

Protótipo de software para troca de dados entre aplicações de comércio eletrônico utilizando o protocolo SOAP

Cristiano Fornari Colpani

Orientando

Alexander Roberto Valdameri

Orientador



Roteiro

- Introdução e Objetivos
- Fundamentação Teórica
- Especificação
- Implementação
- Apresentação do Protótipo
- Conclusão

Introdução

- Internet
- Aplicações para Comércio Eletrônico
- B2B
- Estrutura das aplicações B2B
- Implementações
- Necessidade de simplificação e reaproveitamento

Objetivos

- Criar uma aplicação para troca de catálogos entre aplicações de comércio eletrônico
- Criar outras aplicações de comércio eletrônico para exemplificar o uso da aplicação principal

XML

- Extensible markup language
- Linguagem para descrever documentos estruturados (árvore)
- Elementos
- Atributos
- Namespaces
- Schemas

Documento XML

```
<pre:Pessoa xmlns:pre='http://colpani.com/ex01'>  
  <pre:Nome>Luiz Felipe</pre:Nome>  
  <pre:Idade>33</pre:Idade>  
  <pre:Endereco rua="Almirante" Nro="222">Blumenau</pre:Endereco>  
</pre:Pessoa>
```

Protocolo SOAP

- Troca de Informações
- Processamento Distribuído para aplicações descentralizadas (Internet)
- Ambiente Heterogêneos
- Remote Procedure Call (RPC)

Envelope SOAP

- **HEADER**
- **BODY**



```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope">
  <env:Header>
    <m:reservation xmlns:m="http://travelcompany.example.org/reservation"
      env:actor="http://www.w3.org/2001/12/soap-envelope/actor/next"
      env:mustUnderstand="true">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</reference>
      <m:dateAndTime>2001-11-29T13:20:00.000-05:00</m:dateAndTime>
    </m:reservation>
    <n:passenger xmlns:n="http://mycompany.example.com/employees"
      env:actor="http://www.w3.org/2001/12/soap-envelope/actor/next"
      env:mustUnderstand="true">
      <n:name>John Q. Public</n:name>
    </n:passenger>
  </env:Header>
  <env:Body>
    <p:itinerary xmlns:p="http://travelcompany.example.org/reservation/travel">
      <p:departure>
        <p:departing>New York</p:departing>
        <p:arriving>Los Angeles</p:arriving>
        <p:departureDate>2001-12-14</p:departureDate>
        <p:departureTime>late afternoon</p:departureTime>
        <p:seatPreference>aisle</p:seatPreference>
      </p:departure>
      <p:return>
        <p:departing>Los Angeles</p:departing>
        <p:arriving>New York</p:arriving>
        <p:departureDate>2001-12-20</p:departureDate>
        <p:departureTime>mid morning</p:departureTime>
        <p:seatPreference/>
      </p:return>
    </p:itinerary>
    <q:lodging xmlns:q="http://travelcompany.example.org/reservation/hotels">
      <q:preference>none</q:preference>
    </q:lodging>
  </env:Body>
</env:Envelope>
```

Serialização

- Tipos Simples
- Enumerações
- Matrizes e Estruturas

```
TAluno = Record
  Nome      : String;
  Codigo    : Integer;
  Endereço  : String;
  Telefone  : String;
end;
<soap:Body>
  <m:VerificaMatricula xmlns:m="http://tempuri.org">
    <Aluno>
      <Nome>Cristiano Fornari Colpani</Nome>
      <Codigo>18261</Codigo>
      <Endereco>Avenida Tom Jobim, Copacabana</Endereco>
      <Telefone>47 30373209</Telefone>
    </Aluno>
  </m:VerificaMatricula>
</soap:Body>
```

Web Services

- SOAP sobre HTTP
- WSDL
- Utilização de Web Services

Casos de uso

- Parametrizar Sistema
- Solicitar e Receber catálogos
- Enviar catálogos

Especificação

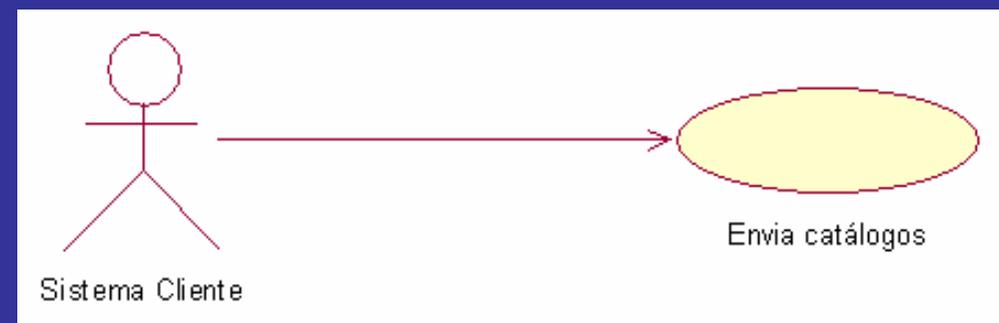
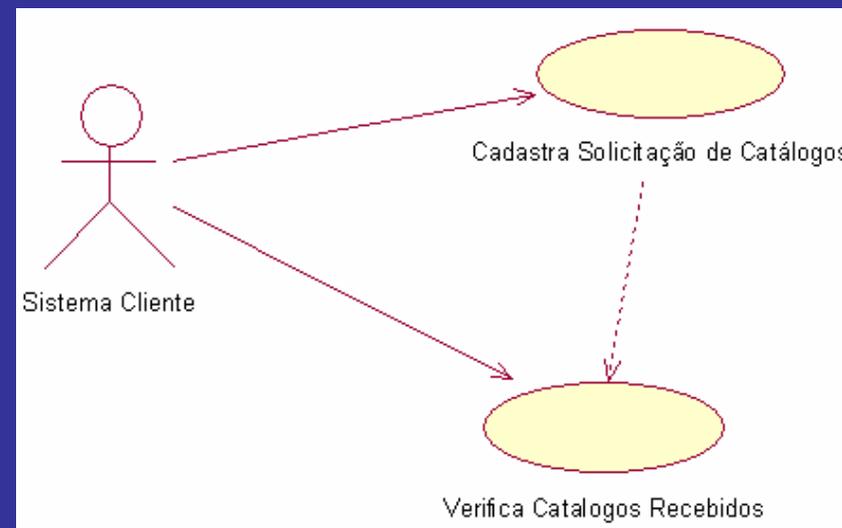
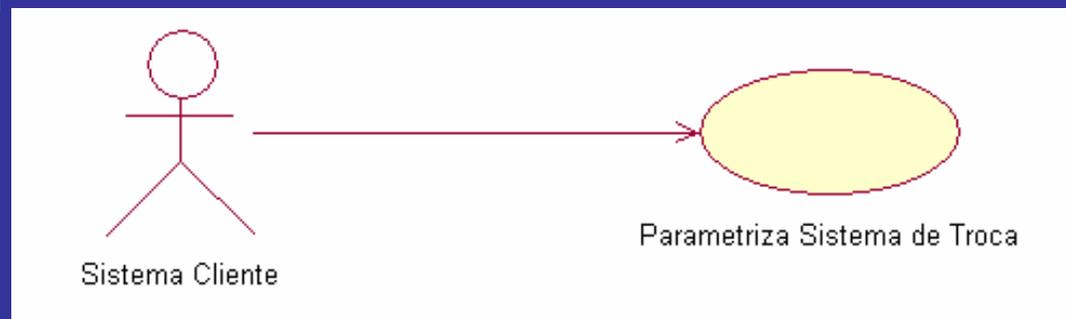
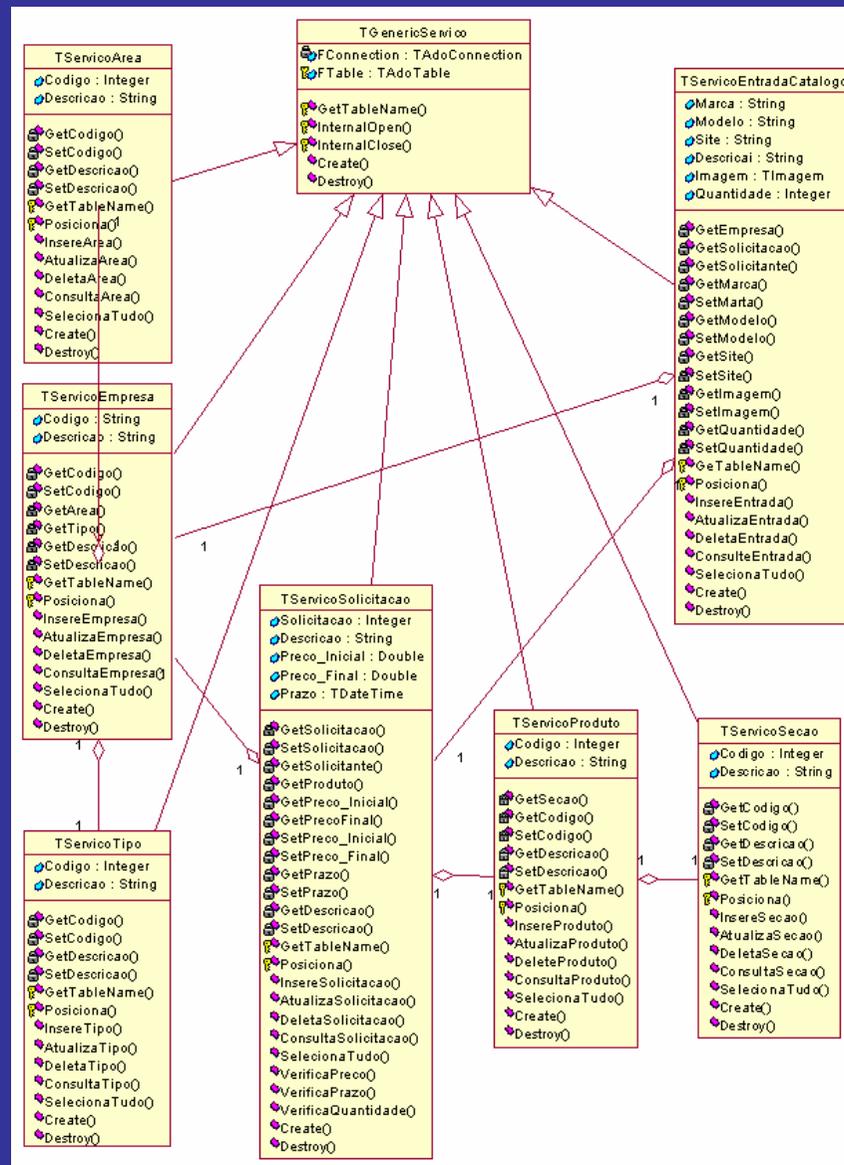


Diagrama de Classes

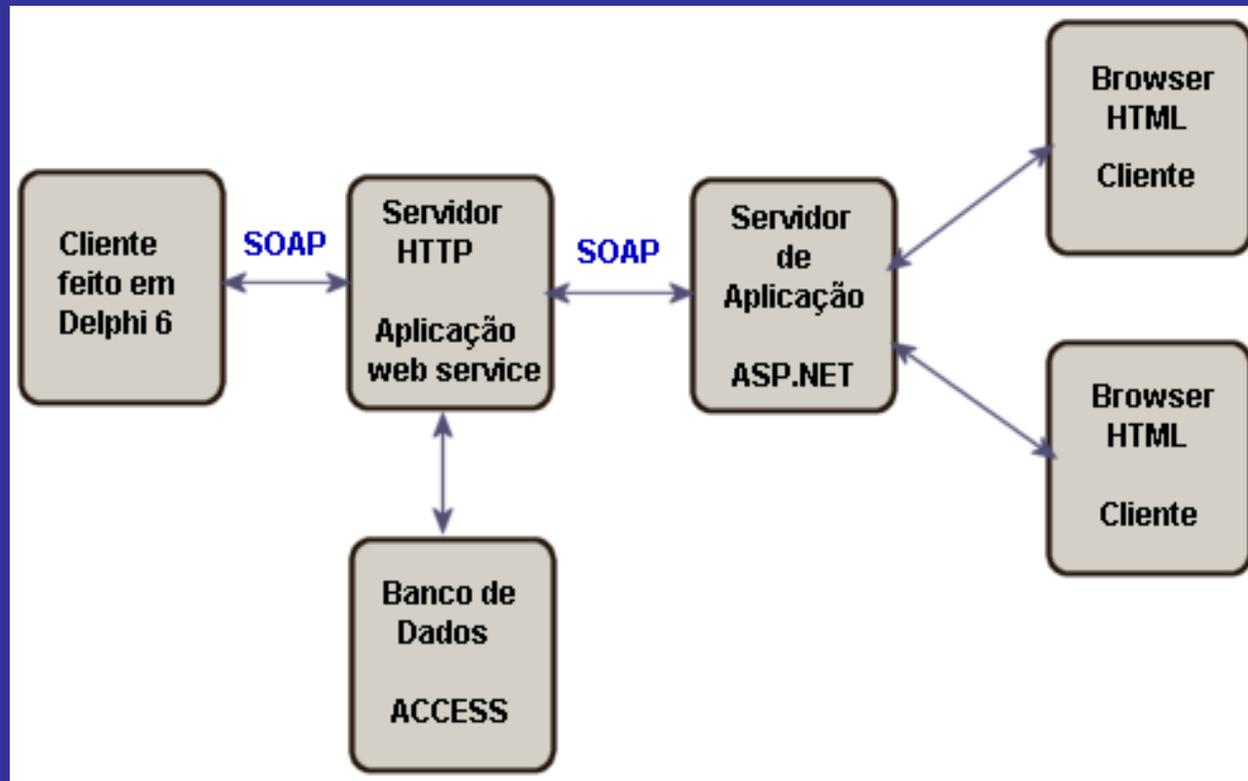
Especificação



Técnicas e Ferramentas

- Internet Information Services
- Delphi6 Enterprise Edition
- Visual Studio .NET
- Microsoft Access

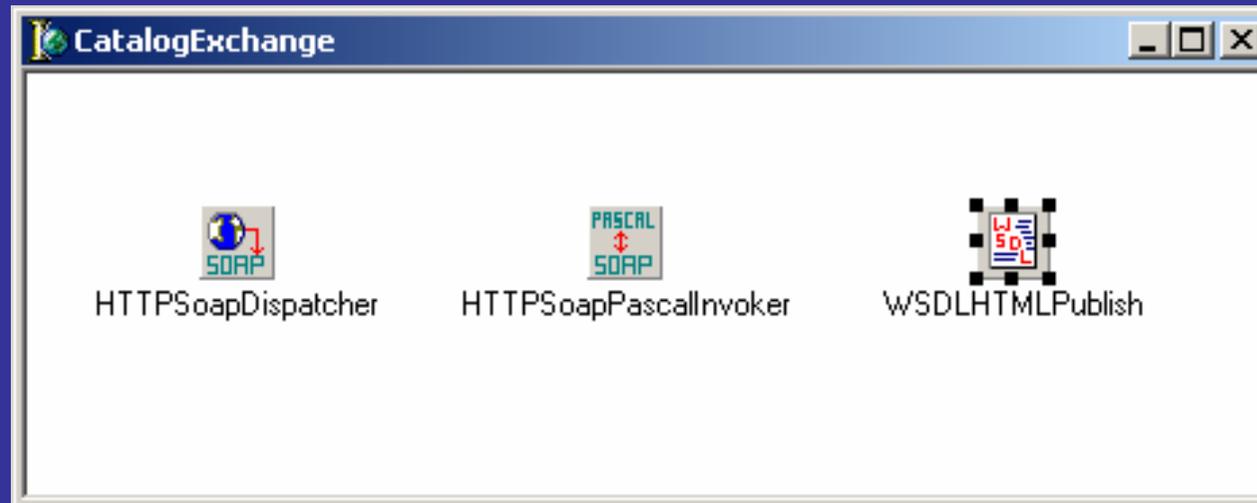
Aplicação Principal



Implementação

```
TServicoArea = class(TGenericService, IServicoArea)
private
    function GetCodigo: Integer;
    procedure SetCodigo(Value: Integer);
    function GetDescricao: String;
    procedure SetDescricao(Value: String);
protected
    function GetTableName: String; override;
    function Posiciona(_Codigo: Integer): Boolean;
public
    procedure InsereArea(_Codigo: Integer; const _Descricao: String); stdcall;
    procedure AtualizaArea(_Codigo: Integer; const _Descricao: String); stdcall;
    procedure DeletaArea(_Codigo: Integer); stdcall;
    function ConsultaArea(_Codigo: Integer; var Descricao_: WideString): Boolean; stdcall;
    function ContaRegistros: Integer; stdcall;
    function SelecionaRegistro(BookMark: Integer;
        var Codigo_: Integer;
        var Descricao_: WideString
    ): Boolean; stdcall;
    property Codigo: Integer read GetCodigo write SetCodigo;
    property Descricao: String read GetDescricao write SetDescricao;
end;
```

```
IServicoArea = interface(IInvokable)
[ '{CC28B2A3-6374-4FE5-BA0D-D8F673720B24}' ]
    procedure InsereArea(_Codigo: Integer; const _Descricao: String); stdcall;
    procedure AtualizaArea(_Codigo: Integer; const _Descricao: String); stdcall;
    procedure DeletaArea(_Codigo: Integer); stdcall;
    function ConsultaArea(_Codigo: Integer; var Descricao_: WideString): Boolean; stdcall;
    function ContaRegistros: Integer; stdcall;
    function SelecionaRegistro(BookMark: Integer;
        var Codigo_: Integer;
        var Descricao_: WideString
    ): Boolean; stdcall;
end;
```



```
uses
```

```
    InvokeRegistry;
```

```
initialization
```

```
    InvRegistry.RegisterInvokableClass(UServiceArea.TServicoArea);
```

```
    InvRegistry.RegisterInvokableClass(UServiceTipo.TServicoTipo);
```

```
    InvRegistry.RegisterInvokableClass(UServiceEmpresa.TServicoEmpresa);
```

```
    InvRegistry.RegisterInvokableClass(UServiceProduto.TServicoProduto);
```

```
    InvRegistry.RegisterInvokableClass(UServiceSecao.TServicoSecao);
```

```
    InvRegistry.RegisterInvokableClass(UServiceSolicitacao.TServicoSolicitacao);
```

```
    InvRegistry.RegisterInvokableClass(UServiceEntradaCatalogo.TServicoEntradaCatalogo);
```

```
    InvRegistry.RegisterInterface(TypeInfo(UServiceArea.IServicoArea));
```

```
    InvRegistry.RegisterInterface(TypeInfo(UServiceTipo.IServicoTipo));
```

```
    InvRegistry.RegisterInterface(TypeInfo(UServiceEmpresa.IServicoEmpresa));
```

```
    InvRegistry.RegisterInterface(TypeInfo(UServiceProduto.IServicoProduto));
```

```
    InvRegistry.RegisterInterface(TypeInfo(UServiceSecao.IServicoSecao));
```

```
    InvRegistry.RegisterInterface(TypeInfo(UServiceSolicitacao.IServicoSolicitacao));
```

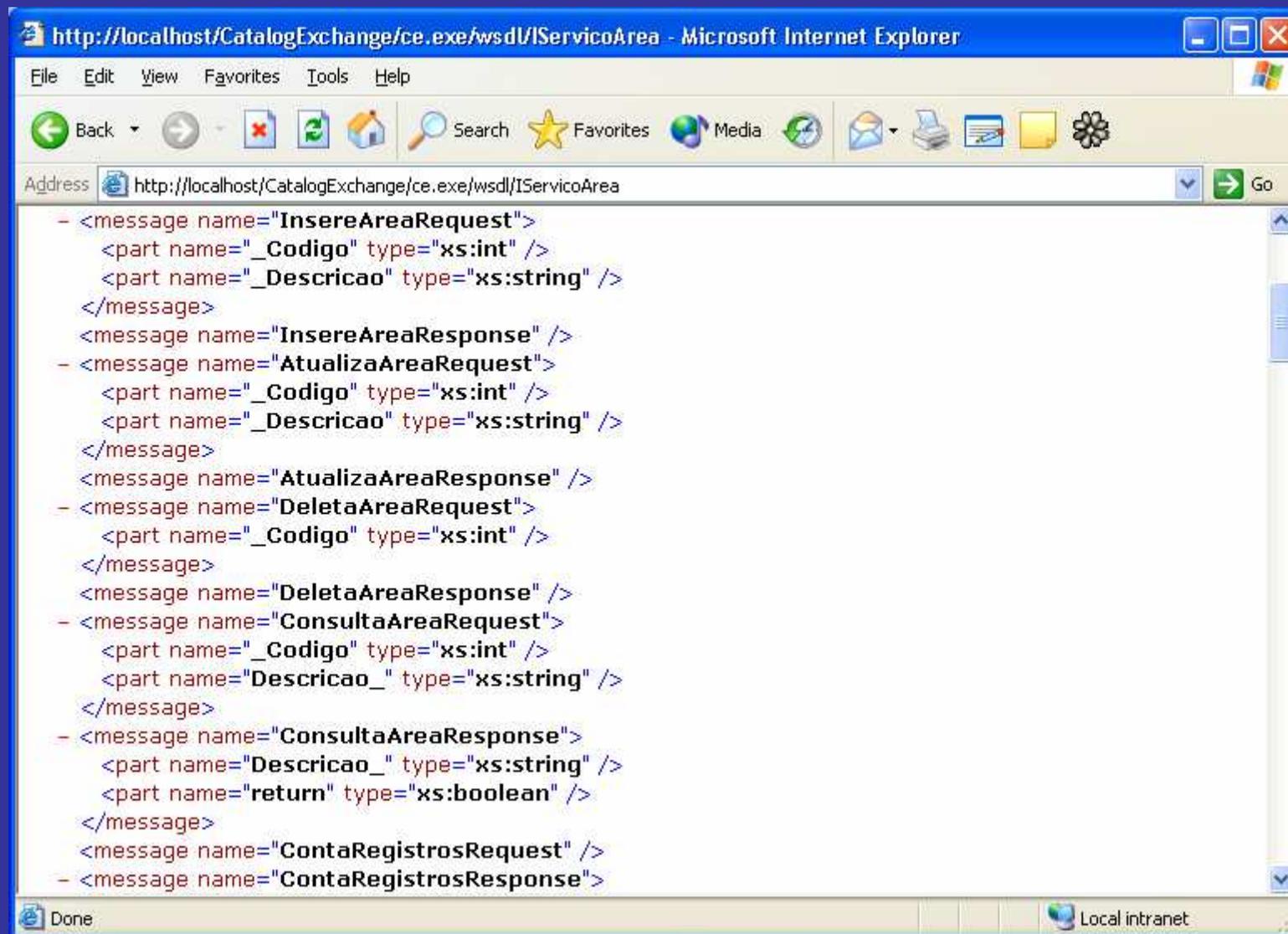
```
    InvRegistry.RegisterInterface(TypeInfo(UServiceEntradaCatalogo.IServicoEntradaCatalogo));
```

The screenshot shows a Microsoft Internet Explorer window displaying a WebService Listing page. The browser's address bar contains the URL `http://localhost/catalogexchange/ce.exe/wsd`. The page title is "ce - WebService Listing". The main content is a table with the following data:

Port Type	Namespace URI	Documentation	WSDL
IServicoArea	urn:UServiceArea-IServicoArea		IServicoArea
IServicoTipo	urn:UServiceTipo-IServicoTipo		IServicoTipo
IServicoEmpresa	urn:UServiceEmpresa-IServicoEmpresa		IServicoEmpresa
IServicoProduto	urn:UServiceProduto-IServicoProduto		IServicoProduto
IServicoSecao	urn:UServiceSecao-IServicoSecao		IServicoSecao
IServicoSolicitacao	urn:UServiceSolicitacao-IServicoSolicitacao		IServicoSolicitacao
IServicoEntradaCatalogo	urn:UServiceEntradaCatalogo-IServicoEntradaCatalogo		IServicoEntradaCatalogo
IWSDLPublish	http://www.borland.com/namespaces/Types	Lists all the PortTypes published by this Service	IWSDLPublish

The browser's status bar at the bottom indicates "Local intranet".

Implementação



The screenshot shows a Microsoft Internet Explorer browser window displaying a WSDL service definition. The address bar shows the URL: `http://localhost/CatalogExchange/ce.exe/wsdl/IServicoArea`. The main content area displays the following XML code:

```
- <message name="InsereAreaRequest">
  <part name="_Codigo" type="xs:int" />
  <part name="_Descricao" type="xs:string" />
</message>
<message name="InsereAreaResponse" />
- <message name="AtualizaAreaRequest">
  <part name="_Codigo" type="xs:int" />
  <part name="_Descricao" type="xs:string" />
</message>
<message name="AtualizaAreaResponse" />
- <message name="DeletaAreaRequest">
  <part name="_Codigo" type="xs:int" />
</message>
<message name="DeletaAreaResponse" />
- <message name="ConsultaAreaRequest">
  <part name="_Codigo" type="xs:int" />
  <part name="Descricao_" type="xs:string" />
</message>
- <message name="ConsultaAreaResponse">
  <part name="Descricao_" type="xs:string" />
  <part name="return" type="xs:boolean" />
</message>
<message name="ContaRegistrosRequest" />
- <message name="ContaRegistrosResponse">
```

The status bar at the bottom shows "Done" and "Local intranet".

Casos de Uso em C#

```
int Secao = Convert.ToInt32(ddArea.SelectedItem.Value);
int Codigo = Convert.ToInt32(tbCodigo.Text);
try
{
    IProduto.DeletaProduto(Secao, Codigo);
}
catch(Exception ecp)
{
    lbError.Visible = true;
    lbError.Text = ecp.Message;
}
```

Caso de Uso em Delphi

```
IArea := GetIServicoArea;
try
  DisableEmpresa;
  Mark := IArea.ContaRegistros;
  for i := 1 to Mark do
  begin
    if IArea.SelecionaRegistro(i, Codigo_, Descricao_) then
    begin
      Rec := TRegistroServicoArea.Create;
      try
        Rec.Codigo := Codigo_;
        Rec.Descricao := Descricao_;
        FAuxArea.AddItem(i, cbArea.Items.Count, Rec);
        cbArea.Items.Add(Descricao_);
      except
        Rec.Free;
        raise;
      end;
    end;
  end;
finally
  IArea := nil;
end;
```

Operacionalidade do Protótipo

- Simulação entre troca de catálogos
- Empresas de Comércio eletrônico
- Solicitação e envio de catálogos de ar-condicionado, lavadora de roupas e produtos automotivos

Conclusões

- O Objetivo do trabalho foi alcançado na implementação do protótipo
- As ferramentas utilizadas mostraram-se capazes de suportar o protocolo SOAP
- O protótipo pode ser acessado através de diferentes plataformas

- Visualização da aplicação na fase de especificação
- Conhecimento sobre o protocolo SOAP e suas implicações
- Utilização da linguagem C# e do ambiente Visual Studio .NET

Extensões

- Comparação entre SOAP e CORBA
- SOAP sobre outros mecanismos de transporte