

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ORDINARIO
8 {
9     0 referencias
10    internal class Program
11    {
12        10 referencias
13        public enum Operator
14        {
15            Agregar_Presidente = 1, Asignar_estado_democrata, Asignar_estado_republicano, Votos_actuales_democratas, Votos_actuales_republi
16        }
17        0 referencias
18        static void Main(string[] args)
19        {
20            Acciones acciones = new Acciones();
21            bool bandera = true;
22            while (bandera)
23            {
24                Console.WriteLine("Hola que deseas hacer");
25                Console.WriteLine("1) Agregar Presidente");
26                Console.WriteLine("2) Asignar estado democrata");
27                Console.WriteLine("3) Asignar estado republicano");
28                Console.WriteLine("4) Votos actuales democratas");
29                Console.WriteLine("5) Votos actuales republicanos");
30                Console.WriteLine("6) Votos totales");
31            }
32        }
33    }
34 }
```

```
28 Console.WriteLine("6) Votos totales");
29 Console.WriteLine("7) Proximo a ganar");
30 Console.WriteLine("8) Salir");
31 Operator opc = (Operator)Convert.ToInt32(Console.ReadLine());
32 switch (opc)
33 {
34     //Agregar Presidente
35     case Operator.Agregar_Presidente:
36         acciones.AgregarPresidente();
37         break;
38     //Asignar estado democrata
39     case Operator.Asignar_estado_democrata:
40         acciones.AsignarDemocrata();
41         break;
42     //Asignar estado republicano
43     case Operator.Asignar_estado_republicano:
44         acciones.AsignarRepublicano();
45         break;
46     //Votos actuales democratas
47     case Operator.Votos_actuales_democratas:
48         acciones.VotosDemocratas();
49         break;
50     //Votos actuales republicanos
51     case Operator.Votos_actuales_republicano:
52         acciones.VotosRepublicanos();
53         break;
54     //Votos totales
55     case Operator.Votos_totales:
56         acciones.VotosTotales();
57         break;
58 }
59 }
```

```
55         break;
56         //Votos totales
57         case Operator.Votos_totales:
58             Console.WriteLine($"El numero de votos totales es: { acciones.VotosTotales()});
59             break;
60         //Proximo a ganar
61         case Operator.Proximo_a_ganar:
62             acciones.ProximoAGanar();
63             break;
64         //Salir
65         case Operator.salir:
66             bandera = false;
67             break;
68     }
69 }
70 }
71 }
72 }
73 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ORDINARIO
8 {
9     7 referencias
10     internal class Presidente
11     {
12         private string nombre;
13         private int edad;
14         private string Pais_Nac;
15         private string Pais_Residencia_Ultimos_14_años;
16         1 referencia
17         public Presidente() { }
18         1 referencia
19         public Presidente(string nombre, int edad, string pais_Nac, string pais_Residencia_Ultimos_14_años)
20         {
21             this.nombre = nombre;
22             this.edad = edad;
23             Pais_Nac = pais_Nac;
24             Pais_Residencia_Ultimos_14_años = pais_Residencia_Ultimos_14_años;
25         }
26         2 referencias
27         public string Nombre
28         {
29             get { return nombre; }
30             set { nombre = value; }
31         }
32     }
33 }
```

```
28 public int Edad
29 {
30     get { return edad; }
31     set { edad = value; }
32 }
33
34 public string Pais_Nacimiento
35 {
36     get { return Pais_Nac; }
37     set { Pais_Nac = value; }
38 }
39
40 public string Pais_Residencia_Ultimos_14_años_
41 {
42     get { return Pais_Residencia_Ultimos_14_años; }
43     set { Pais_Residencia_Ultimos_14_años = value; }
44 }
45 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ORDINARIO
8 {
9     internal class Acciones
10     {
11         //presidente
12         Presidente presidente = new Presidente();
13         private List<Presidente> lista_presidente = new List<Presidente>();
14         //contador presidentes
15         int cp = 0;
16         //contador republicanos
17         int cr = 0;
18         //contador demócratas
19         int cd = 0;
20         //diccionario de estados
21         Dictionary<string, int> Estado = new Dictionary<string, int>()
22         {
23             {"AL", 9 },
24             {"AR", 3 },
25             {"AZ", 11 },
26             {"AR", 6 },
27             {"CA", 55 },
28             {"NC", 15 },
29             {"SC", 11 },
```

This screenshot shows a Visual Studio Code editor window with the file 'Acciones.cs' open. The code consists of a list of state abbreviations, each paired with an integer value. The list includes states like CO, DC, CT, ND, SD, DE, FL, GA, HI, ID, IL, IN, KS, KY, IA, LA, ME, MD, MA, MI, MN, MS, MO, MT, NE, NV, NH, NJ, NY, and NM. The editor interface includes a menu bar at the top, a toolbar with icons for file operations, and a status bar at the bottom showing 'Lista de errores' and 'Salida'.

```
30 {"CO", 9 },
31 {"DC", 3 },
32 {"CT", 7 },
33 {"ND", 3 },
34 {"SD", 3 },
35 {"DE", 3 },
36 {"FL", 29 },
37 {"GA", 16 },
38 {"HI", 4 },
39 {"ID", 4 },
40 {"IL", 20 },
41 {"IN", 11 },
42 {"KS", 6 },
43 {"KY", 8 },
44 {"IA", 6 },
45 {"LA", 8 },
46 {"ME", 3 },
47 {"MD", 10 },
48 {"MA", 11 },
49 {"MI", 16 },
50 {"MN", 10 },
51 {"MS", 6 },
52 {"MO", 10 },
53 {"MT", 3 },
54 {"NE", 4 },
55 {"NV", 6 },
56 {"NH", 4 },
57 {"NJ", 14 },
58 {"NY", 29 },
59 {"NM", 5 },
```

This screenshot shows the same Visual Studio Code editor window, but now displaying the implementation of the 'AgregarPresidente()' method. The code includes the declaration of two dictionaries, 'EstadoD' and 'EstadoR', and the logic for adding a new president. It checks if the number of presidents added is greater than or equal to 2, and if so, it prints a message. Otherwise, it attempts to add the president's name to the dictionary. The editor interface is consistent with the previous screenshot, showing the menu bar, toolbar, and status bar.

```
60 {"OH", 7 },
61 {"OK", 7 },
62 {"OR", 7 },
63 {"PA", 20 },
64 {"RI", 4 },
65 {"TN", 11 },
66 {"TX", 38 },
67 {"UT", 6 },
68 {"VT", 3 },
69 {"VA", 13 },
70 {"WV", 3 },
71 {"WA", 12 },
72 {"WI", 10 },
73 {"WY", 3 },
74
75 };
76 Dictionary<string, int> EstadoD = new Dictionary<string, int>();
77 Dictionary<string, int> EstadoR = new Dictionary<string, int>();
78
79 public void AgregarPresidente()
80 {
81     if (cp >= 2)
82     {
83         Console.WriteLine("Ya se agrego el numero maximo de presidentes");
84     }
85     else
86     {
87         try
88         {
89             Console.WriteLine("Nombre");
```

```
ORDINARIO
89 presidente.Nombre = Console.ReadLine();
90 Console.WriteLine("Edad");
91 presidente.Edad = Int32.Parse(Console.ReadLine());
92 Console.WriteLine("País de nacimiento, ingrese USA U OTRTO");
93 presidente.Pais_Nacimiento = Console.ReadLine();
94 Console.WriteLine("País de residencia de los últimos 14 años, ingrese USA U OTRTO");
95 presidente.Pais_Residencia_Ultimos_14_años_ = Console.ReadLine();
96 if (presidente.Edad >= 35 && presidente.Pais_Nacimiento == "USA" && presidente.Pais_Residencia_Ultimos_14_años_ == "USA")
97 {
98     lista_presidente.Add(new Presidente(presidente.Nombre, presidente.Edad, presidente.Pais_Nacimiento, presidente.Pais
99     Console.WriteLine("Se agrego con exito");
100     cp += 1;
101 }
102 }
103 else
104 {
105     Console.WriteLine("No cumple con alguno de los requisitos el presidente");
106 }
107 }
108 catch (Exception ex)
109 {
110     Console.WriteLine("Error en los datos ingresados");
111 }
112 }
113
114 1 referencia
115 public void AsignarDemocrata()
116 {
117     if (cr >= 270)
118     {
119         Console.WriteLine($"El presidente democrata gana, con un total de {cd} votos, no se puede asignar mas estados");
120     }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
```

```
ORDINARIO
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
128 }
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
```

This screenshot shows the Visual Studio IDE with the file 'Acciones.cs' open. The code defines a method 'VotosDemocratas()' that iterates through a dictionary 'Estado'. For each entry, it prints the state name and its value, prompts the user to select a state, and if the selection exists, it increments a counter 'value' and removes the state from the dictionary. The method concludes by printing the total number of states assigned to Democrats.

```
147 Console.WriteLine($"El presidente republicano gana, con un total de {cr} votos, no se puede asignar mas estados");
148 }
149 else
150 {
151     Console.WriteLine("Los estados disponibles son:");
152     foreach (KeyValuePair<string, int> user in Estado)
153     {
154         Console.WriteLine($"Estado {user.Key} con un valor de {user.Value}");
155     }
156     Console.WriteLine("Que estado deseas elegir?");
157     string edo = Console.ReadLine();
158     if (Estado.ContainsKey(edo))
159     {
160         int value;
161         Estado.TryGetValue(edo, out value);
162         Estado.Add(edo, value);
163         Estado.Remove(edo);
164         Console.WriteLine("El partido fue asignado exito");
165         cr += value;
166     }
167     else
168     {
169         Console.WriteLine("El estado no existe o ya fue asignado, pruebe con otro");
170     }
171 }
172 }
173
174 public void VotosDemocratas()
175 {
176     Console.WriteLine("Los estados democratas fueron");
```

This screenshot shows the continuation of the 'Acciones.cs' file in Visual Studio. It includes three methods: 'VotosRepublicanos()' which prints the total votes for Republicans and the list of states; 'VotoSTotales()' which returns the sum of votes for both parties; and 'ProximoAGanar()' which determines and prints the next party to win based on the current vote counts.

```
176     foreach (KeyValuePair<string, int> user in EstadoD)
177     {
178         Console.WriteLine($"Estado {user.Key} con un valor de {user.Value}");
179     }
180     Console.WriteLine($"Con un total de {cd} votos");
181 }
182
183 public void VotosRepublicanos()
184 {
185     Console.WriteLine("Los estados republicanos fueron");
186     foreach (KeyValuePair<string, int> user in EstadoR)
187     {
188         Console.WriteLine($"Estado {user.Key} con un valor de {user.Value}");
189     }
190     Console.WriteLine($"Con un total de {cr} votos");
191 }
192
193 public int VotoSTotales()
194 {
195     return cd+cr;
196 }
197
198 public void ProximoAGanar()
199 {
200     if (cd> cr)
201     {
202         Console.WriteLine("El partido democrata es el proximo a ganar");
203     }
204     else
205     {
206         Console.WriteLine("El partido republicano es el proximo a ganar");
```

