



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 }
```

```

108 // proyecto contactes
109 // Program
110 // EditContact(ref Dictionary<int, string> names, ref Dictionary<int, string> phones, ref Dictionary<int, string> emails, ref List<string> addresses, ref List<int> ids)
111
112     Console.WriteLine("\n--- LISTA DE CONTACTOS ---");
113
114     if (ids.Count == 0)
115     {
116         Console.WriteLine("No hay contactos en la agenda.");
117         return;
118     }
119
120     Console.WriteLine("Id\tNombre\tTeléfono\tEmail\tDirección");
121     Console.WriteLine("-----");
122
123     foreach (var id in ids)
124     {
125         if (names.ContainsKey(id) && phones.ContainsKey(id) && emails.ContainsKey(id))
126         {
127             string dirección = id <= addresses.Count ? addresses[id - 1] : "No especificada";
128             Console.WriteLine($"{id}\t{names[id]}\t{phones[id]}\t{emails[id]}\t{dirección}");
129         }
130     }
131
132     static void EditContact(ref Dictionary<int, string> names, ref Dictionary<int, string> phones, ref Dictionary<int, string> emails, ref List<string> addresses, ref List<int> ids)
133     {
134         Console.WriteLine("\n--- MODIFICAR CONTACTO ---");
135
136         if (ids.Count == 0)
137         {
138             Console.WriteLine("No hay contactos para modificar.");
139             return;
140         }
141
142         ViewContacts(ref names, ref phones, ref emails, ref addresses, ref ids);
143
144         try
145         {
146             Console.Write("Digite el Id de Contacto Para Editar: ");
147             int idSeleccionado = Convert.ToInt32(Console.ReadLine());
148
149             if (!ids.Contains(idSeleccionado))
150             {
151                 Console.WriteLine("Error: El ID especificado no existe.");
152                 return;
153             }
154
155             var nombreSeleccionado = names[idSeleccionado];
156             var teléfonoSeleccionado = phones[idSeleccionado];
157             var emailSeleccionado = emails[idSeleccionado];
158             string direcciónSeleccionada = addresses[idSeleccionado - 1];
159
160             Console.WriteLine($"El nombre actual es: {nombreSeleccionado}, Digite el Nuevo Nombre: ");
161             var nuevoNombre = Console.ReadLine();
162             if (!string.IsNullOrWhiteSpace(nuevoNombre))
163             {
164                 names[idSeleccionado] = nuevoNombre;
165             }
166         }
167
168         catch (FormatException)
169         {
170             Console.WriteLine("Error: Por favor ingresa un ID válido.");
171         }
172         catch (Exception ex)
173         {
174             Console.WriteLine($"Error al modificar contacto: {ex.Message}");
175         }
176
177     }
178
179     static void DeleteContact(ref Dictionary<int, string> names, ref Dictionary<int, string> phones, ref Dictionary<int, string> emails, ref List<string> addresses, ref List<int> ids)
180     {
181         Console.WriteLine("\n--- ELIMINAR CONTACTO ---");
182
183         if (ids.Count == 0)
184         {
185             Console.WriteLine("No hay contactos para eliminar.");
186             return;
187         }
188
189         ViewContacts(ref names, ref phones, ref emails, ref addresses, ref ids);
190
191         try
192         {
193             Console.Write("Digite el Id de Contacto Para Eliminar: ");
194             int idSeleccionado = Convert.ToInt32(Console.ReadLine());
195
196             if (!ids.Contains(idSeleccionado))
197             {
198                 Console.WriteLine("Error: El ID especificado no existe.");
199                 return;
200             }
201
202             Console.WriteLine($"Está seguro que desea eliminar el contacto '{names[idSeleccionado]}?'");
203             Console.WriteLine("1. Sí, 2. No: ");
204             int opción = Convert.ToInt32(Console.ReadLine());
205
206             if (opción == 1)
207             {
208                 names.Remove(idSeleccionado);
209                 phones.Remove(idSeleccionado);
210                 emails.Remove(idSeleccionado);
211                 addresses.RemoveAt(idSeleccionado - 1);
212             }
213
214         }
215     }

```

```

C# proyecto contactes                                ▾ Program                                     ▾ EditContact(ref Dictionary<int, string> names,
215
216         if (opcion == 1)
217         {
218             names.Remove(idSeleccionado);
219             phones.Remove(idSeleccionado);
220             emails.Remove(idSeleccionado);
221             addresses.RemoveAt(idSeleccionado - 1);
222             ids.Remove(idSeleccionado);
223
224             Console.WriteLine("¡Contacto eliminado exitosamente!");
225         }
226         else
227         {
228             Console.WriteLine("Eliminación cancelada.");
229         }
230     }
231     catch (FormatException)
232     {
233         Console.WriteLine("Error: Por favor ingresa un número válido.");
234     }
235     catch (Exception ex)
236     {
237         Console.WriteLine($"Error al eliminar contacto: {ex.Message}");
238     }
239 }
240
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)
241 {
242     Console.WriteLine("\n--- BUSCAR CONTACTO ---");
243
244     if (ids.Count == 0)
245     {
246         Console.WriteLine("No hay contactos para buscar.");
247         return;
248     }
249
250     Console.WriteLine("1. Buscar por ID");
251     Console.WriteLine("2. Buscar por Nombre");
252     Console.WriteLine("Elige una opción de búsqueda: ");
253
254     try
255     {
256         int opcionBusqueda = Convert.ToInt32(Console.ReadLine());
257
258         if (opcionBusqueda == 1)
259         {
260             ViewContacts(ref names, ref phones, ref emails, ref addresses, ref ids);
261             Console.Write("Digite el Id de Contacto Para Mostrar: ");
262             int idSeleccionado = Convert.ToInt32(Console.ReadLine());
263
264             if (ids.Contains(idSeleccionado))
265             {
266                 MostrarDetalleContacto(idSeleccionado, ref names, ref phones, ref emails, ref addresses);
267             }
268             else
269
270                 MostrarDetalleContacto(idSeleccionado, ref names, ref phones, ref emails, ref addresses);
271             else
272                 Console.WriteLine("Error: El ID especificado no existe.");
273         }
274         else if (opcionBusqueda == 2)
275         {
276             Console.Write("Digite el nombre o parte del nombre a buscar: ");
277             string nombreBuscado = Console.ReadLine().ToLower();
278
279             bool encontrado = false;
280             Console.WriteLine("\n--- RESULTADOS DE BÚSQUEDA ---");
281
282             foreach (var id in ids)
283             {
284                 if (names.ContainsKey(id) && names[id].ToLower().Contains(nombreBuscado))
285                 {
286                     MostrarDetalleContacto(id, ref names, ref phones, ref emails, ref addresses);
287                     encontrado = true;
288                 }
289             }
290
291             if (!encontrado)
292             {
293                 Console.WriteLine("No se encontraron contactos con ese nombre.");
294             }
295             else
296             {
297                 Console.WriteLine("Opción de búsqueda no válida.");
298             }
299         }
300     }
301     catch (FormatException)
302     {
303         Console.WriteLine("Error: Por favor ingresa un número válido.");
304     }
305 }
306
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)
307 {
308     Console.WriteLine($"\\n--- DETALLES DEL CONTACTO ID: {id} ---");
309     Console.WriteLine($"Nombre: {names[id]}");
310     Console.WriteLine($"Teléfono: {phones[id]}");
311     Console.WriteLine($"Email: {emails[id]}");
312     Console.WriteLine($"Dirección: {addresses[id - 1]}");
313     Console.WriteLine("-----");
314 }
315
316 }

```

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, TAMBIÉN VALIDACIONES QUE VERIFICAN 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.