



Universidade Federal de Ouro Preto - UFOP
Escola de Minas
Colegiado do curso de Engenharia de Controle e
Automação - CECAU



Camilo Esteves Mendes de Avelar

**Domótica aplicada no controle de água utilizando comunicação
MQTT, arquitetura de microsserviços e Docker: uma solução IoT.**

Monografia de Graduação em Engenharia de Controle e Automação

Ouro Preto, 2019

Camilo Esteves Mendes de Avelar

Domótica aplicada no controle de água utilizando comunicação MQTT, arquitetura de microsserviços e Docker: uma solução IoT.

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Filipe Augusto Santos Rocha

Ouro Preto, 2019

Camilo Esteves Mendes de Avelar

Domótica aplicada no controle de água utilizando comunicação MQTT, arquitetura de microsserviços e Docker: uma solução IoT./ Camilo Esteves Mendes de Avelar. – Ouro Preto, 2019-

33 p. : il. (algumas color.) ; 30 cm.

Orientador: Filipe Augusto Santos Rocha

Monografia de Graduação em Engenharia de Controle e Automação – Universidade Federal de Ouro Preto - UFOP, 2019.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade xxx. III. Faculdade de xxx. IV. Título

CDU 02:141:005.7

Monografia defendida e aprovada, em *XX* de *XX* de 2019, pela comissão avaliadora constituída pelos professores:

Filipe Augusto Santos Rocha
Orientador

Convidado

Convidado

Ouro Preto, 2019

Agradecimentos

Os agradecimentos depois de pronto.

Resumo

Palavras-chaves: iot, sistema, monitoramento, esp8266

Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Figura 1	Diferença entre microprocessador de microcontrolador	13
Figura 2	ESP8266	14
Figura 3	RaspberryPi	14
Figura 4	Sensor de fluxo de água YF-S201	15
Figura 5	Teclado Matricial de Membrana	15
Figura 6	Circuito do teclado	16
Figura 7	Pinagem do teclado	16
Figura 8	Válvula Solenoide	17
Figura 9	Dashboard genérico do HomeAssistant	18
Figura 10	Exemplo arquitetura de microsserviços	21
Figura 11	Exemplo da arquitetura do MQTT	22

Sumário

1	Introdução	10
1.1	Objetivos	11
1.1.1	Objetivos específicos	11
1.2	Justificativas e Relevância	12
1.3	Organização do trabalho	12
2	Materiais e Softwares	13
2.1	Microcontroladores e Microprocessadores	13
2.1.1	ESP8266	13
2.1.2	RaspberryPi	14
2.2	Sensores e atuadores	14
2.2.1	Sensor de fluxo YF-S201	14
2.2.2	Teclado matricial de membrana	15
2.2.3	Válvula solenoide	16
2.3	Softwares	17
2.3.1	HomeAssistant	17
2.3.2	Banco de dados de séries temporais	18
2.3.2.1	InfluxBD	19
2.3.3	Banco de dados relacional	19
2.3.3.1	PostgreSQL	19
2.3.4	Node.JS	20
2.3.4.1	Arquitetura de microsserviços	20
2.3.5	Protocolo de comunicação	20
2.3.5.1	MQTT	20
2.3.6	MQTT Mosquitto	22
2.3.7	Docker	22
2.3.8	Grafana	23
3	Metodologia	24
3.1	Configuração do RaspberryPi	24
3.2	Microsserviços	24
3.3	Criação das tabelas no PostgreSQL	24
3.4	Configuração do InfluxDB	24
3.4.1	Sistema de usuários	25
3.4.2	Sistema de comunicação MQTT	25
3.5	Configuração do HomeAssistant	26

3.6	Configuração do Grafana	26
3.7	Configuração do Docker	26
4	Experimentos e Resultados	27
5	Considerações Finais	28
	Referências	29
	Apêndices	32
	Anexos	33

1 Introdução

Setenta por cento da superfície do planeta é coberta por água, quase toda salgada e, portanto, imprópria para o consumo humano. Apenas 2,5% desse total é potável e a maior parte das reservas (cerca de 80%) está concentrada em geleiras nas calotas polares. Essa quantidade mínima de recursos aliada ao contínuo e intenso crescimento demográfico ao longo dos anos, o desenvolvimento industrial e, por consequência, o aumento do consumo de água nas grandes cidades, tem sido um dos principais temas de discussões e palestras de conscientização por todo o mundo.([ÁGUA...](#),)

Pesquisas mostram que, em poucas décadas, as reservas de água doce do planeta não serão suficientes para suprir as necessidades humanas caso os níveis de consumo não sejam controlados desde já ([DIÁRIAS et al., 2007](#)). A escassez deste recurso essencial à vida acarretará em problemas de ordem política, econômica, sanitária, podendo até originar conflitos similares aos causados pelo domínio do petróleo.

A economia de água é um assunto recorrente que há muito deixou de ser restrito às regiões áridas e desérticas com baixa disponibilidade de água per capita, faz com que governos e organizações de todo o mundo estejam com atenções voltadas para a criação de políticas de consumo sustentável, programas de educação ambiental, alternativas e soluções para a redução e controle do uso da água.([FERREIRA; HEROSO; ZALESKI](#),).

A fim de evitar consequências como a escassez da água, o consumo responsável encabeça a lista de medidas a serem tomadas, por se tratar de uma atitude factível a todas as pessoas. ([DIÁRIAS et al., 2007](#)). Recentemente, avanços em recursos computacionais e tecnologias de eletrônicos permitiram a criação do paradigma do IoT (Internet of Things ou Internet das Coisas). ([PERUMAL; SULAIMAN; LEONG, 2016](#)) descreve a Internet das Coisas como sendo um método para conectar coisas em torno do ambiente e realizar um certo tipo de troca de mensagem entre eles, integrando-os.

O IoT representa uma rede mundial de objetos interconectados e unicamente endereçados. É uma interconexão de dispositivos sensores e atuadores que proveem a habilidade de compartilhar informações entre plataformas através de um framework unificado, desenvolvendo uma comum capacidade de criar aplicações inovadoras. Isto é possível devido a sensores, análise de dados e representação de informações através de Computação em Nuvem como o framework unificado. ([Risteska Stojkoska; TRIVODALIEV, 2017](#))

Uma das grandes influências do IoT é no campo do monitoramento do ambiente

físico em que vivemos, sistemas de alarmes e análise de dados ambientais (PERUMAL; SULAIMAN; LEONG, 2016). Este trabalho propõe a elaboração, planejamento e implementação de um sistema de baixo custo para monitoramento e economia de água no uso de chuveiros elétricos residenciais.

Segundo (Varela De Souza et al.,), a palavra "Domótica" resulta da junção da palavra latina "Domus", que significa casa, com "Robótica", que pode ser entendido como controle automatizado de algum processo ou equipamento, seu uso pode trazer significativas vantagens aos seus usuários como a otimização e gestão de recursos, praticidade e segurança, controle e monitoramento remoto dos dispositivos automatizados.

Compreender a evolução da tecnologia ajuda a entender como a Domótica evoluiu na forma como vivemos. O desenvolvimento de tecnologias de infraestrutura no início do século XX, como as redes de água e esgoto, gás encanado e eletricidade fizeram com que a residências se conectassem com o meio externo, tornando-se um nó de uma grande rede (FORTY, 2007). Com o advento da Internet, essa ligação se acentuou, permitindo ainda mais conectividade. (Varela De Souza et al.,)

1.1 Objetivos

Desenvolver um sistema modular, de baixo custo, baseado em microsserviços e no paradigma do IoT, para monitoramento e controle de consumo de água de chuveiros elétricos através da integração entre microprocessadores, microcontroladores, sensores e atuadores.

O sistema será capaz de armazenar e exibir dados vindos dos sensores de fluxo de água em sistemas remotos, conectados através da rede Wi-Fi, além de comandar um atuador para interromper o fornecimento da água.

1.1.1 Objetivos específicos

- Armazenar dados para levantamentos estatísticos;
- Implementar do sistema de identificação e controle de usuários;
- Implementar do sistema de interface;
- Implementar do sistema de atuação;
- Integrar todos os sistemas;
- Monitorar online dos parâmetros do sistema;

1.2 Justificativas e Relevância

Segundo ([Alves Da Silva; Gomes De Santana,](#)), o crescente consumo de água tem feito do uso consciente uma necessidade primordial. Essa prática deve ser considerada parte de uma atividade mais abrangente que é o uso racional da água, o qual inclui também, o controle de perdas e a redução do consumo de água.

Ao passar dos anos, os desperdícios da água utilizada atingem níveis nunca imaginados ([REBOUÇAS, 2003](#)). Ao se juntar o interesse e conhecimento em eletrônica e automação, será possível otimizar o monitoramento do gasto de água em residências e prédios, visando coletar dados para diminuir este tipo de desperdício. Este trabalho se justifica pela urgente necessidade de controle do uso da água em todas as esferas da sociedade.

1.3 Organização do trabalho

O presente trabalho está organizado em 5 Capítulos. No Capítulo 1 encontra-se a apresentação do problema e suas possíveis soluções, além de apresentar os objetivos propostos, que consistem no desenvolvimento de um sistema para domótica, baseado no IoT para monitorar o consumo de água, utilizando a arquitetura de microserviços e microcontroladores.

O Capítulo 2 consiste na revisão sobre os hardwares e softwares utilizados no projeto, encontram-se as definições e explicações dos mesmos.

No Capítulo 3 é apresentado a estrutura geral do sistema, a metodologia em que foi construído. Explica-se também as etapas do desenvolvimento e códigos implementados.

O Capítulo 4 nos fala sobre os testes do sistema e seus resultados.

No Capítulo 5 são abordadas as considerações finais do trabalho e os possíveis trabalhos futuros.

2 Materiais e Softwares

2.1 Microcontroladores e Microprocessadores

Para uma melhor eficiência no processamento de dados, na década de 70 começaram a ser utilizados microprocessadores em computadores (MARTINS, 2005). Os microprocessadores são componentes dedicados ao processamento de informações com capacidade de cálculos matemáticos e endereçamento de memória externa (CHASE; ALMEIDA, 2007).

Já os microcontroladores são pequenos sistemas computacionais poderosos que englobam em um único chip: interfaces de entrada/saída digitais e analógicas, memória RAM, memória FLASH, interfaces de comunicação serial, conversores analógicos/digitais e temporizadores/contadores. Com o advento dos microcontroladores de 16 e 32 bits (atualmente o padrão é de 8bits), a capacidade de gerenciar soluções mais complexas e maior velocidade de processamento se iguala ao do microprocessador. (CHASE; ALMEIDA, 2007).

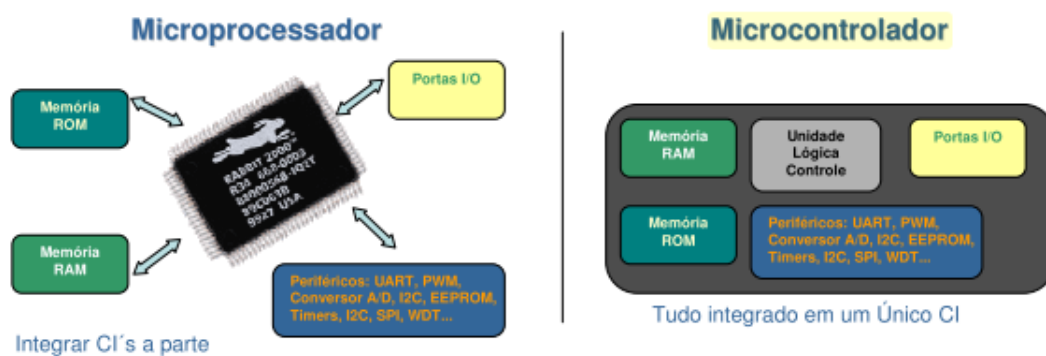


Figura 1 – Diferença entre microprocessador de microcontrolador

2.1.1 ESP8266

O ESP8266 é um circuito integrado, com interfaces de I/O digitais e analógicas e, interface Wi-Fi, com um processador de 32 bits, capaz de executar tarefas a 160 MHz.

Os módulos baseados no microcontrolador ESP8266 representam um grande avanço na relação de preço-recursos e pode ser um componente muito interessante para soluções IoT.(OLIVEIRA, 2017)

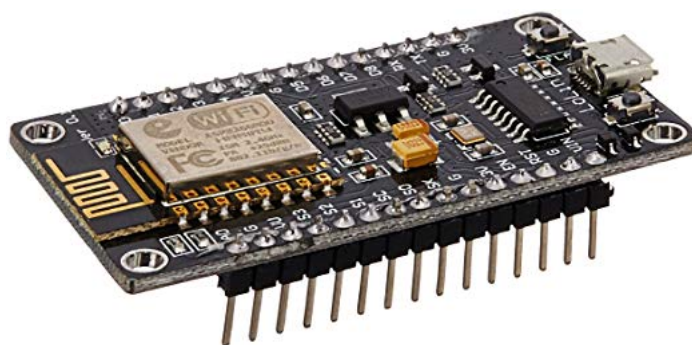


Figura 2 – ESP8266

2.1.2 RaspberryPi

Raspberry Pi é um minicomputador criado pela Raspberry Pi Foundation com o objetivo de estimular o ensino da ciência da computação nas escolas e universidades. Apesar de o Raspberry Pi possuir o hardware em uma única placa eletrônica de tamanho reduzido, seu potencial de processamento é significativo. O Raspberry Pi pode ser usado em diversos projetos tecnológicos, como experimentos remotos nos quais sua função é ser um micro servidor web. (CROTTI et al., 2013)



Figura 3 – RaspberryPi

2.2 Sensores e atuadores

2.2.1 Sensor de fluxo YF-S201

São sensores do tipo turbina que medem a quantidade de líquido que passa pela tubulação, girando uma turbina que gera pulsos de onda quadrada através de um sensor de efeito Hall (ROQUE; SABINO, 2018) O sensor usa esse efeito para enviar um sinal PWM e, através da contagem deste pulso é possível mensurar a quantidade de água que passa pelo cata-vento no interior do sensor, cada pulso mede aproximadamente 2,25 mm. (JÚNIOR; ARÊAS; SENA, 2017)



Figura 4 – Sensor de fluxo de água YF-S201

2.2.2 Teclado matricial de membrana

Como ([BLOG...](#),) definiu, teclados são geralmente utilizados em aplicações na qual o usuário precisar interagir com um sistema, como computadores, calculadoras, controles remotos entre outros. Segundo ([PORTAL...](#),) O Teclado Matricial de Membrana 4X4 com 16 teclas foi desenvolvido com a finalidade de facilitar a entrada de dados em projetos com plataformas microcontroladas. Este teclado possui 16 teclas, onde 10 teclas são numerais, 4 literais e 2 de caracteres. As 16 teclas estão dispostas em 4 linhas por 4 colunas e o teclado possui um conector de 8 pinos para ligação.

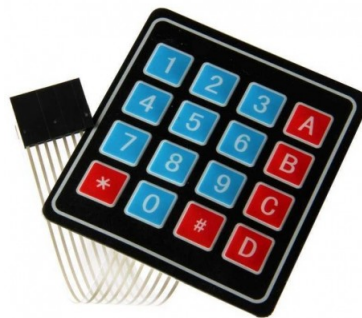


Figura 5 – Teclado Matricial de Membrana

Como o nome indica, este teclado é formado de botões organizados em linhas e colunas de modo a formar uma matriz. Quando pressionado, um botão conecta a linha com a coluna na qual está ligado.

O teclado matricial possui a seguinte pinagem:

- Pino 1 (Esquerda) – Primeira Linha (L1)
- Pino 2 – Segunda Linha (L2)
- ...

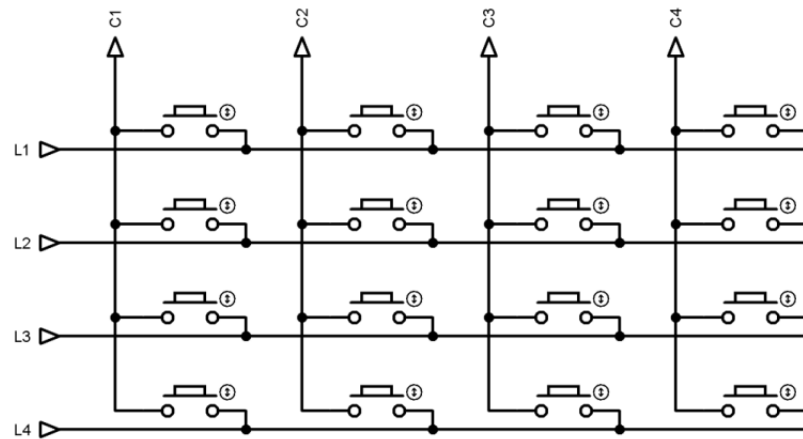


Figura 6 – Circuito do teclado

- Pino 5 – Primeira Coluna (C1)
- ...
- Pino 8 – Quarta Coluna (C4)

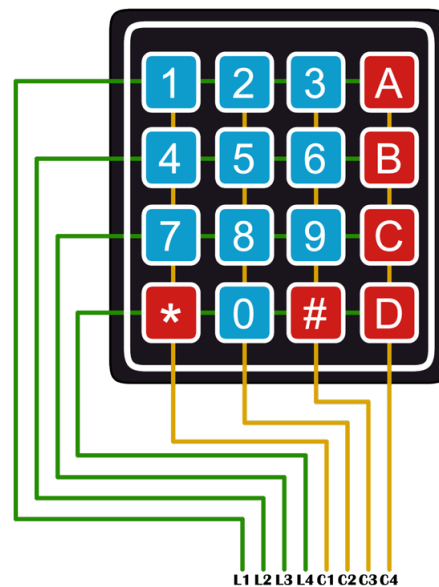


Figura 7 – Pinagem do teclado

2.2.3 Válvula solenoide

Seguindo a definição de (SILVA; LAGO, 2002), solenóides são dispositivos eletromecânicos baseados no deslocamento causado pela ação de um campo magnético gerado por uma bobina e são muito utilizados na construção de outros dispositivos, como é o caso das válvulas para controle de fluidos. Em particular, as válvulas para baixas vazões

(da ordem de mililitros por minuto) e baixas pressões têm sido amplamente aplicadas em equipamentos e montagens para uso em laboratórios clínicos e químicos. Elas são de pequenas dimensões e requerem baixa tensão e corrente de acionamento.

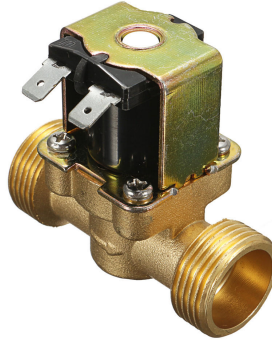


Figura 8 – Válvula Solenoide

Ainda segundo (SILVA; LAGO, 2002), a estratégia para fechamento e abertura dos canais fluídicos depende do fabricante, mas o princípio de acionamento elétrico é comumente o mesmo. Uma tensão de alguns volts é aplicada sobre um solenoide que faz com que um núcleo metálico ferromagnético se desloque, causando a alteração do estado da válvula. O núcleo ferromagnético comprime uma mola que é a responsável por deslocar o núcleo para sua posição original quando a corrente elétrica é interrompida.

2.3 Softwares

2.3.1 HomeAssistant

HomeAssistant é uma plataforma de automação escrita em Python. E, como (LUNDRIGAN et al., 2017) definem, inclui componentes contribuídos por usuários que permite a interface com dispositivos e Web Services. Em seu núcleo, HomeAssistant é um protocolo de mensageiria, facilitando a comunicação entre dispositivo e componentes funcionais na rede, provendo simples abstrações de componentes de automação residencial como sensores, câmeras, players de música, etc.

Seguindo a definição de (GOMES; SOUSA; VALE, 2018), o HomeAssistant tem suporte para diversos tipos de protocolos wireless, como BLE, ZigBee, Z-Wave e WiFi. Conta também com um RESTful API e suporta HTTP, MQTT, TCP sockets e componentes customizados. Estes componentes customizados permitem aos usuários a adicionar funções próprias ao HomeAssistant sem a necessidade de mudar o seu código fonte. Isto torna a integração de novos dispositivos e sensores muito mais fácil com o Home Assistant.

O HomeAssistant conta com uma grande comunidade de desenvolvedores com mais de 1.450 contribuidores e 23.700 estrelas no GitHub (HOME...,), o que significa uma

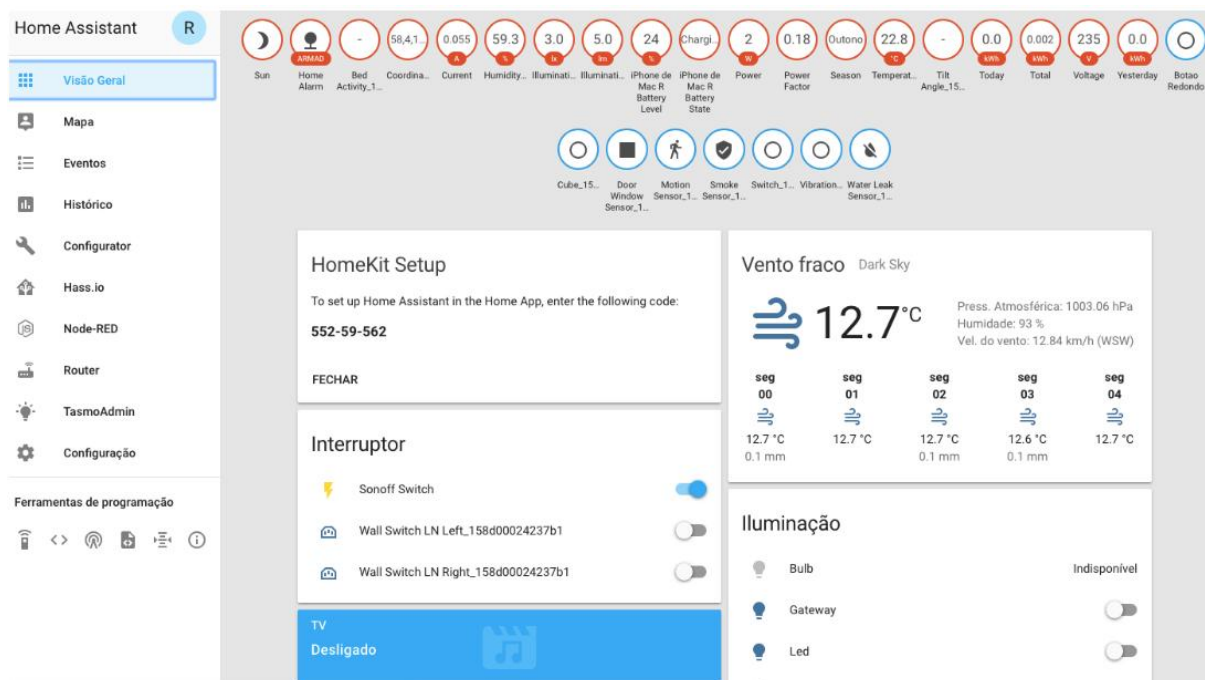


Figura 9 – Dashboard genérico do HomeAssistant

abundância em sua documentação sobre os seus componentes, fóruns e chats para conseguir ajuda de outros usuários, e diversos posts em blogs e vídeos sobre como começar a utilizar o programa.

Embora o Home Assistant tenha suporte a uma grande gama de protocolos wireless, este trabalho foca em sensores WiFi pois, ainda segundo (LUNDRIGAN et al., 2017):

- Hardware WiFi são mais baratos e palpáveis.
- Wi-Fi é o mais comum do que outros protocolos wireless.
- Sensores WiFi conseguem integrar com o resto da casa pois utiliza IP, tornando o sensor mais fácil de debugar e monitorar.

(Almeida Costa,) nos diz que o HomeAssistant pode ser instalado em qualquer sistema operacional, suportado pelo Python 3 e é muito pequeno e leve, o que é compatível para o uso no Raspberry Pi como um hub de automação pequeno e barato. É importante lembrar que o Home Assistant age apenas como uma central de controlo que pode informar outros serviços, como o Philips Hue ou o Nest, para realizar alguma função. O Home Assistant é barato e de fácil configuração.

2.3.2 Banco de dados de séries temporais

Como (NOOR et al., 2017) define, um Banco de Dados de Séries Temporais (TSDB) é um tipo de banco que é otimizado para dados que contém traços de tempo ou séries

temporais. É construído especificamente para lidar com métricas, eventos ou medidas que variam inclusive no tempo. Um TSDB é otimizado para medidas que mudam com o tempo, permite o usuário criar, enumerar, alterar, destruir e organizar várias séries temporais de um método mais eficiente. Atualmente, a maioria das empresas estão gerando uma quantidade gigante de dados sobre métricas e eventos que levam em consideração o tempo, mostrando que a necessidade de bancos de dados de séries temporais é inevitável.

Ainda segundo (NOOR et al., 2017), aplicações comuns para os TSDBs são IoT, DevOps, Análise de Dados, etc. Alguns casos de uso incluem monitoramento de sistemas de software como máquinas virtuais, monitoramento de sistemas físicos como algum equipamento, dispositivos conectados, o ambiente, sistemas de automação residencial, corpo humano, dentre outros.

2.3.2.1 InfluxBD

InfluxDB é o Banco de Dados de Séries Temporais usado neste projeto. Segundo (LUNDRIGAN et al., 2017), InfluxDB é projetado especificamente para dados do tipo séries temporais. O que combina perfeitamente com os dados coletados do sensor que será guardado no banco.

É um projeto *open-source* com o opcional de armazenamento em nuvem desenvolvido pela empresa InfluxData. É escrito na linguagem de programação Go e otimizado para lidar com dados de séries temporais. Provém de uma linguagem de consulta parecida com o SQL. (NOOR et al., 2017)

2.3.3 Banco de dados relacional

Segundo (BOSCARIOLI et al.,), um Banco de dados relacional possui uma coleção de tabelas, todas com nomes únicos, que compõem a base de dados, podendo estar relacionada a uma ou mais tabelas. Conceitos como integridade referencial de dados – que garantem que um dado referenciado em uma tabela esteja presente na tabela que está sendo referenciada – e chaves primárias estão presentes e garantem que um conjunto de informações possa ser representado de maneira consistente, independente da forma de acesso.

2.3.3.1 PostgreSQL

Segundo (STONES; MATTHEW, 2006) PostgreSQL é uma excelente implementação de um banco de dados relacional, cheio de funções, de código aberto e sem custos para o uso. PostgreSQL pode ser usado com a maioria das linguagens de programação existentes.

O PostgreSQL ganhou diversos prêmios, incluindo o *Linux Journal Editor's Choice Award for Best Database* três vezes e o *2004 Linux New Media Award for Best Database*

System.

2.3.4 Node.JS

Node.JS, também chamado de Node, é um ambiente de servidor que utiliza a linguagem de programação JavaScript. Como (TILKOV; VINOSKI, 2010) define, é baseado no runtime do Google, o chamado motor V8. O V8 e o Node são basicamente implementados em C e C++, focando na performance e baixo consumo de memória. Embora o V8 suporte principalmente o uso de JavaScript no navegador, o Node foca no suporte de processos de servidores.

Segundo (SAPES; SOLSONA, 2016), Node.JS utiliza um paradigma baseado em eventos e não-bloqueador de I/O, o que o torna leve e eficiente. É perfeito para aplicações de tempo real que lidam com dados intensos em dispositivos de baixo poder de processamento.

O Node é um dos ambientes e *frameworks* mais famosos que suportam o desenvolvimento de servidores utilizando o JavaScript. A comunidade criou um grande ecossistema de bibliotecas e vasta documentação que dão suporte ao Node. (TILKOV; VINOSKI, 2010)

2.3.4.1 Arquitetura de microsserviços

Seguindo a definição de (Moskovskii gosudarstvennyi universitet im. M.V. Lomonosova. Open Information Technologies Lab; SNEPS-SNEPPE, 2014), a arquitetura de microsserviços é uma abordagem no desenvolvimento de uma aplicação que baseia na existência de diversos pequenos serviços independentes. Cada um dos serviços deve rodar em seu próprio processo independente. Estes serviços podem comunicar entre si utilizando mecanismos leves de comunicação (geralmente em torno no HTTP). Os serviços devem ser absolutamente independentes.

Baseado em (PAHL; JAMSHIDI,), microsserviços é os resultados da decomposição funcional de uma aplicação. São caracterizados pela definição de sua interface e função no sistema. Como cada serviço deve ser independente, uma alteração na sua implementação não deve afetar o funcionamento dos demais.

2.3.5 Protocolo de comunicação

2.3.5.1 MQTT

MQTT significa Message Queuing Telemetry Transport, é um protocolo de transporte leve que otimiza o uso da largura de banda de rede.(MQTT..., a) O MQTT trabalha no TCP e garante a entrega de mensagens de um nó para um servidor. Sendo um protocolo orientado por troca de mensagens, MQTT é ideal para aplicações IoT, que tem recursos e capacidades limitados.

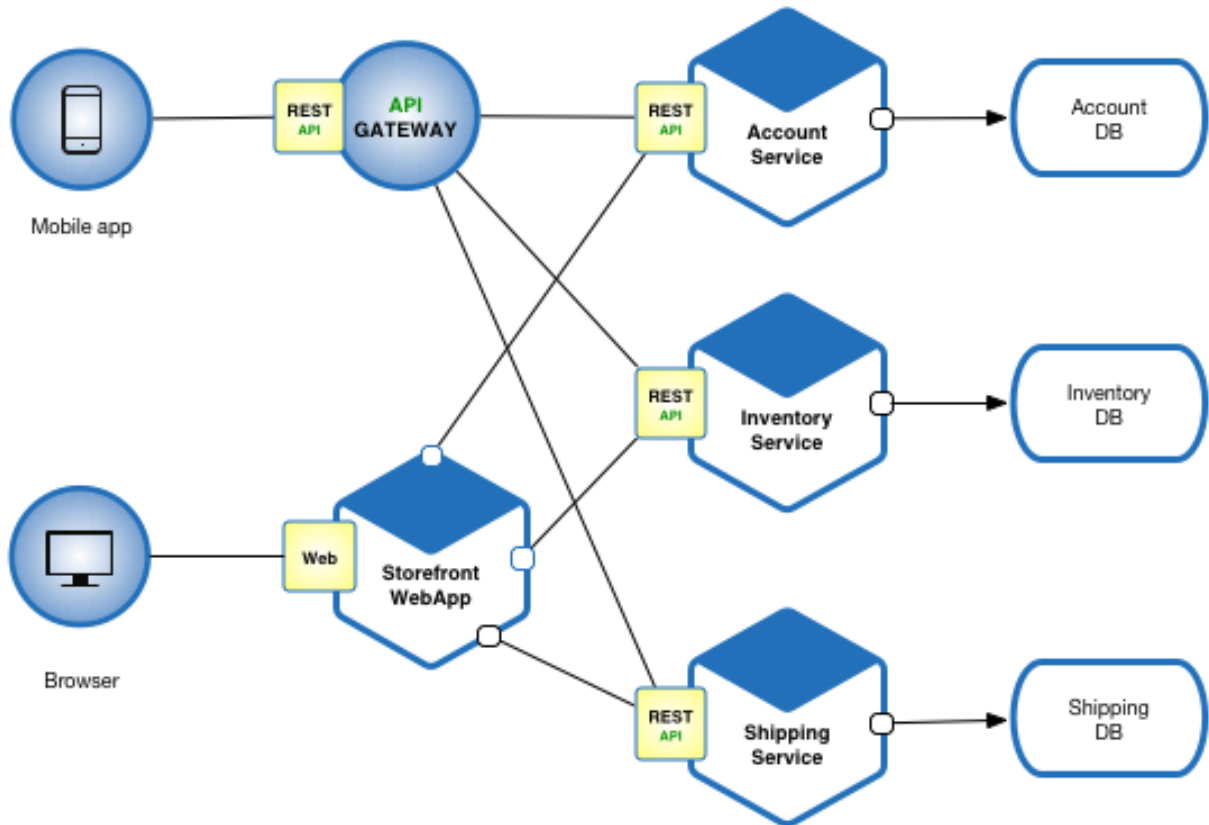


Figura 10 – Exemplo arquitetura de microsserviços

É um protocolo inicialmente desenvolvido pela IBM ([HIVEMQ](#),) em 1999, e recentemente foi reconhecido como padrão pela OASIS (Organization for the Advancement of Structured Information Standards).([MQTT...](#), b)

([KODALI; SORATKAL, 2017](#)) definiu o MQTT como um protocolo baseado em *publish/subscribe*. Qualquer conexão MQTT envolve dois tipos de agentes, os clientes MQTT e um MQTT *broker* público, ou servidor MQTT. Os dados que são transportados pelo MQTT são referenciados como mensagens da aplicação. Qualquer dispositivo ou programa que é conectado pela rede e troca mensagens através do MQTT é chamado de cliente MQTT. Um cliente MQTT pode ser tanto um *publisher* ou um *subscriber*. Um *publisher* publica mensagens e um *subscriber* requisita o recebimento de mensagens. Um MQTT *server* é um programa que interconecta os clientes MQTT. Ele aceita e transmite as mensagens através de múltiplos clientes conectados à ele.

Dispositivos como sensores, celulares, são considerados como clientes MQTT, quando um cliente MQTT tem alguma informação para transmitir, ele publica o dado para o *broker* MQTT.

O *broker* MQTT, ou servidor MQTT, responsável por coletar e organizar os dados. As mensagens publicadas por clientes MQTT são transmitidas para outros clientes MQTT que se inscreverem ao tópico. O MQTT é desenhado para simplificar a implementação no

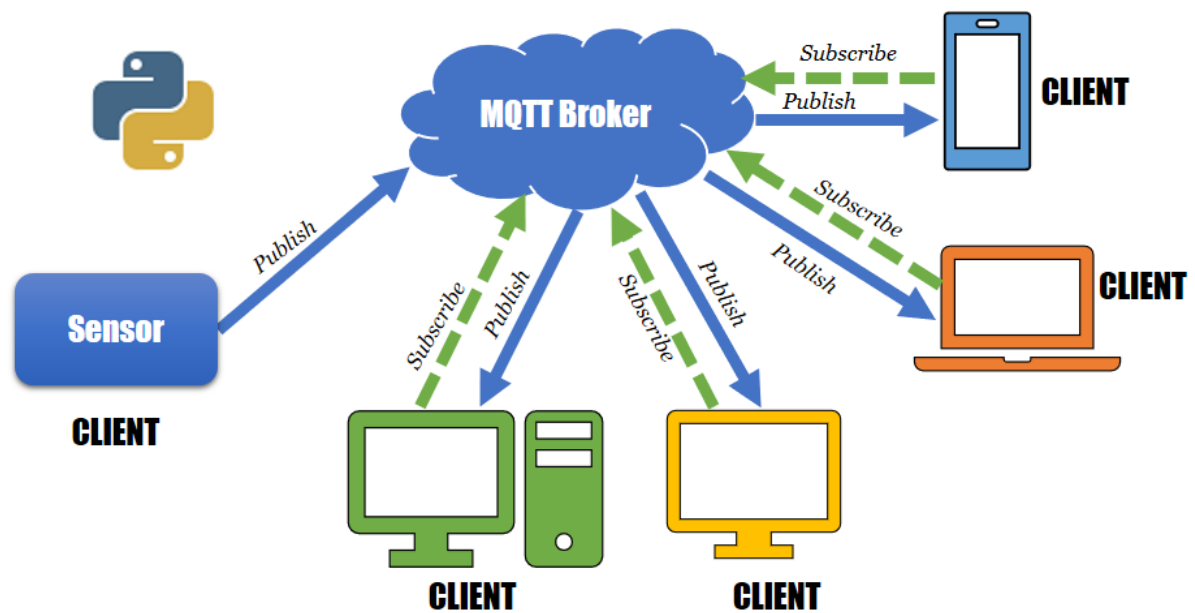


Figura 11 – Exemplo da arquitetura do MQTT

cliente por concentrar todas as complexidades no *broker*. Os *publishers* e *subscribers* são isolados, o que significa que eles não precisam conhecer a existência do outro.

2.3.6 MQTT Mosquitto

O Mosquitto é um *broker* MQTT de código aberto (KODALI; SORATKAL, 2017) que entrega uma implementação padrão de servidor e cliente MQTT. Utiliza o modelo *publish/subscribe*, tem uma baixa utilização de rede e pode ser implementado em dispositivos de baixo custo como microcontroladores que podem ser usados em sensores remotos de IoT. (LIGHT,)

Segundo (LIGHT,), Mosquitto é recomendado para o uso sempre em que se necessita de mensagens leves, particularmente em dispositivos com recursos limitados.

O Projeto Mosquitto é um membro da Eclipse Foundation. Existem três partes no projeto:

- O servidor principal Mosquitto.
- Os clientes mosquitto *pub* e mosquitto *sub*, que contém ferramentas para se comunicar com o servidor MQTT.
- Uma biblioteca cliente MQTT, escrita em C.

2.3.7 Docker

Falar sobre o Docker

2.3.8 Grafana

Falar sobre o Grafana.

3 Metodologia

Este Capítulo apresenta a estrutura geral do sistema, a configuração inicial do RaspberryPi, a construção dos microsserviços, assim como as etapas de realização do projeto. A primeira etapa consiste no desenvolvimento do microsserviço de cadastro e controle de usuários. Em seguida, foi desenvolvido o sistema que recebe os dados via MQTT, guardando-os no InfluxDB. Com os dados sendo recebidos e devidamente guardados, foi configurado o HomeAssistant juntamente com o Grafana. Para facilitar os testes, também foi desenvolvido um sistema que simula os dados que seriam recebidos via sensor de fluxo.

3.1 Configuração do RaspberryPi

Os sistemas deste projeto foram desenvolvidos diretamente no RaspberryPi, e, para possibilitar este desenvolvimento, é necessário primeiramente configurar o ambiente do RaspberryPi.

O RaspberryPi foi configurado para ser *headless*, o que faz com que ele não necessite de teclado, mouse ou monitor para poder ser acessado. Para conseguir acessar o Raspberry nesta configuração é necessária a utilização do SSH, ou *Secure Shell*, que é um protocolo de rede criptográfico para operação de serviços de rede de forma segura(COLOCAR LINK NO RODAPÉ). O SSH permite o login remoto no sistema operacional do Raspberry, possibilitando o completo controle do sistema operacional de forma remota.

3.2 Microsserviços

Aqui será apresentado as etapas de desenvolvimento dos microsserviços utilizados no sistema elaborado, assim como o serviço de simulação para facilitar os testes e validação do sistema.

3.3 Criação das tabelas no PostgreSQL

INFORMAÇÕES SOBRE AS TABELAS DO POSTGRESQL

3.4 Configuração do InfluxDB

FALAR SOBRE A CONFIGURAÇÃO DO INFLUXDB

3.4.1 Sistema de usuários

O propósito desta aplicação está em ter o controle e informações sobre o consumo de água em residências, para isso, é necessário ter um meio para conseguir as informações sobre os usuários do sistema.

O microserviço de usuários é responsável por cadastrar e editar usuários, cadastrar informações sobre o tempo de banho do usuário e autorizar o usuário à tomar o banho.

As operações são realizadas através dos *endpoints*, que são *links* em que são passados parâmetros contendo a informação específica que o *endpoint* requer.

Este microserviço salva todos os dados recebidos nas tabelas do PostgreSQL.

O sistema possui os seguintes *endpoints*:

- */cadastrar*: possibilita cadastrar o usuário com as informações do nome, senha e tempo de banho permitido.
- */autorizar*: recebe o id do usuário e a senha, compara a senha enviada com a senha cadastrada e retorna se o usuário está ou não autorizado.
- */editar-tempo*: possibilita editar o tempo de banho permitido do usuário.
- */editar-senha*: possibilita editar a senha do usuário.
- */banho*: salva no banco de dados informações do banho, o tempo e o usuário que tomou o banho.
- */banho/:id*: retorna informações sobre todos os banhos do usuário.

3.4.2 Sistema de comunicação MQTT

Este microserviço é responsável pelo recebimento e manipulação dos dados recebidos dos sensores e do teclado numérico via MQTT. Também é responsável por comunicar com o sistema de usuários para autorizar o usuário, por ligar ou desligar o atuador, além de salvar os dados no InfluxDB.

O sistema de comunicação recebe informações sobre as teclas digitadas e comunica com o sistema de usuários, para autorizar ou não o ligamento do atuador, que significa o início do banho.

O sistema se inscreve nos seguintes tópicos do *broker* MQTT para receber as informações:

- *keys*: é o tópico em que contém a tecla digitada no teclado numérico
- *actuator*: é o tópico para enviar informações do atuador, para ligá-lo ou desligá-lo

- *user*: é o tópico para enviar informações do usuário, como o nome e se ele está autorizado ou não.
- *sensor*: é o tópico para enviar informações do sensor de fluxo.

3.5 Configuração do HomeAssistant

FALAR SOBRE A CONFIGURAÇÃO DO HOMEASSISTANT

3.6 Configuração do Grafana

FALAR SOBRE A CONFIGURAÇÃO DO GRAFANA

3.7 Configuração do Docker

FALAR SOBRE A CONFIGURAÇÃO DO DOCKER

4 Experimentos e Resultados

5 Considerações Finais

Referências

- Almeida Costa, R. *Instituto Politécnico de Viseu*. [S.l.]. Citado na página 18.
- Alves Da Silva, M.; Gomes De Santana, C. *REUSO DE ÁGUA: possibilidades de redução do desperdício nas atividades domésticas*. [S.l.]. Disponível em: <<https://www.tratamentodeagua.com.br/wp-content/uploads/2016/05/REUSO-DE-ÁGUA-possibilidades-de-redução-do-desperdício-nas-a>>. Citado na página 12.
- BLOG Masterwalker. <http://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-teclado-matricial-de-membrana-4x4/>. Accessed: 2019-06-02. Citado na página 15.
- BOSCARIOLI, C. et al. *Uma reflexão sobre Banco de Dados Orientados a Objetos*. [S.l.]. Citado na página 19.
- CHASE, O.; ALMEIDA, F. Sistemas embarcados. *Mídia Eletrônica. Página na internet*: <www.sbjovem.org/chase>, capturado em, 2007. v. 10, n. 11, p. 13, 2007. Citado na página 13.
- CROTTI, Y. et al. Raspberrypi e experimentação remota. In: *ICBL2013. International Conference on Interactive Computer aided Blended Learning*. [S.l.: s.n.], 2013. Citado na página 14.
- DIÁRIAS, A. et al. ANÁLISE DE CONSUMO E DESPERDÍCIO DE ÁGUA EM. 2007. v. 3, p. 0–5, 2007. Citado na página 10.
- FERREIRA, H. S.; HEROSO, L. F.; ZALESKI, R. H. Sistema de monitoramento de consumo de água doméstico com a utilização de um hidrômetro digital. Citado na página 10.
- FORTY, A. *Objetos de desejo*. [S.l.]: Editora Cosac Naify, 2007. Citado na página 11.
- GOMES, L.; SOUSA, F.; VALE, Z. An agent-based IoT system for intelligent energy monitoring in buildings. *IEEE Vehicular Technology Conference*, 2018. IEEE, v. 2018-June, p. 1–5, 2018. ISSN 15502252. Citado na página 17.
- HIVEMQ. <http://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt>. Accessed: 2019-05-26. Citado na página 21.
- HOME Assistant. <https://github.com/home-assistant/home-assistant>. Accessed: 2019-05-26. Citado na página 17.
- JÚNIOR, J. M. M. L.; ARÊAS, R. C.; SENA, A. J. C. Automação residencial: Monitoramento de consumo de energia elétrica e água. *INOVA TEC*, 2017. v. 1, 2017. Citado na página 14.

- KODALI, R. K.; SORATKAL, S. R. MQTT based home automation system using ESP8266. *IEEE Region 10 Humanitarian Technology Conference 2016, R10-HTC 2016 - Proceedings*, 2017. IEEE, p. 1–5, 2017. Citado 2 vezes nas páginas 21 e 22.
- LIGHT, R. A. Mosquitto: server and client implementation of the MQTT protocol. Disponível em: <<https://www.theoj.org/joss-papers/joss.00265/10.21105.joss.00265.pdf>>. Citado na página 22.
- LUNDRIGAN, P. et al. *EpiFi: An In-Home Sensor Network Architecture for Epidemiological Studies*. [S.l.], 2017. Disponível em: <<https://arxiv.org/pdf/1709.02233.pdf>>. Citado 3 vezes nas páginas 17, 18 e 19.
- MARTINS, N. A. Sistemas microcontrolados. *Uma abordagem com o Microcontrolador PIC 16F84*. Editora Novatec Ltda, 1ª edição, 2005. 2005. Citado na página 13.
- Moskovskii gosudarstvennyi universitet im. M.V. Lomonosova. Open Information Technologies Lab, D.; SNEPS-SNEPPE, M. *International journal of open information technologies*. [S.l.: s.n.], 2014. 24–27 p. ISSN 2307-8162. Citado na página 20.
- MQTT v3.1 protocol specification. <http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>. Accessed: 2019-05-26. Citado na página 20.
- MQTT version 3.1.1 becomes an oasis standard. <https://www.oasis-open.org/news/announcements/mqtt-version-3-1-1-becomes-an-oasis-standard>. Accessed: 2019-05-26. Citado na página 21.
- NOOR, S. et al. *Time Series Databases and InfluxDB*. [S.l.], 2017. Citado 2 vezes nas páginas 18 e 19.
- OLIVEIRA, S. de. *Internet das Coisas com ESP8266, Arduino e Raspberry Pi*. [S.l.]: Novatec Editora, 2017. Citado na página 13.
- PAHL, C.; JAMSHIDI, P. *Microservices: A Systematic Mapping Study*. [S.l.: s.n.]. ISBN 9789897581823. Citado na página 20.
- PERUMAL, T.; SULAIMAN, M. N.; LEONG, C. Y. Internet of Things (IoT) enabled water monitoring system. *2015 IEEE 4th Global Conference on Consumer Electronics, GCCE 2015*, 2016. IEEE, p. 86–87, 2016. Citado 2 vezes nas páginas 10 e 11.
- PORTAL Vida de Silício. <https://portal.vidadesilicio.com.br/teclado-matricial-e-multiplexacao/>. Accessed: 2019-06-02. Citado na página 15.
- REBOUÇAS, A. d. C. Água no brasil: abundância, desperdício e escassez. *Bahia análise & dados*, 2003. v. 13, p. 341–345, 2003. Citado na página 12.
- Risteska Stojkoska, B. L.; TRIVODALIEV, K. V. A review of Internet of Things for smart home: Challenges and solutions. 2017. 2017. Disponível em: <<http://dx.doi.org/10.1016/j.jclepro.2016.10.006>>. Citado na página 10.
- ROQUE, I. R.; SABINO, J. H. M. Sistema de diluição de líquidos concentrados e envase automático. 2018. 2018. Citado na página 14.

- SAPES, J.; SOLSONA, F. Fingerscanner: Embedding a fingerprint scanner in a raspberry pi. *Sensors (Switzerland)*, 2016. v. 16, n. 2, p. 1–18, 2016. ISSN 14248220. Citado na página 20.
- SILVA, J. A. F. da; LAGO, C. L. do. Módulo eletrônico de controle para válvulas solenóides. *Química Nova*, 2002. SciELO Brasil, v. 25, n. 5, p. 842–843, 2002. Citado 2 vezes nas páginas 16 e 17.
- STONES, R.; MATTHEW, N. *Beginning databases with PostgreSQL: from novice to professional*. [S.l.]: Apress, 2006. Citado na página 19.
- TILKOV, S.; VINOSKI, S. Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 2010. v. 14, n. 6, p. 80–83, 2010. ISSN 10897801. Citado na página 20.
- Varela De Souza, M. et al. *UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE INSTITUTO METRÓPOLE DIGITAL PÓS-GRADUAÇÃO EM ENGENHARIA DE SOFTWARE Domótica de baixo custo usando princípios de IoT*. [S.l.]. Citado na página 11.
- ÁGUA e consumo consciente. <http://www.brasil.gov.br/noticias/educacao-e-ciencia/2010/10/agua-e-consumo-consciente>. Accessed: 2018-12-05. Citado na página 10.

Apêndices

Anexos