



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA EN CIENCIAS Y SISTEMAS
INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1

Practica 1 PAC-MAN

Nombre: Angee Ximena Chojolán Oliva.

Carné: 202505018 Cui: 3902547890101 Sección: E

Docente: EDUARDO ISAÍ AJSVINAC XICO

1. MANUAL TÉCNICO

-Requerimientos técnicos

Requerimiento	Descripción
Lenguaje de programación	Java
Versión JDK	7-8
IDE	NetBeans
Librerías utilizadas	java.util.Scanner (entrada de datos), java.util.Random (generación aleatoria)

-Estructura del proyecto.

➤ Package pacman

- Pacman.java (Clase principal)
- Tablero.java (Manejo del tablero)
- Partida.java (Lógica del juego)
- Historial.java (Registro de partidas)

-Métodos

Clase Pacman.java

- main(String[] args)
- leerOpcion(int min, int max)
- iniciarJuego()

Clase Tablero.java

- Tablero(int filas, int columnas, int cantPremios, int cantParedes, int cantFantasmas)
- colocarElementos(char elemento, int cantidad)
- colocarPremios(int total)
- mostrar()
- colocarPacman(int fila, int columna)
- moverPacman(int filaActual, int columnaActual, int nuevaFila, int nuevaColumna)
- getContenido(int fila, int columna)

- hayFantasma(int fila, int columna)
- hayPared(int fila, int columna)
- getFilas()
- getColumnas()

Clase Partida.java

- Partida(String nombreUsuario, Tablero tablero, int filaInicial, int columnaInicial, Historial historial)
- colocarPacmanEnPosicionLibre()
- jugar()
- mostrarEstado()
- procesarMovimiento(char tecla)
- mostrarMenuPausa()

Clase Historial.java

- Historial()
- agregarPartida(String usuario, int puntaje)
- mostrarHistorial()
- getUltimoUsuario()
- getUltimoPuntaje()

-Observaciones Técnicas

- **No se utilizan ArrayList ni colecciones dinámicas:** Cumpliendo con los requisitos de la práctica
- **Solo arreglos nativos:** Se utilizan vectores y matrices tradicionales
- **Capacidad del historial:** Máximo 100 partidas almacenadas
- **Formato de salida:** Tablero con bordes de guiones y barras verticales para mejor visualización
- **Aleatoriedad:** Uso de Random para distribución de elementos
- **Modularidad:** Separación de responsabilidades en 4 clases

2. MANUAL DE USUARIO

USUARIO:

PUNTEO: 0

VIDAS: 3

====MENU DE INICIO ====

1. Iniciar Juego
2. Historial de partidas
3. Salir

POR FAVOR, INGRESE LOS SIGUIENTES VALORES: ||

===== ESPECIFICAR TABLERO =====

POR FAVOR, INGRESE LOS SIGUIENTES VALORES

TABLERO: P (Pequeno 5x6)

G (Grande 10x10)

SELECCIONE: G

PREMIOS [1-40]: 20

PAREDES [1-20]: 20

TRAMPAS [1-20]: 20

	@	X	0	0	\$	X	
		@	@	X	X	0	@ @
	@	0	X	X	@	0	0
	X		X		0		
	X	X	X		\$	0	X
		@	@	0	X	@	\$
	@	0	X	X		X	
	@	@	X	@	X	0	
		@	0	@			
	\$	@	@			@ X	0

ESPECIFICAR TABLERO

POR FAVOR, INGRESE LA POSICIÓN INICIAL DE JUEGO

FILA: |

ESPECIFICAR TABLERO
POR FAVOR, INGRESE LA POSICION INICIAL DE JUEGO
FILA: 2
COLUMNAS: 2

USUARIO: Angee

PUNTEO: 0

VIDAS: 3

| @ X 0 0 \$ X |
| @ @ X X 0 @ @ |
| @ 0 < X X @ 0 0 0 |
| X X 0 |
| X X X \$ 0 X 0 |
| @ @ 0 X @ \$ |
| @ 0 X X X X |
| @ @ X @ X 0 0 |
| @ 0 @ @ |
\$ @ @ @ X 0

8: ARRIBA 5: ABAJO 6: DERECHA 4: IZQUIERDA F: PAUSA
MOVIMIENTO: ||

USUARIO: Angee

PUNTEO: 0

VIDAS: 3

| @ X 0 0 \$ X |
| @ @ X X 0 @ @ |
| @ 0 < X X @ 0 0 0 |
| X X 0 |
| X X X \$ 0 X 0 |
| @ @ 0 X @ \$ |
| @ 0 X X X X |
| @ @ X @ X 0 0 |
| @ 0 @ @ |
\$ @ @ @ X 0

8: ARRIBA 5: ABAJO 6: DERECHA 4: IZQUIERDA F: PAUSA
MOVIMIENTO: 5

USUARIO: Angee

PUNTEO: 0

VIDAS: 3

| @ X 0 0 \$ X |
| @ @ X X 0 @ @ |
| @ 0 X X @ 0 0 0 |
| X X 0 |
| X X X \$ 0 X 0 |
| @ @ 0 X @ \$ |
| @ 0 X X X X |
| @ @ X @ X 0 0 |
| @ 0 @ @ |
\$ @ @ @ X 0

8: ARRIBA 5: ABAJO 6: DERECHA 4: IZQUIERDA F: PAUSA
MOVIMIENTO: 4

USUARIO: Angee

PUNTEO: 0

VIDAS: 3

```
-----  
| @ X      0 0 $   X   |  
|     @ @ X X 0   @ @ |  
| @ 0   X X @ 0 0   0 |  
| X       X   0   |  
| X   X X   $ 0 X   0 |  
| < @ @ 0 X   @   $ |  
| @ 0 X X   X   X   |  
| @ @ X @ X   0   0 |  
| @ 0 @   @   |  
| $ @ @       @ X 0 |  
-----
```

8: ARRIBA 5: ABAJO 6: DERECHA 4: IZQUIERDA F: PAUSA

MOVIMIENTO: 6

♦Te encontraste con un fantasma! Vidas restantes: 2

USUARIO: Angee

PUNTEO: 0

VIDAS: 2

```
-----  
| @ X      0 0 $   X   |  
|     @ @ X X 0   @ @ |  
| @ 0   X X @ 0 0   0 |  
| X       X   0   |  
| X   X X   $ 0 X   0 |  
| < @ @ 0 X   @   $ |  
| @ 0 X X   X   X   |  
| @ @ X @ X   0   0 |  
| @ 0 @   @   |  
| $ @ @       @ X 0 |  
-----
```

8: ARRIBA 5: ABAJO 6: DERECHA 4: IZQUIERDA F: PAUSA

MOVIMIENTO: 5

♦Premio! +10 puntos

USUARIO: Angee

PUNTEO: 10

VIDAS: 1

```
-----  
| @ X      0 0 $   X   |  
|     @ @ X X 0   @ @ |  
| @ 0   X X @ 0 0   0 |  
| X       X   0   |  
| X   X X   $ 0 X   0 |  
| @ 0 X   @   $   |  
| X   X X   X   X   |  
| @ @ X @ X   0   0 |  
| < @ @ @   @   |  
| $ @ @       @ X 0 |  
-----
```

8: ARRIBA 5: ABAJO 6: DERECHA 4: IZQUIERDA F: PAUSA

MOVIMIENTO: 5

♦Premio! +15 puntos

USUARIO: Angee

PUNTEO: 25

VIDAS: 1

USUARIO: Angee

PUNTEO: 25

VIDAS: 1

```
-----  
| @ X 0 0 $ X |  
| @ @ X X 0 @ @ |  
| @ 0 X X @ 0 0 0 |  
| X X 0 |  
| X X $ 0 X 0 |  
| @ 0 X @ X 0 @ $ |  
| X X X X |  
| @ @ X @ X 0 0 |  
| @ 0 @ 0 @ |  
| < @ @ @ X 0 |  
-----
```

8: ARRIBA 5: ABAJO 6: DERECHA 4: IZQUIERDA F: PAUSA

MOVIMIENTO: f

PAUSA

POR FAVOR, SELECCIONE UNA OPCI N

1. REGRESAR
2. TERMINAR PARTIDA

OPCI N:

PAUSA

POR FAVOR, SELECCIONE UNA OPCI N

1. REGRESAR
2. TERMINAR PARTIDA

OPCI N: 2

USUARIO: Angee

PUNTEO: 25

VIDAS: 3

====MENU DE INICIO ===

1. Iniciar Juego
2. Historial de partidas
3. Salir

POR FAVOR, INGRESE LOS SIGUIENTES VALORES: 2

HISTORIAL DE PARTIDAS

No.	USUARIO	PUNTEO
-----	---------	--------

1.	Angee	25

USUARIO: Angee

PUNTEO: 25

VIDAS: 3

====MENU DE INICIO ===

1. Iniciar Juego
2. Historial de partidas
3. Salir

POR FAVOR, INGRESE LOS SIGUIENTES VALORES: |

3. INFORME DE DESARROLLO

▪ Introducción.

El presente documento describe el proceso de desarrollo de un juego de PACMAN en consola, implementado en Java como parte de la Práctica 1 del curso Introducción a la Programación y Computación 1. El proyecto consiste en un juego interactivo donde el jugador debe recolectar premios evitando fantasmas, aplicando conceptos fundamentales de programación.

▪ Desarrollo.

El proyecto se dividió en 4 clases:

- Pacman.java: Clase principal que maneja el menú y la configuración inicial
- Tablero.java: Gestiona la matriz del juego y los elementos
- Partida.java: Controla la lógica del juego y los movimientos
- Historial.java: Administra el registro de partidas

▪ Problemas en el desarrollo.

Formato del tablero . El tablero se veía desordenado y difícil de leer.

Movimiento en borde. Al llegar al borde, el juego generaba errores de índice

Errores de sintaxis.

▪ Conclusiones.

- Se cumplieron los objetivos del proyecto
- Se manejaron de forma correcta de arreglos, métodos, clases, condiciones.

■ Código en Java

→ Pacman.java

```
1 package pacman;
2
3 import java.util.Scanner;
4 import java.util.Random;
5
6 public class Pacman {
7     private static Scanner scanner = new Scanner(System.in);
8     private static Random random = new Random();
9     private static Historial historial = new Historial();
10
11    public static void main(String[] args) {
12        int opcion;
13        do {
14            System.out.println("\nUSUARIO: " + (historial.getUltimoUsuario() != null ? historial.getUltimoUsuario() : ""));
15            System.out.println("PUNTEO: " + historial.getUltimoPuntaje());
16            System.out.println("VIDAS: 3");
17            System.out.println("\n====MENU DE INICIO ====");
18            System.out.println("1. Iniciar Juego");
19            System.out.println("2. Historial de partidas");
20            System.out.println("3. Salir");
21            System.out.print("POR FAVOR, INGRESE LOS SIGUIENTES VALORES: ");
22
23            opcion = leerOpcion(1, 3);
24
25            switch (opcion) {
26                case 1:
27                    iniciarJuego();
28                    break;
29                case 2:
30                    historial.mostrarHistorial();
31                    break;
32                case 3:
33                    System.out.println("¡Gracias por jugar!");
34                    break;
35            }
36        } while (opcion != 3);
37    }
38
39    private static int leerOpcion(int min, int max) {
40        int opcion;
41        do {
42            while (!scanner.hasNextInt()) {
43                System.out.print("Error. Ingrese un número válido: ");
44                scanner.next();
45            }
46            opcion = scanner.nextInt();
47            if (opcion < min || opcion > max) {
48                System.out.print("Opción fuera de rango. Intente nuevamente: ");
49            }
50        } while (opcion < min || opcion > max);
51        return opcion;
52    }
53
54    private static void iniciarJuego() {
55        System.out.print("\nINGRESE SU NOMBRE DE USUARIO: ");
56        String nombre = scanner.next();
57
58        System.out.println("\n==== ESPECIFICAR TABLERO ====");
59        System.out.println("POR FAVOR, INGRESE LOS SIGUIENTES VALORES");
60        System.out.println("TABLERO: P (Pequeno 5x6)");
61        System.out.println("G (Grande 10x10)");
62        System.out.print("SELECCIONE: ");
63        String tipoTablero = scanner.next().toUpperCase();
64
65        while (!tipoTablero.equals("P") && !tipoTablero.equals("G")) {
66            System.out.print("TABLERO INVÁLIDO. SELECCIONE P o G: ");
67            tipoTablero = scanner.next().toUpperCase();
68        }
69    }
}
```

```

70 |         int filas = tipoTablero.equals("P") ? 5 : 10;
71 |         int columnas = tipoTablero.equals("P") ? 6 : 10;
72 |         int totalEspacios = filas * columnas;
73 |
74 |         System.out.print("PREMIOS [1-" + (int)(totalEspacios * 0.4) + "]: ");
75 |         int cantidadPremios = scanner.nextInt();
76 |         while (cantidadPremios < 1 || cantidadPremios > (int)(totalEspacios * 0.4)) {
77 |             System.out.print("VALOR INVÁLIDO. INGRESE [1-" + (int)(totalEspacios * 0.4) + "]: ");
78 |             cantidadPremios = scanner.nextInt();
79 |         }
80 |
81 |         System.out.print("PAREDES [1-" + (int)(totalEspacios * 0.2) + "]: ");
82 |         int cantidadParedes = scanner.nextInt();
83 |         while (cantidadParedes < 1 || cantidadParedes > (int)(totalEspacios * 0.2)) {
84 |             System.out.print("VALOR INVÁLIDO. INGRESE [1-" + (int)(totalEspacios * 0.2) + "]: ");
85 |             cantidadParedes = scanner.nextInt();
86 |         }
87 |
88 |         System.out.print("TRAMPAS [1-" + (int)(totalEspacios * 0.2) + "]: ");
89 |         int cantidadFantasmas = scanner.nextInt();
90 |         while (cantidadFantasmas < 1 || cantidadFantasmas > (int)(totalEspacios * 0.2)) {
91 |             System.out.print("VALOR INVÁLIDO. INGRESE [1-" + (int)(totalEspacios * 0.2) + "]: ");
92 |             cantidadFantasmas = scanner.nextInt();
93 |         }
94 |
95 |         Tablero tablero = new Tablero(filas, columnas, cantidadPremios, cantidadParedes, cantidadFantasmas);
96 |         tablero.mostrar();
97 |
98 |         System.out.println("\n**ESPECIFICAR TABLERO**");
99 |         System.out.println("POR FAVOR, INGRESE LA POSICIÓN INICIAL DE JUEGO");
100 |        System.out.print("FILA: ");
101 |        int filaPacman = scanner.nextInt();
102 |        System.out.print("COLUMNA: ");
103 |        int columnaPacman = scanner.nextInt();
104 |
105 |        Partida partida = new Partida(nombre, tablero, filaPacman, columnaPacman, historial);
106 |        partida.jugar();
107 |
108 |

```

→ Tablero.java

```

1  package pacman;
2
3  import java.util.Random;
4
5  public class Tablero {
6      private char[][] tablero;
7      private int filas;
8      private int columnas;
9      private int premiosRestantes;
10     private Random random = new Random();
11
12     public Tablero(int filas, int columnas, int cantPremios, int cantParedes, int cantFantasmas) {
13         this.filas = filas;
14         this.columnas = columnas;
15         this.tablero = new char[filas][columnas];
16         this.premiosRestantes = cantPremios;
17
18         for (int i = 0; i < filas; i++) {
19             for (int j = 0; j < columnas; j++) {
20                 tablero[i][j] = ' ';
21             }
22         }
23
24         colocarElementos('X', cantParedes);
25         colocarElementos('@', cantFantasmas);
26         colocarPremios(cantPremios);
27     }
28
29     private void colocarElementos(char elemento, int cantidad) {
30         int colocados = 0;
31         while (colocados < cantidad) {
32             int fila = random.nextInt(filas);
33             int columna = random.nextInt(columnas);
34             if (tablerofila][columna] == ' ') {
35                 tablerofila][columna] = elemento;
36                 colocados++;
37             }
38         }
39     }
40
41     public void mostrar() {
42         for (int i = 0; i < filas; i++) {
43             for (int j = 0; j < columnas; j++) {
44                 System.out.print(tablero[i][j]);
45             }
46             System.out.println();
47         }
48     }
49
50     public void moverPacman(int fila, int columna, char dirección) {
51         if (dirección == 'N' && fila > 0 && tablerofila - 1][columna] == ' ') {
52             tablerofila - 1][columna] = '@';
53             tablerofila][columna] = ' ';
54             fila--;
55         } else if (dirección == 'S' && fila < filas - 1 && tablerofila + 1][columna] == ' ') {
56             tablerofila + 1][columna] = '@';
57             tablerofila][columna] = ' ';
58             fila++;
59         } else if (dirección == 'E' && columna < columnas - 1 && tablerofila][columna + 1] == ' ') {
60             tablerofila][columna + 1] = '@';
61             tablerofila][columna] = ' ';
62             columna++;
63         } else if (dirección == 'O' && columna > 0 && tablerofila][columna - 1] == ' ') {
64             tablerofila][columna - 1] = '@';
65             tablerofila][columna] = ' ';
66             columna--;
67         } else {
68             System.out.println("Movimiento no válido");
69         }
70     }
71
72     public void moverFantasma(int fila, int columna, char dirección) {
73         if (dirección == 'N' && fila > 0 && tablerofila - 1][columna] == ' ') {
74             tablerofila - 1][columna] = '@';
75             tablerofila][columna] = ' ';
76             fila--;
77         } else if (dirección == 'S' && fila < filas - 1 && tablerofila + 1][columna] == ' ') {
78             tablerofila + 1][columna] = '@';
79             tablerofila][columna] = ' ';
80             fila++;
81         } else if (dirección == 'E' && columna < columnas - 1 && tablerofila][columna + 1] == ' ') {
82             tablerofila][columna + 1] = '@';
83             tablerofila][columna] = ' ';
84             columna++;
85         } else if (dirección == 'O' && columna > 0 && tablerofila][columna - 1] == ' ') {
86             tablerofila][columna - 1] = '@';
87             tablerofila][columna] = ' ';
88             columna--;
89         } else {
90             System.out.println("Movimiento no válido");
91         }
92     }
93
94     public void colisionPacmanFantasma() {
95         for (int i = 0; i < filas; i++) {
96             for (int j = 0; j < columnas; j++) {
97                 if (tablerofila][columna] == '@' && tablerofila][columna] == 'X') {
98                     System.out.println("¡Ganaste!");
99                     System.exit(0);
100                }
101            }
102        }
103    }
104
105    public void colisionPacmanPared() {
106        for (int i = 0; i < filas; i++) {
107            for (int j = 0; j < columnas; j++) {
108                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
109                    System.out.println("¡Perdiste!");
110                    System.exit(0);
111                }
112            }
113        }
114    }
115
116    public void colisionFantasmaPared() {
117        for (int i = 0; i < filas; i++) {
118            for (int j = 0; j < columnas; j++) {
119                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
120                    System.out.println("¡Perdiste!");
121                    System.exit(0);
122                }
123            }
124        }
125    }
126
127    public void colisionFantasmaPacman() {
128        for (int i = 0; i < filas; i++) {
129            for (int j = 0; j < columnas; j++) {
130                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
131                    System.out.println("¡Perdiste!");
132                    System.exit(0);
133                }
134            }
135        }
136    }
137
138    public void colisionFantasmaFantasma() {
139        for (int i = 0; i < filas; i++) {
140            for (int j = 0; j < columnas; j++) {
141                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
142                    System.out.println("¡Perdiste!");
143                    System.exit(0);
144                }
145            }
146        }
147    }
148
149    public void colisionPacmanPunto() {
150        for (int i = 0; i < filas; i++) {
151            for (int j = 0; j < columnas; j++) {
152                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
153                    System.out.println("¡Ganaste!");
154                    System.exit(0);
155                }
156            }
157        }
158    }
159
160    public void colisionFantasmaPunto() {
161        for (int i = 0; i < filas; i++) {
162            for (int j = 0; j < columnas; j++) {
163                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
164                    System.out.println("¡Ganaste!");
165                    System.exit(0);
166                }
167            }
168        }
169    }
170
171    public void colisionPuntoPunto() {
172        for (int i = 0; i < filas; i++) {
173            for (int j = 0; j < columnas; j++) {
174                if (tablerofila][columna] == '$' && tablerofila][columna] == '$') {
175                    System.out.println("¡Ganaste!");
176                    System.exit(0);
177                }
178            }
179        }
180    }
181
182    public void colisionPuntoPared() {
183        for (int i = 0; i < filas; i++) {
184            for (int j = 0; j < columnas; j++) {
185                if (tablerofila][columna] == '$' && tablerofila][columna] == '#') {
186                    System.out.println("¡Perdiste!");
187                    System.exit(0);
188                }
189            }
190        }
191    }
192
193    public void colisionPuntoFantasma() {
194        for (int i = 0; i < filas; i++) {
195            for (int j = 0; j < columnas; j++) {
196                if (tablerofila][columna] == '$' && tablerofila][columna] == '@') {
197                    System.out.println("¡Perdiste!");
198                    System.exit(0);
199                }
200            }
201        }
202    }
203
204    public void colisionFantasmaFantasmaPunto() {
205        for (int i = 0; i < filas; i++) {
206            for (int j = 0; j < columnas; j++) {
207                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
208                    System.out.println("¡Ganaste!");
209                    System.exit(0);
210                }
211            }
212        }
213    }
214
215    public void colisionFantasmaFantasmaPared() {
216        for (int i = 0; i < filas; i++) {
217            for (int j = 0; j < columnas; j++) {
218                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
219                    System.out.println("¡Perdiste!");
220                    System.exit(0);
221                }
222            }
223        }
224    }
225
226    public void colisionFantasmaFantasmaFantasma() {
227        for (int i = 0; i < filas; i++) {
228            for (int j = 0; j < columnas; j++) {
229                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
230                    System.out.println("¡Perdiste!");
231                    System.exit(0);
232                }
233            }
234        }
235    }
236
237    public void colisionFantasmaFantasmaFantasmaPunto() {
238        for (int i = 0; i < filas; i++) {
239            for (int j = 0; j < columnas; j++) {
240                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
241                    System.out.println("¡Ganaste!");
242                    System.exit(0);
243                }
244            }
245        }
246    }
247
248    public void colisionFantasmaFantasmaFantasmaPared() {
249        for (int i = 0; i < filas; i++) {
250            for (int j = 0; j < columnas; j++) {
251                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
252                    System.out.println("¡Perdiste!");
253                    System.exit(0);
254                }
255            }
256        }
257    }
258
259    public void colisionFantasmaFantasmaFantasmaFantasma() {
260        for (int i = 0; i < filas; i++) {
261            for (int j = 0; j < columnas; j++) {
262                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
263                    System.out.println("¡Perdiste!");
264                    System.exit(0);
265                }
266            }
267        }
268    }
269
270    public void colisionFantasmaFantasmaFantasmaFantasmaPunto() {
271        for (int i = 0; i < filas; i++) {
272            for (int j = 0; j < columnas; j++) {
273                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
274                    System.out.println("¡Ganaste!");
275                    System.exit(0);
276                }
277            }
278        }
279    }
280
281    public void colisionFantasmaFantasmaFantasmaFantasmaPared() {
282        for (int i = 0; i < filas; i++) {
283            for (int j = 0; j < columnas; j++) {
284                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
285                    System.out.println("¡Perdiste!");
286                    System.exit(0);
287                }
288            }
289        }
290    }
291
292    public void colisionFantasmaFantasmaFantasmaFantasmaFantasma() {
293        for (int i = 0; i < filas; i++) {
294            for (int j = 0; j < columnas; j++) {
295                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
296                    System.out.println("¡Perdiste!");
297                    System.exit(0);
298                }
299            }
300        }
301    }
302
303    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
304        for (int i = 0; i < filas; i++) {
305            for (int j = 0; j < columnas; j++) {
306                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
307                    System.out.println("¡Ganaste!");
308                    System.exit(0);
309                }
310            }
311        }
312    }
313
314    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
315        for (int i = 0; i < filas; i++) {
316            for (int j = 0; j < columnas; j++) {
317                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
318                    System.out.println("¡Perdiste!");
319                    System.exit(0);
320                }
321            }
322        }
323    }
324
325    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
326        for (int i = 0; i < filas; i++) {
327            for (int j = 0; j < columnas; j++) {
328                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
329                    System.out.println("¡Perdiste!");
330                    System.exit(0);
331                }
332            }
333        }
334    }
335
336    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
337        for (int i = 0; i < filas; i++) {
338            for (int j = 0; j < columnas; j++) {
339                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
340                    System.out.println("¡Ganaste!");
341                    System.exit(0);
342                }
343            }
344        }
345    }
346
347    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
348        for (int i = 0; i < filas; i++) {
349            for (int j = 0; j < columnas; j++) {
350                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
351                    System.out.println("¡Perdiste!");
352                    System.exit(0);
353                }
354            }
355        }
356    }
357
358    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
359        for (int i = 0; i < filas; i++) {
360            for (int j = 0; j < columnas; j++) {
361                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
362                    System.out.println("¡Perdiste!");
363                    System.exit(0);
364                }
365            }
366        }
367    }
368
369    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
370        for (int i = 0; i < filas; i++) {
371            for (int j = 0; j < columnas; j++) {
372                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
373                    System.out.println("¡Ganaste!");
374                    System.exit(0);
375                }
376            }
377        }
378    }
379
380    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
381        for (int i = 0; i < filas; i++) {
382            for (int j = 0; j < columnas; j++) {
383                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
384                    System.out.println("¡Perdiste!");
385                    System.exit(0);
386                }
387            }
388        }
389    }
390
391    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
392        for (int i = 0; i < filas; i++) {
393            for (int j = 0; j < columnas; j++) {
394                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
395                    System.out.println("¡Perdiste!");
396                    System.exit(0);
397                }
398            }
399        }
400    }
401
402    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
403        for (int i = 0; i < filas; i++) {
404            for (int j = 0; j < columnas; j++) {
405                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
406                    System.out.println("¡Ganaste!");
407                    System.exit(0);
408                }
409            }
410        }
411    }
412
413    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
414        for (int i = 0; i < filas; i++) {
415            for (int j = 0; j < columnas; j++) {
416                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
417                    System.out.println("¡Perdiste!");
418                    System.exit(0);
419                }
420            }
421        }
422    }
423
424    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
425        for (int i = 0; i < filas; i++) {
426            for (int j = 0; j < columnas; j++) {
427                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
428                    System.out.println("¡Perdiste!");
429                    System.exit(0);
430                }
431            }
432        }
433    }
434
435    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
436        for (int i = 0; i < filas; i++) {
437            for (int j = 0; j < columnas; j++) {
438                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
439                    System.out.println("¡Ganaste!");
440                    System.exit(0);
441                }
442            }
443        }
444    }
445
446    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
447        for (int i = 0; i < filas; i++) {
448            for (int j = 0; j < columnas; j++) {
449                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
450                    System.out.println("¡Perdiste!");
451                    System.exit(0);
452                }
453            }
454        }
455    }
456
457    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
458        for (int i = 0; i < filas; i++) {
459            for (int j = 0; j < columnas; j++) {
460                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
461                    System.out.println("¡Perdiste!");
462                    System.exit(0);
463                }
464            }
465        }
466    }
467
468    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
469        for (int i = 0; i < filas; i++) {
470            for (int j = 0; j < columnas; j++) {
471                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
472                    System.out.println("¡Ganaste!");
473                    System.exit(0);
474                }
475            }
476        }
477    }
478
479    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
480        for (int i = 0; i < filas; i++) {
481            for (int j = 0; j < columnas; j++) {
482                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
483                    System.out.println("¡Perdiste!");
484                    System.exit(0);
485                }
486            }
487        }
488    }
489
490    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
491        for (int i = 0; i < filas; i++) {
492            for (int j = 0; j < columnas; j++) {
493                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
494                    System.out.println("¡Perdiste!");
495                    System.exit(0);
496                }
497            }
498        }
499    }
500
501    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
502        for (int i = 0; i < filas; i++) {
503            for (int j = 0; j < columnas; j++) {
504                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
505                    System.out.println("¡Ganaste!");
506                    System.exit(0);
507                }
508            }
509        }
510    }
511
512    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
513        for (int i = 0; i < filas; i++) {
514            for (int j = 0; j < columnas; j++) {
515                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
516                    System.out.println("¡Perdiste!");
517                    System.exit(0);
518                }
519            }
520        }
521    }
522
523    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
524        for (int i = 0; i < filas; i++) {
525            for (int j = 0; j < columnas; j++) {
526                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
527                    System.out.println("¡Perdiste!");
528                    System.exit(0);
529                }
530            }
531        }
532    }
533
534    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
535        for (int i = 0; i < filas; i++) {
536            for (int j = 0; j < columnas; j++) {
537                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
538                    System.out.println("¡Ganaste!");
539                    System.exit(0);
540                }
541            }
542        }
543    }
544
545    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
546        for (int i = 0; i < filas; i++) {
547            for (int j = 0; j < columnas; j++) {
548                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
549                    System.out.println("¡Perdiste!");
550                    System.exit(0);
551                }
552            }
553        }
554    }
555
556    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
557        for (int i = 0; i < filas; i++) {
558            for (int j = 0; j < columnas; j++) {
559                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
560                    System.out.println("¡Perdiste!");
561                    System.exit(0);
562                }
563            }
564        }
565    }
566
567    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
568        for (int i = 0; i < filas; i++) {
569            for (int j = 0; j < columnas; j++) {
570                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
571                    System.out.println("¡Ganaste!");
572                    System.exit(0);
573                }
574            }
575        }
576    }
577
578    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
579        for (int i = 0; i < filas; i++) {
580            for (int j = 0; j < columnas; j++) {
581                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
582                    System.out.println("¡Perdiste!");
583                    System.exit(0);
584                }
585            }
586        }
587    }
588
589    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
590        for (int i = 0; i < filas; i++) {
591            for (int j = 0; j < columnas; j++) {
592                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
593                    System.out.println("¡Perdiste!");
594                    System.exit(0);
595                }
596            }
597        }
598    }
599
600    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
601        for (int i = 0; i < filas; i++) {
602            for (int j = 0; j < columnas; j++) {
603                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
604                    System.out.println("¡Ganaste!");
605                    System.exit(0);
606                }
607            }
608        }
609    }
610
611    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
612        for (int i = 0; i < filas; i++) {
613            for (int j = 0; j < columnas; j++) {
614                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
615                    System.out.println("¡Perdiste!");
616                    System.exit(0);
617                }
618            }
619        }
620    }
621
622    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
623        for (int i = 0; i < filas; i++) {
624            for (int j = 0; j < columnas; j++) {
625                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
626                    System.out.println("¡Perdiste!");
627                    System.exit(0);
628                }
629            }
630        }
631    }
632
633    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
634        for (int i = 0; i < filas; i++) {
635            for (int j = 0; j < columnas; j++) {
636                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
637                    System.out.println("¡Ganaste!");
638                    System.exit(0);
639                }
640            }
641        }
642    }
643
644    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
645        for (int i = 0; i < filas; i++) {
646            for (int j = 0; j < columnas; j++) {
647                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
648                    System.out.println("¡Perdiste!");
649                    System.exit(0);
650                }
651            }
652        }
653    }
654
655    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
656        for (int i = 0; i < filas; i++) {
657            for (int j = 0; j < columnas; j++) {
658                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
659                    System.out.println("¡Perdiste!");
660                    System.exit(0);
661                }
662            }
663        }
664    }
665
666    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
667        for (int i = 0; i < filas; i++) {
668            for (int j = 0; j < columnas; j++) {
669                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
670                    System.out.println("¡Ganaste!");
671                    System.exit(0);
672                }
673            }
674        }
675    }
676
677    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
678        for (int i = 0; i < filas; i++) {
679            for (int j = 0; j < columnas; j++) {
680                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
681                    System.out.println("¡Perdiste!");
682                    System.exit(0);
683                }
684            }
685        }
686    }
687
688    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
689        for (int i = 0; i < filas; i++) {
690            for (int j = 0; j < columnas; j++) {
691                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
692                    System.out.println("¡Perdiste!");
693                    System.exit(0);
694                }
695            }
696        }
697    }
698
699    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
700        for (int i = 0; i < filas; i++) {
701            for (int j = 0; j < columnas; j++) {
702                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
703                    System.out.println("¡Ganaste!");
704                    System.exit(0);
705                }
706            }
707        }
708    }
709
710    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
711        for (int i = 0; i < filas; i++) {
712            for (int j = 0; j < columnas; j++) {
713                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
714                    System.out.println("¡Perdiste!");
715                    System.exit(0);
716                }
717            }
718        }
719    }
720
721    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
722        for (int i = 0; i < filas; i++) {
723            for (int j = 0; j < columnas; j++) {
724                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
725                    System.out.println("¡Perdiste!");
726                    System.exit(0);
727                }
728            }
729        }
730    }
731
732    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
733        for (int i = 0; i < filas; i++) {
734            for (int j = 0; j < columnas; j++) {
735                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
736                    System.out.println("¡Ganaste!");
737                    System.exit(0);
738                }
739            }
740        }
741    }
742
743    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
744        for (int i = 0; i < filas; i++) {
745            for (int j = 0; j < columnas; j++) {
746                if (tablerofila][columna] == '@' && tablerofila][columna] == '#') {
747                    System.out.println("¡Perdiste!");
748                    System.exit(0);
749                }
750            }
751        }
752    }
753
754    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasma() {
755        for (int i = 0; i < filas; i++) {
756            for (int j = 0; j < columnas; j++) {
757                if (tablerofila][columna] == '@' && tablerofila][columna] == '@') {
758                    System.out.println("¡Perdiste!");
759                    System.exit(0);
760                }
761            }
762        }
763    }
764
765    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPunto() {
766        for (int i = 0; i < filas; i++) {
767            for (int j = 0; j < columnas; j++) {
768                if (tablerofila][columna] == '@' && tablerofila][columna] == '$') {
769                    System.out.println("¡Ganaste!");
770                    System.exit(0);
771                }
772            }
773        }
774    }
775
776    public void colisionFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaFantasmaPared() {
777        for (int i = 0; i < filas; i++) {
778            for (int j = 0; j < columnas; j++) {
779                if (tablerofila][columna] == '@' &&amp
```

```

41  private void colocarPremios(int total) {
42      int simples = (int)(total * 0.8);
43      for (int i = 0; i < simples; i++) {
44          colocarElementos("0", 1);
45      }
46      for (int i = 0; i < total - simples; i++) {
47          colocarElementos("$", 1);
48      }
49  }
50
51  public void mostrar() {
52      String lineaGuiones = "-".repeat(columnas * 3 + 1);
53
54      System.out.println(lineaGuiones);
55
56      for (int i = 0; i < filas; i++) {
57          System.out.print("|");
58          for (int j = 0; j < columnas; j++) {
59              System.out.print(" " + tablero[i][j] + " ");
60          }
61          System.out.println("|");
62      }
63
64      System.out.println(lineaGuiones);
65  }
66
67  public boolean colocarPacman(int fila, int columna) {
68      if (fila >= 0 && fila < filas && columna >= 0 && columna < columnas &&
69      tablero[fila][columna] == ' ') {
70          tablero[fila][columna] = '<';
71          return true;
72      }
73      return false;
74  }
75
76  public void moverPacman(int filaActual, int columnaActual, int nuevaFila, int nuevaColumna) {
77
78      nuevaFila = (nuevaFila + filas) % filas;
79      nuevaColumna = (nuevaColumna + columnas) % columnas;
80
81      char contenido = tablero[nuevaFila][nuevaColumna];
82
83      if (contenido != 'X') {
84          tablero[filaActual][columnaActual] = ' ';
85          tablero[nuevaFila][nuevaColumna] = '<';
86      }
87
88  public char getContenido(int fila, int columna) {
89      fila = (fila + filas) % filas;
90      columna = (columna + columnas) % columnas;
91      return tablero[fila][columna];
92  }
93
94  public void limpiarCasilla(int fila, int columna) {
95      fila = (fila + filas) % filas;
96      columna = (columna + columnas) % columnas;
97      tablero[fila][columna] = ' ';
98  }
99
100 public boolean hayPremio(int fila, int columna) {
101     char c = getContenido(fila, columna);
102     return c == '0' || c == '$';
103 }
104
105 public int getPuntosPremio(int fila, int columna) {
106     char premio = getContenido(fila, columna);
107     if (premio == '0') {
108         premiosRestantes--;
109         return 10;
110     } else if (premio == '$') {
111         premiosRestantes--;
112         return 15;
113     }

```

```

112     }
113     return 15;
114   }
115 }
116
117 public boolean hayFantasma(int fila, int columna) {
118   return getContenido(fila, columna) == '@';
119 }
120
121 public boolean hayPared(int fila, int columna) {
122   return getContenido(fila, columna) == 'X';
123 }
124
125 public int getFilas() { return filas; }
126 public int getColumnas() { return columnas; }
127 public int getPremiosRestantes() { return premiosRestantes; }
128 }
```

→ Partida.java

```

1 package pacman;
2
3 import java.util.Scanner;
4
5 public class Partida {
6   private String nombreUsuario;
7   private Tablero tablero;
8   private int filaPacman;
9   private int columnaPacman;
10  private int vidas;
11  private int puntaje;
12  private boolean juegoActivo;
13  private Historial historial;
14  private Scanner scanner = new Scanner(System.in);
15
16  public Partida(String nombreUsuario, Tablero tablero, int filaInicial, int columnaInicial, Historial historial) {
17    this.nombreUsuario = nombreUsuario;
18    this.tablero = tablero;
19    this.filaPacman = filaInicial;
20    this.columnaPacman = columnaInicial;
21    this.vidas = 3;
22    this.puntaje = 0;
23    this.juegoActivo = true;
24    this.historial = historial;
25
26    if (!tablero.colocarPacman(filaInicial, columnaInicial)) {
27      colocarPacmanEnPosicionLibre();
28    }
29  }
30
31  private void colocarPacmanEnPosicionLibre() {
32    for (int i = 0; i < tablero.getFilas(); i++) {
33      for (int j = 0; j < tablero.getColumnas(); j++) {
34        if (tablero.getContenido(i, j) == ' ') {
35          filaPacman = i;
36          columnaPacman = j;
37          tablero.colocarPacman(i, j);
38        }
39      }
40    }
41  }
42}
```

```

37             tablero.colocarPacman(i, j);
38         }
39     }
40 }
41 }
42 }
43
44 public void jugar() {
45     while (juegoActivo) {
46         mostrarEstado();
47         System.out.println("8: ARRIBA      5: ABAJO      6: DERECHA      4: IZQUIERDA      F: PAUSA");
48         System.out.print("MOVIMIENTO: ");
49         String entrada = scanner.next();
50         char tecla = entrada.charAt(0);
51
52         if (tecla == 'F' || tecla == 'f') {
53             mostrarMenuPausa();
54         } else {
55             procesarMovimiento(tecla);
56         }
57
58         if (tablero.getPremiosRestantes() == 0) {
59             System.out.println(";FELICIDADES! Has ganado la partida.");
60             juegoActivo = false;
61         }
62
63         if (vidas == 0) {
64             System.out.println(";GAME OVER!");
65             juegoActivo = false;
66         }
67     }
68
69     historial.agregarPartida(nombreUsuario, puntaje);
70 }
71
72 private void mostrarEstado() {
73     System.out.println("\nUSUARIO: " + nombreUsuario);

73     System.out.println("\nUSUARIO: " + nombreUsuario);
74     System.out.println("PUNTEO: " + puntaje);
75     System.out.println("VIDAS: " + vidas);
76     tablero.mostrar();
77 }

78
79 private void procesarMovimiento(char tecla) {
80     int nuevaFila = filaPacman;
81     int nuevaColumna = columnaPacman;
82
83     switch (tecla) {
84         case '8': nuevaFila--; break;
85         case '5': nuevaFila++; break;
86         case '6': nuevaColumna++; break;
87         case '4': nuevaColumna--; break;
88         default:
89             System.out.println("Tecla inválida");
90             return;
91     }
92
93     nuevaFila = (nuevaFila + tablero.getFilas()) % tablero.getFilas();
94     nuevaColumna = (nuevaColumna + tablero.getColumnas()) % tablero.getColumnas();
95
96     if (tablero.hayPared(nuevaFila, nuevaColumna)) {
97         System.out.println(";Hay una pared! No puedes pasar.");
98         return;
99     }
100
101     if (tablero.hayFantasma(nuevaFila, nuevaColumna)) {
102         vidas--;
103         System.out.println(";Te encuentras con un fantasma! Vidas restantes: " + vidas);
104         tablero.limpiarCasilla(nuevaFila, nuevaColumna);
105     }
106
107     if (tablero.hayPremio(nuevaFila, nuevaColumna)) {
108         int puntos = tablero.getPuntosPremio(nuevaFila, nuevaColumna);
109         puntaje += puntos;
110     }

```

```

109     }
110     ...
111     ...
112     ...
113     ...
114     ...
115     ...
116     ...
117     ...
118     private void mostrarMenuPausa() {
119         System.out.println("\n**PAUSA**");
120         System.out.println("POR FAVOR, SELECCIONE UNA OPCIÓN");
121         System.out.println("1. REGRESAR");
122         System.out.println("2. TERMINAR PARTIDA");
123         System.out.print("OPCIÓN: ");
124
125         int opcion = scanner.nextInt();
126
127         if (opcion == 2) {
128             juegoActivo = false;
129         }
130     }
131 }
```

→Historial.java

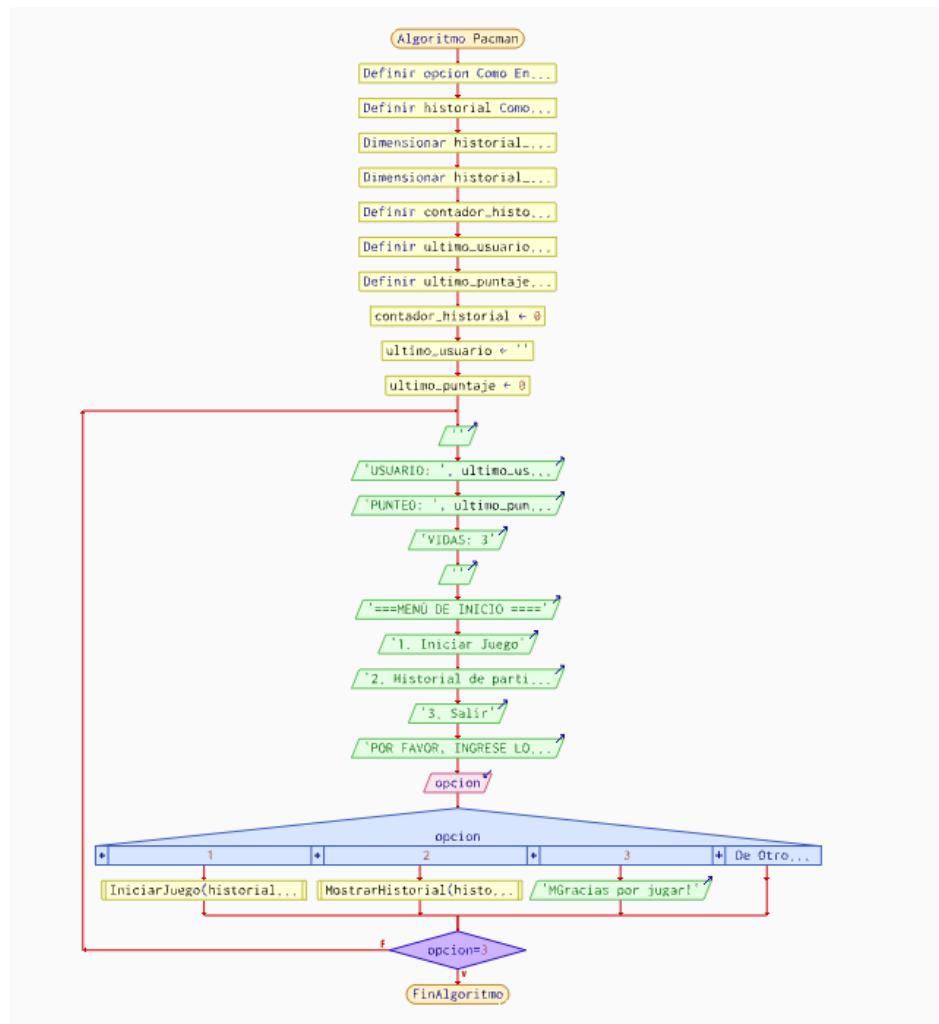
```

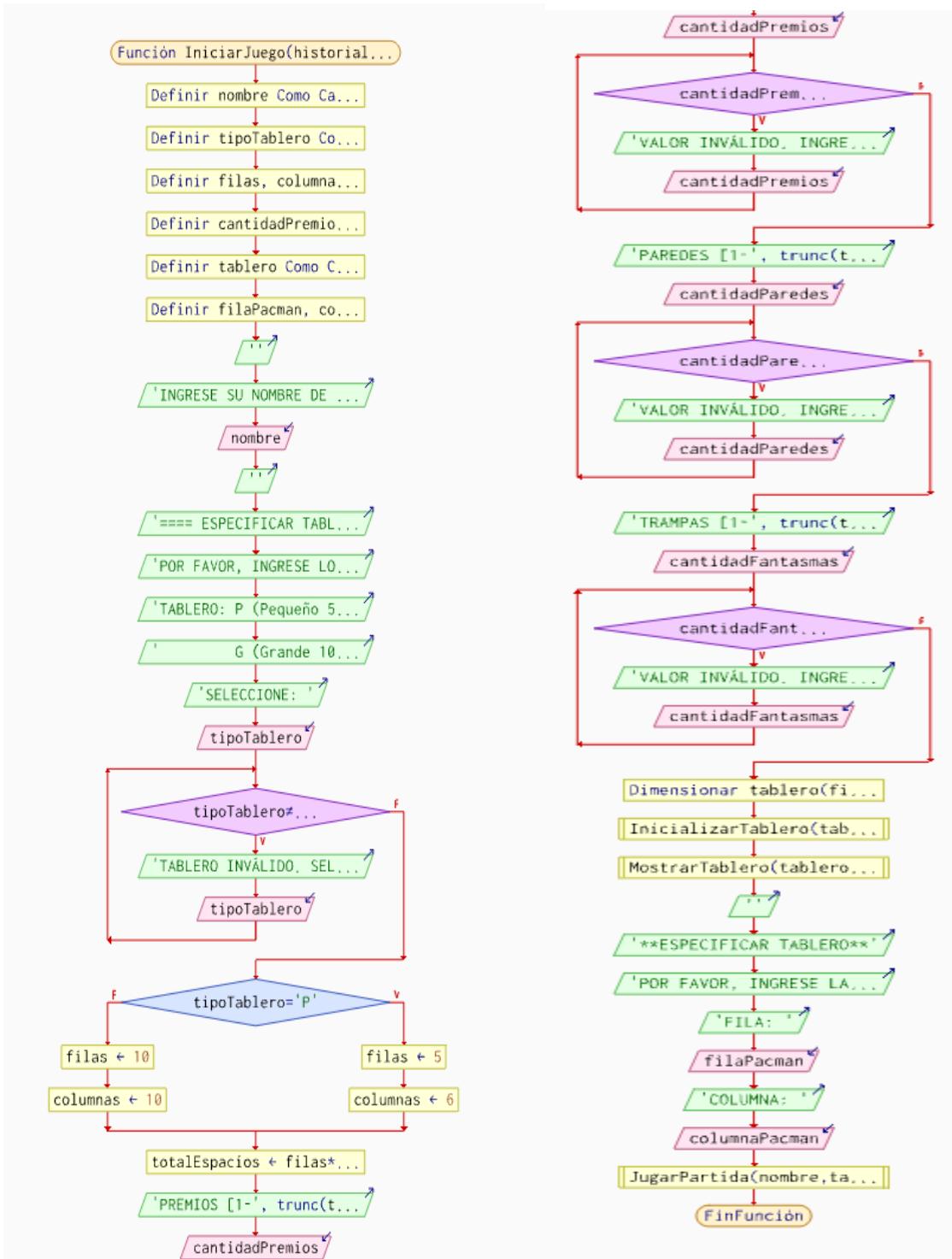
1 package pacman;
2
3 public class Historial {
4     private String[] usuarios;
5     private int[] puntajes;
6     private int contador;
7     private String ultimoUsuario;
8     private int ultimoPuntaje;
9
10    public Historial() {
11        usuarios = new String[100];
12        puntajes = new int[100];
13        contador = 0;
14        ultimoUsuario = "";
15        ultimoPuntaje = 0;
16    }
17
18    public void agregarPartida(String usuario, int puntaje) {
19        if (contador < usuarios.length) {
20            for (int i = contador; i > 0; i--) {
21                usuarios[i] = usuarios[i-1];
22                puntajes[i] = puntajes[i-1];
23            }
24            usuarios[0] = usuario;
25            puntajes[0] = puntaje;
26            contador++;
27            ultimoUsuario = usuario;
28            ultimoPuntaje = puntaje;
29        }
30    }
31
32    public void mostrarHistorial() {
33        if (contador == 0) {
34            System.out.println("\n**HISTORIAL DE PARTIDAS**");
35            System.out.println("No hay partidas en el historial.");
36        }
37    }
}
```

```
37     }
38
39     System.out.println("\n**HISTORIAL DE PARTIDAS**");
40     System.out.println("No.\tUSUARIO\tPUNTEO");
41     System.out.println("-----");
42     for (int i = 0; i < contador; i++) {
43         System.out.println((i+1) + ".\t" + usuarios[i] + "\t" + punajes[i]);
44     }
45 }
46
47 public String getUltimoUsuario() {
48     return contador > 0 ? usuarios[0] : "";
49 }
50
51 public int getUltimoPuntaje() {
52     return contador > 0 ? punajes[0] : 0;
53 }
54 }
```

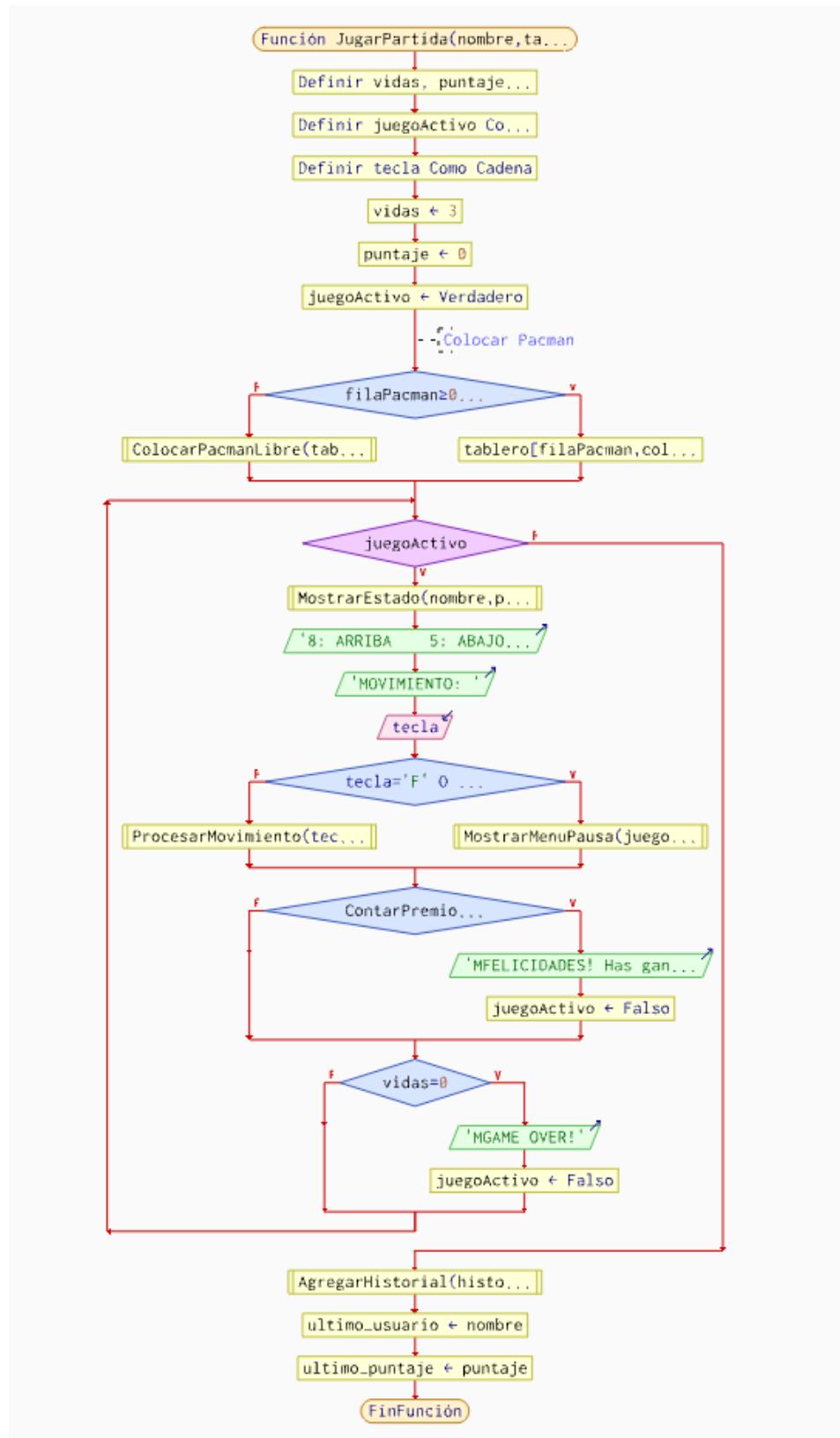
-Diagramas de Flujo

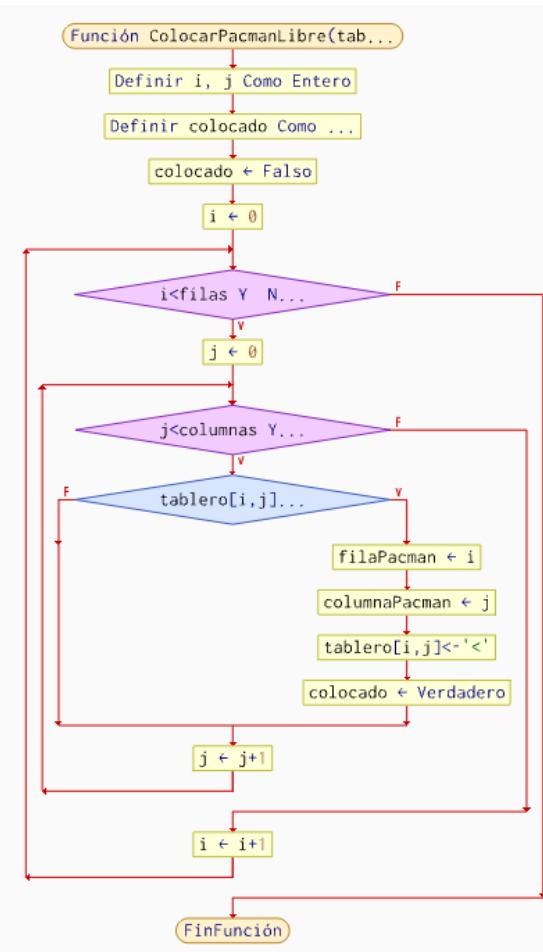
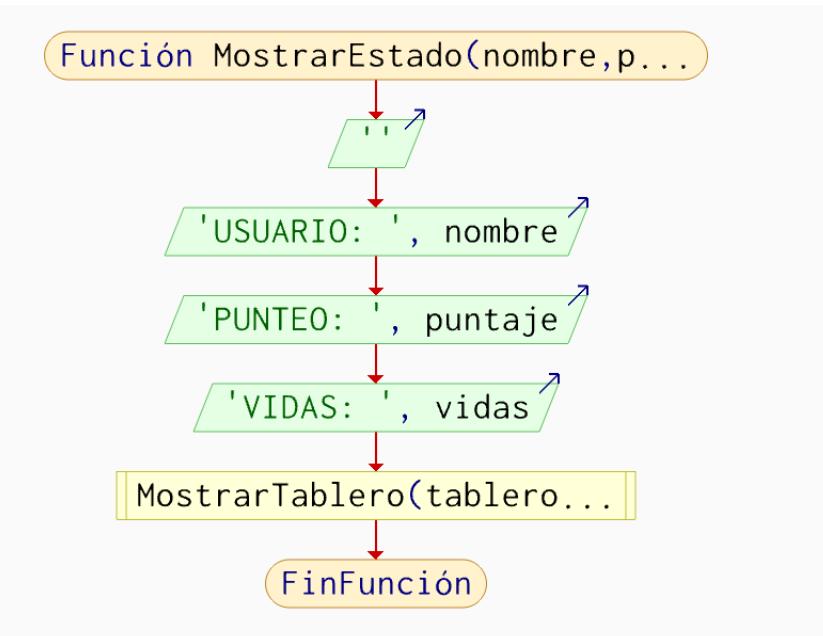
Pacman.java

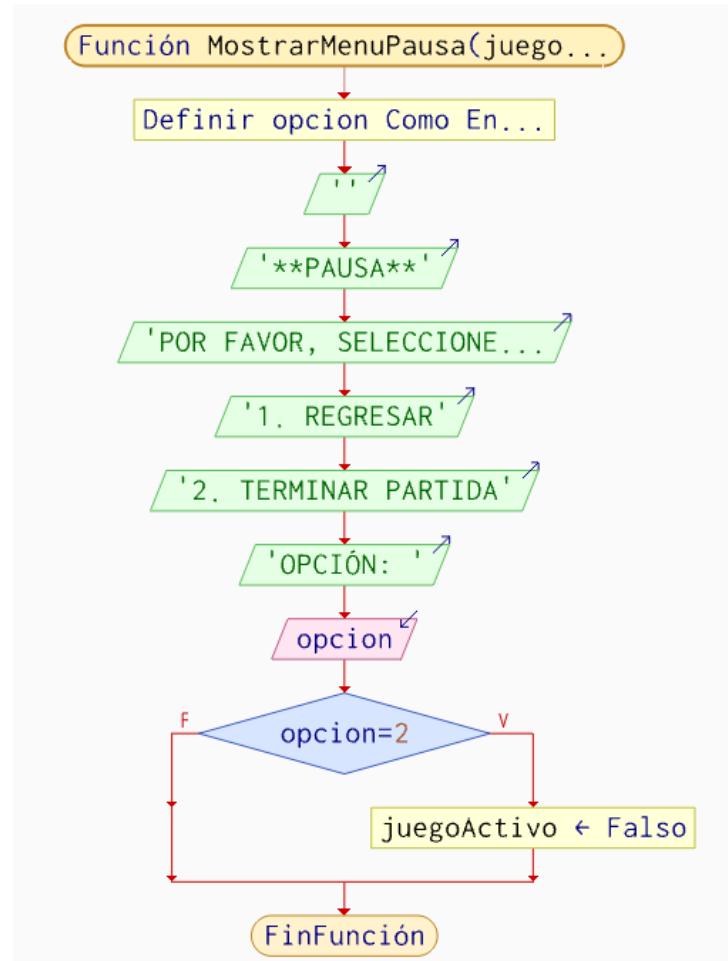




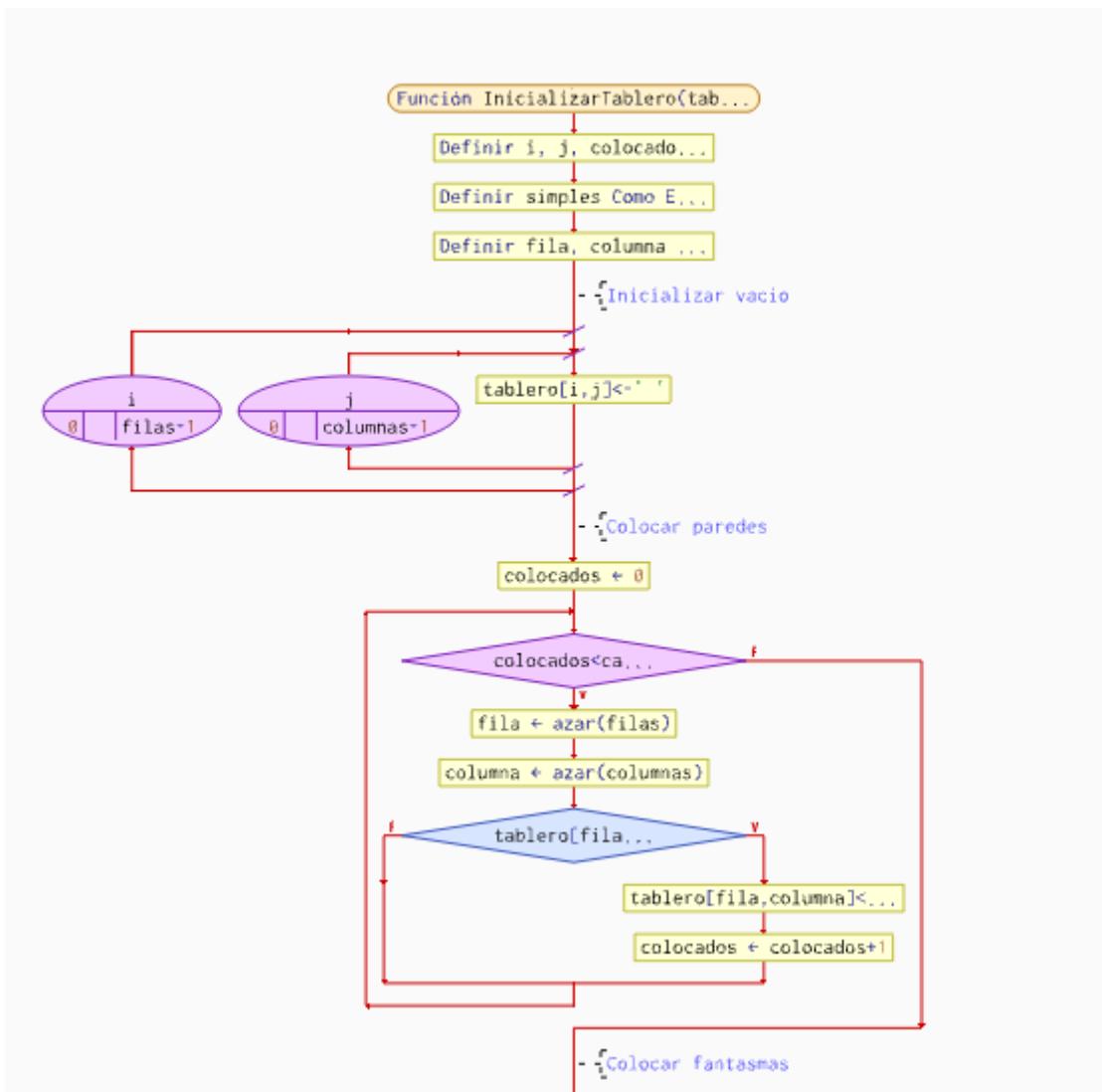
Partida.java

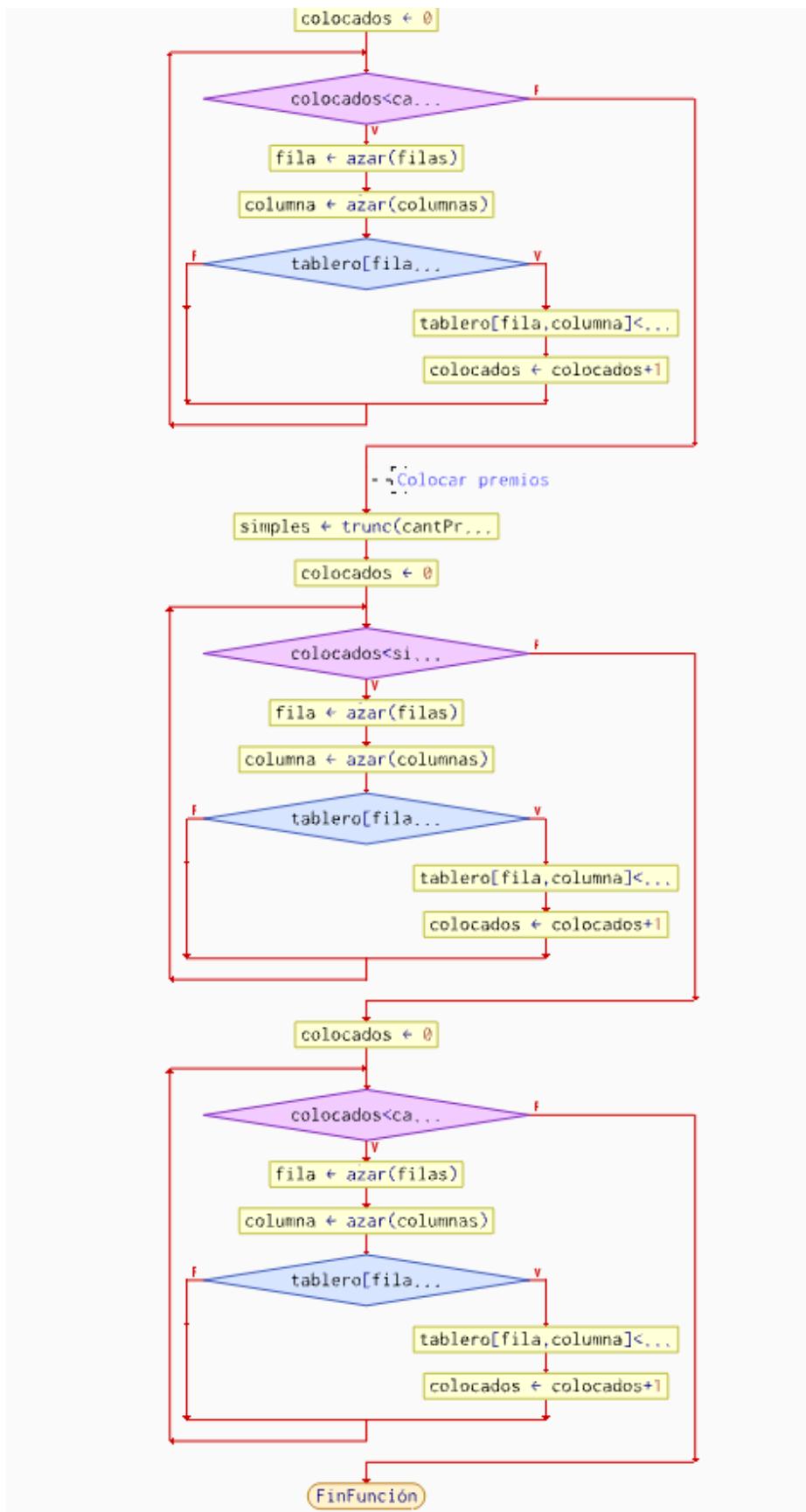


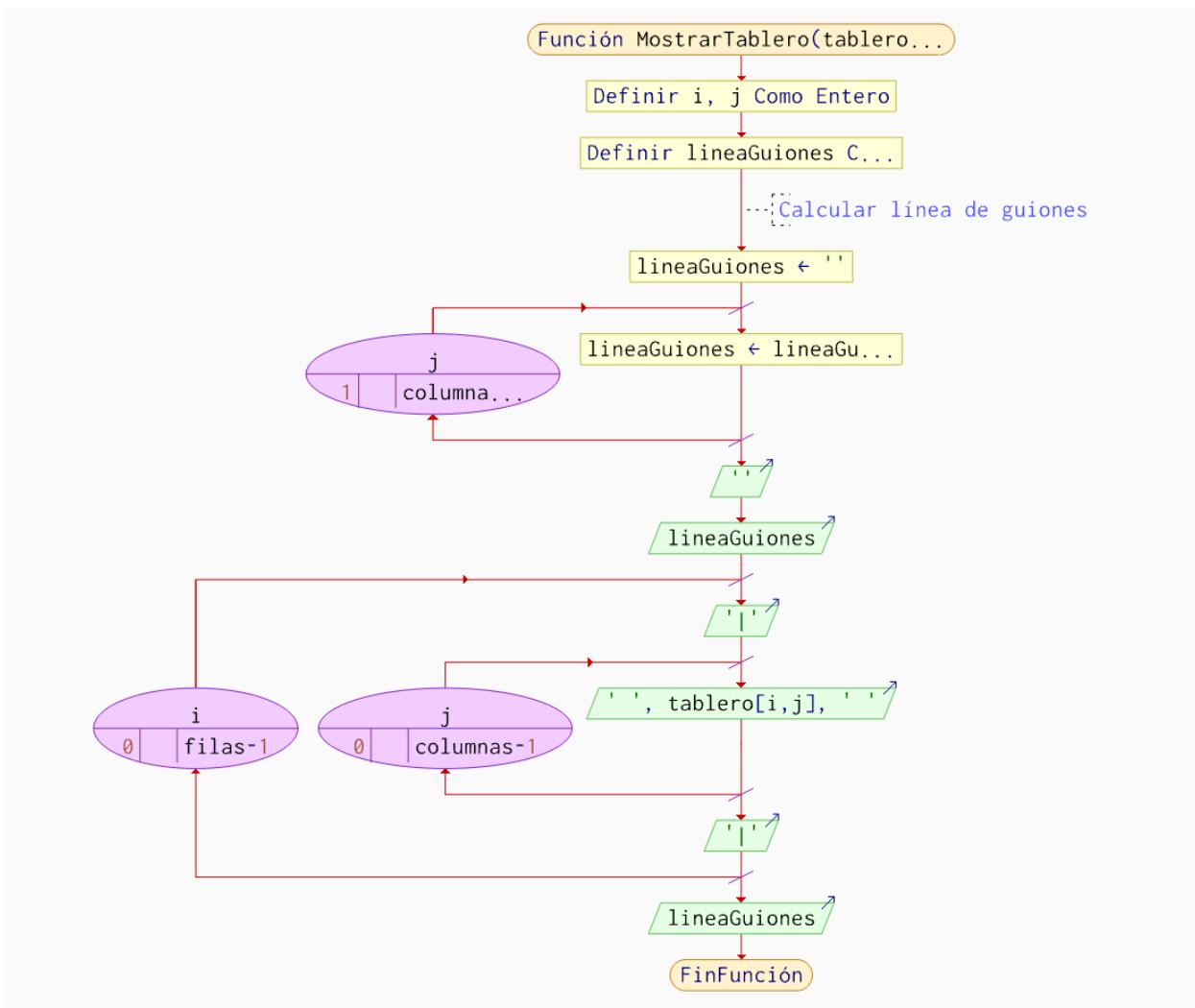
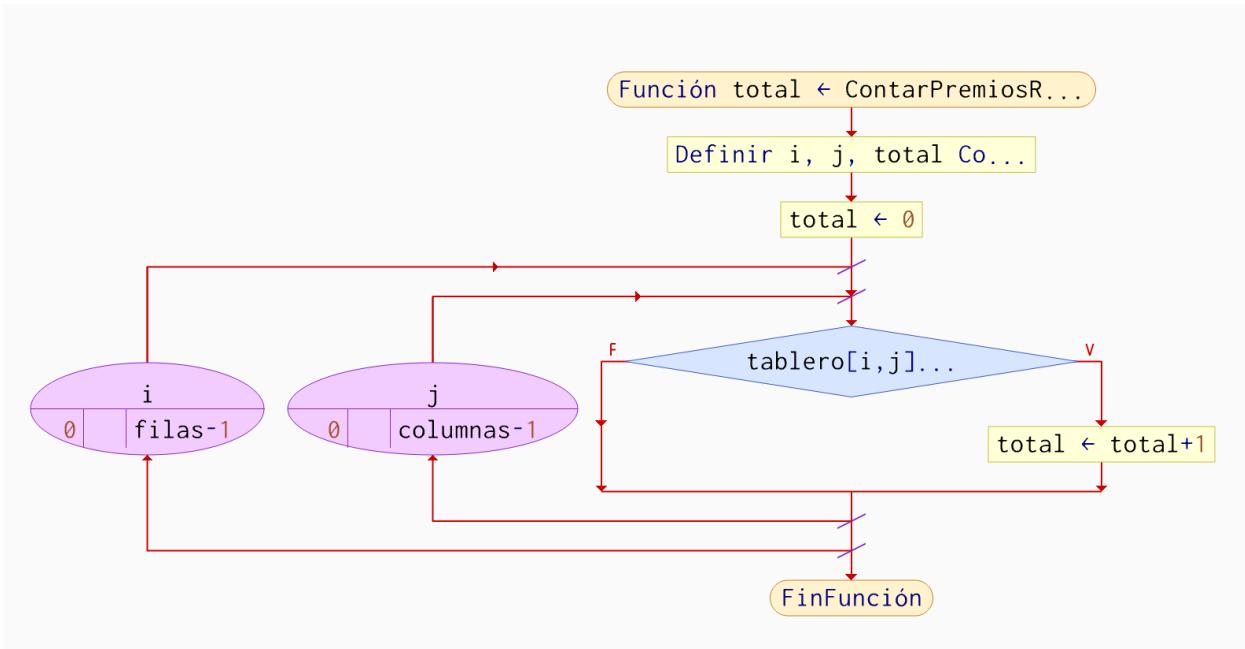




Tablero.java







Historial.java

