



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

“JUEGO DEL GATO EN LENGUAJE C”

PROYECTO FINAL

MATERIA:
FUNDAMENTOS DE
PROGRAMACIÓN

ALUMNA: CAROLINA VARGAS LÓPEZ

PROFESOR: MARCO ANTONIO MARTÍNEZ QUINTANA

FECHA DE ELABORACIÓN: 22 DE NOVIEMBRE DEL
2021 A 10 DICIEMBRE DEL 2021

SEMESTRE 2022-1

Proyecto final: “Juego del gato en lenguaje C”

En el siguiente documento, se dará una pequeña introducción a la historia del “Juego del gato”, como se conoce en México, con el fin de saber sus orígenes y cómo es su funcionamiento. Además, conoceremos porque son tan importantes los juegos en la infancia, y cómo contribuyen al buen desarrollo de los niños que posteriormente se convertirán en adultos.

A partir de esta información, se llevará a cabo su automatización, es decir, se llevará del lápiz y papel a la pantalla de tu computadora. Con esto, se busca ejemplificar que cualquier proceso puede llevarse a su automatización y por ende hacer uso de la programación para fines recreativos, escolares, laborales, etc.

Durante la lectura, te encontrarás con el algoritmo, diagrama de flujo y distintos procesos que se tuvieron que seguir para el correcto funcionamiento del programa, además se mostrarán los recursos informáticos utilizados durante la realización del proyecto, también una tabla de costos asociados al mismo y un diagrama de Gantt que refleja las actividades realizadas y sus fechas correspondientes.

Se mostrarán en capturas de pantalla el funcionamiento del juego, y los diferentes resultados que obtendremos cuando los participantes comiencen a jugar, esto con el fin de saber que es lo que pasará si, por ejemplo, el jugador #1 gana el juego; el jugador #2 gana el juego; o se llega a un empate. De manera que, observaremos cómo se resolvieron estos problemas utilizando los temas vistos en clase y conociendo los temas básicos del lenguaje de programación en C.

También, se dará una descripción de los diferentes recursos de programación en lenguaje C que se utilizaron, como las estructuras de control selectivas *if*, *if-else* y las estructuras de control repetitivas *do-while* y *for*. Las cuales, como ya se mencionó, sirvieron para llevar a cabo de manera correcta el funcionamiento del proyecto.

Para llevar a cabo este proyecto, se usó de referencia el canal de YouTube de Luis Fernando, en los videos que llevan por nombre “Juego del gato en C en 30 min” los cuales constan de 3 partes, durante estos videos se muestra la codificación y funcionamiento del juego. Estos videos fueron de gran utilidad y la base para llevar a cabo el juego realizado, ya que se siguió la estructura mostrada en los videos, con algunas implementaciones realizadas posteriormente.

Con esto, se busca que el lector se interese por jugar el juego del gato a través de una terminal, en vez de hacerlo tallando una piedra o en lápiz y papel. Más adelante sabrás porque se habla de tallar una piedra.

Sin más que agregar, se dará paso a conocer que hay detrás de este juego tan famoso y que todos hemos jugado en algún punto de nuestra vida.

La historia del juego del gato

También conocido como el juego del tres en raya, tres en línea o tic tac toe, tiene mucha más historia de la que se piensa.

Los juegos de mesa surgieron en el Neolítico y fueron evolucionando en la Edad de Bronce.

La historia del juego del gato, se remonta a la lejana Persia hace casi mil años, desde donde mercederes italianos lo exportaron a sus tierras y lo extendieron. En poco tiempo se convirtió en uno de los juegos más populares de las clases bajas en la Edad Media. De hecho, en España se conservan numerosas pruebas de lo extendido que estaba este juego.

Como el juego es muy fácil de recrear su rápida expansión fue muy notoria. Tanto que hasta el día de hoy lo seguimos jugando.

Cabe mencionar que tuvo un periodo negro, entre los siglos X y XI, debido a su asociación con rituales paganos y su descalificación por parte de las clases altas de la sociedad. Tal fue su importancia y fama que el Papa Bonifacio VI creó una bula papal que prohibía su práctica para todos los cristianos, de manera que el juego era considerado un elemento de la magia negra o pagana.

Después de un tiempo, su práctica fue resurgiendo poco a poco en las universidades, volviéndose muy popular en Salamanca y Bolonia.

Fue en el Renacimiento cuando este juego volvió a formar parte de la cultura popular y se extendió como nunca antes, sin embargo, no se siguió jugando en el medievo.

En España hay restos de tableros tallados en piedra, los cuales están ubicados donde las clases bajas solían pasar su tiempo libre.

El tablero que a continuación se muestra, está ubicado en la parte más alta de la iglesia barcelonesa de Santa María del Mar, esto da la impresión de que era una pasatiempo para los guardianes de aquella iglesia.



¿Cómo se juega?

El juego requiere de dos participantes, el jugador 1 y el jugador 2. Se lleva a cabo en una cuadrícula de 3 filas y 3 columnas.

El jugador 1 tiene asignada la letra “X”, mientras que el jugador 2 tiene asignada la letra “O”.

El objetivo del juego es llenar una fila, columna o diagonal con la letra que tiene asignada. Como en la imagen que se muestra a continuación:



De esta manera, si el jugador 1 llena una fila, columna o diagonal con la letra “X” gana el juego, y análogamente; si el jugador 2 llena una fila, columna o diagonal con la letra “O” entonces el jugador 2 gana el juego.

Si se ha llenado el tablero y ninguno de los participantes logró llenar una fila, columna o diagonal con su figura asignada, entonces el juego termina y es un empate entre los jugadores.

Algo interesante sobre este juego es que la mayoría de las probabilidades se inclinan a que el juego terminará empatado, por lo que ha servido como herramienta pedagógica, sobre todo en niños pequeños, sobre cómo divertirse sin que tenga que existir un ganador y por consiguiente, un perdedor.

Ya que conocemos un poco más sobre este juego, parece increíble cuánto tiempo ha pasado después de su creación y que hasta la fecha lo sigamos jugando y nos siga divirtiendo.

Pero ahora, no es necesario tallar un tablero en piedras como antes lo hacían, y tampoco ya es necesario un lápiz y papel, sino que ahora lo podemos hacer mediante la computadora, es decir lo automatizamos. Esto refleja lo mucho que hemos avanzado en la tecnología, aunque se puede considerar algo bastante simple a lo que hoy se puede realizar gracias a los avances tecnológicos, sirve como ejemplo para dimensionar lo mucho que hemos logrado.

Desarrollo del proyecto

De manera que, este proyecto busca visibilizar que todo proceso, así sea el más simple o complejo, puede ser automatizado y realizarse desde una computadora.

Además, de que tiene el propósito de recordar la importancia que tienen los juegos en la vida del ser humano, pues estos a parte de hacer que compartamos un rato divertido con nuestros seres queridos, también influyen en el aprendizaje y adaptación social de los niños, ya que ayudan a desarrollar las capacidades motoras, emocionales, mentales y sensoriales.

Además, aportan creatividad; estimulan la empatía del ser humano; ayudan a tolerar la frustración, a tener paciencia y a esperar turno; a desarrollar la orientación espacial; a fortalecer la autonomía, despiertan la imaginación; estimulan el lenguaje; ayudan a establecer estrategias y a tomar decisiones, también mejoran la memoria y la concentración.

Es decir, podemos considerar a los juegos como una forma de inversión a largo plazo, ya que las personas van a estar interesadas en el buen desarrollo y desempeño de sus hijos e incluso de ellos mismos. Por eso, si podemos crear más juegos que puedan ayudar y complementar al buen desarrollo cognitivo y

analítico, tendremos más niños y niñas con mejor comprensión, por ende, en el futuro tendremos jóvenes adultos que lleven al país por un buen camino.

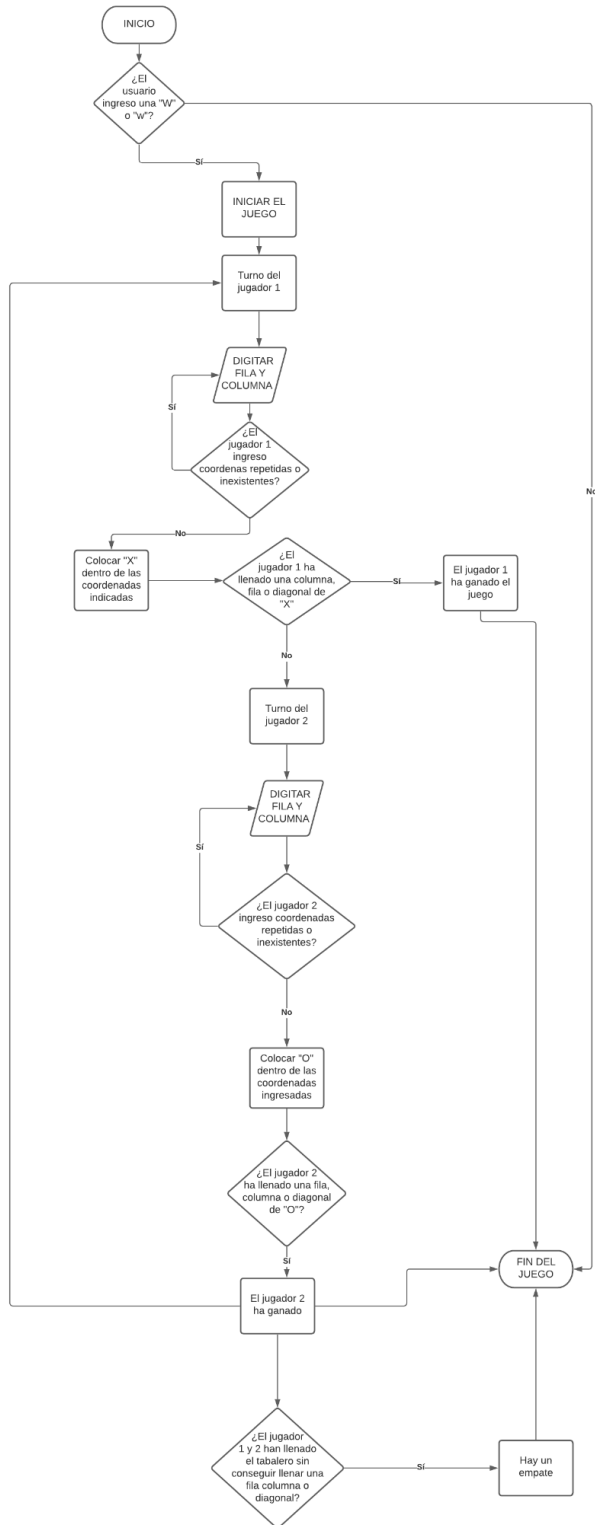
A continuación, veremos las distintas etapas y procesos que se tuvieron que seguir para el desarrollo del juego.

Para poder llevar este juego a la terminal de una computadora, fue necesario primero realizar un **algoritmo**, como el que se muestra a continuación:

1. Pedir al usuario que ingrese "W" o "w" para iniciar el juego sino salir del juego
2. Si el usuario presiona la tecla "W" o "w" entonces comienza el juego, si el usuario ingresa una letra diferente, entonces el juego no empieza.
3. El juego empieza con el jugador 1, el cual tendrá asignado el carácter "X".
4. Pedir al usuario que ingrese la fila y la columna en la cual quiere que su carácter se ubique
5. Si el jugador ingresa coordenadas repetidas o que no existen, entonces volver a preguntar hasta que el usuario digite coordenadas correctas
6. Si el jugador 1 llena con "X" una fila, columna o diagonal, entonces el ganador es el jugador 1.
7. Si el jugador 1 aún no llena una columna, fila, o diagonal de "X", entonces será el turno del jugador 2.
8. Preguntar al jugador 2 las coordenadas en las que quiere que su carácter "O" se ubique.
9. Si el jugador 2 digita coordenadas repetidas o inexistentes, entonces volver a pedir coordenadas hasta que el jugador 2 digite las coordenadas correctas.
10. Si el jugador 2 llena con "O" una fila, columna o diagonal, entonces el ganador del juego es el jugador 2.
11. Si el jugador 2 aún no llena una columna, fila o diagonal del "O", entonces vuelve a ser el turno del jugador 1.
12. Si el jugador 1 y el jugador 2 han llenado todas las casillas y ninguno de los dos consiguió completar una fila, columna o diagonal, entonces será un empate.

Este algoritmo describe los pasos que se deben seguir para el funcionamiento del juego, el cual junto con el diagrama de flujo, ayudaron a codificarlo de manera correcta.

A continuación, el **diagrama de flujo** que se construyó con base al algoritmo:



Pseudocódigo

INICIO

n, m, fila, col, ganador =0, turno=1 : ENTERO
matriz [3][3], opc = CARÁCTER

ESCRIBIR

“Bienvenido al juego del gato!”

ESCRIBIR

“Presione W para jugar / V para salir:”

LEER

opc

SI opc es = “W” || opc = “w”

ENTONCES

HACER

SI (turno%2==1)

ENTONCES

ESCRIBIR “Jugador 1:”

“Digite fila”

LEER fila

ESCRIBIR

“Digite columna”:

LEER col

SI matriz (matriz [fila][col] = ‘x’ || matriz[fila][col] = ‘o’ || fila > 2 || col>2)

ENTONCES

ESCRIBIR “Coordenadas no válidas, pruebe otra vez”

MIENTRAS (matriz [fila][col] = ‘x’ || matriz[fila][col] = ‘o’ || fila > 2 ||
col>2);

FIN

ESCRIBIR matriz[fila][col]=‘x’

TAMBIÉN SI (turno%2==0)

ENTONCES

HACER

ESCRIBIR

```

        "Jugador 2:"
        "Digite fila:"
        LEER fila
    ESCRIBIR
        "Digite columna"
        LEER col

    SI (matriz[fila][col]== 'x' || matriz[fila][col] == 'o' || fila > 2 || col >
2 )
        ENTONCES
            ESCRIBIR "Coordenadas no válidas pruebe otra vez"

    MIENTRAS (matriz[fila][col]== 'x' || matriz[fila][col] == 'o' || fila > 2 || col
> 2 ) FIN

    ESCRIBIR matriz[fila][col]='o'

    SI ( matriz[0][0]== 'x' && matriz[0][0] == matriz[0][1] &&
matriz[0][0]== matriz[0][2]
|| matriz[1][0]== 'x' && matriz[1][0] == matriz[1][1] && matriz[1][0]==
matriz[1][2]
|| matriz[2][0]== 'x' && matriz[2][0] == matriz[2][1] && matriz[2][0]==
matriz[2][2]
||matriz[0][0]== 'x' && matriz[0][0] == matriz[1][0] && matriz[0][0]==
matriz[2][0]
||matriz[0][1]== 'x' && matriz[0][1] == matriz[1][1] && matriz[0][1]==
matriz[2][1]
||matriz[0][2]== 'x' && matriz[0][2] == matriz[1][2] && matriz[0][2]==
matriz[2][2]
||matriz[0][0]== 'x' && matriz[0][0] == matriz[1][1] && matriz[0][0]==
matriz[2][2]
||matriz[0][2]== 'x' && matriz[0][2] == matriz[1][1] && matriz[0][2]==
matriz[2][0] )

        ENTONCES
            ESCRIBIR "Felicidades! ganó el jugador #1

    SI (matriz[0][0]== 'o' && matriz[0][0] == matriz[0][1] && matriz[0][0]==
matriz[0][2]

```

```
|| matriz[1][0]== 'o' && matriz[1][0] ==matriz[1][1] && matriz[1][0]==  
matriz[1][2]  
|| matriz[2][0]== 'o' && matriz[2][0] == matriz[2][1] && matriz[2][0]==  
matriz[2][2]  
||matriz[0][0]== 'o' && matriz[0][0] == matriz[1][0] && matriz[0][0]==  
matriz[2][0]  
||matriz[0][1]== 'o' && matriz[0][1] == matriz[1][1] && matriz[0][1]==  
matriz[2][1]  
||matriz[0][2]== 'o' && matriz[0][2] == matriz[1][2] && matriz[0][2]==  
matriz[2][2]  
||matriz[0][0]== 'o' && matriz[0][0] == matriz[1][1] && matriz[0][0]==  
matriz[2][2]  
||matriz[0][2]== 'o' && matriz[0][2] == matriz[1][1] && matriz[0][2]==  
matriz[2][0] )
```

ENTONCES

ESCRIBIR “Felicidades! Ganó el jugador #2

FIN

SINO

ENTONCES

ESCRIBIR “Ok entonces no hay juego, hasta pronto”

Código fuente

De esta manera fue como se llegó al código fuente del programa, en donde podemos observar que se encuentra con distintos comentarios, que especifican lo que se realizó y así poder entender de mejor manera la codificación realizada.

El código fuente se muestra en las siguientes capturas de pantalla:

```

1  #include<stdio.h>
2  int main()
3  {
4      //Creamos matriz
5
6      char matriz[3][3], opc;
7      int n, m, fila, col, ganador=0, turno=1; // Declaramos variables
8      //Mensaje de bienvenida
9      printf("\nHola jugadores, bienvenidos al juego del gato!\n");
10
11
12      for(n=0; n<3; n++)
13      { //representa a la fila de la matriz
14          for(m=0; m<3; m++)
15          { //representa a la columna de la matriz
16              matriz[n][m]=' ';
17              printf("[%c]",matriz[n][m]); //Imprime la matriz
18          }
19          printf("\n");
20      }
21      printf("\n\nPresione W para jugar / V para salir: ");
22      scanf("%c",&opc);
23      if (opc == 'W' || opc == 'w'){
24
25          do{ //Empieza el juego
26              if(turno%2==1){ //Empieza con el jugador 1
27                  do{
28                      printf("\nJugador 1: \n");
29                      printf("Digite fila:");
30                      scanf("%d", &fila);
31                      printf("Digite columna:");
32                      scanf("%d",&col);
33
34                      if(matriz[fila][col]== 'x' || matriz[fila][col]== 'o' || fila > 2 || col > 2 )
35                      {
36                          printf("\nCoordenadas no válidas, pruebe otra vez\n");
37                      }
38
39                      }while(matriz[fila][col]== 'x' || matriz[fila][col]== 'o' || fila > 2 || col > 2 );
40
41                      matriz[fila][col]='x'; //Sale del bucle e imprime la matriz
42
43
44                      for(n=0; n<3; n++)// Representa fila de la matriz
45                      {
46                          for(m=0; m<3; m++) //Representa columna de la matriz
47                          {
48                              printf("[%c]",matriz[n][m]); //Imprime la matriz
49                          }
50                          printf("\n");
51                      }
52                      turno ++; // Contador que va sumando de 1 en 1
53              } else
54              {
55                  if(turno%2==0)
56                  { //Empieza con el jugador 2
57                      do{
58                          printf("\nJugador 2: \n");
59                          printf("Digite fila:");
60                          scanf("%d", &fila);
61                          printf("Digite columna:");
62                          scanf("%d",&col);
63
64                          if(matriz[fila][col]== 'x' || matriz[fila][col]== 'o' || fila > 2 || col > 2 )
65                          {
66                              printf("\nCoordenadas no válidas, pruebe otra vez\n");
67                          }
68

```

```

65         {
66             printf("\nCoordenadas no válidas, pruebe otra vez\n");
67         }
68     }
69     while(matriz[fila][col] == 'x' || matriz[fila][col] == 'o' || fila > 2 || col > 2 );
70     matriz[fila][col] = 'o'; //Sale del bucle e imprime la matriz
71
72     for(n=0; n<3; n++) //Representa fila de la matriz
73     {
74         for(m=0; m<3; m++) //Representa columna de la matriz
75         {
76             printf("[%c]", matriz[n][m]);
77         }
78         printf("\n");
79     }
80     turno ++;
81 }
82 //Comprobaciones para el ganador
83 if(matriz[0][0] == 'x' && matriz[0][0] == matriz[0][1] && matriz[0][0] == matriz[0][2]
84 || matriz[1][0] == 'x' && matriz[1][0] == matriz[1][1] && matriz[1][0] == matriz[1][2]
85 || matriz[2][0] == 'x' && matriz[2][0] == matriz[2][1] && matriz[2][0] == matriz[2][2]
86 || matriz[0][0] == 'x' && matriz[0][0] == matriz[1][0] && matriz[0][0] == matriz[2][0]
87 || matriz[0][1] == 'x' && matriz[0][1] == matriz[1][1] && matriz[0][1] == matriz[2][1]
88 || matriz[0][2] == 'x' && matriz[0][2] == matriz[1][2] && matriz[0][2] == matriz[2][2]
89 || matriz[0][0] == 'x' && matriz[0][0] == matriz[1][1] && matriz[0][0] == matriz[2][2]
90 || matriz[0][2] == 'x' && matriz[0][2] == matriz[1][1] && matriz[0][2] == matriz[2][0] )
91 {
92     ganador=1;
93     printf("\nFelicidades! Ganó el jugador #1\n");
94 }
95
96 if(matriz[0][0] == 'o' && matriz[0][0] == matriz[0][1] && matriz[0][0] == matriz[0][2]
97 || matriz[1][0] == 'o' && matriz[1][0] == matriz[1][1] && matriz[1][0] == matriz[1][2]
98 || matriz[2][0] == 'o' && matriz[2][0] == matriz[2][1] && matriz[2][0] == matriz[2][2]
99 || matriz[0][0] == 'o' && matriz[0][0] == matriz[1][0] && matriz[0][0] == matriz[2][0]
100 || matriz[0][1] == 'o' && matriz[0][1] == matriz[1][1] && matriz[0][1] == matriz[2][1]
101 || matriz[0][2] == 'o' && matriz[0][2] == matriz[1][2] && matriz[0][2] == matriz[2][2]
102 || matriz[0][0] == 'o' && matriz[0][0] == matriz[1][1] && matriz[0][0] == matriz[2][2]
103 || matriz[0][2] == 'o' && matriz[0][2] == matriz[1][1] && matriz[0][2] == matriz[2][0] ){
104     ganador=1;
105     printf("\nFelicidades! Ganó el jugador #2\n");
106 }
107 if (ganador==0 && turno==10)
108 {
109     printf("\nHay un empate!!\n");
110     ganador=1;
111 }
112 while(ganador != 1);
113
114 } else
115 {
116     printf("\nAnímense a jugar, es divertido!!\n");
117 }
118 return 0;
119 }

```

Resultados del proyecto

A continuación, se muestra el funcionamiento del juego:

En el primer momento, el programa pregunta al usuario o usuarios, si quieren jugar, de manera que si el usuario ingresa una “W” o “w”, entonces el juego inicia, y si no el resultado es el siguiente:

```
iMac-de-Sonia:ejemplos zonalv$ ./proyecto.out

Hola jugadores, bienvenidos al juego del gato!
[ ][ ]
[ ][ ]
[ ][ ]

Presione W para jugar / V para salir: v

Animense a jugar, es divertido!:)
iMac-de-Sonia:ejemplos zonalv$ █
```

Si el usuario presiona “W” o “w” entonces inicia el juego y se obtienen los siguientes resultados.

Tenemos el caso de cómo se vería la terminal de nuestra computadora cuando el jugador #1 es quien gana el juego:

```
Hola jugadores, bienvenidos al juego del gato!
[ ][ ]
[ ][ ]
[ ][ ]

Presione W para jugar / V para salir: w

Jugador 1:
Digite fila:1
Digite columna:0
[ ][ ]
[x][ ][ ]
[ ][ ][ ]

Jugador 2:
Digite fila:1
Digite columna:1
[ ][ ][ ]
[x][o][ ]
[ ][ ][ ]

Jugador 1:
Digite fila:0
Digite columna:0
[x][ ][ ]
[x][o][ ]
[ ][ ][ ]

Jugador 2:
Digite fila:2
Digite columna:2
[x][ ][ ]
[x][o][ ]
[ ][ ][o]

Jugador 1:
Digite fila:2
Digite columna:2
Coordenadas no válidas, pruebe otra vez

Jugador 1:
Digite fila:2
Digite columna:0
[x][ ][ ]
[x][o][ ]
[x][ ][o]

Felicidades! Ganó el jugador #1
iMac-de-Sonia:ejemplos zonalv$ █
```

Como podemos observar, el programa empieza pidiéndole al jugador #1 la fila y la columna en la cual quiere colocar su carácter correspondiente, para el jugador #1 el carácter asignado es “X” recordando que las filas y las columnas empiezan desde 0 y terminan en 2.

De esta manera, si el jugador #1 logra llenar con su caracter “X” una fila, columna o diagonal, el programa identifica ésto y concluye con que el ganador es el jugador #1.

Esto mismo pasa para el jugador #2, pero en éste caso, el programa en vez de identificar que una fila, columna o diagonal este llena de "X", identifica que este llena de "O", el cual es el carácter asignado para el jugador #2.

El resultado cuando el ganador es el jugador #2 se verá como:

```
Hola jugadores, bienvenidos al juego del gato!
[ ][ ][ ]
[ ][ ][ ]
[ ][ ][ ]

Presione W para jugar / V para salir: W

Jugador 1:
Digite fila:1
Digite columna:1
[ ][ ][ ]
[ ][x][ ]
[ ][ ][ ]

Jugador 2:
Digite fila:2
Digite columna:2
[ ][ ][ ]
[ ][x][ ]
[ ][ ][o]

Jugador 1:
Digite fila:1
Digite columna:0
[ ][ ][ ]
[x][x][ ]
[ ][ ][o]

Jugador 2:
Digite fila:1
Digite columna:2
[ ][ ][ ]
[x][x][o]
[ ][ ][o]

Jugador 1:
Digite fila:0
Digite columna:1
[ ][x][ ]
[x][x][o]
[ ][ ][o]

Jugador 2:
Digite fila:0
Digite columna:2
[ ][x][o]
[x][x][o]
[ ][ ][o]

Felicidades! Ganó el jugador #2
iMac-de-Sonia:ejemplos zonialv$ █
```

También, tenemos el caso de cuando el usuario digita coordenadas en las que ya hay un valor, es decir, cuando la casilla ya está ocupada con una "X" o con una "O", o cuando el usuario digita coordenadas que no existen, es decir una fila mayor a 2 o una columna mayor a 2, en este caso el programa imprime en pantalla "Coordenadas no válidas" y vuelve a preguntarle al usuario las coordenadas, de modo que entra en un ciclo hasta que el usuario ingrese coordenadas correctas.

```
Hola jugadores, bienvenidos al juego del gato!
[ ][ ][ ]
[ ][ ][ ]
[ ][ ][ ]

Presione W para jugar / V para salir: w

Jugador 1:
Digite fila:0
Digite columna:0
[x][ ][ ]
[ ][ ][ ]
[ ][ ][ ]

Jugador 2:
Digite fila:0
Digite columna:1
[x][o][ ]
[ ][ ][ ]
[ ][ ][ ]

Jugador 1:
Digite fila:0
Digite columna:0

Coordenadas no válidas, pruebe otra vez

Jugador 1:
Digite fila:
```


Cuando hay un empate, el programa lo que hace es identificar que si los turnos que han pasado son igual a 10 y que el tablero ya esta lleno, es decir que las coordenadas ya no están disponibles, por tanto concluye con que es un empate y se imprime en pantalla lo siguiente:

```
Jugador 1:
Digite fila:2
Digite columna:2
[o][ ][ ]
[ ][x][ ]
[ ][ ][x]
```

```
Jugador 2:
Digite fila:1
Digite columna:2
[o][ ][ ]
[ ][x][o]
[ ][ ][x]
```

```
Jugador 1:
Digite fila:2
Digite columna:1
[o][ ][ ]
[ ][x][o]
[ ][x][x]
```

```
Jugador 2:
Digite fila:0
Digite columna:1
[o][o][ ]
[ ][x][o]
[ ][x][x]
```

```
Jugador 1:
Digite fila:0
Digite columna:2
[o][o][x]
[ ][x][o]
[ ][x][x]
```

```
Jugador 2:
Digite fila:2
Digite columna:0
[o][o][x]
[ ][x][o]
[o][x][x]
```

```
Jugador 1:
Digite fila:2
Digite columna:0
```

Coordenadas no válidas, pruebe otra vez

```
Jugador 1:
Digite fila:1
Digite columna:0
[o][o][x]
[x][x][o]
[o][x][x]
```

Hay un empate!!

iMac-de-Sonia:ejemplos zonialv\$ █

Para la realización de este proyecto, se utilizaron los siguientes recursos informáticos:

HARDWARE	SOFTWARE
Equipo de computo: macOS Sierra Procesador: 2.3 GHz Intel Core i5 Memoria: 8 GB 2133 MHz DDR4 Disco de arranque: Macintosh HD	Sublime Text Compilador GCC Word Zoom Símbolo del sistema

Tabla de costos asociados al proyecto

El proyecto fue realizado en un periodo aproximado de 19 días. Durante estos días, aproximadamente se le dedicaron tres horas por día al proyecto, ya que era necesaria una investigación previa para corregir errores y crear el programa con el mejor funcionamiento posible. Además, se utilizó energía eléctrica e internet.

De modo que el costo de los honorarios será en base a las horas totales dedicadas para la realización del proyecto, en donde ya se consideran los gastos de energía eléctrica e internet, de esta manera se ha creado la siguiente tabla para representar el costo total del proyecto realizado.

Días totales	Horas por día	Horas totales	Costo por hora	Costo total
19 días (22 de noviembre del 2021 a 10 de diciembre del 2021)	3 horas por día	57 horas	\$200.00 MXN	\$11,400.00 MXN

A continuación, se muestra un diagrama de Gantt que refleja las actividades y sus respectivas fechas en las que fueron realizadas

INICIO DE PROYECTO: 22/11/12

ENTREGA DE PROYECTO: 10/12/2021

ACTIVIDADES	FECHA DE INICIO	DURACIÓN EN DÍAS	FECHA FIN
Establecer el tema de mi proyecto	22/11/2021	1 día	23/11/2021
Buscar información sobre el tema	23/11/2021	1 día	24/11/2021
Comenzar con el algoritmo	25/11/2021	1 día	26/11/2021
Comenzar con el diagrama de flujo	26/11/2021	1 día	27/11/2021
Comenzar con la codificación	28/11/2021	3 días	01/12/2021
Hacer ajustes (revisar si hay errores en el reporte escrito)	01/12/2021	3 días	04/12/2021
Grabar video	05/12/2021	1 día	06/12/2021
Editar video	06/12/2021	2 días	08/12/2021
Entregar el proyecto	09/12/2021	1 día	10/12/2021

Manual de usuario del proyecto

Para poder jugar este juego en la terminal de tu computadora, es necesario que cuentes con un compilador GCC, para que este pueda ser ejecutado de manera correcta. Puede llevarse a cabo en sistema operativo Windows o Linux.

Para jugar deberás ingresar la letra “W” o “w”, y para salir del juego deberás ingresar la letra “V”

Antes de jugar, necesitas saber que las coordenadas del juego se manejan de 0 a 2, es decir, fila 0; fila 1; fila 2, y columna 0; columna 1; columna 2. Además, el juego empieza a preguntar por filas y después por columnas.

Por ejemplo, si quiero que mi “X” se encuentre en la posición (0,0) entonces, empezaré ingresando la fila de posición 0 y posteriormente la columna de posición 0, y así sucesivamente para las demás posiciones en las cuales quiero colocar el carácter asignado.

Es necesario saber que no puedes repetir coordenadas en las que la posición ya tiene una “X” o una “O”, si es el caso, el programa te volverá a pedir las coordenadas hasta que ingreses posiciones en las que todavía no hay valores asignados.

Tampoco podrás seguir con el juego si ingresas posiciones mayores a 2, es decir, una fila mayor a 2 o una columna mayor a 2.

El juego se gana en el momento en que se consigue llenar una fila, columna o diagonal con su carácter asignado.

Si los jugadores han llenado el tablero y ninguno ha llenado una fila, columna o diagonal, entonces será un empate, por tanto no hay ganador y tampoco hay un perdedor.

Conclusiones

Durante este proyecto, se aplicaron diversos temas vistos en clase, empezaremos explicando la parte del algoritmo y del diagrama de flujo, y es que como se mencionó en clase, los algoritmos son dos tipos de herramientas usadas para explicar el proceso de un programa, especialmente son usados por programadores, esto con el fin de que antes de hacer la codificación, la estructura de éste sea la más correcta posible y podamos corregir errores si es que los hay.

Por lo tanto, el algoritmo y el diagrama de flujo en este proyecto, fueron de gran ayuda para poder llevar a cabo la codificación. Gracias a estas herramientas se pudieron corregir errores y el programa entonces se pudo ejecutar de manera correcta. Además sirvió de base para observar qué flujo seguiría el programa.

Para la ejecución del programa, se utilizaron estructuras de control selectivas, las cuales permiten controlar las instrucciones de ejecución del programa.

En este programa, utilizamos la estructura de control selectiva if, en la cual se evalúa la expresión lógica y, si se cumple (si la condición es verdadera), se ejecutan las instrucciones que se encuentran entre las llaves de la estructura. Si no se cumple la condición, se continúa con el flujo normal del programa.

También, utilizamos la estructura de control selectiva if-else, la cual consiste en evaluar la expresión lógica y si la condición es verdadera se ejecutan las instrucciones del bloque que se encuentra entre las primeras llaves, si la condición es falsa se ejecuta el bloque de código que está entre las llaves después de la palabra reservada else. Al final de que se ejecute uno u otro código, se continúa con el flujo normal del programa. Aquí, nos fue útil la estructura if-else para que el usuario decidiera empezar el juego, en caso de que presionara la tecla "W", sino presiona esa tecla, entonces el juego no comenzará.

Las siguientes estructuras que utilizamos, son estructuras de repetición, también llamadas cíclicas, iterativas o de bucles, las cuales permiten ejecutar un conjunto de instrucciones de manera repetida mientras que la expresión lógica a evaluar se cumpla.

En esta ocasión, utilizamos la estructura de control repetitiva do-while, la cual ejecuta el bloque que se encuentra dentro de las llaves y después válida la condición, es decir, el bloque de código se ejecuta de una a un determinado número de veces. Esta estructura nos ayudó a que en el juego, si el usuario colocaba coordenadas repetidas o inexistentes, entonces mandará un aviso al usuario, y volviera a preguntar las coordenadas, de manera que se entra en un ciclo hasta que el usuario digite las coordenadas correctas. Cuando las coordenadas que el usuario digitó fueran las correctas, entonces sale del bucle y continúa su flujo normal. En este caso, se imprime la matriz con el carácter asignado para el jugador en turno.

Por otro lado, también se utilizó la estructura de control de repetición for, la cual permite realizar repeticiones cuando generalmente el control de la repetición está definido sobre una variable contador.

Para la creación de la matriz, que va a contener los caracteres “X” y “O” en nuestro juego, utilizamos un arreglo multidimensional usando for, en donde se accede a cada uno de los elementos por la posición que indica el renglón y la columna, todo esto a través de dos ciclos for, uno anidado dentro de otro.

De esta manera, los temas vistos en clase y la ejemplificación de cada uno de ellos, ayudó a la mejor comprensión para llevar a cabo éste juego.

Conclusiones personales

Mientras buscaba información y algunas ideas para realizar mi proyecto, me encontré con un video que mostraba cómo llevar a cabo el juego del gato en lenguaje C, el cual me pareció bastante interesante y no tan difícil de realizar. Pues, necesitaba saber de algunos conceptos solamente, los cuales ya conocía gracias a los temas que revisamos en clase y las prácticas de laboratorio realizadas.

Gracias a lo visto en clase y a los recursos que tenemos de internet, se pudo llevar a cabo este proyecto con éxito, sin duda no lo habría logrado sin ayuda del canal de YouTube de Luis Fernando, en el cual nos enseña y muestra con claridad los pasos para realizar el juego del gato en lenguaje C. Sin embargo, si yo no hubiera conocido y practicado los conceptos que se mencionan, no habría entendido lo que se realizó. También logré realizar correctamente mi proyecto gracias al profesor Marco Antonio que aclaró mis dudas y nos dio algunos consejos.

Para mi no sólo fue importante seguir este tutorial, sino más bien entender cada uno de los temas que se aplicaron y por qué se aplicaron, además de que me percate de algunas cosas que yo podía cambiar o implementar. Como lo fue a la hora de implementar la parte del empate en el juego, pues ésta no estaba especificada en el video del que me apoye, sin embargo me pareció necesario ponerla porque es parte fundamental del juego, ya que es común que los participantes queden en un empate.

Éste proyecto me hizo ver de diferente manera los juegos pues, como en un principio se mencionó, los juegos no son sólo para distraernos y pasar un buen rato, sino también para desarrollar nuestra creatividad y por tanto, nos ayuda a ver distintas formas de solucionar un problema. También, es importante mencionar que ayudan al desarrollo cognitivo de los niños, y con ésto se volverán más ágiles intelectualmente y muy eficaces a la hora de que se encuentren con un contratiempo.

Me pareció bastante interesante la manera en cómo podemos llevar a cabo cosas de la vida real, como en este caso el juego del gato, a una forma automatizada, esto me deja ver que cualquier proceso, por más sencillo o complejo que sea, puede ser llevado a la terminal de una computadora. Por eso, es importante que constantemente estemos actualizandonos sobre la tecnología, pues usar estas herramientas a nuestro favor nos puede ser de mucha utilidad durante nuestra trayectoria escolar y profesional. Con esto quiero decir que, saber automatizar procesos será un gran aliado para nuestro futuro como ingenieros, pues buscamos que las cosas se hagan de manera eficaz y eficiente, y que estos procesos automatizados reduzcan los errores humanos, para así poder tener más beneficios y menos problemas a solucionar.

No cabe duda de que este primer proyecto, aunque parezca muy pequeño, me será de gran utilidad para las próximas cosas que yo quiera implementar y programar, además de que formara parte de mi experiencia, lo cual me ayudará a contar con referencias sobre programación en lenguaje C y además a aplicarlo en la vida cotidiana.

En el siguiente link podrás encontrar la explicación de cómo se lleva a cabo el proyecto realizado

<https://youtu.be/ekaQ-VpLcLY>

En el siguiente link se encuentra el repositorio de GitHub que contiene los archivos del proyecto

https://github.com/carolinavargas7/Proyecto_Final-.git

Referencias:

Luján, I. (2016). *Tres en raya: su historia medieval*. Recuperado de: Universitat de Valencia.

[https://www.uv.es/uvweb/master-historia-formacion-mundo-occidental/es/blog/tres-
raya-historia-medieval-1285960141137/GasetaRecerca.html?id=1285979081034](https://www.uv.es/uvweb/master-historia-formacion-mundo-occidental/es/blog/tres-
raya-historia-medieval-1285960141137/GasetaRecerca.html?id=1285979081034)

Delgado, G. (2020). *Juego del gato*. Recuperado de: Panorama
<https://www.panoramadigital.co.cr/juego-del-gato/>

Canal Luis Fernando. (12 de diciembre de 2018). *Juego del gato en C en 30 min #1 Matrices y ciclos*. Recuperado el 30 de noviembre del 2021. (Archivo de Video). Youtube.
<https://www.youtube.com/watch?v=WgDQkID8Mjs&list=LL&index=3>

Canal Luis Fernando. (12 de diciembre de 2018). *Juego del gato en C en 30 min Matrices #2 y ciclos*. Recuperado el 30 de noviembre del 2021. (Archivo de Video). Youtube. <https://www.youtube.com/watch?v=xakKXcburEU>

Canal Luis Fernando. (12 de diciembre de 2018). *Juego del gato en C en 30 min #3 Matrices y ciclos*. Recuperado el 30 de noviembre del 2021. (Archivo de Video). Youtube. <https://www.youtube.com/watch?v=nmzY7kpaxSk>

García Cano, E. y Solano Gálvez, J. (26 de agosto del 2021). *Guía práctica de estudio 08: Estructuras de repetición*. Recuperado de: Laboratorio de Computación Salas A y B. <http://lcp02.fi-b.unam.mx>

García Cano, E. y Solano Gálvez, J. (26 de agosto del 2021). *Guía práctica de estudio 10: Arreglos multidimensionales*. Recuperado de: Laboratorio de computación Salas A y B. <http://lcp02.fi-b.unam.mx>

García Cano, E. y Solano Gálvez, J. (26 de agosto del 2021). *Guía práctica de estudio 07: Estructuras de selección*. Recuperado de: Laboratorio de computación Sala A y B. <http://lcp02.fi-b.unam.mx>