

Desenvolvimento e implementação de um sistema distribuído aberto para gerenciamento de estações meteorológicas

João Victor Lisboa Porcel¹, Kleber Manrique Trevisani¹

¹Instituto Federal de Ciência e Tecnologia – Campus Presidente Epitácio (IFSP)
R. José Ramos Júnior, 27-50 – Presidente Epitácio – SP – Brasil

joaovictorlisboaporcel@hotmail.com, kleber@ifsp.edu.br

Abstract. *The study of climate has become increasingly important for a range of human activities. In this sense, the monitoring of meteorological variables is necessary to determine predictive, adaptive and corrective actions. In this context, automatic weather stations can be used for more accurate monitoring of the climate in each region. This article describes the development and implementation of an open and distributed system to manage automatic weather stations, with its main objective facilitating access to data collected from these stations so that they can be used for future research.*

Resumo. *O estudo do clima tem se tornado cada vez mais importante para uma gama de atividades humanas. Nesse sentido, o monitoramento de variáveis meteorológicas é necessário para determinar ações preditivas, adaptativas e corretivas. Nesse contexto, estações meteorológicas automáticas podem ser utilizadas para um monitoramento mais preciso do clima de cada região. Este artigo descreve o desenvolvimento e a implementação de um sistema aberto e distribuído para gerenciar estações meteorológicas automáticas, sendo seu objetivo principal facilitar o acesso aos dados coletados destas estações de forma que possam ser utilizados para pesquisas futuras.*

1. Introdução

A Meteorologia é a área da ciência que se responsabiliza pelo estudo da atmosfera da Terra, tendo como principal objetivo o estudo de fenômenos que ocorrem na atmosfera e as interações entre seus estados dinâmicos, físico e químico com os sistemas da litosfera, hidrosfera, criosfera e biosfera, sendo parte das Ciências Ambientais (YNOUE et al., 2017).

O estudo do clima é importante para uma gama de atividades humanas, como por exemplo, aviação, navegação marítima, agricultura, dentre outras atividades. Frequentemente, o clima pode mudar muito rapidamente em um determinado momento numa determinada área. As mudanças climáticas são causadas por mudanças na temperatura, umidade do ar e pressão do ar. Logo, as previsões do tempo são necessárias para prever as condições da atmosfera para um local e horário específicos (NOVIANTY et al., 2019).

Devido às constantes mudanças climáticas e do tempo, o monitoramento das variáveis meteorológicas para o estudo sobre o efeito dessas alterações climáticas se tornou necessário para elaboração de ações preditivas, adaptativas e corretivas. Nesse contexto, são utilizadas diversas abordagens, sendo o uso de imagens de satélites o método mais amplo e conhecido, permitindo a análise de massas de ar, temperatura e a

possibilidade de chuvas. Contudo, para um monitoramento mais preciso de cada região, podem ser utilizadas estações meteorológicas automática, que além de oferecer as variáveis climáticas já citadas, podem fornecer dados sobre materiais particulados como a fuligem, dióxido e monóxido de carbono, direção e velocidade do vento, radiação solar, entre outros (SILVA; FRUETT, 2013).

Considerando o exposto, este trabalho tem como objetivo desenvolver e implementar um sistema de informação distribuído e aberto para gerenciamento de estações meteorológicas automáticas (EMA) de forma a facilitar o acesso aos dados coletados por essas estações. Este sistema será aberto para estações que utilizem o mesmo protocolo de comunicação selecionado/definido.

O sistema desenvolvido deve permitir acesso às informações coletadas pelas referidas estações em tempo real, possibilitando o conhecimento da situação climática atual de um determinado local, bem como, das informações meteorológicas armazenadas ao longo do tempo, via sistema de gerenciamento Web, permitindo a realização de estudos mais aprofundados sobre o comportamento do clima local. É importante ressaltar que nenhuma estação meteorológica será desenvolvida, pois o trabalho em questão fará uso de estações meteorológicas automáticas simuladas, provavelmente implementadas em máquinas virtuais.

O texto está organizado da seguinte forma: A segunda seção apresenta os trabalhos relacionados ao tema e que auxiliaram no desenvolvimento do trabalho em questão, destacando pontos importantes sobre o desenvolvimento de estações meteorológicas automáticas, arquiteturas de software e protocolos de comunicação utilizados. A terceira seção descreve detalhes sobre estações meteorológicas automáticas, considerando questões sobre os tipos de dados coletados e alguns aspectos de hardware dessas estações. A quarta seção apresenta questões sobre sistemas distribuídos no âmbito de IoT, onde são apresentados conceitos de sistemas distribuídos e desafios que aplicações IoT precisam superar. A quinta seção apresenta o projeto do sistema. Nela é apresentada a arquitetura proposta para este trabalho, as tecnologias selecionadas para a implementação e o protótipo do gerenciador Web. Finalmente, a sexta seção apresenta as considerações finais do autor a respeito do trabalho onde são abordados o estado de desenvolvimento atual do trabalho, quais os próximos passos a serem realizados e as eventuais alterações de cronograma e objetivos necessárias.

2. Trabalhos Relacionados

Esta seção descreve alguns trabalhos relacionados ao tema em questão com o objetivo de auxiliar e direcionar o desenvolvimento deste trabalho. Nesse contexto, são apresentados exemplos de estações meteorológicas automáticas, arquiteturas de sistemas similares que podem ser utilizadas no todo ou em parte neste trabalho e protocolos de comunicação utilizados nesses tipos de sistemas.

2.1 Estações Meteorológicas Automáticas

O trabalho de Finholdt et al. (2011), consiste no desenvolvimento de uma estação

meteorológica automática para o manejo de irrigação com tecnologia nacional.

Em termos de hardware a EMA desenvolvida possui a capacidade de armazenar os dados na memória do equipamento a até um ano, possui a disponibilidade de dados diários apresentados por meio do *display* e dados horários mediante a retirada da base de dados por meio de uma conexão acessível para o microcomputador. A aquisição dos dados também é efetuada com auxílio do microcomputador e utiliza o protocolo de comunicação RS-232.

Os sensores do equipamento desenvolvido foram calibrados com base em metodologias recomendadas pela *World Meteorological Organization* (WMO) para os sensores de vento, chuva, umidade relativa, irradiância solar e temperatura. O equipamento também atende aos requisitos técnicos e operacionais para o uso na agricultura, tendo atingido os objetivos de autossuficiência energética, manipulação de dados com e sem o uso de um microcomputador, a facilidade de manutenção e a simplicidade de uso.

A arquitetura proposta para o desenvolvimento da estação descrita no trabalho de Finholdt et al. (2011), foi organizada em um diagrama de blocos ilustrado pela Figura 1, em que cada um representa uma parte do sistema, totalizando oito blocos. O bloco principal seria a “Central de Processamento” que ficou responsabilizada por gerenciar todas as funções do equipamento, unindo as entradas e saídas do sistema. O bloco de “Energia”, promove autossuficiência energética ao sistema por meio do uso de um painel solar. “Sensores” é responsável pelo envio dos sinais elétricos que representam os fenômenos meteorológicos catalogados. “Teclado” envia para a Central de Processamento as operações desejadas pelo usuário. “Relógio” é responsável por informar à central o momento exato de executar as operações. “Memória” é o bloco que armazena os dados, sendo também a fonte de fornecimento de dados históricos do equipamento. “Comunicação com PC” é utilizado para fazer a leitura dos dados históricos e configurar o equipamento de forma geral. Por fim, o bloco “Display” apresenta os dados solicitados pelo usuário.

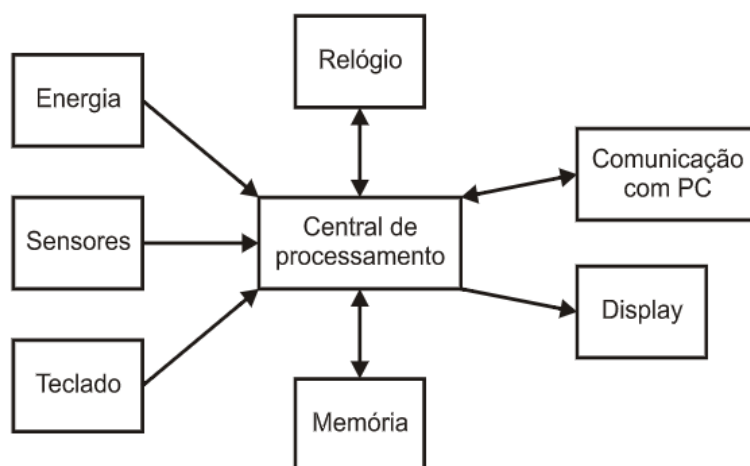


Figura 1. Diagrama simplificado da EMA (FILHOLDT et al., 2011).

Para captar as variáveis meteorológicas, a EMA proposta por Finholdt et al.

(2011), possui sensores analógicos e digitais. O higrômetro, piranômetro e termômetro são sensores analógicos e o anemômetro e o pluviômetro são sensores digitais. As especificações dos componentes de hardware mais importantes do referido trabalho são apresentadas pela Tabela 1. A Figura 2 ilustra alguns tipos de sensores utilizados no referido trabalho.

Tabela 1. Componentes eletrônicos de destaque (elaborado pelo autor).

Função	Nome do Componente	Fornecedor	Observações Adicionais
Sensor de Temperatura	LM35CAZ com encapsulamento plástico TO-92	<i>National Semiconductor</i>	Uma tensão proporcionalmente linear à temperatura em °C, sendo igual a +10,0 mV/°C.
Higrômetro	HIH-4000-004 com encapsulamento SIP	<i>Honeywell</i>	A tensão de alimentação foi ajustada por um regulador de tensão, sendo de 4,096 V
Regulador de Tensão	MCP1541	<i>Microchip</i>	Possui uma calibração de fábrica com grande exatidão em resposta, o erro máximo é de +- 3,5%. Também foi necessário um filtro no sinal de saída para estabilizar a resposta do sensor. As equações para conversão de sinal são especificadas pelo fabricante.
Piranômetro	BPW20R com encapsulamento TO-5	<i>Vishay Semiconductors</i>	
Pluviômetro			É um cilindro de aço inox AISI 430 que sustenta todo o equipamento, a base dele tem como função

			<p>suportar o sensor de acrílico (báscula), além de fechar o sistema pelo lado inferior. Acima do cilindro foi colocado o funil de alumínio para realizar a coleta de água da chuva.</p>
Anemômetro de Concha			<p>É um cilindro torneado com furo no centro, por dentro do cilindro passa um eixo que sustenta as conchas que são espaçadas em um ângulo de 120° cada. O eixo é fixado em dois rolamentos com o objetivo de propiciar maior sensibilidade ao sensor magnético <i>red switch</i>.</p>
Microcontrolador	PIC16F877A-I/P	<i>Microchip Technology Inc.</i>	<p>Compõe o bloco da Central de Processamento. Possui uma memória de 14,3 kbytes.</p>
Memória EEPROM	24LC10125-I/P	<i>Microchip Technology Inc.</i>	

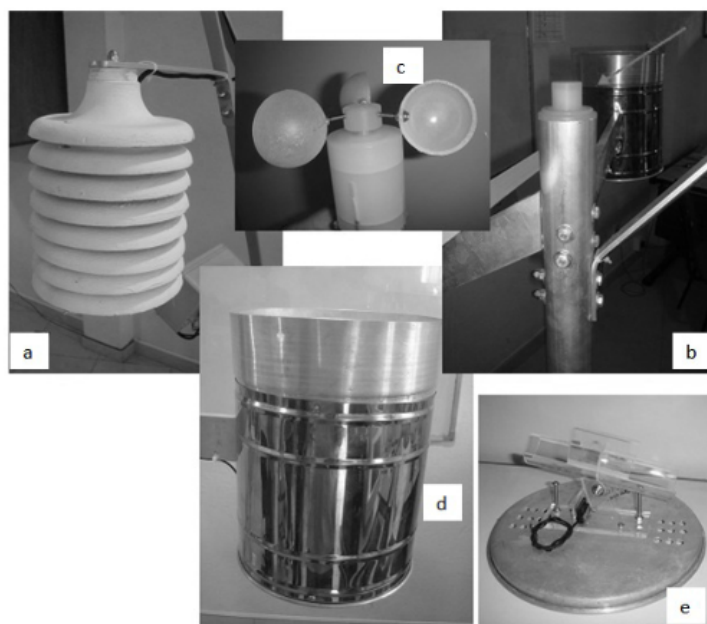


Figura 2. Sensores da estação desenvolvida por (FILHOLDT et al., 2011). Abrigo termométrico (a); Piranômetro (b); Anemômetro (c); Pluviômetro (d) com detalhe da báscula (e).

O trabalho de Finholdt et al. (2011) relata que apesar de ter atendido aos objetivos propostos, ainda é um protótipo, e devido a isso não é uma substituição imediata aos equipamentos similares já existentes no mercado, faltando para tanto testes em laboratórios de metrologia acreditados nacionalmente e internacionalmente para certificar seu funcionamento. Diferente deste trabalho, o projeto de FILHOLDT et al. (2011) não inclui a ideia do desenvolvimento de um software para gerenciamento de várias estações meteorológicas, pois está focado no desenvolvimento da EMA em si. Contudo, o conhecimento das capacidades e funcionalidades de uma EMA é importante para projetar o software de gerenciamento dessas estações.

O trabalho de Pont, Santos e Uggioni (2021) tem por objetivo desenvolver uma estação meteorológica automática baseada em Internet das Coisas (IoT) abastecida por energia solar tendo como objetivo contribuir com informações meteorológicas locais. A estação proposta é capaz de medir cinco variáveis e integra quatro subsistemas, além disso, ela efetua a transmissão dos dados sem fio e os armazena em um banco de dados local para posteriormente apresentar seus dados por meio de uma aplicação web.

O trabalho ainda apresenta o uso de protocolos específicos para a transmissão de dados, como por exemplo, o protocolo MQTT que é utilizado para a troca de mensagens entre dois ou mais dispositivos utilizando o modelo de comunicação publicação e assinatura (publish/subscribe) e o HTTP que é utilizado para transmitir dados entre um cliente e um servidor utilizando requisições e respostas (request/response).

A arquitetura do trabalho de Pont, Santos e Uggioni (2021) é organizada em Subsistemas da EMA, Arquitetura de Hardware e Arquitetura de Software, conforme ilustrado pela Figura 3.

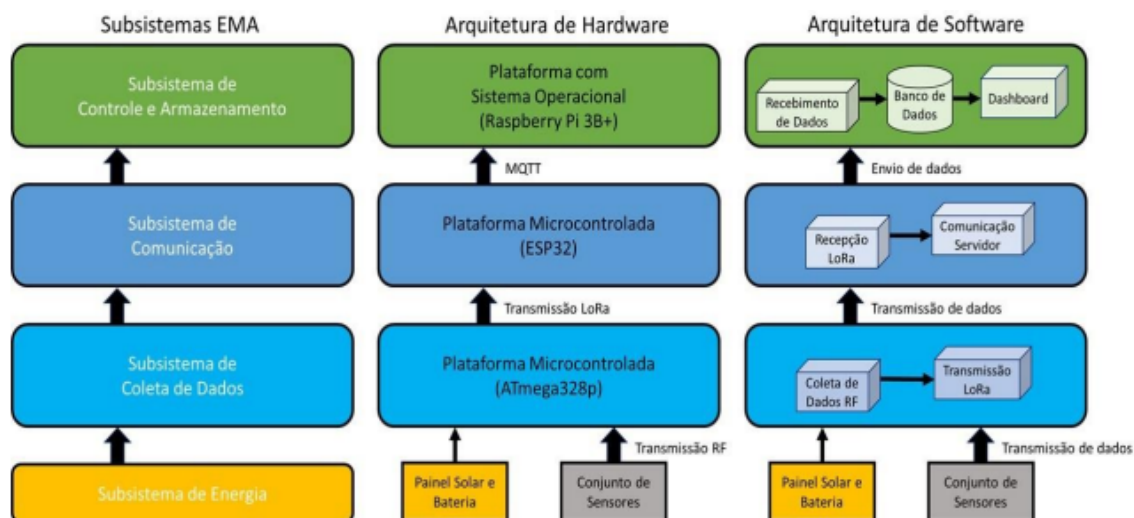


Figura 3. Arquiteturas definida por (PONT, SANTOS e UGGIONI, 2021).

Os Subsistemas da EMA contam com os Subsistemas de Energia, de Coleta de Dados, de Comunicação e de Controle e Armazenamento. O Hardware possui um Painel Solar, Bateria e um conjunto de sensores que ficam conectados à Plataforma Microcontrolada que efetua uma transmissão para outra Plataforma Microcontrolada que por sua vez utiliza o protocolo MQTT para transmitir os dados à Plataforma com o Sistema Operacional da placa Raspberry Pi. A Arquitetura de Software conta com a transmissão dos dados coletados sendo transmitidos para o servidor e o envio desses dados ao banco de dados que por sua vez será utilizado pela dashboard desenvolvida no projeto.

O trabalho de Pont, Santos e Uggioni (2021) se assemelha ao trabalho proposto, pois também propõe a implementação de um sistema de gerenciamento que pode se comunicar com diversas EMAs possibilitando a visualização e análise dos dados de forma centralizada.

Munandar et al. (2017) propuseram o desenvolvimento de um sistema de monitoramento de clima em tempo real em um aplicativo móvel utilizando uma EMA.

No projeto, é proposto uma EMA com sensores pluviométrico, de temperatura, umidade, radiação solar e velocidade do vento. Todos esses sensores foram colocados no topo de um edifício e os dados coletados desses sensores eram enviados para um servidor que ficava localizado no segundo andar deste mesmo prédio.

Os dados eram recebidos em dois formatos, imagens que poderiam ser utilizadas para indicar o estado do clima e servir como dados históricos e arquivos HTX que foram utilizados para o desenvolvimento da interface de usuário. Cada um dos arquivos HTX recebidos eram posteriormente convertidos para um arquivo PHP. Os arquivos de imagem e PHP eram redirecionados para o servidor Web por meio de FTP, com um intervalo de 5 minutos. Em seguida os dados eram lidos por uma aplicação mobile que foi desenvolvida com uma plataforma chamada AppGyver, que é utilizada para

desenvolver a interface da aplicação mobile. A arquitetura de software utilizado pelo projeto em questão é ilustrada pela Figura 4.

O trabalho proposto por Munandar et al. (2017) tem objetivos semelhantes ao do trabalho proposto, Contudo, não se preocupa em desenvolver uma estrutura aberta que permite catalogar diversos dados de diversas EMA.

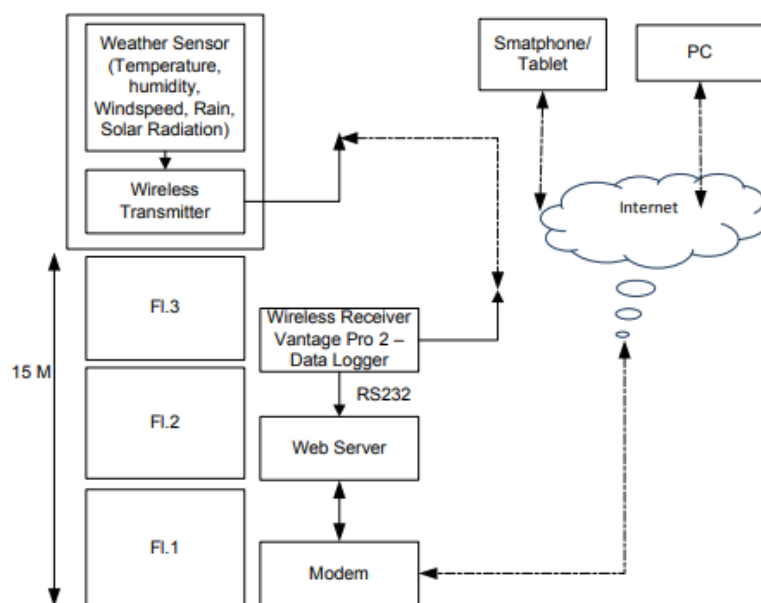


Figura 4. Arquitetura proposta por (MUNANDAR et al., 2017).

2.2 Arquiteturas

O trabalho de Sobin (2020), apresenta diversas arquiteturas e protocolos utilizados em sistemas IoT, além de propor taxonomias adequadas para classificá-los, também são discutidos aspectos técnicos como a segurança/privacidade, interoperabilidade, escalabilidade e eficiência energética. Discorre também sobre o vasto campo de aplicações que as tecnologias IoT podem ser aplicadas.

Sobin (2020) também cita que a organização do software pode ser fator decisivo para possibilitar a interconexão de dispositivos de IoT, considerando sua diversidade de tecnologias e características.

Atualmente existem muitas arquiteturas que estão propostas para aplicações de IoT. Segundo Sobin (2020), as arquiteturas atuais que fornecem a comunicação entre dispositivos distintos, também fornecem modelos inadequados e não abordam questões de segurança e privacidade e problemas de escalabilidade, tornando-se necessário uma arquitetura unificadora para IoT.

O trabalho de Carmo et al. (2022), apresenta uma arquitetura de micro estações meteorológicas que tem por objetivo fornecer aos agricultores informações sobre questões relacionadas à irrigação, como por exemplo, a quantidade de umidade da terra

atual e ao longo de um período. Dessa forma, facilita a tomada de decisões que facilitam a definição da área mais propícia para o plantio de determinada cultura e auxilia na economia de água, energia elétrica e outros recursos.

A arquitetura proposta por Carmo et al. (2022) prevê sensores que se comunicam por meio de uma rede Wi-Fi e enviam os dados coletados para uma estação central, de onde são transmitidos para um *middleware*. Esse é responsável por analisar, armazenar e disponibilizar os dados para serem posteriormente apresentados em interfaces gráficas de usuário. Com essas informações, é possível apresentar gráficos e painéis que auxiliam na tomada de decisão do usuário.

O trabalho Carmo et al. (2022) utiliza o Helix Sandbox NG, que tem como objetivo facilitar a construção de protótipos para ambientes inteligentes e que é uma plataforma aberta e rápida, que facilita a instalação e configuração dos mais importantes *Generics Enablers (GEs)* que são um conjunto de APIs que podem ser usados com componentes de terceiros para o desenvolvimento de sistemas inteligentes, eles são fornecidos pelo framework FIWARE.

O Fiware é um *framework* de componentes de código aberto que é utilizado para desenvolver soluções inteligentes sendo inclusive versátil para diversas finalidades e customizações. Com relação ao Helix Sandbox, ele tem a finalidade de acelerar o desenvolvimento de soluções inteligentes, sendo composto de uma plataforma modular e por diversos GEs que são recolhidos de acordo com a necessidade e o objetivo.

A respeito da arquitetura do Helix Sandbox NG, a arquitetura dele se encontra de acordo com a Figura 6.

No trabalho de Carmo et al. (2022) também é especificado o uso do protocolo MQTT como meio de comunicação entre o *Middleware* e o microcontrolador, sendo considerado um protocolo leve e flexível. É ideal para o desenvolvimento da tecnologia de IoT, pois permite a conexão entre os microcontroladores e as máquinas onde os dados serão tratados, sendo popular pela simplicidade, pelo baixo consumo de dados e pela possibilidade de comunicação bilateral, permitindo com que este seja um protocolo versátil para dar auxílio em diversos cenários de aplicativos para dispositivos e serviços de IoT.

2.3 Protocolos

Protocolos de rede são o conjunto de procedimentos e regras de comunicação que são utilizados em ambos os lados que estão se comunicando/trocando dados pela rede, porém nem todos possuem a mesma função ou a exata mesma maneira de como realizar as coisas, por exemplo, alguns protocolos operam no nível de várias camadas da arquitetura IoT, outros podem ser especializados em uma única tarefa correspondente.

Com o passar dos anos o número de dispositivos conectados à Internet aumenta de forma exponencial, o uso de protocolos de comunicação (camada de aplicação) permitem que os dispositivos se comuniquem com outros dispositivos, além de garantir a transferência de dados específicos entre as aplicações em um curto período de tempo e em segurança. Os critérios que foram comparados na pesquisa envolvem as Tecnologias envolvidas, Transporte, Arquitetura, Aptidão, Aplicação, Segurança e QoS, Fatores de Qualidade, Vantagens e Desvantagens e o resultado pode ser visualizado de acordo com as tabelas apresentadas neste trabalho.

O trabalho de Elhadi et al. (2018), analisou alguns dos principais protocolos em evolução relacionados à IoT conforme a Figura 7. Particularmente a pesquisa é centrada na camada de rede e em protocolos de aplicações. O estudo também avalia as capacidades dos protocolos e os compara com suas principais características e comportamentos em termos de vários critérios desses protocolos.

Elhadi et al. (2018) também cita a dificuldade em selecionar uma tecnologia sem fio, pois cada tecnologia possui suas próprias vantagens e desvantagens. Assim, surgem novos protocolos com características que se adaptam às necessidades dos dispositivos que estão conectados, como por exemplo, baixo consumo de energia, grande alcance, facilidade de implementação, baixa taxa de transferência, entre outras. Desta forma, foram classificados os principais protocolos existentes para IoT e em qual camada eles operam.

Aplicações IoT possuem características que podem ser utilizadas para selecionar o protocolo que será utilizado para o desenvolvimento, inclusive, segundo Elhadi et al. (2018), um dos pontos para o sucesso de uma aplicação IoT está na escolha correta do protocolo. Dessa forma, o trabalho em questão discorre sobre “qual protocolo escolher para uma determinada aplicação IoT?” e “quais os tipos de aplicações IoT são indicadas para cada protocolo?”.

O trabalho de Elhadi et al. (2018) também apresenta tabelas comparativas entre os protocolos, a primeira apresenta sobre protocolos da camada de rede, apresentando

sobre suas características como o tipo de rede, topologia utilizada, consumo de energia, frequência de banda, custo, risco de colisão de dados, adoção ao mercado, aplicações, entre outras características. A segunda tabela é com respeito aos protocolos da camada de aplicação, nela é comparado características como onde são aplicados, segurança, fator de qualidade, vantagens, desvantagens, entre outras características.

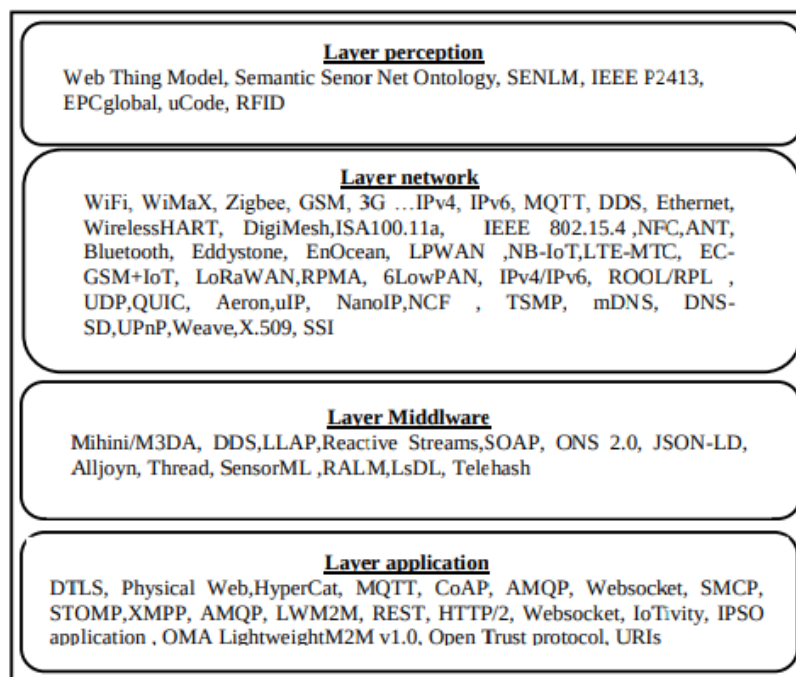


Figura 7. Protocolos de IoT (ELHADI et al., 2018).

O trabalho de Enany, Harb e Attiya (2021) apresenta um breve levantamento dos protocolos para IoT, detalhados por camada, e suas aplicações, além de um levantamento detalhado dos protocolos IoT para avaliar seu desempenho. Como caso de uso para o estudo comparativo, os autores deixam explícito que o protocolo MQTT (*Message Queuing Telemetry Transport*) foi escolhido devido à sua simplicidade e seu modelo de publicação/assinatura para ser avaliado e comparado com outros protocolos da camada de aplicação.

De acordo com os autores, segundo as pesquisas apresentadas, com o passar do tempo e com o rápido crescimento da IoT, a escolha do protocolo adequado para cada aplicação ou dispositivo se tornou um ponto crítico, nesse sentido, pesquisas foram realizadas para deduzir as vantagens e desvantagens dos protocolos, no entanto, segundo os autores, poucas pesquisas foram realizadas para aprimorar o protocolo MQTT porque é um protocolo leve e qualquer modificação nele resulta em um alto consumo de energia. Ele possui um cabeçalho de dois bytes, feito para dispositivos com recursos limitados, é semelhante ao HTTP, conta com o protocolo TCP com menor overhead para a transmissão de dados e substitui o modelo de solicitação/resposta pelo modelo de publicação/assinatura, que é mais adequado para a comunicação entre máquinas e para o ambiente IoT.

No modelo de publicação/assinatura, o servidor foi substituído por uma camada de software intermediária (*broker*) e o cliente foi substituído por um publicador (*publisher*) ou assinante (*subscriber*). O *broker* atua como um intermediário no processo de comunicação, onde cada dispositivo deve estabelecer uma conexão para efetuar a troca de mensagens. O endereço da mensagem é substituído pelo tópico da mensagem onde cada mensagem tem um tópico a ser publicado (AMAZON, 2023).

Sempre que um publicador envia mensagens, o *broker* as envia aos assinantes conectados. Portanto, o MQTT é considerado um protocolo orientado a mensagens. Ao contrário do modelo de solicitação/resposta, tanto o editor quanto os assinantes não precisam se conectar ao *broker* simultaneamente, conforme ilustrado pela Figura 8.

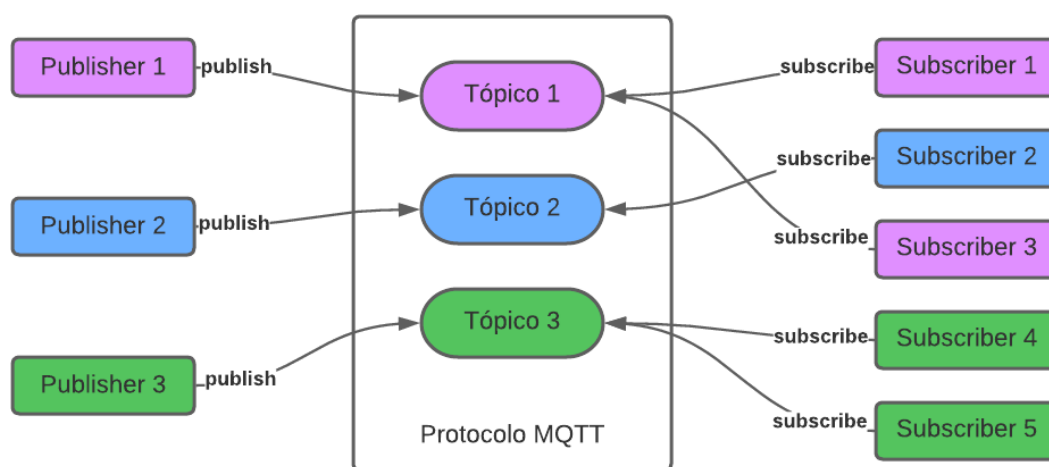


Figura 8. Modelo de publicação/assinatura (elaborado pelo autor).

O *broker* é o responsável por receber as mensagens dos editores e repassar as mensagens aos assinantes que solicitaram as mensagens publicadas. Ele atua como um organizador na arquitetura MQTT. Os *brokers* mais comuns do protocolo MQTT são o Mosquitto, hiveMQ, ActiveMQ, Bevywise MQTTRoute e RabbitMQ, cada um com recursos distintos porém com todos suportando o protocolo em questão.

Por ser baseado no protocolo TCP, o MQTT garante a confiabilidade por meio de três níveis de qualidade de serviço. De acordo com Enany, Harb e Attiya (2021), o MQTT é um protocolo mais confiável que o COAP e outros protocolos. Porém possui latência mais alta em comparação ao COAP devido ao processo de *handshake* e *acknowledgments*.

Segundo Enany, Harb e Attiya (2021), algumas melhorias foram implementadas no MQTT com o passar dos anos com o objetivo de eliminar o tamanho da comunicação entre os *publishers*, *subscribers* e o *broker*, como por exemplo, foram introduzidos dados temporais e agregação de tópicos para minimizar a sobrecarga de comunicação. Além disso, a assinatura é baseada em regras para reduzir o processamento. O protocolo também foi estendido para trabalhar com o protocolo UDP para reduzir a sobrecarga de comunicação no protocolo. Essa extensão foi denominada MQTT-SN e consiste em uma versão modificada do MQTT utilizado para operar com

redes de sensores sem fio onde os tópicos recebem IDs para serem processados em redes de baixa potência. Nesse contexto, devido suas vantagens e simplicidade, pesquisas para aprimorar o MQTT ainda estão em andamento a cada dia para superar suas limitações.

A respeito do levantamento comparativo realizado por Enany, Harb e Attiya (2021), eles foram realizados entre o MQTT e outros protocolos de IoT com o objetivo de localizar fraquezas deste protocolo. O protocolo COAP (*Constrained Applications Protocol*) foi o primeiro candidato a ser comparado ao MQTT. Isso se deve ao fato dele ter sido projetado para aplicativos restritos (*constrained applications*) e é executado sobre o protocolo UDP com menos sobrecarga de rede. Segundo testes realizados para mensurar o desempenho de ambos os protocolos durante certas circunstâncias, o protocolo MQTT possui desvantagens da alta latência e alto consumo de largura de banda, portanto o COAP supera o MQTT quando são utilizados mensagens curtas. Em contraste, quando confiabilidade e mensagens de tamanho grande são necessárias, o MQTT possui maior desempenho e menor atraso. Para redes de alto tráfego, o MQTT supera o COAP devido ao menor atraso, porém com maior consumo de energia.

Outro candidato da comparação já citada foi o AMQP, que se baseia especialmente no modelo de publicação/assinatura, com estrutura de fila usada em redes de alta perda de pacotes. Duas comparações foram realizadas para avaliar o desempenho do MQTT com relação ao AMQP em redes móveis instáveis e redes utilizando tecnologias de rádio. O MQTT manteve a entrega das mensagens ordenadas com um limite de carga útil maior que o AMQP. Em casos de alto atraso, o MQTT demonstra uma alta taxa de perda, pois depende do protocolo TCP para a transmissão confiável, mas em baixo atraso, o COAP mostra perdas menores do que o MQTT.

Como conclusão, o MQTT teve maior vantagem em receber mensagens ordenadas sobre os protocolos AMQP e COAP, também era o protocolo mais adequado para alto tráfego e alto volume de mensagens do que o protocolo COAP.

2.4 Outros Trabalhos Relacionados

Outro trabalho relacionado é o portal REDEMET (Rede de Meteorologia do Comando da Aeronáutica) que é a plataforma oficial utilizada pelo DECEA (Departamento de Controle do Espaço Aéreo) para fornecer dados meteorológicos que são importantes para identificar fenômenos que podem influenciar nas atividades de navegação aérea. É importante destacar que a plataforma permite o acesso a esses dados utilizando um mapa que facilita a geolocalização de cada estação disponível.

Os dados são classificados por cores e plotados no mapa georreferenciado. Além disso, o sistema também apresenta informações provenientes de vários produtos sobrepostas no mesmo mapa, como por exemplo, imagens de radares, imagens de satélites, modelagem numérica, condições meteorológicas, plotagens do SIGMET, entre outros.

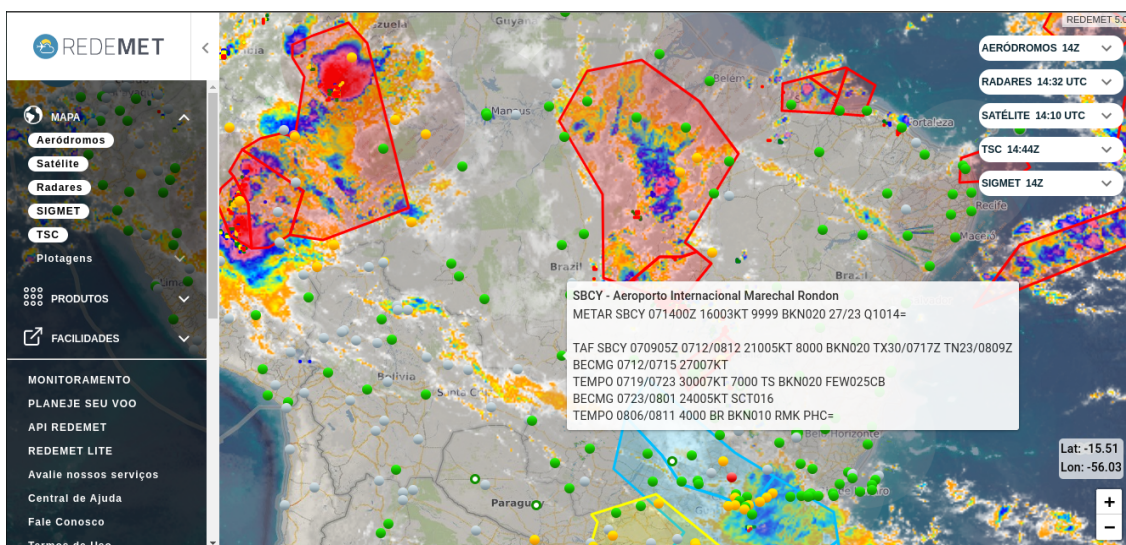


Figura 5. Interface da plataforma REDEMET por REDEMET(2023).

3. Estações Meteorológicas Automáticas

Atualmente, além dos aparatos tecnológicos comuns para a coleta de variáveis meteorológicas, como imagens de satélites e balões meteorológicos, a captação dessas variáveis também pode ocorrer utilizando uma Estação Meteorológica Automática (EMA) ou *Automatic Weather Station (AWS)*.

3.1 Dados meteorológicos

Uma EMA pode catalogar diversas variáveis meteorológicas, dentre elas pode-se citar como exemplos umidade relativa do ar, temperatura do ar, velocidade e direção do vento, radiação solar (incidente e refletida), volume de precipitação, pressão atmosférica, dados sobre materiais particulados como a fuligem, dióxido e monóxido de carbono entre outros (BRAGA et al., 2011) (SILVA; FRUETT, 2013).

O estudo de variáveis meteorológicas como a umidade relativa do ar (utilizando um higrômetro) podem auxiliar muito devido ao fato de que a umidade mede a quantia de vapor d'água atmosférica em relação à saturação do vapor disponível no ambiente, ou seja, quanta água em estado gasoso há no ar em relação ao máximo que poderia haver na temperatura atual, quanto mais próximo de 100%, maior a quantia de água presente na atmosfera. A capacidade de medir esta variável nos auxilia, por exemplo, a otimizar a eficiência dos sistemas de aclimação, fornecendo uma indicação de quanto ar externo sendo ingerido precisa ser condicionado, dependendo da temperatura ambiente do lado de fora.

Já a pressão atmosférica ou pressão barométrica, que é a força exercida pela atmosfera sobre uma unidade de área, é importante para compreender o que causa determinadas sensações em um dado indivíduo ou objeto, pensando em aerodinâmica por exemplo, a grandeza física deve ser considerada quando se busca um rendimento

aerodinâmico, seja para aviões ou para carros.

O conhecimento sobre velocidade e direção do vento, que pode ser crucial em muitos casos e auxilia muito navegadores e pilotos de transportes aéreos a evitarem áreas onde o vento é muito forte, onde há uma possibilidade de tempestade, permitindo com que eles possam contornar essas adversidades não só com essa variável mas com o conjunto de diversos outros dados também.

3.2 Hardware

Para suprir a demanda por dados locais de estudo, pesquisadores costumam instalar e operar estações meteorológicas automáticas, que oferecem certa praticidade pois possibilitam altas taxas de amostragens e pouca manutenção. Tratando do hardware de uma EMA, uma grande variedade de instrumentos pode ser encontrada no mercado brasileiro, o que pode dificultar a tarefa de escolha adequada dos equipamentos por parte de pesquisadores e interessados na construção de uma estação meteorológica automática. Por essa razão, é comum que o comprador selecione uma marca e realize a compra em forma de “pacote”, no qual todos os componentes serão adquiridos de um único fabricante. Os equipamentos que compõem as estações meteorológicas podem ser separados em três categorias principais: sensores, registradores e telemetria (BRAGA et al., 2011).

Tratando de sensores, eles traduzem eventos físicos em sinais elétricos e eletrônicos e são responsáveis por quantificar diversas variáveis meteorológicas. EMAs geralmente operam com apenas um registrador central, que é denominado como *datalogger*, que armazena as leituras dos sensores, podendo também comandar a transmissão telemetria dos dados registrados.

Estações meteorológicas automáticas comuns geralmente estão equipadas com quatro sensores essenciais, onde cada um é responsabilizado por captar uma variável meteorológica distinta. Entre esses sensores estão o Anemômetro, que é utilizado para adquirir dados sobre a velocidade e a direção do vento, o Piranômetro que registra a radiação solar, sensores de temperatura e umidade e um Pluviômetro do tipo báscula, para medir a precipitação.

4. Sistemas Distribuídos em IoT

Para o desenvolvimento de uma EMA, pode-se aplicar conceitos e tecnologias de Internet das Coisas (*Internet of Things – IOT*), que poderia ser definido como uma rede de dispositivos conectados de forma que os dados sejam comunicados sem a necessidade de interação humana, permitindo a coleta das variáveis meteorológicas de forma automática e economizando tempo para a análise das informações (PATEL, 2016). Nesse sentido, o conceito de Sistemas Distribuídos também é aplicado, que consistem em um conjunto de computadores independentes que se apresenta aos seus usuários como um sistema único e coerente (TANENBAUM; STEEN, 2007).

Atualmente existem inúmeras aplicações de IoT em diversos campos, como em nossas casas, na área da saúde, indústrias, varejo, transporte, vigilância e etc, porém a

IoT está enfrentando vários obstáculos em sua forma de ser implementada como uma tecnologia global, afinal, existem muitos problemas em aberto que precisam de uma solução para a implementação eficiente.

Alguns dos problemas seriam a falta de arquitetura e protocolos padrão. Atualmente há diversos protocolos que podem ser utilizados como o MQTT, HTTP, AMQP, CoAP, DDS, M2M, XMPP, LoRaWAN, entre muitos outros.

Segundo SOBIN(2020), a segurança e a privacidade pessoal são os principais requisitos que podem garantir a ampla adoção da tecnologia de IoT. Entretanto, também é o principal desafio, pois deve ser considerada como uma prioridade chave no nível do sistema para IoT e deve ser levada em consideração a arquitetura da aplicação também. Por exemplo, no contexto de uma aplicação sem fio, há uma série de vulnerabilidades de segurança que podem ser exploradas e devido à natureza altamente restrita dos dispositivos e à largura de banda limitada, torna-se desafiador implementar políticas de segurança em IoT. Os algoritmos de segurança devem ser simples e deve haver um número mínimo de trocas de mensagens devido à largura de banda limitada.

Outro desafio é o fato da identificação e da escalabilidade, afinal, como o escopo geral de IoT é bem maior do que a Internet dos Computadores e o número de objetos que estão conectados entre si é muito grande, podem surgir problemas de escalabilidade em diversos níveis, como por exemplo, problemas de nomeação e endereçamento devido ao grande número de dispositivos interconectado. Nesse contexto, o espaço de endereçamento deve ser grande o suficiente para acomodar os dispositivos. Quanto ao gerenciamento de informações, deve-se utilizar mecanismos adequados para a extração das informações úteis dos dados que serão armazenados, afinal, há um grande número de dados que são captados devido ao grande número de sensores.

Sistemas IoT também devem ter a capacidade de auto-organização, afinal, os objetos inteligentes precisam se organizar e se adaptar às situações que vierem a ocorrer sem a necessidade da intervenção humana. Portanto, existe a necessidade de “inteligência” distribuída em IoT de forma a permitir que os objetos sejam capazes de reagir autonomamente a uma ampla gama de situações de modo que a intervenção humana seja pequena. Há recursos a serem utilizados para satisfazer esse desafio como a descoberta automática de serviços e dispositivos sem a necessidade de gatilho externo e a capacidade de ajustar de forma adaptativa o comportamento dos protocolos (SOBIN, 2020).

5. Projeto do Sistema

Esta seção apresenta o projeto do sistema proposto, incluindo sua Arquitetura e as Tecnologias selecionadas para seu desenvolvimento. Além disso, também apresenta a prototipação do gerenciador Web, necessário para que os dados das EMAs monitoradas sejam obtidos pelos usuários.

5.1 Arquitetura do Sistema

Para desenvolver o trabalho em questão, foi necessário definir uma arquitetura de software que possibilitasse uma certa tolerância a falhas de comunicação, considerando que os locais onde as EMAs estarão instaladas nem sempre garantem comunicação a todo momento. Uma representação da arquitetura proposta é ilustrada pela Figura 9.

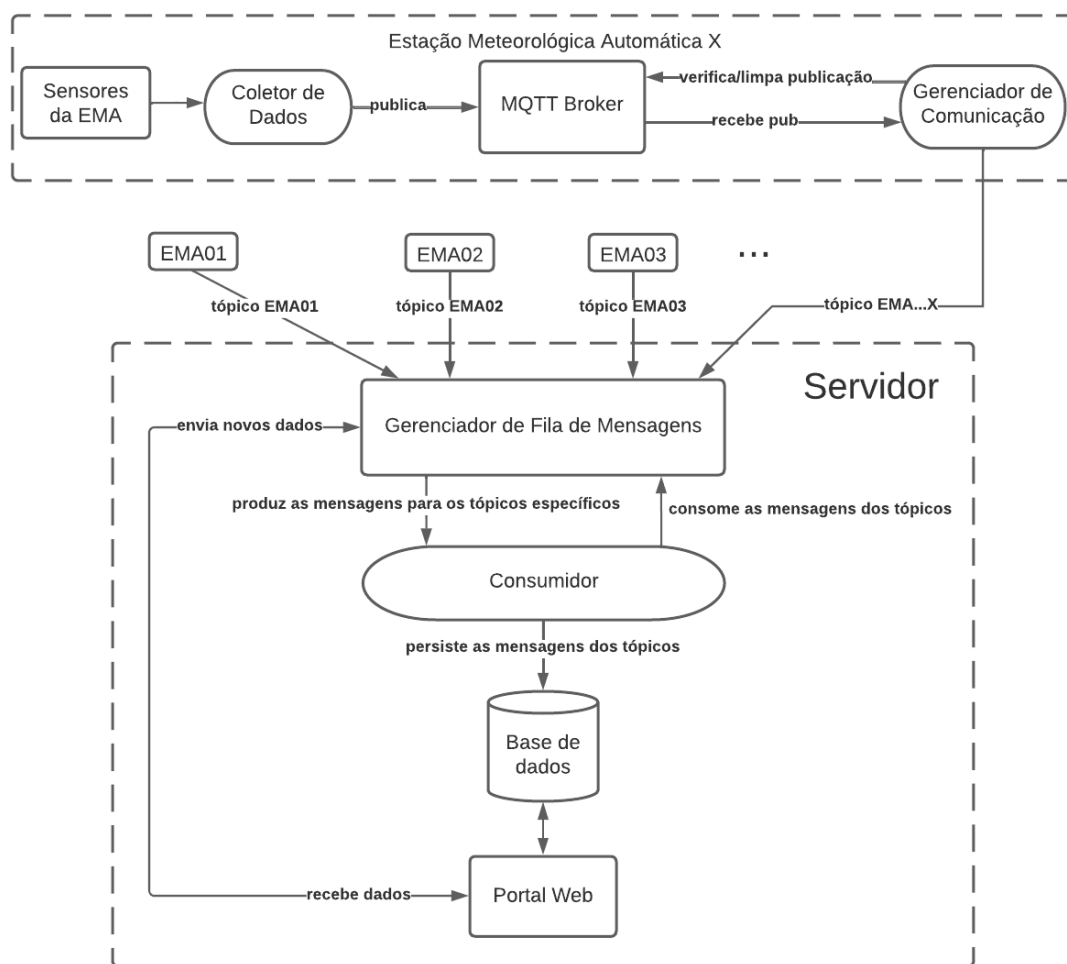


Figura 9. Representação da arquitetura proposta (elaborado pelo autor).

De acordo com a arquitetura proposta, os dados dos sensores são obtidos continuamente pelo módulo “coletor de dados” de uma EMA e são publicados em um único tópico no *broker*, também localizado na EMA. Em paralelo, o módulo “Gerenciador de Comunicação” da EMA permanece verificando se existem novas mensagens publicadas no tópico e as consome assim que disponíveis. Todas essas comunicações, mesmo que internamente à EMA, utilizam o protocolo MQTT.

Assim que uma mensagem é consumida pelo “Gerenciador de Comunicação”, ela será enviada ao “Gerenciador de Fila de Mensagens”. Caso esse envio tenha sucesso, essa mensagem será excluída da fila de mensagens do *broker* para evitar armazenamento de mensagens já consumidas. Em caso de perda de conexão com o Gerenciador de Fila de Mensagens, os dados coletados pela EMA não serão perdidos e podem ser enviados quando a conexão estiver restabelecida.

Quando uma mensagem é recebida pelo Gerenciador de Fila de Mensagens, ele as armazena em um tópico específico, referente à EMA de onde ela veio. Após armazenadas nos tópicos, essas mensagens são coletadas por um processo “consumidor” que as persistirá no banco de dados. A base de dados mantém todas as mensagens consumidas. Além disso, também mantém os dados dos usuários do portal Web e os dados das EMA cadastradas no sistema, como por exemplo, endereço IP, URL, localização, entre outras.

O portal Web tem a função de possibilitar que os usuários do sistema tenham acesso aos dados atuais da EMA, bem como, aos dados históricos da mesma, podendo inclusive baixá-los em formato CSV para utilização em estudos futuros.

A respeito da arquitetura do projeto, vale ressaltar que o foco da proposta não é desenvolver as EMA, mas desenvolver uma plataforma que seja aberta para aceitar EMAs que se comuniquem utilizando o protocolo MQTT, permitindo com que os dados recebidos possam ser consumidos e posteriormente apresentados em um portal web.

5.2 Tecnologias envolvidas

O trabalho faz uso do MQTT (*Message Queuing Telemetry Transport*) que é um protocolo que tem por objetivo transportar mensagens utilizando o modelo de *publish/subscribe*, mas de forma leve e eficiente. Isso é conseguido devido ao cabeçalho das mensagens MQTT terem sido projetados com tamanho bastante reduzido, o que auxilia na diminuição do consumo de largura de banda da rede (MQTT, 2023).

As mensagens deste protocolo podem conter qualquer tipo de dados, desde cadeias de caracteres até objetos complexos representando textos, vídeos, dados de sensores, áudio ou qualquer outro conteúdo digital. Além disso, cada mensagem deve ser atribuída a um “tópico” ao qual ela será vinculada. O “tópico da mensagem” seria um intermediário entre os emissores e os destinatários de uma mensagem, ele contém uma lista de destinatários interessados nas mensagens que são vinculadas à ele.

Quanto ao modelo *publish/subscribe*, o *publisher* ou publicador, é o remetente da mensagem. Ele envia mensagens sobre um tópico apenas uma vez a todos os assinantes deste mesmo tópico. A interação deste publicador para com os seus assinantes (*subscribers*) é uma relação onde há um publicador para um ou muitos assinantes. Os assinantes precisam registrar (ou assinar) os tópicos de interesse, sendo capazes de executar funções diferentes ou fazer algo distinto com a mensagem em paralelo (AMAZON, 2023)(COULOURIS, 2013).

De acordo com os trabalhos relacionados anteriormente apresentados, o MQTT utiliza um *broker*, que é uma camada de software intermediária que interconecta publicadores e assinantes, sendo o responsável pelo recebimento e enfileiramento das mensagens.

A respeito das filas de mensagem distribuídas, elas oferecem um serviço ponto a ponto, usando o conceito de fila de mensagens como uma indireção, obtendo, assim, as propriedades desejadas do desacoplamento temporal e espacial. A respeito das filas serem ponto a ponto, isso indica que o remetente coloca a mensagem em uma fila e isso, então é removido por um único processo. Filas de mensagem também são referidas

como *middleware* orientado a mensagens. Esses sistemas possuem uma propriedade fundamental que é o fato das mensagens serem persistentes, ou seja, as filas armazenam as mensagens indefinidamente até serem consumidas e também as armazenam no disco para permitir uma entrega confiável. Além disso, são tolerantes à falhas para quando partes do sistema ficam *offline* (COULOURIS, 2013).

Existem diversas vantagens no uso do MQTT, entre elas, sua segurança, pois possibilita o envio de mensagens utilizando o protocolo SSL (*Secure Socket Layer*). Este protocolo fornece segurança para protocolos de camada de aplicação que não possuem segurança nativa, como por exemplo, o HTTP. Além disso, as linguagens de programação fornecem APIs com sockets para realizar implementações utilizando SSL. (KUROSE, 2015).

O protocolo MQTT também oferece mais confiança aos usuários ao garantir que as mensagens sejam entregues. Essa confiabilidade ocorre por meio de configurações de qualidade de serviço (*QoS*). Essas configurações de *QoS* estabelecem uma escala (de 0 a 2) que mede a importância e a necessidade do recebimento. De acordo com a escala, no nível 0, a mensagem deve ser recebida até uma vez e não possui confirmação de recebimento e nem faz o armazenamento de mensagens para retransmissão (esta é a configuração utilizada pelo projeto até o momento). No nível 1 indica que a mensagem deve ser recebida ao menos uma vez e deve ter confirmação de entrega. Já no nível 2, a mensagem deve ser recebida apenas uma vez e também há confirmação de recebimento (MQTT, 2023).

Outra vantagem é o fato do protocolo MQTT não sobrecarregar a rede, pois apenas os dados necessários são transmitidos. Ele também é considerado estável em ambientes não confiáveis de alta latência e baixa largura de banda, o que torna a combinação muito boa para os cenários de IoT (AMAZON, 2023).

O *broker* MQTT utilizado neste trabalho é o “Mosquitto”. Seu uso se deve ao fato dele ser leve e adequado para o uso em diversos dispositivos, desde computadores de baixa potência com placa única até em servidores completos. Sua instalação também é simples e ele é muito indicado para utilização com sensores (residenciais e industriais). Além disso, seu código é aberto e está disponível gratuitamente (MOSQUITTO, 2023).

Este trabalho também faz uso do Apache Kafka, que é uma plataforma de transmissão de dados capaz de publicar, assinar, armazenar e processar fluxos de registro em tempo real. O Apache Kafka foi desenvolvido para efetuar o processamento de fluxos de dados provenientes de diversas fontes e entregá-los a uma grande variedade de clientes. A ferramenta é capaz de não só movimentar grandes volumes de um ponto A ao ponto B, mas também de A até Z e para qualquer outro local que for necessário simultaneamente, tornando essa tecnologia excelente para dimensionamento (REDHAT, 2023) (APACHE SOFTWARE FOUNDATION, 2023).

O Apache Kafka também possui uma grande capacidade de armazenamento e consumo de dados e é considerado uma boa tecnologia para cenários que exigem uma série de etapas de processamento de dados e de alto desempenho ou integração de dados de vários sistemas (REDHAT, 2023) (APACHE SOFTWARE FOUNDATION, 2023).

O uso de MQTT e Kafka, pode ser benéfico se for necessário criar processamento de dados de alto desempenho, armazenar grandes quantidades de mensagens, integrar diferentes aplicativos de negócios ou data centers em tempo real.

No caso de houver muitos aplicativos ou dispositivos pequenos que executam em ambientes *wireless* ou instáveis, onde trocam mensagens em tempo real em vários tópicos, o uso do MQTT é necessário.

Em resumo, o que torna óbvio o uso destas duas tecnologias juntas seria o fato da estrutura de comunicação em tópicos e o modelo de comunicação de *publish/subscribe*.

No contexto deste trabalho, a resiliência e a tolerância a falhas são muito importantes, mas difíceis de alcançar, especialmente se for necessária a garantia de ponta a ponta de pelo menos uma vez ou exatamente uma entrega de mensagem. O gerenciador de comunicação só pode confirmar o recebimento do MQTT se tiver encaminhado com sucesso a mensagem para o intermediário Kafka ou precisar armazenar em buffer as mensagens no caso de algo dar errado. É necessário um esforço significativo de desenvolvimento no tratamento de erros e funcionalidade semelhante à tecnologia já encontrada no Kafka e/ou no *broker* MQTT (TECHLETTERS, 2021).

Todo *backend* do trabalho está sendo implementado em linguagem de programação Python. O uso dessa linguagem se deve ao fato dela possuir bibliotecas para o protocolo MQTT (“paho-mqtt”) e o Kafka (“pykafka” e “kafka”), também pelo fato de ser de alto nível (sintaxe ser voltada para o entendimento humano) e muito versátil, sendo capaz de ser utilizada para uma grande diversidade de aplicações.

O banco de dados utilizado é o MySQL Server, sua escolha se deve devido ao fato dos dados serem estruturados e o modelo do banco ser pensado em um banco de dados relacional.

A linguagem utilizada para apresentar os dados na plataforma Web como prova de conceito é o PHP, sua escolha se deve devido ao fato desta linguagem ser compatível com diversos bancos de dados como o MySQL e também pelo fato de ser uma linguagem muito versátil, funcionando em uma grande variedade de plataformas.

5.3 Protótipo do Sistema

A Figura 10 apresenta um protótipo da página principal do portal Web de gerenciamento. Nela o usuário não autenticado terá acesso a dados de EMAs públicas que ficarão fixadas no mapa. Quando o usuário interagir com uma EMA, uma janela sobreposta apresentará alguns dados da EMA selecionada, contendo suas coordenadas geográficas e um botão que possibilita apresentar mais detalhes dos dados coletados por aquela EMA em específico. No canto superior direito da tela, na barra de navegação, é possível visualizar dois botões, o primeiro seria para efetuar o cadastro do usuário e o segundo seria utilizado para efetuar o login para iniciar uma sessão.

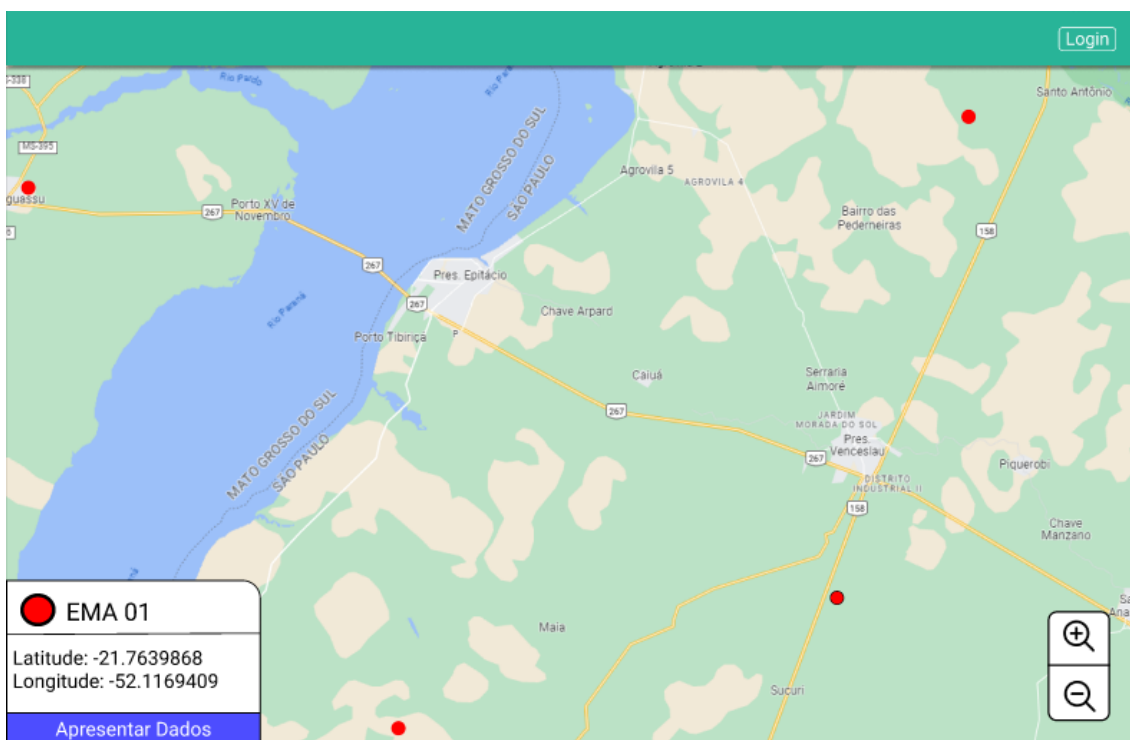


Figura 10. Protótipo da Interface 1 (elaborado pelo autor).

Após o usuário realizar a autenticação e iniciar sua sessão, ele tem acesso a mesma página, porém, com o acréscimo das estações meteorológicas privadas que ele mesmo adicionou conforme a Figura 11. Na barra de navegação superior da página, é possível visualizar a existência de um menu lateral no canto superior esquerdo, nele o usuário terá acesso a função de “Adicionar EMA”, que permitirá que o usuário cadastre uma estação e a configure para que tenha seus dados apresentados na plataforma.

O sistema também permitirá que o usuário acesse os relatórios da EMA que ele selecionar, permitindo que ele apresente os dados da EMA em específico conforme ilustra a Figura 11.

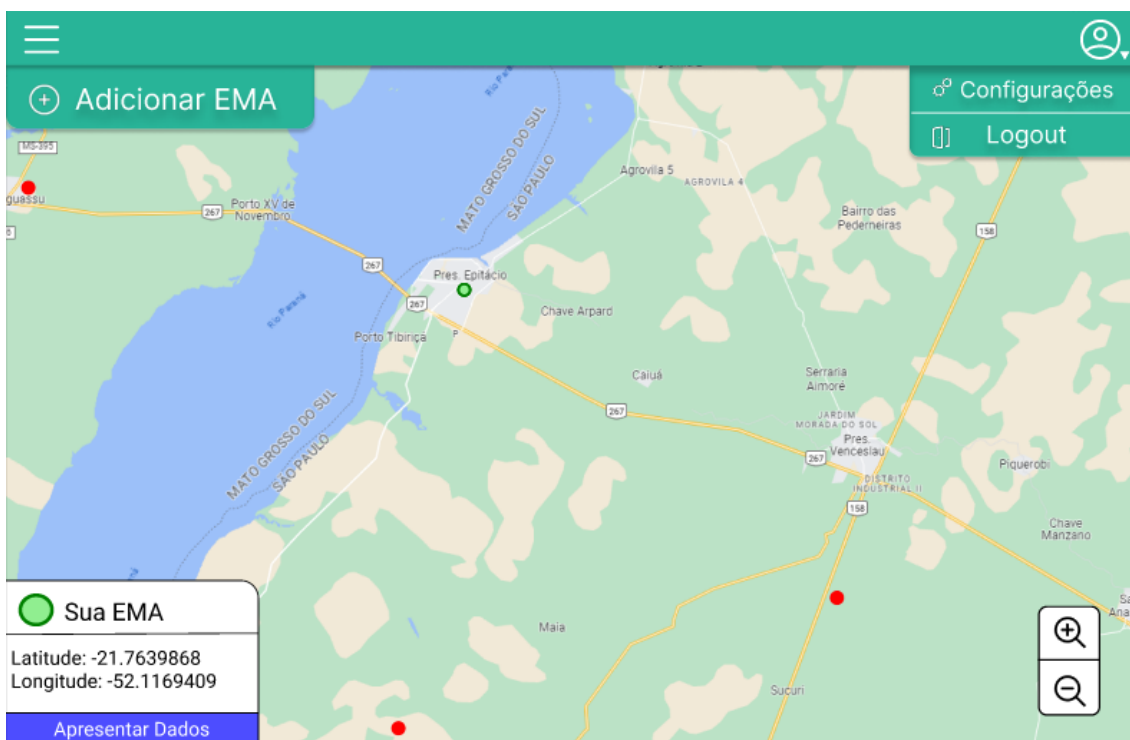


Figura 11. Protótipo da Interface 2 (elaborado pelo autor).

A Figura 12 apresenta os dados coletados por uma EMA. Nela, são apresentados alguns botões que farão com que o gráfico ao lado seja alterado dependendo da opção selecionada. Por exemplo, os dados da EMA que estão sendo apresentados na Figura 12 são relacionados à velocidade do vento que aquela estação captou ao longo do dia, mas poderiam tanto ser dados relacionados à Temperatura, Direção do Vento entre outros.

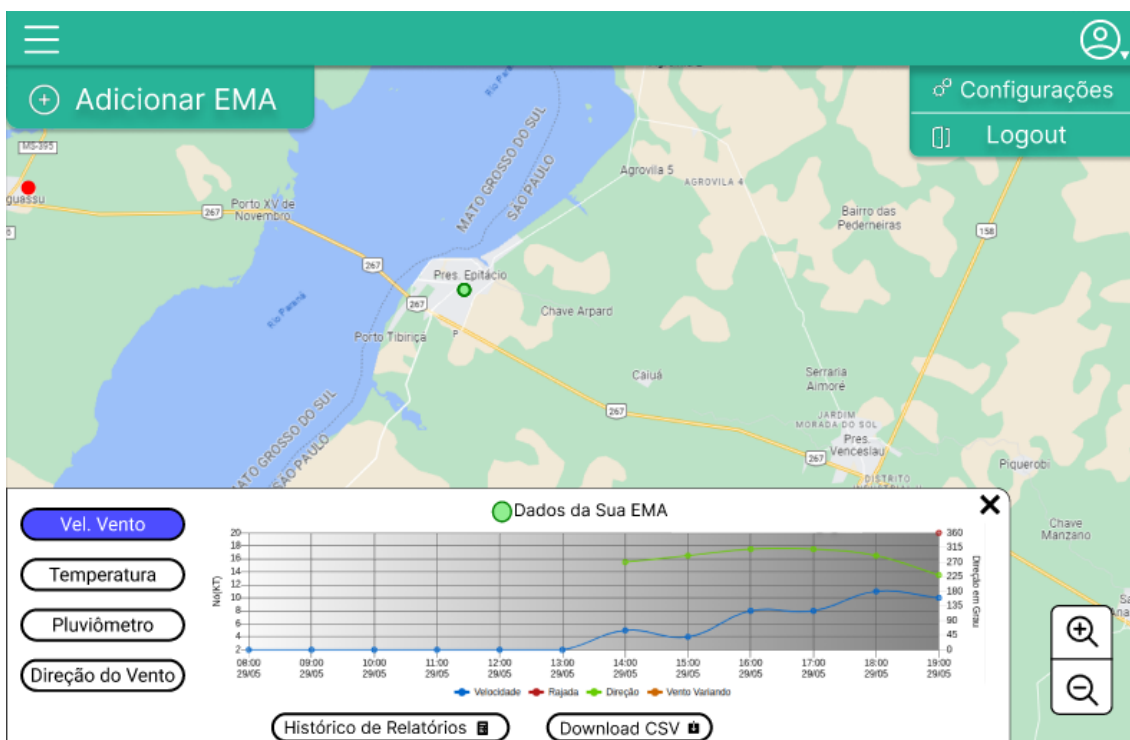


Figura 12. Protótipo de Interface 3 (elaborado pelo autor).

6. Considerações Finais

Considerando o desenvolvimento parcial do trabalho, foi proposta uma arquitetura aberta e que permite a comunicação e o gerenciamento de estações meteorológicas automáticas utilizando o protocolo MQTT.

Como prova de conceito, foi desenvolvido uma EMA simulada, implementada em Python, que envia dados randomizados a cada segundo para o *broker* Mosquitto. O *broker* armazena as mensagens recebidas em um tópico dentro da EMA, após estes tópicos serem assinados, eles são enviados para o gerenciador de fila de mensagens.

O Kafka, por sua vez, armazena a mensagem recebida em um tópico específico da EMA no Kafka. Em seguida, os dados são persistidos para o banco de dados para serem consumidos pela plataforma Web.

Apesar da arquitetura do trabalho já estar definida, o protocolo de camada de aplicação que define o formato das mensagens que serão enviadas entre os componentes do sistema ainda não foi elaborado. Esse protocolo é importante para promover a abertura do sistema, uma vez que as EMA precisam conhecer o formato das mensagens para que possam enviar as mesmas ao sistema. Além disso, este protocolo precisa ser implementado utilizando o MQTT. Dessa forma, essas tarefas devem ser executadas na próxima fase do TCC.

7. Referências

Apache Kafka. Apache Software Foundation, 2023. Disponível em: <https://kafka.apache.org/>. Acesso em: 2 de mai. de 2023.

BRAGA, A.S.; BRAGA, S.M.; FERNANDES, C.V. S.; Estações Meteorológicas Automáticas: Relato de Uma Experiência com Sensores Independentes em Bacia Experimental. In: XIX SIMPÓSIO BRASILEIRO DE RECURSOS HÍDRICOS. Maceió, 2011. Anais [...] Curitiba: Universidade Federal do Paraná – Departamento de Hidráulica e Saneamento. Disponível em: https://abrh.s3.sa-east-1.amazonaws.com/Sumarios/81/0bc8143e9534d18955e138aea552b69e_93806c7f46f088e5a279ffc5c870b5c7.pdf. Acesso em: 10 de mar. de 2023.

CARMO, Jefferson V. et al. Smart Weather: Micro Estações Meteorológicas Baseadas em Rede Mesh e IoT. Faculdade de Tecnologia de São Caetano do Sul do Centro Estadual de Educação Tecnológica Paula Souza.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim; BLAIR, Gordon; Sistemas Distribuídos: Conceitos e Projeto. 5 ed. Bookman, 2013.

Eclipse Mosquitto. Mosquitto, 2023. Disponível em: <https://mosquitto.org/>. Acesso em: 18 de mai. de 2023.

ELHADI, Sakina et al. Comparative Study of IoT Protocols. SSRN, 2018. Disponível em: <https://ssrn.com/abstract=3186315>. Acesso em: 3 de abr. de 2023.

ENANY, M. O. Al; HARB, H. M.; ATTIYA, G. A Comparative analysis of MQTT and IoT application protocols. In: International Conference on Electronic Engineering (ICEEM), Menouf, Egypt, 2021, pp. 1-6, doi: 10.1109/ICEEM52022.2021.9480384.

FINHOLDT, Gustavo et al. Desenvolvimento de uma estação meteorológica automática para manejo de irrigação. Engenharia de Agricultura, v.19, n.4, p. 313–328. 2011.

KUROSE, J. F.; ROSS, K. Redes de Computadores e a Internet. 6 ed. São Paulo: Pearson Prentice Hall, 2015.

MQTT: The Standard for IoT Messaging. MQTT, 2023. Disponível em: <https://mqtt.org/>. Acesso em: 23 de mai. de 2023.

MUNANDAR A. et al. Design of real-time weather monitoring system based on mobile application using automatic weather station. In: 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), Jakarta, Indonesia, 2017, pp. 44-47, doi: 10.1109/ICACOMIT.2017.8253384.

NOVIANTY, Inna et al. Design of Portable Weather Station Using MQTT Protocol. 2nd International Conference of Computer and Informatics Engineering (IC2IE), Banyuwangi, Indonésia, p. 199-202. 2019.

O que é Apache Kafka?. Redhat, 2023. Disponível em: <https://www.redhat.com/pt-br/topics/integration/what-is-apache-kafka>. Acesso em: 14 de mai. de 2023.

O que são Mensagens pub/sub?, Amazon, 2023. Disponível em: <https://aws.amazon.com/pt/what-is/pub-sub-messaging/>. Acesso em 23 de mai. de 2023.

PATEL, Keyur K. et al.. Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. ResearchGate, 2016. Disponível em: https://www.researchgate.net/publication/330425585_Internet_of_Things-IOT_Definition_Characteristics_Architecture_Enabling_Technologies_Application_Future_Challenges. Acesso em: 12 de mar. de 2023.

REDEMET. Rede Meteorológica do Comando da Aeronáutica. Disponível em: <https://www.redemet.aer.mil.br/>. Acesso em: 14 de mai. de 2023.

PONT, Isabelle A. D.; SANTOS, Fernando R.; UGGIONI, Perison P. Desenvolvimento de uma Estação Meteorológica Automática baseada em Internet das Coisas. Metrologia 2021, 2021. Disponível em: <https://metrologia2021.org.br/?p=2161>. Acesso em: de 2023.

SILVA, Marcel S. da; FRUETT, Fabiano. Monitoramento ambiental através de rede de sensores sem fio de baixo custo. In: Anais XVI Simpósio Brasileiro de Sensoriamento Remoto –SBSR. Foz do Iguaçu/PR, Brasil INPE, 2013.

SOBIN, C.C. A Survey on Architecture, Protocols and Challenges in IoT. Springer, 2020. Disponível em: <https://doi.org/10.1007/s11277-020-07108-5>. Acesso em: 10 de mar. 2023

TANENBAUM, Andrew S. STEEN, Maarten V. Sistemas Distribuídos: princípios e paradigmas. 2 ed. São Paulo: Pearson Prentice Hall, 2007.

TECHLETTERS. MQTT and Kafka: How to combine two complementary technologies. Medium, 2021. Disponível em: <https://medium.com/python-point/mqtt-and-kafka-8e470eff606b>. Acesso em: 20 de mai. de 2023.

YNOUE, R. Y et al. Meteorologia: noções básicas. 1.ed. São Paulo: Oficina de Textos,

2017. v.1.