

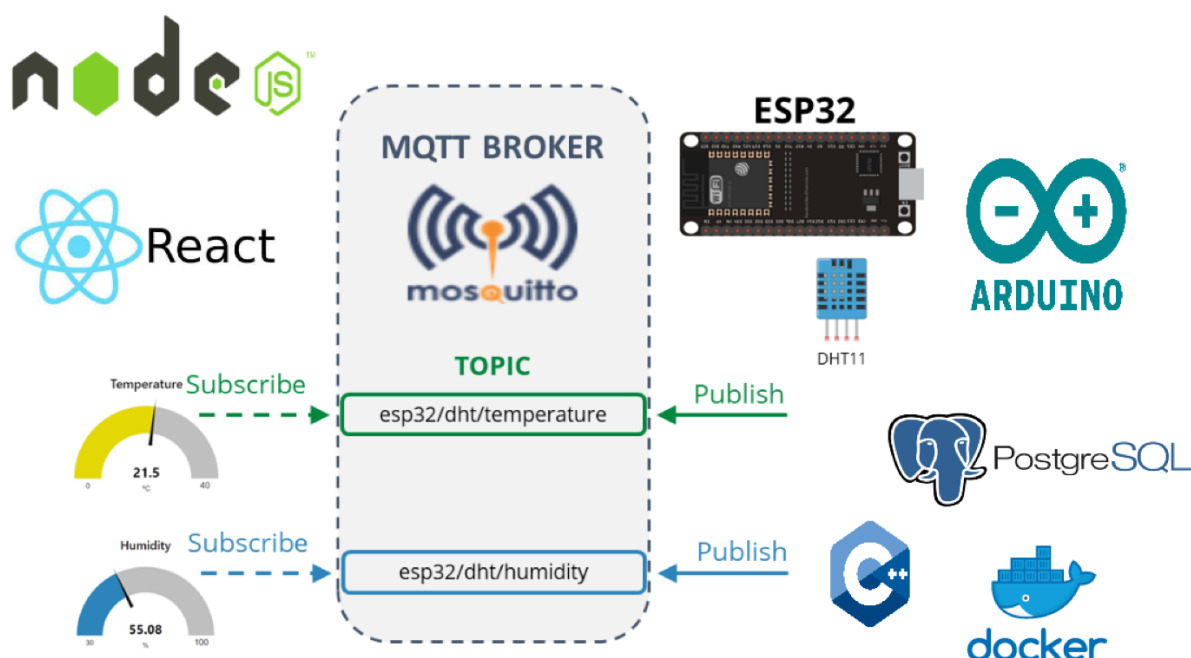
UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

Gustav Shigueo Nicioka Asano 11212355
Marcos Antonio Nobre Coutinho 10716397
Pedro Fernando Christofolletti dos Santos 11218560

**Publicar leituras de temperatura e umidade de um sensor DHT11 via
MQTT com o ESP32 (CHECKPOINT 01)**

São Carlos
2022

1 INTRODUÇÃO



Como proposto na disciplina SSC0952 - Internet das Coisas, esse projeto irá unir hardware + software para medir e conferir a temperatura e umidade de uma sala do ICMC. Foi disponibilizado pelo professor uma máquina virtual e um ESP32, que é uma série de microcontroladores de baixo custo e baixo consumo de energia com microcontrolador integrado, Wi-Fi e Bluetooth. A série ESP32 emprega um microprocessador Tensilica Xtensa LX6 com duas variações dual-core e single-core e inclui uma antena integrada RF tipo balun, amplificador de potência, receptor de baixo ruído amplificado, filtros, gerenciamento de energia dos módulos.

Para começar, teremos a série ESP32 com um sensor de temperatura e umidade do tipo DHT11 acoplado e usando uma alimentação de 5V 2.5A (carregador de celular). Usaremos a plataforma Arduino IDE em linguagem C++ em conjunto com as bibliotecas: Async MQTT Client Library e DHT library from Adafruit para publicar as leituras do sensor em dois tópicos (umidade e temperatura).

Para isso, será instalado o Mosquitto Broker na máquina virtual disponibilizada pelo professor (LaSDPC). Também usaremos SGDB PostgreSQL para armazenamento das informações de usuário e logs. Quanto ao responsável por ler os tópicos e executar queries no banco, teremos uma api em Node Js utilizando o framework Express.js que irá ler os dois tópicos do broker e que também contará com um sistema de autenticação utilizando da tecnologia RFC do JSON Web Token, bem como o query builder Knex para execução das queries no banco. Essa API será consumida por uma interface web.

Por fim, toda essa estrutura estará dentro de um container do Docker.

2 APLICAÇÃO

A interface escolhida é a web, pois assim a aplicação pode ser acessada por qualquer dispositivo que possua um browser instalado (computadores, smartphones, tablets, TVs, etc.). Utilizaremos a biblioteca de javascript React.js, seguindo o padrão “Single Page Application”, também contaremos com o conjunto de pacotes da Material UI do Google para o design.

O DHT11 enviará sinais de temperatura e umidade para a ESP32, e estes sinais serão publicados num tópico específico no broker do Mosquitto. O nosso back-end em NodeJS, que estará consumindo as mensagens publicadas no tópico, poderá ler essas informações e enviar para o nosso front-end em ReactJS para que os usuários autenticados possam ter conhecimento dessas informações.

3 BROKER

O Mosquitto é um broker leve que pode ser utilizado em microcontroladores até grandes computadores. Ele utiliza o protocolo MQTT e servirá como mediador para que a comunicação aconteça, sendo responsável em receber, filtrar, decidir quem precisa e publicar as mensagens para os clientes inscritos. Podendo ser usado localmente ou com a nuvem para acesso a diferentes dispositivos com conexão à internet.

Por utilizar o protocolo MQTT, possui as mesmas desvantagens como um ciclo de transmissão lento, não possui suporte a transmissão de vídeo e a segurança é bem simples com poucas autenticações, apenas para usuários e senhas com formato de texto.

4 ARMAZENAMENTO

O PostgreSQL é um programa open-source de dados que usa a linguagem SQL, que possui uma arquitetura no modelo cliente-servidor.

As vantagens do PostgreSQL é sua fácil utilização, possui tipos de dados definidos pelo usuário, por ser open-source tem suporte da comunidade e tem suporte ao ACID (**A**tomicidade, **C**onsistência, **I**solamento e **D**urabilidade).

Enquanto suas desvantagens se destaca o mal desempenho e velocidade comparado com outras ferramentas.

5 MICRO-SERVIÇO

Seleção da linguagem e do framework de desenvolvimento: como decidimos por utilizar uma plataforma web, devido à sua praticidade (é possível acessar de diversos dispositivos diferentes), e dado que JavaScript é amplamente utilizado para a web, escolhemos então utilizar frameworks JavaScript para o desenvolvimento da

interface web deste projeto. Para o front-end, utilizaremos ReactJS e para o back-end utilizaremos NodeJS.

6 SEGURANÇA

O protocolo MQTT (Message Queuing Telemetry Transport) é um protocolo de mensagens entre máquinas open-source desenvolvido pela IBM. Por ser um protocolo leve e simples é muito utilizado em aplicações para a IoT, o que permite utilizá-lo em locais com internet limitada. Também não sofre sobrecarregamento, pode ser instalado na maior partes dos hardwares, possui um consumo baixo de energia e algum nível de segurança para os dados.

Ele funciona a partir de uma conexão TCP/IP que depois o cliente pode ser autenticado por um certificado SSL/TLS ou por um usuário e senha. Após a autenticação, o cliente pode se inscrever e publicar mensagens e operações.

As desvantagens são um ciclo de transmissão lento, não possui suporte a transmissão de vídeo e a segurança é bem simples com poucas autenticações, apenas para usuários e senhas com formato de texto.

