

Desenvolvimento de um Sistema Web para Gerenciamento de Ordens de Serviço em Assistências Técnicas

*Caio Eduardo Nunes da Costa*¹; *Pedro Henrique Teles*²; *Arthur Rezende Morel*³; *João Vitor Rosa de Souza*⁴; *Matheus de Sousa Valente*⁵

Universidade de Uberaba

ce.costa2015@gmail.com; mateus.valente@uniube.br

Resumo

O presente trabalho apresenta o desenvolvimento de um sistema web para gerenciamento de Ordens de Serviço (OS), voltado para micro e pequenas empresas de assistência técnica. O projeto surgiu da necessidade de oferecer uma solução acessível para empresas que ainda utilizam processos manuais ou sistemas desatualizados para controlar atendimentos, clientes e equipamentos. O sistema foi desenvolvido utilizando as tecnologias Next.js, React, TypeScript e SQLite, adotando uma arquitetura full-stack monolítica que integra interface, lógica de negócio e banco de dados em um único projeto. Entre as funcionalidades implementadas estão o cadastro de clientes, cadastro de ordens de serviço, acompanhamento de status, listagem e gerenciamento de atendimentos. A estrutura do sistema foi planejada para proporcionar simplicidade de uso, facilidade de implantação e baixo custo operacional. Durante o desenvolvimento, houve validação contínua junto a um técnico atuante no setor de assistência técnica, garantindo que as funcionalidades atendessem às necessidades reais do segmento. Os resultados demonstraram que a solução desenvolvida contribui para a organização dos processos, redução da perda de informações e melhoria da gestão operacional. Conclui-se que o sistema atende aos objetivos propostos, oferecendo uma ferramenta prática, funcional e adequada à realidade das micro e pequenas empresas.

Palavras-chave: ordem de serviço; sistema web; desenvolvimento front-end; desenvolvimento back-end; Next.js; React; SQLite.

1 INTRODUÇÃO

Com a constante evolução tecnológica, a maneira como as empresas organizam as suas operações diárias teve um avanço significativo, impulsionado pela presença de novas tecnologias e softwares atualizados. Grandes corporações desenvolvem seus próprios sistemas para gerenciar tais tarefas; no entanto, para micro e pequenas empresas, existem barreiras limitantes, como o alto custo de licenciamento e a falta de uma infraestrutura de rede robusta. Especialmente no setor de prestação de serviços e assistência técnica, estas empresas enfrentam severos desafios para padronizar seus processos. Muitas ainda dependem de registros impressos e pastas físicas para armazenar informações, ou acabam utilizando sistemas datados, que frequentemente são complexos e não recebem atualizações compatíveis com as necessidades operacionais atuais (Picoto, 2021).

A dependência de processos manuais leva a uma desorganização e despadronização estrutural que ameaça a própria sobrevivência. Este impacto pode ser evidenciado por dados levantados pelo Instituto Brasileiro de Geografia e Estatística (IBGE, 2022), onde cerca de 20% das empresas encerram suas atividades ainda no primeiro ano e cerca de 48% em até três anos. Grande parte deste problema está atrelado à falta de controle de gestão operacional nas empresas.

Diante deste contexto, surge a necessidade de desenvolver um sistema para o controle de Ordem de Serviço (OS) que permita ao usuário criar as ordens de serviço, listar, editar e cadastrar clientes (Sauter, 1996). Este sistema visa ser acessível e prático, sendo idealizado como um sistema web, cujo objetivo é melhorar a gestão

das operações realizadas no dia a dia, otimizando o fluxo de atendimento, reduzindo a perda de informações críticas e provendo um sistema organizado.

2 REFERENCIAL TEÓRICO

2.1 Gestão de Processos

Utilizar ferramentas tecnológicas para automatização de processos contribui para a organização das atividades empresariais, permitindo maior controle das operações e melhor tomada de decisão (Sebrae, 2024).

2.2 Desenvolvimento Web com Next.js e React

Next.js é um framework React utilizado para construir aplicações web full-stack, unificando o servidor de API e a interface sem a necessidade de configurar projetos distintos (Next.js, s.d.).

React é uma ferramenta utilizada para a criação de interfaces em desenvolvimento web, facilitando o desenvolvimento de páginas dinâmicas, podendo utilizar componentes reutilizáveis no projeto (React, s.d.).

2.3 Uso do TypeScript

TypeScript é um superconjunto de linguagem que adiciona tipagem estática ao JavaScript, permitindo definir os tipos de dados em variáveis, funções e objetos. Com ele, é possível identificar erros de sintaxe e lógica antes mesmo de o código ser executado, além de oferecer suporte robusto à Programação Orientada a Objetos (POO) (Microsoft, s.d.).

2.4 Uso do SQLite

SQLite é um sistema de gerenciamento de banco de dados relacional, de código aberto, sendo um banco de dados SQL incorporado, não dependendo de um servidor separado, sendo contido em um único arquivo no dispositivo que está o executando (Sqlite, s.d.).

2.5 Arquitetura Monolítica

Essa arquitetura segue um padrão de desenvolvimento de software onde todas as partes da aplicação são desenvolvidas em único projeto. Ideal para sistemas de microempresas, tendo menor complexidade, fácil implantação e sem a necessidade de um servidor externo (Amazon Web Services, 2026).

3 METODOLOGIAS E DESCRIÇÃO DO SISTEMA

3.1 Visão Geral

O projeto foi desenvolvido para ser um sistema web com foco principal em uso no desktop, utilizando Next.js como principal framework da aplicação, React como a biblioteca de interface, TypeScript como a linguagem principal e SQLite, responsável por modelar e relacionar todos os dados, visando agregar acessibilidade e praticidade. O desenvolvimento do sistema foi conduzido utilizando uma abordagem incremental, inspirada em princípios de metodologias ágeis. As funcionalidades foram implementadas em pequenas etapas, permitindo testes e validações durante o processo. Os requisitos foram levantados e definidos com base na análise do problema e nas necessidades identificadas no contexto das assistências técnicas

3.2 Arquitetura do Sistema

O sistema foi desenvolvido com base em uma arquitetura full-stack monolítica, viabilizada pelo modelo App Router do Next.js. Nessa abordagem, o mesmo projeto concentra tanto a interface quanto o backend, sem necessitar de um servidor de aplicação separado. Ele adota uma separação de camadas bem definidas: uma camada de apresentação, composta pelos componentes React responsáveis pela interface visual; a camada de API, formada pelas rotas HTTP que recebem e processam as requisições; e a camada de dados, responsável pela conexão e operações no banco de dados.

A comunicação entre interface e banco de dados ocorre por meio de rotas internas, que executam as operações no banco via biblioteca better-sqlite3. Essa biblioteca permite acesso direto ao SQLite, sem a necessidade de um servidor de banco de dados em execução, simplificando a implantação do sistema em um ambiente desktop.

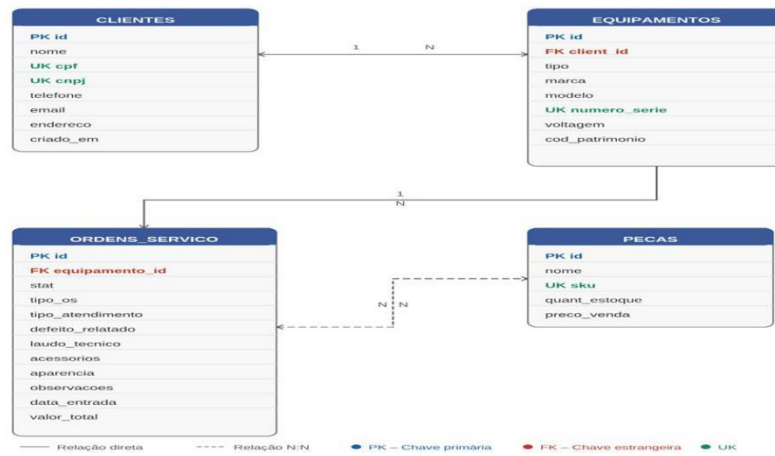
3.3 Tecnologias Utilizadas

As tecnologias foram selecionadas com base no critério de minimizar a complexidade de configuração e implantação. O Next.js foi adotado como framework principal por permitir o desenvolvimento full-stack em um único projeto, unificando o servidor de API e a interface, sem a necessidade de configurar dois projetos distintos. React foi utilizado como biblioteca de construção de toda a interface, dispensando bibliotecas externas adicionais. O TypeScript foi empregado como linguagem principal, oferecendo tipagem estática, reduzindo erros em tempo de desenvolvimento e melhorando a manutenibilidade do código. O SQLite foi utilizado por ser um banco de dados embutido, sem necessitar de instalação ou configuração de servidor, sendo ideal para aplicação desktop de uso local, o que constitui o foco do sistema.

3.4 Banco de Dados

O banco de dados foi estruturado em quatro entidades relacionais: Clientes, Equipamentos, Ordens de Serviços e Peças. O esquema não necessita de configuração manual, sendo executado automaticamente ao iniciar o sistema. A entidade Clientes armazena dados de identificação e contato, a entidade Equipamentos está vinculada a um cliente por chave estrangeira e registra todas as informações técnicas do aparelho. Ordens de Serviço representa o núcleo do sistema, vinculando um equipamento a um atendimento e armazenando as informações pertinentes. Considerando que o sistema armazena dados pessoais dos cliente, como nome, CPF, endereço, telefone e e-mail, o desenvolvimento levou em consideração os princípios estabelecidos pela Lei Geral de Proteção de Dados(LGPD - Lei nº 13.709/2018). As informações cadastradas são utilizadas exclusivamente para a execução dos serviços de assistência técnica, permanecendo armazenadas localmente no banco de dados, reduzindo a exposição em ambientes externos. Além disso, recomenda-se que a utilização do sistema seja realizada apenas por pessoas autorizadas, visando preservar a confidencialidade e a integridade das informações. diagrama entidade-relacionamento é apresentado na Figura 1.

Figura 1 – Diagrama Entidade-Relacionamento do sistema

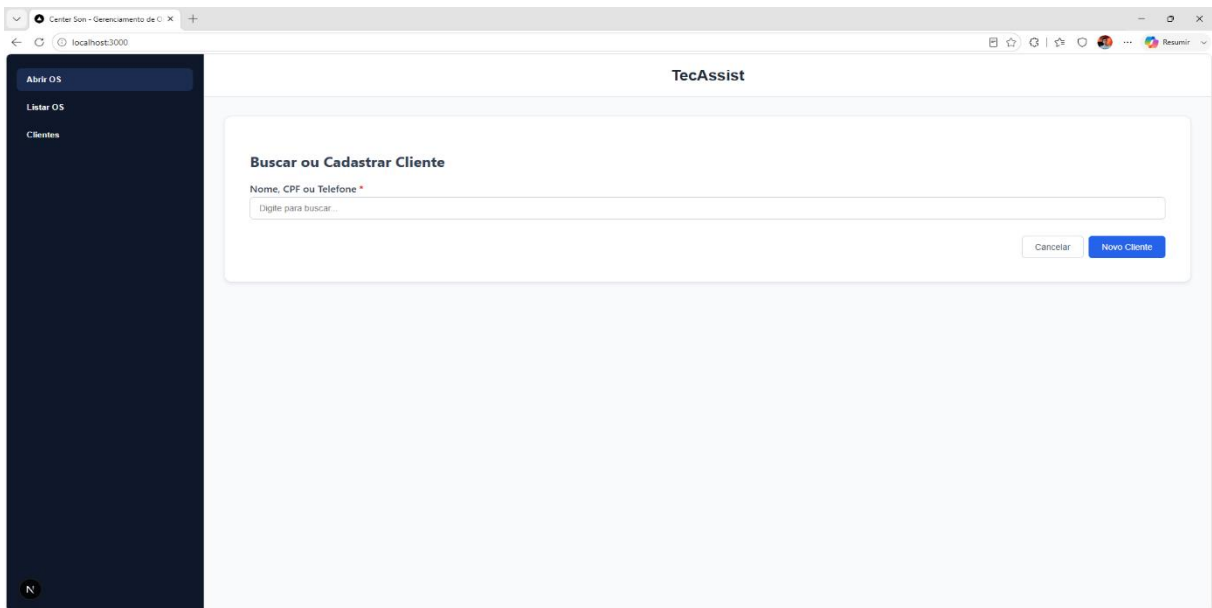


Fonte: Os autores (2026).

3.5 Interface e Usabilidade

O sistema possui uma interface limpa e minimalista, focando na simplicidade e acessibilidade, de modo que qualquer usuário, mesmo sem conhecimentos técnicos avançados, possa utilizá-lo (Rosa, 2025). A estrutura visual conta com uma barra de navegação lateral fixa (sidebar), organizando o fluxo de trabalho em três módulos principais: Criar OS, Listar OS e Cadastrar Clientes. Dentro dos módulos de cadastro, as telas apresentam formulários otimizados e validação de front-end, garantindo a integridade das informações antes do envio ao banco de dados. Além disso, a navegação entre os menus ocorre de maneira fluida, sem a necessidade de recarregar a página toda vez que uma nova opção é acionada, conforme apresentado na Figura 2.

Figura 2 – Interface do Sistema



Fonte: Os autores (2026).

3.6 Funcionalidades do Sistema

O sistema apresenta os seguintes recursos: cadastro de clientes, cadastro de OS, listagem, edição, exclusão e acompanhamento de status (em andamento, finalizada ou aguardando parecer técnico).

3.6.1 Cadastro de Clientes

Esta funcionalidade possibilita o cadastro de clientes por meio da inserção de dados de contato essenciais, como nome, CPF, endereço, número de telefone e e-mail. Visando uma melhor fluidez no atendimento é obrigatório apenas o

preenchimento do nome e do CPF. Essa validação garante que o sistema tenha as informações mínimas necessárias para identificar corretamente o proprietário do equipamento. O módulo de cadastro de cliente é apresentado na Figura 3.

Figura 3 – Cadastro de Cliente

A imagem mostra uma interface web do sistema TecAssist. No topo, há uma barra de navegação com o nome do sistema. À esquerda, um menu lateral contém opções como 'Abrir OS', 'Listar OS' e 'Clientes'. O formulário principal, intitulado 'Buscar ou Cadastrar Cliente', possui campos de entrada para 'Nome *', 'CPF *', 'Telefone', 'E-mail' e 'Endereço'. Abaixo dos campos, há dois botões: 'Voltar' e 'Salvar Cliente'.

Fonte: Os autores (2026).

3.6.2 Cadastro de OS

O cadastro de OS constitui o núcleo central do sistema, permitindo registrar a entrada de novos equipamentos por meio do preenchimento de dados técnicos essenciais: marca, modelo, número de série do aparelho, tipo de atendimento (Presencial, Domicílio ou Remoto), tipo de OS (Orçamento ou Garantia), relato do defeito, estado de conservação e status inicial. O formulário conta também com

campos extras para melhor registro de informações, como acessórios e itens, patrimônio e observações. Embora o sistema permita o registro de atendimentos presenças, remotos e em domicílio, sua utilização foi projetado prioritariamente para computadores instalados na assistência técnica. Em atendimentos externos, as informações podem ser registradas pelo técnico durante a visita e posteriormente inseridas no sistema. A Figura 4 apresenta a tela responsável pelo cadastro das ordens de serviço.

Figura 4 – Cadastro de OS

The screenshot displays the TecAssist web application interface. On the left, a dark sidebar contains navigation options: 'Abrir OS', 'Listar OS', and 'Clientes'. The main content area is titled 'TecAssist' and features a form for creating a new service order. At the top of the form, there is a 'CLIENTE' section with fields for Name (Arthur Rezende Morel), Telephone (34996962263), E-mail (arthurmorel2003@gmail.com), and Address (Rua Jurandir Cordeiro 451). Below this, the 'Dados da OS' section includes several dropdown menus for 'Status' (set to 'Aguardando avaliação'), 'Tipo de Atendimento', and 'Tipo de OS'. There are also text input fields for 'Equipamento / Aparelho', 'Marca', 'Modelo', 'Nº de Série', and 'Código / Patrimônio', each with a small example text below it.

Fonte: Os autores (2026).

3.6.3 Listagem de OS

O sistema possibilita acompanhar o andamento das Ordens de Serviço (OS) ativas por meio de uma interface limpa, projetada para facilitar o acesso rápido às informações. A tela organiza visualmente o fluxo dos atendimentos, permitindo ao usuário monitorar a situação das requisições de acordo com cada status operacional.

Além disso, a ferramenta conta com um recurso de busca por cliente, equipamento ou número da OS, otimizando a localização de registros específicos e garantindo agilidade na verificação do progresso da manutenção. A funcionalidade de acompanhamento e listagem das ordens de serviço pode ser visualizada na Figura 5.

Figura 5 – Listagem de OS

Nº OS	Cliente	Situação	Tipo de OS	Valor	Entrada	Equipamento	Marca	Modelo	Nº Série	Ações
12	Ricardo José Viana	Aguardando, autorização orçamento	Orçamento	R\$680,00	15/06/2026	TELEVISOR	PHILCO	PTV65G3BGT5SBL	8638304638634D	Editar Imprimir
11	Thales e Rita Fotografias Ltda	Aguardando avaliação	Orçamento	R\$0,00	15/06/2026	NOBREAK	SMS	LITE 6005 11S	65820626	Editar Imprimir
10	Luis Diogo Renato Rezende	Autorizado, reparo em andamento	Orçamento	R\$0,00	15/06/2026	TELEVISOR	LG	43UA7500PSA	AZ8346HY3736	Editar Imprimir
9	Emanuelly Dalane da Cruz	Autorizado, aguardando peça	Orçamento	R\$0,00	15/06/2026	CAIXA AMPLIFICADA	POLYVOX	XT-660T	4632862784967289623	Editar Imprimir

Fonte: Os autores (2026).

4 RESULTADO E DISCUSSÃO

O sistema atingiu os objetivos, resultando em uma aplicação web funcional para o gerenciamento de ordens de serviço, voltado para micro e pequenas empresas de

assistência técnica. Houve uma validação juntamente com um técnico atuante em uma microempresa da região de Uberaba, que atua no ramo de eletrodomésticos, o qual orientou o desenvolvimento com base no sistema já utilizado em sua loja, descrito por ele como ultrapassado e que não atendia adequadamente às necessidades do dia a dia, alinhando-se diretamente ao problema identificado.

A participação do técnico durante o desenvolvimento permitiu que as funcionalidades fossem definidas com base em necessidades reais de uso. Sua orientação determinou a definição dos campos presentes no cadastro de OS, como tipo de atendimento, estado de conservação do aparelho, observações e acessórios, refletindo diretamente as informações utilizadas na rotina de uma assistência técnica.

Todas as funcionalidades implementadas – cadastro de clientes, cadastro de ordens de serviço, listagem e controle de status – foram avaliadas positivamente. A interface limpa, acessível e fluida representa diferenciais em relação ao sistema anteriormente utilizado na empresa.

A arquitetura adotada demonstrou-se adequada para o contexto proposto. O SQLite eliminou a necessidade de um servidor externo, simplificando a implantação em ambiente desktop. O modelo App Router do Next.js permitiu concentrar frontend e backend em um único projeto, reduzindo a complexidade do sistema. O TypeScript contribuiu para diminuir os erros durante o desenvolvimento e para a manutenibilidade do código.

Em comparação ao sistema anteriormente utilizado pela empresa, observou-se que a solução existente apresentava uma interface visualmente desatualizada e uma estrutura mais complexa, contendo funcionalidades que não eram frequentemente utilizadas na rotina operacional da assistência técnica. Essa característica dificultava a navegação e o acesso às informações, impactando negativamente a produtividade dos usuários. Em contrapartida, o sistema desenvolvido priorizou uma interface mais

limpa e intuitiva, concentrando-se nas funcionalidades essenciais para o gerenciamento de clientes e ordens de serviço, proporcionando maior praticidade e eficiência no uso diário.

Apesar dos resultados positivos obtidos, algumas limitações foram identificadas durante o desenvolvimento da solução. Atualmente, o sistema não possui um mecanismo de autenticação de usuários, permitindo acesso irrestrito às funcionalidades disponíveis. Além disso, a utilização do SQLite como sistema de gerenciamento de banco de dados torna a solução mais adequada para ambientes com poucos usuários simultâneos. Dessa forma, caso o sistema seja adotado por empresas de maior porte ou que demandem acesso concorrente mais intenso, poderão ser necessárias adaptações na infraestrutura e na arquitetura de armazenamento de dados para garantir melhor desempenho e escalabilidade.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de um sistema web para gerenciamento de ordens de serviço voltado para micro e pequenas empresas de assistência técnica. O sistema desenvolvido por meio de Next.js, React, TypeScript e SQLite resulta em uma aplicação funcional, acessível e de fácil implantação em ambiente desktop.

O objetivo proposto foi alcançado, entregando funcionalidades que atendem diretamente às necessidades operacionais do segmento, como cadastro de clientes, cadastro e listagem de ordens de serviço e controle de status. A validação por parte de um técnico atuante na área confirmou a aderência do sistema à realidade das empresas do setor, que ainda dependem de processos manuais ou sistemas desatualizados.

A arquitetura utilizada mostrou-se adequada para o objetivo de uso, priorizando a simplicidade de implantação sem comprometer a organização e a manutenibilidade do código. A interface desenvolvida com foco na simplicidade permite que qualquer usuário, mesmo sem conhecimentos técnicos avançados, utilize o sistema com facilidade.

Como trabalho futuro, seria interessante ampliar as funcionalidades do sistema por meio da implementação de autenticação de usuários e controle de níveis de acesso, aumentando a segurança das informações armazenadas e garantindo maior controle sobre as operações realizadas na aplicação.

Além disso, o desenvolvimento de uma versão mobile permitiria que os técnicos realizem o registro e a consulta de ordens durante atendimentos externos ou em domicílio. Ampliando a mobilidade da solução, reduzindo a necessidade de registrar as informações posteriormente, tornando o processo de atendimento mais ágil e eficiente.

Outra possibilidade consiste na migração para um sistema gerenciador de banco de dados servidor, como MySQL ou PostgreSQL, permitindo maior escalabilidade, melhor gerenciamento das informações e suporte à utilização por múltiplos usuários simultaneamente. Essas melhorias possibilitariam a adoção da solução em empresas de diferentes portes e com demandas operacionais mais complexas.

REFERÊNCIAS

AMAZON WEB SERVICES (AWS). What's the difference between monolithic and microservices architecture? Disponível em:

<https://aws.amazon.com/compare/the-difference-between-monolithic-and-microservices-architecture/>. Acesso em: 6 jun. 2026.

IBGE. INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Estatísticas do Cadastro Central de Empresas. Rio de Janeiro: IBGE, 2022.

MICROSOFT. TypeScript: Documentation – Modules – Theory. TypeScript Handbook. Disponível em: <https://www.typescriptlang.org/docs/handbook/modules/theory.html>. Acesso em: 6 jun. 2026.

NEXT.JS. Next.js Documentation. Disponível em: <https://nextjs.org/docs>. Acesso em: 6 jun. 2026.

PICOTO, W.; CRESPO, N.; CARVALHO, F. A influência da estrutura tecnologia-organização-ambiente e da orientação estratégica no uso da computação em nuvem, mobilidade empresarial e desempenho. Revista Brasileira de Gestão de Negócios, v. 22, n. 1, 2021.

REACT. Start a New React Project. 2024. Disponível em: <https://react.dev/learn/creating-a-react-app>. Acesso em: 6 jun. 2026.

ROSA, G. H.; CASAGRANDE, T. C. Design de interfaces minimalistas e seu impacto na experiência do usuário. Revista Interface Tecnológica, Taquaritinga, v. 22, n. 2, p. 168-177, 2025.

SAUTER, V. Decision Support Systems: An Applied Managerial Application. 1. ed. New York: IE-Wiley, 1996.

SEBRAE. Ferramentas de gestão para micro e pequenas empresas. Brasília: SEBRAE, 2024.

SQLITE. About SQLite. SQLite Documentation. Disponível em: <https://www.sqlite.org/about.html>. Acesso em: 6 jun. 2026.