

UNIVERSIDADE SAGRADO CORAÇÃO

LEANDRO PERES GONÇALVES

**UTILIZAÇÃO DE SOAP PARA REALIZAR A
COMUNICAÇÃO ENTRE SERVIÇOS WEB**

BAURU
2012

LEANDRO PERES GONÇALVES

**UTILIZAÇÃO DE SOAP PARA REALIZAR A
COMUNICAÇÃO ENTRE SERVIÇOS WEB**

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. Ms. Anderson Francisco Talon.

BAURU
2012

Gonçalves, Leandro Peres

G635u

Utilização de SOAP para realizar a comunicação entre serviços web. / Leandro Peres Gonçalves -- 2012.
52f. : il.

Orientador: Prof. Me. Anderson Francisco Talon.

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

1. SOAP. 2. Serviços Web. 3. XML. I. Talon, Anderson Francisco. II. Título.

LEANDRO PERES GONÇALVES

**UTILIZAÇÃO DE SOAP PARA REALIZAR A COMUNICAÇÃO ENTRE
SERVIÇOS WEB**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade Sagrado Coração como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob orientação do Prof. Ms. Anderson Francisco Talon.

Banca examinadora:

Prof. Ms. Anderson Francisco Talon
Universidade Sagrado Coração

Prof. Ms. Márcio Henrique Castilho Cardim
Universidade Sagrado Coração

Prof. Esp. Henrique Pachioni Martins
Universidade Sagrado Coração

Bauru, 03 de dezembro de 2012.

DEDICATÓRIA

Primeiramente a Deus por me dar esta oportunidade, sempre me protegendo e me abençoando, afinal sem ele nada disto seria possível.

Aos meus pais, Antônio Roberto Gonçalves e Stela Maris Peres Pierini Gonçalves, a minha irmã Natália Peres Gonçalves Pavani, a meu cunhado Izac Pavani, a meu sobrinho Gabriel Gonçalves Pavani, a meu avô Antônio Peres Pierini (*in memoriam*), a minha avó Marlene Pereira Leite e a minha namorada Samara Roversi Trebejo, grandes incentivadores desta minha longa caminhada acadêmica.

Aos meus professores pela dedicação e paciência e principalmente ao meu orientador professor Anderson Francisco Talon, pela dedicação com que me orientou brilhantemente neste trabalho.

AGRADECIMENTOS

Ao Prof. Anderson Francisco Talon, pela confiança, compreensão e dedicação com que orientou meu trabalho.

Aos professores da USC pelos ensinamentos, dedicação e apoio.

A instituição de ensino superior, Universidade Sagrado Coração - USC pelo material de consumo, material didático e apoio técnico de alta qualidade que possibilitaram a execução deste trabalho.

Aos professores Henrique Pachioni Martins, Kelton Augusto Pontara da Costa e Márcio Henrique Castilho Cardim pelas sugestões e incentivo na elaboração do meu trabalho.

A todos os meus colegas de faculdade.

EPÍGRAFE

“Deus não escolhe os capacitados
capacita os escolhidos
Fazer ou não fazer algo
só depende de nossa vontade
e perseverança”.
(Albert Einstein)

RESUMO

Com o crescimento da internet, tornou-se necessário a utilização de tecnologias que conseguissem suprir as necessidades dos usuários em realizar tarefas múltiplas, que necessitavam que dois ou mais sistemas se comunicassem para se chegar a um único resultado. Os serviços web são um tipo exclusivo de serviço eletrônico que usa padrões abertos da Internet para a sua descrição, descoberta e invocação. Serviços Web podem ser vistos como aplicações, disponíveis na Internet, que podem ser usadas para executar determinados serviços eletrônicos. Eles podem ser desde simples serviços até serviços mais complexos. No contexto de serviços web, existem diversos padrões, entre eles, está o Simple Object Access Protocol (SOAP). O SOAP é um protocolo que define um mecanismo para a comunicação geral entre serviços web na Internet. Ele define o formato das mensagens (normalmente com conteúdo de dados XML) que são trocadas entre clientes de serviços, provedores de serviços e diretórios de serviços. Este trabalho demonstra que o SOAP é um protocolo que consegue realizar a comunicação entre aplicações distintas e distribuídas, e ainda permite que sistemas de diferentes plataformas se comuniquem com o objetivo de se ter uma informação relevante para o usuário final. A utilização deste protocolo só torna-se possível juntamente com outros padrões, dentre eles estão: XML (eXtensible Markup Language), WSDL (Web Service Definition Language) e UDDI (Universal Description Discovery and Integration), juntos, eles formam a arquitetura ideal para os serviços web e assim possibilitam a comunicação entre aplicações de diferentes plataformas.

Palavras-chave: SOAP. Serviços Web. XML.

ABSTRACT

With the growth of internet, it has become necessary to use technologies that they could meet the needs of users to perform multiple tasks, they needed two or more systems to communicate to arrive at a single result. Web services are a unique type of electronic service that uses open Internet standards for its description, discovery and invocation. Web Services can be seen as applications available on the Internet, which can be used to perform certain electronic services. They can range from simple services to more complex services. In the context of web services, there are several standards, among them is the Simple Object Access Protocol (SOAP). SOAP is a protocol that defines a general mechanism for communication between web services on the Internet. It defines the format of messages (usually XML data content) that are exchanged between service customers, service providers and service directories. This work demonstrates that SOAP is a protocol that accomplishes communication between different applications and distributed systems and even allows different platforms to communicate with the goal of having relevant information to the end user. Using this protocol only becomes possible with other patterns, among them are: XML (eXtensible Markup Language), WSDL (Web Service Definition Language) and UDDI (Universal Description Discovery and Integration), together they form the ideal architecture for web services and thus enable communication between applications on different platforms.

Keywords: SOAP. Web services. XML.

LISTA DE ILUSTRAÇÕES

Figura 1 - Comunicação entre serviços Web.....	15
Figura 2 - Mensagem SOAP.	19
Figura 3 - Código-fonte da sintaxe da mensagem SOAP.....	20
Figura 4 - Código-fonte do elemento envelope	21
Figura 5 - Código-fonte do elemento header.....	21
Figura 6 - Código-fonte do atributo <i>mustUnderstand</i>	22
Figura 7 - Código-fonte do atributo <i>actor</i>	22
Figura 8 - Código-fonte do elemento <i>body</i>	23
Figura 9 - Código-fonte da resposta SOAP.....	24
Figura 10 - Sub elementos do elemento <i>Fault</i>	24
Figura 11 - Códigos de identificação de erros do elemento <i>Fault</i>	25
Figura 12 - Exemplo de implementação do elemento <i>Fault</i>	25
Figura 13 – Diagrama de sequência do funcionamento do sistema.....	29
Figura 14 - Visualização do serviço disponibilizado pelo nuSoap.	30
Figura 15: Arquitetura dos servidores da aplicação	31
Figura 16: Tela de Login.....	32
Figura 17: Página inicial do sistema.....	33
Figura 18: Cadastro do Voo	34
Figura 19: Cadastro do Passageiro.....	35
Figura 20: Cadastro da Cidade	36
Figura 21: Consulta não disponível.	36
Figura 22: Consulta disponível.....	37
Figura 23: Requisição SOAP.....	37
Figura 24: Resposta SOAP	38
Figura 25: Erro de conexão.....	39
Figura 26: Realizando consulta.....	39
Figura 27: XML gerado pela consulta.....	40
Figura 28: XML gerado pela resposta	41
Figura 29: Tela exibida ao usuário após a chegada da resposta ao cliente.....	41

LISTA DE ABREVIATURAS E SIGLAS

- AOS** - Arquitetura Orientada a Serviços
- CORBA** - *Common Object Request Broker Architecture*
- COS** - Computação Orientada a Serviços
- CPF** - Cadastro de Pessoas Físicas
- DCOM** - *Distributed Component Object Model*
- DTD** - *Document Type Definition*
- JMS** - *Java Message Service*
- HTTP** - *Hypertext Transfer Protocol*
- MD5** - *Message-Digest algorithm 5*
- OASIS** - *Organization for the Advancement of Structured Information Standards*
- QoS** - *Quality of Service*
- SMTP** - *Simple Mail Transfer Protocol*
- SOAP** - *Simple Object Access Protocol*
- TCC** - Trabalho de Conclusão de Curso
- TCP** - *Transmission Control Protocol*
- TCP/IP** - *Transmission Control Protocol/Internet Protocol*
- UDDI** - *Universal Description, Discovery and Integration*
- URI** - *Uniform Resource Identifier*
- UTF-8** - *8-bit Unicode Transformation Format*
- W3C** - *World Wide Web Consortium*
- WSDL** - *Web Services Description Language*
- WWW** - *World Wide Web*
- XML** - *Extensible Markup Language*

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS	13
1.1.1	Objetivo geral	13
1.1.2	Objetivos específicos	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	SERVIÇOS WEB.....	14
2.2	PROTOCOLO SOAP.....	16
2.2.1	Vantagens do SOAP	17
2.2.2	Desvantagens do SOAP	18
2.2.3	Mensagem SOAP	19
2.2.4	Sintaxe da mensagem SOAP	19
2.2.5	O elemento envelope	20
2.2.6	O elemento Header	21
2.2.6.1	<i>O atributo mustUnderstand</i>	22
2.2.6.2	<i>O atributo Actor</i>	22
2.2.7	O elemento Body	23
2.2.8	O elemento <i>Fault</i>	24
2.2.9	Comunicação HTTP SOAP	25
2.2.10	Comunicação SOAP HTTP	26
3	PROTÓTIPO	29
4	RESULTADOS	32
5	CONSIDERAÇÕES FINAIS	42
	REFERÊNCIAS	44
	APÊNDICE A – ARTIGO NO FORMATO DA SBC	46
	APÊNDICE B – ARTIGO APRESENTADO NO FÓRUM DE INICIAÇÃO CIENTÍFICA DA USC	53

1 INTRODUÇÃO

Segundo o grupo OASIS, que é o responsável por diversas padronizações, a arquitetura orientada a serviço (AOS) “é um paradigma para organização e utilização de capacidades distribuídas que podem estar sob o controle de diferentes domínios de propriedade.” (OASIS, 2006, p. 8). Esse paradigma provê um meio regular para oferecer, descobrir e utilizar capacidades. Com isso, se promove a integração dos processos de negócio. Através de uma interface neutra e protocolos de transporte padronizados, AOS usa em sua construção o princípio de componentes de software fracamente acoplados chamados serviços. (SOUZA, 2006).

A computação orientada a serviços (COS) é o modelo da computação que utiliza serviços eletrônicos como elementos fundamentais para o desenvolvimento de aplicações distribuídas. (PAPAZOGLU; GEORGAKOPOULOS, 2003). Ela oferece uma infraestrutura tecnológica, a partir da qual organizações interessadas podem se comunicar, via troca de mensagens, para alcançar os objetivos de um processo de negócio.

No ano de 2000, a W3C (*World Wide Web Consortium*) aceitou a submissão do *Simple Object Access Protocol* (SOAP). Este formato de mensagem baseado em XML estabeleceu uma estrutura de transmissão para comunicação entre aplicações (ou entre serviços) via HTTP. Sendo uma tecnologia que independe do fornecedor, o SOAP disponibilizou uma alternativa atrativa em relação aos protocolos proprietários tradicionais, tais como CORBA e DCOM. (THOMAS, 2007).

SOAP significa Protocolo Simples de Acesso a Objetos, é um protocolo de comunicação baseado em XML que possibilita a comunicação de mensagens entre aplicações via HTTP, normalmente utilizado em WebServices. (W3schools..., c1999-2012). No contexto web, o SOAP torna-se uma ferramenta importante, pois este protocolo é capaz de realizar a comunicação entre sistemas externos e independe de linguagens de programação.

Este Trabalho de Conclusão de Curso demonstrou a importância da comunicação dentro do contexto de Arquitetura de Gestão de Processo de Negócios. Sabendo-se da complexidade de tal arquitetura, um dos grandes problemas atualmente é realizar uma comunicação de forma com que se tenham requisições e respostas totalmente em tempo real e com menos falhas possíveis. O

trabalho proporcionou um aprendizado sobre Serviços Web, além de fornecer um estudo completo sobre um protocolo de alta aceitação, como o SOAP.

Para a sociedade, este trabalho trouxe uma importante abordagem, que é a portabilidade de um sistema ou aplicação. Um sistema onde não se depende de plataformas de hardware ou software é de extrema aceitação no contexto da Tecnologia da Informação, e com os *Web Services* isto se torna totalmente possível, fazendo com que a sociedade tenha em suas mãos informações relevantes na hora em que precisar sem depender de uma plataforma específica. Para o meio acadêmico, o trabalho trouxe uma vasta área de pesquisa, que é a comunicação de aplicações que nem sempre são desenvolvidas nas mesmas plataformas e de alguma forma precisam se interligar para chegar a um resultado final.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Realizar a comunicação entre a organização consumidora, que é classificada como a responsável por consumir os serviços web da organização fornecedora, e a organização fornecedora, que é classificada como a responsável por fornecer os serviços web para a organização consumidora.

1.1.2 Objetivos específicos

- Estudar um protocolo de comunicação de alta aceitação no contexto web.
- Propiciar aos sistemas web uma comunicação amigável.
- Realizar a comunicação entre o cliente e o servidor utilizando o protocolo SOAP, através de troca de mensagens XML.
- Fazer a comunicação entre o cliente, que pode ser classificado como organização consumidora, e o fornecedor, que pode ser classificado como organização fornecedora.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção descreve os conceitos envolvidos na realização desse projeto.

2.1 SERVIÇOS WEB

De acordo com Alonso et al. (2010), um serviço web é um tipo exclusivo de serviço eletrônico que usa padrões abertos da Internet para a sua descrição, descoberta e invocação. Serviços Web podem ser vistos como aplicações, disponíveis na Internet, que podem ser usadas para executar determinados serviços eletrônicos. Eles podem ser desde simples serviços – como a realização de verificações de números de CPF, até serviços complexos, como aplicação de ações de cobranças. Segundo Menéndez (2002), um web service é uma aplicação que aceita solicitações de outros sistemas através de um meio específico chamado internet. Outra definição é a de que são componentes de software que oferecem uma funcionalidade única, podendo ser acessados por diferentes sistemas em diferentes contextos, através de padrões da internet, como HyperText Transfer Protocol (HTTP), eXtensible Markup Language (XML) e Simple Object Access Protocol (SOAP).

Conforme ilustra a Figura 1, existem alguns padrões básicos, que contém: uma linguagem para a descrição de serviços, conhecida como WSDL. Um repositório para armazenar as informações sobre empresas e serviços, conhecido como UDDI, e um protocolo de comunicação que permite a requisição dos serviços, conhecido com SOAP. Todos os padrões citados utilizam XML.

XML: De acordo com Bray et al. (2006) o XML (*eXtensible Markup Language*) é uma linguagem que permite a especificação de documentos contendo dados estruturados na forma de árvores. É uma recomendação da W3C. Um documento XML contém texto.

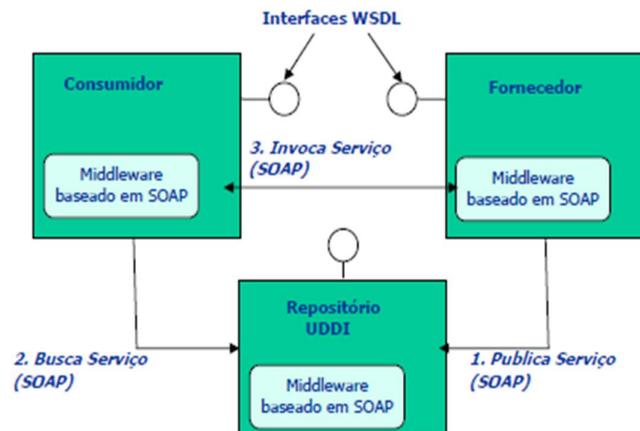


Figura 1 - Comunicação entre serviços Web.
Fonte: Medeiros, Gimenes e Toledo (2007).

A Figura 1 ilustra a comunicação entre serviços e os padrões utilizados. Fornecedores publicam serviços em repositórios. Consumidores buscam serviços nos repositório e os invocam de fornecedores. Conforme a etapa 1, um serviço SOAP é publicado no repositório UDDI. Na etapa 2, acontece a busca do serviço publicado pelo consumidor, que invoca o serviço ao fornecedor (etapa 3). Consumidor e Fornecedor se interagem através do protocolo SOAP, realizando assim a comunicação. Uma observação importante é que o Consumidor, Fornecedor e Repositório UDDI, todos, estão integrados por um *Middleware* baseado em SOAP, que nada mais é um mediador entre o software e as demais aplicações.

WSDL: De acordo com Chinnici et al. (2007) o WSDL (*Web Services Description Language*) é uma linguagem usada para descrever serviços Web. Uma descrição WSDL contém as informações relacionadas ao serviço Web necessárias para sua publicação, descoberta e invocação, incluindo capacidades, interface, comportamento e qualidade de serviços. Cada serviço web é descrito de forma independente por meio de: um tipo de porta (*port type*); operações (*operations*) que o tipo de porta oferece; e a estrutura de mensagens (*message*) de entrada e de saída.

UDDI: De acordo com Clement et al. (2004) o UDDI (*Universal Description, Discovery, and Integration*) é um padrão que define a estrutura e o conteúdo dos diretórios que contém descrições de serviços oferecidos. 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. Esse padrão permite também que clientes de serviços disponibilizem suas necessidades para que provedores de serviços busquem por usuários que precisam de um determinado serviço.

SOAP: De acordo com Miltra e Lafon (2007) o SOAP (*Simple Object Access Protocol*) é um protocolo que define um mecanismo para a comunicação geral entre serviços Web na Internet. Ele define o formato das mensagens (normalmente com conteúdo de dados XML) que são trocadas entre clientes de serviços, provedores de serviços e diretórios de serviços. De uma forma geral, o protocolo SOAP descreve como o protocolo HTTP (*HyperText Transfer Protocol*) pode ser usado para chamada de procedimento remoto na Internet, por meio de uma combinação de um cabeçalho HTTP com um corpo SOAP. Porém, outros protocolos de transporte também podem ser usados, como por exemplo, o SMTP (*Simple Mail Transfer Protocol*).

Os serviços Web atualmente estão sendo apresentados como uma tecnologia potencial para a efetiva automação das interações interorganizacionais na COS (Computação Orientada a Serviços), por facilitar a descoberta e a invocação automática de serviços eletrônicos relevantes. A tecnologia de serviços Web oferece uma infraestrutura apropriada para tornar processos de negócio acessíveis tanto em uma mesma organização quanto entre organizações. O benefício essencial associado à tecnologia de serviços Web é a ampla padronização, já que os padrões podem ser utilizados de forma simples em qualquer plataforma de software e hardware, tornando mais fácil a integração de aplicações. (ALONSO et al., 2004).

2.2 PROTOCOLO SOAP

SOAP é um protocolo superficial que exporta informação de forma centralizada em um ambiente distribuído. Ele é um protocolo baseado em XML que consiste de três partes: um envelope que define um framework para descrever o que é uma mensagem e como processá-la; um conjunto de regras de codificação para expressar instâncias de tipos definidos pela aplicação; e uma convenção para representar chamadas remotas de procedimento e respostas. (BOX, 2000).

O SOAP não está vinculado a uma plataforma de hardware, a um sistema operacional, linguagem de programação ou hardware de rede. Diferentemente de outros sistemas de computação distribuídos, o SOAP é construído no topo de padrões abertos, como HTTP e XML. A utilização de padrões amplamente aceitos e conhecidos facilita o aprendizado, por parte dos desenvolvedores, da nova tecnologia, e também da infraestrutura existente para suportá-la. (BELWOOD, 2002).

Segundo Leopoldo (2002), tira-se a conclusão de que o protocolo SOAP é definido como o elemento principal da infraestrutura dos *web services* e fundamental para o funcionamento dos mesmos independente de plataforma ou linguagem de programação específica.

2.2.1 Vantagens do SOAP

De acordo com Amorim (2004), o SOAP proporciona várias vantagens, em relação a outros sistemas distribuídos, entre elas, pode se citar:

- O SOAP pode atravessar firewalls, com facilidade;
- Os dados do SOAP são estruturados usando XML;
- O SOAP pode ser usado em combinação com vários protocolos de transporte, como HTTP, SMTP e JMS. O HTTP é o protocolo usado na WWW (*World Wide Web*) para a distribuição e recuperação de informação. Toda a troca de informações entre um browser e um servidor web é feita através desse protocolo. O protocolo SMTP controla a maneira como o e-mail é transportado e entregue via Internet ao servidor de destino. Já o protocolo JMS, é uma arquitetura existente dentro do universo J2EE que permite a troca de mensagens de maneira assíncrona.
- O SOAP mapeia satisfatoriamente para o padrão de solicitação/resposta (*request/response*) HTTP e do HTTP *Extension Framework*.
- O SOAP é razoavelmente superficial como um protocolo, pois contém menos recursos do que outros protocolos de computação distribuídos.
- Existe suporte para SOAP, por parte de vários fornecedores.

Devido à possibilidade do SOAP poder ser usado com o protocolo HTTP, ele pode transpor *firewalls* com facilidade, permitindo que aplicações estejam disponíveis internamente (através da intranet) e externamente (Internet) desde a sua distribuição inicial. Isto tem tudo para se tornar um problema sério para a segurança,

onde as aplicações do SOAP seriam acessíveis por partes não autorizadas. (AMORIM, 2004).

Ainda de acordo com o mesmo autor, outra vantagem proporcionada pelo SOAP é que as mensagens são transmitidas usando o protocolo XML, baseado em texto. Ou seja, as mensagens podem ser compreendidas por quase todas as plataformas de hardware, sistemas operacionais, linguagens de programação ou hardware de rede.

2.2.2 Desvantagens do SOAP

De acordo com Amorim (2004), devido à falta de recursos do protocolo SOAP, ele pode apresentar algumas desvantagens:

- Falta de interoperabilidade entre toolkits do SOAP;
- Mecanismos de segurança são imaturos;
- Não existe garantia quanto à entrega da mensagem;
- Não existe publicação nem assinatura.

Embora o SOAP possua um amplo suporte de vários fornecedores, ainda existem problemas de incompatibilidade entre diferentes implementações do SOAP. (AMORIM, 2004).

O SOAP não define um mecanismo para a autenticação de uma mensagem antes que ela seja processada. Também não define um mecanismo para a criptografia do conteúdo de uma mensagem SOAP, o que evitaria que outros tivessem acesso ao conteúdo da mensagem. (AMORIM, 2004).

Caso ocorra alguma falha enquanto uma mensagem estiver sendo transferida, um sistema que permita SOAP não saberá como reenviar a mensagem. (AMORIM, 2004).

Ainda de acordo com o mesmo autor, o cliente SOAP não pode enviar uma solicitação a vários servidores, sem enviar a solicitação a todos os servidores.

As aplicações permitem que o SOAP possa transpor os firewalls com facilidade, pois ele pode ser usado com o HTTP. Isto permite que os softwares que aceitam o SOAP estejam disponíveis internamente e externamente na rede. Esta característica pode ser vista como vantagem e também como desvantagem, já que

pode causar um sério problema de segurança, onde as aplicações do SOAP seriam acessíveis por partes não autorizadas. (HENDRICKS, 2002).

2.2.3 Mensagem SOAP

De acordo com W3s... (c1999-2012), a estrutura de uma mensagem SOAP é demonstrada a seguir.

Uma mensagem SOAP é um documento XML que pode conter três partes: o envelope SOAP, o cabeçalho SOAP e o corpo SOAP.

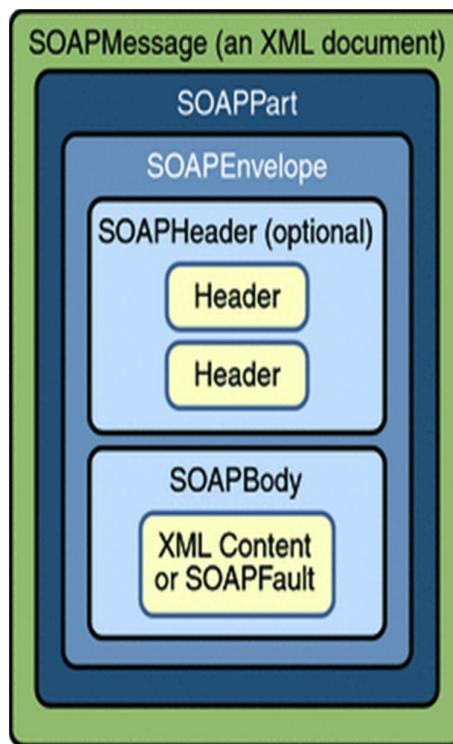


Figura 2 - Mensagem SOAP.
Fonte: W3s... (c1999-2012).

Conforme se ilustra na Figura 2, uma mensagem SOAP é um documento XML comum contendo um elemento chamado Envelope que identifica o documento XML como uma mensagem SOAP, um elemento Header que contém informações sobre o cabeçalho do documento, e um elemento *Body* que é o corpo do documento contendo informações de chamada e resposta, dentro do corpo contém um elemento *Fault* que contém erros e informações de status.

2.2.4 Sintaxe da mensagem SOAP

De acordo com W3s... (c1999-2012), a sintaxe de uma mensagem SOAP é bem simples e contém algumas regras:

- Uma mensagem SOAP deve ser codificada usando XML
- Uma mensagem SOAP usa o SOAP *Envelope namespace*. Os *namespaces* em um objeto (ou documento) XML identificam o tipo de dados contido no objeto. O *namespace* contém um prefixo e uma URI.
- Uma mensagem SOAP deve usar o SOAP *Encoding namespace* (Exemplo: UTF-8).
- Uma mensagem SOAP não contém referência DTD (*Document Type Definition*), que define as regras de formatação para uma dada classe de documentos.
- Uma mensagem SOAP não contém instruções de processamento XML

Conforme ilustra a Figura 3, um envelope SOAP dividido no cabeçalho (*Header*) e no corpo (*Body*), o cabeçalho é aberto na linha 4, e fechado na linha 6, e o mesmo é de uso opcional. Já o corpo é de uso obrigatório.

```

1 <SOAP-ENV:Envelope
2   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
3   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
4 <SOAP-ENV:Header>
5   ... //opcional
6 </SOAP-ENV:Header>
7 <SOAP-ENV:Body>
8   ... //obrigatório
9 </SOAP-ENV:Body>
10 </SOAP-ENV:Envelope>

```

Figura 3 - Código-fonte da sintaxe da mensagem SOAP.
Fonte: W3s... (c1999-2012).

2.2.5 O elemento envelope

De acordo com W3s... (c1999-2012), o elemento envelope é o elemento raiz da mensagem SOAP. Este elemento define o documento XML como uma mensagem SOAP.

Conforme ilustra a Figura 4, o elemento envelope contém dois atributos, na linha 3, o `xmlns:soap`, que define o *namespace* da mensagem SOAP, caso seja utilizado um *namespace* diferente será gerado um erro ao enviar o pacote e a

mensagem será descartada. Já na linha 4, o soap:encodingStyle define os tipos de dados utilizados no documento.

```

1  <?xml version="1.0"?>
2  <soap:Envelope
3      xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
4      soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
5      ...
6      ...
7  </soap:Envelope>

```

Figura 4 - Código-fonte do elemento envelope.
Fonte: W3s... (c1999-2012).

2.2.6 O elemento Header

De acordo com W3s... (c1999-2012), o elemento Header contém informações específicas do aplicativo da mensagem SOAP. É um elemento opcional, se ele estiver presente deve ser o primeiro elemento filho do elemento Envelope.

Conforme a Figura 5 é apresentada um cabeçalho SOAP. Na linha 2, é o elemento raiz do SOAP, e define que essa mensagem é uma mensagem SOAP, essa tag é fechada na linha 13. Na linha 6, o cabeçalho especifica informações específicas como autenticação e é opcional, essa tag é fechada na linha 10. Na linha 7, os atributos do cabeçalho definem como um destinatário deve processar uma mensagem SOAP, esta tag é fechada na linha 9.

```

1  <?xml version="1.0"?>
2  <soap:Envelope
3      xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
4      soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
5
6      <soap:Header>
7          <m:Trans xmlns:m="http://www.w3schools.com/transaction/"
8              soap:mustUnderstand="1">234
9          </m:Trans>
10     </soap:Header>
11     ...
12     ...
13 </soap:Envelope>

```

Figura 5 - Código-fonte do elemento header.
Fonte: W3s... (c1999-2012).

2.2.6.1 O atributo *mustUnderstand*

De acordo com W3s... (c1999-2012), o atributo *mustUnderstand* pode ser usado para indicar se uma entrada de cabeçalho é obrigatória ou opcional para o destinatário processar.

Na Figura 6 pode se observar que há o elemento *mustUnderstand*. Na linha 3, ele é referenciado. Se for adicionado *mustUnderstand* = "1" para um elemento filho do elemento *Header* indica que o receptor deve reconhecer o elemento ao processar o *Header*. Se o receptor não reconhece o elemento irá falhar ao processar o *Header*. O *mustUnderstand* aceita 0 ou 1.

```

1 <soap:Header>
2   <m:Trans xmlns:m="http://www.w3schools.com/transaction/"
3     soap:mustUnderstand="1">234
4   </m:Trans>
5 </soap:Header>

```

Figura 6 - Código-fonte do atributo *mustUnderstand*.
Fonte: W3s... (c1999-2012).

2.2.6.2 O atributo *Actor*

De acordo com W3s... (c1999-2012), o atributo *Actor* é demonstrado a seguir.

Na Figura 7 pode se observar que na linha 3 é apresentado o atributo *Actor*. Uma mensagem SOAP pode viajar do emissor ao receptor passando por diferentes pontos ao longo do caminho até seu último ponto. No entanto uma ou mais partes da mensagem não são destinadas ao último ponto. O Atributo *Actor* é usado pelo *Header* para especificar este ponto.

```

1 <soap:Header>
2   <m:Trans xmlns:m="http://www.w3schools.com/transaction/"
3     soap:actor="http://www.w3schools.com/appml/">234
4   </m:Trans>
5 </soap:Header>

```

Figura 7 - Código-fonte do atributo *actor*.
Fonte: W3s... (c1999-2012).

2.2.7 O elemento Body

De acordo com W3s... (c1999-2012), o elemento *Body* contém a mensagem SOAP pretendida que o usuário espera. Elementos filhos do elemento *Body* pode conter *namespace*, que em um objeto (ou documento) XML identificam o tipo de dados contido no objeto.

Na Figura 8 pode se observar que seria realizado como uma requisição o preço de uma maçã.

Observando, é interpretado como se fosse a chamada de um método, onde tem uma tag `m:getPrice` que seria o nome do método, e estaria passando um parâmetro `m:Item` que é a "Apples".

```

1  <?xml version="1.0"?>
2  <soap:Envelope
3      xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
4      soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
5
6      <soap:Body>
7          <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
8              <m:Item>Apples</m:Item>
9          </m:GetPrice>
10         </soap:Body>
11
12    </soap:Envelope>

```

Figura 8 - Código-fonte do elemento *body*.
Fonte: W3s... (c1999-2012).

Na Figura 9 pode se observar que é retornada uma resposta do mesmo exemplo ilustrado na Figura 8. Na linha 8 da Figura 8, contém o nome do Item a ser atribuído a requisição. Na Figura 9, na linha 8, após recebido uma requisição, foi retornado como resposta o preço da maçã, que no caso seria 1.90.

```

1  <?xml version="1.0"?>
2  <soap:Envelope
3      xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
4      soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
5
6      <soap:Body>
7          <m:GetPriceResponse xmlns:m="http://www.w3schools.com/prices">
8              <m:Price>1.90</m:Price>
9          </m:GetPriceResponse>
10         </soap:Body>
11
12    </soap:Envelope>

```

Figura 9 - Código-fonte da resposta SOAP.
Fonte: W3s... (c1999-2012).

2.2.8 O elemento *Fault*

De acordo com W3s... (c1999-2012), o elemento *Fault* do SOAP é o elemento de falha aonde possui erros e informações de status de uma mensagem SOAP. Este elemento é opcional e quando estiver presente deve aparecer como um elemento filho do elemento *Body*. Ele pode conter sub elementos como é apresentado na Figura 10:

Sub Elemento	Descrição
Faultcode	Código de identificação de erro
Faultstring	Explicação legível da falha
Faultactor	Informações sobre o que pode ter provocado a falha
Detail	Informações específicas sobre o erro

Figura 10 - Sub elementos do elemento *Fault*.
Fonte: W3s... (c1999-2012).

Pode se observar que a principal função do elemento *fault* é informar os status de erros, cada um destes tendo sua função e definição específica.

Na Figura 11, são apresentados os códigos de identificação de erro:

Erro	Descrição
VersionMismatch	Encontrou um namespace inválido para o elemento envelope SOAP
MustUnderstand	Um elemento filho imediato do elemento de Header, com o atributo mustUnderstand definido como "1", não foi compreendido
Cliente	A mensagem foi formada incorretamente ou continham informações incorretas.
Servidor	Houve um problema com o servidor.

Figura 11 - Códigos de identificação de erros do elemento *Fault*.
Fonte: W3s... (c1999-2012).

Na Figura 12, é apresentado um exemplo de implementação do elemento *Fault*.

```

1  <?xml version="1.0"?>
2  <soap:Envelope
3      xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
4      soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
5
6      <soap:Body>
7          ...
8          <soap:Fault>
9              <faultcode>soap:Client</faultcode>
10             <faultstring>
11                 Failed to locate method (ValidateCreditCard) in class
12                 (examplesCreditCard) at /usr/local/ActivePerl-5.6/lib/
13                 site_perl/5.6.0/SOAP/Lite.pm line 1555.
14             </faultstring>
15         </soap:Fault>
16     </soap:Body>
17
18 </soap:Envelope>

```

Figura 12 - Exemplo de implementação do elemento *Fault*.
Fonte: W3s... (c1999-2012).

2.2.9 Comunicação HTTP SOAP

De acordo com W3s... (c1999-2012), o HTTP se comunica através de TCP/IP. Um cliente HTTP conecta a um servidor usando TCP. Depois que a conexão foi estabelecida o cliente pode enviar uma requisição http para o servidor como:

```
1 POST /item HTTP/1.1
```

```

2 Host: 189.123.255.239
3 Content-Type: text/plain
4 Content-Length: 200

```

O servidor processa a requisição e envia uma resposta HTTP para o cliente. A resposta possui um código de status para indicar falhas ou sucesso.

A seguir, o exemplo de uma requisição que retornou com sucesso.

```

1 200 OK
2 Content-Type: text/plain
3 Content-Length: 200

```

Caso a requisição retorne uma falha, irá retornar algo como:

```

1 400 Bad Request
2 Content-Length: 0

```

2.2.10 Comunicação SOAP HTTP

De acordo com W3s... (c1999-2012), um método SOAP é uma requisição/resposta HTTP que está em conformidade com as regras de codificação SOAP especificadas pela W3C.

Ou seja: HTTP+XML = SOAP.

Content-Type

Define o tipo MIME TYPE da mensagem e sua codificação (CHARSET) para o corpo do XML usado na requisição e na resposta.

Content-Length

Especifica o número de bytes no corpo da requisição e resposta.

A seguir é mostrado um exemplo de comunicação SOAP.

Solicitação:

```

1 POST /InStock HTTP/1.1
2 Host: www.example.org
3 Content-Type: application/soap+xml; charset=utf-8
4 Content-Length: nnn
5 <?xml version="1.0"?>
6 <soap:Envelope
7 xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
8 soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

```

```

9 <soap:Body xmlns:m="http://www.example.org/stock">
10 <m:GetStockPrice>
11 <m:StockName>IBM</m:StockName>
12 </m:GetStockPrice>
13 </soap:Body>
14 </soap:Envelope>

```

A solicitação indica diversas características. Dentre elas o tipo de requisição, no caso do exemplo apresentado POST. O host de partida da solicitação, que é a URL de origem. O cabeçalho indicando a codificação e o tipo da aplicação. O Content-Length que obtém o tamanho da resposta, ou seja, a quantidade de bytes contidos na mesma.

Resposta:

```

1 HTTP/1.1 200 OK
2 Content-Type: application/soap+xml; charset=utf-8
3 Content-Length: nnn
<?xml version="1.0"?>
5 <soap:Envelope
6 xmlns:soap=http://www.w3.org/2001/12/soap-envelope
7 soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
8 <soap:Body xmlns:m="http://www.example.org/stock">
9 <m:GetStockPriceResponse>
10 <m:Price>34.5</m:Price>
11 </m:GetStockPriceResponse>
12 </soap:Body>
13 </soap:Envelope>

```

2.2.11 Tipos de Implementação de web services na linguagem PHP

Existem várias maneiras de se implementar web services utilizando a linguagem de programação PHP. De acordo com Souza (2005), é descrito algumas possibilidades baseadas em código livre:

- XML-RPC: é uma implementação de chamadas de procedimentos remotos (RPC) que possibilita o transporte de dados no formato XML entre dois servidores utilizando o protocolo HTTP;

- PEARL SOAP Client/Server for PHP: o PHP *Extension and Application Repository* (PEAR) é um repositório de classes PHP de licença livre que tem uma grande comunicação de colaboradores. Ele possui uma implementação do protocolo SOAP e seus serviços.
- nuSoap: é uma API desenvolvida em PHP que permite a criação de clientes e servidores web services, possui suporte embutido à WSDL e facilidade para a instalação e utilização.
- *Representational State Transfer* (REST): o REST não possui um padrão de implementação e não oferece classes pré-construídas. Utiliza o protocolo HTTP para o envio e recebimento de mensagens no padrão XML. Para realizar a troca de mensagens XML, utilizam-se os métodos padrões do HTTP: *GET*, *POST* e *PUT*. Na sua implementação torna-se necessário o uso de outras ferramentas como PHP *Document Object Model* (DOM), *Simple API for XML* (SAX) ou também *eXtensible Stylesheet Language* (XSL) para realizar o *parsing*;
- PHP SOAP *Extension*: este é a primeira implementação nativa do protocolo SOAP incorporada na versão 5 do PHP. Sua principal vantagem entre as outras implementações escritas em PHP é a velocidade. Pode ser usada para implementar clientes e servidores e tem uma série de funções pré-definidas.

3 PROTÓTIPO

O protótipo desenvolvido é um sistema que realiza consultas a uma base remota contendo os dados de passagens aéreas, através do protocolo de comunicação SOAP. Desta forma, o *web service* possibilita que qualquer aplicação com suporte ao protocolo de comunicação SOAP utilize o método que retorna as informações das passagens aéreas, que foi desenvolvido neste trabalho.

Conforme ilustrado na Figura 13, o usuário, que é representado pelo navegador (browser), envia os dados para a realização da consulta, no caso, é preciso fornecer a cidade e a data em que se deseja pesquisar informações relacionadas às passagens aéreas. Após esta consulta, o Cliente, que é representado pela Agência de Viagem, recebe a consulta do Usuário e faz uma requisição ao Fornecedor, que é representado pela companhia aérea. A companhia processa a requisição e gera uma resposta, que é enviada para o Cliente, que por fim envia para o usuário e exibe os dados retornados através do SOAP para a visualização no browser.



Figura 13 – Diagrama de sequência do funcionamento do sistema.
Fonte: Elaborado pelo autor.

Neste trabalho de conclusão de curso (TCC), foi utilizado a biblioteca de classes nuSoap. O nuSoap conta com diversos recursos e uma facilidade na compreensão, além de fornecer ao desenvolvedor fácil entendimento tanto na publicação do serviço quanto no consumo do mesmo, e isto ajudou na escolha desta

ferramenta em específico. O uso desta API proporcionou um ágil desenvolvimento do sistema, já que ela contém métodos pré-definidos que proporcionaram: a chamada ao fornecedor de serviços, a criação do arquivo WSDL, a publicação deste WSDL no repositório de serviços (UDDI) e a comunicação geral entre Cliente / Fornecedor (SOAP).

De acordo com a Figura 14, o nuSoap fornece a visualização do serviço de uma forma ágil, através do acesso pela url onde se encontra o servidor. Neste caso, a url é acessada por: `http://192.168.56.102:8080/tcc/server.php?wsdl`, onde 192.168.56.102 é o IP da máquina externa, 8080 a porta em que está rodando o servidor web, e `tcc/server.php?wsdl` é o caminho da aplicação. Na Figura 14, o nome do serviço é `getVoo`, e clicando sobre este é exibida as informações gerais do WSDL.

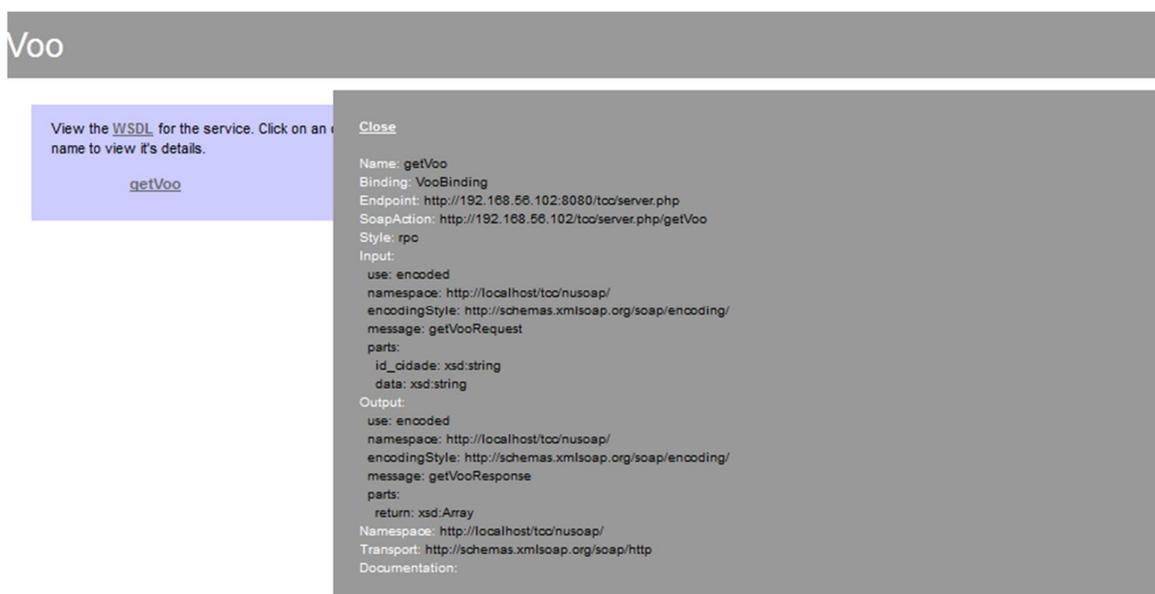


Figura 14 - Visualização do serviço disponibilizado pelo nuSoap.

Fonte: Elaborado pelo autor.

Para demonstrar o funcionamento do SOAP entre sistemas distintos, as aplicações foram separadas em duas arquiteturas de servidores diferentes: Windows e Linux.

De acordo com a Figura 15, é apresentada a arquitetura cliente/servidor utilizada neste Trabalho de Conclusão de Curso (TCC). A aplicação cliente, aonde se localiza o usuário juntamente com o browser, está na plataforma Windows. Já a aplicação servidor (fornecedor), está na plataforma Linux (Ubuntu), juntamente com

o banco de dados em Mysql, onde a aplicação cliente (IP: 127.0.0.1) está conectada remotamente ao banco de dados da aplicação servidor (IP: 192.168.56.102).



Figura 15: Arquitetura dos servidores da aplicação
Fonte: Elaborado pelo autor.

4 RESULTADOS

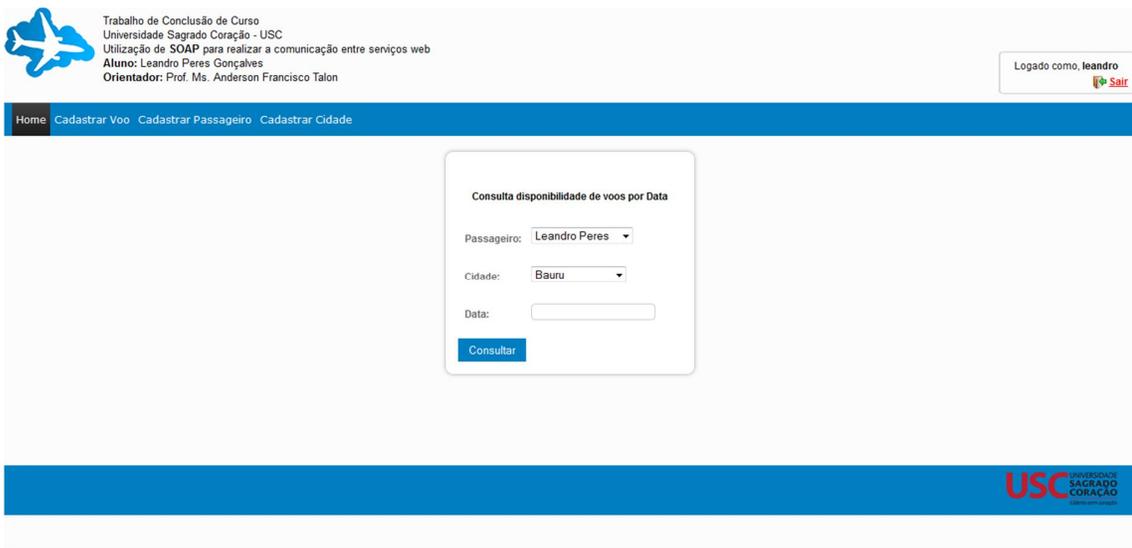
Os resultados deste Trabalho de Conclusão de Curso (TCC) são descritos a seguir.

De acordo com a Figura 16, é apresentada a tela de login. Esta tela tem como objetivo restringir o acesso de qualquer usuário ao sistema. O acesso somente é autorizado aos usuários devidamente cadastrados no sistema. A senha é armazenada em md5. O MD5 (Message-Digest algorithm 5) é um algoritmo de hash de 128 bits unidirecional. Para acessar, o administrador do sistema deve fornecer o seu usuário e a sua senha, após isto clicar no botão Entrar. Caso os seus dados estejam corretos, ele é redirecionado para a página de administração do sistema. Caso os seus dados estejam incorretos, ele volta para a tela de login e exibido a mensagem “Usuário e/ou senha inválidos”.



Figura 16: Tela de Login.
Fonte: Elaborado pelo autor.

De acordo com a Figura 17, é apresentada a tela inicial do sistema. Esta tela tem como objetivo permitir a pessoa autorizada realizar a consulta no sistema da agência de viagens, sendo necessário informar o nome do passageiro, a cidade e a data do voo. O nome do passageiro é uma informação fictícia, somente para demonstrar que caso fosse um sistema aonde existisse uma base enorme com diversos passageiros cadastrados, pudesse escolher entre eles. Já a cidade, é o campo responsável por identificar a cidade de destino que se quer consultar. A data é o campo responsável por informar ao serviço web o dia exato da viagem em que se deseja consultar. A partir disto, o sistema realiza a consulta na base externa do serviço web e retorna as informações solicitadas pelo usuário (browser).



The screenshot shows the initial page of a system. At the top left, there is a logo of an airplane and text: "Trabalho de Conclusão de Curso Universidade Sagrado Coração - USC Utilização de SOAP para realizar a comunicação entre serviços web Aluno: Leandro Peres Gonçalves Orientador: Prof. Ms. Anderson Francisco Talon". At the top right, there is a login status: "Logado como, leandro" with a "Sair" button. Below this is a navigation bar with links: "Home", "Cadastrar Voo", "Cadastrar Passageiro", and "Cadastrar Cidade". The main content area features a form titled "Consulta disponibilidade de voos por Data". The form has three fields: "Passageiro:" with a dropdown menu showing "Leandro Peres", "Cidade:" with a dropdown menu showing "Bauru", and "Data:" with an empty text input field. Below the fields is a blue "Consultar" button. At the bottom right, there is a logo for "USC UNIVERSIDADE SAGRADO CORAÇÃO" with the text "1954" below it.

Figura 17: Página inicial do sistema.
Fonte: Elaborado pelo autor.

De acordo com a Figura 18, é apresentada a tela de cadastro do voo. Esta tela tem como objetivo realizar o cadastro do voo. Para o cadastro do voo, é necessário informar a Cidade, a Data, o horário do voo, o preço e a tarifa (aqui podem ser inseridas a taxa de embarques e outras tarifas que venha a ter aquela viagem). Após o cadastro, o serviço web armazena essas informações no banco de dados e a próxima vez que for realizado uma consulta pela data especificada no cadastro e cidade, o sistema retorna aquelas informações para o usuário permitindo

que ele possa visualizar a data do voo, o horário e o preço total (soma do preço e tarifa).

The screenshot shows a web application interface for flight registration. At the top left, there is a logo of a blue airplane and text: "Trabalho de Conclusão de Curso", "Universidade Sagrado Coração - USC", "Utilização de SOAP para realizar a comunicação entre serviços web", "Aluno: Leandro Peres Gonçalves", and "Orientador: Prof. Ms. Anderson Francisco Talon". At the top right, there is a login status: "Logado como, leandro" with a "Sair" button. Below this is a navigation bar with links: "Home", "Cadastrar Voo", "Cadastrar Passageiro", and "Cadastrar Cidade". The main content area features a "Cadastrar Voo" form with the following fields: "Cidade:" with a dropdown menu showing "Bauru", "Data:", "Horario:", "Preço:" with a value of "0.00", and "Tarifa:" with a value of "0.00". A blue "Cadastrar" button is located at the bottom of the form. At the bottom right of the page, there is the USC logo: "UNIVERSIDADE SAGRADO CORAÇÃO" and "UNIVERSITY OF THE SACRAMENT".

Figura 18: Cadastro do Voo.

Fonte: Elaborado pelo autor.

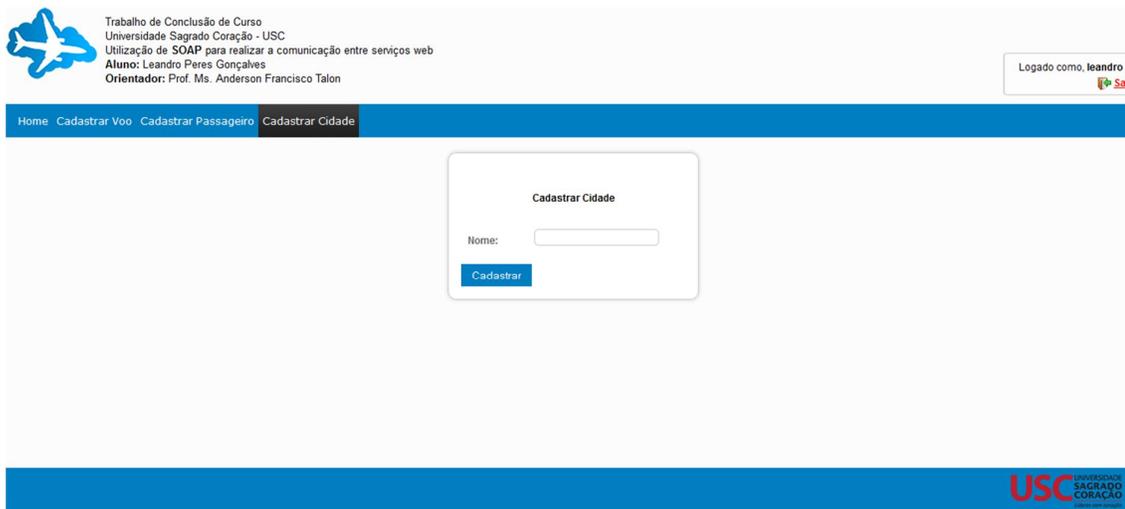
De acordo com a Figura 19, é apresentada a tela de cadastro do passageiro. Esta tela tem como objetivo realizar o cadastro do passageiro no serviço web. O passageiro é um dado fictício que não influencia na consulta, porém pensando em um sistema maior poderia existir uma base com diversos passageiros, e estes através de seus cadastros poderiam realizar uma consulta exclusiva em sua conta e após a consulta já ser feito o pagamento e agendamento do voo.

The screenshot shows a web application interface. At the top left, there is a logo of an airplane and the following text: "Trabalho de Conclusão de Curso", "Universidade Sagrado Coração - USC", "Utilização de SOAP para realizar a comunicação entre serviços web", "Aluno: Leandro Petes Gonçalves", and "Orientador: Prof. Ms. Anderson Francisco Talon". At the top right, there is a login status: "Logado como leandro" with a "Sair" button. Below this is a navigation bar with three items: "Home", "Cadastrar Voo", and "Cadastrar Passageiro" (which is highlighted). The main content area features a form titled "Cadastrar Passageiro" with a "Nome:" label and an input field, and a "Cadastrar" button. At the bottom right, there is a logo for "USC UNIVERSIDADE SAGRADO CORAÇÃO" with the tagline "Fazer com paixão".

Figura 19: Cadastro do Passageiro.

Fonte: Elaborado pelo autor.

De acordo com a Figura 20, é apresentada a tela de cadastro da cidade. Esta tela tem como objetivo realizar o cadastro da cidade no serviço web. A cidade é um fator importante, pois é através dela que o sistema pode saber a data exata em relação à cidade que o usuário deseja realizar a sua viagem. Como o objetivo deste Trabalho de Conclusão de Curso era realizar a comunicação SOAP, não foi separada a cidade por estado.

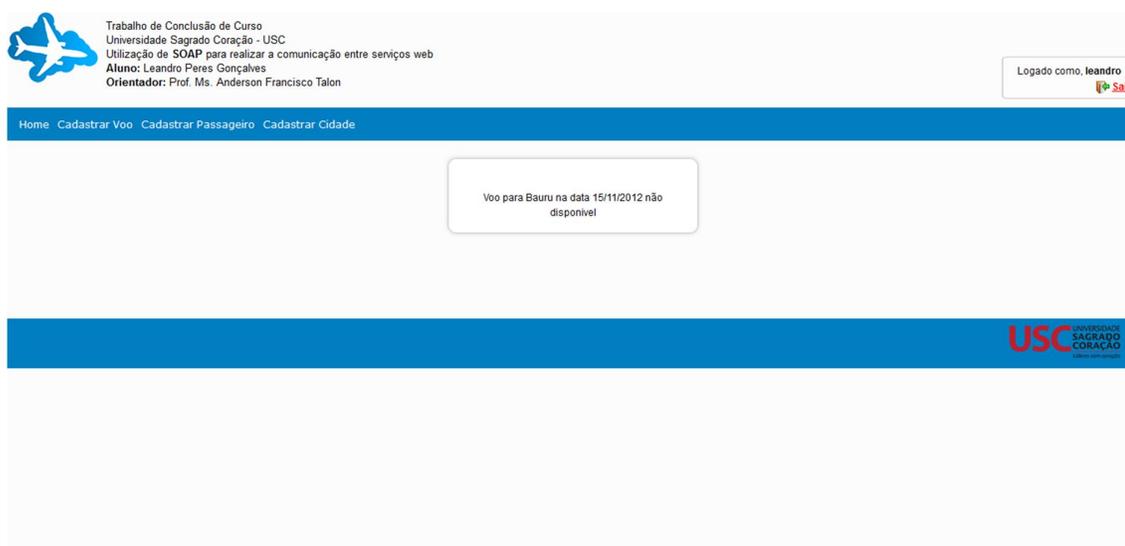


The screenshot shows a web application interface. At the top left, there is a logo of an airplane and text: "Trabalho de Conclusão de Curso", "Universidade Sagrado Coração - USC", "Utilização de SOAP para realizar a comunicação entre serviços web", "Aluno: Leandro Peres Gonçalves", and "Orientador: Prof. Ms. Anderson Francisco Talon". At the top right, there is a login status: "Logado como, leandro" with a "Sair" button. Below the header is a navigation bar with "Home", "Cadastrar Voo", "Cadastrar Passageiro", and "Cadastrar Cidade". The main content area features a form titled "Cadastrar Cidade" with a "Nome:" label, an input field, and a "Cadastrar" button. At the bottom right, there is a logo for "USC UNIVERSIDADE SAGRADO CORAÇÃO".

Figura 20: Cadastro da Cidade.

Fonte: Elaborado pelo autor.

De acordo com a Figura 21, é apresentado o resultado de uma consulta. Neste exemplo, os dados informados pelo usuário não existiam cadastrados no serviço web, e por isto, foi exibido a mensagem de que o voo para a Cidade X na data X não estava disponível.



The screenshot shows the same web application interface as Figure 20. The navigation bar is the same. The main content area now displays a message in a box: "Voo para Bauru na data 15/11/2012 não disponível". The rest of the interface, including the header and footer, remains the same.

Figura 21: Consulta não disponível.

Fonte: Elaborado pelo autor.

De acordo com a Figura 22, é apresentado o resultado de uma consulta. Neste exemplo, os dados informados pelo usuário existem no serviço web. Com isto, a consulta é retornada com sucesso, e as informações dos voos naquela data e naquela cidade são retornadas para o usuário final. Esta consulta é realizada via SOAP, e o retorno destas informações também são retornadas via SOAP.

Todas as requisições e respostas realizadas no sistema são gravadas em um log no formato XML. Na Figura 22, é demonstrada uma requisição e resposta que um resultado e outra que não retornou nenhum resultado.



Trabalho de Conclusão de Curso
Universidade Sagrado Coração - USC
Utilização de SOAP para realizar a comunicação entre serviços web
Aluno: Leandro Peres Gonçalves
Orientador: Prof. Ms. Anderson Francisco Talon

Logado como, leandro

Home Cadastrar Voo Cadastrar Passageiro Cadastrar Cidade

Data	Disponibilidade	Cidade	Horário	Preço	Tarifa	Cancelado	Valor total da passagem
02/11/2012	Voo disponível	Bauru	22:00:00	15,00 R\$	17,00 R\$	Não	32,00 R\$
02/11/2012	Voo não disponível	Bauru	15:00:00	55,00 R\$	27,00 R\$	Sim	82,00 R\$
02/11/2012	Voo disponível	Bauru	22:33:00	223,00 R\$	25,00 R\$	Não	248,00 R\$
02/11/2012	Voo disponível	Bauru	00:00:00	22,00 R\$	30,00 R\$	Não	52,00 R\$
02/11/2012	Voo disponível	Bauru	23:11:00	51,00 R\$	21,00 R\$	Não	72,00 R\$
02/11/2012	Voo disponível	Bauru	19:33:00	234,00 R\$	20,00 R\$	Não	254,00 R\$

USC UNIVERSIDADE SAGRADO CORAÇÃO

Figura 22: Consulta disponível.
Fonte: Elaborado pelo autor.

De acordo com a Figura 23, foi realizada uma consulta informando a cidade e a data.

```

1 POST /tcc/server.php HTTP/1.0
2 Host: 192.168.56.102:8080
3 User-Agent: NuSOAP/0.9.5 (1.123)
4 Content-Type: text/xml; charset=ISO-8859-1
5 SOAPAction: "http://192.168.56.102/tcc/server.php/getVoo"
6 Content-Length: 595
7
8 <?xml version="1.0" encoding="ISO-8859-1"?>
9 <SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
10 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
11 xmlns:tns="http://localhost/tcc/nusoap/">
12 <SOAP-ENV:Body>
13 <tns:getVoo xmlns:tns="http://localhost/tcc/nusoap/">
14 <id_cidade xsi:type="xsd:string">1</id_cidade>
15 <data xsi:type="xsd:string">2012-11-01</data>
16 </tns:getVoo>
17 </SOAP-ENV:Body>
18 </SOAP-ENV:Envelope>

```

Figura 23: Requisição SOAP.
Fonte: Elaborado pelo autor.

De acordo com a Figura 24, foi retornada a resposta da requisição realizada pela Figura 23. Podemos concluir que, na cidade e data especificada pela requisição, temos um voo às 10 horas da manhã. Caso existissem mais voos cadastrados nesta data, seria agregado a este resultado outra tag <item> com as informações do voo especificado dentro dela.

```

1  Server: NuSOAP Server: v0.9.5
2  X-SOAP-Server: NuSOAP/0.9.5 (1.123)
3  Content-Type: text/xml; charset=ISO-8859-1
4  Content-Length: 913
5
6  <?xml version="1.0" encoding="ISO-8859-1"?>
7  <SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
8  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
9  <SOAP-ENV:Body>
10 <ns1:getVooResponse xmlns:ns1="http://localhost/tcc/nusoap/">
11 <return xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="object[1]">
12 <item>
13 <id xsi:type="xsd:string">1</id>
14 <data xsi:type="xsd:string">2012-11-01</data>
15 <status xsi:type="xsd:string">Y</status>
16 <id_cidade xsi:type="xsd:string">Bauru</id_cidade>
17 <horario xsi:type="xsd:string">10:00:00</horario>
18 <preco xsi:type="xsd:string">20213</preco>
19 <tarifa xsi:type="xsd:string">2233</tarifa>
20 <cancelado xsi:type="xsd:string">N</cancelado>
21 </item>
22 </return>
23 </ns1:getVooResponse>
24 </SOAP-ENV:Body>
25 </SOAP-ENV:Envelope>

```

Figura 24: Resposta SOAP.
Fonte: Elaborado pelo autor.

De acordo com a Figura 25, é apresentado o erro de conexão. Esta página tem como finalidade informar ao usuário ou administrador do sistema que houve algum erro relacionado à comunicação das aplicações. No caso de uma consulta, se o servidor externo não tiver conexão no momento, fica impossibilitado de a aplicação se comunicar com a outra e por isto esta mensagem aparece. Ao invés de ser exibido um erro com informações técnicas para o usuário, é preferível ter-se uma tela de fácil compreensão tanto para o usuário quanto para o administrador do sistema.



Figura 25: Erro de conexão.
Fonte: Elaborado pelo autor.

De acordo com a Figura 26, para fins de entendimento do funcionamento do sistema e do protocolo SOAP, é demonstrado a realização de uma consulta.

Consulta disponibilidade de voos por Data

Passageiro:

Cidade:

Data:

Figura 26: Realizando consulta.
Fonte: Elaborado pelo autor.

De acordo com a Figura 27, após esta consulta, é gerado um log em XML com os dados informados pelo usuário através do browser. A cidade e data informada pelo usuário são apresentadas nas linhas 14 e 15, respectivamente. Na

linha 2, é apresentado o host externo para qual esta consulta vai se comunicar. No caso, para o IP 192.168.56.102 porta 8080.

```

1 POST /tcc/server.php HTTP/1.0
2 Host: 192.168.56.102:8080
3 User-Agent: NuSOAP/0.9.5 (1.123)
4 Content-Type: text/xml; charset=ISO-8859-1
5 SOAPAction: "http://192.168.56.102/tcc/server.php/getVoo"
6 Content-Length: 595
7
8 <?xml version="1.0" encoding="ISO-8859-1"?>
9 <SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
10 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
11 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://localhost/tcc/nusoap/">
12 <SOAP-ENV:Body>
13 <tns:getVoo xmlns:tns="http://localhost/tcc/nusoap/">
14 <id_cidade xsi:type="xsd:string">1</id_cidade>
15 <data xsi:type="xsd:string">2012-11-02</data>
16 </tns:getVoo>
17 </SOAP-ENV:Body>
18 </SOAP-ENV:Envelope>

```

Figura 27: XML gerado pela consulta.

Fonte: Elaborado pelo autor.

Após a requisição gerada, a aplicação servidor vai interpretar estas informações (id_cidade e data) e comparar no banco de dados os registros que satisfazem esta condição. Após isto, é gerada uma resposta XML.

De acordo com a Figura 28 é gerada uma resposta com os dados fornecidos pela requisição da Figura 27. Na linha 14 é apresentada a data em que foi fornecida pelo usuário, e que consequentemente existia cadastrada no banco de dados. Na linha 16, é apresentado o nome da cidade, na linha 17 o horário deste voo, no caso 22:00:00. Na linha 24 é apresentada a data em que foi fornecida pelo usuário, na linha 26 a cidade em questão e na 27 o horário, no caso 15:00:00. Neste exemplo, foi retornado mais de um resultado, porque podem existir casos de uma cidade possuir mais viagens para ela em determinado horário.

```

1 Server: NuSOAP Server v0.9.5
2 X-SOAP-Server: NuSOAP/0.9.5 (1.123)
3 Content-Type: text/xml; charset=ISO-8859-1
4 Content-Length: 2713
5
6 <?xml version="1.0" encoding="ISO-8859-1"?>
7 <SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
8 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
9   <SOAP-ENV:Body>
10    <ns1:getVooResponse xmlns:ns1="http://localhost/tcc/nusoap/">
11      <return xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="object[6]">
12        <item>
13          <id xsi:type="xsd:string">2</id>
14          <data xsi:type="xsd:string">2012-11-02</data>
15          <status xsi:type="xsd:string">Y</status>
16          <id_cidade xsi:type="xsd:string">Bauru</id_cidade>
17          <horario xsi:type="xsd:string">22:00</horario>
18          <preco xsi:type="xsd:string">1500</preco>
19          <tarifa xsi:type="xsd:string">1700</tarifa>
20          <cancelado xsi:type="xsd:string">N</cancelado>
21        </item>
22        <item>
23          <id xsi:type="xsd:string">4</id>
24          <data xsi:type="xsd:string">2012-11-02</data>
25          <status xsi:type="xsd:string">N</status>
26          <id_cidade xsi:type="xsd:string">Bauru</id_cidade>
27          <horario xsi:type="xsd:string">15:00</horario>
28          <preco xsi:type="xsd:string">5500</preco>
29          <tarifa xsi:type="xsd:string">2700</tarifa>
30          <cancelado xsi:type="xsd:string">Y</cancelado>
31        </item>

```

Figura 28: XML gerado pela resposta.

Fonte: Elaborado pelo autor.

Após a geração da resposta, esta é enviada novamente ao cliente, que por sua vez retorna os dados no browser para o usuário final visualizar.

De acordo com a Figura 29, são retornados para o usuário final os mesmos dados que constavam na resposta (Figura 28), concluindo assim a comunicação SOAP.

Data	Disponibilidade	Cidade	Horário	Preço	Tarifa	Cancelado	Valor total da passagem
02/11/2012	Voo disponível	Bauru	22:00:00	15,00 R\$	17,00 R\$	Não	32,00 R\$
02/11/2012	Voo não disponível	Bauru	15:00:00	55,00 R\$	27,00 R\$	Sim	82,00 R\$

Figura 29: Tela exibida ao usuário após a chegada da resposta ao cliente

Fonte: Elaborado pelo autor.

5 CONSIDERAÇÕES FINAIS

O sistema desenvolvido procurou demonstrar o funcionamento prático do protocolo SOAP, além de comprovar de que o SOAP é um protocolo que consegue realizar a comunicação entre serviços web. Este protocolo mostrou-se capaz de transportar mensagens de forma centralizada em um ambiente distribuído. As principais vantagens do protocolo estudado são descritas a seguir:

- Fazer com que uma aplicação de diferentes plataformas consiga se interagir através de trocas de mensagens.
- Os dados SOAP são estruturados de forma organizada, utilizando XML.
- O SOAP se mostrou eficiente na sua utilização juntamente com o protocolo HTTP, que é o meio por onde o SOAP consegue trafegar as mensagens.
- O SOAP é suportado por partes de várias plataformas.

Este trabalho implementou um *web service* que realizou a comunicação via SOAP, conforme os resultados apresentados neste trabalho de conclusão de curso.

Para os trabalhos futuros, ficam como sugestões:

- Desenvolver uma tela de cadastro para os passageiros, e consequentemente integrar após a consulta de passagens aéreas a compra e agendamento destas passagens.
- Melhorar a consulta, onde ao invés de o usuário somente informar somente a cidade destino, ele também informar a cidade origem. Exemplo: São Paulo à Rio de Janeiro, e não somente São Paulo.
- A inserção de mais um sistema web, onde juntamente com o sistema de agência de passagens aéreas desenvolvido, o usuário pudesse consultar também os hotéis para a cidade em que ele vai realizar o seu voo.
- Cadastro de usuários por níveis para administrar o sistema. Esta tela não foi desenvolvida, pois o real objetivo deste TCC era demonstrar a comunicação SOAP entre aplicações distintas, e por isto, os cadastros

foram realizados de forma direta no banco de dados. Uma tela de cadastro para os usuários administrativos por nível seria interessante.

REFERÊNCIAS

ALONSO, G. et al. **Web Services: Concepts, Architectures and Applications**. Berlim: Springer - Verlag, 2004.

ALONSO, G. et al. **Web Services: Concepts, Architectures and Applications**. Berlim: Springer – Verlag, 2010.

AMORIM, S. da. S. **A tecnologia Web Services e sua aplicação num Sistema de Gerência de Telecomunicações**. 2004. 94 f. Dissertação de Mestrado (Engenharia de Computação). Universidade Estadual de Campinas (UNICAMP), Instituto de Computação (IC), Campinas, SP, 2004.

BELLWOOD, T. UDDI Version 2.04 API Specification. **Uddi.org**, 2002. Disponível em: <http://uddi.org/pubs/ProgrammersAPI_v2.htm>. Acesso em: 12 abr. 2012.

BOX, D. et al. Simple Object Access Protocol (SOAP) 1.1. **W3.org**, 2000. Disponível em: <<http://www.w3.org/TR/SOAP>>. Acesso em 09 abr. 2012.

BRAY, T. et al. Extensible Markup Language (XML), W3C Recommendation. **W3.org**, 2006. Disponível em: <<http://www.w3.org/XML/>>. Acesso em 13 abr. 2012.

CHINNICI, R. et al. Web Services Description Language Part 1 Core Language Version 2.0 Specification. **W3.org**, 2007. Disponível em: <<http://www.w3.org/TR/2007/REC-wsdl20-20070626/>>. Acesso em: 09 maio 2012.

CLEMENT, L. et al. Universal Description Discovery & Integration Version 3.0.2 Specification. **W3.org**, 2007. Disponível em: <<http://www.w3.org/TR/2007/REC-wsdl20-20070626/>>. Acesso em: 09 maio 2012.

HENDRICKS, M. et al. **Professional Java Web Services**. Rio de Janeiro, Alta Books, 2002.

LEOPOLDO, M. R. B. **Simple Object Access: entendendo o simple object access protocol (SOAP)**. [S.l.], 2012. Disponível em: <<http://wiki.pge.ce.gov.br/images/0/0b/SOAP.pdf>>. Acesso em: 18 ago. 2012.

MEDEIROS, A. K.; GIMENES, I. M. S.; TOLEDO, M. B. **Sistemas de Gestão de Processos de Negócios: desafios e oportunidades**. Post-proceedings Workshop Empresarial ENEGEP, Foz de Iguaçu, Paraná, 2007.

MENÉNDEZ, A.I.M. **Uma ferramenta de apoio ao desenvolvimento de web services**. 97 f. Dissertação de Mestrado (Ciências da Computação). Universidade Federal de Campina Grande, Campina Grande, 2002.

MILTRA, N.; LAFON, Y. SOAP Part 0 Primer Version 1.2 Specification. **W3.org**, 2007. Disponível em: <<http://www.w3.org/TR/soap12-part0/>>. Acesso em: 04 maio 2012.

OASIS. **Oasis-open.org**. 2006. Disponível em: <<http://www.oasis-open.org/committees/download.php/18486/pr-2changes.pdf>>. Acesso em: 07 abr. 2012

PAPAZOGLU, M. P.; GEORGAKOPOULOS, D. **Service Oriented Computing, Communications of the ACM: Service Oriented Computing**, 46(10), ACM Press, p. 25-28, 2003.

SOUZA, V. A. S. M. de. **Uma arquitetura orientada a serviços para desenvolvimento, gerenciamento e instalação de serviços de rede**. 2006. 82 f. Dissertação de Mestrado (Engenharia Elétrica). Universidade Estadual de Campinas (UNICAMP), Campinas, 2006.

SOUZA, V.C. **Mini curso web services com PHP**. São Leopoldo, 2005. Disponível em: <http://www.unisinos.br/seminario/sdsl/_arquivos/WebServicesPHP.pdf>. Acesso em: 19 ago. 2012.

THOMAS, E. **SOA Principles of Service Design**. New York. Prentice Hall, 2007.

W3SCHOOLS.com. **SOAP TUTORIAL**. [S.l.], c1999-2012. Informações sobre padrões Web. Disponível em: <<http://www.w3schools.com/soap/default.asp>>. Acesso em: 15 maio 2012.

APÊNDICE A – ARTIGO NO FORMATO DA SBC

A UTILIZAÇÃO DE UM PROTOCOLO ESPECÍFICO DE
COMUNICAÇÃO ENTRE SERVIÇOS WEB: SOAPLeandro P. Gonçalves¹, Anderson F. Talon², Márcio H. C. Cardim³, Henrique P.
Martins⁴^{1, 2, 3, 4} Centro de Ciências Exatas e Sociais Aplicadas^{1, 2, 3, 4} Universidade Sagrado Coração (USC) – Bauru, SP – Brazilleandrol.goncalves@usc.edu.br, anderson.talon@usc.br, mcardim@usc.br,
henrique.martins@usc.br

Abstract. *With the growth of web services, it becomes necessary to use technologies that can supply the necessity of the users, as the exigencies to perform the multiple tasks which need systems to interact and to reach a single result. Web services are application components that can be developed and published individually and used for distributed applications, using like main mean of communication the Internet. Increasingly, the companies realize the necessity to integrate their systems it brings agility to the process and efficiency in communication between supply chains and logistics, inside or outside of the organization, who are their clients and suppliers. The purpose of this work is to build two web services, which will be destined for the client and another for the server. The system will be responsible for performing the communication using the exchange of XML messages wrapped in SOAP envelopes.*

Resumo. *Com o crescimento dos serviços web, torna-se necessário a utilização de tecnologias que possam suprir as necessidades dos usuários, conforme as exigências dos mesmos em realizar tarefas múltiplas que necessitam de sistemas que se interajam para chegar a um resultado único. Serviços web são componentes de aplicações que podem ser desenvolvidos e publicados individualmente e usados por aplicações distribuídas, utilizando como principal meio de comunicação a internet. Cada vez mais as empresas veem a necessidade de integrar seus sistemas, visando trazer agilidade para os processos e eficiência na comunicação entre cadeias de produção ou de logística, tanto na parte interna, quanto na parte externa da organização, que são seus clientes e fornecedores. A proposta deste trabalho é construir dois serviços web, onde um será destinado para o cliente, e o outro para o servidor. O sistema se encarregará de realizar a comunicação entre o cliente e servidor, utilizando a troca de mensagens XML encapsulada em envelopes SOAP.*

1. Introdução

Segundo o grupo OASIS, que é o responsável por diversas padronizações, a arquitetura orientada a serviço (AOS) “é um paradigma para organização e utilização de capacidades distribuídas que podem estar sob o controle de diferentes domínios de propriedade.” (OASIS, 2006, p. 8). Esse paradigma provê um meio regular para oferecer, descobrir e utilizar capacidades. Com isso, se promove a integração dos processos de negócio. Através de uma interface neutra e protocolos de transporte padronizados, AOS usa em sua construção o princípio de componentes de software fracamente acoplados chamados serviços. (SOUZA,

2006).

SOAP é um protocolo superficial que exporta informação de forma centralizada em um ambiente distribuído. Ele é um protocolo baseado em XML que consiste de três partes: um envelope que define um framework para descrever o que é uma mensagem e como processá-la; um conjunto de regras de codificação para expressar instâncias de tipos definidos pela aplicação; e uma convenção para representar chamadas remotas de procedimento e respostas. (BOX, 2000).

Este protocolo não está vinculado a uma plataforma de hardware, a um sistema operacional, linguagem de programação ou hardware de rede. Diferentemente de outros sistemas de computação distribuídos, o SOAP é construído no topo de padrões abertos, como HTTP e XML. A utilização de padrões amplamente aceitos e conhecidos facilita o aprendizado, por parte dos desenvolvedores, da nova tecnologia, e também da infraestrutura existente para suportá-la. (BELWOOD, 2002).

Segundo Leopoldo (2002), tira-se a conclusão de que o protocolo SOAP é definido como o elemento principal da infraestrutura dos web services e fundamental para o funcionamento dos mesmos independente de plataforma ou linguagem de programação específica. Este Trabalho de Conclusão de Curso foi proposto para demonstrar a importância da comunicação dentro do contexto de Arquitetura de Gestão de Processo de Negócios. Sabendo-se da complexidade de tal arquitetura, um dos grandes problemas atualmente é realizar uma comunicação de forma com que se tenham requisições e respostas totalmente em tempo real e com menos falhas possíveis. O trabalho proporcionou um aprendizado sobre Serviços Web, além de fornecer um estudo completo sobre um protocolo de alta aceitação, como o SOAP. Para a sociedade, este trabalho trouxe uma importante abordagem, que é a portabilidade de um sistema ou aplicação. Um sistema onde não se depende de plataformas de hardware ou software é de extrema aceitação no contexto da Tecnologia da Informação, e com os Web Services isto se torna totalmente possível, fazendo com que a sociedade tenha em suas mãos informações relevantes na hora em que precisar sem depender de uma plataforma específica. Para o meio acadêmico, o trabalho trouxe uma vasta área de pesquisa, que é a comunicação de aplicações que nem sempre são desenvolvidas nas mesmas plataformas e de alguma forma precisam se interligar para chegar a um resultado final.

O objetivo geral deste trabalho de conclusão de curso é realizar a comunicação entre a organização consumidora e fornecedora de serviços. Já os objetivos específicos são: Estudar um protocolo de comunicação de alta aceitação no contexto web, propiciar aos sistemas web uma comunicação amigável, realizar a comunicação entre o cliente e o servidor utilizando o protocolo SOAP, através de troca de mensagens XML, fazer a comunicação entre o cliente, que pode ser classificado como organização consumidora, e o fornecedor, que pode ser classificado como organização fornecedora.

2. Fundamentação Teórica

De acordo com Alonso et al. (2010), um serviço web é um tipo exclusivo de serviço eletrônico que usa padrões abertos da Internet para a sua descrição, descoberta e invocação. Serviços Web podem ser vistos como aplicações, disponíveis na Internet, que podem ser usadas para executar determinados serviços eletrônicos. Eles podem ser desde simples serviços – como a realização de verificações de números de CPF, até serviços complexos, como aplicação de ações de cobranças. Segundo Menéndez (2002), um web service é uma aplicação que aceita solicitações de outros sistemas através de um meio específico chamado internet. Outra definição é a de que são componentes de software que oferecem uma funcionalidade única, podendo ser acessados por diferentes sistemas em diferentes contextos,

através de padrões da internet, como HyperText Transfer Protocol (HTTP), eXtensible Markup Language (XML) e Simple Object Access Protocol (SOAP).

Um repositório para armar Neste trabalho de conclusão de curso (TCC)zenar as informações sobre empresas e serviços, conhecido como UDDI, e um protocolo de comunicação que permite a requisição dos serviços, conhecido com SOAP. Todos os padrões citados utilizam XML.

XML: De acordo com Bray et al. (2006) o XML (*eXtensible Markup Language*) é uma linguagem que permite a especificação de documentos contendo dados estruturados na forma de árvores. É uma recomendação da W3C. Um documento XML contém texto.

WSDL: De acordo com Chinnici et al. (2007) o WSDL (Web Services Description Language) é uma linguagem usada para descrever serviços Web. Uma descrição WSDL contém as informações relacionadas ao serviço Web necessárias para sua publicação, descoberta e invocação, incluindo capacidades, interface, comportamento e qualidade de serviços. Cada serviço web é descrito de forma independente por meio de: um tipo de porta (port type); operações (operations) que o tipo de porta oferece; e a estrutura de mensagens (message) de entrada e de saída.

UDDI: De acordo com Clement et al. (2004) o UDDI (Universal Description, Discovery, and Integration) é um padrão que define a estrutura e o conteúdo dos diretórios que contém descrições de serviços oferecidos. 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. Esse padrão permite também que clientes de serviços disponibilizem suas necessidades para que provedores de serviços busquem por usuários que precisam de um determinado serviço.

SOAP: De acordo com Miltra e Lafon (2007) o SOAP (Simple Object Access Protocol) é um protocolo que define um mecanismo para a comunicação geral entre serviços Web na Internet. Ele define o formato das mensagens (normalmente com conteúdo de dados XML) que são trocadas entre clientes de serviços, provedores de serviços e diretórios de serviços. De uma forma geral, o protocolo SOAP descreve como o protocolo HTTP (HyperText Transfer Protocol) pode ser usado para chamada de procedimento remoto na Internet, por meio de uma combinação de um cabeçalho HTTP com um corpo SOAP. Porém, outros protocolos de transporte também podem ser usados, como por exemplo, o SMTP (Simple Mail Transfer Protocol).

Existem várias maneiras de se implementar web services utilizando a linguagem de programação PHP. De acordo com Souza (2005), é descrito algumas possibilidades baseadas em código livre:

- XML-RPC: é uma implementação de chamadas de procedimentos remotos (RPC) que possibilita o transporte de dados no formato XML entre dois servidores utilizando o protocolo HTTP;
- PEARL SOAP Client/Server for PHP: o PHP Extension and Application Repository (PEAR) é um repositório de classes PHP de licença livre que tem uma grande comunicação de colaboradores. Ele possui uma implementação do protocolo SOAP e seus serviços.
- nuSoap: é uma API desenvolvida em PHP que permite a criação de clientes e servidores web services, possui suporte embutido à WSDL e facilidade para a instalação e utilização.
- Representational State Transfer (REST): o REST não possui um padrão de implementação e não oferece classes pré-construídas. Utiliza o protocolo HTTP para o envio e recebimento de mensagens no padrão XML. Para realizar a troca de mensagens XML, utilizam-se os métodos

padrões do HTTP: GET, POST e PUT. Na sua implementação torna-se necessário o uso de outras ferramentas como PHP Document Object Model (DOM), Simple API for XML (SAX) ou também eXtensible Stylesheet Language (XSL) para realizar o parsing;

- PHP SOAP Extension: este é a primeira implementação nativa do protocolo SOAP incorporada na versão 5 do PHP. Sua principal vantagem entre as outras implementações escritas em PHP é a velocidade. Pode ser usada para implementar clientes e servidores e tem uma série de funções pré-definidas.

Neste trabalho de conclusão de curso (TCC), foi utilizado a biblioteca de classes nuSoap. O nuSoap conta com diversos recursos e uma facilidade na compreensão, além de fornecer ao desenvolvedor fácil entendimento tanto na publicação do serviço quanto no consumo do mesmo, e isto ajudou na escolha desta ferramenta em específico. O uso desta API proporcionou um ágil desenvolvimento do sistema, já que ela contém métodos pré-definidos que proporcionaram: a chamada ao fornecedor de serviços, a criação do arquivo WSDL, a publicação deste WSDL no repositório de serviços (UDDI) e a comunicação geral entre Cliente / Fornecedor (SOAP).

3. Protótipo

O protótipo desenvolvido é um sistema que realiza consultas a uma base remota contendo os dados de passagens aéreas, através do protocolo de comunicação SOAP. Desta forma, o web service possibilita que qualquer aplicação com suporte ao protocolo de comunicação SOAP utilize o método que retorna as informações das passagens aéreas, que foi desenvolvido neste trabalho.

Conforme ilustrado na Figura 1, o usuário, que é representado pelo navegador (browser), envia os dados para a realização da consulta, no caso, é preciso fornecer a cidade e a data em que se deseja pesquisar informações relacionadas às passagens aéreas. Após esta consulta, o Cliente, que é representado pela Agência de Viagem, recebe a consulta do Usuário e faz uma requisição ao Fornecedor, que é representado pela companhia aérea. A companhia processa a requisição e gera uma resposta, que é enviada para o Cliente, que por fim envia para o usuário e exibe os dados retornados através do SOAP para a visualização no browser.



Figura 1. Diagrama de sequência do funcionamento do sistema

Neste trabalho de conclusão de curso (TCC), foi utilizado a biblioteca de classes nuSoap. O nuSoap conta com diversos recursos e uma facilidade na compreensão, além de fornecer ao desenvolvedor fácil entendimento tanto na publicação do serviço quanto no consumo do mesmo, e isto ajudou na escolha desta ferramenta em específico. O uso desta API

proporcionou um ágil desenvolvimento do sistema, já que ela contém métodos pré-definidos que proporcionaram: a chamada ao fornecedor de serviços, a criação do arquivo WSDL, a publicação deste WSDL no repositório de serviços (UDDI) e a comunicação geral entre Cliente / Fornecedor (SOAP).

4. Resultados

De acordo com a Figura 2, para fins de entendimento do funcionamento do sistema e do protocolo SOAP, é demonstrado a realização de uma consulta.

Consulta disponibilidade de voos por Data

Passageiro:

Cidade:

Data:

Figura 2. Realizando consulta

De acordo com a Figura 3, após esta consulta, é gerado um log em XML com os dados informados pelo usuário através do browser. A cidade e data informada pelo usuário são apresentadas nas linhas 14 e 15, respectivamente. Na linha 2, é apresentado o host externo para qual esta consulta vai se comunicar. No caso, para o IP 192.168.56.102 porta 8080.

```

1 POST /tcc/server.php HTTP/1.0
2 Host: 192.168.56.102:8080
3 User-Agent: NuSOAP/0.9.5 (1.123)
4 Content-Type: text/xml; charset=ISO-8859-1
5 SOAPAction: "http://192.168.56.102/tcc/server.php/getVoo"
6 Content-Length: 595
7
8 <?xml version="1.0" encoding="ISO-8859-1"?>
9 <SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
10 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
11 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://localhost/tcc/nusoap/">
12   <SOAP-ENV:Body>
13     <tns:getVoo xmlns:tns="http://localhost/tcc/nusoap/">
14       <id_cidade xsi:type="xsd:string">1</id_cidade>
15       <data xsi:type="xsd:string">2012-11-02</data>
16     </tns:getVoo>
17   </SOAP-ENV:Body>
18 </SOAP-ENV:Envelope>

```

Figura 3. XML gerado pela consulta

Após a requisição gerada, a aplicação servidor vai interpretar estas informações (id_cidade e data) e comparar no banco de dados os registros que satisfazem esta condição. Após isto, é gerada uma resposta XML.

Conforme a Figura 4 é gerada uma resposta com os dados fornecidos pela requisição da Figura 3. Na linha 14 é apresentada a data em que foi fornecida pelo usuário, e que consequentemente existia cadastrada no banco de dados. Na linha 16, é apresentado o nome da cidade, na linha 17 o horário deste voo, no caso 22:00:00. Na linha 24 é apresentada a data em que foi fornecida pelo usuário, na linha 26 a cidade em questão e na 27 o horário, no caso

15:00:00. Neste exemplo, foi retornado mais de um resultado, porque podem existir casos de uma cidade possuir mais viagens para ela em determinado horário.

```

1 Server: NuSOAP Server v0.9.5
2 X-SOAP-Server: NuSOAP/0.9.5 (1.123)
3 Content-Type: text/xml; charset=ISO-8859-1
4 Content-Length: 2713
5
6
7 <?xml version="1.0" encoding="ISO-8859-1"?>
8 <SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
9 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
10 <SOAP-ENV:Body>
11 <ns1:getVooResponse xmlns:ns1="http://localhost/tcc/nussoap/">
12 <return xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="object[6]">
13 <item>
14 <id xsi:type="xsd:string">2</id>
15 <data xsi:type="xsd:string">2012-11-02</data>
16 <status xsi:type="xsd:string">Y</status>
17 <id_cidade xsi:type="xsd:string">Bauru</id_cidade>
18 <horario xsi:type="xsd:string">22:00:00</horario>
19 <preco xsi:type="xsd:string">1500</preco>
20 <tarifa xsi:type="xsd:string">1700</tarifa>
21 <cancelado xsi:type="xsd:string">N</cancelado>
22 </item>
23 <item>
24 <id xsi:type="xsd:string">4</id>
25 <data xsi:type="xsd:string">2012-11-02</data>
26 <status xsi:type="xsd:string">N</status>
27 <id_cidade xsi:type="xsd:string">Bauru</id_cidade>
28 <horario xsi:type="xsd:string">15:00:00</horario>
29 <preco xsi:type="xsd:string">5500</preco>
30 <tarifa xsi:type="xsd:string">2700</tarifa>
31 <cancelado xsi:type="xsd:string">Y</cancelado>
32 </item>

```

Figura 4. XML gerado pela resposta

Após a geração da resposta, esta é enviada novamente ao cliente, que por sua vez retorna os dados no browser para o usuário final visualizar.

Conforme a Figura 5, são retornados para o usuário final os mesmos dados que constavam na resposta (Figura 4), concluindo assim a comunicação SOAP.

Data	Disponibilidade	Cidade	Horário	Preço	Tarifa	Cancelado	Valor total da passagem
02/11/2012	Voo disponível	Bauru	22:00:00	15,00 R\$	17,00 R\$	Não	32,00 R\$
02/11/2012	Voo não disponível	Bauru	15:00:00	55,00 R\$	27,00 R\$	Sim	82,00 R\$

Figura 5. Tela exibida ao usuário após a chegada da resposta ao cliente

5. Considerações Finais

O sistema desenvolvido procurou demonstrar o funcionamento prático do protocolo SOAP, além de reforçar a teoria de que o SOAP é um protocolo que consegue realizar a comunicação entre serviços web. Este protocolo mostrou-se capaz de transportar mensagens de forma centralizada em um ambiente distribuído. As principais vantagens do protocolo estudado são descritas a seguir:

- Fazer com que uma aplicação de diferentes plataformas consiga se interagir através de trocas de mensagens.
- Os dados SOAP são estruturados de forma organizada, utilizando XML.
- O SOAP se mostrou eficiente na sua utilização juntamente com o protocolo HTTP, que é o meio por onde o SOAP consegue trafegar as mensagens.
- O SOAP é suportado por partes de várias plataformas.

Este trabalho propôs e implementou o desenvolvimento de um web service que realizasse a comunicação via SOAP, conforme os resultados apresentados neste trabalho de conclusão de curso.

Para os trabalhos futuros, ficam como sugestões:

- Desenvolver uma tela de cadastro para os passageiros, e conseqüentemente integrar após a consulta de passagens aéreas a compra e agendamento destas passagens.

- Melhorar a consulta, onde ao invés de o usuário somente informar somente a cidade destino, ele também informar a cidade origem. Exemplo: São Paulo à Rio de Janeiro, e não somente São Paulo.
- A inserção de mais um sistema web, onde juntamente com o sistema de agência de passagens aéreas desenvolvido, o usuário pudesse consultar também os hotéis para a cidade em que ele vai realizar o seu voo.
- Cadastro de usuários por níveis para administrar o sistema. Esta tela não foi desenvolvida, pois o real objetivo deste TCC era demonstrar a comunicação SOAP entre aplicações distintas, e por isto, os cadastros foram realizados de forma direta no banco de dados. Uma tela de cadastro para os usuários administrativos por nível seria interessante.

6. Referências

- BELLWOOD, T. UDDI Version 2.04 API Specification. **Uddi.org**, 2002. Disponível em: <http://uddi.org/pubs/ProgrammersAPI_v2.htm>. Acesso em: 12 abr. 2012.
- BOX, D. et al. Simple Object Access Protocol (SOAP) 1.1. **W3.org**, 2000. Disponível em: <<http://www.w3.org/TR/SOAP>>. Acesso em 09 abr. 2012.
- BRAY, T. et al. Extensible Markup Language (XML), W3C Recommendation. **W3.org**, 2006. Disponível em: <<http://www.w3.org/XML/>>. Acesso em 13 abr. 2012.
- CHINNICI, R. et al. Web Services Description Language Part 1 Core Language Version 2.0 Specification. W3.org, 2007. Disponível em: <http://www.w3.org/TR/2007/REC_wsdl20-20070626/>. Acesso em: 09 maio 2012.
- CLEMENT, L. et al. Universal Description Discovery & Integration Version 3.0.2 Specification. W3.org, 2007. Disponível em: <<http://www.w3.org/TR/2007/REC-wsdl20-20070626/>>. Acesso em: 09 maio 2012.
- LEOPOLDO, M. R. B. **Simple Object Access: entendendo o simple object access protocol (SOAP)**. [S.l], 2012. Disponível em: <<http://wiki.pge.ce.gov.br/images/0/0b/SOAP.pdf>>. Acesso em: 18 ago. 2012.
- MENÉNDEZ, A.I.M. **Uma ferramenta de apoio ao desenvolvimento de web services**. 97 f. Dissertação de Mestrado (Ciências da Computação). Universidade Federal de Campina Grande, Campina Grande, 2002.
- MILTRA, N.; LAFON, Y. SOAP Part 0 Primer Version 1.2 Specification. W3.org, 2007. Disponível em: <<http://www.w3.org/TR/soap12-part0/>>. Acesso em: 04 maio 2012.
- OASIS. Oasis-open.org. 2006. Disponível em: <<http://www.oasis-open.org/committees/download.php/18486/pr-2changes.pdf>>. Acesso em: 07 abr. 2012
- SOUZA, V. A. S. M. de. Uma arquitetura orientada a serviços para desenvolvimento, gerenciamento e instalação de serviços de rede. 2006. 82 f. Dissertação de Mestrado (Engenharia Elétrica). Universidade Estadual de Campinas (UNICAMP), Campinas, 2006.

APÊNDICE B – ARTIGO APRESENTADO NO FÓRUM DE INICIAÇÃO CIENTÍFICA DA USC

O USO DE UM PROTOCOLO DE COMUNICAÇÃO ENTRE SERVIÇOS WEB: SOAP

Autores: **Gonçalves, L.P.; Talon, A.F.**

Área do conhecimento: **Ciências Exatas e Sociais Aplicadas**

Instituição de origem: **USC**

Com o crescimento da internet, tornou-se necessário a utilização de tecnologias que conseguissem suprir as necessidades dos usuários em realizar tarefas múltiplas, que necessitavam que dois ou mais sistemas se comunicassem para se chegar a um único resultado. Os serviços web são um tipo exclusivo de serviço eletrônico que usa padrões abertos da Internet para a sua descrição, descoberta e invocação. Serviços Web podem ser vistos como aplicações, disponíveis na Internet, que podem ser usadas para executar determinados serviços eletrônicos. Eles podem ser desde simples serviços até serviços mais complexos. No contexto de serviços web, existem diversos padrões, entre eles, está o *Simple Object Access Protocol* (SOAP). O SOAP é um protocolo que define um mecanismo para a comunicação geral entre serviços web na Internet. Ele define o formato das mensagens (normalmente com conteúdo de dados XML) que são trocadas entre clientes de serviços, provedores de serviços e diretórios de serviços. Este trabalho demonstra que o SOAP é um protocolo que consegue realizar a comunicação entre aplicações distintas e distribuídas, e ainda permite que sistemas de diferentes plataformas se comuniquem com o objetivo de se ter uma informação relevante para o usuário final. A utilização deste protocolo só torna-se possível juntamente com outros padrões, dentre eles estão: XML (eXtensible Markup Language), WSDL (Web Service Definition Language) e UDDI (Universal Description Discovery and Integration), juntos, eles formam a arquitetura ideal para os serviços web e assim possibilitam a comunicação entre aplicações de diferentes plataformas.

Palavras-chave: SOAP. Serviços Web. XML.