

David Gil

1123224

Inicio A

## **Análisis y diseño del proyecto final**

### **¿Qué acciones debe poder hacer su programa? Enumérelas.**

Inicializar y mostrar un tablero de ajedrez con un patrón de colores alternados.

Permitir al usuario agregar diferentes piezas de ajedrez en el tablero.

Evaluar la entrada del usuario para decisiones de tipo sí/no.

Agregar específicamente un caballo al tablero y evaluar sus posibles movimientos.

Validar las casillas de posibles movimientos del caballo y determinar si están vacías o contienen piezas enemigas.

Mostrar los movimientos posibles del caballo en el tablero.

### **¿Con qué datos va a trabajar? ¿Qué información debe pedir al usuario? Defina sus datos de entrada.**

Datos de entrada:

Elección de la pieza a agregar (Rey, Reina, Alfil, Torre, Caballo, Peón).

Color de la pieza a agregar (Blanco o Negro).

Fila y columna del tablero donde se desea colocar la pieza.

Decisión de seguir agregando piezas (Sí o No).

Color del caballo a agregar.

Fila y columna del caballo a agregar.

**¿Qué variables utilizará para almacenar la información? Defina el nombre y el tipo de dato que utilizará para gestionar estos datos principales.**

Variables:

tablero (list of lists): Matriz 8x8 que representa el tablero de ajedrez.

pblancas (list): Lista de piezas blancas en Unicode.

pnegras (list): Lista de piezas negras en Unicode.

lapieza (str): Cadena que almacena la pieza actual a agregar.

fila (int): Entero que representa la fila donde se colocará la pieza.

columna (int): Entero que representa la columna donde se colocará la pieza.

seguir (bool): Booleano para controlar el ciclo de agregar piezas.

valp, valc, valco, valfi (bool): Booleanos para validar diferentes entradas del usuario.

**¿Qué condiciones o restricciones debe tomar en cuenta? ¿Qué cálculos debe hacer?**

Condiciones y restricciones:

Validar que las filas y columnas ingresadas estén en el rango 1-8.

Validar que las entradas sean números enteros cuando corresponda.

Verificar que la casilla elegida esté vacía antes de agregar una nueva pieza.

Verificar que los movimientos del caballo no se salgan del tablero y evaluar si la casilla está vacía o contiene una pieza del enemigo.

Los movimientos del caballo deben seguir las reglas del ajedrez (en L).

Cálculos:

Cálculo de las posiciones posibles del caballo en base a su posición actual.

Determinación de si una casilla contiene una pieza enemiga, y si es así, identificarla.

## ¿Qué funciones implementará?

`imprimirtableroInicial(tablero)`: Inicializa y muestra el tablero con el patrón de colores.

`imprimirtablero(tablero)`: Imprime el estado actual del tablero.

`evaluarSN()`: Evalúa una entrada del usuario que puede ser "s" o "n".

`agregarPiezas(tablero, pNegras, pBlancas)`: Permite al usuario agregar piezas en el tablero.

`agregarCaballo(tablero, pNegras, pBlancas)`: Permite al usuario agregar un caballo y evalúa sus movimientos.

`validarCasilla(fila, columna, vf)`: Valida si las casillas posibles de los movimientos del caballo están dentro del tablero.

`evaluarCaballo(tablero, fila, columna, pNegras, pBlancas, pieza)`: Evalúa los movimientos posibles del caballo.

## Diagramas de flujo



