

**FACULDADES INTEGRADAS DE TAQUARA
CURSO DE SISTEMAS DE INFORMAÇÃO**

**SISCONT:
SISTEMA DE APOIO A GERAÇÃO, TRANSMISSÃO E CONCILIAÇÃO DE DADOS
ATRAVÉS DE ARQUIVOS XML/ASCII**

JULIANO FRANCISCO ANGELI

Taquara

2008

JULIANO FRANCISCO ANGELI

**SISCONT:
SISTEMA DE APOIO A GERAÇÃO, TRANSMISSÃO E CONCILIAÇÃO DE DADOS
ATRAVÉS DE ARQUIVOS XML/ASCII**

Trabalho de Conclusão apresentado ao Curso de Sistemas de Informação das Faculdades Integradas de Taquara, como requisito parcial para obtenção de grau de Bacharel em Sistemas de Informação sob orientação do Prof. Msc. Marcelo Cunha de Azambuja.

Taquara

2008

Dedico este trabalho a Deus o qual, tenho certeza, sempre me supriu de forças até mesmo nos dias mais tempestuosos pelos quais passei.

Também dedico aos meus pais que me ensinaram a lutar sempre, apoiando e incentivando para que estivesse aqui hoje.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me dar forças para seguir em frente e finalizar mais esta etapa da minha vida.

Aos meus pais, familiares e amigos, em especial minha noiva Marisandra. Estes sempre me apoiaram e incentivaram em todos os momentos. Sei que juntamente comigo batalharam para que estivesse onde estou hoje, por isso a todos sou imensamente grato.

A FACCAT, professores e colegas, em especial ao meu orientador Marcelo Cunha de Azambuja, Querte Mehlecke, a galera dos laboratórios, NEO e redes, pela convivência e aprendizado adquirido nestes anos que passamos juntos. Tenho certeza, fiz muitos amigos os quais levarei pelo resto da minha vida.

Enfim, a todos que de alguma forma colaboraram para essa minha conquista o meu muito obrigado.

RESUMO

Este trabalho tem a finalidade de apresentar um sistema de apoio para empresas que tenham a necessidade de realizar trocas de dados através de arquivos, bem como validar e conciliar¹ os arquivos com os dados neles contidos. Normalmente, empresas que terceirizam alguns tipos de serviços possuem tal necessidade, por isso o trabalho apresentado busca dar ênfase para este tipo de negócio. O sistema permite a exportação de arquivos de dados no formato XML e ASCII, os quais serão gerados a partir de configurações e cadastros realizados previamente. Esses arquivos possuem *layout's*² que são configurados pelo administrador, extraindo informações de uma base de dados especificada. O sistema desenvolvido é composto de uma aplicação servidora e outra cliente, as quais se comunicam através de *Web Service*, onde a aplicação cliente pode solicitar arquivos à aplicação servidora para, por exemplo, obter a relação de vendas realizada em um determinado período. A partir de um arquivo recebido no formato XML (com *layout* definido anteriormente pelo administrador do sistema), pode ser realizada uma visualização, em forma de relatório, dos dados nele contidos. Da mesma forma, a partir de dois arquivos (por exemplo, da empresa A e da empresa B), o sistema realiza automaticamente uma conciliação dos dados, exibindo possíveis divergências entre ambos, caso existam.

Palavras-chave: Integração. EAI. Conciliação. Dados. Arquivo XML. Arquivo ASCII. Terceirização.

¹ Conciliar: O termo conciliação será utilizado nesse trabalho para indicar o ato de verificar a igualdade entre dois arquivos de dados. Por exemplo, para verificar a divergência de valores entre a quantidade de vendas informada por uma filial e a informação que a matriz tem por verdadeira. O sistema irá verificar, em normalmente uma longa lista de dados (fornecidos pela empresa A), se estes estão idênticos (valores, datas, etc.) a outra longa lista de dados (fornecidos pela empresa B).

² Layout: Será tratado no trabalho como a disposição e exibição de dados em uma seqüência pré-determinada (*template*).

LISTA DE FIGURAS

Figura 1: Modelo do sistema proposto - SISCONT	18
Figura 2 - Origem dos fatores que convergem para a Integração de Aplicações	22
Figura 3: Modelo atual de conciliação	38
Figura 4: Exemplo de conciliação dos serviços hipotéticos solicitados pela Empresa A (terceirizada) a Empresa B (terceirizante)	40
Figura 5: Diagrama caso de uso - aplicação servidora	50
Figura 6: Diagrama caso de uso - aplicação cliente	52
Figura 7: Diagrama de seqüência – realizar login	54
Figura 8: Diagrama de seqüência – cadastrar SQL	54
Figura 9: Diagrama de seqüência – cadastrar seção	55
Figura 10: Diagrama de seqüência – cadastrar arquivo	55
Figura 11: Diagrama de seqüência – solicitar geração de arquivo (XML, ASCII)	55
Figura 12: Diagrama de seqüência – cadastrar usuário para <i>Web Service</i>	56
Figura 13: Diagrama de seqüência – cadastrar arquivo do servidor para <i>Web Service</i>	56
Figura 14: Diagrama de seqüência – solicitar arquivo XML do servidor via <i>Web Service</i>	57
Figura 15: Diagrama de seqüência – solicitar leitura de arquivo XML	57
Figura 16: Diagrama de seqüência – conciliar arquivo XML	58
Figura 17: Diagrama de classe	58
Figura 18: Modelo ER	59
Figura 19: <i>Layout</i> padrão do arquivo de dados no formato XML	69
Figura 20: Validador XML Schema (XSD)	71
Figura 21: <i>Layout</i> padrão do arquivo de dados no formato TXT	72
Figura 22: Telas de login	73
Figura 23: Telas de login	73
Figura 24: Cadastro de SQL	74
Figura 25: Cadastro de seção	75
Figura 26: Cadastro de arquivo	76
Figura 27: Gerar arquivo	77
Figura 28: Cadastra usuário para <i>Web Service</i>	78
Figura 29: Cadastro do arquivo disponível pela aplicação servidora	79
Figura 30: Buscar arquivo da aplicação servidora	80
Figura 31: Ler arquivo XML	80
Figura 32: Exemplos de operações que podem ser executadas sobre os dados na conciliação	81
Figura 33: Exemplos de operação executada sobre os dados exibindo a conciliação	82
Figura 34: Exemplos de consulta SQL executada de forma <i>on-line</i> pelo administrador	82
Figura 35: Resultado de consulta SQL executada de forma <i>on-line</i> pelo administrador	83

LISTA DE QUADROS

Quadro 1: Nível de complexidade das interfaces	26
--	----

LISTA DE SIGLAS

A2A - Application to Application
ADODB - Database Abstraction Library
AJAX - Asynchronous Javascript And XML
API - Application Programming Interface
ASCII - American Standard Code for Information Interchange
B2B - Bussines to Bussines
BPM - Business Process Management
COM - Component Object Model
CORBA - Common Object Request Broker Architecture
CRM - Customer Relationship Management
CSS - Cascading Style Sheets
DAO - Data Access Object
DOM - Document Object Model
EAI - Enterprise Application Integration
HTML - HyperText Markup Language
HTTP - Hypertext Transfer Protocol
JDBC - Java Database Connectivity
MOM - Message Oriented Middleware
MVC - Model-View-Controller
ODBC - Open Database Connectivity
OO - Object Oriented
ORB - Object Request Brokers
PHP - Hypertext Preprocessor
PRM - Partner Relationship Management
RPC - Remote Procedure Call
SCM - Supply Chain Management
SOAP - Simple Object Access Protocol
SQL - Structured Query Language
SGBD - Sistema Gerenciador de Banco de Dados
SI - Sistemas de Informação
SOAP - Simple Object Access Protocol

TI - Tecnologia de Informação

TMP - Trading Partner Management

UML - Unified Modeling Language

VO - Value Object

WS - Web Services

WSDL - Web Services Description Language

XML - Extensible Markup Language

XSD - XML Schema Definition

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Justificativa	16
1.2	Objetivo	17
1.3	Organização do Trabalho	19
2	REVISÃO DA LITERATURA.....	20
2.1	Integração entre Sistemas - EAI.....	20
2.1.1	Fatores ou necessidades que levam a integração	21
2.1.2	Evolução e abordagens de integração.....	22
2.1.3	Escopo da integração	24
2.1.3.1	Avaliando o escopo do projeto de integração	24
2.1.4	Tipos de integração (por camadas).....	27
2.1.4.1	Integração das plataformas operacionais	27
2.1.4.2	Nível de integração por dados.....	27
2.1.4.3	Nível de integração pela interface da aplicação	28
2.1.4.4	Nível de integração por métodos.....	28
2.1.4.5	Nível de interface do usuário	28
2.1.5	Tecnologias de integração (<i>middleware</i>).....	29
2.1.5.1	Orientados a dados	31
2.1.5.2	Orientados a mensagens (MOM - <i>Message Oriented Middleware</i>)	31
2.1.5.3	Chamadas de procedimentos remotos (RPC – <i>Remote Procedure Call</i>)	32
2.1.5.4	Objetos distribuídos (ORB´s - <i>Object Request Brokers</i>).....	32
2.1.5.5	Monitores de processamento de transações (<i>TP Monitors</i>)	32
2.1.5.6	Servidores de aplicação (<i>Application Servers</i>)	33
2.2	Terceirização (<i>Outsourcing</i>).....	34
3	PROBLEMA	37
3.1	Conciliação	39
3.2	A Empresa	41
3.3	Problemas Tratados	43
3.4	Visão Macro das Funcionalidades do Sistema	44
3.4.1	Delimitações da versão iniciabl do sistema	44
4	METODOLOGIA.....	46
4.1	Análise de Requisitos.....	47

4.1.1	Descrição dos requisitos	47
4.1.1.1	Requisitos comuns à aplicação cliente e servidora	47
4.1.1.2	Requisitos específicos da aplicação servidora	48
4.1.1.3	Requisitos específicos da aplicação cliente	48
4.2	Diagramas UML	49
4.2.1	Diagrama de casos de uso	49
4.2.2	Diagramas de seqüência	54
4.2.3	Diagrama de classe.....	58
4.3	Modelagem Entidade Relacional (ER)	59
4.4	Padrões de Projeto (<i>Patterns</i>)	59
4.4.1	Padrão MVC	60
4.4.2	Padrão DAO.....	60
4.5	Testes e Homologação	61
5	TECNOLOGIAS.....	62
5.1	Apache	62
5.2	PHP 5 + OO e Suporte a Web Services	63
5.3	DHTML (HTML, CSS, JavaScript e DOM)	64
5.4	AJAX	65
5.5	XML.....	65
5.6	Smarty	66
5.7	ADODB.....	66
6	RESULTADOS	68
6.1	Layout Padrão para Arquivos	68
6.2	Funcionalidades Comuns às Aplicações	72
6.2.1	Tela de login e tela inicial das aplicações.....	73
6.2.2	Cadastro de SQL	73
6.2.3	Cadastro de seção	74
6.2.4	Cadastro de arquivo.....	75
6.2.5	Gerar Arquivo	76
6.3	Funcionalidades Específicas da Aplicação Servidora	77
6.3.1	Cadastro usuário para Web Service.....	77
6.4	Funcionalidades Específicas da Aplicação Cliente.....	78
6.4.1	Cadastro do arquivo disponível pela aplicação servidora	78
6.4.2	Buscar arquivo da aplicação servidora	79
6.4.3	Ler arquivo XML	80

6.4.4	Conciliar arquivo XML	81
7	CONCLUSÃO.....	84
	REFERÊNCIAS.....	86

1 INTRODUÇÃO

Atualmente existem muitas organizações e profissionais liberais que terceirizam serviços (*outsourcing*³), forma de trabalho que vem se tornando cada vez mais cotidiana e corriqueira com o passar dos anos. Inicialmente adotada por países estrangeiros, nos dias de hoje incorporada também no Brasil, é vista cada vez mais como uma forte tendência segundo Gonçalves (2005). De acordo com a Info Corporate (2008) e Baguete (2008), no ano de 2008, só no mercado interno de Tecnologia da Informação (TI), terceirizações devem representar cerca de 32% de todos os investimentos na área, o que representa R\$ 15,2 bilhões de reais.

Nesse contexto de trabalho compartilhado entre duas ou mais empresas, a necessidade de troca de informações é muito grande. Esses dados são transmitidos entre sistemas que possuem uma certa integração entre si conforme explica Sordi e Marinho (2006). Mas, é importante salientar, são bases de dados com propriedades independentes, e por esse motivo é necessário o desenvolvimento de alguma forma de integração e transferência dos dados. Organizações que terceirizam serviços tendem a trocar dados que representam os negócios entre duas ou mais organizações parceiras, esses dados são exteriorizados e armazenados de forma que consigam realizar a leitura a partir de seus sistemas.

Cassarro (1999) afirma que sistemas são alimentados com dados, onde posteriormente esses dados são processados e viram informações. “Dado é qualquer elemento identificado em sua forma bruta que, por si só, não conduz a uma compreensão de determinado fato ou situação.” (OLIVEIRA 1999, p. 36). Sistemas que processam dados, e posteriormente os transformam em informações, através de relatórios, listagens, gráficos e outros, são tratados então como Sistemas de Informação (SI). Cassarro (1999, p. 34,35) diz ainda que é “[...] de consenso geral que a informação é ao mesmo tempo matéria-prima e produto acabado da atividade de sistemas [...]”, complementa ainda que informações bem estruturadas ajudam empresas a se tornarem mais dinâmicas.

É importante entender a definição de informação e ou informações, Barreto (1996, p.2) conceitua como sendo: “Estruturas significantes com a competência de gerar conhecimento no indivíduo, em seu grupo, ou a sociedade.”. Já Oliveira (1999, p. 36), em um contexto organizacional, diz:

³ *Outsourcing*: Termo em inglês para definição de terceirização, onde ‘out’ significa fora e ‘source’ ou ‘sourcing’ significa fonte.

A informação é o produto da análise dos dados existentes na empresa, devidamente registrados, classificados, organizados, relacionados e interpretados em um determinado contexto, para transferir conhecimento e permitir a tomada de decisão de forma otimizada.

Informações pertinentes e bem elaboradas podem então apoiar a tomada de decisão, aumentando consideravelmente as chances de acerto nas decisões de quem a detêm, sendo consideradas como matéria-prima por Bogui e Shitsuka (2002).

A proliferação dos SI, alavancada pela substituição de sistemas legados a partir do “bug” do milênio, buscando a prevenção de possíveis problemas, fez surgir uma forte demanda de troca de informações entre alguns deles. Na sua grande maioria, realizadas através da troca de arquivos ou compartilhamento de dados, tanto entre SI internos das organizações A2A (*Application to Application*) quanto entre SI de organizações parceiras B2B (*Bussines to Bussines*). Com essa explosão de SI, chegou um momento em que as organizações precisavam não somente de novos sistemas, mas sim realizar certas integrações entre muitos deles. Nas últimas décadas a integração entre SI tem recebido grande atenção de pesquisadores e organizações, a ponto de tornar-se essencial para o bom desempenho de muitas das organizações, conforme Sordi e Marinho (2006).

As tecnologias que realizam esse papel de integração entre SI são conhecidas como EAI (*Enterprise Application Integration*), ou, Integração de Aplicações Empresarias.

O objetivo da estratégia EAI é integrar e criar um fluxo perfeito dos processos de negócio entre diferentes aplicações e unidades de negócio e ao mesmo tempo permitir a colaboradores, tomadores de decisão, e parceiros de negócio prontamente acessar as informações da empresa e dos clientes, não importando onde as mesmas estejam localizadas. (STOKES, 2001, p. 1)

Charlesworth e Jones (2003, p. 2) dizem que “A promessa de EAI é prover um ponto central capaz de integrar aplicações, processos e pessoas.”, Fernandez (2004, p. 7) complementa que “O papel das tecnologias focadas em EAI está diretamente relacionado à supressão de diferenças entre dados de diferentes fontes e padrões tecnológicos.”.

EAI possibilita a integração a partir de diferentes tipos ou camadas de integração, cada tipo tem as suas vantagens e desvantagens variando em custo, recurso e implementação. Dentro de EAI existem mecanismos que possibilitam que uma entidade (aplicação ou banco de dados) se comunique com outras, esses mecanismos são chamados de *middleware*. O

propósito de um *middleware* é fazer com que as diferenças tecnológicas sejam transparentes na comunicação entre as aplicações, minimizando as dificuldades de programação e viabilizando a troca de informações (FERNANDEZ 2004). Fernandez ainda cita os tipos de integração e *middleware*, que serão detalhados no capítulo 2 desse trabalho, como:

Tipos de integração ou camadas

- Plataformas Operacionais;
- Nível de Integração por dados;
- Nível de Integração pela Interface da Aplicação;
- Nível de Integração por Métodos;
- Nível de Interface do Usuário.

Tipos de *middleware*

- Orientados a Dados;
- Orientados a Mensagens (MOM - *Message Oriented Middleware*);
- Chamadas de Procedimentos Remotos (RPC – *Remote Procedure Call*);
- Objetos Distribuídos (ORB's - *Object Request Brokers*);
- Monitores de Processamento de Transações (*TP Monitor*);
- Servidores de Aplicação (*Application Servers*).

Dentre os tipos de *middleware* apresentados, existe ainda os *Web Services* (WS), que apesar de não ser um componente de integração propriamente dito, possibilita disponibilizar serviços de forma padronizada através de um conjunto de padrões e tecnologias: descritores WSDL (*Web Services Description Language*), linguagem de marcação através de tags XML (*Extensible Markup Language*), protocolo de comunicação de mensagens SOAP (*Simple Object Access Protocol*) tudo disponibilizado através da *Web*⁴ sob protocolo HTTP (*Hypertext Transfer Protocol*) (WSW3C, 2008) e (FERNANDEZ, 2004).

Os WS, em razão da tecnologia baseada em serviços, possibilitam um menor impacto e maior flexibilidade para mudanças tecnológicas ou de negócios no ambiente (FERNANDEZ, 2004). Além da flexibilidade, Charlesworth e Jones (2003) acrescentam que soluções baseadas em WS tendem a ter um menor custo financeiro.

⁴ Web: Denominação da internet (World Wide Web).

As organizações devem escolher técnicas e tecnologias de integração que melhor se adequar a cada demanda, ou seja, não existe uma regra, uma vez que cada técnica e ou tecnologia possui vantagens e desvantagens umas sobre as outras.

Não há uma técnica de integração que possa ser considerada a melhor para uma organização, cada uma delas possui vantagens e desvantagens. O que é viável é a seleção da técnica mais apropriada para cada nova necessidade de integração. As aplicações podem integrar-se, empregando várias técnicas de integração, de forma a obter vantagens daquela que melhor atenda a cada nova demanda. Assim, a camada de integração entre SI pode ser entendida como um ambiente híbrido composto por diferentes técnicas de integração. (HOHPE e WOOLF, 2003 *apud* SORDI e MARINHO, 2006, p. 6)

Estando o leitor familiarizado com os conceitos, tendências e tecnologias apresentadas, esse trabalho vai abordar nos capítulos seguintes alguns pontos relacionados a integrações de SI (EAI) e a necessidade específica de uma empresa de TI (um estudo de caso) em realizar a integração de alguns processos do seu sistema com o sistema de parceiros e clientes. A empresa utilizada como estudo de caso nesse trabalho trata-se de uma empresa provedora de serviços de telecomunicações que oferece serviços como autorização de transações eletrônicas, vendas de recarga de celular pré-paga, entre outros. Baseado na necessidade dessa empresa, esse trabalho irá apresentar o desenvolvimento de um sistema *Web* denominado **SISCONT**, onde foram implementados alguns mecanismos (*middleware*) para comunicação e integração entre sistemas através de uma aplicação servidora e uma cliente.

O **SISCONT** vai apoiar a troca de arquivos XML/ASCII entre as empresas. Os dados que irão compor os arquivos serão extraídos de consultas ao banco de dados das empresas envolvidas e transmitidos entre as aplicações através de chamadas de serviços via *Web Services*.

Além da geração e transmissão dos arquivos de dados, o trabalho explica a necessidade de se realizar uma validação e ou comparação dos dados contidos nos arquivos. Quem recebe os arquivos realiza uma comparação para encontrar possíveis divergências entre eles. O processo de validação e ou comparação será tratado no decorrer do trabalho como uma conciliação entre dados providos de bases de dados distintas, conciliação esta também suportada pelo **SISCONT**. O termo conciliação vai indicar o ato de verificar a igualdade entre dois arquivos de dados. Por exemplo, para verificar a divergência de valores entre a

quantidade de vendas informada por uma filial e a que a matriz tem por verdadeira, além de todos os dados que compõem cada venda.

Devido ao fato do **SISCONT** ser baseado na necessidade de uma empresa de prestação de serviços, e também motivado pelo fato de que nesse ramo empresarial a terceirização é um mercado que cresceu muito e que tende a crescer ainda mais, como apontado por vários autores e meios de comunicação, tais como Giosa (2003), Gonçalves (2005), EBC (2008), Administradores (2007), Info Corporate (2008), Baguete (2008) e outros, esse trabalho vai contextualizar em parte a terceirização (*outsourcing*) como tendência, possibilitando ter uma visão de evolução futura para o sistema, uma vez que esse mercado é promissor.

1.1 Justificativa

Considerando a proliferação de SI dentro das organizações nas últimas décadas e a necessidade de troca de dados entre esses sistemas através de chamadas (*call interface*), mensagens (*messaging*) ou transferência entre arquivos/compartilhamento de dados (*data access/file transfer*), sendo esse último comumente o mais usado de acordo com Sordi e Marinho (2006), tanto entre SI internos das organizações A2A quanto de organizações parceiras B2B, alinhado a necessidade de se realizar uma conciliação desses dados para, por exemplo, conferir quantidades, valores monetários envolvidos ou qualquer outra informação necessária, sistemas que consigam trabalhar nesse tipo de mercado é uma necessidade que se verificou através das pesquisas realizadas durante esse trabalho. Esse tipo de necessidade foi constatado na empresa que serviu de estudo de caso para o desenvolvimento do sistema desenvolvido, representando a necessidade de muitas outras empresas. Conforme já referenciado, o desenvolvimento de um sistema usando tecnologias de integração EAI (*Enterprise Application Integration*) para gerar arquivos de dados, a partir de bases de dados distintas, possibilitando a transmissão dos arquivos entre as organizações envolvidas, confrontando e detectando possíveis dados divergentes, eliminando possíveis erros no processo e auxiliando a tomada de decisão é de extrema importância.

O processo de conciliação dos dados, na maioria das vezes, é feito visualmente através de relatórios. Havendo divergências, se torna um processo manual, relativamente demorado para encontrá-las e propício a erros, assim atrasando o processo como um todo.

Algumas empresas realizam a conciliação a partir de arquivos que possuem um *layout* conhecido sendo lido através de programas especializados por quem o recebe, esse é o caso da empresa do estudo de caso. Esses arquivos podem ser gerados por áreas ou pessoas que possuem habilidades específicas, na maioria das vezes pela área de TI, tornando o processo de geração dependente e centralizado. Em outro caso, os arquivos podem ser gerados por programas que são agendados para inicializar o processo em horários pré-determinados. Como o processo não é *on-line*⁵, os arquivos não podem ser gerados a qualquer momento, limitando o trabalho em determinados casos. Já para quem está recebendo os arquivos, será necessário ter outro programa especializado para ler e conciliar os dados gerados no *layout* pré-determinado. A necessidade de desenvolvimento de programas especializados pelas organizações para geração, leitura e conciliação de dados impede que o processo seja mais dinâmico, pois cada alteração no programa gera um custo adicional, além de necessitar a demanda de funcionários especializados na área de desenvolvimento de software, o que nem sempre existe em organizações que não sejam do segmento de TI.

1.2 Objetivo

O trabalho apresentado, numa visão macro, propõe o desenvolvimento de um sistema *Web* utilizando tecnologias de integração EAI (*Enterprise Application Integration*), denominado **SISCONT**. A função principal do SISCONT é permitir a integração entre duas bases de dados de empresas distintas, possibilitando:

1. a extração dos dados;
2. validação dos dados extraídos;
3. exibição de informações na forma de relatório;
4. exibição de informações que listem suas diferenças;
5. chamadas de serviços, através de *Web Services*, entre uma aplicação cliente e servidora.

No contexto desse trabalho, os dados a serem tratados são, tipicamente, relatórios de venda e prestação de serviços. Por exemplo, a “Empresa A” informa que realizou uma quantia

⁵ on-line: Disponibilizado em tempo real (agora)

x de serviços técnicos para a “Empresa B”; essa precisa verificar se a informação corresponde ao que tem registrado no seu próprio sistema de controle de serviços técnicos utilizados da “Empresa A”. Empresas que tem um fluxo contínuo de dados procedentes de negócios com terceiros tendem a dispensar esforços consideráveis para avaliar a veracidade desses dados. Muitas delas desenvolvem sistemas internos para tratar tal situação, mas normalmente esses tratam paliativamente o problema, atendendo somente uma das pontas envolvidas, não visando o processo do negócio como um todo. O SISCONT procura dar suporte visando melhorar esse processo no todo. A Figura 1 apresenta o modelo proposto pelo SISCONT:

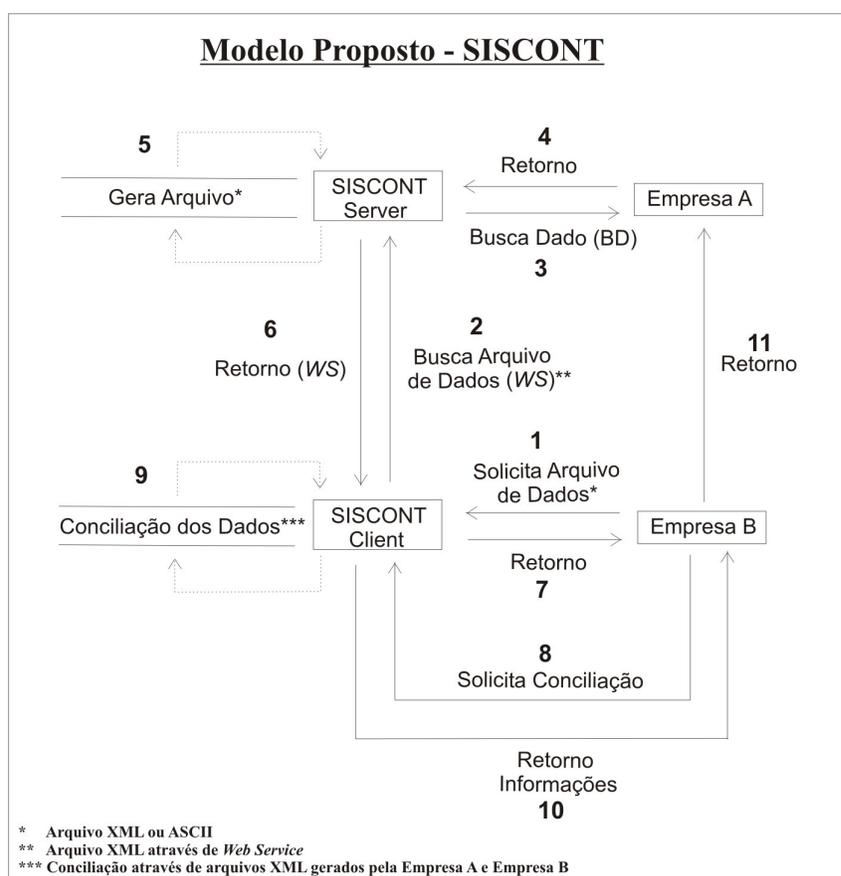


Figura 1: Modelo do sistema proposto - SISCONT

O sistema permite o cadastro do *layout* de um determinado arquivo de dados possibilitando a geração e ou transmissão *on-line* dos arquivos. As requisições dos arquivos ocorrem sem a necessidade de um agendamento prévio entre as partes cliente e servidor. Permite ao cliente a realização da leitura e exibição dos dados em forma de relatório. De acordo com a necessidade, os arquivos poderão ser conciliados e os resultados dessa

validação mostrados na forma de relatórios. Basicamente a conciliação vai exibir uma mensagem de resultado positivo ou exibir as informações dos dados divergentes que auxiliará a tomada de decisão.

É importante frisar que o sistema foi desenvolvido baseado nas necessidades de uma empresa provedora de rede que oferece serviços como autorização de transações eletrônicas, recarga de celular pré-paga, entre outros, porém qualquer pessoa ou empresa que tenha a necessidade de gerar arquivos de dados a partir de um banco de dados, posteriormente disponibilizá-los a um terceiro e conciliá-lo poderá usar o **SISCONT**.

1.3 Organização do Trabalho

O conteúdo do trabalho está distribuído em sete capítulos:

- a) o Capítulo 1 apresentou uma breve introdução, com justificativa e o objetivo que levou ao desenvolvimento desse trabalho e sistema;
- b) o Capítulo 2 traz uma revisão da literatura com definições e conceitos de integração de aplicações empresariais (EAI) e terceirização (*outsourcing*);
- c) o Capítulo 3 expõe o problema em si, as visões macros das funcionalidades e delimitações da versão inicial do sistema;
- d) o Capítulo 4 apresenta metodologia, levantamento de requisitos, padrões utilizados e outros;
- e) o Capítulo 5 trata sobre as tecnologias utilizadas;
- f) o Capítulo 6 aborda os resultados com telas do sistema;
- g) as conclusões são apresentadas no Capítulo 7.

2 REVISÃO DA LITERATURA

Nas subseções seguintes, serão apresentados os conceitos que tem uma correlação com o sistema desenvolvido. Esses formaram a base conceitual para o desenvolvimento do **SISCONT**.

2.1 Integração entre Sistemas - EAI

Aplicações responsáveis por integrar ou interligar SI são conhecidas como EAI (*Enterprise Application Integration*), ou Integração de Aplicações Empresariais. De acordo com Stokes (2001), EAI busca atuar entre SI diferentes, realizando a integração e ajudando no fluxo de informações dos processos inerentes ao negócio da organização. Assim, possibilita que pessoas vinculadas à organização tenham acesso as informações do negócio, não importando onde as mesmas estejam localizadas. Fernandez (2004, p. 7) complementa que “O papel das tecnologias focadas em EAI está diretamente relacionado à supressão de diferenças entre dados de diferentes fontes e padrões tecnológicos.”.

Nas últimas décadas, houve uma proliferação de SI nas organizações, distribuídos em um ambiente heterogêneo de *hardware* e *software*, que colaborou para o surgimento da integração entre sistemas. Um dos principais fatores que impulsionou a demanda por EAI, foi a migração de aplicações monolíticas formando um conjunto de aplicações legadas, onde “[...] esse novo conjunto têm uma multiplicidade de aplicações, cada uma com suas ‘ilhas de informação’ dentro da infraestrutura da tecnologia de informação das corporações.” (LUBLINSK, 2001 *apud* FERNANDEZ, 2004, p. 1).

De acordo com Sordi e Marinho (2006), a integração entre SI passou a ser considerada pelos pesquisadores e praticantes (organizações), de uma forma mais abrangente, apenas no início deste século. Administradores contemporâneos, tem muitos dos seus desafios, ligados a integração de SI ou uso errado das tecnologias de integração. Cummins (2003, *apud* Sordi e Marinho, 2006, p.2) cita alguns problemas administrativos relacionados da integração entre SI:

- perda de competitividade em função de ser incapaz de reduzir o tempo dos processos em decorrência de limitações das integrações empregadas, por exemplo, pelo uso de conexões assíncronas que ocorrem através de transmissão de pacotes de dados em períodos de tempo pré-estabelecidos;
- restringe a capacidade da organização em se habilitar como fornecedor ou parceiro estratégico de redes e ambientes colaborativos que exijam das entidades participantes o domínio de modernas tecnologias de integração;
- eleva custos e aumenta o risco de exposição a erros em decorrência do trabalho humano intensivo na re-digitação de dados de uma base de dados ou SI para outra base ou sistema;
- lentidão da organização em identificar e tratar eventos do negócio possíveis de serem percebidos pela comunicação de ocorrências registradas em outros SI, limitando a capacidade de inovação e pró-atividade da empresa;
- dificulta a evolução e aprimoramento dos processos de negócios em função de se evitar a alteração das tradicionais integrações de sistemas existentes ao longo do processo, postura justificada pelos altos custos e pelos receios de se gerar novos problemas quando da alteração destes meios legados de integração.

2.1.1 Fatores ou necessidades que levam a integração

Diversos fatores podem causar a necessidade de integração entre SI. Os fatores estão relacionados, direta ou indiretamente, com o negócio e os sistemas baseados em tecnologia. Esses fatores podem afetar ambos, causando diversas mudanças no ambiente. Criando um universo heterogêneo, onde estão aplicações legadas juntamente com aplicações de última tecnologia, todas geograficamente distribuídas. Esse universo heterogêneo requer mecanismos que interajam entre as aplicações, realizando a integração. Fatores que contribuem a integração de SI citados por Fernandez (2004):

a) influência da ordem cronológica dos acontecimentos

- diferentes épocas para instalação ou atualização de aplicações;
- uso de plataformas operacionais (S.O. – Sistemas Operacionais e SGBD – Sistema Gerenciador de Banco de Dados) diferentes das usadas anteriormente.

b) mudanças na área de negócio

- fusões, aquisição de outras corporações e mudanças no negócio.

c) carência de planejamento com foco amplo em integração

- aplicações não contemplam ou não são desenvolvidas visando integrações.

d) características ofertadas pelo mercado

- muitos fornecedores e tecnologias, quantidade de aplicações quase que na mesma proporção;

- não existe um padrão de integração consolidado no mercado de *software*.

A Figura 2 apresenta os diversos fatores que representam a necessidade de integração das aplicações e suas respectivas proporcionalidades.

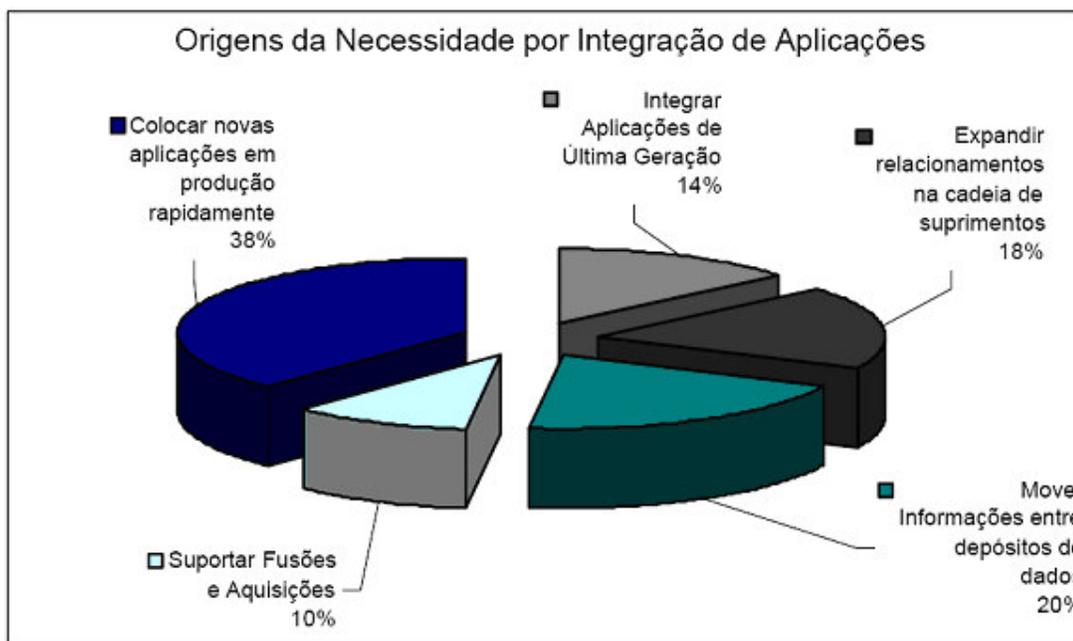


Figura 2 - Origem dos fatores que convergem para a Integração de Aplicações

Fonte: AberdeenGroup, Outubro de 1998 [Linthicum, 2000] adaptado de Fernandez (2004, p. 6)

Tendo apresentado os fatores que levam a necessidade de integração entre SI e sabendo que a evolução e crescimento das organizações tendem a deixar ambientes de sistemas cada vez mais heterogêneos, é natural o crescimento na demanda de aplicações específicas que atendam a integração de SI.

2.1.2 Evolução e abordagens de integração

Existem diversos recursos tecnológicos para integração de SI. De acordo com Sordi e Marinho (2006, p. 7), isso se deve principalmente pelo fato de existir “[...] diversos locais ou pontos do ambiente computacional onde podem ocorrer as ações de integração [...]”, fazendo

com que sejam requeridos certos recursos (pessoas, ferramentas e procedimentos específicos) para que o ambiente se torne propício a integrações.

Os ambientes específicos de integração estão evoluindo. Para evidenciar essa evolução Gartner (2003 *apud* Sordi e Marinho, 2006, p. 7 e 8) separa e conceitua a integração entre SI em 2 abordagens:

- 1) abordagem tradicional: Cada nova demanda de integração é feita pontualmente, como se fosse um novo SI, onde o analista analisa, especifica e gerencia as mudanças para realizar a integração solicitada para aquele caso. Essa não é a melhor situação, porque cria um forte vínculo com o sistema que está na outra ponta, denominando-se uma “perpetuação do legado”. Para medir a entrada de interfaces, apresenta a seguinte fórmula matemática: $I = N * (N - I)$, onde I representa o número total de interfaces e N o número de SI (PURSCHMANN e ALT, 2001 *apud* SORDI e MARINHO, 2006, p. 7). A fórmula representa o pior caso, onde todos se integram com todos.
- 2) abordagem sistêmica: A demanda de integração é analisada por arquitetos de integração, da equipe de integração, que avaliam qual forma é mais apropriada para realizá-la. Numa visão geral, a idéia é tentar utilizar e ou desenvolver métodos de integração que atendam a grande maioria dos SI. Que satisfaçam as necessidades exigidas para a integração, custando o mínimo de esforço possível, de forma que possibilite mudanças futuras não “engessando” o sistema. Flexibilizando a entrada de novas integrações, atuando como um componente estratégico ao desempenho da organização. Abaixo estão detalhados alguns dos principais benefícios da abordagem sistêmica para integração entre SI citados pelo autor:

- a integração dos SI reduz o re-trabalho e favorece a automação de atividades. Redigitação de dados, de um SI para outro, é uma das atividades que é eliminada quando da integração entre sistemas. Ambos passam a acessar o mesmo dado ou transportam dados de um sistema para outro sem esforços humanos;
- o recebimento de dados num determinado SI pode representar um evento de negócio, a passagem direta e íntegra de um sistema para outro, amplia a oportunidade de se identificar e monitorar eventos de negócio. Quanto maior a abrangência dos eventos de negócio tratados, menores serão as exceções a serem tratadas pelos funcionários bem como os custos para operação do processo;
- cresce a capacidade da empresa em adaptar seus processos de negócios, que, tradicionalmente, envolvem grandes números de SI, tanto internos quanto externos à empresa. O ambiente de integração permite alterar de forma rápida, segura e com menores custos, os SI utilizados para implementação de um novo processo de negócio;
- torna viável a expansão da empresa de forma consistente, por exemplo, ao permitir a empresa consolidar amplos conjuntos de SI decorrentes de fusões e aquisições de outras empresas;

- viabiliza novos modelos de negócios, pela facilidade de incorporar e integrar novos SI, por exemplo, facilitando a integração dos diversos softwares necessários à empresa que deseja implementar uma estratégia de gestão de relacionamento;
- maior agilidade e capacidade da empresa em atender as regulamentações do mercado, evitando expor a empresa à penalidades pela não observância de novas regulamentações;
- habilita a empresa para gestão de seus processos de negócio, ao prover-lhe facilidades para medição, automação, revisão, planejamento e evolução do processo, por intermédio da integração dos diversos softwares utilizados para operação e gerenciamento do processo.

2.1.3 Escopo da integração

As integrações devem ser analisadas de uma forma geral dentro do contexto onde os processos problemáticos estão inseridos, não de forma isolada. Segundo Fernandez (2004), quando a integração é vista somente num âmbito técnico, de forma isolada, é comum achar que a solução parece ser extremamente fácil, por muitas vezes até óbvia, assim gerando soluções paliativas. Essa avaliação pontual pode acarretar em maus investimentos gerando insatisfação na solução adotada.

2.1.3.1 Avaliando o escopo do projeto de integração

Fernandez (2004) diz ser difícil estimar os recursos, tanto financeiros quanto de tempo, envolvidos nos projetos de integração de SI devido ao desconhecimento do conjunto das partes a serem integradas. Isso porque, na maioria das vezes, os SI não são desenvolvidos para suportar integrações. Outro fator é a escassez, em muitos casos a inexistência, de documentação ou domínio da sua especificação. Essa análise inicial compreendendo claramente ou, pelo menos, assertivamente o tamanho do escopo do projeto é essencial para o sucesso do mesmo. Levine (2002 *apud* Fernandez, 2004, p. 15 e 16) aponta ainda algumas etapas para orientar na definição de estimativas de recursos necessários ao escopo do projeto:

“[...]”
 - criar uma lista das interfaces;
 - analisar as interfaces;
 - estimar o esforço a ser despendido baseado na complexidade da interface;
 - resumir todas as estimativas;
 - revisar as estimativas e ajustar; [...]”

São criadas listas de interfaces para o projeto proposto, onde são agrupadas por categorias e subcategorias. Interfaces podem ser classificadas da seguinte forma:

- **Migração dos Dados (*Data Migration*):** Interfaces que irão realizar a carga dos dados da bases antigas para a nova. Divide-se nas seguintes subcategorias:
 - Migração de Dados Mestres (*Master Data Migration*): Migração de dados “estáticos”. Exemplo: cadastros de clientes, etc;
 - Migração de Dados Transacionais (*Transaction Data Migration*): Migração de dados que envolvem transações do negócio em si. Exemplo: pedidos, vendas, etc.
- **Ponto de Integração (*Integration Points*):** Interfaces que devem ficar em operação após implantação (Migração dos Dados). Serão responsáveis por manter o fluxo das transações fazendo com que aconteça a integração. Divide-se nas seguintes subcategorias:
 - Pontos Permanentes de Integração (*Permanent Integration Points*): interfaces que ficarão ativas permanentemente no ambiente de produção;
 - Integração Temporária (*Temporary Integration*): são pontos necessários à integração de etapas do projeto. Esses pontos são eliminados conforme forem concluídos os projetos;
 - Múltiplos Pontos Centrais de Integração (*Multiple Master Integration Points*): Interfaces que realizam de algum modo cópia, réplica, de dados que devem ser constantemente sincronizados. Essas integrações podem ser derivadas da Migração de Dados (*Data Migration*).

A complexidade das interfaces são analisadas através de alguns fatores. Esses fatores, segundo Levine (2002 *apud* Fernandez, 2004, p. 18) se classificam em alta, média ou baixa complexidade conforme o quadro a seguir:

Fator a	Métrica	Nível de Complexidade		
		Baixo	Médio	Alto
Mapeamento dos dados entre fontes e destinos	Quantidade de campos	0 até 50	51 até 100	Mais que 100
	Estrutura de dados	<i>single flat format</i>	<i>multiple flat format</i>	<i>hierarchical</i>
	Conversão de dados	Mapeamento campo-a-campo com conversão simples	Combinação de campos com cálculos simples	Múltiplos cálculos com dados derivados
	Fluxo de dados	Entrada única para saída única	Entrada única para múltiplas saídas e vice-versa	Múltiplas entradas para múltiplas saídas
Disponibilidade de metadados e gerador de formato de dados	Capacidade de geração pelo <i>software</i> envolvido	Geração automática	Construção manual para mapeamento de dados de baixa complexidade	Construção manual para mapeamento de dados de média e alta complexidade
Interfaces do legado e de novas aplicações	Nível de mudanças exigidas	Configuração ou pequenos ajustes	Necessita de algumas alterações	Precisa ser criada ou extensamente alterada
	Influência do ciclo de atualização	<i>Batch</i>	<i>near-time e event-driven</i>	<i>real-time</i>
Sistema operacional	Necessidade de configuração	Modificação da configuração	Criação de novas configurações	Requisitos especiais
	Influência do ciclo de atualização	<i>Batch</i>	<i>near-time e event-driven</i>	<i>real-time</i>
Controle de erros	Necessidade de informação e interferência no processo	Parar a integração após o erro e editar o item de dados	Indicar a regra que foi violada e se possível prosseguir com o processamento	Realizar checagens cruzando os campos, identificar múltiplas regras violadas e sugerir modificações

Quadro 1: Nível de complexidade das interfaces

Fonte: Adaptada de Fernandez (2004, p. 18)

Depois da análise, uma revisão deve ser feita visando ajustar eventuais super ou sub-estimativas feitas no levantamento de todas as interfaces.

2.1.4 Tipos de integração (por camadas)

A integração pode ser realizada por camadas que oferecem diferenciadas formas, recursos e custos de implementação. Essas serão apresentadas nas subseções seguintes.

2.1.4.1 Integração das plataformas operacionais

Visa manter a interoperabilidade entre diversos sistemas, principalmente entre sistemas antigos que não tem mais suporte por fabricantes ou, que por ventura, podem ocorrer instabilidades (FERNANDEZ, 2004).

2.1.4.2 Nível de integração por dados

Segundo Linthicum (1999, *apud* Fernandez, 2004), esse tipo de integração é sustentada pela troca direta de dados entre depósitos de dados sem interferir nas aplicações existentes. Essa é de fácil implementação porque não afeta a aplicação em si.

Apesar de sua implementação ser simples, existem alguns complicadores que podem surgir. A integração tende a ser mais complexa nos seguintes casos:

- a) caso exista a necessidade do envolvimento de muitas tecnologias compreendendo os depósitos de dados;
- b) onde existem complexos fluxos de dados corporativos aliados a inexistência de documentos que expliquem sua lógica, regras de processos e necessidade de transformar esses dados.

Os dados podem ser caracterizados em dois aspectos segundo MacGoveran (2003, *apud* Fernandez, 2004), sintática e semanticamente. De forma geral, os dados sintáticos envolvem o formato e a estrutura dos dados. Já a semântica envolve o significado dos dados para o negócio, onde se deve garantir a integridade do mesmo.

2.1.4.3 Nível de integração pela interface da aplicação

A integração por interface da aplicação possibilita que aplicações utilizem funcionalidades disponibilizadas por outras aplicações, provocando uma reutilização dos códigos que implementam tal aplicação. Essas interfaces acessam os recursos da aplicação que tratam processos e dados do negócio.

As API's (*Application Programming Interface*) auxiliam o acesso aos serviços encapsulados na aplicação, quem utiliza essas API's são os programadores. As "API's permitem uma alternativa de integração de mais alto nível, onde uma aplicação pode utilizar uma funcionalidade existente em outra aplicação." (JBLN, 2001 *apud* FERNANDEZ, 2004, p. 29).

2.1.4.4 Nível de integração por métodos

A integração por Métodos consiste na técnica de reutilizar processos do negócio, onde outros servidores de aplicação ou objetos distribuídos podem compartilhar os mesmos processos (LINTHICUM, 2000 *apud* FERNANDEZ, 2004).

Essa técnica tem como premissa o reuso de processos. Fernandez (2004) diz que essa se evidencia como melhor solução EAI, porém salienta que é uma solução onde se faz necessário alterar o código fonte das aplicações envolvidas trazendo muitas das vezes um custo elevado na sua implementação. É preciso avaliar se o custo envolvido é maior que o valor agregado pelo mesmo.

2.1.4.5 Nível de interface do usuário

A integração por interface do usuário trabalha no contexto da visibilidade do usuário, ou seja a tela onde são mostradas as informações. Os dados são mapeados de alguma forma na tela e re-processados. Essa integração só é utilizada quando não é possível utilizar um dos outros tipos de integração. De acordo com Fernandez (2004) essa técnica é usada

normalmente quando existem aplicações proprietárias ou quando é uma aplicação antiga e o custo/benefício da aplicação de outros métodos não é positivo.

2.1.5 Tecnologias de integração (*middleware*)

Dentro do conceito de EAI existem mecanismos, chamados de *middleware*, que são responsáveis pela comunicação entre as entidades. Esse é um “[...] software independente de aplicação que provê serviços que servem como mediadores entre as aplicações.” (GARCIA e SHINOTSUKA, 2001, p. 1).

O propósito de um *middleware* é fazer com que as diferenças tecnológicas sejam transparentes na comunicação entre as aplicações, minimizando as dificuldades de programação e viabilizando a troca de informações (FERNANDEZ 2004).

Os modelos de *middleware* são de dois tipos: lógico e físico (LINTHICUM, 2000 *apud* FERNANDEZ, 2004). Segue:

- 1) modelo lógico: Trata o movimento da informação do ponto de vista conceitual. Segundo o autor requer os seguintes tipos de configuração:
 - a) configuração um-para-um versus muitos-para-muitos: Na configuração um-para-um ou ponto-a-ponto existe um canal único direto entre duas aplicações. Efetivamente essa não é uma solução EAI. Já a configuração muitos-para-muitos, como a própria descrição diz, possibilita a integração entre muitas aplicações atendendo a necessidade de EAI;
 - b) funcionamento síncrono versus assíncrono: No funcionamento assíncrono as requisições podem ser feitas pela aplicação requisitante sem que ela interrompa seu processamento. A aplicação não fica esperando a resposta. A resposta pode não ser imediata, podendo demorar um intervalo de tempo. Esse é o mais apropriado para EAI. No funcionamento síncrono as aplicações precisam aguardar a resposta para prosseguir com seu processamento.
- 2) modelo físico: Trata o movimento da informação como realmente acontece. Os modelos de troca de mensagens são:

- c) com ou sem conexão: Mensagens podem ser trocadas quando existe uma sessão de comunicação com status de conectada (com conexão) ou desconectada (sem conexão);
- d) comunicação direta: A mensagem é passada pelo *middleware* diretamente da aplicação solicitante para solicitada;
- e) comunicação intermediada por fila de mensagem: As mensagens são enviadas de uma aplicação à outra, onde um gerenciador as armazena em uma fila de mensagens. Podem ser requisitadas futuramente, de forma assíncrona, pela aplicação a qual foi destinada. O gerenciador é independente das aplicações e pode conter regras para o atendimento;
- f) comunicação por publicação e assinatura: A comunicação é feita por um intermediador, sem que as aplicações se conheçam. O intermediador recebe as mensagens de aplicações que querem publicar. As aplicações que estão interessadas em receber tais mensagens se “inscrevem ou assinam” na lista de mensagens a serem entregues pelo intermediador. Quando as mensagens são publicadas, o intermediador as envia aos que se inscreveram na lista;
- g) comunicação por requisição e resposta: Realizada através de *middleware* que comporta essa função. Esse padrão é mais utilizado por *message brokers* e *application servers*;
- h) comunicação negociada: Nesse caso o *middleware* vai controlar o estado das regras de negócio, para poder negociar transações dependentes de estado com as aplicações.

Existem então diferentes tecnologias de *middleware* que executam suas funções de diferentes maneiras. Sordi e Marinho (2006) explica que não existe uma técnica ou tecnologia de integração que pode ser considerada melhor para uma organização, cada uma oferece vantagens e desvantagens. A organização deve usar as que mais se adequarem as suas necessidades de integração.

Nas subseções seguintes serão abordados os tipos de *middleware*. Além dos tipos comuns de componentes de *software* voltados as integrações (*middleware*), existem ainda os WS (*Web Services*). Apesar dos WS não serem considerados como *middleware*, são muito utilizados em integrações, pois possibilitam disponibilizar serviços através da *Web* de forma padronizada. Conforme Fernandez (2004) possíveis mudanças no processo do negócio e na

tecnologia tem um peso considerável em EAI. Por esse motivo, o uso de tecnologias que possibilitam deixar as integrações mais flexíveis são de extrema importância.

Tecnologias “autoconfiguráveis” como BPA (Business Process Automation), XML (Extended Markup Language), WS (Web Services) são promessas de maior flexibilidade e alternativas para a sustentação das mudanças de tecnologia e de negócios. (FERNANDEZ, 2004, p. 57)

As tecnologias de WS e XML, as quais serão utilizadas para disponibilizar os serviços entre as aplicações cliente e servidora no **SISCONT**, serão apresentadas na seção 5.1.2.

2.1.5.1 Orientados a dados

Os *middleware* que possuem tecnologias orientadas a dados, conforme Garcia e Shinotsuka (2001) possibilitam uma abstração onde é possível realizar a conexão com vários bancos de dados por uma única interface ou ainda dados distribuídos em arquivos. Como exemplos da tecnologia podemos citar a ODBC (*Open Database Connectivity*) e JDBC (*Java Database Connectivity*).

2.1.5.2 Orientados a mensagens (MOM - *Message Oriented Middleware*)

De acordo com Garcia e Shinotsuka (2001, p. 16) o MOM utiliza-se de mensagens para realizar a integração, onde “[...] provê a habilidades para criar, manipular, armazenar e comunicar essas mensagens.”. Fernandez (2004, p. 40) complementa que baseado nessa habilidade “[...] aplica um método assíncrono de trabalho que suporta falhas e não interrompe o processamento das aplicações.”.

2.1.5.3 Chamadas de procedimentos remotos (RPC – *Remote Procedure Call*)

Conforme Garcia e Shinotsuka (2001), sistemas baseados em RPC possuem aplicações distribuídas que se integram no nível de procedimento, onde realiza chamadas de procedimento pela rede. O cliente faz uma requisição (*request*) e aguarda uma resposta (*response*).

2.1.5.4 Objetos distribuídos (ORB's - *Object Request Brokers*)

Esse utiliza o conceito da tecnologia orientada a objetos para comunicação em um ambiente distribuído (GARCIA e SHINOTSUKA, 2001). Apesar de serem complexos, auxiliam para que a complexidade das plataformas e do ambiente distribuído envolvido fique transparente, por muitas vezes parecendo que o componente está localmente. A performance pode ser prejudicada por algumas dessas tecnologias apresentarem funcionamento síncrono e por serem acessadas remotamente, mas o padrão CORBA e COM já estão suportando mensagens assíncronas que podem suportar as necessidades de EAI (FERNANDEZ, 2004).

2.1.5.5 Monitores de processamento de transações (*TP Monitors*)

São baseados no conceito de transações e preservam a integridade da mesma. Garcia e Shinotsuka (2001) afirmam que esse tipo de *middleware* conserva as propriedades de atomicidade, consistência, isolamento e durabilidade, tornando uma transação segura. Fernandez (2004) complementa que é confiável para a integração em ambientes complexos, heterogêneos e distribuídos, sendo indicado para trabalhos de operações críticas.

2.1.5.6 Servidores de aplicação (*Application Servers*)

Segundo Chaves (2002, p. 2) “[...] servidores de aplicação resumem-se a APIs dentro da aplicação [...]”, possibilitando a desenvolvedores o uso de quantas camadas de *middleware* forem necessárias. Esses podem garantir as regras de negócio mantendo a integridade dos dados ou ainda disponibilizar aplicações completas com muitos serviços de transações possibilitando o acesso aos mesmos pelo cliente. Além de disponibilizar os serviços necessários à lógica da aplicação, ainda atendem a serviços e recursos *Web*.

Os servidores de aplicação além de possibilitar a utilização de outros *middleware* ainda estão em constante evolução agregando cada vez mais tecnologias e padrões de integração, conforme Fernandez (2004, p. 42) adotam alguns padrões e funcionalidades como:

- Padrões
 - SOAP (*Simple Object Access Protocol*): Protocolo para troca de informações em ambiente distribuído;
 - WSDL (*Web Services Description Language*): Descritor para definições de *Web Services* baseado em XML;
 - J2CA: Padrão utilizado para desenvolvimento de aplicações e adaptadores em J2EE;
- Funcionalidades
 - BPM (*Business Process Management*): Gerenciamento de processos de negócio;
 - *Workflow*: Controla o fluxo de trabalho;
 - TMP (*Trading Partner Management*): Gerenciamento de parceiros de negócio;
 - PRM (*Partner Relationship Management*): Gerenciamento de relacionamento com parceiros.

Fernandez ainda comenta que os servidores de aplicação tendem a desempenhar uma solução de integração completa de aplicações internas das organizações (A2A – *Application to Application*) ou externas nos seus negócios (B2B – *Business to Business* e SCM – *Supply Chain Management*, B2C – *Business to Consumer* e CRM – *Customer Relationship Management*, TMP – *Trading Partner Management* ou PRM – *Partner Relationship Management*).

2.2 Terceirização (*Outsourcing*)

Devido ao fato do **SISCONT** ser baseado no estudo de caso de uma empresa de serviços, e nesse ramo a terceirização de atividades ser um mercado que cresceu muito, com tendências de contínuo crescimento como apontado por vários autores e meios de comunicação Giosa (2003), Gonçalves (2005), EBC (2008), Administradores (2007), Info Corporate (2008), Baguete (2008) e outros, esse trabalho vai contextualizar em parte a terceirização (*outsourcing*) como tendência, o que possibilita visualizar um bom mercado para sistemas como o aqui desenvolvido.

A terceirização (*outsourcing*) teve seu nascimento, segundo Leiria (1995), em meados da II Guerra Mundial nos Estados Unidos. Inicialmente adotada pelas indústrias armamentistas, que na época tiveram que buscar apoio de outras empresas destinando a elas serviços básicos como fornecimento de ferramentas, tintas, vernizes e etc, ficando então as indústrias focadas na produção dos armamentos.

A década de 80 foi o auge da terceirização. Empresas focaram sua atenção nos clientes e na sua satisfação para tentar minar a concorrência. No Brasil não tem sido diferente. Assim como em outros países de todo mundo “[...] a terceirização tem se consolidado como uma tendência firme nas empresas brasileiras.” (GONÇALVES, 2005, p.16). Inicialmente adotada por empresas do ramo de manutenção de equipamentos, transporte, segurança, limpeza e alimentação sendo posteriormente adotada por empresas do setor industrial. Nos dias de hoje, estendeu-se para os mais diversos ramos de atividades.

Segundo o dicionário da Língua Portuguesa Michaelis (2007) terceirizar significa:

Delegar, a trabalhadores não pertencentes ao quadro de funcionários de uma empresa, funções exercidas anteriormente por empregados dessa mesma empresa. Muitas vezes, a pessoa terceirizada é um ex-funcionário, que se demite ou é demitido para exercer a mesma função de quando estava empregado.

Porém, terceirização não se aplica somente a funcionários ou pessoas de uma empresa. Pode ser tratada como delegação de tarefas ou serviços prestados por uma empresa a outra empresa, sem ter um vínculo empregatício, podendo ser estabelecido contratualmente custos, prazos, metas e resultados, e outros (MARRAS, 2000).

Franco, através de uma notícia postada na Administradores (2007) relatou que a terceirização pode ser resumida como um instrumento de sobrevivência para as empresas. Administradores e gestores se depararam com despesas de funcionários e outros insumos que não faziam parte do foco do negócio. Partindo da premissa que os custos eram elevados com o que não era a finalidade da organização, decidiram então terceirizar tais serviços dedicado mais tempo as suas atividades fim.

O aumento do outsourcing (terceirização) no mundo é um fato e, estudos mostram que essa modalidade é responsável por mudanças que elevaram o faturamento de grandes corporações. Na Europa, pesquisas revelam que mais de 50% das corporações, que já terceirizam algum setor, querem estender o outsourcing para outras áreas internas e focar, exclusivamente, nas atividades centrais, no core business, ou seja, onde a empresa detém expertise. (ADMINISTRADORES, 2007)

Para Giosa (2003) algumas das atividades executadas internamente pelas empresas podem ser redefinidas e “entregues” a terceiros. Terceirizar pode ser considerado um processo de gestão onde se distribui algumas atividades a pessoas ou empresas, ficando a empresa que terceiriza focada no negócio em si. Esse vínculo torna-se uma parceria que deveria beneficiar a ambos e que visa atingir maiores e melhores resultados.

Segundo Gonçalves (2005) terceirização muitas vezes é um meio de se reduzir custos e continuar competitiva. Não existe uma limitação para se delegar atividades a terceiros, Giosa (2003, p.32) diz:

A terceirização não tem limites. Desde que a empresa se dedique mais a sua vocação, a sua missão, seus esforços tendem a se concentrar menos na execução e mais na gestão exigindo qualidade, preço, prazo e inovações.

Franco cita algumas das principais vantagens de se terceirizar [Administradores (2007)]:

- a) facilidade na administração de funcionários;
- b) eliminação de encargos sociais e ações trabalhistas;
- c) permite ao coordenador da área (gestor) acesso a novos serviços e tecnologias;
- d) redução de custos. Isso quando se terceiriza em grande escala;

e) maior controle do *budget*⁶ através de custos planejados.

Franco salienta ainda que dessa forma o coordenador da área (gestor) pode atingir as metas de bons resultados garantindo a perenidade e sustentabilidade dos negócios.

Ainda que algumas organizações menores não confiem seus trabalhos a terceiros, alegando a perda de controle da situação, Franco afirma que é preciso buscar parceiros de confiança para surgirem os resultados e que “[...] outsourcing é como a globalização, veio para ficar e não tem como evitar.”

⁶

Budget: Significa orçamento planejado

3 PROBLEMA

Conforme apresentado anteriormente, é fato a necessidade de algumas das organizações integrarem seus SI, seja internamente A2A ou ainda externamente entre empresas parceiras B2B. Outro fato é a tendência das organizações em terceirizar funções internas que não fazem parte da finalidade do negócio. Sabendo desses fatos e tendências, será abordada a necessidade de integração entre SI de organizações parceiras, para fins de troca de dados e validação desses dados conciliando os mesmos, onde de um lado está a organização prestadora de serviços (terceirizada) e de outro a organização que terceiriza os serviços (terceirizante).

O **SISCONT** procura atender a necessidade de integração entre duas bases de dados de organizações distintas, possibilitando a extração dos dados, posteriormente os confrontado, validando e exibindo informações que listem suas diferenças. Tipicamente serão dados tais como relatórios de venda e prestação de serviços. Por exemplo, a “Empresa A” (terceirizada) informa que realizou uma quantia x de serviços técnicos para a “Empresa B” (terceirizante), a terceirizante precisa verificar se a informação corresponde ao que tem registrado no seu próprio sistema de controle de serviços técnicos utilizados da “Empresa A”. Empresas que tem um fluxo contínuo de dados procedentes do negócio com terceiros, tendem a dispensar esforços consideráveis para avaliar a veracidade desses dados. Muitas delas desenvolvem sistemas internos para tratar tal situação, mas normalmente são soluções paliativas atendendo somente uma das partes, não resolvendo o problema do processo como um todo.

O meio mais comum de integração utilizada pelas organizações Brasileiras é através de troca de arquivos, conforme Sordi e Marinho (2006). Os dados, na maioria das vezes, são extraídos de um banco de dados interno da organização.

Comumente esses arquivos são gerados pela terceirizada através de programas, em um *layout* onde os dados estão dispostos sintaticamente numa seqüência pré-determinada. Esses programas, em sua maioria, não possuem uma interface gráfica para formatação e geração dos arquivos. Por não existir uma interface que seja amigável, o responsável pela geração desses arquivos precisa ser alguém com conhecimento técnico suficiente para executar o programa, dificultando o processo. Muitas das vezes o responsável por essa tarefa é alguém da área técnica (TI). A geração dos arquivos torna-se ainda mais complexa para as organizações quando há a necessidade de serem gerados vários modelos diferentes de arquivos e

formatação de dados, pois é necessário então a implementação de várias versões dos programas que irão gerar os arquivos.

Esses arquivos devem ser enviados, de alguma forma, a outra organização que está à espera do mesmo. O processo de envio dos arquivos pode não ser bem definido, tendendo a falhas, causando transtorno e demora na comunicação. A empresa que recebe esse arquivo precisa ter obrigatoriamente um programa específico que compreenda o modelo e formatação do mesmo, a fim de realizar a leitura de, muitas vezes, milhares de linhas de dados, e que possa comparar estas informações com o que possui na sua própria base de dados, para então obter informações como, por exemplo, o volume total de serviços realizados, o valor a ser pago e, caso exista divergência, listar itens divergentes.

A necessidade de desenvolvimento de programas para geração, leitura e conciliação de arquivos de dados impede que o processo seja melhorado e agilizado entre as organizações, pois na maioria das vezes o desenvolvimento não é bem visto pelo fato de gerar um custo ou muitas das vezes por não dispor de recursos para tal desenvolvimento, visto que podem ser organizações que não sejam do segmento de TI.

Na Figura 3 pode-se visualizar o processo atual de integração, onde as empresas geram os arquivos com os dados a partir de sistemas próprios e enviam posteriormente à outra empresa, a qual concilia os dados:

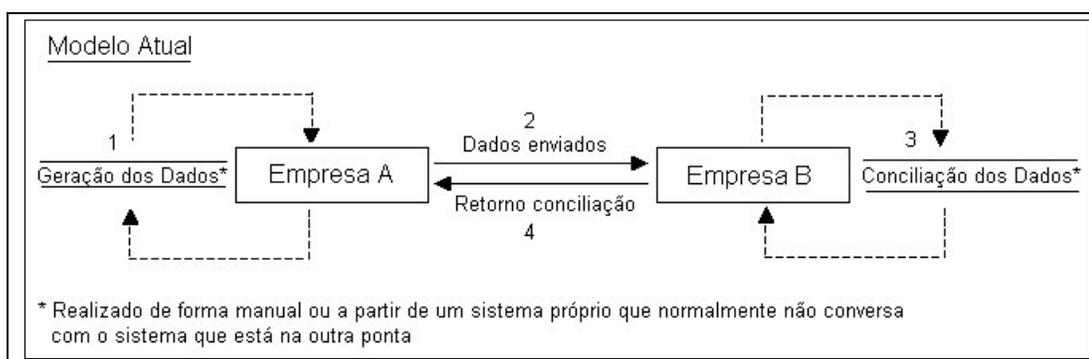


Figura 3: Modelo atual de conciliação

Fonte: Do Autor

Na Figura 1, apresentada no início desse trabalho, foi visto o modelo de processos proposto pelo **SISCONT**, onde a "Empresa B" (que contrata os serviços da "Empresa A") solicita ao **SISCONT Server**, através de *Web Service*, o arquivo que contém os dados referente às vendas realizadas por ela. O **SISCONT** busca as informações necessárias em uma base de dados da "Empresa A", gera o arquivo conforme *layout* previamente cadastrado pelo

administrador do sistema (“Empresa A”) e o retorna. A “Empresa B” recebe o arquivo e solicita a conciliação ao **SISCONT Cliente**, que está sob sua administração (domínio), o qual grava as informações desses arquivos no banco de dados dinamicamente. Após essa etapa são disponibilizadas algumas operações que podem ser realizadas sobre esses dados, através de consultas SQL previamente cadastradas pelos administrados do sistema. Essas operações conciliam os dados e retornam as informações. O processo dessa maneira torna-se rápido e fácil, não havendo necessidade de ambas as empresas implementarem programas ou sistemas separadamente para gerar arquivos com os dados e conciliá-los.

3.1 Conciliação

Para melhor entendimento das ferramentas que o **SISCONT** vai disponibilizar, deve-se entender também o conceito de conciliação. Segundo o dicionário da língua Portuguesa Michaelis (2007) conciliação é a “Ação ou efeito de conciliar.” ou ainda a “Interpretação conciliatória de textos que parecem discordantes.”.

Essa descrição retrata o que foi exposto anteriormente. A pretensão é realizar a conciliação de dados que podem ser mensurados de alguma forma, como quantidades, valores monetários e outros, assim procurando expor possíveis discordâncias, representadas pelos dados, nos negócios realizados entre a terceirizada e a terceirizante.

Vejamos um exemplo hipotético onde temos uma empresa de calçados, chamada de “Empresa A”, que presta serviço e uma outra empresa de calçados, chamada de “Empresa B”, que solicita serviços. A Figura 4 ilustra a realização dos serviços prestados entre ambas as empresas onde existem divergências nos dados apresentados. Logo abaixo dos serviços prestados está um exemplo de conciliação realizada.

Serviços prestados pela Empresa A						Serviços solicitados pela Empresa B									
Contratante: Empresa B Período: 01/10/2008 A 31/10/2008						Contratada: Empresa A Período: 01/10/2008 A 31/10/2008									
Data	Produto	Serviço	Quantidade	Valor unitário	Total	Data	Produto	Serviço	Quantidade	Valor unitário	Total				
1/out	sandália	a	1000	R\$ 0,50	R\$ 500,00	1/out	sandália	a	1000	R\$ 0,50	R\$ 500,00				
2/out	sandália	a	2000	R\$ 0,50	R\$ 1.000,00	2/out	sandália	a	2000	R\$ 0,50	R\$ 1.000,00				
3/out	sapato	d	500	R\$ 0,58	R\$ 290,00	3/out	sapato	d	500	R\$ 0,58	R\$ 290,00				
4/out	sapato	e	700	R\$ 0,40	R\$ 280,00	4/out	sapato	a	700	R\$ 0,40	R\$ 280,00				
5/out	sapato	e	100	R\$ 0,40	R\$ 40,00	5/out	sapato	e	100	R\$ 0,40	R\$ 40,00				
6/out	sandália	b	350	R\$ 0,49	R\$ 171,50	6/out	sandália	b	350	R\$ 0,49	R\$ 171,50				
7/out	sandália	b	350	R\$ 0,49	R\$ 171,50	7/out	sandália	b	350	R\$ 0,49	R\$ 171,50				
8/out	sandália	a	350	R\$ 0,50	R\$ 175,00	8/out	sandália	a	350	R\$ 0,49	R\$ 171,50				
9/out	sandália	c	840	R\$ 0,59	R\$ 495,60	9/out	sandália	a	840	R\$ 0,49	R\$ 411,60				
10/out	sandália	c	100	R\$ 0,59	R\$ 59,00	10/out	sandália	c	100	R\$ 0,59	R\$ 59,00				
11/out	tênis	f	2000	R\$ 0,35	R\$ 700,00	11/out	tênis	f	2000	R\$ 0,35	R\$ 700,00				
12/out	tênis	f	3000	R\$ 0,35	R\$ 1.050,00	12/out	tênis	f	3000	R\$ 0,35	R\$ 1.050,00				
13/out	sapato	e	50	R\$ 0,40	R\$ 20,00	13/out	sapato	e	50	R\$ 0,40	R\$ 20,00				
14/out	sapato	e	75	R\$ 0,40	R\$ 30,00	14/out	sapato	e	75	R\$ 0,40	R\$ 30,00				
15/out	sandália	b	120	R\$ 0,49	R\$ 58,80	15/out	sandália	b	120	R\$ 0,49	R\$ 58,80				
16/out	sandália	a	1000	R\$ 0,50	R\$ 500,00	16/out	sandália	a	100	R\$ 0,50	R\$ 50,00				
17/out	sandália	a	2000	R\$ 0,50	R\$ 1.000,00	17/out	sandália	a	2000	R\$ 0,50	R\$ 1.000,00				
18/out	sapato	d	500	R\$ 0,58	R\$ 290,00	18/out	sapato	d	500	R\$ 0,58	R\$ 290,00				
19/out	sapato	e	700	R\$ 0,40	R\$ 280,00	19/out	sapato	e	700	R\$ 0,40	R\$ 280,00				
20/out	sapato	e	100	R\$ 0,40	R\$ 40,00	20/out	sapato	e	100	R\$ 0,40	R\$ 40,00				
21/out	sandália	b	350	R\$ 0,49	R\$ 171,50	21/out	sandália	b	350	R\$ 0,49	R\$ 171,50				
22/out	sandália	b	350	R\$ 0,49	R\$ 171,50	22/out	sandália	b	350	R\$ 0,49	R\$ 171,50				
23/out	sandália	a	350	R\$ 0,49	R\$ 171,50	23/out	sandália	a	350	R\$ 0,49	R\$ 171,50				
24/out	sandália	c	840	R\$ 0,59	R\$ 495,60	24/out	sandália	c	840	R\$ 0,59	R\$ 495,60				
25/out	sandália	c	100	R\$ 0,59	R\$ 59,00	25/out	sandália	c	100	R\$ 0,59	R\$ 59,00				
26/out	tênis	f	2000	R\$ 0,35	R\$ 700,00	26/out	tênis	f	2000	R\$ 0,35	R\$ 700,00				
27/out	tênis	f	3000	R\$ 0,35	R\$ 1.050,00	27/out	tênis	f	3000	R\$ 0,35	R\$ 1.050,00				
28/out	sapato	e	50	R\$ 0,40	R\$ 20,00	28/out	sapato	e	50	R\$ 0,40	R\$ 20,00				
29/out	sapato	e	75	R\$ 0,40	R\$ 30,00	29/out	sapato	e	75	R\$ 0,40	R\$ 30,00				
30/out	sandália	b	120	R\$ 0,49	R\$ 58,80	30/out	sandália	b	120	R\$ 0,49	R\$ 58,80				
31/out	sandália	b	50	R\$ 0,49	R\$ 24,50	31/out	sandália	b	50	R\$ 0,49	R\$ 24,50				
Totais					23120	R\$	10.103,80	Totais					22220	R\$	9.566,30
CONCILIAÇÃO															
Data	Produto	Serviço	Quantidade	Valor unitário	Total	Data	Produto	Serviço	Quantidade	Valor unitário	Total				
8/out	sandália	a	350	R\$ 0,50	R\$ 175,00	8/out	sandália	a	350	R\$ 0,49	R\$ 171,50				
9/out	sandália	c	840	R\$ 0,59	R\$ 495,60	9/out	sandália	a	840	R\$ 0,49	R\$ 411,60				
16/out	sandália	a	1000	R\$ 0,50	R\$ 500,00	16/out	sandália	a	100	R\$ 0,50	R\$ 50,00				

Figura 4: Exemplo de conciliação dos serviços hipotéticos solicitados pela Empresa A (terceirizada) a Empresa B (terceirizante)

A "Empresa B" (que contrata os serviços da "Empresa A"), recebe as informações referentes dos serviços prestados e necessita verificar se estas informações estão de acordo com o que possui registrado como realmente realizado pela "Empresa A". Essa verificação pode ser realizada de várias maneiras. Porém normalmente é realizada de alguma forma não totalmente automatizada, inclusive algumas vezes sendo feita de forma manual, verificando-se cada registro visualmente. Mesmo as formas automatizadas de verificação ainda carecem de melhorias no processo, pois a falta de softwares especializados, diferentes tipos de programas e sistemas, que normalmente não conversam entre si, e bases de dados gerenciadas por diferentes SGBD impedem que a verificação auxiliada por computador seja implementada de forma rápida e fácil.

Dependendo do tipo de negócio e do volume dos serviços, a necessidade dessa conciliação pode ser diária. Empresas que terceirizam parte do seu negócio e que possuam um volume razoável de serviços com a terceirizada, precisam de alguma maneira realizar uma conferência desses serviços para, por exemplo, realizar o pagamento dos serviços prestados.

Seguindo a linha de raciocínio, a simples geração de relatórios que contenham informações relevantes ao serviço, será satisfatório até o momento em que seja possível

detectar diferenças entre o que apresenta o relatório de alguma das partes e o que a outra parte considera que realmente foi feito. A partir do momento que se torna difícil detectar as diferenças dos dados informados entre os relatórios apresentados, sejam estes relatórios grandes ou pequenos, tornam-se improdutivos.

Na Figura 4 foram exibidos dois relatórios que correspondem aos serviços hipotéticos realizados entre duas empresas de calçados, do lado esquerdo os serviços efetivados segundo a “Empresa A” (terceirizada) e do lado direito os serviços efetivados segundo a “Empresa B” (terceirizante). São relatórios relativamente pequenos, possuem poucas informações e existem divergências nas quantidades apresentadas entre ambos. Mesmo sendo relatórios pequenos e tendo poucas divergências, é trabalhoso realizar a conciliação entre os dois relatórios de uma forma manual, o que permite entender a dificuldade da análise de relatórios maiores, com milhares de linhas. Sem dúvida o processo seria demorado, ou mesmo inviável.

Uma forma normalmente utilizada pelas empresas para realizar a conciliação é através da geração de informações de formas diferentes de um simples relatório, normalmente gerando algum tipo de arquivo, seja arquivo tipo texto, ou planilhas eletrônicas ou até mesmo criando usuários em suas bases de dados com permissões específicas para que possam conectar-se e buscar as informações necessárias. Porém, na maioria das vezes, não apresentam uma solução completa. O **SISCONT** visa a solução completa através das ferramentas disponibilizadas.

3.2 A Empresa

A fim de possibilitar uma aplicação prática do sistema desenvolvido, será apresentada uma situação específica para melhor entendimento. Essa serviu como base para a solução do sistema desenvolvido. Trata-se de uma empresa de TI focada em rede de captura de dados denominada GetNet - <http://www.getnet-tecnologia.com.br>.

A GetNet possui uma rede de captura de dados que abrange a maior parte das cidades do Brasil e através dessa rede disponibiliza aos mais de cem mil estabelecimentos credenciados diversas soluções, tanto próprias quanto de terceiros.

Dentre as soluções, destaca-se a prestação de serviços para as grandes operadoras de telefonia do Brasil. A empresa disponibiliza para estabelecimentos credenciados a recarga de créditos para celular, onde o estabelecimento está habilitado a vender créditos aos clientes que

possuem celulares pré-pagos, sendo comissionado pelas vendas efetuadas. Da mesma forma que a GetNet comissiona os estabelecimentos, as operadoras também comissionam a GetNet. Surge então a necessidade de conciliação destas vendas em quantidades e valores vendidos entre “Estabelecimentos \times GetNet” e “GetNet \times Operadoras”.

Por se tratar de grandes volumes de vendas, a GetNet disponibiliza então uma conciliação diária para operadoras e algumas redes de estabelecimentos. São gerados arquivos em texto puro (ASCII) contendo informações relevantes sobre essas vendas. Tanto as operadoras como as redes de estabelecimentos lêem estes arquivos a partir de um sistema próprio e realizam uma conciliação entre os valores informados pela GetNet e os seus próprios valores. Caso ocorra divergência entre os valores apresentados pela GetNet, tanto operadoras quanto redes de estabelecimentos entram em contato com a empresa para tratar a situação.

As principais dificuldades que podem ser apontadas no modelo atual de geração dos arquivos são:

- a) os arquivos gerados normalmente diferem para cada operadora ou rede de estabelecimento, dificultando o trabalho da equipe técnica, pois não existe um padrão único;
- b) os códigos são gerados a partir de *scripts bash*⁷ e a cada novo *layout* (padrão dos dados nos arquivos) um novo código deve ser reescrito, muitas vezes demorando horas ou dias para a geração de um novo *layout*;
- c) não existe uma interface gráfica para gerenciar *layout* dos arquivos através dos *scripts bash*;
- d) o ambiente do sistema operacional, onde ficam os *scripts bash*, é GNU/Linux sem interface gráfica. Por se tratar de um ambiente GNU/Linux acessível somente via linha de comando, mesmo com a ampla divulgação, aceitação e uso por parte dos profissionais de TI, muitos só conhecem superficialmente e não estão bem adaptados com os comandos disponíveis, assim gerando mais dificuldades na manutenção e geração dos arquivos exportados;
- e) poucos profissionais conhecem ou desenvolvem *scripts bash* a ponto de conseguir gerar um novo *layout* ou até mesmo realizar a manutenção dos que já existem.

⁷

Bash: Referência a scripts para execução de comandos ou uma rotinas de tarefas no GNU/Linux

f) os arquivos não são disponíveis a qualquer hora (não *on-line*), existem agendamentos pré-determinados para geração e envio dos arquivos. Arquivos são enviados para diretórios de arquivos onde as empresas parceiras buscam tais arquivos;

g) é necessário que empresas parceiras desenvolvam alguma forma de buscar e ler os arquivos, seja através de programas/*scripts*, ou, nos piores casos, manualmente;

3.3 Problemas Tratados

Tendo sido apresentados os pontos críticos, é possível identificar os problemas a serem tratados:

a) existe dificuldade na conciliação dos dados disponibilizados através de simples relatórios, quando o volume é considerável;

b) não existe uma interface gráfica para geração dos arquivos e *layout* dos dados;

c) por não existir uma interface gráfica, normalmente quem gera as informações são pessoas da TI. O processo dessa maneira não é simplificado necessitando intervenção de pessoas específicas com conhecimentos específicos;

d) podem existir várias saídas de dados, *layout* diferentes, tendendo a demorar na criação de um novo programa/*script* que gera os dados para o novo *layout* necessário;

e) quem recebe os dados através de um arquivo ou de alguma outra forma mencionada anteriormente, precisa desenvolver um sistema para ler e realizar uma conciliação das informações;

f) por necessitar de um sistema específico para ler as informações, poucos utilizam essa ferramenta e acabam realizando a conciliação de uma forma não automatizada, isso torna o processo demorado e, por consequência dessa demora, podem ocorrer atrasos de pagamentos causando transtornos para ambas as partes;

g) não possui mobilidade e independência nos trabalhos, pois alguém deve estar *in-loco*⁸ ou existir algum processo que gere os arquivos em determinados momentos e os disponibilize de alguma maneira.

⁸

in-loco: No local físico onde se encontra

3.4 Visão Macro das Funcionalidades do Sistema

- a) o sistema **SISCONT** terá uma Interface gráfica, sendo desenvolvido em um ambiente *Web*, possibilitando a criação de arquivos de dados para posterior leitura;
- b) o **SISCONT** deve possibilitar a conexão com um SGBD escolhido pelo usuário e suportado pela biblioteca ADOdb (no capítulo a seguir serão dados mais detalhes sobre esta e outras tecnologias relacionadas nesta sessão);
- c) deverá suportar agrupamentos de dois ou mais *layout* (seções) gerando um novo *layout* único (arquivo);
- d) a consulta dos campos e dados para gerar o *layout* (seções), deve ocorrer através de SQL previamente criado pelo administrador, podendo ser realizada uma conexão a qualquer um dos SGBD suportados pela ADOdb;
- e) poderá existir máscara de formatação para campos dos *layout* (seções) que formam o *layout* único (arquivo) provenientes de um SQL;
- f) gerar arquivos em texto puro (ASCII) ou em XML *on-line*, podendo ser solicitado à geração a qualquer momento;
- g) suportar requisições de arquivos XML através de serviços de *Web Services*;
- h) caso o arquivo gerado estiver no formato de XML, o **SISCONT** poderá ler este arquivo e exibir em tela como se fosse um relatório.
- i) conciliar dois arquivos, no formato de XML, listando divergências entre ambos, caso existam. Neste caso o usuário final não necessitará de um sistema próprio para ler o arquivo e ao mesmo tempo conciliar os dados, o próprio **SISCONT** irá fazer esse trabalho.

3.4.1 Delimitações da versão inicial do sistema

Funcionalidades que poderão ser abordadas em versões futuras do sistema:

- a) geração automática da especificação e documentação de cada *layout* de arquivo;

- b) agendamento para geração dos arquivos usando a *Crontab*⁹ do sistema operacional GNU/Linux e usando o PHP através de chamadas de linha de comando;
- c) realizar a leitura de arquivos XML gerando tabelas dinamicamente no banco de dados armazenando os dados para uma posterior consulta e ou qualquer outro tipo de validação necessária. A disponibilidade dessa operação vai facilitar para a integração entre as organizações podendo importar dados de outras bases de dados;
- d) outros serviços disponíveis através de *Web Services*.

⁹ Crontab: Agendador de tarefas do sistema operacional GNU/Linux

4 METODOLOGIA

Na análise do **SISCONT** foi utilizada a metodologia de desenvolvimento Orientado a Objetos (OO). O modelo de processo de *software* utilizado para implementação foi o incremental. Segundo Pressman (2002) esse modelo combina o modelo seqüencial linear com o modelo de prototipagem, onde o *software* é desenvolvido de forma linear gerando versões (incrementos) do *software*. A cada incremento o *software* vai sendo aperfeiçoado até chegar ao produto final acabado. O modelo incremental foi utilizado no desenvolvimento do **SISCONT** pelo fato desse possibilitar a geração de versões iniciais onde posteriormente podem ser incrementadas ou melhoradas. Outro fator relevante para o uso desse modelo foi à possibilidade de dispensar menos recursos iniciais para o desenvolvimento dos primeiros incrementos.

Analisando os requisitos, onde se levantou as necessidades do sistema, resolveu-se dividir em três incrementos, como segue:

- 1) primeiro incremento: Funções de gestão básicas dos sistemas cliente e servidor para visão do administrador;
- 2) segundo incremento: Aperfeiçoamento das funções de gestão anteriormente criadas adaptando para visão dos usuários não administradores;
- 3) terceiro incremento: Evolução do sistema e possíveis melhorias propostas por usuários.

Dos três incrementos, foram executados os dois primeiros. O terceiro incremento será executado após disponibilizar o **SISCONT** para uso, onde será analisada a aceitação dos usuários, possíveis *feedback*¹⁰ e solicitações de melhorias.

¹⁰ Feedback: Consiste no retorno ou resposta, seja ele bom ou ruim, sobre algo. Nesse caso, os usuários devem retornar informações sobre o uso do sistema, pode ser melhorias, mau funcionamento ou novas necessidades.

4.1 Análise de Requisitos

Segundo Pressman (1995, p. 199) o primeiro passo é identificar a necessidade de um sistema, a “[...] identificação da necessidade é o ponto de partida na evolução de um sistema baseado em computador [...]”. Sobre a análise de requisitos, Pressman (2002) diz que essa prove ao projetista de *software* a representação das funções e comportamentos que serão moldados nas camadas do *software*.

Foram levantados os problemas apresentados pela empresa GetNet e sobre esses especificados os requisitos. Realizou-se também um estudo buscando informações e tecnologias para poder realizar a integração entre os sistemas das organizações.

Na análise levantou-se a necessidade de desenvolvimento de um sistema *Web*. O sistema deve ser baseado em uma aplicação cliente e outra servidora. O sistema obrigatoriamente deve suportar os requisitos mínimos que já foram listados anteriormente nos capítulos 3.3 e 3.4, bem como os descritos na subseção seguinte.

4.1.1 Descrição dos requisitos

Os requisitos foram separados em requisitos comuns para as duas aplicações e requisitos específicos da aplicação cliente e servidora.

4.1.1.1 Requisitos comuns à aplicação cliente e servidora

Sempre, no início do acesso ao sistema, na tela de login deverá ser informado o usuário e a senha para entrar no sistema. Haverá permissões distintas para cada perfil de acesso. O administrador terá acesso a todas as funcionalidades e um usuário não administrador vai ter restrições, podendo acessar somente a algumas das funcionalidades. Na tentativa de login, caso não seja encontrado o usuário com a senha informada, o sistema não deverá permitir o acesso.

Os arquivos de dados poderão conter uma ou várias seções, sendo que cada seção poderá estar ligada a somente um SQL que pode trazer vários registros com vários campos para a seção.

O SQL que vai gerar os dados de uma seção deve ser devidamente cadastrado pela interface gráfica. O SQL deverá suportar o recebimento de parâmetros variáveis que vai depender da necessidade de cada cadastro. A conexão a banco de dados deve ser genérica, de forma que se conectará aos principais bancos de dados disponíveis no mercado, assim não ficando limitado a uma solução.

As seções dos arquivos devem poder configurar a formatação de cada campo vindo da consulta SQL.

Os arquivos devem poder ser exportados no formato XML e ASCII. Quando exportados devem ter um *layout* pré-determinado (padronizado).

4.1.1.2 Requisitos específicos da aplicação servidora

Os arquivos poderão ser requisitados pela aplicação cliente através de um serviço *Web Service*. Nesse caso o arquivo a ser enviado pode ser somente no formato XML. Essas requisições deverão ser seguras com validação de usuário e senha. Por este motivo deverá ter uma tela para cadastro de usuário e senha para o *Web Service*.

O usuário não administrador terá acesso somente à geração ou lista de arquivos.

4.1.1.3 Requisitos específicos da aplicação cliente

Um arquivo XML, que esteja no *layout* padronizado, poderá ser lido exibindo os dados como se fosse um relatório. Deve possibilitar também a conciliação de dois arquivos XML exibindo as possíveis divergências. Deverá suportar a criação de operações para conciliação dos arquivos.

Será necessária uma tela onde sejam cadastrados os arquivos disponíveis pela aplicação servidora. Esses arquivos poderão ter parâmetros estáticos, variáveis ou nenhum. Os arquivos serão solicitados para a aplicação servidora através de um serviço de *Web Service*.

Nesse caso o arquivo a ser solicitado pode ser somente XML. Essas requisições deverão ser seguras com validação de usuário e senha, por isso deve suportar o seu cadastro. Uma aplicação cliente não deve ficar restrita a somente uma aplicação servidora, poderá solicitar arquivos de várias aplicações servidoras evitando assim a necessidade de existir outras aplicações clientes na mesma organização para integração com outras aplicações servidoras.

O usuário não administrador terá acesso somente as funcionalidades de listar ou gerar arquivo de dados, ler ou conciliar arquivos XML e listar ou buscar arquivos do servidor.

4.2 Diagramas UML

Para a fase de análise foram utilizados os diagramas UML para especificar e documentar o projeto do sistema.

Segundo Martins (2004, p. 132) UML é uma linguagem para documentação de projetos de *software* podendo “[...] visualizar, especificar, construir e documentar os elementos de um sistema baseado em *software* [...]”.

Para produzir e descrever os aspectos do sistema, a UML dispõe de vários tipos de diagramas. Os aspectos considerados são funcional (estrutura estática e interação dinâmica), não funcional (tempo de processamento, confiabilidade, produção) e organizacional (organização do trabalho, mapeamento e código) (FURLAN, 1998).

Foram utilizados diagramas de classes para denotar a estrutura estática do sistema e as classes que representam a manipulação de “coisas” do sistema, diagrama de casos de uso para descrever as funcionalidades do sistema com seus atores e diagrama de seqüência que mostram as interações do usuário com o sistema.

4.2.1 Diagrama de casos de uso

Foram especificados os diagramas de caso de uso da aplicação servidora e cliente. Abaixo serão listados os principais casos de uso, casos de uso mais simples como listar os cadastros efetuados não foram descritos abaixo, porém deverão ser implementados no sistema. Segue:

a) aplicação servidora

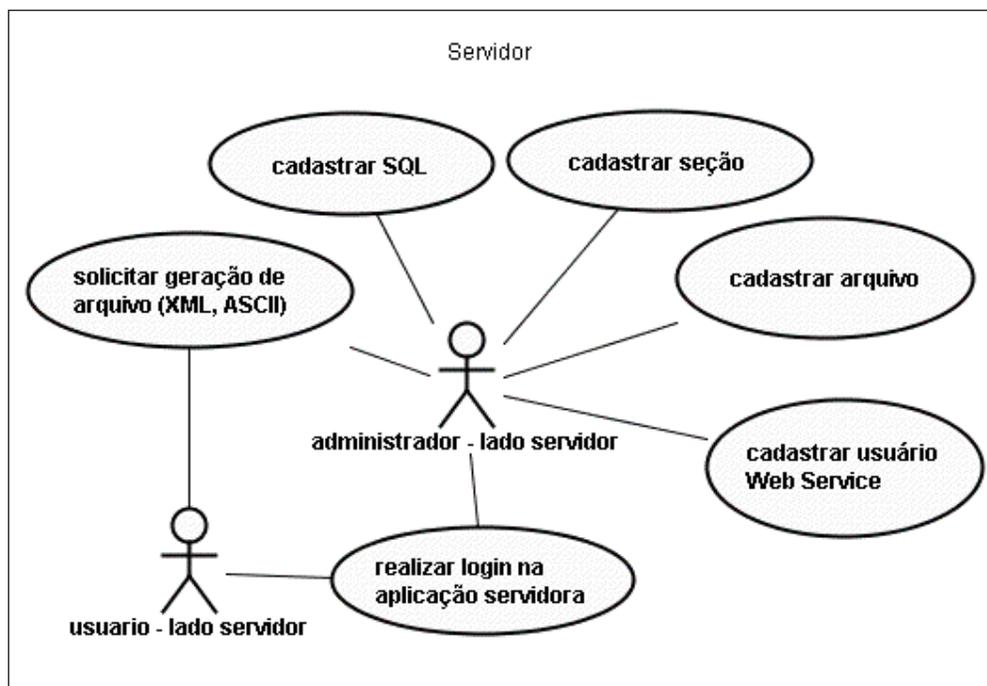


Figura 5: Diagrama caso de uso - aplicação servidora

Realizar Login

- atores: administrador e usuário não administrador.
- descrição: o sistema deverá ser acessado somente por dois tipos de usuários, o administrador terá acesso a todas as funcionalidades e um usuário não administrador que poderá somente gerar os arquivos XML/ASCII ou listar os arquivos cadastrados. Deverá então conter uma tela de login para entrada de usuário e senha e dependendo do tipo de usuário, os menus deverão ser limitados conforme descrito acima.
- dados: usuário e senha.
- resposta: acesso ao sistema ou dados informados não conferem.

Cadastrar SQL

- atores: administrador.
- descrição: administrador cadastra os dados e SQL necessário para poder realizar uma consulta a banco. Este SQL será usado para posteriormente gerar uma seção e

está seção poderá fazer parte de um arquivo. O sql deverá suportar o recebimento de parâmetros variáveis dependendo da necessidade.

- dados: descrição, dados para conexão e SQL.
- resposta: sucesso ou erro no cadastramento

Cadastrar Seção

- atores: administrador.
- descrição: deve ser exibida uma lista dos SQL já cadastrados no sistema. Quando o administrador selecionar o SQL para cadastrar uma seção deverá carregar os campos vindos do sql para inserir as informações e formatação dos mesmos. Nessa mesma tela devem ser exibidas as informações necessárias para a seção em si.
- dados: descrição, apelido para seção e configurações (meta dados) de cada campo.
- resposta: sucesso ou erro no cadastramento.

Cadastrar Arquivo

- atores: administrador.
- descrição: deve ser exibida uma lista das seções já cadastradas no sistema. O administrador vai selecionar as seções que irão fazer parte do arquivo. O arquivo será agrupado e gerado conforme a seqüência em que forem selecionadas as seções.
- dados: descrição, apelido para o arquivo e código das seções que farão parte do arquivo.
- resposta: sucesso ou erro no cadastramento.

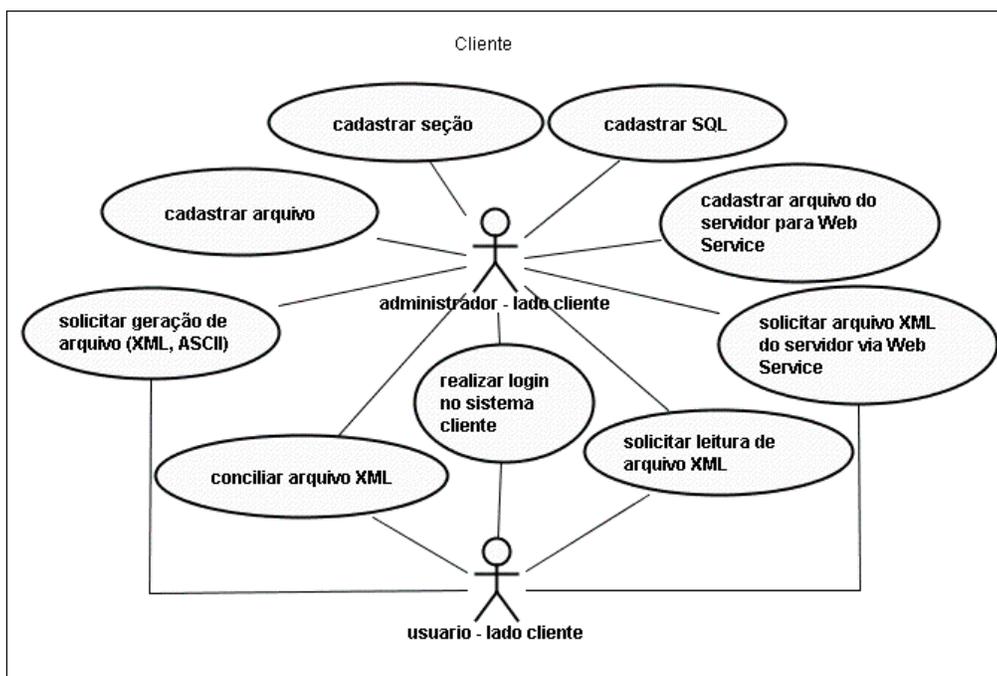
Solicitar Geração de Arquivo (XML, ASCII)

- atores: administrador, usuário não administrador.
- descrição: devem ser exibidas uma lista com a descrição dos arquivos já cadastrados no sistema. O usuário vai selecionar um dos arquivos que pode ser composto por uma ou mais seções e selecionar a opção de gerar no formato ASCII (TXT) ou XML. Caso exista a necessidade de parâmetros, os mesmos deverão ser informados. Após, deverá fazer o *download* do arquivo selecionado.
- dados: arquivo a ser gerado e tipo de arquivo.
- resposta: *download* do arquivo ou erro.

Cadastrar Usuário para *Web Service*

- atores: administrador.
- descrição: deve ser cadastrado o usuário e senha com a descrição para cada usuário.
- dados: descrição, usuário e senha.
- resposta: sucesso ou erro no cadastramento.

b) aplicação cliente



Os casos de uso Realizar Login, Cadastrar SQL, Cadastrar Seção, Cadastrar Arquivo e Solicitar Geração de Arquivo (XML, ASCII) são exatamente iguais nas duas aplicações e por isso não serão listados novamente nos casos de uso da aplicação cliente.

Cadastrar Arquivo do Servidor para *Web Services*

- atores: administrador.
- descrição: O SISCONT cliente vai poder baixar os arquivos em XML realizando uma chamada para o SISCONT server através de um serviço *Web Service*. Para isso o cliente deve disponibilizar uma interface para cadastrar a url do WSDL, o

código do arquivo XML que deverá ser baixado com os seus parâmetros caso seja necessário junto com o usuário e a senha. Os parâmetros para o arquivo poderão ser estáticos ou variáveis conforme a necessidade.

- dados: descrição, URL WSDL, código do arquivo, usuário, senha e parâmetros necessários para o arquivo.
- resposta: sucesso no cadastro ou erro.

Solicitar Arquivo XML do Servidor via *Web Service*

- atores: administrador e usuário não administrador.
- descrição: poderá selecionar um arquivo previamente cadastrado para buscar via *Web Service* no servidor. Se houver a necessidade de parâmetros, antes de fazer a requisição, deverá informá-los.
- dados: dados para conexão *Web Service*, código do arquivo e parâmetros caso houver.
- resposta: retorno do arquivo solicitado em XML ou erro.

Solicitar Leitura de Arquivo XML

- atores: administrador e usuário não administrador.
- descrição: arquivos XML que estejam no formato padrão poderão ser lidos e exibidos em tela em forma de relatório.
- dados: arquivo XML formatado no padrão esperado.
- resposta: exibição em forma de relatório ou erro.

Conciliar Arquivo XML

- atores: administrador e usuário não administrador.
- descrição: o sistema deverá suportar a leitura de dois arquivos XML (um do lado do servidor e outro do lado do cliente) que estejam no formato do padrão esperado e que façam referência aos mesmos campos, realizando uma conciliação entre ambos exibindo o resultado para o usuário. Deve suportar também a realização de determinadas operações sobre os campos. Essas operações deverão ser previamente cadastradas pelo administrador.
- dados: arquivo XML servidor, arquivo XML cliente.

- resposta: conciliação dos arquivos e operações disponibilizadas exibindo os dados na tela ou erro.

4.2.2 Diagramas de seqüência

a) aplicação servidora

Realizar Login

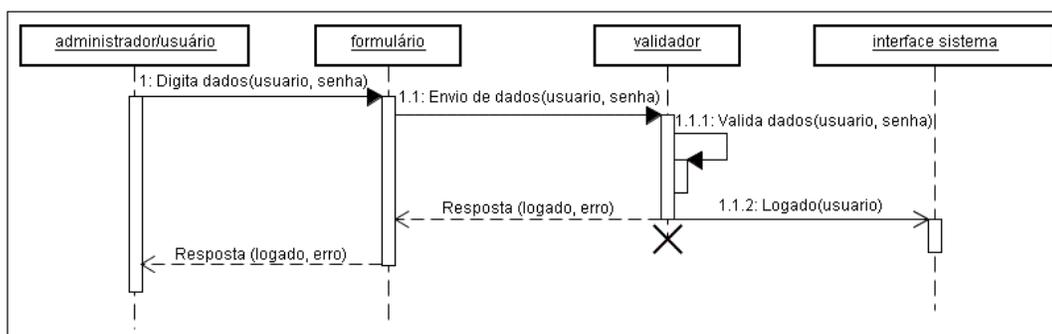


Figura 7: Diagrama de seqüência – realizar login

Cadastrar SQL

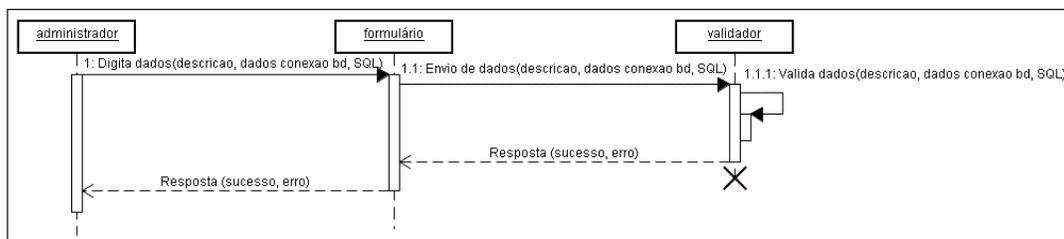


Figura 8: Diagrama de seqüência – cadastrar SQL

Cadastrar Seção

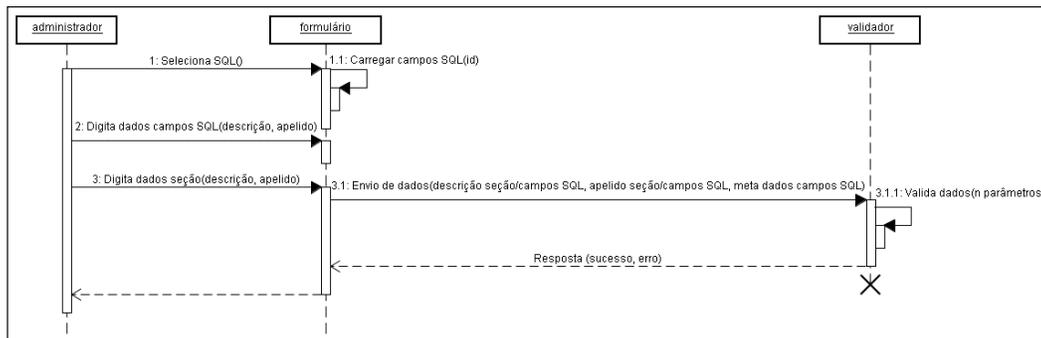


Figura 9: Diagrama de seqüência – cadastrar seção

Cadastrar Arquivo

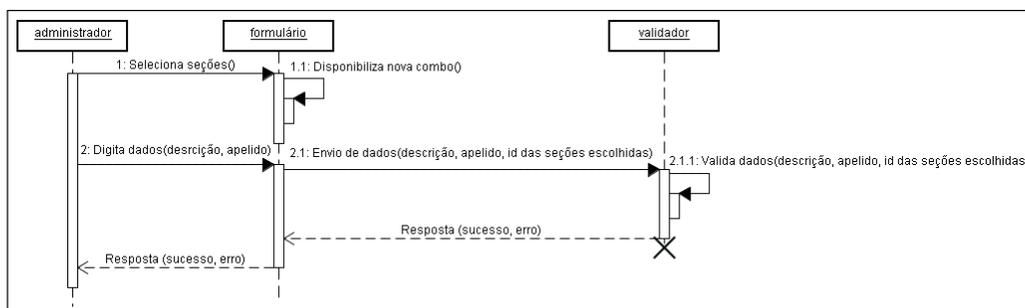


Figura 10: Diagrama de seqüência – cadastrar arquivo

Solicitar Geração de Arquivo (XML, ASCII)

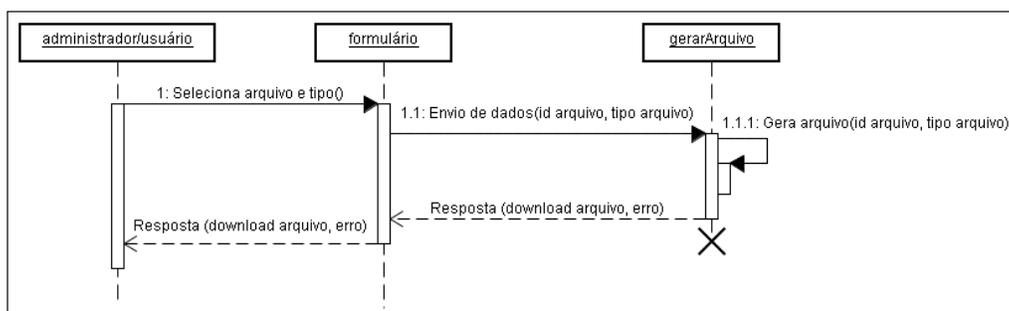


Figura 11: Diagrama de seqüência – solicitar geração de arquivo (XML, ASCII)

Cadastrar Usuário para *Web Service*

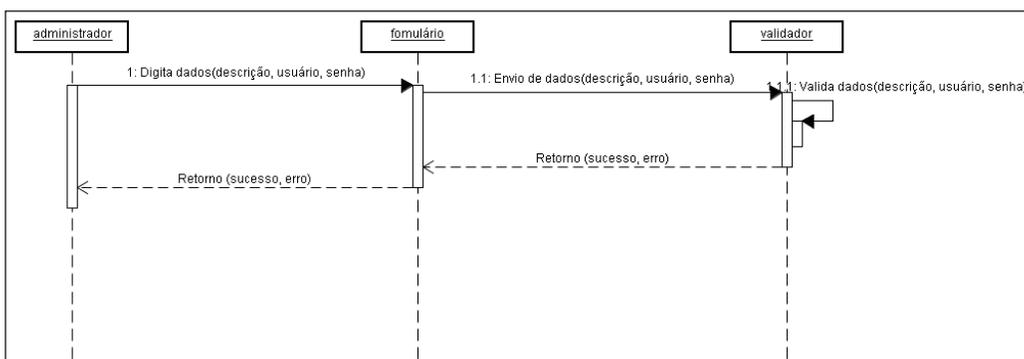


Figura 12: Diagrama de seqüência – cadastrar usuário para *Web Service*

b) aplicação cliente

Os diagramas de seqüência Realizar Login, Cadastrar SQL, Cadastrar Seção, Cadastrar Arquivo e Solicitar Geração de Arquivo (XML, ASCII) são exatamente iguais nas duas aplicações e por isso não serão listados novamente nos diagramas da aplicação cliente.

Cadastrar Arquivo do Servidor para *Web Services*

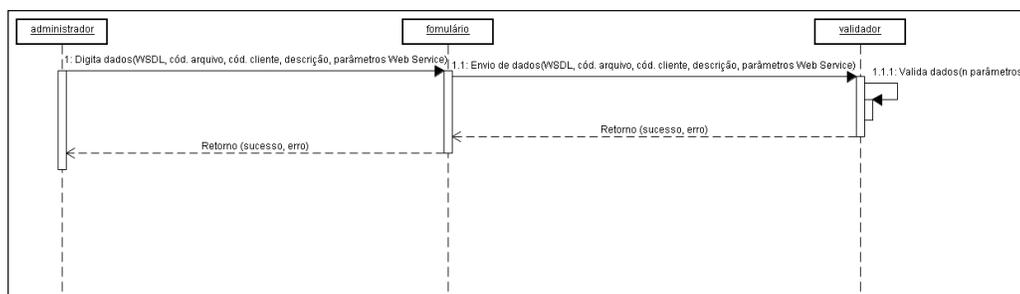


Figura 13: Diagrama de seqüência – cadastrar arquivo do servidor para *Web Service*

Solicitar Arquivo XML do Servidor via *Web Service*

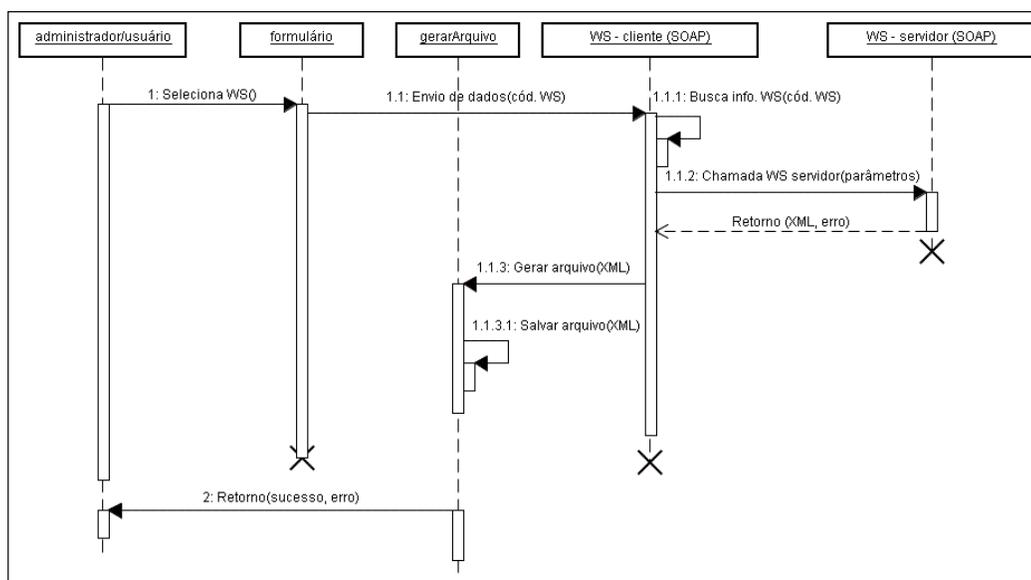


Figura 14: Diagrama de seqüência – solicitar arquivo XML do servidor via *Web Service*

Solicitar Leitura de Arquivo XML

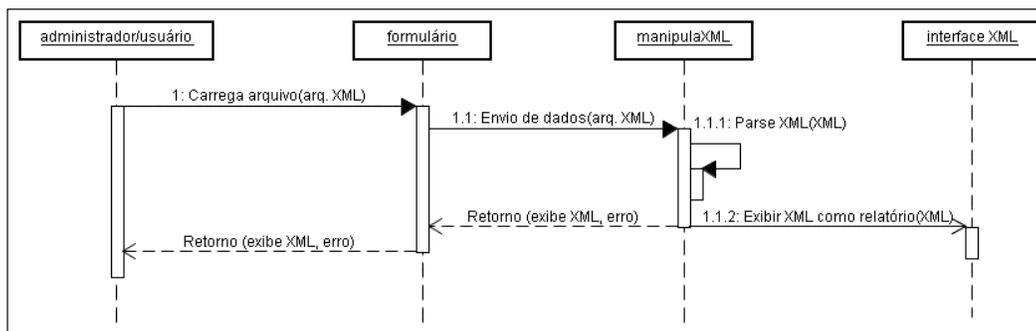


Figura 15: Diagrama de seqüência – solicitar leitura de arquivo XML

Conciliar Arquivo XML

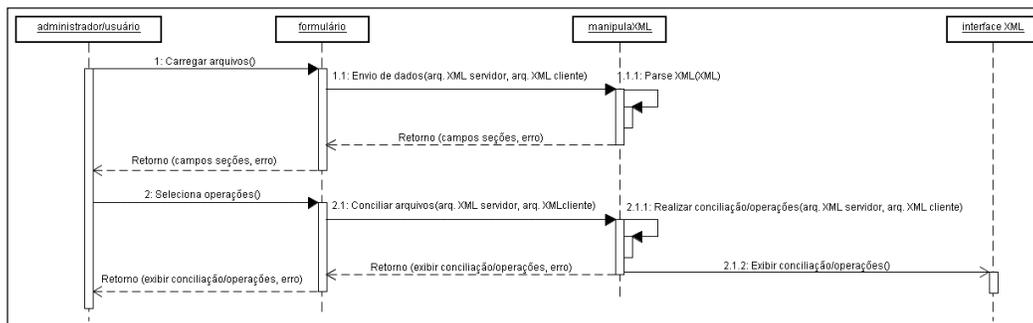


Figura 16: Diagrama de seqüência – conciliar arquivo XML

4.2.3 Diagrama de classe

Devido ao fato de usar no desenvolvimento do **SISCONT** a padronização MVC (*Model-View-Controller*) e na camada de *model* utilizar o padrão DAO (*Data Access Object*) para classes de acesso a banco de dados, padrões esses que serão apresentados na seção 4.4, no diagrama de classes serão mostradas somente as classes DAO de acesso à banco. Porém, existem ainda as classes VO (*Value Object*) e classes Utils que ligam as classes DAO recebendo os parâmetros da VO.

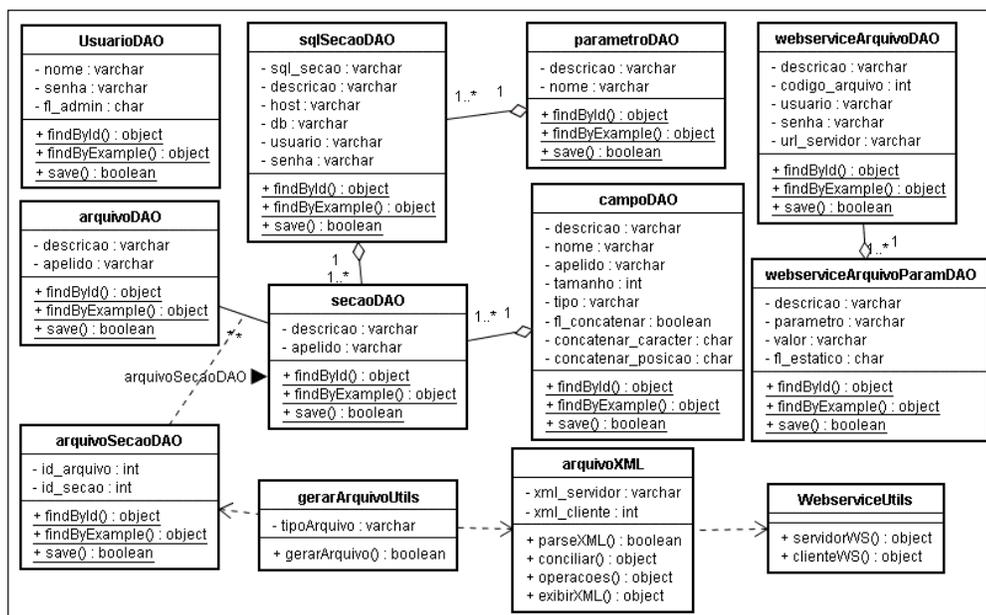


Figura 17: Diagrama de classe

4.3 Modelagem Entidade Relacional (ER)

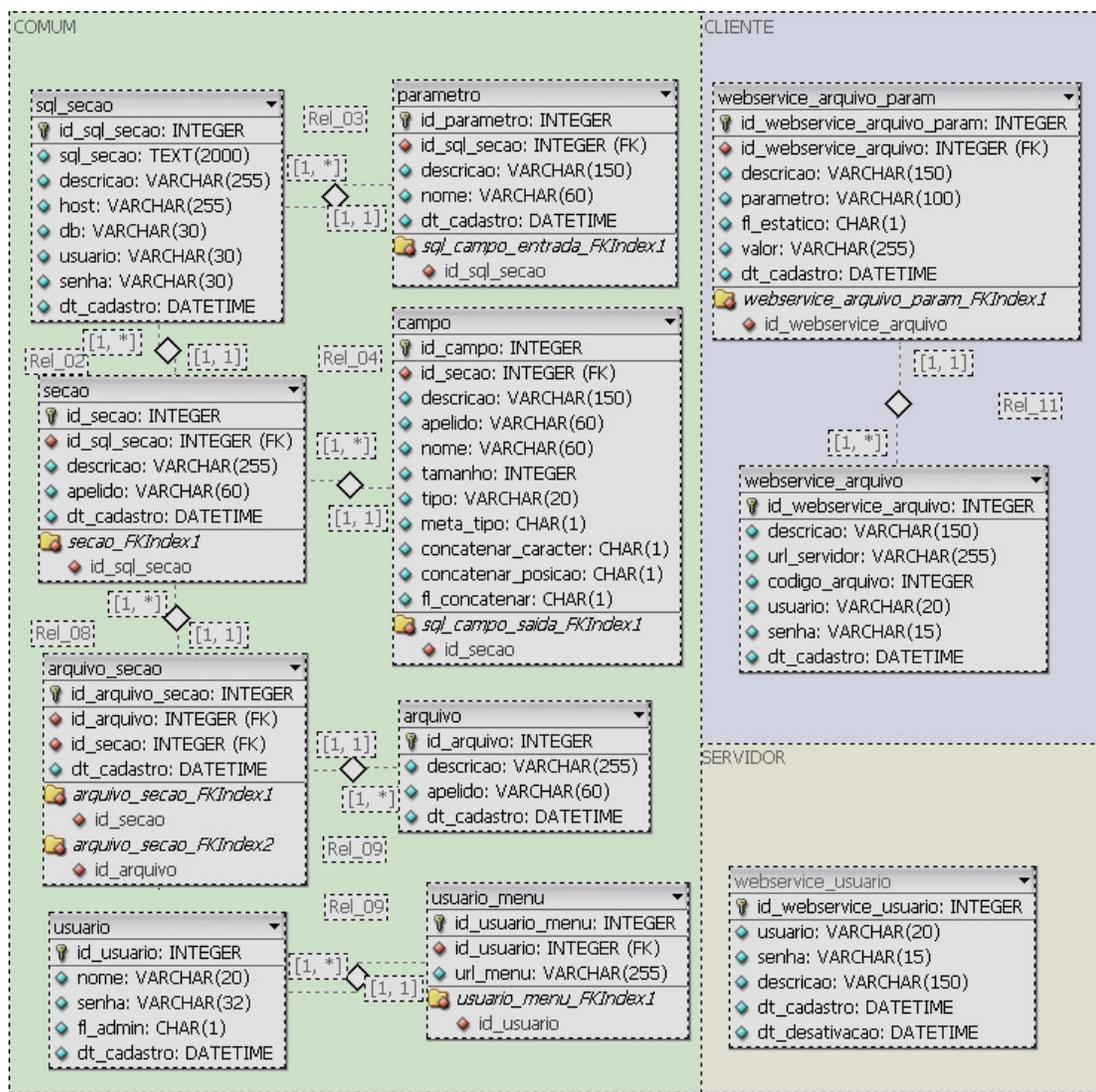


Figura 18: Modelo ER

4.4 Padrões de Projeto (Patterns)

O desenvolvimento do **SISCONT** foi feito a partir do levantamento de requisitos e dos diagramas UML. A arquitetura do sistema foi dividida em camadas adotando a padronização MVC (*Model-View-Controller*). Dentro da camada *model* foi utilizada a padronização DAO (*Data Access Object*) para classes de comunicação com o banco de dados. Nas subseções seguintes serão apresentados esses padrões.

4.4.1 Padrão MVC

O **SISCONT** utilizou a padronização MVC (*Model-View-Controller*) para desenvolver as camadas de modelos de classes, controladores e apresentação.

MVC é uma padronização de arquitetura de software que visa dividir o sistema em camadas separadas de forma que uma não interfira na outra e assim podendo ser desenvolvidas distintamente.

É comum essa divisão do sistema em camadas separadas: apresentação (interface), domínio e acesso a dados. MVC tem três camadas distintas: *Model* (Modelo – classes auxiliares da aplicação), *View* (Visão – templates de exibição) e *Controller* (Controlador – chamadas do sistema) (REVISTAPHP, 2007) e (PHPPATTERNS, 2008).

4.4.2 Padrão DAO

O **SISCONT** utilizou o padrão DAO (*Data Access Objects*) para abstrair a comunicação com a base de dados facilitando a realização das operações básicas.

O DAO (*Data Access Objects*) possibilita a abstração de acesso à base de dados de forma que implementa alguns métodos que realizam as operações básicas como consultas (*Select*), inserções (*Insert*), atualizações (*Update*) e exclusões (*Delete*). A camada de negócios executa as funções disponibilizadas pela DAO, sem necessariamente precisar saber como tal função foi implementada. Os metadados, de uma tabela x qualquer, são mapeados em um objeto (VO – *Value Objects*). Esse objeto, no momento que é chamada uma função, é enviado para a DAO onde internamente trabalha sobre o objeto realizando alguma operação na base de dados (PHPPATTERNS, 2007) e (SUN, 2007).

4.5 Testes e Homologação

Os testes foram realizados no término de cada incremento. Realizaram-se testes unitários para cada módulo (funcionalidade) do sistema e testes do conjunto avaliando o contexto das funcionalidades no todo.

Foram realizados ainda testes de comunicação entre as aplicações cliente e servidora, testando o serviço de *Web Service* em função da segurança e envio dos arquivos.

Foram realizados alguns testes de segurança do sistema, precavendo-se contra injeções de SQL e outros.

Dentro dos testes realizados, foram encontrados alguns problemas e corrigidos. Não foram aplicados, ainda, testes de *stress* e de desempenho. Logicamente foram realizados vários testes combinando várias possibilidades de erros, porém testes específicos de desempenho serão efetuados posteriormente, na continuidade desse trabalho.

A homologação do sistema por enquanto foi realizada somente internamente (versão *alpha*). Após a homologação dos usuários (versão *beta*) será realizado o último dos três incrementos mencionados anteriormente. Nesse último incremento será “ajustado” o sistema para sua primeira versão estável, estando homologado para uso.

5 TECNOLOGIAS

O **SISCONT** foi desenvolvido para atuar sobre a *Web* não importando o sistema operacional. O ambiente *Web* do servidor deve estar configurado e habilitado para suporte PHP. A única dependência no PHP é a biblioteca SOAP a qual deve estar disponível para o uso de *Web Services*. É importante frisar que todas as tecnologias usadas são do tipo *Software Livre*, reduzindo o custo final do sistema. Todas as tecnologias escolhidas estão entre as mais utilizadas no mercado atual.

As seguintes tecnologias foram utilizadas:

- a) servidor Web Apache;
- b) a linguagem de programação PHP 5 + OO com suporte a *Web Services*;
- c) HTML, CSS, JavaScript e DOM;
- d) AJAX;
- e) XML;
- f) Smarty;
- g) ADOdb.

5.1 Apache

O Apache é um projeto que faz parte da Apache Software Foundation, é um servidor *Web* robusto e, segundo seu próprio site, o mais usado no mundo.

O Apache é o responsável pelo recebimento das requisições vindas do cliente, processamento e re-envio dessas informações. Seu código-fonte é disponível gratuitamente na *Web*, é modular e altamente configurável. Esse possibilita que as requisições sejam “escutadas” e atendidas, no caso foi instalado o suporte ao módulo do PHP para o processamento das requisições.

5.2 PHP 5 + OO e Suporte a Web Services

O PHP permite a criação de páginas *Web* dinâmicas de forma simples e poderosa. “Atualmente, existem várias ferramentas para desenvolvimento de sites web dinâmicos, tais como Java, ASP e tantas outras, mas não se encontra ainda uma ferramenta tão completa, estável, simples mas ao mesmo tempo potente como o PHP.” (SOARES, 2004, p. 11).

Segundo Gutmans; Bakken e Rethans (2005) o PHP se faz presente em mais de um terço dos servidores *Web* de todo o mundo tornando-se assim mundialmente conhecido tendo cada vez mais empresas aderindo ao PHP para rodar seus sistemas.

A versão 5 do PHP trouxe uma série de melhorias que deixavam a desejar nas versões anteriores, dentre as principais melhorias o suporte a grande maioria dos conceitos de OO, uma classe simpleXML para manipulação de documentos XML de forma simples e rápida e que será usada para manipulação dos arquivos gerados em XML, reestruturação da API DOM que agora está em conformidade a padronização da W3C (órgão regulador) e ainda a melhoria do suporte ao SOAP para desenvolvimento de *Web Services* (GUTMANS;BAKKEN e RETHANS, 2005).

Os *Web Services* estão sendo amplamente utilizados no mundo dos *software* devido ao fato de possibilitarem a interoperabilidade e comunicação entre aplicações de forma transparente. O conjunto de tecnologias utilizadas pelos *Web Services* faz com que exista uma comunicação de forma organizada e independente entre diferentes tipos de aplicações e plataformas. São baseados em padrões abertos. Podem ser identificados através de uma url onde suas interfaces podem ser descritas e descobertas WSW3C(2008). Basicamente possui as seguintes características:

- a) especificação dos serviços: WSDL (*Web Service Description Language*) que é a linguagem descritora baseada em XML. Essa descreve as funcionalidades disponibilizadas pelo serviço, onde está localizada e como poderá ser acionada;
- b) mensagens: As mensagens são escritas em XML sendo autodescritivas;
- c) transporte: O protocolo SOAP fornece a infraestrutura para a comunicação e transporte das mensagens através de estruturas XML sendo independente de plataforma e linguagem de programação e normalmente utilizado sobre protocolo HTTP.

De acordo com Streibel (2005) a arquitetura dos *Web Services* tem auxiliado na implementação de novas funcionalidades e ou integração entre sistemas legados, de forma a se ter um baixo custo, principalmente para aqueles que possuem tecnologias mais antigas. Sabendo disso, o **SISCONT** utilizou-se dessa tecnologia para a integração entre as aplicações cliente e servidora.

5.3 DHTML (HTML, CSS, JavaScript e DOM)

DHTML segundo a W3C é a combinação de tecnologias *Web* utilizadas para criação de *websites* dinâmicos. As tecnologias usadas são HTML, CSS, JavaScript e DOM. “A combinação dessas tecnologias permite que os desenvolvedores alterem o conteúdo e a estrutura de uma página *Web* dinamicamente” (ASLESON; SCHUTTA, 2006, p. 10).

Segundo a W3C (2007) para publicação de dados ou informações globalmente, faz-se necessário uma linguagem que também seja compreendida globalmente, o HTML é a linguagem conhecida mundialmente para publicações na *Web*. Através do HTML é possível publicar documentos *on-line* a partir dos inúmeros elementos suportadas pela linguagem.

Para simplificar a codificação HTML e melhorar o design e aparência dos conteúdos *on-line* foram criadas as folhas de estilos CSS que permitem a declaração diretamente nos elementos HTML ou ainda para grupos de elementos podendo ser declarados no próprio documento HTML ou externamente.

JavaScript é a linguagem de programação interpretada que auxilia no lado do cliente para interação com usuário realizando validações de formulários e interagindo com as páginas HTML, o código pode ser incluído direto nas páginas HTML ou através de arquivos externos.

DOM é uma especificação da W3C que permite percorrer a estrutura de um documento HTML em forma de árvore podendo excluir, adicionar ou mudar atributos de um elemento, o próprio elemento ou ainda o texto dessa estrutura. Dessa maneira fica fácil realizar interações com JavaScript e DOM, um exemplo pode ser o carregamento de um formulário onde podemos montar toda a estrutura visual e preencher o conteúdo, usando DOM e JavaScript, percorrendo os elementos e atribuindo os valores a eles.

5.4 AJAX

Até pouco tempo atrás páginas HTML só poderiam exibir o conteúdo de maneira síncrona, dessa forma uma simples alteração no conteúdo HTML ocasionava o reenvio de todas as informações do servidor para o cliente gerando demora e tráfego desnecessário, e o que é pior, informações essas que o cliente já havia recebido anteriormente.

O AJAX (*Asynchronous Javascript And XML*) veio para suprir essa necessidade de se realizar alterações assincronamente sem precisar alterar todo o conteúdo HTML. O AJAX é mais uma técnica do que uma tecnologia nova, tanto que desenvolvedores não precisam aprender uma nova tecnologia ou ainda deixar de usar o as que já sabiam. É uma abordagem do lado do cliente que pode interagir com qualquer linguagem *Web* independentemente do servidor (ASLESON e SCHUTTA, 2006).

O AJAX, então “[...] é um mecanismo para separar o dado da aplicação [...] proporcionando mais velocidade, além de aplicações web mais limpas, claras, mais fáceis de desenvolver e gerir [...]” (SOARES, 2006, p. 25), assim ficando a aplicação a cargo do *browser*¹¹ e manipulação dos dados por conta do servidor.

5.5 XML

O XML foi desenvolvido para ser uma linguagem universal para comunicação entre diferentes plataformas e transmissão de dados de uma forma comum. Fornece mecanismos para estruturar e rotular elementos de forma única (GUTMANS; BAKKEN e RETHANS, 2005).

Para realizar a leitura um arquivo XML, deve-se realizar o *parsing*¹² e assim retirar informações do mesmo. No PHP5 existe uma extensão chamada simpleXML que conforme o site da própria linguagem PHP (2007) “provê um conjunto de ferramentas muito fácil e simples de usar para converter XML em objetos que podem ser processados normalmente com seletores de propriedades e iteradores de arrays.”. A estrutura da extensão simpleXML

¹¹ Browser: Software do lado do cliente que recebe e envia informações aos servidores e exibe para o usuário final.

¹² Parsing: É o processo de analisar uma seqüência de entrada (arquivo ou teclado). Os dados são analisados e armazenados, normalmente em forma de árvore, para uso posterior.

não é tão complicada quanto uma API DOM bastando acessar o XML a partir de uma representação da estrutura de dados. Segundo Gutmans, Bakken e Rethans (2005, p. 150) existem quatro simples regras para acessar os dados do arquivo XML com a extensão simpleXML:

1. Propriedades denotam iteradores de elementos.
2. Índices numéricos denotam elementos.
3. Índices não-numéricos denotam atributos.
4. Conversão de strings permite o acesso a dados de TEXTO.

5.6 Smarty

A biblioteca Smarty vai separar o PHP de outros códigos como: HTML, JavaScript, CSS etc. Além de melhorar a visibilidade do código PHP, ela possibilita que o programador se abstraia do *layout*, focando puramente nas regras de negócio. No projeto serão desenvolvidas tanto as regras de negócios quanto os *layout's* (formatos dos arquivos) juntos, mas em grandes empresas isso poderia ser desenvolvido distintamente sem problemas.

A primeira vez que um arquivo *template* do Smarty é compilado ele gera um arquivo PHP e enquanto esse *template* não for alterado não é necessário uma re-compilação, sendo assim o *template* não necessita ser parseado (analisado) a todo instante tornando o processo mais rápido. Outras características interessantes do Smarty são as funções e modificadores de variáveis que existem por *default* e que também podem ser criadas pelo usuário conforme a necessidade facilitando assim trabalhar com dados vindos do PHP (SMARTY, 2007).

5.7 ADOdb

Com uso da biblioteca ADOdb a aplicação não ficará restringida a somente um banco de dados específico, ela possibilita então a abstração de conexão e operações a banco de dados. A biblioteca ADOdb permite criar a conexão com diversos bancos, tais como: MySQL, PostgreSQL, Firebird, Oracle, Access, Sybase entre outros (ADOdb, 2007).

Possui métodos próprios que manipulam os resultados de forma que a saída ou a entrada seja sempre da mesma forma indiferente do banco de dados selecionado. A ADOdb vai possibilitar, por exemplo, que o **SISCONT** na “empresa A” busque informações no MySQL e na “empresa B” concilie realizando uma conexão no Oracle.

6 RESULTADOS

Como resultado do estudo e desenvolvimento desse trabalho, utilizando os conceitos de EAI, foi desenvolvido um sistema *Web* sob uma aplicação cliente e servidora, utilizando uma biblioteca para abstração de banco de dados (ADOdb) e tecnologias atuais (PHP, AJAX, XML, etc), implementando um *Web Service* para comunicação entre as duas aplicações. O sistema permite realizar a integração entre sistemas de organizações parceiras visando a extração, validação e conciliação de dados providos dos bancos de dados das mesmas, através do uso e implementação de *middleware* que auxiliaram para a realização dessa integração.

A seguir será apresentado o sistema desenvolvido e suas principais funcionalidades. Para melhor entendimento, essa apresentação de resultados foi dividida em quatro etapas, que serão apresentadas nas seções seguintes:

- 1) *layout* padrão para arquivos;
- 2) funcionalidades comuns às aplicações;
- 3) funcionalidades específicas da aplicação servidora;
- 4) funcionalidades específicas da aplicação cliente.

6.1 *Layout* Padrão para Arquivos

Para facilitar na integração, os arquivos transmitidos foram padronizados em um *layout* específico. Para arquivos XML foram especificadas as *tags* e os atributos para facilitar a leitura do mesmo, já para os arquivos de texto TXT (ASCII) não existem *tags*, mas sim uma formatação linear dos dados por registro no arquivo.

A Figura 19 mostra o *layout* para o qual deve ser gerado o arquivo XML:

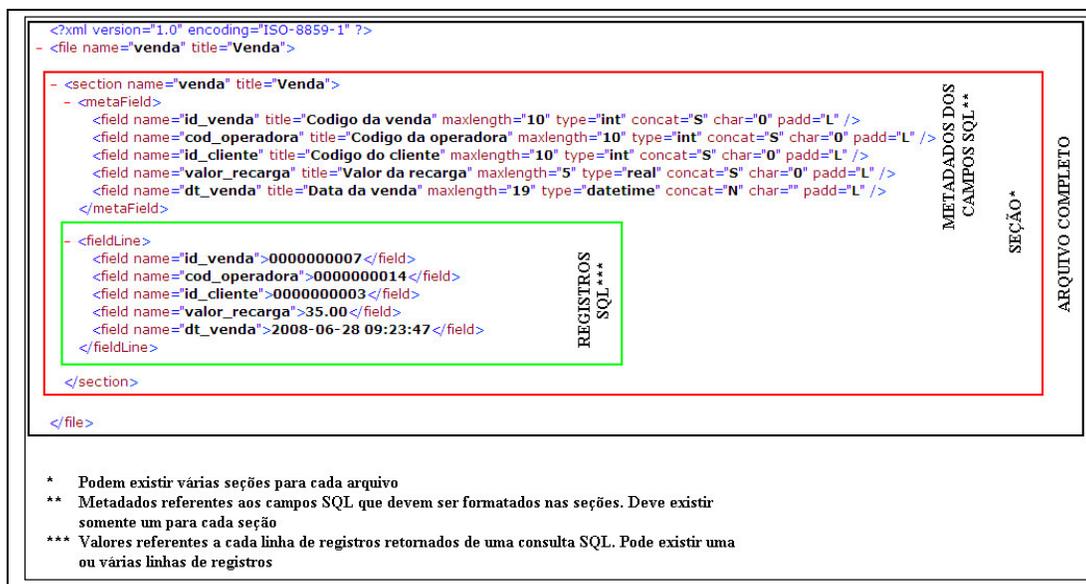


Figura 19: Layout padrão do arquivo de dados no formato XML

Descrição para *tags* do *layout* XML iniciando da mais externa até a mais interna:

Arquivo completo

- <file>: Tag raiz do arquivo;
 - atributo name: Referencia o nome do arquivo;
 - atributo title: Informa o título do arquivo;

Seções

- <section>: Referencia cada seção dentro do arquivo, podem existir uma ou mais seções;
 - atributo name: Referencia a seção dentro do arquivo;
 - atributo title: Título informado a cada seção do arquivo;

Metadados dos campos SQL

- <metaField>: Referencia os metadados dos campos SQL de cada seção. Pode existir somente um para cada seção;
 - <field>: Referencia os metadados de cada campo do SQL. Este é referenciado pelo seu atributo “name” que deve ser único;

- atributo name: Referencia o campo SQL para poder encontrar os metadados desse campo. Este atributo é único e deve ter o mesmo valor do atributo “name” da tag “field” do SQL;
- atributo title: Informa o título para o campo;
- atributo maxlength: Informa o tamanho máximo do campo;
- atributo type: Informa o tipo primitivo do campo;
- atributo concat: Informa se o campo deve ser concatenado com o caracter especificado no atributo “char” e em qual sentido apontado pelo atributo “padd”. Se o valor é “S” concatena, caso contrário não concatena.
- atributo char: Informa o caracter a ser concatenado caso o atributo “concat” esteja com o valor “S”;
- atributo padd: Informa em qual sentido deve ser concatenado os caracter especificado no atributo “char”. Possíveis valores “L” esquerda ou “R” direita.

Registros SQL

- <fieldLine>: Referente a cada linha de registros vindos na consulta SQL, podem existir um ou mais registros;
- <field>: Referente a cada campo vindo da consulta SQL;
 - atributo name: Nome referenciando a cada campo vindo da consulta SQL.

A Figura 20 apresenta a estrutura do arquivo validador do XML Schema (XSD – *XML Schema Definition*) utilizado no **SISCONT**.

```

Validator XML Schema (XSD)
- <xs:schema>
- <xs:element name="file">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="section" maxOccurs="unbounded">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="metaField">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="field" minOccurs="1" maxOccurs="unbounded">
- <xs:complexType>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="title" type="xs:string" use="required"/>
<xs:attribute name="maxlength" type="xs:integer" use="required"/>
<xs:attribute name="type" type="xs:string" use="required"/>
- <xs:attribute name="concat" use="required"/>
- <xs:simpleType>
- <xs:restriction base="xs:string">
<xs:pattern value="S|N"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
- <xs:attribute name="char" use="required">
- <xs:simpleType>
- <xs:restriction base="xs:string">
<xs:maxLength value="1"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
- <xs:attribute name="padd" use="required">
- <xs:simpleType>
- <xs:restriction base="xs:string">
<xs:pattern value="L|R"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
- <xs:element name="fieldLine" minOccurs="1" maxOccurs="unbounded">
- <xs:complexType>
- <xs:sequence>
- <xs:element name="field" minOccurs="1" maxOccurs="unbounded">
- <xs:complexType>
- <xs:simpleContent>
- <xs:extension base="xs:string">
<xs:attribute name="name" type="xs:string" use="required"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="title" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="title" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:schema>

```

Figura 20: Validador XML Schema (XSD)

Para os arquivos gerados no formato TXT (ASCII) não existe um *layout* específico, existem somente duas regras, os campos terão tamanhos iniciados na posição *x* e finalizados na posição *y* (conforme seu tamanho máximo - *maxlength*) e cada linha deverá ter o mesmo

tamanho para cada seção. Da mesma forma que o arquivo XML, serão referenciadas também suas seções, estarão disponíveis sempre nos primeiros caracteres antes dos dados em cada linha do arquivo. A quantidade de caracteres utilizada vai depender da formatação cadastrada na tela de seções.

A exportação de arquivos de texto TXT foi necessária para dar suporte a aplicações antigas, que realizam funções específicas dentro da organização e que foram implementadas de forma que lessem esses arquivos. Dessa maneira o **SISCONT** dá suporte a criação nesse formato de arquivo.

Abaixo está a Figura 21 que mostra o mesmo arquivo, antes exibido no *layout* XML, agora no formato de texto TXT.



Figura 21: *Layout* padrão do arquivo de dados no formato TXT

6.2 Funcionalidades Comuns às Aplicações

Nas subseções seguintes serão apresentadas as funcionalidades comuns as aplicações cliente e servidora.

6.2.1 Tela de login e tela inicial das aplicações

Abaixo está a Figura 22 que mostra a tela de login e a Figura 23 que mostra a tela inicial das aplicações.

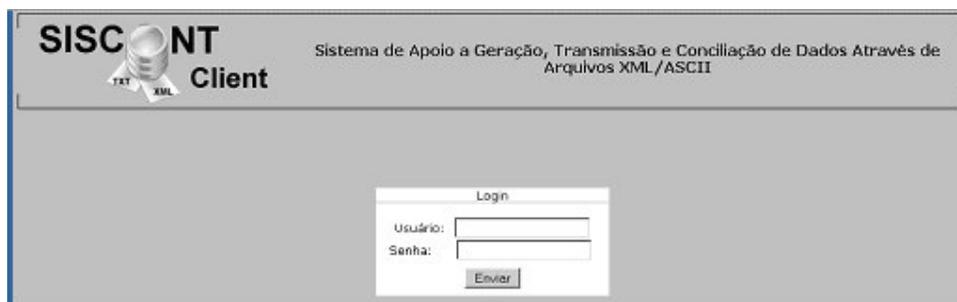


Figura 22: Telas de login



Figura 23: Telas de login

6.2.2 Cadastro de SQL

O administrador do sistema pode cadastrar um SQL, que será usado por uma ou mais seções que farão parte de um arquivo. Deve entrar com a descrição, os dados para conexão e o

SQL a ser executado. O SQL pode ter ou não parâmetros, caso tenha de ser inserida a validação para o parâmetro da seguinte maneira:

```
Select [...] from id = ${nome_para_valor_variavel} [...].
```

Caso o campo seja do tipo string devesse ser usado entre aspas:

```
Select [...] from nome = “${nome_para_valor_variavel}” [...].
```

Caso existam parâmetros, deve clicar no botão “Carregar Parâmetros” e cadastrar a descrição para os mesmos.

A Figura 24 mostra o cadastro de uma consulta SQL.

Figura 24: Cadastro de SQL

6.2.3 Cadastro de seção

A partir de um SQL cadastrado, é possível cadastrar uma seção. O administrador seleciona o SQL que irá fazer parte da seção e caso exista parâmetros vai solicitar a inserção dos mesmos. Após, o **SISCNT** vai buscar os metadados dos campos da tabela no banco conforme cadastrados no SQL, exibindo os mesmos para efetuar a formatação. O

administrador deverá informar os dados para a seção e a formatação esperada para cada campo conforme Figura 25.

Figura 25: Cadastro de seção

6.2.4 Cadastro de arquivo

O administrador deve informar quantas seções irão compor o arquivo. Após deverá inserir os dados para o arquivo e selecionar as seções nos combos exibidos conforme Figura 26.

Após gravar, será exibida uma mensagem com o código do arquivo. Esse código será usado posteriormente pelo cliente para solicitar o arquivo via *Web Service*.

The screenshot displays the SISCONT Client web application. At the top, the header contains the logo 'SISCONT Client' and the subtitle 'Sistema de Apoio a Geração, Transmissão e Conciliação de Dados Através de Arquivos XML/ASCII'. Below the header, a navigation menu on the left lists 'Home', 'SQL', 'Seção', 'Arquivo', and 'Usuário'. The main content area is titled 'Arquivo' and contains the following elements:

- A 'Logout' button next to the user name 'angeli'.
- A 'Quantidade de seções:' label followed by an input field containing the value '1' and a 'Corrigir' button.
- A 'Descrição:' label followed by an input field containing the text 'Arquivo de dados clientes'.
- An 'Apelido:' label followed by an input field containing the text 'arquivo_dados_clientes (Tag para relatório. Ex: nome_tag)'.
- A 'Seções do arquivo' section with a 'Seção 1:' label and a dropdown menu currently showing 'Dados clientes'.
- A 'Cadastrar' button at the bottom of the form.

At the bottom of the page, a footer reads 'Todos os direitos reservados - SISCONT by Juliano F. Angeli'.

Figura 26: Cadastro de arquivo

6.2.5 Gerar Arquivo

Após um cadastro de arquivo, o mesmo já poderá ser gerado. O usuário vai escolher em qual formato gerar (XML ou TXT). Caso tenham sido cadastrados parâmetros, os mesmos serão solicitados antes de gera-los. O SISCONT vai realizar as consultas SQL cadastradas para cada seção do arquivo realizando a formatação dos mesmos conforme *layout* apresentados anteriormente. Após, o processo vai exibir o ícone para realizar o *download* conforme Figura 27.

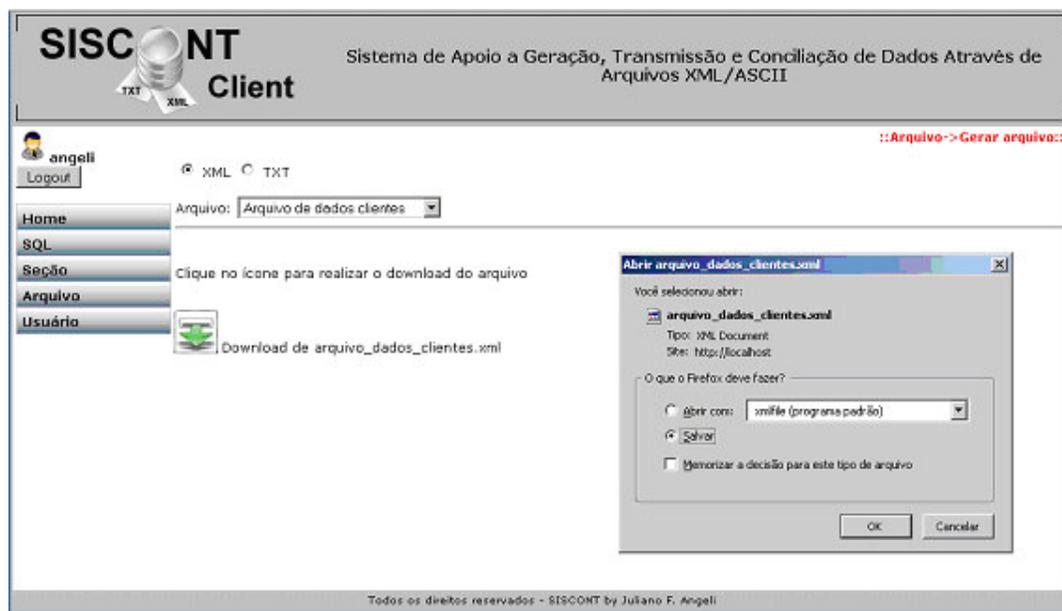


Figura 27: Gerar arquivo

6.3 Funcionalidades Específicas da Aplicação Servidora

Na aplicação servidora, além das funcionalidades comuns, existe ainda a funcionalidade de cadastro de usuários para requisição de arquivos via *Web Service* descrita na subseção a seguir.

6.3.1 Cadastro usuário para Web Service

O administrador da aplicação servidora poderá cadastrar os usuários que terão acesso aos arquivos de dados disponibilizados. Para isso basta informar a descrição, o usuário e a senha conforme figura 28.

The screenshot shows the SISCNT Server web interface. At the top, the logo 'SISCNT Server' is displayed next to a database icon, with the text 'Sistema de Apoio a Geração, Transmissão e Conciliação de Dados Através de Arquivos XML/ASCII'. Below the logo, there is a navigation menu with options: 'Logout', 'Home', 'SQL', 'Seção', 'Arquivo', and 'Usuário'. The 'Usuário' option is selected. To the right of the menu, there is a registration form with the following fields and buttons:

- Description: 'Usuário de teste' (input field)
- User: 'teste' (input field)
- Password: 'senha' (input field)
- 'Cadastrar' button

In the top right corner, there is a red link: '::Usuário->Web Service->Cadastrar::'. At the bottom of the page, there is a footer: 'Todos os direitos reservados - SISCNT by Juliano F. Angeli'.

Figura 28: Cadastra usuário para Web Service

6.4 Funcionalidades Específicas da Aplicação Cliente

Na aplicação cliente, além das funcionalidades comuns, existem outras quatro funcionalidades que serão comentadas nas próximas subseções.

6.4.1 Cadastro do arquivo disponível pela aplicação servidora

O administrador poderá cadastrar os arquivos, disponíveis pela aplicação servidora, onde vai informar inicialmente se existem parâmetros necessários para o mesmo. Após deverá informar uma descrição, o código do arquivo, o usuário e a senha.

Deverão ser informados também os parâmetros, caso sejam necessários, cadastrando a descrição, o nome do parâmetro e se é estático ou não. Caso seja estático, deverá ser informado o valor.

Os campos do código do arquivo, usuário e senha do WS e o nome do parâmetro a ser cadastrado devem ser informados pela organização onde está a aplicação servidora, esses são

fundamentais para sucesso na requisição do arquivo. Assim será possível realizar a autenticação no WS e solicitar o arquivo. A Figura 29 mostra a tela para cadastro do arquivo.

The screenshot displays the SISCNT Client interface. At the top, it reads 'SISCNT Client' and 'Sistema de Apoio a Geração, Transmissão e Conciliação de Dados Através de Arquivos XML/ASCII'. A user profile for 'angeli' is visible with a 'Logout' button. A breadcrumb trail shows the path: '::Arquivo->Servidor->Cadastrar arquivo:'. The main content area is titled 'Arquivo do servidor' and includes a 'Quantidade de parâmetros' field set to '1' and a 'Carregar' button. Below this, there are input fields for 'Descrição' (containing 'Dados dos clientes'), 'Código do arquivo' (containing '13'), 'Usuário' (containing 'teste'), and 'Senha'. A 'Parâmetros do arquivo do servidor' section contains a table with one row: '1 Descrição parâmetro: Atividade do cliente', 'Nome parâmetro: f_ativo', and 'Estático: Não'. A 'Valor do parâmetro' field is also present. A 'Cadastrar' button is at the bottom of the form. The footer contains the text 'Todos os direitos reservados - SISCNT by Juliano F. Angel'.

Figura 29: Cadastro do arquivo disponível pela aplicação servidora

6.4.2 Buscar arquivo da aplicação servidora

O usuário poderá buscar um arquivo XML, disponibilizado pela aplicação servidora, já cadastrado anteriormente. Para isso deve selecionar o arquivo e informar os parâmetros caso sejam necessários. Após o processamento, será exibido um ícone para o *download* conforme mostra a Figura 30.

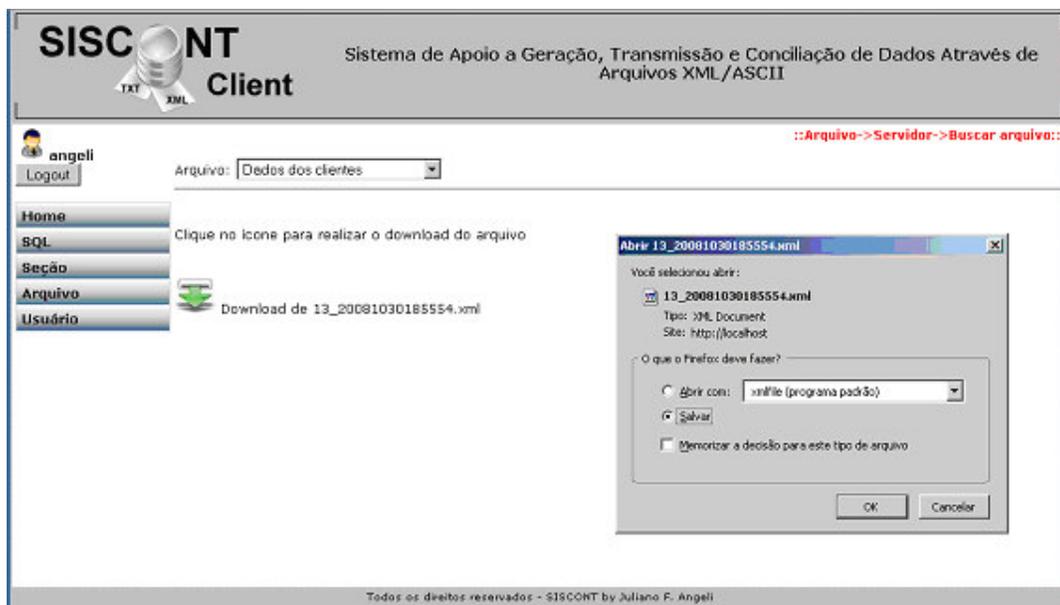


Figura 30: Buscar arquivo da aplicação servidora

6.4.3 Ler arquivo XML

O usuário vai poder ler um arquivo XML, que esteja formatado no *layout* especificado anteriormente, exibindo em tela na forma de um relatório. Para isso, basta selecionar o arquivo, após clicar em “ler” conforme mostra a Figura 31.

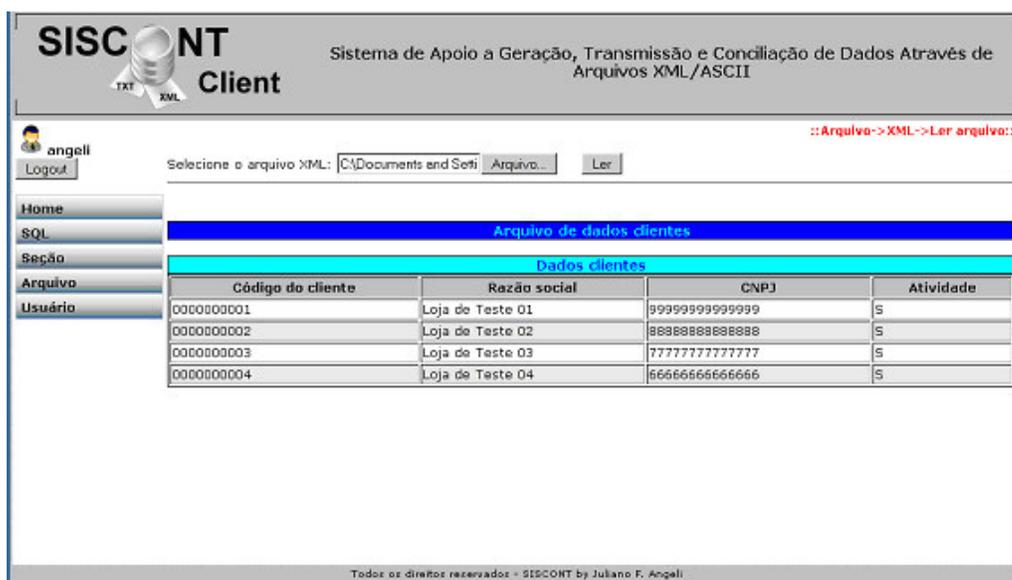


Figura 31: Ler arquivo XML

6.4.4 Conciliar arquivo XML

O usuário pode conciliar dois arquivos XML, que estejam formatados no *layout* especificado anteriormente. Para isso deve selecionar dois arquivos XML e clicar em conciliar.

O **SISCONT** vai avaliar se existem divergências nos arquivos, existindo vai processar e armazenar as informações no banco de dados. Nesse processo é criada uma tabela dinamicamente, onde os dados são inseridos seqüencialmente, separados de acordo com a aplicação (servidora ou cliente), seção e campos de cada linha. Para facilitar e possibilitar uma maior performance, cada linha de dados é agrupada e gerada uma *hash md5*. Após esse processo, são listadas na tela as seções existentes em cada arquivo, o nome da tabela que foi criada dinamicamente, e as operações que podem ser executadas sobre os registros. Essas operações são baseadas em consultas SQL previamente cadastradas. É possível disponibilizar quantas operações (consultas SQL) forem necessárias. A disponibilidade de novas operações, por enquanto, devem ser feitas pelo desenvolvedor (programador) do sistema. Futuramente será disponibilizada a ferramenta para que o próprio administrador do sistema consiga disponibilizar uma nova operação. A Figura 32 mostra um exemplo das operações que podem ser executadas sobre os dados na conciliação.



Figura 32: Exemplos de operações que podem ser executadas sobre os dados na conciliação

Quando selecionada uma operação, dependendo da operação selecionada, pode retornar os registros ou não. Os registros serão exibidos caso a operação (consulta SQL) seja satisfeita. A Figura 33, mostra um exemplo de conciliação possibilitada pela operação “Consultar todos os registros diferentes”.

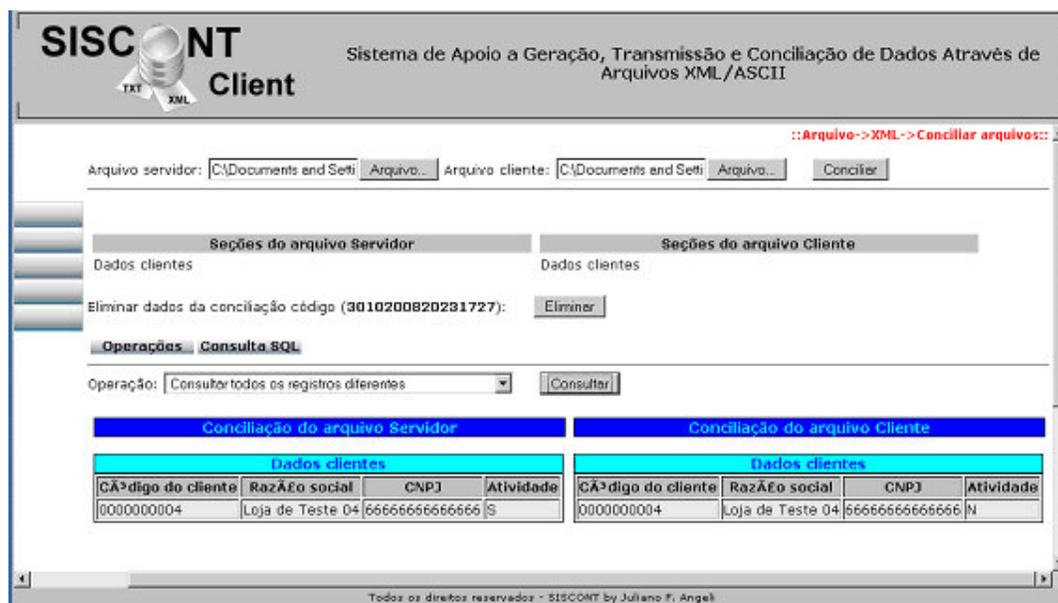


Figura 33: Exemplos de operação executada sobre os dados exibindo a conciliação

Quando o usuário for o administrador, além das operações, é disponibilizado o recurso de realizar a consultas SQL (*on-line*) sobre os registros. O administrador pode elaborar o SQL necessário e executa-lo na mesma ferramenta conforme mostra a Figura 34 e 35.



Figura 34: Exemplos de consulta SQL executada de forma *on-line* pelo administrador

SISC ONT Client
Sistema de Apoio a Geração, Transmissão e Conciliação de Dados Através de Arquivos XML/ASCII

Consultar Restaurar valores

Resultado da consulta

sequencia	aplicacao	section	hashField	fieldLine
1	S	Dados clientes	d5d6624a1e5f05dd72f2f114f392df22	CÃdigo do cliente 000000001#RazÃEo social Loja de Teste 01#CNPJ 9999999999999#Atividade S
2	S	Dados clientes	2a34b9738bc47fa445115e0f5a830475	CÃdigo do cliente 000000002#RazÃEo social Loja de Teste 02#CNPJ 8888888888888#Atividade S
3	S	Dados clientes	4351f91182a409b6310e6e7c75234782	CÃdigo do cliente 000000003#RazÃEo social Loja de Teste 03#CNPJ 7777777777777#Atividade S
4	S	Dados clientes	e77e0c398365a2593e0de4050bb5120f	CÃdigo do cliente 000000004#RazÃEo social Loja de Teste 04#CNPJ 6666666666666#Atividade S
5	C	Dados clientes	d5d6624a1e5f05dd72f2f114f392df22	CÃdigo do cliente 000000001#RazÃEo social Loja de Teste 01#CNPJ 9999999999999#Atividade S
6	C	Dados clientes	2a34b9738bc47fa445115e0f5a830475	CÃdigo do cliente 000000002#RazÃEo social Loja de Teste 02#CNPJ 8888888888888#Atividade S
7	C	Dados clientes	4351f91182a409b6310e6e7c75234782	CÃdigo do cliente 000000003#RazÃEo social Loja de Teste 03#CNPJ 7777777777777#Atividade S
8	C	Dados clientes	df0c5695bb98dc5a8013b3737e349ff6	CÃdigo do cliente 000000004#RazÃEo social Loja de Teste 04#CNPJ 6666666666666#Atividade N

Todos os direitos reservados - SISC ONT by Juliano F. Angeli

Figura 35: Resultado de consulta SQL executada de forma *on-line* pelo administrador

Após realizar as operações necessárias para conciliar os dados contidos nos arquivos, o usuário pode eliminar a tabela que foi criada dinamicamente. Caso não seja eliminada, por motivos de consultas posteriores, a consulta deverá ser feita somente via banco de dados.

7 CONCLUSÃO

O emprego de EAI, juntamente com suas tecnologias norteadoras, tem possibilitado a integração entre os SI's das organizações. Muitos pesquisadores e organizações tem percebido a importância de integrar sistemas, sendo inclusive apontada, essa integração, como essencial para a sobrevivência de muitas organizações.

Assim como EAI é uma tendência e, por muitas vezes, uma forma das organizações sobreviverem e se adequarem ao mercado, também muitas delas entenderam que devem focar os seus esforços no desempenho das tarefas que tenham haver com a finalidade da organização. Sabendo disso, muitas das tarefas antes desempenhadas dentro da organização foram entregues a terceiros. Hoje, a terceirização das tarefas que não fazem parte da finalidade das organizações, na sua maioria, já estão sendo entregues a terceiros. Assim, as organizações têm mais liberdade para focar no seu negócio.

Sabendo disso, não seria incorreto dizer que o mercado precisa de soluções que atendam a demandas de sistemas apoiando e auxiliando no negócio entre organizações parceiras. Sistemas que possam de alguma forma interagir e integrar SI de organizações parceiras são de grande utilidade e ainda podem oferecer uma visão de futuro, pois esse tipo de negócio é promissor.

O trabalho apresentado propôs e implementou o **SISCONT** visando essa integração entre organizações parceiras. O sistema visa suprir uma necessidade freqüente para muitas empresas. Apoiou a integração de sistemas diferentes possibilitando a extração de dados de bancos de dados das organizações envolvidas, implementando mecanismos para transmissão desses dados através de arquivos via *Web*. Além disso, ainda disponibilizou uma ferramenta onde será possível validar e conciliar esses dados, facilitando o negócio entre as organizações.

O **SISCONT** foi desenvolvido de forma que fosse flexível, podendo atender a outras necessidades de integração entre diversos tipos de organizações. Visionou-se não limitar a ampliação dos serviços que podem ser disponibilizados através do sistema. A utilização de tecnologias atuais como *Web Service*, bibliotecas para abstração a bancos de dados, aplicação cliente e servidora, padronizações, e outros, possibilitaram tal flexibilidade. De certa forma, essa preocupação pode possibilitar a maturidade do sistema, tornando-se uma ferramenta essencial dentro das organizações. A utilização das ferramentas e novas necessidades de implementação darão forma ao sistema que deve auxiliar no desempenho das organizações.

Os testes realizados demonstram, por enquanto, que todas as funcionalidades e objetivos propostos foram alcançados.

REFERÊNCIAS

ADODB. **Biblioteca para abstração de banco de dados**. Disponível em: <<http://adodb.sourceforge.net/>>. Acessado em: 01 dez. 2007.

APACHE. **Servidor Web da Apache Software Foundation**. Disponível em: <http://httpd.apache.org/ABOUT_APACHE.html>. Acesso em: 27 nov. 2007.

ASLESON, Ryan; SCHUTTA, Nathaniel T.. **Fundamentos do Ajax**. Rio de Janeiro: Editora Alta Books, 2006.

BAGUETE, 2008. Brasil direciona 41% de investimentos em TI para serviços. **Baguete**. mar. 2007. Disponível em: <<http://www.baguete.com.br/noticiasDetalhes.php?id=17060>>. Acessado em: 20 set. 2008.

BALIEIRO, Silvia. Terceirização representa 30% do mercado de TI. **Info CORPORATE**. São Paulo, jun. 2008. Disponível em: <<http://info.abril.com.br/corporate/noticias/062008/20062008-3.shtml>>. Acesso em: 27 ago. 2008.

BARRETO, Aldo de Albuquerque. A eficiência técnica e econômica e a viabilidade de produtos e serviços de informação. **Revista IBICT, Ciência da Informação**, Vol. 25, n. 3, 1996. Disponível em: <<http://revista.ibict.br/index.php/ciinf/article/view/466>>. Acessado em: 01 ago. 2008.

BOGUI, C.; SHITSUKA, R.. **Sistemas de Informação: Um enfoque dinâmico**. São Paulo: Editora Érica, 2002.

CASSARRO, Antonio Carlos. **Sistemas de informações para tomada de decisões**. 3. ed. São Paulo: Editora Pioneira, 1999.

CHARLESWORTH, Ian; JONES, Teresa. The Web Services and EAI Report. **EAI Journal**, mar. 2003. Disponível em: <<http://www.alignjournal.com/index.cfm?section=article&aid=182#>>. Acessado em: 10 ago. 2008.

CHAVES, Leonardo Grandinetti. EAI: novo acrônimo tecnológico I, II e III. **TI Master: Revista TI**, fev. 2002. Disponível em: <http://www.timaster.com.br/revista/artigos/main_artigo.asp?codigo=517&pag=2>. Acessado em: 05 set. 2008.

DHTML W3C. **DHTML: W3C Schools**. Disponível em:
<<http://www.w3schools.com/dhtml/default.asp>>. Acesso em: 27 nov. 2007.

DHTML W3C. **HTML: Portuguese translations of W3C documents (Português)**. Disponível em: <<http://www.w3.org/2005/11/Translations/Lists/ListLang-pt.html#html401-0>> e <<http://desenaviegas.com/intro.html#h-2.2>>. Acesso em: 27 nov. 2007.

DHTML W3C. **CSS: Portuguese translations of W3C documents (Português)**. Disponível em: <<http://www.w3.org/2005/11/Translations/Lists/ListLang-pt.html#html401-0>> e <<http://www.maujor.com/w3c/craccss.html>>. Acesso em: 27 nov. 2007.

DHTML W3C. **JAVASCRIPT: W3C Schools**. Disponível em:
<http://www.w3schools.com/js/js_intro.asp>. Acesso em: 27 nov. 2007.

DHTML W3C. **DOM: W3C**. Disponível em: <<http://www.w3.org/DOM/>>. Acessado em: 01 dez. 2007.

DICIONÁRIO. MICHAELIS: **Dicionário da língua Portuguesa on-line**. Disponível em:
<<http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=concilia%E7%E3o>>. Acesso em: 05 nov. 2007.

DICIONÁRIO. MICHAELIS: **Dicionário da língua Portuguesa on-line**. Disponível em:
<<http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=terceirizar>>. Acesso em: 03 nov. 2007.

FERNANDEZ, Fernando Henrique. **Discussão de um modelo conceitual para Enterprise Application Integration – EAI**. 2004. 97 f. Dissertação (Mestrado em Engenharia da Computação) – Universidade Estadual de Campinas, Campinas, 2004.

FURLAN, José Davi. **Modelagem de objetos através da UML – the Unified Modeling Language – Análise e desenho orientados a objeto**. São Paulo: Editora Makron Books, 1998.

FRANCO, Roni de Oliveira. O Outsourcing e a Sustentabilidade. **Administradores: O Portal da Administração**, nov. 2007. Disponível em:
<http://www.administradores.com.br/noticias/o_outsourcing_e_a_sustentabilidade/12951/>. Acessado em: 20 set. 2008.

GANDRA, Alana. Terceirização explica crescimento de até 6% nos serviços prestados a empresas. **EBC Empresa Brasil de Comunicação: Agência Brasil**, Rio de Janeiro, fev. 2008. Disponível em: <<http://www.agenciabrasil.gov.br/noticias/2008/02/13/materia.2008-02-13.7263340885/view>> e <<http://www.agenciabrasil.gov.br/noticias/2008/02/13/materia.2008-02-13.4016795198/view>>. Acessado em: 27 set. 2008.

GARCIA, Simone de Souza; SHINOTSUKA, Tania Hiromi. **EAI – Enterprise Application Integration**, Rio de Janeiro, 2001. Disponível em: <<http://www.bax.com.br/Bax/orientacao/alunos/PSI/EAI1.pdf>>. Acessado em: 27 ago. 2008.

GIOSA, Lívio Antonio. **Terceirização: Uma abordagem estratégica**. 5 ed. São Paulo: Editora Pioneira Thomson Learning, 2003.

GONÇALVES, Nilton Oliveira. **Terceirização de mão-de-obra**. São Paulo: Editora LTr, 2005.

GUTMANS, Andi; BAKKEN, Stig Saether; RETHANS, Derick. **PHP 5: Programação poderosa**. Rio de Janeiro: Editora Alta Books, 2005.

LEIRIA, Jerônimo Souto; SARATT, Newton Dorneles. **Terceirização: uma alternativa de flexibilidade empresarial**. São Paulo: Editora Gente, 1995.

LIMA, Márcia. Outsourcing: R\$ 15,2 bi em 2008. **Baguete**, 2008. Disponível em: <<http://www.baguete.com.br/noticiasDetalhes.php?id=24885>>. Acessado em 20 set. 2008.

MARRAS, Jean Pierre. **Administração de recursos humanos do operacional ao estratégico**. 3. ed. São Paulo: Editora Futura, 2000.

MARTINS, José Carlos Cordeiro. **Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML**. Rio de Janeiro: Editora Brasport, 2004.

MORESI, Eduardo Amadeu Dutra. **Delineando o valor do sistema de informação de uma organização**, 2000. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652000000100002&lng=en&nrm=iso&tlng=pt> ou <<http://www.scielo.br/pdf/ci/v29n1/v29n1a2.pdf>>. Acessado em: 21 ago. 2008.

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas de informações gerenciais: estratégicas, táticas, operacionais**. 6. ed. São Paulo: Editora Atlas, 1999.

PHP. **Extensão simpleXML**. Disponível em: <http://www.php.net/manual/pt_BR/ref.simplexml.php>. Acessado em: 01 dez. 2007.

PHPPATTERNS. **Design Patterns: MVC**. Disponível em: <http://www.phppatterns.com/docs/design/archive/model_view_controller_pattern>. Acessado em 01 dez. 2007.

PHPPATTERNS. **Design Patterns: DAO**. Disponível em: <http://www.phppatterns.com/docs/design/data_access_object_pattern_more_widgets?s=dao>. Acessado em 01 dez. 2007.

PRESSMAN, Roger S.. **Engenharia de software**. São Paulo: Editora Makron Books, 1995.

PRESSMAN, Roger S.. **Engenharia de software**, 5. ed.. Rio de Janeiro: Editora McGraw-Hill, 2002.

REVISTAPHP. **Arquitetura MVC**. Disponível em:
<<http://www.revistaphp.com.br/artigo.php?id=50>>. Acessado em : 02 dez. 2007.

SMARTY. **Sistema de templates para o PHP**. Disponível em:
<http://smarty.php.net/manual/pt_BR/what.is.smarty.php>. Acessado em: 01 dez. 2007.

SOARES, Wallace. **PHP 5: Conceitos, programação e integração com banco de dados**. São Paulo: Editora Érica, 2004.

SOARES, Wallace. **AJAX (Asynchronous JavaScript And XML): guia prático para Windows**. São Paulo: Editora Érica, 2006.

SORDI, José Osvaldo de; MARINHO, Bernadete de Lourdes. Integração entre sistemas: Análise das abordagens praticadas pelas corporações brasileiras. CONGRESSO ANUAL DE TECNOLOGIA DA INFORMAÇÃO, 2006. **ANAIS ELETRÔNICOS CATI 2006**. Disponível em: <<http://www.fgvsp.br/CATI/index.cfm?FuseAction=anais> >. Acessado em: 20 ago. 2008.

STOKES, Nigel. **EAI and Beyond: A Multilevel Flow Model**. President and CEO DataMirror Corporation. Disponível em: <http://www.grcdi.nl/eai_and_beyond.pdf >. Acessado em: 21 set. 2008.

STREIBEL, Martin. **Implementando Web Services**. 2005 Monografia. Disponível em: <<http://palazzo.pro.br/artigos/martin.htm>>. Acessado em: 27 mar. 2008.

SUN. **Core J2EE Patterns -Data Access Object**. Disponível em:
<<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>>. Acessado em: 01 dez 2007.

WSW3C. **W3C: Web Services Architecture**. Disponível em:
<<http://www.w3.org/TR/ws-arch>>. Acesso em: 10 jan. 2008.