

# Projeto IoT - API MQTT

## Introdução

O projeto é uma solução para a falta de conhecimento e controle do consumo energético de equipamentos específicos dentro do setor doméstico. Será capturado o nível de tensão consumido de um equipamento através de um sensor que será embutido na tomada de alimentação do mesmo. As informações coletadas serão enviadas através de um microcontrolador com capacidade de comunicação MQTT em uma rede para a internet. Os dados enviados pela internet serão recebidos por um servidor remoto, o qual armazena esses dados em um banco. O servidor remoto também é responsável por transmitir os dados coletados através de uma interface REST para os clientes que quiserem consumir esses dados. O servidor também fica responsável por armazenar regras de funcionamento configuradas pelos clientes e enviar comandos ao atuador presente na tomada, com a possibilidade de, por exemplo, ligar e desligar a tomada.

## Desenvolvimento

### Servidor

O servidor foi construído com três principais objetivos: ser o broker da comunicação MQTT, armazenamento das leituras feitas pelos dispositivos e disponibilizar uma interface REST para o controle dos dispositivos e requisição dos dados armazenados. A construção técnica do servidor foi feita baseada em [Node.js](#) utilizando o superconjunto de javascript [TypeScript](#).

### Broker

Para o desenvolvimento do broker foi utilizado a biblioteca de código aberto [Aedes](#), que tem como ponto forte a facilidade de implementação em cima de qualquer servidor de stream. No protótipo construído foi utilizado o tópico "state" para o envio ao dispositivo cliente seu estado de atuação. Os níveis de consumo emitidos pelo dispositivo cliente são enviados através do tópico "current-value". A comunicação é toda realizada através de QoS 0. A implementação pode ser verificada [aqui](#).

### Banco de Dados

O banco de dados utilizado foi o [SQLite](#). A seleção desse banco foi dada pela sua facilidade em prototipação de pequenos projetos. Não há necessidade de um serviço à parte para seu controle, portanto requer menos consumo computacional, além de ficar acoplado ao sistema. A implementação pode ser verificada [aqui](#).

### REST

A construção da interface REST foi feita utilizando a framework [ExpressJS](#) que fornece uma miríade de métodos utilitários HTTP e middleware, de forma que criar uma API robusta é rápido

e fácil. A API feita fornece duas requisições:

- **GET** /voltage-read
- **GET** /voltage-read?count=X

**Resposta:**

```
{ lastReads: [ { "id": number, "value": number, "date": number } ] }
```

Com a quantidade padrão de leituras sendo 5 e podendo ser modificada para X nos parâmetros da requisição.

- **POST** /set-state

**Corpo:**

```
{ "state": boolean }
```

Com "state" sendo o estado para qual se deseja mandar o dispositivo cliente.

**Respostas:**

200: Requisição executada com sucesso.

400: Requisição com corpo mal feito.

A implementação pode ser verificada [aqui](#).

## Dispositivo Cliente

O processo de sensoriamento, atuação e comunicação com o servidor é controlado por um Arduino Uno. O microcontrolador tem sua leitura sensorial proveniente de um sensor ZMPT101B. Para a atuação o microcontrolador utiliza um driver a relé que faz a ligação entre a rede elétrica e o equipamento que deverá ser alimentado. O envio dos dados para a internet é feito através de um shield HR911105A que fornece conexão Ethernet. Foi feita a implementação da lógica para um cliente MQTT no microcontrolador, de forma que atua enviando o valor da leitura recebida pelo sensor a cada segundo. Além disso, o estado de atuação do driver também é definido através das mensagens recebidas no tópico pré-definido. A implementação feita no Arduino pode ser verificada [aqui](#).