
# INSTITUTO POLITÉCNICO NACIONAL

## Escuela Superior de Computo

# Práctica 5: Tablero

*Teoría de la Computación*

Autor:
Hernández Vergara Eduardo

Junio 2022

# Índice

# 1. Introdución

## 1.1. Objetivo

Elaborar un programa para realizar movimientos ortogonales y diagonales en un tablero de ajedrez de 4x4 con dos piezas.

## 1.2. Automatas finitos (AFD)

Un autómata finito (AF) o máquina de estado finito es un modelo computacional que realiza cómputos en forma automática sobre una entrada para producir una salida. Este modelo está conformado por un alfabeto, un conjunto de estados finito, una función de transición, un estado inicial y un conjunto de estados finales. Su funcionamiento se basa en una función de transición, que recibe a partir de un estado inicial una cadena de caracteres pertenecientes al alfabeto (la entrada), y que va leyendo dicha cadena a medida que el autómata se desplaza de un estado a otro, para finalmente detenerse en un estado final o de aceptación, que representa la salida. La finalidad de los autómatas finitos es la de reconocer lenguajes regulares, que corresponden a los lenguajes formales más simples según la Jerarquía de Chomsky.

## 1.3. Automata Finito No Determinista (AFN)

Un autómata finito no determinista(AFN) tiene la capacidad de estar en varios estados a la vez. Esta capacidad a menudo se expresa como la posibilidad de que el autómata conjeture algo acerca de su entrada. Por ejemplo, cuando el autómata se utiliza para buscar determinadas secuencias de caracteres (por ejemplo, palabras clave) dentro de una cadena de texto larga, resulta útil conjeturar que estamos al principio de una de estas cadenas y utilizar una secuencia de estados únicamente para comprobar la aparición de la cadena, carácter por carácter.

# 2. Solución

Para la implementación de la solución se uso C#, con .Net 6.0, con el IDE Visual Studio 2022, como veremos a continuación con las capturas de resultado, veremos como salio.

## 2.1. Correr el Programa

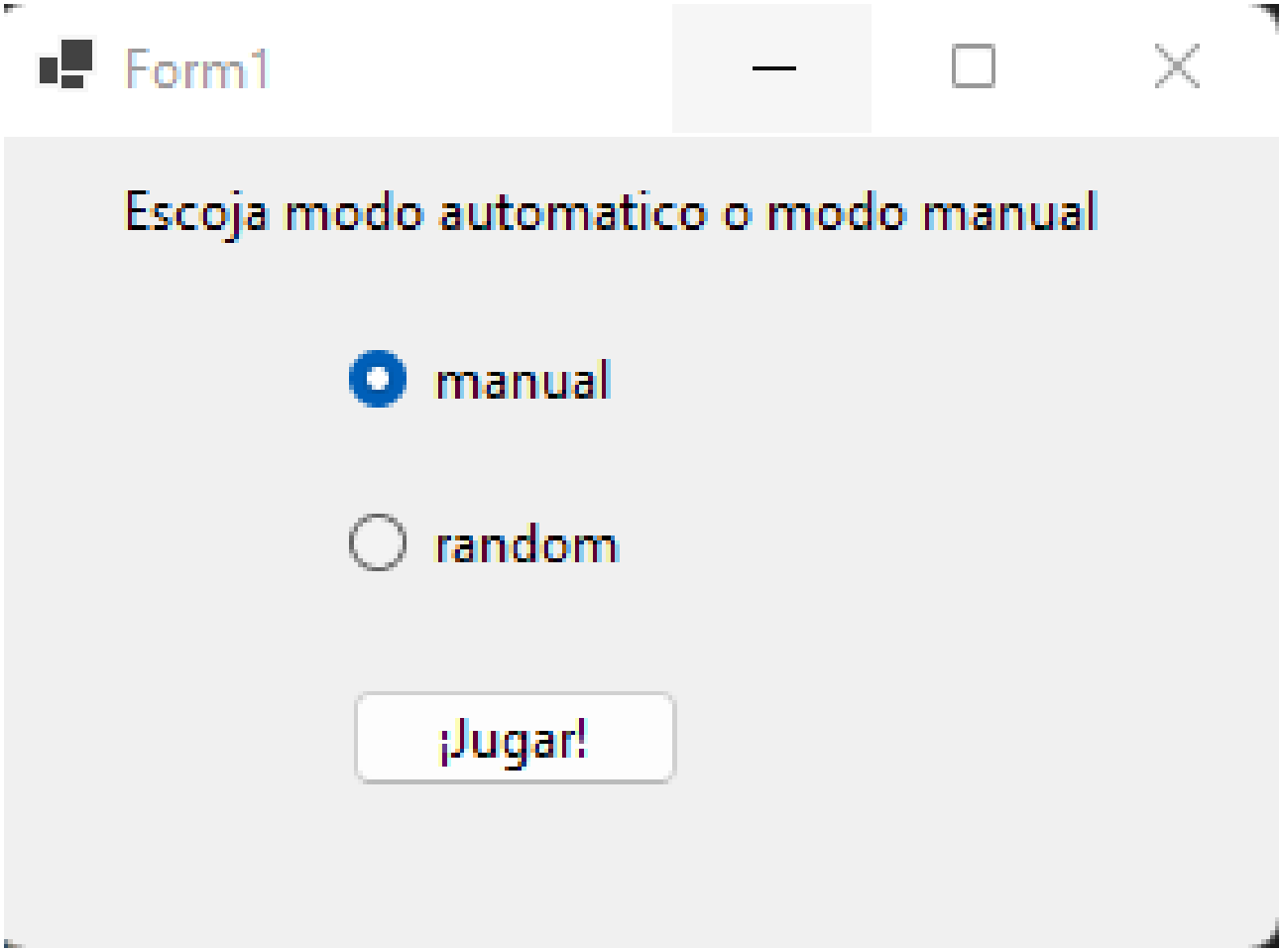


Figura 1: Menu manual o aleatorio

Aqui procedo a realizar un menu para escoger manual o aleatorio

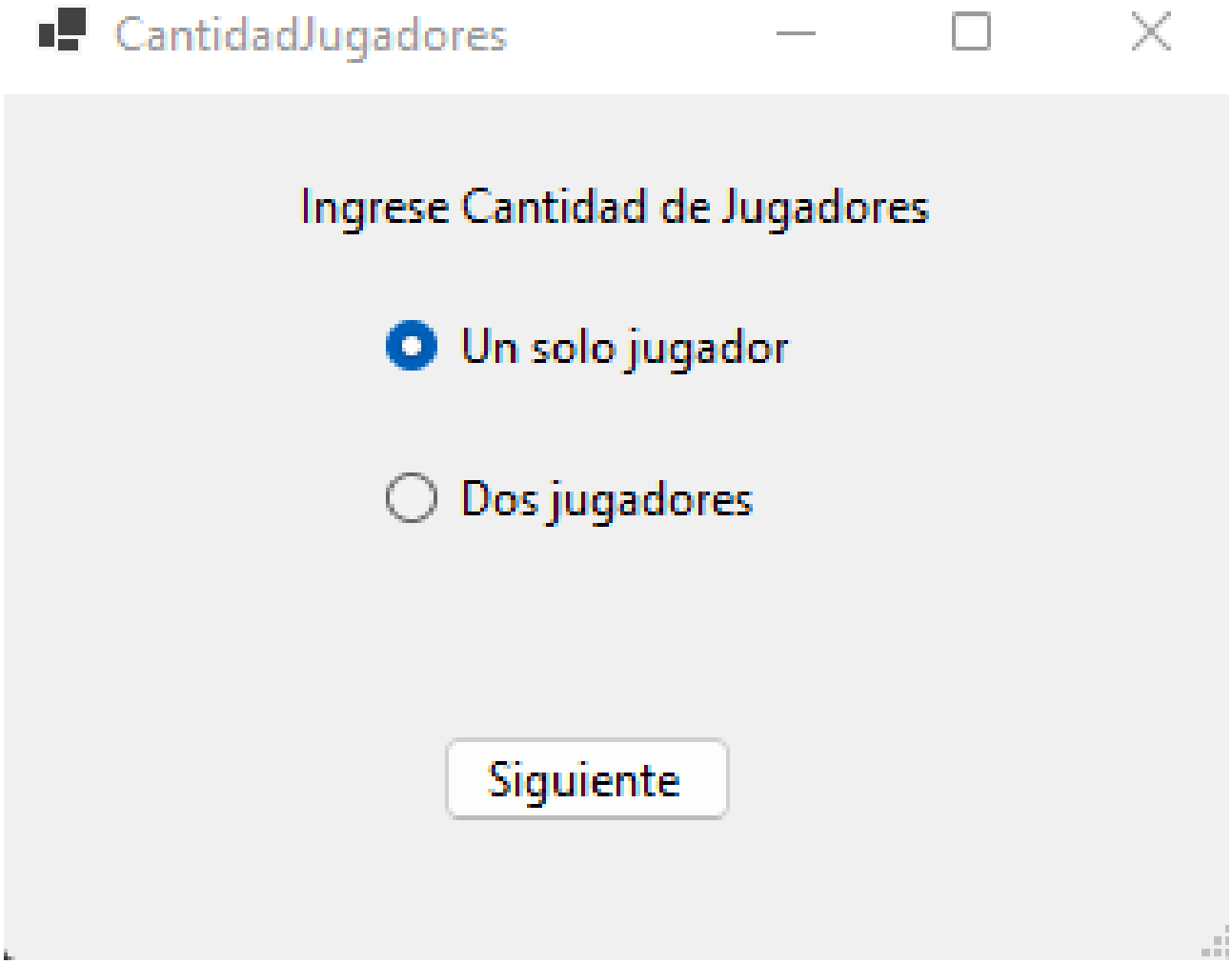


Figura 2: Cantidad Jugadores


Figura 3: Todas las rutas calculadas


Figura 4: Rutas Ganadoras para el jugador 1


Figura 5: Rutas Ganadoras para el Jugador 2


Figura 6: Animacion del Tablero, jugar

# 3. Código

## 3.1. El automata

**Listing 1: Pantalla Principal del Juego C# .net 6.0**

```
namespace Programa5
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (manual.Checked){
                Juego.CantidadJugadores cantidad = new();
                cantidad.ShowDialog();
            }
            else {
                MessageBox.Show("Aleatorio");
                UtilidadesC.Utilidades utilidades = new();
                string o = utilidades.GenerarCadenas();
                string p = utilidades.GenerarCadenas();
                bool c = utilidades.CantidadJugadores();
                UtilidadesC.DatoCad.Cadena1 = o;
                UtilidadesC.DatoCad.Cadena2 = p;
                UtilidadesC.DatoCad.Jugadores = c;
                if (c)
                {
                    MessageBox.Show($"Es un jugador y la cadena es {o}");
                }
                else {
                    MessageBox.Show($"Son dos jugadores y la cadena 1 es {
                        o} y la cadena 2 es {p}");
                }
                Juego.Tablero tablero = new();
                tablero.ShowDialog();
            }
        }
    }
}
```

**Listing 2: Cantidad Jugadores C# .net 6.0**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa5.Juego
{
```

```
13      public partial class CantidadJugadores : Form
14      {
15          public CantidadJugadores()
16          {
17              InitializeComponent();
18          }
19
20          private void Siguiete_Click(object sender, EventArgs e)
21          {
22              if (jugador1.Checked){
23                  Unjugador uno = new();
24                  uno.ShowDialog();
25              }
26              else {
27                  DosJugadores dos = new();
28                  dos.ShowDialog();
29              }
30          }
31      }
32  }
```

**Listing 3: Dos jugadores C# .net 6.0**

```
1   using System;
2   using System.Collections.Generic;
3   using System.ComponentModel;
4   using System.Data;
5   using System.Drawing;
6   using System.Linq;
7   using System.Text;
8   using System.Threading.Tasks;
9   using System.Windows.Forms;
10
11  namespace Programa5.Juego
12  {
13      public partial class DosJugadores : Form
14      {
15          public DosJugadores()
16          {
17              InitializeComponent();
18          }
19
20          private void Generar_Click(object sender, EventArgs e)
21          {
22              UtilidadesC.Utilidades u = new();
23              textBox1.Text = u.GenerarCadenas();
24              textBox2.Text = u.GenerarCadenas();
25          }
26
27          private void Guardar_Click(object sender, EventArgs e)
28          {
29              string? a, b;
30              a = textBox1.Text;
31              b = textBox2.Text;
32              if (a == "" || b == "")
33              {
34                  MessageBox.Show("No se puede guardar una cadena vacia");
35              }
36              else
37              {
```

```
                UtilidadesC.DatoCad.Cadena1 = a;
                UtilidadesC.DatoCad.Cadena2 = b;
                UtilidadesC.DatoCad.Jugadores = false;
                Tablero tablero = new();
                tablero.ShowDialog();
            }
        }
    }
}
```

**Listing 4: Tablero C# .net 6.0**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa5.Juego
{
    public partial class Tablero : Form
    {
        public Tablero()
        {
            InitializeComponent();
        }

        private void Tablero_Paint(object sender, PaintEventArgs e)
        {
            Graphics g = e.Graphics;
            //Dibujo el tablero
            SolidBrush r = new(Color.Red);
            SolidBrush n = new(Color.Black);
            TableroColor(r, n, g);
            SolidBrush brush = new(Color.Black);
            Pen pen = new(brush);
            Font drf = new("Arial", 16);
            StringFormat fmt = new();
            if (UtilidadesC.DatoCad.Jugadores)
            {
                List<char> ax = new();
                String cadena = UtilidadesC.DatoCad.Cadena1;
                ax = dfa(cadena);
                crearRutas(ax, cadena, false);
            }
            else {
                List<char> ax = new();
                List<char> ax2 = new();
                String cadena = UtilidadesC.DatoCad.Cadena1;
                String cadena2 = UtilidadesC.DatoCad.Cadena2;
                ax = dfa(cadena);
                ax2 = dfa2(cadena2);
                crearRutas(ax, cadena, false);
                crearRutas(ax2, cadena2, true);
            }
            dibujarRutas(g, n, r);
```

```
49              g.DrawString(UtilidadesC.DatoCad.Cadena1, drf, brush, 15, 15);
50          }
51          private void TableroColor(SolidBrush r, SolidBrush n, Graphics g)
52          {
53              for (int k = 50; k <= 200; k += 50)
54              {
55                  if (k == 50 || k == 150)
56                  {
57                      for (int i = 50; i < 200; i += 100)
58                      {
59                          g.FillRectangle(n, i, k, 50, 50);
60                      }
61                      for (int i = 100; i <= 200; i += 100)
62                      {
63                          g.FillRectangle(r, i, k, 50, 50);
64                      }
65                  }
66                  if (k == 100 || k == 200)
67                  {
68                      for (int i = 50; i < 200; i += 100)
69                      {
70                          g.FillRectangle(r, i, k, 50, 50);
71                      }
72                      for (int i = 100; i <= 200; i += 100)
73                      {
74                          g.FillRectangle(n, i, k, 50, 50);
75                      }
76                  }
77              }
78
79          }
80          private List<char> dfa(String cadena)
81          {
82              int i = 0;
83              char stado = 'A';
84              List<char> axp = new();
85              axp.Add(stado);
86              while (i < cadena.Length)
87              {
88                  switch (stado)
89                  {
90                      case 'A':
91                          if (cadena[i] == 'r')
92                          {
93                              stado = 'B';
94                          }
95                          else
96                          {
97                              stado = 'C';
98                          }
99                          break;
100                     case 'B':
101                     case 'C':
102                         if (cadena[i] == 'r')
103                         {
104                             stado = 'D';
105                         }
106                         else
107                         {
108                             if (stado == 'B')
```

```
                {
                    stado = 'E';
                }
                if (stado == 'B')
                {
                    stado = 'F';
                }
            }

            break;
        case 'D':
        case 'F':
        case 'G':
        case 'H':
        case 'I':
        case 'J':
        case 'K':
            if (cadena[i] == 'r')
            {
                stado = 'G';
            }
            else
            {
                if (stado == 'D' || stado == 'I')
                    stado = 'H';
                if (stado == 'J')
                    stado = 'F';
                if (stado == 'G' || stado == 'H' || stado == '
                   K')
                    stado = 'K';
                if (stado == 'F')
                    stado = 'J';
            }
            break;
        case 'E':
            if (cadena[i] == 'r')
            {
                stado = 'G';
            }
            else
            {
                stado = 'H';
            }
            break;
        default:
            break;
    }
    axp.Add(stado);
    i++;
}
return axp;
}
private List<char> dfa2(String cadena)
{
    int i = 0;
    char stado = 'A';
    List<char> axp = new();
    axp.Add(stado);
    while (i < cadena.Length)
    {
```

```
168                 switch (stado)
169                 {
170                     case 'A':
171                         if (cadena[i] == 'r')
172                         {
173                             stado = 'B';
174                         }
175                         else
176                         {
177                             stado = 'C';
178                         }
179                         break;
180                     case 'B':
181                     case 'C':
182                         if (cadena[i] == 'n')
183                         {
184                             stado = 'E';
185                         }
186                         else
187                         {
188                             if (stado == 'B')
189                             {
190                                 stado = 'D';
191                             }
192                             if (stado == 'B')
193                             {
194                                 stado = 'F';
195                             }
196                         }
197 
198                         break;
199                     case 'D':
200                     case 'E':
201                     case 'G':
202                     case 'H':
203                     case 'I':
204                     case 'J':
205                     case 'K':
206                         if (cadena[i] == 'n')
207                         {
208                             stado = 'H';
209                         }
210                         else
211                         {
212                             if (stado == 'E' || stado == 'J')
213                                 stado = 'I';
214                             if (stado == 'D')
215                                 stado = 'G';
216                             if (stado == 'H' || stado == 'I' || stado == '
                                K')
217                                 stado = 'K';
218                             if (stado == 'G')
219                                 stado = 'D';
220                         }
221                         break;
222                     case 'F':
223                         if (cadena[i] == 'n')
224                         {
225                             stado = 'J';
226                         }
```

```
227                         else
228                         {
229                             stado = 'I';
230                         }
231                         break;
232                     default:
233                         break;
234                 }
235                 axp.Add(stado);
236                 i++;
237             }
238             return axp;
239         }
240         private void crearRutas(List<char> ax, string cadena, bool s)
241         {
242             int n = ax.Count;
243             int m = cadena.Length;
244             List<List<string>> conjunto = new();
245             if (!s)
246             {
247                 for (int i = 0; i < n; i++)
248                 {
249                     conjunto.Add(estados(ax[i]));
250                 }
251                 List<string> cadena1 = new();
252                 foreach (var c in "final")
253                 {
254                     cadena1.Add(c.ToString());
255                 }
256                 conjunto.Add(cadena1);
257                 ruta("1", cadena, conjunto, 0, 0, m, "1", false);
258                 //MessageBox.Show(cadena1);
259             }
260 
261             else {
262                 for (int i = 0; i < n; i++)
263                 {
264                     conjunto.Add(estados2(ax[i]));
265                 }
266                 List<string> cadena2 = new();
267                 foreach (var c in "final")
268                 {
269                     cadena2.Add(c.ToString());
270                 }
271                 conjunto.Add(cadena2);
272                 ruta("4", cadena, conjunto, 0, 0, m, "4", true);
273             }
274         }
275         private List<string> estados(char e)
276         {
277             List<string> conjuntos = new();
278             switch (e)
279             {
280                 case 'A':
281                     conjuntos.Add("1");
282                     break;
283                 case 'B':
284                     conjuntos.Add("2");
285                     conjuntos.Add("5");
286                     break;
```

```
287                     case 'C':
288                         conjuntos.Add("6");
289                         break;
290                     case 'D':
291                         conjuntos.Add("2");
292                         conjuntos.Add("5");
293                         conjuntos.Add("7");
294                         conjuntos.Add("10");
295                         break;
296                     case 'E':
297                         conjuntos.Add("1");
298                         conjuntos.Add("3");
299                         conjuntos.Add("6");
300                         conjuntos.Add("9");
301                         break;
302                     case 'F':
303                         conjuntos.Add("1");
304                         conjuntos.Add("3");
305                         conjuntos.Add("9");
306                         conjuntos.Add("11");
307                         break;
308                     case 'G':
309                         conjuntos.Add("2");
310                         conjuntos.Add("4");
311                         conjuntos.Add("5");
312                         conjuntos.Add("7");
313                         conjuntos.Add("10");
314                         conjuntos.Add("12");
315                         conjuntos.Add("13");
316                         conjuntos.Add("15");
317                         break;
318                     case 'H':
319                         conjuntos.Add("1");
320                         conjuntos.Add("3");
321                         conjuntos.Add("6");
322                         conjuntos.Add("8");
323                         conjuntos.Add("9");
324                         conjuntos.Add("11");
325                         conjuntos.Add("14");
326                         break;
327                     case 'I':
328                         conjuntos.Add("2");
329                         conjuntos.Add("4");
330                         conjuntos.Add("5");
331                         conjuntos.Add("7");
332                         conjuntos.Add("10");
333                         conjuntos.Add("13");
334                         break;
335                     case 'J':
336                         conjuntos.Add("6");
337                         conjuntos.Add("8");
338                         conjuntos.Add("14");
339                         conjuntos.Add("16");
340                         break;
341                     case 'K':
342                         conjuntos.Add("1");
343                         conjuntos.Add("3");
344                         conjuntos.Add("6");
345                         conjuntos.Add("8");
346                         conjuntos.Add("9");
```

```
347                     conjuntos.Add("11");
348                     conjuntos.Add("14");
349                     conjuntos.Add("16");
350                     break;
351                 default:
352                     break;
353             }
354             return conjuntos;
355         }
356         private List<string> estados2(char e)
357         {
358             List<string> conjuntos = new();
359             switch (e)
360             {
361                 case 'A':
362                     conjuntos.Add("4");
363                     break;
364                 case 'B':
365                     conjuntos.Add("7");
366                     break;
367                 case 'C':
368                     conjuntos.Add("3");
369                     conjuntos.Add("8");
370                     break;
371                 case 'D':
372                     conjuntos.Add("2");
373                     conjuntos.Add("4");
374                     conjuntos.Add("10");
375                     conjuntos.Add("12");
376                     break;
377                 case 'E':
378                     conjuntos.Add("3");
379                     conjuntos.Add("6");
380                     conjuntos.Add("8");
381                     conjuntos.Add("11");
382                     break;
383                 case 'F':
384                     conjuntos.Add("2");
385                     conjuntos.Add("4");
386                     conjuntos.Add("7");
387                     conjuntos.Add("12");
388                     break;
389                 case 'G':
390                     conjuntos.Add("5");
391                     conjuntos.Add("7");
392                     conjuntos.Add("13");
393                     conjuntos.Add("15");
394                     break;
395                 case 'H':
396                     conjuntos.Add("1");
397                     conjuntos.Add("3");
398                     conjuntos.Add("6");
399                     conjuntos.Add("8");
400                     conjuntos.Add("9");
401                     conjuntos.Add("11");
402                     conjuntos.Add("14");
403                     conjuntos.Add("16");
404                     break;
405                 case 'I':
406                     conjuntos.Add("2");
```

```
                    conjuntos.Add("4");
                    conjuntos.Add("5");
                    conjuntos.Add("7");
                    conjuntos.Add("10");
                    conjuntos.Add("12");
                    conjuntos.Add("15");
                    break;
                case 'J':
                    conjuntos.Add("1");
                    conjuntos.Add("3");
                    conjuntos.Add("6");
                    conjuntos.Add("8");
                    conjuntos.Add("11");
                    conjuntos.Add("16");
                    break;
                case 'K':
                    conjuntos.Add("2");
                    conjuntos.Add("4");
                    conjuntos.Add("5");
                    conjuntos.Add("7");
                    conjuntos.Add("10");
                    conjuntos.Add("12");
                    conjuntos.Add("13");
                    conjuntos.Add("15");
                    break;
                default:
                    break;
            }
            return conjuntos;
        }
        private void ruta(string cadenaI, string cadena, List<List<string
            >> datos, int i, int j, int n, string k, bool s)
        {
            String rutas = "D:\\Documentos\\ESCOM\\Teoria Computacional\\
                DatosP5";
            List<string> cadenapro = new();
            foreach (var c in "final")
            {
                cadenapro.Add(c.ToString());
            }
            for (; j < datos[i].Count; j++)
            {
                if (!datos[i + 1].OrderBy(m => m).SequenceEqual(cadenapro.
                    OrderBy(m => m)) && datos[i][j].Equals(k))
                {
                    for (int p = 0; p < datos[i + 1].Count; p++)
                    {
                        bool si = nfa(cadena[i], datos[i][j].ToString(),
                            datos[i + 1][p].ToString());
                        if (si)
                        {
                            string aux = cadenaI;
                            string dato = datos[i + 1][p].ToString();
                            aux += ",";
                            aux += dato;
                            if (i == n - 1)
                            {
                                File.AppendAllText($"{rutas}\\Rutas.txt",
                                    $"{aux}\n");
                                if (s)
```

```
462                                 {
463                                     if (dato == "13")
464                                         File.AppendAllText($"{rutas}\\
                                            RutasA2.txt", $"{aux}\n");
465                                 }
466                                 else
467                                 {
468                                     if (dato == "16")
469                                         File.AppendAllText($"{rutas}\\
                                            RutasA1.txt", $"{aux}\n");
470                                 }
471                             }
472                             ruta(aux, cadena, datos, i + 1, 0, n, datos[i
                                + 1][p], s);
473                         }
474                     }
475
476                 }
477
478             }
479         }
480         private bool nfa(char color, string act, string sig)
481         {
482             bool cumple = true;
483             switch (act)
484             {
485                 case "1":
486                     if (color == 'r')
487                     {
488                         if (sig == "2" || sig == "5")
489                         {
490                             cumple = true;
491                         }
492                         else
493                             cumple = false;
494                     }
495                     else
496                     {
497                         if (sig == "6")
498                         {
499                             cumple = true;
500                         }
501                         else
502                             cumple = false;
503                     }
504
505                     break;
506                 case "2":
507                     if (color == 'r')
508                     {
509                         if (sig == "5" || sig == "7")
510                         {
511                             cumple = true;
512                         }
513                         else
514                             cumple = false;
515                     }
516                     else
517                     {
518                         if (sig == "1" || sig == "3" || sig == "6")
```

```
519                         {
520                             cumple = true;
521                         }
522                         else
523                             cumple = false;
524                     }
525                     break;
526                 case "3":
527                     if (color == 'r')
528                     {
529                         if (sig == "2" || sig == "4" || sig == "7")
530                         {
531                             cumple = true;
532                         }
533                         else
534                             cumple = false;
535                     }
536                     else
537                     {
538                         if (sig == "6" || sig == "8")
539                         {
540                             cumple = true;
541                         }
542                         else
543                             cumple = false;
544                     }
545                     break;
546                 case "4":
547                     if (color == 'r')
548                     {
549                         if (sig == "7")
550                         {
551                             cumple = true;
552                         }
553                         else
554                             cumple = false;
555                     }
556                     else
557                     {
558                         if (sig == "3" || sig == "8")
559                         {
560                             cumple = true;
561                         }
562                         else
563                             cumple = false;
564                     }
565                     break;
566                 case "5":
567                     if (color == 'r')
568                     {
569                         if (sig == "2" || sig == "10")
570                         {
571                             cumple = true;
572                         }
573                         else
574                             cumple = false;
575                     }
576                     else
577                     {
578                         if (sig == "1" || sig == "6" || sig == "9")
```

```
579                         {
580                             cumple = true;
581                         }
582                         else
583                             cumple = false;
584                     }
585                     break;
586                 case "6":
587                     if (color == 'r')
588                     {
589                         if (sig == "2" || sig == "5" || sig == "7" || sig
                               == "10")
590                         {
591                             cumple = true;
592                         }
593                         else
594                             cumple = false;
595                     }
596                     else
597                     {
598                         if (sig == "1" || sig == "3" || sig == "9" || sig
                               == "11")
599                         {
600                             cumple = true;
601                         }
602                         else
603                             cumple = false;
604                     }
605                     break;
606                 case "7":
607                     if (color == 'r')
608                     {
609                         if (sig == "2" || sig == "4" || sig == "10" || sig
                               == "12")
610                         {
611                             cumple = true;
612                         }
613                         else
614                             cumple = false;
615                     }
616                     else
617                     {
618                         if (sig == "3" || sig == "6" || sig == "8" || sig
                               == "11")
619                         {
620                             cumple = true;
621                         }
622                         else
623                             cumple = false;
624                     }
625                     break;
626                 case "8":
627                     if (color == 'r')
628                     {
629                         if (sig == "4" || sig == "7" || sig == "12")
630                         {
631                             cumple = true;
632                         }
633                         else
634                             cumple = false;
```

```
635                         }
636                         else
637                         {
638                             if (sig == "3" || sig == "11")
639                             {
640                                 cumple = true;
641                             }
642                             else
643                                 cumple = false;
644                         }
645                         break;
646                     case "9":
647                         if (color == 'r')
648                         {
649                             if (sig == "5" || sig == "10" || sig == "13")
650                             {
651                                 cumple = true;
652                             }
653                             else
654                                 cumple = false;
655                         }
656                         else
657                         {
658                             if (sig == "6" || sig == "14")
659                             {
660                                 cumple = true;
661                             }
662                             else
663                                 cumple = false;
664                         }
665                         break;
666                     case "10":
667                         if (color == 'r')
668                         {
669                             if (sig == "5" || sig == "7" || sig == "13" || sig
                                    == "15")
670                             {
671                                 cumple = true;
672                             }
673                             else
674                                 cumple = false;
675                         }
676                         else
677                         {
678                             if (sig == "6" || sig == "9" || sig == "11" || sig
                                    == "14")
679                             {
680                                 cumple = true;
681                             }
682                             else
683                                 cumple = false;
684                         }
685                         break;
686                     case "11":
687                         if (color == 'r')
688                         {
689                             if (sig == "7" || sig == "10" || sig == "12" ||
                                sig == "15")
690                             {
691                                 cumple = true;
```

```
692                         }
693                         else
694                             cumple = false;
695                     }
696                     else
697                     {
698                         if (sig == "6" || sig == "8" || sig == "14" || sig
                                == "16")
699                         {
700                             cumple = true;
701                         }
702                         else
703                             cumple = false;
704                     }
705                     break;
706                 case "12":
707                     if (color == 'r')
708                     {
709                         if (sig == "7" || sig == "15")
710                         {
711                             cumple = true;
712                         }
713                         else
714                             cumple = false;
715                     }
716                     else
717                     {
718                         if (sig == "8" || sig == "11" || sig == "16")
719                         {
720                             cumple = true;
721                         }
722                         else
723                             cumple = false;
724                     }
725                     break;
726                 case "13":
727                     if (color == 'r')
728                     {
729                         if (sig == "10")
730                         {
731                             cumple = true;
732                         }
733                         else
734                             cumple = false;
735                     }
736                     else
737                     {
738                         if (sig == "9" || sig == "14")
739                         {
740                             cumple = true;
741                         }
742                         else
743                             cumple = false;
744                     }
745                     break;
746                 case "14":
747                     if (color == 'r')
748                     {
749                         if (sig == "10" || sig == "13" || sig == "15")
750                         {
```

```
751                             cumple = true;
752                         }
753                         else
754                             cumple = false;
755                     }
756                     else
757                     {
758                         if (sig == "9" || sig == "11")
759                         {
760                             cumple = true;
761                         }
762                         else
763                             cumple = false;
764                     }
765                     break;
766                 case "15":
767                     if (color == 'r')
768                     {
769                         if (sig == "10" || sig == "12")
770                         {
771                             cumple = true;
772                         }
773                         else
774                             cumple = false;
775                     }
776                     else
777                     {
778                         if (sig == "11" || sig == "14" || sig == "16")
779                         {
780                             cumple = true;
781                         }
782                         else
783                             cumple = false;
784                     }
785                     break;
786                 case "16":
787                     if (color == 'r')
788                     {
789                         if (sig == "12" || sig == "15")
790                         {
791                             cumple = true;
792                         }
793                         else
794                             cumple = false;
795                     }
796                     else
797                     {
798                         if (sig == "11")
799                         {
800                             cumple = true;
801                         }
802                         else
803                             cumple = false;
804                     }
805                     break;
806             }
807             return cumple;
808         }
809         private void dibujarRutas(Graphics g, SolidBrush n, SolidBrush r)
            {
```

```
810             if (UtilidadesC.DatoCad.Jugadores){
811                 StreamReader lectura = new StreamReader("D:\\Documentos\\
                        ESCOM\\Teoria Computacional\\DatosP5\\RutasA1.txt");
812                 string linea;
813                 SolidBrush green = new(Color.Green);
814                 while ((linea = lectura.ReadLine()) != null ) {
815                     string[] vs = linea.Split(',');
816                     foreach (string v in vs) {
817                         switch (v) {
818                             case "1":
819                                 g.FillEllipse(green, 60, 60, 25, 25);
820                                 Thread.Sleep(500);
821                                 TableroColor(r, n, g);
822                                 break;
823                             case "2":
824                                 g.FillEllipse(green, 110, 60, 25, 25);
825                                 Thread.Sleep(500);
826                                 TableroColor(r, n, g);
827                                 break;
828                             case "3":
829                                 g.FillEllipse(green, 160, 60, 25, 25);
830                                 Thread.Sleep(500);
831                                 TableroColor(r, n, g);
832                                 break;
833                             case "4":
834                                 g.FillEllipse(green, 210, 60, 25, 25);
835                                 Thread.Sleep(500);
836                                 TableroColor(r, n, g);
837                                 break;
838                             case "5":
839                                 g.FillEllipse(green, 60, 110, 25, 25);
840                                 Thread.Sleep(500);
841                                 TableroColor(r, n, g);
842                                 break;
843                             case "6":
844                                 g.FillEllipse(green, 110, 110, 25, 25);
845                                 Thread.Sleep(500);
846                                 TableroColor(r, n, g);
847                                 break;
848                             case "7":
849                                 g.FillEllipse(green, 160, 110, 25, 25);
850                                 Thread.Sleep(500);
851                                 TableroColor(r, n, g);
852                                 break;
853                             case "8":
854                                 g.FillEllipse(green, 210, 110, 25, 25);
855                                 Thread.Sleep(500);
856                                 TableroColor(r, n, g);
857                                 break;
858                             case "9":
859                                 g.FillEllipse(green, 60, 160, 25, 25);
860                                 Thread.Sleep(500);
861                                 TableroColor(r, n, g);
862                                 break;
863                             case "10":
864                                 g.FillEllipse(green, 110, 160, 25, 25);
865                                 Thread.Sleep(500);
866                                 TableroColor(r, n, g);
867                                 break;
868                             case "11":
```

```
869                                 g.FillEllipse(green, 160, 160, 25, 25);
870                                 Thread.Sleep(500);
871                                 TableroColor(r, n, g);
872                                 break;
873                             case "12":
874                                 g.FillEllipse(green, 210, 160, 25, 25);
875                                 Thread.Sleep(500);
876                                 TableroColor(r, n, g);
877                                 break;
878                             case "13":
879                                 g.FillEllipse(green, 60, 210, 25, 25);
880                                 Thread.Sleep(500);
881                                 TableroColor(r, n, g);
882                                 break;
883                             case "14":
884                                 g.FillEllipse(green, 110, 210, 25, 25);
885                                 Thread.Sleep(500);
886                                 TableroColor(r, n, g);
887                                 break;
888                             case "15":
889                                 g.FillEllipse(green, 160, 210, 25, 25);
890                                 Thread.Sleep(500);
891                                 TableroColor(r, n, g);
892                                 break;
893                             case "16":
894                                 g.FillEllipse(green, 210, 210, 25, 25);
895                                 Thread.Sleep(500);
896                                 TableroColor(r, n, g);
897                                 break;
898                         }
899                     }
900                 }
901                 lectura.Close();
902             }
903             else {
904                 try {
905                     StreamReader lectura = new StreamReader("D:\\
                        Documentos\\ESCOM\\Teoria Computacional\\DatosP5\\
                        RutasA1.txt");
906                     StreamReader lectura2 = new StreamReader("D:\\
                        Documentos\\ESCOM\\Teoria Computacional\\DatosP5\\
                        RutasA2.txt");
907                     string linea;
908                     string linea2;
909 
910                     SolidBrush green = new(Color.Green);
911                     SolidBrush blue = new(Color.Cyan);
912                     while (((linea = lectura.ReadLine()) != null))
913                     {
914                         string[] vs = linea.Split(',');
915                         foreach (string v in vs)
916                         {
917                             switch (v)
918                             {
919                                 case "1":
920                                     g.FillEllipse(green, 60, 60, 25, 25);
921                                     Thread.Sleep(500);
922                                     TableroColor(r, n, g);
923                                     break;
924                                 case "2":
```

```
925                         g.FillEllipse(green, 110, 60, 25, 25);
926                         Thread.Sleep(500);
927                         TableroColor(r, n, g);
928                         break;
929                     case "3":
930                         g.FillEllipse(green, 160, 60, 25, 25);
931                         Thread.Sleep(500);
932                         TableroColor(r, n, g);
933                         break;
934                     case "4":
935                         g.FillEllipse(green, 210, 60, 25, 25);
936                         Thread.Sleep(500);
937                         TableroColor(r, n, g);
938                         break;
939                     case "5":
940                         g.FillEllipse(green, 60, 110, 25, 25);
941                         Thread.Sleep(500);
942                         TableroColor(r, n, g);
943                         break;
944                     case "6":
945                         g.FillEllipse(green, 110, 110, 25, 25)
                                ;
946                         Thread.Sleep(500);
947                         TableroColor(r, n, g);
948                         break;
949                     case "7":
950                         g.FillEllipse(green, 160, 110, 25, 25)
                                ;
951                         Thread.Sleep(500);
952                         TableroColor(r, n, g);
953                         break;
954                     case "8":
955                         g.FillEllipse(green, 210, 110, 25, 25)
                                ;
956                         Thread.Sleep(500);
957                         TableroColor(r, n, g);
958                         break;
959                     case "9":
960                         g.FillEllipse(green, 60, 160, 25, 25);
961                         Thread.Sleep(500);
962                         TableroColor(r, n, g);
963                         break;
964                     case "10":
965                         g.FillEllipse(green, 110, 160, 25, 25)
                                ;
966                         Thread.Sleep(500);
967                         TableroColor(r, n, g);
968                         break;
969                     case "11":
970                         g.FillEllipse(green, 160, 160, 25, 25)
                                ;
971                         Thread.Sleep(500);
972                         TableroColor(r, n, g);
973                         break;
974                     case "12":
975                         g.FillEllipse(green, 210, 160, 25, 25)
                                ;
976                         Thread.Sleep(500);
977                         TableroColor(r, n, g);
978                         break;
```

```
                        case "13":
                            g.FillEllipse(green, 60, 210, 25, 25);
                            Thread.Sleep(500);
                            TableroColor(r, n, g);
                            break;
                        case "14":
                            g.FillEllipse(green, 110, 210, 25, 25)
                                ;
                            Thread.Sleep(500);
                            TableroColor(r, n, g);
                            break;
                        case "15":
                            g.FillEllipse(green, 160, 210, 25, 25)
                                ;
                            Thread.Sleep(500);
                            TableroColor(r, n, g);
                            break;
                        case "16":
                            g.FillEllipse(green, 210, 210, 25, 25)
                                ;
                            Thread.Sleep(500);
                            TableroColor(r, n, g);
                            break;
                    }
                }
            }
            while ((linea2 = lectura2.ReadLine()) != null)
            {
                string[] vs = linea2.Split(',');
                foreach (string v in vs)
                {
                    switch (v)
                    {
                        case "1":
                            g.FillEllipse(blue, 60, 60, 25, 25);
                            Thread.Sleep(500);
                            TableroColor(r, n, g);
                            break;
                        case "2":
                            g.FillEllipse(blue, 110, 60, 25, 25);
                            Thread.Sleep(500);
                            TableroColor(r, n, g);
                            break;
                        case "3":
                            g.FillEllipse(blue, 160, 60, 25, 25);
                            Thread.Sleep(500);
                            TableroColor(r, n, g);
                            break;
                        case "4":
                            g.FillEllipse(blue, 210, 60, 25, 25);
                            Thread.Sleep(500);
                            TableroColor(r, n, g);
                            break;
                        case "5":
                            g.FillEllipse(blue, 60, 110, 25, 25);
                            Thread.Sleep(500);
                            TableroColor(r, n, g);
                            break;
                        case "6":
                            g.FillEllipse(blue, 110, 110, 25, 25);
```

```
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "7":
                                    g.FillEllipse(blue, 160, 110, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "8":
                                    g.FillEllipse(blue, 210, 110, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "9":
                                    g.FillEllipse(blue, 60, 160, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "10":
                                    g.FillEllipse(blue, 110, 160, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "11":
                                    g.FillEllipse(blue, 160, 160, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "12":
                                    g.FillEllipse(blue, 210, 160, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "13":
                                    g.FillEllipse(blue, 60, 210, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "14":
                                    g.FillEllipse(blue, 110, 210, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "15":
                                    g.FillEllipse(blue, 160, 210, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                                case "16":
                                    g.FillEllipse(blue, 210, 210, 25, 25);
                                    Thread.Sleep(500);
                                    TableroColor(r, n, g);
                                    break;
                            }
                        }
                    }
                    lectura.Close();
                    lectura2.Close();
                }
                catch (Exception er) {
```

```
1096                        MessageBox.Show("No existe el archivo, no hay rutas
                               ganadoras");
1097                    }
1098
1099                }
1100            }
1101        }
1102    }
```

## 3.2. Utilidades (Paquete C#)

**Listing 5: Clase Utilidades C# .net 6.0**

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Programa5.UtilidadesC
8  {
9      internal class Utilidades
10     {
11         public string GenerarCadenas()
12         {
13             var tam = new Random();
14             var chars = "rn";
15             var arr = new char[tam.Next(2, 10)];
16             var rnd = new Random();
17             for (int i = 0; i < arr.Length; i ++)
18             {
19                 arr[i] = chars[rnd.Next(chars.Length)];
20             }
21             var v = new String(arr);
22             return v;
23         }
24         public bool CantidadJugadores() {
25             bool res;
26             var p = new Random();
27             if (p.Next(2) == 1){
28                 Console.WriteLine("Un jugador");
29                 res = true;
30             }
31             else {
32                 Console.WriteLine("Dos Jugadores");
33                 res = false;
34             }
35             return res;
36         }
37     }
38 }
```

**Listing 6: Clase Datos C# .net 6.0**

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;

namespace Programa5.UtilidadesC
{
    internal class DatoCad
    {
        public static bool Jugadores;
        public static string Cadena1;
        public static string Cadena2;
    }
}
```

---