

## Nesta seção você encontra artigos para iniciantes na linguagem Delphi

### Consumindo Web Services

Aprenda como utilizar na prática recursos de um Web Services e como tirar proveito deles

**Maikel Marcelo Scheid**

(maikelscheid@gmail.com)

é técnico em Informática com ênfase em Análise e Programação de Sistemas. Atua na área de Desenvolvimento de Softwares em Delphi para plataforma Win32 e .NET com banco de dados Firebird e MS SQL. É membro da Equipe Editorial ClubeDelphi.

**Edinei Daniel Steffen**

(edineidaniel@gmail.com)

é técnico e Bacharelando em Sistemas de Informação, Analista e Desenvolvedor de Sistemas. Atua na área de Desenvolvimento de Softwares em Delphi para plataforma Win32 e .NET com banco de dados Firebird e MS SQL.

Com o advento da Internet e seu rápido avanço tanto na parte tecnológica quanto em serviços, cada vez mais existe a necessidade de compartilhamento de informações. Serviços estes que podem ser responsáveis por um simples cálculo, uma simples consulta ou até mesmo uma avançada transação de produtos entre empresas, necessitando apenas de um Web Service, ou seja, uma aplicação que funcione com a utilização de um serviço público na internet ou na rede local.

Lembrando que Web Service não traz consigo o conceito de aplicações com interfaces gráficas para usuários, mas sim serviços que são acessados e disponibilizados em uma linguagem universal XML (“Extensible Markup Language”), ou linguagem de marcação, possibilitando com isso uma comunicação padronizada em um único modelo (XML). Comunicação esta que normalmente é feita através do protocolo HTTP (“Hypertext

Transfer Protocol” – “Protocolo de Transferência de Hipertexto”), fazendo com que sistemas de diferentes linguagens e plataformas troquem dados entre si. Um termo bastante comentado no mundo dos analistas, desenvolvedores e arquitetos de software, mas pouco posto em prática devido à falta de entendimento da prática de utilização do mesmo.

Veremos neste artigo que a implementação desta tecnologia utilizando o Delphi 7. Para tanto faremos o uso e descrição dos serviços de envio de e-mails, conversor de temperaturas e o serviço de consulta de CEP e cidades e a partir destes desenvolveremos as conexões e posteriormente faremos a utilização dos mesmos.

---

**Nota:** A criação e utilização de Web Services está disponível no Delphi a partir da versão 6.0 e hoje encontramos em todas as versões disponíveis incluindo RAD Studio 2007.

---

**Nota:** O escopo deste artigo tem como finalidade mostrar apenas como consumir, ou seja, utilizar Web Services já disponíveis no mercado. Está fora do escopo mostrar como criar WS.

## Desmistificando a tecnologia

Devido a essa tecnologia trabalhar com a transferência de dados através do protocolo HTTP, não existe motivo algum para pensar que esses dados estarão expostos na Internet, pois os mesmos permanecem encapsulados pelo protocolo SOAP ("Simple Object Access Protocol"), fazendo com que assim exista um isolamento entre o serviço e o banco de dados. Assim sendo, para existir uma comunicação com qualquer Webservice é necessário uma implementação do protocolo SOAP determinado no W3C (órgão que desenvolve padrões para a criação e a interpretação dos conteúdos para a Web).

Antes de partirmos para a prática em si, necessitamos de alguns esclarecimentos sobre os termos que serão envolvidos durante a utilização de um Web Services:

- **WSDL** ("Web Services Description Language"), definida como sendo a linguagem de um Webservice, baseado em XML, contendo os métodos, operações e o modo de como acessá-lo;
- **UDDI** ("Universal Description, Discovery and Integration"), é uma especificação que define um serviço de Web Services, nesta contém informações sobre provedores, implementações e serviços;

## Analogia para o Uso

Vamos pensar de uma forma diferente para representar a utilização de um Webservice.

- Imagine-se necessitando de dinheiro;
- Você procura na sua cidade ("UDDI") o melhor banco;
- No banco você escolhe uma linha de crédito ("WSDL");
- Em seguida, você faz o pedido ao atendente ("SOAP"), que leva o pedido até a gerência ("HTTP");
- O gerente prepara a análise de crédito para sua escolha ("Classe que implementa o serviço");

• Depois de pronta, o atendente traz a papelada para você usufruir do dinheiro ("SOAP de resposta");

• É, basicamente, assim que funcionam os Web Services.

## Fazendo o uso dos Serviços

Iremos utilizar para este artigo a versão do Delphi 7 em uma aplicação Win32, mas fique à vontade para o uso de qualquer outra versão superior para dar início à implementação Web Services em seus sistemas.

### Exemplo 01: Envio de E-mails

No menu *File|New>Application* crie uma nova aplicação, altere a propriedade *Name* do formulário inicial para "frmWSEmail" e a propriedade *Caption* para "Web Service 01: Envio de Email". Salve a *Unit* principal do projeto como "uWSEmail.pas" e o projeto salve como "prjWSEmail.dpr".

Em seguida vamos à montagem do layout do formulário, adicione um *Label* no formulário com sua propriedade *Caption* para "Nome do Remetente". Ao lado do *Label* adicione um *Edit* ("edtNomeRem") e limpe a propriedade *Text*. Abaixo adicione uma nova *Label* e em sua propriedade *Caption* configure para "E-mail do Remetente" e ao lado acrescente um *Edit* ("edtEmailRem"). Após isso adicione um *Label* no formulário com sua propriedade *Caption* igual a "Nome do Destinatário". Da mesma forma ao lado da *Label* adicione um *Edit* ("edtNomeDes") e limpe a propriedade *Text*. Abaixo adicione uma nova *Label* e em sua propriedade *Caption* altere para "E-mail do Destinatário" acrescentando ao seu lado um *Edit* ("edtEmailDes"). Necessitando ainda um campo onde o usuário informa o assunto seguindo o mesmo modelo dos demais campos utilizando um *Label* e um *Edit* ("edtAssunto"). E por fim, utilizaremos um componente *Memo* onde será possível o usuário digitar a mensagem do e-mail, portanto adicione um componente *Memo* ("memMensagem"). Adicione também um botão que será responsável por acionar o Web Services (**Figura 1**).

Agora deve ser importado no projeto o **WSDL** ("Web Services") que disponibiliza o serviço de envio de e-mail,

ClubeDelphi PLUS

www.devmedia.com.br/clubedelphi/portal.asp

Acesse agora o mesmo o portal do assinante ClubeDelphi e assista a uma vídeo aula de Guinther Pauli que mostra como trabalhar Web Services e SOAP.

<http://www.devmedia.com.br/articles/viewcomp.asp?comp=6733&hl=>

ClubeDelphi PLUS

www.devmedia.com.br/clubedelphi/portal.asp

Acesse agora o mesmo o portal do assinante ClubeDelphi e assista a uma vídeo aula de Guinther Pauli que mostra como trabalhar Web Services e SOAP.

<http://www.devmedia.com.br/articles/viewcomp.asp?comp=6771&hl=>



## Nota do DevMan

Segundo o verbete [pt.wikipedia.org/wiki/Soap](http://pt.wikipedia.org/wiki/Soap): SOAP (originado do acrônimo inglês Simple Object Access Protocol) é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída, utilizando tecnologias baseadas em XML. Sua especificação define um framework que prevê maneiras para se construir mensagens que podem trafegar através de diversos protocolos e que foi especificado de forma a ser independente de qualquer modelo de programação ou outra implementação específica. Por não se tratar de um protocolo de acesso a objetos, o acrônimo não é mais utilizado.

Geralmente servidores SOAP são implementados utilizando-se servidores HTTP, embora isto não seja uma restrição para funcionamento do protocolo. As mensagens SOAP são documentos XML que aderem a uma especificação fornecida pelo órgão W3C ("Órgão que regulamenta e cria padrões de internet").

O primeiro esforço do desenvolvimento do SOAP foi implementar RPC's sobre XML.

**Nota:** A Chamada de procedimento remoto ou RPC ("Remote Procedure Call") é o tipo de protocolo para chamada remota de procedimentos em qualquer lugar da rede ou uma chamada de função para o método de transferência de controle de parte de um processo para outro, permite a divisão de um software em várias partes, compartilhamento de arquivos e diretórios. O protocolo RPC pode ser implementado sobre diferentes protocolos de transporte, o RPC não especifica como a mensagem é enviada, somente especifica e interpreta. As camadas de transporte que o RPC implementa são a TCP e a UDP.

RPC e Sockets têm a função de implementar numa aplicação as funções de rede, mas fazem isto de maneiras diferentes.

### Definição

Envelope das mensagens, regras de codificação, convenção RPC, ligação com protocolos subjacentes.

O SOAP oferece

- \* Mecanismo para definir a unidade de comunicação;
- \* Mecanismo para lidar com erros;
- \* Mecanismo de extensão que permite evolução;
- \* Mecanismo entre as mensagens SOAP e o HTTP, representar tipos de dados em XML;

Figura 1. Layout do Formulário

o link deste serviço é [www.abysal.com/soap/AbysalEmail.wsdl](http://www.abysal.com/soap/AbysalEmail.wsdl). Se digitar esse endereço no *browser*, notará que ele lhe retornará uma página semelhante à **Figura 2**. Esse retorno nada mais é do que o arquivo XML com as configurações do Web Services.

A importação do WSDL no Delphi é extremamente simples, basta ir até o menu *File|New>Other>Web Services>WSDL Importer* (**Figura 3**).

Ao confirmar a caixa de diálogo, o Delphi abrirá outra janela onde é preciso informar o link do WSDL (**Figura 4**). Na última tela do assistente já teremos uma pré-visualização dos métodos que compõem a interface do Web Services. Clicando no botão *Finish* o Delphi criará uma *Unit* com os métodos de acesso e interação com o serviço, e neste momento já é possível salvar, na mesma pasta do projeto, esta nova *unit*, com o nome sugerido pelo Delphi.

Basta agora utilizar o serviço, mas para isso necessitamos de um componente "*HTTPRIO*", que é o componente responsável por transformar chamadas a métodos de interface, escritas em Pascal, em requisições SOAP. Em seguida, configurar sua propriedade *Url*, com o endereço do serviço, conforme ilustrado na **Figura 4**. Por isso inclua o compo-

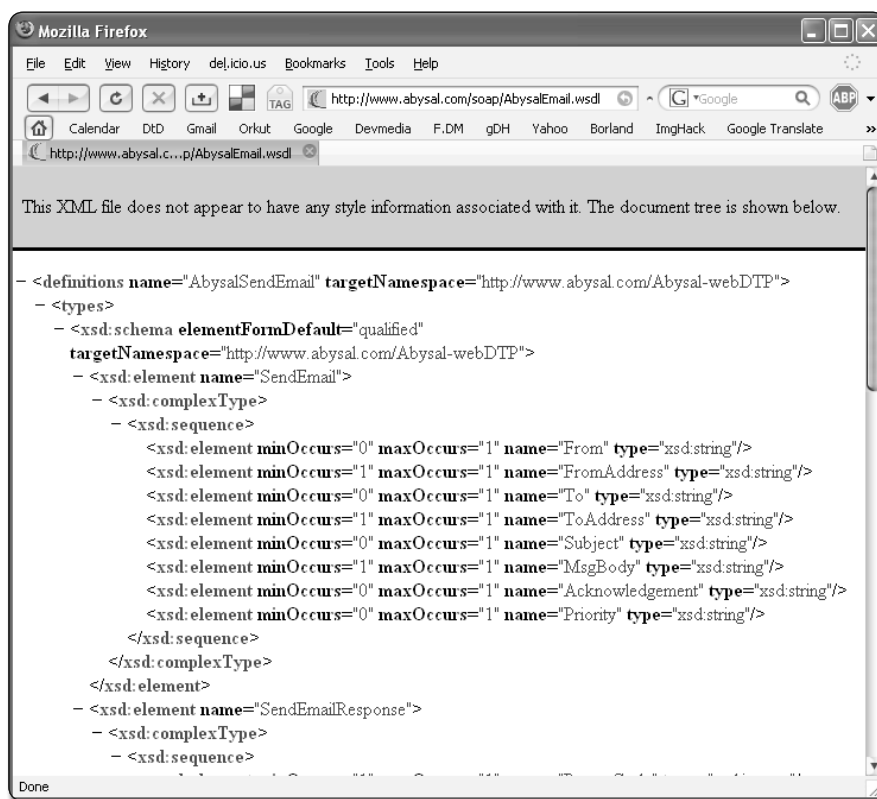


Figura 2. WSDL do Web Service

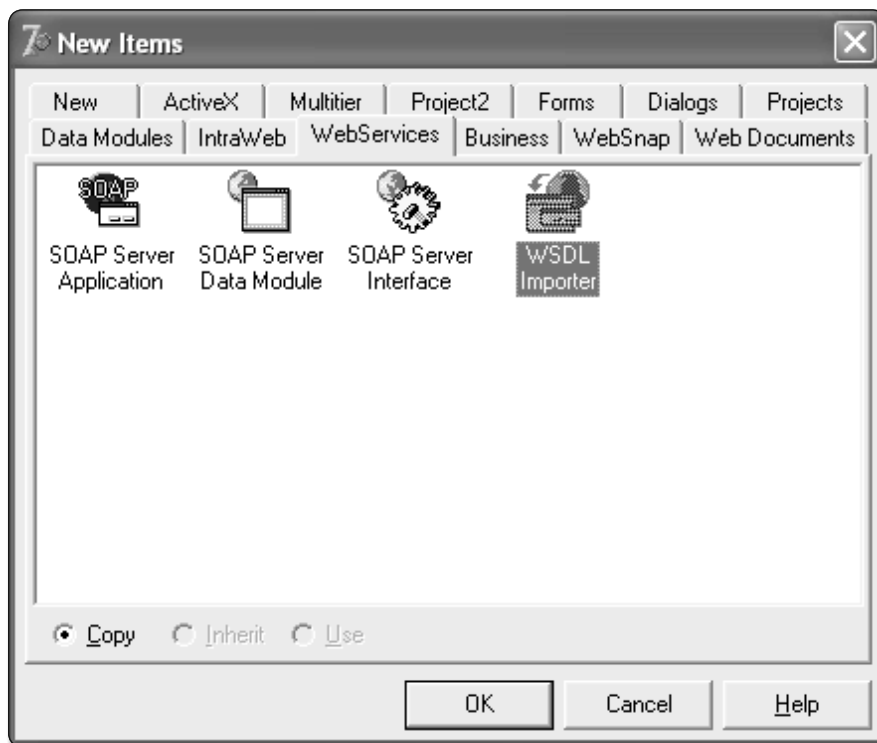


Figura 3. Opção do WSDL Importer



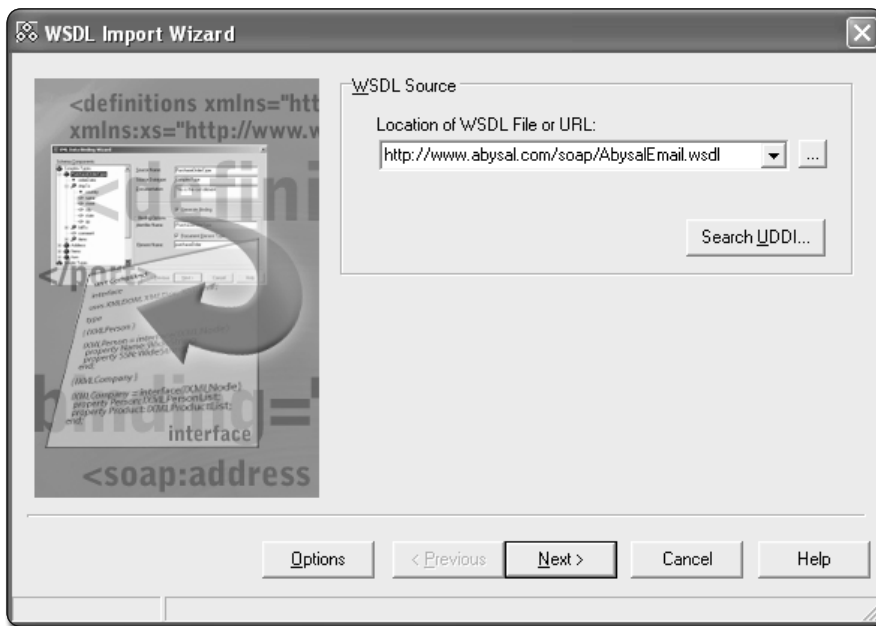


Figura 4. Assistente de importação do WSDL

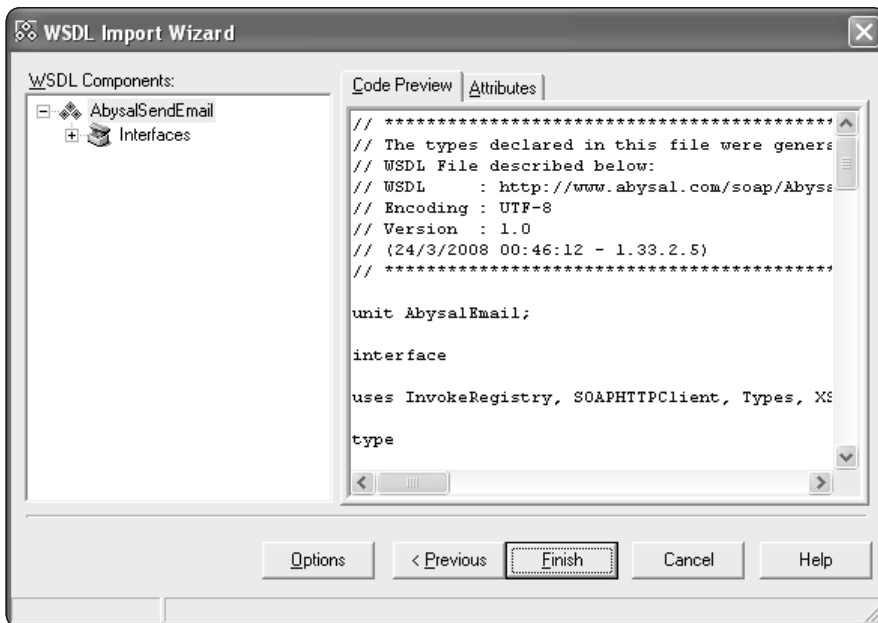


Figura 5. Unit gerada

**Listagem 1.** Envio de e-mail utilizando Web Service

```

procedure TfrmWSEmail.bbtnEnviarClick(Sender:
TObject);
begin
  try
    (HTTPRIO1 as SendEmailPortType).SendEmail
      (edtNomeRem.Text, edtEmailRem.Text,
       edtNomeDes.Text, edtEmailDes.Text,
       edtAssunto.Text, memMensagem.Lines.Text,
       '', '');
    ShowMessage('E-mail enviado com Sucesso!');
  except
    ShowMessage('Problemas na conexão, com o
      serviço de envio de E-mail');
  end;
end;

```

nente indicado e digite o link do Web Service na propriedade mencionada anteriormente. O componente pode ser encontrado na paleta de componentes *WebServices*.

Após a configuração do componente “HTTPRIO”, vamos relacionar a *Unit* do formulário com a *Unit* criada há pouco pelo Delphi (Figura 5), pressionando as teclas ALT + F11 e selecionar a *Unit* que contém os métodos. Agora podemos passar para a parte de codificação e utilização dos métodos que são disponibilizados pelo Web Service.

A codificação se dará no evento *OnClick* do botão “Enviar” onde deverá digitar o código da **Listagem 1**, que fará a chamada ao serviço da Web importado, passando os valores requisitados para a função encapsulada no serviço, enviando assim o e-mail e retornando uma mensagem ao usuário (Figura 6).

**Nota:** A palavra “SendEmailPortType”, apresentada no código da **Listagem 1** diz respeito ao nome da interface do serviço e não ao nome da *Unit* ao qual os métodos foram disponibilizados.

## Exemplo 02: Transformação de Temperaturas

Este exemplo tratará da implementação de um Web Service bem simples, cujo objeto é converter uma temperatura dada

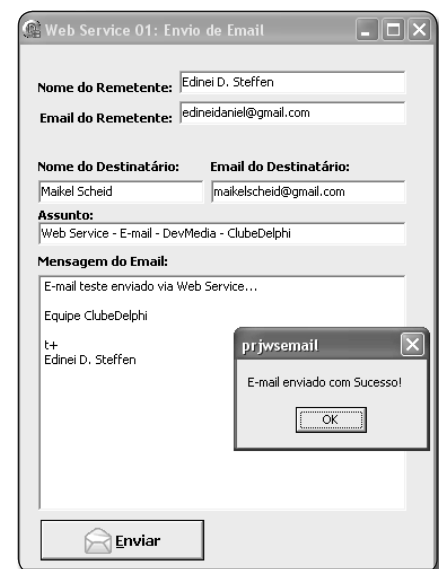


Figura 6. Resultado da implementação do Web Service no Projeto

em graus *Celsius* para uma temperatura na escala *Fahrenheit* e vice-versa.

Iniciando um novo projeto, no menu *File|New>Application* crie uma nova aplicação, altere a propriedade *Name* do formulário inicial para “frmWSConverteTemp” e a propriedade *Caption* para “Web Service 02: Conversão de Temperaturas (CtoF, FtoC)”. Salve a *unit* principal do projeto como “uWSConverte

Temp.pas” e o projeto salve como “prjWSConverteTemp.dpr”.

Para o layout do formulário, utilizaremos primeiramente um componente *RadioGroup* (“rgConfigTemp”), que será responsável pela configuração do tipo de conversão que se deseja realizar. Na propriedade *Caption* será descrito com “Tipo de Conversões”, na propriedade *Items* serão adicionadas duas

linhas. A primeira com a frase “Celsius para Fahrenheit” e na segunda linha “Fahrenheit para Celsius”. E para uma das alternativas ficar marcada será informado o número 0 (zero) na propriedade *ItemIndex*.

Ainda será necessário um *Label* com sua propriedade *Caption* alterado para “Informe a Temperatura”, logo abaixo da *Label* um componente *Edit* (“edtTemp”), para que o usuário possa informar o valor da temperatura a ser convertida e limpe a propriedade *Text*. Para mostrar o resultado da conversão será utilizado um componente *Label* (“lblResultado”) e para finalizar a parte de layout será adicionado um *Button* que será responsável por executar os comandos de interação com Web Service (**Figura 7**).

Repita o processo que realizamos no exemplo anterior para importar o WSDL de conversão, ou seja, acessando o menu *File|New>Other>Web Services>WSDL Importer* e digitando o endereço *developerdays.com/cgi-bin/tempconverter.exe/wsdl/ITempConverter*.

Feita a importação, inclua um componente “*HTTPRIO*” e informe a *Url* do serviço. Não devemos esquecer de salvar a *Unit* criada pelo delphi a partir da importação do WSDL. O nome da *Unit* pode seguir com o nome que o Delphi sugere.

No evento *OnClick* do botão “Buscar”, faremos o uso do código da **Listagem 2**, sem antes esquecer de relacionar as *Unit*’s (ALT + F11). Após o código implementado obtém-se o resultado, conforme ilustrado na **Figura 8**.

### Exemplo 03: Consulta de CEP e Logradouros

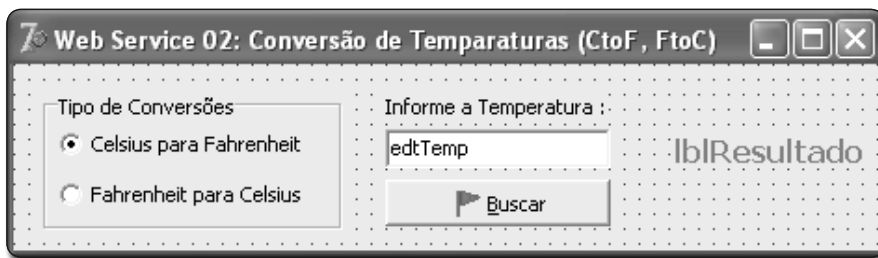
Este exemplo tratará da implementação de um Web Services, cujo objetivo é retornar uma descrição completa de um ou vários logradouros, dependendo da forma que vão ser informados os parâmetros para serviço. Este Web Service foi encontrado na Internet, portanto não sei informar, por quanto tempo os dados estarão disponíveis, mas segundo a documentação, este serviço dispõe de praticamente todas as cidades/logradouros do país. O serviço disponibiliza dois métodos para

**Listagem 2.** Código de utilização do Web Services para conversão de temperatura

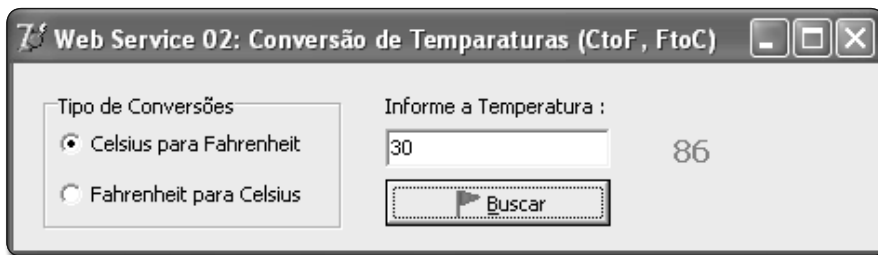
```
case rgConfigTemp.ItemIndex of
  0 : lblResultado.Caption :=
    IntToStr((HTTPRIO1 as ITempConverter ).
      CtoF(StrToInt(edtTemp.Text)));
  1 : lblResultado.Caption :=
    IntToStr((HTTPRIO1 as ITempConverter ).
      FtoC(StrToInt(edtTemp.Text)));
end;
```

**Listagem 3.** Implementação da mudança de formulário de acordo com a pesquisa

```
procedure TfrmWSCEP.rgTipoConsultaClick(Sender:
  TObject);
begin
  case rgTipoConsulta.ItemIndex of
    0:
      begin
        { Mostra o componente Panel com os campos
          na tela }
        pnlParametros.Visible := False;
        { Monta a descrição do campo principal de
          consulta }
        lblCepDescricao.Caption := 'Informe o CEP';
      end;
    1:
      begin
        { Oculta o componente Panel com os campos na tela }
        pnlParametros.Visible := True;
        { Monta a descrição do campo principal de consulta }
        lblCepDescricao.Caption := 'Informe a parte da
          descrição do Lograd.:';
      end;
  end;
end;
```



**Figura 7.** Layout do Formulário.



**Figura 8.** Resultado Final, após a implementação do Web Service

realizar a consulta, um onde somente é necessário informar o CEP retornando o Logradouro, Cidade e UF e outro onde é possível realizar uma busca aproximada pela descrição do logradouro, retornando todos os registros que se encaixam neste tipo de filtro.

Novamente iniciaremos um novo projeto no menu *File\New>Application* e altere a propriedade *Name* do formulário inicial para "frmWSCep" e a propriedade *Caption* para "Web Service 03: Consulta de CEP e Cidades". Salve a *unit* principal do projeto como "uWSCep.pas" e o projeto salve como "prjWSCep.dpr".

Vamos para a montagem do layout do formulário que irá interagir com o Web Service. Colocaremos os seguintes componentes em tela, um *RadioGroup* ("rgTipoConsulta") e na sua propriedade *Items* adicione dois itens ("Por CEP", "Por Descrição de Logradouro"). Abaixo deste componente, adicionaremos uma *Label* ("lblCepDescricao") com o *Caption* "Informe o CEP". Adicionaremos abaixo da *Label* um componente *Edit* ("edtCepDescricao"). Utilizaremos uma *Panel* ("pnlParametros"), sua propriedade *Visible* ficará como "False" e dentro desta *Panel* será adicionado mais quatro componentes, dois *Edit*'s, um sendo identificado por "edtUF" e ao seu lado uma *Label* com a descrição do campo, e o outro *Edit* ("edtNroReg"), com uma *Label* ao seu lado para identificação. Será adicionado um *BitBtn* ("bbtnBuscar") e por fim um componente *Memo* ("memResultado") levando também ao seu lado uma *Label* para descrever a finalidade do campo conforme layout apresentado na **Figura 9**.

O componente *Panel* será exibido conforme a opção de consulta selecionada. No evento *OnClick* do "rgTipoConsulta" implemente o código da **Listagem 3** que se encontra comentado de acordo com a ação a ser codificada.

Em seguida podemos realizar a importação do WSDL do serviço para nossa aplicação, seguindo os mesmos passos dos dois exemplos anteriores. Dessa vez basta usar a url [desenvolvimento.setrem.com.br/ws-](http://desenvolvimento.setrem.com.br/ws-CEP/wsCEP.exe/wsdl/ICEP)

*CEP/wsCEP.exe/wsdl/ICEP*. No menu *File\New>Other>Web Services>WSDL Importer* informe o link mencionado referente ao serviço, seguindo os passos até finalizar o assistente. Após a importação do serviço salve a *Unit* na mesma pasta do projeto.

Agora vamos adicionar um componente *HTTPRIO* para realizar a interação com os métodos do serviço, e em sua propriedade URL inclua o link indicado.

**Nota:** De onde vem o link para preencher a propriedade *Url* do componente *HTTPRIO*? Pois bem, a *Unit* gerada a partir do *WSDL Importer* existe uma única função implementada que é responsável por realizar a comunicação

com o Web Services. E é nessa função que existe uma variável identificando a *Url* que o componente necessita, nesse caso: [desenvolvimento.setrem.com.br/weCEP/wecCEP.exe/soap/ICEP](http://desenvolvimento.setrem.com.br/weCEP/wecCEP.exe/soap/ICEP).

Após feita esta configuração o Web Service está pronto para ser consumido, portanto, precisamos relacionar as *Unit*'s (ALT + F11) e no evento *OnClick* do botão "Buscar" adicionaremos o código da **Listagem 4**.

Como podemos ver, o código segue os mesmos padrões de codificação dos exemplos anteriores. Passamos para o WS os parâmetros necessários e ele nos retorna o CEP encontrado. O resultado é demonstrado através das **Figuras 10 e 11**.

# PENSE...

QUANTO TEMPO  
VOCÊ GASTARIA  
PARA DESENVOLVER  
COBRANÇA COM BOLETOS  
BANCÁRIOS PARA  
APENAS UM BANCO  
NO SEU SOFTWARE

# COBREBEMX

56 BANCOS E MAIS DE 430 CARTEIRAS DE COBRANÇA PARA IMPRESSÃO E/OU ENVIO DE BOLETO BANCÁRIO POR EMAIL;

GERAÇÃO DE BOLETOS ON LINE;

GERAÇÃO E LEITURA DE ARQUIVOS (REMESSA/RETORNO) NOS PADRÕES FEBRABAN E CNAB;

MAIS DE 40 EXEMPLOS EM DIVERSAS LINGUAGENS DE PROGRAMAÇÃO

DOWNLOADS E INFORMAÇÕES EM [WWW.COBBREBEM.COM](http://WWW.COBBREBEM.COM)

**cobrem**  
Tecnologia

**Listagem 4.** Código de requisição do CEP

```
case rgTipoConsulta.ItemIndex of
0: memResultado.Lines.Text :=
(HTTPRIO1 as ICEP).LogradourosPorCep(
edtCepDescricao.Text);
1: memResultado.Lines.Text := (HTTPRIO1 as ICEP).
LogradourosPorNome(edtCepDescricao.Text,
edtUF.Text, edtNroReg.Text);
end;
```

**Figura 9.** Layout do Formulário, após o desenho do mesmo

**Figura 10.** Demonstração da Consulta por CEP

**Figura 11.** Demonstração da Consulta por Descrição do Logradouro

Da mesma forma é demonstrado o resultado, se utilizada a opção “Por Descrição de Logradouro”, conforme ilustrado na **Figura 11**.

Com estes exemplos podemos ter uma boa noção dos recursos que a utilização de Web Services nos oferece. Inúmeras são as possibilidades de implementações em termos de desenvolvimento de software.

## Conclusão

Foram abordados neste artigo os principais conceitos sobre a Web Services, juntamente com exemplos práticos sobre a utilização destes serviços, demonstrando a facilidade de se trabalhar com esse tipo de tecnologia.

Utilizamos padrões abertos como HTTP, SOAP, UDDI, ao invés de tecnologias proprietárias, e tudo isso ainda baseado em um padrão universal de comunicação entre sistemas, o XML.

Agora cabe a você, analisar o emprego desta tecnologia em seu cotidiano, e aplicar, com base nas funcionalidades apresentadas neste artigo, mais recursos em suas aplicações. Explore mais esta área, busque mais, leia mais, aprenda mais, pois o emaranhado de opções que existem nesse tipo de serviço é muito grande. Em uma próxima oportunidade veremos aqui na Easy Delphi como implementar aplicações servidoras.

Um grande abraço e até o próximo artigo. ●

### Dê seu feedback sobre esta edição!

A Clubedelphi tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

[www.devmedia.com.br/clubedelphi/feedback](http://www.devmedia.com.br/clubedelphi/feedback)

