

# Sistema Gateway IoT

🔗 Documento: Sistema Gateway ESP32 - Comunicación WiFi + MQTT + LoRa

## 📘 1. Introducción al Sistema Gateway

### 1.1 Propósito del Sistema

El Gateway ESP32 actúa como el centro de operaciones del ecosistema IoT, funcionando como un puente inteligente entre los sensores remotos (comunicación LoRa) y la plataforma cloud (comunicación WiFi/MQTT). Su función principal es coordinar, procesar y transmitir datos de múltiples fuentes hacia el backend para su almacenamiento y análisis.

### 1.2 Arquitectura General

El sistema sigue una arquitectura de cuatro capas:

Capa de Sensores: Dispositivos ESP8266/ESP32 con módulos LoRa

Capa de Gateway: ESP32 con capacidades multi-protocolo

Capa de Comunicación: WiFi + MQTT sobre WebSocket

Capa de Aplicación: Dashboard web y aplicaciones móviles

### 1.3 Flujo de Datos

Los datos viajan en una secuencia definida:

text

Sensores LoRa → Gateway ESP32 → Procesamiento → MQTT/WebSocket → Plataforma Cloud → Dashboard Web

## 🔌 2. Componentes del Gateway ESP32

### 2.1 Hardware Principal

Microcontrolador: ESP32 con dual-core y WiFi integrado

Módulo LoRa: RA-02 433MHz para comunicación de larga distancia

Interfaz Visual: Pantalla LCD 20x4 o 16x2 con controlador I2C

Entrada de Usuario: Botón físico para reset de configuración

Alimentación: Sistema multi-voltage con reguladores step-up/step-down

## 2.2 Módulos de Comunicación

WiFi 2.4GHz: Para conexión a redes locales e Internet

LoRa 433MHz: Para comunicación con sensores remotos

WebSocket: Para conexión persistente con broker MQTT

Protocolo MQTT: Para publicación/suscripción de datos

## 2.3 Alimentación y Energía

El sistema incorpora múltiples reguladores de voltaje:

3.3V para ESP32 y módulo LoRa

5V para periféricos como el LCD

4.1V específico para módulos GSM (cuando se utilizan)

Gestión de batería con protección TP4056 para operación autónoma

## 3. Comunicación LoRa con Sensores

### 3.1 Protocolo LoRa Implementado

El gateway utiliza LoRa en banda de 433MHz con configuración optimizada para balance entre alcance y consumo energético. Los parámetros técnicos incluyen:

Spreading Factor 7 para velocidad de transmisión equilibrada

Ancho de banda 125kHz para mejor inmunidad al ruido

CRC habilitado para verificación de integridad de datos

Palabra de sincronización personalizada para evitar interferencias

### 3.2 Descubrimiento de Endpoints

El gateway implementa un sistema de descubrimiento automático de sensores:

Broadcast periódico para identificar endpoints disponibles

Registro dinámico de dispositivos conectados

Monitoreo de estado con detección de desconexiones

Límite configurable de hasta 10 endpoints simultáneos

### 3.3 Polling de Datos

El sistema realiza sondeos secuenciales a cada endpoint:

Intervalo configurable (normalmente 45 segundos)

Solicitud individual por cada sensor

Reintentos automáticos en caso de timeout

Procesamiento de respuestas con validación de CRC

### 3.4 Estructura de Datos LoRa

Los mensajes siguen un formato textual delimitado por pipes para eficiencia:

Comandos de control: "LIST\_ENDPOINTS", "GET\_DATA:E01"

Respuestas estructuradas: "DATA:E01|BAT:85|TEMP:23.5|HUM:65"

Metadatos de calidad: RSSI, SNR, y estado de batería

## 🌐 4. Comunicación WiFi y MQTT

### 4.1 Gestión de Conexiones WiFi

El gateway implementa múltiples estrategias de conectividad:

Modo Cliente WiFi

Configuración persistente en memoria no volátil

Reconexión automática con intervalos inteligentes

Monitoreo de calidad de señal con métricas RSSI

Fallback a modo AP cuando no hay redes disponibles

Modo Punto de Acceso (AP)

Activación automática cuando no hay configuración WiFi

Portal web embebido para configuración inicial

Timeout de seguridad (3 minutos) para evitar AP permanente

Interfaz responsive accesible desde cualquier dispositivo

### 4.2 Protocolo MQTT sobre WebSocket

La implementación utiliza MQTT sobre WebSocket para superar restricciones de firewall:

Conexión Persistente

WebSocket como transporte para MQTT

Reconexión automática cada 5 segundos en caso de falla  
Mecanismo de ping/pong para mantener conexión activa  
Handshake de protocolo específico para compatibilidad con brokers  
Cliente MQTT Personalizado  
Implementación manual del protocolo MQTT 3.1.1  
Soporte para QoS 0 con overhead mínimo  
Suscripciones dinámicas a tópicos de control  
Publicación eficiente con manejo de paquetes binarios

#### 4.3 Tópicos MQTT Implementados

El sistema publica en tres categorías principales de tópicos:

- Estado del Gateway
- Información de salud del sistema
  - Calidad de señal WiFi
  - Uptime y métricas de rendimiento
  - Estado de componentes internos
  - Estado de Endpoints
  - Lista de sensores conectados
  - Niveles de batería y estado de carga
  - Calidad de señal LoRa por endpoint
  - Metadatos de última comunicación
- Datos de Sensores
  - Mediciones de temperatura y humedad
  - Timestamps de captura
  - Identificadores únicos de sensor
  - Estados operativos y códigos de error

## 5. Interfaz de Usuario y Control

### 5.1 Pantalla LCD

La interfaz visual proporciona feedback inmediato del estado del sistema:

Estados de Visualización

Modo Normal: Información operativa básica

Modo Configuración: Asistente para setup inicial

Modo Reset: Confirmación y progreso de reset

Modo Error: Indicación de fallos específicos

Información Mostrada

Estado de conexión WiFi y MQTT

Cantidad de endpoints activos

Calidad de señal y métricas

Dirección IP y identificador del gateway

## 5.2 Botón Físico de Control

El botón de reset implementa lógica anti-accidental:

Mecanismo de Confirmación

Detección de presión prolongada (3 segundos)

Feedback visual progresivo en LCD

Cancelación automática si se suelta antes del timeout

Reset completo de configuración WiFi

## 5.3 Portal Web de Configuración

Interfaz accesible vía navegador web:

Características

Diseño responsive para móviles y desktop

Formulario simplificado para configuración WiFi

Guardado persistente en memoria flash

Reinicio automático para aplicar cambios

# ⚡ 6. Procesamiento de Datos

## 6.1 Recepción y Validación

Cada mensaje LoRa pasa por un proceso de validación:

Verificación de CRC para integridad de datos

Parsing estructurado de campos delimitados

Validación de rangos para valores de sensores

Timestamp automático de recepción

## 6.2 Transformación a JSON

Los datos se convierten a formato JSON estandarizado:

Estructura jerárquica para organización lógica

Metadatos enrichidos con información del gateway

Optimización de tamaño para transmisión eficiente

Compatibilidad universal con sistemas backend

## 6.3 Gestión de Estado

El gateway mantiene estado interno consistente:

Registro de endpoints activos e inactivos

Historial de comunicación con timestamps

Métricas de calidad de conexión

Estadísticas operativas para monitoreo

# 7. Estrategias de Robustez

## 7.1 Manejo de Errores

El sistema incorpora múltiples capas de tolerancia a fallas:

Recuperación Automática

Reconexión WiFi cada 10 segundos en caso de desconexión

Reconexión WebSocket cada 5 segundos

Reintentos de comandos LoRa con timeout configurable

Limpieza periódica de endpoints inactivos

Degradación Graceful

Operación offline temporal cuando no hay Internet

Buffer local de datos pendientes de transmisión

Priorización de datos críticos

Sincronización cuando se restablece la conexión

## 7.2 Monitoreo de Salud

Sistema integral de autodiagnóstico:

Verificación periódica de todos los componentes

Métricas de rendimiento en tiempo real

Detección temprana de problemas

Logs detallados para troubleshooting

## 7.3 Seguridad Implementada

Aunque básica, incluye medidas de protección:

Configuración aislada en modo AP

Validación de entrada en formularios web

Reset físico como mecanismo de recuperación

Comunicación LoRa con sync word personalizado

# 8. Métricas y Monitoreo

## 8.1 Datos Publicados

El gateway reporta métricas operativas esenciales:

Métricas de Comunicación

Intensidad de señal WiFi (RSSI)

Calidad de conexión categorizada

Tasa de éxito en transmisiones LoRa

Latencia de comunicación

Métricas de Sistema

Uptime y tiempo de operación

Uso de memoria heap disponible

Contador de reconexiones

Estado de componentes individuales

## 8.2 Alertas y Notificaciones

Sistema de detección de anomalías:

Endpoints no responden por tiempo prolongado

Calidad de señal degradada

Problemas de conectividad persistente

Errores críticos de hardware

## 9. Escalabilidad y Mantenibilidad

### 9.1 Diseño para Escalabilidad

Arquitectura modular para fácil extensión

Límites configurables de endpoints y sensores

Gestión eficiente de memoria y recursos

Interfaces bien definidas entre componentes

### 9.2 Facilidad de Mantenimiento

Código organizado en módulos especializados

Configuración centralizada en archivos de cabecera

Logs descriptivos para debugging

Documentación integrada en comentarios

## 10. Casos de Uso y Aplicaciones

### 10.1 Monitoreo Agrícola

Siros de granos con sensores de temperatura/humedad

Invernaderos con control ambiental

Campos extensivos con cobertura LoRa

### 10.2 Industria 4.0

Monitorización de maquinaria industrial

Control de condiciones ambientales en fábricas

Seguimiento de activos en movimiento

10.3 Smart Cities

Sensores ambientales urbanos

Monitorización de infraestructura pública

Sistemas de alerta temprana

## 11. Mejoras Futuras y Roadmap

11.1 Próximas Características

Integración GSM/LTE para backup de conectividad

Almacenamiento local en tarjeta SD

Protocolo HTTPS para comunicación segura

Actualizaciones OTA (Over-The-Air)

11.2 Optimizaciones Planificadas

Modo sueño profundo para ahorro energético

Compresión de datos para eficiencia en transmisión

Algoritmos de predicción para detección de tendencias

Interfaz gráfica avanzada con gráficos en tiempo real

## Conclusión

El Gateway ESP32 representa una solución completa y robusta para implementaciones IoT que requieren comunicación de larga distancia (LoRa) combinada con conectividad cloud (WiFi/MQTT). Su diseño modular, estrategias de robustez y interfaces de usuario múltiples lo convierten en una plataforma ideal para aplicaciones industriales, agrícolas y de smart city.

La arquitectura permite fácil personalización y extensión, mientras que las características de auto-recuperación aseguran operación continua incluso en entornos desafiantes. El sistema está preparado para escalar desde prototipos hasta implementaciones productivas a gran escala.