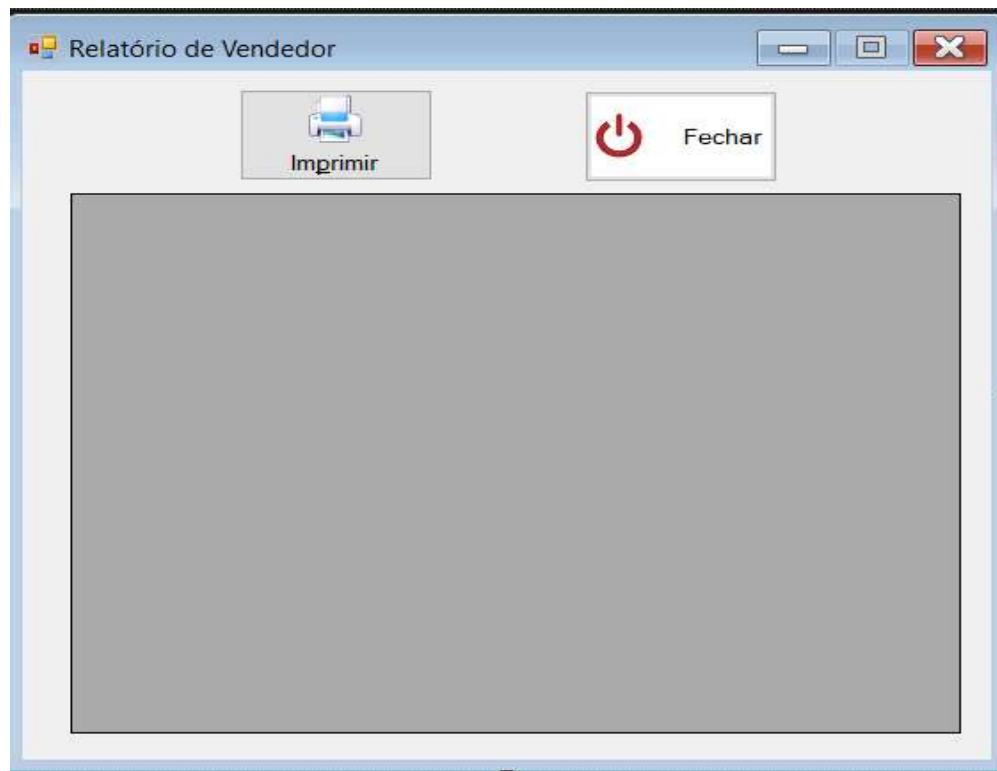


Relatório de Vendedores em cSharp - Controle PrintDocument

Crie a página **formulário frmRelVendedor.cs** conforme imagem abaixo:



Text: Relatório de Vendedor

StartPosition: CenterScreen

MaximizeBox: False

Button - Name: btnImprimir - **Name:** Im&primir

Button - Name: btnFechar - **Name:** &Fechar

DataGridView - Name: dgvVendedor

PrintDocument - Name: prdVendedor

PrintPreviewDocument - Name: ppdVendedor

Codificação - Duplo clique no botão Fechar. Insira o código abaixo:

`this.Dispose(); // Fecha e descarrega o formulário da memória.`

Duplo clique no botão Imprimir. Insira o código abaixo:

`// Configuração do PrintDocument`

`prdVendedor.PrintPage += new PrintPageEventHandler(printDocument1_PrintPage);`

`// Exibe o diálogo de visualização de impressão`

`ppdVendedor.Document = prdVendedor;`

`ppdVendedor.ShowDialog();`

Duplo clique no controle prdVendedor. Insira o código abaixo:

`Font fonte = new Font("Arial", 12);`

`int x = e.MarginBounds.Left;`

`int y = e.MarginBounds.Top;`

`// Imprime os títulos do relatório`

`e.Graphics.DrawString("Relatório do Cadastro de Vendedores", fonte, Brushes.Black, x, y);`

```
// Ajusta a posição y para a próxima linha
y += fonte.Height;
y += fonte.Height;

// Deixa a largura da coluna para cada título
// MeasureString --> Calcula se o texto desejado caberá na folha
int larguraColunaCodigo = (int)e.Graphics.MeasureString("Código", fonte).Width;
int larguraColunaNome = (int)e.Graphics.MeasureString("Nome do Vendedor", fonte).Width;
int larguraColunaContato = (int)e.Graphics.MeasureString("Contato", fonte).Width;

// Adiciona um espaçamento para melhor visualizar os registros
larguraColunaCodigo += 10;
larguraColunaNome += 150;
larguraColunaContato += 10;

// Imprime os títulos das colunas
e.Graphics.DrawString("Código", fonte, Brushes.Black, x, y);
x += larguraColunaCodigo;
e.Graphics.DrawString("Nome do Vendedor", fonte, Brushes.Black, x, y);
x += larguraColunaNome;
e.Graphics.DrawString("Contato", fonte, Brushes.Black, x, y);
y += fonte.Height; // Avança para a próxima linha

x = e.MarginBounds.Left; // Reinicia a posição x

// Loop para exibir os dados no relatório
foreach (DataGridViewRow linha in dgvVendedor.Rows)
{
    e.Graphics.DrawString(linha.Cells["Codigo"].Value.ToString(), fonte, Brushes.Black, x, y);
    x += larguraColunaCodigo;

    e.Graphics.DrawString(linha.Cells["Nome_Vendedor"].Value.ToString(), fonte, Brushes.Black, x, y);
    x += larguraColunaNome;

    e.Graphics.DrawString(linha.Cells["Contato"].Value.ToString(), fonte, Brushes.Black, x, y);
    x += larguraColunaContato;

    y += fonte.Height; // Avança para a próxima linha
    x = e.MarginBounds.Left; // Reinicia a posição x
}
```

Dentro do construtor / método public frmRelVendedor, deixe como abaixo:

InitializeComponent();

exibirGrid(); // Assim que inicializa os controntes, chama e exibe os dados no grid

Será exibida mensagem de erro, na linha do **exibirGrid();**

Logo após a chave que fecha o método acima, crie o método **exibirGrid()**. Deixe como abaixo:

private void exibirGrid()

```
{
    clsVendedor classeVendedor = new clsVendedor();
```

```
dgvVendedor.DataSource = classeVendedor.pesquisa_Vendedor_Rel("");  
cabecalho();  
}
```

Como o método cabecalho não existe, será necessário cria-lo.

private void cabecalho()

```
{  
dgvVendedor.Columns[0].HeaderText = "Código";  
dgvVendedor.Columns[0].Width = 45;  
dgvVendedor.Columns[0].DefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleRight;  
dgvVendedor.Columns[1].HeaderText = "Nome do vendedor";  
dgvVendedor.Columns[1].Width = 160;  
dgvVendedor.Columns[2].HeaderText = "Contato";  
dgvVendedor.Columns[2].Width = 90;  
}
```

Arquivos importantes

Classe clsConexao

```
namespace controleComissao  
{  
public class clsConexao  
{  
private static string conexao = @"Data Source=.\SQLEXPRESS;Initial Catalog=bdComissao; Integrated  
Security=True";  
  
public static string stringConexao  
{  
get { return conexao; }  
}  
}  
}
```

Classe clsVendedor

```
using System;  
using System.Collections.Generic;  
using System.Data;  
using System.Data.SqlClient;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace controleComissao  
{  
public class clsVendedor  
{  
SqlConnection cn = new SqlConnection(); // A classe SqlConnection permite realizar a conexão com o banco
```

de dados

```
SqlCommand cmd = new SqlCommand(); // A classe SqlCommand permite realizar insert, update, delete  
DataTable dt = new DataTable(); // A classe DataTable permite realizar pesquisas na tabela
```

```
public void incluir(string nomeVendedor, string email, string contato, double percentualComissao)  
{  
    SqlConnection cn = new SqlConnection(clsConexao.stringConexao);  
    try  
    {  
        cn.Open(); // Abre a conexão  
        string sql = "INSERT INTO tbVendedor (nome_Vendedor, senha, email, contato, percentual_Comissao)";  
        sql += " VALUES (@nome_Vendedor, @email, @contato, @percentual_Comissao)";  
        cmd.Connection = cn;  
        cmd.Parameters.Add(new SqlParameter("@nome_Vendedor", nomeVendedor));  
        cmd.Parameters.Add(new SqlParameter("@email", email));  
        cmd.Parameters.Add(new SqlParameter("@contato", contato));  
        cmd.Parameters.Add(new SqlParameter("@percentual_Comissao", percentualComissao));  
        cmd.CommandText = sql; // Para a string inserir que foi montada acima para a instancia cmd  
        cmd.ExecuteNonQuery(); // Executa no banco de dados  
    }  
    catch (Exception erro)  
    {  
        throw new Exception(erro.Message); // Permite a exibição de mensagem de erro  
    }  
    finally  
    {  
        cn.Close(); // Fecha a conexão  
        cn.Dispose(); // Descarrega a conexão da memória  
        cmd.Dispose(); // Descarrega o objeto command da memória  
    }  
}
```

```
public void editar(int codigo, string nomeVendedor, string email, string contato, double percentualComissao,  
string situacao)  
{  
    SqlConnection cn = new SqlConnection(clsConexao.stringConexao);  
    try  
    {  
        cn.Open(); // Abre a conexão  
        string sql = "UPDATE tbVendedor SET nome_Vendedor = @nome_Vendedor, email = @email, contato =  
@contato, percentual_Comissao = @percentual_Comissao, situacao = @situacao";  
        sql += " WHERE codigo = @codigo";  
        cmd.Connection = cn;  
        cmd.Parameters.Add(new SqlParameter("@nome_Vendedor", nomeVendedor));  
        cmd.Parameters.Add(new SqlParameter("@email", email));  
        cmd.Parameters.Add(new SqlParameter("@contato", contato));  
        cmd.Parameters.Add(new SqlParameter("@situacao", situacao));  
        cmd.Parameters.Add(new SqlParameter("@percentual_Comissao", percentualComissao));  
        cmd.Parameters.Add(new SqlParameter("@codigo", codigo));  
        cmd.CommandText = sql; // Para a string inserir que foi montada acima para a instancia cmd  
        cmd.ExecuteNonQuery(); // Executa no banco de dados  
    }  
    catch (Exception erro)
```

```
{
throw new Exception(erro.Message); // Permite a exibição de mensagem de erro
}
finally
{
cn.Close(); // Fecha a conexão
cn.Dispose(); // Descarrega a conexão da memória
cmd.Dispose(); // Descarrega o objeto command da memória
}
}

public void editarSenha(int codigo, string nomeVendedor, string senha)
{
SqlConnection cn = new SqlConnection(clsConexao.stringConexao);
try
{
cn.Open(); // Abre a conexão
string sql = "UPDATE tbVendedor SET nome_Vendedor = @nome_Vendedor, senha = @senha";
sql += " WHERE codigo = @codigo";
cmd.Connection = cn;
cmd.Parameters.Add(new SqlParameter("@nome_Vendedor", nomeVendedor));
cmd.Parameters.Add(new SqlParameter("@senha", senha));
cmd.Parameters.Add(new SqlParameter("@codigo", codigo));
cmd.CommandText = sql; // Para a string inserir que foi montada acima para a instancia cmd
cmd.ExecuteNonQuery(); // Executa no banco de dados
}
catch (Exception erro)
{
throw new Exception(erro.Message); // Permite a exibição de mensagem de erro
}
finally
{
cn.Close(); // Fecha a conexão
cn.Dispose(); // Descarrega a conexão da memória
cmd.Dispose(); // Descarrega o objeto command da memória
}
}

public void excluir(int codigo)
{
SqlConnection cn = new SqlConnection(clsConexao.stringConexao);
try
{
cn.Open(); // Abre a conexão
string excluir = "DELETE FROM tbVendedor";
excluir += " WHERE codigo = @codigo";
cmd.Connection = cn;
cmd.Parameters.Add(new SqlParameter("@codigo", codigo));
cmd.CommandText = excluir;
cmd.ExecuteNonQuery();
}
catch (Exception erro)
{
}
```

```
throw new Exception(erro.Message); // Permite a exibição de mensagem de erro
}
finally
{
    cn.Close(); // Fecha a conexão
    cn.Dispose(); // Descarrega a conexão da memória
    cmd.Dispose(); // Descarrega o objeto command da memória
}
}

public DataTable pesquisaCodigo(int codigo)
{
    SqlConnection cn = new SqlConnection(clsConexao.stringConexao);
    try
    {
        cn.Open();
        clsGlobal.pesquisa += " WHERE codigo = @codigo";
        clsGlobal.pesquisa += " ORDER BY nome_Vendedor";
        cmd.Parameters.Add(new SqlParameter("@codigo", codigo));
        cmd.CommandText = clsGlobal.pesquisa.ToString();
        cmd.Connection = cn;
        dt.Load(cmd.ExecuteReader());
        return dt;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
    finally
    {
        cn.Close();
        cn.Dispose();
        cmd.Dispose();
    }
}

public DataTable pesquisa_Vendedor(string nomeVendedor)
{
    SqlConnection cn = new SqlConnection(clsConexao.stringConexao);
    try
    {
        cn.Open();
        //string pesquisa = "select idPessoa as Id, nomePessoa as Nome, emailPessoa as 'E-mail', telefonePessoa as
        Telefone from tbPessoa ";
        clsGlobal.pesquisa += " Where nome_Vendedor LIKE '%' + @nomeVendedor + '%'";
        cmd.Parameters.Add(new SqlParameter("@nomeVendedor", nomeVendedor));
        cmd.CommandText = clsGlobal.pesquisa.ToString();
        cmd.Connection = cn;
        dt.Load(cmd.ExecuteReader());
        return dt;
    }
    catch (Exception erro)
    {

```

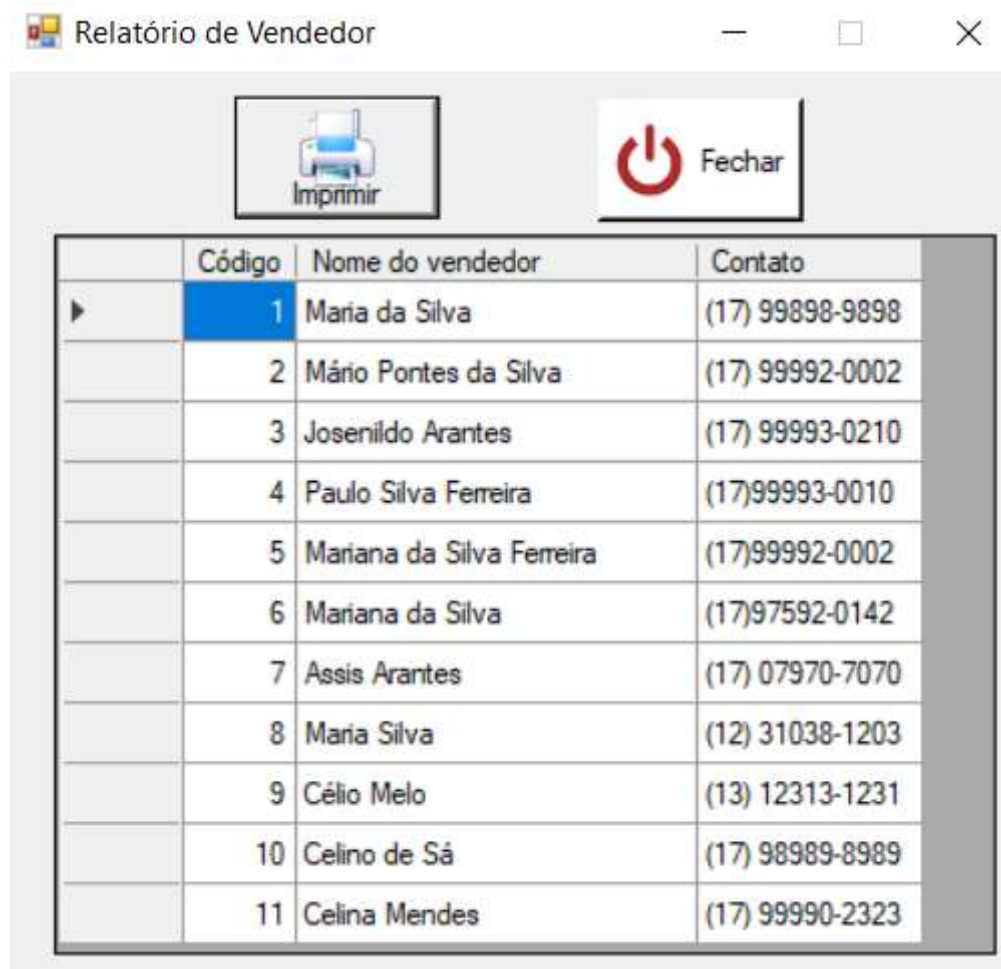
```
throw new Exception(erro.Message); // Permite a exibição de mensagem de erro
}
finally
{
    cn.Close(); // Fecha a conexão
    cn.Dispose(); // Descarrega a conexão da memória
    cmd.Dispose(); // Descarrega o objeto command da memória
}
}

public DataTable pesquisa_Vendedor_Senha(string nomeVendedor, string senha)
{
    SqlConnection cn = new SqlConnection(clsConexao.stringConexao);
    try
    {
        cn.Open();
        clsGlobal.pesquisa += " Where nome_Vendedor = @nomeVendedor AND senha = @senha";
        cmd.Parameters.Add(new SqlParameter("@nomeVendedor", nomeVendedor));
        cmd.Parameters.Add(new SqlParameter("@senha", senha));
        cmd.CommandText = clsGlobal.pesquisa.ToString();
        cmd.Connection = cn;
        dt.Load(cmd.ExecuteReader());
        return dt;
    }
    catch (Exception erro)
    {
        throw new Exception(erro.Message);
    }
    finally
    {
        cn.Close();
        cn.Dispose();
        cmd.Dispose();
    }
}

public DataTable pesquisa_Vendedor_Rel(string nomeVendedor)
{
    SqlConnection cn = new SqlConnection(clsConexao.stringConexao);
    try
    {
        cn.Open();
        string pesquisa = "SELECT Codigo, Nome_Vendedor, Contato FROM tbvendedor Where nome_Vendedor LIKE '%" + @nomeVendedor + '%"';
        cmd.Parameters.Add(new SqlParameter("@nomeVendedor", nomeVendedor));
        cmd.CommandText = pesquisa.ToString();
        cmd.Connection = cn;
        dt.Load(cmd.ExecuteReader());
        return dt;
    }
    catch (Exception erro)
    {
        throw new Exception(erro.Message);
    }
}
```

```
}  
finally  
{  
cn.Close();  
cn.Dispose();  
cmd.Dispose();  
}  
}  
}  
}
```

Pressione a tecla **F5**. Veja abaixo como deverá ficar:



Clique no botão **Imprimir**. Será exibida a janela:

Visualizar impressão

Fechar

Relatório do Cadastro de Vendedores

Código	Nome do Vendedor	Contato
1	Maria da Silva	(17) 99898-9898
2	Mário Pontes da Silva	(17) 99992-0002
3	Josenildo Arantes	(17) 99993-0210
4	Paulo Silva Ferreira	(17)99993-0010
5	Mariana da Silva Ferreira	(17)99992-0002
6	Mariana da Silva	(17)97592-0142
7	Assis Arantes	(17) 07970-7070
8	Maria Silva	(12) 31038-1203
9	Célio Melo	(13) 12313-1231
10	Celino de Sá	(17) 98989-8989
11	Celina Mendes	(17) 99990-2323

Se desejar imprimir, clique no primeiro ícone (botão com símbolo de impressora).