

Bare Demo of IEEEtran.cls for IEEE Journals

Michael Shell, *Member, IEEE*, John Doe, *Fellow, OSA*, and Jane Doe, *Life Fellow, IEEE*

Abstract—Este relatório introduz a arquitetura de um *Web Service* assim como seus componentes, o mesmo relata como foi implementado um Web Server que possui uma página web e serviços para busca de cidade e bairro através do CEP e rastreamento de encomendas dos correios.

Index Terms—IEEE, IEEEtran, journal, L^AT_EX, paper, template.

I. INTRODUÇÃO

Com a disseminação de dispositivos com conexão à Internet na sociedade atual, houvesse a necessidade de criar/gerenciar serviços web, estes necessários para atender as mais diversas funções como exemplo, salas de bate-papo, e-commerces e serviços meteorológicos entre outros.

Este relatório apresenta brevemente a arquitetura de um *Web Service* e como foi implementado, serviços de consulta de cidade e bairro através do CEP e rastreamento de encomendas dos correios, juntamente com uma página web e um aplicativo android que oferece uma interface para os serviços disponíveis.

O código deste relatório está em sua totalidade no endereço <http://github.com/rafaelgov95/SD/Projeto-Correios-CEP>, e o mesmo pode ser reutilizado sobre a licença MIT.

II. WEB SERVICE

Web service uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes, como pode ser observado na figura 1, um *Data Server*, disponibiliza serviços para os mais diversos tipos de dispositivos, em diferentes formatos de objetos.

Com esta tecnologia possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. [?].

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see <http://www.michaelshell.org/contact.html>).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

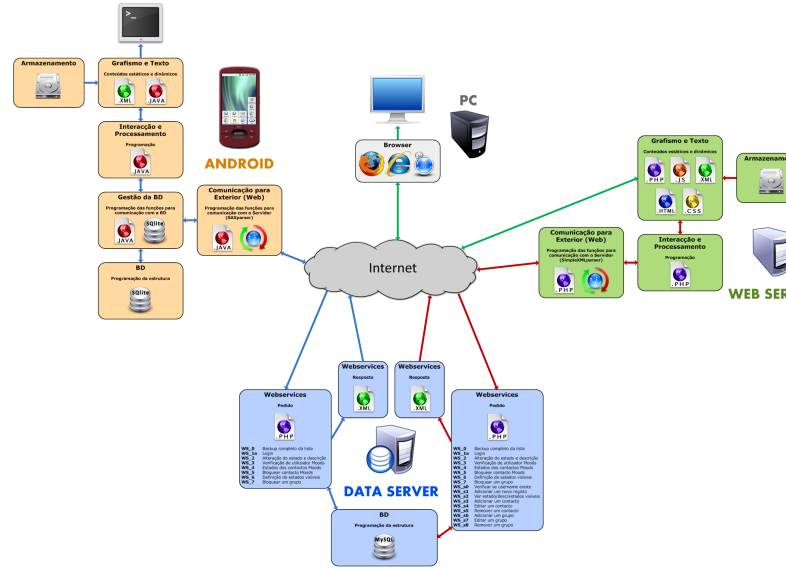


Fig. 1. Sistema implantando, fornecendo serviços para PCs, dispositivos móveis e outros web servers.

Cada dia existe mais tipos de protocolos por existirem 4 tipos que se destacam: Serviço de Transporte - "FTP", Mensagens "XML", Descrição de Serviço "WSDL" e Descoberta de Serviço "UDDI".

Podemos observar na figura 2, a pilha de soluções com a pilha de tecnologia correspondente para cada camada.

- 1) **Descoberta do serviço:** 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) **Descrição do Serviço:** 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).
- 3) **Mensagens XML:** 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.
- 4) **Serviço de transporte:** 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 proto-

colos mais recentes, como o protocolo de intercambio extensvel de blocos (BEEP).

Pilha de Tecnologias



Pilha de Soluções

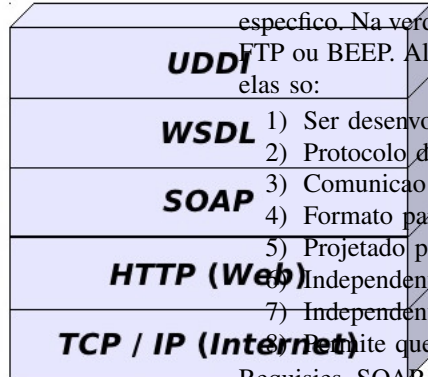


Fig. 2. Pilha de protocolo de transporte e suas tecnologias.

A. Tipos de Web Services

O mesmo pode ser publicado na intranet ou na Internet, provendo trs possveis formatos de servios: Provedor de Servio, Solicitante de Servio e o Registro de Servio. Para mais detalhes desta sesso incentive a consultar a referncia [?].

- 1) **Provedor de servio:** Fornece servios web. O provedor de servios implementa o servio e disponibiliza-o na Internet ou intranet.
- 2) **Solicitante de Servio:** Este um consumidor do servio web. O solicitante utiliza um servio da Web existente abrindo uma conexo de rede e enviando uma solicitao XML.
- 3) **Registro de servio:** Este um diretrio de servios logicamente centralizado. O registro fornece um lugar central onde os desenvolvedores podem publicar novos servios ou encontrar os existentes. Ele serve como centro de compensao centralizado para empresas e seus servios.

B. Componentes de um Web Service

Ao longo dos ltimos anos, diversas tecnologias emergiram como padres mundiais que constituem o ncleo da tecnologia de servios da Web de hoje. Algumas destas tecnologias e suas caractersticas so demonstradas abaixo.

1) **WSDL - Web Services Description Language :** O WSDL um idioma baseado em XML, para descrever os servios da Web e como acess-los, nas sesses III-B e III-C1, ser apresentado dois WSDL dos correios, com diversos servios.

- 1) um protocolo baseado em XML, para troca de informaes em ambientes descentralizados e distribudos.
- 2) o formato padro para descrever um servio web.
- 3) Descreve como acessar um servio da Web e quais as operaes que ele executar.
- 4) Descrever como se relacionar com servios baseados em XML.

5) o idioma que UDDI utiliza.

O WSDL tem um grande papel na arquitetura de um *Web Service*, ele divulga e expem os servios presentes.

2) **SOAP - Simple Object Access Protocol:** O SOAP um protocolo baseado em XML para trocar informaes entre computadores.

O mesmo no est vinculado a nenhum protocolo de transporte especfico. Na verdade, voc pode usar SOAP via HTTP, SMTP, FTP ou BEEP. Algumas caractersticas esto presentes no SOAP elas so:

- 1) Ser desenvolvido como um padro W3C.
- 2) Protocolo de comunicao.
- 3) Comunicao entre aplicativos.
- 4) Formato para enviar mensagens.
- 5) Projetado para se comunicar via Internet.
- 6) Independente da plataforma.
- 7) Independente da linguagem.
- 8) Permite que voc percorra os firewalls.

Requisitos SOAP sero utilizadas nas sesses III-B e III-C1, onde foi utilizado o node-soap uma biblioteca de SOAP para NodeJs.

3) **REST - Representation State Transfer:** uma arquitetura criada para ser mais simples de se usar que o SOAP. Pode ser usado em vrios formatos de texto, como CSV (Comma-separated Values), RSS (Really Simple Syndication), JSON e YAML. Porm, s pode ser utilizado com o protocolo HTTP/HTTPS, por exemplo utilizando os mtodos GET, POST, PUT e DELETE.

- 1) Melhor curva de aprendizado.
- 2) Mensagens menores e mais eficientes como o formato JSON comparado com XML.
- 3) Os dados podem ser colocados em cache, retornando sempre a mesma resposta para a mesma requisio.
- 4) Mais rpido pois precisa de menos processamento que o SOAP.

Como os correios no disponibiliza servios por REST, foi desenvolvido um servio alternativo que ser detalhado na sesso III-C2.

III. Web Service - RAFAEL BUSCAS

Nesta sesso relato como foi o desenvolvimento e a implantao de um *Web Service* em um *Web Server*, que promove servios de busca de CEPs e rastreamento de encomendas dos correios, utilizando uma pgina web e um aplicativo android como interface para consultas.

A. Web Server

O *Web Server* foi implementado utilizando o NodeJS com o framework ExpressJs, onde se cria um ambiente de desenvolvimento para criao de servios web.

O Node.js usa um modelo de E/S no bloqueante, que o torna leve e eficiente. O ecossistema de pacotes Node.js, npm, o maior ecossistema de bibliotecas de cdigo aberto do mundo [?].

O Express um framework para aplicativo da web do Node.js mnimo e flexvel que fornece um conjunto robusto de recursos para aplicativos web e mvel [?].

B. Web Service de Busca de CEP

A busca de cidade e bairro pelo CEP est sendo realiza por um servio disponvel, no WSDL dos correios com a url <https://apps.correios.com.br/SigepMasterJPA/AtendeClienteService/AtendeCliente?wsdl>.

Conforme pode-se observar na figura 3, existem diversos servios neste WSDL, porm s ser utilizado o servio de consulta de cep, com o nome *consultaCEP*.

```
<xs:element name="buscaTarifaValeResponse" type="tns:buscaTarifaValeResponse"/>
<xs:element name="calculaTarifaServico" type="tns:calculaTarifaServico"/>
<xs:element name="calculaTarifaServicoResponse" type="tns:calculaTarifaServicoResponse"/>
<xs:element name="cancelarObjeto" type="tns:cancelarObjeto"/>
<xs:element name="cancelarObjetoResponse" type="tns:cancelarObjetoResponse"/>
<xs:element name="cancelarPedidoScol" type="tns:cancelarPedidoScol"/>
<xs:element name="cancelarPedidoScolResponse" type="tns:cancelarPedidoScolResponse"/>
<xs:element name="consultaCEP" type="tns:consultaCEP"/>
<xs:element name="consultaCEPResponse" type="tns:consultaCEPResponse"/>
<xs:element name="consultaSRO" type="tns:consultaSRO"/>
<xs:element name="consultaSROResponse" type="tns:consultaSROResponse"/>
<xs:element name="consultarPedidosInformacao" type="tns:consultarPedidosInformacao"/>
```

Fig. 3. Um pedao dos servios do WSDL dos correios, com o servio consultaCEP em azul escuro.

Utilizando o node-soap (j comentado na sesso II-B2), uma requisio HTTP-GET de consulta de cep encaminhada para Web Serve Rafael Buscas, na url <http://rafaelbuscas.ddns.net/api/correios/json/cep/Codigo> e recebida por um servio web, que dispara uma requisio de consulta de cep, por SOAP para o servio de consulta dos correios, onde aps retorno da mesma as informaes so repassadas para a requisio inicial.

As linhas de cdigo que recebe a requisio incial, conecta-se ao WSDL dos correios e reenvia a consulta, pode ser observada na figura 4.

```
1 app.get('/api/correios/json/cep/:cep', function (req, res) {
2   var url = "https://apps.correios.com.br/SigepMasterJPA/";
3   var options = {
4     ignoredNamespaces: {
5       namespaces: ['targetNamespace', 'typedNamespace']
6       override: true
7     }
8   };
9   var cep = req.params.cep.replace("/\D/", '');
10  if (cep.length == 8) {
11    soap.createClient(url, options, function (err, client) {
12      if (err) {
13        res.json(false);
14      } else {
15        client.consultaCEP({ cep: req.params.cep }, function (errCli, result) {
16          var resp = [];
17          resp.push(errCli ? false : result['return'])
18          res.json(resp);
19        });
20      }
21    });
22  } else {
23    res.json(false);
24  }
25 });
26
```

Fig. 4. Cdigo de rota, para consulta de cep presente no Web Server Rafael Buscas.

Para mais informaes sobre consulta de cep e outros servios disponveis por esse WSDL consultar a referncia [?].

C. Web Service para rastreamento de encomendas dos Correios

O rastreamento por encomendas dos correios, desenvolvido nesta sesso, dispe de duas tecnologias diferentes de empacotamento.

A primeira utiliza a tecnologia SOAP e ser apresentada na sesso III-C1, a segunda funcional porm alternativa utiliza uma requisio REST HTTP-POST, para obter uma requisio da pgina oficial dos correios, aps receber a resposta em um objeto HTML, realizado um parsear na resposta para extrair os dados da encomenda, est forma de rastreamento ser apresentada na sesso III-C2.

1) Servio de Rastreamento Encomenda via SOAP:

O WSDL dos correios que fornece o servio de rastreamento de encomenda conhecido como BuscaEventos, demonstrado na figura 5, est disponvel na url <http://webservice.correios.com.br/service/rastro/Rastro.wsdl>.

```
<message name="buscaEventos">
  <part name="parameters" element="tns:buscaEventos"></p>
</message>
<message name="buscaEventosLista">
  <part name="parameters" element="tns:buscaEventosLista"></p>
</message>
<message name="buscaEventosResponse">
  <part name="parameters" element="tns:buscaEventosRespo"></p>
</message>
<message name="buscaEventosListaResponse">
  <part name="parameters" element="tns:buscaEventosLista"></p>
</message>
```

Fig. 5. Um pedao dos servios do WSDL de rastreamento de encomendas dos correios, com o servio BuscaEventos em azul escuro.

Na figura 6 utilizando o NodeJS ea biblioteca node-soap, uma solicitao para o Web Servece Rafael Buscas com a url <http://rafaelbuscas.ddns.net/api/correios/json/objeto/CodigoDoRastreament> cria uma requisio SOAP, que essa enviada para o Web Service de rastreamento de encomendas, fornecido pelos correios, que retorna uma mensagem JSON que posteriormente e repassada para requisio incial.

Para mais informaes referente a API de servios dos correios, incentivo consultar a referncia [?].

```

1 app.get("/api/correios/json/objeto/:code", function(req, res) {
2   var url = "http://webservice.correios.com.br/comanda/objeto/:code";
3   var objeto = req.params.code;
4   soap.createClient(url, function(err, client) {
5     if (err) {
6       res.send("ERRO");
7     } else {
8       client.buscaEventos({
9         usuario: "ECT", senha: "SR0", tipo: "L",
10        resultado: "T", lingua: 101, objetos: objeto
11      }, function(errCli, result) {
12        var resp = [];
13        resp.push(errCli ? false : result['return'])
14        res.json(resp);
15      });
16    }
17  });
18 });

```

Na figura 8 consta um mtodo que realiza uma requisio REST - HTTP-POST no Web Server da pgina web dos correios.

```

1 request: function (req) {
2   var correios = {
3     uri: "http://www2.correios.com.br/sistemas/rastreamento",
4     form: {
5       objetos: req.params.code
6     },
7     method: 'POST',
8     headers: {}
9   };
10  response = requestPromise(correios);
11  return response
12 }

```

Fig. 6. Cdigo de rota, para consulta de encomendas presente no Web Server Rafael Buscas.

Esse servio disponibilizado pelos correios apesar de ser gratuito, somente retorna a ltima atualizao do objeto.

2) *Servio de Rastreamento de Encomendas alternativo utilizando Crawling na Pgina dos Correios*: Existe um servio disponvel na pgina dos correios para rastreamento de encomendas, onde possvel realizar um crawling na pgina, para obter um objetivo HTML, contendo as informaes da encomenda buscada.

O servio est disponvel na url "http://www2.correios.com.br/sistemas/rastreamento/", a imagem 7 demonstra como pode-se utilizar a ferramenta Postman, para realizar uma requisio REST HTTP-POST, direto para pgina de rastreamento, com objetivo de extrair informaes da mesma.

```

1 request: function (req) {
2   var correios = {
3     uri: "http://www2.correios.com.br/sistemas/rastreamento",
4     form: {
5       objetos: req.params.code
6     },
7     method: 'POST',
8     headers: {}
9   };
10  response = requestPromise(correios);
11  return response
12 }

```

Fig. 8. Criando Requisio em Java Script.

Após a requisio da figura 8, teremos uma varivel contendo todo o HTML da pgina web, que foi obtida pela requisio.

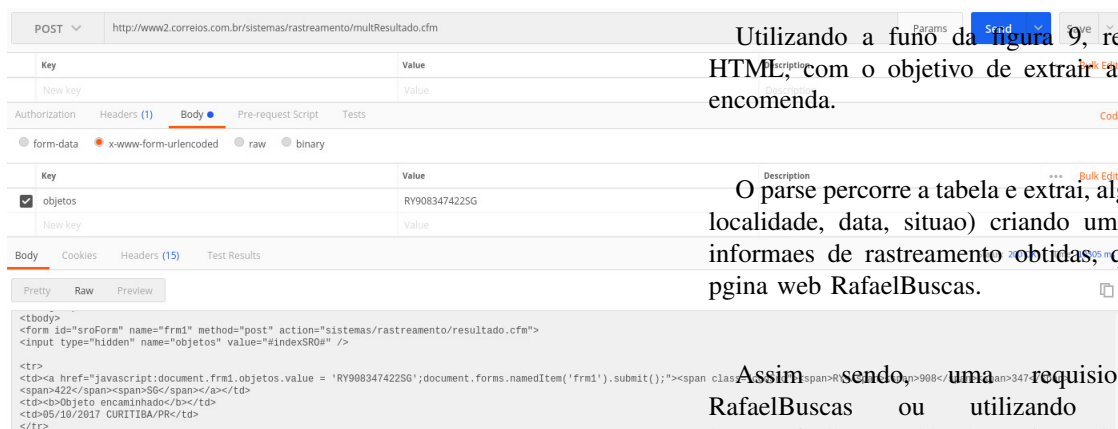


Fig. 7. Postman realizando um requisio REST HTTP-POST.

Para consumir o servio da pgina dos correios, primeiramente, foi desenvolvindo um Web Service com NodeJs e o framework ExpressJs, junto com a biblioteca request-promise para requisio JavaScript e a biblioteca cheerio para realizar

Utilizando a funo da figura 9, realizamos um parse no HTML, com o objetivo de extrair as informaes referente a encomenda.

O parse percorre a tabela e extrai, algumas informaes (cdigo, localidade, data, situao) criando um objeto JSON, com as informaes de rastreamento obtidas, que ser consumido pela pgina web RafaelBuscas.

Assim sendo, uma requisio atravs da pgina RafaelBuscas ou utilizando diretamente a url `http://rafaelbuscas.ddns.net/api/correios/json/aobjeto/codigo-de-rastreamento`, primeiramente a requisio recebida pelo Web Service (que est aguardando na rota da url acima), onde o mesmo envia uma requisio HTTP-POST, para a pgina de rastreamento dos correios com o cdigo de encomenda fornecido, aps o parse realizado as informaes so encaminhadas em um objeto JSON, como resposta para requisio inicial.


```

1  parser: function (data) {
2      var $ = cheerio.load(data);
3      var objetos = [];
4      var tableObjetos = $('table').find('tr');
5      $(tableObjetos).map(function (key, objeto) {
6          objeto = $(objeto).children('td').map(function (key, field) {
7              return $(field).text();
8          }).toArray();
9          if (objeto[0]) {
10             var rastreo = {
11                 codigo: null,
12                 situacao: null,
13                 local: null,
14                 data: null
15             };
16             rastreo.codigo = objeto[0].trim();
17             if (objeto[2]) {
18                 rastreo.situacao = objeto[1];
19                 rastreo.local = objeto[2].substr(11, objeto[2].length).trim();
20                 rastreo.data = objeto[2].substr(10, 10).trim();
21             } else {
22                 rastreo.situacao = 'Objeto ainda não consta no sistema';
23             }
24             objetos["return"] = rastreo;
25         }
26     });
27     return Object.assign({}, objetos["return"]);
28 }
29

```

Fig. 9. Criando um Crawling para parsear as tags do HTML.



Serviços da Página

RY907728402SG

Resultado do Rastreo SOAP

#	Código	Data	Local	Situação	Destino
1	RY907728402SG	02/10/2017	CURITIBA	Objeto encaminhado	Fiscalizacao Receita Federal Do Brasil

Fig. 10. Página Web Rafael Buscas, demonstração de busca de encomendas via SOAP.

Além da página web, um aplicativo para plataforma Android (figura 11), foi desenvolvido para demonstrar a portabilidade dos serviços desenvolvidos neste documento, o mesmo possui as mesmas funcionalidades da página web.



Fig. 11. Aplicativo Android Rafael Buscas, demonstração de busca de encomendas via SOAP.

IV. PÁGINA RAFAEL BUSCAS E APLICATIVO ANDROID

RAFAEL BUSCAS

Para fins didáticos, os serviços de rastreamento de encomendas e buscas de CEP pelo código, relatados estão disponíveis na URL <http://rafaelbuscas.ddns.net>, a mesma possui uma página web como interface para os serviços, estes que estão hospedados em uma máquina virtual do Google Cloud.

Na página inicial no canto inferior, existe um input para o código do CEP ou do Rastreamento (fornecidos pelos correios).

Um componente para seleção de consulta entre Cep ou Rastreamento, com o nome de "TIPO", deve ser marcado como CEP, Rastreo REST ou Rastreo SOAP, antes da busca (boto com uma lupa) como demonstrado na Figura 10.

V. CONCLUSÃO

Neste relatório foi apresentada a estrutura básica de um *Web Service*, e como o próprio pode ser utilizado para consumir serviços de outros *Web Services* disponíveis por outras empresas, outra abordagem exposta seria utilizar os mesmos para criação de um novo serviço, em seu próprio *Web Server*, para atender as mais diversas necessidades, como exemplo o rastreamento de encomendas ou obter a cidade e o bairro através de um CEP.