**PSEUDOCÓDIGO**

**Algoritmo “Cuatro en Línea”**

**Declaración:**

* **Tipos:**

tablero: Arreglo [6,7] de caracter

* **Variables:**

tab:tablero

tot\_jug,gan,gan2,num,err: entero

salir\_apl: cadena

nom\_jug1,nom\_jug2,opc: cadena

**procedimiento** estadísticas

**inicio**

si (tot\_jug=0) entonces

escribir(“No se han jugado partidas”)

sino

escribir(“Estadísticas generales del juego: ”)

escribir (“Total de partidas jugadas: ”,tot\_jug)

escribir(“\*Jugador 1”)

escribir(“ Nombre: ”,nom\_jug1)

escribir(“ Partidas ganadas: ”,gan)

escribir(“ Empates: ”,tot\_jug-(gan+gan2))

escribir(“ Partidas perdidas: ”gan2)

escribir(“\*Jugador 2”)

escribir(“ Nombre: ”,nom\_jug2)

escribir(“ Partidas ganadas: ”,gan2)

escribir(“ Empates: ”,tot\_jug-(gan+gan2))

escribir(“ Partidas perdidas: ”gan)

fin si

**fin procedimiento**

**procedimiento** limpiar\_tablero (↑↓ tb:tablero)

**variables**

x,y:entero

**inicio**

para x (1 hasta 6) hacer

para y (1 hasta 7) hacer

tb[x,y]← ‘ ‘

fin para

fin para

**fin procedimiento**

**procedimiento** dibujar\_tablero (tb:tablero)

**variables**

x,y:entero

**inicio**

escribir (“Tablero de juego: ”)

escribir (“#############################”)

para x (1 hasta 6) hacer

para y (1 hasta 7) hacer

escribir (“# ”)

escribir (tb[x,y])

escribir (“ ”)

fin para

escribir (“#”)

escribir (“#############################”)

fin para

**fin procedimiento**

**función** tablero\_lleno (tb:tablero): booleano

**variables**

x,y,cont:entero

**inicio**

para x (1 hasta 6) hacer

para y (1 hasta 7) hacer

si (tb[x,y]<> ‘ ‘) entonces

cont← cont + 1

fin si

fin para

fin para

si (cont=42) entonces

tablero\_lleno←verdadero

sino

tablero\_lleno←falso

fin si

**fin función**

**función** encontrar\_ganador (x,y: entero; tb:tablero): booleano

**variables**

i,j,Acons,Bcons: entero

encontrado: booleano

**inicio**

Acons←0

Bcons←0

encontrado←falso

para j (1 hasta 7) hasta

si (tb[x,j]= ‘A’) entonces

Acons←Acons+1

Bcons←0

sino si (tb[x,j]= ‘B’) entonces

Bcons←Bcons+1

Acons←0

fin si

si ((Acons>=4) O (Bcons>=4)) entonces

encontrado←verdadero

fin si

si (tb[x,j]= ‘ ‘) entonces

Acons←0

Bcons←0

fin si

fin para

Acons←0

Bcons←0

para i (1 hasta 6) hasta

si (tb[i,y]= ‘A’) entonces

Acons←Acons+1

Bcons←0

sino si (tb[i,y]= ‘B’) entonces

Bcons←Bcons+1

Acons←0

fin si

si ((Acons>=4) O (Bcons>=4)) entonces

encontrado←verdadero

fin si

si (tb[i,y]= ‘ ‘) entonces

Acons←0

Bcons←0

fin si

fin para

Acons←0

Bcons←0

i←x

j←y

mientras ((i>1) Y (j>1)) hacer

i←i-1

j←j-1

fin mientras

mientras ((j<=7) Y (i<=6)) hacer

si (tb[i,j]= ‘A’) entonces

Acons←Acons+1

Bcons←0

sino si (tb[i,j]= ‘B’) entonces

Bcons←Bcons+1

Acons←0

fin si

si ((Acons>=4) O (Bcons>=4)) entonces

encontrado←verdadero

fin si

si (tb[i,j]= ‘ ‘) entonces

Acons←0

Bcons←0

fin si

i←i+1

j←j+1

fin mientras

Acons←0

Bcons←0

i←x

j←y

mientras ((i<7) Y (j>1)) hacer

i←i+1

j←j-1

fin mientras

mientras ((j<=7) Y (i>=1)) hacer

si (tb[i,j]= ‘A’) entonces

Acons←Acons+1

Bcons←0

sino si (tb[i,j]= ‘B’) entonces

Bcons←Bcons+1

Acons←0

fin si

si ((Acons>=4) O (Bcons>=4)) entonces

encontrado←verdadero

fin si

si (tb[i,j]= ‘ ‘) entonces

Acons←0

Bcons←0

fin si

i←i-1

j←j+1

fin mientras

encontrar\_ganador←encontrado

**fin función**

**procedimiento** perfil\_jug1

**inicio**

escribir (“Perfil Jugador 1:”)

escribir (“Nombre actual del jugador: ”,nom\_jug1)

escribir (“Ingrese el nuevo nombre del jugador:”)

leer (nom\_jug1)

mientras (nom\_jug=nom\_jug2) hacer

escribir (“Nombre repetido. Ingrese nuevamente el nombre”)

leer (nom\_jug1)

fin mientras

escribir(“El nuevo nombre del jugador 1 ha sido guardado exitosamente”)

**fin procedimiento**

**procedimiento** perfil\_jug2

**inicio**

escribir (“Perfil Jugador 2:”)

escribir (“Nombre actual del jugador: ”,nom\_jug2)

escribir (“Ingrese el nuevo nombre del jugador:”)

leer (nom\_jug2)

mientras (nom\_jug=nom\_jug2) hacer

escribir (“Nombre repetido. Ingrese nuevamente el nombre”)

leer (nom\_jug2)

fin mientras

escribir(“El nuevo nombre del jugador 2 ha sido guardado exitosamente”)

**fin procedimiento**

**procedimiento** jugar

**variables**

tmpstr,opción: cadena

x,y,cont\_turn,error: entero

salir\_partida,validar\_turno,disp: booleano

**inicio**

limpiar\_tablero(tab)

salir\_partida←falso

cont\_turn←0

mientras (salir\_partida=falso) hacer

dibujar\_tablero(tab)

cont\_turn←cont\_turn+1

cont\_turn←cont\_turn mod 2 \*\*(“mod” es una función en

pascal que calcula el resto de una

division)

si (cont\_turn=1) entonces

escribir (“Turno del jugador de nombre: ”, nom\_jug1)

sino

escribir (“Turno del jugador de nombre: ”, nom\_jug2)

fin si

escribir (“Ingrese el número de la columna en donde desee

colocar su ficha: ”)

leer (y)

validar\_turno←falso

mientras (validar\_turno=falso) hacer

mientras ((y<1) Y (y>7)) hacer

escribir (“Error. El número debe oscilar entre 1 y 7.

Ingrese nuevamente el valor”)

Leer (y)

fin mientras

disp←falso

x←6

mientras ((x>=1) Y (disp=falso)) hacer

si (tab[x,y]= ‘ ‘ ) entonces

disp←verdadero

si (cont\_turn=1) entonces

tab[x,y]← ‘A’

sino

tab[x,y]← ‘B’

fin si

sino

x←x-1

fin si

fin mientras

si (disp=falso) entonces

escribir (“Error. no puede colocar mas fichas en esta

columna. Ingrese nuevamente el número de la columna

(del 1 al 7).”)

leer (y)

fin si

validar\_turno←disp

fin mientras

si (encontrar\_ganador(x,y,tab)=verdadero) entonces

dibujar\_tablero(tab)

escribir (“Fin de la partida”)

tot\_jug←tot\_jug+1

si (cont\_turn=1)

escribir (“Jugador ganador: ”,nom\_jug1)

gan←gan+1

sino

escribir (“Jugador ganador: ”,nom\_jug2)

gan2←gan2+1

fin si

escribir (“Desea jugar otra partida? s/n”)

leer(opc)

mientras ((opc<> ‘s’) Y (opc<> ‘n’)) hacer

escribir (“Error. Ingrese la opción correcta”)

leer (opc)

fin mientras

si (opc= ‘n’) entonces

salir\_partida←verdadero

fin si

cont\_turn←0

limpiar\_tablero (tab)

sino si (tablero\_lleno(tab)=verdadero) entonces

dibujar\_tablero(tab)

escribir (“Tablero lleno. No se puede colocar mas fichas. Hay

un empate.”)

tot\_jug←tot\_jug+1

escribir (“Desea jugar otra partida? s/n”)

leer(opc)

mientras ((opc<> ‘s’) Y (opc<> ‘n’)) hacer

escribir (“Error. Ingrese la opción correcta”)

leer (opc)

fin mientras

si (opc= ‘n’) entonces

salir\_partida←verdadero

fin si

cont\_turn←0

limpiar\_tablero (tab)

fin si

fin mientras

**fin procedimiento**

**Inicio**

tot\_jug←0

gan←0

gan2←0

salir\_apl←falso

mientras (salir\_apl=falso) hacer

escribir (“Menú de opciones: ”)

escribir (“1- Actualizar perfil del jugador 1. ”)

escribir (“2- Actualizar perfil del jugador 2. ”)

escribir (“3- Jugar ”)

escribir (“4- Estadísticas Generales ”)

leer (opc)

mientras ((opc<> ‘1’) Y (opc<> ‘2’) Y (opc<> ‘3’) Y (opc<> ‘4’)) hacer

escribir (Error. Ingrese la opción correcta)

leer (opción)

fin mientras

caso (opc)

‘1’: perfil\_jug1

‘2’: perfil\_jug2

‘3’: jugar

‘4’: estadísticas

fin caso

escribir (“Desea salir de la aplicación? s/n”)

leer(opc)

mientras ((opc <> ‘s’) Y (opc <> ‘n’)) hacer

escribir (“Error. Ingrese la opción correcta”)

leer (opc)

fin mientras

si (opc= ‘s’) entonces

salir\_apl←verdadero

fin si

fin mientras

**Fin Algoritmo**