| class Agenda  {  //Atributos  Contacto[] misContactos;  int indice;  int tamaño;  //Propiedades  public Contacto[] MisContactos { get => misContactos; set => misContactos = value; }  public int Indice { get => indice; }  public int Tamaño { get => tamaño; }  //Constructor  public Agenda(int tamaño)  {  this.tamaño = tamaño;  misContactos = new Contacto[tamaño];  }  //Metodos  public bool AgregarContacto(Contacto contacto)  {  if (indice <= tamaño)  {  misContactos[indice] = contacto;  indice++;  return true;  }  return false;  }  public Contacto BuscarContacto(int nui)  {  Contacto contactoEncontrado = null;  for (int i = 0; i < indice; i++)  {  if (misContactos[i].Niu == nui)  {  contactoEncontrado = misContactos[i];  }  }  return contactoEncontrado;  }  public string MostrarContacto(int indice)  {  return "Nombre: " + misContactos[indice].Nombre + '\n' +  "Telefono: " + misContactos[indice].Telefono + '\n' +  "Direccion: " + misContactos[indice].Direccion + '\n' +  "Email: " + misContactos[indice].Email + '\n' +  "Profesion: " + misContactos[indice].Profesion + '\n' +  "NUI: " + misContactos[indice].Niu + '\n';  }  public string MostrarContacto(Contacto contacto)  {  return "Nombre: " + contacto.Nombre + '\n' +  "Telefono: " + contacto.Telefono + '\n' +  "Direccion: " + contacto.Direccion + '\n' +  "Email: " + contacto.Email + '\n' +  "Profesion: " + contacto.Profesion + '\n' +  "NUI: " + contacto.Niu + '\n';  }  public void GuardarAgenda()  {  StreamWriter archivoEscritura = new StreamWriter("misContactos.txt");  for (int i = 0; i < indice; i++)  {  string infoContacto = misContactos[i].Nombre + ";" +  misContactos[i].Telefono + ";" + misContactos[i].Direccion + ";" +  misContactos[i].Email + ";" + misContactos[i].Profesion + ";" +  misContactos[i].Niu + '\n';  archivoEscritura.Write(infoContacto);  }  archivoEscritura.Close();  }  public void CargarContactos()  {  string linea = "";  string[] datos = new string[6];  StreamReader archivoLectura = new StreamReader("misContactos.txt");  linea = archivoLectura.ReadLine();  while (!string.IsNullOrEmpty(linea))  {  datos = linea.Split(';');  AgregarContacto(new Contacto(datos[0], int.Parse(datos[5]), datos[2],  int.Parse(datos[1]), datos[3], datos[4]));  linea = archivoLectura.ReadLine();  }  archivoLectura.Close();  }  }  class Contacto  {  //atributos  string nombre;  int niu;  int telefono;  string direccion;  string email;  string profesion;  //Constructores  public Contacto()  {  }  public Contacto(string nombre, int niu, int telefono)  {  this.nombre = nombre;  this.niu = niu;  this.telefono = telefono;  }  public Contacto(string nombre, int niu, string direccion, int telefono, string email, string profesion)  {  this.nombre = nombre;  this.niu = niu;  this.direccion = direccion;  this.telefono = telefono;  this.email = email;  this.profesion = profesion;  }  //Propiedades  public string Nombre { get => nombre; set => nombre = value; }  public int Niu { get => niu; set => niu = value; }  public string Direccion { get => direccion; set => direccion = value; }  public int Telefono { get => telefono; set => telefono = value; }  public string Email { get => email; set => email = value; }  public string Profesion { get => profesion; set => profesion = value; }    }  Agenda miAgenda = new Agenda(20);  int op = 1;    while(op!=0)  {  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Agenda de contactos\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine("\n¿Qué acción desea realizar");  Console.WriteLine("1. Para agregar contacto");  Console.WriteLine("2. Para mostrar todos los contactos");  Console.WriteLine("3. Para buscar un contacto por NUI");  Console.WriteLine("4. Para guardar la agenda");  Console.WriteLine("5. Para contactos desde un archivo");  Console.WriteLine("0. Salir");  op = int.Parse(Console.ReadLine());  switch (op)  {  case 1:  {  Console.WriteLine("Ingrese el NUI");  int nui = int.Parse(Console.ReadLine());  var contacto = miAgenda.BuscarContacto(nui);  if (contacto == null)  {  Console.WriteLine("Ingrese el nombre");  string nombre = Console.ReadLine();  Console.WriteLine("Ingrese la direccion");  string direccion = Console.ReadLine();  Console.WriteLine("Ingrese el telefono ");  int telefono = int.Parse(Console.ReadLine());  Console.WriteLine("Ingrese el email");  string email = Console.ReadLine();  Console.WriteLine("Ingrese la profesion u oficio");  string profesion = Console.ReadLine();  bool confirmacion = miAgenda.AgregarContacto(  new Contacto(nombre, nui, direccion, telefono, email, profesion));  if (confirmacion)  Console.WriteLine("Contacto guardo con exito");  else  Console.WriteLine("No es posible guardar el contacto, la agenda está llena");  }  else  Console.WriteLine("Ya existe un contacto con ese NUI");  }  break;  case 2:  for(int i=0; i<miAgenda.Indice; i++)  {  Console.WriteLine("\*\*\*\*\*Contacto {0}\*\*\*\*\*\*\*\*\*\*\*\*\n",i+1);  Console.WriteLine(miAgenda.MostrarContacto(i));  }  break;  case 3:  {  Console.WriteLine("Ingrese el NUI");  int nui = int.Parse(Console.ReadLine());  var contacto = miAgenda.BuscarContacto(nui);  if (contacto != null)  {  Console.WriteLine(miAgenda.MostrarContacto(contacto));  }  else  Console.WriteLine("Contacto no encontrado");  }  break;  case 4:  miAgenda.GuardarAgenda();  Console.WriteLine("Contactos guardados");  break;  case 5:  Console.WriteLine("Contactos cargados");  miAgenda.CargarContactos();  break;  }  Console.WriteLine("Presione una teclas para continuar");  Console.ReadKey();  Console.Clear();  }  as |
| --- |