



# PROGRAMACION ORIENTADA A OBJETOS C#

DESARROLLAR METODOS FALTANTES

Programación I ITDSC3-003-SDN

NOMBRE: Sabrina Amparo Valdez | 2025-1320

FACILITADOR: Starling Alfredo Germosén Reynoso

sección: #1 lunes

carrera: Desarrollo de Software

# Transformar el proyecto "Contactes" a POO o desarrolle su propia solución que cumpla con el requerimiento 1 de "Agenda Personal".

```
C# program
 4 //sabrina amparo valdez | 20251320
 5 Console.WriteLine("Mi Agenda Perrón");
 6 Console.WriteLine("Bienvenido a tu lista de contactos");
 7
 8 AgendaPersonal agenda = new AgendaPersonal();
 9 bool running = true;
10
11 while (running)
12 {
13     Console.WriteLine("1. Agregar Contacto      ");
14     Console.WriteLine("2. Ver Contactos       ");
15     Console.WriteLine("3. Buscar Contactos    ");
16     Console.WriteLine("4. Modificar Contacto  ");
17     Console.WriteLine("5. Eliminar Contacto   ");
18     Console.WriteLine("6. Salir");
19     Console.WriteLine("Elige una opción: ");
20
21     int choice = Convert.ToInt32(Console.ReadLine());
22
23     switch (choice)
24     {
25         case 1:
26             agenda.AgregarContacto();
27             break;
28         case 2:
29             agenda.VerContactos();
30             break;
31         case 3:
32             agenda.BuscarContacto();
33             break;
34         case 4:
35             agenda.ModificarContacto();
36             break;
37         case 5:
38             agenda.EliminarContacto();
39             break;
40         case 6:
41             running = false;
42             break;
43         default:
44             Console.WriteLine("Opción no válida");
45     }
46 }
47
48
49 // Clase que representa un Contacto
50 public class Contacto
51 {
52     public int Id { get; set; }
53     public string Nombre { get; set; }
54     public string Telefono { get; set; }
55     public string Email { get; set; }
56     public string Direccion { get; set; }
57
58     public Contacto(int id, string nombre, string telefono, string email, string direccion)
59     {
60         Id = id;
61         Nombre = nombre;
62         Telefono = telefono;
63         Email = email;
64         Direccion = direccion;
65     }
66
67     public void MostrarInformacion()
68     {
69         Console.WriteLine($"ID: {Id}");
70         Console.WriteLine($"Nombre: {Nombre}");
71         Console.WriteLine($"Teléfono: {Telefono}");
72         Console.WriteLine($"Email: {Email}");
73         Console.WriteLine($"Dirección: {Direccion}");
74     }
75
76     public void MostrarEnLista()
77     {
78         Console.WriteLine($"{Id,-4} {Nombre,-15} {Telefono,-15} {Email,-20} {Direccion,-20}");
79     }
80 }
```

```
C# program
 44
 45
 46
 47
 48
 49 // Clase que representa un Contacto
50 public class Contacto
51 {
52     public int Id { get; set; }
53     public string Nombre { get; set; }
54     public string Telefono { get; set; }
55     public string Email { get; set; }
56     public string Direccion { get; set; }
57
58     public Contacto(int id, string nombre, string telefono, string email, string direccion)
59     {
60         Id = id;
61         Nombre = nombre;
62         Telefono = telefono;
63         Email = email;
64         Direccion = direccion;
65     }
66
67     public void MostrarInformacion()
68     {
69         Console.WriteLine($"ID: {Id}");
70         Console.WriteLine($"Nombre: {Nombre}");
71         Console.WriteLine($"Teléfono: {Telefono}");
72         Console.WriteLine($"Email: {Email}");
73         Console.WriteLine($"Dirección: {Direccion}");
74     }
75
76     public void MostrarEnLista()
77     {
78         Console.WriteLine($"{Id,-4} {Nombre,-15} {Telefono,-15} {Email,-20} {Direccion,-20}");
79     }
80 }
```

```
// Clase que gestiona la agenda de contactos
3 referencias
public class AgendaPersonal
{
    private List<Contacto> contactos;
    private int siguienteId;

    1 referencia
    public AgendaPersonal()
    {
        contactos = new List<Contacto>();
        siguienteId = 1;
    }

    1 referencia
    public void AgregarContacto()
    {
        Console.WriteLine("Vamos a agregar ese contacto que te trae loco.");

        Console.Write("Digite el Nombre: ");
        var nombre = Console.ReadLine();

        Console.Write("Digite el Teléfono: ");
        var telefono = Console.ReadLine();

        Console.Write("Digite el Email: ");
        var email = Console.ReadLine();

        Console.Write("Digite la dirección: ");
        var direccion = Console.ReadLine();

        Contacto nuevoContacto = new Contacto(siguienteId, nombre, telefono, email, direccion);
        contactos.Add(nuevoContacto);
        siguienteId++;
    }

    Console.WriteLine("Contacto agregado exitosamente!\n");
}
}
```

```
117     3 referencias
118     public void VerContactos()
119     {
120         if (contactos.Count == 0)
121         {
122             Console.WriteLine("No hay contactos en la agenda.\n");
123             return;
124         }
125
126         Console.WriteLine("Id      Nombre          Teléfono          Email           Dirección");
127         Console.WriteLine("-----");
128
129         foreach (var contacto in contactos)
130         {
131             contacto.MostrarEnLista();
132         }
133         Console.WriteLine();
134     }

135     1 referencia
136     public void BuscarContacto()
137     {
138         if (contactos.Count == 0)
139         {
140             Console.WriteLine("No hay contactos en la agenda.\n");
141             return;
142         }
143
144         Console.WriteLine("Digite un Id de Contacto Para Mostrar");
145         int idSeleccionado = Convert.ToInt32(Console.ReadLine());
146
147         Contacto contacto = contactos.FirstOrDefault(c => c.Id == idSeleccionado);
148
149         if (contacto != null)
150         {
151             Console.WriteLine("\nInformación del contacto:");
152             Console.WriteLine("-----");
153             contacto.MostrarInformacion();
154         }
155         else
156         {
157             Console.WriteLine("Contacto no encontrado.");
158         }
159     }
}
```

```
18 }  
19 }  
20  
1 referencia  
2 public void ModificarContacto()  
3 {  
4     if (contactos.Count == 0)  
5     {  
6         Console.WriteLine("No hay contactos en la agenda.\n");  
7         return;  
8     }  
9  
9     VerContactos();  
0     Console.WriteLine("Digite un Id de Contacto Para Editar");  
1     int idSeleccionado = Convert.ToInt32(Console.ReadLine());  
2  
3     Contacto contacto = contactos.FirstOrDefault(c => c.Id == idSeleccionado);  
4  
5     if (contacto != null)  
6     {  
7         Console.Write($"El nombre es: {contacto.Nombre}, Digue el Nuevo Nombre: ");  
8         contacto.Nombre = Console.ReadLine();  
9  
10        Console.Write($"El Teléfono es: {contacto.Telefono}, Digue el Nuevo Teléfono: ");  
11        contacto.Telefono = Console.ReadLine();  
12  
13        Console.Write($"El Email es: {contacto.Email}, Digue el Nuevo Email: ");  
14        contacto.Email = Console.ReadLine();  
15  
16        Console.WriteLine($"La dirección es: {contacto.Direccion}, Digue la nueva dirección: ");  
17        contacto.Direccion = Console.ReadLine();  
18  
19        Console.WriteLine("Contacto modificado exitosamente!\n");  
20    }  
21    else  
22    {  
23        Console.WriteLine("Contacto no encontrado.\n");  
24    }  
25 }  
26  
27 }  
28  
29 }  
30 }
```

```
197 }  
198 }  
199 }  
200 }  
201 }  
202 }  
203 }  
204 }  
205 }  
206 }  
207 }  
208 }  
209 }  
210 }  
211 }  
212 }  
213 }  
214 }  
215 }  
216 }  
217 }  
218 }  
219 }  
220 }  
221 }  
222 }  
223 }  
224 }  
225 }  
226 }  
227 }  
228 }  
229 }  
230 }
```



```
AgendaPersonal.cs
```

```
// Método adicional para búsqueda por nombre (puede ser extendido)
0 referencias
public void BuscarPorNombre(string nombre)
{
    var resultados = contactos.Where(c => c.Nombre.Contains(nombre, StringComparison.OrdinalIgnoreCase)).ToList();

    if (resultados.Count > 0)
    {
        Console.WriteLine($"Resultados para '{nombre}':");
        foreach (var contacto in resultados)
        {
            contacto.MostrarInformación();
            Console.WriteLine();
        }
    }
    else
    {
        Console.WriteLine($"No se encontraron contactos con el nombre '{nombre}'");
    }
}
```

```
Elige una opción: 2
No hay contactos en la agenda.

1. Agregar Contacto      2. Ver Contactos      3. Buscar Contactos      4. Modificar Contacto      5. Eliminar Contacto      6. Salir
Elige una opción: 1
Vamos a agregar ese contacto que te trae loco.
Digite el Nombre: hayden
Digite el Teléfono: 8097653421
Digite el Email: haydenc@gmail.com
Digite la dirección: anacaona juan pablo II #5 parte atras
Contacto agregado exitosamente!

1. Agregar Contacto      2. Ver Contactos      3. Buscar Contactos      4. Modificar Contacto      5. Eliminar Contacto      6. Salir
Elige una opción: 3
Digite un Id de Contacto Para Mostrar
1

Información del contacto:
-----
ID: 1
Nombre: hayden
Teléfono: 8097653421
Email: haydenc@gmail.com
Dirección: anacaona juan pablo II #5 parte atras

1. Agregar Contacto      2. Ver Contactos      3. Buscar Contactos      4. Modificar Contacto      5. Eliminar Contacto      6. Salir
Elige una opción:
```