

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCIANA BONADIO FURTADO

**DESENVOLVIMENTO DE SERVIÇOS DA WEB PARA UMA
APLICAÇÃO B2B**

MARÍLIA
2005

LUCIANA BONADIO FURTADO

**DESENVOLVIMENTO DE SERVIÇOS DA WEB PARA UMA
APLICAÇÃO B2B**

Monografia apresentada ao Curso de Ciência da Computação da Fundação de Ensino Eurípides Soares da Rocha, Mantenedora do Centro Universitário Eurípides de Marília - UNIVEM, para obtenção do grau de Bacharel em Ciência da Computação.

Orientador:
Prof. Dr. Marcos Luiz Mucheroni

MARÍLIA
2005

F992d FURTADO, Luciana Bonadio
Desenvolvimento de Serviços Web para uma aplicação B2B / Luciana Bonadio Furtado; orientador: Marcos Luiz Mucheroni. Marília, SP: [s.n.], 2005.
81 f.

Monografia (Bacharelado em Ciência da Computação) — Centro Universitário Eurípides de Marília – Fundação de Ensino Eurípides Soares da Rocha.

1. Conceitos Web 2. Serviços Web 3. Comércio Eletrônico 4. Ferramentas
5. Desenvolvimento

Luciana Bonadio Furtado

“DESENVOLVIMENTO DE SERVIÇOS DA WEB PARA UMA APLICAÇÃO B2B”

Banca examinadora da monografia apresentada ao Curso de Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do grau de Bacharel em Ciência da Computação.

Nota: _____ (_____)

Orientador: Marcos Luiz Mucheroni

1º Examinador: Edmundo Sérgio Spoto

2º Examiador: Ricardo Petruzza do Prado

Marília, 30 de novembro de 2005.

*Dedico este trabalho a Maria do Pilar Caetano Furtado,
minha avó, uma grande mulher que deixou muita saudade.
Determinadas pessoas são eternas em nossas lembranças e poderiam ser imortais.*

AGRADECIMENTOS

Meus agradecimentos pela realização deste trabalho primeiramente a Deus que durante a minha vida me dá tantas graças, me mostra tantos caminhos e me ajuda a alcançar metas e objetivos.

A Maria Santíssima, minha mãezinha do céu que sempre me acalma e mostra que ela me acompanhará por toda a minha vida, sem que eu precise me desesperar.

Ao professor Marcos Mucheroni, grande incentivador deste projeto, a quem poderei chamar sempre de amigo, por me mostrar novos caminhos durante esta pesquisa e até mesmo para minha vida profissional.

A bondade no olhar de minha mãe, Beatriz e ao lindo sorriso de meu pai, José, meus grandes exemplos de vida, presente de Deus juntamente com meus irmãos Guilherme e José Adolpho.

A garra admirável de minha avó Olívia e a serenidade e sabedoria de meu avô Alberto.

A toda a minha família materna e paterna não dá pra citar todos os nomes, pois aqui eu citaria cerca de 50 pessoas, mas em especial minhas priminhas que tanto amo Amanda, Lis Maria e Laís Maria.

A meus amigos da faculdade em especial a Larissa e ao meu grande companheiro Fernando Lazanha.

A verdadeira amizade de minhas queridas amigas Edilayne, Érica, Juliana, Marissol, Maristela e Valéria que nunca deixaram de acreditar em mim.

Ao meu namorado Alessandro, que durante este tempo juntos me trouxe mais do que um lindo sentimento, a nossa história me trouxe a percepção de que todo ser humano deve ter maturidade.

Ao padre Benedito por suas orações.

Ainda agradeço a ajuda do professor Elvis Fusco, e a enorme força do amigo George Paoli.

"Imaginemos marinheiros que, em alto-mar, estejam modificando sua embarcação rudimentar, de uma circular para outra mais afunilada... Para transformar o casco de seu barco utilizam madeira encontrada à deriva e madeira da velha estrutura. Mas não podem colocar a embarcação no seco para reconstruí-la desde o princípio. Durante seu trabalho permanecem no velho barco e lutam contra violentas tormentas e ondas tempestuosas... Esse é o nosso destino como cientistas".

(Otto Neurath)

FURTADO, LUCIANA BONADIO. **Desenvolvimento de Serviços da Web para uma aplicação B2B**. 2005. 81 f. Monografia (Bacharelado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

RESUMO

Este documento apresenta a tecnologia de Web services, seus protocolos, definições de padrões da Web, XML, C# e JAVA, como linguagens de programação que são fortes tendências na implementação de serviços da Web, introdução ao comércio eletrônico, apresentação do TPC-W na avaliação da escalabilidade de sites de comércio eletrônico e aplicativos utilizados no desenvolvimento de um Web Service aplicado ao B2B.

ABSTRACT

This paper presents the Web Services technology, your protocols, definitions of Web standards, XML, C# and JAVA, as program languages that are inclinations at the implement of services from Web, introduction to electronic commerce, TPC-W presentation on scalability analysis of electronic commerce sites and applications used on Web Service development apply to B2B

Palavras-chaves: Web Services, Comércio Eletrônico.

Lista Abreviações

API – Application Programming Interface
ASCII – American Standard Code for Information Interchange
B2B – Business to Business
B2B2C – Business to Business to Consumer
B2C – Business to Consumer
B2E – Business-to-employees
CBMG – Customer Behavior Model Graph
CE – Comércio Eletrônico
CERN – Conseil Europeen Pour la Recherche Nucleari
CLS – Continuous Linked Settlement
(csc) – computer software components
C2B – Consumer to Business
C2C – Consumer to Consumer
DER – Diagrama de Entidade e Relacionamento
DLL – Dynamic Link Libraries
DTD – Document Type Definitions
EDI –Electronic Data Interchange
ERP – Enterprise Resource Planning
E2E – Exchange-to-exchange
G2C – Government-to-citizen
HTTP – Hyper Text Transfer Protocol
IDE – Integrated Development Environment
IETF – Internet Engineering Task Force
IOS – Internetworking Operating System
JAXB - Java Architecture for XML Binding
JAXM – Java API for XML Messaging
JAXP – Java API for XML Parsing
JAXR – Java API for XML Registries
JIT- Just In Time
JSP – Java Server Pages
J2EE – Java 2 Platform Enterprise Edition
MCT – Ministério da Ciência e Tecnologia

MIT – Massachusetts Institute of Technology
MSDE – Microsoft Data Engine
PICS – Platform for Internet Content Selection
RDF – Resource Description Framework
RPC – Remote Procedure Call
SET – Security Electronic Transactions
SOAP – Simple Object Access Protocol
SSL – Security Socket Layer
TPC-W – Transmission Control Protocol
UDDI – Universal Description Discovery Integration
UML – Unified Modeling Language
URC – Uniform Resource Characteristic
URI – Unique Resource Identifier
URL – Universal Resource Language
URN – Uniform Resource Name
WIPS – Web Interactions Per Second
WSDL – Web Services Description Language
WYSIWYG – What You See Is What You Get
W3C – World Wide Web Consortium
XDR – Extreme Data Rate
XML – eXtended Markup Language

Lista de Figuras

Figura 1	Arquitetura em camadas proposta para a Web Semântica (Lee, 2001)	1732
Figura 2.	Mensagem SOAP.....	24
Figura 3.	Esquema de funcionamento do protocolo SOAP	27
Figura 4.	Funcionamento do WSDL segundo AYALA (2002)	32
Figura 5.	Estrutura de um documento WDDL.....	33
Figura 6.	Ambiente de desenvolvimento disponível no Web Matrix.....	36
Figura 7.	Método de compilação de aplicações da plataforma .Net.....	400
Figura 8.	Arquitetura do Sistema de Cotações On-Line.....	57
Figura 9.	Modelagem do Sistema Carino B2B de Cotações On-Line	57
Figura 10.	Tela Listagem de Solicitações	5858
Figura 11.	Tela de Detalhamento das Solicitações	5959
Figura 12.	Tela de Listagem de Cotações.....	600
Figura 13.	Tela de Listagem de Propostas	60
Figura 14.	Tela dos Fornecedores	611
Figura 15.	Tela dos Fornecedores	611
Figura 16.	Cotações Disponíveis.....	62
Figura 17.	Propostas de Cotações	63
Figura 187.	Fluxograma do Sistema desenvolvido	64

Sumário

Introdução	122
Capítulo 1. Apresentação do Assunto	122
1.1. Justificativa	133
1.2. Objetivos:	133
1.3. Organização do Trabalho:	144
Capítulo 2: Conceitos Atuais da Web	166
2.1. Metadados e Metalinguagem	18
2.2. Protocolos na nova Web	233
2.2.1. – SOAP	244
2.2.2. –UDDI	277
2.2.3. –URI	29
2.2.4. – WSDL	311
3. Ferramentas	355
3.1. Web Matrix	35
3.2. .Net	377
3.2.1. Diferencial na compilação	399
3.3. XML	400
3.4. C#	42
3.5. Java e JSP	44
3.5.1. API's	46
Capítulo 4: Comércio Eletrônico	48
4.1. Conceitos e Classificações	48
4.2. Visão Tecnológica	511
5. Escalabilidade	53
6. Sistema de Cotações On-Line	555
6.1. Telas da aplicação	57
6.1.1. Listagens de Solicitações	5858
6.1.2. Detalhamento e listagem das Solicitações	5858
6.1.3. Listagens de Cotações e visualização de propostas	59
6.1.4. Fornecedores	61
6.1.5. Cotações disponíveis	62
6.2. Fluxograma do trabalho Proposto	63
Conclusões	655
Referências Bibliográficas	67
<i>APÊNDICE</i>	70
<i>A. URI PELO REGISTRO OFICIAL IANA</i>	70
<i>B. WSDL EXEMPLO DE DOCUMENTO</i>	75
<i>APÊNDICE C</i>	77
<i>APÊNDICE D</i>	7878

Capítulo 1: Introdução

Com o avanço da tecnologia nos últimos anos a Internet tornou-se um meio essencial de comunicação tanto no âmbito pessoal quanto profissional. A tecnologia da Informação é cada vez mais utilizada no ambiente empresarial. As novas tecnologias da informação difundiram-se pelo mundo em menos de duas décadas rapidamente, introduzindo o mundo a meios de comunicação e informação instantâneos e deixando obsoletas determinadas rotinas que passam hoje por automatizações e reconfigurações de aplicações.

Juntamente com a globalização e o comércio eletrônico, a integração entre organizações tem confirmado as novas tendências mundiais. As informações em torno de um produto ou serviço são de extrema importância e empresas que forem capazes de organizar o fluxo, agilizar a estrutura da informação e ter controle sobre ela permanecerão no mercado.

Segundo KALAKOTA e ROBINSON (2002) o sucesso dos projetos de negócios reagregados depende de um conjunto bem integrado de software e de aplicações empresariais e seu uso é fundamental para o aparecimento da economia digital.

A Internet chegou e se desenvolveu rapidamente, superou as expectativas mais otimistas e alavancou o crescimento dos serviços de criação de *Web sites*. Este é o momento para a utilização da Internet como uma forma interativa e multimídia de publicar informações qualificadas e principalmente disponibilizar serviços. Este trabalho abrange conceitos sobre as novas tendências da Web em disponibilizar serviços respeitando padrões e seguindo arquiteturas seguras.

1.1. Apresentação do Assunto

Este trabalho visa apresentar o projeto de desenvolvimento de um *Web Service* em uma indústria alimentícia auxiliando o processo de cotações de produtos, munindo-se de certos recursos e vencendo alguns obstáculos importantes.

1.2. Justificativa

Durante 10 anos trabalhando com tecnologia, desde o mais simples afazer às mais complexas atividades, levam um profissional a observar as necessidades tecnológicas empresariais.

Um sistema de cotações por si é trabalhoso, de modo a englobar contato com fornecedores envolvidos em um processo de cotação e avaliação de novos fornecedores, anotações e negociações, sem contar no custo envolvido de ligações telefônicas a estes fornecedores, que gera certo nível de stress do administrador de uma empresa ao colocar na ponta do lápis o quanto ele gasta para cotar cada produto que faz parte da sua demanda de pedidos e de seus processos de negócios.

Após o conhecimento adquirido através de algumas literaturas e o desafio de implementar um *Web Service*, a aplicação escolhida tem o intuito de minimizar os custos de um processo de cotação e negociação de compras, utilizando a Internet como o meio de comunicação entre a empresa que necessita de um produto e empresas fornecedoras, levando em consideração a ética de um processo sigiloso de compras, seja este de modo tradicional ou por um serviço atual da Internet.

1.2. Objetivos:

Este trabalho tem como objetivo desenvolver um ambiente para tratar troca de dados entre empresas (B2B) usando o conceito de *Web Services* e a Ferramenta *Web Matrix*. Por último o estudo implementa um ambiente simples que gera dados via Web no relacionamento entre duas ou mais empresas, possibilitando o acesso ao sistema de cotações tanto de fornecedores quanto de usuários e administradores de qualquer parte do mundo através da Internet.

1.3. Organização do trabalho:

Durante o desenvolvimento do sistema foi obedecida a seguinte seqüência dos eventos: aperfeiçoamento pessoal através pesquisas técnicas, treinamentos e experimentos em programação; aperfeiçoamento teórico através da coleta de dados literários, revisão ampla e atualizações de conceitos; estudo de regras de negócio; testes e migrações de bases de dados; formalização visual de requisitos através de UMLs; definição de arquitetura; desenvolvimento de diagrama de entidade relacionamento pra simplificação de implementação; definição de telas; implementação de componentes; desenvolvimento do sistema; definição de configurações; testes automatizados e funcionais.

Modelagem de DER feita com a ferramenta Erwin 4.0. Utilização da linguagem XML, com algoritmos na linguagem C#, disponíveis para o ambiente de serviços .Net e dados de empresas na área alimentícia com o uso de ferramentas especiais disponíveis na Web como o WebMatrix para agilidade na implementação.

No primeiro capítulo são abordados os motivos da escolha do projeto, quais os objetivos pretendidos a serem alcançados e o modo de organização das pesquisas técnicas e científicas.

O segundo capítulo abrange os novos conceitos da Web orientados sob padrões estabelecidos pelo W3C, a definição de metadados encaixada aos padrões, ainda são apresentados os protocolos padronizados utilizados nos *Web Services* e suas estruturas.

O terceiro capítulo apresenta as ferramentas pesquisadas, estudadas e que são utilizadas no desenvolvimento do projeto proposto, conceitos, aplicações, exemplos de código para introdução a linguagem utilizada estão contidos no capítulo três.

O quarto capítulo introduz as definições de comércio eletrônico *e-bussines* a documentação e trás a visão tecnológica das aplicações atuais.

No quinto capítulo apresenta citações de uma análise da escalabilidade de sites de comércio eletrônico.

O sexto capítulo mostra os diagramas de uso de acordo com a descrição da concepção de um sistema de cotações B2B on-line aplicado sob base de dados de uma empresa.

Por fim as Conclusões sobre pesquisas, aplicações e desenvolvimentos serão apresentadas.

Capítulo 2: Conceitos Atuais da Web

A revolução da informação teve seu início com a invenção da escrita na Mesopotâmia e desde então, tem sofrido constantes evoluções. DRUCKER (2004) afirma que a revolução da informação será uma revolução de conceitos. Esta revolução visa definir o significado da informação e qual o seu propósito, por exemplo: definir sobre os novos conceitos das tarefas que são realizadas com a ajuda da informação e das instituições e como efetuar essas tarefas.

Nos últimos anos, uma série de websites disponibilizaram para os usuários da Internet serviços de reservas, compras consultas e notícias, dentre outros. A Web caminha hoje em dia para um novo paradigma de serviços e funcionalidades.

Devido à explosão de fontes de dados associados aos inúmeros sites espalhados pela Internet, o caminho natural para as aplicações na Web é a integração de sistemas diferentes, que objetivam compilar a informação e agrupá-la para clientes potenciais.

LEE (2001), apresentou uma visão sobre os padrões e técnicas que governam a World Wide Web, inspirado no contexto de evolução das aplicações presentes na Internet, foi então apresentada a Web Semântica.

A Web Semântica não é um conceito Web separado, a visão da “nova web” pode ser considerada abstratamente como uma extensão da web atual, onde cada informação possui um significado bem definido, que permite que computadores e pessoas trabalhem em cooperação, sendo criada sobre a mesma infra-estrutura, para permitir uma nova geração de aplicações onde programas devem ser capazes de navegar por uma rede de computadores em busca de informação relevante para realizar então a interação de dados e finalmente alcançar o resultado de uma computação solicitada por algum usuário.

Seguindo conceitos do W3C, para que a Web Semântica funcione, os computadores deverão ter acesso a um conjunto de regras que possam ser utilizadas no processo de análise,

este acesso também deverá englobar coleções estruturadas de informação. A representação do conhecimento como a tecnologia da Web contém a semente de aplicações importantes, mas para atingir seu potencial pleno, deverá estar ligado a um sistema global e único, pois a propriedade essencial da Internet é a universalidade.

A arquitetura apresentada por LEE (2001), contempla a criação de um conjunto de padrões que permitem que aplicações troquem e processem informações de forma automatizada e inteligente. A arquitetura em camadas é apresentada na Figura 1:

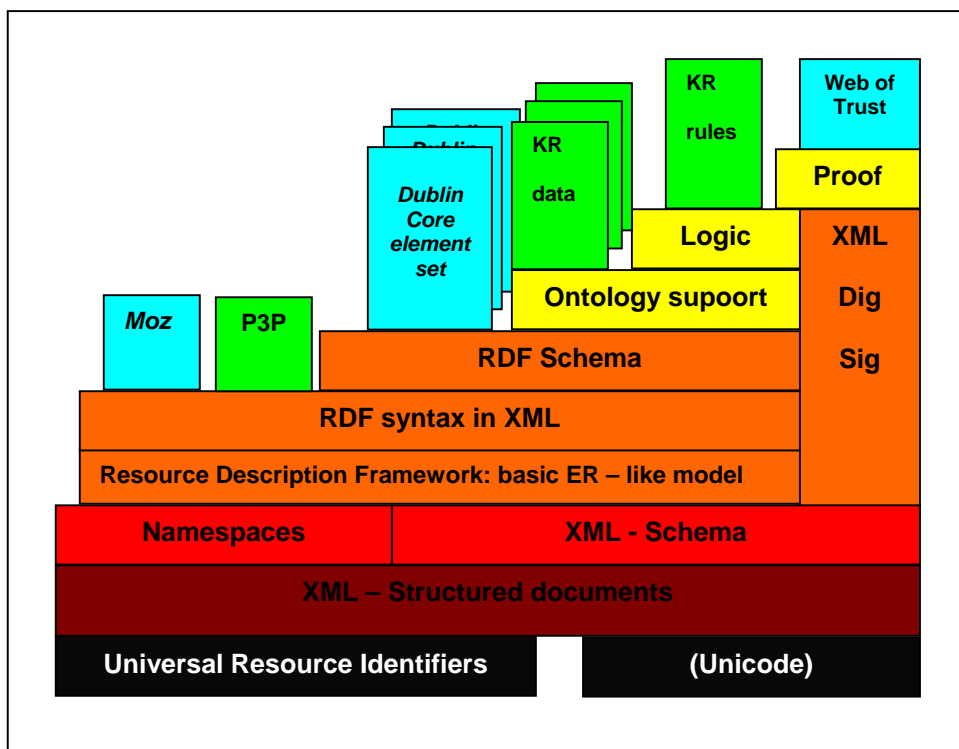


Figura 1. Arquitetura em camadas proposta para a Web Semântica (Lee, 2001)

As três camadas (URI, Unicode, XML e XML Schema/Namespaces) definem como a informação deve ser apresentada. Acima, as camadas fazem referência à semântica da informação (o significado) e os axiomas referentes a um determinado domínio. Para garantia da segurança da informação, não-repudição e reputação das fontes é utilizada estrutura de assinaturas digitais. Os conceitos básicos como ontologias, agentes de software e assinaturas digitais é o fundamento do ponto de vista tecnológico.

Para as próximas gerações de aplicações na Web, as soluções apontadas são as arquiteturas orientadas a serviços, baseadas em padrões abertos e amplamente adotados pelos principais fornecedores do mercado de tecnologia da informação através de diversos padrões da indústria e produtos, sendo a promessa de infra-estrutura para computação distribuída capaz de ser adotada em larga escala.

Os pesquisadores da Web Semântica, entretanto, consideram que os paradoxos e questões sem respostas são os preços a ser pago para que se possa obter versatilidade. Seu grande desafio é criar uma linguagem que expressem dados e regras simultaneamente, permitindo a análise dos dados e regras a eles relacionados em qualquer sistema de representação de conhecimento, para sua posterior exportação para a Web. Adicionar lógica a Web, o que significa utilizar regras para realizar inferências, escolher cursos de ações e responder perguntas, é objetivo atual da comunidade que trabalha com a nova Web.

Um processo fundamental é a união de subculturas quando uma comunicação comum e mais abrangente é necessária. Eventualmente, dois grupos independentes desenvolvem conceitos muito similares e definem que uma relação entre eles traria grandes benefícios.

A linguagem lógica e unificada habilitará os novos conceitos a se unirem, progressivamente, através da Internet. Essa estrutura permitirá que agentes especialistas (softwares) possam realizar análises significativas do conceito e trabalhos realizados pela humanidade, provendo uma nova classe de ferramentas com as quais pode-se viver, trabalhar e aprender juntos.

2.1. Metadados e Metalinguagem

A utilização de metadados com eficácia exige a adoção de formatos padrão para a representação da informação e de linguagens de marcação que permitam uma estruturação

semântica e flexível destes formatos. Os metadados são essenciais para o tratamento de vários recursos na medida em que possibilitam a denominação semântica de seu conteúdo, sua identificação e recuperação seletiva.

Metadados são dados que descrevem o conteúdo, estrutura, representação e contexto de algum conjunto de dados.

Dados sobre dados, ou dados que se relacionam com outros dados, estas são definições unânimes encontradas em alguns documentos de acordo com autores das áreas de Informática e Ciência da Informação.

No específico conceito afirmado por BERNARDINO (1998) encontra-se: “Metadado é uma abstração sobre o dado. É um dado de mais alto nível que descreve um dado de mais baixo nível”.

A definição atribuída por SOUZA, VENDRUSCO e MELO (200) é que Metadado é a descrição do recurso eletrônico ou a catalogação do dado.

ANDRADE e CARVALHO (1999) definem que “metadado é a informação descrita sobre o conteúdo e a organização do dado que pode ser recuperado, manipulado e apresentado de diferentes formas, podendo ser simples e não-estruturado tal como uma descrição textual, ou estruturado e complexo como um dicionário de dados usado para controlar múltiplos bancos de dados”. Seguindo o mesmo raciocínio, TENNANT (1998) explica que “metadado, simplificada é a informação estruturada sobre a informação”, este assegura ainda que estruturação é a questão-chave.

Os bibliotecários estão contribuindo com cientistas da informação na busca de uma solução estruturada para as indicações de categoria de dados para que os *browsers* possam encontrar as informações requeridas pelos usuários esta indicação é denominada de metadados pela autora.

Em benefício ao usuário é que os produtos e serviços de informação são projetados. Formalmente, de acordo com DEMPSEY e HEERY (1998), “metadados são dados associados a objetos, que auxiliam seus usuários potenciais para obter um conhecimento completo sobre sua existência ou características” a definição é completada com a afirmação: “metadados são conhecimentos que permitem usuários humanos e automáticos agirem com inteligência”.

Conforme o tipo e objetivo da aplicação, os autores apresentam diferentes conceitos para metadados. Para chegar a sua essência, decompondo o termo, verifica-se que o prefixo grego *meta* é definido no dicionário MICHAELIS (1998) como um termo que “exprime a idéia de depois de, sucessão”.

Atualmente o principal uso de metadados aplica-se a conceitos digitais, mas de acordo com WEIBEL (19979), metadados foram “originalmente concebidos para a criação da geração de autor nos recursos da Web, foi também atribuído para a descrição de recursos formais de comunidades como museus e bibliotecas”.

O grande impulso para a tecnologia metadados está na necessidade do melhor conhecimento de dados que as organizações mantêm e a necessidade de conhecer com mais detalhes os dados de outras organizações. A estrutura para a descrição padronizada de documentos eletrônicos que torna possível e mais efetiva a recuperação da informação na Internet originou a descrição por metadados.

Sintetizando num breve histórico a origem da utilização de metadados na Web inicia-se em outubro de 1994 quando o criador da Web, Tim Berners-Lee em colaboração com o *Counsell European Pour la Recherche Nucleari* (CERN) fundou no *Laboratory for Computer Science* do *Massachusetts Institute of Tecnology* (MIT) o *World Wide Web Consortium* (W3C).

Os objetivos do projeto eram o de tornar a Web acessível mundialmente, otimizar os recursos disponíveis desenvolvendo mecanismos semânticos para a Web e direcionar a Web para o desenvolvimento de tecnologias confiáveis, legal, comercial e socialmente.

Como enfoque de princípios aos esforços do W3C concentra-se primeiramente a *visão* que abrange a satisfação dos usuários da Web através da capacidade do W3C de identificação de requisitos técnicos, pela contribuição de pesquisadores e engenheiros sendo a Web verdadeiramente um espaço informacional.

Projetar tecnologias para a Web levando em consideração as possibilidades tecnológicas futuras e as existentes atualmente pautando em três componentes como a evolução, a interoperabilidade e a universalidade da Web é a definição do segundo princípio: o *design*.

Como terceiro princípio concentrado nos esforços do W3C está a *padronização* que produz recomendações que especificam e promovem a funcionalidade e a universalidade da Web.

Em outubro de 1995 o W3C iniciou suas atividades nas áreas política e tecnológica da Web com uma plataforma para seleção de conteúdos da Internet, a *Platform for Internet Content Selection* (PICS), que fez uso pela primeira vez de metadados na Web. Um mecanismo que comunica categorizações das páginas Web de um servidor para os clientes sem conteúdo pré-fixar um conjunto de categorias, mas permitir que diferentes pessoas e organizações classifiquem as páginas Web de acordo com seus objetivos e valores, a fim de selecionar ou restringir o acesso àquelas páginas que não vão de encontro aos seus interesses.

O papel do metadado na Web é a identificação de conteúdo, a localização de informações sem identificação de recursos, a descrição do tipo de recurso utilizado, a recuperação de dados e a utilização dos recursos de informação. Em geral, as definições sobre

as funções dos metadados referem-se que os metadados servem de elemento de ligação entre usuário e criador/criação da informação.

Ao dissertar sobre política de informação na Internet, FROEHLICH (1998) afirma que “é preciso estruturar estratégias de metadados a fim de facilitar as necessidades dos usuários” a partir da busca de informações.

Os metadados são utilizados para descrever as características de um documento e seus relacionamentos com outros documentos.

GARDNER (1997) faz considerações ao usuário, afirmando que a administração de metadados dá ao usuário maior controle de dados provendo um esquema onde um tipo de informação deriva de outro e um mapa de alocação de onde o dado está armazenado.

Dando referência ao sentido de um índice BERNARDINO (1998) afirma que “o objetivo primário do metadado é fornecer um guia completo sobre o recurso de dados”.

Chega-se à conclusão de que os metadados estruturados de acordo com funções atribuídas e sintetizados de maneira correta tornam-se grandes ferramentas para a gestão de informações digitais.

Estruturar metadados é dar-lhes uma forma sintetizada de processamento, possibilitando integração sintática em diferentes redes e sistemas de integração semântica entre diferentes representações do conhecimento.

Tendo como aplicação metadados em bibliotecas digitais TENNANT (1998) da enfoque a três categorias como metadados descritivos (indexação da fonte e descrição bibliográfica), metadados estruturais (descreve estrutura de fonte e arquivos interligados para facilitar navegação em fonte digital) e metadados administrativos (propriedades de arquivos digitais e modo de produção).

O fator que dá uma estrutura ao metadado é a predefinição da categoria que o descreve, é importante especificar os atributos ao valor associado.

2.2. Protocolos na nova Web

A idéia de transformar o desenvolvimento de aplicações em um desenvolvimento baseado em componentes não é nova. Entretanto, sua larga adoção ainda depende do desenvolvimento de tecnologias e padrões que sejam capazes de descrever “pedaços” de software de forma completamente independente de sua implementação. Um aspecto extremamente importante é a capacidade de composição dinâmica destes pedaços, realizada de forma automatizada por algum tipo de software responsável pela composição e coordenação destes serviços. Com o advento da Internet e a explosão de fontes de informação e de serviços disponíveis na rede, existem fatores que dificultam essa composição, a semântica das informações envolvidas é um destes fatores. Uma vez que não existe uma forma única de descrever os recursos na rede surgem inconsistências e ambigüidades que precisam ser tratadas de alguma forma.

Por esta razão, o desenvolvimento de aplicações baseadas em serviços distribuídos na Internet envolve a composição tanto das funcionalidades destes serviços. Existe a necessidade de introduzir semânticas nos dados trocados entre as partes do sistema, permitindo que máquinas possam racionalizar tarefas complexas utilizando componentes de software que não foram projetados para trabalhar juntos num primeiro momento.

Como forma de entender a tecnologia de componentes no contexto de uma aplicação baseada em serviços, é importante entender a arquitetura típica da tecnologia de componentes. Tomando Web Services como base, é possível definir três tipos básicos de entidades: Provedor de Serviço, Corretor de Serviço e Requisitante de Serviço.

Um determinado serviço é fornecido por um Provedor, o qual publica suas funcionalidades em algum sistema de busca por serviços que recebe o nome de Corretor de Serviços. Um Requisitante é uma entidade que irá utilizar alguma funcionalidade de um Web Service para realizar alguma tarefa mais completa. Para tal, ela entra em contato com algum

Corretor de serviço através de um padrão para descoberta (por exemplo, (UDDI)). Ao receber a resposta o Service Requester terá as interfaces padrão para acesso aos serviços que atendem às suas necessidades. Essas interfaces fazem parte da descrição do serviço que normalmente é feita em uma linguagem específica para isso (por exemplo, (WSDL)). A partir deste ponto, os serviços fornecidos pelos Services Providers poderão ser acessados utilizando um protocolo de acesso a objetos remotos (por exemplo, (SOAP) ou (OOP)).

Desta maneira fica claro que são necessários alguns padrões para definição de uma tecnologia de componentes. Estes permitem que componentes sejam descritos, acessados e publicados de forma homogênea. Como forma de contextualizar estes três padrões e apresentar as características mais comuns a cada um deles. Serão detalhados a seguir os padrões utilizados na tecnologia de *Web Services*.

2.2.1. – SOAP

O protocolo SOAP – *Simple Object Access Protocol* foi criado em 1998 por um grupo de companhias é baseado em XML e designado a troca de estruturas e tipos de dados entre aplicações em um ambiente distribuído. A comunicação é ponto-a-ponto e não existe conceito de nó central em SOAP, este protocolo é constituído de três diferentes partes:

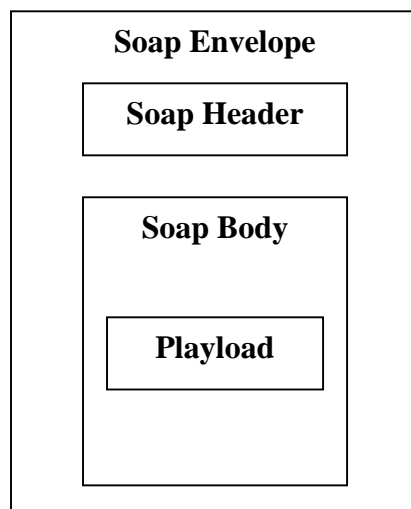


Figura 2. Mensagem Soap

- a) Envelope: descreve o conteúdo da mensagem;
- b) Codificação de tipos de dados: quais tipos estão disponíveis para serem usados em uma mensagem SOAP e;
- c) Chamadas de métodos: chamam a si mesmas em métodos com a definição de como elas são realizadas e qual o retorno de cada uma.

O SOAP é usado para fazer chamadas remotas de procedimentos (RPC) e tem como característica a independência de plataforma.

O Envelope é o elemento raiz do XML que forma uma mensagem SOAP e fornece um repositório para o Header e para o Body. As informações adicionais de processamento estão contidas no cabeçalho (ou Header) e permite transportar dados de autenticação ou controle de transação. Entradas do Header são todos os elementos filhos do elemento Header. A informação em si é o corpo da mensagem (ou Body) onde são carregadas todas as informações obrigatórias para receber o final da mensagem. Entradas do Body são os elementos filhos do Body.

Uma mensagem SOAP pode passar por Web services intermediários antes de chegar ao seu destino final e tem os seguintes atributos:

a) *encodingStyle*: indica as regras de centralização usadas na mensagem SOAP. Para partes que possuem diferentes representações de dados, este atributo define os tipos e mapeamento de dados entre as partes.

b) *actor*: indica qual é a aplicação que deverá processar a entrada no header. Este atributo só é informado no header. Um header sem o atributo actor indica ao receptor da mensagem que ele será o último Web services. Um Web service intermediário sabe que deverá processar uma informação no header através do actor.

c) *mustUnderstand*: Atributo também informado no header, ele indica se o receptor da mensagem SOAP deve interpretar o conteúdo do header. O receptor é indicado pelo

atributo actor. O mustUnderstand tem os valores 0 para indicar que o actor alvo pode, mas não precisa processar a entrada no header e 1 para indicar que o actor alvo deve entender a semântica da entrada no header e processá-la.

O protocolo SOAP é o protocolo base entre o consumidor, fornecedor e o serviço de busca.

Seguindo as definições de WANDER (2003) o SOAP habilita a interoperabilidade entre as aplicações. Ele é de grande viabilidade na interoperabilidade na Internet. Sua arquitetura está baseada no protocolo HTTP, no qual é criado um túnel pra a troca de dados. A segurança é a mesma que for possível de ser aplicada ao HTTP (SSL, SET etc).

O funcionamento do protocolo SOAP pode ser bem compreendido por meio do exemplo. Supondo que uma aplicação esteja rodando em uma intranet e ela tenha recursos de calcular e expor alguns indicadores de performance empresarial. Este recurso em especial deverá ser consumido por outras aplicações, bem como aplicações de parceiros de negócios. Por meio do SOAP, podem-se expor métodos que recebam parâmetros para obter o resultado de tais indicadores, que é representada por uma sintaxe genérica que segue:

```
boolean
    PerformanceDeVendas([in] DataInicial Date,
                        [in] DataFinal Date,
                        [out] ÍndiceDeVendasNoPeriodo double)
```

Ao ser invocado esse método receberá uma data final, com as quais o sistema buscará os dados de vendas em uma origem de dados e retornará verdadeiro ou falso, indicando o sucesso dos cálculos, com um double contendo o indicador desejado.

Se um sistema cliente realizar a chamada desse método usando SOAP o resultado http terá o seguinte formato:

```
POST
    /PerformSys HTTP/ 1.1
    Host:
    Intranet.empresax.com.br
    Content - Type:
    Text/xml-SOAP
    Content-Length:
```

```

MessageType:
Call
<SerializedStream>
<PerformanceDeVendas>
    <DataInicial>20/08/2005</DataInicial>
    <DataFinal>20/09/2005</DataFinal>
</PerformanceDeVendas>
</SerializedStream>

```

O exemplo apresentado demonstra o uso do SOAP mediante uma solicitação http, com método Post.

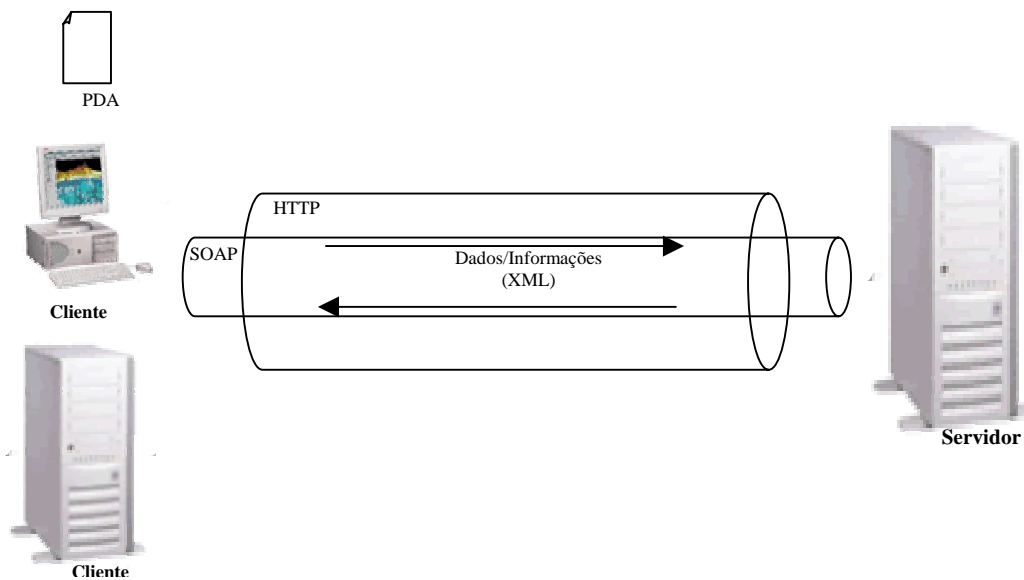


Figura 3. Esquema de funcionamento do protocolo SOAP (WANDER, 2003)

O esquema da Figura 3 demonstra o funcionamento do protocolo SOAP durante a troca de informações.

2.2.2. – UDDI

No ano de 2000 uma parceria entre as empresas Microsoft, IBM e Ariba com o objetivo de criar um padrão único para a descoberta de serviços foi desenvolvido o projeto UDDI (Universal Description, Discovery and Integration).

A especificação UDDI é um conjunto de mensagens baseadas em XML para realizar todas as operações de atualização e consulta de registros.

A linguagem de anotação definida em XML e UDDI. É fundamental para criação de metainformação característica de um *Web Service*.

Uma grande variedade de registros UDDI são agrupados em repositórios, os quais possuem uma interface/API de pesquisa para permitir uma aplicação cliente pesquisar e localizar um serviço. Para uma exemplificação, a especificação UDDI tenta organizar em categorias além de armazenar informações sobre companhias que fornecem serviços. Os chamados brokers UDDI podem ser comparados a páginas amarelas na Internet, disponibilizando um catálogo de informações sobre empresas e suas respectivas atividades.

Um Broker de serviços compatível com UDDI é capaz de organizar e receber informações descrevendo e descobrindo negócios, organizações e outros provedores *Web Services* e interface técnica que pode ser utilizada para acessar esses serviços.

Segundo BELLWOOD (2002), pelo fato de ser baseada em um conjunto comum de padrões empresariais respeitando regras de negócios, incluindo SOAP, HTTP, XML e XML Schema, a UDDI possibilita a interoperabilidade e a infra-estrutura necessária à softwares baseados em *Web Services* de ambientes onde os serviços são disponíveis publicamente e de ambientes onde os serviços são disponíveis apenas internamente na organização.

As chamadas Operadoras UDDI (UDDI Operators) são associações que disponibilizam serviços e hospedam o registro global UDDI cujas funções são:

- Publicar: Abrange como o provedor de *Web Service* registra a si mesmo e ao seu serviço no UDDI;
- Descobrir: Ações que tratam de como o cliente localizará o *Web Service* desejado;
- Ligar: Função de como o cliente conectará e utilizará o *Web Service* desejado.

As operadoras mais conhecidas são a Microsoft e a IBM que fornecem os respectivos serviços (UDDI-1, 2005) e (UDDI-2, 2005) <http://uddi.microsoft.com> e <http://uddi.ibm.com>.

A responsabilidades destas operadoras é de sincronizar a informação de registro, essa sincronização é chamada de replicação. AYALA (2002) enfatiza que as empresas publicam seus serviços e as informações relacionadas utilizando essas operadoras.

2.2.3. –URI

A identificação de um Web Service é realizada através de uma URI (Unique Resource Identifier) que são descritos e definidos usando XML. Para o cliente verificar a interface do web service basta conectar-se através de uma URI. Conectando-se a URI é possível interpretar a estrutura (serviços, parâmetros) do Web Service. É importante ressaltar que a partir da publicação do Web Service na Internet, é impossível a publicação do mesmo pelo cliente que pode ser desenvolvido por qualquer linguagem de programação que suporte do desenvolvimento de clientes Web Service.

O Elemento envelope do SOAP especifica a URI que identifica o namespace utilizado por uma requisição SOAP. É o namespace padrão de todas as mensagens SOAP.

Através de um relatório da W3C/IETF publicado como parte da Atividade URI W3C, grupos de interesse no planejamento de URI conceituam e definem a divisão do universo URI especificadamente, ressaltando o relacionamento entre o conceito de URL, URN e URI. Os conceitos são definidos através do esclarecimento de dois diferentes pontos de vista de URI, chamados de visões clássica e contemporânea.

A *visão clássica* aborda a identificação especificando que uma URI é também uma URL ou uma URN. Com a generalização desta adição de um discreto numero de classes adicionais, por exemplo, uma URI pode apontar para um metadado ao invés de ser recursiva a ela mesma, no caso, a URI poderia ser uma URC (citação). O universo URI passa a ser visto

como divisão de subespaços, sendo a divisão entre duas classes: **URL** e **URN**. Deste modo por exemplo “http:” seria uma scheme URL e “isbm:” deveria ser uma scheme URN. Alguma nova scheme poderia ser molde entre uma ou outra destas duas classes.

A *visão Contemporânea* é um esquema individual que não precisa ser molde entre uma serie de tipos de URI, tal como “URL”, “URN”, “URC”, etc. Desta maneira “http”, “urn:”, são scheme URI, o que define subespaços chamados “namespaces”. Por exemplo, a série de URN. Além disso, de acordo com a visão contemporânea, o termo “URL” não se refere a uma divisão formal do universo URI. Uma URL é um tipo de Uri que identifica um recurso com uma representação de seu mecanismo de acesso primário.

A abrangência dos períodos clássicos e contemporâneos estão no Corpo de documentos (RFCs) cobre a arquitetura URI, suas sintaxe, registro, etc. Quando se vê uma RFC que fala sobre schemes URI (por exemplo RFC 2396), também nota-se outra RFC que fala sobre schemes URL (por exemplo RFC 2717), e ainda outra RFC que fala sobre schemes URN (RFC 2276) Enquanto a RFC 2396 1.2 tenta endereçar a distinção entre URIs, URLs e URNs.

O registro oficial de nomes URI scheme é mantido pela IANA, na (IANA, 2005) <http://www.iana.org/assignments/uri-schemes> (Anexo A).

Existem URI scheme registradas e não registradas para cada qual existem documentos públicos descritos. Também pode haver uma série de URIs obsoletas, algumas até tem RFC, mas não estão inclusas na lista da IANA.

O Documento W3C/IETF ainda não aborda o uso de URIs como identificadores que não identificam na realidade pesquisas da rede, eles identificam um objeto abstrato tais como um schema XML ou um objeto físico podendo ser um livro ou até mesmo uma pessoa.

O documento não considera IRIS: a extensão da sintaxe URI em caso da ausência de ASCII.

Tais assuntos são esperados para discussão em atividades futuras com esforço consecutivo de toda a comunidade científica da Internet.

2.2.4. – WSDL

Um documento WSDL é um documento XML que descreve Web Services como conjunto de pontos de serviços que operam baseados em trocas de mensagens.

O Web Services Description Language através de seu formato XML descreve a interface de um Web Service e define um conjunto de operações suportadas pelo servidor e o formato que os clientes irão utilizar ao requisitarem um serviço. Esta comunicação entre cliente e servidor é inspirada em um contrato efetivo. Encontra-se em AYALA (2002) que através do arquivo WSDL, o cliente requisita um serviço enviando uma solicitação propriamente formatada em SOAP.

Segundo CERAMI (2002) WSDL é uma plataforma inicialmente utilizada (mas, não exclusivamente) para descrever um serviço SOAP e é independente de linguagem. Com o uso do WSDL, o cliente poderá localizar um Web Service invocando qualquer uma das funções públicas disponíveis. A partir deste conceito, afirma-se a importância fundamental em uma arquitetura Web Service que o WSDL representa, disponibilizando uma linguagem comum para descrever a plataforma e os serviços, para a integração automática dos serviços oferecidos.

As informações da Interface que descreve todas as funcionalidades disponíveis publicamente é uma das partes importantes de dados que o WSDL apresenta juntamente com as informações de tipos de dados para todas as mensagens de requisição ou de respostas, informação obrigatória sobre o protocolo de transporte a ser utilizado, e informações de endereçamento para localizar um serviço especificado.

O funcionamento do WSDL pode ser exemplificado na situação de criação de um Web Service que disponibiliza cotações na bolsa de valores para compradores.

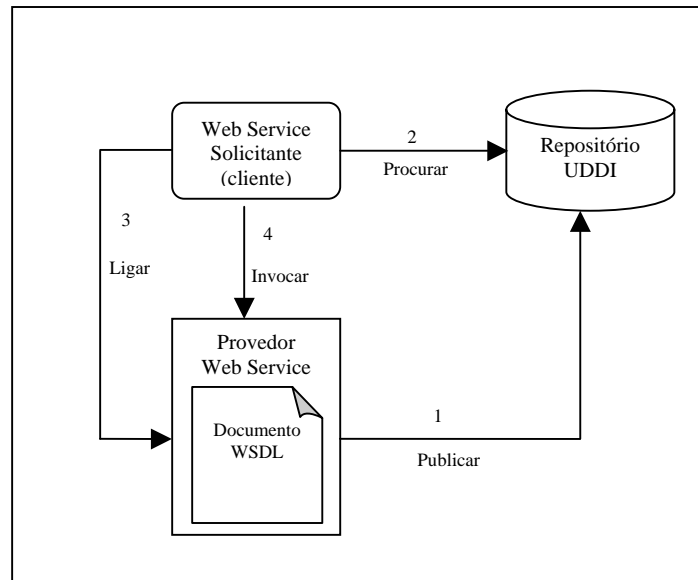


Figura 4. Funcionamento do WSDL segundo AYALA (2002)

Para descrever o serviço é necessária a criação de um arquivo WSDL, que será publicado em um repositório UDDI ou alocado no provedor do Web Service. Os clientes interessados no serviço, primeiramente, obterão uma referência do arquivo pegando uma pesquisa diretamente do provedor ou utilizando a pesquisa ao repositório. Um Web Service solicitante deverá ser criado para que este então, possa invocar o serviço fazendo uma requisição utilizando HTTP post e baseada em SOAP. A requisição é validada pelo servidor que a executa ou passa a requisição ao serviço adequado. Como uma resposta no formato SOAP, retornado para o cliente o resultado é o preço das ações na bolsa de valores.

A especificação para o documento WSDL fornece um conjunto com seis definições de elementos XML:

- O elemento *definitions* é o elemento raiz ou básico no documento WSDL.

- No elemento *types* define-se quais os tipos de dados definidos por algum XML Schema, que serão transmitidos usados nas chamadas de métodos;
- No elemento *message* em um documento WSDL define-se qual mensagem será transmitida, os parâmetros de um serviço, podendo ser de entrada ou de saída;
- No elemento *portType* que agrupa várias operações define-se quais funções serão suportadas;
- No elemento *binding* define-se como será transmitida a mensagem no cabo e quais detalhes específicos do SOAP existem para acessar os métodos;
- No elemento *service* define-se onde o serviço está localizado;

Os últimos seis elementos são aninhados.

A especificação de um documento WSDL é implementada utilizando XML Schema em conjunto com seus tipos suportados. Em uma perspectiva de negócio que requisita o fornecimento de uma interface formal, o Web Services Description Language é o elemento chave.

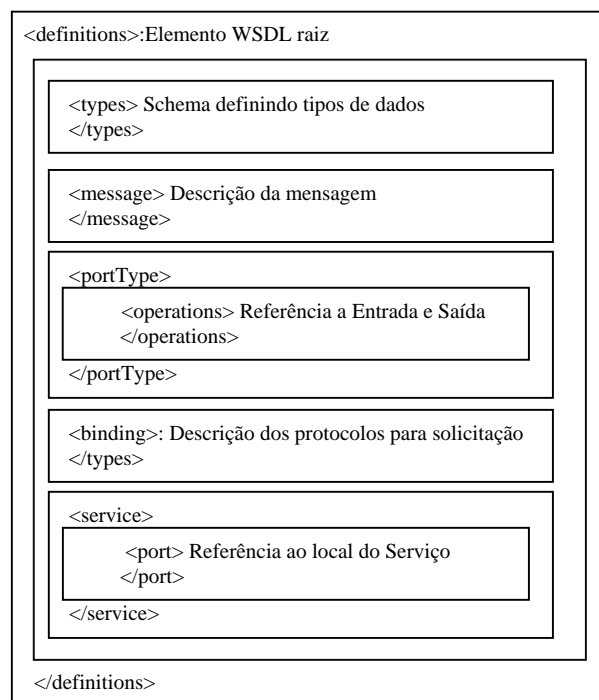


Figura 5. Estrutura de um documento WSDL

O padrão WSDL foi definido em um esforço conjunto pela IBM, Microsoft e Ariba, uma grande empresa no ramo de TI. A versão atual é a 1.1 que foi publicada em Março de 2001.

Nos primeiros Web Services disponíveis, um programa cliente que fizesse a requisição para acessar um serviço deveria ser criado a partir da análise de um WSDL disponibilizado pelo servidor.

Atualmente, existem ferramentas que geram o código necessário para realizar todas as operações disponibilizadas pelo serviço em diversas linguagens, incluindo Java, minimizando assim o contato do usuário com a geração de um envelope SOAP contendo as chamadas necessárias é mínimo. O documento XML contendo a mensagem SOAP é gerado automaticamente através do uso de objetos de um conjunto de classes geradas por esses utilitários que fornecem código com a funcionalidade especificada em um documento WSDL, da mesma forma, podendo ser criado automaticamente por meio de ferramentas específicas, uma vez que um provedor de serviços tenha sido codificado.

Nada impede, que o documento WSDL seja criado de forma manual ou semi-automática, já que tais ferramentas muitas vezes não são capazes de inserir todas as opções de personalização. Um exemplo de documento WSDL no Apêndice B.

Dá-se o mapeamento produzido a seguir através da tradução dos elementos de um documento WSDL para código Java:

Tabela 1. Comparação de elementos WSDL e Java

WSDL	Java
<types> e <portType>	Classes
<operation>	Método
<message>	Retorno de método ou parâmetros

3. Ferramentas

Aplicações via Web são serviços oferecidos por outra aplicação, cujos clientes podem agregá-las com o objetivo de formar um software que possibilite transações comerciais ou criar um novo serviço.

Nas próximas seções serão abordados conceitos das ferramentas utilizadas para a implementação do projeto, suas principais funcionalidades e características de cada aplicação.

Ainda será abordado como tópico a Linguagem de programação Java, não utilizada na implementação deste projeto, mas relacionada a implementações futuras e altamente aplicáveis a web services.

3.1. Web Matrix

O ASP.NET Web Matrix é uma ferramenta voltada ao desenvolvimento coletivo de aplicações Web a partir do ASP.NET. Esta ferramenta é inteiramente construída em C# para o .Net Framework e é distribuída em conjunto com o MSDE. Seus Recursos de desenvolvimento integram a comunidade ASP.Net ativa que permite que tanto desenvolvedores principiantes como profissionais avançados aprendam a desenvolver com o ASP.NET e ofereçam feedback sobre os novos recursos disponíveis.

A IDE da Web Matrix foi desenhada para fornecer uma interface similar ao ambiente do Visual Studio.Net. Todos os seus elementos seguem os atuais padrões visuais das novas versões de aplicações Microsoft.

O Projeto Web Matrix reúne uma comunidade ASP.NET a partir de recursos como mensagens instantâneas e integração de salas de bate-papo, além de uma ajuda baseada nessa mesma comunidade. Sendo o ASP.NET Web Matrix complementar ao Visual Studio.Net o projeto foi criado inicialmente para o desenvolvedor amador – uma vez que o VS.Net é uma extraordinária ferramenta do desenvolvedor corporativo.

Apresentado como um recurso leve e simples o download está disponível em um pacote de aproximadamente 1Mb tudo isso à disposição dos desenvolvedores, além de um Web Form Designer em WYSIWYG para criar aplicações para a Web bastando apenas arrastar e soltar os controles da caixa de ferramentas do Web Matrix.

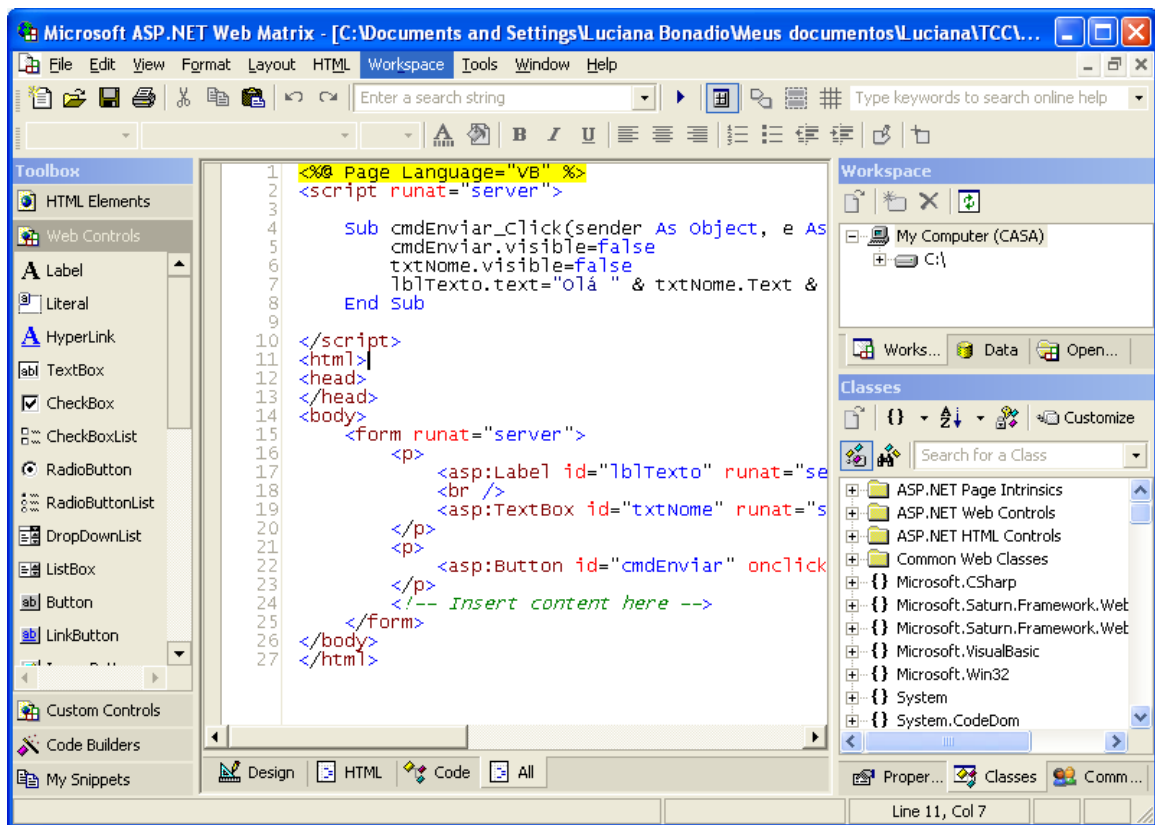


Figura 6. Ambiente de desenvolvimento disponível no Web Matrix

O projeto Web Matrix traz várias páginas de amostras de aplicações, incluindo páginas de dados, Serviços de Web, cachê de saída e páginas de login. Ele permite a criação de aplicações da Web voltadas a dados através da integração de MSDE ou bancos de dados SQL Server, bastando arrastar e soltar as tabelas de dados designer, conectando-os à página em desenvolvimento.

Além da integração de dados, existe uma integração com uma Galeria de controles on-line (ASP.NET, 2005) <http://www.asp.net/webmatrix/default.aspx?tabindex=4&tabid=46>.

E a integração com as bibliotecas de classe .NET Framework que facilita a descoberta de classes necessárias para a criação de aplicações.

Outro recurso é a área de trabalho, que é baseada em arquivos, traz uma alternativa mais leve à área de trabalho baseada em projetos.

O produto já vem acompanhado de um servidor de Web próprio, não requerendo a instalação do IIS (além de oferecer também a hospedagem a Hosts de terceiros).

O Web Matrix disponibiliza um gerador Proxy (WSDL.EXE) para Serviços de Web em XML, isto é, transforma qualquer classe VB.Net ou C# em um Serviço de Web, automaticamente. O editor de textos, em cores, ajuda a tornar o aprendizado de VB.Net e C# mais simples, o programa gera automaticamente o código das funções mais freqüentes (como enviar um e-mail, por exemplo) a partir a especificação de apenas alguns parâmetros.

A estrutura do projeto proposto a ser desenvolvido em um primeiro momento parte da criação de estruturas de armazenamento, em seguida as classes e, por último a interface com o usuário, denotando a aplicação do projeto dividida em camadas de Dados, Regras de Negócio (fluxograma) e Apresentação.

3.2. .Net

A plataforma de desenvolvimento .Net foi criada para ser um padrão de desenvolvimento na Web, permitindo a utilização de diferentes linguagens de programação. Segundo RALHA (2004), dentre as linguagens fornecidas pela Microsoft e que acompanham o produto se destacam o VB.NET, C#, VC++, Jscript .NET e J# (a partir do VS.NET 2003). O autor ressalta que a grande vantagem de se trabalhar com um ambiente único é sentida quando precisa-se mudar de linguagem, seja por opção ou por necessidade, por exemplo pra fazer a manutenção de um sistema.

.Net é o nome genérico para todos os componentes envolvidos de um modo com a plataforma como o Visual Studio .Net, que é um ambiente para desenvolvimento de programas, o .Net Framework SDK é o conjunto pra desenvolvimento incluindo os compiladores das linguagens, porém sem interface gráfica e o .Net Framework Runtime, que fornece os recursos necessários para a execução correta dos aplicativos, sem os compiladores específicos para cada linguagem.

A programação em .NET representa um amadurecimento no processo de desenvolvimento de software na qual a comercialização do mesmo passou a ser encarada como a venda de um serviço e não mais como a de um simples produtos, e a etapa de codificação tornou-se muito mais focada na resolução do problema. Este ganho substancial de tempo deve-se ao fato de que o Framework .NET já possui a maioria das rotinas que os programadores eram obrigados a codificar com frequência, evitando o clássico retrabalho além do que agora, ferramentas de desenvolvimento como Visual Studio .NET preocupam-se em gerar automática todo o código “burocrático” (aquele que não tem a ver com o desenvolvimento da aplicação em si, como o código voltado para o gerenciamento de memória).

A plataforma .Net está dividida em quatro partes principais:

- **Framework .Net** – Núcleo da plataforma .NET.
- **Produtos .Net** – Produtos Microsoft que trabalham com a tecnologia .NET e se integram por meio de XML e protocolo SOAP.
- **Linguagens .Net** – Linguagens desenvolvidas segundo a especificação CLS, descrita no próximo apêndice.
- **Serviços .Net** – Aplicativos e serviços desenvolvidos por terceiros.

3.2.1. Diferencial na compilação

Aplicações .Net diferenciam-se das aplicações tradicionais no método de compilação, ao invés de criar um código executável baseado nas instruções de máquina da arquitetura em que está sendo compilado, a plataforma .Net utiliza um conjunto de classes e métodos próprios da plataforma igual para todas as linguagens suportadas. Sendo assim, existe uma integração entre as linguagens de modo a ser possível derivar uma classe C# de um objeto escrito em Visual Basic .Net.

O arquivo em linguagem intermediária recebe o nome de assembly. Um assembly, na plataforma .Net, é entendido como uma coleção de um ou mais arquivos com tipos e recursos que formam uma unidade funcional contendo dados especiais denominados manifesto, que por sua vez define que arquivos fazem parte, que permissões precisa para ser executado, qual a versão assembly dentre outras informações.

São quatro tipos existentes de assemblies na plataforma que são descritos como:

- **Assemblies estáticos:** arquivos criados com o compilador específico da linguagem utilizada, por exemplo, csc para C#, vbc para Visual Basic e cl para C e C++;
- **Assemblies dinâmicos:** arquivos criados dinamicamente através das classes localizadas no namespace *System.Reflection*;
- **Assemblies privados:** são assemblies estáticos usados em uma aplicação específica;
- **Assemblies públicos:** Também compartilhados, são assemblies estáticos que tem um único nome público e pode ser usado por qualquer aplicação.

O Standard JIT (compilador utilizado por máquinas com muita memória RAM e gera um código otimizado através do armazenamento de ponteiros para os métodos compilados), o Econo JIT (busca maior velocidade na conversão do código intermediário para o código

nativo, porém o arquivo gerado não é otimizado e os ponteiros para os métodos são criados, este é utilizado por máquinas com pouca memória RAM) e o Pré JIT (gera um executável como um compilador de tempo de execução normal, escolhido quando a compilação não pode sofrer queda na performance) são compiladores de tempo real (Just in Time).

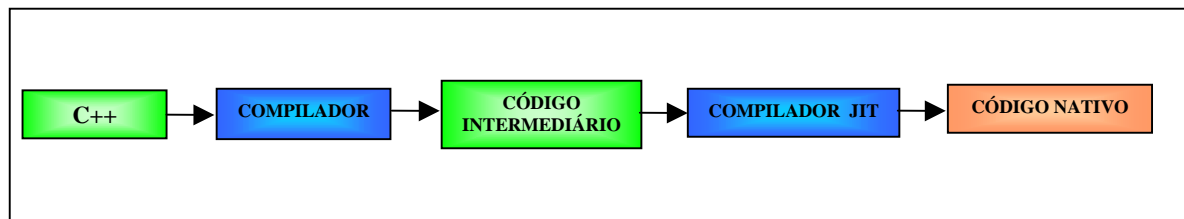


Figura 7. Método de compilação de aplicações da plataforma .Net

3.3. XML

Em unanimidade todos os autores definem XML como sendo uma espécie de HTML melhorado. A eXtensible Markup Language, ou seja, Linguagem de marcação estendida que tem por finalidade descrever informações diferentemente da HTML. Uma linguagem de marcação é um mecanismo para identificar estruturas em um documento. A especificação XML define um caminho padrão para adicionar marcações a documentos. A XML é uma linguagem de marcação para documentos contendo informação estruturada e contém algumas indicações que documentam, por exemplo, a estrutura diferenciada no conteúdo de cabeçalhos e rodapés podendo também documentar o conteúdo de uma tabela de um banco de dados.

A XML não possui tags fixas, isso permite maior flexibilidade na hora de descrever informações, já o HTML é uma linguagem que possui um conjunto enorme de tags fixas.

De acordo com LOTAR (2001), cada documento XML é um mini-banco de dados sem tags fixas, mas com regras específicas que devem ser obedecidas quando for definida a tag de um documento, sendo toda tag deve ter um tag correspondente de fechamento.

Com regras a serem obedecidas, é possível seguir a criação de uma estrutura que dá mais agilidade ao extrair uma informação necessária gerando assim páginas Web mais leves e rápidas.

É importante destacar que a XML faz diferença entre letras minúsculas e maiúsculas.

A XML foi originária da SGML, linguagem de marcação padrão generalizada usada por muitos anos. A própria HTML é uma aplicativo SGML, e a XML é um subconjunto da SGML ideal para ser usada na Internet;

É útil criar tags pra um documento XML, mas se faz necessário definir regras para criar documentos válidos. Para definir os elementos a serem usados no documento e indicar em que ordem estrutural eles deverão ocorrer, se este elemento estará vazio ou deverá conter texto, utiliza-se DTD (definição de tipo de documento) ou XDR Schema para validar a estrutura aplicada.

Em LOTAR (2001), encontra-se a previsão de que a maioria dos aplicativos no futuro utilizará em algum momento XML, o autor ainda enfatiza que XML será muito utilizada em aplicações de comércio eletrônico, pois dá possibilidade de integrar negócios de inúmeras empresas em uma só aplicação.

TURBAN e KING (2004) afirmam que aplicações de CE de qualquer tipo precisam estar conectadas aos sistemas internos de informação preexistentes. Entretanto, o ambiente padrão é útil apenas para exibição de páginas Web com elementos visuais estáticos. Para ampliar a funcionalidade de sites de CE, podem se utilizar JavaScript e outros programas, cujas ferramentas permitam a interação humana, mas ainda não resolvem a necessidade de interconectar sistemas de bancos de dados. Para isso faz-se necessário o desenvolvimento de diversas alternativas de representação padronizada de dados. Um destes padrões é o XML e suas variantes, que são utilizados para melhorar a compatibilidade entre os sistemas díspares

de parceiros de negócio, define o significado dos dados contidos em documentos empresariais.

3.4. C#

C# é uma linguagem de programação da Plataforma .Net, distribuído juntamente com Microsoft Visual Studio.NET, como fortes características da linguagem, C# (pronuncia-se *C Sharp*) é simples, moderna orientada a objetos e fortemente tipada (type-safe). Segundo ARCHER (2001), C# é derivada de C e C++ combinando a alta produtividade do Visual Basic da Microsoft e o poder bruto do C++.

Esta linguagem de programação tem acesso a toda a plataforma do Next Generation Windows Services (NGWS), que incluem uma poderosa biblioteca de classes e um mecanismo de execução comum.

C# é a linguagem nativa para .NET Common Language Runtime (CLR), mecanismo de execução da plataforma .NET. Isso possibilita a convivência com várias outras linguagens especificadas pela Common Language Subset (CLS). Por exemplo, uma classe base pode ser escrita em C#, derivada em Visual Basic e novamente derivada em C#.

A base mínima relativamente alta da CLS (Common Language Specification) permite a criação de um clube de linguagens compatíveis com a CLS. Cada membro desse clube usufrui de benefícios duais: acesso completo à funcionalidade .NET Framework e à rica interoperabilidade com outras linguagens compatíveis. A CLS descreve um nível comum de funcionalidade da linguagem, por exemplo uma classe de Visual Basic pode herdar de uma C# e anular seus métodos virtuais. .Net Common Language Runtime é um ambiente baseado em componentes, e C# é desenhado para facilitar a criação de componentes. Os conceitos de componentes, como propriedades, métodos, eventos e atributos, são fortemente aplicados. Documentação pode ser escrita dentro dos componentes e exportadas para XML

C# não requer a bibliotecas de tipo (type libraries), arquivos de cabeçalho (header files), arquivos IDL. Os componentes são criados em C#, são auto descritivos e não necessitam de processo de registro.

Em C# tudo é objeto, ao contrário de linguagens como Java ou C++, tipos de dados e objetos interagem. C# fornece um “sistema unificado de tipos”, onde todos os tipos são tratados como objetos, sem perda de performance, ao contrário de linguagens como Lisp ou Smalltalk.

A necessidade da existência de softwares robustos e duráveis é básica a partir do momento em que adquire o conhecimento que muitas aplicações são difíceis de depurar e algumas vezes trazem resultados inesperados em tempo de execução. A C# disponibiliza o coletor de lixo (Garbage Collection) que fornece gerenciamento automático de memória, Exceções (Exceptions) que dispara erros que podem ser tratados, segurança no tipo de dados (Type-safety) que assegura a manipulação de variáveis e casts e Versão, são recursos encontrados na linguagem para construção desta categoria de software. C# possibilita utilização de ponteiros na execução de código não gerenciado.

C# permite interoperabilidade com SOAP, XML, componetes DLL, COM e qualquer outra linguagem da Plataforma .NET, mantendo integração total com projetos existentes. O aumento de produtividade e uma rápida curva de aprendizado, estes são garantidos pelo fato de que milhões de linhas de código em C# são encontrados no .NET.

Para ajudar nos conceitos de programação que incluem a linguagem C# segue uma exemplificação de código:

```
using System;
class Empregado
{
    public Empregados (string Nome, string Sobrenome,
                      int idade, double Salario)
    {
        this.Nome = Nome
        this.Sobrenome = Sobrenome;
        this.idade = idade;
        this.Salario = Salário
    }
}
```

```

    }
    protected string Nome;
    protected string Sobrenome;
    protected int Idade;
    protected double Salário;

    public double CalculaSalario (int horasTrabalhadas)
    {
        return (Salário * (double)horasTrabalhadas);
    }
}
class EmpregadoApp
{
    public static void Main ( ){
        Empregado emp = new Empregado("Jose", "Furtado", 59, 1000);
        Console.WriteLine ("\n Salário de Jose é $" +
emp.CalculaSalario(40));
    }
}

```

Considerando o código do programa, a cláusula *using* referencia a classes a serem utilizadas, *System* atua como namespace das classes. O namespace *System* contém muitas classes, uma delas é a *Console*. O método *WriteLine*, simplesmente emite a string no console.

O método *Main* é o ponto de entrada na execução de um programa, no C# não existem funções globais, a classe que será executada inicialmente possui embutida a função estática *Main*. Uma ou mais classes põem conter a função *Main*, portanto apenas uma será o ponto de entrada, indicada na compilação pelo parâmetro *main*: <tipo>, ou simplificando */m*: <tipo>.

3.5. Java e JSP

A Linguagem de programação Java tem uma arquitetura neutra, devido a isso aplicações em Java são ideais para ambientes como a Internet, ela é projetada para resolver problemas nas práticas modernas de programação, incorporando uma série de objetos prontos que facilita a programação.

A Linguagem de programação é originária de um projeto de pesquisa que visava a criação de um software avançado, um ambiente operacional pequeno, confiável, portátil, distribuído, que operasse em tempo real e que atendesse extensa variedade de hardwares de

redes e sistemas embutidos. Para melhorar o endereçamento de problemas que surgiram com as pesquisas do novo ambiente operacional que definiu a sintaxes em C e C++, as decisões de arquitetura e desenho da linguagem Java foram inspiradas em Eiffel, SmallTalk, Objective C.

Em documentação fornecida pela Sun a linguagem Java tem características marcadas por ser uma linguagem *simples*, podendo ser programada sem uma larga experiência anterior ou treinamento intenso diferenciada da linguagem C++ , por não utilizar ponteiros aritméticos, estruturas, uniões, classes básicas virtuais e sobrecarga de operadores. A *orientação a objeto* enfoca múltipla herança que em Java é tratada de forma melhor.

Outras características a linguagem é a *portabilidade e interpretação* que executa o código binário diretamente em alguma máquina para o qual ele tenha sido portado, a fase de linkagem de um programa simples é incremental e leve isso agiliza o processo de desenvolvimento.

As formas de utilização da linguagem Java são através de Applets (executados por um browser, são como programas que interpretam o código Java, transportado pela rede) e *Aplicação* (possui a função main()) e executa como um programa tradicional, podendo estar em uma máquina stand-alone ou em uma máquina servidora).

Java permite a conexão entre cliente e o servidor por meio da implementação de *sockets*, que são protocolos de comunicação de alto nível entre aplicações que são executadas em rede TCP/IP. É preciso levar em consideração a necessidade de definição das classes Sockets tanto no lado do cliente quanto no lado do servidor, para que sejam efetuadas conexões e trocas de mensagens.

A Plataforma Java tem sido uma escolha natural para o desenvolvimento de web services, pois estes dependem da capacidade das empresas utilizarem plataformas de comunicação diferentes para se comunicarem. As novas API's Java para XML aproximam XML da linguagem de programação Java.

Web services precisam ser escaláveis, seguros e eficientes, adicionando a portabilidade de dados e código. Para satisfazer essas necessidades foi projetada a plataforma J2EE, que facilita o desenvolvimento de web services em escala industrial, que é a programação da infra-estrutura que não pode deixar de abranger características como segurança, gerenciamento de transações distribuídas, e gerenciamento do campo de conexão. Como os componentes são reutilizáveis o tempo de desenvolvimento é reduzido.

Java pode utilizar no desenvolvimento de Web Services o JSP (Java Server Pages), uma tecnologia que ajuda na simplificação do processo de desenvolvimento de sites dinâmicos para a Web. Para adicionar conteúdo dinâmico às páginas HTML, o JSP suporta dois estilos diferentes, código Java e tags incorporadas nas páginas, por isso sua principal característica é a de ser híbrido.

A execução de uma página JSP começa com a solicitação da página ao servidor http, que encaminha o container servlet, então o servlet compilador de página é invocado para tratar da solicitação. Este recurso de compilação é um benefício adicional de desempenho, o JSP será mais lento que outras tecnologias apenas na primeira vez que ocorrer uma solicitação.

Segundo OLIVEIRA, 2003 um design de aplicações centradas em componentes é outra vantagem de JSP, sendo estes objetos escritos em Java e sua implementação está de acordo com um conjunto de convenções projetadas para promover modularidade e reusabilidade.

3.5.1. API's

Para escrever inteiramente aplicações Web na linguagem de programação Java é necessário o uso de APIs Java para XML, que podem ser baseadas na comunicação e no processamento de documentos XML.

As APIs baseadas na comunicação tem como responsabilidade a comunicação entre cliente e o servidor, JAXM fornece um meio padronizado de envio de documentos XML pela Internet usando a plataforma Java, JAX-RPC envia e recebe mensagens SOAP através da Internet e JAXR fornece um meio padronizado de acessar registros de negócio e informação compartilhada. As APIs baseadas no processamento de documentos XML, trabalham do lado do cliente e do servidor para a manipulação de documentos XML são JAXP que processa documentos XML usando vários recuperadores de elementos dos arquivos DTD e XML que geram um documento HTML que contém o resultado que deve ser apresentado no browser, estes recuperadores são chamados de parsers, JAXB que processa documentos XML usando schema derivado das classes de componentes JavaBeans e SAAJ que trata documentos XML para serem emitidos em mensagens SOAP através da Internet.

Capítulo 4: Comércio Eletrônico

A realização de processos de negócio através de aplicações das tecnologias de comunicação e de informação é parte do conceito de comércio eletrônico. Nos anos 70, com aplicações como transferir dinheiro eletronicamente, a chamada transferência eletrônica de fundos foi registrada como uma das primeiras aplicações de Comércio Eletrônico (CE). Os avanços vieram a partir da troca eletrônica de dados, tecnologias onde existe a possibilidade de transferência eletrônica de documentos de uma organização, isso engloba fatura, pagamentos eletrônicos entre empresas e ordens de compra, a esta tecnologia deu-se o nome de EDI.

Segundo TURBAN e KING (2004), as inovações das aplicações de CE foram chamadas de “sistemas interorganizacionais” (interorganizational systems –IOS) que envolviam desde negociações de estoques até o sistema de reservas de viagens. A expressão *eletronic commerce* passou a ser utilizada na década de 90 quando ocorreu um aumento da competitividade das pressões sobre os negócios e os usuários passaram a fazer parte da World Wide Web que teve uma rápida expansão a partir do desenvolvimento de novas redes, protocolos, softwares e especificações de padrão.

O MCT (2001) define que CE é parte do Estilo de Vida Web que se refere à compra e a venda, onde as empresas têm a oportunidade de ampliação de suas relações comerciais entre clientes e fornecedores e até mesmo clientes corporativos.

4.1. Conceitos e Classificações

ALBERTIN (2002), enfatizou que os processos do CE podem ser realizados de forma completa ou parcial, abrangendo transações negócio-a-negócio, negócio-a consumidor e intra-organizacional, numa infra-estrutura predominantemente pública de fácil e livre acesso

além de baixo custo. TURBAN e KING (2004) definiram que por Comércio eletrônico entende-se o processo de compra, troca de produtos, venda, serviços e informações pela Internet estes autores ainda citam as perspectivas definidas por outros autores KALAKOTA e WHINSTON (apud TURBAN E KING, 2004):

A perspectiva da comunicação: O CE é a distribuição de produtos, serviços, informação u pagamentos por meio de redes de computadores ou outros meios eletrônicos. **A perspectiva de processo comercial:** o CE é a aplicação de tecnologia para a automação de transações e do fluxo de trabalho. **A perspectiva de serviços:** o CE é uma ferramenta que satisfaz a necessidade de empresas, consumidores e administradores quanto à diminuição de custos e à elevação nos níveis de qualidade e agilidade de atendimento. **A perspectiva on-line:** o CE é a possibilidade de compra e venda de produtos e informações pela Internet e por outros serviços on-line. **A perspectiva da cooperação:** o CE é um instrumento de mediação inter e intracooperativa dentro de uma organização. **A perspectiva comunitária:** o CE é um ponto de encontro para os membros da comunidade poderem aprender, realizar negócios e cooperar uns com os outros.

O comércio eletrônico inclui qualquer negócio transacionado eletronicamente, em que essas transações ocorrem entre dois parceiros de negócio ou entre um negócio e seus clientes utilizando uma infra-estrutura digital.

De acordo com ALBERTIN (2002), as aplicações do CE reforçam os valores significativos para novas estratégias de gerenciamento de clientes pelo fato de conectarem diretamente compradores e vendedores, eliminarem os limites de tempo e lugar, apoiarem troca de informações totalmente digitadas entre as transações e a interatividade podendo adaptar-se dinamicamente ao comportamento do cliente. As aplicações podem ser atualizadas em tempo real, mantendo-se sempre atualizados.

Seguindo os conceitos de TURBAN e KING (2004) o CE é classificado através da origem da transação feita. A seguir destaca-se a classificação dos tipos de CE:

- **B2B – Busines-to-business** – Quando a transação do e-commerce é realizada *entre empresas* ou outros tipos de organizações.

- **B2C – Business-to-consumer** – Também pode ser chamado de varejo eletrônico, dá-se quando os participantes envolvidos na aplicação de CE efetuam transações de varejo entre *empresas e compradores* individuais.
- **B2B2C – Business-to-business-to-consumer** - Nesta classificação uma empresa oferece produtos ou serviços a uma empresa que seja sua cliente, a qual, por sua vez, mantém seus próprios clientes, para os quais o serviço ou produto é repassado.
- **C2B – Consumer-to-business** – Os participantes do C2B por um lado são indivíduos que utilizam a Internet para vender produtos ou serviços a organizações e, por outro, aqueles que procuram venderos a fim de que ofereçam lances, para obter produtos ou serviços de que necessitam.
- **C2C – Consumer-to-consumer** – Nesta categoria os consumidores vendem diretamente uns aos outros
- **MóBILE commerce** - Chamadas de comércio móvel ou m-commerce são atividades de CE realizadas em ambientes sem fio.
- **CE intranegócios** – Engloba atividades internas que envolvem a troca de produtos, serviços ou informações entre diferentes unidades e indivíduos de uma organização.
- **B2E – Business-to-employees** – Originária do CE intranegócios é a categoria na qual a organização oferece serviços informações ou produtos a funcionários.
- **CE nonbusiness** – Esta categoria é utilizada por instituições não-comerciais para reduzir despesas ou aprimorar suas operações e serviços.
- **G2C – Government-to-citizen** – Aplicação do **E-government** onde uma entidade governamental adquire produtos, serviços ou informação de empresas ou de cidadãos, ou, ainda, oferece esses bens a tais empresas e cidadãos.

- **E2E – Exchange-to-exchange** – Esta aplicação é um sistema formal que estabelece a conexão de trocas, ou seja, um mercado eletrônico público com diversos compradores e vendedores. Aumentando o número de participantes deste CE, ocorrem mais trocas.

Nas seções posteriores será avaliado o sistema proposto para o trabalho e sua classificação dentro da teoria de CE.

4.2. Visão Tecnológica

O futuro da Web é amplamente discutível e suas aplicações devem ser analisadas de modo a atender os padrões científicos, tecnológicos e comerciais.

A produtividade no desenvolvimento de aplicações Web é estimulada pela relação entre o resultado de um trabalho e pelos recursos gastos para realizá-lo. Assim para aumentar a produtividade no desenvolvimento de aplicações, busca-se a ferramentas e técnicas que aumentem os resultados, produzindo mais programas e códigos, ao mesmo tempo que diminuam o esforço e o tempo de produção. Este é o objetivo procurado ao se adotar uma ferramenta para desenvolvimento rápido. Os primeiros resultados podem ser facilmente obtidos ao se eliminar boa parte do trabalho repetitivo, automatizando algumas tarefas comuns para a maioria dos programas e, principalmente utilizando códigos prontos na forma de componentes, o trabalho de programação se reduz a um clicar e arrastar de componentes, obtendo-se razoável ganho no tempo de produção e, conseqüentemente, esperando o aumento da produtividade.

A automatização de atividades, por si só não é suficiente, durante o desenvolvimento de um projeto é necessário analisar dois outros aspectos complementares da busca pela produtividade que indicam uma tendência na evolução das ferramentas de apoio. É necessário analisar a importância da arquitetura onde o sistema está sendo desenvolvido, avaliando sua

influência nas possibilidades de automação e também como é possível obter ainda mais produtividade, integrando as diversas etapas do desenvolvimento de um serviço Web.

Quanto mais padronizada for a arquitetura da solução mais facilmente será sua automatização. Isso explica porque as ferramentas de desenvolvimento rápido tiveram um crescimento rápido no auge da arquitetura cliente/servidor.

O fortalecimento da linguagem JAVA e do modelo J2EE – Java 2 Enterprise Edition – como padrão arquitetônico das novas aplicações favorece novamente o desenvolvimento de ferramentas voltadas para este ambiente. Observa-se que quanto mais ricas em linguagem e a arquitetura criam-se novos desafios para os desenvolvedores de ferramentas de automação.

No futuro, espera-se obter o que se poderia chamar de uma aplicação auto-adaptativa. Onde a própria solução observa a evolução dos requisitos que foram definidos na fase de análise e como eles são resolvidos em projeto, fazendo as alterações refletirem no código final da solução. Uma aplicação deste tipo deve ser apoiada por sofisticadas ferramentas de software capazes de se adaptar, imediatamente, ao código distribuído às mudanças nos requisitos. Isso permitiria, por exemplo, que em uma aplicação completa em operação pudesse identificar a necessidade de caracterizar melhor uma classe do problema incluindo nela um novo atributo. Essa necessidade seria detectada pela aplicação e refletida, automaticamente, pelo ambiente de desenvolvimento integrado aos programas e códigos necessários gerando novos campos nas tabelas dos bancos de dados que armazenam aquela informação e criando novas interfaces gráficas para entrada, alteração e exibição da mesma. Este cenário, que ainda é ficcional, mostra que as potencialidades para ferramentas de automação na área de desenvolvimento de software são imensas, e que a revolução está apenas começando.

5. Escalabilidade

A partir do desenvolvimento estruturado e de uma aplicação segura, um desafio lançado é o acesso a escalabilidade de Sites de e-commerce na Web caracterizando a carga de trabalho destes Sites em um caminho representativo.

Uma proposta feita por BARBARA, DODGE JR, e MENASCE (2001) menciona que o único sistema de qualidade para e-commerce é o TPC-W desenvolvido pelo Conselho de Processamento de Transação (TPC).

No comércio eletrônico, consumidores interagem com o site através das sessões que são seqüências de requisições consecutivas para executar funções de e-business. A carga de trabalho de sites da Web que provem informação tem sido amplamente estudado e caracterizado de acordo com o nível de requerimento http. Um caminho pra capturar modelo padrão de navegação dentro de uma sessão é através do Gráfico Modelo de Comportamento do Consumidor (CBMG) que descreve o padrão de comportamento do usuário.

Vários modelos de comportamento de usuário podem ser detectados, por um mesmo site, por exemplo, uma fração de visitas pode vir de compradores ocasionais que gastam o tempo todo na tela de compras, mas muito raramente compram alguma coisa. Outras visitas ao site podem mostrar um padrão geralmente diferente carregado em recursos de TI que suportam o site.

O Conselho de Performance em Processamento de transação (TPC) provem métodos padrão da indústria para quantitativamente medir um sistema de performance no processamento de transação novamente um padrão comum bem como a performance de outros sistemas. O TPC considera uma transação, como ela é comumente entendida no mundo dos negócios: como uma troca comercial de mercadorias, serviços ou dinheiro. O TPC expandiu seu conjunto de sistema de qualidade em 2000 incluindo um sistema de qualidade transacional da web para sistemas de e-commerce, o Sistema de qualidade TPC-W, onde a

carga de trabalho é executado em um controlador da Internet no ambiente comercial que simula as atividades de um negócio orientado transacionado.

A primeira métrica de performance reportado pelo TPC-W é o numero de interações web processadas por segundo (WIPS), que simula diferentes perfis por variação de proporção de busca nas atividades de compra.

Os Componentes diferentes de um framework pra o TPC-W de site de e-commerce são integrados compactadamente. A escala é fundamentalmente definida pelo tamanho da população corrente de consumidores e o tamanho da loja, a escala requer manutenção a proporção entre a cardinalidade das tabelas do banco de dados a requisição de espaço armazenado para o banco de dados e o local do carregamento transacional no site.

A arquitetura do site TPC-W e do TPC-W gerador de carga de trabalho, de acordo com aplicação de busca que sua multi-threading para emular um grupo de cliente de buscas e consiste de um controlador de processos e um numero variável de linhas de busca. Cada linha de busca apresenta uma única carga de trabalho para o sistema de e-commerce segue as probabilidades de navegação do TPC-W.

O processo controlador no gerador de carga de trabalho expande o gerador TPC-W especificado capacita por permitir opcionalmente para variabilidade na carga de trabalho apresentada para o sistema de e-commerce por aumentar ou diminuir o numero de sessões ativas. Quando o controlador de processo precisa reduzir a carga de trabalho, ele sinala as linhas de busca que o numero de sessões ativas tem reduzido. No termino de cada requisição uma linha de busca checará para ver se a população do browser reduziu para o novo nível. Caso contrário, ele decrementa a população conta por um e estabelece sua própria sessão flag de parar para o verdadeiro. O próximo tempo na linha de busca completa uma página requisitada que sairá. Isto é feito pra reter a probabilidade de distribuição no CBMG.

6. Sistema de Cotações On-Line

O trabalho proposto apresenta um sistema de cotação de produtos e serviços que uma empresa disponibiliza a seus fornecedores através do Web Service.

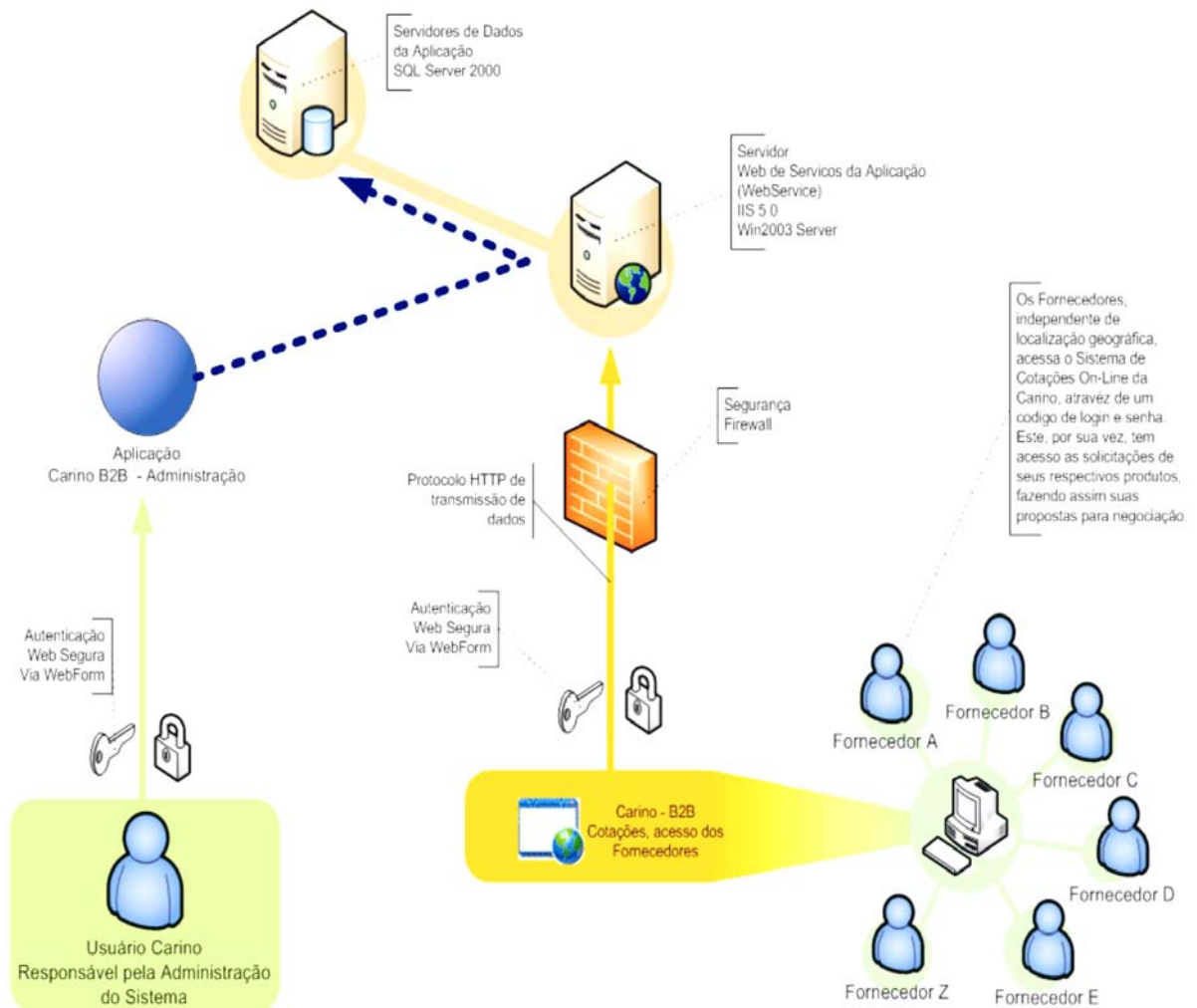


Figura 8. Arquitetura do sistema de Cotações On-Line

A arquitetura apresentada na Figura 8 separa a aplicação do Servidor de dados e do Servidor de Web Services. Ainda na arquitetura é importante destacar a segurança do sistema, que engloba Firewall e Aplicação Web Segura Via Web Form.

O sistema apresentado tem como base a aplicação na empresa Carino Ingredientes, fundada em 12 de abril de 1993, com sede em Marília, cujo segmento de atuação é o setor alimentício.

A base de dados importada para a aplicação do projeto é alimentada através do sistema ERP da Microsiga, o *Advanced Protheus 7.10* implantado na empresa desde 2000.

No sistema ERP, qualquer usuário cadastrado pode consultar o saldo em estoque dos produtos. A partir de uma saída lançada no sistema por requisição, automaticamente o estoque do produto é baixado, este saldo então é comparado com o saldo mínimo do produto cadastrado no sistema, podendo atingir ao saldo mínimo, o chamado ponto de pedido ou inferior a este.

A partir de qualquer necessidade de produtos e ou serviços os usuários do sistema podem inserir no sistema uma solicitação de compras, ou a partir da programação de itens a fabricar pode-se executar uma rotina que automaticamente inclui novas solicitações de compras no sistema de acordo com a demanda industrial. Cada produto ou serviço cadastrado no sistema está amarrado seus fornecedores, em uma tabela da base de dados. A nova solicitação de compras inclusa no sistema, uma rotina envia um e-mail ao fornecedor deste produto ou serviço avisando de que existe um novo item em cotação.

Os fornecedores acessam o serviço B2B On-Line, verificam o produto e quantidade solicitada e atualizam o seu preço na cotação aberta. Esta informação atualiza uma outra tabela, a de cotação.

O usuário com perfil de comprador analisa os preços e prazos de entrega informados na cotação on-line e escolhe um fornecedor para o produto, este usuário finaliza a solicitação de compra. O fornecedor escolhido recebe um e-mail de que sua cotação foi aceita.

No sistema ERP é feito o pedido de compras que eletronicamente é autorizado por gerencia e diretoria, e após esta etapa o pedido é enviado por e-mail ao fornecedor.

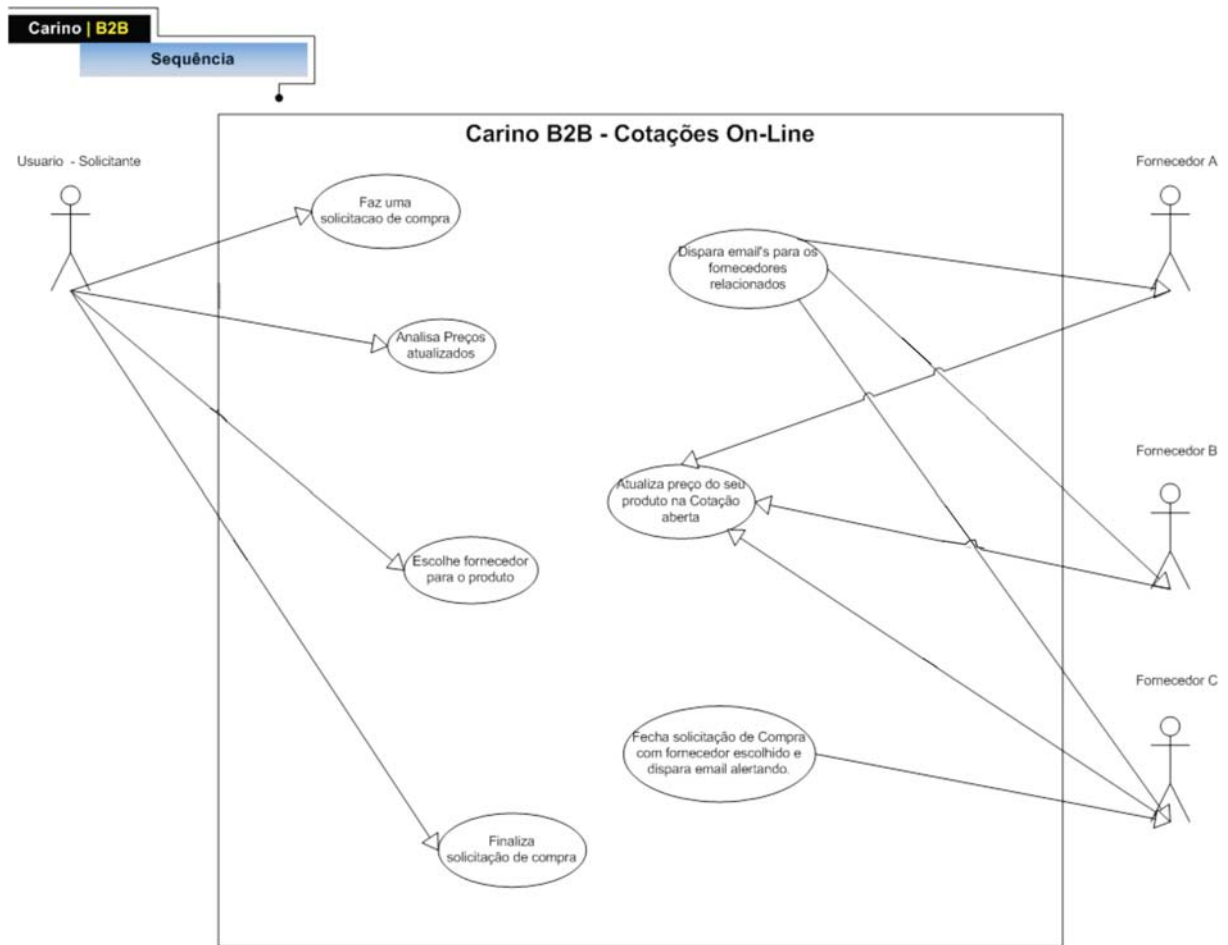


Figura 9. Modelagem do Sistema Carino B2B de Cotações On-Line

O Modelo físico e lógico são encontrados no Apêndice C e D.

6.1. Telas da aplicação

Uma forma de transferir para a Web o processo de gerenciamento de compras de matéria prima, material de consumo ou quais quer suprimentos é o chamado *e-procurement* que minimiza a carga de trabalho e viabiliza a redução de custos.

No caso do sistema desenvolvido, a maior vantagem é diminuir preços em uma cotação, viabilizando o acompanhamento do desempenho dos fornecedores já inseridos no processo, e criando uma integração na cadeia de relacionamento e parcerias da empresa.

Uma primeira forma ainda simples de criar a integração mencionada e futuramente transformá-la em *e-procurement* foi criar um pequeno sistema de cotações conforme descrito abaixo.

6.1.1. Listagens de Solicitações

De acordo com a arquitetura montada existem telas no sistema desenvolvido onde cada uma delas tem a especificação do usuário que tem permissão para operá-la.

The screenshot shows the 'Cotações on-line' web application. At the top left is the 'carino' logo. The main title is 'Cotações on-line B2B'. On the top right, there are links for 'Alterar Senha' and 'Sair', and user information: 'Usuário: Luciana Bonadio', 'Grupo: Administradores', and the date 'Sexta-feira, 04 de novembro de 2005'. A navigation bar contains links for 'Solicitações', 'Cotações', 'Fornecedores', 'Produtos', 'Usuários', 'Grupo de Usuário', and 'Transações'. The main content area is titled 'Solicitações > Listagem' and shows a table with 120 records. The table has columns for 'Select', 'Código', 'Data Prevista', 'Situação', and 'Observação'. Each row in the 'Select' column contains a dropdown menu icon. The bottom right corner indicates 'Páginas: 1,2,3,4,5..'

Select	Código	Data Prevista	Situação	Observação
⌵				
⌵				
⌵				
⌵				
⌵				
⌵				
⌵				
⌵				
⌵				
⌵				
⌵				
⌵				

Figura 10. Tela Listagem de Solicitações

Na Figura 10 observa-se a tela de Listagem de Solicitações onde é possível visualizar o status da cotação (aberta, finalizada ou cancelada). O Administrador do sistema também poderá selecionar a cotação desejada para melhor visualização.

6.1.2. Detalhamento e listagem das Solicitações

Parte do relacionamento com o fornecedor é colocar os produtos e a quantidade negociada adicionando novos produtos, na observação o usuário pode fazer notas quanto a modelo e marca do produto solicitado.

carino
Ingredientes

Cotações on-line
B2B

Alterar Senha | Sair
Usuário: Luciana Bonadio
Grupo: Administradores
Sexta-feira, 04 de novembro de 2005

Solicitações | Cotações | Fornecedores | Produtos | Usuários | Grupo de Usuário | Transações

Solicitações > Listagem > Detalhamento

Solicitação: 123456 Obs.: Solicitacao de teste
Data Prevista: 11/11/2005
Situação: Em andamento

Itens

Produto: -- Selecione -- Quantidade: Adicionar

Remover	Produto	Quantidade
..		
..		
..		
..		
..		
..		
..		
..		

Salvar Voltar

by Luciana Bonadio Furtado

Figura 11. Tela de Detalhamento das Solicitações

Na tela de detalhamento das solicitações é possível visualizar a data prevista, sendo esta data, a necessidade de entrega do produto na empresa, os itens contidos na solicitação de compras e suas quantidades além as observações para detalhamento maior quanto a impostos, condições de frete, especificações do produto, etc.

6.1.3. Listagens de Cotações e visualização de propostas

Na Figura 12 apresenta-se a tela de Listagem de Cotações, é possível visualizar a solicitação e status destas, além da possibilidade de um link para a visualização de propostas.

Ao solicitar a visualização das Propostas (Figura 12), o usuário do sistema entra na tela onde pode ver todas as propostas dos fornecedores em relação às cotações enviadas a estes (Figura 13).

Nesta tela o usuário poderá definir pela melhor proposta e fazer a aprovação da cotação.

carino
ingredientes

Cotações on-line
B2B

[Alterar Senha](#) | [Sair](#)
Usuário: **Luciana Bonadio**
Grupo: Administradores
Sexta-feira, 04 de novembro de 2005

Solicitações | Cotações | Fornecedores | Produtos | Usuários | Grupo de Usuário | Transações

Cotações > Listagem

Total: 120 registros

Ver Propostas	Solicitação	Situação
..		
..		
..		
..		
..		
..		
..		
..		
..		
..		

Páginas: 1,2,3,4,5.

Figura 12. Tela de Listagem de Cotações

Depois de definida a melhor proposta de cotação, o sistema envia uma notificação ao fornecedor de que sua proposta foi aceita e que o pedido de compras será enviado a este por e-mail, após aprovação eletrônica dos membros da gerência e diretoria.

carino
ingredientes

Cotações on-line
B2B

[Alterar Senha](#) | [Sair](#)
Usuário: **Luciana Bonadio**
Grupo: Administradores
Sexta-feira, 04 de novembro de 2005

Solicitações | Cotações | Fornecedores | Produtos | Usuários | Grupo de Usuário | Transações

Cotações > Listagem > Propostas

Solicitação: 123456

Propostas

Aprovar

Fornecedor: ParmaSol Ltda.

Obs.:

Produto	Valor Unitário	Valor Total

Aprovar

Fornecedor: Marília Alimentos S.A.

Obs.:

by: Luciana Bonadio Furtado

Figura 13. Tela de Listagem de Propostas

Na tela de visualização de propostas, estão todas as propostas enviadas pelos fornecedores, isso facilita o trabalho do comprador na análise das propostas e na aprovação.

6.1.4. Fornecedores

Na Figura 14 é apresentada a tela de Listagem de Fornecedores, visualizando a primeira instância o código do fornecedor, o nome, a situação (ativo, ou não ativo) e nome fantasia. É possível visualizar detalhes do cadastro do fornecedor nas próximas telas.

carino Ingredientes **Cotações on-line** B2B

[Alterar Senha](#) | [Sair](#)

Usuário: **Luciana Bonadio**
Grupo: Administradores
Sexta-feira, 04 de novembro de 2005

Solicitações | Cotações | Fornecedores | Produtos | Usuários | Grupo de Usuário | Transações

Fornecedores > Listagem

Total: 120 registros

Select	Código	Nome	Situacao	Nome Fantasia
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				

Páginas: 1,2,3,4,5..

Figura 14. Tela dos Fornecedores

Ao selecionar qualquer fornecedor exibido na Listagem, é possível ver todo o detalhamento de cadastro do fornecedor. As seções seguintes mostram o ambiente de operação do fornecedor.

carino Ingredientes **Cotações on-line** B2B

[Alterar Senha](#) | [Sair](#)

Usuário: **Luciana Bonadio**
Grupo: Administradores
Sexta-feira, 04 de novembro de 2005

Solicitações | Cotações | Fornecedores | Produtos | Usuários | Grupo de Usuário | Transações

Fornecedores > Listagem > Detalhamento

Código:

Nome:

Nome Fantasia:

Situação:

Endereço Rua/Av.: Número:

Bairro: Município:

Estado:

CPF/CNPJ:

Telefone:

Fax:

Email:

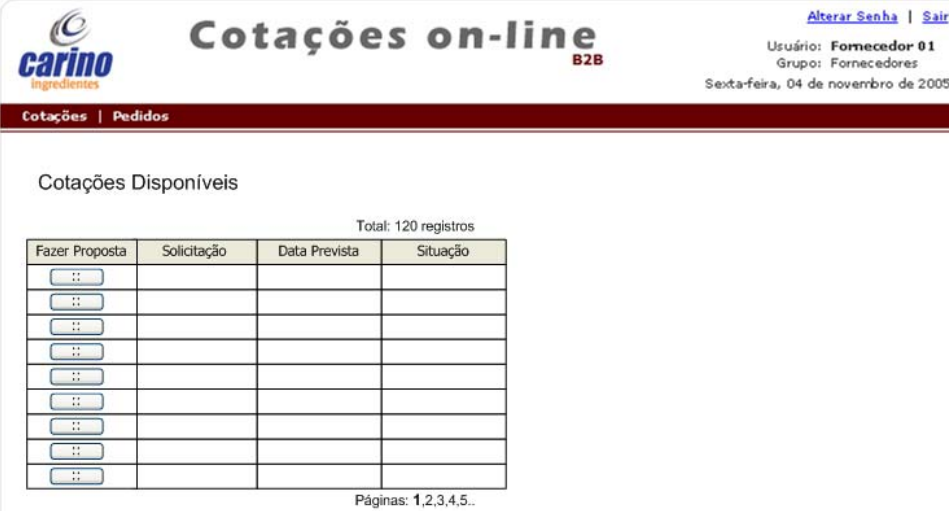
Pessoa Contato:

by Luciana Bonadio Furtado

Figura 15. Tela dos Fornecedores

6.1.5. Cotações disponíveis

O Ambiente de operação dos fornecedores apresenta as cotações disponíveis, nas quais a empresa fornecedora deverá entrar em negociação oferecendo o valor de seu produto ou serviço.



Cotações Disponíveis

Total: 120 registros

Fazer Proposta	Solicitação	Data Prevista	Situação
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			
⋮			

Páginas: 1,2,3,4,5..

Figura 16. Cotações Disponíveis

A tela inicial (Figura 16) especifica a solicitação de compras, a data prevista de entrega de acordo com a necessidade da empresa solicitante, e o status da solicitação (aberta, finalizada ou cancelada).

Clicando na opção para fazer proposta a empresa fornecedora entrará na tela efetiva com o detalhamento da cotação (Figura 17) para que esta possa incluir os dados sobre seu produto ou serviço.

Neste ponto, como é possível notar na figura, o fornecedor é avisado que o cumprimento de da negociação como prazo e quantidade de entrega, serão avaliados pelo sistema de qualidade da empresa solicitante, o não cumprimento de qualquer item acordado acarretará em nota que influência na qualificação, pré-qualificação ou não qualificação da empresa fornecedora.

É importante ressaltar que para manter o conceito de B2B e e-procurement a cotação é exclusiva a empresa fornecedora, isto é uma empresa fornecedora não visualiza os dados (valor, unitário, valor total, efetivação de data de entrega) postados por outra empresa.

Cotações on-line B2B

Alterar Senha | Sair

Usuário: Fornecedor 01
Grupo: Fornecedores
Sexta-feira, 04 de novembro de 2005

Cotações | Pedidos

Cotações Disponíveis > Fazer Proposta

Atenção Fornecedor: A responsabilidade no cumprimento do prazo de entrega será avaliada pela Carino

Solicitação: 123456

Data Prevista: 11/11/2005

Renegociação para data de entrega ou preço:

Produto	Codigo Carino	Quantidade	Valor Unitário	Valor Total

Confirmar Voltar

by Luciana Bonadio Furtado

Figura 17. Propostas de Cotações

6.2. Fluxograma do trabalho Proposto

O Sistema Carino B2B percorre um fluxo de informações para que então atinja o objetivo desejado, ou seja, a minimização de custos durante o processo de cotação de uma empresa solicitante e empresas fornecedoras.

O percurso do fluxograma do sistema segue a descrição do sistema desde o lançamento de uma solicitação de compras no sistema ERP, passando através do web service ao contato com fornecedores, a aprovação da melhor proposta enviada e chegando a emissão do pedido novamente no sistema ERP.

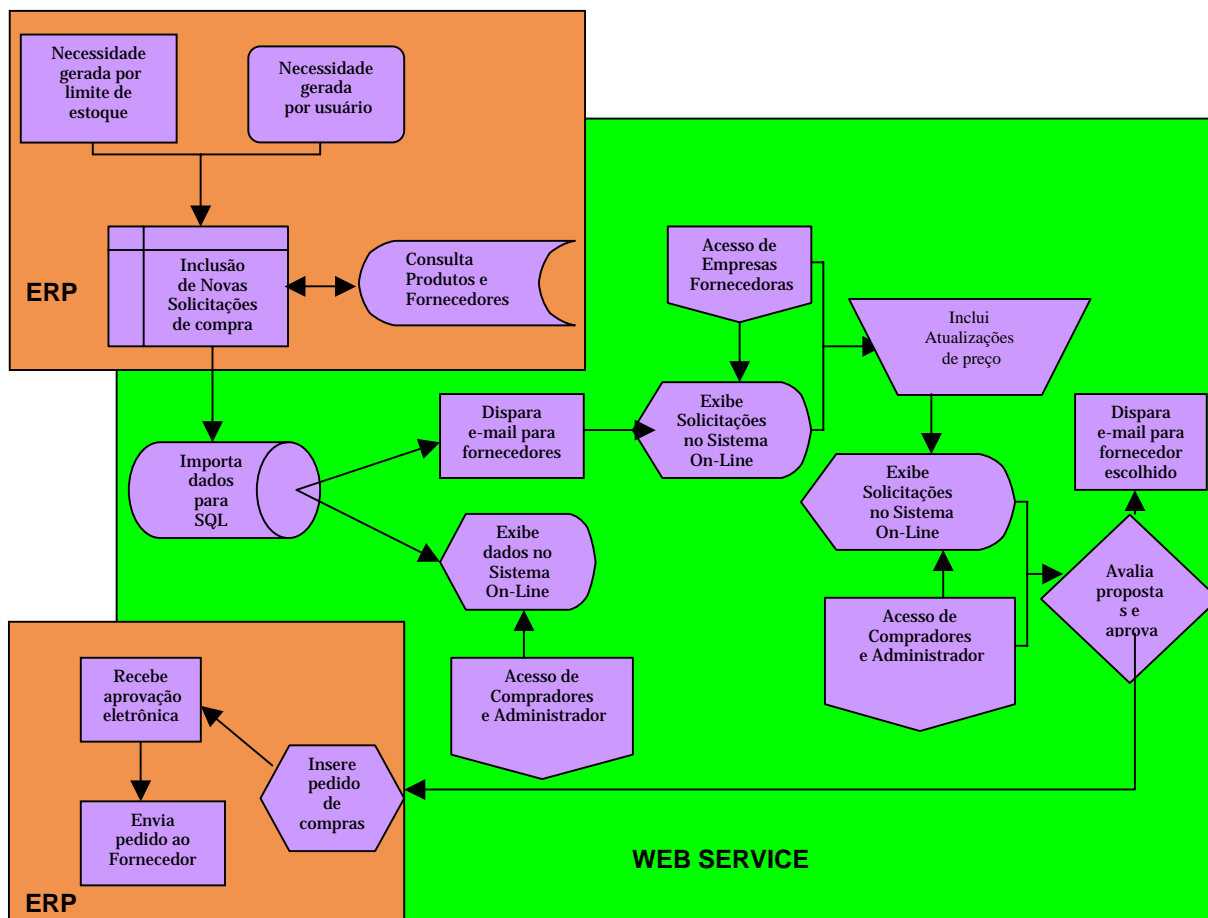


Figura 18. Fluxograma do Sistema desenvolvido

O Modelo Relacional do sistema foi modelado no Erwin e é mostrado na Figura 8.

Será apresentada no Apêndice C, já os códigos fontes estão disponíveis em CD.

7. Conclusões

O trabalho apresentado implementou Web Services de um sistema de cotações on-line entre empresas, para isso foram necessárias pesquisas, científicas, teóricas e práticas.

Inicialmente o projeto contemplaria um sistema de rastreamento de insumos utilizados em processos produtivos, mas o escopo foi alterado devido à necessidade de preservação de certos segredos industriais.

Apesar da alteração de escopo do projeto, todas as pesquisas e ferramentas de uso foram preservadas e grandes esforços foram ponto de concepção das tecnologias abordadas, como padrões, linguagens e ferramentas.

Uma das muitas vantagens na aplicação deste Web Service é o benefício de que tanto o administrador, o usuário como aprovador de uma cotação ou os fornecedores podem acessar o serviço em qualquer lugar conectado a Internet. É a comodidade de não atrasar uma negociação urgente por estar longe da empresa.

Web Services são estruturados a partir de dois ou mais componentes e existem desafios para um gerenciamento correto dos componentes, como a garantia de realização de diferentes tarefas por estes componentes, e também novos mecanismos para o controle de execução de transações e tolerância a falhas, por isso muitas das aplicações da nova Web utilizarão taxonomia para definir conceitos utilizados.

Os novos padrões tendem a ser mais utilizados, introduzindo dados do dia a dia das empresas a aplicações Web Services fazendo da Internet, cada vez mais informacional, mas é de extrema importância respeitar e preservar regras de negócios de cada empresa sem deixar de aumentar os resultados.

Através dos estudos pode-se adquirir inúmeras idéias para futuras aplicações, mas deseja-se obter conhecimentos mais profundos em JAVA para então desenvolver as aplicações futuras. Não apenas no âmbito industrial, como o projeto apresentado, mas

explorar mais áreas focadas em agro business por ser um campo que promete tantas melhorias principalmente econômicas.

Pelo fato de uma aplicação de software representar uma reorganização de trabalhos tradicionais por meio de rotinas adquiridas pelo conhecimento e por análise lógica e sistemática, o desenvolvimento de arquiteturas Web deve ser considerado em todas as áreas de serviços e negócios.

Referências Bibliográficas

- (ALBERTIN, 2002) ALBERTIN, Alberto Luiz. Comércio eletrônico: modelo aspectos e contribuições de suas aplicações. Colaboração Rosa Maria de Moura . 4.ed. São Paulo: Atlas, 2002
- (ANDRADE, 1999) ANDRADE, R.C. S. de; CARVALHO, R. D. de. Utilização de metadados como ferramenta de padronização da informações nos documentos eletrônicos. Rio de Janeiro, 1999 – Monografia – Instituto de Tecnologia da Informação e da Comunicação, Universidade de Santa Úrsula.
- (ARCHER, 2001) ARCHER, Tom. Descobrimo C#. Tradução DocWere Traduções Técnicas, Edson Furmankiewicz e Sandra Figueiredo. Rio de Janeiro: Campus, 2001.
- (ASP.NET, 2005) ASP.NET MICROSOFT. The Asp.net Web Matrix Project (Reloaded!). Disponível em: <http://www.asp.net/webmatrix/default.aspx?tabindex=4&tabid=46>, acesso em 30 de março de 2005.
- (AYALA, 2002) AYALA, Dietrich et al. Professional Open Source Web Services. United Kingdown: Wrox Press Ltda, 2002.
- (BELLWOOD, 2002) BELLWOOD, Tom et al. UDDI Version 3.0. 2002. Disponível em: <http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>. Acesso em 15 de junho de 2005.
- (BERNARDINO, 1998) BERNARDINO, F.P. Gerência de Metadados: como integrar seu Data Warehouse. Rio de Janeiro, 1998. Monografia apresentada a disciplina de tópicos especiais em Banco de Dados, Mestrado em Informática. UFRJ.
- (BERNERS, HENDLER e LASSILA, 2001) BERNERS-LEE, HENDLER, J. e LASSILA, O. 2001. The Semantic Web, Disponível em: http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21. Acesso em: 24 de abril de 2005.
- (MTC, 2001) BRASIL. Ministério da Ciência e Tecnologia Secretaria de Política de Informática. Internet Comercial: conceitos, estatísticas, aspectos legais. Brasília, 2001
- (CERAMI) CERAMI, Ethan. Distributed Applications with XML-RPC, SOAP, UDDI & WSDL. O'reilly Online Catalog Disponível em: <http://www.oreilly.com/catalog/webservess/chapter/ch06.html>. Acessado em 8 de outubro de 2005.
- CZAJKOWSKI, Grzegorz; DAYNÈS, Laurent; TITZER, Bem. A Multi-User Virtual Machine. S.D. Disponível em: <http://research.sun.com/projects/barcelona/papers/usenix03.pdf>. Acesso em 02 de novembro de 2005.

(DEMPSEY, 1998) DEMPSEY, Lorcan; HERRY, Rachel. Metadata: a current view of practice and issues. 1998. Disponível em <http://www.ukoln.ac.uk/metadata/publications/jdmetadata/> Acesso em 7 de maio de 2005.

(DRUCKER, 2004) DRUCKER, Peter. Além da evolução da informação, Revista HSM Management, n18, ano 3.

(FROEHLICH, 1998) FROEHLICH, Thomas J. C. Web Surfer. Responsabilidade social e recursos da Internet. Transinformação, Campinas, v.10, n.2, maio/agosto 1998.

(GARDNER, 1997) GARDNER, Stephen R. The Quest to Standardize Metadadata. 1997. Byte Digest. Disponível em: <http://www.byte.com/art/9711/sec4/art3.htm> Acesso em 18 de junho de 2005.

(GOMES, 2000) GOMES, Helena. E. Uma profissão de futuro. 2000. Disponível em: <http://fgv.br/dg/diti/bib/geral/htm/hpbb12.tm> Acesso em 25 de março de 2005.

(KALAKOTA, 2002) KALAKOTA, Ravi; ROBINSON, Márcia. E-business: estratégias para alcançar o sucesso no mundo digital. Tradução Carlos Alberto Picanço de Carvalho. 2 ed. Porto Alegre: Bookman, 2002

(LOTAR, 2001) LOTAR, Alfredo. XML para Programadores ASP. Rio de Janeiro: Axcel Books do Brasil Editora, 2001

(OASIS & UDDI, 2005) OASIS & UDDI. Arquived Articles. Disponível em <http://www.uddi.org/new2.htm>. Acesso em 14 de junho de 2005.

(RALHA, 2004) RALHA, Cláudio Pereira dos Santos. Segredos do Visula Studio .Net. 1 ed. São Paulo: Digerati Books, 2004

(SCANNELL, 2000) SCANNELL, Ed; SULLIVAN, Tom. XML standard readied for businesses. 2000. Disponível em: <http://networkworld.com/news/2000/0901xmlspcec.htm> Acesso em 14 de junho de 2005.

(SOUZA, VENDURSCULO e MELO, 2000) SOUZA, Márcia I. F.; VENDRUSCULO, Laurimar G.; MELO, Geane C. Metadados para descrição de recursos de informação eletrônica: utilização do padrão Dublin Core. 2000. Disponível em <http://www.ibict.br/cienciadainformacao/include/getdoc.php?id=613&article=309&mode=pdf> Acesso em: 5 de junho de 2005.

(TENNAT, 1998) TENNAT, Roy. The art and Science of Digital Bibliography. Library Journal, 1998 Disponível em <http://www.libraryjournal.com/article/CA156498.html>. Acesso em: 4 de junho de 2005.

(TURBAN e KING, 2004) TURBAN, Efraim; KING, David. Comércio Eletrônico: estratégia e gestão. Tradução Arlete Simille Marques. Revisão técnica Belmiro João, Eurico Veras Marques. São Paulo: Prentice Hall, 2004. Título original: Introduction to e-commerce.

(URI, 2001) URI Planing Interest Group, W3C/IETF. URIs, URLs, and URNs: Clarifications and Recommendations 1.0. 2001. Disponível em: <http://www.w3.org/TR/2001/NOTE-uri-clarification-20010921/>. Acesso em 30 de outubro de 2005.

(UDDI-1, 2005) UDDI MICROSOFT (?), disponível em: <http://uddi.microsoft.com> , acesso em: 18/10/2005 (?) idem abaixo .

(UDDI-2, 2005) e <http://uddi.ibm.com>

(ASP.NET, 2005) ASP.NET MICROSOFT. The Asp.net Web Matrix Project (Reloaded!). Disponível em: <http://www.asp.net/webmatrix/default.aspx?tabindex=4&tabid=46>, acesso em 30 de março de 2005.

(WANDER, 2003) WANDER, Daniel. Desenvolvendo Aplicações ASP.Net com Web Matrix. São Paulo: Novatec Editora Ltda, 2003.

(WIEBEL, 2005) WEIBEL, Stuart. The Dublin Core: A Simple content Description Model for Electronic Resources. Disponível em: <http://www.asis.org/Bulletin/Oct-97/weibel.htm>. Acesso em 8 de maio de 2005.

APENDICE

A. URI pelo registro Oficial IANA

Uniform Resource Identifier (URI) SCHEMES

(last updated 03 November 2005)

This is the Official IANA Registry of URI Schemes

In the Uniform Resource Identifier (URI) definition [RFC3986,RFC1738] there is a field, called "scheme", to identify the type of resource and access method.

Scheme Name	Description	Reference
ftp	File Transfer Protocol	[RFC1738]
http	Hypertext Transfer Protocol	[RFC2616]
gopher	The Gopher Protocol	[RFC-
hoffman-gopher-uri-03.txt]		
mailto	Electronic mail address	[RFC2368]
news	USENET news	[RFC1738]
nntp	USENET news using NNTP access	[RFC1738]
telnet	Reference to interactive sessions	[RFC4248]
wais	Wide Area Information Servers	[RFC4156]
file	Host-specific file names	[RFC1738]
prospero	Prospero Directory Service	[RFC4157]
z39.50s	Z39.50 Session	[RFC2056]
z39.50r	Z39.50 Retrieval	[RFC2056]
cid	content identifier	[RFC2392]
mid	message identifier	[RFC2392]
vemmi	versatile multimedia interface	[RFC2122]
service	service location	[RFC2609]
imap	internet message access protocol	[RFC2192]
nfs	network file system protocol	[RFC2224]
acap	application configuration access protocol	[RFC2244]
rtsp	real time streaming protocol	[RFC2326]
tip	Transaction Internet Protocol	[RFC2371]
pop	Post Office Protocol v3	[RFC2384]
data	data	[RFC2397]
dav	dav	[RFC2518]
opaquelocktoken	opaquelocktoken	[RFC2518]
sip	session initiation protocol	[RFC3261]
sips	secure session intitiaion protocol	[RFC3261]

tel	telephone	[RFC2806]
fax	fax	[RFC2806]
modem	modem	[RFC2806]
ldap	Lightweight Directory Access Protocol	[RFC-ietf-
ldapbis-url-09.txt]		
https	Hypertext Transfer Protocol Secure	[RFC2818]
soap.beep	soap.beep	[RFC3288]
soap.beeps	soap.beeps	[RFC3288]
xmlrpc.beep	xmlrpc.beep	[RFC3529]
xmlrpc.beeps	xmlrpc.beeps	[RFC3529]
urn	Uniform Resource Names	[RFC2141]
	(please see: http://www.iana.org/assignments/urn-	
namespaces)		
go	go	[RFC3368]
h323	H.323	[RFC3508]
ipp	Internet Printing Protocol	[RFC3510]
tftp	Trivial File Transfer Protocol	[RFC3617]
mupdate	Mailbox Update (MUPDATE) Protocol	[RFC3656]
pres	Presence	[RFC3859]
im	Instant Messaging	[RFC3860]
mtqp	Message Tracking Query Protocol	[RFC3887]
iris.beep	iris.beep	[RFC3983]
dict	dictionary service protocol	[RFC2229]
snmp	Simple Network Management Protocol	[RFC4088]
crid	TV-Anytime Content Reference Identifier	[RFC4078]
tag	tag	[RFC4151]
dns	Domain Name System	[RFC-draft-
josefsson-dns-url-13.txt]		

Reserved URI Scheme Names:

afs	Andrew File System global file names
tn3270	Interactive 3270 emulation sessions
mailserver	Access to data available from mail servers

REFERENCES

- [RFC1738] Berners-Lee, T., L. Masinter, and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, CERN, Xerox Corporation, University of Minnesota, December 1994.
- [RFC2056] Denenberg, R., J. Kunze, D. Lynch, "Uniform Resource Locators for Z39.50", RFC 2056, Library of Congress, University of California, San Francisco, SilverPlatter Information Ltd., November 1996.
- [RFC2122] Mavrakis, D., H. Layec, and K. Kartmann, "VEMMI URL Specification", RFC 2122, Monaco Telematique, ETSI, Telecommunication+Multimedia, March 1997.
- [RFC2141] R. Moats, "URN Syntax", RFC 2141, May 1997.
- [RFC2165] Veizades, J., E. Guttman, C. Perkins, and A. Kaplan, "Service Location Protocol", RFC 2165, @Home Network, Sun Microsystems, June 1997.

- [RFC2192] Newman, C., "IMAP URL Scheme", RFC 2192, Innosoft, September 1997.
- [RFC2224] Callaghan, B., "NFS URL Scheme", RFC 2224, Sun Microsystems, October 1997.
- [RFC2229] R. Faith and B. Martin, "A Dictionary Server Protocol", RFC 2229, October 1997.
- [RFC2244] Newman, C., and J. Myers, "ACAP -- Application Configuration Access Protocol", RFC 2244, Innosoft, Netscape, November 1997.
- [RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, Columbia U., Netscape, RealNetworks, April 1998.
- [RFC2368] Hoffman, P., Masinter, L., and J. Zawinski, "The mailto URL scheme", RFC 2368, July 1998.
- [RFC2371] Lyon, J., Evans, K., and J. Klein, "Transaction Internet Protocol Version 3.0", RFC 2371, July 1998.
- [RFC2384] Gellens, R., "POP URL Scheme", RFC 2384, Qualcomm Inc., August 1998.
- [RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, Xison Inc, August 1998.
- [RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.
- [RFC2397] Masinter, L., "The "data" URL scheme", RFC 2397, August 1998.
- [RFC2518] Goland, Y., Whitehead, E., Faizi, A., Carter, S. and D. Jensen, "HTTP Extensions for Distributed Authoring -- WEBDAV", RFC 2518, February 1999.
- [RFC2609] Guttman, E., Perkins, C., Kempf J., "Service Templates and Service: Schemes", RFC 2609, June 1999.
- [RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC2806] Vaha-Sipila, A., "URLs for Telephone Calls", RFC 2806, April 2000.
- [RFC2818] E. Rescorla, "HTTP Over TLS", RFC 2818, May 2000.
- [RFC3261] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC3288] E. O'Tuathail and M. Rose, "Using SOAP in BEEP", RFC 3288, June 2002.
- [RFC3305] URI Planning Interest Group/W3C/IETF URIs, URLs, and URNs: Clarifications and Recommendations Report from

- the joint W3C/IETF URI Planning Interest Group, RFC 3305, August 2002.
- [RFC3368] M. Mealling, "The 'go' URI Scheme for the Common Name Resolution Protocol", RFC 3368, August 2002.
 - [RFC3508] O. Lvein, "H.323 URL Scheme Registration with IANA", RFC 3508, April 2003.
 - [RFC3510] R. Herriot and I. McDonald, "Internet Printing Protocol/1.1: IPP URL Scheme", RFC 3510, April 2003.
 - [RFC3529] W. Harold, "Using XML-RPC in BEEP", RFC 3529, April 2003.
 - [RFC3617] E. Lear, "URI Scheme for the TFTP Protocol", RFC 3617, October 2003.
 - [RFC3656] R. Siemborski, "The MUPDATE Distributed Mailbox Database Protocol", RFC3656, December 2003.
 - [RFC3859] J. Peterson, "Common Profile for Presence (CPP)", RFC 3859, August 2004.
 - [RFC3860] J. Peterson, "Common Profile for Instant Messaging (CPIM)", RFC 3860, August 2004.
 - [RFC3887] T. Hansen, "Message Tracking Query Protocol", RFC 3887, September 2004.
 - [RFC3983] A. Newton and M. Sanz, "IRIS - Using the Internet Registry Information Service (IRIS) over the Blocks Extensible Exchange Protocol (BEEP)", RFC 3983, January 2005.
 - [RFC3986] T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005.
 - [RFC4078] N. Earnshaw, S. Aoki, A. Ashley, and W. Kameyama, "The TV-Anytime Content Reference Identifier (CRID)", RFC 4078, May 2005.
 - [RFC4088] D. Black, K. McCloghrie, and J. Schoenwaelder, "Uniform Resource Identifier (URI) Scheme for the Simple Network Management Protocol (SNMP)", RFC 4088, June 2005.
 - [RFC-ietf-ldapbis-uri-09.txt] M. Smith, Ed. and T. Howes, "LDAP: Uniform Resource Locator", RFC XXXX, Month Year.
 - [RFC4151] T. Kindberg and S. Hawke, "The 'tag' URI scheme", RFC 4151, October 2005.
 - [RFC4156] P. Hoffman, "The wais URI Scheme", RFC 4156, August 2005.
 - [RFC4157] P. Hoffman, "The prospero URI Scheme", RFC 4157, August 2005.
 - [RFC4248] P. Hoffman, "The telnet URI Scheme", RFC 4248, October 2005.
 - [RFC-hoffman-gopher-uri-03.txt] P. Hoffman, "The gopher URI Scheme", RFC XXXX, Month Year.

[RFC-draft-josefsson-dns-url-13.txt]

S. Josefsson, "Domain Name System Uniform Resource Identifiers",
RFC XXXX, Month Year.

B. Exemplo de Documento WSDL

```

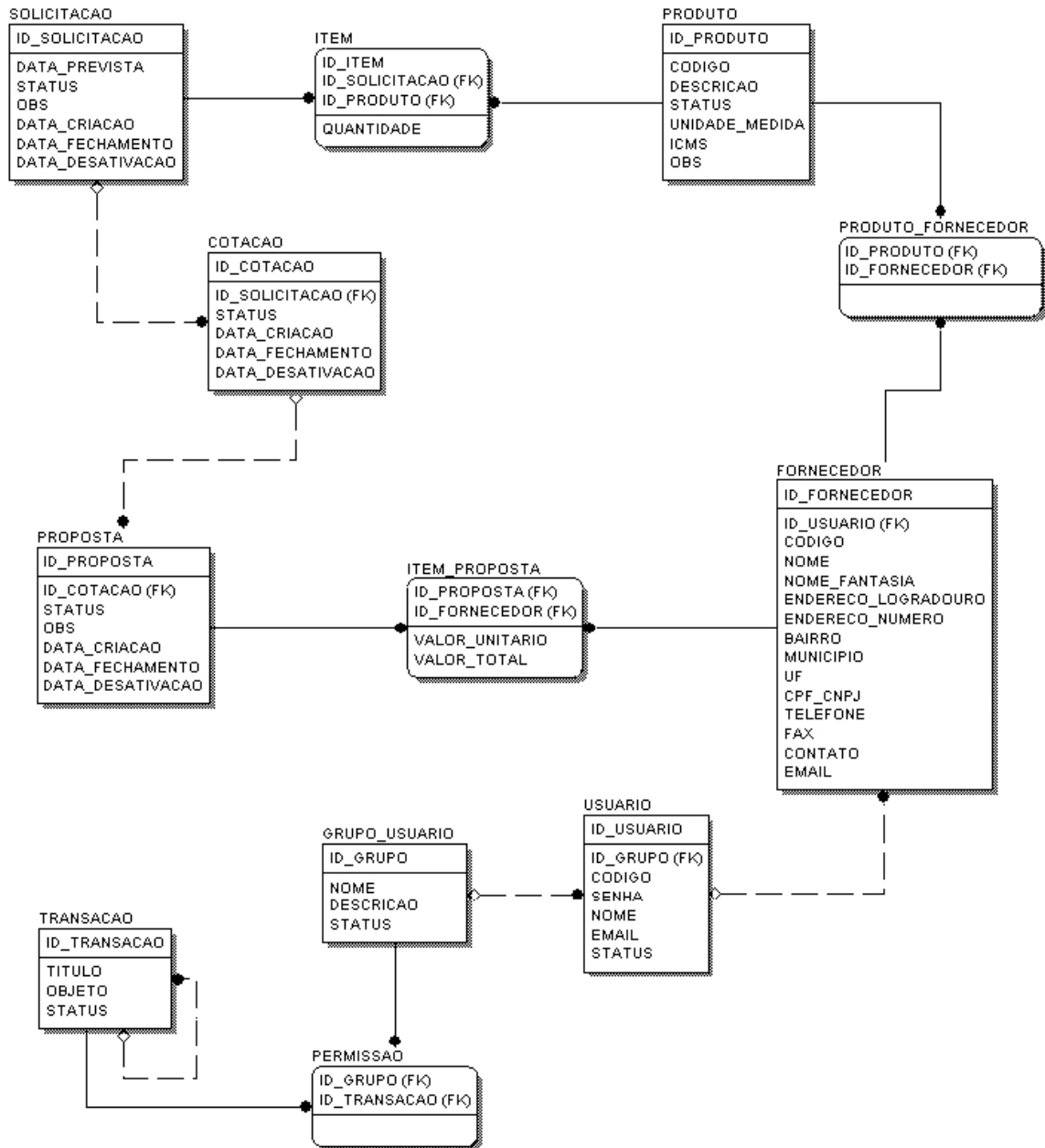
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/Calculator.jws"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://localhost:8080/axis/Calculator.jws"
xmlns:intf="http://localhost:8080/axis/Calculator.jws"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!-- WSDL created by Apache Axis version: 1.2RC2
Built on Nov 16, 2004 (12:19:44 EST)-->
<wsdl:message name="subtractRequest">
<wsdl:part name="i1" type="xsd:int"/>
<wsdl:part name="i2" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="subtractResponse">
<wsdl:part name="subtractReturn" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="addResponse">
<wsdl:part name="addReturn" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="addRequest">
<wsdl:part name="i1" type="xsd:int"/>
<wsdl:part name="i2" type="xsd:int"/>
</wsdl:message>
<wsdl:portType name="Calculator">
<wsdl:operation name="add" parameterOrder="i1 i2">
<wsdl:input message="impl:addRequest" name="addRequest"/>
<wsdl:output message="impl:addResponse" name="addResponse"/>
</wsdl:operation>
<wsdl:operation name="subtract" parameterOrder="i1 i2">
<wsdl:input message="impl:subtractRequest" name="subtractRequest"/>
<wsdl:output message="impl:subtractResponse" name="subtractResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="CalculatorSoapBinding" type="impl:Calculator">
<wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="add">
<wsdlsoap:operation soapAction=""/>
<wsdl:input name="addRequest">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
</wsdl:input>
<wsdl:output name="addResponse">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/axis/Calculator.jws" use="encoded"/>
</wsdl:output>
</wsdl:operation>

```

```
<wsdl:operation name="subtract">
<wsdlsoap:operation soapAction=""/>
<wsdl:input name="subtractRequest">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
</wsdl:input>
<wsdl:output name="subtractResponse">
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/axis/Calculator.jws" use="encoded"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="CalculatorService">
<wsdl:port binding="impl:CalculatorSoapBinding" name="Calculator">
<wsdlsoap:address location="http://localhost:8080/axis/Calculator.jws"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

APÊNDICE C

MODELO LÓGICO



APÊNDICE D

MODELO FÍSICO

