

INTEROPERABILIDADE ENTRE SISTEMAS DISTRIBUÍDOS UTILIZANDO WEB SERVICES

CAMPOS, MARCELO COSTA

Faculdades Integradas Cândido Rondon – UNIRONDON - Cuiabá / MT

marcelocampos@terra.com.br

Abstract. *This article presents as an emergent technology, Web Services, could be being used to support the connection between distributed systems, in practically isolated environments. With this, being able to be supplying to mechanisms of connection and integration of information, that previously to this technology, were restricted á an only platform or computational environment. For this integration of systems she is necessary to make use of the characteristic interoperabilidade, mechanisms of compatibility between them, that it will be reached using the technology of Web Services.*

Resumo. *Este artigo apresenta como uma tecnologia emergente, Web Services, poderá estar sendo utilizada para suportar a conexão entre sistemas distribuídos, em ambientes praticamente isolados. Com isso, podendo estar fornecendo mecanismos de conexão e integração de informações, que anteriormente a esta tecnologia, ficava restrita á uma única plataforma ou ambiente computacional. Para esta integração de sistemas é preciso dispor da característica interoperabilidade, mecanismos de compatibilidade entre eles, que será alcançada utilizando a tecnologia de Web Services.*

1. Introdução

A evolução das telecomunicações, permitindo o uso da Internet em alta velocidade, associada à multiplicidade de ofertas de novos dispositivos eletrônicos que acessam a Internet como PCs de bolso, hand-helds, celulares, TV interativa ou PDAs, estão exigindo mais da tradicional tecnologia da informação. Atualmente, o software não é mais constituído apenas por sistemas operacionais, aplicativos de produtividade ou ferramentas de desenvolvimento. A tecnologia passou a ter que suportar uma nova categoria de programas a serem distribuídos pela Internet, chamados de aplicações Web e Web Services.

Os problemas atuais da informática e do uso da Internet são muito mais uma conseqüência de uma evolução extremamente rápida, e por isso não ordenada, do que de uma falta de tecnologia para resolvê-los. Nas últimas décadas acompanhamos um desenvolvimento sem precedentes dos sistemas de informação. Esta grande evolução levou pouquíssimo tempo e, por isso, não pôde ser planejada em seus detalhes.

Ela foi ocorrendo na medida em que as empresas e as pessoas se adaptavam à nova realidade. Desta forma, não poderíamos esperar que a presente situação fosse a mais perfeita. Muitas concessões tiveram que ser feitas neste processo. Nem sempre uma nova tecnologia foi implantada da maneira

ideal; ela tinha que ser incorporada da melhor forma possível aos processos em uso.

Diante destas evoluções, o desenvolvimento de sistemas se tornou uma encruzilhada tecnológica. Dispomos de uma infra-estrutura nunca antes imaginada, mas ainda não dispomos de sistemas que utilizem todo este potencial. Sistemas que se possa integrar com outros que já foram desenvolvidos utilizando produtos tecnológicos distintos, plataformas e arquiteturas até então incompatíveis.

Os sistemas foram desenvolvidos para redes locais e estão sendo adaptados para operar na grande rede (Internet). Através desta necessidade de adaptação e interação de informações entre sistemas legados, precisamos de uma nova visão de software na qual a Internet esteja presente em todos os detalhes e que possamos distribuir serviços de forma ordenada e controlada.

Analisando tais fatos, pretende-se com este artigo apresentar a solução de Web Services para a utilização em técnicas, tecnologia e metodologia para minimizar o problema da interoperabilidade entre sistemas distribuídos existentes no mercado. Para que se possa obter um elevado nível de compatibilidade entre produtos de diferentes fornecedores, utilizando a nova tecnologia de Web Services, baseado em estruturas XML.

2. Sistemas Distribuídos

Uma **aplicação distribuída** é uma aplicação que foi dividida ou particionada em componentes lógicos individuais chamados de partições. A execução das aplicações pode ocorrer em máquinas diferentes na rede. Uma aplicação pode ser composta de objetos gráficos e scripts lógicos de programação que trabalham em conjunto para executar alguma tarefa. [Orf 2000]

“É uma coleção de computadores independentes que se apresentam aos usuários como um sistema único e coerente.”(Tanenbaum)

“É um sistema em que os componentes se localizam em uma rede de computadores e coordenam suas ações através de passagem de mensagens.”(Coulouris)

Durante a execução, as partições trabalham em conjunto para executar tarefas solicitadas pela aplicação como um todo. A existência de partições é transparente para os usuários da aplicação distribuída. O usuário executa a aplicação de uma simples máquina como se toda a aplicação estivesse nela. [Gal 2004]

O particionamento de uma aplicação distribuída significa que os dados, componentes de processamento e os componentes de apresentação das aplicações podem residir em diferentes máquinas em uma instalação cliente/servidora. [Orf 2000]

Nesse modelo, a plataforma cliente apresenta os dados e gerencia o teclado e outros dispositivos de entrada. Ela também pode conter alguns componentes de lógica. As lógicas das aplicações residem em uma ou mais partições e o seu processamento é remoto podendo ocorrer em um ou mais servidores, chamados de servidores de aplicações. A comunicação entre as aplicações clientes e as partições ocorrem por meio de mensagens.

2.1. Vantagens de Sistemas Distribuídos

Aplicações particionadas podem resolver problemas associados à distribuição de software e de execução que podem ocorrer quando se cria um sistema cliente/servidor em larga escala [Orf 2000]. Dentre as vantagens podemos citar:

- Redução no tempo de carga e de execução nas plataformas Cliente;
- Redução no Tráfego da Rede;
- Distribuição de Software Simplificada;
- Independência de Localização física;
- Alta Disponibilidade da Aplicação;
- Possibilidade de Heterogeneidade de Ambientes;
- Maior poder de Escalabilidade;

3. Arquitetura de um Web Service

O Web Service é uma tecnologia emergente, sobre a qual muito se tem pesquisado. Alguns pesquisadores definem como o caminho a seguir no desenvolvimento de aplicações distribuídas, enquanto que outros vêem nelas apenas mais uma evolução de um conceito antigo. Um web Service é uma aplicação de software que pode ser acessada remotamente usando diferentes linguagens baseadas em XML. [Pot 2003]

Sendo aplicações modulares, auto descritivas, acessíveis através de uma URL, independentes das plataformas de desenvolvimento e que permitem a interação entre aplicações sem intervenção humana, os Web Services apresentam-se como a solução para os atuais problemas de integração de aplicações distribuídas [Lop 2004].

Estas suas características devem-se em grande parte ao fato de se basearem em normas standard, de entre as quais se destacam: XML, SOAP, WSDL e UDDI. Um Web Service é semelhante na medida que é acessado através de uma URL. A diferença esta no conteúdo do que é enviado na requisição do cliente para o servidor. Os clientes de Web Service enviam um documento XML, formatado de uma maneira especial, de acordo com as regras da especificação SOAP. [Cer 2002]

O ciclo de vida de um Web Service compreende quatro estados distintos, conforme a figura 1, são eles:

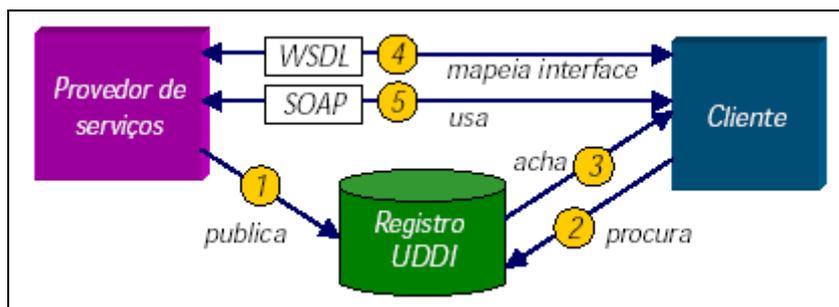


Figura 1 – Ciclo de Vida de um Web Service

Publicação: Processo, opcional, através do qual o fornecedor do Web Service possa conhecer a existência do seu serviço, efetuando o registro do mesmo no repositório de Web Services (UDDI);

Descoberta: Processo, opcional, através do qual uma aplicação cliente toma conhecimento da existência do Web Service pretendido pesquisando num repositório UDDI;

Descrição: Processo pelo qual o Web Service expõem a sua API (documento WSDL); desta maneira a aplicação cliente tem acesso a toda a interface do Web Service, onde se encontram descritas todas as funcionalidades por ele disponibilizadas, assim como os tipos de mensagens que permitem processar as rotinas;

Invocação: Processo pelo qual cliente e servidor interagem, através do envio de mensagens de entrada e de eventual recepção de mensagem de saída do Web Service.

Fazendo com que os Web Services possam ser desenvolvidos e publicados para poderem ser utilizados por um processo de software. Os Web Service funcionam á partir de uma necessidade de software. Que deverá possuir uma solicitação do Cliente, onde o Web Service estará executando as atividades necessárias para poder processar as informações e retornar em formato XML, conforme mostrado na figura 2.

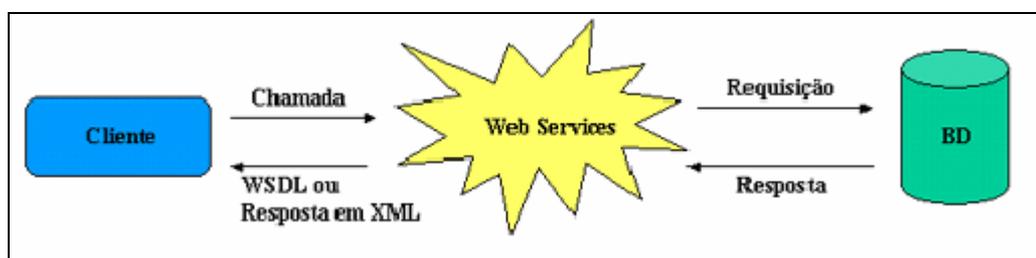


Figura 2 – Funcionamento do Web Service

4. Interoperabilidade entre Sistemas

Para a integração entre sistemas heterogêneos, é preciso dispor da característica da interoperabilidade entre eles. A interoperabilidade entre sistemas pode ser entendida como a capacidade de um sistema se comunicar com outros sistemas, compartilhando dados ou invocando processos comuns, independentes de sua

plataforma, arquitetura, linguagem de programação ou sistema operacional. [Gal 2004]

A interoperabilidade será aberta se essa conversação puder ser efetuada entre sistemas de diferentes fabricantes, fazendo com que as informações possa ser compartilhada e totalmente transparente para o usuário final.

Com as características de um sistema distribuído, os Web Services proporcionará as aplicações, toda essa integração de informações que se precisa atualmente. Gerando aos sistemas todas as vantagens encontradas nesse tipo de arquitetura, tais como, maior escalabilidade, desempenho, interoperabilidade, segurança e rapidez em suas utilizações. [Pot 2003]

A maioria do entusiasmo em torno dos Web Services é baseada na promessa de interoperabilidade. A arquitetura dos Web Services se baseia no envio de mensagens XML em um formato SOAP. As implicações disso são significativas, tais como:

- Não importa que tipo de computador envia a mensagem SOAP, ou em que sistema operacional esta rodando;
- Não importa em que lugar do mundo o cliente esta enviando a mensagem;
- Não importa em que Linguagem foi escrito o Software Cliente;
- Não há necessidade de saber que tipo de processador SOAP esta rodando no Servidor, apenas que esteja no padrão;

Os Web Services podem ser desenvolvidos usando qualquer Linguagem de Programação que suporta conexão WEB. Independente de fornecedores de plataformas de desenvolvimento, os web services se comunicam através dos padrões XML, SOAP e HTTP. [Cer 2002]

Com base em novos padrões de comunicação e com o alcance que a Internet nos proporciona, poderemos dar um passo definitivo em direção a uma nova plataforma para resolver as grandes questões que se colocam para áreas de integração de sistema. Sem excluir os sistemas que foram desenvolvidos anteriormente e torna-los totalmente distribuídos.

5. Integração de Sistemas Distribuídos com Web Services

Este modelo de projeto de integração, é que mais chama a atenção dos desenvolvedores, pois praticamente qualquer sistema pode acessar os Web Services, desde que utilizem algumas tecnologias. Os Web Services são independentes de plataforma e são encontrados em servidores web como componentes, que possam ser utilizados por outros sistemas, independente da linguagem de programação.

A grande motivação para o uso desse tipo de tecnologia é a infra-estrutura WEB ou a arquitetura de Componentes Distribuídos. Que fazem com que este componente de um Servidor Web possa ser utilizado por vários clientes de qualquer lugar da Internet.

Para conseguirmos estabelecer a utilização da característica da Interoperabilidade entre Sistemas Distribuídos e Heterogêneos, estaremos demonstrando como integrar sistemas utilizando Web Service. Desenvolvendo um Web Service em JAVA e consumindo o mesmo por uma aplicação em .Net, planejando assim a utilização de técnicas, tecnologia e metodologia para esta conexão de sistemas.

5.1. Desenvolvendo o Web Service – Plataforma JAVA

Temos um sistema de controle de Frota de Veículos que foi desenvolvido em JAVA. Este sistema armazena informações em um Banco de Dados e esta funcionando sobre a plataforma de um Servidor Web “TOMCAT 5.0”, conforme a figura 3.



Figura 3 – Web Server – TOMCAT 5.0

Foi desenvolvido um Web Service, conforme tabela 1, utilizando a Linguagem JAVA, que realiza uma consulta de informações em um Banco de Dados SQL Server 2000. Este Web Service esta disponível como um componente de software e foi publicado no Servidor Web “TOMCAT 5.0”.

```
/*  
* Web Service - Veiculo  
* Desenvolvido para realizar uma consulta no Banco de Dados Veiculo  
* Metodos do Web Service  
*     1- Consulta a Placa do Veiculo, na Tabela Veiculo pelo campo Placa  
(Caracter de 7)  
*     Retornando o Nome do Veiculo e o Ano  
*/  
import java.io.*;  
import java.sql.*;
```

```

public class WSVeiculo
{
    //Criando uma Variavel para Conexao
    private String query;

    //Metodod de Consulta Veiculo por Placa - Retornando o Nome do Veiculo e Ano
    public String ConsultaPlaca(String Placa)
    {
        try
        {
            Connection con;
            query = "SELECT VEICULO, ANO FROM VEICULO WHERE PLACA = '"+
Placa +"' ";
            //Preparando as Variaveis para a Conexao com o Banco de Dados
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

            //conexao SQL - Criar a Conexao no System DSN - ODBC
            con = DriverManager.getConnection("jdbc:odbc:JAVA","sa","mcc1301");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query);

            boolean more = rs.next();
            if (! more)
            {
                return " ** Veiculo Nao Cadastrado ** ";
            }

            //Retornando a Variavel
            return rs.getString("VEICULO")+rs.getString("ANO");

        }
        catch (Exception e)
        {
            return "ERRO NO WEB-SERVICE";
        }
    }

    //Fim do Metodo ConsultaPlaca
}

```

Tabela 1 – Código do Web Service em Java

Este Web Service, nomeado como “WSVeiculo”, estará disponibilizando para os seus clientes um método de consulta, nomeado de “ConsultaPlaca”. A tarefa que nosso Web Service executará será bem simples, ele receberá um parâmetro do tipo “String”, que será a placa do veículo, e efetuará a conexão no banco de dados e a consulta detalhada na tabela 1. Retornando em formato XML, conforme figura 4, ao cliente da solicitação do Web Service, o nome do Veículo e a Placa, caso o mesmo seja encontrado no Banco de Dados.

```
Address http://localhost:8080/axis/WSVeiculo.jws?wsdl
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="http://localhost:8080/axis/WSVeiculo.jws" xm
  soap" xmlns:impl="http://localhost:8080/axis/WSVeiculo.jws" xmlns:intf="http://
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://w
- <!--
  WSDL created by Apache Axis version: 1.2beta
  Built on Mar 31, 2004 (12:47:03 EST)
```

Figura 4 – Web Service “WSVeiculo” em formato XML

Então possuímos um Ambiente Computacional na arquitetura JAVA. Onde o Web service foi desenvolvido na Linguagem JAVA e publicado no servidor Web padrão JAVA. Conforme comentado durante o artigo, até antes essa tecnologia de Web Service, não poderíamos estar acessando este sistema e utilizando as suas regras de negócios, se não fosse executada por alguns de seu próprios componentes, respeitando a linguagem e a estrutura operacional.

Então para prover a característica da interoperabilidade, foi desenvolvido uma aplicação Web Distribuída, utilizando a plataforma .Net da Microsoft. Desenvolvendo assim, uma aplicação para realizar uma consulta e a integração entre plataformas distintas (JAVA X .Net), através do Web Service “WSVeiculo” desenvolvido em JAVA, veremos a seguir.

5.2. Consumindo o Web Service – Plataforma .Net

Na nossa aplicação, utilizaremos um sistema desenvolvido na arquitetura .Net utilizando a linguagem de programação orientada a objetos C# para podermos consumir o Web Service criado no passo anterior. No ambiente dos Web Services, o papel de uma biblioteca de tipos é desempenhada pela descrição WSDL do mesmo, sendo que uma das maneiras de acessa-la é adicionar a URL o parâmetro WSDL e instanciar o Web Reference, veja figura 5.

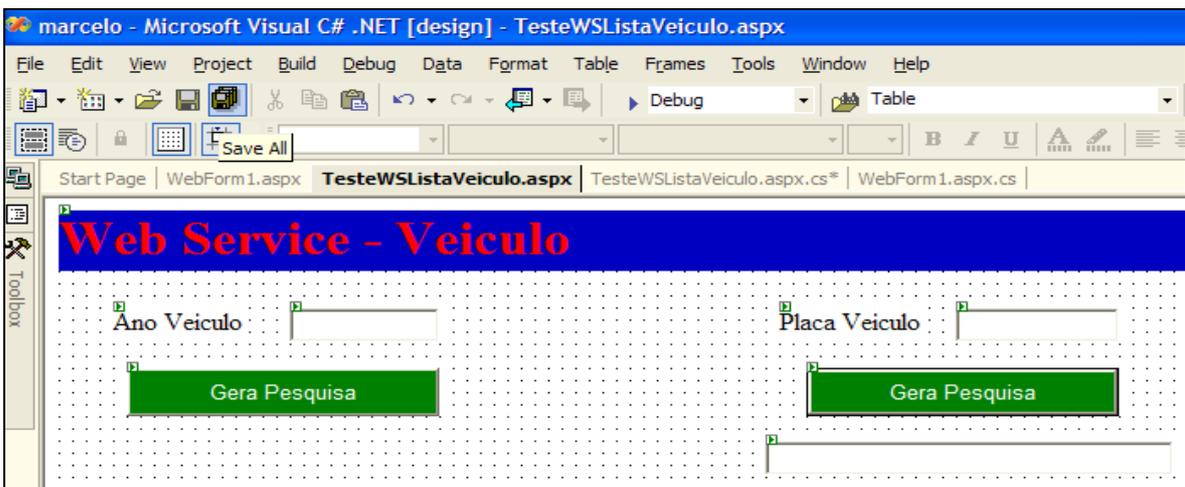


Figura 5– Aplicação desenvolvida em .Net

Então foi desenvolvido uma aplicação em .Net que estará consumindo o Web Service em JAVA. Esta aplicação poderá estar alocada em qualquer servidor

Web distribuído pela Internet, utilizando recursos computacionais diversos. Ela estará referenciando o Web Service no servidor Web TOMCAT, veja figura 6, e repassando um parâmetro (placa do veículo) para o mesmo processar e retornar as informações necessárias.

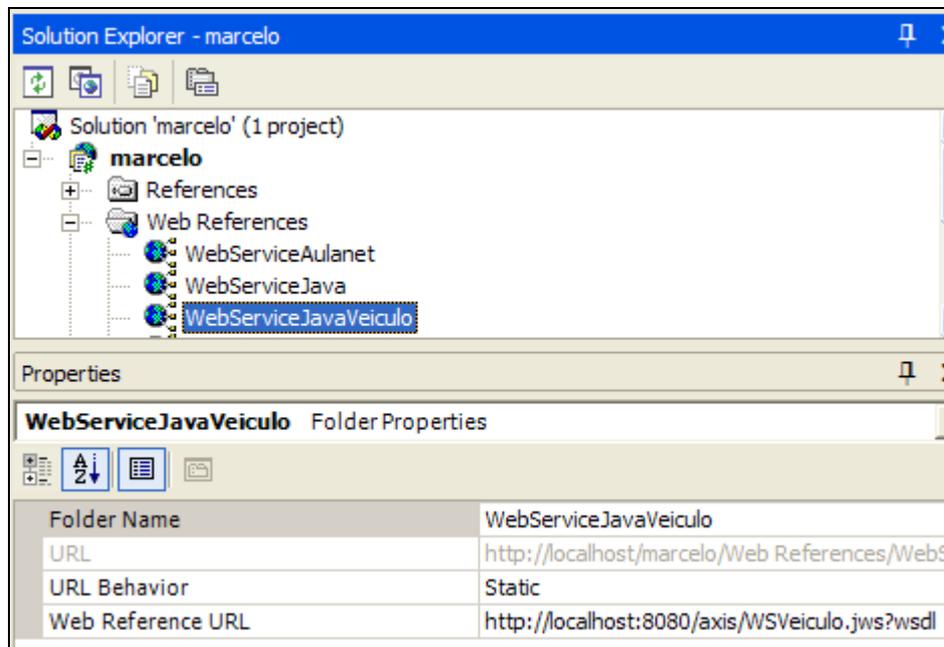


Figura 6– Referenciando o Web Service no Servidor WEB TOMCAT

Com estas configurações, a aplicação em .Net enviará a requisição que será processada pelo Web Service e retornará o resultado requisitado, em formato XML. Este resultado será mostrado na tela, conforme a figura 7.

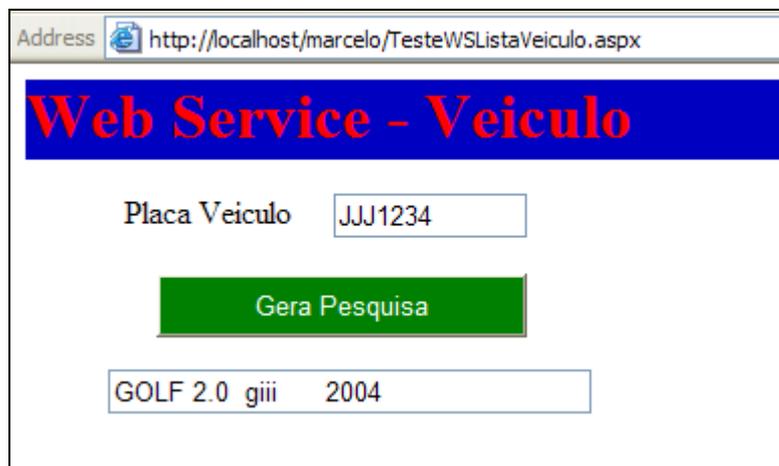


Figura 7– Executando a aplicação em .Net, com o retorno do Web Service JAVA

Com este simples exemplo podemos mostrar a facilidade de se integrar diversas plataformas até antes isoladas. Utilizamos uma estrutura de Web Service em JAVA e uma aplicação em .Net, fazendo assim, a utilização da característica da interoperabilidade entre plataformas.

6. Conclusão

Esta nova tecnologia, os Web Services apresentam-se como uma das soluções para alguns problemas associados ao desenvolvimento e integração entre sistemas distribuídos heterogêneos, principalmente nas questões relacionadas com a integração de negócios e informações.

No entanto o fato de serem componentes de software modulares, autodescritivas, que se baseiam em protocolos e padronizações reconhecidas e que permitem a interação entre aplicações sem intervenção humana, torna os Web Services uma das tecnologias para o futuro dos Sistemas Distribuídos.

Os resultados dos experimentos realizados foram satisfatórios. Muitas das técnicas e metodologias utilizadas neste artigo poderão ser aproveitadas para o desenvolvimento de outros projetos de Web Services. Serviços esses que vão ser independentes de plataforma operacional, linguagem de programação, banco de dados e de utilização distribuída.

Concluimos então que podemos utilizar a tecnologia de Web Services para prover uma das características importantes dos sistemas distribuídos, a interoperabilidade. Fazendo com que os sistemas desenvolvidos em plataformas e linguagens diferentes possam se integrar, respeitando assim todo um legado de sistemas em utilização atualmente. Neste artigo observamos que uma aplicação na plataforma JAVA possa se integrar com uma aplicação na arquitetura .Net.



Figura 8 – Integração de Arquitetura Distintas

São sugeridos como trabalhos futuros, a realização de testes mais extensos com a integração de outras linguagens de desenvolvimento e utilizando diversos ambientes computacionais distribuídos. Utilização conjunta de outras técnicas e metodologia para o desenvolvimento de Web Services.

7. Referências Bibliográficas

- [Cer 2002] Cerami , Ethan. "Web Services Essentials". O´Reilly, 2002.
- [Gal 2004] Galuppo,Fábio; Matheus, Feijó. "Desenvolvendo com C#". 2004.
- [Lop 2004] Lopes, Carlos Feijó. "Web Services: Metodologias de Desenvolvimento". 2004.
- [Orf 2000] Orfali,Robert; Harkey, Dan. "Cliente/Servidor – Guia essencial de Sobrevivência".Editora IBPI Press, 2001.
- [Pot 2003] Potts, stephen; Kopack,Mike. "Aprenda em 24 Horas – Web Services" .Editora Campus, 2003.
- [SOA] Simple object access protocol (soap).<http://www.w3.org/TR/SOAP>.
- [UDD] Universal description, discovery, and integration (uddi) [ww.uddi.org](http://www.uddi.org).
- [WSD] Web services description language (wsdl). www.w3.org/TR/WSDL.