

Taller Inicial Mecatrónica 2025

URBAN-DOMO

PROYECTO MECATRÓNICO

Implementación de un sistema domótico con tecnología Arduino

INTEGRANTES:

Maccarini Marcos Luciano - 14370

Beningazza Gerardo - 14062

Dinnocenzo Sofia Clara - 14113

ÍNDICE

1. Introducción.....	3
2. Descripción general.....	3
3. Diagrama de bloques.....	6
4. Lista de materiales.....	6
5. Descripción de subsistemas.....	7
6. Diagrama de Conexionado.....	8
7. Diagrama de flujo.....	9
8. Documentación código.....	9
9. Dificultades, desafíos y simplificaciones realizadas.....	11
10. Conclusiones, aprendizajes y trabajos futuros.....	11
ANEXO.....	12

1. Introducción

Este proyecto propone el diseño e implementación de un sistema domótico dirigido a espacios de alta concurrencia —como un shopping o un supermercado— cuya finalidad es optimizar el control ambiental, el consumo energético y la gestión del estacionamiento mediante soluciones sencillas, económicas y escalables basadas en microcontroladores.

2. Descripción general

Consiste en el desarrollo de un sistema domótico orientado a la automatización de un centro comercial.

El objetivo principal es optimizar el funcionamiento general del edificio mediante el control automático de iluminación, climatización, y acceso, contribuyendo a la eficiencia energética y al confort de los usuarios.

Una parte del sistema, concretamente el de control de temperatura, estará implementado en un entorno interior de gran superficie, con alto flujo de personas y múltiples áreas funcionales (locales, pasillos, patios de comida, etc). El resto de los sistemas (iluminación y acceso de vehículos) se implementarán en el exterior del edificio.

Deberá ser operado y supervisado por personal técnico y administrativo del centro comercial, quienes podrán monitorear y controlar los distintos subsistemas a través de una interfaz central.

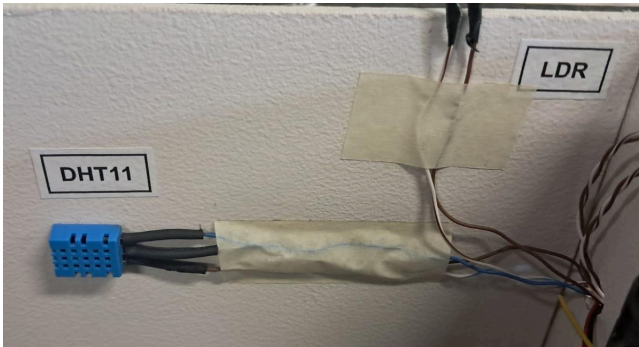
El sistema deberá operar en condiciones variables de temperatura, iluminación y ocupación, garantizando una respuesta confiable y segura en todo momento.

En función de este contexto seleccionamos los siguientes sensores:

- Sensor de movimiento de tipo ultrasónico. Es preferible por sobre el infrarrojo, porque no depende de la luz, tiene mayor alcance y es más confiable ante polvo y condiciones ambientales variables.
- DHT11 como sensor de temperatura, debido a su capacidad para medir simultáneamente la temperatura y la humedad del ambiente, facilitando el control climático dentro del edificio. Es un componente de bajo costo, fácil conexión y buen rendimiento para aplicaciones interiores.
- LDR para control de luminosidad debido a su bajo costo, simplicidad de conexión y capacidad para detectar variaciones graduales de luz.

El proyecto está compuesto por tres módulos principales, cada uno con una función específica, pero integrados bajo una misma plataforma de control basada en un microcontrolador Arduino.

- Control de temperatura interior: utiliza un sensor DHT11 para medir la temperatura y humedad del ambiente. En función de los valores registrados, el sistema acciona un ventilador o sistema de refrigeración cuando la temperatura supera un umbral preestablecido, contribuyendo al confort térmico de los usuarios y a la eficiencia energética del lugar.

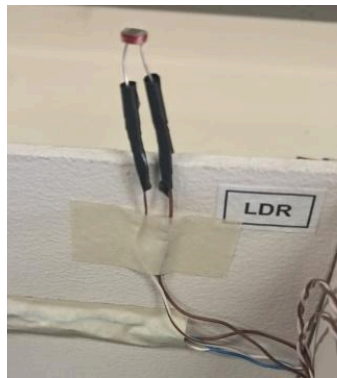


Sensor de humedad y temperatura



Sistema de refrigeración

- Control de iluminación exterior: implementa un sensor LDR (Light Dependent Resistor) para detectar el nivel de luz solar. Según la cantidad de luminosidad ambiente, el sistema enciende o apaga las luces exteriores de forma automática, optimizando el consumo eléctrico y manteniendo una correcta iluminación en todo momento.

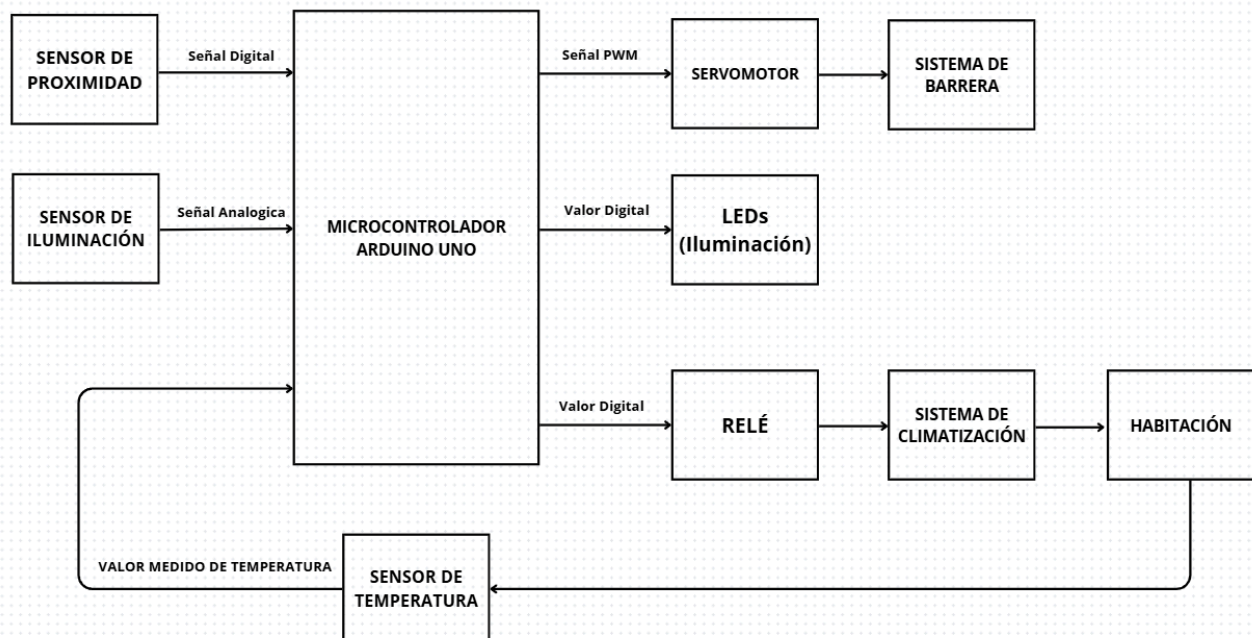


- Control de barrera de estacionamiento: se basa en un sensor de proximidad ultrasónico, encargado de detectar la presencia de un vehículo frente a la barrera. Cuando un automóvil se aproxima, el sistema activa un servomotor que eleva la barrera, permitiendo el acceso. Una vez que el vehículo ha pasado, la barrera vuelve a su posición inicial.



Cada uno de estos sistemas opera de manera independiente, aunque están conectados a un mismo microcontrolador que centraliza el procesamiento de datos y la toma de decisiones. La estructura modular del diseño facilita el mantenimiento, la ampliación futura y la integración con otros sistemas domóticos, como control de acceso, monitoreo remoto o gestión de energía.

3. Diagrama de bloques



El sistema mecatrónico diseñado integra sensores, actuadores y un microcontrolador Arduino Uno para automatizar el control de iluminación, temperatura y acceso mediante

una barrera. Los sensores de proximidad, iluminación y temperatura permiten obtener información del entorno en tiempo real, que es procesada por el microcontrolador, encargado de tomar decisiones según las condiciones detectadas.

Como actuadores, se emplea un servomotor para accionar el sistema de barrera, un conjunto de LEDs para la iluminación automática y un relé que controla el funcionamiento del sistema de climatización. Además, el sensor de temperatura proporciona retroalimentación al microcontrolador, cerrando el lazo de control de la climatización para mantener condiciones adecuadas.

4. Lista de materiales

ITEM	DESCRIPCIÓN	MARCA	CÓDIGO	CANTIDAD
1	Placa Arduino UNO R3	Arduino	A000066	1
2	Protoboard (placa de pruebas)	Elegoo	EL-CP-003	1
3	Sensor ultrasónico HC-SR04	Genérico	HC-SR04	1
4	Servo motor SG90	Tower Pro	SG90	1
5	LED(diodo emisor de luz)	-	-	2
6	LDR(Light Dependent Resistor)	-	-	1
7	Sensor DHT11	-	-	1

5. Descripción de subsistemas

Subsistema eléctrico: el sistema se alimenta principalmente desde la computadora a través del puerto USB, que suministra energía al Arduino y a los sensores conectados (LDR, DHT11 y ultrasónico). De forma independiente, el cooler recibe alimentación de un transformador externo de 12V, el cual es controlado mediante un relé conectado al Arduino.

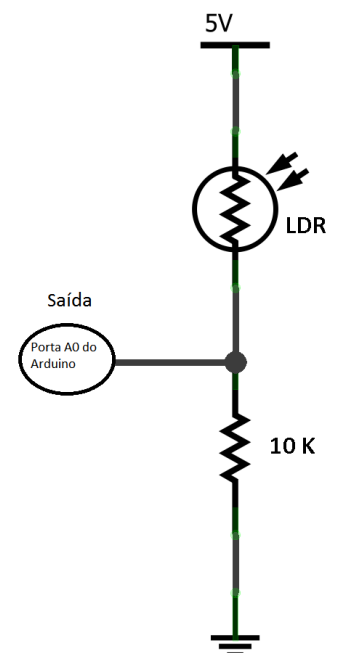
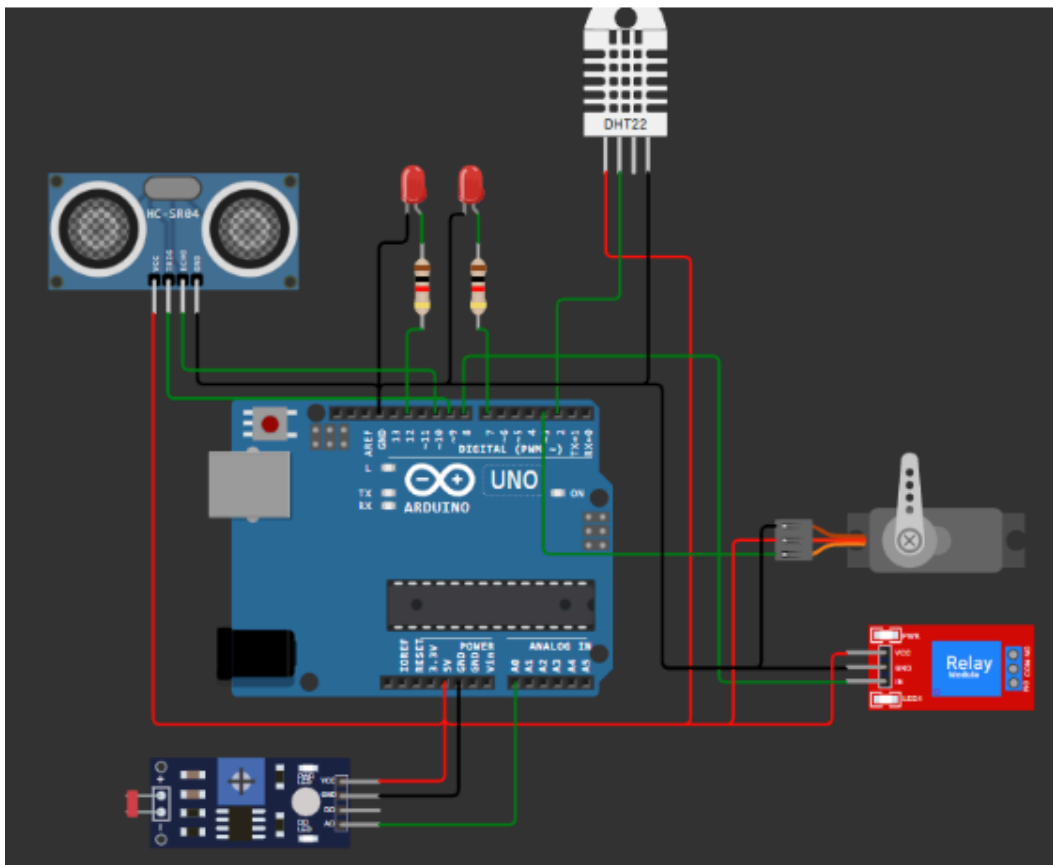
Subsistema electrónico: compuesto por todos los componentes eléctricos y electrónicos que permiten la captura de datos y la activación de los actuadores. Incluye sensores como el LDR, DHT11, y sensores de ultrasonido o infrarrojos para el control de acceso. Los actuadores principales son LEDs que simulan el sistema de iluminación, ventiladores para la climatización automática y un servomotor que controla la apertura de la barrera. El Arduino UNO actúa como el núcleo electrónico que interconecta todos estos dispositivos mediante señales digitales y analógicas.

Subsistema mecánico: comprende los elementos físicos o estructurales que sostienen y permiten el funcionamiento del sistema. Incluye el prototipo del centro comercial, que sirve como base para montar los sensores, actuadores y el cableado. También forman parte las estructuras móviles, como la barrera accionada por el servomotor, y los mecanismos de unión (soporte, tornillos, bisagras, pegamento, silicona, etc.) que aseguran los componentes. Aunque el prototipo no posee grandes mecanismos, este subsistema garantiza la estabilidad, protección y disposición física adecuada de los elementos electrónicos.

Subsistema informático: compuesto por el programa cargado en el microcontrolador Arduino. Su función es procesar la información proveniente de los sensores de luz, temperatura y distancia, comparar los valores obtenidos con los umbrales definidos y enviar las señales de control a los actuadores (LEDs, relé y servomotor). De esta manera, coordina el funcionamiento automático del sistema.

6. Diagrama de Conexionado

Nota: Utilizamos Wokwi para el diagrama de conexionado, esta página no cuenta con el sensor DHT11 así que el DHT22 cumple la función de este sensor. Tampoco contamos con el módulo LDR así que utilizamos únicamente una LDR común y una resistencia de 10k conectados como se muestra en las siguientes imágenes:



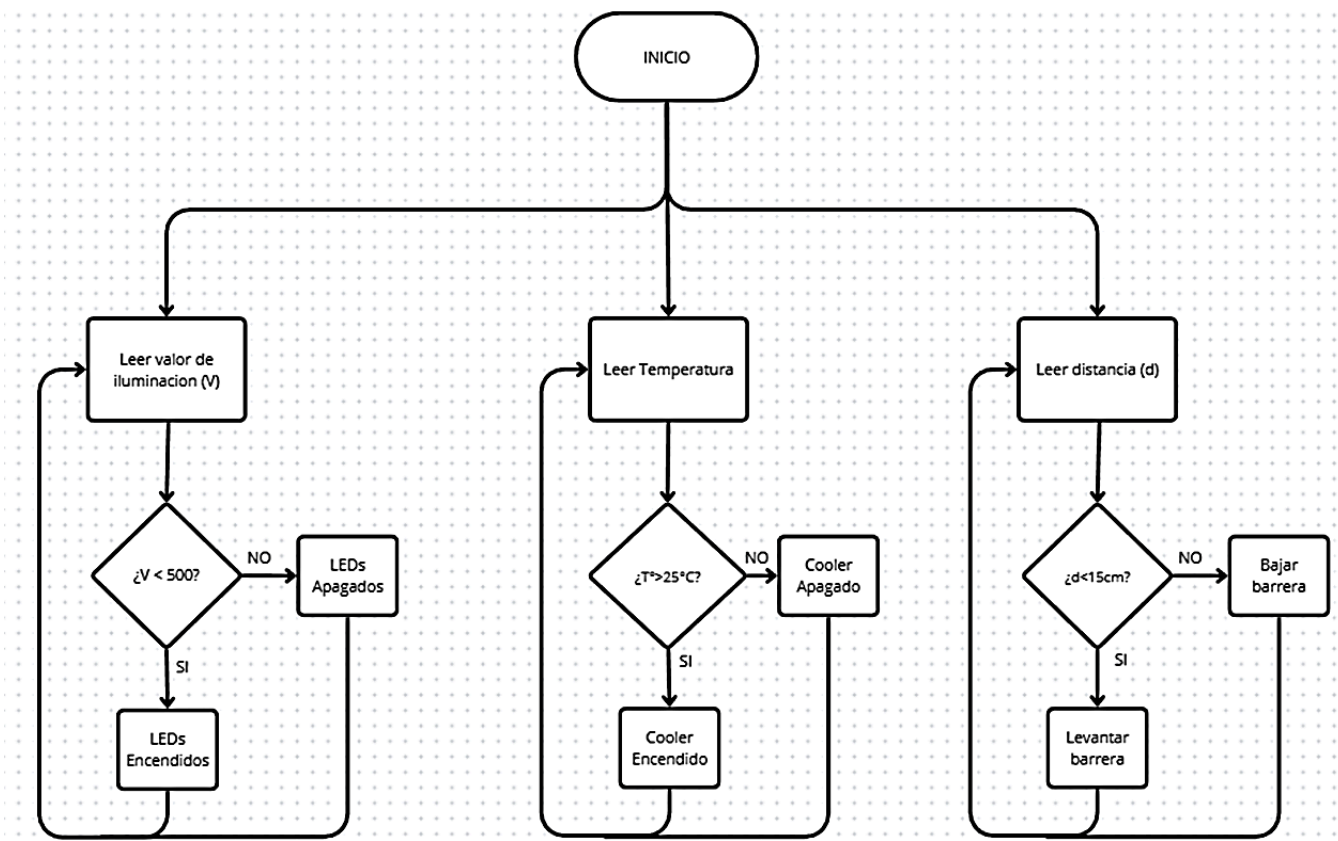
El diagrama muestra la conexión general de los componentes electrónicos que conforman el sistema domótico del proyecto. La placa Arduino UNO funciona como unidad central de control, encargada de recibir las señales de los sensores y enviar las órdenes correspondientes a los actuadores.

Todos los módulos del sistema comparten una alimentación común de 5V y una conexión a tierra (GND), lo que garantiza un funcionamiento estable y una correcta referencia eléctrica entre los dispositivos.

En el proyecto, los componentes se ubicaron de la manera más conveniente según su función: el sensor ultrasónico junto con el sistema de barrera y el servomotor (para detectar el vehículo), y los sensores ambientales (DHT11 y LDR) en zonas donde puedan tomar lecturas precisas del entorno.

7. Diagrama de flujo

En primer lugar, el sistema lee el valor de iluminación; si este es menor a 500, enciende los LEDs, y si es mayor o igual, los apaga. Luego, mide la temperatura; si supera los 25 °C, enciende el cooler, y si no, lo apaga. Por último, mide la distancia; si es menor a 15 cm, levanta la barrera, y si no, la baja. Una vez realizadas estas acciones, el proceso se repite continuamente desde el inicio.



8. Documentación código

```

37 void loop() {
38 //
39 //RELÉ SENSOR ULTRASONICO
40 delay(50);
41 long duracion;
42 int distancia;
43 // Enviar pulso ultrasónico
44 digitalWrite(TRIG, LOW);
45 delayMicroseconds(2);
46 digitalWrite(TRIG, HIGH);
47 delayMicroseconds(10);
48 digitalWrite(TRIG, LOW);
49 // Medir tiempo de retorno
50 duracion = pulseIn(ECHO, HIGH);
51 //Serial.println(duracion);
52 distancia = duracion * 0.0332 / 2; // cm
53 //Serial.print("Distancia: ");
54 //Serial.print(distancia);
55 //Serial.println(" cm");
56 // Si un objeto está cerca (menos de 15 cm), mover el servo
57 if (distancia < 15) {
58     servo.write(90); // mover a 90 grados
59     delay(2000);
60 } else {
61     servo.write(0); // volver a 0 grados
62 }

```

Se definen las variables **duración** y **distancia** para calcular el tiempo de ida y vuelta del pulso ultrasónico

Línea 52: fórmula para determinar la distancia al objeto

Condición: si la distancia es menor a 15 cm, se levanta la barrera (90°), sino, vuelve a su posición inicial (0°) o permanece cerrada

Se pueden descomentar las líneas del monitor serial incluidas en el código.

```

//ILUMINACION
//Lectura de Nivel de Iluminacion
valorLuz = analogRead(sensorLuz);
//Serial.print("Nivel de luz: ");
//Serial.println(valorLuz);
//Control de Iluminacion
if (valorLuz < 500){
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
}else{
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
}

```

La LDR entrega un valor numérico según la intensidad de luz que recibe: el valor aumenta con más luz y disminuye con menos.

Si la luminosidad es menor a 500, los LEDs se encienden; si es mayor, se apagan o permanecen apagados.

```
//TEMPERATURA
//Lecturas de temperatura
if (ultima_medicion_dht + 2000 < millis()) // Tiempo recomendado entre lecturas
{
    ultima_medicion_dht = millis();
    float t = dht.readTemperature(); // Lee temperatura
    Serial.print("Temperatura: ");
    Serial.print(t);
    Serial.println(" °C");
    // Control del relé
    if (t > TEMP_UMBRAL) {
        digitalWrite(RELE_PIN, HIGH); // Activa relé
        //Serial.println("Relé ACTIVADO");
    } else {
        digitalWrite(RELE_PIN, LOW); // Desactiva relé
        //Serial.println("Relé DESACTIVADO");
    }
}
```

Se incorpora un delay de 2000 ms, necesario para que el sensor DHT11 pueda actualizar correctamente las mediciones.

El sistema lee la temperatura ambiente y la compara con el umbral definido de 25 °C. Si se supera ese valor, el relé se activa, encendiendo el sistema de climatización (cooler). En caso contrario, el relé se apaga.

9. Dificultades, desafíos y simplificaciones realizadas

Se nos presentaron diversos inconvenientes, principalmente relacionados con el sensor ultrasónico HC-SR04. En un principio, surgieron problemas de falsos contactos en las conexiones, lo que ocasionó lecturas erráticas o ausencia de mediciones. Una vez verificada la correcta conexión y fijación de los cables, el sensor comenzó a funcionar de forma más estable.

Sin embargo, también se detectaron problemas en la programación. Inicialmente, el programa incluía un delay(2000) dentro del bucle principal, necesario para el correcto funcionamiento del sensor de temperatura DHT11, ya que este requiere al menos dos segundos entre lecturas consecutivas.

El inconveniente fue que dicho retardo afectaba el comportamiento del sensor ultrasónico, haciendo que la detección de objetos fuera demasiado lenta y la barrera no respondiera en tiempo real.

Para solucionar este conflicto, se decidió separar los tiempos de lectura de cada sensor, utilizando la función millis() para controlar de forma independiente el intervalo de medición del DHT11, mientras que al sensor ultrasónico se le asignó un retardo mucho menor (aproximadamente 50 ms).

Con esta modificación, el sistema logró un funcionamiento equilibrado, permitiendo que el sensor ultrasónico actuara con la frecuencia adecuada sin afectar el desempeño del DHT11 ni del relé de ventilación.

Tuvimos la idea de utilizar dos placas Arduino que se comunicaran entre sí a través del monitor serial. El objetivo era que el segundo Arduino controlara un display para mostrar los valores medidos del entorno. Sin embargo, no pudimos llevar a cabo esta implementación debido a la falta de un segundo Arduino Uno. Contábamos con un Arduino Uno y un módulo ESP8266, los cuales operan a diferentes tensiones (5V y 3.3 V, respectivamente). Esto representaba un riesgo de dañar el ESP8266 al compartir la alimentación de forma incorrecta, por lo que decidimos descartar esta parte del proyecto.

10. Conclusiones, aprendizajes y trabajos futuros

El proyecto permitió comprobar la viabilidad de implementar sistemas domóticos en espacios públicos utilizando componentes accesibles y de bajo costo, como el microcontrolador Arduino y distintos sensores. Se integraron varias funciones —control de iluminación, monitoreo de temperatura y apertura automatizada de acceso— dentro de un mismo sistema, demostrando cómo la tecnología puede mejorar el confort, la eficiencia energética y la seguridad en entornos comunitarios.

Durante el desarrollo, el grupo adquirió una comprensión más profunda sobre el funcionamiento de los sensores y actuadores, la importancia del control lógico, y fortaleció habilidades de programación en Arduino, interpretación de diagramas eléctricos, trabajo colaborativo y planificación.

Como trabajos futuros podríamos mejorar el mismo proyecto, por ejemplo incorporando comunicación inalámbrica (Bluetooth, Wi-Fi o IoT) para monitorear y controlar las funciones a distancia mediante una aplicación o plataforma web. También podría añadirse una pantalla LCD/OLED para visualizar datos en tiempo real. A largo plazo, el sistema podría ampliarse y automatizar más funciones, avanzando hacia un modelo de espacio público inteligente y sostenible.

También queremos destacar que la realización de este Taller nos permitió vivir una experiencia práctica que reafirmó nuestra elección por la carrera de Ingeniería Mecatrónica.

ANEXO

Repositorio en GitHub:

<https://github.com/Gera784/UrbanDomo>

