

# MATERIA DE SISTEMAS EMBEBIDOS-LABORATORIO 3

## JUEGO ATARI Y CAJA ARDUINO FUSION 360

Edwin A. Solano Arciniegas

18 de enero de 2021

### 1. Introducción

Se realiza un juego de Atari en el software procesing el cual debe tener conexión virtual con Arduino mediante proteus, este se lo hace creando un puerto com virtual con el software VSPE para poder establecer una comunicación con este. El juego realizado es tenis el cual una pelota se mueve chocando con las paredes y una barra en movimiento horizontal puede hacerla regresar.

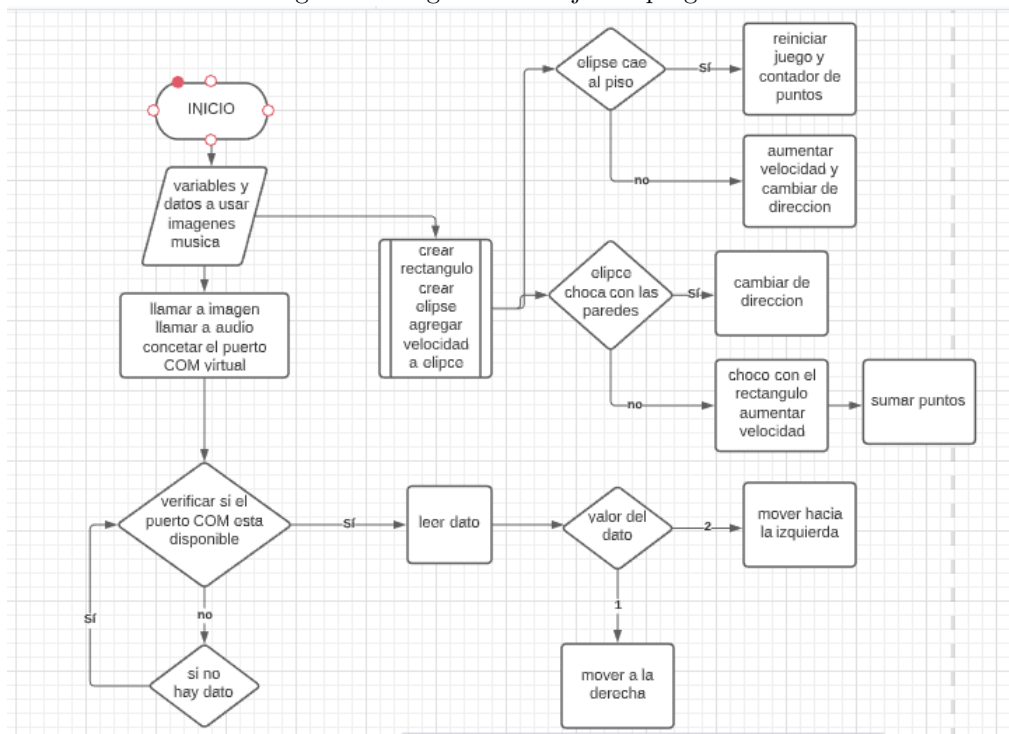
Para este diseño se emplea imágenes y música además que si pierde y la bola cae al piso el juego se reinicia y tendrá que comenzar nuevamente y el contador de puntos se reinicia, además para que el juego sea mas interesante cada vez que la barra toque la pelota esta aumentara su velocidad hasta llegar un momento que no sea posible atraparla.

### 2. Diseño del Sistema

#### 2.1. Diagrama de Flujo

El diagrama de flujo explica como esta desarrollado el juego en porcesing y cuales son sus interacciones con arduino:

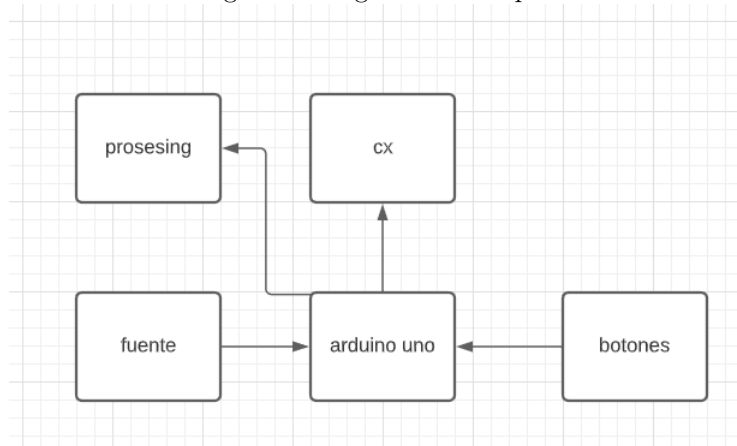
Figura 1: Diagrama de flujo del programa



## 2.2. Diagrama de Bloques

En el diagrama de bloques se identifica que componentes están conectados al Arduino, en este caso se muestra los controles del juego:

Figura 2: Diagrama de bloques



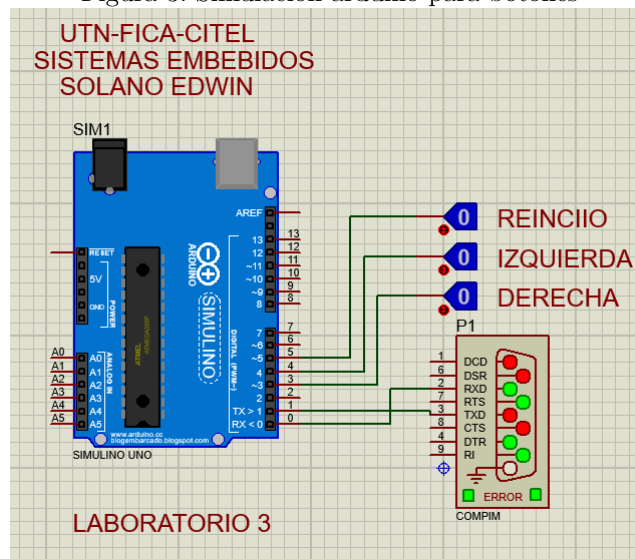
## 3. Desarrollo

### 3.1. Simulación

#### 3.1.1. Simulación Proteus

Se realiza una simulación en proteus para poder manipular el juego en arduino en este caso se lo hace todo de manera virtualizada así se conecta el software processing con arduino montado en proteus.

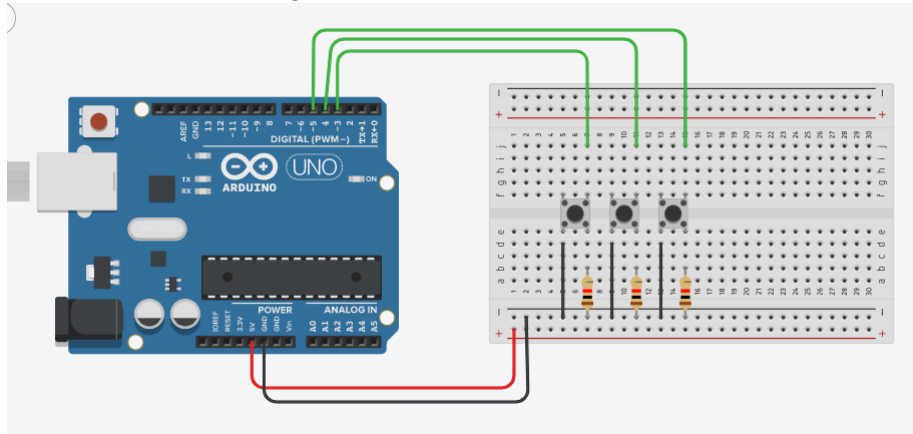
Figura 3: Simulación arduino para botones



#### 3.1.2. Simulación Tinkercad

Se realiza la simulación el programa en línea Tinkercad del arduino y los controles de este para así tener una idea exacta de como debe estar montado el circuito.

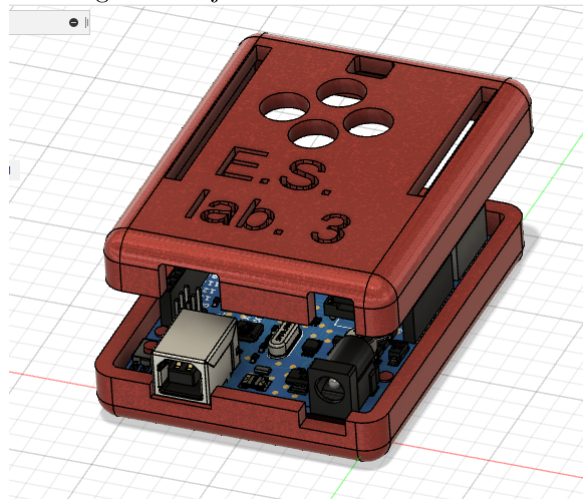
Figura 4: Simulacion en Tinkercad



### 3.1.3. Diseño de caja Fusion 360

Se realiza una simulacion en el porgрма Fusion 360 el cual contiene la caja para el juego atari y el arduino.

Figura 5: caja de arduino en fusion 360



## 4. Análisis de Resultados

### 4.1. programa completo

[https://github.com/edwinsolano/SolanoEdwin\\_Laboratorios/tree/main/SolanoEdwin\\_laboratorio-3](https://github.com/edwinsolano/SolanoEdwin_Laboratorios/tree/main/SolanoEdwin_laboratorio-3)

### 4.2. codigo arduino

pinos a usar y enviar por comunicacion serial los valores:

Figura 6: concexion con proseeing y proteus

```
SolanoEdwin_Juego $
1 /*
2  * UTM-FICA-CITEL
3  * sistemas embebidos
4  * objetivo: concetar pines y cx para enviar datos a procesing
5  * Nombre: Solano Edwin
6  */
7 const int boton1=3;//inicio de pin
8 const int boton2=4;//inicio de pin
9 const int boton3=5;//inicio de pin
10 void setup() {
11   Serial.begin(9600);//comunicacion con CX
12   pinMode(boton1,INPUT);//pin 3 como entrada de datos
13   pinMode(boton2,INPUT);//pin 3 como entrada de datos
14   pinMode(boton3,INPUT);//pin 3 como entrada de datos
15 }
16 void loop() {
17   if(digitalRead(boton1)==HIGH){//cuadno se lea el boton 1 en alto
18     Serial.println("1");//mandar mensaje
19     delay(40);//temporizador
20   }
21   if(digitalRead(boton2)==HIGH){//cuadno se lea el boton 2 en alto
22     Serial.println("2");//mandar mensaje
23     delay(40);//temporizador
24   }
25   if(digitalRead(boton3)==HIGH){//cuadno se lea el boton 3 en alto
26     Serial.println("3");//mandar mensaje
27     delay(40);//temporizador
28   }
29   if(digitalRead(boton1)==LOW && digitalRead(boton2)==LOW && digitalRead(boton3)==LOW){//c
30     Serial.println("5");
31     delay(40);//temporizador
32   }
33 }
```

### 4.3. codigo procesing

codigo el cual se ejecuta una vez inicializado el juego:

Figura 7: codigo procesing juego atari

```

Ejecutar
SolanoEdwin_Juego ▼
1 /*
2  *UTN-FICA-CITEL7
3  *SISTEMAS EMBEBIDOS
4  *OBJETIVO: diseñar un juego de atari
5  *Nombre: Solano Edwin
6  */
7 import ddf.minim.*;
8 import processing.serial.*;//obtener los datos por counicacion serial
9 Serial puerto;//nombre del puerto
10 PImage img;//variable para ingresar imagen
11 int x=100;//posicion inicial en x
12 int y=100; //posicion inicial en y
13 int velX=3;//velocidad inicial de la pelota en x
14 int velY=3;//velocidad inicial de la pelota en y
15 int posicionX;//posicon del la barra
16 String dato="";//cadena de caracteres se inicializa con una variable vacia
17 float valor; //para obtener los datos
18 int Xini; //posicion inicial
19 int cont=0;//contador de puntos
20 boolean gameOver=false;//variable para reiniciar juego
21 Minim finJuego;
22 AudioPlayer play1;
23
24 void setup(){
25   size(600,450);//tamaño de la pantalla
26   finJuego=new Minim(this);
27
28   img=loadImage("12.jpg");//cargamos la imagen
29   image(img,0,0);//posicion de la imagen
30   puerto=new Serial(this,"COM10",9600);//comunicacion con el arduino virtual
31 }
```

Figura 8: codigo procesing juago atari

```

SolanoEdwin_Juego
34 void draw(){
35   if(puerto.available()>0){//verificar si el puerto esta disponible
36     dato=puerto.readStringUntil('\n');
37     if(dato==null){//verificar si el dato no contiene nada
38       return;//volver a verificar el dato
39     }
40     valor=float(dato);//guardar el dato en una variable flotante
41   }
42   if(valor==1){//moverse a la izquierda
43     posicionX-=10;//mover la barra a la izquierda
44   }
45   else if(valor==2){//moverse a la derecha
46     posicionX+=10;//mover la barra a la derecha
47   }
48   println(valor);//ver dato en la consola
49   background(99,99,99);//color de fondo
50   Xini=200+posicionX;//inicializacion de la barra
51   image(img,0,0);//cargar imagen en la posicion
52   fill(#1FDFFF);//color del rectangulo
53   rect(Xini,420,180,30);//creacion e posicion
54   fill(#14FF00);//color de la bola
55   ellipse(x,y,40,40);//creacion y posicion
56   x+=velX;//velocidad de la bola en x
57   if(x>580||x<20){//cuadno choca con las paredes
58     velX=velX*-1;//aumenta la velocidad
59   }
60   y+=velY;//velocidad de la bola en y
61   if(y>height||y<0){//cuando choca
62     velY=velY*-1;//aumento de velocidad
63   }
64 }

```

Figura 9: codigo procesing juago atari

```

SolanoEdwin_Juego
62 velY=velY*-1;//aumento de velocidad
63 }
64 }
65 if(y==410 && (x<Xini+180) && (x>Xini)){//cuando la barra topa
66   velY=(velY+1);//aumentar velocidad
67   velX+=1;//aumentar velocidad
68   cont++;//sumar la puntuacion
69 }
70 textSize(10);//tamaño de texto
71 text("PUNTUACION: ",30,30);//mensaje de la puntuacion
72 text(cont,120,30);//contador de la puntuacion
73 }
74 if(y==450){//cuadno no se halla la bola
75   fill(#FFFAFF);
76   textSize(40);
77   play1=finJuego.loadFile("sonidoFin.wav");
78   text("GAME OVER",200,250);
79   gameOver=true;
80 }
81 if (gameOver==true){//reinicio de juego
82   restart();//llamar al metodo
83 }
84 }
85 }
86 void restart(){//reiniciar juego
87   gameOver=false;//reinicio de variables
88   x=100;
89   y=100;
90   velX=3;
91   velY=3;
92   cont=0;

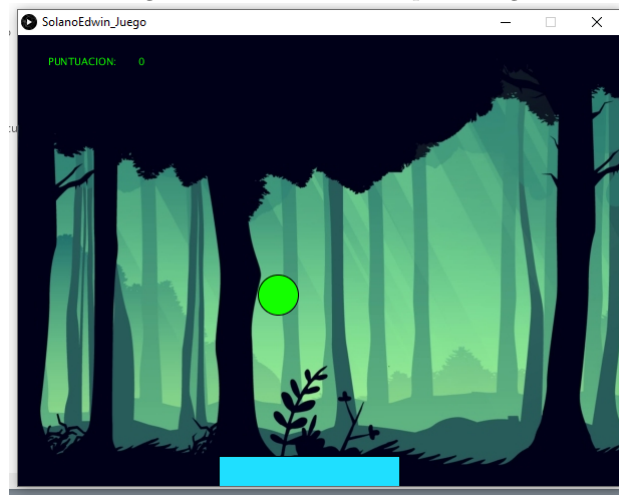
```

## 4.4. simulacion

Ejecución del programa en proteus:

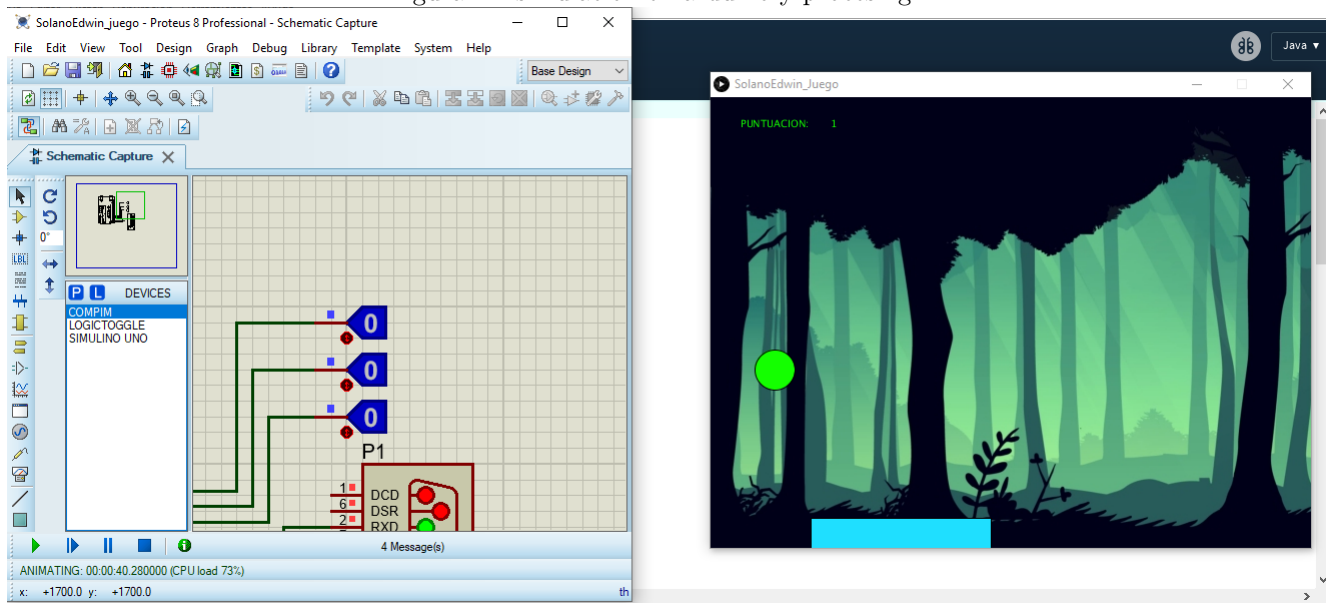
- se conecta una maquina virtual para mandar mensajes seriales ademas de que los botones conectados mandan datos a procesing para que este pueda moverse he interactuar con el juego.

Figura 10: simulación en procesing



arduino y procesing

Figura 11: simulación en arduino y procesing



## 5. Conclusiones y Recomendaciones

### 5.1. Conclusiones

Con el uso del programa fusión 360 se diseña una caja el cual contiene el Arduino y de igual manera sus respectivas salidas para poder conectar los diferentes comandos para el juego y así poder interactuar con este.

En el programa procesing se incluye todas las librerías a usar tanto para el uso de imágenes, audio y la conexión serial virtualizada de manera que este se conecta con el programa proteus donde está cargado el programa de Arduino y hace la función de sus diferentes controles.

Con el programa VSPE se hace la conexión virtualizada del puerto COM con el Arduino para tener interacción entre estos al no contar con un Arduino físico es la manera mas sencilla de conectarlo.

## 5.2. Recomendaciones

Si se hace una modificación en el código Arduino y la simulación en proteus esta corriendo es necesario parar la simulación y volverla a correr para que se guarden los cambios caso contrario este funcionara con la configuración anterior.

El uso de librerías es primordial en estos programas ya que es posible conectar de manera virtualizada todo sin necesidad de tener nada físico, además el uso correcto de variables.

El uso de procesing para diseño de juegos es muy sencilla ya que este cuenta con una interfaz muy amigable muy parecida a la de Arduino y con el lenguaje C++ es mucho más sencillo codificar un código que cumpla con los requerimientos deseados.