



## Tecnicatura Superior en Telecomunicaciones

### Proyecto IoT: ILUMINET

#### Integrantes:

- MÁRQUEZ José Luis
- PAEZ Tiziano
- GONZÁLEZ A. Juan Diego
- CARBALLO Macarena
- GUZMÁN Lilen
- PANTOJA Paola

#### Docente:

- GONZÁLEZ Mario



## MANUAL DEL USO

### Índice

1. INTRODUCCIÓN.....	3
2. COMPONENTES DEL SISTEMA .....	3
3. REQUISITOS PREVIOS .....	4
4. ENCENDIDO Y FUNCIONAMIENTO .....	5
5. VISUALIZACIÓN DE DATOS .....	6
6. CONTROL REMOTO .....	6
7. SOLUCIÓN DE PROBLEMAS .....	6
8. RECOMENDACIONES .....	7
9. CONTACTO Y SOPORTE .....	8

## 1. INTRODUCCIÓN

**ILUMINET** es un sistema de alumbrado público inteligente basado en **Internet de las Cosas (IoT)**.

Su objetivo es optimizar el consumo energético, mejorar la seguridad urbana y permitir la **monitorización remota** del estado de las luminarias mediante sensores, microcontroladores y plataformas de visualización.

Cada luminaria está equipada con sensores que detectan **presencia, nivel de luz ambiental y consumo energético**, enviando los datos a una plataforma central para su análisis.

El sistema está pensado para integrarse en entornos de **Smart Cities**, mejorando la eficiencia, la sustentabilidad y la calidad de vida de la comunidad.

## 2. COMPONENTES DEL SISTEMA

### Hardware principal

- ESP8266: Controla sensores individuales de cada luminaria.
- ESP32: Gateway central que recopila datos vía RF 433 MHz y los envía al servidor IoT.
- Sensores:
  - PIR / Microondas: detección de movimiento.
  - LDR o fotodiodo: mide luz ambiental.
  - ACS712: mide consumo energético.
- Módulo RF 433 MHz: comunicación entre nodos y gateway.
- Módulo SIM800L: conexión 2G/GPRS a internet.
- Luminarias LED: 50 – 100 W (según entorno).

### Software y servicios

- Mosquitto (MQTT): comunicación entre nodos y servidor.
- Node-RED: procesamiento y automatización de flujos de datos.
- InfluxDB: almacenamiento de datos históricos.
- Grafana: visualización de dashboards en tiempo real.
- Nginx: proxy inverso para acceso web seguro.
- Docker / Portainer: despliegue y administración de contenedores.

## **3. REQUISITOS PREVIOS**

### Hardware

- Fuente de alimentación estable para luminarias y ESP32.
- Conexión a red Wi-Fi o datos móviles (según versión).
- Cableado y conectores adecuados (IP65 si se colocan al exterior).
- PC o servidor con Docker instalado (para los servicios Node-RED, Grafana, InfluxDB, Mosquitto).

### Software

- Arduino IDE (para cargar código a los microcontroladores).
- Docker Desktop / Portainer instalado.
- Servicios activos: Mosquitto, Node-RED, InfluxDB, Grafana.
- Acceso al repositorio:

#### 4. ENCENDIDO Y FUNCIONAMIENTO

Paso 1 – Carga del código:

- Abre el archivo .ino correspondiente al nodo o gateway.
- Configura las credenciales de Wi-Fi o APN (si usa SIM800L).
- Compila y carga el código al microcontrolador mediante Arduino IDE.

Paso 2 – Conexión de sensores:

- PIR → Pin digital (detección de movimiento).
- LDR → Entrada analógica.
- ACS712 → Pin analógico (medición de corriente).
- RF433 → Puerto TX/RX.

Paso 3 – Alimentación:

- Enciende el sistema y verifica que los LEDs indicadores del ESP parpadeen.
- Los datos comenzarán a transmitirse al broker MQTT.

Paso 4 – Comprobación de comunicación:

- En Node-RED, verifica la recepción de mensajes desde el topic `iluminet/sensores/+/datos`.
- Confirma que los dashboards de Grafana muestran información.

## 5. VISUALIZACIÓN DE DATOS

Node-RED:

- Accedé desde el navegador a <http://localhost:1880> (o IP del servidor).
- Verificá el flujo de recepción y almacenamiento de datos.

Grafana:

- Ingresá al panel de Grafana (<http://localhost:3000>).
- Visualizá los dashboards configurados:
  - Estado de luminarias (encendida/apagada).
  - Nivel de luminosidad.
  - Consumo energético.
  - Detección de presencia.

## 6. CONTROL REMOTO

Desde Node-RED se pueden realizar controles manuales o automatizados:

- Encendido / apagado de luminarias.
- Ajuste de intensidad según luz ambiental o presencia.
- Programación horaria de funcionamiento.
- Alertas automáticas (Telegram o email).

Ejemplo de topic MQTT para control:

```
iluminet/control/luminaria1 { "estado": "on", "intensidad": 80 }
```

## 7. SOLUCIÓN DE PROBLEMAS

Problema	Posible causa	Solución
No se muestran datos en Grafana	Error de conexión entre MQTT y Node-RED	Verificar configuración de topics y flujo
El ESP32 no conecta a la red	Credenciales Wi-Fi o APN incorrectas	Revisar configuración en el código
Sensores no responden	Cableado incorrecto o fallo del sensor	Verificar conexiones o reemplazar
Node-RED no se inicia	Contenedor detenido	Revisar estado de contenedores en Portainer
No se enciende una luminaria	Relé o salida del ESP defectuosa	Comprobar hardware y código de salida

## 8. RECOMENDACIONES

- Usar nombres únicos de topics MQTT para cada luminaria.
- Proteger físicamente los sensores (IP65 o superior).
- Realizar backups periódicos de la base de datos InfluxDB.
- Revisar las conexiones eléctricas cada 6 meses.
- Actualizar el firmware del ESP32 o ESP8266 ante mejoras de seguridad.
- Mantener Grafana y Node-RED actualizados.

## 9. CONTACTO Y SOPORTE

Equipo de desarrollo ILUMINET – ISPC

Integrantes: José Luis Márquez, Tiziano Páez, Juan Diego González, Macarena Carballo, Lilien Guzmán, Paola Pantoja.

Repositorio oficial:

<https://github.com/ISPC-PI-II-2024/Desarrollo-de-Aplicaciones-IOT---SMARTCITY--->