

Triqui

Edwin Montoya

Version 1.0

Definición

- **El juego consiste en un tablero de 3x3, en el cual de manera alternada, cada jugador coloca un símbolo (“X” ó “O”) y puede haber un ganador si logra colocar tres símbolos en línea (horizontal, vertical o diagonal) o puede no haber ganador.**
- **Un usuario puede tener más de un juego al tiempo, y puede quedarse jugando varias partidas con el mismo jugador hasta que alguno de los 2 lo determinen.**

Versión Standalone

```
public interface ITriquiGame {
    /*
     * Inicia un juego
     */
    public void Start();

    /*
     * Realiza una jugada y alterna el turno.
     * retorna verdadero si la jugada es válida
     * retorna falso si la jugada es inválida
     *
     * Esta son las posiciones que se utilizarán.
     *
     * [1] [2] [3]
     * [4] [5] [6]
     * [7] [8] [9]
     */
    public boolean Play(int pos);

    /*
     * Devuelve el simbolo del jugador actual (X o O)
     */
    public String Player();

    /*
     * Devuelve una versión texto del tablero del triqui
     */
    public String Board();

    /*
     * verifica si hubo ganador, retorna el Simbolo del ganador, o "no hubo"
     */
    public String TestWinner();
}
```

TriquiPlayer

```
public class TriquiPlayer {

    ITriquiGame triqui = null;

    public TriquiPlayer() {
        triqui = new TriquiGame();
    }

    /*
     * Lee un string del teclado
     */
    private String keyboard() {
        String cadena = null;
        InputStreamReader isr = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(isr);
        try {
            cadena = br.readLine();
        } catch (IOException ex) {
            Logger.getLogger(TriquiPlayer.class.getName()).log(Level.SEVERE, null, ex);
        }
        return cadena;
    }
}
```

```
private void play() {  
    triqui.Start();  
    String winner = "N";  
    int contplay = 0;  
    boolean valid = false;  
  
    while (winner.equals("N") && contplay < 9) {  
  
        String p = null;  
        String b = null;  
  
        p = triqui.Player();  
        b = triqui.Board();  
  
        System.out.println("\nTURNO: " + p);  
        System.out.println(b);  
        do {  
            System.out.print("Posicion = ");  
            String pos = keyboard();  
  
            valid = triqui.Play(Integer.parseInt(pos)-1);  
            if (!valid)  
                System.out.println(">>> Jugada invalida");  
        } while (!valid);  
        winner = triqui.TestWinner();  
        contplay++;  
    }  
    System.out.println(triqui.Board());  
    System.out.println("Ganador: " + winner);  
}
```

```
public void run() {  
    String input = "y";  
    while (input.equals("y")) {  
        play();  
        System.out.print("Continuar? (y/n) = ");  
        input = keyboard();  
    }  
}
```

```
public static void main(String[] args) {  
    TriquiPlayer t = new TriquiPlayer();  
    t.run();  
}
```

```
}
```

Triqui con Sockets

- Protocolo:
 - Request: “**START**,”
 - Response: void
 - Request: “**PLAY**,pos”
 - Response: “True” o “False”
 - Request: “**PLAYER**,”
 - Response: “X” o “O”
 - Request: “**BOARD**,”
 - Response: “[X] [] [O]\n....[]”
 - Request: “**TESTWINNER**,”
 - Response: “X” o “O” o “no hubo”

TriquiPlayerSocket

- `send("START,");`
- `send("PLAYER,");`
- `player = recv();`
- `send("PLAY,1");`
- `valid = recv();`
- `send("BOARD,");`
- `board= recv();`
- `send("TESTWINNER,");`
- `win= recv();`