Revise! Um Aplicativo para Inspeção Padronizada de Colmeias de *Apis mellifera* no Semiárido Nordestino

Emilly Paiva Belo[1], **Antonio Rafael Braga**[1], **Breno Magalhães Freitas**[2]

[1] Universidade Federal do Ceará (UFC) – Campus Quixadá
Av. José de Freitas Queiroz, 5003 – 63902-580 – Quixadá, CE, Brasil

[2]Universidade Federal do Ceará (UFC), Centro de Ciências Agrárias
Av. Mister Hull, 2977, Bloco 847 – 60356-000 – Fortaleza, CE, Brasil

emillypaiva@alu.ufc.br, rafaelbraga@ufc.br, freitas@ufc.br

Abstract. *Beekeeping plays a fundamental role in environmental sustainability and biodiversity conservation, especially in semiarid regions where continuous monitoring of hive conditions is essential for colony survival. This paper presents Revise!, a mobile application designed to support structured and standardized apiary inspections. The application offers a guided workflow to record inspection data, enabling historical tracking of colonies and offline data collection in areas with limited connectivity through an offline-first approach and later synchronization. In addition, we report a preliminary user evaluation using the UEQ questionnaire with beekeepers from the Sertão Central region in Ceará, Brazil, indicating positive perceptions regarding usability and reliability.*

Resumo. *A apicultura desempenha papel fundamental na sustentabilidade ambiental e na conservação da biodiversidade, especialmente em regiões semiáridas, onde o monitoramento das colmeias é decisivo para a sobrevivência das colônias. Este artigo apresenta o Revise!, um aplicativo móvel voltado ao suporte de inspeções apícolas estruturadas e padronizadas. A aplicação oferece um fluxo guiado para o registro das revisões, possibilita acompanhamento histórico e incorpora funcionamento offline por meio de uma abordagem offline-first, com sincronização posterior. Adicionalmente, é apresentada uma avaliação com apicultores do Sertão Central do Ceará, Brasil, utilizando o método UEQ, indicando percepções positivas quanto à facilidade, confiabilidade e atratividade da ferramenta.*

1. Introdução

A apicultura possui relevância econômica e ambiental, contribuindo para a geração de renda e para a manutenção de serviços ecossistêmicos, como a polinização [Santos et al. 2016]. No semiárido nordestino, longos períodos de estiagem e altas temperaturas impactam a disponibilidade de recursos e elevam o risco de perdas de abelhas por estresse, fome ou manejo inadequado. Nesse cenário, revisões periódicas e bem conduzidas tornam-se essenciais para acompanhar a colônia, detectar problemas precocemente (p. ex., sinais de doenças e pragas) e orientar intervenções [Spivak and Reuter 2001, Tarpy et al. 2013, Mumbi et al. 2014].

A realização de inspeções padronizadas em colmeias é fundamental para o manejo apícola eficiente, pois permite que os apicultores monitorem a saúde e o desenvolvimento das colônias de abelhas ao longo do tempo com mais eficiência. Apesar dessa

importância, na prática, o registro das revisões ainda é predominantemente manual e pouco estruturado, dificultando a comparação histórica e a tomada de decisão baseada em dados [Zacepins et al. 2012]. Soma-se a isso um obstáculo recorrente: muitos apiários localizam-se em áreas sem conectividade estável, o que limita o uso de ferramentas digitais que dependem de internet constante [Milanez et al. 2020]. Um processo padronizado permite monitorar o bem-estar das colônias de forma sistemática, além de acompanhar a evolução da produtividade dos apiários ao longo do tempo. Com um histórico bem definido em mãos, decisões mais assertivas, baseadas em dados, poderão ser tomadas.

Algumas soluções digitais já foram propostas para apoiar a gestão de apiários, como os aplicativos Apilog[1] e Beekeeping App[2], que oferecem funcionalidades para registro de inspeções e acompanhamento do estado das colônias. Apesar do potencial dessas tecnologias, a adoção de ferramentas digitais na apicultura ainda enfrenta desafios significativos em regiões rurais, especialmente no semiárido brasileiro, onde limitações de conectividade, infraestrutura e até mesmo de interfaces e documentações em língua inglesa, o que pode representar uma barreira de usabilidade dificultam o uso de sistemas baseados em nuvem. Assim, aplicações móveis com suporte *offline* representam uma alternativa promissora para viabilizar a coleta de dados e a gestão de informações em campo.

Diante desse contexto, este artigo apresenta o Revise!, um aplicativo móvel desenvolvido para apoiar inspeções padronizadas em apiários do semiárido, com registro guiado, histórico por colmeia e funcionamento *offline* com sincronização posterior. O objetivo do trabalho é descrever as principais funcionalidades do aplicativo, detalhar a metodologia de desenvolvimento (incluindo a construção do formulário e a implementação *offline-first*) e relatar resultados iniciais de avaliação com usuários.

2. Trabalhos Relacionados

Nos últimos anos, o avanço de tecnologias digitais tem impulsionado o desenvolvimento da chamada *precision beekeeping*, que consiste no uso de sensores, sistemas computacionais e análise de dados para monitorar colônias de abelhas e apoiar decisões de manejo apícola [Danieli et al. 2023, Sabic et al. 2025]. Essas tecnologias buscam tornar a atividade mais eficiente, sustentável e baseada em dados.

Uma das principais linhas de pesquisa envolve o uso de sensores para monitoramento contínuo das condições das abelhas. Sistemas de monitoramento podem registrar variáveis como temperatura, umidade, peso da colmeia e atividade das abelhas, permitindo acompanhar o estado das colônias e detectar eventos relevantes, como enxameação ou escassez de recursos [Meikle and Holst 2015, Cecchi et al. 2020]. Esses sistemas possibilitam a observação de padrões comportamentais e ambientais da colônia.

Outra abordagem relevante envolve plataformas de *precision apiculture* baseadas em sensores ambientais e análise de dados. [Catania and Vallone 2020] propuseram uma plataforma de monitoramento que integra sensores meteorológicos e ambientais para analisar fatores que influenciam a produção de mel. Esse tipo de abordagem permite correlacionar condições ambientais com produtividade e comportamento das colônias.

Mais recentemente, soluções baseadas em Internet das Coisas (IoT) e computação

[1]Disponível em: `https://apilog.app/en`
[2]Disponível em: `https://beekeeping.app/`

distribuída têm sido propostas para monitoramento remoto de apiários. Sistemas como WaggleNet utilizam redes de sensores e protocolos de comunicação de baixo consumo, como LoRa, para coletar dados ambientais dentro e ao redor das colmeias e transmiti-los para plataformas de análise em nuvem [Jeon et al. 2025]. Essas arquiteturas permitem a coleta contínua de dados mesmo em ambientes rurais com infraestrutura limitada.

Além disso, técnicas de aprendizado de máquina e visão computacional têm sido exploradas para automatizar a análise do comportamento das abelhas [Huet et al. 2026]. Sistemas recentes utilizam modelos de detecção de objetos para identificar atividades de forrageamento, presença de pólen e até mesmo infestação por ácaros Varroa diretamente na entrada da colmeia [Narcia-Macias et al. 2023]. Esses métodos permitem monitoramento automatizado sem a necessidade de intervenção manual frequente.

Apesar do avanço dessas soluções tecnológicas, revisões recentes apontam que muitos sistemas de apicultura digital dependem de infraestrutura tecnológica relativamente complexa, incluindo sensores especializados e conectividade contínua com a internet [Danieli et al. 2023, Sabic et al. 2025]. Em regiões rurais ou em contextos de apicultura de pequena escala, essas exigências podem dificultar a adoção dessas tecnologias.

Assim, ferramentas digitais voltadas à coleta estruturada de dados durante inspeções de campo permanecem relevantes como estratégia complementar para apoiar a digitalização da apicultura. Desse modo, as inspeções de campo representam uma alternativa complementar. Em vez de depender exclusivamente de sensores, essas ferramentas permitem digitalizar o processo tradicional de inspeção manual das colmeias, mantendo registros históricos estruturados e possibilitando posterior análise dos dados coletados.

3. Metodologia

O aplicativo Revise![3] foi projetado para uso em dispositivos móveis Android, priorizando simplicidade de interação em ambiente de campo. A interface organiza as informações em três níveis hierárquicos: apiário, colmeia e inspeção. O fluxo de interação inicia com a seleção do apiário, colmeia e, por fim, do preenchimento guiado do formulário de revisão. Essa estrutura busca minimizar erros de registro e facilitar a navegação mesmo durante atividades operacionais no campo. A Figura 1 apresenta uma visão geral do aplicativo, com as funcionalidade de cadastro e listagem de apiários, início de cadastro de revisão, listagem de revisões e edição de revisões.

O desenvolvimento do Revise! foi guiado por necessidades reais de apicultores do semiárido, levantadas por meio de entrevistas e validações iterativas do fluxo de revisão. A metodologia foi organizada em seis etapas: (1) construção e validação do formulário de inspeção, (2) prototipação e implementação do *front-end*, (3) prototipação e implementação do *back-end*, (4) implementação do modo *offline* com abordagem *offline-first*, (5) testes internos e (6) avaliação com usuários finais.

3.1. Construção e validação do formulário de inspeção

A primeira etapa consistiu em definir um formulário de revisão padronizado e compatível com o manejo das regiões semiáridas do nordeste do Brasil. Para isso, foram realizadas

[3]Disponível em: `https://play.google.com/store/apps/details?id=com.colonymonmobile`

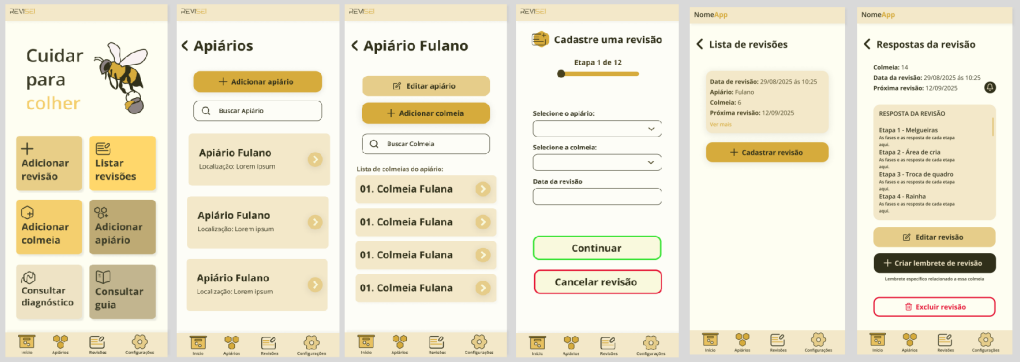

Figura 1. Visão geral do Revise!: navegação entre apiários, colmeias e início de uma revisão.

entrevistas periódicas com um especialista e com o cliente final do projeto (os apicultores), a fim de: (i) identificar quais indicadores são comumente observados durante revisões reais; (ii) transformar observações práticas em perguntas objetivas e registráveis; (iii) definir obrigatoriedade e formato de respostas (seleção, contagem, texto); (iv) revisar a ordem das perguntas para reduzir retrabalho durante a abertura da colmeia.

Um exemplo de uma folha de inspeção que foi estudada foi a utilizada na Inglaterra [BBKA 2026][4], que coleta as seguintes informações: *Date of visit*: data da inspeção, *Q*: presença da abelha rainha, *Brood*: estado da colônia, ex.: se é possível ver os ovos, se as abelhas estão presentes, *Store*: quantidade de alimento armazenado disponível, *Room*: espaço suficiente para o crescimento da colônia, *Health*: estado geral de saúde da colônia e *Varroa destructor*: presença do ácaro *Varroa destructor*.

Outro exemplo de uma folha de inspeção que foi usada como inspiração é a utilizada nos Estados Unidos da América [Jensen 2018]. As informações coletados são [5]: estágio dos membros da colônia (ovos, larvas e pupas) e suas quantidades, se a colônia tem abelhas adultas, se a rainha está presente, se possui reserva de mantimentos suficientes dentro e fora da colmeia, se tem espaço adequado e seguro para o tamanho da colônia a curto prazo para colocar os ovos.

A estrutura final do formulário foi definida com foco em cinco pontos de verificação recorrentes na literatura e na prática: presença da rainha (via postura), disponibilidade de espaço, reservas de alimento, sinais de enxameação e sintomas de doenças/ataques. Desses eixos, o formulário foi expandido para suportar maior detalhamento, incluindo:

- registro do número de melgueiras e preenchimento individualizado por melgueira;
- quadros(melgueiras e ninho), conforme necessidades do acompanhamento;
- sanidade e condições das crias, incluindo indícios de anomalias observáveis;
- presença de inimigos naturais e sinais de estresse.

3.2. *Backend* do aplicativo

O *backend* do aplicativo desempenha um papel fundamental na segurança, escalabilidade e eficiência do sistema. Ele foi projetado para garantir uma comunicação confiável entre

[4]https://www.bbka.org.uk/
[5]https://www.bayer.com/sites/default/files/HCC%20template_v2020Apr09.pdf

o cliente e o servidor, protegendo os dados sensíveis e otimizando o processamento das informações coletadas. Para alcançar esses objetivos, adotamos as seguintes abordagens tecnológicas:

1. *Node.js* e *TypeScript* – Optamos por utilizar o ambiente de execução *Node.js* em conjunto com *TypeScript* para garantir um código mais robusto e seguro. O *TypeScript* possibilita a detecção precoce de erros e melhora a manutenção do sistema ao adicionar tipagem estática ao *JavaScript*.
2. Autenticação com *JSON Web Tokens* (JWT) – Para garantir a segurança e controle de acesso à API, implementamos JWT para autenticação dos usuários. Com essa abordagem, cada requisição ao servidor carrega um *token* seguro, validando a identidade do usuário sem a necessidade de armazenamento contínuo de sessões no servidor.
3. Servidor com *Express.js* – O backend foi construído utilizando o *framework Express.js*, que oferece uma estrutura flexível para a criação de rotas, gerenciamento de *middlewares* e tratamento de requisições HTTP de maneira eficiente.
4. Banco de Dados Relacional com *PostgreSQL* – foi escolhido devido à sua confiabilidade, suporte a transações complexas e escalabilidade. Para facilitar a interação com o banco, utilizamos o ORM Prisma, permitindo consultas mais intuitivas e um melhor gerenciamento das entidades do sistema.
5. Documentação da API com *Swagger* – A API REST foi documentada, permitindo a geração automática de documentação interativa. Isso facilita a integração com outras aplicações e o entendimento das requisições disponíveis.

A estrutura da nossa API REST segue os princípios RESTful, o que facilita a compreensão e a interação com os recursos da aplicação. A escolha dessas tecnologias considerou requisitos de portabilidade, escalabilidade, manutenção e suporte ao desenvolvimento multiplataforma. O uso de React Native permitiu acelerar o desenvolvimento da aplicação móvel com compartilhamento de componentes e lógica entre interfaces. O Node.js foi adotado devido à sua arquitetura orientada a eventos e boa integração com aplicações REST em tempo real. O PostgreSQL foi escolhido pela robustez no tratamento transacional e confiabilidade no armazenamento dos dados das revisões. O Prisma contribuiu para maior produtividade e segurança na manipulação das entidades persistidas, reduzindo erros de acesso ao banco de dados.

3.3. *Front-end* do aplicativo

Com o conteúdo do formulário validado, iniciou-se a prototipação das telas, levando em conta aspectos como a objetividade da linguagem e das opções de resposta, a quantidade de telas, esforço de preenchimento e a organização visual para leitura rápida em campo.

Após essa validação visual, o formulário foi implementado em *React Native*, mantendo a estrutura modular do projeto e separando componentes de interface, validações e navegação. Uma necessidade técnica relevante foi a navegação dinâmica: a quantidade de melgueiras informada pelo usuário determina quantas vezes um conjunto de telas deve ser repetido no fluxo de revisão. Para atender a esse requisito, foi implementado um mecanismo de controle de estado que garante: (i) repetição previsível do bloco de telas conforme o número de melgueiras; (ii) armazenamento temporário consistente dos dados entre telas; (iii) validação de campos obrigatórios antes de avançar. Na Figura 2 é possível observar fluxo da revisão guiada no formulário em 4 etapas específicas do processo

de revisão, no caso, estado da melgueira, estado do quadros do ninho, presença da rainha e estado geral da colméia.

Figura 2. Exemplo do fluxo guiado de inspeção no Revise! com algumas das etapas sequenciais do formulário.

3.4. Modo *offline* com abordagem *offline-first*

Neste trabalho, o conceito de *offline-first* é tratado como uma estratégia arquitetural de desenvolvimento de aplicações móveis, na qual as funcionalidades principais da aplicação permanecem operacionais independentemente da disponibilidade de conexão com a internet. Nessa abordagem, os dados são inicialmente persistidos localmente e sincronizados posteriormente com serviços remotos quando a conectividade é restabelecida.

Considerando a conectividade limitada em áreas rurais [Milanez et al. 2020], a coleta de dados em campo foi tratada como requisito central. O Revise! adota uma abordagem *offline-first*, na qual o registro da revisão não depende de internet para funcionar [Developers 2024]. Em suma, o aplicativo segue os seguintes passos: (i) armazena os dados de revisões em um banco local no dispositivo; (ii) mantém uma fila de sincronização para revisões pendentes; (iii) realiza a sincronização automática quando há conectividade, sem exigir ação do usuário e (iv) preserva integridade dos dados, evitando perda de registros em campo. A primeira tela da esquerda para a direita da Figura2 apresenta a tela do aplicativo quando o mesmo não tem conexão com a internet.

A motivação para essa estratégia é reduzir o risco operacional de o apicultor não conseguir realizar da revisão ou perda a revisão feita no aplicativo. Em versões preliminares, o uso offline dependia de ações prévias do usuário (ex.: pré-carregar dados). Com o *offline-first*, o apicultor consegue registrar revisões mesmo se esquecer de qualquer preparação prévia, aumentando a continuidade de uso e a confiabilidade percebida.

3.5. Testes internos

Antes da realização da avaliação com usuários, foi conduzida uma etapa de testes internos com o objetivo de verificar a estabilidade do aplicativo e validar o funcionamento das principais funcionalidades. Esses testes foram planejados e executados pela equipe de desenvolvimento, utilizando uma abordagem baseada em casos de teste estruturados.

Doze casos de teste foram definidos cobrindo os principais fluxos de uso do sistema, incluindo autenticação e criação de contas de usuário, gerenciamento de apiários e colmeias, criação e edição de revisões e navegação entre as telas da aplicação. Cada caso de teste descrevia o cenário de execução, as ações esperadas do usuário e o comportamento esperado do sistema, permitindo verificar a consistência funcional do aplicativo.

Considerando que o Revise! adota uma arquitetura baseada no paradigma *offline-first*, uma parte significativa dos testes foi dedicada à validação do comportamento do aplicativo em cenários de conectividade limitada ou inexistente. Nesses testes, foram avaliados aspectos como o registro de revisões em modo offline, a persistência de dados no armazenamento local do dispositivo, a manutenção das informações após reinicialização do aplicativo e o processo de sincronização automática com o servidor quando a conectividade é restabelecida.

Além disso, foram realizados testes de validação da interface e do fluxo de interação, incluindo a verificação de campos obrigatórios no formulário de inspeção, a consistência da navegação entre etapas da revisão guiada e a adaptação da interface em cenários online e offline.

Os testes foram executados em diferentes dispositivos Android e também em ambientes de emulação, permitindo avaliar o comportamento do aplicativo em múltiplas configurações de hardware e versões do sistema operacional. Os resultados obtidos indicaram que as funcionalidades avaliadas apresentaram comportamento consistente com os resultados esperados, possibilitando avançar para a etapa de avaliação com usuários. A Tabela 1 mostra uma lista não exaustiva dos principais casos de testes realizados pela equipe de desenvolvimento e testes.

Tabela 1. Exemplos de casos de teste do aplicativo Revise!

Caso	Descrição	Resultado esperado
CT01	Criação de conta de usuário	Usuário criado com sucesso
CT02	Cadastro de apiário	Apiário salvo no banco
CT03	Registro de revisão offline	Revisão salva localmente
CT04	Sincronização após reconexão	Dados sincronizados com servidor
CT05	Validação de campos obrigatórios	Sistema impede avanço sem preenchimento

3.6. Avaliação com usuários finais

Os participantes da avaliação foram selecionados por conveniência, considerando apicultores com experiência prática em manejo de colmeias na região do Sertão Central do Ceará. Como critérios de inclusão, buscou-se selecionar participantes que realizassem revisões periódicas em apiários e que utilizassem dispositivos móveis Android.

Para avaliar a experiência de uso após as mudanças (modo *offline* e melhorias de interface), foi realizada uma avaliação preliminar com cinco apicultores atuantes no Sertão Central do Ceará. A coleta ocorreu via plataforma Google Forms (Google LLC)[Google LLC 2024], com encontros online e presenciais, em que os participantes foram instruídos a instalar e utilizar a versão atualizada do aplicativo e, em seguida, responder ao instrumento.

O método utilizado foi o *User Experience Questionnaire* (UEQ)

[Laugwitz et al. 2008], com seis dimensões: Atratividade, Facilidade (Perspicuidade), Eficiência, Confiabilidade (*Dependability*), Estímulo e Originalidade. As respostas são dadas em escala de 1 a 7 por pares de termos opostos, refletindo a percepção do participante sobre a experiência. Além dos itens UEQ, o questionário incluiu perguntas sobre o cenário de conectividade no local de uso, para evidenciar do modo *offline*.

O UEQ foi escolhido por permitir avaliar simultaneamente aspectos pragmáticos e hedônicos da experiência do usuário, incluindo eficiência, atratividade, estímulo e confiabilidade. Diferentemente de instrumentos como o SUS, que possui foco predominantemente em usabilidade, ou modelos como TAM e UTAUT, voltados principalmente à aceitação tecnológica, o UEQ oferece uma avaliação mais ampla da experiência de interação, sendo útil para análises exploratórias em contexto de uso real.

A avaliação realizada possui caráter exploratório e não envolveu coleta de dados sensíveis, intervenções clínicas ou identificação individual dos participantes. Os apicultores participaram voluntariamente, mediante consentimento informado, sendo preservado o anonimato das respostas, por esse motivo não foi necessário a submissão do projeto ao Comitê de Ética (CEP).

4. Resultados e Discussão

4.1. Avaliação qualitativa e relatos do usuário

Padronização do registro de revisão e histórico de inspeções são as principais contribuições do Revise!, haja vista que, tradicionalmente, as revisões são variáveis entre apicultores e regiões, transformando em um processo mais consistente, com um fluxo guiado, que (i) reduz a omissão de informações por esquecimento; (ii) melhora a comparabilidade entre revisões e (iii) apoia a construção de um histórico por colmeia e por apiário. Além disso, o detalhamento por melgueira e por quadros permite acompanhamento mais granular, favorecendo a identificação de padrões e anomalias.

Os relatos obtidos na avaliação confirmaram a conectividade como barreira relevante: a maioria dos participantes reportou acesso ruim ou inexistente à internet no local das colmeias, reforçando a adequação do *offline-first*. Entre os comentários qualitativos, destacou-se a percepção de redução de esforço operacional: "*Facilita demais não precisar lembrar de baixar antes de sair pro campo.*"

4.2. Resultados da avaliação UEQ

A Tabela 2 apresenta as médias observadas em cada dimensão do UEQ. As maiores pontuações ocorreram em Facilidade e Originalidade (6,60), indicando que os participantes perceberam o aplicativo como intuitivo e a solução offline como um diferencial relevante. Atratividade (6,47) e Confiabilidade (6,53) também se destacaram, sugerindo conforto no uso e previsibilidade do comportamento do sistema.

Eficiência (5,60) e Estímulo (5,60) foram ligeiramente inferiores, apontando oportunidades de melhoria relacionadas à rapidez no preenchimento de revisões e ao engajamento contínuo. Um feedback recorrente foi o desejo de um fluxo ainda mais ágil para perguntas repetitivas, por exemplo: "*As revisões funcionam bem, mas seria muito mais viável adotar um formulário de perguntas mais rápido de ser respondido.*"

Tabela 2. Resultados médios do UEQ obtidos na avaliação do aplicativo Revise! com apicultores do Sertão Central do Ceará (escala 1–7).

Dimensão UEQ	Média
Atratividade	6.47
Facilidade (Perspicuidade)	6.60
Eficiência	5.60
Confiabilidade (Dependability)	6.53
Estímulo	5.60
Originalidade	6.60

Os resultados obtidos na avaliação preliminar indicam que o aplicativo Revise! apresenta potencial para apoiar o processo de registro e organização de informações durante revisões de colmeias. A possibilidade de registrar revisões e armazenar os dados localmente no dispositivo móvel, com sincronização posterior, foi percebida como um fator importante para a viabilidade prática da ferramenta. De forma geral, os resultados sugerem que a solução proposta pode facilitar o gerenciamento histórico das colmeias e apoiar a tomada de decisão por parte dos apicultores. No entanto, a avaliação realizada possui caráter exploratório e foi conduzida com um número limitado de participantes. Dessa forma, como trabalhos futuros deseja-se a avaliação, com um conjunto maior de usuários e períodos mais longos de utilização em condições reais de campo.

5. Conclusão e Trabalhos Futuros

Este trabalho apresentou o Revise!, um aplicativo móvel para apoiar o monitoramento estruturado de apiários no semiárido do nordeste do Brasil, com fluxo padronizado de inspeção, registro histórico e funcionamento *offline* por abordagem *offline-first*. Os resultados iniciais indicam que a ferramenta atende à realidade de campo dos apicultores, mitigando limitações de conectividade e oferecendo organização e confiabilidade no registro de revisões.

Como trabalhos futuros, pretende-se aprimorar o aplicativo em três frentes principais. Primeiramente, busca-se reduzir o tempo necessário para o preenchimento do formulário de inspeção, por meio da implementação de mecanismos de respostas rápidas e do reaproveitamento de padrões de registros anteriores. Em seguida, pretende-se incorporar funcionalidades de análise automática dos dados coletados, permitindo a identificação de tendências ao longo do tempo. Por fim, espera-se evoluir o sistema para oferecer diagnósticos e recomendações de manejo baseados no histórico das revisões realizadas.

Referências

BBKA, T. B. B. A. (2026). Bbka hive keeping record sheet. `https://portsmouthbeekeepers.co.uk/wp-content/uploads/2017/12/{B}{B}{K}{A}_{H}ive_{R}ecording_{S}ystem.pdf`. [Acessado em: 01/03/2026].

Catania, P. and Vallone, M. (2020). Application of a precision apiculture system to monitor honey daily production. *Sensors*, 20(7):2012.

Cecchi, S., Spinsante, S., Terenzi, A., and Orcioni, S. (2020). A smart sensor-based measurement system for advanced bee hive monitoring. *Sensors*, 20(9):2726.

Danieli, P. P., Addeo, N. F., Lazzari, F., Manganello, F., and Bovera, F. (2023). Precision beekeeping systems: State of the art, pros and cons, and their application as tools for advancing the beekeeping sector. *Animals*, 14(1):70.

Developers, A. (2024). Build an offline-first app | App architecture | Android Developers — developer.android.com. `https://developer.android.com/topic/architecture/data-layer/offline-first`. [Accessed 02-03-2026].

Google LLC (2024). Google forms. `https://www.google.com/forms/about/`. Acesso em: 14 maio 2026.

Huet, J.-C., Bougueroua, L., and Metidji, S. A. (2026). An intelligent monitoring system for forecasting and anomaly detection in precision beekeeping. *Scientific Reports*.

Jensen, S. (2018). Predicting honeybee health: The healthy colony checklist, hive scale and weather data. *Data & Analytics for Good*.

Jeon, M., Kim, J., Kim, S., Park, S., Zhang, B., and Smith, A. H. (2025). Wagglenet: A lora and mqtt-based monitoring system for internal and external beehive conditions. *arXiv preprint arXiv:2512.07408*.

Laugwitz, B., Held, T., and Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In *Symposium of the Austrian HCI and usability engineering group*, pages 63–76. Springer.

Meikle, W. G. and Holst, N. (2015). Application of continuous monitoring of honeybee colonies. *Apidologie*, 46(1):10–22.

Milanez, A. Y., Mancuso, R. V., Maia, G. B. d. S., Guimarães, D. D., Alves, C. E. A., Madeira, R. F., et al. (2020). Conectividade rural: situação atual e alternativas para superação da principal barreira à agricultura 4.0 no brasil.

Mumbi, C. T., Mwakatobe, A. R., Mpinga, I. H., Richard, A., and Machumu, R. (2014). Parasitic mite, varroa species (parasitiformes: Varroidae) infesting the colonies of african honeybees, apis mellifera scutellata (hymenoptera: Apididae) in tanzania. *Journal of Entomology and Zoology Studies*, 2(3):188–196.

Narcia-Macias, C. I., Guardado, J., Rodriguez, J., Rampersad-Ammons, J., Enriquez, E., and Kim, D.-C. (2023). Intellibeehive: An automated honey bee, pollen, and varroa destructor monitoring system. *arXiv preprint arXiv:2309.08955*.

Sabic, J., Perkovic, T., Solic, P., and Seric, L. (2025). Buzzing with intelligence: A systematic review of smart beehive technologies. *Sensors*, 25(17):5359.

Santos, J. O. d., Borges, P., et al. (2016). Um estudo sobre a evolução histórica da apicultura. *Programa de Pos Graduação em Sistemas Agroindustriais (24-Mestrado Profissional) Dissertações*, 4(1):93–p.

Spivak, M. and Reuter, G. S. (2001). Resistance to american foulbrood disease by honey bee colonies apis mellifera bred for hygienic behavior. *Apidologie*, 32(6):555–565.

Tarpy, D. R., Lengerich, E. J., Pettis, J. S., et al. (2013). Idiopathic brood disease syndrome and queen events as precursors of colony mortality in migratory beekeeping operations in the eastern united states. *Preventive veterinary medicine*, 108(2-3):225–233.

Zacepins, A., Stalidzans, E., and Meitalovs, J. (2012). Application of information technologies in precision apiculture.