



Primer Parcial

Prácticas con el Laboratorio Remoto IoT y simulación Wokwi

Materia: Robótica

Alumna: Valentina Morana

Universidad: Universidad Abierta Interamericana

Profesores:

- Mg. Ing. Néstor Adrián Balich
- Ing. Franco Adrian Balich

Fecha: 22/11/2025

Año: 2025

Índice

1. Objetivos Generales.....	3
2. Estructura del Parcial.....	3
3. Criterios de Evaluación.....	3
Parte 1: Preguntas Teóricas.....	4
Parte 2: Ejercicios Prácticos.....	5
2.1 Ejercicio 1 – Simulación en Wokwi.....	5
Proyecto 1 - Led.....	6
Proyecto 2 - LDR.....	7
2.2 Ejercicio 2 – Laboratorio Remoto IoT.....	8
Proyecto 3 - Leds + Lab remoto.....	9
Proyecto 4 - LDR + Lab remoto.....	11
Proyecto 5 – Aplicación IoT propia: Light Show Interactivo.....	12
4. Conclusión.....	14

1. Objetivos Generales

- Integrar conocimientos teóricos y prácticos adquiridos durante la cursada.
- Comprender el funcionamiento básico de dispositivos IoT.
- Desarrollar algoritmos simples para el control de placas IoT reales mediante el laboratorio remoto.
- Aprender a utilizar el simulador Wokwi para validar prototipos IoT.
- Interpretar datos capturados de sensores y tomar decisiones sobre actuadores.

2. Estructura del Parcial

El examen consta de dos partes:

1. Parte Teórica – Responder preguntas sobre IoT.
2. Parte Práctica – Realizar ejercicios en el simulador Wokwi y en el Laboratorio Remoto IoT.

El trabajo deberá entregarse en un documento PDF, con capturas, código y respuestas justificadas.

3. Criterios de Evaluación

Criterio	Puntos
Respuestas teóricas	30%
Simulación en Wokwi	30%
Práctica en Lab Remoto	30%
Claridad, capturas, orden y presentación	10%

Total: 100%

Parte 1: Preguntas Teóricas

1. ¿Qué es IoT y cuál es su objetivo principal?
2. ¿Qué diferencia hay entre un sensor y un actuador? Dé un ejemplo de cada uno.
3. ¿Qué es MQTT y por qué se utiliza en IoT?
4. ¿Qué es un "gemelo digital" en sistemas ciber-físicos?
5. ¿Qué ventajas tiene usar simuladores como Wokwi en proyectos IoT?

1. El Internet of Things (IoT) es un paradigma tecnológico que conecta objetos físicos a internet para recolectar, procesar y compartir datos en tiempo real. Su objetivo principal es automatizar procesos, optimizar recursos y mejorar la toma de decisiones mediante la comunicación entre dispositivos inteligentes.

Ejemplos: domótica, monitoreo ambiental o gestión de flotas.

2. Un sensor capta información del entorno físico (como temperatura, luz o sonido) y la convierte en datos digitales. Un actuador, en cambio, ejecuta una acción física en respuesta a una señal recibida del sistema.

Ejemplo de sensor: DHT11 (temperatura y humedad).

Ejemplo de actuador: servomotor que abre una válvula o mueve un brazo robótico.

3. MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero basado en el modelo *publicador-suscriptor*, diseñado para comunicaciones eficientes entre dispositivos con bajo ancho de banda o energía. Se utiliza en IoT porque reduce el consumo de recursos y facilita la comunicación asíncrona entre sensores, gateways y servidores.

4. Un gemelo digital es una réplica virtual de un sistema físico que permite simular, monitorear y predecir su comportamiento en tiempo real. En sistemas ciber-físicos, integra datos de sensores del mundo real con modelos computacionales, mejorando el control, el mantenimiento predictivo y la optimización de procesos industriales o robóticos.
 5. Simuladores como Wokwi permiten diseñar, probar y depurar circuitos IoT sin necesidad de hardware físico. Facilitan la validación temprana de código, reducen costos de prototipado y permiten experimentar con múltiples configuraciones en entornos seguros. Además, favorecen el aprendizaje y la colaboración entre equipos de desarrollo remoto.
-

Parte 2: Ejercicios Prácticos

2.1 Ejercicio 1 – Simulación en Wokwi

Objetivo: Conectar un LED y controlarlo desde un programa simple.

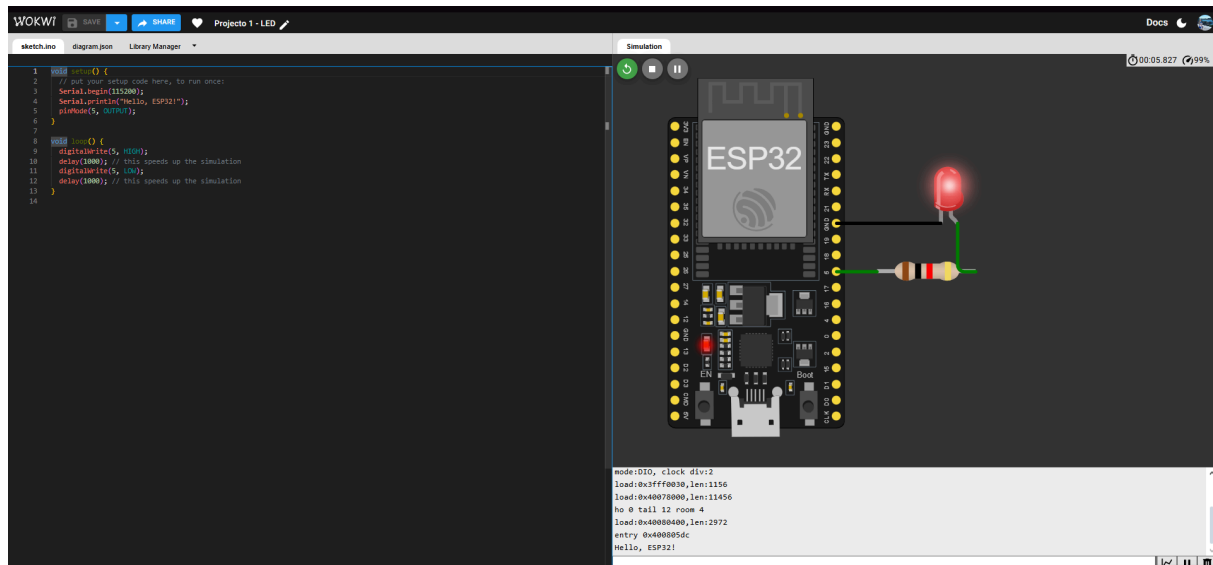
Consigna:

- Crear un proyecto en Wokwi con una placa ESP32 o Arduino.
- Conectar un LED y encenderlo alternando estados cada 1 segundo.
- Incluir el código y captura del simulador funcionando.

Proyecto 1 - Led

<https://wokwi.com/projects/443893796313476097> - Wokwii

https://drive.google.com/file/d/1YRJ2f6E0K98dhXW2711ikQ2_iHe11AZE/view?usp=sharing - Video



Código

```
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(115200);  
    Serial.println("Hello, ESP32!");  
    pinMode(5, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(5, HIGH);  
    delay(1000); // this speeds up the simulation  
    digitalWrite(5, LOW);  
    delay(1000); // this speeds up the simulation  
}
```

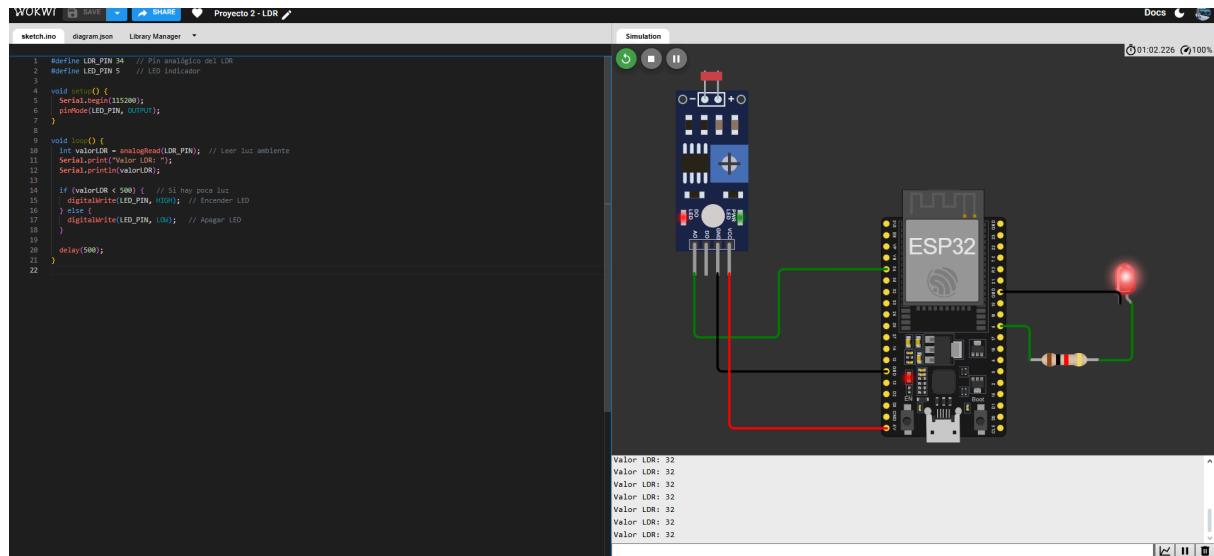
Es el “**Hola Mundo**” del **hardware IoT**. Permite verificar el correcto funcionamiento de la placa ESP32 y comprender el manejo básico de salidas digitales.

El programa enciende y apaga el LED conectado al pin 5 cada 1 segundo, generando un parpadeo constante que confirma la ejecución del bucle principal.

Proyecto 2 - LDR

<https://wokwi.com/projects/443889721786766337> - Wokwi

<https://drive.google.com/file/d/1zO-uYIlogBGXIMGaUqKN8jBISl8gAPYS/view?usp=sharing> - Video



Código:

```
#define LDR_PIN 34 // Pin analógico del LDR
#define LED_PIN 5 // LED indicador

void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  int valorLDR = analogRead(LDR_PIN); // Leer luz ambiente
  Serial.print("Valor LDR: ");
  Serial.println(valorLDR);

  if (valorLDR < 500) { // Si hay poca luz
    digitalWrite(LED_PIN, HIGH); // Encender LED
  } else {
    digitalWrite(LED_PIN, LOW); // Apagar LED
  }

  delay(500);
}
```

Este proyecto demuestra el control básico entre sensor y actuador en un entorno IoT:

Un sensor LDR convierte una magnitud física (luz) en datos, y un actuador (LED) responde según una condición lógica programada.

El sistema podría extenderse para automatizar luminarias o integrarse en una red IoT para monitoreo ambiental remoto.

2.2 Ejercicio 2 – Laboratorio Remoto IoT

Objetivo: Controlar un actuador real y usar un sensor remoto.

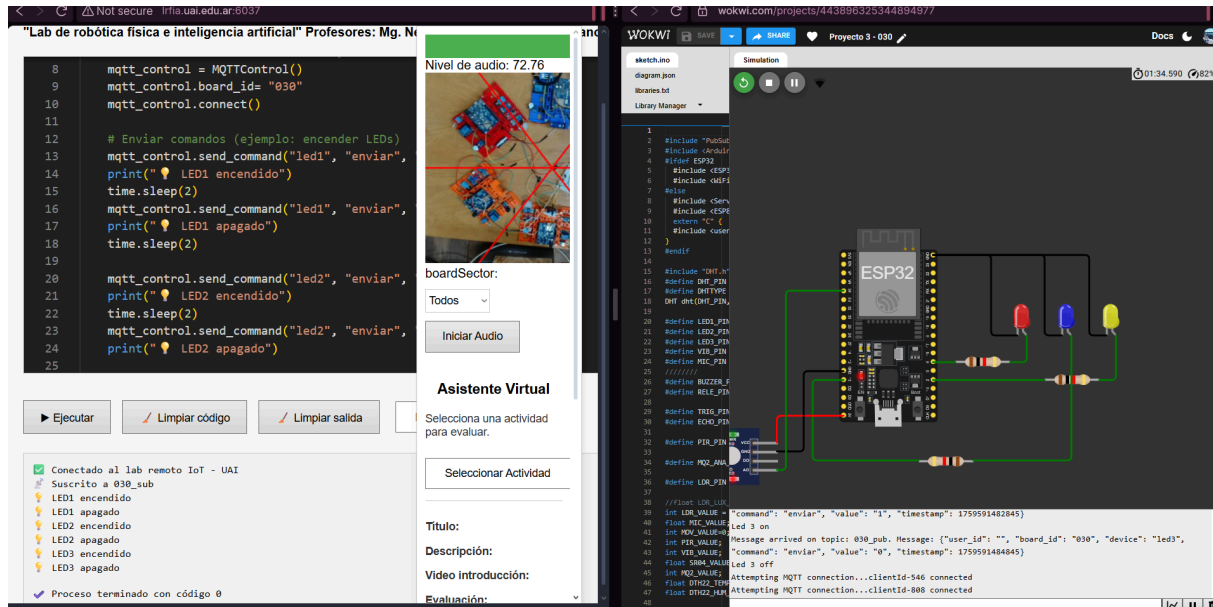
Consigna:

- Conectarse al Lab Remoto IoT.
- Encender y apagar un LED real desde la interfaz.
- Capturar valores de un sensor (ej.: temperatura, luz, humedad).
- Crear un pequeño algoritmo que:
 - Lea el sensor,
 - Compare el valor,
 - Active o desactive un actuador según la condición (ej.: encender LED si la luz < umbral).

Proyecto 3 - Leds + Lab remoto

<https://wokwi.com/projects/443896325344894977> - Wokwi

https://drive.google.com/file/d/1h0QNkJzASvP1jJRscNEiT0_hJ6eKa-B8/view?usp=sharing - Video



Código Lab Remoto:

```
# -*- coding: utf-8 -*-
from lrfia_mqtt_control import MQTTControl # Importa la clase
import time

# === Programa principal ===
if __name__ == "__main__":
    # Crear la instancia de la clase y conectarse al broker
    mqtt_control = MQTTControl()
    mqtt_control.board_id= "030"
    mqtt_control.connect()

    # Enviar comandos (ejemplo: encender LEDs)
    mqtt_control.send_command("led1", "enviar", "1")
    print("💡 LED1 encendido")
    time.sleep(2)
    mqtt_control.send_command("led1", "enviar", "0")
    print("💡 LED1 apagado")
```

```
time.sleep(2)

mqtt_control.send_command("led2", "enviar", "1")
print("💡 LED2 encendido")
time.sleep(2)
mqtt_control.send_command("led2", "enviar", "0")
print("💡 LED2 apagado")

mqtt_control.send_command("led3", "enviar", "1")
print("💡 LED3 encendido")
time.sleep(2)
mqtt_control.send_command("led3", "enviar", "0")
print("💡 LED3 apagado")

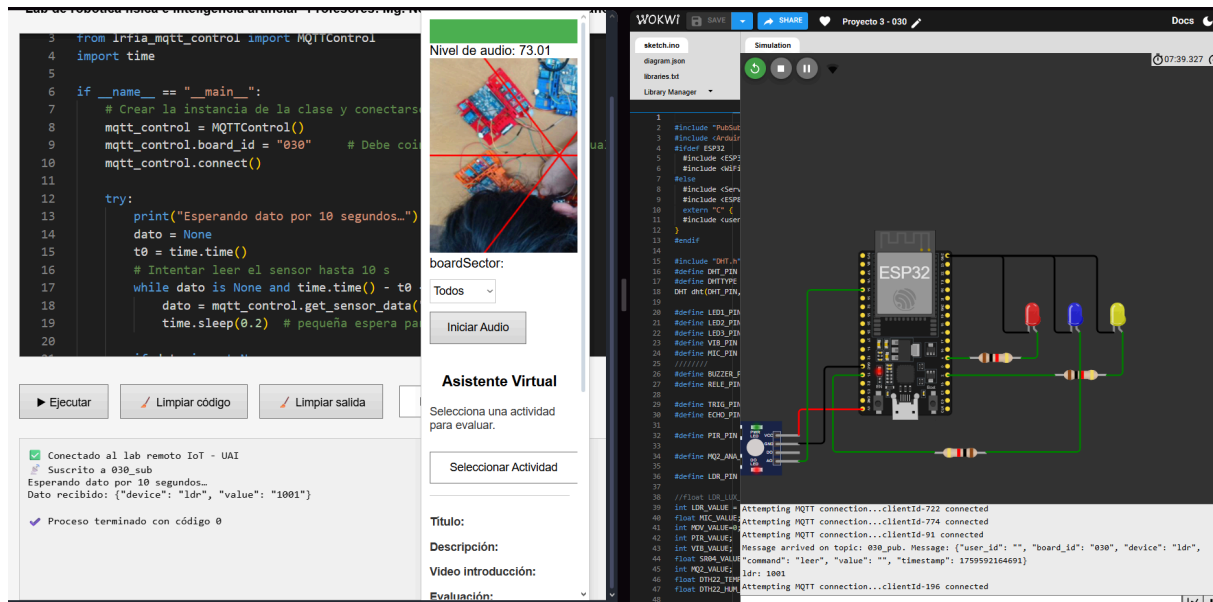
# Finalizar
mqtt_control.stop()
```

El programa se conecta al Laboratorio Remoto IoT mediante la librería `Irfa_mqtt_control`, estableciendo comunicación con la placa física identificada por el `board_id` "030". A través del protocolo MQTT, se envían comandos para encender y apagar tres LEDs reales (`led1`, `led2`, `led3`) con intervalos de dos segundos entre cada acción. Cada instrucción se transmite al broker remoto, que las interpreta y ejecuta sobre el hardware en tiempo real. El código demuestra el control remoto de actuadores mediante mensajería IoT, validando la interacción entre software Python y dispositivos físicos conectados a la red.

Proyecto 4 - LDR + Lab remoto

<https://wokwi.com/projects/443896325344894977> - Wokwi

<https://drive.google.com/file/d/1KonBXDVEAHQY4dYODou8Q1ukNKW42MWt/view?usp=sharing> - Video



Código Lab Remoto:

```
# coding: utf-8

from lrfa_mqtt_control import MQTTControl
import time

if __name__ == "__main__":
    # Crear la instancia de la clase y conectarse al broker
    mqtt_control = MQTTControl()
    mqtt_control.board_id = "030"
    mqtt_control.connect()

    try:
        print("Esperando dato por 10 segundos...")
        dato = None
        t0 = time.time()

        while dato is None and time.time() - t0 < 10:
            dato = mqtt_control.get_sensor_data("ldr")
```

```
        time.sleep(0.2)

    if dato is not None:
        print(f"Dato recibido: {dato}")
    else:
        print("No llegó ningún dato en 10 s (revisá board_id y conexión).")
    finally:
        mqtt_control.stop()
```

Se usa para interactuar con el Laboratorio Remoto IoT de la UAI, utilizando la librería `Irfa_mqtt_control`, que abstrae toda la lógica de conexión al broker MQTT del laboratorio remoto.

El código se conecta al servidor remoto, solicita la lectura de un sensor (en este caso un LDR), espera la respuesta y la muestra por consola.

Proyecto 5 – Aplicación IoT propia: Light Show Interactivo

Además desarrollé un proyecto propio de aplicación IoT, basado en los mismos principios de interacción entre sensores, procesamiento lógico y actuadores físicos.

Si bien no se ejecuta dentro del entorno del Laboratorio Remoto IoT, responde a los mismos objetivos, implementando el ciclo completo de detección, análisis y respuesta automática, en cumplimiento con los requisitos del Ejercicio 2.

El sistema integra un módulo Arduino UNO, una tira LED WS2812B y un sensor de sonido KY-037. El algoritmo analiza la intensidad del audio en tiempo real y activa distintos patrones de iluminación según el nivel de volumen detectado.

El control se complementa con una aplicación móvil (LEDController) desarrollada en Android Studio, que permite seleccionar modos de animación, color y brillo mediante conectividad Bluetooth.

Repositorio del código: https://github.com/valentinamorana/robotica_proyecto

Armado del Robot:

<https://www.youtube.com/watch?v=arAJ2jzalk4&list=PLV7U1PnDNScc9Ildke7sm5rW4sVpfUwcc&pp=gAQB>

Diseño de la aplicación:



4. Conclusión

El desarrollo de los proyectos permitió integrar los conocimientos teóricos y prácticos del IoT mediante el uso combinado de simuladores y entornos reales. A través de Wokwi se validaron los circuitos y algoritmos de control, mientras que el Laboratorio Remoto IoT posibilitó la interacción directa con sensores y actuadores físicos mediante el protocolo MQTT. Se logró demostrar el ciclo completo de un sistema IoT, lectura de sensor, procesamiento lógico y acción sobre un actuador, validando el funcionamiento de la comunicación entre hardware, software y red. En conjunto, las prácticas evidencian la aplicación concreta de la automatización y el control remoto en entornos IoT.