

# DOCUMENTACION DE LA PRACTICA

## 1. Introducción del programa

El programa ha sido creado mediante una clase Juego, que se encarga de la gestión del propio juego, mediante el uso de clases privadas como Jugador y Tablero, (para la gestión del propio tablero y de los jugadores (turnos, numero de casillas ganas...)) como métodos propios de la clase para comprobar si el juego se ha completado, para devolver los datos de una partida, además de otros métodos (fuera de esta clase) para la gestión del programa.

El programa se ha creado como versión completa (Opciones 1-4 disponibles).

## 2. Métodos y explicación de ellos

Primero, explicaré en detalle la clase Juego con sus propias clases privadas y métodos. En hecho de tenerlo así es para el encapsulamiento y para tener una organización mejor de los datos.

**Tablero:** Esta es una clase privada de Juego, que gestiona entre otras cosas el tablero. Contiene variables como la lista bidimensional (tablero) y otras variables para guardar información relevante (caracteres que se colocan en el tablero, los palitos... etc.). Estos son sus métodos:

- concatenarCaracteres (Privado → Ejecutado con el constructor) Devuelve un string con todos los caracteres, de un array unidimensional, que se colocan en las posiciones donde posteriormente se colocaran los palitos.
- generarTablero (Privado → Ejecutado con el constructor) Genera el tablero inicial, donde coloca las propias esquinas de los cuadrados, y los caracteres generados en el método anterior donde se colocarán los palitos.
- tableroIncompleto: Método que devuelve un booleano, en función de si un tablero está incompleto (no contiene todos los palitos) o no lo está.
- comprobarCasillaSeleccionada(String s): Devuelve un booleano en función de si la casilla introducida por el Jugador, es correcta o no. Comprueba si esa casilla esta en el rango de casillas permitidas (El string que genera el primer método) y si se encuentra en el tablero. Si esto es correcto, ejecutará anyadirPared.
- anyadirPared(int i, int j): (Privado) Este método añadir un palito en la casilla del tablero correspondiente si el método anterior devuelve un true.
- comprobarCuadradosCompletos(char simbolo): Este método se ejecuta cuando se ejecuta actualizarDatos() de la clase Juego. Comprueba si las casillas que se pueden ganar, contienen palitos alrededor suyo (comprobarCasilla) y colocará el símbolo correspondiente al jugador que haya ganado esa casilla.

- comprobarCasilla(int i, int j): Privado → Ejecutado por el método anterior: Comprueba si una casilla (tablero[i][j]) esta rodeada de palitos. Devuelve true si lo esta o false si no lo está.

**Jugador:** La otra clase privada de la clase Juego. Esta gestionara las casillas ganadas de cada jugador, que símbolo tiene cada jugador (representará la casilla ganada por ese jugador), el numero (id) y el turno. Estos son sus métodos:

- getSimbolo: Devuelve el símbolo del jugador (char).
- getCasillasGanadas: Devuelve el número de casillas ganadas por el jugador (int).
- getNumJugador: Devuelve el numero del jugador, su id (int).
- alternarTurno: Método que cambia el turno del jugador. Si estaba en true pasa a false y viceversa.
- incrementarCasillas(int a): Aumenta el numero de casillas ganadas por el jugador por cada ronda. Ronda que deberá ser de su turno y que complete en ella un cuadrado.
- turnoJugador: Método que devuelve un booleano en función de si es su turno o no.

**Juego:** Clase principal. Estas son sus variables más importantes:

- Tablero tablero: Variable de tipo Tablero (objeto tipo Tablero).
- Jugador[] JUGADOR: Variable que contiene una lista con los Jugadores (objetos)

Con ellas se gestionarán tanto el tablero como los jugadores. Métodos propios de la clase Juego:

- tieneDuplicados (char[] array): Método privado que se ejecuta con el constructor. Este comprueba si los símbolos (el array de símbolos) son distintos para cada jugador. Si tiene símbolos iguales, lanzará una excepción indicando lo sucedido. (Esto es porque es fallo de la clase, no del usuario que juega)
- imprimirTablero: Método que mandara ejecutar el método imprimirTablero, para nuestra variable tablero (objeto de este tipo).
- comprobarCasillaSeleccionada: Método que mandara ejecutar este método para el objeto tablero que tenemos.
- actualizarDatos: Método que se ejecuta cada vez que un usuario realiza una colocación correcta de un palito. Este comprueba si esta acción ha generado un cuadrado rodeado de palos, lo que si es verdad añade la casilla ganada a un jugador (anyadirCasillasGanadas(int a)), y si es falso, pasa al siguiente jugador (siguienteJugador()).

- anyadirCasillasGanadas(int a): Añade un numero a de casillas ganadas al jugador que tenga el turno activo (Por lo general es 1, pero se ha gestionado así por si existen modificaciones en un futuro del programa y esa casilla valga más puntos o menos)
- jugadorActual: Método que devuelve el jugador actual. Para ello comprueba jugador por jugador de la lista y el que devuelva true mediante el método turnoJugador de la clase Jugador, será el Jugador que devuelva el método.
- siguienteJugador: Método que se ejecuta en actualizarDatos y cambia el turno al siguiente jugador de la lista. Si el jugador era el último, pasara al primero.
- juegoCompleto: Booleano que devuelve si un Juego se ha completado o no (ejecuta el método tableroIncompleto de la clase Tablero)
- imprimirResultadosJugadores: Imprime los resultados de cada uno de los jugadores.
- guardarResultado: Devuelve un string con la información que se guarda cuando finaliza un Juego → Dimensiones del tablero y el número de casillas ganadas por jugador.
- guardaPartida: Devuelve un string con la información que se guarda cuando el usuario decide guardar una partida. Dimensiones del tablero, el estado del tablero actual y el número de jugadores con el turno del que le tocaba.
- cargarPartida(String tab): Método que se ejecuta cuando un usuario decide cargar un juego. Este pasa un string que son la información del tablero de como estaba cuando se guardó. Este se encarga de actualizarlo.

Estos son los métodos de las clases que tenemos. Por lo general, los métodos de las clases privadas no tienen comprobaciones de entrada, ya que estos son ejecutados por los métodos de la clase Juego, por lo que una buena validación en estos métodos evita fallos en los métodos de la clase privada.

**Resto de métodos:** A la hora de ejecutar el programa, hay métodos fuera de la clase Juego, que se ejecutan a medida que funciona el programa.

- imprimirTablon(): Método que imprime el tablon. Se ha realizado para descongestionar todo lo posible el main.
- newGame(Scanner in, int nJugadores): Método que devuelve un objeto tipo Juego, ejecutado cuando se elige la opción 1 del tablon. En él se solicitará al usuario una dimensión para el tablero correcta (2x2 minimo) y creará un nuevo juego.

- loadGame(String archivo): Método ejecutado desde la opción 2 del tablón. "Carga" un juego mediante los datos obtenidos del tablón. Comprueba si estos datos son correctos (No han sido modificados) y devuelve un objeto Juego si esta todo correcto, o null si hay algún error.
- jugar(): Método que se ejecuta una vez hayamos obtenido un objeto juego (nueva partida o cargada). En ella se ejecuta el flujo de la partida, donde se solicita a los jugadores los palitos, se hacen sus comprobaciones si los inputs son correctos.  
Si lo introducido es \*\*, significa que se guarda la partida, lo que ejecuta guardarPartida().  
Si el juego finaliza, (tablero lleno), ejecuta anyadirResultado().
- guardarPartida(): Método que sirve para guardar la partida actual en el archivo correspondiente. Este sobre escribe los datos si los hay, por lo que actualmente solo se puede guardar una partida.
- anyadirResultado(): Método que se ejecuta en jugar si el tablero se ha completado. Guarda la información de la fecha actual y de la partida.
- verResultados(): Ejecutado cuando se selecciona la opción 3 del tablón. Este imprime por pantalla todas las partidas que se han terminado y su información.

\*\*Por lo general, todos los métodos están comentados mediante Javadoc.