

O projeto a seguir apresenta o modelo de acesso a dados em UMA CAMADA e servirá como estudo e/ou comparativo com a programação anterior (duas camadas) e servirá, também, como “apoio” para o desenvolvimento da próxima atividade (ver ao final deste material).

Apresenta utilização do comando “SQL” sem a inclusão de “parâmetros” (utiliza o conceito de “concatenação de variáveis”). A utilização desta prática não é aconselhável pois **permitirá a ocorrência de “injeção de SQL”**, que nada mais é do que um ataque em que um código mal-intencionado é inserido em cadeias de caracteres. Veja o exemplo abaixo, extraído do link:

[“https://learn.microsoft.com/pt-br/sql/relational-databases/security/sql-injection?view=sql-server-ver16](https://learn.microsoft.com/pt-br/sql/relational-databases/security/sql-injection?view=sql-server-ver16)

O script a seguir mostra uma injeção SQL simples. O script cria uma consulta SQL concatenando cadeias de caracteres codificadas com uma cadeia de caracteres inserida pelo usuário:

C#

```
var Shipcity;  
ShipCity = Request.form ("ShipCity");  
var sql = "select * from OrdersTable where ShipCity = '" + ShipCity + "'";
```

O usuário é solicitado a inserir o nome de uma cidade. Se ele inserir Redmond, a consulta criada pelo script terá a seguinte aparência:

SQL

```
SELECT * FROM OrdersTable WHERE ShipCity = 'Redmond'
```

No entanto, suponha que o usuário insira o seguinte:

SQL

```
Redmond'; drop table OrdersTable--
```

Nesse caso, a seguinte consulta é gerada pelo script:

SQL

```
SELECT * FROM OrdersTable WHERE ShipCity = 'Redmond';drop table OrdersTable--'
```

O ponto-e-vírgula (;) denota o término de uma consulta e o início de outra. O hífen duplo (--) indica que o restante da linha atual é um comentário e deve ser ignorado. Se o código modificado estiver sintaticamente correto, será executado pelo servidor. Quando o SQL Server processar essa instrução, o SQL Server selecionará primeiro todos os registros em OrdersTable onde ShipCity é Redmond. Em seguida, o SQL Server descartará OrdersTable.

Prof. Roberto de Castro

Contanto que o código SQL injetado esteja sintaticamente correto, a violação não poderá ser detectada programaticamente. Portanto, é necessário validar todas as entradas de usuário e verificar com cuidado o código que executa comandos SQL construídos no servidor que você está usando.

Após as devidas considerações apresentadas acima, voltamos ao nosso projeto...

O projeto implementará a rotina de “Movimento Estoque” (entrada/saída) na tabela de “Produtos”.

Movimento de Produtos

ENTRADA E SAÍDA DE PRODUTOS EM ESTOQUE

Código do Produto:

Resultado da Pesquisa

Descrição do Produto:

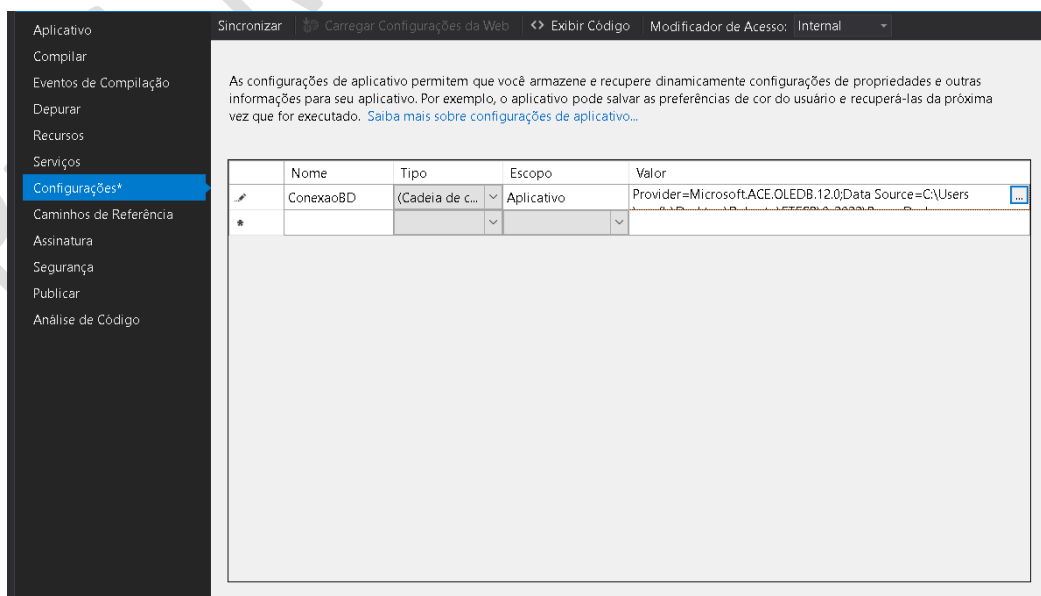
Quantidade Estoque: Quantidade Máxima:

Movimento de Produtos

☒ Entrada ☐ Saída Quantidade:

Após o “desenho” do formulário acima, faremos a criação da “string de conexão”.

Clique em Projeto → Propriedades (ou Project → Properties) → Configurações (Settings) → e vamos adicionar a variável de “Conexão String”. **O nome da variável será “ConexaoBD”:**



E na sequência a programação completa:

```
using System;
using System.Data;
using System.Data.OleDb; //Esta linha foi inserida
using System.Windows.Forms;

namespace Exe039A_BD005_MovtoProd
{
    public partial class FrmMovimento : Form
    {
        //Declaração das variáveis globais:
        //define a string de conexao com provedor caminho e nome do banco de dados
        //OBS: Esta string de conexão foi criada clicando-se em:
        //Projeto --> Propriedades --> Configurações
        string strConexao = Properties.Settings.Default.ConexaoBD;

        string strSql = "";

        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 codigo do produto foi digitado e se é numérico!!

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

            //cria a conexão com o banco de dados
            OleDbConnection conexao = new OleDbConnection(strConexao);

            //define a instrução SQL
            strSql = "SELECT Produtos.CodProd, Produtos.Descricao, " +
                " Produtos.QtdeMaxima, Produtos.QtdeEstoque " +
                "FROM Produtos " +
                "Where Produtos.CodProd = " + varCodProduto;

            //exibe a instrução SQL para testes iniciais, para conferência do comando....
            // MessageBox.Show(strSql);

            //cria o objeto command para executar a instrução sql
            OleDbCommand comando = new OleDbCommand(strSql, conexao);

            //abre a conexao
            conexao.Open();

            //define o tipo do comando
            comando.CommandType = CommandType.Text;
        }
    }
}
```

```
//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
//armanezados
//possam ser manipulados
myDataReader.Read();

//Se o comando de leitura retornou com algum registro, exibe os dados, caso
//contrário, exibe mensagem
//de alerta
if (myDataReader.HasRows)
{
    LblDescricao.Text = myDataReader["Descricao"].ToString();
    LblEstoque.Text = myDataReader["QtdeEstoque"].ToString();
    LblMaxima.Text = myDataReader["QtdeMaxima"].ToString();
}
else
{
    MessageBox.Show("Produto não localizado!!", "ATENÇÃO");
    TxtCodProduto.Text = "";
    TxtCodProduto.Focus();
}

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

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;
    }

    //rotina para atualizar o estoque na base de dados

    //cria a conexão com o banco de dados. A variável strConexao está declarada na
    //área de
    //variáveis globais - no início desta programação
    OleDbConnection conexao = new OleDbConnection(strConexao);
```

```
//define a instrução SQL
strSql = "UPDATE Produtos set QtdeEstoque=" + estoqueAtualizado + " where
        CodProd= " + TxtCodProduto.Text;

//Instrução apenas para consulta ao comando SQL
//MessageBox.Show(strSql);

//Cria o comando que inicia a query
OleDbCommand cmdQry = new OleDbCommand(strSql, conexao);

// abre o banco
conexao.Open();

// executa a query
cmdQry.ExecuteNonQuery();

//
MessageBox.Show("Dados Salvos com sucesso.");

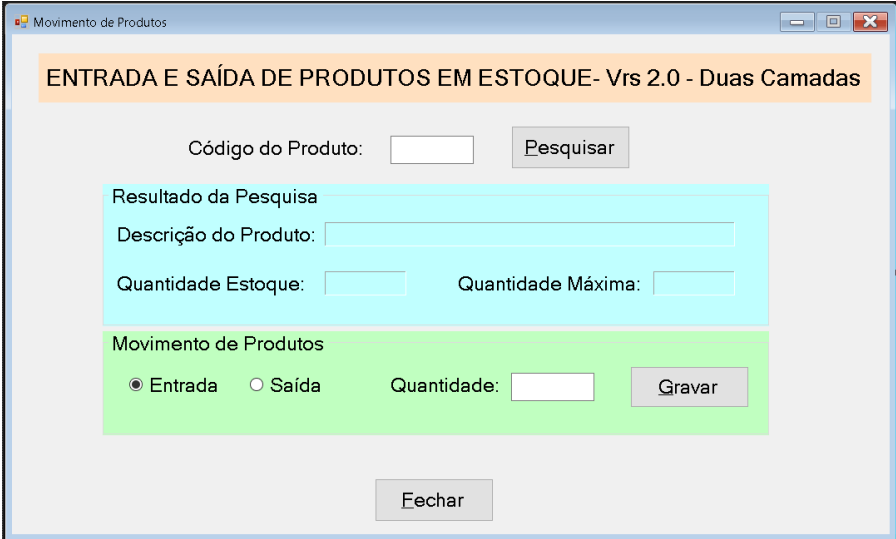
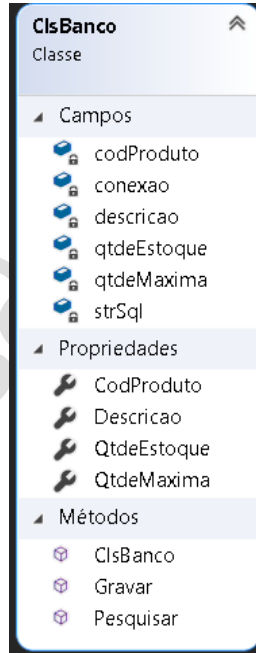
//fecha a conexao
conexao.Close();

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

Prof. Roberto de Castro

*** PROPOSTA DE ATIVIDADE ***

Converter o projeto acima para uma solução 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!

Prof. Roberto de Castro