

1 QUADRO TEÓRICO

Neste capítulo serão descritos os principais conceitos e características das tecnologias a serem utilizadas para o desenvolvimento dos *softwares* propostos nos objetivos dessa pesquisa.

1.1 Web Services

Nos tempos atuais, com o grande fluxo de informação que percorre pelas redes da *internet* é necessário um nível muito alto de integração entre as diversas plataformas, tecnologias e sistemas. Como uma provável solução para esse ponto, já existem as tecnologias de sistemas distribuídos. Porém essas tecnologias sofrem demasiadamente com o alto acoplamento de seus componentes e também com a grande dependência de uma plataforma para que possam funcionar. Com intuito de solucionar a estes problemas e proporcionar alta transparência entre as várias plataformas, foram criados as tecnologias *web services*.

De acordo com Erl (2015):

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 não amarrada a fornecedor, o SOAP disponibilizou uma alternativa atrativa em relação aos protocolos proprietários tradicionais, tais como CORBA e DCOM.

Considera-se então a existência dos *web services* a partir daí. De acordo com Durães (2005), *Web Service* é um componente que tem por finalidade integrar serviços distintos. O que faz com que ele se torne melhor que seus concorrentes é a padronização do XML (*Extensible Markup Language*) para as trocas de informações. A aplicação consegue conversar com o servidor através do WSDL que é documento que contém as regras de funcionamento do *web service*.

Os *web services* além de fornecerem uma padronização de comunicação entre as várias tecnologias existentes, provê transparência na troca de informações. Isso contribui pelo fato de que as novas aplicações possam se comunicar com aplicações mais antigas ou aplicações contruídas sobre outras plataformas.

Além das tecnologias *web services* tradicionais, existe também os *web services* REST que também disponibilizam serviços, porém não necessitam de encapsulamento de suas mensagens assim como os *web Services* SOAP. Este fato influencia diretamente na performance da

aplicação como um todo, haja vista que não sendo necessário o encapsulamento da informação requisitada ao *web service*, somente é necessário o processamento e tráfego da informação que realmente importa. As características do padrão REST serão abordadas a seguir.

1.1.1 REST

Segundo Saudate (2012), REST é a sigla de *Representational State Transfer*, ou em português Transferência de Estado Representativo, desenvolvido por Roy Fielding na defesa de sua tese de doutorado. Segundo o próprio Fielding (2000) REST é um estilo que deriva do vários estilos arquitectónicos baseados em rede e que combinado com algumas restrições, fornecem uma interface simples e uniforme para fornecimento de serviços¹.

Rubbo (2015), afirma que os dados e funcionalidade de um sistema são considerados recursos e podem ser acessados através das URI's (*Universal Resource Identifier*), facilitando dessa forma a comunicação do servidor com o cliente.

Saudate (2012), explica ainda que os métodos do HTTP² podem fazer modificações nos recursos através dos comandos:

- GET – Para recuperar algum dado.
- POST – Para criar algum dado.
- PUT – Para alterar algum dado.
- DELETE – Para excluir algum dado.

Segundo Godinho (2009), não há um padrão de formato para as trocas de informações, mas as que mais são utilizadas é o XML³ e o JSON⁴. O REST⁵ é o mais indicado para aplicações em dispositivos moveis, devido sua agilidade.

¹ Tradução e resumo de informações de responsabilidade do autor da pesquisa.

² HTTP – Abreviação para *HyperText Transfer Protocol*

³ XML – Abreviação para *Extensible Markup Language*.

⁴ JSON – Abreviação para *JavaScript Object Notation*.

⁵ REST – Abreviação para *Representational State Transfer*.

1.2 *Apache Tomcat*

De acordo com a Tomcat (2015), *Apache Tomcat* é uma implementação de código aberto das especificações *Java Servlet* e *JavaServer Pages*. O *Apache Tomcat* é um *Servlet Container*, que disponibiliza serviços através de requisições e respostas.

Segundo Tomcat (2015), o projeto desse *software* começou com a *Sun Microsystems*, que em 1999 doou a base do código para *Apache Software Foundation* onde foi lançada a versão 3.0.

Desta forma, o cliente envia uma requisição através do seu navegador, o servidor por sua vez a recebe, executa o *servlet* e devolve a resposta ao usuário.

1.3 *Engenharia De Software*

1.3.1 *UML*

1.3.2 *Processos de Desenvolvimento de Software*

REFERÊNCIAS

DURÃES, R. : **Web Services para iniciantes**. 2005. Disponível em: <<http://imasters.com.br/artigo/3561/web-services/web-services-para-iniciantes/>>. Acesso em: 10 de Março de 2015.

ERL, T. : **Introdução às tecnologias Web Services: soa, soap, wsdl e uddi**. 2015. Disponível em: <<http://www.devmedia.com.br/introducao-as-tecnologias-web-services-soa-soap-wsdl-e-uddi-parte1/2873>>. Acesso em: 26 de Abril de 2015.

FIELDING, R. T. : **Architectural Styles and the Design of Network-based Software Architectures**. Tese (Doutorado) — University of California, 2000.

GODINHO, R. : **Criando serviços REST com WCF**. 2009. Disponível em: <<https://msdn.microsoft.com/pt-br/library/dd941696.aspx>>. Acesso em: 01 de Março de 2015.

RUBBO, F. : **Construindo RESTful Web Services com JAX-RS 2.0**. 2015. Disponível em: <<http://www.devmedia.com.br/construindo-restful-web-services-com-jax-rs-2-0/29468>>. Acesso em: 03 de Março de 2015.

SAUDATE, A. : **REST: construa api's inteligentes de maneira simples**. São Paulo: Casa do Código, 2012.

TOMCAT, A. : **The Tomcat Story**. 2015. Disponível em: <<http://tomcat.apache.org/heritage.html>>. Acesso em: 08 de Março de 2015.