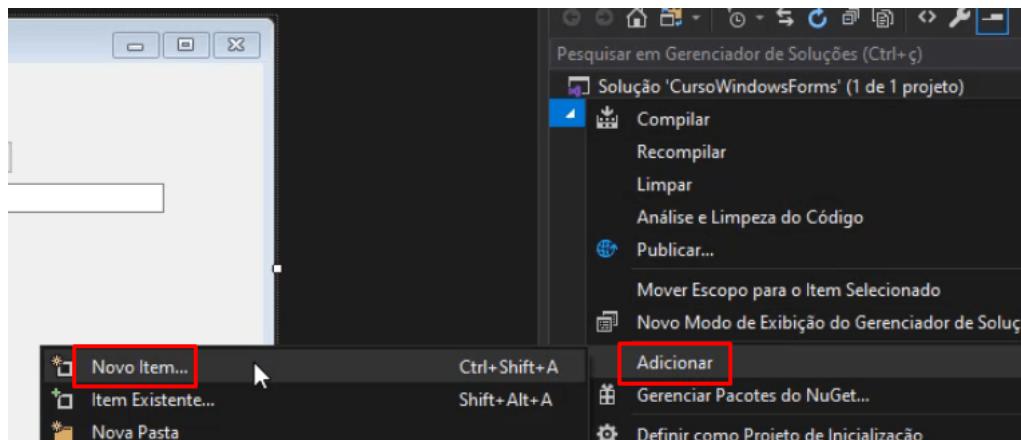


Faça como eu fiz

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

- 1) No **Gerenciador de Soluções**, clique com o botão direito do mouse sobre o seu projeto e selecione **Adicionar** --> **Novo Item**:

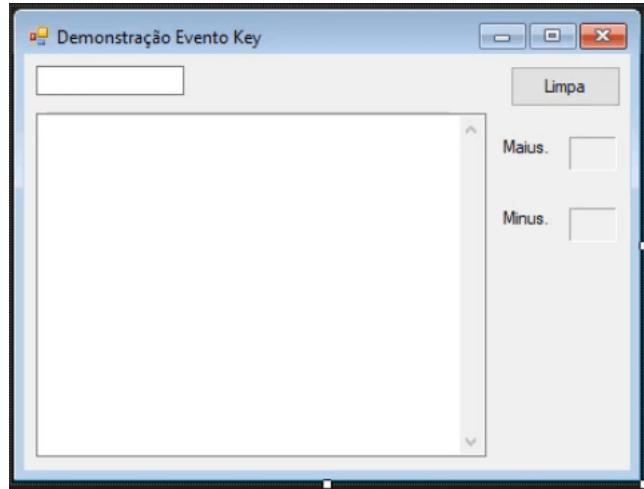


- 2) Escolha um item do tipo **Formulário (Windows Forms)**. No nome, digite **Frm_DemostracaoKey**. Altere os valores das propriedades **Size** e **Text** deste novo formulário para **425; 320** e **Demonstração Evento Key**, respectivamente.

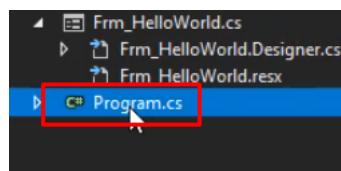
- 3) Inclua novos componentes e modifique as suas propriedades, conforme a tabela abaixo:

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	Location	328;8
		Size	75;28
		Text	Limpa
Label	Lbl_Minus	Location	320;104
		Size	40;16
		Text	Minus.
Label	Lbl_Maius	Location	320;56
		Size	40;16
		Text	Maius.
Label	Lbl_Upper	BorderStyle	Fixed3D
		Autosize	False
		Location	368;56
		Size	32;23
		Text	
Label	Lbl_Lower	BorderStyle	Fixed3D
		Autosize	False
		Location	368;104
		Size	32;23
		Text	

4) O formulário final será:



5) É necessário trocar o formulário padrão. Para isso, efetue um duplo clique sobre o programa **Program.cs**:



6) Modifique o código para:

```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Frm_DemostracaoKey());
}
```

7) Modifique o valor da propriedade **StartPosition**, do formulário **Frm_DemostracaoKey**, para **CenterScreen**.

8) 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();
}
```

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

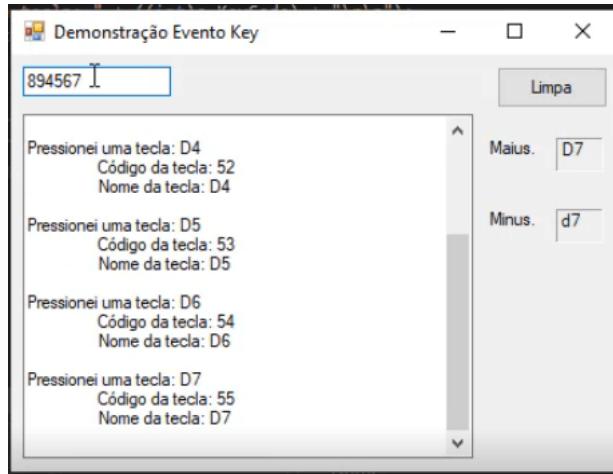
```
private void Btn_Reset_Click(object sender, EventArgs e)
{
    Txt_Msg.Text = "";
    Txt_Input.Text = "";
```

```

    Lbl_Upper.Text = "";
    Lbl_Lower.Text = "";
}

```

10) Salve e execute a aplicação. Quando você digita algo em **Txt_Input**, é escrito em **Txt_Msg** algumas características da tecla pressionada:



11) 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.

12) Caso você ainda não tenha feito o download, baixe [aqui](https://caelum-online-public.s3.amazonaws.com/1548-windows-forms-c-sharp-parte-1/03/windows-forms-parte-1-arquivos-aula-3.zip) (<https://caelum-online-public.s3.amazonaws.com/1548-windows-forms-c-sharp-parte-1/03/windows-forms-parte-1-arquivos-aula-3.zip>) o arquivo **ChecaForcaSenha.cs**. Em seguida, abra-o com um editor de texto.

13) Inclua novos componentes e modifique as suas propriedades, conforme a tabela abaixo:

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
Button	Btn_Reset	Text	
		Location	245; 42
		Size	111; 23
Button	Btn_Valida	Text	Limpa
		Location	245; 71
		Size	111; 23
		Text	Valida

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

```
public class ChecaForcaSenha
{
    public enum ForcaDaSenha
    {
        Inaceitavel,
        Fraca,
        Aceitavel,
        Forte,
        Segura
    }

    public int geraPontosSenha(string senha)
    {
        if (senha == null) return 0;
        int pontosPorTamanho = GetPontoPorTamanho(senha);
        int pontosPorMinusculas = GetPontoPorMinusculas(senha);
        int pontosPorMaiusculas = GetPontoPorMaiusculas(senha);
        int pontosPorDigitos = GetPontoPorDigitos(senha);
        int pontosPorSimbolos = GetPontoPorSimbolos(senha);
        int pontosPorRepeticao = GetPontoPorRepeticao(senha);
        return pontosPorTamanho + pontosPorMinusculas +
            pontosPorMaiusculas + pontosPorDigitos +
            pontosPorSimbolos - pontosPorRepeticao;
    }

    private int GetPontoPorTamanho(string senha)
    {
        return Math.Min(10, senha.Length) * 7;
    }

    private int GetPontoPorMinusculas(string senha)
    {
        int rawplacar = senha.Length - Regex.Replace(senha, "[a-z]", "").Length;
        return Math.Min(2, rawplacar) * 5;
    }

    private int GetPontoPorMaiusculas(string senha)
    {
        int rawplacar = senha.Length - Regex.Replace(senha, "[A-Z]", "").Length;
        return Math.Min(2, rawplacar) * 5;
    }

    private int GetPontoPorDigitos(string senha)
    {
        int rawplacar = senha.Length - Regex.Replace(senha, "[0-9]", "").Length;
        return Math.Min(2, rawplacar) * 6;
    }

    private int GetPontoPorSimbolos(string senha)
    {
        int rawplacar = Regex.Replace(senha, "[a-zA-Z0-9]", "").Length;
        return Math.Min(2, rawplacar) * 5;
    }

    private int GetPontoPorRepeticao(string senha)
    {
    }
```

```

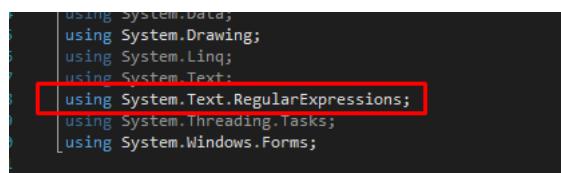
System.Text.RegularExpressions.Regex regex = new System.Text.RegularExpressions.Regex(@'
bool repete = regex.IsMatch(senha);
if (repete)
{
    return 30;
}
else
{
    return 0;
}
}

public ForcaDaSenha GetForcaDaSenha(string senha)
{
    int placar = geraPontosSenha(senha);

    if (placar < 50)
        return ForcaDaSenha.Inaceitavel;
    else if (placar < 60)
        return ForcaDaSenha.Fraca;
    else if (placar < 80)
        return ForcaDaSenha.Aceitavel;
    else if (placar < 100)
        return ForcaDaSenha.Forte;
    else
        return ForcaDaSenha.Segura;
}
}

```

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



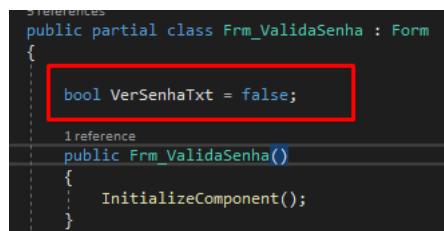
```

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;

```

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

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



```

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

```

```
bool VerSenhaTxt = false;
```

17) 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 = "";
}
```

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

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

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

19) 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";
    }
}
```

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

multiline	False
passwordchar	*
readonly	False

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

