

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERIA DE SISTEMAS
INFORMATICOS
PROGRAMACIÓN I
CICLO II-2021



“GUIA DE LABORATORIO 7”

Docente: Inga. Patricia Mejía

Nombres

José Alfonso Escobar Mejía

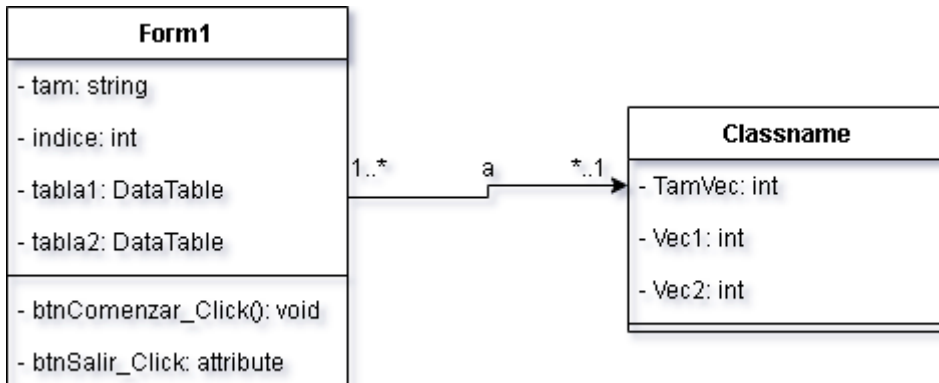
Carné

EM21018

Ejercicio 4 vectores (José Alfonso Escobar Mejía EM21018)

Diseñe un programa que lea dos vectores de igual magnitud o dimensión y luego los compare. Se debe de imprimir los dos vectores y un mensaje que indique si son iguales o no. Recordar que: dos vectores son iguales si cada elemento del primer vector se encuentra exactamente en la misma posición en el segundo.

Diagrama de clases UML:



Diseño del formulario:

The screenshot shows a Windows application window titled "Comparacion de vectores". The window contains the following elements:

- Ingreso de datos** section:
 - A label "Ingresa el tamaño" followed by a numeric spinner box set to "0".
 - Two buttons: "Comprobar" and "Salir".
- Resultados** section:
 - A label "Vector 1" above a text input field.
 - A label "Vector 2" above another text input field.
 - A label "Resultado" at the bottom of the section.

Código de form1:

```
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 Ejercicio4_Guia7
{
    public partial class Form1 : Form
    {
        private DataTable tabla1;
        private DataTable tabla2;
        public Form1()
        {
            InitializeComponent();
        }

        private void btnComenzar_Click(object sender, EventArgs e)
        {
            Vectores magnitud = new Vectores();
            string tam = nudMagnitud.Text;
            int indice;
            indice = int.Parse(tam);
            magnitud.TamVec1 = indice;

            if (magnitud.TamVec1 <= 0)
            {
                epError.SetError(nudMagnitud, "Debe digitar una cantidad mayor a
cero");
            }
            else if (magnitud.TamVec1 > 0)
            {
                epError.Clear();
                tabla1 = new DataTable();
                DataRow row = tabla1.NewRow();
                tabla2 = new DataTable();
                DataRow row1 = tabla2.NewRow();
                int[] vector1 = new int[magnitud.TamVec1];
                int[] vector2 = new int[magnitud.TamVec1];
                int posicion = 0;
                string nombrePosiccion;

                for (int c=1; c <= magnitud.TamVec1; c++)
                {
                    posicion = c - 1;
                    magnitud.Vec11 =
int.Parse(MessageBox.Show("Digite el valor de la
posicion " + (posicion) + " del vector 1"), "Valores del vector 1"));
                    vector1[posicion] = magnitud.Vec11;
                    nombrePosiccion = posicion.ToString();
                    tabla1.Columns.Add(nombrePosiccion);
                }
            }
        }
    }
}
```

```

        dgvVector1.DataSource = tabla1;
        row[nombrePosiccion] = magnitud.Vec11;
    }
    tabla1.Rows.Add(row);
    for (int c=1; c <= magnitud.TamVec1; c++)
    {
        posicion = c - 1;
        magnitud.Vec11 =
int.Parse(MessageBox.Show("Digite el valor de la
posicion " + (posicion) + " del vector 2"), "Valores del vector 2"));
        vector2[posicion] = magnitud.Vec11;
        nombrePosiccion = posicion.ToString();
        tabla2.Columns.Add(nombrePosiccion);
        dgvVector2.DataSource = tabla2;
        row1[nombrePosiccion] = magnitud.Vec11;

    }
    tabla2.Rows.Add(row1);
    for (int c=0; c < vector1.LongLength; c++)
    {
        if (vector1[c] == vector2[c])
        {
            lbResultado.Text = "Los vectores son iguales";
        }
        else if (vector1[c] != vector2[c])
        {
            lbResultado.Text = "Los vectores son diferentes";
            break;
        }
    }
}
nudMagnitud.Text = "0";
}
private void btnSalir_Click(object sender, EventArgs e)
{
    Close();
}
}
}

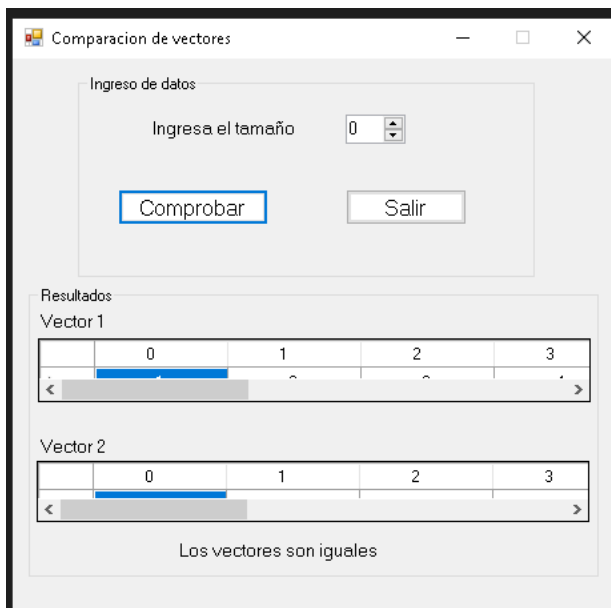
```

Código de Clases:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Ejercicio4_Guia7
{
    class Vectores
    {
        int TamVec;
        int Vec1;
        int Vec2;
        public int TamVec1 { get => TamVec; set => TamVec = value; }
        public int Vec11 { get => Vec1; set => Vec1 = value; }
        public int Vec21 { get => Vec2; set => Vec2 = value; }
    }
}
```

Ejecución del programa:



Ejercicio 3 Herencia (José Alfonso Escobar Mejía EM21018)

Una tienda local necesita un sistema de ventas para sus productos, cada producto tiene los siguientes atributos:

nombre

precio

Los productos que tiene son:

Perecederos, que tienen un tiempo de caducidad

No perecederos: que tiene el atributo "tipo

Cada uno de los productos tendrá una función llamada calcular que calculará el precio

Según las compras hechas:

Para productos no perecederos simplemente se hará la multiplicación de la cantidad de productos llevados por el precio.

Para los productos perecederos se calculará de la siguiente forma:

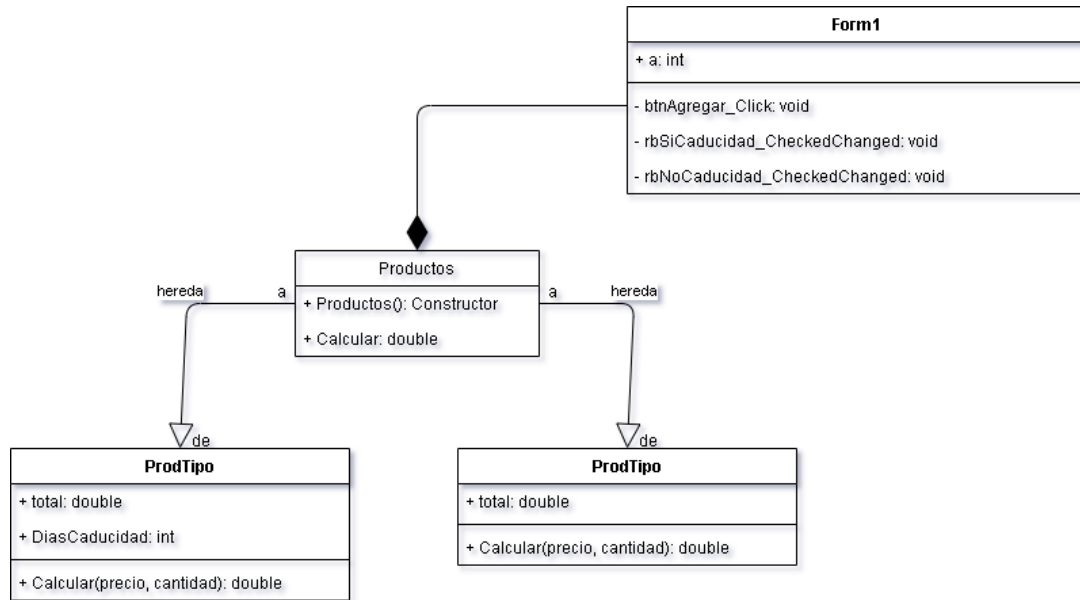
Si hace faltan más de 10 días para que caduque el precio será la mitad

Si hace faltan más de 5 días para que caduque el precio será una tercera parte

Si hace falta un día para que caduque, el precio será una cuarta parte del original.

Mostrar una lista de productos comprados de ambos tipos, junto con su precio y el resultado final.

Diagrama de clases UML:



Diseño del formulario:

Sistema de ventas

Ingreso de productos

Ingrese el nombre del producto

¿El producto tiene fecha de caducidad?
☐ Si ☐ No

Ingrese los días en los que caduca

Ingrese el precio del producto
 \$

Ingrese la cantidad de productos

Precione agregar para agregar a la lista

Lista de productos

	Nombre	Nº	Precio	Fecha de Caducidad	Tipo
*					

Total a pagar

Código de Form1.cs

```
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 Guia_7_Ej_Herencia_3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnAgregar_Click(object sender, EventArgs e)
        {
            int a;
            double b;
            if (rbSiCaducidad.Checked == true || rbNoCaducidad.Checked == true &&
            double.TryParse(tbPrecioProd.Text, out b) && int.TryParse(tbCantidadProd.Text, out
            a) && tbNomProd.Text != "")
            {

```



```

        if (rbSiCaducidad.Checked == true)
        {
            ProdCaduca prodCadusidad = new ProdCaduca();
            prodCadusidad.DiasCaducidad1 = int.Parse(tbCaducaProd.Text);
            tbPagarLista.Text = (double.Parse(tbPagarLista.Text) +
prodCadusidad.Calcular(double.Parse(tbPrecioProd.Text),
int.Parse(tbCantidadProd.Text))).ToString();
            int n = dgvLista.Rows.Add();
            dgvLista.Rows[n].Cells[0].Value = tbNomProd.Text;
            dgvLista.Rows[n].Cells[1].Value = tbCantidadProd.Text;
            dgvLista.Rows[n].Cells[2].Value = "$" + tbPrecioProd.Text;
            dgvLista.Rows[n].Cells[3].Value = "Vence " + tbCaducaProd.Text +
" dias";

            lblTipoProd.Visible = false;
            tbTipoProd.Visible = false;
            tbNomProd.Clear();
            tbCaducaProd.Clear();
            tbTipoProd.Clear();
            tbPrecioProd.Clear();
            tbCantidadProd.Clear();
        }
        if (rbNoCaducidad.Checked == true)
        {
            ProdTipo prodTipo = new ProdTipo();
            tbPagarLista.Text = (int.Parse(tbPagarLista.Text) +
prodTipo.Calcular(double.Parse(tbPrecioProd.Text),
int.Parse(tbCantidadProd.Text))).ToString();
            int n = dgvLista.Rows.Add();
            dgvLista.Rows[n].Cells[0].Value = tbNomProd.Text;
            dgvLista.Rows[n].Cells[1].Value = tbCantidadProd.Text;
            dgvLista.Rows[n].Cells[2].Value = "$" + tbPrecioProd.Text;
            dgvLista.Rows[n].Cells[4].Value = tbTipoProd.Text;
            lblCaducaProd.Visible = false;
            tbCaducaProd.Visible = false;
            tbNomProd.Clear();
            tbCaducaProd.Clear();
            tbTipoProd.Clear();
            tbPrecioProd.Clear();
            tbCantidadProd.Clear();
        }
    }
    else
    {
        MessageBox.Show("No deje nada vacio\r\nPosibles problemas\r\n\r\nLos
dias de caducidad debe ser un numero entero\r\nEl precio de ser un numero\r\nLa
cantidad de productos debe ser un numero entero", "Error de datos");
    }
}

private void rbSiCaducidad_CheckedChanged(object sender, EventArgs e)
{
    if (rbSiCaducidad.Checked==true)
    {
        lblTipoProd.Visible = false;
        tbTipoProd.Visible = false;
        lblCaducaProd.Visible = true;
        tbCaducaProd.Visible = true;
    }
}

```

```

    }
    private void rbNoCaducidad_CheckedChanged(object sender, EventArgs e)
    {
        if (rbNoCaducidad.Checked == true)
        {
            lblTipoProd.Visible = true;
            tbTipoProd.Visible = true;
            lblCaducaProd.Visible = false;
            tbCaducaProd.Visible = false;
        }
    }
}
}
}

```

Código de las Clases

Código de la clase

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Guia_7_Ej_Herencia_3
{
    abstract class Productos
    {
        public Productos()
        {
        }
        public abstract double Calcular(double precio, int cantidad);
    }
}

```

Código de la clase ProdTipo

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```

using System.Text;
using System.Threading.Tasks;

namespace Guia_7_Ej_Herencia_3
{
    class ProdTipo:Productos
    {
        //variables locales de la clase
        double total;
        public override double Calcular(double precio, int cantidad)
        {
            total = precio * cantidad;
            return total;
        }
    }
}

```

Código de la clase ProdCaduca

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Guia_7_Ej_Herencia_3
{
    class ProdCaduca:Productos
    {
        int DiasCaducidad;
        public int DiasCaducidad1 { get => DiasCaducidad; set => DiasCaducidad =
value; }
        //variables locales
        double total;
        public override double Calcular(double precio, int cantidad)
        {
            if (DiasCaducidad>10)
            {
                total = (precio /2)*(cantidad);
            }
            else if (DiasCaducidad>5)
            {
                total = (precio /3) * (cantidad);
            }
            else if (DiasCaducidad>1)
            {
                total = (precio /4) * (cantidad);
            }
            return total;
        }
    }
}

```

Ejecución del programa:

Sistema de ventas

Ingreso de productos

Ingrese el nombre del producto

¿El producto tiene fecha de caducidad?

☐ Si ☒ No

Ingrese el tipo de producto

Ingrese el precio del producto

\$

Ingrese la cantidad de productos

Precione agregar para agregar a la lista

Lista de productos

	Nombre	Nº	Precio	Fecha de Caducidad	Tipo
	Antigripales	3	\$7.00		Medicame...
	Papel higié...	4	\$0.50		Higiene
	Refrescos	4	\$1	Vence 30 ...	
	Pastas	3	\$0.30	Vence 2 días	
	Jamón	4	\$0.50	Vence 5 días	
»»					

Total a pagar