**Tarefa: Validador de CPF**

Você foi designado para criar uma pequena aplicação de validação de CPF em C# usando Windows Forms. O objetivo é criar uma interface onde os usuários possam inserir um número de CPF em um campo de entrada e, em seguida, clicar em um botão para verificar se o CPF é válido ou não. Você deve usar a API MessageBox para exibir mensagens de sucesso ou erro, dependendo do resultado da validação.

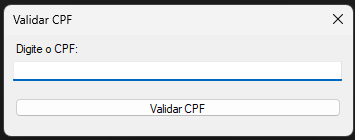
Aqui estão os requisitos da tarefa:

1. Crie uma interface Windows Forms que inclua os seguintes elementos:
   * Um **TextBox** para a entrada do CPF.
   * Um botão "Validar CPF" que os usuários podem clicar para iniciar a validação.
2. Implemente uma função que valide o CPF inserido no **TextBox**. A validação do CPF deve seguir as regras padrão, como garantir que o número contenha 11 dígitos numéricos válidos e que o dígito verificador esteja correto.
3. Quando o botão "Validar CPF" for clicado, a aplicação deve:
   * Verificar se o CPF inserido é válido de acordo com as regras de validação.
   * Exibir uma MessageBox de sucesso caso o CPF seja válido. A mensagem de sucesso deve informar que o CPF é válido.
   * Exibir uma MessageBox de erro caso o CPF seja inválido. A mensagem de erro deve informar que o CPF é inválido.
4. Se houver erros, forneça feedback ao usuário sobre o motivo da invalidade, como "CPF inválido devido ao dígito verificador incorreto" ou "CPF deve conter 11 dígitos numéricos".
5. Implemente tratamento de exceção para lidar com possíveis erros durante a validação, como entrada inválida.
6. Teste a aplicação com vários CPFs de exemplo, tanto válidos quanto inválidos, para garantir que ela funcione corretamente.
7. Validar a execução com as seguintes entradas:
   * 32462474803
   * 43912344134
   * 00000000000
   * 111000111

Lembre-se de criar uma interface amigável para o usuário e forneça mensagens informativas para facilitar a compreensão dos resultados.

Solução:

txtCpfEntrada



lblDigiteOCpf

btnValidarCPF

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTROLE** | **NOME** | **PROPRIEDADE** | **VALOR** |
| Form | frmPrincipal | Text | Validar CPF |
| StartPosition | CenterScreen |
| MaximizeBox | False |
| MinimizeBox | False |
| FormBorderStyle | FixedDialog |
| Size.Width | 483 |
| Size.Height | 200 |
| Label | lblDigiteOCpf | Text | Digite o CPF: |
| Location.X | 12 |
| Location.Y | 9 |
| Size.Width | 106 |
| Size.Height | 20 |
| TextBox | txtCpfEntrada | Location.X | 12 |
| Location.Y | 32 |
| Size.Width | 437 |
| Size.Height | 22 |
| MaxLength | 11 |
| Button | btnValidarCPF | Text | Valida&r |
| Location.X | 15 |
| Location.Y | 77 |
| Size.Width | 434 |
| Size.Height | 23 |

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ValidarCPF

{

public partial class frmPrincipal : Form

{

public frmPrincipal()

{

InitializeComponent();

}

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

{

string cpfDigitado = txtCpfEntrada.Text.Trim();

if (cpfDigitado.Length != 11 || string.IsNullOrWhiteSpace(cpfDigitado))

{

MessageBox.Show("Você não digitou a quantidade correta de digitos de um CPF",

"Erro",

MessageBoxButtons.OK,

MessageBoxIcon.Error);

} else

{

string cpfParcial = cpfDigitado.Substring(0, 9);

int soma = 0;

for (int i = 0; i < 9; i++)

{

soma += int.Parse(cpfParcial[i].ToString()) \* (10 - i);

}

int resto = soma % 11;

resto = resto < 2 ? 0 : 11 - resto;

cpfParcial += resto.ToString();

soma = 0;

for(int i = 0; i < 10; i++)

{

soma += int.Parse(cpfParcial[i].ToString()) \* (11 - i);

}

resto = soma % 11;

resto = resto < 2 ? 0 : 11 - resto;

cpfParcial += resto.ToString();

if ( cpfDigitado.Equals(cpfParcial))

{

MessageBox.Show("Você digitou um CPF válido",

"Sucesso",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

} else

{

MessageBox.Show("Você digitou um CPF inválido",

"Erro",

MessageBoxButtons.OK,

MessageBoxIcon.Exclamation);

}

}

}

private void txtCpfEntrada\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

btnValidarCPF\_Click(sender, e);

} else

{

if (!char.IsDigit(e.KeyChar) && e.KeyChar != (char)Keys.Back)

{

e.Handled = true;

}

}

}

}

}

|  |  |
| --- | --- |
| Validação | |
| 32462474803 |  |
| 43912344134 |  |
| 00000000000 |  |
| 111000111 |  |