

C A R R O B O M B E R O

Autores : Gianfranco Flores , Gustavo Huilla , Leopoldo Contreras

Universidad Nacional de Moquegua

Escuela Profesional de Ingenieria de Sistemas e Informatica

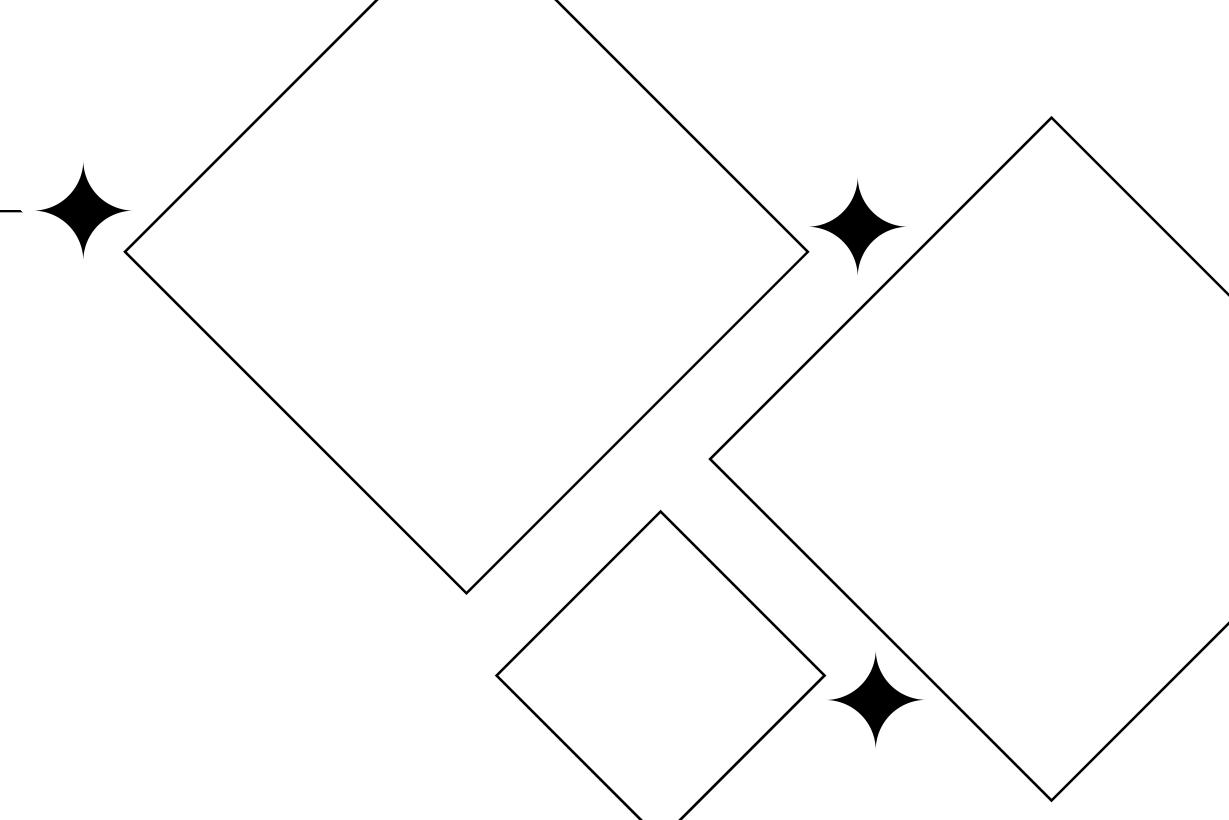
Robótica 2

Docente : Mgr. Yessica Rosas Cuevas

30 de Junio de 2025

Índice de

C O N T E N I D O S

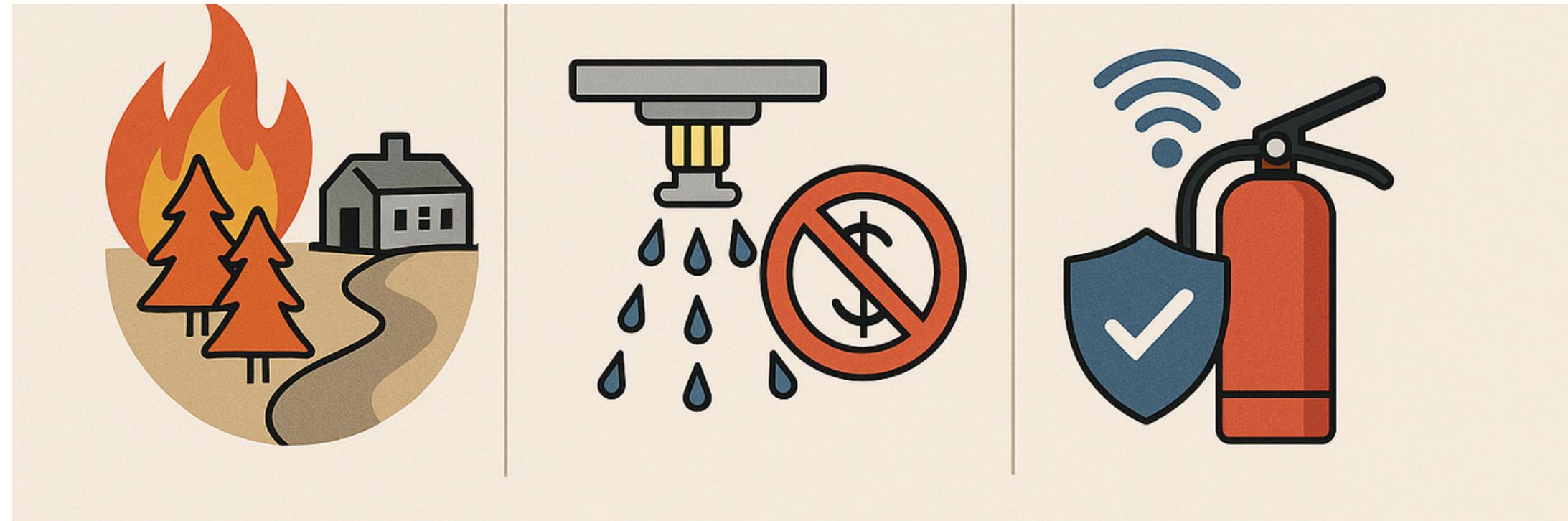
- 
- 01.** Planteamiento del Problema
 - 02.** Solución
 - 03.** Arquitectura del Sistema
 - 04.** Diseño de hardware y software
 - 05.** Pruebas y Resultados
 - 06.** Demostración
 - 07.** Conclusiones

PLANTEAMIENTO DEL PROBLEMA

En lugares sin personas o de difícil acceso, los incendios pueden tardar en ser atendidos.

Los sistemas automáticos como los rociadores son costosos e inaccesibles para muchos.

Se necesitan soluciones tecnológicas accesibles, seguras y automáticas para actuar rápido.



J U S T I F I C A C I O N

Este prototipo busca dar una solución tecnológica y educativa a un problema real: la falta de respuesta rápida ante incendios. Demuestra que es posible crear un sistema autónomo y remoto usando componentes electrónicos accesibles. Estudios previos han validado el uso de sensores, microcontroladores y módulos inalámbricos, confirmando la viabilidad del diseño.

O B J E T I V O S

O B J E T I V O G E N E R A L

Desarrollar un prototipo funcional de robot bombero con teleoperación híbrida, capaz de detectar y extinguir focos de incendio mediante sensores de flama y un sistema de aspersión de agua automatizado.

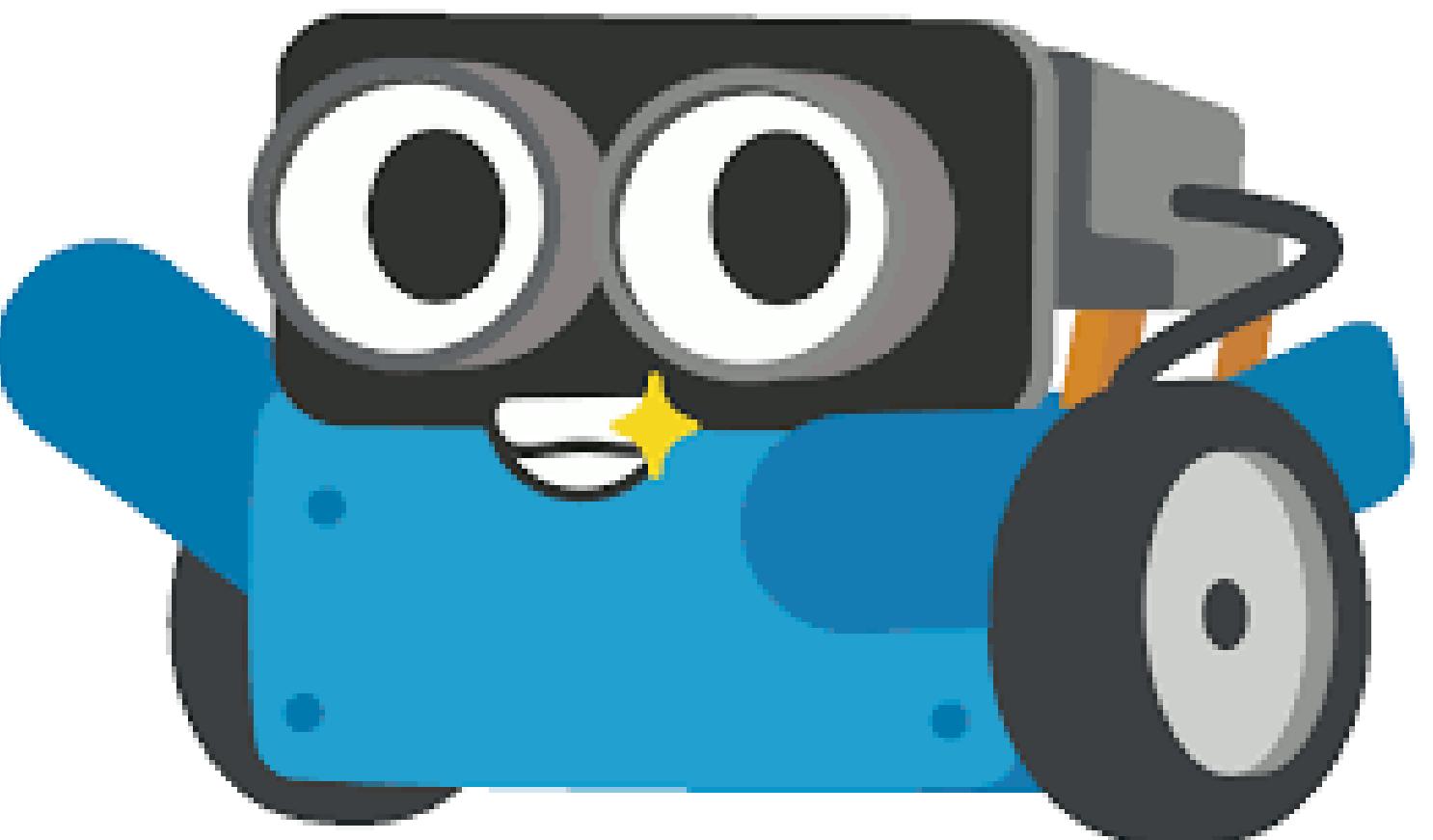
O B J E T I V O S E S P E C I F I C O S

1. Diseñar la estructura móvil del robot con configuración diferencial.
2. Integrar los sensores de flama al sistema de control mediante Arduino UNO.
3. Implementar el sistema de aspersión con bomba, relé y servomotor SG90.
4. Desarrollar la aplicación Android para la teleoperación por Bluetooth.

SOLUCION

Se desarrolló un prototipo funcional de robot bombero móvil capaz de detectar fuego y actuar automáticamente en incendios de baja escala.

Integra sensores, actuadores y control remoto en un diseño accesible y educativo, basado en una configuración de robot diferencial controlado por Arduino UNO.



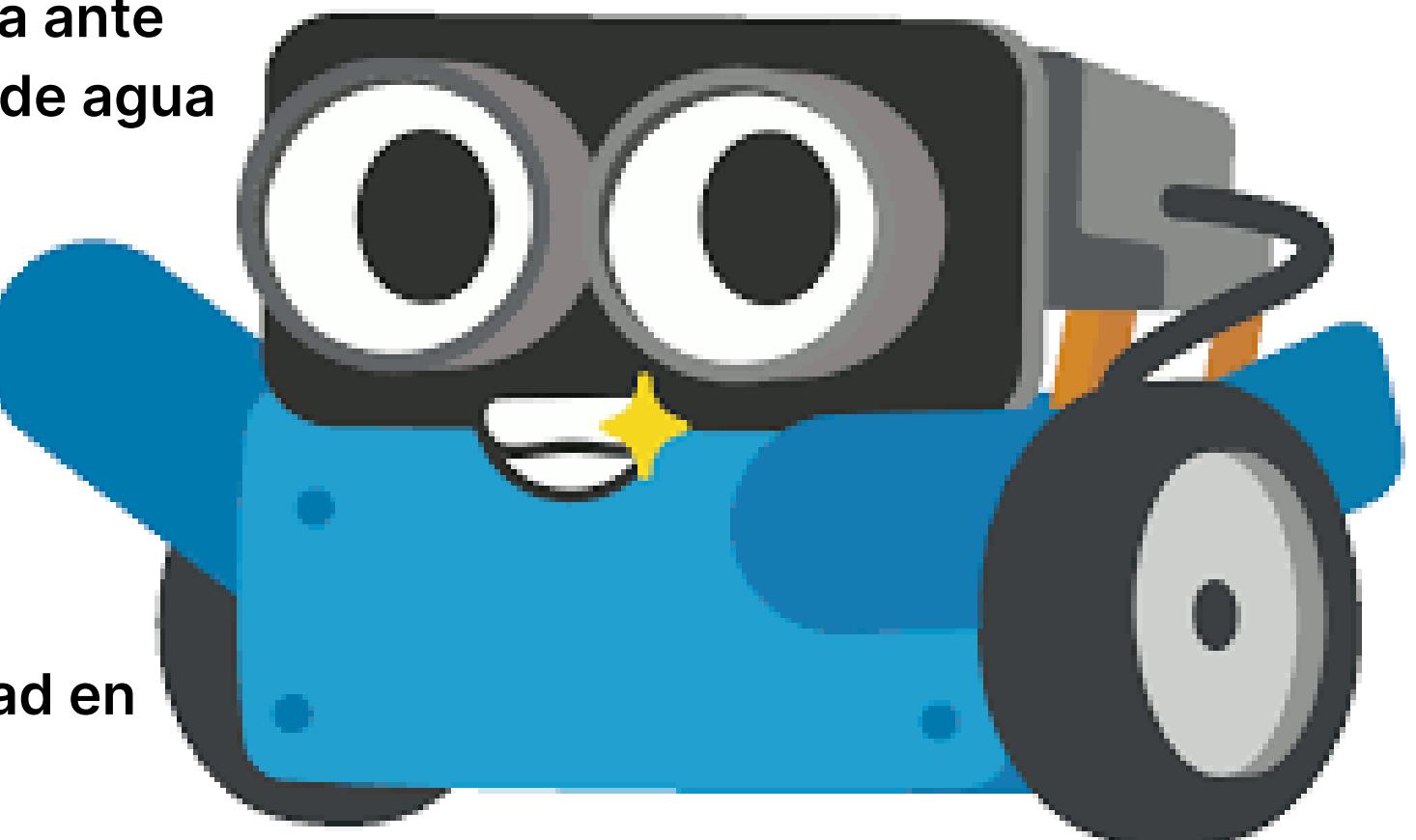
ALCANCE DEL PROYECTO

Prototipo funcional para demostrar una intervención temprana ante fuego. Capaz de detectar llamas y responder con un sistema de agua controlado por servomotor. Control híbrido: manual vía app y automático vía sensores.

LIMITACIONES DEL SISTEMA

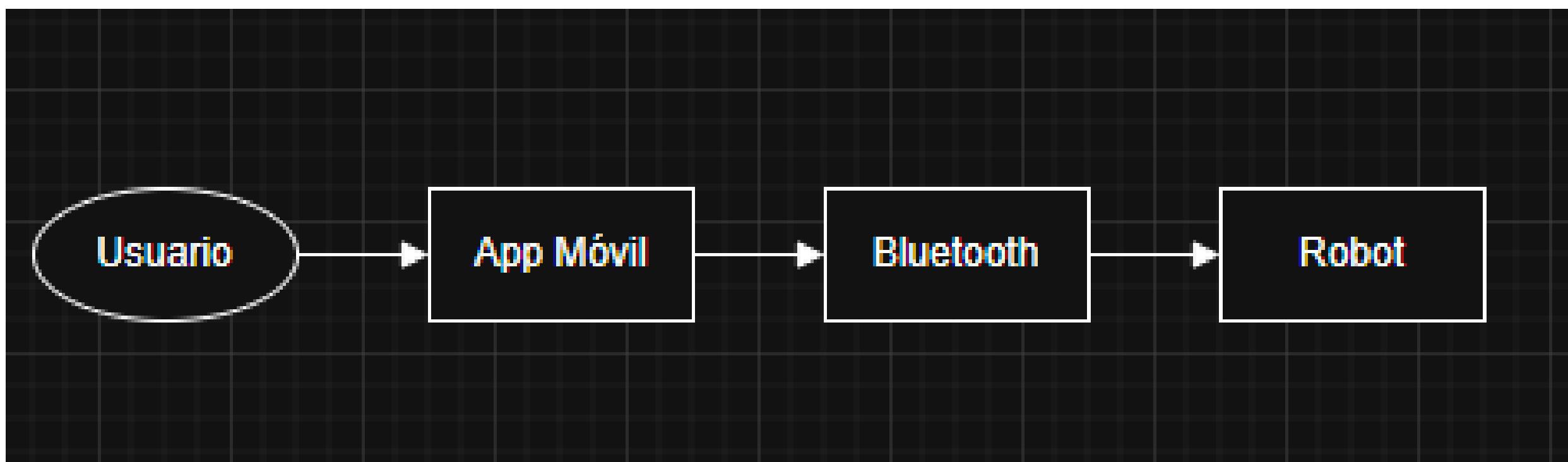
Autonomía limitada por la capacidad de la batería. Operatividad en ambientes interiores y sin obstáculos complejos.

Prototipo no apto para incendios de gran escala ni condiciones extremas.



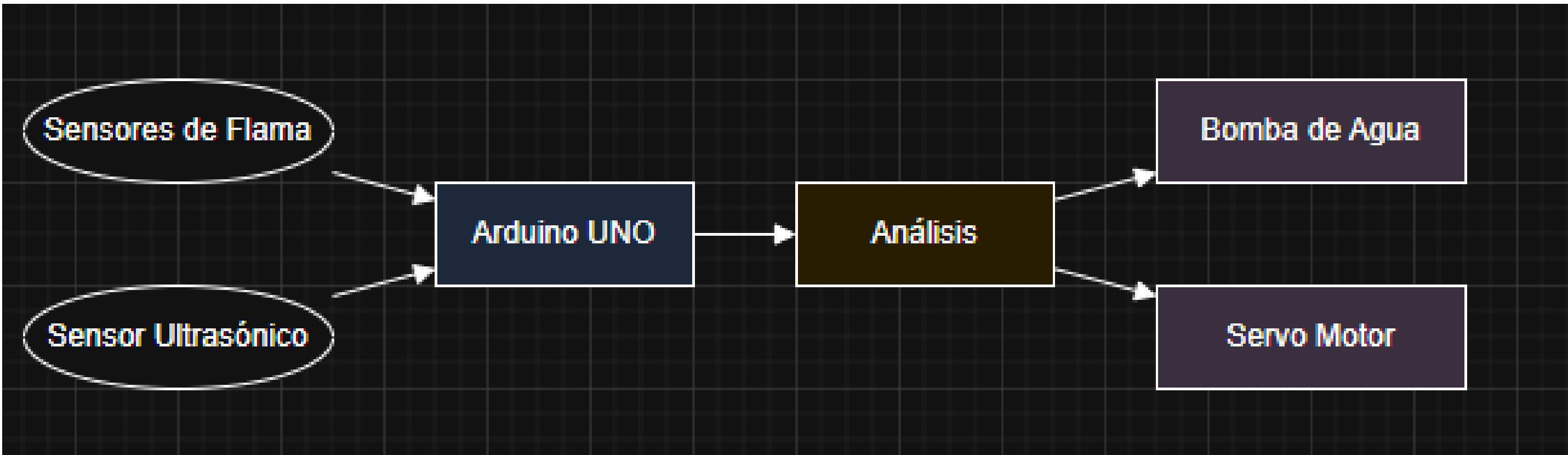
❖ ❖ ❖

MODO AUTÓNOMO (DETECCIÓN Y REACCIÓN AUTOMÁTICA):

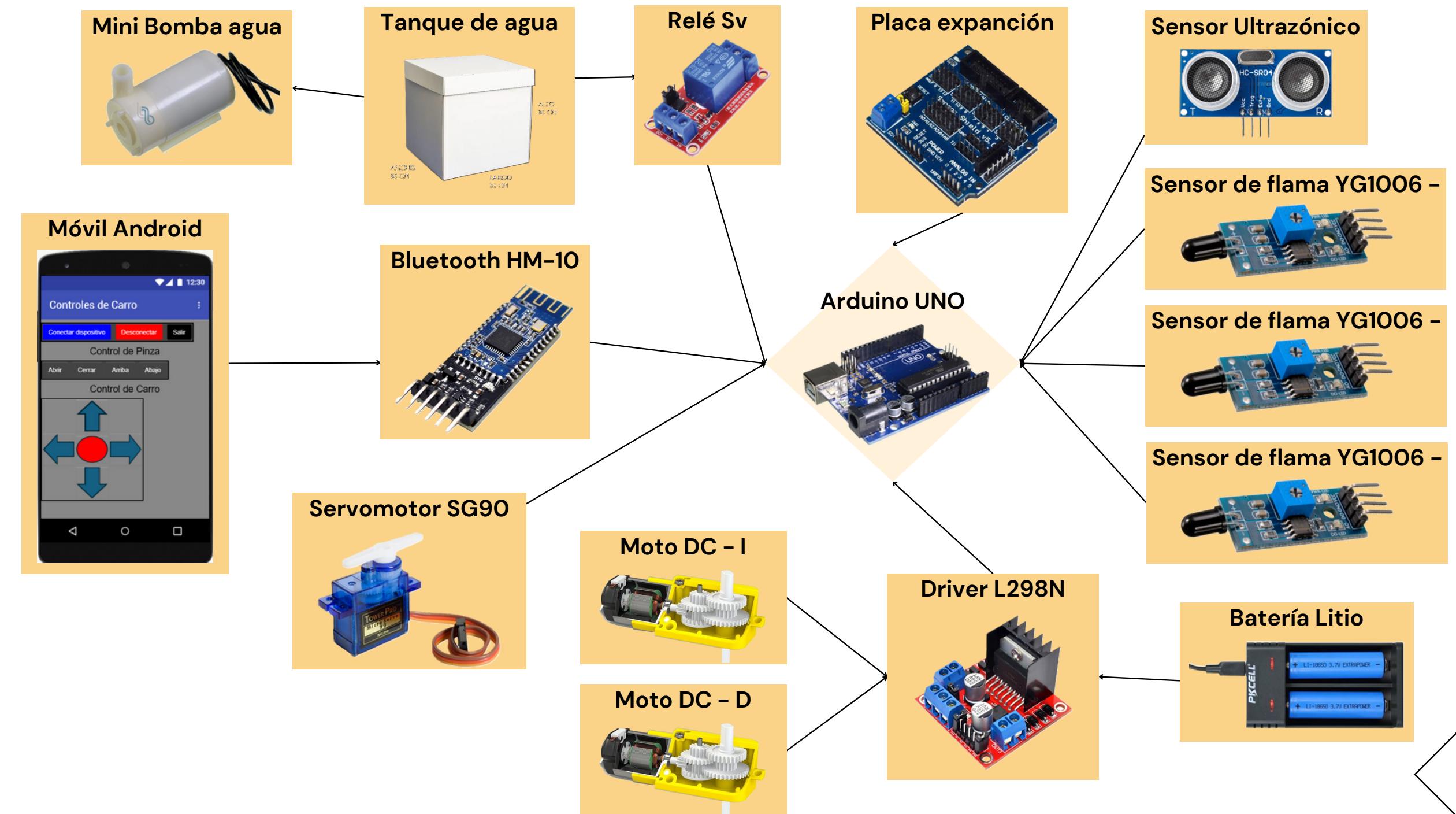


◆ ◆ ◆

CONTROL MANUAL (TELEOPERACIÓN HÍBRIDA):

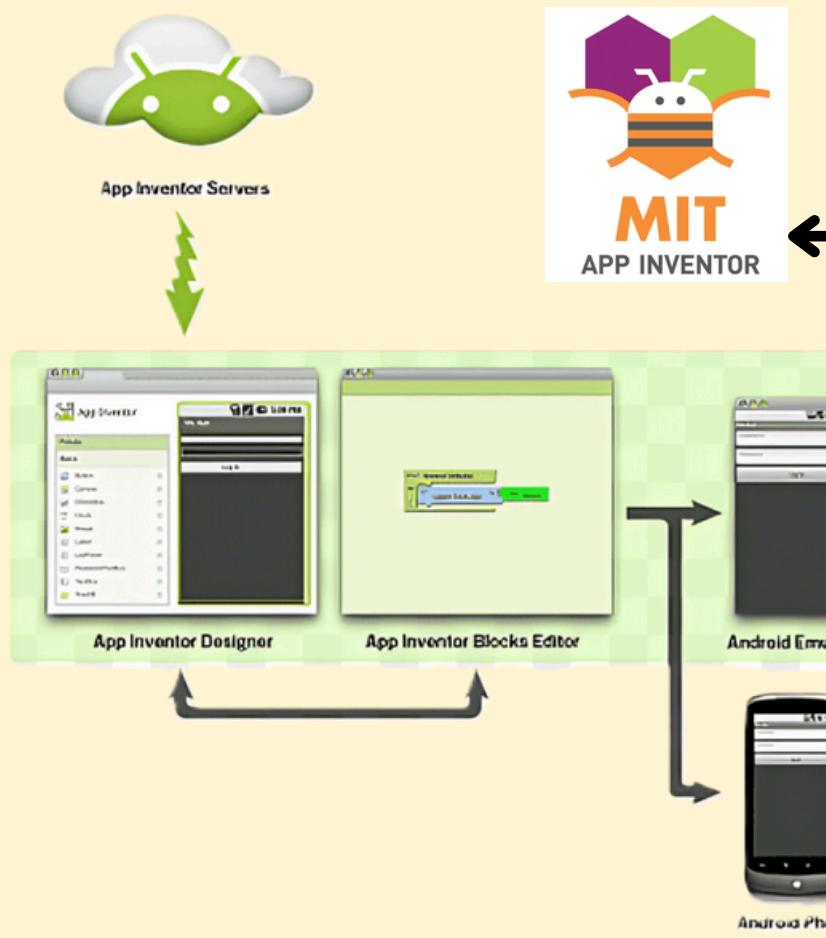


D I A G R A M A D E B L O Q U E S



Capa de presentación

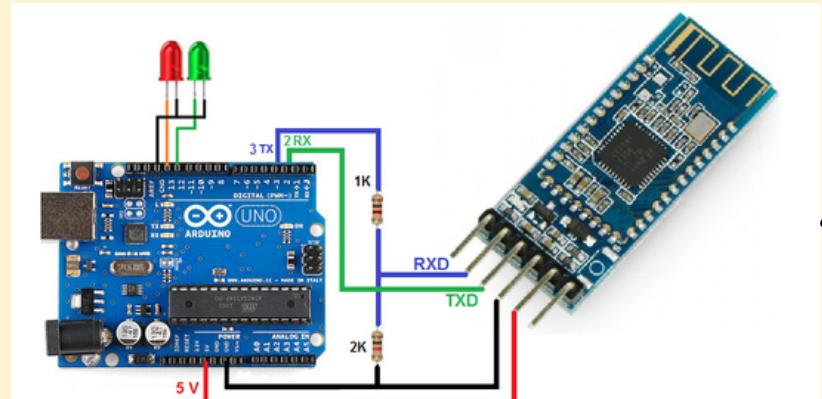
Interfaz



Capa de lógica

Protocolo UART

Módulo HM-10 - Arduino



Conección RX y TX del HM-10 y los pines RX y TX de Arduino



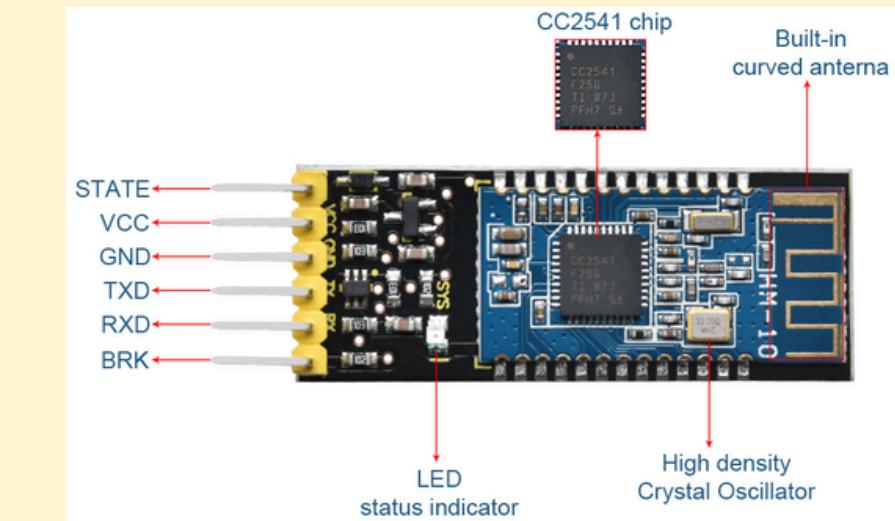
D I A G R A M A

D E A R Q U I T E C T U R A

Capa datos

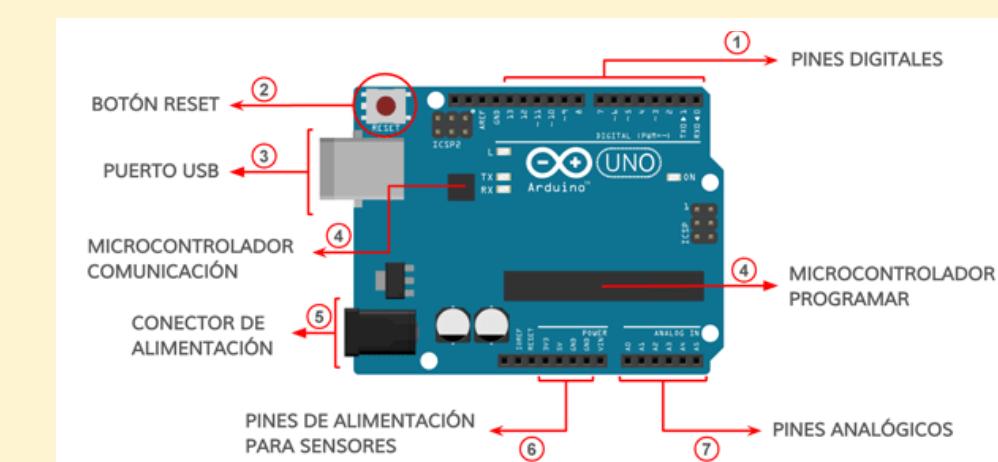
Actuación

Bluetooth HM-10



Lectura y ejecución de
Cmds AT format ASCII

Arduino UNO



Control

Diagrama de Flujo - Conexión

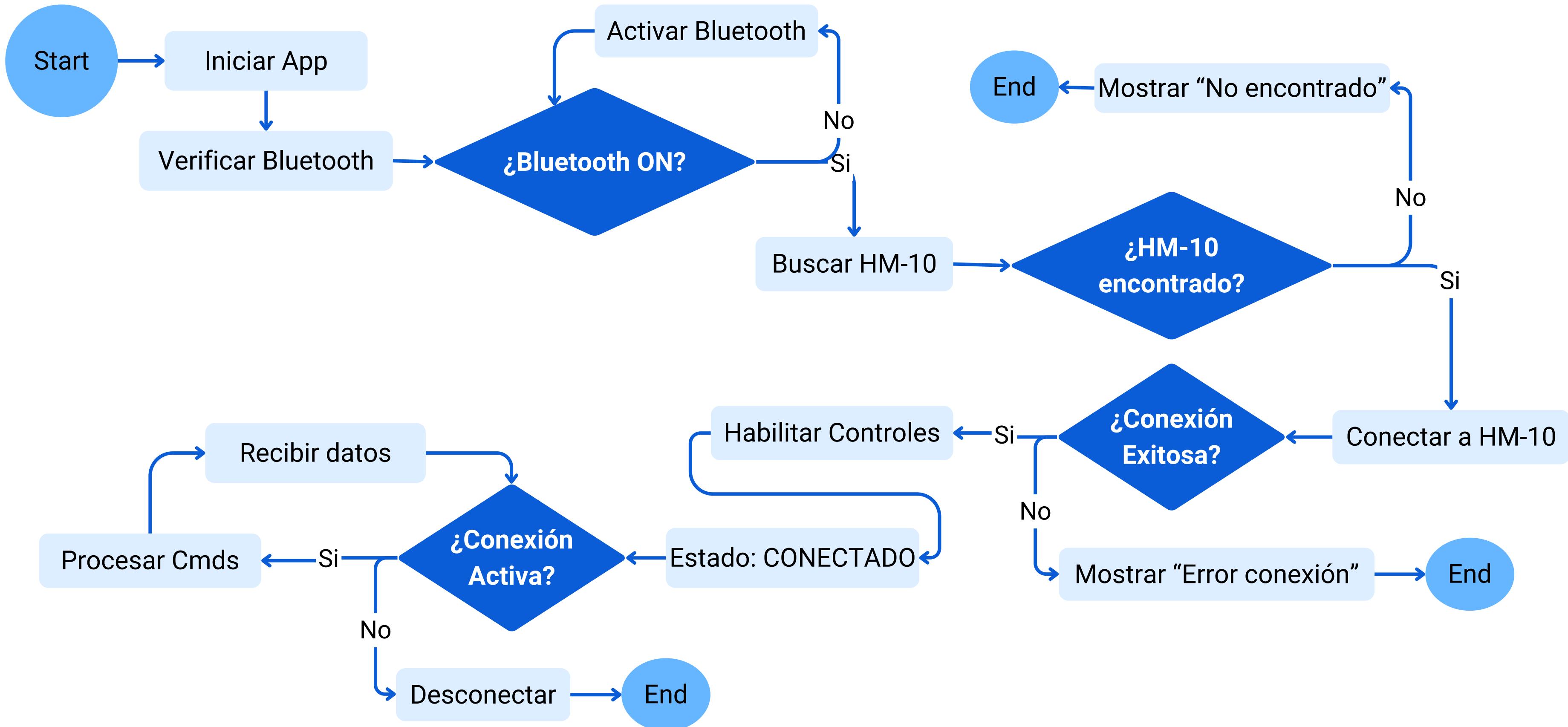
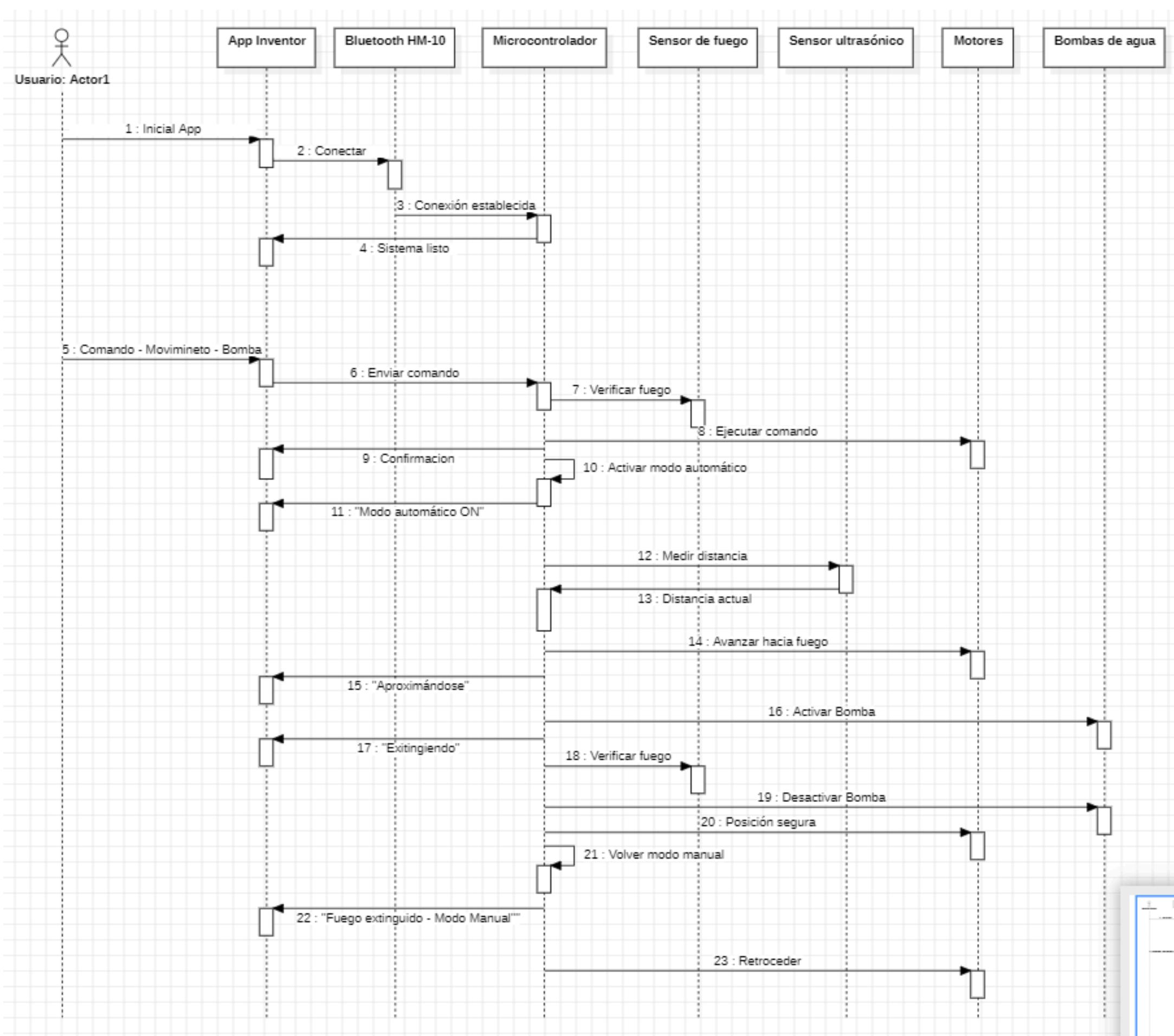


DIAGRAMA DE ARQUITECTURA



F L U J O D E D A T O S

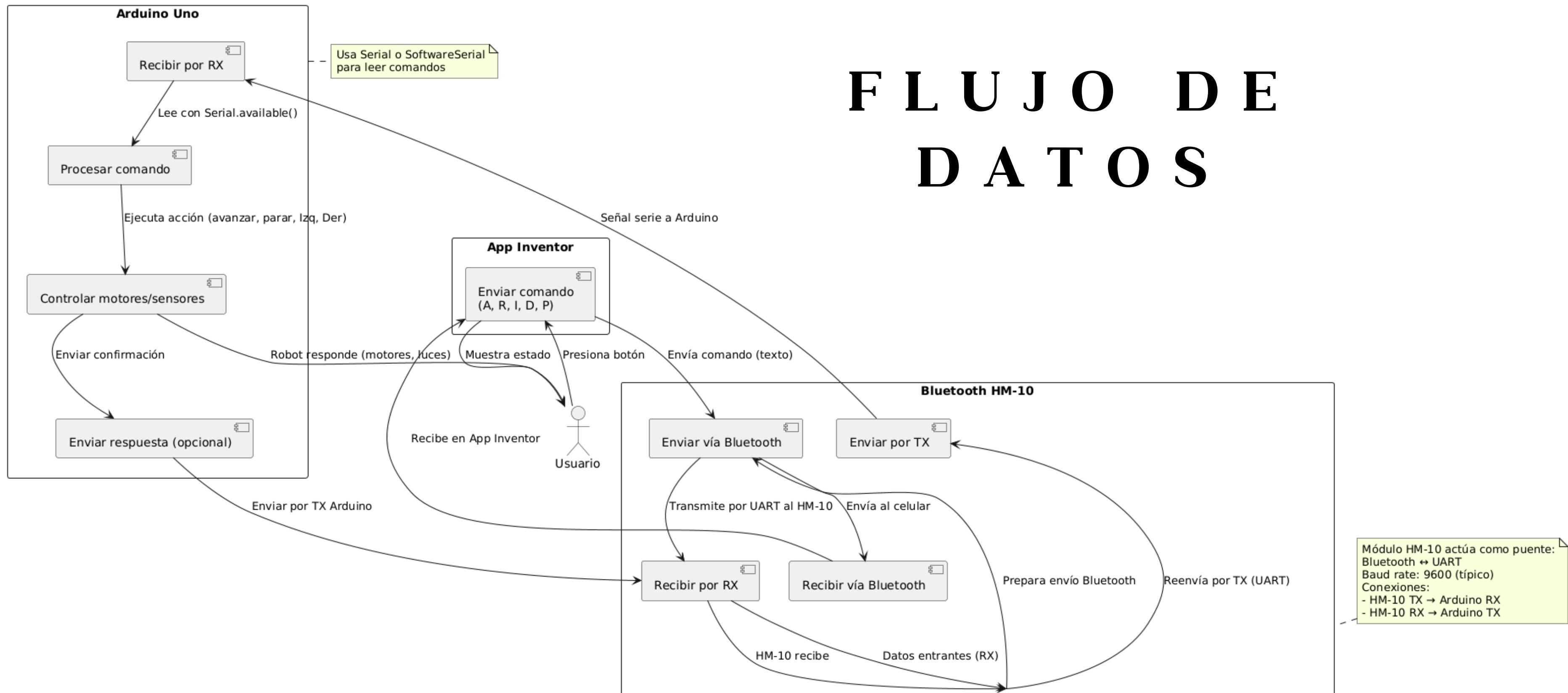
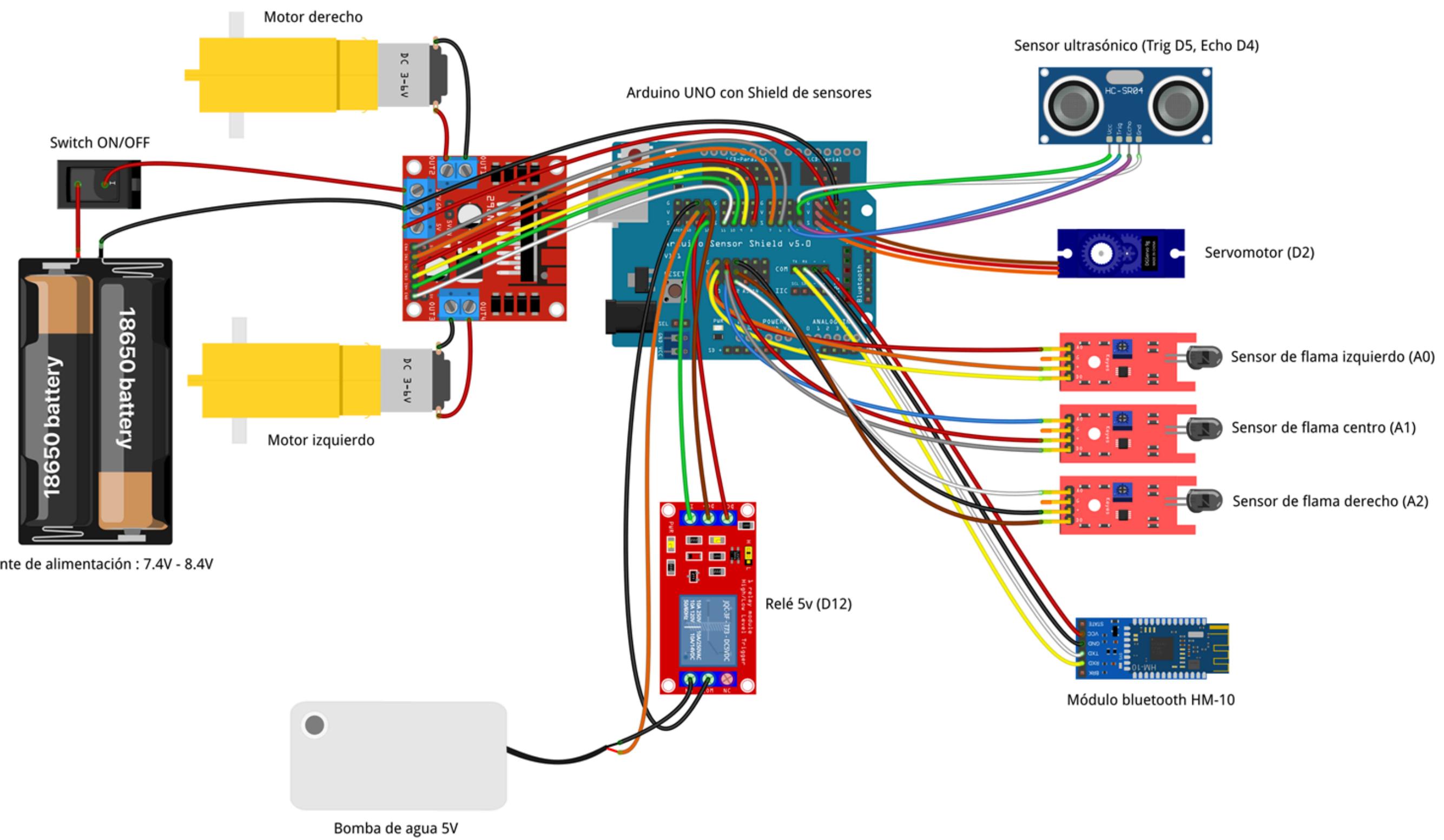


DIAGRAMA DEL CIRCUITO

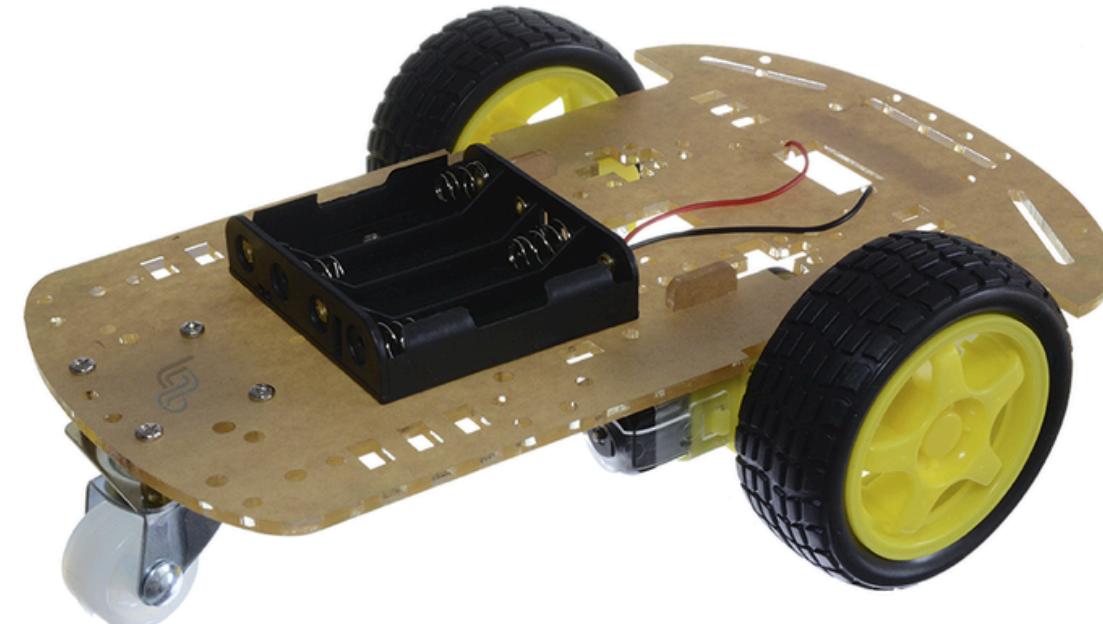


DISEÑO DE HARDWARE

Componentes

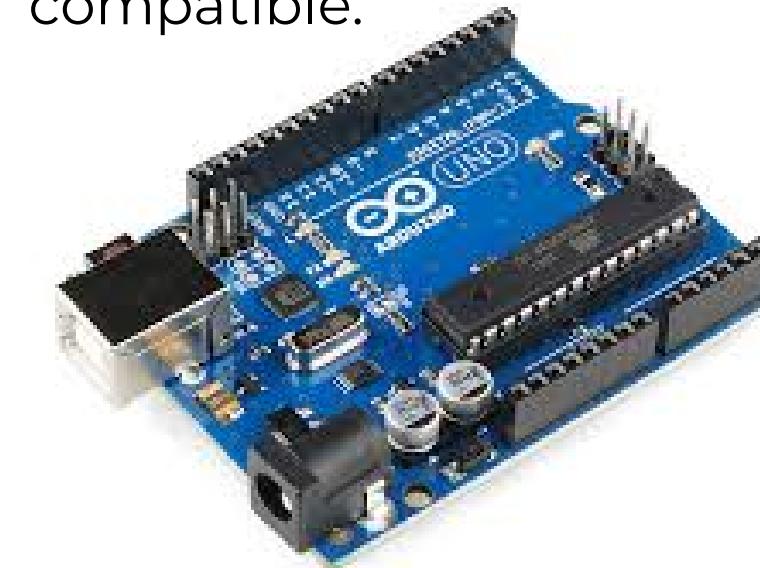
01 *Estructura 2WD*

Proporciona movilidad diferencial ideal para navegación y maniobra.



02 *Arduino UNO*

Microcontrolador confiable y ampliamente compatible.

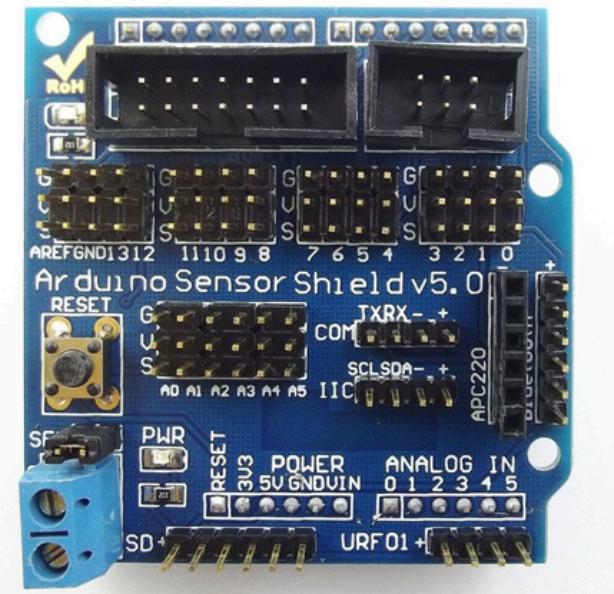


DISEÑO DE HARDWARE

Componentes

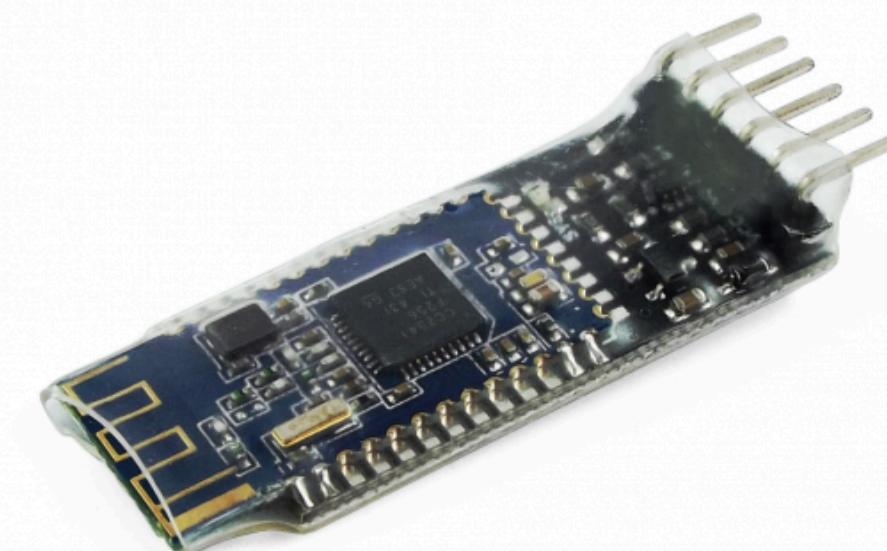
03 *Placa de expansión*

Facilita las conexiones rápidas y ordenadas con sensores y actuadores



04 *Bluetooth HM-10*

Compatible con Android y bajo consumo energético.

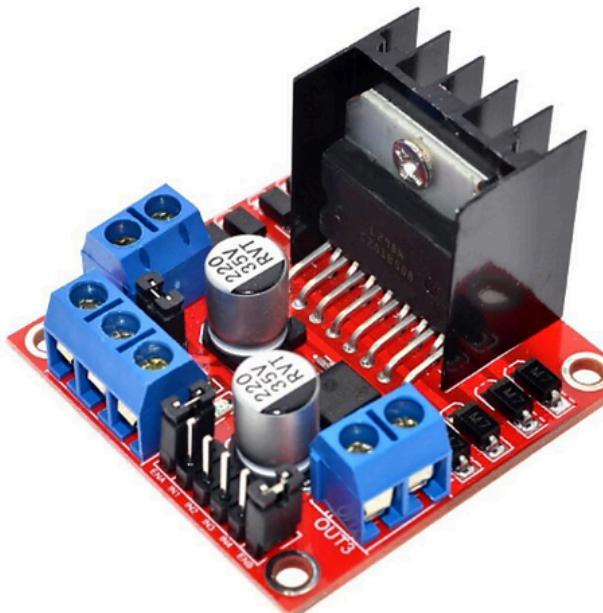


DISEÑO DE HARDWARE

Componentes

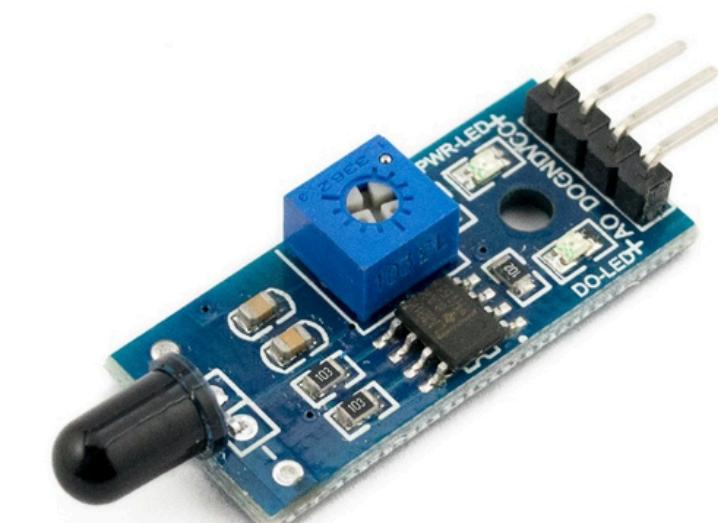
05 *Driver L298N*

Permite el control de los motores del chasis con velocidad variable.



06 *Sensores de flama YG1006*

Detectan focos de calor y permiten ubicar la fuente del incendio.



DISEÑO DE HARDWARE

Componentes

07 *Servomotor SG90*

Mueve el sistema de aspersión con precisión.



08 *Bomba de agua 3-5V*

Permite rociar agua para simular extinción de fuego.

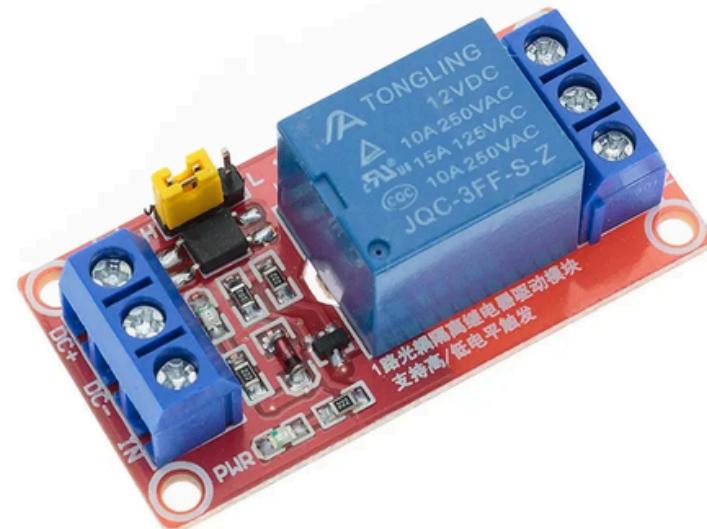


DISEÑO DE HARDWARE

Componentes

09 *Relé de 1 canal*

Controla de forma segura el encendido y apagado de la bomba.



10 *Pilas 18650*

Fuente de alimentación recargable y potente para movilidad autónoma.

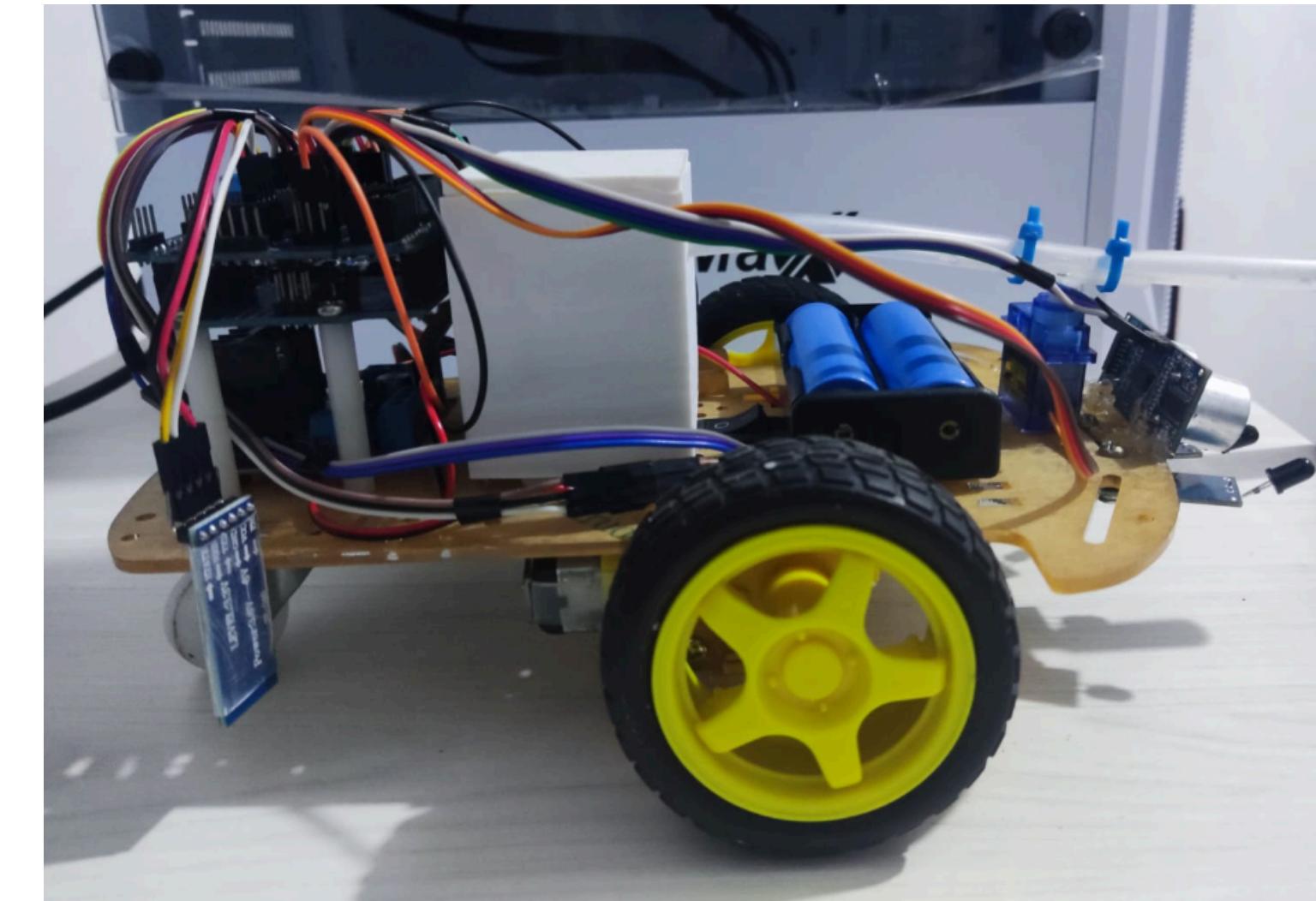
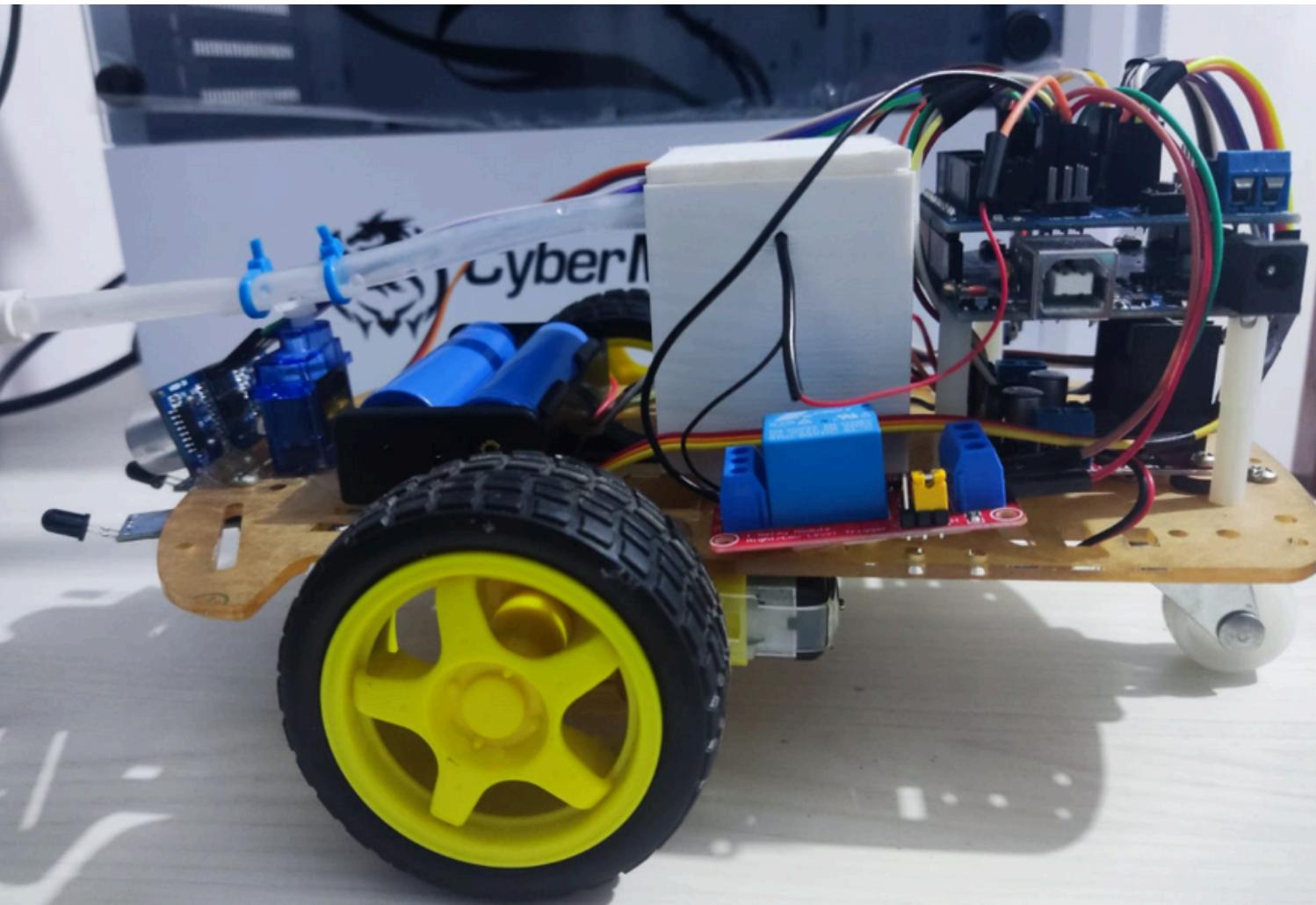


10 *Sensor ultrasónico*

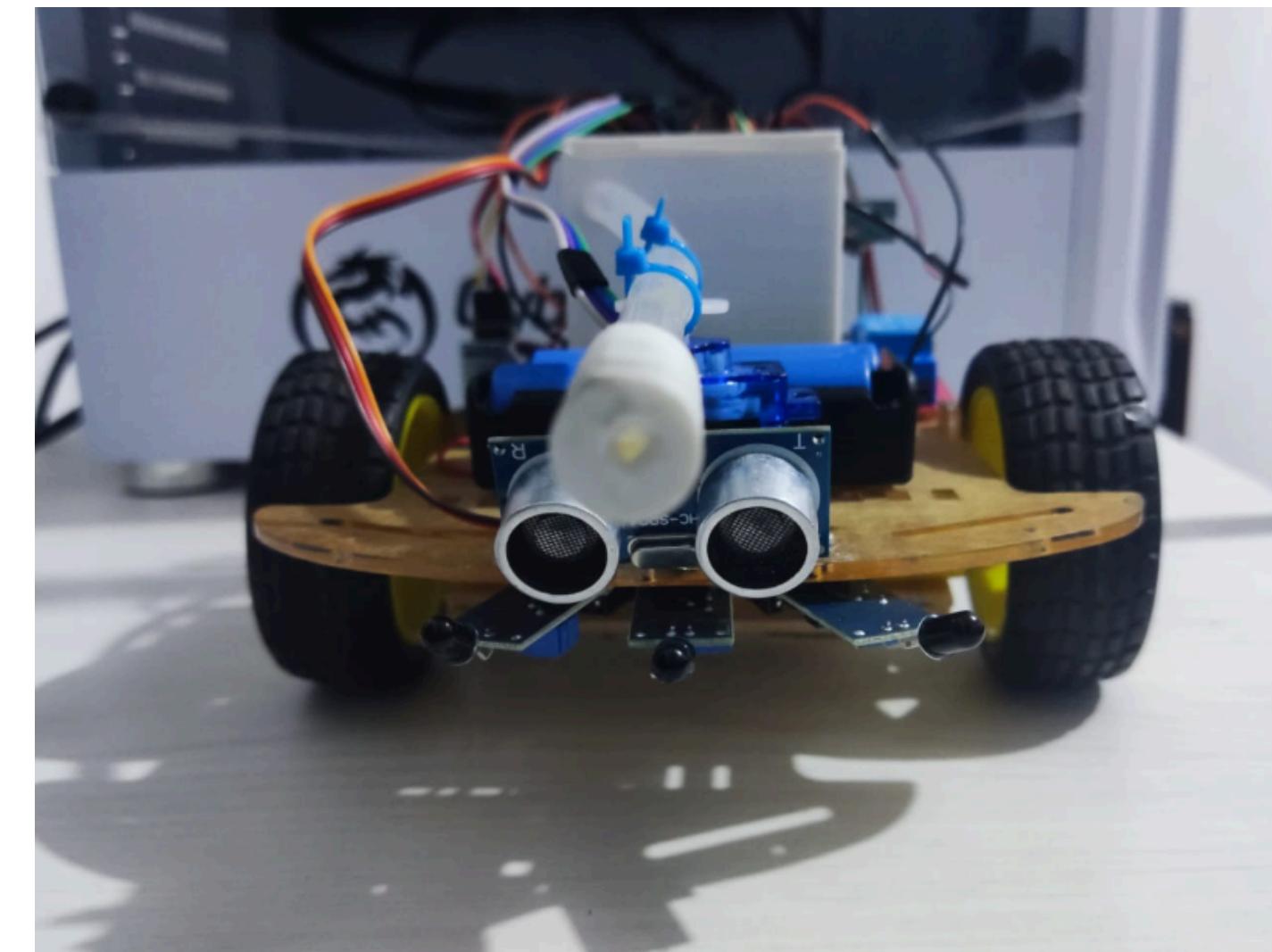
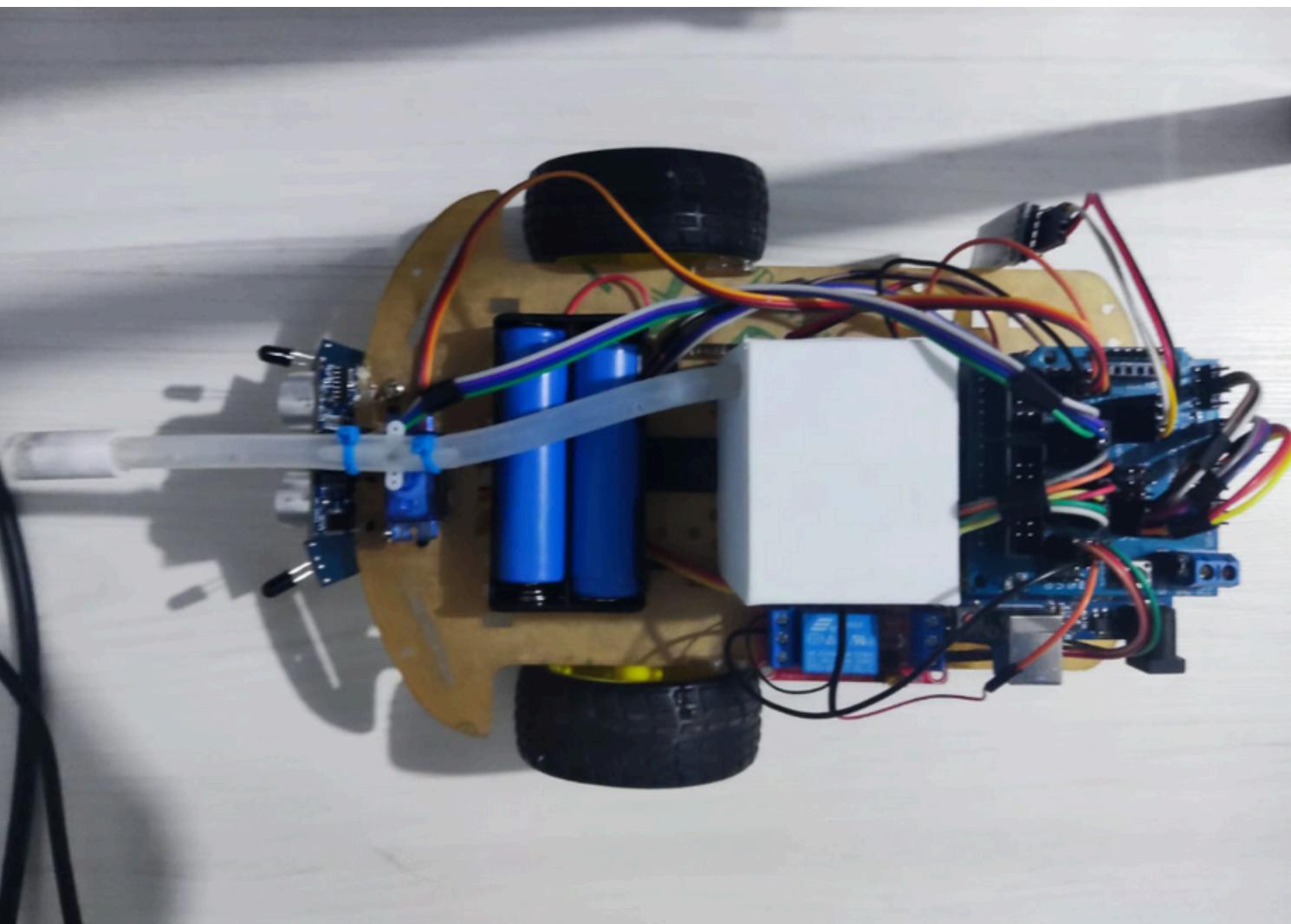
Detecta obstáculos o proximidad al fuego para activar extinción.



PROTOTIPO FINAL



PROTOTIPO FINAL



DISEÑO DE SOFTWARE Y CONTROL

01 Movimiento y control Bluetooth

```
1 // === Pines del Driver L298N ===
2 const int ENA = 6;
3 const int IN1 = 7;
4 const int IN2 = 8;
5 const int IN3 = 9;
6 const int IN4 = 10;
7 const int ENB = 11;
8
9 // === Variable para el modo actual ===
10 String modo_actual = "bluetooth";
11
12 // ===== MOVIMIENTO =====
13 void adelante() {
14     digitalWrite(IN1, HIGH);
15     digitalWrite(IN2, LOW);
16     digitalWrite(IN3, LOW);
17     digitalWrite(IN4, HIGH);
18     analogWrite(ENA, 140);
19     analogWrite(ENB, 110);
20 }
21
22 void atras() {
23     digitalWrite(IN1, LOW);
24     digitalWrite(IN2, HIGH);
25     digitalWrite(IN3, HIGH);
26     digitalWrite(IN4, LOW);
27     analogWrite(ENA, 140);
28     analogWrite(ENB, 110);
29 }
30
```

```
31 void izquierda() {
32     digitalWrite(IN1, HIGH);
33     digitalWrite(IN2, LOW);
34     digitalWrite(IN3, HIGH);
35     digitalWrite(IN4, LOW);
36     analogWrite(ENA, 100);
37     analogWrite(ENB, 100);
38 }
39
40 void derecha() {
41     digitalWrite(IN1, LOW);
42     digitalWrite(IN2, HIGH);
43     digitalWrite(IN3, LOW);
44     digitalWrite(IN4, HIGH);
45     analogWrite(ENA, 100);
46     analogWrite(ENB, 100);
47 }
48
49 void detener() {
50     digitalWrite(IN1, LOW);
51     digitalWrite(IN2, LOW);
52     digitalWrite(IN3, LOW);
53     digitalWrite(IN4, LOW);
54     analogWrite(ENA, 0);
55     analogWrite(ENB, 0);
56 }
```

```
57
58 // ===== CONTROL BLUETOOTH =====
59 void controlBluetooth () {
60     if (modo_actual != "bluetooth") {
61         detener();
62         modo_actual = "bluetooth";
63     }
64
65 if (Serial.available()) {
66     char comando = Serial.read();
67     switch (comando) {
68     case 'A': adelante(); break;
69     case 'S': atras(); break;
70     case 'I': izquierda(); break;
71     case 'D': derecha(); break;
72     case 'T': detener(); break;
73     default: break;
74     }
75 }
76 }
```

DISEÑO SOFTWARE Y CONTROL

02

Apagar Fuego

```
1 #include <Servo.h>
2
3 // === Pin del relé de la bomba ===
4 const int BOMBA = 12;
5
6 // === Servo motor para aspersión ===
7 Servo servo;
8
9 // === Variable modo y sensores (externas) ===
10 extern String modo_actual;
11 extern int sensor_izquierdo;
12 extern int sensor_centro;
13 extern int sensor_derecho;
14
15 // ===== APAGAR FUEGO =====
16 void apagarFuego() {
17     digitalWrite(BOMBA, HIGH);
18
19     for (int i = 90; i < 110; i++) {
20         servo.write(i);
21         delay(10);
22     }
23     for (int i = 110; i > 70; i--) {
24         servo.write(i);
25         delay(10);
26     }
27     for (int i = 70; i < 90; i++) {
28         servo.write(i);
29         delay(10);
30     }
31
32     digitalWrite(BOMBA, LOW);
33 }
34 }
```

DISEÑO SOFTWARE Y CONTROL

02

Detección de Fuego

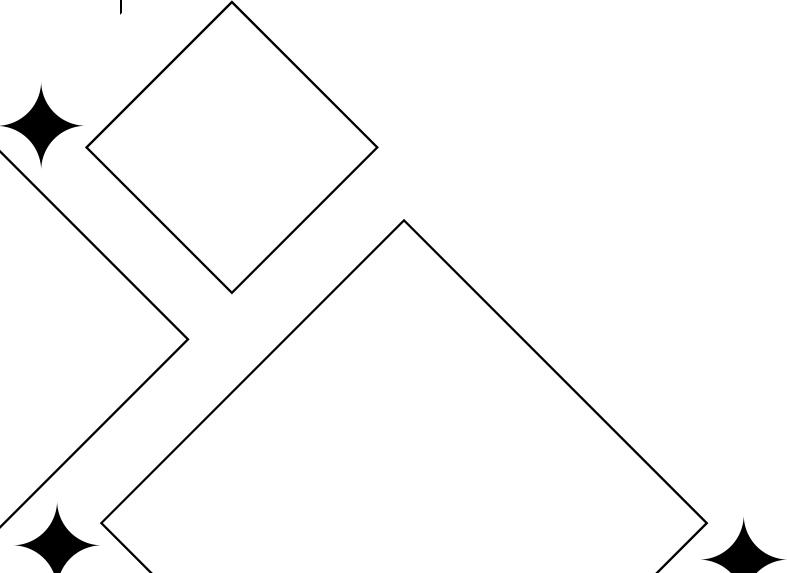
```
1 // === Pines de sensores de flama ===  
2 const int IZQ = A0;  
3 const int CENTRO = A1;  
4 const int DER = A2;  
5  
6 // === Pines del sensor ultrasónico ===  
7 const int trigPin = 5;  
8 const int echoPin = 4;  
9  
10 // === Umbrales de detección ===  
11 const int UMBRAL = 970;  
12 const int UMBRAL_CERCA = 400;  
13  
14 // === Variables globales de sensores ===  
15 int sensor_izquierdo;  
16 int sensor_centro;  
17 int sensor_derecho;  
18  
19 // === Variable modo (compartida con Bluetooth) ===  
20 extern String modo_actual;  
21
```

```
29 // ===== MODO AUTOMÁTICO: SEGUIR FUEGO =====  
30 void seguirFuego() {  
31     int distancia = obtenerDistancia();  
32     if (distancia > 0 && distancia < 20) {  
33         detener();  
34         apagarFuego();  
35     }  
36  
37     if (sensor_izquierdo < UMBRAL_CERCA || sensor_centro < UMBRAL_CERCA || sensor_derecho < UMBRAL_CERCA) {  
38         detener();  
39         apagarFuego();  
40     } else if ((sensor_izquierdo < UMBRAL && sensor_izquierdo > UMBRAL_CERCA) ||  
41             (sensor_centro < UMBRAL && sensor_centro > UMBRAL_CERCA) ||  
42             (sensor_derecho < UMBRAL && sensor_derecho > UMBRAL_CERCA)) {  
43  
44         int minValue = sensor_centro;  
45         adelante();  
46         if (sensor_izquierdo < minValue) {  
47             minValue = sensor_izquierdo;  
48             izquierda();  
49         }  
50         if (sensor_derecho < minValue) {  
51             minValue = sensor_derecho;  
52             derecha();  
53         }  
54     } else {  
55         detener();  
56     }  
57 }
```

DISEÑO SOFTWARE Y CONTROL

03

Detección de Fuego y
Envio de datos



```
59      // ====== SENSOR ULTRASÓNICO ======
60  ↘  int obtenerDistancia() {
61      long duration;
62      float distance;
63
64      digitalWrite(trigPin, LOW);
65      delayMicroseconds(2);
66      digitalWrite(trigPin, HIGH);
67      delayMicroseconds(10);
68      digitalWrite(trigPin, LOW);
69
70      duration = pulseIn(echoPin, HIGH, 30000); // timeout opcional
71      distance = duration * 0.0343 / 2;
72
73      return distance;
74  }
```

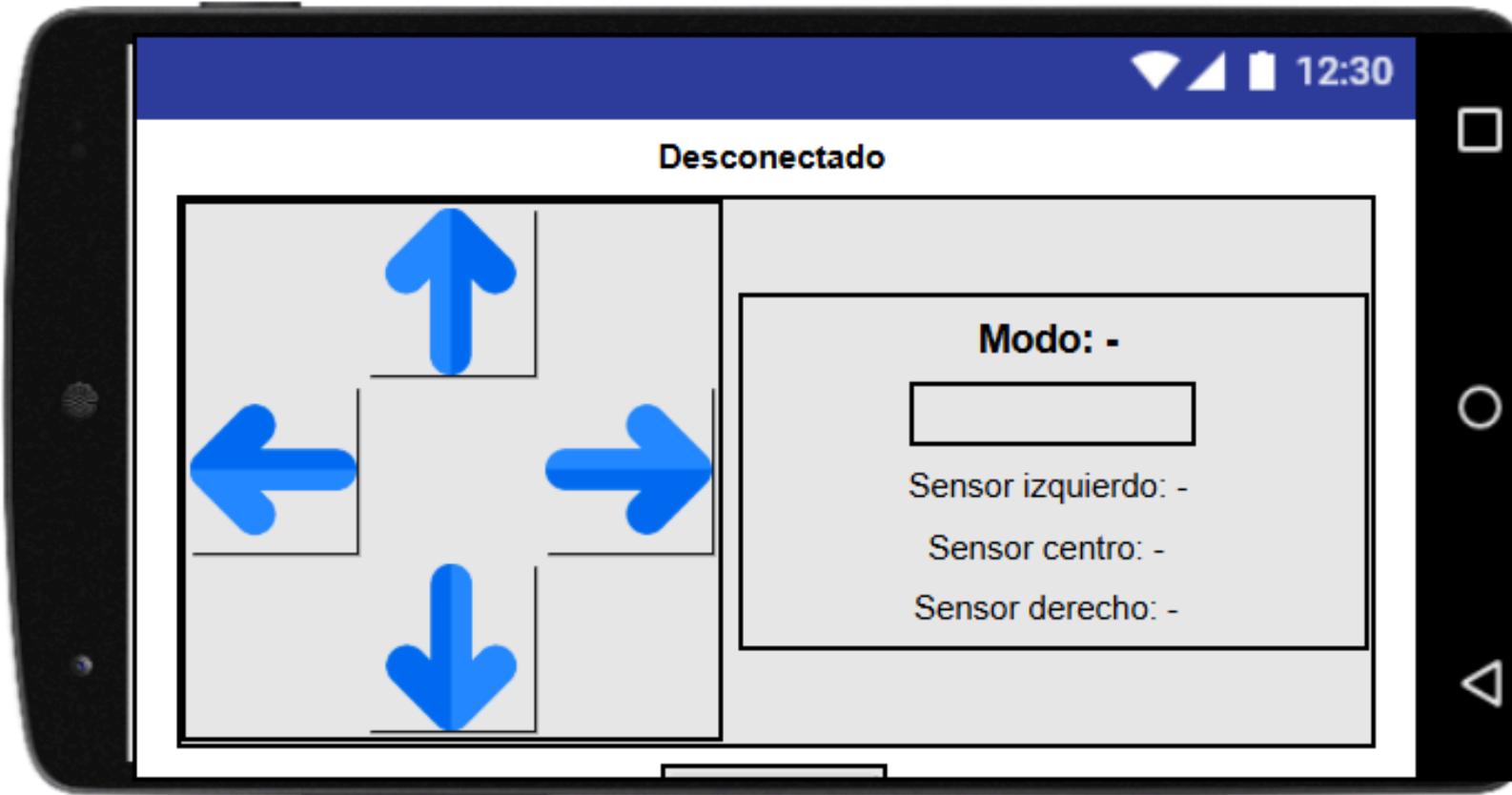
```
35      // ====== ENVÍO DE DATOS BLUETOOTH ======
36  ↘  void enviarDatosBT(int s1, int s2, int s3, String modo) {
37      String buffer;
38      buffer = String(s1) + "," + String(s2) + "," + String(s3) + "," + ((modo == "auto") ? "a" : "m") + "\0";
39      Serial.print(buffer);
40  }
```

A P L I C A C I O N M O V I L

01

Interfaz

Phone (320 x 505) ▾ Android 5+ (Material) ▾ Mostrar en el Visor los componentes ocultos



A P L I C A C I O N M O V I L

02

Programacion

cuando Screen1 .Inicializar
ejecutar llamar BluetoothLE1 .StartScanning

cuando conectar .Presionar
ejecutar llamar BluetoothLE1 .StartScanning

cuando BluetoothLE1 .DeviceFound
ejecutar poner conectar . Elementos como crear una lista vacía
poner conectar . Elementos como lista desde registro csv texto BluetoothLE1 . DeviceList

cuando conectar .DespuésDeSelección
ejecutar llamar BluetoothLE1 .ConnectWithAddress dirección seleccionar elemento de la lista recorta texto conectar Selección en " "
índice 1

cuando BluetoothLE1 .Connected
ejecutar poner Estado . Texto como " Conectado "
poner desconectar . Visible como verdadero
poner conectar . Visible como falso

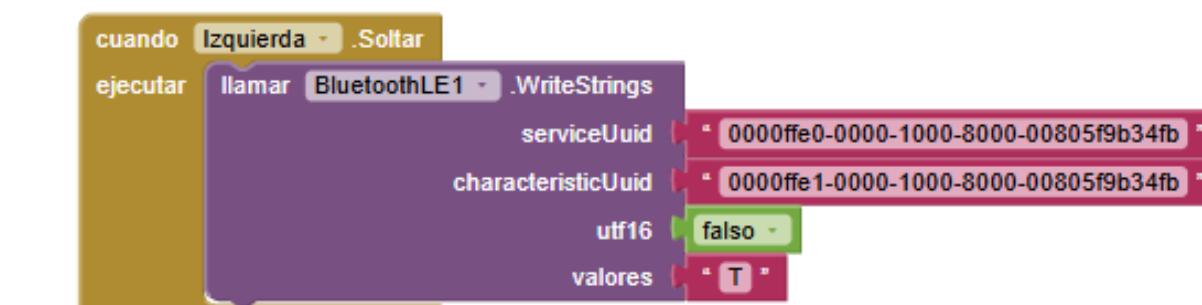
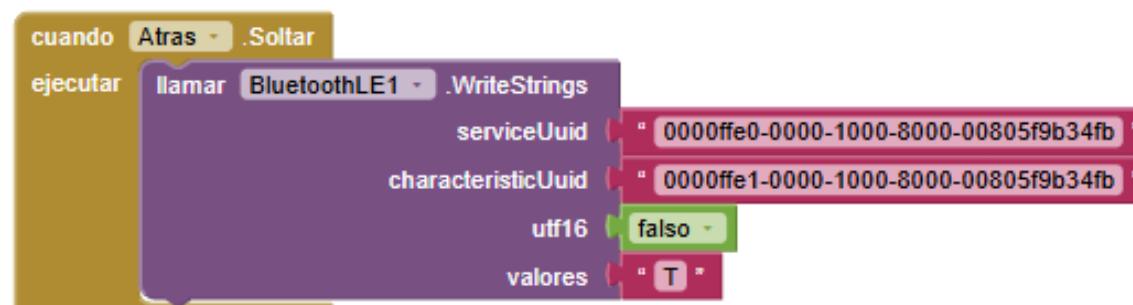
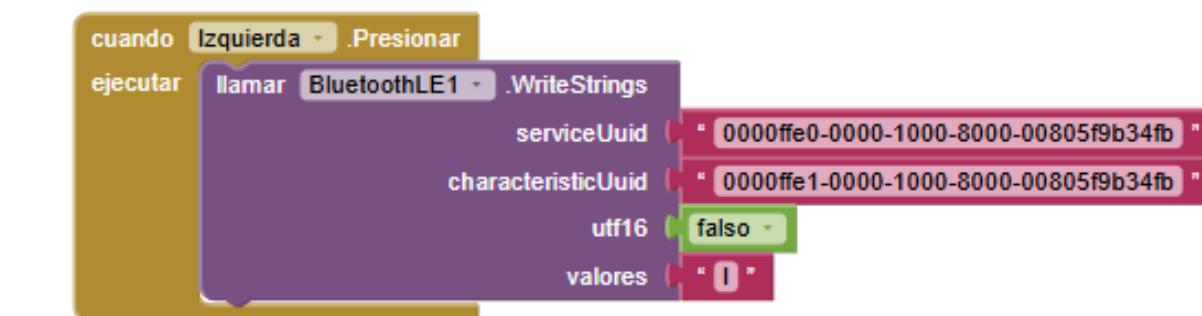
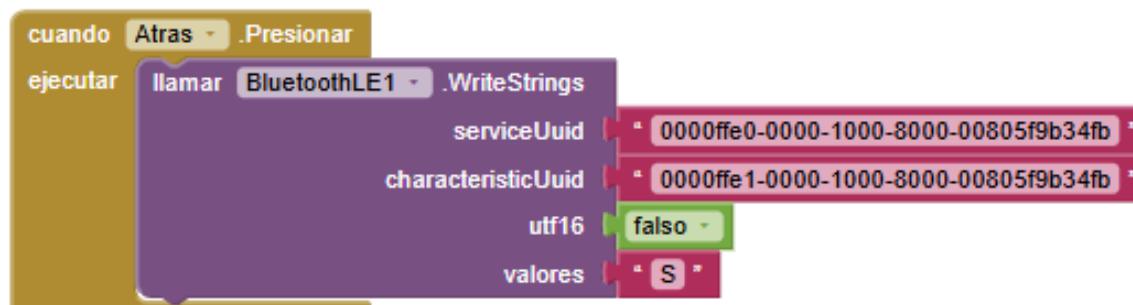
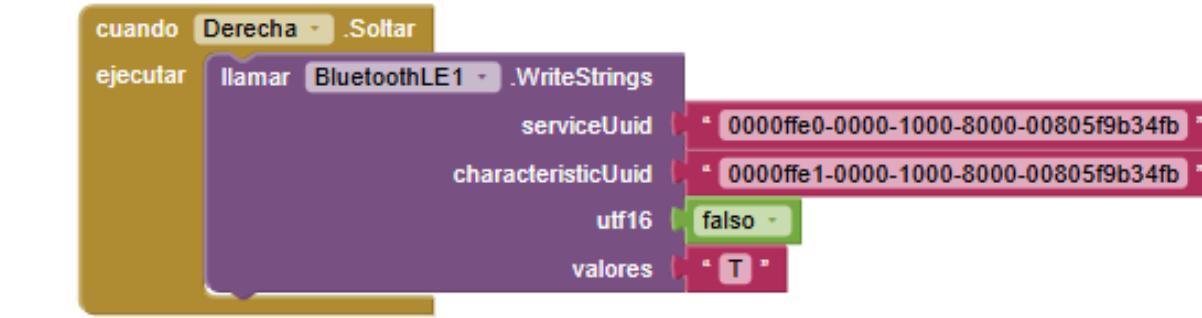
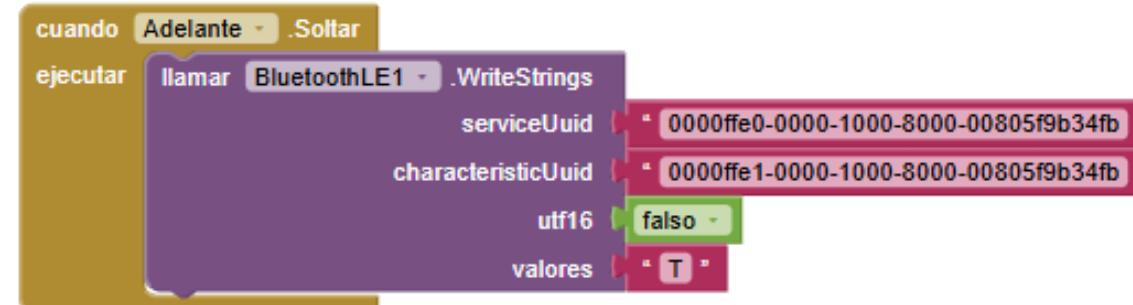
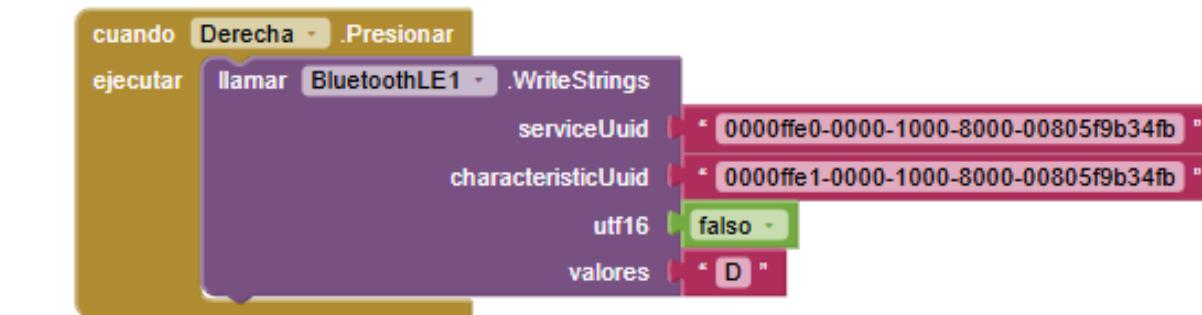
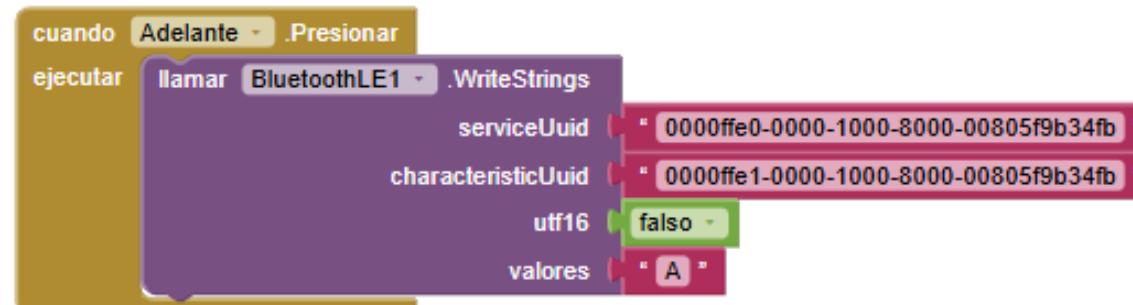
cuando desconectar .Clic
ejecutar si BluetoothLE1 . IsDeviceConnected entonces llamar BluetoothLE1 . Desconectar

cuando BluetoothLE1 .Disconnected
ejecutar poner Estado . Texto como " Desconectado "
poner desconectar . Visible como falso
poner conectar . Visible como verdadero

APLICACION MOVIL

02

Programacion



A P L I C A C I O N M O V I L

02

Programacion

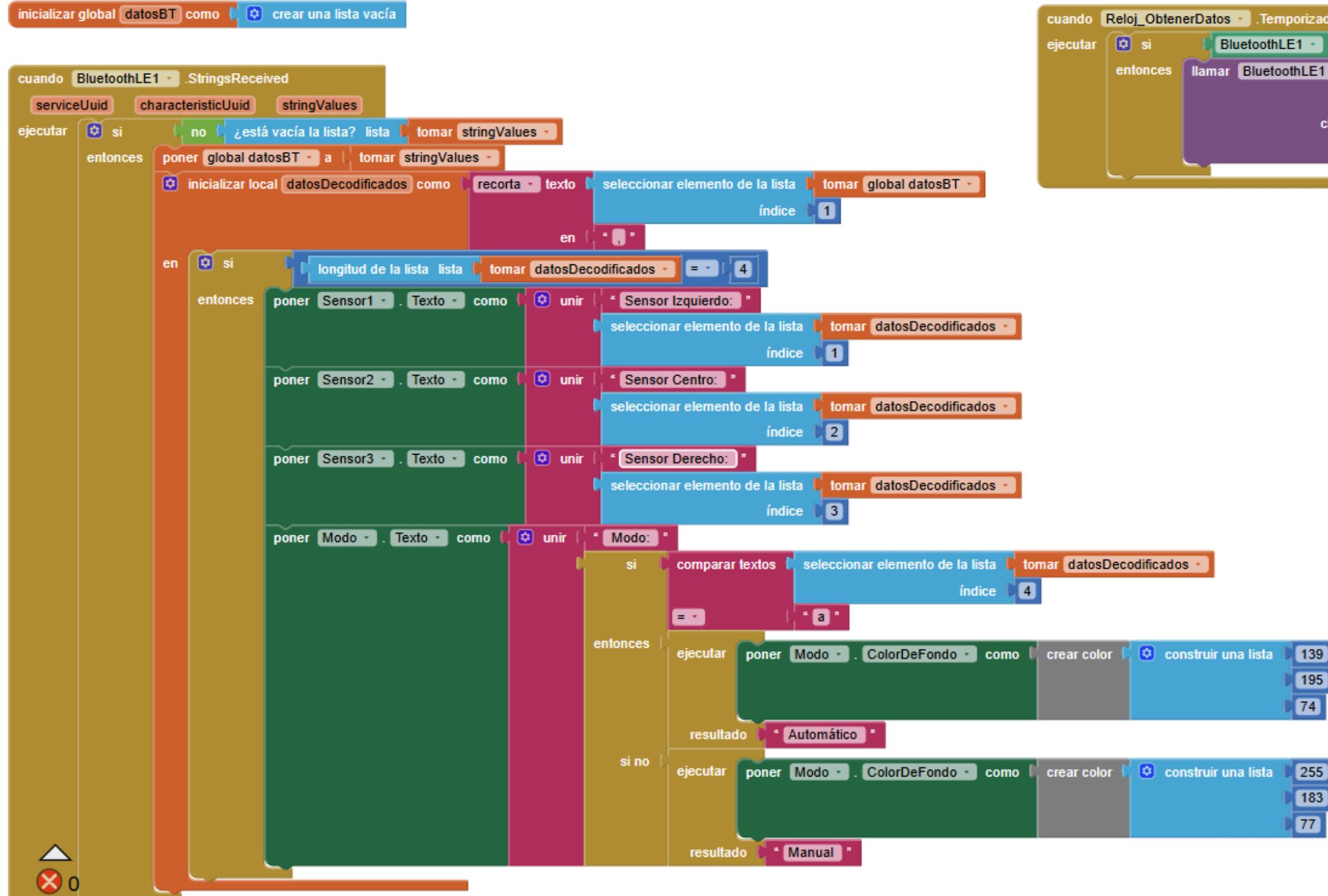
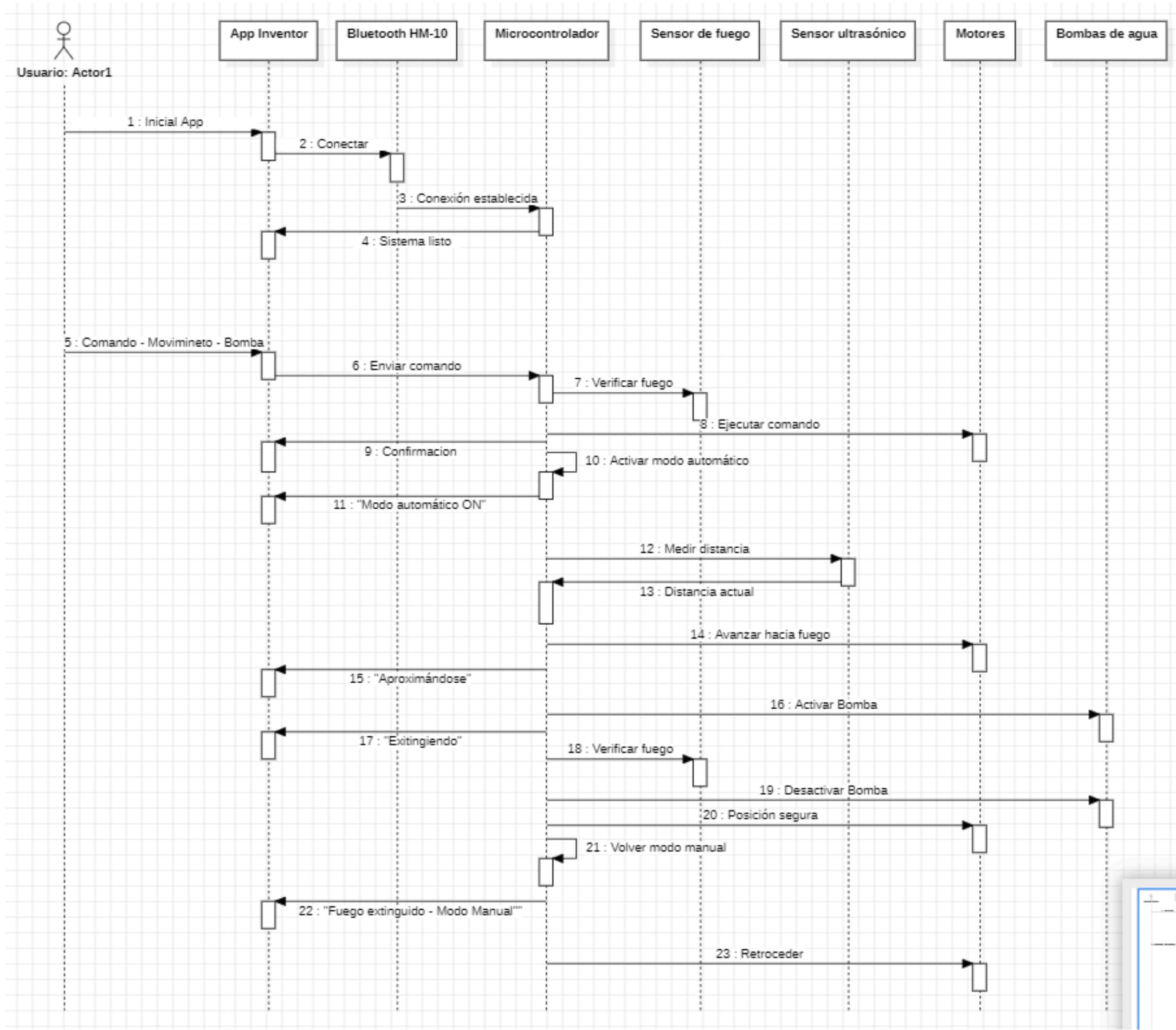


DIAGRAMA DE SUCCESSIONIA



PRUEBAS

01 1. DISEÑAR LA ESTRUCTURA MÓVIL CON TRACCIÓN DIFERENCIAL

Prueba de desplazamiento en línea recta, giros y estabilidad en el suelo.

02 2. INTEGRAR SENSORES DE FLAMA AL SISTEMA DE CONTROL CON ARDUINO

Simulación de fuentes de calor para validar detección por los sensores YG1006.

03 3. IMPLEMENTAR SISTEMA DE ASPERSIÓN CON BOMBA, RELÉ Y SERVOMOTOR SG90

Activación automática del sistema frente a flama y recorrido del servomotor.

04 4. DESARROLLAR APP PARA TELEOPERACIÓN BLUETOOTH

Prueba de conexión desde celular vía HM-10 y control remoto de movimiento.



R E S U L T A D O S O B T E N I D O S

- Todas las pruebas fueron exitosas, cumpliendo los criterios de funcionamiento definidos para cada módulo.
- El sistema de detección de flama reaccionó en menos de 2 segundos.
- El sistema de aspersión logró enfocar correctamente al frente del robot.
- La aplicación móvil respondió sin latencia perceptible (Bluetooth ~5 metros).
- El robot maniobra de forma efectiva en espacios reducidos gracias a su configuración diferencial.

R E S U L T A D O S O B T E N I D O S

Conectado

Modo: Automático

Sensor Izquierdo: 957
Sensor Centro: 688
Sensor Derecho: 967

Desconectar

Conectado

Modo: Manual

Sensor Izquierdo: 1023
Sensor Centro: 1023
Sensor Derecho: 1023

Desconectar



D E M O S T R A C I O N

TELEOPERACION HIBRIDA

Vamos a controlar el carrito
Luego cuando detecte fuego
Estará de forma automática



CONCLUSIONES

- **Integración exitosa de sistemas híbridos:**

El prototipo logró combinar eficazmente la teleoperación manual y el modo automático de detección y extinción de fuego, cumpliendo con el objetivo principal del proyecto.

- **Fiabilidad de sensores y actuadores:**

Los sensores de flama YG1006 y el sistema de aspersión con bomba, relé y servomotor SG90 funcionaron de forma estable, permitiendo una respuesta adecuada ante la presencia de focos de calor.

- **Eficiencia en la comunicación Bluetooth:**

La aplicación desarrollada en MIT App Inventor demostró ser una herramienta funcional para controlar remotamente el robot a través del módulo HM-10, con una conexión rápida y sin interferencias.

- **Aplicabilidad del diseño en contextos educativos y de simulación:**

El robot es ideal para prácticas académicas, demostraciones tecnológicas o simulacros de intervención, mostrando un diseño accesible, reproducible y con fundamentos en la robótica móvil.

TRABAJOS FUTUROS

- **Integración de sensores adicionales**

Incorporar sensores de gas o temperatura para ampliar la capacidad de detección ante diferentes tipos de emergencias.

- **Sistema de navegación autónoma mejorado**

Implementar algoritmos de mapeo y evitación de obstáculos (como A* o SLAM) para permitir desplazamientos más inteligentes en entornos complejos.

- **Notificaciones en tiempo real**

Añadir un módulo GSM o WiFi para enviar alertas o notificaciones al usuario mediante mensajes de texto o correo electrónico ante detección de fuego.

- **Optimización de la interfaz de la app**

Mejorar la aplicación Android con una interfaz gráfica más amigable, visualización en tiempo real de datos y funciones de control avanzado.

- **Mejoras en autonomía energética**

Diseñar un sistema de gestión de energía que permita mayor duración operativa, incluyendo carga por paneles solares o baterías de mayor capacidad.

**GRACIAS POR
SU ATENCIÓN**

www.unsitiogeniales.es