

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 utilizado diversas estações meteorológicas automáticas (EMAs), 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).

Nesse sentido, 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. Os objetivos específicos deste trabalho, são:

- Desenvolver uma arquitetura de software e implementar um protocolo de comunicação capaz de realizar a comunicação entre EMAs e o Servidor.
- Desenvolver uma EMA simulada e um sistema Web de gerenciamento dos das EMAs.
- Construir conhecimento sobre o assunto para motivar o desenvolvimento de trabalhos futuros.

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

1.1 Metodologia

O trabalho deu início com a revisão bibliográfica com o objetivo de reunir conhecimento sobre as informações necessárias para o seu desenvolvimento. Nesse sentido, arquiteturas de sistemas IoT distribuídos, implementações de estações meteorológicas automáticas utilizando microcomputadores, trabalhos que utilizam o protocolos de comunicação e o gerenciadores de fila de mensagens, foram fontes importantes para a conclusão deste trabalho.

Após o levantamento bibliográfico, foi necessário definir a arquitetura e o protocolo que realizaria a comunicação entre a EMA e o Servidor. Para o desenvolvimento do protocolo, foi necessário definir quais variáveis meteorológicas seriam coletadas e enviadas pelas EMAs, nesse sentido, foi selecionado as variáveis que frequentemente apareciam nos trabalhos relacionados e em seguida foi desenvolvido o protocolo MIAP (*Meteorological Information Application Protocol*) e a arquitetura que é utilizada pelo trabalho.

Com a arquitetura e o protocolo definido, foi realizada a implementação dos módulos que o protocolo contém para o tratamento das mensagens que as EMAs coletam e enviam para o Servidor, nesse sentido, com o protocolo realizando o tratamento dos dados conforme o esperado, foi desenvolvido um protótipo do sistema web e em seguida o sistema web de gerenciamento, que facilita o

gerenciamento das estações e a visualização dos dados meteorológicos que elas coletam.

Inicialmente os módulos que representavam a EMA simulada eram utilizados no mesmo ambiente que o módulo que consumia as mensagens coletadas por ela, então foi realizada a implementação de uma EMA simulada utilizando uma máquina virtual com o sistema operacional Raspbian, do hardware Raspberry Pi, onde os dados simulados são gerados e por meio do módulo que realiza a comunicação entre a EMA e o Servidor, eles são publicados em um tópico do gerenciador de fila de mensagens, Kafka, que por sua vez, tem suas mensagens consumidas pelo módulo de consumidor que fica localizado no Servidor.

1.2 Organização do texto

Esta seção apresenta a organização do texto neste trabalho. A segunda seção apresenta detalhes de implementação do trabalho em questão apresentando características sobre a arquitetura desenvolvida, detalhes sobre a implementação da EMA simulada e a forma como ela utiliza o protocolo MQTT e QoS para garantir a entrega das mensagens e detalhes sobre o sistema web de gerenciamento.

A terceira seção descreve o protocolo de comunicação desenvolvido, incluindo a organização das camadas rede e as tecnologias envolvidas, o formato da mensagem, o JSON Schema utilizado para validar as mensagens e o diagrama de estados finitos que formaliza o protocolo.

A quarta seção apresenta as conclusões do trabalho, as dificuldades encontradas e como elas foram mitigadas, além de sugestões de trabalhos futuros.

2. Detalhes de implementação

Esta seção descreve alguns detalhes da implementação do trabalho desenvolvido com o objetivo de apresentar mais características sobre a arquitetura desenvolvida como detalhes sobre a implementação da EMA simulada, do Sistema de Gerenciamento e a formalização do protocolo desenvolvido.

2.1 Geral

A arquitetura do trabalho está definida de acordo com a Figura 1, onde, de acordo com a arquitetura proposta, as EMAs teriam dois módulos, o primeiro seria o módulo de “Coletor de Dados”, que será o responsável por captar os dados recebidos pelos sensores da estação e então publicá-los em um tópico do *broker* MQTT da estação e o segundo módulo seria o “Gerenciador de Comunicação”, ele recebe as publicações então as redireciona à um tópico existente no “Gerenciador de Fila de Mensagens” alocado no Servidor. Ambos os módulos da EMA são softwares implementados utilizando a linguagem Python e a estação foi implementada utilizando uma máquina virtual com o sistema operacional do Raspberry Pi, Raspbian.

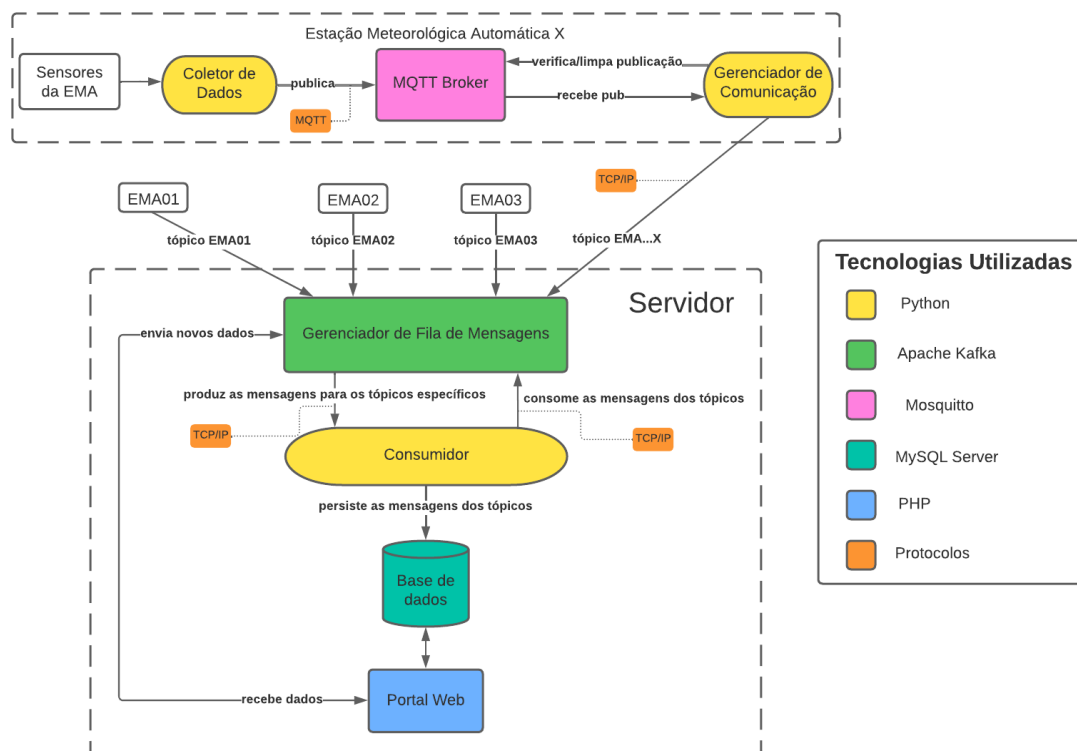


Figura 1. Representação da arquitetura proposta (elaborado pelo autor).

No Servidor, teríamos o “Gerenciador de Fila de Mensagens”, para a implementação foi utilizado o Kafka, o Banco de Dados utilizado é o MySQL por se tratar de dados estruturados e o Portal Web seria o Sistema de Gerenciamento que foi desenvolvido utilizando PHP.

Nesse sentido, nos capítulos a seguir será possível explorar de maneira mais aprofundada a respeito de cada um dos componentes da arquitetura proposta, permitindo o entendimento de como foi desenvolvido a implementação da EMA simulada, do Sistema de Gerenciamento e do Protocolo de Comunicação que é utilizado para tratar das mensagens.

2.2 EMA simulada

Para a entrega de mensagens, foi desenvolvido uma EMA simulada utilizando uma máquina virtual Raspbian, o uso deste sistema operacional se deve ao fato de ser um sistema operacional baseado em Debian para o hardware Raspberry Pi, que é um computador pequeno integrado em uma placa de circuito, frequentemente utilizado para soluções IoT como no caso de casas inteligentes, desenvolvimento de robôs, mini servidores, centrais de multimídia, estações meteorológicas e muitos outros projetos (RASPBERRY PI FOUNDATION, 2023).

Na estação simulada, se encontra o *broker* MQTT, *Mosquitto*, ele é utilizado para publicar as mensagens do módulo de “Coletor de Dados” da estação, este módulo é um software implementado em python na EMA simulada que gera dados randomizados e os encaminha para um tópico do *broker* como um arquivo JSON.

Após o envio do arquivo ser efetuado e ele ser devidamente publicado no tópico do *broker*, o módulo de “Gerenciador de Comunicação” que também é um software implementado em python na EMA simulada, irá consumir esta mensagem e por sua vez produz ela em um tópico específico da estação no Gerenciador de Fila de Mensagens que está alocado no servidor.

Com relação à confirmação da entrega de mensagens da estação simulada, o protocolo MQTT por padrão envia suas mensagens utilizando as configurações de *QoS* (*Quality of Service*) como forma de garantir uma comunicação confiável usando o protocolo de transporte TCP. O *QoS* utilizado no MQTT é um acordo que ocorre entre o remetente e o destinatário da mensagem que define o nível de garantia de entrega para uma mensagem, o protocolo MQTT acaba por fornecer três níveis de *QoS*, de 0 a 2 para garantir a confiabilidade da entrega das mensagens (HIVEMQ, 2023)(ZHOU, 2023).

O nível mais baixo, seria o *QoS* 0, ele oferece um mecanismo de melhor esforço onde o remetente da mensagem não espera uma confirmação ou garantia da entrega da mensagem, isso significa que o destinatário da mensagem não confirma o recebimento e o remetente não armazena ou retransmite a mensagem, este nível é comumente chamado de “dispare e esqueça”, sendo semelhante ao protocolo TCP, onde a mensagem é enviada sem acompanhamento e confirmação da mesma, ainda assim, é possível perder as mensagens dependendo da estabilidade da conexão, ou se ela for fechada ou redefinida, dito isso, há o risco de que as mensagens em trânsito ou no buffer do sistema operacional acabem se perdendo, resultando na entrega mal sucedida da mensagem (HIVEMQ, 2023)(ZHOU, 2023).

O nível 1, foca em garantir a entrega da mensagem pelo menos uma vez ao receptor. Quando uma mensagem é publicada utilizando este nível de *QoS*, o remetente mantém uma cópia da mensagem até receber um pacote do destinatário que confirme que o recebimento da mensagem foi recebido. Caso o pacote não tenha recebido dentro de um determinado prazo, ele irá transmitir a mensagem novamente para garantir sua entrega. Em resumo, esta abordagem permite o equilíbrio entre confiabilidade e eficiência, garantindo que a mensagem seja recebida ao menos uma vez, porém pode acabar resultando em mensagens duplicadas (HIVEMQ, 2023)(ZHOU, 2023).

O maior nível é o 2, ele foca em oferecer um nível mais alto de serviço em MQTT, garantindo que cada mensagem seja entregue exatamente uma vez aos destinatários pretendidos e para isso, ele envolve um handshake de quatro partes que ocorre entre o remetente e o destinatário (HIVEMQ, 2023)(ZHOU, 2023). O handshake ocorre conforme ilustrado pela Figura 2.

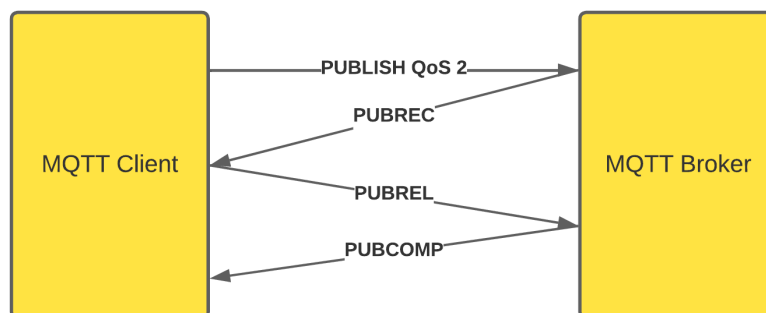


Figura 2. Handshake QoS nível 2 (elaborado pelo autor).

Inicialmente o receptor recebe um pacote *QoS 2 PUBLISH* de um remetente e processa essa mensagem para posteriormente responder ao remetente com um pacote *PUBREC* que confirma o recebimento da mensagem ao reconhecer o pacote *PUBLISH*. Caso o remetente não receber o pacote de confirmação do destinatário, ele enviará novamente a mensagem com um sinalizador duplicado (*DUP*) até receber a confirmação (HIVEMQ, 2023)(ZHOU, 2023).

Após o remetente receber o pacote *PUBREC* do destinatário, o remetente pode descartar com segurança o pacote *PUBLISH* inicial. O remetente irá armazenar o pacote *PUBREC* do receptor e irá responder com um pacote *PUBREL*, com isso, o receptor irá descartar todos os estados armazenados e responderá com um pacote *PUBCOMP* e encerrará o fluxo do *QoS 2*, portanto, garantindo que a mensagem seja entregue exatamente uma vez e sem duplicação (HIVEMQ, 2023)(ZHOU, 2023).

1.3 Sistema Web de Gerenciamento

O Sistema Web de Gerenciamento se encontra no servidor (Figura 1), implementado em PHP que tem a função de permitir que os usuários do sistema tenham acesso aos dados atuais de cada EMA, bem como, aos dados históricos da mesma, podendo inclusive baixá-los em formato CSV para utilização em estudos futuros.

A modelagem do banco de dados utilizada pelo sistema de gerenciamento é ilustrada pela Figura 3, nela é possível ver que o banco de dados possui três tabelas, sendo elas: “usuarios”, “emas” e “observacoes”.

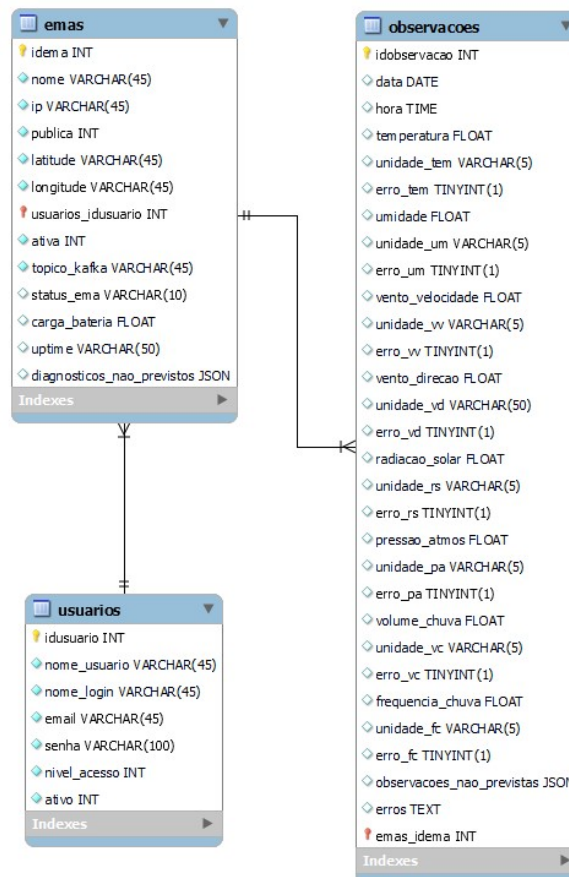


Figura 3. Modelagem do Banco de Dados (elaborado pelo autor).

A tabela de “usuarios” é utilizada para armazenar as informações dos usuários que utilizam o sistema de gerenciamento, ela possui uma relação de um para muitos com a tabela “emas”, devido a um usuário estar vinculado à diversas estações que ele venha a cadastrar.

A tabela “emas” é responsável por guardar as informações das estações que estão sendo cadastradas. Ela, além de se relacionar com a tabela de “usuarios” se relaciona com a tabela de “observacoes”, nesse sentido, uma EMA possui diversas observações meteorológicas.

A tabela de “observacoes” é responsável por guardar toda a informação que for captada pela ema, além das observações meteorológicas, a tabela também consta com os dados de diagnóstico da estação, contando com atributos como a carga da bateria, uptime e afins.

O sistema implementado teve como inspiração o portal REDEMETS (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 assim como a REDEMETS, o sistema de gerenciamento permite o acesso aos dados meteorológicos por

meio de um mapa que facilita a localização de cada estação disponível e conforme a Figura 4.

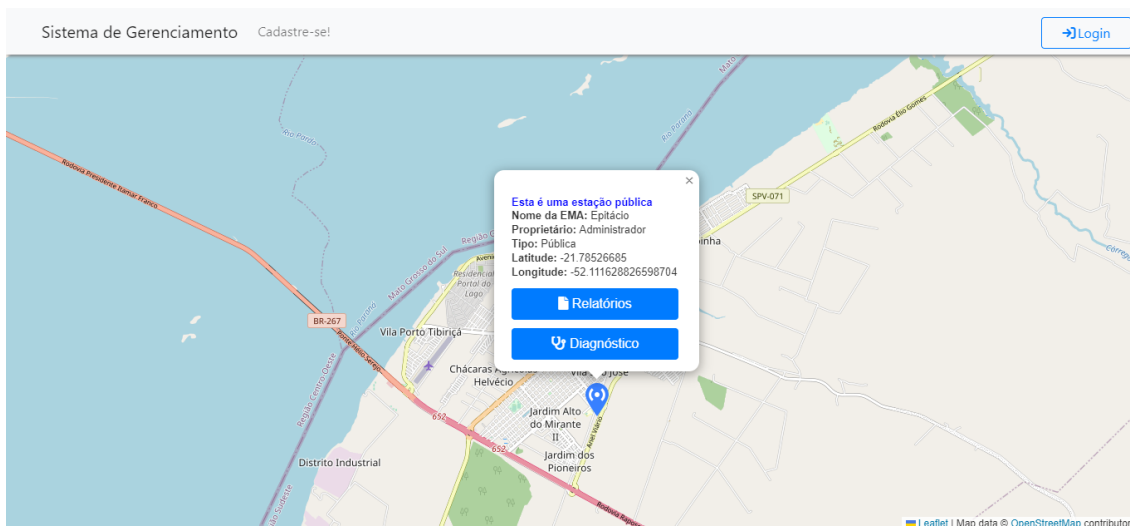


Figura 4. Página Inicial do Sistema de Gerenciamento (elaborado pelo autor).

O Sistema possui dois (2) níveis de acesso, sendo eles o Administrador e o Cliente, apesar disso também é possível utilizá-lo mesmo não sendo um usuário autenticado.

No nível de acesso do Administrador, é possível manter os dados de novos usuários informando seus dados pessoais e o nível de acesso do novo usuário, também é possível manter dados das EMAs informando a localização (latitude e longitude), um nome para a estação e se deseja que os dados dela sejam públicos para qualquer pessoa consultar, após o cadastro da EMA, um tópico será gerado no Gerenciador de Fila de Mensagens que se encontra no servidor, além disso o Administrador pode consultar os dados dos relatórios das EMAs cadastradas no sistema.

No nível de acesso do Cliente, ele pode manter dados de suas EMAs apenas e consultar os relatórios recebidos pela estação.

Usuários não autenticados no sistema podem usá-lo e ter acesso à EMAs públicas, isso irá permitir que o usuário possa consultar dados dos relatórios recebidos pela estação. Além disso, usuários não autenticados também poderão se cadastrar no sistema.

A distinção de se uma EMA é pública, privada ou se ela é sua ocorre de acordo com a cor dela. Se um usuário não autenticado estiver utilizando o sistema, apenas EMAs públicas estarão visíveis para ele, se ele estiver autenticado, uma estação pública que não é dele terá a cor azul, uma estação dele terá a cor verde e no caso do Administrador que pode ver EMAs privadas de outros usuários, será a cor vermelha, conforme a Figura 5.

Todos os usuários (autenticados ou não) ao consultarem os relatórios, podem efetuar o download do histórico de relatórios da estação conforme apresentado na Figura 6.

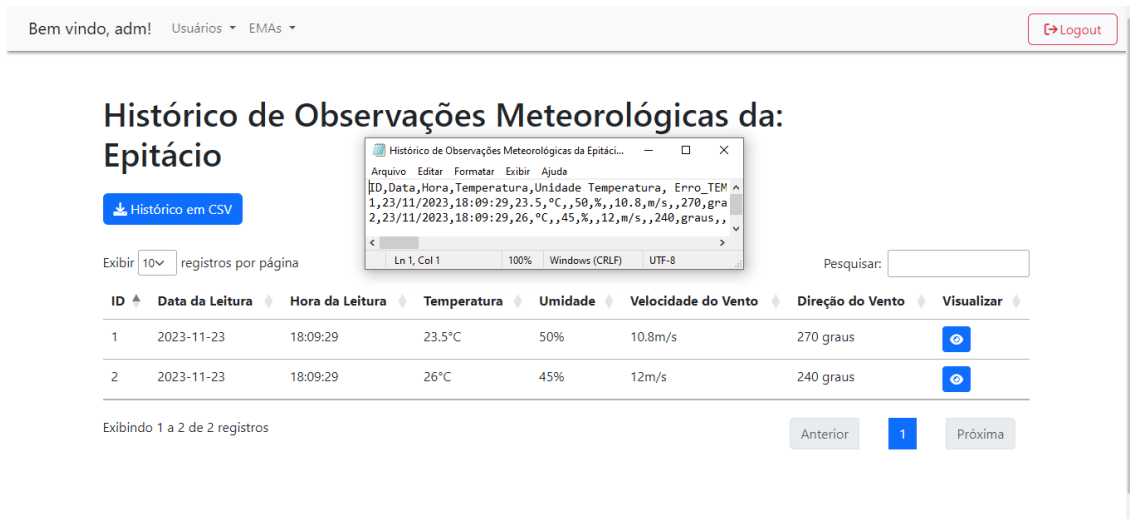


Figura 6. Página Inicial de Histórico de Relatórios de uma EMA (elaborado pelo autor).

Também é possível consultar um relatório em específico e efetuar seu download conforme apresentado na Figura 7.

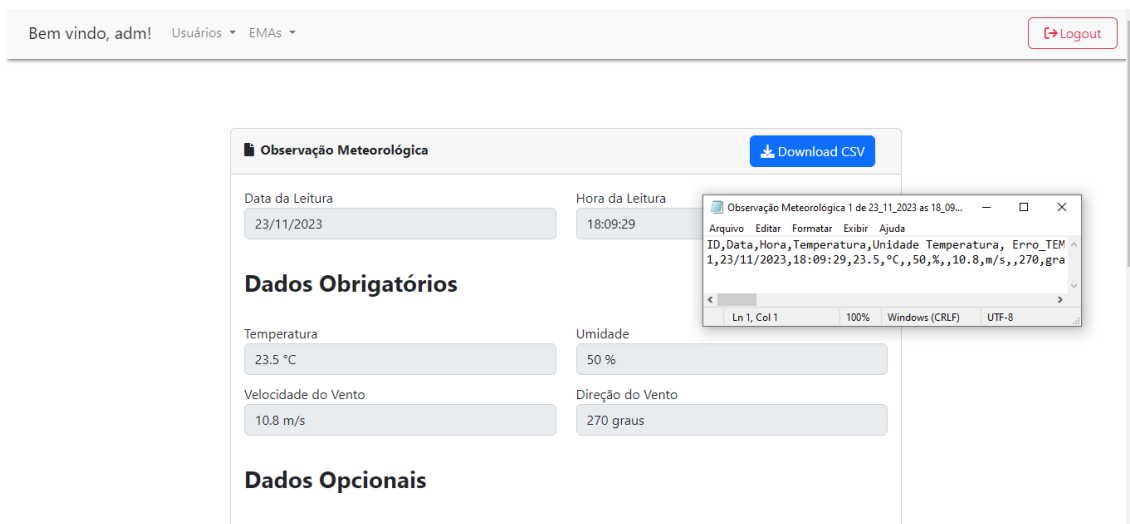


Figura 7. Página de consulta de Relatório (elaborado pelo autor).

O Sistema de Gerenciamento também permite que o usuário saiba se algum dos sensores apresenta algum valor fora do padrão informando os valores que há suspeita de erro conforme a Figura 8.

Bem vindo, adm!
 Usuários ▾
 EMAs ▾
 [Logout](#)

Observação Meteorológica

Download CSV

Data da Leitura

23/11/2023

Hora da Leitura

18:09:29

Dados Obrigatórios

Temperatura

230000 °C

Valor com erro

Umidade

50 %

Velocidade do Vento

10.8 m/s

Direção do Vento

270 graus

Dados Opcionais

Radiação Solar

Pressão Atmosférica

Figura 8. Exemplo de Relatório com erro na Temperatura (elaborado pelo autor).

3. Protocolo de Comunicação

Para permitir a comunicação entre EMAs e o servidor foi necessário desenvolver um protocolo de comunicação que atua na camada de aplicação do modelo ISO/OSI. Ele foi denominado MIAP (*Meteorological Information Application Protocol*) ou (Protocolo de Aplicação de Informações Meteorológicas) e foi desenvolvido para trabalhar junto ao gerenciador de fila de mensagens Kafka, definindo como a comunicação entre as EMAs e servidor deve ser realizada. A Figura 1 ilustra as camadas de rede e as tecnologias utilizadas em cada camada.

Tabela 1. Camadas de rede e tecnologias utilizadas (elaborado pelo autor).

Camada de Aplicação	MIAP + KAFKA
Camada de Transporte	TCP
Camada de Rede	IP
Camada Física + Enlace	Ethernet / 3G / 4G / etc...

3.1 Mensagens

Devido à heterogeneidade das EMAs, principalmente devido a elas não possuírem os mesmos sensores, as mensagens do protocolo MIAP utilizarão formato JSON com objetivo de facilitar a leitura das mensagens, utilizando seu modelo de escrita versátil para inserir informações diversas, mesmo que não previstas pelo protocolo.

As mensagens MIAP serão enviadas entre EMAs e servidor utilizando o Kafka, que foi selecionado por possuir características importantes para esta tarefa, como por exemplo, o fato de ser um sistema de fila de mensagem tolerante a falhas permitindo

hospedar diversos agentes do Kafka em servidores distintos, a escalabilidade que permite adicionar partições para os tópicos das mensagens para distribuir o carregamento da mensagem de forma uniforme (AMAZON, 2023).

A escolha do formato JSON para a transmissão dos dados se deve ao fato de sua sintaxe ser mais compacta, se comparado a outras tecnologias que possuem o mesmo objetivo (ex: XML), proporcionando uma economia de dados que pode ser vantajosa em termos de desempenho de transmissão. Os documentos JSON também são mais simples para o ser humano compreender, se comparados ao XML. Além disso, é considerado um formato flexível, sendo compatível com diversos tipos de dados (AMAZON, 2023).

O protocolo possui um tipo de mensagem apenas, porém possui dois atributos importantes. O primeiro atributo (“observacao”) possui os dados meteorológicos coletados pelas EMAs e enviados ao servidor, como por exemplo, temperatura do ar, direção e velocidade do vento, umidade relativa do ar e outras informações pertinentes. O segundo atributo “diagnostico”, possui dados de diagnóstico de uma EMA que possibilita aos operadores do sistema conhecer o estado atual de cada EMA, como por exemplo, o status, carga da bateria, *uptime*, entre outras informações pertinentes.

As mensagens são enviadas das EMAs para o servidor em intervalos de cinco (5) minutos. A Tabela 2 representa o formato da mensagem utilizado pelo protocolo.

Tabela 2. Formato da Mensagem do Protocolo (elaborado pelo autor).

Mensagem do Protocolo
<pre>{ "observacao":{ "topico":"", "data_leitura":"", "hora_leitura":"", "obrigatorio":{ "temperatura":{ "unidade":"", "valor":0 }, "umidade":{ "unidade":"", "valor":0 }, "velocidade_vento":{ "unidade":"", "valor":0 }, "direcao_vento":{ "unidade":"", "valor":0 } } } }</pre>

```

    }
  },
  "opcional":{
    "radiacao_solar":{
      "unidade":"",
      "valor":0
    },
    "pressao_atmos":{
      "unidade":"",
      "valor":0
    },
    "volume_chuva":{
      "unidade":"",
      "valor":0
    },
    "frequencia_chuva":{
      "unidade":"",
      "valor":0
    }
  },
  "observacoes_nao_previstas":{

  }
},
"diagnostico":{
  "status_ema":"",
  "carga_bateria":0,
  "uptime":"",
  "diagnosticos_nao_previstos":{

  }
}
}

```

Considerando os trabalhos de Finholdt et al. (2011); Munandar et al. (2017); Pont, Santos e Uggioni (2021) e Carmo et al. (2022), com suas respectivas EMAs desenvolvidas, foi possível definir quais atributos seriam considerados obrigatórios e opcionais na mensagem que possui os dados meteorológicos coletados pelas EMAs e dados de diagnóstico da estação.

Como dados obrigatórios, foram selecionadas as variáveis meteorológicas a seguir:

- **Temperatura:** É uma das variáveis fundamentais por afetar em muitos aspectos da vida cotidiana como o conforto humano, agricultura e ecossistemas, com isso o seu acompanhamento permite a detecção de mudanças climáticas e identificar tendências de aquecimento ou resfriamento.

- **Umidade:** A umidade do ar influencia na sensação térmica que desempenha um papel essencial para a formação das nuvens e precipitação, logo, monitorar esta variável é vital para previsões de chuva, avaliação da seca e risco de incêndios florestais, por exemplo.
- **Velocidade do Vento:** A velocidade do vento é uma variável crucial para a aviação, navegação e segurança de estruturas como edifícios e pontes, também é utilizada na previsão de tempo para determinar a intensidade de sistemas meteorológicos como as tempestades.
- **Direção do Vento:** A direção do vento pode fornecer informações sobre de onde o vento está vindo, isso é importante para a navegação e aviação, também é importante para monitorar a qualidade do ar e a previsão das condições climáticas locais, permitindo saber se há um vento muito forte vindo ou uma frente fria, por exemplo.

Como dados opcionais foram selecionados as variáveis meteorológicas a seguir:

- **Radiação Solar:** A radiação solar é uma fonte de energia que desempenha um papel fundamental nos processos climáticos e meteorológicos, seu registro nos ajuda a compreender os padrões de aquecimento e resfriamento da atmosfera e nos auxilia a prever a energia disponível para processos naturais e atividades humanas como a geração de energia solar.
- **Pressão Atmosférica:** A pressão atmosférica é utilizada para prever mudanças do tempo. Variações na pressão indicam a chegada de sistemas meteorológicos como frentes frias e quentes.
- **Frequência de Chuva:** A frequência da chuva é uma variável importante para entender os padrões sazonais de precipitação e identificar períodos de seca ou chuvas intensas, o que acaba por ser uma informação crucial para a agricultura, gestão de recursos hídricos e prevenção de inundações.
- **Volume de Chuva:** O volume da chuva está relacionado ao impacto das chuvas intensas, inundações e deslizamentos de terra.

Os dados não previstos seriam inseridos no campo de “`nao_previstos`” do arquivo JSON, permitindo que as estações ainda assim possam enviar dados de sensores que não estão previstos no JSON que será utilizado para transmitir a mensagem.

Para validar os arquivos recebidos, é utilizado o JSON Schema, a mensagem possui um Schema correspondente que se encontra no **ANEXO I**.

3.2 Formalização do protocolo

A Figura 9 apresenta um diagrama de estados da Mensagem do Protocolo, com ele é possível ver a ilustração de como é efetuado o processo do envio da mensagem pelo protocolo.

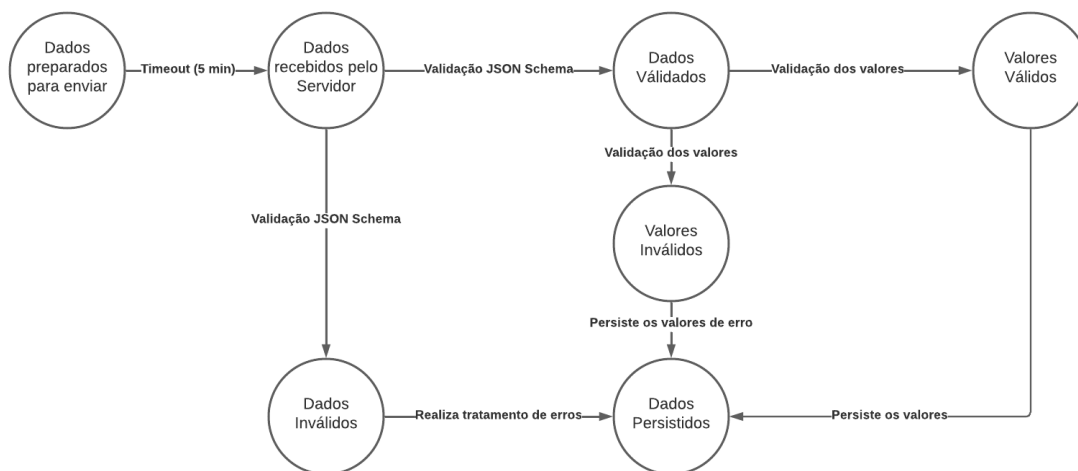


Figura 9. Diagrama de Estados Finitos da Mensagem do Protocolo (elaborado pelo autor).

Após os dados serem coletados pelos sensores da EMA, os dados estão preparados para o envio, após cinco (5) minutos, o envio é efetuado e os dados são recebidos pelo servidor.

No servidor, os dados recebidos são validados pelo JSON Schema, caso o formato do arquivo esteja válido de acordo com o Schema, os valores são verificados para determinar se algum deles esteja fora do padrão, em caso afirmativo, os valores serão persistidos com o erro informando qual sensor apresentou o erro. Caso nenhum sensor possua erro, os dados serão persistidos normalmente.

Caso o formato dos dados esteja inválido de acordo com o JSON Schema, o protocolo dará início ao tratamento de erros e em seguida irá persistir os dados informando que o formato está inválido de acordo com o Schema.

4. Conclusões

Considerando o desenvolvimento do trabalho, foi desenvolvido uma EMA simulada implementada em uma máquina virtual Raspbian. A EMA simulada, utiliza programas desenvolvidos em Python que simulam as mensagens das observações meteorológicas coletadas. Estes são representados pelo “Coletor de Dados” e o “Gerenciador de Comunicação”, ilustrados pela Figura 1.

O módulo de “Coletor de Dados” utiliza dados de clima criados de forma randômica. Estes são transformados em uma mensagem JSON, padronizada pelo protocolo de comunicação desenvolvido, que será enviada para um tópico MQTT da EMA, simulando o processo da coleta dos dados captados pelo sensor da estação. Ao receber esta mensagem, o módulo de “Gerenciador de Comunicação” recebe a mensagem do tópico MQTT e a envia para um tópico específico do gerenciador de fila de mensagens (Kafka) que está no servidor.

No servidor, o módulo de “Consumidor”, recebe a mensagem do tópico do

Kafka e verifica se ele está de acordo com o JSON Schema do **ANEXO I**, onde caso ele esteja de acordo, o protocolo persiste os dados no banco de dados de forma comum e caso contrário, ele irá tratar a observação inteira como um erro, porém irá persistir os dados no banco de dados, para serem visualizados no sistema de gerenciamento que está no servidor da mesma forma.

Há também casos onde os dados podem possuir um valor incomum e de modo automático o módulo de “Consumidor” pode indicar que este valor é um erro se ele passar de um determinado valor e irá persistir aquela variável meteorológica indicando que ela apresenta algum erro, onde pelo sistema de gerenciamento, será indicado pela cor em vermelho e a mensagem de “Valor com erro” conforme a Figura 7.

Com relação às dificuldades encontradas durante o decorrer do trabalho, uma delas seria o fato de que boa parcela dos trabalhos envolvendo esta temática, geralmente focam muito em como criar uma EMA, não se encontra muitos trabalhos que se preocupam em desenvolver uma arquitetura que cuida do tratamento de erros e que permite a abertura para EMAs que utilizem protocolos distintos, como é o caso da arquitetura utilizada no trabalho. Apesar disso, a REDEMET acabou servindo de grande inspiração principalmente para o sistema de gerenciamento com relação à sua interface.

Outro dificuldade encontrada foi relacionada aos dados de Diagnóstico da EMA, onde anteriormente era planejado que o protocolo iria utilizar mais de uma mensagem, onde uma seria para as observações meteorológicas e a outra seria uma mensagem de diagnóstico em tempo real que buscava utilizar um serviço de *Remote Procedure Call* (RPC). Nesse sentido, por meio do sistema web, o usuário faria uma requisição para a EMA e ela por sua vez, o responderia com os dados de diagnóstico da estação. Como essa ideia foi abandonada, as mensagens foram mescladas fazendo com que em uma só mensagem o protocolo apresente tanto os dados de uma observação meteorológica quanto os dados de diagnóstico de uma estação, o que acaba por ser uma vantagem permitindo com que em uma só mensagem, o protocolo satisfaça essa necessidade dos dados de diagnóstico.

Para trabalhos futuros, é sugerido explorar a utilização de uma maior quantidade de nós Kafka em cluster para promover maior tolerância a falhas e/ou capacidade de tratamento de mensagens para uma grande quantidade de EMA. Outra sugestão é utilizar algoritmos preditivos de clima em conjunto de dados coletados por um conjunto de EMAs para prever as possíveis mudanças climáticas em uma determinada região.

Referências Bibliográficas

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.

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 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.

PONT, Isabele 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.

Qual é a diferença entre o Kafka e o RabbitMQ?. Amazon, 2023. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-rabbitmq-and-kafka/>. Acesso em 26 de set. de 2023.

Qual a diferença entre JSON e XML?. Amazon, 2023. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-json-xml/>. Acesso em 21 de ago. de 2023.

Raspberry Pi for home: The big computer that fits in your hand. RASPBERRY PI FOUNDATION, 2023. Disponível em: <https://www.raspberrypi.com/for-home/>. Acesso em 19 de ago. 2023.

Raspberry Pi for industry: Certified. Low cost. Powerful. Production-ready. RASPBERRY PI FOUNDATION, 2023. Disponível em: <https://www.raspberrypi.com/for-industry/>. Acesso em 19 de ago. 2023.

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

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

What is MQTT Quality of Service (QoS) 0,1 & 2? - MQTT Essentials: Part 6. HiveMQ, 2023. Disponível em: <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>. Acesso em: 20 de ago. de 2023.

YNOUE, R. Y et al. Meteorologia: noções básicas. 1.ed. São Paulo: Oficina de Textos, 2017. v.1.

ZHOU, Zibo. Introduction to MQTT QoS 0, 1, 2. EMQX, 2023. Disponível em: <https://www.emqx.com/en/blog/introduction-to-mqtt-qos>. Acesso em: 20 de ago. de 2023.

ANEXO I - JSON Schema da Mensagem

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "observacao": {
      "type": "object",
      "properties": {
        "topico": {
          "type": "string"
        },
        "data_leitura": {
          "type": "string"
        },
        "hora_leitura": {
          "type": "string"
        },
        "obrigatorio": {
          "type": "object",
          "properties": {
            "temperatura": {
              "type": "object",
              "properties": {
                "unidade": {
                  "type": "string"
                },
                "valor": {
                  "type": "number"
                }
              },
              "required": ["unidade", "valor"]
            },
            "umidade": {
              "type": "object",
              "properties": {
                "unidade": {
                  "type": "string"
                },
                "valor": {
                  "type": "number"
                }
              },
              "required": ["unidade", "valor"]
            },
            "velocidade_vento": {
              "type": "object",
              "properties": {
                "unidade": {
                  "type": "string"
                },
                "valor": {
                  "type": "number"
                }
              },
              "required": ["unidade", "valor"]
            },
            "direcao_vento": {
              "type": "object",
              "properties": {
                "unidade": {
                  "type": "string"
                },
                "valor": {
                  "type": "number"
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```

        }
    },
    "required": ["unidade", "valor"]
}
},
"required": ["temperatura", "umidade",
"velocidade_vento", "direcao_vento"]
},
"opcional": {
    "type": "object",
    "properties": {
        "radiacao_solar": {
            "type": "object",
            "properties": {
                "unidade": {
                    "type": "string"
                },
                "valor": {
                    "type": "number"
                }
            }
        },
        "required": ["unidade", "valor"]
    },
    "pressao_atmos": {
        "type": "object",
        "properties": {
            "unidade": {
                "type": "string"
            },
            "valor": {
                "type": "number"
            }
        },
        "required": ["unidade", "valor"]
    },
    "volume_chuva": {
        "type": "object",
        "properties": {
            "unidade": {
                "type": "string"
            },
            "valor": {
                "type": "number"
            }
        },
        "required": ["unidade", "valor"]
    },
    "frequencia_chuva": {
        "type": "object",
        "properties": {
            "unidade": {
                "type": "string"
            },
            "valor": {
                "type": "number"
            }
        },
        "required": ["unidade", "valor"]
    }
}
},
"observacoes_nao_previstas": {
    "type": "object",
    "additionalProperties": true
}

```

```

        },
        "required": ["topico", "data_leitura", "hora_leitura",
"obrigatorio", "opcional"]
    },
    "diagnostico": {
        "type": "object",
        "properties": {
            "status_ema": {
                "type": "string"
            },
            "carga_bateria": {
                "type": "number"
            },
            "uptime": {
                "type": "string"
            },
            "diagnosticos_nao_previstos": {
                "type": "object",
                "additionalProperties": true
            }
        },
        "required": ["status_ema", "carga_bateria", "uptime"]
    },
    "required": ["observacao", "diagnostico"]
}

```