

Proyecto Final: Tanque de Agua con Arduino



Profesor/es: Nano, Mónica y Salamero Martín

Integrantes: Intili Santiago, Iriarte Juan Cruz, Mihoceovich Álvaro,
Purro Santiago y Saieva Matías

Materia: Laboratorio I

División: B

Año: 2023

Introducción

Durante el primer semestre de la materia Laboratorio - I de la carrera Ingeniería en Informática, hemos estado aprendiendo el uso, funcionamiento y programación de las placas Arduino. En esta primera etapa, utilizamos la plataforma Tinkercad para dicho aprendizaje, la cual permite realizar prototipos de manera online.

Uno de los ejercicios realizados en clase fue la simulación de un tanque de agua, el cual se alimentaba mediante el bombeo de agua desde un pozo contiguo. La bomba de agua debía encenderse cuando el nivel del tanque sea menor al 20% y detenerse cuando se alcanzara el 90% de su capacidad.

Se debía verter una cierta cantidad de agua en dos oportunidades por día en unos bebederos, para ello se disponía de pulsadores que accionan de manera remota la bomba de agua que realiza dicha acción. Además, se debía conocer en todo momento el porcentaje de llenado y la temperatura ambiente en la zona del tanque.

Una vez resuelta la consigna planteada de manera virtual, decidimos demostrar que se podía dar una solución real a este problema ficticio, y nos propusimos llevarla a cabo.

Habiendo planteado la situación, reunido los materiales y resuelto todas las problemáticas propias de un trabajo de ingeniería, mostramos los resultados alcanzados en el informe técnico desarrollado a continuación.

Definición del propósito y funcionalidades del prototipo

Propósito del Prototipo Robótico:

El propósito fundamental de este prototipo robótico es brindar una solución automatizada y eficiente para el control y la gestión del nivel de agua en un tanque. A través de la integración de tecnología robótica y electrónica, nuestro objetivo es facilitar la operación y supervisión remota de dos bombas de agua. Estas bombas tienen la capacidad de realizar dos funciones clave: verter agua en el tanque para su llenado o extraer agua del tanque para su vaciado, todo ello manejado de manera intuitiva desde una aplicación móvil.

Funcionalidades del Prototipo:

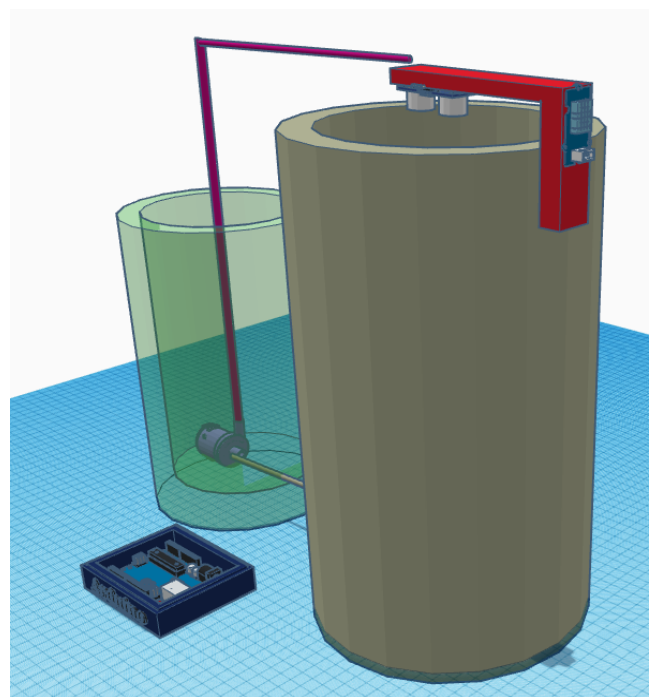
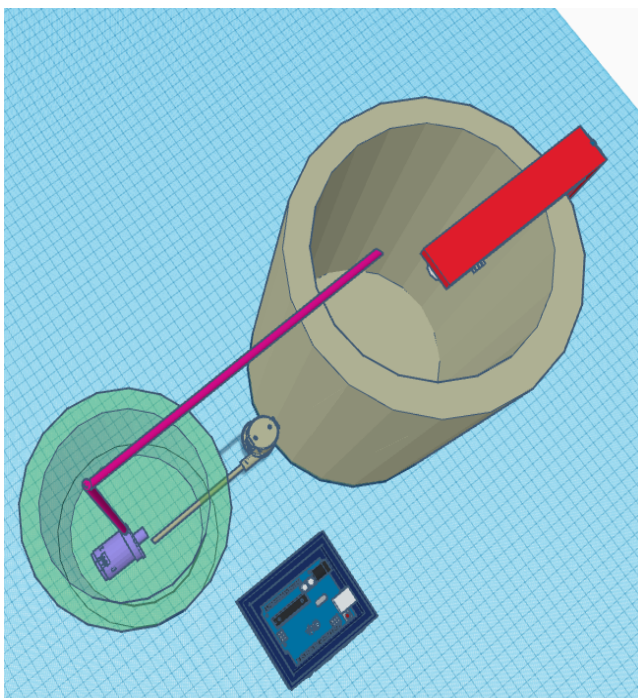
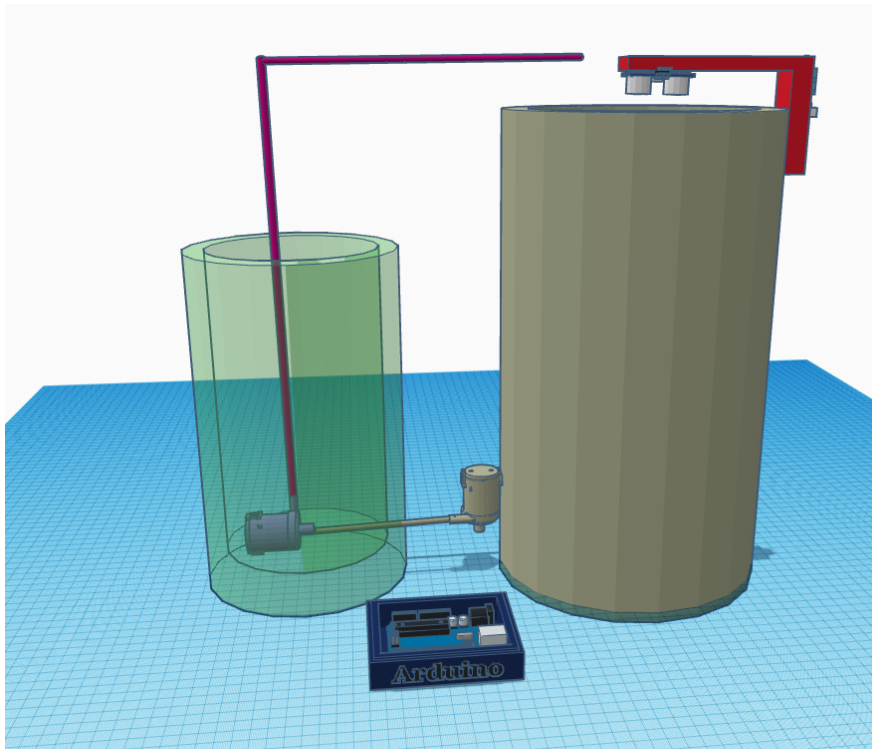
- Control Remoto: El prototipo permitirá controlar las bombas de agua de manera precisa y en tiempo real a través de una aplicación móvil. Esto asegura que los usuarios tengan un control total sobre los procesos de llenado y vaciado.
- Monitoreo de temperatura: utilizando un sensor de temperatura (DHT 11) se podrá conocer la temperatura ambiente en todo momento.
- Monitoreo de cantidad de agua: utilizando un sensor de ultrasonido se podrá calcular la cantidad de agua del tanque, y por lo tanto su porcentaje de llenado.
- Vaciado automático: En caso de superar cierto nivel de llenado, las bombas se activan de manera automática, evitando de esta manera rebosamientos.

Selección de componentes:

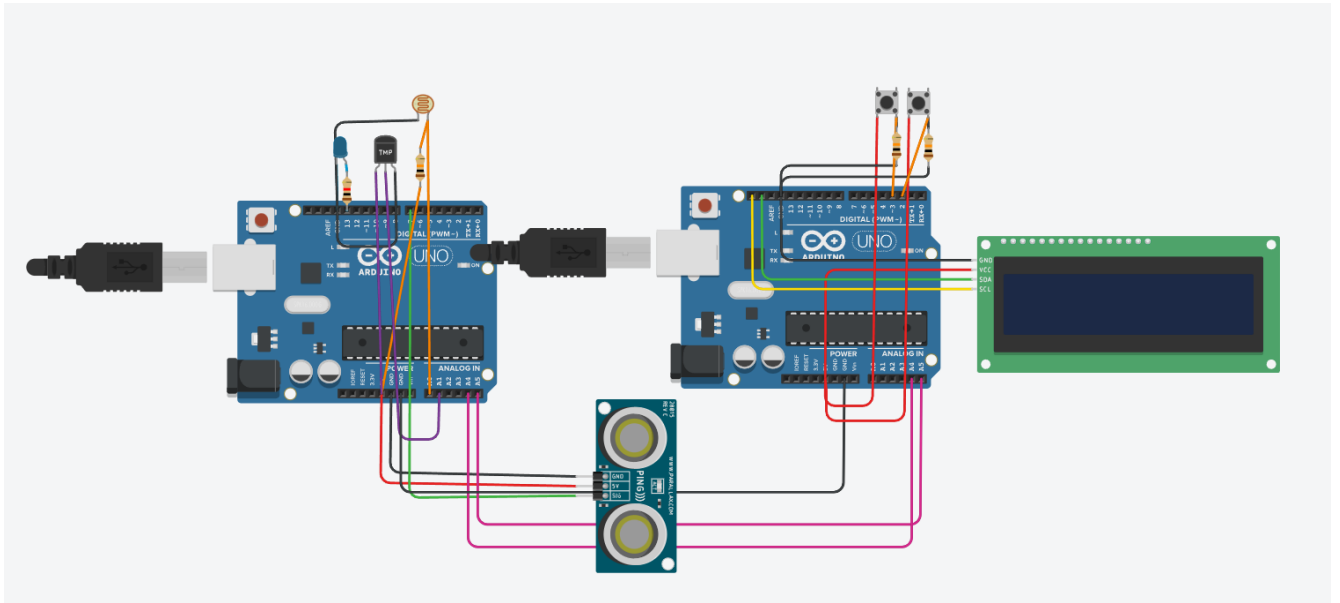
Los componentes necesarios para llevar a cabo nuestro proyecto son:

- Placa Arduino UNO
- Sensor Ultrasónico
- Sensor de temperatura
- 2 bombas de agua
- Un recipiente a modo de tanque
- Tubos manguera de silicona
- Protoboard
- Cables

Diseño en 3D:

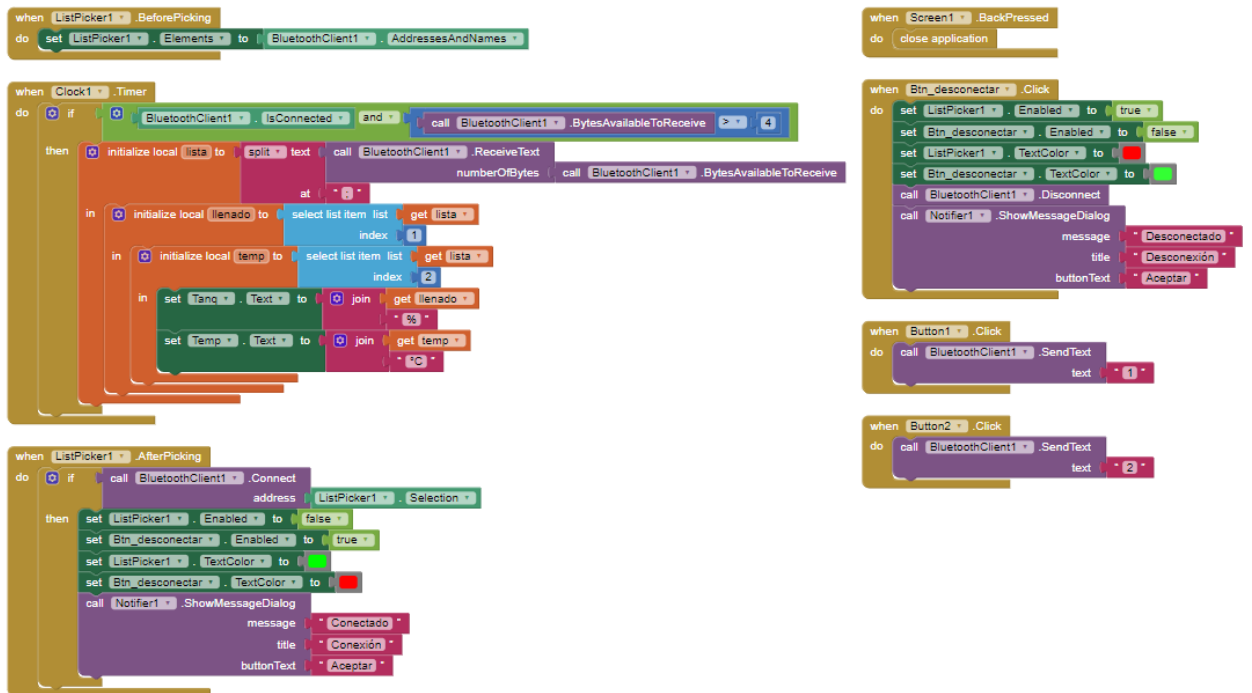


Esquema de conexiones:

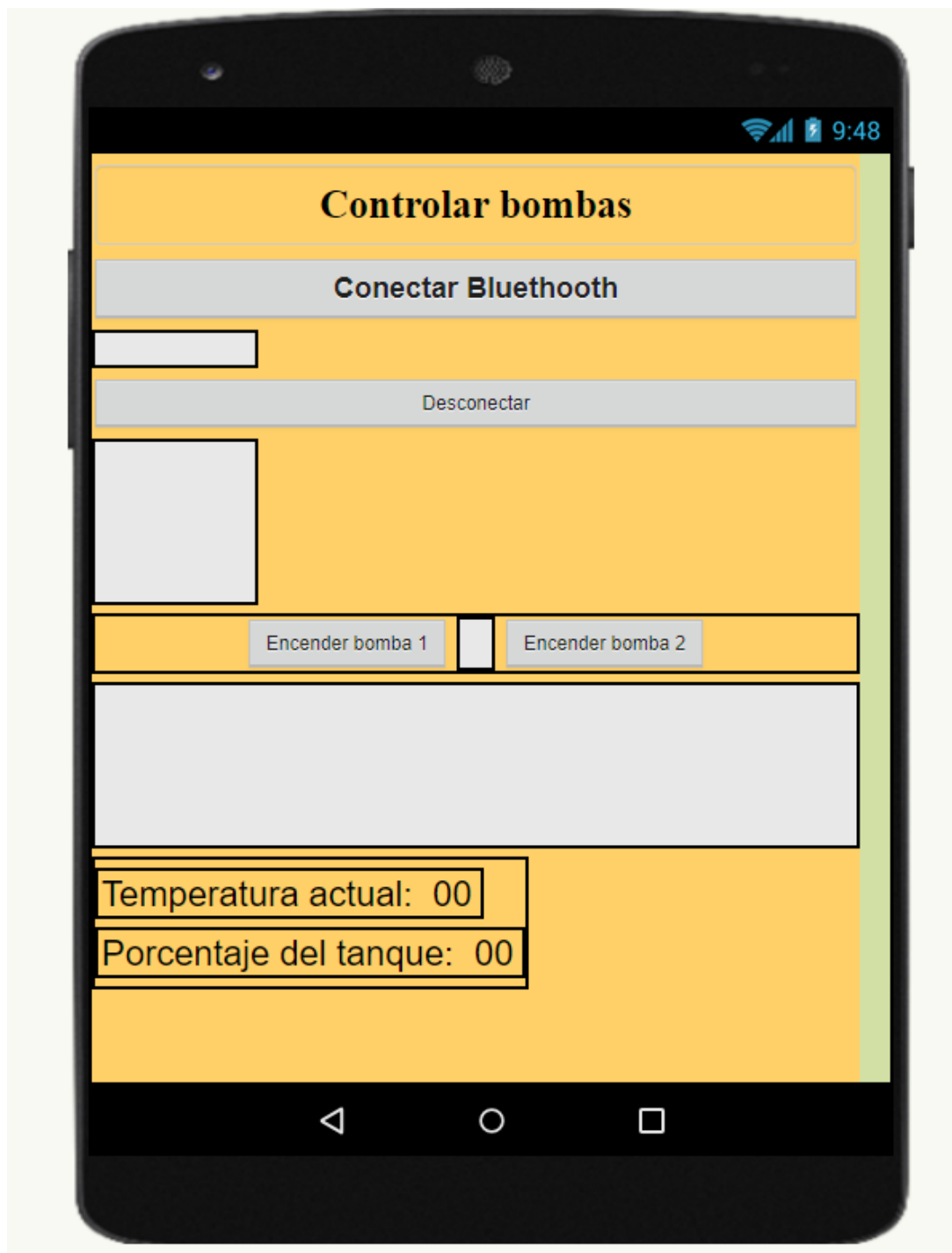


Aplicación en APPInventor:

Bloques:



Vista de la Aplicación de APPInventor:



Código Arduino:

```
1  #include <DHT.h>
2  #include <SoftwareSerial.h>
3
4  #define DHTPIN 7
5  #define DHTTYPE DHT11
6  #define trigPin 8
7  #define echoPin 9
8  #define relePin 13
9  #define relePin2 6
10
11  DHT dht(DHTPIN, DHTTYPE);
12  SoftwareSerial BTSerial(11, 10);
13
14  void setup() {
15      Serial.begin(9600);
16      BTSerial.begin(115200);
17      dht.begin();
18      pinMode (trigPin, OUTPUT);
19      pinMode (echoPin, INPUT);
20      pinMode(relePin, OUTPUT);
21      pinMode(relePin2, OUTPUT);
22      digitalWrite(relePin, LOW);
23  }
24
25  void loop() {
26      float humedad = dht.readHumidity();
27      float temperatura = dht.readTemperature();
28      long duracion;
29      float distancia;
30      const float distancia_maxima = 23.4;
31      const float distancia_minima = 2;
32      float porcentaje;
33      int inByte = 0;
```

```

34
35     digitalWrite(trigPin, HIGH);
36     delay(10);
37     digitalWrite(trigPin, LOW);
38     duracion = pulseIn(echoPin, HIGH);
39     distancia = (duracion / 58.2);
40     porcentaje = 100 - ((distancia - distancia_minima) / (distancia_maxima - distancia_minima)) * 100;
41
42     if (BTSerial.available()) {
43         while (BTSerial.available()) {
44             char receivedChar = BTSerial.read();
45             if (receivedChar == '1') {
46                 digitalWrite(relePin, HIGH);
47                 delay(5000);
48             } else {
49                 digitalWrite(relePin, LOW);
50             }
51             if (receivedChar == '2') {
52                 digitalWrite(relePin2, HIGH);
53                 delay(5000);
54             } else {
55                 digitalWrite(relePin2, LOW);
56             }
57         }
58     }
59

```

```

60     if (porcentaje <= 20) {
61         digitalWrite(relePin, HIGH);
62         delay(3000);
63         digitalWrite(relePin2, LOW);
64     } else if (porcentaje >= 35) {
65         digitalWrite(relePin, LOW);
66         digitalWrite(relePin2, HIGH);
67         delay(5000);
68     }
69
70     BTSerial.print(porcentaje);
71     BTSerial.print(";");
72     BTSerial.print(temperatura);
73     BTSerial.print(";");
74
75     Serial.print("Temperatura: ");
76     Serial.print(temperatura);
77     Serial.print("°C, Humedad: ");
78     Serial.print(humedad);
79     Serial.print("%, Distancia: ");
80     Serial.print(distancia);
81     Serial.println(" cm");
82
83     delay(1500);
84 }
85

```