1. Implementación en playAgentStudent.m

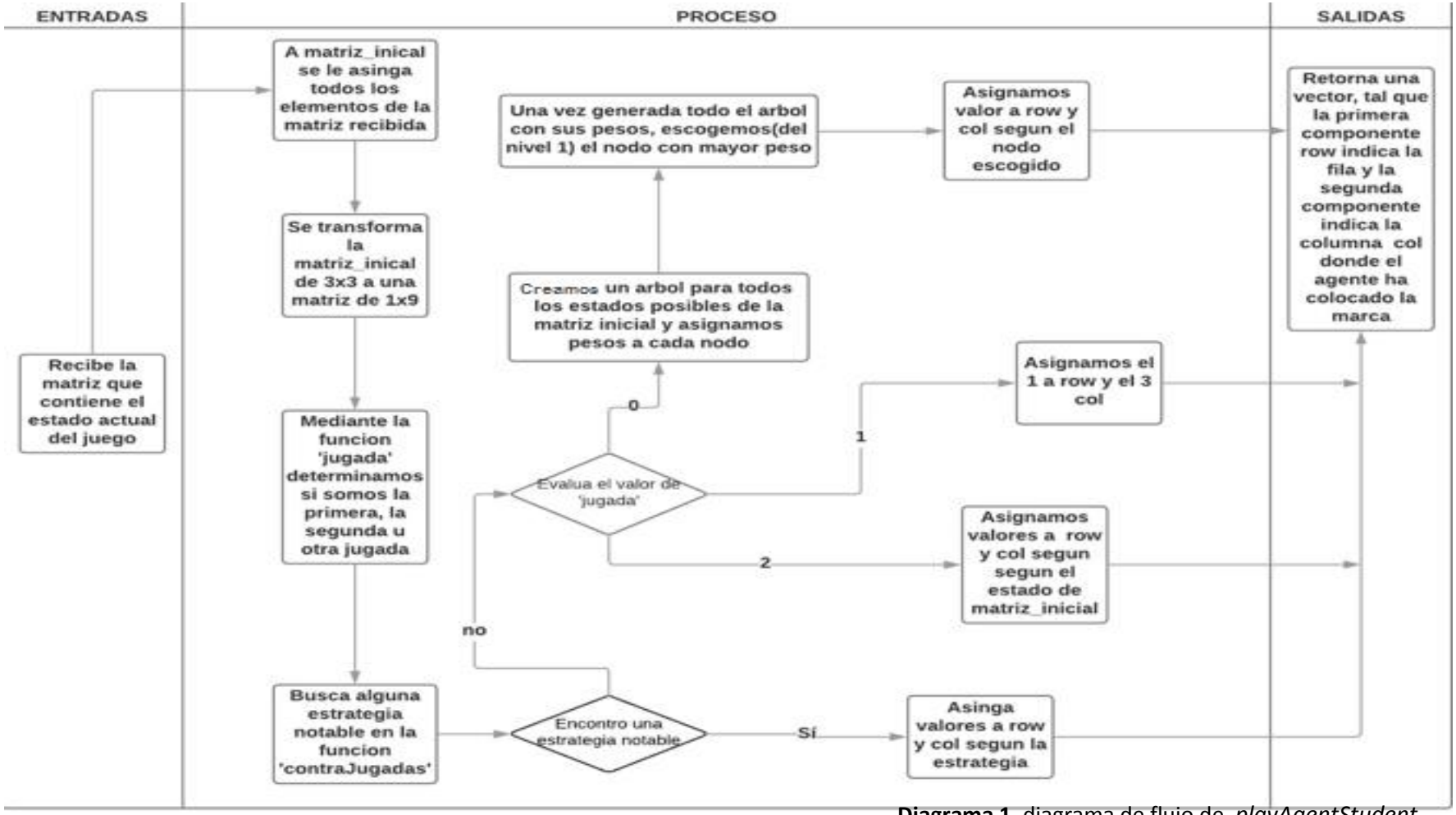


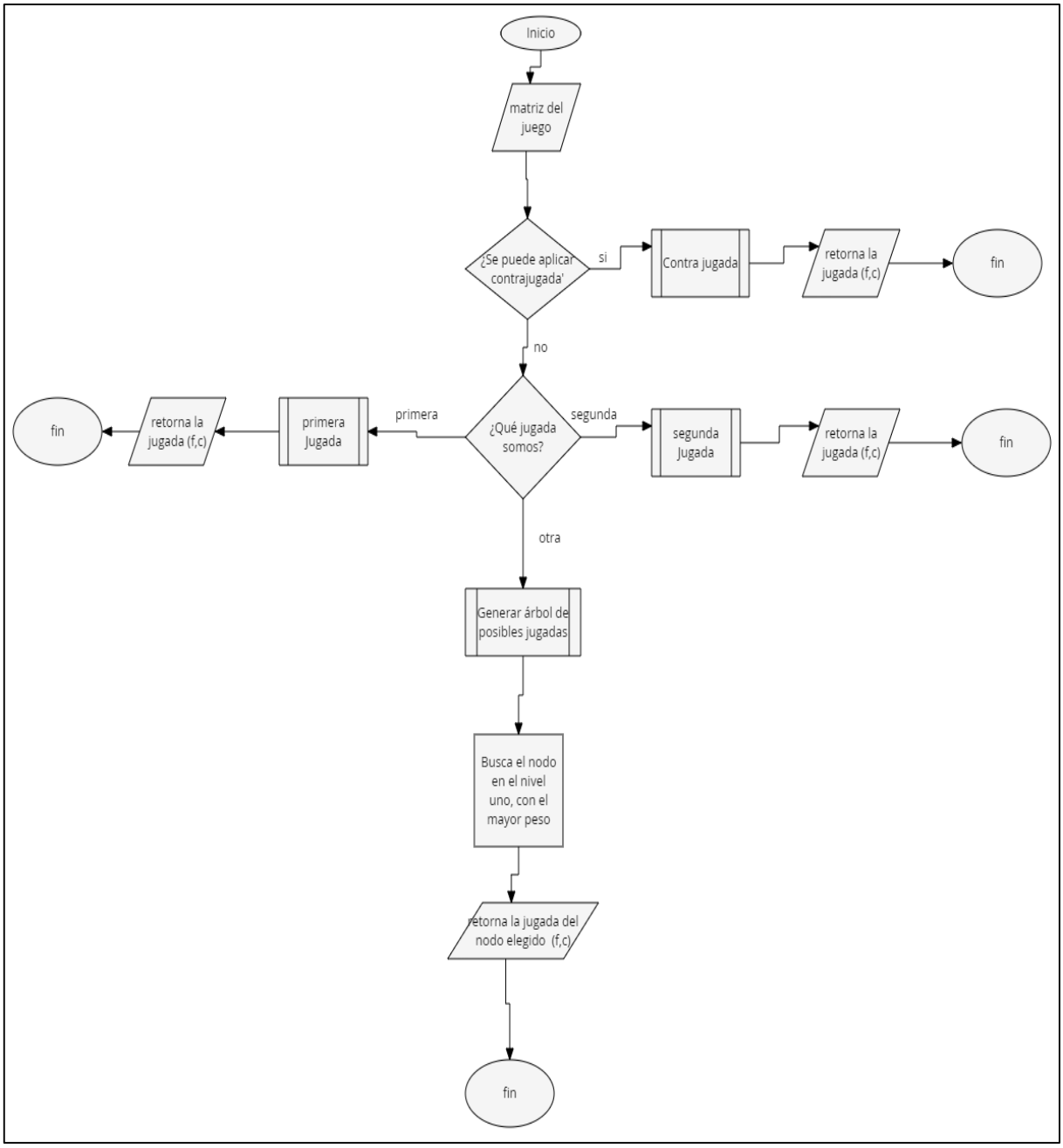
Diagrama 1, diagrama de flujo de *playAgentStudent.*

Escuela Politécnica Nacional

Matemáticas Discretas

Estrategias utilizadas en el agente

El diagrama flujo de *playAgentStudent* también puede ser visto de la siguiente manera:



Escuela Politécnica Nacional

Matemáticas Discretas

Estrategias utilizadas en el agente

Función que genera todas las posibles jugadas del 'tres en raya' a partir de una matriz dada

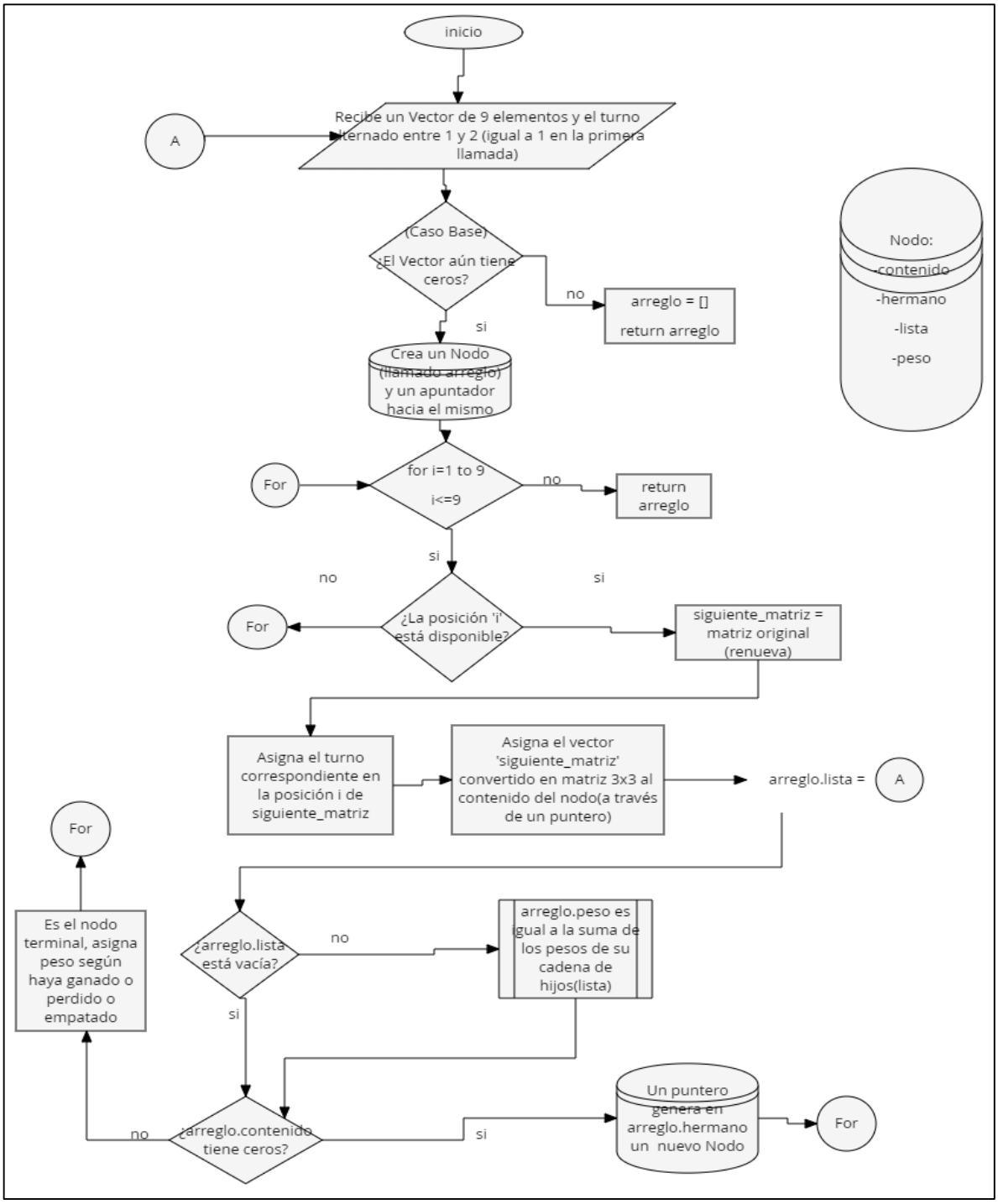
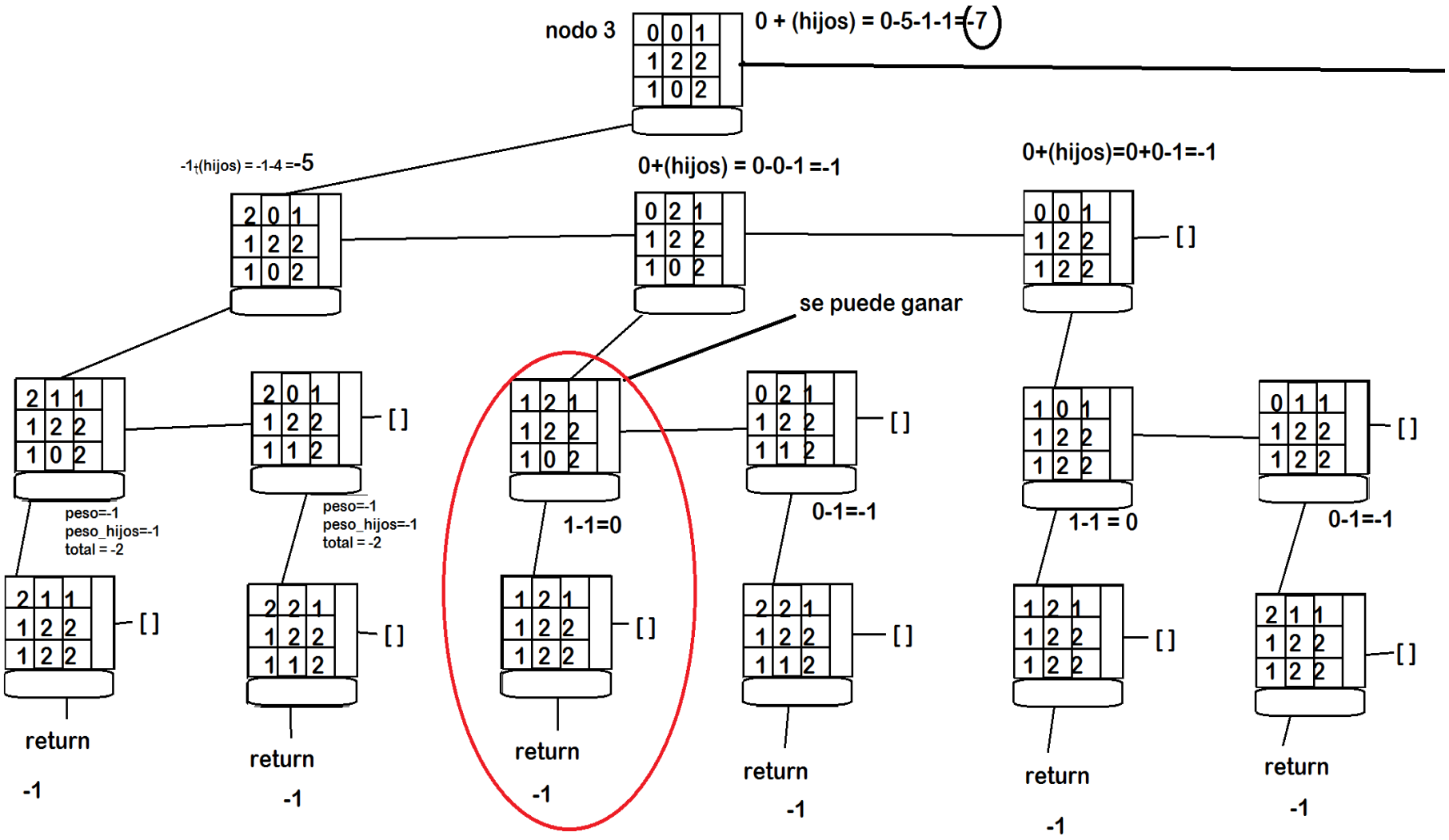


Diagrama 2, Diagrama de flujo de la función recursiva que genera todas las futuras posibles jugadas, (Es parte del diagrama de flujo principal-> Generar árbol de posibles jugadas)

Sin embargo al aplicar esa función, nos genera la siguiente ambigüedad (nos dice que perdimos, cuando en realidad ya ganamos mucho antes), por lo que se optó por corregir y de paso aumentar la eficiencia del algoritmo:



Escuela Politécnica Nacional

Matemáticas Discretas

Estrategias utilizadas en el agente

Función mejorada

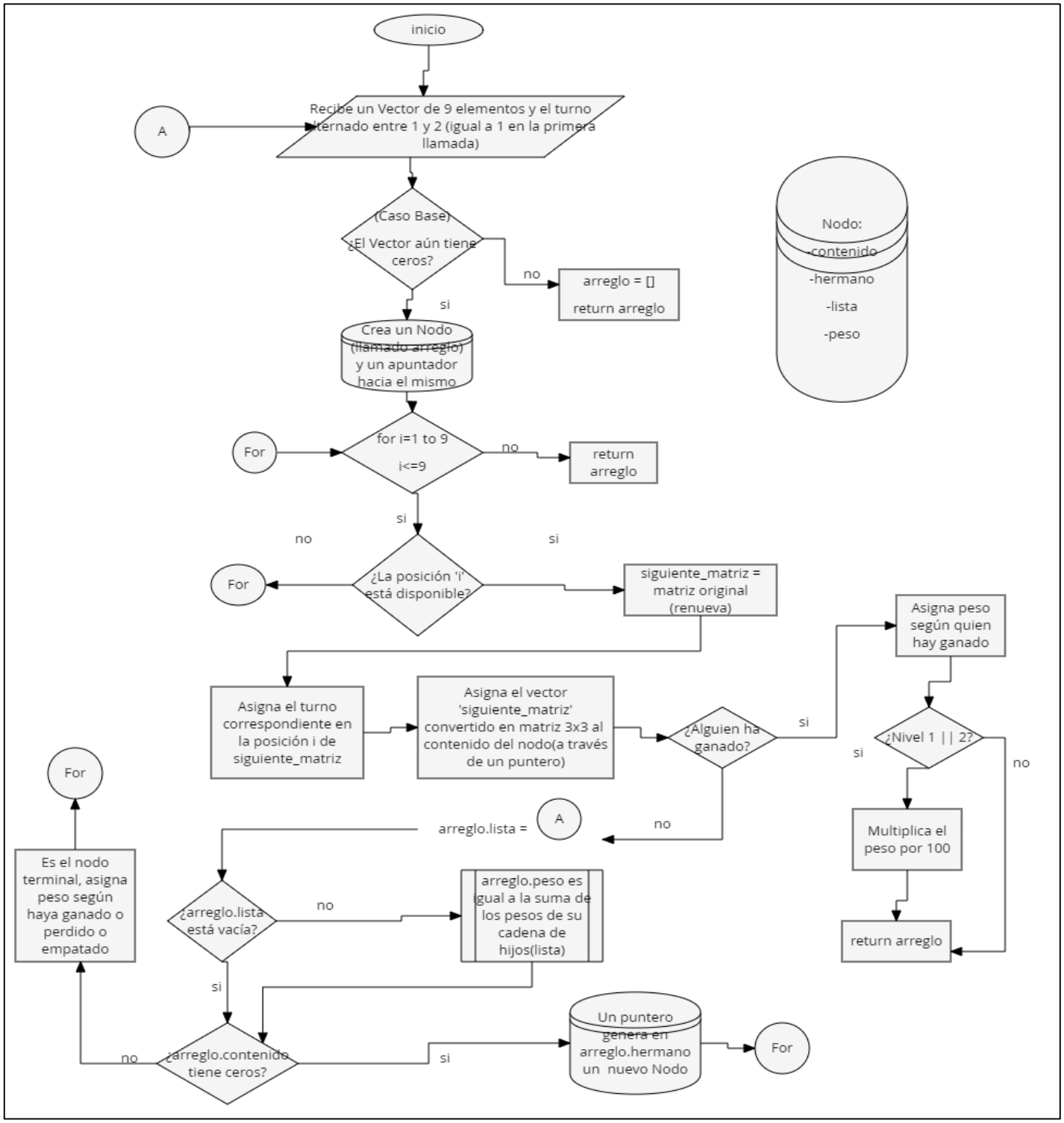
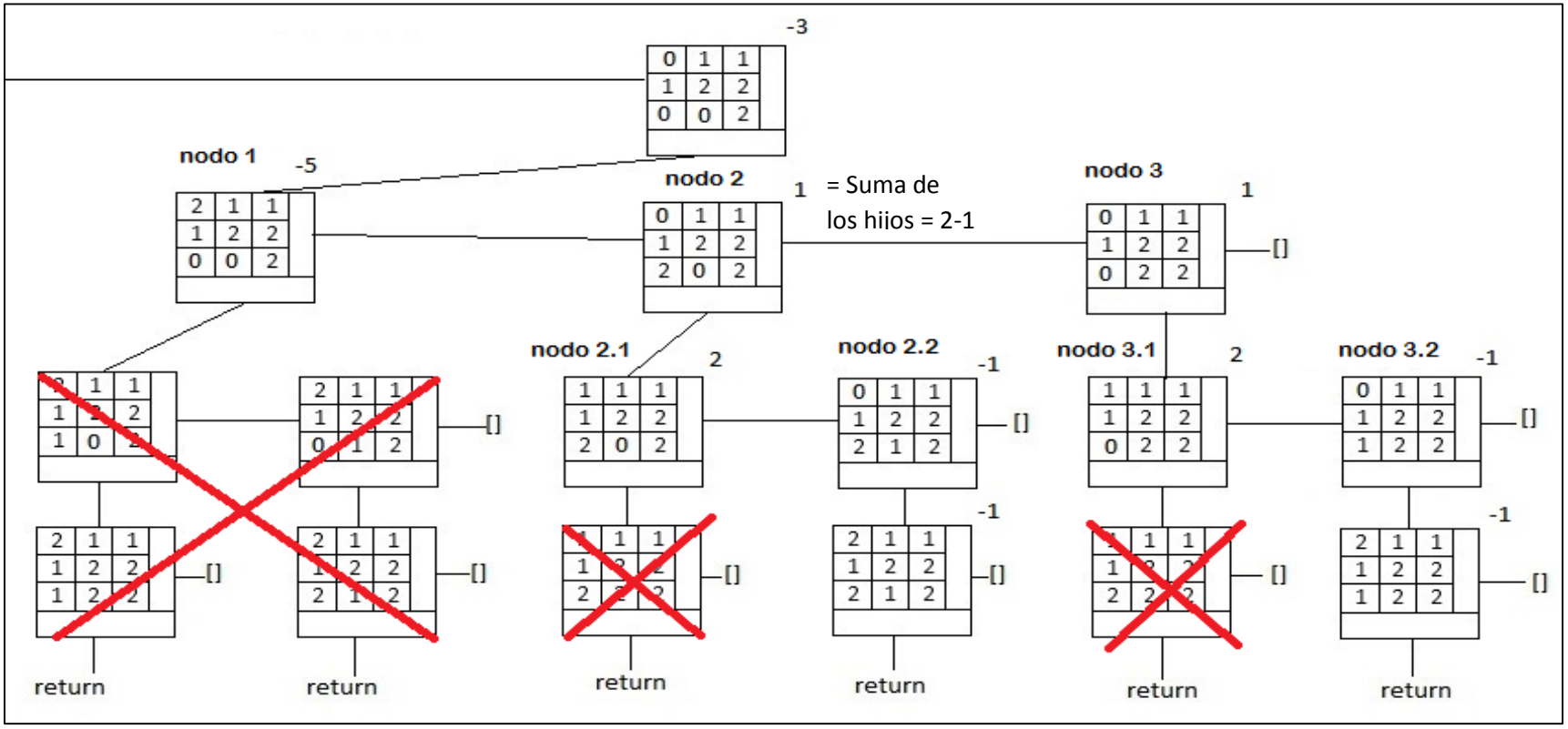


Diagrama 3, diagrama de flujo de la función recursiva mejorada (para ya no generar nodos cuando alguien de los dos ha ganado)

Estrategia en la asignación de pesos

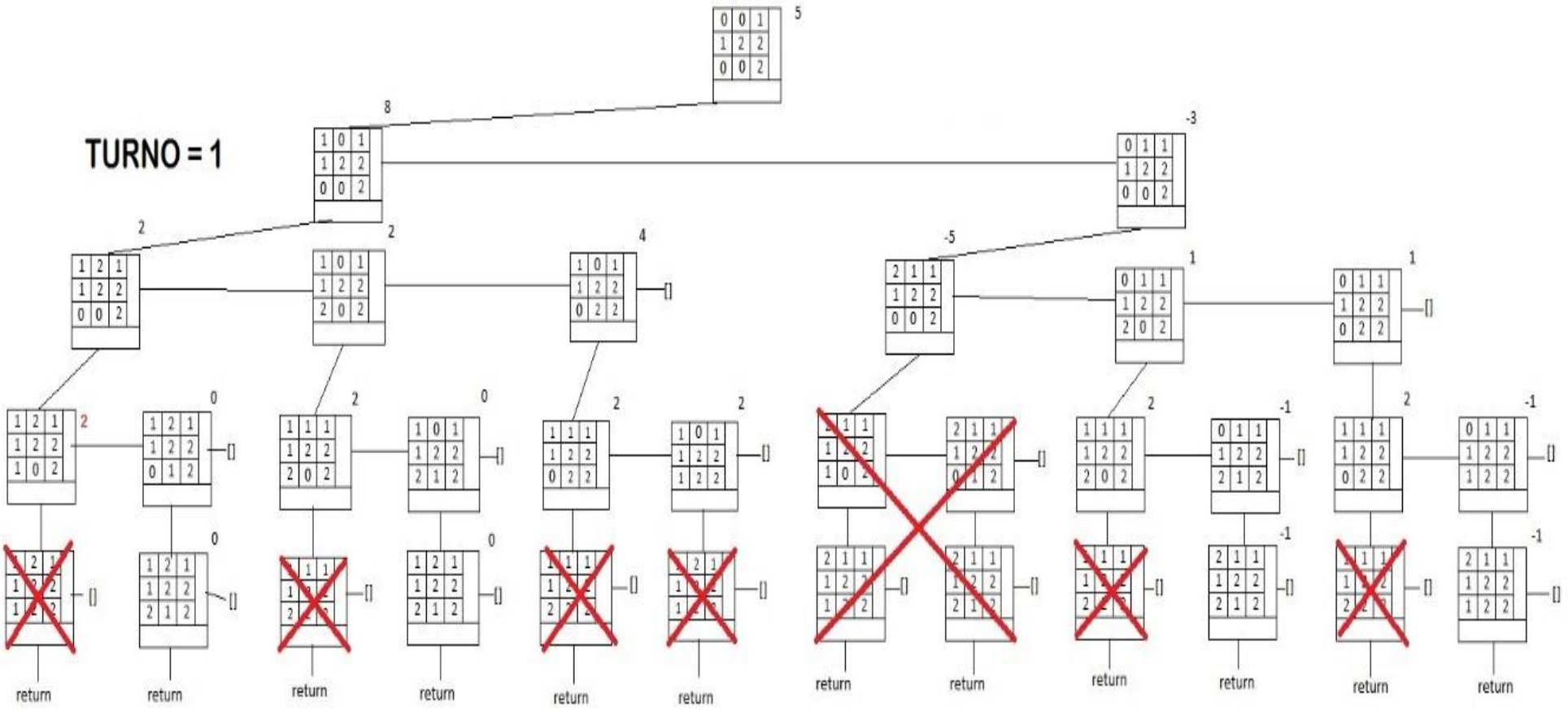


Se asigna un valor positivo, negativo o cero a los nodos terminales según se haya: ganado, perdido o empatado. Los nodos de un nivel superior obtienen su peso sumando todos los pesos de sus hijos del siguiente nivel. Si se realizó un recorte, entonces se le agrega un valor especial, igual a la cantidad de nodos que debió generar.[1]

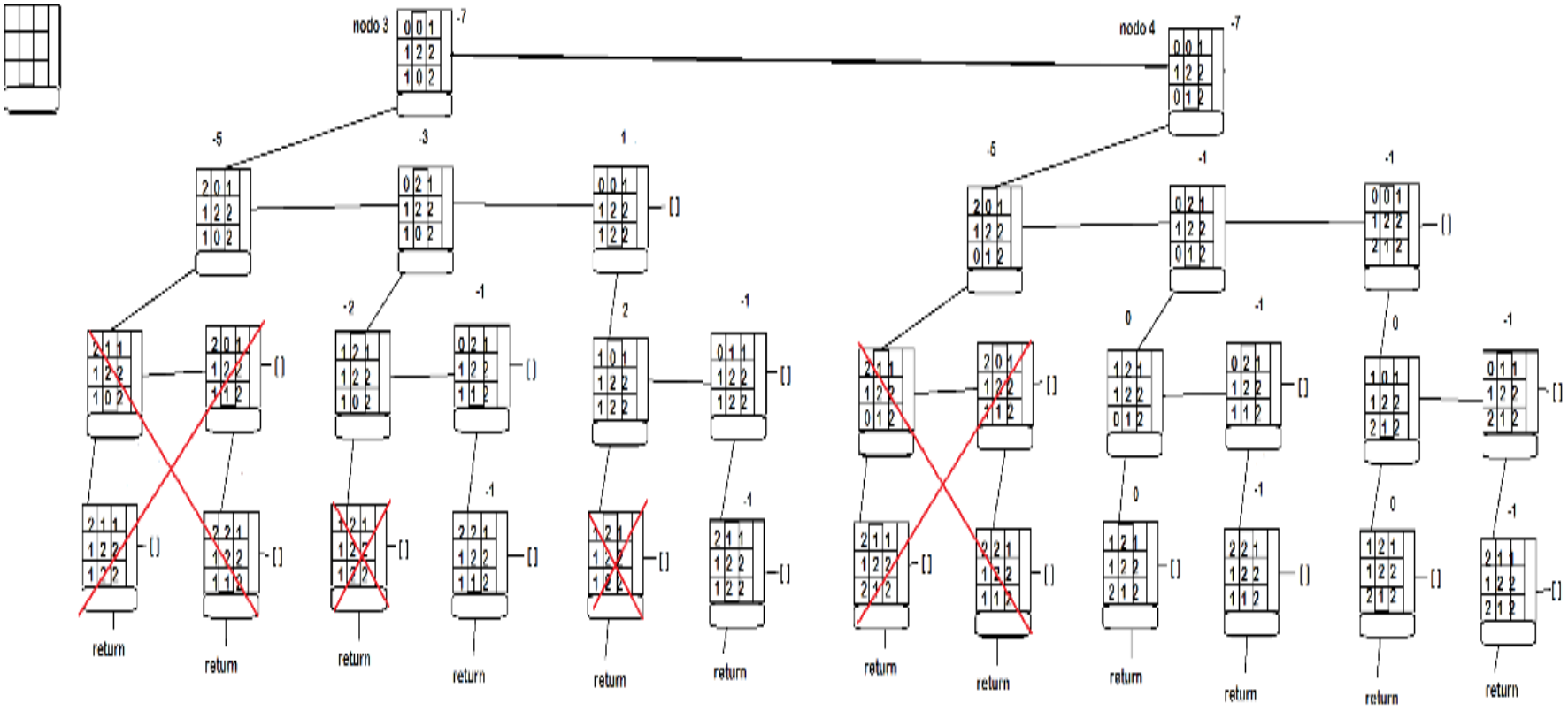
$$estado = -1;\ peso = estado + estado * (cantidad\ de\ nodos\ que\ debió\ generar) = -1 * -1(4) = -5$$

[1] Este proceso se explica particularmente por cada nodo, en el informe. Para esta presentación, estaba restringido usar mucho texto.

Otro ejemplo de cómo se da la asignación de pesos:



Hay 4 posibles jugadas, y los pesos nos indican que el nodo 1 con el peso de 8 es la mejor opción, y tiene razón.



Escuela Politécnica Nacional

Matemáticas Discretas

Estrategias utilizadas en el agente

Cantidad de nodos que genera la matriz con 'n' espacios disponibles

De manera analítica se obtuvo una fórmula para obtener la cantidad de nodos que cada padre generaba (esto es parte fundamental para la estrategia que usamos), para tres espacios disponibles (n=3) es:

$$[1 + (n-1) + (n-1)(n-2)]^2 * n \ (uno)$$

De manera general:

$$[1 + (1)(n-1) + \cdots + (1)(n-1)(n-2) \ldots (n-(n-1))] * n$$

Si desarrollamos la ecuación (uno):

$$= n + n(n-1) + n(n-1)(n-2)$$

$$= \frac{n(n-1)!}{(n-1)!} + \frac{n(n-1)(n-2)!}{(n-2)!} + \frac{n(n-1)(n-2)(n-3)!}{(n-3)!}$$

$$= n!\left(\frac{1}{(n-1)!} + \frac{1}{(n-2)!} + \frac{1}{(n-3)!}\right) = n!\sum_{j=1}^{n}\frac{1}{(n-j)!} \ (dos)$$

[2] Lo que se interpreta como: el primer descendiente (1) más el número de nodos que genera el primer descendiente (n-1) más el número de nodos que genera cada hijo (n-2) multiplicado por los hermanos de ese hijo (n-1) y como solo habíamos considerado el primer descendiente, debemos multiplicar por todos los demás descendientes del primer nivel (n).

Estrategia de la primera y segunda jugada

La ‘Primera’ y ‘Segunda’ jugada nacieron como estrategia para asegurar una victoria o al menos empate y porque resultaba más eficiente crear el árbol a partir de 7 espacios en blancos. Ubicamos manualmente la posición si somos una de las dos.

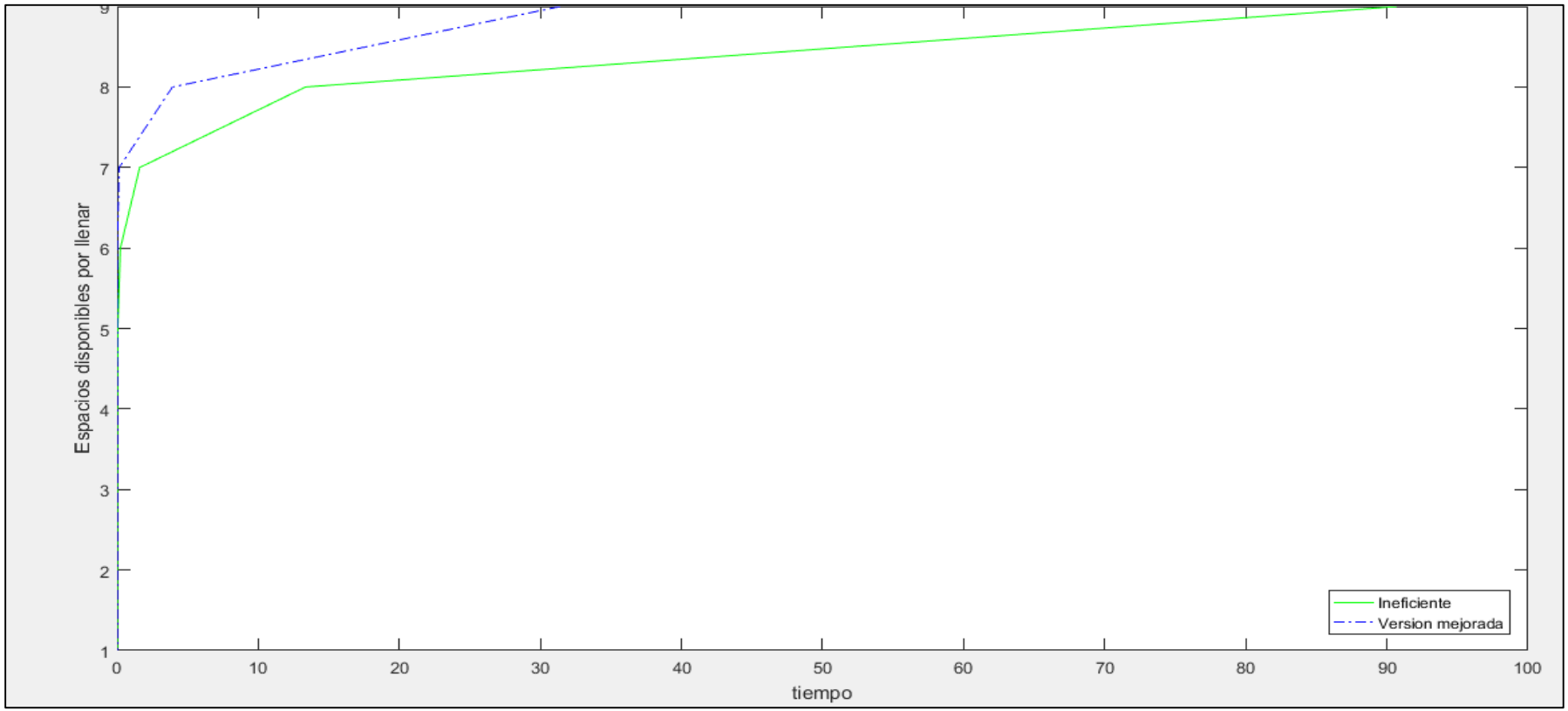


Figura 3, grafica de ‘n’ (cantidad de espacios disponibles por llenar) vs tiempo

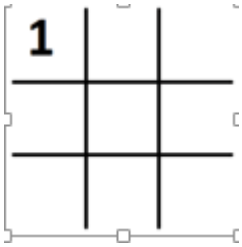
Escuela Politécnica Nacional

Matemáticas Discretas

Estrategias utilizadas en el agente

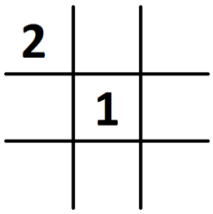
Primera Jugada:

Los jugadores más experimentados colocan la primera jugada en una esquina cada vez que empiezan primero. Esto puede aumentar las probabilidades de victoria o empate.

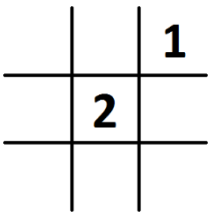


Segunda Jugada:

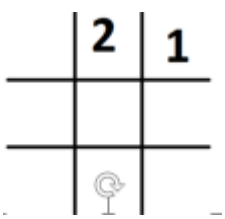
- Si el oponente juega primero y comienza en una esquina, siempre se coloca la segunda jugada en el centro. Con esta estrategia, se estima que el juego terminará seguramente en empate.



- Si el oponente comience colocando en el centro, se coloca la segunda jugada en una esquina.



- Si el oponente coloca la primera jugada en un borde (lado) en lugar de en una esquina o en el centro, hay posibilidad de ganar. Se coloca la segunda jugada en una de las esquinas.



Escuela Politécnica Nacional

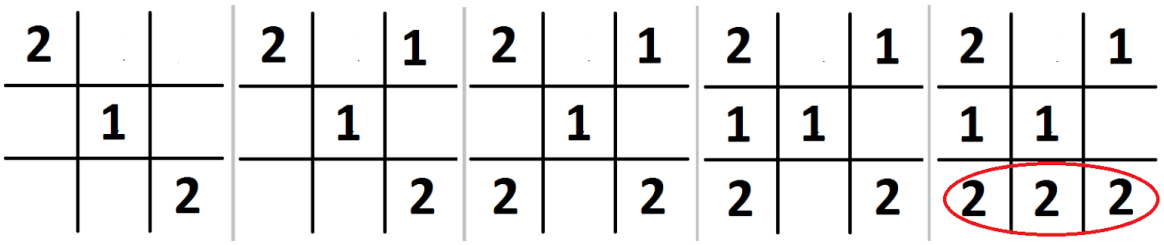
Matemáticas Discretas

Estrategias utilizadas en el agente

Contra-jugadas

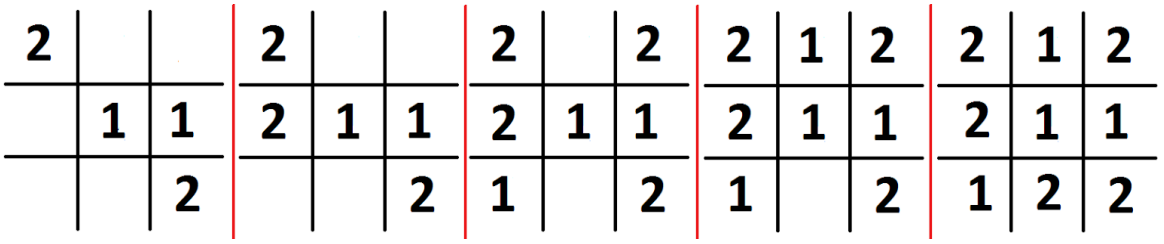
La función implementada “contra Jugadas” sirve para solventar algunas debilidades que pueda tener el algoritmo y también para alimentarlo de “buenas jugadas”. Es decir, si encuentra alguna “jugada notable” en donde la victoria es clara, aplicamos esa jugada para ganar. Por ejemplo:

Supongamos que tenemos esta jugada. Si colocamos un 1 en cualquiera de las esquinas sobrantes perderemos inmediatamente.



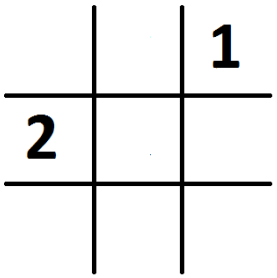
La ‘contra jugada’ está programada para ver casos simétricos de esta jugada con la función `rot90`, y ubicar su jugada en uno de los **lados**.

Ya que si ubicamos en esas posiciones nuestro ‘1’ contrarrestamos esa jugada



Otro caso particular agregado en ‘Contrajugadas’ es el siguiente:

Suponga que tiene la siguiente matriz o algún caso simétrico



La maneara directa para ganar a partir de esa jugada es ubicando un 1 en el centro