Universidad CENFOTEC

Proyecto Wooduko

I Cuatrimestre 2024

Grupo 1

Integrantes:

Kate Marie Cruz Martínez

Stuart Piñeiro Conejo

Bryan Rojas Quesada

Abril 2024

Índice

Representación del Tablero

El tablero en Woodoku es fundamental para el desarrollo del juego. En este espacio se colocarán los símbolos que serán la base de las piezas a utilizar más adelante. Se presenta como una cuadrícula de 10x10 con casillas vacías marcadas con punto “.”. Aquí es donde los jugadores pondrán a prueba su lógica y destreza al ubicar estratégicamente los símbolos para completar líneas horizontales o verticales y sumar puntos. La adecuada colocación en el tablero es clave para alcanzar el éxito en Woodoku.

```
   A B C D E F H I J
 1 . . . . . . . . .
 2 . . . . . . . . .
 3 . . . . . . . . .
 4 . . . . . . . . .
 5 . . . . . . . . .
 6 . . . . . . . . .
 7 . . . . . . . . .
 8 . . . . . . . . .
 9 . . . . . . . . .
10 . . . . . . . . .
```

Representación de las Piezas

Las piezas del juego se representarán con los símbolos ♠, *, %, $, & y #. Estos símbolos serán distribuidos en las celdas en blanco, representadas con un punto “.”, para darle la forma a cada una de las piezas. Dichas piezas son las que utilizará el usuario para interactuar con ellas al colocarlas en el tablero anteriormente representado para formar conjuntos y llenar los espacios vacíos.

A continuación, se presentan las formas de las piezas:

- #

- ♠ ♠
  ♠ ♠

- * *
  *

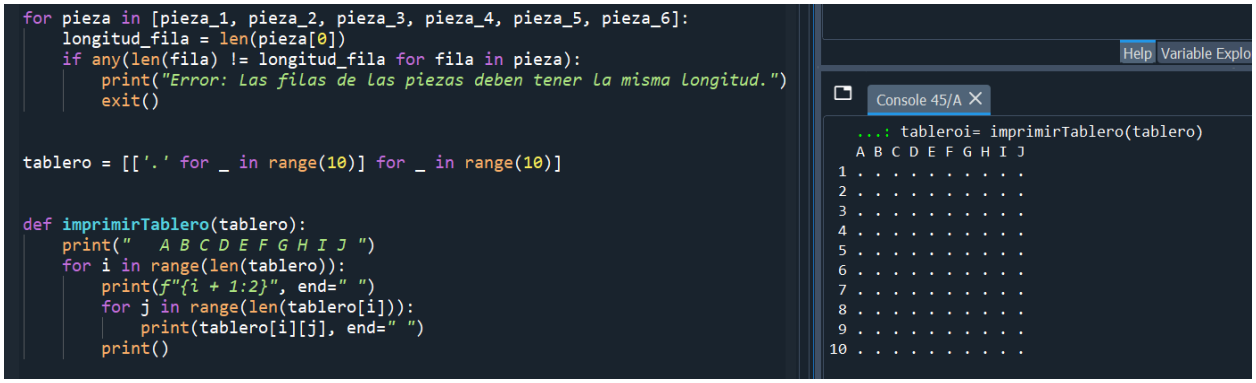
- % %
  % % %

- $
  $

- &
  & &
  & &

Mostrar el Tablero en la Consola

En la siguiente figura se muestra la representación del tablero de juego en la consola del IDE. Junto con él, se encuentra en la izquierda de la siguiente imagen, la propuesta de código que regirán las reglas del juego. El archivo completo del programa llamado “Código Proyecto - Avance 2” se incluye en el mismo espacio de este entregable. De esta manera se logra apreciar una visualización previa de lo que será el juego final.



Condicionales

Las estructuras condicionales permiten establecer los límites que se utilizarán para formar el tablero. Lo anterior se hace con la finalidad de que existan parámetros de jugabilidad y darle un sentido lógico espacial en 2 dimensiones al tablero. Debido a que el tablero consiste en una formación cuadrada de 10x10 representada con puntos (.), se establece que las piezas no puedan sobrepasar dicha estructura. Si la pieza llega a salirse, se pierde el juego.

Además, el juego no va a permitir que el usuario coloque una pieza en donde ya existe una, por lo que el juego terminará si el jugador intenta colocar una pieza encima de otra y se le indicará que perdió.

Seguidamente, se presenta la propuesta de condicionales para los propósitos anteriormente estipulados:

```
def ingresarPieza(tablero, pieza, fila, columna):
    for i in range(len(pieza)):
        for j in range(len(pieza[i])):
            if pieza[i][j] != ' ':
                if (fila + i < 0 or fila + i >= len(tablero) or
                    columna + j < 0 or columna + j >= len(tablero[0])):
                    print("¡Te saliste del tablero! Has perdido.")
                    return False
                if tablero[fila + i][columna + j] != '.':
                    print("¡Hay una pieza en ese lugar! Has perdido.")
                    return False
                tablero[fila + i][columna + j] = pieza[i][j]
    return True
```

Se declaran líneas de código que representen el fin del juego. Esto se hace para determinar cuándo se deja de correr el programa. En la imagen anterior, se presentan dos casos en donde se puede acabar el juego.

A continuación se muestran las validaciones que verifican que los datos ingresados correspondientes a las filas y columnas del tablero sean aceptables. Además, se verifica que se elija una pieza adecuada para jugar.

```
if opcionPieza < 1 or opcionPieza > 6:
    print("Por favor, ingrese un número de pieza válido (1-6).")
    continue

if fila < 0 or fila > 9 or columna < 0 or columna > 9:
    print("Por favor, ingrese una fila y columna válidas.")
    continue
```

Una vez que se verifica la parte anterior, se procede a asociar cada pieza con su respectivo número para que el jugador pueda utilizarlas a su conveniencia.

```
if opcionPieza == 1:
    pieza = pieza1
elif opcionPieza == 2:
    pieza = pieza2
elif opcionPieza == 3:
    pieza = pieza3
elif opcionPieza == 4:
    pieza = pieza4
elif opcionPieza == 5:
    pieza = pieza5
else:
    pieza = pieza6
```

Se le pregunta al usuario si desea rotar la pieza seleccionada. De ser así, se procede a rotar dicha pieza y se imprime en el tablero.

```
rotar = input("¿Desea rotar la pieza (si/no)? ").lower() == 'si'
if rotar:
    pieza = rotarPieza(pieza)

if not ingresarPieza(tablero, pieza, fila, columna):
    return False
```

Si el jugador desea cambiar la posición original de la pieza, se utilizan condicionales para saber qué pieza se debe rotar de acuerdo con la pieza original según lo indique el usuario.

```
def rotarPieza(pieza):
    rotada = []
    if pieza == pieza1:  # No hay rotación para pieza_1
        rotada = pieza
    elif pieza == pieza2:
            rotada = piezaRotada2
    elif pieza == pieza3:
        rotada = piezaRotada3
    elif pieza == pieza4:
        rotada = piezaRotada4
    elif pieza == pieza5:
        rotada = piezaRotada5
    elif pieza == pieza6:
            rotada = piezaRotada6
    return rotada
```

Si el usuario ha alcanzado una puntuación determinada (50 puntos en este caso), se imprime un mensaje de victoria que indica que ha ganado. En caso contrario, continuará jugando.

```
def verificarVictoria(puntaje):
    if puntaje >= 50:
        print("¡Felicidades! Has alcanzado 50 puntos. ¡Has ganado!")
        return True
    return False
```

En esta parte, el usuario decide si desea jugar nuevamente llamando a la función “reiniciarJuego()”. Si el usuario introduce una respuesta diferente a “si”, el juego termina en ese momento.

```
reinicio = True
while reinicio:
    reinicio = reiniciarJuego()
    if not reinicio:
        break
    respuesta = input("¿Quieres jugar de nuevo? (si/no): ").lower()
    if respuesta != 'si':
        break
```

Se define una función llamada verificarLineasCompletas y se inicializa una variable lineasCompletas con el valor 0.

```
def verificarLineasCompletas():
    global tablero
    lineasCompletas = 0
```

Se realiza un bucle for sobre cada fila del tablero.

Dentro del bucle, se verifica si no hay ningún punto '.' en la fila actual del tablero.

Si no hay ningún punto en la fila, se incrementa el contador lineasCompletas en 1, se elimina la fila completa del tablero (tablero.pop(i)), y se inserta una nueva fila vacía al inicio del tablero.

```
    for i in range(len(tablero)):
        if '.' not in tablero[i]:
            lineasCompletas += 1
            tablero.pop(i)
            tablero.insert(0, ['.' for _ in range(10)])
```

Luego, se realiza otro bucle for sobre cada columna del tablero.

Dentro del segundo bucle, se crea una lista columna que contiene los elementos de la columna actual. Se verifica si no hay ningún punto '.' en la columna.

Si no hay ningún punto en la columna, se incrementa el contador lineasCompletas en 1. Se elimina la columna completa del tablero (tablero[i].pop(j)), y se inserta un punto '.' al principio de cada fila del tablero (tablero[i].insert(0, '.')).

```
    for j in range(len(tablero[0])):
        columna = [tablero[i][j] for i in range(len(tablero))]
        if '.' not in columna:
            lineasCompletas += 1
            for i in range(len(tablero)):
                tablero[i].pop(j)
                tablero[i].insert(0, '.')

    return lineasCompletas
```

Finalmente, la función devuelve el valor de lineasCompletas.

Tabla de Variables

Descripción	Nombre	Tipo de Variable	Valor
Pieza número uno del tablero	Pieza1	String	#
Pieza número dos del tablero	Pieza2	String	♠ ♠ ♠ ♠
Pieza número tres del tablero	Pieza3	String	* * *
Pieza número cuatro del tablero	Pieza4	String	% % % % %
Pieza número cinco del tablero	Pieza5	String	$ $
Pieza número seis del tablero	Pieza6	String	& & & & &
Pieza número uno del tablero rotada	PiezaRotada1	String	#
Pieza número dos del tablero rotada	PiezaRotada2	String	♠ ♠ ♠ ♠
Pieza número tres del tablero rotada	PiezaRotada3	String	* * *
Pieza número cuatro del tablero rotada	PiezaRotada4	String	% % % % %
Pieza número cinco del tablero rotada	PiezaRotada5	String	$ $
Pieza número seis del tablero rotada	PiezaRotada6	String	& & & & &
Tablero de juego	Tablero	String	**A B C D E F H I J** **1** . . . . . . **2** . . . . . .

			3 . . . . . . **4** . . . . . . **5** . . . . . . **6** . . . . . . **7** . . . . . . **8** . . . . . . **9** . . . . . . **10** . . . . .

Diagrama General

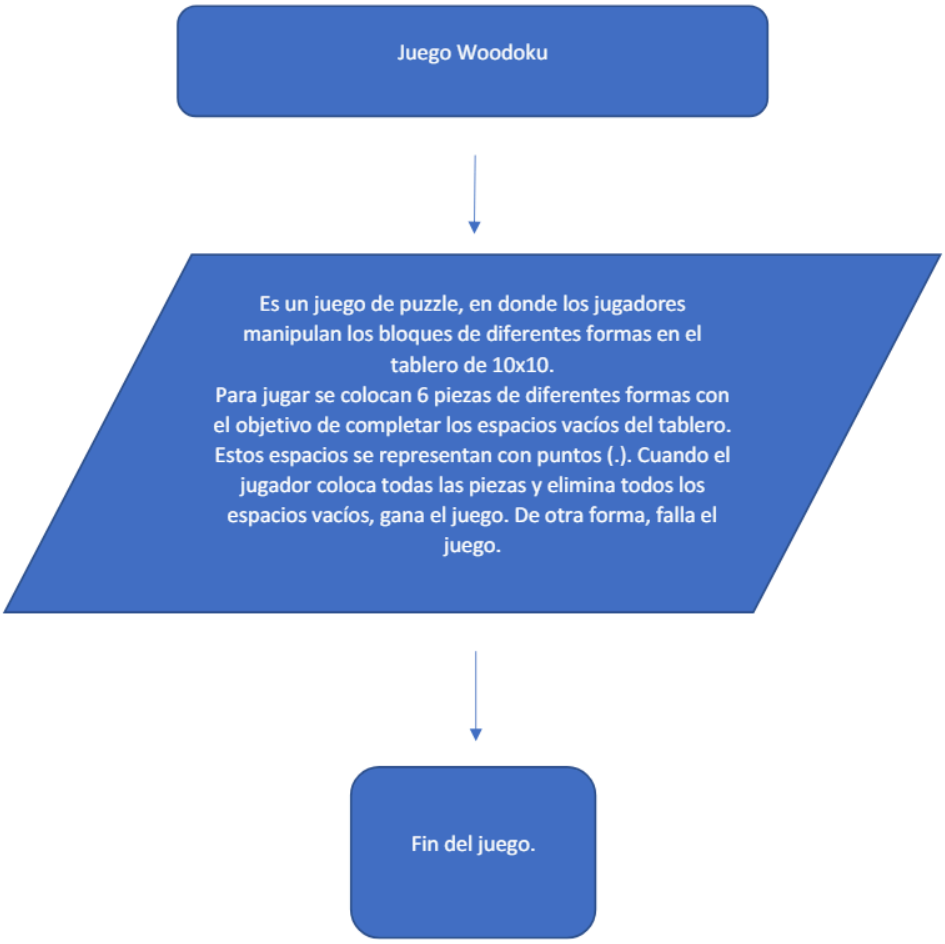
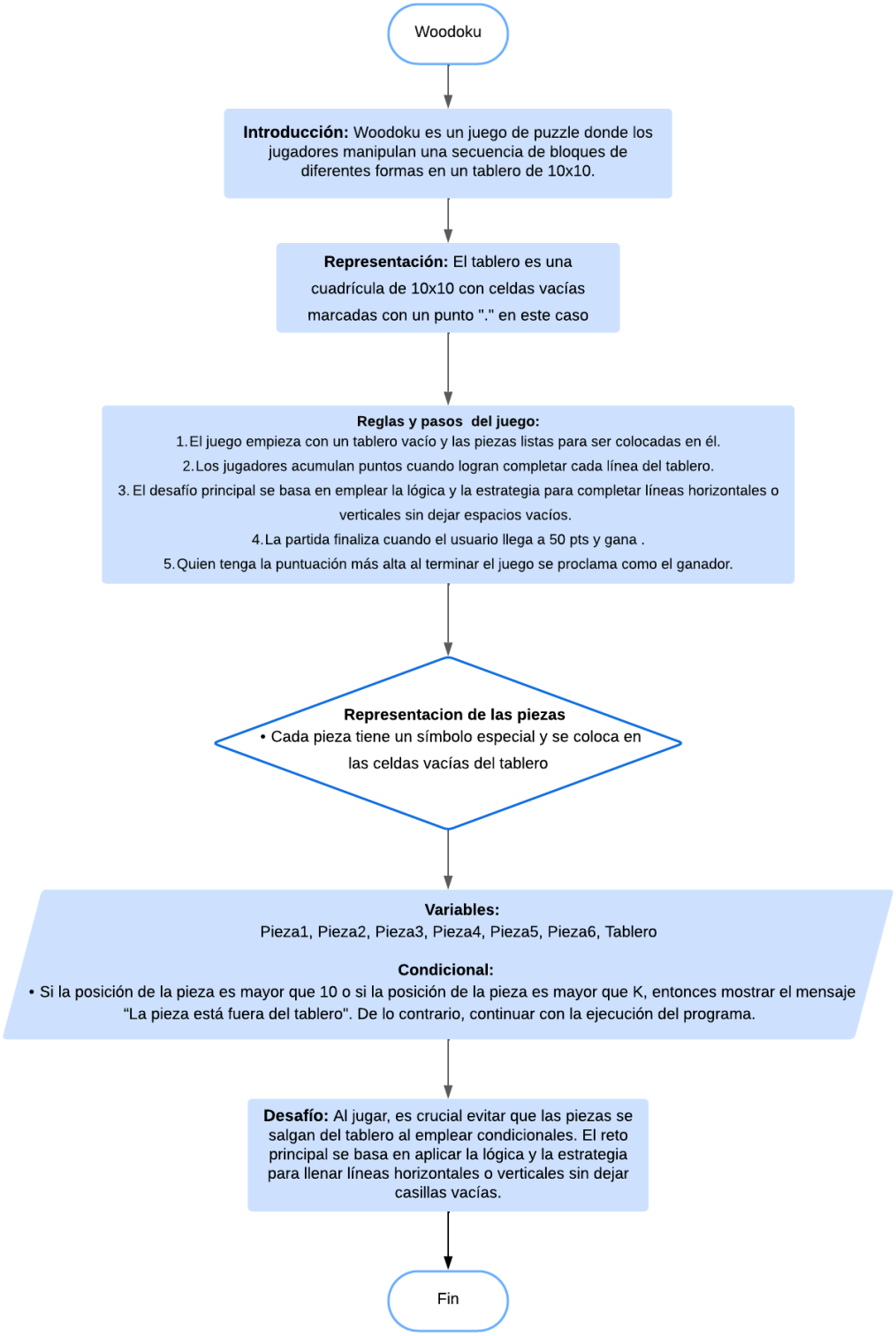


Diagrama Explicativo



Diagramas de Flujo

Diagrama de Flujo (Avance 1)

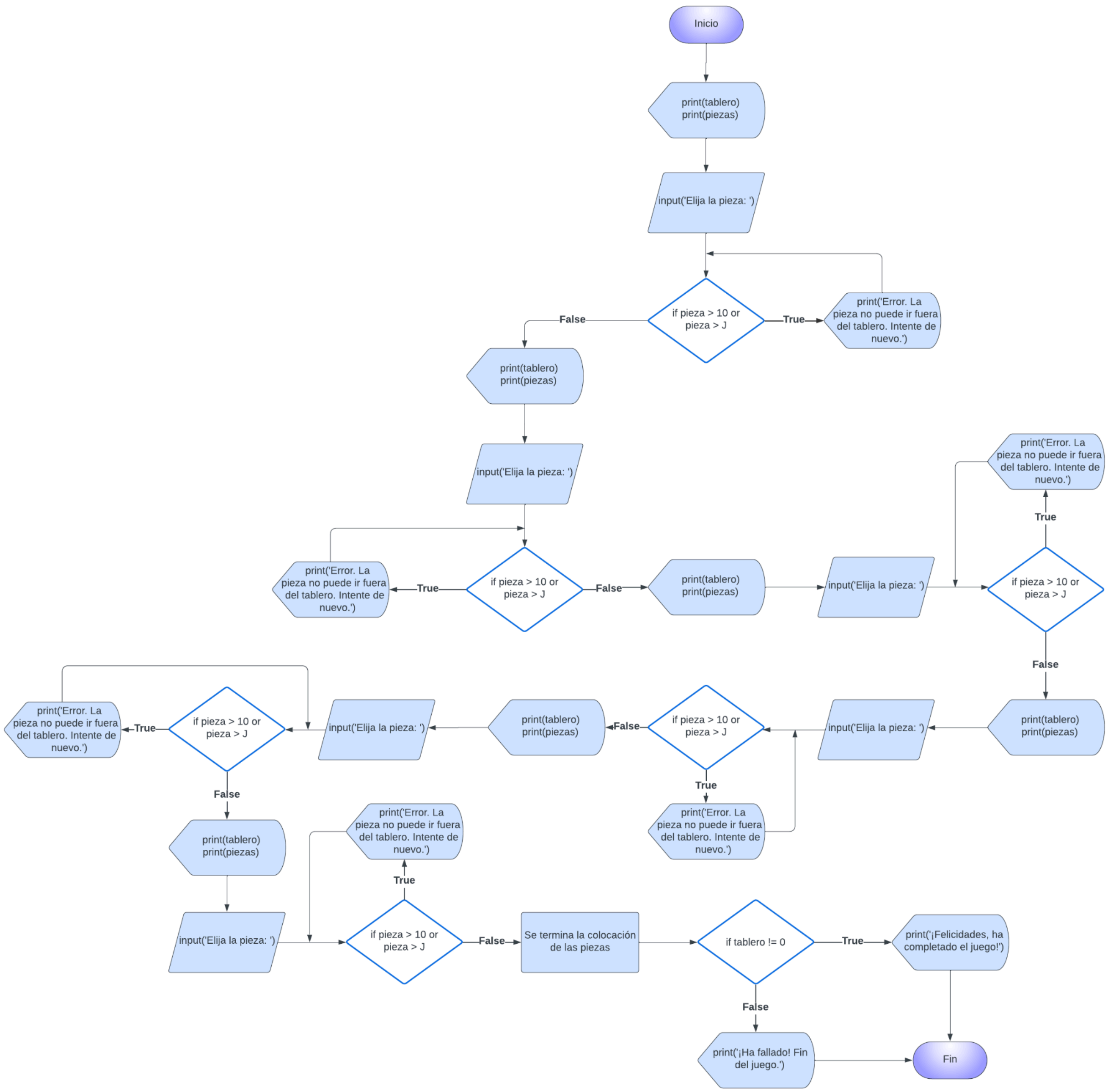


Diagrama de Flujo de Imprimir Tablero

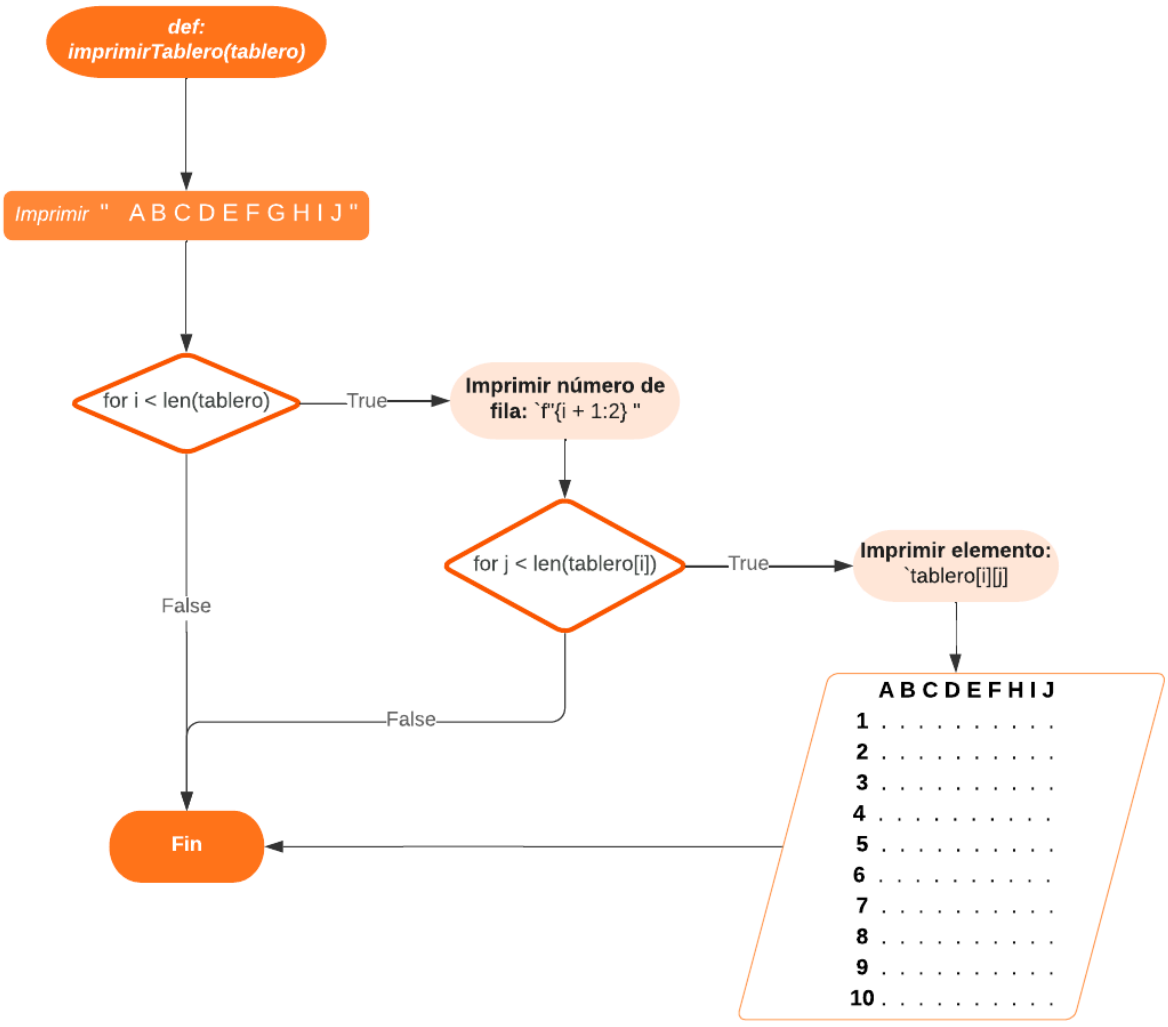


Diagrama de Flujo de Ingresar Pieza

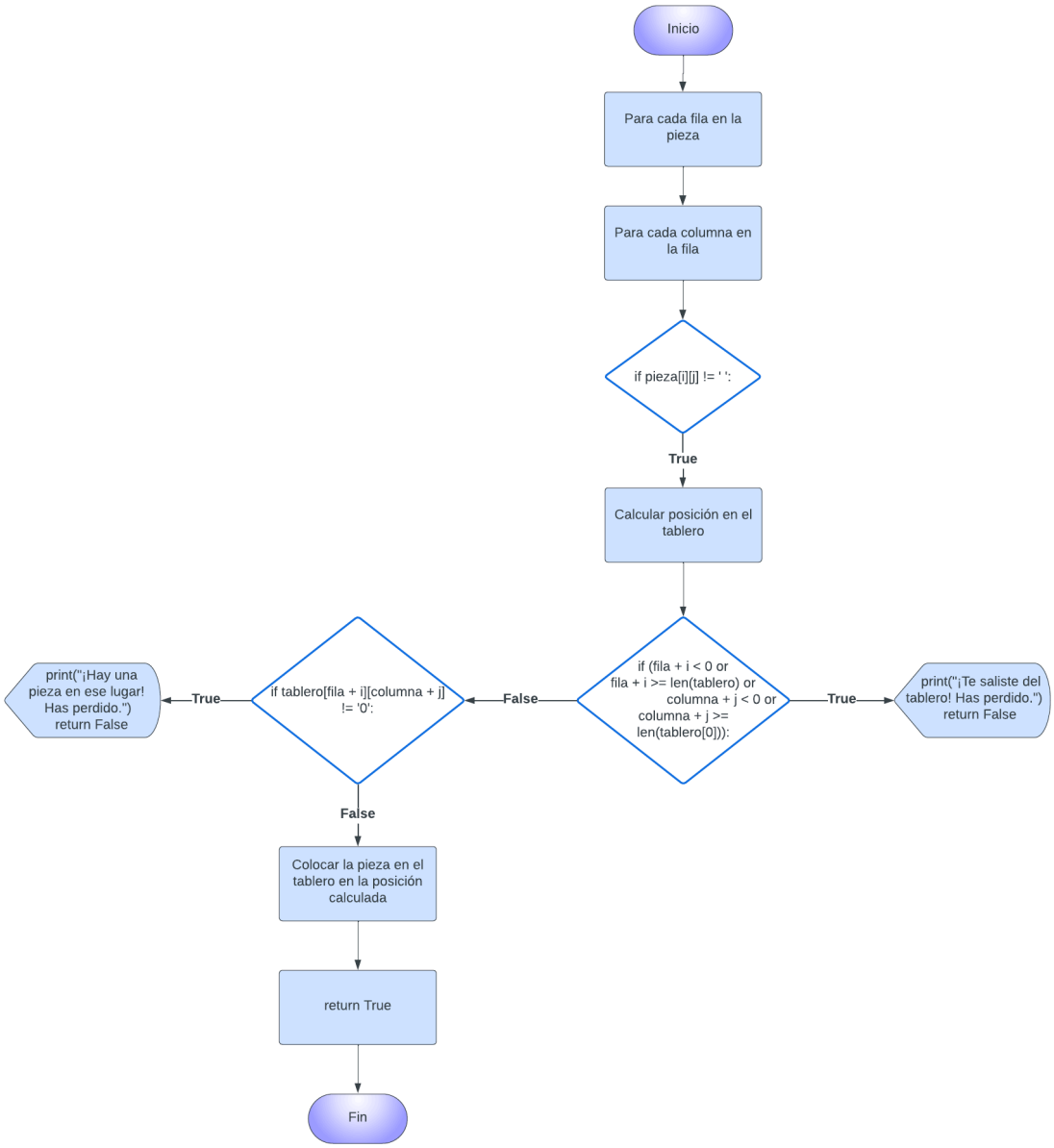


Diagrama de Flujo de Piezas Disponibles

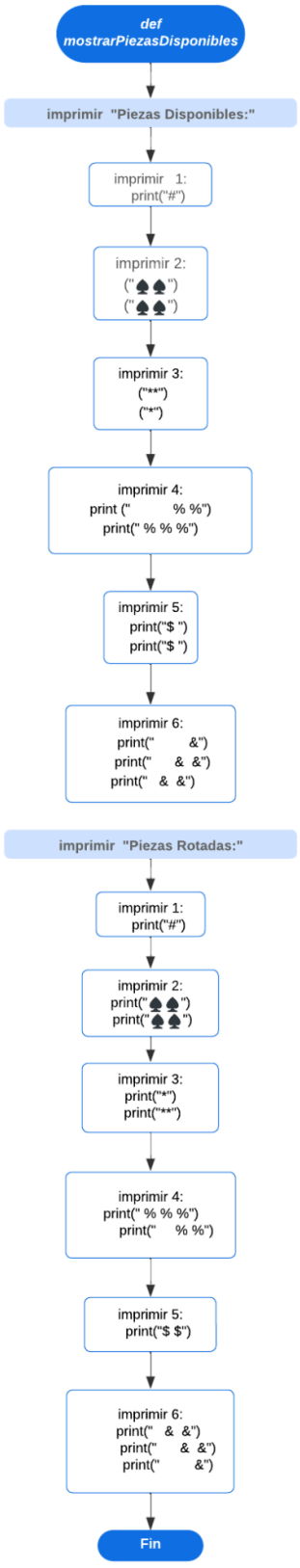


Diagrama de Flujo de Reiniciar Juego

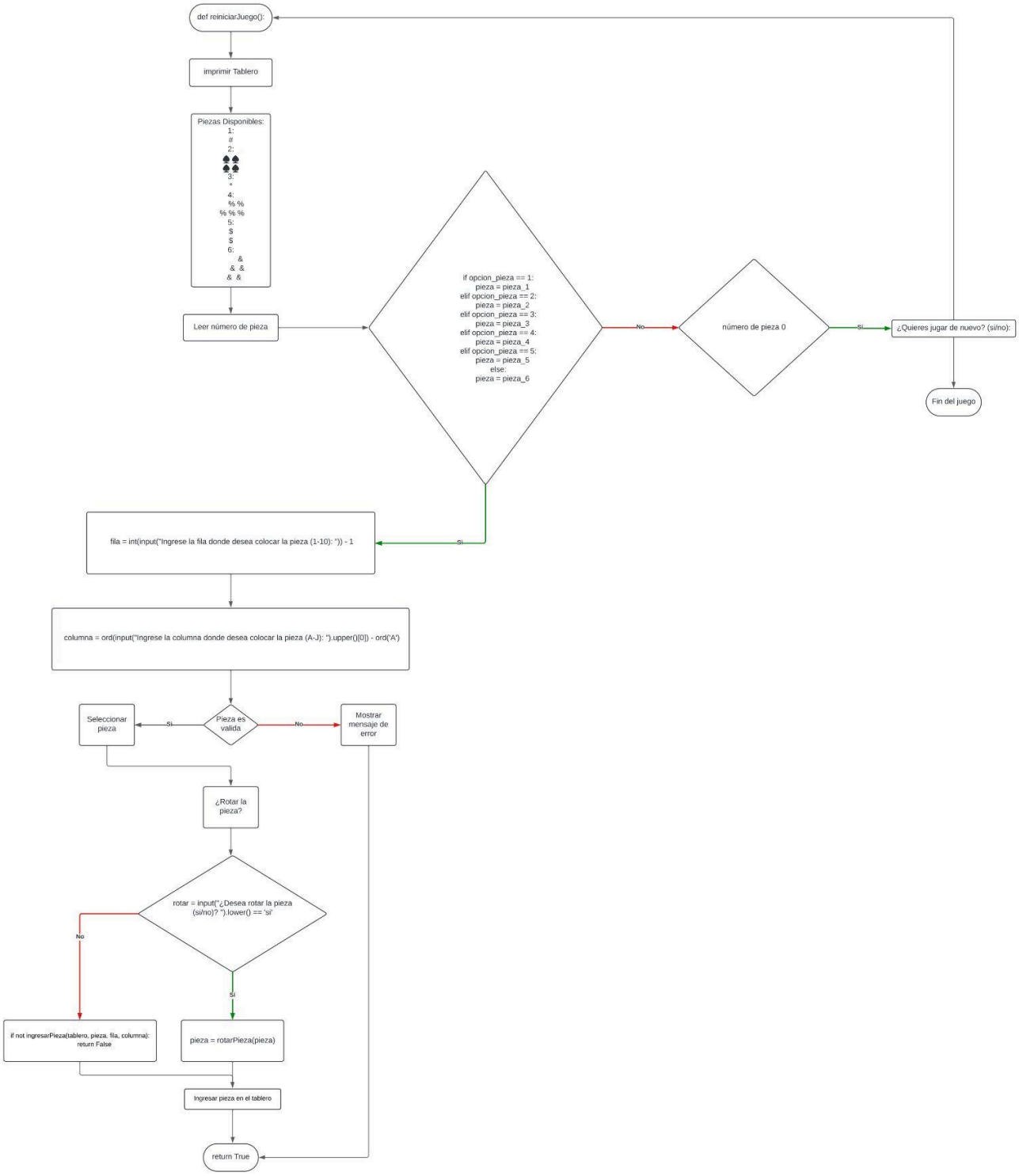


Diagrama de Flujo Rotar Pieza

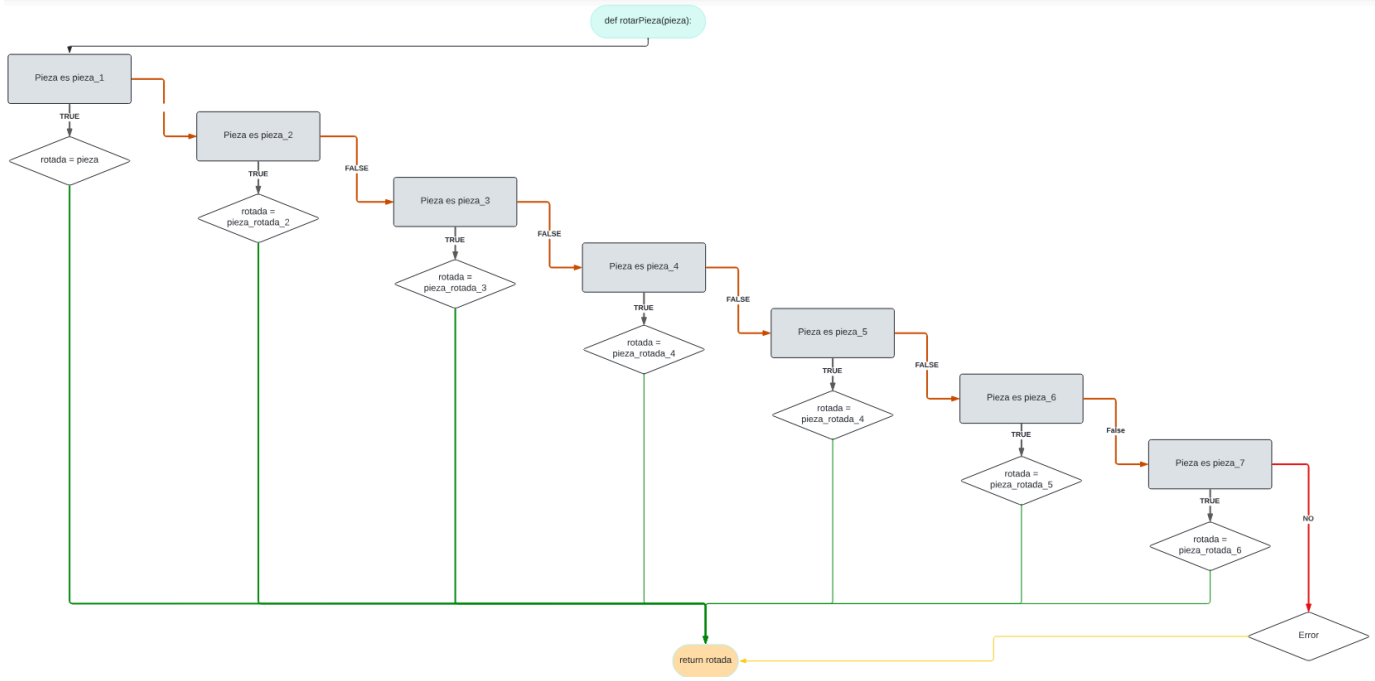


Diagrama de Flujo de Verificar Líneas Completas

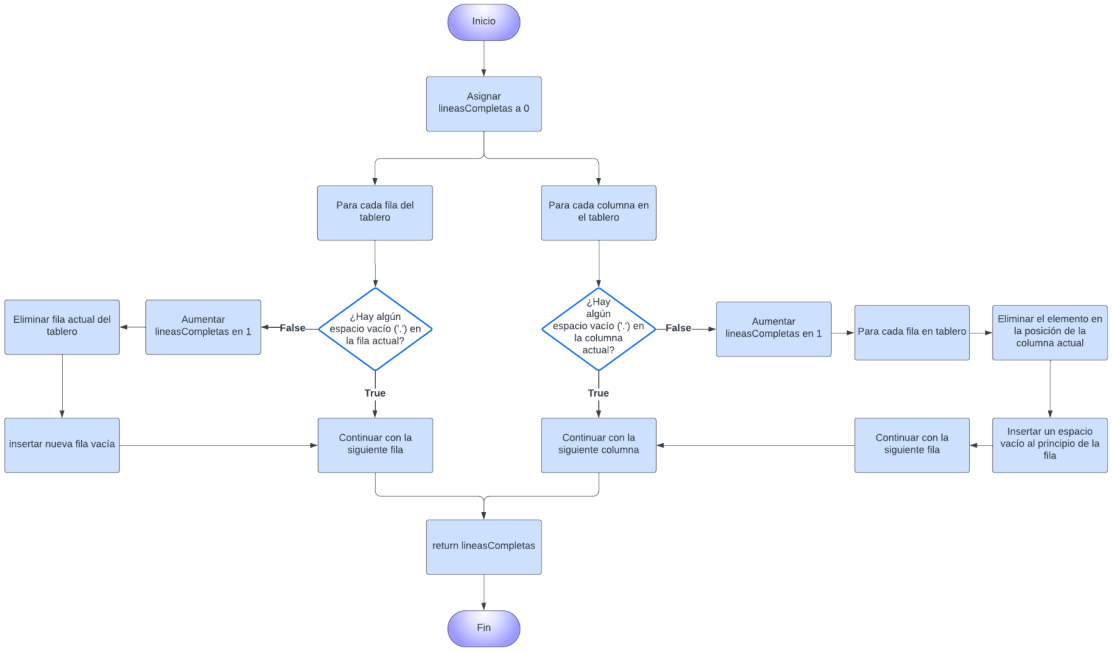


Diagrama de Flujo de Verificar Victoria