

# Descripción de Métodos, Requerimientos y Observaciones Técnicas

## MINI PAC-MAN EN CONSOLA – JAVA

### 1. Descripción General del Proyecto

Esta práctica consiste en el desarrollo de una versión básica del videojuego Pac-Man implementada en lenguaje Java.

El sistema funciona completamente en consola y permite al usuario crear una partida personalizada, mover el personaje dentro de un tablero, recolectar premios, evitar fantasmas y visualizar un historial de partidas jugadas.

El programa fue desarrollado utilizando únicamente las librerías:

- \* java.util.Scanner
- \* java.util.Random

No se utilizaron estructuras avanzadas ni librerías externas adicionales.

### 2. Requerimientos de la Aplicación

El sistema debe:

Mostrar un menú principal con las opciones:

- \* Crear un tablero
- \* Ver puntuación
- \* Salir

Solicitar al usuario:

- \* Nombre de usuario
- \* Cantidad de premios (máximo 40% del tablero)
- \* Cantidad de paredes (máximo 20%)
- \* Cantidad de fantasmas (máximo 20%)

Generar un tablero de dimensiones 5x6.

Permitir al usuario seleccionar la posición inicial del personaje.

Permitir el movimiento del personaje utilizando:

- \* 8 → Arriba
- \* 5 → Abajo
- \* 6 → Derecha
- \* 4 → Izquierda
- \* F → Pausa

Implementar:

- \* Sistema de vidas (3 vidas)
- \* Sistema de puntuación
- \* Premio especial
- \* Historial de partidas

Implementar bordes infinitos:

- \* Si el personaje sale por un extremo, reaparece en el lado contrario.

Mostrar historial ordenado del más reciente al más antiguo.

### 3. Requerimientos No Funcionales

- \* El sistema debe ejecutarse en consola.
- \* Debe validar entradas incorrectas.
- \* Debe funcionar sin errores de ejecución.
- \* No debe utilizar librerías externas adicionales.

### 4. Descripción de los Métodos Utilizados

**main():** Método principal que muestra el menú inicial y controla el flujo general del programa.

**opcionElegida(int opcionInicio):** Controla la ejecución según la opción seleccionada en el menú principal.

**asignarCantidades(String objeto, double porcentaje):** Valida que la cantidad ingresada esté dentro del rango permitido según el porcentaje máximo del tablero.

**colocarElemento(String[][] tablero, String simbolo, int cantidad):** Coloca aleatoriamente los elementos (premios, paredes, fantasmas) en posiciones libres del tablero.

**verificacionPosicionPersonaje(int x, int y, String[][] tablero):** Verifica que la posición elegida por el jugador esté disponible antes de colocar el personaje.

**imprimirTablero(String[][] tablero):** Imprime visualmente el tablero con bordes superiores e inferiores.

**moverPersonaje(String[][] tablero, int[] posicion, String direccion):** Controla:

- \* Movimiento del personaje
- \* Validación de paredes
- \* Recolección de premios
- \* Encuentro con fantasmas
- \* Lógica de bordes infinitos

Retorna un valor entero que indica el resultado del movimiento.

**quedanPremios(String[][] tablero):** Verifica si aún existen premios dentro del tablero para determinar si el jugador ha ganado.

## 5. Observaciones Técnicas Relevantes

- \* Se utilizó una clase interna llamada “Partida” para almacenar nombre y puntaje.
- \* El historial se almacena en memoria utilizando una estructura tipo lista.
- \* El tablero es representado por una matriz bidimensional de tipo String.
- \* Se implementó validación de porcentajes usando el total de espacios del tablero.
- \* El programa reinicia el menú principal al finalizar o ganar una partida.

# Manual de Usuario

Mini Pac-Man en Consola

## 1. Inicio del Sistema

Al ejecutar el programa, se mostrará el menú principal con las siguientes opciones:

1. Crear un tablero
2. Ver puntuación
3. Salir

El usuario debe ingresar el número correspondiente a la opción deseada y presionar Enter.

## 2. Crear Nueva Partida

Al seleccionar la opción 1, el sistema solicitará:

- \* Nombre de usuario
- \* Cantidad de premios
- \* Cantidad de paredes
- \* Cantidad de fantasmas

El sistema validará que las cantidades ingresadas estén dentro del rango permitido.

Posteriormente, se generará automáticamente el tablero y el usuario deberá elegir la posición inicial del personaje ingresando:

- \* Fila
- \* Columna

Si la posición no es válida, el sistema solicitará nuevamente los datos.

### 3. Interfaz del Juego

Durante la partida se mostrará:

- \* Nombre del jugador
- \* Puntaje actual
- \* Cantidad de vidas

También se mostrará el tablero actualizado en cada movimiento.

### 4. Controles del Juego

El jugador debe utilizar los siguientes comandos:

- 8 → Mover hacia arriba
- 5 → Mover hacia abajo
- 6 → Mover hacia la derecha
- 4 → Mover hacia la izquierda
- F → Pausar el juego

### 5. Sistema de Juego

- \* El jugador inicia con 3 vidas.
- \* Cada premio normal otorga 10 puntos.
- \* El premio especial otorga 15 puntos.
- \* Si el jugador encuentra un fantasma, pierde una vida.
- \* Si pierde todas las vidas, la partida termina (Game Over).
- \* Si recoge todos los premios, gana la partida.

El tablero cuenta con bordes infinitos, lo que significa que si el personaje sale por un extremo, reaparece en el lado contrario.

### 6. Menú de Pausa

Al presionar la tecla F, aparecerán las opciones:

- 3. Regresar a la partida
- 4. Terminar partida

Si el usuario elige terminar, el sistema guardará el puntaje en el historial y regresará al menú principal.

## 7. Historial de Partidas

En la opción 2 del menú principal, el usuario podrá visualizar un listado de partidas realizadas.

El historial muestra:

- \* Nombre del jugador
- \* Puntaje obtenido

Las partidas se muestran ordenadas de la más reciente a la más antigua.

# Implementación

## IMPLEMENTACIÓN DEL PROYECTO MINI PAC-MAN EN JAVA

### 1. Descripción de la Implementación

El proyecto consiste en el desarrollo de una versión básica del juego Pac-Man utilizando el lenguaje de programación Java, ejecutado completamente en consola.

El tablero fue implementado mediante una matriz bidimensional de tipo `String[][]`, donde cada posición representa un espacio dentro del juego.

Se definieron constantes para:

- \* Dimensiones del tablero (5 filas x 6 columnas).
- \* Símbolos de cada elemento del juego:
- \* Jugador
- \* Premio normal
- \* Premio especial
- \* Pared
- \* Fantasma
- \* Valores de puntuación para cada tipo de premio.

El sistema utiliza únicamente las siguientes librerías:

- \* `Scanner` para capturar datos ingresados por el usuario.
- \* `Random` para generar posiciones aleatorias dentro del tablero.

También se implementó una clase interna llamada `Partida`, cuya función es almacenar:

- \* Nombre del usuario.
- \* Puntaje obtenido en la partida.

Las partidas se almacenan en memoria dentro de una lista, lo que permite mostrar el historial ordenado desde la más reciente hasta la más antigua.

## 2. Lógica General del Programa

El flujo del sistema funciona de la siguiente manera:

1. Se muestra el menú principal.
2. El usuario selecciona una opción.
3. Si crea una nueva partida:
  - \* Ingresar su nombre.
  - \* Define la cantidad de premios, paredes y fantasmas.
  - \* Se genera el tablero.
  - \* Se colocan los elementos aleatoriamente.
  - \* El usuario elige la posición inicial del personaje.
4. Se inicia el ciclo de juego.
5. En cada movimiento:
  - \* Se actualiza la posición del personaje.
  - \* Se verifica si recoge premio.
  - \* Se verifica si encuentra fantasma.
  - \* Se valida colisión con paredes.
  - \* Se aplica la lógica de bordes infinitos.
6. El juego termina cuando:
  - \* El jugador pierde todas las vidas.
  - \* El jugador recoge todos los premios.
7. Se guarda la partida en el historial.
8. El sistema regresa al menú principal