

Especificação do Trabalho 3: RabbitMQ e gRPC

Sistemas Distribuídos

Professor: Dr. Paulo A. L. Rego

OBJETIVO

O objetivo deste trabalho é consolidar o conhecimento dos alunos sobre invocação de métodos remotos e comunicação indireta, além das ferramentas RabbitMQ e gRPC, através de um trabalho prático que envolva ambas.

DESCRIÇÃO GERAL

O trabalho consiste em simular um ambiente inteligente (por exemplo, casa, escritório, sala de aula, clínica médica, carro, etc). Neste ambiente deverão estar presentes **Sensores** (que coletam dados do ambiente) e **Atuadores** (que podem agir no ambiente para modificá-lo de alguma forma). Por exemplo, em um ambiente residencial inteligente, podem existir sensores de temperatura que coletam periodicamente a temperatura ambiente e aparelhos como ares-condicionados e aquecedores que podem trabalhar como atuadores e agir para modificar a temperatura ambiente.

Dentro deste ambiente inteligente, deverão existir, pelo menos três tipos de sensores e três tipos de atuadores para uma determinada característica do ambiente. Por exemplo, no ambiente residencial, nós podemos ter sensores de temperatura, de luminosidade e de fumaça e, associados a eles como atuadores, podemos encontrar ares-condicionados/aquecedores, lâmpadas inteligentes e sistemas de controle de incêndio. Note que é de extrema importância que cada sensor tenha um atuador relacionado no ambiente.

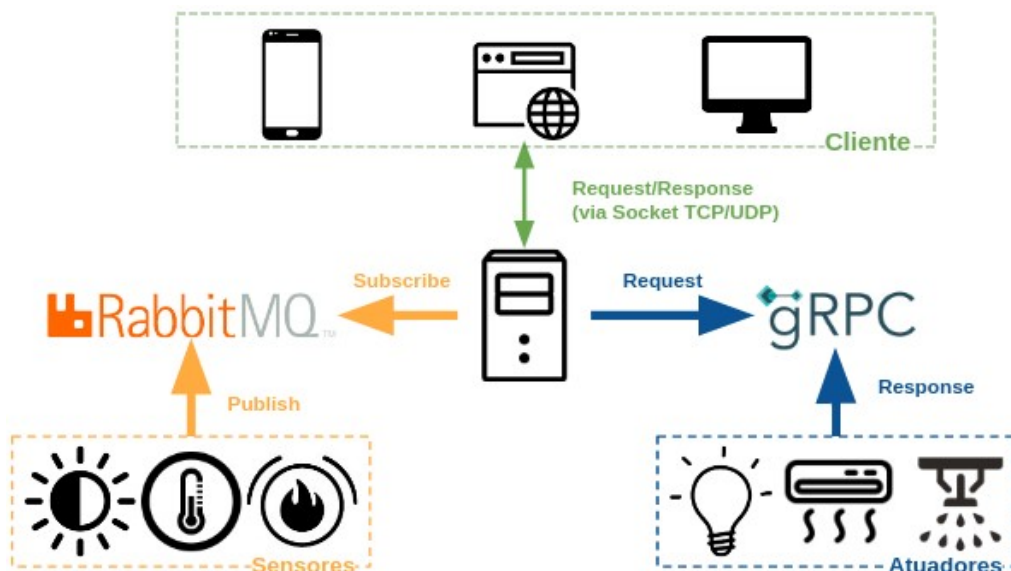
Todos esses sensores e atuadores serão gerenciados por um equipamento servidor chamado **Home Assistant**. Este equipamento deverá interagir com os sensores e os atuadores coletando informações e, eventualmente, agindo sobre o ambiente.

A comunicação entre os sensores e o Home Assistant deverá ocorrer via RabbitMQ, usando o paradigma Publisher/Subscriber, onde o Home Assistant se comportará como Subscriber e cada sensor como Publisher. Cada sensor deverá publicar periodicamente os dados por ele observados em uma fila própria no RabbitMQ, que se encarregará de notificar o Home Assistant sobre a nova mensagem. Por exemplo, o sensor de luminosidade deverá a cada 5 segundos (definido estaticamente no código-fonte) submeter o nível de luminosidade a uma fila *luminosidade.sensor1*. O Home Assistant, que já deverá ser assinante da fila, será notificado do evento e deverá interagir com o Home Assistant para coletar o dado em questão.

A comunicação entre os atuadores e o Home Assistant, por sua vez, deverá ocorrer via gRPC, usando o paradigma Client/Server, onde o Home Assistant se comportará como Client e cada

atuador como Server. Ainda no exemplo anterior, considere uma lâmpada inteligente com duas possíveis ações: ligar ou desligar. A lâmpada então deverá oferecer uma interface de invocação remota com dois métodos: *ligarLampada()* e *desligarLampada()*. Dessa forma, o Home Assistant poderá atuar no ambiente através da invocação remota desses métodos, por exemplo, se ele desejar ligar uma determinada lâmpada, ele deve invocar, via gRPC, o método *ligarLampada*.

O HomeAssistant também deverá se comportar como um servidor para uma aplicação cliente que permita ao usuário interagir com o ambiente. Através dessa aplicação (que poderá ser Desktop, Web ou Mobile), o usuário poderá receber as informações de momento do ambiente (por exemplo, o nível de luminosidade detectado por cada sensor) e também poderá agir sobre ele (por exemplo, ligando ou desligando uma lâmpada). A Figura abaixo resume a arquitetura do ambiente a ser simulado.



INSTRUÇÕES DE ENTREGA

Um aluno de cada grupo deve enviar através do SIGAA (Atividade Trabalho 3 - T3), **até o dia 19/03/2021**, os seguintes dados para a entrega do trabalho:

1. Nome e número de matrícula dos membros do grupo;
2. Link para apresentação de slides que contenha os detalhes de implementação do trabalho, em especial as informações sobre o formato das mensagens trocadas entre os processos do sistema distribuído, linguagens de programação utilizadas, além de frameworks e bibliotecas utilizadas (caso seja necessário).
3. Anexar arquivo zip com o código fonte do trabalho no SIGAA. Opcionalmente, pode mandar o link de algum repositório público onde o código fonte esteja hospedado.
4. Link de vídeo apresentando os detalhes de implementação e demonstração da aplicação funcionando.