

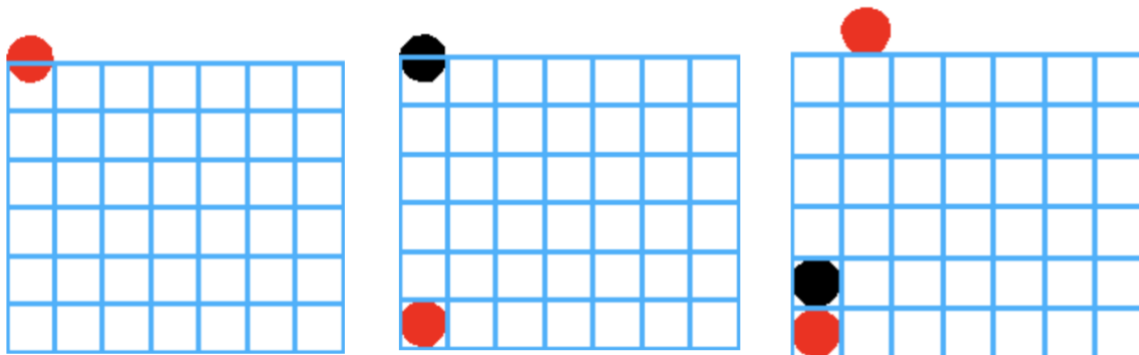


## PROYECTO PRÁCTICO NO. 02



### Descripción

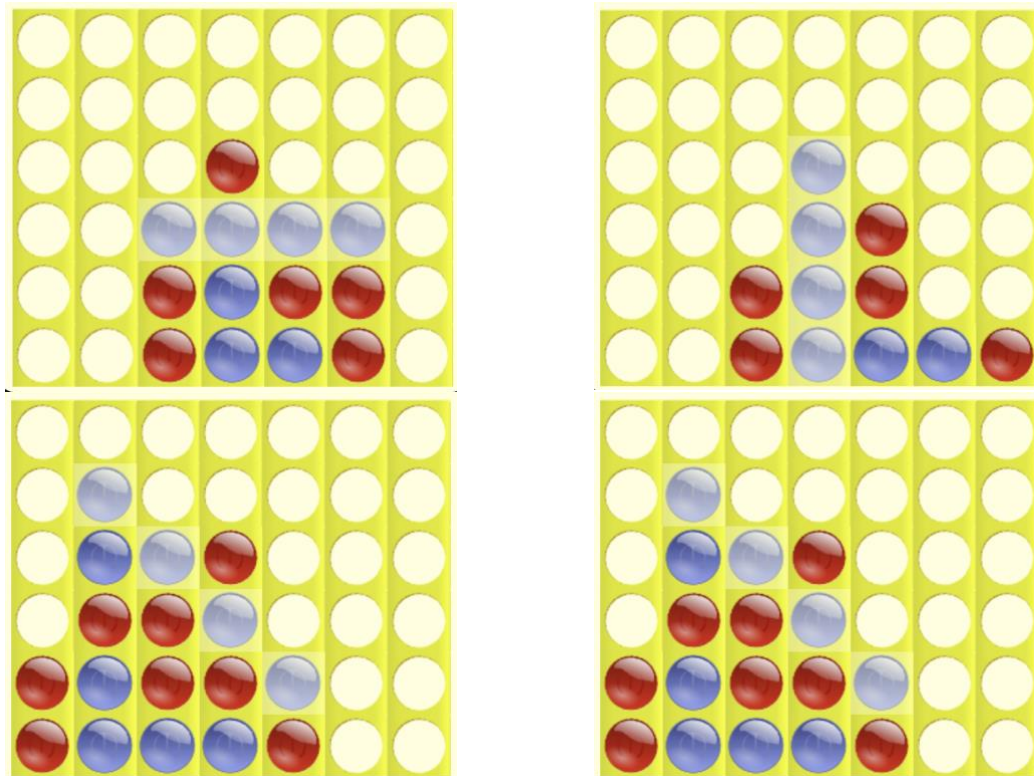
Connect Four es un juego para dos personas en donde cada jugador debe escoger un color y, por turnos, dejan caer una ficha en una cuadrícula de 7 columnas y 6 filas. El jugador escoge una columna y deja caer la ficha hasta la primera casilla disponible, es decir, la primera casilla que no tenga ficha aún o la casilla que esté hasta el fondo del tablero.



*Ilustración 1. Ejemplo de inserción de fichas en dos turnos.*



El jugador que primero logre conectar cuatro fichas de su color gana. Las fichas pueden estar conectadas verticalmente, horizontalmente o en diagonal.



*Ilustración 2. Ejemplo de maneras para ganar Connect Four.*

## Enunciado

El programa debe preguntar al usuario si jugarán dos personas o una persona contra la computadora. Se debe preguntar y almacenar en memoria el nombre de cada persona. Si se escogió jugar contra la computadora, el nombre del segundo jugador siempre será "COMPUTADORA". Si se escoge dos jugadores humanos, debe validar que ninguno de los jugadores se llame "COMPUTADORA" para evitar confusiones.

El juego comienza en el turno del jugador 1, luego el jugador 2 (que podría ser la computadora) y así sucesivamente hasta que gane uno de los dos jugadores o hasta que ninguno de los dos jugadores tenga posibilidades de ganar. Si uno de los dos jugadores gana, el programa debe mostrar la secuencia de fichas que le ayudaron a ganar el juego. Ningún jugador puede saltarse un turno.



Al terminar el juego, el programa no debe cerrarse a menos que el usuario lo indique. En su lugar, deberá preguntarse al usuario si desea iniciar una nueva partida (contra la computadora u otra persona) o cualquiera de las opciones disponibles del programa.

El programa debe tener una opción para mostrar los últimos 10 jugadores en ganar una partida. Esta opción debe mostrar el tiempo (en minutos y segundos) y la cantidad de turnos que le tomó a cada jugador ganar. Si una partida termina en empate, la información de esa partida no debe almacenarse.

Si el usuario escoge jugar contra la computadora, no es necesario programar un jugador "inteligente" con la capacidad de seguir una estrategia. El jugador "COMPUTADORA" podrá insertar fichas aleatoriamente en cualquier columna disponible.

## Entregables

- **Repositorio en GitHub con el código del programa:**
  - El programa debe codificarse en el lenguaje C#, y es requerido que el estudiante utilice Programación Orientada a Objetos (OOP), arreglos, ciclos y cualquier concepto visto durante la clase.
  - Además, el estudiante deberá realizar una interfaz intuitiva para el usuario.
- **Documentación externa:**
  1. Carátula
  2. Introducción
  3. Análisis
    - a. Entradas
    - b. Salidas
    - c. Procesos
    - d. Restricciones
  4. Diseño
    - a. Diagrama de Flujo
  5. Conclusiones
  6. Recomendaciones
  7. Referencias
    - a. Librerías utilizadas y su utilización
  8. Anexos
    - a. Manual de Usuario



### Aspectos para evaluar

- Validación de errores.
- Adecuada aplicación de los conocimientos.
- Calidad de la documentación: ortografía, orden, limpieza y que esté completa.
- Calidad de la solución propuesta: que solucione el problema (que haga lo que requiere el sistema de reserva) en forma eficaz.
- Funcionalidad del programa: debe cumplir a cabalidad con todos los requerimientos.
- **Evidencia de la creación del programa y dominio de los conceptos utilizados.**
- Creatividad.

### Ponderación

Documentación	5
Análisis	5
Diseño	5
Codificación	85
<b>Total</b>	<b>100</b>

### Consideraciones

- Se debe realizar una solución en modo Consola o Windows Forms, en lenguaje de programación C#.
- Toda solución presentada debe compilar correctamente para poder tener derecho a revisión.
- La utilización de código de terceros debe estar completamente documentada, referenciada y justificada. debe demostrarse el dominio completo de lo implementado.
- **Se podrá demandar que en la calificación presencial del proyecto se realicen cambios de funcionalidad.**