



**CRISTIANO MESQUITA GARCIA**

**INTEGRAÇÃO DE SISTEMAS:  
ESTUDO DE CASO EM UM AMBIENTE  
UNIVERSITÁRIO**

**LAVRAS – MG**

**2014**

**CRISTIANO MESQUITA GARCIA**

**INTEGRAÇÃO DE SISTEMAS:  
ESTUDO DE CASO EM UM AMBIENTE UNIVERSITÁRIO**

Trabalho de Conclusão de Curso apresentado ao  
Departamento de Ciência da Computação para  
obtenção do título de bacharel em Sistemas de  
Informação

Orientador

MSc. Ramon Simões Abílio

Co-Orientador

Dr. Neumar Costa

Malheiros

**LAVRAS – MG**

**2014**

**CRISTIANO MESQUITA GARCIA**

**INTEGRAÇÃO DE SISTEMAS: ESTUDO DE  
CASO EM UM AMBIENTE UNIVERSITÁRIO**

Trabalho de Conclusão de Curso de  
Graduação apresentado ao Colegiado do  
Curso de Bacharelado em Sistemas de  
Informação, para obtenção do título de  
Bacharel.

APROVADA em 27 de novembro de 2014.

MSc. Erasmo Evangelista de Oliveira

Dr. Raphael Winckler de Bettio

  
MSc. Ramon Simões Abílio (Orientador)

Dr. Neumar Costa Malheiros (Coorientador)

**LAVRAS-MG  
Novembro/2014**

*Aos meus pais Rudimar e Luisa e minha irmã Marcela,  
Minha namorada Thaís e sua família,  
À toda minha família, em especial meus avós Nenzinha, Nelo, Diva e Lourival,  
Meus tios Flávio e Cláudia,  
Meus orientadores e meus amigos.*

## AGRADECIMENTOS

A Deus, pela minha vida e por me acompanhar durante toda a trajetória, mostrando que tudo tem um porquê; Aos meus pais Luisa e Rudimar, e minha irmã Marcela, pelo amor e apoio incondicional. Amo vocês! À minha namorada Thaís, pela paciência, amor e cuidado, e à sua família, em especial seus pais Sérgio e Cláudia, e seu irmão Alexandre, pessoas a quem tenho como parte de minha própria família; À toda a minha família, em especial Tia Cláudia, Tio Flávio, Vó Nenzinha, Vó Diva; e à pessoas que sempre estarão vivas no meu coração: Vô Nelo, Vô Lôro, e Tio Carlão; A todos os meus amigos e colegas da turma 2010/1 de Sistemas de Informação / UFLA, em especial Ariel, Danilo e Hauberty, e aos meus amigos que conheci na Hungria, em especial os “Produceiros”, pelo companheirismo e amizade. À Universidade Federal de Lavras e à Budapest Business School, pela estrutura e ensino oferecido durante a graduação; e à CAPES, pela oportunidade do período sanduíche na Hungria; Aos professores que tive, em especial aos que tive contato um pouco mais de perto: André Grützmann, Thomaz Chaves, Denilson Pereira, André Pimenta; Aos meus orientadores do TCC: Ramon Abílio e Neumar, pela paciência e por “abraçar” este trabalho tanto quanto eu; Por último mas não menos importante, aos meus colegas da DGTI, pelo apoio e amizade, tornando os dias de trabalho muito mais prazerosos.

A todos, meu muito obrigado!

*“Whether you think that you can or that you can’t, you’re right”*

*(Henry Ford)*

## RESUMO

O uso de Sistemas de Informação (SI) tem crescido expressivamente. O poder de processamento dos computadores vem aumentando cada vez mais, possibilitando sua utilização para tarefas mais complexas. Com isso, instituições públicas e privadas têm investido em SI para otimização de processos e como ferramenta estratégica. Nestas instituições, é comum encontrar um cenário com diversos sistemas de informação, no qual a integração entre os sistemas para compartilhamento dos dados e informações é essencial e, ao mesmo tempo, desafiador. A Universidade Federal de Lavras (UFLA) possuía, até 2011, 24 sistemas de informação, com diferentes características. Além destes, em 2013 uma família de *softwares* passou a ser implantada, aumentando ainda mais a complexidade da integração de sistemas na Universidade. Neste trabalho, por meio de um estudo de caso, foi apresentada e analisada a evolução da integração de sistemas de informação no contexto da UFLA, contribuindo para outras instituições que estejam em situação semelhante.

**Palavras-Chave:** Integração de Sistemas; Sistemas de Informação; *Web Services*; SOA.

## ABSTRACT

The use of Information Systems (IS) have been growing remarkably. The computers' processing is getting more powerful continuously, bringing the possibility of using these computers for more complex tasks. Having this, private and public institutions have been investing in IS for processes optimization and strategic tool. In these institutions, it is common to find a scenario having many information systems, in which the integration for sharing data and information is essential, and simultaneously, challenging. The Federal University of Lavras (UFLA) had, till 2011, 24 information systems, with different features. In addition, in 2013 a software family started being deployed, increasing even more the systems integration's complexity in the University. In this work, through a case study, it is presented and analyzed the evolution of the systems integration in the UFLA's context, contributing to other institutions that are in similar situation.

***Keywords:*** *Systems Integration; Information Systems; Web Services; SOA.*

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>11</b>
1.1	Contexto . . . . .	11
1.2	Objetivos . . . . .	14
1.3	Estrutura do Trabalho . . . . .	15
<b>2</b>	<b>Referencial Teórico</b>	<b>16</b>
2.1	Sistemas de Informação Integrados . . . . .	16
2.2	Abordagens de Integração de Sistemas . . . . .	18
2.3	Arquitetura Orientada a Serviço . . . . .	20
<b>3</b>	<b>Metodologia</b>	<b>25</b>
3.1	Classificação . . . . .	25
3.2	Procedimento Metodológico . . . . .	26
<b>4</b>	<b>Resultados e Discussão</b>	<b>30</b>
4.1	Sistemas de <i>Software</i> e Serviços na Universidade . . . . .	30
4.2	Estudo de Caso . . . . .	33
4.2.1	Pré-Integração . . . . .	34
4.2.2	Momento 1 . . . . .	34
4.2.3	Momento 2 . . . . .	37
4.2.4	Momento 3 . . . . .	41
4.3	Discussão . . . . .	50
<b>5</b>	<b>Trabalhos Relacionados</b>	<b>54</b>
<b>6</b>	<b>Conclusão</b>	<b>57</b>

## LISTA DE FIGURAS

2.1	Exemplo de integração vertical e horizontal entre sistemas. . . . .	17
2.2	Representação conceitual de uma mensagem utilizando SOAP. . . . .	22
2.3	Exemplo de mensagem utilizando SOAP. . . . .	23
2.4	Exemplo de chamada a um serviço em arquitetura REST. . . . .	24
3.1	Classificação de pesquisa: por (JUNG, 2004) . . . . .	25
4.1	Exemplo da interação e dados compartilhados entre sistemas da Integração. . . . .	33
4.2	Integração inicial entre os sistemas da UFLA. . . . .	35
4.3	Integração entre sistemas na UFLA no Momento 2. . . . .	39
4.4	Integração entre sistemas na UFLA no Momento 3. . . . .	43
4.5	Fluxo de chamadas no acesso aos <i>Web Services</i> Privados partindo de um consumidor privado. . . . .	45
4.6	Fluxo de chamadas no acesso aos <i>Web Services</i> Públicos partindo de um consumidor público / externo. . . . .	45
4.7	Fluxo de chamadas utilizando o <i>Web Service</i> . . . . .	46
4.8	<i>Dashboard</i> de monitoramento no sistema gerencial da Integração. . .	48
4.9	Gráfico de execuções de serviços por hora. . . . .	49
4.10	Gráfico de comparação global entre os <i>Web Services</i> da Integração. . .	50
4.11	Gráfico de comparação de percentagem de Sucesso e Falha entre os <i>Web Services</i> da Integração. . . . .	50

## **LISTA DE TABELAS**

4.1	Sistemas mencionados e suas respectivas tecnologias. . . . .	32
4.2	Servidores e seus respectivos Sistemas Operacionais. . . . .	36

# 1 INTRODUÇÃO

## 1.1 Contexto

O uso de Sistemas de Informação (SI) tem crescido exponencialmente (BARROS, 2012). Desde 1950, com a produção do primeiro circuito integrado, computadores passaram a fazer cada vez mais parte do cotidiano, devido à redução de preço que a troca de válvulas para transístores possibilitou (FERREIRA, 1998). O poder de processamento destes computadores também vem aumentando à medida que diminuem de tamanho e preço. Com isso, instituições públicas e privadas, nas mais diversas áreas, das maiores até as mais simples, possuem computadores com um sistema de informação para auxiliar na gerência do negócio.

Em ambientes corporativos, é comum encontrar um cenário com tipos diferentes de sistemas, tanto nos níveis operacional e gerencial quanto no nível estratégico (TURBAN *et al.*, 2004). Na prática, uma série de funções são proporcionadas no formato de um sistema de informação integrado (O'BRIEN, 2010). Com isso, os dados são compartilhados entre sistemas mesmo que não estejam no mesmo nível administrativo, consolidando assim a integração vertical (entre sistemas de diferentes níveis) e a horizontal (entre sistemas de mesmo nível) destes sistemas.

Existem diversas abordagens para integração de sistemas. Diferentes taxonomias podem ser encontradas na literatura, utilizando diferentes focos. Por exemplo, em (MARTINS, 2005), em que as abordagens são classificadas de acordo com uma visão centrada em técnicas, têm-se abordagens como “Informação Centralizada”. Em (HOHPE; WOOLF, 2003), no qual o foco é direcionado à organização e seus processos, têm-se, por exemplo, uma abordagem chamada *Business-to-Business (B2B)*.

Uma das formas de implementação de integração entre sistemas é pela utilização da Arquitetura Orientada a Serviço (SOA). SOA é um paradigma que

objetiva organizar e utilizar recursos que podem estar sob responsabilidade de diferentes proprietários, utilizando um meio uniforme para a utilização, oferta, descoberta e interação com estes recursos em forma de funcionalidades de serviços (OASIS, 2006). A utilização da *SOA* com o propósito específico da integração entre múltiplos sistemas é chamada de *Service-Oriented Integration (SOI)* (HENSLE *et al.*, 2010).

*Web Services* constituem um meio de implementação da *SOI* e representam uma visão sobre a programação distribuída e a disponibilização de recursos, fortemente ligada à Internet (TECHNOLOGIES, 2001). Diversas tecnologias, como *Simple Object Access Protocol (SOAP)*, arquitetura *Representational State Transfer (REST)* e *Java API for Web Services (JAX-WS)* fornecem suporte para a implementação de *Web Services* ((CERAMI, 2002), (ORACLE, 2011)).

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 (COSTA *et al.*, 2010). 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 (ANDERSSON *et al.*, 2014).

A Universidade Federal de Lavras (UFLA) vem tendo um grande crescimento em sua comunidade. Até junho de 2013, em torno de 16581 alunos estavam matriculados na Universidade, entre cursos de graduação, pós-graduação e especialização, e por volta 1635 servidores, entre docentes e técnicos, ocupavam cargos na Instituição (UFLA, 2014). Esse número tende a crescer com a criação de novos cursos (UFLA, 2013).

A UFLA possui uma Diretoria de Gestão de Tecnologia da Informação (DGTI<sup>1</sup>), e, segundo o Plano Diretor de Tecnologia da Informação 2011/2012 (DGTI, 2011), até 2011, a Universidade possuía 24 sistemas de informação mantidos pela DGTI. Cada um destes sistemas possui suas singularidades quanto à opção por banco de dados, plataformas e tecnologias utilizadas no desenvolvimento, além de terem sido produzidos por diferentes empresas. Em 2013, uma “família” de sistemas de informação institucionais integrados começou a ser implantada na Universidade. A implantação destes sistemas tem sido gradual e adiciona ainda mais complexidade no contexto da integração de sistemas de informação.

Além dos sistemas de *softwares* mencionados, a Universidade ainda oferece serviços como email e *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, além de necessária, um desafio.

Até meados de 2006, o compartilhamento de dados entre sistemas era executado manualmente, como por exemplo, dados eram compartilhados entre o sistema acadêmico e o sistema da biblioteca utilizando planilhas eletrônicas. A criação de contas de email era sob demanda, e algumas vezes faltavam informações do usuário do email. Então, foi notada a necessidade de uma integração entre os sistemas para o compartilhamento e consistência de dados. Nesse ponto, foi criado um banco de dados e um sistema *Web* para gerir tais dados.

A integração inicial era a nível de dados, pelo compartilhamento de banco de dados. Serviços e sistemas que não utilizavam a mesma tecnologia de banco de dados foram integrados utilizando *UDF*<sup>2</sup>. Tais *UDF* eram de difícil manutenção,

---

<sup>1</sup><http://www.dgti.ufla.br/>

<sup>2</sup>*User Defined Function* - <http://dev.mysql.com/doc/refman/5.1/en/adding-functions.html> - *UDF*: funções definidas pelo usuário. São funções utilizadas como funções nativas do MySQL.

visto que eram desenvolvidas em C e precisavam ser compiladas dentro do SGBD. O sistema da biblioteca ainda era atualizado manualmente até este ponto.

Numa tentativa de abandonar a utilização de *UDF*, foi iniciada a transição de algumas integrações via *UDF* para o Crontab - agendador de tarefas de sistemas operacionais baseados em Unix -, por exemplo configurando a execução de um *script* de atualização para cada 10 minutos. Os sistemas mais críticos da integração ainda utilizavam *UDF*, porém a utilização do Crontab atendeu as necessidades nas integrações em que foi utilizado.

A escalabilidade e monitorabilidade da integração estavam comprometidas pela falta de padrão entre os *scripts* que a efetivavam. A integração foi repensada para suprir estes pontos e prezar também pela segurança, e a Arquitetura Orientada a Serviço se mostrou como uma opção. Os serviços e consumidores para a efetivação da integração foram então desenvolvidos como conectores entre os sistemas. Em conjunto, foi desenvolvida uma interface de monitoramento e gestão da segurança (utilização de *tokens* e suas permissões).

Este trabalho relata os resultados de um estudo de caso realizado na UFLA e contribui para a melhoria das integrações que já acontecem no ambiente universitário, ressaltando vantagens e desvantagens das abordagens utilizadas e apresentando sugestões de melhoria. Esse relato pode servir de referência para universidades na mesma situação da UFLA no âmbito de Tecnologia da Informação e integração entre seus sistemas.

## 1.2 Objetivos

O objetivo geral deste trabalho é realizar um estudo de caso analisando as soluções de integração entre sistemas de informação na UFLA, destacando os benefícios e as dificuldades encontradas. Para atingir o objetivo geral, foram definidos os seguintes objetivos específicos:

- Realizar revisão bibliográfica sobre sistemas de informação integrados, abordagens para integração de sistemas e Arquitetura Orientada a Serviço;
- Estudar os sistemas de *software* e serviços da UFLA que estão integrados, destacando tecnologias, objetivos do sistema de *software* e dependências entre eles;
- Realizar um estudo de caso com foco nas abordagens utilizadas para integração entre os sistemas de informação da Universidade.

### **1.3 Estrutura do Trabalho**

O trabalho está organizado como a seguir. O Capítulo 2 contempla o Referencial Teórico, que aborda sistemas de informação integrados, abordagens de integração e Arquitetura Orientada a Serviço; no Capítulo 3, a metodologia do trabalho é definida, apresentando a classificação e os procedimentos metodológicos; no Capítulo 4, são apresentados os resultados do estudo sobre os sistemas e serviços da Universidade, bem como o estudo de caso realizado e discussão sobre tais resultados; no Capítulo 5, são apresentados Trabalhos Relacionados, e por fim, no Capítulo 7, é apresentada a Conclusão do trabalho.

## 2 REFERENCIAL TEÓRICO

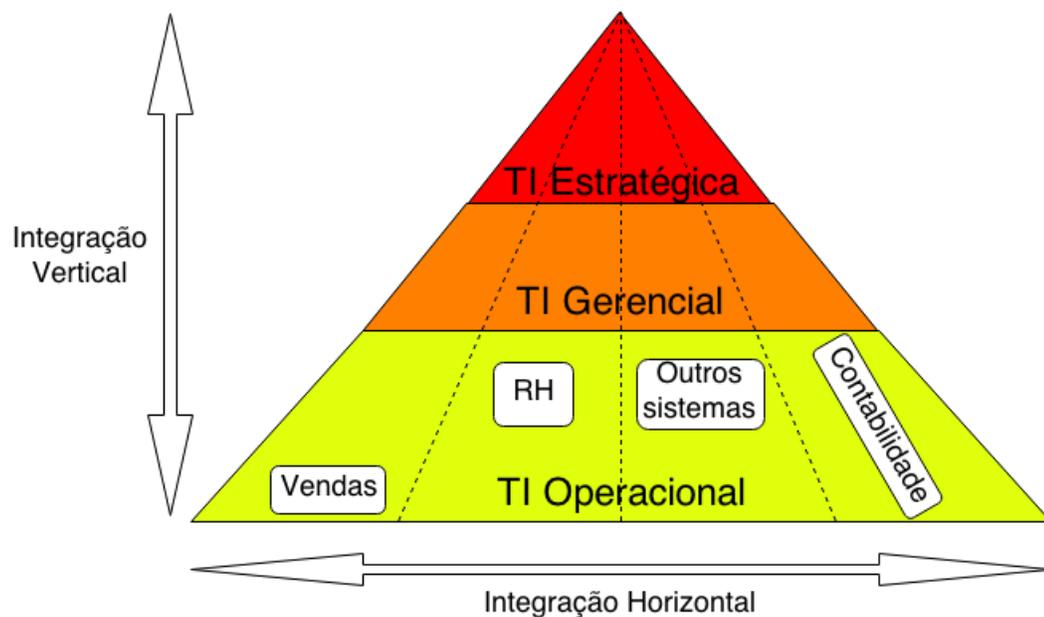
### 2.1 Sistemas de Informação Integrados

Segundo (TURBAN *et al.*, 2004), “Sistemas de Informação são sistemas que coletam, processam, armazenam, analisam e disseminam informação para um propósito específico”. Sistemas de informação podem ser compostos por *hardware*, *software*, processos e pessoas, podendo ainda conter outros subsistemas de informação (TURBAN *et al.*, 2004). Segundo (JESSUP; VALACICH, 2002), “Sistemas de Informação é o campo de estudo de redes complementares de *hardware* e *software* utilizado por pessoas e organizações para coletar, filtrar, processar, criar e distribuir dados.”

O uso de Sistemas de Informação (SI) vem crescendo expressivamente e tende a crescer ainda mais nos próximos anos (BARROS, 2012). O poder de processamento dos computadores também vem aumentando à medida que diminuem de tamanho e preço (INTEL, 2013). Com isso, instituições públicas e privadas nas mais diversas áreas, das maiores até as mais simples já possuem computadores, com um sistema de informação para auxiliar na gerência do negócio e automação de tarefas rotineiras (NASCIMENTO *et al.*, 2011). Por exemplo, SI tem sido utilizado em áreas como turismo (RAMOS, 2010), agrárias (CARMO; ALMEIDA, 2013) e saúde/hospitais ((JORGE; LAURENTI; GOTLIEB, 2010), (MARIN, 2010), (VIDOR; FISHER; BORDIN, 2011), (GERHARDT *et al.*, 2011)).

A utilização de computadores e sistemas de informação é mais do que otimização de tarefas de nível operacional. A Tecnologia da Informação (TI), onde os Sistemas de Informação estão inseridos, passou a ser uma ferramenta estratégica. 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 (LAURINDO *et al.*, 2001).

No mundo real, é comum encontrar empresas utilizando diversos tipos de sistemas, tanto nos níveis operacional, gerencial ou estratégico, em forma de combinações integradas (TURBAN *et al.*, 2004). Tais sistemas dão suporte a processos de negócios, e representam a “chave do sucesso” no sentido de obtenção de vantagem competitiva (O’BRIEN, 2010). Em um único nível, é possível a existência de vários sistemas diferentes para execução e controle de atividades específicas. Então, para se ter consistência nos dados, é preciso ter uma integração de dados tanto vertical (entre sistemas dos níveis operacional, gerencial e estratégico) quanto horizontal (entre sistemas de um único nível), como pode ser visto na Figura 2.1 (O’BRIEN, 2010).



**Figura 2.1:** Exemplo de integração vertical e horizontal entre sistemas.

Na Figura 2.1, tem-se a representação de níveis gerenciais, com exemplos de sistemas transacionais no nível operacional. A intenção é simbolizar a existência de integração entre sistemas de diferentes níveis e diferentes objetivos.

## 2.2 Abordagens de Integração de Sistemas

Existem razões que podem levar empresas à contratar ou desenvolver soluções de integração de sistemas. Tais razões podem ser, por exemplo (DEGAN, 2005):

- Estender à partir da tecnologia já existente, flexibilizando e reduzindo custos na implementação de novos serviços;
- Permitir integração com *stakeholders*, expandindo assim o alcance de serviços;
- Compartilhar informações comuns a 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 suas necessidades de integração (MARTINS, 2005). Uma organização pode realizar a integração de sistemas considerando abordagens com diferentes focos, como implementação, ou organização e seus processos.

Quanto ao nível de implementação, a integração de sistemas pode ser dividida da seguinte forma (MARTINS, 2005):

- **Aplicações Compostas:** aplicações integradas com o uso de *API*<sup>1</sup>. 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 Distribuídos:** conjunto de sistemas independentes que se apresentam de uma forma coerente, como se fosse um único sistema

---

<sup>1</sup>*Application Programming Interface*, ou Interface de programação de aplicação

(STEEN; TANENBAUM, 2001). No contexto de integração de sistemas, são sistemas que comunicam utilizando serviços, disponibilizando e/ou consumindo parte de sua lógica e recursos;

- **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, (HOHPE; WOOLF, 2003) define seis abordagens:

- **Information Portals (Portais de Informação)**, na qual a integração se dá ao nível de interface, na camada de apresentação;
- **Data Replication (Replicação de Dados)**, quando a integração acontece no nível de informação, onde as bases são distribuídas e mantidas atualizadas e sincronizadas;
- **Shared Business Functions (Funções de Negócios Compartilhadas)**, na qual módulos são compartilhados por vários sistemas;
- **Distributed Business Processes (Processos de Negócios Distribuídos)**, são semelhantes à “Funções de negócio compartilhadas”, porém a lógica do processo é armazenada em uma camada à parte, que pode acessar os diversos sistemas a serem integrados;
- **Business-to-Business (B2B) Integration (Integração Negócio-Negócio)**, vai além dos limites empresariais. Representa a disponibilização de funções entre diferentes organizações. Embora os demais conceitos citados em (HOHPE; WOOLF, 2003) também possam ser aplicados à *B2B*, o autor alerta que o uso de redes externas pode levantar novas questões a serem analisadas;

- ***Service-Oriented Architecture (Arquitetura Orientada a Serviço)***, onde 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 em (MARTINS, 2005) e (HOHPE; WOOLF, 2003) possuem semelhanças entre si. Por exemplo, as abordagens “Funções de Negócios Compartilhadas” e “Aplicações Compostas” podem utilizar de *API* para compartilhamento de informações e, assim, efetivar uma possível integração entre sistemas.

### 2.3 Arquitetura Orientada a Serviço

Arquitetura de um *software* representa a estrutura que abrange os componentes do *software*, suas propriedades externamente visíveis e as relações entre ambos. A utilização da Arquitetura de *Software* possui benefícios como facilitação da análise da efetividade do projeto em relação aos requisitos; consideração de alternativas arquiteturais em estágios iniciais do projeto e a diminuição dos riscos inerentes ao *software* (PRESSMAN, 2001).

Arquitetura Orientada a Serviço (*Service-Oriented Architecture (SOA)*) vem sendo considerada um dos principais paradigmas emergentes, inclusive dando origem à uma ramificação da Engenharia de *Software* original chamada Engenharia de *Software* de Serviço (HEUVEL *et al.*, 2009).

A *SOA* é, segundo (OASIS, 2006), “um paradigma para organizar e utilizar recursos ou funcionalidades que podem estar sob controle de diferentes proprietários. Esta arquitetura provê um meio uniforme para oferecer, descobrir, interagir e usar funcionalidades para produzir efeitos desejados e consistentes com condições e expectativas mensuráveis”. A *SOA* é capaz de oferecer uma série de benefícios, como a facilitação de crescimento controlado de sistemas, o provimento e utilização de serviços em escala de Internet e redução de custos em cooperações organização-organização (VALIPOUR *et al.*, 2009).

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 (HENSLE *et al.*, 2010). Esta técnica expõe dados, funcionalidades e processos para serem consumidos pelos sistemas participantes da integração. Foram apresentadas cinco abordagens para a *SOI* com foco na integração entre sistemas já existentes (HENSLE *et al.*, 2010):

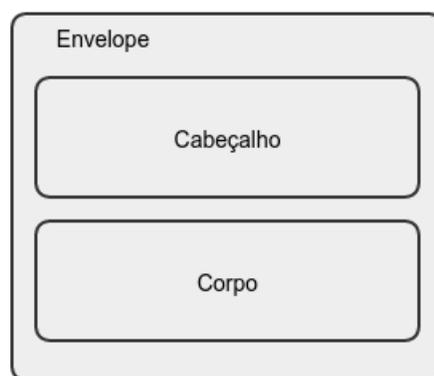
- ***Service (Serviços)***, que utiliza uma camada de serviços entre os sistemas existentes e os consumidores dos serviços;
- ***Process Integration (Integração de Processos)***, para a integração entre processos em um ambiente empresarial, sugerindo a utilização na integração de pequenos processos dentro de grandes processos, com interação humana ou não;
- ***Functional Integration (Integração Funcional)***, que busca englobar fluxos de trabalho executados em diferentes sistemas em uma única funcionalidade (encapsulamento);
- ***Data Integration (Integração de Dados)***, que se refere à uma abordagem que gerencie a complexidade de modelos de dados em diferentes aplicações;
- ***Bottom-Up Service Enablement***, em que os consumidores seriam capazes de identificar e buscar serviços que atendessem suas necessidades. Para efeito de comparação, a abordagem *Top-Down* acontece quando os requisitos dos consumidores são elicitados. Na abordagem *Bottom-Up*, os consumidores possuiriam a habilidade de buscar serviços.

*Web Services* representam um meio de implementar a *SOI*, fornecendo uma interface de serviço que permite que consumidores interajam com provedores de serviço (COULOURIS *et al.*, 2013). 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 interorganizacionais (MUEHLEN; NICKERSON; SWENSON, 2005).

*SOAP* é um protocolo que utiliza *WebServices Description Language (WSDL)*, documento baseado em *eXtensible Markup Language (XML)*, para descrever funcionalidades oferecidas por um *Webservice*. O *SOAP* provê um padrão básico de comunicação, onde cada operação é representada por seu terminal, descrito no documento *XML* enviado na requisição, ao invés de um tipo de mensagem (ou método *HyperText Transfer Protocol (HTTP)*), como utilizado no *REST* (MUEHLEN; NICKERSON; SWENSON, 2005).

Na Figura 2.2, é apresentado uma representação conceitual de uma mensagem SOAP, contendo um Cabeçalho e um Corpo, dentro de um Envelope. No Cabeçalho - que é opcional -, podem ser contidas informações que não estão ligadas diretamente à mensagem, como por exemplo o tempo limite para processamento da requisição. No Corpo, devem estar informações relacionadas à troca de mensagem.



**Figura 2.2:** Representação conceitual de uma mensagem utilizando SOAP.

Na Figura 2.3, é mostrado um exemplo de mensagem utilizando SOAP, com Cabeçalho enviando uma chave *maxTime* e um valor para esta chave. Essa mensagem é uma chamada para a operação *changePassword*, enviando os parâmetros *id*, com o valor *crisiano*, e *password*, com o valor *\$6\$SSrFpIerjrt*.

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

  <env:Header>
    <ufla:maxTime value="10000" xmlns:ufla="http://ufla.br"/>
  </env:Header>

  <env:Body>
    <ufla:operation name="changePassword"
      xmlns:ufla="http://ufla.br/ws" />

    <ufla:param name="id" value="crisiano"
      xmlns:ufla="http://ufla.br/params" />

    <ufla:param name="password" value="$6$SSrFpIerjrt"
      xmlns:ufla="http://ufla.br/params" />
  </env:Body>

</env:Envelope>
```

**Figura 2.3:** Exemplo de mensagem utilizando SOAP.

*REST* é uma abstração dos princípios que fazem a *World Wide Web* (WWW) escalável. Permite fornecer serviços identificados pela *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 (FIELDING, 2000). *URI* é um identificador uniforme de recurso, e o tipo mais conhecido de *URI* é o *Uniform Resource Location (URL)* (MUEHLEN; NICKERSON; SWENSON, 2005).

Com a evolução do *HTTP*, outros métodos puderam ser utilizados, como por exemplo *GetPropertyies* e *GetInstance* (MUEHLEN; NICKERSON; SWENSON, 2005). Por exemplo, *http://www.ufla.br/* é um *URI*. Comparando

com SOAP, na Figura 2.4 têm-se uma chamada a um serviço em arquitetura REST para a troca de senha, mesma atividade que é utilizada acima com SOAP. Esta chamada usa *PUT* para atualização de um registro de usuário, enviando  *cristiano* como valor do parâmetro *id*, e *\$6\$SSrFpIerjrt* como valor do parâmetro *password*. Essa chamada seria:

```
PUT http://www.ufla.br/ws/usuario/id/cristiano/password/$6$SSrFpIerjrt
```

**Figura 2.4:** Exemplo de chamada a um serviço em arquitetura REST.

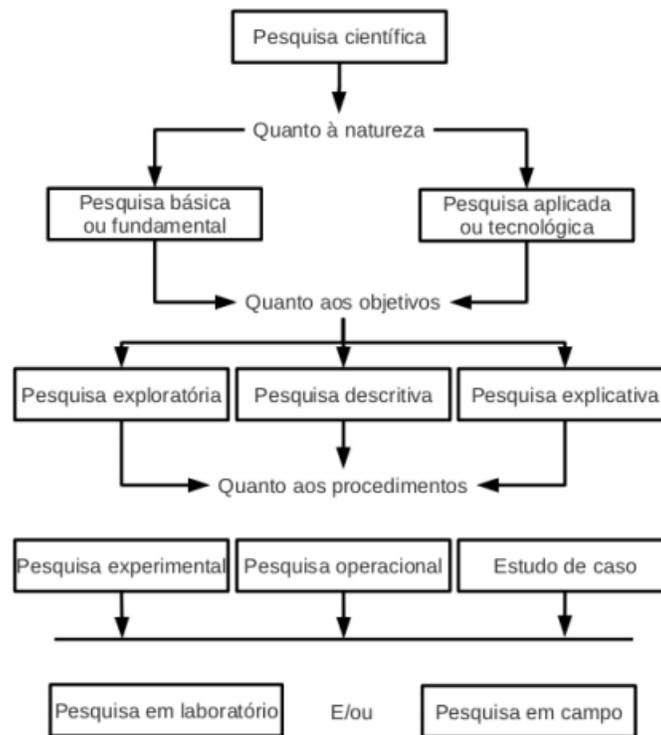
Como REST não se trata de um protocolo, como o SOAP, mas sim de uma arquitetura (FIELDING, 2000), o retorno das chamadas pode ser formatado como o desenvolvedor ou empresa desejar, utilizando *Javascript Object Notation (JSON)* ao invés de XML, por exemplo.

### 3 METODOLOGIA

A classificação do trabalho e os procedimentos metodológicos a serem utilizados estão descritos de forma detalhada neste Capítulo.

#### 3.1 Classificação

Segundo (JUNG, 2004), uma pesquisa científica pode ser classificada em três diferentes formas: em relação à natureza, ao objetivo e aos procedimentos. O diagrama da Figura 3.1 mostra como é feita esta classificação.



**Figura 3.1:** Classificação de pesquisa: por (JUNG, 2004)

Pela classificação de (JUNG, 2004), de acordo com a **Natureza**, esse trabalho tem caráter de **Pesquisa Aplicada e/ou Tecnológica**, pois o conhecimento

adquirido será utilizado para propor melhorias à integração de sistemas utilizados atualmente na Universidade Federal de Lavras (UFLA), além de oferecer um exemplo prático para outras Universidades que estejam passando por situações semelhantes em relação ao assunto do trabalho. Quanto aos **Objetivos**, o trabalho corrente se enquadra em **Pesquisa Descritiva**, pois objetiva observar, registrar e analisar fenômenos, sem interferência do pesquisador, para estabelecer relações entre variáveis. Quanto aos **Procedimentos**, o atual trabalho se trata de um **Estudo de Caso**, pois foram estudadas as abordagens de integração entre os sistemas de informação no ambiente universitário, a fim de entender a evolução desta integração e propor melhorias. Este trabalho foi realizado no ambiente da Universidade nos meses de Agosto a Outubro/2014.

Segundo (MELCHIORI, 2011), este trabalho se enquadra em **Pesquisa de Campo**, pois foi feita uma observação do ambiente universitário da UFLA com uma fundamentação teórica advinda de uma **Pesquisa Bibliográfica**, a fim de se estudar a evolução da integração de sistemas na UFLA, comparando algumas de suas fases, e posteriormente sugerir melhorias.

### 3.2 Procedimento Metodológico

Para manter a infraestrutura de redes de computadores, *hardware* e *software*, a Universidade Federal de Lavras (UFLA) possui a Diretoria de Gestão de Tecnologia da Informação (DGTI). A missão da DGTI é “fornecer serviços e produtos de software ou hardware com qualidade e efetividade, suportando as atividades de ensino, pesquisa e extensão da Universidade Federal de Lavras, no âmbito de Tecnologia da Informação e Comunicações”. Ela tem como visão “ser excelência na prestação de serviços de TI e aumentar o nível de maturidade de governança de TI da UFLA, alinhando a Tecnologia da Informação aos objetivos de negócio (ensino, pesquisa e extensão) das unidades organizacionais (Pró-

Reitorias, diretorias, setores administrativos e departamentos didático-científicos)” (DGTI, 2011).

A DGTI está subdividida em cinco coordenadorias. Uma delas é a Coordenadoria de Sistemas de Informação – CSI. A CSI é 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: 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 (DGTI, 2011).

O trabalho foi realizado na UFLA, mais especificamente na CSI, nos meses de Agosto a Outubro de 2014. Esta coordenadoria possui, como um dos objetivos estratégicos, a integração de sistemas da UFLA, e conta ainda com os responsáveis diretos pelas primeiras tentativas de integração no ambiente universitário.

O trabalho foi desenvolvido nos seguintes passos, cronologicamente e, por algumas vezes, simultaneamente:

**Revisão** bibliográfica no contexto de Sistemas de Informação integrados e Abordagens para integração de sistemas. Foram levantadas as principais abordagens de integração, úteis no mapeamento das abordagens das integrações realizadas no ambiente universitário;

**Estudo** sobre os sistemas de *software* da UFLA, destacando as tecnologias empregadas, os objetivos dos sistemas de *software* e a dependência entre eles;

**Estudo de Caso** realizado utilizando entrevistas com os responsáveis pela integração entre os sistemas na UFLA. Através das respostas do questionário e da pesquisa documental, foram levantadas as abordagens de integração

utilizadas nas integrações, separadas em 3 momentos distintos, a motivação para o uso e a discussão sobre vantagens e desvantagens de cada uma das abordagens;

A entrevista com os responsáveis pela integração de sistemas na Universidade possuía o objetivo de identificar a motivação para o uso de tais abordagens e as expectativas com o uso das mesmas. Onze perguntas foram desenvolvidas para a entrevista:

- “Quais foram as primeiras necessidades de integração entre os sistemas na UFLA?”
- “Quais sistemas foram os primeiros a serem integrados? Quais informações eram necessárias de serem integradas?”
- “Quais foram as abordagens utilizadas? Quais as razões de terem escolhido tais abordagens?”
- “Quais foram as expectativas com a integração entre os sistemas?”
- “Quais as vantagens e desvantagens encontradas nessas abordagens?”
- “As expectativas foram alcançadas?”
- “Houve alteração (adição / mudança) em relação aos sistemas que estavam integrados?”
- “Quais são os sistemas e serviços integrados atualmente (2014)?”
- “Por que a mudança na estratégia de integração? Qual a motivação para isso?”
- “Quais as expectativas com a nova abordagem?”
- “Quais as vantagens e desvantagens encontradas nas abordagens atuais?”

As perguntas foram direcionadas a dois funcionários da DGTI, responsáveis pela integração inicial, sendo que um deles ainda se mantém com as

atividades relacionadas à integração. As respostas foram agrupadas, para que fosse possível estabelecer a ordem cronológica dos acontecimentos e para que as informações obtidas pudessem ser confrontadas. Os funcionários foram contatados posteriormente para esclarecimentos sobre alguns aspectos da integração.

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

## 4 RESULTADOS E DISCUSSÃO

Neste Capítulo, são apresentados os resultados e uma análise das características da integração entre os sistemas na UFLA e como tal integração é realizada hoje, abordando vantagens, desvantagens, características e tecnologias empregadas. Ao final, são discutidas possíveis melhorias para a integração corrente.

### 4.1 Sistemas de *Software* e Serviços na Universidade

A Universidade possuía, até 2011, 24 sistemas. No entanto, nem todos os sistemas estavam integrados. A razão para isso era, principalmente, que a utilização de alguns sistemas era interna a departamentos. Portanto, deu-se ênfase nos sistemas que foram ou estão em integração até 2014.

Os sistemas mencionados no decorrer do trabalho estão descritos a seguir, juntamente com as tecnologias utilizadas (Tabela 4.1).

**CIN\_Cadastro** Sistema de apoio e controle de fotos de cartões de identificação pessoal, emails de setores, de pesquisadores, liberação de acesso do fecho eletrônico do DEG<sup>1</sup>-UFLA, entre outras atividades, utilizado diariamente;

**CPPD** Sistema de cálculo de progressão de professores utilizado até 2011;

**HCS** Sistema de ponto eletrônico, auxilia em atividades como gerência de digitais e registro de ponto;

**Pergamum** Sistema integrado de bibliotecas, utilizado no gerenciamento do acervo, empréstimos de obras, entre outras atividades da Biblioteca da UFLA;

**SCIN** Sistema de suporte utilizado até 2011;

---

<sup>1</sup>Departamento de Engenharia

**SIG-UFLA** Sistema institucional de controle acadêmico. Utilizado por diversos departamentos para controle do processo de chamadas a candidatas, controle de alunos, presença, emissão de documentos, como histórico escolar, controle de matrizes curriculares, cursos, períodos letivos, controle do processo de renovação de matrícula, lançamento de notas, entre outras atividades.

**SIGAA** Sistema que auxilia no controle de pesquisas, bolsistas Bic Junior (alunos de ensino médio) e produção intelectual. A partir de 2015, será utilizado também no controle de cursos e estudantes de Pós Graduação;

**SIGRH** 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;

**RV3Acesso** Sistema que controla catracas na UFLA, presentes no Restaurante Universitário, Biblioteca, Alojamento Estudantil e outros;

**Veículos** Sistema de controle de veículos na Instituição utilizado até 2010.

Na Tabela 4.1, por exemplo, tem-se o SIG-UFLA, que foi desenvolvido utilizando a linguagem PHP<sup>2</sup>, o SGBD MySQL, e é acessado via *browser*, por ser um sistema Web. Os serviços oferecidos pela UFLA mencionados no decorrer do trabalho estão descritos a seguir. Ambos os serviços utilizam o sistema operacional Linux, Fedora para o LDAP e Red Hat para o Zimbra.

**LDAP** Protocolo de aplicação aberto, que permite organizar os recursos de rede de forma hierárquica (HARDWARE, 2005);

---

<sup>2</sup><http://php.net/>

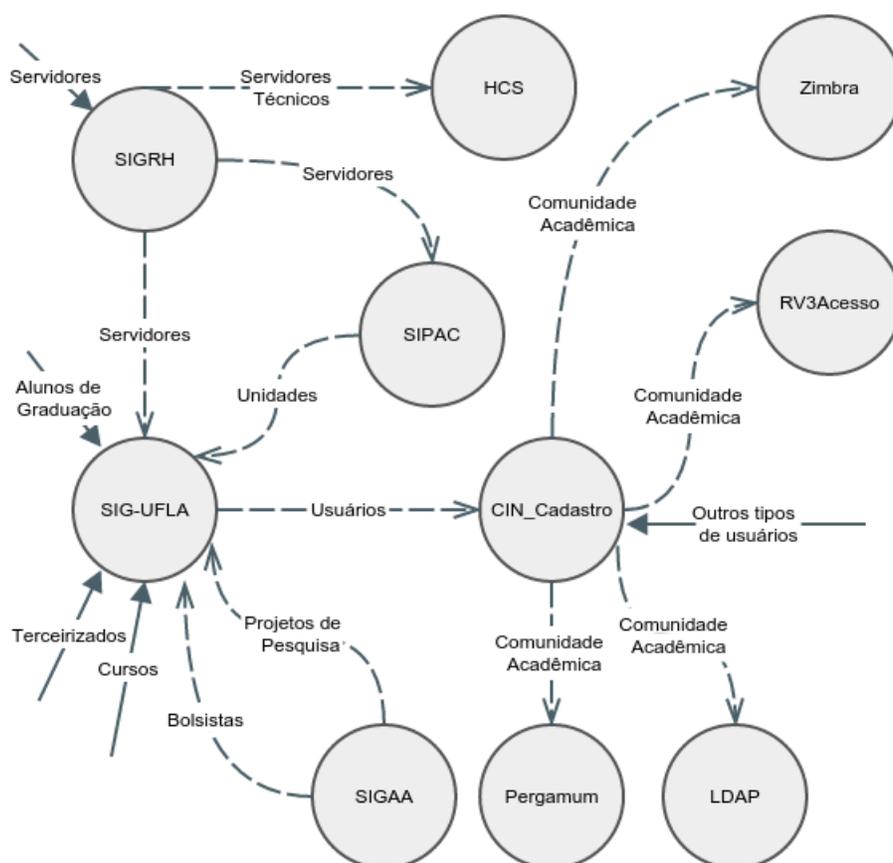
	Linguagem	Banco de Dados	Arquitetura
<b>CIN_Cadastro</b>	PHP	MySQL	Web
<b>CPPD</b>	PHP	MySQL	Web
<b>HCS</b>	-	MySQL	Web
<b>Pergamum</b>	Delphi/Java	SQL Server	Desktop/Web
<b>SCIN</b>	PHP	MySQL	Web
<b>SIG-UFLA</b>	PHP	MySQL	Web
<b>SIGAA</b>	Java	PostgreSQL	Web
<b>SIGRH</b>	Java	PostgreSQL	Web
<b>SIPAC</b>	Java	PostgreSQL	Web
<b>RV3Acesso</b>	-	MySQL	Desktop
<b>Veículos</b>	PHP	MySQL	Web

**Tabela 4.1:** Sistemas mencionados e suas respectivas tecnologias.

**Zimbra** Sistema de mensagens e suíte de colaboração que oferece serviços de email.

Os sistemas que utilizaram a linguagem PHP foram desenvolvidos pela própria DGTI. Houve também a compra de licenças para utilização dos *softwares* Pergamum e RV3Acesso. Os sistemas Zimbra e LDAP são *open-source* e de propósito geral, não sendo específicos de ambiente universitário. Por isso, tais serviços não se encontram na Tabela 4.1.

Os sistemas e serviços na UFLA compartilham dados como nome de usuário, domínio, endereço e email. Todos os sistemas e serviços dependem, em diferente escala, dos dados gerenciados pelo CIN\_Cadastro e SIG-UFLA. Na Figura 4.1, são mostrados alguns dados compartilhados entre sistemas, por exemplo, o CIN\_Cadastro compartilha dados da comunidade acadêmica com os sistemas Pergamum, LDAP, RV3Acesso e Zimbra.



**Figura 4.1:** Exemplo da interação e dados compartilhados entre sistemas da Integração.

## 4.2 Estudo de Caso

Nesta Seção, estão apresentados os resultados do estudo de caso. Após as entrevistas e análises, pôde-se agrupar os fatos em ordem cronológica e identificar quatro momentos:

**Pré-Integração** Descreve o cenário anterior à primeira integração e a motivação para sua implantação, até meados de 2006;

**Momento 1** Período entre meados de 2006 e meados de 2009;

**Momento 2** Período que compreende de meados de 2009 até Junho de 2014;

**Momento 3** Período entre Junho e Novembro de 2014.

#### 4.2.1 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. Até mesmo as contas de email 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<sup>3</sup> MySQL, o banco de dados foi criado passou a ser a base mais atualizada na Instituição. Esta base de dados foi chamada de BD-UFLA e passou a armazenar os dados comuns necessários aos outros sistemas. Após a criação desta base de dados, os sistemas “satélites” começaram a substituir suas próprias tabelas para utilizar as deste banco.

#### 4.2.2 Momento 1

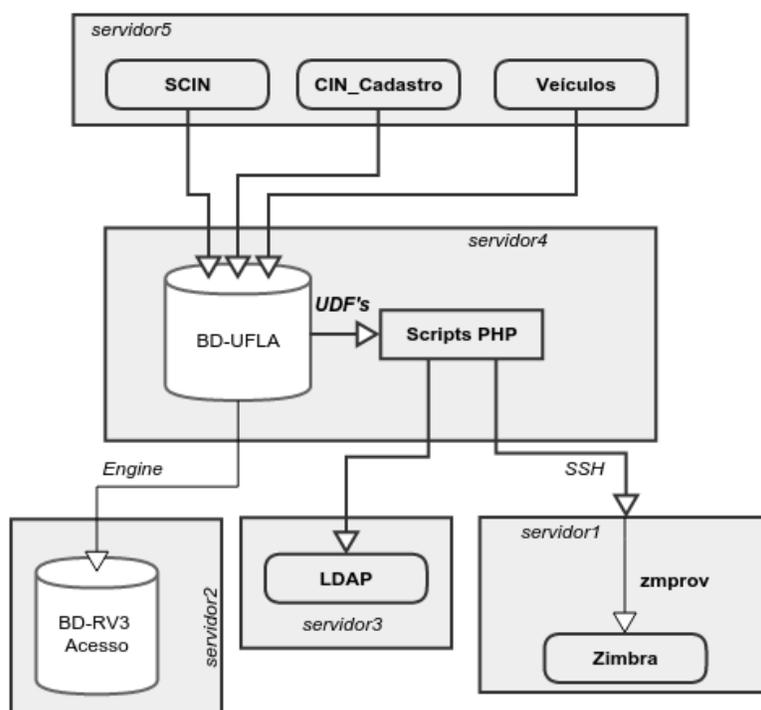
A primeira integração na UFLA teve o compartilhamento de informações entre os sistemas: CIN\_Cadastro, SCIN, Veículos, RV3Acesso, LDAP e Zimbra. As informações compartilhadas entre tais sistemas eram dados como: endereço, nome, cargo e unidades de lotação.

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 que este SGBD disponibiliza, como tabelas, *Triggers*, *Functions*, *Procedures*, *Views* e *Events*. Para o RV3Acesso, foi utilizado o *engine Federated*, que compartilha os dados através da criação de uma tabela virtual, que recebe os dados da tabela original (MYSQL, 2005).

---

<sup>3</sup>Sistema Gerenciador de Banco de Dados

Para os demais sistemas que participavam da integração, a abordagem era através do uso de *UDF*<sup>4</sup>. *UDF* são funções desenvolvidas em C e compiladas dentro do SGBD, utilizadas como funções nativas do MySQL. Utilizando este recurso, foi possível implementar uma atualização “instantânea”. 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 efetivavam a integração com os outros sistemas. Na Figura 4.2, pode-se encontrar um diagrama representando esta integração inicial.



**Figura 4.2:** Integração inicial entre os sistemas da UFLA.

A utilização do PHP foi decidida devido à habilidade dos profissionais que desenvolveram a integração inicial com esta linguagem. De acordo com as

<sup>4</sup><http://dev.mysql.com/doc/refman/5.1/en/adding-udf.html>

informações obtidas por meio do questionário, este foi o único critério para a decisão pela adoção do PHP para este fim.

É importante ressaltar a utilização de diferentes servidores pelos sistemas, configurando também uma heterogeneidade de sistemas operacionais (SO), ainda que pouca. Conforme mostrado na Figura 4.2, a integração é composta por sistemas presentes em 7 servidores. Destes, **servidor1**, **servidor3**, **servidor4**, **servidor5** e **servidor7** utilizam o SO Linux, distribuição Fedora<sup>5</sup> e os outros contam com o sistema operacional Windows Server 2003.

	<b>Sistema Operacional</b>
<b>servidor1</b>	Linux Fedora
<b>servidor2</b>	Windows Server 2003
<b>servidor3</b>	Linux Fedora
<b>servidor4</b>	Linux Fedora
<b>servidor5</b>	Linux Fedora
<b>servidor6</b>	Windows Server 2003
<b>servidor7</b>	Linux Fedora

**Tabela 4.2:** Servidores e seus respectivos Sistemas Operacionais.

A abordagem utilizada neste momento, segundo (LAND; CRNKOVIC, 2003), seria um híbrido de “**Integração em Nível de Dados**” e “**Enterprise Application Integration**”, pois, de certa forma, compartilham o sistema de banco de dados ao mesmo tempo que utilizam conectores para integração entre os sistemas que não participam do compartilhamento a nível de dados.

De acordo com as abordagens centradas na organização e seus processos, vistas em (HOHPE; WOOLF, 2003), esta abordagem se classificaria como “**Data Replication**”, pois os dados são replicados em diferentes bases (bancos de dados, LDAP e Zimbra institucionais) em um processo automático executado pelos *scripts* PHP.

---

<sup>5</sup>fedoraproject.org/

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 Figura 4.2, o *script* PHP, invocado pelas *UDF*, precisava estabelecer uma conexão SSH<sup>6</sup> com o servidor onde o Zimbra estava instalado, para executar comandos da ferramenta **Zmprov**<sup>7</sup>. Esta ferramenta oferecia a possibilidade de execução de comandos direto no Zimbra. Com este encadeamento, a detecção de erros e tratamento de exceções se tornava complexa e de difícil manutenção.

Sistemas como o Pergamum, por exemplo, eram atualizados manualmente. Os dados eram enviados em planilhas, e o profissional de Tecnologia da Informação lotado na Biblioteca da UFLA se encarregava de inserir tais informações neste sistema.

O uso de *UDF* trouxe a sensação de instantaneidade na integração, pois eram executadas assim que uma *trigger* era disparada. Embora esta abordagem tenha atingido seus objetivos, as *UDF* eram difíceis de dar manutenção, pois eram programadas na linguagem C e compiladas dentro do SGBD (MYSQL, 2005). Sua utilização também dificultava a detecção de erros, quando estes aconteciam.

### 4.2.3 Momento 2

Após a primeira integração, ocorreram mudanças. O **SIG-UFLA** começou a ser desenvolvido e, progressivamente, agregou 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-UFLA, CIN\_Cadastro, Pergamum, RV3Acesso, Zimbra e LDAP.

Com a entrada do SIG-UFLA e seu rápido crescimento, surgiu uma série de novos problemas. A crescente utilização deste sistema fazia com que as tabelas

---

<sup>6</sup>Secure Shell

<sup>7</sup><http://wiki.zimbra.com/wiki/Zmprov>

fossem atualizadas mais frequentemente, trazendo consigo o aumento de disparo de *triggers*, que por sua vez invocavam as *UDF* com frequência cada vez maior.

Esta crescente demanda da utilização das *UDF* as sobrecarregava e as tornavam lentas. 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 à sobrecargas na rede.

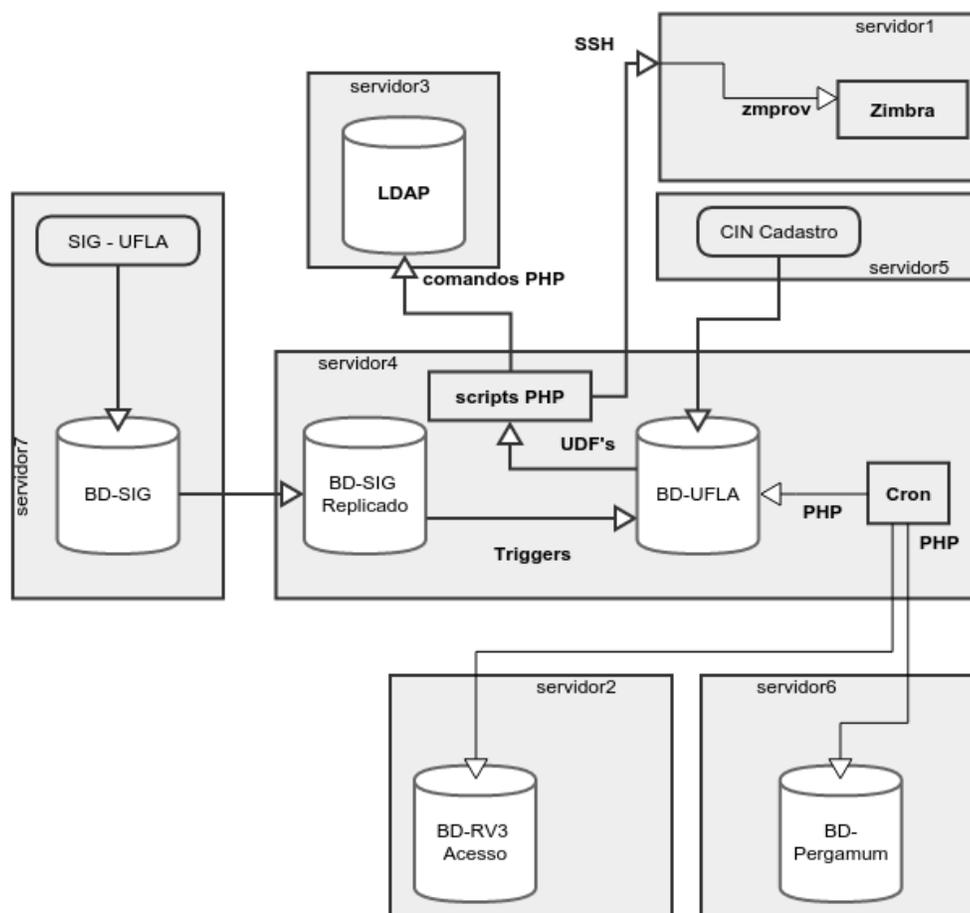
A solução dada para esta situação foi colocar o banco de dados do SIG-UFLA (BD-SIG), em um servidor dedicado. Feito isso, uma réplica assíncrona desta base foi configurada no mesmo servidor onde se encontrada o BD-UFLA, além das consultas e relatórios que requeriam mais processamento terem sido direcionados à ela. Com isso, o servidor de produção do BD-SIG ficou menos requisitado, o que resolveu, em partes, os problemas com as *UDF*, que ficam lentas com a sobrecarga do BD-SIG.

Mesmo com os problemas com as *UDF* parcialmente resolvidos, buscou-se uma opção alternativa à elas. Neste momento, o Cron<sup>8</sup> - agendador de tarefas existente em sistemas operacionais derivados do Unix - se mostrou uma alternativa. A opção pela utilização do Cron trouxe vantagens como facilidade na configuração, utilização e manutenção. As chamadas à *scripts* PHP passaram então 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, deixando para trás apenas sistemas críticos como LDAP e Zimbra. Com o foco no SIG-UFLA, o funcionamento da integração neste Momento está representado pela Figura 4.3.

---

<sup>8</sup><http://www.adminschoice.com/crontab-quick-reference/>



**Figura 4.3:** Integração entre sistemas na UFLA no Momento 2.

1. Alteração no SIG-UFLA dispara modificação nos dados do BD-SIG;
2. BD-SIG original então 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 no *buffer* de atualizações do BD-UFLA, 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-UFLA, gera registros específicos para atualização dos sistemas, definindo qual sistema deve ser atualizado, e em qual registro no BD-UFLA 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-UFLA e os propagam para o BD-Pergamum e BD-RV3Acesso;
6. *Scripts* PHP são executados também a partir de *UDF* do BD-UFLA, para propagar atualizações para LDAP e Zimbra.

Embora as *UDF* tenham passado a ser menos utilizadas, a abordagem do Momento 2 ainda usava esta tecnologia para integrar serviços críticos, como LDAP e Zimbra. Assim como a do Momento 1, a abordagem do Momento 2 também atingiu seu objetivo, que era a integração dos sistemas na Universidade. Porém, a adição de novos sistemas à integração tal qual ela era neste momento ainda trazia complexidades, como 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 monitorável. 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-UFLA, que enviavam email 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.

#### 4.2.4 Momento 3

As *UDF* continuavam em funcionamento, mesmo com suas limitações quanto à performance e manutenibilidade. A atualização do sistema operacional no servidor em que se encontrava o BD-UFLA bem como a atualização da versão do SGBD MySQL de 4 para 5 trouxeram instabilidade às *UDF*, como a incompatibilidade entre tipos de dados devido à modificação dos nomes. Com esta instabilidade, vieram também os já conhecidos problemas relativos ao tratamento de exceções e difícil 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 a necessidade de alta escalabilidade e manutenibilidade.

A Arquitetura orientada a Serviço foi então uma alternativa promissora, trazendo a possibilidade de efetivar uma integração com qualidade e monitorável. Algumas decisões precisaram ser tomadas antes de iniciar o projeto a fim de que as características desejadas, como segurança, escalabilidade, manutenibilidade e tratamento de erros e exceções fossem obtidas. Esta estratégia visou garantir uma vantagem significativa sobre a abordagem anterior. Foram tomadas decisões como:

**Segurança** Este era um item primordial, visto que a integração manipularia dados pessoais, como nome, CPF<sup>9</sup>, senhas, emails e endereços de milhares de pessoas. Então, seria necessária uma forma de segurança que restringisse o acesso apenas à 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 restringisse a utilização de serviços à *tokens* autorizados, acessando a partir de endereços de *IP*<sup>10</sup> específicos;

---

<sup>9</sup>Cadastro de Pessoa Física

<sup>10</sup>*Internet Protocol*

**Linguagem** Foi decidido que toda 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;

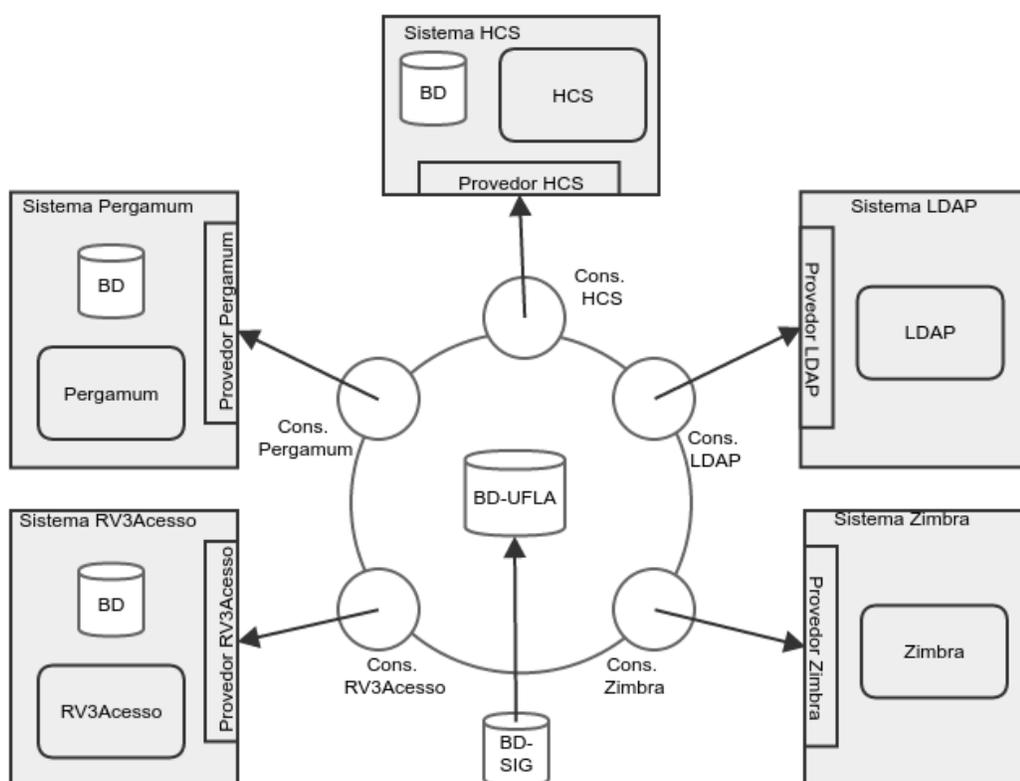
**Relação Provedor vs. Serviço** Foi discutida tal relação no aspecto numérico, na qual foi feita a seguinte pergunta: “Seria mais interessante uma abordagem utilizando um único provedor e vários serviços ou vários provedores com poucos serviços?” A opção foi pela arquitetura com vários provedores e 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: ID, que carregaria o ID da mensagem e do sistema relacionado à integração; uma mensagem; um tipo, como por exemplo: “SUCESSO”, “ERRO\_BD”; e um atributo *system*, que traria o nome do serviço requisitado;

**Abordagem** Com uma visão centrada na implementação, conforme (MARTINS, 2005), optou-se pela abordagem que o autor chama de “Aplicações Compostas”, pois as aplicações seriam integradas através do *SOAP*, que é uma *API*. Quanto às estratégias, de acordo com (LAND; CRNKOVIC, 2003), a abordagem que se assemelhou mais com a forma da integração do Momento 3 foi a “*Enterprise Application Integration*”, pois a integração ocorreria dentro de uma única empresa, utilizando de conectores entre os variados sistemas. Com o foco na organização e seus processos, como

visto em (HOHPE; WOOLF, 2003), se trata da “Arquitetura Orientada a Serviço”, pois os consumidores (sistemas) da integração foram pensados para consumir funcionalidades através de serviços, configurando então uma Integração Orientada a Serviço.

Com base nestas decisões, chegou-se à uma arquitetura apresentada na Figura 4.4 .



**Figura 4.4:** Integração entre sistemas na UFLA no Momento 3.

O funcionamento desta integração é similar às outras abordagens, pois o “disparo” inicial é dado por atualização no SIG-UFLA ou em sistemas conectados ao BD-UFLA. A partir disso:

1. Cron executa os consumidores de serviço, que identificam atualizações para seus respectivos sistemas diretamente na tabela de *buffer* do BD-UFLA;

2. Os consumidores invocam os serviços de atualização, atualizando 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 entre provedores e consumidores, passível de falhas.

No Momento 3, alguns aspectos da integração foram facilitados. Como exemplo, têm-se a atualização de versão do Zimbra, que passou a fornecer acesso à suas funções via *Web Services*. No entanto, alguns complicadores também foram inseridos, como a adição do BD-SIPAC, utilizando o SGBD PostgreSQL, diferente de todos os outros BD's 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 intrinsecamente trás consigo a 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 UFLA para acesso à funcionalidades. Essa interoperabilidade é importante, pois permite a interconexão de *softwares* desenvolvidos em diferentes linguagens, como por exemplo, um consumidor construído em Java pode se conectar aos Serviços, desenvolvidos em PHP, sem problemas.

Para organizar os provedores, foram criadas duas categorias: Provedores Públicos, e Provedores Privados. Os Provedores Privados são utilizados somente pelos sistemas da integração. Os Provedores Públicos foram desenvolvidos para disponibilizar serviços para sistemas externos à integração, isolando os Provedores Privados. Este isolamento é interessante para a questão de segurança, pois os consumidores dos serviços públicos não têm conhecimento dos serviços privados

existentes. As Figuras 4.5 e 4.6 exemplificam essa separação dos Serviços em categorias e mostram como funcionam seus respectivos fluxos de chamadas.

Na Figura 4.5, pode-se ver uma chamada de um consumidor privado a um serviço privado, representado pela seta com linha contínua, e o retorno desta chamada, representado pela seta com linha pontilhada.



**Figura 4.5:** Fluxo de chamadas no acesso aos *Web Services* Privados partindo de um consumidor privado.

Na Figura 4.6, 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 fazem chamadas ao Provedor Privado. Isso também pode ser interessante para encapsular processos como um todo e disponibilizá-los como um único serviço para consumidores públicos.



**Figura 4.6:** Fluxo de chamadas no acesso aos *Web Services* Públicos partindo de um consumidor público / externo.

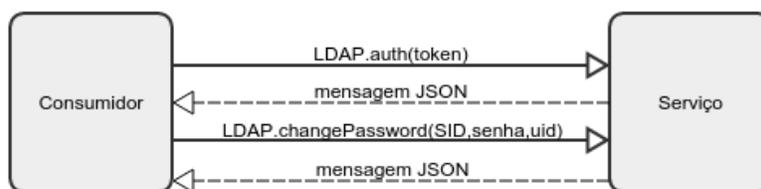
As chamadas entre os Consumidores e os Serviços Privados são realizadas em três passos (Figura 4.7):

1. Consumidor chama o método 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*. O serviço também

carrega as informações sobre as funcionalidades que podem ser acessadas pelo *token* recebido;

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, ou SID. Caso não seja do tipo “SUCESSO”, o atributo *message* será uma mensagem explicativa relacionada ao problema de autenticação.

Na Figura 4.7 é apresentado um fluxo de chamadas e retornos utilizando o Provedor do LDAP, especificamente os serviços *auth* e *changePassword*, para autenticação e troca de senha. O consumidor invoca o serviço *auth*, e recebe uma resposta formatada em JSON, contendo o ID da sessão. Então, o consumidor invoca novamente o Provedor, mas dessa vez o serviço *changePassword*, enviando o ID de sessão recebido previamente, a senha e o login do usuário para a troca de senha no LDAP. Então, o serviço retorna uma resposta também formatada em JSON, sinalizando o sucesso ou a falha da operação.



**Figura 4.7:** Fluxo de chamadas utilizando o *Web Service*.

No caso das chamadas aos Serviços Públicos, não há o passo de autenticação. As chamadas são feitas diretamente às funcionalidades, de forma semelhante à chamada aos Serviços Privados, porém passando o *token* no lugar o SID.

Para que a integração fosse monitorável, foi desenvolvido um sistema gerencial, em que há a possibilidade de cadastro de novos *Web Services*,

funcionalidades para os serviços, *tokens*, vinculação de *tokens* a IP's e funcionalidades, e por fim, um *dashboard* com informações sobre as execuções dos serviços nas últimas horas, comparação entre utilização de serviços nas últimas duas semanas e dados de quando foram as últimas execuções de cada serviço. Três dos gráficos presentes no *dashboard* são apresentados nas Figuras 4.9, 4.10 e 4.11.

Na Figura 4.8, é apresentada a tela principal do Sistema de Gerenciamento da Integração, com diversos gráficos exibindo dados como: momento da última chamada à serviços de cada sistema da Integração e quantidade de chamadas nas últimas 24 horas.

Estadísticas gerais

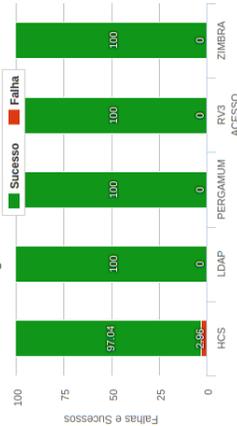
Provedores cadastrados: 7  
 Serviços cadastrados: 40  
 Tokens cadastrados: 13  
 Usuários cadastrados: 6

Última execução - Provedores

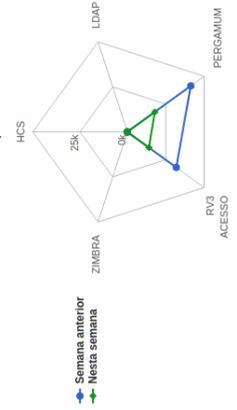
HCS: 28/10/2014 às 17:28:41  
 LDAP: 28/10/2014 às 16:48:20  
 RV3 ACESSO: 28/10/2014 às 17:28:45  
 LOG: Desabilitado  
 PERGAMUM: 28/10/2014 às 17:28:56  
 WS: 28/10/2014 às 16:48:20  
 ZIMBRA: 28/10/2014 às 17:28:36

Estatística de Utilização da Última Semana

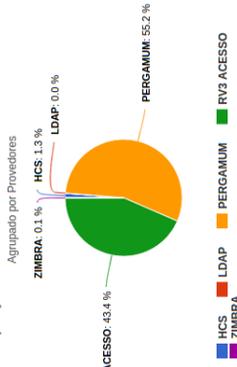
Porcentagem de sucesso e falha



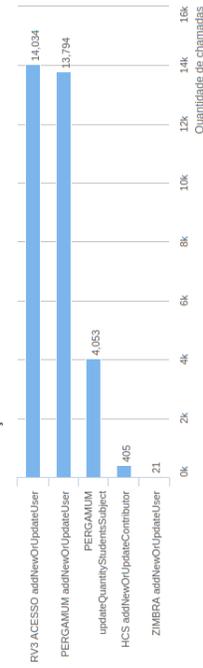
Esta semana x Semana passada



Proporção entre chamadas à Web Services



Serviços mais utilizados na semana



Chamadas nas últimas 24 horas

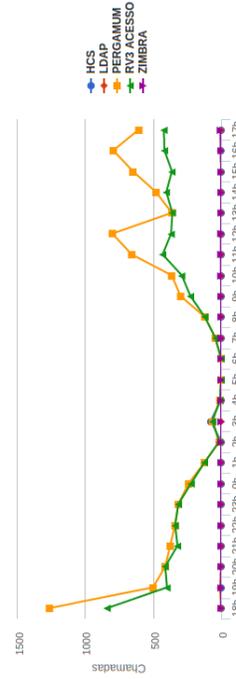
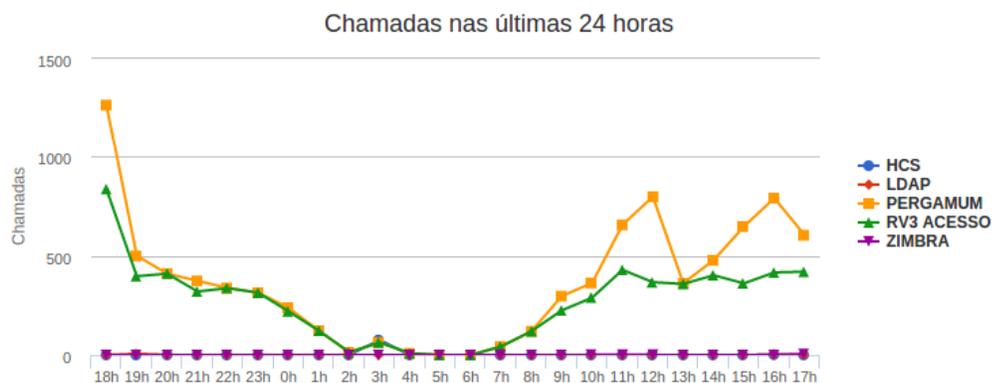


Figura 4.8: Dashboard de monitoramento no sistema gerencial da Integração.

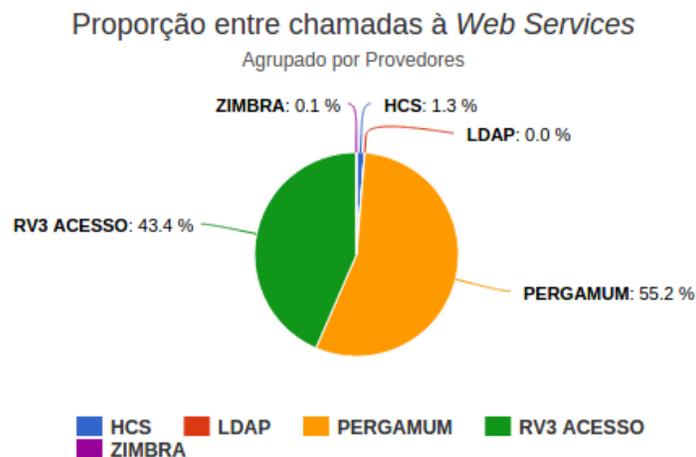
A Figura 4.9 apresenta o gráfico da Sistema de Gerenciamento que exibe a quantidade de chamadas aos serviços de cada sistema nas últimas 24 horas. Por exemplo, entre 18 e 19 horas do dia anterior à visualização deste gráfico, o Provedor Pergamum teve por volta de 1300 chamadas a seus serviços.



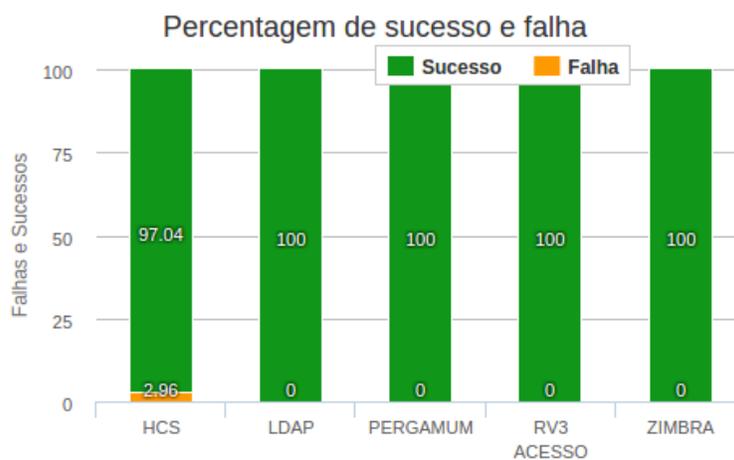
**Figura 4.9:** Gráfico de execuções de serviços por hora.

A Figura 4.10 apresenta o gráfico de pizza comparando 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%.

A Figura 4.11 apresenta gráficos comparando 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.



**Figura 4.10:** Gráfico de comparação global entre os *Web Services* da Integração.



**Figura 4.11:** Gráfico de comparação de percentagem de Sucesso e Falha entre os *Web Services* da Integração.

### 4.3 Discussão

As informações que permitiram a definição dos momentos de integração foram obtidas por meio de entrevista com os envolvidos na integração desde o início e foi utilizada a pouca documentação e os *scripts* existentes sobre as abordagens de integração utilizadas. Com as respostas obtidas nas entrevistas, foi

possível delinear os momentos e os escopos da integração. A documentação e os *scripts* auxiliaram na identificação das tecnologias e as abordagens de integração. Percebeu-se que a integração, inicialmente, envolvia poucos sistemas, era de difícil manutenção e pouco escalável.

Os sistemas e serviços da integração na UFLA possuem diferentes características quanto às tecnologias utilizadas no desenvolvimento, nos bancos de dados, e rodando sobre diferentes sistemas operacionais. A heterogeneidade do ambiente é um desafio, visto que é necessário um “denominador comum” para realizar as atividades da integração, interconectando os sistemas.

A primeira integração utilizava essencialmente uma abordagem de integração via base de dados, quando compartilhavam a tecnologia de SGBD, e *UDF* - recurso do SGBD - para atualização de sistemas externos que não utilizavam o mesmo SGBD. Tais recursos eram de difícil desenvolvimento, visto que eram programadas em C e compiladas dentro do SGBD. A entrada de novos sistemas no ambiente universitário aumentou a heterogeneidade no ambiente, tornando a integração ainda mais desafiadora e complexa.

Após o surgimento de problemas de estabilidade com a utilização das *UDF*, estes novos sistemas foram integrados com auxílio do Crontab, sistema de agendamento de tarefas de sistemas Unix. Com isso, a integração deixou de ser “instantânea” - visto que as *UDF* eram executadas por trigger, logo após a inserção ou atualização de dados. No entanto, esta passou a ser mais confiável, devido à maior facilidade de manutenção, e reduziu a sobrecarga do SGBD. A implantação da integração com utilização do Crontab foi parcial: sistemas críticos como LDAP e Zimbra ainda eram atualizados via *UDF*.

Uma atualização de versão do SGBD, e as *UDF* passaram a ficar instáveis. Uma única atualização nos sistemas LDAP e Zimbra levava de 3 a 5 segundos, tornando a utilização das *UDF* impraticável. Então, a arquitetura da integração foi repensada, de modo a ser monitorável, escalável, manutenível e segura.

Uma dificuldade, durante a fase de projeto da arquitetura do Momento 3, foi a busca por *frameworks* específicos para desenvolvimento de *Web Services*. No entanto, foram encontrados apenas módulos pertencentes a outros *frameworks*, como por exemplo o Zend SOAP Server<sup>11</sup> e CodeIgniter XML-RPC<sup>12</sup>. Este assunto foi discutido pela equipe, e optou-se por PHP Orientado a Objetos, sem *framework* e utilizando a biblioteca nuSOAP<sup>13</sup> que, com poucas adaptações (relativas à codificação de caracteres), atendeu às necessidades da integração.

Com o passar do tempo, a abrangência da integração sobre os sistemas de informação no ambiente Universitário cresceu expressivamente, junto com o aumento da quantidade destes sistemas. A abordagem do Momento 3 trouxe melhorias significativas em relação às abordagens anteriores, provendo uma integração escalável e manutenível. Com a expectativa da continuidade do crescimento do número de sistemas envolvidos nesta integração, algumas medidas podem ser tomadas visando melhorar a integração em um aspecto geral:

- Criação de um serviço de acesso aos dados do BD-UFLA. Essa medida faria com que clientes de *Web Services* de sistemas como Pergamum e RV3Acesso consumissem um serviço encapsulado de dados, ao invés de acessar o banco de dados BD-UFLA diretamente. Com isso, as manutenções relativas ao BD-UFLA poderiam ser feitas em um único componente. Como efeito colateral, têm-se um incentivo pela utilização de sessões para armazenar permissões, visto que o acesso ao banco para obtenção de permissões pode impactar levemente na performance;
- Caso haja a possibilidade de entrada de outro sistema que também possa concentrar dados a serem compartilhados, ou seja, em conjunto com BD-SIG e BD-UFLA, deve-se considerar também a hipótese do desenvolvimento de uma integração à parte para envolver e isolar os bancos

---

<sup>11</sup><http://framework.zend.com/manual/1.12/en/zend.soap.server.html>

<sup>12</sup><https://ellislab.com/codeigniter/user-guide/libraries/xmlrpc.html>

<sup>13</sup><http://sourceforge.net/projects/nusoap/>

de dados com o uso de uma arquitetura semelhante ao padrão de projeto Facade;

- Quanto ao Sistema Gerenciador, seria interessante o desenvolvimento de um *dashboard* para controle do Provedor e Serviços Públicos de forma semelhante como há para os Provedores Privados, exibindo chamadas, logs, origem de acessos; e o desenvolvimento de relatórios com geração de gráficos, exibindo dados limitados por períodos, sistemas ou serviços específicos.
- Requisições e Retornos SOAP exigem a transferência de muita informação, visto que os dados transmitidos ou recebidos são encapsulados em um documento SOAP. Embora essa característica não tenha sido relevante inicialmente, com a evolução da Integração e o aumento da quantidade de sistemas e dados a serem disponibilizados pelos *Web Services*, esses documentos SOAP podem impactar negativamente na performance da Integração. Então, a utilização da arquitetura REST pode ser uma solução para este problema, visto que as mensagens de retorno são definidas exclusivamente pelo sistema, não sendo encapsulado em nenhuma estrutura, como no SOAP. Pela filosofia REST, não deve-se salvar dados na sessão. Logo, o acesso ao banco de dados para a obtenção de dados sobre permissões ocorrerá frequentemente, podendo também impactar na performance da Integração.
- Conforme discutido durante o trabalho, os Provedores foram implementados de forma independente, inclusive em relação aos módulos internos referente à operações em comum, como autenticação e registro de Log. Os Provedores podem ser refatorados, de forma a modularizar as operações comuns, e então, compartilhar tais módulos. Esta medida teria um impacto positivo na manutenibilidade dos Provedores da Integração.

## 5 TRABALHOS RELACIONADOS

Na literatura, existem trabalhos que relatam propostas de integração de sistemas em ambientes acadêmicos, e utilização de *SOA* como abordagem de integração.

Em (ALVES; AMARAL; PIRES, 2005), é apresentado um estudo de caso sobre os sistemas de informação na Escola Superior de Tecnologia e de Gestão (ESTiG) de Bragança (Portugal). Os autores apresentam o processo de desenvolvimento da Intranet desta escola superior. O principal problema abordado foi a falta de aceitação e motivação para o uso do ambiente de aprendizado (*e-learning*), e a solução encontrada pelos autores foi a integração com sistemas de gestão, que auxiliava em atividades como gestão de pesquisa, financeira e registro de faltas. Com uma maior utilização e posteriormente maior aceitação por parte dos professores, as atividades de *e-learning* passariam a ser incentivadas também para os alunos. A abordagem utilizada por eles para integrar os sistemas foi desenvolver dois módulos: *e-learning* e *e-management* para a serem utilizados por professores, alunos e pessoas com cargo administrativo. Essa integração teve uma interface disponibilizada para estes usuários e acessava dados de sistemas legados.

A integração descrita em (ALVES; AMARAL; PIRES, 2005) usa de intranet para integrar os sistemas. As abordagens utilizadas não estão muito detalhadas, no entanto, é possível afirmar que a integração foi o meio para a estratégia de estimular o maior uso do ambiente virtual na ESTiG. O trabalho aqui apresentado teve como foco a apresentação da integração entre sistemas, separados em “Momentos” cronológicos, e tal integração tem como objetivo a automatização e a consistência de dados entre os sistemas e, em especial o Momento 3, teve foco na manutenibilidade e escalabilidade da integração, além de prover uma interface de gerência e monitoramento.

Em (CUNHA; JUNIOR; ALMEIDA, 2005) e (CUNHA; JUNIOR; DORNELAS, 2008), são apresentados aspectos relativos à integração de sistemas

no CEFET-AL (Brasil). Em (CUNHA; JUNIOR; ALMEIDA, 2005), são apresentados os sucessos e fracassos da tentativa de integração entre sistemas externos e internos no CEFET, dentro de um período de 2 anos. São apresentadas também as características do CEFET quanto à organização hierárquica da instituição. Os autores apresentaram cinco problemas da falta de integração, entre eles: registro redundante (e inconsistente) de funcionários e alunos; e falta de conexão entre dados acadêmicos e bibliotecários, dificultando o controle de empréstimos de livros. Para cada um dos cinco problemas, foi sugerida uma solução e foram descritas consequências da implantação da integração, juntamente com uma proposta para a integração no CEFET.

Em (CUNHA; JUNIOR; DORNELAS, 2008), é apresentada uma proposta de integração entre os sistemas externos e internos ao CEFET utilizando uma abordagem orientada a serviço. Os autores apresentam uma sequência do trabalho exposto em (CUNHA; JUNIOR; ALMEIDA, 2005), propondo a utilização da SOA para solucionar os problemas evidenciados. Os autores propuseram a integração dos serviços (consumidores) com os sistemas integráveis através do acesso direto ao banco de dados, enquanto o acesso aos serviços ocorria por meio de uma interface Web.

Assim como no Momento 1, os autores em (CUNHA; JUNIOR; ALMEIDA, 2005) tiveram problemas com a manutenibilidade e falta de padronização entre os sistemas desenvolvidos internamente por alunos e programadores da Gerência de TI da instituição. Os autores apresentaram propostas para a integração entre os sistemas, no entanto sem especificar uma tecnologia.

Em (CUNHA; JUNIOR; DORNELAS, 2008), é apresentada uma proposta orientada a serviço, assim como a abordagem do Momento 3 do trabalho aqui apresentado. Os autores utilizaram a SOA provendo uma interface Web aos usuários, separados por sistemas. Os serviços desenvolvidos por eles acessam

diretamente a base de dados dos sistemas da integração. Comparativamente, no Momento 3, a integração entre os sistemas é todo feito de forma transparente, isso é, a atualização de dados é propagada para os outros sistemas como se todos fossem um único sistema. Por exemplo, uma alteração de nomes no SIG-UFLA reflete, em poucos minutos, na alteração de nome no Zimbra, LDAP, Pergamum e RV3Acesso. A abordagem utilizada em (CUNHA; JUNIOR; DORNELAS, 2008), segundo (HOHPE; WOOLF, 2003), seria *Information Portals*, visto que os sistemas são integrados na camada de apresentação, enquanto no Momento 3, a abordagem é *Data Replication*.

## 6 CONCLUSÃO

A utilização de SI no ambiente empresarial vem crescendo ao longo dos anos. No mundo real, é comum de se encontrar um cenário com vários sistemas, utilizados para diferentes fins. Para a consistência dos dados manipulados por estes sistemas, é ideal que 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 na Universidade Federal de Lavras (UFLA). No estudo, foi possível identificar 4 momentos de integração, desde o momento que não havia integração, em meados de 2006, até após o desenvolvimento de uma arquitetura de integração e sua implantação no ambiente, em Novembro de 2014. A busca por informações sobre os Momentos da integração foi complicada, devido à dependência essencialmente da memória dos funcionários. Havia pouca documentação sobre a integração além do código-fonte dos scripts do Momento 2. No Momento 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 UFLA trouxe um entendimento maior sobre sua evolução, suas necessidades iniciais, vantagens e desvantagens em cada um dos momentos descritos neste trabalho. 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 (e-mail, autenticação centralizada, wireless) presentes no cotidiano universitário.

Este trabalho pode servir como base de referência para outras instituições de ensino em situação semelhante, além de permitir o registro histórico da integração para a DGTI-UFLA. Este registro permitirá à DGTI entender melhor a evolução da integração, podendo auxiliar também em decisões futuras, para as próximas melhorias na integração dos sistemas e serviços. Como trabalhos futuros, sugere-se:

- Estudo e desenvolvimento de um *framework* PHP voltado especificamente para o desenvolvimento de *Web Services*, devido à dificuldade encontrada na fase de projeto;
- Proposta de uma forma de documentar os serviços: pelos problemas encontrados durante o estudo;
- Estudo mais aprofundado sobre os aspectos de segurança em *Web Services* para aplicação neste projeto, pois no projeto da arquitetura do Momento 3, os aspectos de segurança foram tratados a nível de código apenas;
- Proposta de um processo de integração dentro da própria DGTI, que ainda não possui uma descrição dos passos de como adicionar um novo sistema à integração, ou mesmo para solicitação de novos serviços;
- Identificação de pontos que podem ser comuns às instituições de ensino, de forma a fornecer um padrão para integração neste tipo de ambiente;
- Proposta de adequação aos padrões de interoperabilidade e-Ping<sup>1</sup>, do Governo Federal;
- Proposta de integração com sistemas estruturantes do Governo Federal, como: SIAPE, SIGEPE, Comprasnet, visto que a Universidade que teve a integração de seus sistemas como objeto de estudo se trata de uma instituição federal.

---

<sup>1</sup><http://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padroes-de-interoperabilidade>

## REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES, P.; AMARAL, L.; PIRES, J. A. Integração de sistemas de informação em instituições de ensino superior usando intranets: um estudo de caso. 6<sup>a</sup> Conferência da Associação Portuguesa de Sistemas de Informação, CAPSI, 2005.
- ANDERSSON, V. O.; SANTOS, R. T. dos; 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.
- 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.
- CARMO, B. do; 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*, v. 19, n. 1, p. 9, 2013.
- CERAMI, E. *Web services essentials: distributed applications with XML-RPC, SOAP, UDDI & WSDL*. [S.l.]: "O'Reilly Media, Inc.", 2002.
- COSTA, C.; MELO, A. C.; FERNANDES, A.; GOMES, L. M.; GUERRA, H. Integração de sistemas de informação universitários via web services. In: *Actas da 5<sup>a</sup> Conferencia Ibérica de Sistemas y Tecnologías de Información*. [S.l.: s.n.], 2010. p. 290–295.
- COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. *Sistemas Distribuídos-: Conceitos e Projeto*. [S.l.]: Bookman Editora, 2013.
- CUNHA, M. X. C. da; JUNIOR, M. F. de S.; ALMEIDA, H. O. de. Dificuldades com integração e interoperabilidade de sistemas de informação nas instituições públicas de ensino-um estudo de caso no cefet-al. *Anais do SIMPEP, Bauru, Brasil*, v. 12, p. 1–11, 2005.
- CUNHA, M. X. C. da; JUNIOR, M. F. de S.; DORNELAS, J. S. O uso da arquitetura SOA como estratégia de integração de sistemas de informação em uma instituição pública de ensino. *SEGeT – Simpósio de Excelência em Gestão e Tecnologia*, 2008.
- DEGAN, J. O. C. *Integração de dados corporativos: uma proposta de arquitetura baseada em serviços de dados*. Dissertação (Mestrado) — Unicamp - Universidade Estadual de Campinas, 2005.

DGTI. *Plano Diretor de Tecnologia da Informação 2011/2012*. [S.l.]: Universidade Federal de Lavras, 2011.

FERREIRA, J. M. M. *Introdução ao projecto com sistemas digitais e microcontroladores*. [S.l.]: FEUP Edições, 1998.

FIELDING, R. T. *Architectural styles and the design of network-based software architectures*. Tese (Doutorado) — University of California, Irvine, 2000.

GERHARDT, T. E.; PINTO, J. M.; RIQUINHO, D. L.; ROESE, 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, v. 16, n. Supl 1, p. 1221–1232, 2011.

HARDWARE, G. do. LDAP. *Disponível em:* <<http://www.hardware.com.br/termos/ldap>>. *Acessado em 09/09/2014*, 2005.

HENSLE, B.; BOOTH, C.; CHAPPELLE, D.; MCDANIELS, J.; WILKINS, M.; BENNETT, S. *Oracle Reference Architecture - Service-Oriented Integration, Release 3.0*. 3. ed. [S.l.], 2010.

HEUVEL, W.-J. van den; ZIMMERMANN, O.; LEYMAN, F.; LAGO, P.; SCHIEFERDECKER, I.; ZDUN, U.; AVGERIOU, P. Software service engineering: Tenets and challenges. In: IEEE COMPUTER SOCIETY. *Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems*. [S.l.], 2009. p. 26–33.

HOPPE, G.; WOLFF, B. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. [S.l.]: Addison-Wesley Professional, 2003.

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.

JESSUP, L. M.; VALACICH, J. S. *Information systems today*. [S.l.]: Prentice Hall Professional Technical Reference, 2002.

JORGE, M. H. P. de M.; LAURENTI, R.; GOTLIEB, S. L. D. Avaliação dos sistemas de informação em saúde no Brasil. *Cad. Saude Colet*, v. 18, n. 1, p. 07–18, 2010.

JUNG, C. F. *Metodologia Para Pesquisa & Desenvolvimento: Aplicada a novas tecnologias, produtos e processos*. [S.l.]: Axcel Books, 2004.

LAND, R.; CRNKOVIC, I. Software systems integration and architectural analysis - a case study. In: *Software Maintenance, 2003. ICSM 2003. Proceedings. International Conference on*. [S.l.: s.n.], 2003. p. 338–347.

LAURINDO, F. J. B.; SHIMIZU, T.; CARVALHO, M. M. d.; JR, R. R. O papel da tecnologia da informação (TI) na estratégia das organizações. *Gestão & Produção*, SciELO Brasil, v. 8, n. 2, p. 160–179, 2001.

MARIN, H. de F. Sistemas de informação em saúde: considerações gerais. *Journal of Health Informatics*, v. 2, n. 1, 2010.

MARTINS, V. M. M. *Integração de Sistemas de Informação: Perspectivas, normas e abordagens*. Dissertação (Mestrado) — Universidade do Minho - Guimarães - Portugal, 2005.

MELCHIORI, A. P. Metodologia de pesquisa - guia de estudos. *Centro de Apoio à Educação a Distância da Universidade Federal de Lavras (UFLA)*, 2011.

MUEHLEN, M. Z.; NICKERSON, J. V.; SWENSON, K. D. Developing web services choreography standards—the case of rest vs. soap. *Decision Support Systems*, Elsevier, v. 40, n. 1, p. 9–29, 2005.

MYSQL. Adding a new user-defined function. *Disponível em* <<http://dev.mysql.com/doc/refman/5.1/en/adding-udf.html>>. *Acessado em 03/11/2014.*, 2005.

MYSQL. The FEDERATED Storage Engine. *Disponível em* <<http://dev.mysql.com/doc/refman/5.0/en/federated-storage-engine.html>>. *Acessado em 03/11/2014.*, 2005.

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*, v. 5, n. 3, p. 66–82, 2011.

OASIS, O. Service-oriented architecture. *Disponível em:* <<http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>>. *Acessado em 15/09/2014.*, 2006.

O'BRIEN, J. A. *Sistemas de Informação e as Decisões Gerenciais na Era da Internet*. [S.l.]: São Paulo, 2010.

ORACLE. Java™ API for XML Web Services (JAX-WS). *Disponível em* <<http://docs.oracle.com/javase/7/docs/technotes/guides/xml/jax-ws/>>. *Acessado em 22/10/2014.*, 2011.

PRESSMAN, R. S. *Software engineering: a practitioner's approach*. [S.l.]: McGraw-Hill, 2001.

RAMOS, C. M. Sistemas de informação para a gestão turística. *Revista Encontros Científicos-Tourism & Management Studies*, Escola Superior de Gestão, Hotelaria e Turismo, n. 6, p. 107–116, 2010.

STEEN, M. V.; TANENBAUM, A. *Distributed Systems: Principles and Paradigms*. [S.l.]: Prentice Hall, 2001. 2 p.

TECHNOLOGIES, I. Web services definition. *Disponível em* <<http://www.w3.org/2001/03/WSWS-popa/paper13>>. *Acessado em 21/10/2014.*, 2001.

TURBAN, E.; LEIDNER, D.; MCLEAN, E.; WETHERBE, J. *Tecnologia da Informação para Gestão: Transformando os Negócios na Economia Digital 6ed.* [S.l.]: Bookman, 2004.

UFLA. Novos cursos de engenharia da ufla terão Área básica de ingresso. *Disponível em* <<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.

UFLA, A. Números. *Disponível em:* <<http://www.ufla.br/portal/institucional/sobre/numeros/>>. *Acessado em 24/08/2014.*, 2014.

VALIPOUR, M. H.; AMIRZAFARI, B.; MALEKI, K. N.; DANESHPOUR, N. A Brief Survey of Software Architecture Concepts and Service Oriented Architecture. *JASIS*, 2009.

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, v. 45, n. 1, p. 24–30, 2011.