

# Algoritmo codicioso: Kevin Acevedo Rodríguez

Teoría  $\Rightarrow$  En cada paso elegimos la opción más beneficiosa sin mirar hacia el futuro.

## Funcionamiento en la Tarea (Tic Tac Toe)

- Inicia el juego usando la instrucción  $\Rightarrow$  (TTT m n)  
m: número de filas  
n: número de columnas  
además  $\begin{cases} X: \text{posición del usuario} \Rightarrow 1 \\ O: \text{posición del algoritmo} \Rightarrow 2 \end{cases}$

①  $\downarrow$

X		

- El usuario coloca su X en una posición libre de la matriz

$\downarrow$

②  $\Rightarrow$  Se llama al algoritmo codicioso y se le pasa como argumento la representación lógica de la matriz:

- matriz: '((1 0 0) (0 0 0) (0 0 0))' este es el conjunto de candidatos

$\downarrow$

③ Ahora por medio de la función de selección se toman solamente las posiciones que tengan 0's:  
• Se debe generar la lista con esas posiciones.

$\downarrow$

④ Con la función de viabilidad se debe evaluar cada uno de los candidatos y su salida será la misma lista de selección pero con sus plus.

$\downarrow$

⑤ Ahora la función objetiva recorre la lista y toma al que tiene mayor plus. Seguidamente se modifica la matriz.



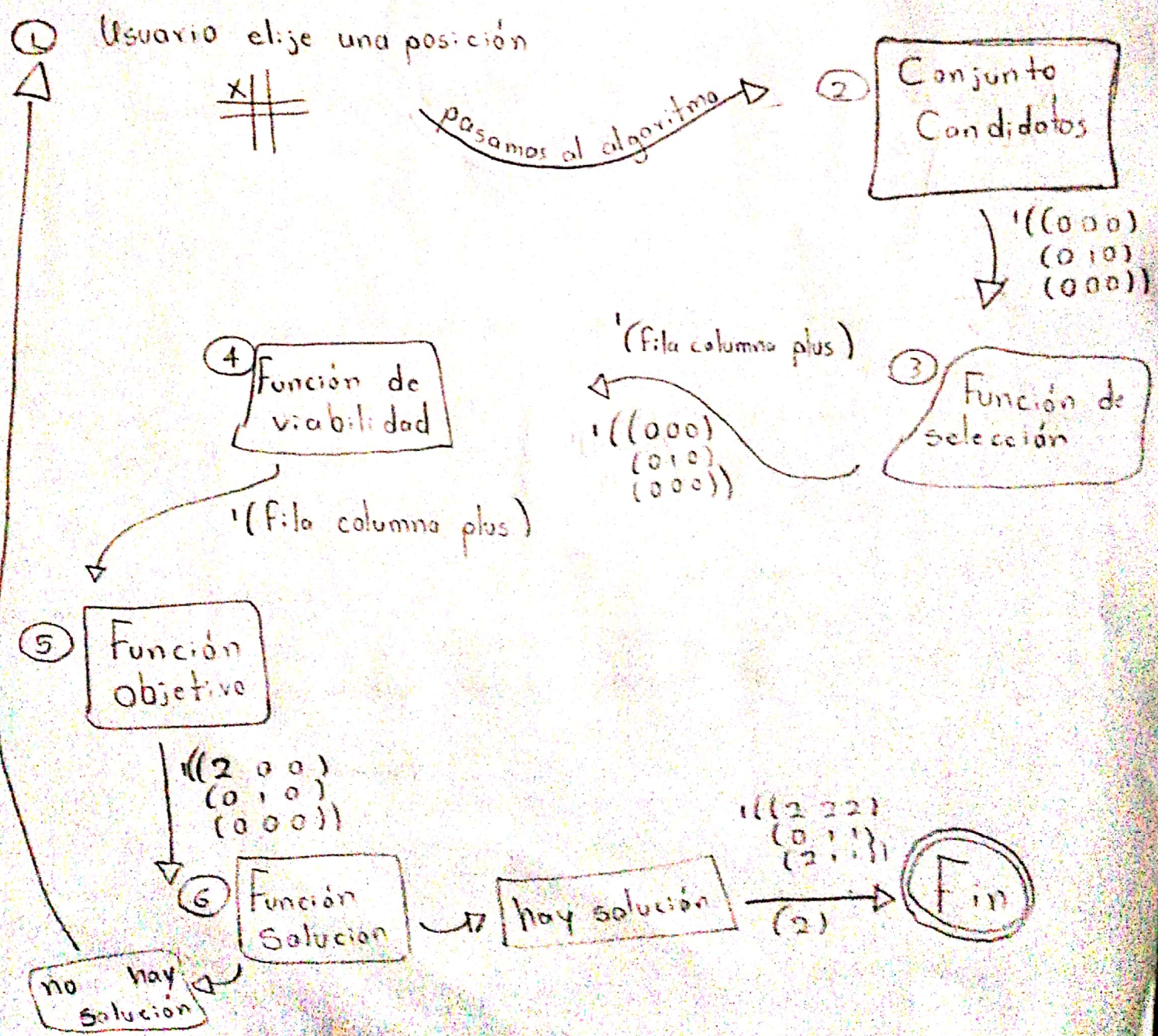
→ ⑥ Usando la función solución se revisa si se ha a una solución. Si hay una solución completa (línea completa) envía un aviso a la interfaz para que presente al ganador.

• Si no hay una solución: se envía la matriz a la interfaz para que pueda mostrarlo



volvemos al paso ① hasta que haya una solución completa

### Esquema del procedimiento:





## Algoritmo codicioso detallado:

Conjunto Candidatos:  $\Rightarrow$ 

X		

 $\Rightarrow$   $\begin{pmatrix} (1\ 0\ 0) \\ (0\ 0\ 0) \\ (0\ 0\ 0) \end{pmatrix}$

### ③ Función selección:

• Recibe:

$\begin{pmatrix} (1\ 0\ 0) \\ (0\ 0\ 0) \\ (0\ 0\ 0) \end{pmatrix}$

• Salida:

$\begin{pmatrix} (1\ 2\ 0) & (1\ 3\ 0) & (2\ 1\ 0) & (2\ 2\ 0) \\ (2\ 3\ 0) & (3\ 1\ 0) & (3\ 2\ 0) & (3\ 3\ 0) \end{pmatrix}$  } esperanza

• notación: (fila columna plus)

• plus: inicialmente en 0  $\rightarrow$  multiplicadores de cada posición

### ④ Función de viabilidad:

• Recibe:

1- matriz  
2- esperanza

• Salida:

1- matriz  
2- vectores solución (fila, columna, diagonales)  
3- esperanza actualizada  
4- vectores solución actualizados

• Salidas específicas:

1- matriz: la matriz actual sin cambio.

2- vectores solución:

$\begin{pmatrix} (1\ 2\ 0) \\ (2\ 1\ 2) \\ (1\ 1\ 1) \end{pmatrix} \Rightarrow$  vectores fila =  $\{(1\ 2\ 0) (2\ 1\ 2) (1\ 1\ 1)\}$   
vectores columna =  $\{(1\ 2\ 1) (2\ 1\ 1) (0\ 2\ 1)\}$   
vectores diagonal =  $\{(1\ 1\ 1)\}$   
vectores diagonal inverse =  $\{(1\ 1\ 0)\}$

Representación en código de los vectores solución:

Vector-solución:  $((\text{fila columna valor}) (\text{fila columna valor}))$



Nota: Se tomarán como vectores solución aquellos que todos sus valores haya un 0 o un 2. Si hay un 1 se elim

### Actualización de la esperanza:

Teniendo:

- La lista esperanza: las posiciones seleccionadas
- Los vectores solución: Filas, columna, diagonal, diagonal<sup>-1</sup>
- Para cada candidato: (fila columna plus)
- Se revisan las intersecciones:
  - $n$  vector Fila  $\Rightarrow +1$
  - $n$  vector Columna  $\Rightarrow +1$
  - $n$  vector Diagonal  $\Rightarrow +1$
  - $n$  vector Diagonal inverso  $\Rightarrow +1$
- Si un vector tiene  $n$  posiciones marcadas con 2's  $\Rightarrow (+1 + n)$  ó  $(x n)$
- ¿Y si llego a tener candidatos con el mismo plus?
  - significa que tengo la misma viabilidad con todos ellos

Luego se va a tener la esperanza actualizada.



## Función objetivo:

Recibe:

- matriz
- esperanza actualizada
- vectores solución actualizados

Salida:

- matriz actualizada
- vectores solución actualizados

### - Actualización de la matriz:

Se llama a la función Modificar-matriz con la posición y el valor.

- Si la esperanza actualizada tiene varios elementos con el mismo plus  $\Rightarrow$  se tomará el primero que aparezca.

## ⑥ Función selección:

Recibe:

- matriz actualizado
- vectores solución actualizados

Salida:

- Caso 1: encontró solución.  
- Notifica a la interfaz
- Caso 2: no hay solución  
- Retorno la matriz actualizada

Viendo el algoritmo como caja negra:

$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow$

Algoritmo  
Codicioso

$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

$\Rightarrow$  "ganador"