

ARQUITETURA

O sistema segue uma arquitetura modular para um maior controlo de dependências, uma maior facilidade no desenvolvimento e uma maior escalabilidade futura. A ideia base é semelhante à de todos os serviços web, onde estão presentes componentes de recolha e processamento de dados, de armazenamento numa base de dados e de interface com o utilizador. Na solução definida não existe um componente central e todos os componentes são importantes para o bom funcionamento do serviço web, no entanto em caso de falhas locais é possível manter o sistema ativo limitando as suas funcionalidades como por exemplo o acesso em tempo real à informação atual disponibilizada pelo sistema.

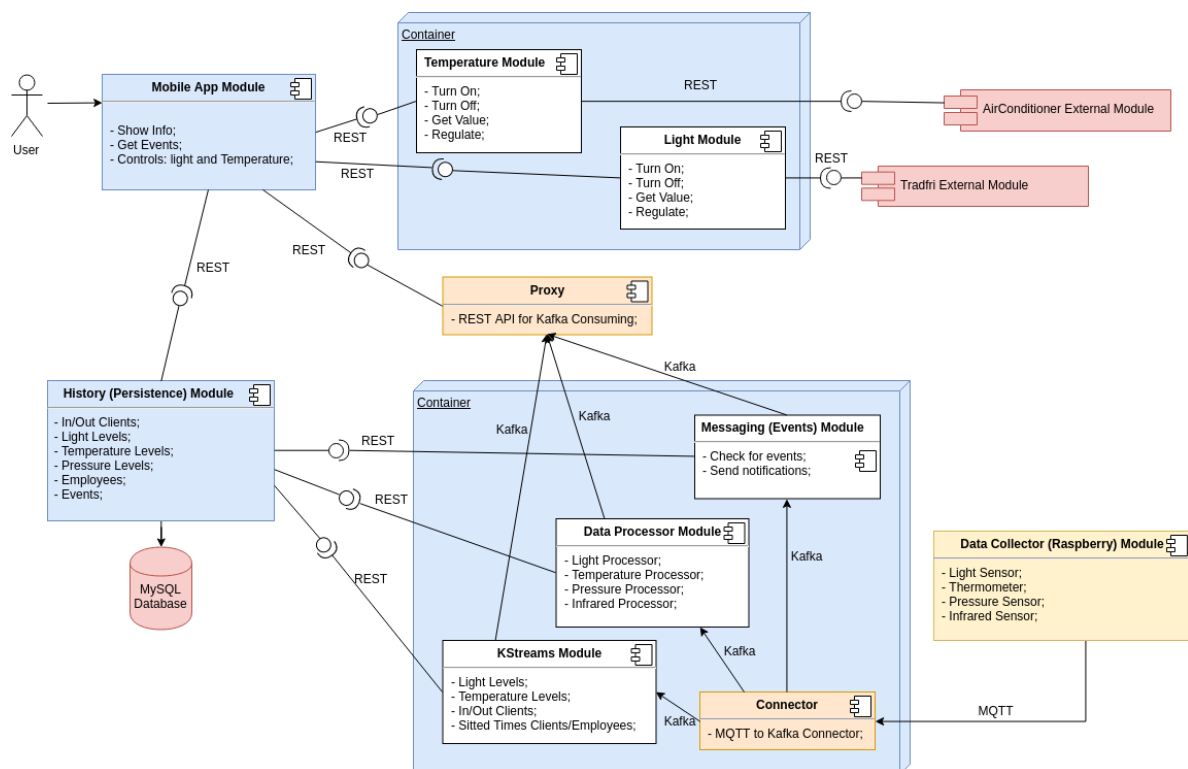
Os módulos definidos são apresentados de seguida:

- Coletor de Dados (ou Módulo *Raspberry*) - responsável por recolher os dados emitidos pelos sensores e envio dos mesmos por MQTT para um script Kafka Connector.
- Processador de Dados (ou Módulo de *Input Data*) - responsável por efetuar o processamento dos dados recebidos pelo Connector por tópicos Kafka, para que o resto do sistema os possa usar. Após processar os dados, envia-os por novos tópicos Kafka para um script Kafka Proxy que serve como REST API para o consumo dos dados por parte da aplicação.
- *Streaming* de Dados (ou Módulo *KStreams*) - responsável pelo processamento dos dados recebidos pelo Connector por tópicos Kafka, para que a aplicação possa ver a evolução dos dados em tempo real. Também recorre a tópicos Kafka enviados para o script Proxy.
- Criador de Mensagens (ou Módulo *Event Handler*) - responsável pela análise dos dados recebidos pelo Connector por tópicos Kafka e envio de eventos por novos tópicos para o Proxy quando a existência destes se verifica.
- Histórico de Dados (ou Módulo de Persistência) - permite armazenar os dados processados para que os mesmos possam ser usados e estudados e haja a garantia de salvaguarda de informação do sistema. Os módulos Processador, Streaming e de Mensagens enviam todos os seus dados para este módulo por pedidos REST. Este módulo armazena a informação numa base de dados MySQL independente.
- Aplicação Móvel (ou Módulo de *User Interface*) - interface de interação com o utilizador. É a partir deste que o mesmo controla o sistema, ajustando os valores necessários. A comunicação deste módulo com os restantes é feita via REST requests.
- Controlador de Temperatura (ou Módulo de Temperatura) - controla os dispositivos externos que servem de ar-condicionados. Este módulo é controlado pelo utilizador via a aplicação e serve de ponte entre o sistema e os dispositivos externos de temperatura.
- Controlador de Luminosidade (ou Módulo de Luminosidade) - controla os dispositivos externos que servem de lâmpadas inteligentes/reguláveis. Este módulo é controlado pelo utilizador via a aplicação e serve de ponte entre o sistema e os dispositivos externos de luminosidade (como o Tradfri).

Em acréscimo estão presentes os módulos externos, não pertencentes ao sistema, mas que interagem e são manipulados por este:

- *Tradfri* - lâmpada inteligente regulada pela aplicação.
- Ar Condicionado - controlador de temperatura regulado pela aplicação.
- Base de Dados (Relacional) - espaço de armazenamento dedicado à informação interna do sistema.

Notas a ter em atenção: apesar de todos estes módulos serem independentes, alguns correm nos mesmos containers para assim reduzir a complexidade do processo de *deployment*. O esquema abaixo apresentado permite visualizar as ligações entre módulos do sistema OpenDoors. Os componentes a azul são parte íntegra do sistema OpenDoors, os a vermelho são componentes externos (sendo que fornecemos 2 dispositivos virtuais para testes da manipulação da temperatura e luminosidade), o a amarelo é um script desenvolvido por nós para coletar os dados dos sensores mas que não faz parte do sistema e os a laranja são scripts externos que utilizamos como pontes entre módulos internos do sistema.



As comunicações internas e externas são feitas segundo protocolos conhecidos e formatos de mensagem standard. O módulo coletor de dados comunica com o processador através do protocolo MQTT – um protocolo de mensagens *publish-subscribe* leve e apropriado para este tipo de comunicações. As restantes comunicações à base de mensagens são feitas com base em Kafka – uma plataforma distribuída de mensagens e streaming. Todas as restantes comunicações são feitas segundo a arquitetura REST operando apenas com os métodos standard *get*, *put*, *post* e *delete*.

Todos os dados transferidos entre módulos são enviados em formato JSON para uma fácil manipulação, legibilidade e usufruto das suas restantes vantagens.