Relatório IoT: Smart Classroom

Bruno Nogueira[1], Daniel Cota[1], Franklin Ventura[1]

[1]Instituto de Computação (IC)

Universidade Federal Fluminse (UFF) – Niterói, RJ – Brazil

1. Introdução

Ao longo de cada período letivo, é possível observar a monótona tarefa que os professores têm que realizar, desde a busca pelas chaves para acessar as diferentes salas de aula até lidar com os dispositivos eletrônicos que compõem a infraestrutura dos estabelecimentos. Esse processo além de ser repetitivo e gerar retrabalho, impacta muitas vezes no tempo de aula dos professores em virtude de eventual esquecimento da chave ou problemas técnicos dos seus componentes.

Nos últimos anos, a sociedade tem presenciado o notório desenvolvimento das redes móveis, especialmente com o advento das tecnologias 4G e 5G. Diante do aumento sem precedentes da conectividade, é possível observar a concretização do paradigma da Internet das Coisas, no qual tecnologias habilitadoras e protocolos de comunicação permitem que objetos físicos sejam capazes de interagir com o ambiente, comunicar-se entre eles, compartilhar informações e coordenar decisões [Al-Fuqaha, Ala, et al., 2015].

Dentre os diferentes domínios de aplicação que tem utilizado IoT é possível citar o smart building. Diante disso, observa-se a possibilidade de utilizar um sistema baseado em IoT para auxiliar os professores da instituição em suas tarefas, por meio do uso de um identificador único para entrar e sair das classes e do controle automático dos dispositivos eletrônicos que fazem parte da infraestrutura das salas de aula. Além disso, aproveitando a infraestrutura que será criada também será feito um monitoramento da temperatura e umidade das salas de aula a fim de fornecer à instituição dados para análise da qualidade do ambiente de ensino.

O projeto tem como objetivos garantir ao professor que ao entrar e sair da sala haverá o acionamento e desligamento automático dos dispositivos que compõem a infraestrutura da sala de aula (luz, ar-condicionado, projetor etc) e fornecer a instituição (UFF) dados para medição da temperatura e umidade das salas, visando melhorar o ambiente da sala de aula. Para tal, foi elaborada uma aplicação web, denominada de Smart Classroom Frontend, cujo objetivo é melhorar a experiência de ensino e aprendizagem, fornecendo :

- A criação das diferentes entidades representando todos os componentes da infraestrutura que compõem o ambiente universitário desde: universidades, campus, andares, salas e os seus dispositivos (ar-condicionado, projetor, ..).
- Dashboard contendo o monitoramento das condições da sala de aula, como temperatura e umidade.

Para tal, este trabalho procura realizar uma prova de conceito da criação de uma sala inteligente, ou seja, não envolverá a elaboração de um caso prático da implementação das salas inteligentes, levando em consideração todas as variáveis que esta proposta envolve. Logo, o projeto consistirá em um mvp, contendo apenas as funcionalidades básicas

envolvendo as salas de aula, que poderão ter seus dispositivos configurados e monitorados via aplicação web .

O documento está dividido da seguinte forma: a seção 2 apresentará as ferramentas utilizadas para a realização do projeto; na seção 3 é apresentada a arquitetura elaborada para a implementação do Smart Classroom Frontend. Já a seção 4 informa os passos para executar a aplicação e finaliza-se com a seção 5 contendo as conclusões do trabalho desenvolvido.

2. Ferramentas utilizadas

Esta seção tem como objetivo apresentar as principais tecnologias utilizadas neste trabalho, tais como Fiware, Orion Context Broker e IoT Agents (IDAS) e os dummy devices usados para aquisição, processamento e atuação nos dados. Além disso, apresentar como cada um dos componentes foi utilizado no projeto.

2.1. Fiware

Segundo [Barriga et al 2022] Fiware é uma plataforma de código aberto que define e implementa um conjunto universal de normas para a gestão de dados contextuais com o objetivo de otimizar o desenvolvimento de ambientes IoT em diferentes domínios. Dentre eles é possível citar: as cidades inteligentes, os edifícios inteligentes, a agricultura inteligente e as smart grids.

2.2. Orion Context Broker

É um broker para geração e manipulação de informações de contexto, ou seja, um servidor que recebe mensagens de seus clientes geradores de dados e os envia aos seus clientes consumidores. É responsável por gerenciar todo o ciclo de vida das informações de contexto, incluindo atualizações,consultas, registros e assinaturas.

2.3. IoT Agents

Um agente IoT atua como um componente de middleware que converte solicitações NGSI-v2 em um protocolo utilizável pelos próprios dispositivos IoT. São responsáveis pela criação dos serviços, mas para isso também será utilizado para este trabalho a aplicação web, que permitirá cadastrar cada serviço através de um formulário, que deverá então ser enviado ao Agente IoT.

2.4. Dummy devices

Os chamados “dummy devices” são aplicações que simulam o funcionamento de um dispositivo que contém um conjunto de sensores e atuadores. A funcionalidade atual destes permite receber solicitações, o que ativará funcionalidades como o envio de dados como se fosse um sensor ou uma solicitação para desativar ou desligar os sensores, o que na realidade seria parar de enviar dados.

Para testes iniciais foi criada uma interface onde são acessados os dados de um dispositivo e a partir desta interface é enviada uma solicitação para ativar um dos dispositivos. O processo de comunicação ocorre da seguinte forma:

- A interface web faz uma solicitação ao Orion, esta solicitação contém o ID do dispositivo e os comandos a serem executados.
- Orion recebe a solicitação da aplicação web, valida a solicitação e envia uma solicitação ao IoT Agent e, tendo comunicação direta com os dispositivos ‘Device’,

faz uma solicitação enviando um JSON que contém o comando a ser executado e as variáveis de configuração.

- O ‘Device’ analisa o comando a ser executado, processa as informações e retorna a resposta ao IoT Agent, mas neste caso as respostas enviadas do ‘Device’ são enviadas para a porta 7826 do IoT Agent.
- O IoT Agent processará os dados e enviará atualizações para o Orion, e o Orion registrará os dados no mongodb.
- A aplicação web pode consumir os dados de atualização usando a API Orion.

O objetivo final é criar uma funcionalidade na aplicações web, para que os dados sejam atualizados periodicamente caso o dispositivo esteja em estado ativo.

2.5. Servidores Proxy

A comunicação entre a aplicação web e o Agente IoT é realizada através de dois servidores proxy, para cada uma de suas portas de IoT Agent (4041, 7896). Os servidores proxy foram criados, pois devido às políticas CORS (Cross-Origin Resource Sharing) em aplicações web, os serviços externos neste caso IoT Agent devem incluir cabeçalhos CORS em suas respostas. E como o IoT Agent não inclui esses cabeçalhos, por meio de servidores proxy, adicionamos cabeçalhos CORS nas respostas do IoT Agent.

A aplicação web também contém uma interface onde são exibidos os diferentes serviços criados no IoT Agent. A criação de ‘Devices’ também é feita através de um formulário na aplicação web. Os dispositivos são registrados usando a API IoT Agent. Este por sua vez enviará uma solicitação ao Orion, registrando uma entidade do tipo Device.

Quando os 'Devices' são iniciados, eles enviam dados periodicamente ao IoT Agent, que é responsável por validar se o 'Device' está autorizado para enviar dados. Após a autorização do Device pelo IoT Agent, seus dados são enviados para o Orion, que se encarrega de registrar esses dados no MongoDB.

Uma das funcionalidades oferecidas pelo Orion é o processo de assinatura, que consiste em monitorar um tipo específico de dados de um 'Device' e, caso haja alguma alteração, comunicar a um servidor externo sobre essa mudança por meio de uma solicitação POST.

Na proposta deste trabalho, o Orion não pode estabelecer uma comunicação direta com a nossa aplicação web, pois esta última não permite a criação de rotas de serviço nem o tratamento de solicitações POST. Visto que esta aplicação web opera no lado do cliente por meio de uma interface e não foi projetada para lidar com lógica de servidor ou receber solicitações diretamente, implementando um micro servidor (server.js). Neste micro servidor, foi criado "rotas" específicas para permitir que o Orion comunique as alterações nas variáveis do 'Device'. Além disso, como os dados provenientes do Orion são periódicos (a cada 5 segundos), foi incorporado um socket que facilita uma comunicação constante entre server.js e a aplicação web.

3. Arquitetura

Nesta seção será apresentado a arquitetura elaborada para implementar o Smart Classroom Frontend.

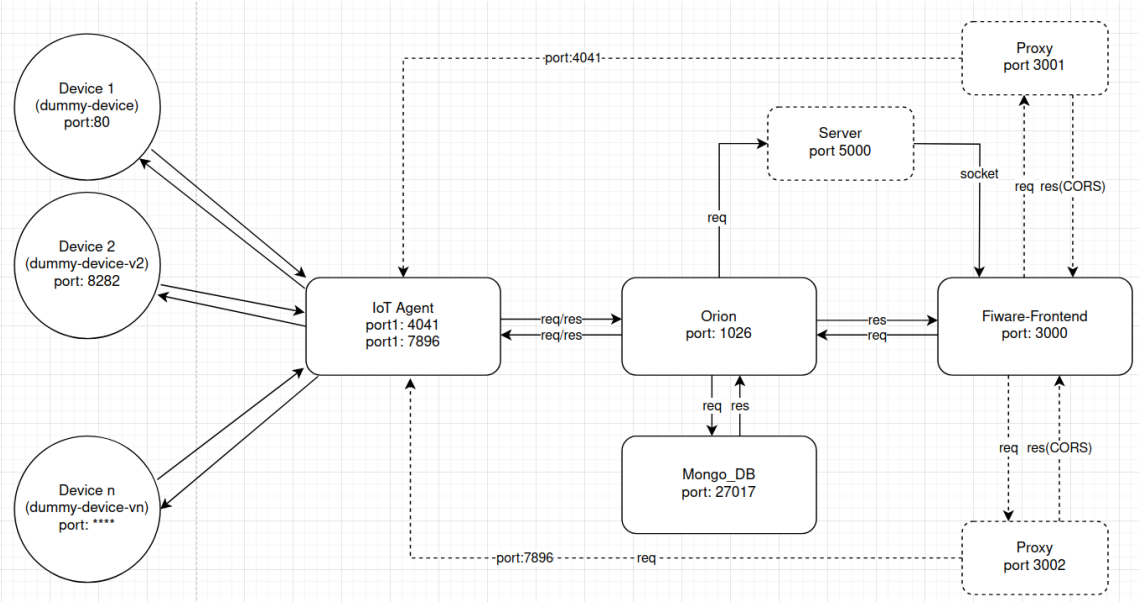

Figura 1: Arquitetura do Smart Classroom Frontend

4. Aplicação

É uma aplicação web criada com React, que permite a criação das diferentes entidades utilizadas no nosso projeto, por sua vez é implementada uma lógica para que as entidades sejam criadas seguindo a seguinte hierarquia:

- Universidade
    - Campus
        - Andares
            - Salas
                - Devices

A implementação do trabalho se vale dos conceitos da Internet das Coisas e de todas as tecnologias por trás do FIWARE, de modo a colocar em prática os conceitos e ferramentas aprendidos em aula. A seguir, está listado um passo a passo para criar o ambiente e executar algumas funções nele. Para baixar o projeto, você precisará clonar o repositório criado no github[1] usando a seguinte linha de comando:

- Para criar as imagens dos containers, você deve entrar na pasta Smart-Classroom-IoT e executar a seguinte linha de comando.
    - linux-ubuntu: sudo docker-compose build
    - windows: docker-compose build
- Para inicializar os containers você deve executar a seguinte linha de comando:
    - linux-ubuntu: sudo docker-compose up

[1] https://github.com/fventuraq/Smart-Classroom-IoT.git

    - windows: docker-compose up -d
- Para inicializar a aplicação web do Smart Classroom Frontend, você deve ir até a pasta /fiware-frontend e executar o seguinte comando:
    - Primeira execução
        - npm install --force
        - npm start
- Se você já instalou as dependências:
    - npm start
- Para inicializar os servidores proxy você deve ir até a pasta fiware-frontend/src e executar os seguintes comandos.
    - node proxy.js
    - node proxy2.js
- Para inicializar o micro servidor você deve ir até a pasta fiware-frontend/src e executar os seguintes comandos.
    - node server.js
- Ao criar a assinatura, lembre-se que o endereço onde as notificações chegaram é no seu IP na porta 5000/notify.

5. Conclusão

O presente trabalho foi capaz de implementar uma interface web representando um ambiente simulado das salas de aula nas universidades, com o intuito de desenvolver aplicações de Internet das Coisas baseadas no Fiware e colocar em prática os conceitos aprendidos ao longo do semestre.

Referências

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE communications surveys & tutorials, 17(4), 2347-2376.

Barriga, J. A., Clemente, P. J., Hernández, J., & Pérez-Toledano, M. A. (2022). SimulateIoT-FIWARE: domain specific language to design, code generation and execute IoT simulation environments on FIWARE. IEEE Access, 10, 7800-7822.