

# PROGRAMACIÓN APLICADA Y LAB.

## PROYECTO FINAL

### < JUEGO DE GATO >

Juan Castillo Carrillo, Ana Regina Morfín, y David Emmanuel Pérez Ávila

*Abstract*—El proyecto consiste en un Juego de Gato con interfaz gráfica GTK. Puede jugarse de dos personas o jugador contra el programa. Tiene la finalidad de divertir al jugador y retarlo, así como interactuar con una buena interfaz gráfica. El juego no sólo debe ser funcional sino también agradable estéticamente. Se pretende que los alumnos que desarrollaron el programa, demuestren los aprendizajes adquiridos a lo largo del semestre con el uso de estructuras, diferenciación del modelo (vista/controlador), uso de funciones con argumentos, uso de archivos de texto binarios, y uso de menús.

#### I. Introducción

El tema del proyecto consiste en un Juego de Gato funcional y estéticamente agradable y amigable en GTK. El programa desarrollado puede ser útil para jugadores que buscan divertirse y retarse a sí mismos, pues al jugar contra el mismo juego no puede ganar el jugador, pero puede aspirar a un empate.

Se utilizan estructuras para manejar la información de cada jugada, y archivos para cargar y guardar juegos previos.

Para el programa se desarrollo primero la “inteligencia” o el algoritmo del juego y una vez que se tenían las partes fundamentales se le incorporó el modelo de vista con GTK.

La forma de ejecutar el juego es correrlo desde la terminal y se abre una ventana en la cual, a partir de ahí, la interacción del usuario con el juego es de manera gráfica.

#### II. Análisis

Para empezar, se identificó el objetivo del proyecto y se dividió en partes todo lo que se debía de elaborar. Se realizaron diagramas para facilitar el análisis y la organización y se elaboró el pseudocódigo posteriormente.

Al ejecutar el programa se abre la ventana principal y el usuario debe de irse al menú y seleccionar el tipo de juego que desea, así como su nombre. Una vez guardado, el jugador 1 (X) lleva prioridad del primer tiro y de ahí en adelante los turnos se alternan.

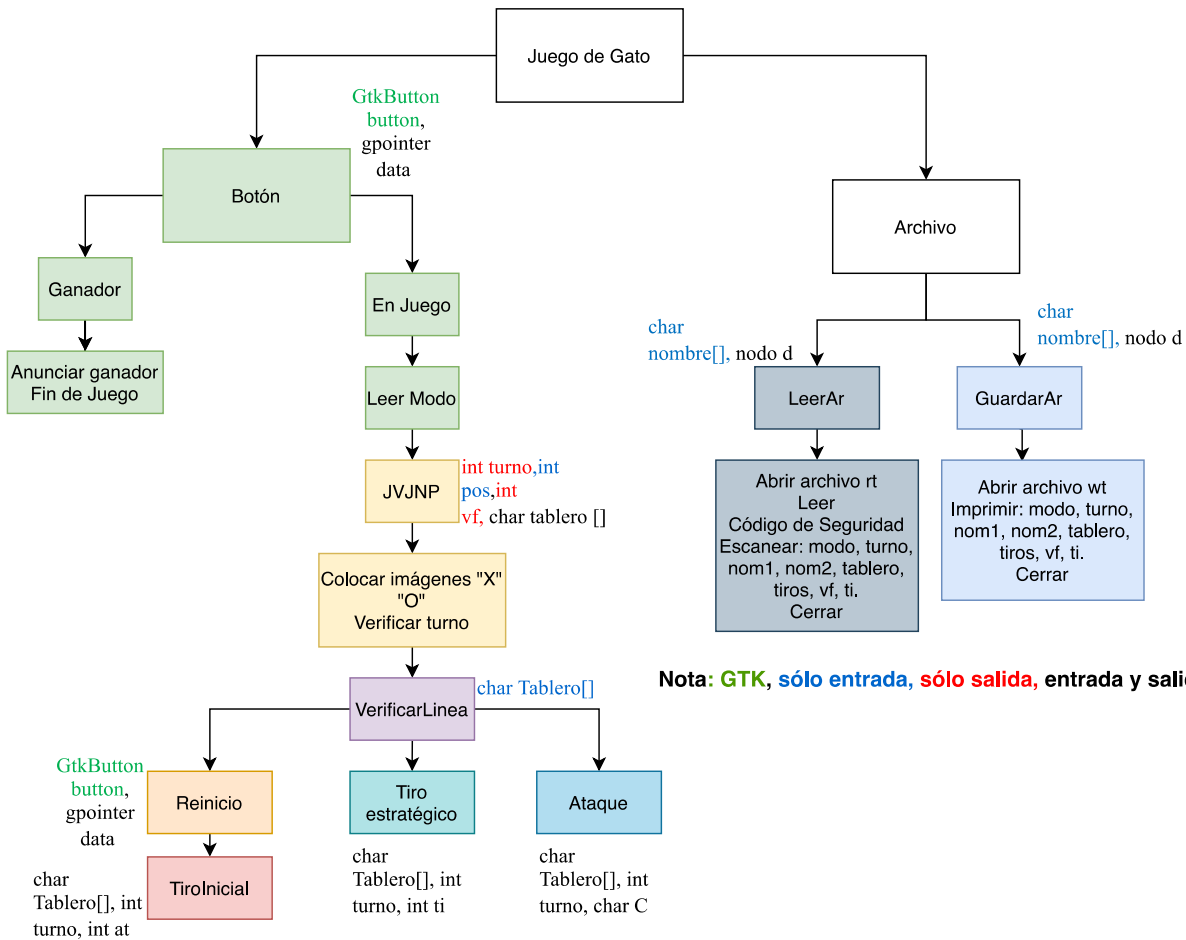
El menú tiene diferentes opciones como Ayuda, donde se encuentra el manual de usuario; Acerca De, con información del juego; Archivo, para cargar o guardar archivos con las jugadas; y Juego para seleccionar la opción de juego y los nombres.

Las funciones son llamadas y ejecutadas de acuerdo a sus controladores en GTK. Una vez finalizado el juego, éste se puede guardar, se puede empezar uno nuevo, consultar información del programa, o salir.

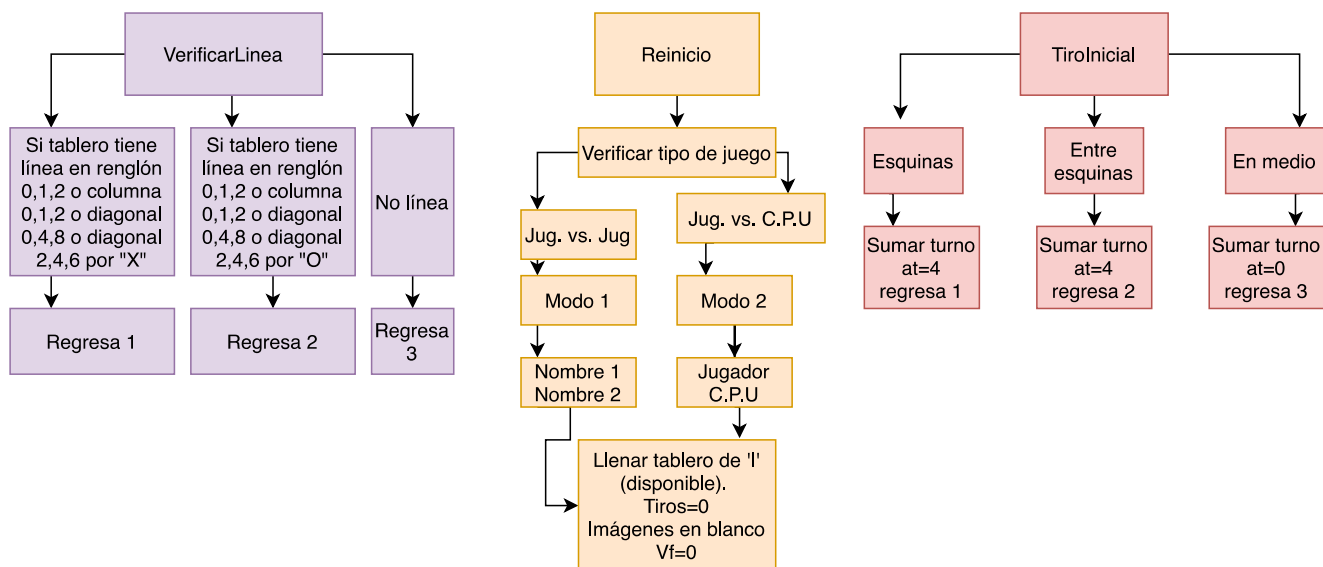
Se elaboraron diagramas y pseudocódigo, así como diseño gráfico de las pantallas.

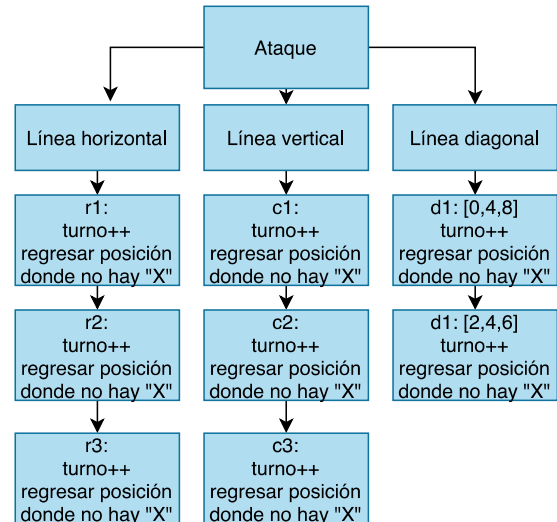
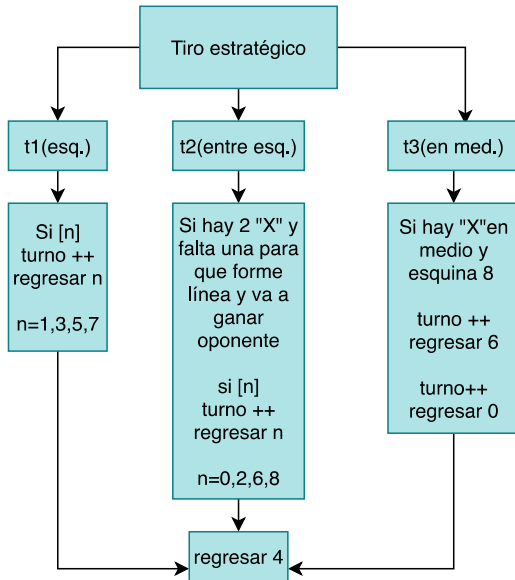
### III. Diseño de Solución

#### A. Diagrama General



#### B. Diagramas de otras funciones





### C. Pseudocódigo

Funciones del Gato:

```

int VerificaLínea (char Tablero[]); 4
void JVJNP (int turno, int pos, int vf, char Tablero []); 2
int TiroInicial (char Tablero[], int turno, int at); 5
int TiroEstratégico (char Tablero[], int turno, int ti); 5
int Ataque (char Tablero[], int turno, char C); 5
void LeeAr(char nombre[], nodo d); a
void GuardaAr(char nombre[], nodo d); b
  
```

GTK con Gato:

```

void Botón(GtkButton button, gpointer data); 1
void Reinicio(GtkMenuItem p, gpointer data); 3
  
```

```

int main(int argc, char argv[])
  
```

```

{
    ↑lista.modo=0;
    ↑lista.turno=1;
    for(i=0;i<9;i++)
    {
        ↑lista.Tablero[i]='I';
        ↑ lista.tiros[i]=0;
    }
    ↑ lista.vf=0;
    ↑lista.ti=-1;
    ↑lista.Tablero[9]='\0';
  
```

```

for(i=0;i<9;i++)
{
    a=i;
    x=(a/100);
    regresa 0;
  
```

```

}
  
```

```

void Botón(GtkButton button, gpointer data)
  
```

```

{
    SI(dato.turno ==10 OR dato.vf==2 OR dato.vf==1 OR
    dato.modo==0)
        Imprime("Ya gano alguien");
    SI NO
        {
            SI(dato.Tablero[j]=='I')
                {
                    JVJNP(&↑dato.turno, j, &↑dato.vf,
                    ↑dato.Tablero);
                    ↑dato.turno=↑dato.turno-1;
                    SI(↑dato.turno==1 OR ↑dato.turno==3 OR
                    ↑dato.turno==5 OR ↑dato.turno==7 OR ↑dato.turno==9)
                        {
                            Colocar imagen "X";
                        }
                    SI(↑dato.turno==2 OR ↑dato.turno==4 OR
                    ↑dato.turno==6 OR ↑dato.turno==8)
                        {
                            Colocar imagen "O";
                        }
                    ↑dato.turno=↑dato.turno +1;
                    SI(↑dato.modo==2 AND ↑dato.turno <> 10 AND
                    ↑dato.vf <> 1)
                        {
                            SI(↑dato.turno==2)
                                {
                                    ↑dato.ti = TiroInicial(↑dato.Tablero,
                                    &↑dato.turno, &at);
                                    ↑dato.turno--;
                                }
                            }
                        }
                }
        }
    }
  
```

```

        JVJNP(&↑dato.turno, at, &↑dato.vf,
        ↑dato.Tablero);
        Colocar imagen "O";
    }
    SI NO
    {
        at=Ataque(↑dato.Tablero, &↑dato.turno,
        'O');
        SI (at==1)
            at=Ataque(↑dato.Tablero, &↑dato.turno,
            'X');
        SI(at==1)
            at=TiroEstratégico(↑dato.Tablero,
            &↑dato.turno, ↑dato.ti);
        ↑dato.turno= ↑dato.turno-1;
        JVJNP(&↑dato.turno, at, &↑dato.vf,
        ↑dato.Tablero);
        Colocar imagen "O";
    }
}
SI(↑dato.vf==1)
    Imprime("Ganó la X");
SI NO SI(↑dato.vf==2)
    Imprime("Ganó la O");
SI NO SI(↑dato.turno==10)
    Imprime("Nadie ganó");
}
}

```

```

void JVJNP (int turn,int pos,int vf, char Tablero[]) //Jugador vs.
Jugador Nueva Partida
{
    turno=turn;
    SI(turno==1 OR turno==3 OR turno==5 OR turno==7 OR
turno==9)
        Tablero[pos]='X';
    SI(turno==2 OR turno ==4 OR turno==6 OR turno==8)
        Tablero[pos]='O';
    turn=turn+1;
    VerificaLínea (Tablero);
}

```

```

int VerificaLínea (char Tablero[])
{
    SI(Tablero[0]=='X' AND Tablero[1]=='X' AND
Tablero[2]=='X')
        regresa 1;
    SI(Tablero[3]=='X' AND Tablero[4]=='X' AND
Tablero[5]=='X')
        regresa 1;

```

```

    SI(Tablero[6]=='X' AND Tablero[7]=='X' AND
Tablero[8]=='X')
        regresa 1;
    SI(Tablero[0]=='X' AND Tablero[3]=='X' AND
Tablero[6]=='X')
        regresa 1;
    SI(Tablero[1]=='X' AND Tablero[4]=='X' AND
Tablero[7]=='X')
        regresa 1;
    SI(Tablero[2]=='X' AND Tablero[5]=='X' AND
Tablero[8]=='X')
        regresa 1;
    SI(Tablero[0]=='X' AND Tablero[4]=='X' AND
Tablero[8]=='X')
        regresa 1;
    SI(Tablero[2]=='X' AND Tablero[4]=='X' AND
Tablero[6]=='X')
        regresa 1;
    SI(Tablero[0]=='O' AND Tablero[1]=='O' AND
Tablero[2]=='O')
        regresa 2;
    SI(Tablero[3]=='O' AND Tablero[4]=='O' AND
Tablero[5]=='O')
        regresa 2;
    SI(Tablero[6]=='O' AND Tablero[7]=='O' AND
Tablero[8]=='O')
        regresa 2;
    SI(Tablero[0]=='O' AND Tablero[3]=='O' AND
Tablero[6]=='O')
        regresa 2;
    SI(Tablero[1]=='O' AND Tablero[4]=='O' AND
Tablero[7]=='O')
        regresa 2;
    SI(Tablero[2]=='O' AND Tablero[5]=='O' AND
Tablero[8]=='O')
        regresa 2;
    SI(Tablero[0]=='O' AND Tablero[4]=='O' AND
Tablero[8]=='O')
        regresa 2;
    SI(Tablero[2]=='O' AND Tablero[4]=='O' AND
Tablero[6]=='O')
        regresa 2;
    SI NO
        regresa 3;
}

```

```

int TiroInicial(char Tablero[], int turno, int at)
{
    SI(Tablero[0]=='X' OR Tablero[2]=='X' OR Tablero[6]=='X'
OR Tablero[8]=='X')
    {
        at=4;

```

Primavera 2020

```

    turno=turno+1;
    regresa 1;
}
SI(Tablero[1]=='X' OR Tablero[3]=='X' OR Tablero[5]=='X'
OR Tablero[7]=='X')
{
    at=4;
    turno=turno+1;
    regresa 2;
}
SI(Tablero[4]=='X')
{
    at=0;
    turno=turno+1;
    regresa 3;
}
regresa 0;
}

```

```

int TiroEstratégico (char Tablero[], int turno, int ti)
{
    SI(ti==1)
    {
        SI(Tablero[1]=='I') // I es disponible
        {
            turno=turno+1;
            regresa 1;
        }
        SI(Tablero[3]=='I')
        {
            turno=turno+1;
            regresa 3;
        }
        SI(Tablero[5]=='I')
        {
            turno=turno+1;
            regresa 5;
        }
        SI(Tablero[7]=='I')
        {
            turno=turno+1;
            regresa 7;
        }
    }
    SI(ti==2)
    {
        SI(turno==4 AND Tablero[5]=='X' AND Tablero[7]=='X')
        {
            turno=turno+1;
            regresa 2;
        }
        SI(Tablero[0]=='I')
    }
}

```

```

{
    turno=turno+1;
    regresa 0;
}
SI(Tablero[2]=='I')
{
    turno=turno+1;
    regresa 2;
}
SI(Tablero[6]=='I')
{
    turno=turno+1;
    regresa 6;
}
SI(Tablero[8]=='I')
{
    turno=turno+1;
    regresa 8;
}
}
SI(ti==3)
{
    SI(turno==4 AND Tablero[8]=='X')
    {
        turno=turno+1;
        regresa 6;
    }
    turno=turno+1;
    regresa 0;
}
regresa 4;
}

int Ataque (char Tablero[], int turno, char C)
{
    //Primera Línea Horizontal
    SI(Tablero[0]==C AND Tablero[1]==C)
    SI(Tablero[2]=='I')
    {
        turno=turno+1;
        regresa 2;
    }
    SI(Tablero[0]==C AND Tablero[2]==C)
    SI(Tablero[1]=='I')
    {
        turno=turno+1;
        regresa 1;
    }
    SI(Tablero[1]==C AND Tablero[2]==C)
    SI(Tablero[0]=='I')
    {
        turno=turno+1;
    }
}

```

Primavera 2020

```

    regresa 0;
}

//Segunda Línea Horizontal
SI(Tablero[3]==C AND Tablero[4]==C)
SI(Tablero[5]==T)
{
    turno=turno+1;
    regresa 5;
}
SI(Tablero[3]==C AND Tablero[5]==C)
SI(Tablero[4]==T)
{
    turno=turno+1;
    regresa 4;
}
SI(Tablero[4]==C AND Tablero[5]==C)
SI(Tablero[3]==T)
{
    turno=turno+1;
    regresa 3;
}

//Tercera Línea Horizontal
SI(Tablero[6]==C AND Tablero[7]==C)
SI(Tablero[8]==T)
{
    turno=turno+1;
    regresa 8;
}
SI(Tablero[6]==C AND Tablero[8]==C)
SI(Tablero[7]==T)
{
    turno=turno+1;
    regresa 7;
}
SI(Tablero[7]==C AND Tablero[8]==C)
SI(Tablero[6]==T)
{
    turno=turno+1;
    regresa 6;
}

//Primera Línea Vertical
SI(Tablero[0]==C AND Tablero[3]==C)
SI(Tablero[6]==T)
{
    turno=turno+1;
    regresa 6;
}
SI(Tablero[0]==C AND Tablero[6]==C)
SI(Tablero[3]==T)
{

```

```

    turno=turno+1;
    regresa 3;
}
SI(Tablero[3]==C AND Tablero[6]==C)
SI(Tablero[0]==T)
{
    turno=turno+1;
    regresa 0;
}
//Segunda Línea Vertical
SI(Tablero[1]==C AND Tablero[4]==C)
SI(Tablero[7]==T)
{
    turno=turno+1;
    regresa 7;
}
SI(Tablero[1]==C AND Tablero[7]==C)
SI(Tablero[4]==T)
{
    turno=turno+1;
    regresa 4;
}
SI(Tablero[4]==C AND Tablero[7]==C)
SI(Tablero[1]==T)
{
    turno=turno+1;
    regresa 1;
}
//Tercera Línea Vertical
SI(Tablero[2]==C AND Tablero[5]==C)
SI(Tablero[8]==T)
{
    turno=turno+1;
    regresa 8;
}
SI(Tablero[2]==C AND Tablero[8]==C)
SI(Tablero[5]==T)
{
    turno=turno+1;
    regresa 5;
}
SI(Tablero[5]==C AND Tablero[8]==C)
SI(Tablero[2]==T)
{
    turno=turno+1;
    regresa 2;
}
//Primera Línea Diagonal
SI(Tablero[0]==C AND Tablero[4]==C)
SI(Tablero[8]==T)
{
    turno=turno+1;

```

Primavera 2020

```

    regresa 8;
}
SI(Tablero[0]==C AND Tablero[8]==C)
SI(Tablero[4]=='T')
{
    turno=turno+1;
    regresa 4;
}
SI(Tablero[4]==C AND Tablero[8]==C)
SI(Tablero[0]=='T')
{
    turno=turno+1;
    regresa 0;
}
//Segunda Línea Diagonal
SI(Tablero[2]==C AND Tablero[4]==C)
SI(Tablero[6]=='T')
{
    turno=turno+1;
    regresa 6;
}
SI(Tablero[2]==C AND Tablero[6]==C)
SI(Tablero[4]=='T')
{
    turno=turno+1;
    regresa 4;
}
SI(Tablero[4]==C AND Tablero[6]==C)
SI(Tablero[2]=='T')
{
    turno=turno+1;
    regresa 2;
}
regresa -1;
}

void Reinicio(GtkMenuItem p, gpointer data)
{
    SI(COMPARE(opc,"J VS. J")==0)
    {
        ↑lista.modo=1;
        strcpy(↑lista.nombre1,"Nom1");
        strcpy(↑lista.nombre2,"Nom2");
    }
    SI(COMPARE(opc,"J VS. CPU")==0)
    {
        ↑lista.modo=2;
        strcpy(↑lista.nombre1,"Jugador");
        strcpy(↑lista.nombre2,"CPU");
    }
    ↑lista.turno=1;

```

```

for(i=0;i<9;i++)
{
    ↑ lista.Tablero[i]='T';
    ↑lista.tiros[i]=0;
    Colocar imagen en blanco;
}
↑ lista.vf=0;
}

void LeeAr(char nombre[], nodo d)
{
    archivo=ABRIR(nombre, "rt");
    LEER(archivo,"%s\n",Línea);
    SI(COMPARE(Línea, "WA21JM10")==0) //código de
seguridad
    {
        LEER(archivo,"%d\n",&↑d.modo);
        LEER(archivo,"%d\n",&↑d.turno);
        LEER(archivo,"%s\n",↑d.nombre1);
        LEER(archivo,"%s\n",↑d.nombre2);
        LEER(archivo,"%s\n",↑d.Tablero);
        for(i=0;i<9;i++)
            LEER(archivo,"%d\n",&↑d.tiros[i]);
        LEER(archivo,"%d\n",&↑d.vf);
        LEER(archivo,"%d\n",&↑d.ti);
    }
    SI NO
        Imprime("Archivo no válido");
    CERRAR(archivo);
}

void GuardaAr(char nombre[], nodo d)
{
    archivo=ABRIR(nombre, "wt");
    ImprimeArchivo(archivo,"WA21JM10\n");
    ImprimeArchivo (archivo,"%d\n",↑d.modo);
    ImprimeArchivo (archivo,"%d\n",↑d.turno);
    ImprimeArchivo (archivo,"%s\n",↑d.nombre1);
    ImprimeArchivo (archivo,"%s\n",↑d.nombre2);
    ImprimeArchivo (archivo,"%s\n",↑d.Tablero);
    for(i=0;i<9;i++)
        ImprimeArchivo (archivo,"%d\n",↑d.tiros[i]);
    ImprimeArchivo (archivo,"%d\n",↑d.vf);
    ImprimeArchivo (archivo,"%d\n",↑d.ti);
    CERRAR(archivo);
}

```

IV. Diseño de Pantallas

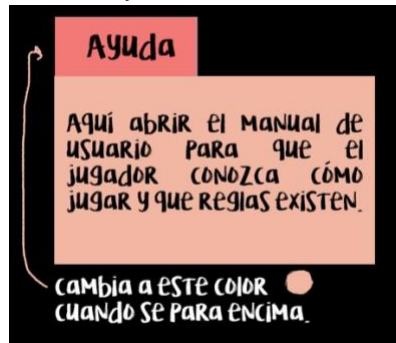
A. Pantalla principal



B. Archivo



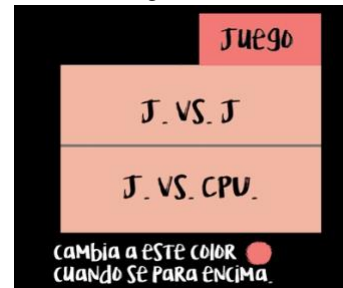
C. Ayuda



D. Acerca de



E. Juego



F. Jugadores

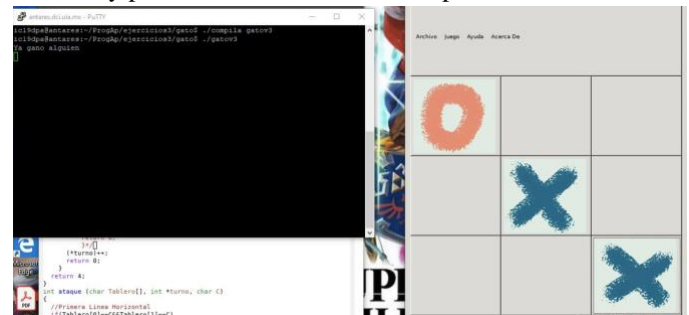


V. Plan de Pruebas

Se revisó cada pseudocódigo previamente de pasarlo a código y se buscaron errores, así como comprobar que las funciones sean llamadas y funcionen debidamente.

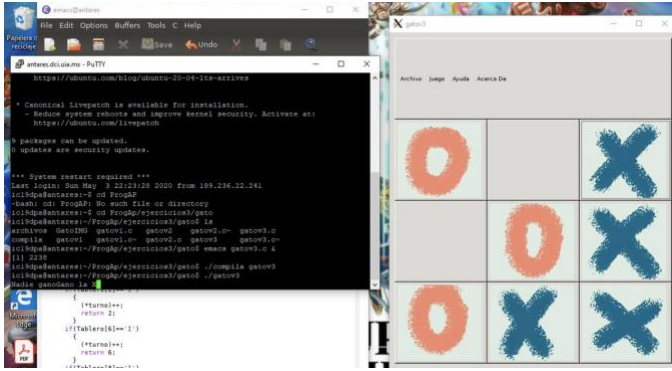
Una vez que se pasó el pseudocódigo a código se buscaron errores de escritura y se corrigieron para que aparecieran menos errores al momento de compilar y evitar complicaciones por nombres en especial.

Al probar el juego se realizaron diferentes jugadas y se encontró que existían dos que no funcionaban de acuerdo a lo esperado. Las casillas están enumeradas a partir de 0. La primera jugada consistía en que cuando el jugador tiraba en la casilla 4 y posteriormente en la 8, no se permitían más tiros.



Cuando se tiraba en las casillas 5, 7 y después 8, el jugador ganaba.





Se solucionó agregando condicionales para esas jugadas ya que no entraban en el algoritmo de resolución.

Al cargar los símbolos de archivos se registraron las jugadas, pero no los símbolos. Al leer el *string* del Tablero se debió de verificar si había una “i” y colocar la imagen que comienza en “b”, si había una “o” colocar la imagen con inicio “c”; y si se encontraba una “x”, entonces se debía de poner la imagen “t”.

Al asignar el turno hubo un problema con los parámetros, se revisaron las asignaciones y se corrigieron. El algoritmo del juego fue desarrollado previamente por lo que al vincularlo con GTK no se asignaban bien los turnos que correspondían a qué jugador. Se tuvo que volver a asignar al juego cuando quedarse en el turno y cuando pasar al del oponente.

## VI. Manual de Usuario

El programa se debe de ejecutar desde la terminal y se abre la ventana principal del gato.

El menú superior muestra opciones para el juego. Si desea cargar un archivo debe de posicionarse sobre *Archivo* en el menú y seleccionar la opción de *Cargar*. Se abrirá una ventana nueva en la cual debe seleccionar el archivo deseado y presionar *Open*. Si desea regresar a la ventana principal sin abrir ningún archivo, presione *Cancel*.

Para guardar un archivo se debe de dirigir a la pestaña de *Archivo* en la barra de menú superior y seleccionar *Guardar*. A continuación, se abrirá una ventana nueva en la cual debe de introducir el nombre del archivo y dónde desea que se guarde y presionar *Save*. Si desea regresar a la ventana principal sin guardar presione *Cancel*.

En cuanto al Juego, existen dos opciones: jugador contra jugador, o jugador contra el programa. Si desea jugar contra otro jugador, colóquese sobre *Juego* en el menú y presione *J. vs. J.* Una vez seleccionado, escriba los nombres de los jugadores en la ventana emergente y presione aceptar. Coloque su tiro sobre el tablero y comienza el juego. Los turnos entre los jugadores se señalan y se van alternando. Cuando hay un ganador, ya no se puede tirar y se muestra en la terminal

quién ganó. Para iniciar una nueva partida, colóquese sobre *Juego* y seleccione una opción.

Para jugar contra el programa colóquese sobre *Juego* y seleccione *J. vs. C.P.U.* Después, escriba su nombre en la ventana emergente y presione *Aceptar*. Presione la casilla sobre el tablero en la que. Desea colocar su “X” y el programa responderá como su oponente “O”. Al finalizar se muestra el ganador en la terminal y ya no se permite tirar más.

En el menú se encuentra la opción de *Acerca De* en la cuál encontrará información acerca del juego y los desarrolladores.

Para encontrar el Manual de Usuarios, debe presionar el botón de *Ayuda*.

Cierre el Juego de Gato presionando “X” en la esquina superior derecha de la ventana.

## VII. Conclusión

La importancia del proyecto es que reta al jugador y propicia el desarrollo de habilidades intelectuales y estratégicas. El Juego de Gato entretiene al jugador en un ambiente estético y puede jugar cuantas veces desee. Si se selecciona la opción de jugar contra el programa, éste es mucho más rápido al momento de responder que un oponente humano. Presenta un reto para el jugador pues el juego esta programado para ganarle o empatar con el jugador.

Hablando desde una perspectiva ambiental, al jugar en la computadora y no en papel, se presenta un ahorro del mismo, evitando muchas hojas de desperdicio.

Del mismo modo, el juego no tiene un costo y se puede practicar con él infinitas veces sin que se quiebre o canse.

Al desarrollar el Juego de Gato se utilizaron diferentes conocimientos adquiridos a lo largo del curso y además se agregó el factor de la creatividad en la resolución de problemas, planeación de la solución, y la parte gráfica del juego en GTK.

Es importante que el juego sea agradable y amigable para que cualquiera lo pueda utilizar y tenga una excelente experiencia al jugar. El juego puede ser de mucha utilidad en estos momentos de cuarentena, ya que es una fuente de entretenimiento que requiere del pensamiento.

Se pueden realizar programas similares que se pueden aplicar en el entretenimiento de las personas, así como hacer aplicaciones de utilidad como de ordenamiento de la información y facilitar cálculos, como lo realizado en clase. Una ventaja de GTK es que permite que el programa sea accesible a personas que no saben programar y poder interactuar de manera fácil con el. Además, se puede realizar un programa con especificaciones tanto funcionales como gráficas para adaptarse a la necesidad.

VIII. Abreviaciones y Acrónimos

JVJNP: Jugador vs. Jugador Nueva Partida,

at: posición,

vf: verifica línea,

ti: tiro inicial,

“X”: imagen tache,

“O”: imagen círculo,

“I”: imagen en blanco.

IX. Referencias

[1] A. Krausse. *Foundations of GTK+ Development*. E.E.U.U. Apress, 2007. pp. 315-325

[2] B. Gottfried. *Programación C*. 2a Edición. España. McGrawHill, 2005. pp. 272-298

[3] S. Prata. *C Primer Plus*. 6ª Edición. E.E.U.U. Pearson, 2013. pp. 644-665