ALUNO CAUÊ CARLOS DE AZEVEDO GUEDES, TDS3 – MANHÃ – SENAI CAMPUS DA INDÚSTRIA

FORMULÁRIO DE CONTATO

|  |
| --- |
| CLASSE Form1  using System;  using System.Drawing;  using System.Windows.Forms;  namespace FormularioContato\_  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  lblAssunto.Font = new Font("Arial", 9, FontStyle.Regular);  lblEmail.Font = new Font("Arial", 9, FontStyle.Regular);  lblMensagem.Font = new Font("Arial", 9, FontStyle.Regular);  lblNome.Font = new Font("Arial", 9, FontStyle.Regular);  lblTelefone.Font = new Font("Arial", 9, FontStyle.Regular);  lblTitulo.Font = new Font("Arial", 12, FontStyle.Regular);  chkNewsLetter.Font = new Font("Arial", 9, FontStyle.Regular);  btnEnviar.Font = new Font("Arial", 10, FontStyle.Regular);  btnLimpar.Font = new Font("Arial", 10, FontStyle.Regular);  cmbAssunto.Items.Add("Dúvida");  cmbAssunto.Items.Add("Reclamação");  cmbAssunto.Items.Add("Sugestão");  }  private void btnEnviar\_Click(object sender, EventArgs e)  {  try  {  Contato contato = new Contato  {  Nome = txtNome.Text,  Email = txtEmail.Text,  Telefone = txtTelefone.Text,  Assunto = cmbAssunto.Text,  Mensagem = txtMensagem.Text,  ReceberNews = chkNewsLetter.Checked  };  if (!contato.Validar(out string erro))  {  MessageBox.Show(erro, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  MessageBox.Show(contato.ToString(), "Mensagem Enviada");  }  catch (Exception ex)  {  MessageBox.Show($"Erro: {ex.Message}", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  private void btnLimpar\_Click(object sender, EventArgs e)  {  LimparFormulario();  }  private void LimparFormulario()  {  txtNome.Clear();  txtEmail.Clear();  txtTelefone.Clear();  txtMensagem.Clear();  cmbAssunto.SelectedIndex = -1;  chkNewsLetter.Checked = false;  }  }  } |

|  |
| --- |
| CLASSE Contato  using System.Text.RegularExpressions;  namespace FormularioContato\_  {  public class Contato  {  public string Nome { get; set; }  public string Email { get; set; }  public string Telefone { get; set; }  public string Assunto { get; set; }  public string Mensagem { get; set; }  public bool ReceberNews { get; set; }  public bool Validar(out string erro)  {  if (string.IsNullOrWhiteSpace(Nome))  {  erro = "Nome é obrigatório!";  return false;  }  if (!Regex.IsMatch(Email, @"^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"))  {  erro = "Por favor, insira um e-mail válido.";  return false;  }  if (!Regex.IsMatch(Telefone, @"^\(?\d{2}\)?\s?\d{4,5}-?\d{4}$"))  {  erro = "Por favor, insira um número de telefone válido.";  return false;  }  if (string.IsNullOrWhiteSpace(Assunto))  {  erro = "Assunto é obrigatório!";  return false;  }  if (string.IsNullOrWhiteSpace(Mensagem))  {  erro = "Mensagem é obrigatória!";  return false;  }  erro = null;  return true;  }  public override string ToString()  {  return $" Nome: {Nome}\n Email: {Email}\n Telefone: {Telefone}\n Assunto: {Assunto}\n Mensagem: {Mensagem}\n Receber NewsLetter: {(ReceberNews ? "Sim" : "Não")}";  }  }  } |