

FACULDADE DE COMPUTAÇÃO - FACOM  
UNIVERSIDADE FEDERAL DO MATO GROSSO DO SUL – UFMS  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
CIÊNCIA DA COMPUTAÇÃO

**Márcio Oshiro**

Estabelecimento de uma arquitetura de  
referência orientada a serviços para  
repositórios de linhas de processos de negócio

CAMPO GRANDE-MS

SETEMBRO/2014

**MÁRCIO OSSHIRO**

**Estabelecimento de uma arquitetura de  
referência orientada a serviços para  
repositórios de linhas de processos de negócio**

Dissertação apresentada ao Programa de Mestrado *Stricto Sensu* em Ciência da Computação da Faculdade de Computação, mantido pela Universidade Federal do Mato Grosso do Sul, como requisito para a obtenção do título de Mestre em Ciência da Computação (Área de Concentração: Engenharia de Software).

**Orientadora:**

**Prof<sup>a</sup>. Dr<sup>a</sup>. Maria Istela Cagnin Machado**

CAMPO GRANDE-MS

SETEMBRO/2014

# Agradecimentos

Eu quero agradecer a todos pela contribuição que recebi durante essa caminhada realizada neste curso.

Em especial, à minha esposa Ariane Osshiro, pela ajuda, compreensão, dedicação, e que sem o seu apoio não poderia chegar até aqui.

Ao meu filho Pedro Osshiro, que é uma inspiração e fonte de alegria.

Aos meus pais, Mário e Tereza, que sempre me incentivaram nos estudos e são exemplos de vida para mim.

Aos meus irmãos, Marcelo, Edson, Eliane e Alexandra, pela compreensão e palavras de incentivo.

A todos os membros da família Lubas Sales, da qual já me sinto parte.

À minha orientadora, professora doutora Maria Istela Cagnin Machado, pelo norteamento na condução da realização deste trabalho e pela amizade e ensinamentos transmitidos.

Aos amigos do IFMS, que torceram por mim.

Aos amigos da UFMS, por permitir caminharmos juntos.

Aos professores da UFMS, pelos ensinamentos sempre valiosos.

# Resumo

Linha de Produto foi introduzida na indústria de software com o objetivo de reduzir custos, diminuir o tempo de desenvolvimento e aumentar a qualidade do produto, sendo denominada por Linha de Produto de Software (LPS). Para apoiar o gerenciamento dos artefatos de uma LPS e para facilitar o reuso dos mesmos durante a instanciação da linha, é importante o uso de repositórios de ativos de software. Diante dos benefícios obtidos por meio da LPS, essa abordagem passou a ser aplicada no contexto de modelagem de processos de negócio, sendo denominada por Linha de Processos de Negócio (LPN). Como ocorre em LPS, também é importante o apoio computacional de repositórios no contexto de LPN. Porém, não foi encontrado na literatura um repositório específico de LPN. Devido à essa lacuna, observa-se a importância do estabelecimento de uma arquitetura de referência de repositórios de LPN para apoiar a construção desse tipo de repositório pelos interessados, colaborando para o avanço do estado da prática na área. Sob essa perspectiva, o objetivo deste trabalho de mestrado é definir uma arquitetura de referência de repositórios de LPN baseada em serviços, denominada Cambuci-LPN. Uma vez que repositórios de LPN podem ser utilizados por diversas ferramentas computacionais, como ferramentas de modelagem, de simulação de modelos e de controle de versão, justifica-se a escolha de uma abordagem orientada a serviços devido ao baixo acoplamento e à interoperabilidade que esse tipo de abordagem provê. A arquitetura de referência proposta é obtida a partir da definição e subsequente especialização de uma arquitetura de referência de repositórios de ativos de software denominada Cambuci, utilizando o processo de definição de arquiteturas de referência ProSA-RA. A arquitetura de referência Cambuci-LPN é avaliada com o apoio de um *checklist* que leva em consideração diferentes tipos de questões, como: visão geral, pontos de vista, visões e módulos, atributos de qualidade, *stakeholders* e seus interesses, e entendimento das decisões arquiteturais. Os resultados da avaliação conduzida indicam que a arquitetura de referência Cambuci-LPN é considerada satisfatória e pode apoiar a construção de um repositório de LPN que contemple, total ou parcialmente, os requisitos funcionais e arquiteturais nela contidos. Adicionalmente, observou-se com a avaliação algumas deficiências na Cambuci-LPN, como não explicitação dos *stakeholders* e de seus interesses e ausência de registros das decisões tomadas durante a definição da arquitetura de referência. Para dirimir tais deficiências, são

propostas neste trabalho sugestões de melhorias para que sejam incorporadas futuramente na Cambuci-LPN.

**Palavras-chave:** Arquitetura de Referência, Linha de Produto de Software, Modelagem de Processo de Negócio, Linha de Processo de Negócio, Repositório.

# Abstract

The Product Line was introduced in the software industry with the aim to reduce costs, decrease development time and increase the quality of the product, being named Software Product Line (SPL). To support the management of the artifacts of an SPL and to facilitate the reuse of the same during the instantiation of the line, it is important to use repositories of software assets. Given the benefits obtained by SPL, this approach has been applied in the context of business process modeling, being called Business Processes Line (BPL). As with SPL, it is also important to the computational support of repositories in the context of BPL. However, a specific BPL repository was not found in the literature. Because of this gap, there is the importance of establishing a reference architecture of BPL repositories to support the construction of such repository by stakeholders collaborating to advance the state of practice in the area. From this perspective, the goal of this master thesis is to define reference architecture of BPL repository based on services, called Cambuci-BPL. Once BPL repositories can be used by several computational tools, as modeling, model simulation and version control tools, it is justifiable to choose a service-oriented approach, mainly due to the low coupling and interoperability that such approach provides. The proposed reference architecture is obtained from the definition and subsequent specialization of the reference architecture of software assets repository called Cambuci, using the ProSA-RA process for defining reference architectures. The Cambuci-BPL reference architecture is evaluated with the help of a checklist that takes into account different types of questions such as: overview, viewpoints, views and modules, quality attributes, stakeholders and their interests, understanding of architectural decisions. The results of the evaluation conducted indicate that the Cambuci-BPL reference architecture is considered satisfactory and can support the construction of a BPL repository involving, in whole or in part, the functional and architectural requirements contained in the reference architecture. Additionally, it is also observed some deficiencies in the Cambuci-BPL evaluation, as no definition of stakeholders and their interests, and the absence of records of decisions made during the definition of the reference architecture. To address such deficiencies, suggestions for improvements are proposed in this work to be incorporated in the Cambuci-BPL later.

**Keywords:** *Reference Architecture, Software Product Line, Business Process Modeling, Business Process Line, Repository.*

# Lista de Figuras

Figura 2.1 - Nível de abstração e número de detalhes de arquiteturas de referência (adaptado de Muller, 2012).....	8
Figura 2.2 - Visões de arquitetura de referência 4+1 (adaptado de Reed, 2002) .....	9
Figura 2.3 - Ciclo para o desenvolvimento de arquiteturas de referência (adaptado de Cloutier, 2010).....	10
Figura 2.4 - Passos do ProSA-RA (adaptado de Nakagawa, 2014) .....	12
Figura 2.5 - Visão geral da RefTEST-SOA (OLIVEIRA, 2011) .....	16
Figura 2.6 - Visão de módulo da RefTEST-SOA (OLIVEIRA, 2011).....	17
Figura 2.7 - Visão em tempo de execução da RefTEST-SOA (OLIVEIRA, 2011).....	19
Figura 2.8 - Visão de implantação da RefTEST-SOA (OLIVEIRA, 2011).....	20
Figura 2.9 - Modelo conceitual da arquitetura de referência do trabalho 1 (MELAND <i>et al.</i> 2009).....	24
Figura 2.10 - Visão de componentes da arquitetura de referência do trabalho 1 (MELAND <i>et al.</i> , 2009) .....	25
Figura 2.11 - Modelo conceitual da arquitetura de referência do trabalho 3 (HONGMIN <i>et al.</i> , 2010).....	27
Figura 3.1 - Exemplo de modelagem de processos de negócio usando a notação BPMN 2.0.2. (adaptado de OMG, 2013) .....	36
Figura 3.2 - Arquitetura de referência de repositórios de MPN (adaptado de Yan <i>et al.</i> , 2012).....	43
Figura 3.3 - Modelo conceitual da ONTO-ResAsset (SILVA, 2013).....	46
Figura 4.1 - Processos da ELPS (adaptado de Pohl <i>et al.</i> , 2005) .....	54
Figura 4.2 - Metamodelo simplificado do modelo ortogonal de variabilidades (adaptado de Pohl <i>et al.</i> , 2005).....	56
Figura 4.3 - Ponto de variação "Pagamento" na notação vrBPMN (adaptado de Schnieders, 2006).....	59
Figura 4.4 - Arcabouço GLPN: gestão de linhas de processos de negócio (LANDRE, 2012) .....	61
Figura 4.5 - Principais artefatos do EDN: <i>template</i> de processos de negócio e modelo de variabilidade (LANDRE, 2012).....	64
Figura 5.1 – Concepção da Cambuci.....	67

Figura 5.2 - Extração de requisitos arquiteturais do contexto de serviços da Cambuci.....	73
Figura 5.3 - Visão geral da arquitetura Cambuci .....	81
Figura 5.4 - Visão de módulo da Cambuci.....	87
Figura 5.5 - Visão em tempo de execução da Cambuci .....	89
Figura 5.6 - Visão de implantação da Cambuci .....	91
Figura 6.1 - Especialização arquitetural da Cambuci.....	93
Figura 6.2 – Modelo conceitual dos elementos do BPMN 2.0.2 utilizados neste trabalho (adaptado de OMG, 2013) .....	95
Figura 6.3 - Elementos que compõem uma LPN .....	97
Figura 6.4 - Modelo conceitual do modelo de variabilidade.....	98
Figura 6.5 - Modelo conceitual do Template de Processos de Negócio.....	100
Figura 6.6 - Visão geral da arquitetura Cambuci-LPN.....	110
Figura 6.7 - Modelo conceitual da LPN .....	115
Figura 6.8 - Visão de módulo da Cambuci-LPN .....	117
Figura 6.9 - Visão em tempo de execução da Cambuci-LPN .....	120
Figura 6.10 - Visão de implantação da Cambuci-LPN.....	121
Figura 6.11 - Empacotamento de ativo com formato zip pela RAS (OMG, 2005).....	122
Figura 6.12 - Instanciação arquitetural da Cambuci-LPN.....	127
Figura A.1 – Ferramenta computacional fictícia de apoio a abordagens LPN – operação Salvar LPN.....	139
Figura A.2 - Ferramenta computacional fictícia de apoio a abordagens LPN – operação Buscar LPN.....	140
Figura A.3 - Ferramenta computacional fictícia de apoio a abordagens LPN – resultado da operação Buscar LPN .....	140
Figura A.4 - Ferramenta computacional fictícia de apoio a abordagens LPN – operação Editar LPN .....	141
Figura A.5 - Arcabouço de apoio computacional a abordagem GLPN – operação instanciar LPN.....	142

# Lista de Quadros

Quadro 2.1 - Artigos selecionados no mapeamento sistemático .....	20
Quadro 2.2 - Tipos de ativo dos repositórios.....	22
Quadro 2.3 - Visões arquiteturais dos trabalhos selecionados .....	23
Quadro 2.4 - Resumo geral das questões de pesquisa e respectivas respostas.....	28
Quadro 2.5 - Comparativo entre processos de desenvolvimento de arquiteturas de referência .....	29
Quadro 3.1 - Objetos de fluxo da notação BPMN 2.0.2 (adaptado de OMG, 2013) .....	32
Quadro 3.2 - Elementos de dados dados da notação BPMN 2.0.2 (adaptado de OMG, 2013) .....	33
Quadro 3.3 - Objetos de conexão da notação BPMN 2.0.2 (adaptado de OMG, 2013).....	34
Quadro 3.4 - Elementos de agrupamento da notação BPMN 2.0.2 (adaptado de OMG, 2013).....	35
Quadro 3.5 - Artefatos da notação BPMN 2.0.2 (adaptado de OMG, 2013) .....	36
Quadro 3.6 - Modelo de gerenciamento de repositórios de MPN (adaptado de Yan <i>et al.</i> , 2012).....	42
Quadro 3.7 - Comparação entre ONTO-ResAsset e arcabouço de repositórios de MPN (Parte 1) .....	47
Quadro 3.8 - Comparação entre ONTO-ResAsset e arcabouço de repositórios de MPN (Parte 2) .....	49
Quadro 4.1 - Adaptação da notação vrBPMN para o GLPN (Landre, 2012).....	63
Quadro 5.1 - Funcionalidades identificadas a partir da ONTO-ResAsset.....	69
Quadro 5.2 - Estudos sobre abordagens de repositórios de ativos de software.....	70
Quadro 5.3 - Funcionalidades identificadas nas abordagens de repositórios de ativos de software .....	71
Quadro 5.4 - Requisitos arquiteturais do contexto de serviço para ferramentas de engenharia de software (adaptado de Oliveira, 2011).....	73
Quadro 5.5 - Requisitos arquiteturais para repositórios de ativos de software .....	75
Quadro 5.6 - Requisitos arquiteturais do contexto de serviços para repositórios de ativos de software .....	77
Quadro 5.7 - Requisitos arquiteturais da camada de aplicação .....	82
Quadro 5.8 - Requisitos arquiteturais da camada de persistência .....	82

Quadro 5.9 - Requisitos arquiteturais da camada de orquestração.....	83
Quadro 5.10 - Requisitos arquiteturais da camada de intermediação .....	83
Quadro 5.11 - Requisitos arquiteturais da camada de apresentação .....	84
Quadro 5.12 - Requisitos arquiteturais da camada de qualidade de serviço .....	84
Quadro 5.13 – Pacotes e sub-pacotes da visão de módulo da Cambuci.....	85
Quadro 5.14 - Serviços de cada pacote da visão de módulo da Cambuci .....	86
Quadro 6.1 - Trabalhos selecionados sobre abordagens de LPN .....	96
Quadro 6.2 - Conceitos de LPN do grupo 2 .....	96
Quadro 6.3 Funcionalidades identificadas nas abordagens LPN .....	101
Quadro 6.4 - Requisitos arquiteturais de repositórios de LPN.....	104
Quadro 6.5 - Requisitos arquiteturais do contexto de serviços para repositórios de LPN	106
Quadro 6.6 - Requisitos arquiteturais da camada de aplicação .....	111
Quadro 6.7 - Requisitos arquiteturais da camada de persistência .....	112
Quadro 6.8 - Requisitos arquiteturais da camada de orquestração.....	112
Quadro 6.9 - Requisitos arquiteturais da camada de intermediação .....	113
Quadro 6.10 - Requisitos arquiteturais da camada de apresentação .....	113
Quadro 6.11 - Requisitos arquiteturais da camada de qualidade de serviço .....	113
Quadro 6.12 - Módulos da visão de módulos para o Cambuci-LPN.....	116
Quadro 6.13 - Distribuição dos serviços em cada pacote e sub-pacote da Cambuci-LPN	118
Quadro 6.14 - Assuntos abordados por cada passo do <i>checklist</i> (Adaptado de Santos, 2013)	123
.....	123
Quadro 6.15 - Exemplos de questão do <i>checklist</i> (Adaptado de Santos, 2013) .....	123
Quadro 6.16 - Justificativas e soluções para as respostas parcialmente atendidas e não atendidas das questões (passo 1) .....	125
Quadro 6.17 - Justificativas e soluções para as respostas parcialmente atendidas e não atendidas das questões (passo 2) .....	126

# Sumário

<b>Capítulo 1 - Introdução .....</b>	<b>1</b>
1.1 Contextualização .....	1
1.2 Motivação e justificativa .....	2
1.3 Objetivos do trabalho .....	3
1.4 Organização do texto.....	4
<b>Capítulo 2 - Arquitetura de referência .....</b>	<b>6</b>
2.1 Considerações iniciais .....	6
2.2 Conceitos e definições.....	6
2.3 Representações .....	8
2.4 Processos de desenvolvimento de arquiteturas de referência .....	9
2.5 Arquitetura de referência RefTEST-SOA .....	12
2.6 Arquiteturas de referência para repositórios de ativos de software .....	20
2.7 Considerações finais.....	28
<b>Capítulo 3 - Modelagem de processos de negócio .....</b>	<b>30</b>
3.1 Considerações iniciais .....	30
3.2 Conceitos.....	30
3.3 Estudos sobre modelagem de processos de negócio.....	36
3.4 Um arcabouço para repositórios de modelos processos de negócio .....	37
3.5 ONTO-ResAsset: ontologia de especificação e gestão de ativos reutilizáveis .....	44
3.6 Estudo comparativo entre o arcabouço de repositórios de modelos de processos de negócio e a ONTO-ResAsset.....	45
3.7 Considerações finais.....	51
<b>Capítulo 4 - Linhas de processos de negócio .....</b>	<b>53</b>
4.1 Considerações iniciais .....	53
4.2 Conceitos e definições.....	53
4.3 Abordagens de linhas de processos de negócio .....	57
4.4 Considerações finais.....	65
<b>Capítulo 5 - Estabelecimento da Cambuci .....</b>	<b>66</b>
5.1 Considerações iniciais .....	66
5.2 Passo 1: Investigação de fontes de informação .....	67
5.3 Passo 2: Análise arquitetural .....	74
5.4 Passo 3: Síntese arquitetural.....	78
5.5 Passo 4: Avaliação da arquitetura de referência .....	91
5.6 Considerações finais.....	91

<b>Capítulo 6 - Estabelecimento da Cambuci-LPN .....</b>	<b>93</b>
6.1 Considerações iniciais .....	93
6.2 Passo 1: Investigação de fontes de informação .....	94
6.3 Passo 2 – Análise arquitetural .....	102
6.4 Passo 3 – Síntese arquitetural .....	107
6.5 Passo 4: Avaliação da arquitetura de referência .....	122
6.6 Considerações finais .....	126
<b>Capítulo 7 - Conclusão .....</b>	<b>128</b>
7.1 Considerações iniciais .....	128
7.2 Contribuições .....	128
7.3 Limitações .....	129
7.4 Trabalhos futuros.....	129
<b>Apêndice A – Exemplo de ferramenta computacional de apoio a abordagens LPN</b>	<b>139</b>

# Capítulo 1 - Introdução

## 1.1 Contextualização

Linha de produto de software (LPS) é uma abordagem de reutilização de software. Segundo Pohl *et al.* (2005), surgiu com a necessidade de redução dos custos de desenvolvimento, melhoria da qualidade, redução do tempo de colocação do produto no mercado, redução dos esforços de manutenção, entre outros. Para viabilizar o desenvolvimento baseado em reuso com o apoio de LPS é essencial o uso de uma plataforma computacional apoiada por um repositório de ativos, pois além de gerenciar os ativos da linha também possibilita a reutilização dos mesmos durante a instanciação da LPS, cujo resultado é um sistema de software.

Diante dos benefícios obtidos por meio da LPS, essa abordagem passou a ser aplicada no contexto de modelagem de processos de negócio (NUNES *et al.*, 2010; BOFFOLI *et al.*, 2012a; ROLLAND e NURCAN, 2010; GIMENES *et al.*, 2008), sendo denominada por linha de processos de negócio (LPN).

O principal motivo que culminou o surgimento de LPN foi a importância da modelagem de processos de negócio para as empresas porque (ALENCAR, 1999; ERIKSSON e PENKER, 2000): i) propicia a melhoria e a evolução do negócio; ii) favorece a gestão do conhecimento pois o conhecimento sobre a empresa torna-se disponível para todos os interessados; e iii) possibilita o desenvolvimento de sistemas de softwares mais adequados visto que, por explicitar o funcionamento do negócio, favorece a elicitação dos requisitos que devem ser contemplados por tais sistemas.

Similarmente à LPS, a LPN permite: redução no esforço e no custo de elaboração da modelagem de negócios visto que o negócio de uma empresa envolve centenas e até milhares de processos de negócio (YAN *et al.*, 2012); melhoria e aprimoramento do negócio e na qualidade da modelagem produzida, pois os modelos de processos de negócio reutilizados já foram validados previamente por especialistas do negócio (LADEIRA, 2008) e, possivelmente, por especialistas em melhoria de processos que detectam problemas, inconsistências e redundâncias nos processos de negócio mapeados no domínio da LPN (AALST *et al.*, 2003).

Além disso, como ocorre em LPS, é imprescindível o uso de repositórios para gerenciar os ativos de LPN. Vários repositórios de modelos de processos de negócio foram encontrados na literatura (ELHADAD *et al.*, 2008; WANG e WU 2011;

LU *et al.*, 2009; YAN *et al.* (2012)), porém não foram encontrados na literatura evidências do uso desse tipo de repositório no contexto de LPN.

Dada a relevância de repositórios para apoiar a reutilização de ativos, em qualquer nível de abstração, observa-se a importância do uso de arquitetura de referência (REED, 2002) para facilitar a criação e a manutenção de repositórios de ativos de software. Considerando que a reutilização sistemática de ativos é obtida se ferramentas computacionais puderem se comunicar com os repositórios de ativos e que a arquitetura orientada a serviços (*Service Oriented-Architecture* - SOA) tem como um dos princípios o baixo acoplamento (ERL, 2009), observa-se que é viável utilizar esse tipo de arquitetura na concepção de repositórios de ativos de software, como é o caso de repositórios de LPN, porque ela possibilita que ferramentas computacionais diversas utilizem os serviços disponíveis pelo repositório de uma forma mais rápida e segura.

Como no contexto de software, a construção de arquiteturas de referências também deve ser realizada de maneira sistemática, com o apoio de processos. Nakagawa (2014) propôs o processo ProSA-RA que apoia a definição de arquiteturas de referência, levando em consideração a validação da arquitetura de referência em construção, diferentemente de outro trabalho encontrado na literatura que apresenta um ciclo para o desenvolvimento de uma arquitetura de referência, mas não possui um detalhamento que permita identificar o processo a ser seguido. (CLOUTIER *et al.*, 2010). Porém, apesar da relevância de processos para a concepção de arquiteturas de referências, os estudos encontrados sobre arquiteturas de referência para repositórios de ativos de software não indicam o uso dos mesmos.

Diante do exposto, evidencia-se a relevância de pesquisas no contexto de arquitetura de referência, orientada a serviço, para linhas de processos de negócio.

## **1.2 Motivação e justificativa**

A partir dos resultados de um mapeamento sistemático sobre arquiteturas de referência de repositórios de ativos de software, conduzido pelo autor deste trabalho (OSSHIRO, 2013), observou-se que há poucos trabalhos na literatura sobre o tema. Dos trabalhos encontrados, uma arquitetura de referência é sobre ativos de segurança (MELAND *et al.*, 2009), a outra sobre modelos de processos de negócio (YAN *et al.*, 2012) e a outra não especifica o tipo de ativo, podendo o repositório ser utilizado para qualquer tipo de ativo de software (HONGMING *et al.*, 2010).

Em especial, o trabalho de Yan *et al.* (2012) trata de uma arquitetura de referência para repositórios de modelos de processos de negócio, concebida a partir de um estudo sobre vinte repositórios desse tipo, que identificou as características comumente encontradas nos repositórios analisados. Essa arquitetura de referência não possui detalhes e é apresentada em alto nível de abstração dificultando o seu entendimento.

Salienta-se que nenhuma das arquiteturas de referência encontradas foi definida com o apoio de um processo sistemático e que todas utilizam ou fazem referência ao modelo conceitual, que é uma forma de representação da arquitetura de referência, e que podem conter conceitos, associações entre conceitos e atributos de conceitos. Essas arquiteturas de referência encontradas não possuem as visões arquiteturais de tempo de execução e de implantação propostas pelo processo ProSA-RA.

Outra justificativa para o desenvolvimento deste trabalho refere-se à necessidade de criar um repositório para apoiar a abordagem para Gestão de LPN (GLPN) (LANDRE, 2012), a qual foi definida pelo grupo de pesquisa de Engenharia de Software da UFMS. Mais detalhes sobre essa abordagem são obtidos na Seção 4.3 do Capítulo 4.

Diante do exposto, observa-se a importância do estabelecimento de uma arquitetura de referência orientada a serviços de LPN com o apoio de um processo de definição desse tipo de arquitetura, visando obter uma arquitetura de referência com melhor qualidade do que se fosse construída sem o apoio de um processo.

### **1.3 Objetivos do trabalho**

O objetivo geral deste trabalho é fornecer uma arquitetura de referência orientada a serviços para repositórios de LPN, denominada Cambuci-LPN, para que interessados possam utilizá-la como base para o desenvolvimento de repositórios de LPN.

A palavra “cambuci” utilizada para compor o nome da arquitetura de referência proposta é de origem tupi-guarani e o seu significado é pote de água, o que remete ao significado de repositório.

Para alcançar o objetivo geral, a seguir são elencados os objetivos específicos deste trabalho:

- definir uma arquitetura de referência orientada a serviços de repositórios de ativos de software, denominada Cambuci, com o apoio do processo ProSA-RA e selecionar as visões arquiteturais mais adequadas para representar a arquitetura visando facilitar o seu entendimento. A arquitetura de referência Cambuci deve poder ser especializada para qualquer domínio de repositórios de ativos de software, como por exemplo, para arquitetura de referência de repositórios de requisitos de software, arquitetura de referência de repositórios de teste de software, entre outras;
- especializar a arquitetura de referência Cambuci, com o apoio do processo ProSA-RA, para conceber a arquitetura de referência orientada a serviços Cambuci-LPN; e
- avaliar a arquitetura de referência Cambuci-LPN visando observar as características de qualidade de arquiteturas de referência, indicadas em um *checklist* de avaliação proposto por Santos (2013), e também identificar e eliminar defeitos de omissão, ambiguidade, inconsistência e informações desconhecidas ou incorretas.

#### **1.4 Organização do texto**

A escrita do texto desta dissertação está organizada em mais sete capítulos.

No Capítulo 2 são apresentados os principais conceitos de arquitetura de referência, o ProSA-RA, que é um processo que apoia a definição de arquiteturas de referência e de arquiteturas de referência orientadas a serviços. Também são apresentados uma arquitetura de referência orientada a serviços para ferramentas de teste de software (RefTEST-SOA), e um estudo sobre arquiteturas de referências para repositórios de ativos de software.

No Capítulo 3 são apresentados os conceitos de modelagem de processos de negócio, alguns estudos atuais sobre modelagem de processos de negócio, repositórios de modelos de processos de negócio, devido ao apoio computacional por eles provido para o gerenciamento do reuso de modelos de processos de negócio. Também é apresentada neste capítulo uma ontologia de especificação e gestão de ativos

reutilizáveis de software, sendo que uma LPN também pode ser considerada um ativo reutilizável de software.

No Capítulo 4 são apresentados alguns trabalhos sobre abordagens de LPN e que contribuem para o estabelecimento da arquitetura de referência proposta, porque são uma das fontes de informação para a obtenção dos requisitos arquiteturais da arquitetura de referência Cambuci-LPN. Nesses estudos são analisadas as formas como representam a LPN, inclusive a sua variabilidade.

No Capítulo 5 é realizado o estabelecimento da arquitetura de referência orientada a serviços Cambuci com o apoio do processo ProSA-RA. Essa arquitetura de referência pode ser especializada para qualquer subdomínio do domínio de repositórios de ativos de software, como é o caso da LPN.

No Capítulo 6 é realizado o estabelecimento da arquitetura de referência Cambuci-LPN, a partir da especialização da Cambuci definida no Capítulo 5, com o apoio do processo ProSA-RA. Também é apresentada a avaliação da arquitetura Cambuci-LPN através de um *checklist*.

Por fim, no Capítulo 7, é apresentada a conclusão deste trabalho, contemplando as suas contribuições para o avanço da área de LPN, as limitações que este trabalho possui e indicações de trabalhos futuros que podem ser desenvolvidos a partir do presente trabalho.

## Capítulo 2 - Arquitetura de referência

### 2.1 Considerações iniciais

Neste capítulo são apresentados os principais conceitos relacionados à arquitetura de referência, necessários para o entendimento da proposta desta dissertação de mestrado. Na Seção 2.2 são apresentados conceitos e definições sobre o tema. Na Seção 2.3 são apresentadas algumas visões que podem ser utilizadas para representar uma arquitetura de referência.

Na Seção 2.4 são apresentados dois processos que apoiam o desenvolvimento de arquiteturas de referência, sendo um deles utilizado neste trabalho para o estabelecimento da arquitetura de referência orientada a serviços Cambuci-LPN. Na Seção 2.5 é apresentada a arquitetura de referência RefTEST-SOA, que é uma arquitetura de referência para ferramentas de teste de software baseada em SOA (*Service Oriented Architecture*) (OLIVEIRA, 2011), pois é baseada na abordagem orientada a serviço que é de interesse deste trabalho.

Na Seção 2.6 são apresentados trabalhos sobre arquiteturas de referência para repositórios de ativos de software obtidos a partir de um mapeamento sistemático realizado por Osshiro (2013). Por fim, na Seção 2.7, são apresentadas as considerações finais do capítulo e como esses estudos serviram de base para a definição da arquitetura de referência orientada a serviços para repositórios de LPN Cambuci-LPN.

### 2.2 Conceitos e definições

Arquitetura de software é um dos mais importantes artefatos desenvolvidos e utilizados no ciclo de vida do desenvolvimento de software. Ela é um meio apropriado de especificar um sistema, entendê-lo, e comunicar seus aspectos estáticos e dinâmicos de mais alto nível para os vários interessados (DEBAUD, 2008).

Existem várias definições para arquitetura de sistemas ou de software, variando conforme o nível de abstração que se deseja manter no documento arquitetural.

Para Bass (2012), arquitetura de software é a estrutura ou estruturas do sistema, que compreende os componentes de software, as propriedades desses componentes visíveis externamente, bem como os relacionamentos entre eles.

Em Booch (2005), uma arquitetura é um conjunto de decisões significativas sobre:

- a organização de um sistema de software;
- a seleção dos elementos estruturais e as suas interfaces nas quais o sistema é composto;
- seu comportamento, tal como especificado nas colaborações entre esses elementos;
- a composição desses elementos estruturais e comportamentais em subsistemas progressivamente maiores; e
- o estilo arquitetônico que orienta esta organização.

Pode-se observar a partir da definição supracitada a necessidade de dividir o sistema em subsistemas e em definir o estilo arquitetônico da arquitetura.

Na tentativa de padronização de quais elementos uma descrição arquitetural deve conter, foi definida a ISO/IEC/IEEE 42010 (2011), que descreve que uma arquitetura de sistemas envolve os conceitos fundamentais ou as propriedades de um sistema em seu ambiente, incorporados em seus elementos, relacionamentos e nos princípios de sua concepção e evolução. A preocupação com a evolução da arquitetura deve ser destacada pois significa que uma arquitetura tem que evoluir conforme novos requisitos sejam identificados.

No contexto de Linha de Produto de Software (LPS) (Seção 4.2), a arquitetura de software tem uma importância ainda maior. As instâncias da LPS compartilham uma arquitetura comum, o que assegura sua integridade conceitual (DEBAUD, 1998).

Segundo Reed (2002), arquitetura de referência é, na essência, um padrão arquitetural pré-definido ou um conjunto de padrões, possivelmente parcial ou completamente instanciado, projetado e testado para o uso de um negócio particular e contexto técnico, junto com suporte de artefatos para habilitar o seu uso.

Conclui-se a partir das afirmações de Debaud (1998) e Reed (2002) que a partir de uma arquitetura de referência utilizada como base, pode-se desenvolver um sistema reutilizando e estendendo as definições contidas nessa arquitetura de referência.

No estudo de Cloutier *et al.*, (2010) é examinado um conjunto de arquiteturas de sistemas existentes para chegar a uma definição mais recente sobre arquitetura de referência. Por conseguinte, esses autores definem arquitetura de

referência como uma arquitetura que provê um guia para as organizações melhorarem ou criarem novas arquiteturas de software tomando como base a missão, a visão, a estratégia e as necessidades de seus clientes. Tais arquiteturas de referência devem levar em consideração arquiteturas técnicas, de negócios e de contexto.

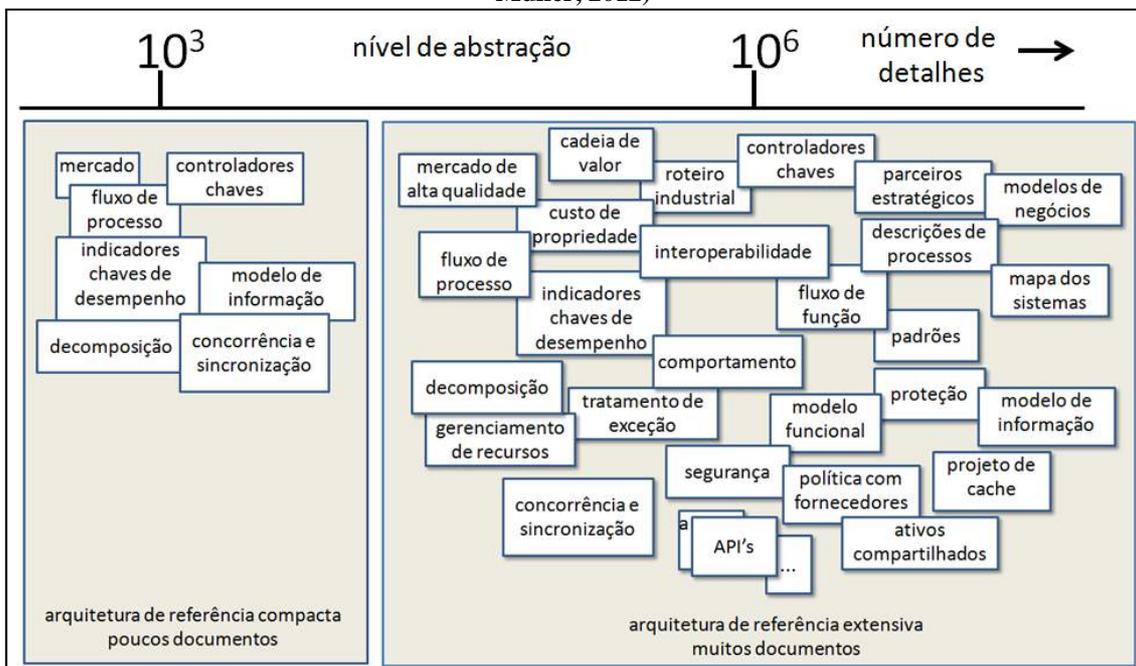
### 2.3 Representações

Conforme o nível de abstração da arquitetura de referência, a sua representação, denominada visão por diversos autores, pode conter vários elementos, conforme representados na Figura 2.1.

Nessa figura, foi observado que quanto maior o nível de abstração da arquitetura de referência, mais fácil será o seu gerenciamento. Por outro lado, quanto mais detalhada a arquitetura de referência é, sendo considerado um guia mais eficiente e completo, mais difícil será a sua manutenção.

Os modelos utilizados para documentar uma arquitetura de referência variam conforme o nível de abstração e conforme o domínio abordado pela arquitetura de referência (MULLER, 2012).

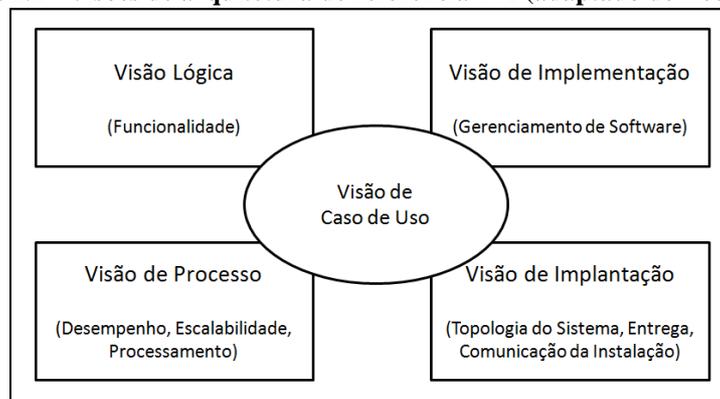
**Figura 2.1 - Nível de abstração e número de detalhes de arquiteturas de referência (adaptado de Muller, 2012)**



Reed (2002) propõe uma representação para arquitetura de referência possuindo 4+1 visões contidas no arcabouço de processo de desenvolvimento de software *Rational Unified Process* (RUP) (IBM Corporation, 2001), conforme

apresentada na Figura 2.2. De acordo com Kruchten (1995), as visões de Caso de Uso e de Lógica são necessárias em todos os projetos porque na camada de Caso de Uso é onde se encontram detalhados os cenários do domínio, e na camada Lógica é onde são projetados os modelos de dados, ou modelos de objetivo no caso de utilizar o paradigma orientado a objetos. As demais visões devem ser utilizadas somente quando requisitadas, por exemplo, a Visão de Processo é utilizada quando há muitas *threads* de controle; a Visão de Implementação é utilizada quando se deseja descrever a organização estática do software e seu ambiente de desenvolvimento, e a Visão de Implantação é utilizada quando se deseja representar o(s) mapeamento(s) do software para o hardware e refletir seus aspectos distribuídos.

**Figura 2.2 - Visões de arquitetura de referência 4+1 (adaptado de Reed, 2002)**



Nakagawa (2006) utiliza três das cinco visões arquiteturais definidas por Merson (2005) para documentar arquiteturas de referência: visão estrutural ou de módulos, visão comportamental ou em tempo de execução, e a visão física ou de implantação. Além dessas, há também a visão de dados e a visão de implementação que não foram utilizadas por Nakagawa devido ao alto nível de abstração nelas contido, no entanto podem ser usadas na representação da arquitetura de referência Cambuci-LPN. Detalhes sobre cada uma dessas visões são descritas na Seção 2.5 deste capítulo.

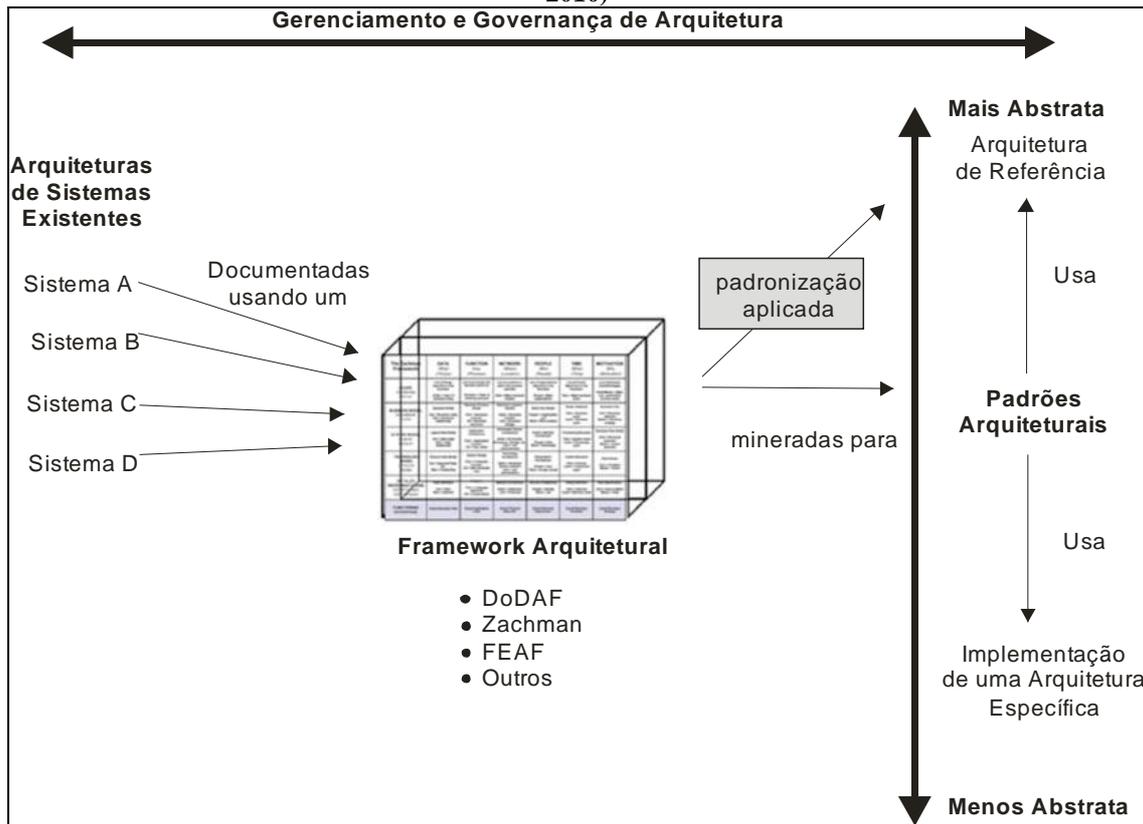
## **2.4 Processos de desenvolvimento de arquiteturas de referência**

O processo de desenvolvimento de software é de extrema relevância no tocante à sistematização das atividades envolvidas com a construção do software, considerando desde a concepção inicial até a fase de manutenção do software. Esse processo define as diferentes etapas ou estágios do ciclo de vida de um software e estabelece os princípios e as diretrizes de como os diferentes estágios devem ocorrer

(NAKAGAWA, 2006). Na definição de uma arquitetura de referência também é importante o uso de um processo sistemático e bem definido para que se alcance um produto de qualidade.

Cloutier (2010) fornece um ciclo para o desenvolvimento de uma arquitetura de referência, conforme ilustrado na Figura 2.3.

**Figura 2.3 - Ciclo para o desenvolvimento de arquiteturas de referência (adaptado de Cloutier, 2010)**



O passo inicial do ciclo de desenvolvimento definido por Cloutier (2010) consiste em obter vários sistemas de software de um mesmo domínio e cujas arquiteturas estejam documentadas utilizando algum *framework* arquitetural, por exemplo, DoDAF (*Department Of Defense Architecture Framework*) (U.S. DEPARTMENT OF DEFENSE, 2010), Zachman (ZACHMAN, 2013) e FEAF (*Federal Enterprise Architecture Framework*) (U.S. FEDERAL GOVERNMENT, 2012). Esse tipo de *framework* arquitetural captura o conhecimento sobre metodologias para criar e documentar arquiteturas (CLOTIER, 2010). O conhecimento sobre tais metodologias varia de visões (de propósito, *stakeholders*, interesses), representações (por exemplo, diagramas de classes e atividades), ferramentas e métodos (decompostos em tarefas, ordenações e diretrizes).

A partir das arquiteturas documentadas dos sistemas de software e aplicando-se padrões arquiteturais, obtêm-se a arquitetura de referência do domínio estudado em um alto nível de abstração. Alternativamente, em níveis mais baixos de abstração, também é possível identificar novos padrões arquiteturais ou implementar uma arquitetura específica a partir das arquiteturas documentadas previamente.

Observa-se que no ciclo proposto por Cloutier (2010), a arquitetura de referência deve evoluir continuamente e deve ser ativamente mantida e refatorada (representado na Figura 2.3 pelo gerenciamento e governança da arquitetura). Por outro lado, esse ciclo possui apenas um tipo de fonte de informação para a concepção da arquitetura de referência.

Outro processo encontrado na literatura é o ProSA-RA, proposto por Nakagawa (2014), que é um processo que sistematiza o desenvolvimento, projeto e avaliação de arquiteturas de referências orientadas a aspectos desenvolvidas para o domínio de Engenharia de Software. Este processo, ilustrado na Figura 2.4, é composto por quatro passos, que são detalhados a seguir.

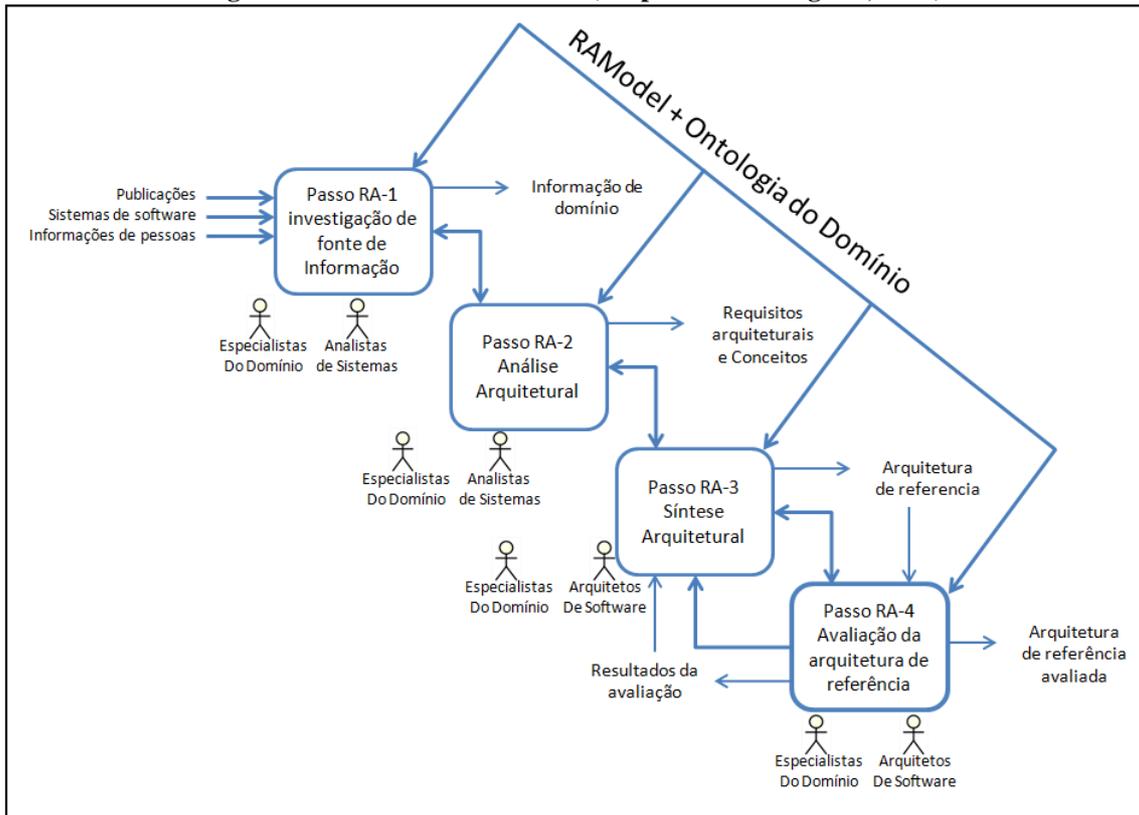
**Passo RA-1 – investigação da fonte de informação:** neste passo são selecionadas as fontes para obter informações a respeito de processos, atividades e tarefas que devem ser automatizados para o domínio. Como fontes de informação podem ser utilizadas: pessoas que atuam neste domínio; sistemas já existentes do domínio; publicações e documentos contendo informações do domínio; modelos de referência e arquiteturas de referência, ontologias do domínio; dentre outras. O resultado deste passo são as fontes de informações selecionadas. Pode-se utilizar o modelo de referência para arquiteturas de referência (RAModel) proposto por Nakagawa (2012).

**Passo RA-2 – análise arquitetural:** neste passo é realizada análise das fontes de informação selecionadas, obtendo um conjunto de requisitos para a arquitetura de referência e os principais conceitos que devem ser considerados neste domínio.

**Passo RA-3 – síntese arquitetural:** neste passo são construídas as visões arquiteturais. Essas visões devem apresentar um tratamento especial para levar em conta os interesses transversais, ou seja, interesses que estão espalhados com outros interesses do sistema encapsulando em unidades chamadas de que são considerados aspectos (NAKAGAWA, 2006) e devem ser representados utilizando uma notação apropriada. Como resultado deste passo, um conjunto de visões arquiteturais é criado para descrever a arquitetura de referência orientada a aspectos.

**Passo RA-4 – avaliação da arquitetura de referência:** neste passo, utilizando a abordagem de inspeção por *checklist*, a qualidade da arquitetura de referência é avaliada. Esse *checklist* avalia as características de qualidade como a manutenibilidade, a performance, a segurança, a usabilidade, a portabilidade e o reúso. Também é avaliada a descrição arquitetural, tentando identificar erros de omissão, inconsistência e informações incorretas.

**Figura 2.4 - Passos do ProSA-RA (adaptado de Nakagawa, 2014)**



O ProSA-RA é o processo de definição de arquiteturas de referência adotado para o estabelecimento da Cambuci-LPN pois contém passos detalhados que devem ser seguidos e artefatos que devem ser elaborados, diferentemente do ciclo de desenvolvimento proposto por Cloutier (2010). Na seção a seguir é apresentada a arquitetura de referência RefTEST-SOA, que é uma arquitetura de referência baseada em serviços e que foi definida com o apoio do ProSA-RA.

## 2.5 Arquitetura de referência RefTEST-SOA

RefTEST-SOA é uma arquitetura de referência orientada a serviços (SOA - *Service Oriented-Architecture*) para ferramentas de teste de software que visa dar apoio

ao desenvolvimento de ferramentas de teste orientadas a serviço (OLIVEIRA, 2011). Para definir a arquitetura de referência RefTEST-SOA foi utilizado o ProSA-RA (Seção 2.4), que também é o processo utilizado para a concepção da Cambuci-LPN.

A computação orientada a serviços (PAPAZOGLU *et al.*, 2008) utiliza os serviços para construir aplicações de um modo rápido, de baixo custo e distribuídas. Nesta abordagem, os componentes da aplicação são montados com pouco esforço utilizando uma rede de serviços que são fracamente acoplados, a qual permite criar processos de negócio dinâmicos e aplicações de maneira ágil.

Segundo Erl (2009), SOA estabelece um modelo arquitetônico que visa aprimorar a eficiência, a agilidade e a produtividade de uma empresa, posicionando os serviços como os principais meios para que a solução lógica seja representada no suporte à realização dos objetivos estratégicos associados à computação orientada a serviços.

SOA segue um conjunto de princípios de *design* que são (ERL, 2009):

- Contrato de serviço padronizado: garante que os pontos de contato estabelecidos pelos serviços sejam consistentes, confiáveis e governáveis;
- Baixo acoplamento de serviço: permite que o *design* e a lógica de um serviço possam evoluir independentemente de sua implementação, garantindo a interoperabilidade básica com consumidores que utilizam as capacidades do serviço;
- Abstração de serviço: enfatiza a necessidade de ocultar o maior número possível de detalhes subjacentes de um serviço;
- Capacidade de reúso de serviço: enfatiza o posicionamento de serviços, como recursos corporativos com contextos funcionais agnósticos, ou seja, a lógica encapsulada pelo serviço associa-se a um contexto para que qualquer cenário de uso possa ser considerado reutilizável;
- Autonomia de serviço: os serviços precisam ter um grau significativo de controle sobre seu ambiente e recursos;
- Independência de estado de serviço: os serviços são projetados para manterem informações de estado apenas quando essas forem necessárias;

- Visibilidade do serviço: os serviços precisam ser facilmente identificados e entendidos quando houver oportunidades de reuso;
- Composição de serviços: capacidade de compor efetivamente os serviços através de configurações de composição;

Assim, SOA tem como principal característica o acoplamento fraco entre os serviços, facilitando a adaptabilidade dos sistemas, possibilitando a construção de sistemas altamente dinâmicos pelo fato que os serviços podem ser substituídos (ou melhorados) de maneira transparente em tempo de execução (TOYOHARA, 2011).

A RefTEST-SOA é representada por uma visão geral e por três visões arquiteturais que são: visão de módulo, visão em tempo de execução e visão de implantação.

A visão geral da RefTEST-SOA, ilustrada na Figura 2.5, é composta por seis camadas, que são: Camada de Aplicação, Camada de Persistência, Camada de Apresentação, Camada de Intermediação, Camada de Processo de Negócio e a Camada de Qualidade de Serviços, descritas a seguir

**A Camada de Aplicação** contempla os conceitos relacionados ao domínio de teste de software, é dividida em quatro grupos de serviços:

- Serviços de Teste Primários: serviços que oferecem apoio ao gerenciamento de Caso de Teste, Critério de Teste, Artefato de Teste e Requisito de Teste;
- Serviços Ortogonais de Suporte: serviços que apoiam as atividades de Engenharia de Software consideradas como atividades de suporte pela ISO/IEC 12207 (2009) e que fazem parte do domínio de teste;
- Serviços Ortogonais Organizacionais: serviços que apoiam as atividades de Engenharia de Software consideradas como atividades organizacionais pela ISO/IEC 12207 (2009) e que fazem parte do domínio de teste; e
- Serviços de Ortogonais Gerais: serviços de propósito gerais, como persistência e segurança.

**A Camada de Persistência** é responsável por armazenar os dados produzidos pelos serviços que compõem as ferramentas de teste e a **Camada de**

**Apresentação** da RefTEST-SOA, por ser uma arquitetura orientada a serviços, é parte de cada serviço oferecido e possui três principais elementos:

- **Descrição do Serviço:** define os tipos de dados utilizados na requisição de funcionalidades e define os tipos de dados retornados aos clientes após o processamento;
- **Engine de Serviço:** processa as requisições de serviços; e
- **Controlador:** trata as requisições encaminhadas pela *Engine* de Serviço.

A **Camada de Intermediação** é onde os serviços de teste podem ser publicados, descobertos, associados e disponibilizados; e é composta por três elementos:

- **Registro de Serviços:** recebe as publicações e pesquisa sobre os serviços disponíveis;
- **Agente de Serviço:** arcabouço de roteamento e transporte de requisições entre serviços clientes e seus provedores; e
- **Escalonador de Serviço:** processa as requisições de serviço feitas por clientes de acordo com as informações de dependência existentes.

A **Camada de Processo de Negócio** permite a construção de novos serviços a partir da composição dos serviços de teste, organizacionais, de suporte e gerais já existentes. Por fim, a **Camada de Qualidade de Serviço** da visão geral da RefTEST-SOA inspeciona o cumprimento de requisitos de qualidade presentes nas demais camadas de serviços.

Na visão de módulo de Oliveira (2011), representada na Figura 2.6, a arquitetura de referência RefTEST-SOA é representada por meio de pacotes, classes e interfaces que descrevem as unidades de código que implementam funcionalidades representadas pelas suas partes que a compõem. Os serviços relacionados às atividades de teste estão agrupados nos seguintes conjuntos: serviços de testes primários (`primaryTestingServices`), serviços ortogonais de suporte (`orthogonalSupportingServices`), serviços ortogonais organizacionais (`orthogonalOrganizationalServices`) e serviços ortogonais gerais (`orthogonalGeneralServices`).

Figura 2.5 - Visão geral da RefTEST-SOA (OLIVEIRA, 2011)

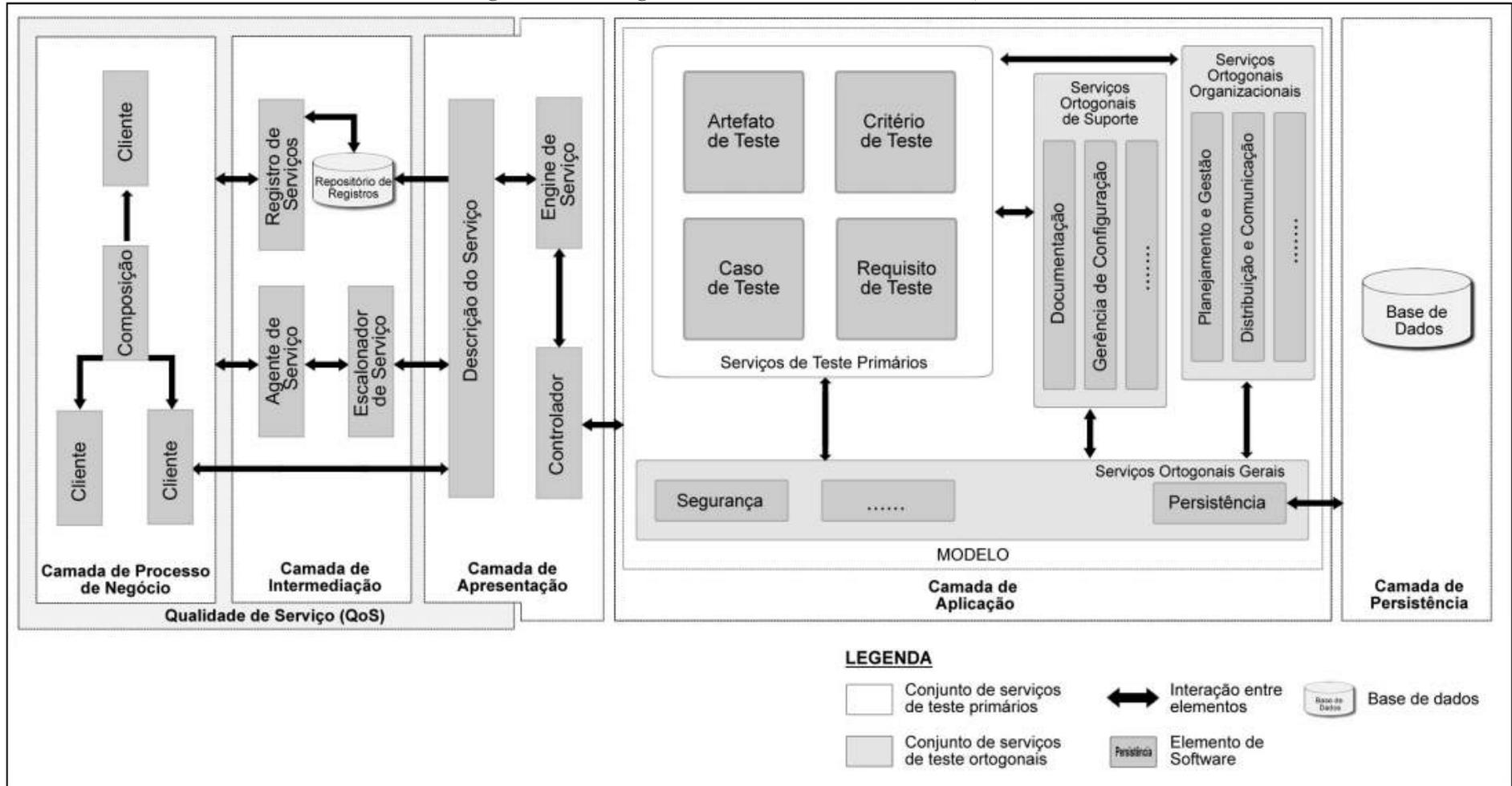
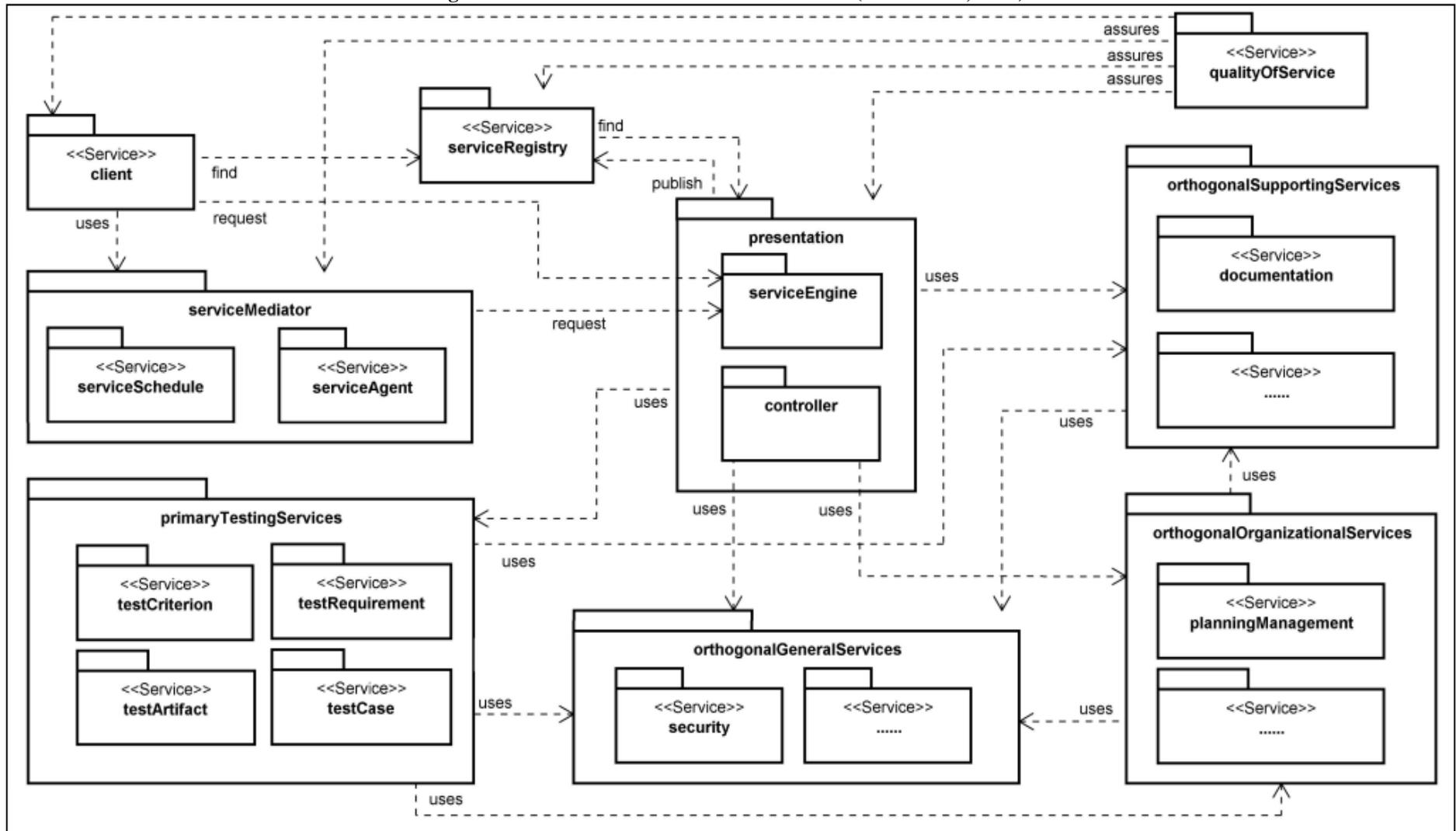


Figura 2.6 - Visão de módulo da RefTEST-SOA (OLIVEIRA, 2011)



Outros serviços (`serviceSchedule`, `serviceAgent`, `serviceRegistry` e `qualityOfService`) permitem as interações entre os serviços de teste, de suporte, organizacionais e gerais.

A visão em tempo de execução de Oliveira (2011), apresentada na Figura 2.7, representa a estrutura do sistema no momento em que ele está sendo executado, por meio de representação de componentes em diferentes níveis de abstração, das interfaces providas e requeridas, do fluxo de dados do sistema, das bases de dados, dos elementos replicados e das execuções que ocorrem em paralelo.

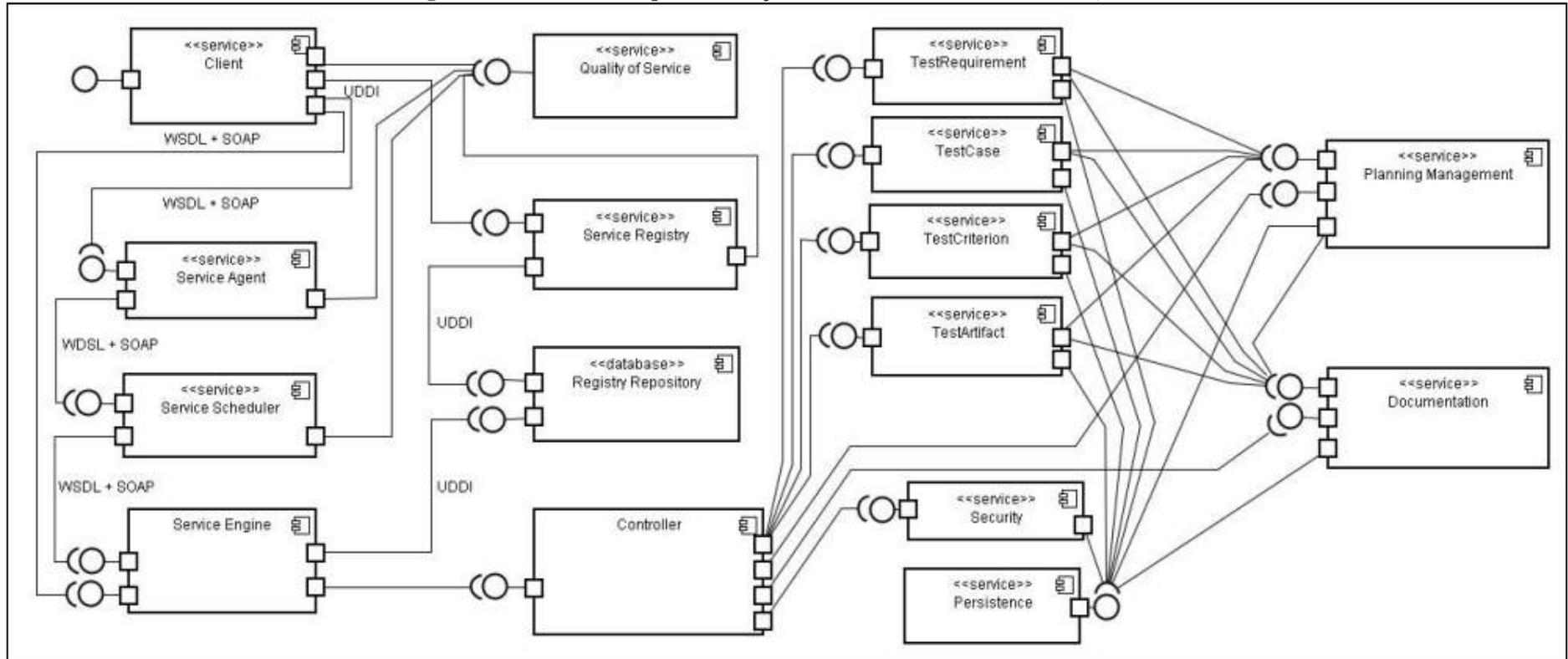
Nesta visão, os componentes que representam os serviços de testes primários são responsáveis por prover as funcionalidades relacionadas aos conceitos do domínio de teste.

O componente que representa o serviço Cliente (`Client`) pesquisa um serviço desconhecido no componente `Service Registry`, que informará o endereço do serviço de teste. Após obter o endereço, o cliente solicita a descrição do serviço que ele deseja utilizar (essa descrição pode ser feita por meio de uma linguagem como a WSDL - *Web Services Description Language* (W3C, 2001)) para poder se comunicar com esse serviço, direta ou indiretamente (através do componente `Service Agent`). O componente `Quality of Service` monitora a comunicação entre os serviços clientes e os serviços provedores.

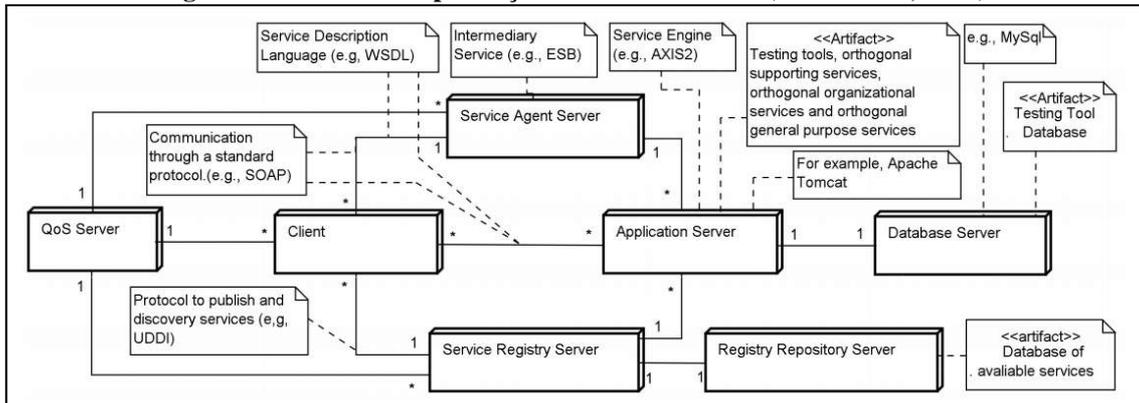
A visão de implantação de Oliveira (2011), apresentada na Figura 2.8, representa a estrutura de hardware sobre a qual os sistemas são alocados e as conexões de rede que simbolizam as interações entre esses dispositivos. Os elementos considerados nessa visão são: servidor de aplicação, servidor de base de dados, servidor de registro de serviços, repositório para armazenamento dos registros de serviço, servidor de agente de serviço e o servidor de qualidade de serviço. Nessa visão são apresentadas soluções vinculadas a cada elemento, que podem ser adotadas para concretizar a visão.

Salienta-se que a RefTEST-SOA é utilizada como fonte de informação para a concepção da Cambuci-LPN, visto que também é uma arquitetura de referência orientada a serviços e também pertence ao domínio de ferramentas de Engenharia de Software.

Figura 2.7 - Visão em tempo de execução da RefTEST-SOA (OLIVEIRA, 2011)



**Figura 2.8 - Visão de implantação da RefTEST-SOA (OLIVEIRA, 2011)**



## 2.6 Arquiteturas de referência para repositórios de ativos de software

No período de novembro de 2012 a abril de 2013 foi realizado um mapeamento sistemático (OSSHIRO, 2013), de acordo com Kitchenham (2004), com objetivo de encontrar trabalhos sobre arquiteturas de referência para repositórios de qualquer ativo de software.

Durante o mapeamento sistemático conduzido foram utilizadas como bases de buscas as bibliotecas digitais *ACM Digital Library*, *IEEEExplore Digital Library*, *Scopus*, *ScienceDirect* e *SpringerLink*; e a *string* de busca "reference architecture" AND repository foi adaptada para a máquina de busca de cada uma dessas bibliotecas. Salienta-se que uma *string* de busca contendo os termos adicionais "business process line" OR "business process family" foi descartada visto que nenhuma arquitetura de referência de repositórios de LPN foi retornada.

Após a aplicação dos critérios de inclusão e de exclusão descritos em OSSHIRO (2003), foram obtidos três trabalhos listados no Quadro 2.1.

**Quadro 2.1 - Artigos selecionados no mapeamento sistemático**

<b>Id.</b>	<b>Autores</b>	<b>Títulos</b>
1	MELAND <i>et al.</i> (2009)	<i>An architectural foundation for security model sharing and reuse.</i>
2	YAN <i>et al.</i> (2012)	<i>Business process model repositories – Framework and survey.</i>
3	HONGMIN <i>et al.</i> (2010)	<i>Software asset repository open framework supporting customizable faceted classification.</i>

Levando em consideração o ano de publicação de cada um dos trabalhos selecionados, observa-se que todos foram publicados recentemente, indicando que a linha de pesquisa de arquitetura de referência de repositórios que apoiam o reúso de software é nova e merece mais estudos para obter maior maturidade. Nas seções a seguir é apresentada uma análise dos trabalhos selecionados levando em consideração

as três questões de pesquisa definidas durante o planejamento do mapeamento sistemático: 1) Alguma técnica, processo, método, metodologia ou padrão é seguido para definir arquiteturas de referência para repositórios de ativos de software? 2) Quais tipos de ativos de software são considerados pelas arquiteturas de referência de repositórios de ativos de software? e 3) Quais visões são utilizadas para representar arquiteturas de referências de repositórios de ativos de software?

### **2.6.1 Uso de processos de definição de arquiteturas de referência**

Nenhum dos trabalhos selecionados segue um processo para apoiar a definição da arquitetura de referência. Isso mostra a carência na literatura do uso de processos específicos para a definição de arquiteturas de referência como é o caso do ProSA-RA citado na Seção 2.4.

O trabalho de Meland *et al.* (2009) trata de um estudo sobre repositório de artefatos de segurança, que foi desenvolvido em colaboração com vários parceiros na Europa, envolvendo instituições acadêmicas e indústrias de software. Esse estudo, apesar de não seguir um processo para a definição da arquitetura de referência, utiliza alguns passos para definir os requisitos funcionais do repositório de artefatos de segurança.

A descrição arquitetural da arquitetura de referência de Meland *et al.* (2009) foi desenvolvida de acordo com as práticas recomendadas pelo *framework* MAFFIA (JOHANSEN, 2003) e pela norma IEEE 1471 (2000), que são respectivamente um *framework* para descrições arquiteturais para sistemas de software intensivo e um padrão IEEE 1471 (2000) para descrição de arquitetura de sistema de software intensivo, também conhecido como arquitetura de software. Sistemas de software intensivo, segundo IEEE 1471 (2000) é qualquer sistema em que o software contribui com influências essenciais para o projeto, construção, implantação e evolução do sistema como um todo.

O segundo trabalho selecionado (YAN *et al.*, 2012) também segue um conjunto de passos para a definição de uma arquitetura de referência de repositórios de modelos de processos de negócio, conforme discutidos na Seção 3.4 do Capítulo 3, mas não segue um processo específico de definição de arquitetura de referência. Além disso, os autores utilizam apenas a visão geral para representar a arquitetura de referência proposta.

O terceiro trabalho selecionado (HONGMIN *et al.*, 2010) trata de uma arquitetura de referência para repositórios de ativos de software, que permite o compartilhamento dos ativos entre os repositórios e entre ferramentas de terceiros. Esse artigo apresenta alguns elementos arquiteturais, como por exemplo, o modelo de informação e a arquitetura do repositório orientado a serviços para o domínio de repositório de ativos de software, mas não descreve “como” esses elementos foram identificados.

### 2.6.2 Tipos de ativo atendidos pelas arquiteturas de referência para repositórios de ativos de software

No Quadro 2.2 são apresentados os tipos de ativos considerados pelas arquiteturas de referência tratadas pelos trabalhos selecionados.

**Quadro 2.2 - Tipos de ativo dos repositórios**

<b>Id.</b>	<b>Autores</b>	<b>Tipo de ativo</b>
1	Meland <i>et al.</i> (2009)	Ativo de Segurança
2	Yan <i>et al.</i> (2012)	Ativo de Modelo de Processo de Negócio
3	Hongmin <i>et al.</i> (2010)	Ativo de Software

Modelo de processo de negócio é o tipo de ativo tratado pela arquitetura de referência de Yan *et al.* (2012). Como esse tipo de ativo é relevante para o contexto de LPN, esse trabalho é utilizado como fonte de informação para a definição da arquitetura de referência Cambuci-LPN.

Meland *et al.* (2009) apresentam uma arquitetura de referência de repositórios de ativos de segurança e tem uma abordagem interessante para a integração desse tipo de repositório com ferramentas de desenvolvimento, como é o caso do Eclipse (ECLIPSE, 2013) e do Netbeans (ORACLE, 2013), visto que os desenvolvedores geralmente não possuem muito conhecimento nesse domínio. Dessa forma, os especialistas da área de segurança constroem os ativos e os disponibilizam no repositório para que os desenvolvedores possam reutilizá-los. Essa arquitetura de referência também é utilizada como fonte de informação para definir a arquitetura de referência Cambuci-LPN quanto a integração de repositórios de LPN com outras ferramentas, como é o caso de ferramentas de modelagem de processos de negócio, de ferramentas de controle de versão e de outros repositórios desse tipo de ativo.

A arquitetura de referência proposta por Hongmin *et al.* (2010) considera qualquer tipo de ativo de software e preocupa-se com a integração do repositório de

ativos de software com outras ferramentas de modelagem e de desenvolvimento, bem como com a comunicação entre repositórios e com ferramentas de terceiros. Isso é viabilizado pela arquitetura de referência por meio da arquitetura orientada a serviços (*Service Oriented-Architecture* - SOA), que também é de interesse deste trabalho. Exemplificando, a arquitetura de referência de Hongmin *et al.* (2010) permite que ferramentas de terceiros utilizem os ativos de software, armazenados no repositório, fazendo requisição de serviço ao sistema de gerenciamento do repositório por meio da arquitetura SOA. Adicionalmente, essa arquitetura de referência apresenta um repositório que permite uma classificação personalizada baseada em faceta, em que cada grupo de usuários tem acesso a determinados recursos do repositório conforme um esquema de facetas previamente definidas.

Assim, a arquitetura de referência de Hongmin *et al.* (2010) também é utilizada na concepção da arquitetura de referência Cambuci-LPN pois as suas características aqui descritas também são relevantes no contexto do repositórios de LPN.

### 2.6.3 Visões arquiteturais utilizadas pelas arquiteturas de referência para repositórios de ativos de software

No Quadro 2.3 são apresentadas as visões utilizadas para representar as arquiteturas de referência dos trabalhos selecionados, sendo que o modelo comumente utilizado é o modelo conceitual. Os trabalhos de Meland *et al.* (2009) e Hongmin *et al.* (2010) utilizam o modelo conceitual com o apoio da notação do diagrama de classes da UML (*Unified Modeling Language*) (OMG, 2011).

**Quadro 2.3 - Visões arquiteturais dos trabalhos selecionados**

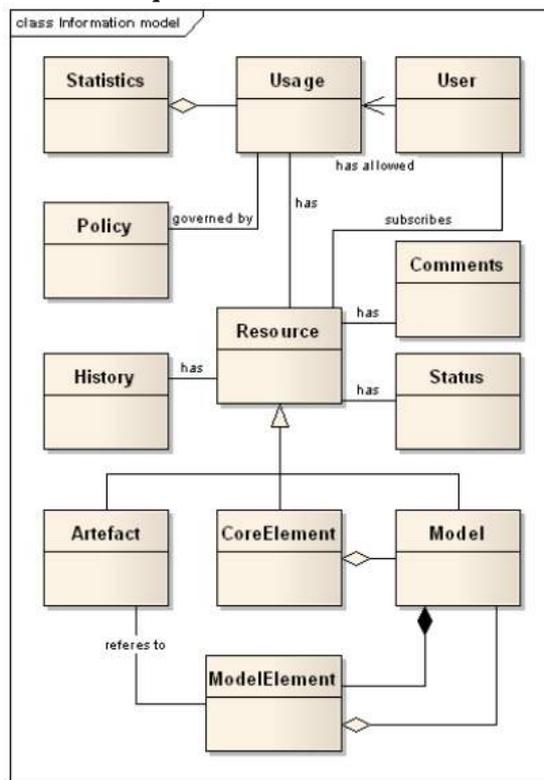
<b>Id.</b>	<b>Trabalhos</b>	<b>Visões Arquiteturais</b>
1	Meland <i>et al.</i> (2009)	Modelo de Informação (correspondente ao Modelo Conceitual) Modelo de Componentes Lista de <i>Stakeholders</i>
2	Yan <i>et al.</i> (2012)	Visão Geral
3	Hongmin <i>et al.</i> (2010)	Modelo de Informação (Modelo Conceitual) Documentação das Interfaces de Serviços

O trabalho de Meland *et al.* (2009) utiliza o modelo de componentes que permite compreender como a arquitetura de referência suportará a comunicação com ferramentas externas de modelagem e de desenvolvimento. Também apresenta nesse

modelo a comunicação com fontes de informações externas. A lista de *stakeholders* é utilizada para documentar os atores que interagem com o sistema.

Na Figura 2.9 é ilustrado o modelo conceitual da arquitetura de referência de repositórios de ativos de segurança de Meland *et al.* (2009). Nesse modelo, além dos conceitos do domínio, encontra-se o conceito *History* que permite o suporte ao versionamento e aprovação do conteúdo no repositório. *Comments* são utilizados para o *feedback* dos usuários do repositório. As permissões são definidas no conceito *Usage* conforme as políticas definidas (conceito *Policies*). O conceito *Statistics* é utilizado para fornecer informações sobre a popularidade dos ativos disponíveis no repositório. Esses conceitos são importantes, independentemente do ativo tratado pelo repositório. Assim, são também necessários em repositórios de LPN e são considerados na arquitetura de referência Cambuci-LPN.

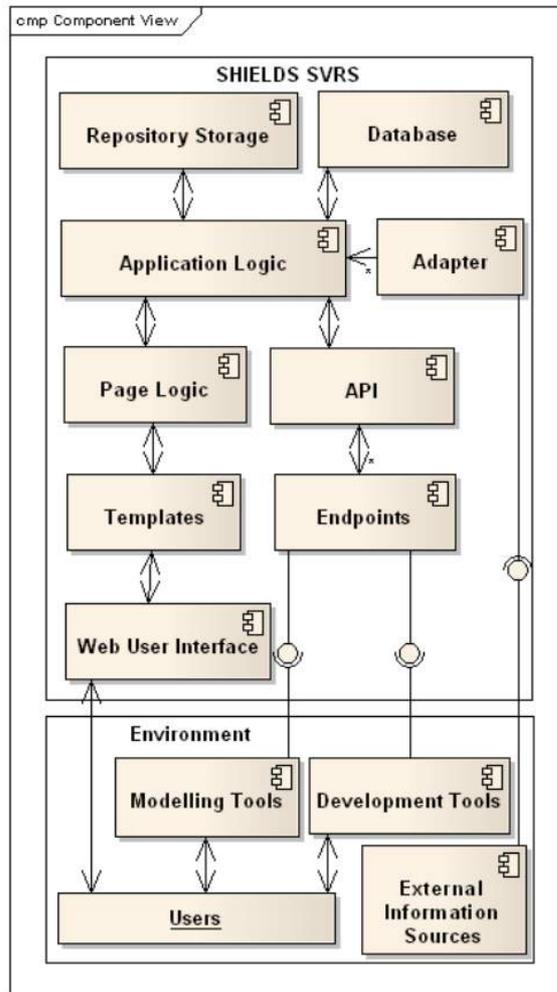
**Figura 2.9 - Modelo conceitual da arquitetura de referência do trabalho 1 (MELAND *et al.* 2009)**



A visão de componentes da arquitetura de referência de Meland *et al.* (2009), ilustrada na Figura 2.10, é fracamente baseada na arquitetura de *Flickr* (2013), que é uma aplicação de sucesso no gerenciamento e compartilhamento de vídeos e fotos, possui um repositório centralizado e altamente escalável, oferece conteúdo por

meio de uma interface *Web* e possui uma interface para aplicativos de terceiros (permitindo as funcionalidades de *download* e submissão de conteúdo).

**Figura 2.10 - Visão de componentes da arquitetura de referência do trabalho 1 (MELAND *et al.*, 2009)**



Na visão de componentes, a camada da lógica de aplicação (*Application Logic*) troca informações com diversas camadas: camada de armazém do repositório (*Repository Storage*) – a qual os ativos de segurança ficam armazenados; camada do banco de dados (*Database*) que contém informações de usuários, sessões e estatísticas; camada adaptador (*Adapter*) - que permite acesso a fontes externas; camada página da lógica – que permite a interação com os *Users* através da *Web*; e camada API (*Application Programming Interface*) – que fornece um subconjunto relevante de todas as funcionalidades presentes na camada lógica para as ferramentas de modelagem e desenvolvimento.

Conforme a descrição da visão de componentes, os usuários podem acessar o repositório de ativos de segurança por meio de interface *Web* ou por meio de ferramentas de modelagem e de desenvolvimento. Nesse último caso, o acesso se dá por meio de *Endpoints*. Os *Endpoints* se comunicam com a lógica da aplicação por meio de *APIs* que definem um subconjunto de funcionalidades que a lógica da aplicação disponibilizará para as ferramentas de terceiros visando compartilhar os ativos de segurança disponíveis no repositório.

A comunicação com ferramentas de terceiros também é de interesse de repositórios de LPN, em que ferramentas de modelagem necessitam interoperar com o repositório. Também existe a necessidade desse tipo de repositório se comunicar com outros repositórios do mesmo domínio.

O trabalho de Yan *et al.* (2012) possui apenas visão geral por causa do nível de abstração da arquitetura de referência proposta pelos autores. Nesse trabalho, apesar de não existir o modelo conceitual, existe o modelo de gerenciamento de repositório de modelos de processos de negócio, a qual é identificada a necessidade do modelo de dados.

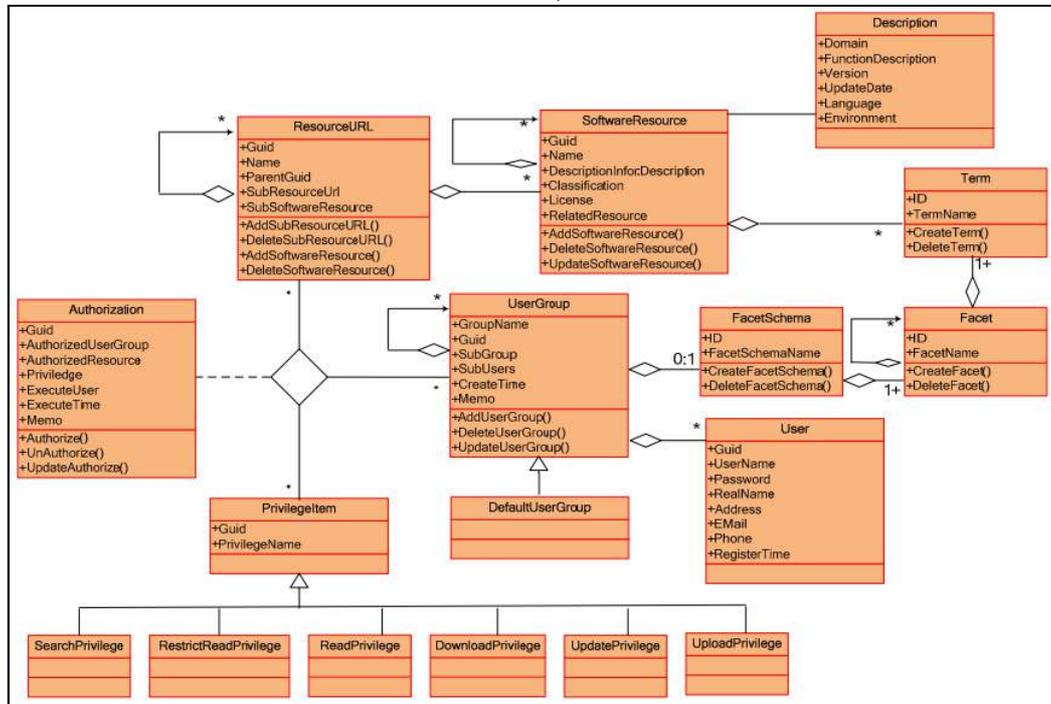
Na arquitetura de referência de Yan *et al.* (2012), apesar de não existir a visão de módulo, através do modelo de gerenciamento de repositório de modelos de processos de negócio, é possível identificar algumas funcionalidades que devem estar presentes na arquitetura de referência de repositórios de modelos de processos de negócio. Mais detalhes de cada uma dessas visões são obtidos na Seção 3.4.

No trabalho de Hongmin *et al.* (2010), além do modelo conceitual, os autores apresentam a documentação das interfaces de serviços, que permitem a integração dos repositórios propostos com outros repositórios e outras ferramentas por meio da disponibilização dos serviços pelo repositório utilizando a arquitetura orientada a serviços.

O modelo conceitual da arquitetura de referência de Hongmin *et al.* (2010) é ilustrado na Figura 2.11. Nesse modelo são apresentados os conceitos de facetas (*FacetSchema*, *Facet*), permitindo o suporte à classificação facetada para o repositório. Um esquema de faceta (*FacetSchema*) é composto por um conjunto de facetas (*Facet*), enquanto que cada faceta é composta por outras facetas e por termos (*Term*). Cada grupo de usuários (*UserGroup*) pode estabelecer e manter esquemas facetados individuais para classificar seus próprios ativos. O acesso a cada recurso é

determinado pelo conceito *Authorization* que atribui um determinado privilégio por meio do conceito *PrivilegeItem*.

**Figura 2.11 - Modelo conceitual da arquitetura de referência do trabalho 3 (HONGMIN *et al.*, 2010)**



### 2.6.4 Outras considerações sobre arquiteturas de referências para repositórios de ativos de software

A preocupação dos trabalhos de Meland *et al.* (2009) e Hongmin *et al.* (2010) com a integração de repositórios com outras ferramentas também é contemplada na Cambuci-LPN.

A abordagem de classificação facetada, considerada na arquitetura de referência de Hongmin *et al.* (2010), é utilizada em repositórios de LPN para permitir a troca de ativos entre os diversos repositórios por meio de buscas mais automatizadas.

No Quadro 2.4 é apresentado um resumo das respostas de cada questão de pesquisa pertencentes ao mapeamento sistemático. Além disso, devido à importância do Processo de Gestão de Ativos da norma ISO/IEC 12207 (2009) para este trabalho, cujo propósito é gerenciar o ciclo de vida de ativos reutilizáveis desde a concepção até a sua desativação, observou-se quais resultados esperados desse processo são atendidos por cada arquitetura de referência selecionada.

A partir da análise dos trabalhos selecionados para responder essas questões, observou-se que os autores não utilizaram um processo específico para apoiar a

definição das arquiteturas de referência estudadas e não se preocuparam com a validação das mesmas. Diferentemente da arquitetura de referência proposta neste trabalho, que é definida com o apoio do processo ProSA-RA, apresentado na Seção 2.4.

Apesar da arquitetura de referência de repositórios de modelos de processos de negócio atender a maior parte dos resultados esperados do Processo de Gestão de Ativos (ISO/IEC 12207, 2009), as visões arquiteturais são apresentadas sem muitos detalhes. As demais arquiteturas de referência atendem apenas alguns resultados esperados da gestão de ativos.

Do ponto de vista da arquitetura de referência Cambuci-LPN, pretende-se contemplar todos os resultados esperados da norma ISO/IEC 12207 (2009) uma vez que a ontologia de especificação e de gestão de ativos ONTO-ResAsset (Seção 3.5 do Capítulo 3) é considerada como fonte de informação e a sua concepção também foi baseada nessa norma.

**Quadro 2.4 - Resumo geral das questões de pesquisa e respectivas respostas**

<b>Trabalho</b>	<b>Tipo de ativo</b>	<b>Maneira em que foi definida</b>	<b>Visões contempladas</b>	<b>Resultados esperados do Processo de Gestão de Ativos atendidos</b>
Meland <i>et al.</i> (2009)	Ativos de Segurança	Conjunto de passos definidos pelos autores	- Modelo Conceitual - Modelo de Componentes - Lista de <i>Stakeholders</i>	- Armazenamento de ativos - Recuperação de ativos
Yan <i>et al.</i> (2012)	Modelos de Processos de Negócio	Conjunto de passos definidos pelos autores	- Visão Geral - modelo de gerenciamento de repositório de modelos de processos de negócio	- Classificação de ativos - Tratamento de aceitação, certificação e desativação de ativos - Armazenamento de ativos - Recuperação de ativos - Controle das alterações dos ativos - Notificação aos usuários
Hongmin <i>et al.</i> (2010)	Ativos de Software	Não apresenta	- Modelo Conceitual	- Classificação de ativos - Armazenamento de ativos - Recuperação de ativos

## 2.7 Considerações finais

Neste capítulo observou-se que o termo arquitetura de referência é amplamente utilizado e, às vezes, de forma diferenciada. Existem vários estudos para propor o que deve conter uma arquitetura de referência, mas a conclusão que se chega é que varia conforme o domínio do problema e conforme o nível de abstração desejado.

No Quadro 2.5 é feito um comparativo entre os processos de desenvolvimento de arquiteturas de referência estudados. No processo de

desenvolvimento proposto por Cloutier (2010), a evolução da arquitetura de referência faz parte do ciclo de desenvolvimento.

**Quadro 2.5 - Comparativo entre processos de desenvolvimento de arquiteturas de referência**

Itens Analisados	Ciclo de desenvolvimento de arquiteturas de referência (CLOUTIER, 2010)	ProSA-RA (NAKAGAWA, 2014)
Tipo de fontes de informação	Arquiteturas de sistemas existentes	Documentos Sistemas existentes Pessoas Ontologia do Domínio
Detalhamento dos passos	Pouco detalhado	Melhor detalhamento
Preocupação com a evolução	Faz parte do ciclo da arquitetura de referência	Não discute
Preocupação com a avaliação da arquitetura	Não considera	Possui um passo para avaliação

O processo proposto por Nakagawa (2014), ao contrário do proposto por Cloutier (2010), possui várias fontes de informação, o que permite uma melhor definição da arquitetura.

Nakagawa (2014) possui um passo específico para conduzir a avaliação da arquitetura de referência proposta, visando certificar a qualidade da arquitetura de referência desenvolvida, tomando como base um *checklist*. Por outro lado, Cloutier (2010) não se preocupa com a avaliação da arquitetura de referência.

Devido ao melhor detalhamento dos passos do processo de estabelecimento de uma arquitetura de referência proposto por Nakagawa (2014), o ProSA-RA é utilizado para o estabelecimento da arquitetura de referência Cambuci-LPN.

Pelo fato da RefTEST-SOA (OLIVEIRA, 2011), apresentada na Seção 2.5, seguir os passos descritos no ProSA-RA para o seu estabelecimento, e principalmente por ser baseada no paradigma orientado a serviços, essa arquitetura de referência foi selecionada para ser utilizada como base para o estabelecimento da arquitetura Cambuci-LPN, que também tem a proposta de ser orientada a serviços.

A partir da discussão dos trabalhos correlatos, apresentada na Seção 2.6, é possível notar que vários estudos apresentam arquiteturas de referência para repositórios considerando algum tipo de ativo de software. No entanto, até o encerramento desta dissertação, não foi encontrada uma arquitetura de referência para repositórios de linhas de processos de negócio, que é o foco deste trabalho.

## Capítulo 3 - Modelagem de processos de negócio

### 3.1 Considerações iniciais

Neste capítulo é abordado o tema de modelagem de processos de negócio, pois é um dos conceitos importantes para o entendimento de LPN, que envolve basicamente o uso de conceitos de LPS no contexto de processos de negócio, discutidos no próximo capítulo.

Na Seção 3.2 são apresentados conceitos e definições sobre modelagem de processos de negócio, bem como é dado enfoque à notação BPMN (*Business Process Model and Notation*) (OMG, 2013), que é a notação padrão da OMG (*Object Management Group*) para a modelagem de processos de negócio.

Na Seção 3.3 são apresentados alguns estudos na área de modelagem de processo de negócio e são discutidos alguns requisitos importantes que devem ser atendidos pela arquitetura de referência Cambuci-LPN, observados a partir da análise desses estudos. Devido à importância de apoio computacional provido pelos repositórios para o gerenciamento do reúso de modelos de processos de negócio e que também são considerados no contexto de LPN, na Seção 3.4 é apresentado o arcabouço para repositórios de modelos de processos de negócio, proposto por Yan *et al* (2012).

Como modelo de processos de negócio é um ativo relevante de LPN e repositórios são o apoio computacional utilizado para a gestão de ativos, na Seção 3.5 é apresentada a ONTO-ResAsset que é um ontologia de especificação e gestão de ativos reutilizáveis. Na Seção 3.6, os mecanismos de gestão de ativos dessa ontologia são comparados com as funcionalidades encontradas no estudo de Yan *et al.* (2012).

Por fim, na Seção 3.7 são apresentadas as considerações finais do capítulo e como esses estudos serviram de base para apoiar na definição da arquitetura de referência Cambuci-LPN.

### 3.2 Conceitos

De acordo com Vaisman (2013), **processo de negócio** é uma coleção de atividades relacionadas e estruturadas, que produzem um específico serviço ou produto o qual serve para alcançar um determinado objetivo de um cliente particular. Complementarmente, um **fluxo de trabalho** (*workflow*) representa a automação de um processo de negócio, no todo ou em parte, durante o qual documentos, informações ou

tarefas são enviados de um participante para outro, de acordo com um conjunto de regras.

Vaisman (2013) também definem um **modelo de processo** como sendo uma visão formalizada de um processo de negócio, representada por um conjunto coordenado de atividades paralelas e/ou sequenciais que são conectadas para alcançar um objetivo comum. Tal modelo é elaborado por meio da **modelagem de processos de negócio (MPN)**, que é uma técnica de modelagem a qual usa métodos, outras técnicas e software para projetar, validar, controlar e analisar processos operacionais envolvendo humanos, organizações, aplicações, documentos e outras fontes de informações.

De acordo com Landre (2012), por meio da explicitação de objetivos, processos, papéis e regras do negócio, bem como os relacionamentos existentes entre eles, a modelagem de negócios é um técnica para documentar o conhecimento tácito sobre o funcionamento das organizações e, por isso, também pode ser usada na gestão do conhecimento dentro das empresas. Adicionalmente, esse tipo de modelagem pode ser útil também na Engenharia de Software e na Gestão de Processos. No primeiro caso porque facilita a elicitación dos requisitos uma vez que permite um melhor entendimento do negócio para o qual o sistema de software será construído; e no segundo caso porque apoia a melhoria dos processos dentro da organização.

Existem várias técnicas e notações que apoiam a representação de processos de negócio, como é o caso das Redes de Petri (REISIG, 1985), UML (*Unified Modeling Language*) (OMG, 2011), BPMN (*Business Process Modeling and Notation*) (OMG, 2013), entre outros. Essa última é apresentada a seguir por ser de interesse deste trabalho.

A representação de processos de negócio utilizando BPMN tem sido muito utilizada pela comunidade de modelagem de processos de negócio (VAISMAN, 2013), o que justifica considerar os seus principais elementos durante a concepção da arquitetura de referência Cambuci-LPN. Outro argumento para sua utilização é que importantes ferramentas de modelagem de processo de negócios, como Aris (ARIS, 2013) e Bizagi (BIZAGI, 2013), utilizam a BPMN como notação.

BPMN (*Business Process Modeling Notation*) é uma notação para modelagem de processos de negócio, definida em 2002 pela BPMI (*Business Process Modeling Initiative*) e aceita em 2006 pela OMG (*Object Management Group*) como modelagem padrão para processos de negócio. Em 2013 foi lançada a versão 2.0.2 da notação BPMN pela OMG.

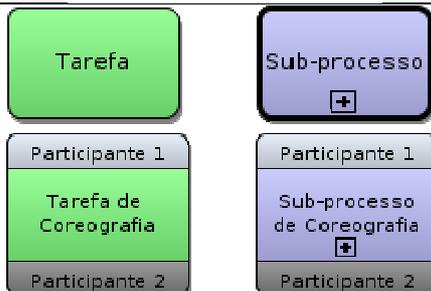
Os elementos que compõem a BPMN podem ser separados nas seguintes categorias (OMG, 2013): objetos de fluxos, objetos de dados, objetos de conexão, partições e artefatos.

Os objetos de fluxos são os elementos principais para definir comportamento de um processo de negócio e podem ser de três tipos (eventos, atividades, desvios), conforme descritos no Quadro 3.1.

Os objetos de dados são representados pelos seguintes elementos: dados, dados de entradas, dados de saídas, dados armazenados, conforme ilustrado Quadro 3.2.

Os objetos de conexão podem ser representados pelos seguintes objetos: fluxo de sequência, fluxo de mensagem, associações, associações de dados, conforme ilustrados no Quadro 3.3.

**Quadro 3.1 - Objetos de fluxo da notação BPMN 2.0.2 (adaptado de OMG, 2013)**

Objetos de Fluxo		
Elemento	Descrição	Notação Gráfica
Evento	Evento é um objeto de fluxo que representa algo que acontece durante um processo ou uma coreografia. Afetam o fluxo e normalmente possuem uma causa (gatilho – <i>trigger</i> ) ou um impacto (resultado). Os três eventos possíveis são o <b>evento de início</b> , <b>evento intermediário</b> e <b>evento de fim</b> .	 <p>Evento de início</p> <p>Evento intermediário</p> <p>Evento de fim</p>
Atividade	Atividade é um objeto de fluxo que representa um trabalho que um ator ou participante realiza. Uma atividade pode ser atômica ou não-atômica e são usadas tanto em Processos como em Coreografias. Os tipos de atividades são as <b>tarefas</b> (atividades atômicas), <b>sub-processos</b> , <b>tarefas de coreografia</b> e <b>sub-coreografias</b> .	 <p>Tarefa</p> <p>Sub-processo</p> <p>Participante 1</p> <p>Participante 2</p> <p>Tarefa de Coreografia</p> <p>Sub-processo de Coreografia</p> <p>Participante 1</p> <p>Participante 2</p>

Quadro 3.1(conclusão)

Objetos de Fluxo		
Elemento	Descrição	Notação Gráfica
Desvios	Um desvio é um objeto de fluxo usado para controlar a divergência e convergência de fluxos de sequência em um processo, ou seja, determina caminhos de ramificações, bifurcações, fusões e junções no diagrama. Há sete tipos de desvios: <b>desvio exclusivo</b> , <b>desvio inclusivo</b> , <b>desvio paralelo</b> , <b>desvio baseado em eventos</b> , <b>desvio paralelo baseado em eventos</b> , <b>desvio exclusivo baseado em eventos</b> e <b>desvio complexo</b> .	 <p>Exclusivo implícito    Exclusivo</p> <p>Inclusivo    Paralelo</p> <p>Baseado em evento    Exclusivo baseado em eventos</p> <p>Paralelo baseado em eventos    Complexo</p>

Quadro 3.2 - Elementos de dados dados da notação BPMN 2.0.2 (adaptado de OMG, 2013)

Elementos de Dados		
Elemento	Descrição	Notação Gráfica
Objeto de Dado	O objeto de dado fornece informação sobre o que uma atividade necessita para ser realizada e o que tal atividade produz.	 <p>Objeto de Dado</p>
Dado de Entrada e Dado de Saída	Os dados de entrada e dados de saída representam a mesma informação que um objeto de dado, porém, para processos e não para atividades. Em outras palavras, <b>atividade</b> necessita de um <b>objeto de dado</b> e produz outro objeto de dado, enquanto um <b>processo</b> necessita de um <b>dado de entrada</b> e produz um <b>dado de saída</b> .	 <p>Dado de Entrada    Dado de Saída</p>
Coleção de Objetos de Dado	Tanto objetos de dados, quanto dados de entrada e saída podem representar objetos únicos ou uma coleção de objetos, nesse caso utiliza-se a coleção de objetos de dado para representá-los.	 <p>Coleções de Dados</p>

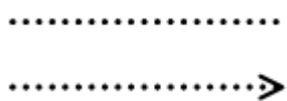
**Quadro 3.2 (conclusão)**

Elementos de Dados		
Elemento	Descrição	Notação Gráfica
Repositório de Dados	Fornece um mecanismo para atividades recuperar ou atualizar informações armazenadas que irão persistir além do escopo do processo. O mesmo repositório de dados pode ser visualizado por meio de uma referência em um ou mais locais do processo.	 Repositório de Dados
Mensagem	Uma mensagem descreve o conteúdo de uma comunicação entre dois participantes.	 Mensagem

**Quadro 3.3 - Objetos de conexão da notação BPMN 2.0.2 (adaptado de OMG, 2013)**

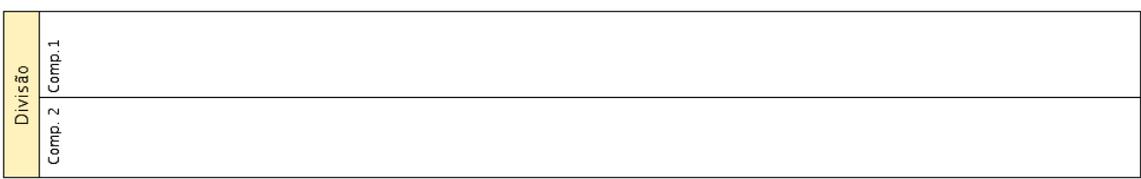
Objetos de Conexão		
Elemento	Descrição	Notação Gráfica
Fluxo de Sequência	Usado para mostrar a ordem em que as atividades serão realizadas em um processo e em uma coreografia. Um <b>fluxo de sequência</b> pode ter uma expressão de condição para determinar se o mesmo será usado ou não (durante a execução do processo), nesse caso tal fluxo é dito <b>fluxo condicional</b> . Para desvios exclusivos ou inclusivos, um tipo de fluxo possível é o <b>fluxo padrão</b> , a execução do processo só passará por tal fluxo se a condição do outro fluxo condicional não for verdadeira. Por fim, o <b>fluxo de exceção</b> ocorre fora do fluxo normal do processo e é baseado em um evento intermediário anexado a uma atividade que ocorre durante a execução.	 Fluxo de Sequência   Fluxo Condicional   Fluxo Padrão   Fluxo de Exceção

**Quadro 3.3 (conclusão)**

Objetos de Conexão		
Elemento	Descrição	Notação Gráfica
Fluxo de Mensagem	Usado para mostrar o fluxo das mensagens trocadas entre dois participantes que são preparados para enviá-las e recebê-las.	 <p>Fluxo de mensagem</p>
Associação e Associação de Dados	<p>A <b>associação</b> é usada para vincular informações e artefatos com elementos gráficos BPMN. Quando é <b>direcionada</b>, indica a direção do fluxo, quando apropriado.</p> <p>Uma <b>associação de dados</b> é usada para mostrar como objeto de dados são entradas ou saídas para atividades e usa a mesma notação que uma associação direcionada.</p>	 <p>Associações</p>

As partições são usadas para agrupar os elementos de modelagem primária e são representadas por divisões e compartimentos, conforme ilustrado no Quadro 3.4.

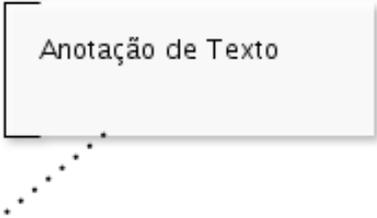
**Quadro 3.4 - Elementos de agrupamento da notação BPMN 2.0.2 (adaptado de OMG, 2013)**

Partições	
Elemento	Descrição
Divisão e Compartimento	<p><b>Divisão</b> é a representação gráfica do processo em um Diagrama de Processo ou a representação gráfica de um participante em um Diagrama de Colaboração, nesse caso pode conter elementos, representando assim o processo privado de cada participante, ou pode estar vazia, “caixa-preta”, quando os detalhes internos do participante não são relevantes ao contexto do diagrama. Desempenha o papel de uma partição para separar um conjunto de atividades de outras divisões, normalmente no contexto de B2B.</p> <p>O <b>compartimento</b> é uma sub-partição dentro de um processo ou de um participante, no caso dos diagramas de colaboração. São utilizados para organizar e categorizar as atividades, representando, por exemplo, papéis ou departamentos dentro do processo de uma organização.</p>
<b>Notação Gráfica</b>	
	

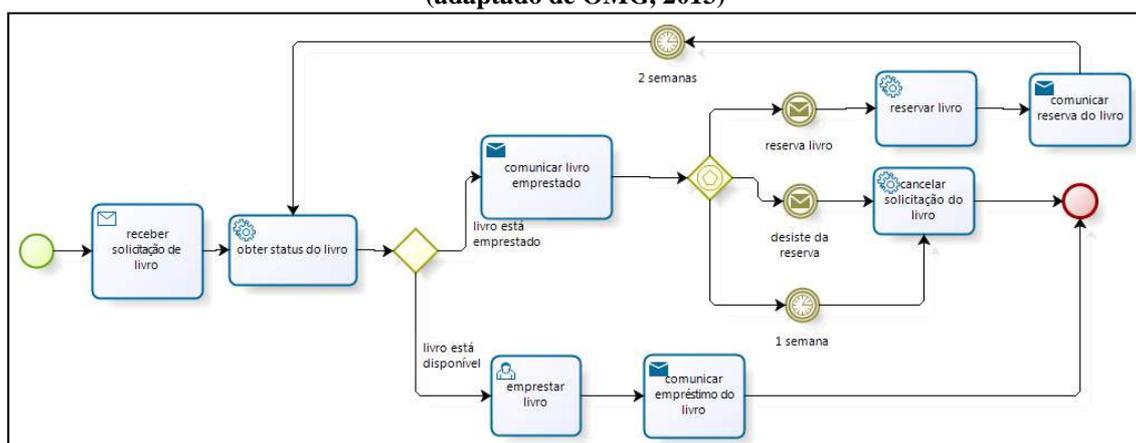
Os artefatos são usados para fornecer informação adicional sobre os processos e não afetam o fluxo do processo. Os artefatos são representados pelo agrupamento e pela anotação de texto, conforme ilustrado no Quadro 3.5.

Exemplificando, na Figura 3.1 é ilustrada a modelagem do processo de negócio de uma requisição de livro em uma biblioteca utilizando a notação BPMN.

**Quadro 3.5 - Artefatos da notação BPMN 2.0.2 (adaptado de OMG, 2013)**

Objetos de Conexão		
Elemento	Descrição	Notação Gráfica
Agrupamento	Utilizado para agrupar elementos do diagrama que possuem características semânticas semelhantes, ou seja, elementos de mesma categoria. Esse tipo de agrupamento não afeta o fluxo de sequência. O nome da categoria aparece como um rótulo.	
Anotação de Texto	Um mecanismo para o modelador fornecer informações textuais adicionais para o leitor de um diagrama BPMN. Também não influencia no fluxo de sequência.	

**Figura 3.1 - Exemplo de modelagem de processos de negócio usando a notação BPMN 2.0.2. (adaptado de OMG, 2013)**



### 3.3 Estudos sobre modelagem de processos de negócio

No estudo de Cruz *et al.* (2012) é apresentada uma abordagem para obter o modelo de dados de um sistema de software a partir da modelagem de processos de negócio. Nesse caso, a modelagem de processos de negócio apoia atividades de Engenharia de Software, em especial, aquelas relacionadas ao projeto (*design*) de dados. Assim, no contexto deste trabalho, observa-se a importância de repositórios de LPN permitirem a integração com outras ferramentas, tanto de modelagem de processos de negócio como de desenvolvimento de software.

Türetken *et al.* (2011) propõem um método de modelagem de processo de negócio descentralizado na qual cada executor do processo tem a responsabilidade de descrever e melhorar seus processos sem qualquer estrutura centralizada, visando a construção e a manutenção do processo base da organização de maneira

desburocratizada. Esse método permite um comprometimento das pessoas com a melhoria dos processos. O estudo evidencia o caráter dinâmico das organizações e a necessidade dos modelos refletirem o estado das organizações, para tanto, a arquitetura de referência Cambuci-LPN deve permitir o rastreamento das mudanças que estão sendo efetuadas e as autorizações necessárias para realizar essas modificações.

Santana *et al.* (2011) investigam o uso do conceito de LPS para promover o reúso de modelos de processos de negócio por meio de uma revisão sistemática e concluem que os trabalhos encontrados não apresentam uma base sólida para a reutilização de modelos de processos de negócio. Isso mostra a importância deste trabalho para colaborar com a evolução de pesquisas na área, especialmente relacionadas ao apoio computacional no reúso de processos de negócio.

Mturi e Johansson (2013) propõem um modelo para anotar semanticamente processos de negócio armazenados em repositórios, facilitando as buscas de modelos de processos, suportando a navegação pelos modelos no repositório e proporcionando melhor entendimento dos mesmos. Isso é um requisito relevante e que é considerado na definição da arquitetura de referência Cambuci-LPN.

### **3.4 Um arcabouço para repositórios de modelos processos de negócio**

É comum organizações descreverem suas operações em termos de processos de negócio. Coleções de modelos de processos de negócio crescem e podem conter centenas e até milhares de modelos de processos de negócio. O gerenciamento de uma grande quantidade de processos é uma tarefa complexa que lida com algumas questões como: encontrar o processo de negócios mais adequado em uma coleção de processos, em conformidade com as necessidades dos interessados; gerenciar diferentes versões de processos de negócio e manter a consistência do conteúdo quando múltiplas pessoas estão editando o mesmo processo de negócio ao mesmo tempo (YAN *et al.*, 2012).

Repositórios que apoiam a inserção, a busca, a seleção e a gestão de ativos reutilizáveis, como são o caso de requisitos, modelos de projeto, código fonte e teste, podem ser estendidos para armazenar modelos de processos de negócio (YAN *et al.*, 2012). Esse tipo de repositório é denominado repositório de modelos de processos de negócio, mencionados nesta seção por repositórios de MPN.

Repositórios de MPN têm sido propostos e desenvolvidos (YAN *et al.*, 2012) e além das funcionalidades de um repositório tradicional, tais repositórios devem atender funcionalidades específicas de processos de negócio, como por exemplo,

auxiliar no gerenciamento do ciclo de vida dos processos de negócio, ajudar a manter a consistência entre os processos privados e os processos públicos de uma organização e auxiliar no gerenciamento de configuração dos processos de negócio que são compostos por (versões de) subprocessos e tarefas (YAN *et al.*, 2012).

Yan *et al.* (2012) realizaram uma análise de 20 repositórios de MPN atuais e definiram um arcabouço para repositórios de modelos de processos de negócio, composto por um modelo de gerenciamento e por uma arquitetura de referência. Dentre os repositórios analisados, dois deles permitem o armazenamento de arquiteturas de referências de processos de negócio (FIORINI *et al.*, 2001; THELING *et al.*, 2005), outros dois permitem o armazenamento de modelos de referência de processos de negócio (ELHADAD *et al.*, 2008; WANG e WU, 2011) e um deles permite o armazenamento de variantes de um processo (LU *et al.*, 2009). Esses são os repositórios analisados por Yan *et al.* (2012) e mais intrinsecamente relacionados aos repositórios de LPN, porém, o conceito de LPN não é considerado em nenhum dos repositórios estudados, o que reforça a relevância do atual trabalho.

O modelo de gerenciamento de repositórios de MPN do arcabouço de Yan *et al.* (2012) apresenta todas as funcionalidades comumente encontradas nos repositórios analisados, e a arquitetura de referência de repositórios de MPN apresenta os componentes que fornecem tais funcionalidades. Neste trabalho, essa arquitetura de referência será estendida para o contexto de repositórios de LPN.

A arquitetura de referência de Yan *et al.* (2012) é baseada no modelo de gerenciamento, por isso é apresentada a seguir uma breve descrição dos elementos que compõem esse modelo.

O modelo de gerenciamento é composto por três partes: modelo de dados de processo de negócio, modelo de função de processo de negócio e modelo de gerenciamento de processos de negócio, as quais estão descritas a seguir:

- ***Modelo de dados de processo de negócio***

Estabelece os tipos de modelos de processos de negócio e dados relacionados que podem ser armazenados em um repositório de MPN, sendo composto pelo metamodelo, pela apresentação, pelo modelo de armazenamento e pelo modelo de índice dos dados do processo de negócio.

O **metamodelo de dados de processo de negócio** prescreve as informações que podem e as que devem ser armazenadas no repositório, definindo os conceitos que serão usados no repositório e as relações entre esses conceitos. Essas informações foram

classificadas por Yan *et al.* (2012) em aspectos do processo de negócios e tipos de artefatos do processo de negócios.

Os aspectos de processos de negócio são elencados a seguir:

- atividade: realizada no processo de negócio;
- fluxo de Controle: relacionamento entre as atividades do processo de negócio e as possíveis sequências de execução das mesmas;
- dados: informação que é usada e alterada durante a execução de um processo de negócio;
- recurso: recursos físicos que são necessários para executar um processo de negócio, incluindo os recursos humanos;
- autorização: descreve quem é autorizado a realizar qual parte do processo de negócio;
- organização: descreve a estrutura organizacional, que consiste em pessoas e unidades organizacionais, relacionada a uma coleção de processos de negócio;
- objetivo: descreve a hierarquia de objetivos estratégicos, o relacionamento entre tais objetivos e os processos de negócio que devem ser executados para atingi-los;
- relacionamento: representa a relação entre processos de negócio e outros processos de negócio, como por exemplo, processos e subprocessos, ou seja, processos que se comunicam por meio de troca de mensagens;
- semântica: descreve semanticamente conceitos para processos de negócio;
- monitoramento: consiste de conceitos para definir como o desempenho de um processo de negócios poderia ser monitorado;
- controle de gerenciamento: contém conceitos para definir controles que são implementados por processos de negócio, como o acompanhamento dos indicadores de processos de negócio.

Os tipos de artefatos de processo de negócio, definidos no modelo de dados, são elencados a seguir:

- processo de negócio específico da empresa: representa as operações da própria empresa;
- processo de negócio de referência: é um processo de negócio padrão e abstrato que pode ser reusado e adaptado para desenvolver processos de negócio específicos da empresa;
- padrão de processo de negócio: é uma parte de um processo de negócio que descreve uma melhor prática sumarizada, a partir de experiências anteriores;
- instância de processo de negócio: refere-se a uma execução de um processo de negócio para um cliente, ou seja, uma instância de um processo sendo executada.
- informação histórica: são registros com informações sobre execuções de instâncias de processos de negócio.

A **apresentação do modelo de dados** estabelece como as informações armazenadas no repositório são exibidas para o usuário final. Tais informações podem ser exibidas, por exemplo, de maneira textual ou por meio de notações específicas, como é o caso da notação BPMN, de interesse deste trabalho porque é um padrão da OMG e está sendo utilizada por ferramentas atuais que apoiam a modelagem de negócios.

O **modelo de armazenamento** define como a informação original sobre o processo de negócio precisa ser tecnicamente fornecida para o repositório (*modelo de dados externo*) e como precisa ser armazenada internamente pelo repositório (*modelo de dados interno*).

O **modelo de índice** dos dados do processo de negócio prescreve os índices que devem ser mantidos para os modelos de processos de negócio, visando permitir que os usuários do repositório possam rapidamente navegar ou pesquisar na coleção de processos de negócio. Uma aplicação prática dos índices está na classificação que os analistas de negócios fazem para os processos de negócio, como por exemplo, processos de finanças, processos de vendas, processos de suporte, etc.

- **Modelo de função de processo de negócio**

Prescreve as diversas funções que um repositório de MPN deve suportar para manipular de maneira efetiva os processos de negócio que ele armazena. O arcabouço estabelece as funções de armazenamento, recuperação e integração de processos de negócio.

Dentre as **funções de armazenamento**, estão funções para criar, atualizar e remover processos de negócio ou partes de processos de negócio, além das funções para importar e exportar processos de negócio para e do repositório, respectivamente.

As **funções de recuperação** que devem estar presentes em repositórios de MPN são: navegação, que consiste em examinar manualmente a coleção de processos de negócio disponível visando selecionar o mais adequado; busca simples, na qual deve prover um mecanismo para recuperar processos de negócio que possuam informações que condizem com os critérios informados; e busca avançada, que consiste na busca avançada por meio de linguagens de consultas específicas do repositório.

As **funções de integração** possibilitam que o repositório se comunique com ferramentas externas, como por exemplo, ferramentas de modelagem de processos de negócio, por exemplo, Bizagi (BIZAGI, 2013), e ambientes de apoio ao desenvolvimento, por exemplo, Eclipse (ECLIPSE, 2013) e Netbeans (ORACLE, 2013).

- **Modelo de gerenciamento de processo de negócio**

Estabelece as funções avançadas de gerenciamento do repositório, que são classificadas em funções gerais e funções específicas.

As **funções gerais** são: gerenciamento de acesso, gerenciamento de integridade, gerenciamento de transação, gerenciamento de *check-in/check-out*, gerenciamento de despacho, gerenciamento de notificação das mudanças que ocorrem nos processos de negócio do repositório e gerenciamento de contextos.

As **funções específicas** para o gerenciamento de processos de negócio são: gerenciamento de versão, gerenciamento de configuração, gerenciamento de visão e gerenciamento de ciclo de vida. Gerenciamentos de versão, de configuração e de visão fazem parte das funções de repositórios gerais, porém, foram especializadas para atender requisitos específicos de processos de negócio (AALST *et al.*, 2003, CHOI *et al.*, 2003; LAPOUCHNIAN e MYLOPOULOS, 2007; ZHAO e LIU, 2007).

No Quadro 3.6 é apresentado um sumário do modelo de gerenciamento de repositório de MPN com as funções que um repositório de MPN deve contemplar, de acordo com Yan *et al* (2012), e que foram discutidas nesta seção.

A arquitetura de referência do arcabouço de Yan *et al.* (2012) possui alto nível de abstração e é representada apenas na visão geral, conforme ilustrado na Figura 3.2. Apesar de outras visões de arquiteturas de referência não serem apresentadas, através do modelo de gerenciamento de repositório da MPN pode-se verificar que a visão lógica corresponde ao modelo de dados dos processos, e que a visão de processo corresponde aos modelos de função de processos e de gerenciamento de processos, ambos estão descritos no modelo de gerenciamento da MPN.

**Quadro 3.6 - Modelo de gerenciamento de repositórios de MPN (adaptado de Yan *et al.*, 2012)**

Modelo de gerenciamento de repositório de MPN		
Modelo de dados de processo de negócio	Meta modelo de processo	Aspecto de processo
		Tipos de artefatos de processo
	Apresentação de processo	Notação do processo
	Modelo de armazenamento de processo	Modelo externo de dados de processo
		Modelo interno de dados de processo
		Modelo de dados relacionados ao processo
	Modelo de índices de processo	Classificações de processos
Outros índices		
Modelo de função de processo de negócio	Funções de armazenamento	Criar
		Remover
		Atualizar
		Importar
		Exportar
	Funções de recuperação	Navegação
		Busca <i>Query</i>
Funções de integração	[dependente de ferramentas externas]	
Modelo de gerenciamento de processo de negócio	Funções específicas	Gerenciamento de versão
		Gerenciamento de configuração
		Gerenciamento de ciclo de vida
		Gerenciamento de visão de processo
	Funções gerais	Gerenciamento de acesso
		Gerenciamento de integridade
		Gerenciamento de transação
		Gerenciamento de check-in/out
		Gerenciamento de despacho (fluxo de trabalho na especificação do ativo)
		Gerenciamento de notificação
		Gerenciamento de contexto (permite a criação e manipulação de coleções de ativos no repositório).

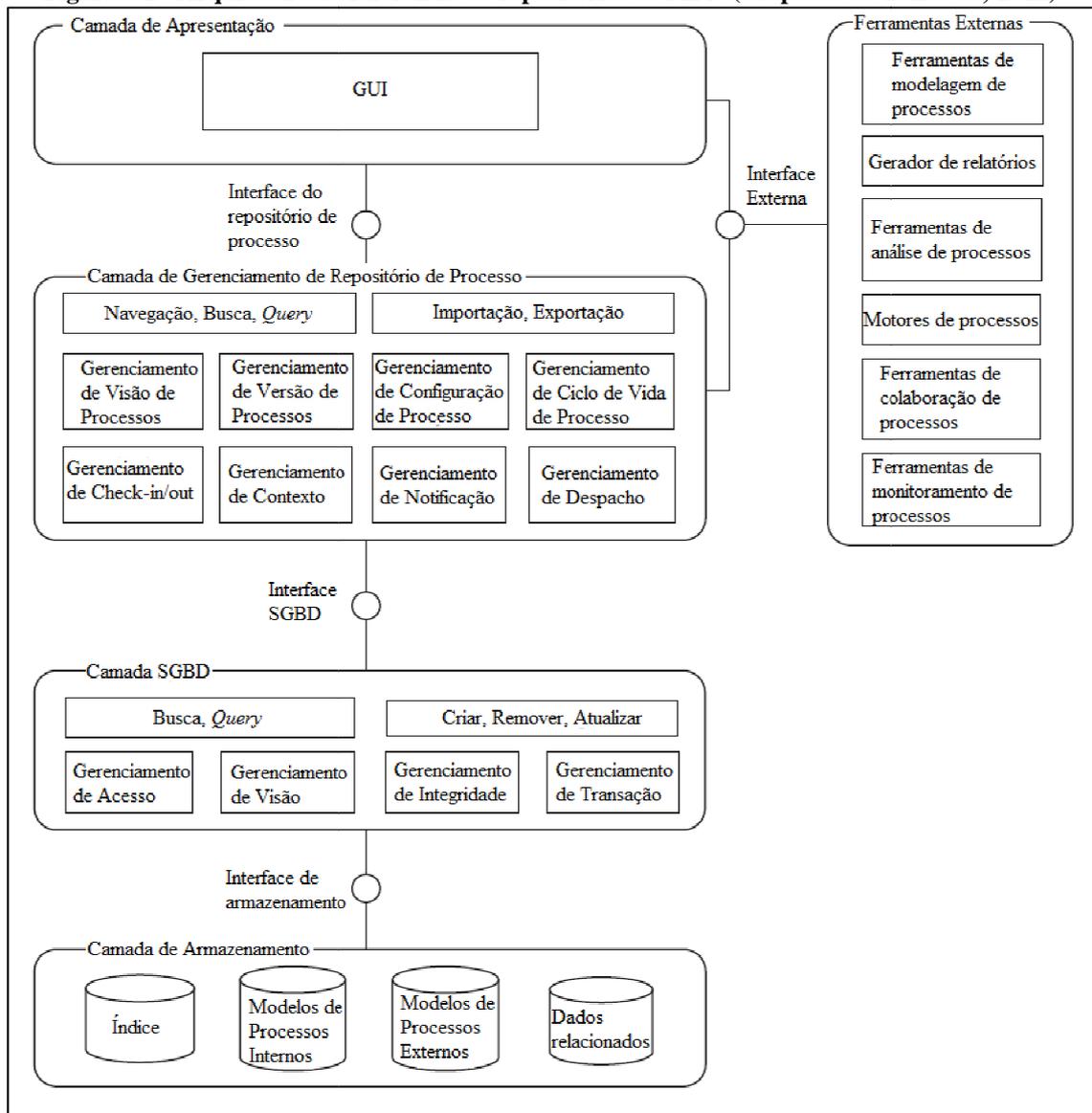
Yan *et.al* (2012) não menciona o processo de definição da arquitetura utilizado, mas apresenta a concepção da arquitetura por meio de três passos. No primeiro passo, iniciou-se com uma arquitetura em três camadas (**apresentação, lógica,**

**armazenamento**) porque muitos dos repositórios estudados utilizavam essa arquitetura. No segundo passo, dividiu-se a camada lógica em camada de **gerenciamento de banco de dados** e **camada de gerenciamento de repositório**.

Finalizando, no terceiro passo foram adicionadas as funcionalidades específicas de repositórios de processos para a camada de gerenciamento de repositório, e renomeou essa camada para **camada de gerenciamento de repositório de processos de negócio**. Depois adicionou as **Ferramentas Externas** como componente separado porque foi verificada em outros estudos a interação com outras ferramentas.

Assim, a arquitetura de referência de Yan *et al.* (2012) é formada por quatro camadas, conectadas por meio de interfaces bem definidas: camada de apresentação, camada de gerenciamento de repositório de processos de negócio, camada de gerenciamento de banco de dados e camada de armazenamento.

**Figura 3.2 - Arquitetura de referência de repositórios de MPN (adaptado de Yan *et al.*, 2012)**



A camada de apresentação é responsável por prover uma interface gráfica amigável na qual os usuários possam interagir com o repositório e com as suas funções.

A camada de gerenciamento é responsável por implementar as funções de recuperação, as funções específicas de gerenciamento de processos de negócio, bem como, as seguintes funções gerais de gerenciamento de repositórios: gerenciamento de *check-in/check-out*, gerenciamento de despacho, gerenciamento de notificação e gerenciamento de contexto. Além disso, as funções de importar e exportar processos de negócio também são implementadas nessa camada. As demais funções de armazenamento e de gerenciamento de repositórios, em geral, são implementadas na camada de gerenciamento de banco de dados (Camada SGBD).

Na camada de armazenamento todos os dados relacionados aos processos de negócio são armazenados, ou seja, o tratamento do modelo de dados de processo de negócio é responsabilidade dessa camada.

### **3.5 ONTO-ResAsset: ontologia de especificação e gestão de ativos reutilizáveis**

A ontologia ONTO-ResAsset (SILVA, 2013) contempla todos os elementos necessários para compor o metadado de um ativo, que pode ser um requisito, um modelo de análise ou de projeto, um documento de teste, um modelo de processo de negócio, dentre outros; bem como a gestão do reuso de um ativo, que pode ser apoiada computacionalmente por repositórios.

Para a identificação dos conceitos da ONTO-ResAsset foram consideradas diversas fontes de informação, das quais as mais importantes são a especificação de ativos reutilizáveis (*Reusable Asset Specification - RAS*) da OMG (2005) e o Processo de Gestão de Reuso de Ativos da ISO/IEC 12207 (2009).

RAS possui recomendações sobre a estrutura, definição, conteúdo e descrição de ativos reutilizáveis, que são importantes para compor o metadado dos elementos de um repositório. O processo de gestão de reuso de ativos tem por finalidade gerenciar o ciclo de vida de ativos reusáveis desde sua concepção até sua desativação, possuindo um conjunto de atividades que devem ser contempladas por um repositório de ativos reusáveis.

Segundo Silva (2013), a ONTO-ResAsset foi avaliada por 22 avaliadores (sendo dois especialistas na área de reuso de software e 20 não especialistas - nove

alunos de pós-graduação da Facom e onze colaboradores, que fazem parte de uma equipe de desenvolvimento de software de uma empresa real) no segundo semestre de 2013. A avaliação por especialistas foi feita por meio de um questionário composto por seis questões e a avaliação por não especialistas foi realizada através de um questionário composto por 31 questões estabelecidas durante a especificação da ontologia. Como resultado observou-se que a ontologia é capaz de representar o conhecimento sobre a especificação e a gestão de ativos reutilizáveis e que pode apoiar na definição de critérios para seleção de repositórios, bem como na definição de requisitos e da estrutura dos mesmos.

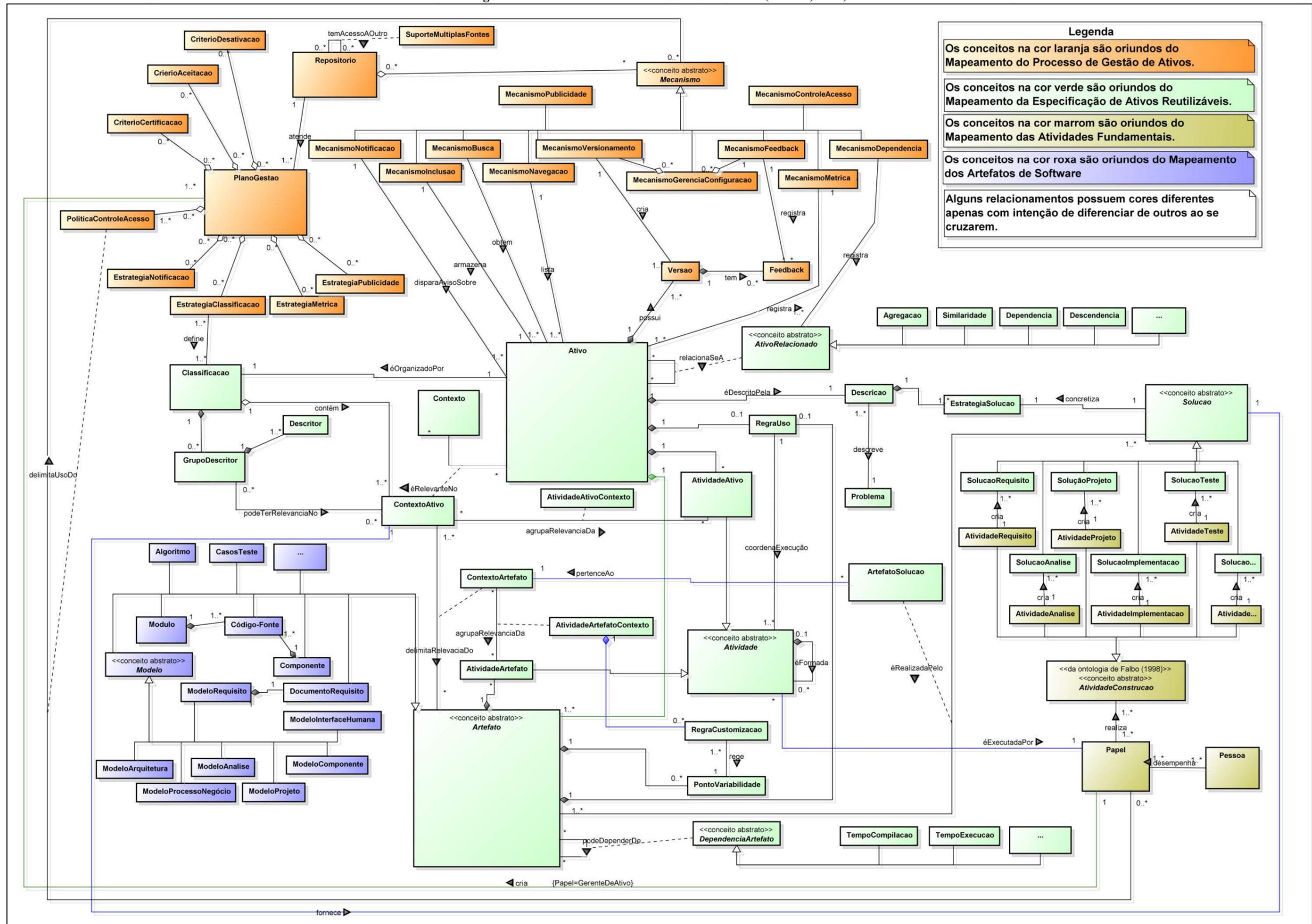
Na Figura 3.3 é apresentado o modelo conceitual que ilustra todos os conceitos da ONTO-ResAsset. Os conceitos representados pela cor verde referem-se aos principais elementos de um ativo. Os conceitos na cor azul correspondem aos demais elementos que compõem a especificação de um ativo. Os conceitos laranja referem-se à gestão de ativos. Os conceitos na cor azul representam alguns exemplos de tipos de artefatos que podem ser reutilizados durante as atividades do desenvolvimento de software. Os conceitos na cor amarela representam os tipos de atividades realizadas durante o desenvolvimento de software e abstração dos mecanismos de um repositório, que proveem apoio computacional para os elementos de um plano de gestão de reuso de ativos.

### **3.6 Estudo comparativo entre o arcabouço de repositórios de modelos de processos de negócio e a ONTO-ResAsset**

Para averiguar a completude do arcabouço de repositórios de modelos de processos de negócio, proposto por Yan *et al.* (2012), foi realizada uma análise comparativa dos elementos do modelo de gerenciamento desse arcabouço, no qual a arquitetura de referência proposta por Yan *et al.* (2012) foi baseada, com os conceitos da ONTO-ResAsset.

Conceitos como *AtivoRelacionado*, *GrupoDescritor*, *Descritor*, *RegraUso*, *RegraCustomização*, *Solução*, *Descrição*, *AtividadeArtefatoContexto*, *AtividadeAtivoContexto* não foram encontrados na proposta de Yan *et al.* (2012), devido ao alto grau de abstração de sua arquitetura de referência, mas esses conceitos devem estar presentes na arquitetura de referência Cambuci-LPN.

Figura 3.3 - Modelo conceitual da ONTO-ResAsset (SILVA, 2013)



No Quadro 3.7 é apresentada uma comparação entre os conceitos da ONTO-ResAsset, referentes a especificação de ativos reutilizáveis, com o modelo de dados de processo de negócio do arcabouço de Yan *et al.* (2012), que se refere à especificação do processo de negócio em repositórios MPN.

No repositório proposto por Yan *et al.* (2012) pode-se verificar que não existe uma distinção muito clara entre artefatos e ativos do repositório, ambos são tratados como artefatos. O conceito *PontoVariabilidade* não foi encontrado mas é de grande importância para a arquitetura de referência Cambuci-LPN.

Conceitos como *AtivoRelacionado*, *GrupoDescritor*, *Descritor*, *RegraUso*, *RegraCustomização*, *Solução*, *Descrição*, *AtividadeArtefatoContexto*, *AtividadeAtivoContexto* não foram encontrados na proposta de Yan *et al.* (2012), devido ao alto grau de abstração de sua arquitetura de referência, mas esses conceitos devem estar presentes na arquitetura de referência Cambuci-LPN.

**Quadro 3.7 - Comparação entre ONTO-ResAsset e arcabouço de repositórios de MPN (Parte 1)**

Conceito da ONTO-ResAsset	Descrição Semântica (SILVA, 2013)	Modelo de dados de processo de negócio (YAN <i>et al.</i> , 2012)
Artefato	Um artefato é qualquer produto que pode ser criado, armazenado e manipulado por produtores e consumidores ou por uma ferramenta, dentro de um processo de desenvolvimento de software.	Processos específicos de uma companhia, processos de referência, padrões de processos, instância de processos, informações históricas
Atividade	É a atividade que o consumidor ou ferramenta deve realizar quando utiliza o ativo. Uma atividade pode ser composta por outras atividades e cada atividade deve ser desempenhada por um papel.	Atividades que o usuário pode realizar quando utilizar um ativo. Pode ser atividades de armazenamento (criar, deletar, atualizar), busca, e de gerenciamento (controle de versão, gerenciamento de configuração, gerenciamento de ciclo de vida)
AtividadeArtefato	Determina as atividades associadas a um artefato específico, podem existir diversas atividades para um mesmo artefato, assim como a atividade do artefato pode estar relacionada a um ou mais contextos do artefato.	Atividades associadas a um artefato específico
AtividadeAtivo	Representa a atividade relacionada ao ativo como um todo, podendo ser agrupada a outras atividades de ativo.	Mesmo que AtividadeArtefato
Ativo	É o elemento principal da RAS; ele descreve e empacota os artefatos e provê a documentação sobre como e qual solução ele resolve.	Mesmo que artefato

**Quadro 3.7 (continuação)**

<b>Conceito da ONTO-ResAsset</b>	<b>Descrição Semântica (SILVA, 2013)</b>	<b>Modelo de dados de processo de negócio (YAN <i>et al.</i>, 2012)</b>
AtivoRelacionado	Um ativo pode estar relacionado a outros ativos. Existem quatro tipos de relacionamentos: agregação, similaridade, dependência, generalização.	Não descrito.
Classificação	Classificam as características e comportamentos importantes dos ativos, por exemplo, domínio, autores, palavras-chave, etc. Contém grupos de descritores, que podem pertencer a contextos.	Possui o modelo de índice que são mantidos para o modelo de processos.
Contexto	O contexto esclarece onde um ativo é relevante. Por exemplo, um modelo de processo de negócios pertence ao contexto “Negócios” e também ao contexto “Documentação”. Isso indica que esse ativo é relevante para o contexto específico de negócios e também colabora para a documentação do sistema.	Permite a criação e manipulação de contexto para os processos
GrupoDescritor	Representa um agrupamento de descritores para ajudar a definir o contexto do ativo.	Não descrito
Descritor	Descreve os comportamentos e as características do ativo.	Não descrito
PontoVariabilidade	Determinam os pontos (locais) nos artefatos que pode ocorrer uma variabilidade, ou seja, que pode ser feita uma customização para um contexto específico.	Não descrito
RegraUso	Representa a informação sobre como aplicar ou usar um ativo ou artefato. Essas instruções podem ser para todo o ativo ou para um artefato específico do ativo; são informações que restringem e definem também como as atividades dos artefatos e dos ativos serão executadas.	Não descrito
RegraCustomização	Representa as regras, restrições e orientações para que o consumidor do ativo possa personalizar o ponto de variabilidade do artefato. Isso é feito durante a execução de atividades do artefato.	Não descrito
Solução	Representa a solução fornecida pelo ativo, por meio da coleção de artefatos que o compõe. A solução é fornecida pelo(s) artefato(s) de acordo com o contexto do ativo que o artefato pertence. Existem alguns tipos de solução de acordo com a atividade que cria a solução: requisitos, análise, projeto, implementação e teste.	Não descrito

**Quadro 3.7 (conclusão)**

<b>Conceito da ONTO-ResAsset</b>	<b>Descrição Semântica (SILVA, 2013)</b>	<b>Modelo de dados de processo de negócio (YAN <i>et al.</i>, 2012)</b>
Descrição	Descreve o problema tratado pelo ativo e as suas principais estratégias de solução.	Não descrito
ContextoArtefato	Representa o contexto do ativo em que o artefato pode ser utilizado.	Sim. Permite criar e manipular contextos para os processos
DependênciaArtefato	Um artefato pode depender de outros artefatos. Como exemplos de dependência entre artefatos têm-se: dependência em tempo de compilação, dependência em tempo de execução.	Não descrito
AtividadeArtefatoContexto	Representa a atividade do artefato em um contexto específico.	Não descrito
AtividadeAtivoContexto	Representa a atividade do ativo em um contexto específico.	Não descrito

No Quadro 3.8 é apresentada uma comparação entre os conceitos da ONTO-ResAsset, referentes a gestão de ativos reutilizáveis, com o modelo de função e o modelo de gerenciamento de processo de negócio do arcabouço de Yan *et al.* (2012), que se referem à gestão de processos de negócio em repositórios MPN.

**Quadro 3.8 - Comparação entre ONTO-ResAsset e arcabouço de repositórios de MPN (Parte 2)**

<b>Conceito da ONTO-ResAsset</b>	<b>Descrição Semântica (SILVA, 2013)</b>	<b>Modelo de função e o modelo de gerenciamento de processo de negócio (YAN <i>et al.</i>, 2012)</b>
PlanoGestão	Criado pelo gerente de ativos e é onde se definem os recursos e os procedimentos que serão realizados para a gestão do ativo reutilizável. Esse plano deve ser executado pelo gerente de ativos e ser revisado levando em consideração o processo de revisão de software ISO/IEC 12207 (2009). No plano de gestão estão descritas todas as informações (critérios, políticas e procedimentos) para a realização da gestão de ativos.	Não descrito
CritérioAceitação	São os critérios definidos no plano de gestão para que um ativo seja aceito ou não pelo gerente de ativos, o qual é responsável pela gestão dos ativos.	Não descrito
CritérioCertificação	São os critérios definidos no plano de gestão para que o ativo seja certificado ou não pelo gerente de ativos, ou seja, garantem que o ativo esteja de acordo com a qualidade esperada.	Não descrito

**Quadro 3.8 (continuação)**

Conceito da ONTO-ResAsset	Descrição Semântica (SILVA, 2013)	Modelo de função e o modelo de gerenciamento de processo de negócio (YAN <i>et al.</i> , 2012)
CritérioDesativação	São os critérios definidos no plano de gestão para que os ativos sejam descartados.	Não descrito
EstratégiaClassificação	É um documento, criado pelo gerente de ativos, que compõe o plano de gestão e tem por objetivo definir a estratégia para a classificação dos ativos.	Não descrito
EstratégiaNotificação	Definida no plano de gestão e determina quais notificações serão enviadas para os usuários do repositório sobre alterações que ocorrerem nos ativos.	Realizado pela atividade de gerenciamento de notificação
MecanismoBusca	É o mecanismo que permite a busca e a recuperação dos ativos no repositório. Existem algumas formas para realizar as buscas, como por exemplo, por meio de palavras-chave, busca textual que pode ser realizada levando em consideração os elementos que compõem a especificação do ativo, ou mesmo por facetas que faz uso de uma combinação de pares (atributo, valor).	Realizado pela atividade de navegação, busca e <i>queries</i> .
MecanismoDependência	É o mecanismo que permite o registro de dependência entre ativos. Essa dependência pode ser: agregação, similaridade, dependência ou origem.	Realizado pela atividade de gerenciamento de configuração de processos
MecanismoFeedback	É o mecanismo que permite o registro de impressões dos usuários a respeito da versão do ativo que eles utilizaram.	Não descrito
MecanismoGerenciaConfiguração	É o recurso que permite a realização dos mecanismos de <i>feedback</i> e de versionamento com objetivo de executar a gerência de configuração do ativo.	Realizado pela atividade de gerenciamento de configuração de processos
MecanismoInclusão	É o mecanismo de armazenamento que permite a inclusão de ativos no repositório.	Realizado pela atividade de gerenciamento do ciclo de vida dos processos
MecanismoMétrica	É o mecanismo do repositório que registra as métricas coletadas sobre a utilização do ativo.	Não descrito
MecanismoNavegação	É o mecanismo que, a partir de algum grupo de descritores da classificação do ativo, ou de algum método de agrupamento ou listagem simples, permite a navegação entre os ativos armazenados.	Realizado pela atividade de navegação, busca e <i>queries</i>

**Quadro 3.8 (conclusão)**

Conceito da ONTO-ResAsset	Descrição Semântica (SILVA, 2013)	Modelo de função e o modelo de gerenciamento de processo de negócio (YAN <i>et al.</i> , 2012)
MecanismoPublicidade	É o mecanismo que oferece informações relativas ao reúso, iniciativas de reúso, ativos mais usados, etc.	Não descrito
MecanismoVersionamento	É o mecanismo que permite a criação e o armazenamento de múltiplas versões de um mesmo ativo.	Realizado pela atividade de gerenciamento de versões
PolíticaControleAcesso	Determina o acesso aos mecanismos oferecidos pelo repositório, de acordo com o papel que o usuário assume.	Realizado pela atividade de gerenciamento de acessos
Repositório	Representa o conceito de um armazém de ativos, que é responsável por prover um conjunto de mecanismos para a gestão e uso dos ativos.	Representa o conceito de um armazém de modelos de processos de negócio
SuporteMultiplasFontes	É a capacidade que um repositório deve ter de ser capaz de aceitar múltiplas fontes de origem de ativos, com objetivo de facilitar a integração entre equipes e entre repositórios diferentes.	Sim. Permite a integração com ferramentas externas

A arquitetura proposta por Yan *et al.* (2012) não possui um plano de gerenciamento dos ativos para realizar a sua implantação também devido ao seu alto nível de abstração. Por isso, vários documentos que fazem parte do plano de gerenciamento dos ativos não estão presentes, por exemplo, o plano de gestão, os critérios de aceitação, certificação e desativação, e as estratégias de classificação e notificação. Esses documentos são considerados como conceitos durante a concepção da arquitetura de referência Cambuci-LPN.

Os mecanismos de *feedback*, métrica e publicidade não foram descritos na proposta de Yan *et al.* (2012) e também são contemplados pela arquitetura de referência proposta neste estudo.

### 3.7 Considerações finais

Neste capítulo foram apresentados os conceitos sobre modelagem de processos de negócio e também foi discutido sobre a necessidade de apoio computacional por meio de repositórios de MPN para gerenciar adequadamente os modelos de processos de negócio das empresas.

Nesse contexto, Yan *et al.* (2012) apresentam um arcabouço de repositórios de MPN que provê uma arquitetura de referência de modelos de processos de negócio. Essa arquitetura de referência possui algumas deficiências que são cobertas pela arquitetura de referência Cambuci-LPN. Como deficiências observadas tem-se que a arquitetura de referência de Yan *et al.* (2012) possui alto nível de abstração pois é representada em apenas uma visão, conseqüentemente, provê pouco detalhamento dificultando o seu entendimento. Isso pode ser decorrente do modo como foi construída, ou seja, os autores não seguiram um processo de desenvolvimento de arquiteturas de referência e não apresentam se a arquitetura proposta foi validada e nem como foi validada.

Além disso, a arquitetura de referência de Yan *et al.* (2012), diferentemente da ontologia ONTO-ResAsset, não cobre diversos elementos importantes de especificação de ativos, como *PontoVariabilidade*, *RegraUso*, *RegraCustomização*, *Solução*, *Descrição*, e nem funções importantes de gestão de ativos, como o *feedback*, métricas e publicidade. Também não possui o plano de gerenciamento dos ativos onde é especificado o plano de gestão, os critérios de aceitação, certificação e desativação, e as estratégias de classificação e notificação.

Outro ponto que merece ser discutido neste momento refere-se às funcionalidades consideradas pelas arquiteturas de referência para repositórios de ativos de software encontradas na literatura (Seção 2.6), as quais estão representadas na ontologia ONTO-ResAsset. Portanto, devido à completude dessa ontologia na especificação e gestão de ativos reutilizáveis, como é o caso de LPN, ela é utilizada como fonte de informação para a identificação das funcionalidades contempladas pela arquitetura de referência Cambuci-LPN. Adicionalmente, a notação BPMN é considerada como fonte de informação da arquitetura de referência proposta por ser atualmente a representação mais utilizada para a modelagem de processos de negócio (CHINOSI e TROMBETTA (2011); JURIC e PANT (2008)), além de ser a notação padrão da *Object Management Group* (OMG) para a representação de processos de negócio.

## Capítulo 4 - Linhas de processos de negócio

### 4.1 Considerações iniciais

Este capítulo apresenta o embasamento teórico sobre LPN, necessário para o entendimento e para o desenvolvimento deste trabalho. Na Seção 4.2 são apresentados os principais conceitos e definições sobre LPN, explicitando sua origem que é proveniente da LPS. Na Seção 4.3 são apresentadas algumas abordagens existentes para a construção e instanciação de LPN. Nessa seção é dada ênfase em uma das abordagens estudadas, denominada GLPN (Gestão de LPN) Landre (2012), a qual foi definida pelo grupo de pesquisa de Engenharia de Software da UFMS no qual este projeto de mestrado está sendo desenvolvido e é de interesse deste trabalho. Por fim, na Seção 4.4 são apresentadas as considerações finais do capítulo.

### 4.2 Conceitos e definições

No domínio automobilístico (POHL *et al.*, 2005), Ford inventou a linha de produção, que atendia ao mercado com produtos mais baratos do que produzidos individualmente e muitas vezes manualmente. No entanto esse tipo de produção, baseada em linha, passou a reduzir a possibilidade de diversificação do produto.

Para resolver o problema da entrega de produtos personalizados para cada cliente, com produção em massa, surgiu a customização em massa, que é a produção em larga escala de bens direcionada às necessidades individuais dos clientes (DAVIS, 2007).

No domínio do software, também existe a necessidade de produtos mais baratos do que os desenvolvidos individualmente, mas que necessitam ser personalizados para cada cliente. Para atender a essa demanda, surgiu a Engenharia de LPS (ELPS).

Os principais objetivos da ELPS são reduzir custos durante o desenvolvimento de software, melhorar a qualidade do software produzido, reduzir o tempo de colocação do produto no mercado, reduzir esforço com manutenção, propiciar capacidade de lidar com evolução e complexidade do software, melhorar a estimativa de custos, dentre outros (POHL *et al.*, 2005).

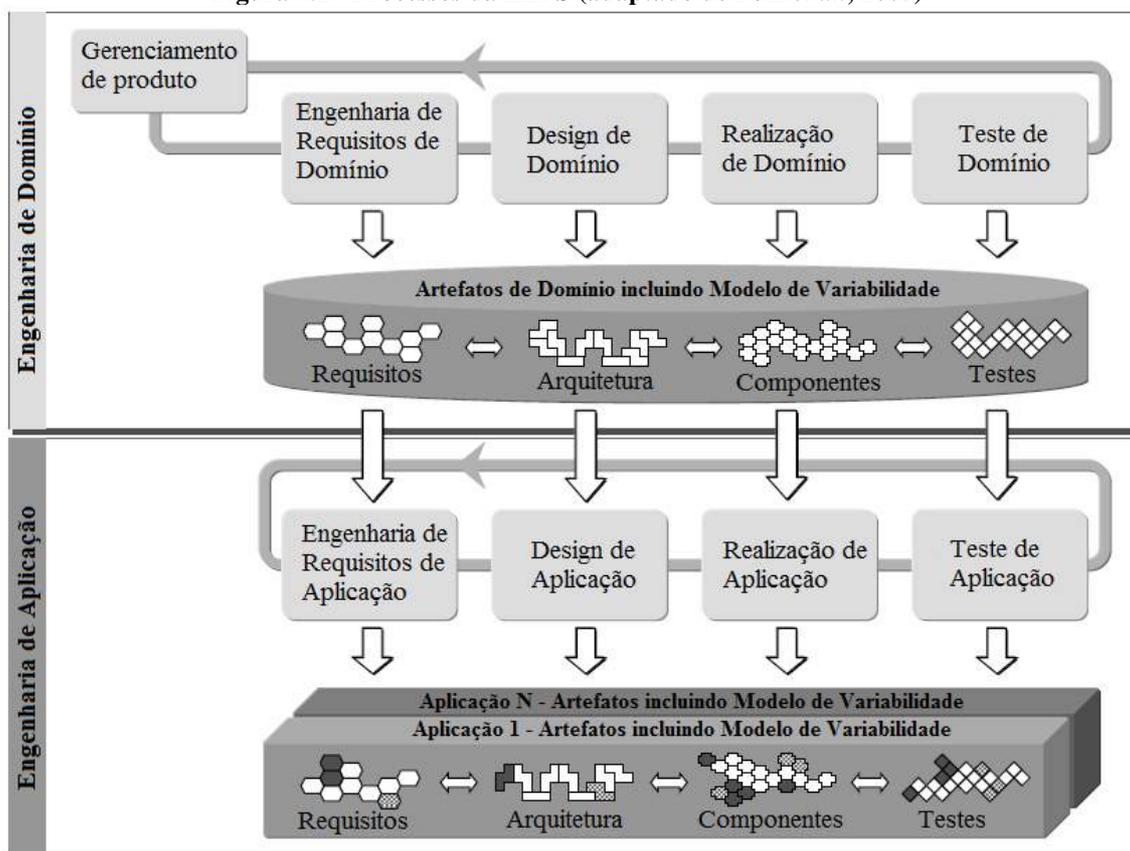
ELPS tem como base a customização em massa e o paradigma para desenvolver aplicações de software usando plataformas de software (POHL *et al.*, 2005). Sendo que plataforma de software é um conjunto de subsistemas e interfaces de

software que formam uma estrutura comum do qual um conjunto de produtos derivados podem ser eficientemente desenvolvidos e produzidos (MEYER, 1997).

O paradigma da ELPS possui dois processos principais, conforme mostrado na Figura 4.1:

- Engenharia de Domínio: processo no qual as características em comum da LPS e as suas variabilidades são definidas e realizadas; e
- Engenharia da Aplicação: processo no qual as aplicações são construídas por meio da configuração da LPS, explorando as variabilidades adequadas ao contexto da aplicação, e por meio da reutilização de artefatos do domínio.

Figura 4.1 - Processos da ELPS (adaptado de Pohl *et al.*, 2005)



Na Engenharia de Domínio, todas as etapas desse processo possuem atividades relacionadas com a identificação e documentação das variabilidades. Por exemplo, na etapa de Engenharia de Requisitos de Domínio, são identificadas e documentadas as variabilidades do Documento de Requisitos, enquanto as variabilidades dos casos de testes são identificadas e documentadas durante a etapa de

Teste de Domínio. Dessa forma, o processo Engenharia do Domínio é responsável por estabelecer uma plataforma reusável, que consiste em todos os tipos de artefatos de softwares.

A Engenharia da Aplicação é o processo da ELPS em que as aplicações, instâncias da linha de produto, são construídas. Para isso, as variabilidades definidas na Engenharia do Domínio são exploradas e os artefatos criados são reutilizados de maneira a atender as necessidades específicas de cada instância derivada a partir da plataforma estabelecida pela Engenharia do Domínio. Como resultado, esse processo possui atividades relacionadas ao levantamento de requisitos, projeto, realização e teste da aplicação a ser desenvolvida para o usuário final.

Por meio desses dois processos supracitados, conclui-se que a variabilidade é a base que permite a flexibilidade na customização em massa no contexto da LPS. A partir da plataforma definida na Engenharia de Domínio, na qual são especificadas as possíveis variabilidades do domínio, é possível criar aplicações adaptadas a diferentes realidades dos clientes.

Para identificar as variabilidades, Pohl *et al.* (2005) sugerem três perguntas:

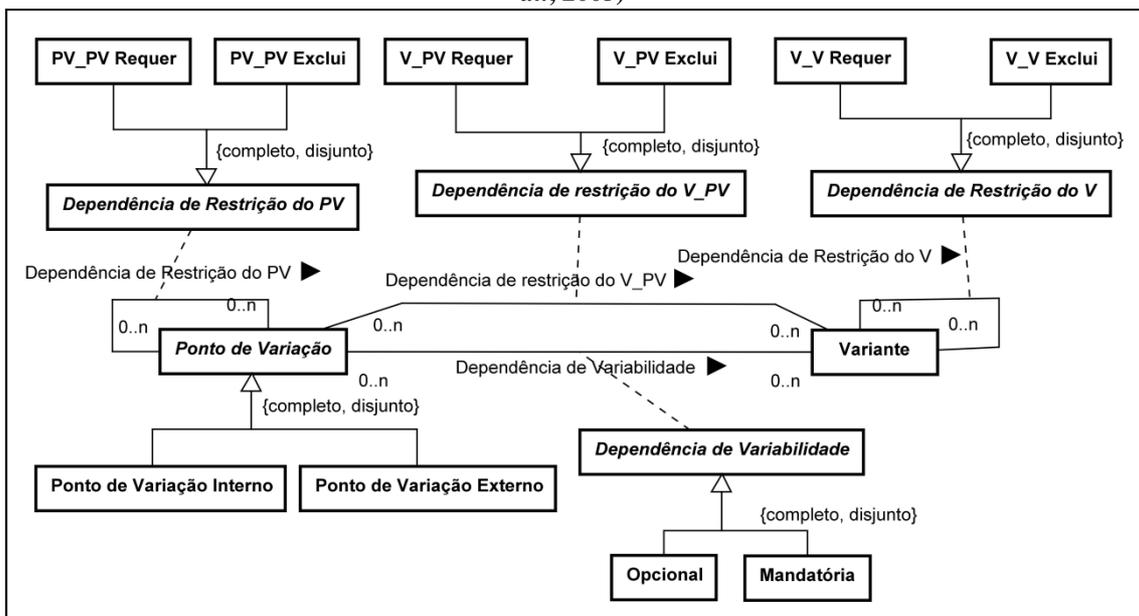
- “O que varia?” – Essa questão apoia a identificação do sujeito da variabilidade no mundo real. Por exemplo, em uma linha de produtos de carros nota-se que a cor do carro pode variar, portanto, “cor” é um sujeito da variabilidade.
- “Como varia?” – Por meio dessa questão identificam-se os objetos da variabilidade, também no mundo real. Dando continuidade ao exemplo anterior, todas as cores possíveis são consideradas objetos da variabilidade.
- “Por que varia?” – Com essa questão é possível identificar as causas da variabilidade, informação importante para decidir se a variabilidade é necessária para a linha de produto. Por exemplo, a cor é uma variabilidade importante porque permite a customização do produto para satisfazer necessidades de clientes específicos, nesse caso, clientes diferentes gostam de cores diferentes.

Uma variabilidade é descrita por um ponto de variação e suas variantes. Ponto de variação é a representação do sujeito (o que varia) da variabilidade dentro do artefato de domínio enriquecido com informações contextuais (por que varia). Variante é a representação do objeto de variabilidade (como varia) com artefatos do domínio.

Cada variante deve estar relacionada a uma realização de variante, que segundo Pohl *et al.* (2005), pode ser um componente a ser selecionado para compor o produto final.

Pohl *et al.* (2005) sugerem a utilização do modelo ortogonal de variabilidade para documentar as variabilidades, cujos elementos estão ilustrados no metamodelo da Figura 4.2. O Ponto de Variação é uma classe abstrata que é especializada em Ponto de Variação Interno (invisível para o cliente) e Ponto de Variação Externo (visível ao cliente). A classe abstrata e associativa Dependência de Variabilidade é especializada em Opcional e Mandatória. A partir desse metamodelo observa-se que cada ponto de variação oferece uma ou mais variantes e cada variante pode estar associada a um ou mais pontos de variação.

**Figura 4.2 - Metamodelo simplificado do modelo ortogonal de variabilidades (adaptado de Pohl *et al.*, 2005)**



A classe Opcional indica que uma Variante de um Ponto de Variação pode ou não ser selecionada durante a instanciação de uma LPS, isto é, a sua dependência é opcional. A classe Mandatória indica que uma Variante de um Ponto de Variação deve ser selecionada na instanciação da LPS se o Ponto de Variação é um requisito da aplicação que está sendo desenvolvida, isto é, a sua dependência é obrigatória.

A classe abstrata *Dependência de Restrição do PV* refere-se à dependência de restrição de um ponto de variação para outro ponto de variação (classe associativa), que pode ser do tipo de requisição ou de exclusão (*PV\_PV Requer*, *PV\_PV Exclui*). A

classe *Dependência de Restrição do V* refere-se à dependência de restrição de um variante para outro variante (classe associativa), que pode ser do tipo requisição ou de exclusão (*V\_V Requer*, *V\_V Exclui*). A classe *Dependência de Restrição do PV\_V* refere-se à dependência de restrição de variante para um ponto de variação (classe associativa), que pode ser do tipo requisição ou de exclusão (*V\_PV Requer*, *V\_PV Exclui*).

Desde 2003, conceitos de LPS passaram a ser utilizados com o intuito de promover o reúso na modelagem de processos negócios (ROCHA e FANTINATO, 2013), dando origem as Linhas de Processos de Negócio (LPN).

Boffoli *et al.* (2012a) definem LPN como sendo um portfólio de processos de negócio intimamente relacionados, com características, variações e contextos de operação. Esse portfólio consiste em um conjunto de semelhanças, um conjunto de variabilidades e um conjunto de regras.

Semelhanças são ativos invariantes, ou seja, são partes comuns que formam a base para todos os processos de negócio da LPN. O conjunto de variabilidades corresponde ao conjunto de ativos variáveis, que podem ser selecionados para adaptar o processo a ser derivado da LPN. O conjunto de regras que correspondem aos objetivos do negócio e explicitam a tarefa de tomada de decisão para realizar a composição flexível de ativos de processos.

A maneira como as variabilidades são documentadas influenciam na reutilização dos artefatos de uma LPS. No caso da LPN, os artefatos são os processos de negócio e com os conceitos de dependência e restrição é possível tornar as variabilidades rastreáveis, permitindo a seleção do conjunto correto de variantes de processo de negócio no momento da instanciação da LPN (LANDRE, 2012).

Conforme ressaltado por Landre (2012), as variabilidades no contexto de LPN poderão ser associadas com requisitos não funcionais ou até mesmo aos objetivos estratégicos da organização, de maneira que cada organização selecione conjuntos diferentes de variantes de processos de negócio alinhadas aos seus objetivos estratégicos.

### **4.3 Abordagens de linhas de processos de negócio**

Existem diversas abordagens que tratam da criação e da instanciação de LPNs, algumas delas são comentadas a seguir.

Rolland e Nurcan (2010) e Nunes *et al.* (2010) propõem abordagens LPN direcionadas a Gestão Por Processos, no entanto ambas as abordagens desconsideram a visão de atividades, que descreve o processo como um conjunto de atividades a serem realizadas, e o relacionamento entre elas, relacionadas entre si por fluxo pré-definidos de controle e dados. Essas abordagens consideram apenas a visão de objetivos (ROLLAND e NURCAN, 2010) e a visão de comunicação (NUNES *et al.*, 2010).

Gimenes *et al.* (2008) apresentam uma experiência de linha de produto e modelo de características<sup>1</sup> (análogo ao modelo ortogonal de variabilidade, sendo útil para a modelagem de variabilidades) (KANG *et al.*, 1990) aplicados ao contexto de modelagem de processos de negócio na área de *e-services*, direcionando o estudo ao desenvolvimento de serviços web.

Schnieders e Puhmann (2006) apresentam um processo completo de LPN direcionado aos sistemas orientados a processos, denominado PEFP (Processo de Engenharia da Família de Processos). Esse processo é composto pelo processo de engenharia da família de processos (EFP) e pelo processo engenharia da aplicação (EA), análogos respectivamente ao processo de engenharia de domínio e ao processo de engenharia da aplicação de LPS.

O processo PEFP utiliza o modelo de características para representar as variabilidades da família de processos e faz o uso do vrBPMN (SCHNIEDERS e PUHLMANN 2006), que é uma notação proposta pelos próprios autores para representar as variabilidades no próprio processo de negócio. Essa notação é uma extensão do BPMN 1.0 (WHITE, 2004) utilizando estereótipos da UML 2 e também é utilizada pela abordagem GLPN descrita mais adiante e de interesse deste trabalho.

Na Figura 4.3 é ilustrado um exemplo de processo de negócios modelado em vrBPMN, o qual explicita o ponto de variação pagamento e as respectivas variantes cartão de crédito e cartão de crédito ou dinheiro. É possível observar o uso dos estereótipos da UML 2 nessa figura. A atividade “Pagamento” representa um ponto de variação e é rotulada com o estereótipo <<Abstract>>, e suas variantes, as atividades “Cartão de Crédito” e “Cartão de Crédito ou Dinheiro”, são conectadas por meio de associações rotuladas com o estereótipo <<Implementation>>.

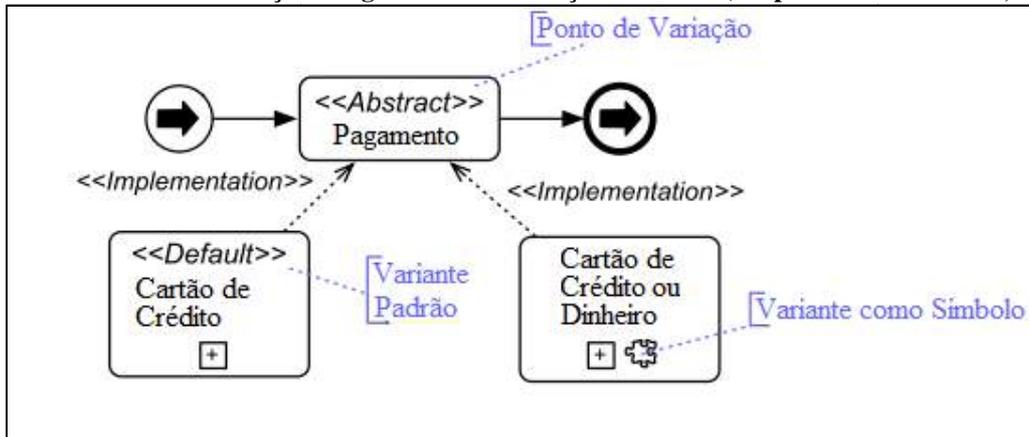
A implementação padrão pode ser representada com o estereótipo <<Default>>, auxiliando na configuração do diagrama vrBPMN para a geração de

---

<sup>1</sup> São as características de um sistema visíveis ao usuário final (KANG *et al.*, 1990).

código fonte. O símbolo de quebra-cabeça é proposto na linguagem vrBPMN para enriquecer o modelo de maneira a representar uma variante sem precisar escrever o estereótipo completo, uma alternativa a esse símbolo é declarar explicitamente o estereótipo <<Variant>> no elemento de interesse.

Figura 4.3 - Ponto de variação "Pagamento" na notação vrBPMN (adaptado de Schnieders, 2006)



Maracci (2010) apresenta um processo de ELPS como apoio para o desenvolvimento de software orientado a serviço. Observando dessa forma que a abordagem proposta não está direcionada ao gerenciamento de processos de negócio da organização, mas sim ao desenvolvimento de software.

Boffoli *et al.* (2012a) propõem a construção de uma LPN por meio de duas atividades, a Análise de Domínio e a Modelagem das Variabilidades. O conjunto de regras que compõe a LPN é documentado por meio de tabelas de decisão (SCA, 1970). Apesar da abordagem proposta por Boffoli *et al.* (2012a) não documentar explicitamente os objetivos de negócio da LPN, as tabelas de decisão fornecem um meio para a resolução das variabilidades.

A abordagem de Boffoli *et al.* (2012a) não deixa claro qual notação é adotada para a modelagem dos processos de negócio que compõem a LPN e a atividade de Análise de Domínio pressupõe que os processos de negócio já estão mapeados, sendo necessário apenas a análise de características e a construção do modelo de características, cujo objetivo é documentar as características que fazem parte do núcleo da linha, ou seja, comum a todos os processos de negócio, bem com as características que representam as variabilidades da linha, que podem ser opcional ou alternativas.

Complementar ao trabalho de Boffoli *et al.* (2012a), Boffoli *et al.* (2012b) definem Linhas de Sistemas SOA, por meio da aplicação da LPN proposta, e de DOP (Desenvolvimento Orientado a Processos).

Por meio da LPN, Boffoli *et al.* (2012b) adotam reuso de ativos e mecanismos de variação (práticas LPS) para modelar uma variante de processo adequada às necessidades de clientes específicos.

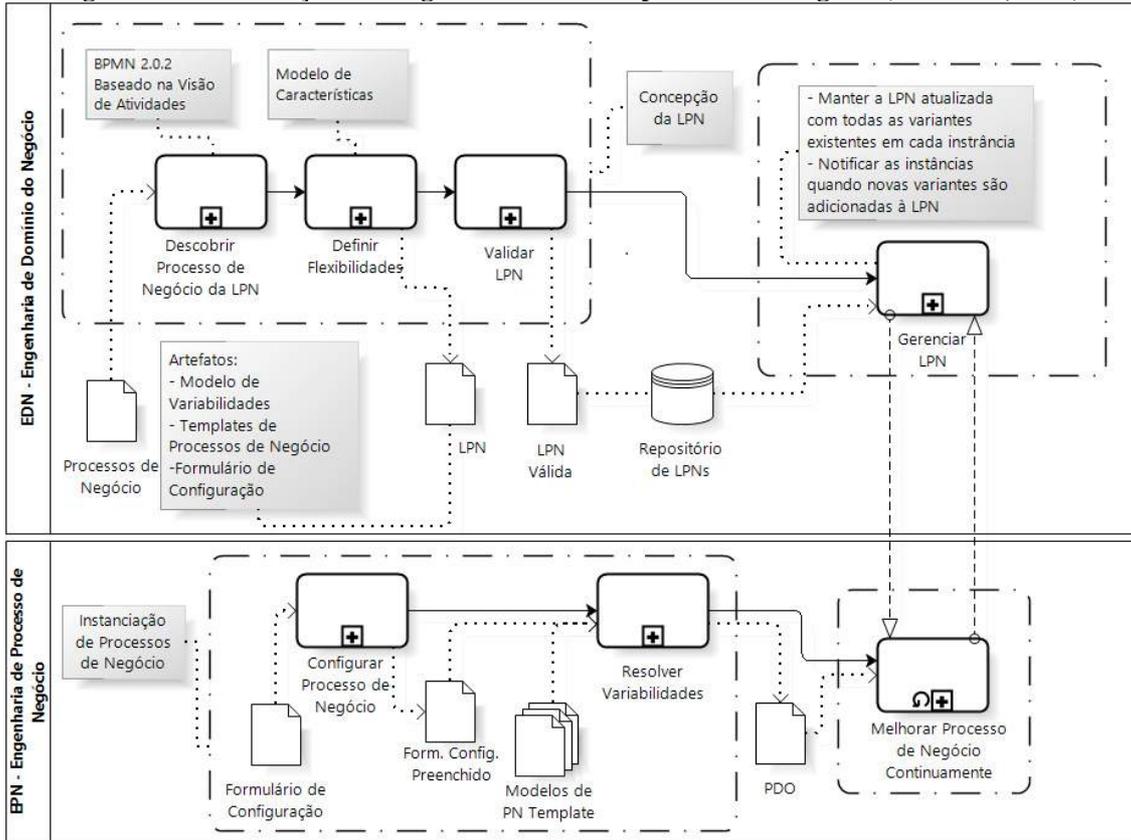
O DOP é usado para transformar a variante de processo modelada em um Sistema SOA por meio de transformações sucessivas, fazendo com que o modelo de processo seja compreensível por um mecanismo de execução.

Maracci (2010), Schnieders e Puhmann (2006) e Boffoli *et al.* (2012b) direcionam a adoção das práticas de LPN no contexto de desenvolvimento de software, ou seja, a instância da LPN é um produto de software e não um modelo de processos de negócio como proposto pela abordagem GLPN de Landre (2012). Essa última é descrita em mais detalhes a seguir porque é de interesse deste trabalho visto que a arquitetura de referência Cambuci-LPN será instanciada futuramente para a criação de um repositório de LPN para apoiar a GLPN. As demais abordagens estudadas (ROLLAND e NURCAN, 2010; e NUNES *et al.*, 2010) tem como instância um modelo de objetivos e de comunicação respectivamente, que está fora do escopo deste trabalho.

A GLPN é uma abordagem de LPN composta por dois processos: o EDN (Processo de Engenharia de Domínio do Negócio) e o EPN (Processo de Engenharia de Processos de Negócio), conforme ilustrado na Figura 4.4.

A GLPN é direcionada à Gestão por Processos e tem por finalidade apoiar o ciclo de melhoria contínua dos processos de negócio, manter e reusar modelos de todos os processos de negócio de organizações de um mesmo domínio. Para buscar tal reuso, adotou-se como base o conceito de LPS, em particular, o Arcabouço de Engenharia de LPS proposto por Pohl *et al.* (2005), adaptando-o para o contexto da gestão de processos de negócio e introduzindo o conceito de LPN de maneira que o produto final da LPN seja Processos do Domínio do Negócio (PDO), ou seja, um conjunto completo de processos de negócio de uma organização.

**Figura 4.4 - Arcabouço GLPN: gestão de linhas de processos de negócio (LANDRE, 2012)**



O processo EDN do GLPN é equivalente ao processo de Engenharia de Domínio do arcabouço de Pohl *et al.* (2005), e tem como principais objetivos a criação e a manutenção da LPN, com o foco central no Gerenciamento das Variabilidades dos Processos de negócio no contexto da visão de atividades.

O EDN é dividido nas seguintes fases: Descoberta dos Processos de Negócio, Definição de Flexibilidades, Validação da LPN e Gerenciamento da LPN. Os principais artefatos do EDN são: *Template* de Processos de Negócio e Modelo da Variabilidade.

O *Template* de Processos de Negócio é a forma de representação de cada processo de negócio que compõe um PDO coberto pela LPN.

O Modelo da Variabilidade representa as dependências de variabilidade, ou seja, como é o relacionamento entre os pontos de variação e suas variantes. Para documentar as fases do processo EDN e gerar os seus artefatos, a GLPN utiliza duas notações de modelagem, uma para a modelagem de variabilidades e outra para modelagem de processos de negócio.

A primeira notação deve conter elementos para representar as dependências de variabilidade, ou seja, como é o relacionamento entre os pontos de variação e suas variantes, por exemplo: variantes opcionais, variantes mandatórias, variantes alternativas. Com essa notação deve ser possível representar também as restrições de variabilidades, isto é, quando uma variante requer ou exclui outra variante ou ponto de variação. Para essa modelagem pode ser utilizado o modelo ortogonal de variabilidades de Pohl *et al.* (2005) ou o modelo de características de KANG *et al.* (1990), sendo que outras notações podem ser utilizadas desde que satisfaça as pré-condições estabelecidas pela GLPN.

A segunda notação deve conter elementos para representar os processos de negócio. Essa notação deve também ter elementos para assinalar os pontos de variações no modelo de processo de negócio, conectando-o com o modelo de variabilidades. Por fim, a notação deve permitir a modelagem da realização de cada variante dos pontos de variações no modelo de processo de negócio, que pode ser feita por meio da mesma notação de processos de negócio.

O vrBPMN atende parcialmente aos requisitos da notação de modelos de processo de negócio, pois como é baseado no BPMN possui os elementos para modelagem de processo de negócio. Além disso, é possível assinalar os pontos de variação e conectá-los às realizações de suas variantes diretamente no modelo de processo de negócio construído, isso se dá por meio dos estereótipos adicionados.

Entretanto, o vrBPMN não permite modelar as restrições de variabilidades e também não fornece um mecanismo para conectar as realizações de variantes ao modelo de variabilidades. A GLPN propõe então uma adaptação do vrBPMN, composta por um subconjunto de estereótipos e complementada com *tagged value's* da UML (OMG, 2005) anexadas aos estereótipos, visando documentar identificadores para os elementos, conforme ilustrado no Quadro 4.1.

Dessa forma, é possível vincular o *template* de processo de negócio ao modelo de variabilidades.

Apesar da sugestão do uso do vrBPMN outra notação pode ser utilizada no processo EDN desde que satisfaça as pré-condições estabelecidas pela GLPN.

**Quadro 4.1 - Adaptação da notação vrBPMN para o GLPN (Landre, 2012)**

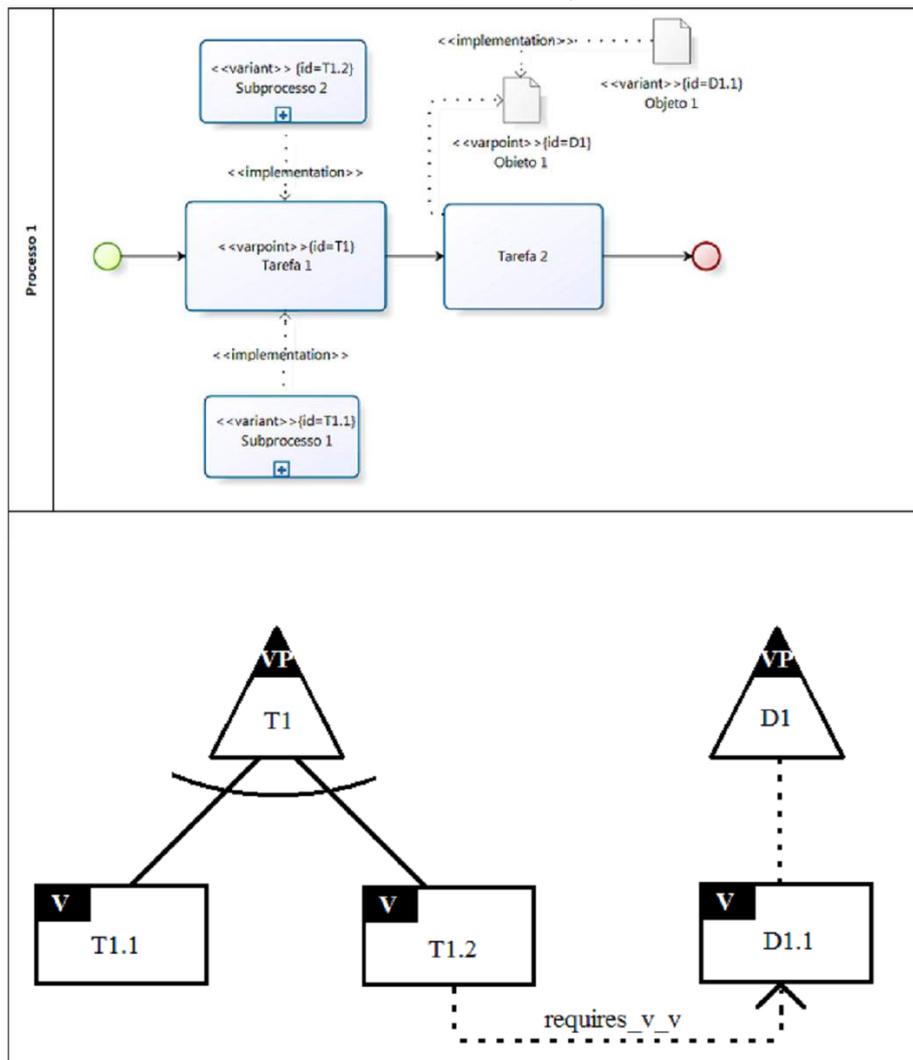
<b>Estereótipo + Tagged Value</b>	<b>Descrição</b>
<<varpoint>> {id= "X"}	Local no processo de negócio onde ocorre o ponto de variação identificado por "X".
<<variant>> {id= "X.1"}	Realização da variante identificada por "X.1"
<<implementation>>	Associação entre a realização de uma variante e o seu ponto de variação. Necessário para diferenciar essa associação de uma associação comum da notação BPMN.

Na Figura 4.5 é ilustrada a adoção da notação vrBPMN adaptada para o *template* de processo de negócio (parte superior da figura) em conjunto com o Modelo Ortogonal para o modelo de variabilidade da LPN da GLPN (parte inferior da figura). Pelo modelo ortogonal é possível observar que o ponto de variação T1 possui duas variantes, T1.1 e T1.2. A dependência entre T1 e suas variantes é mandatória e alternativa, conforme representado no modelo de variabilidades. Além disso, o ponto de variação D1 possui uma variante opcional D1.1. O modelo ortogonal mostra também que existe uma restrição entre as variantes T1.2 e D1.1, ou seja, a variante T1.2 requer a variante D1.1.

Os locais no processo de negócio onde ocorrem T1 e D1 são assinalados no *template* de processo de negócio, bem como são representadas as realizações de T1.1, T1.2 e D1.1.

Landre *et al.* (2014) fizeram um estudo para analisar a cobertura do vrBPMN quanto a representação dos elementos do modelo de características e observaram que a dependência IOR do modelo de característica não pôde ser representada no vrBPMN. Para resolver esse problema, os autores propuseram uma extensão dessa notação denominada vrBPMN\* e mostraram que o modelo de características e essa notação estendida modelam de maneira adequada LPNs. Assim, quando o modelo de variabilidade adotado durante o uso da GLPN for o modelo de características, recomenda-se utilizar a notação vrBPMN\* para representar o *template* de processos de negócio.

Figura 4.5 - Principais artefatos do EDN: *template* de processos de negócio e modelo de variabilidade (LANDRE, 2012)



Cada fase do processo EDN é descrita resumidamente a seguir:

- Descobrir Processo de Negócio da LPN (DPN): o objetivo é obter os Processos do Domínio do Negócio (PDO) iniciais para que a LPN seja criada. No mínimo três PDOs são necessários para a criação de uma LPN.
- Definir Flexibilidades: o objetivo é definir as semelhanças e as diferenças nos PDOs iniciais que vão compor a LPN.
- Validar LPN: o objetivo é garantir que cada modelo de processo de negócio não viole restrições de boa formação de acordo com o *template* de processo de negócio.

- Gerenciar LPN: composta pelas atividades Armazenar informações atualizadas sobre as instâncias da LPN no repositório da LPN e Propagar mudanças gerais para as instâncias da LPN.

As fases que compõem o processo EPN são: Configurar Processos de Negócio, Resolução das Variabilidades e Melhoria Contínua de Processos de Negócio.

Nesse processo os artefatos que devem ser criados são as instâncias dos processos de negócio para uma organização específica. Para a criação desses artefatos, são necessários os artefatos obtidos do EDN (modelo de variabilidade e *template* de processos de negócio).

Cada fase do processo EPN é descrita resumidamente a seguir:

- Configurar Processos de Negócio (CPN): objetiva atribuir valores aos elementos configuráveis identificados no EDN, de maneira a retratar a realidade na qual o PDO, criado a partir da instanciação da LPN, irá ocorrer.
- Resolver Variabilidades: possibilita analisar cada ponto de variação do modelo de variabilidades, demarcadas no *template* de processos de negócio, e permite selecionar as variantes necessárias para a instanciação do novo processo de negócio. Por fim, os modelos de processos de negócio já configurados e com as variantes resolvidas é gerado, compondo assim o PDO resultante da instanciação da LPN.
- Melhoria Contínua de Processos de Negócio: tem como principal objetivo adequar o PDO às mudanças da organização de maneira ágil. Nesta fase ocorre o envio dos *feedbacks* a respeito das mudanças a serem feitas para a fase Gerenciar LPN do EDN que aplica as alterações e encaminha para o repositório de LPS.

#### **4.4 Considerações finais**

Neste capítulo foram apresentados os principais conceitos sobre LPN e, em seguida, foram apresentadas algumas abordagens de LPN, em especial, a abordagem GLPN, desenvolvida pelo grupo de pesquisa no qual este projeto de mestrado está inserido. Essa abordagem requer um repositório para apoiar a gestão das LPNs definidas, evidenciando a importância e relevância da arquitetura de referência Cambuci-LPN que é definida por este trabalho.

## Capítulo 5 - Estabelecimento da Cambuci

### 5.1 Considerações iniciais

Neste capítulo é apresentada a arquitetura de referência baseada em serviços para ativos de software Cambuci. O objetivo dessa arquitetura de referência é apoiar a fase de projeto arquitetural durante o desenvolvimento de repositórios de ativos de software, dentre eles, repositórios de LPN, que é de interesse deste trabalho. Para viabilizar isso, a Cambuci é especializada no próximo capítulo em uma arquitetura de referência de repositórios de LPN (RefRefLPN-SOA).

Para a definição da Cambuci foi utilizado o ProSA-RA, que é um processo que sistematiza o desenvolvimento e a avaliação de arquiteturas de referências por meio de quatro passos bem definidos, conforme descrito na Seção 2.4 do Capítulo 2.

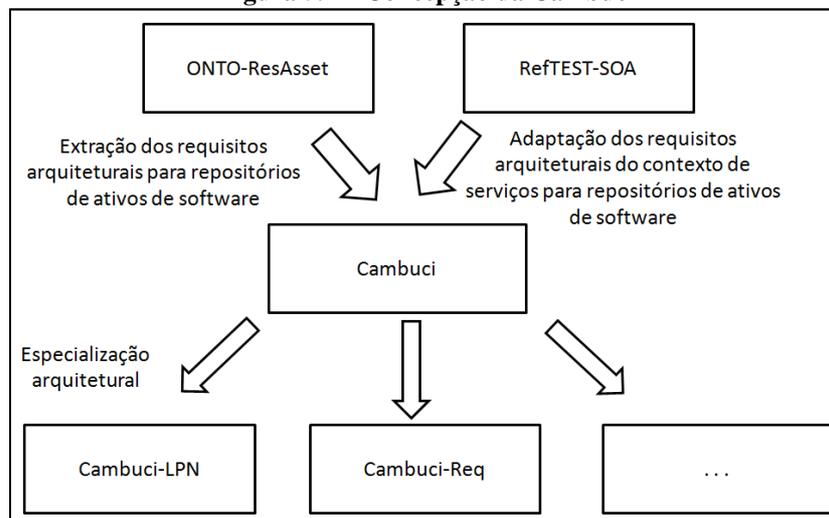
A arquitetura de referência RefTEST-SOA (OLIVEIRA, 2011), apresentada na Seção 2.5 do Capítulo 2, é uma das fontes de informação importante para o estabelecimento da Cambuci visto que é uma arquitetura de referência de ferramentas de software, assim como a Cambuci, e é uma arquitetura de referência baseada em serviços que é de interesse deste trabalho. Assim, os serviços da RefTEST-SOA são utilizados como base para a definição dos serviços da arquitetura de referência orientada a serviços Cambuci.

A arquitetura de referência RefTEST-SOA possui um conjunto de requisitos arquiteturais para o contexto de serviços que são adaptados para o domínio de repositórios de ativos de software para o estabelecimento da arquitetura de referência Cambuci, conforme ilustrado na Figura 5.1. A arquitetura de referência Cambuci é utilizada como base para a especialização de outras arquiteturas de referência para repositórios de ativos de software, como é o caso da arquitetura de referência Cambuci-LPN e de arquiteturas de referência baseadas em serviços para repositórios de requisitos de software (Cambuci-Req).

Neste capítulo é apresentada a aplicação de cada passo do ProSA-RA durante a definição da Cambuci. Na Seção 5.2 é realizada a investigação das fontes de informação utilizadas na identificação dos requisitos arquiteturais da Cambuci. Na Seção 5.3 é apresentado o conjunto de requisitos arquiteturais identificados a partir das fontes de informação. Na Seção 5.4 é descrita a arquitetura de referência Cambuci por meio de uma visão geral e três visões arquiteturais. Na Seção 5.5 é descrito como a

arquitetura de referência Cambuci é avaliada. Na Seção 5.6 são apresentadas as considerações finais deste capítulo.

**Figura 5.1 – Concepção da Cambuci**



## 5.2 Passo 1: Investigação de fontes de informação

Durante o primeiro passo do ProSA-RA foram identificadas diferentes fontes para obtenção de informações sobre o domínio de repositórios de ativos de software. Para essa atividade foram considerados os seguintes grupos de fontes de informação: (i) ontologia de especificação e gestão de ativos reutilizáveis ONTO-ResAsset (Seção 3.5 do Capítulo 3); (ii) estudos sobre arquiteturas de referências para repositórios de ativos de vários domínios, descritos na Seção 2.6 do Capítulo 2; e (iii) estudo sobre requisitos arquiteturais do contexto de serviços da arquitetura de referência RefTEST-SOA, descritos na Seção 2.5 do Capítulo 2. Nas subseções a seguir são apresentados os estudos presentes em cada grupo de fonte de informação utilizado para a definição da Cambuci.

### 5.2.1 Grupo 1: ontologia ONTO-ResAsset

A fonte de informação deste grupo é basicamente o trabalho de Silva (2013), que trata de uma Ontologia de Ativos Reutilizáveis, denominada ONTO-ResAsset, no domínio de especificação e gestão de ativos reutilizáveis de software, conforme descrito na Seção 3.5. Essa ontologia foi selecionada porque os mecanismos de gestão de ativos nela descritos correspondem às funcionalidades que devem estar presentes em repositórios de ativos de software e, conseqüentemente, devem ser levados em consideração na definição de arquiteturas de referência desse tipo de software.

As funcionalidades para repositórios de ativos reutilizáveis identificadas neste grupo, a partir dos mecanismos de gestão de ativos presentes na ONTO-ResAsset, estão relacionadas no Quadro 5.1. Nesse quadro, cada funcionalidade é relacionada a conceitos relevantes do domínio de repositórios de ativos de software, que são classificados como *Serviços Primários de Repositórios de Ativos de Software*, que se referem aos principais serviços de um repositório de ativo de software, ou a conceitos relevantes relacionados aos outros serviços da camada de aplicação definidos por Oliveira (2011), que são: *Serviços Ortogonais de Suporte* (atividades de apoio consideradas de suporte pela ISO/IEC 12207 (2009)), *Serviços Ortogonais Organizacionais* (atividades de apoio consideradas organizacionais pela ISO/IEC 12207 (2009)) e *Serviços Ortogonais Gerais* (atividades de propósito gerais, como persistência e segurança). Essa classificação é útil para a execução do passo 2 (estabelecimento dos requisitos arquiteturais) do ProSA-RA para a definição da Cambuci, conforme descrito na Seção 5.3.

Na ONTO-ResAsset estão definidos os conceitos de *Especificação de Ativos* e de *Gestão de Ativos* de modo bem delimitado para o entendimento do domínio de especificação e de gestão de ativos. No contexto de repositórios, a especificação de um ativo é completamente dependente da inclusão desse ativo no repositório. Assim, para este trabalho, justifica-se que a funcionalidade *Inclusão de Ativo* está relacionada ao conceito *Especificação de Ativo* e não ao conceito *Gestão de Ativo*, pois é a funcionalidade que permite especificar um ativo no repositório.

Ressalta-se que as funcionalidades de *Gerência de Configuração*, *Controle de Dependência e Versionamento*, relacionadas com o conceito de *Gestão de Ativo* por Silva (2013) são determinadas como *Serviços Ortogonais de Suporte* neste trabalho. As funcionalidades de *Notificação de Ativo*, *Feedback*, *Publicidade* e *Métrica*, também relacionadas com o conceito de *Gestão de Ativo* por Silva (2013), são definidas como *Serviços Ortogonais Organizacionais*. Já a funcionalidade *Controle de Acesso*, considerada como *Gestão de Ativo*, é determinada como *Serviços Ortogonais Gerais*.

**Quadro 5.1 - Funcionalidades identificadas a partir da ONTO-ResAsset**

<b>Funcionalidades Identificadas</b>	<b>Descrição (SILVA, 2013)</b>	<b>Conceitos</b>	<b>Serviços</b>
Inclusão de Ativo	Inclui um ativo contendo os artefatos que o compõe	Especificação de Ativo	Serviços Primários
Classificação de Ativo	Permite classificar um ativo, informando o contexto de sua utilização	Especificação de Ativo	Serviços Primários
Aceitação e Certificação de Ativo	Permite aceitar e certificar um ativo tornando-o disponível para reuso, segundo os critérios de aceitação e certificação estabelecidos	Gestão de Ativo	Serviços Primários
Desativação de Ativo	Permite tornar um ativo indisponível para reuso	Gestão de Ativo	Serviços Primários
Busca	Permite a busca e a recuperação dos ativos no repositório	Gestão de Ativo	Serviços Primários
Navegação	Permite a navegação pelos ativos armazenados	Gestão de Ativo	Serviços Primários
Suporte às múltiplas fontes	Aceita múltiplas fontes de origem de ativos, com o objetivo de facilitar a integração entre equipes e entre diferentes repositórios	Gestão de Ativo	Serviços Primários
Gerência de configuração	Permite o gerenciamento de configuração, por exemplo, a definição dos itens do ativo que são configuráveis e o controle de mudanças dos configuráveis	Gerência de Configuração	Serviços Ortogonais de Suporte
Controle de dependência	Permite o registro de dependência entre ativos	Controle de Dependência	Serviços Ortogonais de Suporte
Versionamento	Permite a criação e o armazenamento de múltiplas versões de um mesmo ativo	Versionamento	Serviços Ortogonais de Suporte
Notificação de Ativo	Permite realizar a notificação conforme ocorram mudanças no ativo	Comunicação	Serviços Ortogonais Organizacionais
<i>Feedback</i>	Permite o registro de impressões dos usuários a respeito da versão do ativo que eles utilizaram	Comunicação	Serviços Ortogonais Organizacionais
Métrica	Permite o registro de métricas coletadas sobre a utilização do ativo	Métrica	Serviços Ortogonais Organizacionais
Publicidade	Oferece informações relativas ao reuso, iniciativas de reuso, ativos mais usados, etc	Comunicação	Serviços Ortogonais Organizacionais
Controle de Acesso	Determina o acesso ao repositório, de acordo com o papel que o usuário assume	Segurança	Serviços Ortogonais Gerais

### 5.2.2 Grupo 2: Estudos de abordagens de repositórios de ativos de software

Os trabalhos deste grupo referem-se à três trabalhos que tratam de arquiteturas de referência para repositórios de ativos de software, descritos na Seção 2.6 do Capítulo 2 e estão relacionados no Quadro 5.2. Basicamente, o primeiro trabalho trata de repositórios de ativos de segurança, o segundo trabalho está relacionado aos repositórios de modelos de processos de negócio, e o terceiro trabalho aborda sobre repositórios mais genéricos de ativos de software.

**Quadro 5.2 - Estudos sobre abordagens de repositórios de ativos de software**

<b>Id.</b>	<b>Autores</b>	<b>Títulos</b>
1	MELAND <i>et al.</i> (2009)	<i>An architectural foundation for security model sharing and reuse.</i>
2	YAN <i>et al.</i> (2012)	<i>Business process model repositories – Framework and survey.</i>
3	HONGMIN <i>et al.</i> (2010)	<i>Software asset repository open framework supporting customizable faceted classification.</i>

Os trabalhos citados no Quadro 5.2 foram analisados visando levantar outras funcionalidades de repositórios de ativos de software, além daquelas identificadas a partir da ONTO-ResAsset (Seção 5.2.1). Para isso, o Quadro 5.3 foi elaborado contendo as funcionalidades identificadas nos três estudos supracitados, a indicação dos estudos em que cada funcionalidade é considerada, a correspondência de cada funcionalidade com os mecanismos de gestão de ativos da ONTO-ResAsset, bem como os conceitos relacionados a cada funcionalidade. Cada funcionalidade também foi classificada como serviço, conforme critério de classificação utilizado no Grupo 1 de fonte de informação visando facilitar a execução do passo 2 (Estabelecimento de Requisitos Arquiteturais) do ProSA-RA apresentada na Seção 5.3.

Verifica-se que a maioria das funcionalidades, citada no Quadro 5.3, pertence ao estudo de Yan *et al.* (2012), com exceção das funcionalidades de integridade e de transação. Essas também não estão descritas na ONTO-ResAsset, mas são funcionalidades que devem ser providas por repositórios de ativos de software conforme estudos das fontes de informação do Grupo 2.

**Quadro 5.3 - Funcionalidades identificadas nas abordagens de repositórios de ativos de software**

Funcionalidades identificadas	Estudos contemplados	Equivalência com a ONTO-ResAsset	Conceitos	Serviços
Gerenciamento de ativo	1,2,3	Mecanismos de inclusão, aceitação, certificação, desativação, notificação e classificação.	Especificação de Ativo e Gestão de Ativo	Serviços Primários
Aprovação	1	Mecanismo de aprovação	Gestão de Ativo	Serviços Primários
Armazenamento	2	Mecanismo de inclusão	Especificação de Ativo	Serviços Primários
Gerenciamento de faceta	3	Mecanismo de classificação	Especificação de Ativo	Serviços Primários
Gerenciamento de metadados	2,3	Mecanismo de inclusão	Especificação de Ativo	Serviços Primários
Compartilhamento de ativos com ferramentas de terceiros	1,2,3	Suporte à múltiplas fontes	Gestão de Ativo	Serviços Primários
Compartilhamento de ativos com outros repositórios	1,3	Suporte à múltiplas fontes	Gestão de Ativo	Serviços Primários
Gerenciamento de contexto	2	Mecanismo de classificação	Especificação de Ativo	Serviços Primários
Função de busca	2	Mecanismo de busca	Gestão de Ativo	Serviços Primários
Versionamento	1,2	Mecanismo de versionamento	Versionamento	Serviços Ortogonais de Suporte
Gerenciamento de configuração	2	Mecanismo de gerência de configuração	Gerência de Configuração	Serviços Ortogonais de Suporte
Gerenciamento de <i>checkin/checkout</i>	2	Mecanismo de gerência de configuração	Gerência de Configuração	Serviços Ortogonais de Suporte
Gerenciamento de notificação	2	Mecanismo de notificação	Comunicação	Serviços Ortogonais Organizacionais
<i>Feedback</i>	1	Mecanismo de <i>feedback</i>	Comunicação	Serviços Ortogonais Organizacionais
Estatística de utilização	1	Mecanismo de publicidade	Comunicação	Serviços Ortogonais Organizacionais
Gerenciamento de despacho	2	Não	Comunicação	Serviços Ortogonais Organizacionais
Gerenciamento de visão (público, privado)	2	Mecanismo de controle de acesso	Segurança	Serviços Ortogonais Gerais
Gerenciamento de integridade	2	Não, pois esta é uma funcionalidade específica de sistemas de gerenciamento de banco de dados	Persistência	Serviços Ortogonais Gerais

**Quadro 5.3 (conclusão)**

<b>Funcionalidades identificadas</b>	<b>Estudos contemplados</b>	<b>Equivalência com a ONTO-ResAsset</b>	<b>Conceitos</b>	<b>Serviços</b>
Gerenciamento de usuário	1,2,3	Mecanismo de controle de acesso	Segurança	Serviços Ortogonais Gerais
Gerenciamento de autorização	1,2,3	Mecanismo de controle de acesso	Segurança	Serviços Ortogonais Gerais
Gerenciamento de transação	2	Não, pois esta é uma funcionalidade específica de sistemas de gerenciamento de banco de dados	Persistência	Serviços Ortogonais Gerais

### 5.2.3 Grupo 3: requisitos arquiteturais do contexto de serviços

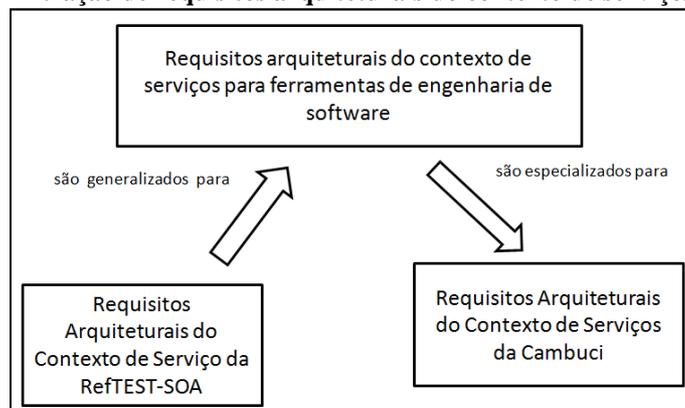
A fonte de informação que faz parte deste grupo é basicamente o trabalho de Oliveira (2011), que foi selecionado justamente pelo fato de apresentar a definição de uma arquitetura de referência orientada a serviços, de interesse deste trabalho, para o domínio de ferramentas de teste.

Como ferramentas de teste e repositórios de ativos de software são ferramentas de engenharia de software e, uma vez que Oliveira (2011) definiu os requisitos arquiteturais do contexto de serviços utilizando diretrizes para o desenvolvimento de sistemas orientados a serviço (ARSANJANI *et al.*, 2007; DILLON *et al.*, 2007; LAN *et al.*, 2008; OASIS, 2006; OASIS, 2008; e ZIMMERMANN *et al.*, 2009) e arquiteturas de referência orientadas a serviço projetadas para diferentes domínios (BREHM e GÓMES, 2007; CHOI *et al.*, 2009; DURO *et al.*, 2005; FIORAVANTI *et al.*, 2007; HEMALATHA *et al.*, 2008; LEPPANIEMI *et al.*, 2009; MURAKAMI *et al.*, 2007; PERISTERAS *et al.*, 2009; RAMANATHAN *et al.*, 2008; REIFF-MARGANIEC *et al.*, 2008; e ZHENG *et al.*, 2008), esses requisitos arquiteturais foram abstraídos para o contexto de ferramentas de engenharia de software orientadas a serviço, conforme apresentado no Quadro 5.4.

Para isso, a descrição de cada requisito arquitetural de ferramentas de teste baseadas em SOA, definido por Oliveira (2011), foi generalizada a fim de obter o requisito correspondente para o contexto de ferramentas de engenharia de software conforme ilustrado na Figura 5.2. Esses requisitos arquiteturais do contexto de serviço para ferramentas de engenharia de software podem ser especializados para formar os requisitos arquiteturais do contexto de serviço para qualquer domínio de ferramentas de engenharia de software, como por exemplo, o domínio de repositórios de ativos de

software. Isso será feito na execução do passo 2 (Estabelecimento dos Requisitos Arquiteturais) do ProSA-RA (Seção 5.3).

**Figura 5.2 - Extração de requisitos arquiteturais do contexto de serviços da Cambuci**



**Quadro 5.4 - Requisitos arquiteturais do contexto de serviço para ferramentas de engenharia de software (adaptado de Oliveira, 2011)**

ID	Requisitos	Conceitos
RA-S [1]	A arquitetura de referência deve possibilitar que ferramentas de engenharia de software desenvolvidas para automatizar diferentes etapas do processo de desenvolvimento possam ser facilmente integradas	Requisito Geral
RA-S [2]	A arquitetura de referência deve possibilitar que ferramentas de engenharia de software implementadas em linguagens de programação distintas e sob diferentes plataformas possam ser facilmente integradas	Requisito Geral
RA-S [3]	A arquitetura de referência deve prover mecanismos para que ferramentas de engenharia de software na forma de serviços possam ser publicadas e posteriormente descobertas por aplicações cliente	Publicação de Serviço
RA-S [4]	A arquitetura de referência deve prover mecanismos para que ferramentas de engenharia de software orientadas a serviço possam ser compostas por processos de negócio ou utilizadas por aplicações cliente	Interação entre Serviços, Composição de Serviços
RA-S [5]	A arquitetura de referência deve viabilizar o desenvolvimento de ferramentas de engenharia de software que disponibilizem informações sobre suas características e direções normativas de uso, por meio de descrições padronizadas	Descrição de Serviço
RA-S [6]	A arquitetura de referência deve viabilizar o desenvolvimento de ferramentas de engenharia de software que disponibilizem descrições semânticas, permitindo assim sua classificação nos repositórios de serviço	Descrição de Serviço, Publicação de Serviço
RA-S [7]	A arquitetura de referência deve viabilizar o desenvolvimento de ferramentas de engenharia de software que tenham à disposição informações e documentos relacionados às suas características de qualidade	Descrição de Serviço, Qualidade de Serviço
RA-S [8]	A arquitetura de referência deve prover mecanismos para a captura, monitoramento, registro e sinalização do não cumprimento de requisitos de qualidade estabelecidos entre serviços provedores e serviços clientes	Qualidade de Serviço
RA-S [9]	A arquitetura de referência deve viabilizar o desenvolvimento de ferramentas de engenharia de software escaláveis, capazes de evoluir de maneira incremental, por meio da composição de novas funcionalidades disponíveis na forma de serviços	Composição de Serviços

**Quadro 5.4 (conclusão)**

<b>ID</b>	<b>Requisitos</b>	<b>Conceitos</b>
RA-S [10]	A arquitetura de referência deve possibilitar que serviços de interesse de ferramentas de engenharia de software e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma	Descrição de Serviço, Publicação de Serviço, Interação entre Serviços, Composição de Serviços
RA-S [11]	A arquitetura de referência deve possibilitar que serviços de interesse de ferramentas de engenharia de software possam interagir diretamente ou por meio do uso de barramentos de serviço	Interação entre Serviços
RA-S [12]	A arquitetura de referência deve possibilitar que ferramentas de engenharia de software disponíveis como serviços, desenvolvidas sem o uso dessa arquitetura, possam ser integradas às ferramentas desenvolvidas segundo a arquitetura de referência	Requisito Geral
RA-S [13]	A arquitetura de referência deve possibilitar sua instanciação parcial, ou seja, ferramentas desenvolvidas a partir dessa arquitetura podem ser construídas sem a necessidade de implementação de todos os módulos propostos	Requisito Geral
RA-S [14]	A arquitetura de referência deve permitir que ferramentas de engenharia de software possam evoluir ao longo de diferentes versões de maneira simples, ou seja, mudanças de versão das ferramentas de ES não devem afetar a interação com os serviços que a utilizam	Interação entre Serviços, Composição de Serviços
Legenda: RA-S refere-se à um Requisito Arquitetural do contexto de Serviço		

Cada requisito arquitetural está classificado de acordo com os conceitos explicitados na arquitetura de referência RefTEST-SOA estabelecida por Oliveira (2011), apresentado na Seção 2.5 do Capítulo 2, a fim de facilitar a execução do passo 2 (Estabelecimento dos Requisitos Arquiteturais) do ProSA-RA.

### **5.3 Passo 2: Análise arquitetural**

Neste trabalho, na execução do passo 2 do ProSA-RA, foram utilizadas as fontes de informações selecionadas na Seção 5.2 para elicitar os requisitos arquiteturais da arquitetura de referência Cambuci, os quais são divididos em dois grupos descritos nas subseções a seguir. O primeiro grupo se relaciona com os requisitos arquiteturais para o domínio de repositórios de ativos de software e o segundo grupo se relaciona aos requisitos arquiteturais do contexto de serviços.

#### **5.3.1 Requisitos arquiteturais do domínio de repositórios de ativos de software**

As funcionalidades identificadas nos estudos dos Grupos 1 e 2, apresentados na Seção 5.2, são tomadas como base para a elicitação dos requisitos arquiteturais do domínio de repositórios de ativos de software. A descrição de cada requisito arquitetural é baseada no trecho de texto "A arquitetura de referência deve possibilitar que repositórios de ativos de software" seguido pelo conteúdo de cada funcionalidade identificada para compor a arquitetura de referência (Quadro 5.1 e Quadro 5.3).

No Quadro 5.5 são listados os requisitos arquiteturais relacionados aos repositórios de ativos de software para o estabelecimento da Cambuci. Nesse quadro é apresentada uma relação entre os requisitos arquiteturais, os conceitos do domínio, bem como a classificação de cada requisito arquitetural de acordo com os módulos Serviços Primários, Serviços Ortogonais de Suporte, Serviços Ortogonais Organizacionais e Serviços Ortogonais Gerais da visão geral da arquitetura de referência da RefTest-SOA (Seção 2.5 do Capítulo 2). Essa classificação foi baseada na classificação previamente realizada na Seção 5.2.

**Quadro 5.5 - Requisitos arquiteturais para repositórios de ativos de software**

<b>ID</b>	<b>Descrição</b>	<b>Conceitos</b>	<b>Serviços</b>
RA-AS[1]	A arquitetura de referência deve possibilitar que repositórios de ativos de software incluam um novo ativo, que pode ser composto por vários artefatos;	Especificação de Ativo	Serviço Primário
RA-AS[2]	A arquitetura de referência deve possibilitar que repositórios de ativos de software forneçam mecanismo para aceitação e certificação de ativos;	Gestão de Ativo	Serviço Primário
RA-AS[3]	A arquitetura de referência deve possibilitar que repositórios de ativos de software desativem ativos que não serão mais utilizados;	Gestão de Ativo	Serviço Primário
RA-AS[4]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam a classificação de um ativo e também informar o contexto de sua utilização;	Especificação de Ativo	Serviço Primário
RA-AS[5]	A arquitetura de referência deve possibilitar que repositórios de ativos de software registrem a dependência entre ativos;	Especificação de Ativo	Serviço Primário
RA-AS[6]	A arquitetura de referência deve possibilitar que repositórios de ativos de software notifiquem os interessados sobre mudanças que aconteçam no ativo;	Comunicação	Serviços Ortogonais Organizacionais
RA-AS[7]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam realizar buscas e recuperação dos ativos;	Gestão de Ativo	Serviço Primário
RA-AS[8]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam a navegação entre ativos;	Gestão de Ativo	Serviço Primário
RA-AS[9]	A arquitetura de referência deve possibilitar que repositórios de ativos de software aceite múltiplas fontes de origem de ativos, com o objetivo de facilitar a integração entre equipes e entre repositórios diferentes;	Gestão de Ativo	Serviço Primário
RA-AS[10]	A arquitetura de referência deve possibilitar que repositórios de ativos de software criem e armazenem múltiplas versões de um mesmo ativo;	Versionamento	Serviços Ortogonais de Suporte
RA-AS[11]	A arquitetura de referência deve possibilitar que repositórios de ativos de software gerencie a configuração, como por exemplo, a definição dos itens do ativo que são configuráveis, o controle de mudanças dos itens do ativo que são configuráveis;	Gerência de Configuração	Serviços Ortogonais de Suporte
RA-AS[12]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permita o registro de impressões dos usuários a respeito da versão do ativo que eles utilizaram;	Comunicação	Serviços Ortogonais Organizacionais

**Quadro 5.5 (conclusão)**

<b>ID</b>	<b>Descrição</b>	<b>Conceitos</b>	<b>Serviços</b>
RA-AS[13]	A arquitetura de referência deve possibilitar que repositórios de ativos de software registrem métricas coletadas sobre a utilização do ativo;	Métrica	Serviços Ortogonais Organizacionais
RA-AS[14]	A arquitetura de referência deve possibilitar que repositórios de ativos de software ofereçam informações relativas ao reúso, iniciativas de reúso, ativos mais usados, etc;	Comunicação	Serviços Ortogonais Organizacionais
RA-AS[15]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam o acesso de acordo com o papel que o usuário assume;	Segurança	Serviços Ortogonais Gerais
RA-AS[16]	A arquitetura de referência deve possibilitar que repositórios de ativos de software garantam a integridade dos ativos, ou seja, que eles não sofram alterações não autorizadas;	Persistência	Serviços Ortogonais Gerais
RA-AS[17]	A arquitetura de referência deve possibilitar que repositórios de ativos de software realizem o gerenciamento de transação, garantindo a atomicidade, consistência, isolamento e durabilidade;	Persistência	Serviços Ortogonais Gerais
Legenda: RA-AS refere-se à um Requisito Arquitetural do contexto de repositórios de Ativos de Software			

### 5.3.2 Requisitos arquiteturais do contexto de serviços

Os requisitos arquiteturais do contexto de serviços para ferramentas de engenharia de software (Seção 5.2.3), abstraídos a partir dos requisitos arquiteturais identificados por Oliveira (2011), são adaptados para o contexto de repositório de ativos de software. Para isso, as descrições dos requisitos arquiteturais de ferramentas de engenharia de software são alteradas para se relacionar com o contexto de repositório de ativo de software.

No Quadro 5.6 são listados os requisitos arquiteturais do contexto de serviços para repositórios de ativos de software visando a concepção da Cambuci. Adicionalmente, são apresentados os conceitos do domínio de serviços e as camadas de serviços correspondentes. Salienta-se que os requisitos arquiteturais são adaptações dos requisitos arquiteturais da Seção 5.2.3, os conceitos do domínio de serviços são os mesmos conceitos da Seção 5.2.3 e as camadas de serviços foram definidas por Oliveira (2011) na visão geral da arquitetura e são utilizadas no projeto arquitetural que se encontra na Seção 5.4.

**Quadro 5.6 - Requisitos arquiteturais do contexto de serviços para repositórios de ativos de software**

<b>ID</b>	<b>Requisitos</b>	<b>Conceitos</b>	<b>Serviços</b>
RA-S[1]	A arquitetura de referência deve possibilitar que repositórios de ativos de software desenvolvidos para persistir diferentes tipos de ativos possam ser facilmente integrados	Requisito Geral	Processo de negócio
RA-S[2]	A arquitetura de referência deve possibilitar que repositórios de ativos de software implementados em linguagens de programação distintas e sob diferentes plataformas possam ser facilmente integrados	Requisito Geral	Processo de negócio
RA-S[3]	A arquitetura de referência deve prover mecanismos para que repositórios de ativos de software na forma de serviços possam ser publicados e posteriormente descobertos por aplicações cliente	Publicação de Serviço	Intermediação
RA-S[4]	A arquitetura de referência deve prover mecanismos para que repositórios de ativos de software orientados a serviço possam ser compostos por processos de negócio ou utilizados por aplicações cliente	Interação entre Serviços, Composição de Serviços	Processo de negócio
RA-S[5]	A arquitetura de referência deve viabilizar o desenvolvimento de repositórios de ativos de software que disponibilizem informações sobre suas características e direções normativas de uso, por meio de descrições padronizadas	Descrição de Serviço	Apresentação
RA-S[6]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software que disponibilizem descrições semânticas, permitindo assim sua classificação nos repositórios de serviço	Descrição de Serviço, Publicação de Serviço	Apresentação, Intermediação
RA-S[7]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software que tenham à disposição informações e documentos relacionados às suas características de qualidade	Descrição de Serviço, Qualidade de Serviço	Apresentação, Qualidade de Serviço
RA-S[8]	A arquitetura de referência deve prover mecanismos para a captura, monitoramento, registro e sinalização do não cumprimento de requisitos de qualidade estabelecidos entre serviços provedores e serviços clientes	Qualidade de Serviço	Qualidade de Serviço
RA-S[9]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software escalável, capaz de evoluir de maneira incremental, por meio da composição de novas funcionalidades disponíveis na forma de serviços	Composição de Serviços	Processo de negócio
RA-S[10]	A arquitetura de referência deve possibilitar que serviços de repositório de ativos de software e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma	Descrição de Serviço, Publicação de Serviço, Interação entre Serviços, Composição de Serviços	Apresentação, Intermediação, Processo de negócio
RA-S[11]	A arquitetura de referência deve possibilitar que serviços do repositório de ativos de software possam interagir diretamente ou por meio do uso de barramentos de serviço	Interação entre Serviços	Processo de negócio

**Quadro 5.6 (conclusão)**

<b>ID</b>	<b>Requisitos</b>	<b>Conceitos</b>	<b>Serviços</b>
RA-S[12]	A arquitetura de referência deve possibilitar que repositório de ativos de software disponíveis como serviços, desenvolvidos sem o uso dessa arquitetura, possam ser integrados aos repositórios de ativos de software desenvolvidos segundo a arquitetura de referência	Requisito Geral	Processo de negócio
RA-S[13]	A arquitetura de referência deve possibilitar sua instanciação parcial, ou seja, repositórios de ativos de software desenvolvidos a partir dessa arquitetura podem ser construídos sem a necessidade de implementação de todos os módulos propostos	Requisito Geral	Processo de negócio
RA-S[14]	A arquitetura de referência deve permitir que os repositórios de ativos de software possam evoluir ao longo de diferentes versões de maneira simples, ou seja, mudanças de versão dos repositórios de ativos de software não devem afetar a interação com os serviços que a utilizam	Interação entre Serviços, Composição de Serviços	Processo de negócio

Legenda: RA-S refere-se à um Requisito Arquitetural do contexto de Serviço

### 5.4 Passo 3: Síntese arquitetural

Tomando como base os requisitos arquiteturais elicitados no passo anterior, na execução deste terceiro passo do ProSA-RA (Projeto Arquitetural) é realizado o projeto arquitetural da Cambuci. Merson (2005) propõe as seguintes visões arquiteturais: visão de módulo, visão de implantação, visão de tempo de execução, visão de dados e visão de implementação. As visões utilizadas para descrever a Cambuci são: visão geral (Seção 6.4.1), visão de módulo (Seção 6.4.2), visão em tempo de execução (Seção 6.4.3) e visão de implantação (Seção 6.4.4). Conforme discutido por Nakagawa (2006), as visões de implementação e de dados não são construídas em arquiteturas de referência representadas em alto nível de abstração, como é o caso da Cambuci. Assim essas duas visões não são criadas neste trabalho.

#### 5.4.1 Visão geral

A visão geral da Cambuci foi desenvolvida tomando como base a visão geral da RefTEST-SOA proposta por Oliveira (2011), apresentada na Seção 2.5, e levando em consideração os requisitos arquiteturais identificados no passo anterior (Seção 5.3).

Os serviços do domínio de repositórios de ativos de software (serviços primários de repositórios de ativos de software, serviços ortogonais de apoio, serviços ortogonais organizacionais e serviços ortogonais gerais) foram alocados na camada de aplicação, enquanto que os serviços do contexto de serviços foram alocados nas outras camadas (processo de negócio, intermediação, apresentação ou de qualidade de serviço), conforme o seu propósito.

Enquanto que a visão geral de Oliveira (2011) contém os serviços de teste primários relacionados aos conceitos do domínio de teste (artefato de teste, critério de teste, caso de teste e requisito de teste), a visão geral da Cambuci contém os serviços primários de repositórios relacionados aos conceitos do domínio de repositório de ativos de software (especificação de ativo e gestão de ativo).

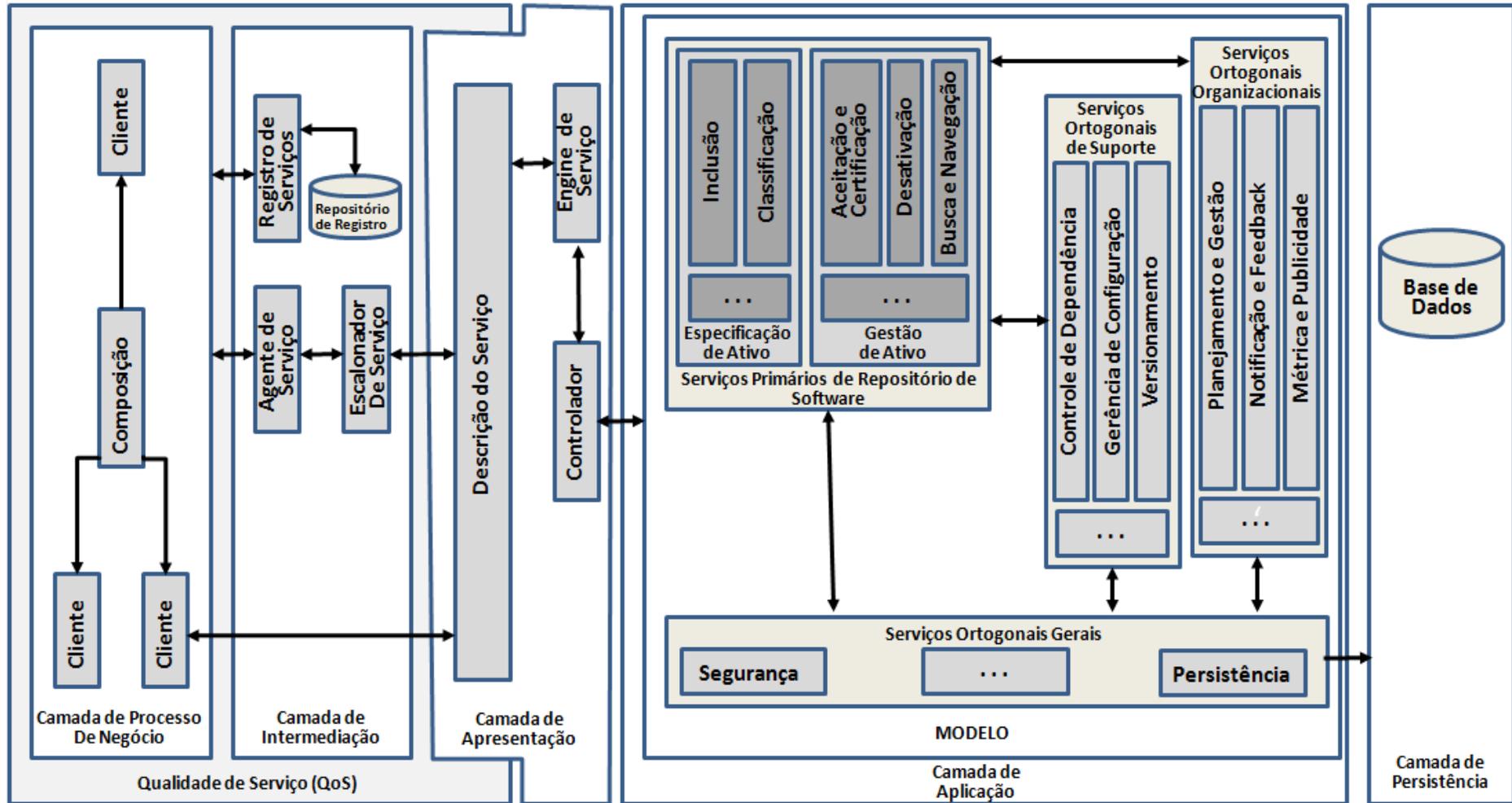
Na Figura 5.3 é apresentada a visão geral da Cambuci, a qual é dividida em seis camadas, descritas a seguir:

- **Camada de Aplicação:** contempla os serviços oferecidos relativos ao domínio de repositórios de ativos de software. Esses serviços são divididos nos seguintes conjuntos, em conformidade com a descrição da visão geral do RefTEST-SOA de Oliveira (2011): (i) Serviços Primários de Repositórios de Ativos de Software, que se referem aos serviços de especificação de ativos e aos serviços de gestão de ativos; (ii) Serviços Ortogonais de Suporte, que se referem aos serviços que apoiam as atividades de suporte conforme a ISO/IEC 12207 (2009); (iii) Serviços Ortogonais Organizacionais, que se referem aos serviços que apoiam as atividades organizacionais conforme a ISO/IEC 12207 (2009); (iv) Serviços Ortogonais Gerais, que se referem aos serviços de propósito geral. Os serviços do grupo (ii), (iii) e (iv) são considerados ortogonais porque podem ser utilizados por diversos outros serviços durante a atividade de especificação e gestão de ativos.
- **Camada de Persistência:** é responsável por armazenar os ativos de software. O serviço de persistência atua diretamente nessa camada. Os ativos devem ser persistidos em uma base de dados. Pode-se utilizar um sistema de gerenciamento de banco de dados para realizar esta tarefa, mas também pode-se utilizar um sistema de arquivos e diretórios.
- **Camada de Apresentação:** como a Cambuci é baseada em serviços, essa camada possui os seguintes elementos, de acordo com Oliveira (2011): (i) Descrição de Serviço, que define como um serviço pode ser chamado e o que ele retorna; (ii) *Engine* de Serviço, que processa as requisições de serviços; (iii) Controlador, que é o responsável pelo tratamento das requisições encaminhadas pela *Engine* de Serviço.

- **Camada de Intermediação:** permite que os serviços de repositórios de ativos de software sejam publicados, descobertos, associados e disponibilizados. Possui os seguintes elementos: (i) Registro de Serviços, que permite a publicação de um serviço e a pesquisa de um serviço disponível; (ii) Agente de Serviço, que faz o papel de mediador entre o cliente do serviço e o provedor do serviço; (iii) Escalonador de Serviço, que processa as requisições de serviço que possuem dependências.
- **Camada de Orquestração:** permite a criação de novos serviços realizando uma composição dos serviços já existentes. Essa camada foi definida pela RefTEST-SOA como sendo a **Camada de Processo de Negócio**, mas em virtude do termo fazer parte do domínio da LPN, essa camada foi renomeada neste trabalho para camada de orquestração.
- **Camada de Qualidade de Serviço:** camada que verifica o cumprimento dos requisitos de qualidade presentes nas demais camadas de serviços. Ela pode atuar nas camadas de Orquestração, Intermediação e de Apresentação.

Para complementar a visão geral da Cambuci, apresentada na Figura 5.3, o Quadro 5.7, o Quadro 5.8, o Quadro 5.9, o Quadro 5.10, o Quadro 5.11 e o Quadro 5.12 listam os requisitos arquiteturais de cada camada de serviço.

Figura 5.3 - Visão geral da arquitetura Cambuci



**Quadro 5.7 - Requisitos arquiteturais da camada de aplicação**

Camada de Aplicação		
ID	Requisitos	Serviços
RA-AS[1]	A arquitetura de referência deve possibilitar que repositórios de ativos de software incluam um novo ativo, que pode ser composto por vários artefatos;	Serviço Primário
RA-AS[2]	A arquitetura de referência deve possibilitar que repositórios de ativos de software forneçam mecanismo para aceitação e certificação de ativos;	Serviço Primário
RA-AS[3]	A arquitetura de referência deve possibilitar que repositórios de ativos de software desativem ativos que não serão mais utilizados;	Serviço Primário
RA-AS[4]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam a classificação de um ativo e também informar o contexto de sua utilização;	Serviço Primário
RA-AS[5]	A arquitetura de referência deve possibilitar que repositórios de ativos de software registrem a dependência entre ativos;	Serviço Primário
RA-AS[7]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam realizar buscas e recuperação dos ativos;	Serviço Primário
RA-AS[8]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam a navegação entre ativos;	Serviço Primário
RA-AS[9]	A arquitetura de referência deve possibilitar que repositórios de ativos de software aceite múltiplas fontes de origem de ativos, com o objetivo de facilitar a integração entre equipes e entre repositórios diferentes;	Serviço Primário
RA-AS[6]	A arquitetura de referência deve possibilitar que repositórios de ativos de software notifiquem os interessados sobre mudanças que aconteçam no ativo;	Serviços Ortogonais Organizacionais
RA-AS[12]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permita o registro de impressões dos usuários a respeito da versão do ativo que eles utilizaram;	Serviços Ortogonais Organizacionais
RA-AS[13]	A arquitetura de referência deve possibilitar que repositórios de ativos de software registrem métricas coletadas sobre a utilização do ativo;	Serviços Ortogonais Organizacionais
RA-AS[14]	A arquitetura de referência deve possibilitar que repositórios de ativos de software ofereçam informações relativas ao reuso, iniciativas de reuso, ativos mais usados, etc;	Serviços Ortogonais Organizacionais
RA-AS[10]	A arquitetura de referência deve possibilitar que repositórios de ativos de software criem e armazenem múltiplas versões de um mesmo ativo;	Serviços Ortogonais de Suporte
RA-AS[11]	A arquitetura de referência deve possibilitar que repositórios de ativos de software gerencie a configuração, como por exemplo, a definição dos itens do ativo que são configuráveis, o controle de mudanças dos itens do ativo que são configuráveis;	Serviços Ortogonais de Suporte
RA-AS[15]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam o acesso de acordo com o papel que o usuário assume;	Serviços Ortogonais Gerais
RA-AS[16]	A arquitetura de referência deve possibilitar que repositórios de ativos de software garantam a integridade dos ativos, ou seja, que eles não sofram alterações não autorizadas;	Serviços Ortogonais Gerais
RA-AS[17]	A arquitetura de referência deve possibilitar que repositórios de ativos de software realizem o gerenciamento de transação, garantindo a atomicidade, consistência, isolamento e durabilidade;	Serviços Ortogonais Gerais

Legenda: RA-AS refere-se à um Requisito Arquitetural do contexto de repositórios de Ativos de Software

**Quadro 5.8 - Requisitos arquiteturais da camada de persistência**

Camada de Persistência		
ID	Requisitos	Serviço
RA[1]	A arquitetura de referência deve armazenar os dados produzidos pelos serviços que compõem o repositório de ativos de software	Serviço de Persistência

Legenda: RA refere-se à um Requisito Arquitetural

**Quadro 5.9 - Requisitos arquiteturais da camada de orquestração**

<b>Camada de Orquestração</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviço</b>
RA-S[1]	A arquitetura de referência deve possibilitar que repositórios de ativos de software desenvolvidos para persistir diferentes tipos de ativos possam ser facilmente integrados	Serviço de Orquestração
RA-S[2]	A arquitetura de referência deve possibilitar que repositórios de ativos de software implementados em linguagens de programação distintas e sob diferentes plataformas possam ser facilmente integrados	Serviço de Orquestração
RA-S[4]	A arquitetura de referência deve prover mecanismos para que repositórios de ativos de software orientados a serviço possam ser compostos por processos de negócio ou utilizados por aplicações cliente	Serviço de Orquestração
RA-S[9]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software escalável, capaz de evoluir de maneira incremental, por meio da composição de novas funcionalidades disponíveis na forma de serviços	Serviço de Orquestração
RA-S[10]	A arquitetura de referência deve possibilitar que serviços de repositório de ativos de software e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma	Serviço de Orquestração
RA-S[11]	A arquitetura de referência deve possibilitar que serviços do repositório de ativos de software possam interagir diretamente ou por meio do uso de barramentos de serviço	Serviço de Orquestração
RA-S[12]	A arquitetura de referência deve possibilitar que repositório de ativos de software disponíveis como serviços, desenvolvidos sem o uso dessa arquitetura, possam ser integrados aos repositórios de ativos de software desenvolvidos segundo a arquitetura de referência	Serviço de Orquestração
RA-S[13]	A arquitetura de referência deve possibilitar sua instanciação parcial, ou seja, repositórios de ativos de software desenvolvidos a partir dessa arquitetura podem ser construídos sem a necessidade de implementação de todos os módulos propostos	Serviço de Orquestração
RA-S[14]	A arquitetura de referência deve permitir que os repositórios de ativos de software possam evoluir ao longo de diferentes versões de maneira simples, ou seja, mudanças de versão dos repositórios de ativos de software não devem afetar a interação com os serviços que a utilizam	Serviço de Orquestração

Legenda: RA-S refere-se à um Requisito Arquitetural do contexto de Serviço

**Quadro 5.10 - Requisitos arquiteturais da camada de intermediação**

<b>Camada de Intermediação</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviço</b>
RA-S[3]	A arquitetura de referência deve prover mecanismos para que repositórios de ativos de software na forma de serviços possam ser publicados e posteriormente descobertos por aplicações cliente	Serviço de intermediação
RA-S[6]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software que disponibilizem descrições semânticas, permitindo assim sua classificação nos repositórios de serviços	Serviço de intermediação
RA-S[10]	A arquitetura de referência deve possibilitar que serviços de repositório de ativos de software e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma	Serviço de intermediação

Legenda: RA-S refere-se à um Requisito Arquitetural do contexto de Serviço

**Quadro 5.11 - Requisitos arquiteturais da camada de apresentação**

<b>Camada de Apresentação</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviço</b>
RA-S[5]	A arquitetura de referência deve viabilizar o desenvolvimento de repositórios de ativos de software que disponibilizem informações sobre suas características e direções normativas de uso, por meio de descrições padronizadas	Serviço de Apresentação
RA-S[6]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software que disponibilizem descrições semânticas, permitindo assim sua classificação nos repositórios de serviço	Serviço de Apresentação
RA-S[7]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software que tenham à disposição informações e documentos relacionados às suas características de qualidade	Serviço de Apresentação
RA-S[10]	A arquitetura de referência deve possibilitar que serviços de repositório de ativos de software e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma	Serviço de Apresentação

Legenda: RA-S refere-se à um Requisito Arquitetural do contexto de Serviço

**Quadro 5.12 - Requisitos arquiteturais da camada de qualidade de serviço**

<b>Camada de Qualidade de Serviço</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviço</b>
RA-S[7]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software que tenham à disposição informações e documentos relacionados às suas características de qualidade	Serviço de Qualidade de Serviço
RA-S[8]	A arquitetura de referência deve prover mecanismos para a captura, monitoramento, registro e sinalização do não cumprimento de requisitos de qualidade estabelecidos entre serviços provedores e serviços clientes	Serviço de Qualidade de Serviço

Legenda: RA-S refere-se à um Requisito Arquitetural do contexto de Serviço

#### 5.4.2 Visão de módulo

A visão de módulo da Cambuci é ilustrada na Figura 5.4 e representa a arquitetura de referência por meio de pacotes, sub-pacotes, classes e interfaces. Esses elementos descrevem as unidades que implementam as funcionalidades representadas pelas partes que compõem a visão de módulo. Ela é descrita em UML e foi construída tomando como base a visão de módulo da arquitetura RefTEST-SOA proposta por Oliveira (2011); bem como os conceitos e os requisitos arquiteturais do Cambuci, definidos na Seção 5.3, e as camadas da visão geral da Cambuci, definidas na Seção 5.4.1.

Para cada categoria de serviço de cada camada da visão geral da Cambuci foi identificado um pacote correspondente para compor a visão de módulo, conforme ilustrado no Quadro 5.13. Os serviços de cada categoria foram definidos como sub-pacotes do pacote correspondente à categoria.

De acordo com o Quadro 5.13, verifica-se que os serviços relacionados aos repositórios de ativos de software, descritos na camada de aplicação da visão geral da

Cambuci, são representados na visão de módulo pelos pacotes: *ServiçosPrimáriosRepositórioAtivoSoftware*, *ServiçosOrtogonaisSuporte*, *ServiçosOrtogonaisOrganizacionais* e *ServiçosOrtogonaisGerais*. O pacote de *ServiçosPrimáriosRepositórioAtivoSoftware* é subdividido em dois pacotes: o pacote de *EspecificaçãoDeAtivo* e o pacote de *GestãoDeAtivos*.

**Quadro 5.13 – Pacotes e sub-pacotes da visão de módulo da Cambuci**

Camadas da visão geral da Cambuci	Pacotes	Sub-pacotes
Aplicação	<i>ServiçosPrimáriosRepositórioAtivoSoftware</i>	<i>EspecificaçãoDeAtivo</i> , <i>GestãoDeAtivo</i>
	<i>ServiçosOrtogonaisSuporte</i>	<i>ControleDependência</i> , <i>Versionamento</i> , <i>GerênciaConfiguração</i>
	<i>ServiçosOrtogonaisOrganizacionais</i>	<i>PlanejamentoGestão</i> , <i>Comunicação</i> , <i>Métrica</i>
	<i>ServiçosOrtogonaisGerais</i>	<i>Segurança</i> , <i>Persistência</i>
Persistência	tratada pelo sub-pacote <i>Persistência</i> do pacote <i>ServiçosOrtogonaisGerais</i>	-
Orquestração	<i>MediadorServiço</i>	<i>RegistroServiço</i> , <i>EscalonadorServiço</i> , <i>AgenteServiço</i>
Intermediação	<i>MediadorServiço</i>	<i>RegistroServiço</i> , <i>EscalonadorServiço</i> , <i>AgenteServiço</i>
Apresentação	<i>Apresentação</i>	<i>EngineServiço</i> , <i>Controlador</i>
Qualidade de Serviço	<i>QualidadeDeServiço</i>	-

Os serviços do pacote *Serviços Ortogonais* podem ser utilizados para apoiar os *ServiçosPrimáriosRepositórioAtivoSoftware*, enquanto que os serviços do pacote *MediadorServiço* permite a interação com os serviços relacionados aos repositórios de ativos de software. Já o pacote de *Apresentação* é responsável pelas solicitações de serviços (pacote *EngineServiço*), e por tratar essas requisições de serviço (pacote *Controlador*).

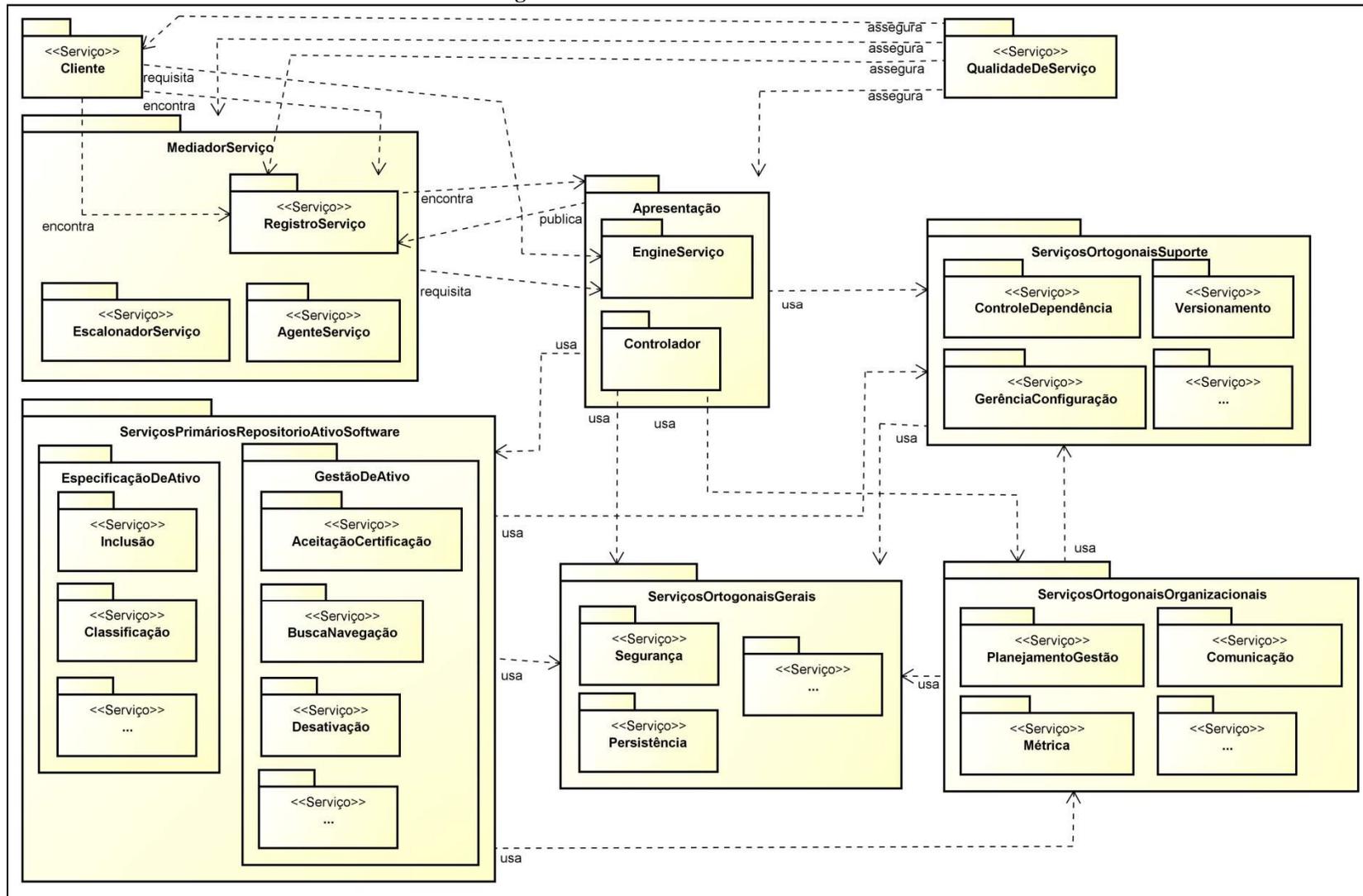
O pacote *Cliente* representa como os clientes podem fazer uso dos serviços disponíveis nos repositórios de ativos de software, e o pacote *QualidadeDeServiço* fornece serviços que permitem a inspeção da qualidade dos serviços oferecidos.

Devido à restrição de espaço na Figura 5.4 para representar todos os serviços de cada um dos pacotes, o Quadro 5.14 foi elaborado. Nesse quadro são apresentados os serviços de cada pacote.

**Quadro 5.14 - Serviços de cada pacote da visão de módulo da Cambuci**

<b>Pacotes</b>	<b>Sub-Pacotes</b>	<b>Serviços</b>
Cliente	-	Representar o cliente
QualidadeDeServiço	-	Assegurar a qualidade dos serviços
MediadorServiço	RegistroServiço	Publicar Serviço, pesquisar serviços publicados
MediadorServiço	EscalonadorServiço	Processar requisição de serviço
MediadorServiço	AgenteServiço	Mediar os serviços cliente e os provedores desses serviços
Apresentação	EngineServiço	Processar requisição de serviço
Apresentação	Controlador	Tratar requisição de serviço
ServiçosOrtogonais Suporte	ControleDependência	Realizar o controle de dependência entre ativos de software
ServiçosOrtogonais Suporte	Versionamento	Realizar o versionamento do ativo
ServiçosOrtogonais Suporte	GerênciaConfiguração	Realizar a tarefa de gerência de configuração do ativo
ServiçosOrtogonais Gerais	Segurança	Controlar o acesso e a autenticação
ServiçosOrtogonais Gerais	Persistência	Realizar a persistência
ServiçosOrtogonais Organizacionais	PlanejamentoGestão	Realizar o planejamento e gestão do ativo
ServiçosOrtogonais Organizacionais	Comunicação	Registrar o <i>feedback</i> dos usuários, notificar aos usuários sobre mudanças, divulgar informações sobre o reúso
ServiçosOrtogonais Organizacionais	Métrica	Registrar métricas de utilização do ativo
ServiçosPrimários RepositórioAtivo Software/ EspecificaçãoDeAtivo	Inclusão	Incluir um ativo
ServiçosPrimários RepositórioAtivo Software/ EspecificaçãoDeAtivo	Classificação	Classificar um ativo
ServiçosPrimários RepositórioAtivo Software/ GestãoDeAtivo	AceitaçãoCertificação	Aceitar e certificar o ativo, tornando-o disponível para reúso
ServiçosPrimários RepositórioAtivo Software/ GestãoDeAtivo	BuscaNavegação	Buscar um ativo, navegar entre ativos
ServiçosPrimários RepositórioAtivo Software/ GestãoDeAtivo	Desativação	Indisponibilizar um ativo

Figura 5.4 - Visão de módulo da Cambuci



### 5.4.3 Visão em tempo de execução

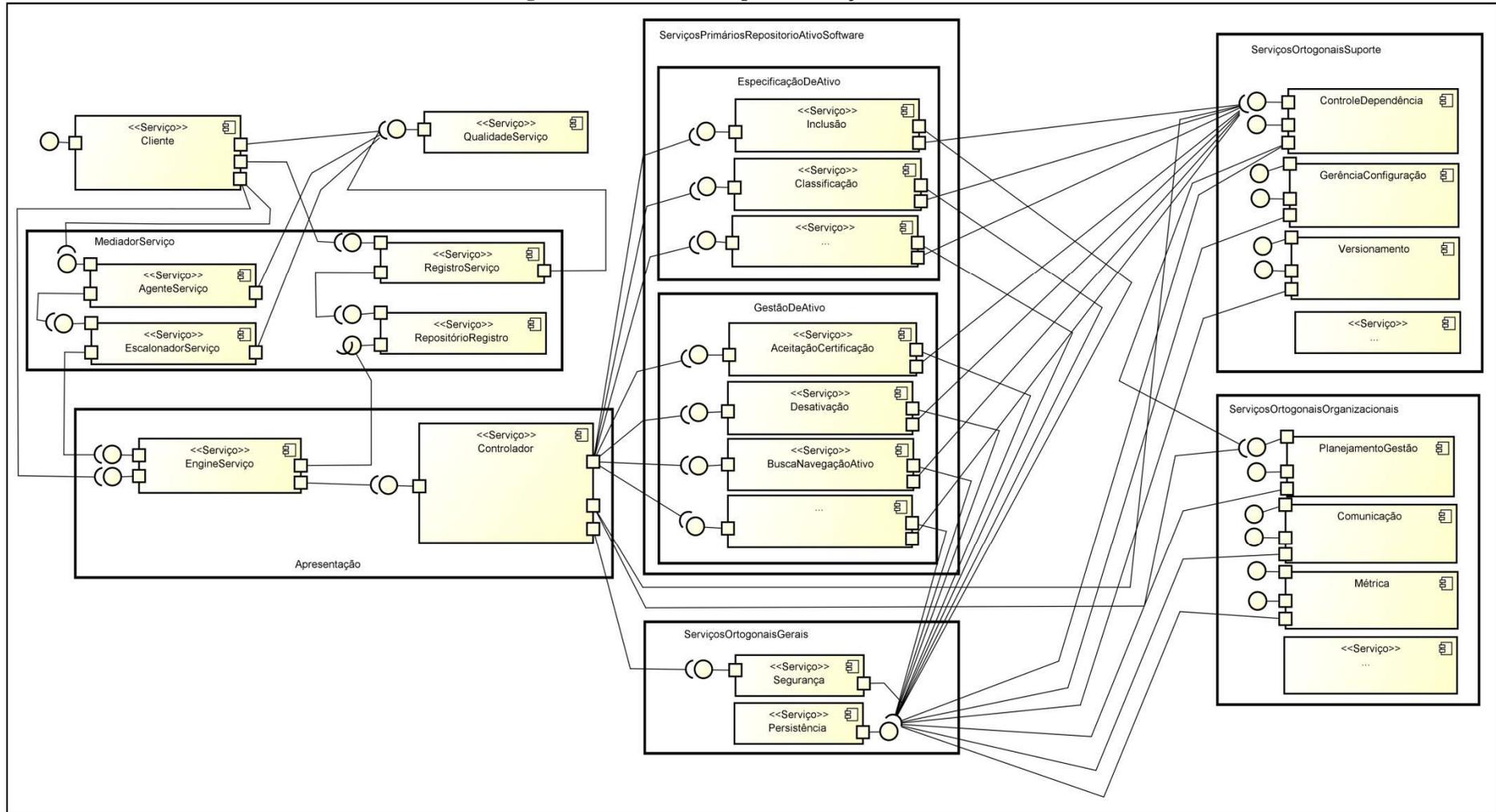
A visão em tempo de execução da Cambuci, apresentada na Figura 5.5, representa a arquitetura de referência por meio de componentes em diferentes níveis de abstração e pelas interfaces providas e requeridas. Essa visão apresenta a estrutura do sistema no momento em que ele está sendo executado, no caso, um repositório de ativos de software, mostrando as interações entre os componentes que compõem o sistema. Essa visão foi abstraída a partir da visão em tempo de execução da arquitetura RefTEST-SOA de Oliveira (2011).

Os pacotes identificados na visão de módulo correspondem aos componentes presentes na visão de tempo de execução. Após analisar as possíveis interações que possam acontecer entre esses componentes, essas interações são ilustradas na Figura 5.5. Nesta figura, o símbolo  indica que o componente está provendo o serviço, enquanto que o símbolo  indica que o componente está requisitando o serviço.

Os componentes pertencentes aos `ServiçosPrimárioRepositórioAtivoSoftware`, `EspecificaçãoDeAtivo` e `GestãoDeAtivo`, são os responsáveis por fornecer as funcionalidades principais dos repositórios de ativos de software.

O `Serviço Cliente` pode buscar informações sobre um serviço disponível por meio de uma pesquisa no `RegistroServiço`, que retorna o endereço do serviço solicitado. Através deste endereço, a comunicação pode ser realizada diretamente com a `EngineServiço` ou indiretamente através do `AgenteServiço`. A qualidade dos serviços é garantida pelo componente `QualidadeServiço`, que monitora a comunicação entre os serviços clientes e todos os demais componentes que proveem serviços, que são os serviços provedores. A `EngineServiço` encaminha as requisições ao `Controlador` que por sua vez direciona para algum componente relacionado às funcionalidades de repositórios de ativos de software (`ServiçosPrimáriosRepositórioAtivoSoftware`, `ServiçosOrtogonaisSuporte`, `ServiçosOrtogonaisOrganizacionais` e `ServiçosOrtogonaisGerais`).

Figura 5.5 - Visão em tempo de execução da Cambuci



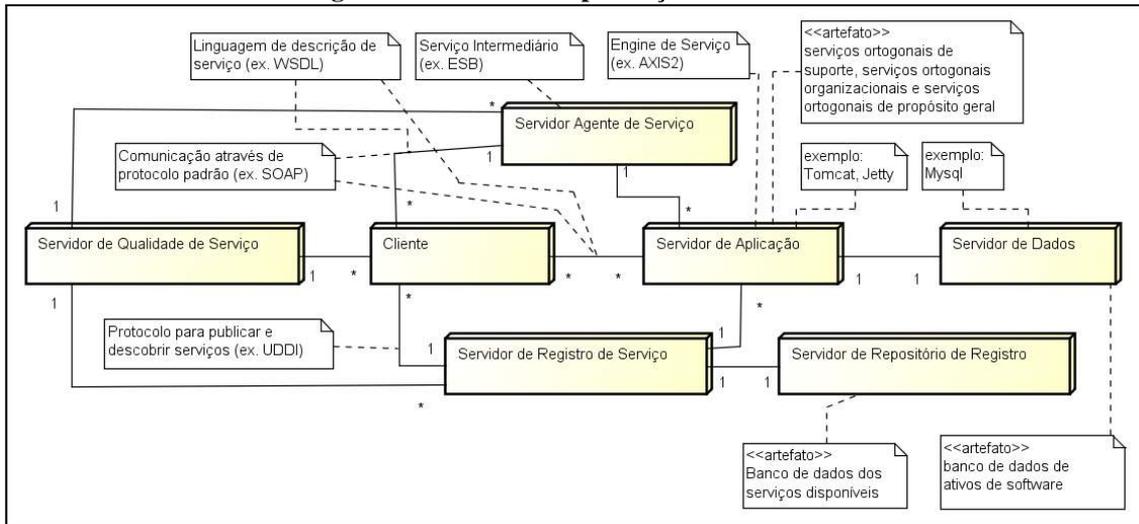
Nesta visão de tempo de execução, é visualizado que o serviço `ControleDependência`, localizado no componente `ServiçosOrtogonaisDeApoio` pode receber requisição de serviço de qualquer serviço localizado no componente de `ServiçosPrimáriosRepositórioAtivoSoftware` e do Controlador e o serviço de `ControleDependência` pode requisitar serviços para algum componente do módulo de `ServiçosOrtogonaisOrganizacionais` e para algum serviço do componente de `ServiçosOrtogonaisGerais`.

#### **5.4.4 Visão de implantação**

A visão de implantação da Cambuci, apresentada na Figura 5.6, representa a arquitetura de referência por meio da estrutura de hardware sobre a qual os sistemas são alocados e as conexões de rede que simbolizam as interações entre esses dispositivos. Essa visão de implantação é igual a visão apresentada na arquitetura RefTEST-SOA de Oliveira (2011) devido à abordagem orientada a serviços presente nas arquiteturas de referência RefTEST-SOA e Cambuci.

Nesta visão são considerados os seguintes elementos: (i) Servidor de Aplicação, que contém os serviços primários de repositórios de ativos e os serviços ortogonais (de suporte, transacionais e gerais). Também pode conter neste servidor a *Engine* de Serviços; (ii) Servidor de Dados, que é o responsável por armazenar os dados produzidos pelos serviços encontrados no Servidor de Aplicação; (iii) Servidor de Registro de Serviços, que gerencia informação sobre os serviços disponíveis de repositórios de ativos de software e disponibiliza a pesquisa a ser realizada pelos Serviços Cliente; (iv) Servidor de Repositório de Registro, que armazena os registros de serviços; (v) Servidor de Agente de Serviço, que armazena aplicações relacionadas ao funcionamento do agente de serviços; (vi) Servidor de Qualidade de Serviço, que armazena as aplicações relacionadas à garantia da qualidade e auditoria dos serviços disponíveis.

**Figura 5.6 - Visão de implantação da Cambuci**



### 5.5 Passo 4: Avaliação da arquitetura de referência

A avaliação da arquitetura de referência objetiva garantir a qualidade da arquitetura proposta e pode ser conduzida por meio de um *checklist* (NAKAGAWA, 2014). A avaliação da arquitetura de referência Cambuci é realizada por meio de uma especialização da mesma, com o intuito de obter a arquitetura de referência Cambuci-LPN (Capítulo 6), que consiste da arquitetura de referência orientada a serviços para repositórios de LPN sendo a principal contribuição deste trabalho.

### 5.6 Considerações finais

Utilizando a ontologia de especificação e gestão de ativos reutilizáveis ONTO-ResAsset como fonte de informação para o domínio de repositórios de ativos de software e a RefTEST-SOA como fonte de informação para o contexto de serviço, neste capítulo foi estabelecido a Cambuci, que é uma arquitetura de referência orientada a serviços para repositórios de ativos de software. Essa arquitetura de referência pode ser especializada para repositórios de quaisquer tipos de ativos de software, como repositório de requisitos, repositório de código fonte, repositório de teste, etc.

Para obter a Cambuci foram seguidos os passos definidos pelo processo ProSA-RA. No primeiro passo foi realizada investigação de fontes de informação úteis para a concepção da arquitetura de referência, selecionando-se a ontologia ONTO-ResAsset e a arquitetura de referência RefTEST-SOA. Nesse passo, os requisitos arquiteturais no contexto de serviços da RefTEST-SOA foram adaptados para o domínio de ferramentas de engenharia de software.

No segundo passo foram obtidos os requisitos arquiteturais do domínio de repositórios de ativos de software e os requisitos arquiteturais no contexto de serviços para repositórios de ativos de software. Esses últimos foram definidos a partir da especialização dos requisitos arquiteturais do contexto de serviço para ferramentas de engenharia de software. No terceiro passo foi realizado o projeto arquitetural da Cambuci por meio da criação de quatro visões arquiteturais, sendo a visão geral, a visão de módulo, a visão em tempo de execução e a visão de implantação. No quarto e último passo foi realizado por meio da especialização da Cambuci, apresentada no próximo capítulo, com o intuito de obter a arquitetura de referência orientada a serviços Cambuci-LPN.

## Capítulo 6 - Estabelecimento da Cambuci-LPN

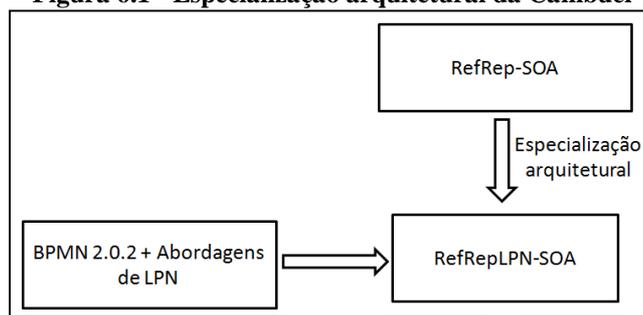
### 6.1 Considerações iniciais

Nesse capítulo é apresentada a arquitetura de referência orientada a serviços para repositórios de LPN, denominada Cambuci-LPN. O objetivo dessa arquitetura de referência é apoiar a fase de projeto arquitetural durante o desenvolvimento de repositórios de LPN.

A definição da Cambuci-LPN é baseada na especialização da arquitetura de referência Cambuci, definida no Capítulo 5, que é uma arquitetura de referência orientada a serviços para repositórios de ativos de software. A especialização da Cambuci para a definição da Cambuci-LPN é ilustrada na Figura 6.1.

Na definição da Cambuci-LPN é utilizado o ProSA-RA, que é um processo que sistematiza o desenvolvimento e a avaliação de arquiteturas de referências por meio de quatro passos bem definidos, descritos na Seção 2.4 do Capítulo 2.

**Figura 6.1 - Especialização arquitetural da Cambuci**



Neste capítulo é apresentada a execução de cada passo do ProSA-RA para a definição da Cambuci-LPN. Na Seção 6.2 é realizada a investigação das fontes de informação utilizadas para o entendimento do domínio de LPN. Para auxiliar no entendimento do domínio, os conceitos identificados são representados em modelos conceituais utilizando a notação do diagrama de classes da UML. Na Seção 6.3 é apresentado o conjunto de requisitos arquiteturais identificados a partir das fontes de informação. Na Seção 6.4 é mostrada e descrita a representação da arquitetura de referência Cambuci-LPN, sendo composta por uma visão geral da arquitetura e por mais quatro visões arquiteturais. Na Seção 6.5 é conduzida a avaliação da arquitetura de referência Cambuci-LPN. Na Seção 6.6 são apresentadas as considerações finais deste capítulo.

## **6.2 Passo 1: Investigação de fontes de informação**

Durante o primeiro passo do ProSA-RA, para a especialização da arquitetura de referência Cambuci visando a concepção da Cambuci-LPN, foram identificadas diferentes fontes para obtenção de informações sobre o domínio de repositórios de LPN. Neste passo foram considerados os seguintes grupos de fontes de informação: (i) especificação BPMN; (ii) estudos sobre abordagens de LPN; e (iii) arquitetura de referência Cambuci.

Nas subseções a seguir são apresentados os estudos presentes em cada grupo de fonte de informação utilizados para a definição da Cambuci-LPN.

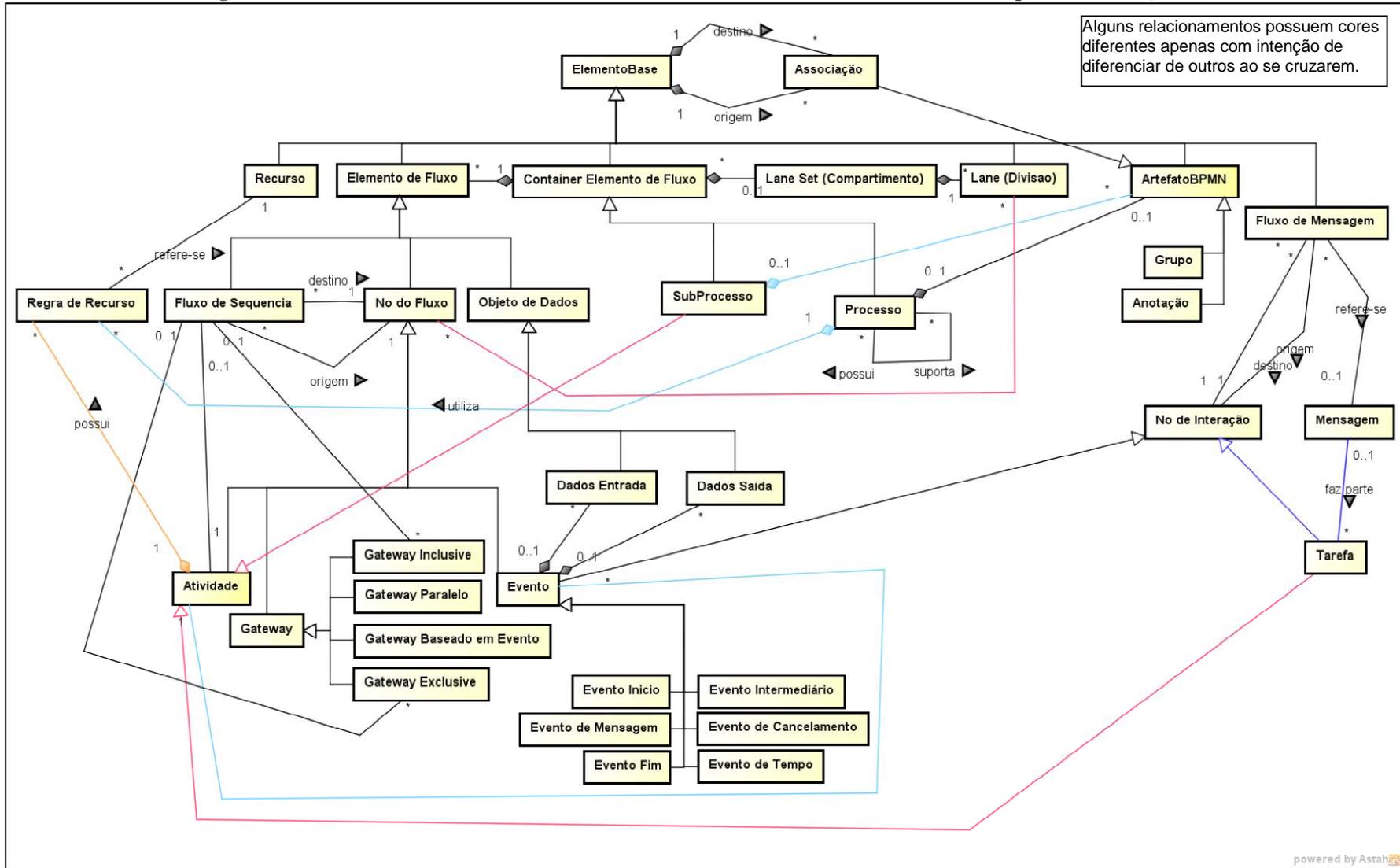
### **6.2.1 Grupo 1: Especificação BPMN**

O estudo utilizado neste grupo foi a especificação BPMN versão 2.0.2 (OMG, 2013). A BPMN é a modelagem padrão da OMG para a representação de processos de negócio e é a notação que foi tomada como base para a definição da Cambuci-LPN.

Os principais conceitos deste estudo foram os conceitos correspondentes aos elementos de modelagem básica do BPMN, de acordo com a OMG (2013). Porém, alguns elementos estendidos da BPMN, que não pertencem à modelagem básica, também são utilizados, como atividade, subprocesso, tarefa e alguns *gateways* (inclusivo, paralelo, baseado em evento, exclusivo); pois de acordo com Schnieders e Puhlman (2006) podem ter variabilidades. Os elementos do BPMN utilizados neste trabalho são descritos na Seção 3.2 do Capítulo 3.

Na Figura 6.2 encontra-se o modelo conceitual extraído a partir dos diagramas de classes da OMG (2013) referentes aos elementos do BPMN considerados neste trabalho. O `ElementoBase` pode ser um `Recurso`, um `Elemento de Fluxo`, um `Container de Elemento de Fluxo`, um `Lane`, um `ArtefatoBPMN` ou um `Fluxo de Mensagem`. Já, um `Elemento de Fluxo` pode ser um `Fluxo de Sequência`, um `Nó de Fluxo` ou um `Objeto de Dados`. O `Container de Elemento de Fluxo` pode ser um `Processo` ou um `SubProcesso`. O `Nó de Fluxo` pode ser uma `Atividade`, um `Gateway` ou um `Evento`. O estudo da especificação da notação BPMN para identificar seus principais conceitos é justificado porque, além de ser atualmente a notação padrão para a modelagem de processos de negócio, há uma extensão dessa notação que pode ser utilizada para a modelagem de LPN, no caso a notação `vrBPMN*` (Landre *et al.*, 2014).

Figura 6.2 – Modelo conceitual dos elementos do BPMN 2.0.2 utilizados neste trabalho (adaptado de OMG, 2013)



## 6.2.2 Grupo 2: Abordagens de linhas de processos de negócio

Os estudos deste grupo foram identificados a partir do mapeamento sistemático realizado por Landre (2012), selecionando somente os artigos cujo produto final das LPNs é processos de negócio baseados na visão de atividades, tratado pela abordagem GLPN. Essa foi a visão mais comum encontrada por Landre (2012) e pode ser representada utilizando o BPMN. Também foi acrescentada a este grupo a abordagem para gestão de LPN proposta por Landre (2012), pois foi desenvolvida pelo mesmo grupo de pesquisa deste trabalho de mestrado e é de interesse do mesmo. Os trabalhos considerados como fonte de informação deste grupo estão citados no Quadro 6.1.

**Quadro 6.1 - Trabalhos selecionados sobre abordagens de LPN**

	<b>Autor(es)</b>	<b>Título</b>
1	Landre (2013)	GLPN – Uma Abordagem para Gestão de Linhas de Processos de negócio
2	Boffoli <i>et al</i> (2012)	<i>Business Process Lines and Decision Tables Driving Flexibility by Selecion</i>
3	Rosa <i>et al</i> (2010)	<i>Configurable multi-perspective business process models</i>
4	Ludwig (2011)	<i>Process variation analysis using empirical methods a case study</i>
5	Gröner (2011)	<i>Validation of families of business processes</i>

A partir da leitura e entendimento dos estudos deste grupo foram identificados os principais conceitos sobre LPN, os quais estão elencados no Quadro 6.2. Como em alguns casos os autores das abordagens de LPN utilizam termos distintos, optou-se por estabelecer um conceito unificado para termos equivalentes, apresentado na última coluna deste quadro. Esses conceitos são representados posteriormente em um modelo conceitual que é tomado como base para a concepção da arquitetura de referência Cambuci-LPN.

**Quadro 6.2 - Conceitos de LPN do grupo 2**

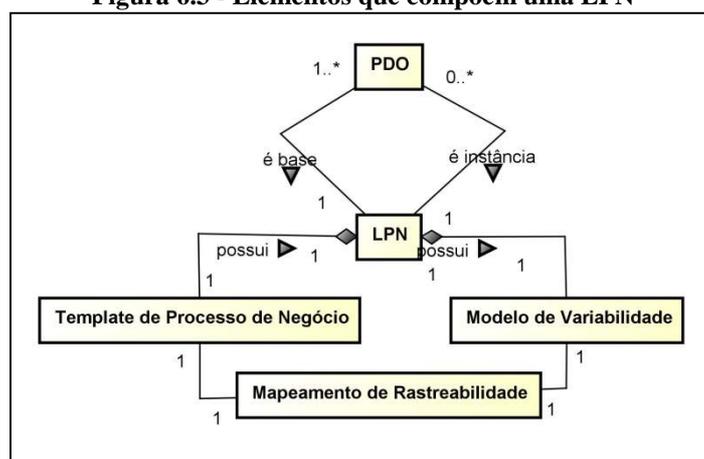
	<b>Autores</b>					<b>Conceito adotado</b>
	<b>Landre (2012)</b>	<b>Boffoli <i>et al.</i> (2012)</b>	<b>Rosa <i>et al.</i> (2010)</b>	<b>Ludwing (2011)</b>	<b>Gröner (2011)</b>	
Linha de processo de negócio	LPN	LPN	modelo de processo configurável	padronização de processos de negócio	família de processo de negócio	LPN
Representação das variabilidades	modelo ortogonal	modelo de característica/ tabela de decisão	Questionário	-	lógica descritiva	Modelo de variabilidade
Notação para representar as variabilidades de processos de negócio	vrBPMN	-	C-IEPC	BPMN/BPEL	BPMN e modelo de características	<i>Template</i> de Processo de Negócio

**Quadro 6.2 (conclusão)**

	Autores					Conceito adotado
	Landre (2012)	Boffoli et al. (2012)	Rosa et al. (2010)	Ludwing (2011)	Gröner (2011)	
Elementos para representar as variabilidades	Ponto de Variação	-	Ponto de Variação	-	-	Ponto de Variação
	Variante	Variante	Variante	Variante	Variante	Variante
	Variante exclui Ponto de Variação	-	-	-	-	Variante exclui Ponto de Variação
	Variante exclui Variante	-	-	-	-	Variante exclui Variante
	Variante requer Ponto de Variação	-	-	-	-	Variante requer Ponto de Variação
	Variante requer Variante	-	-	-	-	Variante requer Variante

A partir de definições sobre LPN estudadas na literatura (LANDRE, 2012; BOFFOLI et al., 2012; ROSA et al., 2010; LUDWING, 2011; e GRÖNER, 2011) e dos conceitos de LPN obtidos (Quadro 6.2), identificou-se que uma LPN é composta por um Modelo de Variabilidade, por um Template de Processos de Negócio e por um Mapeamento de Rastreabilidade entre esses dois artefatos, o qual representa a rastreabilidade existente entre eles, como ilustrado na Figura 6.3. Esse mapeamento pode ser realizado em um artefato distinto (BOFFOLI et al., 2012) ou no próprio Template de Processos de Negócio (LANDRE, 2012).

**Figura 6.3 - Elementos que compõem uma LPN**

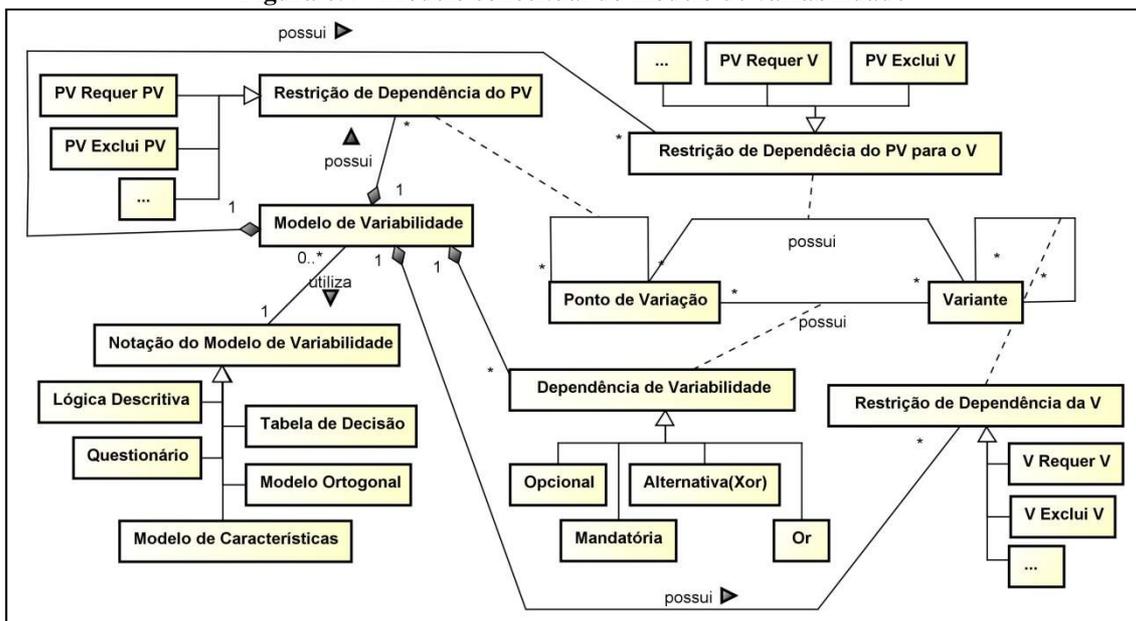


Para produzir uma LPN sugere-se considerar como base pelo menos um PDO (processos do domínio de negócio), quando a opção for construir uma LPN a partir de

apenas um PDO e, posteriormente evoluir esta LPN. Por outro lado, pode-se utilizar como base para a construção da LPN pelo menos três PDOs para facilitar a análise de domínio e a identificação das variabilidades. A partir da LPN constituída, podem ser gerados vários PDO's, que serão instâncias da LPN.

Na Figura 6.4 é ilustrado o modelo conceitual dos conceitos importantes relacionados ao modelo de variabilidade da LPN. Esse modelo foi construído tomando como base as abordagens estudadas neste grupo e cujos principais conceitos estão apresentados no Quadro 6.2. Os elementos do modelo ortogonal e elementos do modelo de características (notações utilizadas para representar o modelo de variabilidades), que estão presentes nesses estudos, foram utilizados para auxiliar na elaboração do modelo conceitual.

**Figura 6.4 - Modelo conceitual do modelo de variabilidade**



O Modelo de Variabilidade pode ser representado por várias notações, como exemplos: Lógica Descritiva, Tabela de Decisão, Questionário, Modelo Ortogonal e Modelo de Características. No Modelo de Variabilidade, tem-se as restrições (Restrição de Dependência do PV, Restrição de Dependência do PV para o V e Restrição de Dependência da V). Também existe a dependência de variabilidade entre um Ponto de Variação e as suas Variantes, que podem ser Opcional, Alternativa(XOR), Mandatória e OR. Por exemplo, em uma venda, a forma de pagamento é uma característica necessário, ou seja, o ponto de variação “forma de pagamento” possui variantes mandatórias. Se nessa dependência puder escolher apenas

uma forma de pagamento, então as formas de pagamento “em dinheiro”, “no cartão”, “no boleto” são alternativas entre si. Se puder escolher mais de uma forma de pagamento, possuem dependência de variação do tipo OR.

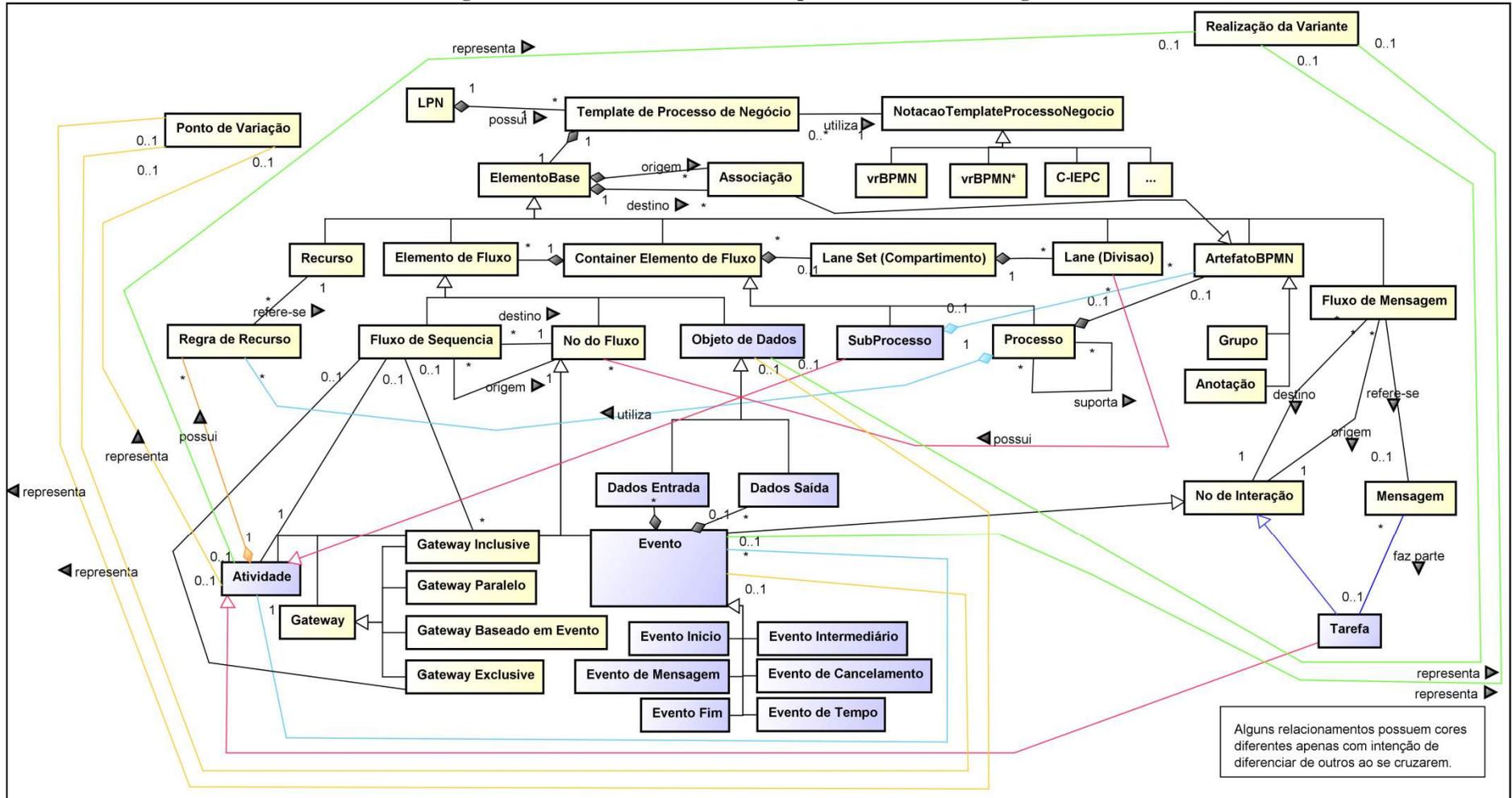
As Restrições de Dependência (Restrição de Dependência do PV, Restrição de Dependência do PV para o V, Restrição de Dependência da V), podem ser do tipo `Requer` ou `Excluir`. Por exemplo, a Restrição de Dependência do PV pode ser `PV Requer PV`, a qual um ponto de variação requer outro ponto de variação, ou pode ser `PV Exclui PV`, a qual um ponto de variação exclui um outro ponto de variação.

Na Figura 6.5 é ilustrado o modelo conceitual que representa os conceitos que fazem parte do *Template* de Processos de Negócio da LPN. A elaboração desse modelo conceitual é baseada no estudo dos trabalhos de Rosa *et al.* (2010), Ludwing (2011), Gröner (2011) e Landre (2012) do grupo 2 da fonte de informação. Neste trabalho, considera-se que o *Template* de Processos de Negócio esteja representado na notação `vrBPMN` (SCHINEDER E PUHLMANN, 2006) ou na notação `vrBPMN*` pois são baseadas no BPMN, e portanto é composto por diversos `ElementosBase` do BPMN 2.0.2.

Utilizando o `vrBPMN` como fonte de informação para apoiar na definição dos pontos de variação e das variantes em um modelo de processos de negócio representado em BPMN, definiu-se que os conceitos que podem ser `Ponto de Variação` ou `Realização de Variante` são as `Atividades` (Schnieders e Puhlman, 2006), bem como `SubProcessos` e `Tarefas`, que são conceitos que herdam das `Atividades`. Outros conceitos que foram adicionados como `Ponto de Variação` ou `Realização de Variante` são os `Eventos` e os `Objetos de Dados`.

Após o entendimento do domínio de LPN, foi realizado um levantamento das funcionalidades que devem ser contempladas por repositórios de LPN. Para isso, foi feita uma seleção das abordagens de LPN estudadas no grupo 2 considerando como critério de seleção aquelas que se preocupam com repositórios. A partir disso, apenas a abordagem GLPN foi selecionada. Para a identificação das funcionalidades foi feito um estudo de cada fase da GLPN e procurando identificar como um repositório de LPN pode apoiar essa fase.

Figura 6.5 - Modelo conceitual do Template de Processos de Negócio



A partir desse estudo, verificou-se que os principais conceitos envolvidos com repositórios de LPN são:

- Especificação da LPN: criação da LPN;
- Gestão da LPN: gerenciamento do ciclo de vida da LPN;
- Instanciação da LPN: criação de um PDO a partir da LPN; e
- Rastreabilidade entre LPN e suas Instâncias: manter a rastreabilidade entre as diferentes versões da LPN e entre as suas instâncias.

No Quadro 6.3 são listadas as funcionalidades para repositórios de LPN encontradas nesse estudo. Neste quadro, cada funcionalidade é relacionada com os conceitos de repositórios de LPN, sendo que esses conceitos são classificados como serviços primários e serviços primários organizacionais, conforme classificação utilizada por Oliveira (2011) e utilizada na concepção da Cambuci. Essa classificação é também utilizada no estabelecimento da arquitetura de referência Cambuci-LPN.

Em especial, a funcionalidade *Feedback* é considerada como um serviço ortogonal organizacional e está relacionada com o conceito de Comunicação.

**Quadro 6.3 Funcionalidades identificadas nas abordagens LPN**

<b>Funcionalidade</b>	<b>Descrição</b>	<b>Conceito</b>	<b>Serviços</b>
Salvar LPN	Permite salvar uma LPN	Especificação da LPN	Serviços primários
Salvar Instância da LPN	Permite salvar uma instância da LPN	Instanciação da LPN	Serviços primários
Gerenciar uma LPN	Permite propagar mudanças na LPN para as instâncias	Gestão da LPN	Serviços primários
Controlar versão entre LPN e instâncias	Permite que as instâncias mantenham sincronizadas com diferentes versões da LPN	Rastreabilidade entre LPN e suas Instâncias	Serviços primários
<i>Feedback</i>	Registrar impressões dos usuários a respeito da utilização da LPN	Comunicação	Serviços ortogonais organizacionais
Gerenciar mudanças ocorridas entre LPN e instâncias	Permite gerenciar a rastreabilidade bidirecional entre cada elemento da instância com o elemento correspondente na LPN	Rastreabilidade entre LPN e suas Instâncias	Serviços primários
Consultar diferenças entre versões de LPN	Permite identificar as diferenças existentes entre as versões de uma LPN	Rastreabilidade entre LPN e suas Instâncias	Serviços primários
Consultar diferenças entre instâncias de versões distintas de LPN	Permite identificar as diferenças entre instâncias de diferentes versões de uma LPN	Rastreabilidade entre LPN e suas Instâncias	Serviços primários

A funcionalidade Validação da Instância da LPN, a qual é verificada se uma instância foi corretamente construída não faz parte das funcionalidades de um repositório de LPN e sim de ferramentas computacionais de apoio a abordagens de LPN, como é o caso do *BLP framework* que é um arcabouço computacional de apoio a criação, instanciação e gerenciamento de LPNs (TERENCIANI, 2014), que está sendo desenvolvido para apoiar as fases da abordagem GLPN.

No Apêndice A é ilustrada como seria uma ferramenta computacional de apoio à abordagens LPN, para criar LPNs e suas instâncias correspondentes, a qual deve ser integrada com um repositório de LPN (que pode ser desenvolvido por meio da instanciação da Cambuci-LPN), fazendo o uso dos serviços disponibilizados por ele.

### **6.2.3 Grupo 3 - Requisitos arquiteturais da arquitetura de referência Cambuci**

A fonte de informação que faz parte deste grupo é basicamente os requisitos arquiteturais da Cambuci, apresentados na Seção 5.3 e baseados nos requisitos arquiteturais da RefTest-SOA. Esses requisitos são divididos em dois grupos, conforme citados a seguir, e são adaptados na Seção 6.3 para o contexto de repositório de LPN:

- Requisitos arquiteturais do domínio de repositório de ativos de software (apresentados na Seção 5.3.1 do Capítulo 5).
- Requisitos arquiteturais do contexto de serviços (apresentados na Seção 5.3.2 do Capítulo 5).

## **6.3 Passo 2 – Análise arquitetural**

Na execução do segundo passo do ProSA-RA foi realizada a elicitación dos requisitos arquiteturais da Cambuci-LPN. Para isso foram utilizados os conceitos identificados na Seção 6.2. Neste passo, os requisitos arquiteturais da Cambuci-LPN são divididos em dois grupos e apresentados nas subseções a seguir: requisitos arquiteturais do domínio de repositórios de LPN e requisitos arquiteturais no contexto de serviços.

### **6.3.1 Requisitos arquiteturais do domínio de repositórios de LPN**

A partir dos estudos do Grupo 2 e dos requisitos arquiteturais do domínio de repositórios de ativos de software do Grupo 3, mencionados na Seção 6.3.2 e descritos na íntegra na Seção 5.3, foram elicitados os requisitos arquiteturais para o domínio de repositórios de LPN.

Os conceitos identificados nos estudos do Grupo 2 que pertencem ao domínio de repositórios de LPN e utilizados durante a elicitação dos requisitos arquiteturais da Cambuci-LPN são:

**Especificação de LPN:** envolve todas as atividades necessárias para a criação da LPN, como exemplos a inclusão e a classificação de uma LPN;

**Gestão de LPN:** envolve todas as atividades necessárias para gerir o ciclo de vida de uma LPN, como aceitação, certificação e busca de uma LPN;

**Instanciação de LPN:** envolve todas as atividades necessárias para utilizar a LPN, criando uma instância, como a criação de uma LPN; e

**Rastreabilidade entre LPN e suas Instâncias:** envolve todas as atividades necessárias para manter a rastreabilidade entre diferentes versões da LPN e as versões de suas respectivas instâncias.

Outros conceitos utilizados para o levantamento dos requisitos arquiteturais da Cambuci-LPN pertencem ao domínio de repositórios de ativos de software que foram identificados na Seção 5.2.1.

No Quadro 6.4 são listados os requisitos arquiteturais relacionados aos repositórios de LPN que foram identificados. Os requisitos 1 a 17 foram extraídos a partir da adaptação dos requisitos arquiteturais para o domínio de repositório de ativos de software. Para isso, a descrição de cada um dos requisitos do domínio de repositórios de ativos de software foi adaptada para o contexto de repositórios de LPN. Exemplificando, o requisito arquitetural RA-AS[1] da Cambuci, “*A arquitetura de referência deve possibilitar que repositórios de ativos de software incluam um novo ativo, que pode ser composto por vários artefatos*”, foi adaptado para o requisito arquitetural RA-RLPN[1] da Cambuci-LPN como “*A arquitetura de referência deve possibilitar que repositórios de LPN incluam uma nova LPN, que pode ser composta por vários artefatos*”.

Na definição dos requisitos arquiteturais de 1 a 17, têm-se o reuso dos requisitos arquiteturais da arquitetura de referência Cambuci. Esses mesmos requisitos arquiteturais podem ser reutilizados na definição de outras arquiteturas de referência do domínio de repositórios de ativos de software, como exemplos, repositórios de requisitos de software, repositórios de testes de software, entre outros.

Os requisitos 18 a 23 foram extraídos tomando como base as funcionalidades de repositórios de LPN identificadas nos estudos do Grupo 2, sendo que as funcionalidades Salvar LPN, *Feedback* e Apoiar Validação da LPN já foram

mapeadas anteriormente e estão representadas nos requisitos RA-RLPN[1], RA-RLPN[12] e RA-RLPN[2], respectivamente.

Adicionalmente, no Quadro 6.4 é apresentada uma relação entre os requisitos arquiteturais para o domínio de repositórios de LPN, os conceitos do domínio, bem como a classificação de cada requisito arquitetural de acordo com os módulos Serviços Primários, Serviços Ortogonais de Suporte, Serviços Ortogonais Organizacionais e Serviços Ortogonais Gerais da visão geral da arquitetura de referência RefTest-SOA (Seção 2.5). Essa classificação foi baseada na classificação previamente realizada no passo 1 (Seção 6.2).

**Quadro 6.4 - Requisitos arquiteturais de repositórios de LPN**

<b>ID</b>	<b>Descrição</b>	<b>Conceitos</b>	<b>Serviços</b>
RA-RLPN[1]	A arquitetura de referência deve possibilitar que repositórios de LPN incluam uma nova LPN, que pode ser composta por vários artefatos;	Especificação da LPN	Serviço Primário
RA-RLPN [2]	A arquitetura de referência deve possibilitar que repositórios de LPN forneçam mecanismo para aceitação e certificação da LPN;	Gestão da LPN	Serviço Primário
RA-RLPN [3]	A arquitetura de referência deve possibilitar que repositórios de LPN desativem LPNs que não serão mais utilizadas;	Gestão da LPN	Serviço Primário
RA-RLPN [4]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam a classificação de uma LPN e também informar o contexto de sua utilização;	Especificação da LPN	Serviço Primário
RA-RLPN [5]	A arquitetura de referência deve possibilitar que repositórios de LPN registrem a dependência entre LPNs;	Especificação da LPN	Serviço Primário
RA-RLPN [6]	A arquitetura de referência deve possibilitar que repositórios de LPN notifiquem os interessados sobre mudanças que aconteçam na LPN;	Comunicação	Serviços Ortogonais Organizacionais
RA-RLPN [7]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam realizar buscas e recuperação das LPNs;	Gestão da LPN	Serviço Primário
RA-RLPN [8]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam a navegação entre LPNs;	Gestão da LPN	Serviço Primário
RA-RLPN [9]	A arquitetura de referência deve possibilitar que repositórios de LPN aceitem múltiplas fontes de origem de LPNs, com o objetivo de facilitar a integração entre equipes e entre repositórios diferentes;	Gestão da LPN	Serviço Primário
RA-RLPN [10]	A arquitetura de referência deve possibilitar que repositórios de LPN criem e armazenem múltiplas versões de uma mesma LPN;	Versionamento	Serviços Ortogonais de Suporte
RA-RLPN [11]	A arquitetura de referência deve possibilitar que repositórios de LPN gerenciem a configuração, como por exemplo, a definição dos itens da LPN que são itens de configuração, o controle de mudanças dos artefatos da LPN que são itens de configuração;	Gerência de Configuração	Serviços Ortogonais de Suporte

**Quadro 6.4 (conclusão)**

<b>ID</b>	<b>Descrição</b>	<b>Conceitos</b>	<b>Serviços</b>
RA-RLPN [12]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam o registro de impressões dos usuários a respeito da versão da LPN que eles utilizaram;	Comunicação	Serviços Ortogonais Organizacionais
RA-RLPN [13]	A arquitetura de referência deve possibilitar que repositórios de LPN registrem métricas coletadas sobre a utilização da LPN;	Métrica	Serviços Ortogonais Organizacionais
RA-RLPN [14]	A arquitetura de referência deve possibilitar que repositórios de LPN ofereçam informações relativas ao reuso, iniciativas de reuso, LPNs mais usadas, etc;	Comunicação	Serviços Ortogonais Organizacionais
RA-RLPN [15]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam o acesso de acordo com o papel que o usuário assume;	Segurança	Serviços Ortogonais Gerais
RA-RLPN [16]	A arquitetura de referência deve possibilitar que repositórios de LPN garantam a integridade das LPNs;	Persistência	Serviços Ortogonais Gerais
RA-RLPN [17]	A arquitetura de referência deve possibilitar que repositórios de LPN realizem o gerenciamento de transação, garantindo a atomicidade e o isolamento da transação, e a consistência e a durabilidade dos dados;	Persistência	Serviços Ortogonais Gerais
RA-RLPN[18]	A arquitetura de referência deve possibilitar que repositórios de LPN incluam uma nova instância da LPN;	Instanciação da LPN	Serviços Primários
RA-RLPN[19]	A arquitetura de referência deve possibilitar que repositórios de LPN comuniquem mudanças ocorridas na LPN para suas instâncias;	Comunicação	Serviços Ortogonais Organizacionais
RA-RLPN[20]	A arquitetura de referência deve possibilitar que repositórios de LPN propaguem a mudança na LPN para uma determinada instância;	Gestão da LPN	Serviços primários
RA-RLPN[21]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam que cada instância da LPN esteja vinculada com no mínimo uma versão da LPN e sincronizada com apenas uma delas;	Rastreabilidade entre LPN e suas Instâncias	Serviços primários
RA-RLPN[22]	A arquitetura de referência deve possibilitar que repositórios de LPN mantenham a rastreabilidade bidirecional entre cada elemento da instância da LPN com o elemento correspondente na LPN;	Rastreabilidade entre LPN e suas Instâncias	Serviços primários
RA-RLPN[23]	A arquitetura de referência deve possibilitar que repositórios de LPN identifiquem as diferenças entre as versões de uma LPN;	Rastreabilidade entre LPN e suas Instâncias	Serviços primários
RA-RLPN[24]	A arquitetura de referência deve possibilitar que repositórios de LPN identifiquem as diferenças entre instâncias de diferentes versões de uma LPN;	Rastreabilidade entre LPN e suas Instâncias	Serviços primários

Legenda: RA-RLPN refere-se à um Requisito Arquitetural de Repositórios de LPN

### 6.3.2 Requisitos arquiteturais do contexto de serviço

A partir dos arquiteturais do contexto de serviço presentes na Cambuci, mencionados na Seção 6.2.3 e descritos na íntegra na Seção 5.3.2, os requisitos arquiteturais do contexto de serviços da Cambuci-LPN são abstraídos. Para isso, as descrições dos requisitos arquiteturais do contexto de serviços da Cambuci foram alteradas para se adequarem ao contexto de repositórios de LPN. Exemplificando, o

requisito arquitetural RA-S[1] da Cambuci, “A arquitetura de referência deve possibilitar que repositórios de ativos de software desenvolvidos para persistir diferentes tipos de ativos possam ser facilmente integrados” foi adaptado para o requisito arquitetural RA-S[1] da Cambuci-LPN como “A arquitetura de referência deve possibilitar que repositórios de LPN desenvolvidos para persistir diferentes tipos de LPNs possam ser facilmente integrados”.

No Quadro 6.5 são listados os requisitos arquiteturais do contexto de serviços para repositórios de LPN visando a concepção da Cambuci-LPN. Nesse quadro é apresentada uma relação entre os requisitos arquiteturais para a definição da Cambuci-LPN, os conceitos do domínio de serviços e as camadas de serviços correspondentes. Sendo que os conceitos do domínio de serviços são os mesmos conceitos citados da Seção 5.3.2 e as camadas de serviços foram definidas por Oliveira (2011) na visão geral da arquitetura e são utilizadas no projeto arquitetural que se encontra na Seção 6.4.

**Quadro 6.5 - Requisitos arquiteturais do contexto de serviços para repositórios de LPN**

ID	Descrição	Conceitos	Serviços
RA-S[1]	A arquitetura de referência deve possibilitar que repositórios de LPN desenvolvidos para persistir diferentes tipos de LPNs possam ser facilmente integrados	Requisito Geral	Processo de negócio
RA-S[2]	A arquitetura de referência deve possibilitar que repositórios de LPN implementados em linguagens de programação distintas e sob diferentes plataformas possam ser facilmente integrados	Requisito Geral	Processo de negócio
RA-S[3]	A arquitetura de referência deve prover mecanismos para que repositórios de LPN na forma de serviços possam ser publicados e posteriormente descobertos por aplicações cliente	Publicação de Serviço	Intermediação
RA-S[4]	A arquitetura de referência deve prover mecanismos para que repositórios de LPN orientados a serviço possam ser compostos por processos de negócio ou utilizados por aplicações cliente	Interação entre Serviços, Composição de Serviços	Processo de negócio
RA-S[5]	A arquitetura de referência deve viabilizar o desenvolvimento de repositórios de LPN que disponibilizem informações sobre suas características e direções normativas de uso, por meio de descrições padronizadas	Descrição de Serviço	Apresentação
RA-S[6]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de LPN que disponibilizem descrições semânticas, permitindo assim sua classificação nos repositórios de serviço	Descrição de Serviço, Publicação de Serviço	Apresentação, Intermediação
RA-S[7]	A arquitetura de referência deve viabilizar o desenvolvimento de repositórios de LPN que tenham à disposição informações e documentos relacionados às suas características de qualidade	Descrição de Serviço, Qualidade de Serviço	Apresentação, Qualidade de Serviço
RA-S[8]	A arquitetura de referência deve prover mecanismos para a captura, monitoramento, registro e sinalização do não cumprimento de requisitos de qualidade estabelecidos entre serviços provedores e serviços clientes	Qualidade de Serviço	Qualidade de Serviço

**Quadro 6.5 (conclusão)**

<b>ID</b>	<b>Descrição</b>	<b>Conceitos</b>	<b>Serviços</b>
RA-S[9]	A arquitetura de referência deve viabilizar o desenvolvimento de repositórios de LPN escaláveis, capazes de evoluir de maneira incremental, por meio da composição de novas funcionalidades disponíveis na forma de serviços	Composição de Serviços	Processo de negócio
RA-S[10]	A arquitetura de referência deve possibilitar que serviços de repositórios de LPN e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma	Descrição de Serviço, Publicação de Serviço, Interação entre Serviços, Composição de Serviços	Apresentação, Intermediação, Processo de negócio
RA-S[11]	A arquitetura de referência deve possibilitar que serviços de repositórios de LPN possam interagir diretamente ou por meio do uso de barramentos de serviço	Interação entre Serviços	Processo de negócio
RA-S[12]	A arquitetura de referência deve possibilitar que repositórios de LPN disponíveis como serviços, desenvolvidos sem o uso dessa arquitetura, possam ser integrados aos repositórios de LPN desenvolvidos segundo a arquitetura de referência	Requisito Geral	Processo de negócio
RA-S[13]	A arquitetura de referência deve possibilitar sua instanciação parcial, ou seja, repositórios de LPN desenvolvidos a partir dessa arquitetura podem ser construídos sem a necessidade de implementação de todos os módulos propostos	Requisito Geral	Processo de negócio
RA-S[14]	A arquitetura de referência deve permitir que os repositórios de LPN possam evoluir ao longo de diferentes versões de maneira simples, ou seja, mudanças de versão dos repositórios de LPN não devem afetar a interação com os serviços que a utilizam	Interação entre Serviços, Composição de Serviços	Processo de negócio

Legenda: RA-S refere-se à um Requisito Arquitetural do contexto de Serviço

### 6.4 Passo 3 – Síntese arquitetural

A partir da execução do passo 3 do ProSA-RA é realizado o projeto arquitetural da Cambuci-LPN, tomando como base os requisitos arquiteturais elicitados no passo anterior, bem como o projeto arquitetural da Cambuci visto que a Cambuci-LPN é uma especialização da Cambuci. Assim, as mesmas visões arquiteturais utilizadas para representar a Cambuci são utilizadas para representar a Cambuci-LPN.

Além dessas visões, foi incluída a visão conceitual como forma de representar a Cambuci-LPN. Essa visão é importante porque descreve os conceitos de LPN que devem ser persistidos no repositório. A visão geral, a visão conceitual, a visão de módulo, a visão de tempo de execução e a visão de implantação da Cambuci-LPN são apresentadas nas subseções a seguir.

#### 6.4.1 Visão geral

A visão geral da Cambuci-LPN foi desenvolvida tomando como base a visão geral da Cambuci proposta na Seção 5.4.1 e levando em consideração os requisitos arquiteturais identificados nas seções anteriores deste capítulo. Os serviços do

domínio de repositórios de LPN foram alocados na camada de aplicação, enquanto que os serviços no contexto de serviço foram alocados nas outras camadas (processo de negócio, intermediação, apresentação ou de qualidade de serviço), conforme o seu propósito.

Enquanto que a visão geral da Cambuci contém os serviços de repositórios primários relacionados aos conceitos de repositórios de ativos de software (especificação de ativo e gestão de ativo), a visão geral da Cambuci-LPN contém os serviços primários de repositórios relacionados aos conceitos do domínio de repositórios de LPN (especificação da LPN, gestão da LPN, instanciação da LPN e rastreabilidade entre LPN e suas instâncias).

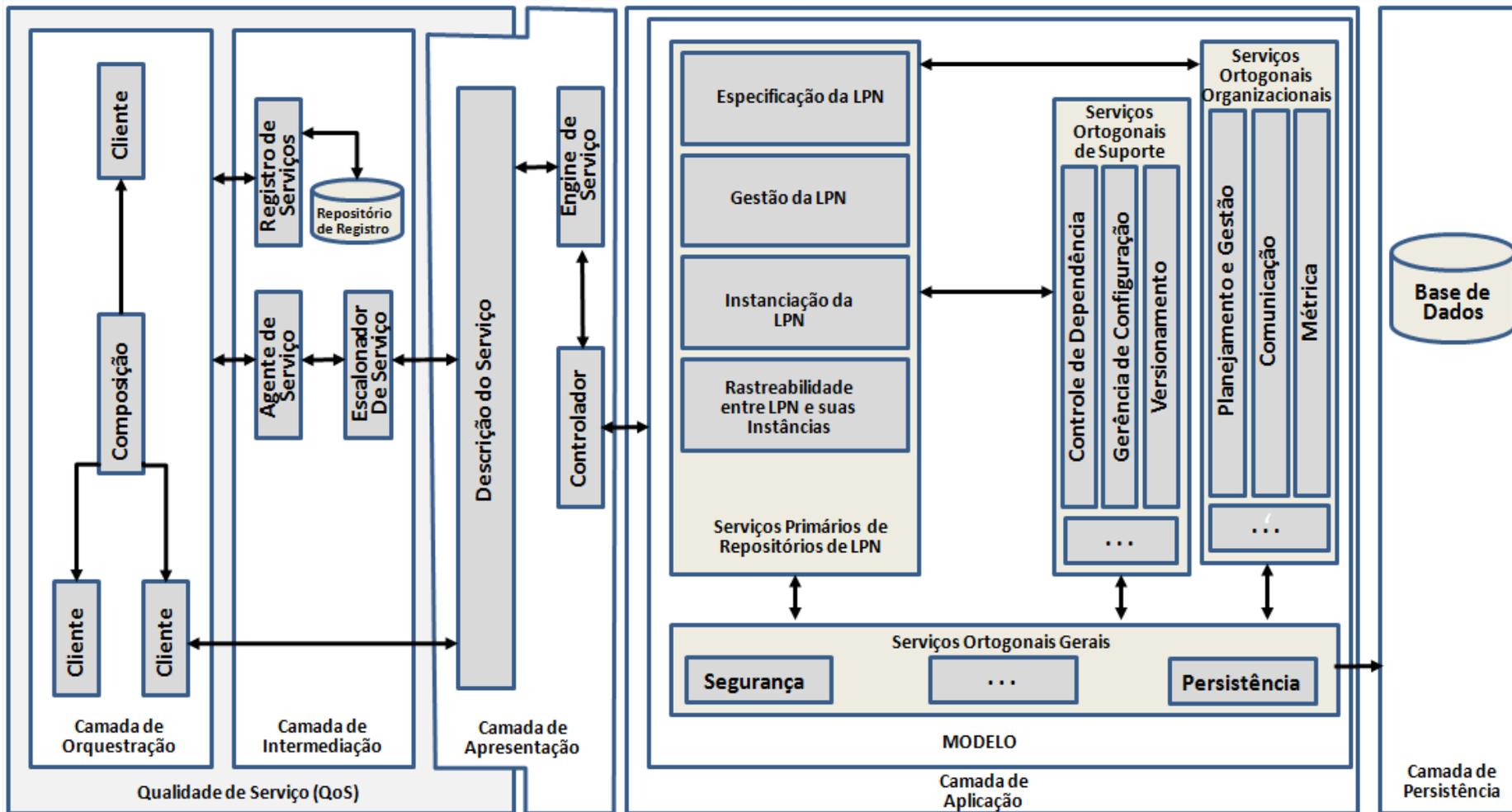
Na Figura 6.6 é apresentada a visão geral da Cambuci-LPN. Essa arquitetura de referência é dividida em seis camadas, descritas a seguir:

- **Camada de Aplicação:** contempla os serviços oferecidos relativos ao domínio de repositórios de LPN. Esses serviços são divididos nos seguintes grupos, conforme descrito na visão geral do Cambuci, e adaptados para o domínio de repositório de LPN: (i) Serviços Primários de Repositórios de LPN, que se referem aos serviços de Especificação da LPN, aos serviços de Gestão da LPN, aos serviços de Instanciação da LPN e aos serviços de Rastreabilidade entre LPN e suas Instâncias; (ii) Serviços Ortogonais de Suporte, que se referem aos serviços que apoiam as atividades de suporte conforme a ISO/IEC 12207 (2009), por exemplo, Controle de Dependência, Gerenciamento de Configuração e Versionamento; (iii) Serviços Ortogonais Organizacionais, que se referem aos serviços que apoiam as atividades organizacionais conforme a ISO/IEC 12207 (2009), como é o caso do Planejamento e Gestão, Comunicação e Métrica; (iv) Serviços Ortogonais Gerais, que se referem aos serviços de propósito geral, por exemplo, Segurança e Persistência. Os serviços do grupo (ii), (iii) e (iv) são considerados ortogonais porque podem ser utilizados por diversos outros serviços durante as atividades correspondentes aos Serviços Primários de Repositórios de LPN.

- **Camada de persistência:** é responsável por armazenar as LPNs e suas instâncias. O serviço de persistência atua diretamente nessa camada. As LPNs e suas instâncias devem ser persistidas em uma base de dados. Pode-se utilizar um sistema de gerenciamento de banco de dados para realizar esta tarefa, mas, também se pode utilizar um sistema de arquivos e diretórios.
- **Camada de apresentação:** como a arquitetura do repositório é orientada a serviços, essa camada possui os seguintes elementos: (i) Descrição do Serviço, que define como um serviço pode ser chamado e o que ele retorna; (ii) *Engine* de Serviço, que processa as requisições de serviços; (iii) Controlador, que é o responsável pelo tratamento das requisições encaminhadas pela *Engine* de Serviço.
- **Camada de intermediação:** camada que permite que os serviços de repositório sejam publicados, descobertos, associados e disponibilizados. Possui os seguintes elementos: (i) Registro de Serviços, que permite a publicação de um serviço e a pesquisa de um serviço disponível; (ii) Agente de Serviço, que faz o papel de mediador entre o cliente do serviço e o provedor do serviço; (iii) Escalonador de Serviço, que processa as requisições de serviço que possuem dependências.
- **Camada de orquestração:** permite a criação de novos serviços realizando uma composição dos serviços já existentes.
- **Camada de qualidade de serviço:** camada que verifica o cumprimento dos requisitos de qualidade presentes nas demais camadas de serviços. Ela pode atuar nas camadas de Orquestração, Intermediação, e na camada de Apresentação.

Para complementar a descrição da visão geral da Cambuci-LPN apresentada na Figura 6.6, o Quadro 6.6, Quadro 6.7, Quadro 6.8, Quadro 6.9, Quadro 6.10 e Quadro 6.11 listam os requisitos arquiteturais de cada camada de serviço.

Figura 6.6 - Visão geral da arquitetura Cambuci-LPN



**Quadro 6.6 - Requisitos arquiteturais da camada de aplicação**

<b>Camada: Aplicação</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviços</b>
RA-RLPN[1]	A arquitetura de referência deve possibilitar que repositórios de LPN incluam uma nova LPN, que pode ser composto por vários artefatos;	Serviço Primário
RA-RLPN [2]	A arquitetura de referência deve possibilitar que repositórios de LPN forneçam mecanismo para aceitação e certificação da LPN;	Serviço Primário
RA-RLPN [3]	A arquitetura de referência deve possibilitar que repositórios de LPN desativem LPNs que não serão mais utilizados;	Serviço Primário
RA-RLPN [4]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam a classificação de uma LPN e também informar o contexto de sua utilização;	Serviço Primário
RA-RLPN [5]	A arquitetura de referência deve possibilitar que repositórios de LPN registrem a dependência entre LPN;	Serviço Primário
RA-RLPN [6]	A arquitetura de referência deve possibilitar que repositórios de LPN notifiquem os interessados sobre mudanças que aconteçam na LPN;	Serviços Ortogonais Organizacionais
RA-RLPN [7]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam realizar buscas e recuperação das LPNs;	Serviço Primário
RA-RLPN [8]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam a navegação entre LPNs;	Serviço Primário
RA-RLPN [9]	A arquitetura de referência deve possibilitar que repositórios de LPN aceite múltiplas fontes de origem de LPNs, com o objetivo de facilitar a integração entre equipes e entre repositórios diferentes;	Serviço Primário
RA-RLPN [10]	A arquitetura de referência deve possibilitar que repositórios de LPN criem e armazenem múltiplas versões de uma mesma LPN;	Serviços Ortogonais de Suporte
RA-RLPN [11]	A arquitetura de referência deve possibilitar que repositórios de LPN gerencie a configuração, como por exemplo, a definição dos itens da LPN que são configuráveis, o controle de mudanças dos itens da LPN que são configuráveis;	Serviços Ortogonais de Suporte
RA-RLPN [12]	A arquitetura de referência deve possibilitar que repositórios de LPN permita o registro de impressões dos usuários a respeito da versão da LPN que eles utilizaram;	Serviços Ortogonais Organizacionais
RA-RLPN [13]	A arquitetura de referência deve possibilitar que repositórios de LPN registrem métricas coletadas sobre a utilização da LPN;	Serviços Ortogonais Organizacionais
RA-RLPN [14]	A arquitetura de referência deve possibilitar que repositórios de LPN ofereçam informações relativas ao reúso, iniciativas de reúso, LPNs mais usadas, etc;	Serviços Ortogonais Organizacionais
RA-RLPN [15]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam o acesso de acordo com o papel que o usuário assume;	Serviços Ortogonais Gerais
RA-RLPN [16]	A arquitetura de referência deve possibilitar que repositórios de LPN garantam a integridade das LPNs, ou seja, que eles não sofram alterações não autorizadas;	Serviços Ortogonais Gerais
RA-RLPN [17]	A arquitetura de referência deve possibilitar que repositórios de LPN realizem o gerenciamento de transação, garantindo a atomicidade, consistência, isolamento e durabilidade;	Serviços Ortogonais Gerais
RA-RLPN[18]	A arquitetura de referência deve possibilitar que repositórios de LPN incluam uma nova instância da LPN	Serviços Primários
RA-RLPN[19]	A arquitetura de referência deve possibilitar que repositórios de LPN propaguem a mudança na LPN para suas instâncias	Serviços primários
RA-RLPN[20]	A arquitetura de referência deve possibilitar que repositórios de LPN permitam que instâncias da LPN mantenham sincronizadas com diferentes versões da LPN	Serviços primários

**Quadro 6.6 (conclusão)**

RA-RLPN[21]	A arquitetura de referência deve possibilitar que repositórios de LPN mantenham a rastreabilidade bidirecional entre cada elemento da instância da LPN com o elemento correspondente na LPN	Serviços primários
RA-RLPN[22]	A arquitetura de referência deve possibilitar que repositórios de LPN identifiquem as diferenças entre as versões de uma LPN	Serviços primários
RA-RLPN[23]	A arquitetura de referência deve possibilitar que repositórios de LPN identifiquem as diferenças entre instâncias de diferentes versões de uma LPN	Serviços primários

Legenda: RA-RLPN refere-se à um Requisito Arquitetural de Repositórios de LPN

**Quadro 6.7 - Requisitos arquiteturais da camada de persistência**

<b>Camada: Persistência</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviços</b>
RA[1]	A arquitetura de referência deve armazenar os dados produzidos pelos serviços que compõem o repositório de LPN	Persistência

Legenda: RA refere-se à um Requisito Arquitetural

**Quadro 6.8 - Requisitos arquiteturais da camada de orquestração.**

<b>Camada: Processo de Negócio</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviços</b>
RA-S[1]	A arquitetura de referência deve possibilitar que repositórios de LPN desenvolvidos para persistir diferentes tipos de LPNs possam ser facilmente integrados	Serviço de Orquestração
RA-S[2]	A arquitetura de referência deve possibilitar que repositórios de LPN implementados em linguagens de programação distintas e sob diferentes plataformas possam ser facilmente integrados	Serviço de Orquestração
RA-S[4]	A arquitetura de referência deve prover mecanismos para que repositórios de LPN orientados a serviço possam ser compostos por processos de negócio ou utilizados por aplicações cliente	Serviço de Orquestração
RA-S[9]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de LPN escalável, capaz de evoluir de maneira incremental, por meio da composição de novas funcionalidades disponíveis na forma de serviços	Serviço de Orquestração
RA-S[10]	A arquitetura de referência deve possibilitar que serviços de repositório de LPN e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma	Serviço de Orquestração
RA-S[11]	A arquitetura de referência deve possibilitar que serviços do repositório de LPN possam interagir diretamente ou por meio do uso de barramentos de serviço	Serviço de Orquestração
RA-S[12]	A arquitetura de referência deve possibilitar que repositório de LPN disponíveis como serviços, desenvolvidos sem o uso dessa arquitetura, possam ser integrados aos repositórios de LPN desenvolvidos segundo a arquitetura de referência	Serviço de Orquestração
RA-S[13]	A arquitetura de referência deve possibilitar sua instanciação parcial, ou seja, repositórios de LPN desenvolvidos a partir dessa arquitetura podem ser construídos sem a necessidade de implementação de todos os módulos propostos	Serviço de Orquestração
RA-S[14]	A arquitetura de referência deve permitir que os repositórios de LPN possam evoluir ao longo de diferentes versões de maneira simples, ou seja, mudanças de versão dos repositórios de LPN não devem afetar a interação com os serviços que a utilizam	Serviço de Orquestração

Legenda: RA-S refere-se à um Requisito Arquitetural do contexto de Serviço

**Quadro 6.9 - Requisitos arquiteturais da camada de intermediação**

<b>Camada: Intermediação</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviços</b>
RA-S[3]	A arquitetura de referência deve prover mecanismos para que repositórios de LPN na forma de serviços possam ser publicados e posteriormente descobertos por aplicações cliente	Serviços de Intermediação
RA-S[6]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de LPN que disponibilizem descrições semânticas, permitindo assim sua classificação nos repositórios de serviço	Serviços de Intermediação
RA-S[10]	A arquitetura de referência deve possibilitar que serviços de repositório de LPN e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma	Serviços de Intermediação

Legenda: RA-S refere-se a um Requisito Arquitetural do contexto de Serviço

**Quadro 6.10 - Requisitos arquiteturais da camada de apresentação**

<b>Camada: Apresentação</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviço</b>
RA-S[5]	A arquitetura de referência deve viabilizar o desenvolvimento de repositórios de LPN que disponibilizem informações sobre suas características e direções normativas de uso, por meio de descrições padronizadas	Serviços de Apresentação
RA-S[6]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de LPN que disponibilizem descrições semânticas, permitindo assim sua classificação nos repositórios de serviço	Serviços de Apresentação
RA-S[7]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de LPN que tenham à disposição informações e documentos relacionados às suas características de qualidade	Serviços de Apresentação
RA-S[10]	A arquitetura de referência deve possibilitar que serviços de repositório de LPN e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma	Serviços de Apresentação

Legenda: RA-S refere-se a um Requisito Arquitetural do contexto de Serviço

**Quadro 6.11 - Requisitos arquiteturais da camada de qualidade de serviço**

<b>Camada: Qualidade de Serviço</b>		
<b>ID</b>	<b>Requisitos</b>	<b>Serviço</b>
RA-S[7]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de LPN que tenham à disposição informações e documentos relacionados às suas características de qualidade	Serviço de Qualidade de Serviço
RA-S[8]	A arquitetura de referência deve prover mecanismos para a captura, monitoramento, registro e sinalização do não cumprimento de requisitos de qualidade estabelecidos entre serviços provedores e serviços clientes	Serviço de Qualidade de Serviço

Legenda: RA-S refere-se a um Requisito Arquitetural do contexto de Serviço

#### 6.4.2 Visão conceitual

Como muitos conceitos são necessários para o entendimento de LPN, optou-se por adicionar a visão conceitual para a arquitetura de referência Cambuci-LPN. Para representar essa visão, é utilizado o modelo conceitual por meio da notação do diagrama de classes da UML.

Na Figura 6.7 é apresentada a visão conceitual da Cambuci-LPN. Os elementos contidos neste modelo conceitual são os elementos que irão compor uma LPN e que podem ser armazenados como metadados no repositório de LPN.

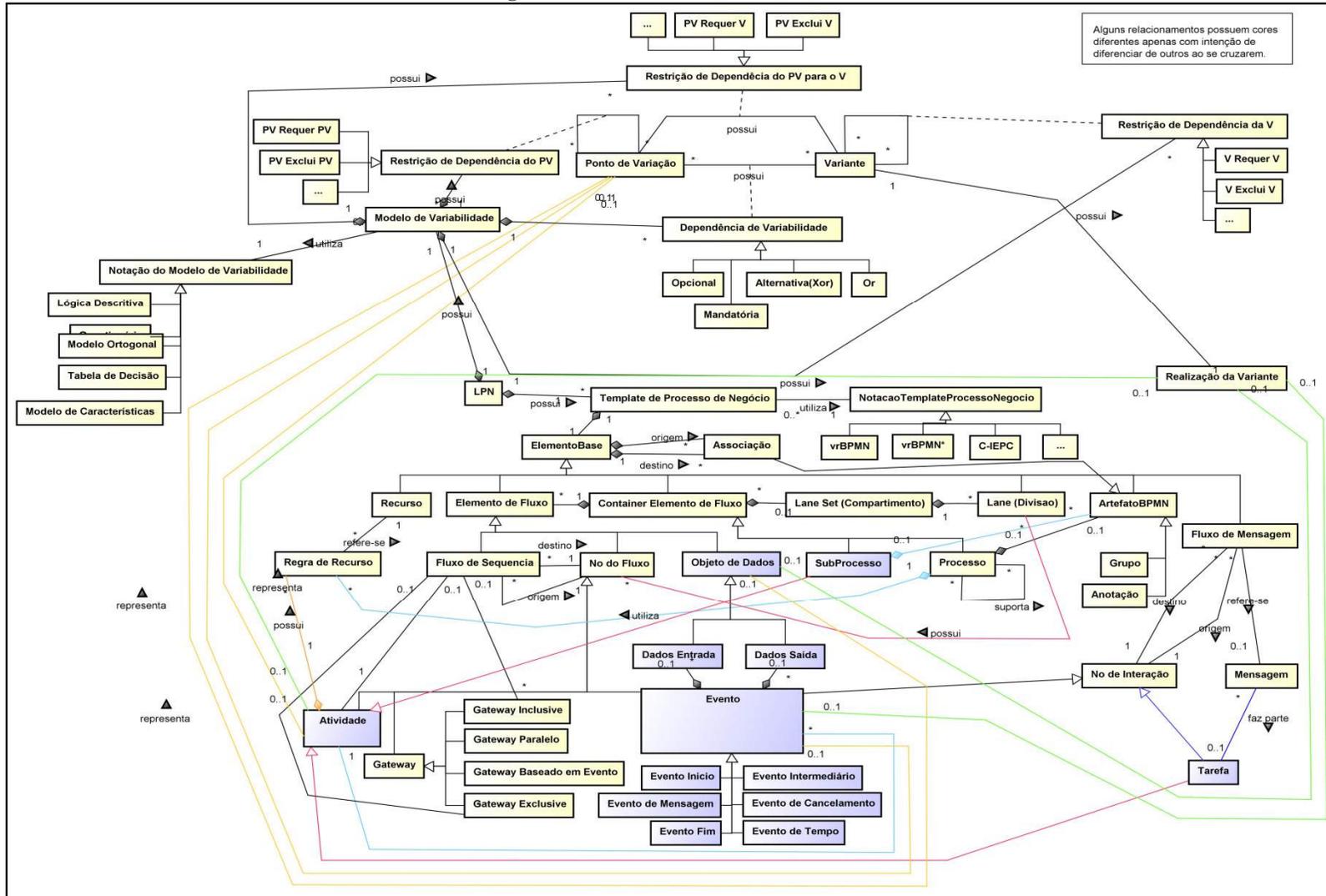
A construção desse modelo conceitual foi baseada nos modelos conceituais do Modelo de Variabilidade (Figura 6.4) e do *Template* de Processo de Negócio (Figura 6.5). A ligação entre os conceitos de ambos modelos ocorre por meio do Ponto de Variação, Variante e Realização da Variante, pois o Mapeamento de Rastreabilidade (Figura 6.3) está representado implicitamente no Modelo de Variabilidade e no *Template* de Processos de Negócio, de acordo com o vrBPMN, e não em um artefato específico.

### **6.4.3 Visão de módulo**

A visão de módulo da Cambuci-LPN é descrita em UML e é construída tomando como base a visão de módulo da arquitetura Cambuci (Capítulo 6), os conceitos e os requisitos arquiteturais definidos na Seção 6.3 e as camadas da visão geral definida na Seção 6.4.1.

A partir das camadas da visão geral da Cambuci-LPN, foram estabelecidos os pacotes e os serviços da visão de módulo da arquitetura de referência Cambuci-LPN, conforme ilustrados no Quadro 6.12. Os pacotes correspondem aos agrupamentos dos serviços conforme as suas funcionalidades implementadas. Conforme a quantidade de serviços em cada pacote, é realizado o agrupamento em sub-pacotes.

Figura 6.7 - Modelo conceitual da LPN



**Quadro 6.12 - Módulos da visão de módulos para o Cambuci-LPN**

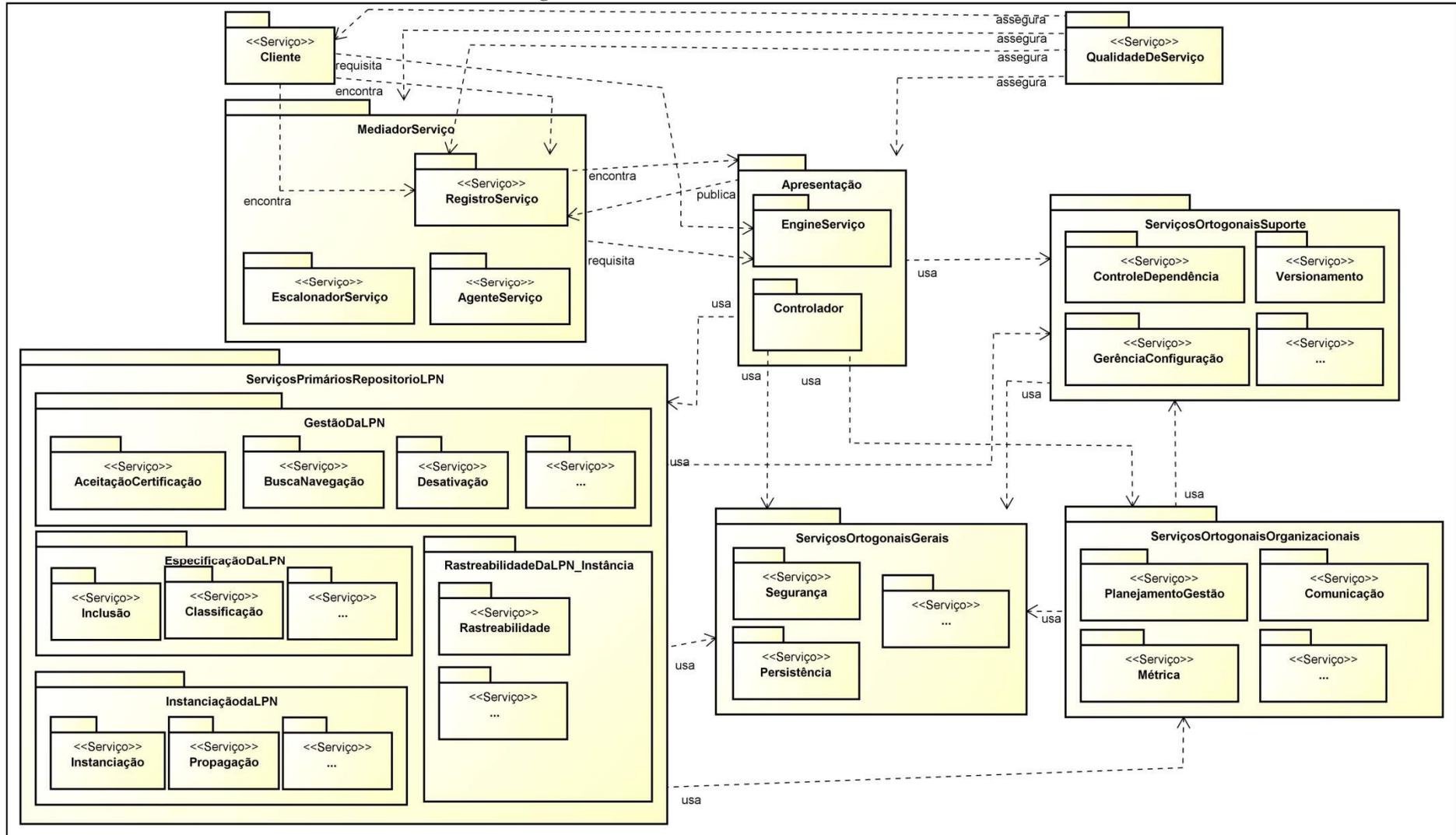
Camadas da visão geral da Cambuci-LPN	Pacotes	Sub-pacotes
Aplicação	ServiçosPrimáriosRepositórioLPN	EspecificaçãoDaLPN, GestãoDaLPN, InstanciaçãoDaLPN, RastreabilidadeDaLPN_Instância
	ServiçosOrtogonalisSuporte	ControleDependência, Versionamento, GerenciaConfiguracao
	ServiçosOrtogonalisOrganizacionais	PlanejamentoGestão, Comunicação, Métrica
	ServiçosOrtogonalisGerais	Segurança, Persistência
Persistência	ServiçosOrtogonalisGerais	Persistência
Orquestração	MediadorServiço	RegistroServiço, EscalonadorServiço, AgenteServiço
Intermediação	MediadorServiço	RegistroServiço, EscalonadorServiço, AgenteServiço
Apresentação	Apresentação	EngineServiço, Controlador
Qualidade de Serviço	QualidadeDeServiço	-

De acordo com a Figura 6.8, verifica-se que os serviços relacionados aos repositórios de LPN, descritos na camada de aplicação da visão geral da Cambuci-LPN, são representados na visão de módulo por meio dos pacotes: `ServiçosPrimáriosRepositórioLPN`, `ServiçosOrtogonalisSuporte`, `ServiçosOrtogonalisOrganizacionais` e `ServiçosOrtogonalisGerais`. O pacote de `ServiçosPrimáriosRepositórioLPN` é subdividido nos seguintes pacotes: pacote de serviço de `EspecificaçãoDaLPN`, pacote de `Serviços de GestãoDaLPN`, pacote de `Serviços de InstanciaçãoDaLPN` e pacote de `Serviços de RastreabilidadeDaLPN_Instância`.

Os pacotes de `Serviços Ortogonalis` podem ser utilizados para apoiar os `ServiçosPrimáriosRepositórioLPN`, enquanto que os serviços do pacote `MediadorServiço` permite a interação com os serviços relacionados aos repositórios de LPN. Já o pacote de `Apresentação` é responsável pelas solicitações de serviços (pacote `EngineServiço`), e por tratar essas requisições de serviço (pacote `Controlador`).

O pacote `Cliente` representa como os clientes podem fazer uso dos serviços disponíveis nos repositórios de LPN, e o pacote `QualidadeDeServiço` fornece serviços que permitem a inspeção da qualidade dos serviços oferecidos.

Figura 6.8 - Visão de módulo da Cambuci-LPN



Para complementar a descrição da visão de módulos da Cambuci-LPN, no Quadro 6.13 são descritos os serviços de cada pacote e sub-pacote.

**Quadro 6.13 - Distribuição dos serviços em cada pacote e sub-pacote da Cambuci-LPN**

Pacotes	Sub-pacotes	Serviços
Cliente	-	Representar o cliente
QualidadeDeServiço	-	Assegurar a qualidade dos serviços
MediadorServiço	RegistroServiço	Publicar Serviço, pesquisar serviços publicados
MediadorServiço	EscalonadorServiço	Processar requisição de serviço
MediadorServiço	AgenteServiço	Mediar os serviços cliente e os provedores desses serviços
Apresentação	EngineServiço	Processar requisição de serviço
Apresentação	Controlador	Tratar requisição de serviço
ServiçosOrtognaisSuporte	ControleDependência	Realizar o controle de dependência entre LPNs
ServiçosOrtognaisSuporte	Versionamento	Realizar o versionamento da LPN
ServiçosOrtognaisSuporte	GerênciaConfiguração	Realizar a tarefa de gerência de configuração da LPN
ServiçosOrtognaisGerais	Segurança	Controlar o acesso e a autenticação
ServiçosOrtognaisGerais	Persistência	Realizar a persistência
ServiçosOrtognaisOrganizacionais	Planejamento Gestão	Realizar o planejamento e gestão da LPN
ServiçosOrtognaisOrganizacionais	Comunicação	Registrar o <i>feedback</i> dos usuários, notificar aos usuários sobre mudanças, divulgar informações sobre o reuso
ServiçosOrtognaisOrganizacionais	Métrica	Registrar métricas de utilização da LPN
ServiçosPrimáriosRepositórioLPN/ EspecificaçãoDaLPN	Inclusão	Incluir uma LPN
ServiçosPrimáriosRepositórioLPN/ EspecificaçãoDaLPN	Classificação	Classificar uma LPN
ServiçosPrimáriosRepositórioLPN/ GestãoDaLPN	AceitaçãoCertificação	Aceitar e certificar a LPN, tornando-o disponível para reuso
ServiçosPrimáriosRepositórioLPN/ EspecificaçãoDaLPN	BuscaNavegação	Buscar uma LPN, navegar entre LPN
ServiçosPrimáriosRepositórioLPN/ EspecificaçãoDaLPN	Desativação	Indisponibilizar uma LPN
ServiçosPrimáriosRepositórioLPN/ InstanciaçãoDaLPN	Instanciação	Permitir a gestão da instância da LPN
ServiçosPrimáriosRepositórioLPN/ InstanciaçãoDaLPN	Propagação	Permitir a propagação de mudanças na LPN
ServiçosPrimáriosRepositórioLPN/ RastreabilidadeDaLPN_Instância	Rastreabilidade	Permitir manter a rastreabilidade entre as versões da LPN e suas instâncias

#### 6.4.4 Visão em tempo de execução

A visão em tempo de execução da Cambuci-LPN, ilustrada na Figura 6.9, representa a estrutura de um repositório de LPN no momento em que ele está sendo executado. Essa visão é abstraída tomando como base a visão em tempo de execução da arquitetura Cambuci (Capítulo 5).

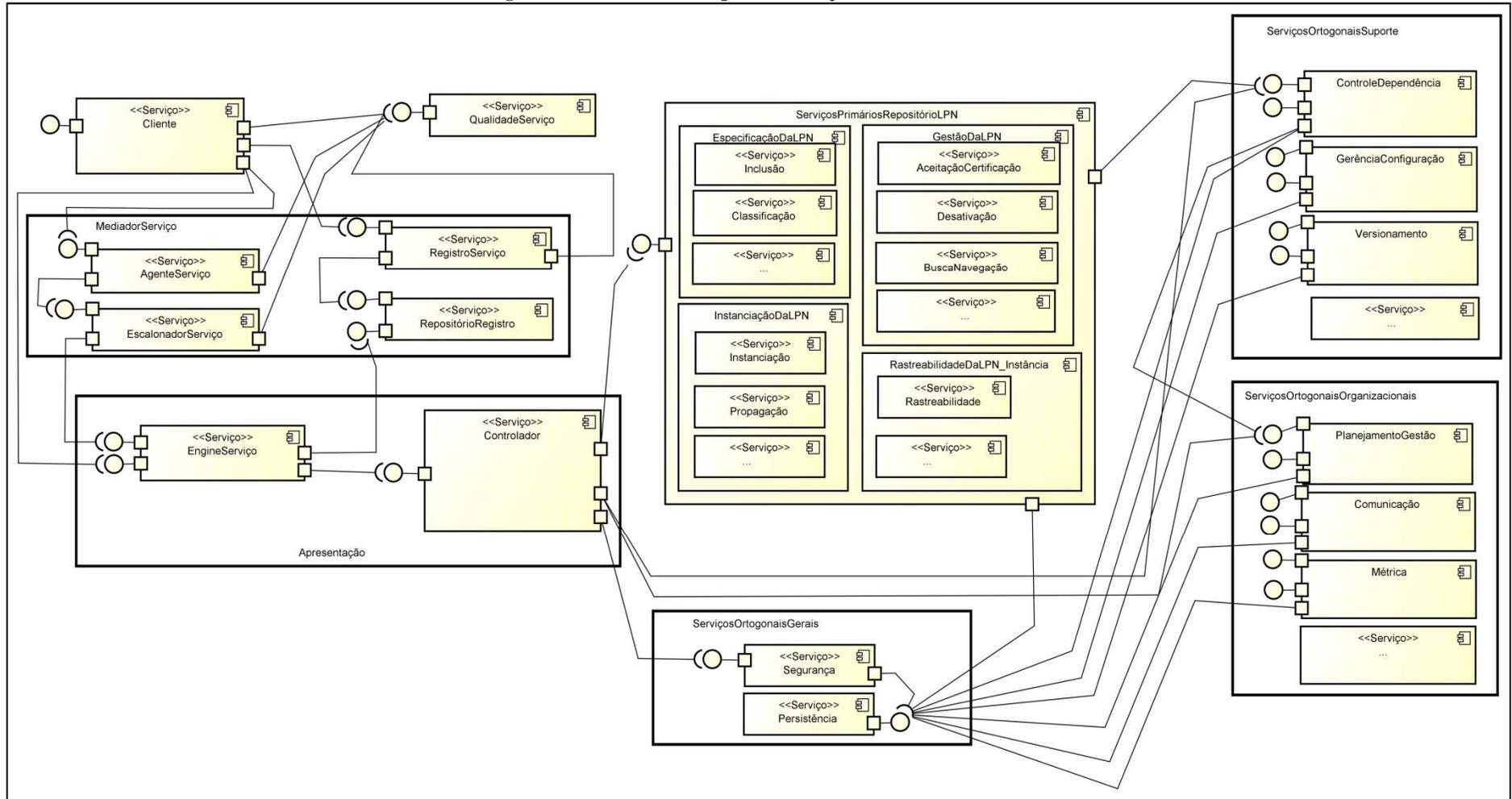
Os pacotes identificados na visão de módulo correspondem aos componentes presentes na visão de tempo de execução da Cambuci-LPN. Na Figura 6.9 são apresentadas as interações entre os componentes desta visão, a qual foi elaborada após análise das possíveis interações que podem ocorrer entre seus componentes. Nesta figura, o símbolo  indica que o componente está provendo o serviço, enquanto que o símbolo  indica que o componente está requisitando o serviço.

Os serviços primários presentes nos componentes `EspecificaçãoDaLPN`, `GestãoDaLPN`, `InstanciaçãoDaLPN` e `RastreabilidadeDaLPN_Instância` são os responsáveis por fornecer as funcionalidades principais dos repositórios de LPN.

O serviço `Cliente` pode buscar informações sobre um serviço disponível por meio de uma pesquisa no `RegistroServiço`, que retorna o endereço do serviço solicitado. Através desse endereço, a comunicação pode ser realizada diretamente com a `EngineServiço` ou indiretamente através do `AgenteServiço`. A qualidade dos serviços é garantida pelo componente `QualidadeServiço`, que monitora a comunicação entre os serviços clientes e todos os demais componentes que proveem serviços, que são os serviços provedores. A `EngineServiço` encaminha as requisições ao `Controlador` que por sua vez direciona para algum serviço, que pertence a algum componente e que se refere a alguma funcionalidade de repositórios de LPN (`ServiçosPrimárioRepositórioLPN`, `ServiçosOrtogonaisSuporte`, `ServiçosOrtogonaisOrganizacionais` e `ServiçosOrtogonaisGerais`).

Nesta visão de tempo de execução, verifica-se que o serviço `ControleDependência`, localizado no componente `ServiçosOrtogonaisSuporte` pode receber requisição de serviço de qualquer serviço localizado no componente de `ServiçosPrimáriosRepositórioLPN` e do `Controlador` e o serviço de `ControleDependência` pode requisitar serviços para algum componente de `ServiçosOrtogonaisOrganizacionais` e para algum componente de `ServiçosOrtogonaisGerais`.

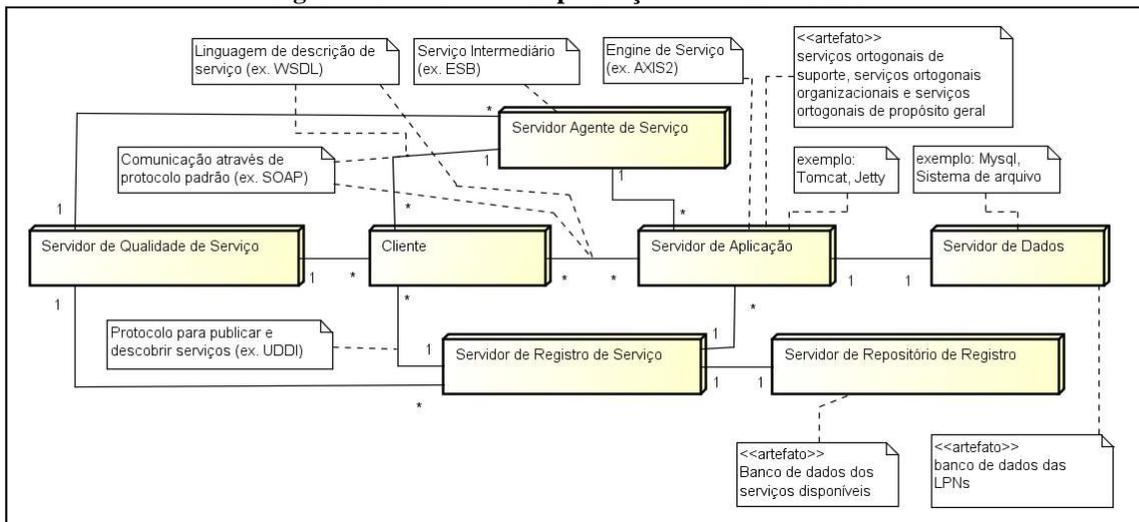
Figura 6.9 - Visão em tempo de execução da Cambuci-LPN



### 6.4.5 Visão de implantação

A visão de implantação (Figura 6.10) da arquitetura de referência Cambuci-LPN é a mesma apresentada pela arquitetura de referência Cambuci (ver Seção 5.4.4 do Capítulo 5) da qual a Cambuci-LPN é especializada. Por sua vez, a visão de implantação da Cambuci é igual a visão de implantação da arquitetura de referência RefTEST-SOA (OLIVEIRA, 2011) apresentado na Seção 2.5 do Capítulo 2. Esta semelhança acontece devido à abordagem orientada a serviços presentes nas três arquiteturas de referência.

**Figura 6.10 - Visão de implantação da Cambuci-LPN**

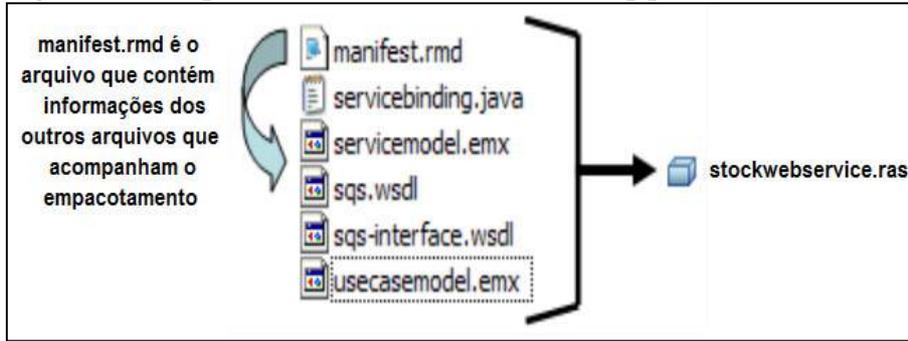


A visão de implantação da Cambuci-LPN especifica que o Servidor de Dados não precisa ser necessariamente um servidor de banco de dados, podendo opcionalmente utilizar a estrutura do sistema de arquivos para armazenar as LPNs.

Essa forma de armazenamento é sugerida pela RAS (OMG, 2005), que é uma especificação de ativos reutilizáveis. Essa especificação foi uma das fontes de informação utilizada em Silva (2013) para o estabelecimento da ONTO-ResAsset pois ele provê recomendações sobre a estrutura, definição, conteúdo e descrição de ativos reutilizáveis.

Nessa especificação é descrito sobre o empacotamento de ativos, sendo que um ativo pode ser empacotado em um único arquivo compactado, com a extensão .ras, contendo vários artefatos e um arquivo manifesto, que contém informações sobre os outros arquivos (artefatos) que fazem parte desse ativo. O conteúdo de um arquivo compactado, que representa um ativo empacotado, é mostrado na Figura 6.11 de acordo com a RAS.

**Figura 6.11 - Empacotamento de ativo com formato zip pela RAS (OMG, 2005)**



#### **6.5 Passo 4: Avaliação da arquitetura de referência**

A avaliação da arquitetura de referência Cambuci-LPN é realizada por meio de um *checklist*, composto por quatro passos, proposto por Santos (2013). Cada passo do *checklist* contém perguntas que devem ser respondidas pelos avaliadores.

De acordo com Santos (2013), podem participar da avaliação de uma arquitetura de referência todos os *stakeholders*, como: arquiteto de software, especialista do domínio, gerente, desenvolvedor, especialista em garantia de qualidade do software, especialista de teste e analista de sistemas. Como o *checklist* de Santos (2013) foi proposto para o domínio de sistemas embarcados, ele foi adaptado neste trabalho para o contexto de repositórios de LPN, conforme ilustrado no Quadro 6.14.

O primeiro passo tem como objetivo verificar a completude das informações gerais, da construção e do conteúdo, e é dividido em dois conjuntos. O primeiro conjunto possui nove questões, envolvendo informações gerais, pontos de vistas, visões, modelos, *stakeholders* e os interesses. O segundo conjunto contém 16 questões sobre a construção da arquitetura de referência, como por exemplo, se a arquitetura de referência segue algum regulamento legal, ou se está em conformidade com a ISO/IEC/IEEE 42010 (2011).

O segundo passo tem como objetivo verificar a adequação da arquitetura de referência para liberação de uso e é composto por 52 questões. O terceiro passo possui três questões com o objetivo de verificar a conclusão e a análise geral da arquitetura de referência.

As questões dos três primeiros passos do *checklist* proposto por Santos (2013) foram mantidas. Já no quarto passo, como a arquitetura de referência envolve domínios diferentes, 8 questões foram construídas para o domínio de repositórios de LPN envolvendo conceitos de SOA, repositório de ativo de software e repositório de LPN.

**Quadro 6.14 - Assuntos abordados por cada passo do *checklist* (Adaptado de Santos, 2013)**

Passo	Conjunto	Questões	Assunto abordado
1	1	9	Informações gerais, pontos de vista, visões, modelos, <i>stakeholders</i> , e interesses
	2	16	Regulamentos legais, conformidade com a ISO/IEC 42010, questões de projeto e desenvolvimento, questões específicas do domínio, conformidade com outros artefatos, atributos de qualidade
2	1	52	Questões específicas para os <i>stakeholders</i>
3	1	3	Conclusão da análise geral
4	1	8	SOA, repositório e repositório LPN

No Quadro 6.15 são apresentadas algumas questões de exemplo e para quem as questões são direcionadas.

**Quadro 6.15 - Exemplos de questão do *checklist* (Adaptado de Santos, 2013)**

Passo	Conjunto	Respondido por	Questão de exemplo
1	1	Arquiteto	A arquitetura de referência apresenta: informações gerais, data de atualização, versão, proprietário, histórico de mudanças, descrição resumida, escopo, terminologia de domínio, decisões abertas e material de suporte?
		Arquiteto	O nível de detalhes favorece o entendimento da arquitetura de referência?
	2	Arquiteto	Cada visão representa corretamente o seu ponto de vista?
2	1	Todos os <i>stakeholders</i>	Os pontos de vistas selecionados englobam os interesses de todos os <i>stakeholders</i> ?
		Todos os <i>stakeholders</i>	Para cada ponto de vista, seus modelos são claros e bem definidos? Os modelos fornecem bastante informações para determinar se os interesses englobados pelos pontos de vistas são satisfeitos?
		Arquiteto	Você pode mostrar como foi produzida a lista de <i>stakeholders</i> e seus interesses?
		Especialista de domínio	Os objetivos do domínio do sistema estão claramente articulados e priorizados?
		Especialista de domínio	Existe rastreabilidade entre os objetivos do domínio e seus requisitos?
		Gerente	A descrição da arquitetura de referência permite uma estimativa de esforço para implementá-la?
		Gerente	A descrição da arquitetura de referência mostra quais partes podem ser implementadas usando componentes de terceiros e componentes <i>opensource</i> ?
		Desenvolvedor	Você consegue identificar as dependências permitidas e proibidas entre as partes da arquitetura de referência?
		Integrador	Você entende os pontos de adaptação da arquitetura de referência?
		Especialista de garantia da qualidade	Há um processo para assegurar conformidade com os requisitos de atributos de qualidade?
Especialista de teste	Para cada partição da arquitetura de referência, você pode determinar o que é necessário testar?		

**Quadro 6.15 (conclusão)**

Passo	Conjunto	Respondido por	Questão de exemplo
2	1	Analista de Sistemas	Se a arquitetura de referência é parte de um ciclo de vida ou processo que inclui uma tomada de decisão, a arquitetura de referência contém as informações apropriadas para dar suporte ao processo de tomada de decisão?
3	1	Todos os <i>stakeholders</i>	O documento atual é completo no sentido de que todas as informações são documentadas? Se não, há espaços reservados para o que ainda tem que ser documentado, juntamente com descrições do que ainda precisa ser trabalhado?
4	1	Especialistas de domínio	1) Você consegue listar os benefícios que a abordagem orientada a serviços fornece à arquitetura de referência proposta? 2) Os principais conceitos do domínio de repositórios foram identificados pela arquitetura de referência? 3) Os principais conceitos do domínio de repositórios de LPN foram identificados pela arquitetura de referência? 4) Através da arquitetura de referência, você consegue vislumbrar a sua instanciação? 5) Através da arquitetura de referência, você consegue vislumbrar a integração de uma instância dessa arquitetura de referência com outros sistemas/ferramentas? 6) Importantes funcionalidades de repositórios estão incorporadas na arquitetura de referência? 7) Importantes funcionalidades para repositórios de LPN estão incorporadas na arquitetura de referência? 8) A arquitetura de referência permite vislumbrar como seria a implantação de suas instâncias?

Neste trabalho, o *checklist* foi respondido por uma única pessoa, com o perfil de especialista do domínio e de gerente de software. O tempo gasto para a leitura da documentação da arquitetura de referência foi de 1 hora e 50 minutos e o tempo para responder o *checklist* foi de 2 horas.

Do total das questões do *checklist*, 38% não foram respondidas em virtude do perfil do entrevistado não fazer parte do perfil o *stakeholder* para o qual as questões eram direcionadas.

As porcentagens a seguir, referem-se àquelas relacionadas ao total de questões respondidas (62%).

Como resultado do primeiro passo, pôde-se observar que aproximadamente 50% das questões foram atendidas na totalidade, 24% parcialmente e 26% não foram atendidas. As justificativas que apresentam a razão das questões não estarem sendo

completamente atendidas são mostradas no Quadro 6.16, acompanhadas das respectivas soluções indicadas pelo autor desta dissertação de mestrado.

Nesse passo conclui-se que a documentação da Cambuci-LPN está satisfatória, contendo as informações mais importantes e atende os principais requisitos elencados pelo *checklist*. A arquitetura de referência apresenta um grau de abstração em que os serviços identificados em cada módulo correspondem às funcionalidades que o módulo deve disponibilizar. Na construção da arquitetura do sistema, essas funcionalidades devem ser atendidas por vários serviços presentes no módulo. Por exemplo, o serviço Assegurar a qualidade dos serviços, presente no pacote QualidadeDeServiço, deve ser disponibilizado por meio de mecanismos que colem métricas relacionadas a aspectos de qualidade esperados pelos serviços. Essas métricas podem ser relacionadas, por exemplo, a desempenho, segurança e confiabilidade.

**Quadro 6.16 - Justificativas e soluções para as respostas parcialmente atendidas e não atendidas das questões (passo 1)**

<b>Questões</b>	<b>Justificativa</b>	<b>Solução</b>
Questões parcialmente atendidas	- o conceito serviço foi utilizado de uma maneira muito abstrata. O serviço “Assegurar qualidade dos serviços” declarado no módulo QualidadeDeServiço corresponde a uma funcionalidade que deve ser atendida por um ou mais serviços.	Indicar na documentação da Cambuci-LPN que todos os serviços de cada pacote da visão de módulo devem ser detalhados quando a mesma estiver sendo instanciada.
	- atributos de qualidade não estão explicitamente definidos, mas fornece um meio para medir e acompanhar os atributos de qualidade estabelecidos.	Incluir novos requisitos arquiteturais para que contemplem elementos relacionados ao domínio de atuação da arquitetura de referência, como: legislação, regulamentos, padrões, atributos de qualidade, entre outros.
Questões não atendidas	- Algumas informações referentes à gestão do desenvolvimento da arquitetura não estão incluídas, como a data de atualização, versão, histórico de mudança, entre outros.	Adicionar na documentação arquitetural da Cambuci-LPN informações relacionadas ao processo de evolução da mesma, para que se possa compreender as decisões arquiteturais que foram tomadas e quando foram tomadas.
	- A lista de <i>stakeholders</i> não está incluída na arquitetura de referência. - Os interesses dos <i>stakeholders</i> não estão sendo apresentados pela arquitetura de referência.	Incluir na documentação arquitetural da Cambuci-LPN um diagrama de casos de uso para representar os <i>stakeholders</i> , relacionando-os com os seus interesses.

Na construção da arquitetura de referência, os *stakeholders* e os seus interesses não ficaram bem estabelecidos, bem como o acompanhamento da gestão da

construção da arquitetura de referência, no sentido de documentar melhor as decisões arquiteturais que foram sendo realizadas.

Como resultado do segundo passo, pôde-se observar que aproximadamente 61% das questões foram atendidas na totalidade, 5% parcialmente e 34% não foram atendidas. As justificativas que apresentam a razão das questões não estarem sendo completamente atendidas no segundo passo do *checklist* são mostradas no Quadro 6.17, acompanhadas das respectivas soluções indicadas pelo autor desta dissertação.

**Quadro 6.17 - Justificativas e soluções para as respostas parcialmente atendidas e não atendidas das questões (passo 2)**

Questões	Justificativa	Solução
Questões parcialmente atendidas	<ul style="list-style-type: none"> <li>- os benefícios da arquitetura de referência não estão explicitamente definidos.</li> <li>- objetivos do domínio do sistema não estão claramente definidos.</li> </ul>	Incluir na documentação arquitetural da Cambuci-LPN os objetivos e as necessidades por ela atendidos.
Questões não atendidas	<ul style="list-style-type: none"> <li>- as ameaças na instanciação da arquitetura de referência não estão definidas.</li> </ul>	Incluir na documentação arquitetural da Cambuci-LPN os riscos e as ameaças que possam existir durante a sua instanciação.

Observa-se, por meio das respostas obtidas das questões do segundo passo do *checklist*, que a arquitetura de referência Cambuci-LPN é satisfatória, mas alguns pontos necessitam de melhor detalhamento, como os benefícios que a arquitetura de referência proporciona, a explicitação dos objetivos do sistema e a rastreabilidade entre os objetivos do sistema com os requisitos arquiteturais.

Como resultado do terceiro passo, pôde-se observar que 100% das questões foram atendidas na totalidade, concluindo que a arquitetura possui um nível adequado para liberação.

Analogamente ao passo anterior, no quarto passo 100% das questões do *checklist* foram atendidas na totalidade, concluindo que a arquitetura de referência consegue expressar corretamente a abordagem orientada a serviços, o domínio de repositórios de ativos de software e o domínio de repositórios de LPN.

## 6.6 Considerações finais

Utilizando a arquitetura de referência Cambuci, estabelecida no Capítulo 5, e a abordagem GLPN, apresentada na Seção 4.3 do Capítulo 4, como fontes de informação, neste capítulo foi estabelecida a Cambuci-LPN, que é uma arquitetura de referência orientada a serviços para repositórios de LPN.

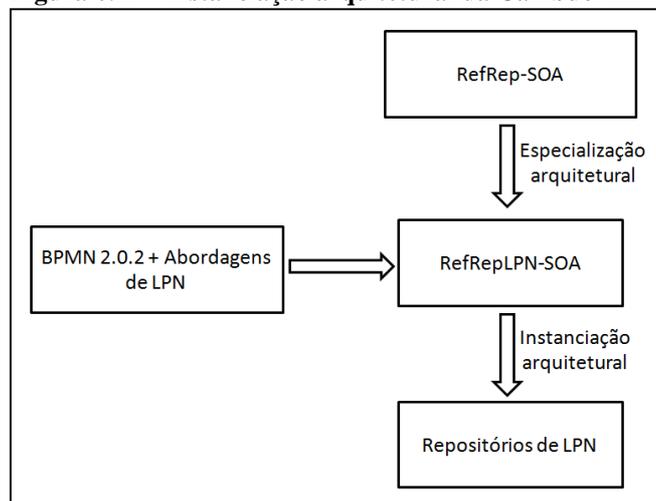
Para obter a Cambuci-LPN foram seguidos os passos definidos pelo ProSA-RA, em que, no primeiro passo, tem-se a investigação das fontes de informação Cambuci e GLPN.

No segundo passo foram obtidos os requisitos arquiteturais do domínio de repositórios de LPN e os requisitos arquiteturais no contexto de serviços para a Cambuci-LPN. Os requisitos referentes ao domínio de repositórios de ativos de software e ao contexto de serviços da Cambuci foram adaptados para o domínio da LPN. Outros requisitos arquiteturais foram obtidos por meio da abordagem GLPN.

No terceiro passo do ProSA-RA foi realizado o projeto arquitetural da Cambuci-LPN. Nesse passo foram estabelecidas uma visão geral e quatro visões arquiteturais para a Cambuci-LPN, que são: visão de módulo, visão conceitual, visão em tempo de execução e visão de implantação.

O quarto passo, que consiste na avaliação da arquitetura de referência, foi executado com o apoio do *checklist* proposto por Santos (2013), que permitiu concluir que a arquitetura de referência pode ser liberada para uso, permitindo a instanciação de repositórios de LPN, conforme ilustrado na Figura 6.12.

**Figura 6.12 - Instanciação arquitetural da Cambuci-LPN**



No próximo capítulo são apresentadas as conclusões do trabalho realizado.

## Capítulo 7 - Conclusão

### 7.1 Considerações iniciais

Este capítulo apresenta as conclusões observadas a partir do trabalho realizado. Na Seção 7.2 são apresentadas as principais contribuições obtidas. Na Seção 7.3 são discutidas as principais limitações do trabalho e na Seção 7.4 são apresentadas sugestões de trabalhos futuros para dar continuidade ao trabalho realizado.

### 7.2 Contribuições

A principal contribuição deste trabalho é a arquitetura de referência orientada a serviços para repositórios de LPN, denominada Cambuci-LPN (Capítulo 6), criada com o apoio de um processo de definição de arquiteturas de referência (ProSA-RA). O uso de um processo durante este trabalho viabilizou o estabelecimento de uma arquitetura de referência que terá aceitação satisfatória por parte de seus usuários, de acordo com a avaliação realizada por meio de um *checklist* específico para avaliação de arquiteturas de referência (SANTOS, 2013).

A Cambuci-LPN tem como objetivo apoiar a construção de repositórios de LPN, que podem ser utilizados tanto pela área de Gestão de Processos como pela área de Engenharia de Software, visto que a modelagem de processos de negócio é importante para ambas áreas, conforme discutido no Capítulo 3. Adicionalmente, como a Cambuci-LPN é baseada em serviços, permitirá que os repositórios resultantes de sua instanciação sejam integrados com outras ferramentas computacionais, agilizando as tarefas de analistas de processos e de engenheiros de software.

Adicionalmente, a Cambuci-LPN também contribuirá com a abordagem GLPN (Landre, 2012), visto que possibilitará a construção de um repositório de LPN, por meio de sua instanciação com o apoio do ProSA-RA.

Outra contribuição do trabalho é a arquitetura de referência orientada a serviços para repositórios de ativos de software, denominada Cambuci (Capítulo 5), definida também com o apoio do processo ProSA-RA. Apesar de haver diversas arquiteturas de referência de ativos de software na literatura, nenhuma detalha a arquitetura utilizando a abordagem orientada a serviços e nenhuma contempla todos os requisitos comumente encontrados em repositórios de ativos de software e em conformidade com a ISO/IEC 12207 (2009). Ressalta-se que essa arquitetura de referência contribui efetivamente com a área de reúso de software, pois pode ser

especializada para qualquer sub-domínio do domínio de repositórios de ativos de software, como repositórios de LPN (Capítulo 6), repositórios de requisitos, repositórios de modelos de análise, repositórios de teste de software, entre outros.

O trabalho também contribuiu para extrair da arquitetura de referência RefTEST-SOA (OLIVEIRA, 2011), os requisitos arquiteturais do contexto de serviço para arquiteturas de referência baseadas em serviços do domínio de ferramentas de engenharia de software (Seção 5.2.3 do Capítulo 5). Dessa forma, esses requisitos arquiteturais de serviços para ferramentas de engenharia de software podem ser utilizados durante a definição de arquiteturas de referência nesse domínio, como ocorreu neste trabalho durante a definição da arquitetura de referência Cambuci.

### **7.3 Limitações**

Após o desenvolvimento deste trabalho foram observadas algumas limitações, as quais estão elencadas a seguir:

- As arquiteturas de referência Cambuci e Cambuci-LPN não foram instanciadas, com isso não é possível garantir a qualidade das mesmas sob essa perspectiva;
- A validação da Cambuci-LPN foi conduzida por apenas uma pessoa, com o perfil de especialista de domínio e gerente de software, o que ocasionou em uma avaliação muito restrita a esse tipo de *stakeholder*;
- Com a avaliação conduzida, observou-se que alguns pontos não estão explícitos em nenhuma visão da arquitetura de referência Cambuci-LPN, como é o caso de: *stakeholders* e seus interesses, objetivos do sistema proposto pela arquitetura e a rastreabilidade dos objetivos do sistema com os requisitos arquiteturais; e
- A visão de implementação, importante para apoiar a instanciação de arquitetura de referência, não foi considerada durante a concepção das arquiteturas de referência Cambuci e Cambuci-LPN.

### **7.4 Trabalhos futuros**

O trabalho apresentado nesta dissertação de mestrado colabora para o avanço da área de pesquisa LPN, que é ainda pouco explorada, e propicia diversos outros trabalhos futuros, conforme citados a seguir:

- Evolução das arquiteturas de referência Cambuci e Cambuci-LPN visando incorporar outras visões, como é o caso da visão de implementação, para facilitar a especialização e instanciação das mesmas;
- Incorporar melhorias na arquitetura de referência Cambuci-LPN para eliminar as deficiências identificadas durante a avaliação conduzida baseada em *checklist*. Na Seção 6.5(Quadro 6.16 e Quadro 6.17), o autor desta dissertação apresenta algumas soluções para dirimir tais deficiências e que podem ser utilizadas para apoiar o refinamento da Cambuci-LPN;
- Instanciação da arquitetura de referência Cambuci-LPN visando a construção de um repositório baseado em serviço de LPN para apoiar a abordagem GLPN;
- Integração do repositório baseado em serviço de LPN, criado a partir da instanciação da Cambuci-LPN, com outras ferramentas computacionais;
- Criação de um perfil de LPN para a ontologia ONTO-ResAsset, considerando os conceitos sobre LPN identificados neste trabalho;
- Especialização da arquitetura de referência Cambuci visando a concepção de arquiteturas de referência de repositórios de outros tipos de ativos, como requisitos, casos de teste, etc;
- Validação da Cambuci e da Cambuci-LPN utilizando um número significativo de avaliadores a fim de eliminar ameaças à validade da avaliação conduzida; e
- Condução de estudos empíricos para comprovar a eficácia das arquiteturas de referência Cambuci e Cambuci-LPN.

# Referências

AALST, W.VAN DER; TER HOFSTEDE, A.; WESKE, M. **Business process management: A survey**. BPM'03 Proceedings of the 2003 international conference on Business process management. **Anais...**Berlin, Heidelberg: Springer-Verlag, p. 1-12, 2003.

ALENCAR, F.M.R. **Mapeando a Modelagem Organizacional em Especificações Precisas**. 1999. Tese (Doutorado em Ciências da Computação) Centro de Ciências Exatas e da Natureza, Universidade Federal de Pernambuco, UFPE, Brasil, 1999.

ARIS. **Aris Community**. Disponível em: <<http://www.ariscommunity.com/aris-express>>. Acesso em: 10 jul. 2013.

ARSANJANI, A.; ZHANG, L.J.; ELLIS, M.; ALLAM, A.; CHANNABASAVIAH, K. S3: **A service-oriented reference architecture**. IT Professional, v. 9, n. 3, p.10-17, 2007.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software architecture in practice**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2012.

BIZAGI. **Bizagi BPM Suite**. Disponível em: <<http://www.bizagi.com>>. Acesso em: 10 jul. 2013.

BOFFOLI, N. et al. **Driving flexibility and consistency of business processes by means of product-line engineering and decision tables**. Product Line Approaches in Software Engineering (PLEASE), 2012 3rd International Workshop on. **Anais...**p. 33-36, 2012a

BOFFOLI, N. et al. Business Process Lines and Decision Tables Driving Flexibility by Selection. In: GSCHWIND, T. et al. (Eds.). **Software Composition**. Berlin, Heidelberg: Springer Berlin Heidelberg, v. 7306p. 178-193. 2012b.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language user guide**. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2005.

BOEHM, B. **A view of 20<sup>th</sup> and 21<sup>st</sup> century software engineering**. In Proceedings of the 28<sup>th</sup> International Conference on Software Engineering (ICSE 2006), New York, NY, USA: ACM, 2006, p. 12-29.

CHINOSI, M.; TROMBETTA, A. **BPMN: An introduction to the Standard**. **Computer Standards & Interfaces**, v. 34, n. 1, p. 124–134. 2011.

CHOI, I. et al. An XML-based process definition language for integrated process management. **Comput. Ind.**, v. 50, n. 1, p. 85-102, 2003.

CHOI, H.; LIM, C.; KIM, J. **Defining reference architecture for NTIS development.** In: Proceeding of the 11<sup>th</sup> International Conference on Advanced Communication Technology (ICACT 2009), Phoenix Park, Korea, 2009, p. 284-287.

CLOUTIER, R. et al. The Concept of Reference Architectures. **Syst. Eng.**, v. 13, n. 1, p. 14-27, 2010.

CRUZ, E. F.; MACHADO, R. J.; SANTOS, M. Y. **From Business Process Modeling to Data Model: A Systematic Approach.** Proceedings of the 2012 Eighth International Conference on the Quality of Information and Communications Technology. **Anais...** Washington, DC, USA: IEEE Computer Society, p. 205-210, 2012.

DAVIS, M. M.; AQUILIANO, N. J.; CHASE, R. B. **Fundamentos da Administração da Produção.** 3a. ed., Ed. Bookman. 2007.

DEBAUD, J.-M.; FLEGE, O.; KNAUBER, P. **PuLSE-DSSA - A method for the development of software reference architectures.** Proceedings of the third international workshop on Software architecture. **Anais...**New York, NY, USA: ACM, p. 25-28, 1998.

DILLON, T.S.; WU, C.; CHANG, E. **Reference architectural styles for service-oriented computing.** In: Proceeding of the 4<sup>th</sup> International Conference of Network and Parallel Computing (NPC 2007), Dalian, China: Springer-Verlag, 2007, p. 543-555 (LNCS v.4672).

DURO, N.; MOREIRA, F.; ROGADO, J.; REIS, J.; PECCIA, N. **Technology harmonization – developing a reference architecture for the ground segment software.** In: IEEE Aerospace Conference, Big Sky, Montana, USA, 2005, P. 3968-3979.

ECLIPSE FOUNDATION. **Eclipse.** Disponível em: <<http://www.eclipse.org/>>. Acesso em: 10 jul. 2013.

ELHADAD, M.; BALABAN, M.; STURM, A. Effective Business Process Outsourcing: The Prosero Approach. **International Journal of Interoperability in Business Information Systems**, v. 3, n. 1, p. 8-31, 2008.

ERIKSSON, H.; PENKER, M. **Business Modeling with UML – Business Patterns at Work.** John Wiley & Sons, Inc. 2000.

ERL, T. **SOA-Princípios de design de serviços.** São Paulo: Pearson Prentice Hall, 2009.

FIORAVANTI, M. L.; NAKAGAWA, E. Y.; BARBOSA, E. F. **EDUCAR: uma arquitetura de referência para ambientes educacionais.** In: Anais do 21<sup>o</sup> Simpósio Brasileiro de Informática na Educação (SBIE 2010), João Pessoa, Paraíba, 2010, p. 1-10.

FIORINI, S. T.; LEITE, J. C. S. DO P.; DE LUCENA, C. J. P. **Process Reuse Architecture**. Proceedings of the 13th International Conference on Advanced Information Systems Engineering. **Anais...**London, UK: Springer-Verlag, v. 2068, p. 284-298, 2001.

FLICKR. **Flickr**. Disponível em: <<http://www.flickr.com/>>. Acesso em: 10 jul. 2013.

GIMENES, I. M. D. S.; FANTINATO, M.; TOLEDO, M. B. F. DE. A Product Line for Business Process Management. **2008 12th International Software Product Line Conference**, p. 265-274, 2008.

GRÖNER, G.; WENDE, C.; BOSKOVIC, M. *et al.* **Validation of families of business process**. Lecture Notes in Computer Science (Including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in ioinformatics), v. 6741 LNCS, p. 551-565, 2011.

HEMALATHA, T.; ATHISHA, G.; JEYANTHI, S. **Dynamic web service based image processing system**. In Proceeding of the 16<sup>th</sup> International Conference on Advanced Computing and Communications (ADCOM 2008), Chennai, India, 2008, p. 323-328.

HONGMIN, R. .; JIN, L. .; JINGZHOU, Z. . **Software asset repository open framework supporting customizable faceted classification**. Proceedings 2010 IEEE International Conference on Software Engineering and Service Sciences, ICSESS 2010. **Anais...**Beijing: p. 1-4, 2010.

IBM CORPORATION. **Rational Unified Process - Best Practices for Software Development Teams**. 2011.

IEEE, **Recommended practices for architectural descriptions of software-intensive systems**, IEEE, Tech. Rep. Std 1471-2000, 2000.

ISO/IEC 12207. **Engenharia de Sistemas e Software – Processos de Ciclo de Vida de Software**. ABNT, 2009.

ISO/IEC/IEEE 42010. Systems and software engineering -- Architecture description. **ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)**, p. 1-46, 2011.

JOHANSEN, U.; STAV, E.; WANDERHAUG, S. **The mafia handbook - an architectural description framework for information integration systems**. SINTEF ICT, Tech. Rep. STF90. A05139, 2003.

JURIC, M. B.; PANT, K. **Business Process Driven SOA using BPMN and WS-BPEL: From Business Process Modeling to Orchestration and Service Oriented Architecture**. Birmingham: Packt Publishing. 2008.

KANG, K. C.; COHEN, S. G.; HESS, J. A.; NOVAK, W. E.; PETERSON, A. S.. **Feature-Oriented Domain Analysis (FODA) Feasibility Study**. Carnegie-Mellon University Software Engineering Institute. 1990.

KITCHENHAM, B. Procedures for Performing Systematic Reviews. **Keele UK Keele University**, v. 33, n. TR/SE-0401, p. 28, 2004.

KRUCHTEN, P. Architectural Blueprints: The “4+1” View Model of Software Architecture. **IEEE Software**, v. 12, n. 6, p. 42-50, 1995.

LADEIRA, S. A. Z.; CAGNIN, M. I. **Reutilização De Modelagem De Negócios Baseada Em Visões**. 2008. Dissertação (Mestrado em Ciência da Computação) – Centro Universitário Eurípedes de Marília, Fundação de Ensino Eurípedes Soares da Rocha, Marília, SP, 2008.

LAN, J.; LIU, Y.; CHAI, Y. **A solution model for service-oriented architecture**. In: Proceedings of the 7<sup>th</sup> World Congress on Intelligent Control and Automation (WCICA 2008), Chongqing, China, 2008, p. 4184-4189.

LANDRE, G. B. **GLPN – Uma Abordagem para Gestão de Linhas de Processos de Negócio**. Dissertação (Mestrado em Ciência da Computação)-FACOM-UFMS, Campo Grande, MS, 2012.

LANDRE, G.; PALMA, E.; PAIVA, D.; NAKAGAWA, E. Y.; CAGNIN, M. I. **vrBPMN\* and Feature Model: An Approach to Model Business Process Line**. In: 5th International Workshop on Process Model Collections: Management and Reuse (PMC-RM 2014), Eindhoven, Netherlands, p. 1-12, 2014.

LAPOUCHNIAN, A.; YU, Y.; MYLOPOULOS, J. **Requirements-driven design and configuration management of business processes**. Proceedings of the 5th international conference on Business process management. **Anais...Berlin, Heidelberg: Springer-Verlag**, p. 246-261, 2007.

LEPPANIEMI, J.; LINNA, P.; SONI, J.; JAAKKOLA, H. **Toward a flexible service-oriented reference architecture for situational awareness systems in distributed disaster knowledge management**. In: Proceedings of the International Conference on Management of Engineering & Technology (PICMET 2009), Portland, Oregon, USA, 2009, p. 959-965.

LU, R.; SADIQ, S.; GOVERNATORI, G. **On Managing Business Processes Variants**. *Data and Knowledge Engineering*, v. 68, n. 7, p. 642-664, 2009.

LUDWIG, H.; RANKIN, Y.; ENYEDI, R.; ANDERSON, L. **Process Variation Analysis Using Empirical Methods: A Case Study**. In RINDERLE-MA, S.; TOUMANI, F.; WOLF, K. (Eds.). *Business Process Management*. Springer Berlin / Heidelberg, 2011. V. 6896, p. 62-65.

MARACCI, F. V. **RofPN - Abordagem para reutilização de processos de negócio**. UEM, Maringá, PR, 2010.

MELAND, P. H. . et al. **An architectural foundation for security model sharing and reuse**. Proceedings - International Conference on Availability, Reliability and Security, ARES 2009. **Anais...**Fukuoka, Fukuoka Prefecture: p. 823-828, 2009.

MERSON, P. **Como documentar arquiteturas de software**. 19a. SBES (Simpósio Brasileiro de Engenharia de Software). **Anais...**Uberlândia, MG: 2005.

MEYER, M. H.; LEHNERD, A. P. **The Power of Product Platforms: Building Value and Cost Leadership**. Free Press. New York. 1997.

MTURI, E.; JOHANNESSON, P. A context-based process semantic annotation model for a process model repository. **Business Process Management Journal**, p. 404-430, 2013.

MURAKAMI, E.; SARAIVA, A. M.; RIBEIRO, JUNIOR, L. C. M.; CUGNASCA, C. E.; HIRAKAWA, A. R.; CORREA, P. L. P. **Na infraestrutura for the development of distributed service-oriented information systems for precision agriculture**. Computers and Eletronics in Agriculture, v. 58, n. 1, p. 37-48, 2007.

MULLER, G. **A Reference Architecture Primer**. Projeto Gaudi. 2012.

NAKAGAWA, E. Y. **Uma Contribuição ao Projeto Arquitetural de Ambiente de Engenharia de Software**. São Carlos, SP: ICMC-USP, São Carlos, SP, 2006.

NAKAGAWA, E. Y.; OQUENDO, F.; BECKER, M. **RAModel: A Reference Model for Reference Architectures**Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on. **Anais...**p. 297-301, 2012

NAKAGAWA, E. Y. et al. **Consolidating a Process for the Design, Representation, and Evaluation of Reference Architectures**Proceedings of the 2014 IEEE/IFIP Conference on Software Architecture. **Anais...**Washington, DC, USA: IEEE Computer Society, p. 143-152, 2014.

NUNES, V. .; WERNER, C. .; SANTORO, F. M. . **Context-based process line**ICEIS 2010 - Proceedings of the 12th International Conference on Enterprise Information Systems. **Anais...**Funchal, Madeira, Portugal: p. 277-282, 2010.

**OASIS Reference model for service oriented architecture 1.0**. Relatório Técnico, Advanced Open Standards for the Information Society (OASIS), 2006.

**OASIS Reference architecture for service oriented architecture version 1.0**. Relatório Técnico, Advanced Open Standards for the Information Society (OASIS), 2008.

OLIVEIRA, Lucas Bueno Ruas de. **Estabelecimento de uma arquitetura de referência orientada a serviços para ferramentas de teste de software**. 2011. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2011.

OMG-OBJECT MANAGEMENT GROUP. **Reusable Asset Specification**. OMG, 2005.

OMG-OBJECT MANAGEMENT GROUP. **Documents Associated With Business Process Model And Notation (BPMN) Version 2.0.2**. OMG, 2013.

OMG-OBJECT MANAGEMENT GROUP. **Documents Associated With UML Version 2.4.1**. OMG, 2011.

ORACLE. **Netbeans**. Disponível em: <[www.netbeans.org](http://www.netbeans.org)>. Acesso em: 10 jul. 2013.

OSSHIRO, Márcio. **Arquitetura de Referência para Repositórios de Linhas de Processos de Negócio**. Qualificação (Mestrado em Ciência da Computação)-FACOM-UFMS, Campo Grande, MS, 2013.

PAPAZOGLU, M.P.; TRAVERSO, P.; Dustdar, S.; LEYMANN, F. **Service-oriented computing: a research roadmap**. International Journal of Cooperative Information System, v. 17, n.2, p. 223-255, 2008.

PERISTERAS, V.; FRADINHO, M.; LEE, D.; PRINZ, W.; RULAND, R.; IQBAL, K.; DECKER, S. **CERA: A collaborative environment reference architecture for interoperable CWE systems**. Service Oriented Computing and applications, v. 3, n. 1, p. 3-23, 2009.

POHL, K.; BOCKLE, G.; VAN DER LINDEN, F. J. **Software Product Line Engineering: Foundations, Principles and Techniques**. 1st. ed. [S.l.] Springer Publishing Company, Incorporated, 2005.

RAMANATHAN, S.; ALEXANDER, M.; KERR, G. **The IBM telecommunications service delivery platform**. IBM Systems Journal, v. 47, n. 3, p. 433-443, 2008.

REED, P. **Reference Architecture: The best of best practices**. Disponível em: <<http://www.ibm.com/developerworks/rational/library/2774.html>>. 2002.

REIFF-MARGANIEC, S.; TRUONG, H. L.; CASELLA, G.; DORN, C.; DUSTDAR, S.; MORETZKY, S. **The incontext pervasive collaboration services architecture**. In: Proceedings of the 1<sup>st</sup> European Conference on Towards a Service-Based Internet, Madrid, Spain: Springer-Verlag, 2008, p. 134-146 (LNCS v. 5377).

REISIG, W. **Petri nets: an introduction**. New York, NY, USA: Springer-Verlag New York, Inc., 1985.

ROLLAND, C.; NURCAN, S. **Business Process Lines to Deal with the Variability**. HICSS'2010. Anais... Koloa, Kauai, HI, USA: p.1-10, 2010

SANTANA, E. C.; MACIEL, C.; TURINE, M. A. S. Investigando o uso de conceitos de Linhas de Produto de Software para o reúso de Modelos de Processos de Negócios. **VII Simposio Brasileiro de Sistemas de Informação**, p. 525-532, 2011.

SANTOS, J. F. M.; GUESSI, M.; GALSTER, M.; FEITOSA, D.; NAKAGAWA, E., **A checklist for evaluation of reference architectures for embedded systems**, in SEKE'13, Boston, USA, p. 1-4, 2013.

ROCHA, R. S.; FANTINATO, M. **The use of software product lines for business process management: A systematic literature review**. Information and Software Technology, v. 55, p. 1355-1373, 2013.

ROSA, M. LA; DUMAS, M.; TER HOFSTEDE, A. H. M.; MENDLING, J. **Configurable multi-perspective business process models**. Information Systems, v. 36, n. 2, p. 313-340, 2011.

SCA, S. C. OF C – STANDARDS COUNCIL OF CANADA, CONSEIL CANADIEN DES NORMES. **Decision Tables**. [S.l.: s.n.]. Disponível em: <<http://www.scc.ca/en/standardsdb/standards/1961>>. SCA. 1970.

SCHNIEDERS, A.; PUHLMANN, F. **Variability Mechanisms in E-Business Process Families** (W. Abramowicz & H. C. Mayr, Eds.)Engineering. **Anais...GI**, Berlin, p. 583-601, 2006.

SILVA, L. E. P. DA. **ONTO-ResAsset: Ontologia de Ativos Reutilizáveis**.Dissertação (Mestrado em Ciência da Computação)-FACOM-UFMS, Campo Grande, MS, 2013.

TERENCIANI, M.F. **Um arcabouço computacional de apoio à criação de linha de processos de negócio**. 2014. Qualificação (Mestrado em Ciência da Computação)-FACOM-UFMS, Campo Grande, MS, 2014.

THELING, T. et al. **An architecture for collaborative scenarios applying a common BPMN-repository**Distributed Applications and Interoperable Systems. **Anais... Springer-Verlag**, Berlin, Heidelberg, p. 169-180, 2005.

TOYOHARA, Rubens Kenji Takaki. **Estudo da influência da web services no desempenho de uma arquitetura orientada a serviços com QoS**. 2011. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, University of São Paulo, São Carlos, 2011.

TÜRETKEN, O.; DEMIRÖRS, O. Plural: A decentralized business process modeling method. **Information & Management**, v. 48, n. 6, p. 235-247, 2011.

U.S. DEPARTMENT OF DEFENSE. **DoDAF - Department of Defense Architecture Framework version 2.02**. 2010.

U.S. FEDERAL GOVERNMENT. **FEAF- The common approach to Federal Enterprise Architecture**. 2012.

VAISMAN, A. An Introduction to Business Process Modeling. In: AUFAURE, M.-A.; ZIMÁNYI, E. (Eds.). **Business Intelligence SE - 2**. [S.l.] Springer Berlin Heidelberg, v. 138, p. 29-61, 2013.

W3C. **Web Services Description Language (WSDL) 1.1**. W3C, 2001.

WANG, H. J.; WU, H. Supporting process design for e-business via an integrated process repository. **Inf. Technol. and Management**, v. 12, n. 2, p. 97-109, 2011.

WHITE, S. A. **Introduction to BPMN**. IBM Corporation. 2004.

YAN, Z.; DIJKMAN, R.; GREFFEN, P. Business process model repositories – Framework and survey. **Information and Software Technology**, v. 54, n. 4, p. 380-395, 2012.

ZACHMAN, J. A. **About the Zachman Framework**. Disponível em: <<http://www.zachman.com/about-the-zachman-framework>>. Acesso em: 10 jul. 2013.

ZHAO, X.; LIU, C. Version Management in the Business Process Change Context. In: ALONSO, G.; DADAM, P.; ROSEMANN, M. (Eds.). **Business Process Management**. [S.l.] Springer Berlin / Heidelberg, v. 4714p. 198-213. 2007.

ZHENG, Q. H.; DONG, B.; TIAN, F.; CHEN, W. **A service-oriented approach to integration of e-learning information and resource management systems**. In: Proceeding of the 12<sup>th</sup> International Conference on Computer Supported Cooperative Work in Design, Xian, China, 2008, p. 1047-1052.

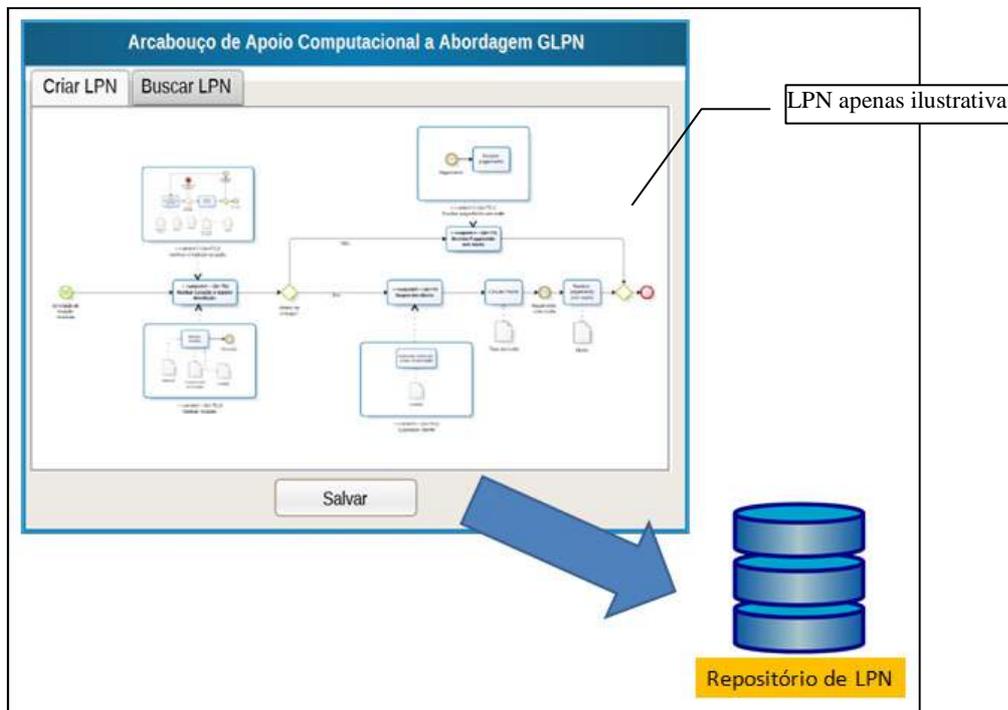
ZIMMERMAN, O.; KOOP, P.; PAPPE, S. **Architectural knowledge in an SOA infrastructure reference architecture**. In: Software Architecture Knowledge Management, Berlin, Heidenberg: Springer-Verlag, p. 217-241, 2009.

## Apêndice A – Exemplo de ferramenta computacional de apoio a abordagens LPN

A seguir são apresentados alguns cenários em que pode haver interação entre ferramentas computacionais de apoio a abordagens de LPN, como é o caso do *BLP framework*, e repositórios de LPN.

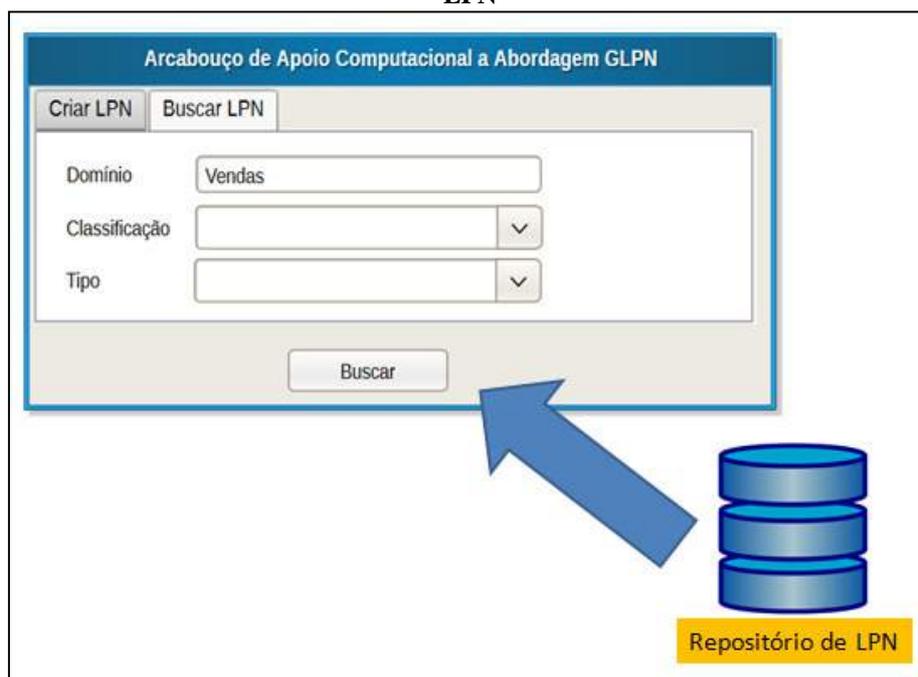
Na Figura A.1 tem-se a interação de um repositório de LPN com uma ferramenta computacional fictícia de apoio a abordagens LPN durante a operação Salvar LPN, no qual é solicitado ao repositório de LPN para armazenar o ativo de LPN.

**Figura A.1 – Ferramenta computacional fictícia de apoio a abordagens LPN – operação Salvar LPN**

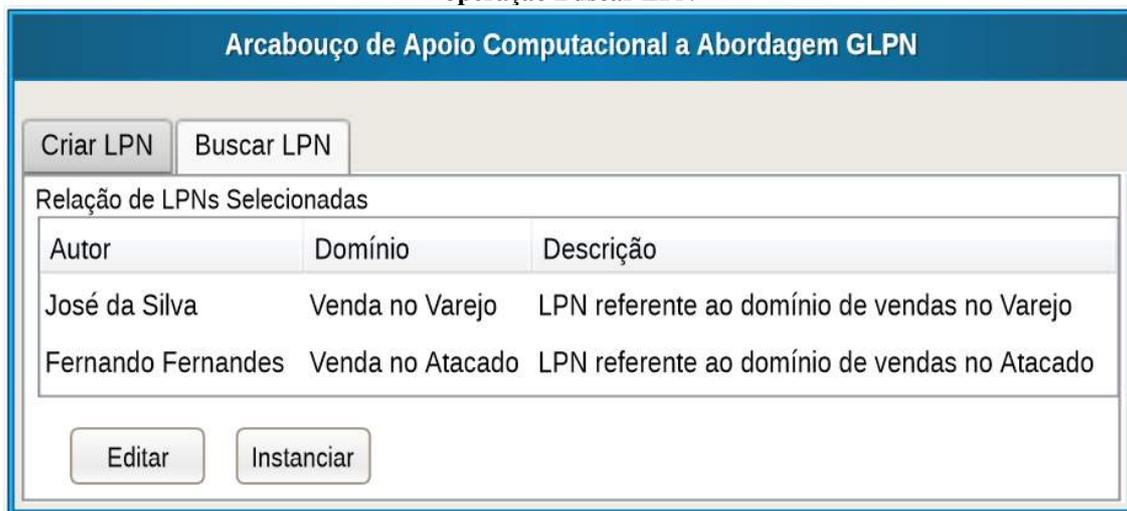


Na Figura A.2 é mostrada a interação entre a ferramenta computacional e o repositório durante a operação “Buscar” LPN. Após informar os parâmetros da busca, é solicitado ao repositório que execute o seu mecanismo de busca para retornar as LPNs correspondentes aos parâmetros informados. Um exemplo de resultado de busca é mostrado na Figura A.3. A partir de uma LPN retornada pela busca, a ferramenta computacional pode permitir a sua edição (*Editar*) e a sua instanciação (*Instanciar*).

**Figura A.2 - Ferramenta computacional fictícia de apoio a abordagens LPN – operação Buscar LPN**

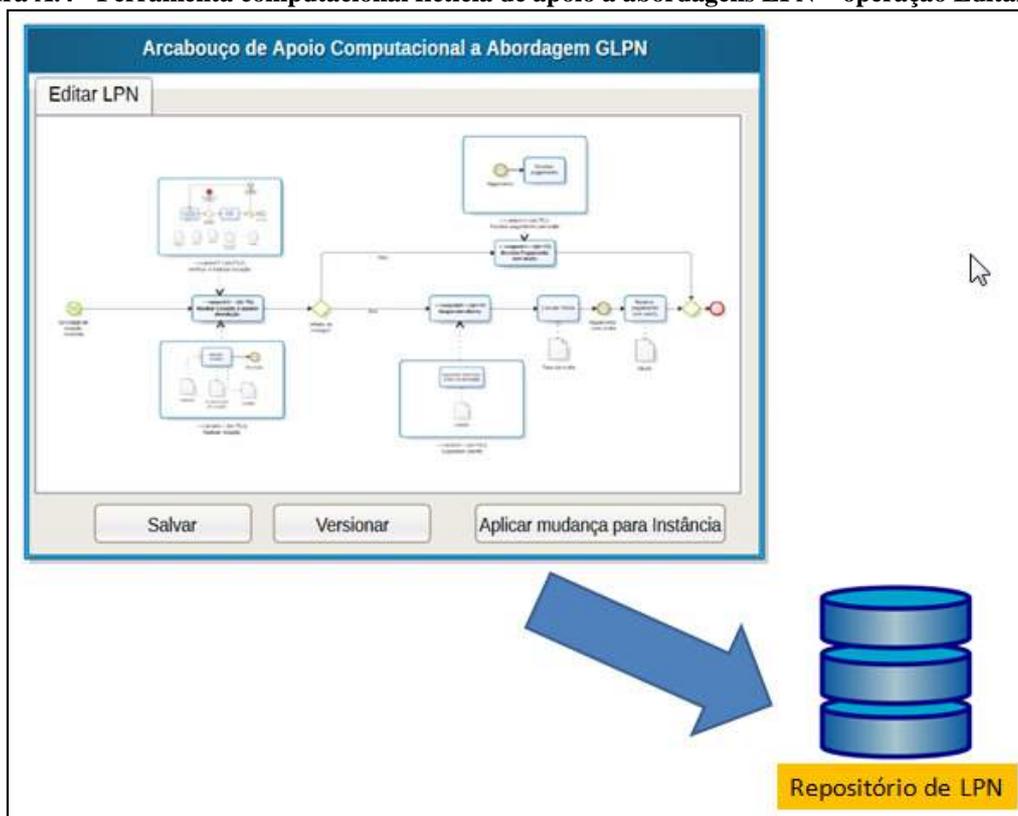


**Figura A.3 - Ferramenta computacional fictícia de apoio a abordagens LPN – resultado da operação Buscar LPN**



Na operação `Editar`, a LPN selecionada pode ser modificada, conforme ilustrado na Figura A.4. No momento de salvar a LPN, pode-se criar uma nova versão (utilizando o mecanismo de versionamento do repositório) e também pode ser necessário espalhar essas alterações para as instâncias dessa LPN (utilizando o mecanismo de rastreabilidade do repositório entre a LPN e as suas instâncias).

Figura A.4 - Ferramenta computacional fictícia de apoio a abordagens LPN – operação Editar LPN



Na operação Instanciar LPN, ilustrada na Figura A.5, a ferramenta computacional solicita a configuração da LPN para que o modelo de processo de negócio possa ser obtido (instância da LPN). Após essa configuração, a ferramenta computacional também pode permitir a inclusão de novos elementos BPMN no processo de negócio. Quando a instância da LPN for armazenada no repositório, a ferramenta computacional irá verificar se as mudanças realizadas na instância também podem ser aplicadas na LPN. Essa funcionalidade pertence ao mecanismo de *feedback* do repositório de LPN. Se as modificações propostas na instância forem incorporadas na LPN, ocorre o mesmo processo anteriormente descrito, que é o de espalhar as alterações para as instâncias dessa LPN (mecanismo de rastreabilidade do repositório entre a LPN e as suas instâncias).

Figura A.5 - Arcabouço de apoio computacional a abordagem GLPN – operação instanciar LPN

