

```
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5 using System.Security.Cryptography;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace Proyecto1
10 {
11     internal class Gestion_de_Matricula
12     {
13         static List<Estudiante> estudiantes = new List<Estudiante>();
14
15         static void Main(string[] args)
16         {
17             while (true)
18             {
19                 Console.Clear();
20                 Console.WriteLine("Hola, bienvenido al menu de Gestion de ➤
21                     Matricula de Estudiantes - Liceo Regional de Flores");
22                 Console.WriteLine("1- Crear Matricula de Estudiante");
23                 Console.WriteLine("2- Consultar Matricula de Estudiante");
24                 Console.WriteLine("3- Modificar datos de Matricula de ➤
25                     Estudiante");
26                 Console.WriteLine("4- Salir");
27                 Console.WriteLine("Por favor selecciona una de las ➤
28                     siguientes opciones");
29
30                 string opcion = Console.ReadLine();
31
32                 switch (opcion)
33                 {
34                     case "1":
35                         Crear();
36                         break;
37                     case "2":
38                         Consultar();
39                         break;
40                     case "3":
41                         Modificar();
42                         break;
43                     case "4":
44                         return;
45                     default:
46                         Console.WriteLine("Opcion no valida, digite del 1 ➤
47                             al 4");
48                         break;
49                 }
50             }
51         }
52     }
53 }
```

```
46     }
47 }
48
49 static void Crear()
50 {
51     Console.WriteLine("Creacion de Matricula de Estudiante");
52
53     Console.WriteLine("Digite nombre y apellidos del estudiante");
54     string nombre_y_apellidos_estudiante = Console.ReadLine();
55
56     Console.WriteLine("Digite el numero de cedula del estudiante");
57     string numero_cedula_estudiante = Console.ReadLine();
58
59     Console.WriteLine("Digite nombre y apellidos del encargado legal");
60     string nombre_y_apellidos_encargado_legal = Console.ReadLine();
61
62     Console.WriteLine("Digite numero de cedula del encargado legal");
63     string numero_cedula_encargado_legal = Console.ReadLine();
64
65     Console.WriteLine("Digite la direccion de residencia");
66     string direccion_estudiante = Console.ReadLine();
67
68     Console.WriteLine("Digite el Numero de Telefono");
69     string numero_telefonico = Console.ReadLine();
70
71     Console.WriteLine("Digite los datos del lugar de trabajo del encargado legal");
72     string datos_trabjo_encargado = Console.ReadLine();
73
74     Console.WriteLine("Digite el Grado lectivo por matricular (numero del 7 al 11)");
75     int grado_a_matricular = Convert.ToInt32(Console.ReadLine());
76
77     Console.WriteLine("¿Adjunta Documento: Notas o certificación de último curso lectivo aprobado? (S/N)");
78     string adjuntosInput = Console.ReadLine();
79     bool adjuntos = adjuntosInput.ToUpper() == "S";
80
81     Console.WriteLine("¿Solicita beca de comedor? (S/N)");
82     string becaComedorInput = Console.ReadLine();
83     bool beca_de_comedor = becaComedorInput.ToUpper() == "S";
84
85     Estudiante estudiante = new Estudiante
86     {
87         NombreEstudiante = nombre_y_apellidos_estudiante,
```

```
188         CedulaEstudiante = numero_cedula_estudiante,
189         NombreEncargado = nombre_y_apellidos_encargado_legal,
190         CedulaEncargado = numero_cedula_encargado_legal,
191         Direccion = direccion_estudiante,
192         Telefono = numero_telefonico,
193         TrabajoEncargado = datos_trabjo_encargado,
194         Grado = grado_a_matricular,
195         NotasAdjuntas = adjuntos,
196         BecaComedor = beca_de_comedor
197     };
198
199     estudiantes.Add(estudiante);
200
201     GuardarDatos(estudiante);
202
203     Console.WriteLine($"Resumen de datos de matricula:
204         {nombre_y_apellidos_estudiante}, {numero_cedula_estudiante},
205         {nombre_y_apellidos_encargado_legal},
206         {numero_cedula_encargado_legal}, {direccion_estudiante},
207         {numero_telefonico}, {datos_trabjo_encargado},
208         {grado_a_matricular}, {adjuntos}, {beca_de_comedor}");
209
210     Console.WriteLine("Registro creado con exito");
211     Console.ReadKey();
212 }
213
214 static void GuardarDatos(Estudiante estudiante)
215 {
216     using (StreamWriter sw = new StreamWriter("matriculas.txt",
217         true))
218     {
219         sw.WriteLine($"Nombre del Estudiante:
220             {estudiante.NombreEstudiante}");
221         sw.WriteLine($"Cedula del Estudiante:
222             {estudiante.CedulaEstudiante}");
223         sw.WriteLine($"Nombre del Encargado:
224             {estudiante.NombreEncargado}");
225         sw.WriteLine($"Cedula del Encargado:
226             {estudiante.CedulaEncargado}");
227         sw.WriteLine($"Direccion: {estudiante.Direccion}");
228         sw.WriteLine($"Telefono: {estudiante.Telefono}");
229         sw.WriteLine($"Trabajo del Encargado:
230             {estudiante.TrabajoEncargado}");
231         sw.WriteLine($"Grado: {estudiante.Grado}");
232         sw.WriteLine($"Notas Adjuntas:
233             {estudiante.NotasAdjuntas}");
234         sw.WriteLine($"Beca de Comedor:
235             {estudiante.BecaComedor}");
236     }
237 }
```

```
124
125     static void Consultar()
126     {
127         Console.WriteLine("Consultar Matricula de Estudiante");
128         Console.WriteLine("Digite el numero de cedula del estudiante  ➤
            para buscar su registro:");
129         string cedulaEstudiante = Console.ReadLine();
130
131         Estudiante estudiante = estudiantes.Find(e => ➤
            e.CedulaEstudiante == cedulaEstudiante);
132
133         if (estudiante != null)
134         {
135             Console.WriteLine($"Nombre del Estudiante: ➤
                {estudiante.NombreEstudiante}");
136             Console.WriteLine($"Cedula del Estudiante: ➤
                {estudiante.CedulaEstudiante}");
137             Console.WriteLine($"Nombre del Encargado: ➤
                {estudiante.NombreEncargado}");
138             Console.WriteLine($"Cedula del Encargado: ➤
                {estudiante.CedulaEncargado}");
139             Console.WriteLine($"Direccion: {estudiante.Direccion}");
140             Console.WriteLine($"Telefono: {estudiante.Telefono}");
141             Console.WriteLine($"Trabajo del Encargado: ➤
                {estudiante.TrabajoEncargado}");
142             Console.WriteLine($"Grado: {estudiante.Grado}");
143             Console.WriteLine($"Notas Adjuntas: ➤
                {estudiante.NotasAdjuntas}");
144             Console.WriteLine($"Beca de Comedor: ➤
                {estudiante.BecaComedor}");
145         }
146         else
147         {
148             Console.WriteLine("Estudiante no encontrado.");
149         }
150
151         Console.ReadKey();
152     }
153
154     static void Modificar()
155     {
156         Console.WriteLine("Modificar datos de Matricula de ➤
            Estudiante");
157         Console.WriteLine("Digite el numero de cedula del estudiante ➤
            para buscar su registro:");
158         string cedulaEstudiante = Console.ReadLine();
159
160         Estudiante estudiante = estudiantes.Find(e => ➤
            e.CedulaEstudiante == cedulaEstudiante);
```

```
161
162         if (estudiante != null)
163         {
164             Console.WriteLine("Digite el nuevo nombre y apellidos del  ➤
                estudiante (deje en blanco para no cambiar:");
165             string nuevoNombre = Console.ReadLine();
166             if (!string.IsNullOrEmpty(nuevoNombre))
167             {
168                 estudiante.NombreEstudiante = nuevoNombre;
169             }
170
171             Console.WriteLine("Digite el nuevo numero de cedula del  ➤
                estudiante (deje en blanco para no cambiar:");
172             string nuevaCedula = Console.ReadLine();
173             if (!string.IsNullOrEmpty(nuevaCedula))
174             {
175                 estudiante.CedulaEstudiante = nuevaCedula;
176             }
177
178             Console.WriteLine("Digite el nuevo nombre y apellidos del  ➤
                encargado legal (deje en blanco para no cambiar:");
179             string nuevoEncargado = Console.ReadLine();
180             if (!string.IsNullOrEmpty(nuevoEncargado))
181             {
182                 estudiante.NombreEncargado = nuevoEncargado;
183             }
184
185             Console.WriteLine("Digite el nuevo numero de cedula del  ➤
                encargado legal (deje en blanco para no cambiar:");
186             string nuevaCedulaEncargado = Console.ReadLine();
187             if (!string.IsNullOrEmpty(nuevaCedulaEncargado))
188             {
189                 estudiante.CedulaEncargado = nuevaCedulaEncargado;
190             }
191
192             Console.WriteLine("Digite la nueva direccion de residencia  ➤
                (deje en blanco para no cambiar:");
193             string nuevaDireccion = Console.ReadLine();
194             if (!string.IsNullOrEmpty(nuevaDireccion))
195             {
196                 estudiante.Direccion = nuevaDireccion;
197             }
198
199             Console.WriteLine("Digite el nuevo numero de telefono  ➤
                (deje en blanco para no cambiar:");
200             string nuevoTelefono = Console.ReadLine();
201             if (!string.IsNullOrEmpty(nuevoTelefono))
202             {
203                 estudiante.Telefono = nuevoTelefono;
```

```
204     }
205
206     Console.WriteLine("Digite los nuevos datos del lugar de trabajo del encargado legal (deje en blanco para no cambiar:");
207     string nuevoTrabajo = Console.ReadLine();
208     if (!string.IsNullOrEmpty(nuevoTrabajo))
209     {
210         estudiante.TrabajoEncargado = nuevoTrabajo;
211     }
212
213     Console.WriteLine("Digite el nuevo grado lectivo por matricular (numero del 7 al 11, deje en blanco para no cambiar:");
214     string nuevoGradoInput = Console.ReadLine();
215     if (!string.IsNullOrEmpty(nuevoGradoInput))
216     {
217         estudiante.Grado = Convert.ToInt32(nuevoGradoInput);
218     }
219
220     Console.WriteLine("¿Adjunta nuevo Documento: Notas o certificación de último curso lectivo aprobado? (S/N, deje en blanco para no cambiar:");
221     string nuevoAdjuntosInput = Console.ReadLine();
222     if (!string.IsNullOrEmpty(nuevoAdjuntosInput))
223     {
224         estudiante.NotasAdjuntas = nuevoAdjuntosInput.ToUpper();
225     }
226
227     Console.WriteLine("¿Solicita nueva beca de comedor? (S/N, deje en blanco para no cambiar:");
228     string nuevaBecaComedorInput = Console.ReadLine();
229     if (!string.IsNullOrEmpty(nuevaBecaComedorInput))
230     {
231         estudiante.BecaComedor = nuevaBecaComedorInput.ToUpper();
232     }
233
234     Console.WriteLine("Registro modificado con exito");
235 }
236 else
237 {
238     Console.WriteLine("Estudiante no encontrado.");
239 }
240
241 Console.ReadKey();
242 }
243
```

```
244     class Estudiante
245     {
246         public string NombreEstudiante { get; set; }
247         public string CedulaEstudiante { get; set; }
248         public string NombreEncargado { get; set; }
249         public string CedulaEncargado { get; set; }
250         public string Direccion { get; set; }
251         public string Telefono { get; set; }
252         public string TrabajoEncargado { get; set; }
253         public int Grado { get; set; }
254         public bool NotasAdjuntas { get; set; }
255         public bool BecaComedor { get; set; }
256     }
257 }
258 }
259
```