

Eder Santiago Eraso Ortega - 22501966
Javier Esteban Bedoya Ruiz - 22501361
Ana Sofia Ledesma Garcia - 2251352
Johan David Vivas Esquivel - 22501603

Enlaces de github:

<https://github.com/javier85bedoya-crypto/PROGRAMING>

https://github.com/JohanVivas/project_games

<https://github.com/santiagoerasso/Programming>

<https://github.com/sofialedesma011-hash/PROGRAMMING-PROYECT-1>

Proyecto #1

Diseñe e implemente una aplicación que permita jugar los siguientes juegos contra la máquina:

- Triqui
- Juego de Memoria

Al inicio se deben mostrar los dos juegos y el usuario debe seleccionar uno de ellos.

- Triqui:

- Se selecciona aleatoriamente quien hace el primer movimiento.
- El usuario ingresa las coordenadas dónde quiere poner su símbolo (usted elige si X o O). Tenga en cuenta que debe indicar si esa posición es válida (puede que esté fuera del cuadro o que ya esté ocupada).
- La máquina generará la posición donde pondrá su símbolo (debe encontrar un espacio libre para poner su símbolo). Esta generación puede ser aleatoria o ustedes pueden definir las reglas que consideren.
- Al final, se debe indicar quién fue el ganador o si hubo un empate.
- Se debe regresar al menú con las opciones de los juegos.

- Juego de memoria:

- Se crea un tablero de 6x5 posiciones para ubicar 15 pares de figuras (ustedes las escogen) de forma aleatoria.
- La máquina genera de forma aleatoria su elección destapando un par de figuras.
- El usuario introduce su elección para destapar un par de figuras. Ustedes definen la mejor manera de hacerlo.
- Cuando se acierte un par, este queda destapado y el jugador que lo logre se anota un punto.
- Cuando no se acierta un par, las figuras se vuelven a cubrir.
- Al final, el ganador será quién haya logrado descubrir más pares.

La aplicación debe estar construida usando funciones.

Ustedes son libres de escoger los elementos gráficos y de jugabilidad que consideren adecuados. También son libres de elegir las soluciones que no se encuentren definidas dentro de las especificaciones.

Entradas:

- Selección de menú:
 - Cadena de caracteres con valores '1', '2' o '3'.
 - Debe ser una de estas opciones; si no, se pide nuevamente.
- Para Triqui:
 - Coordenadas en formato: Ingresando un número de 1 a 9.
 - La posición debe estar vacía para que se pueda colocar el símbolo.
- Para Memoria:
 - Cuatro enteros separados por espacio: fila1, col1, fila2, col2.
 - Restricciones:
 - Filas entre 0 y 5, columnas entre 0 y 4.
 - Ninguna de las posiciones ya debe estar destapada.
 - Las posiciones deben ser distintas.

Salidas:

- Menú principal:
Mensaje: "Bienvenido al menú, seleccioná un juego."
Opciones:
 - Triqui
 - Memoria
 - SalirMensaje de opción inválida: "Opción inválida. Intente nuevamente."
Mensaje de salida: "Gracias por jugar! Saliendo..."
- Triqui:
 - Instrucciones al inicio: "Juego de Triqui: Usted es 'X'. Ingrese coordenadas como 'fila columna' (0-2)."
 - Mostrar tablero actual: matriz 3x3 con símbolos ' ', 'X', 'O'.
 - Mensajes de entrada inválida:
 - "Entrada inválida. Use 'fila columna' (ej: 0 1)."
 - "Posición inválida (fuera de rango o ocupada)."
 - Mensajes de turno máquina: "Turno de la máquina... Máquina coloca en [fila] [columna]."
 - Mensajes de resultado: "Ganaste", "Ganó la máquina", o "Empate."
- Memoria:

- Instrucciones al inicio: "Juego de Memoria: Tablero 6x5 con 15 pares. Ingrese 'fila1 col1 fila2 col2'."
- Mostrar tablero de memoria con filas 0-5, columnas 0-4, usando '*' para cartas cubiertas o el símbolo cuando está destapada.
- Mensajes de entrada inválida para usuario:
 - "Entrada inválida. Use 'fila1 col1 fila2 col2' (ej: 0 1 2 3)."
 - "Posiciones inválidas (fuera de rango, ya reveladas o iguales)."
- Mensajes de turno máquina: "Turno de la máquina... Máquina elige [fila1] [col1] y [fila2] [col2]."
- Mensajes de turno resultado:
 - "¡Par encontrado! +1 punto para [usted/máquina]."
 - "No es un par. Se cubren nuevamente."
- Resultado final: "Juego terminado! Puntos usuario: [n], Puntos máquina: [m]. [Usted gana / Máquina gana / Empate!]"

Restricciones:

- Tablero Triqui:
 - Tamaño 3x3, solo símbolos ' ', 'X', 'O'.
- Tablero Memoria:
 - Tamaño 6x5 (30 celdas) con 15 pares exactos, símbolos no repetidos más de dos veces.
- No se permite salida prematura del juego salvo en el menú.
- IA en Triqui:
 - Debe usar algoritmo minimax o similar para que la máquina juegue inteligentemente.
- IA en Memoria:
 - Genera movimientos aleatorios válidos.

Descomposición en funciones:

- Menú y estructura principal:
 - menu_principal(): Bucle infinito que muestra el menú e invoca los juegos basados en selección.
- Triqui:
 - inicializar_tablero_triqui(): Crea tablero vacío 3x3.
 - imprimir_tablero_triqui(): Muestra el tablero actual.
 - hay_ganador_triqui(): Verifica ganador en filas, columnas y diagonales.
 - tablero_lleno_triqui(): Verifica si tablero está lleno (empate).
 - movimientos_disponibles_triqui(): Lista posiciones vacías.
 - minimax_triqui(): Algoritmo recursivo para que la IA evalúe movimientos (máximiza IA, minimiza usuario).
 - movimiento_ia_triqui(): Elige mejor movimiento usando minimax.

- jugar_triqui(): Lógica principal que inicializa tablero, alterna turnos, valida entradas y verifica condición final.
- Memoria:
 - inicializar_tablero_memoria(): Crea tablero oculto con pares aleatorios y tablero visible con '*'.
 - imprimir_tablero_memoria(): Muestra tablero visible actual.
 - es_par_memoria(): Compara dos posiciones en tablero oculto para validar par.
 - destapar_y_verificar_memoria(): Revela temporalmente dos cartas, verifica si forman par, y actualiza el tablero y puntajes.
 - movimiento_ia_memoria(): La IA escoge aleatoriamente dos posiciones cubiertas para destapar.
 - input_usuario_memoria(): Obtiene y valida la entrada del usuario para destapar dos cartas.
 - memoria_terminada(): Verifica si el tablero visible ya tiene todas las cartas destapadas (fin del juego).
 - jugar_memoria(): Lógica principal que inicializa, alterna turnos, gestiona puntajes, muestra tablero y verifica fin de juego.

Pseudocódigo

Menú principal

INICIO Menu_Principal

MOSTRAR "Bienvenido a la aplicación de juegos contra la máquina!"

MOSTRAR "1. Triqui"

MOSTRAR "2. Juego de Memoria"

MOSTRAR "3. Salir"

LEER opcion (cadena)

SI opcion ES "1" ENTONCES

LLAMAR Jugar_Triqui()

SINO SI opcion ES "2" ENTONCES

LLAMAR Jugar_Memoria()

SINO SI opcion ES "3" ENTONCES

MOSTRAR "Gracias por jugar! nos vemos..."

ROMPER bucle

SINO

MOSTRAR "Opción inválida. Intentá nuevamente."

FIN-SI

FIN-MIENTRAS

FIN

Inicializar_Tablero_Triqui

INICIO Inicializar_Tablero_Triqui

 CREAR tablero como matriz 3x3 con todas las celdas vacías (" ")

 RETORNAR tablero

FIN

Imprimir_Tablero_Triqui

INICIO Imprimir_Tablero_Triqui(tablero)

 MOSTRAR "Tablero actual ("seleccioná una celda del 1 al 9 según esta guía, o 'R' para rendirte)".

 MOSTRAR "1 | 2 | 3"

 MOSTRAR "-----"

 MOSTRAR "4 | 5 | 6"

 MOSTRAR "-----"

 MOSTRAR "7 | 8 | 9"

 MOSTRAR "Estado actual del juego:"

 PARA cada fila i de 0 a 2

 CREAR lista vacía para mostrar

 PARA cada columna j de 0 a 2

 AGREGAR contenido de celda (i,j) ("X", "O" o " ") a la lista

 FIN-PARA

 MOSTRAR lista unida con " | "

 SI i < 2 ENTONCES

 MOSTRAR "-----"

 FIN-SI

 FIN-PARA

FIN

 FIN-SI

 FIN-PARA

FIN

Celda_a_Coordenadas_Triqui

INICIO Celda_a_Coordenadas_Triqui

 RESTAR 1 a celda, esta para usar 0-8 internamente

 CALCULAR fila = celda dividido 3, una división entera

 CALCULAR columna = celda módulo 3

 RETORNAR (fila, columna)

FIN

Coordenadas_a_Celda_Triqui

INICIO Coordinadas_a_Celda_Triqui

CALCULAR número de celda = fila*3 + columna + 1

RETORNAR número de celda

FIN

Hay_Ganador_Triqui

INICIO Hay_Ganador_Triqui

PARA cada fila i de 0 a 2

SI todas las celdas en fila i son simbolo ENTONCES

RETORNAR Verdadero

FIN-SI

SI todas las celdas en columna i son simbolo ENTONCES

RETORNAR Verdadero

FIN-SI

FIN-PARA

SI diagonal principal (0,0), (1,1), (2,2) tiene simbolo ENTONCES

RETORNAR Verdadero

FIN-SI

SI diagonal secundaria (0,2), (1,1), (2,0) tiene simbolo ENTONCES

RETORNAR Verdadero

FIN-SI

RETORNAR Falso

FIN

Tablero_Lleno_Triqui

INICIO Tablero_Lleno_Triqui

PARA cada celda en tablero

SI celda es vacía (" ") ENTONCES

RETORNAR Falso

FIN-SI

FIN-PARA

RETORNAR Verdadero

FIN

Movimientos_Disponibles_Triqui

INICIO Movimientos_Disponibles_Triqui

CREAR lista vacía de movimientos

PARA cada fila i de 0 a 2

PARA cada columna j de 0 a 2

SI celda (i,j) es vacía (" ") ENTONCES

AGREGAR (i,j) a la lista

FIN-SI

FIN-PARA

FIN-PARA

RETORNAR lista de movimientos

FIN

Minimax_Triqui

INICIO Minimax_Triqui

SI Hay_Ganador_Triqui(tablero, "O") ENTONCES

RETORNAR 10 - profundidad

FIN-SI

SI Hay_Ganador_Triqui(tablero, "X") ENTONCES

RETORNAR profundidad - 10

FIN-SI

SI Tablero_Lleno_Triqui(tablero) ENTONCES

RETORNAR 0

FIN-SI

SI es_maximizador ENTONCES

ASIGNAR max_eval = -infinito

PARA cada movimiento en Movimientos_Disponibles_Triqui(tablero)

COLOCAR "O" en movimiento

CALCULAR eval = Minimax_Triqui(tablero, profundidad + 1, Falso)

DESHACER movimiento (volver a " ")

SI eval > max_eval ENTONCES

ASIGNAR max_eval = eval

FIN-SI

FIN-PARA

RETORNAR max_eval

SINO

ASIGNAR min_eval = infinito

PARA cada movimiento en Movimientos_Disponibles_Triqui(tablero)

COLOCAR "X" en movimiento

CALCULAR eval = Minimax_Triqui(tablero, profundidad + 1, Verdadero)


```
    DESHACER movimiento (volver a " ")

    SI eval < min_eval ENTONCES

        ASIGNAR min_eval = eval

    FIN-SI

FIN-PARA

RETORNAR min_eval

FIN-SI

FIN
```

Movimiento_IA_Triqui

```
INICIO Movimiento_IA_Triqui(tablero)

    ASIGNAR mejor_puntaje = -infinito

    ASIGNAR mejor_movimiento = Ninguno

    PARA cada movimiento en Movimientos_Disponibles_Triqui(tablero)

        COLOCAR "O" en movimiento

        CALCULAR puntaje = Minimax_Triqui(tablero, 0, Falso)

        DESHACER movimiento (volver a " ")

        SI puntaje > mejor_puntaje ENTONCES

            ASIGNAR mejor_puntaje = puntaje

            ASIGNAR mejor_movimiento = movimiento

        FIN-SI

    FIN-PARA

    SI Generar_numero_aleatorio(0, 100) < 20 ENTONCES // 20% de probabilidad

        ASIGNAR lista_movimientos = Movimientos_Disponibles_Triqui(tablero)
```

```
ELEGIR aleatoriamente un movimiento de lista_movimientos  
RETORNAR movimiento_aleatorio  
  
SINO  
  
RETORNAR mejor_movimiento  
  
FIN-SI  
  
FIN
```

Jugar_Triqui

```
INICIO Jugar_Triqui  
  
ASIGNAR tablero = Inicializar_Tablero_Triqui()  
  
ASIGNAR usuario = "X", IA = "O"  
  
ELEGIR aleatoriamente turno_usuario (Verdadero o Falso)  
  
MOSTRAR "Juego de tríqui"  
  
MOSTRAR "Sos 'X'. Seleccioná una celda del 1 al 9 según la guía, o 'R' para rendirte".  
  
LLAMAR Imprimir_Tablero_Triqui(tablero)  
  
MIENTRAS Verdadero  
  
SI turno_usuario ENTONCES  
  
MIENTRAS Verdadero  
  
LEER entrada (cadena)  
  
SI entrada ES "R" ENTONCES  
  
MOSTRAR "Te rendiste. Volvés al menú principal."  
  
ROMPER bucle principal (volver al menú)  
  
FIN-SI
```

INTENTAR

CONVERTIR entrada a número entero como celda

SI celda está entre 1 y 9 ENTONCES

CALCULAR (fila, columna) = Celda_a_Coordenadas_Triqui(celda)

ENTONCES
SI fila y columna están en rango (0-2) Y celda (fila, columna) es vacía

COLOCAR "X" en (fila, columna)

ROMPER bucle interno

SINO

MOSTRAR "Celda inválida, ya está ocupada o fuera de rango. Intentá de nuevo.".

FIN-SI

SINO

MOSTRAR "Número inválido. Usá un número del 1 al 9, o 'R' para rendirte.".

FIN-SI

SINO

MOSTRAR "Entrada inválida. Ingresá un número, ej: 5, o 'R' para rendirte.".

FIN-INTENTAR

FIN-MIENTRAS

SINO

MOSTRAR "Turno de la máquina..."

ASIGNAR movimiento = Movimiento_IA_Triqui(tablero)

SI movimiento no es Ninguno ENTONCES

COLOCAR "O" en movimiento

CALCULAR número_celda = Coordenadas_a_Celda_Triqui(movimiento)

MOSTRAR "Máquina coloca en celda [número_celda]"

FIN-SI

FIN-SI

LLAMAR Imprimir_Tablero_Triqui(tablero)

SI Hay_Ganador_Triqui(tablero, "X") ENTONCES

MOSTRAR "¡ganaste vos!"

ROMPER bucle

SINO SI Hay_Ganador_Triqui(tablero, "O") ENTONCES

MOSTRAR "¡La máquina ganó!"

ROMPER bucle

SINO SI Tablero_Lleno_Triqui(tablero) ENTONCES

MOSTRAR "¡Empate!"

ROMPER bucle

FIN-SI

CAMBIAR turno_usuario = no turno_usuario

FIN-MIENTRAS

FIN

Memoria

Inicializar_Tablero_Memoria

INICIO Inicializar_Tablero_Memoria

CREAR lista de 15 frutas = ['🍎', '🍌', '🍒', '🍇', '🍉', '🍓', '🍑', '🍍', '🥝', '🍋', '🥥', '🍅', '🥑', '🍆', '🥕']

CREAR pares = lista de frutas duplicada (30 elementos)

MEZCLAR pares aleatoriamente

CREAR tablero_oculto 6x5 dividiendo pares en filas

CREAR tablero_visible 6x5 con "*" en todas las celdas

RETORNAR tablero_oculto, tablero_visible

FIN

Imprimir_Tablero_Memoria

INICIO Imprimir_Tablero_Memoria(tablero_visible)

MOSTRAR "seleccioná dos celdas del 1 al 30 donde haya *, o 'R' para rendirte".

PARA cada fila i de 0 a 5

CREAR lista vacía para mostrar

PARA cada columna j de 0 a 4

SI celda (i,j) es "*" ENTONCES

CALCULAR número de celda = $i*5 + j + 1$ (1 a 30)

AGREGAR número de celda a la lista

SINO

AGREGAR fruta de la celda a la lista

FIN-SI

FIN-PARA

MOSTRAR "Fila [i]: [lista unida con espacios]"

FIN-PARA

FIN

Celda_a_Coordenadas_Memoria

INICIO Celda_a_Coordenadas_Memoria(celda)

 RESTAR 1 a celda (para usar 0-29 internamente)

 CALCULAR fila = celda dividido 5 (división entera)

 CALCULAR columna = celda módulo 5

 RETORNAR (fila, columna)

FIN

Coordenadas_a_Celda_Memoria

INICIO Coordenadas_a_Celda_Memoria(fila, columna)

 CALCULAR número de celda = fila*5 + columna + 1

 RETORNAR número de celda

FIN

Es_Par_Memoria

INICIO Es_Par_Memoria(tablero_oculto, pos1, pos2)

 OBTENER (fila1, col1) = pos1

 OBTENER (fila2, col2) = pos2

 SI fruta en (fila1, col1) es igual a fruta en (fila2, col2) Y posiciones son diferentes
 ENTONCES

 RETORNAR Verdadero

 SINO

 RETORNAR Falso

FIN-SI

FIN

Destapar_y_Verificar_Memoria

INICIO Destapar_y_Verificar_Memoria

OBTENER (fila1, col1) = pos1

OBTENER (fila2, col2) = pos2

ASIGNAR tablero_visible[fila1][col1] = fruta de tablero_oculto[fila1][col1]

ASIGNAR tablero_visible[fila2][col2] = fruta de tablero_oculto[fila2][col2]

LLAMAR Imprimir_Tablero_Memoria(tablero_visible)

ESPERAR 2 segundos

SI Es_Par_Memoria(tablero_oculto, pos1, pos2) ENTONCES

RETORNAR Verdadero

SINO

ASIGNAR tablero_visible[fila1][col1] = "*"

ASIGNAR tablero_visible[fila2][col2] = "*"

RETORNAR Falso

FIN-SI

FIN

Movimiento_IA_Memoria

INICIO Movimiento_IA_Memoria

CREAR lista de posiciones con "*" (celdas cubiertas)

SI lista tiene menos de 2 posiciones ENTONCES

RETORNAR Ninguno, Ninguno

FIN-SI

ELEGIR pos1 aleatoriamente de la lista

REMOVER pos1 de la lista

ELEGIR pos2 aleatoriamente de la lista

RETORNAR pos1, pos2

FIN.

Input_Usuario_Memoria

INICIO Input_Usuario_Memoria(tablero_visible)

MIENTRAS Verdadero

LEER entrada (cadena)

SI entrada ES "R" ENTONCES

MOSTRAR "Te rendiste. Volvés al menú principal."

ROMPER bucle principal (volver al menú)

FIN-SI

INTENTAR

DIVIDIR entrada en dos números (celda1, celda2)

SI celda1 y celda2 están entre 1 y 30 Y son diferentes ENTONCES

CALCULAR pos1 = Celda_a_Coordenadas_Memoria(celda1)

CALCULAR pos2 = Celda_a_Coordenadas_Memoria(celda2)

SI tablero_visible[pos1] es "*" Y tablero_visible[pos2] es "*" ENTONCES

RETORNAR pos1, pos2

SINO

MOSTRAR "Celdas inválidas, ya están destapadas. Intentá de nuevo."

FIN-SI

SINO

MOSTRAR "Números inválidos. Usá dos números del 1 al 30, distintos, o 'R' para rendirte."

FIN-SI

SINO

MOSTRAR "Entrada inválida. Ingresá dos números, ej: 1 2, o 'R' para rendirte."

FIN-INTENTAR

FIN-MIENTRAS

FIN

Memoria_Terminada

INICIO Memoria_Terminada(tablero_visible)

PARA cada celda en tablero_visible

SI celda es "*" ENTONCES

RETORNAR Falso

FIN-SI

FIN-PARA

RETORNAR Verdadero

FIN

Jugar_Memoria

INICIO Jugar_Memoria

ASIGNAR (tablero_oculto, tablero_visible) = Inicializar_Tablero_Memoria()

ASIGNAR puntos_usuario = 0, puntos_ia = 0

ELEGIR aleatoriamente turno_usuario (Verdadero o Falso)

MOSTRAR "--- Juego de Memoria ---"

MOSTRAR "Elija dos números de celda (1-30) para destapar posiciones, o 'R' para rendirse."

LLAMAR Imprimir_Tablero_Memoria(tablero_visible)

MIENTRAS NO Memoria_Terminada(tablero_visible)

SI turno_usuario ENTONCES

ASIGNAR (pos1, pos2) = Input_Usuario_Memoria(tablero_visible) // Aquí se maneja rendirse si "R"

SI Destapar_y_Verificar_Memoria(tablero_oculto, tablero_visible, pos1, pos2)
ENTONCES

SUMAR 1 a puntos_usuario

MOSTRAR "¡Par encontrado! +1 punto para usted."

SINO

MOSTRAR "No es un par. Se cubren nuevamente."

LLAMAR Imprimir_Tablero_Memoria(tablero_visible)

FIN-SI

SINO

MOSTRAR "Turno de la máquina..."

ASIGNAR (pos1, pos2) = Movimiento_IA_Memoria(tablero_visible)

SI pos1 y pos2 no son Ninguno ENTONCES

CALCULAR celda1 = Coordinadas_a_Celda_Memoria(pos1)

CALCULAR celda2 = Coordinadas_a_Celda_Memoria(pos2)

MOSTRAR "Máquina elige celdas [celda1] y [celda2]"

SI Destapar_y_Verificar_Memoria(tablero_oculto, tablero_visible, pos1, pos2)
ENTONCES

SUMAR 1 a puntos_ia

MOSTRAR "¡Par encontrado! +1 punto para la máquina."

SINO

MOSTRAR "No es un par. Se cubren nuevamente."

LLAMAR Imprimir_Tablero_Memoria(tablero_visible)

FIN-SI

FIN-SI

FIN-SI

CAMBIAR turno_usuario = no turno_usuario

FIN-MIENTRAS

MOSTRAR "Juego terminado!"

MOSTRAR "Puntos usuario: [puntos_usuario]"

MOSTRAR "Puntos máquina: [puntos_ia]"

SI puntos_usuario > puntos_ia ENTONCES

MOSTRAR "¡Usted gana!"

SINO SI puntos_ia > puntos_usuario ENTONCES

MOSTRAR "¡La máquina gana!"

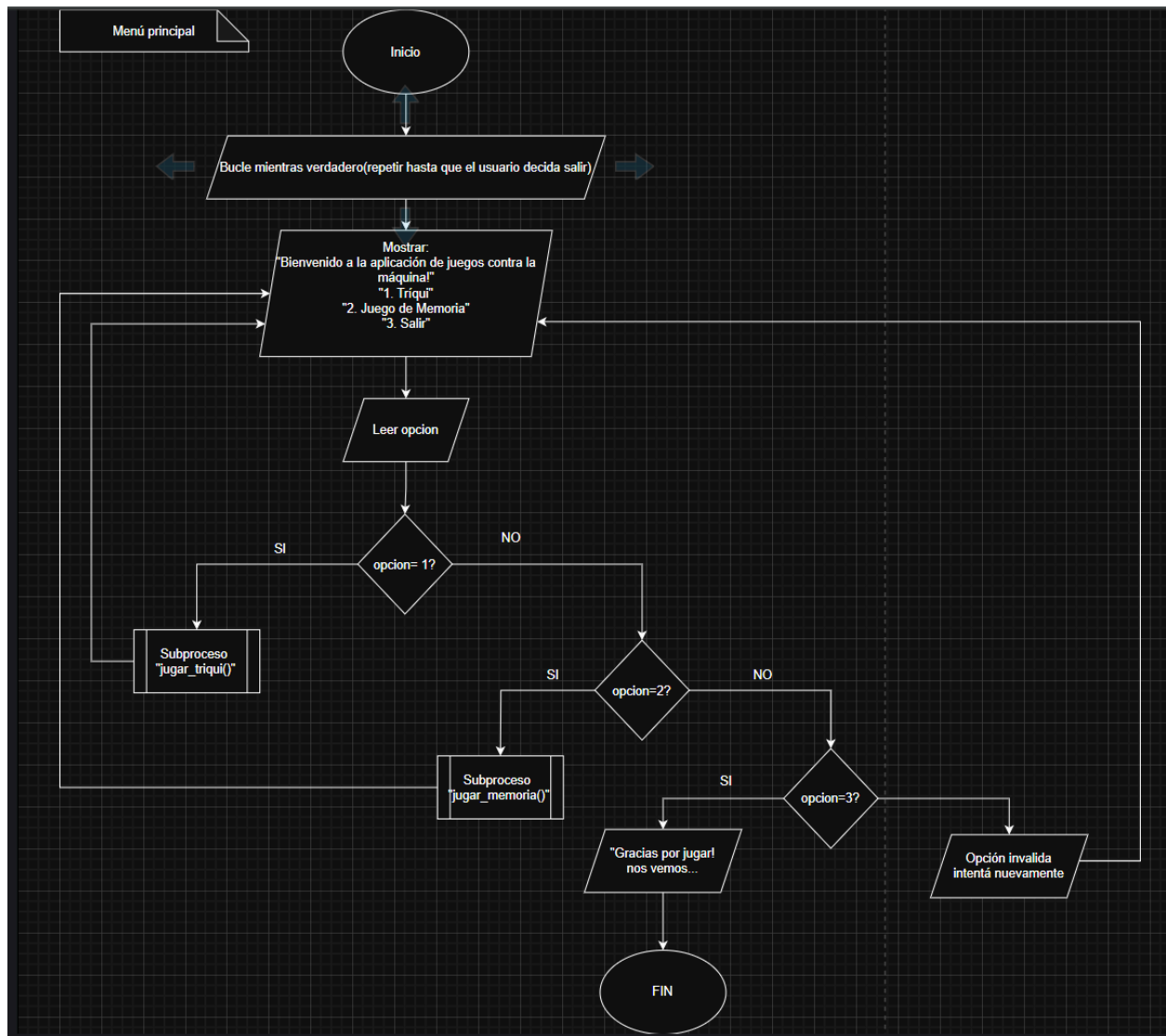
SINO

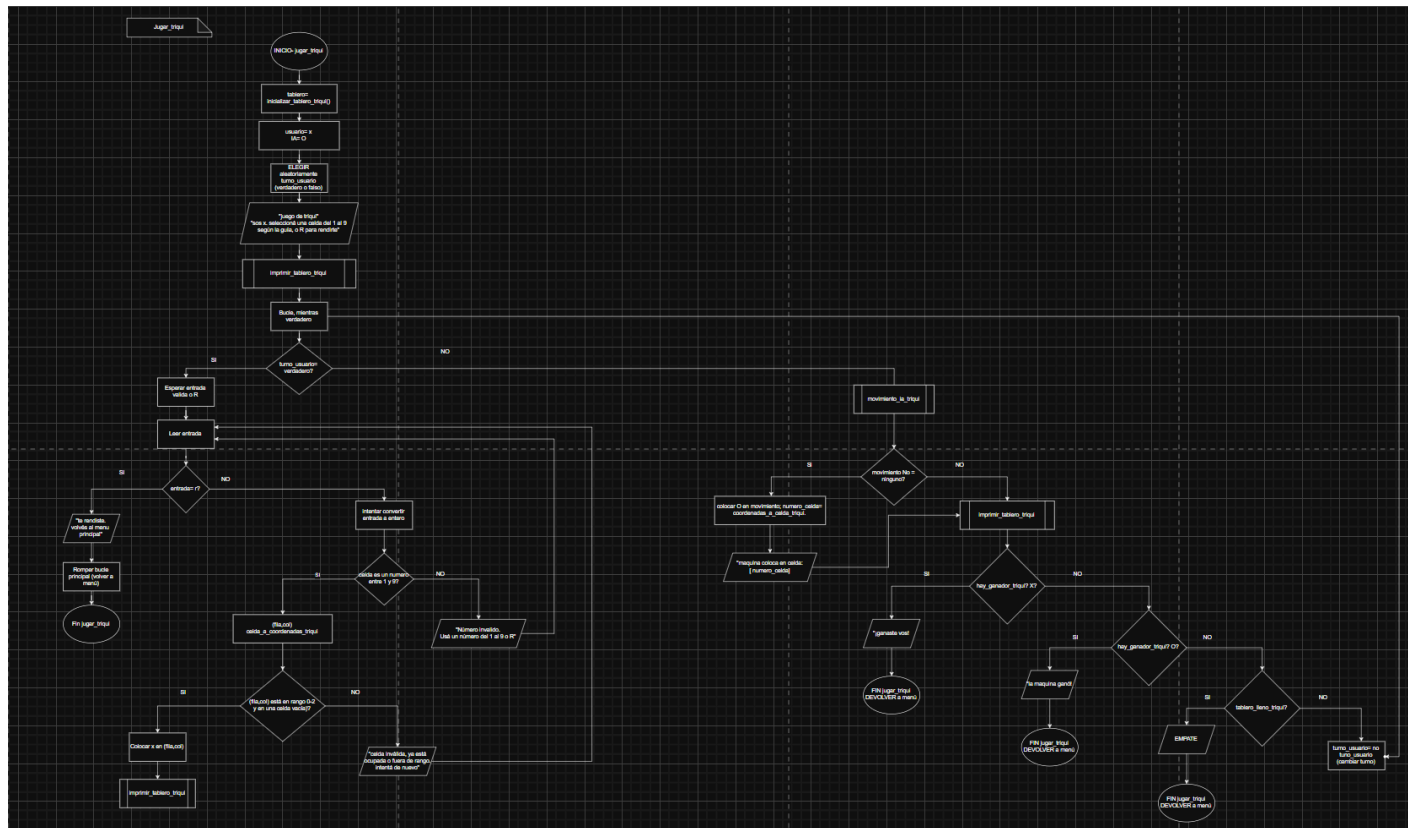
MOSTRAR "¡Empate!"

FIN-SI

FIN

Diagrama de flujo





Movimiento_IA_triqui

inicio
movimiento_IA_triqui

mejor_puntaje =
-infinito; mejor_mov= ninguno

Para cada movimiento en Movimientos_Disponibles_Triqui:
Colocar "O" en movimiento
puntaje = Minimax_Triqui(tablero, profundidad=0, es_maximizador=Falso)
Deshacer movimiento (volver a " ")
Si puntaje > mejor_puntaje:
mejor_puntaje = puntaje
mejor_mov = movimiento
Fin Para

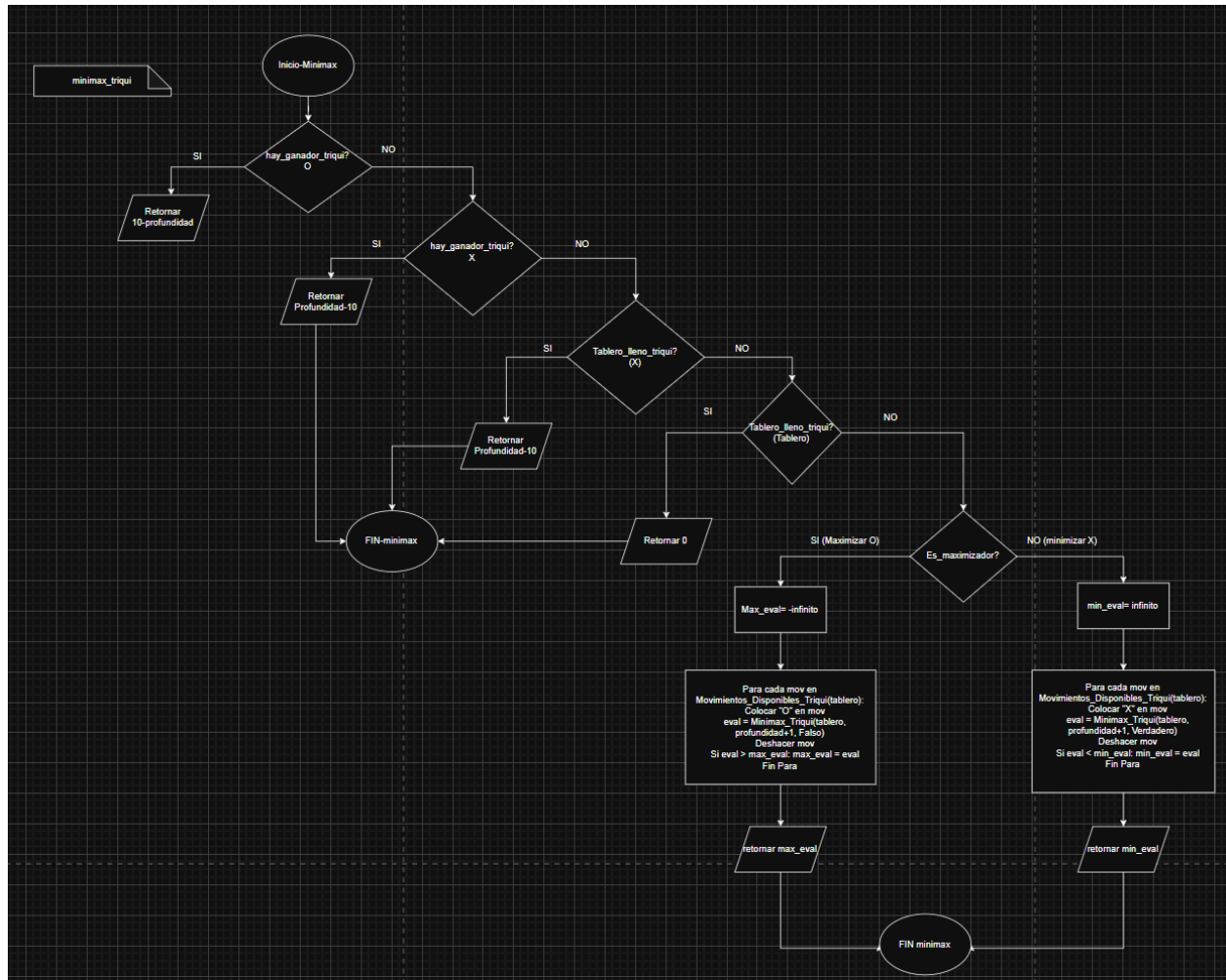
SI NO
generar_numero_aleatorio
(0,100) < 20?
(20% de probabilidad)

lista_movimientos =
Movimientos_Disponibles_Triqui(tablero)
Elegir aleatoriamente movimiento_aleatorio
de lista_movimientos

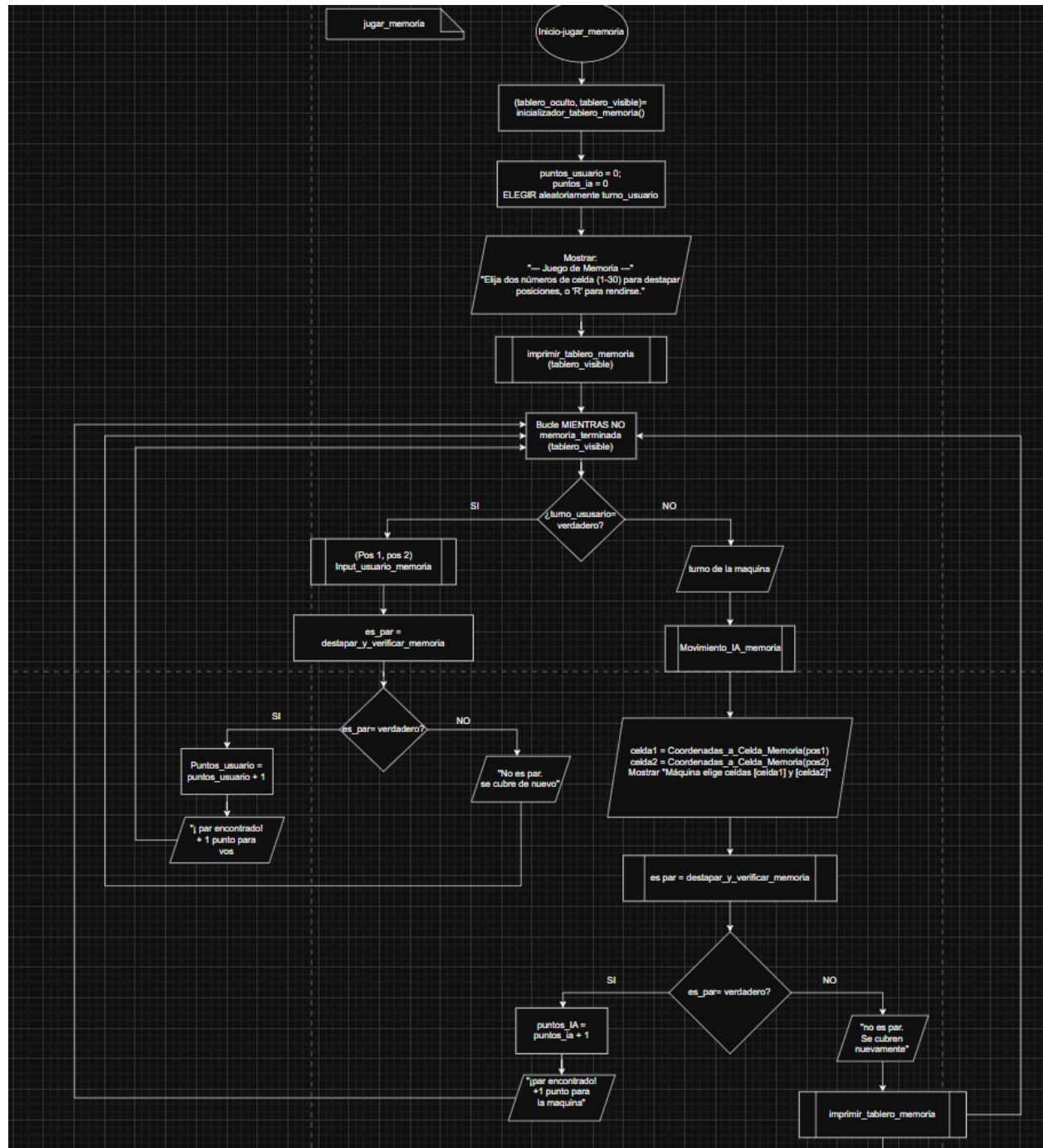
retomar
mejor_movimiento

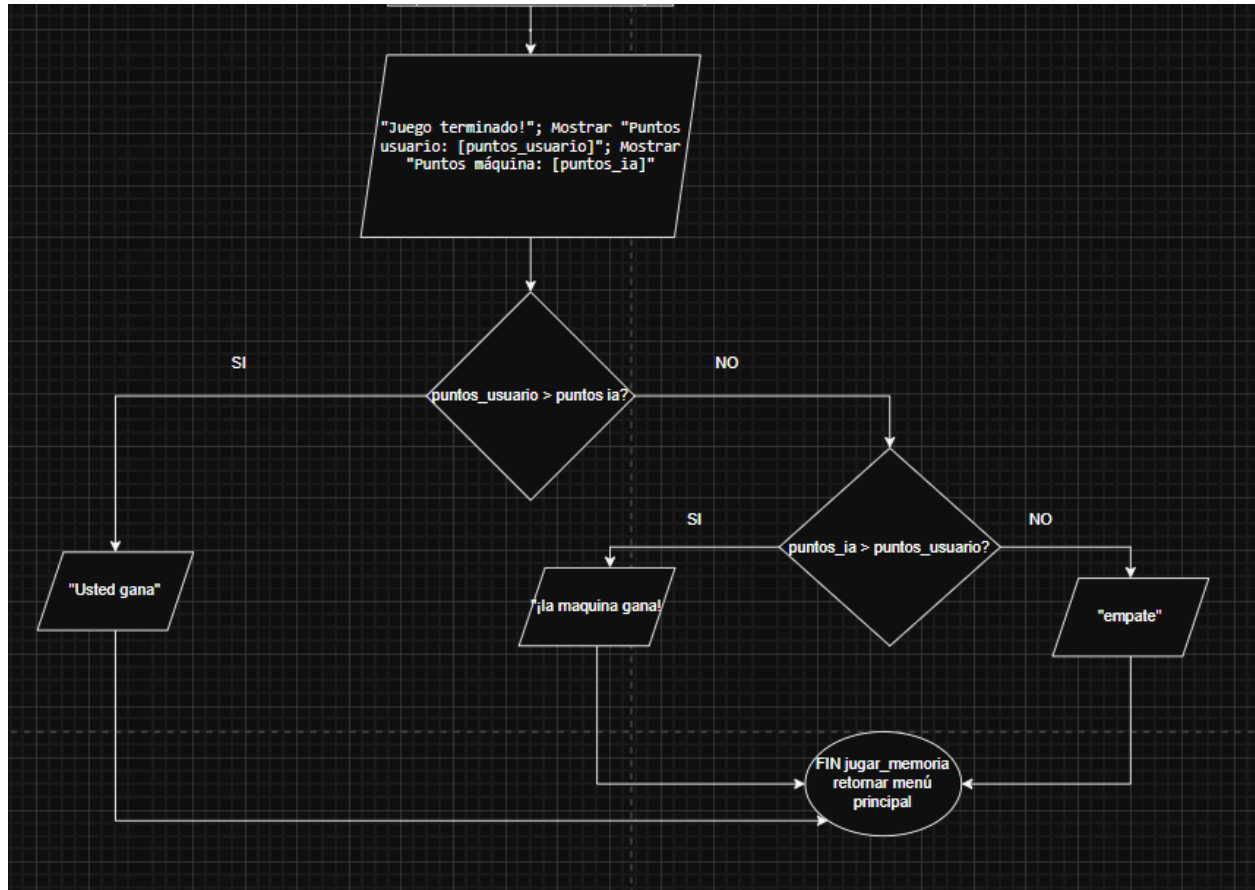
retomar
movimiento_aleatorio

Fin-retoma
movimiento



El diagrama de flujo de la función jugar_memoria está cortado en 2 imágenes porque es muy largo.





Pruebas de escritorio

Función principal

Entrada usuario	Proceso esperado	Salida
1	El menú detecta que eligió Triqui : se llama jugar_triqui()	Aparece tablero guía de Triqui
2	El menú detecta que eligió Memoria : se llama jugar_memoria()	Aparece tablero oculto de 6x5 con posiciones numéricas
3	El menú detecta Salir : termina el bucle	“Gracias por jugar! Saliendo...”
Q	El menú detecta opción inválida	“Opción inválida.” y vuelve al menú

Función triqui - Empate

Turno	Acción	Tablero después de la jugada (3x3)
1	Usuario pone en celda 1	X+---+--- ---+---+--- ---+---+---
2	IA pone en celda 5	X+---+--- ---+O+--- ---+---+---
3	Usuario pone en celda 2	X + X +--- ---+ O+--- ---+---+---
4	IA pone en celda 3	X + X + O ---+ O +--- ---+---+---
5	Usuario pone en celda 7	X + X + O ---+ O +--- X +---+---
6	IA pone en celda 9	X + X + O ---+ O +--- X +---+ O
7	Usuario pone en celda 8	X + X + O ---+ O +--- X + X + O
8	IA pone en celda 4	X + X + O O + O +--- X + X + O
9	Usuario pone en celda 6	X + X + O O + O + X X + X + O

Triqui - Máquina gana

Turno	Acción	Tablero después de la jugada (3x3)
1	Usuario pone en celda 1	X+---+--- ---+---+--- ---+---+---
2	IA pone en celda 5	X+---+--- ---+O+--- ---+---+---
3	Usuario pone en celda 2	X + X +--- ---+ O+--- ---+---+---
4	IA pone en celda 8	X + X + — ---+ O +--- ---+ O +---
5	Usuario pone en celda 3	X + X + X ---+ O +--- — +O +---
6	IA pone en celda 7	X + X + X ---+ O +--- O +O +---
7	Usuario pone en celda 9	X + X + X ---+ O +--- X + O + X
8	IA pone en celda 4	X + X + X O + O +--- O + O + X









Triqui - Usuario gana

Turno	Acción	Tablero después de la jugada (3x3)
1	Usuario pone en celda 1	X+---+--- ---+---+--- ---+---+---
2	IA pone en celda 2	X+ O+--- ---+---+--- ---+---+---
3	Usuario pone en celda 5	X + O +--- ---+ X +--- ---+---+---
4	IA pone en celda 3	X + O + O ---+ X +--- ---+ — +---
5	Usuario pone en celda 9	X + O + O ---+ X +--- — +--- + X

Memoria - usuario gana

Acción	Entrada usuario	Estado tablero visible	Resultado esperado
Inicio	Tablero oculto aleatorio con 15 pares	Todas las celdas en *	Tablero con números 1-30
Usuario elige	1 2	Se destapan 🍎 y 🍎	Si son iguales, permanecen descubiertas, puntos_usuario +1
Usuario sigue	Encuentra otro par	Se destapan 🍌 y 🍌	Mensaje: “¡Par encontrado! +1 punto para usted.”
...	Usuario sigue encontrando más pares
Fin del juego	Todas las celdas descubiertas	Usuario con 7 punto, IA con 2 puntos	“¡Usted gana!”

Memoria - usuario pierde

Acción	Entrada usuario / IA	Estado tablero visible	Resultado esperado
Inicio	Tablero oculto aleatorio con 15 pares	Todas las celdas en * (1–30)	Tablero con números 1-30
Usuario turno 1	1 2 ( y  , diferentes)	Se muestran y se tapan de nuevo	“No es un par. Se cubren nuevamente.”
IA turno 1	Máquina elige 3 4 () y )	Permanecen descubiertas	“¡Par encontrado! +1 punto para la máquina.”
Usuario turno 2	5 6 ( y  , diferentes)	Se muestran y se vuelven a tapar	Usuario no suma puntos
IA turno 2	Máquina elige 7 8 () y )	Permanecen descubiertas	Puntos IA = 2
...	La IA sigue encontrando más pares
Fin del juego	Todas las celdas descubiertas	Usuario con 1 punto, IA con 7 puntos	“¡La máquina gana!”

Pruebas de funcionamiento

Prueba menú

¡Bienvenido a la aplicación de juegos contra la máquina!

1. Juego de triqui
2. Juego de memoria
3. Salir

Selecciona un juego: 1, 2, o 3: 1

Juego de triqui

Eres 'X'. Selecciona una celda del 1 al 9 según la guía, o 'R' para rendirte.

Tablero actual, seleccioná una celda del 1 al 9 según esta guía, o 'R' para rendirte:

1 | 2 | 3

4 | 5 | 6

7 | 8 | 9

Estado actual del juego:

| |

| |

| |

Turno de la máquina...

La máquina juega en la celda 5

Tablero actual, seleccioná una celda del 1 al 9 según esta guía, o 'R' para rendirte:

1 | 2 | 3

4 | 5 | 6

7 | 8 | 9

```
Bienvenido a la aplicación de juegos contra la máquina!
1. Juego de tríqui
2. Juego de memoria
3. Salir
Selecciona un juego: 1, 2, o 3: 2

Juego de memoria
Seleccioná dos números del 1 al 30 que representen celdas cubiertas, o 'R' para rendirte.

Tablero de memoria, seleccioná dos números del 1 al 30 que representen celdas cubiertas, o 'R' para rendirte:
Fila 0: 1 2 3 4 5
Fila 1: 6 7 8 9 10
Fila 2: 11 12 13 14 15
Fila 3: 16 17 18 19 20
Fila 4: 21 22 23 24 25
Fila 5: 26 27 28 29 30
Turno de la máquina...
La máquina elige las celdas 5 y 27

Tablero de memoria, seleccioná dos números del 1 al 30 que representen celdas cubiertas, o 'R' para rendirte:
Fila 0: 1 2 3 4 🍌
Fila 1: 6 7 8 9 10
Fila 2: 11 12 13 14 15
Fila 3: 16 17 18 19 20
Fila 4: 21 22 23 24 25
Fila 5: 26 🍌 28 29 30
No es un par. Se tapan de nuevo.

Tablero de memoria, seleccioná dos números del 1 al 30 que representen celdas cubiertas, o 'R' para rendirte:
Fila 0: 1 2 3 4 5
Fila 1: 6 7 8 9 10
Fila 2: 11 12 13 14 15
Fila 3: 16 17 18 19 20
Fila 4: 21 22 23 24 25
Fila 5: 26 27 28 29 30
Tu jugada, ingresá dos números del 1 al 30, ej: 1 2, o 'R' para rendirte: 
```

```
Bienvenido a la aplicación de juegos contra la máquina!
1. Juego de tríqui
2. Juego de memoria
3. Salir
Selecciona un juego: 1, 2, o 3: 3
Gracias por jugar! Nos vemos.
```

TRIQUI
EMPATE

Estado actual del juego:

0 | 0 | x

x | x | 0

0 | x |

Turno de la máquina...

La máquina juega en la celda 9

Tablero actual, seleccioná una celda del 1 al 9 según esta guía, o 'R' para rendirte:

1 | 2 | 3

4 | 5 | 6

7 | 8 | 9

Estado actual del juego:

0 | 0 | x

x | x | 0

0 | x | 0

Empate!

MAQUINA GANO

x | | x

Turno de la máquina...

La máquina juega en la celda 6

Tablero actual, seleccioná una celda del 1 al 9 según esta guía, o 'R' para rendirte:

1 | 2 | 3

4 | 5 | 6

7 | 8 | 9

Estado actual del juego:

0 | x |

0 | 0 | 0

x | | x

La máquina ganó!

GANÓ JUGADOR

Estado actual del juego:

0 | x | 0

x | 0 | 0

x | | x

Tu jugada, ingresá un número del 1 al 9 o 'R' para rendirte: 8

Tablero actual, seleccioná una celda del 1 al 9 según esta guía, o 'R' para rendirte:

1 | 2 | 3

4 | 5 | 6

7 | 8 | 9

Estado actual del juego:

0 | x | 0

x | 0 | 0

x | x | x

Ganaste!

MEMORIA

Gana el jugador

Tablero de memoria. Selecciona dos números del 1 al 30 que representen celdas cubiertas, o 'R' para rendirte:

Fila 0: 🍌 🍌 🍌 🍌 🍌

Fila 1: 6 🍌 🍌 🍌 🍌 🍌

Fila 2: 🍌 🍌 🍌 🍌 🍌

Fila 3: 🍌 🍌 🍌 🍌 🍌

Fila 4: 🍌 🍌 🍌 🍌 25

Fila 5: 🍌 🍌 🍌 🍌 🍌

¡Encontraste un par! Sumas 1 punto.

Turno de la máquina...

La máquina elige las celdas 6 y 25

Tablero de memoria. Selecciona dos números del 1 al 30 que representen celdas cubiertas, o 'R' para rendirte:

Fila 0: 🍌 🍌 🍌 🍌 🍌

Fila 1: 🍌 🍌 🍌 🍌 🍌

Fila 2: 🍌 🍌 🍌 🍌 🍌

Fila 3: 🍌 🍌 🍌 🍌 🍌

Fila 4: 🍌 🍌 🍌 🍌 🍌

Fila 5: 🍌 🍌 🍌 🍌 🍌

La máquina encontró un par! Suma 1 punto.

¡Juego terminado!

Tus puntos: 12

Puntos de la máquina: 3

¡Ganaste!

Gana la maquina

Fila 5: 🍌 🍑 🍌 🍌 🍌

La máquina encontró un par! Suma 1 punto.

¡Juego terminado!

Tus puntos: 1

Puntos de la máquina:12

La máquina ganó!