

Prof. Roberto de Castro

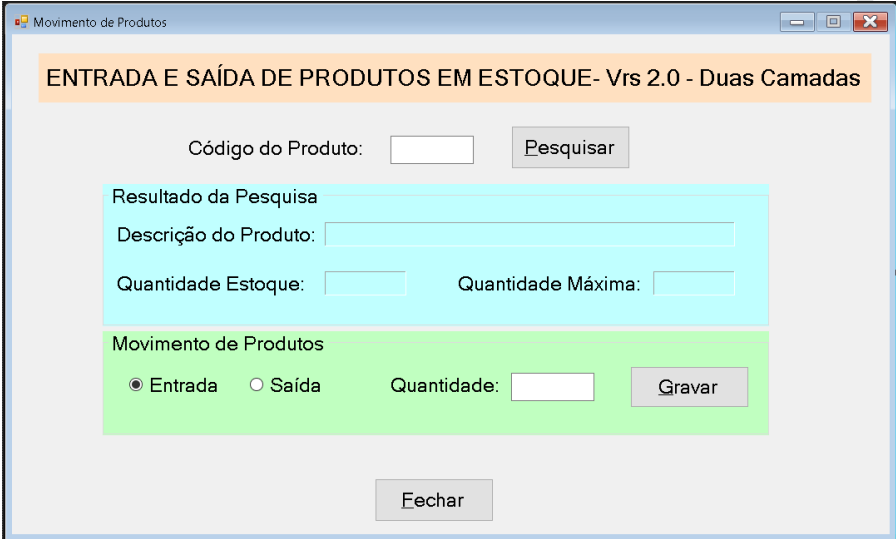
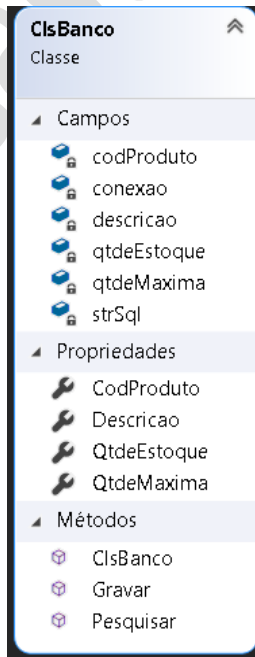
Escola Técnica Estadual “São Paulo” - ETESP

Linguagem C# - Visual Studio 2019

Prof. Roberto de Castro

***** SOLUÇÃO ATIVIDADE *****

**Converter o projeto em DUAS CAMADAS
(utilização de classe)**

FORMULÁRIO	DIAGRAMA DE CLASSE
	

O desenvolvimento da atividade poderá ser em dupla e valerá como desenvolvimento de atividade (parte da menção do bimestre)

A solução da atividade (programação “formulário” + programação classe) DEVERÁ ser copiada para um documento Word (.doc) e encaminhada **EXCLUSIVAMENTE** pelo TEAMS (**utilize o chat para encaminhamento direto ao professor**), **até o dia 05/12 (impreterivelmente)**. **Encaminhar SOMENTE o texto copiado no Word!**

IMPORTANTE: Encaminhamento fora do prazo (após o dia 05/12) e/ou não encaminhado pelo TEAMS diretamente ao professor, será desconsiderado!

SOLUÇÃO

Formulário:

```
using System;
using System.Windows.Forms;

//a linha abaixo é necessária para a utilização da classe
//dentro da pasta AcessoBD
using Exe039A_BD005_MovtoProd.AcessoBD;

namespace Exe039A_BD005_MovtoProd
{
    public partial class FrmMovimento : Form
    {
        public FrmMovimento()
        {
            InitializeComponent();
        }

        private void BtnFechar_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void BtnPesquisar_Click(object sender, EventArgs e)
        {
            //Falta validar se o código do produto foi digitado e se é numérico!!
            int varCodProduto = Convert.ToInt32(TxtCodProduto.Text);

            //Criação da instância da classe. Este momento o método construtor
            //será executado!!
            ClsBanco objBanco = new ClsBanco();

            objBanco.Pesquisar(varCodProduto);

            LblDescricao.Text = objBanco.Descricao;
            LblEstoque.Text = objBanco.QtdeEstoque.ToString();
            LblMaxima.Text = objBanco.QtdeMaxima.ToString();

            if (LblDescricao.Text == "") //Produto não localizado
            {
                MessageBox.Show("Produto não existe!!", "ATENÇÃO");
                TxtCodProduto.Text = "";
                TxtCodProduto.Focus();
            }
        }
    }
}
```

Prof. Roberto de Castro

```
private void BtnGravar_Click(object sender, EventArgs e)
{
    //Neste procedimento falta implementar testes no TxtQuantidade para validar dado
    //numérico e > 0
    //Falta também validar se após a entrada de mercadoria em estoque, se o estoque
    //atualizado é maior que
    //a QtdeMaxima - isto não pode ocorrer
    //Falta verificar também se após a saída de mercadoria em estoque, se o estoque
    //atualizado é negativo
    //isto não pode ocorrer!!
    int estoqueAtual = Convert.ToInt32(LblEstoque.Text);
    int qtdeEntrada = Convert.ToInt32(TxtQuantidade.Text);
    int estoqueAtualizado = 0;

    if (RdbEntrada.Checked == true)
    {
        estoqueAtualizado = estoqueAtual + qtdeEntrada;
    }
    else
    {
        estoqueAtualizado = estoqueAtual - qtdeEntrada;
    }

    int varCodProduto = Convert.ToInt32(TxtCodProduto.Text);

    //Criação da instância da classe. Neste momento o método construtor
    //será executado!!
    ClsBanco objBanco = new ClsBanco();

    int status = objBanco.Gravar(varCodProduto, estoqueAtualizado);

    if (status == 0)
    {
        MessageBox.Show("Erro na atualização!!", "ATENÇÃO");
    }
    else
    {
        MessageBox.Show("Dados Salvos com sucesso.");
    }

    //Limpar o formulário
    TxtCodProduto.Text = "";
    LblDescricao.Text = "";
    LblEstoque.Text = "";
    LblMaxima.Text = "";
    TxtQuantidade.Text = "";
    RdbEntrada.Checked = true;
    TxtCodProduto.Focus();
}
}
```

Prof. Roberto de Castro

PROGRAMAÇÃO CLASSE – ClsBanco

```
using System;
//Linhas incluídas
using System.Data;
using System.Data.OleDb;

using System.Windows.Forms; //inclusão necessária para utilizar o MessageBox

namespace Exe039A_BD005_MovtoProd.AcessoBD
{
    class ClsBanco
    {
        //Declaração das variáveis globais:
        string strSql = "";

        //cria a conexão com o banco de dados
        OleDbConnection conexao = new
            OleDbConnection(Properties.Settings.Default.ConexaoBD);

        //Descrição dos atributos - campos
        private int codProduto;
        private string descricao;
        private int qtdeMaxima;
        private int qtdeEstoque;

        //Declaração dos gets/sets - propriedades
        //Observar que algumas propriedades são somente GET (só leitura)
        //e outras somente GET/SET (leitura e gravação)
        public int CodProduto { get => codProduto; }
        public string Descricao { get => descricao; }
        public int QtdeMaxima { get => qtdeMaxima; }
        public int QtdeEstoque { get => qtdeEstoque; set => qtdeEstoque = value; }

        //Declaração do construtor
        public ClsBanco()
        {
            //abre a conexão com o banco
            try
            {
                conexao.Open();
            }
            catch
            {
                MessageBox.Show("Erro ao abrir o banco de dados. VERIFIQUE!!");
            }
        }
    }
}
```

```
//Método pesquisar
public void Pesquisar(int varCodProduto)
{
    //cria o objeto command para executar a instrução sql
    OleDbCommand comando = new OleDbCommand(strSql, conexao);

    comando.CommandType = CommandType.Text;

    //adiciona parametros
    comando.Parameters.Add("@CodProd", SqlDbType: SqlDbType.Integer).Value =
        varCodProduto;

    //define a instrução SQL
    strSql = "SELECT Produtos.CodProd, Produtos.Descricao, Produtos.QtdeMaxima,
        Produtos.QtdeEstoque FROM Produtos Where Produtos.CodProd = @CodProd";
    comando.CommandText = strSql;

    //cria o objeto DataReader e executa o comando SQL
    OleDbDataReader myDataReader = comando.ExecuteReader();

    //Realiza a leitura do DataReader. Isto se faz necessário para que os dados nele
    //armazenados possam ser manipulados
    myDataReader.Read();

    //Se o comando de leitura retornou com algum registro,
    //transfere os dados para os campos da classe.
    if (myDataReader.HasRows)
    {
        descricao = myDataReader["Descricao"].ToString();
        qtdeMaxima = Convert.ToInt32(myDataReader["QtdeMaxima"].ToString());
        qtdeEstoque = Convert.ToInt32(myDataReader["QtdeEstoque"].ToString());
    }
    else
    {
        descricao = "";
        qtdeMaxima = 0;
        QtdeEstoque = 0;
    }

    //Fecha a conexao
    conexao.Close();
}

public int Gravar(int varCodProduto, int estoqueAtualizado)
{
    //Cria o comando que inicia a query
    OleDbCommand comando = new OleDbCommand(strSql, conexao);

    comando.CommandType = CommandType.Text;

    //adiciona parametros
    //IMPORTANTE: Os parametros PRECISAM estar na ordem em que são
    //              incluídos no comando SQL
    comando.Parameters.Add("@EstoqueAtualizado", SqlDbType: SqlDbType.Integer).Value
        = estoqueAtualizado;
    comando.Parameters.Add("@CodProd", SqlDbType: SqlDbType.Integer).Value =
        varCodProduto;
}
```

Prof. Roberto de Castro

```
//define a instrução SQL
strSql = "UPDATE Produtos set QtdeEstoque = @EstoqueAtualizado where CodProd =
        @CodProd";
```

```
comando.CommandText = strSql;
```

```
// executa a query
int status = comando.ExecuteNonQuery();
```

```
//fecha a conexao
conexao.Close();
```

```
return status;
```

```
}
```

```
}
```

```
}
```