

Label

Name: lblNome

Text: Informe o seu nome

TextBox

Name: txtNome

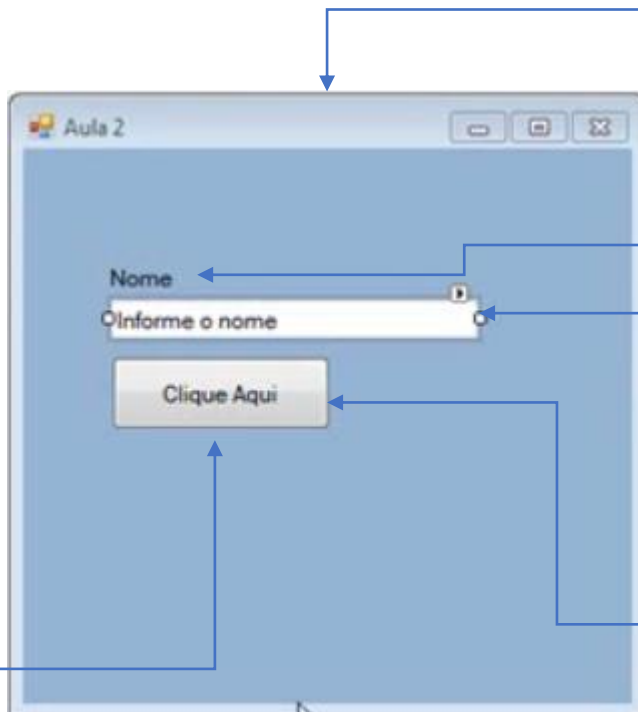
Text: Digite o seu nome

Button

Name: btnExibir

Text: Clique aqui

```
private void btnExibir_Click(object sender, EventArgs e)
{
    MessageBox.Show(txtNome.Text);
}
```



Formulário

Name: frmPrincipal
Text: Aula

Label

Name: lblNome
Text: Nome

TextBox

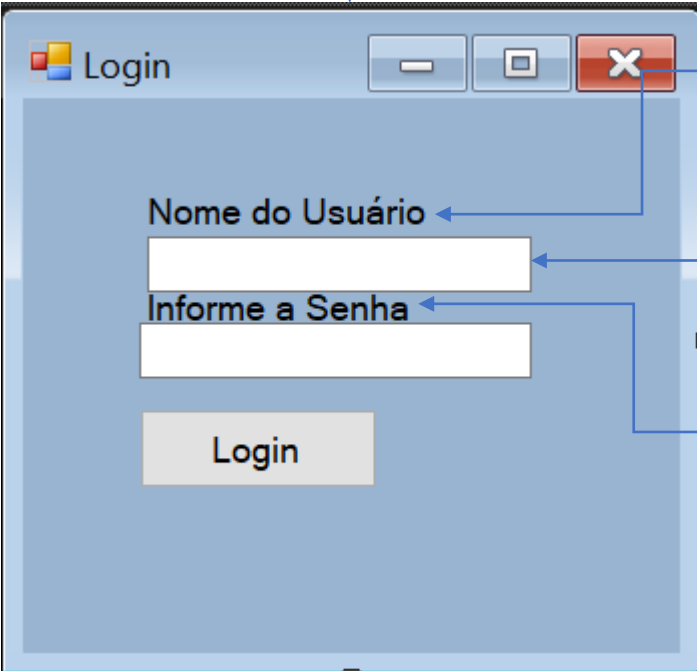
Name: txtNome
Text: Informe o nome

Button

Name: btnExibir
Text: Cklique aqui

```
private void btnExibir_Click(object sender, EventArgs e)
{
    MessageBox.Show("Ola" + txtNome.Text);
}
```

Formulário
Name: frmPrincipal
Text: Login



Label
Name: lblNome
Text: Informe o seu nome

TextBox
Name: txtNome
Text:

Label
Name: lblSenha
Text: Informe a Senha

TextBox
Name: txtSenha
Text:

Button
Name: btnLogin
Text: Login

```
private void btnLogin_Click(object sender, EventArgs e)
{
    if(txtNome.Text == "rodox" && txtSenha.Text == "nunsey")
    {
        MessageBox.Show("Querido " + txtNome.Text + " Bem-vindo ao Sistema!!!");
    }
    else
    {
        MessageBox.Show("USUÁRIO OU SENHA INVÁLIDO");
    }
}
```

1. Crie um formulário com as mesmas características do anterior

Formulário
Name: frmPrincipal
Text: Login

Nome do Usuário

Informe a Senha

Login

Formulario
Name: frmSistema
Text: Sistema Principal - Tela Inicial
BackColor: Yellow

Bem-Vindo

Label
Name: lblBemVindo
Text: Bem-Vindo
font: Ravie; 28,2pt

```
private void btnLogin_Click(object sender, EventArgs e)
{
    if (txtUsuario.Text == "rodox" && txtSenha.Text == "nunsey")
    {
        frmSistema frm2 = new frmSistema();
        frm2.ShowDialog();
    }
    else
    {
        MessageBox.Show("USUÁRIO OU SENHA INVÁLIDO");
    }
}
```

1 Na tela de Bem-Vindo crie a nova Label

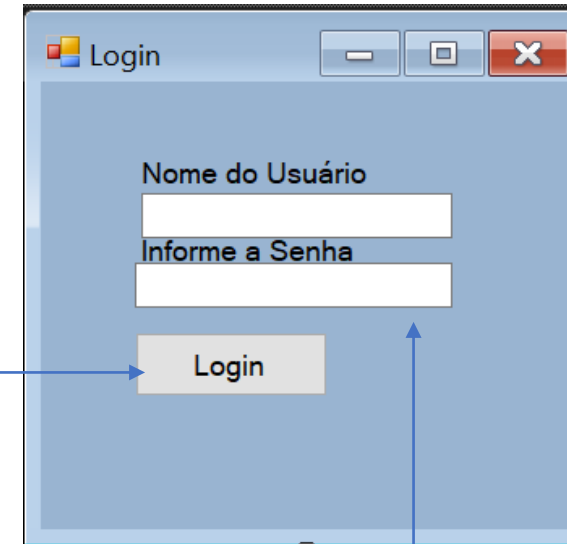


Label
Name: lblInfo
Text:
font: Ravie; 16

3 No Evento Clique do botão login altere

```
public static string NomeDoUsuario = "";  
private void btnExibir_Click(object sender, EventArgs e)  
{  
    if (txtNome.Text == "rodox" && txtSenha.Text == "numsey")  
    {  
        frmSistema frm2 = new frmSistema();  
        NomeDoUsuario = txtNome.Text;  
        frm2.ShowDialog();  
    }  
    else  
    {  
        MessageBox.Show("Usuario/Senha Inválido(s)");  
    }  
}
```

2 Na tela de login



Em **txtSenha** procure a propriedade **PasswordChar** e inclua um Asterisco

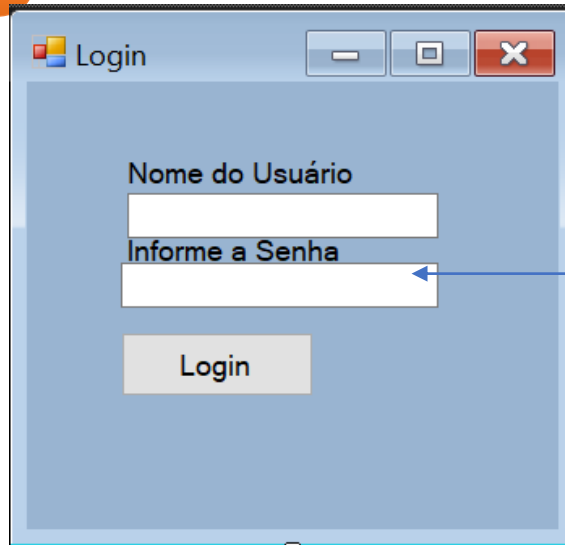
PasswordChar *

4 Duplo clique em frmSistema e acrescente

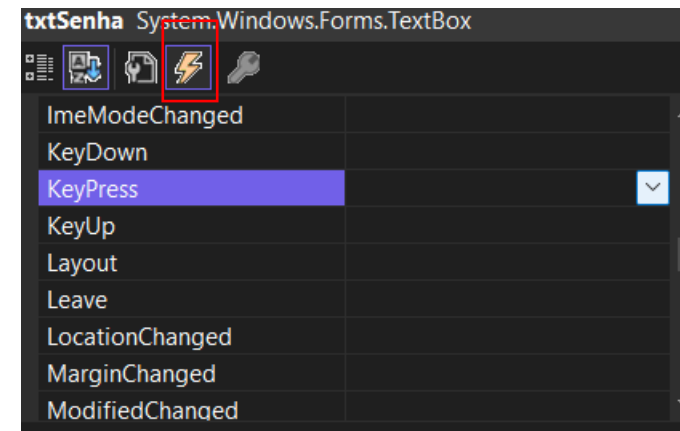
```
public partial class frmSistema : Form  
{  
    public frmSistema()  
    {  
        InitializeComponent();  
    }  
  
    private void frmSistema_Load(object sender, EventArgs e)  
    {  
        lblInfo.Text = frmPrincipal.NomeDoUsuario;  
    }  
}
```

5

Em txtSenha selecione o evento KeyPress



Evento KeyPress



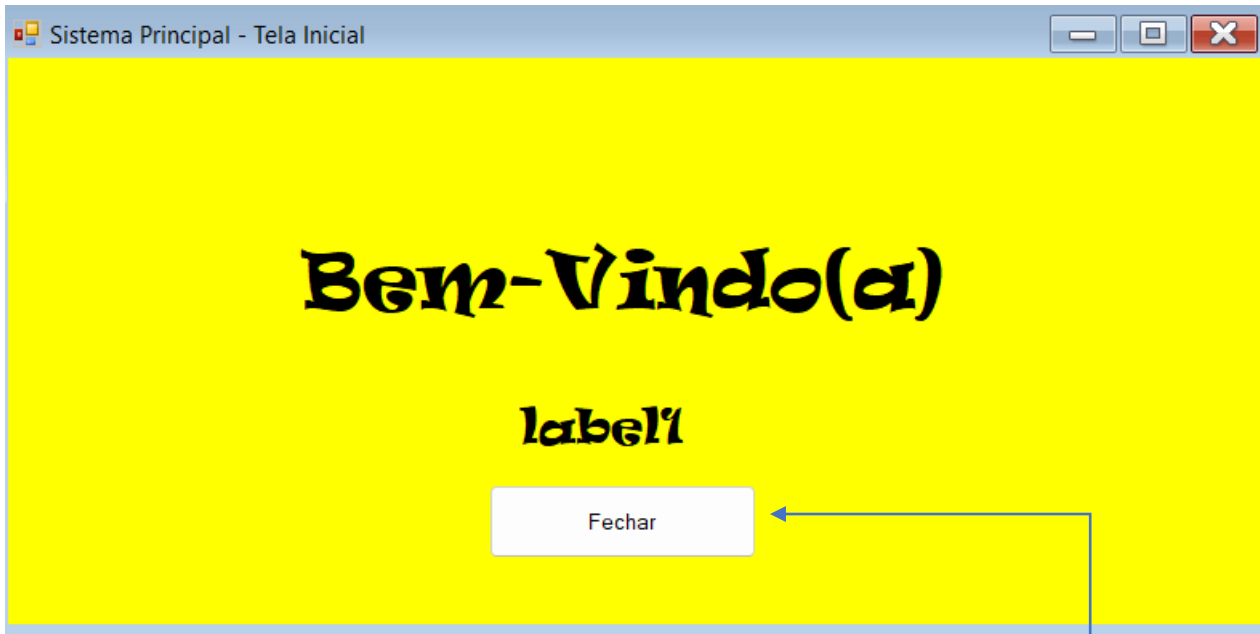
```
private void txtSenha_KeyPress(object sender, KeyPressEventArgs e)
{
    int asc = (int)e.KeyChar;
    if (!char.IsDigit(e.KeyChar) && asc != 08 && asc != 127)
    {
        e.Handled = true;
        MessageBox.Show("Este campo apenas aceita numeros!");
        txtSenha.Text = "";
        txtSenha.Focus();
    }
}
```

6

Com o evento KeyPress digite

7

Adione um Botão

**Button**

Name: btnFechar

Text: Fechar

8

Duplo clique e adicione

```
private void btnFechar_Click(object sender, EventArgs e)
{
    this.Close();
}
```

1

Vamos criar o botão Fechar na tela de login

Informe o Nome do Usuário

Informe a Senha

Email

Clique Aqui Fechar

Name: lblEmail

Text: Email

Name: txtEmail

Text:

Name: btnFechar

Text: Fechar

```
private void btnFechar_Click(object sender, EventArgs e)
{
    this.Close();
}
```

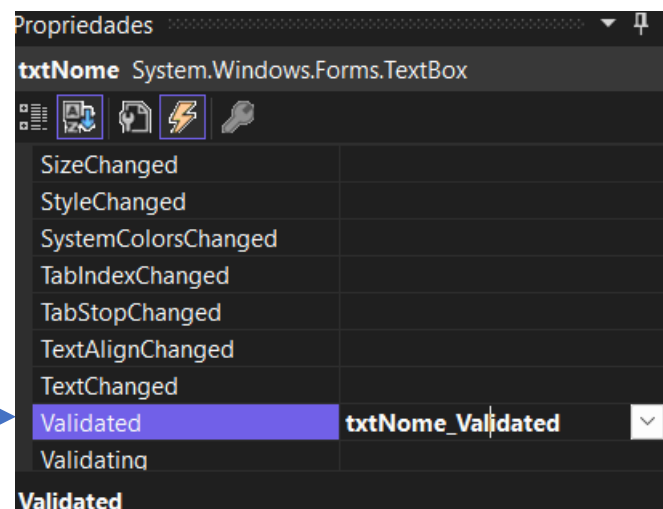
2

Inserindo Validação – Clique em txtUsuario e em eventos selecione a opção Validated

Informe o Nome do Usuário

Informe a Senha

Clique Aqui Fechar



3

Vamos criar uma condição que se estiver vazio o MessageBox aparece

```
private void txtNome_Validated(object sender, EventArgs e)
{
    if (txtNome.Text == string.Empty)
    {
        MessageBox.Show("Informe o nome do usuário");
    }
}
```

4

Em btnFechar acrescente a autovalidadte

```
private void btnFechar_Click(object sender, EventArgs e)
{
    AutoValidate = AutoValidate.Disable;
    this.Close();
}
```


1 Na caixa de ferramentas localize a opção ErrorProvider

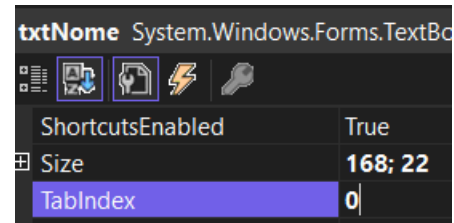
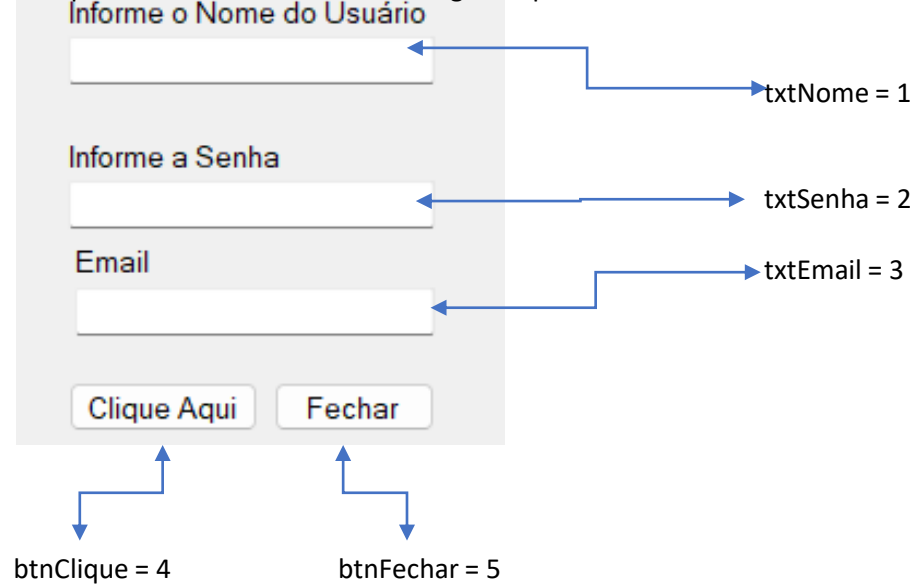
2 Em txtNome_Validated substitua por:

```
private void txtNome_Validated(object sender, EventArgs e)
{
    if (txtNome.Text == string.Empty)
    {
        //MessageBox.Show("Informe o nome do usuário");
        errorProvider1.SetError(txtNome, "Informe o nome do usuário")
    }
    else
    {
        errorProvider1.SetError(txtNome, "");
    }
}
```

Utilizando KeyPreview para que o enter possa passar para o próximo elemento com o TAB

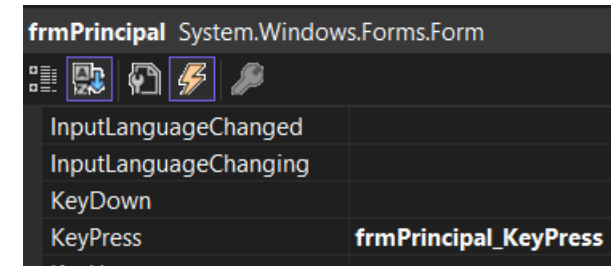
1

Clique em cada textBox abaixo e organize pelo TabIndex.



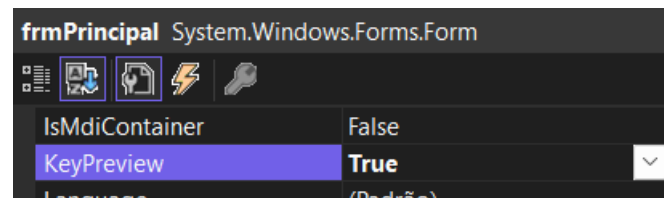
3

Clique em frmPrincipal em **EVENTOS** cliquem em KeyPress



2

Clique em frmPrincipal clique em **Propriedades** e altere para KeyPreview para True



4

Digite

```
private void frmPrincipal_KeyPress(object sender, KeyPressEventArgs e)
{
    if(e.KeyChar == Convert.ToChar(13))
    {
        e.Handled= true;
        SendKeys.Send("{TAB}");
    }
}
```

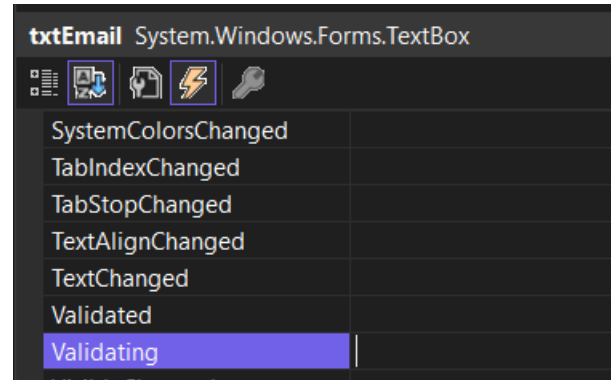
Informe o Nome do Usuário

Informe a Senha

Email

1

Clique no txtEmail em EVENTOS e clique em Validating

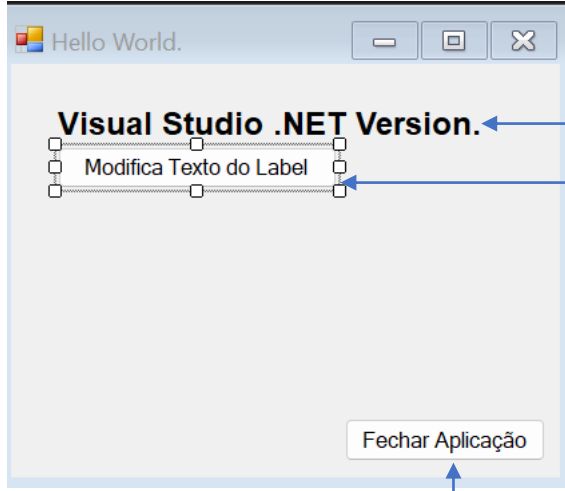


2

Digite

```
private void txtEmail_Validating(object sender, CancelEventArgs e)
{
    string strRegex = "^[a-zA-Z0-9_\\-\\.]+@((\\[[0-9]{1,3}" +
        "\\.[0-9]{1,3}\\.[0-9]{1,3}\\.)|(([a-zA-Z0-9\\-]" +
        ".)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\\[\\])?$";
    Regex re = new Regex(strRegex);
    if ((re.IsMatch(txtEmail.Text)))
    {
        txtEmail.BackColor = Color.Yellow;
    }
    else
    {
        txtEmail.BackColor = Color.WhiteSmoke;
        MessageBox.Show("Email invalido");
        txtEmail.Text = "";
        e.Cancel = true;
    }
}
```

Criando projeto de Formulários,
componentes e eventos



Name, do *Label*, para **Lbl_Titulo**.

Text para **Visual Studio .NET Version**.

Font, de **Lbl_Titulo**, para **Arial; 12pt; style=Bold**.

- Name: Btn_ModificaLabel**
- Location: 34; 52**
- Size: 209; 23**
- Text: Modifica Texto do Label**

```
private void Btn_ModificaLabel_Click(object sender, EventArgs e)
{
    Lbl_Titulo.Text = "Label Modificado";
}
```

Form, mude o valor da propriedade **Name** para **Frm_HelloWorld**.

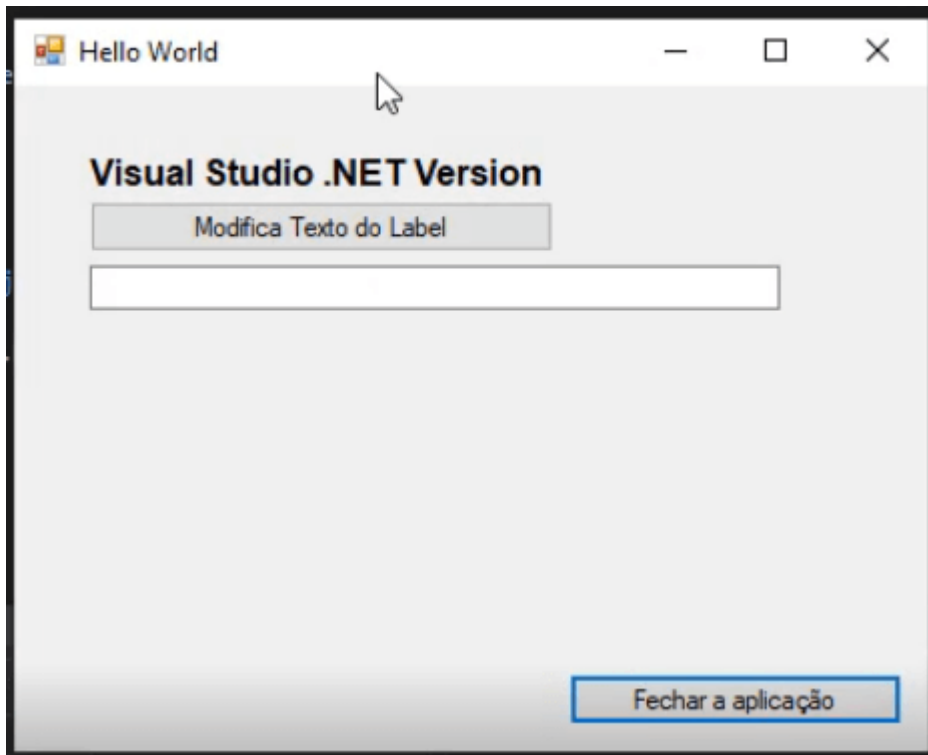
Button, para **Btn_Sair**.

Text, de **Btn_Sair**, para **Fechar a aplicação**.

evento **Click**, efetue um duplo clique sobre ela:

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

StartPosition e mude para **CenterScreen**.



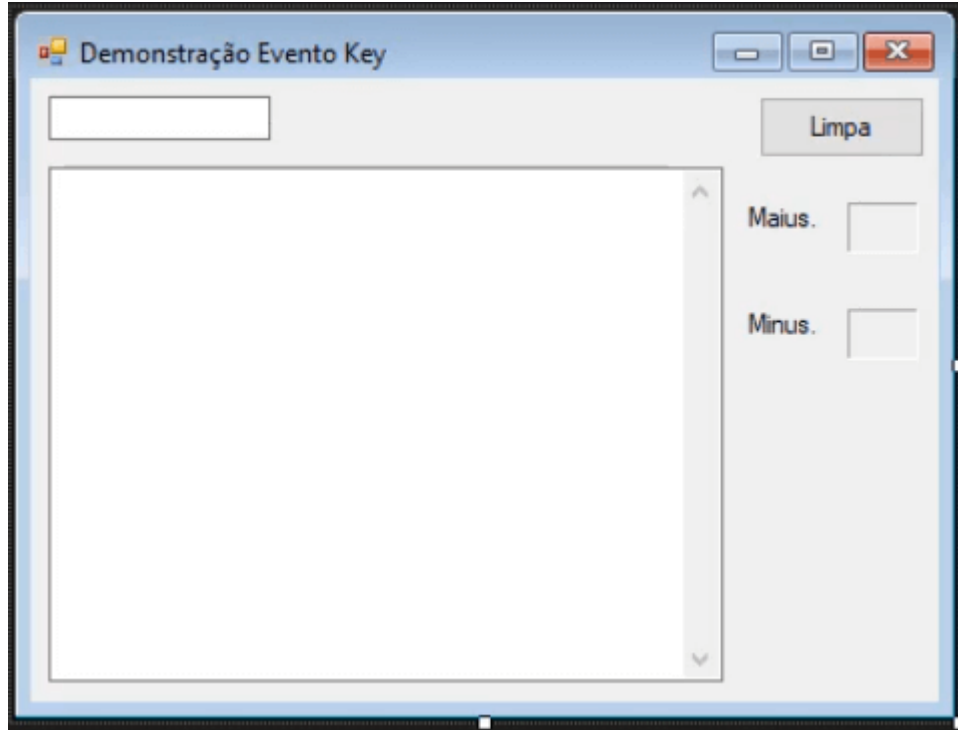
Volte ao projeto e arraste um componente **TextBox** para o formulário **Frm_HelloWorld**.

Inclua as seguintes propriedades:

- Name:** **Txt_ConteudoLabel**
- Location:** **34; 81**
- Size:** **311; 20**

```
private void Btn_ModificaLabel_Click(object sender, EventArgs e)
{
    Lbl_Titulo.Text = Txt_ConteudoLabel.Text;
}
```

Frm_DemostracaoKey

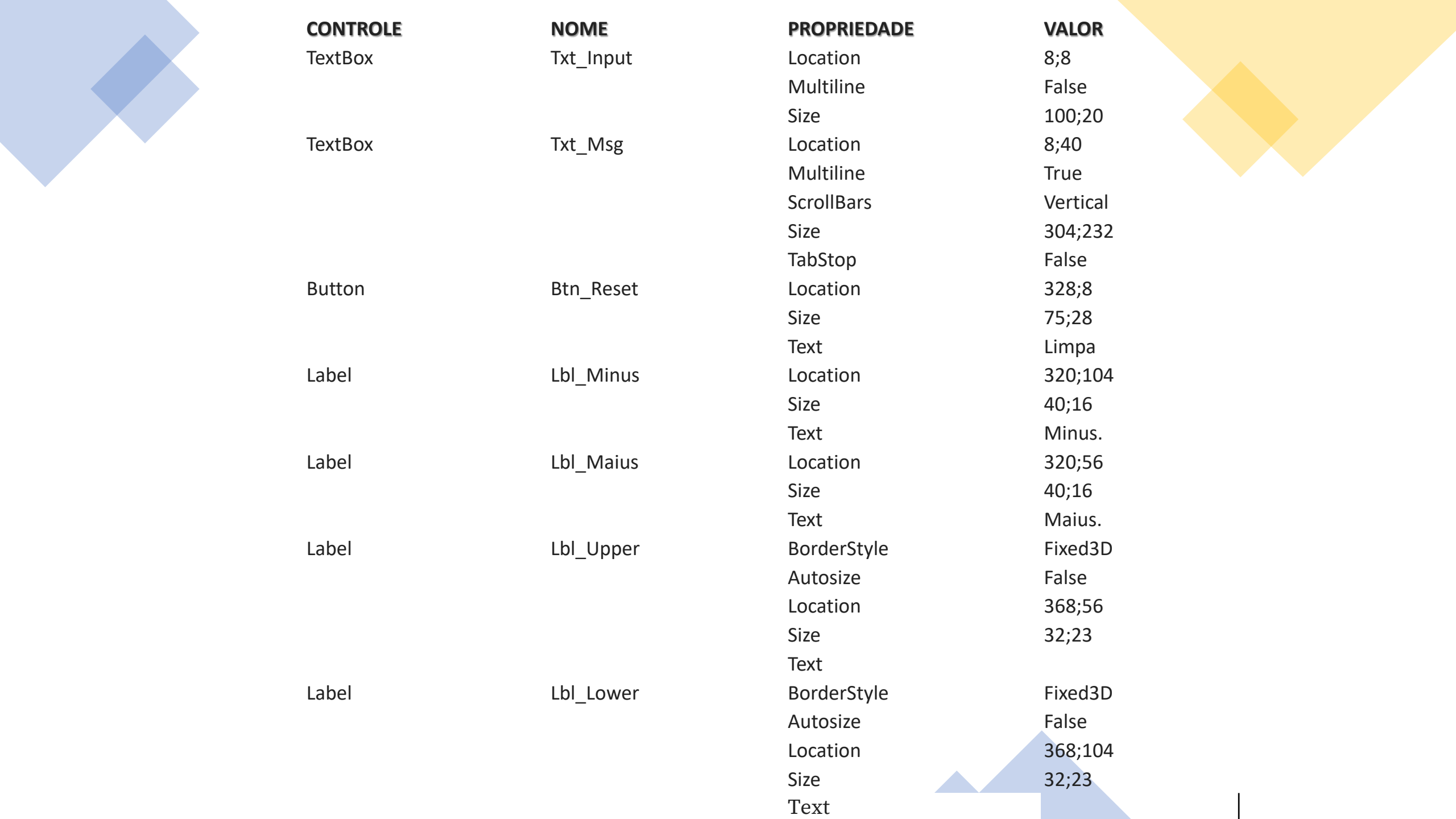


Formulario:

Name: Frm_DemostracaoKey.

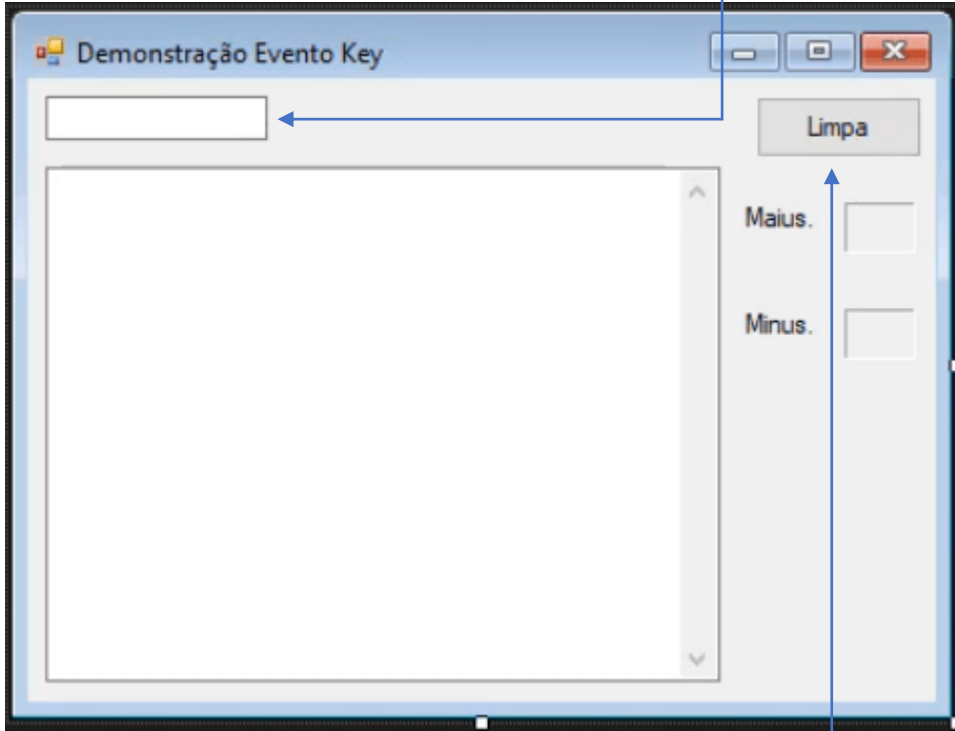
Size: 425; 320

Text: Demonstração Evento Key.



CONTROLE	NOME	PROPRIEDADE	VALOR
TextBox	Txt_Input	Location	8;8
		Multiline	False
		Size	100;20
TextBox	Txt_Msg	Location	8;40
		Multiline	True
		ScrollBars	Vertical
		Size	304;232
Button	Btn_Reset	TabStop	False
		Location	328;8
		Size	75;28
Label	Lbl_Minus	Text	Limpa
		Location	320;104
		Size	40;16
Label	Lbl_Maius	Text	Minus.
		Location	320;56
		Size	40;16
Label	Lbl_Upper	Text	Maius.
		BorderStyle	Fixed3D
		Autosize	False
		Location	368;56
Label	Lbl_Lower	Size	32;23
		Text	
		BorderStyle	Fixed3D
		Autosize	False
		Location	368;104
		Size	32;23
		Text	

Nos eventos de **Txt_Input**, na área ao lado do evento **KeyDown**, efetue um duplo clique sobre ela. Digite o seguinte código:



```
private void Txt_Input_KeyDown(object sender, KeyEventArgs e)
{
    Txt_Msg.AppendText("\r\n" + "Pressionei uma tecla: " + e.KeyCode + "\r\n");
    Txt_Msg.AppendText("\t" + "Código da tecla: " + ((int)e.KeyCode) + "\r\n");
    Txt_Msg.AppendText("\t" + "Nome da tecla: " + e.KeyData + "\r\n");
    Lbl_Lower.Text = e.KeyCode.ToString().ToLower();
    Lbl_Upper.Text = e.KeyCode.ToString().ToUpper();
}
```

No evento **Click**, de **Btn_Reset**, digite o seguinte código:

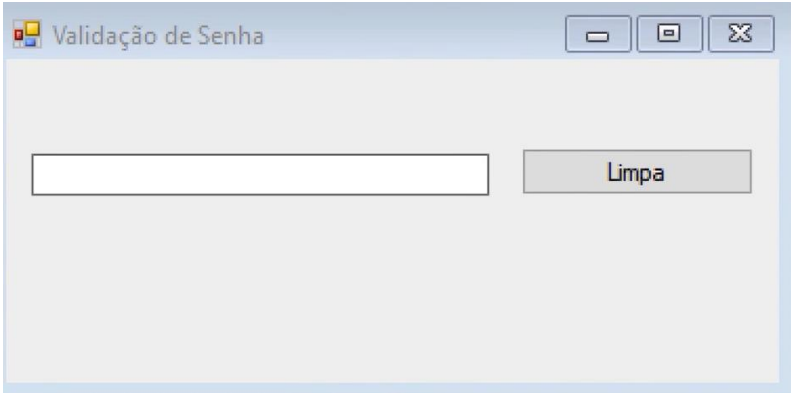
```
private void Btn_Reset_Click(object sender, EventArgs e)
{
    Txt_Msg.Text = ""; Txt_Input.Text = "";
    Lbl_Upper.Text = ""; Lbl_Lower.Text = "";
}
```

Frm_ValidaSenha

CONTROLE	NOME	PROPRIEDADE	VALOR
TextBox	Txt_Senha	Location	12;45
		Size	218; 20
Label	Lbl_Resultado	Location	12; 90
		Font	Arial; 14,25pt; style=Bold
		Text	
Button	Btn_Reset	Location	245; 42
		Size	111; 23
		Text	Limpa
Button	Btn_Valida	Location	245; 71
		Size	111; 23
		Text	Valída

Adicione ao projeto um outro formulário, com o nome **Frm_ValidaSenha**. Altere os valores das propriedades **Size**, **StartPosition** e **Text** deste novo formulário para **384; 193**, **CenterScreen** e **Validação da Senha**, respectivamente.

arquivo **ChecaForcaSenha.cs**. Em seguida, abra-o com um editor de texto.



CONTROLE	NOME	PROPRIEDADE	VALOR
Form	Frm_ValidaSenha	Size	384; 193
		Text	Validação de Senhas
		StartPosition	CenterScreen
TextBox	Txt_Senha	Location	12;45
		Multiline	False
		Size	218; 20
		Text	<Blank>
Label	Lbl_Resultado	Location	12; 90
		Font	Arial; 14,25pt; style=Bold
		Text	<Blank>
Button	Btn_Reset	Location	245; 42
		Size	111; 23
		Text	Limpa

1 No código do formulário **Frm_ValidaSenha**, abaixo da sua classe, inclua o código contido em **ChecaForcaSenha.cs**:

2 Alguns erros apareceram. Para corrigi-los, inclua a referência a uma DLL do .NET Framework para manipular expressões regulares:

```
using  
System.Text.RegularExpressions;
```

3 Inicialize uma variável, válida para toda a classe, para controlar a ação do botão:

```
public partial class Frm_ValidaSenha : Form  
{  
    bool VerSenhaTxt = false;  
    1reference  
    public Frm_ValidaSenha()  
    {  
        InitializeComponent();  
    }  
}
```

4

No código do evento **Click**, do botão **Btn_Reset**, inclua:

```
private void Btn_Reset_Click(object sender, EventArgs e)  
{  
    Txt_Senha.Text = "";  
    Lbl_Resultado.Text = "";  
}
```

5

No código do evento **KeyDown**, de **Txt_Senha**, inclua:

```
private void Txt_Senha_KeyDown(object sender, KeyEventArgs e)
{
    ChecaForcaSenha verifica = new ChecaForcaSenha();
    ChecaForcaSenha.ForcaDaSenha forca;
    forca = verifica.GetForcaDaSenha(Txt_Senha.Text);
    Lbl_Resultado.Text = forca.ToString();

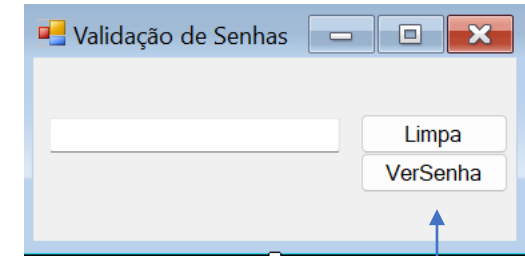
    if (Lbl_Resultado.Text == "Inaceitavel" | Lbl_Resultado.Text == "Fraca")
    {
        Lbl_Resultado.ForeColor = Color.Red;
    }
    if (Lbl_Resultado.Text == "Aceitavel")
    {
        Lbl_Resultado.ForeColor = Color.Blue;
    }
    if (Lbl_Resultado.Text == "Forte" | Lbl_Resultado.Text == "Segura")
    {
        Lbl_Resultado.ForeColor = Color.Green;
    }
}
```

6

Vamos adicionar botão

Name: Btn_VerSenha

Text: Ver senha



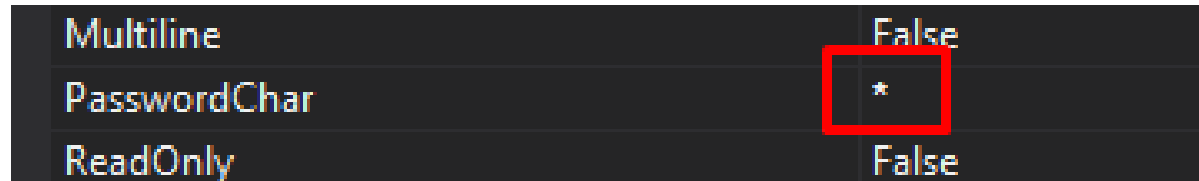
7

No código do evento **Click**, do botão **Btn_VerSenha**, inclua

```
private void Btn_VerSenha_Click(object sender, EventArgs e)
{
    if (VerSenhaTxt == false)
    {
        Txt_Senha.PasswordChar = '\0';
        VerSenhaTxt = true;
        Btn_VerSenha.Text = "Esconder Senha";
    }
    else
    {
        Txt_Senha.PasswordChar = '*';
        VerSenhaTxt = false;
        Btn_VerSenha.Text = "Ver Senha";
    }
}
```

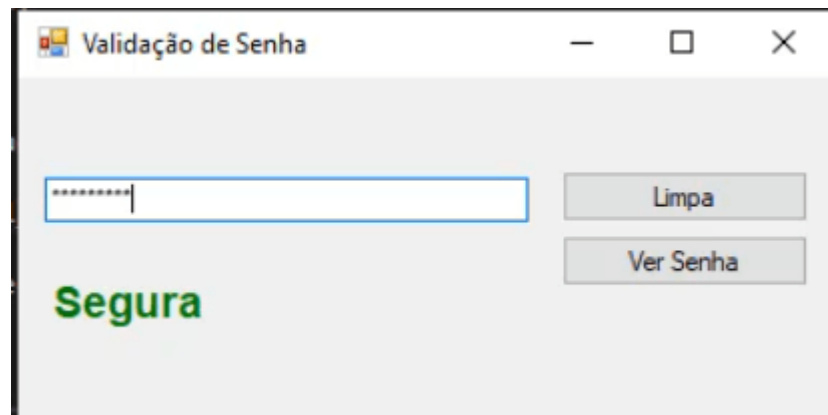
8

Vá na propriedade **PasswordChar**, de **Txt_Senha** e inclua o valor * (asterico):



9

Salve e execute a aplicação. Defina uma senha e você verá que a classificação é obtida a cada tecla incluída na caixa de texto:



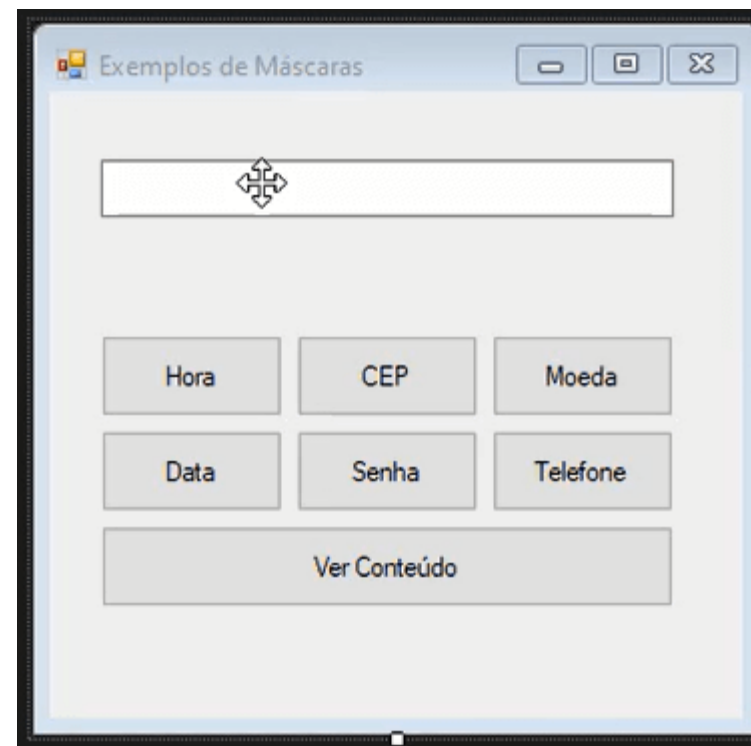
Frm_mascara

1

Adicione um novo formulário ao projeto, chamado **Frm_Mascara**. Altere os valores das propriedades **Size**, **StartPosition** e **Text** de ste novo formulário para **328; 322**, **CenterScreen** e **Exemplos de Máscaras**, respectivamente.

2

Adicione os seguintes componentes ao formulário **Frm_Mascara**, com as seguintes propriedades:



CONTROLE	NOME	PROPRIEDADE	VALOR
MaskedTextBox	Msk_TextBox	Location	23; 31
		Size	258; 26
		Font	Arial; 12pt; style=Bold
		Text	
Label	Lbl_Conteudo	Location	23; 246
		Font	Arial; 12pt; style=Bold
		Text	
Label	Lbl_MascaraAtiva	Location	23; 75
		Font	Arial; 12pt; style=Bold
		Text	
Button	Btn_Hora	Location	23; 110
		Size	82; 37
		Text	Hora
Button	Btn_CEP	Location	111; 110
		Size	82; 37
		Text	CEP

CONTROLE	NOME	PROPRIEDADE	VALOR
Button	Btn_Moeda	Location	199; 110
		Size	82; 37
		Text	Moeda
Button	Btn_Data	Location	23; 153
		Size	82; 37
		Text	Data
Button	Btn_Senha	Location	111; 153
		Size	82; 37
		Text	Senha
Button	Btn_Telefone	Location	199; 153
		Size	82; 37
		Text	Telefone
Button	Btn_VerConteudo	Location	23; 196
		Size	258; 37
		Text	Ver Conteúdo

3

Inclua o seguinte código associado ao evento **Click** de cada botão: