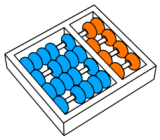

Monitoramento de colmeias
com Internet das Coisas

C. V. M. Gomes *J. F. Theodoro* *L. Q. Borges*

J. V. Dos S. Oliveira *L. F. Bittencourt*

Relatório Técnico - IC-PFG-24-26

Projeto Final de Graduação

2024 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE COMPUTAÇÃO

Monitoramento de colmeias com Internet das Coisas

Caio V. M. Gomes* José F. Theodoro** Leonardo Q. Borges***
João V. Dos S. Oliveira**** Luiz F. Bittencourt*****
[1]

Resumo

Este é o relatório do projeto de Conclusão de Curso realizado em parceria com o Prof. Roberto Greco do Instituto de Geociências sob tutoria do Prof. Luiz F. Bittencourt do Instituto de Computação, cujo objetivo é desenvolver um sistema para coleta de dados de colmeias que serão instalados em escolas para aprendizado dos estudantes. O sistema faz a coleta de dados do ambiente da colmeia via Wi-fi com uma placa embarcada, mantendo a segurança das abelhas e a integridade da placa. O projeto teve quatro placas fornecidas, além dos sensores, pelo Prof. Fabiano Fruett da Faculdade de Engenharia Elétrica e de Computação. Os sensores realizam medição da temperatura, umidade, som, pressão e proximidade. Também é realizada a coleta de dados de condições climáticas provenientes da API OpenWeatherMap.

1 - Introdução

O presente relatório detalha a continuidade do projeto de Monitoramento de Colmeias utilizando Internet das Coisas (IoT), iniciado em 2022 por alunos do Instituto de Computação da UNICAMP, em colaboração com o Prof. Roberto Greco, do Instituto de Geociências, e o Prof. Fabiano Fruett, da Faculdade de Engenharia Elétrica e de Computação, sob orientação do Prof. Luiz Fernando Bittencourt. O objetivo inicial era desenvolver um sistema capaz de coletar dados ambientais de colmeias utilizando sensores para medir temperatura, umidade e som, transmitindo essas informações via Bluetooth para um aplicativo móvel. A primeira versão do sistema, baseada em uma placa ESP32, buscava tanto facilitar o aprendizado dos alunos quanto contribuir para a preservação das abelhas, com a instalação de colmeias no assentamento Milton Santos, em Americana, SP para o auxílio da comunidade e das famílias residentes, polinizando a vegetação local.

Após a implementação inicial [18], surgiram desafios técnicos significativos, como problemas de conectividade e limitações dos sensores. Para solucioná-los, o projeto contou com o suporte do Prof. Fabiano Fruett, que sugeriu uma revisão completa do hardware. Em 2023 [19], foi desenvolvida uma nova versão do sistema, utilizando a placa Raspberry Pi Pico

[1] * : c214192@dac.unicamp.br
**: j219081@dac.unicamp.br
***: l177829@dac.unicamp.br
****: j199815@dac.unicamp.br
*****: bit@unicamp.br

W, equipada com um microprocessador RP2040, 2 MB de memória flash e conectividade Wi-Fi. Além disso, novos sensores foram integrados, incluindo sensores de pressão atmosférica, proximidade e umidade, ampliando a capacidade de monitoramento. Essa nova versão do sistema também permitiu a transmissão dos dados via Wi-Fi, garantindo uma comunicação mais estável e facilitando o monitoramento remoto [10].

Em 2023, o projeto teve continuidade sob a orientação do Prof. Luiz F. Bittencourt. Uma nova versão da placa foi projetada, incorporando sensores de temperatura, umidade, som e proximidade. Essa nova placa, representada na figura 1, foi disponibilizada pelo Prof. Fabiano Fruett e utilizava tecnologia IoT para comunicação via Wi-Fi com um aplicativo Android. A partir desse ponto, o projeto foi dividido em duas frentes: uma voltada para a programação e ajustes do microcontrolador e outra para o desenvolvimento do aplicativo.

Nesse estágio, o projeto dividiu-se em duas frentes: programação e ajustes no microcontrolador e desenvolvimento do aplicativo Android para visualização dos dados. Contudo, o sistema ainda enfrentou desafios, como intermitências na conexão Wi-Fi, memória insuficiente na placa para armazenamento prolongado de dados e dificuldades técnicas com os sensores de som e BME680 [8] devido às limitações do software implementado . [PFG-23-28.pdf].

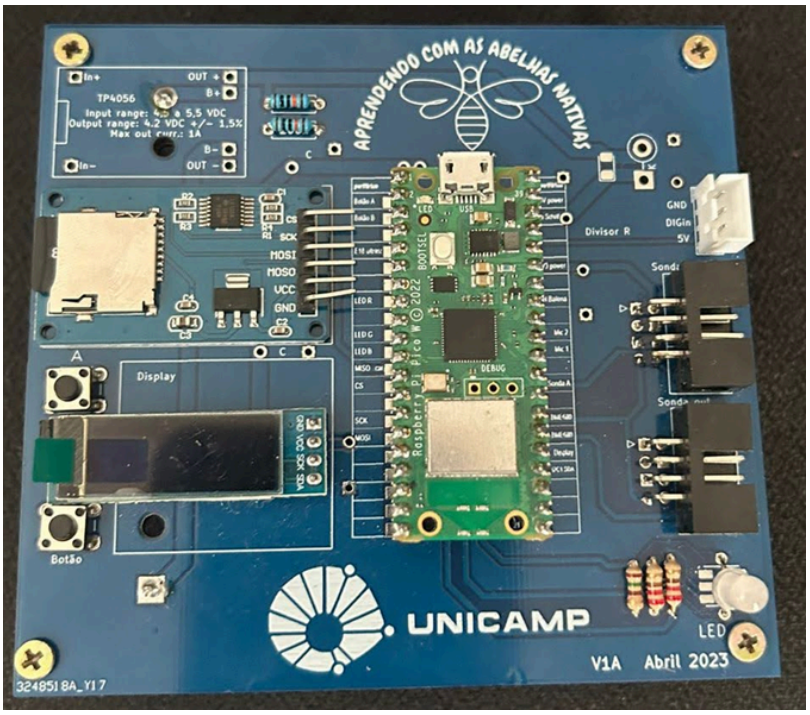

Figura 1: Placa elaborada para monitoramento de colmeias nos projetos dos anos 2022 e 2023

No primeiro semestre de 2024, outro grupo de alunos realizou o último avanço no projeto. Após novo contato com o Prof. Fabiano Fruett, decidiu-se adotar a placa *BitDogLab* [7], uma solução de código aberto mais robusta e amplamente utilizada em projetos educacionais semelhantes. A figura 2 apresenta a placa *BitDogLab*, que não apenas manteve as funcionalidades das versões anteriores, mas também trouxe melhorias significativas, como maior capacidade de processamento e suporte à comunicação por ondas de radiofrequência LoRa [22]. Essa tecnologia permite conexões em distâncias substancialmente maiores, o que seria especialmente vantajoso para uma possível implementação de multicast em um projeto futuro, possibilitando que várias placas realizem coletas simultâneas e se comuniquem com uma placa central, responsável por enviar os dados ao Firebase [2][3].

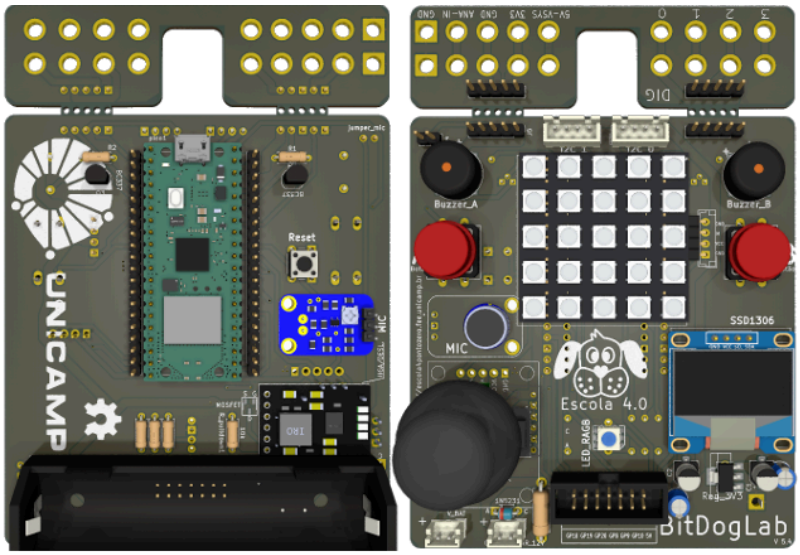

Figura 2: Placa *BitDogLab* na revisão v5.4 utilizada no projeto do primeiro semestre e no presente projeto

Apesar das vantagens oferecidas pela nova placa, o grupo encontrou dificuldades em integrar o hardware ao aplicativo desenvolvido anteriormente. Para superar esses obstáculos, foi criada uma nova aplicação web utilizando o framework React [5]. Essa aplicação, integrada ao banco de dados Firebase, foi projetada para exibir os dados coletados em tempo real, reunindo todas as informações em um único gráfico, que apresenta os dados de forma consolidada [1].

Posteriormente, quando nosso grupo assumiu a continuidade do projeto, o Prof. Roberto Grecco avaliou a aplicação web [11] existente e apontou limitações em sua funcionalidade. Ele destacou que a exibição dos dados de maneira consolidada em um único

gráfico dificultava análises mais detalhadas e a compreensão separada de cada métrica coletada, sobretudo devido às diferentes escalas dos dados analisados. Além disso, a ausência de recursos interativos limitava a experiência do usuário e o potencial de uso para fins educacionais e de pesquisa. A figura 3 apresenta como era a antiga interface da aplicação.

Diante dessas observações, o Prof. Roberto Greco solicitou uma reformulação completa na estrutura visual da aplicação. O objetivo era permitir a separação e visualização individual dos diferentes tipos de dados coletados, além de oferecer ferramentas interativas que ampliassem a capacidade de análise. Ele também sugeriu a inclusão de novas medições, de forma a enriquecer o sistema com informações mais abrangentes e relevantes.

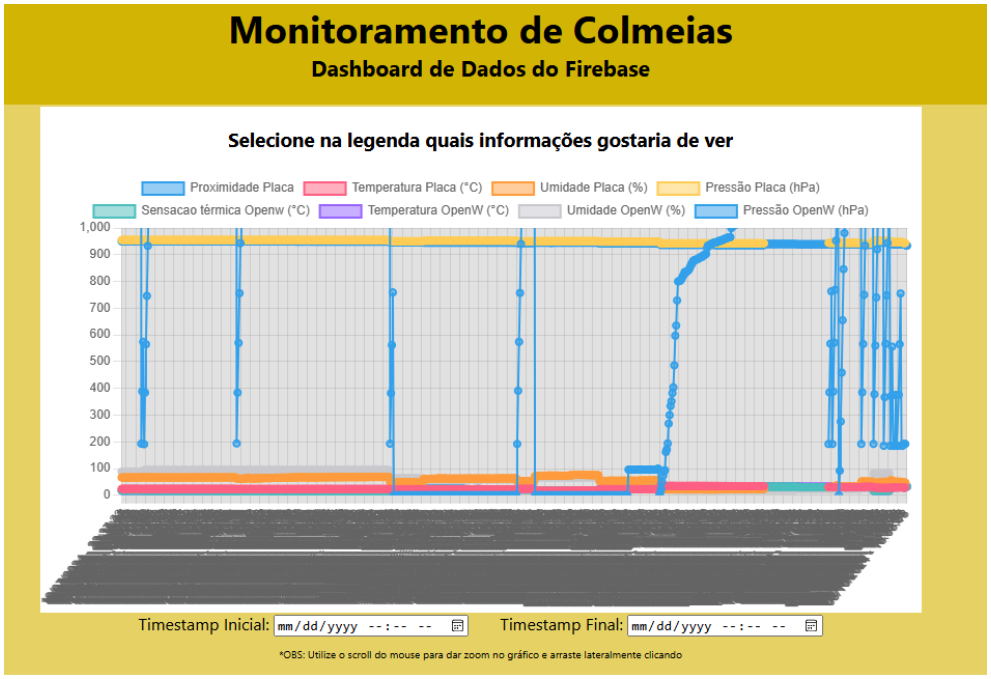

Figura 3: Interface da aplicação web desenvolvida pelo grupo do primeiro semestre de 2024

Sendo assim, este relatório aborda as melhorias na aplicação web para o monitoramento de colmeias, com foco na reformulação da estrutura visual dos gráficos e na adição de novos parâmetros de medição. As mudanças visam permitir a análise separada dos dados e introduzir recursos interativos, tornando o sistema mais versátil e adequado para fins educacionais e de pesquisa.

O relatório se encontra dividido nas seções: Objetivos, que descreve os objetivos que foram definidos para o escopo do projeto; Metodologia Colaborativa e Contínua, descreve as

etapas e métodos utilizados para realizar o projeto; As seções Placa, Armazenamento em Nuvem, Obtenção de Dados e Aplicação Web descrevem com mais detalhes os métodos empregados e as ações tomadas para cada desenvolvimento; Resultado, que apresenta os principais resultados alcançados até o momento e discute os desafios encontrados durante o desenvolvimento; Conclusão e Trabalhos Futuros, no qual resume as principais conclusões do projeto e detalha os próximos passos, incluindo melhorias planejadas e novas funcionalidades e Referências, com uma lista das fontes e documentações utilizadas como suporte ao desenvolvimento do projeto.

2 - Objetivos

O objetivo principal deste projeto é aprimorar a aplicação web utilizada para o monitoramento de colmeias, com base no conceito de Internet das Coisas (IoT). A primeira etapa consiste em reformular a estrutura visual dos gráficos, de modo a permitir a separação e análise individual dos diferentes tipos de dados coletados, como temperatura, umidade e som. Essa mudança visa proporcionar uma visão mais clara e detalhada de cada variável, facilitando a interpretação dos dados. Além disso, o projeto inclui a adição de novos parâmetros de medição, com o intuito de expandir as informações disponíveis e aumentar a capacidade de análise do sistema, tornando-o mais robusto e preciso.

Outro objetivo importante é a introdução de recursos interativos na aplicação, que proporcionarão uma experiência mais dinâmica e intuitiva para os usuários. A ideia é permitir que os usuários possam interagir com os gráficos e dados de forma mais envolvente, facilitando a exploração e a interpretação das informações. Além disso, as melhorias visam não apenas atender às demandas acadêmicas e de pesquisa, mas também aumentar a utilidade do sistema em contextos educacionais, como a sua implementação em escolas. Com essas alterações, espera-se tornar o sistema mais versátil, oferecendo uma ferramenta poderosa tanto para o monitoramento das colmeias quanto para atividades didáticas, possibilitando uma exploração mais detalhada e personalizada dos dados coletados.

3 - Metodologia Colaborativa e Contínua

O desenvolvimento deste projeto adotou uma abordagem inspirada em metodologias ágeis, priorizando a colaboração constante entre os membros da equipe, feedback frequente e uma estrutura de trabalho interativa. A equipe foi organizada com base nas habilidades e

expertises de cada integrante, permitindo uma divisão eficiente de responsabilidades e maximizando a produtividade.

3.1 - Estrutura de Trabalho

Os membros foram divididos em dois grupos principais:

1. **Equipe de Desenvolvimento de Software**: Formada por integrantes com maior familiaridade com programação e desenvolvimento de sistemas. Essa equipe foi responsável pela implementação do código, tanto na programação do firmware da placa *BitDogLab* quanto na criação da interface web para visualização dos dados.
2. **Equipe de Comunicação e Avaliação**: Composta por membros focados na interação contínua com os orientadores e na análise do progresso do projeto. Essa equipe:
    - Realizou reuniões regulares com o professor orientador para alinhar expectativas e discutir possíveis desafios.
    - Monitorou a integração das diferentes partes do sistema, sugerindo melhorias na arquitetura e no funcionamento.
    - Testou o sistema em diversos cenários, levantando pontos de melhoria e colhendo feedbacks para ajustes.

3.2 - Ciclo Iterativo de Trabalho

O grupo seguiu um ciclo iterativo, semelhante ao modelo **Scrum**, com entregas incrementais em "sprints". Cada ciclo de trabalho foi dividido nas seguintes etapas:

1. **Planejamento**: Durante as reuniões iniciais de cada ciclo, a equipe definiu objetivos claros e tangíveis, como a integração de um novo sensor, a melhoria na estabilidade da comunicação com o Firebase ou o ajuste na interface web.
2. **Execução**: A equipe de desenvolvimento focava em implementar as mudanças planejadas. Simultaneamente, a equipe de comunicação e avaliação realizava testes paralelos em funcionalidades já entregues.
3. **Revisão**: No final de cada ciclo, eram realizadas reuniões para apresentar os resultados obtidos, discutir problemas enfrentados e identificar melhorias.
4. **Retroalimentação**: Com base no feedback das revisões, novas prioridades eram definidas, refinando o escopo e adaptando os objetivos para os próximos ciclos.

3.3 - Benefícios da Metodologia

A adoção da metodologia colaborativa trouxe diversos benefícios que foram essenciais para o sucesso do projeto. Em primeiro lugar, a abordagem permitiu um foco claro nas especialidades de cada integrante da equipe. Ao distribuir as tarefas de acordo com as forças de cada membro, evitamos tanto a sobrecarga de trabalho em algumas áreas quanto a falta de engajamento em outras. Essa especialização possibilitou um desempenho mais eficiente e a otimização do tempo, além de garantir que cada parte do projeto fosse trabalhada com o conhecimento e a experiência necessários.

Outro benefício importante foi o feedback contínuo, promovido pelas interações frequentes entre a equipe e os orientadores. Esse fluxo constante de comunicação ajudou a alinhar o projeto às expectativas, permitindo ajustes rápidos e eficazes. Quando surgiam problemas ou dificuldades, a capacidade de obter orientações imediatas contribuiu para uma resolução mais ágil e eficaz, evitando atrasos ou desvios significativos do planejamento inicial.

A flexibilidade e a adaptação também se destacaram como pontos positivos da metodologia. A organização do trabalho de forma iterativa permitiu que o planejamento fosse ajustado conforme novas demandas ou desafios surgissem ao longo do desenvolvimento. Essa capacidade de adaptação foi crucial, especialmente em um projeto que envolvia várias mudanças e melhorias tecnológicas, permitindo que a equipe respondesse de maneira proativa a imprevistos.

Além disso, a metodologia favoreceu a melhoria contínua. A divisão clara de responsabilidades entre desenvolvimento, avaliação e comunicação garantiu que as entregas fossem constantemente testadas e validadas. Isso não só assegurou que o trabalho estivesse sempre alinhado aos requisitos do projeto, mas também proporcionou uma oportunidade de aprimoramento constante, permitindo que ajustes fossem feitos de forma incremental e eficiente.

4 - Placa

Como mencionado na introdução deste projeto, optamos por dar continuidade ao uso da placa adotada pelo grupo do primeiro semestre de 2024. Dessa forma, a ferramenta de coleta utilizada foi a BitDogLab, uma placa desenvolvida no âmbito do Projeto Escola 4.0 da UNICAMP. A escolha se deve ao fato de que todas as funcionalidades essenciais de coleta da placa continuam operando de maneira eficiente, sendo a principal limitação identificada a exibição dos dados pela aplicação web. Para abordar essa questão, decidimos aproveitar as implementações realizadas pelo grupo anterior, utilizando as melhorias já desenvolvidas como base e direcionando nossos esforços para aprimorar a interface de visualização dos dados.

Consequentemente, boa parte desta seção se fundamenta no que foi descrito no relatório [1] produzido pelo grupo do primeiro semestre de 2024, com adaptações realizadas para contemplar as modificações introduzidas neste semestre.

4.1 - Sensores e Componentes

Abaixo estão listados os componentes que compõem a *BitDogLab*, acompanhados de um resumo de suas funções:

- **Raspberry Pi Pico W**: Equipada com um microcontrolador RP2040, possui 2MB de memória flash, suporta conexões Wi-Fi e Bluetooth, e possui entrada micro USB B com 40 pinos para diferentes funcionalidades.
- **Sensor de Proximidade Ultrassônico E18-D80NK - NPN**: Usado para detectar a entrada e saída de abelhas, com alcance de detecção de 3 a 80 cm.
- **Microfone de Eletreto Módulo MAX4466**: Captura sons ambientes da colmeia, conectado ao GPIO28, permitindo análises acústicas das atividades internas.
- **Sensor Ambiental BME680**: Mede temperatura, umidade, pressão atmosférica e VOCs, fornecendo dados vitais para o estudo do ambiente da colmeia.
- **LED RGB cátodo comum**: Conectado aos GPIOs 12, 13 e 11 para indicação visual através de cores variadas.
- **Botões A e B**: Conectados aos GPIOs 5 e 6, com resistores de pull-up internos para ações de interação manual.
- **Joystick analógico KY023**: Fornece controle manual através dos GPIOs 26 e 27 para os eixos e um botão no GPIO 22.
- **Display OLED (0.96 polegadas)**: Utiliza comunicação I2C, conectado aos GPIOs 14 e 15, para apresentar dados e diagnósticos.

- **Buzzers A e B**: Emitindo sinais sonoros através dos GPIOs 21 e 10, para alertas ou feedback operacional.
- **Matriz de LEDs WS2812B**: Conectada ao GPIO7, usada para efeitos visuais e indicadores de status.

4.2 - Ciclo da Placa

A utilização da placa *BitDogLab* no monitoramento de colmeias segue um ciclo operacional bem definido. Portanto, é importante que, assim como feito no relatório do grupo do primeiro semestre de 2024, esse comportamento seja novamente detalhado para que grupos futuros possam compreender melhor o funcionamento da placa e, assim, corrigir possíveis problemas que venham a surgir. Esse processo pode ser dividido nas seguintes etapas:

1. **Inicialização do Sistema**: Ao ser ligada, a placa realiza uma série de verificações para garantir que os sensores e as conexões de rede estão funcionando corretamente. Esses testes iniciais ajudam a evitar erros durante a coleta de dados, assegurando maior confiabilidade.
2. **Coleta de Dados dos Sensores**: Em intervalos regulares, os sensores da placa realizam medições de variáveis ambientais, como temperatura, umidade, som, gases VOC e movimento. Dados climáticos adicionais também são coletados via API do Openweathermap. A frequência dessas leituras pode ser ajustada conforme a necessidade, especialmente em períodos de maior atividade das abelhas, como na primavera.
3. **Armazenamento em Nuvem com Firebase**: Após a validação, os dados são enviados para o Firebase, uma plataforma de banco de dados em tempo real. Essa solução baseada em nuvem protege as informações contra falhas locais e permite que pessoas autorizadas acessem os dados de forma remota e instantânea caso seja necessário. O Firebase também possibilita a criação de automações, como alertas ou respostas baseadas nas condições observadas nos dados coletados, o que pode vir a ser útil futuramente.
4. **Gerenciamento de Conexões e Comunicação**: A placa aproveita sua conectividade sem fio para realizar operações de troca de dados, utilizando protocolos como HTTP/1.1 para enviar e receber informações. Isso permite integração com sistemas

externos, como plataformas de monitoramento remoto e aplicativos, viabilizando análises e acesso aos dados em dispositivos variados.

Essa abordagem garante que a *BitDogLab* opere de forma confiável e independente, se tornando uma ótima ferramenta de monitoramento de colmeias e contribuindo com informações valiosas para o estudo do comportamento e da saúde das abelhas.

4.3 - Arquitetura de software

O sistema é composto por diversos módulos em micropython [12] que desempenham funções específicas para garantir a operação integrada. O script *main.py* gerencia o fluxo principal do sistema, desde a conexão Wi-Fi até o acionamento da classe de sensores. O módulo *connections_manager.py* utiliza a biblioteca network para estabelecer e manter a conexão Wi-Fi, que é indispensável para a transmissão de dados coletados. Já o *sensors_manager.py*, onde a classe *SensorsManager* está localizada, organiza a inicialização dos sensores, realiza as coletas de dados. Complementando o sistema, o módulo *openweathermap.py* é responsável por acessar dados climáticos externos e oferecer contexto às medições locais, enquanto *http.py* gerencia as requisições HTTP [9] para a comunicação eficiente com APIs externas. Por fim, o módulo *firebase.py* lida diretamente com a sincronização de dados no Firebase, oferecendo métodos para leitura, envio e atualização das informações. A figura 4 apresenta um diagrama que representa a arquitetura do software do sistema.

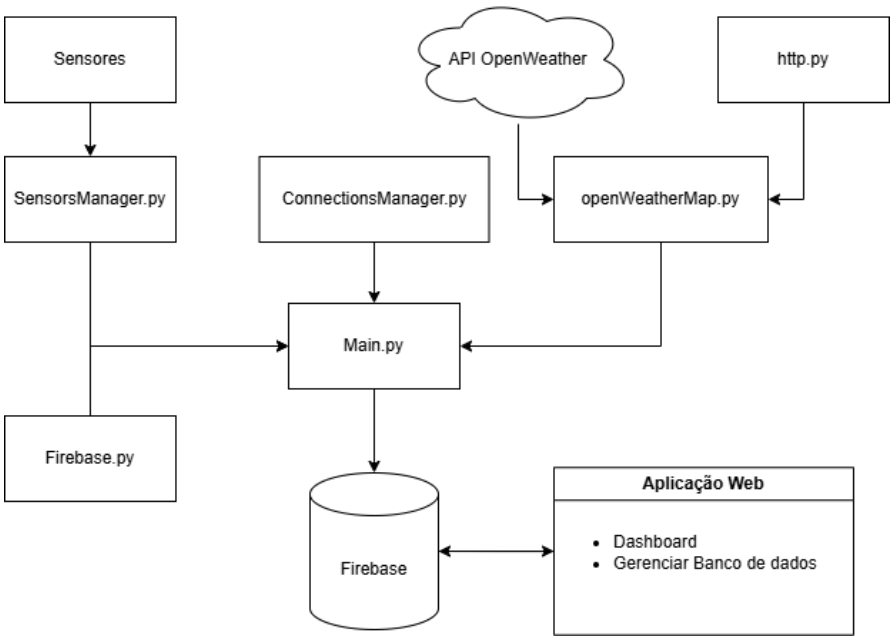

Figura 4: Representação visual da arquitetura de software do sistema

O *SensorsManager* foi projetado para gerenciar múltiplos sensores, com atenção especial ao sensor BME680 [16], que realiza medições de temperatura, pressão, umidade e gases. A configuração do sensor é feita por métodos internos que verificam sua disponibilidade e garantem que ele seja reinicializado em caso de falha, assegurando resiliência ao sistema.

A coleta de dados segue uma estratégia cíclica, gerida pelo *main.py*, que implementa um loop contínuo para definir a frequência das coletas e o envio dos dados. Este loop utiliza temporizadores para manter intervalos regulares e gerar timestamps precisos, criando um registro cronológico das medições. A integração com o Firebase é fundamental para armazenar e disponibilizar os dados em tempo real, utilizando os métodos GET, PUT e PATCH para garantir atualizações eficientes. Além disso, medidas de otimização, como a coleta de lixo (garbage collection), são aplicadas regularmente para gerenciar os recursos limitados da memória da placa. Essa estrutura, com sua organização modular e foco em estabilidade, integra medições locais, dados climáticos externos e sincronização na nuvem, resultando em um sistema robusto para monitoramento de colmeias. O código base deste processo está localizado no link [15].

4.4 - Alterações de Software

Dentre as sugestões feitas pelo professor Roberto Grecco, uma das mudanças mais significativas foi a inclusão de novos dados a serem coletados pela placa, especificamente informações de poluição atmosférica obtidas através da API de poluição do OpenWeatherMap. Essa atualização exigiu alterações no módulo *openweathermap.py*, incorporando uma nova URL destinada à coleta desses dados.

Para implementar a funcionalidade, a URL para acesso aos dados de poluição [21] foi adicionada ao código, permitindo que o sistema requisitasse essas informações diretamente da API. Posteriormente, as medições de poluentes e outros índices de qualidade do ar, foram integradas ao array *data*, que já armazenava informações climáticas previamente coletadas, como temperatura, umidade e pressão atmosférica. Essa abordagem simplifica a gestão dos dados, mantendo todas as informações provenientes do OpenWeatherMap [20] centralizadas e organizadas para envio ao Firebase e posterior análise. Abaixo, exemplificamos a alteração realizada no código:

```python
openweathermap.py
import http as req

API_KEY = "56b0fa0dffacca8c03a3f17abc12de2c"  # Substitua pela sua chave de API do OpenWeatherMap
BASE_URL = "https://api.openweathermap.org/data/2.5/weather?"
POLLUTION_URL = "https://api.openweathermap.org/data/2.5/air_pollution?"

def get_location():
    """Obtém latitude e longitude (simulação para MicroPython)."""
    # Em um dispositivo real, você usaria um módulo GPS ou serviço de localização.
    latitude = -22.9083  # Latitude de Campinas
    longitude = -47.0608 # Longitude de Campinas
    return latitude, longitude

def get_weather_data():
    """Obtém dados climáticos do OpenWeatherMap."""
    lat, lon = get_location()
    complete_url = BASE_URL + "lat={}&lon={}&appid={}".format(lat, lon, API_KEY)
    data = req.http_request(complete_url)
    complete_url = POLLUTION_URL + "lat={}&lon={}&appid={}".format(lat, lon, API_KEY)
    data += req.http_request(complete_url)
    return data
```

Além da integração dos dados de poluição, outra mudança relevante solicitada pelo professor Roberto Grecco foi a modificação do intervalo de tempo das medições no módulo *main.py*. O intervalo entre coletas foi ajustado para 15 minutos, o que também impactou o tempo que o sensor de proximidade permanece coletando dados. Essa alteração foi realizada para otimizar a coleta e o volume de dados coletados a longo prazo, uma vez que o professor indicou que medições a cada 10 segundos eram desnecessárias.

Durante o desenvolvimento, foi identificado um comportamento inesperado nos dados coletados pelo sensor de proximidade. Este apresentava um padrão cumulativo, onde os valores eram somados continuamente até o final do dia. Em vez de refletir a proximidade medida no momento da coleta, o gráfico exibia um valor acumulado, o que comprometia a análise em tempo real. Para corrigir esse problema, ajustamos o método *read_proximity* no

módulo *sensors_manager.py*. A solução implementada consiste em zerar a variável que armazena a contagem de proximidade sempre que os dados são enviados ao Firebase, ou seja, a cada 15 minutos. Essa abordagem eliminou o comportamento cumulativo, garantindo que os gráficos passem a apresentar os valores medidos no momento de cada coleta. O código alterado ficou da seguinte forma:

Python
sensors_manager.py

```python
def __read_proximity(self, sensor):
        initial_time = time.time()
        print("Lendo o sensor de proximidade")
        while True:
            if time.time() - initial_time >= self.timer:
                aux = self.proximity_counter
                self.proximity_counter = 0
                return aux

            if self.__new_day():
                old_proximity_counter = self.proximity_counter
                self.proximity_counter = 0

                return old_proximity_counter

            reading = sensor.value()
            self.proximity_counter += 1 if reading == 0 else 0
            sleep(0.05)
```

5 - Armazenamento em Nuvem

O armazenamento em nuvem desempenha um papel essencial no monitoramento remoto das colmeias, pois permite o acesso em tempo real aos dados coletados pelos sensores. Para este projeto, optamos por manter o uso do banco de dados adotado pelo grupo do primeiro semestre de 2024, dada sua eficácia comprovada no contexto do sistema. Assim, utilizamos o *Firebase Realtime Database* [6], uma solução fornecida pelo Google que possibilita a sincronização instantânea das informações entre os dispositivos conectados e o

banco de dados. Essa escolha garante que todas as leituras sejam imediatamente acessíveis após o registro, aumentando a eficiência no monitoramento e análise dos dados coletados.

No Firebase, os dados seguiam uma estrutura hierárquica em formato de árvore JSON, onde cada conjunto de informações era armazenado em um nó identificado por uma timestamp. Esse modelo facilita tanto a organização quanto a consulta dos registros históricos. Entretanto, nosso grupo implementou alterações significativas em relação à estrutura do banco de dados original do projeto do primeiro semestre de 2024. Além de incorporar as leituras de poluição, ajustamos o sistema para possibilitar o armazenamento de registros provenientes de diferentes placas.

Essa modificação trouxe maior flexibilidade ao banco de dados, permitindo que o projeto se expanda para novos cenários de uso. Agora, é possível selecionar os dados que se deseja observar diretamente na aplicação web, utilizando o nome da placa como critério de escolha. Mas para que isso fosse possível, foi necessária a adição de uma nova camada de profundidade na estrutura de dados, organizando os conjuntos de informações de acordo com o nome atribuído a cada placa:

```JavaScript
{"bee_data": {...},
"bee_data01": {...},
"bee_data2": {...},
"bee_data_1": {...},
"bee_data_3": {...},
"bee_data_4": {...},
}
```

Os dados dentro desses conjuntos são então indexados pela data e horário do momento da coleta, sendo incluídos tanto os dados coletados pela placa quanto os provenientes do OpenWeatherMap.

```JavaScript
{"bee_data": {
      "25_11_2024-20_22_58": {
            "openweathermap": {...}
```

```
            "sensor": {...}
            }
      "25_11_2024-20_38_04": {
            "openweathermap": {...}
            "sensor": {...}
            }
      ...
},}
```

Após as atualizações sobre as estruturas dos dados sobre a poluição, como explicado anteriormente, é necessário adaptar a formatação deles, de modo a adequar os novos dados de poluição como segue abaixo. Vale ressaltar que a estrutura em si é muito maior pois ela agrega todos os dados que OpenWeatherMap envia, porém os outros dados não possuem relevância para as análises então não serão indicados, mas seguem sendo adicionados ao banco de dados para que futuros grupos possam fazer proveito dos mesmos.

JavaScript

```
"openweathermap":{
      ...
      "main": {
          "feels_like": 298.28,
          "grnd_level": 941,
          "humidity": 65,
          "pressure": 1015,
          "sea_level": 1015,
          "temp": 298.04,
          "temp_max": 298.04,
          "temp_min": 297.31
        },
      "pollution": {
            "list": [
            {
              "components": {
                "co": 594.14,
                "nh3": 7.92,
                "no": 0,
                "no2": 28.45,
                "o3": 104.43,
                "pm10": 32.13,
```

```
                "pm2_5": 20.41,
                "so2": 12.99
              },
              "dt": 1732577914,
              "main": {
                "aqi": 3
              }
            }
          ]
      },
      ...
}
```

Os dados coletados pelos sensores não sofreram alterações e continuam indicando os dados adquiridos pelos sensores das placas.

```JavaScript
"sensor":{
      "id_placa": 1,
        "pressao_interna": 893.95,
        "proximidade": 94,
        "som": 2,
        "temperatura_interna": 292.3144,
        "umidade_interna": 64.35

}
```

Por fim, é importante destacar que, caso seja necessário visualizar o JSON completo, incluindo os dados que ainda não estão sendo utilizados, é possível acessá-lo diretamente em nosso Firebase através do link [13].

6 - Obtenção de Dados

A estrutura de dados foi gradualmente modificada para se adaptar à coleta de dados utilizando a placa BitDogLab [7]. Inicialmente, os dados foram coletados seguindo a mesma estrutura empregada no projeto anterior, utilizando apenas informações provenientes do OpenWeatherMap e dos sensores, sem alterações na arquitetura. Posteriormente, foi realizada uma adaptação na estrutura para incorporar os dados de poluição, o que exigiu uma reconfiguração completa tanto da arquitetura de coleta quanto dos dados armazenados, conforme descrito na seção anterior. Por fim, uma última coleta foi realizada, abrangendo todos os dados, porém com uma modificação no intervalo de coleta.

Essa evolução permitiu dividir o processo de coleta em três etapas distintas:

1. **Primeira Medição:** Nessa etapa, todos os dados disponíveis (exceto os de poluição) foram coletados em intervalos de 10 segundos. A figura 5 apresenta parte dos dados resultantes da primeira medição, realizada com o sensor de proximidade em funcionamento.
2. **Segunda Medição**: Realizada sem o sensor de proximidade, coletando dados de poluição e outros disponíveis, mas sem os dados relacionados aos sensores devido à ausência do sensor de proximidade, ainda com um intervalo de 10 segundos. A figura 6 exibe alguns dos dados coletados na segunda medição, realizada sem o sensor de proximidade.
3. **Terceira Medição**: Realizada também sem o sensor de proximidade, mas coletando todos os dados em intervalos de 15 minutos. A figura 7 exemplifica parte dos dados coletados.

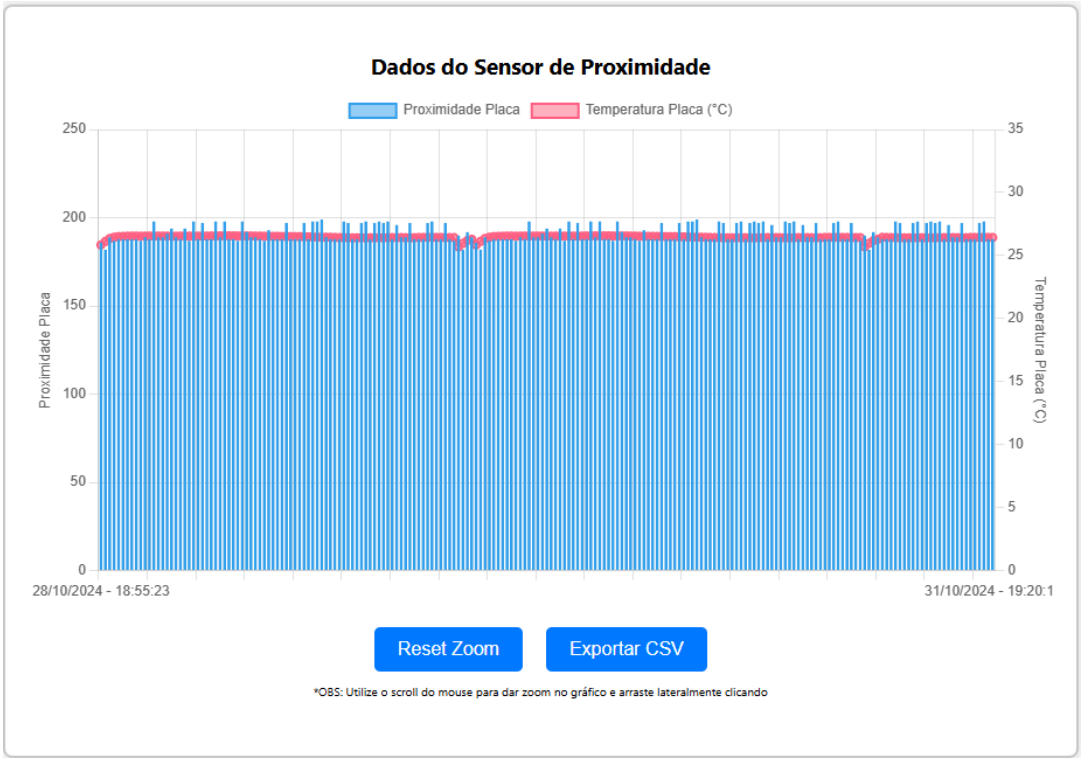

Figura 5: Gráfico sobre os dados de proximidade e temperatura coletados da primeira medição

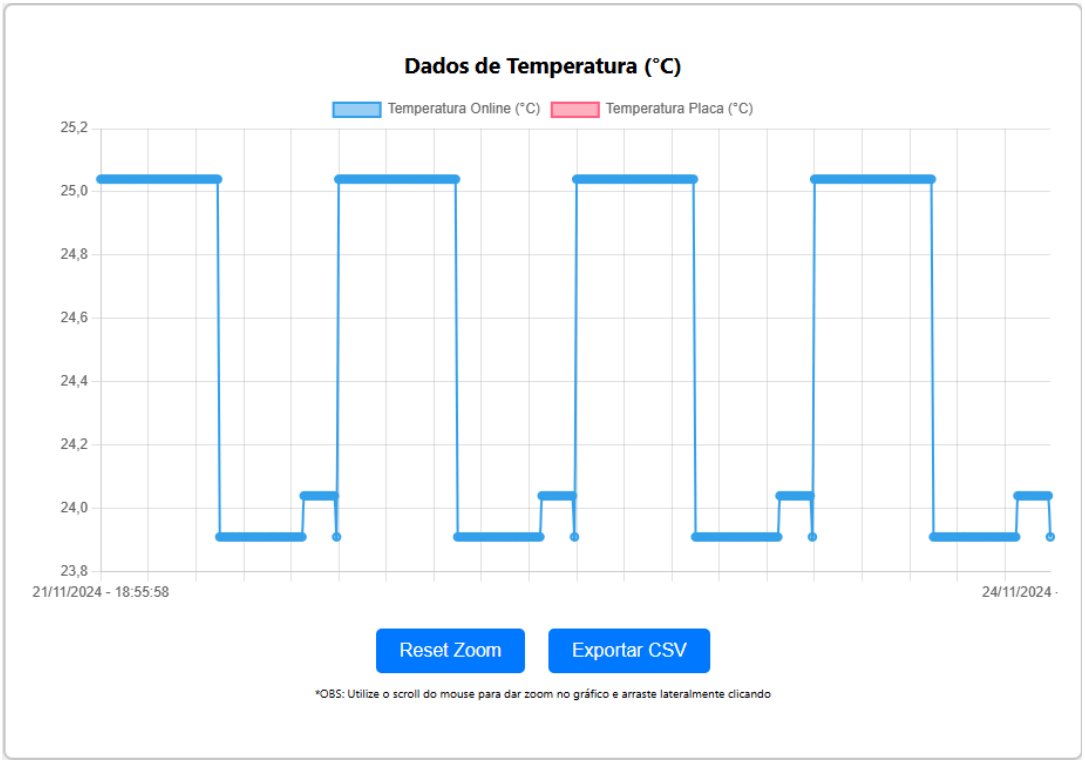

Figura 6: Gráfico sobre os dados de temperatura coletados da segunda medição

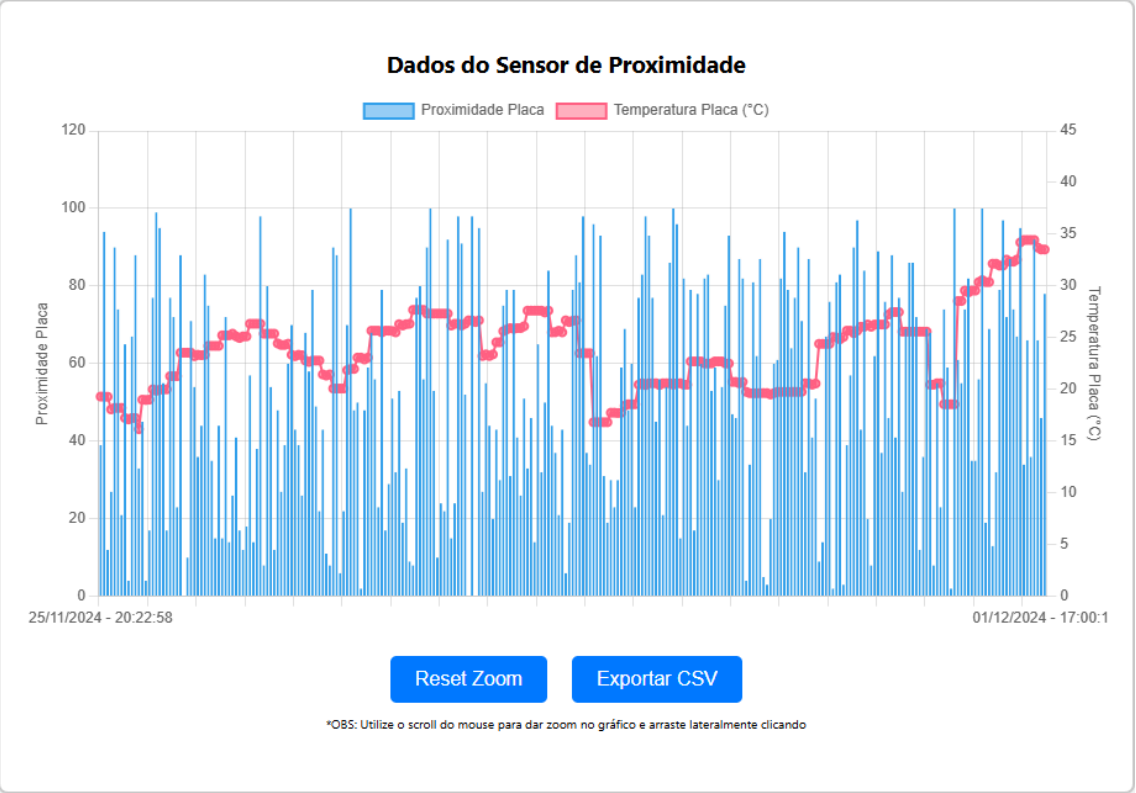

Figura 7: Gráfico sobre os dados de proximidade e temperatura coletados da terceira medição

No parágrafo anterior, foi mencionado um momento de coleta sem o sensor de proximidade. É importante destacar que a medição com a placa depende do funcionamento de todos os sensores para operar corretamente. Essa necessidade, aliada ao fato de que apenas uma das quatro placas disponibilizadas possuía o sensor de proximidade [17], limitou a capacidade de realizar coletas completas. Como havia mais de um grupo atuando no mesmo projeto, mas em frentes diferentes, as placas precisaram ser compartilhadas, sendo duas destinadas a cada grupo. Essa situação exigiu a troca constante de placas entre os grupos, dificultando a coleta contínua e tornando necessária a criação de dados artificiais para viabilizar a análise dos dados de poluição.

Além dessa limitação, as placas apresentaram problemas técnicos durante as medições. Um dos problemas observados foi a interrupção aleatória da coleta de dados, mesmo quando devidamente alimentadas, o que gerava lacunas no registro. Outro problema recorrente estava relacionado à dessincronização de data e horário. Quando conectadas diretamente à tomada, as placas registravam todos os dados como coletados em 01/01/2021, situação que também ocorria após qualquer interrupção no fornecimento de energia e reinício do processo de coleta.

Por fim, foi realizado um cálculo para estimar o espaço necessário no banco Firebase para armazenar os dados de um ano de coleta. Utilizando como referência os dados coletados em 26/11/2024, com medições realizadas a cada 15 minutos, foi registrado o primeiro dado às 08:09:43 e o último às 22:45:42, totalizando 59 registros e consumindo 136,6 kB de memória. Isso equivale a uma média de 2,315 kB por dado e cerca de 9,261 kB por hora. Considerando uma coleta diária de 12 horas, o consumo estimado é de 111,137 kB por dia, resultando em um total de 39,61 MB por ano ou 39,72 MB em anos bissextos.

7 - Aplicação Web

O projeto de monitoramento de colmeias passou por diversas fases ao longo de três anos, com contribuições distintas de diferentes grupos para seu aprimoramento. Em 2022, o primeiro grupo desenvolveu um aplicativo Android que utilizava exclusivamente comunicação via Bluetooth, permitindo interação apenas com uma única placa. No ano seguinte, em 2023, um novo grupo criou um aplicativo em Flutter [4], ainda focado no sistema Android. Essa versão adicionou funcionalidades como requisições HTTP e cadastro via Bluetooth, além de reaproveitar o recurso de visualização de gráficos desenvolvido na primeira versão. Finalmente, em 2024, um terceiro grupo substituiu o aplicativo anterior por um WebApp em React, que trouxe uma integração direta com o banco de dados Realtime Database do Firebase. Com isso, eliminou-se a necessidade de configuração via Bluetooth, proporcionando maior eficiência no acesso e gerenciamento dos dados coletados pelas placas.

Ao assumir o projeto, nosso grupo identificou diversas limitações na versão anterior do WebApp, tanto no design quanto nas funcionalidades. Por esse motivo, realizamos uma reformulação completa, mantendo o React como base tecnológica e implementando melhorias significativas para otimizar a interação com os dados.

7.1 Melhorias Implementadas

As melhorias implementadas pelo grupo durante o desenvolvimento do WebApp foram substanciais e visam resolver problemas identificados nas versões anteriores, além de adicionar novas funcionalidades que ampliaram o escopo e a usabilidade do sistema, todas as alterações estão presentes no github [14].

O WebApp passou por um redesenho completo, com foco na melhoria da experiência do usuário e na usabilidade do sistema. Entre as principais mudanças, destaca-se a adoção de

uma nova paleta de cores que reflete melhor o tema do projeto, utilizando tons suaves e profissionais para oferecer uma experiência visual mais agradável e harmônica. Os componentes foram reposicionados estrategicamente, seguindo uma hierarquia visual clara que prioriza a navegabilidade e a facilidade de uso. A Figura 9 destaca essa atualização visual, evidenciando a reestruturação completa da interface.

Além disso, anteriormente, existia apenas um único gráfico que agrupava toda a informação coletada, tornando a visualização confusa e pouco organizada, dificultando a análise específica de cada dado coletado. Sendo assim, foi realizada a separação dos gráficos por tipo de dado, permitindo uma visualização clara e direcionada. Agora, os usuários podem escolher gráficos específicos, como temperatura, umidade, pressão ou proximidade, e explorar os dados de maneira mais detalhada e intuitiva. A Figura 8 apresenta um outro ponto de melhoria relevante sobre os gráficos criados na aplicação web, a adição de tooltips que ajudam na identificação de cada ponto de dados, facilitando a interpretação dos mesmos.

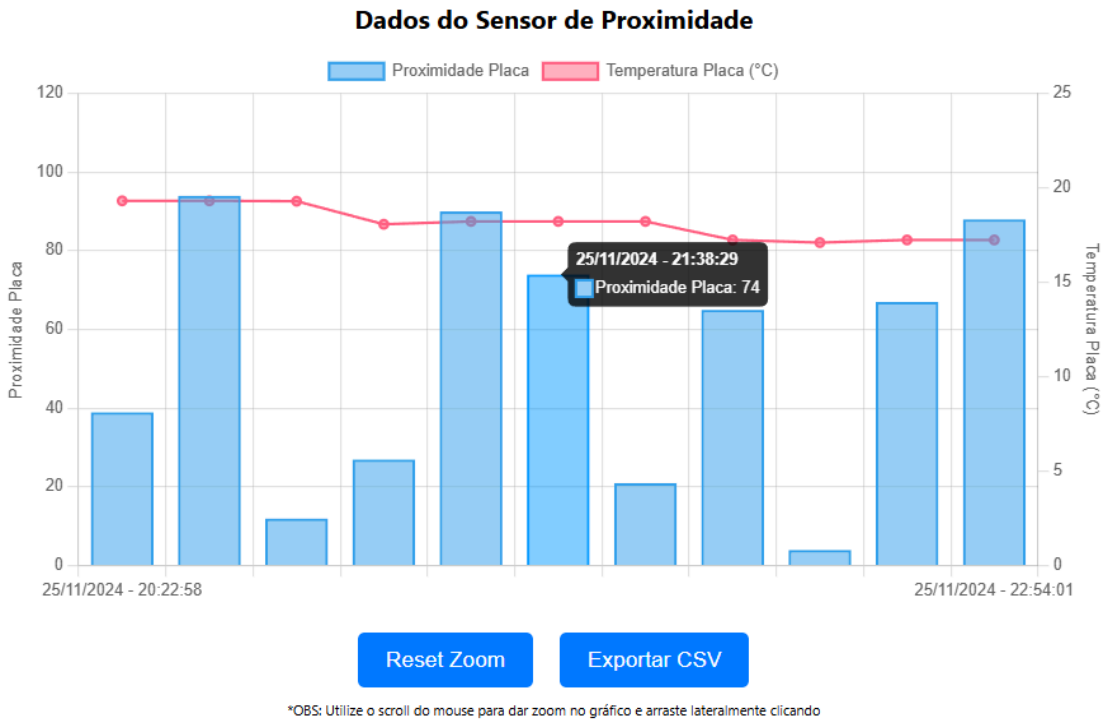

Figura 8: Função de tooltip sobre um dos dados do gráfico de proximidade

Uma das principais melhorias implementadas foi a adição da funcionalidade de exportação de dados para o formato CSV. Essa funcionalidade é particularmente útil para pesquisadores e administradores que necessitam realizar análises mais aprofundadas

utilizando ferramentas externas, como Excel ou Python. O objetivo principal deste WebApp é fornecer uma visão geral dos dados coletados pelas placas, de forma simplificada e em formato de dashboard. No entanto, a exportação para CSV permite que os usuários façam uma análise mais detalhada e personalizada, utilizando softwares especializados para criar gráficos mais complexos, cruzar dados e realizar outras formas de análise avançada. Essa melhoria amplia a flexibilidade e o valor dos dados, permitindo que sejam utilizados em uma variedade de contextos e metodologias de pesquisa.

Além das funcionalidades previamente existentes, a nova versão do WebApp incorporou a análise de dados relacionados à poluição atmosférica. Agora, é possível visualizar gráficos detalhados sobre a qualidade do ar (AQI) e componentes específicos, como CO, NO, NO_2, entre outros, através da API do OpenWeatherMap, que já era utilizada pelos grupos anteriores. Essa melhoria ampliou significativamente o alcance do projeto, permitindo a análise do comportamento das abelhas em relação a fatores ambientais adicionais, além dos dados que já estavam sendo coletados. Isso possibilita a correlação entre a saúde das colmeias e a qualidade do ar, o que pode contribuir para uma compreensão mais profunda dos impactos ambientais no comportamento das abelhas.

Diferentemente da versão anterior, que permitia exibir dados de apenas uma placa específica, o novo design foi pensado levando em consideração a possibilidade de o sistema evoluir para uma versão distribuída no futuro. Nesse cenário, seria fundamental permitir a visualização de dados de diferentes placas. Para atender a essa necessidade, além de alterar o formato de armazenamento dos dados no banco de dados Firebase, implementamos um menu de seleção que possibilita ao usuário escolher a placa cujos dados deseja visualizar. Essa funcionalidade torna a análise de informações de múltiplas placas mais intuitiva e acessível. Com essas mudanças, o sistema se torna mais escalável, preparado para lidar com um volume crescente de dados à medida que o projeto se expande.

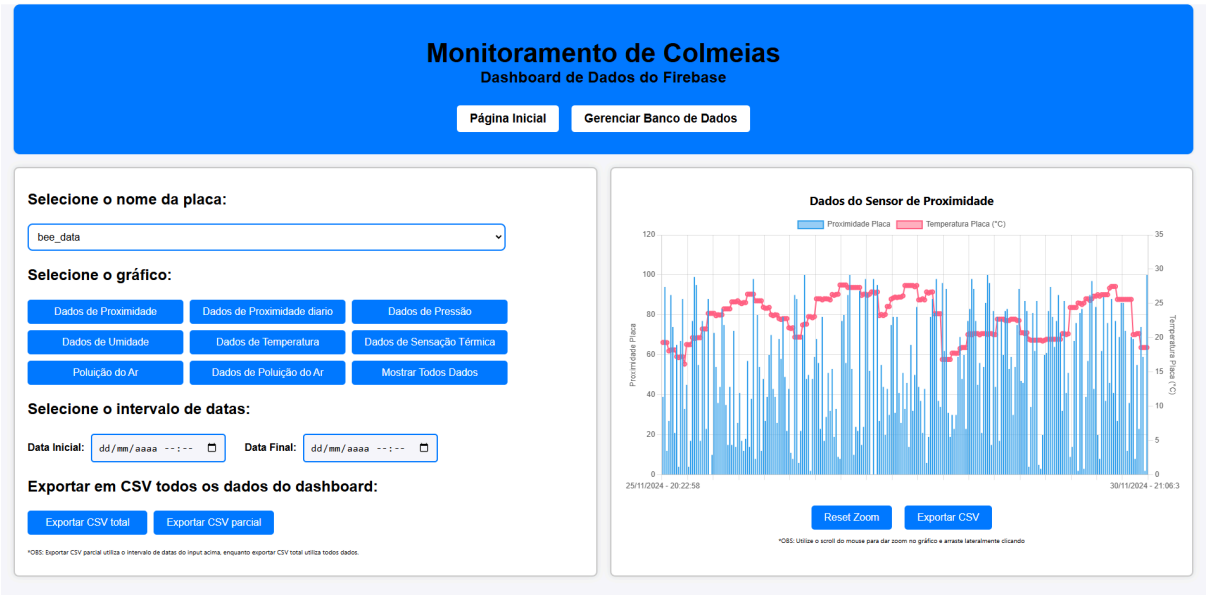

Figura 9: Nova organização da interface dos elementos da aplicação web

Com o objetivo de facilitar a manutenção do banco de dados, foi desenvolvida uma página específica para a exclusão de dados por placa, cuja interface pode ser observada na figura 10. Embora ainda esteja em uma fase inicial, essa funcionalidade desempenha um papel essencial na remoção de informações irrelevantes ou incorretas. Essa ferramenta tem o potencial de se tornar um recurso valioso para a otimização do armazenamento e para o aprimoramento da qualidade dos dados disponíveis para análise.

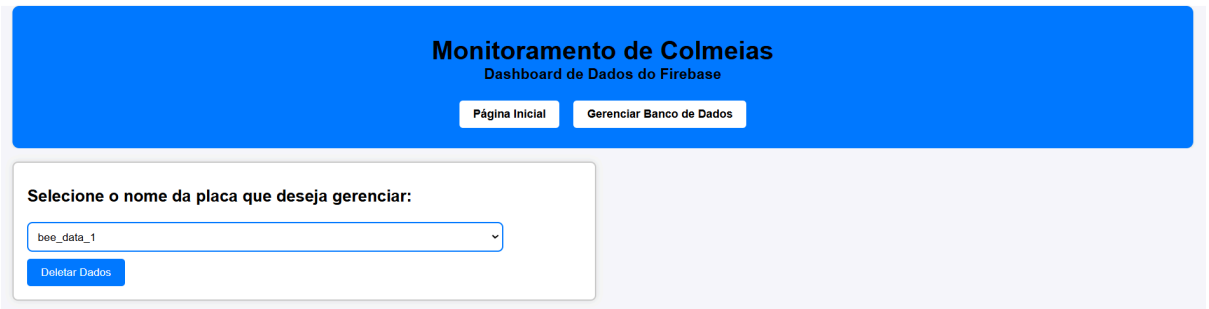

Figura 10: Página de Gerenciamento do Banco de Dados da aplicação web

Outro ponto que merece destaque é que a manipulação de dados é realizada diretamente no front-end do projeto em alguns casos, como na geração do gráfico de proximidade diária. Nesse processo, o front-end é responsável por somar os dados de aproximação armazenados no banco de dados, agrupando-os por dia para exibição no gráfico.

8. Resultados

Nesta fase do projeto, os principais resultados obtidos refletem as melhorias implementadas na aplicação web e no sistema de monitoramento como um todo, com foco em aprimorar a experiência do usuário, a precisão dos dados coletados e a robustez da infraestrutura tecnológica.

A reformulação da aplicação web trouxe mudanças fundamentais, como a separação dos gráficos por tipo de dado e a inclusão de ferramentas interativas que facilitam uma análise detalhada das informações coletadas. Essas alterações não apenas atenderam às observações feitas pelo Prof. Roberto Grecco, mas também enriqueceram a experiência de uso, tornando o sistema mais intuitivo e visualmente organizado. Os novos gráficos permitem a análise individual de variáveis, como temperatura, umidade e qualidade do ar, promovendo uma compreensão mais clara e profunda dos dados. Outra adição importante foi o gráfico que combina as informações do sensor de proximidade com os dados de temperatura. Essa funcionalidade possibilita a análise de correlação entre essas variáveis e estabelece uma base para a criação de gráficos similares que explorem a relação entre proximidade e outros dados coletados. Mesmo que de forma inicial, essa abordagem já oferece um potencial significativo para análises futuras e uma visão mais integrada do comportamento dos dados.

Ainda na aplicação, a adição de funcionalidades de gestão de dados, como o campo de seleção de placa e o botão de exclusão de dados do banco, permitiu um maior controle para o usuário sobre os dados coletados, além de garantir maior flexibilidade na visualização, possibilitando sumarizar em um só lugar os registros obtidos por diferentes placas.

Além disso, a integração dos dados de poluição atmosférica, por meio da API do OpenWeatherMap, ampliou o escopo do projeto, permitindo que se explorem correlações entre fatores ambientais e o comportamento das abelhas. Este novo parâmetro de análise é particularmente relevante para entender o impacto das condições climáticas e de qualidade do ar na saúde das colmeias, contribuindo para pesquisas mais abrangentes sobre polinização e preservação das abelhas.

A funcionalidade de exportação de dados para CSV é outro avanço relevante, permitindo aos pesquisadores utilizarem os dados em ferramentas externas para análises mais avançadas e personalizadas. Isso aumenta o valor do sistema como um instrumento de pesquisa e ensino, oferecendo uma base rica para investigações futuras.

De maneira geral, a aplicação desenvolvida possibilitou uma evolução significativa na frente de manipulação e visualização dos dados, além de garantir maior robustez, eficiência e confiabilidade.

9. Conclusão e trabalhos futuros

O projeto de monitoramento de colmeias com o uso de Internet das Coisas (IoT) evoluiu significativamente ao longo das últimas etapas, com avanços importantes na infraestrutura de hardware e software, tornando o sistema mais eficiente e capaz de fornecer uma visão mais detalhada dos dados ambientais. As melhorias na aplicação web e a adoção de novas tecnologias permitiram expandir o escopo do projeto, contribuindo tanto para a preservação das abelhas quanto para a formação de futuros pesquisadores na área de tecnologia aplicada ao monitoramento ambiental.

Entretanto, alguns objetivos levantados pelo Prof. Roberto Grecco não puderam ser totalmente implementados nesta fase do projeto. O desenvolvimento de um sistema de controle de sessão para o website, que permitiria a cada usuário cadastrar suas próprias placas e observar dados de outros, permanece como um desafio a ser explorado. Além disso, a implementação de uma gerência de usuários através de um super-Usuário, com capacidades administrativas, e a possibilidade de alterar a latitude e longitude das placas diretamente pela interface do website também foram aspectos mencionados que não foram possíveis de realizar devido ao tempo limitado e à complexidade das tarefas.

Para os próximos passos, sugerimos os seguintes trabalhos futuros:

1. **Controle de Sessão e Cadastro de Usuários**: Implementar um sistema de autenticação que permita aos usuários registrar suas próprias placas, mas também compartilhar dados de forma controlada. Essa funcionalidade garantiria uma personalização maior e aumentaria a segurança do sistema, possibilitando uma interação mais rica e colaborativa entre os participantes do projeto.
2. **Gerenciamento de super-Usuário**: Desenvolver uma camada de gerenciamento de usuários que inclua a figura de um super-Usuário. Esse super-Usuário teria privilégios para gerenciar cadastros, visualizar todas as placas, e realizar alterações, garantindo uma supervisão mais detalhada e controle sobre o sistema.

3. **Edição de Localização das Placas**: Facilitar a configuração da latitude e longitude diretamente pelo website tornaria o sistema mais prático para os usuários, especialmente em situações onde as colmeias são realocadas. Essa funcionalidade permitiria uma atualização rápida dos parâmetros de localização, mantendo a precisão dos dados.
4. **Implementação de Alertas Inteligentes**: Utilizar técnicas de aprendizado de máquina para identificar padrões críticos nos dados e implementar alertas automáticos para situações que exijam intervenção, como mudanças bruscas na temperatura ou altos níveis de poluição que possam impactar as colmeias.

Esses trabalhos têm o potencial de ampliar significativamente a capacidade e a aplicabilidade do sistema, tornando-o uma ferramenta ainda mais poderosa para monitoramento, pesquisa e educação. Com as melhorias sugeridas, esperamos que o projeto continue evoluindo e contribuindo para a preservação das abelhas e o desenvolvimento de tecnologias inovadoras.

10. Referências

[1] GONÇALVES, J. C.; DOS S. P. MONROE V. A. SCHOLZE F. FRUETT L. F. BITTENCOURT, D. M. D. S. L. J. S. Monitoramento de colmeias com Internet das Coisas. Disponível em: Monitoramento de colmeias com Internet das Coisas .

[2] Firebase. Disponível em: Firebase.

[3] Firebase cli. Disponível em: Firebase CLI reference.

[4] Flutter. Disponível em: Flutter.

[5] React. Disponível: React

[6] Realtime database. Disponível em: Firebase Realtime Database.

[7] Repositório bitdoglab. Disponível em: BitDogLab

[8] Bosch, Datasheet sensor bme680. Disponível em: https://cdn-shop.adafruit.com/product-files/3660/BME680.pdf.

[9] R. T. Fielding, M. Nottingham, and J. Reschke, HTTP Semantics. RFC 9110, June 2022.

[10] F. Fruett and R. Greco, Multiple sensor beehive monitoring. Disponível em: https://docs.google.com/document/d/1Mz-MCR6UlPnqEdyOTakaD4TGH3IIKm6izRvWRS6cQM8/edit, Fevereiro 2023.

[11] J. C. Gonçalves, D. M. D. Santos, L. J. S. D. S. P. Monroe, and V. A. Scholze, apicultores app. Disponível em: https://github.com/VictorScholze/ apicultoresWeb.

[12] bee monitoring micropython. Disponível em: https://github.com/JonasCardoso/bee_monitoring_BitDogLab.

[13] Site para visualização dos dados. Disponível em: https://bees-f6e6b.web.app/.

[14] Códigos realizados para funcionamento do webapp relacionado ao firebase do projeto: https://github.com/jose219081/MC030-bees.

[15] Códigos realizados para a placa para realizar a coleta dos dados: https://github.com/jose219081/MC030-codes-board.

[16] R. Hammelrath, Micropython driver for a bme680 breakout. Disponível em: https: //github.com/robert-hh/BME680-Micropython.

[17] U. Info, Datasheet sensor e18-d80nk. Disponível em: https://www.usinainfo.com.br/sensor-de-proximidade/ sensor-de-proximidade-e18-d80nk-infravermelho-npn-deteccao-3-a-80cm-2791. html

[18] G. J. S. Moraes, V. A. M. Dantas, A. C. L. C. C. F. Renoldi, H. F. Zimerman, P. P. Alves, J. V. F. Costa, L. S. L. Carmo, and L. F. Bittencourt, Monitoramento de colmeias com internet das coisas, (2022).

[19] M. C. Rosa, L. C. Castello, C. A. A. Trujillo, and L. F. Bittencourt, Monitoramento de colmeias com internet das coisas, (2023).

[20] OpenWeather, Weather api. Disponível em: https://openweathermap.org/api.

[21] OpenWeather Pollution, Weather api. Disponível em: https://openweathermap.org/api/air-pollution.

[22] Sistema da placa BitDogLab : https://embarcados.com.br/bitdoglab-uma-jornada-educativa-com-eletronica-embarcados-e-ia/