

**GUILHERME DE ANDRADE LEMESZENSKI**

**APLICAÇÃO DO MODELO RM-ODP PARA A PROPOSTA DE UMA  
ARQUITETURA DE SOFTWARE ORIENTADA A SERVIÇOS COM APLICAÇÃO  
EM SISTEMAS DE ENSINO A DISTÂNCIA**

São Paulo  
2014

**GUILHERME DE ANDRADE LEMESZENSKI**

**APLICAÇÃO DO MODELO RM-ODP PARA A PROPOSTA DE UMA  
ARQUITETURA DE SOFTWARE ORIENTADA A SERVIÇOS COM APLICAÇÃO  
EM SISTEMAS DE ENSINO A DISTÂNCIA**

Monografia apresentada à Escola Politécnica  
da Universidade de São Paulo para a conclusão  
do curso de MBA em Tecnologia de  
Informação.

São Paulo  
2014

**GUILHERME DE ANDRADE LEMESZENSKI**

**APLICAÇÃO DO MODELO RM-ODP PARA A PROPOSTA DE UMA  
ARQUITETURA DE SOFTWARE ORIENTADA A SERVIÇOS COM APLICAÇÃO  
EM SISTEMAS DE ENSINO A DISTÂNCIA**

Monografia apresentada à Escola Politécnica  
da Universidade de São Paulo para a conclusão  
do curso de MBA em Tecnologia de  
Informação.

Orientador: Prof. Dr. Reginaldo Arakaki

Coorientador: Leandro Rodrigues de Souza

São Paulo  
2014

MBA/II  
2014  
L 543a

612



Escola Politécnica - EPEL



31500023612

## FICHA CATALOGRÁFICA

M2014 AK \*

**Lemeszenski, Guilherme de Andrade**

**Aplicação do modelo RM-ODP para a proposta de uma arquitetura de software orientada a serviços com aplicação em sistemas de ensino a distância / G.A. Lemeszenski. -- São Paulo, 2014.**

**54 p.**

**Monografia (MBA em Tecnologia da Informação) - Escola Politécnica da Universidade de São Paulo. Programa de Educação Continuada em Engenharia.**

**1.Arquitetura orientada a serviços I.Universidade de São Paulo. Escola Politécnica. Programa de Educação Continuada em Engenharia II.t.**

[2713536]

## **AGRADECIMENTOS**

Ao Prof. Dr. Reginaldo Arakaki por sua atenção e orientação indispensável para a elaboração deste trabalho.

Ao meu coorientador Leandro Rodrigues Souza por toda a sua ajuda de vital importância para a conclusão desta monografia.

A minha família pela paciência e apoio durante toda a execução deste trabalho.

## RESUMO

O objetivo deste trabalho é apresentar a proposta de uma arquitetura de software orientada a serviços *SOA (Service Oriented Architecture)* para sistemas de uma instituição real de ensino a distância com o intuito de solucionar alguns tipos de erros comuns que ocorrem nos processos de negócio e reduzir esforços em tarefas manuais na realização destes processos. Esta proposta é desenvolvida utilizando as visões do modelo de referência para processamento aberto distribuído *RM-ODP (Reference Model for Open Distributed Processing)* e conceitos de Arquitetura de Software. Este trabalho realiza a validação da proposta através da aplicação do método para análise de arquiteturas *ATAM (Architecture Tradeoff Analysis Method)*.

## **ABSTRACT**

This work presents a service oriented architecture (SOA) proposal for a real distance learning education institution to solve some common occurrence of errors in the current business process and save man hours to complete those processes. The proposal is developed using the views of the Reference Model for Open Distributed Processing (RM-ODP) and Software Architecture concepts. This work validates the proposal using the Architecture Tradeoff Analysis Method (ATAM).

## Lista de Ilustrações

Figura 1- Gráfico de distribuição de requisições de páginas ao longo do dia .....	25
Figura 2- Processo de matrícula atual .....	26
Figura 3 - Fluxo de atividades EaD .....	27
Figura 4 - Visão Informação: hierarquia de cursos, disciplinas e turmas.....	30
Figura 5 - Visão Informação: processo matrícula .....	30
Figura 6 - Visão Informação: processo acessar conteúdo .....	31
Figura 7 - Visão Informação: processo obter histórico escolar .....	32
Figura 8 - Visão Computação: processo de matrícula.....	33
Figura 9 - Visão Computação: confirmação de matrícula.....	34
Figura 10 - Visão Computação: processo acessar conteúdo.....	34
Figura 11 - Visão Computação: processo de obtenção de histórico escolar .....	35
Figura 12 - Visão Engenharia: diagrama de infra-estrutura/integração .....	36
Figura 13 - Visão Engenharia: serviços de matrícula.....	37
Figura 14 - Visão Engenharia: serviços de consulta de histórico escolar .....	37
Figura 15 - Visão Engenharia: serviços de consulta de conteúdo didático .....	38
Figura 16 - Visão Engenharia: barramento de integração .....	41



## **Lista de Tabelas**

Quadro 1- Árvore de utilidades do ATAM .....	45
Quadro 2 - Análise Arquitetural – Cenário C1 .....	46
Quadro 3 - Análise Arquitetural - Cenário C2 .....	47
Quadro 4 - Análise Arquitetural - Cenário C4 .....	48
Quadro 5 - Análise Arquitetural - Cenário C7 .....	49

## **Lista de Abreviaturas e Siglas**

<b>ABED</b>	Associação Brasileira de Educação a Distância
<b>ATAM</b>	<i>Architecture Tradeoff Analysis Method</i>
<b>AVA</b>	Ambiente virtual de aprendizagem
<b>EaD</b>	Ensino a Distância
<b>ESB</b>	<i>Enterprise Service Bus</i>
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>MTBF</b>	<i>Mean Time Between Failures</i>
<b>MTTR</b>	<i>Mean Time To Repair</i>
<b>OKI</b>	<i>Open Knowledge Initiative</i>
<b>OSID</b>	<i>Open Service Interface Definitions</i>
<b>RM-ODP</b>	<i>Reference Model for Open Distributed Processing</i>
<b>SOA</b>	<i>Service Oriented Architecture</i>
<b>SOAP</b>	<i>Simple Object Access Protocol</i>
<b>TI</b>	Tecnologia de Informação
<b>UML</b>	<i>Unified Modeling Language</i>
<b>URL</b>	<i>Uniform Resource Location</i>
<b>WSLD</b>	<i>Web Services Description Language</i>
<b>XML</b>	<i>Extensible Markup Language</i>

## SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	Objetivo .....	12
1.2	Motivação .....	12
1.3	Metodologia .....	12
1.4	Estrutura do Trabalho .....	13
2	CONCEITOS TEÓRICOS .....	14
2.1	Ensino à Distância .....	14
2.2	Atributos de Qualidade de Software .....	15
2.3	Arquitetura de Software .....	16
2.4	O modelo de referência RM-ODP .....	17
2.5	Arquitetura Orientada a Serviço .....	18
2.5.1	Barramento de Serviços Corporativos .....	19
2.5.2	Serviço Web .....	19
2.6	Método de Avaliação de Arquitetura por <i>Tradeoff</i> (ATAM) .....	20
3	PROPOSTA DE ARQUITETURA DE INTEGRAÇÃO.....	22
3.1	Restrições e Características da Proposta.....	22
3.2	Visões RM-ODP .....	22
3.2.1	Visão Empresa .....	23
3.2.2	Dificuldades encontradas nos processos atuais .....	25
3.2.3	Definição da nova proposta .....	26
3.2.4	Visão Informação .....	28
3.2.5	Visão Computação.....	33
3.2.6	Visão Engenharia.....	35
3.2.7	Visão Tecnologia.....	41
4	Verificação e Avaliação do Resultado.....	43
4.1	Passo 1 – Apresentação do ATAM.....	43
4.2	Passo 2 – Apresentação dos objetivos do negócio.....	43
4.3	Passo 3 – Apresentação da arquitetura .....	44
4.4	Passo 4 – Identificação das decisões arquiteturais.....	44
4.5	Passo 5 – Geração da árvore de utilidades.....	44
4.6	Passo 6 – Análise das decisões arquiteturais .....	46
4.7	Passo 9 – Consolidar Resultados .....	49

5	CONSIDERAÇÕES FINAIS .....	51
5.1	Objetivo Atingido e Contribuições .....	51
5.2	Dificuldades Encontradas .....	51
5.3	Trabalhos Futuros .....	52
6	LISTA DE REFERÊNCIAS.....	53

## 1 INTRODUÇÃO

Atualmente, o Ensino a Distância (EaD) vêm se mostrando cada vez mais como um habilitador para a democratização do acesso à educação oferecendo novas formas de aprendizagem aos seus alunos. O crescimento desta modalidade ocorre por conta de sua flexibilidade na definição de locais e horários para a realização de atividades de ensino.

Impulsionado pelo avanço de novas tecnologias principalmente das últimas décadas, o EaD suporta cada vez mais o oferecimento de atividades de aprendizado a baixos custos superando barreiras de distância e tempo. Litwin (2001) afirma em seu livro que, com a utilização de novas tecnologias, a distância física deixou de ser a característica principal da educação a distância.

Com a intenção de comparar o ensino a distância ao ensino presencial, Tori (2010) acredita que a introdução de tecnologias interativas na educação, e a consequente convergência entre educação virtual e convencional, as tradicionais formas de classificação de cursos como “presenciais” e “a distância” se tornaram obsoletas.

Para tornar possível o desenvolvimento e evolução de questões tecnológicas, cada vez mais empresas e instituições perceberam a importância da área de Tecnologia de Informação (TI).

Com um número cada vez maior de sistemas de informação criados para o atendimento de necessidades específicas de processos de negócio de instituições de ensino, questões como manutenção, reutilização e interoperabilidade são desafios cada vez maiores para as áreas de TI.

A execução de projetos para a solução destes desafios geralmente são caras e demoradas, além do fato de impactarem na forma como os processos de negócio são realizados. Estes fatores somados influenciam contra a adoção de medidas evolutivas em processos e sistemas já bem estabelecidos. Apesar dos contrapontos mencionados, os benefícios alcançados com a solução destas dificuldades são observados na maior agilidade de instituições em resposta às novas tendências e redução de custos com a economia de horas de trabalho necessárias para as manutenções sistêmicas e automatização de processos manuais.

Neste contexto, sistemas desenvolvidos em SOA (*Service Oriented Architecture*) oferecem uma nova perspectiva na tentativa de tratar estas dificuldades geralmente encontradas, promovendo o reuso de componentes e facilitando a integração entre sistemas, permitindo que alterações possam ser realizadas com maior velocidade atendendo as demandas do negócio.

Este tipo de arquitetura oferece tratamento para questões de tecnologia com uma perspectiva mais próxima dos processos de negócio, desta forma, a entrega de funcionalidades dos sistemas na forma de serviços é realizada levando em conta os valores desejados pelos usuários. Isto permite que as áreas de negócio e tecnologia possam evoluir de forma conjunta.

## **1.1 Objetivo**

O objetivo deste trabalho é propor uma arquitetura de integração orientada a serviços que atenda as necessidades de uma instituição pública de ensino a distância. A arquitetura proposta visa superar dificuldades causadas por procedimentos manuais e economizar horas de trabalho relativas aos atuais processos de negócio em execução.

## **1.2 Motivação**

Apesar do avanço contínuo na área de arquitetura de software estar em constante evolução criando novos caminhos e desenvolvendo métodos, ainda é possível se observar a dificuldade de empresas e instituições na aplicação destes conceitos.

O planejamento e aplicação de tais conceitos nas fases iniciais de um projeto podem garantir que requisitos de qualidade sejam alcançados, representando economia de tempo e recursos em fases mais avançadas do projeto. Valipour (2009) afirma que soluções baseadas em arquiteturas fornecem meios de controlar a complexidade e evolução de softwares.

Este trabalho apresenta uma proposta de arquitetura de software utilizando padrões, modelos e conceitos de engenharia de software para resolução de problemas encontrados em uma instituição real de ensino à distância.

A contribuição deste trabalho reside na apresentação de uma arquitetura de software que contribua com a superação das dificuldades encontradas nos processos estudados e apresente viabilidade prática de aplicação por parte da instituição.

## **1.3 Metodologia**

Este trabalho foi realizado à partir de observações sobre os processos de negócio atuais de uma instituição pública de ensino a distância e das dificuldades encontradas pela mesma. À partir destas observações, foi realizada uma pesquisa inicial para se obter o embasamento teórico sobre conceitos de arquitetura de software e SOA.

A próxima etapa realizada foi propor uma solução utilizando os conceitos pesquisados que pudesse ser aplicada ao cenário da instituição de ensino estudada para a resolução das dificuldades identificadas.

A estruturação e desenvolvimento da proposta foi realizada utilizando as cinco visões do Modelo de Referência para Processamento Aberto e Distribuído RM-ODP (*Reference Model for Open Distributed Processing*) e conceitos de Arquitetura de Software.

A última fase consistiu na aplicação do Método de Análise Arquitetural por Tradeoff ATAM (*Architecture Tradeoff Analysis Method*) para validação da arquitetura através da análise dos requisitos de qualidade e dos riscos existentes na proposta.

#### **1.4 Estrutura do Trabalho**

Neste capítulo é apresentada a introdução sobre o tema abordado, os objetivos do trabalho e a motivação encontrada.

O capítulo 2 foca na apresentação dos conceitos necessários para a compreensão deste trabalho. São exibidos conceitos básicos sobre EaD, além de uma breve explicação sobre Atributos de Qualidade de Software, Arquitetura de Software, SOA, RM-ODP e o ATAM.

O capítulo 3 apresenta o desenvolvimento da solução proposta por este trabalho. É utilizado o modelo RM-ODP para a definição de uma arquitetura orientada a serviços aplicada aos sistemas de EaD e, em seguida, realizada a validação utilizando-se o método ATAM.

O capítulo 4 encerra o trabalho, apresentando as conclusões tomadas durante o desenvolvimento deste documento e sugestões para futuros trabalhos.

## 2 CONCEITOS TEÓRICOS

### 2.1 Ensino à Distância

EaD é uma modalidade de ensino na qual prevalece o auto estudo, ou seja, o aluno se instrui utilizando materiais criados e oferecidos pela instituição de ensino. Geralmente, o acompanhamento é realizado por um ou mais professores que podem participar de forma síncrona ou assíncrona das atividades planejadas. A comunicação entre professor e aluno pode ser realizada por diferentes tipos de meios, entre eles estão os meios impressos, mecânicos, eletrônicos e outros.

Sobre a evolução do EaD, existem diferentes definições sobre as gerações que representam o desenvolvimento desta modalidade, porém alguns autores concordam com a divisão em quatro gerações. De acordo com Anderson e Dron (2014) e Gerhardt (2005), estas gerações são as seguintes:

- Primeira geração: marcada pelo envio de materiais por postagem, representava a distribuição do ensino além da localização física das instituições;
- Segunda geração: início da utilização de meios de comunicação em massa (como televisão, rádio, fitas cassetes, etc.) para a difusão dos materiais de aprendizagem;
- Terceira geração: iniciada com a introdução de tecnologias interativas envolvendo diversos participantes em rede. Nesta fase, é iniciado o uso de videoconferências;
- Quarta geração: a quarta etapa está fortemente ligada aos avanços da *internet*, como a utilização de *wikis* e redes sociais, além da utilização de dispositivos móveis como *tablets* e *smartphones*.

Anderson e Dron (2014) afirmam que, com o avanço da tecnologia, esta modalidade começou a oferecer benefícios como baixo custo e atividades de aprendizagem com alta interatividade superando as barreiras de tempo e distância.

Segundo Gerhardt (2005), esta modalidade permite que universidades possam oferecer seus cursos para novos mercados fora de seu alcance geográfico regular, facilitando o processo educacional de uma forma geral.

Com relação aos sistemas de ensino a distância, além da utilização de sistemas tradicionais de gestão acadêmica, os Ambientes Virtuais de Aprendizagem (AVA) se apresentam como a principal forma de difusão de EaD pela internet.



## 2.2 Atributos de Qualidade de Software

Para a definição do que é qualidade de software, Dehkharghani e Yilmaz (2013) entendem que qualidade de software está associada ao *design* de um produto de software e ao quanto este produto satisfaz às expectativas de seus usuários. Seguindo a mesma linha, Bass; Clements e Kazman (2003) sugerem que a representação de funcionalidades sistêmicas em estruturas de software determina a capacidade de uma arquitetura suportar certas qualidades. Bass; Clements e Kazman (2003) ainda prosseguem definindo que considerações de negócio determinam quais qualidades devem ser contempladas pela arquitetura do software.

Seguindo estes conceitos, é possível de se observar o papel importante que o *design* de arquiteturas de software tem sobre a concretização das qualidades esperadas pelos usuários. Já Pontes (2012) ressalta que atributos de qualidade não existem isoladamente e acabam exercendo influência uns sobre os outros, pois afetam partes comuns da arquitetura. Esta influência entre atributos de qualidade é conhecida como *tradeoff*.

A Organização Internacional para Padronização (da sigla em inglês ISO) define o padrão ISO/IEC 9126-1:2007 para avaliação de software. Este padrão, por sua vez, apresenta a lista de atributos de qualidade de software exibida a seguir:

- **Funcionalidade:** qualidade que representa a capacidade do software em realizar as funções esperadas. Este atributo pode ser caracterizado por questões de adequação, precisão, interoperabilidade e segurança;
- **Confiabilidade:** representa a capacidade de um sistema em manter determinado nível de serviço, mesmo sobre a ocasião de cenários adversos. Questões como maturidade, tolerância a falhas e recuperabilidade são aspectos deste atributo;
- **Usabilidade:** qualidade que representa a facilidade do usuário em aprender e utilizar o sistema. Aspectos como compreensibilidade, facilidade de aprendizado e operabilidade são componentes deste atributo;
- **Eficiência:** este atributo está relacionado ao uso de recursos do sistema durante a execução de suas atividades.
- **Manutenibilidade:** representa a capacidade do software em ser alterado durante seu processo de evolução. As principais características deste atributo são analisabilidade, testabilidade e modificabilidade;

- **Portabilidade:** este atributo representa a quantidade de alterações necessárias para que o sistema tenha seus requisitos em ambiente de execução alterados. Quanto menor o número de alterações necessárias, maior será o grau de portabilidade do sistema. Características deste atributo são instalabilidade, adaptabilidade e poder de coexistência.

## 2.3 Arquitetura de Software

Arquitetura de software pode ser definida como a estrutura do sistema ou estruturas de um sistema, que aborda os elementos do software, as propriedades visíveis externamente destes elementos e os relacionamentos entre eles (BASS; CLEMENTS e KAZMAN, 2003). Segundo Valipour et al. (2009) a arquitetura de software está interessada na escolha de elementos arquiteturais, suas interações, e as restrições aplicadas a estes elementos e suas respectivas restrições.

A finalidade de uma arquitetura é identificar os elementos de software e como eles cooperam para a obtenção dos objetivos desejados. Esta especificação visa demonstrar a interação entre os elementos, omitindo detalhes privados e demonstrando a parte pública das interfaces utilizadas para comunicação entre os elementos, ou seja, os mecanismos usados internamente não são relevantes neste momento. As decisões tomadas durante a fase de especificação da arquitetura serão fundamentais para garantir a cobertura dos requisitos do sistema durante a fase de implementação.

A definição de uma arquitetura facilita a compreensão de grandes sistemas pela apresentação em um nível de abstração mais simples para o entendimento, porém deve ser completo o bastante na apresentação de detalhes que possam ser úteis para os diferentes tipos de interessados no sistema, como desenvolvedores, analistas de negócio e analistas de suporte, por exemplo. Como estes envolvidos podem apresentar diferentes necessidades para o entendimento da arquitetura, geralmente a arquitetura é dividida em visões.

Utilizando a definição dada por Bass; Clements e Kazman (2003), uma visão é uma representação de um conjunto coerente de elementos arquiteturais da forma como são descritos e entendidos pelos interessados no sistema.

## 2.4 O modelo de referência RM-ODP

RM-ODP é um padrão de modelagem de arquiteturas de processamento distribuído baseado em objetos que separa preocupações e simplifica a especificação de sistemas heterogêneos de processamento distribuído e aberto (PUTMAN, 2000).

Neste modelo, é possível analisar arquiteturas complexas de sistemas distribuídos através da divisão em visões distintas, cada uma destas podendo refletir um conjunto diferente de características.

Este modelo representa um esforço conjunto entre *International Organization for Standardization* (ISO), o *International Electrotechnical Commission* (IEC) e *Telecommunication Standardization Sector* (ITU-T) na definição de um *framework* para sistemas distribuídos, fornecendo uma base de terminologias e regras para o endereçamento de características funcionais e não funcionais relacionadas às questões de interoperabilidade e comportamental.

Este modelo oferece meios para a representação de necessidades de negócio, arquitetura de processamento distribuído e tecnologias escolhidas de forma consistente e completa. Suas características principais são: independência de sintaxe de representação e linguagem de implementação e separação de representação em visões arquiteturais.

O modelo RM-ODP utiliza cinco níveis de abstrações diferentes, denominados como pontos de vista, que funcionam de forma independente e complementar com o objetivo de facilitar a visualização da arquitetura descrita. As cinco visões são as seguintes:

- **Empresa:** de acordo com Santana et al. (2012), a Visão Empresa deve responder a pergunta “Para que?”, justificando a necessidade do sistema. Para realizar esta justificativa, os seguintes aspectos são levados em consideração: propósito, escopo e políticas/restrições envolvidas;
- **Informação:** a Visão Informação deve oferecer uma perspectiva da semântica das informações contidas no escopo definido e suas estruturas. Além de representar as informações contidas nas outras visões, esta visão deve apresentar as restrições impostas em suas especificações. Este nível de visão deve ser independente de detalhes de distribuição e implementação;
- **Computação:** de acordo com Putman (2000), a visão computacional está focada na decomposição de um sistema em termos de objetos que interagem via interfaces e em restrições aplicáveis às ações dos objetos e interações;

- **Engenharia:** a Visão Engenharia está focada nos mecanismos que suportam as interações entre os objetos especificados nas visões anteriores, ou seja, nesta visão são especificados os mecanismos pelos quais os objetos lógicos devem interagir;
- **Tecnologia:** no último nível do RM-ODP, a visão de tecnologia define como o sistema é implementado. Nesta visão são definidas as configurações de objetos tecnológicos em termos de software e hardware.

## 2.5 Arquitetura Orientada a Serviço

SOA é um estilo arquitetural de software baseado nos princípios de computação distribuída. Este estilo arquitetural vem ganhando atenção nos últimos anos prometendo agilizar processos de negócio (GONZALEZ; PENALVO e GUERRERO, 2009).

Um importante benefício na utilização de SOA é a habilidade de se trabalhar com sistemas complexos de forma flexível, permitindo integração entre sistemas independentemente da linguagem em que foram construídos ou plataforma utilizada. Isto é possível devido à publicação de interfaces separadamente de suas implementações através da utilização de padrões de comunicação comuns. Para os consumidores dos serviços, não é importante como os serviços são implementados, mas como os dados são armazenados ou como estão estruturados (GONZALEZ; PENALVO e GUERRERO, 2009).

Essa arquitetura permite que sistemas separados (provedores e consumidores de serviços) possam funcionar de forma conjunta visando à realização dos processos de negócio empresariais. As redes formadas por estes sistemas não necessitam estar no domínio de um único dono, ou seja, sistemas gerenciados por times, departamentos ou até mesmo companhias diferentes podem ser integrados.

De acordo com Josuttis (2007), os conceitos fundamentais de SOA são três:

- 1) **Serviços:** são representações abstratas do negócio sob o ponto de vista de Tecnologia de Informação. Apesar de serem implementados de forma técnica por TI, oferecem interfaces que possam ser compreendidas pelo pessoal de negócio;
- 2) **Interoperabilidade:** permite a comunicação entre sistemas heterogêneos distribuídos;
- 3) **Baixo acoplamento:** este conceito implica em baixa dependência entre elementos distintos. Baixa dependência permite que sistemas sejam mais flexíveis, contribui para a tolerância a falhas e crescimento dos sistemas (escalabilidade).

### 2.5.1 Barramento de Serviços Corporativos

O Barramento de Serviços Corporativos (ESB - *Enterprise Service Bus*) é a infraestrutura que permite a alta interoperabilidade entre sistemas. Segundo Josuttis (2007), o papel principal do ESB é a permitir a chamada de serviços entre sistemas heterogêneos. Josuttis (2007) ainda completa que as responsabilidades do ESB também incluem transformação de dados, roteamento, segurança e confiabilidade, gerenciamento de serviços, monitoramento e *logging*.

De acordo com Papazoglou e Heuvel (2007) as duas principais funções do ESB são as seguintes: promover baixo acoplamento e quebrar a lógica de integração em pedaços de fácil gerenciamento.

O barramento de serviços é um barramento de mensagens aberto e baseado em padrões, que permite a implementação, implantação e gerenciamento de soluções baseadas em SOA com foco em montar, implantar e gerenciar SOA distribuído (PAPAZOGLU; HEUVEL, 2007).

### 2.5.2 Serviço Web

Nos últimos anos, Serviços web (*Web Service*) estão se tornando a principal tecnologia de processamento distribuído para atingir as promessas da Arquitetura Orientada a Serviços de aumentar o compartilhamento de serviços, reutilização e interoperabilidade (PAPAZOGLU; HEUVEL, 2007). Segundo a definição de Josuttis (2007), serviços web referem-se a uma coleção de padrões que cobrem interoperabilidade.

A utilização desta tecnologia permite que sistemas distribuídos pela internet possam se comunicar através da utilização de padrões abertos. Estes padrões permitem que sistemas distintos se comuniquem independentemente de plataforma e linguagem de programação.

Josuttis (2007) afirma que os cinco principais padrões utilizados por serviços web são:

- 1) *Extensible Markup Language* (XML): formato geral utilizado para a descrição de modelos, formatos e tipos de dados;
- 2) *Hypertext Transfer Protocol* (HTTP): protocolo utilizado para envio de requisições e respostas pela internet;
- 3) *Web Services Description Language* (WSDL): padrão utilizado para a definição de interfaces de serviços web;

- 4) *Simple Object Access Protocol* (SOAP): protocolo utilizado para troca de dados entre serviços web;
- 5) *Universal Description, Discovery and Integration* (UDDI): protocolo utilizado para o gerenciamento de serviços web.

## 2.6 Método de Avaliação de Arquitetura por *Tradeoff* (ATAM)

O *Architecture Tradeoff Analysis Method* (ATAM) é um método para a análise de arquiteturas de software repetível e transacional. A aplicação deste método permite verificar como as decisões arquiteturais tomadas afetam os objetivos de qualidade definidos.

Segundo Bass; Clements e Kazman (2003) existem três principais objetivos na utilização do ATAM:

- Facilitar a avaliação de arquiteturas de sistemas complexos que, geralmente possuem arquiteturas complexas, em um tempo limitado;
- Permite realizar uma conexão entre os objetivos do negócio contemplados pelo sistema e as decisões técnicas tomadas;
- Sistemas complexos possuem geralmente uma grande quantidade de pessoas interessadas e envolvidas. Coletar as perspectivas destes sistemas em um tempo limitado torna-se uma tarefa com certo grau de complexidade para ser gerenciada.

O resultado final da aplicação deste método é a identificação de riscos, não riscos, pontos de sensibilidade e pontos de *tradeoff*.

Riscos representam pontos que podem apresentar efeitos negativos em determinados aspectos de qualidade do sistema, enquanto não riscos, por sua vez, representam pontos seguros na arquitetura.

Pontos de sensibilidade são decisões que impactam diretamente nos atributos de qualidade. Pontos de *tradeoff* são questões que afetam mais de um aspecto da qualidade, ou seja, representam decisões que podem afetar positivamente o alcance de algum atributo de qualidade, porém afetar negativamente outro.

O ATAM consiste na aplicação de nove passos que visam realizar uma análise completa da arquitetura, passando de um nível mais alto, representado pelas necessidades e objetivos do negócio, até questões técnicas de implementação e testes de cenários específicos. Estes passos são apresentados de forma resumida a seguir:

- 1) Apresentação do ATAM – esta etapa consiste na apresentação do método para todos os participantes da avaliação;

- 2) Apresentação dos Direcionadores do Negócio – é apresentada uma visão geral do sistema do ponto de vista do negócio, na qual os principais direcionadores do negócio são demonstrados para toda a equipe;
- 3) Apresentação da Arquitetura – neste passo, o arquiteto ou equipe de arquitetura responsável pelo projeto realizam uma explicação geral da arquitetura utilizando os designs adotados e documentação criada;
- 4) Identificação das Decisões Arquiteturais – os padrões e decisões arquiteturais são listados. Esta lista fica visível para todos os envolvidos na análise com o objetivo de compartilhar o conhecimento sobre como a arquitetura afeta os atributos de qualidade;
- 5) Geração da Árvore de Atributos de Qualidade – os atributos de qualidade mais importantes são revistos e é criada uma árvore com cenários aplicáveis a estes atributos. Os cenários são avaliados e priorizados;
- 6) Análise das Decisões Arquiteturais – os cenários escolhidos na fase 5 são examinados e o arquiteto deve explicar como a arquitetura está preparada para cada um deles. Durante esta etapa, são levantados os riscos, pontos de sensibilidade e *tradeoffs*;
- 7) Priorização de Cenários (*brainstorm*) – são levantados novos cenários com uma equipe maior de envolvidos. Caso os cenários levantados nesta etapa sejam muito diferentes dos levantados no passo 5, isto poderá indicar uma grande diferença na visão do arquiteto e dos demais envolvidos caracterizando assim um risco;
- 8) Análise das Decisões Arquiteturais – os cenários levantados no passo anterior deverão ser analisados pelo arquiteto que, por sua vez, deverá explicar como suas decisões arquiteturais contribuem para cada um;
- 9) Apresentação dos Resultados – nesta última etapa, todas as informações geradas pelo ATAM são capturadas e organizadas em documentos e apresentações. O resultado é mostrado para todos os envolvidos e os riscos identificados são agrupados por temas. Finalmente, os riscos são confrontados com os direcionadores do negócio, fechando assim o ciclo do processo.

### **3 PROPOSTA DE ARQUITETURA DE INTEGRAÇÃO**

Este capítulo apresenta uma proposta de arquitetura de integração baseada nas cinco visões da RM-ODP para a integração de sistemas utilizados no ensino a distância. Para a realização deste trabalho, são considerados os sistemas atualmente utilizados pela instituição estudada no oferecimento de cursos de graduação e os conceitos de arquitetura orientada a serviços são aplicados para a integração dos sistemas.

#### **3.1 Restrições e Características da Proposta**

A arquitetura proposta por este trabalho visa atender a demanda por cursos de graduação de uma instituição pública de ensino a distância, porém a arquitetura proposta pode ser aplicada também em instituições privadas e em cursos de especialização e pós-graduação.

Ao invés de se criar uma solução totalmente nova refazendo as funcionalidades já existentes, este trabalho propõem uma solução que reaproveite os sistemas atualmente em operação, mantendo a implementação existente da lógica de negócio. A proposta realizada por este trabalho aborda a integração dos sistemas existentes na instituição através da criação de uma camada de serviços web, que permita a intercomunicação dos mesmos, através de um novo componente chamado barramento de integração ESB. Esta abordagem permite também que outros sistemas legados ou novos sistemas sejam anexados a esta rede causando o mínimo impacto possível.

Para a mensuração e definição da proposta, são utilizados valores encontrados no cenário atual da instituição. Estes valores são apresentados na definição da visão de empresa do RM-ODP. A proposta foi estruturada com base nas visões Empresa, Informação, Computação, Engenharia e Tecnologia do RM-ODP utilizadas nesta ordem.

#### **3.2 Visões RM-ODP**

Para a criação da arquitetura proposta, conforme mencionado anteriormente, foi aplicada a metodologia RM-ODP, desta forma foram coletadas informações sobre os processos, políticas e requisitos da instituição (Visão Empresa) e a estrutura dos objetos existentes nos sistemas utilizados na prática de ensino a distância (Visão Informação) e, a partir deste mapeamento, são definidos objetos lógicos, interfaces e as regras que se aplicam entre objetos e interfaces (Visão Computação). Também é definida a infraestrutura necessária



para suportar estes objetos lógicos e suas interfaces (Visão Engenharia), além das tecnologias empregadas (Visão Tecnologia).

A seguir, são apresentadas as cinco visões do RM-ODP para a proposta da arquitetura de integração foco deste trabalho.

### 3.2.1 Visão Empresa

A instituição estudada por este trabalho é uma instituição pública de ensino superior e oferece cursos de graduação, especialização e pós-graduação. Esta instituição conta com um núcleo especializado no oferecimento de cursos à distância.

Uma parte significativa dos cursos da modalidade EaD oferecidos são patrocinados por verbas específicas de projetos do governo como, por exemplo, a Rede São Paulo de Formação Docente (Redefor), a Universidade Virtual do Estado de São Paulo (Univesp) e a Universidade Aberta do Brasil (UAB). Existem também cursos oferecidos com verbas da própria instituição para o treinamento de seu corpo de funcionários e cursos oferecidos em conjunto com instituições ou empresas parcerias.

Como projetos com diferentes características são executados, geralmente são envolvidas diversas unidades internas e externas à instituição, ou seja, apesar da produção de conteúdo dos cursos e manutenção dos sistemas utilizados pelos alunos serem responsabilidade de uma unidade específica, a troca de dados cadastrais, notas e frequências dos alunos é muito comum entre unidades, sendo que, esta troca de dados é realizada de forma manual (geralmente na forma de planilhas).

Cada projeto executado respeita uma estrutura organizacional com papéis bem definidos. Os papéis presentes no oferecimento dos cursos são os seguintes:

- **Coordenador de projeto:** responsável geral pelo projeto, realiza o acompanhamento das diversas áreas, profissionais e atividades;
- **Professor especialista:** cada disciplina componente de um curso é acompanhada por um professor especialista. Sua responsabilidade é gerar o conteúdo apresentado na disciplina em questão;
- **Tutor:** representa um papel fundamental no oferecimento de cursos a distância oferecendo suporte aos alunos. Em diversos momentos, funciona como o intermediário entre alunos, professores especialistas e coordenação do projeto;

- **Designer educacional:** organizam os conteúdos oferecidos pelos professores especialistas, auxiliando na geração de matérias de textos, vídeo-aulas e atividades;
- **Web designer:** profissional responsável por disponibilizar os materiais utilizados por cada disciplina no AVA;
- **Designer:** este profissional cuida de toda a identidade visual de cada curso como cadernos, logos, agendas, entre outros;
- **Produtor de vídeo:** responsável pela gravação e edição das vídeo-aulas, oferece suporte aos professores especialistas para a definição de roteiros e conteúdos dos vídeos;
- **Revisor de produção textual:** realiza a revisão e verifica a utilização de normas na criação dos matérias textuais oferecidos nos cursos;
- **Analista de sistemas:** profissionais responsáveis pelos sistemas e serviços oferecidos pelo grupo de tecnologia de informação.
- **Secretária:** responsável por oferecer suporte para demandas por serviços dos alunos, registros, contatos e outras atividades acadêmicas.

Com relação aos *softwares* utilizados, a instituição tem como política a preferência pela utilização de *softwares* livres e sempre são levados em consideração *softwares* com os quais os usuários já tenham familiaridade. Para o oferecimento dos cursos atuais, são envolvidos o sistema de graduação, a plataforma de ensino (ou AVA) e o sistema de controle de matrículas.

Com relação ao perfil dos alunos de EaD, geralmente estes possuem pouca disponibilidade de tempo durante a semana, pois a maioria dos cursistas desta modalidade são alunos entre 18 e 40 anos que estudam e trabalham, segundo o Censo EaD Brasil de 2012 realizado pela Associação Brasileira de Educação a Distância (ABED), o que torna fundamental que o sistema ofereça respostas rápidas, poupando assim um valioso tempo de seus usuários.

A instituição estudada contou com um valor aproximado de sete mil alunos inscritos e ativos simultaneamente em seus cursos durante o ano de 2012, sendo que o período do dia em que os alunos realizam o maior número de acessos está localizado no intervalo entre as dezenove horas e vinte e uma horas. A figura 1 apresenta um gráfico com a distribuição do número de visualizações de página da plataforma de ensino ocorridas no dia 03 de Setembro de 2012.

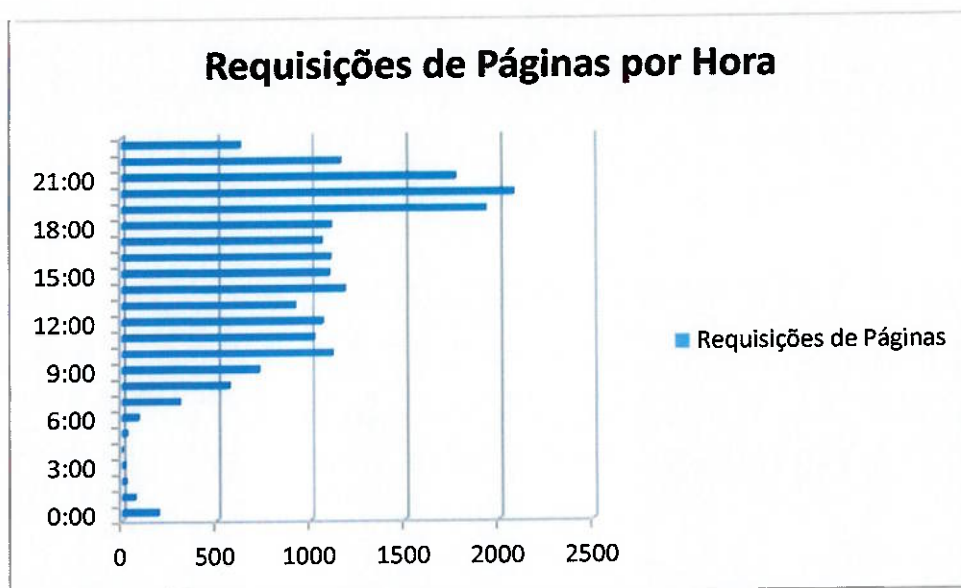


Figura 1- Gráfico de distribuição de requisições de páginas ao longo do dia

O dia representado pela figura 1 foi escolhido por conter o maior número de requisições em uma única hora durante o ano de 2012. No período entre as vinte horas e vinte horas e cinquenta e nove minutos deste dia, foram registradas duas mil e setenta e oito requisições de visualização de página. Outro aspecto demonstrado pela figura 1 é que ocorrem acessos ao sistema durante o dia inteiro, inclusive durante a madrugada.

O ano de 2012 e os anos anteriores apresentaram distribuições do número de requisições parecidas ao longo dos dias.

### 3.2.2 Dificuldades encontradas nos processos atuais

Para demonstração das dificuldades e problemas encontrados nos atuais processos e sistemas, é estudado o caso do processo de matrícula de alunos nos cursos de graduação.

Este processo envolve três setores e três sistemas diferentes da instituição, sendo que a troca de dados e realização dos cadastros é feita de forma manual. A figura 2 demonstra como este processo ocorre.

Conforme é possível se observar na figura 2, os dados dos alunos chegam à seção de graduação primeiramente. A seção de graduação é responsável pelo cadastro do aluno no sistema de matrículas e controle da documentação recebida. Após a confirmação de matrícula, os dados dos alunos são enviados para a secretaria específica do projeto, esta por sua vez se responsabiliza pelo cadastro dos alunos no sistema de graduação. A última etapa consiste no envio dos mesmos dados do aluno da secretaria do projeto para a equipe de TI responsável

pela plataforma de ensino, novamente os dados são checados e inseridos neste sistema de forma manual.

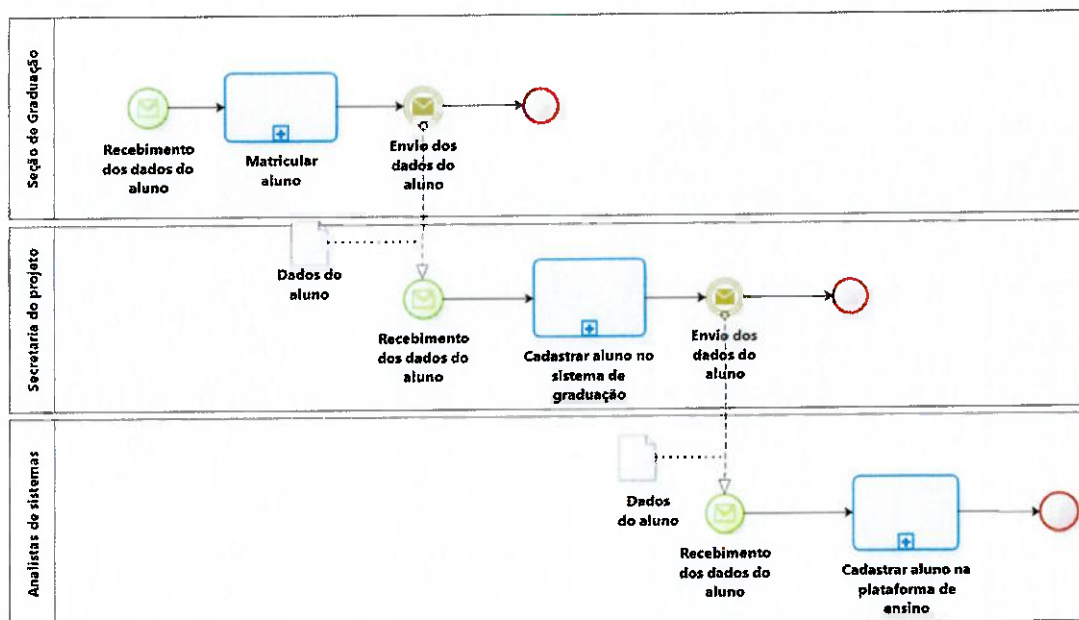


Figura 2- Processo de matrícula atual

A execução deste processo atualmente permite a ocorrência de alguns problemas conhecidos. Os problemas mais comuns são:

- 1) Inconsistências nos dados entre os sistemas, pois sempre existe a possibilidade de erros no momento do cadastro ou atualização apenas em parte dos sistemas;
- 2) Consumo de tempo de diversos funcionários para verificação e inserção dos mesmos dados nos diferentes sistemas.

### 3.2.3 Definição da nova proposta

Este trabalho foi desenvolvido como uma proposta de solução arquitetural baseada nas dificuldades encontradas nos processos atuais de negócio da instituição foco de estudo. Para especificação da arquitetura, são escolhidos três processos do ciclo normal de um curso realizado por seus alunos. A figura 3 demonstra de forma simplificada as principais atividades deste ciclo.

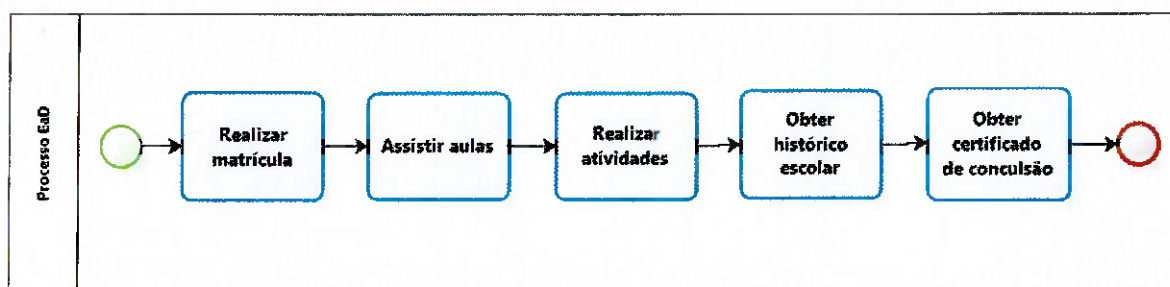


Figura 3 - Fluxo de atividades EaD

Para a definição dos elementos componentes das cinco visões do modelo RM-ODP utilizadas por este trabalho, o processo de matrícula, acessar conteúdo didático das aulas e consulta de histórico escolar foram selecionados por apresentarem aspectos e restrições importantes para a solução final.

O processo de matrícula é vital ao fluxo, pois é neste momento que os alunos são cadastrados nos diversos sistemas utilizados para a realização do curso. Em caso de ocorrência de falhas neste processo, os alunos poderão ser impactados em todos os processos subsequentes da execução do curso.

O processo de obtenção de conteúdo didático das aulas permite que os alunos tenham acesso às vídeo-aulas, material de leitura, imagens e outros materiais que possuem o conteúdo apresentado por cada disciplina do curso.

Já o processo de consulta de histórico escolar foi selecionado por envolver dados contidos em diferentes sistemas da instituição, sendo que são detectadas inconsistências em muitos casos entre estes valores em estágios avançados do curso.

Para representação dos objetos e restrições da Visão Empresa para os processos escolhidos, as seguintes definições são utilizadas:

- 1) Comunidades: no escopo desta arquitetura, foram identificadas duas grandes divisões de comunidades: comunidades internas ou núcleo de educação a distância e comunidades externas ao núcleo. As comunidades internas são compostas por comunidade de produção de materiais (formada por *designers*, *web designers*, revisores textuais, professores especialistas, produtores de vídeo-aulas e *designers* educacionais), comunidade de administração de cursos (formada pela coordenação e secretaria) e a comunidade suporte (formada pelos tutores e analistas de sistemas);
- 2) Papéis: os papéis destacados no processo de execução de cursos são coordenador, professor especialista, tutor, *designer* educacional, *designer*, *web designer*, revisor textual, produtor de vídeo-aula, secretaria, analistas de sistemas e aluno. Para este

trabalho, são focados os papéis do aluno e secretaria ( este segundo representado como seção de graduação para cursos de graduação);

- 3) Objetivo do sistema: dar suporte ao oferecimento de cursos de graduação na modalidade EaD através da utilização de conteúdos gerados por professores e *designers* educacionais qualificados, garantindo a conformidade com as políticas definidas pela instituição;
- 4) Políticas e restrições: para a definição de políticas e restrições, serão utilizadas as principais restrições de qualidade que o sistema deve atender. De acordo com as dificuldades levantadas anteriormente neste trabalho e as características do negócio, quatro principais requisitos referentes a atributos de qualidade foram levantados:
  - I. O sistema deve atender a um total de sete mil alunos ativos e suportar a demanda de duas mil e cem requisições em um intervalo de uma hora durante o horário de pico;
  - II. O sistema deve prover interoperabilidade entre os diversos subsistemas utilizados, diminuindo assim problemas causados por falhas de comunicação ou erros em procedimentos manuais redundantes e reduzir custos através da diminuição da quantidade de retrabalhos efetuados;
  - III. Alta disponibilidade na oferta de serviços (24 horas por dia e 7 dias por semana) permitindo que alunos e outros usuários possam acessar as funcionalidades do sistema de acordo com suas agendas;
  - IV. Para garantir a satisfação dos alunos e demais usuários do sistema, os serviços de consulta ao conteúdo didático, que representam o maior número de requisições durante o período entre o início e fim de cada curso, devem apresentar um tempo de resposta médio inferior a 4 segundos.

### 3.2.4 Visão Informação

A Visão Informação deve oferecer uma perspectiva da semântica das informações contidas no escopo definido e suas estruturas. Além de representar as informações contidas nas outras visões, esta visão deve apresentar as restrições impostas em suas especificações. Este nível de visão deve ser independente de detalhes de distribuição e implementação.

Para a definição desta arquitetura, foi definido um modelo de dados único utilizado para promover a intercomunicação entre os componentes da solução. Segundo Findling (2014), a

definição de um padrão único em arquiteturas orientadas a serviço provê formas mais limpas de acesso aos dados desejados e permite reutilizá-los, substituindo assim a utilização de diversos padrões proprietários de interfaces resultando em redes complexas que afetam a agilidade de alterações sistêmicas para demandas de negócio.

Seguindo-se esta linha de raciocínio, foram utilizados e customizados conceitos apresentados pelo padrão OSID (*Open Service Interface Definitions*) que visa promover a interoperabilidade entre sistemas de arquitetura orientada a serviços. Este padrão foi desenvolvido pelo OKI (*Open Knowledge Initiative*) criado à partir de uma parceria entre o Instituto de Tecnologia de Massachusetts com a Fundação Andrew W. Mellon para o desenvolvimento de sistemas educacionais em instituições de ensino superior e oferece os conceitos necessários para a definição do modelo de dados desejado para esta proposta.

### 3.2.4.1 Processo Matrícula

Antes da representação dos objetos do sistema, um aspecto de fundamental importância é a definição da hierarquia de cursos, disciplinas e turmas e como a mesma funciona dentro da nova solução. Para realização desta definição, foram utilizados e adaptados os conceitos apresentados por Open Knowledge Initiative (2014).

A figura 4 é a representação de uma estrutura genérica utilizada para demonstração da hierarquia supracitada.

De acordo com a exibição da figura 4, existem três classificações distintas: *CursoCanônico*, *OfertaCurso* e *SecaoCurso*. Cada uma destas classificações contém as seguintes características:

- *CursoCanônico*: este elemento proporciona flexibilidade a classificação dos elementos da hierarquia, pois possibilita a criação de elementos de mesmo tipo como “filhos”. Esta classificação está prevista na rotulação de cursos e disciplinas contendo atributos como título, descrição, tipo, número de créditos, pré-requisitos, entre outros;
- *OfertaCurso*: esta classe representa uma oferta de curso ou disciplina “real” pois é neste nível que os alunos são matriculados e as regras para cálculo de notas são definidas;
- *SecaoCurso*: a *SecaoCurso* é uma subdivisão de *OfertaCurso* na qual são definidos os dados como horários das atividades online (síncronas), agenda,

grupos de estudantes, entre outros. Realizando-se um paralelo com o cenário atual, neste nível são definidas as turmas das disciplinas e cursos ofertados.

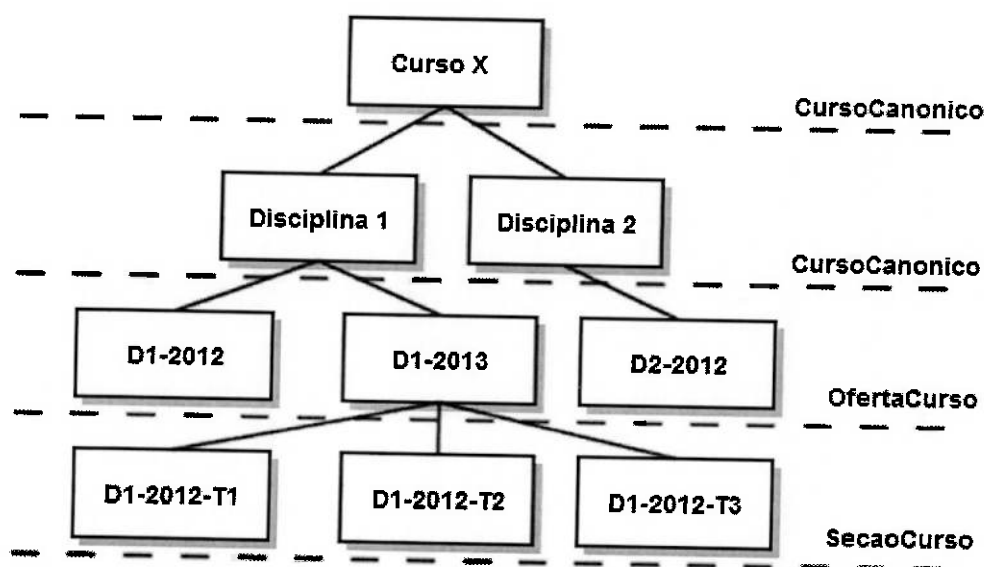


Figura 4 - Visão Informação: hierarquia de cursos, disciplinas e turmas

Após a organização dos cursos, turmas e disciplinas, a próxima etapa apresentada é a organização destes elementos para o processo de matrícula. A figura 5 representa a Visão Informação do processo de matrícula.

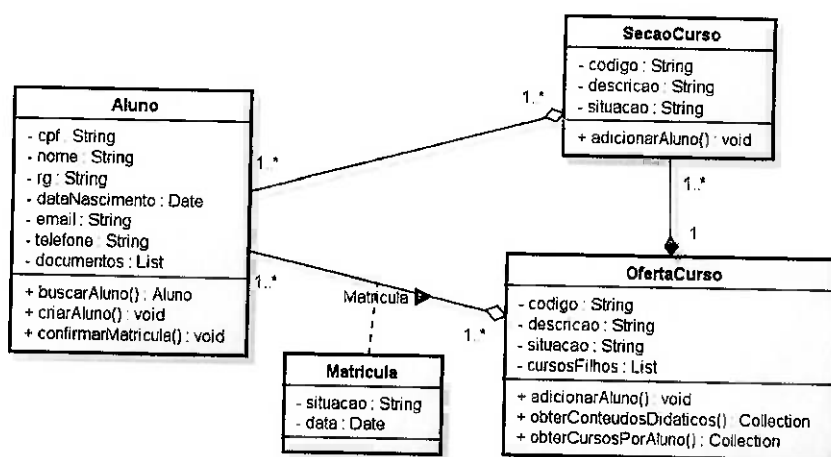


Figura 5 - Visão Informação: processo matrícula

A figura 5 representa os serviços dos objetos de domínio *Aluno*, *SecaoCurso*, *OfertaCurso* e *Matricula*. Sempre que um aluno for adicionado a uma oferta de curso, um



novo registro de matrícula é criado automaticamente pelo sistema. Como regra da instituição, um único aluno pode estar inscrito simultaneamente em diversos cursos.

### 3.2.4.2 Processo Acessar Conteúdo Didático

O processo de acessar conteúdo didático corresponde a um dos processos mais comuns realizados pelo sistema. Neste processo, o aluno autenticado consegue ter acesso aos textos e vídeos disponibilizados para seus cursos e disciplinas. A representação deste processo na Visão Informação está demonstrada pela figura 6.

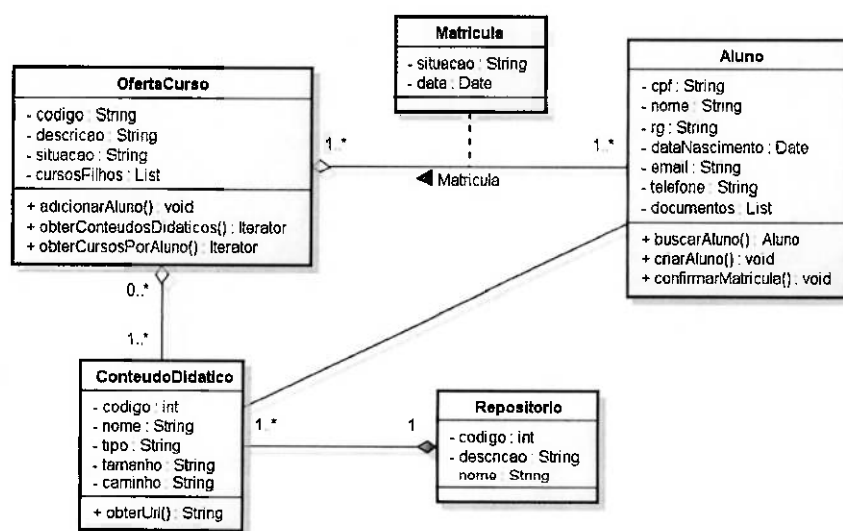


Figura 6 - Visão Informação: processo acessar conteúdo

Na figura 6 constam as representações dos objetos *Aluno*, *Matricula* e *OfertaCurso* utilizados no processo de matrícula. Os objetos *ConteudoDidatico* (representando um conteúdo didático como um arquivo de vídeo, texto para leitura, entre outros) e *Repositorio* (representante de um agrupamento de conteúdos didáticos com relação entre si) foram introduzidos para completar a representação dos objetos utilizados pelo processo de consulta aos conteúdos didáticos.

O elemento *ConteudoDidatico* permite a obtenção dos principais dados referentes ao arquivo digital armazenado como nome, tamanho, tipo de arquivo e endereço URL (*Uniform Resource Locator*).

### 3.2.4.3 Processo Obter Histórico Escolar

O processo de obtenção de histórico escolar fornece aos alunos os dados sobre seu desempenho durante cada disciplina do curso em termos de notas e frequência.

Na instituição estudada, a frequência é medida pelo número de atividades executadas pelo aluno em comparação ao número de atividades oferecidas, fornecendo assim um valor final representado em uma escala entre zero (0) e cem (100) por cento.

Com relação à nota alcançada, para cada disciplina é definida uma fórmula utilizada para o cálculo da média final considerando todas as atividades avaliativas e seus respectivos valores.

A figura 7 demonstra os objetos do sistema utilizados para a representação de notas e frequências finais totalizadas por disciplina oferecida.

Na figura 7, os objetos *RegistroNotaCurso* e *RegistroFrequenciaCurso* são os responsáveis pelo armazenamento dos dados de nota e frequência finais respectivamente para cada *OfertaCurso* na qual o aluno está inscrito. Os elementos *RegistroNotaCurso* e *RegistroFrequenciaCurso* oferecem métodos para que sejam obtidos todos os registros para cada aluno nos cursos desejados.

O elemento *NotaCurso* representa a nota obtida pelo aluno e contém como atributos o tipo de nota (neste caso seria nota final), o valor de nota (valor igual a dez ou a letra "A", por exemplo), a escala de notas (por exemplo, apenas valores entre zero e dez são permitidos) e a definição de pontuação (por exemplo, zero representa a menor nota possível enquanto dez representa a nota máxima).

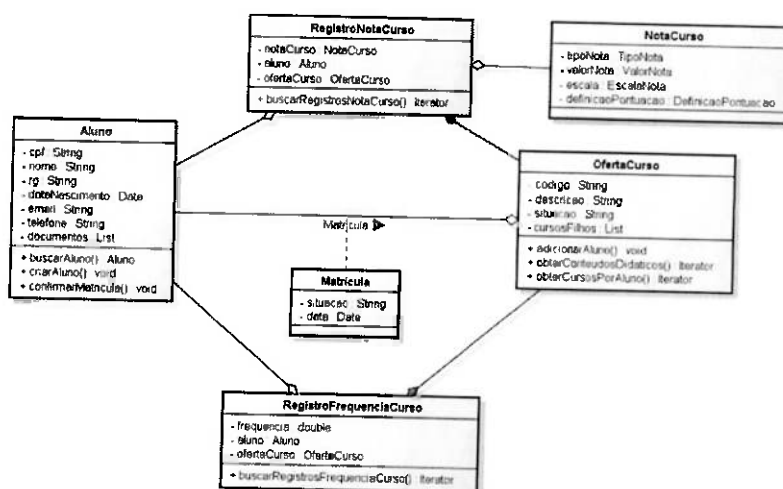


Figura 7 - Visão Informação: processo obter histórico escolar

### 3.2.5 Visão Computação

De acordo com Putman (2000), a visão computacional está focada na decomposição de um sistema em termos de objetos que interagem via interfaces e em restrições aplicáveis às ações dos objetos e interações.

Os três processos escolhidos para o desenvolvimento das visões são abordados adiante.

#### 3.2.5.1 Processo Matrícula

A figura 8 apresenta as atividades de cadastro e matrícula de alunos em seus respectivos cursos e disciplinas na forma de diagrama de sequência UML. Este diagrama contém o papel Seção Graduação e as atividades criar aluno, matricular aluno em disciplina e inscrever aluno em turma. Ao contrário dos diagrama apresentados na Visão Informação, esta representação foca no fluxo de dados entre os elementos da arquitetura.

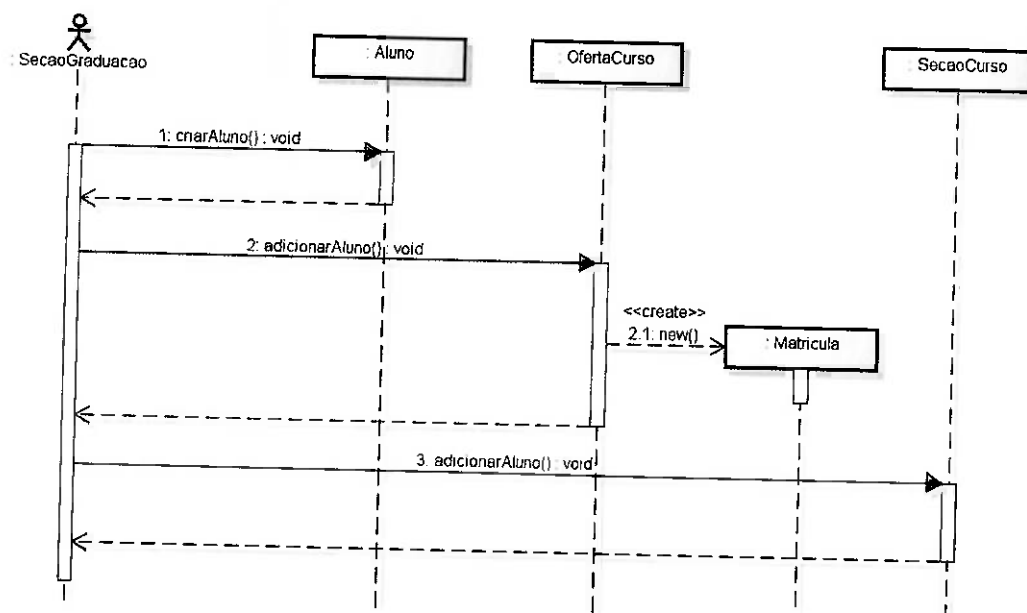


Figura 8 - Visão Computação: processo de matrícula

A figura 8 apresenta as atividades do processo de matrícula realizadas pelo papel da seção de graduação. Neste fluxo, os serviços para criação do aluno, matrícula do aluno em curso/disciplina e inscrição do aluno em uma turma são representados nesta ordem.

A confirmação de matrícula realizada pelo aluno pode ser observada na figura 9.

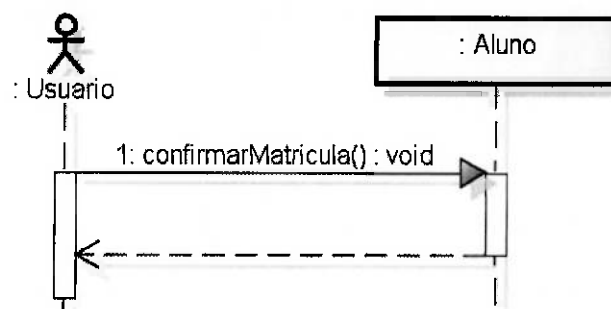


Figura 9 - Visão Computação: confirmação de matrícula

Como última etapa do processo de matrícula, a figura 9 apresenta o usuário, neste caso o aluno, enviando uma requisição de confirmação de matrícula ao sistema.

### 3.2.5.2 Processo Acessar Conteúdo por Curso

Para representação do fluxo de atividades utilizadas no acesso aos conteúdos didáticos, a figura 10 foi criada.

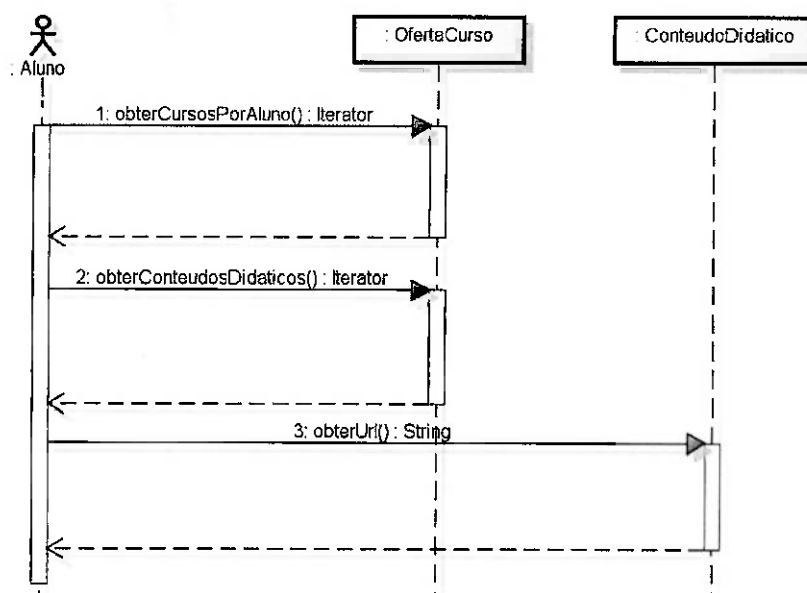


Figura 10 - Visão Computação: processo acessar conteúdo

No diagrama de sequência da figura 10, para ter acesso ao conteúdo, o aluno realiza uma requisição para visualizar todos os cursos na qual está inscrito. Em uma segunda interação, o aluno seleciona um dos cursos e envia uma mensagem solicitando a listagem de

todos os conteúdos didáticos cadastrados para a opção desejada. Por fim, na etapa de número 3, o aluno seleciona um dos materiais listados e requisita a URL para obtenção do mesmo.

### 3.2.5.3 Processo Obter Histórico Escolar

O processo de obtenção de histórico escolar consiste no envio dos registros de notas e frequências obtidas por cada aluno no decorrer do curso. A sequência de requisições deste processo pode ser verificada na figura 11.

Conforme é possível de se observar na figura 11, são necessárias duas requisições para se obter os registros de frequência e notas para cada aluno: *buscarRegistrosNotaCurso()* e *buscarRegistrosFrequenciaCurso()*. Os valores retornados são iterados e combinados para a formação dos valores apresentados no histórico escolar.

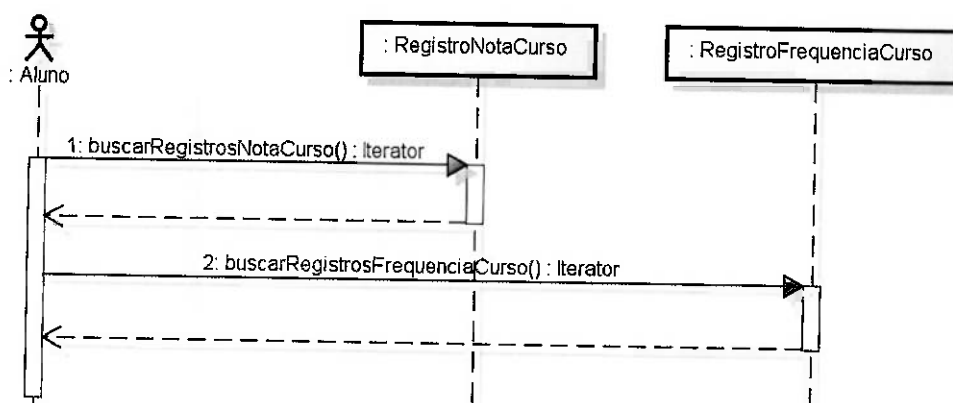


Figura 11 - Visão Computação: processo de obtenção de histórico escolar

### 3.2.6 Visão Engenharia

Já a Visão Engenharia, está focada nos mecanismos que suportam as interações entre os objetos especificados nas visões anteriores, ou seja, nesta visão são especificados os mecanismos pelos quais os objetos lógicos devem interagir.

Na figura 12 é realizada a representação da disposição da infraestrutura da arquitetura. Nesta figura, são demonstrados os sistemas legados representados pelos sistemas de graduação, matrícula e plataforma de ensino, além do barramento de integração e um servidor de cache (este último componente é tratado mais adiante nesta seção).

O barramento de integração é o intermediário entre os consumidores e os provedores de serviços, realizando o papel de *service broker*, ou seja, este é o responsável por

transformações e conversões necessárias nos dados de entrada, além de realizar a conexão entre possíveis protocolos diferentes.

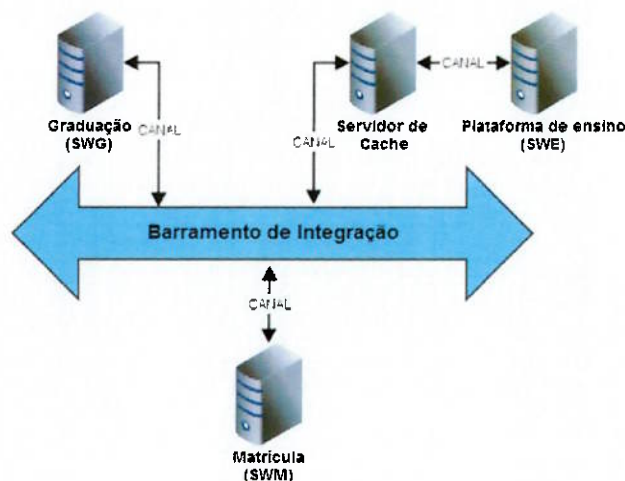


Figura 12 - Visão Engenharia: diagrama de infra-estrutura/integração

Para cada sistema legado existente atualmente, está prevista a criação de uma camada externa responsável pela exposição dos serviços na forma de serviços web. As camadas de exposição dos serviços são representadas na figura 12 pelas siglas SWG - para o sistema de graduação, SWP - para a plataforma de ensino e SWM – para o sistema de matrícula.

Para a representação da organização dos serviços web, dos processos de negócio e dos componentes sistêmicos utilizados, são utilizados diagramas de camadas SOA. Estes diagramas apresentam cinco camadas:

- 1) A primeira camada representa a separação entre a implementação da interface de apresentação acessada pelo usuário e a implementação do serviço;
- 2) A camada de processo de negócio, por sua vez, demonstra a utilização de serviços para a composição de fluxos de processo;
- 3) A terceira camada, chamada de camada de serviços, representa os serviços desenhados para o sistema;
- 4) A camada de componentes representa os componentes contidos nos sistemas donos dos códigos utilizados que suportam a execução dos serviços oferecidos;
- 5) Por último, a quinta camada é a camada operacional representada pelos sistemas legados utilizados.

A figura 13 representa o processo de matrícula utilizando o diagrama de camadas SOA. Esta representação permite a observação da disposição dos serviços de matrícula nos componentes e subsistemas desenvolvidos. Conforme a restrição de interoperabilidade

apresentada Visão Empresa, os serviços web utilizados pelo novo processo de matrícula realizam a manipulação dos dados de alunos nos três sistemas utilizados pela instituição.

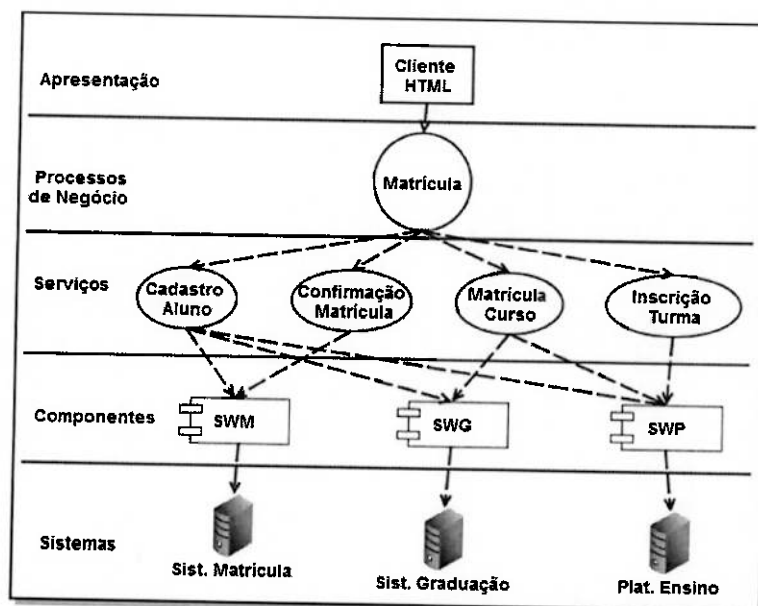


Figura 13 - Visão Engenharia: serviços de matrícula

Seguindo a mesma linha da figura 13, a figura 14 representa o processo de consulta de histórico escolar na Visão Engenharia.

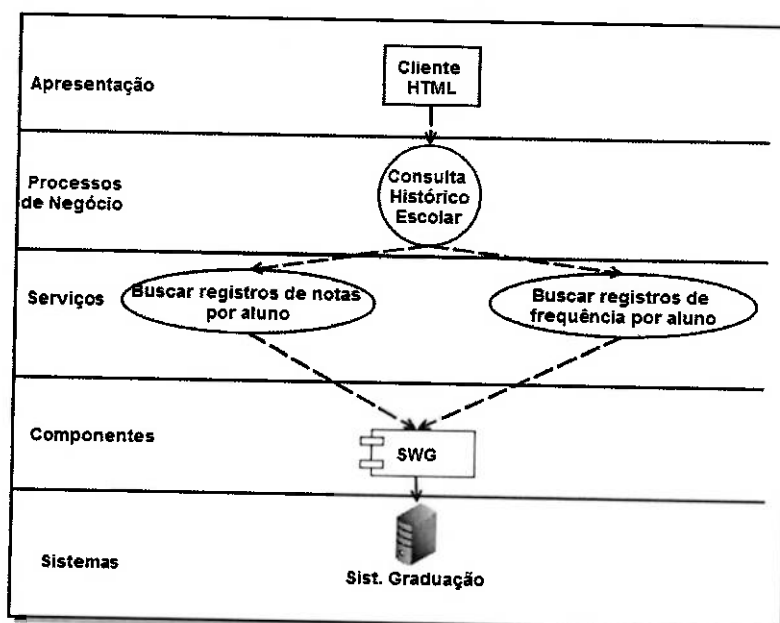


Figura 14 - Visão Engenharia: serviços de consulta de histórico escolar

A figura 14 apresenta os serviços web de busca de registros de nota e frequência de alunos por curso. Os resultados retornados são processados, agrupados e exibidos pelo

sistema responsável pela interface com o usuário. O único sistema utilizado para consulta de notas e frequência é o sistema de graduação, pois apesar destes dados também estarem disponíveis na plataforma de ensino, este é considerado oficialmente o sistema responsável pelo armazenamento destes dados.

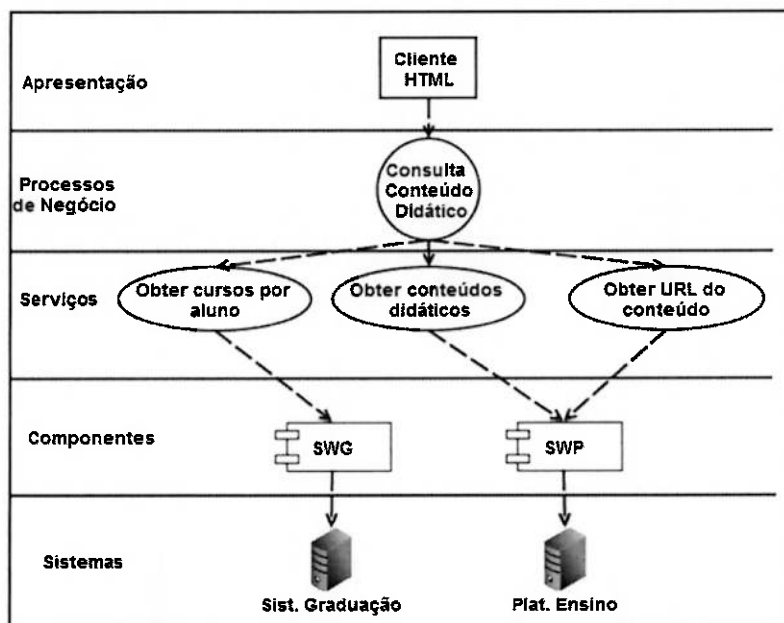


Figura 15 - Visão Engenharia: serviços de consulta de conteúdo didático

O último processo apresentado nesta seção do trabalho é o processo de consulta aos conteúdos didáticos e está representado pela figura 15. Esta figura demonstra a utilização do sistema de graduação e da plataforma de ensino para o suporte ao processo.

Com a utilização das figuras 13, 14 e 15 é possível realizar a conexão entre a infraestrutura apresentada pela figura 12 da Visão Engenharia e os métodos e atividades apresentadas nas Visões Informação e Computação.

Nesta proposta, os requisitos de interoperabilidade são tratados através da definição de um modelo único de dados, utilização de serviços web e da introdução de um *service broker*. Com o objetivo de alcançar os demais requisitos de qualidade indicados na Visão Empresa deste trabalho, abaixo são listadas as táticas arquiteturais empregadas para disponibilidade e desempenho. De acordo com Bass; Clements e Kazman (2003), táticas arquiteturais são decisões de design que permitem a entrega de qualidades necessárias na arquitetura do sistema.



A arquitetura proposta deve atender ao requisito de desempenho levantado na Visão Empresa que determina o tempo médio de resposta menor que 4 segundos para os serviços de consulta de conteúdo.

Desempenho é um atributo de qualidade referente ao tempo que o sistema leva para responder a algum evento (por exemplo: mensagens e requisições de usuários). Segundo Bass; Clements e Kazman (2003), um cenário de desempenho é iniciado quando uma requisição a algum serviço do sistema ocorre, sendo que o sistema utiliza recursos para atender a requisição.

Para que a arquitetura atenda ao requisito de desempenho determinado, a utilização das seguintes táticas arquiteturais está prevista:

- **Bound queue size:** a definição de uma fila de entrada com o tamanho máximo igual ao pico de requisições (valor previsto igual a duas mil e cem requisições) afeta positivamente o desempenho e a disponibilidade do sistema. Esta ação visa evitar a degradação nos serviços do sistema devido à superação do número limite de requisições previsto para os recursos de hardware. Para as requisições excedentes, uma mensagem de "Sistema Ocupado - tente novamente mais tarde" é retornada;
- **Introduce concurrency:** a concorrência está presente nesta proposta com a introdução de clusters nos servidores de aplicação do sistema. O balanceamento de carga entre servidores do barramento é realizado por um servidor dedicado conforme apresentado na figura 13 adiante. Para os servidores dos sistemas legados, o próprio barramento de integração é indicado para a realização da distribuição de carga entre os mesmos;
- **Maintain multiple copies of data:** esta tática é aplicada com a utilização de um servidor de cache distribuído contendo os dados dos conteúdos didáticos conforme apresentado pela figura 12. Como os serviços de consulta aos conteúdos representam a maior parte dos acessos e ocorrem durante todas as fases de execução dos cursos, é fundamental a criação de um cache mantendo estes dados mais acessados em memória. A aplicação desta tática representa uma significativa economia de tempo às respostas de requisições solicitadas ao sistema.

A alta disponibilidade representa outro atributo de qualidade destacado nas restrições do sistema. Uma das definições para disponibilidade é a seguinte: "Disponibilidade está relacionada a falhas no sistema e as consequências associadas. Uma falha no sistema ocorre

quando o sistema para de entregar um serviço de forma consistente com sua especificação” (BASS; CLEMENTS e KAZMAN, 2003). Disponibilidade é geralmente calculada através da seguinte fórmula:

$$\alpha = \frac{MTBF}{(MTBF + MTTR)}$$

Na equação apresentada, Alpha representa o valor da disponibilidade medido em porcentagem, MTBF (Mean Time Between Failure) representa o tempo médio entre falhas do sistema e MTTR (Mean Time to Repair) é o tempo médio para reparo de falhas após sua ocorrência. A partir desta fórmula, é possível verificar que, para se aumentar a disponibilidade, é necessário aumentar o MTBF e/ou diminuir o MTTR.

Com o objetivo de garantir a alta disponibilidade do sistema, esta proposta utiliza as seguintes táticas arquiteturais:

- **Ping/echo:** a monitoração constante dos recursos realizada através do envio de requisições constantes para todos os servidores pertencentes ao sistema ajuda na diminuição do MTTR contribuindo com melhoria na disponibilidade. Nos casos de atraso ou falha em respostas, o sistema deverá enviar alertas para a equipe de suporte para que procedimentos de verificação e reparo possam ser avaliados;
- **Clusters:** a criação de clusters permite que servidores funcionem em conjunto simulando um único servidor. A formação de clusters está prevista para os servidores de aplicação dos sistemas legados e para o barramento de integração, de forma que, em caso de falhas, as requisições do servidor afetado possam ser direcionadas para outros servidores “saúdáveis”, aumentando-se assim o MTBF destes elementos;
- **Heartbeat:** funciona como o batimento cardíaco de um componente, enviando mensagens com intervalos constantes para um elemento monitor. O emprego desta tática consiste no envio de mensagens dos elementos dos clusters para os elementos monitores responsáveis pela distribuição de carga;
- **Checkpoint/Rollback:** a realização de backups incrementais diariamente e um backup completo mensal está prevista para os servidores do sistema. Robôs são os responsáveis por realizar estes backups em fitas. Em caso de falhas graves, esta tática possibilita a recuperação dos componentes do sistema para um estado anterior a falha e consistente;

- **Removal from service:** a desativação de componentes defeituosos ocorre de forma automática em casos de falhas detectadas via *heartbeat*. Como explicado anteriormente, as requisições destinadas ao componente defeituoso são redirecionadas e distribuídas entre os outros elementos do cluster;
- **Intermediate routing:** para os casos de falhas em hardware dos servidores de aplicação dos legados, o barramento de integração funciona como um roteador intermediário capaz de identificar o problema e distribuir a demanda entre os demais servidores.

A figura 16 representa a utilização das principais táticas supracitadas no barramento de integração.



Figura 16 - Visão Engenharia: barramento de integração

O barramento representa o componente mais crítico da proposta e, tratando-se do ponto focal de todas as requisições realizadas ao sistema, pode causar a interrupção de todos os serviços prestados pelo sistema em caso de ocorrência de falhas.

### 3.2.7 Visão Tecnologia

No último nível do RM-ODP, a visão de tecnologia define como o sistema é implementado. Nesta visão são definidas as configurações de objetos tecnológicos em termos de software e hardware.

Um dos pontos mais importantes da arquitetura é a utilização de um barramento de serviços corporativos (ESB) para a implementação do barramento de integração. O mercado

atualmente conta com diversas possibilidades de software para o ESB que satisfaçam às necessidades apontadas nas visões anteriores. As principais opções difundidas no mercado são: Mule ESB, JBoss ESB, WSO2 ESB e Oracle Service Bus. Para a realização deste trabalho, está sendo sugerida a adoção do Mule ESB por se tratar de uma ferramenta leve e de código aberto, além de atender aos demais requisitos especificados.

A principal forma de implementação dos serviços está definida na forma de serviços web através da disponibilização de suas interfaces em WSDLs. Para esta forma de comunicação, foi selecionado o padrão SOAP em cima do protocolo HTTP. Esta escolha foi realizada considerando a restrição de interoperabilidade definida na Visão Empresa. Com este conjunto de características, é possível que os serviços sejam desenvolvidos e executados independentemente das linguagens de programação e plataformas utilizadas, tornando possível a reutilização dos sistemas legados através da criação de uma camada externa que expõem os serviços oferecidos pelos sistemas na forma de serviços web, integrando-os ao ESB.

Com relação aos sistemas operacionais utilizados pelos nós do sistema, os servidores mantidos pela instituição utilizam como padrão o sistema operacional Debian do Linux, podendo, desta forma, ser mantido este padrão. Apenas em casos de existência de restrições específicas para os nós, também é possível a utilização de servidores Windows.

Para os sistemas de graduação e controle de matrícula, os softwares utilizados são soluções caseiras moldadas de acordo com as necessidades da instituição estudada. A criação e utilização de uma camada externa de exposição de serviços web permite que sejam adotadas soluções próprias ou de terceiros para estes sistemas.

Para o papel da plataforma de ensino, existem opções de software-livre amplamente utilizadas como o Moodle, Teleduc e Sakai. A sugestão deste trabalho é a utilização do Moodle com uma base de dados MySQL. Esta combinação está sendo escolhida por se tratar da opção realizada atualmente em produção pela instituição estudada.

Por fim, o servidor de cache distribuído, utilizado para respostas das requisições mais frequentes de acesso ao conteúdo didático, pode ser implementado utilizando os softwares Memcached, EHCACHE ou Oracle Coherence. Para a proposta atual, é sugerida a utilização do Memcache por se tratar de um software gratuito e de código aberto, além de possuir bibliotecas em diversas linguagens para a realização do acesso.

## **4 Verificação e Avaliação do Resultado**

Para validação da arquitetura proposta, são utilizadas as etapas de dois a seis do ATAM. Apesar do método ATAM contemplar nove etapas ao todo, esta avaliação não está sendo realizada com uma equipe conforme previsto pelo método e, neste caso, as etapas sete e oito, que consistem na realização de um *brainstorm* com os envolvidos no projeto e uma nova priorização e análise dos cenários, não são aplicáveis ao trabalho atual.

A avaliação é iniciada pela etapa de número um logo abaixo:

### **4.1 Passo 1 – Apresentação do ATAM**

Esta etapa corresponde à apresentação do método de avaliação arquitetural. Para esta apresentação, será considerado o conteúdo apresentado na seção seis do segundo capítulo deste trabalho.

### **4.2 Passo 2 – Apresentação dos objetivos do negócio**

Para o caso da arquitetura proposta, estes objetivos foram levantados na Visão Empresa e representam os motivadores para a execução deste trabalho. Os requisitos de qualidade fundamentais para esta arquitetura são três: interoperabilidade, disponibilidade e desempenho.

A arquitetura proposta foi desenvolvida com o objetivo de integrar os diversos sistemas utilizados nos processos de negócio de ensino à distância realizados pela instituição foco do estudo. Esta integração é fundamental de forma que a comunicação entre departamentos e sistemas seja realizada de maneira confiável, evitando assim inconsistência em dados comuns e retrabalhos executados por diversos papéis dentro da instituição.

Outro aspecto fundamental já citado é a alta disponibilidade do sistema, pois uma das vantagens de cursos de ensino a distância com relação aos presenciais é a utilização de atividades assíncronas. Este tipo de atividade permite que os alunos as executem conforme sua programação e tempo livre durante a semana, sendo desta forma fundamental que o sistema apresente alta disponibilidade permitindo que os usuários possam acessar os serviços oferecidos a qualquer hora do dia e qualquer dia da semana.

O último aspecto considerado um dos direcionadores deste trabalho é o desempenho do sistema. Conforme mencionado no cenário atual encontrado, alunos de EaD geralmente

possuem pouca disponibilidade de tempo livre, o que torna fundamental que o sistema ofereça respostas rápidas e economize o tempo do usuário.

### **4.3 Passo 3 – Apresentação da arquitetura**

A apresentação da arquitetura foi elaborada utilizando as cinco visões do RM-ODP conforme observado nas seções anteriores deste capítulo. Para compreensão do funcionamento da arquitetura, as Visões Informação, Computação, Engenharia e Tecnologia podem ser adotadas.

### **4.4 Passo 4 – Identificação das decisões arquiteturais**

De acordo com Pontes (2012), as decisões arquiteturais são importantes, pois através delas é possível entender como a arquitetura contempla os atributos de qualidade prioritários.

Durante a definição das cinco visões do RM-ODP foram tomadas decisões que afetam a arquitetura do sistema, desde o nível de negócio até o nível tecnológico, desta forma, este quarto passo corresponde ao capítulo 3 deste trabalho.

### **4.5 Passo 5 – Geração da árvore de utilidades**

A etapa cinco consiste na geração da árvore de atributos de qualidade. Utilizando os objetivos do negócio e levando em consideração o barramento de integração como ponto crítico da arquitetura, a quadro 1 foi criado para representação da árvore de atributos.

Os principais aspectos da árvore de atributos exibida pelo quadro 1 são os atributos de qualidade de maior importância para o sistema, a descrição dos cenários de avaliação e a prioridade dos cenários.

Para a definição da prioridade, foram atribuídas classificações em alta (A), média (M) e baixa (B) dos pontos de vista de negócio e da complexidade de contemplação na arquitetura. Os cenários que foram marcados com dois valores A, ou seja, com importância para o negócio alta e complexidade alta, são analisados na próxima etapa do processo de avaliação.

Quadro 1- Árvore de utilidades do ATAM

Atributo de Qualidade	Refinamento de Atributo	Identificador cenário	Descrição do cenário	Prioridade/Complexidade
1. Interoperabilidade		C1	O sistema de matrícula envia dados para cadastro de um novo aluno na plataforma de ensino, sendo que os sistemas já se tem uma comunicação prévia, a requisição deverá ser aceita e o cadastro realizado com sucesso.	A,A
2. Disponibilidade		C2	Um dos servidores do barramento fica indisponível com o sistema em modo de operação normal e o sistema deverá continuar a operar em modo normal.	A,A
		C3	Ocorreu uma falha de hardware em um dos servidores de aplicação dos subsistemas da arquitetura. O sistema deverá continuar a operação sem interrupções.	A,M
		C4	O link de dados para acesso ao barramento de integração cai quando o sistema está operando em modo normal e o sistema deverá manter a operação.	A,A
		C5	A fonte principal de alimentação de energia dos servidores do barramento fica indisponível e o sistema deverá manter a operação.	A,B
		C6	O sistema de graduação começa a oferecer respostas atrasadas (com mais de 4 segundos de intervalo) quando o sistema está em modo de operação normal e a equipe de suporte deverá ser notificada.	A,M
3. Desempenho	Tempo de resposta de transação	C7	Um aluno realiza consulta de seus dados de perfil durante o horário de pico e obtém resposta em menos de 4 segundos.	A,A
	Vazão	C8	O sistema deverá completar duas transações por segundo em horário de pico.	A,B

#### 4.6 Passo 6 – Análise das decisões arquiteturais

A próxima etapa no ATAM representa a análise das decisões arquiteturais baseada nos cenários mais importantes levantados pela fase anterior. É neste momento que o arquiteto apresenta as soluções arquiteturais tomadas para atender aos cenários mais importantes do sistema catalogando os riscos, não riscos, pontos sensíveis e *tradeoffs*.

Para apresentação da avaliação de cada um dos cenários escolhidos, foi utilizado um modelo adaptado do modelo utilizado por Bass; Clements e Kazman (2003).

Os quadros 2, 3, 4 e 5 demonstram as características dos cenários de disponibilidade, interoperabilidade e desempenho escolhidos na etapa anterior da avaliação arquitetural. Nestes quadros são representadas as identificações e descrições dos cenários, atributos alvo, estado do ambiente, estímulos e respostas esperadas. Além destes fatores, também são apresentadas as táticas arquiteturais escolhidas que os afetam descrevendo os riscos, não riscos, pontos de sensibilidade e *tradeoffs* realizados.

Quadro 2 - Análise Arquitetural – Cenário C1

Identificação Cenário: C1	Descrição Cenário: Comunicação entre sistemas			
Atributo(s)	Interoperabilidade			
Estado do Ambiente	Comunicação definida na fase de <i>design</i>			
Estímulo	Sistema de matrícula envia dados para cadastro de aluno na plataforma de ensino			
Resposta	Requisição aceita e processada com sucesso			
Decisões arquiteturais	Sensibilidade	<i>Tradeoff</i>	Risco	Não risco
Especificação de um modelo de dados		T1		
Criação de serviços web				
Implementação de um <i>service broker</i>		T2		
<b>Argumentação:</b> T1 = a comunicação via serviços web demanda mais recursos do sistema, pois os dados são transformados em XML antes do envio. T2 = as atividades executadas pelo <i>service broker</i> implicam em aumento de demanda por processamento, afetando negativamente o desempenho.				

O quadro 3 apresenta a análise arquitetural do cenário C1, o qual está focado na interoperabilidade entre os sistemas de matrícula e plataforma de ensino. O cenário prevê que a comunicação foi definida na fase de *design*, ou seja, durante a construção da camada de



serviços, foram definidos os dados necessários para a efetivação da comunicação. Neste quadro são identificadas as duas decisões arquiteturais para interoperabilidade descritas na Visão Engenharia: especificação de um modelo de dados, a utilização de serviços web e de um *service broker*. Na avaliação deste cenário foi identificado que a implementação de um *service broker* terá um impacto negativo sobre o desempenho, já que as transformações realizadas pelo barramento demandam mais recursos do sistema.

O quadro 3 apresenta a análise arquitetural do cenário C2, que está focada na disponibilidade do sistema. O cenário avalia uma falha de hardware em um dos servidores do barramento de integração ocorrida durante a operação normal do sistema, sendo que o sistema não pode interromper os serviços oferecidos. Neste quadro são identificadas as seguintes decisões arquiteturais para disponibilidade descritas na Visão Engenharia: *heartbeat*, *ping/echo*, *removal from service* e *intermediate routing*.

Ainda referente a análise do cenário C2, as táticas *heartbeat* e *intermediate routing* afetam negativamente o desempenho do sistema.

Quadro 3 - Análise Arquitetural - Cenário C2

Identificação Cenário: C2	Descrição Cenário: Detecção e recuperação de uma falha de hardware em um servidor do barramento			
Atributo(s)	Disponibilidade			
Estado do Ambiente	Operação normal			
Estímulo	Crash em um dos servidores do barramento de integração			
Resposta	Continuação das operações normalmente			
Decisões arquiteturais	Sensibilidade	Tradeoff	Risco	Não risco
<i>Heartbeat</i>		T1		N1
<i>Ping/echo</i>				
<i>Active Redundancy</i>				N1
<i>Removal from service</i>				
<i>Intermediate Routing</i>	S1	T2		
<b>Argumentação:</b> <b>T1</b> = o envio constante de informações sobre o estado do servidor irá causar uma degradação no desempenho. <b>N1</b> = a utilização de redundância de servidores e <i>heartbeat</i> para monitoração dos funcionamentos dos mesmos aumentam significativamente a disponibilidade do barramento. <b>T2</b> = ESB é um intermediário na comunicação entre dois ou mais servidores e sua utilização produz um overhead de comunicação afeta desempenho. <b>S1</b> = o ESB representa mais um ponto possível de falha.				

O cenário C4 é avaliado no quadro 4. Este cenário verifica o caso de queda na conexão de dados que liga a rede da instituição à internet. Atualmente, a instituição não conta com um serviço redundante para a troca de dados, o que representa um risco a disponibilidade do sistema.

Quadro 4 - Análise Arquitetural - Cenário C4

<b>Identificação Cenário: C4</b>	<b>Descrição Cenário: Queda no link de comunicação com o barramento de integração</b>
<b>Atributo(s)</b>	Disponibilidade
<b>Estado do Ambiente</b>	Operação normal
<b>Estímulo</b>	Queda no link de comunicação com do barramento de integração
<b>Resposta</b>	Requisição aceita e processada com sucesso
<b>Decisões arquiteturais:</b> <i>Ping/echo</i>	
<b>Identificação de riscos, não riscos, <i>tradeoffs</i> e pontos de sensibilidade:</b>  <b>R1</b> = hoje a instituição não conta com um link redundante para comunicação de dados.	

A análise do cenário C7 é representada pelo quadro 5. Neste último quadro de análise, é realizada a avaliação de um cenário de desempenho no qual é verificado se o tempo de resposta é inferior a quatro segundos para o serviço de listagem de materiais didáticos.

O quadro 5 apresenta a introdução de concorrência e a utilização de um servidor de *cache* específico para os serviços mais acessados, neste caso os serviços de acesso aos conteúdos didáticos, irá permitir uma queda no tempo de resposta médio.

Quadro 5 - Análise Arquitetural - Cenário C7

Identificação Cenário: C7	Descrição Cenário: Resposta para consulta de dados do perfil em menos de 4 segundos.			
Atributo(s)	Desempenho			
Ambiente	Horário de pico			
Estimulo	Um aluno requisita o serviço de listagem de conteúdos didáticos para uma disciplina na qual está matriculado			
Resposta	Resposta deve ser retornada em menos de 4 segundos			
Decisões arquiteturais	Sensibilidade	Tradeoff	Risco	Não risco
Bound queue size	S1			
Introduce concurrency				N1
Maintain multiple copies of data				N2
<b>Identificação de riscos, não riscos, tradeoffs e pontos de sensibilidade:</b> <b>S1</b> = caso a demanda ultrapasse o máximo previsto para o tamanho da fila, a disponibilidade do sistema será afetada para os demais usuários. <b>N1</b> = a introdução desta tática permite uma melhora significativa no desempenho com a adição de múltiplos servidores. <b>N2</b> = o <i>cache</i> criado para acesso aos dados da plataforma permitirá que os principais dados acessados sejam retornados com uma velocidade superior ao tempo de resposta de acesso aos discos rígidos.				

#### 4.7 Passo 9 – Consolidar Resultados

A última etapa do ATAM consiste na organização e apresentação dos resultados obtidos durante toda a avaliação. Esta consolidação é realizada com a documentação de todos os riscos, não riscos, pontos de sensibilidade e *tradeoffs* identificados durante o processo do método.

Para a solução proposta por este trabalho, os seguintes pontos foram identificados:

- **Riscos:** Devido a não utilização de uma rede de contingência, a disponibilidade do sistema está vulnerável a uma possível queda no link de dados. Como a instituição conta apenas com um fornecedor para este serviço atualmente, este risco será registrado e monitorado até que seja possível a aquisição de mais um link;
- **Não riscos:** A utilização de um ESB para integração dos subsistemas permite que os mesmos possam comunicar-se e compartilhar dados de forma segura e efetiva. Com a utilização de *clusters*, questões de disponibilidade e desempenho estão tratadas para os servidores de aplicação.
- **Pontos de sensibilidade:** A introdução de mais componentes introduz mais pontos possíveis de falha no sistema, porém as medidas tomadas para a

manutenção de disponibilidade tratam esta questão e devem ser garantidas. Para o caso da fila de entrada de requisições do sistema, os picos de demanda devem ser monitorados e, caso ocorram superações do limite com frequência, os *hardwares* do sistema e o tamanho da fila devem ser revistos.

- **Tradeoffs:** Para se alcançar uma melhor interoperabilidade, o desempenho foi afetado com a aplicação de táticas arquiteturais para interoperabilidade. Apenas considerando a organização da arquitetura não é possível garantir o tempo de resposta abaixo dos quatro segundos desejados. É interessante a criação de um protótipo para a realização de testes e averiguação do alcance deste quesito.

## **5 CONSIDERAÇÕES FINAIS**

### **5.1 Objetivo Atingido e Contribuições**

A realização deste trabalho possibilitou a aplicação de diversos tópicos de Tecnologia de Informação na criação de uma arquitetura de sistema visando o atendimento de necessidades reais de uma instituição pública de ensino à distância.

A utilização do modelo RM-ODP, táticas e padrões arquiteturais de software, e do método ATAM demonstraram a viabilidade de aplicação da arquitetura proposta em um caso de estudo real.

Durante o desenvolvimento deste trabalho, as cinco visões do modelo RM-ODP serviram como direcionadoras para a organização e estruturação da proposta, seguindo o caminho desde a identificação das necessidades da área de negócio até a solução técnica final.

A escolha pela utilização de uma arquitetura orientada a serviços permite a reutilização dos sistemas legados já existentes com a implementação de uma camada de interoperabilidade e integração ao ESB, desta forma, ocorreria economia de tempo de desenvolvimento e seriam minimizados os impactos para os diversos departamentos envolvidos e usuários atuais. Esta abordagem também permite flexibilidade para a integração de novos sistemas e disponibilização de novos serviços.

Por fim, o método de análise arquitetural ATAM permitiu a avaliação da solução proposta com a identificação dos principais pontos de risco e pontos seguros da solução relacionados ao atendimento dos atributos de qualidade. Esta abordagem permitiu a verificação da viabilidade da proposta antes mesmo do início da construção e implementação da mesma.

### **5.2 Dificuldades Encontradas**

Durante o desenvolvimento deste trabalho, diversas dúvidas surgiram sobre a representação dos processos de negócio nas diversas visões do RM-ODP. Foram desenvolvidas diversas formas de representações para estes processos até a criação das versões presentes nesta monografia.

Também apareceram dúvidas sobre a utilização de componentes comuns em arquiteturas orientadas a serviços e a aplicação de outros padrões além dos citados neste

trabalho. Estas últimas dúvidas foram sanadas após a troca de experiências com o orientador, o coorientador e outros colegas.

### 5.3 Trabalhos Futuros

Como possíveis trabalhos futuros, as seguintes sugestões são levantadas:

- O foco deste trabalho foi aplicado a uma arquitetura que atendesse aos três principais direcionadores de negócio (interoperabilidade, disponibilidade e desempenho) descritos na Visão Empresa da solução, porém outros aspectos importantes de qualidade como segurança, modificabilidade e usabilidade não foram abordados. Estes aspectos não representados poderiam representar os desafios de possíveis trabalhos futuros;
- Conforme definido por Gerhardt (2005), a nova geração de ensino à distância consiste na prática desta modalidade via dispositivos com acesso a redes wireless como celulares, *tablets* e outros dispositivos móveis, desta forma, seria possível o estudo de serviços e suas restrições limitando-se a estes tipos de dispositivos;
- Para o desenvolvimento deste trabalho, apenas alguns aspectos fundamentais de arquiteturas orientadas a serviços foram aplicados. Outras propostas de arquiteturas poderiam se aprofundar e utilizar estes conceitos de SOA não utilizados pela presente solução.

## 6 LISTA DE REFERÊNCIAS

ANDERSON, T.; DRON, J. Learning technology through three generations of technology enhanced distance education pedagogy. *EURODL - European Journal of Open, Distance and E-Learning*, 2014. Disponível em: <<http://www.eurodl.org/?article=523>>. Acesso em: 22 jan. 2014.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software architecture in practice**. Boston: Addison-Wesley, 2003.

BIANCO, P. et al. Architecting service-oriented systems. **Software Engineering Institute**, 2011. Disponível em: <<http://www.sei.cmu.edu/reports/11tn008.pdf>>. Acesso em: 25 Jan. 2014.

DEHKHARGHANI, R.; YILMAZ, C. Automatically identifying a software product's quality attributes through sentiment analysis of tweets. 2013 In: 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE). San Francisco, CA, USA: [s.n.]. Maio 2013. p. 25-30. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6611717>> Acesso em: 30 Jan. 2014.

FINDLING, M. Why Data Modeling is Now Critical to SOA Success. **Embarcadero Developer Network**. Disponível em: <<http://edn.embarcadero.com/article/40560>>. Acesso em: 05 jan. 2014.

GERHARDT, L. A. The future of distance learning - the process and the product. In: 6th International **Conference** on Information Technology Based Higher Education and Training, 2005. ITHET 2005. [S.l.]: [s.n.]. 2005. p. F1A/1 - F1A/2. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1560265>> Acesso em: 30 Jan. 2014.

GONZALEZ, M. A. C.; PENALVO, F. J. G.; GUERRERO, M. J. C. **Adapting LMS architecture to the SOA: an Architectural Approach**. In: Fourth International **Conference** on Internet and Web Applications and Services, 2009. ICIW '09. [S.l.]: [s.n.]. 2009. p. 322-327. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5072539>> Acesso em: 30 Jan. 2014.

JOSUTTIS, N. M. **SOA in practice: the art of distributed systems design**. [S.l.]: O'Reilly, 2007.

LITWIN, E. **Educação a distância: temas para o debate de uma nova agenda educativa**. Porto Alegre: ArtMed, 2001.

OPEN KNOWLEDGE INITIATIVE. Open Knowledge Initiative (OKI). **sourceforge**. Disponível em:

<[http://ufpr.dl.sourceforge.net/project/okiproject/Doc%20%28current%20OSID%20version%29/OSID%20Documentation/OSID-Full\\_Doc\\_Set-version2.0.0-A4.pdf](http://ufpr.dl.sourceforge.net/project/okiproject/Doc%20%28current%20OSID%20version%29/OSID%20Documentation/OSID-Full_Doc_Set-version2.0.0-A4.pdf)>. Acesso em: 29 Jan. 2014.

PAPAZOGLU, M. P.; HEUVEL, W. J. Service oriented architectures: approaches, technologies and research issues. **The VLDB Journal**, v. 16, n. 3, p. 389-415, Mar. 2007.

PONTES, D. P. N. **Evolução de software baseada em avaliação de arquiteturas**. 2012. 99p. Dissertação (Mestrado) - Departamento de Engenharia de Computação e Sistemas Digitais. Universidade de São Paulo. São Paulo. 2012.

PUTMAN, J. R. **Architecting with RM-ODP**. Upper Saddle River, New Jersey: Prentice Hall, 2000.

SANTANA, F. S. et al. **A complete RM-ODP case-study to integrate geospatial services and ecological niche modeling systems**. 2012 In: IEEE 13th International Conference on Information Reuse and Integration (IRI). Las Vegas, NV, USA: [s.n.]. 8-10 Ago. 2012. p. 239-246. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6303016>> Acesso em: 30 Jan. 2014.

TANAKA, A. et al. **Applying ODP enterprise viewpoint language to hospital information systems**. In: Fifth IEEE International Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Seattle, WA, USA: [s.n.]. 2001. p. 188-192. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=950438>> Acesso em: 30 Jan. 2014.

TORI, R. **Educação sem distância: as tecnologias interativas na redução de distâncias em ensino e aprendizagem**. São Paulo: Senac, 2010.

VALIPOUR, M. H. et al. **A brief survey of software architecture concepts and service oriented architecture**. In: 2nd IEEE International Conference on Computer Science and Information Technology, 2009. ICCSIT 2009. Beijing, China: [s.n.]. Agosto 2009. p. 34-38. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5235004>> Acesso em: 30 Jan. 2014.