Universidad Rafael Landívar

Facultad de Ingeniería.

Ingeniería en Informática y Sistemas.

Curso de Introducción a la programación (práctica) - Sección: 04

Catedrático: Ingeniero Rene Daniel Mejía Alvarado

**PROYECTO PRÁCTICO NO. 02**

**CONNECT FOUR**

Estudiante: Héctor José Flores Pineda

Carné:1199923

Guatemala, 29 de abril de 2023

# **INTRODUCCIÓN**

Conecta 4 es un popular juego de mesa en el cual dos jugadores colocan fichas de colores en una rejilla vertical de 7 columnas y 6 filas, con la intención de alinear cuatro fichas del mismo color de manera vertical, horizontal o diagonal antes que su oponente lo haga.

En este proyecto se pretende crear un programa de consola que simule el juego Conectar 4, haciendo uso de la programación orientada a objetos, arreglos y matrices para almacenar la información del tablero, fichas, jugadores, ganadores y todo lo relacionado con la información del juego. Además, se documentarán las entradas, salidas, restricciones y procesos para garantizar una comprensión de la elaboración del programa.

El programa permitirá a dos jugadores jugar en una misma computadora y seguirá las reglas básicas del juego Conectar 4. Los jugadores podrán colocar una ficha en la columna que elijan, siempre y cuando la columna no esté llena. El juego termina cuando uno de los jugadores alinea cuatro fichas del mismo color de manera vertical, horizontal o diagonal, o cuando todas las casillas del tablero están llenas y no hay un ganador.

El objetivo de este proyecto es crear un programa completo y documentado que permita a los usuarios jugar Conectar 4 de manera digital y que incluya las características necesarias para una experiencia de juego satisfactoria, simulando a la experiencia en la vida real.

**ANÁLISIS**

1. **ENTRADAS**

* Opción: La opción que elegirá el usuario al navegar por los menús, siendo está representada con un número.
* Jugador: El nombre con el que el usuario desea ser identificado en la partida.
* Modo de juego: El usuario podrá decidir el modo de juego que desea jugar.
* Ficha de juego: El usuario elige cual de las dos fichas disponibles desea jugar.
* Primer turno: En el modo JugadorVsJugador, los usuarios eligen cuál de los dos jugadores iniciará con el primer turno.
* Columna: La columna que elige el jugador, en donde desea colocar su ficha.

1. **SALIDAS**

* Tablero de juego: El programa imprime el tablero de juego, donde se colocarán las fichas y se mostrara sus posiciones.
* Ganador: Al finalizar la partida, el programa muestra al jugador que ganó.
* Empate: Si el tablero esta lleno, el juego no puede continuar, por lo que, se muestra un “empate”.
* Nombre del jugador: El programa muestra el nombre del jugador, para indicar que es su turno.
* Ficha del jugador: El programa muestra la ficha del jugador, indicando con que ficha está jugando.
* Registro de puntuaciones: El programa muestra los últimos diez jugadores en ganar
* Turnos del jugador: Muestra los turnos que le tomaron al jugador para ganar.
* Tiempo de partida: Muestra el tiempo en que el jugador tardó en ganar.

1. **PROCESOS**

**Menú principal**

El programa imprime una “bienvenida” al usuario e imprime el siguiente menú

1. Nueva Partida
2. Tabla de jugadores
3. Salir del juego

El usuario seleccionará una opción, escribiendo el número de la opción.

La primera opción mostrará al usuario a un pequeño menú, con la opción de crear una partida en los diferentes modos de juego; la segunda opción mostrará una tabla con los últimos diez jugadores en ganar, con los turnos y el tiempo que le tomo a cada jugador en ganar una partida; la tercera opción cerrará el programa.

**Nueva partida**

El programa muestra un menú con las siguientes opciones

1. Vs Jugador
2. Vs Computadora

De la misma forma, el usuario seleccionará una opción , ingresando el número correspondiente.

**Vs Jugador**

El programa solicita al usuario, ingresar el nombre del jugador 1, asimismo, solicita ingresar el nombre del jugador 2.

Luego, el programa pregunta al usuario, quién tendrá el primer turno (jugador1 o jugador2), y dependiendo de su respuesta, le pregunta al usuario con el “primer turno”, con cual ficha jugará (o / x), asignando la ficha resultante al jugador del segundo turno.

Con esta información, el programa crea el tablero, que es una matriz de 6 filas y 7 columnas, e inicia con el turno del jugador con el “primer turno”, al cual se le solicita la columna donde quiere jugar su ficha, esto se repite con el turno del otro jugador; este proceso se realiza dentro de un bucle, el cual se romperá al encontrar un ganador, es decir, a un jugador que logró encadenar 4 fichas o cuando el tablero esté lleno, significando un empate.

Luego, el programa desplegará un pequeño menú con las siguientes opciones:

1. Volver al menú principal
2. Salir del juego

La primera opción, retornará al usuario al menú principal; y la segunda opción cerrará el programa.

**Vs Computadora**

El programa solicita al usuario el nombre del jugador 1 y la ficha que utilizará,

después, el programa le dará el primer turno al jugador 1, solicitándole la columna en donde sea jugar su ficha, luego, pasa al turno de la computadora, que elegirá una columna al azar dentro de los limites del tablero y colocará su ficha, este proceso se repite en bucle hasta que uno de los jugadores encadenar 4 fichas o cuando el tablero esté lleno, significando un empate.

Luego, el programa desplegará un pequeño menú con las siguientes opciones:

1. Volver al menú principal
2. Salir del juego

La primera opción, retornará al usuario al menú principal; y la segunda opción cerrará el programa.

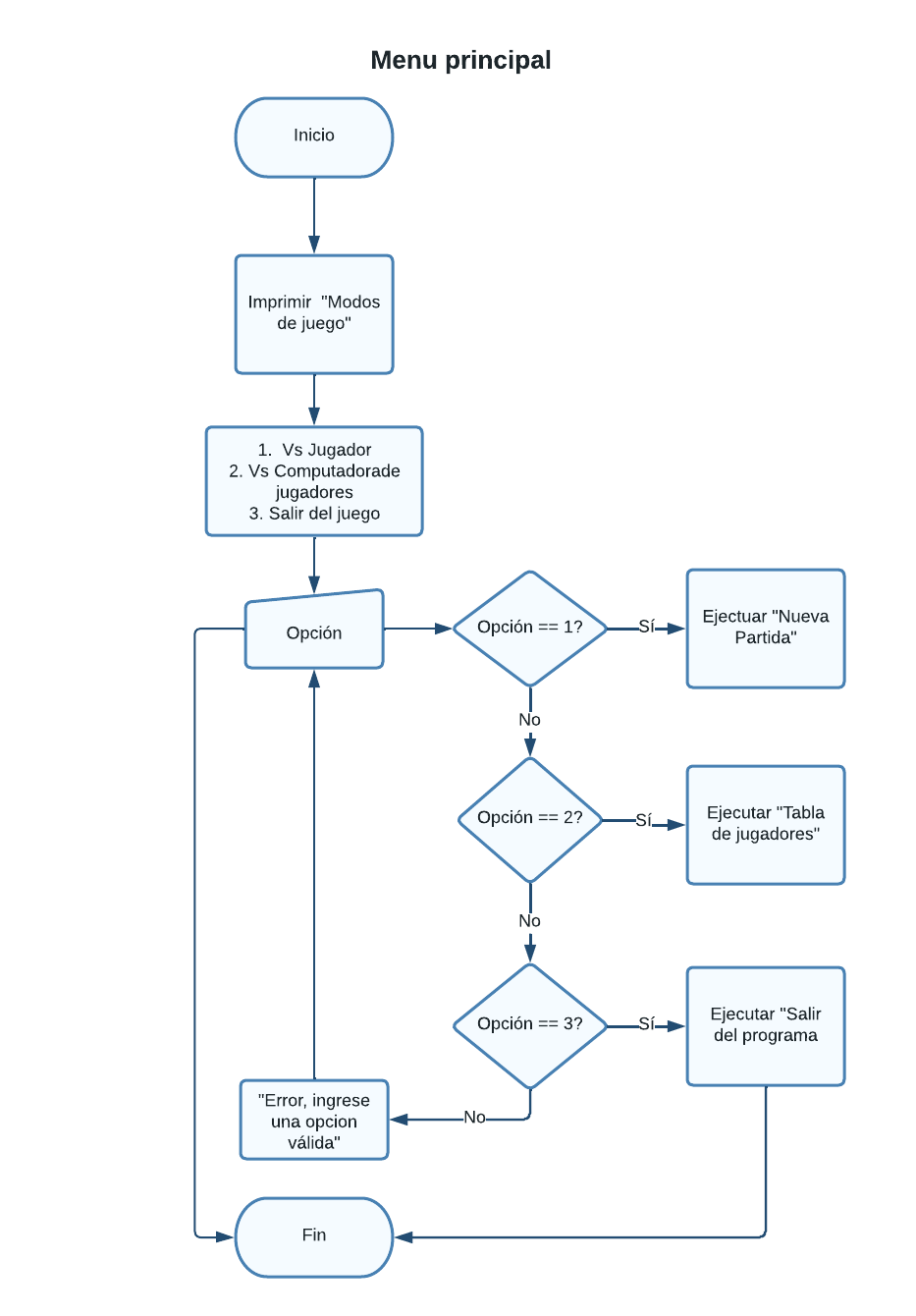
**Tabla de jugadores**

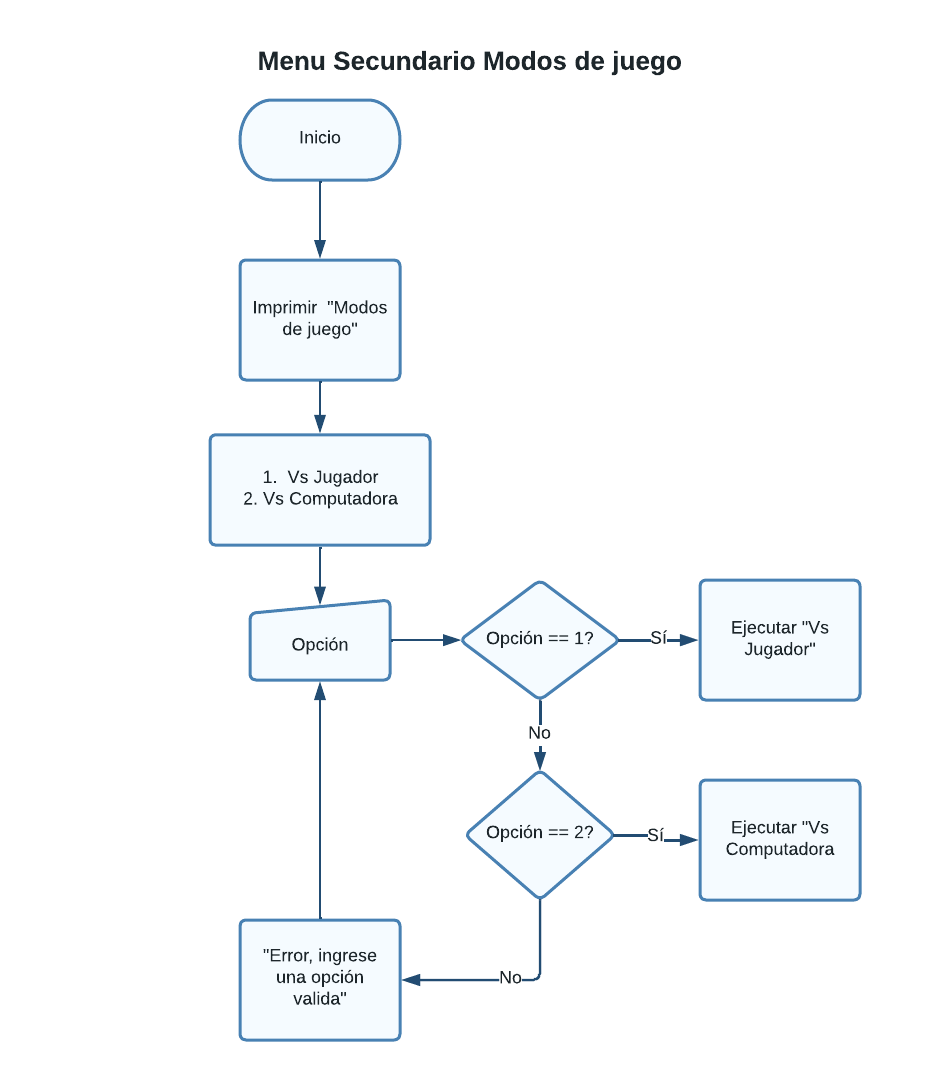
El programa muestra una tabla con tres encabezados que son “Jugadores”, “Tiempo” y “Turnos”, donde se mostrarán la información de los 10 ganadores más recientes en el modo vsJugador, imprimiendo el nombre del jugador ganador, el tiempo y los turnos que le tomó en ganar. Actualizando la tabla con la entrada de un nuevo ganador.

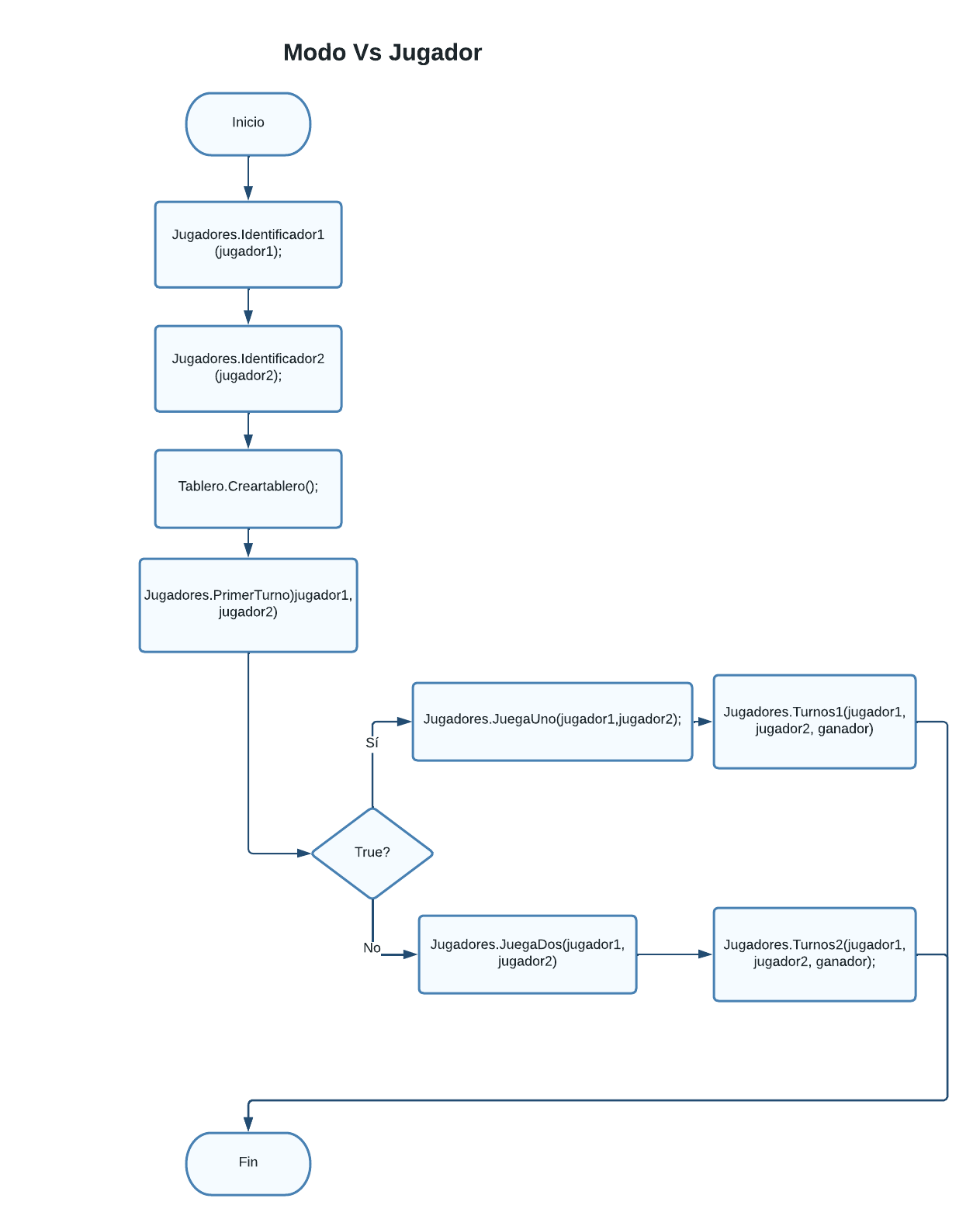
1. **RESTRICCIONES**

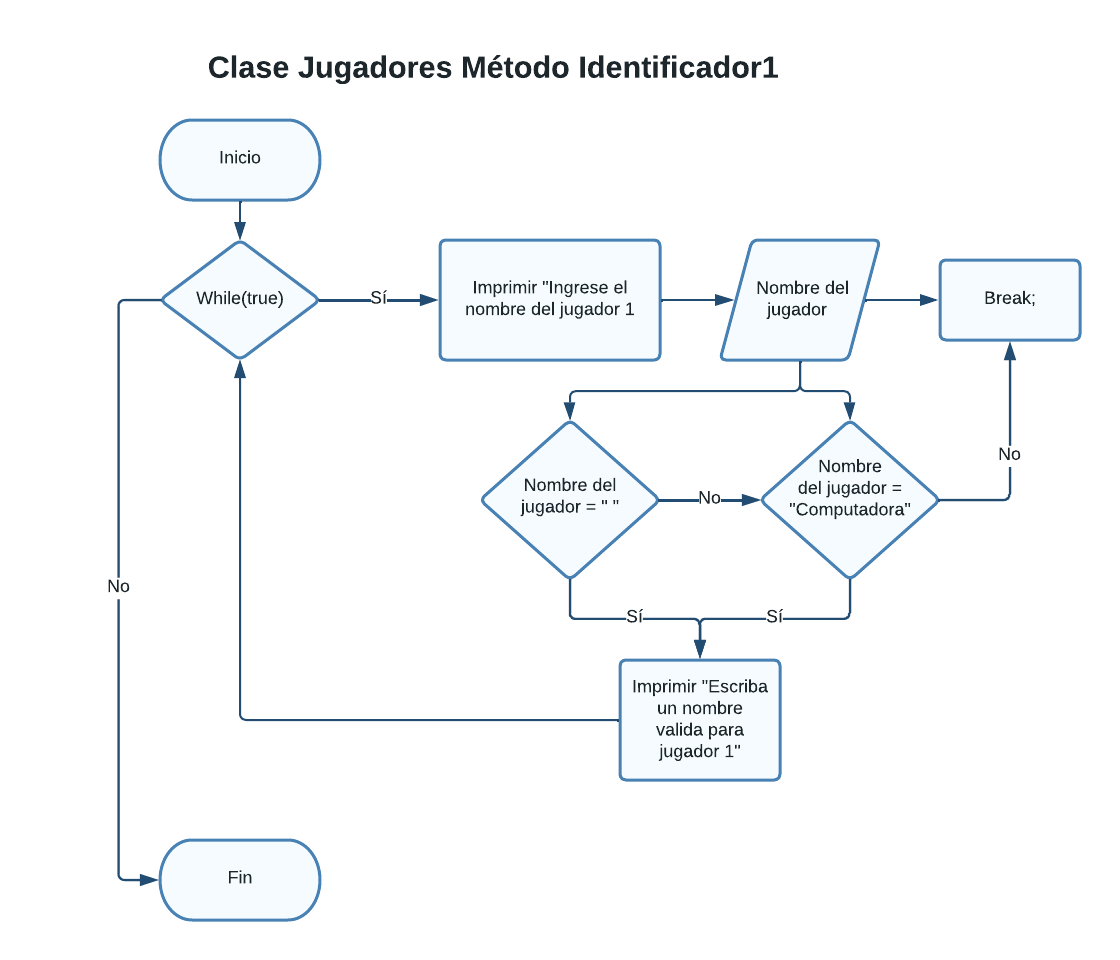
* Limitación de opciones de menú: Los menús dentro del programa solo aceptan las opciones mostradas en los menús; por lo que, ingresar un dato que no sea un número o un numero fuera de las opciones del menú.
* Limitación de nombre del jugador: El programa solo aceptará nombres válidos para jugadores, es decir, el jugador no podrá dejar el espacio vacío y tampoco podrá llamarse “Computadora”, para evitar confusiones, en tal caso surja un error, el programa informará sobre el mismo y solicitara nuevamente el nombre del jugador.
* Limitación de ficha del jugador: El programa solo aceptará las fichas proporcionadas por el mismo, el ingreso de otro símbolo extraño resultará en un error, el cual será notificado y se solicitará nuevamente la elección de ficha.
* Limitación de columna: Los jugadores podrán elegir la columna en donde depositar su ficha, siempre y cuando está columna se encuentre dentro de los límites del tablero y la columna no este llena de fichas, en tal caso se produzca un error, el programa solicitará nuevamente, al jugador, la columna donde desea jugar su ficha y su turno no se omitirá.
* Limitación de tabla de jugadores: Si la partida termina en empate, la información de la partida no se almacenará para mostrarse en la tabla de jugadores, al igual que si el modo de juego es Vs Computadora, no se guardará la información, debido a la sencillez de la IA de la Computadora, no amerita almacenar esa información.

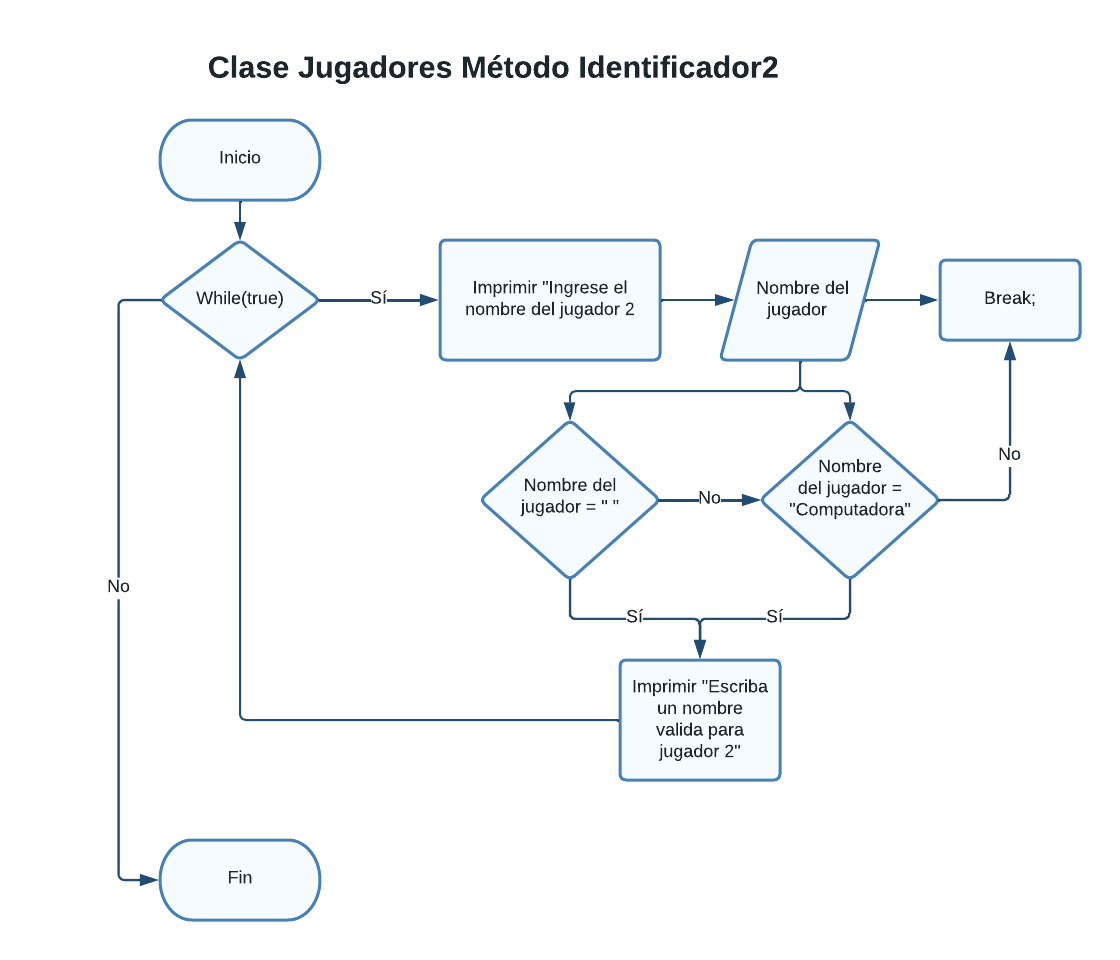
**DISEÑO**

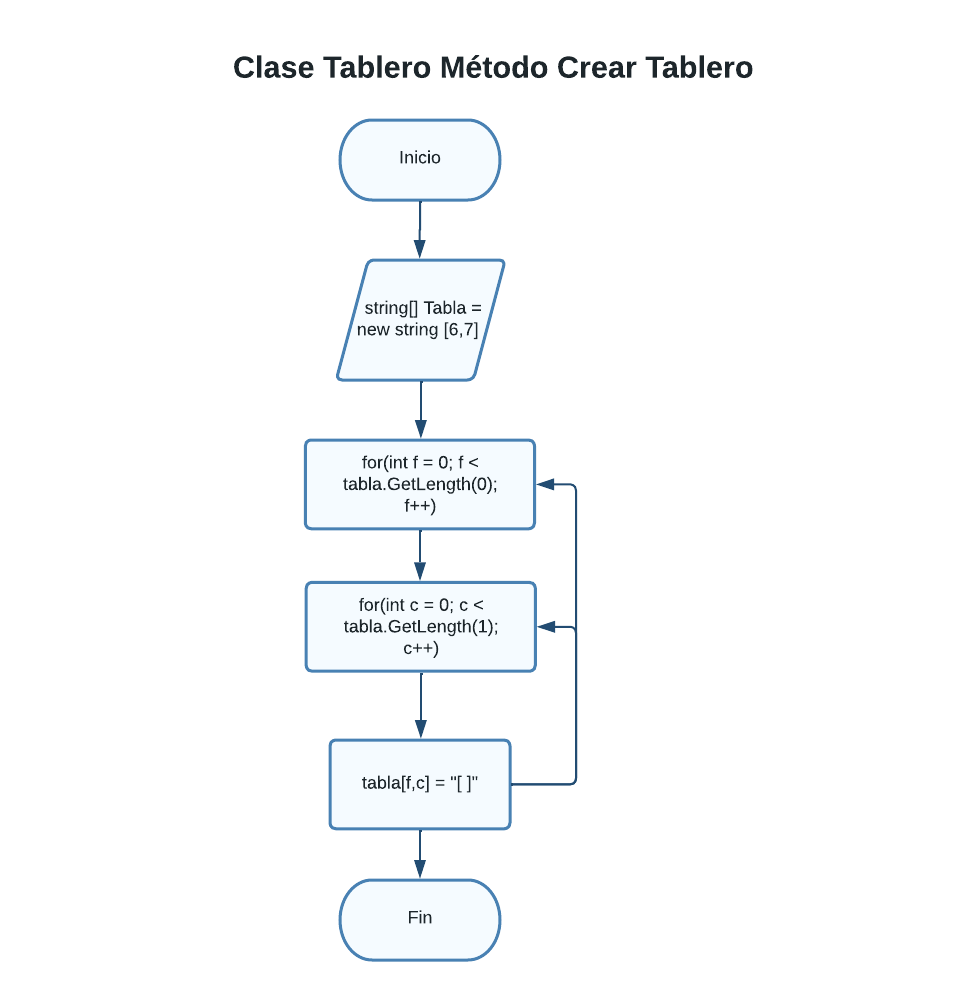


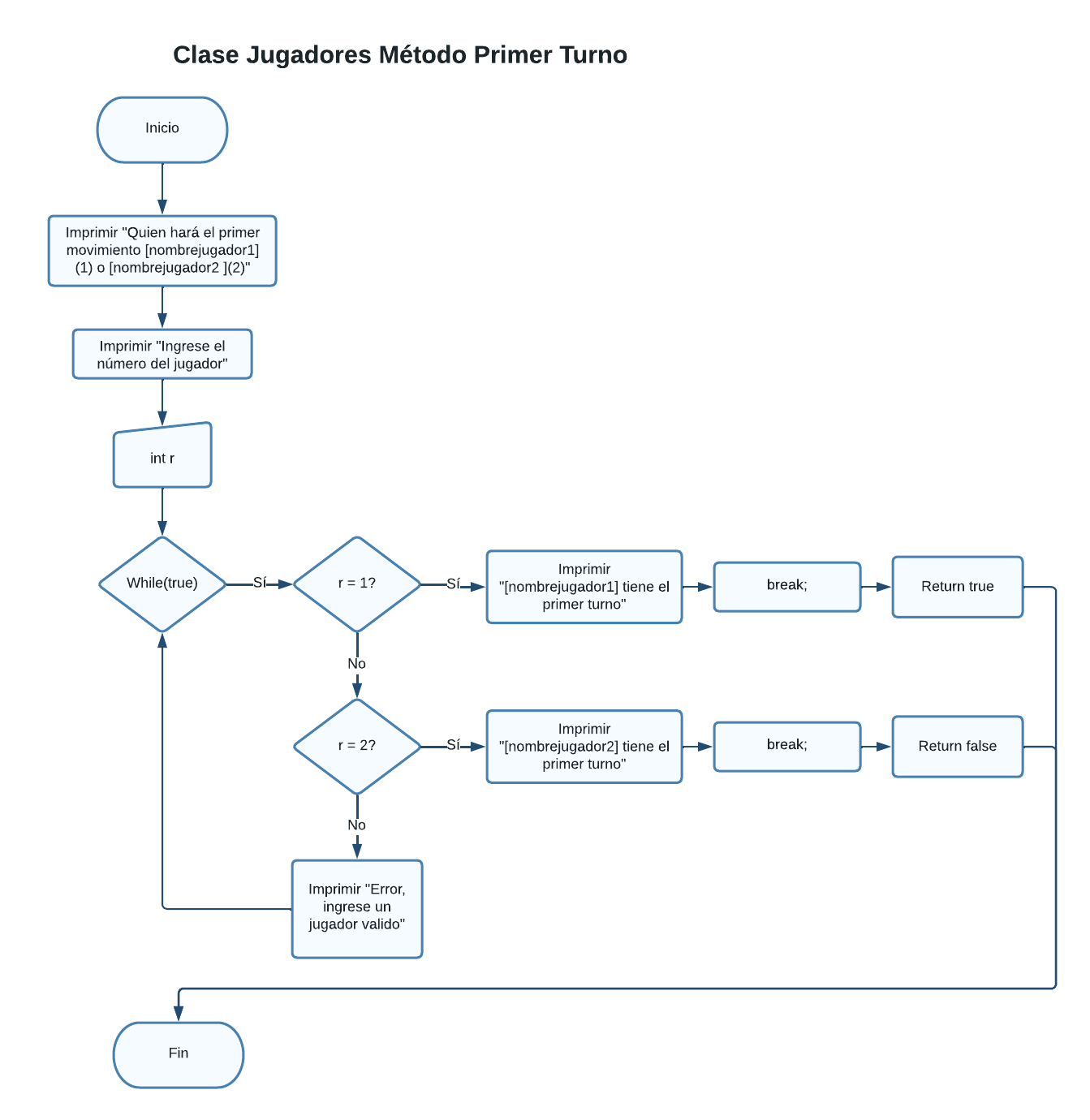


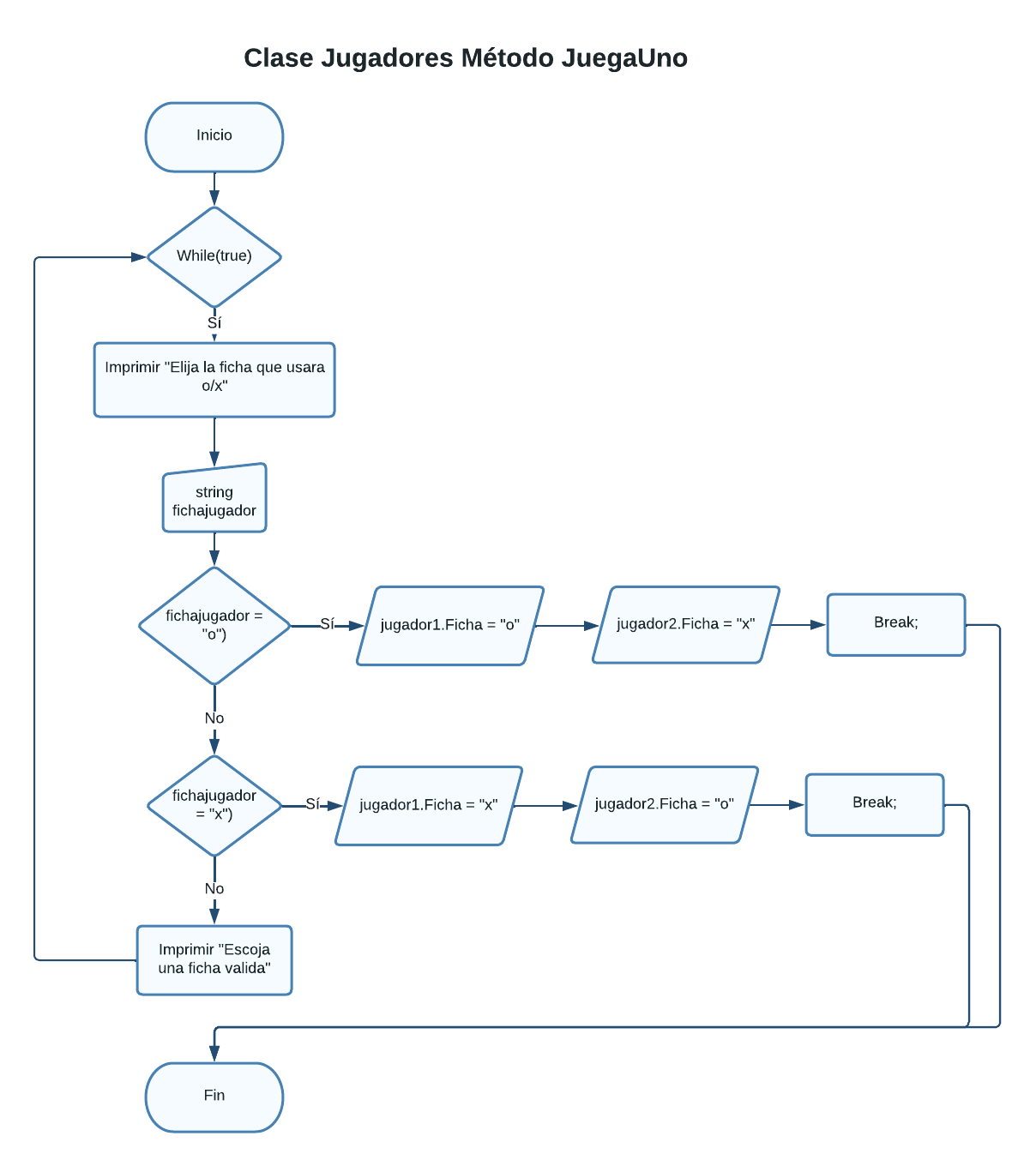


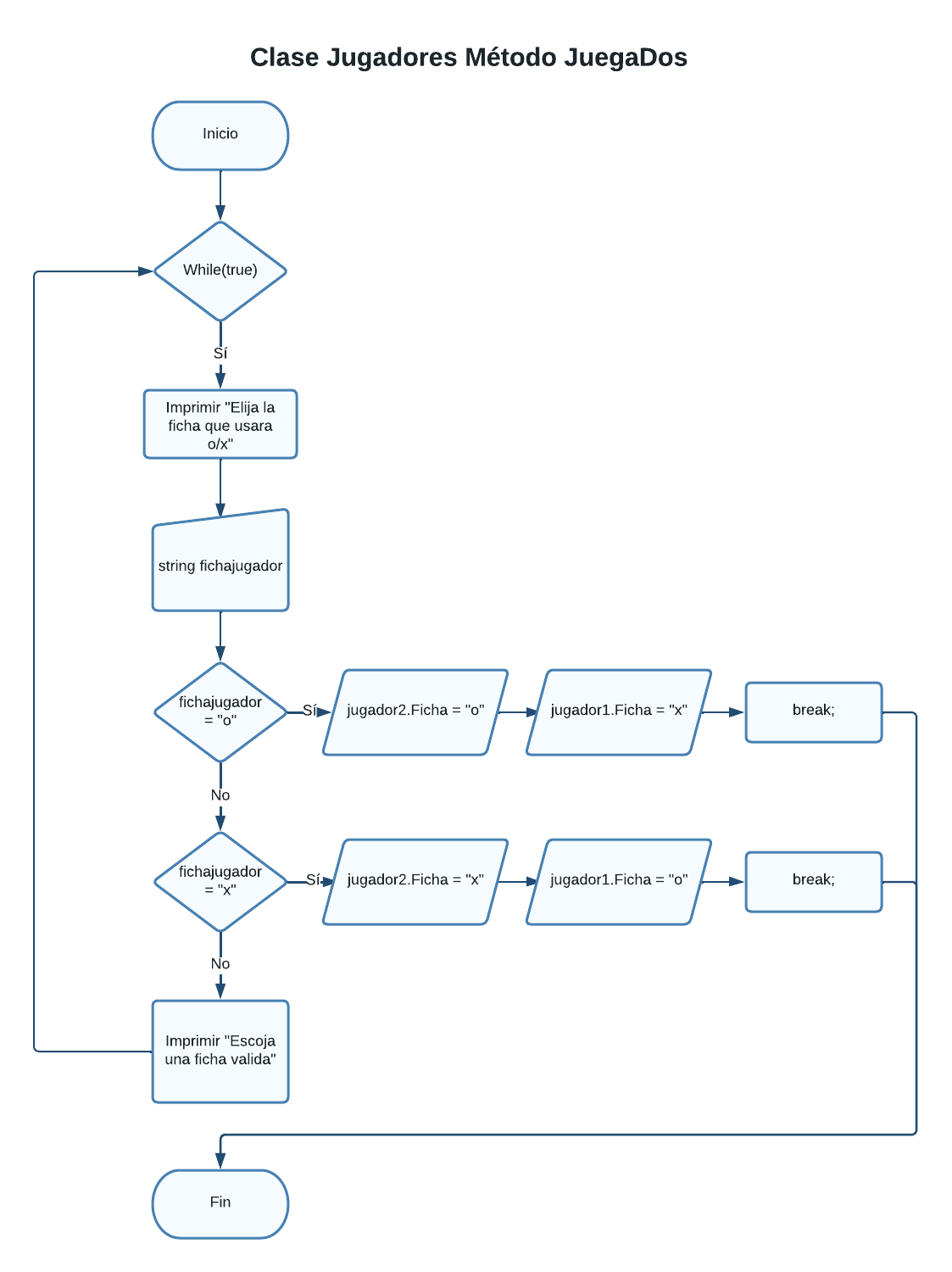


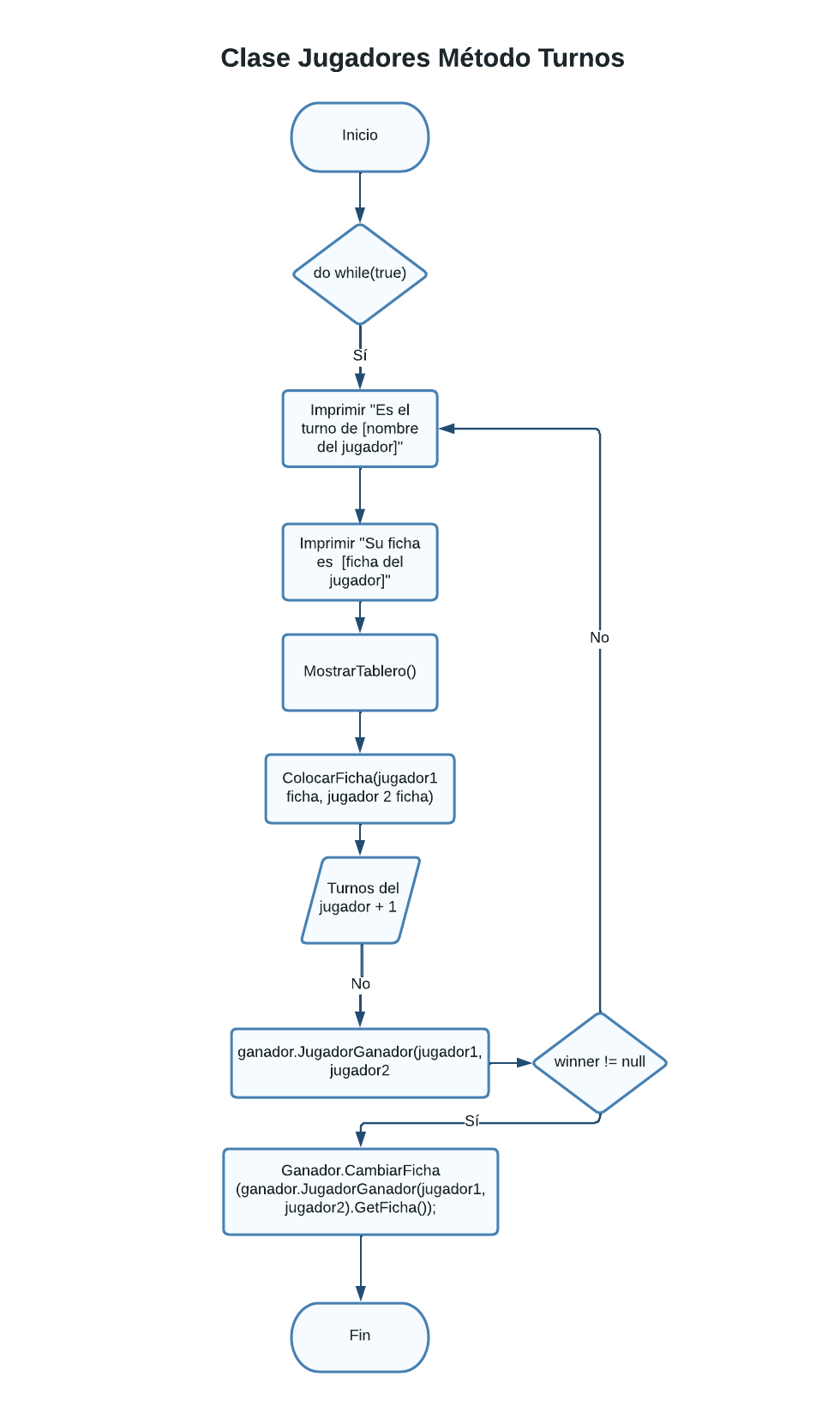


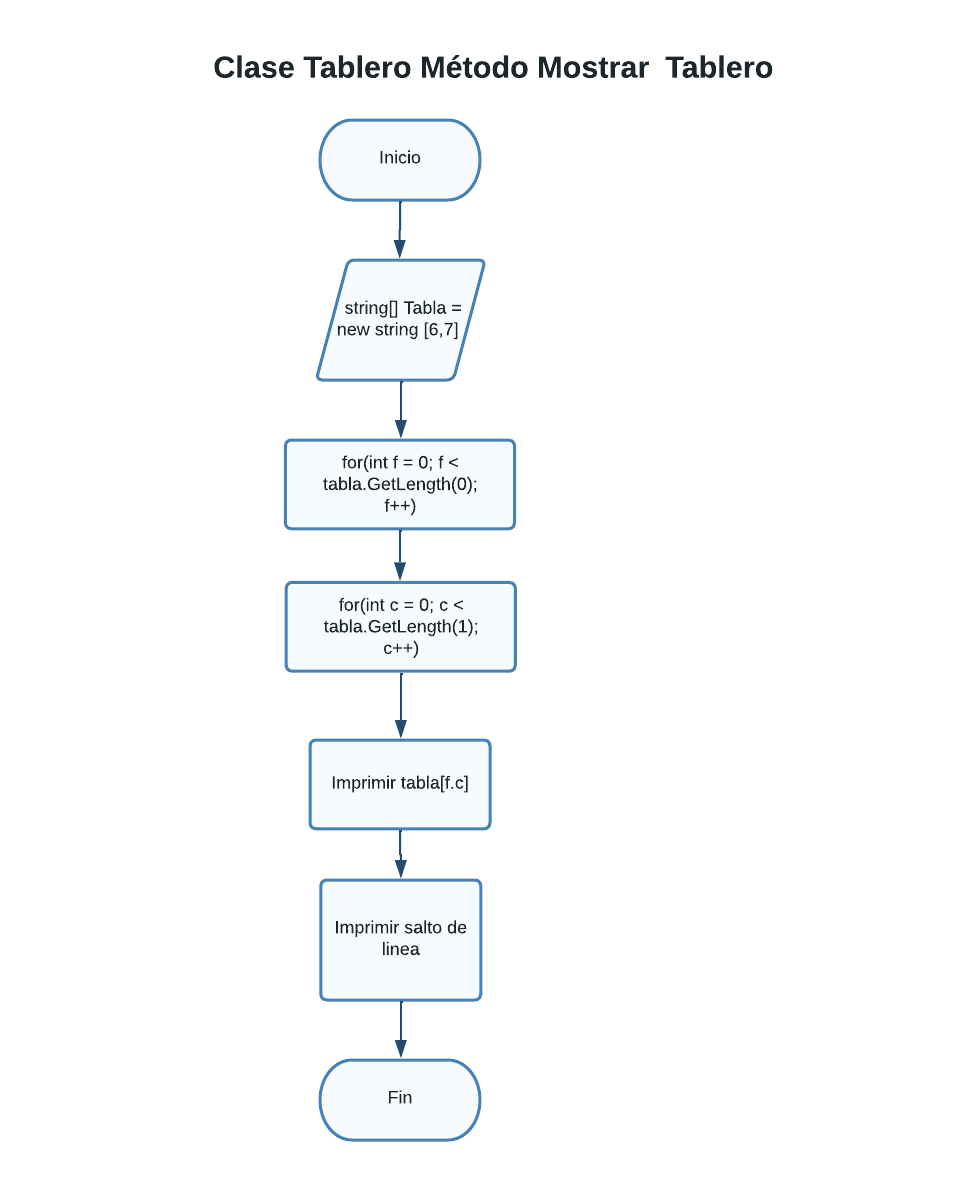


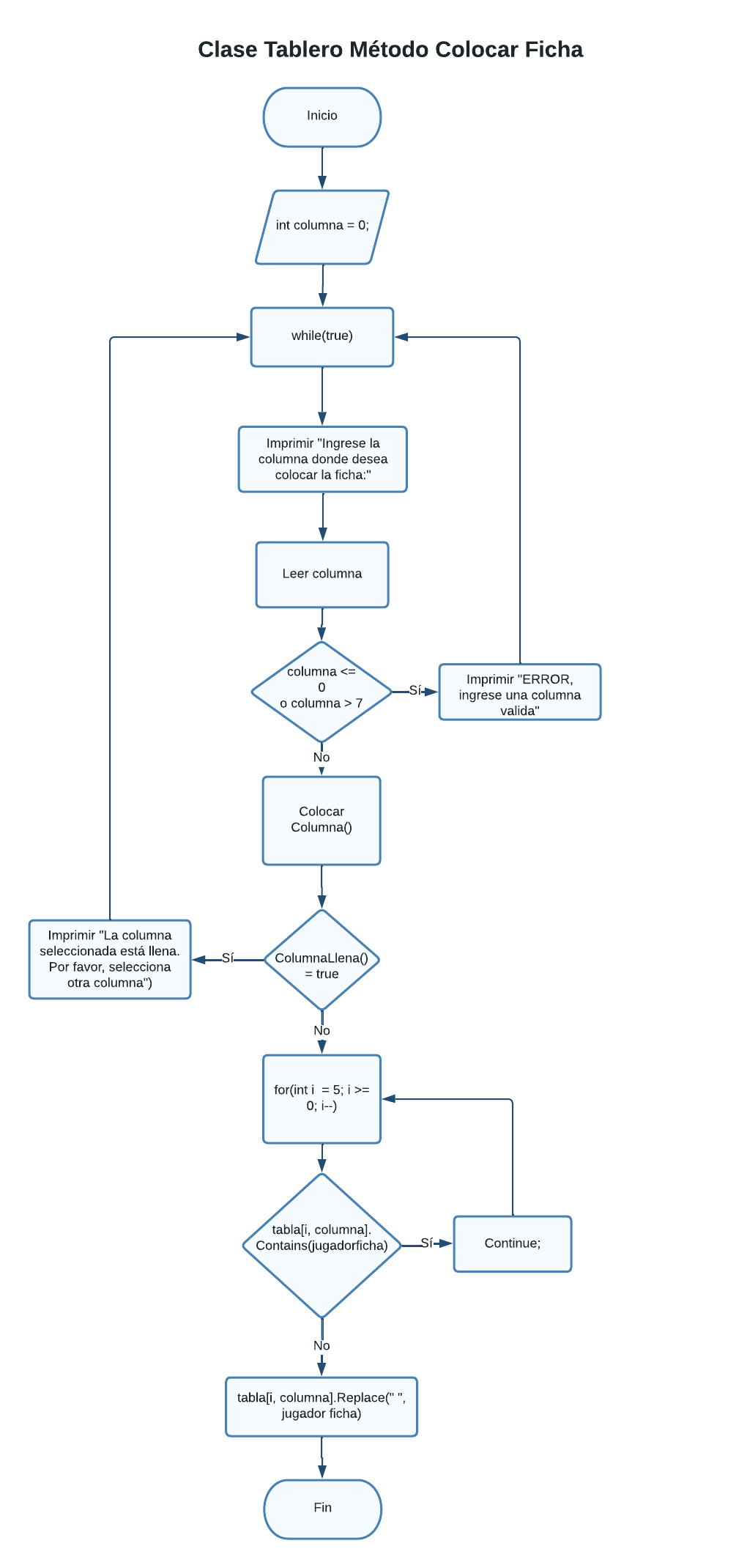


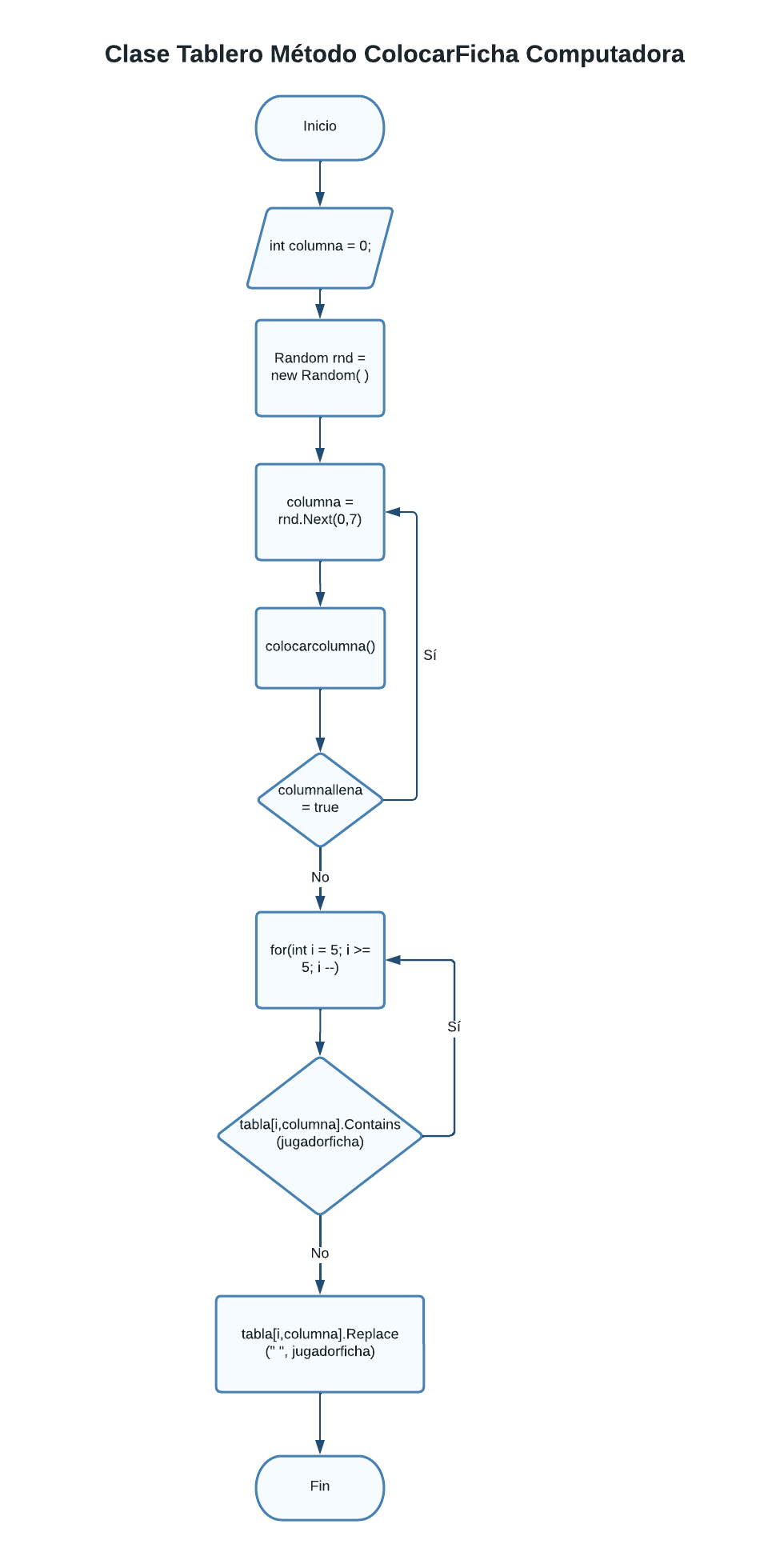


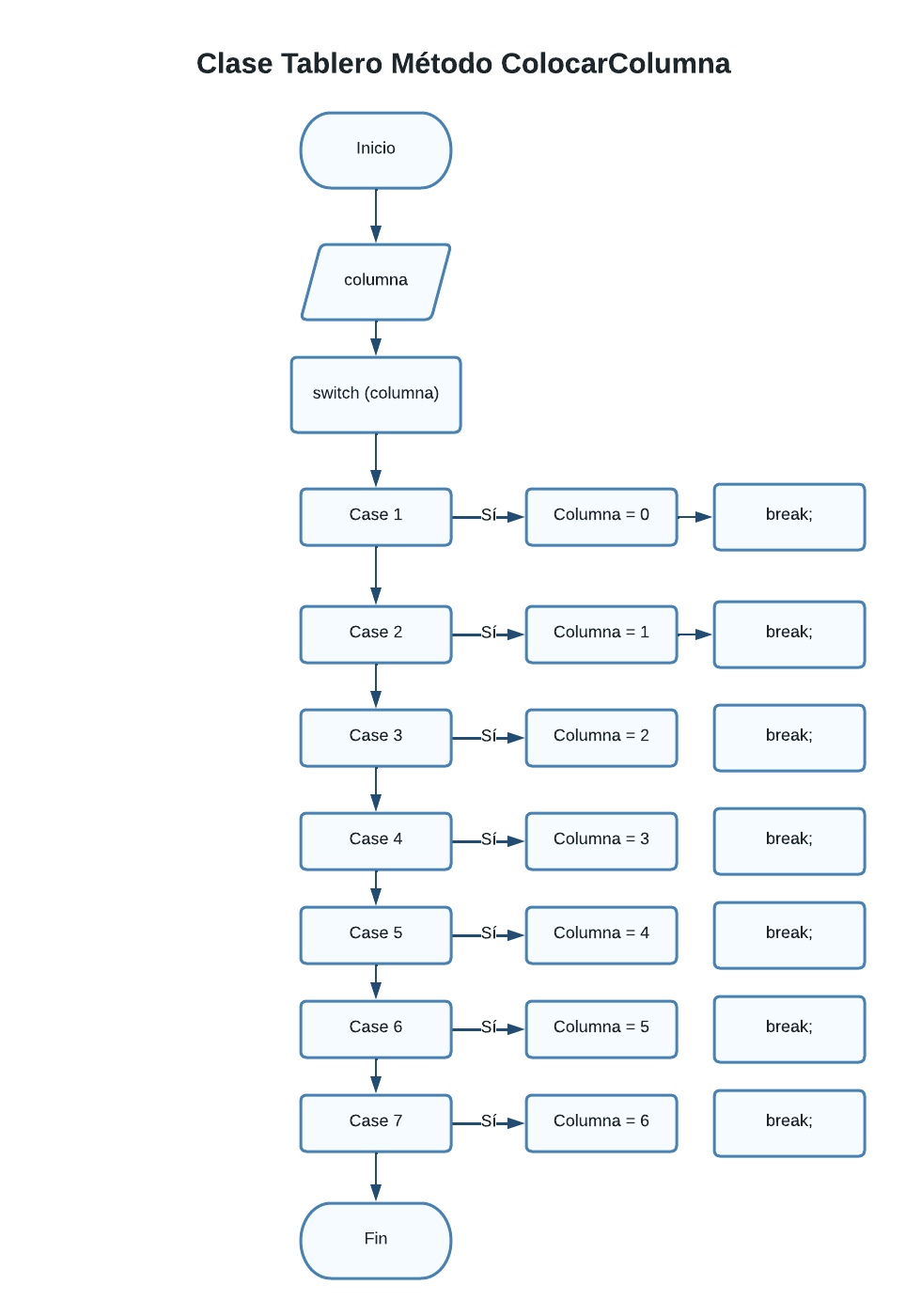


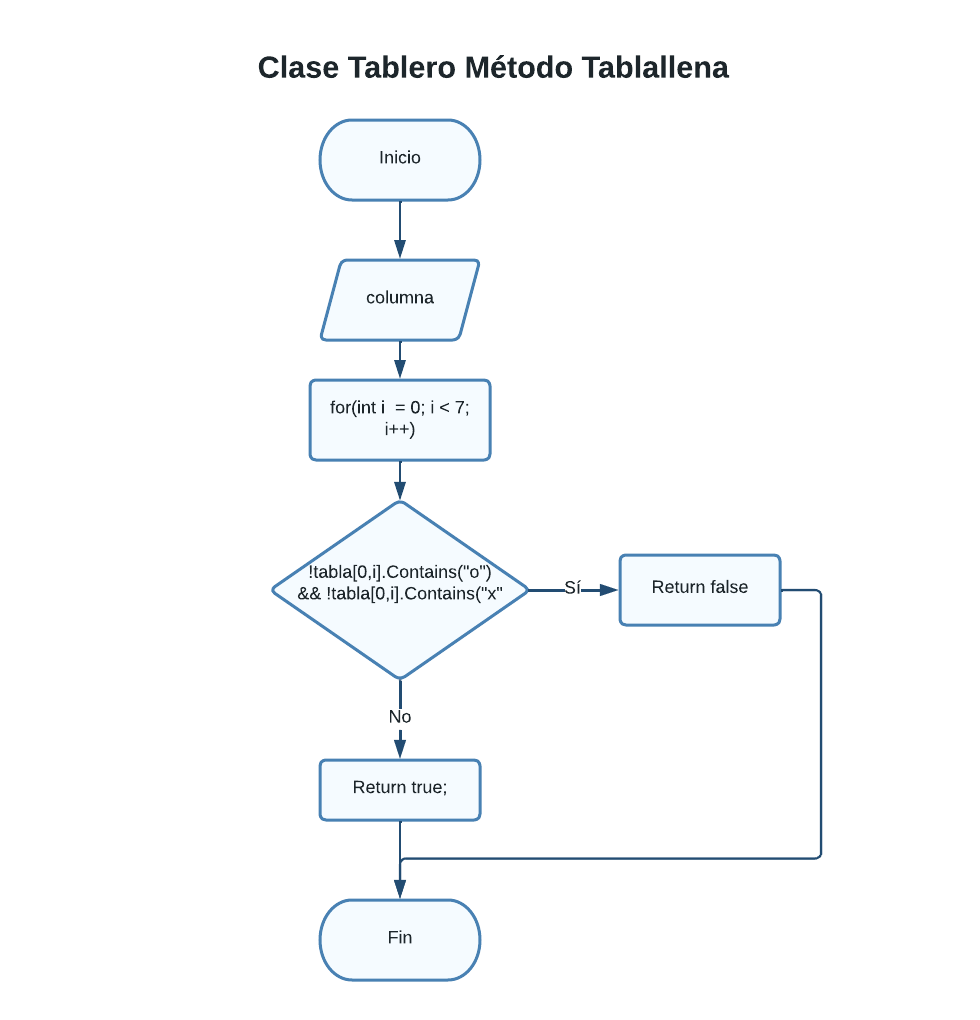


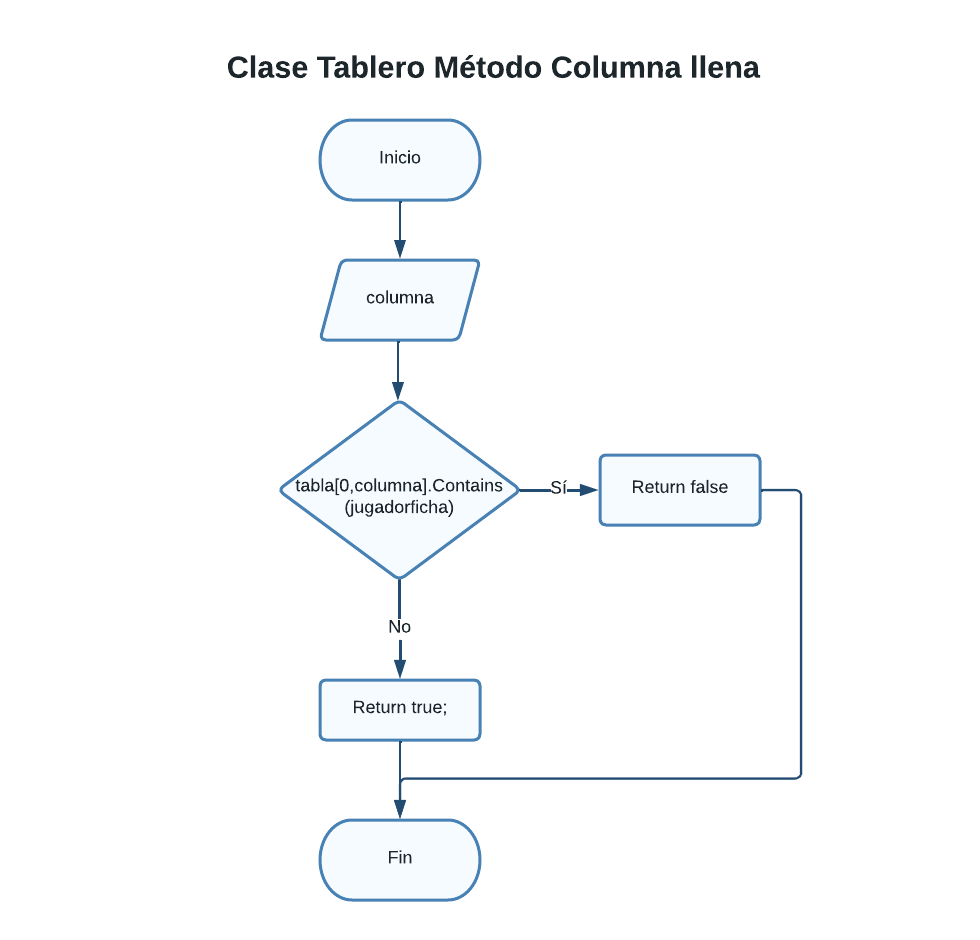


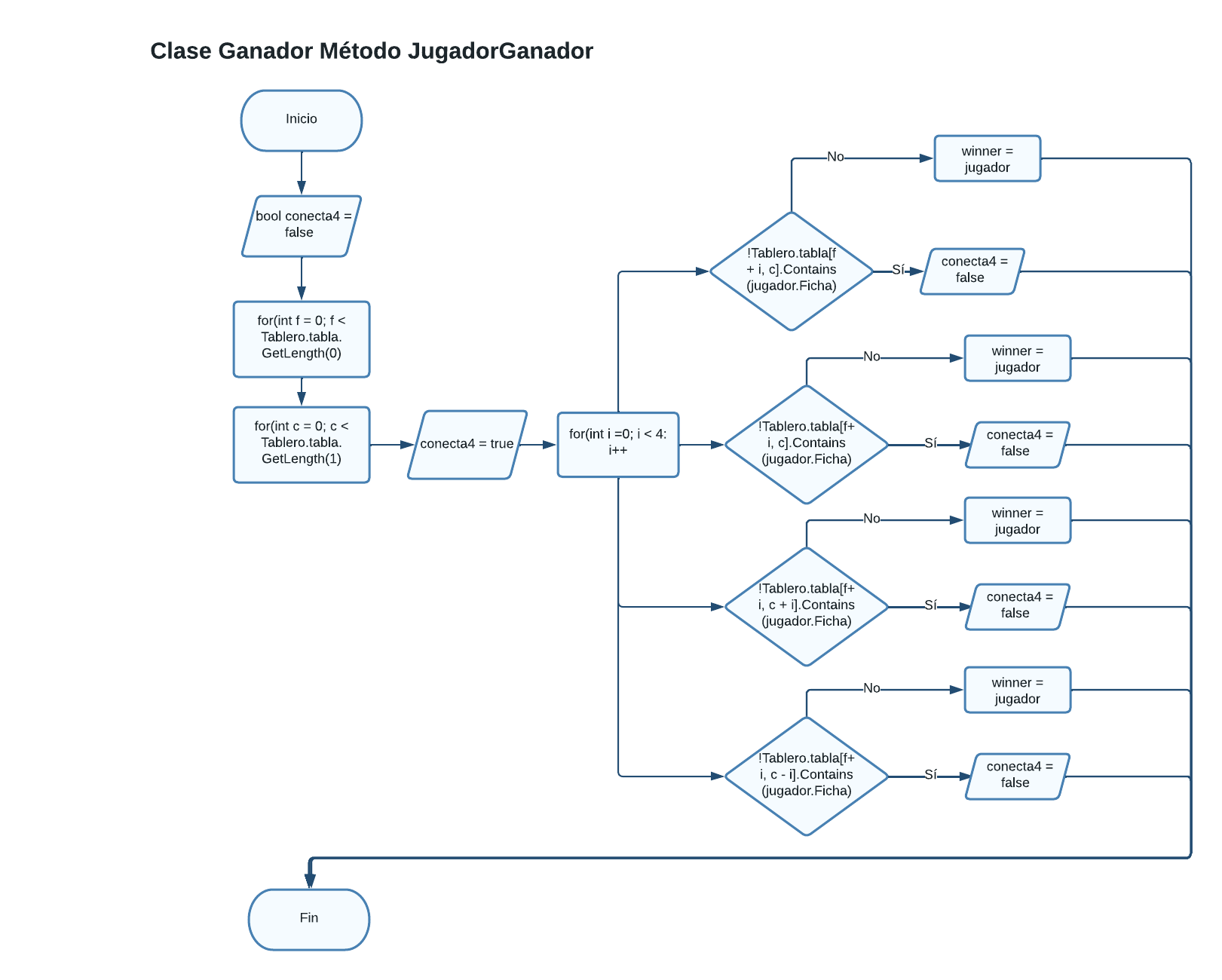


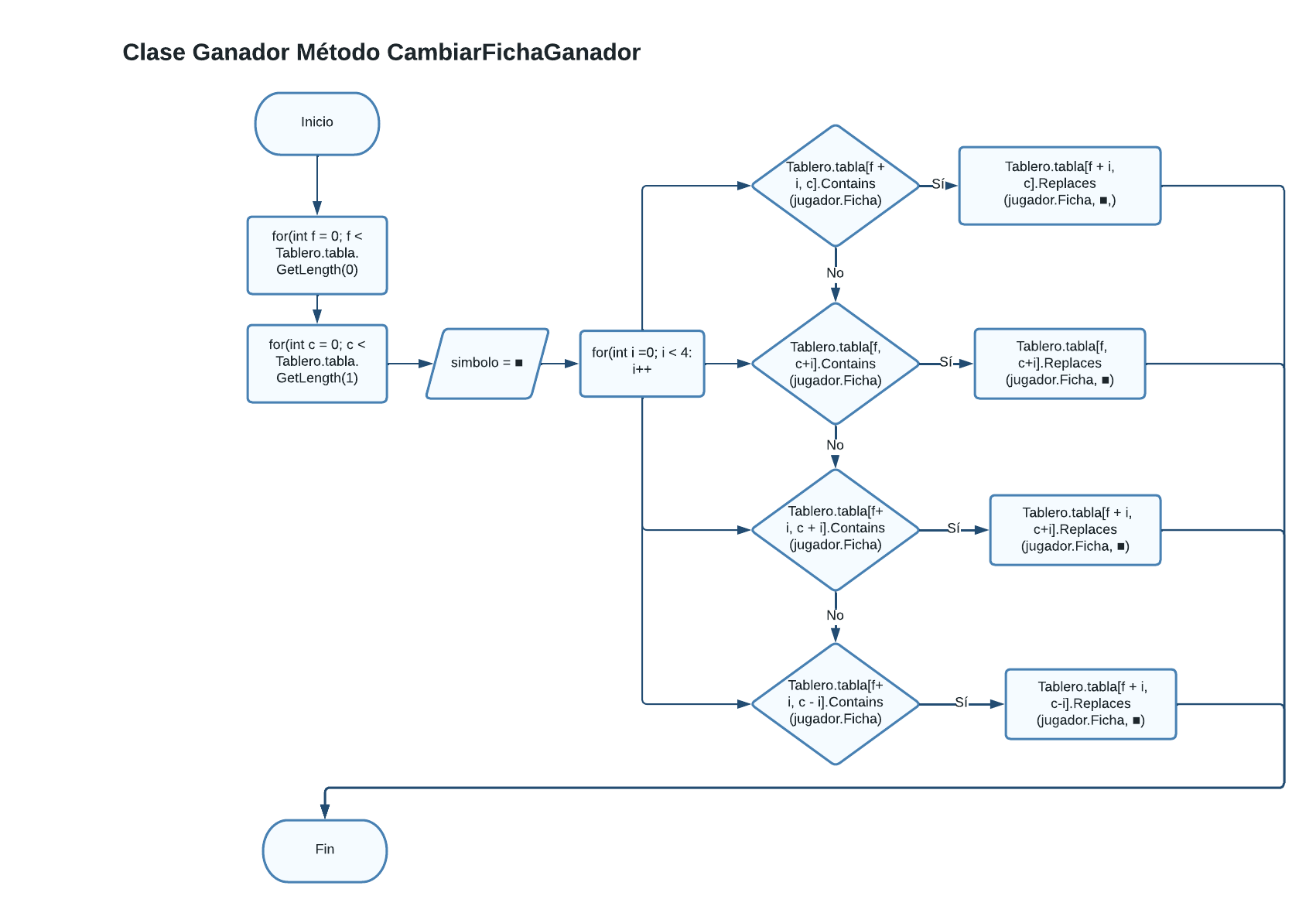


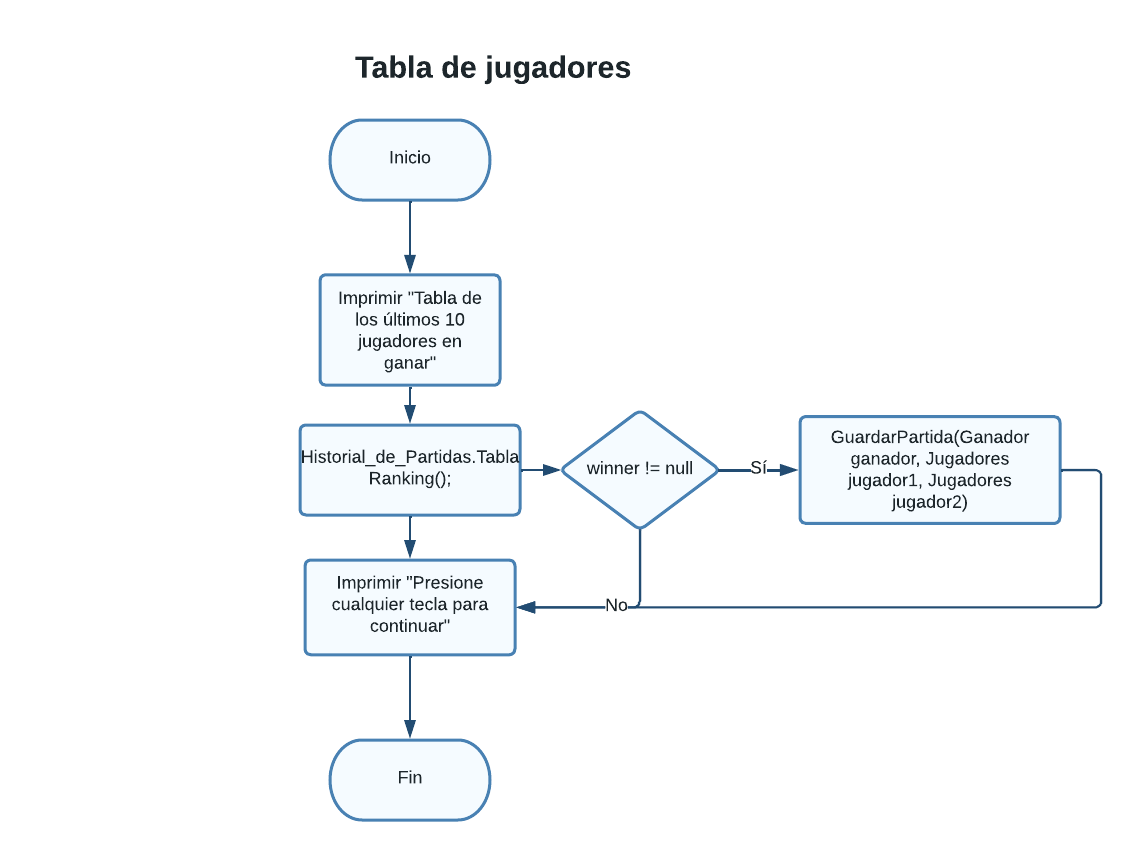












**CONCLUSIONES**

Se determinó la utilidad de la programación orientada a objetos, permitiendo modelar los “objetos” del juego” Conecta 4” (tablero, fichas, jugadores) como clases con propiedades y métodos, significando en una representación más clara y organizada del juego y su funcionamiento, asimismo, la eficiencia de los arreglos y matrices para la representación y manipulación de datos en el juego, como el estado actual del tablero, las fichas que fueron colocadas por los jugadores o el historial de los últimos diez jugadores en ganar un juego en el modo vsJugador, por último, la importancia de los bucles para la implementación de la lógica del juego, iterando sobre los elementos del tablero, comprobando si se produjo una condición de victoria o efectuando los turnos de los jugadores correspondientes.

**RECOMENDACIONES**

* Planear y diseñar el programa antes de comenzar a codificar: Es fundamental tener una idea clara de los requerimientos y el diseño del programa antes de comenzar a escribir código. Esto ayudará a evitar errores y problemas de diseño en etapas posteriores del proyecto.
* Utilizar la programación orientada a objetos: La programación orientada a objetos facilita el modelado de los objetos del juego de manera clara y organizada, lo que facilita la organización y funcionamiento del programa.
* Implementar la lógica del juego en un bucle principal, permitiendo que los jugadores realicen movimientos de fichas y comprobando si se ha producido una victoria o empate.
* Utilizar arreglos y matrices para representar el estado del juego, es una manera eficiente de representar el estado del tablero y las fichas en el juego.
* Implementar la validación de entrada de usuario, asegurándose de que el usuario no pueda introducir datos incorrectos o maliciosos.

**REFERENCIAS**

**System.Collections.Generic**

Librería utilizada para realizar la matriz del tablero, en el cual se colocarán las fichas de los jugadores, y estos últimos puedan decidir en que columna colocar sus respectivas fichas.

**System.ComponentModel.DataAnnotations.Schema**

Librería utilizada para guardar los resultados de los jugadores, como sus atributos, y las partidas para hacer referencia de la mismas en la tabla de los 10 últimos jugadores en ganar, mostrando dichos datos.

**System.ComponentModel.Design**

Librería utilizada para el diseño de componentes en el entorno de desarrollo de Visual Studio.

**System.Linq**

Librería utilizada para buscar las secuencias de fichas conectadas de forma horizontal, vertical o diagonalmente, para determinar el ganador y el fin del juego, también es utilizado para comprobar si la columna esta llena de fichas de ambos jugadores o si el tablero de juego esta lleno, significando un empate.

**System.Text**

Librería utilizada para trabajar con cadenas de texto que representan los nombres de los jugadores, mensajes de error, mensajes de victoria, entre otros aspectos del juego que involucren algún texto; también utilizado para crear mensajes personalizados de victoria en función de los nombres de los jugadores.

**System.Threading.Tasks**

Librería utilizada para realizar varias tareas a la vez, como la comprobación de la posición de las fichas mientras los usuarios van decidiendo la columna en donde jugaran su ficha, y la van colocando.

**System.Diagnostics**

Librería utilizada para crear un cronometro que comience con el inicio de la partida, parando al finalizar la misma, guardando el tiempo transcurrido, para hacer referencia de este tiempo en la tabla de los últimos 10 jugadores en ganar una partida.

**System.Runtime.Intrinsics.X86**

Librería utilizada para la aceleración del calculo de las operaciones de la lógica del juego, como la verificación de posibles combinaciones ganadoras y el intercambio de las fichas ganadoras por otros símbolos, para su fácil reconocimiento.

**System.Runtime.CompilerServices**

Librería utilizada para reconocer los puntos de optimización presentes en el código, que puedan mejorar el rendimiento del juego, y para implementar características avanzadas en el juego.

**MANUAL DE USUARIO**

[**https://youtu.be/XH7jI\_NH2H4**](https://youtu.be/XH7jI_NH2H4)