INSTITUTO FEDERAL
Bahia
Campus Santo Antônio de Jesus

Instituto Federal de Educação, Ciência e Tecnologia da Bahia

Sistema IOT de irrigação inteligente multirregional

Luis Guilherme Passos Ramos, Náthaly de Jesus Brito

INSTITUTO FEDERAL
Bahia
Campus Santo Antônio de Jesus

Agosto, 2025

INSTITUTO FEDERAL
Bahia
Campus Santo Antônio de Jesus

Projeto -

Infraestrutura IoT segura e interoperável para irrigação inteligente com sensores em zona remota (com uso de LoRaWAN para envio de dados de sensores) e zona urbana (com uso de MQTT local). Os dados vão para um servidor central que autentica dispositivos, criptografa transmissões com TLS/DTLS e garante privacidade.

Dispositivos simulados

- Sensor de zona remota via LoRaWAN - envia dados simulados de umidade e temperatura; Usa um client UDP para simular pacote LoRaWAN;
- Sensor de zona urbna via MQTT – publica dados em tópico MQTT;

Servidor central

- Recebe dados via HTTP REST e armazena;
- Usa JWT para autenticação;
- Encriptação via TLS com certificado local;
- Log dos eventos;

Segurança

- TLS/DTLS entre dispositivos e sevidor- o TLS protege o MQTT e HTTP;
- JWT para controle de acesso – garante autenticação forte para o envio de dados;
- Logs - permite rastreabilidade e auditoria;

Monitoramento

- Geração de logs de comunicação e autenticação;

Principais Componentes do sistema -

API REST (api_rest/app.py)

- O framework é Flask com JWT para autenticação;
- A função é receber dados dos sensores via gateway UDP, exigindo uma autenticação JWT, registrando logs de autenticação e comunicação;
- Para segurança temos HTTPS (TLS) com certificados próprios (certs/);
- Temos dois endpoints, /login para autenticar sensores retornando um token JWT, e /sensores que recebe dados dos sensores exigindo JWT e registrando logs;

Gateway UDP (udp_gateway/gateway_udp.py)

- A função é atuar como bridge entre os sensores remotos UDP e a API REST (HTTP);
- O fluxo acontece assim:
  Recebe pacotes UDP de sensores remotos.
  Registra cada papcote recebido em logs (logs/comms.log).
  Encapsula os dados em uma requisição HTTP POST para a API REST, usando JWT para autenticação.
- A segurança se dá através do HTTPS e JWT.

Sensor remoto UDP (udp_gateway/sensor-udp.py)

- Ele simula um sensor remoto, enviando, periodicamente, dados via UDP para o gateway;

Sensor Urbano MQTT (mqtt/sensor_mqtt.py)

- Ele simula um sensor urbano, publicando dados, periodicamente em um tópico MQTT seguro (TLS) garantindo segurança.

Logs e monitoramento (logs/)

- Auth.log: registra autenticações e tentativas de login;
- Comms.log: registra comunicações dos sensores;
- Packets.pcap: pode capturar pacotes de rede para análise;

Certificados (certs/)

Garantem criptografia de ponta a ponta;

Em resumo, o fluxo se dá pelo sensor remoto udp gerar dados e enviá-los via UDP para o gateway UDP. Já esse gateway os recebe e registra em log repassando os dados via HTTP POST com JWT para a API REST que valida o JWT, anonimiza o ID do sensor, registra logs e responde. O sensor Urbano MQTT publica os dados em um tópico MQTT seguro (TLS) simulando sensores urbanos.

Interoperabilidade

A interoperabilidade se dá através da comunicação e integração de dados pelo uso de protocolos distintos por sensores distintos de forma segura e transparente em uma única infraestrutura.

Sensores Urbanos usam MQTT (protocolo leve, bem comum em IoT, que é baseado em publicação e assinatura). Já os Sensores Remotos utilizam UDP (simulando LoRaWAN, que é comum em áreas remotas por possuir baixo consumo e eficiência).

O gateway UDP funciona como tradutor, ele recebe dados dos sensores remotos via UDP convertendo-os para o formato esperado pela API REST(HTTP e JSON) e autentica com JWT.

Assim, a interoperabilidade do nosso sistema é garantida através do uso do gateway que faz a ponte entre os diferentes protocolos que usamos, pela padronização dos dados e pela centralização do processamento na API REST.

INSTITUTO FEDERAL
Bahia
Campus Santo Antônio de Jesus

Diferença entre os protocolos e segurança utilizada

1. Protocolo

Protocolo	**Uso**	**Característica**
MQTT	Sensores urbanos	• Protocolo de mensagens publish/subscribe; • Transporte TCP com TLS; • Baixa latência, permite comunicação bidirecional; • O broker centraliza o recebimento e distribuição das mensagens;
UDP	Sensores Remotos	• Protocolo de transporte sem conexão; • Transporte UDP (que não garante a entrega, ordem oou integridade); • Leve, possui baixo consumo de energia; • Precisa do uso do gateway para traduzir o UDP para HTTP;
HTTP/REST	Comunicação entre gateway e API central	• Protocolo de aplicação baseado em requisições e respostas; • Transporte TCP com TLS; • É padronizado, suporta autenticação robusta;

2. Segurança

Protocolo	**Criptografia**	**Autenticação**
MQTT	TLS	TLS
UDP	Não possui nativamente	Não possui nativamente
HTTP/REST	HTTPS (TLS)	JWT

Arquitetura

A arquitetura é modular, distribuída e segura já que é projetada para integrar sensores IoT de tecnologias diferentes em um sistema centralizado o que garante a interoperabilidade e segurança dos dados;

Os componentes dessa arquitetura são:

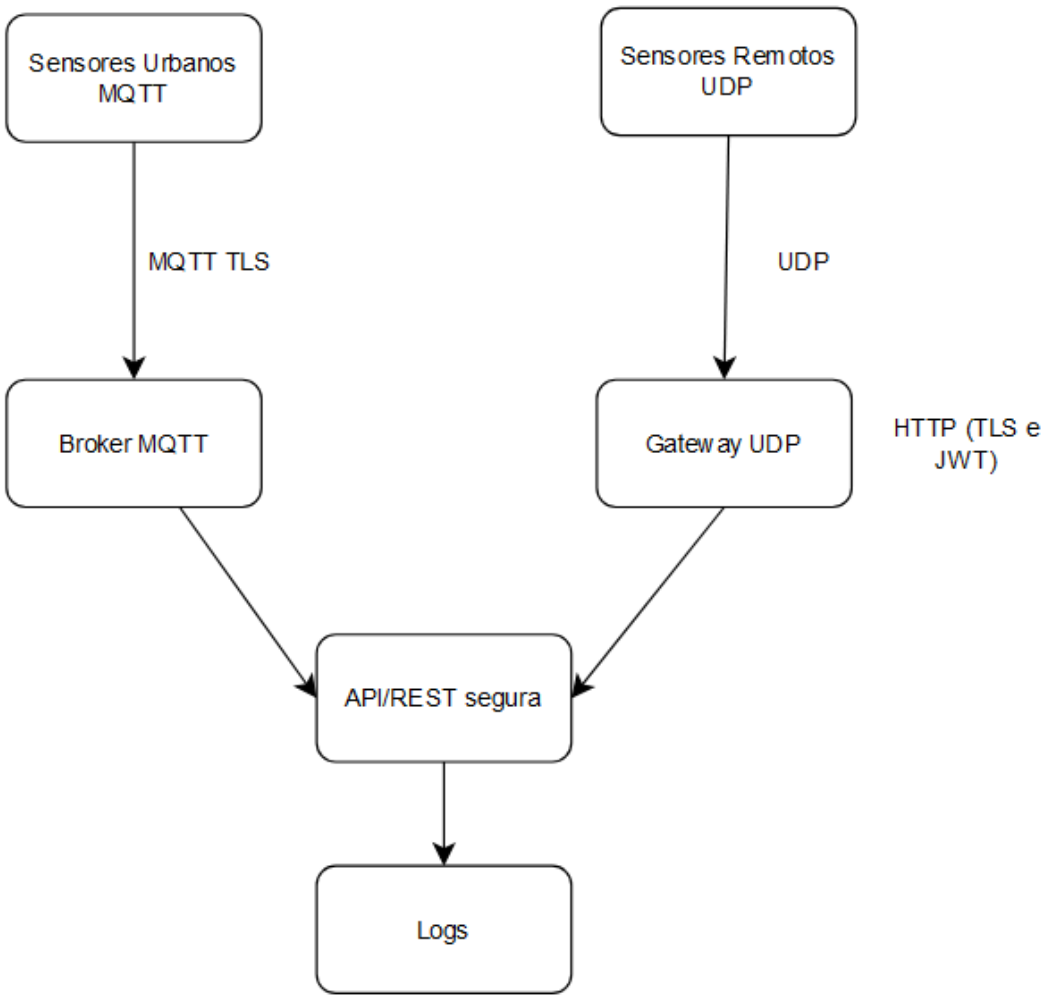
- Os sensores que são urbanos (MQTT) e remotos (UDP);
- Gateway UDP que atua como ponte entre os sensores remotos e o sistema central.
- Broker MQTT que centraliza a comunicação dos sensores urbanos;
- API/REST segura que recebe os dados de sensores diretamente ou via gateway;

O fluxo da arquitetura:

Sensores (MQTT/UDP) -> Broker MQTT/Gateway UDP

Gateway UDP -> (HTTP com JWT) -> API REST

Diagrama


Sensores Urbanos MQTT
Sensores Remotos UDP
MQTT TLS
UDP
Broker MQTT
Gateway UDP
HTTP (TLS e JWT)
API/REST segura
Logs