

# WEB SERVICES: INTEGRAÇÃO E PADRONIZAÇÃO DE SERVIÇOS

Leonardo Parra Vigo<sup>1</sup>; Eliane Vendramini de Oliveira<sup>2</sup>; Adriane Cavichioli<sup>3</sup>, Elaine Parra Affonso<sup>4</sup>

## Resumo

Atualmente os sistemas de informação têm papel fundamental nas empresas, nos órgãos públicos e educacionais, sendo na maioria dos casos o principal instrumento ligado ao funcionamento adequado destas. Com o passar do tempo a tecnologia da informação criou força e está em constante evolução. Essa evolução trouxe realidades tecnológicas diferentes entre as empresas, criando um obstáculo para a integração de seus serviços. Toda empresa tem como requisito básico prover serviços, tanto para clientes como para outras empresas. Cada vez mais as empresas procuram o controle da informação e flexibilidade nos seus serviços, por isso, a integração de sistemas tornou-se uma das grandes prioridades das organizações. A padronização da comunicação entre os serviços de diferentes empresas traz inúmeros benefícios, dentre eles, agilizar a realização destes. Com o crescimento da Internet, houve também crescimento da variedade de serviços oferecidos na *web* e diversidade de sistemas coexistindo nas organizações, assim, grande parte das empresas, órgãos públicos e outras entidades aderiram ao uso dos *Web Services*. Os *Web Services* permitem que aplicações se comuniquem umas com as outras através da *web*. Este artigo apresenta um estudo sobre a tecnologia *Web Service*, mostrando sua finalidade, objetivando mostrar as vantagens de sua utilização dentro das empresas e organizações. A natureza da pesquisa é pura ou básica. Portanto, concluiu-se com a pesquisa realizada para este artigo que a tendência desta tecnologia é evoluir cada vez mais, já que está evoluindo juntamente com a Internet, e onde a interação de processos é uma tendência, esta tecnologia que é amplamente empregada, terá a sua utilização cada vez maior com o passar do tempo.

**Palavras-chave:** Padronização, integração, tecnologia.

## Abstract

Nowadays, Information Systems have had a fundamental importance for companies, public and educational agencies, being in the most of the cases, the main instrument for their proper operation. In the course of time, Information System acquired importance and is in a constant evolution process. This evolution is responsible for the technological differences among companies, causing an obstacle for their services integration. Every company has as a basic requirement, provide services in both clients and other companies. Increasingly, companies try to control information and adapt their services, therefore, systems integration became one of the main priorities for

<sup>1</sup> Graduado em Análise e Desenvolvimento de Sistemas pela Faculdade de Tecnologia de Presidente Prudente - FATEC. E-mail: leonardovigo@hotmail.com.

<sup>2</sup> Doutoranda em Engenharia Elétrica pela Universidade Estadual Paulista Júlio de Mesquita Filho - UNESP/Ilha Solteira, mestra pela instituição, professora da Faculdade de Tecnologia de Presidente Prudente - FATEC e das Faculdades Adamantinenses Integradas - FAI. E-mail: eliane.oliveira3@fatec.sp.gov.br.

<sup>3</sup> Mestra em Ciência da Computação pela Fundação Eurípides Soares da Rocha - UNIVEM/Marília, professora da Faculdade de Tecnologia de Presidente Prudente - FATEC. E-mail: adriane.cavichioli@fatec.sp.gov.br.

<sup>4</sup> Doutoranda em Ciência da Informação pela Universidade Estadual Paulista - UNESP/Marília, professora da Faculdade de Tecnologia de Presidente Prudente - FATEC. E-mail: elaine\_affonso@yahoo.com.br.

organizations. The equalizing process of communication among services have offered for different companies result in huge benefits, including its agility. The development of the Internet have provided the increase of Web services variety and the diversity of systems used for companies, this way, a great amount of companies, public agencies and institutions have adhered to the Web Services. Applications communication is allowed by the Web Services through the Web. This article brings a study about the Web Service technology, explaining its aim, intending to show the advantages of its usage for companies and institutions. The nature of this research is pure or basic, therefore, we conclude through it, that the tendency of this technology is increasingly evolving, because it is evolving side by side the Internet, and the place where process interaction is a tendency, this widely used technology will be used more and more in the course of time.

**Keywords:** Complex Thought; Integrated Education, Curriculum, Work.

## 1 Introdução

Nos últimos anos, com a evolução e padronização da Internet surgiu uma nova tecnologia que tem chamado a atenção de muitos analistas e desenvolvedores, o *Web Service*. Essa tecnologia é uma solução utilizada para suprir a necessidade de integração entre diferentes sistemas e permitir a comunicação entre aplicativos distintos. “Os *Web Services* permitem a comunicação entre aplicações com linguagens heterogêneas, fornecendo funções e serviços, comunicando-se através de uma linguagem padrão, o XML” (HANSEN, PINTO, *online*, 2003, p.2).

Com o crescimento da Internet, *Web Service* é mais que uma tendência, pois com o uso dele aplicações em diferentes locais podem enviar e receber dados, trazendo agilidade e eficiência nos processos das empresas que aderirem à tecnologia. Utilizando a tecnologia *Web Service*, uma aplicação pode invocar outra para efetuar tarefas simples ou complexas mesmo que as duas aplicações estejam em diferentes sistemas e escritas em linguagens diferentes.

“Os *Web Services* apresentam uma estrutura arquitetural que permite a comunicação entre aplicações. Um serviço pode ser invocado remotamente ou ser utilizado para compor um novo serviço juntamente com outros serviços.” (HANSEN, PINTO, *online*, 2003, p.1).

Como toda empresa tem como requisito básico prover serviços, tanto para clientes como para outras empresas, a padronização da comunicação entre os serviços de diferentes empresas agilizará a realização destes. Por exemplo, segundo Reckziegel (*online*, 2006), um sistema de reserva de passagens aéreas feito em Java e rodando em um servidor Linux pode acessar, com transparência, um serviço de reserva de hotel feito em .Net rodando em um servidor Microsoft.

No entanto, ainda é necessário que um conjunto de evoluções se consolide para se conseguir integrações excelentes para grandes corporações com um grande número de sistemas, como a consolidação de padrões de fato e melhoramento dos mecanismos de segurança, entre outros.

Com esse estudo, tem-se a oportunidade de penetrar em um ambiente tecnológico que está sendo muito explorado.

Justifica-se o estudo deste artigo, pois trata-se de um assunto de grande relevância no cenário atual das organizações, já que a integração entre os sistemas pode ser um fator que determina o aumento de produtividade e sucesso de uma empresa.

Portanto, o objetivo do estudo é analisar o papel dos *Web Services* e suas vantagens dentro das empresas; detalhar como surgiu e como ocorreu a evolução dessa tecnologia; relatar para qual finalidade os *Web Services* são usados atualmente e suas vantagens e potencial que possui; descrever quais empresas fazem uso dessa tecnologia e analisar as tendências dessa tecnologia.

## **2 A Integração de Sistemas**

A integração de Sistemas de Informação (SI) é uma área complexa que pode ser abordada de várias perspectivas diferentes, de acordo com o tipo de organização ou tecnologia empregada. Ela pode ajudar a organizar e aumentar a agilidade na tomada de decisões.

A integração de SI engloba três áreas importantes nomeadamente, a informação, os sistemas informáticos e os processos organizacionais, que devem ser coordenadas e ajustadas sob pena de a integração não se adequar à dinâmica da organização” (PEREIRA, *online*, 2009, p.15).

Cada vez mais as empresas procuram o controle da informação e flexibilidade nos seus serviços. Isso fez com que a integração de sistemas se tornasse uma das grandes prioridades organizacionais. A constante evolução das Tecnologias de Informação (TI) criou realidades tecnológicas diferentes provocando uma maior necessidade de compartilhar dados, informações e funcionalidades entre os sistemas.

Assim, a questão sobre tal integração tornou-se cada vez mais complexa e importante para as organizações, sendo um grande desafio quanto à sua adaptabilidade, implementação, manutenção, flexibilidade e gestão.

## 2.1 Histórico – Web Services

Com a evolução da Internet, o crescimento da variedade de serviços oferecidos na web e a diversidade de sistemas coexistindo nas empresas, surgiu a necessidade de integrar esses diferentes sistemas e de realizar a comunicação entre aplicativos distintos.

Essa necessidade fez com que certos padrões fossem propostos para realizarem tal integração, como o DCOM e CORBA, mas nenhum obteve êxito suficiente, tanto pela independência de plataforma como pelo modelo proposto. O padrão CORBA foi criado pela OMG (*Object Management Group*), um grupo formado por mais de 800 empresas da indústria da informática e roda em multiplataforma, já o DCOM foi criado pela Microsoft e só roda em ambiente Windows.

Além do DCOM e CORBA existem ainda outras tecnologias antecessoras aos *Web Services* que se mostram mais complexas e de manutenção mais difícil, como o EDI, RPC e RMI.

A tecnologia EDI (*Electronic Data Interchange*) é de difícil implementação, além de ser complexa e cara; CORBA (*Common Object Request Broker Architecture*) e DCOM (*Distributed Component Object Model*), além de competirem entre si, são difíceis de entenderem, tendo pouco suporte das companhias de *software*; RPC (*Remote Procedure Call*) ficou muito restrito ao uso dentro do universo Unix e Java RMI (*Remote Procedure Call*) possui suporte de poucos fornecedores. (DALL’OGLIO, *online*, 2005, p.6).

A expressão “*Web Services*” surgiu por volta de 1999, para substituir as tradicionais estratégias de EAI. O significado de EAI “Integração de Aplicações Corporativas” mudou com o tempo. Há alguns anos, o termo era usado exclusivamente para designar a tentativa de uma empresa interligar suas aplicações internas para que os dados pudessem ser compartilhados entre elas. Recentemente, o significado foi expandido para também englobar a união de dados e processos com outros parceiros comerciais externos.

*Web Services* pode ser descrita como uma “solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas

desenvolvidos em plataformas e linguagens diferentes sejam compatíveis” (OLIVEIRA, *online*, 2010).

Atualmente muitos sistemas legados estão sendo integrados aos novos sistemas através de *Web Services*.

Foi criado originalmente como uma interface padronizada baseada em tecnologias já conhecidas, como a especificação XML, utilizada para possibilitar a troca de mensagens entre os serviços, e o protocolo HTTP (*HyperText Transfer Protocol* – protocolo de transferência de hipertexto), normalmente utilizado para o transporte dos dados. Essa padronização foi desenvolvida pela W3C, que se trata de um consórcio internacional de empresas de tecnologia, órgãos governamentais e organizações independentes que tem como o objetivo desenvolver padrões para a criação e a interpretação de conteúdos para a *web*.

Grandes empresas como IBM, Microsoft e outras como a BEA Systems, concordaram em dar suporte às especificações de comunicações entre programas e dispositivos, através da integração dos seus sistemas.

A IBM e a Microsoft trabalharam na elaboração de uma forma para descrever e especificar como acessar os *Web Services*. A especificação desenvolvida por essas duas empresas foi fundida resultando na WSDL (*Web Services Description Language*). Após isso, IBM, Microsoft e Ariba iniciaram os trabalhos para elaborarem uma solução que fornecesse recursos para publicar os serviços *web* disponíveis. Assim, em 2000 foi anunciada a versão 1.0 do UDDI (*Universal Description, Discovery and Integration*).

De acordo com Jagiello e Perpétuo Júnior (2003) tecnologias e *frameworks* de desenvolvimento surgiram ultimamente, com o objetivo de uma maior integração entre os diversos aplicativos e serviços disponíveis na Internet. Este novo modelo em crescimento trata tarefas complexas, como o gerenciamento de transações, disponibilizando serviços distribuídos que utilizem interfaces de acesso simples e bem definidas.

*Web Service* é uma tecnologia emergente, sendo uma grande inovação advinda da Internet. Os *Web Services* permitem que aplicações se comuniquem umas com as outras, combinando suas funcionalidades de forma independente de plataforma ou linguagem.

Segundo Jagiello e Perpétuo Júnior (2003) os *Web Services* permitem a integração entre sistemas diferentes de forma simples e direta. Basicamente um *Web Service* funciona como uma página *web*, com a diferença que ao invés de HTML, utiliza-se

XML. Assim os dados podem ser descritos e o pacote da mensagem pode ser manipulado com grande facilidade por quem envia e por quem recebe.

## 2.2 Limitações da Tecnologia Web Services

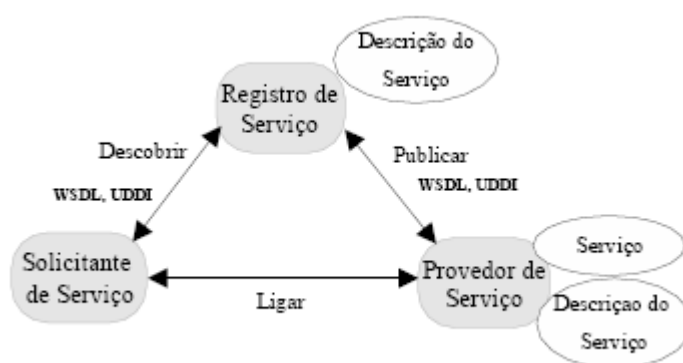
Muitas das limitações da tecnologia de *Web Services* estão relacionadas com o ambiente no qual devem operar, a Internet. Este ambiente pode ser de certa forma considerado hostil, pois trata-se de um ambiente aberto. Algumas das limitações estão relacionadas também com as tecnologias de Internet sobre as quais os *Web Services* estão fundamentados. Outras limitações são inerentes às próprias especificações elaboradas para os mesmos (ABINADER; LINS, 2006, p. 54).

Dentre as suas limitações pode-se citar sua descoberta e localização, a confiabilidade no ambiente de Internet, a sua segurança, escalabilidade, transações, desempenho e disponibilidade.

## 2.3 Principais Padrões nos quais os Web Services se Baseiam

Os *Web Services* são baseados em padrões e protocolos conhecidos ( HTTP, XML, SOAP, WSDL e UDDI).

Um *Web Service* pode ser definido como um serviço disponibilizado na Internet, que é descrito via WSDL, registrado via UDDI, acessado utilizando o protocolo SOAP e com os dados transmitidos em XML, sendo representado pela Figura 1 a seguir:



**Figura 1 - Web Services: papéis, operações e artefatos**  
Fonte: (HANSEN; PINTO, *online*, 2003)

A arquitetura de *Web Services* apresentada na Figura 1 está baseada nas interações entre um Provedor de Serviço para disponibilizar um serviço, um Solicitante de Serviço

que faz uso do serviço e um Registro de Serviço onde os provedores publicam as descrições dos serviços. (HANSEN; PINTO, *online*, 2003)

Os protocolos HTTP, SOAP, WSDL e UDDI e a linguagem XML serão descritos detalhadamente a seguir..

### **2.3.1 HTTP**

“O HTTP é o protocolo padrão da *web*. Ele é composto basicamente de dois momentos distintos: Um conjunto de solicitações do *browser* ao servidor e um conjunto de respostas que retornam do servidor ao *browser*.” (PINTO, 2002, p. 28). Ele utiliza o modelo cliente-servidor, como a maioria dos protocolos de rede, baseando-se no paradigma de requisição e resposta.

Seu uso para a obtenção de recursos interligados levou ao estabelecimento da *World Wide Web*. Foi desenvolvido para ser utilizado na Internet, embora seu desenho definitivo permita que ele possa ser usado em outras formas de aplicação.

Segundo Zavalik (2004, *online*, p. 31) o HTTP (*Hypertext Transfer Protocol*) é um protocolo de aplicação para sistemas de informação hipermídia, colaborativos e distribuídos. Ele pode ser utilizado para várias tarefas, além de ser utilizado para hipertexto, como sistemas de gerenciamento de objetos distribuídos e servidores de nomes.

Na arquitetura dos *Web Services*, o HTTP faz parte da camada de transporte. Eles trafegam sob HTTP utilizando os métodos *Post* e *Get*.

É o mais utilizado para a comunicação entre os *Web Services* devido a questões de segurança, já que normalmente esse protocolo não é bloqueado por *firewalls* em equipamentos conectados à Internet.

Outros protocolos podem realizar o transporte de mensagens, como o SMTP, Jabber e FTP.

### **2.3.2 XML**

Desde a sua especificação em 1998, o XML rapidamente adquiriu grande popularidade entre os peritos informáticos. XML (*Extensible Markup Language*) é uma recomendação da W3C, sendo uma linguagem de marcação especialmente criada para armazenar dados, e muito utilizada para transferir dados entre sistemas, projetada para permitir o uso do SGML, que trata-se de uma “metalinguagem através da qual se pode definir linguagens de marcação para documentos” (Pereira Filho, *online*, 2006).

Um dos fatores que determinaram o rápido avanço da XML foi o suporte garantido pelas grandes empresas do mercado do *software*, como a Microsoft, Oracle, IBM, SUN, entre outras.

É um dos pontos fortes da arquitetura dos *Web Services*, já que o “XML torna a representação dos dados independente de linguagem e de plataforma, de maneira que um dado descrito em um XML pode ser convertido para representações específicas para cada plataforma ou linguagem de programação.” (FREIRE, 2002).

O XML foi desenvolvido para estruturar, armazenar e enviar os dados das aplicações, suprimindo uma carência deixada pelo HTML no desenvolvimento de aplicações avançadas para a Internet, já que o HTML não foi concebido para manipular dados. O XML é um complemento ao HTML e não um substituto deste, pelo fato de o XML ter sido desenvolvido para descrever dados e focar no que os dados são e o HTML foi desenvolvido para mostrar dados e focar em como os dados aparecem.

Com o XML tornou-se fácil para sistemas de diferentes ambientes trocarem informações. Através dele programas escritos em linguagens diferentes e em plataformas distintas podem se comunicar de maneira ininterrupta.

Sistemas de computador e banco de dados, em geral, possuem dados em formatos incompatíveis, tornando-se um grande desafio para os desenvolvedores trocarem dados entre sistemas e a Internet. “Convertendo dados para XML pode-se reduzir em muito esta complexidade e criar dados que podem ser lidos por uma variedade de diferentes tipos de sistemas.” (JAGIELLO; PERPÉTUO JÚNIOR, 2003, p.23).

Segundo KÜHL (*online*, 2005, p.24) “um documento XML contém características especiais permitindo descrever o documento de forma inteligente, tornando o mais compreensível tanto para os seres humanos como para os computadores.” Um documento XML é composto por três elementos:

- Conteúdo dos dados: são as informações armazenadas entre as *tags*;
- Estrutura: é a organização dos elementos dentro do documento, que pode possuir diversos tipos de formato, como um memorando, um contrato, de acordo com as necessidades de marcação da informação;
- Apresentação: é a forma como as informações serão apresentadas ao leitor do documento, que pode ser de maneiras diferentes.

Um dos fatores que contribuiu para o aumento da utilização de *Web Services* é o uso da Linguagem XML na integração destes.



O XML como linguagem base dos *Web Services* permite a comunicação, troca de mensagens, entre as linguagens dos sistemas ou aplicações, mesmo que se diferem uma da outra.

### 2.3.3 SOAP

SOAP é um protocolo de comunicação projetado para invocar aplicações remotas através de RPC (Chamadas Remotas de Procedimento) ou trocas de mensagens, em um ambiente independente de plataforma e linguagem de programação. Esse padrão fornece um meio simples e poderoso para um componente de *software* invocar ações através do uso de interações de mensagens. É responsável pela troca de mensagens entre os *Web Services*.

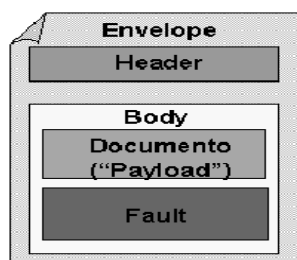
“É o protocolo padrão para transmissão de dados dentro da arquitetura de *Web Services* proposta pelo W3C” (GOMES, 2010, p.16). É simples, leve, baseado em XML e segue o modelo “*REQUEST-RESPONSE*” do HTTP.

Os maiores objetivos do SOAP são a simplicidade e extensibilidade. Ele obedece a esses requisitos e trafega sobre o HTTP que é suportado por todos os servidores e *browsers*, com diferentes tecnologias e linguagens de programação.

A estrutura de comunicação de SOAP consiste em:

- Modelo de processamento: modelo distribuído no qual uma mensagem enviada pode passar por nenhum ou vários intermediários. Esse modelo especifica como um receptor SOAP processa uma mensagem de forma isolada em relação a qualquer outra mensagem.
- Modelo de extensibilidade: “SOAP provê a extensão do ambiente de comunicação. Assim, as características omitidas para tornar o sistema mais simples podem ser estendidas, possibilitando a criação de novos modelos de comunicação” (SCHEPKE; SOUZA; VIANA, *online*, 2007).
- Estrutura de ligação do protocolo - Através desta é possível utilizar diferentes protocolos subjacentes. O conjunto de regras formais *Binding* (de vinculação) é que define a passagem de uma mensagem SOAP para um outro protocolo.
- Construção de mensagem: Define como deverá ser enviada uma mensagem SOAP através de um documento XML.

Uma mensagem SOAP consiste basicamente dos seguintes elementos: Envelope, *Header* e *Body*, como mostra a Figura 2 logo abaixo:



**Figura 2** - Estrutura de uma mensagem SOAP

Fonte: (CUNHA, *online*, 2002)

- **Envelope:** O envelope SOAP é o elemento mais importante. É o elemento raiz do SOAP que define o documento XML como uma mensagem SOAP. O envelope pode conter declarações de *namespaces* e também atributos adicionais como o *encoding style* (estilo de codificação) que define como os dados são representados no documento XML.
- **Header:** O elemento *header* nada mais é do que um cabeçalho. Ele contém informações específicas da aplicação, sobre a mensagem SOAP, como por exemplo, se a mensagem deve ser processada por um determinado nó intermediário, já que até alcançar o destino final a mensagem pode passar por diversos pontos intermediários. Quando utilizado, deve ser o primeiro elemento *child* (filho) do envelope.
- **Body:** Este elemento é obrigatório e contém a mensagem SOAP presente, planejada para a conclusão do “ponto final” da mensagem. O elemento *Body* pode conter um elemento opcional “*Fault*”, usado para indicar mensagens de *status* e erros retornadas ao processarem a mensagem. O conteúdo do corpo da mensagem SOAP depende se ela é uma requisição ou uma resposta. As Figuras 3 e 4 logo a seguir mostram um exemplo de uma requisição e uma resposta, respectivamente:

```
<soap:Envelope
  xmlns:xsi="Schema-Instance"
  xmlns:xsd="Schema"
  xmlns:soap="Envelope">
  <soap:Body>
    <Converte xmlns="http://conv.fractalti.com.br">
      <Valor>10</Valor>
      <De>DEC</De>
      <Para>BIN</Para>
    </Converte>
  </soap:Body>
</soap:Envelope>
```

**Figura 3** - Uma requisição SOAP convertendo de decimal para binário.

Fonte: (LEOPOLDO, *online*, 2003, p.3)

```

<soap:Envelope
  xmlns:xsi="Schema-Instance"
  xmlns:xsd="Schema"
  xmlns:soap="Envelope">
  <soap:Body>
    <ConverteResponse xmlns="http://site.com.br">
      <ValorResult>1010</ValorResult>
    </ConverteResponse>
  </soap:Body>
</soap:Envelope>

```

Figura 4. Uma resposta SOAP que retorna um valor convertido para binário.  
 Fonte: (LEOPOLDO, *online*, 2003, p.3)

### 2.3.4 WSDL

O WSDL (*Web Services Description Language*) é um protocolo baseado em XML e que tem como objetivo descrever detalhadamente um *Web Service*. Esse documento descreve o serviço, especifica como acessá-lo, quais as operações ou métodos disponíveis e define de forma clara como deve ser o formato de entrada e saída de cada operação.

De acordo com Reckziegel (*online*, 2006), “o seu principal objetivo é descrever as interfaces apresentadas e apontar a localização dos seus serviços, disponíveis em um local previsível e bem conhecido, na rede, o qual permite que o cliente acesse de maneira confiável.” Usando WSDL, um componente de *software* pode integrar qualquer função disponível no *Web Service*. Por ser um documento XML sua leitura se torna fácil e acessível.

O WSDL se difere das linguagens de definição de interfaces (IDL) anteriores como CORBA, DCOM, em termos de mecanismos de acesso e localização do serviço. Ele envolve duas partes diferentes: abstrata e concreta, como é mostrada na Figura 5:

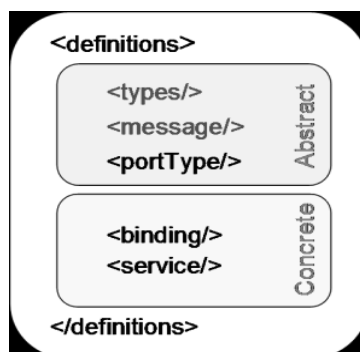


Figura 5. Estrutura WSDL  
 Fonte: (SPIES, *online*, 2008)

Elementos da estrutura WSDL:

- *types*: define os tipos de dados usados pelo *Web Service*. Utiliza a especificação W3C XML *Schema*.
- *message*: permite descrever as mensagens que são trocadas entre o serviço e o consumidor do serviço. Uma mensagem pode possuir várias partes, sendo que cada parte possui um nome e um tipo de dados. Define os dados trocados numa operação.
- *portType*: define um conjunto de operações suportadas pelo serviço e que possuem uma relação entre si. Existem quatro diferentes tipos de mensagens de operações: Operação unidirecional (o cliente envia uma mensagem ao serviço, sem resposta), solicitação-resposta (o cliente envia uma solicitação a um serviço, e recebe uma mensagem de resposta com o resultado dessa solicitação. É o tipo mais utilizado), pedido-resposta (o serviço envia uma mensagem ao cliente, resultando em uma mensagem enviada do cliente de volta para o serviço) e notificação (É quando o serviço envia uma mensagem, notificação para o cliente).
- *binding*: define o formato da mensagem e detalhes de protocolos.
- *service*: especifica a localização real onde o serviço está disponível e o *encoding* utilizado.

### 2.3.5 UDDI

Quando um *Web Service* é construído é necessário que os serviços sejam acessados por um cliente em algum lugar da Internet e, uma das maneiras é fazer com que a aplicação cliente conheça a URI do serviço.

UDDI (*Universal Description, Discovery and Integration*) é o protocolo desenvolvido para a organização e registro de *Web Services*. Ele serve para que as empresas integradas a um sistema *Web Service* registrem os serviços que oferecem para que outras possam utilizá-los. Foi escrito em 2000 e fundado pela IBM, Ariba e Microsoft, proposto originalmente como um núcleo padrão de *Web Service*. Possui três versões que demonstram a evolução da especificação. Novas preocupações e funcionalidades foram incorporadas no projeto a cada nova versão, principalmente para melhorar diversas questões, como: segurança, interação de serviços, relacionamento de negócios entre outros.

Esse mecanismo visa atender tanto ao cliente *Web Services* quanto ao provedor. O UDDI também possui a finalidade de armazenar arquivos WSDL.

O modelo de dados do UDDI é dividido em três partes distintas, onde cada uma possui objetivos característicos na especificação do registro:

- *White pages* (páginas brancas): Informações gerais de cada organização, como nome, endereço, telefone e outros. Essa parte da estrutura publica o *Web Service*.
- *Yellow pages* (páginas amarelas): Informações de organizações e de categoria de serviços que o caracteriza, tal como tipo do negócio, localização e produtos. Essa parte da estrutura descobre o *Web Service*.
- *Green pages* (páginas verdes): Informações técnica sobre serviços disponibilizados pelas organizações, tal como interagir com eles. Essa parte da estrutura liga o *Web Service*.

## 2.4 Arquitetura de um Web Service

A Figura 7 mostra os componentes envolvidos e a seqüência em que ocorre uma chamada de um *Web Service* SOAP, segundo Gomes (2010, p.15).

Os componentes exibidos na Figura 7 se integram e os números referenciados nessa figura serão utilizados para descrever a seqüência das operações, de acordo com Gomes (2010, p.18, p.19).

1. Registra e publica o *Web Service*: Os *Web Services* e os arquivos WSDL, que descrevem os *Web Services*, ficam armazenados em provedores para que eles sejam localizados por algum cliente através de seu URI. Assim o *Web Service* precisa ser registrado e publicado em um diretório de registros, também conhecido como UDDI.
2. Obtém informações sobre o *Web Services*: o desenvolvedor de *software* necessita ter a informação de onde está o *Web Service* desejado e seu respectivo arquivo WSDL, ou seja, ele precisa pesquisar em diretórios de *Web Services* (UDDI) pelo tipo de WS que deseja, assim o UDDI fornece ao desenvolvedor o endereço (URI) do *Web Service* e de seu WSDL.
3. Efetua *download* do WSDL: após obter os URIs e efetuar o *download* do WS e do arquivo WSDL, o desenvolvedor pode criar um *software* cliente para fazer uma chamada ao *Web Service* e obter uma resposta, de acordo com o serviço solicitado.
4. Envia solicitação XML: O *software* cliente faz chamadas ao *Web Service*, enviando-lhe solicitações no formato XML.

5. Recebe resposta XML: O *Web Service* recebe a solicitação, efetua o processamento e produz um resultado, que pode ou não ser enviado ao *software* solicitante como uma resposta, de acordo com o serviço solicitado, no formato XML.

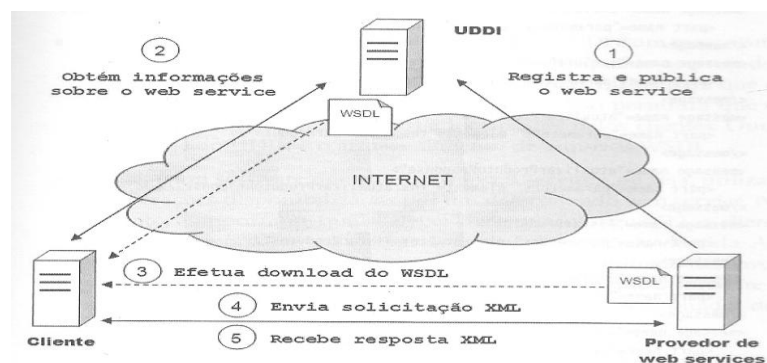


Figura 7 - Arquitetura para *Web Services* SOAP criada pela W3C.  
Fonte: (GOMES, 2010, p.15)

A arquitetura de *Web Services* se baseia na interação das entidades provedor do serviço, cliente do serviço e servidor de registro. O provedor de serviço disponibiliza um serviço, um cliente do serviço faz uso do serviço e o servidor de registro é onde os provedores publicam as descrições dos serviços.

Ao solicitar um serviço, o solicitante interage com o provedor do serviço, que possui a aplicação propriamente dita. Eventualmente, o solicitante poderá procurar informações em uma das agências de descobrimento de serviços. O provedor de serviços disponibiliza ao cliente somente a descrição dos serviços e o serviço propriamente dito. Detalhes particulares da aplicação não são revelados. O provedor de serviços pode, ainda, publicar os serviços que disponibiliza em agências de descobrimento de serviços. (ZAVALLIK, 2004, p. 29).

Um exemplo de prestador de serviço na Internet que fornece a descrição dos serviços disponibilizados em seu *site* de forma pública é o Google. Ele apenas exige que o usuário faça um cadastro.

Os *Web Services* permitem que a integração de sistemas seja realizada de maneira compreensível, reutilizável e padronizada. A arquitetura dos *Web Services* define uma interação entre solicitantes de serviços, trocador de mensagens e provedores de serviços.

Essa arquitetura representa um novo paradigma para as arquiteturas distribuídas, por consistir de componentes, funções (papéis) e operações.

### 3. Vantagens e Desvantagens dos Web Services

Os *Web Services* possuem diversos fatores e vantagens que faz dessa tecnologia um destaque referente à integração de sistemas. Dentre essas vantagens estão:

- Integração entre aplicações construídas em diferentes tecnologias: “Uma aplicação cliente de *Web Service* não precisa necessariamente estar usando a mesma tecnologia que o fornecedor de serviço” (ZAVALIK, 2004, p.35).
- São baseados nos protocolos HTTP, SOAP, WSDL e UDDI, e na linguagem XML.
- Extensibilidade: Conforme forem necessitadas novas funcionalidades, é possível que estas sejam acrescentadas pelo desenvolvedor.
- “Representa uma forma de programar aplicações distribuídas, aproveitando a infra-estrutura e a flexibilidade que a Internet oferece” (Moreira, *online*, 2005, p.28).
- Tecnologia precisa: O protocolo WSDL garante conformidade com os padrões estabelecidos entre os provedores e requisitantes de serviços.
- Permitem comunicação síncrona e assíncrona.
- Acessibilidade: Podem ser acessados a partir de qualquer local da Internet.
- Combinação de tecnologias: Possui a característica de conseguir combinar as melhores características da programação orientada a objetos com a programação para a *web*.
- Resultados compreensíveis: Apresentam como resultado um documento XML, que é de fácil entendimento para o ser humano, sendo consideravelmente menos complexo do que as tecnologias que o antecedeu.
- Elaboração segundo o modelo SOA: os *Web Services* são elaborados segundo o modelo de Arquitetura Orientada a Serviços onde as aplicações desenvolvidas podem ser publicadas como um serviço qualquer disponível na *web*.
- Menor custo de desenvolvimento: Por utilizarem normas abertas e serem independentes de plataforma e linguagem, ocorre uma grande reutilização de código.
- Desenvolvimento mais rápido de aplicações.

- Integração com empresas ou entidades externas que sejam parceiros da organização: a padronização da comunicação entre os serviços de diferentes empresas e organizações, agilizará a realização destes.

Quanto as suas desvantagens, além das relacionadas com a Internet, como a escalabilidade e transações, pode-se citar as relacionadas com a sua segurança. As mensagens que trafegam entre um fornecedor *Web Service* e seu cliente são transportadas em formato XML, usando o protocolo SOAP. Isto significa que estas informações não estão devidamente criptografadas e poderiam ser interceptadas por sistemas não autorizados. (ZAVALIK, 2004, p.36).

Para solucionar este problema estão sendo propostos novos padrões para a segurança das informações transportadas. De acordo com Jagiello e Perpétuo Júnior, os *Web Services* podem não apresentar um alto desempenho devido ao tamanho e à sobrecarga necessária para o tratamento dos documentos XML. Outra desvantagem apresentada é que os custos de integração e construção de *Web Services* nem sempre são baixos.

#### **4 O Uso dos Web Services nas Empresas**

De acordo com O'Brien (2003) a eficaz gestão da informação se tornou um dos pontos centrais, quase que decisivo, para o sucesso de uma organização. As empresas que conseguem disponibilizar informações relevantes e precisas em tempo hábil a seus funcionários, clientes e fornecedores, alavancam o seu poder de decisão, minimizam erros e, assim, têm maior potencial para satisfazer seus clientes.

Para as empresas o ideal é incorporar funcionalidades novas e implantar novos processos aos seus sistemas sem haver a necessidade de alterar drasticamente a estrutura de TI. A partir dessa idéia surgiram propostas de novas arquiteturas para que as empresas pudessem com facilidade interagir com funções, “partes” de sistemas, *softwares* que fossem de interesse à suas operações. A partir desse cenário surgiu o conceito de *Web Services*.

Segundo Westbridge (2004), em uma pesquisa realizada em 2004 entre as maiores empresas do mundo, 37% dos 273 entrevistados disseram que já possuíam sistemas em produção utilizando WS, e 44% estão em processo de pesquisa e desenvolvimento.

Atualmente várias empresas de grande e médio porte utilizam *Web Services* para diversas finalidades. Isso porque através do seu uso é possível atingir uma redução da



dependência em tecnologias proprietárias, uma flexibilização dos sistemas e uma melhor capacidade de cooperação entre empresas e entidades.

Outros benefícios que empresas possuem ao utilizar os *Web Services* é a redução dos custos de desenvolvimento e integração. Pode-se dizer que o principal objetivo desta tecnologia foi aproveitar os paradigmas da Internet para constituir e promover normas abertas de integração entre sistemas.

As parcerias usando *Web Services* são uma possibilidade que estão crescendo cada vez mais. Os serviços desenvolvidos ficam disponíveis para uma interação imediata, sem necessidade de acordos ou reuniões prévias entre os interessados.

Exemplo de grandes empresas que utilizam *Web Services*: Correios, Microsoft UDDI, Maplink e Amazon.

Essas empresas são de enorme expressão e cada uma delas utilizam *Web Services* para determinados fins. Os Correios utilizam para realizar consulta aos CEP's; a Microsoft UDDI utiliza para descobrir outros *Web Services*; a Maplink para disponibilizar mapas e rotas e a Amazon para comércio eletrônico.

Esta última, por exemplo, fornece um conjunto de *Web Services* muito completo, o "Amazon Web Services" disponível em: <<http://aws.amazon.com/>>, onde é possível criar uma loja virtual em parceria com a Amazon.com sem a necessidade de negociação. Esse conjunto de *Web Services* permite a esses possíveis parceiros analisar todas as informações de livros, DVD e CD, e qualquer produto vendido no seu *web site*.

## **5 Tendências da Tecnologia**

A Internet é um meio de comunicação poderoso que cresce a cada momento e que deve ser utilizado para a interação entre programas. Essa interação é efetuada pelos *Web Services*.

*Web Services* não é apenas uma tecnologia do momento, e sim uma tecnologia que está evoluindo com o decorrer do tempo e que surgiu no momento em que se busca uma maior integração de sistemas da informação e a adoção de padrões abertos para a comunicação entre estes.

Como a tendência futura é a automatização e interação dos processos, percebe-se que essa tecnologia, que hoje é amplamente empregada, será cada vez mais utilizada com o passar do tempo.

Segundo Girardi (*online*, 2004), pelo fato de suas troca de mensagens serem baseadas no padrão XML, ter flexibilidade quanto à linguagem de programação e por torná-las fracamente acopladas com as outras partes do código da aplicação, há uma grande motivação para a implementação de aplicações *web* baseadas em *Web Services*. Assim, as aplicações tornam-se flexíveis em relação às várias plataformas que estão disponíveis no mercado.

Um *Web Service* geralmente é implementado para disponibilizar uma determinada funcionalidade autocontida visando a reusabilidade do *Web Service*. Portanto, tende-se a possibilidade de um sistema utilizar um ou mais *Web Services* para realizar uma determinada transação, criando um esquema de composição de *Web Services*. Com a adoção dessa tecnologia, nota-se a tendência cada vez maior de criação de *Web Services* com o uso de agentes de *software* que podem se adaptar ao contexto de cada requisição e produzir resultados diferentes para cada situação.

## **6 Metodologia**

Esta pesquisa utilizou o método dedutivo, pois tem como finalidade explicar o conteúdo das premissas. A natureza da pesquisa é pura ou básica, já que objetiva apenas gerar conhecimentos úteis para o avanço da ciência sem aplicação prática prevista.

A forma de abordagem foi qualitativa, pois não tem a intenção de utilizar técnicas estatísticas para analisar os dados obtidos. A pesquisa pode ser classificada, quanto aos objetivos, como exploratória e descritiva. Quanto aos procedimentos a pesquisa é bibliográfica, desenvolvida a partir de livros e artigos. Foram utilizados livros, artigos científicos, base de dados virtual para a realização do trabalho, bem como computadores com acesso a Internet para a realização e conclusão da pesquisa aqui apresentada.

## **7 Resultados e Conclusão**

A tecnologia *Web Service* proporciona a integração de sistemas independente de sua linguagem e plataforma, baseados em padrões e protocolos conhecidos (HTTP, XML, SOAP, WSDL e UDDI).

Conforme visto neste artigo, os *Web Services* possuem a finalidade de integrar sistemas para que os mesmos possam combinar suas funcionalidades para a realização de serviços.

O setor de *Web Services* desenvolveu-se rapidamente, mas permanece em estágio de consolidação, onde um dos principais assuntos discutidos pelos desenvolvedores e analistas é a sua segurança, já que como o *Web Service* está em desenvolvimento acelerado, aplica-se isso também à segurança.

A arquitetura dos *Web Services* representa um novo paradigma para as arquiteturas distribuídas, por consistir de componentes, papéis e operações, permitindo que a integração de sistemas seja realizada de maneira compreensível e padronizada. Isso também é um dos fatores que proporcionou a sua grande aceitação no conceito de integração de sistemas.

Nota-se que entre as diversas vantagens dos *Web Services*, muitas estão relacionadas com a Internet. A acessibilidade, portabilidade e integração que esse ambiente possui, permite que as empresas possam utilizar e integrar serviços com outras organizações que possam estar do outro lado do planeta.

De acordo com Westbridge (2004), uma pesquisa realizada em 2004 mostrou que entre as maiores empresas do mundo, 37% dos 273 entrevistados disseram que já possuíam sistemas em produção utilizando WS, e 44% estavam em processo de pesquisa e desenvolvimento, o que mostra o crescimento da utilização dos *Web Services* por parte de grandes empresas e organizações, pois essas empresas observaram que essa tecnologia é a que possui maior independência de plataforma e linguagens, conseguindo assim ser a melhor solução para a integração de seus sistemas e serviços. Dentre essas grandes empresas que utilizam WS pode-se citar os Correios, Microsoft e Amazon.

A implementação dos *Web Services* obriga o desenvolvedor ter conhecimento e domínio sobre tecnologias, padrões e protocolos como HTTP, XML, SOA e WSDL. Isso dificulta o desenvolvimento de *Web Services* principalmente para pequenas e médias empresas localizadas em regiões onde o profissional apto para trabalhar com essa tecnologia é escasso.

As empresas que optam por utilizar os *Web Services* apresentam um grande potencial em melhorar a capacidade de colaboração com outras empresas e reduzir a complexidade de soluções distribuídas diminuindo custos de desenvolvimento e integração de sistemas.

Como proposta para trabalho futuro seria interessante elaborar uma pesquisa aprofundada sobre a segurança para *Web Services*.

## Referências

- ABINADER, J. A.; LINS, R. D, **Web Services em Java**. Rio de Janeiro: Brasport, 2006.
- ARROYO, M. **Trabalho e conhecimento: dilemas na educação do trabalhador**. São Paulo: Cortez, 2002.
- BRASIL, Ministério da Educação. **PROEJA–Documento Base**. MEC, SETEC: Brasília, 2007.
- CAMPOS, C. A. **Os desafios da implementação do currículo integrado no PROEJA em Rio Verde – GO**. Dissertação (PPGE) Faculdade de Educação, Universidade Federal de Goiás, 2010.
- CONFITEA. **Declaração de Hamburgo – Agenda para o futuro**. Brasília: Sesi\unesco, 1999. Disponível em <http://unesdoc.unesco.org/images/0012/001297/129773porb.pdf>> Acesso em Jan 2012.
- COSTA, M. A. **PROEJA: Desafios para a organização curricular na perspectiva da inclusão socioeducacional**. CEFET-MG: 2010. Disponível em: [http://www.senept.cefetmg.br/galerias/Anais\\_2010/Artigos/GT7/PROEJA\\_DESAFIOS.pdf](http://www.senept.cefetmg.br/galerias/Anais_2010/Artigos/GT7/PROEJA_DESAFIOS.pdf)> Acesso em Jan 2012.
- CUNHA, D. **Web Services, SOAP e Aplicações Web**. 2002. 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)> Acesso em: 26 out. 2010.
- DALL’OGLIO, P. **PHP 5 e Web Services**. 2005. Disponível em: <http://www.rau-tu.unicamp.br/nou-rau/softwarelivre/document/?code=154>>. Acesso em: 24, out. 2011.
- DI PIERRO, M. C.; JOIA, O.; RIBEIRO, V. M. **Visões da educação de jovens e adultos no Brasil**. Cad. Cedes, Ano XXI, n.55, Nov\ 2001. Disponível em: <http://www.scielo.br/pdf/ccedes/v21n55/5541.pdf>> Acesso em Jan 2012.
- DORE, R. **Gramsci e o debate sobre a escola pública no Brasil**. Cad. Cedes, Campinas, vol.26, n.70, p.329-352, set\dez.2006.
- FREIRE, H. Web Services: A nova arquitetura da internet. **Revista Developer’s Magazine**, n. 73, set. 2002. Disponível em: <http://www.cntt.com.br/files/webservices.pdf>. Acesso em: 20 set 2010.
- GIRARDI, R. A. D. **Framework para coordenação e mediação de Web Services modelados como Learning Objects para ambientes de aprendizado na Web**. 2004. Dissertação (Mestrado em Informática) – Pontifca Universidade Católica do Rio de Janeiro, Rio de Janeiro. Disponível em: [http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0220942\\_04\\_pretextual.pdf](http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0220942_04_pretextual.pdf)>. Acesso em: 05 de maio de 2011.
- GOMES, D. A. **Web Services SOAP em Java**. São Paulo: Novatec, 2010.

- HANSEN, R.; PINTO, S. C. **Construindo ambientes de educação baseada na web através de web services educacionais**. Canoas, 2003. Disponível em: <<http://www.nce.ufrj.br/sbie2003/publicacoes/paper07.pdf>>. Acesso em: 20 set. 2010.
- JAGIELLO, I. L.; PERPÉTUO JÚNIOR, E. **Web Services Uma Solução para aplicações distribuídas na Internet**. 2003. 58 f. Trabalho de conclusão de curso (Pós-graduação em Tecnologia para Sistemas de informação) – Pontifícia Universidade Católica do Paraná, Centro de Ciências Exatas e de Tecnologia, Curitiba.
- KÜHL, M. G. **Acesso Remoto de um Braço Eletrônico via Web Service**. 2005. 79 f. Trabalho de conclusão de curso (Especialização em Informática – Desenvolvimento em ambiente Internet.) - Centro Federal de Educação tecnológica do Paraná, Pato Branco. Disponível em: <<http://www.apostilando.com/download.php?cod=3104&categoria=XML>>. Acesso em: 19 mar. 2011.
- LEOPOLDO, M. R. B. **Simple Object Access Protocol - Entendendo o Simple Object Access Protocol (SOAP)**. 2003. Disponível em: <<http://www.msdnbrasil.com.br/secure/sharepedia/arquivos/SOAP.pdf>>. Acesso em: 21 mar. 2011.
- M. S. **Introdução aos processos siderúrgicos**. São Paulo: ABM, 2005.
- MOREIRA, A. F. B. Currículo, cultura e formação de professores. **Revista Educar**, Curitiba, n.17, p. 39-52. 2001. UFPR. Disponível em: <<http://educa.fcc.org.br/pdf/er/n17/n17a04.pdf>> Acesso em Jan 2012.
- MOREIRA, J. J. M. **WSQL: UMA ARQUITECTURA DE SOFTWARE BASEADA EM WEB SERVICES**. 2005. Disponível em: <http://repositorio-aberto.up.pt/bitstream/10216/11646/2/Texto%20integral.pdf> Acesso em: 04 mai. 2011.
- MORIN, E. **A cabeça bem-feita: repensar a reforma, reformar o pensamento**. Trad. Eloá Jacobina. 8 ed. - Rio de Janeiro: Bertrand Brasil, 2003
- MORIN, E. **Os sete saberes necessários à educação do futuro**. Trad. Catarina E. F. da Silva e Jeanne Sawaya. 2 ed. – São Paulo: Cortez, 2000.
- O'BRIEN, J. **Sistemas de Informação e as Decisões Gerenciais na Era da Internet**. São Paulo: Saraiva, 2003.
- OLIVEIRA, E. C.; CEZARINO, K. R. **Os sentidos do PROEJA: possibilidades e impasses na produção de um novo campo de conhecimento na formação de professores**. UFES, 2007. Disponível em: <<http://www.anped.org.br/reunioes/31ra/1trabalho/GT18-4782--Int.pdf>> Acesso Jan 2012.
- OLIVEIRA, E. M. **Definição do Web Service – Web Service sistema distribuído**. 2010. Disponível em: <<http://forum-gudo.forumeiros.com/t25-definicao-do-web-service-web-service-sistema-distribuido>>. Acesso em: 10 mar. 2010.
- PEREIRA FILHO, A. B. **XML Básico**. 2006. Disponível em <<http://www.abpsoft.com/criacaooweb/xmlbasico.html>>. Acesso em: 02 jan. de 2012.
- PEREIRA, R. C. **A norma WPS na integração do cadastro**. 2009. 119 f. Dissertação (Mestrado em Engenharia Informática e de Computadores) – Universidade Técnica de

- Lisboa, Lisboa. Disponível em:  
<https://dspace.ist.utl.pt/bitstream/2295/568320/1/dissertacao.pdf>. Acesso em: 13 mar. 2011.
- PINTO, G. **Estudos dos aspectos de segurança envolvidos no comércio eletrônico: Uma abordagem ao protocolo HTTPS**. 2002. 172 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Guarulhos, Guarulhos.
- RECKZIEGEL, M. **Descrevendo um Web Service – WSDL**. 2006. Disponível em:  
<[http://imasters.com.br/artigo/4422/webservices/descrevendo\\_um\\_web\\_service\\_-\\_wsdl/](http://imasters.com.br/artigo/4422/webservices/descrevendo_um_web_service_-_wsdl/)>. Acesso em: 7 mar. 2011.
- RECKZIEGEL, M. **Descrevendo, descobrindo e integrando Web Service – UDDI**. 2006. Disponível em:  
<[http://imasters.com.br/artigo/4474/webservices/descrevendo\\_descobrimo\\_e\\_integrando\\_o\\_web\\_services\\_-\\_uddi/](http://imasters.com.br/artigo/4474/webservices/descrevendo_descobrimo_e_integrando_o_web_services_-_uddi/)>. Acesso em: 7 mar. 2011.
- SANTOS, S. V. Possibilidades para a EJA, possibilidades para a educação profissional: o PROEJA. **La salle – Revista de Educação, Ciência e Cultura**, v.15, n.2, jul\dez 2010.
- SAVIANI, D. **Trabalho e educação: fundamentos ontológicos e históricos**. **Rev.Brasileira de Educação**, v.12, n.34 Jan\Abr 2007. Disponível em:  
<<http://www.scielo.br/pdf/rbedu/v12n34/a12v1234.pdf>> Acesso em Jan 2012.
- SCHEPKE, C.; SOUZA, S.; VIANA, V. **Avaliação de Desempenho de SOAP sobre HTTP, SMTP e BEEP**. Disponível em: <  
<http://www.inf.ufrgs.br/~cschepke/outras/errc07.pdf>>. Acesso em: 01 nov. 2010.
- SOUZA, A. M. **Currículo integrado e proeja: por uma outra educação do trabalhador**. CEFET – PA Disponível em:  
<[http://www.senept.cefetmg.br/galerias/Arquivos\\_senept/anais/terca\\_tema6/TerxaTema6Artigo8.pdf](http://www.senept.cefetmg.br/galerias/Arquivos_senept/anais/terca_tema6/TerxaTema6Artigo8.pdf)>  
Acesso em Jan 2012.
- SPIES, B. **Web Services, Part 2: WSDL and WADL**. 2008. Disponível em:  
<<http://www.ajaxonomy.com/2008/xml/web-services-part-2-wsdl-and-wadl>>. Acesso em: 22 mar. 2011.
- VERSIEUX, D. P. **O PROEJA e a formação integral dos trabalhadores**. CEFET-MG. Disponível em:  
<[http://www.senept.cefetmg.br/galerias/Arquivos\\_senept/anais/terca\\_tema6/TerxaTema6Artigo20.pdf](http://www.senept.cefetmg.br/galerias/Arquivos_senept/anais/terca_tema6/TerxaTema6Artigo20.pdf)>  
Acesso em Jan 2012.
- WESTBRIDGE Corp. **Web Service Usage Survey**. 2004. Disponível em <  
<http://www.westbridge.com/>>. Acesso em: 01 nov. 2010.
- ZABALA, A. **Enfoque globalizador e pensamento complexo**. Trad. Ernani Rosa. Porto Alegre-RS: Artmed,2002.
- ZAVALIK, C. **Integração de Sistemas de Informação através de Web Services**. 2004. 72 f. Dissertação (Mestrado em Informática) – Universidade Federal do Rio Grande do Sul, Instituto de Informática, Porto Alegre.