MechTrack: Aplicativo Mobile para Gestão de Equipes Técnicas em Serviços Externos

Guilherme Moser Montes[1], Enrique Augusto da Roza[1]

[1]Centro Universitário Campo Real

Rua Comendador Norberto, 1299 - Santa Cruz – Guarapuava – PR – Brasil

{engs-guilhermemontes@camporeal.edu.br, prof_enriqueroza@camporeal.edu.br}

Resumo

Gerenciar equipes técnicas em serviços externos é um grande desafio para empresas de manutenção industrial e agroindústrias, especialmente devido aos conflitos gerados pelo uso de planilhas compartilhadas e à falta de privacidade no acesso às informações. Este trabalho apresenta o *MechTrack*, um aplicativo móvel que substitui as planilhas por uma solução digital com controle de acesso diferenciado, permitindo que os mecânicos visualizem apenas suas próprias informações, enquanto os supervisores acompanham de forma completa o desempenho da equipe. Desenvolvido com *React Native* e *NestJS*, o sistema possui autenticação por perfis, registro de atividades com geolocalização e controle de privacidade, além de recursos como funcionalidade *offline-first*, sincronização automática e cálculo de quilometragem. Os resultados demonstram que a solução é tecnicamente viável e capaz de reduzir conflitos internos, aprimorar o controle gerencial em tempo real e aumentar a eficiência operacional.

Palavras-chave: Controle gerencial. Offline-first. Manutenção industrial. Controle de acesso. React Native.

Abstract

Managing technical teams in field services is a major challenge for industrial maintenance companies and agribusinesses, particularly due to conflicts arising from the use of shared spreadsheets and the lack of privacy in information access. This work presents MechTrack, a mobile application that replaces spreadsheets with a digital solution featuring differentiated access control. The system allows mechanics to view only their own data, while supervisors can comprehensively

monitor team performance. Developed with React Native and NestJS, the system integrates profile-based authentication, activity logging with geolocation and privacy management. Additional features include offline-first functionality, automatic synchronization, and mileage calculation. The results demonstrate that the solution is technically feasible and capable of reducing internal conflicts, enhancing real-time managerial control, and improving operational efficiency.

Keywords: Management control. Offline-first. Industrial maintenance. Access control. React Native.

1. Introdução

O gerenciamento de equipes técnicas em serviços externos é um desafio cada vez maior para as empresas. Embora a transformação digital avance rapidamente, com 87% das empresas já adotando sistemas integrados (TIINSIDE, 2025), o controle operacional de equipes em campo ainda é realizado, em muitos casos, por meio de planilhas compartilhadas e registros manuais. Essa abordagem tem se mostrado não apenas insuficiente, mas também uma fonte de inconsistências e conflitos internos.

O setor de manutenção industrial, que movimentou R$14,88 bilhões somente em 2024 (REVISTA O EMPREITEIRO, 2025), depende muito da eficiência de suas equipes técnicas. Além disso, os técnicos em campo passam 40% de seu dia realizando atividades que não agregam valor, como por exemplo o preenchimento de horas trabalhadas (MCKINSEY, 2019). Isso mostra como as soluções tecnológicas voltadas para melhorar o gerenciamento das atividades externas são importantes para garantir mais privacidade e transparência nas operações.

Considerando este contexto, observou-se que, embora existam soluções digitais para gestão de equipes externas, como plataformas voltadas a promotores de vendas e equipes de *merchandising*, essas ferramentas não atendem de forma adequada às necessidades específicas de pequenas equipes técnicas no setor de manutenção industrial. Muitas empresas deste setor ainda utilizam planilhas online compartilhadas para registrar as atividades de campo, o que acaba expondo informações pessoais entre os membros da equipe. Quando técnicos e mecânicos

têm acesso a dados como horários, quilometragem, serviços realizados e até salários de colegas, podem surgir comparações e conflitos que afetam o ambiente de trabalho e a produtividade.

Além disso, os supervisores enfrentam dificuldades para acompanhar as atividades realizadas e gerar relatórios confiáveis, principalmente quando as equipes estão em locais com baixa conectividade. Assim, faz-se necessária uma solução que permita armazenar informações de maneira *offline* e sincronizá-las automaticamente quando a conexão for restabelecida, ao mesmo tempo em que implemente diferentes níveis de acesso e mecanismos de segurança para garantir o controle adequado das informações sensíveis.

Diante dessa situação, o objetivo principal deste trabalho é desenvolver um aplicativo móvel para a gestão das atividades de equipes técnicas externas. A ideia é substituir as planilhas por uma solução digital com controle de acesso segmentado, geolocalização e sincronização em tempo real, bem como estratégias *offline-first* para melhorar a usabilidade de usuários com limitações de conectividade à *internet* no ambiente de trabalho.

O projeto, chamado *Mechtrack,* foi pensado especialmente para o setor de manutenção industrial e agroindústrias. Considerando as especificidades deste setor, os objetivos específicos incluem: a criação de um sistema de *login* com perfis diferentes para mecânicos e supervisores, o desenvolvimento de uma interface simples e prática para uso em campo, a implementação de funcionamento *offline* e a integração de recursos de localização para calcular a quilometragem percorrida.

O aplicativo atende dois perfis principais: os técnicos e mecânicos, que precisam registrar suas atividades de forma rápida e simples, e os supervisores, que precisam acompanhar o andamento das operações e tomar decisões com base em dados precisos. O desenvolvimento levou em conta as condições do setor agrícola, que muitas vezes necessita de ferramentas que funcionem bem mesmo em locais com pouca *internet*.

A motivação para este trabalho surgiu do relato de um profissional da área que enfrentou problemas em sua empresa devido ao uso de planilhas compartilhadas. A exposição de informações sigilosas causou desconforto entre os

colaboradores e mostrou como a falta de privacidade pode prejudicar o ambiente de trabalho e reduzir a eficiência das equipes. As competências desenvolvidas ao longo do curso de Engenharia de *Software* possibilitaram propor uma solução que integra privacidade, praticidade e eficiência.

2. Referencial teórico

Este capítulo apresenta os conceitos fundamentais que sustentam o desenvolvimento do *MechTrack*. A compreensão desses fundamentos é essencial para contextualizar as escolhas tecnológicas e metodológicas adotadas. São abordados os aspectos da gestão de manutenção industrial, as tecnologias utilizadas no desenvolvimento voltado à dispositivos móveis e lógica do sistema, os mecanismos de controle de acesso, recursos de geolocalização e questões relacionadas à privacidade de dados no ambiente corporativo.

2.1 Gestão de Manutenção Industrial e Equipes Externas

A manutenção industrial é uma função estratégica essencial para a competitividade das empresas. A gestão moderna de manutenção passou de uma abordagem reativa para modelos preditivos e prescritivos, o que exige sistemas de informação cada vez mais sofisticados, uma vez que “a eficiência da manutenção está diretamente relacionada à qualidade das informações disponíveis para tomada de decisão” (KARDEC; NASCIF, 2022, p. 87).

No que diz respeito às equipes externas, Xenos (2020) aponta que a rastreabilidade das atividades e a documentação precisa como fatores críticos para a eficácia da gestão de manutenção em campo. A falta desses fatores pode levar à perda de informações, à redução da visibilidade gerencial e, consequentemente, ao comprometimento da qualidade dos serviços prestados e da tomada de decisão pelos gestores responsáveis.

A norma ABNT NBR 5462:2021 define manutenção como uma “combinação de ações técnicas, administrativas e gerenciais durante o ciclo de vida de um item, destinadas a mantê-lo ou recolocá-lo em um estado no qual possa desempenhar a função requerida” (ABNT, 2021), destacando a importância de sistemas que integrem o registro de atividades, o controle de equipes e a geração de relatórios.

2.2 Desenvolvimento *Mobile* com *React Native*

O desenvolvimento de aplicações móveis multiplataforma tem se consolidado como alternativa eficiente ao desenvolvimento nativo tradicional. *React Native, framework* criado pelo Facebook (atual Meta) em 2015, possibilita a criação de aplicações móveis para *iOS* e *Android* utilizando *JavaScript* e *React*, com desempenho próximo ao nativo (FACEBOOK, 2023).

O *React Native* se destaca por sua capacidade de compartilhar código entre *iOS* e *Android*, o que pode reduzir os custos de desenvolvimento em até 50% enquanto mantém a experiência nativa do usuário (EINSENMAN, 2021, p.23). Além disso, o *framework* se destaca por ter uma comunidade ativa, um ecossistema repleto de bibliotecas e um suporte corporativo forte, tudo isso ajuda na manutenção e evolução contínua das aplicações.

A escolha pelo *React Native* ao invés de alternativas como *Flutter* ou desenvolvimento nativo faz sentido por vários motivos: a maturidade do ecossistema, a quantidade de desenvolvedores disponíveis no mercado, a fácil integração com *APIs JavaScript* e, principalmente, o suporte sólido para funcionalidades *offline.*

Para aplicações que precisam funcionar em áreas com conectividade limitada, algo essencial no contexto agrícola e industrial, Eisenman (2021) destaca a importância do *AsyncStorage* e de bibliotecas como *WatermelonDB,* que implementam padrões eficientes de sincronização de dados.

A integração nativa com sensores do dispositivo, como o GPS, permite a implementação de funcionalidades como a geolocalização sem depender de soluções externas complexas. Essa característica é muito relevante para o objetivo de implementar recursos de localização para calcular a quilometragem percorrida.

2.3 Arquitetura *Backend* com *NestJS* e *APIs RESTful*

O *NestJS*, *framework NodeJS* com suporte nativo a *TypeScript*, implementa princípios de *design* orientado a objetos e programação funcional reativa (KAMIL, 2023). Segundo a documentação oficial, o *NestJS* oferece uma arquitetura pronta que possibilita o desenvolvimento de aplicações altamente testáveis, escaláveis, com baixo acoplamento e de fácil manutenção (NESTJS, 2024). Essa arquitetura

modular baseada em controladores, serviços e módulos, favorece a aplicação de padrões como a injeção de dependência e a separação de responsabilidades.

Para *APIs RESTful*, Masse (2021) define princípios essenciais de *design* que incluem o uso adequado de métodos *HTTP (Hypertext Transfer Protocol)*, a organização lógica de recursos e a implementação correta de códigos de status. O *NestJS* adota esses padrões de forma nativa por meio de decoradores *TypeScript*, simplificando o desenvolvimento de *APIs (Application Program Interface)* conforme as melhores práticas.

A integração entre *React Native* e *NestJS* possibilita a implementação de uma sincronização de dados eficiente, essencial para a comunicação em tempo real e para estratégias *offline-first*. Nessas estratégias, os dados são armazenados localmente no dispositivo móvel e posteriormente sincronizados com o servidor quando a conectividade é restabelecida.

2.4 Controle de Acesso Baseado em Perfis e Segurança

A implementação de controle de acesso diferenciado é um requisito essencial para sistemas que lidam com informações sensíveis de vários usuários. O Controle de Acesso Baseado em Papéis (RBAC - *Role-Based Access Control*) é um modelo no qual as permissões são atreladas a perfis específicos e os usuários são associados conforme suas responsabilidades, o que facilita a gestão das autorizações e reduz a complexidade do controle de acesso em sistemas multiusuário (FERRAIOLO; KUHN, 2020, p. 156).

No caso do *MechTrack*, o modelo RBAC fundamenta a criação de um sistema de autenticação com perfis distintos, permitindo que mecânicos vejam apenas suas próprias informações, enquanto supervisores têm acesso aos dados da equipe. Essa separação de acesso resolve o problema que surgia com o uso de planilhas compartilhadas, que expunha informações sensíveis entre os membros da equipe, gerava conflitos e comprometia o ambiente de trabalho.

A autenticação baseada em *JWT* (*JSON Web Tokens* - *tokens* compactos e assinados digitalmente que carregam informações do usuário) tornou-se o padrão para aplicações sem estado. Segundo Jones e Bradley (2021), os *JWTs* permitem a transmissão segura de informações entre partes por meio de objetos *JSON*

compactos e auto-contidos, sendo ideais para ambientes distribuídos. Essa característica auto-contida torna a autenticação em aplicações *mobile* mais simples, eliminando a necessidade de consultas frequentes ao servidor para validar a sessão.

Para aplicações *mobile*, a *Open Web Application Security Project* (OWASP*)* — organização internacional sem fins lucrativos dedicada a promover boas práticas de segurança em *software* — define no *Mobile Application Security Verification Standard* requisitos de segurança que incluem o armazenamento seguro de credenciais, comunicação criptografada via *HTTPS (Hypertext Transfer Protocol Secure)* e validação rigorosa de entradas de dados (OWASP, 2023). A adoção desses requisitos assegura que informações sensíveis, como *tokens* de autenticação e dados de atividades, permaneçam protegidas contra acessos não autorizados.

2.5 Geolocalização e Cálculo de Quilometragem

A geolocalização constitui recurso fundamental para sistemas de gestão de equipes externas. Os dispositivos móveis integram receptores GPS que fornecem coordenadas geográficas com precisão variável conforme as condições ambientais. Para o cálculo de distâncias entre coordenadas, a fórmula de Haversine, apresentada por Sinnott (1984), calcula a distância entre dois pontos na superfície de uma esfera considerando suas latitudes e longitudes, fornecendo resultados precisos para os deslocamentos diários de equipes técnicas.

No ecossistema *React Native*, certas bibliotecas oferecem *APIs* simplificadas para acesso aos serviços de localização, abstraindo diferenças entre *iOS* e *Android* (REACT NATIVE COMMUNITY, 2024). Em áreas rurais e industriais, a precisão do GPS pode ser afetada por obstrução de sinal e interferência eletromagnética, exigindo implementações robustas com técnicas de filtragem de dados imprecisos.

2.6 Privacidade de Dados e Gestão de Conflitos Organizacionais

A Lei Geral de Proteção de Dados (Lei 13.709/2018) estabelece princípios fundamentais para o tratamento de dados pessoais. A proteção de dados dos colaboradores exige equilíbrio entre as necessidades gerenciais legítimas e os direitos individuais à privacidade (BIONI, 2021). Conflitos organizacionais podem

surgir de comparações sociais inadequadas, fenômeno intensificado quando informações sobre desempenho individual são expostas inapropriadamente, o que reforça a necessidade de sistemas que equilibrem a proteção da privacidade individual com a transparência operacional necessária para uma gestão eficaz (ROBBINS; JUDGE, 2023).

Além disso, a teoria da equidade (ADAMS, 1965) demonstra que percepções de injustiça nas relações entre esforço e recompensa geram desmotivação, justificando controles de acesso diferenciados para o aumento da privacidade dos usuários como proposto no *MechTrack*.

3. Estado da Arte

Este capítulo apresenta um mapeamento sistemático sobre aplicativos móveis para gestão de equipes técnicas em serviços externos. O estudo busca analisar se existem soluções que abordem especificamente o problema de privacidade e controle de acesso diferenciado em sistemas de gestão de manutenção, além de identificar tecnologias utilizadas e funcionalidades implementadas.

A principal contribuição do estudo é levantar informações sobre ferramentas existentes, suas abordagens de privacidade e lacunas que justifiquem o desenvolvimento do *MechTrack*. O mapeamento sistemático é uma metodologia que compreende uma pesquisa criteriosa na literatura, tendo como objetivo avaliar a abrangência, o escopo e o volume de estudos publicados dentro de um domínio específico de interesse (PETERSON et al, 2008).

3.1 Questões da Pesquisa

A pesquisa teórica busca identificar respostas para questões centrais do projeto, possibilitando a análise de trabalhos e iniciativas semelhantes. Tal investigação oferece suporte teórico ao estudo, ao evidenciar soluções, limitações e práticas consolidadas que podem orientar e aperfeiçoar o desenvolvimento do projeto.

As Questões de Pesquisa (QP) consideradas neste mapeamento são as seguintes:

- **QP1:** Quais são as soluções tecnológicas existentes para gestão de equipes técnicas em serviços externos?

- **QP2:** Como as soluções identificadas abordam a questão de privacidade e controle de acesso diferenciado?
- **QP3:** Quais tecnologias *mobile* são utilizadas no desenvolvimento de aplicações para gestão de manutenção industrial?
- **QP4:** Existem aplicações móveis específicas para o setor agrícola e agroindustrial que consideram conectividade intermitente?
- **QP5:** Quais funcionalidades são consideradas essenciais em sistemas de gestão de atividades externas para equipes técnicas?

3.2 String de Busca

Para a identificação dos trabalhos acadêmicos, foram elaboradas *strings* de busca contendo palavras-chaves relacionadas ao tema dos artigos. A pesquisa foi conduzida na plataforma *Google Scholar*[1], amplamente reconhecida por seu uso em investigações acadêmicas.

O Quadro 1 mostra as *strings* utilizadas na pesquisa, estas *strings* foram escolhidas por combinar três conceitos fundamentais: a plataforma tecnológica, o domínio de aplicação e o controle de acesso.

Quadro 1 - *Strings* de busca

String	Expressão
1	("React Native" OR "aplicativo móvel") AND ("gestão de manutenção" OR "equipes técnicas") AND ("controle de acesso" OR "privacidade")
2	("aplicativo móvel" OR "mobile application") AND ("gestão agrícola" OR "agronegócio" OR "agricultura") AND ("offline" OR "sincronização" OR "controle de equipes")

Fonte: O Autor, 2026

3.3 Critérios de Seleção

Para a análise e seleção dos artigos, foram estabelecidos Critérios de Inclusão (CI):

- **CI1:** Artigos publicados nos últimos 6 anos (2019-2025).

[1] https://scholar.google.com/

- **CI2:** Artigos em português e inglês.
- **CI3:** Trabalhos que abordem desenvolvimento ou análise de aplicações móveis.
- **CI4:** Estudos relacionados à gestão de equipes, manutenção ou serviços de campo.

Após a aplicação dos CI, os artigos selecionados foram submetidos à análise dos Critérios de Exclusão (CE), resultando na eliminação daqueles que não os atenderam:

- **CE1**: Artigos duplicados.
- **CE2**: Artigos que não puderam ser acessados.
- **CE3**: Trabalhos que não abordaram aplicações móveis como foco principal.

3.4 Análise dos Artigos e Resultados

A partir das *strings* de busca aplicadas no *Google Scholar*, foram identificados inicialmente 544 trabalhos, 31 pela primeira *string* e 513 pela segunda. Todos atenderam aos critérios de inclusão e exclusão definidos para a pesquisa. Contudo, ao analisar seus conteúdos, foi observado que nenhum deles trata especificamente da gestão de equipes técnicas externas com controle de acesso diferenciado voltado à privacidade entre os membros da equipe. Ainda assim, alguns apresentaram informações e abordagens relacionadas ao mesmo contexto.

Dentre os trabalhos analisados, destaca-se o de Reigado (2019), que desenvolveu a aplicação *web* e *mobile* AgroPT para auxiliar entidades do setor agrícola na identificação de doenças em vinhas e oliveiras. Neste sentido, o autor comenta sobre a importância da conexão e da persistência de dados *offline* em ambientes externos para que, posteriormente, seja realizada uma sincronização. Entretanto, o trabalho não implementa questões de privacidade interna nem controle de funcionários por um supervisor.

Outro trabalho relevante é o de Silva, Basso e Ruellas (2019), que desenvolveram um aplicativo *mobile* para controle e gestão do agronegócio de hortifrúti voltado para pequenos e médios agricultores. Os autores propõem a digitalização de processos por meio de uma solução digital em *smartphones*,

permitindo o controle de frequência de funcionários, o acompanhamento dos serviços externos e as vendas de produtos. Contudo, o trabalho não aborda questões de privacidade entre membros da equipe ou controle de acesso diferenciado. Além disso, os autores não comentam sobre recursos de geolocalização para rastreamento de deslocamentos ou funcionalidades *offline-first*, os quais são requisitos importantes para o contexto de equipes técnicas em trabalhos externos, como abordado pelo *MechTrack*.

A partir dos trabalhos analisados, foram respondidas as questões de pesquisa da seguinte forma:

QP1: Quais são as soluções tecnológicas existentes para gestão de equipes técnicas em serviços externos?

As soluções mais comuns incluem sistemas ERP tradicionais com módulos *mobile* limitados, aplicações *web* responsivas adaptadas para dispositivos móveis e, em poucos casos, aplicativos *mobile-first* desenvolvidos especificamente para equipes de campo. As plataformas oferecem funcionalidades como registro de atividades, controle de horários, visualização de tarefas e geração básica de relatórios.

Porém, a maioria dessas soluções não foi projetada para as necessidades de equipes técnicas em campo, resultando em interfaces complexas que exigem múltiplos passos para executar tarefas simples, o que dificulta o uso em ambientes de trabalhos externos.

QP2: Como as soluções identificadas abordam a questão de privacidade e controle?

Há poucos estudos que exploram implementações sólidas de controle de acesso com foco em privacidade entre membros da equipe. Nos casos identificados, predominam sistemas simples de autenticação com dois níveis: administrador e usuário comum.

A maioria permite que supervisores acessem dados de toda a equipe, mas não impõem restrições para que membros visualizem apenas suas próprias informações. O foco principal está na proteção contra acessos externos, não na

separação de informações entre usuários internos do mesmo sistema. Esta lacuna é especialmente importante para o *MechTrack*.

QP3: Quais tecnologias *mobile* são utilizadas no desenvolvimento de aplicações para gestão de manutenção industrial?

React Native e *Flutter* destacam-se como os *frameworks* multiplataforma predominantes, respondendo juntos pela maior parcela dos projetos móveis profissionais (STACK OVERFLOW, 2024). O *React Native* se beneficia do ecossistema *JavaScript*, enquanto o *Flutter* é preferido em projetos que priorizam desempenho e controle de interface.

O desenvolvimento nativo (*Java/Kotlin* e *Swift/Objective-C*) permanece em aplicações que exigem maior integração com recursos específicos. *Frameworks* híbridos como o *Ionic* ainda aparecem em sistemas mais antigos, mas vêm sendo substituídos por tecnologias mais modernas.

QP4: Existem aplicações móveis específicas para o setor agrícola e agroindustrial que consideram conectividade intermitente?

Foi observado uma lacuna significativa de soluções especializadas para o setor agrícola e agroindustrial. A maioria dos estudos foca em contextos urbanos ou industriais com conectividade estável.

Nos poucos casos voltados ao setor agrícola, algumas abordagens incluem *cache* local de dados e sincronização automática, porém tratadas como recursos secundários, não como requisitos primários de *design*. A falta de soluções *offline-first* representa uma oportunidade de inovação.

QP5: Quais funcionalidades são consideradas essenciais em sistemas de gestão de atividades externas para equipes técnicas?

As funcionalidades mais comuns que encontramos incluem: registro de *check-in* e *check-out* com geolocalização, documentação fotográfica das atividades, assinatura digital de ordens de serviço, um histórico completo das tarefas realizadas e a geração de relatórios gerenciais.

Recursos como controle automático de quilometragem, cálculo de horas trabalhadas e notificações *push* para novas atribuições se destacam como diferenciais em soluções mais robustas. No entanto, é importante notar que

interfaces simplificadas para usuários com pouca familiaridade tecnológica são bastante raras, o que sugere que a maioria das soluções prioriza a variedade de funcionalidades em detrimento da usabilidade.

3.5 Análise de Mercado

Além da revisão da literatura científica, realizou-se uma pesquisa exploratória nas lojas de aplicativos para identificar soluções digitais voltadas à gestão de equipes externas. Foram encontrados três aplicativos relevantes: o *FieldControl*[2], focado na gestão de ordens de serviço e despacho de técnicos; o *Checkmob*[3], voltado a equipes de vendas externas e visitas a clientes; e o *PromoterK*[4], direcionado ao controle de jornada de promotores de venda.

Embora cada um atenda parcialmente às demandas de gestão em campo, nenhum se destina especificamente a pequenas equipes técnicas do setor de manutenção industrial. O *MechTrack* se diferencia por reunir controle de jornada com geolocalização, cálculo de quilometragem e registro de atividades com transcrição de áudio, funcionalidades orientadas às necessidades reais desse segmento.

3.6 Considerações Finais do Mapeamento Sistemático

Com base no mapeamento sistemático e na pesquisa exploratória realizada, conclui-se que, embora existam soluções para gestão de equipes externas, nenhuma delas atende simultaneamente às necessidades de equipes técnicas de manutenção industrial com controle de acesso diferenciado baseado em privacidade e operação *offline-first* como requisito primário de *design.*

Os aplicativos encontrados nas lojas como *AppStore*[5] e *PlayStore*[6], *FieldControl*, *Checkmob* e *PromoterK*, possuem focos distintos, voltados respectivamente a ordens de serviço, vendas externas e *merchandising*. Embora ofereçam algum nível de controle de acesso, a privacidade entre membros da própria equipe não é tratada como requisito central nessas soluções.

O *MechTrack* preenche esta lacuna ao combinar tecnologias consolidadas (*React Native, NestJS, RBAC*) para resolver o problema central identificado:

---

[2] https://fieldcontrol.com.br
[3] https://www.checkmob.com
[4] https://www.promoterk.com.br
[5] https://www.apple.com/br/app-store/
[6] https://play.google.com/store/games?hl=pt_BR&pli=1

conflitos organizacionais gerados por exposição inadequada de informações. Com funcionalidade *offline-first* e simplicidade operacional, o aplicativo preserva privacidade individual enquanto fornece visibilidade gerencial adequada.

O próximo capítulo apresenta a metodologia de desenvolvimento do *MechTrack*, detalhando as tecnologias utilizadas e processos adotados para transformar os requisitos identificados em solução funcional.

4. Metodologia

Este capítulo descreve os métodos, tecnologias e processos adotados no desenvolvimento do *MechTrack*, desde a identificação dos requisitos até a implementação e validação da solução.

4.1 Levantamento de Requisitos

O processo de levantamento de requisitos teve início com a realização de uma entrevista com um profissional atuante na área de manutenção agroindustrial. A entrevista é uma das técnicas mais eficazes para elicitação de requisitos, pois permite compreender o contexto real de uso do sistema e identificar necessidades que documentos formais não capturam (SOMMERVILLE, 2019).

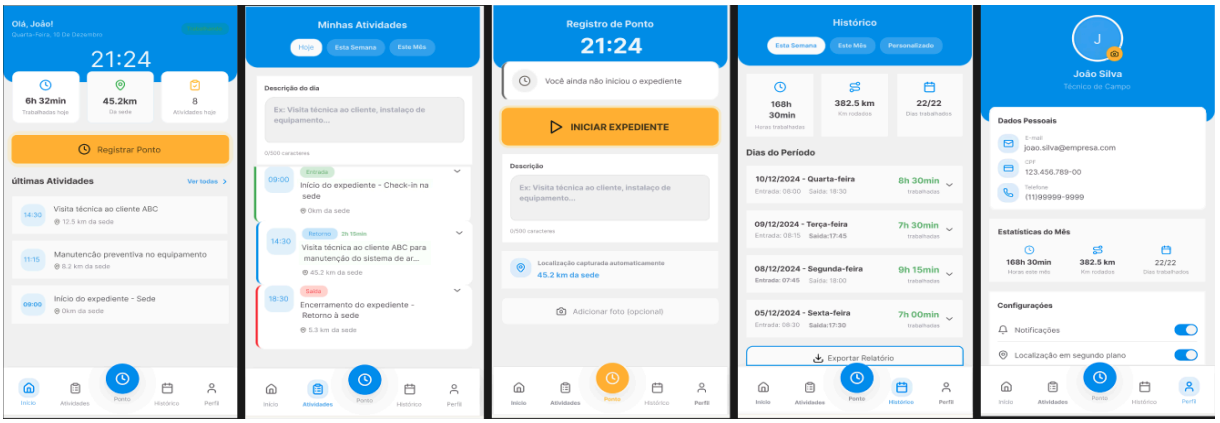
Durante esse contato, foram identificados os principais problemas enfrentados no gerenciamento de equipes técnicas externas, em especial o uso de planilhas compartilhadas como ferramenta de controle operacional. Segundo o profissional entrevistado, a exposição de informações sensíveis, como horários, quilometragem percorrida e salários entre os membros da equipe, gerava conflitos internos e comprometia o ambiente de trabalho.

A partir deste relato, foram definidos os requisitos funcionais do sistema, priorizando três aspectos centrais: o controle de acesso diferenciado por perfil de usuário, a funcionalidade *offline-first* para uso em áreas com conectividade instável, e uma interface com fluxos diretos e reduzidos, que minimize o número de interações necessárias para o registro de informações. Esse último requisito foi considerado importante devido ao perfil dos técnicos de campo, que necessitam registrar informações de forma rápida e direta durante suas atividades.

Como complemento à entrevista, foram elaborados esboços de telas no

Figma[7], conforme apresentado na Figura 1, contemplando as principais telas e fluxos de navegação do aplicativo. Esses protótipos serviram como referência visual durante o desenvolvimento e, vale ressaltar que, o produto final apresenta evoluções e adaptações em relação às telas inicialmente prototipadas, adicionando melhorias identificadas ao longo das etapas de implementação.

Figura 1 - Protótipos do *Figma*

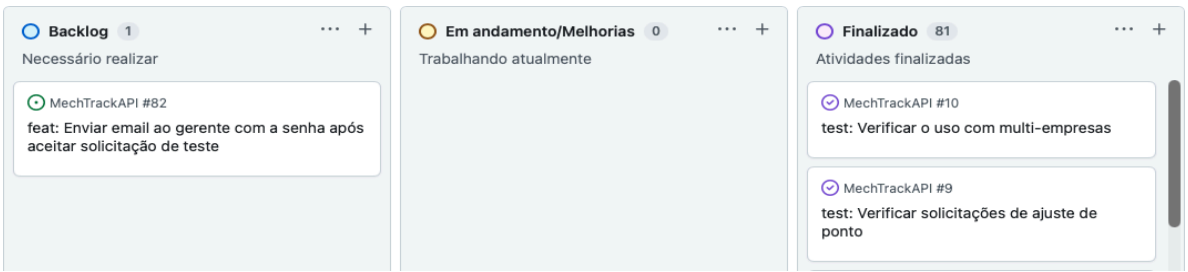


Fonte: O Autor, 2026

4.2 Metodologia de Desenvolvimento

O desenvolvimento do *MechTrack* foi conduzido adotando o método *Kanban* como metodologia ágil de gestão de projeto, implementado por meio do *GitHub Projects*[8], apresentado na Figura 2. Segundo Martins (2006), o *Kanban* organiza o fluxo de trabalho por meio de quadros visuais que indicam o status de cada tarefa, favorecendo a visibilidade e o controle das etapas de produção. Essa abordagem permitiu organizar a atividades em colunas de progresso, como "*Backlog*", "Em andamento/melhorias" e "Finalizado".

Figura 2 - Quadro *Kanban*



Fonte: O Autor, 2026

[7] https://www.figma.com
[8] https://github.com/projects

A utilização da metodologia ágil foi fundamental para o desenvolvimento do *MechTrack*, promovendo um fluxo de trabalho organizado e orientado a entregas contínuas. A abordagem incremental permitiu que cada funcionalidade fosse implementada, testada manualmente e integrada ao sistema antes do avanço para a próxima etapa, reduzindo o risco de retrabalho e favorecendo ajustes ao longo do processo. Dessa forma, a metodologia assegurou que o sistema evoluísse de forma consistente conforme as necessidades operacionais e de negócio identificadas durante o desenvolvimento.

4.3 Tecnologias Utilizadas

Considerando as necessidades do *software*, diversas tecnologias foram utilizadas, sendo elas o *Expo, React Native*, *Typescript* e *NestJS.* Essas ferramentas se destacam por sua capacidade de integração, flexibilidade e robustez, permitindo assim a criação de aplicações escaláveis e seguras.

4.3.1 *Frontend*

O *React Native, framework* desenvolvido pela Meta, foi adotado como base para o desenvolvimento do aplicativo, permitindo o compartilhamento de código entre as plataformas *iOS* e *Android* a partir de uma única base de código. O ambiente *Expo* foi adotado como plataforma de desenvolvimento e distribuição, oferecendo um conjunto de ferramentas que agiliza a configuração, disponibiliza *APIs* nativas pré-configuradas e facilita o processo de *build* e *deploy*. Isso permitiu que o desenvolvimento fosse iniciado de forma imediata, sem configuração manual do ambiente nativo, além de fornecer acesso direto à bibliotecas como *expo-location* para geolocalização e *expo-av* para gravação de áudio, recursos essenciais para as funcionalidades do *MechTrack*.

4.3.2 *Backend*

No *backend*, foi utilizado o *NestJS*, *framework* para *Node.js* construído sobre *TypeScript* que adota princípios de arquitetura modular, injeção de dependência e separação de responsabilidades. Conforme sua documentação oficial, o *NestJS* fornece uma arquitetura que favorece o desenvolvimento de aplicações altamente testáveis, escaláveis e de fácil manutenção (NESTJS, 2024). Como servidor *HTTP*, foi utilizado o *Fastify* em substituição ao *Express* padrão, por apresentar menor

overhead de processamento e maior desempenho em termos de latência e *throughput*, características relevantes para uma *API* que deve responder com agilidade a requisições de dispositivos móveis em campo.

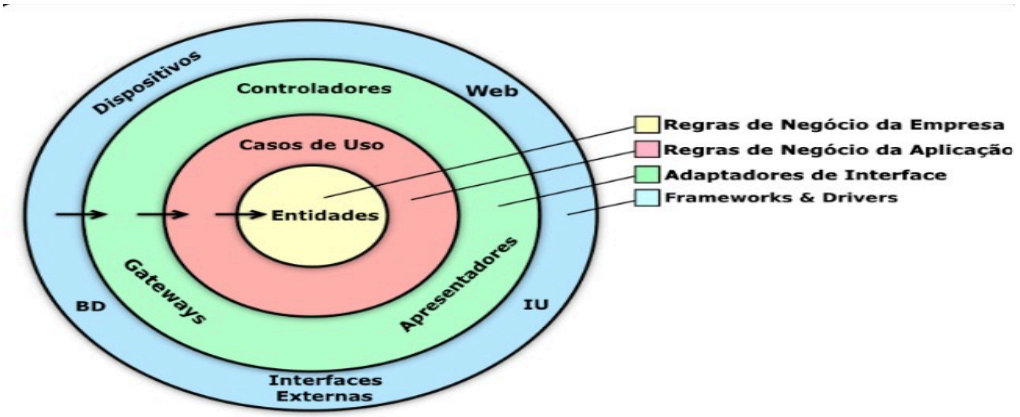
4.3.3 Banco de Dados

Para o armazenamento de dados foi utilizado o SGBD *PostgreSQL*, sistema gerenciador de banco de dados relacional de código aberto reconhecido por sua robustez, confiabilidade e conformidade com o padrão *SQL* (*Structured Query Language*). Sistemas que implementam as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade) garantem que transações sejam processadas de forma confiável, assegurando a integridade dos dados mesmo diante de falhas (ELMASRI; NAVATHE, 2019). O *PostgreSQL* atende plenamente a esse padrão e ainda oferece recursos avançados como suporte a tipos de dados complexos e extensibilidade por meio de extensões.

4.4 Arquitetura do Sistema

O *backend* foi estruturado seguindo os princípios da *Clean Architecture*, que organiza o código em camadas concêntricas com dependências apontando sempre para o interior (MARTIN, 2012), conforme apresentado na Figura 3. No *MechTrack*, essa estrutura foi implementada em três camadas: Domínio, concentrando entidades e regras de negócio sem dependência de *frameworks* externos; Aplicação, contendo os casos de uso que orquestram as operações do sistema; e Infraestrutura, abrigando as implementações concretas dos repositórios, *schemas* do *TypeORM*, *DTOs (Data Transfer Object)* e *controllers* do *NestJS*.

Figura 3 - Diagrama da *Clean Architecture*



Fonte: Martin (2012)

Essa separação garante que as regras de negócio não dependam de decisões de infraestrutura, como o banco de dados ou o *framework web* utilizado, tornando o sistema mais resiliente a mudanças tecnológicas e facilitando a manutenção ao longo do tempo.

4.5 Testes e Validação do Sistema

A validação do sistema foi realizada por meio de testes manuais e unitários conduzidos ao longo do desenvolvimento. A cada módulo implementado, foram executados cenários de uso representativos, verificando o comportamento da aplicação em condições normais e em situações de erro, como ausência de conexão com a *internet*, dados inválidos e tentativas de acesso não autorizado. Os testes foram realizados em dispositivos físicos com sistema *iOS* e *Android*, garantindo que o comportamento do aplicativo fosse avaliado em ambiente real de uso. Foram aplicados três tipos de testes:

- Testes unitários: implementados com o *framework Jest* para verificar de forma isolada a lógica dos principais casos de uso do *backend*, garantindo que cada regra de negócio se comportasse conforme o esperado independentemente das demais camadas da aplicação.
- Testes de integração manuais: realizados sem ferramentas automatizadas, consistiram em simulações de uso completo do aplicativo em ambiente local, cobrindo fluxos como registro de ponto, sincronização *offline*, ajuste de ponto e controle de acesso por perfil de usuário.
- Validação de requisições: realizada por meio do *Postman*, ferramenta para teste e documentação de *APIs,* com o objetivo de testar as chamadas à *API* e garantir a correta comunicação entre cliente e servidor, incluindo autenticação via *JWT* e respostas de erro esperadas.

5. Resultados e Discussões

Este capítulo apresenta os principais resultados obtidos com o desenvolvimento do *MechTrack.* São descritas as interfaces implementadas, os fluxos de uso do sistema e as funcionalidade técnicas que constituem os diferenciais da solução.

5.1 Interfaces do Sistema

O *MechTrack* foi desenvolvido com interfaces distintas para cada perfil de usuário. O controle de acesso é aplicado no *backend*, que rejeita requisições não autorizadas independentemente do cliente e o *frontend* complementa essa segurança exibindo apenas as telas e funcionalidades pertinentes ao perfil autenticado. A seguir, são apresentadas as principais telas implementadas.

A tela de *login*, ilustrada na Figura 4, é o ponto de entrada do sistema para todos os perfis. Após a autenticação, o redirecionamento é realizado automaticamente com base no perfil de usuário.

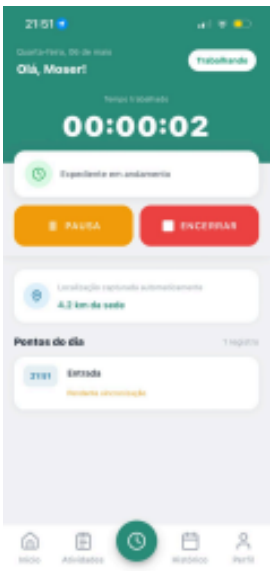
Figura 4 - Tela de *Login*



Fonte: O Autor, 2026

A interface do funcionário é centrada no registro de ponto. A tela principal, apresentada na Figura 5, exibe o *timer* do expediente em tempo real, os botões de ação para entrada, intervalo e encerramento, e o status atual do dia de trabalho. O *design* foi desenvolvido com foco na simplicidade, considerando o perfil de usuários com menor familiaridade tecnológica que atuam em campo.

Figura 5 - Tela de registro de ponto



Fonte: O Autor, 2026

A janela de atividade, exibida na Figura 6 permite ao funcionário registrar descrições das tarefas realizadas ao longo do expediente. A funcionalidade de transcrição de áudio está integrada diretamente a esse campo, possibilitando que o usuário dite a descrição em vez de digitá-la, o que agiliza o registro em situações de campo.

Figura 6 - Janela de registro de atividade com transcrição de áudio

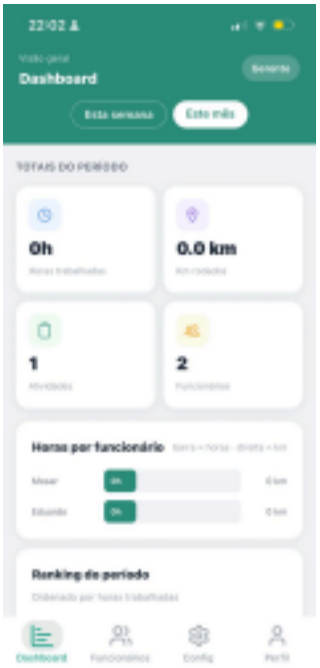


Fonte: O Autor, 2026

A interface de gerente é orientada à visão gerencial da equipe. O *dashboard*, apresentado na Figura 7, consolida as informações dos funcionários no período selecionado, exibindo totais de horas trabalhadas, quilometragem percorrida e atividades registradas, melhorando a atividade gerencial dos usuários com este perfil

de acesso. Além disso, é exibido um *ranking* dos funcionários de acordo com o total de horas trabalhadas no período, permitindo ao gerente identificar rapidamente o desempenho individual de cada colaborador. Ao selecionar um funcionário, é possível acessar o detalhamento completo do seu expediente, incluindo pontos registrados, as atividades descritas e a quilometragem percorrida, além de gerar e compartilhar um relatório em PDF com essas informações.

Figura 7 - Dashboard do Gerente



Fonte: O Autor, 2026

O painel do administrador, ilustrado na Figura 8, centraliza a gestão de empresas e usuários da plataforma. A partir dessa interface, o administrador pode visualizar e gerenciar todas as empresas cadastradas, criar e editar usuários vinculados a cada empresa, e analisar as solicitações de teste gratuito submetidas por novos clientes. Ao aprovar uma solicitação, o administrador cadastra a empresa e o usuário gerente responsável, que recebe automaticamente as credenciais de acesso. Além disso, o painel consolida os chamados abertos pelos gerentes das equipes, permitindo que o administrador acompanhe e responda cada solicitação. Os chamados ainda não atendidos são exibidos com o status “Pendente”, sinalizando ao solicitante que a demanda está em análise.

Figura 8 - Painel do Administrador



Fonte: O Autor, 2026

Por fim, as funcionalidades disponíveis para cada perfil de usuário do sistema são apresentadas na Figura 9, elaborada com o Keynote[9]. O perfil Funcionário permite registrar pontos, descrever atividades, solicitar ajuste de ponto, abrir chamados ao gerente e verificar seu histórico de expedientes. O Gerente e Supervisor acessam o dashboard da equipe, aprovam ajustes, resolvem chamados, exportam relatórios e gerenciam funcionários e a sede. O Administrador gerencia todas as empresas da plataforma, analisa solicitações de teste e responde chamados dos gerentes.

Figura 9 - Perfis de Usuário e Funcionalidades



Fonte: O Autor, 2026

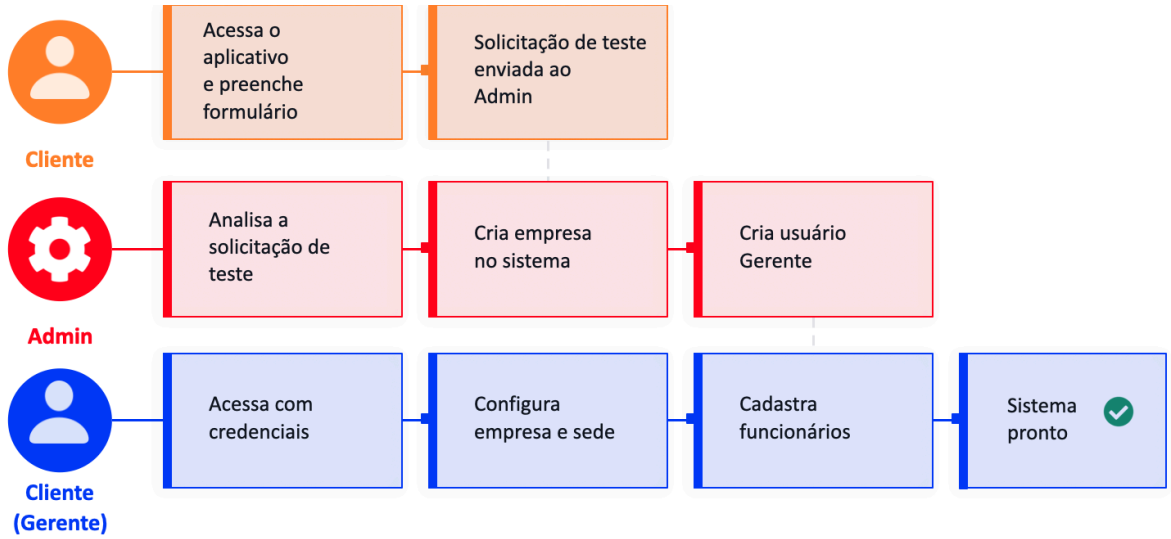
[9] https://www.apple.com/br/keynote/

5.2 Fluxos de Uso

Para ilustrar o funcionamento do sistema sob a perspectiva de cada perfil, são descritos a seguir os fluxos de uso correspondentes ao processo de *onboarding*, ao expediente do funcionário e à gestão realizada pelo gerente.

O fluxo de *onboarding* contempla três atores: o cliente interessado, o administrador da plataforma e o gerente recém cadastrado. O processo tem início quando o cliente acessa o formulário de solicitação de teste no aplicativo e submete seus dados. O administrador recebe a solicitação, analisa as informações, cria a empresa na plataforma e cadastra o usuário gerente responsável pela conta. A partir desse momento, o gerente recebe suas credenciais de acesso e inicia a configuração da empresa, cadastrando os dados da sede, as coordenadas geográficas e os funcionários da equipe. Ao concluir essa etapa, o sistema está pronto para uso, conforme apresentado na Figura 10, elaborada no *Keynote*.

Figura 10 - Fluxo de Onboarding: do interesse do sistema ao funcionamento

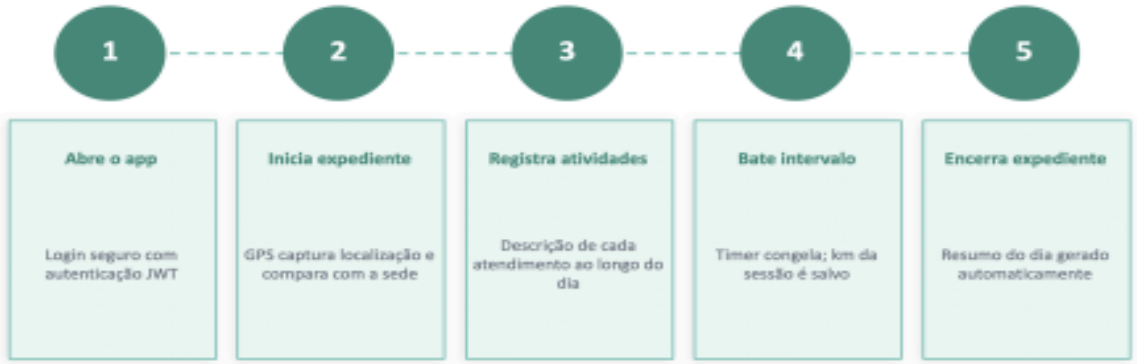


Fonte: O Autor, 2026

O fluxo do expediente do funcionário representa o ciclo completo de um dia de trabalho. Após o *login* com autenticação *JWT*, o funcionário registra a entrada, momento em que o sistema captura as coordenadas GPS e verifica se o dispositivo está dentro do raio da sede. O *timer* do expediente é iniciado e a quilometragem começa a ser acumulada para deslocamentos fora da sede.

Ao longo do dia, o funcionário registra as atividades realizadas, podendo utilizar transcrição de áudio. Quando necessário, pode registrar o intervalo, que congela o *timer* e salva a quilometragem da sessão, ao retornar, o *timer* é retomado do ponto em que foi pausado. Ao final, o expediente é encerrado com um resumo do dia gerado automaticamente. Esse fluxo está representado na Figura 11, elaborada no *Keynote*.

Figura 11 - Fluxo do Expediente do Funcionário



Fonte: O Autor, 2026

O gerente, por sua vez, acompanha a equipe pelo *dashboard*, que consolida as informações de todos os funcionários no período selecionado. A partir dessa visão, pode-se acessar o detalhamento individual de cada colaborador, aprovar ou recusar solicitações de ajuste ponto, resolver chamados abertos pelos funcionários e exportar relatórios em PDF para fins de controle operacional.

5.3 Funcionalidades Técnicas Relevantes

Esta seção descreve as funcionalidades técnicas que representam os principais diferenciais do *MechTrack*: a operação *offline-first*, o cálculo de quilometragem por geolocalização e a transcrição de áudio para texto.

5.3.1 Offline-First e Sincronização

O *MechTrack* foi projetado com uma estratégia *offline-first*, o que significa que todas as ações do usuário, como registro de ponto, atividades e quilometragem, são armazenadas localmente no dispositivo por meio do *AsyncStorage* antes de qualquer comunicação com o servidor. Essa abordagem garante que o sistema funcione plenamente em áreas com conectividade instável ou ausente, como zonas rurais e ambientes industriais com sinal limitado.

A sincronização com o *backend* ocorre automaticamente em dois momentos: quando o aplicativo retorna ao primeiro plano, e quando a conectividade de rede é restabelecida. A biblioteca *NetInfo* monitora continuamente o estado da conexão, e cada registro possui um campo de controle que indica se já foi sincronizado, evitando duplicações e garantindo a integridade dos dados.

5.3.2 Geolocalização e Cálculo de Quilometragem

O rastreamento de deslocamento é realizado por meio da captura periódica de coordenadas GPS utilizando a biblioteca *expo-location*. As coordenadas são coletadas a cada trinta segundos durante o expediente ativo, e a distância entre pontos consecutivos é calculada pela fórmula de *Haversine,* descrita na Figura 12, que determina a distância em linha reta entre dois pontos na superfície esférica da Terra a partir de suas latitudes e longitudes, conforme formulado por Sinnott (1984).

A quilometragem é acumulada de sessão a sessão ao longo do expediente. Funcionários que se encontram dentro do raio configurado da sede não têm o deslocamento contabilizado, uma vez que não há custo de deslocamento nessa situação. O raio é definido pelo gerente e verificado a cada captura de coordenada pela mesma fórmula.

Figura 12 - Fórmula de Haversine

$$d = 2r\ arcsine\left(\sqrt{sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right) + cos\varphi_1 . cos\varphi_2 . sin^2\left(\frac{\omega_2-\omega_1}{2}\right)}\right)$$

d — Distância entre os pontos
r — Raio da Terra (~6.371 km)
ϕ — Latitudes dos pontos
ω — Longitudes dos pontos

Fonte: O Autor, 2026

5.3.3 Transcrição de Áudio

Para facilitar o registro de atividades em campo, o *MechTrack* oferece transcrição de áudio para texto integrado à tela de registro de atividades. O usuário grava um relato em voz sobre o serviço realizado, e o áudio é enviado ao *backend*, que o encaminha para a *API* do *Groq*[10], plataforma de inferência de inteligência

[10] https://groq.com/

artificial com alta velocidade de processamento e camada gratuita adequada ao contexto do projeto, utilizando o modelo *Whisper Large V3 Turbo,* especializado em transcrição de fala em múltiplos idiomas. O texto transcrito é retornado para o aplicativo, eliminando a necessidade de digitação manual em campo.

5.4 Modelagem do Banco de Dados

A modelagem do banco de dados foi desenvolvida com base nos requisitos funcionais do sistema, utilizando o *PostgreSQL* e o *TypeORM* como camada de mapeamento objeto-relacional. As entidades foram projetadas para refletir os fluxos reais de trabalho das equipes técnicas em campo, priorizando a rastreabilidade das ações e a separação de responsabilidades entre os perfis de usuário. As principais entidades do sistema são:

- **User**: representa os usuários do sistema, podendo assumir os perfis de Administrador, Gerente, Supervisor ou Funcionário. Cada usuário está vinculado a uma empresa e possui um papel (*role*) que determina suas permissões de acesso.
- **Company**: representa as empresas cadastradas no sistema. Cada empresa possui um endereço com coordenadas geográficas e um raio configurável.
- **DayOfWork**: representa o dia de trabalho de um funcionário, armazenando o status do expediente (TRABALHANDO, INTERVALO, ENCERRADO), o total de segundos trabalhados e a quilometragem percorrida no dia.
- **TimeRecord**: registra cada ponto batido pelo funcionário (ENTRADA, INTERVALO, RETORNO, SAÍDA), com *timestamp*, coordenadas geográficas, quilometragem da sessão e indicação de presença na sede.
- **Activity**: registra as atividades realizadas pelo funcionário durante o expediente, vinculadas ao dia de trabalho, com descrição textual e suporte a URL (*Uniform Resource Locator*) de áudio transcrito.
- **TrialRequest**: representa as solicitações de teste gratuito enviadas por potenciais clientes, contendo os dados da empresa e do responsável.

A separação entre DayOfWork, TimeRecord e Activity foi uma decisão central de *design*, permitindo que o dia de trabalho acumule os totais (horas e km) de forma incremental a cada ponto batido, enquanto as atividades são registradas de forma independente e associadas ao dia, sem interferir no cálculo da jornada.

5.5 Análise dos Resultados

O desenvolvimento do *MechTrack* resultou em uma solução funcional que atende aos objetivos estabelecidos no início do projeto, oferecendo controle de acesso segmentado que elimina a exposição indevida de informações sensíveis entre os membros da equipe, fundamentado nos princípios do RBAC (FERRAIOLO; KUHN, 2020).

A funcionalidade *offline-first* permitiu o registro das atividades independentemente da disponibilidade de conexão, com sincronização automática ao restabelecer a rede. O cálculo de quilometragem via fórmula de *Haversine* permitiu a rastreabilidade dos deslocamentos de forma automatizada, e a transcrição de áudio reduziu o número de interações necessárias para o registro de atividades, sendo especialmente relevante para usuários em campo.

Os resultados obtidos indicam que o *MechTrack* preenche a lacuna identificada no mapeamento sistemático, combinando controle de acesso diferenciado, operação *offline* e simplicidade de uso em um único sistema, aspectos que serão retomados nas considerações finais.

6. Considerações Finais

Este trabalho teve como objetivo o desenvolvimento do *MechTrack*, um aplicativo móvel para gestão de equipes técnicas em serviços externos, motivado pela identificação de um problema real: o uso de planilhas compartilhadas como ferramenta de controle operacional em empresas de manutenção industrial e agroindústrias, que gerava conflitos internos pela exposição indevida de informações sensíveis entre os membros da equipe.

A solução desenvolvida atende aos objetivos propostos na introdução. O sistema de autenticação com perfis diferenciados, como funcionário, supervisor, gerente e administrador, implementado com base nos princípios do RBAC, assegurou que cada usuário acesse exclusivamente as informações pertinentes à sua função, resolvendo o problema central identificado na entrevista com o profissional da área. A funcionalidade *offline-first*, com sincronização automática ao restabelecer a conexão, garantiu a operação do sistema em áreas com conectividade instável, requisito essencial para o contexto agroindustrial. O cálculo

de quilometragem por geolocalização, implementado com a fórmula de *Haversine,* automatizou o rastreamento dos deslocamentos de campo, e a interface simplificada atendeu ao perfil de usuários com menor familiaridade tecnológica.

Do ponto de vista técnico, a adoção da *Clean Architecture* no *backend* proporcionou uma base organizada, com separação clara entre regras de negócio e infraestrutura, o que favorece a manutenção e a evolução do sistema ao longo do tempo. A combinação de *React Native* com *Expo* no *frontend* permitiu o desenvolvimento de uma solução multiplataforma com acesso a recursos nativos do dispositivo, como GPS e microfone, sem comprometer a experiência do usuário.

O mapeamento sistemático realizado no Estado da Arte identificou soluções existentes para gestão de equipes externas, como *Field Control*, *Checkmob* e *PromoterK*, porém com focos distintos, ordens de serviço, vendas e merchandising, sem atender de forma específica às necessidades de equipes técnicas de manutenção industrial com controle de acesso diferenciado por privacidade interna e operação *offline-first*. O *MechTrack* preenche essa lacuna ao integrar essas características de forma coesa, representando uma contribuição prática para o setor.

Como limitações, destaca-se a ausência de testes automatizados e de validação formal com usuários finais, aspectos que constituem oportunidades para trabalhos futuros. Recomenda-se, para versões subsequentes, a implementação de testes unitários e de integração aproveitando o isolamento proporcionado pela arquitetura adotada, a condução de testes de usabilidade com técnicos de campo para validação das interfaces, e a expansão de funcionalidades como notificações *push*, relatórios avançados e suporte a múltiplos idiomas, visando ampliar o alcance da solução para outros mercados e segmentos.

Por fim, o desenvolvimento do *MechTrack* reafirma o potencial da tecnologia móvel para modernizar processos operacionais que ainda dependem de métodos manuais e inadequados. A solução demonstra que é possível construir ferramentas digitais práticas, seguras e acessíveis para ambientes de trabalho exigentes, contribuindo para a eficiência das equipes, a privacidade dos colaboradores e a qualidade da gestão nas empresas do setor.

Referências

ABNT. **NBR 5462: Confiabilidade e mantenabilidade.** Rio de Janeiro: ABNT, 2021.

TIINSIDE. Disponível em: https://tiinside.com.br/04/11/2025/87-das-empresas-tem-sistemas-integrados-mas-635-sofrem-com-dados-conflitantes-revela-pesquisa/. Acesso em: 14 jun. 2026

MCKINSEY & COMPANY. **Sua equipe em campo é de fato lean?** Disponível em: https://www.mckinsey.com/capabilities/operations/our-insights/how-lean-is-your-field-force-really. Acesso em: 14 jun. 2026.

ADAMS, J. S. **Inequity in social exchange**. *Advances in Experimental Social Psychology,* v. 2, p. 267–299, 1965.

BIONI, B. R. **Proteção de dados pessoais: a função e os limites do consentimento.** 2. ed. Rio de Janeiro: Forense, 2021.

BRASIL. **Lei nº 13.709, de 14 de agosto de 2018.** Lei Geral de Proteção de Dados Pessoais (LGPD). Brasília, DF: Presidência da República, 2018.

EISENMAN, B. **Learning React Native: Building Native Mobile Apps with JavaScript.** 3rd ed. Sebastopol: O'Reilly Media, 2021.

FACEBOOK. **React Native Documentation.** Menlo Park: Meta Platforms, 2023. Disponível em: https://reactnative.dev/docs/getting-started. Acesso em: 1 nov. 2025. FERRAIOLO, D. F.; KUHN, D. R. **Role-Based Access Control.** 2nd ed. Boston: Artech House, 2020.

JONES, M.; BRADLEY, J. **JSON Web Token (JWT).** RFC 7519. IETF, maio 2021.

KAMIL, M. Y. **NestJS: A Progressive Node.js Framework.** New York: Apress, 2023. KARDEC, A.; NASCIF, J. **Manutenção: função estratégica.** 5. ed. Rio de Janeiro: Qualitymark, 2022.

MASSE, M. **REST API Design Rulebook.** Sebastopol: O'Reilly Media, 2021.

NESTJS. **NestJS Official Documentation.** 2024. Disponível em: https://docs.nestjs.com. Acesso em: 1 nov. 2025.

OWASP. **Mobile Application Security Verification Standard (MASVS).** OWASP Foundation, 2023. Disponível em: https://owasp.org/www-project-mobile-app-security. Acesso em: 1 nov. 2025.

PETERSON, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: **Proceedings of the International Conference**

on Evaluation and Assessment in Software Engineering, p. 68–77, 2008.
REACT NATIVE COMMUNITY. **@react-native-community/geolocation.** 2024. Disponível
em: https://github.com/react-native-community/geolocation. Acesso em: 1 nov. 2025. ROBBINS, S. P.; JUDGE, T. A. **Organizational behavior.** 19th ed. Harlow: Pearson Education, 2023.
SINNOTT, R. W. Virtues of the haversine. *Sky and Telescope,* v. 68, n. 2, p. 159, 1984.
XENOS, H. G. P. **Gerenciando a manutenção produtiva.** 3. ed. São Paulo: INDG, 2020.
REVISTA O EMPREITEIRO. Disponível em: https://revistaoe.com.br/ranking-engenharia-brasileira-2025/. Acesso em: 14 jun. 2026
REIGADO, J. M. A. AGROPT: **Plataforma Tecnológica Soluções para Vinhas e Olivais**. 2019. Dissertação (Mestrado) – Instituto Politécnico da Guarda, Guarda, 2019.
SILVA, M. H. B.; BASSO, A. F. L.; RUELLAS, L. C. **Desenvolvimento de um aplicativo mobile para o controle e gestão do agronegócio de hortifrúti**. In: CONGRESSO BRASILEIRO DE AGROINFORMÁTICA, 12., 2019, [Local]. Anais [...]. [Local]: [Editora], 2019.
MARTINS, G; LAUGENI, F. P. **Administração da produção**–2.ed.rev.aum. E atual. São Paulo: Saraiva, 2006.
MARTIN, Robert C. **The Clean Architecture.** ***Clean Coder Blog***, 13 ago. 2012. Disponível em: https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html. Acesso em: 4 de maio, 2026.. Acesso em: 04 mai. 2026.
SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2019. Acesso em: 4 de maio, 2026.
ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 7. ed. São Paulo: Pearson, 2019. Acesso em: 4 de maio, 2026.
STACK OVERFLOW. Developer Survey 2024. Disponível em: https://survey.stackoverflow.co/2024/. Acesso em: 14 jun. 2026.