**BIBLIOTECA DE CÓDIGO MAIS UTILIZADO**

1. **Codigo que permite o Visual Studio Reconhecer códigos SQL**

using System.Data.SqlClient;

1. **Mudanças de Propriedades para melhorarem a execução dos formulários**

* Para manter o formulário aberto a uma dimensão pré definida utiliza a opção **AutoSizeMode**

**Exemplo**: AutoSizeMode = GrowAndShrink

* Para manter um Botão activo utiliza – se a **opção AccetButton**

**Instrução**: AccetButton = Selecionar o botão que pretende que seja actvo.

* Para manter o formulário maximizado ou minimizado ( Abrir formulário ecrã inteiro), utiliza – se a opção WindowState = (Normal, Minimized ou Maximized).

1. **Para modificar todos ícones do formulário na caixa de propriedades clique na opção.**

FormBorderStyle = None

1. **Inserir progressbar no formulário**
2. Inserir um **Timer** no formulário
3. Inserir **progressbar** no formulário

No **Timer** inserir seguinte código

private void timer1\_Tick(object sender, EventArgs e)

{

if (progressBar1.Value < 100)

{

progressBar1.Value = progressBar1.Value + 2;

}

else

{

timer1.Enabled = false;

Login Y= new Login();

Y.Show();

this.Visible = false;

}

}

}

}

Depois de inserir o código, para executar o progressbar de forma automática click em Timer na propriedade na opção **Enabled = True**

1. **Chamar um formulário (Form2 ou Form3)**
2. **Método**

Form2 Outroform = new Form2();

Outroform.Show();

this.Visible = false;

1. **Método**

Form2 f2 = new Form2();

f2.Show();

This.Hide(); retira esta linha de codigo para não desaparecer...

1. **Método**

login f2 = new login();

f2.Show();

this.Visible = false;

**OBS:** Não existe diferença entre os três métodos de chamar um formulário acima referenciado, apenas a diferença está no 3 Método, se usamos a sintaxe **f2.ShowDialog ();** o botão executado chama o formulário seguinte mais não fecha o formulário actual, o que não acontece no 1º e 2º método chama e fecha o formulário actual.

1. **Código para inserir data e hora do (PC) no formulário.**

Inserir um **Timer** no formulário

No **Timer** inserir seguinte código abaixo,

label1.Text = DateTime.Now.ToShortDateString();

label2.Text = DateTime.Now.ToShortTimeString();

Depois de inserir o código, para a hora e data de forma automatica, click em Timer em propriedades na opção **Enabled = True**

1. **Retroceder no form1**

This.Close ();

1. **Código botão Voltar**

Close();

Form1 log = new Form1();

log.Show();

1. **Exibindo o resultado na textbox3**

txtmedia.Text = Media.ToString();

1. **Limpar uma Textbox**
2. **Método**

txtNomeProduto.Text = "";

txtValorUnitario.Text = "";

txtQuantidade.Text = "";

1. **Método**

TextBox1.Text = String.Empty;

1. **Método**

Txtnota1.Clear ();

1. **Focus**

txtnota1.Focus();

**OBS**: O Código **txtnota1.Focus ();** significa que mesmo depois de executar o programa o cursor vai sempre parar na textbox1

Onde: txtnota1 é o nome dado na textbox do programa em questão e Focus (); é o Código Fonte C#.

1. **Código para sair ou fechar um formulário. (terminar a sessão)**

~

Para inserir botão **Sair** do formulario, basta dar duplo click sobre o mesmo e digitar o seguinte código:

1. **Método**

private void btsair\_Click(object sender, EventArgs e)

{

Application.Exit();

}

1. **Método**

private void btsair\_Click(object sender, EventArgs e)

{

Environment.Exit (0);

}

1. **Método**

if (MessageBox.Show("Deseja Sair da Aplicação?", "Fechar", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

this.Close();

1. **Inserir Mensagebox no formulário.**

O código é inserido no botão **Entrar**

1. **Método**

MessageBox.Show ("Novo Funcionário cadastrado com sucesso");

1. **Método**

MessageBox.Show(“Cadastramento feito com sucesso!”,”Viso”,MessageBoxButtons.Ok,MessageBoxIcon.Information);

1. **Codigo Botão Eliminar**

Para Eliminar Produto na base de dados basta adicionar botão **Eliminar Produto** no form e dar duplo click sobre o mesmo e digitar o seguinte código:

**Codificação**

private void bteliminar\_Click(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection(Y);

string EliminarProdutos = "delete Produtos where CodProduto=@CodProduto";

try

{

SqlCommand cmd = new SqlCommand(EliminarProdutos, conn);

cmd.Parameters.AddWithValue("@CodProduto", this.textBox1.Text);

textBox1.Text = "";

conn.Open();

cmd.ExecuteNonQuery();

// MessageBox.Show("Dados Eliminado com Sucesso!");

if (MessageBox.Show("Deseja realmente Eliminar este Dado?", "Pembelesoft2020",

MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

textBox1.Text = "";

}

conn.Close();

}

catch (Exception)

{

MessageBox.Show("Erro ao Eliminar Dados");

}

}

}

}

**VALIDAÇÃO DE CAMPOS NO FORMULÁRIO**

**Exitem basicamente três tipos de validação**

1. Campos Obrigatório (Vázio)
2. Tamanho do Campo
3. Tipo de Dados

**Campos Obrigatório (Vázio) “um usuário”**

if ((codigoTextBox1.Text == ""))

{

MessageBox.Show ("Preencha os Campos

Vazios","Erro",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

Else

{

MessageBox.Show ("Cadastramento feito com sucesso!", "sucesso", MessageBoxButtons.OK,

MessageBoxIcon.Information);

}

}

**Campos Obrigatório (Vázio) “dois usuárioa”**

**Validar dois campo como obrigatório**

private void bt\_entrar\_Click(object sender, EventArgs e)

{

if

((txt\_usuario.Text == "") || (txt\_senha.Text == ""))

{

MessageBox.Show("erro", "error", MessageBoxButtons.OK);

}

else

MessageBox.Show ("novo funcionário cadastrado com sucesso");

Form2 f2 = new Form2();

F2.ShowDialog();

This.Visible = false;

}

}

}

**Validaçao por Tamanho do Campo**

Para **validar** um campo definindo **Tamanho de carácter na Textbox, faz se o seguinte:** Vá até propriedade do C# na opção **MaxLengt.**

ExemploMaxLength = 12 significa a Textbox só receberá 12 caracteres no máximo.

**Validaçao pos tipo de dados**

Normalmente é definida a aparti dos tipos de dados no acto da programação.