

## Diseño de la solución

- **Matriz**

El diseño de la matriz, para la lógica del juego, será mediante una matriz  $m \times n$ , donde  $m$  y  $n$  los decide el usuario. En cada posición del tablero, tendrá 3 posibles situaciones:

- Nada: 0
- Ficha negra: 1
- Ficha blanca: 2
- Cursor: 20: por ejemplo, si en una posición se encuentra un 22, se sabe que hay una ficha blanca y el cursor (por el último dígito). El 20 para poder manejar en bits, y hacer un tratamiento con ensamblador (ver con más detalle en el método `pintar_estado_actual_tablero`).<sup>1</sup>

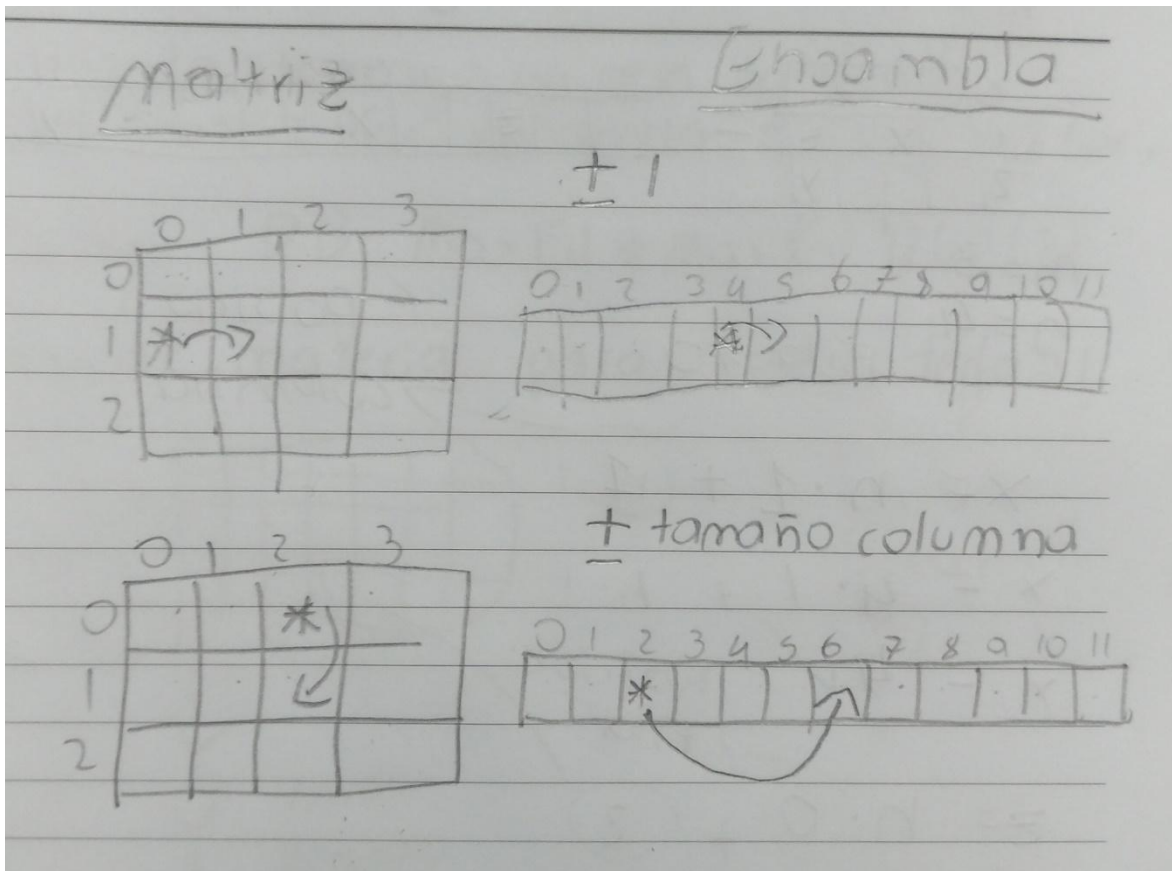
Para **movilizarse**, se hace mediante sumas, por ejemplo, si apretó la tecla derecha/izquierda, tengo que restar la posición actual con un 20, para “quitar” el cursor, sumarle a la posición actual  $\pm 1$  un 20 ( $\pm 1$  por el hecho de que me estoy movilizando hacia la derecha/izquierda como un vector), si fuera el caso de que tengo que moverme hacia abajo/arriba, tengo que sumarle un 20 a la *posición\_actual*  $\pm$  tamaño\_columna. Note que, hay que verificar que, al hacer la suma, no se salga de los límites de la matriz, en dado caso, la posición actual se mantiene. Por lo que, el orden correcto para al presionar una de las flechas para movilizarse es el siguiente:

- Hacer la suma o resta de la posición.
- Verificar que se encuentre dentro de los límites de la matriz (que no se salga).
- Si está dentro de los límites:
  - Reste a la posición actual un 20.
  - Sume a la posición donde se va a movilizar un 20 (acorde a la flecha).

---

<sup>1</sup> Cambio hecho después de realizar esta parte del diseño.

- Mande a imprimir el tablero.



Representación de cómo moverse, la relación de la representación con una matriz y en ensamblador.

**¿Cuándo se sale de los límites?** Como en ensamblador la matriz se maneja como vector, se va a tratar dicha matriz como lo que es realmente. Se maneja por casos:

- Arriba: se resta el tamaño de columnas a la posición actual, teniendo en cuenta que no se puede llegar a un valor negativo.
- Abajo: se suma el tamaño de columnas a la posición actual, teniendo en cuenta el límite, no puede llegar a ser mayor a columnas \* filas.

Por la naturaleza de ensamblador y por el enunciado, los límites para moverse a la derecha e izquierda son los mismos que para arriba y abajo, acorde a su comportamiento:

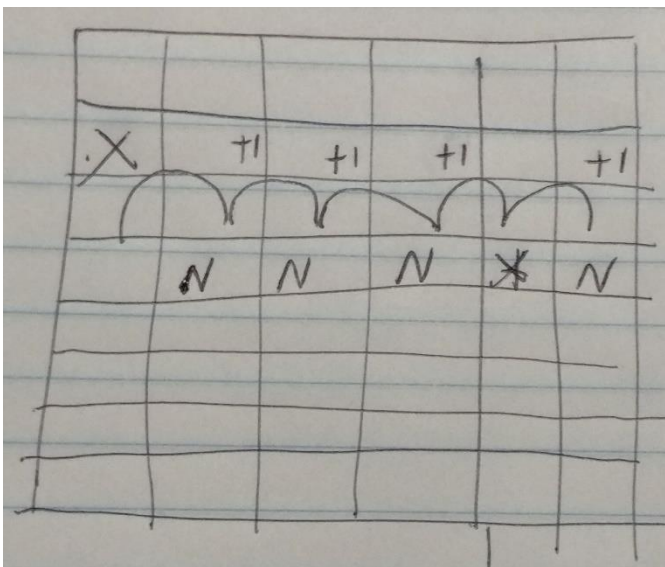
- Derecha: se suma 1 a la posición actual.
- Izquierda: se resta 1 a la posición actual.

Si quiero **poner** una ficha, hay que tener en cuenta los siguientes aspectos: **1)** que la posición actual del cursor esté disponible (con un 10), en caso contrario, no se puede poner la ficha; **2)** el jugador actual: para verificar si tengo que sumar un 1 o 2 a la posición actual y, luego de poner la ficha, **3)** verificar si el jugador actual ganó. Note que al hacer la suma de 1 o 2 no se modifica el estado natural del tablero, por ejemplo, voy a poner una ficha en un lugar disponible, por lo que la posición actual tiene un 10 como valor, pongo una ficha negra, la posición actual valdría 11, si muevo el cursor, dicha posición tendría un 1, por lo tanto, mantengo el estado de que haya una ficha negra pese a que me movilité.

- **Ganador**

Hay un ganador cuando hay una línea de 5 fichas del mismo color. Como cada vez que se ponga una ficha se tiene que verificar si hay un ganador, se empieza desde la posición actual (donde se puso la ficha); desde ahí se recorre a todas las direcciones en busca del patrón ganador (5 fichas del mismo color seguidas), en caso de que se encuentre una posición con un valor distinto al esperado (ficha de otro color o nada), se detiene el proceso y se sigue con otra dirección.

Para hacer este algoritmo más eficiente, se va a seguir esta analogía: al empezar en un punto (siempre en la posición actual del cursor después de haber puesto una ficha), cabe la posibilidad de estar en el patrón ganador, pero haber caído en el medio de la línea, por lo que se va a la dirección en ambas direcciones, de igual manera, contando las fichas, por ejemplo, izquierda-derecha: empiezo a recorrer hacia la izquierda hasta llegar a un punto donde no pueda contar (ficha del otro jugador, casilla vacía, un límite), luego, recorro hacia la derecha (tomando en cuenta que sé dónde empezar por tener el registro de la posición actual del cursor) contando las fichas, ahí determino si hay ganador, o tengo que hacer el mismo proceso con otra dirección, ver dibujo para entender mejor:



Total 4,  $\therefore$  Ganador N.

\* = Posición actual del cursor, tiene una N, porque es después de haber puesto una ficha.  
 X = Me detengo

$$\#Fila = \lfloor \text{Posición actual} \div \text{columnas} \rfloor$$

Viéndolo como Matriz:

$$\text{Límite Izquierdo} = \text{columnas} * \#Fila$$

$$\text{Límite derecho} = \text{Límite izquierdo} + \text{columnas} - 1$$

0	0	1	2	3
1	4	5	6	7
2	8	9	10	11

3x4

$$\#Fila = \lfloor 10 / 4 \rfloor = 2$$

0	0	1	2	3	4
1	5	6	7	8	9

2x5

$$L.I. = 5 * 1$$

$$L.D. = 5 + 5 - 1 = 9$$

(suponiendo que  
esté en Fila 1)

Estrategia necesaria para determinar el ganador en un vector de ensamblador. Para reconocer cuándo detenerse en los límites de la "matriz imaginaria".