



UAI

Universidad Abierta Interamericana

EVALUACIÓN PARCIAL – RESOLUCIÓN DE CASOS

ALUMNO:

- Di Salvio Tiago
 - 42727210
 - tiago.disalvio@alumnos.uai.edu.ar

CARRERA: Ingeniería en Sistemas Informáticos

ASIGNATURA: Programación I

FECHA: 04/10/2021

Aspectos procedimentales

1. Desarrolle un programa que por medio de un control Hscrollbar se pueda configurar el intervalo de un timer entre 0 y 5 segundos. El valor seleccionado se debe observar en un control label. Colocar un boton de comando que permita activar y desactivar el timer. Cuando el timer esté activado, en la medida que transcurra el tiempo configurado en segundos, se deberán generar números aleatorios.

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Parcial1Ejercicio1
12 {
13     4 referencias
14     public partial class Form1 : Form
15     {
16         int Contador = 1;
17
18         1 referencia
19         public Form1()
20         {
21             InitializeComponent();
22         }
23
24         1 referencia
25         private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)
26         {
27             string ScrollValue = Convert.ToString(hScrollBar1.Value);
28             LabelValor.Text = ScrollValue;
29         }
30
31         1 referencia
32         private void inicioTimer_Click(object sender, EventArgs e)
33         {
34             Contador = 1;
35             if (hScrollBar1.Value == 0)
36             {
37                 MessageBox.Show("Ingresar un valor en el rango de 1 a 5 segundos");
38             }
39             else
40             {
41                 timer.Start();
42             }
43         }
44
45         1 referencia
46         private void finTimer_Click(object sender, EventArgs e)
47         {
48             timer.Stop();
49         }
50
51         1 referencia
52         private void timer_Tick(object sender, EventArgs e)
53         {
54             int ScrollValue = hScrollBar1.Value;
55             if (Contador <= ScrollValue)
56             {
57                 Random random_number = new Random();
58                 LabelContar.Text = Convert.ToString(Contador);
59                 LabelRandom.Text = Convert.ToString(random_number.Next(100));
60                 Contador++;
61             }
62             else
63             {
64                 timer.Stop();
65             }
66         }
67     }
68 }
```

- Desarrolle un programa que utilizando una lista desarrollada por Ud. emule la cobranza de una caja. Para ello en la cola se deberán cargar los clientes que genere el usuario. Un cliente posee nombre y un importe que puede contener valores decimales. Este valor decimal emula el importe de compra realizado. Solo defina las operaciones requeridas para este enunciado.

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11  namespace Parcial2Mueve
12  {
13      [Serializable]
14      public partial class Form1 : Form
15      {
16          public int BuscarMaximo(int ContadorVenta)
17          {
18              int i = 0;
19              bool salir = false;
20
21              while (salir != true) //Búsqueda de maximo
22              {
23                  if (matrizImporte[ContadorVenta, i] == 1000000) //valor buscado
24                  {
25                      salir = true;
26                      i++;
27                  }
28                  if (i < 99)
29                  {
30                      i = 0;
31                      return i;
32                  }
33                  else
34                  {
35                      return i;
36                  }
37              }
38
39              public int Contador = 0;
40              public int ContadorVenta = 0;
41              bool VentaFinalizada = false;
42              public string[] matrizNombreCliente = new string[100, 100];
43              public float[,] matrizImporte = new float[100, 100];
44              public string[] matrizConcepto = new string[100, 100];
45
46              private void InitializeComponent()
47              {
48                  listViewData.Columns.Add("Detalle", 150);
49                  listViewData.Columns.Add("Importe", 100);
50                  listViewData.View = View.Details;
51
52                  listViewDataOld.Columns.Add("Detalle", 150);
53                  listViewDataOld.Columns.Add("Importe", 100);
54                  listViewDataOld.View = View.Details;
55              }
56
57              public Form1()
58              {
59                  InitializeComponent();
60              }
61
62              private void Form1_Load(object sender, EventArgs e)
63              {
64                  InitializeComponent();
65                  listViewDataOld.Visible = false;
66              }
67
68              private void TextBoxNombre_TextChanged(object sender, EventArgs e)
69              {
70              }
71
72              private void listViewData_SelectedIndexChanged(object sender, EventArgs e)
73              {
74              }
75
76              private void ButtonIngresar_Click(object sender, EventArgs e)
77              {
78                  if (VentaFinalizada == true)
79                  {
80                      MessageBox.Show("Venta Finalizada! ¿Presione Iniciar Venta para continuar!");
81                  }
82                  else
83                  {
84                      float importe;
85                      bool result = float.TryParse(TextBoxImporte.Text, out importe);
86                      if (TextBoxNombre.Text.Length > 0) //verificación
87                      {
88                          if (result == true)
89                          {
90                              matrizNombreCliente[ContadorVenta, Contador] = TextBoxNombre.Text;
91                              matrizImporte[ContadorVenta, Contador] = float.Parse(TextBoxImporte.Text);
92                              labelNombreCliente.Text = "Nombre del cliente:" + TextBoxNombre.Text;
93                              labelNumeroPedido.Text = "Número de pedido:" + ContadorVenta;
94                              labelImportePedido.Text = "Importe del pedido:" + TextBoxImporte.Text;
95                              listViewData.Items.Add(new ListViewItem { Text = matrizNombreCliente[ContadorVenta, Contador],
96                                                                                               Text2 = matrizImporte[ContadorVenta, Contador].ToString() });
97                              Contador++;
98                          }
99                          else
100                          {
101                              MessageBox.Show("ERROR! \ningresar un valor numerico");
102                          }
103                      }
104                      else
105                      {
106                          MessageBox.Show("ERROR! \ningresar Nombre del Cliente");
107                      }
108                  }
109              }
110
111              private void labelNombre_Click(object sender, EventArgs e)
112              {
113              }
114
115              private void ButtonCerrarPedido_Click(object sender, EventArgs e)
116              {
117                  matrizConcepto[ContadorVenta, Contador] = "1000000"; //guarda registro
118                  matrizImporte[ContadorVenta, Contador] = 1000000; //guarda registro
119                  ContadorVenta++;
120                  VentaFinalizada = true;
121                  Contador = 0;
122              }
123
124              private void Button_Click(object sender, EventArgs e)
125              {
126                  if (VentaFinalizada == false)
127                  {
128                      MessageBox.Show("ERROR! \nVenta no terminada");
129                  }
130                  if (VentaFinalizada == true)
131                  {
132                      Contador = 0;
133                      VentaFinalizada = false;
134                      listViewData.Clear();
135                      InitializeComponent();
136                  }
137              }
138
139              private void listViewData_SelectedIndexChanged(object sender, EventArgs e)
140              {
141              }
142
143              private void ButtonMostrarPedido_Click(object sender, EventArgs e)
144              {
145                  listViewDataOld.Clear();
146                  InitializeComponent();
147                  int buscar = Convert.ToInt32(numeroPedidoMostrarVenta.Value);
148                  float total = 0;
149                  int fin = BuscarMaximo(buscar);
150                  for (int i = 0; i < fin; i++)
151                  {
152                      int mostrarContadorVenta = Convert.ToInt32(numeroMostrarVenta.Value);
153                      string concepto = matrizConcepto[mostrarContadorVenta, i];
154                      string importe = matrizImporte[mostrarContadorVenta, i];
155                      labelNombreClienteAnterior.Text = "Nombre del cliente:" + matrizNombreCliente[mostrarContadorVenta, i];
156                      labelNumeroPedidoAnterior.Text = "Número de pedido:" + concepto;
157                      labelImportePedidoAnterior.Text = "Importe del pedido:" + importe;
158                      listViewDataOld.Items.Add(new ListViewItem { Text = concepto,
159                                                                                               Text2 = importe });
160                      total = total + float.Parse(importe);
161                  }
162                  labelMostrarTotal.Text = "Total: $" + Convert.ToString(total);
163              }
164          }
165      }
166  }

```

Cliente Importe Concepto

Ingresar Nuevo Pedido Cerrar Pedido Mostrar Pedido

Nombre Cliente: Numero de pedido: Nombre Cliente: Historial de Pedidos 0

Total: 0