



ESTRUCTURAS DE DATOS

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

COMPLETE LOS MÉTODOS DE LA CLASE "CONTACTES" O DESARROLLE SU PROPIA SOLUCIÓN QUE CUMPLA CON EL REQUERIMIENTO 1 DEL PROYECTO "AGENDA PERSONAL".

```
4 //sabrina amparo valdez 20251320
5 0 referencias
6 class Program
7 {
8     0 referencias
9     static void Main(string[] args)
10    {
11        Console.WriteLine("Mi Agenda Perrón");
12        Console.WriteLine("Bienvenido a tu lista de contactos");
13
14        Dictionary<int, string> names = new Dictionary<int, string>();
15        Dictionary<int, string> phones = new Dictionary<int, string>();
16        Dictionary<int, string> emails = new Dictionary<int, string>();
17        List<string> addresses = new List<string>();
18        List<int> ids = new List<int>();
19
20        bool running = true;
21        while (running)
22        {
23            Console.WriteLine("\n--- MENÚ PRINCIPAL ---");
24            Console.Write("1. Agregar Contacto      ");
25            Console.Write("2. Ver Contactos      ");
26            Console.Write("3. Buscar Contactos      ");
27            Console.Write("4. Modificar Contacto      ");
28            Console.Write("5. Eliminar Contacto      ");
29            Console.WriteLine("6. Salir");
30            Console.Write("Elige una opción: ");
31
32            try
33            {
34                int choice = Convert.ToInt32(Console.ReadLine());
35
36                switch (choice)
37                {
38                    case 1:
39                        AddContact(ref names, ref phones, ref emails, ref addresses, ref ids);
40                        break;
41                    case 2:
42                        ViewContacts(ref names, ref phones, ref emails, ref addresses, ref ids);
43                        break;
44                    case 3:
45                        SearchContact(ref names, ref phones, ref emails, ref addresses, ref ids);
46                        break;
47                    case 4:
48                        EditContact(ref names, ref phones, ref emails, ref addresses, ref ids);
49                        break;
50                    case 5:
51                        DeleteContact(ref names, ref phones, ref emails, ref addresses, ref ids);
52                        break;
53                    case 6:
54                        running = false;
55                        Console.WriteLine("¡Hasta luego!");
56                        break;
57                    default:
58                        Console.WriteLine("Opción no válida");
59                        break;
60                }
61            }
62        }
63    }
64 }
```

```
projecto contactos    ▾ Program    ▾ Main(string[] args)
59        }
60        catch (FormatException)
61        {
62            Console.WriteLine("Error: Por favor ingresa un número válido.");
63        }
64    }
65}
66
67    1 referencia
68 static void AddContact(ref Dictionary<int, string> names, ref Dictionary<int, string> phones, ref Dictionary<int, string> emails, ref List<string> addresses, ref List<int> ids)
69 {
70     Console.WriteLine("\n--- AGREGAR CONTACTO ---");
71     Console.WriteLine("Vamos a agregar ese contacto que te trae loco.");
72
73     try
74     {
75         int id = ids.Count > 0 ? ids.Max() + 1 : 1;
76         ids.Add(id);
77
78         Console.Write("Digite el Nombre: ");
79         var name = Console.ReadLine();
80         if (string.IsNullOrWhiteSpace(name))
81         {
82             Console.WriteLine("Error: El nombre no puede estar vacío.");
83             ids.Remove(id);
84             return;
85         }
86         names.Add(id, name);
87
88         Console.Write("Digite el Teléfono: ");
89         var telefono = Console.ReadLine();
90         phones.Add(id, telefono);
91
92         Console.Write("Digite el Email: ");
93         var email = Console.ReadLine();
94         emails.Add(id, email);
95
96         Console.Write("Digite la dirección: ");
97         var address = Console.ReadLine();
98         addresses.Add(address);
99
100        Console.WriteLine("|Contacto agregado exitosamente!");
101    }
102    catch (Exception ex)
103    {
104        Console.WriteLine($"Error al agregar contacto: {ex.Message}");
105    }
106
107    4 referencias
108 static void ViewContacts(ref Dictionary<int, string> names, ref Dictionary<int, string> phones, ref Dictionary<int, string> emails, ref List<string> addresses, ref List<int> ids)
109 {
110     Console.WriteLine("\n--- LISTA DE CONTACTOS ---");
111 }
```

```
100
101
102
103
104
105
106
107
108
109     Console.WriteLine("\n--- LISTA DE CONTACTOS ---");
110
111     if (ids.Count == 0)
112     {
113         Console.WriteLine("No hay contactos en la agenda.");
114         return;
115     }
116
117     Console.WriteLine("Id\tNombre\tTeléfono\tEmail\tDirección");
118     Console.WriteLine("-----");
119
120     foreach (var id in ids)
121     {
122         if (names.ContainsKey(id) && phones.ContainsKey(id) && emails.ContainsKey(id))
123         {
124             string dirección = id <= addresses.Count ? addresses[id - 1] : "No especificada";
125             Console.WriteLine($"{id}\t{names[id]}\t{phones[id]}\t{emails[id]}\t{dirección}");
126         }
127     }
128 }
129
130 // referencia
131 static void EditContact(ref Dictionary<int, string> names, ref Dictionary<int, string> phones, ref Dictionary<int, string> emails, ref List<string> addresses, ref List<int> ids)
132 {
133     Console.WriteLine("\n--- MODIFICAR CONTACTO ---");
134
135     if (ids.Count == 0)
136     {
137         Console.WriteLine("No hay contactos para modificar.");
138         return;
139     }
140
141     ViewContacts(ref names, ref phones, ref emails, ref addresses, ref ids);
142
143     try
144     {
145         Console.Write("Digite el Id de Contacto Para Editar: ");
146         int idSeleccionado = Convert.ToInt32(Console.ReadLine());
147
148         if (!ids.Contains(idSeleccionado))
149         {
150             Console.WriteLine("Error: El ID especificado no existe.");
151             return;
152         }
153
154         var nombreSeleccionado = names[idSeleccionado];
155         var teléfonoSeleccionado = phones[idSeleccionado];
156         var emailSeleccionado = emails[idSeleccionado];
157         string direcciónSeleccionada = addresses[idSeleccionado - 1];
158
159         Console.WriteLine($"El nombre actual es: {nombreSeleccionado}, Digite el Nuevo Nombre: ");
160         var nuevoNombre = Console.ReadLine();
161         if (!string.IsNullOrWhiteSpace(nuevoNombre))
162         {
163             names[idSeleccionado] = nuevoNombre;
164         }
165     }
166 }
```

```
# proyecto contactes
165     Console.WriteLine($"El Teléfono actual es: {telefonoSeleccionado}, Dígame el Nuevo Teléfono: ");
166     var nuevoTelefono = Console.ReadLine();
167     phones[idSeleccionado] = nuevoTelefono;
168
169     Console.WriteLine($"El Email actual es: {emailSeleccionado}, Dígame el Nuevo Email: ");
170     var nuevoEmail = Console.ReadLine();
171     emails[idSeleccionado] = nuevoEmail;
172
173     Console.WriteLine($"La dirección actual es: {direccionSeleccionada}, Dígame la nueva dirección: ");
174     var nuevaDireccion = Console.ReadLine();
175     addresses[idSeleccionado - 1] = nuevaDireccion;
176
177     Console.WriteLine("Contacto modificado exitosamente!");
178 }
179 catch (FormatException)
180 {
181     Console.WriteLine("Error: Por favor ingresa un ID válido.");
182 }
183 catch (Exception ex)
184 {
185     Console.WriteLine($"Error al modificar contacto: {ex.Message}");
186 }
187 }
188
189 // referencia
190 static void DeleteContact(ref Dictionary<int, string> names, ref Dictionary<int, string> phones, ref Dictionary<int, string> emails, ref List<string> addresses, ref List<int> ids)
191 {
192     Console.WriteLine("\n--- ELIMINAR CONTACTO ---");
193
194     if (ids.Count == 0)
195     {
196         Console.WriteLine("No hay contactos para eliminar.");
197         return;
198     }
199
200     ViewContacts(ref names, ref phones, ref emails, ref addresses, ref ids);
201
202     try
203     {
204         Console.Write("Dígame el Id de Contacto Para Eliminar: ");
205         int idSeleccionado = Convert.ToInt32(Console.ReadLine());
206
207         if (!ids.Contains(idSeleccionado))
208         {
209             Console.WriteLine("Error: El ID especificado no existe.");
210             return;
211         }
212
213         Console.WriteLine($"¿Está seguro que desea eliminar el contacto '{names[idSeleccionado]}?'");
214         Console.WriteLine("1. Sí, 2. No: ");
215         int opcion = Convert.ToInt32(Console.ReadLine());
216
217         if (opcion == 1)
218         {
219             names.Remove(idSeleccionado);
220             phones.Remove(idSeleccionado);
221             emails.Remove(idSeleccionado);
222             addresses.RemoveAt(idSeleccionado - 1);
223             ids.Remove(idSeleccionado);
224
225             Console.WriteLine("Contacto eliminado exitosamente!");
226         }
227     }
228     catch (Exception ex)
229     {
230         Console.WriteLine($"Error al eliminar contacto: {ex.Message}");
231     }
232 }
```

projecto contactes

Program

EditContact(ref Dictionary<int, string> names,

```

215     if (opcion == 1)
216     {
217         names.Remove(idSeleccionado);
218         phones.Remove(idSeleccionado);
219         emails.Remove(idSeleccionado);
220         addresses.RemoveAt(idSeleccionado - 1);
221         ids.Remove(idSeleccionado);
222
223         Console.WriteLine("¡Contacto eliminado exitosamente!");
224     }
225     else
226     {
227         Console.WriteLine("Eliminación cancelada.");
228     }
229 }
230 catch (FormatException)
231 {
232     Console.WriteLine("Error: Por favor ingresa un número válido.");
233 }
234 catch (Exception ex)
235 {
236     Console.WriteLine($"Error al eliminar contacto: {ex.Message}");
237 }
238 }

1 referencia
static void SearchContact(ref Dictionary<int, string> names, ref Dictionary<int, string> phones, ref Dictionary<int, string> emails, ref List<string> addresses, ref List<int> ids)
{
    Console.WriteLine("\n--- BUSCAR CONTACTO ---");

    if (ids.Count == 0)
    {
        Console.WriteLine("No hay contactos para buscar.");
        return;
    }

    Console.WriteLine("1. Buscar por ID");
    Console.WriteLine("2. Buscar por Nombre");
    Console.WriteLine("Elige una opción de búsqueda: ");

    try
    {
        int opcionBusqueda = Convert.ToInt32(Console.ReadLine());

        if (opcionBusqueda == 1)
        {
            ViewContacts(ref names, ref phones, ref emails, ref addresses, ref ids);
            Console.Write("Digite el Id de Contacto Para Mostrar: ");
            int idSeleccionado = Convert.ToInt32(Console.ReadLine());

            if (ids.Contains(idSeleccionado))
            {
                MostrarDetalleContacto(idSeleccionado, ref names, ref phones, ref emails, ref addresses);
            }
            else

```

```

269             MostrarDetalleContacto(idSeleccionado, ref names, ref phones, ref emails, ref addresses),
270         }
271     else
272     {
273         Console.WriteLine("Error: El ID especificado no existe.");
274     }
275     else if (opcionBusqueda == 2)
276     {
277         Console.Write("Digite el nombre o parte del nombre a buscar: ");
278         string nombreBuscado = Console.ReadLine().ToLower();

279         bool encontrado = false;
280         Console.WriteLine("\n--- RESULTADOS DE BÚSQUEDA ---");

281         foreach (var id in ids)
282         {
283             if (names.ContainsKey(id) && names[id].ToLower().Contains(nombreBuscado))
284             {
285                 MostrarDetalleContacto(id, ref names, ref phones, ref emails, ref addresses);
286                 encontrado = true;
287             }
288         }

289         if (!encontrado)
290         {
291             Console.WriteLine("No se encontraron contactos con ese nombre.");
292         }
293     }
294     else
295     {
296         Console.WriteLine("Opción de búsqueda no válida.");
297     }
298 }
299 catch (FormatException)
300 {
301     Console.WriteLine("Error: Por favor ingresa un número válido.");
302 }
303 }

2 referencias
static void MostrarDetalleContacto(int id, ref Dictionary<int, string> names, ref Dictionary<int, string> phones, ref Dictionary<int, string> emails, ref List<string> addresses)
{
    Console.WriteLine($"\\n--- DETALLES DEL CONTACTO ID: {id} ---");
    Console.WriteLine($"Nombre: {names[id]}");
    Console.WriteLine($"Teléfono: {phones[id]}");
    Console.WriteLine($"Email: {emails[id]}");
    Console.WriteLine($"Dirección: {addresses[id - 1]}");
    Console.WriteLine("-----");
}

```

```
Mi Agenda Perrón
Bienvenido a tu lista de contactos

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

--- LISTA DE CONTACTOS ---
No hay contactos en la agenda.

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

--- AGREGAR CONTACTO ---
Vamos a agregar ese contacto que te trae loco.
Digite el Nombre: sabrina
Digite el Teléfono: 8095555432
Digite el Email: amparovsbrina@gmail.com
Digite la dirección: la loma #12 parte atras
¡Contacto agregado exitosamente!

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

--- MODIFICAR CONTACTO ---

--- LISTA DE CONTACTOS ---
Id      Nombre      Teléfono      Email      Dirección
-----
1       sabrina     8095555432     amparovsbrina@gmail.com     la loma #12 parte atras
Digite el Id de Contacto Para Editar: 1
El nombre actual es: sabrina, Digite el Nuevo Nombre: amparo
El Teléfono actual es: 8095555432, Digite el Nuevo Teléfono:
```

EN ESTE CASO, AGREGUÉ MEJORAS AL CÓDIGO PARA HACERLO MÁS ROBUSTO Y FÁCIL DE USAR, INCLUYENDO MANEJO DE ERRORES CON TRY-CATCH PARA EVITAR QUE EL PROGRAMA SE CIERRE PUES INESPERADAMENTE, TAMBÍEN VALIDACIONES QUE VERIFICA SI EXISTEN CONTACTOS ANTES DE REALIZAR OPERACIONES, UNA FUNCIÓN DE BÚSQUEDA MEJORADA QUE PERMITE ENCONTRAR CONTACTOS TANTO POR ID COMO POR NOMBRE, Y MENSAJES MÁS CLAROS QUE EN MI EXPERIENCIA PUES ME HAN AYUDADO DURANTE TODO EL PROCESO.