

Universidad Rafael Landívar

Facultad de Ingeniería

Laboratorio del curso Pensamiento Computacional, Sección 04

Docente: Ing. Luis Pedro Ovalle

Estudiante Auxiliar: Amanda Ramírez

PROYECTO DE LABORATORIO No.2

Estudiante: Emilio Andrés Arias García 1204124

Guatemala, 20 de mayo del 2024

ÍNDICE

ACCIONES QUE DEBE PODER HACER EL PROGRAMA.....	3
DATOS CON LOS QUE VA A TRABAJAR Y DATOS DE ENTRADA	4
VARIABLES PARA ALMACENAR INFORMACIÓN	4
CONDICIONES O RESTRICCIONES A CONSIDERAR	5
FUNCIONES A IMPLEMENTAR.....	5
Algoritmo que implementará en el programa, descrito mediante el Diagrama de Flujo elaborado en Draw.io, para mostrar la lógica de las diferentes acciones. Realizar un diagrama de flujo por cada función y un diagrama general del programa principal.....	6

ANÁLISIS Y DISEÑO DEL PROGRAMA

ACCIONES QUE DEBE PODER HACER EL PROGRAMA

1. Inicializar el tablero de ajedrez: Esto implica crear una matriz 8x8 que represente el tablero y llenarla con espacios en blanco al inicio del juego.
2. Mostrar el tablero: Es fundamental que el programa pueda mostrar el estado actual del tablero en la consola para que los jugadores puedan ver dónde están ubicadas las piezas.
3. Colocar una pieza: Permite al usuario agregar una pieza en una posición específica del tablero. Esto es esencial para comenzar el juego y para agregar más piezas posteriormente.
4. Obtener pieza en una posición: Esta acción permite al usuario obtener información sobre qué pieza se encuentra en una posición determinada del tablero. Es útil para verificar el estado del juego o para realizar movimientos estratégicos.
5. Validar una posición: Antes de colocar una pieza en una posición, es importante verificar si esa posición está dentro de los límites del tablero y si está vacía. Esta validación asegura que las reglas del juego se cumplan correctamente.
6. Obtener movimientos válidos para un caballo: Calcula y muestra los movimientos válidos que puede realizar un caballo en una posición dada. Esto es útil para que los jugadores planifiquen sus movimientos y estrategias durante el juego.

DATOS CON LOS QUE VA A TRABAJAR Y DATOS DE ENTRADA

El programa trabajará con los siguientes datos:

1. Tablero de ajedrez: Representado como una matriz 8x8, donde cada celda puede contener una pieza de ajedrez o estar vacía.
2. Piezas de ajedrez adicionales: Información sobre las piezas que el usuario desea colocar en el tablero, incluyendo el tipo de pieza (alfil, peón, etc.), el color de la pieza (blanco o negro) y la posición en la que se colocará la pieza.

Los datos de entrada incluyen:

- La posición inicial del caballo.
- La cantidad y la información de las piezas adicionales que el usuario desea colocar en el tablero.

VARIABLES PARA ALMACENAR INFORMACIÓN

1. tamaño: Representa el tamaño del tablero de ajedrez y se inicializa con el valor 8.
2. tablero: Es una matriz que representa el estado del tablero de ajedrez, donde cada elemento puede contener una pieza o estar vacío.
3. fila_caballo y columna_caballo: Almacenan las coordenadas del caballo en el tablero.
4. num_piezas, tipo_pieza, color_pieza y posicion: Son variables que almacenan la cantidad, tipo, color y posición de las piezas adicionales que se colocarán en el tablero.

CONDICIONES O RESTRICCIONES A CONSIDERAR

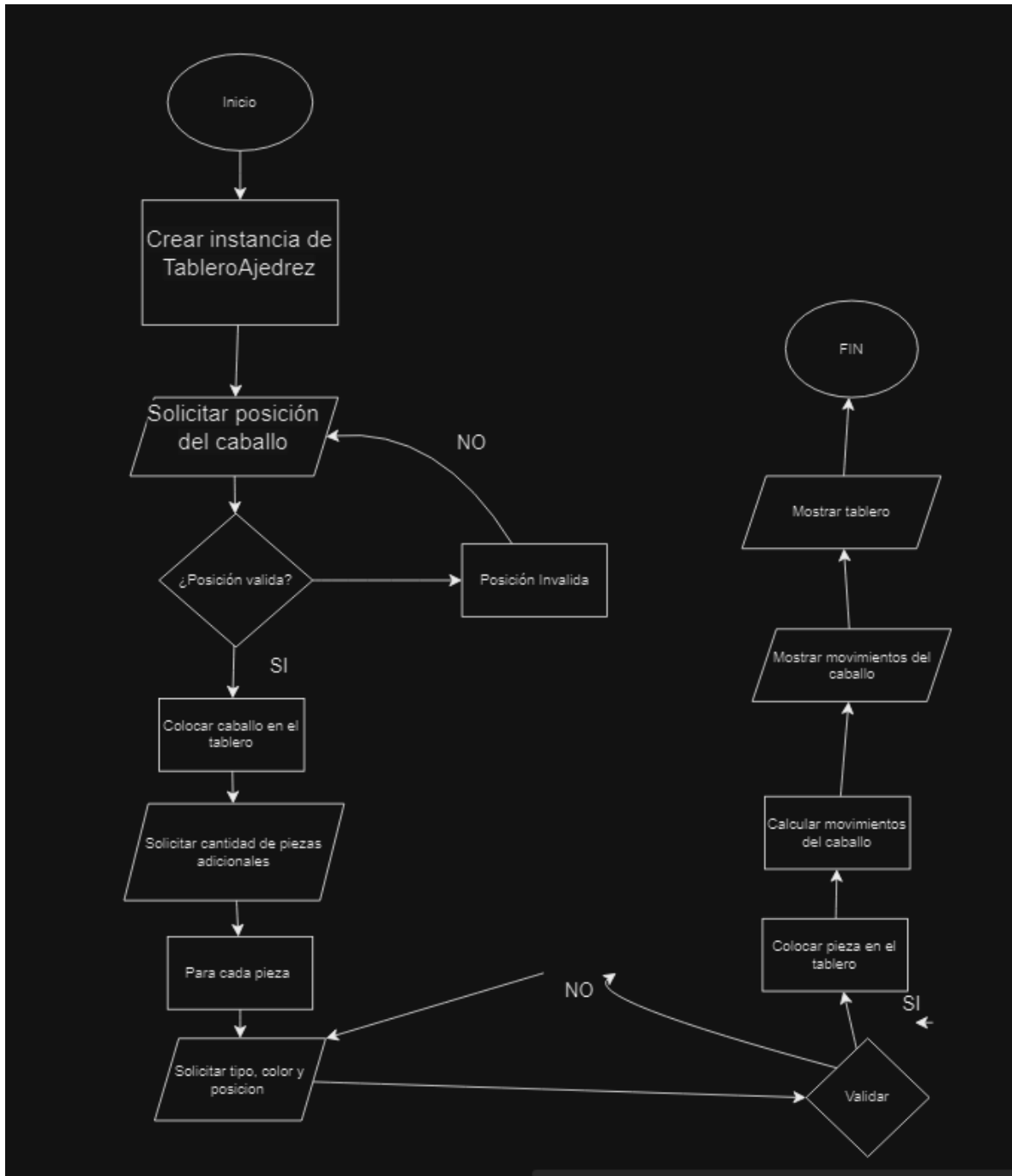
1. Límites del tablero: Las posiciones deben estar dentro de los límites del tablero, es decir, las filas y columnas deben estar entre 0 y 7.
2. Posiciones ocupadas: No se pueden colocar dos piezas en la misma posición del tablero.
3. Validación de posiciones y tipos de piezas: Las posiciones y los tipos de piezas ingresados por el usuario deben ser válidos según las reglas del ajedrez.
4. Movimientos del caballo: Al calcular los movimientos válidos para el caballo, se deben considerar las restricciones del tablero y las posiciones ocupadas por otras piezas.

FUNCIONES A IMPLEMENTAR

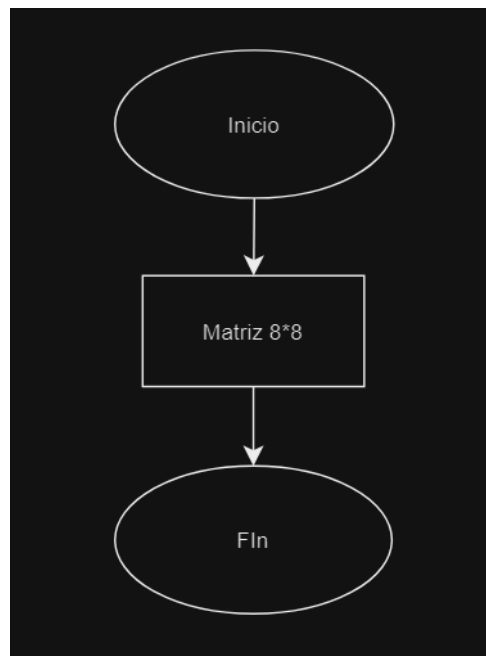
1. `__init__()`: Esta función inicializa el tablero de ajedrez al comienzo del juego.
2. `mostrar_tablero()`: Muestra el estado actual del tablero en la consola.
3. `colocar_pieza(fila, columna, pieza)`: Permite colocar una pieza en una posición específica del tablero.
4. `obtener_pieza(fila, columna)`: Devuelve la pieza que se encuentra en una posición determinada del tablero.
5. `validar_posicion(fila, columna)`: Verifica si una posición especificada está dentro de los límites del tablero y si está vacía.
6. `obtener_movimientos_caballo(fila, columna)`: Calcula y devuelve los movimientos válidos que puede realizar un caballo en una posición dada.

Algoritmo que implementará en el programa, descrito mediante el Diagrama de Flujo elaborado en Draw.io, para mostrar la lógica de las diferentes acciones. Realizar un diagrama de flujo por cada función y un diagrama general del programa principal.

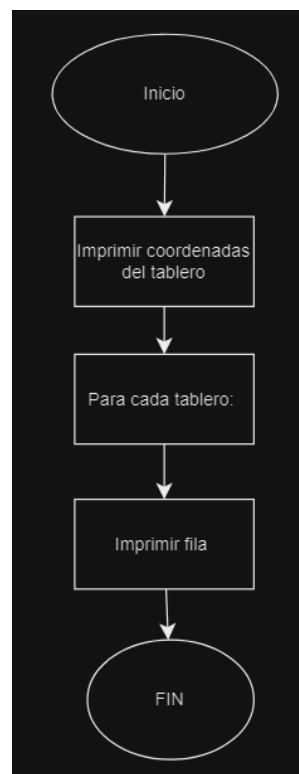
General:



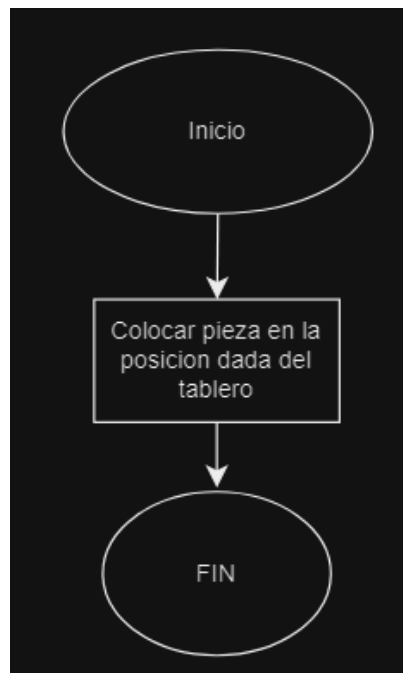
Función **__init__**:



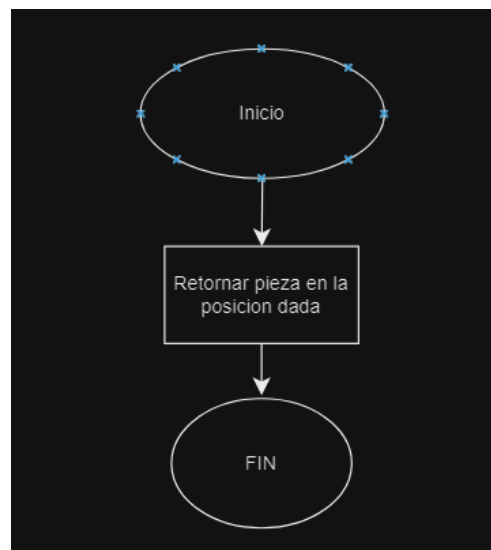
Función **mostrartablero**:



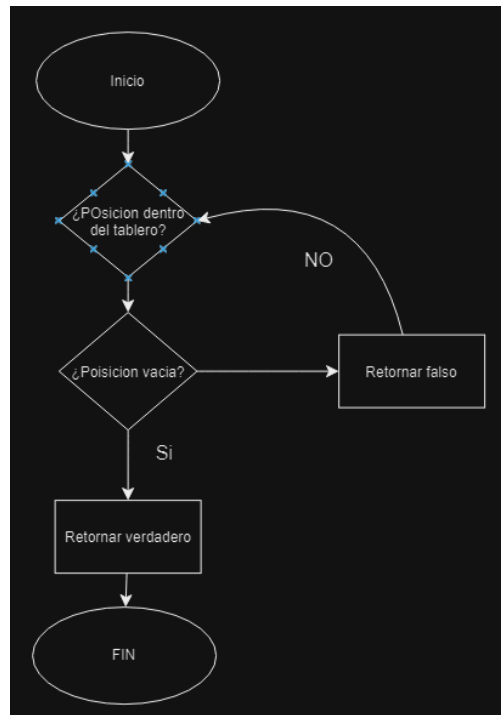
Función **colocarpieza**:



Función **obtenerpieza**:



Función **validarposicion**:



Función **obtenermovimientoscaballo**:

