APLICATIVO PARA MEDIÇÃO DE SAÚDE DE PLANTAS VIA MQTT

Carlos Miguel Moliterno de Oliveira, Mario Rodrigues Braga Matos, Cauã Macedo Lemes, Bruno Michel Pera.

Colégio Técnico "Antônio Teixeira Fernandes", Rua Paraibuna, 75, Jardim São Dimas - 12245-020 - São José dos Campos-SP, Brasil, carlinhosmoliterno@gmail.com, mariorodriguesbragamatos@gmail.com, cauamace82@gmail.com, brunomichel00@gmail.com.

Resumo

A saúde vegetal está relacionada ao acompanhamento de condições ambientais que podem influenciar o desenvolvimento das plantas e à identificação de situações de estresse. Este trabalho apresenta o PlantsIOT, um protótipo de baixo custo para monitoramento vegetal baseado em Internet das Coisas (IoT). O sistema integra ESP32-CAM, sensores, protocolo MQTT, broker HiveMQ, Firebase Realtime Database e aplicativo Android desenvolvido em Kotlin. A aplicação permite o cadastro e configuração de plantas, acompanhamento de dados, consulta ao histórico de leituras e visualização gráfica das informações. Nos testes funcionais realizados com dados simulados, a comunicação MQTT possibilitou a atualização das informações em tempo real, enquanto o banco de dados permitiu a persistência e recuperação dos registros. Os resultados demonstraram a viabilidade da integração das tecnologias empregadas para o monitoramento vegetal, com potencial de aplicação educacional e em pequenos cultivos. A detecção automática de pragas por processamento de imagens permaneceu como limitação e possibilidade para trabalhos futuros.

Palavras-chave: Internet das Coisas. Monitoramento vegetal. MQTT. Aplicativo móvel. Agricultura de precisão.

Área do Conhecimento: Ciências Exatas e da Terra.

Introdução

As plantas sustentam processos ecológicos essenciais, a produção de alimentos e diferentes cadeias econômicas. Ainda assim, muitas pessoas apresentam dificuldade para percebê-las como organismos vivos relevantes, fenômeno discutido como cegueira botânica. Esse distanciamento pode reduzir o interesse por sua conservação e pelo acompanhamento dos fatores que condicionam seu desenvolvimento (Silva; Brito; Purificação, 2024).

Na agricultura, a observação sistemática também é indispensável. A fitopatologia estuda as doenças vegetais, seus agentes causais, as condições favoráveis à infecção e os métodos de prevenção e controle (Amorim et al., 2016). Em um país tropical e de grande extensão territorial, o monitoramento de cultivos é particularmente importante, pois diferentes fitopatógenos e condições ambientais podem comprometer a produtividade (Gasparotto et al., 2023).

Aplicativos como Plantix, Plantum e Agrio indicam o interesse por ferramentas digitais destinadas ao cuidado das plantas. Entretanto, modelos de assinatura e funções pagas podem limitar o acesso de estudantes, entusiastas e pequenos produtores. Paralelamente, plataformas de Internet das Coisas (IoT) permitem aproximar sensores, redes e aplicativos, transformando medições ambientais em informações que apoiam decisões de manejo.

Nesse contexto, o objetivo deste trabalho foi desenvolver o PlantsIOT, uma solução composta por hardware e aplicativo móvel para monitorar variáveis relacionadas à saúde vegetal. Como objetivos específicos, buscou-se estruturar a comunicação MQTT entre o dispositivo e o aplicativo, persistir leituras no Firebase, cadastrar plantas e seus parâmetros, apresentar um painel sintético de saúde e disponibilizar histórico e gráficos de evolução.

Metodologia

O estudo possui natureza aplicada e caráter de desenvolvimento tecnológico. O protótipo foi organizado em quatro camadas: aquisição de dados, comunicação, persistência e apresentação. Na aquisição, adotou-se o ESP32-CAM, microcontrolador com conectividade Wi-Fi e capacidade

adequada a aplicações embarcadas de IoT (Espressif Systems, 2023). A arquitetura prevê sensores para temperatura, umidade do ar, umidade do solo, luminosidade e pH, além de parâmetros de altura e coloração vegetal.

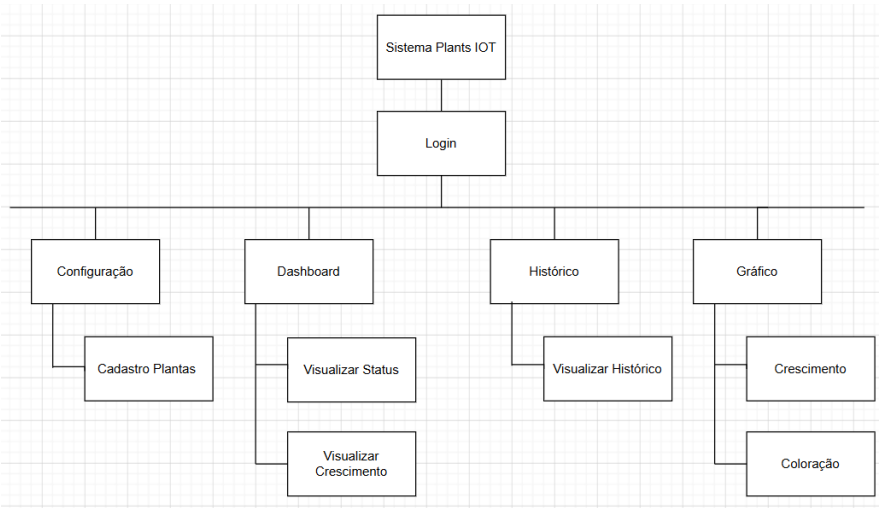
A comunicação foi planejada com o MQTT, protocolo leve baseado no modelo de publicação e assinatura. O HiveMQ foi empregado como broker para intermediar as mensagens. Cada planta recebe um tópico configurável; o dispositivo publica as leituras nesse canal e o aplicativo assina o mesmo endereço, permitindo separar o tráfego de diferentes cultivos. Essa arquitetura é apropriada a cenários de baixa largura de banda e reduz o acoplamento entre o hardware e a interface móvel (Kato, 2024).

O aplicativo foi desenvolvido no Android Studio, com lógica em Kotlin e interfaces em XML. Foram criados módulos de cadastro, autenticação, menu, dashboard, histórico, gráficos e configurações. A linguagem Kotlin favoreceu a organização do código e sua integração com o ecossistema Android (Resende, 2024), enquanto a separação entre XML e lógica de programação facilitou a manutenção das telas (Silva, 2023).

Para persistência, utilizou-se o Firebase Realtime Database, banco NoSQL em nuvem que representa os registros em uma árvore JSON e sincroniza alterações com os clientes conectados (Google, 2025). A raiz foi dividida nos nós users e plants. Cada planta armazena identificação, espécie, tópico MQTT e um subnó readings; as leituras incluem data e hora, altura, índices verde e amarelo, pontuação de saúde, status e anotações de manejo.

O desenvolvimento da interface móvel fundamenta-se na estruturação prévia de sua arquitetura de informação, representada por meio do diagrama hierárquico. A intenção primária deste modelo consiste em mapear o fluxo de navegação do usuário e estabelecer a hierarquia lógica das funcionalidades do aplicativo, garantindo que a experiência de uso seja intuitiva e organizada. Como mostrado na Figura 1.

Figura 1 – Arquitetura de navegação do aplicativo.



Fonte: elaborado pelo próprio autor.

A visualização estatística foi implementada com a biblioteca MPAndroidChart. Um gráfico de linhas representa a evolução da altura ao longo das leituras, enquanto um gráfico de setores compara os índices de tonalidade verde e amarela. O dashboard converte os valores mais recentes em uma pontuação de zero a cem e em uma classe qualitativa, facilitando a interpretação por usuários sem formação em fitopatologia.

A avaliação concentrou-se em testes funcionais das rotas de autenticação, cadastro e configuração; publicação e recepção de mensagens; gravação e recuperação de registros; atualização do dashboard; ordenação cronológica do histórico; e geração dos gráficos. Foram simulados cenários de cultivo classificados como crítico, atenção, recuperando e excelente. Como critérios de aceitação, consideraram-se a entrega da mensagem ao aplicativo, a persistência correta, a atualização da interface e a coerência das informações exibidas.

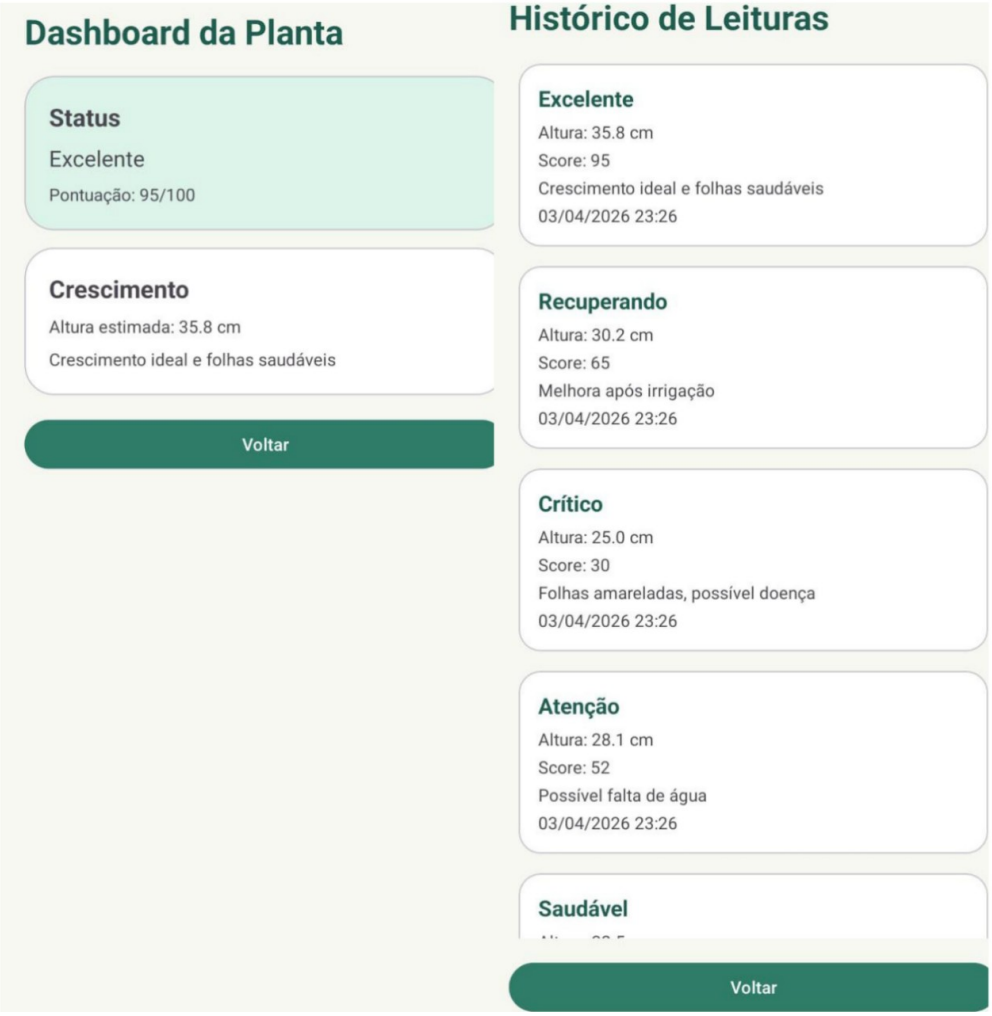
Resultados

O protótipo resultou em um aplicativo com fluxo completo de acesso e monitoramento. Após o cadastro e a autenticação, o usuário pode registrar a planta, informar seu nome popular e científico e definir o tópico MQTT. O menu centraliza o acesso ao dashboard, ao histórico, aos gráficos e às configurações, reduzindo a quantidade de passos necessários para consultar os dados.

Nos testes com dados simulados, o fluxo MQTT transmitiu as leituras ao cliente móvel e os cartões de informação foram atualizados sem recarregamento manual. A estrutura do Firebase armazenou cada registro sob um identificador único, preservando data, hora, métricas e observações. Ao recuperar os dados, o aplicativo apresentou o histórico em ordem cronológica inversa, com a leitura mais recente no topo.

O dashboard sintetizou a situação atual por meio de pontuação, classificação e altura. Em um dos cenários, a planta recebeu 95/100 e o estado “Excelente”. Os registros também representaram transições entre “Crítico”, associado a folhas amareladas e suspeita de problema; “Atenção”, relacionado à possível falta de água; “Recuperando”, após irrigação; e “Excelente”. Essa organização tornou visíveis tanto o estado instantâneo quanto a resposta do cultivo às intervenções registradas.

Figura 2 – Dashboard e histórico de leituras do PlantsIOT.



Fonte: elaborado pelo próprio autor.

O módulo de gráficos exibiu a curva de crescimento e a proporção dos índices de coloração foliar. A combinação entre painel, histórico e gráficos permitiu consultar três níveis de informação: a condição

atual, a sequência de eventos e a tendência acumulada. Também foi possível alterar planta e tópico sem modificar o código do aplicativo, aspecto importante para a reutilização do protótipo.

Quanto às limitações, as variáveis foram avaliadas predominantemente com cenários simulados e parâmetros previamente configurados. Restrições de tempo e orçamento impediram a implementação integral do processamento de imagens no ESP32-CAM para identificação automática de pragas. Assim, os índices greenIndex e yellowIndex foram inseridos manualmente ou simulados no banco. A calibração prolongada dos sensores de pH e umidade e a estabilidade das conexões físicas em campo permanecem etapas necessárias antes de validar o sistema em escala produtiva.

Discussão

Os resultados indicam que a integração de componentes acessíveis pode produzir uma cadeia funcional de monitoramento, desde a aquisição até a apresentação. O MQTT mostrou-se adequado ao envio de pequenos pacotes de telemetria, em concordância com sua aplicação em sistemas com ESP32 (Kato, 2024). A separação por tópicos também favorece a expansão do sistema para mais de uma planta, desde que sejam acrescentadas políticas de autenticação, autorização e qualidade de serviço.

A persistência em árvore JSON simplificou o armazenamento e a sincronização com o Android, características descritas na documentação do Firebase (Google, 2025). A transformação de valores brutos em classes qualitativas favoreceu a leitura rápida, mas não deve ser confundida com diagnóstico fitopatológico. Como as doenças vegetais envolvem a interação entre hospedeiro, agente causal e ambiente (Amorim et al., 2016), as pontuações devem funcionar como alerta para observação e manejo. A gratuidade e a integração com hardware acessível também favorecem atividades escolares e pequenos cultivos, aproximando os usuários das respostas das plantas e contribuindo para enfrentar a cegueira botânica (Silva; Brito; Purificação, 2024).

Por outro lado, ainda não é possível generalizar o desempenho agronômico do sistema. A ausência de ensaios prolongados com plantas reais, de comparação com instrumentos calibrados e de métricas quantitativas de latência e disponibilidade limita as conclusões. O principal resultado desta etapa é a prova de conceito da arquitetura integrada, que deverá ser submetida a validação experimental controlada.

Conclusão

O PlantsIOT atingiu o objetivo de integrar aplicativo Android, mensageria MQTT, banco de dados em tempo real e arquitetura de hardware baseada em ESP32-CAM em um protótipo de monitoramento vegetal. Foram implementados cadastro, autenticação, configuração de planta e tópico, dashboard, histórico e gráficos, formando um fluxo coerente para receber, armazenar e interpretar leituras.

Os testes funcionais com dados simulados demonstraram a viabilidade da comunicação, da persistência e da apresentação da evolução temporal em uma interface acessível. A solução possui potencial educacional e pode apoiar pequenos produtores, desde que seus indicadores sejam tratados como suporte à observação. Recomenda-se, em continuidade, calibrar os sensores, realizar ensaios com plantas reais, mensurar o desempenho da rede, reforçar a segurança e implementar visão computacional e atuação controlada por bombas e válvulas.

Referências

AMORIM, L. *et al*. **Manual de fitopatologia**. São Paulo: Agronômica Ceres, 2016.

ESPRESSIF SYSTEMS. **ESP32 Series Datasheet. Shanghai**: Espressif Systems, 2023. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 5 ago. 2026.

GASPAROTTO, L. *et al*. **Glossário de fitopatologia**. 4. ed. Brasília, DF: Embrapa, 2023.

GOOGLE. **Firebase Realtime Database**: documentação. 2025. Disponível em: https://firebase.google.com/docs/database. Acesso em: 5 ago. 2026.

KATO, C. V. D. **Desenvolvimento de um sistema de automação residencial com microcontrolador ESP32 e protocolo MQTT**. 2024. Trabalho de Conclusão de Curso – Universidade Estadual Paulista, *[S. l.]*, 2024.

RESENDE, K. **Kotlin com Android**: crie aplicativos de maneira fácil e divertida. 1. ed. *[S. l.]*: Kufunda, 2024.

SILVA, R. **Desenvolvendo aplicações mobile com Android Studio**. Osasco: Fundação Bradesco, 2023.

SILVA, W. R. P.; BRITO, D. G. P.; PURIFICAÇÃO, M. M. **Entendendo a cegueira botânica**: a relação humana com as plantas. Revista Ibero-Americana de Humanidades, Ciências e Educação, v. 10, n. 12, p. 1601-1607, 2024.

Agradecimentos

Os autores agradecem aos Colégios Univap – Unidade Centro pelo apoio institucional e pela infraestrutura disponibilizada para o desenvolvimento do projeto.