



# UNIVERSIDAD DE ANTIOQUIA

---

## Facultad de Ingeniería

Etapas 1: Análisis y Diseño

Informática II

Parcial II

Mario Esteban Estrada Gonzalez

CC: 1233191679

Andrés Felipe Agudelo Zapata

TI: 1017926361

Docente

Aníbal José Guerra Soler

2023-2

UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERÍA

SEDE MEDELLÍN

Octubre 2023

Medellín, Colombia

1.

## 2. Contextualización:

El juego a realizar se desarrolla en una matriz cuadrada de tamaño inicial 8x8 (pero que puede cambiar si así se decide) en la que por turnos 2 jugadores deberán de colocar fichas en las casillas disponibles, el objetivo es que al finalizar el juego se debe de contar con mayor cantidad de fichas que el adversario.

Al inicio del juego se encuentran cuatro fichas en el centro de la matriz, dos negras y dos blancas, un jugador puede colocar una ficha en caso de que se pueda generar un encierro de tipo sandwich a una o más fichas enemigas, un encierro tipo sandwich significa encerrar una o más fichas del color contrario entre dos fichas de tu color, este encierro se puede dar en cualquiera de las diagonales, de forma horizontal o de forma vertical. Si se genera un encierro de tipo sándwich, todas las fichas atrapadas cambian al color del que las encerró.

Si un jugador no cuenta con ninguna posición para generar un encierro tipo sandwich, entonces se pierde el turno, el juego debe de finalizar si ninguno de los dos jugadores cuenta con movimientos posibles o si se llena la matriz y se debe de guardar el registro de los jugadores, quien gano y con qué puntaje.

## 3. Análisis:

El análisis del problema se generó con la visión de “divide y vencerás”, de manera que del problema principal se desglosan problemas más específicos que al darles solución encaminan al desarrollo final del proyecto, en este orden de ideas nacen las siguientes preguntas.

### *¿Qué estructura de datos se usará?*

Dentro de las dinámicas del juego se puede observar que está cimentado en un tablero de 8x8 filas y columnas respectivamente, las cuales se pueden representar a través de una matriz de la mismas dimensiones, puesto que dentro de las limitaciones presentadas no se puede hacer uso de contenedores STL se opta por usar punteros dobles a arreglos de tamaño 8 donde cada posición de la matriz corresponde al espacio que una ficha puede ocupar, para facilitar el funcionamiento se tendrán tres tipos de identificadores que son:

COLOR FICHA	SÍMBOLO
Negro	-
Blanco	*
Vacío	o

### ***¿Qué clases se crearán?***

Para representar la lógica del juego se tienen 3 clases que el correcto funcionamiento las clases son:

#### ***Jugador:***

Estos objetos representan a cada uno de los jugadores que participan en la partida

#### ***Tablero:***

Contiene las reglas y el funcionamiento general del juego, además de permitir la interacción de los jugadores.

#### ***Juego:***

Permite el acceso al historial de las partidas, además de dar el inicio y el fin de las mismas.

### ***¿Cómo funciona el juego?***

El funcionamiento interno del tablero se lleva a cabo en 3 partes, la primera es la creación de la matriz con las 4 fichas iniciales como se ve en la siguiente figura.

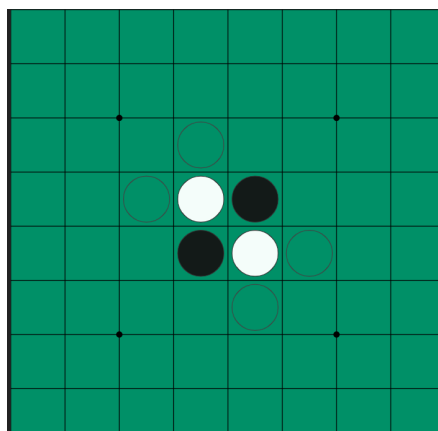


Fig 1

La segunda parte del funcionamiento requiere determinar cuáles son los movimientos posibles que el siguiente jugador puede realizar, para esto se crea la función `Fichas_alredor()` que determina si una posición vacía dentro del tablero, la cual será ocupada por una ficha de cierto color, está rodeada por fichas de color contrario.

A continuación a través de la función `Sandwich_posible()` se establece si al colocar una ficha en esa posición crea un efecto sandwich, finalmente con `Insertar_Ficha()` introduce una ficha en el tablero efectuando las validaciones y cambios de color respectivos

Como parte final, una vez todos los espacios del tablero hayan sido ocupados por fichas, se establecerá como finalizada la partida y se le otorgara a los jugadores la opción de volver a jugar o finalizar, sin importar cuál sea la respuesta dada, cada partida queda registrada en un archivo de texto externo al programa.

#### 4. Diseño de la solución:

Clase Juego	Clase Jugador	Clase Tablero
<b>Atributos:</b> nombre, jugadores actuales, fecha y hora, puntaje, nombre, archivo, texto	<b>Atributos:</b> Turno, color, Nombre	<b>Atributos:</b> Matriz, Cantidad fichas negras, Cantidad fichas blancas, Cantidad espacios vacíos
<b>Métodos:</b> <ul style="list-style-type: none"> <li>● Leer archivo</li> <li>● Imprimir registro</li> <li>● Actualizar archivo</li> <li>● Crear tablero</li> <li>● Cerrar juego</li> </ul>	<b>Métodos:</b> <ul style="list-style-type: none"> <li>● Acceder al turno</li> </ul>	<b>Métodos:</b> <ul style="list-style-type: none"> <li>● Posibles movimientos</li> <li>● Buscar alrededor</li> <li>● Check movimiento Sandwich</li> <li>● Insertar nueva ficha</li> <li>● Validación posición</li> <li>● Actualización de matriz</li> <li>● Check fin del juego</li> <li>● Perder turno</li> </ul>