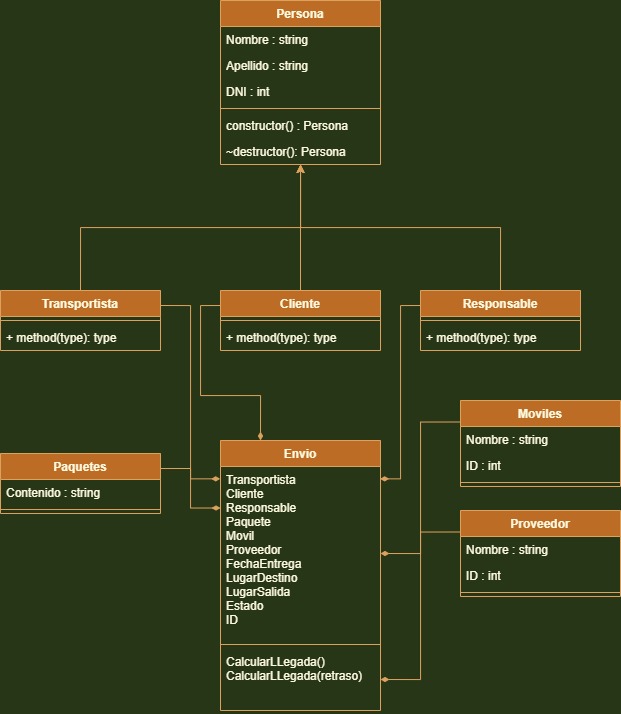


Consigna Tema 11 Sistema de seguimiento de envíos

El sistema de seguimiento de envíos se va a ocupar de recepcionar algún tipo de paquete el cual será remitido a algún tipo de cliente desde algún tipo de proveedor. El sistema de seguimiento de envíos deberá constar de un registro de clientes, de proveedores, registro de móviles que se van a ocupar de la entrega y un cierto orden de entrega dependiendo de la fecha en que se recepciona el paquete. Los paquetes deberán ser entregados en el término de 72 horas por lo cual, en el momento de registrar el envío, automáticamente el sistema debe calcular la fecha de llegada del paquete a destino. El sistema de registración de envío de seguimiento va a contar con los nombres de los responsables, tanto del registro del pedido como así también del registro de la persona que traslada el paquete y de la persona que recepciona el envío. Por cada envío será registrado con el móvil la fecha, lugar de salida, lugar de destino, y el costo que ha de tener ese envío. El sistema permitirá ver por cada Destinatario los envíos realizados y por cada envío los detalles del mismo. Cada vez que se confirma la recepción del paquete el sistema dará por cerrado el proceso en cuestión. Si existe demora el sistema deberá registrar las causas. Puede persistir datos en archivos CSV.

Diagrama de clases (Esta dentro del proyecto en la carpeta Docs)



Clase Clientes

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lastra\_julian\_POO\_Parcial2

{

public class Clientes : Persona

{

// Propiedad de solo lectura

public readonly int ID;

// Constructor

public Clientes(string nombre, string apellido, int dni, int id)

{

Nombre = nombre;

Apellido = apellido;

DNI = dni;

ID = id;

}

}

}

Clase Envio

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lastra\_julian\_POO\_Parcial2

{

public class Envio

{

// Persona que transporta el paquete

public Transportista transportista { get; set; }

// Persona que va a recibir el paquete

public Clientes cliente { get; set; }

// Persona que quiere enviar el paquete

public Proveedor proveedor { get; set; }

// Persona que es responsable de la carga del envio/paquete

public Responsables responsable;

// Movil que va a enviar el paquete

public Moviles movil { get; set; }

// Paquete que se va a trasnportar

public Paquetes paquete { get; set; }

public DateTime FechaLLegada { get; set; }

public string LugarSalida { get; set; }

public string LugarDestino { get; set; }

public string Estado { get; set; } // Recibido, En proceso

public float Costo { get; set; }

public string CausaDemora { get; set; }

public int ID { get; set; }

#region Metodos

// Calculo automatico de la fecha del envio

public DateTime CalcularLLegada()

{

var fechaActual = DateTime.Now;

return fechaActual.AddDays(3);

}

// Metodo Sobrecargado

public DateTime CalcularLLegada(int demora,DateTime fechaAnterior)

{

var fechaActual = fechaAnterior;

return fechaActual.AddDays(demora);

}

#endregion

}

}

Clase Moviles

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lastra\_julian\_POO\_Parcial2

{

public class Moviles

{

public int IDMovil { get; set; }

public string Nombre { get; set; }

}

}

Clase Paquetes

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lastra\_julian\_POO\_Parcial2

{

public class Paquetes

{

public string Contenido { get; set; }

}

}

Clase Persona

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lastra\_julian\_POO\_Parcial2

{

public class Persona

{

public string Nombre { get; set; }

public string Apellido { get; set; }

public int DNI { get; set; }

// Destructor

~Persona()

{

Nombre = "";

Apellido = "";

DNI = 0;

}

}

}

Clase Proveedor

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lastra\_julian\_POO\_Parcial2

{

public class Proveedor

{

public int IDProveedor { get; set; }

public string Nombre { get; set; }

}

}

Clase Responsables

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lastra\_julian\_POO\_Parcial2

{

public class Responsables : Persona

{

}

}

Clase Transportista

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lastra\_julian\_POO\_Parcial2

{

public class Transportista : Persona

{

}

}

Formulario principal “Interfaz”

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace lastra\_julian\_POO\_Parcial2

{

public partial class Interfaz : Form

{

public int counterClientes = 0;

public int counterMoviles = 0;

public int counterProveedores = 0;

public int counterEnvios = 0;

public List<Clientes> listaClientes = new List<Clientes>();

public List<Proveedor> listaProveedores = new List<Proveedor>();

public List<Moviles> listaMoviles = new List<Moviles>();

public List<Responsables> listaResponsables = new List<Responsables>();

public List<Transportista> listaTransportistas = new List<Transportista>();

public List<Paquetes> listaPaquetes = new List<Paquetes>();

public List<Envio> listaEnvios = new List<Envio>();

public Interfaz()

{

InitializeComponent();

}

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

{

}

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

{

// Agregar nuevo cliente a la lista

// Obtengo el ID de cliente

counterClientes += 1;

Clientes cliente = new Clientes(textBox1.Text, textBox2.Text,

Int32.Parse(textBox3.Text), counterClientes);

listaClientes.Add(cliente);

comboBox1.DataSource = null;

comboBox1.DataSource = listaClientes;

comboBox1.DisplayMember = "Apellido";

dataGridView1.DataSource = null;

dataGridView1.DataSource = listaClientes;

}

private void button3\_Click(object sender, EventArgs e)

{

// Agregar nuevo proveedor a la lista

Proveedor proveedor = new Proveedor();

counterProveedores += 1;

proveedor.Nombre = textBox4.Text;

proveedor.IDProveedor = counterProveedores;

listaProveedores.Add(proveedor);

comboBox2.DataSource = null;

comboBox2.DataSource = listaProveedores;

comboBox2.DisplayMember = "Nombre";

}

private void button4\_Click(object sender, EventArgs e)

{

// Agregar nuevo movil a la lista

Moviles movil = new Moviles();

counterMoviles += 1;

movil.Nombre = textBox5.Text;

movil.IDMovil = counterMoviles;

listaMoviles.Add(movil);

comboBox3.DataSource = null;

comboBox3.DataSource = listaMoviles;

comboBox3.DisplayMember = "Nombre";

}

private void button5\_Click(object sender, EventArgs e)

{

//Agregar nuevo responsable a la lista

Responsables responsable = new Responsables();

responsable.Nombre = textBox6.Text;

responsable.Apellido = textBox8.Text;

responsable.DNI = Int32.Parse(textBox7.Text);

listaResponsables.Add(responsable);

comboBox4.DataSource = null;

comboBox4.DataSource = listaResponsables;

comboBox4.DisplayMember = "Apellido";

}

private void button6\_Click(object sender, EventArgs e)

{

//Agregar nuevo transportista a la lista

Transportista transportista = new Transportista();

transportista.Nombre = textBox11.Text;

transportista.Apellido = textBox10.Text;

transportista.DNI = Int32.Parse(textBox9.Text);

listaTransportistas.Add(transportista);

comboBox5.DataSource = null;

comboBox5.DataSource = listaTransportistas;

comboBox5.DisplayMember = "Apellido";

}

private void button7\_Click(object sender, EventArgs e)

{

//Agregar nuevo paquete a la lista

Paquetes paquete = new Paquetes();

paquete.Contenido = textBox17.Text;

listaPaquetes.Add(paquete);

comboBox6.DataSource = null;

comboBox6.DataSource = listaPaquetes;

comboBox6.DisplayMember = "Contenido";

}

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

{

//Agregar nuevo envio a la lista

Envio envio = new Envio();

envio.cliente = (Clientes)comboBox1.SelectedItem;

envio.proveedor = (Proveedor)comboBox2.SelectedItem;

envio.movil = (Moviles)comboBox3.SelectedItem;

envio.responsable = (Responsables)comboBox4.SelectedItem;

envio.transportista = (Transportista)comboBox5.SelectedItem;

envio.paquete = (Paquetes)comboBox6.SelectedItem;

envio.FechaLLegada = envio.CalcularLLegada();

envio.LugarSalida = textBox15.Text;

envio.LugarDestino = textBox14.Text;

envio.Costo = float.Parse(textBox16.Text);

counterEnvios += 1;

envio.ID = counterEnvios;

listaEnvios.Add(envio);

comboBox7.DataSource = null;

comboBox7.DataSource = listaEnvios;

comboBox7.DisplayMember = "ID";

// Lambda + LinQ

// Ordenar los paquetes por orden de recepcion

listView2.Items.Clear();

var listaOrdenEnvio = listaEnvios.OrderBy(e => e.FechaLLegada);

foreach (var envio2 in listaEnvios)

{

listView2.Items.Add(envio2.FechaLLegada.ToString());

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button8\_Click(object sender, EventArgs e)

{

// Evento

// Calculo la cantidad de envio por el destinatario seleccionado

List<Envio> listaEnviosCompletados = new List<Envio>();

listView1.Items.Clear();

if (listaClientes.Count > 0 && listaEnvios.Count > 0)

{

foreach (var envio in listaEnvios)

{

if (envio.cliente == (Clientes)comboBox1.SelectedItem && envio.Estado == "Recibido")

{

listaEnviosCompletados.Add(envio);

listView1.Items.Add(envio.paquete.Contenido);

}

}

if (listaEnviosCompletados.Count == 0)

MessageBox.Show("El destinatario no recibio ningun envio todavia.", "Sistema de Envios",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

MessageBox.Show("El destinatario no recibio ningun envio todavia.", "Sistema de Envios",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void button9\_Click(object sender, EventArgs e)

{

// Le agrego la causa de la demora

var envioActual = (Envio)comboBox7.SelectedItem;

envioActual.CausaDemora = textBox18.Text;

int demoraEnDias = Int32.Parse(textBox19.Text);

// Le agrego los dias de demora al envio

envioActual.FechaLLegada = envioActual.CalcularLLegada(demoraEnDias,envioActual.FechaLLegada);

}

private void button10\_Click(object sender, EventArgs e)

{

// Le cambio el estado para saber que fue entregado

var envioActual = (Envio)comboBox7.SelectedItem;

envioActual.Estado = "Recibido";

}

private void button11\_Click(object sender, EventArgs e)

{

var envioActual = (Envio)comboBox7.SelectedItem;

// Cliente

textBox1.Text = envioActual.cliente.Nombre;

textBox2.Text = envioActual.cliente.Apellido;

textBox3.Text = envioActual.cliente.DNI.ToString();

// Proveedor

textBox4.Text = envioActual.proveedor.Nombre;

// Movil

textBox5.Text = envioActual.movil.Nombre;

// Responsable

textBox6.Text = envioActual.responsable.Nombre;

textBox8.Text = envioActual.responsable.Apellido;

textBox7.Text = envioActual.responsable.DNI.ToString();

// Transportista

textBox11.Text = envioActual.transportista.Nombre;

textBox10.Text = envioActual.transportista.Apellido;

textBox9.Text = envioActual.transportista.DNI.ToString();

// Paquete

textBox17.Text = envioActual.paquete.Contenido;

// Envio

textBox15.Text = envioActual.LugarSalida;

textBox14.Text = envioActual.LugarDestino;

textBox6.Text = envioActual.Costo.ToString();

}

}

}