

Sistemas de Comunicação Móvel, 3º Trabalho de Laboratório:

Em anexo seguem-se 4 ficheiros **python** e um ficheiro **ino** necessários para correr este projeto.

Descrição do sistema desenvolvido:

Arquitetura Geral e Descrição do Sistema:

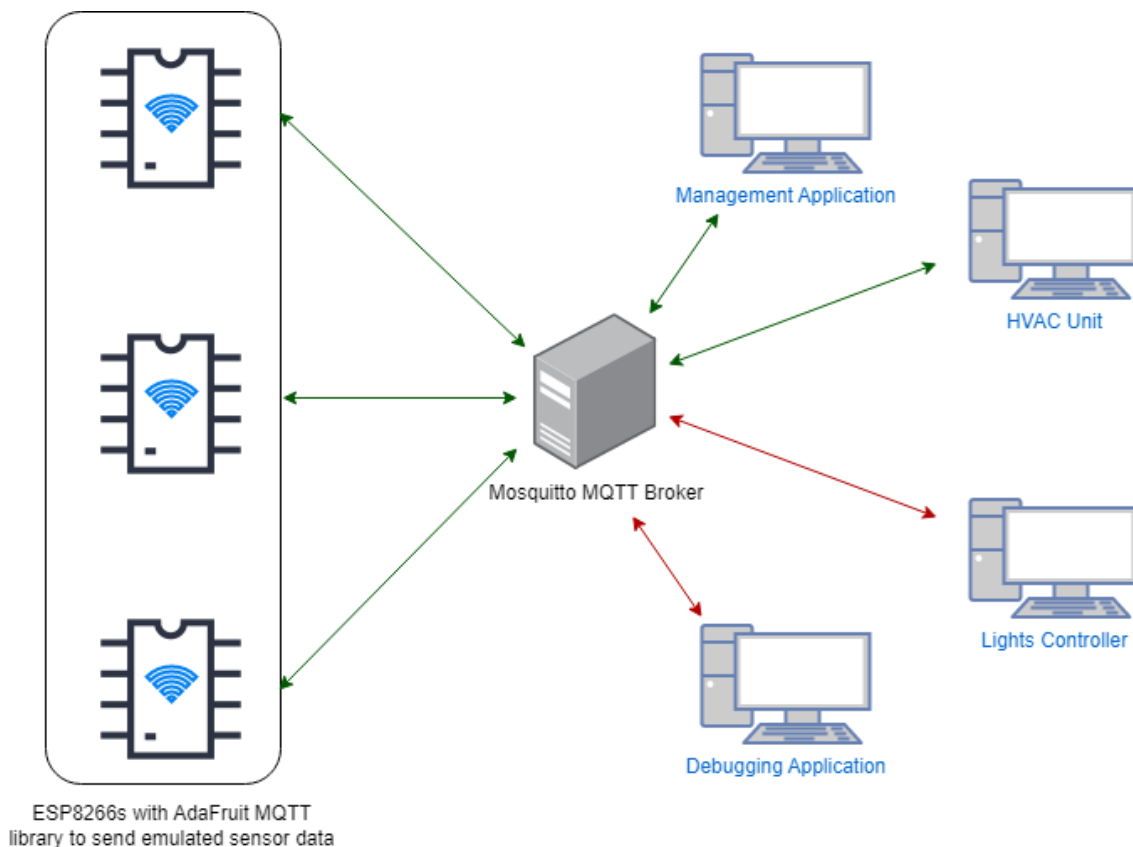


Figura 1 – Arquitetura geral do sistema desenvolvido

O principal objetivo deste projeto consiste em permitir que valores correspondentes a temperatura, dióxido de carbono, luz e presença sejam emulados e enviados através de MQTT para uma aplicação de gestão e a unidade **HVAC**, de forma a poder controlar de forma automática ar condicionado, aquecedor, ventilação e luzes em cada sala correspondente a cada **Arduino**. Do lado do **Arduino** toma-se partido da biblioteca AdaFruit MQTT para estabelecer esta conexão, nas aplicações **Python** usa-se Paho MQTT. A aplicação de gestão é encarregue de definir os limites nos quais os valores se devem inserir para não ativar nenhuma máquina.

Comportamento esperado:

Primeiramente deve-se garantir que os **Arduinos** só serão iniciados após a aplicação de gestão se encontrar em funcionamento. Isto deve-se ao facto de, apesar de termos efetuado bastante pesquisa, chegámos à conclusão de que a opção de retenção do lado do **Arduino** não se encontrava disponível para usar, o que impossibilita esta funcionalidade. Apesar de existirem outras bibliotecas MQTT para usar, o projeto já se encontrava praticamente completo, tornando uma possível alteração de biblioteca um processo bastante complexo. As 4 aplicações **python** devem ser iniciadas logo ao início, contudo todas à exceção da aplicação de gestão possuem mecanismos de retenção que permitem uma ligação tardia, sem qualquer perda de informação relevante para o seu funcionamento.

Ao conectar um ou mais **Arduinos**, a aplicação de gestão irá pedir ao utilizador que insira os limites para cada parâmetro, sendo necessário para calcular no **HVAC** e nele próprio as ações a tomar para garantir o funcionamento adequado de cada sala. É de notar que ao inserir estes valores no primeiro dispositivo, o primeiro input não corresponde ao início destes limites, mas sim à leitura de comandos sempre disponível ao utilizador. De seguida, a mesma fica operacional, aceitando os seguintes comandos:

- **thresholds:** Pede ao utilizador para introduzir o nome de um **Arduino** para poder introduzir novos limites no mesmo.
- **Info:** Pede ao utilizador para introduzir o nome de um **Arduino**, devolvendo toda a informação que sabe relativa ao mesmo.
- **update:** Pede a escolha de um ou múltiplos **Arduinos** de forma a poder enviar-lhes uma nova versão para instalar. Ao efetuar a atualização, o dispositivo irá incrementar a sua versão em 0.1 unidades.
- **Help:** Devolve todos os comandos disponíveis para o utilizador.

O **HVAC** e o **Lights Controller** não requerem nenhum input por parte do utilizador, após serem inicializados, processam toda a informação de forma a poder agir da melhor maneira no caso do HVAC (imprimindo no ecrã sempre que existe uma mudança de estado ou de limites) e recebem toda a informação para ligar ou desligar as luzes das várias salas no caso do **Lights Controller**. Por fim a **aplicação de debug** serve apenas para mostrar todo o tráfego de informação trocado através do MQTT ao longo da utilização do sistema.

Do lado do **Arduino** o utilizador tem ao seu dispor um terminal para poder emular os dados de sensor, com recurso aos comandos: **“temp”**, **“co2”**, **“light”** e **“presence”**. Qualquer outro input devolve uma mensagem de erro, sendo necessário reintroduzir uma palavra correta. Ao atualizar o valor, existe um intervalo de 10 segundos para introduzir um valor numérico diferente do atual.

Descrição das soluções MQTT:

No esquema seguinte (**Figura 2**) segue-se uma interpretação visual dos tópicos utilizado ao longo do projeto, assim como informação dos valores presentes nos mesmos.

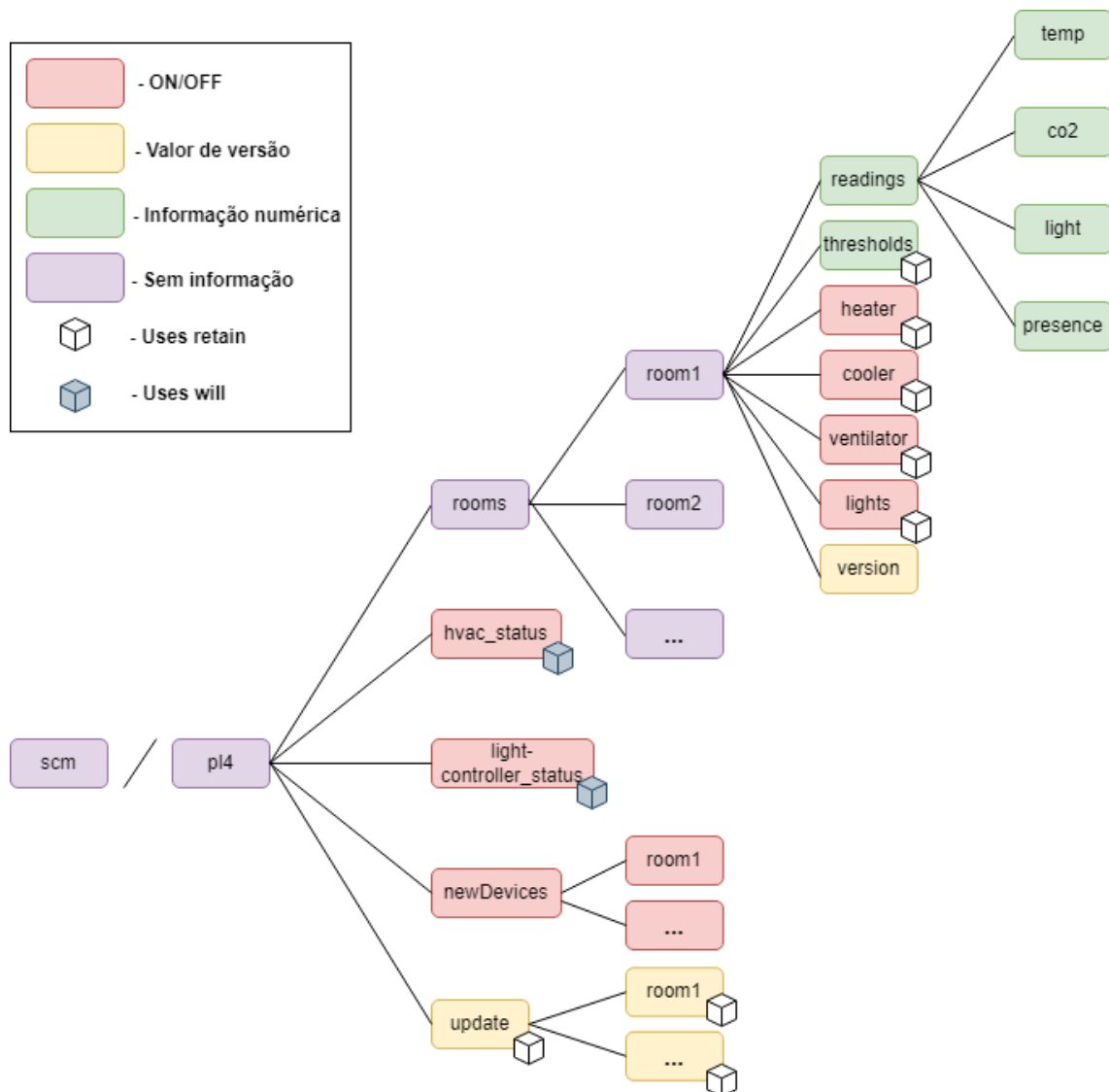


Figura 2 – Estrutura de tópicos MQTT