* Datos de base de datos

create database agendaVirtual

create table agenda(

id int primary key identity(1,1) not null,

nombre varchar(30) not null,

apellido varchar(30),

correo varchar(40),

telefono int not null,

fechaNacimiento varchar(15),

direccion varchar(40)

)

use agendaVirtual

--busca los datos de la tabla dinamicamente ej: escribo j y me trae todos los nombre con j por el parametro buscar

go

create procedure SP\_buscarRegistro

@buscar varchar(20)

as

select \* from agenda

where nombre like @buscar + '%'

--agrega

go

create procedure SP\_agregarRegistro

@nombre varchar(30), @apellido varchar(30), @correo varchar(40), @telefono int, @fechaNacimiento varchar (15), @direccion varchar(40)

as

insert into agenda values(@nombre, @apellido, @correo, @telefono, @fechaNacimiento, @direccion)

--edita 5 campos del cliente

go

create procedure SP\_editarRegistro

@id int,@nombre varchar(30), @apellido varchar(30), @correo varchar(40), @telefono int, @fechaNacimiento varchar (15), @direccion varchar(40)

as

update agenda set nombre = @nombre, apellido = @apellido, correo = @correo, telefono = @telefono, fechaNacimiento = @fechaNacimiento, direccion = @direccion

where id = @id

--elimina una fila mediante el id

go

create procedure SP\_eliminarRegistro

@id int

as

delete agenda

where id = @id

insert into agenda values('Joel','Mojica','el\_joe03@hotmail.com',6558795,'14/03/1994','calle Juan Jose duarte')

select \* from agenda

* Capa Entidad

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace capaEntidad

{

public class E\_agenda

{

private int id;

private string nombre;

private string apellido;

private string correo;

private int telefono;

private string fechaDeNacimiento;

private string dirección;

public int Id { get => id; set => id = value; }

public string Nombre { get => nombre; set => nombre = value; }

public string Apellido { get => apellido; set => apellido = value; }

public string Correo { get => correo; set => correo = value; }

public int Telefono { get => telefono; set => telefono = value; }

public string FechaDeNacimiento { get => fechaDeNacimiento; set => fechaDeNacimiento = value; }

public string Dirección { get => dirección; set => dirección = value; }

}

}

* capa de datos

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Configuration;

using System.Data;

using capaEntidad;

namespace CapaDatos

{

public class D\_agenda

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["sqlcon"].ConnectionString);

SqlCommand comando;

public List<E\_agenda> buscarContacto(string buscar)

{

SqlDataReader leerFilas;

comando = new SqlCommand("SP\_buscarRegistro", con);

comando.CommandType = CommandType.StoredProcedure;

con.Open();

comando.Parameters.AddWithValue("@buscar", con);

leerFilas = comando.ExecuteReader();

List<E\_agenda> listar = new List<E\_agenda>();

while (leerFilas.Read())

{

listar.Add(new E\_agenda

{

Id = leerFilas.GetInt32(0),

Nombre = leerFilas.GetString(1),

Apellido = leerFilas.GetString(2),

Correo = leerFilas.GetString(3),

Telefono = leerFilas.GetInt32(4),

FechaDeNacimiento = leerFilas.GetString(5),

Dirección = leerFilas.GetString(6)

});

}

con.Close();

leerFilas.Close();

return listar;

}

public void agregarContacto(E\_agenda Contacto)

{

comando = new SqlCommand("SP\_agregarRegistro", con);

comando.CommandType = CommandType.StoredProcedure;

con.Open();

comando.Parameters.AddWithValue("@nombre", Contacto.Nombre);

comando.Parameters.AddWithValue("@apellido", Contacto.Apellido);

comando.Parameters.AddWithValue("@correo", Contacto.Correo);

comando.Parameters.AddWithValue("@telefono", Contacto.Telefono);

comando.Parameters.AddWithValue("@fechaNacimiento", Contacto.FechaDeNacimiento);

comando.Parameters.AddWithValue("@direccion", Contacto.Dirección);

comando.ExecuteNonQuery();

con.Close();

}

public void editarContacto(E\_agenda Contacto)

{

comando = new SqlCommand("SP\_editarRegistro", con);

comando.CommandType = CommandType.StoredProcedure;

con.Open();

comando.Parameters.AddWithValue("@id", Contacto.Id);

comando.Parameters.AddWithValue("@nombre", Contacto.Nombre);

comando.Parameters.AddWithValue("@apellido", Contacto.Apellido);

comando.Parameters.AddWithValue("@correo", Contacto.Correo);

comando.Parameters.AddWithValue("@telefono", Contacto.Telefono);

comando.Parameters.AddWithValue("@fechaNacimiento", Contacto.FechaDeNacimiento);

comando.Parameters.AddWithValue("@direccion", Contacto.Dirección);

comando.ExecuteNonQuery();

con.Close();

}

public void eliminarContacto(E\_agenda Contacto)

{

comando = new SqlCommand("SP\_eliminarRegistro", con);

comando.CommandType = CommandType.StoredProcedure;

con.Open();

comando.Parameters.AddWithValue("@id", Contacto.Id);

comando.ExecuteNonQuery();

con.Close();

}

}

}

* capa de negocios

using CapaDatos;

using capaEntidad;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace capaNegocios

{

public class N\_agenda

{

E\_agenda objEntida = new E\_agenda();

D\_agenda objDatos = new D\_agenda();

public List<E\_agenda> buscarRegistroContacto(string buscar)

{

return objDatos.buscarContacto(buscar);

}

public void agregarRegistroContacto(E\_agenda Contacto)

{

objDatos.agregarContacto(Contacto);

}

public void editarRegistroContacto(E\_agenda Contacto)

{

objDatos.editarContacto(Contacto);

}

public void eliminarRegistroContacto(E\_agenda Contacto)

{

objDatos.eliminarContacto(Contacto);

}

}

}

proceso final hacer la presentacion o vista