

## Projeto Final do Ciclo 1 - IoT

### 1 Arquitetura Geral da Solução

Neste projeto, será necessário implementar um dispositivo IoT que realiza a leitura de dados de sensores<sup>1</sup> e encaminha esses dados para um software, denominado de Middleware, que vai estar executando na nuvem, por meio de uma conexão TCP. Cada dispositivo IoT terá um identificador único e deverá enviar esse identificador para o Middleware logo após se conectar a ele.

Por outro lado, um conjunto de aplicações podem se conectar a esse software intermediário executando na nuvem para receber os dados gerados pelos dispositivos IoT. As aplicações precisarão se conectar ao software intermediário por meio de uma conexão TCP e informar o identificador de um dos dispositivos que deseja receber os dados. Após isso, todos os dados do dispositivo de interesse serão encaminhados pelo mesmo socket aberto para a conexão inicial.

A Figura 1 mostra a arquitetura do sistema. É importante mencionar que o software intermediário (middleware) será fornecido pronto. Neste projeto, será necessário implementar um dispositivo IoT e uma aplicação, seguindo o protocolo que será detalhado nas próximas seções.

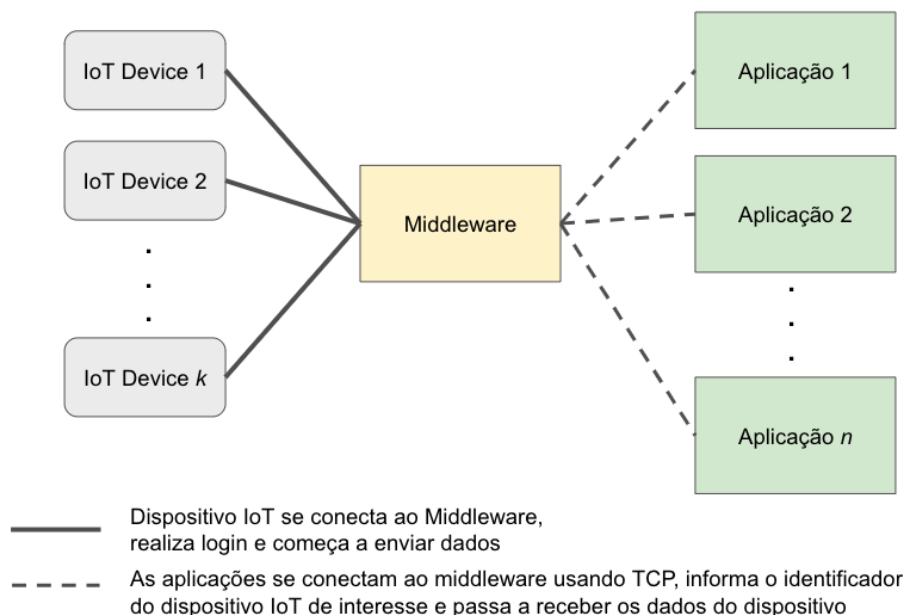


Figura 1: Arquitetura do sistema, descrevendo a conexão entre os dispositivos IoT, o Middleware e as aplicações.

---

<sup>1</sup>por simplicidade, pode-se gerar os valores dos sensores em software, caso não seja possível integrar a partir do simulador ou caso não haja disponibilidade do elemento de hardware

## 2 Funcionamento e Requisitos para o Dispositivo IoT

Antes de começar a enviar os dados, o dispositivo deve se conectar a uma rede Wi-Fi, com acesso à Internet, e realizar um procedimento simples de “Login”. Neste procedimento, deve ser enviado um identificador único do dispositivo. Caso o identificador seja válido, será retornada uma mensagem “ok”, caso contrário, será retornada uma mensagem “fail”. Para esta atividade, será utilizado como identificador único do dispositivo a matrícula do discente responsável pelo projeto no IFPB. Caso o trabalho esteja sendo realizado em dupla, qualquer uma das matrículas dos integrantes do grupo poderá ser utilizada.

Após realizar esse procedimento de login com o Proxy, em caso de sucesso, as seguintes *tasks* devem ser criadas e colocadas em execução:

- Uma *task* que lê um novo dado de um sensor de forma periódica (período maior que 30 segundos) e salva esse dado em uma variável. Após ler um novo dado do sensor, deve-se notificar (usando um semáforo) a segunda tarefa, descrita a seguir. Por simplicidade, pode-se considerar um valor inteiro de 8 bits sem sinal;
- Uma *task* que transmite dados coletados de sensores por meio do socket TCP criado durante o processo de conexão com o Middleware. Esta *task* fica aguardando uma notificação por meio de um semáforo. Quando a notificação ocorre, significa que um novo dado foi salvo e armazenado e uma variável que deve ser acessível pelas duas *tasks*;
- Uma *task* que transmite uma mensagem com a string "alive" para o Proxy a cada 10 segundos, para indicar que o dispositivo está “vivo”. Essa mensagem deve enviar o identificador único do dispositivo.

Perceba que duas *tasks* acessam o socket TCP, que é um recurso compartilhado. Portanto, deve-se utilizar um Mutex para garantir a exclusão mútua e o acesso atômico ao recurso compartilhado.

Como informado no início do documento, pode-se utilizar um sensor “real” disponível no simulador ou por meio de uma implementação com hardware real. O grupo deverá decidir que sensor vai ser utilizado, de acordo com o contexto dos projetos que pretendem desenvolver. No entanto, caso não haja disponibilidade do sensor, pode-se enviar dados gerados via software para simular a geração de dados pelo sensor. Pode-se utilizar a seguinte biblioteca para gerar números aleatórios:

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/random.html>

## 3 Funcionamento e Requisitos para a Aplicação

Do lado da aplicação, deve-se escrever um software simples, em qualquer linguagem de programação (ex: Python), que abre uma conexão TCP com o Proxy e solicita a recepção de dados de um único dispositivo IoT conectado ao Proxy. Logo após realizar a conexão com o Proxy, a aplicação deve enviar a string “app”. Após isso, deve aguardar o retorno do Proxy. Caso seja recebida a mensagem "ok", deve-se enviar o identificador do dispositivo IoT de interesse. Caso o identificador possua um formato válido será recebida uma mensagem "ok" de volta do Proxy. Caso contrário, será retornada a mensagem "fail".

Após isso, a aplicação pode ficar esperando receber dados por meio do socket. Sempre que um valor novo for encaminhado por um dispositivo IoT para o Proxy, ele automaticamente

encaminhará esse valor para todas as aplicações que se inscreveram para receber dados do dispositivo em questão. O dispositivo IoT não precisa estar conectado ao Proxy a priori. A aplicação pode se conectar antes e se associar ao dispositivo. No momento em que o dispositivo se conectar e gerar novos dados, eles serão encaminhados para as aplicações inscritas.

É importante mencionar também que o Proxy não faz persistência e nem bufferiza os dados. Portanto, caso uma aplicação não esteja executando no momento em que o dispositivo gerar um novo dado, esse dado não será armazenado para ser enviado à aplicação posteriormente.

A aplicação a ser desenvolvida neste projeto deve apresentar no mínimo duas funcionalidades baseadas nos dados recebidos (ex: gráfico mostrando os dados em tempo real, cálculo de estatísticas básicas sobre os dados, média dos valores mais recentes, detecção de valores que ultrapassem certo limiar, salvar os dados em um banco de dados de séries temporais etc). Essas duas funcionalidades básicas garantem a aprovação. No entanto, usem a criatividade e pensem em aplicações interessantes. Quanto mais funcionalidades interessantes, melhor será a nota.

## 4 Instruções para Conectar ao Proxy

O código do Proxy está disponível em:

[https://github.com/ruandg/simple\\_proxy\\_iot](https://github.com/ruandg/simple_proxy_iot)

Ele poderá ser usado para testes de forma local (ou seja, usando endereço IP *localhost*). O Proxy também estará implantado e disponível na nuvem, na URL `playback.laced.com.br` e porta 50000. Após o desenvolvimento e testes iniciais, deve-se realizar a integração do dispositivo IoT e da aplicação com esse Middleware executando na Nuvem. Nesse cenário, pode-se executar o dispositivo IoT a partir de qualquer rede e também a aplicação a partir de qualquer rede, desde que possuam acesso à Internet.

No repositório também são fornecidos exemplos de código em Python e em Go, para auxiliar na implementação das aplicações. Também é fornecido um código em Python que simula um dispositivo IoT, que se conecta ao Proxy e envia uma sequência de valores. Esse simulador pode ser usado para testar as aplicações enquanto o sistema embarcado não for concluído.

## 5 Entrega da Atividade

Grave um vídeo curto mostrando o funcionamento do seu dispositivo IoT e da sua aplicação e submeta junto com os códigos fontes no Classroom.