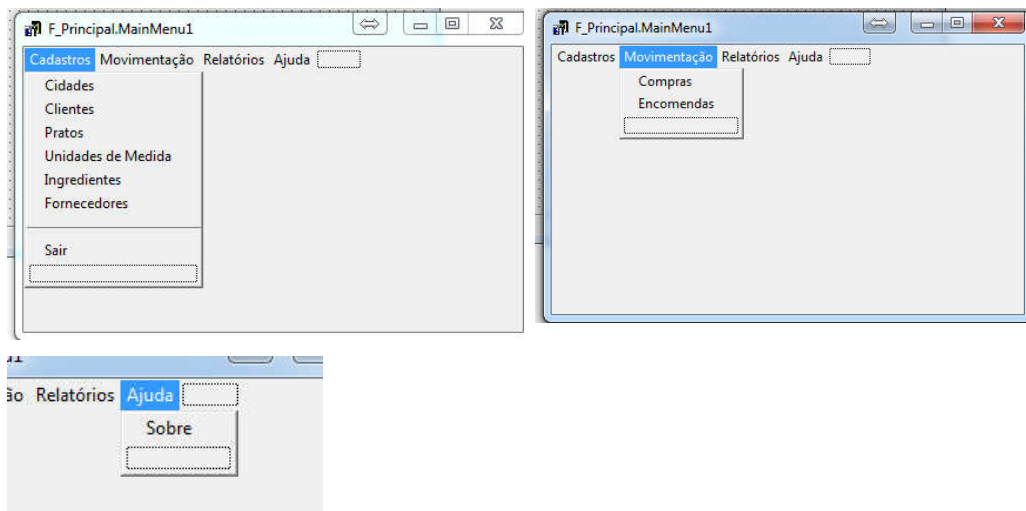


CRIANDO UMA PALICAÇÃO COMPLETA EM DELPHI

1. Abra o Delphi e crie uma nova aplicação VCL Forms.
Menu File => New => VCL Forms Application - Delphi
2. Crie uma pasta específica para o projeto, salve nesta pasta o arquivo do formulário como u_principal, e o nome do arquivo do projeto salve como Restaurante.
3. Altere as seguintes propriedades do formulário:

Propriedade	Valor
Name	F_principal
Caption	Sistema de Gestão Comercial
Form Style	fsMDIForm
WindowState	wsMaximized

4. Adicione no formulário um componente MainMenu, dê um clique duplo no componente e configure os seguintes menus:



5. Adicione um componente StatusBar, dê um clique duplo no componente e insira dois Panels, com os seguintes as seguintes larguras "width" (Panel[0] = 200; e Panel[1] = 500.
6. Adicione um componente Timer e configure as seguintes propriedades:

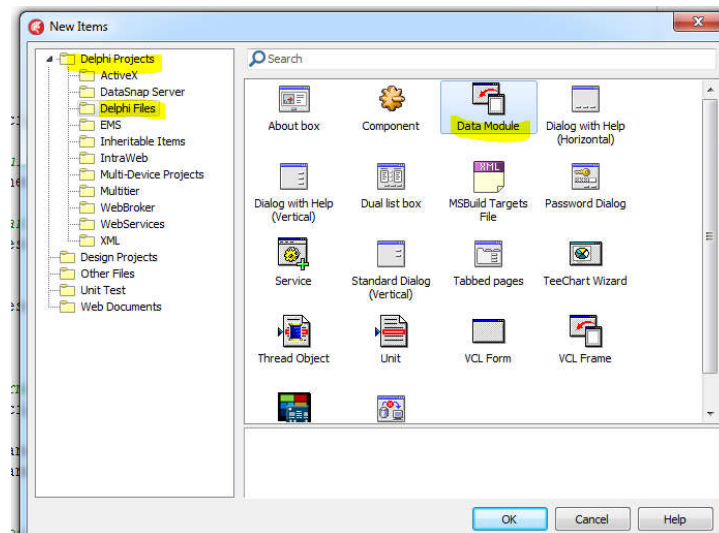
Propriedade	Valor
Enable	True
Interval	1000

7. Programe os seguintes eventos:

```
// OnCreate do Formulário F_principal
procedure TF_Principal.FormCreate(Sender: TObject);
begin
    StatusBar1.Panels[0].Text:= DateTimeToStr(now);
    StatusBar1.Panels[1].Text:= 'Usuário: André Mendes';
end;

// OnTimer do componente Timer
procedure TF_Principal.Timer1Timer(Sender: TObject);
begin
    StatusBar1.Panels[0].Text:= DateTimeToStr(now);
end;
```

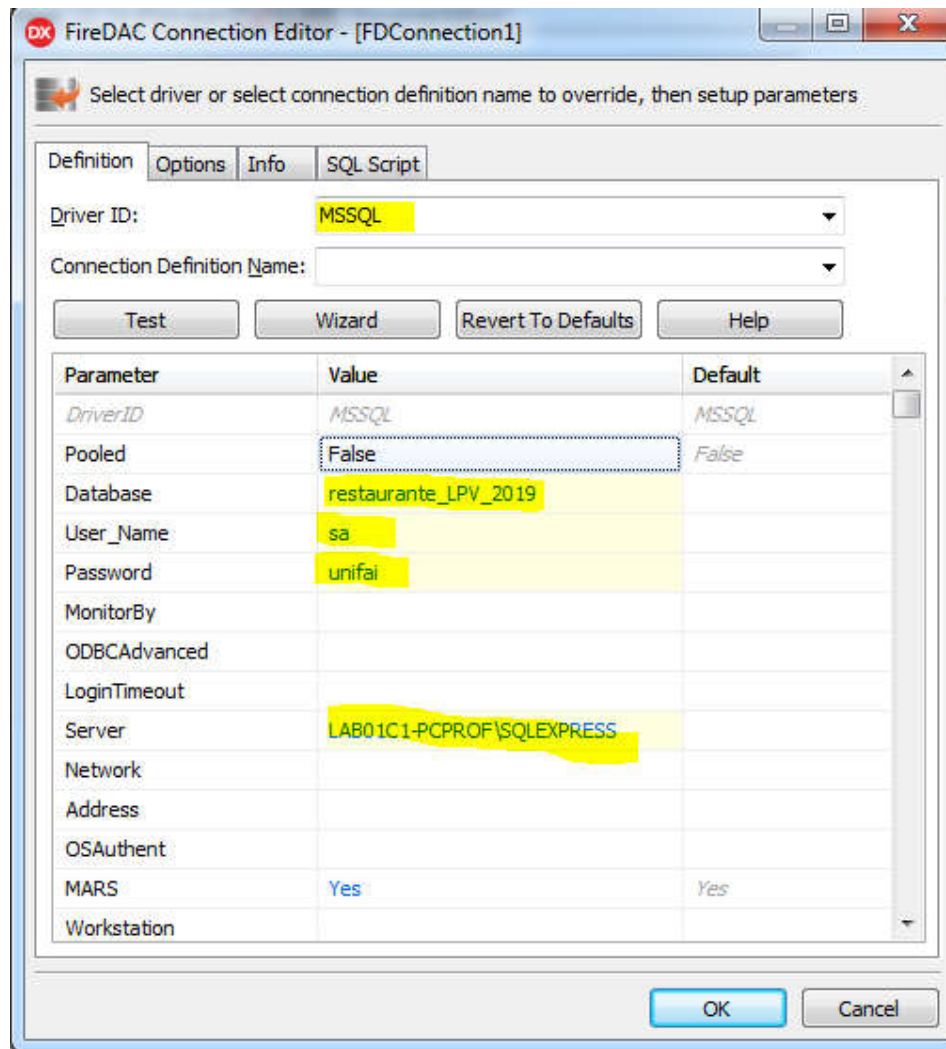
8. Adicione um Data Módulo. Clique em File => New => Other. Selecione o Data Module de acordo com a seguinte imagem:



Salve a unit do data módulo como u_dm. Altere a propriedade **name** para **DM**.

9. Adicione os seguintes componentes FDConnection, FDPhysMSSQLDriverLink, FDGUIxWaitCursor e FDQuery.

10. Crie a base de dados no MS SQL Server com o nome de **restaurante_LPV_2019**, utilizando os scripts sql fornecido, e após isso, dê um duplo clique no FDConnection para configurar as seguintes propriedades:



=====➔ Note que a propriedade **Server**, deve conter o ip ou nome da máquina do servidor de banco de dados. Verifique sua máquina como está.

11. Altere as seguintes propriedades do FDConnection no Object Inspector:

Propriedade	Valor
ConnectionName	MINHA_CONEXAO
LoginPrompt	False
Name	FDConnection1

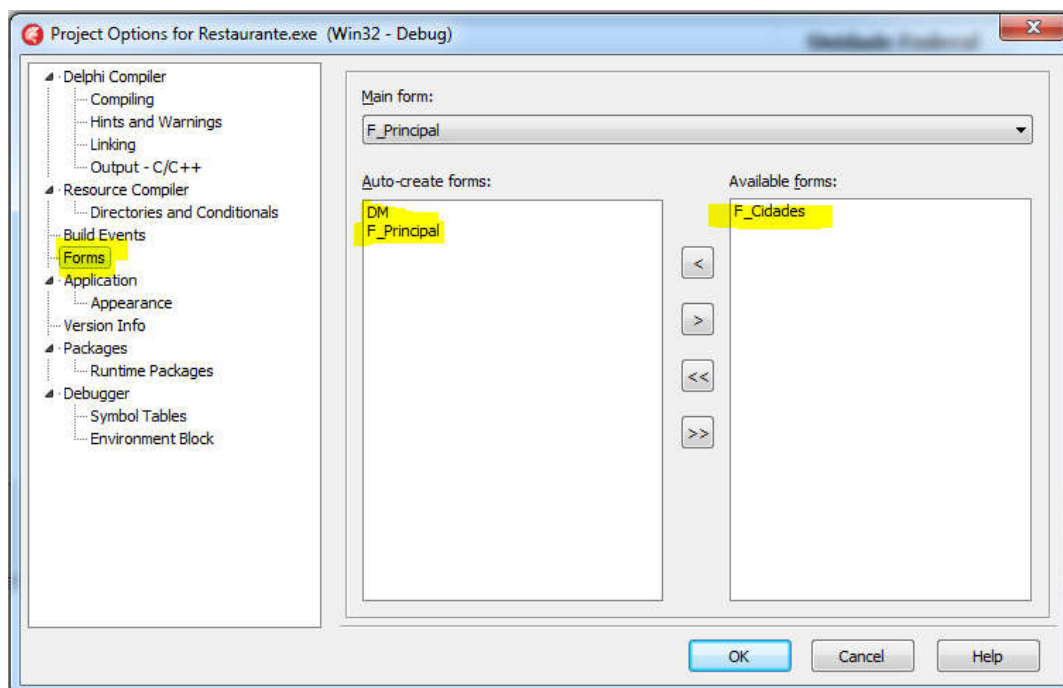
12. Altere as seguintes propriedades do FDQuery:

Propriedade	Valor
Name	Q_geral
Connection	FDConnection1
ConnectionName	MINHA_CONEXAO

13. Adicione um novo Formulário: File => New => VCL Form - Delphi. Salve a unit como o nome **u_cidades**. Altere as seguintes propriedades:

Propriedade	Valor
Name	F_cidades
Caption	Cadastro de Cidades
FormStyle	fsMDIChild
BorderStyle	bsSingle
Position	poScreenCenter

14. Retire o Formulário do Auto-Create e faça com que o Data Modulo seja o primeiro a ser criado arrastando com o Mouse. Clique no menu Project => Options. Consulte a figura abaixo para esta tarefa:



15. Deixe o formulário com o seguinte layout, adicionando os componentes descritos no item 16:

16. Adicione os seguintes componentes e altere as respectivas propriedades:

Panel

Propriedade	Valor
Align	alTop

Bevel : Adicione o componente Bevel dentro do Panel, para o efeito de profundidade.

Edit

Propriedade	Valor
Name	edit_pesquisa
Text	Deixe em branco

FDQuery

Propriedade	Valor
Name	Q_cidades
Connection	DM.FDConnection1
ConnectionName	MINHA_CONEXAO
SQL	Select * from cidades order by nome

DataSource

Propriedade	Valor
Name	DS_Cidades
DataSet	Q_cidades

DBGrid

Propriedade	Valor
DataSource	DS_Cidades
TitleFont	Altere como quiser a fonte de título
Options	dgEditing = false dgRowSelect = true dgAlwaysShowSelection = true

Dê um clique duplo no DBGrid e adicione Três campos, e configure-os no object inspector.

BitBtn

Propriedade	Valor
Name	B_incluir
Caption	Incluir

BitBtn

Propriedade	Valor
Name	B_alterar
Caption	Alterar

BitBtn

Propriedade	Valor
Name	B_excluir
Caption	Excluir

BitBtn

Propriedade	Valor
Name	B_fechar
Caption	Fechar

17. Adicione o seguinte procedimento público na classe do formulário:

Protótipo na classe:

```
public
{ Public declarations }

procedure carregar_query(s : string);
```

Implementação do procedimento:

```
procedure TF_Cidades.carregar_query(s : string);
begin
    q_cidades.SQL.Clear;
    q_cidades.SQL.add(' select * ');
    q_cidades.SQL.add(' from cidades ');
    q_cidades.SQL.add(' where nome like ' + QuotedStr('%'+s+'%') + ' ');
    q_cidades.SQL.add(' order by nome ');
    q_cidades.Open;
    //q_cidades.Active:= true;
    { Observações:
      - A função QuotedStr, coloca entre aspas simples o conteúdo de seu parâmetro
    }
end; // carregar_query
```

18. Programe os seguintes eventos:

```
// OnChange do edit_pesquisa
procedure TF_Cidades.edit_pesquisaChange(Sender: TObject);
begin
    carregar_query(edit_pesquisa.text);
end;

// OnCreate do formulário f_cidades
procedure TF_Cidades.FormCreate(Sender: TObject);
begin
    carregar_query('');
end;

// OnClose de F_Cidades
procedure TF_Cidades.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Action:= caFree;
    f_cidades:= nil;
end;

// OnClose do botão fechar
procedure TF_Cidades.b_fecharClick(Sender: TObject);
begin
    close;
end;
```

// OnClick do botão excluir

```
procedure TF_Cidades.b_excluirClick(Sender: TObject);
begin
    // se a query cidades não conter dados
    if q_cidades.IsEmpty then
    begin
        ShowMessage('Não há um registro selecionado para exclusão !');
        exit;
    end;

    // verificar se há clientes e fornecedores relacionados
    dm.q_geral.sql.Clear;
    dm.q_geral.sql.add(' select count(*) as total ');
    dm.q_geral.sql.add(' from CLIENTES ');
    dm.q_geral.sql.add(' where cod_cidade = ' + QuotedStr(q_cidades.FieldByName('cod_cidade').AsString) );
    dm.q_geral.Open;
    if dm.Q_Geral.FieldByName('total').AsInteger > 0 then
    begin
        ShowMessage('Não é possível excluir essa cidade porque há clientes relacionados !');
        exit;
    end;

    dm.q_geral.sql.Clear;
    dm.q_geral.sql.add(' select count(*) as total ');
    dm.q_geral.sql.add(' from FORNECEDORES ');
    dm.q_geral.sql.add(' where cod_cidade = ' + QuotedStr(q_cidades.FieldByName('cod_cidade').AsString) );
    dm.q_geral.Open;
    if dm.Q_Geral.FieldByName('total').AsInteger > 0 then
    begin
        ShowMessage('Não é possível excluir essa cidade porque há fornecedores relacionados !');
        exit;
    end;

    if MessageDlg('Deseja realmente excluir a cidade ' + q_cidades.FieldByName('nome').AsString + ' ??',
        mtConfirmation, [mbYes, mbNo], 0) = mrYes then
    begin
        dm.q_geral.sql.Clear;
        dm.q_geral.sql.add(' delete from cidades ');
        dm.q_geral.sql.add(' where cod_cidade = ' + QuotedStr(q_cidades.FieldByName('cod_cidade').AsString) );
        dm.q_geral.ExecSql;
        // observação:
        // o Método ExecSql é utilizado para os comandos SQL: Insert, Update e Delete
        // o Método Open é utilizado para o comando SELECT

        // Atualizando a query q_cidades
        carregar_query('');
    end;
end; // fim do onclick do botão excluir
```

19. Adicione o seguinte código no menu Cidades do formulário principal, para a abrir o

Cadastro de Cidades:

```
procedure TF_Principal.Cidades1Click(Sender: TObject);
begin
    // se o formulário não estiver instanciado
    if not Assigned(F_cidades) then
    begin
        // instanciando o formulário
        F_cidades:= TF_Cidades.create(application);
    end
    else begin
        F_cidades.Show;
    end;
end;
```


20. Adicione um novo Formulário: File => New => VCL Form - Delphi. Salve a unit como o nome **u_cidades_ficha**. Altere as seguintes propriedades:

Propriedade	Valor
Name	F_cidades_ficha
Caption	Cadastro de Cidades
FormStyle	fsMDIChild
BorderStyle	bsSingle
Position	poScreenCenter

Observação: Retire o formulário do auto-create.

21. Adicionar dois BitBtn e altere as seguintes propriedades:

Propriedade	Valor
Name	B_gravar
Kind	bkOk
ModalResult	mrNone
Default	False

Propriedade	Valor
Name	B_cancelar
Kind	bkCancel
Cancel	False

22. Adicione um panel, apague seu caption, e mude a propriedade Align para alTop. Adicione um bevel no interior do panel.

23. Adicione os seguintes componentes e deixe parecido com o seguinte layout de tela:

TEdit

Propriedade	Valor
Name	Edit_nome
Text	apague
MaxLength	100

TEdit

Propriedade	Valor
Name	Edit_uf
Text	apague
MaxLength	2

TMaskEdit

Propriedade	Valor
Name	Edit_cep_padrao
Text	apague
EditMask	99999-999;0;_

24. Adicione a seguinte propriedade e protótipo de procedimento na área pública da classe do formulário f_cidades_ficha:

```

{ Private declarations }
public
{ Public declarations }

cod_cidade : string;
procedure carregar_dados;

```

25. Faça a implementação do procedimento carregar_dados:

```
procedure Tf_cidades_ficha.carregar_dados;
begin
    dm.q_geral.sql.clear;
    dm.q_geral.sql.add(' select * ');
    dm.q_geral.sql.add(' from cidades ');
    dm.q_geral.sql.add(' where cod_cidade = ' + QuotedStr(cod_cidade) + ' ');
    dm.q_geral.open;

    if dm.q_geral.isempty then
    begin
        ShowMessage('Cidade não encontrada !');
        close;
        exit;
    end;

    edit_nome.Text:= dm.q_geral.fieldbyname('nome').AsString;
    edit_uf.Text:= dm.q_geral.fieldbyname('uf').AsString;
    edit_cep_padrao.Text:= dm.q_geral.fieldbyname('cep_padrao').AsString;

end;
```

26. Programe os seguintes eventos na unit de f_cidades_ficha:

```
procedure Tf_cidades_ficha.b_gravarClick(Sender: TObject);
// OnClick do botão gravar
begin
    if trim(edit_nome.Text) = '' then
    begin
        ShowMessage('O campo nome deve ser preenchido !');
        exit;
    end;

    if trim(edit_uf.Text) = '' then
    begin
        ShowMessage('O campo Unidade Federal deve ser preenchido !');
        exit;
    end;

    if cod_cidade = '' then
    begin
        dm.q_geral.sql.clear;
        dm.q_geral.sql.add('insert into cidades (nome, uf, cep_padrao) values (:nome, :uf, :cep_padrao) ');

        dm.q_geral.ParamByName('nome').AsString:= trim(edit_nome.Text);
        dm.q_geral.ParamByName('uf').AsString:= trim(edit_uf.Text);
        dm.q_geral.ParamByName('cep_padrao').AsString:= trim(edit_cep_padrao.Text);
    end
    else begin
        dm.q_geral.sql.clear;
        dm.q_geral.sql.add(' update cidades set ');
        dm.q_geral.sql.add('     nome = :nome, ');
        dm.q_geral.sql.add('     uf = :uf, ');
        dm.q_geral.sql.add('     cep_padrao = :cep_padrao ');
        dm.q_geral.sql.add(' where cod_cidade = :cod_cidade ');

        dm.q_geral.ParamByName('nome').AsString:= trim(edit_nome.Text);
        dm.q_geral.ParamByName('uf').AsString:= trim(edit_uf.Text);
        dm.q_geral.ParamByName('cep_padrao').AsString:= trim(edit_cep_padrao.Text);
        dm.q_geral.ParamByName('cod_cidade').AsString:= cod_cidade;
    end;

    dm.q_geral.ExecSql;
    f_cidades.carregar_query('');
    close;
end;

procedure Tf_cidades_ficha.FormClose(Sender: TObject; var Action: TCloseAction);
// OnClose do Formulário
begin
    Action:= caFree;
    f_cidades_ficha:= nil;
end;

procedure Tf_cidades_ficha.b_cancelarClick(Sender: TObject);
// OnClick do botão Cancelar
begin
    close;
end;
```

27. Programe os seguintes eventos na unit de f_cidades:

```
procedure TF_Cidades.b_incluirClick(Sender: TObject);
// OnClick do botão Incluir
begin
    // se o formulário não estiver instanciado
    if not Assigned(F_cidades_ficha) then
    begin
        // instanciando o formulário
        F_cidades_ficha:= TF_cidades_ficha.create(application);
        F_cidades_ficha.cod_cidade:= '';
    end
    else begin
        F_cidades_ficha.Show;
    end;
end;

end;

procedure TF_Cidades.b_alterarClick(Sender: TObject);
// OnClick do botão alterar
begin
    // se a query cidades não conter dados
    if q_cidades.IsEmpty then
    begin
        ShowMessage('Não há um registro selecionado para alterar !');
        exit;
    end;

    // se o formulário não estiver instanciado
    if not Assigned(F_cidades_ficha) then
    begin
        // instanciando o formulário
        F_cidades_ficha:= TF_cidades_ficha.create(application);
        F_cidades_ficha.cod_cidade:= q_cidades.FieldByName('cod_cidade').AsString;
        F_cidades_ficha.carregar_dados;
    end
    else begin
        F_cidades_ficha.Show;
    end;
end;

end;
```