

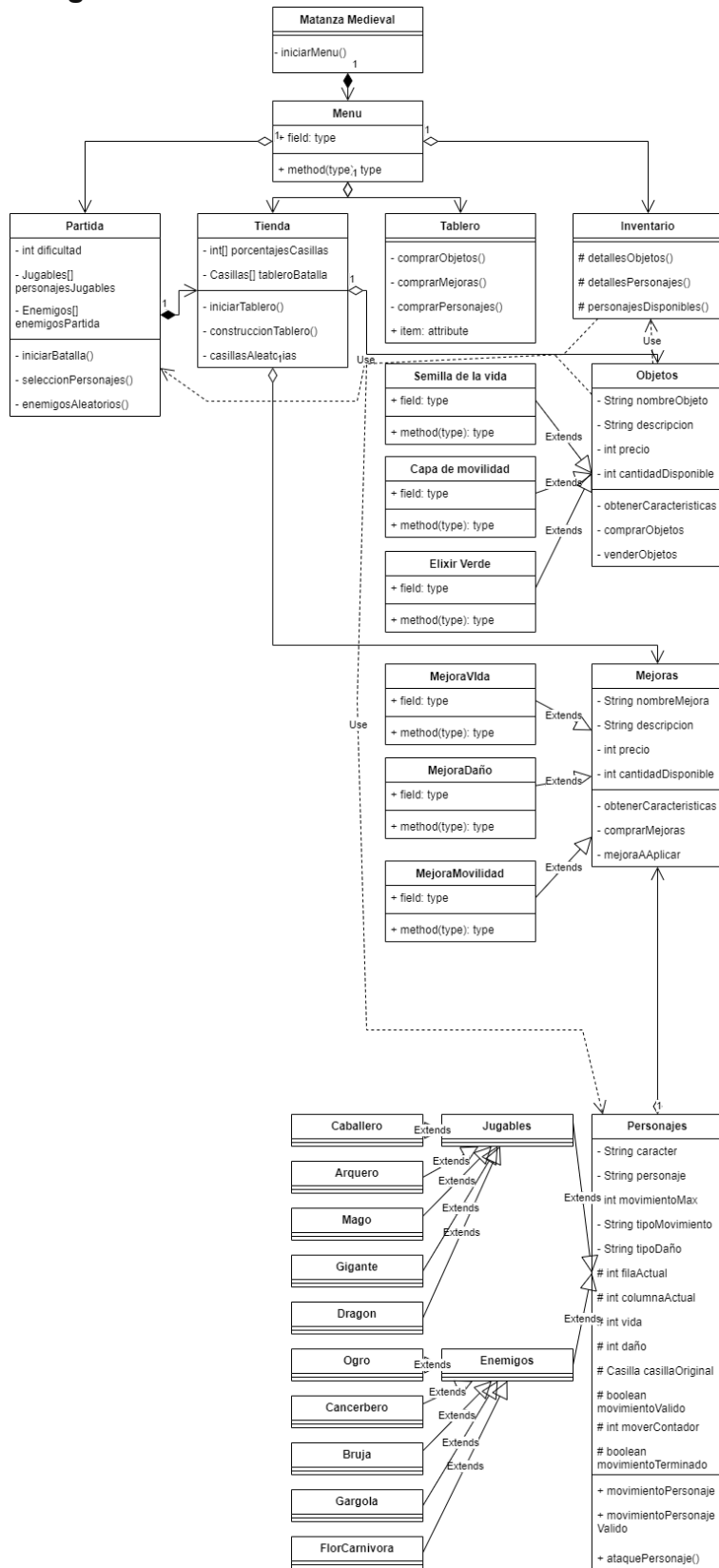
# MANUAL TÉCNICO

**IDE utilizada:** Netbeans

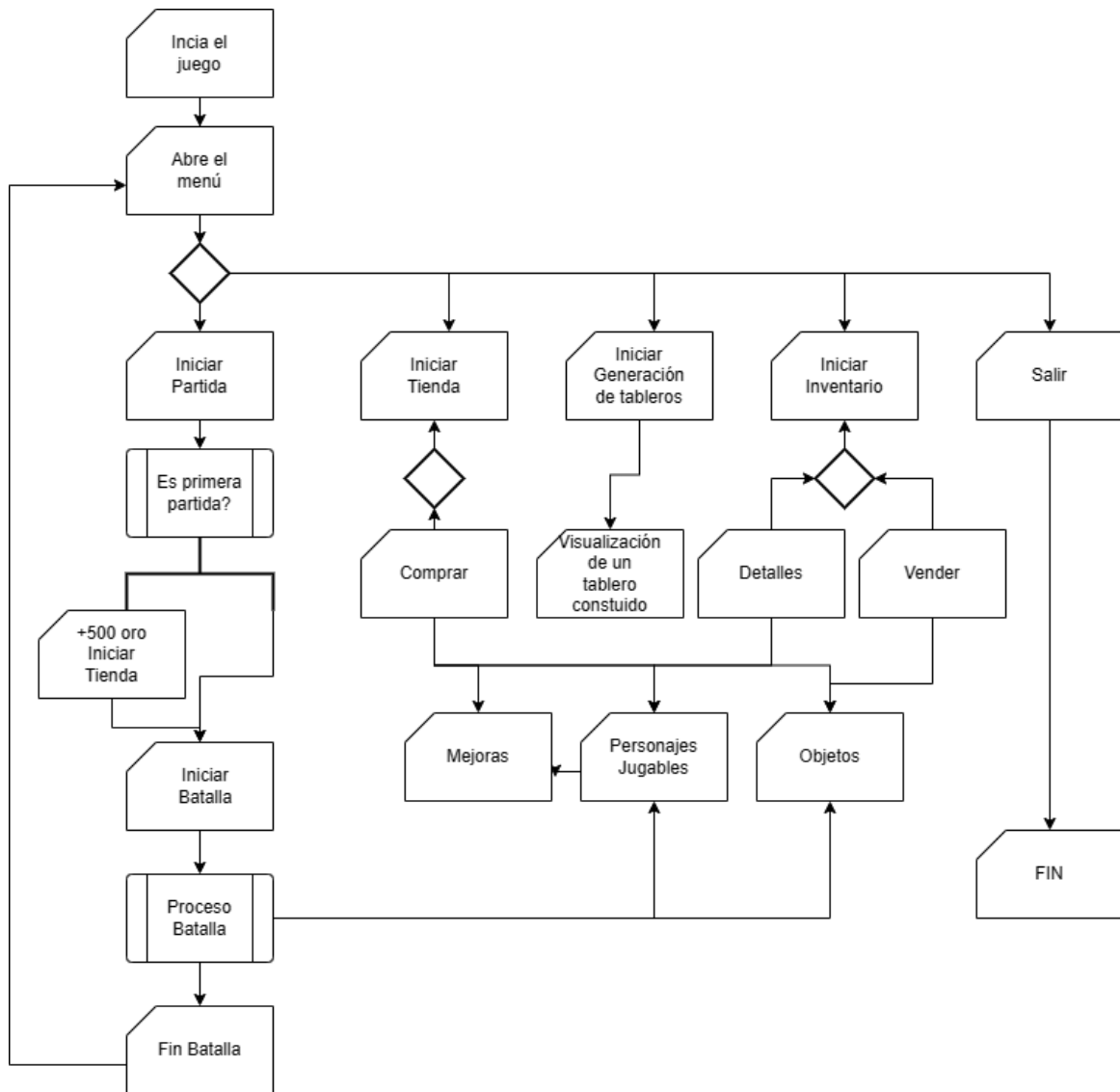
**Sistema operativo objetivo:** Windows, Linux, Macintosh, OS X

**Lenguaje de programación utilizado:** Java

## Diagrama de clases



## Diagrama de flujo general



## Métodos importantes

cargarArchivo

Crear un objeto de tipo JFileChooser

Crear un filtro para que reconozca solo archivos tipo .th y .txt

Obtener la ruta del nuevo archivo y almacenarlo en la clase de tablerosGuardados

guardarArchivo

Obtener el texto almacenado en el text área

Se crea un nuevo objeto tipo archivo, se guarda en tableros guardados y se notifica como guardado correctamente

Si no, se notifica que hubo un error

asignarBotones

Obtener el arreglo de tipo String con los caracteres obtenidos del archivo tablero

Crear un arreglo de botones con las dimensiones del tablero

Asignar al arreglo de botones según el carácter extraído en el archivo de tablero  
-, ~, #, T

movimientoPersonajeValido

destinoFila, destinoColumna

verificar que estén dentro de los rangos del tablero

Si es así, marcar que el movimiento es válido

Si no, marcar que el movimiento es inválido

movimientoPersonaje

apagar todos los botones

Verificar la dirección que se usará

Es Valido? Entonces encenderá el botón para que este pueda ser accionado por el usuario

ataquePersonajeValido

destinoFila, destinoColumna

verificar que estén dentro de los rangos del tablero

Si es así, marcar que el movimiento es válido

Si no, marcar que el movimiento es inválido

ataquePersonaje

apagar todos los botones

Verificar la dirección que se usará

Es Valido? Entonces encenderá el botón para que este pueda ser accionado por el usuario

construccionTablero

Obtener la cantidad de filas de la lectura de archivos

Obtener la cantidad de columnas de la lectura de archivos

Multiplicar filas y columnas para obtener la cantidad total de casillas de botones

Inicializar un arreglo bidimensional para almacenar los botones

Según la cantidad de casillas se asignan la cantidad de bombas

Según los caracteres obtenidos de la lectura del archivo se crean los tipos de botones

seleccionDeTiposTableros

Limpiar el texto en pantalla

Recorrer el arreglo de tableros guardados

Si el arreglo es nulo en el espacio 0

Entonces muestra que no hay tableros

Sino, muestra los tableros existentes que cumplan con las condiciones

Condiciones:

Según el tamaño

seleccionTablero

Obtener la opción seleccionada en el text field

Si la opción es valida

Entonces, iniciar una partida, enviando el archivo seleccionado para que se pueda iniciar el nuevo form y ocultar el actual, donde la opción = opción – 1, debido a que el arreglo empieza desde 0, de esta forma la primera opción será la que se almacena en tablerosGuardados[0]

Sino, notificar y que repita el ingreso de opción

partidaActiva

Llamar al método de construccion de tablero

Obtener el nuevo arreglo de botones

mostrarTablero

mostrarTablero

Obtenemos el tamaño de las casillas al dividir el largo y el ancho del panel entre la cantidad de botones

Desde 0 hasta filas

La posición Y es igual a la  
 $\text{casillaTamañoFilas} * i$

Desde 0 hasta columnas

La posición X es igual a la  
 $\text{casillaTamañoColumnas} * j$

Se elimina todos los ActionListener del  
botón

Setear la coordenada del botón usando i y j

Setear el color de fondo gris al botón

Setear el tamaño obtenido anteriormente  
para cada casilla

Se agrega el ActionListener al botón

Se agrega el botón al panel

Se repinta y revalida el panel

actionPerformed

Recorrer todos los botones

Según el tipo de acción a realizar

Ubicar personaje: almacena la casilla del  
terreno y declara al personaje en la nueva  
posición

Mover: moviliza al personaje dejando la  
casilla anterior en su lugar y almacenando  
la nueva, finalmente ubicandolo sobre la  
misma

Atacar: verifica que sea una casilla válida

para atacar y le descuenta el daño a la vida

Setear los nuevos valores en los labels de estado de vida de los personajes

Si ya no hay enemigos en el tablero

Entonces ganó la partida

Sino, Si el jugador ha muerto

Entonces, agrega al otro jugador

Si los dos jugadores mueren, entonces pierde la partida

caracteristicas() similar en varias clases

Mostrar la vida del personaje

Mostrar el daño del personaje

Mostrar el tipo de movimiento

Mostrar el movimiento máximo

Mostrar el tipo de ataque

comprar() similar en varias clases

Mostrar el oro disponible

Verificar si ya se compró el personaje

    Sí, entonces mostrar que ya lo adquirió

    No, entonces continuar

Mostrar ¿Desea comprar al personaje por precio de oro?

Verificar la respuesta

    Sí, entonces verificar la cantidad de oro

    Es mayor?

        Sí, entonces descontar el precio y marcar como comprado

        No, entonces mostrar que no tiene suficiente oro



personajeEstado()

El personaje está sobre lava?

Si, entonces vida > 5?

Sí, entonces descontar el 0.05 de su vida

No, entonces descontar 1 de su vida

No, entonces no hace nada

Mostrar vida

