

# Certamen 1: Pregunta 1 de Desarrollo

ELO329, 1º Sem. 2022

**Usted debe programar el juego gato o tick tack toe con 16 casillas.** El juego consiste en 4 casillas horizontales por cuatro casillas verticales, gana el jugador que logre 4 en línea horizontal o vertical **OJO QUE NO SE PIDE REVISAR LAS DIAGONALES, POR SIMPLICIDAD.**

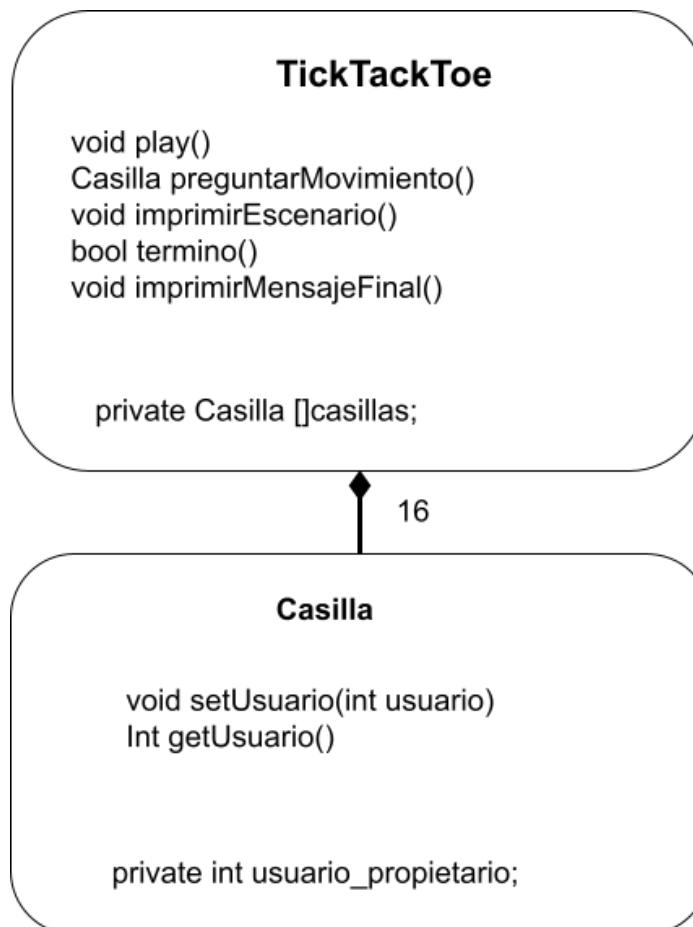
Debe imprimir por pantalla un 1 si el jugador uno eligió la casilla, un 2 si la escogió el jugador 2 o un 0 si está disponible.

El jugador 1 comienza eligiendo casilla, luego el dos y volverá a escoger el jugador uno y luego el dos hasta que se logre ganar o no existan más casillas para elegir.

Ejemplo de juego con una casilla elegida por el jugador 1 y otra por el jugador 2

```
0 0 0 0
1 0 2 0
0 0 0 0
0 0 0 0
```

A continuación, se mostrará un diagrama que se espera con el que implementen la solución.



Nota: por simplicidad suponer que el usuario utiliza correctamente el juego y solo elige casillas disponibles (con valor 0)

## El método main debe contener:

Main:

solo crea un ticktacktoe y llama a play()

## Para el método play debe implementar el siguiente pseudo código:

Play:

1. Primero se debe inicializar el arreglo bidimensional de Casillas
2. Mientras no se termine el juego: se debe validar si gano jugador 1 o gano jugador 2 o ya no quedan casillas.
  - a. preguntarCasillaAJugador: utilizando Scanner preguntar por jugada al jugador actual, se puede hacer dos preguntas una por casilla horizontal y vertical
  - b. marcarCasilla: Usando el arreglo de casillas, cambiar el poseedor de la casilla actual al usuario
  - c. ImprimirEscenario: Se debe recorrer con doble for el array bidimensional e imprimir el usuario de la casilla (o cero si no hay)
  - d. cambiarJugadorActual: Jugador actual cambia
3. Imprimir mensaje final: si gano algún jugador imprimir el jugador ganador, sino imprimir que fue empate.

Suba a AULA su respuesta en formato .zip, .rar o .tar si son varios archivos; en otro caso, un archivo .java.

Clase main 8 puntos:

```
public class Main {
    public static void main(String[] args) {
        TickTackToe4 t = new TickTackToe4();
        t.play();
    }
}
```

Clase Casilla 12 puntos:

```
public class Casilla {
    public Casilla(){user=0;}

    public int getUser() {
        return user;
    }
    public void setUser(int user) {
        this.user = user;
    }
    private int user;
}
```

Clase TickTackToe 30 puntos:

Criterio: Cada método vale 5 puntos (Constructor, termino, play, imprimirTablero, escoger Casiilla e imprimir mensaje final), notar que al ser 6 métodos se puede evaluar los mejores 5 siempre considerando termino como uno de ellos.

```

import java.util.Scanner;

public class TickTackToe4 {
    public TickTackToe4(){
        currentPlayer = 1;
        winner = 0;
        c = new Casilla[4][4];
        for (int i = 0; i < 4; i++)
            for (int j = 0; j < 4; j++)
                c[i][j] = new Casilla();
    }

    public void play(){
        while(!termino()){
            escogerCasilla();
            imprimirTablero();
            if(currentPlayer==1) currentPlayer=2; else currentPlayer=1;
        }
        imprimirMEnsajeFinal();
    }

    void imprimirTablero(){
        for(int row=0; row < 4; row++ ){
            for(int col=0; col<4; col++)
                System.out.print( c[row][col].getUser() );
            System.out.print("\n");
        }
    }

    private void escogerCasilla() {
        Scanner s = new Scanner(System.in);
        System.out.println("Jugador " + String.valueOf(currentPlayer));
        System.out.println("fila:");
        String row = s.nextLine();
        System.out.println("col:");
        String col = s.nextLine();
        c[Integer.parseInt(row)-1][Integer.parseInt(col)-1].setUser(currentPlayer);
    }

    public boolean termino(){
        for(int row=0; row < 4; row++ ) {
            int user_to_test = c[row][0].getUser();
            if(user_to_test == 0)break;
            if(c[row][0].getUser() == c[row][1].getUser() &&
                c[row][1].getUser() == c[row][2].getUser() &&
                c[row][2].getUser() == c[row][3].getUser()){
                winner = user_to_test;
                return true;
            }
        }
    }
}

```

```

for(int col=0; col < 4; col++ ){
    int user_to_test = c[0][col].getUser();
    if(user_to_test == 0)break;
    if(c[0][col].getUser() == c[1][col].getUser() &&
        c[1][col].getUser() == c[2][col].getUser() &&
        c[2][col].getUser() == c[3][col].getUser()){
        winner = user_to_test;
        return true;
    }
}

int totalCasillasUsed = 0;
for(int row=0; row < 4; row++ )
    for(int col=0; col<4; col++)
        if(c[row][col].getUser()!=0)totalCasillasUsed++;
if(totalCasillasUsed==16) return true;

return false;
}
void imprimirMensajeFinal(){
    if(winner == 1){
        System.out.println("Felicidades jugador 1");
    }else if(winner==2){
        System.out.println("Felicidades jugador 2");
    }
    else{System.out.println("Empate");}
}

private Casilla[][] c;
private int currentPlayer;
private int winner;
}

```