



Abordagens e Tecnologias para Integração de Sistemas: Um Estudo de Caso na Universidade Federal de Lavras

C. M. Garcia, R. Abilio – *Diretoria de Gestão de Tecnologia da Informação (DGTI)*
N. Malheiros – *Departamento de Ciência da Computação (DCC)*
Universidade Federal de Lavras (UFLA)
{*cristiano.garcia,ramon.abilio*}@*dgti.ufla.br*, *neumar@dcc.ufla.br*

Resumo—O objetivo desse trabalho foi analisar abordagens e tecnologias para integração entre diversos sistemas de informação e serviços de rede em um ambiente acadêmico. Foi realizado um estudo de caso sobre o histórico da integração de sistemas na Universidade Federal de Lavras. Os dados foram coletados via questionários e análise de documentação. Nessa análise, identificaram-se 4 fases distintas, quanto às tecnologias e abordagens utilizadas. Além disso, foram discutidos os benefícios e as desvantagens encontradas em cada fase. A principal contribuição deste trabalho é a análise de diferentes abordagens para integração entre sistemas de informação e outros serviços.

Palavras-Chave — Arquitetura Orientada a Serviços, Integração de Sistemas, Web Services

Abstract—This case study aims to analyze the integration's approaches and technologies among information systems and services in an academic environment. It has been done a study on the integration history in the Federal University of Lavras. The data had been gathered via questionnaires and documentation analysis. In this analysis, 4 distinct phases were specified. Besides, the advantages and disadvantages of each phase were discussed. The main contribution of this work is the analysis of different integration approaches among information systems and other services.

Index Terms—Information Systems Integration, Service-Oriented Architecture, Web Services.

I. INTRODUÇÃO

Instituições públicas e privadas, das mais diversas áreas e tamanhos, possuem sistemas de informação para auxiliar na gerência dos processos. Em ambientes corporativos, é comum encontrar um cenário com tipos diferentes de sistemas, tanto nos níveis operacional e gerencial, quanto no nível estratégico [23]. Com isso, os dados são compartilhados mesmo entre sistemas que não estão no mesmo nível administrativo, caracterizando, assim, dois tipos de integração:

a vertical (entre sistemas de diferentes níveis) e a horizontal (entre sistemas de mesmo nível).

Existem diversas classificações para integração de sistemas, como por exemplo, as centradas no nível de implementação e as com foco na organização e seus processos. A “Informação Centralizada” [17] é um exemplo de abordagem centrada no nível de implementação e a *Business-to-Business* (B2B) [13] é um exemplo de abordagem com foco na organização e seus processos.

Não importando a abordagem escolhida, uma das formas de implementação de integração entre sistemas é utilizar a Arquitetura Orientada a Serviço (*Service Oriented Architecture* – SOA) [19]. A utilização da SOA com o propósito específico da integração de sistemas é chamada de *Service-Oriented Integration* (SOI) [12] e um meio para a implementação da SOI é utilizando *Web Services*, que representam uma visão sobre a programação distribuída e a disponibilização de recursos, fortemente ligada à Internet [22]. Diversas tecnologias e padrões, como *Simple Object Access Protocol* (SOAP), arquitetura *Representational State Transfer* (REST) e *Java API for Web Services* (JAX-WS¹), fornecem suporte para a implementação de *Web Services* [3].

A utilização de *Web Services* para a integração entre sistemas pode se estender, também, às instituições acadêmicas. Na Universidade de Açores (Portugal), um conjunto de *Web Services* foi desenvolvido para a otimização de tarefas críticas, que envolvem informações financeiras e estratégicas [4]. Na Universidade Federal de Pelotas (UFPEL - Brasil), foi também desenvolvido um conjunto de *Web Services* para a consistência de informações entre seus sistemas e serviços, como o restaurante universitário, o Ambiente Virtual de Aprendizado e a rede sem fio institucional [1].

A Universidade Federal de Lavras (UFLA), objeto deste estudo, vem tendo um crescimento acentuado em sua comunidade. Até junho de 2013, em torno de 16000 alunos

¹ <http://docs.oracle.com/javase/7/docs/technotes/guides/xml/jax-ws/>

estavam matriculados na Universidade, e por volta de 1600 servidores ocupavam cargos na Instituição [25]. Esse número tende a crescer com a criação de novos cursos [24].

Segundo o Plano Diretor de Tecnologia da Informação 2011/2012 [8], até 2011, a UFLA possuía 24 sistemas de informação mantidos pela Diretoria de Gestão de Tecnologia da Informação (DGTI). Cada um destes sistemas possui singularidades quanto ao seu objetivo (acadêmico, administrativo e de apoio), ao tipo de sistema gerenciador de bancos de dados, plataformas e tecnologias utilizadas no desenvolvimento, além de terem sido desenvolvidos por diferentes empresas.

Em 2013, uma “família” de sistemas de informação institucionais integrados começou a ser implantada na Universidade. Esses sistemas são desenvolvidos com tecnologias diferentes das até então presentes na Instituição e isso adiciona ainda mais complexidade no contexto da integração.

Além dos sistemas de softwares mencionados, a Universidade ainda oferece serviços de rede como email e acesso *wireless* para toda a comunidade acadêmica, existindo a necessidade de integração entre serviços e sistemas do ambiente universitário.

Essa heterogeneidade torna a integração de sistemas um grande desafio. O principal objetivo deste trabalho foi realizar um estudo de caso para analisar as soluções de integração entre sistemas de informação na UFLA, destacando os benefícios e as desvantagens de cada solução utilizada de forma a fazer com que este desafio seja transposto da melhor forma. Uma das contribuições deste trabalho é a análise de diferentes abordagens para integração entre sistemas que poderá auxiliar profissionais a escolher a abordagem mais adequada para seu cenário.

O presente trabalho está organizado da seguinte forma: na Seção 2, é apresentada uma breve introdução sobre integração entre sistemas de informação; após, na Seção 3, a metodologia é apresentada; na Seção 4 e na Seção 5, são apresentados os resultados e as conclusões, respectivamente.

II. INTEGRAÇÃO ENTRE SISTEMAS DE INFORMAÇÃO

O uso de Sistemas de Informação (SI) vem crescendo expressivamente e tende a crescer ainda mais nos próximos anos [2]. O poder de processamento agregado nas grandes organizações também vem aumentando na medida em que os computadores diminuem de tamanho e preço [14]. Com isso, instituições públicas e privadas nas mais diversas áreas, das maiores até as mais simples já possuem computadores, com sistemas de informação para auxiliar na gerência do negócio e automação de tarefas rotineiras [18].

A utilização de computadores e sistemas de informação é mais do que a automatização de tarefas de nível operacional. A Tecnologia da Informação (TI) passou a ser uma ferramenta estratégica, pois a empresa que utiliza os recursos de TI de modo eficaz, integrando as estratégias de TI às do negócio, é

capaz de obter vantagem competitiva [15].

Existem razões que podem levar empresas a contratar ou desenvolver soluções de integração de sistemas. Tais razões podem ser [7]:

- Estender a tecnologia já existente para flexibilizar e reduzir custos na implementação de novos serviços;
- Permitir integração com *stakeholders*, expandindo assim o alcance de serviços;
- Integrar informações comuns em diferentes bases de dados, resultantes de fusões, aquisições ou sistemas legados.

A integração de sistemas corporativos é uma tarefa complicada e que envolve riscos, pois cada organização possui suas características e necessidades de integração [17]. Uma organização pode realizar a integração de sistemas considerando diferentes focos, como na implementação ou na organização e seus processos.

A. Classificações da Integração de Sistemas

A integração de sistemas de informação pode ser classificada, por exemplo, considerando-se o nível de implementação [17] ou a organização e seus os processos [13]. Quanto ao nível de implementação, a integração de sistemas pode ser classificada como [17]:

- Aplicações Compostas: aplicações integradas com o uso de API². Nesta integração, a API funciona com um conector entre sistemas;
- Informação Centralizada: ocorre quando diferentes sistemas tem acesso à mesma base de dados, compartilhando também os metadados;
- Sistemas Integrados de Gestão: sistemas fechados e compostos por módulos internos independentes. Esta integração é realizada, comumente, a nível de código-fonte.

Com foco na organização e seus processos, a integração pode ser classificada como [13]:

- *Data Replication* (Replicação de Dados): a integração acontece no nível de informação. As bases são distribuídas e mantidas atualizadas e sincronizadas;
- *Business-to-Business (B2B) Integration* (Integração Negócio-Negócio): extrapola os limites empresariais. Representa a disponibilização de funções entre diferentes organizações. Embora os demais conceitos também possam ser aplicados à integração B2B, o uso de redes externas pode levantar novas questões a serem analisadas [13];

² *Application Programming Interface*, ou Interface de Programação de Aplicação

- *Service-Oriented Architecture* (Arquitetura Orientada a Serviço): os sistemas disponibilizam funções como serviços, que no contexto de computação, significa “função bem definida e universalmente disponível”.

Apesar dos diferentes focos, as taxonomias encontradas possuem semelhanças entre si [13][17]. Por exemplo, as abordagens “*Business-to-Business (B2B) Integration*” e “Aplicações Compostas” podem utilizar de API para compartilhamento de informações e, assim, efetivar uma possível integração entre sistemas.

B. Arquitetura Orientada a Serviço

A arquitetura de um software representa a estrutura que abrange os componentes do *software*, suas propriedades externamente visíveis e as relações entre ambos [20]. Arquitetura Orientada a Serviço (*Service-Oriented Architecture* (SOA)) vem sendo considerada um dos principais paradigmas no projeto de sistemas distribuídos, inclusive dando origem à uma ramificação da Engenharia de Software chamada Engenharia de Software de Serviço [27].

A SOA é um paradigma que visa a organização e utilização de recursos que podem estar sob controle de diferentes proprietários, pela disponibilização de um meio uniforme de oferta, descoberta, interação e uso de funcionalidades para produção de efeitos desejados e consistentes [19]. Este paradigma é capaz de oferecer uma série de benefícios, como o controle do crescimento de sistemas, o provimento e utilização de serviços em escala global e redução de custos em cooperações organização-organização [26].

Utilizando SOA com o propósito de integração, há a *Service-Oriented Integration* (SOI), cujo objetivo principal é criar uma integração entre múltiplos sistemas, modificando pouco ou nada suas implementações [12]. Essa técnica expõe dados, funcionalidades e processos para serem consumidos pelos sistemas participantes da integração. Existem diversas abordagens para a SOI com foco na integração entre sistemas já existentes [12], entre as quais pode-se destacar:

- *Service* (Serviços): utiliza uma camada de serviços entre os sistemas existentes e os consumidores dos serviços;
- *Process Integration* (Integração de Processos): integra processos em um ambiente empresarial, sugerindo a utilização na integração de pequenos processos dentro de grandes processos, com interação humana ou não;
- *Data Integration* (Integração de Dados): se refere à uma abordagem que gerencie a complexidade de modelos de dados em diferentes aplicações.

Web Services representam um meio de implementar a SOI, fornecendo uma interface de serviço que permite que consumidores interajam com provedores de serviço [5]. As

formas de implementação de *Web Services* mais comumente encontradas são *Simple Object Access Protocol* (SOAP) e *Representational State Transfer* (REST), citadas também como solução para integração de processos entre organizações [29].

SOAP é um protocolo que utiliza a linguagem *Web Services Description Language* (WSDL), baseada em XML, para descrever funcionalidades oferecidas por um *Web Service* [29]. O SOAP provê um padrão básico de comunicação, no qual cada operação é representada por seu terminal, descrito no documento XML enviado na requisição, ao invés de um método *HyperText Transfer Protocol* (HTTP), como utilizado no REST [29].

REST é uma abstração dos princípios que fazem a *World Wide Web* (WWW) escalável [29]. Permite fornecer serviços identificados por um *Uniform Resource Identifier* (URI), com a utilização de métodos HTTP 1.0: GET, POST, DELETE e PUT. O uso destes métodos define a operação a ser feita: GET lista registros, POST insere novo registro, DELETE remove um registro, e PUT atualiza um registro [10]. Um URI é uma forma uniforme de identificação de recursos em rede. O tipo mais conhecido de URI é o *Uniform Resource Location* (URL) [29].

Como REST não se trata de um protocolo, como o SOAP, mas, sim, de uma arquitetura [10], o retorno das chamadas pode ser formatado conforme os requisitos da aplicação, por exemplo, utilizando *Javascript Object Notation* (JSON) ao invés de XML.

O estudo sobre as formas de classificação da integração de sistemas permitiu a identificação das abordagens utilizadas para a integração dos sistemas de informação e serviços de redes de computadores da Instituição. Além disso, os conceitos relacionados à Arquitetura Orientada a Serviço permitiram entender a arquitetura utilizada na quarta fase do histórico de integração de sistemas da UFLA.

III. METODOLOGIA

Para manter a infraestrutura de redes de computadores, hardware e software, a UFLA possui uma Diretoria de Gestão de Tecnologia da Informação (DGTI). A DGTI está subdividida em cinco coordenadorias. Uma delas é a Coordenadoria de Sistemas de Informação – CSI, responsável pela “definição, análise, programação, implantação, manutenção e documentação de sistemas de informação dos órgãos administrativos da Instituição”. Dentre os objetivos estratégicos da DGTI, alinhados à responsabilidade da CSI, estão [8]: i) Melhorar o gerenciamento de projetos na DGTI; ii) Integrar os sistemas de informação utilizados na Instituição; e iii) Melhorar a qualidade de desenvolvimento e aquisição de software.

Este trabalho foi realizado na CSI, entre Agosto e Outubro de 2014, junto aos responsáveis diretos pela integração de sistemas e serviços no ambiente universitário. O trabalho foi desenvolvido em três etapas:

Etapa 1: Identificação dos sistemas de software da Universidade, destacando-se as tecnologias empregadas, os objetivos dos sistemas de software e a dependência entre eles;

Etapa 2: Realização de estudo de caso por meio de questionário e pesquisa documental. O objetivo desse estudo foi identificar as abordagens de integração, a motivação para o uso de tais abordagens, as expectativas com o uso das mesmas e discutir vantagens e desvantagens de cada uma das abordagens. O questionário utilizado possuiu onze perguntas:

1. Quais foram as primeiras necessidades de integração entre os sistemas na Universidade?
2. Quais sistemas foram os primeiros sistemas e serviços a serem integrados? Quais informações eram necessárias serem integradas?
3. Quais foram as abordagens utilizadas? Quais as razões de terem escolhido tais abordagens?
4. Quais foram as expectativas com a integração entre os sistemas?
5. Quais as vantagens e desvantagens encontradas nessas abordagens?
6. As expectativas foram alcançadas?
7. Houve alteração (adição / mudança) em relação aos sistemas que estavam integrados?
8. Quais são os sistemas e serviços integrados atualmente (2014)?
9. Por que a mudança na estratégia de integração? Qual a motivação para isso?
10. Quais as expectativas com a nova abordagem?
11. Quais as vantagens e desvantagens encontradas nas abordagens atuais?

O questionário foi enviado por e-mail para a equipe responsável pela integração entre sistemas na Universidade, na época formada por dois funcionários. Durante a análise das respostas, eles foram questionados pessoalmente para fornecer mais explicações. As respostas foram agrupadas, para que fosse possível estabelecer a ordem cronológica dos acontecimentos e para que as informações obtidas pudessem ser confrontadas.

Etapa 3: Análise das informações coletadas e da evolução da integração dos sistemas na Universidade. Foi realizada uma discussão sobre as informações obtidas e foram feitas sugestões para possíveis melhorias.

IV. RESULTADOS E DISCUSSÃO

Nesta Seção, são apresentados os sistemas de software e serviços que estão integrados e uma análise das características

da integração entre eles, abordando vantagens, desvantagens, características e tecnologias empregadas.

A. *Sistemas de Software e Serviços na Universidade*

A Universidade possuía, até 2011, 24 sistemas aplicados à gestão de recursos e serviços. No entanto, nem todos os sistemas estavam integrados. A razão para isso era, principalmente, que a utilização de alguns sistemas era interna a alguns departamentos. Portanto, deu-se ênfase nos sistemas que foram ou estão em integração até 2014. Esses sistemas foram agrupados de acordo com seu objetivo:

1. Acadêmico: sistemas de software cujo objetivo principal é gerenciar dados de cursos, disciplinas, docentes, discentes, pesquisa, assistência estudantil e atividades de extensão e cultura;
2. Administrativo: sistemas cujo objetivo principal é auxiliar na gestão de pessoas e/ou no planejamento e na gestão da Universidade; e
3. Apoio: sistemas utilizados para gerenciar usuários e controles de acesso.

Os sistemas acadêmicos são:

- CPPD: Sistema de cálculo de progressão de docentes utilizado até 2011;
- SIG: Sistema institucional de controle acadêmico. Ele é utilizado por diversas unidades, por exemplo, para controle de: processo de chamadas a candidatos; alunos; presença; matrizes curriculares; cursos; períodos letivos; processo de renovação de matrícula; lançamento de notas; e emissão de documentos, como histórico escolar e atestados;
- Pergamum: Sistema integrado de bibliotecas utilizado no gerenciamento do acervo, empréstimos de obras, entre outras atividades da Biblioteca da Universidade;
- SIGAA: Sistema que auxilia no controle de pesquisas e produção intelectual. A partir de 2015, será utilizado também no controle de cursos e estudantes de Pós Graduação.

Os sistemas administrativos são:

- SIGH: Sistema utilizado para atividades relacionadas a Recursos Humanos, como marcação e alteração de férias, cálculos de aposentadoria, controle de frequência, concursos, capacitações, avaliações funcionais;
- SIPAC: Sistema que integra atividades como requisição de materiais, manutenção, gerência de compras, licitações, patrimônio, contratos e convênios.

Os sistemas de apoio são:

- CIN_Cadastro: Sistema de apoio e controle de fotos de cartões de identificação pessoal, emails de unidades, de pesquisadores, liberação de acesso do fecho eletrônico, entre outras atividades;
- SCIN: Sistema utilizado para abertura e gerenciamento de requisições à equipe de suporte em TI;
- RV3Acesso: Sistema para controle de catracas de controle de acesso presentes no Restaurante Universitário, Biblioteca e Moradia Estudantil;
- HCS: Sistema de ponto eletrônico, auxilia em atividades como gerência de digitais e registro de ponto;
- Veículos: Sistema de cadastro de veículos para controle de acesso ao campus utilizado até 2010.

O grafo com a necessidade de dados entre os sistemas integrados na Instituição pode ser visto na Fig. 1. Por exemplo, o sistema HCS compartilha dados de Servidores Técnicos obtidos do sistema SIGRH.

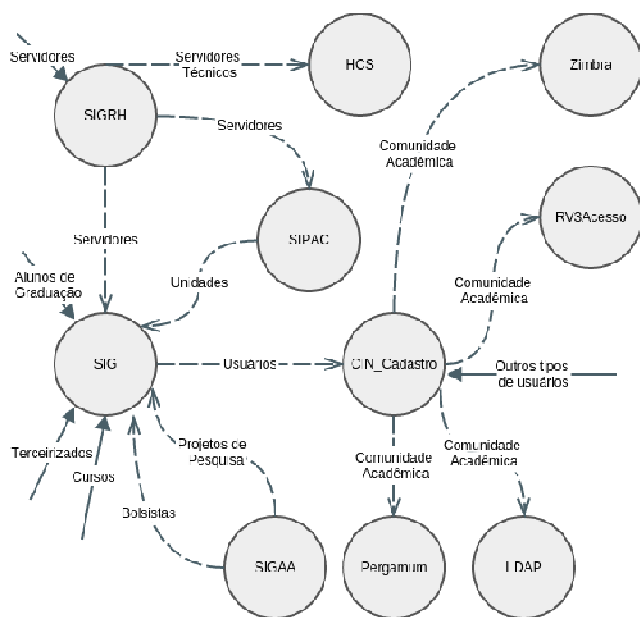


Fig. 1. Fluxo de Dados entre os Sistemas Integrados

A Tabela I apresenta os sistemas de software juntamente com a linguagem de programação utilizada, o sistema gerenciador de banco de dados (SGBD) e a plataforma. Por exemplo, o sistema SIG foi desenvolvido com PHP, utiliza um banco de dados MySQL e sua plataforma é Web. Basicamente, a CSI desenvolveu todos os sistemas que utilizam a linguagem PHP e houve a aquisição de licenças para utilização do Pergamum, do HCS e do RV3Acesso.

TABELA I
SISTEMAS DE SOFTWARE E RESPECTIVAS TECNOLOGIAS

Sistema	Linguagem	Banco de Dados	Plataforma
CPPD	PHP	MySQL	Web
SIG	PHP	MySQL	Web
Pergamum	Delphi/Java	SQL Server	Desktop/Web
SIGAA	Java	PostgreSQL	Web
SIGRH	Java	PostgreSQL	Web
SIPAC	Java	PostgreSQL	Web
CIN_Cadastro	PHP	MySQL	Web
SCIN	PHP	MySQL	Web
RV3Acesso	-	MySQL	Desktop
HCS	-	MySQL	Desktop
Veículos	PHP	MySQL	Web

Além dos sistemas de software, a Universidade oferece os serviços de rede, como e-mail (Zimbra³ - Sistema de mensagens e suíte de colaboração) e redes de acesso sem fio à Internet a toda a comunidade acadêmica. Existe ainda um serviço de diretório LDAP⁴ institucional para autenticação de usuários.

Os sistemas de informação e serviços de rede estão instalados em diversos servidores com diferentes sistemas operacionais (SO). Por exemplo, o servidor do Zimbra utiliza o SO Red Hat Linux e o LDAP está instalado em um servidor com SO Fedora Linux. Essa heterogeneidade impõe mais dificuldades, visto que cada SO possui características e configurações que podem ser diferentes entre si.

Durante o estudo, foi possível identificar sistemas presentes em 7 servidores. Destes 7 servidores, um (servidor1) utiliza o SO Red Hat, quatro (servidor3, servidor4, servidor5 e servidor7) utilizam o SO Fedora Linux, e dois (servidor2 e servidor6) contam com o sistema operacional Windows Server 2003. Esses servidores são identificados na Fig. 3 e na Fig. 4 juntamente com os sistemas de informação e com os serviços de rede que estão instalados nos mesmos.

Os sistemas e serviços na Universidade compartilham dados como nomes de usuário e endereços de e-mail. Todos os sistemas e serviços dependem, em diferente escala, dos dados gerenciados pelo CIN_Cadastro, SIG e SIGRH.

B. Análise histórica das abordagens de integração

Após a análise das respostas dos questionários, pôde-se agrupar os fatos em ordem cronológica e identificar quatro fases, como ilustrado na Fig. 2: 1) Pré-Integração: cenário anterior à primeira integração em 2006; 2) Fase 1: período entre meados de 2006 e meados de 2009; 3) Fase 2: período de meados de 2009 até Junho de 2014; e 4) Fase 3: período entre Junho e Outubro de 2014. Cada fase agrupa abordagens de integração diferentes e representa uma evolução na segurança, escalabilidade e flexibilidade na integração dos sistemas e serviços.

³ <https://www.zimbra.com/>

⁴ LDAP: Protocolo de aplicação aberto, que permite organizar os recursos de rede de forma hierárquica (<http://www.hardware.com.br/termos/ldap>)

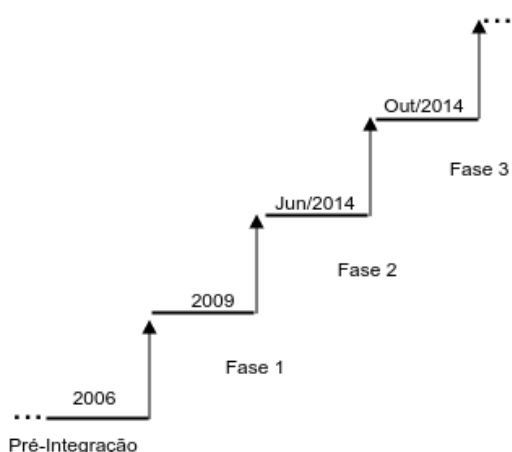


Fig. 2. Fases da Integração entre Sistemas de 2006 a 2014

1) Pré-integração

Até meados de 2006, nenhum sistema era integrado de forma automatizada, isto é, a transferência de dados entre os sistemas era feita manualmente. Mesmo as contas de e-mail eram criadas manualmente, no servidor Zimbra da época, sem informações como endereço, departamento, cargo ou mesmo o nome do usuário. Com isso, foi observada a necessidade da criação de uma base de dados única para o compartilhamento dessas informações.

Então, utilizando o SGBD MySQL, o banco de dados BD-UF foi criado e passou a centralizar dados, como: login, e-mail, nome, endereço, telefones, cursos e unidades de lotação dos servidores. Após a criação desta base de dados, sistemas de informação não institucionais (utilizados apenas em determinadas unidades) passaram a consultar dados atualizados na base central.

2) Fase 1

A primeira integração na Universidade teve o compartilhamento de informações entre os sistemas CIN_Cadastro, SCIN, Veículos, RV3Acesso, LDAP e Zimbra. As informações compartilhadas eram: nome, endereço, cargo e unidades de lotação dos servidores. O CIN_Cadastro permite o gerenciamento de usuários. Os dados desses usuários são replicados no LDAP e utilizados na criação do e-mail institucional no Zimbra. Os dados de servidores e discentes são replicados no RV3Acesso para controle de acesso ao Restaurante Universitário e Biblioteca. O SCIN e o Veículos utilizam os dados replicados em suas bases de dados para autenticação e relacionamento dos usuários com chamados de suporte e registro de veículos, respectivamente.

A abordagem para integração baseava-se no banco de dados. Se as bases compartilhassem MySQL como tecnologia, a integração era feita utilizando as próprias ferramentas do SGBD, como *triggers*, *functions*, *procedures*, *views* e *events*.

Para o RV3Acesso, foi utilizado o *engine* Federated⁵, que compartilha os dados por meio da criação de uma tabela virtual, que recebe os dados da tabela original.

Para os demais sistemas que participavam da integração, a abordagem era através do uso de *User-Defined Functions*⁶ (UDF). UDF são funções desenvolvidas em C e compiladas dentro do SGBD, utilizadas como funções nativas do MySQL. Utilizando-se esse recurso, foi possível automatizar as atualizações das bases de dados. No momento em que as tabelas eram atualizadas, as *triggers* relacionadas à elas eram disparadas automaticamente. Estas *triggers*, então, invocavam as UDF, que por sua vez, invocavam scripts PHP, que realizavam a integração com os outros sistemas. Na Fig. 3, é apresentado um diagrama dessa integração inicial.

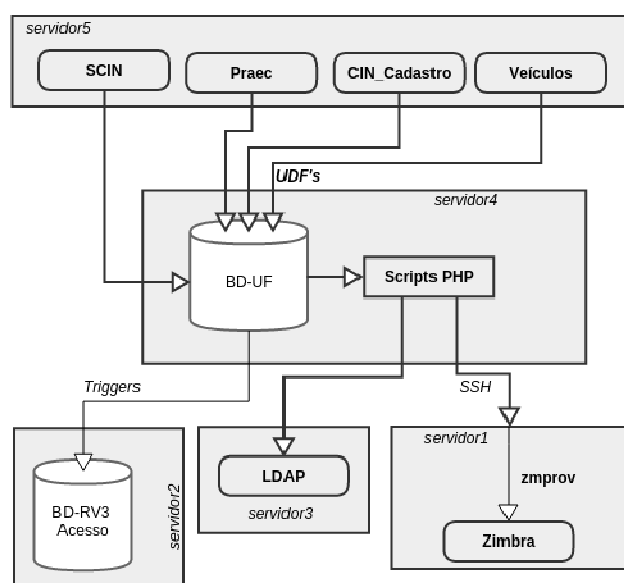


Fig. 3. Integração inicial entre os sistemas

A utilização do PHP foi decidida devido à habilidade dos profissionais que desenvolveram a integração inicial com esta linguagem. De acordo com as informações obtidas por meio do questionário, este foi o único critério para a decisão pela adoção do PHP para este fim.

De acordo com as abordagens centradas na organização e seus processos [13], esta abordagem se classificaria como *Data Replication*, pois os dados são replicados em diferentes bases (bancos de dados, LDAP e Zimbra) em um processo automático executado pelos scripts PHP.

A integração com o Zimbra era bastante complicada. Até sua versão 5.0, este sistema não possuía API de comunicação ou execução de comandos remotos, o que trouxe dificuldades à integração. Conforme ilustrado pela Fig. 3, o script PHP,

⁵ <http://dev.mysql.com/doc/refman/5.5/en/federated-storage-engine.html>

⁶ <http://dev.mysql.com/doc/refman/5.1/en/adding-udf.html>

invocado pelas UDFs, precisava estabelecer uma conexão SSH com o servidor onde o Zimbra estava instalado, para executar comandos da ferramenta Zmprov. Esta ferramenta oferecia a possibilidade de execução de comandos direto no Zimbra. Com este encadeamento de operações, a detecção de erros e tratamento de exceções se tornava bastante complexa. Outros sistemas, como o Pergamum, eram atualizados manualmente.

O uso de UDF trouxe a sensação de "instantaneidade" na integração, pois as atualizações eram executadas assim que uma *trigger* era disparada. Embora esta abordagem tenha atingido seus objetivos, as UDFs eram de difícil manutenção, pois eram programadas na linguagem C e compiladas dentro do SGBD. Sua utilização também dificultava a detecção de erros, quando estes aconteciam.

3) Fase 2

Após a primeira integração, ocorreram mudanças. Um novo sistema de gestão, o SIG, começou a ser desenvolvido e, progressivamente, agregando funcionalidades de antigos sistemas, posteriormente excluídos da integração. Este sistema, bem como o sistema Pergamum, passou a fazer parte da integração. Até Junho de 2014, os sistemas participantes da integração na UFLA eram: SIG, CIN_Cadastro, Pergamum, RV3Acesso, Zimbra e LDAP.

Com a entrada do SIG e seu rápido crescimento, surgiu uma série de novos problemas. A crescente utilização deste sistema fazia com que as tabelas fossem atualizadas mais frequentemente, aumentando o disparo de *triggers*, que por sua vez invocavam as UDF. Esta crescente demanda da utilização das UDF sobrecarregava o sistema. À época, uma atualização demorava de 3 a 5 segundos para ser propagada para Zimbra e LDAP. Estes sistemas eram integrados utilizando uma cadeia de chamadas, trazendo a possibilidade de problemas de difícil detecção em cada etapa, além de serem sensíveis às sobrecargas na rede.

A solução dada para esta situação foi colocar o banco de dados do SIG (BD-SIG) em um servidor dedicado. Feito isso, uma réplica assíncrona desta base foi configurada no mesmo servidor no qual se encontrava o BD-UF, e consultas e relatórios que requeriam mais processamento foram direcionados à ela. Com isso, o servidor de produção do BD-SIG ficou menos requisitado, amenizando os problemas com as UDF, que ficavam lentas com a sobrecarga do BD-SIG.

Mesmo com os problemas com as UDF amenizados, buscou-se uma alternativa a elas. Neste momento, o Cron - ferramenta que permite agendar a execução de tarefas em sistemas operacionais derivados do Unix - se mostrou uma alternativa interessante. A opção pela utilização do Cron trouxe vantagens como facilidade na configuração, utilização e manutenção. As chamadas a scripts PHP passaram a ser diretas, sem a utilização de UDF. A desvantagem desta abordagem era a falta de instantaneidade da atualização dos dados nos outros sistemas, embora não fosse considerado um problema, pois os sistemas eram atualizados em até 10

minutos após a atualização inicial ser registrada.

Esta abordagem utilizando o Cron foi implementada parcialmente. Sistemas críticos como LDAP e Zimbra não utilizavam essa abordagem. Com o foco no SIG, o funcionamento da integração nessa Fase acontecia da seguinte forma (como ilustrado na Fig. 4):

1. Alteração no SIG dispara modificação nos dados do BD-SIG;
2. BD-SIG original tem os dados replicados para uma cópia do banco de dados em um outro servidor;
3. *Triggers* na réplica do BD-SIG adicionam as alterações em um *buffer* de atualizações do BD-UF, tabela que armazena as atualizações a serem efetivadas, contendo dados como sistema e registro a serem buscados os dados que serão propagados para os outros sistemas da integração;
4. Um evento do MySQL, disparado a cada 10 minutos no BD-UF, gera registros específicos para atualização dos sistemas, definindo qual sistema deve ser atualizado, e em qual registro no BD-UF esta atualização deve se basear;
5. O Cron executa scripts PHP, cada um relativo a um sistema que deve ser atualizado. Os scripts buscam os dados na tabela de *buffer* no BD-UF e os propagam para o BD-Pergamum e BD-RV3Acesso;
6. Scripts PHP são executados também a partir de UDF do BD-UF, para propagar atualizações para LDAP e Zimbra.

Embora as UDF tenham passado a ser menos utilizadas, a abordagem da Fase 2 ainda usava esta tecnologia para integrar serviços críticos, como LDAP e Zimbra. Assim como a da Fase 1, a abordagem da Fase 2 atingiu seu objetivo, que era a integração dos sistemas na Universidade. Porém, a adição de novos sistemas à integração tal qual ela era nesta fase ainda trazia complexidades. Por exemplo, a necessidade do desenvolvimento de sua própria UDF, além de sua manutenção.

Mesmo com a substituição parcial das UDF pelo Cron, a integração não era escalável nem passível de monitoramento. Não havia uma política clara nem um padrão ou guia de como a adição de novos sistemas à integração deveria ser feita. O monitoramento da integração era realizado através de mecanismos programados no BD-UF, que enviavam e-mail quando alguma falha, como travamento na *procedure* de geração de registros, fosse detectada.

O crescimento da Universidade trazia demandas cada vez maiores e a adição de novos sistemas ao ambiente universitário era iminente, tornando a integração uma tarefa progressivamente mais complexa.

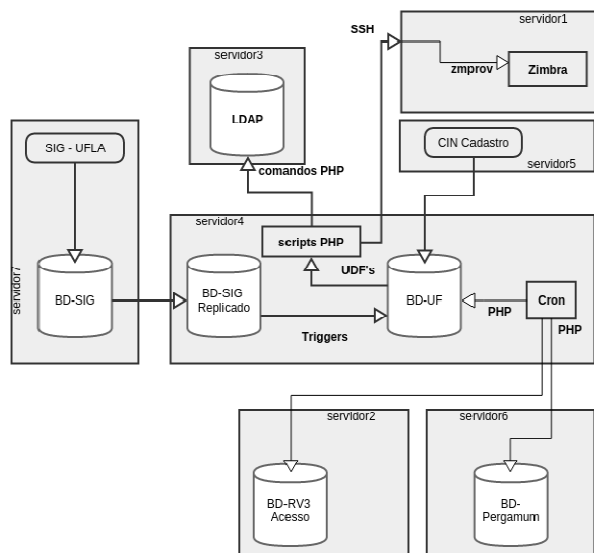


Fig. 4. Integração entre sistemas na Universidade na Fase 2

4) Fase 3

Na Fase 3, as UDF continuavam em funcionamento, mesmo com suas limitações quanto à performance e manutenibilidade. A atualização do SO no servidor em que se encontrava o BD-UF bem como a atualização da versão do SGBD MySQL de 4 para 5 trouxeram instabilidade às UDFs, como a incompatibilidade entre tipos de dados devido à modificação dos nomes. Com esta instabilidade, vieram também os já conhecidos problemas de tratamento de exceções e detecção de erros. Então, a arquitetura de integração foi repensada. Outras tecnologias de integração foram estudadas para a proposta de uma nova abordagem que suprisse as necessidades de alta escalabilidade e manutenibilidade. A SOA foi então uma alternativa promissora, trazendo a possibilidade de se efetivar uma integração com qualidade e com mecanismos que pudessem ser monitorados.

Esta estratégia visou garantir uma vantagem significativa sobre a abordagem anterior. Foram tomadas decisões sobre: segurança, linguagem de programação, relação provedor/serviço, protocolo e abordagem.

- Segurança: item primordial, visto que a integração manipularia dados pessoais, como nome, CPF, senhas, e-mails e endereços de milhares de pessoas. Então, seria necessária uma forma de segurança que restringisse o acesso apenas às pessoas ou sistemas autorizados. Neste quesito, o protocolo SOAP foi escolhido, pois o cliente precisa conhecer as operações que podem ser utilizadas por ele. Além

dessa características do SOAP, foi pensada uma implementação que limitasse a utilização de serviços a *tokens* autorizados, acessando a partir de endereços IP específicos;

- Linguagem de Programação: foi decidido que a arquitetura de integração seria desenvolvida em PHP, devido à familiaridade da maioria da equipe com a linguagem, além da existência de módulos nativos e não nativos que facilitariam a utilização do SOAP. A performance da linguagem não foi um critério para a escolha da mesma, visto que não ofereceria sobrecarga na base de dados como as UDF;
- Relação Provedor vs. Serviço: foi discutida tal relação no aspecto numérico, na qual foi feita a seguinte pergunta: “Seria mais interessante uma abordagem utilizando um único provedor e vários serviços ou vários provedores com poucos serviços?” A opção foi pela arquitetura com vários provedores e poucos serviços, que facilitaria a escalabilidade, isolando a integração por sistemas específicos. Essa opção também seria útil para a manutenibilidade da integração, pois quando um provedor precisasse ser parado para manutenção, os demais não seriam afetados;
- Protocolo: juntamente com o SOAP, foi definido que o retorno das chamadas aos serviços sempre seriam um objeto *Javascript Object Notation* (JSON) composto por quatro atributos: 1) ID: que carregaria o ID da mensagem e do sistema relacionado à integração; 2) uma mensagem; 3) um tipo, como por exemplo: “SUCESSO”, “ERRO BD”; e 4) um atributo *system*, que traria o nome do provedor requisitado;
- Abordagem: com uma visão centrada na implementação, optou-se pela abordagem de “Aplicações Compostas” [17], pois as aplicações seriam integradas através do SOAP, que é uma API. Quanto às estratégias, com o foco na organização e seus processos, se trata da “Arquitetura Orientada a Serviço” [13], pois os consumidores (sistemas) da integração foram pensados para consumir funcionalidades por meio de serviços, configurando então uma Integração Orientada a Serviço.

Com base nestas decisões, chegou-se à arquitetura apresentada na Fig. 5.

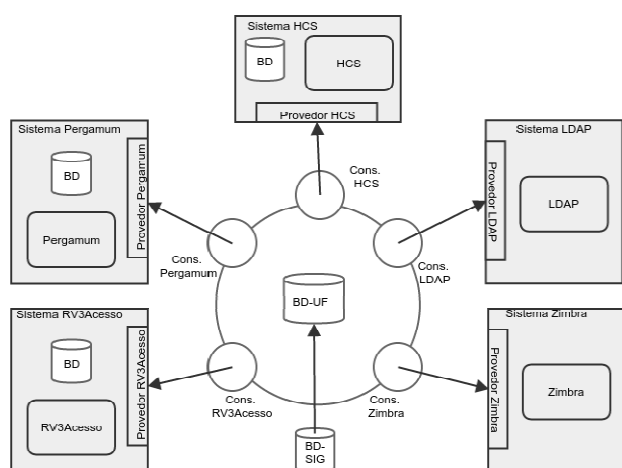


Fig. 5. Integração entre sistemas na Universidade na Fase 3

O funcionamento desta integração é similar às outras abordagens, pois o “disparo” inicial é dado por atualização no SIG ou em sistemas conectados ao BD-UF. A partir disso, o Cron executa os consumidores de serviço, que identificam atualizações para seus respectivos sistemas diretamente na tabela de *buffer* do BD-UF. Estes consumidores invocam os serviços de atualização, que atualizam seus sistemas.

Todas as atividades executadas pelos provedores de serviço bem como pelos consumidores são salvas em uma tabela de Log. Isto se faz necessário principalmente pela existência de uma rede de comunicação passível de falhas entre provedores e consumidores.

Na Fase 3, alguns aspectos da integração foram facilitados. Como exemplo, têm-se a atualização de versão do Zimbra, que passou a fornecer acesso às suas funções via *Web Services*. No entanto, alguns complicadores também foram inseridos, como a adição do BD-SIPAC, utilizando o SGBD PostgreSQL, diferente dos outros SGBDs da integração até este momento.

Com o desenvolvimento dos serviços e consumidores relacionados, chegou-se a um “esqueleto” básico, a partir do qual qualquer novo serviço ou consumidor pode ser desenvolvido. Este fato traz consigo uma alta escalabilidade, uma das características essenciais e desejadas para a integração na Universidade.

Além disso, a interoperabilidade que o uso do SOAP provê também auxiliou na integração com sistemas externos, ou seja, sistemas que utilizam os *Web Services* da Universidade para acesso à funcionalidades. Essa interoperabilidade é importante, pois permite a interconexão de sistemas de software desenvolvidos em diferentes linguagens, como por exemplo, um consumidor construído em Java pode se conectar aos serviços desenvolvidos em PHP.

Para organizar os provedores, foram criadas duas categorias: Provedores Públicos e Provedores Privados. Essa categorização permite um maior controle em relação à

segurança, pois: os Provedores Privados são utilizados somente pelos sistemas da integração; e os Provedores Públicos disponibilizam serviços para sistemas externos à integração, isolando os Provedores Privados. Este isolamento é interessante, pois os consumidores dos serviços públicos não têm conhecimento dos serviços privados existentes, e para a ocultação de complexidade das operações aos consumidores.

Na Fig. 6, pode-se perceber que o Provedor Privado é encapsulado pelo Provedor Público: as chamadas do consumidor público passam primeiro pelo Provedor Público, que por sua vez faz chamadas ao Provedor Privado.

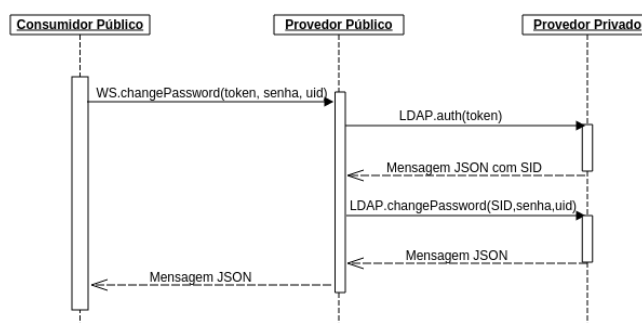


Fig. 6. Fluxo de chamadas entre Consumidor e Provedor Público

As chamadas entre os Consumidores e os Serviços Privados são realizadas em três passos (como apresentado na Fig. 6):

1. Consumidor chama o serviço de autenticação enviando seu *token*;
2. Serviço verifica se o *token* recebido existe e se o IP de origem está na lista dos IP’s liberados para acesso para o respectivo *token*.
3. Consumidor recebe uma mensagem formatada em JSON. O atributo *tipo* definirá a mensagem. Caso seja “SUCESSO”, o atributo *message* será uma *string*, que deverá ser o primeiro parâmetro nas chamadas subsequentes ao serviço em questão. Essa *string* é uma ID de Sessão (SID). Do contrário, o atributo *message* será uma mensagem explicativa relacionada ao problema de autenticação.

Na Fig. 7, é apresentado um fluxo de chamadas e retornos utilizando o Provedor do LDAP, exemplificando o uso dos serviços *auth* e *changePassword* para autenticação e troca de senha.

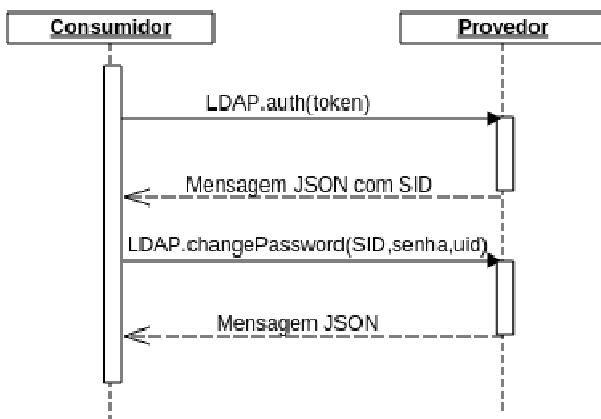


Fig. 7. Fluxo de chamadas aos Provedores

As chamadas aos Serviços Públicos são feitas diretamente às funcionalidades, como as chamada aos Serviços Privados, porém enviando o *token* no lugar do SID. Como parte deste trabalho, foi desenvolvido um sistema de gerência para que os mecanismos de integração pudessem ser monitorados. Este sistema de monitoramento inclui cadastro de Provedores, Serviços, *tokens*, vinculação de *tokens* a IP's e serviços, e um *dashboard* com informações sobre as execuções dos serviços nas últimas horas, comparação entre utilização de serviços nas últimas duas semanas e dados das últimas execuções de cada serviço.

Três dos tipos de gráficos presentes no *dashboard* são ilustrados na Fig. 8, na Fig. 9 e na Fig. 10. A Fig. 8 apresenta o gráfico do Sistema de Gerenciamento que exibe a quantidade de chamadas aos serviços de cada sistema nas últimas 24 horas. Por exemplo, entre 18 e 19 horas do dia anterior à visualização deste gráfico, o Provedor Pergamum teve por volta de 1300 chamadas a seus serviços realizadas pelo seu consumidor privado.



Fig. 8. Número de requisições atendidas por cada serviço por hora.

A Fig. 9 apresenta um gráfico ("tipo pizza") que permite comparar a quantidade de chamadas aos serviços de cada sistema. Por exemplo, o total de chamadas ao Provedor Pergamum representa 55.2% do total de chamadas, enquanto o total de acessos ao Provedor RV3Acesso representa 43.4%.

Esses dois sistemas utilizam informações de servidores e discentes da Universidade. Portanto, sempre que algum dado é inserido ou alterado nos sistemas concentradores de dados SIG-UFLA e Cin-Cadastro, ele deve ser propagado para os demais sistemas. Essa propagação é realizada por meio de requisições do Consumidor privado ao seu Provedor resultando nessa quantidade elevada de chamadas.

A Fig. 10 apresenta um gráfico que permite comparar a proporção de Falhas e Sucesso nas operações utilizando os *Web Services* de cada sistema. Por exemplo, o Provedor HCS teve 97.04% de sucesso e 2.86% de falhas na execução de seus serviços. O sistema HCS gerencia os dados de servidores que registram ponto eletrônico e esse percentual de falhas indica que o provedor encontrou inconsistência nos dados enviados pelo consumidor.

Para ilustrar a importância do *dashboard*, durante o desenvolvimento deste trabalho, a equipe que cuida da integração observou nos gráficos uma quantidade não esperada de chamadas a serviços, e então pôde investigar as causas desse comportamento e as corrigir de forma eficiente e eficaz.

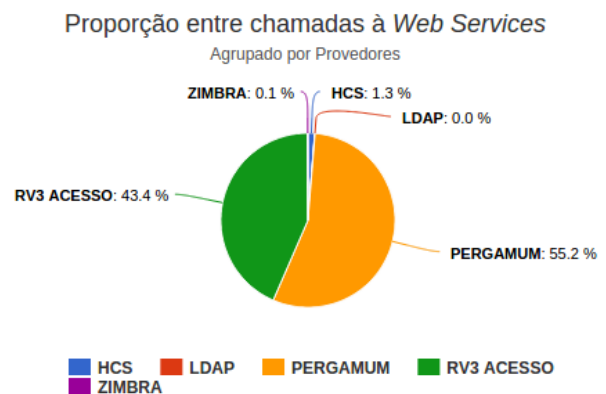


Fig. 9. Comparação global entre os Provedores da Integração

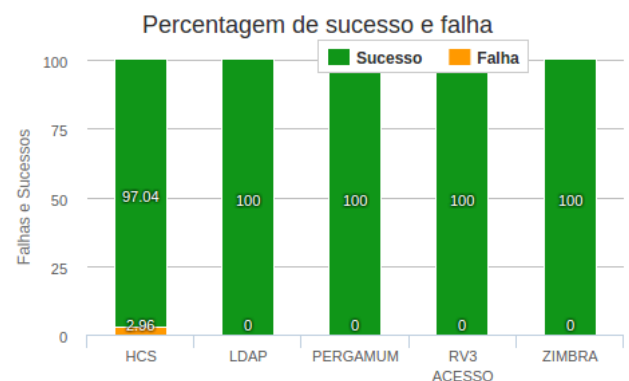


Fig. 10. Gráfico de comparação global entre os Provedores da Integração

C. Resumo comparativo

Na Tabela II, têm-se um resumo comparativo das abordagens utilizadas nas Fases estudadas, mostrando vantagens e desvantagens para cada uma delas. Por exemplo, a abordagem “Replicação de Dados (*Triggers, Functions e Procedures*)” tem como vantagem a sensação de instantaneidade, mas tem como desvantagem a dependência da tecnologia do Sistema Gerenciador de Banco de Dados (SGBD).

Todas as abordagens estudadas possuem vantagens e desvantagens e elas devem ser consideradas a partir do contexto da Universidade. Isso significa que dependendo das tecnologias, da quantidade de sistemas e serviços envolvidos, uma abordagem pode ser mais adequada que outra. No contexto da UFLA, a abordagem utilizando SOAP e *Web Services* é a mais adequada, pois permite uma maior escalabilidade e flexibilidade, visto a expansão de sistemas de software e serviços de rede integrados.

TABELA II
QUADRO-RESUMO DE ABORDAGENS

Abordagem	Vantagens	Desvantagens
Replicação de Dados (<i>Triggers, Functions e Procedures</i>)	Sensação de instantaneidade	Dependência da tecnologia do Sistema Gerenciador Banco de Dados
UDF	Sensação de instantaneidade e integração com tecnologias externas ao SGBD	Dependência da tecnologia do SGBD; Recuperação de falhas; Velocidade
Master/Slave	Sensação de instantaneidade	Dependência da tecnologia do SGBD e Rede, podendo gerar registros inconsistentes de um servidor para outro
CRON + Scripts PHP	Manutenibilidade	Atualizações não são instantâneas
<i>Web Services</i>	Escalabilidade; Isolamento de funções	Dependência de rede externa; dependência da tecnologia
SOAP	Serviços bem definidos	Requisições e retornos com formato padronizado, contendo muita informação

V. CONCLUSÃO

O ambiente de TI das organizações de médio e grande porte é complexo, devido à heterogeneidade e escala das plataformas e sistemas de informação. Para a consistência dos dados manipulados por estes sistemas, é ideal que eles estejam integrados, funcionando como um único sistema.

Em ambientes universitários, o cenário não é diferente: diversos sistemas são utilizados, com diferentes objetivos. Neste trabalho, foi apresentado um estudo de caso sobre a integração de sistemas e serviços no ambiente universitário, envolvendo 11 sistemas e 2 serviços. Nesse estudo, foi

possível identificar 4 Fases de integração, desde o momento que não havia integração, em meados de 2006, até o desenvolvimento de uma arquitetura de integração e sua implantação no ambiente, em Outubro de 2014.

Na Fase 3, foi projetada uma arquitetura orientada a serviço para a integração entre os sistemas na Universidade. Juntamente com esta arquitetura, foi desenvolvida uma interface de monitoramento e gerenciamento de serviços e permissões da integração.

O estudo histórico da integração na Universidade trouxe um entendimento maior sobre sua evolução, suas necessidades iniciais, vantagens e desvantagens de cada uma das soluções adotadas. Este entendimento pode evitar a utilização de abordagens que se provaram ineficientes no contexto da integração de sistemas de informação e serviços de rede na Universidade.

A integração de sistemas e serviços, especialmente em ambientes acadêmicos, é muito interessante, pois lida com sistemas administrativos (gestão de pessoas, compras, licitações, almoxarifado), sistemas acadêmicos (gestão de cursos e alunos, matrícula, geração de atestados, históricos, projetos de pesquisa) e serviços de rede (e-mail, autenticação centralizada, *wireless*) presentes no cotidiano universitário, além da integração com sistemas não institucionais.

Como trabalhos futuros, sugere-se:

- Estudo mais aprofundado sobre os aspectos de segurança em *Web Services*;
- Identificação de pontos que podem ser comuns à instituições de ensino, de forma a fornecer um padrão para integração neste tipo de ambiente;
- Proposta de adequação aos padrões de interoperabilidade e-Ping⁷ do Governo Federal;
- Proposta de integração com sistemas estruturantes do Governo Federal, como: SIAPE, SIGEPE, Comprasnet.

REFERÊNCIAS

- [1] Andersson, V. O.; dos Santos, R. T.; Tillmann, A. L. C. & Noguez, J. H. S. *COBALTO Webservice: Solução para consistência de informações*. Resumo Publicado na VIII Workshop de Tecnologia da Informação e Comunicação das IFES (2014).
- [2] Barros, F. *Mercado de software nacional vai crescer 400% em 10 anos*. Disponível em: <<http://convergenciadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?infoid=32006>>. Acessado em 10/09/2014, (2012).
- [3] Cerami, E. *Web services essentials: distributed applications with XML-RPC, SOAP, UDDI & WSDL*. O'Reilly Media, Inc., (2002)
- [4] Costa, C., Melo, A. C., Fernandes, A., Gomes, L. M. & Guerra, H. *Integração de Sistemas de Informação Universitários via Web Services*. Actas da 5ª Conferencia Ibérica de Sistemas y Tecnologías de Información, p. 290-295 (2010).

⁷ <http://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padroes-de-interoperabilidade>

- [5] Coulouris, G., Dollimore, J., Kindberg, T. & Blair, G. *Sistemas Distribuídos-: Conceitos e Projeto*. Bookman Editora. (2013)
- [6] De Mello Jorge, M. H. P., Laurenti, R. & Gotlieb, S. L. D. *Avaliação dos sistemas de informação em saúde no Brasil*. Cad. Saude Colet **18**, 07-18 (2010).
- [7] Degan, J. O. C. *Integração de dados corporativos: uma proposta de arquitetura baseada em serviços de dados* Unicamp - Universidade Estadual de Campinas, (2005)
- [8] DGTI. *Plano Diretor de Tecnologia da Informação 2011/2012* Universidade Federal de Lavras (2011).
- [9] Do Carmo, B. & Almeida, D. Uso de Sistemas de Informação geográfica na avaliação da Microbacia do Ribeirão das Alagoas, Conceição das Alagoas, Minas Gerais *Publicatio UEPG-Ciências Exatas e da Terra, Agrárias e Engenharias*, **19**, **9**. (2013).
- [10] Fielding, R. T. *Architectural styles and the design of network-based software architectures*. University of California, Irvine (2000).
- [11] Gerhardt, T. E.; Pinto, J. M.; Riquinho, D. L.; Roesse, A.; Santos, D. L. d. & Lima, M. C. R. d. *Utilização de serviços de saúde de atenção básica em municípios da metade sul do Rio Grande do Sul: análise baseada em sistemas de informação*. Ciência & Saúde Coletiva, SciELO Brasil, **16**, 1221-1232 (2011).
- [12] Hensle, B.; Booth, C.; Chappelle, D.; McDaniels, J.; Wilkins, M. & Bennett, S. *Oracle Reference Architecture - Service-Oriented Integration*, Release 3.0.Oracle, (2010)
- [13] Hohpe, G. & Woolf, B. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, (2003).
- [14] Intel. *Computadores ficaram 61% mais baratos nos últimos 10 anos*. <Disponível em: <http://newsroom.intel.com/docs/DOC-3946>>. Acessado em 26/08/2014, (2013)
- [15] Laurindo, F. J. B.; Shimizu, T.; Carvalho, M. M. d. & Rabechini Jr, R. *O papel da tecnologia da informação (TI) na estratégia das organizações*. Gestão & Produção, SciELO Brasil, **8**, 160-179. (2001)
- [16] Marin, H. d. F. *Sistemas de informação em saúde: considerações gerais*. Journal of Health Informatics, **2**, (2010)
- [17] Martins, V. M. M. *Integração de Sistemas de Informação: Perspectivas, normas e abordagens*. Universidade do Minho - Guimarães - Portugal. (2005)
- [18] Nascimento, A. M.; Luft, M. C. M. S.; Araujo, G. F. d. & Dacorso, A. L. R. *Implantação de Sistemas de Informação em uma secretaria estadual* Revista Pensamento Contemporâneo em Administração, **5**, 66-82 (2011)
- [19] OASIS, Organization for the advancement of structured information standards. *Service-Oriented Architecture*. Disponível em: [/docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf](http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf)>. Acessado em 15/09/2014, (2006).
- [20] Pressman, R. S. *Software engineering: a practitioner's approach*. McGraw-Hill, McGraw-Hill, (2001)
- [21] Ramos, Célia. M. Q. *Sistemas de informação para a gestão turística*. Revista Encontros Científicos-Tourism & Management Studies, Escola Superior de Gestão, Hotelaria e Turismo, 107-116 (2010).
- [22] Technologies, IOWA. *Web Services Definition*. <Disponível em [/www.w3.org/2001/03/WSWS-popa/paper13](http://www.w3.org/2001/03/WSWS-popa/paper13)>. Acessado em 21/10/2014., (2001)
- [23] Turban, E., Leidner, D., McLean, E. & Wetherbe, J. *Tecnologia da Informação para Gestão: Transformando os Negócios na Economia Digital* 6ed. Bookman, (2004)
- [24] UFLA. *Novos cursos de Engenharia da UFLA terão Área Básica de Ingresso*. Disponível em [/www.ufla.br/ascom/2014/05/30/novos-cursos-de-engenharia-da-ufla-terao-area-basica-de-ingresso/](http://www.ufla.br/ascom/2014/05/30/novos-cursos-de-engenharia-da-ufla-terao-area-basica-de-ingresso/)>.Acessado em 24/08/2014 (2013)
- [25] UFLA, ASCOM. *Números*. Disponível em: [/www.ufla.br/portal/institucional/sobre/numeros/](http://www.ufla.br/portal/institucional/sobre/numeros/)>. Acessado em 24/08/2014 (2014)
- [26] Valipour, M. H.; Amirzafari, B.; Maleki, K. N. & Daneshpour, N. A *Brief Survey of Software Architecture Concepts and Service. Oriented Architecture*. JASIS, (2009)
- [27] Van den Heuvel, W.-J.; Zimmermann, O.; Leymann, F.; Lago, P.; Schieferdecker, I.; Zdun, U. & Avgeriou, P. *Software service engineering: Tenets and challenges*. Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, 26-33 (2009)
- [28] Vidor, A. C.; Fisher, P. D. & Bordin, R. *Utilização dos sistemas de informação em saúde em municípios gaúchos de pequeno porte*. Revista Saúde Pública, SciELO Public Health, **45**, 24-30 (2011).
- [29] Zur Muehlen, M.; Nickerson, J. V. & Swenson, K. D. *Developing web services choreography standards — the case of REST vs. SOAP*. Decision Support Systems, Elsevier, **40**, 9-29 (2005)