\_monte.pdf](http://www.businessprocessday.com.br/bpd2009.1/palestras_bpd2008/pdfs/Plenaria4_LAPie_monte.pdf). Acessado em 28 de outubro de 2011.
- JONES, M. **Fundamentos do desenho orientado a objetos com UML**. Ed Makron Books, São Paulo, 2002.
- MICHAELIS. <http://michaelis.uol.com.br>. Acessado em 06 de outubro de 2011.
- O'BRIEN, L.; MERSON, P.; BASS, L. **Quality Attributes for Service-Oriented Architectures**. In: International Workshop on Systems Development in SOA Environments, IEEE Computer Society Washington, DC, USA, 7 p., 2007. Disponível em: [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=4273291](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4273291). Acesso em: 13 out. 2011.

OMG. **Business Process Model and Notation (BPMN)**. 2009. <http://www.omg.org/spec/BPMN/1.2/PDF/> . Acessado em 06 de outubro de 2011.

PAIM, Rafael; CARDOSO, Vinícius; CAULLIRAUX, Heitor. **Gestão de processos: pensar, agir e aprender**. São Paulo: Bookman, 2009.

PIRES, Mônica. Iniciativas BPM, [ideiasbpm.blogspot.com](http://ideiasbpm.blogspot.com). Acessado em 11 de dezembro de 2011.

SANTOS, R. **Introdução à programação orientada a objetos usando JAVA**. 1 ed. Rio de Janeiro: Ed. Campus, 2003. 320 p. ISBN: 85-352-1206-X.

SANTOS, Rildo. **Mapeamento e Modelagem de Processos de Negócio com BPMN**, <http://www.slideshare.net/Ridlo/mapeamento-e-modelagem-de-processos-de-negcio-com-bpmn> , 2009A. Acessado em 06 de outubro de 2011.

SANTOS, Rildo. **Gestão por Processo**, <http://www.slideshare.net/Ridlo/gesto-por-processo> , 2009B. Acessado em 27 de outubro de 2011.

SANTOS, Rildo. **Notação BPMN v.1.2**, <http://www.slideshare.net/Ridlo/notao-bpmn-v-12-3754516> , 2010. Acessado em 12 de outubro de 2011.

SANTOS, M. S. **UML Unified Modeling Language**, Faculdade de Tecnologia de Pernambuco, Disponível em [http://www.leonardomelosantos.com.br/arquivos/Introducao\\_ao\\_UML.pdf](http://www.leonardomelosantos.com.br/arquivos/Introducao_ao_UML.pdf), Acessado em 18 de novembro de 2011.

SEBESTA, W. R. **Conceitos de linguagens de programação**. 4 ed. PortoAlegre: Ed. Bookman, 2000. 624p.

TONSIG, L. S. **Projeto Orientado a Objetos e U.M.L.**, PUCCamp, Disponível em [http://www.micropic.com.br/noronha/Informatica/PD/AL/projeto\\_orientado\\_a\\_objeto\\_uml.pdf](http://www.micropic.com.br/noronha/Informatica/PD/AL/projeto_orientado_a_objeto_uml.pdf), Acessado em 18 de novembro de 2011.

W3C, **Web Services Architecture**, 2002. Disponível em <http://www.w3.org/TR/2002/WD-ws-arch-20021114/>, Acessado em 11 de dezembro de 2011

WfMC. **XML Process Definition Language**. 2008. Disponível em [http://www.wfmc.org/index.php?option=com\\_docman&task=doc\\_download&Itemid=72&gid=132](http://www.wfmc.org/index.php?option=com_docman&task=doc_download&Itemid=72&gid=132), Acessado em 13 de dezembro de 2011.



## APÊNDICES

APÊNDICE A – Manual de usuário entregue às pessoas que testaram o sistema.

### MANUAL PROMAP

O PROMAP tem, como objetivo, auxiliar os colaboradores de uma empresa a conhecerem melhor seus processos.

Após utilização do sistema, favor responder ao seguinte questionário online:

<https://docs.google.com/spreadsheet/viewform?formkey=dEZrd0JXQXJ3LWkya1BxUHVZUmMxQmc6MQ>

Acesse o sistema: [www.amadeuw.net/promap/](http://www.amadeuw.net/promap/)

Faça o login no sistema, utilizando o e-mail e a senha que foram cadastrados.

O sistema possui dois tipos de usuários:

- O Analista de Processos, responsável por cadastrar os processos dentro do sistema.
- O Colaborador, que acessará o sistema para visualizar seus processos e sanar dúvidas relativas aos mesmos. Esse usuário também poderá incluir comentários nas atividades de seus processos.

#### Casos de Uso:

No final deste documento, está anexa a documentação destes casos de uso.

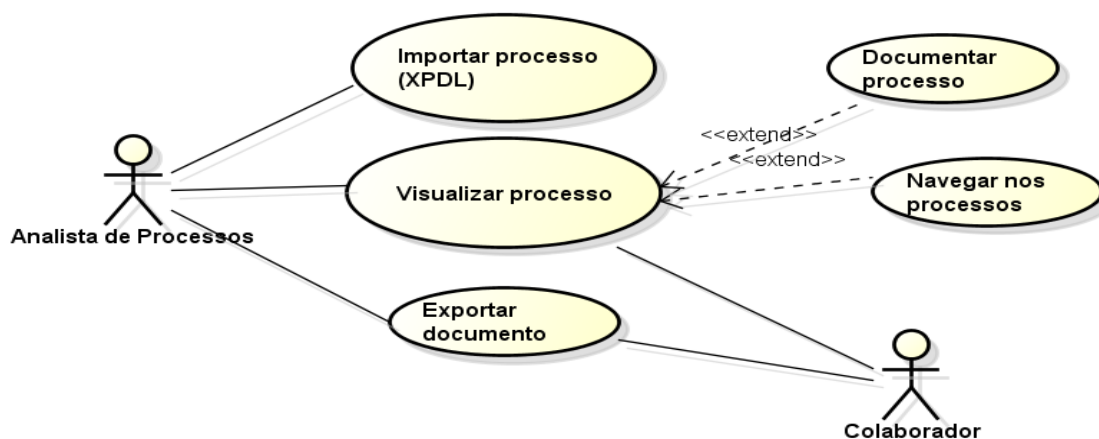
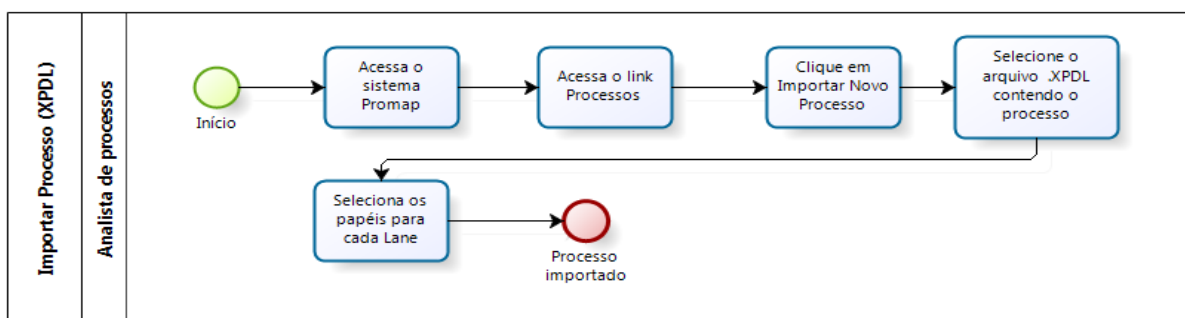


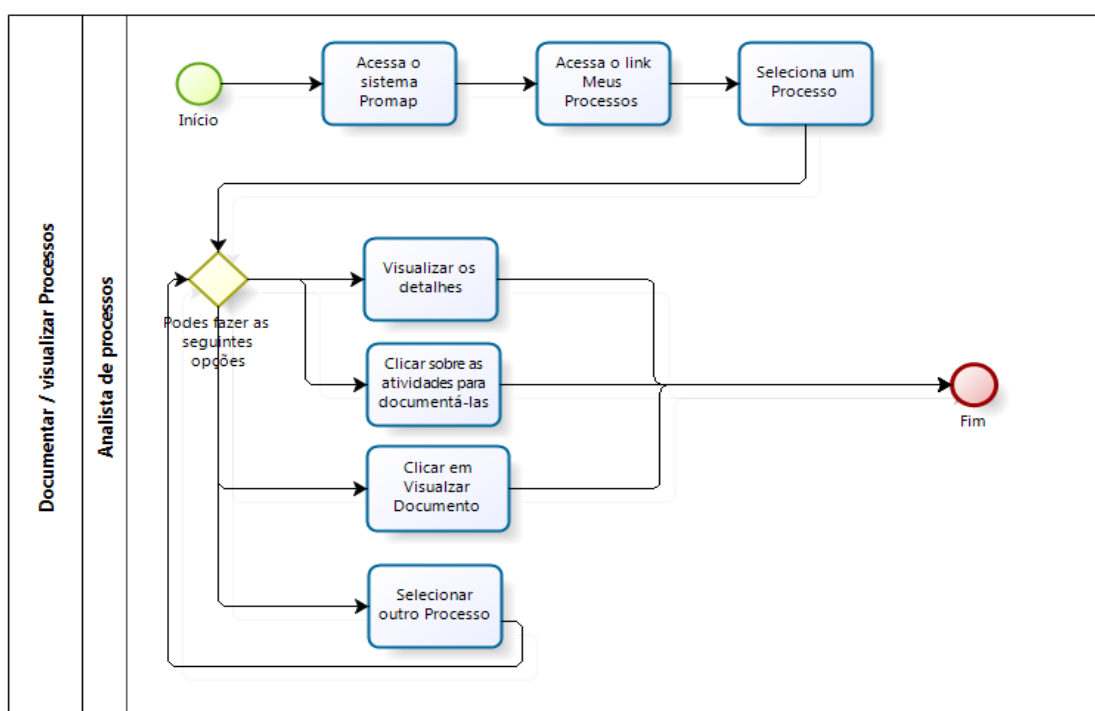
Imagem 1: Casos de Uso

## Guia de utilização do sistema:

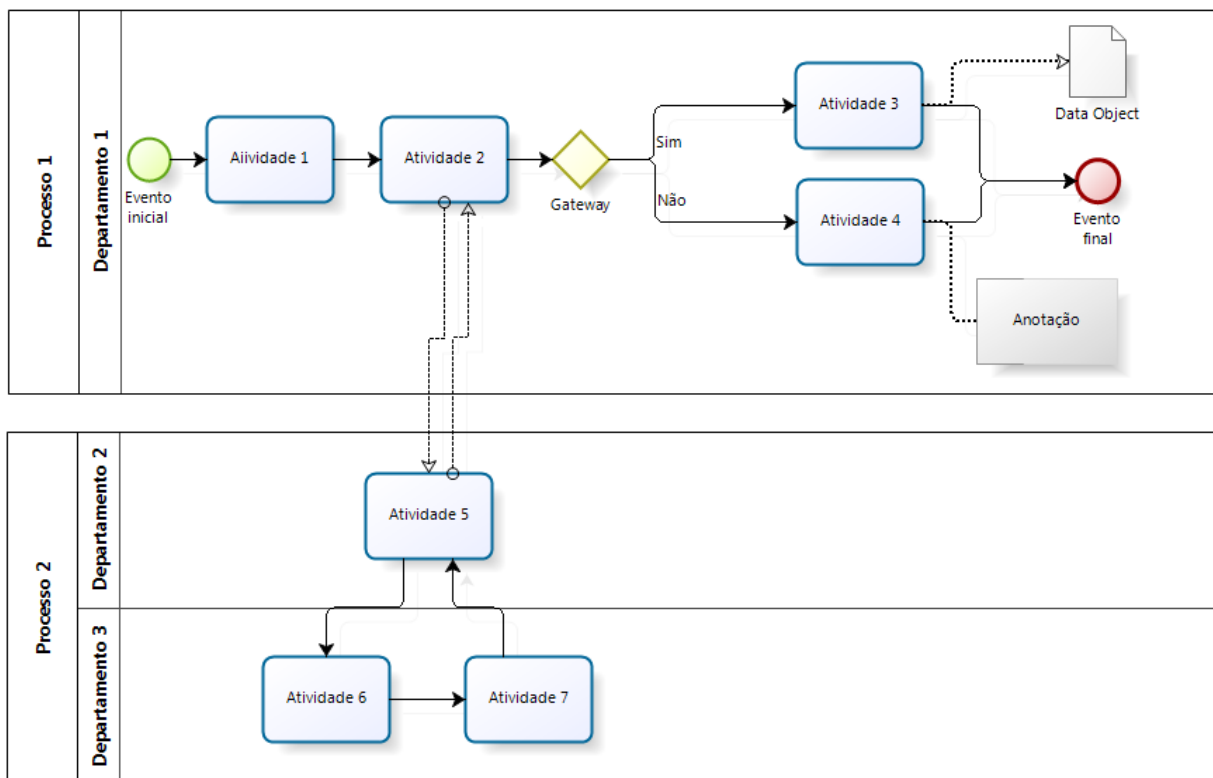
### Importar Processo (XPDL)



### Documentar / visualizar processos



## Todos elementos utilizáveis para importação



Qualquer dúvida, comentário ou sugestão pode ser encaminhada via e-mail para:

[amadeuw@gmail.com](mailto:amadeuw@gmail.com)

APÊNDICE B – Questionário disponibilizado às pessoas que testaram o sistema.

### Questionário sobre utilização do ProMap

Favor responder apenas uma vez o formulário. Para entrar no sistema acesse o endereço:

[www.amadeuw.net/promap](http://www.amadeuw.net/promap)

\* Obrigatório

**Você considera o software adequado para a gerência de processos? \***

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Você considera o modelo de processo importado na ferramenta fiel ao modelo criado no software bizagi?**

*Responda apenas se você fez importação de um processo novo.*

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Como você considera a segurança no acesso à informação (papéis e processos)? \***

*Levando em consideração que cada usuário pode ver apenas os processos referentes ao seu papel na empresa.*

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**A inserção de informações nas atividades de um processo são fáceis de utilizar? \***

*Seguindo o seguinte roteiro: 1. Clicando em Meus Processos; 2. Selecionando um processo; 3. Clicando sobre uma das atividades.*

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Você consideraria o software maduro para utilização em um cenário real (em produção)?\***

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Como você analisa a qualidade do software perante ocorrência de erros, falhas? \***

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Você considera os termos utilizados no software fáceis de compreender? \***

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Você considera a estrutura de navegação do software intuitiva? \***

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Você considera o layout geral da aplicação atraente? \***

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Você considera aceitável o comportamento do software em relação ao tempo de resposta? \***

*Velocidade com que as informações são expostas na tela.*

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Como você classifica o software em relação à utilização de recursos? \***

*Em relação ao software ser muito pesado para a utilização em browser.*

1 2 3 4 5

Ruim ☐ ☐ ☐ ☐ ☐ Excelente

**Observações**

## APÊNDICE C – Tela inicial do sistema ProMap.

