

# Sistema de CEP e Rastreamento de Objetos dos Correios

## Exercício Computacional I - Sistemas Distribuídos

Rafael Gonçalves de Oliveira Viana<sup>1</sup>

<sup>1</sup>Sistemas de Informação – Universidade Federal do Mato Grosso do Sul (UFMS)  
Caixa Postal 79400-000 – Coxim – MS – Brazil

rafael.viana@aluno.ufms.br

**Resumo.** *Este relatório descreve como funciona um WebService assim como sua arquitetura, e relata como foi implementado um sistema cliente e um sistema servidor com interface gráfica para busca de CEP e rastreamento de encomendas dos Correios, utilizando Angular 2 e Nodejs.*

### 1. Introdução

### 2. Arquitetura de um WebService

Para poder identificar qual arquitetura que um *WebService* deve possuir devemos examinar os papéis individuais de cada ator no *WebService* e examinar a pilha emergente de protocolo que o *WebService* pretende utilizar. O mesmo pode ser publicado na intranet ou na Internet, provendo três possíveis formatos de serviços: Provedor de Serviço, Solicitante de Serviço e o Registro de Serviço.

#### 2.0.1. Provedor de serviço

Este é o fornecedor do serviço web. O provedor de serviços implementa o serviço e disponibiliza-o na Internet ou intranet.

#### 2.0.2. Solicitante de Serviço

Este é um consumidor do serviço web. O solicitante utiliza um serviço da Web existente abrindo uma conexão de rede e enviando uma solicitação XML.

#### 2.0.3. Registro de serviço

Este é um diretório de serviços logicamente centralizado. O registro fornece um lugar central onde os desenvolvedores podem publicar novos serviços ou encontrar os existentes. Ele serve como centro de compensação centralizado para empresas e seus serviços.

### 2.1. Pilha de protocolo de serviço da Web

Uma segunda opção para identificar qual arquitetura o *WebService* deve utilizar é examinar a pilha emergente de protocolo que pretende utilizar. Cada dia existe mais tipos de protocolos porém existem 4 tipos que se destacam: Serviço de Transporte - "FTP", Mensagens "XML", Descrição de Serviço "WSDL" e Descoberta de Serviço "UDDI"

### **2.1.1. Serviço de transporte**

Esta camada é responsável pelo transporte de mensagens entre aplicativos. Atualmente, esta camada inclui o protocolo de transporte de hipertexto (HTTP), protocolo de transferência de correio simples (SMTP), protocolo de transferência de arquivos (FTP) e protocolos mais recentes, como o protocolo de intercâmbio extensível de blocos (BEEP).

### **2.1.2. Mensagens XML**

Esta camada é responsável por codificar mensagens em um formato XML comum para que as mensagens possam ser entendidas em cada uma das extremidades. Atualmente, esta camada inclui XML-RPC e SOAP.

### **2.1.3. Descrição do Serviço**

Esta camada é responsável por descrever a interface pública para um serviço web específico. Atualmente, a descrição do serviço é tratada através do Web Service Description Language (WSDL).

### **2.1.4. Descoberta do serviço**

Esta camada é responsável por centralizar os serviços em um registro comum e fornecer funcionalidades fáceis de publicação / pesquisa. Atualmente, a descoberta do serviço é tratada através de Descrição Universal, Descoberta e Integração (UDDI).

## **2.2. Serviço de Transporte**

A parte inferior da pilha de protocolos do serviço web é o transporte de serviços. Essa camada é responsável por transportar mensagens XML entre dois computadores.

### **2.2.1. Hyper Text Transfer Protocol (HTTP)**

Atualmente, o HTTP é a opção mais popular para o transporte de serviços. O HTTP é simples, estável e amplamente implantado. Além disso, a maioria dos firewalls permitem o tráfego HTTP. Isso permite que mensagens XML-RPC ou SOAP se mostrem como mensagens HTTP. Isso é bom se você quiser integrar aplicativos remotos, mas eleva uma série de preocupações de segurança.

### **2.2.2. Bloqueia o protocolo de troca extensível (BEEP)**

Esta é uma alternativa promissora para o HTTP. O BEEP é uma nova estrutura da Task Force de Engenharia da Internet (IETF) para a construção de novos protocolos. O BEEP está em camadas diretamente no TCP e inclui uma série de recursos internos, incluindo um protocolo inicial de handshake, autenticação, segurança e tratamento de erros. Usando

BEEP, pode-se criar novos protocolos para uma variedade de aplicações, incluindo mensagens instantâneas, transferência de arquivos, distribuição de conteúdo e gerenciamento de rede.

O SOAP não está vinculado a nenhum protocolo de transporte específico. Na verdade, você pode usar SOAP via HTTP, SMTP ou FTP. Uma idéia promissora é, portanto, usar SOAP sobre BEEP.

### **3. Componentes de um WebService**

Ao longo dos últimos anos, três tecnologias primárias emergiram como padrões mundiais que constituem o núcleo da tecnologia de serviços da Web de hoje. Essas tecnologias são demonstradas abaixo.

#### **3.1. XML-RPC**

Este é o protocolo XML mais simples para trocar informações entre computadores.

1. XML-RPC é um protocolo simples que usa mensagens XML para executar RPCs.
2. Os pedidos são codificados em XML e enviados via HTTP POST.
3. As respostas XML são incorporadas no corpo da resposta HTTP.
4. O XML-RPC é independente da plataforma.
5. O XML-RPC permite que diversas aplicações se comuniquem.
6. Um cliente Java pode falar XML-RPC para um servidor Perl.
7. XML-RPC é a maneira mais fácil de começar com os serviços da Web.

#### **3.2. WSDL**

O WSDL é um idioma baseado em XML para descrever os serviços da Web e como acessá-los.

1. WSDL significa Web Services Description Language.
2. O WSDL foi desenvolvido conjuntamente pela Microsoft e pela IBM.
3. WSDL é um protocolo baseado em XML para troca de informações em ambientes descentralizados e distribuídos.
4. WSDL é o formato padrão para descrever um serviço web.
5. A definição WSDL descreve como acessar um serviço da Web e quais as operações que ele executará.
6. O WSDL é um idioma para descrever como se relacionar com serviços baseados em XML.
7. WSDL é parte integrante do UDDI, um registro de negócios mundial baseado em XML.
8. WSDL é o idioma que UDDI usa.
9. WSDL é pronunciado como "Wiz-Dull" e explicado como "WSD-L".

#### **3.3. SOAP**

O SOAP é um protocolo baseado em XML para trocar informações entre computadores.

1. O SOAP é um protocolo de comunicação.
2. O SOAP é para comunicação entre aplicativos.
3. O SOAP é um formato para enviar mensagens.

4. O SOAP é projetado para se comunicar via Internet.
5. O SOAP é independente da plataforma.
6. O SOAP é independente da linguagem.
7. O SOAP é simples e extensível.
8. O SOAP permite que você percorra os firewalls.
9. O SOAP será desenvolvido como um padrão W3C.

### 3.4. Exemplo Ilustrativo

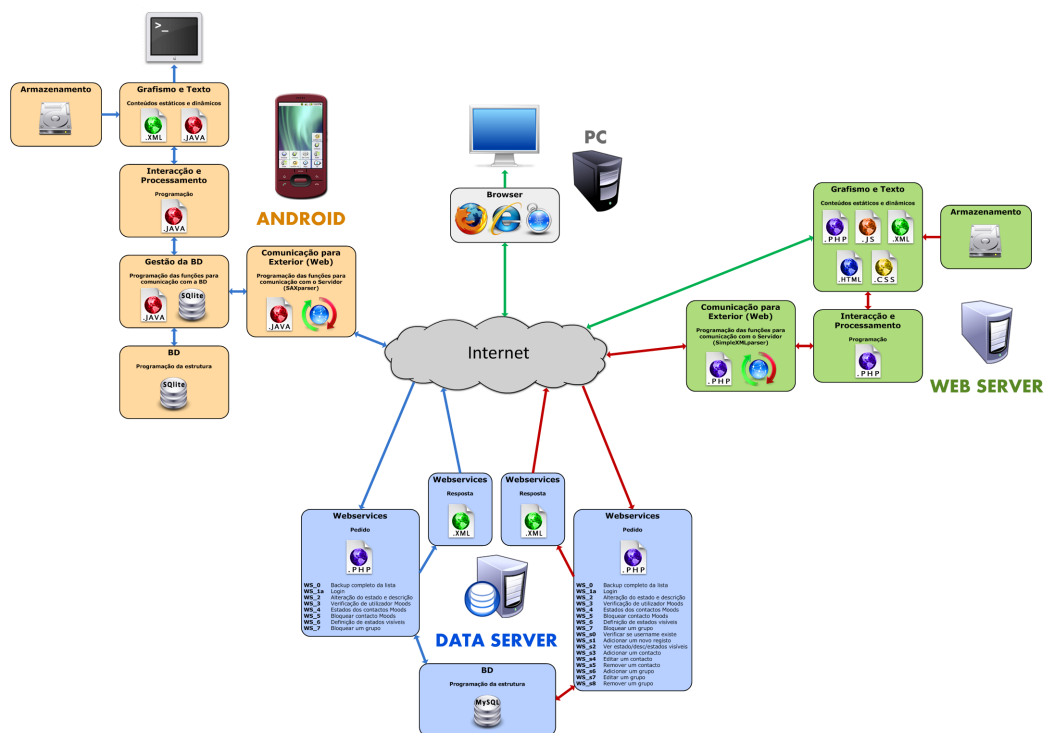


Figura 1. Legenda

## 4. Metodologia

### 4.1. Tecnologias utilizadas

### 4.2. Front-End

### 4.3. Back-end

### 4.4. Hospedagem

## 5. Conclusão