

DOCUMENTO DE DISEÑO ENTREGA FINAL - PROYECTO PC

- **Función *menú principal***

ANALISIS:

Objetos conocidos	*Las opciones que el usuario puede escoger
Objetos desconocidos	Se conocen todos los objetos porque se pasar por parámetro a la función.
Condiciones	

DISEÑO:**ENTRADAS**

- Opción de tipo entero

SALIDAS

- Opción de tipo entero (modificada)

CONDICIONES

No tiene ninguna condición.

- **Función *menú nuevo juego***

ANALISIS:

Objetos conocidos	*Las opciones que el usuario puede escoger
Objetos desconocidos	Se conocen todos los objetos porque se pasar por parámetro a la función.
Condiciones	

DISEÑO:**ENTRADAS**

- Opción de tipo entero

SALIDAS

- Opción de tipo entero (modificada)

CONDICIONES

No tiene ninguna condición.

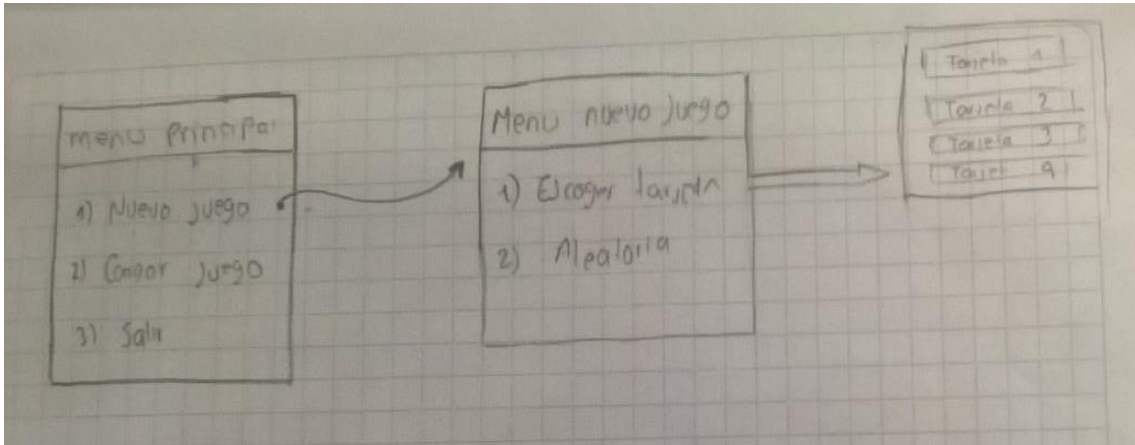


Imagen 1(Flujo de los menús)

- ***Función imprimir tablero con el movimiento***

ANALISIS:

Objetos conocidos	*Los caracteres de impresión ya que son constantes. *La dimensión del tablero que es 4x4. *Ubicación espacial de la nomenclatura del tablero, tal como, las letras de la a la d y los números del 1 al 4 .
Objetos desconocidos	Se conocen todos los objetos porque se pasar por parámetro a la función.
Condiciones	*Dependiendo del estado del estado del tablero (vacío o no), se finalizará el juego o continuará.

DISEÑO:**ENTRADAS**

- Espacio, de tipo *cadena de caracteres*
- Contador, de tipo *entero*
- Matriz, de tipo *cadena de caracteres*
- Posición, de tipo *apuntador a arreglo de entero*
- Ficha, de tipo *cadena de caracteres*

- Movimiento, de tipo *arreglo de caracteres*

SALIDAS

El tablero impreso a partir de ciclos que contienen caracteres que dan forma al tablero

- Retorna cero (0) o uno (1)

CONDICIONES

- En primer lugar, para imprimir las letras (a, b, c y d) estas se deben imprimir antes de iniciar los ciclos, e imprimir espacios para que quede centrado, estas corresponden a las columnas del tablero.
- Luego imprimimos las líneas horizontales en el ciclo interno.
- En el ciclo externo se imprimen las líneas verticales, y en la segunda línea imprimimos el número correspondiente a la fila.



Figura 1 (Diagrama de flujo función: *imprimir tablero con el movimiento*).

- **Función *transformar movimiento***

ANALISIS:

Objetos conocidos	*Movimiento que desea realizar el jugador.
Objetos desconocidos	*Se desconocen las coordenadas “reales”, es decir, las coordenadas de la ficha en la matriz.
Condiciones	*Se convirtió el movimiento en número que corresponda a posiciones validas en la matriz. Ej: a = 0, 1=0, b = 1, 2 = 1, etc.

DISEÑO:**ENTRADAS**

La única entrada que tiene esta función es ***movimiento***, la cuál es la entrada en el formato establecido para que se pueda mover la ficha, y en este caso, mover la torre.

- Movimiento, de tipo *cadena de caracteres*
- Posición, de tipo *apuntador a arreglo de entero*
- Ficha, de tipo *cadena de caracteres*

SALIDAS

- Posición, de tipo *apuntador a arreglo de entero (modificada)*

CONDICIONES

- Como el formato para el movimiento está conformado por una letra, un número, guion, otra letra, y otro número, entonces, lo primero que debemos hacer es determinar que letra digitó el usuario, y está convertirla a un número, como se muestra en la siguiente tabla, esto, con el fin de poder realizar las validaciones pertinentes en el tablero.

A	0
B	1
C	2
D	3

Tabla 1 (Conversión de letras).

- Lo mismo realizamos con los números.

1	0
2	1
3	2
4	3

Tabla 2 (Conversión de números).

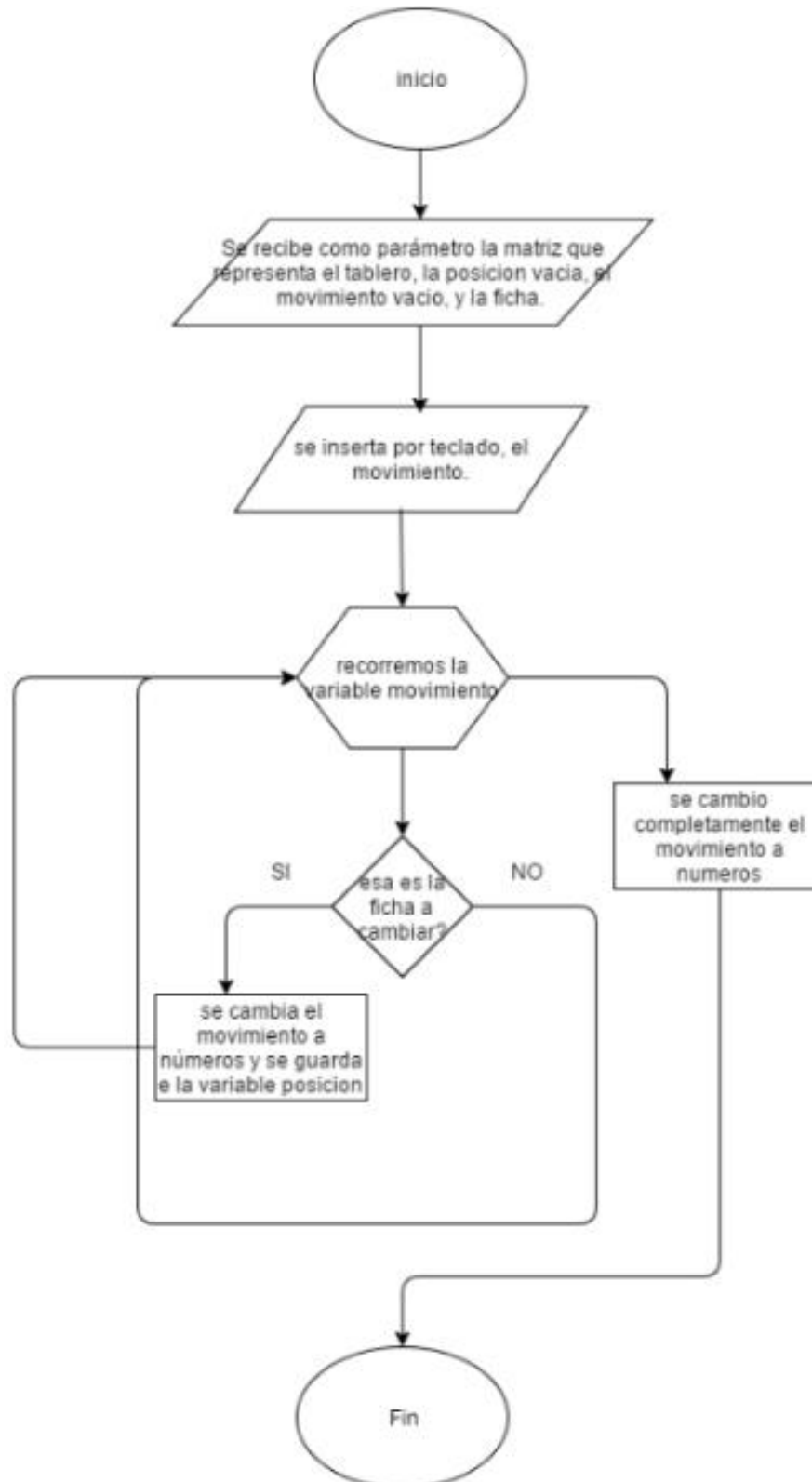


Figura 2 (Diagrama de flujo función: *insertar movimiento torre*).

- **Función *mover ficha***

ANALISIS:

Objetos conocidos	*Se conoce la posición, la ficha y el movimiento que desea realizar el usuario.
Objetos desconocidos	
Condiciones	*Si el valor que retorna la función <i>validación movimiento torre</i> es verdadero la ficha se cambiara de posición, y si no se mostrara un mensaje de <i>movimiento invalido</i> .

DISEÑO:**ENTRADAS**

Esta función tiene como entrada el valor que me devuelva la función *validación movimiento torre*, que puede ser **verdadero** o **falso**.

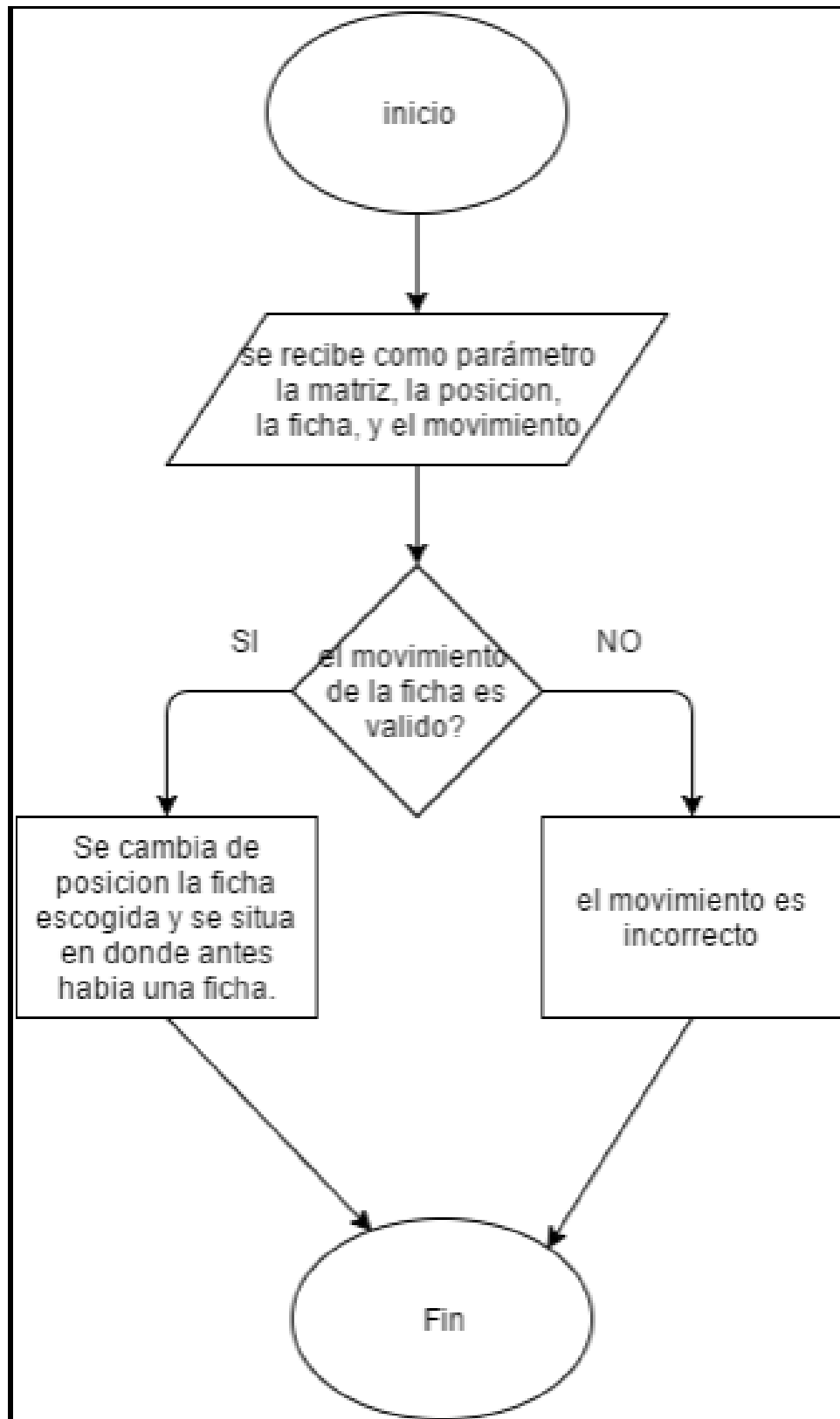
- Matriz, de tipo *cadena de caracteres*
- Posición, de tipo *apuntador a arreglo de entero*
- Ficha, de tipo *cadena de caracteres*

SALIDAS

Retorna un cero o un uno dependiendo de la condición

CONDICIONES

La única condición que tiene esta función depende del valor de retorno de mi función *validación movimiento ficha*, que es **verdadero** o **falso**, y de acuerdo a esto, realiza las acciones pertinentes para cada caso, retornando un 1 si el movimiento es válido y un 0 si no es así.

Figura 3 (Diagrama de flujo función: *mover ficha*).

- **Función *validación movimiento ficha***

ANALISIS:

Objetos conocidos	*Se conoce la posición actual y la posición destino la ficha en el tablero.
Objetos desconocidos	
Condiciones	*Si el movimiento de la ficha es válido o no, mostrará un mensaje en pantalla.

DISEÑO:
ENTRADAS

- Matriz, de tipo *cadena de caracteres*
- Posición, de tipo *apuntador a arreglo de entero*
- Ficha, de tipo *cadena de caracteres*

SALIDAS

Se mostrará por pantalla un mensaje, el cual, sí el usuario digitó un movimiento correcto saldrá ("Bien"), retornando *verdadero*, de lo contrario, saldrá por pantalla ("Movimiento incorrecto"), retornando *falso*.

CONDICIONES

- verificar que la casilla de origen haya una ficha
- verificar si en la casilla destino hay otra ficha, o, en otras palabras, no está vacía

- **Función *terminado***

ANALISIS:

Objetos conocidos	*Se conoce únicamente la matriz (tablero de juego).
Objetos desconocidos	
Condiciones	*Si no hay un espacio en tablero de juego el contador incrementará en uno.

DISEÑO:**ENTRADAS**

- Matriz, de tipo *cadena de caracteres*
- Contador, de tipo *entero*

SALIDAS

- Retorna verdadero o falso dependiendo del estado del tablero.

CONDICIONES

- Si hay una ficha retorna verdadero, de lo contrario, si hay más fichas, el contador incrementa y retorna falso.



Figura 4 (Diagrama de flujo función: *terminado*).

- **Función *imprimir tarjeta***

ANALISIS:

Objetos conocidos	*se conoce el número que el usuario digitó
Objetos desconocidos	*se desconoce la tarjeta en sí que se va a utilizar
Condiciones	*dependiendo del número se escoge la tarjeta y se llena en la variable

DISEÑO:**ENTRADAS**

- Opción, de tipo entero
- Arreglotarjetas, de tipo ficha

SALIDAS

- Se modifica la variable arreglotarjetas

CONDICIONES

- El número que hay en la variable opción, es el que decide cual case se escoge y así mismo se llena con los datos que están en ese case la variable arreglotarjetas.

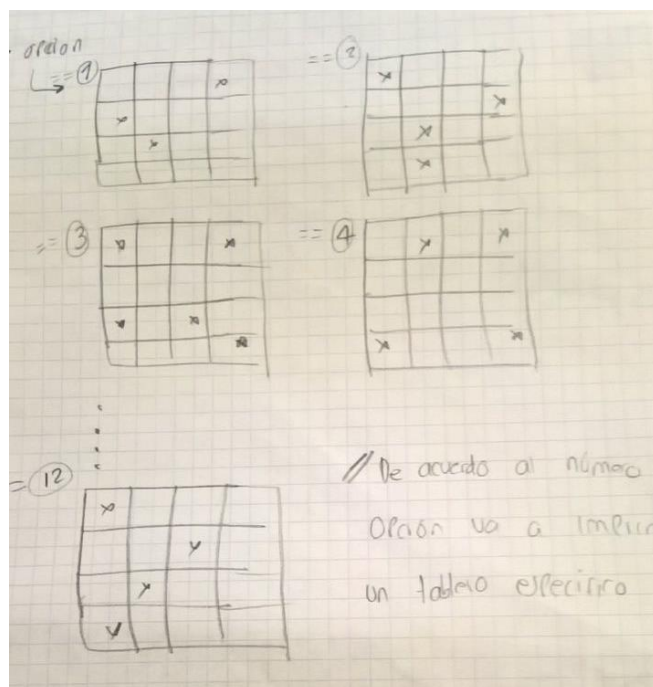


Imagen 2 (Ejemplo tarjetas)

- **Función *salvar partida en archivo***

ANALISIS:

Objetos conocidos	*se conoce el tablero de juego, el movimiento, si quiso o no guardar la partida, y el nombre del archivo
Objetos desconocidos	No siempre se conoce el nombre del archivo, debido a que todo se guarda si o si
Condiciones	*se verifica si el archivo está abierto, y si es así se guarda la partida en un archivo auxiliar, y si además de eso el usuario quiso guardar la partida, entonces se cambia el nombre de ese archivo al que el usuario digito

DISEÑO:
ENTRADAS

- NombreArchivo, de tipo cadena de caracteres
- Matriz, de tipo cadena de caracteres
- Movimiento, de tipo arreglo de caracteres
- GuardarPartida, de tipo caracter
- Contador, de tipo entero
- Espacio, de tipo cadena de caracteres
- Auxiliar, de tipo arreglo de caracteres

SALIDAS

Se modifica el archivo auxiliar, el cual cambia de nombre y se llena con la partida del usuario

CONDICIONES

Se crea un archivo auxiliar.txt, luego lo abrimos y verificamos si efectivamente está abierto. Luego guardamos el movimiento del usuario, las coordenadas de cada ficha y la ficha de esa coordenada y cerramos el archivo. Y esto lo hacemos así el usuario no pida aun guardar la partida. Ahora si el usuario decide guardar la partida, el archivo auxiliar.txt se cambia con el nombre que el usuario digite.

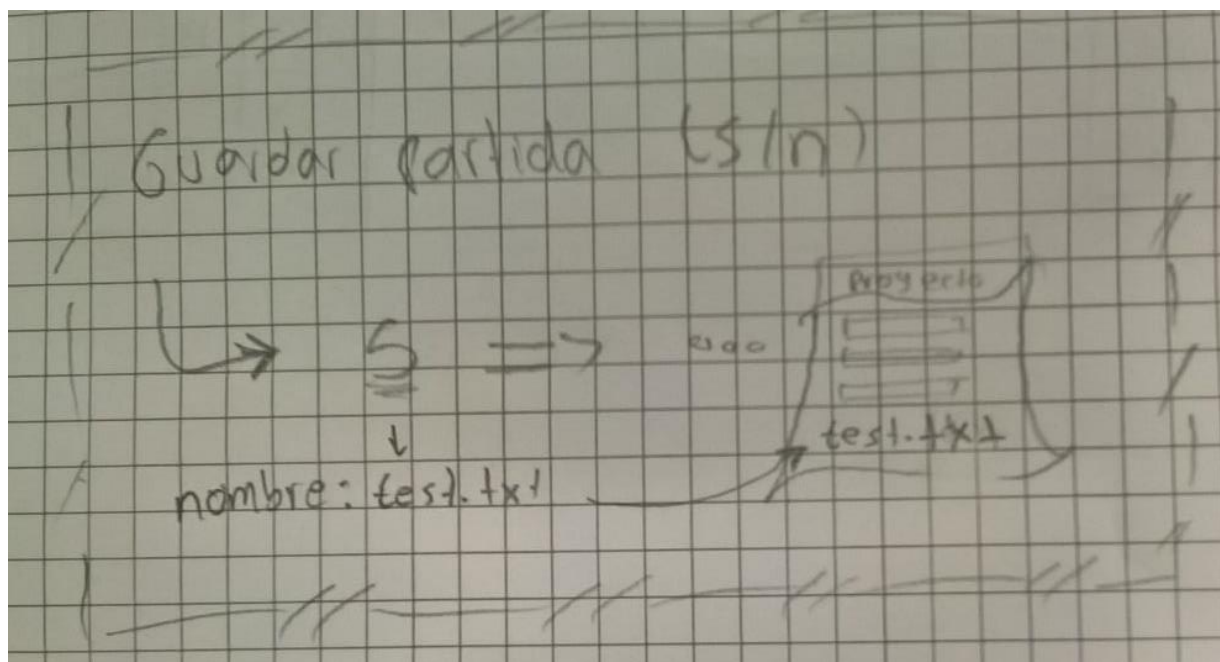


Imagen 3 (Función salvar partida)

- **Función cargar partida**

ANALISIS:

Objetos conocidos	*Se conoce el nombre del archivo
Objetos desconocidos	*Se desconoce la tarjeta que esta guardada en el archivo (claro este si el usuario no se acuerda)
Condiciones	*nada más la abertura del archivo,

DISEÑO:**ENTRADAS:**

- NombreArchivo, de tipo cadena de caracteres
- Arreglotarjetas, de tipo Ficha
- Fila, de tipo entero
- Columna, de tipo entero
- Ficha, de tipo cadena de caracteres

SALIDAS:

La variable arreglotarjetas modificada con lo que contiene el archivo

CONDICIONES:

Primero se abre el archivo solicitado, se verifica si efectivamente está abierto, y continuamente se extrae el movimiento, las coordenadas y la ficha, de la primera jugada que hizo el usuario. Se cierra el archivo.

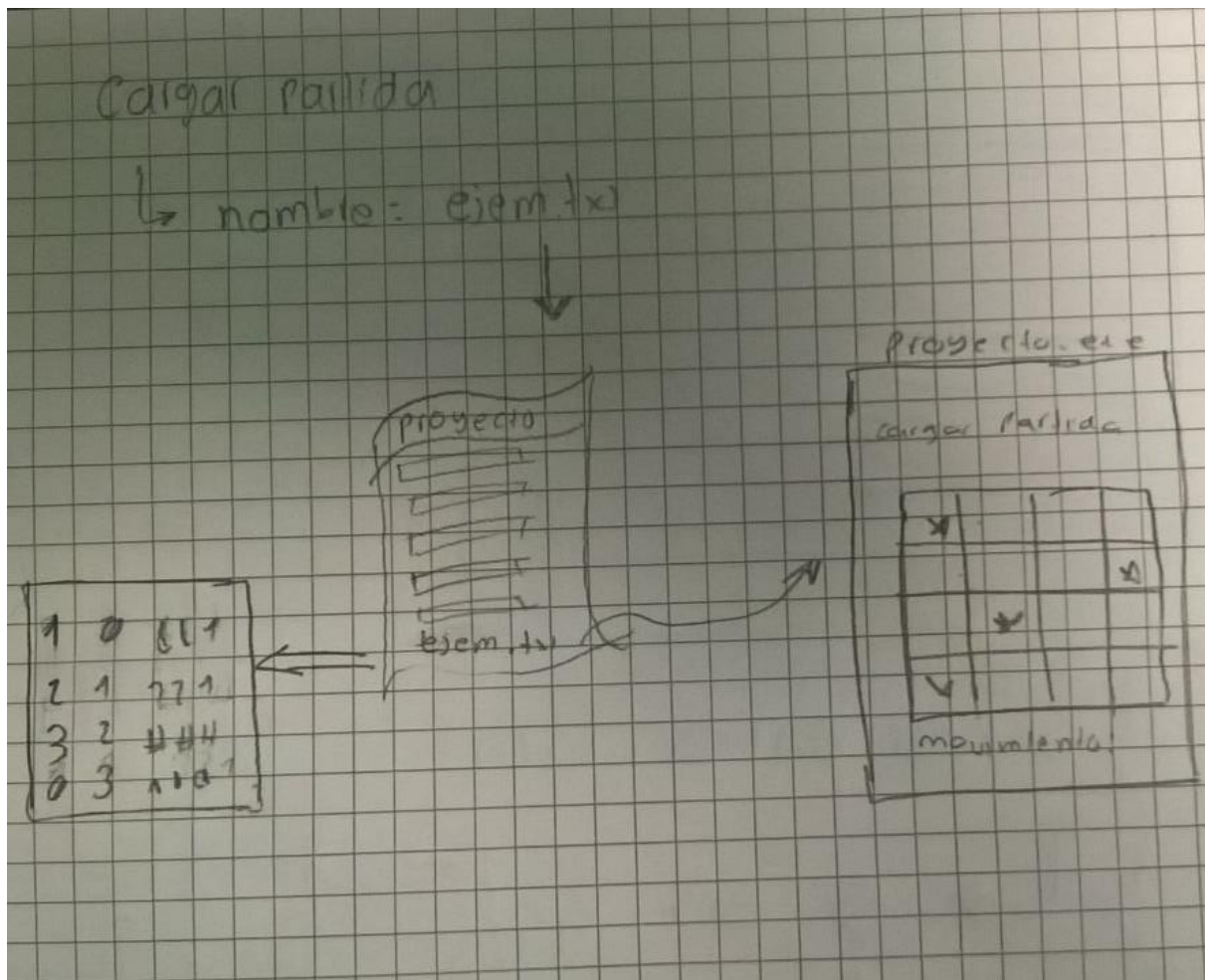


Imagen 4 (Función cargar partida)

- **Función *lista tarjetas***

ANALISIS:

Objetos conocidos	*se conoce el tablero de juego, y la tarjeta
Objetos desconocidos	
Condiciones	

DISEÑO:**ENTRADAS**

- Matriz, de tipo cadena de caracteres
- Arreglo tarjetas, de tipo Ficha

SALIDAS

- Muestra una tarjeta en pantalla

CONDICIONES

Dependiendo de la tarjeta que se envíe por parámetro de la función se imprime en pantalla esta con la configuración de la tarjeta escogida

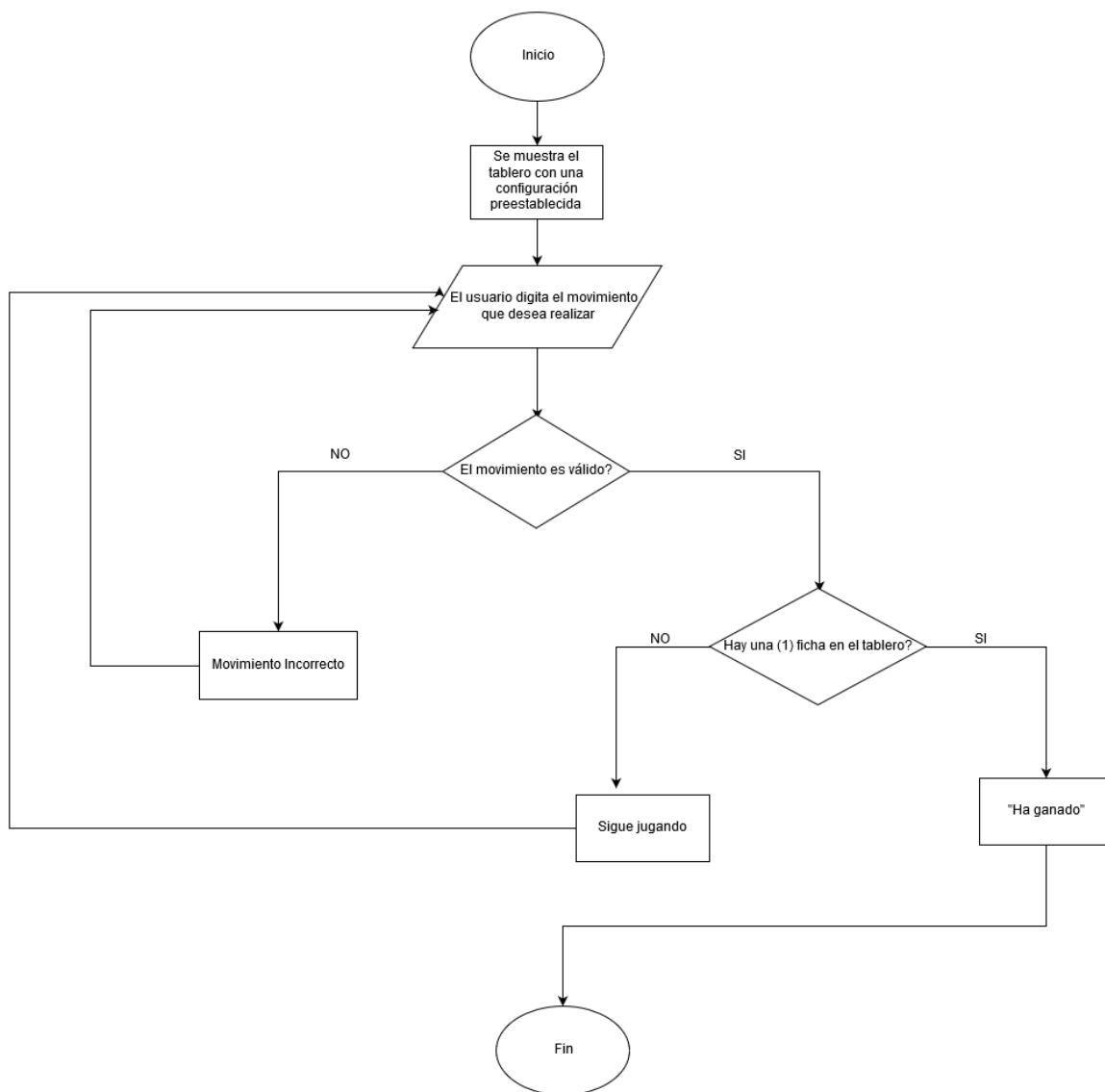
DIAGRAMA DE FLUJO Y DIBUJO DEL FUNCIONAMIENTO DEL PROGRAMA

Figura 5 (Diagrama de flujo funcionamiento general del programa).

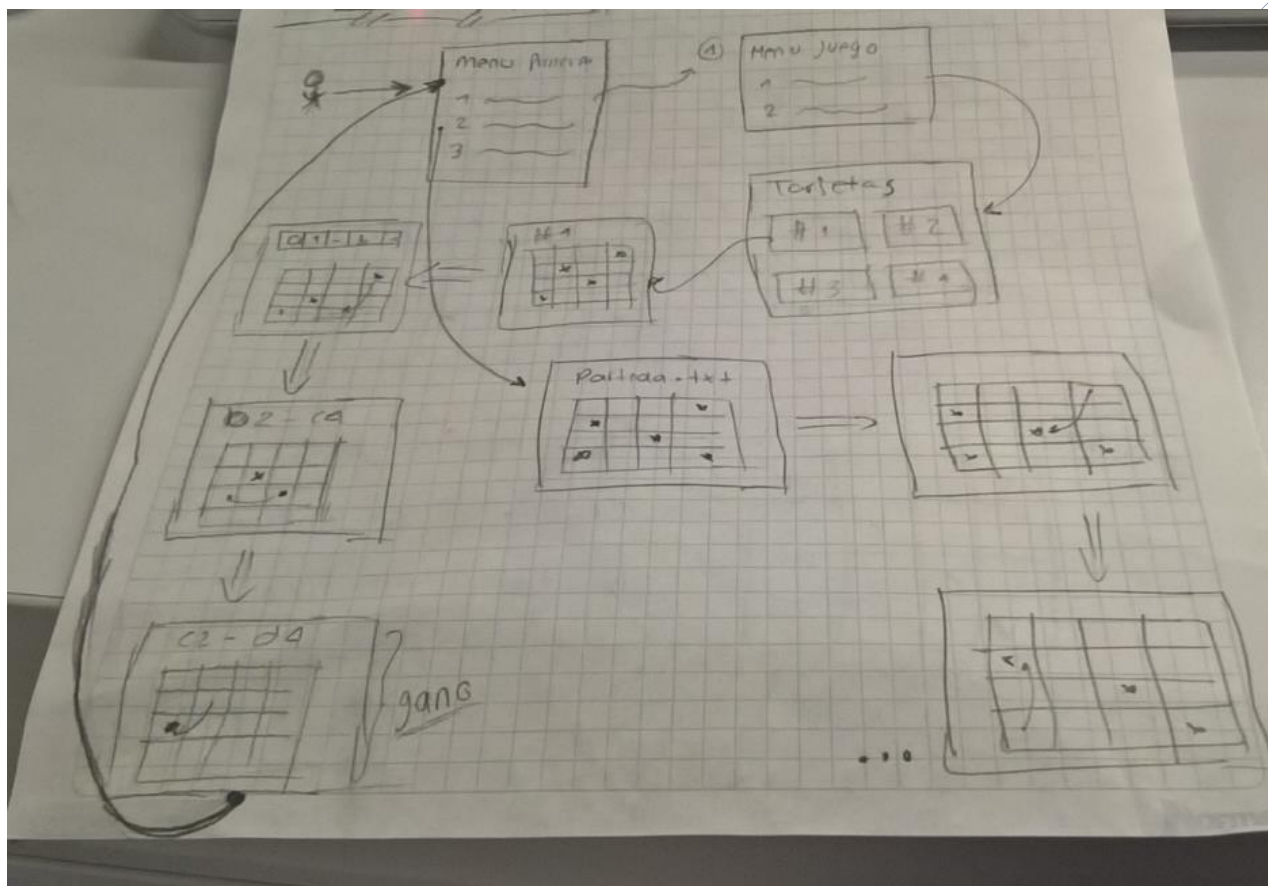


Imagen 5 (Funcionamiento del programa, boceto)