UNIVERSIDAD GERARDO BARRIOS

INGENIERIA EN SISTEMAS Y REDES IMFORMATICAS CICLO: I-2024

PROGRAMACION COMPUTACIONAL II

CATEDRATICO: Lic. Luis Enrique Hernández Sánchez

Tema: Windows Forms en C#

RICARDO FLORES ECHEVERRIA

ISNP171322

SANTA TECLA 28 DE ABRIL DEL 2024

**INDICACIONES**

1. Crear un proyecto de Windows Forms en Visual Studio C# el cual debe nombrarse como **“CodigoEstudiante\_Unidad2”,** en donde **CodigoEstudiante** corresponde a su código de estudiante, ejemplo: **SMIS00123\_Unidad2**.
2. Luego, para cada ejercicio propuesto, deberá crear un formulario (Windows Forms), al cual deberá nombrar con el número del ejercicio, ejemplo: **Ejercicio1, Ejercicio2, Ejercicio3, …** etc.
3. Finalizada la guía de ejercicios, en un documento de Word, debe agregar para cada ejercicio su código de solución respectivo, una caratula, y guárdelo con el nombre del proyecto.
4. Por último, agregue en una sola carpeta, el proyecto Windows Forms y el documento de Word, guárdelo con el nombre de la actividad, en formato .zip o .rar, y envíela al espacio de tarea: **Unidad 2 – Actividad 2 – Windows Forms en C#**

**EJERCICIOS PROPUESTOS**

1. Desarrolle una aplicación que permita convertir de libras a kilogramos y viceversa.
2. using System;
3. using System.Collections.Generic;
4. using System.ComponentModel;
5. using System.Data;
6. using System.Drawing;
7. using System.Globalization;
8. using System.Linq;
9. using System.Text;
10. using System.Threading.Tasks;
11. using System.Windows.Forms;
12. namespace Ejercicio\_1
13. {
14. public partial class ComversorForms: Form
15. {
16. public ComversorForms()
17. {
18. InitializeComponent();
19. }
20. private void btnConver1\_Click(object sender, EventArgs e)
21. {
22. if (double.TryParse(textBox1.Text, NumberStyles.Any, CultureInfo.InvariantCulture, out double libras))
23. {
24. double kg = libras \* 0.453592; // 1 libra = 0.453592 kilogramos
25. textBox3.Text = kg.ToString("0.0000");
26. }
27. else
28. {
29. MessageBox.Show("Por favor, introduzca un valor válido en libras.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
30. }
31. }
32. private void button1\_Click(object sender, EventArgs e)
33. {
34. if (double.TryParse(textBox2.Text, NumberStyles.Any, CultureInfo.InvariantCulture, out double kg))
35. {
36. double libras = kg / 0.453592; // 1 kilogramo = 2.20462 libras
37. textBox4.Text = libras.ToString("0.0000");
38. }
39. else
40. {
41. MessageBox.Show("Por favor, introduzca un valor válido en kilogramos.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
42. }
43. }
44. private void button2\_Click(object sender, EventArgs e)
45. {
46. Application.Exit();
47. }
48. }
49. }
50. Desarrolle una aplicación que convierta de metros a yardas, centímetros y milímetros y viceversa.
51. using System;
52. using System.Collections.Generic;
53. using System.ComponentModel;
54. using System.Data;
55. using System.Drawing;
56. using System.Linq;
57. using System.Text;
58. using System.Threading.Tasks;
59. using System.Globalization;//libreria usada para validar los textbox
60. using System.Runtime.InteropServices;//libreria usada para mover el form
61. using System.Windows.Forms;
62. using static System.Windows.Forms.VisualStyles.VisualStyleElement;
63. namespace Ejercicio\_2
64. {
65. public partial class ConversorLongForms : Form
66. {
68. public ConversorLongForms()
69. {
70. InitializeComponent();
72. // Llenar el ComboBox con las opciones de longitud
73. cmbOrigen.Items.Add("Metros");
74. cmbOrigen.Items.Add("Yardas");
75. cmbOrigen.Items.Add("Centímetros");
76. cmbOrigen.Items.Add("Milímetros");
78. // Seleccionar la primera opción por defecto
79. cmbOrigen.SelectedIndex = 0;
80. // Llenar el ComboBox con las opciones de longitud
81. cmbdestino.Items.Add("Metros");
82. cmbdestino.Items.Add("Yardas");
83. cmbdestino.Items.Add("Centímetros");
84. cmbdestino.Items.Add("Milímetros");
85. // Seleccionar la primera opción por defecto
86. cmbdestino.SelectedIndex = 1;
87. }
88. private void RealizarConversion(string unidadResultado)
89. {
91. try
92. {
94. double valor = double.Parse(texCant.Text);
95. double resultado = 0.0;
97. // Realizar la conversión basada en las unidades seleccionadas en los ComboBoxes
98. if (cmbOrigen.SelectedIndex == 0 && cmbdestino.SelectedIndex == 1) // Metros a Yardas
99. {
100. resultado = valor \* 1.09361;
101. unidadResultado = "yrd";
102. }
103. else if (cmbOrigen.SelectedIndex == 1 && cmbdestino.SelectedIndex == 0) // Yardas a Metros
104. {
105. resultado = valor / 1.09361;
106. unidadResultado = "mt";
107. }
108. else if (cmbOrigen.SelectedIndex == 0 && cmbdestino.SelectedIndex == 2) // Metros a Centímetros
109. {
110. resultado = valor \* 100;
111. unidadResultado = "cm";
112. }
113. else if (cmbOrigen.SelectedIndex == 2 && cmbdestino.SelectedIndex == 0) // Centímetros a Metros
114. {
115. resultado = valor / 100;
116. unidadResultado = "mt";
117. }
118. else if (cmbOrigen.SelectedIndex == 0 && cmbdestino.SelectedIndex == 3) // Metros a Milímetros
119. {
120. resultado = valor \* 1000;
121. unidadResultado = "mm";
122. }
123. else if (cmbOrigen.SelectedIndex == 3 && cmbdestino.SelectedIndex == 0) // Milímetros a Metros
124. {
125. resultado = valor / 1000;
126. unidadResultado = "mt";
127. }
128. else if (cmbOrigen.SelectedIndex == 1 && cmbdestino.SelectedIndex == 3) // Yardas a Milímetros
129. {
130. resultado = valor \* 914.4;
131. unidadResultado = "mm";
132. }
133. else if (cmbOrigen.SelectedIndex == 3 && cmbdestino.SelectedIndex == 1) // Milímetros a Yardas
134. {
135. resultado = valor / 914.4;
136. unidadResultado = "yrd";
137. }
138. else if (cmbOrigen.SelectedIndex == 2 && cmbdestino.SelectedIndex == 3) // Centímetros a Milímetros
139. {
140. resultado = valor \* 10;
141. unidadResultado = "mm";
142. }
143. else if (cmbOrigen.SelectedIndex == 1 && cmbdestino.SelectedIndex == 2) // Yardas a Centímetros
144. {
145. resultado = valor \* 91.44;
146. unidadResultado = "cm";
147. }
148. else if (cmbOrigen.SelectedIndex == 2 && cmbdestino.SelectedIndex == 1) // Yardas a Centímetros
149. {
150. resultado = valor / 91.44;
151. unidadResultado = "cm";
152. }
153. else if (cmbOrigen.SelectedIndex == 3 && cmbdestino.SelectedIndex == 2) // Milímetros a centímetros
154. {
155. resultado = valor / 10;
156. unidadResultado = "cm";
157. }
158. // Agregar más condiciones según sea necesario para otras conversiones
159. resultado = Math.Round(resultado, 4);
160. texReslut.Text = resultado.ToString() + " " + unidadResultado;
161. }
162. catch (Exception)
163. {
164. MessageBox.Show("Error por favor ingrese los valores solicitados", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
165. }
166. }
167. private void texCant\_KeyPress(object sender, KeyPressEventArgs e)
168. {
169. // Verificar si la tecla presionada no es un número o una tecla de control
170. if (!char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar))
171. {
172. // Mostrar una advertencia al usuario
173. MessageBox.Show("Por favor, ingresa solo números.", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);
174. // Detener la tecla presionada para que no se escriba en el TextBox
175. e.Handled = true;
176. }
177. }
178. private void btnconver\_Click(object sender, EventArgs e)
179. {
180. RealizarConversion(texReslut.Text);
182. }
183. private void btnclear\_Click(object sender, EventArgs e)
184. {
185. //usado para dejar en blanco los textbox
186. texCant.Text = "";
187. texReslut.Text = "";
188. }
189. private void btnexit\_Click(object sender, EventArgs e)
190. {
191. Application.Exit();
192. }
194. private void label6\_Click(object sender, EventArgs e)
195. {
196. AboutBox1 formAcercaDe = new AboutBox1();
197. formAcercaDe.ShowDialog();
198. }
199. }
200. }
201. Desarrollar una calculadora que permita realizar las cuatro operaciones básicas y además elevar a determinada potencia y extraer la raíz cuadrada de un número.
202. using System;
203. using System.Collections.Generic;
204. using System.ComponentModel;
205. using System.Data;
206. using System.Drawing;
207. using System.Linq;
208. using System.Text;
209. using System.Threading.Tasks;
210. using System.Windows.Forms;
211. namespace Ejericio\_3
212. {
213. public partial class Form1 : Form
214. {
215. public Form1()
216. {
217. InitializeComponent();
218. Timer timer = new Timer();
219. timer.Interval = 1000; // 1 segundo
220. timer.Tick += tTime\_Tick;
221. timer.Start();
222. }
223. private void rdbExponencial\_CheckedChanged(object sender, EventArgs e)
224. {
225. if (rdbExponencial.Checked)
226. {
227. lblSimbol.Text = "n^n";
228. }
229. else
230. {
231. lblSimbol.Text = "";
232. }
233. }
234. private void rdbRaiz\_CheckedChanged(object sender, EventArgs e)
235. {
236. if (rdbRaiz.Checked)
237. {
238. texNum1.Text = "2"; // Establecer num1 como 2 cuando se selecciona la raíz cuadrada
239. lblSimbol.Text = "√";
240. texNum1.Enabled = false; // Deshabilitar la edición de txtNum1
241. }
242. else
243. {
244. texNum1.Enabled = true; // Habilitar la edición de txtNum1 si no se selecciona la raíz cuadrada
245. texNum1.Text = "";
246. lblSimbol.Text = "";
247. }
248. }
249. private void rdbRaizCub\_CheckedChanged(object sender, EventArgs e)
250. {
251. if (rdbRaizCub.Checked)
252. {
253. texNum1.Text = "3"; // Establecer num1 como 3 cuando se selecciona la raíz cúbica
254. lblSimbol.Text = "∛";
255. texNum1.Enabled = false; // Deshabilitar la edición de txtNum1
256. }
257. else
258. {
259. texNum1.Enabled = true; // Habilitar la edición de txtNum1 si no se selecciona la raíz cuadrada
260. texNum1.Text = "";
261. lblSimbol.Text = "";
262. }
263. }
264. private void btnCalcular\_Click(object sender, EventArgs e)
265. {
266. try
267. {
268. double num1 = double.Parse(texNum1.Text);
269. double num2 = double.Parse(texNum2.Text);
270. double respuesta = 0;
271. if (num1 == 0 || num2 == 0)
272. {
273. MessageBox.Show("No se puede calcular con cero.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
274. return;
275. }
276. if (rbdSuma.Checked)
277. {
278. respuesta = num1 + num2;
279. }
280. else if (rdbResta.Checked)
281. {
282. respuesta = num1 - num2;
283. }
284. else if (rdbMulti.Checked)
285. {
286. respuesta = num1 \* num2;
287. }
288. else if (rdbDivision.Checked)
289. {
290. respuesta = num1 / num2;
291. }
292. else if (rdbExponencial.Checked)
293. {
294. respuesta = Math.Pow(num1, num2);
295. }
296. else if (rdbRaiz.Checked)
297. {
298. respuesta = num1 == 2 ? Math.Sqrt(num2) : Math.Pow(num2, num1); // Raiz cuadrada
299. }
300. else if (rdbRaizCub.Checked)
301. {
302. respuesta = num1 == 3 ? Math.Pow(num2, 1.0 / 3.0) : Math.Sqrt(num1); // Raíz cúbica
303. }
304. respuesta = Math.Round(respuesta, 10);
305. texResulta.Text = respuesta.ToString();
306. }
307. catch (Exception)
308. {
309. MessageBox.Show("Error, por favor ingreso los valores solicitados", "Calculadora", MessageBoxButtons.OK, MessageBoxIcon.Error);
310. }
311. }
312. private void btnClear\_Click(object sender, EventArgs e)
313. {
314. //usado para dejar en blanco los textbox
315. texNum1.Text = "";
316. texNum2.Text = "";
317. texResulta.Text = "";
318. }
319. private void lblDateTime\_Click(object sender, EventArgs e)
320. {
321. // Obtener la fecha y hora actual
323. DateTime currentDateTime = DateTime.Now;
324. // Mostrar la fecha y hora en un Label llamado "lblDateTime"
325. lblDateTime.Text = currentDateTime.ToString();
326. }
327. private void Form1\_Load(object sender, EventArgs e)
328. {
329. // Obtener la fecha y hora actual
330. DateTime currentDateTime = DateTime.Now;
331. // Mostrar la fecha y hora en un Label llamado "lblDateTime" en un formato específico
332. lblDateTime.Text = "Fecha: " + currentDateTime.ToString("dd/MM/yyyy");
333. }
334. private void tTime\_Tick(object sender, EventArgs e)
335. {
336. DateTime currentDateTime = DateTime.Now;
337. string formattedTime = currentDateTime.ToString("tt");
338. // Obtener la hora actual sincronizada
339. DateTime horaActual = DateTime.Now;
340. lblTime.Text = $"Hora: {horaActual.ToString("hh:mm:ss")} "+ formattedTime;
341. }
342. private void button1\_Click(object sender, EventArgs e)
343. {
344. Application.Exit();
345. }
346. }
347. }

4. Desarrollar un programa que permita determinar el total a pagar por una compra de la cual se sabe el valor unitario y la cantidad. Se debe tener en cuenta que se realiza un descuento del 15% por compras inferiores a $20000 y del 35% por compras mayores o iguales a $20000.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlTypes;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Ejercicio\_4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

Timer timer = new Timer();

timer.Interval = 1000; // Esto es Igual a 1 segundo

timer.Tick += timer1\_Tick;

timer.Start();

}

private void button1\_Click(object sender, EventArgs e)

{

// Verificar si los campos de entrada están en blanco

if (string.IsNullOrWhiteSpace(txtUnitario.Text) || string.IsNullOrWhiteSpace(txtCant.Text))

{

MessageBox.Show("Por favor, complete todos los campos.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return; // Detener la ejecución del método por dejar campos vacios

}

double Uni = double.Parse(txtUnitario.Text);

int Cant = int.Parse(txtCant.Text);

if (Uni == 0 || Cant == 0)

{

MessageBox.Show("No se puede calcular con cero.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

// Obtener el valor unitario ingresado por el usuario

decimal vUnitario = decimal.Parse(txtUnitario.Text);

// Obtener la cantidad ingresada por el usuario

int cantidad = int.Parse(txtCant.Text);

// Calcular el total sin descuento

decimal totalSinDescu = CalTotalSinDescu(vUnitario, cantidad);

// Calcular el descuento aplicado

decimal descuento = CalcularDescuento(totalSinDescu);

// Calcular el total a pagar con descuento

decimal totalConDescu = totalSinDescu - descuento;

// Mostrar el resultado en la etiqueta correspondiente separando por coma los miles

txtTotalP.Text = "" + totalConDescu.ToString("#,##0.00");

// Bloquear el campo de texto del monto a pagar para no modificar el valor final

txtDescuento.ReadOnly = true;

txtTotalP.ReadOnly = true;

}

// Método para calcular el total sin descuento

private decimal CalTotalSinDescu(decimal valorUnitario, int cantidad)

{

return valorUnitario \* cantidad;

}

// Método para calcular el descuento aplicado

private decimal CalcularDescuento(decimal totalSinDescuento)

{

decimal descuento;

if (totalSinDescuento < 20000)

{

descuento = totalSinDescuento \* 0.15m; // 15% de descuento

txtDescuento.Text = "15%";

}

else

{

descuento = totalSinDescuento \* 0.35m; // 35% de descuento

txtDescuento.Text = "35%";

}

return descuento;

}

// Método para calcular el total a pagar ya aplicando el respectivo descuento

private decimal CalcularTotalPagar(decimal totalSinDescuento, decimal descuento)

{

return totalSinDescuento - descuento;

}

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

{

//Con este podemos limpiar todos los textbox

txtUnitario.Text = "";

txtCant.Text = "";

txtDescuento.Text = "";

txtTotalP.Text = "";

}

private void button2\_Click(object sender, EventArgs e)

{

//Este metodo sirve para cerrar la aplicación

Application.Exit();

}

private void Form1\_Load(object sender, EventArgs e)

{

// Obtener la fecha y hora actual

DateTime currentDateTime = DateTime.Now;

// Mostrar la fecha y hora en un Label llamado "lblDateTime" en un formato específico

lblDate.Text = "Fecha: " + currentDateTime.ToString("dd/MM/yyyy");

}

private void timer1\_Tick(object sender, EventArgs e)

{

DateTime currentDateTime = DateTime.Now;

string formattedTime = currentDateTime.ToString("tt");

// Obtener la hora actual sincronizada

DateTime horaActual = DateTime.Now;

lblTime.Text = $"Hora: {horaActual.ToString("hh:mm:ss")} " + formattedTime;

}

private void txtCant\_TextChanged(object sender, EventArgs e)

{

// Verificar si el contenido del campo txtCant es un número entero

if (!int.TryParse(txtCant.Text, out int cantidad))

{

// Mostrar mensaje de error si no es un número entero y eliminar el último carácter ingresado

MessageBox.Show("Solo se pueden ingresar valores enteros en el campo de cantidad.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

txtCant.Text = txtCant.Text.Substring(0, txtCant.Text.Length - 1); // Eliminar el último carácter ingresado

txtCant.SelectionStart = txtCant.Text.Length; // Colocar el cursor al final del texto

}

}

}

}

1. Desarrollar una aplicación que permita a un alumno conocer el promedio final de Programación II conociendo los laboratorios y parcial de los tres cómputos.

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 Ejercicio\_5

{

public partial class Form1 : Form

{

private const int NumeroLaboratorios = 3;

private const int NumeroParciales = 2;

public Form1()

{

InitializeComponent();

}

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

{

// Obtener las calificaciones de los laboratorios y parciales

double lab1 = Convert.ToDouble(txtLab1.Text);

double lab2 = Convert.ToDouble(txtLab2.Text);

double lab3 = Convert.ToDouble(txtLab3.Text);

double parcial1 = Convert.ToDouble(txtPar1.Text);

double parcial2 = Convert.ToDouble(txtPar2.Text);

double parcial3 = Convert.ToDouble(txtPar3.Text);

// Calcular el promedio final

double promedioLaboratorios = (lab1 + lab2 + lab3) / 3;

double promedioParciales = (parcial1 + parcial2 + parcial3) / 3;

double promedioFinal = (promedioLaboratorios + promedioParciales) / 2;

// Mostrar el promedio final en el cuadro de texto correspondiente

txtNotaFin.Text = promedioFinal.ToString("0.00");

}

}

}

1. Desarrolle una aplicación que convierta un número del 1 al 20 a su equivalente en Romano.

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 Ejercicio\_6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btnConvertir\_Click\_1(object sender, EventArgs e)

{

int numDecimal;

if (int.TryParse(txtnumero.Text, out numDecimal))

{

if (numDecimal >= 1 && numDecimal <= 20)

{

string numeroRomano = ConvertirANumRomano(numDecimal);

lblResltado.Text = $"Romano equivalente es: {numeroRomano}";

}

else

{

MessageBox.Show("Ingrese un número entre 1 y 20.", "Número Inválido", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

MessageBox.Show("Ingrese un número válido.", "Número Inválido", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private string ConvertirANumRomano(int numero)

{

string[] unidades = { "", "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX" };

string[] decenas = { "", "X", "XX", "XXX", "XL", "L" };

if (numero <= 9)

{

return unidades[numero];

}

else if (numero <= 19)

{

return decenas[numero / 10] + unidades[numero % 10];

}

else

{

return "XX";

}

}

private void button1\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

1. Desarrolle una aplicación para sumar los primeros 100 números naturales.

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 Ejercicio\_7

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

{

// Realizamos la suma de los primeros 100 números naturales

int suma = SumarPrimerosNNumeros(100);

// Mostramos el resultado en el Label

lblResultado.Text = $"La suma de los primeros 100 números naturales es: {suma}";

}

{

// Realizamos la suma de los primeros 100 números naturales

int suma = SumarPrimerosNNumeros(100);

// Mostramos el resultado en el Label

lblResultado.Text = $"La suma de los primeros 100 números naturales es: {suma}";

}

}

private int SumarPrimerosNNumeros(int n)

{

// Inicializamos la suma en cero

int suma = 0;

// Iteramos desde 1 hasta n y sumamos cada número

for (int i = 1; i <= n; i++)

{

suma += i;

}

return suma;

}

private void button1\_Click(object sender, EventArgs e)

{

MessageBox.Show("¡Gracias por usar este programa!", "¡Hasta luego!", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close(); // Cierra la aplicación

}

}

}

1. Desarrollar una aplicación que muestre la tabla de multiplicar de un número del 1 al 10 en el formato 1x1=1, 1x2=2.
2. using System;
3. using System.Collections.Generic;
4. using System.ComponentModel;
5. using System.Data;
6. using System.Drawing;
7. using System.Linq;
8. using System.Text;
9. using System.Threading.Tasks;
10. using System.Windows.Forms;
11. using System.Xml.Linq;
12. namespace Ejercicio\_8
13. {
14. public partial class Form1 : Form
15. {
16. public Form1()
17. {
18. InitializeComponent();
19. }
20. private void btnMostrar\_Click(object sender, EventArgs e)
21. {
22. // Limpiamos el TextBox antes de mostrar la nueva tabla de multiplicar
23. txtTablaMulti.Clear();
24. // Verificamos si el usuario ingresó un número válido
25. if (int.TryParse(txtNTabla.Text, out int numeroTabla))
26. {
27. // Si el número es mayor que 10, mostramos un mensaje de error
28. if (numeroTabla > 10)
29. {
30. MessageBox.Show("Solo se puede mostrar hasta la tabla del 10.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
31. numeroTabla = 10; // Cambiamos el número a 10 para mostrar la tabla del 10
32. }
33. // Construimos la tabla de multiplicar del número ingresado
34. txtTablaMulti.AppendText($"Tabla del {numeroTabla}:\r\n");
35. for (int i = 1; i <= 10; i++)
36. {
37. int resultado = numeroTabla \* i;
38. txtTablaMulti.AppendText($"{numeroTabla} x {i} = {resultado}\r\n");
39. }
40. }
41. else
42. {
43. MessageBox.Show("Por favor, ingrese un número válido (1-10).", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
44. }
45. }
46. private void btnExit\_Click(object sender, EventArgs e)
47. {
48. MessageBox.Show("¡Gracias por usar este programa!", "¡Hasta luego!", MessageBoxButtons.OK, MessageBoxIcon.Information);
49. this.Close(); // Cierra la aplicación
50. }
51. }
52. }
53. Desarrollar una aplicación para encontrar el mayor de 10 números.

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 Ejercicio\_9

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

// Array para almacenar los números ingresados por el usuario

double[] numeros = new double[10];

// Obtener los números ingresados por el usuario y almacenarlos en el array

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

{

if (!double.TryParse(Controls[$"txtNumero{i + 1}"].Text, out numeros[i]))

{

MessageBox.Show("Por favor, ingrese números válidos en todos los campos.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

}

// Encontrar el número mayor en el array

double mayor = numeros.Max();

// Mostrar el número mayor en el Label

lblResult.Text = $"El mayor de los 10 números es: {mayor}";

}

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

{

MessageBox.Show("¡Gracias por usar este programa!", "¡Hasta luego!", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close(); // Cierra la aplicación

}

}

}

1. Desarrollar una aplicación que solicite al usuario y contraseña y que si se ingresa ugb en ambos muestre un mensaje Bienvenido, caso contrario muestre acceso denegado.

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 Ejercicio\_10

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

// Obtener el nombre de usuario y la contraseña ingresados por el usuario

string usuario = txtUser.Text.Trim();

string contraseña = txtPass.Text.Trim();

// Verificar si ambos campos están llenos

if (string.IsNullOrWhiteSpace(usuario) || string.IsNullOrWhiteSpace(contraseña))

{

MessageBox.Show("Por favor, llene todos los campos.", "Advertencia", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return; // Salir del método para evitar continuar con la verificación

}

// Verificar si el nombre de usuario y la contraseña son "ugb"

if (usuario == "ugb" && contraseña == "ugb")

{

// Mostrar mensaje de bienvenida si las credenciales son correctas

lblResult.Text = "Bienvenido Al Portal del Estudiante";

}

else

{

// Mostrar mensaje de acceso denegado si las credenciales son incorrectas

lblResult.Text = "Acceso denegado. ¿Tienes problemas para iniciar sesión?";

}

}

private void button2\_Click(object sender, EventArgs e)

{

MessageBox.Show("¡Gracias por utilizar este programa!", "¡Hasta luego!", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close(); // Cierra la aplicación

}

}

}