

Universidad Rafael Landívar  
Facultad de Ingeniería.  
Ingeniería en informática en sistemas.  
Introducción de la programación (Práctica) - Sección: 04



## **“PROYECTO NO. 2 CONNECT FOUR”**

Estudiante: Jose Rodrigo Valle Riveiro.  
Carné: 1149123.

Guatemala, 28 de abril de 2023

Contenido

I. INTRODUCCIÓN ..... 3

II. ANÁLISIS..... 4

    a. ENTRADAS:..... 4

    b. SALIDAS:..... 5

III. DISEÑO ..... 9

IV. CONCLUSIONES ..... 10

V. RECOMENDACIONES ..... 11

VI. REFERENCIAS ..... 12

VII. ANEXOS..... 13

## I. INTRODUCCIÓN

*Connect Four* es un juego de mesa para dos jugadores. Consiste en un tablero de seis filas y siete columnas en el que los dos jugadores se turnan para ingresar fichas con el objetivo de alinear cuatro fichas del mismo tipo ya sea de manera horizontal, vertical o diagonal.

El siguiente proyecto tiene como propósito replicar en una aplicación de consolas en el lenguaje C# *Connect Four* Otorgando al usuario las siguientes opciones:

- Juego para dos jugadores.
- Juego para un jugador (Contra la maquina).
- Un registro de las ultimas diez partidas.

El siguiente informe contiene una explicación detallada del funcionamiento del programa diseñado para el cliente, así como su documentación tal como los diagramas de flujo elaborados para planificar el programa e instructivos para el operador del programa.

## **II. ANÁLISIS**

### **a. ENTRADAS: MENU PRINCIPAL:**

- Opción 1: Jugar contra la computadora.
- Opción 2: Jugar contra jugador.
- Opción 3: Ver el registro.
- Opción 4: Salir.

### **JUGAR CONTRA LA COMPUTADORA**

- Ingresar el nombre del jugador uno.
- Ingresar el número de columna para ingresar la ficha.
- Revancha (si o no).
- Regresar al menú principal (si o no).

### **JUGAR CONTRA JUGADOR**

- Ingresar el nombre del jugador uno.
- Ingresar el nombre del jugador dos.
- Ingresar el número de columna para ingresar la ficha (jugador 1).
- Ingresar el número de columna para ingresar la ficha (jugador 2).
- Revancha (si o no).
- Regresar al menú principal (si o no).

### **VER EL REGISTRO**

- Regresar al menú principal (si o no).

**b. SALIDAS:**  
**MENU PRINCIPAL:**

- Menú principal: (opción 1, 2, 3 y 4).

**JUGAR CONTRA LA COMPUTADORA**

- Mensaje para solicitar el nombre del jugador.
- Tablero de juego actualizado.
- Condición de victoria.
- Solicitud para revancha y regresar al menú.

**JUGAR CONTRA JUGADOR**

- Mensaje para solicitar el nombre del jugador uno.
- Mensaje para solicitar el nombre del jugador dos.
- Tablero de juego actualizado.
- Condición de victoria.
- Solicitud para revancha y regresar al menú.

**VER EL REGISTRO:**

- Nombre de los últimos diez jugadores.
- Turnos necesarios para ganar de los últimos diez jugadores.
- Tiempo necesario para ganar de los últimos diez jugadores.
- Solicitud para regresar al menú.

## PROCESOS:

### MENU PRINCIPAL

- Existen 4 posibles opciones, 1) Jugar contra la máquina. 2) Para dos jugadores 3) Ver el registro 4) Salir.
- 1) Jugar contra la máquina. Ejecutará el juego contra la maquina y se le pedirá la revancha al usuario hasta que este seleccione que no. Posteriormente se almacenará el nombre del jugador ganador, el número de turnos jugados y el tiempo transcurrido para almacenarlo en el récord.
- 2) Jugar contra otro jugador. Ejecutará el juego contra la maquina y se le pedirá la revancha al usuario hasta que este seleccione que no. Posteriormente se almacenará el nombre del jugador ganador, el número de turnos jugados y el tiempo transcurrido para almacenarlo en el récord.
- 3) Ver registro. Se mostrará en pantalla el registro de los últimos 10 jugadores. Incluye: Nombre del jugador, número de turnos y tiempo transcurrido.
- 4) Salir. Solicitara una ultima vez al usuario si quiere ir al menú principal y se este decide que no se cerrara el programa.

### MENU 1

- Solicita el nombre de los dos jugadores verificando que no se llamen "computadora".

### MENU 2

- Solicita el nombre del jugador verificando que no se llamen "computadora"

### RANKING

- Almacena el nombre del jugador ganador para almacenarlo en un vector con los últimos 10 jugadores.

### RANKINGTURNOS

- Almacena el número de turnos jugados de la última partida para almacenarlo en un vector con las ultimas 10 partidas.

### RANKINGTIEMPOS

- Almacena el tiempo transcurrido de la última partida para almacenarlo en un vector con las ultimas 10 partidas.

### ESCRIBIR RANKING

- Grafica el registro.

## ESCRIBIR TABLERO

- Grafica el tablero de juego con los valores actualizados

## TURNO 1

- Permite ingresar una ficha para el jugador 1, haciendo las verificaciones necesarias.

## TURNO 2

- Permite ingresar una ficha para el jugador 2, haciendo las verificaciones necesarias.

## TURNO MAQUINA

- Permite ingresar una ficha para la computadora, haciendo las verificaciones necesarias.

## VICTORIA 1

- Verifica la condición de victoria para el jugador 1 y devuelve un valor booleano.

## VICTORIA 2

- Verifica la condición de victoria para el jugador 2 (o la computadora) y devuelve un valor booleano.

## ESTALLENO

- Verifica si el tablero esta lleno y devuelve un valor booleano.

## TABLERO

- Inicia el juego para dos jugadores.

## TABLERO MAQUINA

- Inicia el juego para un jugador.

## CRONOMETRO CERO

- Inicia el cronometro.

## CRONOMETRO F

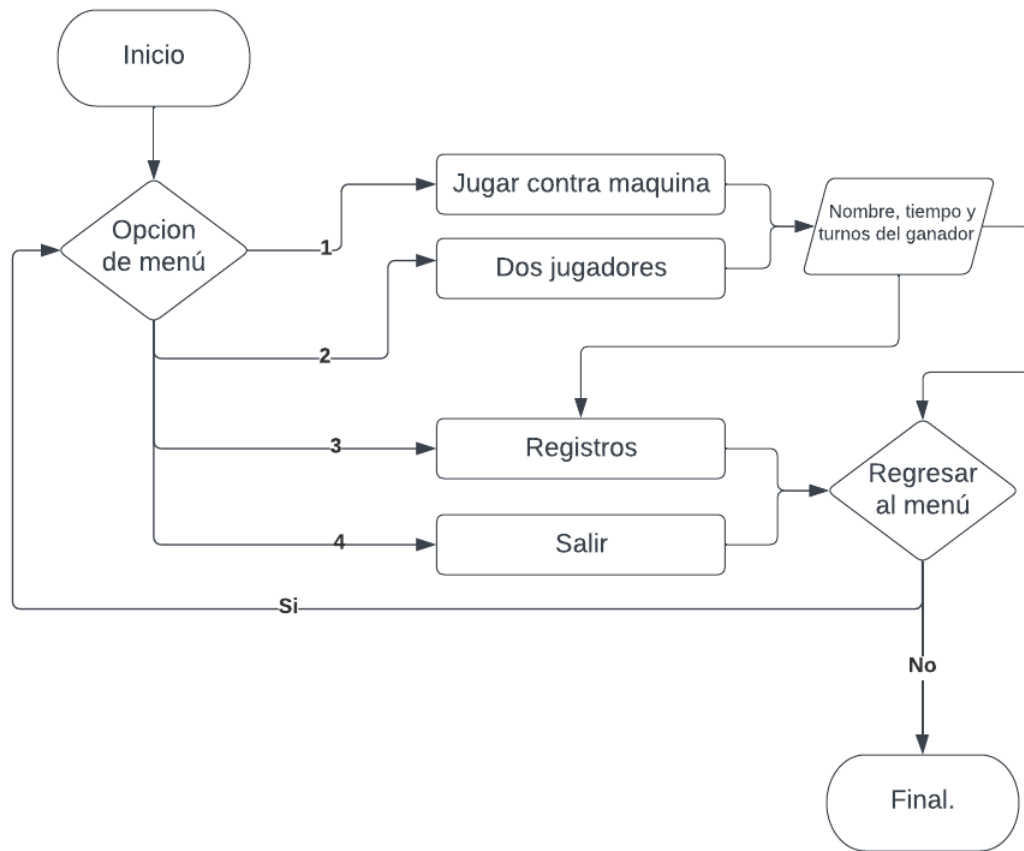
- Finaliza el cronometro, devuelve un string con el tiempo.

**RESTRICCIONES:**

- El juego esta preparado para que el usuario nunca ingrese valores inválidos y que no ingrese fichas en columnas llenas.



### III. DISEÑO



#### **IV. CONCLUSIONES**

Se realizó un programa que emula el famoso juego de mesa *Connect Four* empleando la programación orientada a objetos (POO), arreglos, ciclos y todos los conceptos vistos en clase.

El programa funciona mediante el uso de arreglos, uno para emular el tablero de juego y otros para almacenar el registro de las ultimas diez partidas. Dicho programa esta preparado para jugarse por un jugador o dos jugadores, así mismo tiene código defensivo para prevenir errores.

Durante la elaboración del programa se determinó que el uso de métodos facilita la organización del código, así mismo se determinó que los parámetros facilitan la colaboración entre varios métodos.

## **V. RECOMENDACIONES**

El programa esta preparado para evitar errores cometidos por el usuario, sin embargo, siempre es necesario tener en cuenta de que hay que ingresar datos coherentes para el correcto funcionamiento del programa.

El programa permite cerrarlo después de terminar cada opción por motivos de conveniencia, sin embargo, hay que tener cuidado de no salir del programa por accidente.

## **VI. REFERENCIAS**

**System:** Elementos más básicos de C#

**System.Collections.Generic:** Centrados en interfaces

**System.Linq:** Centrados en matrices

**System.Text:** Manipulación de strings.

**System.Threading.Tasks:** Programación orientada a objetos.

**System.Diagnostics:** Se utilizo para el cronometro.

## **VII. ANEXOS**

Manual de usuario: <https://www.youtube.com/watch?v=CR5gEIJXQzc>