

**Universidade Estadual de Goiás (UEG)
Câmpus Nordeste
Unidade Universitária de Posse
Instituto Acadêmico de Ciências Tecnológicas
Bacharelado em Sistemas de Informação**

**INTEROPERABILIDADE DE APLICAÇÕES COM
SOAP (*SIMPLE OBJECT ACCESS PROTOCOL*)**

**Aluno: Douglas Diniz Berça
Professor: Ronaldo Ferreira da Silva**

**Posse-GO
Janeiro/2021**

DOUGLAS DINIZ BERÇA

**INTEROPERABILIDADE DE APLICAÇÕES COM SOAP (*SIMPLE OBJECT
ACCESS PROTOCOL*)**

Relatório técnico-científico apresentado ao Curso de Sistemas de Informação da Universidade Estadual de Goiás (UEG) – Unidade Universitária de Posse, como requisito parcial para obtenção da nota da disciplina Programação IV.

Professor: Ronaldo Ferreira da Silva

**Posse-GO
Janeiro/2021**

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UEG
com os dados fornecidos pelo(a) autor(a).

DD737 Diniz Berça, Douglas
i INTEROPERABILIDADE DE APLICAÇÕES COM SOAP (SIMPLE
 OBJECT ACCESS PROTOCOL) / Douglas Diniz Berça;
 orientador Ronaldo Ferreira da Silva. -- Posse-GO,
 2021.
 16 p.

 Graduação - Sistemas de Informação -- Unidade de
 Posse, Universidade Estadual de Goiás, 2021.

 1. Interoperabilidade. 2. SOAP. 3. Protocolo. 4. Web
 services. 5. XML. I. Ferreira da Silva, Ronaldo,
 orient. II. Título.

SUMÁRIO

1. INTRODUÇÃO	3
2. REVISÃO DE LITERATURA	5
2.1 <i>WEB SERVICES</i>	5
2.2 <i>XML (EXTENDED MARKUP LANGUAGE)</i>	5
2.3 <i>WSDL (WEB SERVICES DESCRIPTION LANGUAGE)</i>	6
2.4 <i>SOAP (SIMPLE OBJECT ACCESS PROTOCOL)</i>	7
2.4.1. <i>Interoperabilidade entre Sistemas e outros protocolos</i>	8
2.4.2. <i>Arquitetura do SOAP</i>	8
2.5 <i>API (APPLICATION PROGRAMMING INTERFACE)</i>	9
2.6 <i>REST e RESTful</i>	9
3. RESULTADOS E CONCLUSÕES	11
4. REFERÊNCIAS	13

1. INTRODUÇÃO

Atualmente a comunicação remota permite a execução de aplicativos em diferentes computadores com o objetivo de interagir uns com os outros na rede. Os avanços nos serviços de comunicações nos últimos anos possibilitaram o surgimento de várias tecnologias. Nesse contexto, evidenciam-se as tecnologias denominadas *Web Services*, que vem ganhando visibilidade nos sistemas que executam em redes. Onde, têm-se como destaque sua simplicidade e padronização, responsáveis por contribuir para a obtenção de interoperabilidade entre aplicações e geograficamente distribuídas. Muitas empresas vêm adotando esse tipo de sistema, pois além de promover a interoperabilidade, este proporciona uma maior flexibilização dos sistemas e uma melhor capacidade de cooperação entre empresas (PERAZZOLI, 2011; ALVES, 2007).

Responsável por garantir a comunicação entre os sistemas, os protocolos de comunicação desempenham papel fundamental, servindo de sustentação para a arquitetura de desenvolvimento. No caso dos *Web Services*, o protocolo de comunicação a ser utilizado necessitaria de características particulares para dar possibilidades de implementação capazes de superar as limitações de outros protocolos de função semelhante já existentes. Esse novo protocolo precisaria garantir entre outros recursos leveza, flexibilidade, facilidade de implementação e depuração.

Diante desse pressuposto, denota-se que várias tecnologias de comunicação remota encontram-se disponíveis para desenvolvedores de aplicativos desde a introdução da computação distribuída, uma destas tecnologias é o SOAP (*Simple Object Access Protocol*).

O SOAP é um protocolo largamente utilizado para troca de mensagens na *Web Services*, pois este promove a comunicação entre aplicativos escritos em diferentes linguagens e implantados em diferentes plataformas, pela rede, sem a necessidade de usar adaptadores personalizados (PERAZZOLI, 2011; DAVIS e ZHANG, 2004). Este protocolo foi projetado com o objetivo de aumentar a interoperabilidade entre uma ampla gama de programas e ambientes. Mais que outro protocolo de comunicação, o SOAP idealiza ser uma tecnologia na qual se baseia um novo paradigma no desenvolvimento de sistemas distribuídos, garantindo a interoperabilidade de serviços heterogêneos, remotamente distribuídos, com todos os requisitos necessários. E isso só se torna possível

devido os protocolos onde o mesmo se baseia, que o permite entender e transmitir mensagens de diferentes plataformas. O SOAP se baseia especificamente em duas tecnologias comuns: HTTP (*HyperText Transfer Protocol*) e XML (*Extensible Markup Language*), cada uma delas desempenhando papéis específicos (PERAZZOLI, 2011; DAVIS e ZHANG, 2004).

Diante destes postulados, o seguinte estudo objetiva estudar e transcrever a interoperabilidade de aplicações existente utilizando o protocolo SOAP. Pormenorizando suas características, e destacando as quais delas fazem com este protocolo ganhe destaque nas aplicações atuais.

2. REVISÃO DE LITERATURA

2.1 WEB SERVICES

Os *Web Services* ou serviços da *Web*, têm como papel principal fornecer aos desenvolvedores de sistemas distribuídos uma solução para o problema da comunicação entre os componentes de uma aplicação distribuída e heterogênea, através de redes distintas de computadores que possuem no ponto de interconexão, dispositivos de segurança, como firewalls, que permitem apenas o tráfego de informações em formatos e de maneiras previamente determinadas (ERL, 2009).

Além de possibilitar que as informações fornecidas possam transpor sem dificuldades as barreiras dos *firewalls*¹, é preciso garantir que cada componente da aplicação que irá receber a informação saiba que tipo de informação está recebendo e como essa informação pode ser lida. Isso é possível através do uso da linguagem XML, que é uma linguagem específica para a marcação de dados. Segundo Rodrigues (2011) os *Web Services* são uma maneira para programas em uma máquina usarem os recursos de outra máquina, por exemplo, por meio do uso de Chamadas de Procedimentos Remotos (RPC – *Remote Procedure Call*).

Embora o RPC binário esteja em evidência por décadas, os *Web Services* codificam suas conversações em XML e falam genericamente através do protocolo base da *Web*, o HTTP. (DAVIS e ZHANG, 2004; BORGES, 2011).

Denota-se que o que torna os *Web Services* tão interessantes e atraentes é o fato da sua estrutura permitir que implementações em linguagens diferentes, sendo executadas em servidores de fornecedores distintos possam interagir sem precisar levar em conta como cada uma das implementações foi escrita. Christensen (2001) destaca que para prover a funcionalidade de disponibilizar serviços capazes de interagir entre si através da internet, além do XML, os serviços da *Web* utilizam outros padrões adicionais para definição e registro dos recursos a serem disponibilizados como a WSDL (*Web Services Description Language*) e o SOAP.

2.2 XML (EXTENDED MARKUP LANGUAGE)

¹ Aparato de hardware e/ou software presente em uma rede de computadores, que tem por finalidade aplicar regras de segurança a um determinado ponto da rede, esses sistemas geralmente estão associados a redes TCP/IP.

O XML é uma linguagem extensível cuja principal finalidade é a marcação de dados, e por isso é uma tecnologia amplamente utilizada atualmente para a padronização e transmissão de mensagens. A transmissão de mensagens e dados entre componentes de sistemas distribuídos tem sido uma área em que possui grande dificuldade quando existe a necessidade de interoperabilidade entre sistemas diferentes. Quando os componentes de um sistema pertencem à mesma arquitetura, como por exemplo CORBA (*Common Object Request Broker Architecture*) e estão registrados dentro da mesma estrutura de rede, a transmissão de dados não é um problema considerável, uma vez que essas arquiteturas possuem mecanismos que deixam essas operações transparentes. (ERL, 2009; PERAZZOLI, 2011).

Entretanto, quando aplicações de arquiteturas diferentes precisam “conversar” ou mesmo quando aplicações de mesma arquitetura precisam trocar dados através de estruturas de segurança, como *firewalls*, bem como quando aplicações de uma arquitetura precisam interagir com aplicações cuja estrutura é desconhecida pelo desenvolvedor, é fundamental que as informações trocadas entre as aplicações estejam padronizadas e em um formato conhecido, bem definido e estruturado. Diante disso Rodrigues (2001) destaca que:

O XML surgiu como uma boa solução, visando fornecer uma representação clara de dados, bem como um conjunto de validações e de regras bem definidas. (RODRIGUES, 2001).

Partindo destes postulados, o XML tornou-se um veículo excelente para embalagem de dados de maneira que ambos os lados da transmissão possam facilmente ler e compreender. Contudo, vale ressaltar que o XML tem seus inconvenientes; como por exemplo, que transmitir informações contidas em um XML provoca uma utilização da rede muito grande, e essa característica apesar de ser um preço particularmente pequeno a pagar com as atuais redes de alta velocidade, deve ser considerada seriamente a depender do tipo de aplicação que pretende desenvolver. É válido destacar que se a utilização da rede é um ponto crítico da aplicação, a utilização de XML na comunicação deve ser avaliada de forma extremamente criteriosa.

2.3 WSDL (*WEB SERVICES DESCRIPTION LANGUAGE*)

WSDL acrônimo para *Web Services Description Language* é a linguagem que utiliza XML para servir como um meio para fornecedores de serviços disponibilizarem suas interfaces de acesso. Estas interfaces, representadas em XML, contêm informações

sobre todas as funcionalidades do serviço, bem como os tipos de dados necessários para a sua utilização (comunicação), em outras palavras, tudo que um cliente precisa para saber como utilizar um serviço. A WSDL é definida como sendo a linguagem padrão para descrever a interface e a localização de um serviço. Especificamente WSDL fornece um número de peças-chave de informação podendo essas ser dispostas através da:

- Definição do formato das mensagens que são passadas entre dois pontos finais usando seus elementos *<types>* e *<message>* e definições apropriadas de esquemas (*schema*).
- Semântica do serviço: como ele deve ser chamado para fazer uma transmissão síncrona de solicitação e resposta, síncrona apenas de resposta ou comunicação assíncrona.
- Ponto final e o transporte do serviço através do elemento *<service>*: isto é, quem fornece o serviço.
- Ligação através do elemento *<binding>*, isto é, como o serviço é alcançado.

2.4 SOAP (SIMPLE OBJECT ACCESS PROTOCOL)

Segundo Rodrigues (2011), SOAP é um protocolo de comunicação leve, baseado na linguagem XML e concebido para utilização na troca de informações entre sistemas em um ambiente descentralizado, distribuído. Por esse motivo, este tornou-se um dos mais conhecidos formatos de mensagens e protocolo utilizado por *Web Services* baseados em XML.

Em consequente, o acrônimo SOAP é referido frequentemente como *Service-Oriented Architecture (or Application) Protocol* (Protocolo de Arquitetura Orientada a Serviços) ao invés de *Simple Object Access Protocol*. (OLIVEIRA, 2014).

SOAP é o protocolo de comunicação padrão dos *Web Services*. A especificação SOAP define como podem ser representados diversos tipos de dados através de XML, define também o padrão de representação de mensagens, chamadas RPC e a ligação com o protocolo HTTP. Salienta-se que o poder do SOAP vem do fato de ser simples, fácil de implementar, e bem suportado na indústria (SAGANICH, 2001). SOAP é fundamentalmente uma estrutura que consiste de uma combinação de informações textuais compartilhada via a Internet. A informação textual é codificada na forma de arquivo XML, que tem regras específicas para codificação e processamento. A transmissão real dos dados do XML é administrada pelo protocolo de transporte, que é

hoje geralmente HTTP servido por um servidor *Web*. A combinação do estilo aberto da estrutura da linguagem XML e a penetração do protocolo HTTP tornam o SOAP um protocolo de elevado grau de interoperabilidade. (RODRIGUES, 2001; ERL, 2009).

2.4.1. Interoperabilidade entre Sistemas e outros protocolos

Sonda e Montez (2006) descrevem que estruturas de segurança estabelecidas na Internet (*firewall*) tendem a ser por natureza, bastante seletivas no que diz respeito ao tráfego de informações através das mesmas. A utilização de padrões abertos e comercialmente difundidos é um fator que possibilita aos pacotes SOAP trafegarem normalmente através de tais estruturas. Tal funcionalidade, aliada a mecanismos de segurança, como criptografia, representa uma economia significativa à medida que dispensam canais especiais de comunicação para a interoperabilidade de sistemas. O próprio protocolo de transporte, responsável por conduzir os pacotes SOAP, geralmente HTTP, representa uma estrutura cômoda e consistente de transporte.

Quando desejamos obter interação entre sistemas corporativos dentro da própria estrutura da empresa, pode ser mais rentável em termos de performance do sistema utilizar um protocolo com melhor sintaxe de transferência, como por exemplo IIOP (*Internet Inter-Orb Protocol* – CORBA). Entretanto, se tratando de sistemas distribuídos através da Internet, SOAP será sempre um candidato com destaque. Em contrapartida, a interação com outros protocolos é uma possibilidade que resulta da natureza da estrutura do protocolo. A interoperabilidade entre protocolos pode trazer funcionalidades extras ao sistema e uma economia considerável.

Por exemplo, através da interoperabilidade IIOP/SOAP é possível fazer com que objetos CORBA possam interagir com serviços na Internet através de *firewalls*, e da mesma maneira permitir que sistemas baseados em CORBA possam interoperar com sistemas de outras arquiteturas. (RODRIGUES, 2001).

2.4.2. Arquitetura do SOAP

SOAP é um protocolo simples, comercialmente independente, e que não define a semântica da aplicação onde será utilizado como, por exemplo, o modelo de programação; ao invés disto, define uma estrutura simples para expressar a semântica da aplicação por meio de um mecanismo modular de empacotamento de informações e codificação dos tipos de dados contidos nos pacotes. Sempre é importante lembrar que SOAP em si é apenas um protocolo e não uma arquitetura completa de objetos

distribuídos. Por outro lado, arquiteturas completas de objetos distribuídos são projetadas em torno dos seus protocolos para maior eficiência (ERL, 2009).

A principal característica do SOAP é o objetivo de manter a simplicidade e extensibilidade. Isto faz com que vários recursos das arquiteturas atuais de objetos distribuídos não estejam incluídos na especificação do SOAP, e o torna um protocolo atrativo a um grande número de projetos. Por exemplo, alguns destes recursos são:

- *Garbage collection* distribuído.
- Passagem de objetos por referência (que requer *garbage collection* distribuído).

Também, diferente das arquiteturas contemporâneas de objetos distribuídos, SOAP usa abertamente tecnologias disponíveis que, quando combinadas, especificam um protocolo consistente, que pode ser utilizado para auxiliar, principalmente, sistemas distribuídos através da Internet.

O fato de o SOAP utilizar padrões bem definidos e aceitos no mercado, como por exemplo a linguagem XML na sua estrutura, também é estendido aos protocolos de comunicação (DURÃES, 2020).

Erl (2009) complementa e afirma que SOAP permite a sua utilização em combinação com uma variedade de outros protocolos de comunicação, como por exemplo HTTP (*HyperText Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*) e FTP (*File Transfer Protocol*).

O SOAP geralmente usa o protocolo HTTP para transportar, encapsulados em arquivos XML, dados serializados de argumentos de métodos, de sistema para sistema. Estes dados serializados de argumentos são utilizados no lado remoto para executar as chamadas aos procedimentos que o cliente fez àquele sistema (ERL, 2009).

2.5. API (APPLICATION PROGRAMMING INTERFACE)

API (*Application Programming Interface*) trata-se de um conjunto de rotinas e padrões estabelecidos e documentados por uma aplicação A, para que outras aplicações consigam utilizar as funcionalidades desta aplicação A, sem precisar conhecer detalhes da implementação do software. Desta forma, entendemos que as APIs promovem a interoperabilidade entre aplicações, em outras palavras, a comunicação entre aplicações e entre os usuários (PIRES, 2018).

2.6. REST e RESTful

O REST (*Representational State Transfer*), ou Transferência de Estado Representacional, trata-se de princípios/regras/*constraints* que, quando seguidas,

permitem a criação de um projeto com interfaces bem definidas. Desta forma, permitindo, por exemplo, que aplicações se comuniquem.

O REST é um “estilo arquitetônico” que basicamente explora a tecnologia existente e protocolos da *Web*. Enquanto RESTful é um serviço *Web* que utiliza esse paradigma, ou seja, é o termo normalmente usado para referir a implementação de *Web Services* que utilizam tal arquitetura. (OLIVEIRA, 2014; DURÃES, 2020; PIRES, 2018).

Existe uma certa confusão quanto aos termos REST e RESTful. Entretanto, ambos representam os mesmos princípios. A diferença é apenas gramatical. Em outras palavras, sistemas que utilizam os princípios REST são chamados de RESTful.

- **REST:** conjunto de princípios de arquitetura
- **RESTful:** capacidade de determinado sistema aplicar os princípios de REST.

3. RESULTADOS E CONCLUSÕES

Apesar do protocolo SOAP ser uma solução para uma série de problemas de integração entre sistemas distribuídos, uma visão crítica do protocolo envolvendo todos os aspectos da sua utilização nos permite facilmente perceber que, embora possa trazer grandes benefícios de imediato, ele não é a solução para todos os problemas e a decisão de se utilizar o SOAP como protocolo de comunicação do sistema distribuído deve ser tomada somente após uma cuidadosa avaliação da natureza e finalidade da aplicação, ambiente de execução, tipos de processamentos envolvidos e recursos de equipamentos disponíveis.

Sob a ótica de uma breve comparação entre as arquiteturas do SOAP e REST, por exemplo, temos alguns apontamentos que indicam algumas vantagens do REST em relação ao SOAP, tais como:

- O REST permite uma variedade maior de formatos de dados, enquanto o SOAP permite apenas XML;
- Juntamente com o JSON (*JavaScript Object Notation*) que normalmente funciona melhor com dados e oferece análise mais rápida, o REST é geralmente considerado mais fácil de trabalhar;
- Graças ao JSON, o REST oferece melhor suporte;
- O REST fornece desempenho superior, principalmente por meio do armazenamento em cache de informações que não são alteradas nem dinâmicas;
- É o protocolo usado com mais frequência em grandes serviços como Yahoo, Ebay, Amazon e até Google;

Através desse comparativo, vale também destacar que o REST geralmente é mais rápido e usa menos largura de banda. O REST é tido também como mais fácil integrar-se a sites existentes, sem a necessidade de refatorar a infraestrutura do site. Permitindo que os desenvolvedores trabalhem mais rápido e gastem menos tempo reescrevendo um site do zero, maximizando seu tempo, tendo em vista eles podem simplesmente adicionar novas funcionalidades.

Davis e Zhang (2004) destacam ainda que, embora o SOAP ofereça um valor de plataforma cruzada interoperabilidade, os desenvolvedores de aplicativos devem estar cientes de que os aplicativos baseados em SOAP podem sofrer um desempenho mais lento em comparação a aplicativos que usam tecnologias específicas de plataforma.

Contudo, apesar dos problemas abordados anteriormente, o SOAP, ainda assim é considerado de modo geral, uma boa aposta para os aplicativos que devem oferecer suporte a comunicações entre plataformas. Com o design certo, a degradação existente no desempenho pode ser mitigada.

De acordo com DAVIS e ZHANG, 2004, ainda há uma esperança de que novas ferramentas, como por exemplo analisadores XML e *frameworks* se tornem mais rápidas, de modo que possam ajudar a melhorar o desempenho do aplicativo SOAP bem como sua performance. Vale ressaltar que para os aplicativos que funcionam em uma única plataforma dentro de redes locais e não tiver a necessidade de se comunicar através de *firewalls*, não há de fato, razão óbvia de se utilizar o protocolo SOAP.

4. REFERÊNCIAS

ALVES, Rodrigo. S. **Comparação e Avaliação dos Protocolos NETCONF e SOAP para Configuração de Dispositivos**. Instituto de Informática. Porto Alegre: Ed. UFRGS, 2007.

BORGES, Fernando Oliveira. **Uma abordagem à interoperabilidade aplicada à integração de sistemas de redes sociais usando Web Services**. XIV Encontro Latino Americano de Iniciação Científica e X Encontro Latino Americano de Pós-Graduação. São Paulo, 2011.

CHRISTENSEN, Erik. **Web Services Description Language (WSDL) 1.1**. Microsoft Corporation & IBM Research, 2001.

DAVIS, Alexander; ZHANG, Du. **A comparative study of SOAP and DCOM**. The Journal of Systems and Software. 76. ed. California: Ed. Elsevier, 15 de junho de 2004.

DURÃES, Geison. **REST X SOAP, Qual o melhor tipo de integração**, 2020. Disponível em: <https://blog.tecnospeed.com.br/rest-x-soap/#:~:text=SOAP%20%C3%A9%20um%20sistema%20de,senhas%2C%20leads%20e%20objetos%20personalizados>. Acesso em: 27 de janeiro de 2021.

ERL, Thomas. **Web Service**, 2009. Disponível em: <https://www.devmedia.com.br/web-services/2873>. Acesso em: 30 de janeiro de 2021.

OLIVEIRA, Eric C. M. **Web Services: Java e XML juntos pela interoperabilidade**, 2020. Disponível em: <http://www.linhadecodigo.com.br/artigo/378/web-services-java-e-xml-juntos-pela-interoperabilidade.aspx>. Acesso em: 25 de janeiro de 2021.

OLIVEIRA, Rhaniel. **SOAP X REST**, 2014. Disponível em: <https://pt.slideshare.net/Rhaniel/arquitetura-soap>. Acesso em: 28 de janeiro de 2021.

PERAZZOLI, Kellen Kristine. **Integração de aplicações utilizando web servisse**, 2011.

PIRES, Jackson. **O que é API? REST e RESTful? Conheça as definições e diferenças**, 2018. Disponível em: <https://becode.com.br/o-que-e-api-rest-e-restful/>. Acesso em: 28 de janeiro de 2021.

REDHAT. **Integração REST X SOAP**. Disponível em: <https://www.redhat.com/pt-br/topics/integration/whats-the-difference-between-soap-rest>. Acesso em 30 de janeiro de 2021.

RODRIGUES, Wilhelm de Araújo. **Interoperabilidade entre serviços utilizando soap**. Centro de Ciências Formais e Tecnologia – CCFT. Aracaju, 2001.

SAGANICH, A. **Java and Web services, Part I**, 7 de Agosto de 2001.

SONDA, Gláucia C.S., MONTEZ, Carlos. **Uma Proposta de Implementação de Diferenciação de Serviços na Arquitetura de Web Services**, 2006. Disponível em https://www.academia.edu/1409373/Uma_Proposta_de_Implementa%C3%A7%C3%A3o_de_Diferencia%C3%A7%C3%A3o_de_Servi%C3%A7os_na_Arquitetura_de_Web_Services. Acesso em 30 de janeiro de 2021.