



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
Unidade Acadêmica de Educação a Distância e Tecnologia
Programa de Pós-Graduação em Tecnologia e Gestão em
Educação a Distância

ATHOS DENIS EULÁLIO

Modelo de integração entre Ambientes Virtuais de Aprendizagem e
Sistemas de Gestão Acadêmica baseado em Arquitetura Orientada a
Serviços e Computação em nuvem

Recife 2016



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
Unidade Acadêmica de Educação a Distância e Tecnologia
Programa de Pós-Graduação em Tecnologia e Gestão em
Educação a Distância

ATHOS DENIS EULÁLIO

Modelo de integração entre Ambientes Virtuais de Aprendizagem e
Sistemas de Gestão Acadêmica baseado em Arquitetura Orientada a
Serviços e Computação em nuvem

Dissertação apresentada ao Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância como requisito parcial para a obtenção do título de Mestre em Tecnologia e Gestão em Educação a Distância.

Área de Concentração: Ferramentas Tecnológicas para Educação a Distância

Orientador: Prof. Dr. Rodrigo de Souza
DEINFO/UFRPE

Recife 2016

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

E88m Eulálio, Athos Denis
Modelo de integração entre ambientes virtuais de aprendizagem e sistemas de gestão acadêmica baseado em arquitetura orientada a serviços e computação em nuvem / Athos Denis Eulálio. – 2016.
95 f. : il.

Orientador: Rodrigo Nonamor Pereira Mariano de Souza.
Dissertação (Mestrado) – Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância, Recife, BR-PE, 2016.
Inclui referências e apêndice(s).

1. Integração 2. Ambiente virtual de aprendizagem 3. Sistema de gestão acadêmica I. Souza, Rodrigo Nonamor Pereira Mariano de, orient. II. Título

CDD 371.394422

Universidade Federal Rural de Pernambuco
Unidade Acadêmica de Educação a Distância e Tecnologia
Pró-Reitoria de Pesquisa e Pós-Graduação
Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância

Modelo de integração entre Ambientes Virtuais de Aprendizagem e Sistemas de Gestão Acadêmica baseado em Arquitetura Orientada a Serviços e Computação em nuvem

Athos Denis Eulálio

Dissertação julgada adequada para obtenção do título de Mestre em Tecnologia e Gestão em Educação a Distância, defendida e aprovada em 16/12/2016 pela Banca Examinadora.

Orientador:

Prof. Dr. Rodrigo Nonamor Pereira Mariano de Souza
Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância – UFRPE

Profa. Dra. Juliana Regueira Basto Diniz
Membro Interno PPGTEG - UFRPE

Profa. Dra. Sônia Virgínia Alves França
Membro Interno PPGTEG – UFRPE

Prof. Dr. Fernando Antônio Aires Lins
Membro Externo – Programa de Pós-Graduação em Informática Aplicada - UFRPE

AGRADECIMENTOS

Agradeço à minha linda e amada esposa Tatyana pelo companheirismo, suporte e amor incondicional que sempre demonstrou e que me guiou e motivou durante esta jornada;

A minha também linda e amada filha Mariana que me presenteou com a alegria de ser pai e por ter me ensinado a sempre buscar meu aprimoramento como pessoa;

Aos meus pais, irmãos e sobrinho por terem sido sempre referências, apoio de todas as horas e pelas experiências e alegrias vividas;

Ao professor Dr. Rodrigo de Souza, por ter abraçado este projeto e acreditado em minhas ideias. Obrigado pela serenidade e cortesia que sempre pautou nossa trajetória acadêmica e que servirão de exemplo durante minha trajetória profissional;

Ao meu grande amigo Manuel que desde o início acreditou nesse projeto e sempre foi meu grande incentivador e referência de todas as horas durante esta caminhada;

Aos meus amigos de apartamento durante os períodos em Recife (Geraldo, Tiago, Nelson, Markenio e Maykol) por me lembrarem de que a arte da convivência é difícil e por terem sido pacientes com meu mau-humor eventual;

Aos meus amigos de todos os momentos Wilton, Edilberto e Edson que sempre torceram pelo meu sucesso nessa jornada e me apoiaram em tudo;

E todos aqueles que direta ou indiretamente contribuíram para concretização deste trabalho. A todos meu muito obrigado.

EPÍGRAFE

“Agora compreendo que os limites entre o barulho e o som são apenas conceitos e que todos os limites são conceitos esperando serem transcendidos. Podemos transcender qualquer conceito, se acreditarmos que isso é possível. Em momentos assim percebo que minha vida vai muito além das minhas limitações.” (Cloud Atlas).

RESUMO

No âmbito acadêmico, a educação à distância vem cada vez mais ganhando espaço em decorrência do aumento de vagas nesta modalidade de ensino, contudo surge também a necessidade de tornar transparente a comunicação entre os Sistemas de Gestão Acadêmica (SGA) – responsável por gerir todo percurso acadêmico do discente nas instituições de ensino que oferecem essa modalidade - com os chamados Ambientes Virtuais de Aprendizagem (AVA) - sistemas nos quais são realizados todos os processos relativos ao ensino e aprendizagem do discente. Ocorre que, normalmente, estes sistemas não se comunicam entre si, fazendo-se necessário estabelecer uma estratégia de integração com o objetivo de garantir a interoperabilidade entre ambos. Diante desse quadro, com base em pesquisas na literatura, trabalhos realizados nesse campo de pesquisa e levantamento de requisitos comuns para os sistemas envolvidos, é proposto um modelo de alto nível que faça uso de tecnologias *web services* e computação em nuvem que possibilite integrar SGA e AVA. O objetivo deste modelo é fornecer interoperabilidade entre os sistemas envolvidos, sendo que o mesmo funcione de forma descentralizada estabelecendo funcionalidades que melhorem o tempo despendido nos sistemas supramencionados, assim como oferecer elementos para ajudar docentes e membros de equipes multidisciplinares no processo de operacionalização e gerência dos discentes a fim de equacionar a problemática destacada, podendo este modelo ser utilizado por qualquer instituição com demanda semelhante.

Palavras-chave: Interoperabilidade, Ambiente Virtual de Aprendizagem, Sistema de Gestão Acadêmica.

ABSTRACT

In the academic field, distance education is increasingly gaining ground as a result of the increase in vacancies in this type of education, however it is also a need to make communication clear between Academic Management Systems (SGA) - it is responsible for managing all the courses Academic systems of the student in the educational institutions that offer this modality - with they are called Virtual Environments of Learning (AVA) - systems in which all the processes are made all the processes related to the teaching and learning of the student. It occurs, these systems usually do not communicate with one another, making it necessary to establish an integration strategy with the aim of ensuring interoperability between both. In view of this framework, it based on researches in the literature, the works carried out in this research field and survey of common requirements for the systems involved, a high level model is proposed that makes use of web services technologies and cloud computing that allows integrating SGA and AVA. The objective of this model is to provide interoperability among the involved systems, in which it functions in a decentralized way establishing functionalities that improve the spent time in the above mentioned systems, as well as it offers elements to help teachers and members of multidisciplinary teams in the process of operationalization and it managements of students in order to equate the outstanding problem , this model can be used by any institution with similar demand.

Keywords: Interoperability, Learning Management System, Academic Management System

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1: Processo de troca de dados/informações..... | 13 |
| Figura 2: Representação gráfica de um sistema..... | 18 |
| Figura 3: Funcionamento de um Web Service..... | 23 |
| Figura 4: Soluções da Nuvem Computacional..... | 26 |
| Figura 5: AVA Moodle instalado no Ubuntu 14.04..... | 35 |
| Figura 6: Consulta de dados a serem importados do MS SQL Server 2014..... | 36 |
| Figura 7: Cursos, períodos e polos disponíveis para importação..... | 38 |
| Figura 8: Dados importados para o Moodle..... | 39 |
| Figura 9: Disciplinas importadas para o Moodle..... | 39 |
| Figura 10: Importação dos discentes em suas respectivas disciplinas..... | 40 |
| Figura 11: Funcionamento de um Web service..... | 42 |
| Figura 12: Representação de dados em XML..... | 43 |
| Figura 13: Web Services Standards..... | 46 |
| Figura 14: Funcionamento do XML SOAP..... | 46 |
| Figura 15: Solicitação em XML..... | 47 |
| Figura 16: Resposta em XML..... | 47 |
| Figura 17: Funcionamento do REST..... | 49 |
| Figura 18. Visão geral da Arquitetura..... | 60 |
| Figura 19: Diagrama de Casos de Uso da solução..... | 62 |
| Figura 20: Diagrama de Classes..... | 64 |

LISTA DE TABELAS

| | |
|--|----|
| Quadro 1: Características da computação em Nuvem..... | 24 |
| Quadro 2: Abordagens de integração entre sistemas e organizações..... | 32 |
| Quadro 3: Dados a serem importados..... | 34 |
| Quadro 4: Tabelas a serem preenchidos no MySQL e dados a serem importados..... | 36 |
| Quadro 5: Ações a serem realizadas para importação de dados entre os sistemas..... | 37 |
| Quadro 6: Dados a serem importados para AVA..... | 50 |
| Quadro 7: Resumo de padrões..... | 51 |
| Quadro 8: Perfil dos participantes..... | 56 |
| Quadro 9: Serviços ofertados..... | 63 |
| Quadro 10: Tipos de Relação de Recursos e Regras de Mapeamento de URIs..... | 65 |
| Quadro 11: Mapeamento e URIs produzidos..... | 66 |

LISTA DE GRAFICOS

| | |
|-------------------------------------|----|
| Gráfico 1: Avaliação do modelo..... | 75 |
| Gráfico 2: Nível de aceitação..... | 75 |
| Quadro 3: Adoção do modelo..... | 75 |

LISTA DE SIGLAS

AVA - Ambiente Virtual de Aprendizagem
EAD - Educação a Distância
EAI - Enterprise Application Integration
ERP – Enterprise Resources Planning
ETAPI – Escola Técnica Aberta do Brasil
IAAS - Infrastructure as a Service
IES - Instituição de Ensino Superior
IFPI – Instituto de Educação, Ciência e Tecnologia do Estado do Piauí
JSON - JavaScript Object Notation
MOODLE- Modular Object-Oriented Dynamic Learning
PAAS - Platform as a Service
PHP - Hypertext Preprocessor, originalmente Personal Home Page
REST - Representational State Transfer
SAAS - Software as a Service
SGA - Sistema de Gestão Acadêmica
SOA - Arquitetura Orientada a Serviços
SOC – Computação Orientada a Serviços
SOAP - Simple Object Access Protocol
TIC - Tecnologia da Informação e Comunicação
UAB - Universidade Aberta do Brasil
UML - *Unified Modeling Language*
URI – Uniform Resource Identifier
W3C - World Wide Web Consortium
WSDL - Web Service Description Language
XML - Extensible Markup Language

SUMÁRIO

| | |
|--|----|
| 1. INTRODUÇÃO..... | 15 |
| 1.1 Considerações Iniciais e contextualização..... | 15 |
| 1.2. Problema da Pesquisa | 17 |
| 1.4. Objetivos..... | 21 |
| 1.4.1. Objetivo Geral..... | 21 |
| 1.4.2. Objetivo Específicos..... | 21 |
| 1.5. Justificativa | 21 |
| 1.6 Organização | 22 |
| 2. REVISÃO DA LITERATURA E TRABALHOS RELACIONADOS..... | 24 |
| 2.1 Sistemas de Informação | 24 |
| 2.2. Sistema de Gestão Acadêmica – SGA..... | 26 |
| 2.3. Ambiente Virtual de Aprendizagem – AVA..... | 27 |
| 2.4. Web Services e Computação em Nuvem | 28 |
| 2.4.1. Web Services | 29 |
| 2.4.2. Computação em Nuvem..... | 31 |
| 2.5. Trabalhos Relacionados..... | 34 |
| 2.5.1. Carvalho (2010) | 35 |
| 2.5.2. Silva (2012) | 35 |
| 2.5.3. Furtado et al (2015)..... | 35 |
| 2.5.4. Moura e Bernardino (2010)..... | 36 |
| 2.5.5. Furtado (2016)..... | 37 |
| 2.5.6. Bjeljac et al (2015) | 37 |
| 2.5.7. Observações preliminares | 37 |
| 3. ABORDAGENS PARA INTEGRAÇÃO | 39 |
| 3.1. Integração de sistemas orientado a dados..... | 41 |
| 3.2. Integração de sistemas orientado a serviços | 49 |
| 3.2.1 Formato de dados..... | 50 |
| 3.2.1.1 O formato <i>XML</i> | 50 |
| 3.2.1.2 O formato <i>JSON</i> | 51 |
| 3.2.2. Padrões para desenvolvimento de <i>web services</i> | 52 |
| 3.2.2.1. Web services SOAP | 53 |
| 3.2.2.2 Web services REST ou RESTful..... | 57 |

| | |
|--|----|
| 3.2.3. Integração de sistemas no meio acadêmico na modalidade EAD..... | 59 |
| 3.2.4. Comparação de abordagens..... | 60 |
| 4. PROCEDIMENTO METODOLÓGICO..... | 62 |
| 4.1. Levantamento Bibliográfico..... | 62 |
| 4.1.1. Escolha do padrão arquitetural <i>REST</i> | 62 |
| 4.1.2. Escolha do paradigma de computação em nuvem | 63 |
| 4.2 Entrevista semiestruturada | 64 |
| 4.3 Resultados preliminares | 66 |
| 5. ARQUITETURA E MODELAGEM | 68 |
| 5.1. Validação | 76 |
| 5.2. Restrições previstas..... | 78 |
| 6. CONCLUSÕES FINAIS E TRABALHOS FUTUROS..... | 80 |
| 7. PUBLICAÇÕES..... | 83 |
| REFÊNCIAS | 84 |
| APÊNDICE A – QUESTIONÁRIO SEMIESTRUTURADO PARA REALIZAÇÃO DE ENTREVISTA..... | 89 |
| APÊNDICE B – QUESTIONÁRIO SEMIESTRUTURADO PARA LEVANTAMENTO DE REQUISITOS..... | 90 |
| APÊNDICE C – FORMULÁRIO DE VALIDAÇÃO PARA MODELO DE INTEGRAÇÃO ENTRE SGA E AVA. | 94 |

1. INTRODUÇÃO

1.1 Considerações Iniciais e contextualização

A exponencial evolução da modalidade de educação a distância (EAD), impulsionada pela evolução tecnológica capitaneada pelo acesso cada vez maior de banda larga por parte da população brasileira assim como aparatos tecnológicos tais como computadores pessoais, *notebooks*, *tablets*, *e-books*, *smartphones*, etc, fizeram aumentar a oferta de cursos nesta modalidade na forma de programas e cursos de formação iniciada, cursos técnicos de nível médio, cursos superiores de bacharelado e licenciaturas e até mesmo pós-graduações, proporcionando à sociedade mecanismos para buscar qualificação de forma continuada ou permanente.

Filatro (2007) destaca que o avanço tecnológico obtido nos últimos anos fez emergir modalidades de ensino não presenciais em condições de melhor equacionar a diferença entre o número restrito de vagas da rede de ensino e a necessidade de incluir socialmente maior parcela da população. Desta forma a modalidade de educação a distância veio ao encontro da grande necessidade por qualificação profissional, assim como possibilitou minimizar a distorção existente na relação oferta e demanda por vagas na modalidade de ensino presencial.

A modalidade de educação a distância é uma destas modalidades citadas por Filatro, e que segundo os autores Moore e Keasley (2007) caracteriza-se por ser um aprendizado planejado que ocorre normalmente em lugar diferente do local do ensino, exigindo técnicas especiais de criação do curso e de instrução, comunicação por meio de várias tecnologias e disposições organizacionais e administrativas especiais.

A utilização de várias tecnologias da informação e comunicação (TIC), na forma como estas estarão dispostas dentro das organizações/instituições, são um fator determinante para obtenção de vantagem competitiva, o aumento da produtividade e essencialmente para melhoria dos resultados que se espera alcançar.

Para a modalidade de educação a distância praticada por instituições de ensino superior (IES), vislumbra-se a necessidade de que estas forneçam a infraestrutura adequada para uma maior eficiência e eficácia operacional da modalidade a fim de que prestem bons serviços e continuem atuais e de vanguarda, através do uso das TIC's, para facilitar e gerenciar suas atividades acadêmicas.

A partir da escolha do ambiente virtual de aprendizagem (AVA) para modalidade de educação a distância, deve-se analisar todos os aspectos estratégicos envolvidos com o objetivo

de abranger todas as informações que serão criadas e utilizadas no processo de construção desta modalidade dentro da IES e alinhá-los com a missão e finalidade desta com o propósito de aumentar a eficiência e eficácia dos recursos utilizados. Segundo Laurindo e Rotondaro (2011) é também bastante aceito nos meios acadêmicos e profissionais que o alinhamento estratégico entre TI e negócio direciona para resultados mais eficazes dos investimentos feitos em tecnologia.

Partindo desta premissa, IES que ofertam diversas modalidades de ensino, dentre elas a modalidade de educação a distância, possuem sistemas de gestão acadêmica (SGA) para controlar e gerenciar o percurso acadêmico de seus discentes, mas ocorre que o ambiente virtual de aprendizagem (AVA) adotado pela instituição nem sempre está integrado ao seu sistema de gestão acadêmica, podendo resultar em fluxos de trabalho complexos, lentos e propícios a erros e que muitas vezes dependem da cooperação de diferentes setores ou órgãos dentro da instituição, fatores limitadores ao bom funcionamento da modalidade.

A integração dos sistemas de gestão acadêmica e o ambiente virtual de aprendizagem, doravante referenciados neste trabalho apenas de SGA e AVA, atribui maior grau de eficiência e complementaridade no uso de tais sistemas para o enriquecimento do processo de gestão acadêmica de ambos.

Este trabalho, portanto, propõe-se a investigar e discutir as melhores abordagens para integração de SGAs e AVAs com a finalidade de contribuir no desenvolvimento das atividades de gestão da informação dentro de IES que apresentem demanda semelhante, por meio do uso de tecnologias que utilizem *web services* e computação em nuvem.

Particularmente o autor deste trabalho atuou como Coordenador de Tecnologia da Informação e Comunicação do Núcleo de Educação a Distância da Universidade Estadual do Piauí – NEAD/UESPI no período correspondente de setembro de 2010 a agosto de 2014, tendo enfrentado o problema destacado inicialmente que trata da falta de integração entre o SGA e AVA.

Gerir e manter informações acadêmicas mantidas em dois sistemas de informações é uma atividade trabalhosa que em se tratando de vários cursos na modalidade EAD sendo praticados simultaneamente gerava diversos problemas que serão destacados em seguida, contudo cabe ressaltar que o autor hoje atua no Instituto Federal do Piauí – IFPI que também possui a modalidade EAD e que, surpreendentemente, também possui o mesmo problema apresentado pela Universidade Estadual do Piauí, ou seja, esta situação não é característica tão somente de uma instituição, pois este mesmo problema acomete outras IES que poderão beneficiar-se do modelo de solução a ser apresentado.

Desta maneira, o estudo destacado neste trabalho foi motivado por um problema concreto que acomete diversas IES, portanto buscar soluções para resolver o problema da forma mais eficiente o possível serve de elemento balizador para concretização deste estudo.

1.2. Problema da Pesquisa

Inicialmente destacado, a modalidade de educação a distância, doravante chamada apenas de modalidade EAD, surge como uma alternativa eficaz para equacionar a desigualdade educacional de um país de dimensões continentais como o Brasil. Para modalidade presencial torna-se difícil a equitativa distribuição de recursos para uma educação de qualidade e uniforme.

Contudo, o investimento na modalidade EAD exige capacitação e treinamento adequado dos profissionais envolvidos (docentes e equipe multidisciplinar); produção de materiais didáticos; aquisição de equipamentos; segurança; escolha e preparação do AVA; desenvolvimento de sistemas de gestão e operacionalização, sem jamais esquecer que a ciência e tecnologia avançam continuamente, o que leva a uma necessidade contínua de atualização dos conteúdos didáticos e equipamentos.

No âmbito das atividades docentes, estes têm de lidar com certos aspectos da modalidade EAD como as atividades inerentes de elaborar e montar disciplinas em salas virtuais do AVA, bem como acompanhar e avaliar os resultados obtidos pelos discentes ao passo que, em muitos casos, o fazem em sistemas heterogêneos (SGA e AVA).

Simplificando, o SGA atém-se a questões de controle administrativo, e o sistema de gestão de aprendizagem, o qual se caracteriza por ser um AVA, em que há ferramentas que visam facilitar a prática docente, oferecem recursos para promover a implementação de um ensino de excelência, comprometido com o processo de desenvolvimento social e intelectual, segundo Carvalho (2010).

A utilização dos supracitados sistemas de forma dissociada gera retrabalho e conseqüentemente risco de erros não apenas por parte dos docentes, mas dos membros que compõe a equipe multidisciplinar que atuam na modalidade EAD, que podem se deparar com as seguintes situações:

- **Cadastros incorretos de docentes e tutores no AVA**

Considerando que os dados corretos dos cadastros destes usuários seriam provenientes do SGA, estes também serão parte integrante do AVA, e

consequentemente podem vir a ficar inconsistentes caso não sejam provenientes da mesma base de informações.

- **Divergência no processo de matrícula curricular**

Por padrão a matrícula curricular do discente é efetuada no SGA a cada período/semestre letivo e o mesmo deve ocorrer no AVA. Sem um processo de integração, caso esse procedimento seja feito de forma manual, podem surgir erros de matrícula.

- **Listas de alunos incorretas**

A ausência de um processo de integração pode levar a uma série de inconsistência de dados. Para exemplificar, o discente pode ter requerido trancamento de seu curso no SGA, mas aparece curricularmente matriculado no AVA ou mesmo ter efetuado matrícula no SGA, mas não possuir acesso ao AVA.

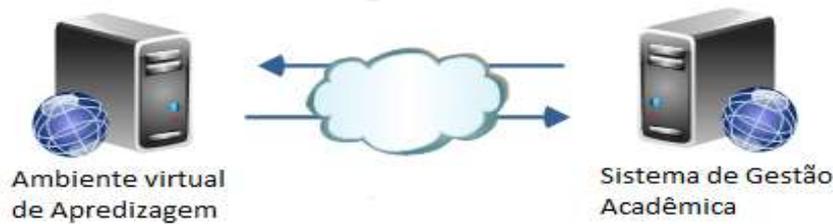
- **Redundância de avaliações e notas**

Normalmente discentes da modalidade EAD acompanham os resultados de suas avaliações e notas pelo AVA, contudo para registros oficiais as mesmas devem ser registradas no SGA. A falta de um processo de integração pode gerar redundância na digitação em ambos sistemas que podem levar a inconsistências.

Desta forma, o problema de pesquisa envolvendo este trabalho está na possibilidade de construção de um modelo que possibilite integrar o SGA e AVA de modo haver interoperabilidade entres eles, que funcione de forma descentralizada, estabelecendo assim funcionalidades que melhorem o tempo despendido nos sistemas supramencionados assim como oferecer elementos para ajudar docentes e membros de equipes multidisciplinares no processo de operacionalização e gerencia dos discentes.

Conforme destacado na figura 1 o objetivo a ser atingido neste trabalho visa a troca de dados/informações entre SGA e AVA que garanta um fluxo de operacionalização entre ambos automatizado e dinâmico.

Figura 1: Processo de troca de dados/informações



Fonte: Adaptado pelo autor

De maneira simplificada os sistemas envolvidos funcionariam ora como clientes consumindo dados/informações, ora como servidores provendo estes mesmos dados/informações por meio da troca de mensagens entre os participantes independentes da infraestrutura que dê suporte a seu funcionamento.

1.3. Motivação

A fim de evidenciar a motivação que move este trabalho, é descrito a seguir o contexto no qual se encontra o Instituto Federal de Educação, Ciência e Tecnologia do Estado do Piauí (IFPI).

A modalidade EAD do IFPI foi contemplada no Edital 01/2007/SEED/SETEC/MEC, referente ao Programa Escola Técnica Aberta do Brasil (ETAPI). O E-Tec Brasil (<http://portal.mec.gov.br/rede-e-tec-brasil>) é um programa de formação profissional técnica a distância, que se constitui em uma das ações do Plano de Desenvolvimento da Educação.

Após aplicação de uma entrevista semiestruturada com membros da direção geral da Rede E-Tec EAD IFPI, constatamos haver 08 cursos de nível técnico na modalidade EAD distribuídos em 14 municípios do Estado do Piauí com 1.553 discentes curricularmente matriculados nestes cursos.

O IFPI está em processo de credenciamento para Universidade Aberta do Brasil (UAB) podendo assim ofertar em breve cursos de nível superior de graduação, tendo ofertado no último semestre de 2015 pós-graduação na modalidade EAD. Ou seja, há uma demanda a suprir no que diz respeito a gestão das informações acadêmicas e administrativas destes discentes, visto os mesmos fazerem uso de um AVA para realização de suas atividades acadêmicas diárias, ao tempo em que precisam estar em sincronia com as informações administrativas geridas pelo SGA da instituição.

Esta demanda pode ser expandida também aos cursos da modalidade presencial, pois de acordo com a portaria número 4.059 do Ministério da Educação de 10 de dezembro de 2004:

Art.1o. As instituições de ensino superior poderão introduzir, na organização pedagógica e curricular de seus cursos superiores reconhecidos, a oferta de disciplinas integrantes do currículo que utilizem modalidade semipresencial, com base no art. 81 da Lei n. 9.394, de 1.996, e no disposto nesta Portaria.

Ou seja, as IES podem ofertar disciplinas ou parte delas na modalidade EAD fazendo uso do AVA, e considerando o fato que ambos sistemas são dissociados um do outro e, portanto, não trocam informações entre si, apresenta-se um quadro no qual existem muitos fluxos de trabalhos desnecessários que podem ser otimizados a partir da integração entre SGA e AVA.

De acordo com Lima e Neto (2009), as universidades modernas têm como uma das características mais importante um bom sistema de informação que registre todas as atividades da vida de sua academia. Desta forma, percebe-se a necessidade e importância do uso de mecanismos ou ferramentas que possibilitem a integração e, conseqüentemente, a gerencia efetiva e eficaz das informações provenientes desses sistemas.

Nestes termos, o AVA atua como responsável por todo processo de ensino e aprendizagem do discente da modalidade EAD e o SGA se encarrega do processo de controle administrativo de sua vida acadêmica.

Desta maneira, a questão principal deste trabalho repousa em como a integração do SGA e do AVA pode contribuir para melhoria do fluxo de processos de operacionalização dos mesmos e por meio da elaboração de um modelo de integração que garanta interoperabilidade entre eles.

Propor um modelo de solução entre sistemas heterogêneos que contribua para modalidade EAD e seus atores pode trazer benefícios específicos dentre os quais podem-se destacar:

- Estabelecer um meio de integração entre sistemas, independentemente da plataforma de *hardware* e *software* utilizados.
- Simplificar fluxos de processos, condensando atividades que não acrescentem valor ao processo de operacionalização do SGA e do AVA.
- Aumentar sensivelmente a produtividade do processo de operacionalização do SGA e AVA.

1.4. Objetivos

1.4.1. Objetivo Geral

Propor um modelo de integração por meio de abordagens que sejam capazes de prover uma solução para sistemas originalmente independentes, especificamente entre SGA e AVA, provendo interoperabilidade entre os mesmos.

1.4.2. Objetivo Específicos

- Discutir e caracterizar as abordagens para integração de sistemas de informação;
- Analisar e adequar uma metodologia dentre as abordagens existentes para integração de sistemas que permita apoiar o desenvolvimento da solução proposta;
- Desenvolver um modelo de integração entre SGA e AVA.

1.5. Justificativa

A utilização da tecnologia da informação e comunicação (TIC) tornou-se estratégica e dispor dos recursos provenientes da TIC de forma eficaz e de modo efetivo tornou-se fator de destaque.

As IES não estão imunes a este fato, pois a utilização de sistemas de informações permeia todos os níveis da sociedade atual, sendo que em diversos casos é comum a utilização de vários sistemas, cada um com sua finalidade, e que podem resultar em informações duplicadas.

No contexto deste trabalho o AVA necessita de diversas informações para povoar seu banco de informações, tais como: cursos e disciplinas ofertadas no período/semestre acadêmico, os municípios em que estes serão ofertados, a efetivação de matrícula curricular dos discentes para o período/semestre, dentre outras informações, e considerando a possibilidade da realização do cadastro destas mesmas informações no SGA, temos um quadro no qual há a possibilidade concreta dos dados estarem dissociados e, portanto, passíveis de falhas, pois o fluxo destas informações pode conter erros que comprometam o processo de gerência que envolvam tanto o SGA como o AVA, da mesma forma como o processo de tomada de decisões.

Quando as entradas são limitadas a uma atividade da empresa, as saídas também serão, o que torna o sistema de informação isolado, limitando as decisões tomadas. Para aumentar a capacidade de decisão, é necessária a disponibilidade de informações mais abrangentes, disponíveis em mais de um sistema de informação que, portanto, devem ser integrados. Favaretto (2011, p. 1)

Os benefícios oriundos da integração desses sistemas trarão padronização de processos, visto que a interoperabilidade entre ambos reduz erros no fluxo da informação, aumenta a velocidade da informação compartilhada e gera-se uma significativa melhora no desempenho de processos administrativos e acadêmicos.

De tal maneira, as equipes de trabalho destes sistemas terão acesso à informação mais rapidamente de forma atualizada e confiável, assim como acesso a informações estratégicas:

O estudo da integração entre sistemas de gestão acadêmica e ambientes virtuais de aprendizagem apresenta uma série de benefícios que justificam o empenho de esforços na elaboração desta pesquisa [...]. Desta forma, observa-se que o crescente uso do sistema de gestão acadêmica e do sistema de gestão da aprendizagem, facilita, respectivamente, o controle administrativo e o controle pedagógico nas IES. Contudo, o uso isolado dos sistemas de gestão acadêmica e de gestão da aprendizagem, referidos anteriormente no texto, ainda não alcança as reais necessidades dos docentes e discentes envolvidos no processo de ensino-aprendizagem. Isso, porque, para cada atividade exercida, o docente dedica certo tempo, em cada um dos ambientes, de forma a fazê-lo repetir a mesma atividade, o que pode resultar em inconsistência de informação, além da considerável perda de tempo na realização de atividades simples como lançamento de notas, por exemplo. Silva (2012, p. 22)

Desta forma processo de integração une e mescla informações do SGA e do AVA e disponibiliza informações estratégicas para melhor tomada de decisões e planejamento institucional das IES.

1.6 Organização

Este trabalho está organizado da seguinte forma:

Capítulo 2 – Revisão da literatura e Trabalhos relacionados: Apresenta a revisão da literatura efetuada para dar embasamento a este trabalho e os conceitos que o norteiam assim como apresenta trabalhos correlatos e sua relevância para realização da pesquisa atual.

Capítulo 3 – Abordagens para integração: Este capítulo pretende debater as abordagens e tecnologias existentes para o processo de integração de sistemas a fim de demonstrar os motivos que levaram a escolha da solução que será empregada neste trabalho.

Capítulo 4 – Metodologia aplicada: Apresenta-se a metodologia de pesquisa utilizada para

realização deste trabalho.

Capítulo 5 – Arquitetura e modelagem: Neste capítulo objetiva-se apresentar e descrever o modelo de solução proposta e seu processo de validação, assim como possíveis dificuldades oriundas a partir desta proposta.

Capítulo 6 – Considerações finais: Este capítulo destaca as considerações finais e possíveis trabalhos futuros.

Capítulo 7 - Publicações: Este capítulo aborda as publicações obtidas a partir do estudo efetuado para concretização deste trabalho.

2. REVISÃO DA LITERATURA E TRABALHOS RELACIONADOS

2.1 Sistemas de Informação

É perceptível em todos os segmentos da sociedade a utilização de algum tipo de sistema de informação para dar suporte a algum tipo de atividade de trabalho. De fato, quando se menciona o termo “sistema”, nos dias atuais, a grande maioria pensa em sistemas de informação tais como os sistemas operacionais que acompanham equipamentos de TIC (notebooks, smartphones, tablets, etc), os sistemas da rede bancária de caixas eletrônicos, os sistemas tributários, fiscal e de impostos da Receita Federal, etc., contudo o conceito de sistemas é muito mais amplo e recente.

O biólogo alemão Von Bertalanffy¹ publicou entre os anos de 1950 e 1958 uma série de trabalhos sobre o comportamento dos organismos vivos tendo constatado que, independente de características, variedades e formas, esses organismos biológicos possuíam pontos em comum, tendo desenvolvido a ideia de que o organismo é um todo maior que a soma das suas partes.

Seus estudos em organismos vivos se estenderam a outros organismos (mecânicos, eletrônicos, sociais, etc) tornando-o multidisciplinar, sendo que todos esses organismos não naturais têm como principal característica o alcance de um objetivo comum, mesmo possuindo diversos e variados componentes, todos trabalham e interagem para concretização de uma finalidade central única de um determinado propósito.

Segundo o dicionário Michaelis a definição de sistema é ampla: 1 Conjunto de princípios verdadeiros ou falsos, donde se deduzem conclusões coordenadas entre si, sobre as quais se estabelece uma doutrina, opinião ou teoria. 2 Corpo de normas ou regras, entrelaçadas numa concatenação lógica e, pelo menos, verossímil, formando um todo harmônico. 3 Conjunto ou combinação de coisas ou partes de modo a formarem um todo complexo ou unitário.

Há, pelo menos, uma definição possível para cada área do saber que envolva o conceito de sistemas, seja na física, biologia, química, administração, etc., contudo é correto presumir que sistema é um conjunto de partes que, apesar de apresentarem uma certa independência, necessitam uma das outras para atingirem seus objetivos, que somados abrangem um objetivo único de maior propósito.

Nesse contexto o foco do estudo em questão é a área da Informática e, portanto, a

1 Karl Ludwig von Bertalanffy (1901-1972) Criador da teoria geral dos sistemas. Fez seus estudos em biologia e interessou-se desde cedo pelos organismos e pelos problemas do crescimento. Fonte: <https://pt.wikipedia.org/>

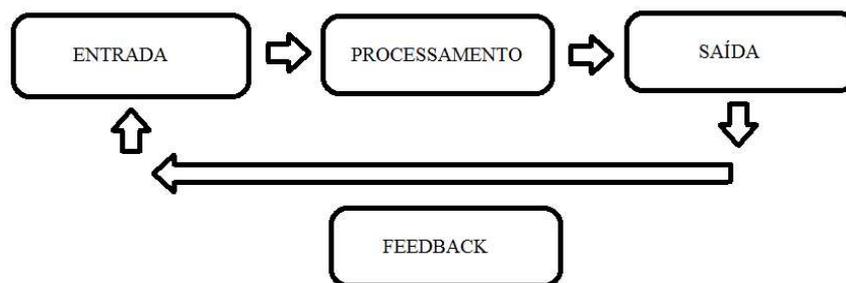
definição para sistemas de informação também é ampla e dispõe de diversos conceitos conforme diversos autores e pesquisadores.

O conceito de sistema de informação segundo O'Brien (2004) é um conjunto organizado de pessoas, *hardware*, *software*, rede de comunicação e recursos de dados que coleta, transforma e dissemina informações em uma organização. Para Laudon e Laudon (2012), um sistema de informação pode ser definido tecnicamente como um conjunto de componentes inter-relacionados que coleta, processa, armazena e distribui informações destinadas a apoiar as tomadas de decisões, a coordenação e o controle de uma organização. De forma complementar:

É um sistema de elementos ou componentes inter-relacionados, numa ordem específica que coletam (entrada), manipulam (processamento), disseminam (saída) os dados e informações e fornecem um mecanismo de feedback (retroalimentação). Essas informações são então utilizadas pelos usuários para a tomada de decisões. Albertão, apud Cortês (2008).

Em síntese sistemas de informações tem por finalidade auxiliar e facilitar o controle, planejamento, análise e a tomada de decisão dentro de uma organização, sendo que o mesmo deve possuir dados de entrada, os quais são processados ou transformados para gerar informações de saída somados a um elemento de *feedback* ou retroalimentação para “ajustar” a entrada, permitindo assim que o sistema de informação se corrija automaticamente com a finalidade de suprir as necessidades do tomador de decisão, como destacado na figura 2 a seguir.

Figura 2: Representação gráfica de um sistema



Fonte: Adaptado pelo autor.

No âmbito das IES, estas necessitam de sistemas de informações para executar o tratamento automático de seus dados e informações tanto no aspecto administrativo quanto no acadêmico. Para isso, esses sistemas devem trabalhar de forma integrada, provendo dessa forma uma relação harmônica e de dependência entre eles a fim de evitar retrabalhos desnecessários e dados/informações inconsistentes ou redundantes.

2.2. Sistema de Gestão Acadêmica – SGA

De acordo com Furtado et al (2015) os sistemas de gestão acadêmica são plataformas geralmente desenvolvida em ambiente web para atender as necessidades de gestão e planejamento de uma instituição de ensino, possibilitando a otimização de seus recursos.

Como citado anteriormente o SGA se atém a questões de controle administrativo, pois este dispõe de funcionalidades inerentes ao processo gerencial e administrativo das IES, controlando procedimentos de dados de discentes, cursos, matrizes curriculares, disciplinas, oferta de turmas e de períodos/semestres letivos, históricos acadêmicos, emissão de diplomas e outros mais.

A integridade e consistência das informações geridas pelo SGA permite o acompanhamento e controle adequado de todo processo administrativo possibilitando para IES mecanismos que auxiliam na tomada de decisão, melhoria no monitoramento da qualidade de ensino e a utilização eficiente dos recursos.

Portanto, considerando o contexto atual o sistema de gestão acadêmica é uma ferramenta fundamental ao controle dos processos administrativos e gerenciais da gestão acadêmica das instituições de ensino. Tal sistema propõe-se a controlar e agilizar os processos da instituição, possibilitando a consolidação de informações relevantes para a gestão, através da análise de dados como: matrículas, aproveitamento acadêmico, frequência, evasão, e de vários outros indicadores. Carvalho (2010, p. 45)

Assim sendo, é indiscutível a significativa relevância de um SGA para uma IES e considerando o aumento da oferta de cursos na modalidade EAD, com a consequente utilização de um AVA, a consolidação das informações de ambos deve estar alinhada ao controle administrativo e o controle pedagógico das IES.

Existe no mercado uma grande variedade de SGAs que se enquadram na categoria de *softwares proprietários*², ou seja, o mesmo não pode ser alterado e deve ser adquirido, na maioria dos casos, por meio de licenças de uso de elevado custo para as IES. Da mesma forma, existem SGAs que são pertencentes a categoria de *softwares livres* que em grande parte dos casos são gratuitos e disponibilizam seu *código-fonte*³ para adaptações.

² O software proprietário, privativo ou não livre é um software para computadores que é licenciado com direitos exclusivos para o produtor. Conforme o local de comercialização do software este pode ser abrangido por patentes, direitos de autor assim como limitações para a sua exportação e uso em países terceiros. Seu uso, redistribuição ou modificação é proibido, ou requer que você peça permissão, ou é restrito de tal forma que você não possa efetivamente fazê-lo livremente. Fonte: <https://pt.wikipedia.org/>

³ O código-fonte (também chamado de fonte ou código) é uma versão do software da forma em que ele foi originalmente escrito (digitado em um computador) por um humano em texto puro (caracteres alfanuméricos humanamente

O atual SGA do IFPI se enquadra na categoria de software proprietário, o mesmo chama-se Q-Acadêmico e é um produto de *software* que pertence à empresa Qualidata (<http://www2.qualidata.com.br/gestaoacademica.htm>), segundo informações obtidas na reunião com a equipe de EAD do IFPI, ou seja, seu código fonte não pode ser alterado.

Portanto, no contexto abordado neste trabalho, faz-se necessária uma solução cujo modelo possibilite um nível de interoperabilidade entre os sistemas por meio de tecnologias que façam sistemas distintos se comunicarem. O que no âmbito deste trabalho é um fator importante que será debatido no capítulo referente à discussão teórica.

2.3. Ambiente Virtual de Aprendizagem – AVA

Um AVA é um sistema de informação que agrupa diferentes funções e ferramentas com o propósito básico de apoiar o processo de ensino e aprendizagem na modalidade EAD ou semipresencial.

Ambiente virtual de Aprendizagem (AVA), também conhecido como Learning Management System (LMS), é uma infraestrutura de e-learning com as funções de entrega dos cursos que permitem a colaboração e a avaliação do desempenho do aluno, registrando dados e gerando relatórios para maximizar a eficácia da organização de aprendizagem. Yasar e Adiguzel, (2010, p. 1)

A intenção dos AVA é proporcionar não somente conteúdos para o processo de ensino e aprendizagem, mas principalmente plena interação e interatividade entre seus usuários, permitindo assim o processo de construção do conhecimento.

Existem atualmente diversos AVA comerciais e gratuitos no mercado dos quais destacam-se:

- *Eureka*;
- *LatitudeLearning*;
- *Geenio*;
- *E-Proinfo*;
- *BlackBoard*;
- *Sakai*;
- *TalentLMS*;

- *Docebo*;
- *Atutor*;
- *Moodle*;

Contudo, existem algumas características básicas que são recomendadas para um AVA, conforme destacado por Bione (2014):

1. Administração: O software deve permitir aos administradores gerenciar os registros de usuários e perfis, definir os papéis, os currículos, caminhos de certificação, atribuir funções aos tutores, funções de autor de cursos, gerenciar conteúdo;
2. Adesão coerente de padrões de aprendizagem: O AVA tem de obedecer aos principais padrões de *e-learning*, como SCORM – *Sharable Content Object Reference Model* e os emitidos pelo *Institute of Electrical and Electronics Engineers – IEEE* e *Learning Technology Standards Committee – LTSC*;
3. Arquitetura modular: Embora não seja obrigatório para uma plataforma AVA, é desejável a possibilidade de integrar nativamente módulos. O uso de código aberto contribui para esse aspecto, ao permitir ao público em geral a modificação do sistema e a inserção de módulos. Em geral, as modificações do usuário são absorvidas pelo projeto principal, e com isso o software evolui no sentido de incorporar os valores da comunidade de usuários.

Dentro do contexto deste trabalho o *Moodle (Modular Object-Oriented Dynamic Learning Environment)* será utilizado como AVA para o processo de interação entre sistemas proposto, de acordo com item 3.1 mais à frente, pois o mesmo reúne todas as características básicas elencadas anteriormente. Além disso, trata-se de um software livre e gratuito que é utilizado em milhares de instituições em mais de 200 países, tendo sido traduzido em mais de 70 línguas. Também é o AVA adotado pelo IFPI para sua modalidade EAD.

2.4. Web Services e Computação em Nuvem

A solução proposta neste trabalho para equacionar a falta de integração entre os sistemas supracitados passa pela utilização de *web services* e computação em nuvem para prover uma solução que funcione de forma descentralizada constituindo um modelo que possa ser utilizado pela maioria das IES. Assim sendo, da mesma forma que foram apresentados e conceituados o SGA e AVA, faz-se necessário apresentar e destacar a importância destas tecnologias.

2.4.1. Web Services

Integração de sistemas de informação, independente de plataformas de *software* e *hardware*, é um desafio central em *sistemas distribuídos*⁴ na *Internet*, como na infraestrutura subjacente a plataformas de EAD. Ademais, a proliferação massiva de dispositivos nesse contexto reforça a necessidade de padronizações, provendo meios de integração e troca de dados em um cenário fortemente híbrido.

Para Gomes (2014) *web services* são uma tecnologia de integração de sistemas empregada principalmente em ambientes heterogêneos. De posse dessa tecnologia é possível que novas aplicações possam interagir com aquelas existentes num mesmo ambiente e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Para Kalin (2010) um *web service* é, então, uma aplicação distribuída, cujos componentes podem ser aplicados e executados em dispositivos distintos.

O próprio termo sintetiza bem sua representação, pois trata-se de um serviço que é disponibilizado pela *Internet* podendo ser acessado de qualquer lugar, conforme destacado:

A demanda por soluções de negócios integradas e distribuídas cresce a cada dia, e a teoria de *web services* vem ao encontro dessas necessidades para dar uma solução definitiva quanto a forma de se materializar novas estratégias de negócio orientadas a serviços. Sua capacidade de criar soluções distribuídas e descentralizadas nos dá ferramentas para, por exemplo, manter equipes de trabalho em diferentes países operando cooperativamente e simultaneamente. Marzullo (2009, p. 151).

A utilização comum desta tecnologia está na construção de componentes de software ou no suporte à interoperabilidade de dados entre sistemas. Entretanto, é perceptível a utilização de *web services* construídos com a finalidade de troca de dados entre aplicações. Em resumo, *web services* proveem um padrão de interoperabilidade entre sistemas de informações distintos que são direcionados a construção de serviços.

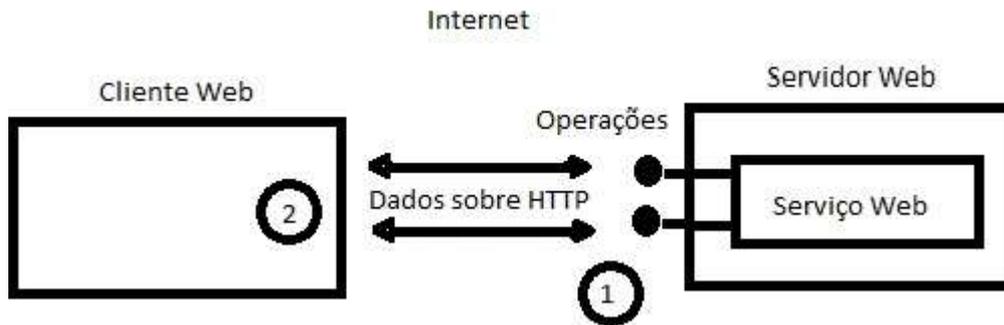
Do ponto de vista conceitual, Machado (2016) apresenta o funcionamento de um *web service* sob uma perspectiva abstrata: (1) o serviço é executado em um *servidor Web*⁵ e expõe operações na

⁴ Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente. Fonte: *Sistemas Distribuídos: princípios e paradigmas* / Andrew S. Tanenbaum.

⁵ O Servidor WEB é o programa responsável pela publicação de documentos, imagens ou qualquer outro objeto que venha a ser acessado por um cliente através de um navegador. Fonte: www.devmedia.com.br

rede por meio do protocolo *HTTP*⁶, (2) o *cliente Web*⁷ executa as operações trocando dados com o serviço conforme a figura 3 abaixo:

Figura 3: Funcionamento de um *Web Service*



Fonte: Machado (2016)

Partindo deste princípio é importante destacar a filosofia SOA⁸, chamada de Arquitetura Orientada a Serviços (SOA): esta não é uma tecnologia, não é uma metodologia, não é um serviço, mas é um conceito de arquitetura que provê a integração entre aplicações que venham a ser construídas através da reutilização de serviços e não da criação de um novo código fonte, fazendo com que as instituições que possuem seus sistemas de informações baseadas em *Internet* venham a oferecer comodidade para seus usuários por meio de um conjunto de rotinas para comunicação com diferentes plataformas.

Com a finalidade de apoiar a arquitetura SOA é importante destacar o paradigma Computação Orientada a Serviços (SOC – *Service-oriented Computing*)⁹. O termo tem ganhado espaço no contexto das TIC, pois prega o fraco acoplamento e a interoperabilidade utilizando serviços no desenvolvimento de soluções para integrar ambientes originalmente independentes.

A partir do paradigma SOC, a Arquitetura Orientada a Serviços (SOA – *Service-oriented Architecture*) surge como uma forma lógica de projetar sistemas sob o paradigma SOC, isso é, provendo serviços distribuídos por meio de interfaces de descoberta e publicação de serviços. Papazoglou et al, apud Souza e Fantinato (2013).

⁶ HTTP é sigla de HyperText Transfer Protocol que em português significa Protocolo de Transferência de Hipertexto. É um protocolo de comunicação entre sistemas de informação que permite a transferência de dados entre redes de computadores, principalmente na World Wide Web (Internet). Fonte: www.significados.com.br

⁷ O cliente web é qualquer máquina que requisita informações ou serviços de outro. O termo tipicamente se refere aos computadores utilizados no dia a dia, que interage através do teclado, monitor, mouse ou interface similar. Fonte: www.ehow.com.br

⁸ SOA - (Service-Oriented Architecture) – Arquitetura Orientada a Serviços. É uma especificação para a troca de dados entre sistemas, ou seja, uma especificação de formato de dados para envio de estruturas de dados entre serviços, com um padrão para permitir a interoperabilidade entre eles.

⁹ SOC – Paradigma de computação que utiliza serviços como elementos fundamentais para o desenvolvimento de aplicações.

Uma das maneiras possíveis de implementar SOC é através do uso de *web services*, onde *web services* desempenha a função de prover serviços.

Ainda neste capítulo no tópico referente a trabalhos relacionados será fácil perceber a importância desta tecnologia no trabalho em curso. Já no capítulo referente a discussão teórica serão discutidas as abordagens mais utilizadas para implementação de *web services* destacando suas características para escolha do modelo.

2.4.2. Computação em Nuvem

Todas as tendências da área de Tecnologia da Informação e Comunicação (TIC) para os próximos anos destacam conceitos como *big data*¹⁰, mobilidade, computação em nuvem, dentre outras. Estes termos já são realidade em diversos contextos tais como empresariais, de negócios e de novos serviços, ou senão pelo menos fazem parte dos planos de diversos gestores da área de Tecnologia da Informação e Comunicação para os anos vindouros.

O *National Institute of Standards and Technology* (NIST) (<http://www.nist.gov/itl/cloud/>) define computação em nuvem como um modelo para permitir acesso ubíquo, conveniente e sob demanda através da rede a um “*pool*” configurável de recursos computacionais (por exemplo, redes de computadores, servidores, armazenamento em disco, aplicações e serviços) que podem ser rapidamente provisionados e liberados para uso com um esforço de gerenciamento mínimo e de preferência sem interferência direta do provedor de serviço.

Santos (2015) destaca que os benefícios da computação em nuvem (*cloud computing*) quando bem aplicados geram uma flexibilidade e produtividade essencial para o desenvolvimento de qualquer organização.

Alguns fatores que contribuíram para o crescimento da computação em nuvem foram a diminuição do custo do hardware e de dispositivos de armazenamento, a adoção do processo de virtualização de *hardware* e o aumento da capacidade das redes. As características destacadas são apresentadas no quadro 1:

¹⁰ Conjuntos de dados extremamente amplo e que, por este motivo, necessita de ferramentas especialmente preparadas para lidar com grandes volumes, de forma que toda e qualquer informação nestes meios possa ser encontrada, analisada e aproveitada em tempo hábil. Fonte: [www.infowester.com /big-data.php](http://www.infowester.com/big-data.php)

Quadro 1: Características da computação em Nuvem.

| | |
|------------------------------------|--|
| Elasticidade (Escalabilidade) | Flexibilidade de escalar e poder se adaptar conforme o volume de demanda do usuário. A alocação de recursos varia, aumentando para atender aos grandes momentos de pico e diminuindo para os momentos de baixa demanda. |
| Aprovisionamento Automatizado | O usuário requisita uma determinada quantidade de recursos de computação (<i>software</i> , armazenamento) para o provedor. Desta maneira os usuários podem provisionar os serviços da nuvem sem ter que passar por um extenso processo burocrático, mantendo adequada a magnitude da sua demanda. |
| Interfaces Padronizadas | Define a forma como os clientes utilizam os serviços expostos na nuvem. Esta interface é formada por um conjunto de normas e instruções que estabelecem a comunicação entre os componentes, camadas e usuários que participam da solução implantada na nuvem. |
| Multitenacidade / Multi-Inquilinos | Permite que vários locatários compartilhem a mesma instância do serviço oferecido pela nuvem. Vários clientes podem estar centralizados no mesmo provedor, o que resulta numa divisão e diminuição dos custos. Vale a pena mencionar que o termo Multitenacidade está sendo utilizado também para se referir a esta característica. |
| Virtualização | Por conta da virtualização é possível realizar o compartilhamento de servidores e dispositivos de armazenamento. Por exemplo, executar múltiplos Sistemas Operacionais em uma única máquina, compartilhando assim a plataforma de hardware. Desta forma as aplicações podem ser migradas entre servidores físicos com maior facilidade. Atualmente a virtualização já é considerada uma alternativa estabilizada para o compartilhamento de <i>hardware</i> . |
| Custo | Esta característica da computação em nuvem talvez seja a que mais desperta o interesse dos usuários da nuvem. Ao invés de ter que constituir toda uma plataforma de <i>hardware</i> , aplicativos e dispositivos de armazenamento, se atribui toda essa responsabilidade para o meio externo, economizando com custos de manutenção, monitoramento, espaço físico e evolução da plataforma. Além disso, se ganha flexibilidade na adaptação da sua demanda com os recursos da nuvem sendo utilizados. Por exemplo, é possível provisionar mais servidores para o período diurno e dispensar grande parte desses servidores no período noturno, onde sua demanda é baixa. |

Fonte: Adaptado de <http://www.devmedia.com.br/computacao-java-nas-nuvens-revista-java-magazine-108/25961>

Computação em nuvem é dividida em três modelos de serviço: *IaaS* (Infraestrutura como serviço), *PaaS* (Plataforma como Serviço) e *SaaS* (Software como Serviço).

Software como Serviço (*Software as a Service - SaaS*) é um modelo em que um aplicativo é oferecido com um serviço aos clientes que o acessam através da *Internet*, normalmente usando um navegador. O *SaaS* é utilizado em praticamente todos os serviços de *Internet* nos dias de hoje, como *webmail* (*Outlook, Gmail*) ou mecanismos de busca (*Google*). O conceito de *software* como serviço é muito abrangente e mais aplicado no contexto de aplicações de negócios.

A adoção de computação em nuvem, de modo particular ao *SaaS*, é uma excelente forma de entregar rapidamente soluções tecnológicas para atender necessidades de empresas e usuários comuns.

Plataforma como Serviço (*Platform as a Service - PaaS*) é o modelo mais utilizado por

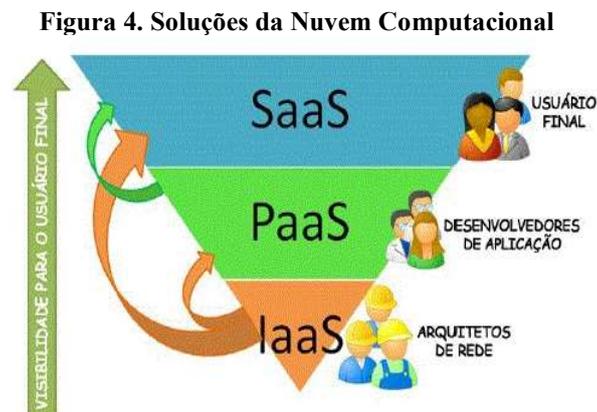
desenvolvedores, pois este fornece todos os recursos necessários para construir serviços e aplicações sem a necessidade de baixar e instalar *softwares*. O usuário/desenvolvedor não pode controlar ou administrar os recursos da infraestrutura como armazenamento, processamento, e memória, mas possui controle sobre os aplicativos desenvolvidos e as configurações no ambiente de hospedagem.

O *PaaS* é útil para atividades que necessitem colaboração em equipe, integração de serviços, integração de banco de dados, segurança, escalabilidade, gerenciamento de armazenamento.

Infraestrutura como Serviço (*Infrastructure as a Service – IaaS*) é um modelo que fornece controle sobre redes, armazenamento, processamento e outros. O usuário pode instalar sistemas para o funcionamento da máquina virtual como sistema operacional e aplicações.

O usuário não pode administrar ou controlar a infraestrutura do servidor, estas atribuições competem aos arquitetos de rede, contudo os usuários têm controle sobre o sistema operacional, armazenamento das informações, controle de memória, aplicativos implantados.

Os modelos de serviços elencados acima podem ser visualizados conforme destacado na figura 4.



Fonte: Mizukoshi e Garcia (2012)

Se comparado ao modelo atual, no qual possuir um *Datacenter*¹¹ dentro da IES requer infraestrutura de médio e grande portes que podem exceder as necessidades computacionais e estas ficarem ociosas uma grande parte do tempo, computação em nuvem se apresenta como uma abordagem de relação custo-benefício interessante além de promover novas possibilidades para aplicações orientada a serviços.

¹¹ O Datacenter é um ambiente projetado para abrigar servidores e outros componentes como sistemas de armazenamento de dados (storages) e ativos de rede (switches, roteadores). O objetivo principal de um Data Center é garantir a disponibilidade de equipamentos que rodam sistemas cruciais para o negócio de uma organização. Fonte: <https://www.telecorp.com.br/glossario/data-center/>

Cuidar da infraestrutura de servidores é algo complexo e envolve gerenciar recursos de redes, armazenamento, *backup*, processamento, memória, energia etc [...]. Esta abordagem clássica é muito utilizada, mas está perdendo cada vez mais espaço para a computação em nuvem, que inverte os papéis e permite a empresa ou ao desenvolvedor estar totalmente no controle e gerenciar seu próprio datacenter de forma simples e ágil. Lecheta (2015, p. 29).

Dentre as características elencadas no quadro 1, a que se refere a custo está ligada diretamente ao modelo *pay-per-use* (pagar por uso) no qual se paga somente pelo que for utilizado, tornando os recursos facilmente escaláveis, podendo-se aumentar ou diminuir a quantidade ou demanda dos mesmos sem grandes esforços ou custos que estariam vinculadas as limitações de um *datacenter* administrado por uma IES, por exemplo.

Isto pode ser comprovado pelo fato de grandes empresas da área de Tecnologia da Informação disponibilizarem o serviço de computação em nuvem com alto grau de eficiência. Velte, Velte e Elsenpeter (2011) destacam que existem dezenas de fornecedores de nuvem lá fora, mas não deve ser surpresa que alguns dos maiores nomes em computação em nuvem sejam os maiores nomes do mundo da informática. Empresas como *Apple, Facebook, Google, Amazon, IBM e Microsoft* disponibilizam serviços em computação em nuvem por meio de serviços online confiáveis.

O maior objetivo de soluções em computação em nuvem é proporcionar maior produtividade, pois essa tecnologia não permite somente armazenar dados na nuvem, mas também a construção e conexão de aplicações e na flexibilidade oferecida de não realizar investimento maciços em hardware e pessoal para gerenciar sua infraestrutura.

Destacados os fatos, esta tecnologia confere a este trabalho a flexibilidade e descentralização que serão úteis na concretização do objetivo geral deste.

2.5. Trabalhos Relacionados

Reafirmando, o objetivo geral deste trabalho é o de prover um modelo de integração por meio de tecnologias de integração que sejam capazes de apoiar uma solução para sistemas originalmente independentes, garantindo a interoperabilidade entre eles.

Nessa perspectiva, a revisão de literatura especializada, nos termos e tecnologias a serem utilizadas apenas no âmbito didático/teórico, não bastaria para dar suporte a este trabalho. Desta maneira trabalhos e pesquisas práticas oriundas do meio acadêmico e ligadas no campo de integração de sistemas foram realizados para embasar os termos destacados até aqui e sinalizar a relevância desta pesquisa em comparação a outras similares já realizadas.

A partir deste ponto, serão destacados alguns dos trabalhos pesquisados que são produtos de dissertações, artigos e publicações cujas características e contribuições ratificam os tópicos discutidos anteriormente.

2.5.1. Carvalho (2010)

Sob o título “Sistemas de Gestão de Aprendizagem e Sistemas de Gestão Acadêmica: Avaliados pela ótica docente”, a autora Rosângela Saraiva Carvalho enfoca o problema do não atendimento das reais necessidades docentes na utilização dos sistemas de gestão acadêmica e de gestão de aprendizagem e propõe estabelecer requisitos funcionais que otimizem o tempo que estes atores dispõem nos sistemas citados.

Não são tratados diretamente aspectos técnicos de integração de sistemas no trabalho da autora, contudo, ela aborda a construção de um modelo teórico por meio de questionários, entrevistas semiestruturadas e análise documental para comprovar a eficácia de seu objeto de estudo.

2.5.2. Silva (2012)

Com o título de “Arquitetura Empresarial: Um estudo de caso sobre a integração entre a plataforma Moodle e o SIGAA da UFRN ”, o autor César Augusto Barreto da Silva evidencia um dos aspectos da gestão da informação compreendida como conjunto de políticas e escolhas técnicas de padronização e integração de sistemas de informação sob a perspectiva da área de Administração.

O autor trabalha num estudo de caso da UFRN para estabelecer um alicerce de execução para integração de dois sistemas específicos desta IES: O AVA Moodle e o SGA SIGAA com objetivo de auxiliar no desenvolvimento das atividades de gestão da informação.

Este trabalho descreve características detalhadas do sistema AVA Moodle no que concerne a distribuição do sistema, interatividade, usabilidade, princípios pedagógicos, etc e o sistema SGA SIGAA (sistema proprietário da própria UFRN) e os compara segundo diretrizes de uma arquitetura empresarial dentro da IES.

2.5.3. Furtado et al (2015)

Os autores em epígrafe produziram artigo apresentado no XII Congresso Brasileiro de

Ensino Superior a Distância (ESUD 2015) intitulado “A importância de integrar Ambientes Virtuais de Aprendizagem com Sistemas de Gestão Acadêmica – um modelo de integração”.

São destacados a importância da modalidade EAD e a necessidade da integração destes sistemas de informação com objetivo de que estes se aperfeiçoem para benefício das rotinas administrativas no contexto no qual estão inseridos.

É destacado o preceito de EAI – *Enterprise Application Integration*, e como ele apoia a implementação de ferramentas de integração. Este trabalho aponta a utilização da Arquitetura Orientada a Serviços e *web services* como solução viável por possuírem funcionalidades bem caracterizadas e por serem implementadas numa linguagem de programação e hardwares diversos garantindo interoperabilidade entre sistemas distintos.

É destacado um padrão para utilização de *web services* por meio do protocolo SOAP, protocolo a ser discutido e detalhado mais adiante neste trabalho, para comunicação com as bases de dados de AVAs e SGAs por intermédio de um *software* “Gerenciador” que possui as funções de configuração de parâmetros de interconexão entre as bases de dados e de prover diversas funcionalidades para gerencia dos usuários.

2.5.4. Moura e Bernardino (2010)

O artigo intitulado “Integration Moodle and Academics System” dos autores portugueses Ricardo Moura e Jorge Bernardino trazem a premissa básica de integrar e alinhar sistemas isolados com o objetivo de permitir o compartilhamento de informações entre estes.

Os autores apresentam um modelo de integração para o AVA Moodle e o Sistema Integrado de Gestão do Ensino Superior (SiGES). O trabalho em questão reforça a importância dos *web services* como tecnologia capaz de implementar uma arquitetura orientada a serviços.

O modelo proposto contempla a abordagem de integração e os requisitos a serem atendidos, pois como há um sistema proprietário (SiGES) os acessos a ele são restritos se comparados ao AVA Moodle, por este ser um sistema completamente aberto, ou seja, a solução requer acessos ao SGA apenas por meio de leitura sendo os dados replicados no AVA Moodle.

Da mesma maneira destacada por Furtado et al (2015), no subitem anterior, a solução em epígrafe faz uso do protocolo no padrão SOAP no qual seria possível somente importar os dados do SGA SiGES, por este se tratar de software proprietário e de acesso restrito ao código-fonte, para o do AVA Moodle, não ocorrendo a importação de dados no sentido inverso, ou seja, do AVA para o SGA.

2.5.5. Furtado (2016)

A dissertação de Ulisses Melo Furtado com o título “Modelo de Integração Adaptável entre Ambientes Virtual de Aprendizagem e Sistemas de Gestão Acadêmica baseado em Arquitetura Orientada a Serviços” é constituída de pesquisas na literatura e se baseia nas diretrizes do *Enterprise Application Integration (EAI)* e da Arquitetura Orientada a Serviços (SOA)

É proposto um mecanismo configurável e de fácil ajuste para integrar sistemas de informações que atendam a demanda de AVA e SGA a fim de que estes tenham um bom nível de integração, compartilhando informações importantes para o gerenciamento dos cursos à distância.

O autor apresenta a modelagem e posterior implementação de sua solução, validando o modelo por meio de testes práticos em um ambiente real de utilização. Novamente, a solução proposta pelo autor faz uso do protocolo SOAP com formato de dados XML, formato de dados também a ser discutido e detalhado adiante neste trabalho, contudo sua solução é validada para uso no SGA SIGAA (sistema proprietário) e nos AVAs Moodle e Atutor.

2.5.6. Bjeljic et al (2015)

O artigo dos autores Petar Bjeljic, Igor Zecevic e Branko Perisc com o título “*Integrating Learning Management System and Faculty Information System – Service Oriented Approach*” destaca os desafios e as abordagens para utilização de sistemas de gestão de aprendizagem e ambientes virtuais de aprendizagem.

O trabalho em destaque analisa uma proposta de arquitetura para integrar os sistemas mencionados na Faculdade de Ciências e Tecnologia de Novi Sad, na Sérvia.

Não diferente dos demais trabalhos elencados, os autores propõem a utilização de *web services* no padrão arquitetural *REST* com formato de dados *JSON*, termos a serem destacados e discutidos mais adiante, para integrar o SGA *Faculty Information System (FIS)* com o AVA *Moodle*.

2.5.7. Observações preliminares

Não é difícil perceber a importância e necessidade do processo de integração de sistemas de informações da forma como apresentados até aqui, sendo relevante destacar as técnicas a

abordagens utilizadas para realização deste processo e as diferentes visões e soluções apresentadas por diversos autores para solucionar esta problemática comum a diversas IES.

A contribuição oriunda deste trabalho perante os demais já destacados está ligada ao modelo de alto nível a ser estabelecido nos capítulos a seguir. Em sua totalidade os trabalhos relacionados anteriormente destacam-se por promoverem uma solução ponto a ponto entre um SGA e um AVA específicos de acordo com as demandas de cada IES.

A idealização do modelo para integração entre os citados sistemas visa a reprodução do comportamento de ambos a fim de produzir uma abstração com propósito de descrever e simular os processos necessários para sua integração.

Ao longo deste trabalho, pretende-se também contribuir com as discussões acerca de uma solução que seja de fácil implementação, que funcione de forma desacoplada de plataforma, possua uma infraestrutura leve e adaptável que funcionem por meio de abordagens ligadas a orientação a serviços e computação em nuvem.

Diante do exposto neste capítulo, pretende-se discutir no próximo capítulo as tecnologias e abordagens existentes para o processo de integração de sistemas a fim de demonstrar as razões da escolha da solução que será empregada neste trabalho.

3. ABORDAGENS PARA INTEGRAÇÃO

Integrar sistemas independentes é um desafio para os desenvolvedores em decorrência do fato da integração consistir em uma atividade de sincronização de dados/informações no qual é mapeado um par correspondente de cada registro nos sistemas de informações envolvidos, ou seja, um registro existente no sistema X deve ter uma cópia no sistema Y e vice-versa de maneira que haja interoperabilidade entre eles.

Laurindo e Rotondaro (2011) destacam que os chamados sistemas integrados de gestão ou ERP¹² (*Enterprise Resources Planning*) trariam a proposta de que seria possível promover esta integração das informações e do fluxo das atividades de uma só vez, desde que estes sistemas estivessem devidamente implantados. Contudo, os chamados ERP não conseguiram integrar uma grande parcela dos sistemas existentes, pois é comum organizações/instituições possuírem diversos sistemas de informações trabalhando de forma *standalone*, ou seja, sistemas que trabalham isolados e de forma independente uns dos outros, constituindo o que em alguns casos são chamados de *sistemas legados*¹³.

Segundo Hohpe e Woolf (2011), existem mais de uma abordagem para integração de sistemas, e cada abordagem destaca critérios de integração conforme cada cenário. Assim, surgiram diferentes abordagens e perspectivas que possibilitam a integração de sistemas de informação, cada uma com soluções direcionadas para cada caso que se apresente.

Na literatura, de forma ampla, é possível encontrar diversas perspectivas e abordagens de como proceder um projeto para integração de sistemas, contudo na década de 90 surgiu o conceito de EAI (*Enterprise Application Integration*) ou simplesmente Integração de Aplicações Corporativas que contempla o planejamento, ferramentas e métodos que possibilitam uma integração de todos os sistemas, unificando-os de forma a compartilharem informações.

Segundo o site *WHATIS* (www.whatis.com) o conceito de EAI diz que:

¹² ERP é um sistema de informação que integra todos os dados e processos de uma organização em um único sistema. A integração pode ser vista sob a perspectiva funcional (sistemas de finanças, contabilidade, recursos humanos, fabricação, marketing, vendas, compras etc) e sob a perspectiva sistêmica (sistema de processamento de transações, sistemas de informações gerenciais, sistemas de apoio a decisão, etc). Fonte: <https://pt.wikipedia.org/>

¹³ Termo utilizado em referência aos sistemas computacionais de uma organização que, apesar de serem bastante antigos, fornecem serviços essenciais. Geralmente utilizam bancos de dados obsoletos. Normalmente são aplicações complexas, de difícil manutenção e que pelo grau de criticidade e custo para modernização, continuam ativas.

EAI é um termo da computação de negócios para os planos, métodos e ferramentas que objetivam a modernização, consolidação e coordenação dos sistemas de informação de uma empresa. As empresas costumam ter aplicações e bases de dados legadas que desejam continuar usando, ao mesmo tempo em que querem incorporar um novo conjunto de aplicações que façam uso da Internet, comércio eletrônico, extranet e outras tecnologias modernas. A integração através de EAI consiste no desenvolvimento de uma visão totalmente nova sobre os negócios da empresa e suas aplicações, procurando encaixar as aplicações existentes na nova visão, buscando caminhos para reusar de forma eficiente o que já existe, ao mesmo tempo em que se incorpora novas aplicações de dados (2016).

Os benefícios oriundos da utilização da EAI são de permitir o compartilhamento e a integração de dados entre diferentes sistemas de informação sem ter de modificar muito as aplicações existentes, ou seja, investir em EAI é fazer os processos fluírem mais facilmente e manter todos os sistemas da empresa interconectados.

O Quadro 2, a seguir, resume alguma dessas abordagens descrevendo suas características.

Quadro 2: Abordagens de integração entre sistemas e organizações

| | |
|-------------------------------------|---|
| Orientado a Dados | Envolve o processo de extrair informações de um banco de dados para outro. O processo de manipulação dos dados necessários para estabelecer a integração pode envolver grandes conjuntos de dados oriundos de um conjunto diversificado de grandes bancos de dados e, além disso, o processo de transferência de informação pode exigir uma aplicação de regras de negócio. Na integração orientada a dados, aplicações que suportam os sistemas de informação não sofrem modificações e permanecem não integradas; |
| Orientado a Interface de Aplicações | Esta integração, por meio de interfaces de aplicações, tem como base a exposição de funções de negócio ou mecanismos de acesso a dados. Ela tem sido adotada por aplicações comerciais como os ERP. Para integrar tais sistemas, mesmo dentro da mesma organização, é necessário usar as interfaces para obter informações, colocá-las no formato da aplicação destino e posteriormente transmiti-las (para a aplicação destino); |
| Orientado a Métodos | Visa compartilhar a implementação de funcionalidades além das interfaces. Por exemplo, o código de um método para realizar determinada operação pode ser compartilhado entre as mais diversas aplicações, dentro ou fora da organização. As tecnologias que compartilham métodos podem incluir objetos distribuídos, servidores de aplicação e processamento de transações; |
| Orientado a Interface de Usuário | Ocorrem pela representação do acesso a informações, pertencentes a diferentes organizações, em um mesmo local. Na <i>Web</i> essa integração normalmente é estabelecida por um mecanismo denominado portal. Nesta abordagem, as organizações evitam a complexidade e custo da integração de aplicações provendo a integração no nível de interface do usuário. Por exemplo, na compra de uma passagem com cartão de crédito, no portal de uma companhia aérea, a autorização de compra no cartão pode ser dirigida diretamente à operadora do cartão; |

| | |
|-----------------------------------|--|
| Orientado a Processos de Negócios | Estabelece uma camada de abstração de processos no topo das aplicações e tecnologias utilizadas para a realização da ISI (Integração de Sistemas de Informação). Para as organizações envolvidas, tal camada provê uma visão de negócios e dos fluxos de informação da integração. No nível da integração de processos de negócio, os elementos da ISI são conceitos do domínio de aplicação da integração, por exemplo, pessoas, pedidos e empresas. |
| Orientado a Serviços | Semelhante à integração orientada a métodos, a integração orientado a serviços visa expor as funções de negócio definidas no contexto de uma organização por meio de serviços, ou seja, componentes de <i>software</i> . Eles são disponibilizados através de interfaces padronizadas e acessíveis a outras aplicações por meio de uma rede. Esta integração estabelece arquiteturas baseada na Arquitetura Orientada a Serviços – <i>SOA</i> , onde clientes, provedores e registros de serviços visam a busca, localização, ligação e execução dinâmica de serviços. |

Fonte: Adaptado de Costa, 2008.

Embora seja perceptível a gama de possibilidades para procedimentos de integração entre sistemas de informações, é pertinente observar, que nem todas estas abordagens seriam adequadamente aplicáveis ao problema de integração entre o SGA e o AVA, pois em algumas dessas abordagens há a necessidade de se alterar o código fonte das aplicações que podem ser de terceiros, ou seja, sistemas proprietários protegidos que garantem ao autor/produtor direitos e patentes.

Analisando este fato serão debatidos em seguida duas abordagens que podem ser aplicáveis ao problema que está sendo apresentado neste trabalho: orientado a dados e orientado a serviços.

A escolha em particular destas duas abordagens decorre do fato que segundo Kaneshima (2013) ao definir a tecnologia que deverá ser utilizada o foco é no uso de *web services* e *middleware* de banco de dados que são as tecnologias mais utilizadas atualmente pelas empresas em geral.

Os tópicos a seguir irão destacar as características das abordagens escolhidas de forma apresentar suas vantagens e desvantagens.

3.1. Integração de sistemas orientado a dados

“Há duas estratégias importantes para integrar produtos de software que não foram projetados para trabalhar em conjunto. Quando um número pequeno de produtos necessita ser integrado, os profissionais de TI podem construir interfaces entre eles. No nível mais simples, os dados podem ser extraídos de uma fonte e introduzidos em outra. Mesmo assim, alguma programação frequentemente é exigida, porque os dados extraídos necessitam ser formatados adequadamente para entrada no pacote de software de destino” Gordon e Gordon (2006, p. 239)

A citação em questão se refere à integração entre sistemas orientado a dados, na qual este tipo de integração envolve o processo de capturar dados de um banco de dados e inseri-los em outro. Mesmo parecendo simples, este procedimento exige uma modelagem das regras de negócio, além da possibilidade de envolver grande quantidade de dados e diversidade de banco de dados.

Esta abordagem é uma alternativa eficaz visto que possibilita fazer as operações necessárias em baixo nível diretamente na camada de banco de dados. Contudo exige conhecer bem a estrutura dos bancos de dados envolvidos. A implementação pode ser feita em qualquer linguagem de programação.

No caso deste tópico em particular foi implementado um experimento prático de integração utilizando abordagem de integração orientado a dados envolvendo a importação de dados de um SGA genérico¹⁴ para o AVA Moodle. O objetivo focal deste experimento é o de mensurar as vantagens e benéficos decorrentes desta abordagem de integração.

Este processo de integração terá um módulo adicional implementado na estrutura do AVA Moodle que acessará o SGA genérico a ser utilizado, sendo que este módulo será o responsável por conectar os bancos de dados e processar a replicação, ou seja, a importação de dados.

A proposta básica a ser concretizada neste experimento envolverá a importação de dados do SGA genérico conforme a estrutura apresentada no quadro 3 que se segue:

Quadro 3: Dados a serem importados

| | |
|------------------|---|
| Municípios/Polos | Nome da localidade de ensino dos discentes |
| Cursos | Descrição dos cursos em que estão matriculados os discentes. |
| Turmas | Descrição das turmas de cada curso ofertado |
| Disciplinas | Descrição das disciplinas que serão cursadas por período/semestre |
| Matrículas | Vínculo para cada discente aos cursos, turmas e disciplinas. |
| Usuários | Dados cadastrais de cada discente. |

Fonte: Elaborado pelo autor.

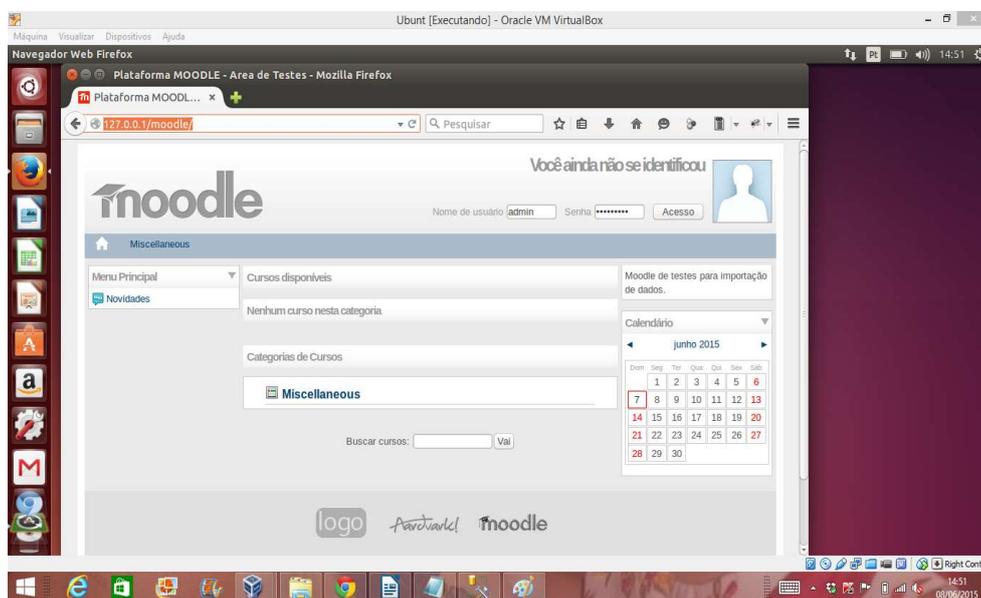
E como consequência o povoamento do banco de dados do AVA Moodle com estas informações a fim de tornar eficaz e eficiente a transmissões destas. A integração destas informações ocorre de forma sistemática de sobrescrita das informações na base de dados do AVA Moodle, sendo à base de dados do SGA genérico sempre a base das informações básicas utilizadas pelos dois

¹⁴ Um banco de dados que abrange diversos elementos comuns a um SGA, tais como: discentes, cursos, disciplinas, turmas, matrículas, campi etc.

sistemas.

Este experimento envolveu o uso do software de virtualização *VirtualBox*¹⁵ instalado sobre o sistema operacional hospedeiro *Windows 8.1*. No *VirtualBox* foi instalado e configurado o sistema operacional *Ubuntu 14.04*¹⁶ e dentro deste foi instalado e configurado o AVA Moodle. Fez-se necessário a configuração de uma rede virtual entre o sistema operacional convidado (*Ubuntu 14.04*) e o host (*Windows 8.1*) para que houvesse a comunicação necessária para troca de informações que ocorreria entre os dois sistemas conforme figura 5:

Figura 5: AVA Moodle instalado no Ubuntu 14.04



Fonte: Elaborado pelo autor.

No sistema anfitrião, no caso o *Windows 8.1*, está instalado o banco de dados *MS SQL Server 2014*¹⁷ que abriga o banco de dados a ser utilizado como SGA genérico. Neste banco há tabelas correspondentes a dados cadastrais de usuários, cursos, disciplinas, turmas e matrículas que serão parte do processo de importação dos dados a ser concretizado.

A consulta ao banco de dados *MS SQL Server 2014* destacada na figura 6 demonstra os dados

¹⁵ Software que visa criar ambientes para instalação de sistemas distintos. Ele permite a instalação e utilização de um sistema operacional dentro de outro, assim como seus respectivos softwares, como dois ou mais computadores independentes, mas compartilhando fisicamente o mesmo hardware.

¹⁶ Sistema Operacional de código aberto baseado na distribuição Debian GNU/Linux.

¹⁷ É um SGBD - sistema gerenciador de Banco de dados relacional desenvolvido pela Microsoft. Como um Banco de dados, é um produto de software cuja principal função é a de armazenar e recuperar dados solicitados por outras aplicações de software

a serem importados para o banco de dados do AVA Moodle.

Figura 06: Consulta de dados a serem importados do MS SQL Server 2014

```

SQLQuery1.sql - ATHOS.BDGA (sa (56))* X
select distinct matri AS MATRICULA, dadopnom AS NOME, cidadnom AS CIDADE,
cursonom AS CURSO, polocid AS POLO, discinom AS DISCIPLINA,
crdblo AS BLOCO from dadoc
inner join dadop on dadopdadoc=dadop
inner join curso on dcucur=curso
inner join polo on dadocpolo=polo
inner join matri on dadocmatri=matri
inner join disci on matriidis=disci
inner join cidad on dadopcid=cidad
inner join crd on crdcur=curso
where matriper=20132 and curso in (73) and crdblo=1
order by cursonom,polocid, dadopnom
order by cursonom,polocid, dadopnom

```

Fonte: Elaborado pelo autor.

Esta consulta remete aos dados de um hipotético curso técnico em informática do polo situado na cidade de Piracuruca – PI, e estão sendo consultadas informações correspondentes respectivamente ao número de matrícula do discente, seu nome completo, a cidade de nascimento, o curso ao qual está vinculado, o polo onde está matriculado, o nome das disciplinas que serão ministradas e o bloco correspondente ao período/semestre letivo de 2013.2.

Do lado do AVA Moodle é importante destacar que o banco de dados que será utilizado é o MySQL¹⁸ versão 5.5.43, ou seja, este experimento faz uso de banco de dados diferentes com o propósito de provar, conforme mencionado anteriormente, que o processo de integração de sistemas independe de linguagem de programação e que pode envolver uma diversidade de bancos de dados.

Para povoar o MySQL é necessário mapear as tabelas que serão preenchidas pelos dados constantes na consulta demonstrada no quadro 4 conforme descrito a seguir:

Quadro 4: Tabelas a serem preenchidas no MySQL e dados a serem importados.

| Tabelas MySQL a preencher | Dados do SQL Server a serem importados |
|---------------------------|---|
| mdl_user | Número de matrícula, nome do discente, cidade de origem |
| mdl_course | Disciplinas |
| mdl_course_categories | Polo, curso e bloco/Período |
| mdl_role_assignments | Matrícula nos cursos |

Fonte: Elaborado pelo autor.

¹⁸ MySQL é um SGBD (sistema de gerenciamento de banco de dados) de código aberto que faz uso de linguagem SQL (Linguagem de consulta estruturada).

A solução de importação deste experimento estabelece um conjunto de ações com base em consultas SQL¹⁹ que deverão realizar as seguintes ações conforme quadro 5:

Quadro 5: Ações a serem realizadas para importação de dados entre os sistemas.

| SGA – GENÉRICO | AÇÕES | AVA – MOODLE |
|---|-------|---|
| - Cadastro de Polos (Municípios em que são ofertados os cursos); | > | - Cria uma categoria de primeiro nível com o nome do Polo. |
| - Cadastro de um Curso (Cursos em EAD a ser ofertado nos polos); | > | - Cria uma categoria de segundo nível com o nome do Curso. |
| - Cadastro de Bloco/Período (Momento letivo em que ocorre os cursos); | > | - Cria uma categoria de terceiro nível com o valor do Bloco/Período. |
| - Cadastro de Disciplina (No AVA uma disciplina equivaleria a um curso); | > | - Cria uma disciplina, onde serão inseridas as atividades padrões para a disciplina. |
| - Matricular Discente; | > | - Adiciona o discente. - Matricula o discente em todas as Disciplinas (cursos) que irá cursar |
| - Vincula Docente/Tutor a Turma e Disciplina (Definição de Docente/tutores); | > | - Adiciona o Docente na disciplina. - Adiciona o tutor(es) na disciplina. |

Fonte: Elaborado pelo autor.

Para efetuar o processo de integração entre os sistemas supracitados um módulo adicional, que funciona como interface, foi desenvolvido e adicionado dentro do AVA *Moodle*, aproveitando sua arquitetura modular, característica elencada no quadro 1 do tópico 2.3, com o propósito de estabelecer a conexão entre os bancos de dados *MS SQL Server* e *MySQL* e agrupar as funções básicas de operacionalização dos dados envolvidos.

O módulo de integração foi desenvolvido em linguagem PHP²⁰ e *JavaScript*²¹. As funções existentes no módulo permitem conectar o banco de dados *SQL Server* e dele obter os dados necessários a serem importados para o banco de dados *MySQL* no AVA *Moodle*.

Dentre estes dados constam o nome dos cursos, os períodos letivos em que ocorrem estes

¹⁹ Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional). Muitas das características originais do SQL foram inspiradas na álgebra relacional.

²⁰ PHP (um acrônimo recursivo para "PHP: Hypertext Preprocessor", originalmente Personal Home Page) é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico.

²¹ JavaScript é uma linguagem de programação interpretada. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor.

cursos e os polos onde estes estão sendo ofertados conforme mostra a figura 7.

Figura 7: Cursos, períodos e polos disponíveis para importação.

The screenshot shows a web browser window with the URL `127.0.0.1/moodle/modulo/2/pagina=home`. The page is titled "Módulo de integração ..." and contains a "Menu" on the left with options: Opcoes, Home, Cadastrar Usuario, Zerar senhas, Inserir Tutor(Varios Cursos), and Migrar Turma. The main content area is titled "Funcionalidades Disponíveis" and features a warning box: "AVISO! O BOTAO ABAIXO IMPORTARÁ AS DISCIPLINAS DO SISTEMA ACADEMICO PARA O AMBIENTE VIRTUAL DE APRENDIZAGEM - MOODLE OFERTAR AS TURMAS NO SISTEMA ACADEMICO ANTES DE IMPORTAR AS DISCIPLINAS". Below the warning, there are form fields for "Curso" (set to "TEC EM INFORMATICA"), "Periodo" (set to "20132"), and "Escolha os Polos:" with radio buttons for "PIRIPIRI - PI", "CORRENTE - PI", "CAMPO MAIOR - PI", "OEIRAS - PI", and "PIRACURUCA - PI" (which is selected). An "IMPORTAR" button is located at the bottom right.

Fonte: Elaborado pelo autor

Conforme a imagem acima é possível escolher entre os cursos técnicos existentes (no caso em questão curso Técnico em Informática), o mesmo consta no catálogo de cursos do banco de dados *MS SQL Server* assim como o período/semestre letivo e os polos para os quais o curso pode ser ofertado.

A funcionalidade em destaque importará o curso, período e o polo para o banco de dados *MySQL* do *AVA Moodle* quando clicado o botão IMPORTAR no canto inferior direito da figura. No exemplo em discussão foi importado o curso técnico em informática do período letivo de 2013.2 do polo de Piracuruca – PI. O resultado desta operação pode ser visto na figura 8 onde percebe-se a construção das categorias de cursos dentro do *AVA Moodle*.

O polo Piracuruca – PI foi importado para uma categoria de primeiro nível. Uma subcategoria de segundo nível chamada Tec. em Informática foi importada para a categoria do polo no qual o curso é ofertado e na sequência uma subcategoria de terceiro nível intitulada bloco também foi importada e faz referência direta ao curso anteriormente importado.

Figura 8: Dados importados para o Moodle.

The screenshot shows the Moodle course management interface. The breadcrumb navigation is 'Administração > Cursos > Acrescentar/modificar cursos'. The page title is 'Miscellaneous PIRACURUCA - PI'. There are buttons for 'Habilitar Edição de Bloco' and 'Desativar edição'. The left sidebar shows the site administration menu with 'Cursos' expanded. The main content area is titled 'Categorias de Cursos' and contains a table with the following data:

| Categorias de Cursos | Cursos | Editar | Mova a categoria para: |
|----------------------|--------|---------------------------|--------------------------------------|
| Miscellaneous | 0 | [Edit] [Delete] [Refresh] | Início |
| PIRACURUCA - PI | 0 | [Edit] [Delete] [Refresh] | Início |
| TEC EM INFORMATICA | 0 | [Edit] [Delete] [Refresh] | PIRACURUCA - PI |
| Bloco V | 7 | [Edit] [Delete] [Refresh] | PIRACURUCA - PI / TEC EM INFORMATICA |

Fonte: Elaborado pelo autor.

Nesta categoria bloco é possível identificar a quantidade de disciplinas que foram importadas segundo o período/semestre letivo conforme figura 9:

Figura 9: Disciplinas importadas para o Moodle.

The screenshot shows the Moodle course management interface with the breadcrumb navigation 'Administração > Cursos > Acrescentar/modificar cursos'. The page title is 'Miscellaneous PIRACURUCA - PI'. There are buttons for 'Habilitar Edição de Bloco' and 'Desativar edição'. The left sidebar shows the site administration menu with 'Cursos' expanded. The main content area is titled 'Categorias de Cursos' and shows the selected category 'PIRACURUCA - PI / TEC EM INFORMATICA / Bloco V'. There are buttons for 'Editar categoria' and 'Acrescentar sub-categoria'. Below this is a table with the following data:

| Cursos | Editar | Selecionar |
|---|---------------------------|--------------------------|
| FUNDAMENTOS DE COBIT-3373501-20132 | [Edit] [Delete] [Refresh] | <input type="checkbox"/> |
| FUNDAMENTOS DE GOVERNANCA DE TI-3373501-20132 | [Edit] [Delete] [Refresh] | <input type="checkbox"/> |
| APLICACOES WINDOWS MOBILE-3373501-20132 | [Edit] [Delete] [Refresh] | <input type="checkbox"/> |
| APLICACOES JAVA MOBILE-3373501-20132 | [Edit] [Delete] [Refresh] | <input type="checkbox"/> |
| FUNDAMENTOS DAS APLICACOES MOBILE-3373501-20132 | [Edit] [Delete] [Refresh] | <input type="checkbox"/> |
| ILUSTRACAO E DESIGNER GRAFICO-3373501-20132 | [Edit] [Delete] [Refresh] | <input type="checkbox"/> |
| MODELAGEM DE DADOS-3373501-20132 | [Edit] [Delete] [Refresh] | <input type="checkbox"/> |

At the bottom of the table, there is a button 'Mover cursos selecionados para...'.

Fonte: Elaborado pelo autor.

Ao clicar em qualquer das disciplinas e acessar o módulo de participantes constante na disciplina é possível também perceber que todos os discentes que foram matriculados no curso tec. em informática do polo de Piracuruca-PI foram importados conforme figura 10.

Figura 10: Importação dos discentes em suas respectivas disciplinas.

Lista de usuários: [Menos detalhes](#)

Papel: Student

Todos os participantes: 30

(A inscrição de usuários é cancelada automaticamente se a conta não for usada num período de 120 dias consecutivos)

Nome : Todos A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Sobrenome : Todos A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Página: 1 2 (Próximo)

| Foto do usuário | Nome / Sobrenome | Cidade/Município | País | Último acesso | Selecionar |
|-----------------|-------------------------------|----------------------|--------|---------------|--------------------------|
| | ANA CRISTINA FROTA OLIVEIRA | PIRACURUCA-PI | Brasil | Nunca | <input type="checkbox"/> |
| | NATALICE CERQUEIRA SOUSA | SÃO JOSÉ DO DIVINO-P | Brasil | Nunca | <input type="checkbox"/> |
| | CARLA ANDRÉIA DA SILVA | BATALHA-PI | Brasil | Nunca | <input type="checkbox"/> |
| | MARIA DO CARMO SOUSA MEDEIROS | SÃO JOSÉ DO DIVINO-P | Brasil | Nunca | <input type="checkbox"/> |
| | RAIMUNDA PEREIRA DA SILVA | PIRACURUCA-PI | Brasil | Nunca | <input type="checkbox"/> |
| | FRANCIELLE DE CASTRO SILVA | PIRACURUCA-PI | Brasil | Nunca | <input type="checkbox"/> |

Fonte: Elaborado pelo autor.

Outra funcionalidade importante e disponível no módulo de importação existente no AVA *Moodle* inclui o recurso de cadastramento do docente/tutor nas disciplinas ofertadas na qual é possível determinar o polo, o período e a disciplina em que ele atuará.

Outras funções também disponíveis e implementadas no módulo desenvolvido proveem a possibilidade de zerar as senhas de acessos dos usuários (docentes, discentes, tutores) do AVA *Moodle* e a função para importar disciplinas aleatórias de qualquer dos cursos existentes no catálogo do SGA genérico para serem ofertadas para o período/semestre letivo vigente.

Neste experimento foram abordados os recursos e funcionalidades básicos para importação de dados entre o SGA e o AVA, sendo o propósito deste detalhar aspectos desta abordagem e apresentar suas características.

O nível de integração obtido por meio da orientação de dados tem como vantagem ser prático e funcional, contudo este experimento detalha características não desejáveis para solução proposta nesta pesquisa:

- É fortemente acoplada: Os sistemas envolvidos possuem forte dependência entre si, o que vai de encontro a um dos princípios de SOA, no qual os serviços precisam ser autônomos.

- Não é flexível: O experimento é adequado e prático para os SGA genérico e ao AVA *Moodle* não sendo maleável para outras demandas que não este eles. Solução encontrada na maioria dos trabalhos relacionados destacados anteriormente.
- Intrincada implementação: Faz-se necessário é compreender detalhadamente os esquemas dos bancos de dados de origem e de destino, com objetivo de selecionar quais dados irão se deslocar de um banco para o outro.

Nestes termos a abordagem por banco de dados não pode ser utilizada com solução para o problema a ser equacionado neste trabalho. A seguir, será debatido a abordagem de integração de sistemas orientado a serviços.

3.2. Integração de sistemas orientado a serviços

Conforme destacado no tópico 2.4 do capítulo de revisão de literatura, *web services* despontam como principal meio para integração de sistemas entre aplicações heterogêneas. Segundo Marzullo (2009) *web services* provê uma infraestrutura leve e desacoplada de plataforma e de fácil implementação, o que facilita a integração de diferentes tecnologias, melhorando a forma como aplicações distribuídas são desenvolvidas e recurso corporativos compartilhados na Internet.

O paradigma de *web services* abre novos caminhos para a engenharia de software porque oferece um modelo alternativo de desenvolvimento ágil e eficiente, melhorando a forma com que integramos dinamicamente sistemas e serviços corporativos e aumentamos a capacidade de criar novos processos de negócio a partir dessa integração. Marzullo (2009, p. 151)

Os componentes *web services* possibilitam às aplicações enviar e receber dados, por exemplo, em formato *XML*²². Por exemplo, a aplicação *GoogleMaps*, mantida pelo *Google*, disponibiliza *web service* que permite localizar cidades, endereços e bairros. Caso um desenvolvedor deseje localizar endereços de seus clientes em suas aplicações, por exemplo, poderá consumir essas informações diretamente do *web service* do *Google*, independente da linguagem de programação utilizada pela aplicação: *Java*, *C#*, *Perl*, *Ruby*, *Microsoft .NET*, *Visual Studio*, *PHP*, *Python*, etc.

O *web service* em questão fornece as informações necessárias no formato *XML* e recebe a resposta também no mesmo formato por meio do protocolo *HTTP* conforme destacado na figura 11.

²² XML é a sigla para Extensible Markup Language, que significa em português Linguagem Extensível de Marcação Genérica. É uma recomendação para gerar linguagens de marcação para necessidades especiais, tais como textos, banco de dados ou desenhos vetoriais. A linguagem XML é classificada como extensível porque permite definir os elementos de marcação.

Figura 11 – Funcionamento de um *Web service*.



Fonte: Adaptado pelo autor

Em vista dessas funcionalidades, conclui-se que *web services* proporcionam uma ferramenta viável para cursos de modalidade EAD, onde, em um cenário ideal de funcionamento, SGA e AVA interagem constantemente em uma escala de centenas ou milhares de usuários. Neste tópico, serão discutidos os principais conceitos e tecnologias empregadas para a implementação de *web services*, e discutido uma possibilidade de uso no problema, enfrentado em muitas IES, de prover um intercâmbio automático de dados entre sistemas de SGA e AVA.

3.2.1 Formato de dados

Conforme comentado na seção anterior, o formato de dados *XML* apresenta-se como forma tradicional de solicitar e receber informações por meio de *web services*, contudo há outros formatos (*HTML*, *TXT*) que podem representar dados de forma estruturada e aninhada; o *JSON*²³ é um deles. A presente seção apresenta os dois formatos.

3.2.1.1 O formato *XML*

O formato *XML* é recomendado pela W3C²⁴ para criar documento com dados organizados de forma hierárquica. Estão entre suas principais características o fato de fornecer uma sintaxe básica que pode ser utilizada para compartilhar informações entre diferentes arquiteturas e aplicações.

Quando aliado com outros padrões, torna possível definir o conteúdo de um documento separadamente de seu formato, tornando simples para reutilizar o código em outras aplicações para

²³ JSON – JavaScript Object Notation (Notação de Objetos JavaScript).

²⁴ O World Wide Web Consortium (W3C) é a principal organização de padronização da World Wide Web. Consiste em um consórcio internacional com quase 400 membros, agregando empresas, órgãos governamentais e organizações independentes com a finalidade de estabelecer padrões para a criação e a interpretação de conteúdos para a Web.

diferentes propósitos. Por outro lado, o uso de *XML* pode representar um *overhead*²⁵ de espaço, em razão do uso sistemático de marcadores. Citando Mitchell (2013):

Qualquer pessoa que já tenha passado um bom tempo na Internet entenderá o estilo “cheio de sinais de maior e de menor” do *XML* e será capaz de lê-lo. O *XML* é um formato bem prolixo; a pontuação adicional e o escopo para atributos, os dados de caracteres e as tags aninhadas podem gerar dados um pouco mais extensos do que gerados por outros formatos. Mitchell (2013, p. 61)

Considere o exemplo que se segue da representação de dados em *XML* por meio de uma lista de cidades do Estado do Piauí: Piripiri, Corrente, Campo Maior, Oeiras, Piracuruca. Uma possível representação em *XML* seria:

Figura 12 – Representação de dados em XML.

```

1 <?xml version="1.0"?>
2 <lista>
3 <item>Piripiri</item>
4 <item>Corrente</item>
5 <item>Campo Maior</item>
6 <item>Oeiras</item>
7 <item>Piracuruca</item>
8 </lista>

```

Fonte: Elaborado pelo autor

Outra de suas principais características é a maneira de representar dados de forma bastante detalhadas, pois é possível incluir informações sobre os tipos de dados com os quais se está trabalhando, não só isso, mas com tipos personalizados de dados.

O *XML* é bastante difundido no meio dos desenvolvedores e por diversas plataformas tecnológicas tais como *Oracle*, *.NET* e *Java*, e, portanto, é o formato preferido de trabalho. Contudo, quanto ao volume de dados a serem transferidos neste formato, dependendo dos dispositivos, da largura de banda e do formato mais extenso aplicado aos dados, pode apresentar problemas significativos.

3.2.1.2 O formato *JSON*

Este formato de dados, que pode ser escrito e lido facilmente por diversas linguagens, além

²⁵ *Overhead*: Qualquer processamento ou armazenamento em excesso, seja de tempo de computação, de memória, de largura de banda ou qualquer outro recurso que seja requerido para ser utilizado ou gasto para executar uma determinada tarefa. Como consequência pode piorar o desempenho do aparelho que sofreu o overhead.

de JavaScript²⁶, é caracterizado por ser um formato leve, bastante simples e capaz de representar dados estruturados e aninhados.

Tomando o exemplo anterior utilizado no formato *XML* da lista de cidades do Estado do Piauí, uma representação em *JSON* poderia ser:

```
{“Lista”: [“Piripiri”, “Corrente”, “Campo Maior”, “Oeiras”, “Piracuruca”]}
```

Múltiplos valores podem ser agrupados entre colchetes e as chaves e os valores são separados por dois pontos e cada registro é separado por vírgula. Esse exemplo mostra que os dados listados possuem um valor/chave chamado “Lista”. É possível representar estes mesmos dados de uma forma bastante simplificada:

```
[“Piripiri”, “Corrente”, “Campo Maior”, “Oeiras”, “Piracuruca”]
```

A característica mais positiva do *JSON* está no fato de ser um formato de dados simples e que não exige muito espaço se comparado ao *XML*, assim como não é complicada sua transferência por meio de conexões lentas, sendo ótimo para uso de aplicações *mobile*, pois, seu custo de decodificação é baixo.

O seu uso é recomendado quando as informações sobre o formato de dados não forem essenciais, uma vez que ele não fornece informações específicas sobre os tipos de dados que estão sendo usados.

3.2.2. Padrões para desenvolvimento de *web services*

Os conceitos aqui explorados visam auxiliar na definição da escolha da arquitetura de software que será utilizada para desenvolver *web service*, pois esta escolha definirá todos os componentes que devem ser utilizados na concepção do projeto de integração de sistemas heterogêneos.

Atualmente existem dois padrões de destaque para o desenvolvimento de *web services*. O padrão *SOAP* e o padrão *REST* ou *RESTful*.

²⁶ JavaScript: Linguagem de programação interpretada originalmente implementada para navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com usuário

3.2.2.1. Web services SOAP

Originalmente *SOAP* significava *Simple Object Access Protocol* (Protocolo de Acesso a Objetos), mas nos dias atuais essa descrição não é mais válida. Segundo Gomes (2014) *SOAP* é o protocolo padrão para transmissão de dados dentro da arquitetura de *web services* proposta pelo W3C. O *SOAP* é um protocolo baseado no *XML* e segue o modelo “*Request-Response*” (solicitação – resposta) do HTTP.

Por padrão, a troca de informação de tecnologias baseadas em *XML SOAP* é descrita pela *WSDL*²⁷. A *WSDL* descreve a localização de um serviço em particular, os tipos de dados, valores, métodos e parâmetros utilizados.

Uma questão a destacar refere-se ao protocolo sobre o qual trafegam as mensagens *SOAP*, embora prioritariamente estas utilizem o *HTTP*, a arquitetura proposta pelo W3C especifica que as respostas e solicitações *XML* possam trafegar por outros protocolos como *FTP*, *TCP*, *SMP*, etc.

Como mencionado no item 3.3.1.1, referente ao formato *XML*, a pontuação adicional e o escopo para atributos, os dados de caracteres e as *tags* aninhadas podem gerar volumes de dados mais extensos do que gerados por outros formatos. Isso ocorre devido ao fato de existirem uma série de normatizações denominados *Web Services Standards*.

O *web services Standards* define especificações de, por exemplo, segurança conforme mostra a figura 13. Cada padrão estabelece um conjunto de normativas em *XML* de como se deve desenvolver para o padrão *SOAP*, ou seja, caso o desenvolvedor esteja utilizando o padrão *XML SOAP* para desenvolver sua aplicação *web service* e necessite implementar as especificações de segurança, deverá consultar o *web service Standards* na seção *Security Specifications* para estar em acordo com a forma com a qual deve trabalhar utilizando este padrão.

²⁷ WSDL - *Web Service Description Language* (linguagem para descrição de web service). É um arquivo do tipo XML, cuja finalidade é descrever detalhadamente um web service. Essa descrição especifica as operações que compõem o web service e define de forma clara como deve ser o formato de entrada e saída de cada operação.

Figura 13: Web Services Standards

► Security Specifications

WS-Security
1.1
OASIS
OASIS-Standard

▲ **WS-Security** is a communications protocol providing a means for applying security to Web Services.

WS-SecurityPolicy
1.1
IBM, Microsoft,
RSA Security, VeriSign
Public Draft

▲ **WS-SecurityPolicy** defines how to describe policies related to various features defined in the WS-Security specification.

**WS-Security:
SOAP Message Security**
1.1
OASIS
Public Review Draft

▲ **WS-Security: SOAP Message Security** describes enhancements to SOAP messaging to provide message integrity and confidentiality. Specifically, this specification provides support for multiple security token formats, trust domains, signature formats and encryption technologies. The token formats and semantics for using these are defined in the associated profile documents.

**WS-Security:
Username Token Profile**
1.1
OASIS
Public Review Draft

▲ **WS-Security: Username Token Profile** describes how a Web Service consumer can supply a Username Token as a means of identifying the requestor by username, and optionally using a password (or shared secret, etc.) to authenticate that identity to the Web Service producer.

**WS-Security:
Kerberos Binding**
1.0
Microsoft, IBM, OASIS
Working Draft

▲ **WS-Security: Kerberos Binding** defines how to encode Kerberos tickets and attach them to SOAP messages. As well, it specifies how to add signatures and encryption to the SOAP message, in accordance with WS-Security, which uses and references the Kerberos tokens.

WS-Federation
1.0
IBM, Microsoft, BEA Systems,
RSA Security, and VeriSign
Initial Draft

▲ **WS-Federation** describes how to manage and broker the trust relationships in a heterogeneous federated environment including support for federated identities.

**WS-Security:
SAML Token Profile**
1.1
OASIS
Public Review Draft

▲ **WS-Security: SAML Token Profile** defines the use of Security Assertion Markup Language (SAML) v1.1 assertions in the context of WSS: SOAP Message Security including for the purpose of securing SOAP messages and SOAP message exchanges.

WS-Trust
BEA Systems, Computer Associates, IBM, Layer 7
Technologies, Microsoft, Netegrity, Oblix,
OpenNetwork, Ping Identity Corporation,
Reactivity, RSA Security, VeriSign and Westbridge
Technology · Initial Draft

▲ **WS-Trust** describes a framework for trust models that enables Web Services to securely interoperate. It uses WS-Security base mechanisms and defines additional primitives and extensions for security token exchange to enable the issuance and dissemination of credentials within different trust domains.

**WS-Security: X.509
Certificate Token Profile**
1.1
OASIS
Public Review Draft

▲ **WS-Security: X.509 Certificate Token Profile** describes the use of the X.509 authentication framework with the WS-Security: SOAP Message Security specification.

WS-SecureConversation
BEA Systems, Computer Associates, IBM,
Layer 7 Technologies, Microsoft, Netegrity,
Oblix, OpenNetwork,
Ping Identity Corporation, Reactivity,
RSA Security, VeriSign and Westbridge
Technology · Public Draft

▲ **WS-SecureConversation** specifies how to manage and authenticate message exchanges between parties including security context exchange and establishing and deriving session keys.

O *web services Standards* define especificações não apenas de segurança, mas também de especificações de mensagens (*Messaging Specifications*), especificações de metadados (*Metadata Specifications*), especificações de processos de negócios (*Business Process Specifications*), especificações de gerenciamento (*Management Specifications*), problemas de interoperabilidade (*Interoperability Issues*), etc.

O desenvolvedor de posse do arquivo *WSDL* poderá criar uma aplicação cliente que efetuará uma chamada ao provedor web service que conseguirá obter resposta conforme o serviço que foi solicitado. A figura 14 destaca como deve ocorrer este procedimento.

Figura 14: Funcionamento do XML SOAP.



Fonte: Adaptado pelo autor.

As solicitações do cliente serão encaminhadas em formato *XML*; por sua vez, o provedor web service receberá a solicitação e produzirá uma resposta no mesmo formato. O *WSDL* funciona como um descriptor cujo objetivo é especificar o formato de entrada e saída de cada operação realizada de acordo com as especificações *web service Standards*.

Sobre o formato da solicitação e resposta (*Request – Response*) do formato XML entre o cliente e o provedor *web service*, ambos se apresentam da seguinte forma:

Figura 15: Solicitação em XML.

```

POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>

</soap:Envelope>

```

Fonte: http://www.w3schools.com/webservices/ws_soap_example.asp

Figura 16: Resposta em XML.

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>

</soap:Envelope>

```

Fonte: http://www.w3schools.com/webservices/ws_soap_example.asp

Com o formato *XML* no padrão *SOAP* todo o código das solicitações e respostas estão bem mapeados, ou seja, este padrão de arquitetura é muito completo e robusto. Contudo, como já dissemos, ele tem um custo, pois enquanto é gerado um pedido do cliente que será enviado ao servidor *web service SOAP*, este responde “empacotando” em um envelope a resposta de acordo com os padrões estabelecidos pelo descritor (*WSDL*). Isso pode tornar muito lento o tempo de resposta e aumentando o consumo de banda se comparado com outro padrão de arquitetura como *REST*.

3.2.2.2 Web services REST ou RESTful

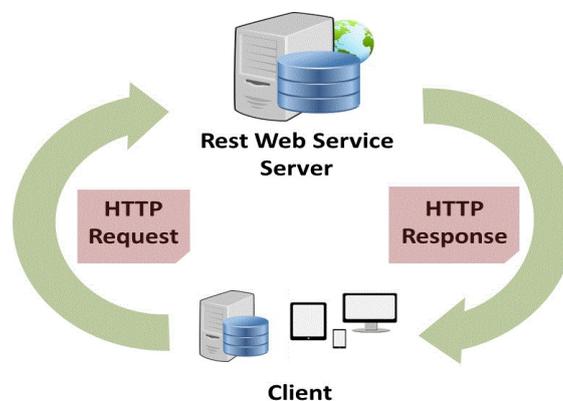
O padrão arquitetural *REST* é acrônimo de *REpresentational State Transfer* (Transferência de Estado Representacional). Não é um protocolo, mas sim um conjunto de boas práticas na modelagem de web services estabelecidas por Roy Fielding (<http://roy.gbiv.com>) para sua tese de PhD, que descreve todas as diretrizes do estilo REST.

A tese de Fielding está disponível para consulta no seguinte endereço eletrônico: (<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>). O princípio básico do *REST* é possibilitar o desenvolvimento de *web services* de boa performance, que sejam simples, e que permitam trabalhar com informações via HTTP.

Ao contrário do padrão arquitetural fixo *XML SOAP*, o *REST* pode ter uma representação de formato por XML, JSON ou qualquer outra opção que represente um formato de dados. De fato, ao caracterizar o padrão *REST*, este é bastante similar se comparado em termos de finalidade ao padrão *SOAP*; contudo, o elemento diferencial entre ambos está no fato do padrão *REST* não trazer consigo a carga estabelecida no padrão *SOAP* por meio de seu *Web Services Standards* e *WSDL* – o mais próximo disso que há em REST é o *WADL* (*Web Application Description Language*) Linguagem de descrição de aplicação *web*,) é um vocabulário XML utilizado para descrever os serviços web REST.

Segundo Richardson e Ruby (2007), como no *WSDL*, um cliente genérico pode carregar um arquivo *WADL* e ter uma permissão imediata para acessar a funcionalidade completa do *web service* correspondente. Contudo, *WADL* é mais falado do que utilizado, pois com base no princípio da simplicidade na qual opera o padrão REST, ele funcionaria conforme descrito na figura a seguir:

Figura 17 – Funcionamento do REST.



Fonte: http://www.chemaxon.com/app/themes/chemaxon/images/product_pages/jws/rest.jpg

Sem a necessidade de estabelecer descrições existentes no padrão *XML SOAP* por meio do *WSDL*, *web services REST* se tornam simples de serem desenvolvidos. As características citadas anteriormente definem o modelo de arquitetura REST e caso uma aplicação siga todas as funcionalidades e definições deste modelo é chamada de *RESTful*.

Entre as características-chave do *REST* pode-se também destacar o fato de ser um modelo Cliente-Servidor sem estado (*stateless*), na qual cada mensagem HTTP contém toda a informação necessária para compreender o pedido e em que grande parte ou todo o código é executado em cima do Servidor, minimizando problemas na aplicação oriunda da parte do Cliente.

Entre as vantagens do *REST* podem ser destacadas:

1. O tempo reduzido de aprendizagem para implantar o modelo REST, assim como sua baixa complexidade de codificação favorecem a redução do tempo necessário entre a construção da aplicação e a sua disponibilização;
2. O consumo de rede e processamento de informações/dados são menores se comparados com o padrão *XML SOAP*;
3. A implementação da arquitetura *REST* pode ocorrer com qualquer linguagem que tenha suporte ao protocolo HTTP;
4. Arquitetura *REST* fornece serviços independentemente da plataforma e tecnologia, garantido interoperabilidade.

Entre suas desvantagens podem ser destacadas:

1. Em se tratando de aplicações críticas no qual o desempenho seja um requisito, a arquitetura *REST* pode não atender satisfatoriamente;
2. A ausência de um descritor para especificar o formato de entrada e saída de cada operação realizada para auxiliar desenvolvedores a criarem suas aplicações, tal qual como ocorre com o descritor *WSDL* no padrão *SOAP*.

Em resumo o *REST* é um modelo arquitetural prático capaz de atender requisitos de integração sem agregar complexidade, por meio de boas práticas de programação aliado a um modelo arquitetural simplista e robusto que reduz consideravelmente o tempo necessário para construção de aplicações.

3.2.3. Integração de sistemas no meio acadêmico na modalidade EAD

Para justificar a escolha do padrão a ser adotado no desenvolvimento da solução deste trabalho, será utilizada a mesma referência de informações anteriormente utilizada no experimento de orientado a banco de dados no quadro 3 a fim de ilustrar o caso em questão:

Quadro 6: Dados a serem importados para AVA

| | |
|------------------|---|
| Municípios/Polos | Nome da localidade de ensino dos discentes |
| Cursos | Descrição dos cursos em que estão matriculados os discentes. |
| Turmas | Descrição das turmas de cada curso ofertado |
| Disciplinas | Descrição das disciplinas que serão cursadas por período/semestre |
| Matrículas | Vínculo para cada discente aos cursos, turmas e disciplinas. |
| Usuários | Dados cadastrais de cada discente. |

Fonte: Elaborado pelo autor

Da mesma forma os dados que devem ser importados do AVA para o SGA, por exemplo: as médias aritméticas das notas obtidas pelos discentes no AVA devem compor um processo sincronizado entre ambos os sistemas, assim toda informação que precise ser replicada no SGA para o AVA ocorrerá de forma automática e também de forma contrária, do sistema AVA para o SGA.

Considerando um cenário no qual os cursos na modalidade de EAD possuam uma estrutura onde cursos ofertados podem ser compostos por várias turmas, estas por sua vez possuirão diversas disciplinas nas quais deverá existir um registro de cada discente curricularmente matriculado por período/semestre nos diversos municípios/polos em que estarão sendo ofertados cada um destes cursos, hierarquicamente deverá ser criado no AVA uma estrutura que comporte, por exemplo, a seguinte situação:

Município/Polo
 Cursos
 Turma 1
 Disciplina 1
 Disciplina 2
 Disciplina n ...
 Turma n...
 Disciplina 1
 Disciplina 2
 Disciplina n ...

Para detalhar claramente, cada discente curricularmente matriculado fará parte de uma turma na qual estará ligado a um conjunto de disciplinas. Em uma situação hipotética supondo 15.000 discentes, cada um destes matriculados em 06 disciplinas por período/semestre, haveriam cerca de 90.0000 registros a serem importados do SGA para o AVA.

3.2.4. Comparação de abordagens

Conforme destacado anteriormente sobre os padrões *SOAP/REST*, ambos possuem vantagens e desvantagens a serem consideradas para o processo de integração de sistemas.

Na problemática apresentada no tópico 3.3.3 tem-se um volume considerável de registros a serem importados. O quadro abaixo apresenta a diferença entre a utilização de cada um dos padrões no desenvolvimento da solução que visa integrar os sistemas descritos.

Quadro 7: Resumo de padrões

| PADRÃO | SOAP | REST |
|-----------------------|--|--|
| Transporte de dados | Complexo, pois necessita empacotar em um envelope os dados de acordo com o padrão. | Simplificado, pois não possui um padrão determinado para os dados. |
| Tipo de dados | XML | Qualquer tipo: XML, JSON ou qualquer outra opção. |
| Modelo de Arquitetura | É um protocolo reconhecido pela W3C. | Conjunto de boas práticas na modelagem de web services. |
| Funcionalidades | Possui diversas previstas no Web Service Standards. | Determina o básico dos serviços. |

Fonte: Adaptado de: <http://www.devmedia.com.br/web-services-rest-versus-soap/32451>

Ao considerar o padrão *SOAP* para integração dos sistemas deve se levar em conta o volume de dados a serem importados, pois como o mesmo está atrelado a um rígido e pouco flexível modelo de arquitetura, pode-se comprometer o desempenho da estrutura de rede tornando-a muito lenta em decorrência do tempo de resposta para processar os registros a serem importados.

O padrão *REST* por outro lado apresenta condições mais favoráveis para resolução da problemática apresentada devido ao fato de ser mais ágil e leve em virtude de não seguir um modelo de arquitetura rígido, no qual é possível determinar o melhor tipo de dados para a situação descrita anteriormente e por se basear em um conjunto de boas práticas para construção de *web services*.

Este subitem em particular, 3.2 (Integração de sistemas orientado a serviços), objetivou discutir as características presentes nos padrões de arquiteturas *SOAP/REST* para auxiliar a escolha sobre qual padrão adotar no desenvolvimento da solução, por meio de *web services* e computação em nuvem, para que estas sejam capazes de atender aos requisitos de integração entre sistemas de informação e interoperabilidade entre eles, assim como debateu algumas funcionalidades de ambas arquiteturas que visam a definição de qual formato de dados seria o mais adequado (*XML, JSON*).

A opção sobre qual arquitetura optar no momento do desenvolvimento de aplicações tem gerado discussões em cima do tema, pois ambas apresentam características favoráveis e desfavoráveis de acordo com alguns aspectos a serem considerados:

O padrão *SOAP* possui um arcabouço estrutural bem definido e robusto que lhe garante confiabilidade, sendo o mesmo amparado pelo *W3C* como protocolo padrão para o desenvolvimento de *web services*. No entanto, a grande quantidade de serviços que precisa ser gerenciada lhe confere complexidade para desenvolver e sua desempenho depende do servidor onde o serviço está publicado, como também da rede.

O padrão *REST* é tido como simples para desenvolvimento de aplicações e de bom desempenho, e a possibilidade de adoção de qualquer formato de dados lhe confere flexibilidade, porém para aplicações críticas e de alta disponibilidade sua performance pode ser prejudicada e a falta de padrão para desenvolvedores dificulta a criação de um modelo padrão para arquitetura.

Ao final, a abordagem escolhida para arquitetura orientada a serviços aplicada neste trabalho será o padrão *REST* que juntamente com a tecnologia de computação em nuvem irá compor o modelo de solução para falta de integração do AVA e SGA desta pesquisa conforme justificativas a seguir.

4. PROCEDIMENTO METODOLÓGICO

Este capítulo tem por objetivo apresentar os principais procedimentos metodológicos adotados neste trabalho, considerando-se os seus objetivos e o contexto na qual a pesquisa foi realizada. No desenvolvimento do mesmo serão destacadas e justificadas as escolhas empregadas para abordagem em curso.

Esta pesquisa está classificada como exploratória, uma vez que ajuda na compreensão mais precisa do que se pretende pesquisar e na busca de variáveis relevantes que precisam ser consideradas no decorrer do estudo em andamento.

Este tipo de pesquisa tem como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses. A grande maioria dessas pesquisas envolve: (a) levantamento bibliográfico; (b) entrevistas com pessoas que tiveram experiências práticas com o problema pesquisado; e (c) análise de exemplos que estimulem a compreensão. Gil (2007, p. 41).

Como destacado na citação de Gil (2007), este trabalho considerou e adaptou, em sua metodologia, as etapas descritas de: levantamento bibliográfico (tópico 4.1), entrevistas com pessoas que tiveram experiências práticas com o problema da pesquisa (tópico 4.2) e análise de exemplos para compreensão do mesmo.

4.1. Levantamento Bibliográfico

4.1.1. Escolha do padrão arquitetural *REST*

Durante o levantamento bibliográfico realizado foi percebido que o modelo de solução proposto para solucionar a falta de integração entre o AVA e o SGA deve seguir a abordagem orientada a serviços por meio de *web services* no padrão *REST*, haja vista este padrão arquitetural prover condições necessárias para integração de *softwares* heterogêneos e por fazer uso do protocolo *HTTP* e seus cabeçalhos:

- *GET*: Para consultas;
- *POST*: Para inserções;
- *PUT*: Para atualizações;
- *DELETE*: Para remoções.

Conforme mencionado na seção 3.3.2.2, o modelo arquitetural *REST* utiliza o protocolo *HTTP* para fazer suas requisições e obter suas respostas, contudo o ponto de destaque do padrão arquitetural *REST* é a padronização de como essas chamadas são feitas e as técnicas utilizadas para tal. Essas chamadas são totalmente baseadas em *URIs* (*Uniform Resource Identifier*), uma sintaxe

universal para identificar os recursos, onde cada recurso deve conter uma *URI* para que seja possível consultar informações sobre ele.

Em síntese, as justificativas para esta escolha residem em características já descritas anteriormente, tais como:

1. Ser desvinculado de tecnologia, por se tratar de um protocolo de arquitetura, ou seja, se preocupa em como devem ser arquitetadas as requisições nas camadas a serem utilizadas;
2. O formato de dados do *REST* é muito mais simples (*XML* ou *JSON*) se comparado a complexidade imposta pelos requisitos exigidos no descritor *WSDL* do padrão *SOAP* que faz uso somente de *XML*;
3. As respostas no *REST* são legíveis, algo que nem sempre ocorre com o protocolo *SOAP*, cujas respostas são encapsuladas em um formato verboso;

4.1.2. Escolha do paradigma de computação em nuvem

Depois de construir uma aplicação se faz necessário disponibilizar a mesma em algum servidor de *internet*. A abordagem tradicional sugere a utilização de um *datacenter* para esse procedimento, contudo o paradigma de computação em nuvem tornou possível ter aplicações e dados disponibilizados via *web*, sem demandar instalações ou armazenamento de dados locais. De acordo com o Centro de Pesquisa Pew em seu artigo *The future of cloud computing* (<http://www.pewinternet.org/2010/06/11/the-future-of-cloud-computing/>):

Uma sólida maioria de especialistas em tecnologia e *stakeholders* que participam da quarta pesquisa sobre o Futuro da Internet esperam que até 2020 a maioria das pessoas acessará os aplicativos de software de forma online e compartilhará e acessará informações através do uso de redes de servidores remotos, em vez de depender principalmente de ferramentas e informações alojadas em seus computadores pessoais. Eles dizem que a computação em nuvem se tornará mais dominante do que o *desktop* na próxima década. Em outras palavras, a maioria dos usuários executará a maior parte das atividades de computação e comunicação através de conexões com servidores operados por empresas externas.

O paradigma de computação em nuvem deverá continuar a se expandir por meio de empresas especializadas em explorar o serviço conforme destacado na seção 2.4.2.

As finalidades provenientes da utilização do paradigma de computação em nuvem podem ser destacadas em:

1. Para recuperação de desastres e continuidade de negócios; uma vez que gerenciar e manter um *datacenter* requer custos elevados, sendo que o mesmo requer ambientes controlados e seguros;
2. Trabalho colaborativo; equipes desenvolvem soluções ou mesmo trabalham em projetos diversificados com suporte do paradigma de computação em nuvem;

3. Armazenar arquivos na *Internet*; o exemplo clássico de finalidade padrão para computação em nuvem;

Entretanto, é válido destacar que a definição de computação em nuvem é bastante ampla, afinal, vários são os tipos de recursos de TI que podem ser alocados sob demanda para serem acessados.

A vantagem para utilização do paradigma em nuvem reside no simples fato de organizações/instituições poderem expandir seus serviços a clientes e equipes de trabalho sem a necessidade de investimentos em infraestrutura de equipamentos servidores ou *datacenters*.

A empresa garante um ambiente seguro, flexível e expansível para acomodar quantos usuários o negócio necessitar, tendo realizado investimentos no cenário dos negócios baseados na nuvem com o objetivo de fornecer aos clientes soluções que melhor se encaixem a suas necessidades. O maior objetivo da solução da nuvem é proporcionar maior produtividade. Junior (2015, p.20).

4.2 Entrevista semiestruturada

Seguindo a metodologia de Gil (2007) no seu item (b), esta etapa procedeu com entrevistas com pessoas que tiveram experiências práticas com o problema da pesquisa para coletar dados que sejam requisitos funcionais a serem utilizados pelo modelo.

Foram realizadas entrevistas semiestruturadas em decorrência do fato desta permitir maior interação entre os participantes. As análises destas entrevistas estão focadas no questionário denominado “Questionário semiestruturado para levantamento de requisitos”, e consta no Apêndice B.

Considerando o contexto da pesquisa foram escolhidas três IES que possuem a mesma problemática de falta de integração entre SGA e AVA. Considerando esse critério foram selecionadas as IES:

1. Instituto Federal de Educação do Piauí – IFPI – Que tem o sistema Q-Acadêmico como SGA e utiliza o *Moodle* como AVA;
2. Universidade Estadual do Piauí – UESPI – Que tem o sistema CA (controle acadêmico) como SGA e utiliza o *Moodle* como AVA;
3. Centro Universitário Uninovafapi – Que tem o ZEUS como SGA e utiliza o *Moodle* como AVA;

O perfil dos participantes estabeleceu como critério que os mesmos deveriam ter atuação direta com os sistemas a serem integrados, ou seja, são usuários que possuem conhecimento operacional ou experiência técnica com o manuseio de pelo menos um dos sistemas. O perfil dos participantes está evidenciado no quadro 8, abaixo.

Quadro 8- Perfil dos participantes

| Participante | Modalidade | Experiência SGA | Experiência AVA | Grau Escolaridade |
|---------------------|-------------------|------------------------|------------------------|--------------------------|
| Participante 1 | Presencial e EAD | Sim | Sim | Mestre Computação |
| Participante 2 | Presencial | Sim | Não | Bacharel em Sistemas |
| Participante 3 | Presencial e EAD | Sim | Sim | Bacharel Computação |
| Participante 4 | Presencial e EAD | Sim | Sim | Bacharel Computação |
| Participante 5 | Presencial | Sim | Não | Bacharel Computação |
| Participante 6 | Presencial | Sim | Não | Pedagoga |
| Participante 7 | Presencial e EAD | Sim | Sim | Bacharel Computação |
| Participante 8 | Presencial | Sim | Não | Pedagoga |

Fonte: Elaborado pelo autor

Conforme destacado no quadro 9 os participantes atuam tanto na modalidade presencial como EAD ou apenas em uma delas, contudo é destacado a experiência que estes possuem com a utilização dos sistemas envolvidos nesta pesquisa. As contribuições oriundas destes participantes agregaram significativamente para o desenho do modelo proposto.

Em comum ambas instituições utilizam o mesmo AVA para suas atividades em EAD, no caso em questão o *Moodle*, contudo seus SGA's se enquadram na categoria de sistemas proprietários (IFPI) ou sistemas legados (UESPI e Uninovafapi).

Foram identificados vários aspectos quanto a utilização dissociada de ambos sistemas da mesma forma que deficiências pontuais na utilização dos mesmos e sugestões de como o processo de integração pode contribuir para melhora na operacionalização destes.

4.3 Resultados preliminares

Vários são os recursos utilizados pelos participantes no que tange o SGA: registro de presenças, registro de notas/conceitos/avaliações, matrículas de discentes, consultas/cadastro de dados de discentes/docentes, disponibilização de material on line, oferta de período/semestre acadêmico, procedimentos similares a todos com pequenas variações de acordo a política de cada IES. Contudo, no que se refere às dificuldades encontradas em gerir o processo de informações acadêmicas no SGA obteve-se algumas respostas conforme evidenciado a seguir:

“Por ser uma solução proprietária tudo o que tange a novas funcionalidades, correções de bugs entre outros depende da empresa proprietária do software já que o código fonte não é aberto.” (Participante 2).

“Como trabalhamos com um sistema manual, a maior dificuldade é a perda de tempo na inserção das informações.” (Participante 9).

“Como sistema é muito antigo e como não realizamos manutenções no seu código, quando determinada funcionalidade falha, as vezes existe a necessidade de atualizações no próprio banco de dados.” (Participante 5).

É perceptível que estes participantes utilizam recursos comuns para seus respectivos SGA's que constituem os principais processos e suas ações administrativas básicas de operacionalização, entretanto as dificuldades descritas por diferentes participantes apenas ratificam a necessidade de um modelo de integração que permita o gerenciamento destas informações para outros sistemas.

A modalidade EAD, conforme destacado pelos participantes da pesquisa, faz uso de um AVA que por sua vez utiliza recursos de comunicação como e-mail, fórum, notícias, atividades, etc., possibilitando a aprendizagem colaborativa, a discussão através de chat, a visualização de vídeos, o acesso ao ensino por meio de dispositivos móveis, entre outros recursos existentes, com o objetivo de obter êxito no processo de ensino aprendizagem do discente. Outrossim, as citações a seguir evidenciam a fala de alguns participantes quanto as dificuldades encontradas em sua prática de uso com o AVA:

“Sincronizações das informações entre o setor acadêmico e o AVA, quando um aluno tranca sua matrícula esta informação não é trocada automaticamente entre os dois sistemas.” (Participante 1).

“Dificuldade na integração entre o ambiente virtual e controle acadêmico principalmente o gerenciamento das notas dos alunos entre os dois sistemas.” (Participante 3).

“Inserir os professores e tutores manualmente no sistema, por não haver um processo que não automatize.” (Participante 4).

“Criar e modificar turmas e disciplinas, inserir alunos e professores nas suas respectivas

turmas e disciplinas.” (Participante 4).

Primordialmente o AVA é desenhado para servir como ferramenta de apoio a modalidade EAD. Trata-se tipicamente do uso da tecnologia em favor do processo de ensino e aprendizagem, contudo seu manuseio é dificultado pela falta de troca de informações com o SGA que detêm as informações acadêmicas necessárias para povoar com os dados de cursos, disciplinas, discentes e docentes o banco de dados do AVA.

Da mesma forma, os participantes destacaram de que maneira deve ser realizada a integração entre os supracitados sistemas e os benefícios oriundos deste processo:

“Centralização das atualizações de dados evitando o retrabalho. O cadastro seria realizado apenas no setor acadêmico e o AVA receberia as alterações automaticamente.” (Participante 1).

“Facilitar a matrícula, gestão das notas e atividades e integração do diário de classe. Toda a informação acadêmica que diz respeito aos alunos e professores estarem disponíveis em ambos os sistemas.” (Participante 3).

“A unificação das informações, faz com que gere um trabalho melhor e com êxito. A integralização de todos os setores e também a unificação das informações.” (Participante 8).

“Contribuíram para dinamizar o processo. Inserção de professores de forma automática no AVA e de notas do controle acadêmico para o AVA.” (Participante 4).

Analisando os dados obtidos nesta pesquisa percebe-se a necessidade pelo processo de integração entre AVA e SGA de forma que o SGA constitui o sistema provedor do maior volume de dados a serem trocados e sincronizados entre ambos sistemas, uma vez que as informações básicas sobre discentes, matrículas, disciplinas, docentes, etc. são armazenadas em sua base de dados. Cabe ao AVA disponibilizar os resultados (notas/conceitos/avalições) obtidos pelos discentes para procedimentos de controle acadêmicos mantidos pelo SGA. O modelo de integração proposto, entretanto, deve prover também um mecanismo que sincronize as informações entre os sistemas à medida que forem feitas atualizações entre ambos no decorrer do semestre/período letivo.

5. ARQUITETURA E MODELAGEM

O estudo para um modelo de integração entre SGA e AVA passa pela necessidade de determinar seus recursos e funções principais, ou seja, determinar padrões comuns dos sistemas envolvidos perante um grande quantitativo de sistemas dessa natureza disponíveis no mercado.

A integração dos sistemas envolvidos constitui uma prática que deve levar em consideração alguns itens:

- Visualizar todos os pontos que serão integrados;
- Definir o processo que envolva a troca de mensagens entre os serviços expostos;
- Determinar o modelo de dados da integração uma vez que cada sistema envolvido possui sua própria estrutura;

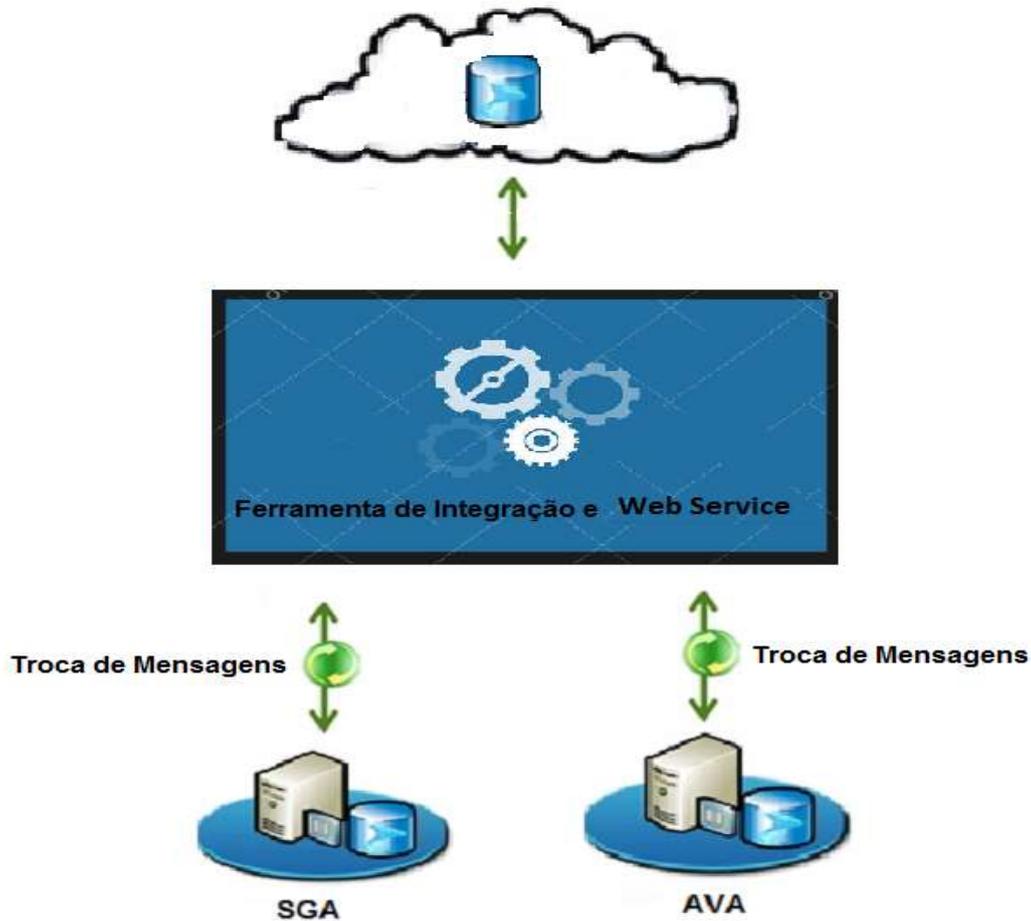
O princípio a ser seguido visa criar e disponibilizar um modelo de alto nível e fracamente acoplado baseado no conceito de serviços, sendo serviço um componente de *software* com uma *interface*²⁸ bem definida e que tem com principais características ser modular e reutilizável.

Os serviços que apoiarão os processos de negócio e atenderão aos seus requisitos devem ser definidos de forma adequada, para que apresentem atributos como reusabilidade e flexibilidade e estejam alinhados com as metas de negócio. Além disso, os serviços devem ser especificados de modo a fazer parte de um repositório de serviços reusáveis, que possam ser orquestrados (compostos) em sistemas para apoiar processos de negócios. Fugita e Hiram (2012, p2).

A figura 18 a seguir destaca uma visão geral da arquitetura para o modelo proposto:

²⁸ Interface: Contrato entre o consumidor e fornecedor. Em um serviço a implementação é escondida do consumidor, sendo disponibilizada sua interface.

Figura 18. Visão geral da Arquitetura.



Fonte: Elaborado pelo Autor.

Importante destacar que a Ferramenta de Integração controla os serviços ofertados pelo *web service*, agindo como interface com os usuários e responsável pela configuração inicial para utilização do modelo.

O papel do *web service* é o de disponibilizar os serviços providos pela Ferramenta de Integração e está se encarrega de realizar consultas e inserções diretamente nos bancos de dados dos sistemas envolvidos.

As informações geradas pelo SGA são transportadas para um banco de dados hospedado na nuvem que será chamado de banco de integração. A Ferramenta de Integração será responsável por ler estas informações no banco de integração e determinar a ação necessária executando esta ação no banco do AVA.

Entretanto há restrições quanto ao acesso de códigos-fonte tanto de SGA's quanto de alguns AVA's que são propriedades de empresas que licenciam seus produtos mais detêm os direitos autorais

sobre os mesmos, fato este que impossibilita a análise e modificação de suas estruturas. Não apenas isso, mas ocorre que alguns desses sistemas se enquadram na categoria de sistemas legados cujo custo para manutenção, atrelado ao uso de tecnologias normalmente ultrapassadas, inviabilizam sua modernização, mas que devido ao grau de criticidade que, por vezes, desempenham para a organização/instituição continuam ativas.

Um dos preceitos da EAI consiste no desenvolvimento de uma visão totalmente nova sobre os negócios da empresa e suas aplicações procurando encaixar as aplicações existentes na nova visão, buscando caminhos para reutilizar de forma eficiente o que já existe ao mesmo tempo em que se incorporam novas aplicações de dados.

Considerando o cenário no qual não há acesso ao código-fonte dos sistemas a serem integrados ou mesmo por estes se tratarem de sistemas legados e seguindo princípio de reusar de forma eficiente o que já existe, faz-se necessário a utilização de uma abordagem que permita contemplar todas essas variáveis. Esta abordagem deve também ser capaz de incluir o planejamento de testes, o projeto de casos de teste e a coleta e avaliação de resultados.

A abordagem utilizada neste trabalho é denominada caixa preta (*black box*) que segundo Pressman (2011) especifica o comportamento de um sistema ou uma parte de um sistema. O sistema (ou parte) responde a estímulos específicos (eventos) aplicando um conjunto de regras de transição que mapeiam os estímulos de uma resposta. Na abordagem caixa preta não é necessário conhecer ou acessar o código para que possam ser realizados testes de aceitação. Diferentemente, o sistema é testado através da interface disponível, a partir da qual o usuário/cliente do sistema pode alimentar o sistema com informações de entrada, as quais serão processadas gerando as saídas.

Os testes caixa-preta, ou funcionais, como o próprio nome diz, são empregados para demonstrar que as funções do software estão operacionais, que a entrada é adequadamente aceita, a saída é corretamente produzida e que a integridade da informação externa é mantida. Testes caixa preta examinam algum aspecto fundamental do sistema, pouco se preocupando com a estrutura lógica interna do software. Gava (2009, p.50)

Em relação ao modelo de solução a abordagem caixa preta será utilizada para demonstrar como as iterações entre o usuário e o sistema são simuladas e interpretadas pelo modelo.

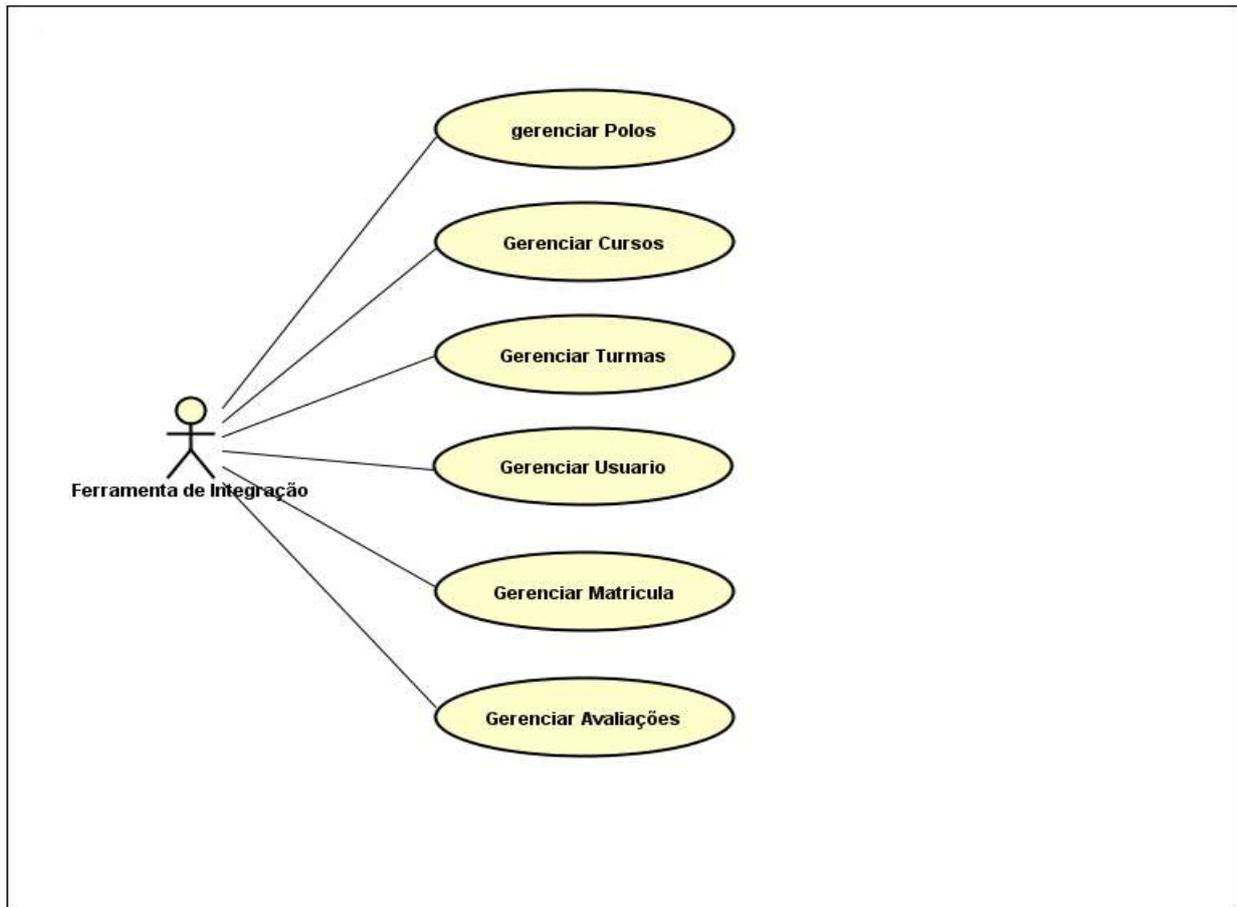
O processo de modelagem para solução proposta foi realizada por meio da utilização de UML (*Unified Modeling Language*), uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos. É uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação (Guedes, 2011).

Com vistas aos requisitos obtidos por meio de entrevistas destacadas na seção 4.2 foi possível modelar o diagrama de Casos de Uso para solução proposta. Esse diagrama possibilita a

compreensão do comportamento externo do sistema e as funcionalidades oferecidas por ele.

O diagrama de Casos de Uso representado na figura abaixo destaca uma visão geral das funcionalidades que o sistema tem a oferecer aos usuários, contudo o mesmo não se preocupa com a questão de como tais funcionalidades serão implementadas.

Figura 19 – Diagrama de Casos de Uso da solução



Fonte: Elaborado pelo Autor.

Conforme demonstrado na figura 19, destaca-se o papel do ator Ferramenta de Integração que se relaciona com *web service* para trabalhar os requisitos do sistema. Os casos de uso propriamente ditos representam os diversos serviços disponibilizados aos usuários da Ferramenta de Integração.

Os serviços disponibilizados, face os casos de usos, para o modelo proposto são descritos no quadro a seguir considerando o contexto deste trabalho de possibilitar a integração entre SGA e AVA de acordo com os requisitos obtidos.

Quadro 9 – Serviços ofertados

| Serviço | Ação correspondente |
|---------------------|---|
| Gerenciar polo | - Dados oriundos do SGA para registrar uma ação na base de integração dos polos onde serão ofertados cursos na modalidade EAD. |
| Gerenciar curso | - Dados oriundos do SGA para registrar uma ação na base de integração dos cursos que serão ofertados para modalidade EAD. |
| Gerenciar turma | - Dados oriundos do SGA para registrar uma ação na base de integração das turmas ofertadas para os cursos na modalidade EAD. As disciplinas a serem ministradas serão inseridas dentro de uma Turma. |
| Gerenciar usuário | - Definição do perfil dos usuários: Discente, Professor, Tutor, Administrador. - Cadastro de usuários oriundos do SGA ou manualmente. |
| Gerenciar matrícula | - Dados oriundos do SGA para inserção no bando de integração dos discentes curricularmente matriculados no período/semestre letivo. Estabelecer mecanismo para sincronismo das ações de matrículas dos discentes na base de dados do AVA. |
| Gerenciar avaliação | - Dados oriundos do AVA para registrar uma ação na base de integração - O resultado oriundo de Avaliações deverá ser registrado no SGA |

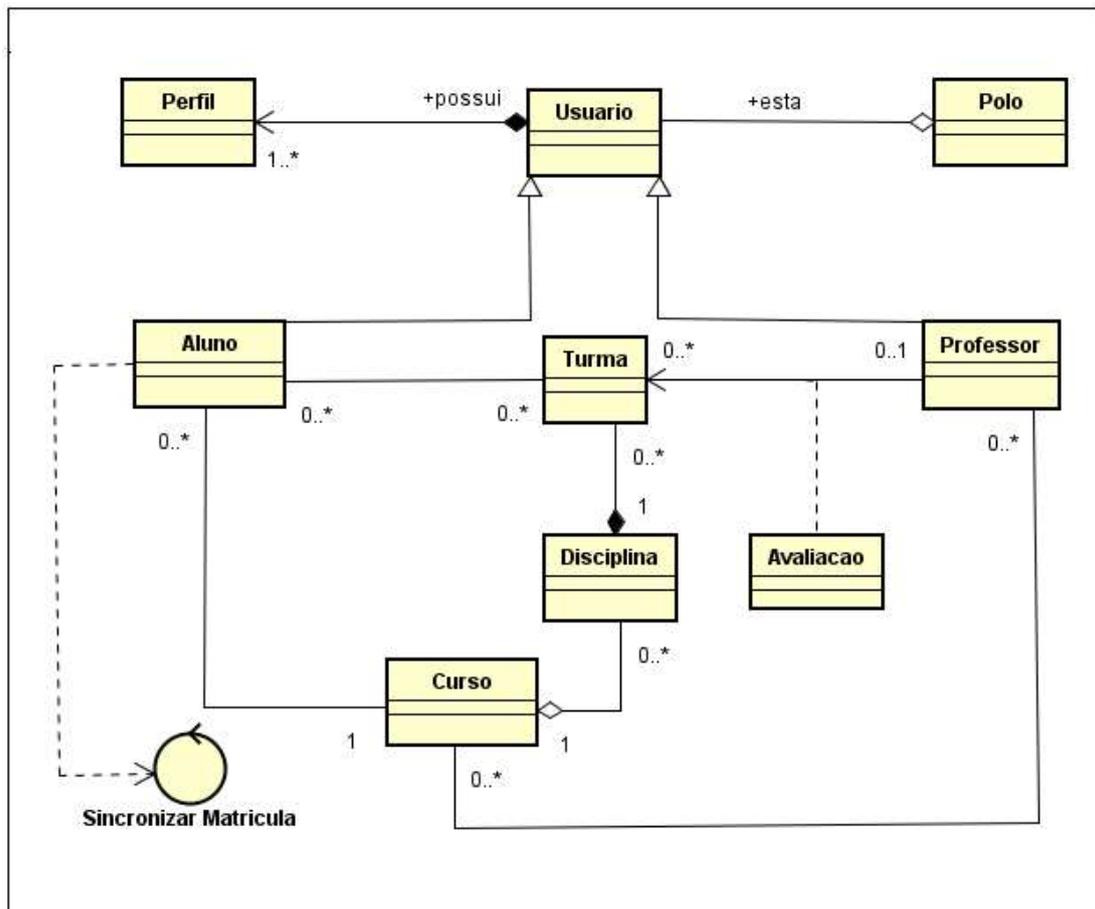
Fonte: Elaborado pelo Autor.

Analisados os requisitos obtidos, bem como os trabalhos existentes nesta área, propomos um modelo para um sistema de integração de dados entre SGA e AVA para troca de dados que destacam os seguintes serviços:

- Exportação de polos;
- Exportação de cursos;
- Exportação de turmas;
- Exportação de discentes;
- Exportação de avaliações;

A figura 20 a seguir apresenta o diagrama de Classes, sem detalhes dos atributos, que dá suporte as ações estabelecidas pelos serviços destacados:

Figura 20 – Diagrama de Classes



Fonte: Elaborado pelo Autor.

O diagrama de Classes da figura 20 destaca a classe “Usuario”, pois este está relacionado com um ou vários perfis de acordo com o papel que desempenhará, bem como se relaciona com a classe “Polo”. As classes “Aluno” e “Professor” herdam as características da classe “Usuario”. A classe “Aluno” se relaciona diretamente com as classes “Turma” e “Curso”, assim como a classe “Professor” se relaciona com as classes “Turma” e “Avaliação”. A classe “Aluno” está alinhada com “Sincronizar Matricula” para manter atualizado o status do discente.

Identificados os serviços (exportação de polos, cursos, turmas, discentes e avaliações) pela compreensão dos sistemas envolvidos e seus requisitos levantados, é possível identificar serviços candidatos com base em entidades informativas, conjunto de dados que representam recursos, por meio de relacionamentos existentes gerando *URIs* para representar web service *REST*.

Liu et al (2008) destaca um processo comum para reengenharia de sistemas utilizado o padrão arquitetural *REST*. O procedimento proposto consiste em:

1. Identificar e extrair os recursos candidatos por intermédio de engenharia reversa, análise das camadas do modelo ou identificado por peritos;
2. Criar serviços com análise de relações estáticas em sistemas legados nas quais diferentes tipos de relacionamentos podem ser mapeados para diferentes partes dos recursos;
3. Projetar *URIs* para representar serviços onde web services *REST* serão determinados pelos recursos expostos, escopo, restrição nos recursos e operações padrões;

Alguns serviços podem ser determinados por mais de uma entidade informativa tais como: Diagramas ER, Diagramas *UML*, requisitos, documentos ou especialistas. Os recursos podem ser classificados pela maneira que as entidades informativas se relacionam entre si, sendo possível identificar tipos de recursos e *URIs* sob regras predefinidas conforme demonstrado no quadro abaixo:

Quadro 10: Tipos de Relação de Recursos e Regras de Mapeamento de URIs

| Relation Type | Sketch Map | Rule | URI |
|---------------------------|--|------|---|
| Generalization | <pre> classDiagram Faculty < -- Lecturer Faculty < -- Professor </pre> | “/” | http://myschool.example.com/faculty , http://myschool.example.com/faculty/professor |
| Aggregation ²⁹ | <pre> classDiagram Class o-- "*" Students : students </pre> | “:” | (1) http://myschool.example.com/class http://myschool.example.com/class/student (2) http://myschol.example.com/student |
| Composition | <pre> classDiagram Academy *-- "1..*" Department : thedepartment </pre> | “:” | http://myschool.example.com/academy , http://myschool.example.com/academy:department |
| Association | <pre> classDiagram Student "*" -- "*" Course </pre> | “-“ | http://myschool.example.com/student-course |

Fonte: Adaptado de Liu et al (2008)

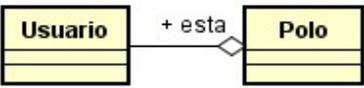
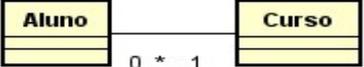
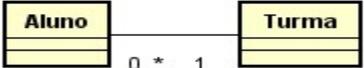
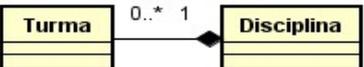
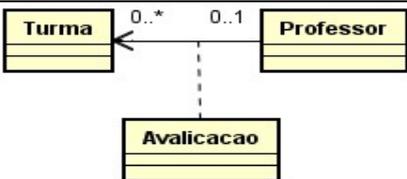
O relacionamento de generalização (Generalization) representa uma hierarquia e cada entidade nesse relacionamento tem um pedaço de informação do escopo; a agregação (Aggregation) representa o relacionamento típico todo/parte, a composição (Composition) é exatamente igual à

²⁹ Se a entidade parte puder existir independentemente, sem a entidade todo em algum contexto, ela deve ter sua própria URI.

agregação, exceto que a "parte" é controlada pelo "todo" e não podem existir independentemente e associação (Association) representa uma entidade associada a outra. Importante destacar a relação de dependência para qual não há relação direta entre duas entidades, mas as mesmas podem ser conectadas combinando as duas entidades no mesmo nível com vírgulas em uma mesma hierarquia.

Considerando o Diagrama de Classes proposto para o modelo de integração podemos determinar os seguintes serviços na forma de *URIs* ao assumirmos a existência de uma raiz *URI*: <http://localhost:8888/webservice/rest>.

Quadro 11: Mapeamento e URIs produzidos

| Ordem | Mapeamento | URI |
|-------|---|---|
| 1 |  | http://localhost:8888/webservice/rest/polo |
| 2 |  | http://localhost:8888/webservice/rest/aluno-curso |
| 3 |  | http://localhost:8888/webservice/rest/aluno-turma |
| 4 |  | http://localhost:8888/webservice/rest/turma:disciplina |
| 5 |  | http://localhost:8888/webservice/rest/professor,avaliacao |

Fonte: elaborado pelo autor

O serviço destacado pela *URI* (1) carrega dados sobre a criação de polos acadêmicos para os cursos da modalidade EAD onde estarão matriculados os discentes. O serviço destacado pela *URI* (2) fornece a descrição dos cursos ofertados para cada polo. O da *URI* (3) fornece a descrição da(s) turma(s) dos cursos ofertados em cada polo. O serviço da *URI* (4) fornece a descrição das disciplinas ofertadas para as turmas nos cursos existentes para o polo e por fim o serviço destacado para *URI* (5)

fornece os resultados/conceitos obtidos pelos discentes. O vínculo de matrícula aos cursos, turmas e disciplinas de cada discente ocorre pelo sincronismo de matrícula do SGA para o AVA.

Conforme destacado na figura 18 este modelo foi projetado para *web services REST* e a aplicação *web* denominada Ferramenta de Integração ambas na mesma arquitetura, pois tanto os recursos da aplicação e serviços providos pelo *web services* podem ser concebidos numa mesma visão que possam simplificar processos de análise e projeto. Desta forma ações referentes às definições de perfis dos usuários (Administrador, tutor, coordenador, etc), configuração de conexão com os sistemas SGA e AVA ficam a cargo da Ferramenta de Integração.

5.1. Validação

O modelo concebido para este estudo foi validado por quatro especialistas em desenvolvimento de sistemas que ratificaram os requisitos e a aplicabilidade do mesmo.

Este mesmo procedimento de validação foi utilizado por Lewis et al (2016) para validar os modelos presentes na pesquisa intitulada “*A Decision Model for Cyber-Foraging Systems*”.

Os especialistas em desenvolvimento que colaboraram com este trabalho possuem experiência em várias abordagens para integração de sistemas (*SOAP, REST*, orientado a dados, etc) e atuam profissionalmente há mais de 5 anos (3 especialistas) ou a pelo menos 2 anos (1 especialista), contudo a experiência em abordagens para integração ou atuação profissional não foram os únicos fatores para escolha dos especialistas em desenvolvimento.

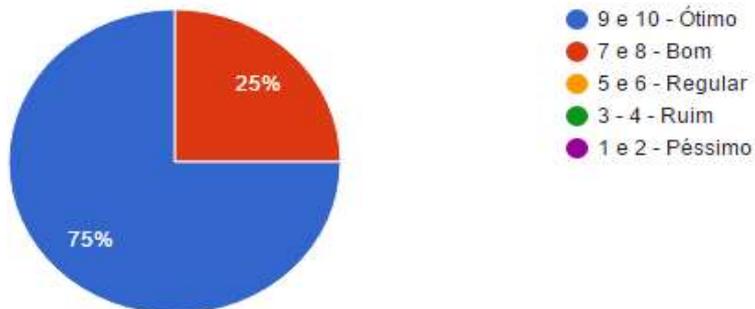
Todos já estiveram contato com os sistemas a serem integrados neste modelo, pois trabalham ou já trabalharam em IES e são conscientes da realidade e necessidade da integração dos mesmos.

Sobre estes mesmos especialistas, é oportuno destacar também que dois deles possuem mestrado em Engenharia de *Software* e um (1) atua profissionalmente como perito forense na área de informática.

Os especialistas em desenvolvimento de sistemas foram apresentados ao modelo proposto e solicitados a responder as seguintes perguntas para obtenção de um parecer quanto sua viabilidade:

01. Em uma escala de 1 a 10 como você avalia o modelo proposto. Utilize a escala de conceitos a seguir:

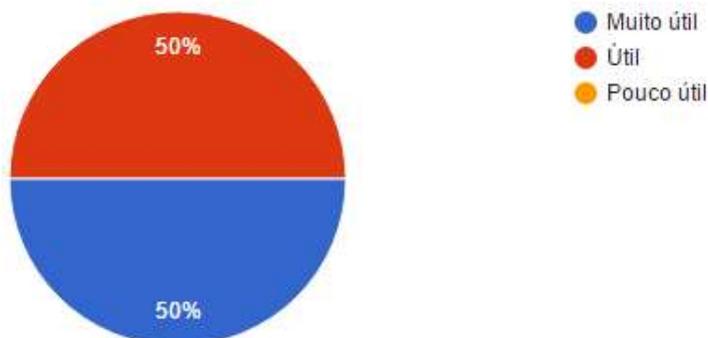
Gráfico 1 – Avaliação do modelo



Fonte: Elaborado pelo autor

02. Quanto ao seu nível de aceitação para o modelo proposto. Utilize a escala de conceitos abaixo:

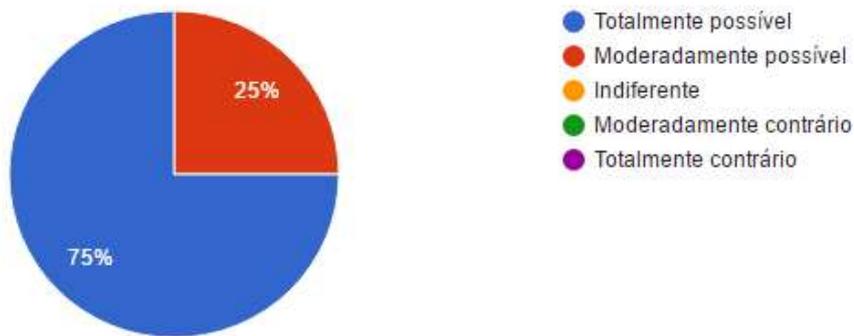
Gráfico 2 - Nível de aceitação



Fonte: Elaborado pelo autor

03. Quanto a possibilidade de adoção do modelo proposto. Utilize a escala de conceitos abaixo:

Gráfico 3 – Adoção do modelo



Fonte: Elaborado pelo autor

Quando indagados se o modelo proposto ajudava na avaliação e, particularmente, na identificação e mitigação de risco, todos foram unânimes em afirmar que sim.

Três dos quatro especialistas em desenvolvimento consultados apresentaram suas considerações para melhoria do modelo proposto:

O primeiro especialista em desenvolvimento, mestre em engenharia de software, recomendou adicionar esquemas de autenticação unificada para o modelo;

O segundo, também mestre em engenharia de software, sugere deixar as operações de atualizações serem realizadas exclusivamente pelo SGA, uma vez que o mesmo já possui todas as regras de integridade. A ferramenta de integração realizaria chamadas aos métodos do SGA passando os argumentos oriundos das repostas do *web service*;

E o terceiro, perito forense da área de informática, destaca que como o protocolo *REST* não se prende à utilização de padrões pré-definidos ter um documento/modelo de arquitetura que especifique detalhadamente a utilização do formato da mensagem é indispensável para garantir a interoperabilidade em futuro próximo da ferramenta com futuros softwares/serviços a serem adquiridos ou implementados pela instituição.

O formulário aplicado para validação do modelo foi denominado “Formulário de validação para modelo de integração entre SGA e AVA” e consta no Apêndice C deste trabalho.

5.2. Restrições previstas

Os problemas previstos para implementação envolvem determinar a metodologia para definição da infraestrutura de suporte necessária à integração que garanta a integridade e a segurança das informações. A questão de segurança é vital quando se trata de troca de informações pela *Internet*.

O ponto primeiro a ser levantado determina a utilização do protocolo *HTTPS*, pois este é suportado por *web services* e provém níveis aceitáveis de confiabilidade e segurança. Outra contramedida para proteger *web service* é exigir autenticação do usuário para acessá-lo por meio de *login* e de uma senha o que pode ser feito de várias maneiras.

A maior dificuldade entretanto é a observância e necessidade de atendimento ao Decreto da Presidência da República nº 8.135, de 4 de novembro 2013 que dispõe sobre as comunicações de dados da administração pública federal direta, autárquica e fundacional, e sobre a dispensa de licitação nas contratações que possam comprometer a segurança nacional, que em seu artigo primeiro determina que as comunicações de dados da administração pública federal direta, autárquica e fundacional deverão ser realizadas por redes de telecomunicações e serviços de tecnologia da informação fornecidos por órgãos ou entidades da administração pública federal, incluindo empresas públicas e sociedades de economia mista da União e suas subsidiárias.

Contudo, um ato conjunto dos Ministros de Estado da Defesa, do Planejamento, Orçamento e Gestão e das Comunicações disciplinou o disposto neste artigo e estabeleceu procedimentos, abrangência e prazos de implementação de até 60 meses para migração dos serviços de redes de telecomunicações, pois os programas e equipamentos deverão ter características que permitam auditoria para fins de garantia da disponibilidade, integridade, confidencialidade e autenticidade, assim como o armazenamento e a recuperação de dados deverá ser realizada em centro de processamento de dados fornecido por órgãos e entidades da administração pública federal das informações, na forma da regulamentação de que trata, ou seja, este trabalho pode ser executado e concluído em um curto ou médio prazo e posteriormente migrado para um centro de processamento regulamentado pelo decreto no prazo determinado em atendimento as novas normas vigentes.

6. CONCLUSÕES FINAIS E TRABALHOS FUTUROS

Este trabalho teve como objetivos discutir e apresentar uma solução para um modelo de interoperabilidade entre sistemas heterogêneos de alto nível. O modelo proposto e suas funcionalidades irão beneficiar os usuários na operacionalização do SGA e AVA, podendo evoluir posteriormente para novas funcionalidades. A definição por uma abordagem orientada a serviços que utiliza *web services* e computação em nuvem permitirá um fraco acoplamento entre os sistemas envolvidos, o que lhes garantirá independência, ao passo que estes estarão sincronizados, pois todas as alterações efetuadas no SGA serão propagadas no AVA e vice-versa.

O problema de pesquisa é proveniente do mundo real e atinge diversas IES, principalmente os atores que fazem uso do SGA e AVA (discentes, docentes, funcionários administrativos e equipes multidisciplinares) que tem que lidar com operacionalização dissociada de ambos sistemas e suas consequências como fluxos desnecessários de trabalho, retrabalhos e riscos de erros operacionais que comprometem o bom desempenho das atividades administrativa de qualquer IES. Os benefícios oriundos de um processo de integração entre estes sistemas visam aumentar a produtividade em relação a operacionalização dos mesmos e na simplificação de seus fluxos de processos permitindo aos atores envolvidos tempo para concretização de outras atividades mais relevantes.

Fez-se necessário, portanto, discutir e caracterizar as melhores abordagens existentes para um processo de integração entre os supracitados sistemas a fim de estabelecer um arcabouço de informações que melhor provesse uma metodologia que atendesse as demandas existentes sem gerar grandes impactos dentro das IES. Sob essa perspectiva uma revisão sistemática da literatura acerca do objeto da pesquisa foi realizada conjuntamente com a percepção de diversos trabalhos relacionados (artigos científicos, dissertações, teses, periódicos especializados, etc.) visando estabelecer um “estado da arte” do que seria um processo de integração.

Os estudos feitos nessa área nos levaram a diversas abordagens, contudo devido a restrições inerentes aos sistemas a serem integrados (sistemas proprietários ou mesmo sistemas legados) somente duas abordagens se destacaram: orientada a dados e orientada a serviços. Ambas com suas respectivas vantagens e desvantagens. A orientação a dados se mostrou funcional e prática, contudo suas desvantagens relativas a forte dependência ente os sistemas envolvidos e por não ser flexível não a tornam candidata para um modelo de solução que possa ser adaptado e implementado por qualquer IES.

Assim, a orientação a serviços, por meio de *web services*, despontou como abordagem a ser

utilizada por este trabalho pelas suas características principais: facilidade de integração de diferentes tecnologias e ser desacoplada de plataforma, fatos que melhoram a forma como sistemas distribuídos e heterogêneos são desenvolvidos e compartilhados. Entretanto fez-se necessário analisar as principais formas de implementação de um processo de integração com base em orientação a serviços por intermédio de *web services*: O protocolo *SOAP* e o modelo arquitetural *REST*.

O protocolo *SOAP* conta com apoio do *W3C* e tem a seu favor ser um protocolo robusto e confiável com funcionalidades bem definidas pelo seu *Web Service Standards*, entretanto sua complexidade apoiada por um inflexível e rígido modelo de arquitetura (seu próprio *Web Service Standards*) podem comprometer sua performance, fator que não pode ser desconsiderado.

O modelo arquitetural *REST*, ao contrário do *SOAP*, demonstra características mais atraentes para apoiar o modelo de solução proposto, pois sua estrutura determina apenas o básico dos serviços a serem executados, podendo trabalhar com diversos tipos de dados (*XML*, *JSON*, *TXT*, etc) se comparado ao *SOAP* que utiliza apenas *XML*. De fato, este modelo arquitetural vem cada vez mais ganhando espaço entre desenvolvedores de sistemas por apresentar-se como um conjunto de boas práticas flexíveis e simplificadas.

A definição da escolha do paradigma de nuvem para compor o modelo de solução proposto foi naturalmente percebida pela dimensão que este paradigma vem ganhando ao longo dos últimos anos. Embora para a grande maioria dos usuários que a utilizam para armazenar arquivos, o paradigma de nuvem oferece muito mais a desenvolvedores que se beneficiam dos recursos computacionais com um esforço de gerenciamento mínimo se comparado com o uso de *datacenters*. O interessante, entretanto, foi perceber que os estudos realizados em abordagens e trabalhos relacionados a integração pesquisados neste trabalho não foram encontradas nenhuma citação para utilização deste paradigma para os sistemas de SGA e AVA. Fato que nos chamou atenção para possibilidade de explorar esta área por meio do modelo de solução proposto.

Definidas as abordagens do modelo proposto cabia então definir uma metodologia para sua arquitetura que foram formalizadas por meio de diagramas da UML, entretanto os requisitos funcionais foram obtidos por meio de entrevistas semiestruturadas com diversos participantes de três IES que compartilham da mesma problemática de pesquisa. Reforçando a citação anterior inerentes ao fato dos sistemas a serem integrados poderem ser sistemas proprietários ou mesmo sistemas legados o modelo proposto estabeleceu toda a parte conceitual e de planejamento das *URLs* e do arcabouço como um todo. Foi utilizada uma abordagem caixa preta (Black box) para especificar o comportamento do sistema e destacar que as funções da solução proposta são operacionais.

Foram apresentados os diagramas de Estudo de Caso e diagrama de Classes da solução, por meio dos quais foi possível mapear e representar os serviços a serem disponibilizados pela *web service REST* por intermédio da Ferramenta Integradora descrita na solução. O processo de validação do modelo ocorreu pela avaliação do mesmo por especialistas na área de desenvolvimento e integração de sistemas que em sua maioria se manifestaram favoráveis ao modelo proposto considerando-o ótimo e totalmente possível de adoção. As sugestões e considerações feitas pelos mesmos devem ser incorporadas pelo modelo proposto.

As restrições apresentadas são de natureza burocrática e administrativas e afligem entes da esfera Federal da Administração Pública, aquelas de natureza técnicas não causam impactos significativos, pois os requisitos de segurança podem ser implementados no modelo proposto.

As contribuições advindas deste trabalho visaram a elaboração de um modelo de alto nível com base em requisitos funcionais oriundos de IES que compartilham a mesma problemática. Este modelo pode ser implementado por qualquer linguagem ou framework a fim de prover facilidades e praticidade aos envolvidos no processo de ensino e aprendizagem que façam uso de um SGA e AVA.

Como trabalhos futuros destacamos a possibilidade de implementação do modelo proposto e conseqüentemente sua avaliação quanto aos benefícios que o mesmo pode prover no contexto da integração entre SGA e AVA.

7. PUBLICAÇÕES

O tema de integração de sistemas e o estudo das abordagens e soluções possíveis para o problema envolvendo o desenvolvimento de um modelo que solucionasse essa questão, resultou em algumas publicações e apresentações em conferências, com destaque para:

7.1. Eulálio, Athos Denis; Souza, Rodrigo de. **Web Services: Integração de sistemas orientado a serviços com uma proposta de aplicação na EAD.** In: XII Congresso Brasileiro de Ensino Superior a Distância (ESUD) 2015 – Salvador/BA.

Este trabalho foi convidado para compor a segunda edição de 2016 da Revista de Informática Aplicada (RIA) mantida em parceria entre a Universidade Municipal de São Caetano do Sul (USCS) e a Universidade Federal do ABC (UFABC), sendo a mesma classificada no sistema *Web Qualis* nas áreas de Ciência da Computação, Engenharias e Ensino (ISSN: 1809-5585; online: 2179-2518);

7.2. Eulálio, Athos Denis; Souza, Rodrigo de. **Integração de sistemas de gestão acadêmica e ambiente virtual de aprendizagem: uma abordagem orientada a banco de dados.** In: XIII Congresso Brasileiro de Ensino Superior a Distância (ESUD) 2016 – São João del-Rei/MG.

Este trabalho foi considerado um dos 20 destaques do ESUD 2016 e foi convidado também a compor o vol. 3 n.2 da Revista de Educação a Distância EmRede, um periódico da Associação Universidade em Rede – UniRede (ISSN online: 2359-6082);

7.3. Eulálio, Athos Denis; Souza, Rodrigo de; Diniz, Juliana Regueira: **Integration of learning management systems with academic management systems using cloud computing.** In: Euro American Conference on Telematics and Information Systems (EATIS) 2016 – Cartagena / Colômbia.

REFÊNCIAS

AVA ATUTOR: **Ambiente virtual de aprendizagem Atutor**. Disponível em:<
<http://www.atutor.ca/>>. Acesso em 29 de dezembro de 2016;

AVA BLACKBOARD: **Ambiente virtual de aprendizagem BlackBoard**. Disponível em:<
<http://blackboard.grupoa.com.br/>>. Acesso em 29 de dezembro de 2016;

AVA DOCEBO: **Ambiente virtual de aprendizagem Docebo**. Disponível em:<
<https://www.docebo.com/>>. Acesso em 29 de dezembro de 2016;

AVA E-PROINFO: **Ambiente virtual de aprendizagem E-Proinfo**. Disponível em: <
<http://eproinfo.mec.gov.br/>>. Acesso em 29 de dezembro de 2016;

AVA EUREKA: **Ambiente virtual de aprendizagem Eureka**. Disponível em: <
https://eureka.pucpr.br/apresentacao/conteudo/acessar/eureka_oquee.html>. Acesso em: 29 de dezembro de 2016;

AVA GEENIO: **Ambiente virtual de aprendizagem Geenio**. Disponível em: < <https://geen.io/>>.
Acesso em 29 de dezembro de 2016;

AVA LATITUDE LEARNING: **Ambiente virtual de aprendizagem Latitude Learning**.
Disponível em: < <http://www.latitudelearning.com/our-lms/lms-features/free-courses>>. Acesso em: 29 de dezembro de 2016;

AVA MOODLE: **Ambiente virtual de aprendizagem Moodle**. Disponível em:<
<https://moodle.org/>>. Acesso em 29 de dezembro de 2016;

AVA SAKAI: **Ambiente virtual de aprendizagem Sakai**. Disponível em:<
<https://www.sakaiproject.org/>>. Acesso em 29 de dezembro de 2016;

AVA TALENTLMS: **Ambiente virtual de aprendizagem TalentLMS**. Disponível em:<
<https://www.talentlms.com/index/aff:getapp>>. Acesso em 29 de dezembro de 2016;

BEAL, A.; **Gestão Estratégica da Informação**. São Paulo: Atlas, 2004;

BIONE, A. A. G.; **Solução distribuída de baixo custo para utilização em ambientes virtuais de aprendizagem**. Recife, Universidade Federal Rural de Pernambuco. 2013. 69 pag. Dissertação, Pós-Graduação em Gestão e Tecnologia em EAD, Recife, 2013.

BJELJAC, P; ZECEVIC, I; PERISIC, B; **Integrating Learning Management System and Faculty Information System – Service Oriented Approach**. Disponível em:
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7160375>. Acesso em 10 de novembro de 2016;

CARVALHO, R. S.; **Sistemas de Gestão de Aprendizagem e Sistemas de Gestão Acadêmica: Avaliados pela ótica Docente**. Recife. Universidade Federal de Pernambuco. 2010. 191 pag. Dissertação, Pós-Graduação em Ciência da Computação. Recife, 2010.

COSTA, M. C. B.; **Bass: Um arcabouço de software para integração de sistemas de informação**. 2008. 89 f - Centro UFMG, Belo Horizonte, 2008.

CORRÊA, J.; (organizadora). **Educação a distância: orientações metodológicas**. 1 edição – Porto Alegre, Artmed, 2007;

CÔRTEZ, P. L.; **Administração de sistemas de informação**. São Paulo: Saraiva, 2008.

CUMMINS, F. A. **Enterprise integration: an architecture for enterprise application and systems integration**. New York: John Wiley & Sons, 2002;

Desenvolvimento Moodle - **Estrutura do banco de dados e principais API, Moodle Moot 2013**, disponível em:
<<http://www.moodlemoot.com.br/2013/moodle/mod/resource/view.php?id=50>> Acesso em 25 abril de 2016;

Dicionário Michaelis. <Disponível em: <http://michaelis.uol.com.br/>> Acesso em 15 abril 2016;

FARAVETTO, F.; **Integração de sistemas utilizando Data Warehouse: possibilidades para sistemas de informação de controle de estoques**. Disponível em:
<<http://revista.feb.unesp.br/index.php/gepros/article/viewFile/104/43>>. Acesso em 12 maio 2015;

GAVA, T. B. S.; **Engenharia de software**. Vitória: IFES, 2009. 75 p.: il. ISBN 978-85-62934-55-1. Instituto Federal do Espírito Santo;

FILATRO, A.; **Design Instrucional contextualizado**. 2ª edição – São Paulo, Editora Senac São Paulo, 2007;

FUGITA, H. S, HIRAMA, K.; **SOA: Modelagem, análise e design**. Rio de Janeiro: Editora Elsevier, 2012;

FURTADO, U. M.; **Modelo de integração adaptável entre Ambientes Virtuais de Aprendizagem e Sistemas de Gestão Acadêmica baseado em Arquitetura Orientada a Serviços**. Mossoró, Universidade Federal Rural do Semi-Árido. 2016. 88 pag. Dissertação, Pós-Graduação em Ciência da Computação, Mossoró, 2016;

FURTADO, U. M.; LIMA, R. W.; F. A. M. G.; DANTAS, C. C. F.; **A importância de integrar ambientes virtuais de aprendizagem com sistemas de gestão acadêmica – Um modelo de integração**. Disponível em <<http://esud2015.uneb.br/tpl/mobile/index.html#p=1>>. Acesso em 25 de Abril de 2016;

GIL, A. **Métodos e técnicas de pesquisa social**. 6. Edição. São Paulo: Atlas, 2008;

_____. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2007.

GOMES, D. A.; **Web Services SOAP em Java: Guia prático para o desenvolvimento de web services em Java** 2 edição – São Paulo: Novatec 2014;

GORDON, S. R; GORDON, J. R.; **Sistemas de Informação – Uma abordagem gerencial**. 3.

edição – Rio de Janeiro, RJ, LTC, 2006;

GUEDES, G. T. A.; **UML 2: Uma abordagem prática**. 2. Edição – São Paulo, SP, Novatec Editora, 2011;

HOHPE G. & WOOLF, B. **Enterprise integration patterns: Designing, building, and deploying messaging solutions**. Boston MA. Addison-Wesley Professional, Fifteenth printing, May 2011;

JUNIOR, H. E.; **Computação em Nuvem com o office 365**. 1 edição - São Paulo: Novatec Editora, 2015;

KANESHIMA, E.; **Utilização de serviços na integração de aplicações empresariais**. São Carlos, Universidade de São Paulo. 2013. 157 pag. Dissertação, Pós-Graduação em Ciência da Computação e Matemática Computacional, São Carlos, 2013.

LAUDON K. C. & LAUDON J. P. **Management Information Systems: managing the digital firm**. Prentice Hall, 2012;

LAURINDO, F. J. B.; ROTONDARO, R. G. (coordenadores). **Gestão Integrada de Processos e da Tecnologia da Informação**. 1 edição – São Paulo, Atlas, 2011;

LECHETA, R. R.; **Web Services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google**. 1 edição - São Paulo, Novatec Editora Ltda, 2015;

LEWIS, G.A.; LAGO, P.; AVGERIOU, P. **A Decision Model for Cyber-Foraging Systems** <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7516811>> 2016 13th Working IEEE/IFIP Conference on Software Architecture. 2016. Acesso em 20 de novembro de 2016;

LIU, Y.; WANG, Q.; ZHUANG, M.; ZHU, Y.; **Reengineering Legacy Systems with RESTful Web Service** <<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4591667>> Annual IEEE International Computer Software and Applications Conference. 2008. Acesso em 20 de novembro de 2016;

LIMA, G. A. F; NETO, A. F. R.; **Turma Virtual do SIGAA como Ferramenta de Apoio ao Ensino**. Natal, UFRN, 2009;

LOPES, E. M. M.; **EAD: Mudança De Paradigmas Pedagógicos**. Disponível em: <<http://www.administradores.com.br/artigos/carreira/ead-mudanca-de-paradigmas-pedagogicos/23386/>> Acesso em: 15 maio 2015;

MACHADO, R. P.; **Desenvolvimento de Software III: programação de sistemas web orientada a objetos em Java**. Porto Alegre: Bookman, 2016;

MARZULLO, F. P.; **SOA na prática: Inovando seu negócio por meio de soluções orientadas a serviços**. São Paulo: Novatec Editora, 2009;

MITCHELL, L. J.; **Web Services em PHP: APIs para a web moderna**. São Paulo: Novatec Editora, 2013;

- MIZUKOSHI, A. M.; GARCIA R. **Cloud Computing, um paradigma.** Disponível em: <http://intertemas.unitoledo.br/revista/index.php/ETIC/article/viewFile/3837/3597> . Volume 8, N. 8 (2012). Acesso em: 15 maio de 2016
- MOORE, M. & KEARSLEY, G.; **Educação a Distância: uma visão integrada.** Tradução Roberto Galman. São Paulo: Thomson Learning, 2007;
- MULLER, C. C.; **EAD nas Organizações.** 1 edição – Curitiba, Paraná, IESD, 2012;
- O'BRIEN, J. **Sistemas de informações e as decisões gerenciais na era da Internet,** São Paulo: Saraiva, 2004;
- PEWRESEARCHCENTER, **The future of cloud computing.** Disponível em: <http://www.pewinternet.org/2010/06/11/the-future-of-cloud-computing/>>. Acesso em 20 de outubro de 2016.
- PORTAL DEVMEDIA, <http://www.devmedia.com.br/integracao-de-sistemas-tipos-e-niveis-de-integracao-revista-engenharia-de-software-magazine-46/23814>> Acesso em 10 de junho de 2015;
- PRESSMAN, R. S. **Engenharia de Software: Uma abordagem profissional;** Tradução Ariovaldo Griesi; revisão técnica Reginaldo Arakaki, Julio Arakaki, Renato Manzan de Andrade – 7. Ed. –Porto Alegre; AMGH, 2011;
- REZENDE, D. A.; **Tecnologia da Informação e Planejamento Estratégico.** São Paulo: Brasport, 2008;
- RICHARDSON, L; RUBY, S.; **RESTful Serviços Web: web services para o mundo real,** Editora: O'reilly. 2007;
- SANTOS, G. R.; **Cloud Computing – Data Center Virtualizado: Gerenciamento – Monitoramento – Segurança:** Rio de Janeiro, Editora Ciência Moderna Ltda, 2015
- SILVA, C. A. B.; **Arquitetura empresarial: Um estudo de caso sobre a integração entre a plataforma Moodle o SIGAA na UFRN,** 2012
- SILVA, R. S.; **Moodle para autores e tutores.** 2 edição – São Paulo, Novatec Editora, 2011;
- SOUSA, K. S., FANTINATO, M. **Explorando a Engenharia de Requisitos Orientada a Serviços: Uma Revisão Sistemática da Literatura.** Anais do IX Simpósio Brasileiro de Sistemas de Informação (SBSI): Trilhas Técnicas, v. 1. Porto Alegre – RS: Sociedade Brasileira de Computação – SBC, pp. 272-283, 2013;
- TANEMBAUM, A. S. **Sistemas Distribuídos: princípios e paradigmas.** Tradução Arlete Simille Marques; Revisor técnico Wagner Zucchi. 2. Ed – São Paulo: Pearson Prentice Hall, 2007;
- VELTE, A. T., VELTE, T. J., ELSENPETER R.; **Cloud Computing: Computação em Nuvem – Uma abordagem prática.** 1. Ed – Rio de Janeiro: Alta Books Editora, 2011;
- WHATIS, **Definitions, Dictionary for Internet & Computer Technologies.** Disponível em:

<www.whatis.com>. Acesso 21 de maio de 2016.

YASAR, O.; ADIGUZEL, T. **A working sucessor of learning management system: SLOODLE**. Procedia Social and Behavioral Sciences, Stanbul, 22 January 2010.

**APÊNDICE A – QUESTIONÁRIO SEMIESTRUTURADO PARA
REALIZAÇÃO DE ENTREVISTA.**

**UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DE EDUCAÇÃO A DISTÂNCIA E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA E GESTÃO EM
EDUCAÇÃO A DISTÂNCIA**

QUESTÕES PARA ORIENTAÇÃO DE ENTREVISTA

ABORDAGEM

| INTEGRAÇÃO ENTRE SGA E AVA | |
|-----------------------------------|--|
| Q-1 | A modalidade EAD está presente em quantos municípios? |
| Q-2 | Quantos cursos na modalidade EAD estão sendo ofertados nesse semestre? |
| Q-3 | Qual a quantidade de discentes matriculados nestes cursos? |
| Q-4 | São praticados cursos de nível superior (bacharelado ou licenciatura) ou mesmo pós-graduações na modalidade EAD? |
| Q-5 | Qual o sistema utilizado como ambiente virtual de aprendizagem para modalidade EAD e qual o sistema utilizado como gestor acadêmico? |
| Q-6 | Existe algum mecanismo de integração que permita que estes sistemas troquem informações entre si de forma automatizada? |

APÊNDICE B – QUESTIONÁRIO SEMIESTRUTURADO PARA LEVANTAMENTO DE REQUISITOS.

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO UNIDADE ACADÊMICA DE EDUCAÇÃO A DISTÂNCIA E TECNOLOGIA PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA E GESTÃO EM EDUCAÇÃO A DISTÂNCIA

Este questionário faz parte uma pesquisa, em andamento, para a obtenção da titulação de Mestre em Gestão e Tecnologia em Educação a Distância pela Universidade Federal Rural de Pernambuco (UFRPE), e tem por objetivo geral estabelecer requisitos de funcionalidade que possibilitem integrar os ambientes de gestão acadêmica e o ambiente virtual de aprendizagem.

O objetivo é prover um modelo para integração destes sistemas a fim de otimizar o tempo que é gasto na operacionalização de ambos. Para a viabilizar este estudo, a sua participação é fundamental. Desta forma, solicitamos a sua colaboração no preenchimento deste questionário, que tem por objetivo coletar informações para pesquisa em curso com sua experiência com sistema de gestão acadêmica e ambiente virtual de aprendizagem. Agradecemos a sua contribuição, e em caso de dúvida, nos colocamos à disposição para esclarecê-las.

1. Sobre o Sistema de Gestão Acadêmica (SGA):

1.1. Qual sistema de gestão acadêmica sua instituição adota? Há quanto tempo é utilizado?

1.2. O sistema de gestão acadêmica utilizado atende plenamente as necessidades de sua instituição? Como?

1.3. Você faz uso de todas as funcionalidades disponíveis do sistema de gestão acadêmica? Por quê?

1.4. Quais funcionalidades são mais utilizadas e como?

1.5. Descreva como é processo de matrícula acadêmica dos discentes a cada semestre/período letivo no sistema de gestão acadêmica.

1.6. Quais as dificuldades encontradas em gerir todo processo de informações acadêmicas no sistema de gestão acadêmica utilizado pela sua instituição?

2. Sobre o Ambiente Virtual de Aprendizagem (AVA):

2.1. Qual Ambiente Virtual de Aprendizagem sua instituição adota? Há quanto tempo é utilizado?

2.2. O Ambiente Virtual de Aprendizagem utilizado atende plenamente as necessidades de sua instituição? Como?

2.3. Você faz uso de todas as funcionalidades disponíveis do Ambiente Virtual de Aprendizagem? Por quê?

2.4. Quais funcionalidades são mais utilizadas e como?

2.5. Descreva como é processo de cadastramento acadêmico dos discentes a cada semestre/período letivo no Ambiente Virtual de Aprendizagem.

2.6. Quais as dificuldades encontradas em gerir todo processo de informações acadêmicas no s Ambiente Virtual de Aprendizagem utilizado pela sua instituição?

3. Sobre a integração do Ambiente Virtual de Aprendizagem (AVA) e o Sistema de Gestão Acadêmica (AVA)

3.1. Como a integração entre os sistemas envolvidos pode contribuir para operacionalização dos mesmos?

3.3. Como você imagina o processo de integração “ideal” entre o ambiente virtual de aprendizagem e o sistema de gestão acadêmica?

APÊNDICE C – FORMULÁRIO DE VALIDAÇÃO PARA MODELO DE INTEGRAÇÃO ENTRE SGA E AVA.

**UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DE EDUCAÇÃO A DISTÂNCIA E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA E GESTÃO EM
EDUCAÇÃO A DISTÂNCIA**

Sobre o colaborador:

01. Qual seu tempo de experiência em desenvolvimento e integração de sistemas?

- 05 anos ou mais
- 04 anos
- 03 anos
- 02 anos
- 01 anos ou menos

02. Qual das abordagens para integração de sistemas você utiliza ou já utilizou?

- Web Services SOAP
- Web Service REST
- Orientada a dados
- Orientada a métodos
- Orientado a Interface de Usuário
- Outras

Sobre o modelo proposto:

03. Em uma escala de 1 a 10 como você avalia o modelo proposto. Utilize a escala de conceitos abaixo:

- 9 e 10 - Ótimo
- 7 e 8 - Bom
- 5 e 6 - Regular
- 3 - 4 - Ruim
- 1 e 2 – Péssimo

04. Quanto ao seu nível de aceitação para o modelo proposto. Utilize a escala de conceitos abaixo:

- Muito útil
- Útil
- Pouco útil

05. Quanto à possibilidade de adoção do modelo proposto. Utilize a escala de conceitos abaixo:

- Totalmente possível
- Moderadamente possível
- Indiferente
- Moderadamente contrário
- Totalmente contrário

06. O modelo ajuda na avaliação da arquitetura e, particularmente, na identificação e mitigação de riscos?

- Sim
- Não
- Indiferente

07. O que mais poderia ser útil para o modelo de integração proposto?
