

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA - CEFET/RJ

DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
COORDENADORIA DO PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA

DISSERTAÇÃO

UTILIZAÇÃO DA ARQUITETURA ORIENTADA AOS SERVIÇOS PARA A
INTEGRAÇÃO DE SISTEMAS DE INFORMAÇÃO HETEROGÊNEOS

Leonardo de Carvalho Vidal

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE PÓS-
GRADUAÇÃO EM TECNOLOGIA COMO PARTE DOS REQUISITOS NECESSÁRIOS
PARA A OBTENÇÃO DO GRAU DE MESTRE EM TECNOLOGIA.

Marina Rodrigues Brochado, D.Sc.
Orientadora

Rui Manuel Alves da Silva e Sousa, Ph.D.
Co-Orientador

RIO DE JANEIRO, RJ - BRASIL
MARÇO / 2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

SUMÁRIO

	Pág.
INTRODUÇÃO	1
CAPÍTULO I- REVISÃO BIBLIOGRÁFICA	6
I.1- Evolução das Organizações e a Tecnologia da Informação	6
I.1.1- Estruturas Organizacionais	7
I.1.2- Arquiteturas Organizacionais	9
I.1.3- Arquitetura de TI	12
I.1.4- Arquitetura Corporativa	15
I.2- Evolução dos Sistemas de Computação	18
I.2.1- Arquiteturas de Software	18
I.2.2- Integração de Sistemas	19
I.2.3- Arquitetura de Software Orientadas a Serviços	20
I.3- A Importância da Inovação: Abordagem Teórica	21
I.3.1- Estágios para Atingir a Inovação	22
I.3.2- Modalidades de Inovação	23
CAPÍTULO II- A ARQUITETURA ORIENTADA AOS SERVIÇOS (SOA)	24
II.1- Os WEB Services	25
II.2- Evolução da Tecnologia - WEB Services	27
II.2.1- O World Wide Web Consortium (W3C)	27
II.2.2- Software Aberto <i>versus</i> Software Proprietário	27
II.3- Componentes da Arquitetura	29
II.3.1- Extensible Markup Language (XML)	29
II.3.2- Simple Object Access Protocol (SOAP)	30
II.3.3- Web Services Description Language (WSDL)	34
II.3.4- Enterprise Service Bus (ESB)	35
II.3.5- Universal Description, Discovery and Integration (UDDI)	36
II.3.6- Business Process Management (BPM)	37
II.4- Valor Agregado dos Web Services	38
CAPÍTULO III- MODELO PARA IMPLEMENTAÇÃO E GESTÃO DA SOA COMO INOVAÇÃO PARA A INTEGRAÇÃO DE SISTEMAS DE INFORMAÇÃO	41
III.1- Modelo de Gestão da Inovação	41
III.2- Competências de Gestão da Inovação	42
III.2.1- Gestão Estratégica	42

III.2.2- Gestão de Projetos	44
III.2.3- Gestão Funcional	45
III.2.4- Gestão da Mudança	46
III.3- Ciclo de Implementação e Gestão da SOA	47
III.3.1- Fase de Modelagem	48
III.3.2- Fase de Montagem	49
III.3.3- Fase de Implementação	49
III.3.4- Fase de Gerenciamento	49
III.4- Modelo Proposto para a Implementação e Gestão da SOA	50

CAPÍTULO IV- ESTUDO DE CASO: INTEGRAÇÃO DE SISTEMAS

HETEROGÊNEOS DE DUAS EMPRESAS DE VAREJO UTILIZANDO A SOA	52
IV.1- Necessidade de Negócio	52
IV.2- Fase de Levantamento e Definição das Alternativas	52
IV.3- Aplicação da Metodologia	56
IV.3.1- Gestão Funcional e de Mudança	56
IV.3.2- Gestão de Projeto	58
IV.3.3- Gestão Estratégica	60
IV.3.4- Fase de Montagem	64
IV.3.3- Fase de Implementação	65

CONCLUSÃO	68
------------------	-----------

REFERÊNCIAS BIBLIOGRÁFICAS	70
-----------------------------------	-----------

Às mulheres de minha vida, em ordem de aparecimento:
minha queridas mãe e irmã, minha esposa e filha.

Agradecimentos

- Aos Professores Marina Rodrigues Brochado (D.Sc.) e Rui Manuel Alves da Silva e Sousa (Ph.D.), pelo empenho no trabalho de orientação, dedicação e incentivo que muito contribuíram para a elaboração deste trabalho.
- Ao Professor Antonio José Caulliriaux Pithon (Ph.D.), pela grande contribuição.
- Ao Professor Vinícius Cardoso (D.Sc.), da UFRJ, pelas discussões e sugestões.
- Aos alunos de iniciação científica do CEFET-RJ: Sérgio Varga, André Ricardo e Gustavo Spritz, pela colaboração.

Resumo da dissertação submetida ao PPTEC/CEFET-RJ como parte dos requisitos necessários para a obtenção do grau de mestre em tecnologia (M.T.).

UTILIZAÇÃO DA ARQUITETURA ORIENTADA AOS SERVIÇOS PARA A INTEGRAÇÃO DE SISTEMAS DE INFORMAÇÃO HETEROGÊNEOS

Leonardo de Carvalho Vidal

Março de 2007

Orientadora: Marina Rodrigues Brochado, D.Sc.

Co-Orientador: Rui Manuel Alves da Silva e Sousa, Ph.D.

Programa: PPTEC

As constantes mudanças e evolução da tecnologia da informação requer das organizações flexibilidade para mudar ou criar processos de negócios pO

Abstract of dissertation submitted to PPTEC/CEFET/RJ as partial fulfillment of the requirements for the degree of Master in Technology (M.T.).

SERVICE ORIENTED ARCHITECTURE FOR HETEROGENEOUS INFORMATION SYSTEMS INTEGRATION

Leonardo de Carvalho Vidal

March / 2007

Supervisor: Marina Rodrigues Brochado, D.Sc.

Co-Supervisor: Rui Manuel Alves da Silva e Sousa, Ph.D.

Program: PPTEC

Constant business process changes and fast technology evolution requires from the organizations flexibility to quickly create and change business processes, in order to gain a competitive advantage. But the traditional enterprise systems are mostly rigid, monolithic, centralized, not based on open standards and non up-to-date with new methodologies, unable to support this new reality. Such business requirements are creating new challenges in the process of information systems integration, demanding a new flexible and adaptable software environment in order to achieve such requirements. This work presents a study on the use of Service Oriented Architecture (SOA) and the use of Web Services in order to achieve seamless integration and reutilization of existing software, increase collaboration with business partners and suppliers, and propose the alignment between information technology strategy and business strategy. A referring bibliographical revision was used as methodology, covering the evolution of the organizations, organizational architectures and information technology, focusing on the evolution of computer systems and systems integration. As case study, a business-oriented processes modeling was carried through SOA, and its application in the integration of heterogeneous systems of two retail companies using Web Services. The use of SOA allowed the encapsulation of the existing systems and the integration of its functionalities as services to be reused and accessed through open standards.

Keyword: SOA, Enterprise Architecture, Web Services, IT.

Lista de Tabelas

	Pág.
Tabela I.1- Arquitetura em silos <i>versus</i> arquitetura corporativa. Fonte: LIGUORI (2006).	16
Tabela I.2- Comparação: a empresa tradicional e a empresa inovadora. Fonte: FREIRE (2002).	21
Tabela III.1- Controle dos projetos de inovação. Adaptada de FREIRE (2002).	44
Tabela III.2- Reações negativas à mudança pela inovação. Adaptada de FREIRE (2002).	46
Tabela III.3- Estágios para adoção da SOA na organização. Fonte: GANCI et al. (2006).	48
Tabela IV.1- Aplicações de negócio da Empresa A.	53
Tabela IV.2- Aplicações de negócio da Empresa B.	53
Tabela IV.3- Comparação entre alternativas e parâmetros de implementação.	55
Tabela IV.4- Gestão funcional do projeto.	57
Tabela IV.5- Interfaces a serem implementadas como serviços.	64

Lista de Figuras

	Pág.
Figura Introdução.1- Estrutura da dissertação.	2
Figura I.1- Estrutura organizacional piramidal.	8
Figura I.2- Estrutura organizacional descentralizada por unidades de negócio. Fonte: RODRIGUEZ e FERRANTE (1995).	9
Figura I.3- Modelo organizacional. Fonte: TUSHMAN e NADLER (1993).	10
Figura I.4- Exemplo do mapa das aplicações e suas interfaces em uma organização. Fonte: GORDON (2006).	14
Figura I.5- Modelo da arquitetura corporativa. Fonte: LIGUORI (2006).	16
Figura I.6- Evolução das arquiteturas de software. Fonte: MARKS e WERRELL (2003).	19
Figura II.1- Estrutura básica da mensagem SOAP. Fonte: ROBICHAUX (2006).	31
Figura II.2- Requisição de um serviço através do SOAP.	33
Figura II.3- O <i>Enterprise Service Bus</i> (ESB).	35
Figura II.4- Processo de descoberta de serviços via UDDI.	37
Figura II.5- Funcionamento de uma ferramenta BPM.	38
Figura II.6- Custos associados à implementação de sistemas. Fonte: ATHENA (2006).	39
Figura III.1- Ciclo de inovação. Fonte: FREIRE (2002).	41
Figura III.2- Modelo de gestão da inovação. Fonte: FREIRE (2002).	42
Figura III.3- Ciclo de implementação e gestão da SOA. Fonte: GANCI et al. (2006).	47
Figura III.4- Metodologia proposta para a implementação e gestão da SOA.	51
Figura IV.1- Sistema Flash de Vendas.	54
Figura IV.2- Sistema Catálogo de Vendas.	54
Figura IV.3- Visualização das etapas do projeto.	59
Figura IV.4- Modelagem do fluxo do processo de pedidos de cliente.	61
Figura IV.5- Modelagem do fluxo do processo de verificação dos pedidos.	62
Figura IV.6- Modelagem do fluxo do processo de pagamento.	63
Figura IV.7- Função GETPRODUCTAVAIL existente na aplicação Catálogo de Vendas.	65
Figura IV.8- Método GETPRODUCTAVAIL implementado na aplicação Catálogo de Vendas.	66
Figura IV.9 - Arquivo WSDL gerado para o método GETPRODUCTAVAIL.	66

Abreviaturas e Símbolos

Abreviatura / Símbolo	Significado	Unidade
TI	Tecnologia da Informação	-
SOA	Service Oriented Architecture	-
CEFET-RJ	Centro Federal de Educação Tecnológica Celso Suckow da Fonseca- Rio de Janeiro	-
HTML	Hyper Text Markup Language	-
SOAP	Simple Object Access Protocol	-
XML	Extensible Markup Language	-
W3C	Word Wide Web Consortium	-
WSDL	Web Services Description Language	-
TCP/IP	Transmission Control Protocol / Internet Protocol	-
OSS	Open Source Software	-
B2B	Business to Business	-
EA	Enterprise Architecture	-
BPEL	Business Process Execution Language	-
WWW	Word Wide Web	-
ESB	Enterprise Service Bus	-
UDDI	Universal Description, Discovery and Integration	-
DCOM	Distributed Component Object Model	-
CORBA	Common Object Request Broker Architecture	-
RMI	Remote Method Invocation	-
API	Application Programming Interface	-

INTRODUÇÃO

As transformações tecnológicas vêm ao longo dos séculos produzindo radicais modificações nas organizações e na sociedade como um todo, de forma cada vez mais acentuada. Dentro deste contexto, a revolução da informação vem gradualmente fazendo evoluir a sociedade industrial para uma sociedade da informação, sendo a tecnologia da informação (TI) o alicerce desta transformação. Como consequência, atualmente o peso do sistema econômico produtivo é cada vez mais centrado no fator informação, assim apontando outras características que o desenvolvimento das tecnologias da informação veio confirmar como interação em redes, a globalização, etc.

Assim, o valor informacional vem desencadeando a necessidade das organizações reverem suas estratégias competitivas, seja na geração de novos produtos ou serviços, bem como nas suas relações com clientes e fornecedores; enfim, todo o seu ambiente competitivo. O mercado, competitividade, qualidade, retenção e conquista de clientes fazem parte da agenda permanente das empresas produzindo uma busca incessante pela melhoria contínua de seus processos e revisão de suas práticas gerenciais. A adoção de soluções independentes geradas a partir de orientações departamentais gera na maioria das organizações um complexo mapa dos sistemas e de suas interfaces, conforme descrito por GORDON (2006).

Contudo, tal realidade compromete em muito o resultado destas empresas pela impossibilidade de, baseado em dados coletados e armazenados de forma adequada, planejar-se, avaliar-se e executar-se as atividades inerentes a realização de seus negócios. A dificuldade em interpretar, monitorar e tratar as informações de forma colaborativa impõem sobre estas organizações severas dificuldades para adaptar-se às mudanças, prejudicando seus resultados e colocando-as em uma posição desfavorável em relação aos seus concorrentes.

Tradicionalmente, em tais estruturas descentralizadas, uma determinada área de negócio solicita ao departamento de TI um determinado sistema para atender suas novas necessidades. Este analisa e qualifica as necessidades, desenvolvendo interna ou externamente um conjunto de programas que irão atender todas as necessidades do problema apresentado, os dados de entrada e saída, além da plataforma (*hardware* e *software*) na qual os novos programas irão ser executados, conforme descrito por HURWITZ J. et al. (2006). Seja qual for a arquitetura que tenha sido definida e implementada por TI, a maior parte (ou totalidade) das estruturas são criadas para atender àquela determinada necessidade, contendo inúmeras particularidades. Desta forma, não foram originalmente criadas para serem conectadas às outras aplicações de negócio existentes. Esta orientação é denominada por muitos autores como arquitetura de TI orientada a silos.

Sob o ponto de vista de TI, as soluções desenvolvidas sem uma perspectiva corporativa podem implicar em grande esforço de integração, falta de suporte aos requerimentos funcionais e não-funcionais, dificuldade de gerência e operação e conseqüentemente um maior investimento, em uma análise de longo prazo. Essa visão global permite um entendimento da arquitetura de TI atual, ou seja, um mapeamento da infra-estrutura de *hardware* e de *software*, das aplicações e das suas interfaces, dos usuários dos sistemas, da governança e dos processos de negócio.

Mas percebe-se que os sistemas tradicionais de informação empresariais são, na generalidade dos casos, rígidos, monolíticos, centralizados, fechados, dependentes de tecnologia “proprietária” e baseados em metodologias desatualizadas e não adequadas às novas realidades. É hoje crescente a necessidade de dispor de ferramentas para apoiar paradigmas organizacionais com elevados requisitos de flexibilidade, de cooperação e de coordenação, em detrimento das soluções de TI orientadas às áreas funcionais (silos) das organizações.

A questão apresentada é semelhante à mudança de paradigma ocorrida no começo dos anos 90 do século XX que moveu os sistemas monolíticos dos mainframes para uma arquitetura cliente-servidor e distribuída. Hoje, a necessidade de uma maior integração e reutilização de *software* já existentes, para sustentar o aumento da colaboração com parceiros e fornecedores, em um contexto de alinhamento da estratégia de negócio com a estratégia de TI das organizações, tem-se a gradativa mudança da arquitetura distribuída para a Arquitetura Orientada a Serviços (SOA), através da implementação dos Web Services. Ambiente no qual os Web Services estão se posicionando como a plataforma padrão para os novos projetos de desenvolvimento e integração de sistemas, combinada com a emergência dos padrões abertos.

Conforme descrito por LAURINDO et al. (2001), ter uma clara visão estratégica para o negócio e para a TI é ponto de partida para que seja possível um impacto significativo no desempenho da organização. A TI precisa ser vista como um meio para a empresa obter vantagens competitivas no mercado em que atua, alinhada com a operação e a estratégia da empresa.

Portanto, a hipótese levantada neste trabalho é integração o modelo de arquitetura de gestão corporativa com as abordagens de planejamento da arquitetura da TI sob o enfoque da introdução de inovação tecnológica: A visão da TI como arma estratégica competitiva tem sido discutida e enfatizada, pois não só sustenta as operações de negócio existentes, mas também permite que se viabilizem novas estratégias empresariais através da implementação dos Web Services.

Objetivos

Geral:

O presente trabalho tem como objetivo estudar a utilização da Arquitetura Orientada a Serviços (SOA) para integração de sistemas de informação heterogêneos em ambiente de colaboração entre empresas de comércio varejista.

Específicos:

- Investigar a bibliografia sobre os sistemas de informações gerenciais;
- Investigar a evolução dos sistemas de computação Arquitetura Orientada a Serviços (SOA) para integração de sistemas de informação heterogêneos;
- Propor a utilização da SOA para integração de sistemas de informação heterogêneos em ambiente de colaboração entre empresas de comércio varejista.

Metodologia

Para o desenvolvimento da pesquisa, foram adotados os métodos de revisão bibliográfica e estudo de caso descritivo. Conforme descrito por Yin (2005), este é utilizado para descrever uma intervenção e o contexto na vida real em que o fato ocorre. Os estudos de caso representam a estratégia mais adequada quando se colocam questões do tipo como e porquê, quando o pesquisador tem pouco controle sobre os acontecimentos e quando o foco do trabalho se encontra em fenômenos contemporâneos, inseridos em algum contexto da vida real. A Figura 1.0 apresenta o desenho de pesquisa, identificando os métodos selecionados e seu encaixe, de forma a atingir o objetivo proposto.

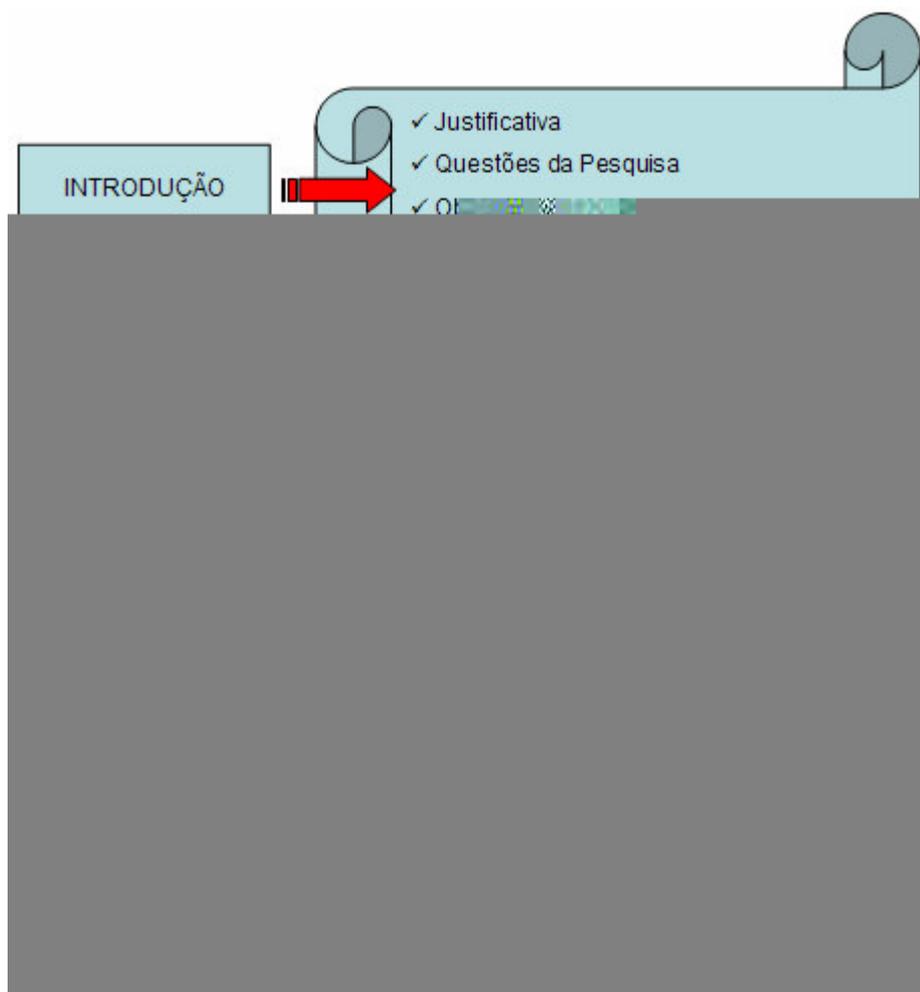


Figura Introdução.1- Estrutura da dissertação.

A pesquisa bibliográfica procura explicar o problema dos desafios de negócios em um novo ambiente operacional sob demanda, com base na SOA (Arquitetura Orientada a Serviços, acrônimo do inglês *Service Oriented Architecture*). Em uma segunda fase, foi realizada uma pesquisa documental que se diferencia da pesquisa bibliográfica pelo fato desta envolver dados primários, ou seja, dados que nunca foram tratados antes com a finalidade de analisar a implementação da SOA como uma inovação dentro das organizações. Ao final, foi proposta um modelo de aplicação da SOA como meio do mapeamento e entendimento dos processos, sistemas, pessoas e tecnologias, viabilizando o trabalho e planejamento conjunto das unidades de negócio e a organização de TI.

Estes dados foram colhidos a partir de técnicas de entrevistas com especialistas, com o objetivo de investigar as necessidades de informação para integração de sistemas de informação heterogêneos em ambiente de colaboração entre empresas de comércio varejista.

Organização do Trabalho

Este trabalho encontra-se estruturado da seguinte forma: no Capítulo 1, é realizada uma revisão bibliográfica acerca da evolução das estruturas e arquiteturas organizacionais e a proposta de uma nova arquitetura corporativa de TI, de forma a alinhar a estratégia de negócio com a estratégia de TI de uma organização. O Capítulo 1 descreve ainda a evolução dos sistemas de computação, as metodologias empregadas para integração de sistemas e a evolução destas para a SOA. No capítulo 2, são analisados os componentes principais e aspectos mais relevantes da SOA, tendo como ferramenta principal desta arquitetura a utilização dos Web Services. No Capítulo 3 realiza-se uma abordagem da implementação da SOA como uma inovação dentro das organizações, propondo um modelo de competências de gestão para implementação da SOA. No Capítulo 4, é realizado um estudo de caso referente à integração de sistemas heterogêneos dentro de duas empresas de varejo que se fundiram através da aquisição da Empresa B pela Empresa A, utilizando como base a metodologia proposta no Capítulo 3. O Capítulo 5 descreve os resultados do estudo, sumário e propõe futuros estudos acerca do tema.

CAPÍTULO I- REVISÃO BIBLIOGRÁFICA

Neste capítulo é realizada uma revisão bibliográfica acerca da emergência da sociedade da informação, a evolução das arquiteturas organizacionais e da TI. Esta revisão também aborda a nova proposta de arquitetura corporativa de TI, como forma de promover dentro das organizações o alinhamento entre a estratégia de negócio com a estratégia de TI. Descreve ainda a evolução dos sistemas de computação, as metodologias empregadas para integração de sistemas e como estas tecnologias evoluíram até à SOA. Ao final do capítulo, é apresentada a importância da inovação como peça fundamental ao desenvolvimento de qualquer organização.

I.1- Evolução das Organizações e a Tecnologia da Informação

A história da humanidade nos apresenta uma contínua evolução. As mudanças tecnológicas são parte desta evolução, e vêm ao longo dos séculos (e de forma cada vez mais acentuada) produzindo radicais transformações nas organizações e na sociedade como um todo. Este foi o caso, por exemplo, da revolução industrial, que alterou radicalmente a organização vigente na época, substituindo a sociedade agrícola por um novo tipo de organização: a sociedade industrial. No último século, ficou claro que a velocidade com que as transformações ocorrem é cada vez mais acentuada. Observa-se que a evolução tecnológica e em particular a revolução da informação vem gradualmente fazendo evoluir a sociedade industrial para uma sociedade da informação, conforme descrito por RODRIGUEZ e FERRANTE (1995).

Historicamente, cada sociedade está baseada em diferentes valores, serviços e agentes de mudança, descritos abaixo:

- 1) Sociedade Agrícola: apresenta três pilares básicos: a terra, a força de trabalho escrava e o capital. Foi representada pelos povos da antiguidade e pela organização das sociedades dos séculos XVII e XVIII. Esta sociedade se estabeleceu e durou por milhares de anos, tendo como agente de mudança a Igreja.
- 2) Sociedade Industrial: apresenta três pilares básicos: os meios de transporte, a energia e a indústria. Neste caso, o modelo de trabalho implementado por Henry Ford encontrou um campo fértil com a especialização e a divisão de tarefas, subdividindo o processo de fabricação em diversas especialidades, cada qual apresentando gerências intermediárias que garantiam o cumprimento das tarefas. Se estabeleceu após a revolução industrial ocorrida na Inglaterra no séc. XVIII e durou cerca de 300 anos (duração esta bem menor do que a primeira), e teve como principal agente de mudança o estado.

- 3) Sociedade da Informação: os pilares desta nova sociedade, que é fortalecida a cada dia, está baseada na tecnologia da informação (TI). Nascida há cerca de 30 anos, possui como principal agente de mudança a empresa. Por definição, tecnologia da Informação compreende todos os recursos tecnológicos envolvidos na armazenagem, tratamento e recuperação de dados, que são então tratados (ou processados) e transformados em informações úteis à sociedade.

I.1.1- Estruturas Organizacionais

Na sociedade industrial, houve uma busca pela redução dos custos operacionais através da produção em larga escala. Esta era gerida através de uma rígida cadeia hierárquica, resultando na criação de empresas do tipo vertical, com tendências de isolamento do mundo exterior. A estrutura organizacional destas empresas é denominada piramidal. Henry Ford tem sido aceito mundialmente como o pioneiro deste tipo de organização empresarial. Conforme descrito por FREEMAN e STONER (1994), seu nome é sinônimo de produção em massa e do desenvolvimento da moderna teoria da administração, tendo-a implementado no início do séc. XX nas linhas de produção dos automóveis Ford.

Esta estrutura foi largamente difundida e eficientemente utilizada ao longo dos anos seguintes, tanto por empresas quanto pela Igreja e por organizações militares. Baseava-se em três diferentes níveis: o nível estratégico (no topo), o nível tático (no meio) e o nível operacional (na base), conforme apresentado na Figura I.1. Esta estrutura operacional rígida espelhava a tecnologia da informação da época, centralizada e baseada em grandes centros isolados de processamento de dados.

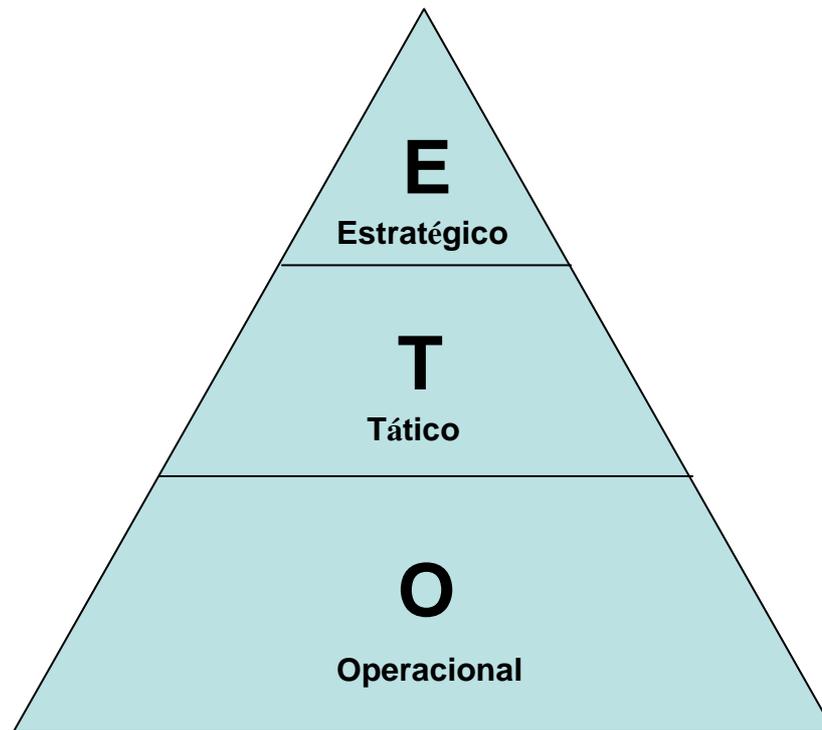


Figura I.1 - Estrutura organizacional piramidal.

A organização do tipo vertical apresentava limitações. Pouca flexibilidade, processos centralizados, isolamento e tomada de decisão exclusiva ao nível estratégico (com pouca ou nenhuma delegação). Acompanhando a tecnologia, o ambiente de negócios é dinâmico, e as regras precisam ser constantemente reescritas. Desta forma, as organizações evoluíram da típica estrutura hierárquica da sociedade industrial para estruturas apresentando uma base operacional mais larga e um reduzido número de níveis hierárquicos. Diminuindo o número de gerências intermediárias, as unidades de negócio passaram a operar com maior autonomia, produzindo constantes e contínuas melhorias operacionais, tomando o primeiro passo rumo à uma filosofia de trabalho baseada em grupos de trabalho integrados porém auto-gerenciáveis (em inglês *workgroups*), conforme ilustrado na Figura I.2, adaptada de RODRIGUEZ e FERRANTE (1995):

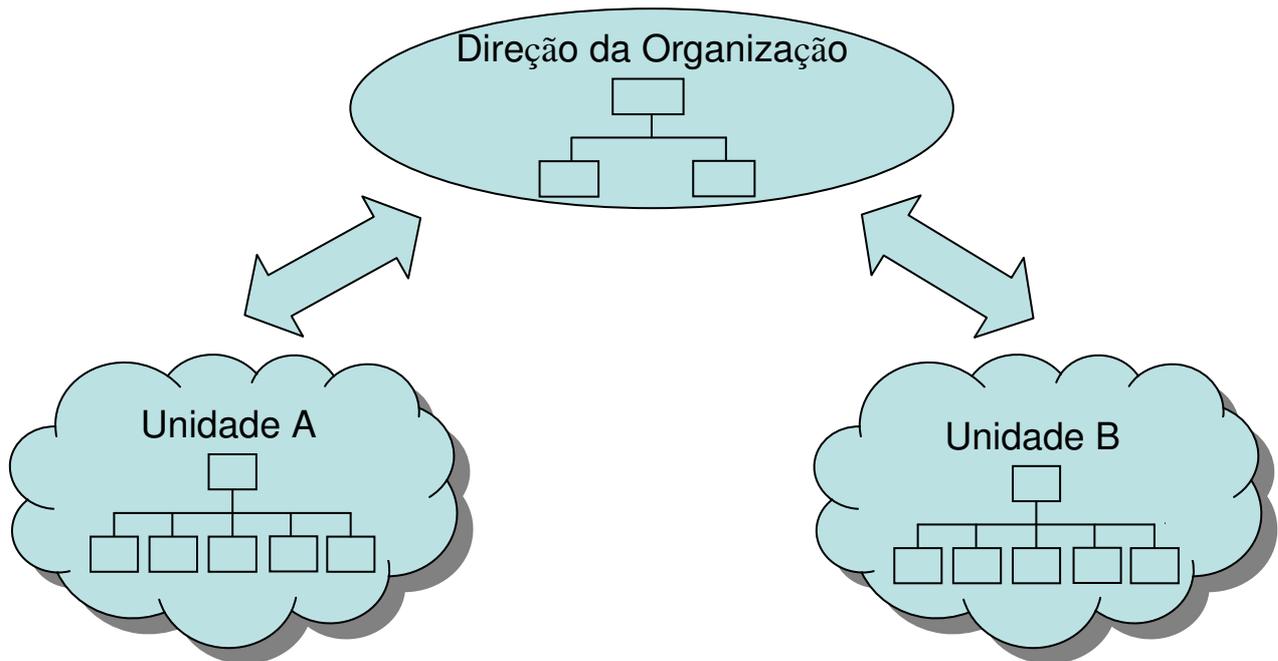


Figura I.2- Estrutura organizacional descentralizada por unidades de negócio. Fonte: RODRIGUEZ e FERRANTE (1995)

Tais grupos de trabalho, embora auto-gerenciáveis, são baseados em objetivos comuns. Numa estrutura deste tipo, torna-se fundamental que haja uma total integração entre os grupos de trabalho, viabilizada através do uso intensivo das modernas tecnologias de informação. Desta forma, a TI evoluiu de uma orientação tradicional de suporte administrativo para um papel estratégico dentro desta nova estrutura organizacional.

I.1.2- Arquiteturas Organizacionais

De acordo com JANSON (1991) apud GERSTEIN (1993), a arquitetura organizacional pode ser definida como “*a arte de modelar o espaço organizacional para satisfazer as necessidades e aspirações humanas*”. Existem duas perspectivas válidas, e que devem ser consideradas a respeito das arquiteturas organizacionais. De um lado, deve se considerar como a arquitetura permitirá à organização executar suas várias estratégias e realizar o trabalho exigido, sendo esta uma perspectiva de desempenho-estratégia-tarefa. Do outro lado, deve se considerar como a arquitetura se harmonizará com os indivíduos que trabalham para a organização, e como os impactará. Esta é denominada perspectiva social-cultural.

A respeito da estrutura funcional, existem ainda maneiras distintas de pensar sobre as organizações. Na visão clássica da administração, existe apenas a estrutura formal, representada pelo organograma da empresa. Este modelo vê apenas as relações estáveis, formais, entre as tarefas e unidades de trabalho como o fator primordial dentro de uma organização. Esta visão (limitada) exclui o comportamento de liderança, o impacto do ambiente, as relações informais e a distribuição de poder. Nas últimas décadas, se observou

um consenso crescente com relação à visão da organização como um sistema social aberto, sendo este definido a partir de uma série de elementos interrelacionados. A mudança em um deles afeta todos os demais. Um sistema aberto é portanto um sistema que interage com o seu ambiente. Nesta abordagem, o sistema recebe um insumo do ambiente, transformando este insumo em produto, conforme descrito na Figura I.3:

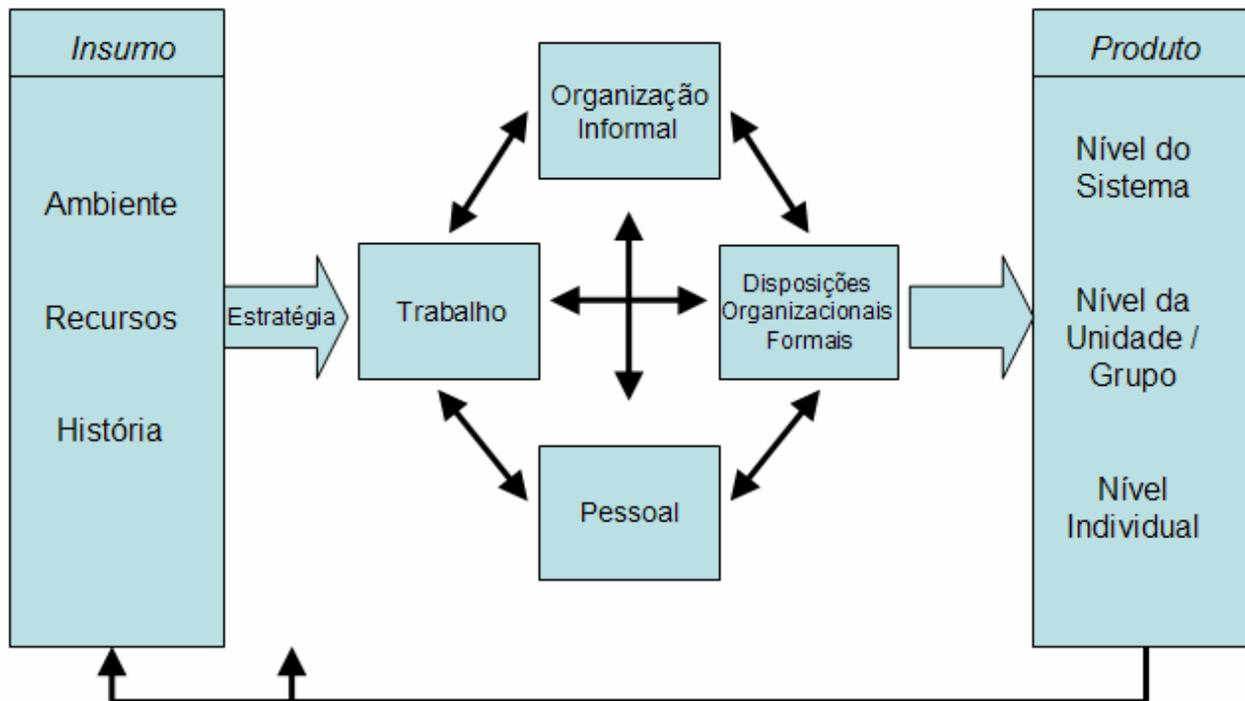


Figura I.3 - Modelo organizacional. Fonte: TUSHMAN e NADLER (1993).

A seguir são detalhados todos os componentes deste sistema.

- 1) Insumo: Os insumos são os elementos componentes do contexto enfrentado pela organização, inclusive o material que ela tem que trabalhar. Os fatores deste contexto são respectivamente:
 - a) Ambiente: inclui mercados, clientes, fornecedores, órgãos governamentais e reguladores, condições tecnológicas, sindicatos, concorrentes, instituições financeiras e grupos de interesse especiais. O ambiente faz constantemente exigências à organização. Por exemplo, o mercado pode exigir determinados produtos ou serviços, certos níveis de qualidade ou quantidade. Além disso, o ambiente pode impor limitações à ação das organizações. Estas variam desde a escassez de capital, matéria-prima e tecnologia até a proibições causadas por regulamentações governamentais ou impostos. Mas o ambiente também oferece oportunidades a serem exploradas pela organização. Quando uma organização é analisada, tais fatores do ambiente devem ser considerados para se determinar novas demandas, limitações e oportunidades;

- b) Recursos: qualquer organização tem acesso a uma série de bens tais como funcionários, tecnologia, capital e informação, podendo modelar os dispor destes recursos de várias maneiras. Para uma análise da organização, são fundamentais dois aspectos: a qualidade relativa destes recursos e as proporções com que podem ser reformulados - ou a sua flexibilidade;
 - c) História: é importante compreender as principais fases de desenvolvimento da organização no decorrer do tempo, bem como o impacto atual de acontecimentos passados. Por exemplo, decisões estratégicas anteriores, natureza de crises passadas, comportamento dos líderes e as reações da organização.
 - d) Estratégia: compreende as decisões que distribuem os recursos limitados em função das oportunidades de um determinado ambiente. Pode ser definida como as opções de mercados, ofertas, tecnologia e competência. Em função disso, os administradores precisam tomar decisões sobre os tipos de produto a oferecer, quais mercados ofertá-los e a maneira pela qual a empresa deve ser distinguida.
- 2) Componentes Organizacionais: o modelo organizacional proposto por TUSHMAN e NADLER (1993) define quatro componentes principais nas organizações:
- a) Trabalho: atividade básica a que a organização se dedica, determinada pela sua estratégia. A análise do trabalho deve considerar as diferentes tarefas necessárias e seus fluxos de trabalho específicos. A avaliação de todos os demais componentes depende fundamentalmente do claro entendimento das tarefas a serem executadas;
 - b) Pessoal: responsável pela realização das tarefas. Ao se analisar este componente, deve-se considerar a natureza do conhecimento e habilitações individuais, as diferentes necessidades e percepções dos indivíduos e fatores como idade e sexo, que podem influenciar o comportamento individual;
 - c) Disposições organizacionais formais: correspondem às estruturas e procedimentos explícitos e formalmente desenvolvidos de forma que as pessoas realizem as tarefas de acordo com a estratégia organizacional.
 - d) Organização informal: por mais complexas que sejam as disposições organizacionais formais existentes em qualquer organização, outra série de disposições tendem a surgir com o tempo. São implícitas e tácitas, mas podem exercer considerável influência sobre o comportamento. Complementam, em alguns casos, as disposições organizacionais formais, proporcionando estruturas que facilitam a realização do trabalho. Em outras circunstâncias, podem surgir como uma reação à estrutura formal. Vários aspectos da organização informal apresentam um efeito crítico sobre o comportamento, incluindo a liderança, os valores, as relações entre os grupos e também dentro deles.

3) Produto: Composto por tudo aquilo que a organização produz, como se desempenha e o seu grau de eficiência. Além do produto básico do sistema, que corresponde aos produtos e serviços ou retorno econômico, outros produtos também contribuem para o desempenho organizacional, como o funcionamento dos grupos, unidades e indivíduos dentro da organização. Portanto, três fatores devem ser considerados na avaliação do desempenho organizacional:

- a) A realização da meta ou eficiência com que a organização atinge os seus objetivos definidos na sua estratégia;
- b) Qual a eficiência com que a organização utiliza os recursos disponíveis, avaliando desta forma se ela utiliza ou não todo o seu potencial;
- c) A adaptabilidade, ou seja, se a organização se mantém favorável em seu ambiente, adaptando-se às mudanças ambientais.

Já que o funcionamento das unidades organizacionais contribui para o produto organizacional, este também é influenciado pelo comportamento individual. Certos fatores como reações afetivas, nível de estresse, satisfação pessoal ou bom ambiente de trabalho podem desta forma se tornar um produto desejado.

A organização pode portanto ser considerada como uma série de componentes, dentro de um sistema aberto. A relação entre estes componentes afetam a maneira pela qual se combinam para gerar o produto.

I.1.3- Arquitetura de TI

Vimos que as organizações tendem a evoluir suas estruturas organizacionais para refletir as realidades e as circunstâncias mutáveis do cotidiano, de forma a se manterem competitivas. A tendência de descentralização das unidades de negócio abordada no Capítulo I.1.1 gerou uma conseqüente descentralização não somente dos sistemas de informação, mas de toda a estrutura de TI.

BUCHANAN e LINOWES (1980) apud LAURINDO et al. (2001) desenvolveram alguns trabalhos analisando a descentralização da TI nas empresas. Segundo estes autores, há três razões que levaram à descentralização da TI:

- 1) Pressões por diferenciação: heterogeneidade de tarefas e funções ou distribuição geográfica que levem as unidades administrativas a terem diferenças em metas, perspectivas de tempo e estrutura;
- 2) Desejo de controle direto: necessidade de controlar sistemas que são críticos para sua operação ou quando há o desejo de controlar a prioridade no desenvolvimento;
- 3) Ligação do suporte à Informática ao poder: quando o controle das informações é um

dos fatores para obtenção do poder ou a TI é um veículo para implementar mudanças estruturais.

Tradicionalmente, em tais estruturas descentralizadas, uma determinada área de negócio solicita ao departamento de TI um determinado sistema para atender suas novas necessidades. Este analisa e qualifica as necessidades, desenvolvendo interna ou externamente um conjunto de programas que irão atender todas as necessidades do problema apresentado, os dados de entrada e saída, além da plataforma (*hardware* e *software*) na qual os novos programas irão ser executados. Seja qual for a arquitetura que tenha sido definida e implementada por TI, a maior parte (ou totalidade) das estruturas são criadas para atender àquela determinada necessidade, contendo inúmeras particularidades. Desta forma, não foram originalmente criadas para serem conectadas às outras aplicações de negócio existentes. Esta orientação é denominada por muitos autores como arquitetura de TI orientada a silos.

A adoção de soluções independentes geradas a partir de orientações departamentais gera na maioria das organizações um complexo mapa dos sistemas e de suas interfaces, conforme ilustrado na Figura I.4:



Figura I.4 - Exemplo do mapa das aplicações e suas interfaces em uma organização. Fonte: GORDON (2006).

Sob o ponto de vista de TI, as soluções desenvolvidas sem uma perspectiva corporativa podem implicar em grande esforço de integração, falta de suporte aos requerimentos funcionais e não-funcionais, dificuldade de gerência e operação e conseqüentemente um maior investimento, em uma análise de longo prazo.

De acordo com LIGUORI (2006), a Arquitetura de TI pode ser definida como “*um mapa que representa TI de uma empresa desenvolvido, implementado, mantido e usado para explicar e guiar como TI e seus componentes interagem de uma maneira eficiente para ajudar no atingimento dos objetivos da organização*”.

Essa visão global permite um entendimento da arquitetura de TI atual, ou seja, um mapeamento da infra-estrutura de *hardware* e de *software*, das aplicações e das suas interfaces, dos usuários dos sistemas, da governança e dos processos de negócio.

I.1.4- Arquitetura Corporativa

Conforme descrito por LAURINDO et al. (2001), ter uma clara visão estratégica para o negócio e para a TI é ponto de partida para que seja possível um impacto significativo no desempenho da organização. A TI precisa ser vista como um meio para a empresa obter vantagens competitivas no mercado em que atua, alinhada com a operação e a estratégia da empresa. A visão da TI como arma estratégica competitiva tem sido discutida e enfatizada, pois não só sustenta as operações de negócio existentes, mas também permite que se viabilizem novas estratégias empresariais.

Portanto, a Arquitetura Corporativa é um modelo utilizado para planejar a TI da corporação por meio do mapeamento e entendimento dos processos, sistemas, pessoas e tecnologias, viabilizando o trabalho e planejamento conjunto das unidades de negócio e a organização de TI. O principal benefício de um arquitetura corporativa é o alinhamento de TI com o Negócio, conforme descrito na Figura I.5:



Figura I.5 - Modelo da arquitetura corporativa. Fonte: LIGUORI (2006).

A Arquitetura Corporativa ajuda a responder algumas questões básicas e importantes:

- 1) Quais são os processos de negócio de uma organização e como TI os suporta?
- 2) A arquitetura de TI atual suporta e adiciona valor à organização?
- 3) Como a arquitetura de TI está se desenvolvendo para suportar a visão de futuro da organização?

Conforme descrito anteriormente, a aplicação do modelo de Arquitetura Corporativa é uma mudança na abordagem de TI de inúmeras empresas, visto que a maioria pratica um modelo de arquitetura de silos. A Arquitetura Corporativa traz grandes benefícios organizacionais, operacionais e torna TI mais visível e justificável para o negócio. Tais benefícios podem ser visualizados através da Tabela I.1 abaixo:

Tabela I.1 - Arquitetura em silos *versus* arquitetura corporativa. Fonte: LIGUORI (2006).

Arquitetura em Silos	Arquitetura Corporativa
Várias Arquiteturas	Uma única Arquitetura
Diferentes Soluções e Tecnologias	Solução Integrada
Falta de padronização	Padronização
Dificuldade de integração	Viabiliza a Integração
Duplicidade	Ganho de Sinergia
Distribuição do poder decisório	Centralização do poder decisório

Arquitetura em Silos	Arquitetura Corporativa
Falta de visão da Arquitetura da empresa	Mapeamento da Arquitetura da Empresa
Cada “arquitetura” endereça problemas de negócio da área/depto/silo	Endereça problemas de negócio da corporação

Conforme descrito por LIGUORI (2006), a elaboração e manutenção de uma Arquitetura Corporativa envolve determinadas etapas, descritas a seguir:

- 1) O Entendimento da arquitetura de TI atual. Ou seja, um mapeamento da infraestrutura de *hardware* e de *software*, das aplicações e de suas interfaces, dos usuários dos sistemas, da governança e dos processos de negócio. Tal mapeamento permite identificar a criticidade de cada sistema e sua respectiva infraestrutura, em função dos processos de negócio que eles suportam. Permite também mapear os serviços comuns que podem ser compartilhados entre as diversas áreas usuárias (analisaremos isto posteriormente ao longo deste trabalho).
- 2) A definição dos princípios de TI da organização, ou seja, os “mandamentos” que irão gerir a utilização e implementação de soluções de TI dentro da organização. Tais princípios irão garantir que novas iniciativas de TI sejam criadas e suportadas de maneira integrada e consistente, tornando-as padronizadas e mais eficientes.
- 3) O entendimento da visão de futuro da organização e alinhamento com TI por intermédio da identificação dos requerimentos de negócio e de TI, das tendências da indústria e da evolução tecnológica.
- 4) A criação de uma arquitetura de TI futura, tendo como base a arquitetura atual, respeitando os princípios definidos e atendendo aos requerimentos identificados. A arquitetura futura fornece uma visão de curto e longo prazo de TI, viabilizando a definição de uma estratégia alinhada com o negócio.

De forma resumida:

Arquitetura Corporativa => Arq.Futura = Arq.Atual + Princípios de TI + Requerimentos

Através da criação de uma visão única da arquitetura da empresa, diversas iniciativas podem ser conciliadas:

- Padronização da tecnologia;
- Reutilização de tecnologia;
- Integração;
- Resolução de problemas corporativos e não somente setoriais;

- Reduzir complexidade e redundância na Infra-estrutura de TI;
- Maximizar o retorno do investimento na infra-estrutura de TI existente;
- Alinhar com os objetivos principais da corporação;
- Tornar TI mais visível e justificável para o negócio;
- Tornar as decisões de definição e compra de TI mais simples, devido a existência de um plano alinhado com uma visão de futuro;
- Melhor gestão da informação/conhecimento.

I.2- Evolução dos Sistemas de Computação

Desde o desenvolvimento do primeiro computador digital eletrônico de alta velocidade, o ENIAC (Electrical Numerical Integrator and Calculator), em meados da década de 1940, temos presenciado o incansável surgimento de computadores cada vez mais rápidos e mais poderosos, de acordo com MARKS e WERRELL (2003). Nos últimos 60 anos foram realizados enormes progressos no *hardware* dos computadores, desde o surgimento dos computadores baseados em transistores até os microprocessadores integrados. Atualmente, devido às limitações físicas dos microprocessadores, a nanotecnologia representa um promissor futuro para o desenvolvimento dos novos *chips* de computadores.

De forma similar, ocorreram enormes avanços com relação às linguagens de programação. Os primeiros computadores (tais como o ENIAC) foram programados utilizando os códigos binários de máquina, compostos apenas pelos algarismos “0” e “1”. Nas últimas duas décadas, presenciamos o surgimento e proliferação das linguagens de terceira e quarta geração, tais como: JAVA, Visual Basic, Smalltalk, entre outras. Estas linguagens estão muito mais próximas da linguagem humana do que da linguagem binária que os computadores compreendem, permitindo o desenvolvimento de programas extremamente complexos em tempo reduzido.

Dentro de pouco tempo os programas e suas linhas de código cresceram exponencialmente, causando um caos para a gerência dos sistemas e conseqüentemente para os negócios, pois estes se tornavam cada vez mais dependentes de TI para serem executados com eficiência. Na tentativa de organizar este quadro, os programadores adotaram estruturas sistemáticas para organizar seus programas, chamadas de Arquiteturas de Software.

I.2.1- Arquiteturas de Software

Conforme descrito por HURWITZ et al. (2006), toda construção apresenta uma estrutura de alguma ordem. A idéia de arquitetura corresponde a um planejamento de acordo com determinadas regras e padrões. Uma boa arquitetura de *software* deve prever quais serão os

tipos de dados, como os mesmos serão armazenados, como será feita a interação com os usuários, como os programas e módulos se comunicarão entre si, e muito mais.

Nos últimos 40 anos, podemos destacar quatro principais tendências em termos de Arquiteturas de Software, ilustradas na Figura I.6:

- **Sistemas Monolíticos:** arquitetura de sistemas na qual o processamento, os dados e a interface do usuário estão residentes no mesmo sistema. Implementada e difundida através dos primeiros computadores de grande porte (mainframes);
- **Cliente-servidor:** arquitetura de rede que distingue dois elementos básicos: o cliente e o servidor. Cada instância do cliente pode enviar requisições para o servidor, que “escuta” e atende as requisições através da rede;
- **Computação Distribuída:** descentralizada, utiliza o poder de computação de dois ou mais computadores em uma rede para atingir determinado objetivo ou resultado;
- **Service Oriented Architecture (SOA):** nova tendência emergente no início deste milênio, e foco deste estudo. Iremos falar mais detalhadamente desta arquitetura nos capítulos posteriores.

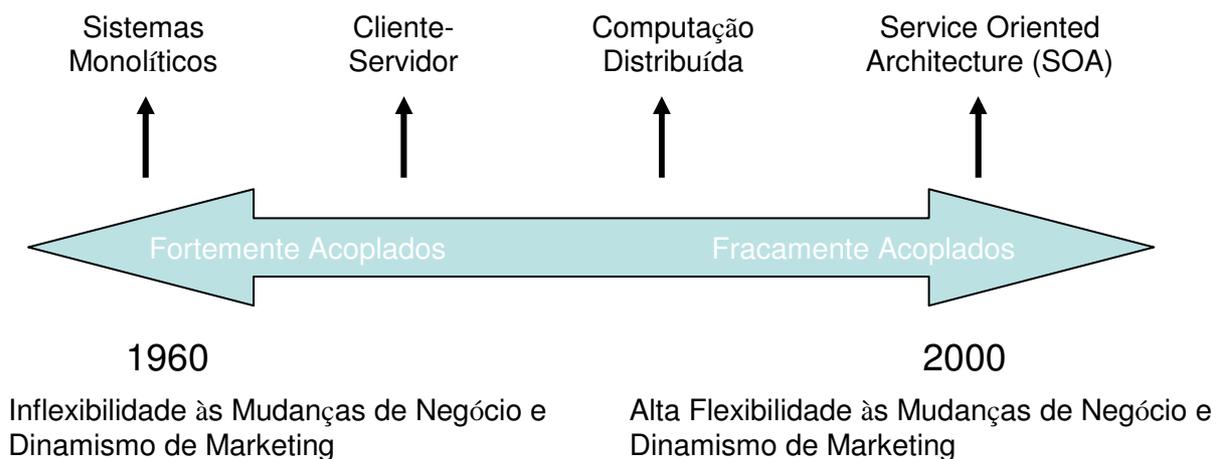


Figura I.6 - Evolução das arquiteturas de software. Fonte: MARKS e WERRELL (2003).

Seja qual for a arquitetura que tenha sido definida e implementada para a criação de um determinado sistema, a maior parte (ou totalidade) das estruturas são criadas para atender àquela determinada necessidade, contendo inúmeras particularidades. Desta forma, não foram originalmente criadas para serem conectadas a outras aplicações de negócio existentes.

I.2.2- Integração de Sistemas

Falar em integração de sistemas não parece algo novo, à primeira vista. Na década de 1990 observou-se a proliferação das soluções integradas de *software*, tais como os *Enterprise Resource Planning* (ERP), os *Customer Relationship Management* (CRM), e os sistemas do

tipo *Supply-Chain Management* (SCM). A grande maioria das médias e grandes organizações passaram a utilizar tais sistemas integrados para suportar e gerenciar significativos aspectos de negócio, permitindo a integração de todas as aplicações associadas às diferentes áreas funcionais de uma organização tais como: financeira, comercial, produtiva, de aprovisionamentos, de recursos humanos, de qualidade, de manutenção, etc.

Conforme descrito por LAURINDO e MESQUITA (2000), apesar da promessa de tantos benefícios, ao invés da fórmula tradicional de desenvolvimento ou aquisição de um sistema que seja adequado à empresa, na realidade as empresas necessitaram se adaptar às características das soluções integradas, causando diferentes impactos, conforme o tipo de negócio e a estrutura da empresa. Além disso, a especificidade de algumas áreas industriais demanda a necessidade de módulos externos ou mesmo a ligação a sistemas “legados”. Assim, cada vez mais se assistiu ao enriquecimento dos sistemas empresariais pela adição de novas funcionalidades, originando novas dificuldades de implementação e resultando num esforço crescente de reconfiguração dos sistemas de informação.

I.2.3- Arquitetura de Software Orientadas a Serviços

Segundo BIEBERSTEIN et al. (2006), o conceito do desenvolvimento de arquiteturas de software orientadas a serviços não é de fato uma novidade, e data de pelo menos uma década atrás. São encontradas implementações orientadas a serviços nos sistemas COBOL/CICS dos mainframes, em sistemas de orientação a objetos C++ e também em aplicações JAVA. Alguns destes conceitos evoluíram a partir de antigas idéias e uma variedade de tecnologias de *software*. Portanto, não se pode afirmar que a SOA é uma nova idéia. Muitas soluções tem sido implementadas de acordo com tais princípios, embora de forma proprietária (veremos maiores detalhes posteriormente), e conseqüentemente envolvendo maiores gastos. Além disso, apesar de implementadas a níveis departamentais e/ou multidepartamentais, são extremamente difíceis de serem compartilhadas a níveis de grandes organizações e estendidas a outras empresas e parceiros de negócio.

As grandes empresas de desenvolvimento de *software* tentaram estabelecer parcerias para compatibilizar seus produtos, tornando-os padrões de mercado. A razão do termo SOA se tornar agora de fato uma nova tendência é devido ao surgimento de uma nova tecnologia que tornou as soluções baseadas em SOA viáveis e produtivas: os Web Services.

Os Web Services serão abordados de forma detalhada no Capítulo 2, mas pode-se antecipar que eles foram fundamentais para o impulso da SOA, pois quebrou as barreiras proprietárias entre os diferentes desenvolvedores e fornecedores de *software*. Provou um comprometimento das grandes empresas como BEA, IBM, Oracle, SAP e Microsoft em

padronizar suas respectivas ofertas de *hardware* e *software*, permitindo o compartilhamento de dados e informação.

Os Web Services vêm influenciando cada vez mais a maneira com que os sistemas são desenvolvidos e implementados. Similar ao paradigma no início dos anos 90 que moveu os sistemas monolíticos baseados nos mainframes para a arquitetura cliente-servidor e a computação distribuída, vemos hoje o início de uma nova mudança de paradigma: dos sistemas distribuídos baseados na arquitetura cliente-servidor para a SOA, através da implementação dos Web Services.

I.3- A Importância da Inovação: Abordagem Teórica

Conforme descrito por FREIRE (2002), “*Através da inovação, a empresa constrói no presente as bases de seu desenvolvimento futuro*”. O meio envolvente e a visão do negócio variam, mas a necessidade de inovação é constante. Muitas empresas porém limitam-se em satisfazer as necessidades dos clientes ou seguir as iniciativas da concorrência, sem investir na inovação. Conseguem desta forma salvar custos, oferecendo produtos a preços mais competitivos. Mas a longo prazo, a estagnação da oferta devido à ausência de inovação induz os clientes aos produtos e serviços da concorrência.

Faz-se necessário portanto que as empresas acompanhem, ou melhor, antecipem a evolução das necessidades dos clientes. As empresas que não buscam a inovação correm o risco de perder clientes, reduzir a rentabilidade e até mesmo abandonar o negócio. Novas idéias e projetos dão origem a produtos e serviços inovadores, sustentando o crescimento das empresas a médio e longo prazo. A Tabela I.2, extraída de FREIRE (2002), compara as atitudes entre a empresa tradicional e a empresa inovadora.

Tabela I.2- Comparação: a empresa tradicional e a empresa inovadora. Fonte: FREIRE (2002).

	Empresa Tradicional	Empresa Inovadora
Atitude face à mudança	Se funcionar bem, não se muda	Se não se muda, mais cedo ou mais tarde deixa de funcionar bem
Principal receio	Fazer erros	Não melhorar
Atitude face às novas idéias	Rejeitar as idéias alheias e aplicar só idéias próprias	Adaptar as boas idéias alheias e também desenvolver idéias próprias
Responsabilidade pela experimentação	Gestores de topo e quadros técnicos	Todos os membros da organização
Orientação competitiva	Ter os melhores produtos e	Ter as melhores pessoas e

	Empresa Tradicional	Empresa Inovadora
	serviços	produtos de gestão
Resultado	Visão de curto prazo e estagnação	Visão de médio e longo prazo e crescimento

Nos últimos anos, a inovação vêm se tornando cada vez mais importante para todos os setores, devido às tendências de:

- Redução do ciclo de vida dos produtos;
- Excesso de capacidade instalada;
- Individualização da oferta;
- Aumento da intensidade competitiva;
- Acréscimo de sofisticação dos clientes;
- Aceleração da evolução tecnológica;
- Globalização das economias;
- Escassez dos recursos;
- Expectativas do mercado financeiro.

FREIRE (2002) descreve ainda que as empresas tendem a se dividir em dois blocos: competidores fortes, com capacidade de inovação; e competidores fracos, dotados de recursos baratos e baixa tecnologia/qualidade, que se limitam a reagir às pressões do mercado e da concorrência.

I.3.1- Estágios para Atingir a Inovação

Nem todas as organizações conseguem em um primeiro momento a busca da inovação. A grande maioria delas atravessa alguns estágios para conseguir atingir este objetivo. Tais estágios podem ser descritos como sendo:

- 1) Cópia: assimilação das tecnologias mais avançadas existentes no mercado, através de alianças, aquisições ou Engenharia Reversa. Uma vez identificadas as empresas ou institutos de pesquisa que lideram a tecnologia do setor em questão, é estabelecido o contato para licenciamentos ou acordos de transferência tecnológica;
- 2) Melhoria: trata-se do aperfeiçoamento das tecnologias adquiridas externamente ou desenvolvidas internamente. Basicamente são compostas por redução dos custos de produção, aumento de qualidade dos produtos e serviços, expansão da funcionalidade e minimização do tempo de produção e/ou entrega;
- 3) Inovação: Processo de criação e introdução de algo novo na organização ou no

mercado. Requer um investimento substantivo nas áreas de pesquisa e desenvolvimento. Objetiva criar uma oferta distinta, que permita à empresa diferenciar-se da concorrência do mercado.

I.3.2- Modalidades de Inovação

A Inovação não se resume à geração de novas idéias, pois requer a invenção de algo novo e a sua posterior aplicação dentro da própria organização ou do mercado. Esta é a principal diferença entre invenção e inovação: a invenção independe do uso, mas a inovação consiste na utilização da invenção no contexto interno e externo à empresa. Portanto, se um invenção não tem utilidade, não se trata de uma inovação.

As inovações podem ser divididas dentro de dois grupos: as inovações de processo, estas aplicadas dentro da organização, e as inovações de produtos ou serviços, que são orientadas para o mercado. Neste contexto, a SOA se caracteriza como uma inovação de processo. Em ambos os casos, a difusão da inovação depende da aceitação que esta encontre junto dos seus utilizadores potenciais.

Os novos processos, produtos e serviços podem ser enquadrados com relação à sua natureza incremental, distintiva ou revolucionária.

- 1) Inovação Incremental: Aperfeiçoamento gradual de determinado produto, processo ou serviço. Como exemplo, a evolução de determinado modelo de um produto;
- 2) Inovação Distintiva: Trata-se de uma melhoria de determinado produto, processo ou serviço. Como exemplo, o lançamento da venda de produtos via Internet e a engenharia paralela, visando reduzir o tempo de lançamento de um produto no mercado;
- 3) Inovação Revolucionária: Melhoria significativa através de recurso com nova tecnologia, fundamentalmente diferente da anterior. Tipicamente, uma inovação revolucionária gera inúmeras inovações distintivas e incrementais.

CAPÍTULO II- A ARQUITETURA ORIENTADA AOS SERVIÇOS (SOA)

Para uma definição clara da SOA, primeiramente precisamos definir um fundamental componente desta arquitetura: o serviço (*service*).

De acordo com KOCH (2006), serviços são porções - ou componentes - de *software* construídas de tal modo que possam ser facilmente vinculadas a outros componentes de *software*. A idéia por trás destes serviços é simples: definir partes dos códigos de *software* em porções significativas o suficiente para serem compartilhadas e reutilizadas em diversas áreas. Ou seja, os serviços são uma metodologia para conectar e comunicar.

Um serviço representa uma função independente, sem estado (*stateless*) que aceita uma ou mais requisições e retorna uma ou mais respostas através de uma interface padronizada e bem definida. Serviços podem também realizar partes discretas de um processo tal como editar ou processar uma transação. Os serviços não devem depender do estado de outras funções ou processos, e a tecnologia utilizada para prover o serviço (como uma linguagem de programação), não pode fazer parte da definição do mesmo.

Na grande maioria dos casos (embora não necessariamente), a comunicação entre o sistema cliente e aquele que disponibiliza o serviço é realizada através dos Web Services. Tais sistemas podem residir em um mesmo computador, ou podem estar distribuídos em computadores diferentes ao redor do mundo, conectados através de uma rede ou da própria Internet. Tanto as aplicações quanto os computadores não precisam ter similaridade. É possível, por exemplo, que dois sistemas desenvolvidos utilizando a tecnologia Microsoft .Net em um mesmo computador disponibilizem serviços e se comuniquem entre si. Da mesma forma, é possível que um sistema desenvolvido em JAVA em um computador do tipo RISC no Brasil possa se comunicar com um outro sistema desenvolvido em C++ em um microcomputador do tipo CISC na China, independente do sistema operacional em uso. Neste capítulo (e em todo o estudo), abordaremos exclusivamente a implementação e compartilhamento de serviços através da tecnologia dos Web Services.

Desta forma, várias peças independentes de software podem se comunicar entre si através dos Web Services, que consiste em uma troca de mensagens através de um formato universal pré-definido. Para que a SOA se torne eficiente, é importante que o envio e recebimento destas mensagens seja feito de forma rápida e garantida. E para este propósito surgiu a tecnologia conhecida como ESB (*Enterprise Service Bus*). O ESB é uma camada de *software* existente para o transporte das mensagens entre os provedores e consumidores de serviço.

Em estágios mais avançados de implementação da SOA (isso será abordado de forma mais detalhada no Capítulo 3), a localização dos serviços por parte de um cliente pode ser

realizada pelo *Universal Description, Discovery and Integration* (UDDI), que é uma especificação técnica para descrever, descobrir e integrar os Web Services.

Outra característica da SOA é que esta arquitetura permite tornar as mudanças nos processos de negócio mais flexíveis. Isto ocorre porque a SOA permite que TI desenvolva aplicações compostas por funções de negócios já existentes, através da exposição de suas interfaces. Desta forma, os processos de negócio podem ser modificados com alto grau de abstração em relação à tecnologia. Existem várias ferramentas no mercado que auxiliam na orquestração dos processos, sendo tais ferramentas conhecidas como *Business Process Management* (BPM). Desta forma, este capítulo irá descrever:

- A tecnologia por trás dos Web Services;
- A camada de transporte de mensagens ESB;
- A forma de localização dos Web Services através do UDDI;
- A utilização de ferramentas BPM para orquestração de processos de negócio;
- O valor agregado dos web Services.

II.1- Os WEB Services

OS Web Services podem ser definidos como uma solução utilizada para a integração de sistemas e comunicação entre aplicações diferentes, conforme descrito em WIKIPEDIA (2006). Através desta tecnologia, é possível que novas aplicações possam interagir com aplicações existentes, e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Cada aplicação pode ter a sua própria linguagem, que é traduzida para uma linguagem universal, o formato XML, permitindo às aplicações o envio e recebimento de dados. De forma ainda mais detalhada, MARKS e WERRELL (2006) descrevem que os Web Services podem ser definidos através de cinco características:

“Web Services são componentes fracamente acoplados, auto-descritos e acessados por intermédio de programas através de uma rede de computadores, permitindo a troca de dados independente do fornecedor do software, plataforma e linguagem”.

Vamos agora analisar cada um dos componentes mencionados na definição acima:

- 1) ***Fracamente Acoplados:*** Conceito que remonta décadas. SAMPAIO (2006) apud Page-Jones (1980) descreve que o conceito é aplicado à estrutura modular de um sistema. Um módulo de sistema seria uma coleção de instruções de programa, contendo quatro atributos básicos: entrada/saída, função, lógica e dados internos. Desta forma, um módulo poderia ser uma sub-rotina ou função, sendo o acoplamento uma medida

da interdependência entre dois (ou mais) módulos. Quanto mais acoplados, mais dependentes e vulneráveis à mudança. Os sistemas baseados nos *mainframes* da década de 1970 eram implementados através de enormes blocos de funcionalidades fortemente acoplados, que eram executados em um único computador central. De forma oposta, os sistemas implementados através de Web Services são fracamente acoplados, podendo ser executados a partir de diversos componentes diferentes de *hardware* e *software*. Uma das grandes vantagens dos Web Services é que os componentes fracamente acoplados são mais flexíveis e facilmente reconfigurados, permitindo que os componentes do sistema sejam substituídos ou trocados com relativa facilidade.

- 2) **Auto-descrito:** O *Web Services Description Language* (WSDL) é um documento no formato XML que descreve as entradas e saídas dos Web services de uma forma estruturada. O documento WSDL permite que outro *software* determine como chamar determinado serviço para ser executado, e qual resultado o serviço deve retornar. Abordaremos estes conceitos mais detalhadamente nos capítulos posteriores.
- 3) **Acesso via programação:** Os Web services não apresentam uma interface gráfica que permitam interação com usuários. Do contrário, são chamados por outras aplicações através de programas implementados através do *Simple Object Access Protocol* (SOAP). Esta forma de acesso permite com que os Web services sejam incorporados a outras aplicações, *web sites*, ou até mesmo a outros serviços. Toda e qualquer comunicação entre sistemas passa a ser dinâmica e principalmente segura, pois é livre da intervenção humana.
- 4) **Distribuídos através da rede:** Os Web services são acessados utilizando-se protocolos e formatos de dados da Internet, tais como TCP/IP, HTTP e XML. Utilizando tais protocolos e formatos de dados, os Web services são compatíveis com os padrões e políticas de segurança vigentes nas organizações (como por exemplo, os *firewalls* corporativos). Esta característica torna possível que os Web Services sejam implementados e acessados dentro (Intranet) e fora (Internet) das redes corporativas.
- 5) **Troca de dados utilizando protocolos, linguagens e plataformas neutras:** Esta seja talvez a característica mais importante dos Web services, representando a capacidade de um Web service trocar dados

independentemente do fabricante, plataforma e linguagem através de padrões abertos, aceitos e adotados pela indústria de TI.

II.2- Evolução da Tecnologia - WEB Services

Nesta seção será descrita a evolução dos padrões que formaram a base para a criação dos Web services, sendo possível compreender porque isto é tão significativo para a importância e benefícios desta tecnologia.

II.2.1- O World Wide Web Consortium (W3C)

Em 1994, um consórcio internacional formado por especialistas em

notificados. Cada modificação normalmente passa por um processo estruturado de aceitação por parte da comunidade desenvolvedora do *software*, e as melhorias homologadas são incorporadas às novas versões do *software*. Atualmente, muitas empresas de tecnologia (e não somente indivíduos) vem contribuindo para o desenvolvimento dos *softwares* abertos.

Conforme destacado por ALKALAY (2006), é comum e incorreto afirmar que o *software* aberto represente o fim do *software* proprietário (ou de código fechado). Isto não ocorre devido à tendência de que as inovações venham a ser criadas através de um modelo fechado, naturalmente permitindo aos desenvolvedores maximizarem seus lucros através do controle de seus inventos. Contudo, à medida que tais invenções se popularizam criando um mercado de usuários e potenciais concorrentes, a abordagem do *software* aberto surge como a forma mais inovadora para se reimplementar e explorar o que foi inventado anteriormente.

Hoje, muitos *softwares* de padrão aberto apresentam alto grau de maturidade e são utilizados largamente em escala corporativa, tais como LINUX, , m, i ora r

detalhadamente como ocorre a interação entre sistemas heterogêneos através da utilização de Web Services.

Desta forma, é possível responder à pergunta anterior: qual o melhor tipo de *software* a ser utilizado: aberto ou fechado? O ideal é adotar o uso de padrões abertos, balanceando a utilização de *softwares* de código aberto ou fechado, conforme cada necessidade. Pois são os padrões abertos que garantem a interoperabilidade entre as camadas de aplicações, possibilitando que as organizações componham o melhor para suas necessidades através da liberdade de escolha de *hardware* e *software*.

II.3- Componentes da Arquitetura

Nesta seção abordaremos as tecnologias que formam a base para a construção dos Web Services.

II.3.1- Extensible Markup Language (XML)

O XML é a abreviação de *Extensible Markup Language* (a tradução poderia ser linguagem extensível de marcação). Trata-se de uma recomendação do W3C para gerar uma linguagem de marcação capaz de descrever diversos tipos de dados, e criada para superar certas limitações do HTML, que é o padrão utilizado para a construção de páginas WEB. Não se trata de uma linguagem de programação, mas uma maneira de estruturar dados através da independência da plataforma e/ou linguagem. A sintaxe do XML permite com facilidade a manipulação dos dados pelos programas, porém através de marcações - em inglês *tags* - compreensíveis na linguagem humana.

A linguagem XML foi definida como o formato universal para a estruturação de dados na WEB. A extensibilidade do XML é tanta que muitas corporações vêm adicionando funções XML em seus produtos, como a Microsoft, Oracle, IBM, Google e Sun. De acordo com ALECRIM (2006), o XML tende a alcançar um sucesso cada vez maior, não só no segmento de comércio eletrônico, como vem acontecendo, mas em praticamente todas as áreas da WEB.

Pelo fato do XML proporcionar com facilidade o compartilhamento de informações através da Internet, foi adotado como padrão de comunicação dos Web Services. Neste contexto, o XML não é apenas utilizado como formato para troca de mensagens, mas também como a forma através da qual os serviços são definidos. Através de esquemas XML, é possível validar as informações separadamente e descrever os atributos e características dos dados. Duas partes só podem trocar informações em XML e entender os elementos da mesma forma se compartilharem uma mesma definição sobre quais e como os elementos podem ser utilizados.

O Exemplo abaixo ilustra a formatação dos dados referentes à um pedido de compras, definido em um documento do tipo XML:

```
<?xml version="1.0" ?>
xmlns="http://www.w3.org/2007/exemplo"
  <OrdemdeCompra>
    <Cliente>
      <Nome>Roberto Fagundes </Nome>
      <Endereco>Av Bartolomeu Mitre 204 Ap 202 Leblon</Endereco>
    </Cliente>
    <Fornecedor>
      <Nome>ACME Servicos Integrados</Nome>
      <Endereco>Av Pasteur 138</Endereco>
    </Fornecedor >
    <Pedido>
      <Item>Notebook IBM Thinkpad Modelo T60</Item>
    </Pedido>
  </OrdemdeCompra>
```

No exemplo acima podemos verificar os atributos XML referentes a um pedido de compras. É interessante destacar a utilidade da linha definida como `xmlns=http://www.w3.org/2007/exemplo`. Trata-se do *namespace* que foi utilizado. Um namespace no XML é uma recomendação do W3C para prover nome de elementos e atributos únicos em cada instância XML. Por exemplo, um elemento XML `OrdemdeCompra` e outro elemento XML `Cadastro_Clientes` podem ambos utilizar um elemento definido com o nome `Cliente`. Desde que especificados em namespace distintos, não ocorrerá duplicidade.

II.3.2- Simple Object Access Protocol (SOAP)

Vimos que a comunicação entre os Web Services ocorre através do formato XML, portanto as aplicações podem trocar mensagens neste formato. Porém, como fazer com que elas interpretem os resultados passados?

O protocolo SOAP (acrônimo do inglês *Simple Object Access Protocol*) especifica como as mensagens XML dos Web Services são formatadas. Ele define qual é a estrutura das mensagens, de forma que as aplicações entendam como enviar e interpretar os dados recebidos. A estrutura básica da mensagem SOAP está descrita na Figura II.1:

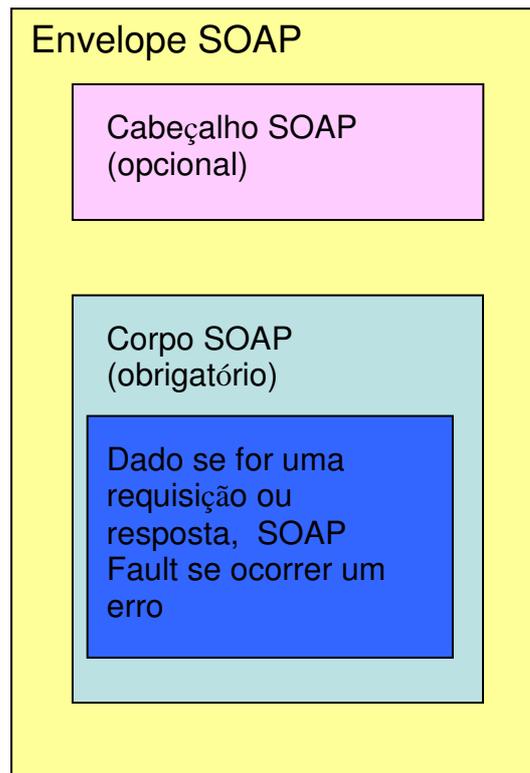


Figura II.1 - Estrutura básica da mensagem SOAP. Fonte: ROBICHAUX (2006).

Na forma mais básica, a mensagem SOAP contém o corpo SOAP, e dentro deste o dado que está sendo transmitido. Em alguns casos, há o cabeçalho SOAP (opcional), que pode conter informações extras. Além disso, o formato da mensagem SOAP apresenta uma estrutura que contém a descrição do erro apresentado, caso tenha ocorrido. O modelo de dados SOAP oferece definições para os tipos de dados que serão trocados, tais como caracter (string), inteiros (integer), ponto flutuante (float), decimais duplos (double) e data (date). O exemplo abaixo ilustra o formato da mensagem SOAP:

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

<soap:Header>
...
...
</soap:Header>

<soap:Body>
...
```

```

...
<soap:Fault>
...
...
</soap:Fault>
</soap:Body>

</soap:Envelope>

```

O SOAP é portanto um protocolo para troca de informações em um ambiente de computadores descentralizado e distribuído, que permite comunicação entre aplicações heterogêneas de forma simples, independentemente de sistema operacional, linguagem de programação ou plataforma. A comunicação é realizada através de uma troca de mensagens flexível e padronizada, no formato XML, que inclui os parâmetros usados na chamada e os dados de resultado. Para o transporte dos dados, utiliza um protocolo bem simples e difundido, o HTTP (muito embora possa ser utilizado com outros protocolos como o SMTP e o FTP).

Inicialmente o protocolo SOAP foi criado para permitir a requisição remota de métodos através da Internet, na época em que poucas alternativas eram possíveis (tais como DCOM, CORBA ou RMI). Apresentando características como simplicidade, independência de fabricante, independência da linguagem, independência do modelo de objetos e independência do protocolo de transporte, evoluiu para se tornar um protocolo completo de mensagens XML e é hoje um padrão regulado pelo W3C.

Os exemplos abaixo exemplificam uma requisição e uma resposta SOAP referente a uma consulta de preço de um determinado produto (Banana):

Requisição:

```

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body>
  <m:ObtemPreco xmlns:m="http://www.leovidal.com/precos">
    <m:Item>Banana</m:Item>
  </m:ObtemPreco>
</soap:Body>

</soap:Envelope>

```

Resposta:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body>
  <m:ObtemPrecoResposta xmlns:m=" http://www.leoival.com/precos ">
    <m:Preco>4.50</m:Preco>
  </m: ObtemPrecoResposta >
</soap:Body>
</soap:Envelope>
```

A Figura II.2 ilustra como um determinado serviço (Web Service) é requisitado através do SOAP:

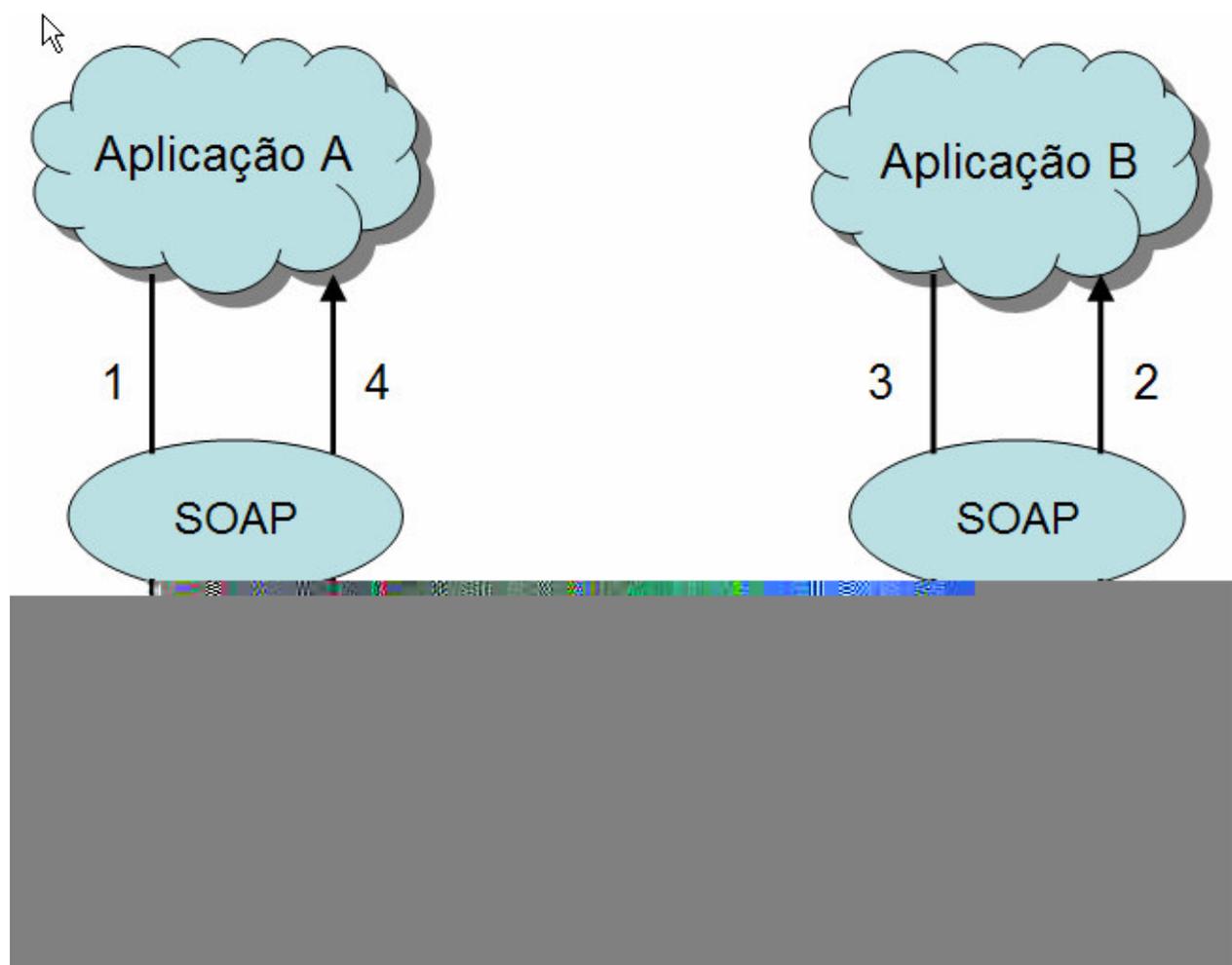


Figura II.2 - Requisição de um serviço através do SOAP.

Abaixo é detalhado funcionamento de cada uma das etapas listadas acima:

1. Através do protocolo HTTP, a aplicação A requisita um determinado serviço via rede, utilizando uma requisição SOAP;
2. A requisição SOAP é encaminhada via rede ao endereço internet (*IP Address*) da Aplicação B;
3. A Aplicação B interpreta a requisição SOAP, processa e retorna uma outra mensagem, do tipo resposta SOAP;
4. A resposta SOAP é encaminhada via rede ao endereço internet da Aplicação A, que requisitou o determinado serviço.

Portanto, uma mensagem SOAP utiliza o protocolo HTTP para realizar a conexão e o envio da mensagem através da rede:

HTTP + XML = SOAP

Uma requisição SOAP pode utilizar ambos os métodos HTTP: POST ou GET.

II.3.3- Web Services Description Language (WSDL)

Cada aplicação provedora de um determinado serviço através de Web Services utiliza um tipo de arquivo denominado WSDL (Web Service Description Language), que descreve quais os serviços que estão disponíveis e como eles devem ser invocados. Assim como o SOAP, o WSDL é um formato baseado na linguagem XML, que descreve como é feita a comunicação entre os Web Services. No centro do arquivo WSDL, o provedor do Web Service define os métodos que serão disponibilizados para serem consumidos pelas demais aplicações. Exemplificando, uma determinada aplicação pode disponibilizar um determinado método chamado *RetornaDescricaoProduto*, que retorna a descrição de um determinado produto referente a um código de produto recebido como entrada. O arquivo WSDL lista o método e outras informações tais como:

- 1) O nome específico utilizado para invocar o método (neste caso, *RetornaDescricaoProduto*);
- 2) Os parâmetros que devem ser utilizados na requisição, caso existam (nome dos parâmetros e tipos de dados especificados);
- 3) O formato do valor ou valores que serão retornados se a requisição for processada com sucesso (novamente, nome dos parâmetros e tipos de dados especificados);
- 4) A URL (endereço WEB) e protocolo que deve ser usado para invocar o método.

Além disso, também são disponibilizadas no arquivo WSDL outras informações como a indicação de como a mensagem SOAP deve ser formatada, o *namespace* que será utilizado, a ordem e estrutura dos parâmetros e os protocolos de ligação, que determinam as formas de transmissão das mensagens pela rede até os destinatários. Porém, uma explicação detalhada de cada componente de um arquivo WSDL e como estes componentes se combinam está fora do escopo deste estudo. Isto porque na maioria dos casos, um desenvolvedor de Web Services não precisa necessariamente entender como ler, passar e interpretar o conteúdo do arquivo WSDL. A maioria das modernas ferramentas de software realizam esta função para o programador, cabendo a este apenas a tarefa de fornecer os dados para a chamada de um Web Service e manipular os dados retornados como resposta. No capítulo 4 será apresentado o formato de arquivo WSDL, utilizado para a criação de um Web Service para o estudo de caso.

Assim como no SOAP, existe uma especificação padrão e universal para o WSDL, o padrão WSDL 1.1. De acordo com ROBICHAUX (2006), em breve este padrão será substituído pelo padrão WSDL 2.0.

II.3.4- Enterprise Service Bus (ESB)

Através da SOA, diversos componentes distintos de *software* podem se comunicar entre si através da troca de mensagens via Web Services. Quanto maior for o escopo e a utilização dos Web Services, maior será o volume desta troca de mensagens. É importante portanto garantir o envio e recebimento destas mensagens. Para este propósito, surgiu a tecnologia conhecida como ESB (*Enterprise Service Bus*). O ESB é uma camada de software existente para o transporte das mensagens entre os provedores e consumidores de serviço, gerenciando todo o percurso das mensagens de uma parte a outra da rede, conforme ilustrado na Figura II.3:

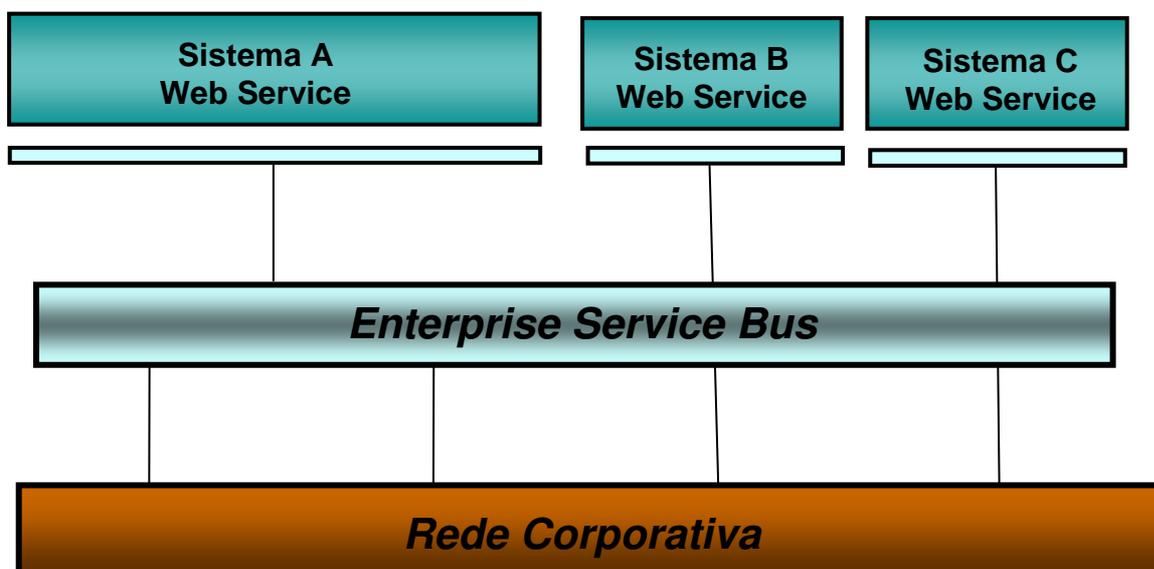


Figura II.3 - O Enterprise Service Bus (ESB).

Quando um determinado componente utiliza o serviço, ele simplesmente se conecta ao ESB passando a mensagem no formato correto informando o endereço do componente de destino para o qual a mensagem deverá ser entregue. O ESB se encarrega de todo o resto.

II.3.5- Universal Description, Discovery and Integration (UDDI)

Para que um serviço seja utilizado é necessário que o requisitante consiga localizá-lo, e esta localização pode ser feita através do UDDI, que é uma especificação técnica para descrever, descobrir e integrar Web Services. Para isso, existe um registro global público, chamado UDDI *business registry*, no qual toda a informação está disponível para consultas em geral. Cada provedor de serviço deve especificar que serviços serão disponibilizados, e em que categoria eles devem ser classificados. O registro de serviços é também responsável por tornar as interfaces dos serviços disponíveis para os consumidores de serviços. Os implementadores de um serviço devem levar em consideração o escopo no qual o serviço será implementado. Por exemplo, existem registros de serviço disponíveis na Internet, para uma audiência irrestrita (Ex: Google, Amazon.com, Submarino.com, etc...), assim como existem registros de serviço disponíveis para acesso exclusivamente dentro das redes corporativas. O consumidor do serviço localiza (descobre) entradas no registro de serviços e então acessa o provedor do serviço requisitando sua utilização, conforme descrito na Figura II.4. Um registro privado pode adicionar controle de segurança para proteger os dados e prevenir acessos não autorizados.

Um serviço de registro UDDI é também um Web Service, que gerencia informação sobre provedores, implementações e metadados de serviços. A especificação UDDI define:

- APIs SOAP utilizadas para publicar e obter informações de um registro UDDI;
- Esquemas XML do modelo de dados do registro e do formato das mensagens SOAP;
- Definições WSDL das APIs SOAP;
- Definições de registro UDDI (modelos técnicos - *tModels*) de diversos sistemas de identificação e categorização, que podem ser utilizados para identificar e categorizar registros UDDI.

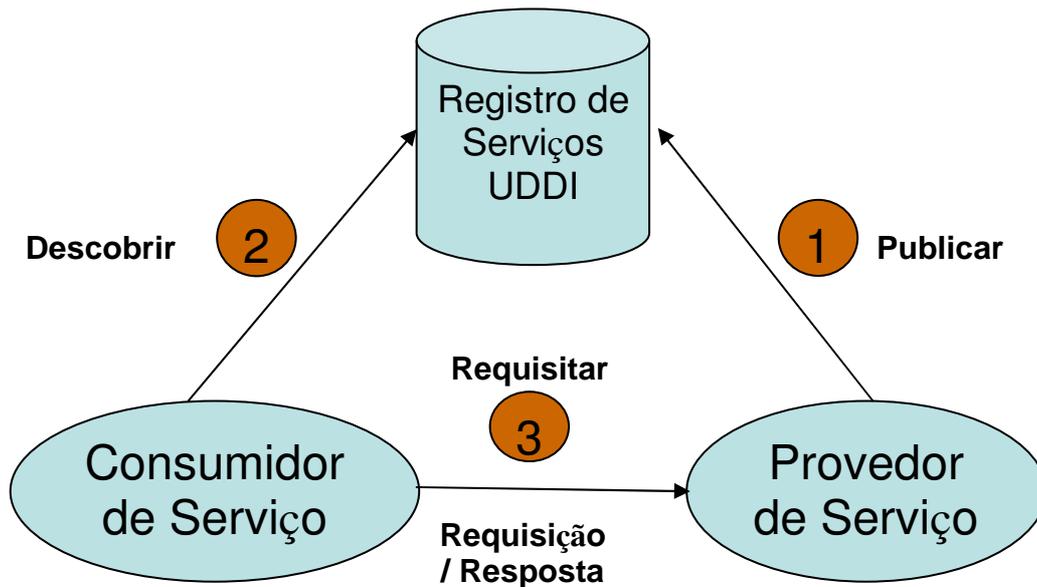


Figura II.4 - Processo de descoberta de serviços via UDDI.

II.3.6- Business Process Management (BPM)

Conforme visto anteriormente, uma importante característica da SOA é que ela permite a combinação de elementos fracamente acoplados. Desta forma, possibilita a construção de aplicações compostas por funções de negócios já existentes, através da exposição de suas interfaces, flexibilizando desta forma as mudanças nos processos de negócio. Os mesmos podem ser modificados com alto grau de abstração em relação à tecnologia empregada. Existem várias ferramentas no mercado que auxiliam na modelagem dos processos, sendo tais ferramentas conhecidas como *Business Process Management* (BPM).

As ferramentas de BPM podem organizar os fluxos de trabalho, descrevendo as funções de negócios existentes e criando novas. Além disso, servem para monitorar os processos de negócio, identificando possibilidades de mudanças de processo que resultam em aumento de eficiência.

De acordo com HURWITZ et al. (2006), a codificação dos processos de negócio é um passo fundamental para qualquer organização sujeita a normas de regulamentação tais como Sarbanes-Oxley Act, Health Insurance Portability and Accountability Act (HIPAA), e outras.

Existem diferentes fabricantes e ferramentas de software BPM, porém a grande maioria delas compartilha três aspectos comuns:

- 1) Criação de novas funções de negócio: permite adição de novas funções de negócio ou simplesmente adicionar uma lógica para ser executada antes ou após outra função;
- 2) Agrega as funções de negócios das aplicações existentes: As ferramentas BPM podem interagir com o registro de serviços, identificando que serviços estão

publicados, montando novos fluxos ou modificando os atuais a partir de serviços disponíveis. Além disso, podem adicionar novos serviços no registro;

- 3) Programa a engrenagem do fluxo de dados que irá sustentar o processo de negócios: utilizando as ferramentas BPM, o analista de negócio desenha os fluxos e especifica como será realizado o fluxo de trabalho de um departamento (ou indivíduo) para outro dentro da lógica do fluxo de negócio. Desta forma, ligam as aplicações que serão utilizadas às funções que devem ser executadas.

A Figura II.5 representa graficamente como uma ferramenta BPM realiza as três atividades:

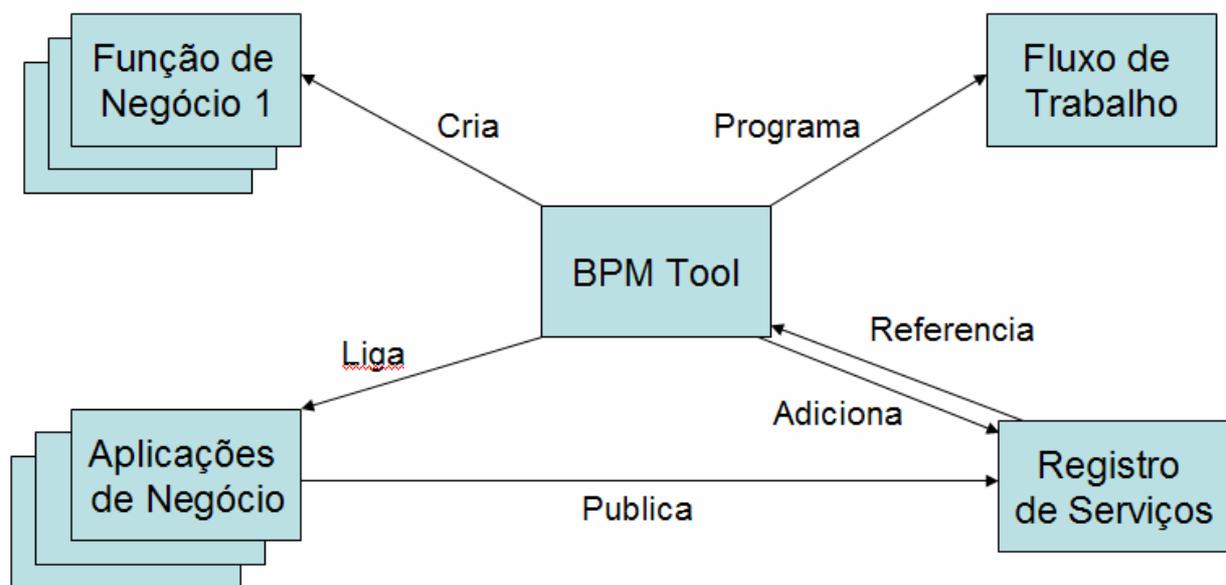


Figura II.5 - Funcionamento de uma ferramenta BPM.

II.4- Valor Agregado dos Web Services

Vimos que na utilização dos Web Services, uma aplicação consumidora do serviço lê um arquivo WSDL, formata e envia uma mensagem SOAP baseada nas definições daquele arquivo. Finalmente, recebe uma outra mensagem SOAP como resposta, oriunda do provedor do serviço. Os dados são transferidos no formato XML, encapsulados pelo protocolo SOAP e transportados através do protocolo HTTP. Vimos também que estes são todos protocolos abertos, permitindo interoperabilidade independentemente do fabricante do *software*. Desta forma, o uso de Web Services reintroduz a flexibilidade de negócios que os sistemas integrados tais como os ERP e CRM removeram, devido às suas rígidas definições de processos e interfaces proprietárias. Conforme descrito por ATHENA (2006) e apresentado na Figura II.6, os custos associados à interoperabilidade dos sistemas durante a implementação de novas aplicações corresponde a 40% dos investimentos de TI das organizações.

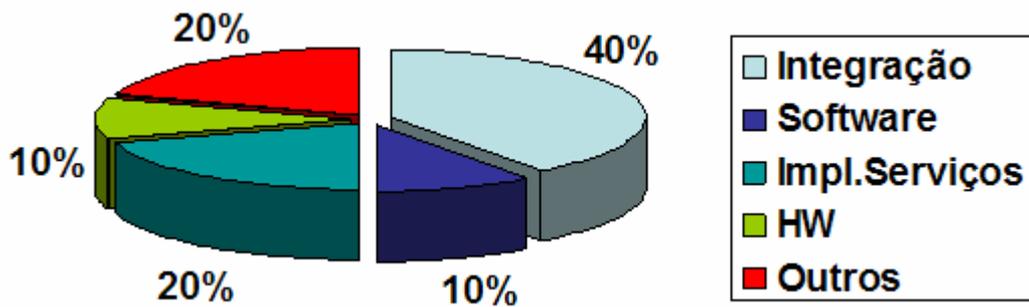


Figura II.6 - Custos associados à implementação de sistemas. Fonte: ATHENA (2006).

Portanto, as aplicações podem se comunicar independente da linguagem em que foram escritas ou plataforma em que são executadas. No passado, existiam formatos de dados específicos requeridos para cada aplicação, que podiam ou não ser acessados por programas escritos em outras linguagens. A utilização do XML como padrão pelos Web Services garante a interoperabilidade entre todos os serviços disponibilizados pelas aplicações.

Além disso, o uso dos Web Services traz agilidade aos processos e eficiência na comunicação entre sistemas e cadeias de produção ou de logística. Toda e qualquer comunicação entre sistemas passa a ser dinâmica e principalmente segura, pois não há intervenção humana.

De acordo com MARKS e WERREL (2003), na atual era de negócios globalizados, a colaboração entre parceiros de negócio vem se tornando a regra geral, e não mais a exceção. Desta forma, os Web Services permitem hoje a implementação de novos níveis de colaboração entre as empresas através das cadeias de negócio existentes, impulsionando novos tipos de relacionamentos com parceiros de negócios através dos sistemas do tipo B2B (do inglês *Business to Business*). As organizações compreenderam os benefícios significativos que podem ser obtidos através de uma maior cooperação e compartilhamento de informações com seus clientes e fornecedores. As aplicações existentes, no entanto, não foram originalmente desenhadas para permitir colaboração e integração, na maioria dos casos interna ou externamente, através da Internet. O uso dos Web Services estende a capacidade destes aplicativos, suportando a integração de aplicações internas e externas (através dos *firewalls* corporativos). Mais do que isso, torna-os capazes de responder de forma ágil às constantes mudanças de negócio.

O grande valor agregado dos Web Services vêm do resultado de todos estes fatores reunidos. Como a tecnologia atual torna possível adotar um modo padrão para que as aplicações diferentes se comuniquem através das redes, é possível hoje projetar as aplicações corporativas diferentemente. Em vez das aplicações monolíticas atendendo as especificidades

de um determinado departamento, com funcionalidade própria, é possível agora escrever programas modulares, alinhados com uma estratégia global e única. Isto permite uma redução de custos com compra de *software*, implementação e integração.

Os Web services oferecem portanto uma ponte entre o legado das arquiteturas de TI departamentais (orientadas aos silos) e o novo conceito de arquitetura corporativa.

Porém, refletindo através das experiências dos últimos anos, muitos executivos da área de TI devem estar se perguntando: “Porque devo investir em Web services, especialmente quando meus investimentos anteriores em comércio eletrônico não me renderam o retorno que os revendedores de *software* e consultores de TI me prometeram?”. Devem também se perguntar: “Não são os Web services apenas uma criação da indústria de *software* e serviços de TI para incrementar suas rendas?”. Tal contestação é compreensível, se analisarmos as tentativas e fracassos anteriores.

Ao levantar estas questões, os executivos devem primeiramente analisar e compreender a principal diferença entre as tendências de TI anteriores e o que vemos hoje: o desenvolvimento dos Web Services. Hoje, se observa um nível de colaboração sem precedentes no desenvolvimento de padrões de interoperabilidade envolvendo os Web Services, partindo de grande empresas como IBM, Microsoft, SUN e BEA. Isto representa um claro sinal que a indústria de Software está entrando em uma fase de maturação, na qual padrões consistentes de interoperabilidade se tornam cada vez mais algo mandatório para a indústria e mercado.

Segundo estudos da Forrester Research descritos em COMPUTERWORLD (2006), aplicações centrais desenvolvidas internamente, conectadas a pacotes de fornecedores e suportadas por SOA será a nova tendência no mercado de TI.

Porém, KOCH (2006) descreve que antes de mais nada, os benefícios da SOA devem ser contextualizados. Se uma determinada empresa não for grande ou complexa, isto é, se não tiver mais de dois sistemas primários que exijam algum nível de integração, é improvável que o modelo proporcione grandes benefícios. Assim, o primeiro passo é descobrir se existem aplicativos redundantes e mal integrados que poderiam ser consolidados ou eliminados como resultado da adoção. Se este for o caso, então há benefícios potenciais. Além disso, empresas pequenas ou empresas extremamente descentralizadas talvez não consigam justificar uma equipe central de gerentes de projeto, arquitetos e desenvolvedores.

CAPÍTULO III- MODELO PARA IMPLEMENTAÇÃO E GESTÃO DA SOA COMO INOVAÇÃO PARA A INTEGRAÇÃO DE SISTEMAS DE INFORMAÇÃO

A inovação é fundamental ao desenvolvimento de qualquer organização. Porém, de forma a tornar possível tal desenvolvimento, é necessário que as empresas adotem um modelo global para gestão da inovação, que alinhe o ciclo de inovação com as suas competências de gestão. Este Capítulo propõe uma metodologia para a implementação e gestão da SOA como uma inovação dentro das organizações.

III.1- Modelo de Gestão da Inovação

A inovação é fundamental ao desenvolvimento de qualquer organização. Mas para que seja possível tal desenvolvimento, é necessário que as empresas adotem um modelo global para gestão da inovação, que alinhe o ciclo de inovação com as suas competências de gestão.

Conforme descrito por FREIRE (2002), qualquer projeto de inovação evolui ao longo de um ciclo com seis fases (Figura III.1):

1. Detecção de Oportunidades
2. Geração de Idéias
3. Desenvolvimento das idéias selecionadas
4. Teste dos protótipos
5. Introdução do novo produto, serviço ou processo
6. Difusão no mercado



Figura III.1 - Ciclo de inovação. Fonte: FREIRE (2002).

Este ciclo pode ser gerenciado de maneiras distintas pelas empresas. Para sustentá-lo, as empresas deve dominar quatro competências de gestão básicas à inovação, conforme detalhado na Figura III.2 e adaptada de FREIRE (2002):

- 1) Gestão Estratégica: a iniciativa de inovação deve estar alinhada à orientação estratégica da empresa, contribuindo assim para a sua competitividade;
- 2) Gestão de Projetos: Os projetos devem ser conduzidos dentro de práticas e modelos, a nível de planejamento, execução e controle;
- 3) Gestão Funcional: As atividades de inovação devem ser suportadas pelas políticas

funcionais da empresa;

- 4) Gestão de Mudanças: A cultura organizacional deve estar aberta à mudança, num espírito proativo.

O domínio das competências de gestão é uma condição necessária, mas não suficiente para o sucesso de projetos de inovação. É da permanente interação e união do ciclo de inovação com as competências de gestão que resulta o sucesso da inovação.

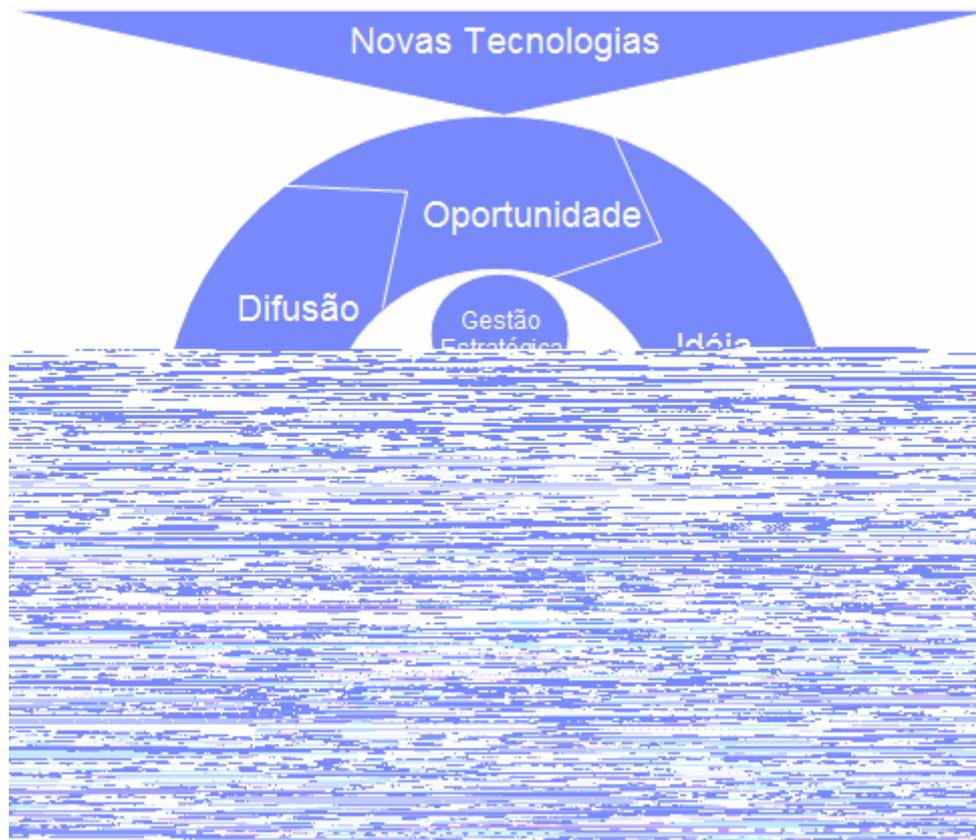


Figura III.2 - Modelo de gestão da inovação. Fonte: FREIRE (2002).

III.2- Competências de Gestão da Inovação

Ao longo deste estudo, estaremos abordando a SOA como uma inovação tecnológica dentro da organização. Vimos anteriormente que uma organização deve dominar as quatro competências de gestão básicas à inovação, sendo esta uma condição necessária (mas não suficiente) para o sucesso de projetos de implementação da SOA.

III.2.1- Gestão Estratégica

A primeira competência de gestão essencial à inovação é a gestão estratégica. Os novos produtos, serviços ou processos só podem contribuir para reforçar a competitividade da empresa se estiverem interligados aos fatores críticos de sucesso do negócio. A iniciativa de

inovação deve estar alinhada à orientação estratégica da empresa, contribuindo assim para a sua competitividade. Porém, a inovação não confere, por si só, uma vantagem competitiva. Constitui antes uma potencial fonte de geração de vantagens competitivas para a empresa.

Para delinear a sua estratégia de inovação, a organização deve considerar não apenas os interesses próprios, mas também as potenciais áreas de inovação das outras entidades, uma vez que a competitividade é sempre uma medida relativa. Desta forma, pode adotar uma estratégia reativa ou pró-ativa, dependendo de uma série de fatores tais como: dimensão e acesso ao mercado, nível de competitividade, extensão dos recursos da empresa e proteção da inovação. Enquanto na orientação pró-ativa a inovação visa antecipar e intervir nas tendências do negócio para colocar a empresa numa posição competitiva mais favorável, na orientação reativa a inovação serve apenas para preservar os níveis relativos de competitividade e para defender a empresa de pressões externas. A estratégia de inovação pró-ativa pode ser implementada de várias formas:

- Desenvolvimento pioneiro de inovações: visa assegurar a liderança tecnológica e comercial da empresa que desenvolve novos produtos, serviços ou processos;
- Autonomização de novos negócios: Consiste na estruturação de unidades autônomas para dar continuidade a projetos de inovação próprios;
- Apropriação de competências alheias: Ocorre quando a empresa adquire ou associa-se a entidades detentoras de know-how considerado relevante para os projetos de inovação;
- Defesa da posição competitiva: Visa defender a porção de mercado ou aaaaaoocdo oç 1508.96

- Tecnologias marginais: aquelas que não contribuem ou podem vir a contribuir significativamente para os fatores críticos do sucesso do negócio.

Nas áreas tecnológicas marginais, a empresa deve limitar-se a seguir as inovações tecnológicas alheias e adotar as que considera adequadas aos seus produtos, serviços ou processos. Do contrário, nas tecnologias centrais, a empresa deve liderar a inovação se forem verificadas três condições:

- Sustentabilidade da liderança da inovação: se a empresa consegue sustentar o avanço tecnológico, deve procurar liderar a inovação;
- Benefício das vantagens da iniciativa: a empresa deve poder usufruir da capacidade de ser a primeira a inovar, se beneficiando da originalidade de seus produtos, serviços ou processos inovadores;
- Defesa das vantagens da iniciativa: a empresa não deve ficar prejudicada por ser a primeira a inovar, devendo para isso estar apta a explorar a nova tecnologia.

Portanto, a gestão estratégica pode e deve suportar o raciocínio para a tomada de decisão, cabendo aos dirigentes da empresa a responsabilidade pelo direcionamento e pela estratégia de inovação.

III.2.2- Gestão de Projetos

Os projetos devem ser conduzidos dentro de práticas e modelos, a nível de planejamento, execução e controle, de forma que se possa apurar eventuais desvios e tomar as medidas corretivas no prazo mais rápido possível. Desta forma, é recomendada a divisão do projeto nas suas três vertentes: técnica, comercial e organizacional, conforme ilustrado na Tabela III.1:

Tabela III.1- Controle dos projetos de inovação. Adaptada de FREIRE (2002).

Data	Desenvolvimento Técnico	Desenvolvimento Comercial	Desenvolvimento Organizacional
01/2007	Início do projeto	Conceito Comercial	Equipe de projeto
03/2007	Conceito técnico	Pré-estudo de mercado	Modelo de gestão do projeto
05/2007	Estudo de viabilidade técnica	Estimativas Econômicas	Operacionalização da equipe
08/2007	Protótipo laboratorial	Estudo de Mercado	Ponto de situação com o líder
09/2007	Testes laboratoriais	Teste de Mercado	Requisitos de gestão
10/2007	Protótipo operacional	Conceito comercial final	Conceito organizacional
12/2007	Testes operacionais	Plano de negócio	Equipe para a introdução

Data	Desenvolvimento Técnico	Desenvolvimento Comercial	Desenvolvimento Organizacional
02/2008	Versão final	Plano de <i>Marketing</i> e Vendas	Estruturação da introdução

Outro fator importante para o controle dos projetos é a análise da progressão financeira do mesmo, estimando de uma forma dinâmica o tempo de recuperação do investimento (*payback*). Através da definição e acompanhamento de variáveis, é possível quantificar o fator de retorno dos novos projetos.

Por último, o controle da execução do projeto serve ainda para determinar a continuidade do desenvolvimento do novo produto, serviço ou processo. Se os desvios encontrados forem excessivos, repetitivos e se não houverem perspectivas de correção adequadas, é preferível abandonar o projeto, evitando desta forma prejuízos maiores numa iniciativa de pouco futuro.

III.2.3- Gestão Funcional

Para o sucesso de um projeto de inovação, é preciso traduzir as orientações globais em políticas funcionais concretas, de forma a enquadrar as atividades criativas e operacionais de todos os colaboradores. Portanto, as atividades de inovação devem ser suportadas pelas políticas funcionais da empresa. Para garantir que os projetos de inovação estejam alinhados com a estratégia da empresa e de forma a explorar recursos e equipes indivisíveis, algumas decisões relativas às novas idéias e projetos devem ser tomadas (e integralmente suportadas) pelo topo da hierarquia.

Tipicamente, as organizações adotam uma divisão funcional das tarefas nos níveis inferiores de suas hierarquias. Por isso, as estratégias são normalmente transpostas para políticas de gestão funcionais, cabendo aos responsáveis de cada função executar as atividades sob sua alçada em coordenação com a atuação das funções restantes. Neste contexto, também a estratégia de inovação é difundida por toda a organização através da atribuição de responsabilidades às várias áreas funcionais, competindo aos respectivos diretores assegurar que a implementação seja bem-sucedida.

Para lidar com a incerteza associada ao desenvolvimento dos novos produtos, serviços ou processos, todos os projetos de inovação necessitam de uma liderança efetiva. É necessário que os líderes dos projetos tenham uma clara perspectiva dos objetivos a serem alcançados, e reúnam as condições fundamentais à correta implementação da estratégia de implementação.

Uma vez escolhida uma liderança para o projeto, é necessário construir a equipe que irá executar as múltiplas atividades requeridas ao longo do ciclo de inovação. Por definição, uma

equipe é um conjunto de pessoas com competências e conhecimentos complementares que partilha a responsabilidade por alcançar um objetivo comum.

No contexto da inovação, as equipes tendem a apresentar um dimensionamento que varia de 2 a 20 pessoas. Se a complexidade do projeto justificar o envolvimento de um número maior que 20 pessoas, é recomendada a divisão da iniciativa de inovação em vários subprojetos.

III.2.4- Gestão da Mudança

Por definição, inovação implica mudança. Mas nem sempre as organizações estão preparadas para implementar uma estratégia baseada nos novos produtos, serviços ou processos. Embora as mudanças sejam potencialmente geradoras de benefícios, também geram ansiedade e reatividade, em função da desconfiança no sucesso da inovação. Os aspectos positivos da mudança pela inovação são claramente perceptíveis, porém os aspectos negativos não o são. A Tabela III.2 descreve a origem das reações negativas às mudanças e a medidas corretivas.

Tabela III.2- Reações negativas à mudança pela inovação. Adaptada de FREIRE (2002).

Reação	Origem	Medidas Corretivas
Racional	Incompreensão dos objetivos e da estratégia	Explicar os objetivos e estratégias de forma mais clara
	Percepção de que a inovação é desnecessária	Descrever o que aconteceria caso a empresa não inovasse
	Falta de confiança nos projetos de inovação	Indicar exemplos próprios e alheios de projetos de inovação bem-sucedidos
	Expectativa de consequências negativas para a empresa	Envolver todos os níveis da organização para gerar empenho
Emocional	Receio de assumir riscos sem a aprovação dos superiores	Incentivar a assumpção de risco e explicitar as regras de aprovação
	Receio de ser alvo de críticas de superiores e colegas	Envolver os superiores e colegas em avaliações intermediárias dos projetos
	Receio de perda de poder	Promover a estabilidade do emprego

A gestão da mudança requer portanto a formação de equipes polivalentes dedicadas à execução das múltiplas atividades da inovação, além do envolvimento de líderes (gerentes) de projeto competentes. Tais pessoas que assumem a responsabilidade pela condução dos novos projetos são denominadas agentes da mudança. São necessários vários agentes de mudança, nos diferentes níveis da hierarquia.

A integração da gestão da mudança em seus vários níveis- topo da hierarquia, liderança dos projetos e equipes de inovação- são essenciais aos projetos de inovação.

III.3- Ciclo de Implementação e Gestão da SOA

Além da adoção de um modelo global para gestão da inovação, que alinhe o ciclo de inovação com as competências de gestão, existem ainda fases específicas e fundamentais para a implementação e gestão da SOA dentro de uma organização. A Figura III.3 representa as fases componentes deste ciclo:

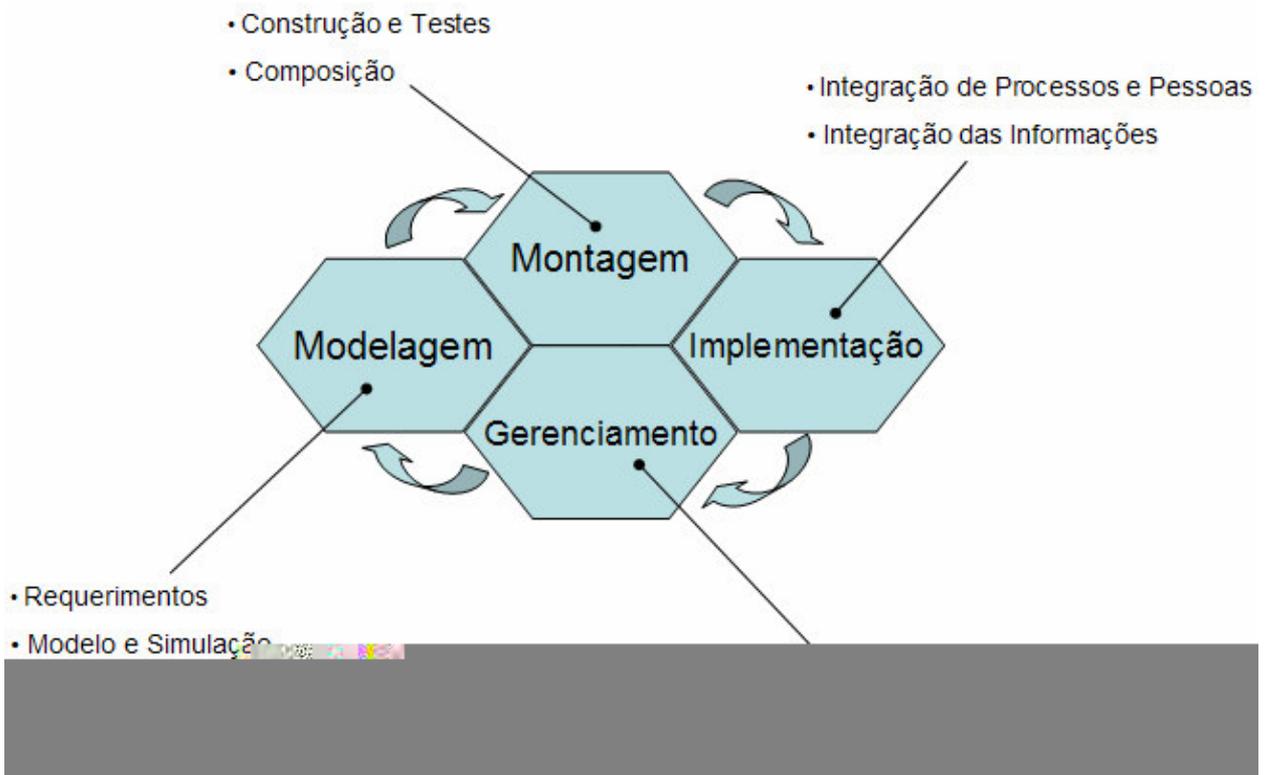


Figura III.3 - Ciclo de implementação

organização. A Tabela III.3 descreve quais são os estágios de adoção da SOA dentro de uma organização.

Tabela III.3 - Estágios para adoção da SOA na organização. Fonte: GANCI et al. (2006).

Estágio	Descrição
Estágio 1: Adoção Discreta	Compreende o uso individual de Web Services em uma particular unidade de negócio ou determinada aplicação.
Estágio 2: Adoção Parcial	Integração de algumas funções de negócio, porém não necessariamente a organização como um todo.
Estágio 3: Integração Total	Transformação total da empresa; integração vertical e horizontal entre as unidades de negócio. Desenvolvimento de novos sistemas e aplicativos de negócio.
Estágio 4: Colaboração com Parceiros	Estende o modelo de negócios ao mundo exterior, compartilhando e integrando funções de negócio com os parceiros.

A seguir serão detalhadas cada uma das fases componentes do ciclo de implementação e gestão da SOA.

III.3.1- Fase de Modelagem

Entende-se por Modelagem o processo de captura dos desenhos de negócio a partir de um entendimento claro dos seus objetivos e requerimentos. Estes são traduzidos numa especificação de processos de negócio, contendo as metas e premissas, permitindo a criação de um modelo. Em muitos casos, a complexidade dos negócios impede a modelagem através de ferramentas simples como documentos de texto ou diagramas. A captura do desenho de negócio usando técnicas e ferramentas (*software*) especializadas permite a visualização dos vários cenários do negócio e parametrização dos mesmos. Desta forma, o processo pode ser então simulado usando os parâmetros definidos para prever o efeito que terão no negócio e nos sistemas de TI. Se os resultados não forem de acordo com os objetivos de negócio, a definição dos processos pode ser refinada. A modelagem também captura indicadores-chave de desempenho (métricas de negócio), que são importantes medidores do negócio. Por exemplo, é possível incluir uma métrica para as compras realizadas em um determinado mês. Estes indicadores-chave são a entrada (*input*) para a montagem da aplicação, podendo inclusive ser monitorados para capturar dados críticos que permitem medir se os objetivos de negócio estão sendo atingidos.

III.3.2- Fase de Montagem

O processo de Montagem consiste em converter os processos de negócio desenhados na etapa de Modelagem em um sistema de informação que irá implementar o desenho. Durante este processo, ocorre uma busca nas aplicações, sistemas e interfaces existentes na tentativa de encontrar os componentes que atendam o desenho de negócio. Algumas aplicações irão atender perfeitamente, outras terão que ser redesenhadas e outras terão que ser reorganizadas para atender os requerimentos definidos. Este processo irá gerar um novo aplicativo baseado em componentes heterogêneos. Por último, a fase de Montagem deve incluir ainda um conjunto de políticas e condições que irão controlar como as aplicações irão operar no ambiente de produção.

III.3.3- Fase de Implementação

A fase de Implementação contempla a criação do ambiente para hospedagem e implementação das aplicações. Isto inclui a resolução das dependências das aplicações, condições operacionais, requerimentos de capacidade e restrições de integridade e acessibilidade. Diversas preocupações são relevantes para a construção deste ambiente de hospedagem, tais como a presença da atual infra-estrutura suportando as aplicações e os serviços já existentes. Além disso, devem ser consideradas as plataformas apropriadas para hospedar as interações lógicas, fluxos de processo, serviços e informações lógicas.

III.3.4- Fase de Gerenciamento

A fase de gerenciamento compreende as atividades, tecnologia e *software* utilizados para gerenciar e monitorar os aplicativos e processos de negócio implementados no ambiente de produção. A monitoração é um elemento crítico para garantir que os aplicativos e sistemas de TI estejam sendo executados com bom desempenho, mantendo os níveis de serviço e disponibilidade necessários para atingir as metas de negócio estabelecidas. Ocorre durante esta fase a monitoração da requisição dos serviços e a medição do tempo de resposta dos mesmos.

A fase de gerenciamento inclui ainda as tarefas, tecnologia e *software* utilizado para gerenciar e monitorar os serviços e processos de negócio executados no ambiente de produção. A monitoração realizada na fase de gerenciamento permite a verificação dos registros (*logs*) dos sistemas, de forma a identificar falhas nos vários componentes e serviços, bem como restaurar o estado operacional dos sistemas. O monitoramento é um elemento crítico para garantir que os sistemas de TI estão *on-line*, de forma a atingir os requerimentos do negócio. A fase de gerenciamento também contempla a manutenção de rotinas, administração e segurança dos aplicativos, recursos e usuários, sendo de muito relevância para a

implementação dos projetos de SOA dentro das organizações de uma maneira controlada e avaliada.

III.4- Modelo Proposto para a Implementação e Gestão da SOA

Conforme descrito no Capítulo I.1, um modelo organizacional pode ser considerado como uma série de componentes, dentro de um sistema aberto. Este sistema permitirá a organização realizar o trabalho exigido para a execução da estratégia proposta. Desta forma, a primeira competência de gestão essencial para o modelo de implementação da SOA é a gestão estratégica. Esta iniciativa deve estar totalmente alinhada à orientação estratégica da empresa, de forma a contribuir para a sua competitividade, se tornando um fator crítico de sucesso do negócio. Desta forma, o modelo proposto utiliza os conceitos de arquitetura corporativa definidos no Capítulo I.1.4 de forma a planejar a TI da corporação por meio do mapeamento e entendimento dos processos, sistemas, pessoas e tecnologias, viabilizando o trabalho e planejamento conjunto das unidades de negócio e a organização de TI.

O novo modelo deve considerar como a nova arquitetura proposta irá interagir com os indivíduos que trabalham para a organização, e como os impactará. Esta é denominada perspectiva social-cultural. Conforme visto anteriormente no Capítulo III.2.4, embora a mudança seja potencialmente geradora de benefícios, também gera ansiedade e reatividade, em função da desconfiança no sucesso da nova arquitetura. Portanto, a gestão de mudança é de fundamental importância dentro do modelo proposto. Para lidar com tal incerteza, os projetos de implementação da SOA necessitam de uma liderança efetiva, que deve ter uma clara perspectiva dos objetivos a serem alcançados. Além disso, é necessário construir a equipe que irá executar as atividades requeridas ao longo de cada uma das quatro fases componentes do ciclo de implementação e gestão da SOA, definindo suas atividades e responsabilidades de forma alinhada com a estratégia global. Esta estratégia deve ser integralmente suportada pelos mais altos níveis hierárquicos dentro da organização. Como suporte a esta perspectiva social-cultural, o modelo proposto define a utilização da gestão funcional, descrita anteriormente neste Capítulo.

No Capítulo III.3 descrevemos as quatro fases específicas e fundamentais para a implementação e gestão da SOA dentro de uma organização. Estas fases devem ser conduzidas dentro de práticas e modelos, a nível financeiro, de planejamento, execução, e controle, de forma que se possa apurar eventuais desvios e tomar as medidas corretivas no prazo mais rápido possível. Esta metodologia propõe a utilização do conceito de gestão de projetos, de forma a controlar sua execução e determinar se o mesmo atingiu as métricas de sucesso estabelecidas. A Figura III.4 ilustra a metodologia proposta neste estudo para a implementação e gestão da SOA.



Figura III.4- Metodologia proposta para a implementação e gestão da SOA.

CAPÍTULO IV- ESTUDO DE CASO: INTEGRAÇÃO DE SISTEMAS HETEROGÊNEOS DE DUAS EMPRESAS DE VAREJO UTILIZANDO A SOA

Neste capítulo, abordaremos o estudo de caso da integração de dois sistemas heterogêneos de duas empresas de varejo, que se fundiram através da aquisição da Empresa B pela empresa A, utilizando a metodologia proposta no Capítulo 3 para a implementação e gestão da SOA.

IV.1- Necessidade de Negócio

Uma determinada empresa líder em seu segmento (comércio varejista), que iremos doravante chamar de Empresa A, recentemente adquiriu outra empresa (Empresa B), como parte de sua nova estratégia de negócios. Ambas as empresas utilizam sistemas distintos, para a realização de atividades semelhantes.

A Empresa A tem como foco o comércio varejista, apresentando uma cadeia de centenas de lojas. A empresa B também tem como foco vendas através de comércio eletrônico (Internet). Através da aquisição da Empresa B, a Empresa A pretendeu expandir suas vendas para a Internet, através do comércio eletrônico, complementando sua área de atuação.

O nível estratégico da Empresa A solicitou ao seu departamento de TI que tornasse viável a visualização das informações de vendas da Empresa B, pelos mesmos sistemas disponíveis na Empresa A, de forma a acompanhar e monitorar de perto os resultados obtidos com os novos negócios. Além disso, foi solicitado que o cadastro de produtos, fornecedores e controle de estoque das empresas fossem unificados, de forma que a empresa B pudesse obter as mesmas vantagens comerciais oferecidas pelos fornecedores e parceiros de negócio da Empresa A.

O departamento de TI da Empresa A identificou quais seriam as alternativas possíveis para atender esta nova necessidade de negócio, e definiu quais os parâmetros deveriam ser analisados para a escolha da melhor alternativa. Através da análise deste parâmetros, o departamento de TI da Empresa A optou por utilizar a metodologia para implementação e gestão da SOA como alternativa para a integração das funções de negócio e definição dos novos sistemas que irão suportar ambas as empresas, elaborando um projeto piloto dentro da organização.

IV.2- Fase de Levantamento e Definição das Alternativas

Durante esta etapa inicial, o departamento de TI da Empresa A realizou um levantamento dos sistemas de informação das Empresas A e B. Os sistemas de TI existentes na Empresa A foram classificados em dois grupos distintos:

- 1) Frente de loja, composto pelos PDV (Pontos de Venda) e aplicativos de loja;
- 2) Retaguarda de loja, composto pelos sistemas de catálogo de produtos, fornecedores e clientes (Catálogo) e pelo sistema de informações de vendas (Flash de Vendas).

As Tabelas IV.1 e IV.2 ilustram as aplicações de Negócio (retaguarda de loja) identificadas de ambas as empresas, de forma a atender os novos requerimentos :

Tabela IV.1 - Aplicações de negócio da Empresa A.

Aplicação	Finalidade
Catálogo de Vendas	Sistema desenvolvido através da tecnologia IBM Lotus Notes, apresenta as funções de cadastro de clientes (CRM), cadastro de produtos e cadastro de fornecedores. Ilustrado na Figura IV.1.
Flash de Vendas	Sistema desenvolvido através da tecnologia IBM Lotus Notes, apresenta a consolidação de vendas. Integrado com o correio eletrônico da Empresa A, recebe e consolida as informações de venda dos sistemas de frente de loja, através de visões, e emite reportes periódicos via e-Mail. Ilustrado na Figura IV.2.

Tabela IV.2 - Aplicações de negócio da Empresa B.

Aplicação	Finalidade
Sistema B	Sistema WEB desenvolvido através da tecnologia JAVA, apresenta as funções de Comércio Eletrônico e controle de estoque e disponibilidade de produtos.
Resumo de Vendas	Sistema desenvolvido utilizando o banco de dados relacional Oracle, permite a consolidação do movimento de vendas do Sistema B, histórico fiscal e emissão de reportes.

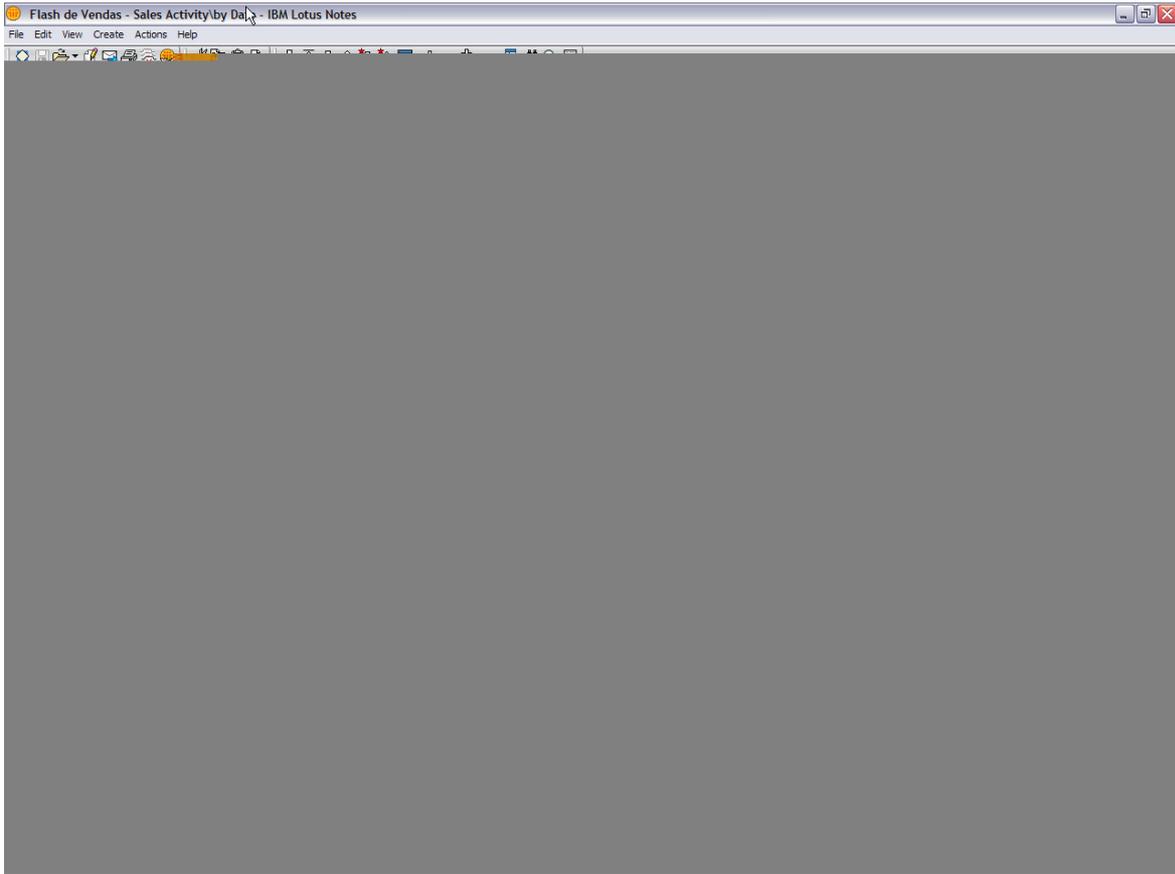


Figura IV.1- Sistema Flash de Vendas.



Figura IV.2- Sistema Catálogo de Vendas.

Após algumas reuniões de estudo, o departamento de TI da Empresa A identificou quais seriam as alternativas possíveis para atender a nova necessidade de negócio da Empresa A, sendo elas:

- 1) Desenvolvimento de um novo sistema de informação comum para ambas as empresas;
- 2) Disponibilização dos atuais sistemas utilizados na Empresa A para a Empresa B;
- 3) Disponibilização dos atuais sistemas utilizados na Empresa B para a Empresa A;
- 4) Criação de um novo sistema de informação baseado em componentes e funções de negócio já existentes, através de integração dos sistemas através de padrões de interoperabilidade.

Ao longo da fase inicial de levantamento e definição, foram estabelecidos quais os parâmetros deveriam ser levados em consideração para a escolha da melhor alternativa:

- 1) Custo Inicial: custo inicial para atender a solução, composto por aquisição de servidores, licença de *software*, horas de desenvolvimento, gerência de projeto, etc... ;
- 2) Custo a Longo Prazo: composto pelo valor mensal da manutenção dos componentes da solução por um período de quatro anos, considerado este como período de vida útil dos sistemas de informação e dos equipamentos (*hardware*) necessários;
- 3) Treinamento: custos associados ao treinamento dos usuários dos sistemas atuais na nova solução;
- 4) Satisfação do usuário final: desempenho do novo sistema, comparado com o desempenho dos atuais;
- 5) Viabilidade técnica: Viabilidade técnica para a execução da solução;
- 6) Esforço de implementação: Prazo para a implementação e esforço requerido.

A partir dos parâmetros estabelecidos, foi montada a Tabela IV.3 a fim de comparar dentro das alternativas possíveis aquele que melhor atenderia a nova necessidade de negócio:

Tabela IV.3 Comparação entre alternativas e parâmetros de implementação.

Parâmetros	Alternativa 1	Alternativa 2	Alternativa 3	Alternativa 4
1) Custo Inicial	Alto	Baixo	Baixo	Alto
2) Custo a Longo Prazo	Alto	Baixo	Baixo	Baixo
3) Treinamento	Alto	Média	Média	Baixo
4) Satisfação do usuário				

Através da análise da tabela acima, pode-se verificar que os custos de implementação, manutenção e treinamento da alternativa A seriam extremamente elevados, se comparados às outras alternativas. Além disso, o desenvolvimento de um novo sistema de informação iria exigir um longo tempo para as fases de levantamento, desenvolvimento e testes, o que tornaria demasiadamente alto o esforço de implementação. As alternativas 2 e 3 apresentaram um baixo custo inicial, pois seriam disponibilizados sistemas existentes, já adquiridos e pagos. Porém, ambas alternativas exigem necessidade de treinamento, sendo este necessário para os usuários dos sistemas da Empresa B (no caso da alternativa 2) e para os usuários dos sistemas da empresa A (no caso da alternativa 3). A viabilidade técnica desta solução foi considerada média, pois apesar dos sistemas serem semelhantes, nem todos os módulos existentes nos sistemas da Empresa A apresentam correspondência nos sistemas da Empresa B, e vice-versa, tornando também alto o esforço de implementação.

Desta forma, a análise da Tabela IV.3 demonstrou que a Alternativa 4 apresentaria um custo inicial alto, por se tratar de um projeto piloto na organização utilizando uma tecnologia nova. Porém, este custo seria compensado em uma perspectiva de longo prazo, pois a utilização dos atuais sistemas tornaria baixos os custos de manutenção. Esta alternativa eliminaria também a necessidade e os custos associados ao treinamento dos usuários, apresentando alta viabilidade técnica, pois poderia ser implementada através do uso das novas tecnologias abertas disponíveis. A alternativa 4 exigiria um médio esforço de implementação, pois os sistemas atuais não precisariam ser redesenhados, apenas a eles seriam adicionadas novas funcionalidades permitindo sua integração.

Desta forma, a fase de levantamento e definição das alternativas permitiu identificar que a alternativa 4, representada pela criação de um novo sistema de informação baseado nos componentes e funções de negócio já existentes, através de integração dos sistemas através de padrões de interoperabilidade, seria a melhor forma de atender os novos requerimentos de negócio da Empresa A.

IV.3- Aplicação da Metodologia

De forma a viabilizar a implementação dos novos requerimentos de negócio de forma controlada, o departamento de TI da Empresa A optou por utilizar o modelo descrito no Capítulo III.4 para a implementação e gestão da SOA.

IV.3.1- Gestão Funcional e de Mudança

Inicialmente, a Empresa A estruturou uma equipe de profissionais para atender as novas necessidades. A Tabela IV.4 descreve como esta equipe foi estruturada, de acordo com as funções e respectivas atividades:

Tabela IV.4- Gestão funcional do projeto.

Função	Descrição das Atividades
Analista de Negócios	Responsável pela simulação dos diversos processos de negócio para promover o alinhamento das metas organizacionais, de acordo com o que foi estabelecido pelo nível estratégico da Empresa A. Deve também selecionar os melhores processos de negócio a serem mantidos, integrados e/ ou substituídos (cada elemento do processo de negócio poderá ser automatizado como um ou mais serviços).
Arquiteto de TI	Responsável por modelar os processos de negócio em uma ferramenta de orquestração, alinhado com as definições realizadas pelo Analista de Negócios. Em paralelo, o time de arquitetura técnica trabalha os requisitos suplementares do sistema (dados, segurança ou desempenho, por exemplo) e levanta os riscos técnicos do projeto através de testes preliminares. Como resultado, uma arquitetura executável é produzida como resultado desta tarefa. Neste estudo de caso, foi utilizada como ferramenta para modelagem de negócios o software IBM Websphere Business Modeler Basic versão 6.0.1.
Gerente de Projeto	Responsável pela liderança efetiva do projeto, através da coordenação dos prazos das atividades a serem executadas e também por garantir a comunicação entre todos os membros da equipe. Responsável ainda por todo o controle financeiro e de execução do projeto. Deve ter uma clara perspectiva dos objetivos a serem alcançados.
Desenvolvedores (Programadores)	Após uma fase de maturação dos requisitos, o sistema é implementado e testado funcionalmente, pelo time de programadores. Os sistemas das Empresas A e B passam a ser não mais uma peça monolítica, mas um conjunto de serviços (Web Services) que juntos formam workflows (processos de negócio derivados dos modelos BPM dos analistas de negócios). O sistema é colocado em ambiente de testes de homologação até que atinja estabilidade para ser colocado em produção.

IV.3.2- Gestão de Projeto

O projeto de implementação foi conduzidos dentro das práticas e modelos definidos no Capítulo III.1.1, a nível de planejamento, execução e controle. Através da utilização da ferramenta *Microsoft Project*, foi possível o mapeamento gráfico de todas as etapas do projeto e sua visualização através do Gráfico de Gantt, conforme ilustrado na Figura IV.3.

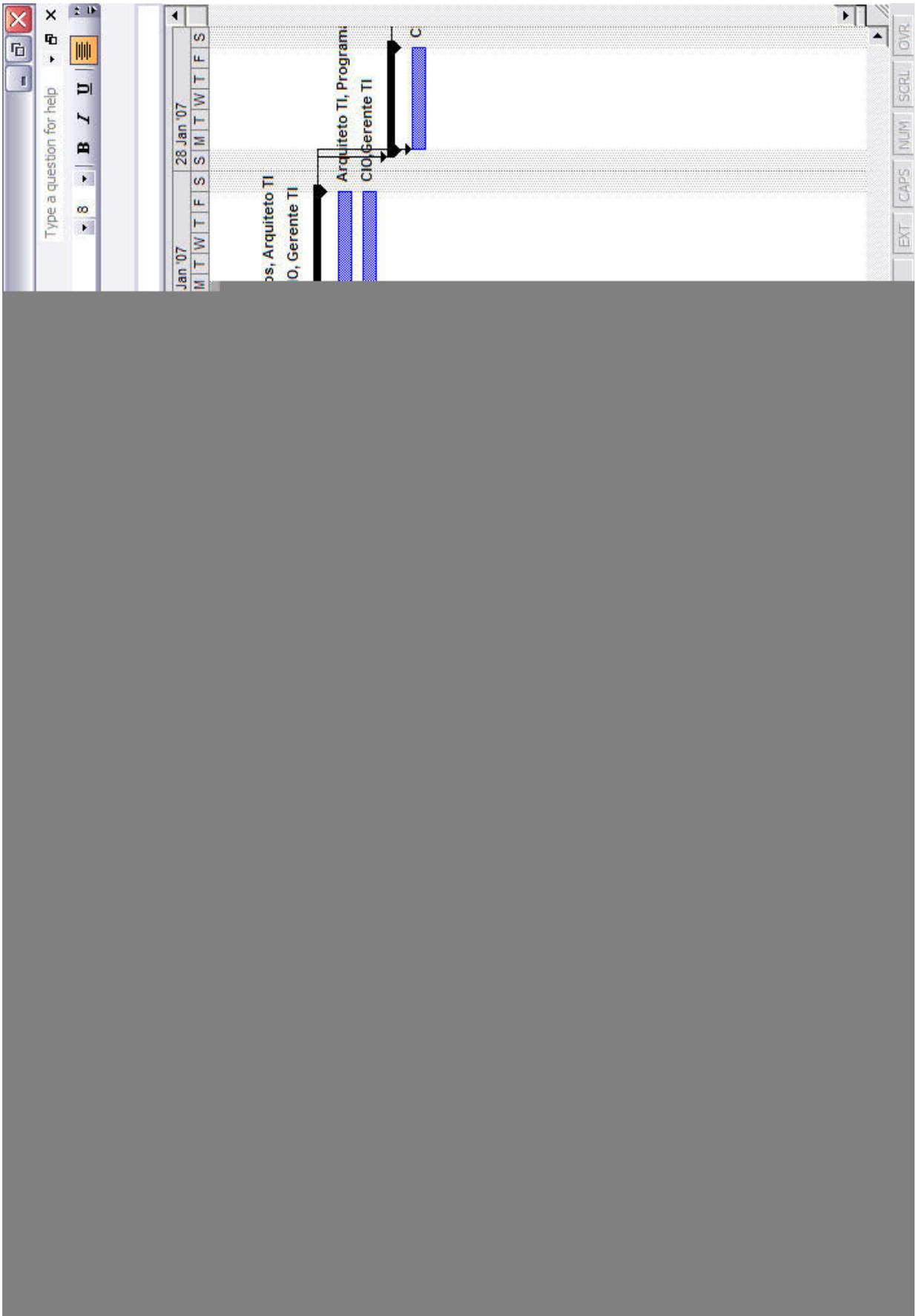


Figura IV.3- Visualização das etapas do projeto.

IV.3.3- Gestão Estratégica

Conforme visto anteriormente, uma competência de gestão essencial à inovação é a gestão estratégica. A nova arquitetura implementada só pode contribuir para reforçar a competitividade da empresa se estiver interligada aos fatores críticos de sucesso do negócio, definidos pelo nível estratégico das organizações.

A partir da definição dos novos requerimentos de negócio pelo nível estratégico da Empresa A, o analista de negócios e o arquiteto do projeto realizaram a modelagem dos processos, de forma a atender os objetivos traçados:

1. Visualização das informações de vendas da Empresa B, pelos mesmos sistemas disponíveis na Empresa A, de forma a acompanhar e monitorar de perto os resultados obtidos com os novos negócios;
2. Unificação do cadastro de produtos e fornecedores das Empresas A e B, de forma que a empresa B pudesse obter as mesmas vantagens comerciais oferecidas pelos fornecedores e parceiros de negócio da Empresa A.

Durante a fase de modelagem, a captura dos desenhos de negócio atuais foi realizada através da utilização da ferramenta *IBM Websphere Business Modeler*, que permitiu a visualização dos vários cenários do negócio a partir de um entendimento claro dos novos objetivos e requerimentos. As Figuras IV.4, IV.5 e IV.6 ilustram o resultado da modelagem dos fluxos de processo de negócio da Empresa B.

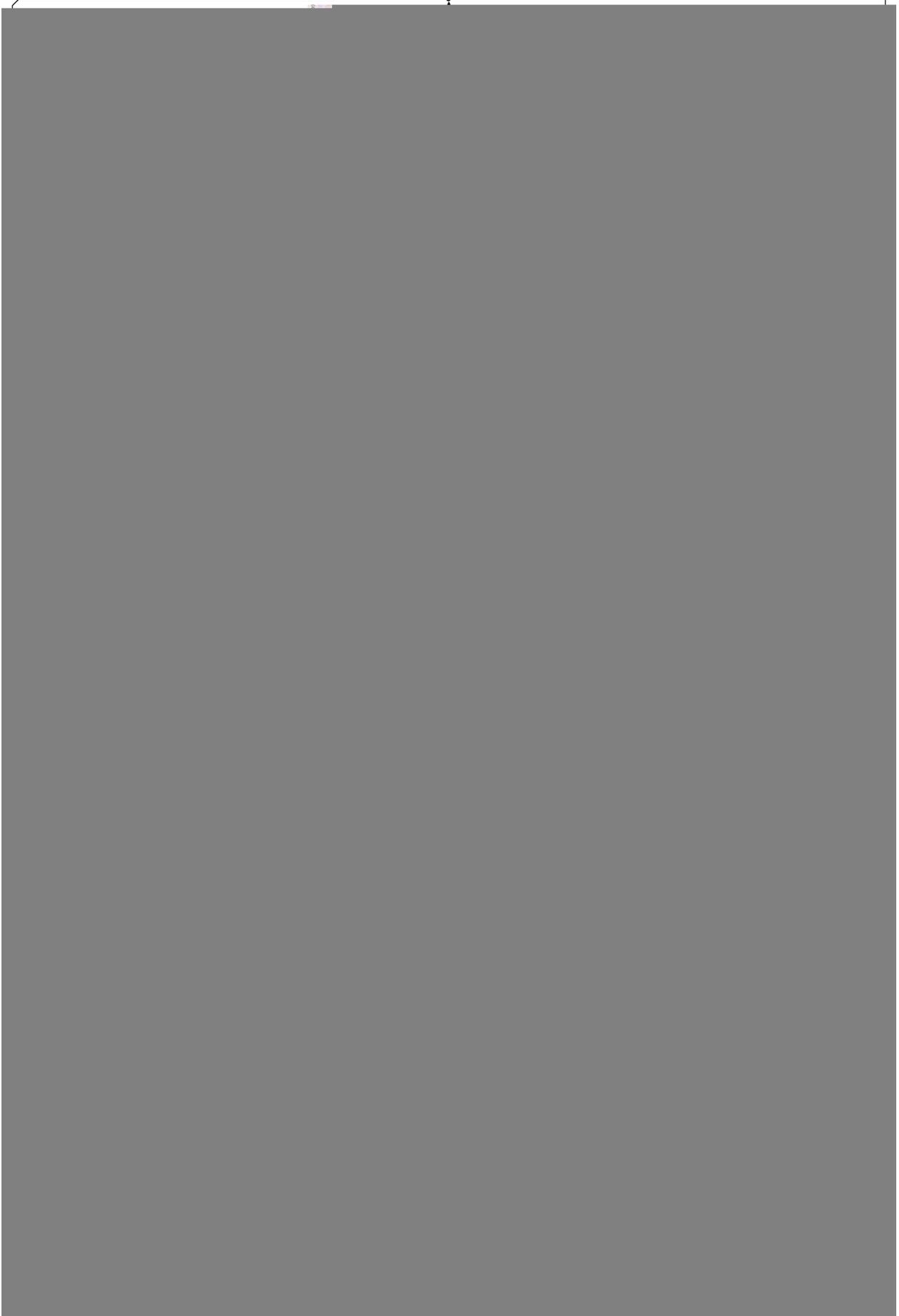


Figura IV.4- Modelagem do fluxo do processo de pedidos de cliente

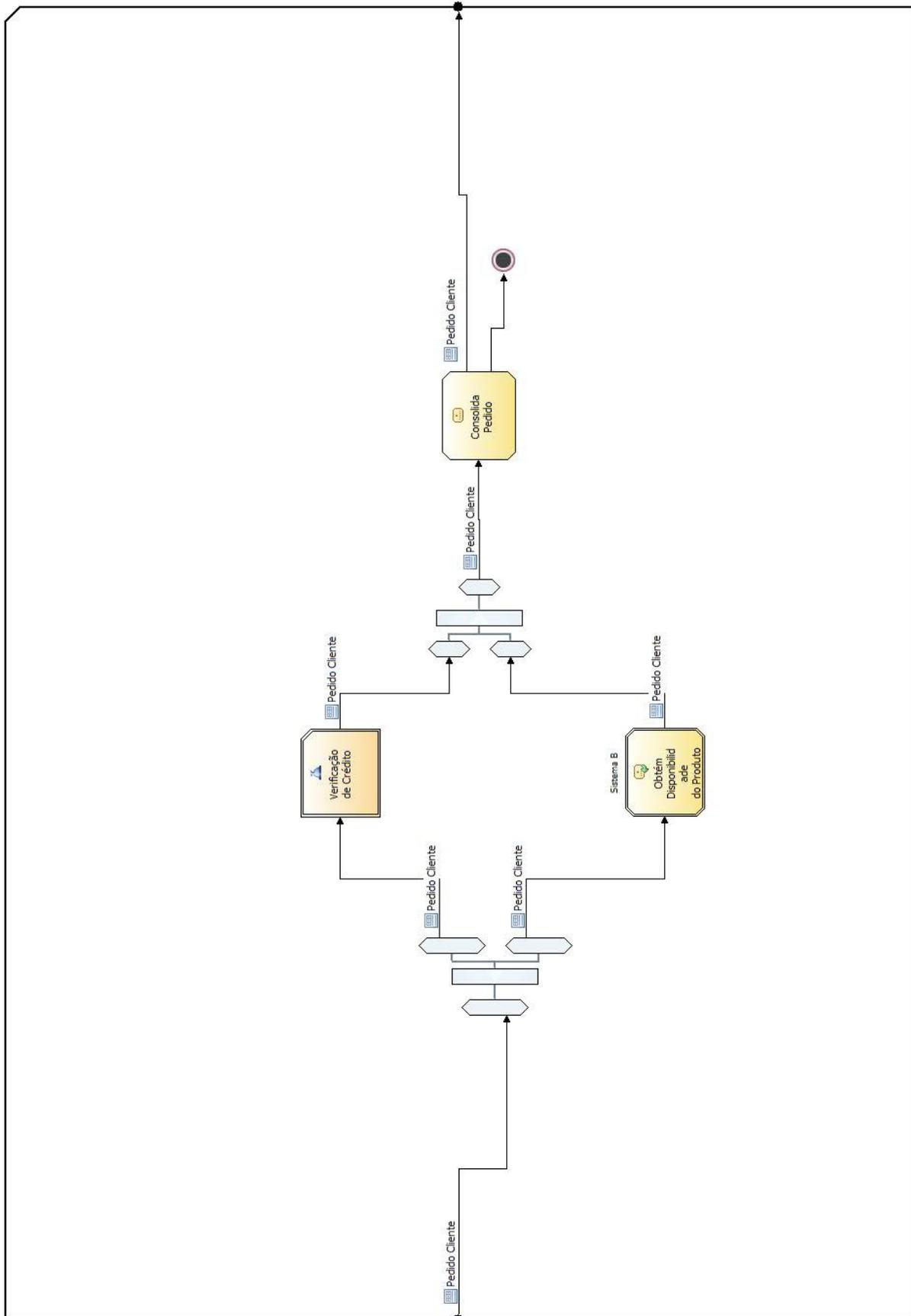


Figura IV.5- Modelagem do fluxo do processo de verificação dos pedidos.

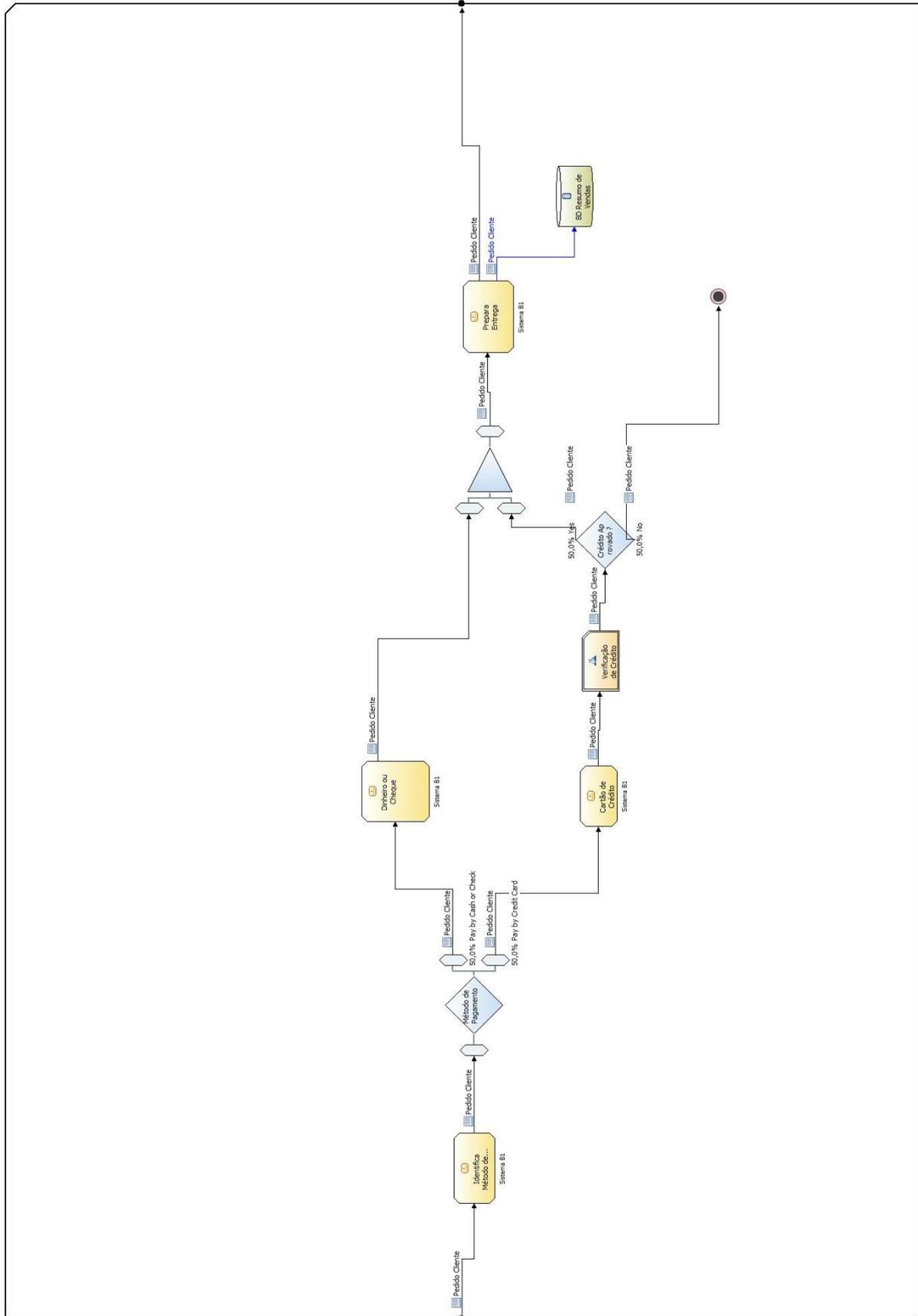


Figura IV.6- Modelagem do fluxo do processo de pagamento.

IV.3.4- Fase de Montagem

O processo de Montagem consiste em converter os processos de negócio desenhados na etapa de Modelagem no sistema de informação que irá implementar o desenho. Durante este processo, ocorreu uma busca nas aplicações, sistemas e interfaces existentes na tentativa de encontrar os componentes que atendessem o desenho de negócio, de forma a montar o novo aplicativo baseado em componentes heterogêneos.

Através da modelagem dos processos realizada, foi possível verificar cada um dos serviços dentro de cada aplicação que deveriam ser disponibilizados e compartilhados, de modo a atender às novas necessidades de negócio da Empresa A. A Tabela IV.5 ilustra as funções selecionadas após a fase de modelagem para serem implementadas como serviço:

Tabela IV.5 - Interfaces a serem implementadas como serviços.

Função Atual	Descrição	Sistema	Tipo	Dados Processados
Trata Pagamento	Grava Pedido do Cliente no Banco Resumo de Vendas	Sistema B (Empresa B)	Consumidor	Cod_Loja Data Valor
Verifica Produto	Obtém a disponibilidade do produto em estoque	Sistema A (Empresa B)	Consumidor	Cod_Prod
Disponibilidade de Produto	Informa a disponibilidade de determinado produto em estoque	Catálogo de Vendas (Empresa A)	Provedor	Cod_Prod Disponibilidade
Atividade de Vendas	Consolida as informações de vendas (volume) por loja	Flash de Vendas (Empresa A)	Provedor	Cod_Loja Data Valor
Cadastro de Produtos	Lista todos os produtos existentes na Empresa A	Catálogo de Vendas (Empresa A)	Provedor	Vetor (<i>Array</i>) de Produtos

IV.3.3- Fase de Implementação

Conforme visto anteriormente no Capítulo III.3.3, a fase de implementação contempla a criação do ambiente para hospedagem e implementação das aplicações. De forma a ilustrar a utilização da tecnologia dos Web Services e ao mesmo tempo apresentar um dos produtos da fase de implementação, ao longo deste estudo de caso foi desenvolvido o serviço para atender as requisições de disponibilidade de produto na aplicação Catálogo de Vendas da Empresa A.

De forma a atender tais requisições, a aplicação Catálogo de Vendas da Empresa A apresenta uma função denominada *getProductAvail*, que recebe o código de determinado produto e retorna a disponibilidade do mesmo em estoque. A Figura IV.7 ilustra o código, desenvolvido em linguagem *Lotus Script* dentro da aplicação Catálogo de Vendas:

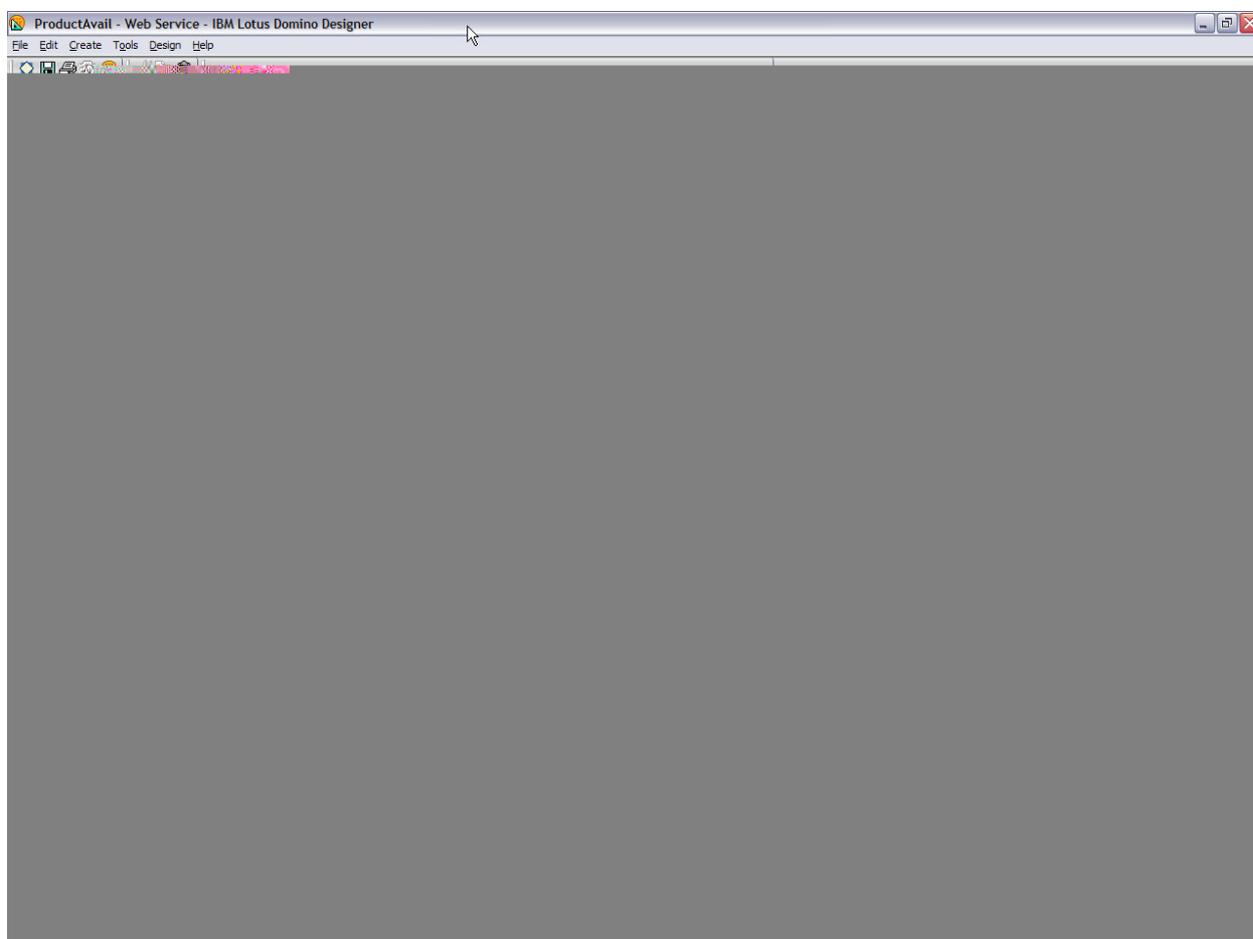


Figura IV.7 - Função GETPRODUCTAVAIL existente na aplicação Catálogo de Vendas.

Um Web Service chamado GETPRODUCTAVAIL foi implementado dentro da aplicação Catálogo de Vendas da Empresa A, permitindo estender as capacidades desta consulta para outros sistemas. As Figuras IV.8 e IV.9 ilustram respectivamente o método e o arquivo WSDL utilizados para prover o serviço GETPRODUCTAVAIL, definido para atender as requisições de disponibilidade de determinado produto:



Figura IV.8 - Método GETPRODUCTAVAIL implementado na aplicação Catálogo de Vendas.

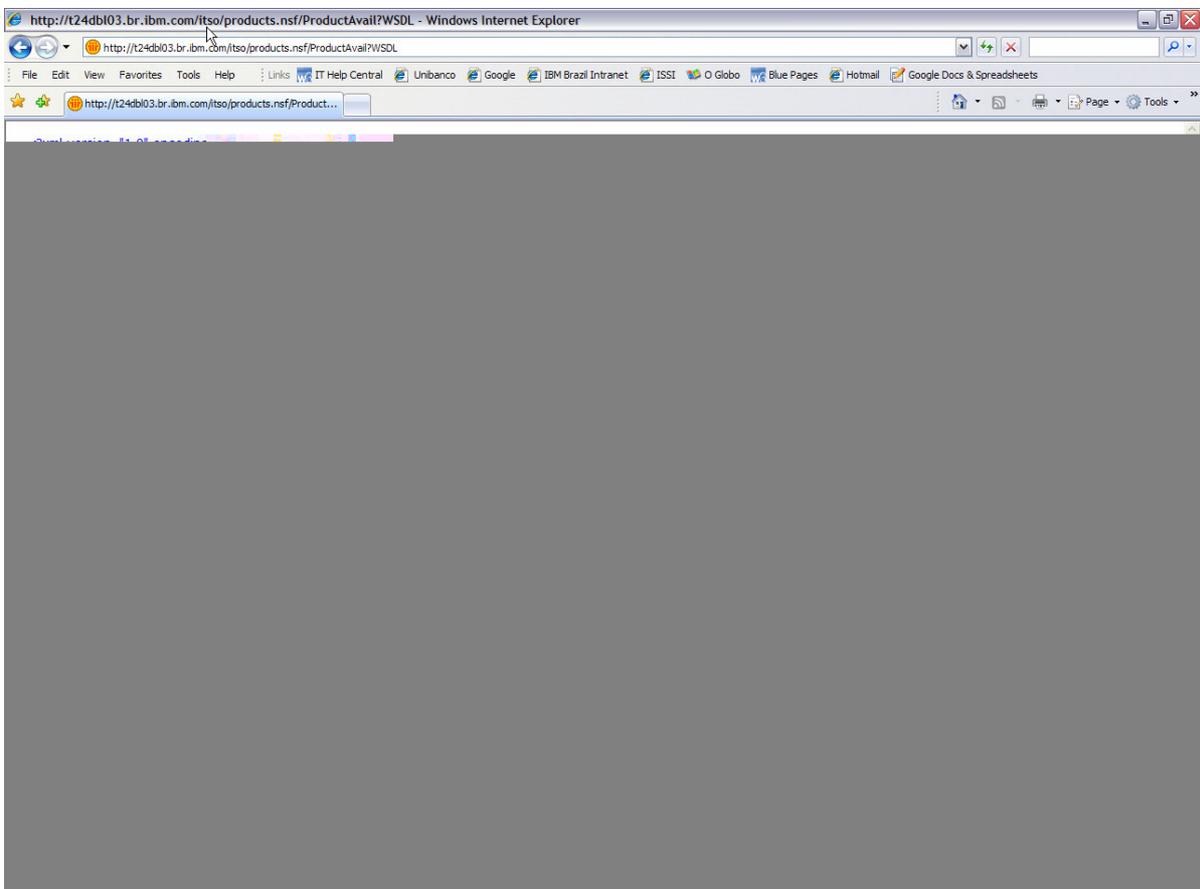


Figura IV.9 - Arquivo WSDL gerado para o método GETPRODUCTAVAIL.

Os exemplos abaixo exemplificam o formato da requisição e da resposta SOAP para o método GETPRODUCTAVAIL, ilustrando o funcionamento das requisições e respostas dos métodos através do protocolo SOAP:

Exemplo de requisição SOAP para um produto de código 45000:

```

soapenv:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:DefaultNamespace">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GETPRODUCTAVAIL
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <PRODUCTNUMBER xsi:type="xsd:string">45000</PRODUCTNUMBER>
    </urn:GETPRODUCTAVAIL>
  </soapenv:Body>
</soapenv:Envelope>

```

Exemplo de resposta SOAP para um produto de código 45000:

```

soapenv:Envelope
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:GETPRODUCTAVAILResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:DefaultNamespace">
      <GETPRODUCTAVAILReturn
xsi:type="xsd:string">No</GETPRODUCTAVAILReturn>
    </ns1:GETPRODUCTAVAILResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

CONCLUSÃO

A evolução tecnológica e em particular a revolução da informação vem gradativamente evoluindo a sociedade industrial para uma sociedade da informação, baseada na TI. Esta deve ser vista como um meio para que as organizações obtenham significativas vantagens em seus segmentos de mercado, devendo para isso estar alinhada com a estratégia da organização e permitir a sua viabilização.

Mas a constante evolução da TI requer das organizações flexibilidade para mudar ou criar processos de negócios com extrema rapidez, de forma a atingir os seus novos objetivos. Porém a rigidez, inflexibilidade e falta de integração dos tradicionais sistemas de informação não os torna adequados e esta nova realidade.

Os desafios de negócios deste ambiente estão criando uma série de obrigações de TI que exigem um novo ambiente operacional sob demanda, com base na Arquitetura Orientada a Serviços (SOA). A busca por uma maior integração e reutilização de *software* já existentes, aumento da colaboração com parceiros e fornecedores, alinhamento da estratégia de negócio com a estratégia de TI das organizações combinada com a emergência dos padrões abertos, vem criando um ambiente no qual os Web Services estão se posicionando como a plataforma padrão para os novos projetos de desenvolvimento e integração de sistemas.

Assim, semelhante à mudança de paradigma que ocorreu começo dos anos 90 do século XX, movendo os sistemas monolíticos dos mainframes para uma arquitetura cliente-servidor e distribuída, hoje tem-se a gradativa mudança da arquitetura de sistemas de informação distribuída para a SOA, através da implementação dos Web Services. Deste modo neste trabalho desenvolveu uma aplicação da SOA integrando sistemas de informação heterogêneos, através de um estudo de caso em ambiente de colaboração entre duas empresas de comércio varejista.

De forma contrária às mudanças de paradigma anteriores, a migração para a SOA não propõe descartar e reescrever os sistemas e aplicativos existentes. Conforme apresentando **no** estudo de caso, os Web Services permitiram exportar através de serviços reutilizáveis as funcionalidades dos sistemas existentes em duas empresas de varejo, viabilizando através de padrões abertos definidos o acesso a estas funcionalidades. Através da utilização da SOA, foi possível que os códigos que constituem os serviços pudessem ser reutilizados da próxima vez que a equipe de desenvolvimento precisar de uma função específica para um novo aplicativo a ser desenvolvido.

Da forma como foi implementado, o novo sistema, gerado a partir do desenvolvimento de todos serviços identificados, está alinhado com o negócio. Se o negócio mudar, o ciclo é realizado em termos de manutenções evolutivas. Entretanto, esta dinâmica pode ser realizada de forma muito mais ágil que os antigos sistemas monolíticos, pois o novo sistema está

implementado como um conjunto de serviços, dirigido por fluxos de negócio. Os serviços podem então ser reagrupados muito mais facilmente para se adaptar às mudanças, de forma mais ágil e com menor custo.

Além disso, a modelagem dos processos de negócio através de uma ferramenta de modelagem tornou possível a visualização dos processos de negócio da nova empresa adquirida, permitindo a modificação dos mesmos com alto grau de abstração em relação à tecnologia.

Apesar dos benefícios obtidos através da implementação da SOA de acordo com o modelo proposto, tais benefícios devem ser contextualizados. Dependendo do tamanho e da complexidade da empresa, pode não haver a necessidade de integração entre sistemas primários, impossibilitando que o modelo proporcione grandes benefícios. Portanto, é fundamental descobrir se existem aplicativos redundantes e mal integrados que poderiam ser consolidados ou eliminados em consequência da adoção da SOA, proporcionando desta forma que o modelo traga benefícios. Esta é considerada uma restrição técnica da implementação do modelo. Além disso, empresas pequenas ou empresas extremamente descentralizadas talvez não consigam justificar uma equipe central de gerentes de projeto, arquitetos e desenvolvedores, se tornando desta forma uma restrição de custo do modelo.

Sugere-se como trabalhos futuros a criação dos Web Services provedores e consumidores definidos durante a fase de Montagem, que não foram implementados neste protótipo. Além disso, sugere-se a criação de um registro global público (UDDI business registry), no qual toda a informação sobre os Web Services da Empresa A ficará disponível para consultas em geral, permitindo que os requisitantes consigam localizá-los através do repositório UDDI.

Quanto maior for o escopo e a utilização dos Web Services, maior será o volume desta troca de mensagens. É importante portanto garantir o envio e recebimento destas mensagens. Para este propósito, sugere-se a implementação da tecnologia ESB (Enterprise Service Bus). Diversos fabricantes de *software* oferecem ferramentas para implementação de ESB, dentre eles pode-se destacar o IBM WebSphere Enterprise Service Bus.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ALECRIM, E. **Linguagem XML**. 30 de Junho de 2003 (atualizado em 03 de Janeiro de 2006). Disponível em: <<http://www.infowester.com/lingxml.php>>. Acesso em: 26 de Janeiro de 2007.
- [2] ALKALAY, A. **Open Source**. MiniPaper Series Ano 2- Dezembro de 2006. IBM Academy of Technology no Brasil.
- [3] ATHENA. **AP1- Introduction to Service- Oriented Interoperability (SOI)** .versão 5. 2006.
- [4] AZEVEDO A.L. **A Emergência da Empresa Virtual e os requisitos para os sistemas de Informação**. p.208-225, Dezembro de 2000.
- [5] BIEBERSTEIN N. et al.; **cl B** h7.92474 0 Td (.)Tj 3.12187 0 Td ()Tj 6.72402 0 Td (e)Tj 6.00359 0 Td (t)

- [18] LAURINDO, F.J.B. et al. **O papel da tecnologia da informação (TI) na estratégia das organizações**. Gest. Prod., São Carlos, v. 8, n. 2, 2001. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X2001000200005&lng=en&nrm=iso>. Acesso em: 06 de Fevereiro de 2007. Pré-publicação. doi: 10.1590/S0104-530X2001000200005
- [19] LAURINDO F.J.B.; MESQUITA M.A.; **Material Requirements Planning: 25 Anos de História - Uma Revisão do Passado e Prospecção do Futuro**, v.7, n.3 engenharia de Produção da Escola Politécnica de São Paulo, Dezembro de 2000. p.320-337.
- [20] LIGUORI, L.F. **Enterprise Architecture: Planejando TI de Maneira Corporativa**. MiniPaper Series Ano 2 Nº 20- Agosto de 2006. IBM Academy of Technology no Brasil.
- [21] MARKS E.A.; WERRELL M.J.; **Executive's Guide to Web Services**, 1 ed. John Wiley & Sons, 2003. 224 pp.
- [22] NADLER, D.A. et al. **Arquitetura Organizacional: A Chave para a Mudança Empresarial.**; 2 ed., capítulo 2, Rio de Janeiro, Ed. Campus, 1993.
- [23] ROBICHAUX, J. **Practical Web Services in IBM Lotus Domino 7: What are Web services and why are they important ?**. 07 de Novembro de 2005 (atualizado em 21 de Novembro de 2006). Disponível em: <<http://www-128.ibm.com/developerworks/lotus/library/web-services1/>>. Acesso em: 26 de Janeiro de 2007.
- [24] RODRIGUEZ, M.V.; FERRANTE, A.J. **Tecnologia de Informação e Mudança Organizacional**. Livraria e Editora Infobook S.A., 1995.
- [25] SADTLER, C. et al.; **Patterns: SOA Foundation Service Connectivity Scenario**, 1 ed. International Technical Support Organization, Agosto 2006.
- [26] SAMPAIO, C.; **SOA e Web Services em Java**, Ed. Brasport. Rio de Janeiro.
- [27] TUSHMAN, M. L.; NADLER, D.A.; “Projetos de Organizações com Boa Adequação: Uma Moldura para Compreender as Novas Arquiteturas”. In: NADLER, D.A. et al. (eds), **Arquitetura Organizacional: A Chave para a Mudança Empresarial**, 2 ed., capítulo 2, Rio de Janeiro, Ed. Campus, 1993.
- [28] UNIVERSIDADE FEDERAL DO RIO DE JANEIRO. **XML - Extensible Markup Language**. Disponível em: <http://www.gta.ufrj.br/grad/00_1/miguel/link2.htm>. Acesso em: 6 de Dezembro de 2006.
- [29] WIKIPEDIA. **SOA**. 2 de Dezembro de 2006. Disponível em: <http://pt.wikipedia.org/wiki/SOA#Defini.C3.A7.C3.B5es_de_SOA>. Acesso em: 5 de Dezembro de 2006.
- [30] WIKIPEDIA. **Web Services**. 29 de Novembro de 2006. Disponível em: <http://pt.wikipedia.org/wiki/Web_Services>. Acesso em: 5 de Dezembro de 2006.
- [31] W3C. **About the World Wide Web Consortium (W3C)**. 2004. Disponível em: <<http://www.w3.org/Consortium/>>. Acesso em: 03 de Janeiro de 2007.

[32] YIN, R.K. **Estudo de Caso: planejamento e método**. Porto Alegre: Bookman, 2005.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)