

**UNIVERSIDADE DO SAGRADO CORAÇÃO**

**ALINE VALEZI**

**PUBLICAÇÃO E DESCOBERTA DE SERVIÇOS EM  
SERVIDORES DE DIRETÓRIOS UDDI**

BAURU  
2013

**ALINE VALEZI**

**PUBLICAÇÃO E DESCOBERTA DE SERVIÇOS EM  
SERVIDORES DE DIRETÓRIOS UDDI**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção do título de Bacharel em Ciências da Computação, sob orientação do Prof. Esp. Henrique Pachioni Martins.

BAURU  
2013

V168p Valezi, Aline

Publicação e descoberta de serviços em servidores de diretórios UDDI / Aline Valezi -- 2013.  
74f. : il.

Orientador: Prof. Esp. Henrique Pachioni Martins.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.

1. UDDI. 2. Arquitetura Orientada a Serviços. 3. Serviços web. I. Martins, Henrique Pachioni. II. Título.

**ALINE VALEZI**

**PUBLICAÇÃO E DESCOBERTA DE SERVIÇOS EM  
SERVIDORES DE DIRETÓRIOS UDDI**

Trabalho de conclusão de curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Esp. Henrique Pachioni Martins.

Banca examinadora:

---

Prof. Esp. Henrique Pachioni Martins  
Universidade Sagrado Coração

---

Prof. Dr. Elvio Gilberto da Silva  
Universidade Sagrado Coração

---

Prof.<sup>a</sup> M.<sup>a</sup> Patrícia Bellin Ribeiro  
Universidade Sagrado Coração

Bauru, 10 de Dezembro de 2013.

*Dedico este trabalho de conclusão de curso à minha família, que sempre esteve presente em minhas decisões e conquistas.*

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, aos meus pais Dalva e Luiz, meus irmãos Luiz Eduardo e Juliana, minha prima Luana, minhas avós Lurdes e Andrelina, obrigada por todo amor, compreensão e incentivo.

Especialmente aos meus pais que não mediram esforços para proporcionar a educação que nunca puderam ter.

A todos os professores e amigos de faculdade, em especial a Sabrina Miranda Chioca, Alexandre Meira Lima e Juliano Firmino que estiveram desde o primeiro ano de faculdade ao meu lado. Obrigada por toda paciência, carinho, companheirismo e, principalmente, por confiarem a mim a amizade de vocês, independentemente da distância que houvesse.

Ao Professor Especialista Henrique Pachioni Martins, por toda a paciência, incentivo e sabedoria compartilhada.

Aos meus amigos da equipe de informática da Prefeitura Municipal de Bauru que, ao longo deste trabalho de conclusão de curso, disponibilizaram seu tempo em me auxiliar e aconselhar. Em especial a Luciana Cristina Martins que sempre me deu força e ideias para conseguir concluir o trabalho.

E a todos os colegas que, de alguma forma, colaboraram, direta ou indiretamente, para a realização deste trabalho.

## RESUMO

Com o mundo globalizado há sempre urgência na execução de tarefas. A agilidade ao realizá-las se tornou primordial, mas sempre tendo em vista a facilidade, qualidade e segurança, empresas que atendam esses preceitos estão disparando no mercado, de modo que a arquitetura orientada a serviços tem uma grande importância nesse papel, seu modelo visa possibilitar que fornecedores criem e publiquem seus serviços Web na rede mundial de computadores, unificando em um único repositório todos os seus negócios, de tal modo que possam ser pesquisados por clientes, tendo a disposição em um único ambiente, a capacidade de localizar o serviço facilitando pesquisas e cotação de preços. Essa característica de disposição cria muita vantagem competitiva tanto aos que oferecem quando aqueles que o requerem. Partindo desse conceito, o presente estudo apresenta um protótipo de servidor que demonstra como são feitos os processos de criação, publicação e descoberta desses serviços Web, as configurações utilizadas para o funcionamento correto do repositório de diretórios, demonstrando os padrões, ferramentas e requisitos necessários para sua implementação. A análise do estudo de caso realizado pela presente pesquisa demonstrou o desempenho positivo do servidor UDDI com relação às buscas realizadas.

**Palavras Chave:** UDDI, Arquitetura Orientada a Serviços, serviços web.

## **ABSTRACT**

With the globalized world there is always urgency in performing tasks. The agility to perform them became paramount, but always keeping in mind the ease, quality and safety, companies that meet these precepts are firing in the market, so that the service-oriented architecture is of great importance in this role, your model aims enable suppliers to create and publish their Web services on the World Wide Web, unifying into a single repository all your business, so they can be searched by customers, and the provision in a single environment, the ability to find the service facilitating research and price quote. This feature creates a lot of competitive advantage available to both when they offer those who require it. Based on this concept, this study presents a prototype server that demonstrates how they are made the process of creating, publishing and discovery of these Web services, the settings used for the proper functioning of the repository directories, demonstrating patterns, tools and requirements for implementation. The analysis of the case study conducted by this research demonstrated the positive performance of the UDDI server with respect to the searches performed.

**Keywords** UDDI, Service Oriented Architecture, web services.



## LISTA DE FIGURAS

Figura 1 - Paradigma básico para Arquiteturas Orientadas a Serviços (SOA).....	14
Figura 2 - Diferentes camadas de uma aplicação orientada a serviços .....	14
Figura 3 - Ciclo de vida de um SW .....	18
Figura 4 - Arquitetura do WSDL .....	20
Figura 5 - Arquitetura Orientada a Serviços .....	22
Figura 6 - Representa o diretório distribuído de sites-operadores UDDI.....	26
Figura 7 - Ciclo de vida de um SW .....	27
Figura 8 - JUDDI – Console .....	31
Figura 9 - Autenticação no servidor UDDI.....	32
Figura 10 - Token de Autenticação .....	33
Figura 11 - Resposta do Token de Autenticação .....	33
Figura 12 - XML da empresa Fornecedora A .....	36
Figura 13 - Resposta XML da publicação da Fornecedora A.....	37
Figura 14 - XML da empresa Fornecedora B .....	37
Figura 15 - Resposta XML da publicação da Fornecedora B.....	38
Figura 16 - XML da empresa Fornecedora C .....	39
Figura 17 - Resposta XML da publicação da Fornecedora C.....	40
Figura 18 - Publicação do serviço da Fornecedora A.....	41
Figura 19 - Publicação do serviço da Fornecedora B.....	41
Figura 20 - Publicação do serviço da Fornecedora C .....	42
Figura 21 - Pesquisa da empresa Fornecedora A.....	43
Figura 22 - Resultado da pesquisa da Fornecedora A.....	44
Figura 23 - Pesquisa de todas as empresas fornecedoras .....	45
Figura 24 - Resultado da pesquisa de todos os fornecedores .....	46
Figura 25 - Pesquisa por data e local.....	47
Figura 26 - Resultado da pesquisa por data e local .....	47
Figura 27 - Pesquisa por horário de embarque .....	48
Figura 28 - Resultado da pesquisa por horário de embarque .....	49

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>9</b>
1.1 MOTIVAÇÃO .....	10
1.2 OBJETIVOS .....	10
1.2.1 Objetivos gerais.....	10
1.2.2 Objetivos específicos.....	10
<b>2 ORGANIZAÇÃO DO TRABALHO</b> .....	<b>11</b>
<b>3 AOS (ARQUITETURA ORIENTADA A SERVIÇOS)</b> .....	<b>13</b>
<b>4 SERVIÇOS WEB</b> .....	<b>17</b>
4.1 WSDL (Web Services Description Language) .....	19
4.2 URI (Uniform Resource Identifier) .....	21
<b>5 UDDI (UNIVERSAL DESCRIPTION, DISCOVERY, AND INTEGRATION)</b> .....	<b>22</b>
5.1 UDDI encontrando Web Services.....	23
5.2 Registro UDDI .....	23
5.3 Modelo de informação estruturada UDDI .....	25
<b>6 METODOLOGIA</b> .....	<b>27</b>
6.1 Ferramentas utilizadas para a configuração do protótipo de servidor UDDI ...	28
6.1.1 Sistema operacional servidor.....	29
6.1.2 Apache jUDDI versão 3.0.4 .....	29
6.1.3 Apache Tomcat 6.9.....	29
6.1.3.1 Servlets .....	30
6.1.4 MySQL 5.5.....	30
6.1.5 Java JDK 1.5.0 .....	30
6.1.6 SoapUI 4.5 .....	30
6.2 Configuração do Servidor UDDI.....	31
6.3 Autenticação no Servidor UDDI .....	32
6.4 Publicação dos fornecedores e seus serviços Web .....	33
6.5 Consulta de serviços no Servidor de repositório UDDI .....	34
<b>7 RESULTADOS</b> .....	<b>35</b>
7.1 Desenvolvimento do protótipo de servidor UDDI.....	35
7.1.1 Descrição XML das empresas fornecedoras .....	35
7.1.2 Descrição XML dos serviços.....	40
7.2 Descoberta de serviços Web.....	42
<b>8 CONSIDERAÇÕES FINAIS</b> .....	<b>50</b>
8.1 TRABALHOS FUTUROS .....	50
<b>REFERÊNCIAS</b> .....	<b>52</b>
<b>APÊNDICE A – ARTIGO DE NEGÓCIOS ELETRÔNICOS E SERVIÇOS WEB</b>	
<b>NEGÓCIOS ELETRÔNICOS E SERVIÇOS WEB</b> .....	<b>55</b>
<b>ANEXO B – SCRIPT DE CRIAÇÃO DO BANCO DE DADOS UDDI</b> .....	<b>63</b>

## 1 INTRODUÇÃO

O mundo virtual cada vez mais acopla tarefas cotidianas na rede mundial de computadores. Deveres tais como, ir ao supermercado ou efetuar o pagamento de uma conta se tornaram ágeis e simples. Percebendo esse potencial de consumo na internet, as empresas que buscam agregar valor estão investindo fortemente nas vendas virtuais e obtendo um retorno positivo. Com tamanho crescimento, foi necessário pensar em segurança e padronização de operações, para que as mesmas se concretizem com sucesso evitando transtornos. Na necessidade de uma composição que pudesse definir e auxiliar no funcionamento desses processos surgiu o paradigma de Computação Orientada a Serviços (COS), que veio para definir aplicações distribuídas, esse paradigma utiliza serviços eletrônicos como elementos fundamentais para o desenvolvimento de aplicações distribuídas. Os papéis nesse paradigma são: fornecedores e consumidores de serviços. (PAPAZOGLU et al., 2003, SANTOS, 2011)

Com a finalidade de disponibilizar esses serviços usa-se a Arquitetura Orientada a Serviço (AOS) que é um novo conceito abstrato, no qual se cria um ambiente de descoberta dinâmico e se faz o uso de Serviços Web através da rede (REIS, 2012).

Nesta estrutura um componente importante é o repositório UDDI que, segundo Reckziegel (2006), é uma especificação técnica que tem como objetivo descrever, descobrir e integrar serviços web. Dessa forma, funciona como intermediário entre os consumidores e fornecedores dos serviços.

Para isso, foi desenvolvido um protótipo para demonstrar como são feitos os processos de publicação, solicitação e localização dos serviços, demonstrando como este novo modelo funciona e possibilitando visualizar uma amostra do seu potencial.

## 1.1 MOTIVAÇÃO

UDDI é uma especificação técnica para descrever, descobrir e integrar serviços na web. Assim é, portanto, um recurso importante para ampliar a distribuição dos serviços prestados, habilitando usuários dos serviços a publicarem e descobrirem serviços na web.

O tema Publicação e Descoberta de Serviços em Servidores de Diretórios UDDI foi escolhido pela sua inovação no mercado e também porque está transformando o conceito de vendas virtuais e agregando valor às empresas, a fim de demonstrar o procedimento de alocação e descoberta dos serviços prestados em um servidor UDDI foi proposto desenvolver um protótipo que mostre como são realizados os processos para publicação, solicitação e localização dos serviços.

## 1.2 OBJETIVOS

### 1.2.1 Objetivos gerais

Os objetivos principais do trabalho foram a instalação e configuração de um servidor de repositórios UDDI, publicação de serviços dentro dos padrões necessários no servidor e posteriormente localizados pelos clientes.

### 1.2.2 Objetivos específicos

Para alcançar os objetivos gerais, alguns objetivos específicos foram necessários, tais como:

- Instalação e configuração do servidor de repositórios UDDI.
- Criação de três fornecedores fictícios de venda de passagens aéreas.
- Desenvolvimento de um serviço de passagem aérea com os padrões WSDL e descrito na linguagem XML que tem como descrição a data da passagem e o horário de saída que será oferecido pela empresa fornecedora A.
- Desenvolvimento de um segundo serviço de passagem aérea com os padrões WSDL e descrito na linguagem XML que tem como descrição a

data e o valor da passagem que será oferecido pela empresa fornecedora B.

- Desenvolvimento de um terceiro serviço de passagem aérea com os padrões WSDL e descrito na linguagem XML que tem como descrição a hora de saída e o valor da passagem que será oferecido pela empresa fornecedora C.
- Publicação desses serviços no servidor no repositório UDDI.
- Criação de um cliente fictício que necessite de uma passagem aérea que atenda as suas necessidades.
- O cliente fictício realiza a solicitação de um serviço no servidor de registro de serviços.
- Localizado o registro a resposta do serviço é devolvida para o cliente.

## **2 ORGANIZAÇÃO DO TRABALHO**

Para melhor compreensão, o projeto está organizado como segue:

O capítulo 3 explica sobre todo o contexto que envolve a Arquitetura Orientada a Serviços, quais seus conceitos e sua composição.

O capítulo 4 explica o conceito de um serviço web, quais suas particularidades, linguagem utilizada, protocolos, padronização e definição da sua estrutura. No capítulo 4.1 é descrito o padrão WSDL para composição das do XML que descreve a interface de um Serviço Web. No capítulo 4.2 descreve o endereçamento URI.

O capítulo 5 relata sobre o servidor de repositório UDDI, seus padrões, modelagem e composição. No capítulo 5.1 refere-se à publicação de um serviço web em um servidor de diretórios UDDI. No capítulo 5.2 descreve um registro de serviço no servidor UDDI. O capítulo 5.3 descreve o modelo de estruturação de um arquivo UDDI.

O capítulo 6 expõe a metodologia que compõe o trabalho e descrição da criação do protótipo. No capítulo 6.1 descreve as ferramentas utilizadas para a instalação e configuração do servidor UDDI. O capítulo 6.2 descreve como foi feita a configuração do repositório. No capítulo 6.3 mostra como é realizado uma

autenticação no servidor. O capítulo 6.4 esclarece a necessidade do token. No capítulo 6.5 relata sobre como o cliente pode realizar pesquisas.

No capítulo 0 são apresentados os resultados obtidos e a discussão acerca desses resultados. O capítulo 7.1 encontra-se detalhado os processos do protótipo para a modelagem dos serviços até as pesquisas realizadas com seus resultados obtidos.

Por fim, no capítulo 8 são apresentadas as conclusões e considerações finais sobre o projeto, e no capítulo 8.1 citação de trabalhos que podem contribuir para o desenvolvimento de projetos futuros.

### 3 AOS (ARQUITETURA ORIENTADA A SERVIÇOS)

Nos últimos anos Arquitetura Orientada a Serviços (AOS) e Computação Orientada a Serviços (COS) emergiram como um paradigma para lidar com a complexidade de aplicações distribuídas. A arquitetura é um conceito abstrato e sua definição é composta por um conjunto de decisões importantes em questões como: organização do sistema, seleção de seus elementos estruturais e interfaces, considerando comportamento, decomposição desses elementos e, finalmente, o estilo para composição dos elementos estáticos e dinâmicos, dentro do sistema (OLIVEIRA et al., 2008, PAPAZOGLU et al., 2003).

As principais características da SOA segundo Grossi (2005) são:

- Serviços são componentes de software que publicaram seus contratos (ou interfaces); esses contratos são independentes de plataforma, linguagem ou sistema operacional;
- Consumidores podem encontrar serviços dinamicamente;
- Serviços são interoperáveis.

É importante destacar que, mesmo sendo uma condição essencial, serviços por si só não implementam SOA (OLIVEIRA et al., 2008).

Uma visão geral do paradigma SOA é mostrado na Figura 1 - Paradigma básico para Arquiteturas Orientadas a Serviços (SOA), observamos a entidade Produtor que cria e registra os seus serviços em um servidor de diretórios e a partir dessa publicação a entidade consumidora pode fazer buscas e localizar os serviços publicados, após a localização a entidade consumidora pode invocá-los.

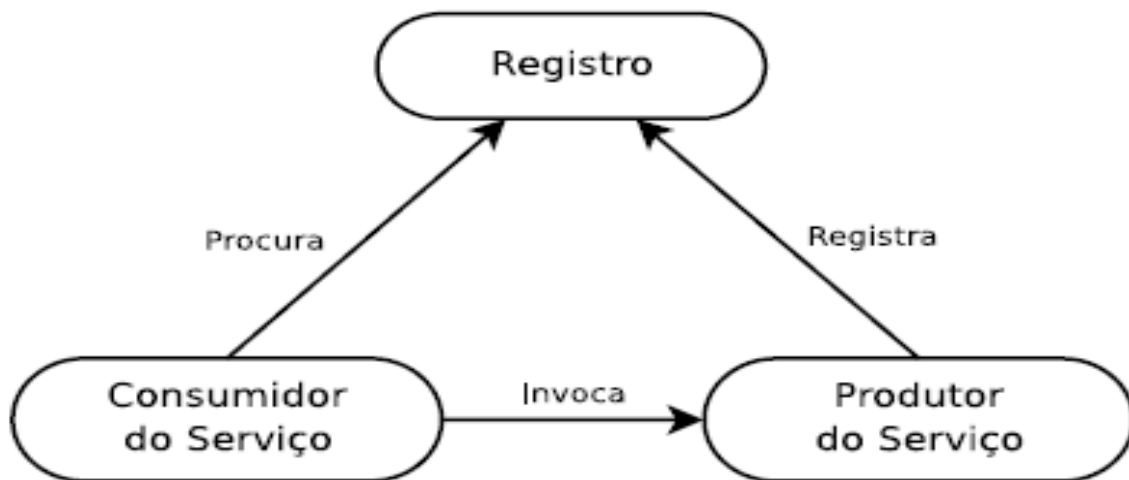


Figura 1 - Paradigma básico para Arquiteturas Orientadas a Serviços (SOA)  
 Fonte: Grossi, 2005, p. 9

Uma aplicação que utiliza SOA normalmente é dividida em camadas, como ilustrado na Figura 2 - Diferentes camadas de uma aplicação orientada a serviços.

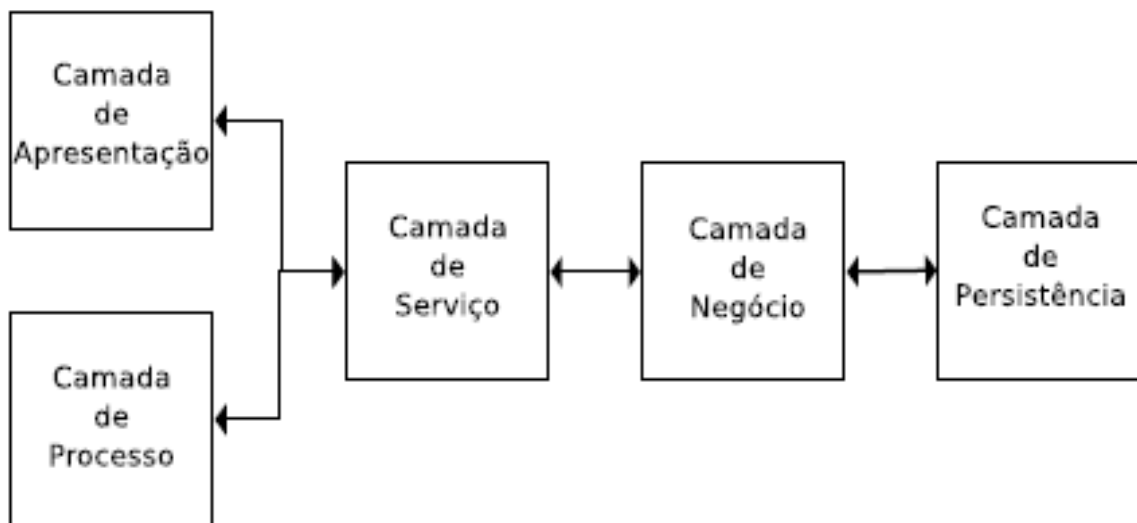


Figura 2 - Diferentes camadas de uma aplicação orientada a serviços  
 Fonte: Grossi, 2005, p. 9

As camadas correspondem as seguintes descrições (STRELESKI et al., 2009):

- **Camada Corporativa ou Camada de Apresentação:** Camada que se preocupa em gerar um modelo responsável por descrever o negócio da empresa. Esse modelo é capaz de identificar os processos de negócio



chaves que trazem vantagem competitiva para a empresa, e os processos de negócios que sustentam a organização.

- **Camada de Processos:** Camada que identifica e caracteriza os processos de negócio. Cada processo pode ser composto por sub-processos e tem como responsabilidade atender a uma determinada área funcional. Os sub-processos podem ser decompostos para expor suas dependências da camada de serviço. É errado pensar que processo e serviço é a mesma coisa. Processos são definidos uma única vez, e usados dentro de um contexto único. Já os serviços podem ser construídos dentro de processos de negócio e são criados para estarem disponíveis para o consumo de usuários, sistemas e outros serviços.
- **Camada de Serviços:** Camada que tem como responsabilidade mapear os serviços que compõem as funcionalidades básicas, técnicas e de negócio. A equipe de negócio identifica as necessidades críticas de negócio, enquanto a área de TI cria soluções técnicas para atender aos requisitos de negócio.
- **Camada de Componentes ou Camada de Negócio:** Camada que mapeia os componentes que possuem potencial para se transformarem em serviço, é comum se utilizar a metodologia bottom-up (parte-se da visão de criar aplicações que contém potencial para serem utilizadas em outros contextos de negócio). Componentes são blocos de construção de serviços na arquitetura SOA.
- **Camada de Objetos ou Camada de Persistência:** Camada responsável por identificar e caracterizar uma grande quantidade de classes de objetos, seus atributos e relacionamentos.

O mais importante princípio de SOA é a interoperabilidade. É fundamental que todos os componentes se comuniquem independente da linguagem em que foram construídos, do sistema operacional em que estão sendo executados e da arquitetura de hardware. Entretanto, apesar de a interoperabilidade do Web Services ser o seu benefício chave, a diversidade de implementações de bibliotecas para

suporte de *Web Services* e a flexibilidade permitida em alguns pontos da especificação torna difícil a concretização dessa característica (GROSSI, 2005).

A arquitetura permite a automação de processos de negócio da empresa através da integração de diversos componentes com funções bem definidas, chamados de serviços. SOA possui como base tecnologias web services, tendo como características prioritárias o fraco acoplamento, composição, padronização, interoperabilidade e a reutilização (STRELESKI et al., 2009).

## 4 SERVIÇOS WEB

A evolução das aplicações Web é nítida ao longo dos anos, criação de modelos e a padronização de processos realizados online estão facilitando o surgimento de novas tecnologias que atenda a crescente demanda do consumo na rede mundial de computadores. Segundo Albinader (2005, p.20) essas poderosas ferramentas também podem ser usadas para transações entre corporações, governos entre outros. Surgindo assim o conceito de serviços Web que para Santos (2011, p.7) pode ser considerado como uma aplicação que tem uma interface descrita em WSDL, registrada em um diretório de serviços via protocolo UDDI, e que interage com clientes usando troca de mensagens XML encapsulada em envelopes SOAP. Complementando a ideia Fantinato (2007, p.37) descreve um serviço como:

Um tipo específico de serviço eletrônico que usa padrões abertos da Internet para a sua descrição, busca e requisição de operações. Serviços Web podem ser vistos como aplicações, disponíveis via Internet, que podem ser usadas para executar determinados serviços eletrônicos. Eles podem ser desde simples serviços – como a realização de verificações de números de cartões de crédito, até serviços complexos – como manipulação de aplicações de hipotecas

Um serviço Web segue alguns processos que podem ser observados na - Ciclo de vida de um SW. Primeiramente (etapa 1) o provedor de serviço depois de tê-lo descrito na linguagem XML e estando no padrão WSDL publica-os no registro de serviços, assim esse serviço já está apto a ser descoberto, em seguida (etapa 2) um cliente faz a solicitação de um serviço no servidor de registro, posteriormente (etapa 3) o servidor o encontra e devolve a resposta para o cliente, O mesmo contata o provedor de serviços (etapa 4) e é feito um contrato eletrônico entre as partes (etapa 5).

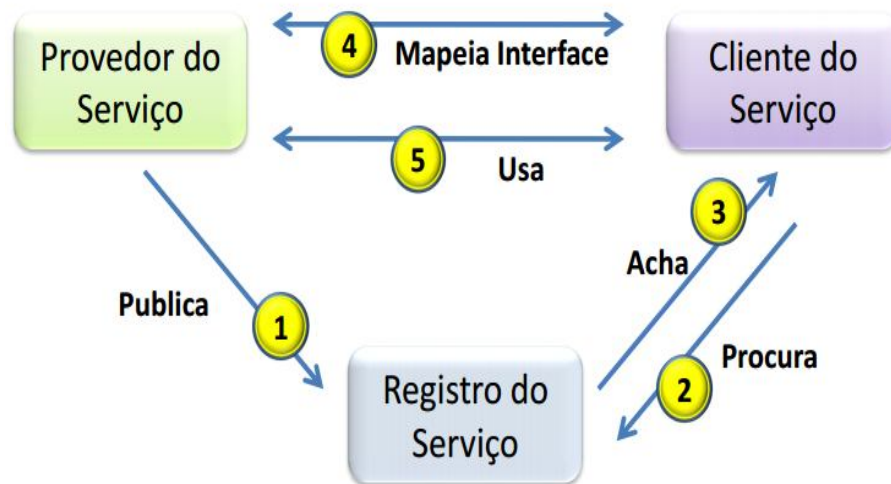


Figura 3 - Ciclo de vida de um SW  
 Fonte: Tizzo, 2010, p. 10

A concepção de um serviço Web é uma estrutura modelada que permite a livre comunicação. Fantinato (2007, p.39) reforça essa ideia dizendo que a padronização dos serviços permite que eles sejam executados em qualquer plataforma de software e hardware. Assim, os aplicativos se tornam mais fáceis de serem disponibilizados e consumidos já que todos usam o mesmo padrão.

Principais padrões usados para a tecnologia de serviços Web:

- **WSDL (Web Services Description Language):** Um arquivo WSDL é um documento XML que descreve um conjunto de mensagens SOAP e como elas são retornadas. Incluindo capacidade, interface, comportamento e qualidade de serviço. Os arquivos têm padrões bem definidos tornando viável para descrever as interfaces web services (VIEIRA et al., 2002, FANTINATO, 2007).
- **UDDI (Universal Description, Discovery, and Integration):** Um repositório para publicação e busca de serviços web padronizados e estruturados, são armazenadas informações sobre os serviços. Segundo Fantinato (2007) o padrão UDDI permite que provedores de serviços possam registrar seus serviços Web, usando as descrições WSDL, para que clientes de serviços possam descobri-los. A arquitetura dos serviços web segue o conceito de SOA (Service Oriented Architecture) – arquitetura orientada a serviços.

- **SOAP (Simple Object Access Protocol):** protocolo que define um mecanismo para a comunicação geral entre serviços Web na Internet. Ele define o formato das mensagens que são trocadas entre clientes de serviços, provedores de serviços e diretórios de serviços (MITRA, 2010).

O arquivo precisa ser descrito por uma linguagem, preferencialmente interpretável por computador. O padrão atual para a especificação de negócios é a linguagem WS-BPEL. WS-BPEL é uma linguagem para especificar o comportamento de processos de negócios baseados em web services. (OASIS, 2007)

Oasis (2007) descreve que a finalidade da linguagem WS-BPEL é:

Os processos de negócios podem ser descrito de duas maneiras. Negócio executável processa comportamento do modelo real de um participante de uma interação de negócios. Processos de negócios abstratos são processos parcialmente especificados que não se destinam a ser executado. Um processo abstrato pode esconder alguns dos requeridos operacionais. Processos abstratos têm um papel descritivo, com mais de um caso de uso possível, incluindo o comportamento observável e modelo de processo. WS-BPEL é feito para ser usado para modelar o comportamento de ambos executável e processos abstratos.

WS-BPEL oferece uma linguagem para a especificação de processos de negócio executáveis e abstrato. Ao fazer isso, ele estende o modelo de interação Web Services e permite que ele suporte a transações de negócios. WS-BPEL define um modelo de integração interoperável que deve facilitar a expansão do processo de integração automatizada, tanto no intra-empresas e os espaços de business-to-business.

Web Services é identificado por uma URI (Unique Resource Identifier), e são descritos e definidos usando XML (CUNHA, 2002).

#### 4.1 WSDL (WEB SERVICES DESCRIPTION LANGUAGE)

O WSDL é uma linguagem padrão descrita em XML que descreve a interface de um Serviço Web. Um documento WSDL define o nome de um Serviço Web, os nomes e argumentos de seus métodos, protocolo de acesso etc. Um Arquivo WSDL possui as informações necessárias para que o serviço seja publicado, descrito e encontrado, entre dados que descreve a qualidade do serviço, comportamento e interface. (TIZZO, 2010, CHINNICI et al., 2006).

Para Fantinato (2010) ela possibilita a descrição de informações da interface de um serviço, tais como seus parâmetros de entrada e de saída, de modo que um usuário possa usá-lo sem necessariamente conhecer sua implementação.

Os elementos do WSDL são ilustrados na

Figura 4 - Arquitetura do WSDL. Dentre esses, os principais elementos são os tipos de porta (port type), operações (operations) e mensagens (messages).

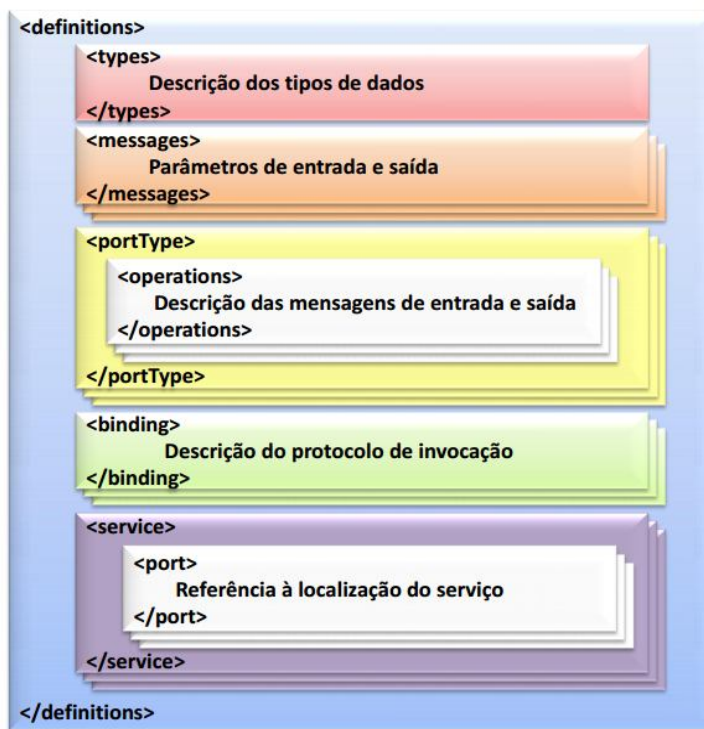


Figura 4 - Arquitetura do WSDL  
Fonte: Tizzo, 2010, p. 21

Segue a descrição das tags de identificação do WSDL como ilustrado na Figura 4.

- **Type:** define um tipo de dado complexo, baseado em tipos padrão de dados simples, para ser usado em situações específicas;
- **Operation:** define uma ação oferecida por um serviço Web;
- **Port Type:** define um conjunto de operações oferecido por um serviço Web;
- **Message:** define um tipo de dado a ser transmitido pelo sistema, durante a invocação a operações de serviços Web e o retorno deles;

- **Binding:** define um protocolo de comunicação e o formato de dados para as mensagens de um determinado tipo de porta (port type);
- **Port:** define um ponto final único, formado por uma combinação de uma ligação (binding) e um endereço de rede;
- **Service:** define um conjunto de pontos finais (port) relacionados.

## 4.2 URI (UNIFORM RESOURCE IDENTIFIER)

Qualquer recurso disponível na Web – documento HTML, imagem, vídeo, programa etc. – tem que ter um endereço único para que seja possível encontrá-lo de qualquer lugar do mundo (URI). Para fazer uma ligação para outra página HTML será preciso referenciar o endereço URI desta outra página (URI).

Um Identificador Uniforme de Recursos (Uniform Resource Identifier - URI) e uma string de caracteres curta que identifica um recurso abstrato ou físico (por exemplo, uma página da Web). Um Localizador Uniforme de Recursos (Uniform Resource Locator - URL) e uma URI que identifica um recurso através do seu mecanismo de acesso primário (por exemplo, sua "localização" de rede), em vez do nome ou algum outro atributo daquele recurso. Um Nome Uniforme de Recurso (Uniform Resource Name - URN) é uma URI que precisa manter-se única e persistente globalmente, mesmo quando o recurso deixa de existir ou se torna indisponível (DAVID, 2010).

## 5 UDDI (UNIVERSAL DESCRIPTION, DISCOVERY, AND INTEGRATION)

Web services são significativos somente se os potenciais utilizadores podem encontrar informações suficientes para permitir a sua execução. O foco da UDDI é a definição do que deve e como ser feito, independentemente de plataformas, para que negócios possam descrever publicamente seus serviços disponíveis, localizar outros serviços do seu interesse e compartilhar informações sobre pontos convergentes em um registro global (ALBINADER et al., 2006, OASIS, 2004).

O padrão UDDI permite que provedores de serviços possam registrar seus serviços Web, usando as descrições WSDL, para que clientes de serviços possam descobri-los (FANTINATO, 2007). Do ponto de vista operacional, UDDI pode ser entendido como mecanismo que possibilita as organizações localizarem outras organizações e conduzirem transações de negócios de forma muito rápida, empregando padrões estabelecidos e aceitos por todos os participantes (ALBINADER et al., 2006). Podemos observar esse comportamento na Figura 5 - Arquitetura Orientada a Serviços, os serviços são publicados em um servidor de registros UDDI, ele é o responsável pelo armazenamento e possibilita que o consumidor descubra os serviços.

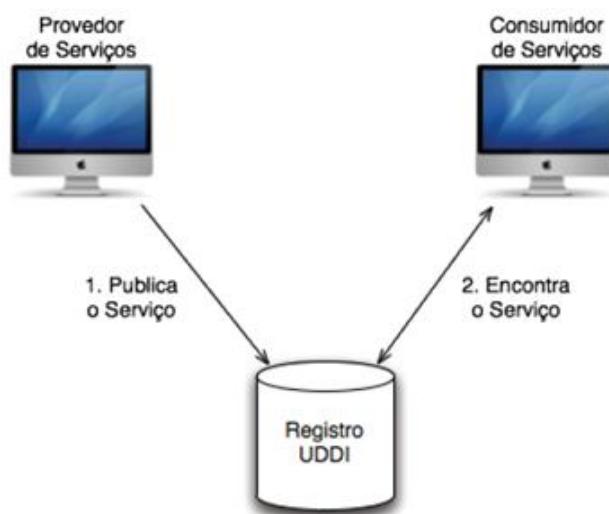


Figura 5 - Arquitetura Orientada a Serviços  
Fonte: adaptada de (ALONSO et al., 2004).



## 5.1 UDDI ENCONTRANDO WEB SERVICES

Um registro UDDI contém informações sobre negócios, dados da organização e dos serviços que ela oferece. Com isso, um usuário pode escolher, em uma lista, que web service (informações climáticas, data e hora, cotação da bolsa de valores etc.) que ele deseja acessar. Todas as APIs, bem como a definição de UDDI, são especificadas em XML, empacotadas dentro de envelopes SOAP e utilizam o HTTP como protocolo de transporte (ALBINADER et al., 2006).

## 5.2 REGISTRO UDDI

Um registro UDDI assemelha-se muito com um motor de pesquisa da internet, como Google, AltaVista e outros. Entretanto, a literatura faz questão de compará-lo a uma lista telefônica, que disponibiliza a possibilidade aos potenciais consumidores de serviços de localizar facilmente as entidades de negócios desejadas e os serviços fornecidos por estas (ALBINADER et al., 2006).

O UDDI é constituído em três grandes partes ou divisões descritas a seguir.

- **Yellow Pages:** Localizam negócios por tipo de negócio (TODD, 2003). Incluem informações de classificação baseada na taxionomia (Teoria ou nomenclatura das classificações científicas. Fonte: Dicionário Aurélio da língua portuguesa) padrão da indústria ou segmento da qual a empresa e os serviços, por ela, oferecidos fazem parte. A metáfora empregada no mecanismo das páginas permite que qualquer termo válido da taxonomia seja associada a qualquer página branca de negócio (ALBINADER et al., 2006).
- **White Pages:** Localizam negócios por nomes, endereços e assim por diante (TODD, 2003). Essa página consta informações que permite ao consumidor verificar endereços, informações para contato com a empresa, tomar conhecimento verificar endereços, informação para contato com a empresa, tomar conhecimento dos identificadores utilizados pelo negócio. A consulta se assemelha a consulta que se realiza à lista telefônica, tendo apenas o nome da empresa como

informação e se deseja obter o endereço e o telefone da mesma (ALBINADER et al., 2006).

- **Green Pages:** Localizam informações sobre os serviços que um negócio oferece (TODD, 2003). Os serviços na internet oferecidos pelas empresas das páginas brancas são descritos nas páginas verdes (ALBINADER et al., 2006).

As informações que permitem que os dados sejam acessados dessa maneira são armazenadas em uma estrutura XML. Essa estrutura é parte da especificação UDDI.

Segundo Todd (2003) é especificado as estruturas de dados UDDI da seguinte maneira:

- Um XML Schema que define uma entrada UDDI.
- Uma UDDI API abrangendo métodos para publicar serviços e métodos para acessar serviços.
- Uma especificação para a replicação das informações de serviços entre registros.
- Uma especificação para operadores de registros abrangendo questões como segurança.

Quatro tipos de informações, entidades lógicas, compõem o registro UDDI: entidade de negócio, serviço de negócio, especificação de ponteiros e tipos de serviços. Todas as entidades de negócios possuem identificador único, nome de negócio, informações simples de contato, descrição resumida sobre negócios, informações simples de contato, descrição resumida sobre o negócio, lista de categorias que descrevem e classificam o negócio e uma URL que aponta para informações adicionais sobre negócios (ALBINADER et al., 2006).

O serviço de negócio dispõe da descrição de serviços disponibilizados, lista de categorias que descreve e classifica o serviço e uma URL com informações adicionais sobre serviços como informações que o compõe (ALBINADER et al., 2006).

Cada entidade de serviço de negócio possui uma listagem para modelo de documentos que apontam para informações adicionais sobre serviços. (ALBINADER et al., 2006)

Cada entidade de serviço de negócio inclui a lista de ligações para templates (modelos) que apontam para informações adicionais sobre o serviço. A ligação com o template pode apontar para a URL, que disponibiliza informações de como o serviço pode ser invocado. A entidade de especificação de ponteiros, terceiro componente do registro UDDI, contém informações que associam o serviço com um tipo de serviço, outra entidade do registro UDDI. (ALBINADER et al., 2006)

Para a padronização (tipificação de serviços) pode ser feita pela Service Type, Technical Model (tModel), que compõe o registro UDDI. O tModel define tipos de serviços que as empresas podem oferecer, baseando nesses tipos as empresas oferecem seus serviços. Este componente define a informação que cada serviço oferecido dentro do padrão deve conter o nome tModel, o nome da organização utilizado pelo tModel, as categorias que definem o serviço e os ponteiro para as especificações de tipo de serviços, incluindo as definições de interface, protocolos de mensagens e formatos e protocolos de segurança. Tipicamente, o tModel refere-se a um documento WSDL para um serviço. (ALBINADER et al., 2006)

### 5.3 MODELO DE INFORMAÇÃO ESTRUTURADA UDDI

Em seu núcleo primário, UDDI é composto por quatro tipos de estruturas: entidade de negócios, entidade serviço negócios, template de ligação e tModel. (ALBINADER et al., 2006)

UDDI usa a linguagem de esquema XML e sua terminologia, como "sequência" e "escolha" para descrever formalmente suas estruturas de dados. (OASIS, 2004)

UDDI Business Registry, também conhecido como "UDDI cloud services" é uma implementação operacional completa da especificação UDDI. Tal parte habilita qualquer um a buscar dados UDDI existentes, e também, a qualquer empresa auto-registra e seus respectivos serviços. (OASIS, 2007)

A arquitetura técnica de UDDI, segundo OASIS (2007), consiste de três partes:

- O Modelo de Informação UDDI – Um XML Schema para descrever negócios e serviços Web.
- A API UDDI – Uma API baseada em SOAP para publicação e busca de informação UDDI.
- O UDDI Business Registry (UDDI cloud services) – Sites-operadores que proveem implementações da especificação UDDI e sincronizam todos os dados.

Os UDDI Business Registries (UBR's) proporcionam um diretório de sites-operadores UDDI, fisicamente distribuído, mas logicamente centralizado. Isto significa que dados submetidos a um site-operador serão automaticamente replicados através de todos os outros sites-operadores. Tal replicação não ocorre instantaneamente, mas os sites-operadores se sincronizam para atualização de registros, em intervalos de tempo diariamente (ALBINADER et al., 2006), assim como ilustrado na Figura 6 - Representa o diretório distribuído de sites-operadores UDDI

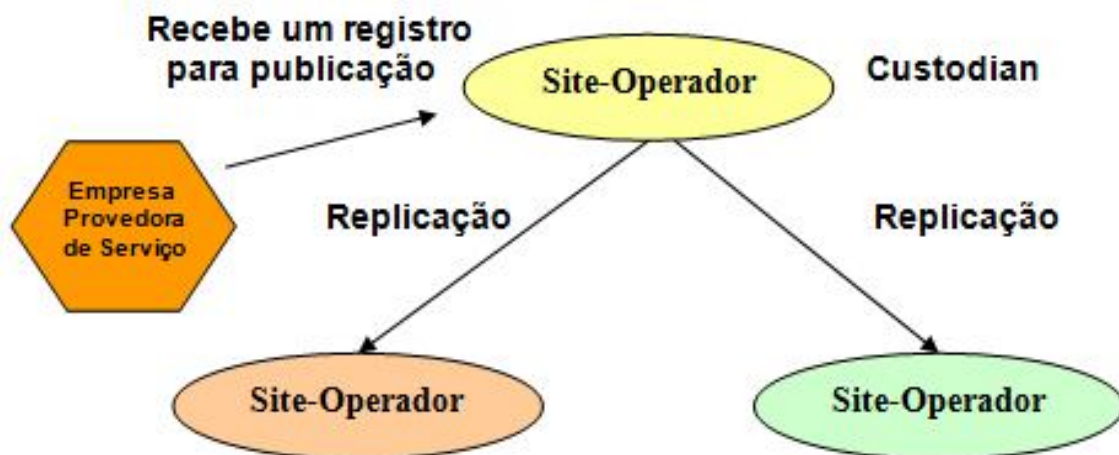


Figura 6 - Representa o diretório distribuído de sites-operadores UDDI  
Fonte: Albinader et al., 2006.

## 6 METODOLOGIA

Esse trabalho focou em estudar uma parte da arquitetura orientada a serviços, dentro dessa arquitetura existe um processo técnico chamado de UDDI, esse processo tem a função de descrever, descobrir e integrar serviços na Web. Portanto, é uma parte crítica da pilha de protocolos dos serviços Web para habilitar os usuários a publicação e descoberta. Para demonstrar esse processo será necessária a representação da Figura 7

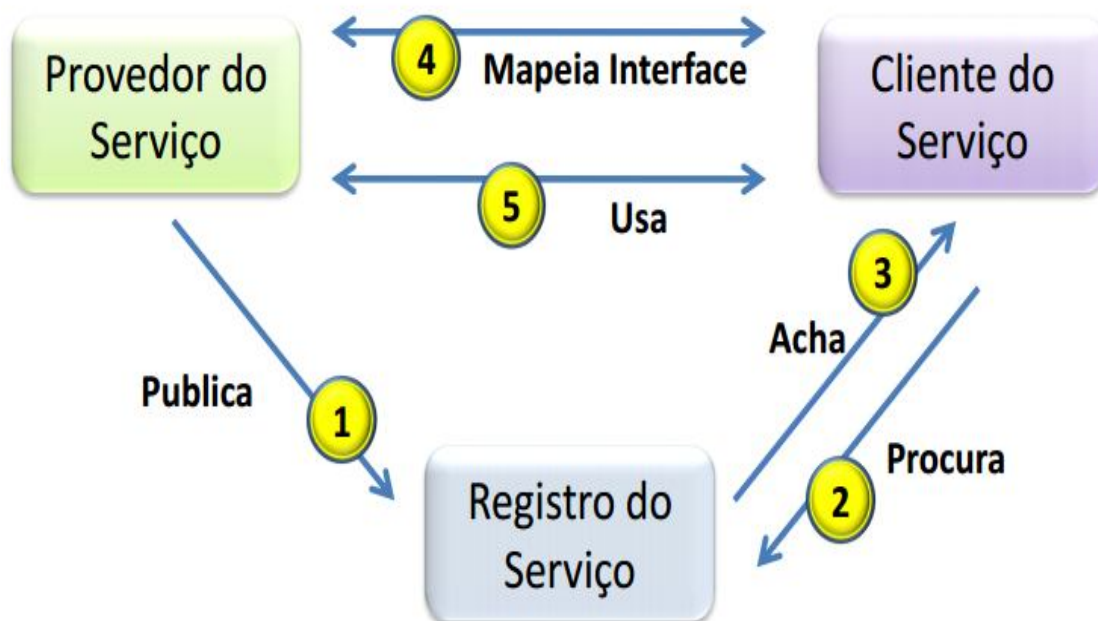


Figura 7 - Ciclo de vida de um SW  
Fonte: Tizzo, 2010, p. 10

Na etapa 1 existe um fornecedor que implementa nos moldes de um arquivo WSDL, a partir da descrição o serviço é publicado em um repositório de arquivos UDDI. A etapa 2 mostra que após os serviços serem publicados em um repositório UDDI os clientes podem pesquisar os serviços que atendam as suas necessidades empresariais e a partir disso é feita uma busca no repositório UDDI. A etapa 3 faz a devolução dos serviços mediante a pesquisa realizada pelo cliente. Na etapa 4 existe uma negociação entre o cliente e o fornecedor posterior a localização do serviço, o cliente começa a fazer as requisições para o servidor que o responde

normalmente por um protocolo de comunicação que é o SOAP. Para a etapa 5, uma vez que a negociação começa vai ter a comunicação do SOAP e para confirmar se ambas as partes estão de acordo ou não existe uma entidade que é a monitora que após o contrato eletrônico fica responsável pelo monitoramento dos serviços prestados pelo fornecedor.

O foco deste trabalho foi a instalação e a configuração do servidor UDDI para realização das operações, posteriormente foi desenvolvido as etapas um, dois e três, ou seja, a partir da descrição dos serviços oferecidos pelo fornecedor foi gerado um WSDL, em seguida fora realizada a etapa dois que é a publicação dos negócios em um repositório UDDI, depois, a etapa três que permitiu ao cliente realizar buscas dos serviços oferecidos pelos fornecedores no servidor trazendo os negócios que foram localizados segundo os parâmetros utilizados nas pesquisas. As demais etapas serão trabalhos futuros.

O cliente fictício que fará as requisições e os fornecedores que disponibilizaram os serviços poderão estar em máquinas independente da qual está alocado o repositório UDDI, acompanhando a ideia central do repositório de serviços de realizar consultas ao servidor onde quer que o consumidor se encontre independente de plataforma, Para facilitar o desenvolvimento a parte do fornecedor e cliente se encontra na mesma máquina onde está alocado o servidor UDDI.

O desenvolvimento do protótipo de servidor UDDI utilizou de ferramentas descritas no tópico 6.1.

## 6.1 FERRAMENTAS UTILIZADAS PARA A CONFIGURAÇÃO DO PROTÓTIPO DE SERVIDOR UDDI

Primeiramente, serão descritos as ferramentas utilizadas para a instalação e configuração do protótipo de servidor UDDI, o qual é responsável pelo gerenciamento dos serviços Web criado pelos fornecedores e das pesquisas realizadas pelos consumidores de serviços.

### **6.1.1 Sistema operacional servidor**

Para a alocação do servidor UDDI foi instalado o Sistema Operacional Servidor Windows Server 2008 R2, é uma plataforma que suporta aplicações e serviços Web (MICROSOFT). O Sistema Operacional fornece serviços e recursos para outros sistemas em uma rede tendo um abundante conjunto de recursos e opções de configuração, tem todos os mecanismos necessários pra controlar, administrar, e distribuir os programas, aplicativos e funções necessárias, assim sendo um ambiente favorável para alocação do servidor UDDI.

### **6.1.2 Apache jUDDI versão 3.0.4**

O Apache jUDDI (pronuncia-se "Judy") é um servidor web com o fonte aberto de implementação Java da Universal Description, Discovery, and Integration (UDDI) com especificação de Serviços Web (APACHE jUDDI).

Principais características:

- Código aberto
- Independente de Plataforma
- Use com qualquer banco de dados relacional que suporta o padrão ANSI SQL (MySQL, Oracle, DB2, Sybase, Derby etc)
- Implementável em qualquer servidor de aplicação Java que suporta a especificação Servlet
- Fácil integração com sistemas de autenticação existentes

### **6.1.3 Apache Tomcat 6.9**

O Tomcat é um servidor Web Java, mais especificamente, um container de servlets. Ele tem a capacidade de atuar também como servidor web, ou pode funcionar integrado a um servidor Web dedicado como o Apache ou o IIS (APACHE TOMCAT). Para a manipulação dos recursos do servidor UDDI foi necessário um ambiente servidor web HTTP. O Apache tem a função de ler o programa, e converter fazendo os sistemas rodarem em modo usuário, ou seja, transformando tudo o que

foi escrito em linhas de comando para uma forma visual e de interação com os utilizadores do sistema (APACHE TOMCAT).

#### **6.1.3.1 SERVLETS**

Servlets são classes Java, desenvolvidas de acordo com uma estrutura bem definida que quando instaladas e configuradas em um Servidor que implemente um Servlet Container, podem tratar requisições recebidas de clientes Web, como por exemplo os Browsers (ORACLE).

#### **6.1.4 MySQL 5.5**

O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface (MYSQL). O banco de dados tem papel fundamental no armazenamento das autenticações, fornecedores e seus serviços.

#### **6.1.5 Java JDK 1.5.0**

Java Development Kit (JDK) significa Kit de Desenvolvimento Java, e é um conjunto de utilitários que permitem criar sistemas de software para a plataforma Java. É composto por compilador e bibliotecas (ORACLE). Através do Java foi possível realizar a comunicação do cliente e o servidor de repositórios UDDI.

#### **6.1.6 SoapUI 4.5**

SoapUI é uma solução de teste gratuito e open source multi-plataforma funcional. Com uma interface gráfica e recursos de classe empresarial, SoapUI oferece cobertura de teste completo e suporta todos os protocolos e tecnologias padrão (SOAPUI).



## 6.2 CONFIGURAÇÃO DO SERVIDOR UDDI

Para instalação e configuração do protótipo de servidor UDDI foi instalado o Windows Server 2008. Posteriormente inserido o Java para comunicação e interpretação dos serviços publicados. Para que possam ser armazenados os serviços Web publicados, o MySql foi instalado, em seguida um banco de dados foi criado segundo o script do ANEXO B – SCRIPT DE CRIAÇÃO DO BANCO DE DADOS UDDI, assim estando apto a receber os registros necessária de armazenamento das empresas, seus respectivos serviço e suas autenticações.

O Apache jUDDI é uma implementação Java de um servidor UDDI, para a comunicação do servidor com a aplicação foi necessário instalar o Java JDK, os arquivos e bibliotecas do Apache jUDDI foram alocado e configurados no Windows Server 2008, possibilitando as operações de comunicação com o banco de dados. O jUDDI fornece um interface para operar diretamente com a API usando mensagens SOAP assim como mostrado na Figura 8 - JUDDI – Console.

**Apache jUDDI version 3.0.4**

*Welcome to Apache jUDDI!*

**jUDDI Documentation**

- [jUDDI Portal](#)
- [View](#) service listing
- [Visit](#) the Apache-jUDDI Home Page
- [Report](#) a bug
- [Participate! We love you..](#)
- [UDDI library](#)
- [jUDDI User Guide](#)
- [jUDDI Developer Guide](#)
- [UDDI v3.0.2 Specification](#)
- [jUDDI wiki](#) - add your own docs
- [jUDDI Blog](#)

**jUDDI Installation**

jUDDI has been successfully installed!

**Node Information**

Root Partition: uddi:juddi.apache.org  
 Node Id: uddi:juddi.apache.org:businesses-asf  
 Name: An Apache jUDDI Node  
 Description: This is a UDDI v3 registry node as implemented by Apache jUDDI.

Figura 8 - JUDDI – Console  
 Fonte: Elaborado pelo autor.

### 6.3 AUTENTICAÇÃO NO SERVIDOR UDDI

Para segurança das publicações dos fornecedores o servidor UDDI requer autenticações para realizar as operações como ilustrado na Figura 9 - Autenticação no servidor UDDI.



Figura 9 - Autenticação no servidor UDDI  
Fonte: Elaborado pelo autor.

A fim de facilitar o entendimento das publicações foi utilizado o SoapUI que é um software de código aberto interface amigável para a realização de publicação e consulta em servidores UDDI, o SoapUI pode ser utilizado tanto da parte fornecedora facilitando a publicação dos seus serviços como da parte do cliente que faz as buscas no servidor.

Para o SoapUI realizar as operações foi requisitado ao servidor UDDI o token de segurança do servidor. A parte responsável pela segurança do UDDI define a API para obter um token. As APIs UDDI são SOAP baseado na interoperabilidade de sistemas. Com um token de segurança válido um editor pode publicar no servidor e pode ser usado para todas as operações.

Para a requisição do token de segurança uma mensagem XML nos padrões WSDL é enviada ao servidor como ilustrado na Figura 10 - Token de Autenticação. Na linha 4 os itens "userID" e "cred" estão preenchidos com as credenciais de autenticação do fornecedor no servidor UDDI.

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:uddi-org:api_v3">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <urn:get_authToken userID="root" cred="root"/>
5   </soapenv:Body>
6 </soapenv:Envelope>

```

Figura 10 - Token de Autenticação  
Fonte: Elaborado pelo autor.

Para cada fornecedor deve ser gerada uma autenticação única dando segurança a empresa, porém, pensando na facilidade do estudo foi utilizada a penas uma autenticação que será utilizada por todos os fornecedores.

O servidor retorna uma mensagem com o token de autenticação a ser utilizado como mostra na Figura 11 - Resposta do Token de Autenticação.

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <authToken xmlns:ns3="urn:uddi-org:policy_v3">
4       <authInfo>authtoken:5eaeac94-49ac-4f0a-a83d-9f0b7e28201b</authInfo>
5     </authToken>
6   </soap:Body>
7 </soap:Envelope>

```

Figura 11 - Resposta do Token de Autenticação  
Fonte: Elaborado pelo autor.

A linha 4 da Figura 11 contem a resposta de token do servidor. O token serve como uma identificação de autorização para publicações no servidor, essa identificação será necessária para as transações realizadas.

Após autorização as publicações podem ser realizadas no servidor.

## 6.4 PUBLICAÇÃO DOS FORNECEDORES E SEUS SERVIÇOS WEB

Os tokens funcionam como uma chave de validação única, assim, assegurando a empresa segurança nas suas operações e manutenção de seus serviços. Após a autenticação os respectivos fornecedores estão aptos a descreverem seus serviços nos padrões WSDL na linguagem XML e os publicaram no servidor.

## 6.5 CONSULTA DE SERVIÇOS NO SERVIDOR DE REPOSITÓRIO UDDI

O cliente faz requisições de serviços nos padrões WSDL na linguagem XML, o servidor recebendo essa solicitação começa o processo de localização baseados nos parâmetros de pesquisas informados pela descrição do XML de busca, encontrado o(s) serviço(s) o servidor retorna a resposta baseado nos padrões WSDL. É importante resaltar que na parte cliente não é necessário realizar nenhuma autenticação, já que o mesmo só possui o privilégio da busca.

## **7 RESULTADOS**

### **7.1 DESENVOLVIMENTO DO PROTÓTIPO DE SERVIDOR UDDI**

O capítulo descreve um protótipo, cuja aplicação está voltada para a área empresarial, que visa auxiliar na disputa competitiva mediante fácil disponibilização de seus serviços prestados.

Para a validação da proposta deste trabalho fez-se um estudo de caso com três companhias que vendem passagens aéreas, o serviço da primeira companhia aérea ofereceu como descrição do seu negócio a data e hora para embarque de um voo de Bauru a Salvador; a segunda companhia aérea ofereceu como descrição do seu negócio a data e valor da passagem para um voo de Bauru a Salvador; e a terceira companhia aérea descreveu no seu negócio o valor e horário para embarque de um voo de Bauru a Salvador, a partir dessas descrições foram criado os respectivos serviços descritos em WSDL, após a criação dos arquivos os mesmos foram publicados no servidor UDDI, em seguida para validação do estudo de caso o cliente fez requisições no servidor passando os parâmetros para a busca dos negócios obteve as respostas do servidor com êxito.

O servidor UDDI foi devidamente configurado seguindo os requisitos descritos no capítulo 6.2 e autenticado como mostrado no capítulo 6.3.

A seguir serão mostradas as mensagens XML que foram utilizadas no processo de publicação dos fornecedores e seus respectivos serviços, os quais são responsáveis pela visualização dos resultados obtidos desse protótipo.

#### **7.1.1 Descrição XML das empresas fornecedoras**

Para que os fornecedores possam publicar seus serviços precisam, primeiramente, serem registrados como uma entidade empresarial no servidor, para isso foi inserido no servidor UDDI a descrição de três fornecedores fictícios que tem como principal exercício a venda de passagens aéreas, cada fornecedor conta com seus dados de identificação e disponibilizando aos clientes opções de contato tais

como e-mail ou telefone, assim como uma descrição da empresa para facilitar o reconhecimento dos serviços prestados.

A primeira empresa publicada no servidor foi a Fornecedor A. A mensagem XML descreve a entidade e seus parâmetros como mostra na Figura 12.

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:uddi-org:api_v3">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <urn:save_business>
5       <urn:authInfo>authToken:0e22dd3e-19f8-4704-920c-baa56fb62cab</urn:authInfo>
6       <urn:businessEntity businessKey="uddi:juddi.apache.org:Fornecedor_A">
7         <urn:discoveryURLs>
8           <urn:discoveryURL useType="home">www.fornecedor.com.br</urn:discoveryURL>
9         </urn:discoveryURLs>
10        <urn:name xml:lang="en">Fornecedor A</urn:name>
11        <urn:description xml:lang="en">A fornecedora A é uma empresa séria de atuação mundial</urn:description>
12        <urn:contacts>
13          <urn:contact useType="Consultor">
14            <urn:description xml:lang="en">Consultor</urn:description>
15            <urn:personName xml:lang="en">Ivam</urn:personName>
16            <urn:phone useType="phone">(14) 3258-5212</urn:phone>
17            <urn:email useType="email">fornecedoraa@fornecedoraa.com.br</urn:email>
18          </urn:contact>
19        </urn:contacts>
20      </urn:businessEntity>
21    </urn:save_business>
22  </soapenv:Body>
23 </soapenv:Envelope>

```

Figura 12 - XML da empresa Fornecedor A  
Fonte: Elaborado pelo autor.

A Linha 5 insere o token de autorização para publicações no servidor. Na linha 6 o parâmetro passando no “businessKey” é a identificação única da empresa. Cada empresa possui sua identificação facilitando na publicação e segurança dos processos. A linha 8 fornece o endereço virtual da empresa. Na linha 10 é descrito o nome da empresa. A linha 11 descreve a entidade. Na linha 14 descreve o responsável direto pela empresa. A linha 15 consta o nome do responsável. A linha 16 informa o telefone de contato. Na linha 17 o email para contato.

Após a descrição da mensagem e publicação foi utilizado à ferramenta SoapUI no servidor e o mesmo devolveu a mensagem XML da publicação como mostrado na Figura 13.

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
2   <soap:Body>
3     <businessDetail xmlns:ns10="urn:uddi-org:vs_v3">
4       <businessEntity businessKey="uddi:juddi.apache.org:fornecedor_a">
5         <discoveryURLs>
6           <discoveryURL useType="home">www.fornecedora.com.br</discoveryURL>
7         </discoveryURLs>
8         <name xml:lang="en">Fornecedora A</name>
9         <description xml:lang="en">A fornecedora A é uma empresa séria de atuação mundial</description>
10        <contacts>
11          <contact useType="Consultor">
12            <description xml:lang="en">Consultor</description>
13            <personName xml:lang="en">Ivam</personName>
14            <phone useType="phone">(14) 3258-5212</phone>
15            <email useType="email">fornecedoraa@fornecedoraa.com.br</email>
16          </contact>
17        </contacts>
18      </businessEntity>
19    </businessDetail>
20  </soap:Body>
21 </soap:Envelope>

```

Figura 13 - Resposta XML da publicação da Fornecedora A  
Fonte: Elaborado pelo autor.

Na linha 4 o parâmetro passando no “businessKey” mostra que a identificação foi aceita. A linha 6 mostra o endereço virtual da empresa. Na linha 8 mostra o nome da empresa. A linha 9 mostra a descrição da entidade. Na linha 12 mostra o responsável direto pela empresa. A linha 13 consta o nome do responsável. A linha 14 mostra o telefone de contato. Na linha 15 o email para contato.

O servidor não retornou nenhuma mensagem de erro como resposta, assim mostrando que a publicação foi bem sucedida.

A segunda descrição XML publicada no servidor foi da empresa Fornecedora B como mostrado na Figura 14.

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:uddi-org:api_v3"
2   <soapenv:Header/>
3   <soapenv:Body>
4     <urn:save_business>
5       <urn:authInfo>authtoken:0e22dd3e-19f8-4704-920c-baa56fb62cab</urn:authInfo>
6       <urn:businessEntity businessKey="uddi:juddi.apache.org:fornecedor_b">
7         <urn:discoveryURLs>
8           <urn:discoveryURL useType="site">www.forb.com.br</urn:discoveryURL>
9         </urn:discoveryURLs>
10        <urn:name xml:lang="en">Fornecedora B</urn:name>
11        <urn:description xml:lang="en">A fornecedora B é líder em passagens aéreas</urn:description>
12        <urn:contacts>
13          <urn:contact useType="Vendedor">
14            <urn:description xml:lang="en">Vendedor</urn:description>
15            <urn:personName xml:lang="en">Carolina Mendez</urn:personName>
16            <urn:phone useType="phone">(140) 3265-8522</urn:phone>
17            <urn:email useType="?">carolina@fornecedorb.com.br</urn:email>
18          </urn:contact>
19        </urn:contacts>
20      </urn:businessEntity>
21    </urn:save_business>
22  </soapenv:Body>
23 </soapenv:Envelope>

```

Figura 14 - XML da empresa Fornecedora B  
Fonte: Elaborado pelo autor.

A Linha 5 insere o token de autorização para publicações no servidor. Na linha 6 o parâmetro passando no “businessKey” é a identificação única da empresa. A linha 8 fornece o endereço virtual da empresa. Na linha 10 é descrito o nome da empresa. A linha 11 descreve a entidade. Na linha 14 descreve o responsável direto pela empresa. A linha 15 consta o nome do responsável. A linha 16 informa o telefone de contato. Na linha 17 o email para contato.

Após a descrição da mensagem foi publicado através da ferramenta SopaUI no servidor e o mesmo devolveu a mensagem XML da publicação como mostrado na Figura 15.

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
2   <soap:Body>
3     <businessDetail xmlns:ns10="urn:uddi-org:vs_v3" xmlns:ns9="urn:uddi-org:policy_v3_instanceParms">
4       <businessEntity businessKey="uddi:juddi.apache.org:fornecedor_b">
5         <discoveryURLs>
6           <discoveryURL useType="site">www.forb.com.br</discoveryURL>
7         </discoveryURLs>
8         <name xml:lang="en">Fornecedora B</name>
9         <description xml:lang="en">A fornecedora B é lider em passagens aéreas</description>
10        <contacts>
11          <contact useType="Vendedor">
12            <description xml:lang="en">Vendedor</description>
13            <personName xml:lang="en">Carolina Mendez</personName>
14            <phone useType="phone">(140 3265-8522</phone>
15            <email useType="?">carolina@fornecedorb.com.br</email>
16          </contact>
17        </contacts>
18      </businessEntity>
19    </businessDetail>
20  </soap:Body>
21 </soap:Envelope>

```

Figura 15 - Resposta XML da publicação da Fornecedora B  
Fonte: Elaborado pelo autor.

Na linha 4 o parâmetro passando no “businessKey” mostra que a identificação foi aceita. A linha 6 mostra o endereço virtual da empresa. Na linha 8 mostra o nome da empresa. A linha 9 mostra a descrição da entidade. Na linha 12 mostra o responsável direto pela empresa. A linha 13 consta o nome do responsável. A linha 14 mostra o telefone de contato. Na linha 15 o email para contato.

Assim como na publicação da Fornecedora A a publicação da Fornecedora B no servidor não retornou nenhuma mensagem de erro como resposta, assim mostrando que a publicação foi bem sucedida.

A última descrição XML publicada no servidor foi a do Fornecedor C como mostrado na Figura 16.



```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
2   <soapenv:Header/>
3   <soapenv:Body>
4     <urn:save_business>
5       <urn:authInfo>authtoken:0e22dd3e-19f8-4704-920c-baa56fb62cab</urn:authInfo>
6       <urn:businessEntity businessKey="uddi:juddi.apache.org:fornecedor_c">
7         <urn:discoveryURLs>
8           <urn:discoveryURL useType="site">www.fornecedorac.com.br</urn:discoveryURL>
9         </urn:discoveryURLs>
10        <urn:name xml:lang="en">Fornecedora C</urn:name>
11        <urn:description xml:lang="en">A fornecedora C é uma ótima empresa</urn:description>
12        <urn:contacts>
13          <urn:contact useType="Vendedor">
14            <urn:description xml:lang="en">Vendedor</urn:description>
15            <urn:personName xml:lang="en">João Carlos</urn:personName>
16            <urn:phone useType="phone">(14) 99852-8544</urn:phone>
17            <urn:email useType="?">jcarlos@fornecedorc.com.br</urn:email>
18          </urn:contact>
19        </urn:contacts>
20      </urn:businessEntity>
21    </urn:save_business>
22  </soapenv:Body>
23 </soapenv:Envelope>

```

Figura 16 - XML da empresa Fornecedora C  
 Fonte: Elaborado pelo autor.

A Linha 5 insere o token de autorização para publicações no servidor. Na linha 6 o parâmetro passando no “businessKey” é a identificação única da empresa. A linha 8 fornece o endereço virtual da empresa. Na linha 10 é descrito o nome da empresa. A linha 11 descreve a entidade. Na linha 14 descreve o responsável direto pela empresa. A linha 15 consta o nome do responsável. A linha 16 informa o telefone de contato. Na linha 17 o email para contato.

Após a descrição da mensagem foi publicado através da ferramenta SoapUI no servidor e o mesmo devolveu a mensagem XML da publicação como mostrado na Figura 17.

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <businessDetail xmlns:ns10="urn:uddi-org:vs_v3" xmlns:ns9="urn:uddi-org:policy_v3_instanceParms">
4       <businessEntity businessKey="uddi:juddi.apache.org:fornecedor_c">
5         <discoveryURLs>
6           <discoveryURL useType="site">www.fornecedorac.com.br</discoveryURL>
7         </discoveryURLs>
8         <name xml:lang="en">Fornecedora C</name>
9         <description xml:lang="en">A fornecedora C é uma ótima empresa</description>
10        <contacts>
11          <contact useType="Vendedor">
12            <description xml:lang="en">Vendedor</description>
13            <personName xml:lang="en">João Carlos</personName>
14            <phone useType="phone">(14) 99852-8544</phone>
15            <email useType="?">jcarlos@fornecedorc.com.br</email>
16          </contact>
17        </contacts>
18      </businessEntity>
19    </businessDetail>
20  </soap:Body>
21 </soap:Envelope>

```

Figura 17 - Resposta XML da publicação da Fornecedora C  
Fonte: Elaborado pelo autor.

Na linha 4 o parâmetro passando no “businessKey” mostra que a identificação foi aceita. A linha 6 mostra o endereço virtual da empresa. Na linha 8 mostra o nome da empresa. A linha 9 mostra a descrição da entidade. Na linha 12 mostra o responsável direto pela empresa. A linha 13 consta o nome do responsável. A linha 14 mostra o telefone de contato. Na linha 15 o email para contato.

A publicação da Fornecedora C assim como as demais foi bem sucedida.

### 7.1.2 Descrição XML dos serviços

Após a publicação das empresas fornecedoras cada um pode realizar sua operação de inclusão dos serviços prestados, no servidor UDDI foram inclusos um serviço para cada empresa fornecedora.

A empresa Fornecedora A descreveu o seu serviço de passagem aérea com o detalhamento da data e hora do embarque de um voo de Bauru a Salvador como mostra a Figura 18.

```

1  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
2  <soap:Body>
3  <serviceDetail xmlns:ns10="urn:uddi-org:vs_v3" xmlns:ns9="urn:uddi-org:policy_v3_instanceParms"
4  xmlns:ns8="urn:uddi-org:policy_v3" xmlns:ns7="urn:uddi-org:subr_v3" xmlns:ns6="urn:uddi-org:sub_v3"
5  xmlns:ns5="urn:uddi-org:repl_v3" xmlns:ns4="urn:uddi-org:vsacache_v3" xmlns:ns3="urn:uddi-org:custody_v3"
6  xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" xmlns="urn:uddi-org:api_v3">
7  <businessService businessKey="uddi:juddi.apache.org:fornecedor_a"
8  serviceKey="uddi:juddi.apache.org:9c710d82-5a72-4134-8b6b-e772e36205dc">
9  <name xml:lang="en">Passagem aérea de Bauru a Salvador com saída 08/12/2013 às 22:00</name>
10 <description xml:lang="en">Passagem aérea de Bauru a Salvador com saída 08/12/2013 às 22:00</description>
11 <bindingTemplates>
12 <bindingTemplate bindingKey="uddi:juddi.apache.org:4a3570f8-e509-461e-a273-66a78f916b0f">
13 <description xml:lang="en">Ligação entre o tModel A com o Serviço da Fornecedora A</description>
14 <accessPoint useType="endPoint">http://127.0.0.1:8080/axis2/services/someservice</accessPoint>
15 <tModelInstanceDetails>
16 <tModelInstanceInfo tModelKey="uddi:juddi.apache.org:1df7850b-42cd-44f2-b9a8-78b607e4a6ed"/>
17 </tModelInstanceDetails>
18 </bindingTemplate>
19 </bindingTemplates>
20 </businessService>
21 </serviceDetail>
22 </soap:Body>
23 </soap:Envelope>

```

Figura 18 - Publicação do serviço da Fornecedora A  
Fonte: Elaborado pelo autor.

Na linha 7 é inserido a qual empresa o serviço pertence passando no “businessKey” o valor de identificação da sua empresa. A linha 9 consta o serviço que será disponibilizado mostrando a data e hora do embarque. Na linha 10 vem descrito o serviço.

O serviço foi publicado no servidor UDDI através da ferramenta SoapUI não havendo nenhum erro.

A empresa Fornecedora B descreveu seu serviço de passagem aérea com a descrição da data do embarque e valor da passagem de um voo de Bauru a Salvador como mostra a Figura 19.

```

1  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
2  <soap:Body>
3  <serviceDetail xmlns:ns10="urn:uddi-org:vs_v3" xmlns:ns9="urn:uddi-org:policy_v3_instanceParms"
4  xmlns:ns8="urn:uddi-org:policy_v3" xmlns:ns7="urn:uddi-org:subr_v3"
5  xmlns:ns6="urn:uddi-org:sub_v3" xmlns:ns5="urn:uddi-org:repl_v3" xmlns:ns4="urn:uddi-org:vsacache_v3"
6  xmlns:ns3="urn:uddi-org:custody_v3" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" xmlns="urn:uddi-org:api_v3">
7  <businessService businessKey="uddi:juddi.apache.org:fornecedor_b"
8  serviceKey="uddi:juddi.apache.org:286d5ed5-cda5-4329-a563-5bfbfa5fbc6">
9  <name xml:lang="en">Passagem aérea de Bauru a Salvador com Saída 25/12/2013 Valor: 232,56</name>
10 <description xml:lang="en">Passagem aérea de Bauru a Salvador com Saída 25/12/2013 Valor: 232,56</description>
11 <bindingTemplates>
12 <bindingTemplate bindingKey="uddi:juddi.apache.org:7127b6cf-6f7e-4f03-aaca-98bf2d05ad67">
13 <description xml:lang="en">Ligação entre o tModel B com o Serviço da Fornecedora B</description>
14 <accessPoint useType="endPoint">http://127.0.0.1:8080/axis2/services/someservice</accessPoint>
15 <tModelInstanceDetails>
16 <tModelInstanceInfo tModelKey="uddi:juddi.apache.org:bd985d40-0418-4841-a8e6-5a9fe7a0f533"/>
17 </tModelInstanceDetails>
18 </bindingTemplate>
19 </bindingTemplates>
20 </businessService>
21 </serviceDetail>
22 </soap:Body>
23 </soap:Envelope>

```

Figura 19 - Publicação do serviço da Fornecedora B  
Fonte: Elaborado pelo autor.

Na linha 7 é inserido a qual empresa o serviço pertence passando no “businessKey” o valor de identificação da sua empresa. A linha 9 consta o serviço que será disponibilizado com a descrição da data do embarque e valor da passagem de um voo de Bauru a Salvador. Na linha 10 vem descrito o serviço.

A empresa Fornecedora C descreveu o seu serviço de passagem aérea com o detalhamento de hora do embarque e o valor da passagem de um voo de Bauru a Salvador como mostra a Figura 20.

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:uddi-org:api_v3"
2   xmlns:xd="http://www.w3.org/2000/09/xmldsig#"
3   <soapenv:Header/>
4   <soapenv:Body>
5     <urn:save_service>
6       <urn:authInfo>authtoken:0e22dd3e-19f8-4704-920c-baa56fb62cab</urn:authInfo>
7       <urn:businessService businessKey="uddi:juddi.apache.org:fornecedora_c"
8         serviceKey="uddi:juddi.apache.org:f4469201-c711-4b9a-b1d0-59c6d3f11822">
9         <urn:name xml:lang="en">Passagem aérea de Bauru a Salvador com Saida às 22:00 Valor: 232,56</urn:name>
10        <urn:description xml:lang="en">Passagem aérea de Bauru a Salvador com Saida às 22:00 Valor: 232,56</urn:description>
11        <urn:bindingTemplates>
12          <urn:bindingTemplate>
13            <urn:description xml:lang="en">Ligação entre o tModel C com o Serviço da Fornecedora C</urn:description>
14            <urn:accessPoint useType="endPoint">http://127.0.0.1:8080/axis2/services/someservice</urn:accessPoint>
15            <urn:tModelInstanceDetails>
16              <urn:tModelInstanceInfo tModelKey="uddi:juddi.apache.org:25264b68-8289-4842-9754-d4dd022b4951"/>
17            </urn:tModelInstanceDetails>
18          </urn:bindingTemplate>
19        </urn:bindingTemplates>
20      </urn:businessService>
21    </urn:save_service>
22  </soapenv:Body>
23 </soapenv:Envelope>

```

Figura 20 - Publicação do serviço da Fornecedora C  
Fonte: Elaborado pelo autor.

Na linha 7 é inserido a qual empresa o serviço pertence passando no “businessKey” o valor de identificação da sua empresa. A linha 9 consta o serviço que será disponibilizado com descrição da hora do embarque e o valor da passagem. Na linha 10 vem descrito o serviço.

## 7.2 DESCOBERTA DE SERVIÇOS WEB

Com todos os fornecedores devidamente autenticados e seus respectivos serviços descritos e publicados no protótipo de servidor UDDI os mesmo estão aptos a serem descobertos, para tal finalidade e interatividade visual com o servidor o software SoapUI foi utilizado para que o cliente fictício faça as pesquisas.

O cliente fictício que fará as requisições e os fornecedores que disponibilizaram os serviços pode estar em máquinas independente da qual está

alocado o protótipo, acompanhando a ideia central do repositório de serviços que é realizar consultas ao servidor onde quer que o consumidor se encontre independente de plataforma, Para facilitar o desenvolvimento a máquina utilizada pelo fornecedor e cliente é a mesma onde está alocado o servidor UDDI.

Para realizar as pesquisas o cliente não precisa de nenhuma autenticação ressaltando que o mesmo só pode realizar pesquisas no servidor.

A primeira pesquisa realizada por parte do cliente fictício foi realizada no servidor como mostra a Figura 21.

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:uddi-org:api_v3">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <urn:find_business>
5       <urn:findQualifiers>
6         <!--1 or more repetitions:-->
7         <urn:findQualifier>approximatematch</urn:findQualifier>
8       </urn:findQualifiers>
9       <!--Zero or more repetitions:-->
10      <urn:name>$Fornecedora A$</urn:name>
11    </urn:find_business>
12  </soapenv:Body>
13 </soapenv:Envelope>

```

Figura 21 - Pesquisa da empresa Fornecedora A  
Fonte: Elaborado pelo autor.

Na linha 7 foi passado o tipo de pesquisa a ser realizada. O `approximatematch` é um parâmetro que se aplica a um registro UDDI, o que significa correspondência aproximada, onde sinal de porcentagem (%) indica qualquer número de caracteres ou um conjunto de caracteres encontrados em qualquer parte da frase. Para a primeira pesquisa no servidor foi descrito no parâmetro “Fornecedora A” como mostra na linha 10.

A pesquisa devolveu por parte do repositório os resultados descritos na Figura 22.

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <businessList xmlns:ns10="urn:uddi-org:vs_v3" xmlns:ns9="urn:uddi-org:policy_v3_instanceParms"
4       xmlns:ns8="urn:uddi-org:policy_v3" xmlns:ns7="urn:uddi-org:subr_v3" xmlns:ns6="urn:uddi-org:sub_v3"
5       xmlns:ns5="urn:uddi-org:repl_v3" xmlns:ns4="urn:uddi-org:vscache_v3" xmlns:ns3="urn:uddi-org:custody_v3"
6       xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" xmlns="urn:uddi-org:api_v3">
7       <listDescription>
8         <includeCount>1</includeCount>
9         <actualCount>1</actualCount>
10        <listHead>1</listHead>
11      </listDescription>
12      <businessInfos>
13        <businessInfo businessKey="uddi:juddi.apache.org:fornecedor_a">
14          <name xml:lang="en">Fornecedora A</name>
15          <description xml:lang="en">A fornecedora A é uma empresa séria de atuação mundial</description>
16          <serviceInfos>
17            <serviceInfo businessKey="uddi:juddi.apache.org:fornecedor_a"
18              serviceKey="uddi:juddi.apache.org:9c710d82-5a72-4134-8b6b-e772e36205dc">
19              <name xml:lang="en">Passagem aérea de Bauru a Salvador com saída 08/12/2013 às 22:00</name>
20            </serviceInfo>
21          </serviceInfos>
22        </businessInfo>
23      </businessInfos>
24    </businessList>
25  </soap:Body>
26 </soap:Envelope>

```

Figura 22 - Resultado da pesquisa da Fornecedora A  
Fonte: Elaborado pelo autor.

A linha 8 informa que fora encontrado apenas 1 registro no servidor segundo o parâmetro de pesquisa utilizado, na linha 9 consta a quantidade de resultados de empresas ativas encontradas, e na linha 10 consta a quantidade de páginas listadas.

A linha 13 mostra no “businessKey” qual é a chave de identificação única da empresa encontrada. A linha 14 informa o nome da empresa. Na linha 15 vem à descrição da empresa. A linha 17 mostra que existe um serviço ligado à empresa e dentro do parâmetro “businessKey” informa a identificação da empresa a qual pertence o serviço. A linha 18 mostra o “serviceKey” que é a chave única de identificação do serviço. A linha 19 descreve o serviço oferecido pela empresa.

A pesquisa realizada mostrou-se satisfatória retornando justamente a empresa requisitada.

Realizando uma nova pesquisa utilizando como parâmetro apenas a palavra “Fornecedora” como demonstra a Figura 23.

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:uddi-org:api_v3">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <urn:find_business>
5       <urn:findQualifiers>
6         <urn:findQualifier>approximatematch</urn:findQualifier>
7       </urn:findQualifiers>
8       <urn:name>%Fornecedora%</urn:name>
9     </urn:find_business>
10  </soapenv:Body>
11 </soapenv:Envelope>

```

Figura 23 - Pesquisa de todas as empresas fornecedoras

Fonte: Elaborado pelo autor.

Na linha 6 foi passado o tipo de pesquisa a ser realizada no servidor. O parâmetro da pesquisa na linha 8 foi a palavra “Fornecedora” estando entre o sinal de porcentagem assim fazendo com que a pesquisa retorne qualquer resultado que conste a palavra independente de onde na frase ela se encontre.

A pesquisa retornou os resultados mostrados na Figura 24.

Na linha 5 informa a quantidade de resultados encontrados, na linha 6 mostra a quantidade de resultados ativos encontrados e na linha 7 mostra a quantidade de páginas geradas pela pesquisa.

A linha 10 mostra qual é a chave de identificação unida da primeira empresa encontrada. Na linha 11 mostra o nome da empresa. A linha 12 informa a descrição da empresa. A linha 14 mostra que existe um serviço ligado à empresa e dentro do parâmetro “businessKey” informa a identificação da empresa a qual pertence o serviço. A linha 15 mostra o “serviceKey” que é a chave única de identificação do serviço. A linha 16 descreve o serviço oferecido pela empresa.

A linha 20 mostra a segunda empresa encontrada e sua chave de identificação unida. Na linha 21 mostra o nome da empresa. A linha 22 informa a descrição da empresa. A linha 24 mostra que existe um serviço publicado pela segunda empresa fornecedora encontrada e dentro do parâmetro “businessKey” informa a identificação da empresa que pertence o serviço. A linha 25 mostra o “serviceKey” que é a chave única de identificação do serviço. A linha 26 descreve o serviço oferecido pela empresa.

A linha 30 mostra a terceira empresa encontrada e sua chave de identificação unida. Na linha 31 mostra o nome da empresa. A linha 32 informa a descrição da empresa. A linha 34 mostra que existe um serviço publicado pela terceira empresa fornecedora encontrada e dentro do parâmetro “businessKey” informa a identificação



da empresa que pertence o serviço. A linha 35 mostra o “serviceKey” que é a chave única de identificação do serviço. A linha 36 descreve o serviço oferecido pela empresa.

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <businesslist xmlns:ns10="urn:uddi-org:vs_v3">
4       <listDescription>
5         <includeCount>3</includeCount>
6         <actualCount>3</actualCount>
7         <listHead>1</listHead>
8       </listDescription>
9     <businessInfos>
10      <businessInfo businessKey="uddi:juddi.apache.org:forneceador_a">
11        <name xml:lang="en">Fornecedora A</name>
12        <description xml:lang="en">A fornecedora A é uma empresa séria de atuação mundial</description>
13        <serviceInfos>
14          <serviceInfo businessKey="uddi:juddi.apache.org:forneceador_a">
15            <serviceKey="uddi:juddi.apache.org:9c710d82-5a72-4134-8b6b-e772e36205dc">
16              <name xml:lang="en">Passagem aérea de Bauru a Salvador com saída 08/12/2013 às 22:00</name>
17            </serviceInfo>
18          </serviceInfos>
19        </businessInfo>
20      <businessInfo businessKey="uddi:juddi.apache.org:forneceador_b">
21        <name xml:lang="en">Fornecedora B</name>
22        <description xml:lang="en">A fornecedora B é líder em passagens aéreas</description>
23        <serviceInfos>
24          <serviceInfo businessKey="uddi:juddi.apache.org:forneceador_b">
25            <serviceKey="uddi:juddi.apache.org:a6544b7a-46e1-47fc-96a6-9e8bc783b3f7">
26              <name xml:lang="en">Passagem aérea de Bauru a Salvador com Saída 25/12/2013 Valor: 232,56</name>
27            </serviceInfo>
28          </serviceInfos>
29        </businessInfo>
30      <businessInfo businessKey="uddi:juddi.apache.org:forneceador_c">
31        <name xml:lang="en">Fornecedora C</name>
32        <description xml:lang="en">A fornecedora C é uma ótima empresa</description>
33        <serviceInfos>
34          <serviceInfo businessKey="uddi:juddi.apache.org:forneceador_c">
35            <serviceKey="uddi:juddi.apache.org:f4469201-c711-4b9a-b1d0-59c6d3f11822">
36              <name xml:lang="en">Passagem aérea de Bauru a Salvador com Saída às 22:00 Valor: 232,56</name>
37            </serviceInfo>
38          </serviceInfos>
39        </businessInfo>
40      </businessInfos>
41    </businessList>
42  </soap:Body>
43 </soap:Envelope>

```

Figura 24 - Resultado da pesquisa de todos os fornecedores  
Fonte: Elaborado pelo autor.

A pesquisa realizada mostrou-se satisfatória retornando todas as empresas que conste no nome a palavra “Fornecedora”.

Agora as busca foram focadas nos serviços oferecidos pelos fornecedores,

O cliente utilizou como parâmetro para a pesquisa a data de embarque e o segundo parâmetro o local de destino da passagem como mostra a Figura 25.



```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:uddi-org:api_v3">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <urn:find_service>
5       <urn:findQualifiers>
6         <urn:findQualifier>approximateMatch</urn:findQualifier>
7       </urn:findQualifiers>
8       <urn:name xml:lang="en">%08/12/2013%</urn:name>
9       <urn:name xml:lang="en">%Salvador%</urn:name>
10    </urn:find_service>
11  </soapenv:Body>
12 </soapenv:Envelope>

```

Figura 25 - Pesquisa por data e local  
Fonte: Elaborado pelo autor.

Na linha 6 foi passado o tipo de pesquisa a ser realizada no servidor. A linha 8 consta a data de interesse do cliente para a viagem e na linha 9 o segundo parâmetro de qual o destino de interesse. O resultado da pesquisa é mostrado na Figura 26.

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <servicelist xmlns:ns10="urn:uddi-org:vs_v3">
4       <listDescription>
5         <includeCount>3</includeCount>
6         <actualCount>3</actualCount>
7         <listHead>1</listHead>
8       </listDescription>
9       <serviceInfos>
10        <serviceInfo businessKey="uddi:juddi.apache.org:forneecedor_a"
11          serviceKey="uddi:juddi.apache.org:9c710d82-5a72-4134-8b6b-e772e36205dc">
12          <name xml:lang="en">Passagem aérea de Bauru a Salvador com saída 08/12/2013 às 22:00</name>
13        </serviceInfo>
14        <serviceInfo businessKey="uddi:juddi.apache.org:forneecedor_b"
15          serviceKey="uddi:juddi.apache.org:a6544b7a-46e1-47fc-96a6-9e8bc783b3f7">
16          <name xml:lang="en">Passagem aérea de Bauru a Salvador com Saída 25/12/2013 Valor: 232,56</name>
17        </serviceInfo>
18        <serviceInfo businessKey="uddi:juddi.apache.org:forneecedor_c"
19          serviceKey="uddi:juddi.apache.org:f4469201-c711-4b9a-b1d0-59c6d3f11822">
20          <name xml:lang="en">Passagem aérea de Bauru a Salvador com Saída às 22:00 Valor: 232,56</name>
21        </serviceInfo>
22      </serviceInfos>
23    </servicelist>
24  </soap:Body>
25 </soap:Envelope>

```

Figura 26 - Resultado da pesquisa por data e local  
Fonte: Elaborado pelo autor.

A linha 5 informa a quantidade de serviços encontrados, na linha 6 mostra a quantidade de serviços ativos encontrados e na linha 7 mostra a quantidade de páginas geradas pela pesquisa.

A linha 10 mostra o primeiro serviço encontrado e dentro do parâmetro “businessKey” informa a identificação da empresa que pertence o serviço. O serviço

é pertencente à empresa Fornecedora A. Na linha 11 mostra o “serviceKey” que é a chave única de identificação do serviço. Na linha 12 descreve o serviço.

A linha 14 mostra o segundo serviço encontrado e dentro do parâmetro “businessKey” informa a identificação da empresa que pertence o serviço. . O serviço é pertencente à empresa Fornecedora B. Na linha 15 mostra o “serviceKey” que é a chave única de identificação do serviço. Na linha 16 descreve o serviço.

A linha 18 mostra o terceiro serviço encontrado e dentro do parâmetro “businessKey” informa a identificação da empresa que pertence o serviço. O serviço é pertencente à empresa Fornecedora C. Na linha 19 mostra o “serviceKey” que é a chave única de identificação do serviço. Na linha 20 descreve o serviço.

O resultado trazido pelo servidor retornou todos os serviços publicados devido aos parâmetros utilizados, apenas um serviço disponibiliza a passagem para o dia 08/12/2013 entretanto todos os serviços têm como descrição como destino Salvador.

A pesquisa realizada na Figura 27 mostra que o único parâmetro utilizado foi o horário de embarque.

```

1 | <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:uddi-org:api_v3">
2 |   <soapenv:Header/>
3 |   <soapenv:Body>
4 |     <urn:find_service>
5 |       <urn:findQualifiers>
6 |         <urn:findQualifier>approximateMatch</urn:findQualifier>
7 |       </urn:findQualifiers>
8 |       <urn:name xml:lang="en"%22:00%</urn:name>
9 |     </urn:find_service>
10 |   </soapenv:Body>
11 | </soapenv:Envelope>

```

Figura 27 - Pesquisa por horário de embarque  
Fonte: Elaborado pelo autor.

Na linha 6 foi passado o tipo de pesquisa a ser realizada no servidor. A linha 8 consta a hora de embarque do voo. O resultado da pesquisa é mostrado na Figura 28.

```

1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <serviceList xmlns:ns10="urn:uddi-org:vs_v3">
4       <listDescription>
5         <includeCount>2</includeCount>
6         <actualCount>2</actualCount>
7         <listHead>1</listHead>
8       </listDescription>
9       <serviceInfos>
10        <serviceInfo businessKey="uddi:juddi.apache.org:fornecedor_a"
11          serviceKey="uddi:juddi.apache.org:9c710d82-5a72-4134-8b6b-e772e36205dc">
12          <name xml:lang="en">Passagem aérea de Bauru a Salvador com saída 08/12/2013 às 22:00</name>
13        </serviceInfo>
14        <serviceInfo businessKey="uddi:juddi.apache.org:fornecedor_c"
15          serviceKey="uddi:juddi.apache.org:f4469201-c711-4b9a-b1d0-59c6d3f11822">
16          <name xml:lang="en">Passagem aérea de Bauru a Salvador com Saída às 22:00 Valor: 232,56</name>
17        </serviceInfo>
18      </serviceInfos>
19    </serviceList>
20  </soap:Body>
21 </soap:Envelope>

```

Figura 28 - Resultado da pesquisa por horário de embarque  
Fonte: Elaborado pelo autor.

A linha 5 informa a quantidade de serviços encontrados, na linha 6 mostra a quantidade de serviços ativos encontrados e na linha 7 mostra a quantidade de páginas geradas pela pesquisa.

A linha 10 mostra o primeiro serviço encontrado e dentro do parâmetro “businessKey” informa a identificação da empresa que pertence o serviço. O serviço é pertencente à empresa Fornecedora A. Na linha 11 mostra o “serviceKey” que é a chave única de identificação do serviço. Na linha 12 descreve o serviço.

A linha 14 mostra o segundo serviço encontrado e dentro do parâmetro “businessKey” informa a identificação da empresa que pertence o serviço. O serviço é pertencente à empresa Fornecedora C. Na linha 15 mostra o “serviceKey” que é a chave única de identificação do serviço. Na linha 16 descreve o serviço.

Diferente da pesquisa realizada por data e destino da passagem a pesquisa por horário de embarque trouxe somente a empresa Fornecedora A e Fornecedora C que são as únicas que descreveram em seus serviços o horário de saída do voo.

O protótipo do servidor UDDI, demonstrou todo ser capaz de armazenar e retornar os serviços solicitados.

## 8 CONSIDERAÇÕES FINAIS

O presente estudo relatou o funcionamento de uma parte da composição de uma Arquitetura Orientada a Serviços, dentre os pontos relatados o que obteve maior ênfase foi a estrutura que cerca um servidor de repositórios UDDI que é o objeto central do trabalho.

O projeto apresentado demonstra o grande potencial que os serviços Web têm a oferecer tanto para quem os fornece quando aos que os consome, a oportunidade de crescimento e ganho de mercado para empresas que não contam com recursos financeiros para competir com empresas de grande porte é enorme, sendo que a simples publicação de seus serviços pode alcançar um público imenso que cada dia mais recorre à internet para realizar compras. Aos consumidores traz a facilidade de pesquisar em um único repositório todos os serviços facilitando buscas e cotação de preços, assim trazendo economia e aumentando a competitividade entre as companhias possibilitando a queda dos preços, além da comodidade e rapidez.

Para tal pesquisa foi utilizado todo o levantamento teórico que demonstra quais os requisitos necessários para o desenvolvimento do protótipo. O trabalho teve como resultado a publicação e descoberta de serviços utilizando servidores UDDI obtidos com êxito.

### 8.1 TRABALHOS FUTUROS

Várias diretrizes de pesquisa podem ser delineadas a partir dos resultados obtidos nesse estudo.

A primeira diretriz aponta para elaboração de uma pesquisa mais eficiente por parte do servidor, a implementação de técnicas de inteligência artificial proporcionaria uma melhor eficiência nos resultados obtidos por parte do servidor.

Outra diretriz importante é uma melhoria na indexação dos dados, proporcionando uma agilidade nas respostas dos serviços.

Outro trabalho relacionado a este, seria o desenvolvimento de uma ferramenta gráfica que proporcione uma maior facilidade de entendimento dos dados pesquisados no servidor.

## REFERÊNCIAS

- ALBINADER Neto, JORGE Abilio. **Web Services em Java**. Rio de Janeiro, Brasport, 2006.
- ALONSO G., CASATI F., KUNO H., MACHIRAJU V. **Web Services: Concepts, Architectures and Applications**. Springer-Verlag, Berlin, Germany, October 2004.
- Apache jUDDI, **jUDDI Versão 3.0.4**, Datado em: 10 de Fevereiro de 2009, Disponível em: <<http://juddi.apache.org/>> Último acesso em: 18 de Novembro de 2013.
- Apache Tomcat, **Tomcat 6.9**, Disponível em: <<http://tomcat.apache.org/>> Último acesso em: 18 de Novembro de 2013.
- CHINNICI R., MOREAU J., RYMAN A., WEERAWARANA S. 2006. **Web Services Description Language (WSDL)** version 2.0 part 1: Core language. Disponível em: <<http://www.w3.org/TR/wsdl20/>> Acesso em: 13 de Maio 2013.
- COSTA, D. G. **Java em rede: recursos avançados de programação**. Brasport, 2008.
- CUNHA, D. **Web Services, SOAP e Aplicações Web**. Disponível em: <[http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index\\_pt\\_br.html](http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index_pt_br.html)> Publicado em: 10 Dez 2002. Revisado em: 10 Dez 2002. Último acesso em: 02 de Junho de 2013.
- DAVID, A. W. **Ubuntu Manpage Repository**. 2010. Disponível em: <<http://manpages.ubuntu.com/manpages/precise/pt/man7/uri.7.html>> Traduzido por: NOGUEIRA, R. J. Revisado por: CANOVA, A. L. F. Último acesso em: 02 de Junho de 2013.
- DE SORDI, J. O. **Gestão por processos: uma abordagem da moderna administração**. São Paulo: Saraiva 2005.
- ENOKI, C. H. **Gestão de processos de negócio: uma contribuição para a avaliação de soluções de business process management (BPM) sob a ótica da estratégia de operações**. 2006. Dissertação (Mestrado em Engenharia de Produção) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2006. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3136/tde-01122006-170526/>>. Último acesso em: 13 de Maio de 2013.
- FANTINATO M. 2007. **Uma Abordagem Baseada em Características para o Estabelecimento de Contratos Eletrônicos para Serviços Web**. Tese de Doutorado. Universidade Estadual de Campinas (UNICAMP), Instituto de Computação (IC), Campinas, SP.
- GIL, A. C., **Como elaborar projetos de pesquisa**. 4. ed., São Paulo: Atlas, 2002.

GROSSI, B. E. 2005. **Estudo do modelo de computação orientada a serviços e sua aplicação a um sistema de mineração de dados**, Tese de Pós-Graduação. Universidade Federal de Minas Gerais (UFMG), Instituto de Ciências Exatas (ICEX), Belo Horizonte, MG. Disponível em: <<http://hdl.handle.net/1843/RVMR-6EAFWY>> Último acesso em: 03 de Junho de 2013

Microsoft, **Windows Server 2008 R2**, Disponível em: <<http://www.microsoft.com/pt-br/default.aspx>> Último acesso em: 18 de Novembro de 2013.

MITRA N., Y. L. **SOAP Version 1.2 Part 0: Primer (Second Edition)**. W3C Recommendation 27 April 2007. Disponível em: <<http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>> Último acesso: 22 de maio 2013.

MOSSI, W. **BPM – Modelagem de Processos de Negócios**. 2009. Disponível em: <<http://www.en-sof.com.br/downloads/BPM-modelagem-processos.pdf>>, Último acesso em: 13 de Maio de 2013.

MySQL, **MySQL 5.1**, Disponível em: <<http://dev.mysql.com/doc/refman/5.5/en/>> Último acesso em: 18 de Novembro de 2013.

NETTO, F. S. **Gerenciamento de Processos de Negócio: um estudo teórico-comparativo sob as óticas da Gestão Empresarial e da Tecnologia da Informação**. 2008. Disponível em: <[http://www.aedb.br/seget/artigos08/254\\_254\\_SEGeT\\_BPM\\_2008\\_final.pdf](http://www.aedb.br/seget/artigos08/254_254_SEGeT_BPM_2008_final.pdf)>. Último acesso em: 08 de maio de 2013

OASIS, **Serviços Business Web Processo de Execução Idioma Versão 2**. Datado em: 27 de Abril de 2007. Disponível em: <<http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>> Último acesso em: 15 de maio 2013.

OASIS, **UDDI Versão 3.0.2**, Datado em: 19 de Setembro de 2004, Disponível em: <[http://uddi.org/pubs/uddi-v3.0.2-20041019.htm#\\_Toc85907967](http://uddi.org/pubs/uddi-v3.0.2-20041019.htm#_Toc85907967)> Último acesso em: 13 de Maio de 2013.

OLIVEIRA, P. A; JUNIOR C. A. D; OLIVEIRA P. F. A. 2008. **Proposição de Infra-Estrutura de Dados Espaciais (SDI) Local, Baseada em Arquitetura Orientada por Serviços (SOA)** Disponível em: <<http://homepages.dcc.ufmg.br/~clodoveu/files/100.40/AC044.%202008%20Proposicao%20de%20infra-estrutura%20de%20dados%20espaciais%20local%20baseada%20em%20arquitetura%20orientada%20por%20servicos.pdf>> Último acesso em: 02 de Junho de 2013.

Oracle, **Java JDK 1.5.0**, Disponível em: <<http://www.oracle.com/index.html>> Último acesso em: 18 de Novembro de 2013.

RECKZIEGEL, M. 2006. **Descrevendo, descobrindo e Integrando Web Services**. Disponível em: <[http://www.imasters.com.br/artigo/4474/webservices/descrevendo\\_descobrimdo\\_e\\_integrando\\_web\\_services\\_-\\_uddi/](http://www.imasters.com.br/artigo/4474/webservices/descrevendo_descobrimdo_e_integrando_web_services_-_uddi/)>. Último acesso em: 03 de Agosto de 2007.

REIS, J. C. dos; CÉLIO, M. V. de P. **SOA – UDDI. 2007**. Disponível em: <[http://www.sensedia.com/br/anexos/wp\\_soauddi.pdf](http://www.sensedia.com/br/anexos/wp_soauddi.pdf)> Último acesso em: 20 de Maio de 2013.

RODRIGUES, L. C; MACCARI, E. A; RISCAROLLI, V. **Arquitetura e coopetição em redes interorganizacionais**. JISTEM J.Inf.Syst. Technol. Manag. (Online), São Paulo, v. 4, n. 2, 2007. Disponível em <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1807-17752007000200004&lng=pt&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1807-17752007000200004&lng=pt&nrm=iso)>. Último acesso em 20 maio 2013. <http://dx.doi.org/10.1590/S1807-17752007000200004>.

SAMAPIO, C. **SOA e Web Services em Java**. Rio de Janeiro: Brasport, 2006.

SANTOS, L. L. 2011. **Monitoramento de contratos eletrônicos baseados em características**. Dissertação de Mestrado. Universidade Estadual de Campinas (UNICAMP), Instituto de Computação (IC), Campinas, SP.

SoapUI, **SoapUI 4**, Disponível em: <<http://www.soapui.org/>> Último acesso em: 18 de Novembro de 2013.

SPANYI, A. **Business Process Management is a Team Sport**. Tampa: Anclote Press, 2003. VENKATRAMAN, N. IT induced business reconfiguration. In: VENKATRAMAN, N. The corporation of the 1990s. New York: Oxford University Press.

STRELESKI, J. R; HISAMITSU, C. G. de O. 2009. **SOA – ARQUITETURA ORIENTADA A SERVIÇOS**. Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP). Bauru. SP

TIZZO, N. P. 2010 – **Arquitetura Orientada a Serviços: A Evolução de Estruturas Complexas a partir de Estruturas Simples**. PUC, Minas; Disponível em: <[http://www.ifsc.usp.br/~9geinfo/images/stories/palestras/soa\\_neil.pdf](http://www.ifsc.usp.br/~9geinfo/images/stories/palestras/soa_neil.pdf)> Último acesso em: 10 mai. 2013.

TODD N. SZOLKOWSKI M. **JavaServer pages: o guia do desenvolvedor** : tradução de FURMANKIEWICZ E. Rio de Janeiro. 2003; 3<sup>o</sup> impressão.

URI, disponível em:

<<http://www.nce.ufrj.br/ginape/cursoshtml/conteudo/ligacoes/uri.htm>> Último acesso em: 13 de Maio de 2013.

VIEIRA, O. FILHO, S. **Microsoft.NET: Uma visão para programação**. São Paulo: Editora SENAC São Paulo, 2002.

WESKE, M. 2007. **Business Process Management: Concepts, Languages, Architectures**. Springer Verlag, first edition, November.



## APÊNDICE A – ARTIGO DE NEGÓCIOS ELETRÔNICOS E SERVIÇOS WEB NEGÓCIOS ELETRÔNICOS E SERVIÇOS WEB

**Aline Valezi; Prof. Esp. Henrique Pachioni Martins;  
Prof. Dr. Elvio Gilberto da Silva; Prof.<sup>a</sup> M.<sup>a</sup> Patrícia Bellin Ribeiro**

Programadora de Sistemas de Informação na Prefeitura Municipal de Bauru, estudante matriculada no curso de Ciência da Computação na Universidade Sagrado Coração, USC, São Paulo, SP, Brasil.

e-mail: alinevalezi@gmail.com

**Abstract.** *The increasing consumption of services through the virtual world has considerably increased some tools and techniques are providing greater ease and security to the achievement of the negotiations. The Service Oriented Architecture (SOA) is an abstract concept that standardizes the technical publishing and consumption of Web services System of Business Process Management (SGPN) Management uses tools and methods for the management of electronic contracts and finally the UDDI server (Universal Description, Discovery, and Integration) is responsible for the storage and retrieval of published services. Through this server you can request a service without the need for the client to have technical knowledge, thus returning the deal that meets the requirements specified by the consumer, with the great advantage of platform independence.*

**Resumo.** *O crescente consumo de serviços através do mundo virtual tem aumentando consideravelmente, algumas ferramentas e técnicas estão proporcionando maior facilidade e segurança para a concretização das negociações. A Arquitetura Orientada a Serviços (AOS) é um conceito abstrato que padroniza as técnicas de publicação e consumo dos serviços Web. O Sistema de Gerenciamento de Processos de Negócios (SGPN) utiliza de ferramentas e métodos para a gestão dos contratos eletrônicos e, por fim, o servidor UDDI (Universal Description, Discovery, and Integration) é responsável pelo armazenamento e recuperação dos serviços publicados. Através desse servidor é possível solicitar um serviço sem a necessidade de o cliente ter conhecimento técnico, assim retornando o negócio que mais atende as necessidades especificadas pelo consumidor, tendo como grande vantagem à independência de plataforma.*

## 1. Introdução

A competição entre empresas é um fator importante para o público que consome seus serviços e produtos, pois estimula as companhias a realizarem constantes melhorias. Com essa forte disputa, as empresas estão expandindo seus métodos de negociação e o mundo virtual está desempenhando um forte papel nesse quesito. O novo ambiente empresarial é fundamentalmente baseado no ambiente digital, que tem como componente básico a Internet, considerada infraestrutura de comunicação pública de acesso fácil, livre e de baixo custo ALBERTIN (2000).

Percebendo esse grande potencial, as empresas vislumbram o grande alcance que a internet tem para negociações.

Na necessidade de uma composição que pudesse definir e auxiliar no funcionamento desses processos surgiu o paradigma de Computação Orientada a Serviços (COS), que veio para definir aplicações distribuídas (Papazoglou et al., 2003 citado por Santos, 2011). Para Santos (2011), esse paradigma utiliza serviços eletrônicos como elementos fundamentais para o desenvolvimento de aplicações distribuídas. Os papéis nesse paradigma são: fornecedores e consumidores de serviços.

O UDDI (Universal Description, Discovery, and Integration) é um servidor que tem a função de armazenar e disponibilizar os serviços publicados pelos fornecedores, encontrando dinamicamente o negócio que mais atende as necessidades do cliente. Diversos conceitos giram em torno dos negócios eletrônicos, tais como o conceito abstrato de Arquitetura Orientada a Serviços (AOS) e o Sistema de Gerenciamento de Processos de Negócios (SGPN) que têm a função de padronizar e organizar as operações entre cliente e fornecedor.

Este artigo analisa os conceitos que envolvem a criação, publicação e descoberta de um serviço eletrônico e os padrões empregados.

## 2. Desenvolvimento

A crescente popularização da internet facilita que empresas de pequeno e médio porte divulguem seus serviços, retornando um resultado positivo àquelas que buscam, através do mundo virtual, a sua consolidação. Segundo ALBERTIN (2000) esse novo ambiente tem fornecido para os agentes econômicos, tanto empresas quanto indivíduos, canais alternativos para trocar informações, comunicar-se, transferir diferentes tipos de produtos e serviços e iniciar transações comerciais.

Toda a negociação digital deve seguir requisitos mínimos para que haja resultados positivos nas negociações. Deve-se considerar a eficácia das transações, segurança nas informações e padronização de operações. Para tal objetivo surgiu a Arquitetura Orientada a Serviços (AOS) que tem seu conceito abstrato (OLIVEIRA et al., 2008) visando à padronização, criação e disposição de serviços na Web. GROSSI (2005) cita as principais características da AOS: componentes de softwares que publicam seus contratos (ou interface) independentes de plataforma, serviços interoperáveis sendo encontrados dinamicamente.

A ideia básica de criação, registro ou consumo de serviços é exemplificada na Figura 1 - Paradigma básico para Arquiteturas Orientadas a Serviços (AOS).

Observamos a entidade Produtor que cria e registra os seus serviços em um servidor de diretórios e a partir dessa publicação a entidade consumidora pode fazer buscas e localizar os serviços publicados dinamicamente, após a localização a entidade consumidora pode invocá-los.

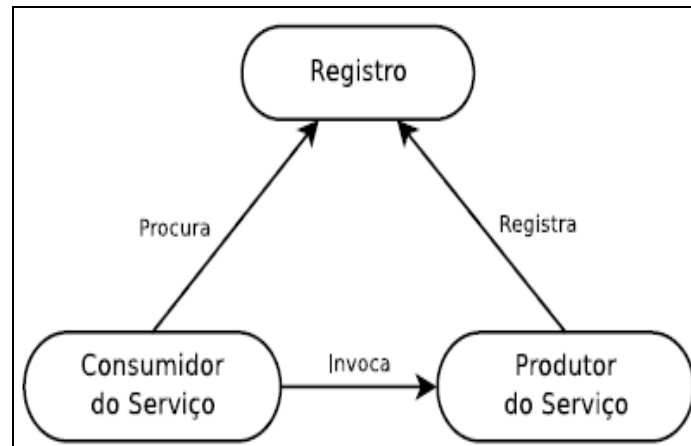


Figura 1 - Paradigma básico para Arquiteturas Orientadas a Serviços (SOA)  
Fonte: Grossi, 2005, p. 9

Um serviço pode ser encontrado e invocado mediante técnicas que utilizam padrões de interface descrita em WSDL (Web Services Description Language), registrados em um diretório de serviços via protocolo UDDI (Universal Description, Discovery, and Integration), e interagem com o cliente usando troca de mensagens XML (Extensible Markup Language) encapsuladas em envelopes SOAP (Simple Object Access Protocol) SANTOS (2011, p. 7).

Um serviço Web segue alguns processos que podem ser observados na Figura 2 - Ciclo de vida de um SW. Primeiramente (etapa 1) o provedor de serviço, depois de tê-lo descrito na linguagem XML e estando no padrão WSDL, publica-os no registro de serviços (diretório de servidor UDDI), assim esse serviço já está apto a ser descoberto. Em seguida (etapa 2), um cliente faz a solicitação de um serviço no servidor de registro. Posteriormente (etapa 3), o servidor de registro encontra dinamicamente o serviço que mais se adéqua as necessidades do cliente e devolve a resposta ao cliente de serviços mediante protocolo SOAP. O mesmo contata o provedor de serviços (etapa 4) e assim é feito um contrato eletrônico entre as partes (etapa 5).

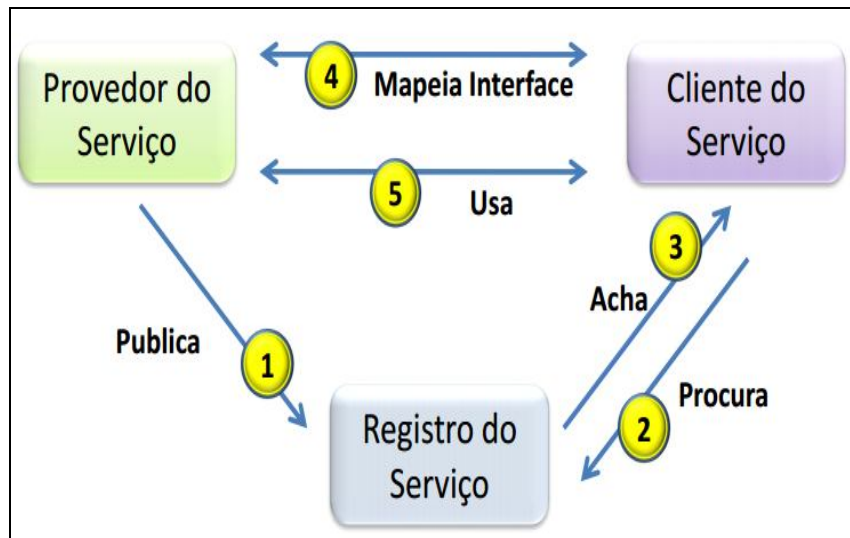


Figura 2 - Ciclo de vida de um SW  
 Fonte: Tizzo, 2010, p. 10

As interfaces devem seguir o padrão WSDL que descreve seus serviços em XML onde é detalhada a interface de um serviço Web. Um Arquivo WSDL possui as informações necessárias para que o serviço seja descrito, publicado e encontrado.

A interface possibilita a descrição de informações da interface de um serviço, tais como seus parâmetros de entrada e de saída, de modo que um usuário possa usá-lo sem necessariamente conhecer sua implementação. (FANTINATO, 2007).

A concepção de um serviço Web é uma estrutura modelada que permite a livre comunicação. Fantinato (2007) reforça essa ideia dizendo que a padronização dos serviços permite que eles sejam executados em qualquer plataforma de software e hardware. Assim, os aplicativos se tornam mais fáceis de serem disponibilizados e consumidos já que todos usam o mesmo padrão.

Principais padrões usados para a tecnologia de serviços Web:

- **WSDL (Web Services Description Language):** Um arquivo WSDL é um documento XML que descreve um conjunto de mensagens SOAP e como elas são retornadas (Vieira et al., 2002, p. 72). Incluindo capacidade, interface, comportamento e qualidade de serviço (Fantinato, 2007). Os arquivos têm padrões bem definidos tornando viável para descrever as interfaces Web Services (Vieira et al., 2002).
- **UDDI (Universal Description, Discovery, and Integration):** Um repositório para publicação e busca de serviços web padronizados e estruturados, são armazenadas informações sobre os serviços. Segundo Fantinato (2007) o padrão UDDI permite que provedores de serviços possam registrar seus serviços Web, usando as descrições WSDL, para que clientes de serviços possam descobri-los. A arquitetura dos serviços web segue o conceito de SOA (Service Oriented Architecture) – arquitetura orientada a serviços.
- **SOAP (Simple Object Access Protocol):** protocolo que define um mecanismo para a comunicação geral entre serviços Web na Internet.

Ele define o formato das mensagens que são trocadas entre clientes de serviços, provedores de serviços e diretórios de serviços (Mitra, 2010).

O arquivo precisa ser descrito por uma linguagem, preferencialmente interpretável por computador. O padrão atual para a especificação de negócios é a linguagem WS-BPEL (OASIS, 2010 citado por Santos 2011, p. 7). WS-BPEL é uma linguagem para especificar o comportamento de processos de negócios baseados em Web Services (OASIS, 2007).

OASIS (2007) descreve que a finalidade da linguagem WS-BPEL é:

Os processos de negócios podem ser descritos de duas maneiras. Negócio executável processa comportamento do modelo real de um participante de uma interação de negócios. Processos de negócios abstratos são processos parcialmente especificados que não se destinam a ser executados. Um processo abstrato pode esconder alguns dos requeridos operacionais. Processos abstratos têm um papel descritivo, com mais de um caso de uso possível, incluindo o comportamento observável e modelo de processo. WS-BPEL é feito para ser usado para modelar o comportamento de ambos executável e processos abstratos.

WS-BPEL oferece uma linguagem para a especificação de processos de negócios executáveis e abstratos. Ao fazer isso, ele estende o modelo de interação Web Services e permite que ele suporte a transações de negócios. WS-BPEL define um modelo de integração interoperável que deve facilitar a expansão do processo de integração automatizada, tanto no intra-empresas e os espaços de business-to-business.

Web Services é identificado por uma URI (Unique Resource Identifier), e são descritos e definidos usando XML (Cunha, 2002).

O Sistema de Gerenciamento de Processos de Negócios (SGPN) é um conceito muito importante para a manutenção dos negócios juntamente com a tecnologia da informação. O SGPN utiliza de métodos, técnicas e ferramentas para analisar, modelar, publicar, aprimorar e controlar processos envolvendo recursos humanos, aplicações, documentos e outras fontes de informação (Mossi, 2009).

De Sordi (2005) fala que o processo de negócio é um meio integrador de todos os ativos organizacionais. E o seu gerenciamento acontece quando estes ativos trabalham em sincronia, assegurando-lhes eficiência. Assim como um contrato convencional, o SGPN utiliza de contratos eletrônicos para definir as partes envolvidas e os serviços prestados.

O ciclo de vida de um contrato eletrônico consiste em alguns elementos citados a seguir e ilustrado na Figura 3 - Ciclo de vida dos contratos eletrônicos

**1. Estabelecimento de contratos eletrônicos:** processo de decisão que estabelece como o processo de negócio deverá ser realizado entre a organização fornecedora e a organização consumidora. Durante a negociação são definidas as partes envolvidas, os serviços a serem prestados e as cláusulas contratuais que devem ser cumpridas durante a realização do contrato. O processo de negociação pode ser realizado seguindo um protocolo em que os papéis e as responsabilidades são bem definidos por meio de atividades sistemáticas que cada organização envolvida deve seguir (SANTOS, 2011).

2. **Execução e monitoramento do contrato eletrônico:** execução e cumprimento dos termos estabelecidos no contrato eletrônico, por meio da execução dos serviços eletrônicos previstos e cumprimento das cláusulas contratuais estabelecidas. Para garantir o cumprimento das cláusulas, elas devem ser monitoradas durante a execução dos serviços eletrônicos. A organização consumidora também pode executar operações de monitoramento do processo conforme restrições estabelecidas no contrato eletrônico (SANTOS, 2011).

3. **Renegociação do contrato eletrônico:** caso algumas das cláusulas não sejam cumpridas, o contrato eletrônico pode ser renegociado para atender as novas condições de execução do processo (SANTOS, 2011).

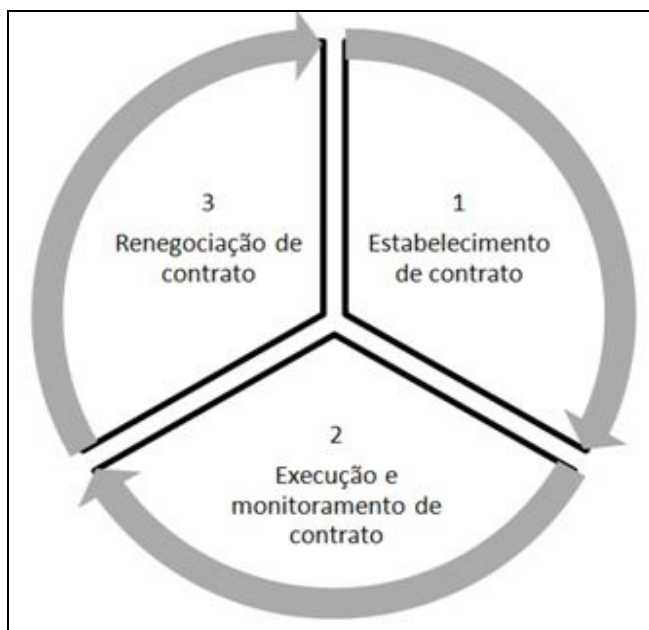


Figura 3 - Ciclo de vida dos contratos eletrônicos  
Fonte: Santos, 2011, p. 6.

A contratação de um serviço se inicia com a criação e publicação de um arquivo WSDL em um servidor UDDI. Esse servidor é o responsável por armazenar e selecionar um serviço mediante uma solicitação de um usuário.

O foco da UDDI é a definição do que deve e como ser feito, independentemente de plataformas, para que negócios possam descrever publicamente seus serviços disponíveis, localizar outros serviços do seu interesse e compartilhar informações sobre pontos convergentes em um registro global (ALBINADER et al., 2006).

Os serviços são registrados no servidor UDDI pelos próprios fornecedores de acordo com os padrões necessários; posterior publicação o serviço pode ser solicitado por um cliente, sendo o servidor o responsável por encontrar dinamicamente o que mais se aproxime das solicitações.

## Considerações Finais

Este artigo observou as características envolvidas em um negócio eletrônico abordando a funcionalidade de uma composição baseada na Arquitetura Orientada a Serviços, como os serviços Web, que são essenciais para um contrato eletrônico, são descritos, publicados e invocados, e os padrões que são empregados demonstrando o grande potencial que envolve os negócios *on-line*.

## Referências

ALBERTIN, Alberto Luiz. O comércio eletrônico evolui e consolida-se no mercado brasileiro. Rev. adm. empres., São Paulo, v. 40, n. 4, dez. 2000 . Disponível em <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0034-75902000000400009&lng=pt&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0034-75902000000400009&lng=pt&nrm=iso)>. Último acesso em 14 out. 2013. <http://dx.doi.org/10.1590/S0034-75902000000400009>.

ALBINADER Neto, JORGE Abilio. Web Services em Java. Rio de Janeiro, Brasport, 2006.

CUNHA, D. Web Services, SOAP e Aplicações Web. Disponível em: <[http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index\\_pt\\_br.html](http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index_pt_br.html)> Publicado em: 10 Dez 2002. Revisado em: 10 Dez 2002. Último acesso em: 02 de Junho de 2013.

DE SORDI, J. O. Gestão por processos: uma abordagem da moderna administração. São Paulo: Saraiva 2005.

FANTINATO M. 2007. Uma Abordagem Baseada em Características para o Estabelecimento de Contratos Eletrônicos para Serviços Web. Tese de Doutorado. Universidade Estadual de Campinas (UNICAMP), Instituto de Computação (IC), Campinas, SP.

GROSSI, B. E. 2005. Estudo do modelo de computação orientada a serviços e sua aplicação a um sistema de mineração de dados, Tese de Pós-Graduação. Universidade Federal de Minas Gerais (UFMG), Instituto de Ciências Exatas (ICEX), Belo Horizonte, MG. Disponível em: <<http://hdl.handle.net/1843/RVMR-6EAFWY>> Último acesso em: 10 de Outubro de 2013

MITRA N., Y. L. SOAP Version 1.2 Part 0: Primer (Second Edition). W3C Recommendation 27 April 2007. Disponível em: <<http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>> Último acesso: 22 de maio 2013.

MOSSI, W. BPM – Modelagem de Processos de Negócios. 2009. Disponível em: <<http://www.en-sof.com.br/downloads/BPM-modelagem-processos.pdf>>, Último acesso em: 10 de Outubro de 2013.

OLIVEIRA, P. A.; JUNIOR C. A. D; OLIVEIRA P. F. A. 2008. Proposição de Infra-Estrutura de Dados Espaciais (SDI) Local, Baseada em Arquitetura Orientada por Serviços (SOA) Disponível em: <<http://homepages.dcc.ufmg.br/~clodoveu/files/100.40/AC044.%202008%20Proposicao%20de%20infra-estrutura%20de%20dados%20espaciais%20local%20baseda%20em%20arquitetura%20orientada%20por%20servicos.pdf>> Último acesso em: 14 de Outubro de 2013.

TIZZO, N. P. 2010 – Arquitetura Orientada a Serviços: A Evolução de Estruturas Complexas a partir de Estruturas Simples. PUC, Minas; Disponível em: <[http://www.ifsc.usp.br/~9geinfo/images/stories/palestras/soa\\_neil.pdf](http://www.ifsc.usp.br/~9geinfo/images/stories/palestras/soa_neil.pdf)> Último acesso em: 10 de Outubro de 2013.

SANTOS, L. L. 2011. Monitoramento de contratos eletrônicos baseados em características. Dissertação de Mestrado. Universidade Estadual de Campinas (UNICAMP), Instituto de Computação (IC), Campinas, SP



**ANEXO B – SCRIPT DE CRIAÇÃO DO BANCO DE DADOS UDDI**

```
CREATE TABLE J3_PERSON_NAME (
    ID BIGINT NOT NULL,
    LANG_CODE VARCHAR(5),
    NAME VARCHAR(255) NOT NULL,
    CONTACT_ID BIGINT NOT NULL
);

CREATE TABLE J3_TRANSFER_TOKEN (
    TRANSFER_TOKEN VARCHAR(51) NOT NULL,
    EXPIRATION_DATE TIMESTAMP NOT NULL
);

CREATE TABLE J3_ADDRESS_LINE (
    ID BIGINT NOT NULL,
    KEY_NAME VARCHAR(255),
    KEY_VALUE VARCHAR(255),
    LINE VARCHAR(80) NOT NULL,
    ADDRESS_ID BIGINT NOT NULL
);

CREATE TABLE J3_INSTANCE_DETAILS_DOC_DESCR (
    ID BIGINT NOT NULL,
    DESCR VARCHAR(1024) NOT NULL,
    LANG_CODE VARCHAR(5),
    TMODEL_INSTANCE_INFO_ID BIGINT NOT NULL
);

CREATE TABLE J3_OVERVIEW_DOC_DESCR (
    ID BIGINT NOT NULL,
    DESCR VARCHAR(1024) NOT NULL,
    LANG_CODE VARCHAR(5),
    OVERVIEW_DOC_ID BIGINT
);

CREATE TABLE J3_PUBLISHER (
    AUTHORIZED_NAME VARCHAR(255) NOT NULL,
    EMAIL_ADDRESS VARCHAR(255),
    IS_ADMIN VARCHAR(5),
    IS_ENABLED VARCHAR(5),
    MAX_BINDINGS_PER_SERVICE INTEGER,
    MAX_BUSINESSES INTEGER,
    MAX_SERVICES_PER_BUSINESS INTEGER,
    MAX_TMODELS INTEGER,
    PUBLISHER_NAME VARCHAR(255) NOT NULL
);

CREATE TABLE J3_PUBLISHER_ASSERTION (
    FROM_KEY VARCHAR(255) NOT NULL,
    TO_KEY VARCHAR(255) NOT NULL,
    FROM_CHECK VARCHAR(5) NOT NULL,
    KEY_NAME VARCHAR(255) NOT NULL,
    KEY_VALUE VARCHAR(255) NOT NULL,
    TMODEL_KEY VARCHAR(255) NOT NULL,
    TO_CHECK VARCHAR(5) NOT NULL
);
```

```
CREATE TABLE J3_ADDRESS (
  ID BIGINT NOT NULL,
  SORT_CODE VARCHAR(10),
  TMODEL_KEY VARCHAR(255),
  USE_TYPE VARCHAR(255),
  ADDRESS_ID BIGINT NOT NULL
);

CREATE TABLE J3_UDDI_ENTITY (
  ENTITY_KEY VARCHAR(255) NOT NULL,
  AUTHORIZED_NAME VARCHAR(255) NOT NULL,
  CREATED_TIMESTAMP,
  MODIFIED_TIMESTAMP NOT NULL,
  MODIFIED_INCLUDING_CHILDREN_TIMESTAMP,
  NODE_ID VARCHAR(255)
);

CREATE TABLE J3_SUBSCRIPTION (
  SUBSCRIPTION_KEY VARCHAR(255) NOT NULL,
  AUTHORIZED_NAME VARCHAR(255) NOT NULL,
  BINDING_KEY VARCHAR(255),
  BRIEF SMALLINT,
  CREATE_DATE TIMESTAMP NOT NULL,
  EXPIRES_AFTER TIMESTAMP,
  LAST_NOTIFIED_TIMESTAMP,
  MAX_ENTITIES INTEGER,
  NOTIFICATION_INTERVAL VARCHAR(255),
  SUBSCRIPTION_FILTER CLOB(2147483647) NOT NULL
);

CREATE TABLE J3_BUSINESS_SERVICE (
  ENTITY_KEY VARCHAR(255) NOT NULL,
  BUSINESS_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_BUSINESS_DESCR (
  ID BIGINT NOT NULL,
  DESCR VARCHAR(1024) NOT NULL,
  LANG_CODE VARCHAR(5),
  ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_BUSINESS_CATEGORY_BAG (
  ID BIGINT NOT NULL,
  ENTITY_KEY VARCHAR(255)
);

CREATE TABLE J3_CLIENT_SUBSCRIPTIONINFO (
  SUBSCRIPTION_KEY VARCHAR(255) NOT NULL,
  LAST_NOTIFIED_TIMESTAMP,
  FROMCLERK_CLERK_NAME VARCHAR(255),
  TOCLERK_CLERK_NAME VARCHAR(255)
);

CREATE TABLE J3_KEYED_REFERENCE_GROUP (
  ID BIGINT NOT NULL,
  TMODEL_KEY VARCHAR(255),
  CATEGORY_BAG_ID BIGINT NOT NULL
);
```

```
CREATE TABLE J3_CONTACT (
  ID BIGINT NOT NULL,
  USE_TYPE VARCHAR(255),
  ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_TMODEL (
  ENTITY_KEY VARCHAR(255) NOT NULL,
  DELETED SMALLINT,
  LANG_CODE VARCHAR(5),
  NAME VARCHAR(255) NOT NULL
);

CREATE TABLE OPENJPA_SEQUENCE_TABLE (
  ID SMALLINT NOT NULL,
  SEQUENCE_VALUE BIGINT
);

CREATE TABLE J3_OVERVIEW_DOC (
  ID BIGINT NOT NULL,
  OVERVIEW_URL VARCHAR(255),
  OVERVIEW_URL_USE_TYPE VARCHAR(255),
  ENTITY_KEY VARCHAR(255),
  TOMODEL_INSTANCE_INFO_ID BIGINT
);

CREATE TABLE J3_TMODEL_INSTANCE_INFO (
  ID BIGINT NOT NULL,
  INSTANCE_PARMS VARCHAR(512),
  TMODEL_KEY VARCHAR(255) NOT NULL,
  ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_INSTANCE_DETAILS_DESCR (
  ID BIGINT NOT NULL,
  DESCR VARCHAR(1024) NOT NULL,
  LANG_CODE VARCHAR(5),
  TMODEL_INSTANCE_INFO_ID BIGINT NOT NULL
);

CREATE TABLE J3_TMODEL_DESCR (
  ID BIGINT NOT NULL,
  DESCR VARCHAR(1024) NOT NULL,
  LANG_CODE VARCHAR(5),
  ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_SUBSCRIPTION_MATCH (
  ID BIGINT NOT NULL,
  ENTITY_KEY VARCHAR(255) NOT NULL,
  SUBSCRIPTION_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_BINDING_CATEGORY_BAG (
  ID BIGINT NOT NULL,
  ENTITY_KEY VARCHAR(255)
);

CREATE TABLE J3_EMAIL (
  ID BIGINT NOT NULL,
```

```
EMAIL_ADDRESS VARCHAR(255) NOT NULL,
USE_TYPE VARCHAR(255),
CONTACT_ID BIGINT NOT NULL
);

CREATE TABLE J3_SERVICE_DESCR (
  ID BIGINT NOT NULL,
  DESCR VARCHAR(1024) NOT NULL,
  LANG_CODE VARCHAR(5),
  ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_PHONE (
  ID BIGINT NOT NULL,
  PHONE_NUMBER VARCHAR(50) NOT NULL,
  USE_TYPE VARCHAR(255),
  CONTACT_ID BIGINT NOT NULL
);

CREATE TABLE J3_TMODEL_INSTANCE_INFO_DESCR (
  ID BIGINT NOT NULL,
  DESCR VARCHAR(1024) NOT NULL,
  LANG_CODE VARCHAR(5),
  TMODEL_INSTANCE_INFO_ID BIGINT NOT NULL
);

CREATE TABLE J3_CLERK (
  CLERK_NAME VARCHAR(255) NOT NULL,
  CRED VARCHAR(255),
  PUBLISHER_ID VARCHAR(255) NOT NULL,
  NODE_NAME VARCHAR(255)
);

CREATE TABLE J3_TMODEL_CATEGORY_BAG (
  ID BIGINT NOT NULL,
  ENTITY_KEY VARCHAR(255)
);

CREATE TABLE J3_BINDING_DESCR (
  ID BIGINT NOT NULL,
  DESCR VARCHAR(1024) NOT NULL,
  LANG_CODE VARCHAR(5),
  ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_TRANSFER_TOKEN_KEYS (
  ID BIGINT NOT NULL,
  ENTITY_KEY VARCHAR(255),
  TRANSFER_TOKEN VARCHAR(255) NOT NULL
);

CREATE TABLE J3_DISCOVERY_URL (
  ID BIGINT NOT NULL,
  URL VARCHAR(255) NOT NULL,
  USE_TYPE VARCHAR(255) NOT NULL,
  ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_TMODEL_IDENTIFIER (
  ID BIGINT NOT NULL,
```

```
KEY_NAME VARCHAR(255),
KEY_VALUE VARCHAR(255) NOT NULL,
TMODEL_KEY_REF VARCHAR(255),
ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_CONTACT_DESCR (
  ID BIGINT NOT NULL,
  DESCR VARCHAR(1024) NOT NULL,
  LANG_CODE VARCHAR(5),
  CONTACT_ID BIGINT NOT NULL
);

CREATE TABLE J3_CATEGORY_BAG (
  ID BIGINT NOT NULL
);

CREATE TABLE J3_SERVICE_PROJECTION (
  BUSINESS_KEY VARCHAR(255) NOT NULL,
  SERVICE_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_BUSINESS_NAME (
  ID BIGINT NOT NULL,
  LANG_CODE VARCHAR(5),
  NAME VARCHAR(255) NOT NULL,
  ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_SERVICE_CATEGORY_BAG (
  ID BIGINT NOT NULL,
  ENTITY_KEY VARCHAR(255)
);

CREATE TABLE J3_NODE (
  NAME VARCHAR(255) NOT NULL,
  CUSTODY_TRANSFER_URL VARCHAR(255) NOT NULL,
  FACTORY_INITIAL VARCHAR(255),
  FACTORY_NAMING_PROVIDER VARCHAR(255),
  FACTORY_URL_PKGS VARCHAR(255),
  INQUIRY_URL VARCHAR(255) NOT NULL,
  JUDDI_API_URL VARCHAR(255),
  MANAGER_NAME VARCHAR(255) NOT NULL,
  PROXY_TRANSPORT VARCHAR(255) NOT NULL,
  PUBLISH_URL VARCHAR(255) NOT NULL,
  SECURITY_URL VARCHAR(255) NOT NULL,
  SUBSCRIPTION_URL VARCHAR(255) NOT NULL
);

CREATE TABLE J3_BINDING_TEMPLATE (
  ENTITY_KEY VARCHAR(255) NOT NULL,
  ACCESS_POINT_TYPE VARCHAR(255),
  ACCESS_POINT_URL VARCHAR(4000),
  HOSTING_REDIRECTOR VARCHAR(255),
  SERVICE_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_KEYED_REFERENCE (
  ID BIGINT NOT NULL,
  KEY_NAME VARCHAR(255),
```

```

        KEY_VALUE VARCHAR(255) NOT NULL,
        TMODEL_KEY_REF VARCHAR(255),
        CATEGORY_BAG_ID BIGINT,
        KEYED_REFERENCE_GROUP_ID BIGINT
    );

CREATE TABLE J3_AUTH_TOKEN (
    AUTH_TOKEN VARCHAR(51) NOT NULL,
    AUTHORIZED_NAME VARCHAR(255) NOT NULL,
    CREATED TIMESTAMP NOT NULL,
    LAST_USED TIMESTAMP NOT NULL,
    NUMBER_OF_USES INTEGER NOT NULL,
    TOKEN_STATE INTEGER NOT NULL
);

CREATE TABLE J3_SUBSCRIPTION_CHUNK_TOKEN (
    CHUNK_TOKEN VARCHAR(255) NOT NULL,
    DATA INTEGER NOT NULL,
    END_POINT TIMESTAMP,
    EXPIRES_AFTER TIMESTAMP NOT NULL,
    START_POINT TIMESTAMP,
    SUBSCRIPTION_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_SERVICE_NAME (
    ID BIGINT NOT NULL,
    LANG_CODE VARCHAR(5),
    NAME VARCHAR(255) NOT NULL,
    ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_BUSINESS_IDENTIFIER (
    ID BIGINT NOT NULL,
    KEY_NAME VARCHAR(255),
    KEY_VALUE VARCHAR(255) NOT NULL,
    TMODEL_KEY_REF VARCHAR(255),
    ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE TABLE J3_BUSINESS_ENTITY (
    ENTITY_KEY VARCHAR(255) NOT NULL
);

CREATE INDEX I_J3_TSCR_TMODEL ON J3_TMODEL_DESCR (ENTITY_KEY ASC);

CREATE UNIQUE INDEX SQL101221160128000 ON J3_TMODEL (ENTITY_KEY ASC);

CREATE INDEX I_J3_PHON_CONTACT ON J3_PHONE (CONTACT_ID ASC);

CREATE INDEX I_J3_BRVC_BUSINESSENTITY ON J3_BUSINESS_SERVICE
(BUSINESS_KEY ASC);

CREATE INDEX I_J3_S_NM_BUSINESSSERVICE ON J3_SERVICE_NAME (ENTITY_KEY
ASC);

CREATE INDEX I_J3_CNNF_TOCLERK ON J3_CLIENT_SUBSCRIPTIONINFO
(TOCLERK_CLERK_NAME ASC);

CREATE UNIQUE INDEX SQL101221160121420 ON J3_CLIENT_SUBSCRIPTIONINFO
(SUBSCRIPTION_KEY ASC);

```

```
CREATE INDEX I_J3_BSCR_BINDINGTEMPLATE ON J3_BINDING_DESCR (ENTITY_KEY
ASC);

CREATE INDEX I_J3_DRSS_CONTACT ON J3_ADDRESS (ADDRESS_ID ASC);

CREATE UNIQUE INDEX SQL101221160120410 ON J3_BUSINESS_NAME (ID ASC);

CREATE UNIQUE INDEX SQL101221160126530 ON J3_SERVICE_PROJECTION
(BUSINESS_KEY ASC, SERVICE_KEY ASC);

CREATE INDEX I_J3_P_NM_CONTACT ON J3_PERSON_NAME (CONTACT_ID ASC);

CREATE UNIQUE INDEX SQL101221160129560 ON J3_TRANSFER_TOKEN
(TRANSFER_TOKEN ASC);

CREATE UNIQUE INDEX SQL101221160123590 ON J3_KEYED_REFERENCE_GROUP (ID
ASC);

CREATE INDEX I_J3_VSCR_OVERVIEWDOC ON J3_OVERVIEW_DOC_DESCR
(OVERVIEW_DOC_ID ASC);

CREATE UNIQUE INDEX SQL101221160129830 ON J3_TRANSFER_TOKEN_KEYS (ID
ASC);

CREATE UNIQUE INDEX SQL101221160126780 ON J3_SUBSCRIPTION
(SUBSCRIPTION_KEY ASC);

CREATE UNIQUE INDEX SQL101221160118890 ON J3_BINDING_DESCR (ID ASC);

CREATE INDEX I_J3_CTCT_BUSINESSENTITY ON J3_CONTACT (ENTITY_KEY ASC);

CREATE UNIQUE INDEX SQL101221160123870 ON J3_NODE (NAME ASC);

CREATE UNIQUE INDEX SQL101221160128670 ON J3_TMODEL_IDENTIFIER (ID ASC);

CREATE UNIQUE INDEX SQL101221160122440 ON J3_EMAIL (ID ASC);

CREATE UNIQUE INDEX SQL101221160124160 ON J3_OVERVIEW_DOC (ID ASC);

CREATE INDEX I_J3_TSCR_TMODELINSTANCEINFO ON
J3_TMODEL_INSTANCE_INFO_DESCR (TMODEL_INSTANCE_INFO_ID ASC);

CREATE UNIQUE INDEX SQL101221160130130 ON J3_UDDI_ENTITY (ENTITY_KEY
ASC);

CREATE INDEX I_J3_NSCR_TMODELINSTANCEINFO ON J3_INSTANCE_DETAILS_DESCR
(TMODEL_INSTANCE_INFO_ID ASC);

CREATE UNIQUE INDEX SQL101221160125640 ON J3_PUBLISHER_ASSERTION
(FROM_KEY ASC, TO_KEY ASC);

CREATE UNIQUE INDEX SQL101221160119730 ON J3_BUSINESS_DESCR (ID ASC);

CREATE UNIQUE INDEX SQL101221160119500 ON J3_BUSINESS_CATEGORY_BAG (ID
ASC);

CREATE UNIQUE INDEX SQL101221160127550 ON J3_SUBSCRIPTION_CHUNK_TOKEN
(CHUNK_TOKEN ASC);
```

```
CREATE UNIQUE INDEX SQL101221160126090 ON J3_SERVICE_DESCR (ID ASC);

CREATE INDEX I_J3_BTFR_BUSINESSENTITY ON J3_BUSINESS_IDENTIFIER
(ENTITY_KEY ASC);

CREATE UNIQUE INDEX SQL101221160120180 ON J3_BUSINESS_IDENTIFIER (ID
ASC);

CREATE UNIQUE INDEX SQL101221160128210 ON J3_TMODEL_CATEGORY_BAG (ID
ASC);

CREATE INDEX I_J3_BSCR_BUSINESSENTITY ON J3_BUSINESS_DESCR (ENTITY_KEY
ASC);

CREATE INDEX I_J3_BPLT_BUSINESSSERVICE ON J3_BINDING_TEMPLATE
(SERVICE_KEY ASC);

CREATE INDEX I_J3_T_NF_BINDINGTEMPLATE ON J3_TMODEL_INSTANCE_INFO
(ENTITY_KEY ASC);

CREATE UNIQUE INDEX SQL101221160118450 ON J3_AUTH_TOKEN (AUTH_TOKEN ASC);

CREATE INDEX I_J3_B_BG_BINDINGTEMPLATE ON J3_BINDING_CATEGORY_BAG
(ENTITY_KEY ASC);

CREATE UNIQUE INDEX SQL101221160120860 ON J3_CATEGORY_BAG (ID ASC);

CREATE INDEX I_J3_V_DC_TMODEL ON J3_OVERVIEW_DOC (ENTITY_KEY ASC);

CREATE INDEX I_J3_S_BG_BUSINESSSERVICE ON J3_SERVICE_CATEGORY_BAG
(ENTITY_KEY ASC);

CREATE UNIQUE INDEX SQL101221160123320 ON J3_KEYED_REFERENCE (ID ASC);

CREATE UNIQUE INDEX SQL101221160121930 ON J3_CONTACT_DESCR (ID ASC);

CREATE UNIQUE INDEX SQL101221160121170 ON J3_CLERK (CLERK_NAME ASC);

CREATE UNIQUE INDEX SQL101221160127780 ON J3_SUBSCRIPTION_MATCH (ID ASC);

CREATE UNIQUE INDEX SQL101221160120640 ON J3_BUSINESS_SERVICE (ENTITY_KEY
ASC);

CREATE UNIQUE INDEX SQL101221160128950 ON J3_TMODEL_INSTANCE_INFO (ID
ASC);

CREATE INDEX I_J3_CLRK_NODE ON J3_CLERK (NODE_NAME ASC);

CREATE INDEX I_J3_STCH_SUBSCRIPTION ON J3_SUBSCRIPTION_MATCH
(SUBSCRIPTION_KEY ASC);

CREATE UNIQUE INDEX SQL101221160119940 ON J3_BUSINESS_ENTITY (ENTITY_KEY
ASC);

CREATE INDEX I_J3_EMIL_CONTACT ON J3_EMAIL (CONTACT_ID ASC);

CREATE UNIQUE INDEX SQL101221160118210 ON J3_ADDRESS_LINE (ID ASC);

CREATE UNIQUE INDEX SQL101221160128420 ON J3_TMODEL_DESCR (ID ASC);
```



```
CREATE INDEX I_J3_T_BG_TMODEL ON J3_TMODEL_CATEGORY_BAG (ENTITY_KEY ASC);

CREATE INDEX I_J3_KRNC_CATEGORYBAG ON J3_KEYED_REFERENCE (CATEGORY_BAG_ID
ASC);

CREATE UNIQUE INDEX SQL101221160126300 ON J3_SERVICE_NAME (ID ASC);

CREATE UNIQUE INDEX SQL101221160125370 ON J3_PUBLISHER (AUTHORIZED_NAME
ASC);

CREATE INDEX I_J3_SSCR_BUSINESSSERVICE ON J3_SERVICE_DESCR (ENTITY_KEY
ASC);

CREATE UNIQUE INDEX SQL101221160122740 ON J3_INSTANCE_DETAILS_DESCR (ID
ASC);

CREATE UNIQUE INDEX SQL101221160129300 ON J3_TMODEL_INSTANCE_INFO_DESCR
(ID ASC);

CREATE INDEX I_J3_KRNC_KEYEDREFERENCEGROUP ON J3_KEYED_REFERENCE
(KEYED_REFERENCE_GROUP_ID ASC);

CREATE INDEX I_J3_B_NM_BUSINESSENTITY ON J3_BUSINESS_NAME (ENTITY_KEY
ASC);

CREATE UNIQUE INDEX SQL101221160125870 ON J3_SERVICE_CATEGORY_BAG (ID
ASC);

CREATE INDEX I_J3_D_LN_ADDRESS ON J3_ADDRESS_LINE (ADDRESS_ID ASC);

CREATE INDEX I_J3_TTFR_TMODEL ON J3_TMODEL_IDENTIFIER (ENTITY_KEY ASC);

CREATE INDEX I_J3_NSCR_TMODELINSTANCEINFO1 ON
J3_INSTANCE_DETAILS_DOC_DESCR (TMODEL_INSTANCE_INFO_ID ASC);

CREATE UNIQUE INDEX SQL101221160124440 ON J3_OVERVIEW_DOC_DESCR (ID ASC);

CREATE UNIQUE INDEX SQL101221160124690 ON J3_PERSON_NAME (ID ASC);

CREATE INDEX I_J3_CNNF_FROMCLERK ON J3_CLIENT_SUBSCRIPTIONINFO
(FROMCLERK_CLERK_NAME ASC);

CREATE UNIQUE INDEX SQL101221160119280 ON J3_BINDING_TEMPLATE (ENTITY_KEY
ASC);

CREATE UNIQUE INDEX SQL101221160117940 ON J3_ADDRESS (ID ASC);

CREATE INDEX I_J3_KGRP_CATEGORYBAG ON J3_KEYED_REFERENCE_GROUP
(CATEGORY_BAG_ID ASC);

CREATE UNIQUE INDEX SQL101221160124970 ON J3_PHONE (ID ASC);

CREATE UNIQUE INDEX SQL101221160130460 ON OPENJPA_SEQUENCE_TABLE (ID
ASC);

CREATE UNIQUE INDEX SQL101221160123010 ON J3_INSTANCE_DETAILS_DOC_DESCR
(ID ASC);

CREATE INDEX I_J3_CSCR_CONTACT ON J3_CONTACT_DESCR (CONTACT_ID ASC);
```

```
CREATE UNIQUE INDEX SQL101221160121690 ON J3_CONTACT (ID ASC);

CREATE INDEX I_J3_TKYS_TRANSFERTOKEN ON J3_TRANSFER_TOKEN_KEYS
(TRANSFER_TOKEN ASC);

CREATE INDEX I_J3_V_DC_TMODELINSTANCEINFO ON J3_OVERVIEW_DOC
(TOMODEL_INSTANCE_INFO_ID ASC);

CREATE UNIQUE INDEX SQL101221160122200 ON J3_DISCOVERY_URL (ID ASC);

CREATE INDEX I_J3_B_BG_BUSINESSENTITY ON J3_BUSINESS_CATEGORY_BAG
(ENTITY_KEY ASC);

CREATE UNIQUE INDEX SQL101221160118680 ON J3_BINDING_CATEGORY_BAG (ID
ASC);

CREATE INDEX I_J3_D_RL_BUSINESSENTITY ON J3_DISCOVERY_URL (ENTITY_KEY
ASC);

ALTER TABLE J3_SUBSCRIPTION_MATCH ADD CONSTRAINT SQL101221160127780
PRIMARY KEY (ID);

ALTER TABLE J3_SERVICE_DESCR ADD CONSTRAINT SQL101221160126090 PRIMARY
KEY (ID);

ALTER TABLE J3_AUTH_TOKEN ADD CONSTRAINT SQL101221160118450 PRIMARY KEY
(AUTH_TOKEN);

ALTER TABLE J3_TMODEL_INSTANCE_INFO_DESCR ADD CONSTRAINT
SQL101221160129300 PRIMARY KEY (ID);

ALTER TABLE J3_TMODEL_DESCR ADD CONSTRAINT SQL101221160128420 PRIMARY KEY
(ID);

ALTER TABLE J3_BUSINESS_CATEGORY_BAG ADD CONSTRAINT SQL101221160119500
PRIMARY KEY (ID);

ALTER TABLE J3_INSTANCE_DETAILS_DESCR ADD CONSTRAINT SQL101221160122740
PRIMARY KEY (ID);

ALTER TABLE J3_TMODEL_IDENTIFIER ADD CONSTRAINT SQL101221160128670
PRIMARY KEY (ID);

ALTER TABLE J3_TRANSFER_TOKEN ADD CONSTRAINT SQL101221160129560 PRIMARY
KEY (TRANSFER_TOKEN);

ALTER TABLE J3_TMODEL ADD CONSTRAINT SQL101221160128000 PRIMARY KEY
(ENTITY_KEY);

ALTER TABLE J3_TMODEL_INSTANCE_INFO ADD CONSTRAINT SQL101221160128950
PRIMARY KEY (ID);

ALTER TABLE J3_CLERK ADD CONSTRAINT SQL101221160121170 PRIMARY KEY
(CLERK_NAME);

ALTER TABLE J3_OVERVIEW_DOC ADD CONSTRAINT SQL101221160124160 PRIMARY KEY
(ID);

ALTER TABLE J3_INSTANCE_DETAILS_DOC_DESCR ADD CONSTRAINT
SQL101221160123010 PRIMARY KEY (ID);
```

```
ALTER TABLE J3_ADDRESS_LINE ADD CONSTRAINT SQL101221160118210 PRIMARY KEY
(ID);

ALTER TABLE J3_OVERVIEW_DOC_DESCR ADD CONSTRAINT SQL101221160124440
PRIMARY KEY (ID);

ALTER TABLE J3_NODE ADD CONSTRAINT SQL101221160123870 PRIMARY KEY (NAME);

ALTER TABLE J3_BUSINESS_ENTITY ADD CONSTRAINT SQL101221160119940 PRIMARY
KEY (ENTITY_KEY);

ALTER TABLE J3_KEYED_REFERENCE_GROUP ADD CONSTRAINT SQL101221160123590
PRIMARY KEY (ID);

ALTER TABLE J3_SUBSCRIPTION ADD CONSTRAINT SQL101221160126780 PRIMARY KEY
(SUBSCRIPTION_KEY);

ALTER TABLE OPENJPA_SEQUENCE_TABLE ADD CONSTRAINT SQL101221160130460
PRIMARY KEY (ID);

ALTER TABLE J3_BUSINESS_IDENTIFIER ADD CONSTRAINT SQL101221160120180
PRIMARY KEY (ID);

ALTER TABLE J3_EMAIL ADD CONSTRAINT SQL101221160122440 PRIMARY KEY (ID);

ALTER TABLE J3_KEYED_REFERENCE ADD CONSTRAINT SQL101221160123320 PRIMARY
KEY (ID);

ALTER TABLE J3_CONTACT_DESCR ADD CONSTRAINT SQL101221160121930 PRIMARY
KEY (ID);

ALTER TABLE J3_PHONE ADD CONSTRAINT SQL101221160124970 PRIMARY KEY (ID);

ALTER TABLE J3_CONTACT ADD CONSTRAINT SQL101221160121690 PRIMARY KEY
(ID);

ALTER TABLE J3_BINDING_DESCR ADD CONSTRAINT SQL101221160118890 PRIMARY
KEY (ID);

ALTER TABLE J3_PUBLISHER ADD CONSTRAINT SQL101221160125370 PRIMARY KEY
(AUTHORIZED_NAME);

ALTER TABLE J3_UDDI_ENTITY ADD CONSTRAINT SQL101221160130130 PRIMARY KEY
(ENTITY_KEY);

ALTER TABLE J3_CATEGORY_BAG ADD CONSTRAINT SQL101221160120860 PRIMARY KEY
(ID);

ALTER TABLE J3_BUSINESS_DESCR ADD CONSTRAINT SQL101221160119730 PRIMARY
KEY (ID);

ALTER TABLE J3_DISCOVERY_URL ADD CONSTRAINT SQL101221160122200 PRIMARY
KEY (ID);

ALTER TABLE J3_BUSINESS_SERVICE ADD CONSTRAINT SQL101221160120640 PRIMARY
KEY (ENTITY_KEY);

ALTER TABLE J3_TRANSFER_TOKEN_KEYS ADD CONSTRAINT SQL101221160129830
PRIMARY KEY (ID);
```

```
ALTER TABLE J3_CLIENT_SUBSCRIPTIONINFO ADD CONSTRAINT SQL101221160121420
PRIMARY KEY (SUBSCRIPTION_KEY);

ALTER TABLE J3_SUBSCRIPTION_CHUNK_TOKEN ADD CONSTRAINT SQL101221160127550
PRIMARY KEY (CHUNK_TOKEN);

ALTER TABLE J3_SERVICE_CATEGORY_BAG ADD CONSTRAINT SQL101221160125870
PRIMARY KEY (ID);

ALTER TABLE J3_PERSON_NAME ADD CONSTRAINT SQL101221160124690 PRIMARY KEY
(ID);

ALTER TABLE J3_ADDRESS ADD CONSTRAINT SQL101221160117940 PRIMARY KEY
(ID);

ALTER TABLE J3_PUBLISHER_ASSERTION ADD CONSTRAINT SQL101221160125640
PRIMARY KEY (FROM_KEY, TO_KEY);

ALTER TABLE J3_TMODEL_CATEGORY_BAG ADD CONSTRAINT SQL101221160128210
PRIMARY KEY (ID);

ALTER TABLE J3_SERVICE_PROJECTION ADD CONSTRAINT SQL101221160126530
PRIMARY KEY (BUSINESS_KEY, SERVICE_KEY);

ALTER TABLE J3_SERVICE_NAME ADD CONSTRAINT SQL101221160126300 PRIMARY KEY
(ID);

ALTER TABLE J3_BINDING_CATEGORY_BAG ADD CONSTRAINT SQL101221160118680
PRIMARY KEY (ID);

ALTER TABLE J3_BUSINESS_NAME ADD CONSTRAINT SQL101221160120410 PRIMARY
KEY (ID);

ALTER TABLE J3_BINDING_TEMPLATE ADD CONSTRAINT SQL101221160119280 PRIMARY
KEY (ENTITY_KEY);
```