using System;

using System.Reflection.Emit;

using System.Text;

using System.Windows.Forms;

using static System.Net.Mime.MediaTypeNames;

namespace PracticaCSharpGUI

{

static class Program

{

[STAThread]

static void Main()

{

// Inicialización de WinForms en .NET 6+

ApplicationConfiguration.Initialize();

// Crear formulario

Form mainForm = new Form()

{

Text = "Cifrado César - GUI",

Width = 500,

Height = 350,

FormBorderStyle = FormBorderStyle.FixedDialog,

MaximizeBox = false

};

// Etiqueta y TextBox para entrada de texto

Label lblInput = new Label() { Text = "Texto:", Left = 10, Top = 20, AutoSize = true };

TextBox txtInput = new TextBox() { Left = 70, Top = 18, Width = 400 };

// Etiqueta y NumericUpDown para clave

Label lblKey = new Label() { Text = "Clave:", Left = 10, Top = 60, AutoSize = true };

NumericUpDown numKey = new NumericUpDown()

{

Left = 70,

Top = 58,

Width = 60,

Minimum = -25,

Maximum = 25,

Value = 3

};

// RadioButtons para elegir cifrar o descifrar

RadioButton rbEncrypt = new RadioButton() { Text = "Cifrar", Left = 150, Top = 58, Checked = true };

RadioButton rbDecrypt = new RadioButton() { Text = "Descifrar", Left = 230, Top = 58 };

// Botón para ejecutar

Button btnExecute = new Button() { Text = "Ejecutar", Left = 10, Top = 100, Width = 460 };

// Etiqueta y TextBox para salida

Label lblOutput = new Label() { Text = "Resultado:", Left = 10, Top = 150, AutoSize = true };

TextBox txtOutput = new TextBox() { Left = 10, Top = 170, Width = 460, Height = 100, Multiline = true, ReadOnly = true, ScrollBars = ScrollBars.Vertical };

// Agregar controles al formulario

mainForm.Controls.AddRange(new Control[]

{

lblInput, txtInput,

lblKey, numKey,

rbEncrypt, rbDecrypt,

btnExecute,

lblOutput, txtOutput

});

// Evento del botón

btnExecute.Click += (sender, e) =>

{

string text = txtInput.Text;

int key = (int)numKey.Value;

int effectiveKey = ((key % 26) + 26) % 26;

if (rbDecrypt.Checked) effectiveKey = -effectiveKey;

txtOutput.Text = Transform(text, effectiveKey);

};

Application.Run(mainForm);

}

static string Transform(string text, int key)

{

var sb = new StringBuilder(text.Length);

foreach (char c in text)

{

if (!char.IsLetter(c)) { sb.Append(c); continue; }

char baseChar = char.IsUpper(c) ? 'A' : 'a';

int pos = c - baseChar;

int newPos = (pos + key + 26) % 26;

sb.Append((char)(baseChar + newPos));

}

return sb.ToString();

}

}

}