

DESENVOLVENDO UMA APLICAÇÃO PARA WEB NA IDE DELPHI 2006

Ana Claudia Câmara PEREIRA*

RESUMO

Neste trabalho será explicado como construir uma aplicação voltada para Web envolvendo cadastro com conexão ao banco de dados, bem como tela de consulta. Será utilizada a tecnologia do momento: ASP.Net e Delphi for .Net. Serão mostrados todos os passos necessários para desenvolver um Cadastro de Cliente com Banco de Dados SQL Server 2005 for SQL Server, incluindo também: consulta pelo CPF do Cliente, definição de máscara, validação de campo requerido e de e-mail e controle de usuário entre os formulários desenvolvidos. Para o desenvolvimento dessa aplicação será necessário ter instalado em sua máquina o SQL Server 2000 e o Delphi 2006. Essa matéria necessita como pré-requisitos, pessoas que já têm noções de programação.

PALAVRAS-CHAVE: Web. Banco de Dados. Tecnologia ASP. Net.

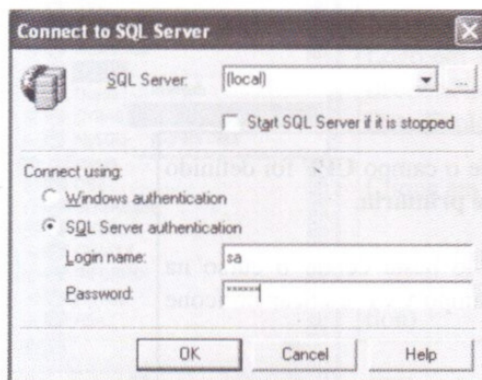
INTRODUÇÃO

Inicialmente será definido o banco de dados da aplicação, onde será mostrado passo a passo como criar um banco, construir uma tabela e definir uma chave primária. Logo após, será desenvolvido a aplicação contendo os formulários de cadastro, de consulta e de controle do usuário.

1. Criando Banco de Dados

Crie uma pasta no C:\ chamada Loja

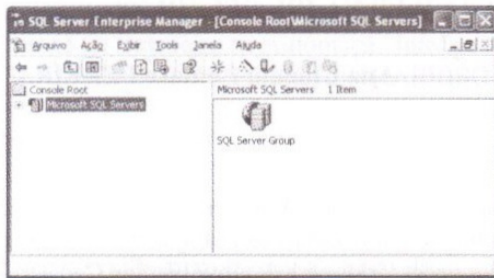
Botão Iniciar ↘ Programas ↘ Microsoft SQL Server ↘ Query Analyser



Selecione
- SQL Server authentication
Login name: sa (letras minúsculas)
Password: master (letras minúsculas)
 Clique em **OK**
 Feche essa tela.

Clique em Botão Iniciar ↘ Programas ↘ Microsoft SQL Server ↘ Enterprise Manager

* Faculdade de Tecnologia de Taquaritinga - FATEC/TQ - anacla.cp@uol.com.br



Dê duplo clique em: **Microsoft SQL Servers**, depois em **SQL Server Group** e depois no nome da máquina, que nesse exemplo é **Ana**. Clique com botão direito do mouse em **Databases** e escolha **New Database**

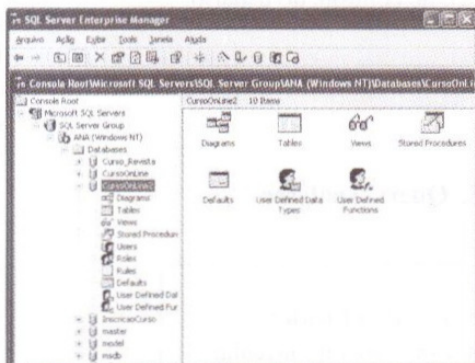
Na aba **General**, opção **Name**, coloque o nome do banco de dados - **Loja**

Na aba **Data Files**, clique em da opção **Location** e escolha o caminho em que será salvo o banco de dados. (neste caso, dentro da pasta que foi criada no **C:** chamada **Loja**)

Na aba **Transaction Log**, clique em da opção **Location** e escolha o mesmo caminho que foi escolhido acima.

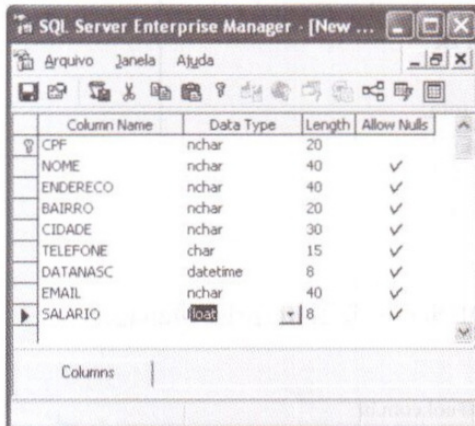
Clique no botão **OK**. Depois em **OK** novamente.

Clique no botão **+** da opção **Databases** (que fica do lado esquerdo da tela) e depois no botão **+** do nome do banco, que nesse exemplo é **Loja**.



Clique com o botão direito do mouse em **Tables** e escolha a opção **New Table**.

Preencha a tela conforme abaixo



Observe que o campo **CPF** foi definido como **chave primária**.

Para defini-lo basta deixar o curso na frente do campo **CPF** e clicar no ícone

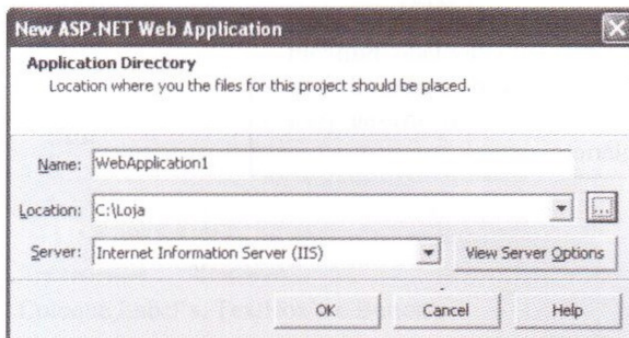
Clique no ícone para salvar a tabela. Digite **CLIENTE** e **OK**. Feche a janela.

2. Abrindo o ambiente Borland Developer Studio 2006

Iniciar ↪ Programas ↪ Borland Developer Studio 2006 ↪ Delphi for the Microsoft .NET Framework

3. Criando uma aplicação

Clique no menu **File** ↪ **New** ↪ **ASP .NET Web Application – Delphi for .NET**
Escolha o caminho a ser gravado a aplicação e clique em **OK**



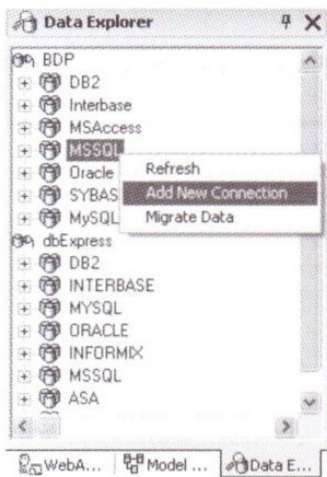
Observe que em **Server** está sendo usado o **IIS**. Caso você não tenha essa opção, será necessário instalar através do CD do Windows.

Clique em **OK**

Será inserido um formulário – **WebForm1.aspx**. Vá até a janela do **Object Inspector** e escolha **DOCUMENT** e na propriedade **PageLayout**, escolha **GridLayout**.

4. Conectando a aplicação com o banco de dados

Observe a tela do **Project Manager**, que fica do lado direito e selecione a aba **Data Explorer**.



Clique com o botão direito do mouse sobre **MSSQL** do **BDP** e escolha a opção **Add New Connection**

Logo em seguida vamos dar um nome para nossa conexão (**ConnectionStringName**) que nada mais é do que um aliás para aparecer no Data Explorer. Neste exemplo será **AliasLoja**.

Clique em **OK**

A próxima etapa será modificar a conexão que acabamos de criar para que acesse o banco de dados que desejamos. (**Modify Connection**)

Clique no **+** que está antes do **MSSQL** do **BDP**.
Clique com botão direito sob o **AliasLoja** e escolha **Modify Connection**

Database: nome do banco, neste exemplo é **Loja**

HostName: localhost


UserName: as

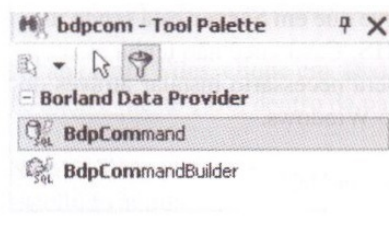
UserPassword: master

Será aberta a janela **Connection Editor**. Preencha: Após às alterações, clique no botão **Test**. Se estiver tudo OK, será apresentada a tela de conexão com sucesso.

Clique no nome do alias que é **Loja** e arraste para o formulário. Será colocado o componente **BdpConnection**.

Insira o componente **BDPCommand** – que fica na aba Borland Data Provider.


Observe que você poderá usar o filtro clicando no ícone . Digite o nome do componente que a procura será feita automaticamente.

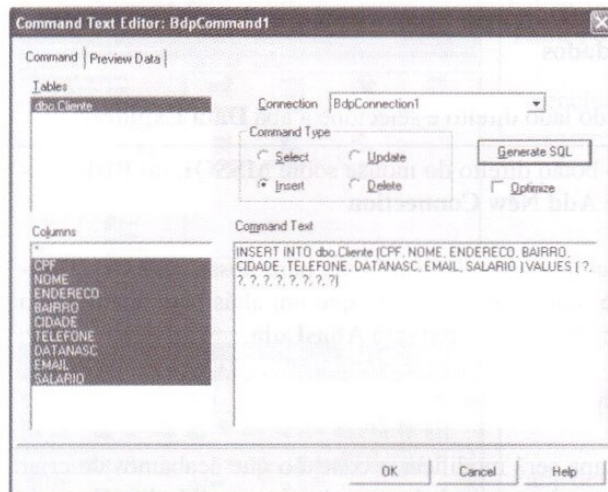


No exemplo, foi digitado **bdbcom** e o componente **BdpCommand** foi encontrado. Agora é só arrastar para o formulário.

Mude as propriedades do **BdpCommand1**:

Connection – escolha **BDPConnection1**

CommandText – Clique em , escolha a tabela, que nesse exemplo é **dbo.Cliente**.

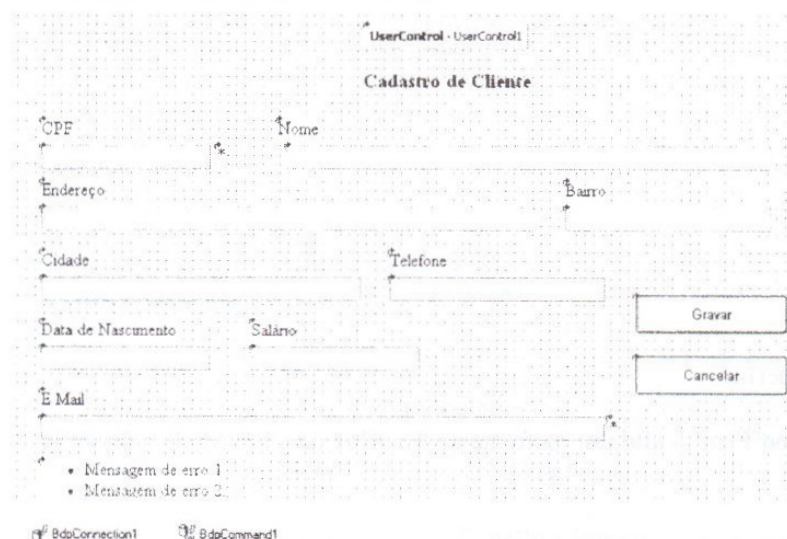


Em **Columns**, selecione todos os campos.

Na opção **Command Type** selecione **Insert**.

Clique no botão **Generate SQL** e depois em **OK**

5. Montando a tela de Cadastro de Cliente



Coloque Label's, TextBox's e Button's

Mude os nomes (propriedade **ID**) dos TextBox, respectivamente:

txtcpf, txtnome, txtendereco, txtbairro, txtcidade, txttelefone, txtdatanasc, txtsalario e txtemail

Duplo clique no botão Gravar

```

procedure TWebForm1.btngravar_Click(sender: System.Object; e: System.EventArgs);
begin
    try
        BdpConnection1.open;
        BdpCommand1.Parameters[0].Value := txtcpf.Text;
        BdpCommand1.Parameters[1].Value := txtnome.Text;
        BdpCommand1.Parameters[2].Value := txtendereco.Text;
        BdpCommand1.Parameters[3].Value := txtbairro.Text;
        BdpCommand1.Parameters[4].Value := txtcidade.Text;
        BdpCommand1.Parameters[5].Value := txttelefone.Text;
        BdpCommand1.Parameters[6].Value := txtdatanasc.Text;
        BdpCommand1.Parameters[7].Value := txtemail.Text;
        BdpCommand1.Parameters[8].Value := txtsalario.Text;
        BdpCommand1.ExecuteNonQuery;
        txtcpf.Text := '';
        txtnome.Text := '';
        txtendereco.Text := '';
        txtbairro.Text := '';
        txtcidade.Text := '';
        txttelefone.Text := '';
        txtdatanasc.Text := '';
        txtemail.Text := '';
        txtsalario.Text := '';
    finally
        BdpConnection1.Close;
    end;
end;

```

Duplo clique no botão Cancelar

```

procedure TWebForm1.btncancelar_Click(sender: System.Object; e: System.EventArgs)
begin
    txtcpf.Text:='';
    txtnome.Text:='';
    txtenderaco.text:='';
    txtbairro.Text:='';
    txtcidade.Text:='';
    txttelefone.Text:='';
    txtdata nasc.Text:='';
    txtemail.Text:='';
    txtsalario.Text:='';
end;
    
```

6. Validando Campo Requerido

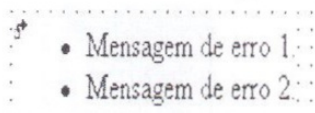
Insira o componente **Required Field Validator** na frente no TextBox do CPF. Mude a propriedade **Text** para *



Na propriedade **Error Message** digite a mensagem que será mostrada ao usuário caso o campo não for preenchido. Por exemplo: CPF – Preenchimento Requerido.

Na propriedade **ControlToValidate** escolha o componente que está relacionado. Neste exemplo **TXTCPF**.

Insira o componente **Validation Summary** no rodapé do formulário para que as mensagens possam aparecer.



7. Validando E-Mail

Insira o componente **RegularExpressionValidator** na frente no TextBox do E-Mail. Mude a propriedade **Text** para *

Na propriedade **Error Message** digite a mensagem que será mostrada ao usuário caso o e-mail digitado estiver no formato incorreto. Por exemplo: E-mail está incorreto

Na propriedade **ControlToValidate** escolha o componente que está relacionado. Neste exemplo **TXTEMail**

Na propriedade **ValidationExpression**, clique em ... e escolha **Endereço de EMail Francês. OK.**

8. Colocando Máscara nos campos Data e Salário

Escolha o form **TWebForm1** na tela **Object Inspector** e no Evento **Load** digite:

```
procedure TWebForm1.Page_Load(sender: System.Object; e: System.EventArgs);
begin
  if not IsPostBack then
    begin
      txtsalario.Attributes.Add('onKeyPress', 'javascript:FormataValor(txtsalario,13)');
      txtdatanasc.Attributes.add('onKeyPress', 'javascript:Mascara("DDMMYYYY", txtdatanasc)');
    end;
end;
```

Aperte a tecla <F12> para voltar ao formulário

Na parte inferior do formulário, clique na aba **WebForm1.aspx**

Insira as funções **function Mascara** e **Function FormataValor** que está entre as tags <script language="JavaScript"> e </script>

Obs. Digitar igualzinho está no modelo, ou seja, onde estiver letra maiúscula / letra minúscula coloque também.

```
<html>
<head>
  <title></title>
  <script language="JavaScript">
    function Mascara(formato,objeto) {
      campo = eval(objeto);
      if (formato=='DDMMYYYY') {
        separador = '/';
        conjunto1 = 2;
        conjunto2 = 5;
        if (window.event.keyCode >= 48 && window.event.keyCode <= 57) {
          if (campo.value.length == conjunto1) {
            campo.value = campo.value + separador; }
          if (campo.value.length == conjunto2) {
            campo.value = campo.value + separador; } }
        else {
          window.event.keyCode = 0; } } }

    function FormataValor(objeto, tammax) {
      campo = eval(objeto);
      vr = campo.value;
      vr = vr.replace( "/", "" );
      vr = vr.replace( "\", "" );
      vr = vr.replace( ",", "" );
      vr = vr.replace( ".", "" );
      vr = vr.replace( " ", "" );
      vr = vr.replace( " ", "" );
      vr = vr.replace( " ", "" );
      tam = vr.length;
      if (tam < tammax && window.event.keyCode != 8) {
        tam = vr.length + 1; }
```

```

if (window.event.keyCode == 8 ) {
    tam = tam - 1; }
if ( window.event.keyCode == 8 ||
window.event.keyCode >= 48 && window.event.keyCode <= 57 ||
window.event.keyCode >= 96 && window.event.keyCode <= 105 ){
    if (tam <= 2){
        campo.value = vr; }
    if ((tam > 2) && (tam <= 5)) {
        campo.value = vr.substr( 0, tam - 2 ) + ',' +
vr.substr( tam - 2, tam ); }
    if ((tam >= 6) && (tam <= 8)){
        campo.value = vr.substr( 0, tam - 5 ) + '.' +
vr.substr( tam - 5, 3 ) + ',' +
vr.substr( tam - 2, tam ); }
    if ((tam >= 9) && (tam <= 11)){
        campo.value = vr.substr( 0, tam - 8 ) + '.' +
vr.substr( tam - 8, 3 ) + '.' +
vr.substr( tam - 5, 3 ) + ',' + vr.substr( tam - 2, tam );}
    if ((tam >= 12) && (tam <= 14)) {
        campo.value = vr.substr( 0, tam - 11 ) + '.' +
vr.substr( tam - 11, 3 ) + '.' +
vr.substr( tam - 8, 3 ) + '.' + vr.substr( tam - 5, 3 ) +
',' + vr.substr( tam - 2, tam ); }
    if ((tam >= 15) && (tam <= 17)){
        campo.value = vr.substr( 0, tam - 14 ) + '.' +
vr.substr( tam - 14, 3 ) + '.' +
vr.substr( tam - 11, 3 ) + '.' + vr.substr( tam - 8, 3 ) +
',' +
vr.substr( tam - 5, 3 ) + ',' + vr.substr( tam - 2, tam );}})
</script>
</head>

```

9. Construindo o formulário de Consulta

Menu **File** → **Other ...** → **New ASP.Net Files** (lado esquerdo) → **ASP .NET Page** (lado direito)

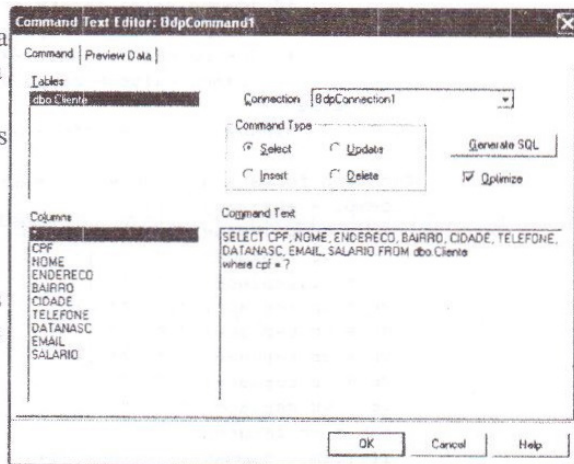
Insira o componente **BdpConnection**. Ligue a propriedade **ConnectionString** com **AliasLoja**

Insira o componente **BdpCommand**. Mude as propriedades:

Connection – ligar com **BdpConnection1**

CommandText – clique em ... selecione todos os campos (**columns**), marque a opção **Select** e **Optimize**, clique no botão **Generate SQL**

Complete o código conforme modelo ao lado,



ou seja, **where cpf=?**

Clique em OK

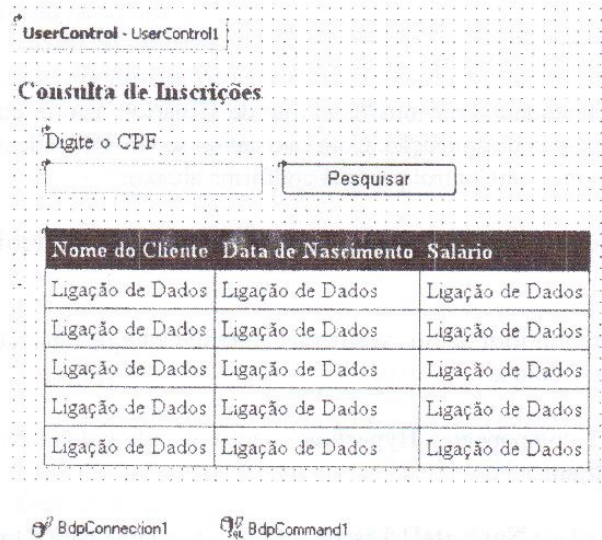
Parameters clique em ... clique no botão **Adicionar** e preencha a tela conforme modelo.

- **BdpType** – String
- **SourceColumn** - CPF
- **ParameterName** - cpf

Mude o **name** do **TextBox** para **txtcpf**

Clique com o botão direito no **DataGrid** e escolha a opção **Criador de Propriedades**.

Selecione a aba **Colunas**. Desmarque a opção **Criar colunas automática/e durante a execução**. Clique em **Coluna Bound** (lado esquerdo) e jogue para à direita (3 vezes).

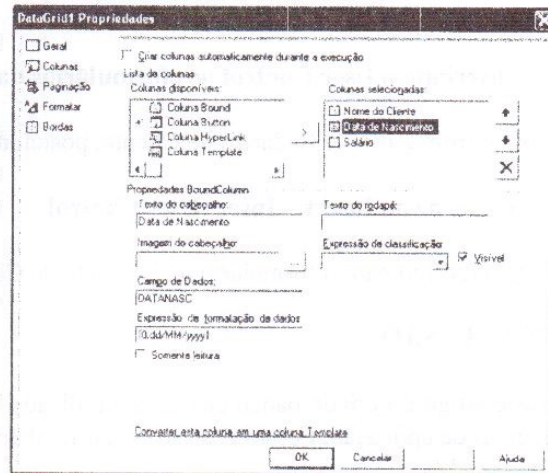


1ª Coluna escreva no **Texto do cabeçalho** - **Nome do Cliente** e no **Campo de Dados** – **NOME**

2ª Coluna escreva no **Texto do cabeçalho** – **Data de Nascimento** e no **Campo de Dados** – **DATANASC** e na **Expressão de formatação de dados** {0:dd/MM/yyyy}

3ª Coluna escreva no **Texto do cabeçalho** – **Salário**, no **Campo de Dados** – **SALARIO**

Para formatar o DataGrid, clique com o botão direito sobre o **datagrid** e escolha **auto-formatação**. Escolha o modelo desejado.



Duplo clique no botão Pesquisar

```
procedure TWebForm2.Button1_Click(sender: System.Object; e: System.EventArgs);  
begin  
    BdpConnection1.Open;  
    BdpCommand1.Parameters[0].Value:=txtcpf.Text;  
    DataGrid1.DataSource := BdpCommand1.ExecuteReader;  
    BdpConnection1.Close;  
    DataGrid1.DataBind;  
end;
```

10. UserControl

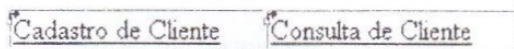
Tem por finalidade criar um “Controle de Usuário”, isto é, controles que poderão ser usados em qualquer página da aplicação. Com isso ganha-se em agilidade e produtividade no desenvolvimento. Para criar um UserControl proceda conforme abaixo:

Menu **File** ↪ **Other ...** ↪ **New ASP.Net Files** (lado esquerdo) ↪ **ASP .NET User Control** (lado direito)

Será aberto um formulário **WebUserControl**. Coloque uma tabela através do menu **Insert - Table** – 1 linha e 2 colunas.

Coloque 2 componentes **Hyperlink** – um em cada coluna e mude a propriedade **Text** conforme mostra a figura.

Na propriedade **NavigateUrl** escolha a referida página para o link, com a extensão **aspx**.



11. Inserindo o User Control nos formulários da aplicação

Abra o formulário de Cadastro de Cliente, posicione o cursor na parte superior da tela.

Clique no menu **Insert – Insert User Control**

Faça o mesmo para o formulário de Consulta de Cliente.

CONCLUSÃO

Neste artigo foi criado banco de dados e utilizado User Controls que serviu para navegação entre as páginas da aplicação. Foram desenvolvidas também, as telas de cadastro e de consulta de clientes.

Com o desenvolvimento dessa aplicação, o programador poderá conhecer recursos básicos do Delphi e .NET Framework.

A tecnologia aplicada neste artigo é de grande valia, pois, grandes empresas de desenvolvimento de software instaladas na cidade e região empregam o .Net em suas aplicações. No entanto, existe uma grande demanda por parte das empresas, mas uma pequena oferta de profissionais qualificados na área.

ABSTRACT

In this work it will be explained how to construct an application directed toward Web containing a register connected to the data base, as well as screen information. The actual technology will be used: ASP.Net and Delphi for .Net. It will be shown all the necessary steps for developing a customer register with a data base SQL Server 2005 for SQL Server, also including : consultation using the customer tax payer register, definition of pre-format, validation of the required scope and email and the control of user among developed forms. For the development of this application it will be necessary to have installed in its machine the SQL Server 2000 and Delphi 2006. This subject needs as prerequisite people who already have programming slight knowledge.

KEYWORDS: *Web. Data base. ASP. Net Technology.*

REFERÊNCIAS

- ALVES, W. P. Delphi 2005 Aplicações de Banco de Dados com InterBase 7.5 e MySQL 4.0.23. São Paulo: Érica.
- CANTÙ, M. Dominando o Delphi 2006. São Paulo: Makron Books.
- PACHECO, X. Guia do Desenvolvedor de Delphi For .net. São Paulo: Pearson Education.
- REVISTA CLUBE DELPHI, Ano 6 Edições 74, 75 e 76.