

Informática II

Trabajo Final

TennisBall

06/12/2023

Profesor: Ing. Marcelo Ledda
JTP: Ing. DIEDRICHS, Ana

2R11

Alumno: Leandro Flores

Contenido

Introducción	3
Diagrama funcional	4
Diagramas de flujo de Processing	4
FUNCION: setup	4
FUNCION: draw	5
FUNCION: lógica	5
FUNCION: interpretarSenal	6
Diagramas de flujo de Arduino.....	7
Código fuente de los Programas	9
Codigo fuente de Processing.....	9
Codigo fuente de Arduino	13
Salidas del Programa	15
.....	17
Anexos	18
LISTADO DE COMPONENTES	18
REFERENCIAS.....	18

Introducción

Este proyecto da vida al clásico juego de TennisBall, reinventándolo con un enfoque moderno y dinámico. La integración de Arduino y Processing a través de la conexión serial ha permitido la creación de un juego interactivo donde el usuario controla una barra para golpear una pelota en constante movimiento. La programación modular, con sus clases y funciones, ha sido fundamental para coordinar los movimientos y la lógica del juego.

Al emplear la animación fluida, he logrado una experiencia visual envolvente que complementa la interacción entre el jugador y el entorno virtual. Sin embargo, se han identificado ciertas limitaciones en la mecánica del juego, como la respuesta de la bola cerca de los bordes, aspecto que será abordado en futuras mejoras.

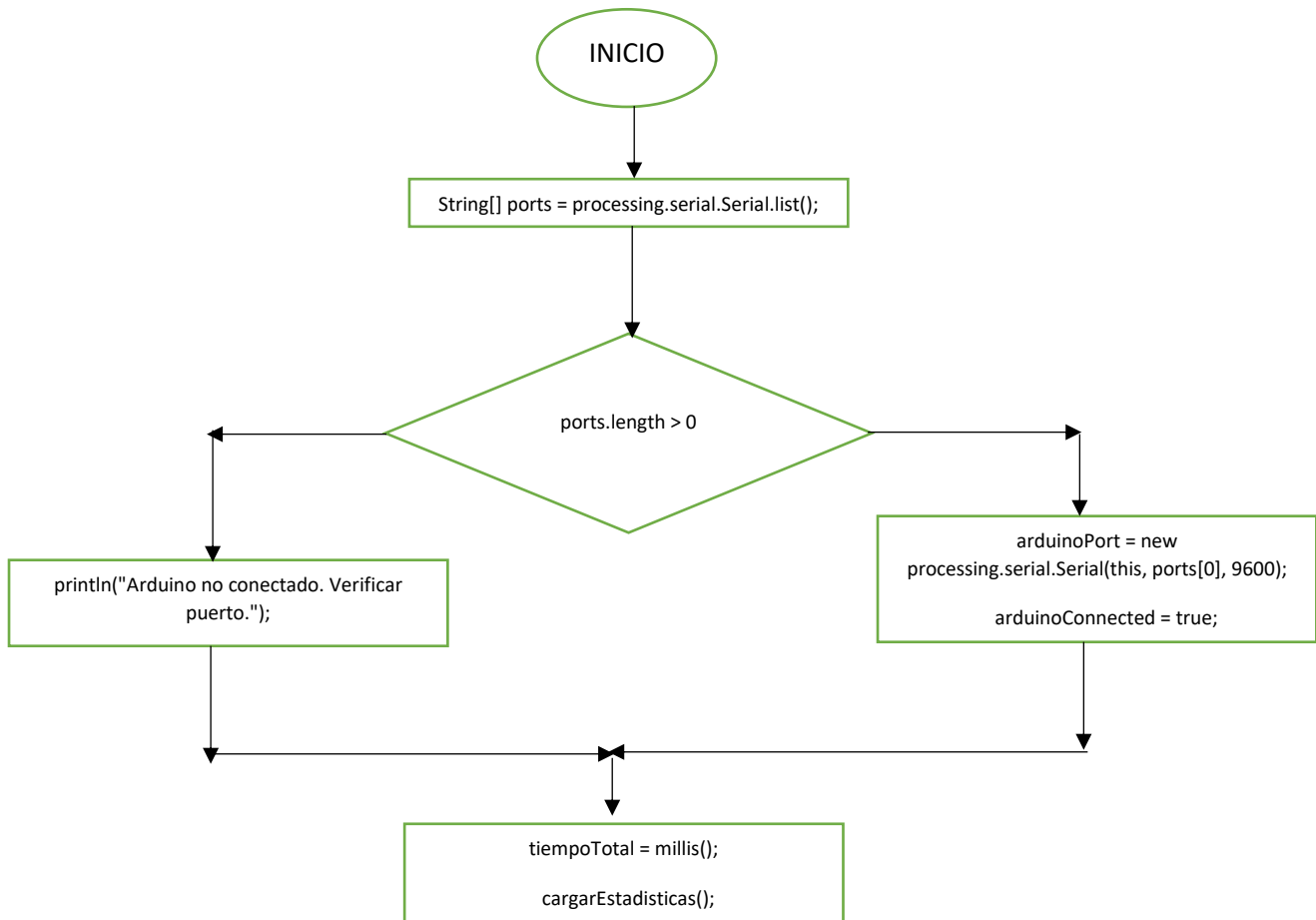
A pesar de estas limitaciones, el TennisBall se presenta como un proyecto emocionante que destaca por su combinación de hardware y software, ofreciendo una experiencia de juego única y cautivadora.

Diagrama funcional

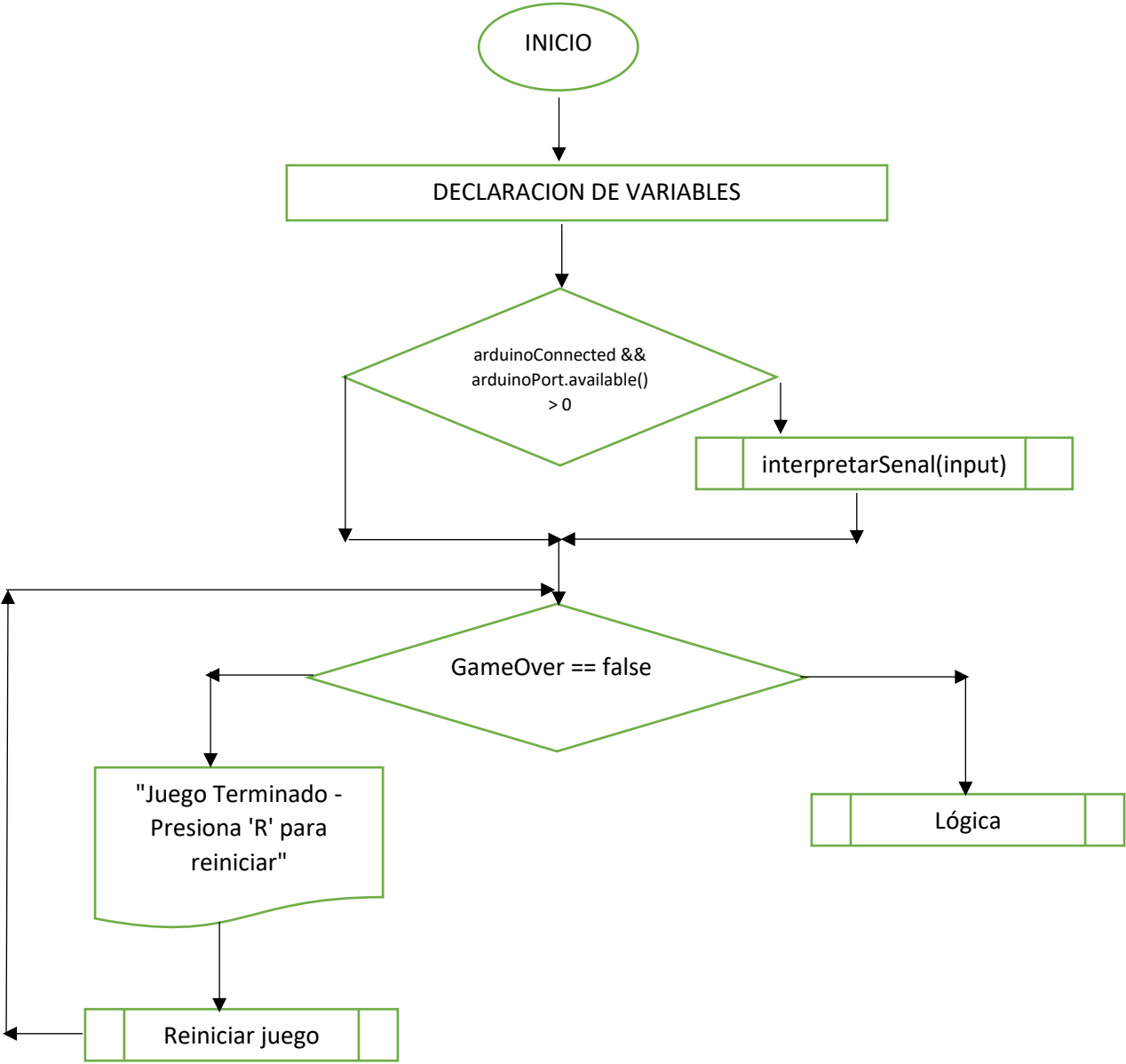
Diagramas de flujo de Processing

Conexión con Arduino

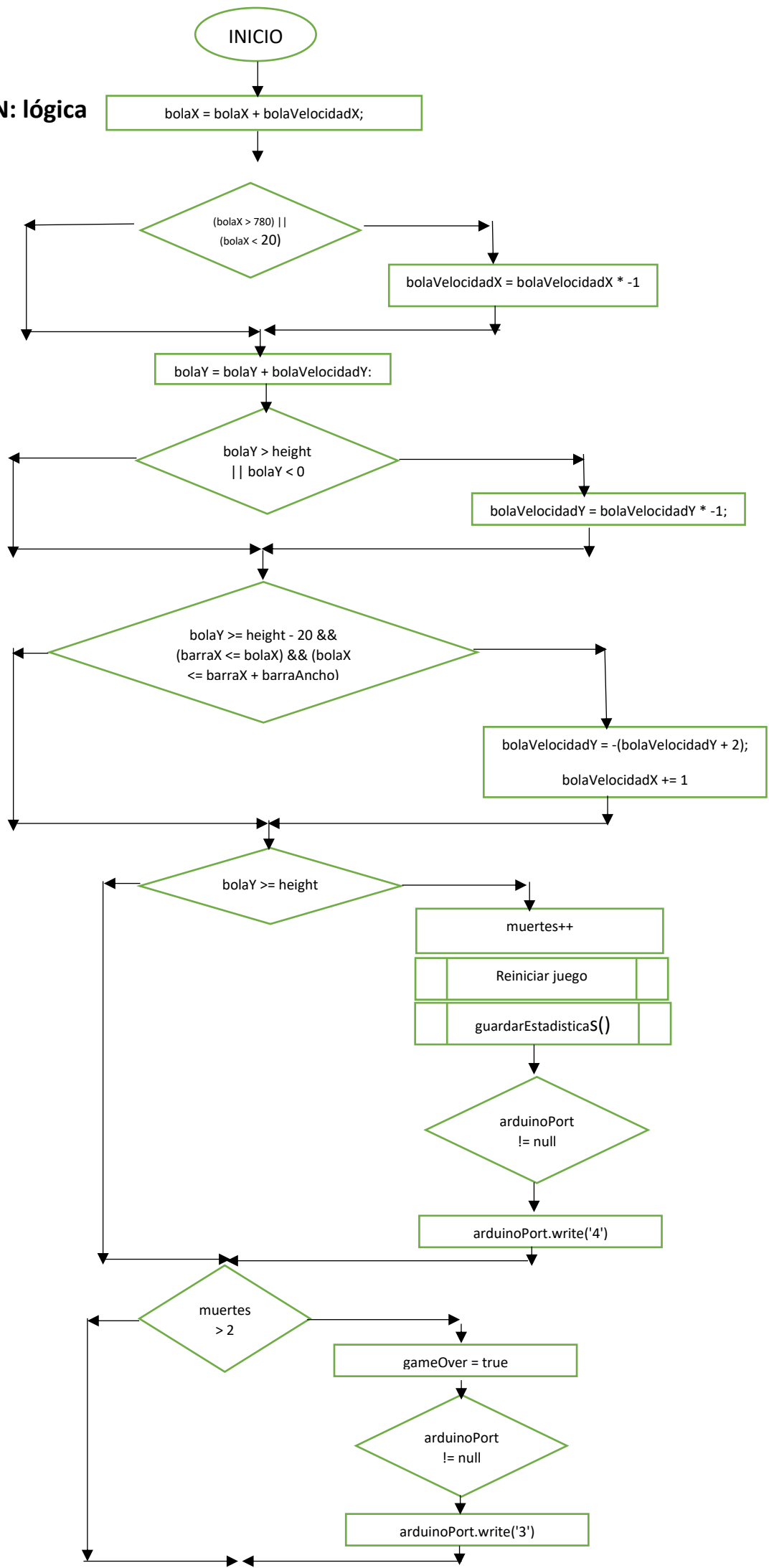
FUNCION: setup



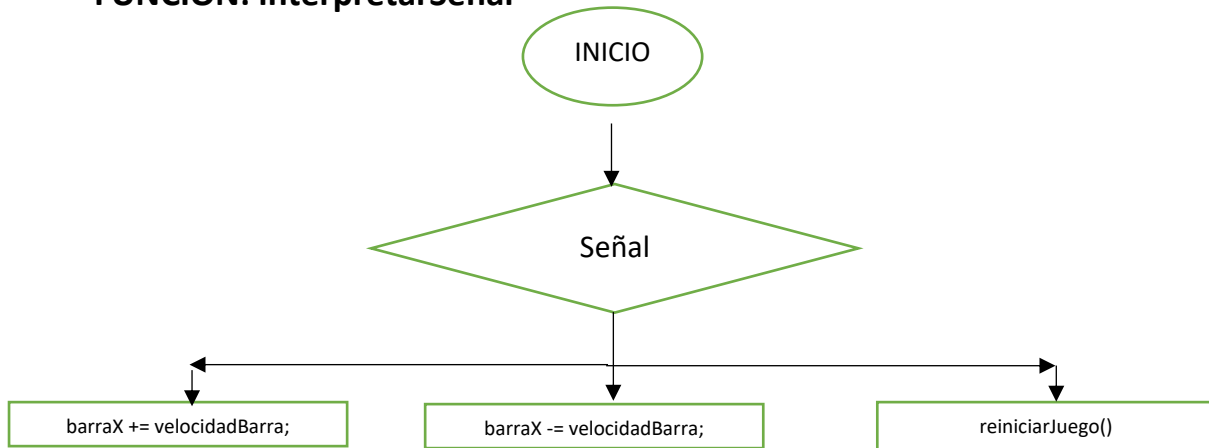
FUNCION: draw



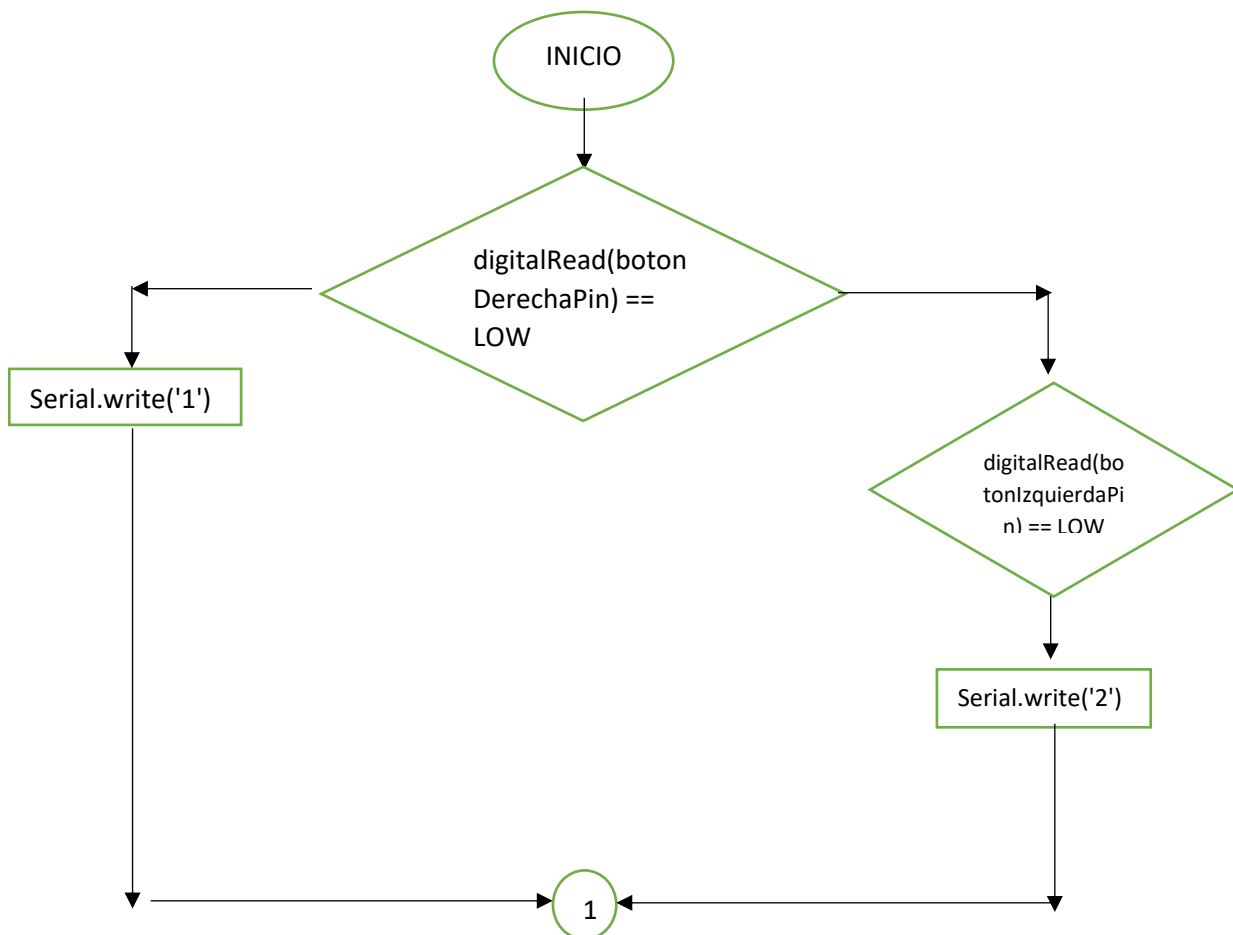
FUNCION: lógica

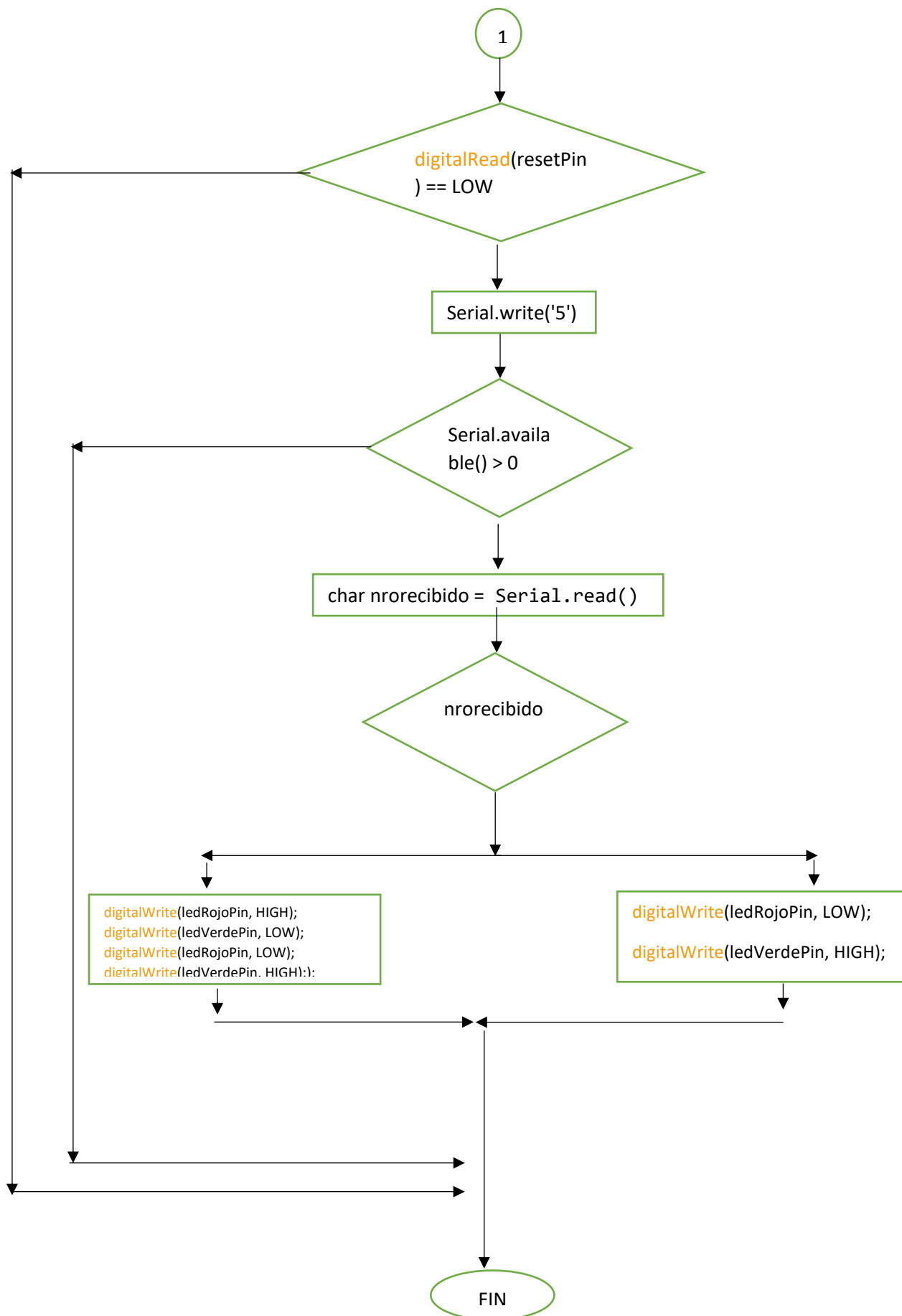


FUNCION: interpretarSenal



Diagramas de flujo de Arduino





Código fuente de los Programas

Codigo fuente de Processing

```
import processing.serial.*;

import java.io.*;

import java.util.ArrayList;


processing.serial.Serial arduinoPort;

boolean arduinoConnected = false;

int barraX;

int velocidadBarra = 20;

int barraAncho = 200;

int barraAlto = 15;

int bolaX;

int bolaY;

int bolaVelocidadX = 5;

int bolaVelocidadY = 5;

int bolaDiametro = 20;

boolean gameOver = false;

int muertes = 0;

PImage fondo;

char input;


ArrayList<Estadisticas> estadisticasList = new ArrayList<Estadisticas>();

int tiempoTotal;


void setup() {

    fondo = loadImage("C:/Users/leandro/OneDrive/Desktop/TennisBall8/sketch8/imagen1.jpg");

    String[] ports = processing.serial.Serial.list();

    if (ports.length > 0) { //hay puertos seriales disponibles?

        arduinoPort = new processing.serial.Serial(this, ports[0], 9600);

        arduinoConnected = true;

    } else {

        println("Arduino no conectado. Verificar puerto.");

    }

    size(800, 600);

    barraX = width / 2 - barraAncho / 2;
```

```

bolaX = width / 2;

bolaY = 50;


tiempoTotal = millis();
cargarEstadisticas();
}


void draw() {
    background(fondo);


    fill(0, 100, 10);

    rect(barraX, height - barraAlto, barraAncho, barraAlto);


    fill(255, 0, 0);

    ellipse(bolaX, bolaY, bolaDiametro, bolaDiametro);


    if (arduinoConnected && arduinoPort.available() > 0) {
        input = arduinoPort.readChar();
        interpretarSenal(input);
    }
    if(input == '5'){
        reiniciarJuego();
    }


    if (gameOver == false) {
        logica();
    } else {
        textSize(32);
        fill(255);

        text("Juego Terminado - Presiona 'R' para reiniciar", width / 2 - 300, height / 2);
    }
}


void interpretarSenal(char senal) {
    switch (senal) {
        case '1':
            // Lógica para mover la barra a la derecha
            barraX += velocidadBarra;

```

```

        break;

    case '2':

        // Lógica para mover la barra a la izquierda

        barraX -= velocidadBarra;

        break;

    case '5':

        // Lógica para reiniciar el juego

        reiniciarJuego();

        muertes = 0;

        gameOver = false;

        break;

    }

}

void logica() {

    textSize(30);

    fill(255);

    text("Muertes:", 10, 30);

    text(muertes, 130, 30);

    bolaX = bolaX + bolaVelocidadX;

    if ((bolaX > 780) || (bolaX < 20)) {

        bolaVelocidadX = bolaVelocidadX * -1;

    }

    bolaY = bolaY + bolaVelocidadY;

    if (bolaY > height || bolaY < 0) {

        bolaVelocidadY = bolaVelocidadY * -1; }

    if (bolaY >= height - 20 && (barraX <= bolaX) && (bolaX <= barraX + barraAncho)) {

        bolaVelocidadY = -(bolaVelocidadY + 2);

        bolaVelocidadX += 1;

    }

    if (bolaY >= height) {

        reiniciarJuego();

        muertes++;

        guardarEstadisticas();

    }

    if (arduinoPort != null) {

        arduinoPort.write('3');

    }

}

```

```

        if (muertes > 2) {

            gameOver = true;

            if (arduinoPort != null) {

                arduinoPort.write('4');

            }

        }

    }

    void reiniciarJuego() {

        bolaX = width / 2;

        bolaY = 50;

        bolaVelocidadY = 5;

        bolaVelocidadX = 5;

    }

    void keyPressed() {

        if (key == 'r' || key == 'R') {

            reiniciarJuego();

            muertes = 0;

            gameOver = false;

        } else if (key == CODED) {

            if (keyCode == LEFT) {

                barraX = barraX - velocidadBarra;

            } else if (keyCode == RIGHT) {

                barraX = barraX + velocidadBarra;

            }

        }

    }

    class Estadisticas {

        int muertes;

        int tiempoJuego;

        Estadisticas(int muertes, int tiempoJuego) {

            this.muertes = muertes;

            this.tiempoJuego = tiempoJuego;

        }

    }

    void cargarEstadisticas() {

        try {

```

```

        BufferedReader br = new BufferedReader(new
FileReader("C:/Users/leandro/OneDrive/Desktop/TennisBall8/sketch8/estadisticas.txt"));

        String line;

        while ((line = br.readLine()) != null) {

            String[] parts = line.split(": ");

            if (parts.length == 2) {

                int muertes = Integer.parseInt(parts[1].split(",")[0]);

                int tiempoJuego = Integer.parseInt(parts[1].split(": ")[1].replace(" milisegundos", ""));

                estadisticasList.add(new Estadisticas(muertes, tiempoJuego));

            } }

        br.close();

    } catch (IOException e) {

        e.printStackTrace();

    }

}

void guardarEstadisticas() {

    int tiempoJuego = millis() - tiempoTotal;

    try {

        PrintWriter writer = new PrintWriter(new
FileWriter("C:/Users/leandro/OneDrive/Desktop/TennisBall8/sketch8/estadisticas.txt", true));

        writer.println("Muertes: " + muertes + ", Tiempo total de juego: " + tiempoJuego + " milisegundos");

        writer.close();

    } catch (IOException e) {

        e.printStackTrace();

    }

}

```

Codigo fuente de Arduino

```

const int botonDerechaPin = 2;

const int botonIzquierdaPin = 3;

const int resetPin = 5;

const int ledRojoPin = 8;

const int ledVerdePin = 9;


void setup() {

    Serial.begin(9600);

    pinMode(botonDerechaPin, INPUT_PULLUP);

    pinMode(botonIzquierdaPin, INPUT_PULLUP);

    pinMode(resetPin, INPUT_PULLUP);

    pinMode(ledRojoPin, OUTPUT);

```

```

pinMode(ledVerdePin, OUTPUT);}

void loop() {

  if (digitalRead(botonDerechaPin) == LOW) {

    Serial.write('1'); // Enviar señal para mover la barra a la derecha

  } else if (digitalRead(botonIzquierdaPin) == LOW) {

    Serial.write('2'); // Enviar señal para mover la barra a la izquierda

  }


  if (digitalRead(resetPin) == LOW) {

    Serial.write('5');

  }


  delay(50); // Evitar rebotes


  digitalWrite(ledVerdePin, HIGH);


  // Leer desde Serial
  if (Serial.available() > 0) {

    char nrorecibido = Serial.read();


    // Lógica para interpretar la señal desde Processing
    switch (nrorecibido) {

      case '3': //hubo una muerte

        digitalWrite(ledRojoPin, HIGH);

        digitalWrite(ledVerdePin, LOW);

        delay(1000);

        digitalWrite(ledRojoPin, LOW);

        digitalWrite(ledVerdePin, HIGH);

        break;

      case '4':

        // El juego ha terminado

        digitalWrite(ledRojoPin, HIGH);

        digitalWrite(ledVerdePin, LOW);

        break;

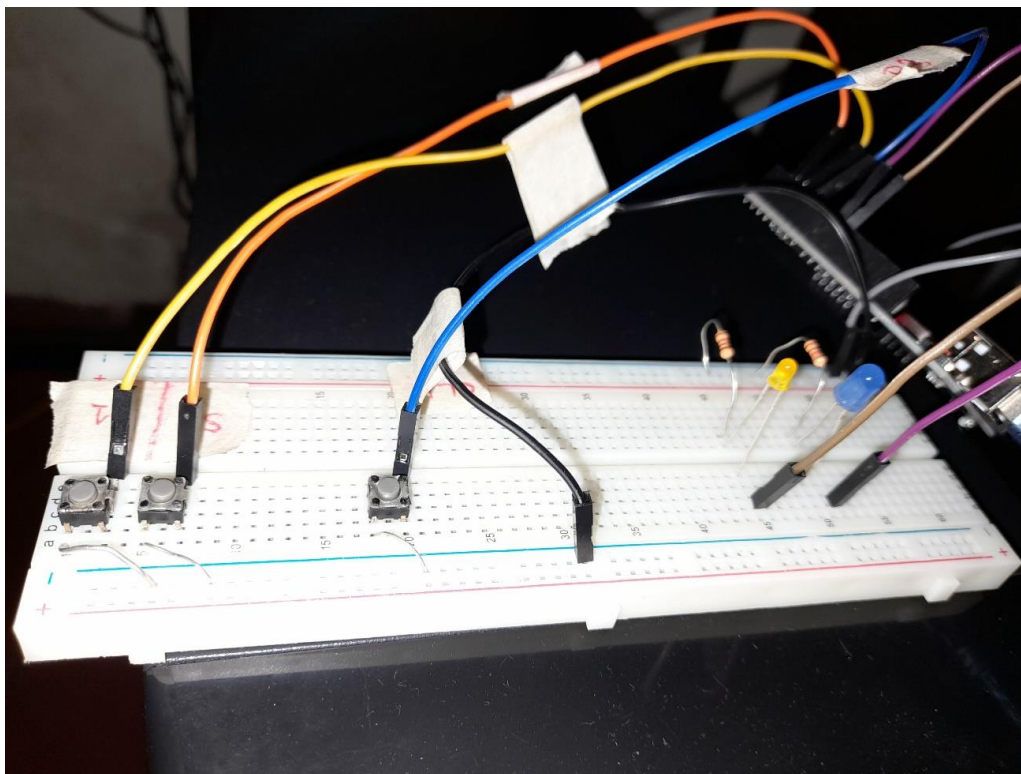
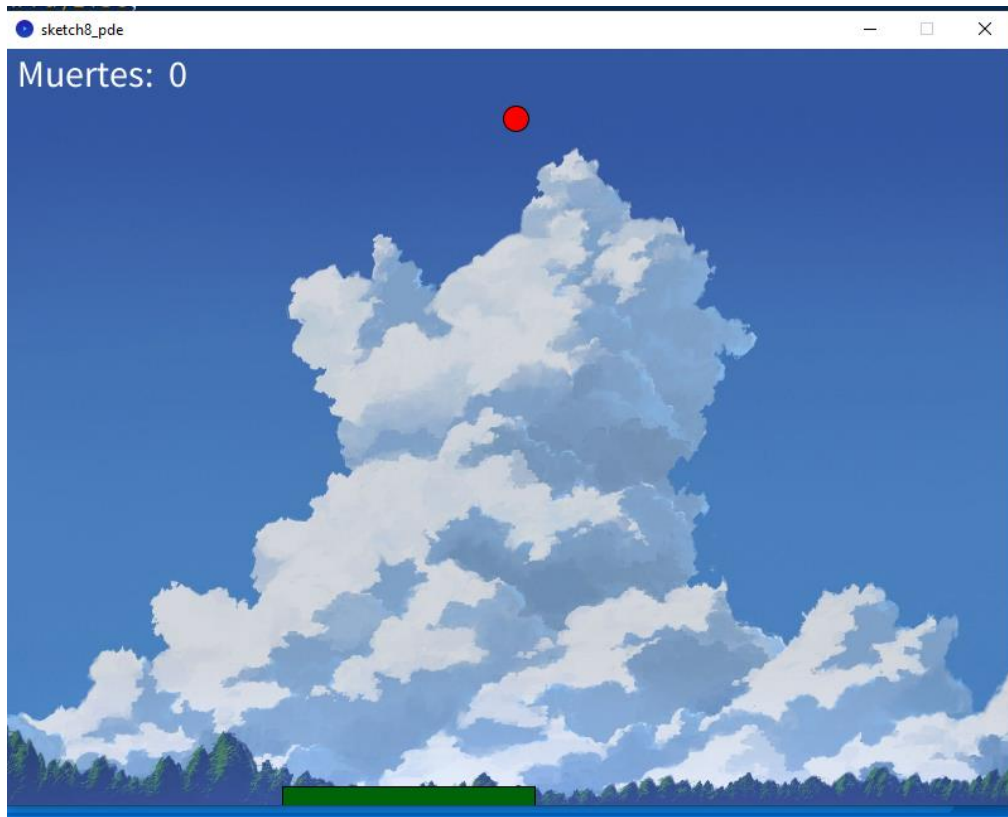
    }

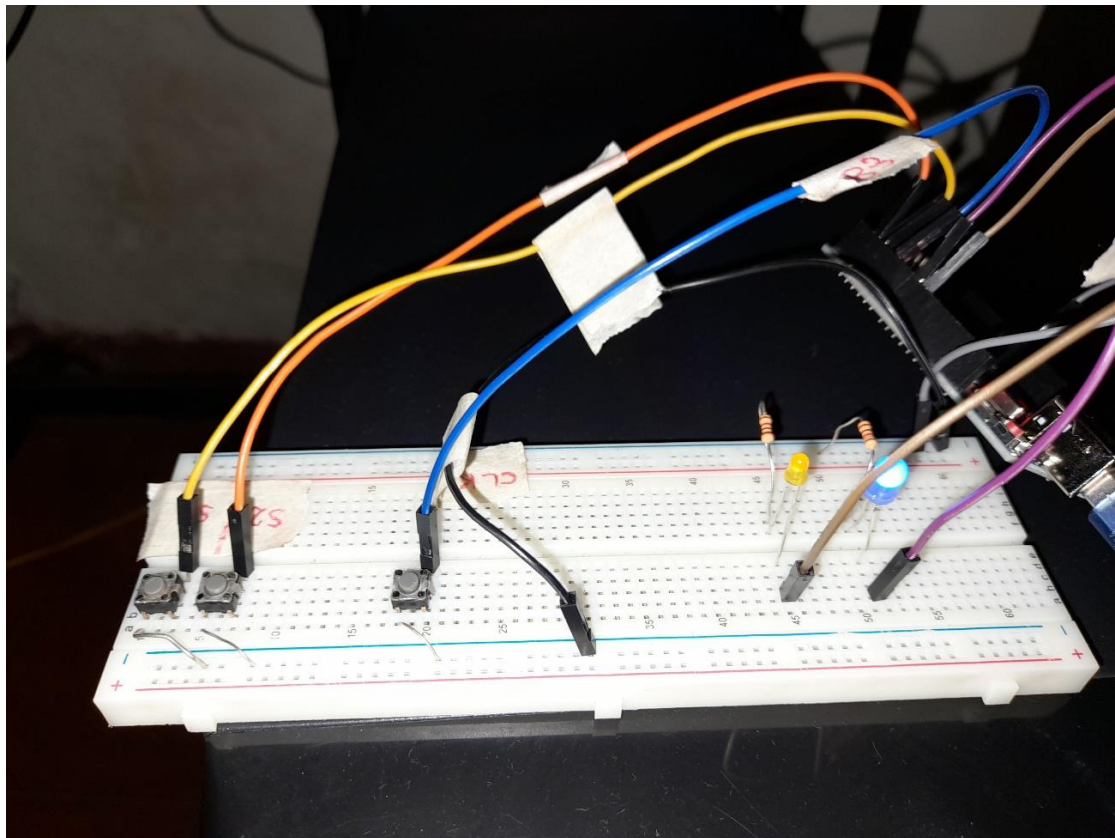
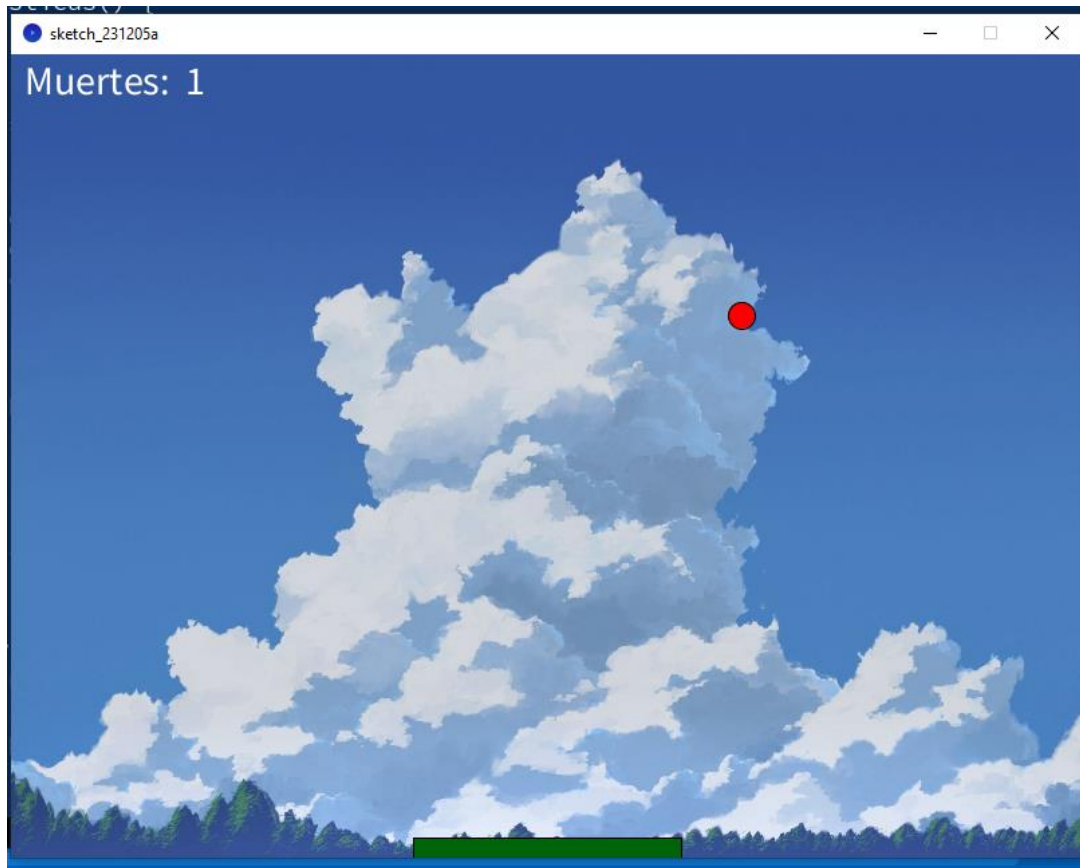
  }

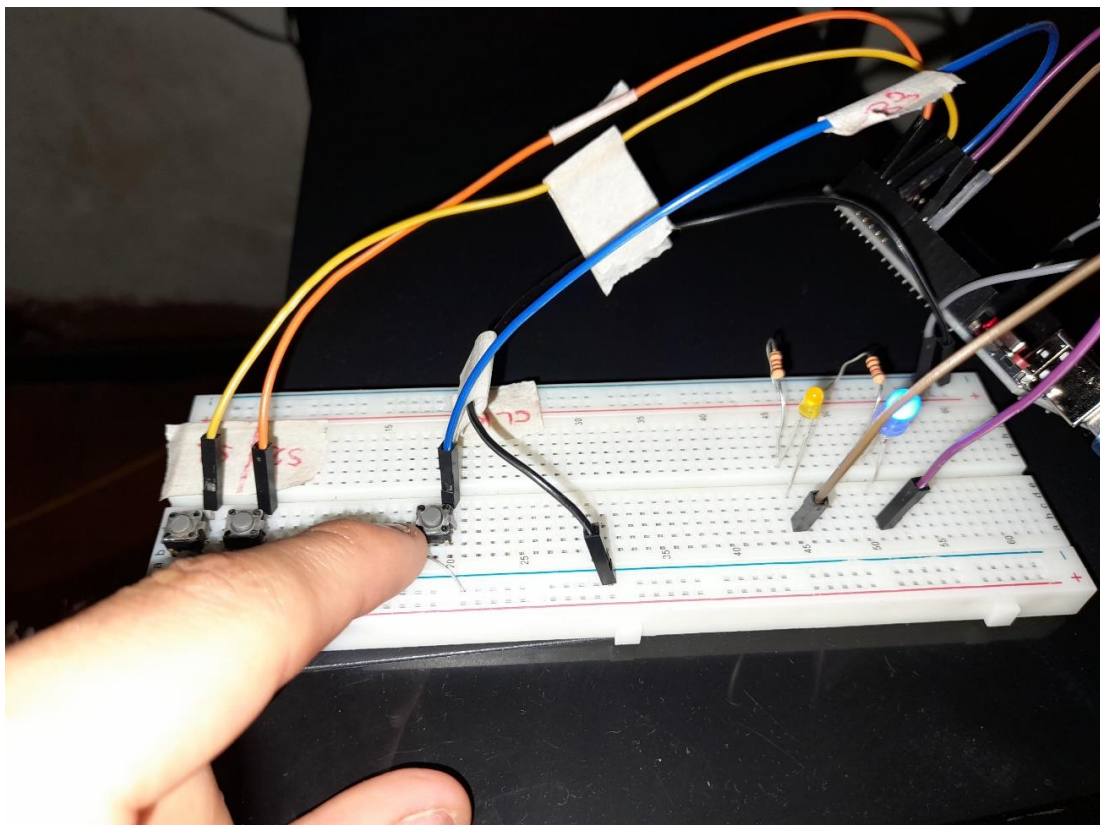
}

```

Salidas del Programa







Anexos

LISTADO DE COMPONENTES

Arduino Uno

Pulsadores (3 o más)

Resistencias (para los pulsadores)

Cables de conexión

Computadora con Processing instalado

REFERENCIAS

<https://es.wikipedia.org/wiki/Arkanoid>

https://programacion.net/codigo/arkanoid_en_c_73#google_vignette

https://www.lawebdelprogramador.com/codigo/C-Visual-C/6804-Juego-de-tipo-Breakout-Arkanoid.html#google_vignette