



**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 AJSIVINAC 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      0 |
| X      X      0      |
| X      X X      $ 0 X      0 |
|      @ @ 0 X      @      $ |
| @      0 X X      X      X  |
|      @ @ X @ X 0      0 |
|      @ 0 @      @      |
| $      @ @      @ X 0 |
-----
```

\*\*ESPECIFICAR TABLERO\*\*

POR FAVOR, INGRESE LA POSICION INICIAL DE JUEGO

FILA: |

**\*\*ESPECIFICAR TABLERO\*\***

POR FAVOR, INGRESE LA POSICIÓN INICIAL DE JUEGO

FILA: 2

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

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

\*\*PAUSA\*\*  
POR FAVOR, SELECCIONE UNA OPCION  
1. REGRESAR  
2. TERMINAR PARTIDA  
OPCION:

\*\*PAUSA\*\*  
POR FAVOR, SELECCIONE UNA OPCION  
1. REGRESAR  
2. TERMINAR PARTIDA  
OPCION: 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 los 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 }
```

```
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 }
```

```

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 (tablero[fila][columna] == ' ') {
35                 tablero[fila][columna] = elemento;
36                 colocados++;
37             }
38         }
39     }
40 }

```

```

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('$', 1);
48     }
49 }
50
51 public void mostrar() {
52     String lineaGuiones = "-".repeat(columnas * 3 + 1);
53
54     System.out.println(lineaGuiones);
55
56     for (int i = 0; i < filas; i++) {
57         System.out.print("|");
58         for (int j = 0; j < columnas; j++) {
59             System.out.print(" " + tablero[i][j] + " ");
60         }
61         System.out.println("|");
62     }
63
64     System.out.println(lineaGuiones);
65 }
66
67 public boolean colocarPacman(int fila, int columna) {
68     if (fila >= 0 && fila < filas && columna >= 0 && columna < columnas &&
69         tablero[fila][columna] == ' ') {
70         tablero[fila][columna] = '<';
71         return true;
72     }
73     return false;
74 }
75
76 public void moverPacman(int filaActual, int columnaActual, int nuevaFila, int nuevaColumna) {
77     nuevaFila = (nuevaFila + filas) % filas;
78     nuevaColumna = (nuevaColumna + columnas) % columnas;
79
80     char contenido = tablero[nuevaFila][nuevaColumna];
81
82     if (contenido != 'X') {
83         tablero[filaActual][columnaActual] = ' ';
84         tablero[nuevaFila][nuevaColumna] = '<';
85     }
86 }
87
88 public char getContenido(int fila, int columna) {
89     fila = (fila + filas) % filas;
90     columna = (columna + columnas) % columnas;
91     return tablero[fila][columna];
92 }
93
94 public void limpiarCasilla(int fila, int columna) {
95     fila = (fila + filas) % filas;
96     columna = (columna + columnas) % columnas;
97     tablero[fila][columna] = ' ';
98 }
99
100 public boolean hayPremio(int fila, int columna) {
101     char c = getContenido(fila, columna);
102     return c == '0' || c == '$';
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;

```

```

112         return 15;
113     }
114     return 0;
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);

```

```

37         tablero.colocarPacman(i, j);
38         return;
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 encontraste 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;

```

```

109         puntaje += puntos;
110         System.out.println(";Premio! +" + puntos + " puntos");
111     }
112
113     tablero.moverPacman(filaPacman, columnaPacman, nuevaFila, nuevaColumna);
114     filaPacman = nuevaFila;
115     columnaPacman = nuevaColumna;
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

```

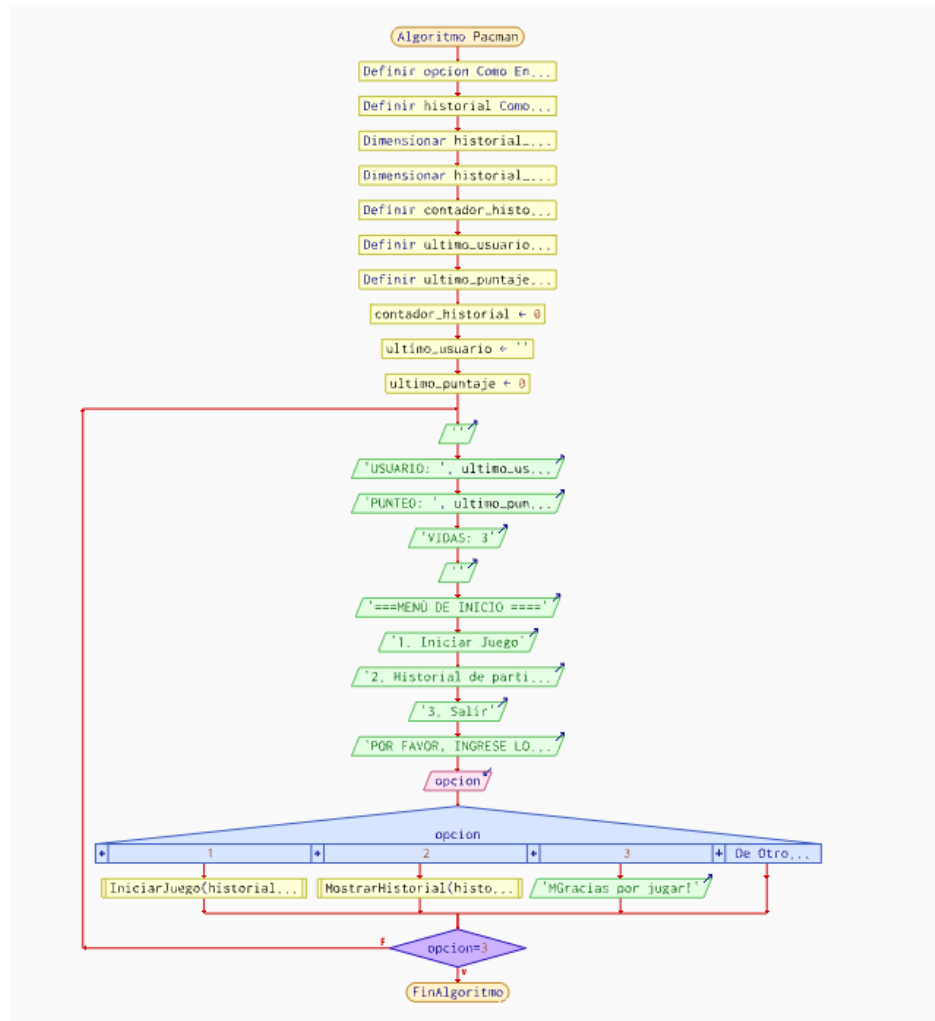
1 package pacman;
2
3 public class Historial {
4     private String[] usuarios;
5     private int[] puntajes;
6     private int contador;
7     private String ultimoUsuario;
8     private int ultimoPuntaje;
9
10    public Historial() {
11        usuarios = new String[100];
12        puntajes = new int[100];
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            return;
37        }
38    }
39 }

```

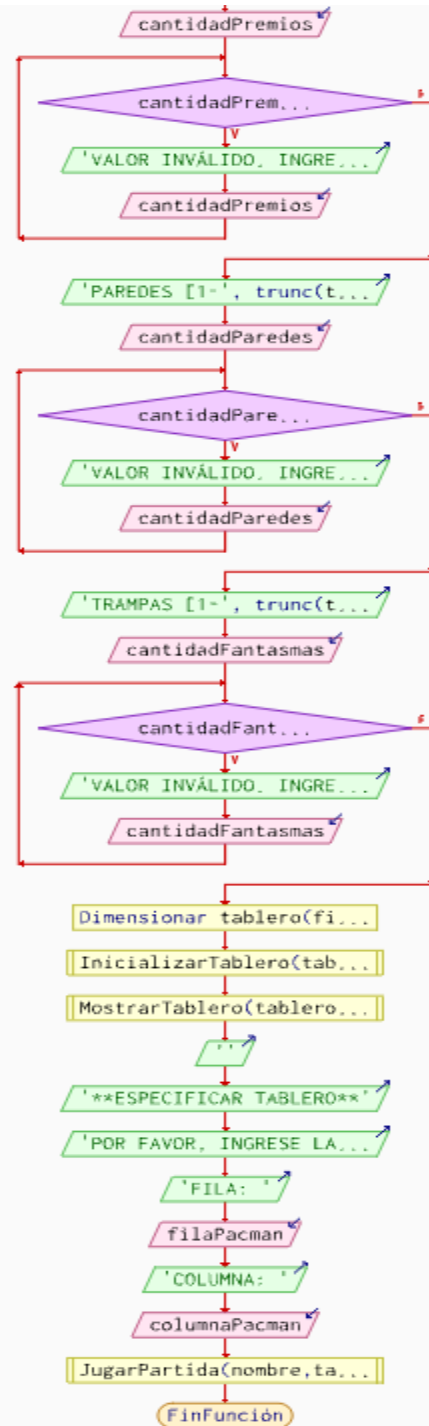
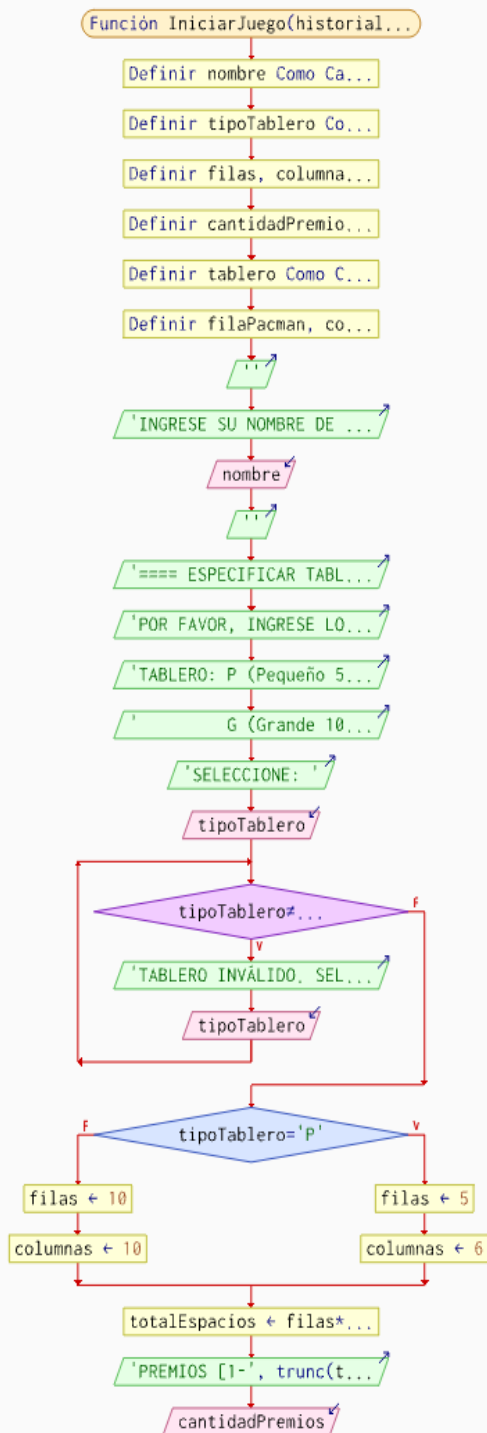
```
37     }
38
39     System.out.println("\n**HISTORIAL DE PARTIDAS**");
40     System.out.println("No.\tUSUARIO\t\tPUNTEO");
41     System.out.println("-----");
42     for (int i = 0; i < contador; i++) {
43         System.out.println((i+1) + ".\t" + usuarios[i] + "\t\t" + puntajes[i]);
44     }
45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

# -Diagramas de Flujo

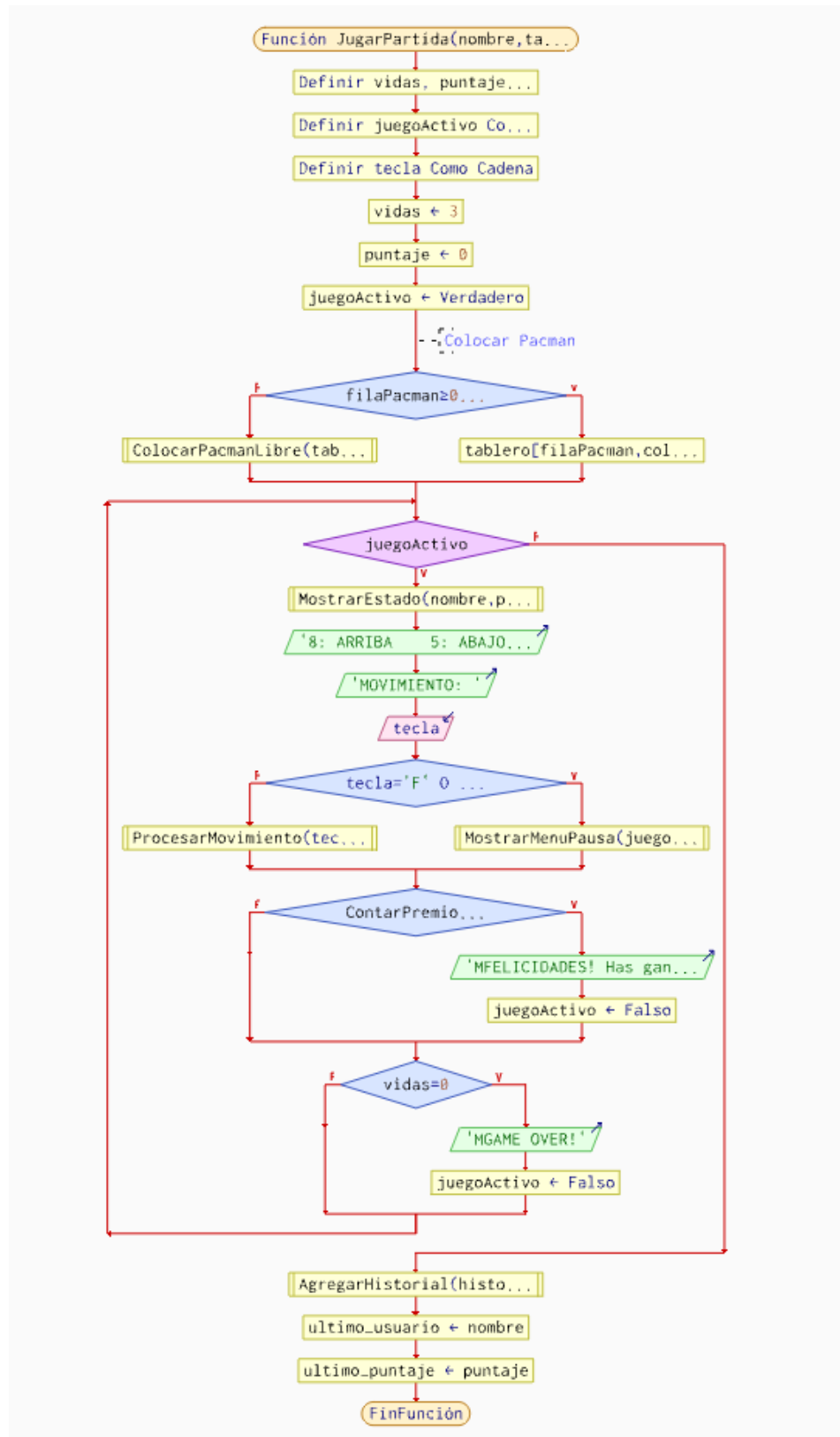
## Pacman.Java



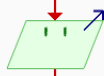




# Partida.Java



Función MostrarEstado(nombre,p...



'USUARIO: ', nombre

'PUNTEO: ', puntaje

'VIDAS: ', vidas

MostrarTablero(tablero...

FinFunción

Función ColocarPacmanLibre(tab...

Definir i, j Como Entero

Definir colocado Como ...

colocado ← Falso

i ← 0

i < filas Y N...

j ← 0

j < columnas Y...

tablero[i,j]...

filaPacman ← i

columnaPacman ← j

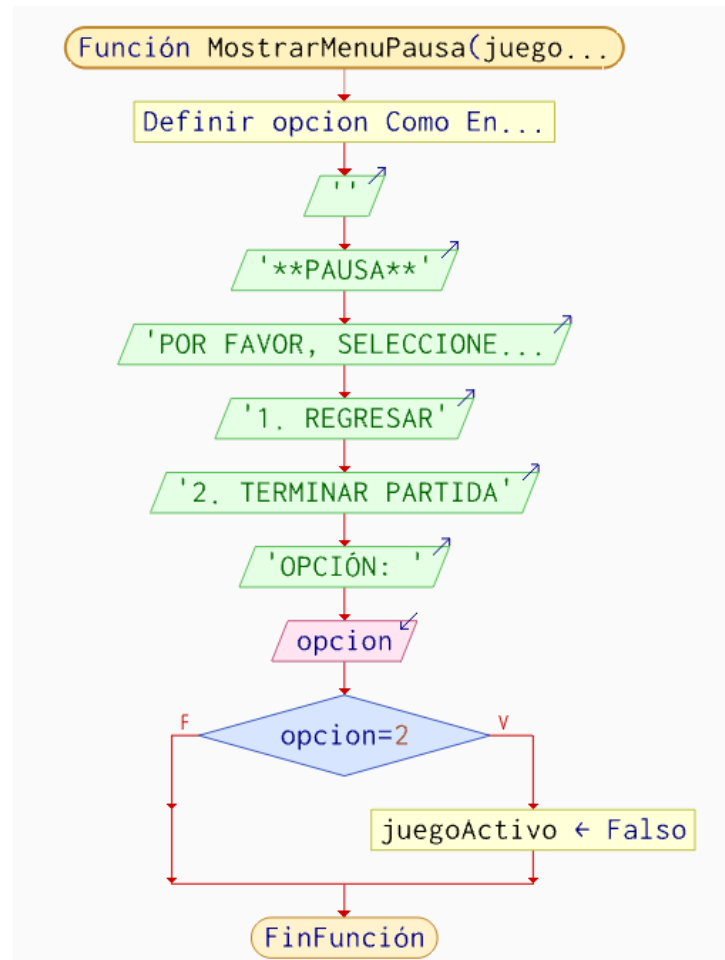
tablero[i,j] ← '<'

colocado ← Verdadero

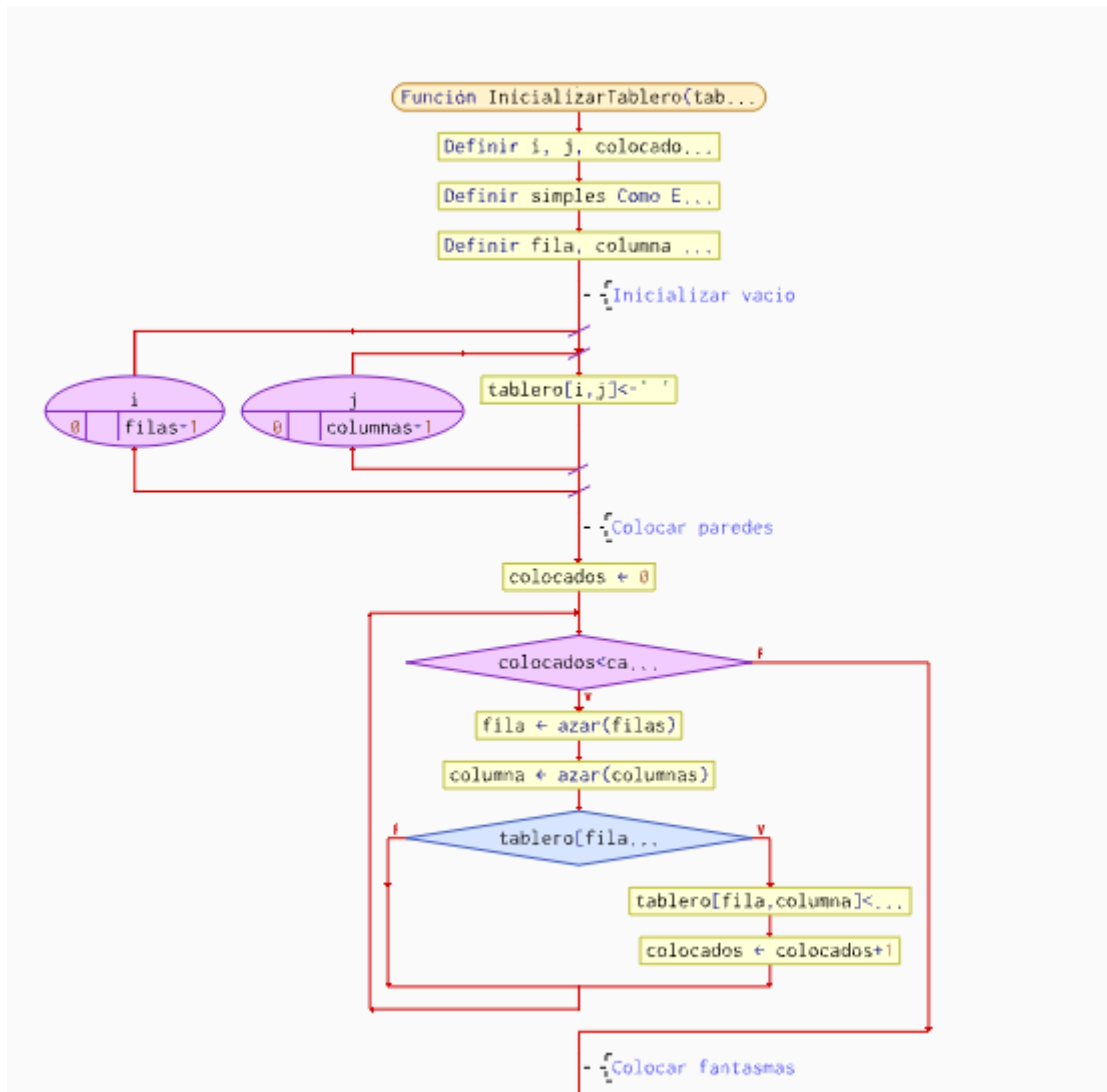
j ← j+1

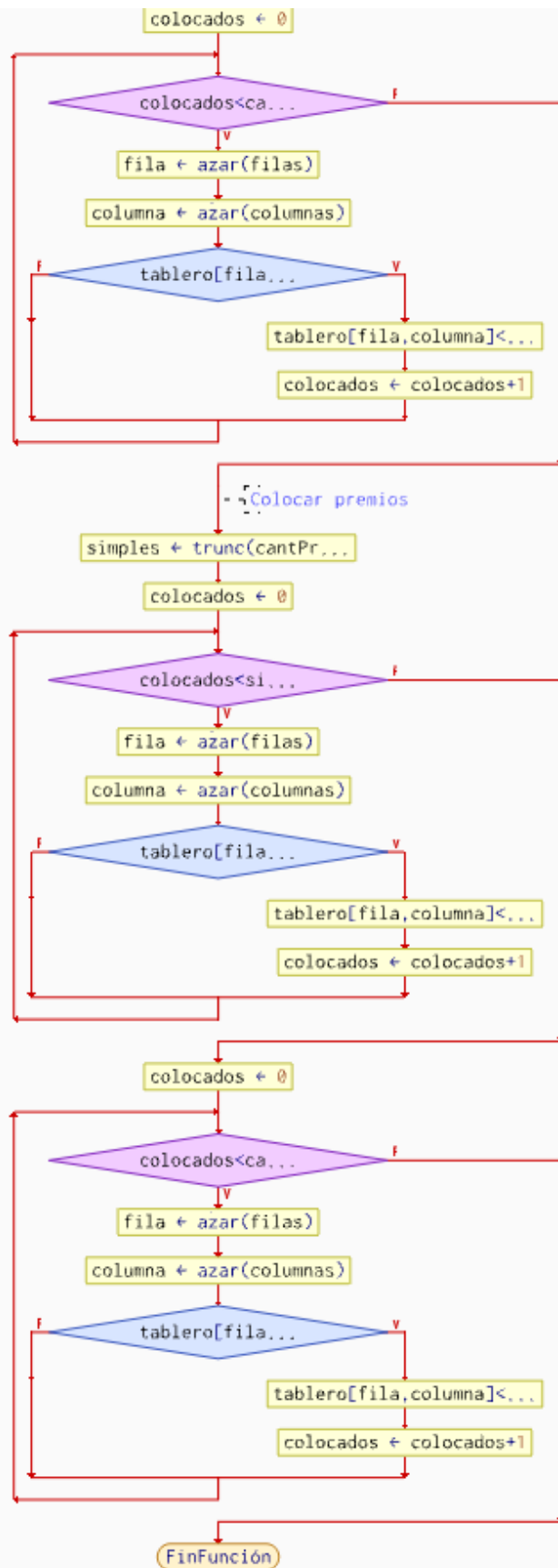
i ← i+1

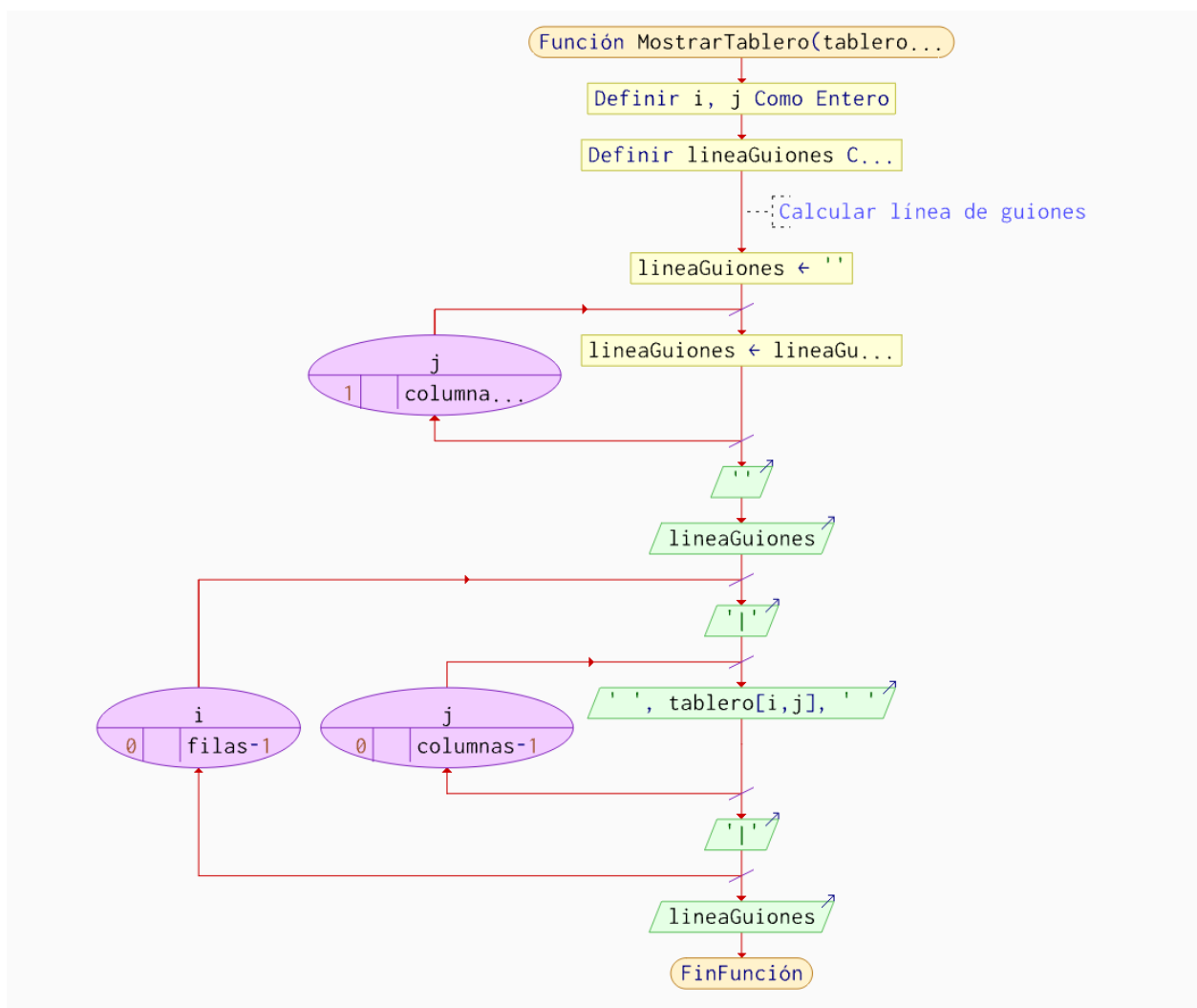
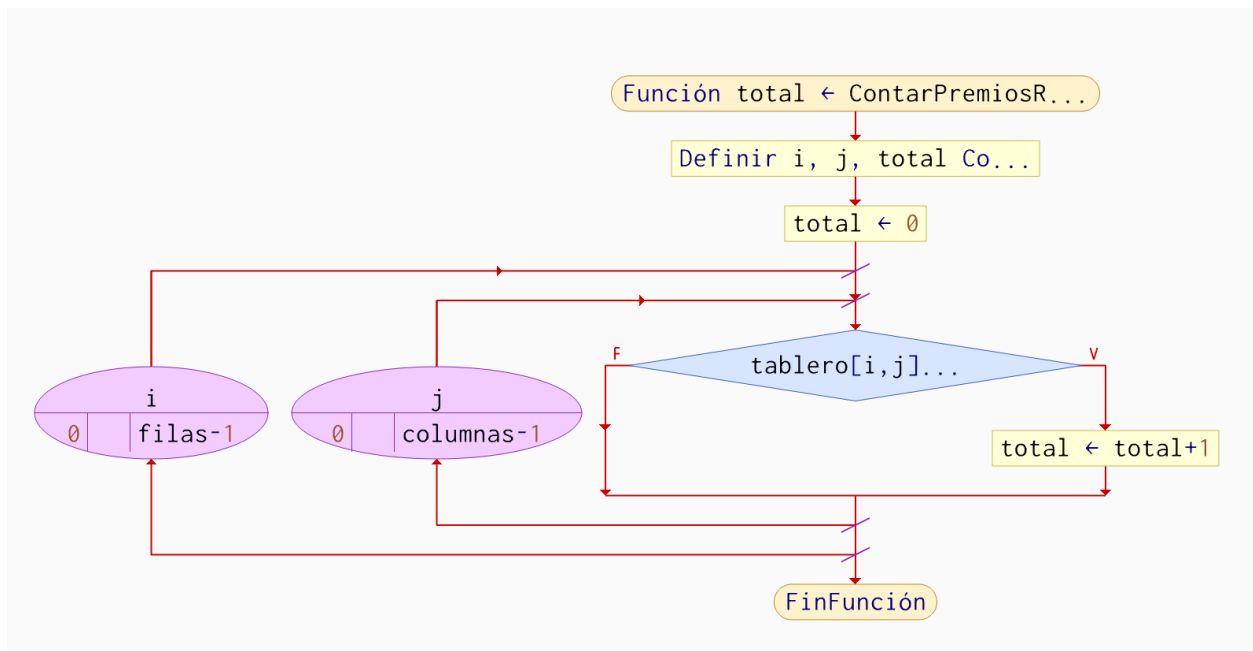
FinFunción



# Tablero.java







# Historial,java

