



Instituto Politécnico Nacional

# INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE COMPUTO

## PRÁCTICA 7: MÁQUINA DE TURING

*Teoría de la Computación*

Autor:  
Hernández Vergara Eduardo

Junio 2022

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Objetivo . . . . .	2
1.2. Maquina de Turing . . . . .	2
1.3. Como funciona? . . . . .	2
<b>2. Solución</b>	<b>4</b>
2.1. Correr el Programa . . . . .	4
<b>3. Código</b>	<b>7</b>
3.1. El automata . . . . .	7
3.2. Utilidades (Paquete C#) . . . . .	21

# 1. Introducción

## 1.1. Objetivo

En la práctica desarrollada tuvimos que diseñar una máquina de Turing para obtener la suma de  $n + m$ , esto por medio de pipes, veremos como mas adelante.

## 1.2. Maquina de Turing

La máquina de Turing es un dispositivo creado en 1936, que representa un modelo idealizado de computación capaz de almacenar/procesar información virtualmente infinita. El sistema es una abstracción matemática que se construye de un modo extraordinariamente sencillo, pero que facilita la comprobación empiricista de un abanico amplio de preguntas sobre las teorías de la computabilidad y/o de la complejidad. Su ideación marcó un gran hito en la historia de la informática, hasta el punto de ser considerada como el origen de los actuales ordenadores (y de las tecnologías afines, como las tabletas o los teléfonos móviles).

La máquina de Turing es el precursor histórico de los modernos "computadores de programa almacenado", que permiten tanto el guardado de los datos como de los algoritmos sobre los que estos se construyen. Su ventaja, y uno de los factores por los que genera fascinación en los teóricos del cómputo, es su sencillez y sus enormes posibilidades de configuración a nivel técnico; y es que posibilita la experimentación a través de cómo se dispongan sus elementos físicos y se plantee la "pregunta" con la que se programa su uso (mediante algoritmos, que se traducen en una "sucesión" de códigos que se inspiran en el lenguaje lógico). Esta capacidad tan versátil obedece a la naturaleza misma de los datos con los que opera, sujetos a un nivel de abstracción enorme.

De esta manera, la máquina de Turing puede ser programada para ejecutar instrucciones de carácter concreto, que respondan a preguntas más o menos complejas. Todo ello implica que se debe conocer su particular lenguaje, con el objetivo de adaptar a este el algoritmo para su funcionamiento, conscientes de que no existe un código universal para clarificar la totalidad de las incógnitas matemáticas que dormitan en la naturaleza misma (como indica la ley de Church-Turing). Por tanto, el sistema requiere una mente humana detrás, que se pregunte a sí misma la duda a formular y que sepa cómo ha de "dirigirse" al dispositivo para resolverla.

## 1.3. Como funciona?

Junto al fundamento lógico/matemático que se ha descrito, la máquina de Turing requiere una serie de elementos físicos, los cuales tienen la función de ejecutar los comandos introducidos con anterioridad. La disposición de los mismos puede ser diversa, pues existirían casi infinitos diseños de este sistema, pero se requieren los siguientes necesariamente: una cinta de papel o de un material similar, un cabezal móvil cuyo extremo es capaz de realizar trazos (símbolos o números) y un procesador central en el que codificar los algoritmos que se requieran o que faciliten el análisis.

La cinta es el elemento más esencial de todos ellos. No es más que una tira longitudinal, que se divide en una sucesión de cuadros de igual tamaño (o casillas), y cuya longitud dependerá en gran parte del "esfuerzo" que deba llevarse a cabo para solventar la pregunta que plantea

el usuario (pudiendo ser tan corta o tan larga como se estime pertinente). Las casillas están reservadas para que el cabezal trace símbolos distintos (como el 0-1 en el código binario) en cada una, y constituyen el producto de cálculo que habrá de ser comprobado tras su parada. En términos informáticos, estas cintas podrían ser la memoria de un ordenador moderno. Las primeras celdas suelen tener un contenido ya establecido (input), quedando el resto vacías y preparadas para ser ocupadas tras el proceso de computación.

Asímismo, la máquina de Turing consta de un cabezal, un apéndice mecánico (móvil) que se desplaza a la izquierda o la derecha siguiendo la orden que el sistema dispone para él. En su extremo cuenta con una elongación capaz de grabar un trazo sobre la cinta, dando su forma a los números o las figuras que correspondan según el código que determina el movimiento. El modelo original contaba con un cabezal de tecnología rudimentaria, pero el avance en los ámbitos de la robótica ha permitido la irrupción de nuevos diseños más avanzados y precisos. El cabezal "lee" los contenidos de las celdas y se desplaza una única casilla a cualquiera de sus lados (según cuál sea su estado concreto) para seguir ejecutando la instrucción.

En tercer lugar, existe un procesador central al que se destina la función de almacenar código y algoritmos que contienen instrucciones para la actividad del aparato, expresadas siguiendo términos matemáticos y lógicos. Este lenguaje tiene un matiz universal, aunque permite cierto grado de maniobra para introducir expresiones operativas formuladas por el usuario (siempre que haya operativizado el significado). De esta manera, su cabezal facilitaría la ejecución de instrucciones almacenadas en el procesador, que equivaldrían a lo que hoy se conoce como programas o aplicaciones (app). Este sistema permitiría reproducir cualquier cálculo posible y se alzaría como el predecesor de cualquiera de los ordenadores actuales.

## 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

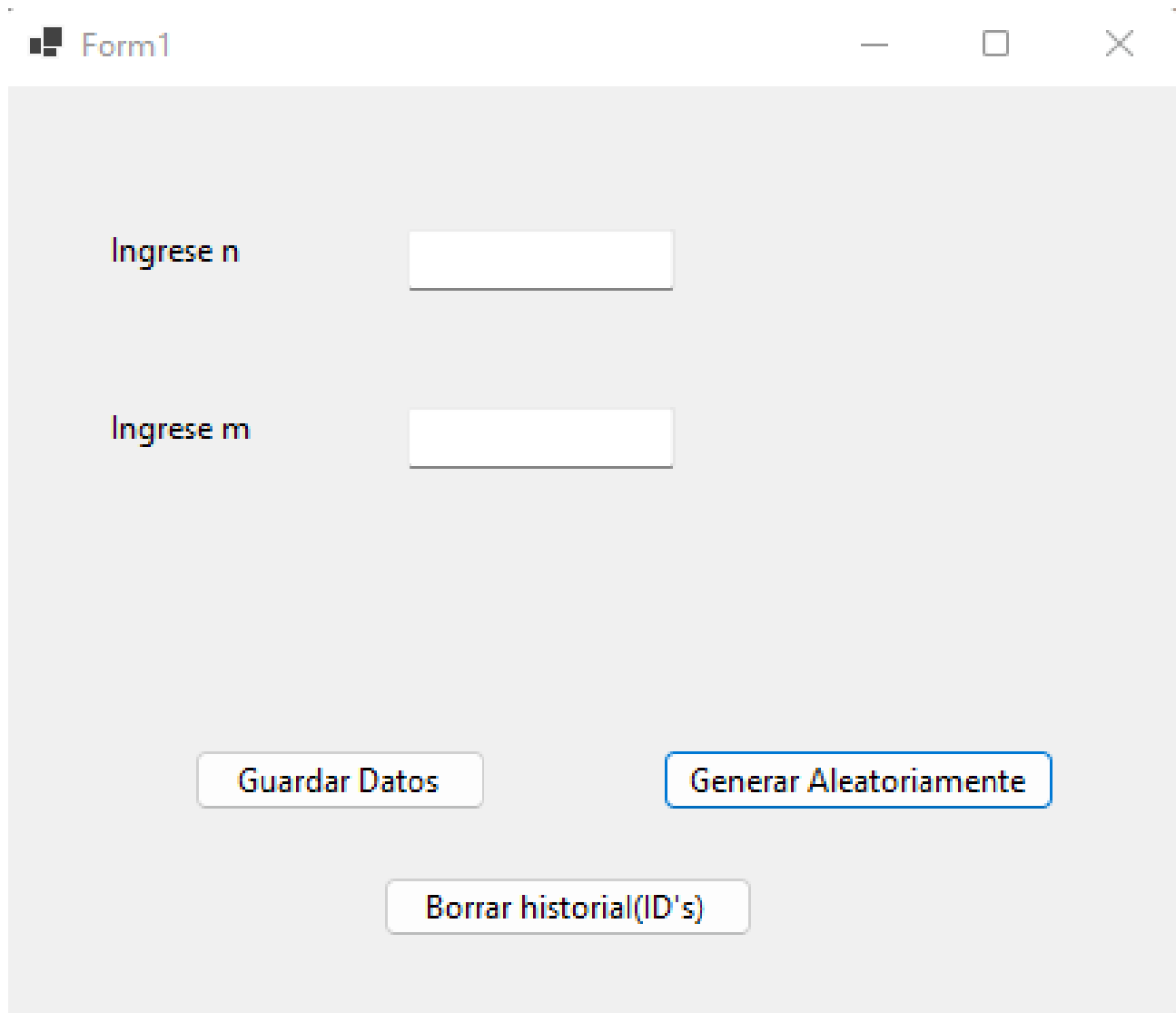


Figura 1: Es el menu principal

Primeramente tenemos nuestro menu principal, donde tenemos un textbox y 3 botones, el boton generar cadena es para generarlo automaticamente, mientras que el guardar cadena es para ingresar la cadena nosotros, el boton borrar historial es para borrar nuestro archivo con ID's.

Al darle clic nos llevara a la graficacion del automata

Aqui podemos ver la animación de la máquina de Turing, claro que no podemos apreciar como cambia de estado por ser una imagen, sin embargo se verá en la revision del programa

The image shows a screenshot of a Windows application window titled "Form1". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main area of the form is light gray. It contains two input fields with labels to their left. The first input field is labeled "Ingrese n" and contains the number "5". The second input field is labeled "Ingrese m" and contains the number "3". Below these input fields, there are three buttons. Two buttons are positioned side-by-side: "Guardar Datos" on the left and "Generar Aleatoriamente" on the right. A third button, "Borrar historial(ID's)", is positioned below the other two, centered horizontally.

Label	Value
Ingrese n	5
Ingrese m	3

Buttons:

- Guardar Datos
- Generar Aleatoriamente
- Borrar historial(ID's)

Figura 2: Le ingresamos una cadena a nuestro menu

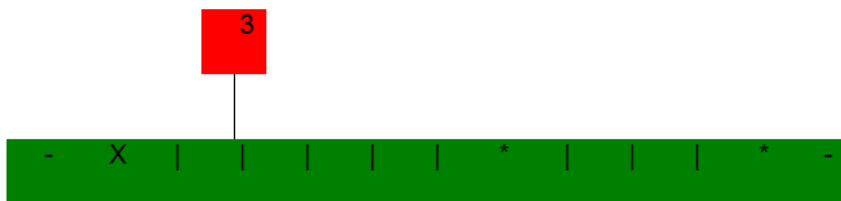


Figura 3: Animacion Automata



Figura 4: Descripcion Instantanea de la maquina de Turing

## 3. Código

### 3.1. El automata

Listing 1: Pantalla Principal de la máquina de Turing C# .net 6.0

```
1 namespace Programa7
2 {
3     public partial class Form1 : Form
4     {
5         public Form1()
6         {
7             InitializeComponent();
8         }
9
10        static int mP, nP;
11        private void Alet_Click(object sender, EventArgs e)
12        {
13            var ra = new Random();
14            n.Text = ra.Next(1,25).ToString();
15            m.Text = ra.Next(1, 25).ToString();
16            mP = Convert.ToInt32(m.Text);
17            nP = Convert.ToInt32(n.Text);
18            Utilidades.Datos.mF = mP;
19            Utilidades.Datos.nF = nP;
20        }
21
22        private void Borrar_Click(object sender, EventArgs e)
23        {
24            bool siono = File.Exists("D:\\Documentos\\ESCOM\\Teoria
                Computacional\\DatosP7\\ID.txt");
25            if (siono == true)
26            {
27                File.Delete("D:\\Documentos\\ESCOM\\Teoria Computacional\\
                DatosP7\\ID.txt");
28            }
29            else {
30                MessageBox.Show("No existe el archivo");
31            }
32        }
33
34        private void Guardar_Click(object sender, EventArgs e)
35        {
36            mP = Convert.ToInt32(m.Text);
37            nP = Convert.ToInt32(n.Text);
38            Utilidades.Datos.mF = mP;
39            Utilidades.Datos.nF = nP;
40            App.Turing turing = new();
41            turing.ShowDialog();
42        }
43    }
44 }
```

Listing 2: Turing 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 Programa7.App
12 {
13     public partial class Turing : Form
14     {
15         public Turing()
16         {
17             InitializeComponent();
18         }
19
20         private void Turing_Paint(object sender, PaintEventArgs e)
21         {
22             Utilidades.UtilidadP utilidad = new();
23             Graphics g = e.Graphics;
24             SolidBrush escr = new(Color.Black);
25             Pen pen = new(escr);
26             Font f = new("Arial", 16);
27             StringFormat fmt = new();
28             SolidBrush verde = new(Color.Green);
29             string cadenaEvaluar;
30             cadenaEvaluar = utilidad.GenerarCadenas();
31             //g.DrawString(cadenaEvaluar, f, escr, 15, 15);
32             DibujarRectangulosTuring(g, verde, cadenaEvaluar, pen, f, fmt)
33             ;
34             Borrar(g);
35             Thread.Sleep(2000);
36             string a = Turinga(g, verde, cadenaEvaluar, pen, f, fmt);
37             DibujarRectangulosTuring(g, verde, a, pen, f, fmt);
38         }
39         private void DibujarRectangulosTuring(Graphics g, SolidBrush verde
40         , string cadenaEvaluar, Pen pen, Font f, StringFormat fmt) {
41             int j = 100;
42             foreach (var c in cadenaEvaluar) {
43                 g.FillRectangle(verde, j, 200, 50, 50);
44                 g.DrawString(c.ToString(), f, (SolidBrush)new(Color.Black)
45                 , j+25, 200);
46                 j +=50;
47             }
48         }
49         private void Borrar(Graphics g) {
50             SolidBrush borra = new(Color.White);
51             g.FillRectangle(borra, 0, 0, 3000, 200);
52         }
53         private void Marcador(Graphics g, int i, Pen pen, Font f,
54         StringFormat fmt) {
55             SolidBrush rojo = new(Color.Red);
56             int k = 100;
57             switch (i) {
58                 case 0:
59                     g.FillRectangle(rojo, k, 100, 50, 50);
60                     g.DrawString("0", f, (SolidBrush)new(Color.Black), k +
61                     25, 100);
62                     g.DrawLine(pen, k + 25, 150, k + 25, 200);
63                     break;
64                 case 1:

```

```

60         g.FillRectangle(rojo, k+50, 100, 50, 50);
61         g.DrawString("1", f, (SolidBrush)new(Color.Black), k +
           75, 100);
62         g.DrawLine(pen, k + 75, 150, k + 75, 200);
63         break;
64     case 2:
65         g.FillRectangle(rojo, k + 100, 100, 50, 50);
66         g.DrawString("2", f, (SolidBrush)new(Color.Black), k +
           125, 100);
67         g.DrawLine(pen, k + 125, 150, k + 125, 200);
68         break;
69     case 3:
70         g.FillRectangle(rojo, k + 150, 100, 50, 50);
71         g.DrawString("3", f, (SolidBrush)new(Color.Black), k +
           175, 100);
72         g.DrawLine(pen, k + 175, 150, k + 175, 200);
73         break;
74     case 4:
75         g.FillRectangle(rojo, k + 200, 100, 50, 50);
76         g.DrawString("4", f, (SolidBrush)new(Color.Black), k +
           225, 100);
77         g.DrawLine(pen, k + 225, 150, k + 225, 200);
78         break;
79     case 5:
80         g.FillRectangle(rojo, k + 250, 100, 50, 50);
81         g.DrawString("5", f, (SolidBrush)new(Color.Black), k +
           275, 100);
82         g.DrawLine(pen, k + 275, 150, k + 275, 200);
83         break;
84     case 6:
85         g.FillRectangle(rojo, k + 300, 100, 50, 50);
86         g.DrawString("6", f, (SolidBrush)new(Color.Black), k +
           325, 100);
87         g.DrawLine(pen, k + 325, 150, k + 325, 200);
88         break;
89     case 7:
90         g.FillRectangle(rojo, k + 350, 100, 50, 50);
91         g.DrawString("7", f, (SolidBrush)new(Color.Black), k +
           375, 100);
92         g.DrawLine(pen, k + 375, 150, k + 375, 200);
93         break;
94     case 8:
95         g.FillRectangle(rojo, k + 400, 100, 50, 50);
96         g.DrawString("8", f, (SolidBrush)new(Color.Black), k +
           425, 100);
97         g.DrawLine(pen, k + 425, 150, k + 425, 200);
98         break;
99     case 9:
100         g.FillRectangle(rojo, k + 450, 100, 50, 50);
101         g.DrawString("9", f, (SolidBrush)new(Color.Black), k +
           475, 100);
102         g.DrawLine(pen, k + 475, 150, k + 475, 200);
103         break;
104     case 10:
105         g.FillRectangle(rojo, k + 500, 100, 50, 50);
106         g.DrawString("10", f, (SolidBrush)new(Color.Black), k
           + 525, 100);
107         g.DrawLine(pen, k + 525, 150, k + 525, 200);
108         break;
109     case 11:

```

```

110         g.FillRectangle(rojo, k + 550, 100, 50, 50);
111         g.DrawString("11", f, (SolidBrush)new(Color.Black), k
112             + 575, 100);
113         g.DrawLine(pen, k + 575, 150, k + 575, 200);
114         break;
115     case 12:
116         g.FillRectangle(rojo, k + 600, 100, 50, 50);
117         g.DrawString("12", f, (SolidBrush)new(Color.Black), k
118             + 625, 100);
119         g.DrawLine(pen, k + 625, 150, k + 625, 200);
120         break;
121     case 13:
122         g.FillRectangle(rojo, k + 650, 100, 50, 50);
123         g.DrawString("13", f, (SolidBrush)new(Color.Black), k
124             + 675, 100);
125         g.DrawLine(pen, k + 675, 150, k + 675, 200);
126         break;
127     case 14:
128         g.FillRectangle(rojo, k + 700, 100, 50, 50);
129         g.DrawString("14", f, (SolidBrush)new(Color.Black), k
130             + 725, 100);
131         g.DrawLine(pen, k + 725, 150, k + 725, 200);
132         break;
133     case 15:
134         g.FillRectangle(rojo, k + 750, 100, 50, 50);
135         g.DrawString("15", f, (SolidBrush)new(Color.Black), k
136             + 775, 100);
137         g.DrawLine(pen, k + 775, 150, k + 775, 200);
138         break;
139     case 16:
140         g.FillRectangle(rojo, k + 800, 100, 50, 50);
141         g.DrawString("16", f, (SolidBrush)new(Color.Black), k
142             + 825, 100);
143         g.DrawLine(pen, k + 825, 150, k + 825, 200);
144         break;
145     case 17:
146         g.FillRectangle(rojo, k + 850, 100, 50, 50);
147         g.DrawString("17", f, (SolidBrush)new(Color.Black), k
148             + 875, 100);
149         g.DrawLine(pen, k + 875, 150, k + 875, 200);
150         break;
151     case 18:
152         g.FillRectangle(rojo, k + 900, 100, 50, 50);
153         g.DrawString("18", f, (SolidBrush)new(Color.Black), k
154             + 925, 100);
155         g.DrawLine(pen, k + 925, 150, k + 925, 200);
156         break;
157     case 19:
158         g.FillRectangle(rojo, k + 950, 100, 50, 50);
159         g.DrawString("19", f, (SolidBrush)new(Color.Black), k
160             + 975, 100);
161         g.DrawLine(pen, k + 975, 150, k + 975, 200);
162         break;
163     case 20:
164         g.FillRectangle(rojo, k + 1000, 100, 50, 50);
165         g.DrawString("20", f, (SolidBrush)new(Color.Black), k
166             + 1025, 100);
167         g.DrawLine(pen, k + 1025, 150, k + 1025, 200);
168         break;
169     case 21:

```

```

160         g.FillRectangle(rojo, k + 1050, 100, 50, 50);
161         g.DrawString("21", f, (SolidBrush)new(Color.Black), k
162             + 1075, 100);
163         g.DrawLine(pen, k + 1075, 150, k + 1075, 200);
164         break;
165     case 22:
166         g.FillRectangle(rojo, k + 1100, 100, 50, 50);
167         g.DrawString("22", f, (SolidBrush)new(Color.Black), k
168             + 1125, 100);
169         g.DrawLine(pen, k + 1125, 150, k + 1125, 200);
170         break;
171     case 23:
172         g.FillRectangle(rojo, k + 1150, 100, 50, 50);
173         g.DrawString("23", f, (SolidBrush)new(Color.Black), k
174             + 1175, 100);
175         g.DrawLine(pen, k + 1175, 150, k + 1175, 200);
176         break;
177     case 24:
178         g.FillRectangle(rojo, k + 1200, 100, 50, 50);
179         g.DrawString("24", f, (SolidBrush)new(Color.Black), k
180             + 1225, 100);
181         g.DrawLine(pen, k + 1225, 150, k + 1225, 200);
182         break;
183     case 25:
184         g.FillRectangle(rojo, k + 1250, 100, 50, 50);
185         g.DrawString("25", f, (SolidBrush)new(Color.Black), k
186             + 1275, 100);
187         g.DrawLine(pen, k + 1275, 150, k + 1275, 200);
188         break;
189     }
190 }
191
192 private string CadenaResponsiva(List<char> k) {
193     string fe = "";
194     foreach (var c in k)
195     {
196         fe += c;
197     }
198     return fe;
199 }
200
201 private void ID(List<char> r, String ruta, String cadenaEvaluar,
202     List<char> valor)
203 {
204     valor = new();
205     string a = "";
206     foreach (var q in r)
207     {
208         a += q;
209     }
210     File.AppendAllText($"{ruta}\\ID.txt", $"({a})+");
211 }
212
213 private string Turinga(Graphics g, SolidBrush verde, string
214     cadenaEvaluar, Pen pen, Font f, StringFormat fmt)
215 {
216     String ruta = "D:\\Documentos\\ESCOM\\Teoria Computacional\\
217     DatosP7";
218     char estado = '1';
219     String cadenaReal, cadMod;
220     List<char> k = new();
221     List<char> r = new();

```

```

212     List<char> valor = new();
213     bool bandera = true;
214     int i = 1;
215     cadenaReal = $"-{cadenaEvaluar}-";
216     foreach (char c in cadenaReal){
217         k.Add(c);
218     }
219     foreach (char c in cadenaReal)
220     {
221         valor.Add(c);
222     }
223     foreach (char c in cadenaReal)
224     {
225         r.Add(c);
226     }
227     while (bandera == true)
228     {
229         switch (estado)
230         {
231             case '1':
232                 switch (k[i]) {
233                     case '*':
234                         k[i] = 'X';
235                         r[i] = 'q';
236                         ID(r, ruta, cadenaEvaluar, valor);
237                         i++;
238                         estado = '2';
239                         if (Utilidades.Datos.nF + Utilidades.Datos
240                             .mF < 10)
241                         {
242                             Thread.Sleep(1000);
243                         }
244                         Borrar(g);
245                         Marcador(g, i, pen, f, fmt);
246                         string a = CadenaResponsiva(k);
247                         r = new();
248                         foreach (char c in a)
249                         {
250                             r.Add(c);
251                         }
252                         DibujarRectangulosTuring(g, verde, a, pen,
253                             f, fmt);
254                         break;
255                     case '2':
256                         switch (k[i])
257                         {
258                             case '*':
259                                 r[i] = 'q';
260                                 ID(r, ruta, cadenaEvaluar, valor);
261                                 i++;
262                                 estado = '3';
263                                 if (Utilidades.Datos.nF + Utilidades.Datos
264                                     .mF < 10)
265                                 {
266                                     Thread.Sleep(1000);
267                                 }
268                                 Borrar(g);
269                                 Marcador(g, i, pen, f, fmt);

```

```

269         string a = CadenaResponsiva(k);
270         r = new();
271         foreach (char c in a)
272         {
273             r.Add(c);
274         }
275         DibujarRectangulosTuring(g, verde, a, pen,
276             f, fmt);
277         break;
278     case '|':
279         r[i] = 'q';
280         ID(r, ruta, cadenaEvaluar, valor);
281         i++;
282         if (Utilidades.Datos.nF + Utilidades.Datos
283             .mF < 10)
284         {
285             Thread.Sleep(1000);
286         }
287         Borrar(g);
288         Marcador(g, i, pen, f, fmt);
289         string b = CadenaResponsiva(k);
290         r = new();
291         foreach (char c in b)
292         {
293             r.Add(c);
294         }
295         DibujarRectangulosTuring(g, verde, b, pen,
296             f, fmt);
297         break;
298     case '3':
299         switch (k[i])
300         {
301             case '*':
302                 k[i] = 'X';
303                 r[i] = 'q';
304                 ID(r, ruta, cadenaEvaluar, valor);
305                 r = new();
306                 i--;
307                 estado = '4';
308                 if (Utilidades.Datos.nF + Utilidades.Datos
309                     .mF < 10)
310                 {
311                     Thread.Sleep(1000);
312                 }
313                 Borrar(g);
314                 Marcador(g, i, pen, f, fmt);
315                 string a = CadenaResponsiva(k);
316                 foreach (char c in a)
317                 {
318                     r.Add(c);
319                 }
320                 DibujarRectangulosTuring(g, verde, a, pen,
321                     f, fmt);
322                 break;
323             case '|':
324                 r[i] = 'q';
325                 ID(r, ruta, cadenaEvaluar, valor);

```

```

324         i++;
325         if (Utilidades.Datos.nF + Utilidades.Datos
            .mF < 10)
326         {
327             Thread.Sleep(1000);
328         }
329         Borrar(g);
330         Marcador(g, i, pen, f, fmt);
331         string b = CadenaResponsiva(k);
332         r = new();
333         foreach (char c in b)
334         {
335             r.Add(c);
336         }
337         DibujarRectangulosTuring(g, verde, b, pen,
            f, fmt);
338         break;
339     }
340     break;
341     case '4':
342         switch (k[i]) {
343             case '*':
344                 r[i] = 'q';
345                 ID(r, ruta, cadenaEvaluar, valor);
346                 i--;
347                 if (Utilidades.Datos.nF + Utilidades.Datos
                    .mF < 10)
348                 {
349                     Thread.Sleep(1000);
350                 }
351                 Borrar(g);
352                 Marcador(g, i, pen, f, fmt);
353                 string c = CadenaResponsiva(k);
354                 r = new();
355                 foreach (char ce in c)
356                 {
357                     r.Add(ce);
358                 }
359                 DibujarRectangulosTuring(g, verde, c, pen,
                    f, fmt);
360                 break;
361             case '|':
362                 //cadenaReal.Replace(cadenaReal[i], 'a');
363                 k[i] = 'a';
364                 r[i] = 'q';
365                 ID(r, ruta, cadenaEvaluar, valor);
366                 i++;
367                 estado = '5';
368                 if (Utilidades.Datos.nF + Utilidades.Datos
                    .mF < 10)
369                 {
370                     Thread.Sleep(1000);
371                 }
372                 Borrar(g);
373                 Marcador(g, i, pen, f, fmt);
374                 string a = CadenaResponsiva(k);
375                 r = new();
376                 foreach (char ce in a)
377                 {
378                     r.Add(ce);

```

```

379         }
380         DibujarRectangulosTuring(g, verde, a, pen,
381                                 f, fmt);
382         break;
383     case 'X':
384         r[i] = 'q';
385         ID(r, ruta, cadenaEvaluar, valor);
386         i++;
387         estado = '7';
388         if (Utilidades.Datos.nF + Utilidades.Datos
389             .mF < 10)
390         {
391             Thread.Sleep(1000);
392         }
393         Borrar(g);
394         Marcador(g, i, pen, f, fmt);
395         string b = CadenaResponsiva(k);
396         r = new();
397         foreach (char ce in b)
398         {
399             r.Add(ce);
400         }
401         DibujarRectangulosTuring(g, verde, b, pen,
402                                 f, fmt);
403         break;
404     case '5':
405         switch (k[i])
406         {
407             case '-':
408                 //cadenaReal.Replace(cadenaReal[i], '|');
409                 k[i] = '|';
410                 r[i] = 'q';
411                 ID(r, ruta, cadenaEvaluar, valor);
412                 k.Add('-');
413                 r.Add('-');
414                 i--;
415                 estado = '6';
416                 if (Utilidades.Datos.nF + Utilidades.Datos
417                     .mF < 10)
418                 {
419                     Thread.Sleep(1000);
420                 }
421                 Borrar(g);
422                 Marcador(g, i, pen, f, fmt);
423                 string a = CadenaResponsiva(k);
424                 r = new();
425                 foreach (char ce in a)
426                 {
427                     r.Add(ce);
428                 }
429                 DibujarRectangulosTuring(g, verde, a, pen,
430                                         f, fmt);
431                 break;
432             case '*':
433                 r[i] = 'q';
434                 ID(r, ruta, cadenaEvaluar, valor);
435                 i++;
436                 if (Utilidades.Datos.nF + Utilidades.Datos

```



```

434         .mF < 10)
435     {
436         Thread.Sleep(1000);
437     }
438     Borrar(g);
439     Marcador(g, i, pen, f, fmt);
440     string b = CadenaResponsiva(k);
441     r = new();
442     foreach (char ce in b)
443     {
444         r.Add(ce);
445     }
446     DibujarRectangulosTuring(g, verde, b, pen,
447         f, fmt);
448     break;
449 case '|':
450     r[i] = 'q';
451     ID(r, ruta, cadenaEvaluar, valor);
452     i++;
453     if (Utilidades.Datos.nF + Utilidades.Datos
454         .mF < 10)
455     {
456         Thread.Sleep(1000);
457     }
458     Borrar(g);
459     Marcador(g, i, pen, f, fmt);
460     string c = CadenaResponsiva(k);
461     r = new();
462     foreach (char ce in c)
463     {
464         r.Add(ce);
465     }
466     DibujarRectangulosTuring(g, verde, c, pen,
467         f, fmt);
468     break;
469 case 'X':
470     r[i] = 'q';
471     ID(r, ruta, cadenaEvaluar, valor);
472     i++;
473     if (Utilidades.Datos.nF + Utilidades.Datos
474         .mF < 10)
475     {
476         Thread.Sleep(1000);
477     }
478     Borrar(g);
479     Marcador(g, i, pen, f, fmt);
480     string d = CadenaResponsiva(k);
481     r = new();
482     foreach (char ce in d)
483     {
484         r.Add(ce);
485     }
486     DibujarRectangulosTuring(g, verde, d, pen,
487         f, fmt);
488     break;
489 }
490 break;
491 case '6':
492     switch (k[i])
493     {

```

```

488     case '*':
489         r[i] = 'q';
490         ID(r, ruta, cadenaEvaluar, valor);
491         i--;
492         if (Utilidades.Datos.nF + Utilidades.Datos
493             .mF < 10)
494         {
495             Thread.Sleep(1000);
496         }
497         Borrar(g);
498         Marcador(g, i, pen, f, fmt);
499         string a = CadenaResponsiva(k);
500         r = new();
501         foreach (char ce in a)
502         {
503             r.Add(ce);
504         }
505         DibujarRectangulosTuring(g, verde, a, pen,
506             f, fmt);
507         break;
508     case '|':
509         r[i] = 'q';
510         ID(r, ruta, cadenaEvaluar, valor);
511         i--;
512         if (Utilidades.Datos.nF + Utilidades.Datos
513             .mF < 10)
514         {
515             Thread.Sleep(1000);
516         }
517         Borrar(g);
518         Marcador(g, i, pen, f, fmt);
519         string b = CadenaResponsiva(k);
520         r = new();
521         foreach (char ce in b)
522         {
523             r.Add(ce);
524         }
525         DibujarRectangulosTuring(g, verde, b, pen,
526             f, fmt);
527         break;
528     case 'a':
529         r[i] = 'q';
530         ID(r, ruta, cadenaEvaluar, valor);
531         k[i] = '|';
532         i--;
533         estado = '4';
534         if (Utilidades.Datos.nF + Utilidades.Datos
535             .mF < 10)
536         {
537             Thread.Sleep(1000);
538         }
539         Borrar(g);
540         Marcador(g, i, pen, f, fmt);
541         string c = CadenaResponsiva(k);
542         r = new();
543         foreach (char ce in c)
544         {
545             r.Add(ce);
546         }
547         DibujarRectangulosTuring(g, verde, c, pen,

```

```

        f, fmt);
543     break;
544 case 'X':
545     r[i] = 'q';
546     ID(r, ruta, cadenaEvaluar, valor);
547     r = new();
548     i--;
549     if (Utilidades.Datos.nF + Utilidades.Datos
        .mF < 10)
550     {
551         Thread.Sleep(1000);
552     }
553     Borrar(g);
554     Marcador(g, i, pen, f, fmt);
555     string d = CadenaResponsiva(k);
556     foreach (char ce in d)
557     {
558         r.Add(ce);
559     }
560     DibujarRectangulosTuring(g, verde, d, pen,
        f, fmt);
561     break;
562 }
563 break;
564 case '7':
565     switch (k[i])
566     {
567         case '*':
568             r[i] = 'q';
569             ID(r, ruta, cadenaEvaluar, valor);
570             i++;
571             estado = '8';
572             if (Utilidades.Datos.nF + Utilidades.Datos
                .mF < 10)
573             {
574                 Thread.Sleep(1000);
575             }
576             Borrar(g);
577             Marcador(g, i, pen, f, fmt);
578             string a = CadenaResponsiva(k);
579             r = new();
580             foreach (char ce in a)
581             {
582                 r.Add(ce);
583             }
584             DibujarRectangulosTuring(g, verde, a, pen,
                f, fmt);
585             break;
586         case '|':
587             r[i] = 'q';
588             ID(r, ruta, cadenaEvaluar, valor);
589             i++;
590             if (Utilidades.Datos.nF + Utilidades.Datos
                .mF < 10)
591             {
592                 Thread.Sleep(1000);
593             }
594             Borrar(g);
595             Marcador(g, i, pen, f, fmt);
596             string b = CadenaResponsiva(k);

```

```

597         r = new();
598         foreach (char ce in b)
599         {
600             r.Add(ce);
601         }
602         DibujarRectangulosTuring(g, verde, b, pen,
603             f, fmt);
604         break;
605     }
606     break;
607 case '8':
608     switch (k[i])
609     {
610         case '-':
611             r[i] = 'q';
612             ID(r, ruta, cadenaEvaluar, valor);
613             k[i] = '*';
614             i--;
615             estado = '9';
616             if (Utilidades.Datos.nF + Utilidades.Datos
617                 .mF < 10)
618             {
619                 Thread.Sleep(1000);
620             }
621             Borrar(g);
622             Marcador(g, i, pen, f, fmt);
623             string a = CadenaResponsiva(k);
624             r = new();
625             foreach (char ce in a)
626             {
627                 r.Add(ce);
628             }
629             DibujarRectangulosTuring(g, verde, a, pen,
630                 f, fmt);
631             break;
632         case '|':
633             r[i] = 'q';
634             ID(r, ruta, cadenaEvaluar, valor);
635             i++;
636             if (Utilidades.Datos.nF + Utilidades.Datos
637                 .mF < 10)
638             {
639                 Thread.Sleep(1000);
640             }
641             Borrar(g);
642             Marcador(g, i, pen, f, fmt);
643             string b = CadenaResponsiva(k);
644             r = new();
645             foreach (char ce in b)
646             {
647                 r.Add(ce);
648             }
649             DibujarRectangulosTuring(g, verde, b, pen,
650                 f, fmt);
651             break;
652         case 'X':
653             r[i] = 'q';
654             ID(r, ruta, cadenaEvaluar, valor);
655             k[i] = '*';
656             i++;

```

```

652         if (Utilidades.Datos.nF + Utilidades.Datos
653             .mF < 10)
654         {
655             Thread.Sleep(1000);
656         }
657         Borrar(g);
658         Marcador(g, i, pen, f, fmt);
659         string c = CadenaResponsiva(k);
660         r = new();
661         foreach (char ce in c)
662         {
663             r.Add(ce);
664         }
665         DibujarRectangulosTuring(g, verde, c, pen,
666             f, fmt);
667         break;
668     case '9':
669         switch (k[i])
670         {
671             case '*':
672                 r[i] = 'q';
673                 ID(r, ruta, cadenaEvaluar, valor);
674                 i--;
675                 if (Utilidades.Datos.nF + Utilidades.Datos
676                     .mF < 10)
677                 {
678                     Thread.Sleep(1000);
679                 }
680                 Borrar(g);
681                 Marcador(g, i, pen, f, fmt);
682                 string a = CadenaResponsiva(k);
683                 r = new();
684                 foreach (char ce in a)
685                 {
686                     r.Add(ce);
687                 }
688                 DibujarRectangulosTuring(g, verde, a, pen,
689                     f, fmt);
690                 break;
691             case '|':
692                 r[i] = 'q';
693                 ID(r, ruta, cadenaEvaluar, valor);
694                 i--;
695                 if (Utilidades.Datos.nF + Utilidades.Datos
696                     .mF < 10)
697                 {
698                     Thread.Sleep(1000);
699                 }
700                 Borrar(g);
701                 Marcador(g, i, pen, f, fmt);
702                 string b = CadenaResponsiva(k);
703                 r = new();
704                 foreach (char ce in b)
705                 {
706                     r.Add(ce);
707                 }
708                 DibujarRectangulosTuring(g, verde, b, pen,
709                     f, fmt);

```

```

706         break;
707     case 'X':
708         //cadenaReal.Replace(cadenaReal[i], '*');
709         k[i] = '*';
710         r[i] = 'f';
711         ID(r, ruta, cadenaEvaluar, valor);
712         k.Add('-');
713         r.Add('-');
714         bandera = false;
715         if (Utilidades.Datos.nF + Utilidades.Datos
            .mF < 10)
716         {
717             Thread.Sleep(1000);
718         }
719         Borrar(g);
720         Marcador(g, i, pen, f, fmt);
721         string c = CadenaResponsiva(k);
722         r = new();
723         foreach (char ce in c)
724         {
725             r.Add(ce);
726         }
727         DibujarRectangulosTuring(g, verde, c, pen,
            f, fmt);
728         break;
729     }
730     break;
731 }
732
733 }
734 string fe = "";
735 foreach (var c in k)
736 {
737     fe += c;
738 }
739 Console.Beep(100, 5000);
740 return fe;
741 }
742 }
743 }

```

## 3.2. Utilidades (Paquete C#)

**Listing 3: 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 Programa7.Utilidades
8 {
9     internal class UtilidadP
10    {
11        public string GenerarCadenas() {
12            var cad = "*";
13            var aux = "";

```

```

14         var aux2 = "";
15         for (int i = 0; i < Datos.nF; i++) {
16             aux += "|";
17         }
18         for (int i = 0; i < Datos.mF; i++){
19             aux2 += "|";
20         }
21         string cadeF = $"{cad}{aux}{cad}{aux2}{cad}";
22         return cadeF;
23     }
24 }
25 }

```

---

#### Listing 4: Clase Datos 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 Programa7.Utilidades
8 {
9     internal class Datos
10     {
11         public static int nF;
12         public static int mF;
13     }
14 }

```

---