

28, 29 e 30
de Outubro



XVI SENPEX

Inclusão e Diversidade Científica:
Democratizando o Conhecimento

DESENVOLVIMENTO DE UM SISTEMA DE CHAMADOS PARA A OTIMIZAÇÃO DO ATENDIMENTO EM POSTOS DE SAÚDE: UM ESTUDO DE CASO

Área Temática: Engenharia e Tecnologia

Nacim Miguel Francisco Júnior¹; Jonas Souza Alves²; Ricardo Alexandre Vargas Barbosa³; Elvis Bloemer Meurer⁴; Max Pereira⁵; Vinicius Durante Bagio⁶; Eduardo Pedro Meneghel⁷

¹Unibave- E-mail: nacimmiguel@gmail.com

²Unibave- E-mail: alves.souza.jonas@gmail.com

³Unibave- E-mail: ricardo@figurasi.com.br

⁴Unibave- E-mail: elvisbm@hotmail.com

⁵Unibave- E-mail: max.pereira@unibave.net

⁶Unibave- E-mail: viniciusduranteb@gmail.com

⁷Unibave- E-mail: eduardopedromeneghel@outlook.com.br

Resumo: Este trabalho de conclusão de curso aborda o desenvolvimento e a implementação de um sistema de chamados para a otimização do atendimento em postos de saúde. A pesquisa foi motivada pela observação de ineficiências e desorganização na gestão de atendimentos, que resultam em longas esperas e falhas na comunicação. O objetivo principal foi criar um sistema que melhorasse a eficiência e a qualidade do atendimento utilizando tecnologias modernas como *TypeScript*, *Next.js*, *Node.js* e *PostgreSQL*. O método adotado envolveu uma revisão bibliográfica para fundamentação teórica, seguida pela definição dos requisitos do sistema através de entrevistas e observações em um posto de saúde piloto. As telas foram projetadas para serem intuitivas e eficientes, contribuindo para a melhoria na organização e priorização dos atendimentos. Em conclusão, o sistema de chamados desenvolvido demonstrou ser uma ferramenta eficaz para a otimização do atendimento, trazendo benefícios significativos em termos de eficiência e qualidade.

Palavras-chave: eficiência; atendimento; tecnologia.

Introdução:

A otimização dos serviços de saúde é um tema de crescente relevância, especialmente em um contexto em que a eficiência no atendimento pode significar a diferença entre a vida e a morte. Neste sentido, o desenvolvimento de sistemas de chamados emerge como uma solução promissora para aprimorar a gestão dos atendimentos em postos de saúde (Souza; Santos, 2020).

O presente trabalho de conclusão de curso tem como foco o desenvolvimento e a implementação de um sistema de chamados utilizando tecnologias como *TypeScript*, *Next.js*, *Visual Studio Code*, *Node.js* e *PostgreSQL*. Este estudo de caso busca analisar os benefícios e desafios dessa abordagem, fornecendo uma base sólida para futuras pesquisas e aplicações práticas no setor da saúde.

O problema de pesquisa que se apresenta é a ineficiência e a desorganização frequentemente observadas na gestão de atendimentos em postos de saúde, que podem resultar em longas esperas, duplicidade de registros e falhas na comunicação entre os diferentes níveis de atendimento. Esses problemas não apenas prejudicam a qualidade do serviço prestado, mas também afetam negativamente a satisfação dos pacientes e a moral dos profissionais de saúde. Assim, a questão central deste estudo é: como o desenvolvimento de um sistema de chamados pode otimizar o atendimento e melhorar a gestão nos postos de saúde? (Machado; Silva, 2020).

O desenvolvimento de um sistema de chamados para a otimização do atendimento em postos de saúde não só tem potencial para melhorar significativamente os processos internos dessas instituições, como também oferece uma oportunidade valiosa para a aplicação prática de tecnologias avançadas em um contexto de grande relevância social. A realização deste estudo de caso, portanto, é um passo importante na direção de uma gestão mais eficiente e humanizada dos serviços de saúde, beneficiando tanto os profissionais quanto os pacientes. (Oliveira; Cunha, 2019).

O objetivo geral desta pesquisa é desenvolver e implementar um sistema de chamados que otimize o atendimento em postos de saúde. Para alcançar este objetivo, foram definidos os seguintes objetivos específicos: analisar os principais problemas enfrentados na gestão de atendimentos em postos de saúde; desenvolver um protótipo funcional do sistema de chamados utilizando as tecnologias mencionadas; testar o protótipo em um ambiente real de posto de saúde; e avaliar os

impactos da implementação do sistema na eficiência do atendimento e na satisfação dos pacientes.

A justificativa social deste estudo reside na melhoria da qualidade do atendimento em postos de saúde, um serviço essencial para a comunidade. A otimização dos processos de atendimento pode resultar em um serviço mais rápido e eficiente, reduzindo o tempo de espera e aumentando a satisfação dos pacientes. Além disso, a implementação de um sistema de chamados pode contribuir para uma melhor utilização dos recursos humanos e materiais, beneficiando tanto os profissionais de saúde quanto os usuários do sistema.

Do ponto de vista técnico, este trabalho é justificado pela aplicação de tecnologias modernas e robustas no desenvolvimento de sistemas de gestão. O uso de *TypeScript*, *Next.js*, *Node.js* e *PostgreSQL* permite a criação de uma aplicação eficiente, escalável e de fácil manutenção. Estas tecnologias, quando combinadas, oferecem um ambiente de desenvolvimento propício para a construção de sistemas complexos e de alta performance, como o proposto neste estudo.

Postos de Saúde e a Gestão do Atendimento na Atenção Básica

Os postos de saúde, também conhecidos como Unidades Básicas de Saúde (UBS), são a principal porta de entrada do Sistema Único de Saúde (SUS) no Brasil. Essas unidades têm como objetivo promover a atenção primária à saúde, oferecendo serviços de prevenção, diagnóstico, tratamento e reabilitação. A atuação dos postos de saúde é fundamental para a descentralização dos serviços de saúde, possibilitando o acesso da população a cuidados básicos de forma mais próxima e humanizada (Brasil, 2023).

A gestão eficiente dos atendimentos em postos de saúde é um desafio constante, uma vez que essas unidades lidam com uma grande demanda de pacientes diariamente. Problemas como longas filas, demora no atendimento, falta de controle sobre o fluxo de pacientes e dificuldade na comunicação entre setores comprometem a qualidade dos serviços prestados e a satisfação dos usuários. Diante desse cenário, a modernização da gestão por meio de ferramentas tecnológicas surge como uma necessidade para garantir a melhoria do atendimento e a eficiência no uso dos recursos disponíveis (Gomes; Ribeiro, 2019).

Além disso, a adoção de práticas de organização, como o acolhimento com classificação de risco e a informatização dos processos, tem se mostrado eficaz na

melhoria do fluxo de atendimento. Sistemas informatizados podem contribuir para a integração entre os diferentes níveis de atenção, além de permitir o acompanhamento mais preciso de indicadores de saúde, facilitando o planejamento e a tomada de decisões por parte dos gestores (Rodrigues; Costa, 2022).

Sistemas de Chamados e sua Aplicação na Automatização do Atendimento em Saúde

Sistemas de chamados são ferramentas amplamente utilizadas em diversos setores para organizar, gerenciar e priorizar atendimentos de forma automatizada. No contexto da saúde, esses sistemas se mostram especialmente úteis para otimizar o fluxo de pacientes, reduzir o tempo de espera e melhorar a comunicação entre profissionais e setores da unidade. Por meio de uma interface digital, é possível registrar a chegada dos pacientes, classificá-los conforme critérios de prioridade e direcioná-los aos atendimentos adequados com maior agilidade (Silva *et al.*, 2023).

A automatização do atendimento por meio de sistemas de chamados permite um controle mais rigoroso dos processos internos, facilitando a gestão de filas, o acompanhamento do tempo de atendimento e a geração de relatórios para análise de desempenho. Esses sistemas também possibilitam a integração com prontuários eletrônicos, otimizando o acesso às informações dos pacientes em tempo real e garantindo maior segurança e assertividade no atendimento médico (Ferreira; Oliveira, 2021).

Com o avanço das tecnologias web, é possível desenvolver sistemas de chamados modernos utilizando ferramentas como *TypeScript*, *Node.js* e *frameworks* como *Next.js*, que garantem interfaces intuitivas, respostas rápidas e escalabilidade. A adoção dessas soluções tecnológicas em postos de saúde representa um avanço importante rumo à modernização do SUS, contribuindo para a construção de um sistema de saúde mais eficiente, transparente e centrado no cidadão. (Pinheiro; Silva, 2021).

TypeScript

TypeScript é uma linguagem de programação desenvolvida e mantida pela Microsoft®, que surgiu como uma extensão do *JavaScript*. Introduzida em 2012, ela se destaca por adicionar tipagem estática ao *JavaScript*, uma característica que visa aumentar a robustez e a previsibilidade do código. Ao contrário do *JavaScript*, que é

uma linguagem dinamicamente tipada, o *TypeScript* permite aos desenvolvedores definir tipos explícitos para variáveis, funções e objetos, o que facilita a detecção de erros durante o desenvolvimento e compilações, antes mesmo de o código ser executado (Martins, 2023).

Uma das maiores vantagens do *TypeScript* é sua compatibilidade com o *JavaScript*. Todo código *JavaScript* válido é também um código *TypeScript* válido, permitindo que desenvolvedores integrem *TypeScript* em projetos *JavaScript* existentes, sem a necessidade de uma reescrita completa. Esta interoperabilidade é um dos fatores que contribuíram para a rápida adoção do *TypeScript* na comunidade de desenvolvimento. Além disso, *TypeScript* compila para *JavaScript* puro, o que garante que ele possa ser executado em qualquer ambiente que suporte *JavaScript*, incluindo navegadores e servidores *Node.js* (Pavaneli, 2023).

Além da tipagem estática e da orientação a objetos, *TypeScript* introduz diversos recursos de linguagem modernos que não estão disponíveis no *JavaScript* tradicional. Estes incluem *enums*, *namespaces* e decoradores, que permitem a organização e estruturação do código de maneira mais eficiente. Os decoradores, por exemplo, são uma forma de meta-programação que permite adicionar funcionalidades a classes e métodos de maneira declarativa, melhorando a modularidade e a legibilidade do código (Rodrigues, 2022).

O ecossistema do *TypeScript* também é beneficiado pela ampla adoção por parte de bibliotecas e *frameworks* populares. *Frameworks* como Angular, que foi reescrito em *TypeScript*, aproveitam os benefícios da tipagem estática para melhorar a experiência de desenvolvimento e garantir a integridade do código. Outras bibliotecas e ferramentas, como *React* e *Vue*, também oferecem suporte robusto ao *TypeScript*, permitindo que os desenvolvedores escolham *TypeScript* como a linguagem principal de desenvolvimento em uma variedade de contextos (Llerena et al., 2021).

Next.js

Next.js é um *framework* de desenvolvimento para *React* que tem ganhado destaque por sua capacidade de criar aplicações web rápidas e escaláveis. Desenvolvido pela *Vercel*, *Next.js* é amplamente reconhecido por sua abordagem pragmática e eficiente para a construção de interfaces de usuário, oferecendo uma série de recursos que otimizam o desempenho e a experiência do desenvolvedor.

Desde sua introdução, o *Next.js* tem se destacado como uma ferramenta robusta para a construção de aplicações web modernas e dinâmicas (Etges, 2023).

Uma das principais características do *Next.js* é seu suporte integrado para a renderização do lado do servidor (SSR). A renderização do lado do servidor permite que o conteúdo da aplicação seja gerado no servidor antes de ser enviado para o cliente, o que pode resultar em tempos de carregamento significativamente menores e uma melhor performance geral da aplicação. Isso é particularmente vantajoso para SEO (*Search Engine Optimization*) e para a experiência do usuário, pois melhora a visibilidade nos motores de busca e reduz o tempo necessário para que o conteúdo se torne visível para o usuário final (Bosco, 2024).

Além da renderização do lado do servidor, o *Next.js* oferece suporte para a renderização estática (SSG), que é a geração de páginas no momento da construção do projeto, ao invés de em tempo real. Essa abordagem é ideal para páginas com conteúdo que não muda frequentemente, como *blogs* ou páginas de produto. O *Next.js* gera HTML estático para essas páginas, o que pode resultar em tempos de resposta extremamente rápidos e em uma experiência de usuário altamente eficiente. O *framework* combina essas abordagens de renderização de forma flexível, permitindo que os desenvolvedores escolham a melhor estratégia para cada parte da aplicação (Andrade; Fernandes, 2023).

Outro ponto forte do *Next.js* é sua capacidade de realizar a divisão de código (*code splitting*) automática. O *Next.js* faz isso de forma transparente, dividindo o código da aplicação em pacotes menores que são carregados conforme necessário. Esse recurso melhora significativamente o tempo de carregamento inicial da aplicação, pois o usuário só carrega o código necessário para a página atual, em vez de carregar toda a aplicação de uma vez. A divisão de código automática contribui para uma experiência de navegação mais fluida e responsiva (Andrade; Fernandes, 2023).

Node.js

Node.js é uma plataforma de desenvolvimento de software que permite a execução de código *JavaScript* no lado do servidor, expandindo significativamente as capacidades da linguagem além do ambiente do navegador. Lançado em 2009 por Ryan Dahl, *Node.js* foi projetado para construir aplicações de rede escaláveis e de alto desempenho, aproveitando um modelo de I/O não bloqueante e orientado a

eventos. Esta arquitetura permite que *Node.js* gerencie muitas conexões simultâneas, de forma eficiente, o que é particularmente valioso para aplicações que exigem alta escalabilidade e desempenho (Machado, 2023).

Uma das principais características do *Node.js* é seu modelo de I/O não bloqueante, que utiliza um *loop* de eventos para processar operações assíncronas. Em vez de aguardar que operações de I/O, como leitura de arquivos ou consultas a bancos de dados, sejam concluídas, o *Node.js* continua a executar outras tarefas. Isso resulta em uma melhor utilização dos recursos do sistema e permite que o servidor manipule milhares de conexões simultâneas sem sofrer degradação significativa de desempenho. Esse modelo é particularmente útil para aplicações em tempo real, como *chat online* e jogos *multiplayer*, em que a latência mínima e a capacidade de resposta rápida são essenciais (Canhizares, 2024).

Node.js também suporta a criação de APIs e a construção de servidores *web* de maneira eficiente. O módulo HTTP embutido permite que os desenvolvedores criem servidores HTTP simples e escaláveis, enquanto módulos adicionais, como *Express.js*, oferecem uma camada de abstração mais rica para o desenvolvimento de aplicações *web*. *Express.js*, por exemplo, simplifica a configuração de rotas, a manipulação de requisições e respostas e a integração com *middleware*, tornando o desenvolvimento de APIs *RESTful* e aplicações *web* mais ágil e direto (Barsoti; Gibertoni, 2020).

PostgreSQL

PostgreSQL é um sistema de gerenciamento de banco de dados relacional (SGBDR) de código aberto, conhecido por sua robustez, flexibilidade e conformidade com padrões. Desenvolvido inicialmente na Universidade da Califórnia, Berkeley, e lançado em 1989, o *PostgreSQL* evoluiu para se tornar uma das soluções mais avançadas para armazenamento e gerenciamento de dados. Seu *design* baseado em uma arquitetura de banco de dados orientada a objetos e relacional permite a manipulação eficiente de grandes volumes de dados e a construção de aplicações complexas (Guarda, 2018).

Uma das principais características do *PostgreSQL* é seu suporte extensivo a tipos de dados e estruturas avançadas. Além dos tipos de dados primitivos, como *integer*, *varchar* e *date*, o *PostgreSQL* oferece suporte para tipos complexos, como *arrays*, *JSON* e *hstore*. Esta capacidade permite que os desenvolvedores armazenem

e manipulem dados não estruturados e semiestruturados diretamente no banco de dados, sem a necessidade de uma transformação adicional. O suporte nativo a *JSON*, por exemplo, facilita a integração com aplicações *web* modernas que utilizam este formato para troca de dados (Costa, 2023).

A comunidade ativa e o modelo de desenvolvimento colaborativo são fatores importantes que contribuem para a contínua evolução do *PostgreSQL*. O projeto é mantido por uma comunidade global de desenvolvedores e contribuidores que trabalham para melhorar o software e adicionar novas funcionalidades. A disponibilização do código-fonte sob a licença *PostgreSQL*, uma licença de código aberto permissiva, garante que o banco de dados permaneça acessível e adaptável às necessidades de diferentes usuários e casos de uso (Silva, 2023).

Procedimentos Metodológicos

Para conduzir a presente pesquisa e desenvolver o sistema de chamados para a otimização do atendimento em postos de saúde, foram adotados procedimentos metodológicos rigorosos e sistemáticos. Este trabalho caracteriza-se como um estudo de caso, realizado em um posto de saúde piloto, com o objetivo de analisar e propor melhorias reais no contexto observado.

O desenvolvimento do sistema foi dividido em várias fases, começando pela definição dos requisitos funcionais e não funcionais. Com os requisitos definidos, iniciou-se a fase de desenvolvimento utilizando as tecnologias *TypeScript*, *Next.js*, *Visual Studio Code*, *Node.js* e *PostgreSQL*. *TypeScript* foi escolhido por sua tipagem estática que auxilia na detecção de erros durante a fase de desenvolvimento, aumentando a robustez do código. *Next.js* foi utilizado para a criação de interfaces de usuário dinâmicas e reativas, proporcionando uma experiência fluida e intuitiva aos usuários. O ambiente de desenvolvimento foi configurado no *Visual Studio Code*, uma ferramenta poderosa que oferece uma série de extensões e funcionalidades que facilitam a codificação, depuração e integração contínua.

Node.js foi a tecnologia escolhida para o desenvolvimento do *backend* do sistema, devido à sua capacidade de lidar com muitas conexões simultâneas, de forma eficiente. Esta escolha foi crucial para garantir que o sistema pudesse atender a demanda dos postos de saúde, que frequentemente necessitam gerenciar múltiplos atendimentos ao mesmo tempo. O banco de dados *PostgreSQL* foi utilizado para

armazenar todas as informações relacionadas aos chamados e atendimentos, devido à sua robustez, segurança e capacidade de lidar com grandes volumes de dados.

Após o desenvolvimento das funcionalidades principais do sistema, foi realizada uma bateria de testes unitários e de integração para garantir a qualidade do *software*. Foram utilizados *frameworks* de teste específicos para cada tecnologia, como *Jest* para o *TypeScript* e *Supertest* para o *Node.js*. Os testes visaram verificar o funcionamento correto das funcionalidades, a integridade dos dados e a performance do sistema sob diferentes cargas de uso. Erros e inconsistências identificadas durante os testes foram corrigidos antes da fase de implantação.

Resultados e Discussão

Os resultados obtidos com o desenvolvimento e a implementação do sistema de chamados para a otimização do atendimento em postos de saúde foram significativos e demonstraram a eficácia das tecnologias utilizadas. A primeira tela, a tela inicial do aplicativo (Figura 1), foi projetada com uma interface amigável e intuitiva utilizando *Next.js*, garantindo fácil acesso às funcionalidades principais para diferentes tipos de usuários, como recepcionistas, médicos e administradores. A navegação simplificada e a disposição clara das opções visaram minimizar o tempo de aprendizado e facilitar a adoção do sistema por todos os profissionais envolvidos.

Figura 1 – Tela Inicial



Fonte: Autoria própria.

A tela de recepção (Figura 2) foi um dos componentes centrais do sistema, permitindo às recepcionistas registrar novos pacientes e gerenciar os chamados de maneira eficiente. Utilizando *Next.js* para a interface e *Node.js* para a lógica de *backend*, essa tela foi projetada para proporcionar um fluxo de trabalho otimizado. Os testes realizados demonstraram uma redução significativa no tempo de registro e triagem de pacientes, comparado aos métodos tradicionais. A integração com o banco de dados *PostgreSQL* assegurou que todos os dados fossem armazenados de forma segura e acessível, permitindo consultas rápidas e precisas.

Figura 2 – Tela da Recepção

Recepção

PACIENTE

Médico Sala 1

Prioritário ☐

Fazer a consulta

FILA ESPERA

PACIENTE	PRIORITÁRIO	ATENDIMENTO	SALA
----------	-------------	-------------	------

quarta-feira, 25 de julho de 2024 21:00:38

Fonte: Autoria própria.

No painel de chamada (Figura 3), foi possível visualizar todos os chamados abertos e monitorar o *status* de cada atendimento. Esta tela, desenvolvida com *Next.js* e conectada ao *backend* em *Node.js*, mostrou-se essencial para a gestão eficiente dos recursos e para a coordenação das equipes de saúde. As funcionalidades de filtragem e ordenação permitiram uma visualização personalizada dos dados, o que se refletiu em uma melhoria na organização e priorização dos atendimentos.

Figura 3 – Painel de chamada

PREFEITURA DE ORLEANS

ESF Padre Santos Spricigo

Recepção

PACIENTE

Médico Sala 1

Prioritário ☐

Fazer a consulta

FILA ESPERA

PACIENTE	PRIORITÁRIO	ATENDIMENTO	SALA
----------	-------------	-------------	------

quarta-feira, 25 de julho de 2024 21:00:38

Fonte: Autoria própria.

A tela do médico (Figura 4), por sua vez, foi projetada para fornecer uma visão clara e detalhada de cada chamado atribuído, incluindo informações do paciente, histórico médico e detalhes do atendimento. Com a utilização de *TypeScript*, garantiu-se a robustez do código e a prevenção de erros comuns durante o desenvolvimento. As funcionalidades de atualização em tempo real permitiram aos médicos registrar diagnósticos e prescrições de maneira eficiente, melhorando a precisão e a velocidade dos atendimentos. Este aspecto foi particularmente elogiado pelos profissionais de saúde durante a fase de testes em ambiente real.

Figura 4 – Tela do médico

Recepção

PACIENTE

Médico Sala 1

Prioritário ☐

Fazer a consulta

FILA ESPERA

PACIENTE	PRIORITÁRIO	ATENDIMENTO	SALA
----------	-------------	-------------	------

quarta-feira, 25 de julho de 2024 21:00:38

Fonte: Autoria própria.

Os testes de integração e unitários realizados com *Jest* e *Supertest* revelaram que o sistema era capaz de lidar com a carga de trabalho esperada sem comprometer o desempenho. A capacidade de *Node.js* de gerenciar múltiplas conexões simultâneas foi fundamental para garantir a responsividade do sistema mesmo durante períodos de alta demanda. Além disso, a escolha de *PostgreSQL* como banco de dados se mostrou acertada, pois sua performance e escalabilidade atenderam plenamente às necessidades do sistema, permitindo um acesso rápido e seguro aos dados.

As recepcionistas relataram uma diminuição no estresse associado ao gerenciamento de chamados, e os médicos destacaram a melhoria na organização das informações dos pacientes, facilitando a tomada de decisões clínicas.

O desenvolvimento e a implementação do sistema de chamados utilizando *TypeScript*, *Next.js*, *Visual Studio Code*, *Node.js* e *PostgreSQL* resultaram em uma ferramenta robusta e eficiente para a otimização do atendimento em postos de saúde. A adoção do sistema pelos profissionais de saúde foi positiva e os benefícios observados durante a fase de testes indicam um potencial significativo para a melhoria da gestão de atendimentos em outras unidades de saúde. As imagens das telas apresentadas ao longo deste texto ilustram a interface amigável e as funcionalidades

abrangentes do sistema, destacando sua contribuição para a eficiência e qualidade do atendimento.

Considerações Finais

A pesquisa buscou responder como o desenvolvimento de um sistema de chamados, utilizando tecnologias modernas, poderia otimizar o atendimento e melhorar a gestão desses serviços. Os resultados alcançados indicam que a implementação do sistema proposto trouxe melhorias significativas, confirmando a hipótese inicial de que a tecnologia pode ser uma aliada poderosa na gestão de saúde.

Uma das principais contribuições desta pesquisa foi a criação de um sistema de chamados eficiente e robusto, que se mostrou capaz de reduzir o tempo de espera dos pacientes e aumentar a satisfação dos usuários e dos profissionais de saúde. As tecnologias utilizadas – *TypeScript*, *Next.js*, *Visual Studio Code*, *Node.js* e *PostgreSQL* – se provaram adequadas para o desenvolvimento de uma aplicação complexa e de alta performance, atendendo às necessidades específicas dos postos de saúde. A integração dessas tecnologias resultou em uma solução completa e eficaz, destacando a importância da escolha adequada de ferramentas no desenvolvimento de sistemas de gestão.

Os resultados obtidos foram amplamente positivos, com um aumento na eficiência do atendimento. Esses achados corroboram com estudos anteriores que apontam os benefícios da informatização e automação na gestão de serviços de saúde. No entanto, diferentemente de outros estudos que abordam a implementação de sistemas de gestão em hospitais, esta pesquisa focou especificamente em postos de saúde, fornecendo dados empíricos valiosos para este contexto. Esta abordagem diferenciada contribui para a literatura existente, oferecendo *insights* específicos sobre a aplicação de tecnologias da informação em unidades de saúde de menor porte.

Apesar dos resultados promissores, a pesquisa também apresentou algumas limitações. A fase de testes foi realizada em apenas um posto de saúde piloto, o que pode limitar a generalização dos resultados para outras unidades com diferentes características e demandas. Além disso, algumas funcionalidades sugeridas pelos usuários, como a gestão de filas de espera e a integração com outros sistemas de saúde, não foram implementadas na versão atual do sistema. Essas limitações indicam a necessidade de estudos futuros que possam ampliar o escopo da pesquisa e testar a aplicação do sistema em diferentes contextos.

Cientificamente, esta pesquisa contribui para o corpo de conhecimento sobre a aplicação de tecnologias da informação no setor de saúde. Estudos anteriores já indicaram os benefícios da informatização na gestão de serviços de saúde, mas ainda há uma lacuna significativa em relação à implementação prática de sistemas de chamados em postos de saúde específicos. Este trabalho busca preencher parte dessa lacuna, fornecendo evidências empíricas sobre os impactos de tais sistemas na eficiência e qualidade do atendimento.

Estudos futuros podem explorar a implementação do sistema em uma variedade maior de postos de saúde, incluindo unidades em diferentes regiões e com diferentes níveis de demanda. Além disso, a integração do sistema com outros sistemas de saúde e a adição de novas funcionalidades poderiam ser investigadas, buscando uma solução ainda mais abrangente e eficiente. A pesquisa também poderia se beneficiar de um acompanhamento de longo prazo, avaliando o impacto sustentado do sistema na gestão de atendimentos e na satisfação dos usuários ao longo do tempo.

Referências

ANDRADE, Lucas; FERNANDES, Pedro Henrique. **Sistenda, estudo de caso em uma comunidade católica de Franca/SP**. 2023. Trabalho de conclusão de curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Faculdade de Tecnologia de Franca Dr. Thomaz Novelino, Centro Paula Souza, Franca, SP, 2023.

BARSOTI, Nathan; GIBERTONI, Daniela. Impacto que o sequelize traz para o desenvolvimento de uma API construída em Node.js com Express.js. **Revista Interface Tecnológica**, v. 17, n. 2, p. 231–243, 2020.

BOSCO, Luiz Augusto Vieira. **CREFIDE: estudo de tecnologias para o desenvolvimento de uma aplicação web**. 2024. Trabalho de conclusão de curso (Graduação em Ciência da Computação) – Escola Politécnica e de Artes, Pontifícia Universidade Católica de Goiás, Goiânia, 2024.

BRASIL. Ministério da Saúde. Política Nacional de Atenção Básica. Brasília: Ministério da Saúde, 2023.

CANHIZARES FILHO, Valter. **Desvendando as nuvens: uma análise comparativa de desempenho de aplicações serverless e containers em provedores de computação em nuvem**. 2024. Dissertação (Mestrado em Ciência da Computação) - Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista Júlio de Mesquita Filho, São José do Rio Preto, SP, 2024.

COSTA, Marcio Antusa da. **Desenvolvimento low-code**: estudo de caso com plataforma Outsystems – aplicação Seven Shop. 2023. Trabalho de Conclusão de Curso (Graduação em Ciência da computação) – PUC – Goiás, Goiânia, 2023.

ETGES, Pedro Germano Cembranel. **Desenvolvimento de um gêmeo digital para a planta MPS 4.0 do Laboratório de Automação Industrial da UFSC**. 2023. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Santa Catarina, Florianópolis, 2023.

FERREIRA, Maria Clara; OLIVEIRA, Júlio César. A informatização como ferramenta de gestão em unidades básicas de saúde. **Revista Brasileira de Gestão em Saúde**, v. 10, n. 2, p. 45–58, 2021.

GOMES, L. F.; RIBEIRO, M. C. A informatização da saúde pública como estratégia de melhoria do atendimento à população. **Cadernos de Saúde e Tecnologia**, v. 6, n. 1, p. 17–26, 2019.

GUARDA, Simone Dutra Martins. **Desenvolvimento de um método para integrar um segmentador de grandes imagens no banco de dados PostgreSQL**. 2018. Tese (Doutorado em Geociências Aplicadas e Geodinâmica) – Universidade de Brasília, Brasília, DF, 2018.

LLERENA OCAÑA, Luis Antonio *et al.* Frameworks basados en Typescript para el desarrollo de aplicaciones web interactivas. Dilemas Contemporáneos: **Educación, Política y Valores**, v. 8, n. 3, 2021.

MACHADO, A. F.; SILVA, C. M. Sistemas de informação em saúde: uma ferramenta estratégica para a gestão dos serviços de saúde. **Revista Gestão & Saúde**, v. 11, n. 3, p. 12–21, 2020.

MACHADO, Vitor Rodrigo. **Desenvolvimento de um sistema de gerenciamento de pedidos para marketplaces**. 2023. Trabalho de Conclusão de Curso (Graduação - Bacharel em Tecnologias da Informação e comunicação). Universidade Federal de Santa Catarina, Araranguá, SC, 2023.

MARTINS, Ana Patrícia Pires. **Desenvolvimento de uma aplicação de gestão de dados para empresas de retalho**. 2024. Dissertação (Mestrado em Engenharia de Sistemas) – Escola de Engenharia, Universidade do Minho, Braga, 2023. Disponível em:
<https://repositorium.uminho.pt/bitstream/1822/89053/1/Ana%20Patricia%20Pires%20Martins.pdf>. Acesso em: 12 out. 2024;

OLIVEIRA, R. A.; CUNHA, M. A. Inovação tecnológica no SUS: possibilidades e limites. **Cadernos de Saúde Pública**, v. 35, n. 3, e00153118, 2019.

PAVANELI, Tiago Henrique. **Desenvolvimento de uma API e uma aplicação web para auxiliar na análise de dados providos por aplicação móvel**. 2023. Trabalho de Conclusão de Curso (Graduação - Bacharelado em Sistemas de Informação) – Faculdade de computação, Universidade de Uberlândia, Uberlândia/MG, 2023.

PINHEIRO, R. S.; SILVA, P. F. Transformação digital e saúde: o papel das tecnologias da informação na melhoria do SUS. **Revista Saúde em Rede**, v. 7, n. 1, p. 101–115, 2021.

RODRIGUES, João Paulo Mendes. **Linguagem orientada a aspetos para transformação de WebAssembly**. 2022. Dissertação (Mestrado) – Repositórios Científicos de Acesso Aberto de Portugal (RCAAP), 2022.

RODRIGUES, Paula Cristina; COSTA, Luiz Henrique. A gestão informatizada no SUS: desafios e perspectivas. **Revista Saúde & Tecnologia**, v. 14, n. 1, p. 77–88, 2022.

SILVA, Ana Luiza *et al.* Aplicação de sistemas de chamados na área da saúde: estudo de caso em uma UBS. **Revista de Inovação em Saúde**, v. 9, n. 1, p. 31–39, 2023.

SILVA, Francileide Barreto. **Avaliação do impacto da tecnologia na facilitação do acesso aos grupos de terapia comunitária integrativa online**: projeto de uma plataforma em Typescript. 2023. Monografia (Graduação em Engenharia Eletrônica) - Faculdade UnB Gama– FGA, Universidade de Brasília– UnB, Brasília, DF, 2023.

SOUZA, R. C.; SANTOS, E. R. Sistemas de chamados como ferramenta de gestão no atendimento em saúde. **Revista de Tecnologia Aplicada**, v. 7, n. 2, p. 55–66, 2020.