

MANUAL TECNICO JUEGO DEL GATO

DAVID SANTIAGO MOGOLLON RODRIGUEZ - 20181020127
LEEYNYKER ANDRÉS MONTAÑO RUÍZ - 20181020140

Universidad Distrital Francisco José De Caldas
Ingeniería De Sistemas
Bogotá D.C
2018

INTRODUCCIÓN

En el siguiente trabajo se sustentarán las bases sobre las que se realizó el proyecto final de la asignatura Programación Básica y que consistía en desarrollar un programa que permita el juego entre un humano y el computador, y que cuenta con una doble funcionalidad, por una parte es un jugador dentro del juego, determinando los lugares donde puede jugar, buscando una posición adecuada para ganar, esto determinando si hay alguna casilla que al ser marcada gane el juego, y si no hay una casilla que cumpla con las condiciones anteriores que juegue en una posición aleatoria. Y por otra, administra el mismo juego realizando tareas como lo son: Simular el lanzamiento del dado tanto para la máquina como para el jugador, desplegar el tablero con las jugadas cada vez que alguno de los jugadores realice su movimiento, indicar el ganador o el correspondiente empate y dar la posibilidad de seguir jugando.

Para sustentar el trabajo realizado con las anteriores directrices a continuación se encontrarán los componentes para la realización de dicho trabajo y que son comprendidos por los objetivos que se tuvieron en cuenta a la hora de realizarlo, el desarrollo del problema, en donde se analiza el problema, el diseño realizado en diagrama de flujo y en pseudocódigo, el código fuente del programa mismo en C++, el manual de usuario indicando el funcionamiento del programa de forma didáctica y las conclusiones respecto al trabajo realizado.

OBJETIVOS

OBJETIVO GENERAL:

Desarrollar un programa en el lenguaje c++ en el cual se permita realizar el juego del gato entre máquina y usuario, por medio del correspondiente análisis y desarrollo del problema, para utilizar como herramientas y sustentar los conocimientos adquiridos a lo largo de la asignatura programación básica durante el periodo 2018-1.

OBJETIVOS ESPECÍFICOS:

- Analizar el desarrollo del pensamiento de la máquina (IA) para que ésta pueda realizar las jugadas con lógica en el programa.
- Utilizar todas las herramientas necesarias para realizar el programa y que éste funcione correctamente.
- Realizar los soportes que demuestren el trabajo realizado con el que se resolvió el problema, tales como el análisis del problema, el diseño realizado en diagrama de flujo y en pseudocódigo, el código fuente del programa en C++, el manual de usuario indicando el funcionamiento del programa de forma didáctica y las conclusiones respecto al trabajo realizado.

MARCO TEORICO

El juego del gato es un juego de suma cero, lo que quiere decir que si ambos jugadores (en este caso H vs C) juegan muy bien, ninguno ganará. Sin embargo, si el usuario pule sus habilidades y domina algunas estrategias

simples puede llegar a ganar.

El tablero está conformado por una matriz de 3x3, lo que significa que se compone de una red de nueve cuadrados. Esto quiere decir que tiene tres filas de tres cuadrados.

Para que dé inicio el jugador deberá elegir la posición en la que realizara su primera jugada. Se hará uso de símbolos diferentes para cada jugador, símbolos se colocarán en el tablero para tratar de tener tres de estos de manera consecutiva. Si empiezas el juego, entonces la mejor maniobra que puedes hacer es movilizarte en el centro. Esto maximizará tus oportunidades de ganar, ya que podrás crear una fila de tres en más combinaciones (cuatro) de esta forma en comparación con lo que podrías haber logrado si hubieras escogido un cuadrado diferente.

Luego de dar inicio el segundo jugador debe colocar su símbolo (que será diferente al del primer jugador). Puede tratar de bloquear al primer jugador y evitar que el primero logre una fila de tres símbolos o puede enfocarse en crear la suya. Lo ideal es que el jugador pueda hacer ambas opciones.

El primer jugador en dibujar tres de sus símbolos de manera consecutiva habrá ganado tres en raya, ya sea que los haya hecho vertical, horizontal o diagonalmente. Sin embargo, si ambos jugadores juegan con una estrategia óptima, entonces hay una gran probabilidad de que nadie gane porque han bloqueado mutuamente sus oportunidades de crear una fila de tres símbolos.

DESARROLLO DEL PROBLEMA

1. Análisis

Para dar solución del problema, el programa inicialmente debe declarar un vector global de tamaño 9, a continuación, se declaran diferentes funciones:

1. Función de tipo carácter llamada “*carácter*” para mostrar en la pantalla H y/o C.
2. Función de tipo void llamada “*mostrarC*” que muestra el cuadro de juego.
3. Función de tipo entero llamada “*ganador*” el cual determina si hay un ganador.
4. Función de tipo entero llamada *ia_serd*” el cual analiza los movimientos.
5. Función de tipo void llamada “*jmaquina*” para el juego que realizará la máquina.
6. Función de tipo void llamada “*jusuario*” para el juego que realizará el usuario.
7. Función de tipo void llamada “*dados*” que realiza el lanzamiento de los dados el cuál determinará quién inicia el juego.
8. Función de tipo void llamada “*inicializar*” para inicializar el vector con las posiciones del juego.

Continuamente se crean variables de tipo bool las cuales retornan un valor de verdadero o falso, una es *bTurno* con valor “true”, otra *bContinue* sin valor y la última *bGanador* con un valor de “false. Se crean luego dos variables de tipo entero, *nTurno*=0 y *nRepeat*=0. Finalmente se crea una variable de tipo carácter llamada *cEnter*.

Ahora, el programa deberá mostrar un mensaje “BIENVENIDO AL JUEGO DEL GATO”, a continuación, se realiza un ciclo en el cual, mientras *nRepeat* sea ==0, *nTurno* será igual 0, *bGanador* obtendrá un valor “false” y entra la función #8 “inicializar”. La función de tipo void #8 llamada “inicializar”, inicializa el vector, dentro de esta función se declara una variable de tipo entero *nPos* que tendrá un valor igual a 0, se crea un ciclo en donde para una nueva variable *i* de tipo entero con un valor inicial igual a 0, hasta que esa variable *i* sea menor que 9, se incrementará *i* en 1, y dentro el valor del vector *vCuadro*

en la posición *i* será igual al valor de la variable *nPos*, y se cierra el ciclo al igual que la función. Luego se muestra el mensaje " Para lanzar dados oprima Enter\n Para cancelar y salir oprima cualquier tecla".

Continuamente el usuario tendrá que oprimir Enter para la realización de lanzamiento de dados, se usa *getch* para guardar en *cEnter* lo ingresado por el usuario, luego se realiza en análisis para ver si la tecla oprimida es la indicada. Si la variable *cEnter* es diferente de 13, el valor de *bContinue* será "falso" de lo contrario, si éste es igual a 13 el valor de *bContinue* será "true". Si el valor de *bContinue* es verdadero entra la función "dados" para realizar el lanzamiento de dados e inicia un ciclo hasta que se defina un ganador inicial o se genere un empate.

En la función de tipo void #7 llamada "dados", se declaran dos variables de tipo entero, *nDama* que es igual a 0 y *nDaus* que es igual a 0. Se realiza un ciclo en donde mientras la variable *nDama* sea igual a la variable *nDaus* se realizará el lanzamiento de dados hasta que ambos valores sean diferentes, mientras ambos valores sean iguales se hará el lanzamiento del dado de la máquina por una función aleatoria, entonces se muestra el mensaje "Máquina: " y se mostrará el valor de la variable *nDama* que será igual al número aleatorio lanzado entre 1 y 6, luego se espera un segundo, se mostrará el mensaje "tu: " y se muestra el valor de la variable *nDaus* que será igual al número aleatorio lanzado entre 1 y 6. Se crea otra condicional en donde si el valor de la variable *nDama* es igual al valor de la variable *nDaus* muestra el mensaje "Se volverán a lanzar los dados" , sino entra a otro condicional en donde si el valor de la variable de *nDama* es mayor al valor de la variable *nDaus* se muestra el mensaje "COMIENZA MAQUINA" y el valor de la variable *bTurno* será igual a "false" dando así inicio del juego la máquina, de lo contrario si el valor de la variable *nDaus* es mayor que el valor de la variable *nDama*, se mostrará el mensaje "COMIENZA USUARIO" y la variable *bTurno* tendrá el valor de "true" dando así el inicio del juego al usuario. Se cierran los condicionales y antes de que se cierre el ciclo, los mensajes mostrados serán vistos por dos segundos y se limpiará la pantalla dando así el cierre de la función.

Se realizará que, si el valor de *bTurno* es "true" se limpia la pantalla y continuamente se muestra el mensaje de "TU TURNO" junto a "Escoge posición a jugar". Entra a un

condicional en donde si el valor de *nTurno* es menor a 9 entra a realizar la función *jusuari*. La función de tipo void #6 llamada “*jusuari*” realiza la jugada del usuario, aquí se declaran dos variables de tipo entero, *nUs* y *nConter*, se ingresa por medio del usuario el valor de *nUs*. Se crea un condicional en donde si *nUs* es menor que 1 o *nUs* es mayor que 9, se mostrará el mensaje “Ingrese una posición valida” y vuelve a llamar la función para hacer el mismo procedimientos, si el valor está entre 1 y 9 la variable *nConter* obtendrá un valor de 1 y se iniciará un ciclo, en donde para una variable de tipo entero “*i*” que tendrá un valor de 0, hasta que *i* sea menor que nueve, *i* incrementará en 1, dentro de ese ciclo se tiene un condicional en donde si la variable *nConter* es igual a la variable *nUs*, entra a otro condicional para validar cuales posiciones están ocupadas, que dice que si el vector *vCuadro* en la posición es diferente de 0 se mostrará el mensaje “No puede jugar en esta posición, escoja una casilla desocupada” y volverá a entrar la función “*jusuari*”. De lo contrario el vector *vCuadro* en la posición *i* será igual a -1 y se marcará la jugada del usuario. Se cierran los condicionales, luego la variable *nConter* incrementará en uno, se cierra el ciclo y la función.

Luego de salir del condicional incrementa el valor de *nTurno* para contar los turnos, luego se cambia el valor de *bTurno* a “falso” para realizar el cambio de jugador. Luego si el valor de *bTurno* es falso se vuelve a limpiar la pantalla, muestra el mensaje de “TURNO MAQUINA”, entra a un condicional en donde si el valor de *nTurno* es menor a 9, se realiza la función *jmaquina* en donde se realiza la jugada de la máquina. La función de tipo void #5 llamada “*jmaquina*” hace que juegue la máquina. Aquí se declaran dos variables, una es *nMover* que es igual a -1 y otra es *nJugar* que es igual a -2. Se realiza un ciclo en donde para una variable de tipo entero *i* que es igual a 0, hasta que esa variable sea menor que 9 se incrementa esa variable en 1.

Dentro del ciclo se encuentra un condicional el cual verifica las posiciones del vector que estén vacías, en donde si el vector *vCuadro* en la posición *i* es igual a 0 entonces el vector *vCuadro* en la posición *i* será igual a 1, luego dentro de ese condicional se llama la función #4 llamada “*ia_serd*” y realiza el procedimiento de esta. En la función de tipo entero #4 llamada “*ia_serd*” se declara una variable de tipo entero *nJuego*, en esta función se hace el análisis de donde jugar con respecto a la máquina. Se declara una variable que tendrá

el valor de la función “ganador”. Entra a una condicional en donde si la variable *nGanador* es diferente de 0 se retornará la multiplicación de la variable *nGanador*nJuego*. Hasta ahí solo analiza si ya hay o no un ganador. Luego se declaran tres variables tipo entero, *nMover* con un valor de -1, *nPunto* con un valor -2 e *i*. Entra un ciclo para probar todas las posiciones en donde para la variable *i* que tendrá un valor inicial de 0, hasta que *i* sea menor que nueve, *i* aumentará en 1.

Dentro del ciclo se crea un condicional en donde si el vector *vCuadro* en la posición *i* es igual a 0, el vector *vCuadro* tendrá el valor de la variable *nJuego*. Continuamente se declara una variable *nRaya* que será igual a la función *ia_serd* multiplicando por -1. Luego se abre otro condicional en donde si el valor de *nRaya* es mayor al valor de la variable *nPunto*, la variable *nPunto* tendrá el valor de la variable *nRaya*, y la variable *nMover* será igual al valor de *i* y se cierra el condicional. Dentro del primer condicional de la función se dará que el valor del vector *vCuadro* en la posición *i* será igual a 0 para así volver a vaciar las posiciones del vector, se cierra ese condicional y el ciclo, luego se crea otro condicional en donde si la variable *nMover* es igual a -1 se retornará 0 y finalmente se retornará el valor de la variable *nPunto* para así devolver la mejor opción a jugar y se cierra la función. Luego al vector *vCuadro* en la posición *i* lo iguala a 0, entra otro condicional en donde si la variable *nPunto* es mayor a la variable *nJugar*, la variable *nJugar* tendrá el valor de la variable *nPunto*.

finalmente, en esa condicional la variable *nMover* se iguala a *i* determinando así la posición en la cual se va a jugar. Se cierran las condicionales y el ciclo interno a la función y antes de finalizar la función al vector *vCuadro* en la posición del valor *nMover* será igual a 1 realizando la jugada en la posición deseada y así se cierra el proceso de la función “jmaquina. Luego el programa espera un segundo, vuelve a aumentar el valor de *nTurno* sumándole 1 y cambia el valor de *bTurno* a “true” para realizar el cambio de jugador.

Se genera un switch en donde se tiene en cuenta el que retorna la función “ganador” y con respecto a ese se realiza una operación correspondiente a cada caso. La función de tipo entero #3 llamada “ganador”, determina si hay algún ganador. Se declara la matriz de tipo entero *nPos* de tamaño 8x3 determinando las combinaciones de posiciones y/o

espacios del vector en donde se es posible ganar. Inicia un ciclo en donde para una variable de tipo entero i que es igual a 0, hasta que i sea menor que 8 se realizará el proceso con aumento en 1. Adentro del ciclo se realizará la comparación de la matriz con las posibles combinaciones para ganar con el vector. Entonces si el vector $vCuadro$ en la posición i , 0 es diferente de 0 y el vector $vCuadro$ en la posición i , 0 es igual al vector $vCuadro$ en la posición i , 1 y el vector $vCuadro$ en la posición i , 0 es igual al vector $vCuadro$ en la posición i , 2 retornará el vector $vCuadro$ en la posición i , 2.

Luego antes de finalizar la función retornará 0 y finaliza la función. En el caso 0 si la variable $nTurno$ tiene un valor de 10, muestra en pantalla el mensaje de “EMPATE”. En el caso 1, se limpia la pantalla y muestra el mensaje “PERDISTE”, entra el procedimiento de la función mostrar, se cambia el valor de la variable $bGanador$ a “true”. En el caso -1, se limpia la pantalla, muestra el mensaje “GANASTE”, entra el procedimiento de la función mostrar. La función de tipo void #2 llamada `mostrarC`, muestra el cuadro del juego en la pantalla, y llena los espacios teniendo en cuenta la función carácter para cualquier espacio del cuadro. Luego se cambia el valor de la variable $bGanador$ a “true”. Finalmente se cierra el procedimiento para los casos. Todo esto mientras $bGanador$ sea falso y $nTurno$ sea menor o igual a 9. La función de tipo carácter #1 llamada “carácter” es donde se introduce una variable de tipo entero llamada i . En esta función entra un switch aplicada para la variable i , en donde en el caso -1 retorna el carácter H, en el caso 0 retorna un espacio y para el caso 1 retorna el carácter C, luego se finaliza la función.

¿Continuamente el programa espera un segundo, limpia la pantalla y muestra el mensaje “Desea jugar?” “0=SI 1=NO”, realizando la pregunta si se desea seguir jugando para luego validar. Entra un condicional en donde si la variable $nRepeat$ es igual a 0 entonces se limpia la pantalla, si no, se limpia la pantalla, ¿se muestra el mensaje “ADIOS!!” y retorna 0;

2. Pseudo código

```
Funcion mostrarC2 <- mostrarC ( vCuadro )
  Escribir Character(vCuadro[1]),' | ',Character(vCuadro[2]),' | ',Character(vCuadro[3])
  Escribir '-----'
  Escribir Character(vCuadro[4]),' | ',Character(vCuadro[5]),' | ',Character(vCuadro[6])
  Escribir '-----'
  Escribir Character(vCuadro[7]),' | ',Character(vCuadro[8]),' | ',Character(vCuadro[9])

Fin Funcion

Funcion a <- jUsuario1 (a, vCuadro)
  Definir nUs,nConter,i Como Entero
  Leer nUs
  Si nUs<1 O nUs>9 Entonces
    Escribir 'Ingrese una posición válida'
    a <- a
  FinSi
  nConter <- 1
  Para i<-1 Hasta 9 Hacer
    Si nConter=nUs Entonces
      Si vCuadro[i]!=0 Entonces
        Escribir 'No puede jugar en esta posición, escoja una casilla
desocupada.'
        a <- a
      Sino
        vCuadro[i] <- -1
      FinSi
    FinSi
  FinPara
  nConter <- nConter+1
FinFuncion

Funcion b1Turno <- dados (b1Turno)
  Definir nDama,nDaus Como Entero
  Mientras nDama=nDaus Hacer
    nDama <- 1+azar(6)
    Escribir 'Máquina: ',nDama
    nDaus <- 1+azar(6)
    Escribir 'Tu : ',nDaus
    Si nDama=nDaus Entonces
      Escribir 'Se volverán a lanzar los dados.'
    Sino
      Si nDama>nDaus Entonces
        Escribir 'COMIENZA MÁQUINA'
        bTurno <- Falso
      Sino
        Escribir 'COMIENZA USUARIO'
        bTurno <- Verdadero
      FinSi
    FinSi
  FinMientras
  Borrar Pantalla
FinFuncion

Funcion a <- jMaquina1 (vCuadro)
```

```

Definir nMover,nJugar,i,nPunto Como Entero
nMover <- -1
nJugar <- -2
Para i<-1 Hasta 9 Hacer
    Si vCuadro[i]=0 Entonces
        vCuadro[i] <- 1
        nPunto <- -iaserd1(-1, vCuadro)
        vCuadro[i] <- 0
        Si nPunto>nJugar Entonces
            nJugar <- nPunto
            nMover <- i
        FinSi
    FinSi
FinPara
vCuadro[nMover] <- 1
FinFuncion

Funcion a <- ganador1 (vCuadro)
    Definir i1,nPos Como Entero
    Dimension nPos[8,3]
    nPos[1,1]<-1
    nPos[1,2]<-2
    nPos[1,3]<-3
    nPos[2,1]<-4
    nPos[2,2]<-5
    nPos[2,3]<-6
    nPos[3,1]<-7
    nPos[3,2]<-8
    nPos[3,3]<-9
    Para i1<-1 Hasta 8 Hacer
        Si vCuadro[nPos[i,1]]!=0 Y vCuadro[nPos[i,1]]=vCuadro[nPos[i,2]] Y
vCuadro[nPos[i,1]]=vCuadro[nPos[i,3]] Entonces
            ganador2 <- vCuadro(nPos(i))
        FinSi
    FinPara
    ganador2 <- 0
FinFuncion

Funcion a <- iaserd1 (a, vCuadro)
    Definir nGanador,nMover,i,nRaya,nPunto Como Entero
    nGanador <- ganador1(a)
    Si nGanador!=0 Entonces
        a <- nGanador*nJuego
    FinSi
    nMover <- -1
    nPunto <- -2
    Para i<-1 Hasta 9 Hacer
        Si vCuadro[i]=0 Entonces
            vCuadro[i] <- nJuego
            nRaya <- -a*(-1)
            Si nRaya>nPunto Entonces
                nPunto <- nRaya
                nMover <- i
            FinSi
            vCuadro[i] <- 0
        FinSi
    FinSi

```

```

FinPara
Si nMover=-1 Entonces
    a <- 0
FinSi
a <- nPunto
FinFuncion

```

```

Funcion iX <- character (i)
Segun i Hacer
    Caso -1:
        iX='H'
    caso 0:
        ix=''
    caso 1:
        ix='C'
Fin Segun
Fin Funcion

```

```

Funcion a <- inicializar1 (a, vCuadro)
    Definir nPos,i Como Entero
    Para i<-1 Hasta 10 Hacer
        vCuadro[i] <- nPos
    FinPara
FinFuncion

```

```

Algoritmo juego_gato
    Definir ganador,iaserd,player,nTurno,nRepeat,i,inicializar,cEnter, mostrars Como Entero
    Definir jMaquina,jUsuari,bContinue,bGanador,bTurno Como Logico
    bTurno <- Verdadero
    Escribir 'Bienvenido al juego del gato'
    Mientras nRepeat=0 Hacer
        nTurno <- 0
        bGanador <- Falso
        Escribir 'Para lanzar dados oprima Enter, para cancelar y salir oprima cualquier tecla.'
        Leer cEnter
        Si cEnter!=0 Entonces
            bContinue <- Falso
        Sino
            bContinue <- Verdadero
        FinSi
        Si bContinue=Verdadero Entonces
            bTurno <- dados(bTurno)
            Repetir
                Si bTurno=Verdadero Entonces
                    Borrar Pantalla
                    Escribir 'TU TURNO.'
                    Escribir 'Escoge posición a jugar'
                    mostrars <- mostrarC(vCuadro)
                    Si nTurno<9 Entonces
                        a<-a
                    FinSi
                    nTurno <- nTurno+1
                    bTurno <- Falso
                Sino
                    Borrar Pantalla
                    Escribir 'TURNO MÁQUINA.'
            FinRepetir
        FinMientras
    FinFuncion

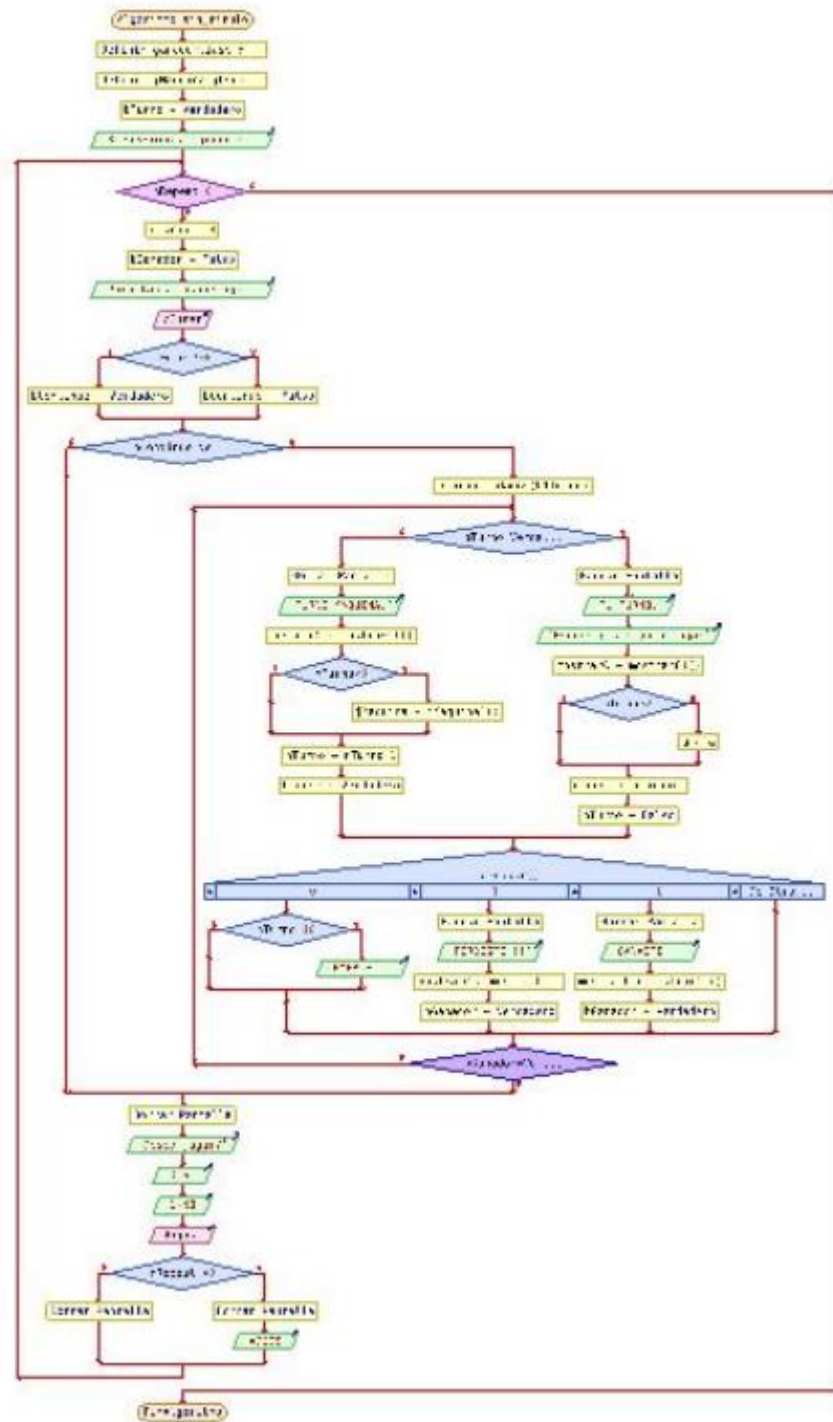
```

```

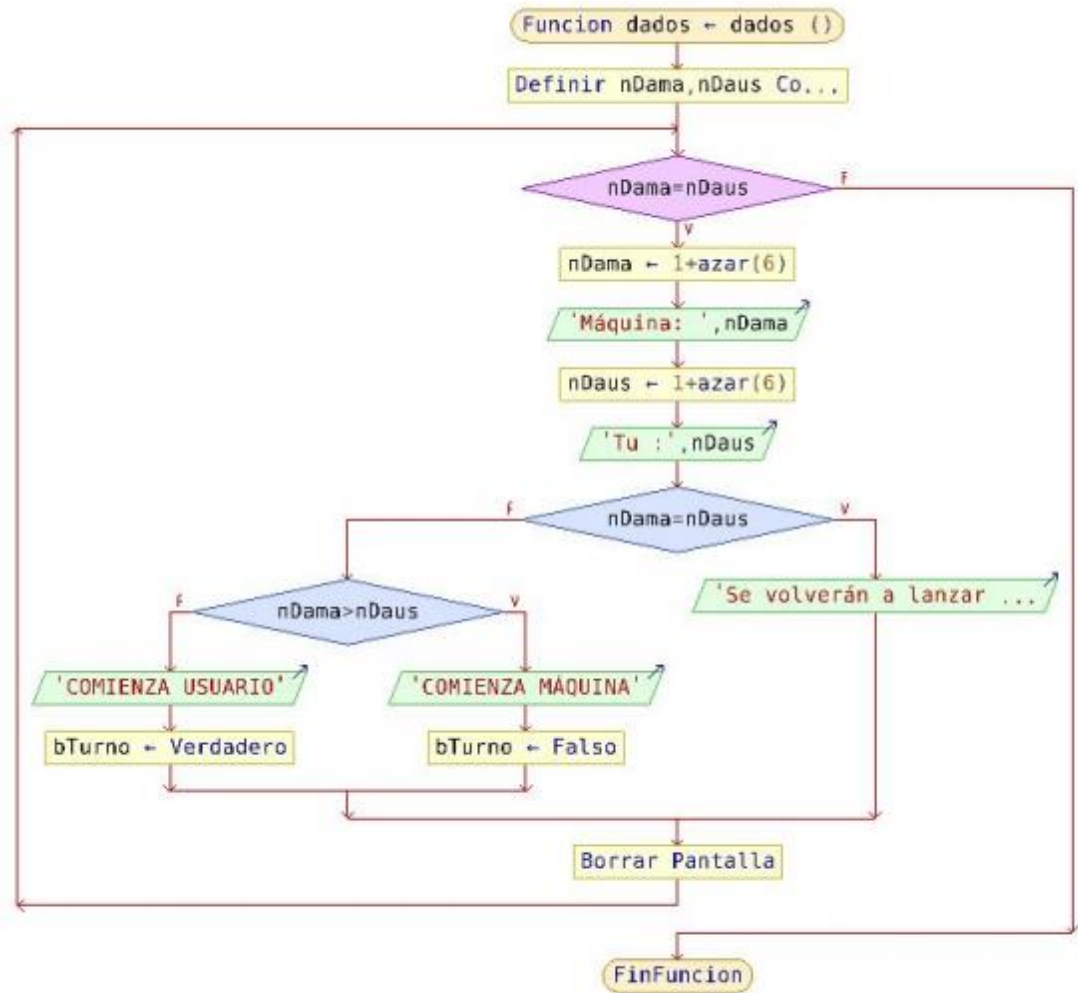
                                mostrars <- mostrarC(vCuadro)
                                Si nTurno<9 Entonces
                                    jMaquina <- jMaquina1(vCuadro)
                                FinSi
                                nTurno <- nTurno+1
                                bTurno <- Verdadero
                                FinSi
                                Segun ganador[] Hacer
                                    0:
                                        Si nTurno=10 Entonces
                                            Escribir 'EMPATE !!'
                                        FinSi
                                    1:
                                        Borrar Pantalla
                                        Escribir 'PERDISTE !!'
                                        mostrars <- mostrarC(vCuadro)
                                        bGanador <- Verdadero
                                    -1:
                                        Borrar Pantalla
                                        Escribir 'GANASTE !!'
                                        mostrars <- mostrarC(vCuadro)
                                        bGanador <- Verdadero
                                FinSegun
                                Hasta Que bGanador=Verdadero Y nTurno<=9
                                FinSi
                                Borrar Pantalla
                                Escribir 'Desea jugar?'
                                Escribir '0=SI'
                                Escribir '1=NO'
                                Leer nRepeat
                                Si nRepeat!=0 Entonces
                                    Borrar Pantalla
                                    Escribir 'ADIOS'
                                Sino
                                    Borrar Pantalla
                                FinSi
                                FinMientras
                                FinAlgoritmo

```

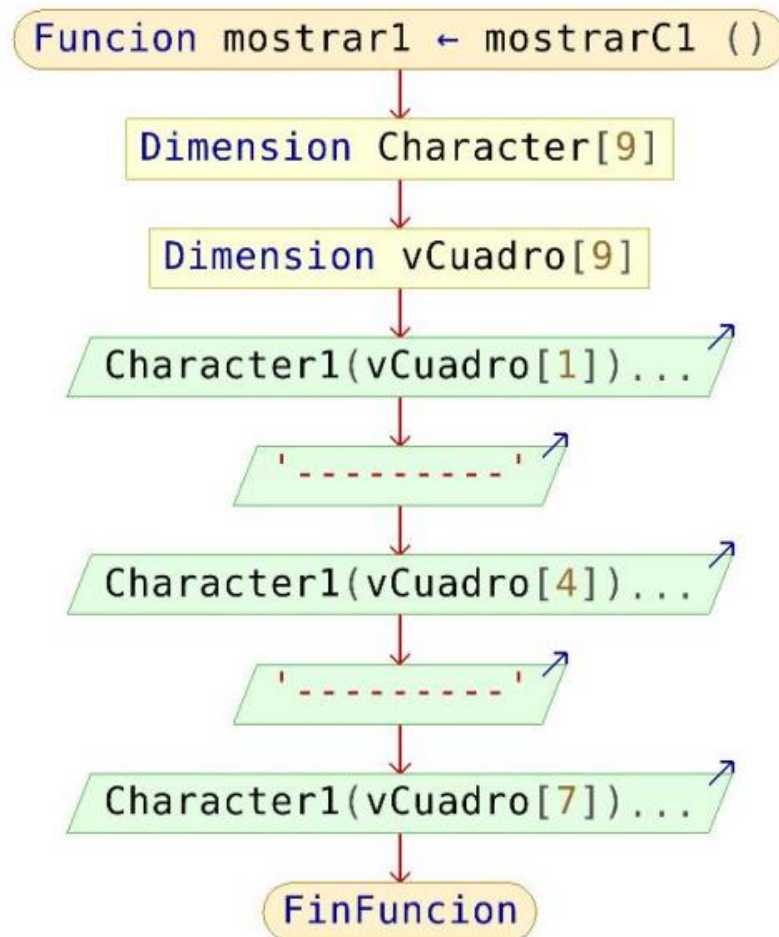
a. Función principal



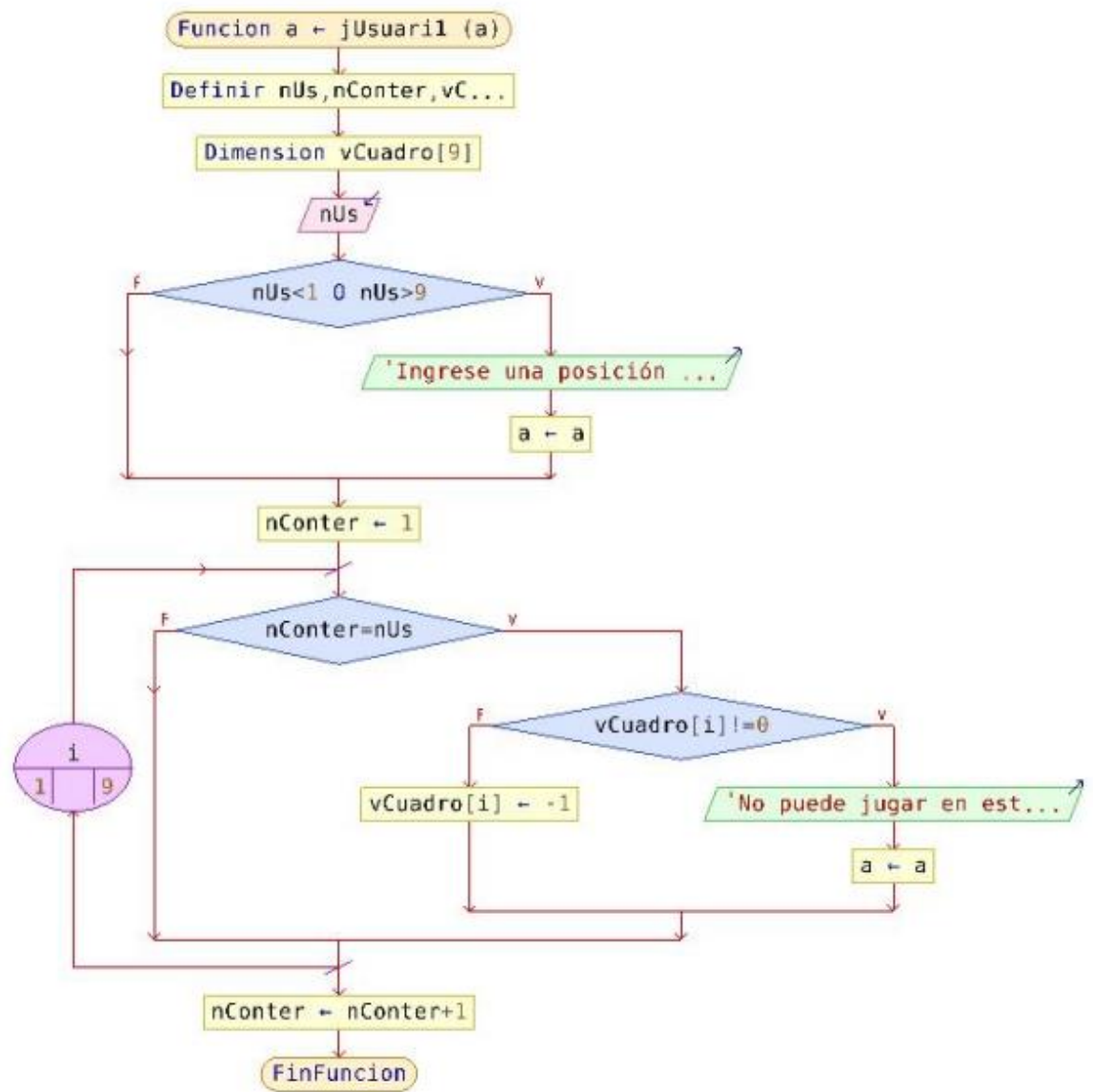
b. Función dados



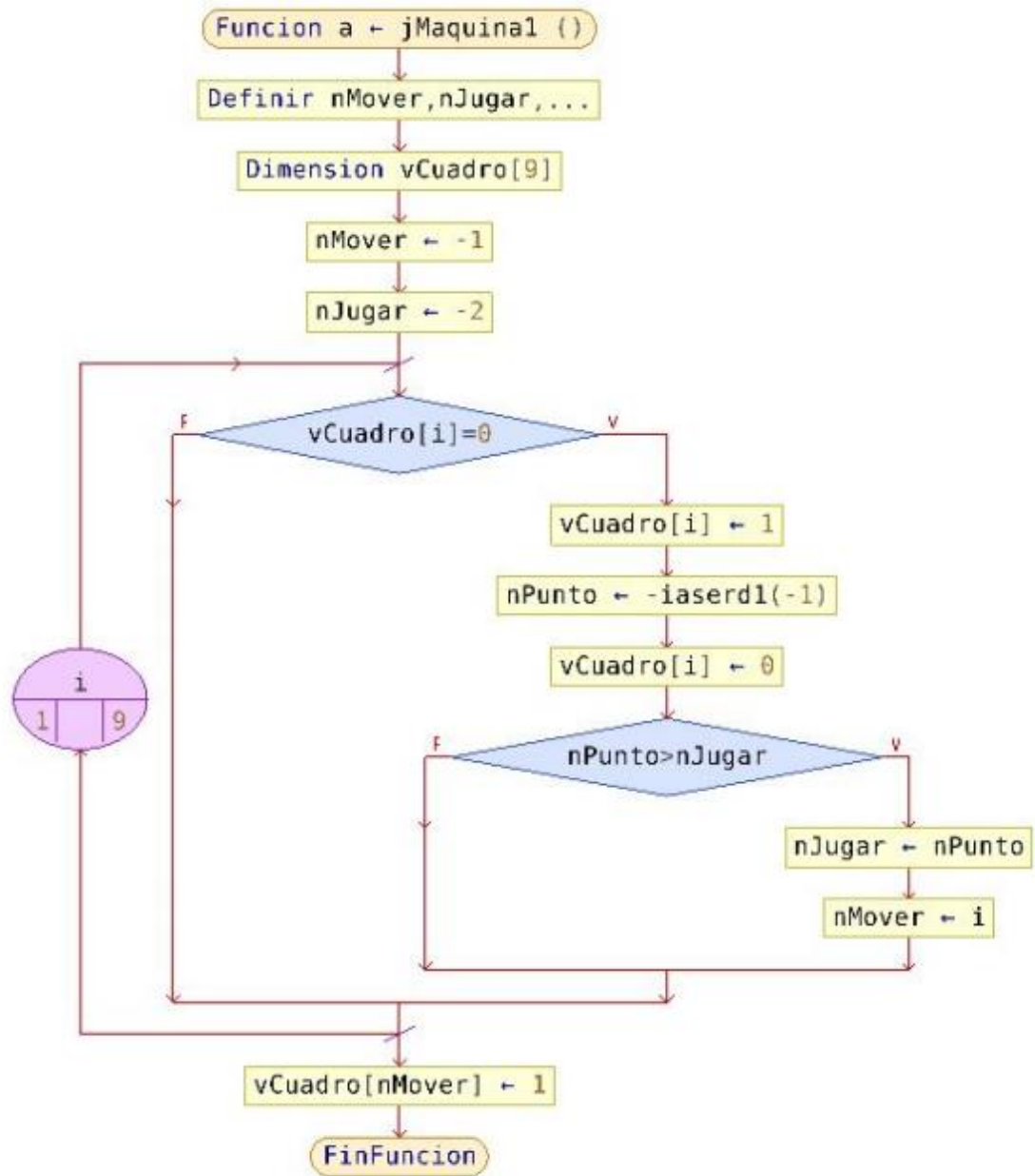
c. Función mostrarC



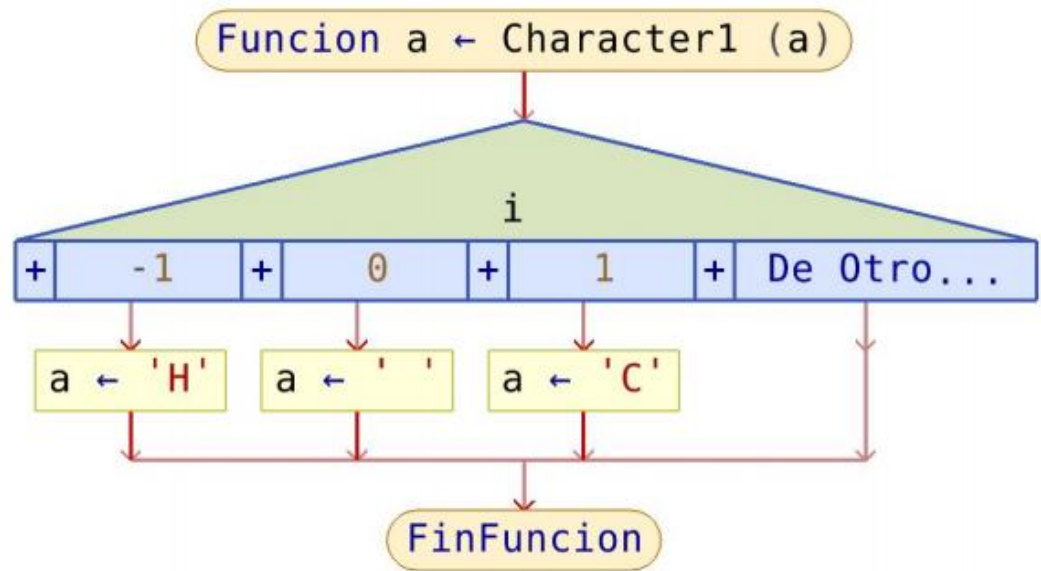
d. Función jUsuario



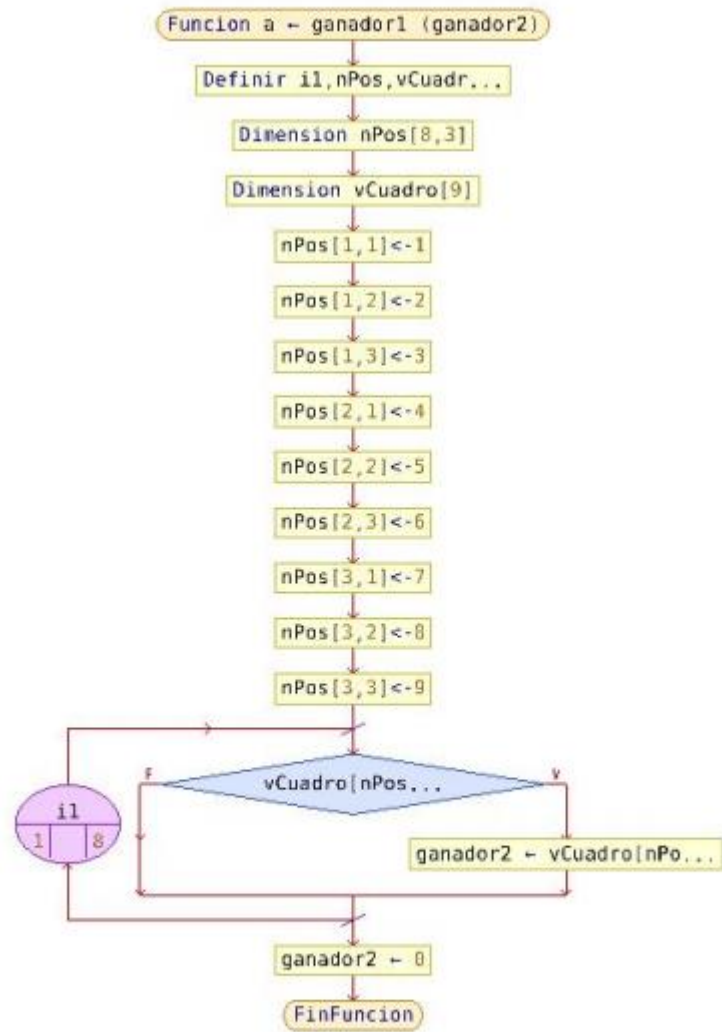
e. Función jMaquina



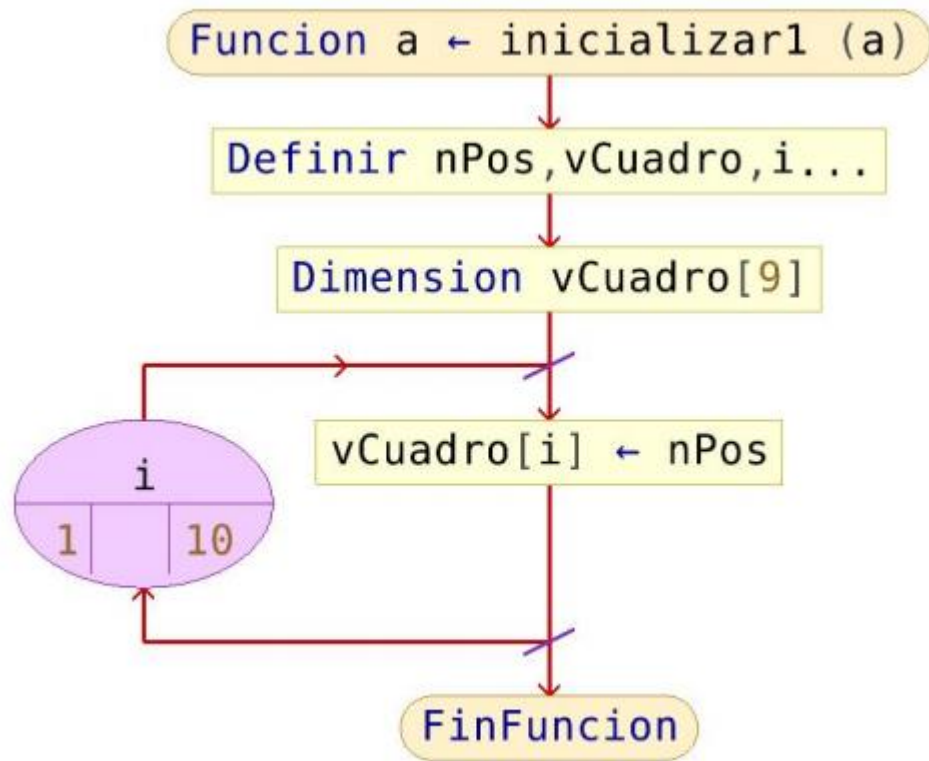
f. Función carácter



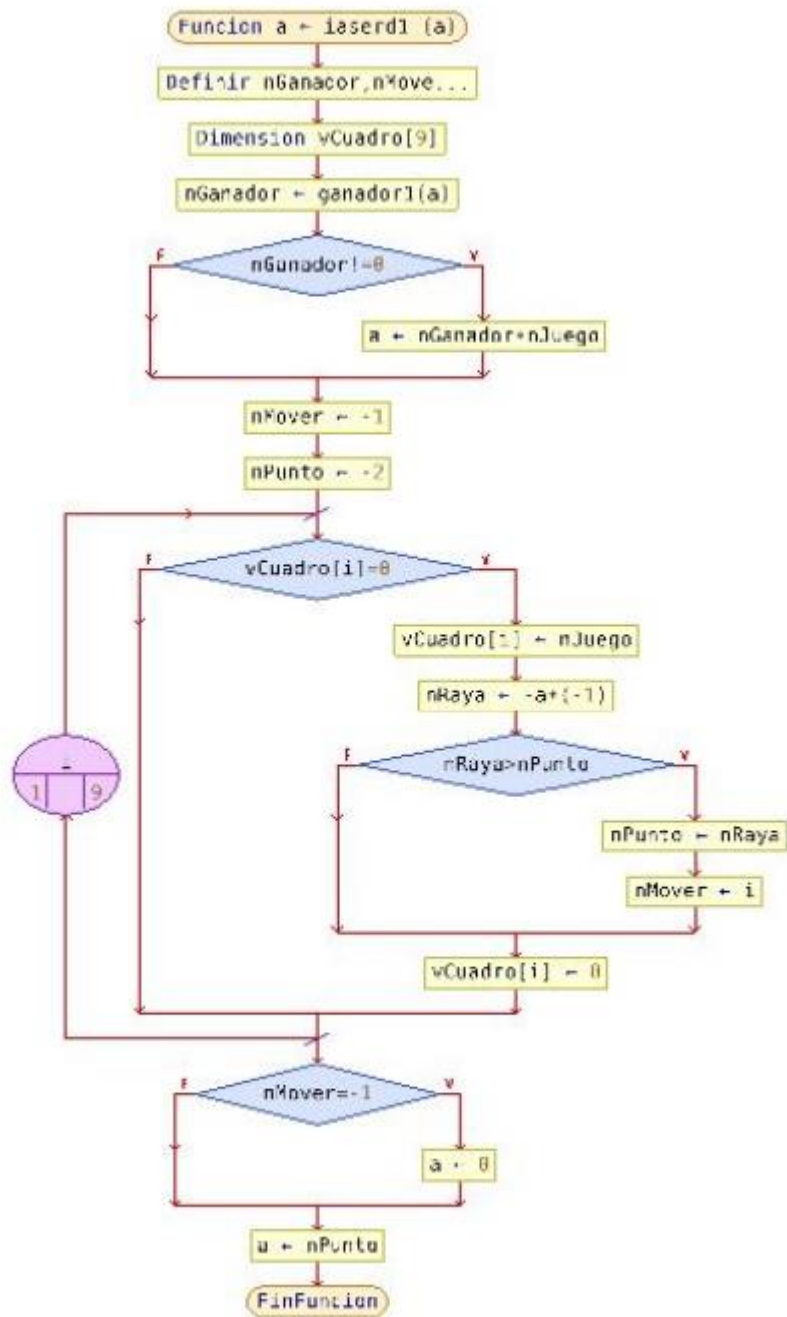
g. Función ganador



h. Función ia_serd



i. Función inicializar



4. Código fuente

```
#include<iostream >

#include<windows.h>

#include<conio.h>

//#include<stdlib.h>

#include<time.h>

//#include<sstream>

#include <stdio.h>

int vCuadro[9];//Creamos un vector global


char caracter(int i);//Funcion para mostrar en pantalla H y/o C

void mostrarC();//Funcion para mostrar el cuadro de juego

int ganador();//funcion para determinar si hay un ganador

int ia_serd(int player);//Funcion para el analisis del movimiento

void jmaquina();//Funcion para el juego de la maquina

void jusuari();//Funcion para el juego del usuario

void dados(bool &bTurno); //Funcion para lanzamiento de dados y determinar quien inicia

void inicializar();//Funcion para inicializar el vector con las posiciones del juego


using namespace std;


int main()

{

    bool bTurno=true,bContinue,bGanador=false;

    int nTurno=0,nRepeat=0;

    char cEnter;


    cout<<"t -----" <<endl<<endl;

    cout<<"t |  BIENVENIDO AL JUEGO DEL GATO  |" <<endl<<endl;

    cout<<"t -----" <<endl<<endl;


    while(nRepeat==0){

        nTurno=0;

        bGanador=false;

        inicializar();

        cout<<" Para lanzar dados oprima Enter\n Para cancelar y salir oprima cualquier tecla" <<endl<<endl;;
```

```
cEnter=getch(); //Damos a la variable el valor de la tecla oprimida
```

```
//Analizamos si la tecla presionada fue Enter
```

```
if(cEnter!=13)bContinue =false;
```

```
if(cEnter==13)bContinue =true;
```

```
if(bContinue)
```

```
{
```

```
    dados(bTurno); //Relizar lanzamiento de dados
```

```
    do //realizar ciclo hasta que haya un ganador o se genere empate
```

```
    {
```

```
        if (bTurno)
```

```
        {
```

```
            system("CLS");
```

```
            cout<<"\t TU TURNO \n  Escoge posicion a jugar \n"<<endl;
```

```
            mostrarC();
```

```
            if(nTurno<9)
```

```
            {
```

```
                jusuari(); //Juega el usuario
```

```
            }
```

```
            nTurno++; //contar los turnos
```

```
            bTurno=false; //Cambio de valor para realizar el cambio de jugador
```

```
        }
```

```
    else
```

```
    {
```

```
        system("CLS");
```

```
        cout<<"\tTURNO MAQUINA \n\n "<<endl;
```

```
        mostrarC();
```

```
        if(nTurno<9)
```

```
        {
```

```
            jmaquina(); //Juega la maquina
```

```
        }
```

```
        Sleep(1000); //Esperar un segundo
```

```
        nTurno++;
```

```
        bTurno=true; //Cambio de valor para realizar el cambio de jugador
```



```
}
```

```
switch(ganador())//Validar si hay ganador o empate
```

```
{
```

```
case 0:
```

```
if(nTurno==10)
```

```
{
```

```
cout<<"\t EMPATE !! \n"<<endl;
```

```
}
```

```
break;
```

```
case 1:
```

```
system("CLS");
```

```
cout<<"\t PERDISTE !! \n"<<endl;
```

```
mostrarC();
```

```
bGanador=true;
```

```
break;
```

```
case -1:
```

```
system("CLS");
```

```
cout<<"\t GANASTE !! \n"<<endl;
```

```
mostrarC();
```

```
bGanador=true;
```

```
break;
```

```
}
```

```
} while(!bGanador && nTurno<=9);
```

```
}
```

```
Sleep(2000);
```

```
system("CLS");
```

```
cout<<" Desea jugar? \n 0=SI 1=NO"<<endl;//Preguntar si desea jugar y validar
```

```
cin>>nRepeat;
```

```
if(nRepeat==0)
```

```
{
```

```
system("CLS");
```

```
}
```

```
else
```

```
{
```

```
system("CLS");
```

```
cout<<"\n \n \t ADIOS !!!"<<endl;
```

```
return 0;

}

}

}
```

```
//funcion para mostrar en pantalla H y/o C
```

```
char caracter(int i) {

switch(i)

{//segun lo que haya en el espacio indocado del vector

case -1:

return 'H';

case 0:

return ' ';

case 1:

return 'C';

}

}
```

```
//Funcion para mostrar el cuadro de juego
```

```
void mostrarC() {

//se pasa el espacio del vector como parametro a l funcion caracter
```

```
for(int i=0; i<9; i++){
```

```
if(i==3 || i==6){

cout<<endl;

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), FOREGROUND_GREEN);

cout<<"t-----"<<endl;

}
```

```
if(vCuadro[i]==1){

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), FOREGROUND_BLUE);//Juega maquina color azul

if(i==0 || i==3 || i==6){

cout<<"t";

}

cout<<caracter(vCuadro[i]);
```

```

} else if (vCuadro[i] == 0) {
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), FOREGROUND_GREEN); // Posicion vacia verde

    if (i == 0 || i == 3 || i == 6) {
        cout << "t";
    }

    cout << i + 1;
} else {
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), FOREGROUND_RED); // Juega humano color rojo

    if (i == 0 || i == 3 || i == 6) {
        cout << "t";
    }

    cout << caracter(vCuadro[i]);
}

if (i == 0 || i == 1 || i == 3 || i == 4 || i == 6 || i == 7) {
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), FOREGROUND_GREEN);

    cout << " | ";
}

cout << endl;

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), FOREGROUND_INTENSITY);
}

// Funcion para determinar si hay algún ganador
int ganador() {
    // matriz con las posiciones/espacios del vector posibles de ganar
    int nPos[8][3] = { {0,1,2}, {3,4,5}, {6,7,8}, {0,3,6}, {1,4,7}, {2,5,8}, {0,4,8}, {2,4,6} };

    for (int i = 0; i < 8; ++i)
    {
        // comparar la matriz con las posibles posiciones a ganar con el vector
        if (vCuadro[nPos[i][0]] != 0 && vCuadro[nPos[i][0]] == vCuadro[nPos[i][1]] && vCuadro[nPos[i][0]] ==
            vCuadro[nPos[i][2]])
        {
            return vCuadro[nPos[i][2]];
        }
    }

    return 0;
}

```

```

//Funcion para hacer analisis de donde jugar

int ia_serd(int nJuego) {
    int nGanador = ganador();
    if(nGanador != 0) return nGanador*nJuego;

    int nMover = -1;
    int nPunto = -2;
    int i;
    for(i = 0; i < 9; ++i) { //Probar para todas las posiciones
        if(vCuadro[i] == 0) {
            vCuadro[i] = nJuego;
            int nRaya = -ia_serd(nJuego*-1);
            if(nRaya > nPunto) { //seleccionar la mejor opcion
                nPunto = nRaya;
                nMover = i;
            }
            vCuadro[i] = 0; //Volver a vaciar las posiciones del vector
        }
    }
    if(nMover == -1) return 0;
    return nPunto; //devolver mejor opcion
}

//Funcion para que juegue la maquina
void jmaquina() {
    int nMover = -1;
    int nJugar = -2;
    //int i;
    for(int i = 0; i < 9; ++i) {
        if(vCuadro[i] == 0) { //verificar las posiciones del vector que estan vacias
            vCuadro[i] = 1;
            int nPunto = -ia_serd(-1); //Se llama a la funcion ia_serd
            vCuadro[i] = 0;
            if(nPunto > nJugar) {
                nJugar = nPunto;
                nMover = i; //determinar posicion en la que se va jugar
            }
        }
    }
}

```

```
}
```

```
vCuadro[nMover] = 1; //Se juega en la posicion deseada
```

```
}
```

```
//funcion para que juegue el usuario
```

```
void jusuari() {
```

```
int nUs,nConter;
```

```
cin>>nUs;
```

```
if (nUs<1 || nUs>9)//validar que ingrese posicion valida
```

```
{
```

```
cout<<" Ingrese una posicion valida"<<endl;
```

```
jusuari();
```

```
}
```

```
nConter=1;
```

```
for (int i=0;i<9;i++){
```

```
if (nConter==nUs)
```

```
{
```

```
if (vCuadro[i] != 0)//Validar que posiciones estan ocupadas
```

```
{
```

```
cout<<" No puede jugar en esta posicion, escoja una casilla desocupada"<<endl;
```

```
jusuari();
```

```
}
```

```
else
```

```
{
```

```
vCuadro[i] = -1;//marcar jugada del usuario
```

```
}
```

```
}
```

```
nConter++;
```

```
}
```

```
}
```

```
void dados(bool &bTurno)
```

```
{
```

```
int nDama=0,nDaus=0;
```

```
while (nDama == nDaus) //Lanzamiento de dados se realizara hasta que los valores de ambos dados sean diferentes
```

```
{
```

```

srand(time(NULL));

//Lanzamiento del dado de la maquina

nDama = 1 + rand() % 6;

cout<<"Maquina: "<<nDama<<endl<<endl;

Sleep(1000); //Esperar un segundo

//(Al realizar cambio de segundo se reducen las posibilidades de que los dados del usuario y la maquina tengan valores iguales)

nDaus = 1 + rand() % 6;

cout<<"Tu: "<<nDaus<<endl<<endl;

//Se decide si se volveran a lanzar los dados o que jugador inicia

if(nDama == nDaus){

cout<<"Se Volveran a lanzar los dados"<<endl;

}

else

{

if(nDama > nDaus)

{

cout<<"COMIENZA MAQUINA"<<endl;

bTurno=false;

}

else

{

cout<<"COMIENZA USUARIO"<<endl;

bTurno=true;

}

}

Sleep(2000); //Esperar dos segundos para dar tiempo de leer los mensajes en pantalla

system("CLS"); //limpiar pantalla

}

}

//Funcion para inicializar el vector

void inicializar()

{

int nPos=0;

for (int i=0;i<9;i++)

{

vCuadro[i]=nPos;

}

}

}

```

CONCLUSIÓN

En el anterior trabajo se desarrolló un programa en el lenguaje c++ en el cual se permite realizar el juego del gato entre máquina y usuario, sustentando así las bases sobre las que se realizó el proyecto final de la asignatura Programación Básica, y sustentando así los conocimientos adquiridos a lo largo de la asignatura durante el periodo 2018-1.

Además, se sustentó el trabajo realizado con las directrices a seguir, mostrando los componentes para la realización de dicho trabajo y que son comprendidos por los objetivos que se tuvieron en cuenta a la hora de realizarlo, el desarrollo del problema, en donde se analiza el problema, el diseño realizado en diagrama de flujo y en pseudocódigo, el código fuente del programa mismo en C++ y el manual de usuario indicando el funcionamiento del programa de forma didáctica.

MANUAL DEL USUARIO

Este manual está hecho para explicar el funcionamiento del programa del juego del gato realizado en el lenguaje de programación c++. Principalmente el juego consiste en ir colocando un signo en una matriz 3x3 y la idea es realizar una fila, columna o diagonal con tres símbolos iguales consecutivos para así ganar

El programa inicialmente mostrará el mensaje “BIENVENIDO AL JUEGO DEL GATO”, consecuentemente el programa mostrará el siguiente mensaje: “Para lanzar dados oprima “Enter” “Para cancelar y salir oprima cualquier tecla “. Al oprimir el botón/teclado “Enter” se lanzarán los dados. Se tendrá que realizar el lanzamiento de los dados para así definir quien iniciará jugando. Se lanzan los dados dando así dos números aleatorios entre 1 y 6. El primero será dado para la máquina y el segundo es dado al usuario. Quién obtenga el número de mayor cantidad, entre ambos será quien inicie. Se puede dar el caso en el que el numero el cual ambos obtengan sea el mismo, y aquí se mostrará el mensaje: “Se Volverán a lanzar los dados” y automáticamente se realizará nuevamente el lanzamiento de dados, esto hasta obtener dos números diferentes y definir quién será el jugador que inicie

Antes de iniciar hay que tener en cuenta que el símbolo que marcará la máquina es= “C” Y que el símbolo que ingresará y/o marcará el usuario es= “H.

Al ya haber definido quien iniciará a jugar, procedemos a jugar. Si inicia la máquina el programa mostrará el mensaje: “COMIENZA MAQUINA”, continuamente la máquina jugará y esta pondrá un símbolo marcado como “C” en alguna posición de la matriz 3x3 como, por ejemplo:

C		

Si en dado caso inicia jugando el usuario, el programa mostrará el mensaje: “COMIENZA USUARIO”, éste tendrá que elegir una posición de la matriz en la cual desee jugar. Aquellas posiciones se seleccionan ingresando un número correspondiente entre 1 y 9, cada casilla está marcada de la siguiente forma:

1	2	3
4	5	6
7	8	9

No se puede colocar y/o marcar una posición la cual esté anteriormente marcada, si se realiza esa acción, el programa mostrará el siguiente mensaje: “No puede jugar en esta posición, escoja una casilla desocupada” y dará la opción de volver elegir otra posición.

Igualmente, no se puede ingresar un número que no corresponda a ninguna de las casillas de la matriz (es decir un número entre 1 y 9), si se realiza esa acción el programa mostrará el siguiente mensaje: “Ingrese una posición valida”, y dará la opción de volver a elegir otra posición.

Al realizar haber acabado el juego se analiza quien gana o si hay algún empate, si se llena la matriz 3x3 y en ningún lugar se realiza una línea vertical, horizontal o diagonal con el mismo símbolo, el juego es declarado como empate y se muestra en pantalla el mensaje: “EMPATE”. Si la máquina realiza alguna de las líneas ya dichas con su propio símbolo (“C”), se define como ganador la máquina y muestra el mensaje: “PERDISTE”. Finalmente, el usuario es quien realiza alguna línea con su propio símbolo (“H”), se define como ganador al usuario y se muestra el mensaje: “GANASTE”.

Finalmente, al haber acabado la partida del juego del gato, sale un mensaje para elegir si se quiere seguir jugando o no, el cuál es mostrado como: “¿Desea jugar? Y mostrando continuamente: “0=SI 1=NO”. Si se quiere seguir jugando se tiene que ingresar el número “0” y así se volverá a realizar el proceso, de lo contrario se tiene

que ingresar el número “1” como opción para NO seguir jugando y el programa mostrará un mensaje de despida: “ADIÓS” y al volver a ingresar “enter” el programa se cerrará.

FORMAS DE GANAR:

H		
H		
H		

	H	
	H	
	H	

		H
		H
		H

H	H	H

H	H	H

H	H	H

H		
	H	
		H

H		
	H	
		H