

Proyecto Final
Manual Técnico



Laura Sofía Dueñas Bulla
Código: 20172020102
Andrés Felipe Wilches Torres
Código: 20172020114

Primer Semestre

Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería
Ingeniería de Sistemas
Bogotá
2017

Proyecto Final
Manual Técnico



Laura Sofía Dueñas Bulla
Código: 20172020102
Andrés Felipe Wilches Torres
Código: 20172020114

Primer Semestre

José David Álvarez Plata
Ingeniero de Sistemas

Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería
Ingeniería de Sistemas
Bogotá
2017

Tabla de contenido

Tabla de contenido.....	3
Introducción	4
Objetivo General	4
Objetivos Específicos.....	4
Desarrollo del problema	5
Análisis del problema.....	5
Planteamiento del problema.....	5
Desarrollo del juego en pseudocodigo.....	5
Diagramas de Flujo	9
Desarrollo del juego en c++	11
Conclusiones.....	25
Bibliografía	26

Introducción

El presente Manual Técnico tiene como fin describir y explicar detalladamente el desarrollo del juego del Gato (Triqui) que corresponde al proyecto final de Programación Básica, el cual se llevó a cabo en el lenguaje de programación c/c++, pseudocódigo y DFD. En donde se implementó lo aprendido en clase a lo largo del semestre utilizando estructuras de control, tipos de datos, bibliotecas, funciones (predefinidas o creadas por nosotros) y finalmente comentarios que facilitaran la comprensión del programa.

Cabe mencionar que el presente no pretende ser un curso de aprendizaje de cada una de las herramientas empleadas en la solución del problema, sino documentar y sustentar la aplicación, desarrollo y lógica que realizamos de este. Si se desea obtener un mayor detalle acerca de cada una de las herramientas utilizadas, su forma de operación y aplicación, se recomienda consultar los respectivos manuales de cada una de estas.

Objetivo General

- Proporcionar al lector una guía que facilite la comprensión del desarrollo y solución del programa del juego del Gato (Triqui).

Objetivos Específicos

- Implementar correctamente las herramientas de programación aprendidas a lo largo del semestre.
- Complementar dichas herramientas que permitieran llevar a cabo la solución del problema.

Desarrollo del problema

Análisis del problema

Planteamiento del problema

El juego del Gato (Triqui), se juega entre dos personas y consiste en que alternadamente cada uno de los jugadores hace una jugada en alguna de las casillas disponibles, intentando ser el primero en completar con sus fichas una línea de tres casillas.

El objetivo consiste en construir un programa que implemente en lenguaje C/C++ el juego del Gato de tal forma que uno de los jugadores sea una persona, la que ejecuta el programa, y que el otro jugador sea el computador.

Se dispone de 9 posiciones ordenadas de forma matricial como lo indica la figura:

1	2	3
4	5	6
7	8	9

El juego comienza cuando los jugadores lanzan un dado para determinar quien juega primero. El jugador que obtenga el valor más alto debe comenzar.

Al comienzo del juego las 9 posiciones están vacías. El jugador que inicia el juego selecciona una casilla vacía y la marca. A continuación, el segundo jugador selecciona una casilla vacía y la marca. En seguida juega nuevamente el primer jugador, marcando alguna casilla vacía. Esto continua hasta que alguno de los dos jugadores marca 3 casillas que estén en línea (puede ser diagonal), si ninguno de los jugadores logra este objetivo y no hay más casillas vacías, el juego se declara en empate.

Desarrollo del juego en pseudocodigo

Se tomó este código como base para posteriormente realizar el código en c++.

Funcion cuadrícula(triqui)

Borrar Pantalla

Escribir '-----'

Para i<-1 Hasta 3 Hacer

Si i<>1 Entonces

Escribir '-----'

FinSi

Para j<-1 Hasta 3 Hacer

Escribir ' ', triqui[i,j], ' ' Sin Saltar

FinPara

Escribir "

FinPara

Escribir '-----'

 Escribir "

FinFuncion

Funcion numero <- leernumero(variable)

 Repetir

 Escribir 'Seleccion el numero de ',variable,':' Sin Saltar

 Leer numero

 Si numero<1 O numero>3 Entonces

 Escribir 'Numero no valido'

 FinSi

 Hasta Que numero=1 O numero=2 O numero=3

FinFuncion

Algoritmo pseudocodigotriqui

 Dimension triqui[3,3]

 Para i<-1 Hasta 3 Hacer

 Para j<-1 Hasta 3 Hacer

 triqui[i,j]<-' '

 FinPara

 FinPara

 Ganador <- ' '

 desicion <- azar(2)=1

 turno <- 0

 Mientras turno<9 Y Ganador<>'C' Y Ganador<>'H' Hacer

 Si desicion Entonces

 Repetir

 i <- 1+azar(3)

 j <- 1+azar(3)

 Hasta Que triqui[i,j]=' '

```

    triqui[i,j]<-'C'
SiNo
    cuadricula(triqui)
Repetir
    i <- leernumero('fila')
    j <- leernumero('columna')
    Si triqui[i,j]<>' ' Entonces
        Escribir 'La casilla seleccionada esta
ocupada.'
        Escribir "
    FinSi
    Hasta Que triqui[i,j]=' '
    triqui[i,j]<-'H'
FinSi
i <- 1
Mientras Ganador=' ' Y i<=3 Hacer
    Si triqui[i,1]<>' ' Y triqui[i,1]=triqui[i,2] Y triqui[i,2]=triqui[i,3]
Entonces
        Ganador <- triqui[i,1]
    FinSi
    i <- i+1
FinMientras
i <- 1
Mientras Ganador=' ' Y i<=3 Hacer
    Si triqui[1,i]<>' ' Y triqui[1,i]=triqui[2,i] Y triqui[2,i]=triqui[3,i]
Entonces
        Ganador <- triqui[1,1]
    FinSi
    i <- i+1
FinMientras
Si Ganador=' ' Y triqui[1,1]<>' ' Y triqui[1,1]=triqui[2,2] Y
triqui[2,2]=triqui[3,i] Entonces

```

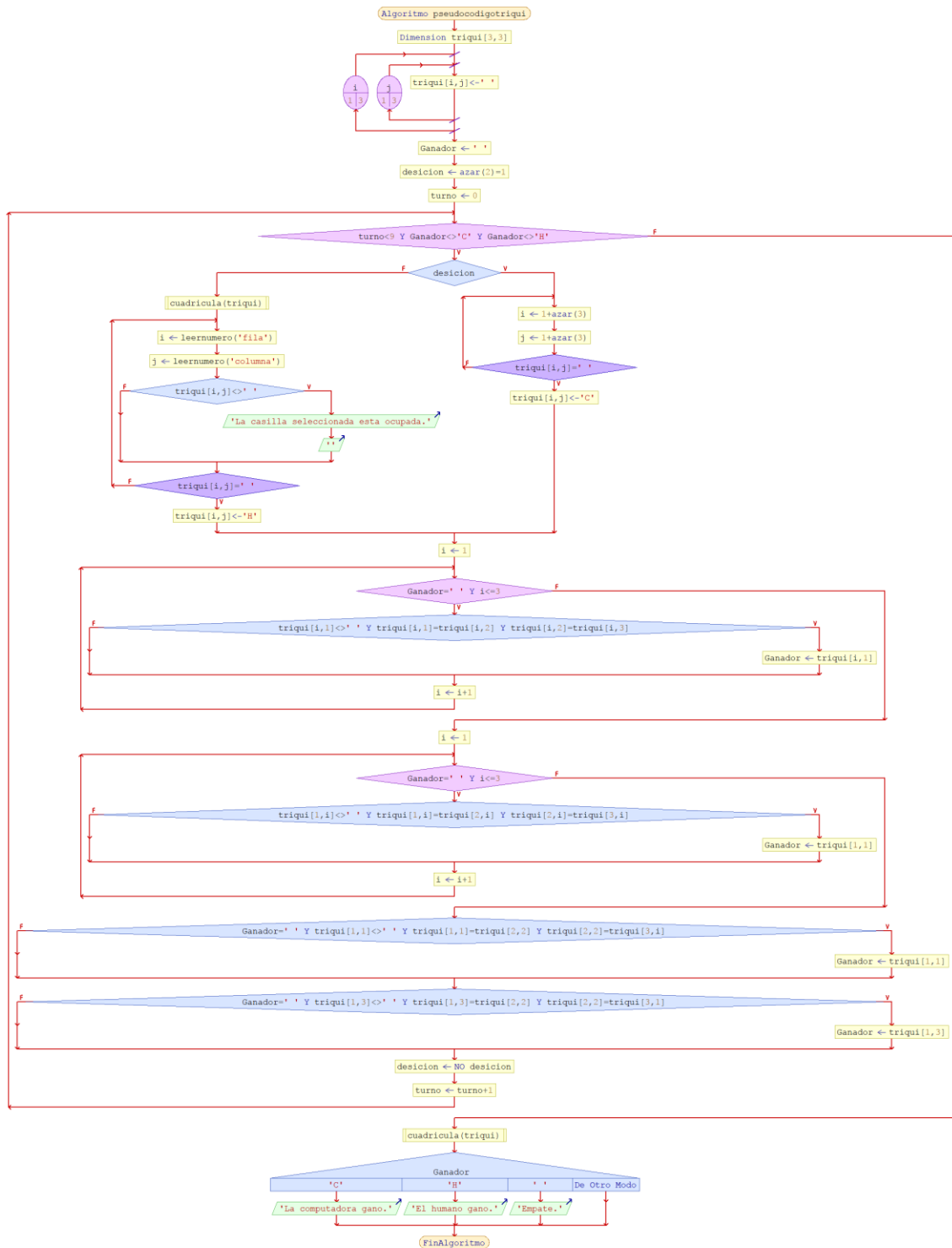
```

        Ganador <- triqui[1,1]
    FinSi
    Si Ganador=' ' Y triqui[1,3]<>' ' Y triqui[1,3]=triqui[2,2] Y
    triqui[2,2]=triqui[3,1] Entonces
        Ganador <- triqui[1,3]
    FinSi
    desicion <- NO desicion
    turno <- turno+1
FinMientras
cuadricula(triqui)
Segun Ganador Hacer
    'C':
        Escribir 'La computadora gano.'
    'H':
        Escribir 'El humano gano.'
    ' ':
        Escribir 'Empate.'
FinSegun
FinAlgoritmo

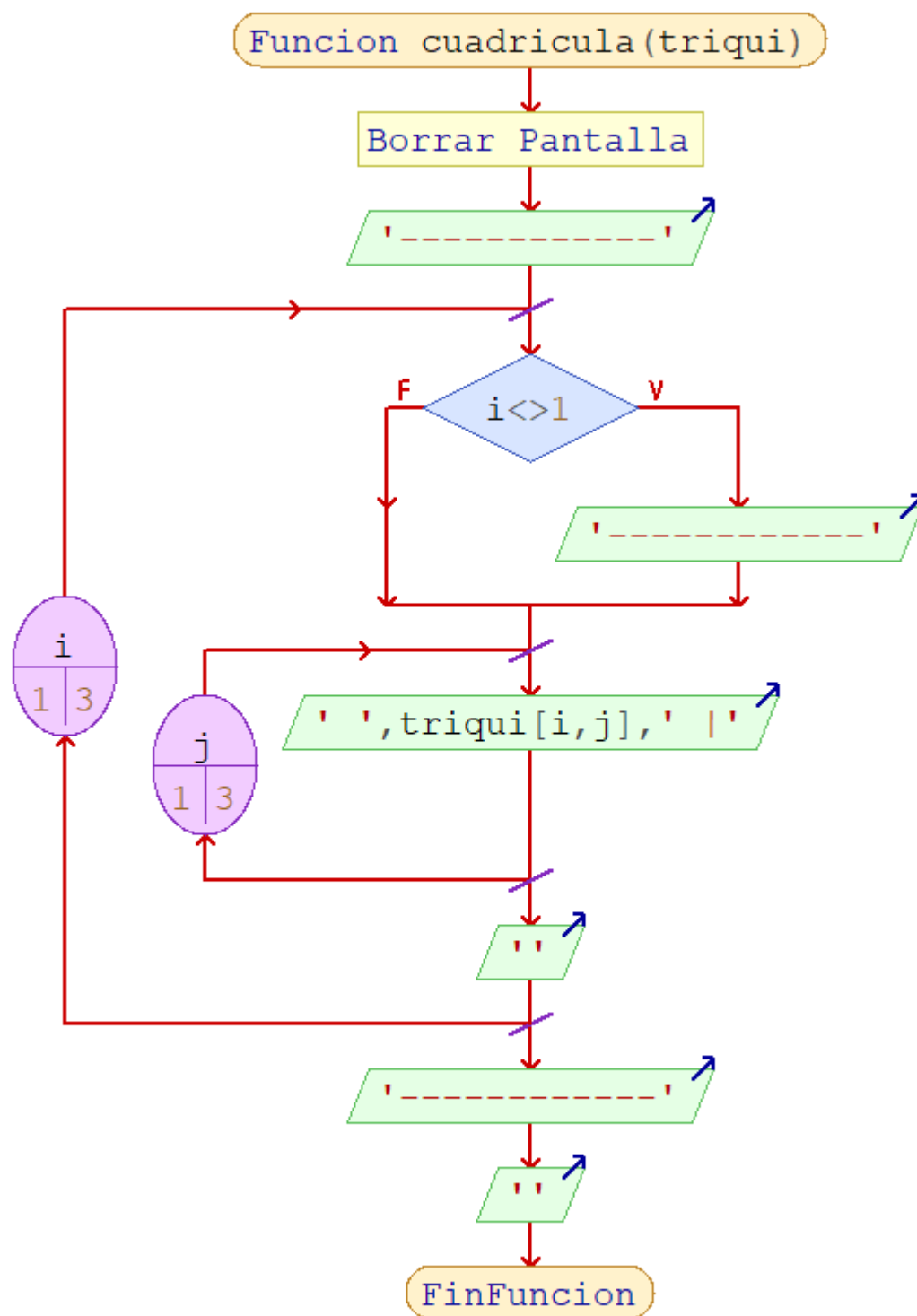
```


Diagramas de Flujo

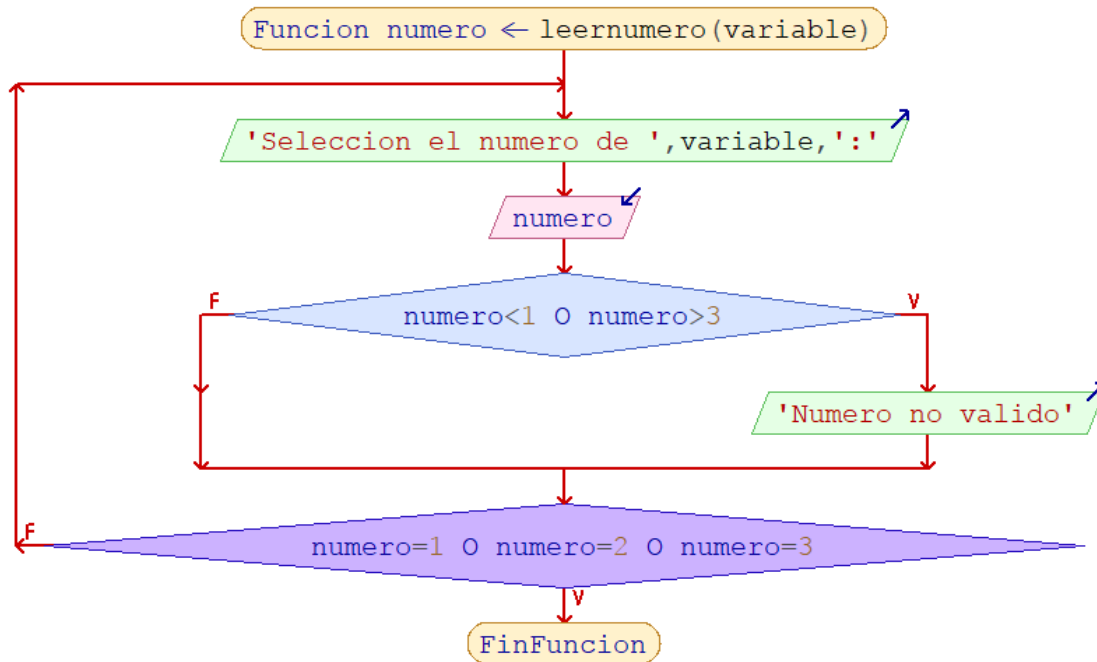
El pseudocodigo tiene como proceso principal del triqui:



El pseudocódigo tiene dos subprocesos el primero muestra la cuadrícula del triqui:



El segundo subproceso se encarga de que el usuario escoja la posición del triqui con el respectivo número de filas y columnas, más adelante se podrá observar cómo se logró optimizar este proceso al realizar el programa en c++ ya que en este último solo será necesario seleccionar el número de la posición del triqui para marcarla.



Desarrollo del juego en c++

Basándonos en el pseudocódigo y en los diagramas de flujo mostrados anteriormente realizamos el siguiente código en c++:

Para comenzar se definieron las siguientes funciones:

- `int cti (string cl){ //Esta función va a permitir convertir de cadena a int`
`Return atof(cl.c_str());`
`}`
- `void mostrar (string Triqui [3][3]){//Mostrara en pantalla el juego del triqui //actual`
`cout<<"-----"<<endl;`
`for(int ii=0;ii<3;ii++){// recorre las filas una a la vez`
`for(int jj=0;jj<3;jj++){//recorre las columnas una a la vez`
`if(Triqui[ii][jj]!="C" && Triqui[ii][jj]!="H"){`
`cout<<"|"<<" ";`
`}`
`else{//`
`cout<<"|"<<Triqui[ii][jj];`
`}`
`}`

```

        cout<<"|"<<endl;
        cout<<"-----"<<endl;
    }
    cout<<endl;

```

- **bool pertenece(vector<int> v,int x){ // Verifica si el numero x ingresado pertenece al vector v**

```

        bool mm=false;
        for(int ii=0;ii<v.size();ii++){
            if(v.at(ii)==x){
                mm=true;
                break;
            }
        }
        return mm;
    }

```
- **void resetear(string Triqui[3][3]){ // Resetea el juego para volver a jugar**

```

        for(int ii=0;ii<3;ii++){//Recorre las filas una a la vez
            for(int jj=0;jj<3;jj++){//Recorre las columnas una a la vez
                Triqui[ii][jj]=" ";//Dejando vacías las casillas del triqui //para empezar un nuevo juego
            }
        }
    }

```
- **int eleccion_computador(string Triqui[3][3],int casillas_ocupadas){//la computadora analiza el juego y toma una elección**

```

        srand(time(NULL));//genera un valor aleatorio
        int eleccion;
        int contador;
        contador=1;
        if(casillas_ocupadas>=3){ // para que el computador escoja una casilla que al marcarla gane el juego debe existir por lo menos 3 casillas ya jugadas
            for(int ii=0;ii<3;ii++){//verifica posibilidades en las filas
                for(int jj=0;jj<3;jj++){//verifica posibilidades en las columnas
                    if(Triqui[ii][jj]=="C" || Triqui[ii][jj]=="H"){
                        casillas_ocupadas+=1;//aumentan una a la //vez por cada turno
                    }
                }
            }
        }
    }

```

Se definen los siguientes ciclos for y condicionales if:

- **contador=1;**
for(int ii=0;ii<3;ii++){ //verifica en cada fila del triqui si hay un lugar en el cual el computador pueda ganar o bloquear un lugar para que el humano no gane en la siguiente partida
//A continuación se analizan todas las posibilidades que tiene el computador para ganar o bloquear un lugar para que el humano no

gane en la siguiente partida, en este caso son tres posibilidades para bloquear y tres para ganar en cada fila.

```
if(Triqui[ii][0]=="C" && Triqui[ii][1]=="C" && Triqui[ii][2]!="H" ||
Triqui[ii][0]=="H" && Triqui[ii][1]=="H" && Triqui[ii][2]!="C"){
elección=contador+2;
}
if(Triqui[ii][0]=="C" && Triqui[ii][1]!="H" && Triqui[ii][2]=="C" ||
Triqui[ii][0]=="H" && Triqui[ii][1]!="C" && Triqui[ii][2]=="H"){
elección=contador+1;
}
if(Triqui[ii][0]!="H" && Triqui[ii][1]=="C" && Triqui[ii][2]=="C" ||
Triqui[ii][0]!="C" && Triqui[ii][1]=="H" && Triqui[ii][2]=="H"){
elección=contador;
}
contador=contador+3;
}
```

Lo anterior se puede observar como:

Fila 0 →	1	2	3
Fila 1 →	4	5	6
Fila 2 →	7	8	9

POSICIONES

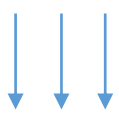
- contador=7;
for(int ii=0;ii<3;ii++){ **//verifica en cada columna del triqui si hay un lugar en el cual el computador pueda ganar o bloquear un lugar para que el humano no gane en la siguiente partida.**
//A continuación se analizan todas las posibilidades que tiene el computador para ganar o bloquear un lugar para que el humano no gane en la siguiente partida, en este caso son tres posibilidades para bloquear y tres para ganar en cada columna.

```
if(Triqui[0][ii]=="C" && Triqui[1][ii]=="C" && Triqui[2][ii]!="H" ||
Triqui[0][ii]=="H" && Triqui[1][ii]=="H" && Triqui[2][ii]!="C"){
eleccion=contador;
}
if(Triqui[0][ii]=="C" && Triqui[1][ii]!="H" && Triqui[2][ii]=="C" ||
Triqui[0][ii]=="H" && Triqui[1][ii]!="C" && Triqui[2][ii]=="H"){
eleccion=contador-3;
}
if(Triqui[0][ii]!="H" && Triqui[1][ii]=="C" && Triqui[2][ii]=="C" ||
Triqui[0][ii]!="C" && Triqui[1][ii]=="H" && Triqui[2][ii]=="H"){
eleccion=contador-6;
}
contador=contador+1;
```

}

Lo anterior se puede observar como:

Columnas 0 1 2



1	2	3
4	5	6
7	8	9

POSICIONES

- **//Verifica las diagonales del triqui si hay un lugar en el cual el computador pueda ganar o bloquear un lugar para que el humano no gane en la siguiente partida**

```
if(Triqui[0][0]=="C" && Triqui[1][1]=="C" && Triqui[2][2]!="H" ||
Triqui[0][0]=="H" && Triqui[1][1]=="H" && Triqui[2][2]!="C"){
eleccion=9;
}
if(Triqui[0][0]=="C" && Triqui[1][1]!="H" && Triqui[2][2]=="C" ||
Triqui[0][0]=="H" && Triqui[1][1]!="C" && Triqui[2][2]=="H"){
eleccion=5;
}
if(Triqui[0][0]!="H" && Triqui[1][1]=="C" && Triqui[2][2]=="C" ||
Triqui[0][0]!="C" && Triqui[1][1]=="H" && Triqui[2][2]=="H"){
eleccion=1;
}
if(Triqui[2][0]=="C" && Triqui[1][1]=="C" && Triqui[0][2]!="H" ||
Triqui[2][0]=="H" && Triqui[1][1]=="H" && Triqui[0][2]!="C"){
eleccion=3;
}
if(Triqui[2][0]=="C" && Triqui[1][1]!="H" && Triqui[0][2]=="C" ||
Triqui[2][0]=="H" && Triqui[1][1]!="C" && Triqui[0][2]=="H"){
eleccion=5;
}
if(Triqui[2][0]!="H" && Triqui[1][1]=="C" && Triqui[0][2]=="C" ||
Triqui[2][0]!="C" && Triqui[1][1]=="H" && Triqui[0][2]=="H"){
eleccion=7;
}
```

Lo anterior se puede observar como:

Diagonales

1	2	3
4	5	6
7	8	9

POSICIONES

```

if(eleccion!=1 && eleccion!=2 && eleccion!=3 && eleccion!=4 &&
eleccion!=5 && eleccion!=6 && eleccion!=7 && eleccion!=8 &&
eleccion!=9){
    eleccion=14;
}
    }else{
        eleccion=14;
    }
if(eleccion==14){ //Si no hay una casilla en la cual el computador
pueda ganar o el computador pueda bloquear al humano se busca
una casilla vacía y el computador toma una aleatoria

    int cont=1;
    vector<int> casillas_vacias;
    for(int ii=0;ii<3;ii++){ //se le agrega al vector casillas_vacias las
posiciones de las casillas del triqui vacias
        for(int jj=0;jj<3;jj++){
            if(Triqui[ii][jj]!="C" && Triqui[ii][jj]!="H"){
                casillas_vacias.push_back(ii+jj+cont);
            }
        }
        cont+=2;
    }
    eleccion=casillas_vacias.at(rand()%(casillas_vacias.size()));
}

    return eleccion;

```

Otra de las funciones definidas fue la siguiente:

- void marcar(string Triqui[3][3],int posicion,string X){ // **marcar una posicion del triqui con la variable X**

```

if(posicion==1){
    Triqui[0][0]=X;
}
if(posicion==2){
    Triqui[0][1]=X;
}

if(posicion==3){
    Triqui[0][2]=X;
}
if(posicion==4){
    Triqui[1][0]=X;
}
if(posicion==5){
    Triqui[1][1]=X;
}

```

```

    }
    if(posicion==6){
        Triqui[1][2]=X;
    }
    if(posicion==7){
        Triqui[2][0]=X;
    }
    if(posicion==8){
        Triqui[2][1]=X;
    }
    if(posicion==9){
        Triqui[2][2]=X;
    }
}

• string ganador(string Triqui[3][3]){ // verifica quien es el ganador en la
//partida
string gana;
// verificando si gana el computador
for(int i=0;i<3;i++){
    if((Triqui[i][0]=="C") && (Triqui[i][1]=="C") && (Triqui[i][2]=="C")){
        gana="C";
    }
}
for(int i=0;i<3;i++){
    if((Triqui[0][i]=="C") && (Triqui[1][i]=="C") && (Triqui[2][i]=="C")){
        gana="C";
    }
}
if((Triqui[0][0]=="C") && (Triqui[1][1]=="C") && (Triqui[2][2]=="C")){
    gana="C";
}
if((Triqui[2][0]=="C") && (Triqui[1][1]=="C") && (Triqui[0][2]=="C")){
    gana="C";
}
// verificando si gana el humano
for(int i=0;i<3;i++){
    if((Triqui[i][0]=="H") && (Triqui[i][1]=="H") && (Triqui[i][2]=="H")){
        gana="H";
    }
}
for(int i=0;i<3;i++){
    if((Triqui[0][i]=="H") && (Triqui[1][i]=="H") && (Triqui[2][i]=="H")){
        gana="H";
    }
}

```



```

    }
}
if((Triqui[0][0]=="H") && (Triqui[1][1]=="H") && (Triqui[2][2]=="H")){
    gana="H";
}
if((Triqui[2][0]=="H") && (Triqui[1][1]=="H") && (Triqui[0][2]=="H")){
    gana="H";
}
return gana;
}

```

Después de definir las funciones que usaremos a lo largo del programa se definen las variables para el lanzamiento del dado, para llenar el triqui y para mostrar en pantalla las reglas del juego:

- string Triqui [3][3];
- int Dadoh=0,Dadoc=0,first=0,second=0;
- string desicion;
- string pdado;
- string Desicionreglas;
- int computador,humano;
- string humano2;
- int contadoreglas=1;
- srand(time(NULL));//**Genera números aleatorios**

Las reglas se describen claramente a continuación:

```
cout<<"¡Bienvenido al juego del triqui!"<<endl;
```

```
cout<<"¿Deseas conocer las reglas?(s=1)"<<endl;
```

```
cin>> Desicionreglas;
```

```
if (Desicionreglas == "s"){
```

```
do {
```

```
    contadoreglas=1;
```

```
cout<<"1)En este juego competiras contra la computadora."<<endl;
```

```
cout<<"2)Deberas seguir las instrucciones tal y como las plantea el programa."<<endl;
```

```
cout<<"3)Para llenar una casilla deberas oprimir el numero de la respectiva casilla,"<<endl;
```

```
cout<<" para saber que casilla es cuenta desde arriba a la izquierda (1), hacia la derecha,"<<endl;
```

```
cout<<" bajando a la siguiente columna leyendo de izquierda a derecha, hasta llegar a la ultima posicion"<<endl;
```

```

cout<<" la cual es abajo a la derecha (9)"<<endl;
cout<<" Ejemplo: "<<endl;
cout<<" "<<endl;
for(int ii=0;ii<3;ii++){
for(int jj=0;jj<3;jj++){
if(Triqui[ii][jj]!="C" && Triqui[ii][jj]!="H"){
cout<<"|"<<" "<<contadoreglas<<" ";
contadoreglas++;
}
}
cout<<"|"<<endl;
cout<<"-----"<<endl;
}
cout<<endl;
cout<<"¿Deseas volver a leer las reglas?(s=1)"<<endl;
cin>>Desicionreglas;
}while(Desicionreglas == "s");
cout<<"Muy bien vamos a empezar"<<endl;

```

Una vez el usuario ya conoce las reglas del juego los jugadores lanzan un dado para determinar quien juega primero. El jugador que obtenga el valor más alto debe comenzar.

Dentro de 1 do while se anidaron 2 do while:

El primero:

```

do{
    cout<<"lance su dado oprimiendo s"<<endl; //Se lanza el dado
    cin>>pdado;
    if(pdado!="s"){
        cout<<"entrada invalida"<<endl;
    }
}while (pdado != "s");//unicamente se lanza el dado mientras que el usuario oprima s

```

Dadoh = 1+rand()%(7-1);**//se generan los numeros lanzados por el dado humano**

cout<<"El humano saco: "<<Dadoh<<endl;

Dadoc = 1+rand()%(7-1);**//se generan los numeros lanzados por el dado de la maquina**

cout<<"La computadora saco: "<<Dadoc<<endl;

En caso de un empate:

if (Dadoc == Dadoh){

cout<<"Sacaron el mismo numero, vuelvan a lanzar los dados"<<endl;

}else{

}

El segundo while:

}while (Dadoh == Dadoc);**//se creó para lanzar nuevamente el dado en caso de //empate**

if (Dadoh>Dadoc){**//se define quien empieza**

cout<<"Empieza el humano"<<endl;

first = Dadoh;**//variable first como primer para llenar el triqui**

second = Dadoc;**//variable second como segundo para llenar el**

triqui

}else{

cout<<"Empieza la computadora"<<endl;

first = Dadoc;

second = Dadoh;

}

int kk=1;

int cc; **// cc=0 empieza el humano, cc=1 empieza el computador**

mostrar(Triqui);

if(Dadoh>Dadoc){ **// es turno del humano**

cc=0;

}else{ **// es turno de la computadora**

```
cc=1;
```

```
}
```

Una vez se determina quien inicia a jugar se comienza el triqui:

```
string g; // g es el ganador
```

```
int casillas_ocupadas; // cantidad de casillas ya jugadas
```

```
if(cc==0){//Empieza el humano
```

```
while(kk<=5){
```

```
int cont=1;
```

```
vector<int> casillas_vacias;
```

```
for(int ii=0;ii<3;ii++){
```

```
for(int jj=0;jj<3;jj++){
```

```
if(Triqui[ii][jj]!="C" && Triqui[ii][jj]!="H"){
```

```
casillas_vacias.push_back(ii+jj+cont);
```

```
}
```

```
}
```

```
cont+=2;
```

```
}
```

```
do{
```

```
cout<<"Es tu turno:"<<endl;
```

```
cin>>humano2;
```

```
humano=cti(humano2);
```

```
if(pertenece(casillas_vacias,humano)!=true){
```

```
cout<<"entrada invalida"<<endl;
```

```
}
```

```
}while(pertenece(casillas_vacias,humano)!=true);
```

```
marcar(Triqui,humano,"H");
```

```
mostrar(Triqui);
```

```
if(kk>=3){ // después de la quinta partida existe la posibilidad de que  
//haya un ganador
```

```

g=ganador(Triqui);
if(g=="H"){
cout<<"El humano gana"<<endl;
    break;
}
if(g=="C"){
cout<<"La computadora gana"<<endl;
    break;
}
}
if(kk!=5){
cout<<"Es turno de la computadora"<<endl;
    casillas_ocupadas=0;
    for(int ii=0;ii<3;ii++){
        for(int jj=0;jj<3;jj++){
            if(Triqui[ii][jj]=="C" || Triqui[ii][jj]=="H"){
                casillas_ocupadas+=1;
            }
        }
    }

computador=eleccion_computador(Triqui,casillas_ocupadas);
cout<<computador<<endl;
marcar(Triqui,computador,"C");
mostrar(Triqui);
g=ganador(Triqui);
if(g=="H"){
cout<<"El humano gana"<<endl;
    break;
}
if(g=="C"){

```

```

        cout<<"La computadora gana"<<endl;
                                break;
                                }
        }
        kk++;
    }
    if(g!="C" && g!="H"){ // si no hay un ganador entonces es un
empate
        cout<<"Se ha producido un empate"<<endl;
    }
}

if(cc==1){//Empieza el computador
    while(kk<=5){
        cout<<"Es turno de la computadora"<<endl;
        casillas_ocupadas=0;
        for(int ii=0;ii<3;ii++){
            for(int jj=0;jj<3;jj++){
                if(Triqui[ii][jj]=="C" || Triqui[ii][jj]=="H"){
                    casillas_ocupadas+=1;
                }
            }
        }
        computador=eleccion_computador(Triqui,casillas_ocupadas);
        cout<<computador<<endl;
        marcar(Triqui,computador,"C");
        mostrar(Triqui);
        if(kk>=3){ // después de la quinta partida existe la posibilidad de que
haya un ganador
            g=ganador(Triqui);
            if(g=="C"){
                cout<<"La computadora gana"<<endl;

```

```

        break;
    }
    if(g=="H"){
        cout<<"El humano gana"<<endl;
        break;
    }
}

if(kk!=5){
    int cont=1;
    vector<int> casillas_vacias;
    for(int ii=0;ii<3;ii++){
        for(int jj=0;jj<3;jj++){
            if(Triqui[ii][jj]!="C" && Triqui[ii][jj]!="H"){

                casillas_vacias.push_back(ii+jj+cont);

            }
        }
        cont+=2;
    }

    do{
        cout<<"Es tu turno:"<<endl;
        cin>>humano2;
        humano=cti(humano2);
        if(pertenece(casillas_vacias,humano)!=true){
            cout<<"entrada invalida"<<endl;
        }
    }while(pertenece(casillas_vacias,humano)!=true);
    marcar(Triqui,humano,"H");
    mostrar(Triqui);
    g=ganador(Triqui);
    if(g=="H"){

```

```

cout<<"El humano gana"<<endl;
        break;
    }
    if(g=="C"){
        cout<<"La computadora gana"<<endl;
            break;
        }
    }

    kk++;
}

```

En caso de empate:

```

}
if(g!="C" && g!="H"){ // si no hay un ganador entonces es un empate
    cout<<"Se ha producido un empate"<<endl;
}
}

```

Una vez terminado el juego se reinicia la partida en donde el humano decide si quiere jugar nuevamente o no:

resetear(Triqui); // **reinicia la partida**

// Triqui

```

do{
    cout<<"¿Desea jugar nuevamente (0 = Si / 1 = No)?"<<endl;
        cin>>desicion;
        if(desicion!="0" && desicion!="1"){
            cout<<"entrada invalida"<<endl;
        }
        }while (desicion!="0" && desicion!="1");
    }while (desicion=="0");
    return 0;
}

```


Conclusiones

- Con el fin de que en el juego el computador fuese inteligente al momento de escoger sus jugadas para ganar o bloquear la siguiente partida del humano, se hallaron todas las posibilidades que tenía el computador para ganar o bloquear en cada fila y columna, lo cual generó que en cierta medida el computador estuviese al nivel del humano.
- Se definieron claramente las reglas del juego dado caso que el usuario las desconociera.
- Con el fin de mejorar la comprensión de las herramientas de programación utilizadas en la solución del problema, entre ellas: estructuras de control, tipos de datos, bibliotecas, funciones, etc. Se agregaron comentarios que facilitaron dicho proceso.
- Se optimizó el programa de pseudocódigo al realizarlo en c++ al momento de escoger la posición del triqui que se deseaba marcar, reduciendo el tiempo entre las partidas de los jugadores.
- Se logró llevar a cabo el propósito principal que consistía en implementar las herramientas de programación aprendidas en clase a lo largo del semestre integrándolas correctamente en la solución del problema.

Bibliografía

<http://ri.ufg.edu.sv/jspui/bitstream/11592/7034/11/338.4791-A473d-Anexo%207.pdf>

<https://www.programarya.com/Cursos/C++/Estructuras-de-Datos>

<https://www.programarya.com/Cursos/C++/Funciones>