



**UNIVERSIDADE  
FEDERAL RURAL  
DE PERNAMBUCO**



Desenvolvimento de uma Aplicação Web para a Gestão de Atendimentos Presenciais ao Público Usuário de Serviços de Saneamento Básico

---

**Relatório Técnico relativo ao Trabalho de Conclusão Curso  
do Bacharelado em Sistemas de Informação na modalidade Empresa**

---

**Aluno**

Marcos Vinícius de Araujo Souza Peixoto

**Orientadora**

Roberta Macêdo Marques Gouveia  
Departamento de Estatística e Informática - DEINFO

12 de março de 2026

Marcos Vinícius de Araujo Souza Peixoto

## **Desenvolvimento de uma Aplicação Web para a Gestão de Atendimentos Presenciais ao Público Usuário de Serviços de Saneamento Básico**

Relatório Técnico apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

**Orientadora: Roberta Macêdo Marques Gouveia**

Recife

Fevereiro de 2026

## **Dedicatória**

Dedico este trabalho, primeiramente, à memória de meu pai, que, mesmo diante de inúmeras dificuldades, jamais deixou de acreditar que a educação seria o caminho para transformar a realidade de seus filhos. Desde a infância, fez tudo o que esteve ao seu alcance para manter os quatro filhos na melhor escola da cidade, sacrificando-se continuamente para garantir oportunidades que ele próprio não teve. Mesmo nos momentos em que eu, ainda criança, demonstrava desinteresse pelos estudos, aprontava e cheguei a afirmar que não queria mais frequentar a escola, situação que causou minha saída daquela instituição, ele nunca deixou de investir no meu futuro.

Recordo-me com clareza que no dia seguinte em que, ainda pela manhã, meu pai me levou até a escola na tentativa de me manter estudando. Naquele momento, sentou-se comigo para conversar e reforçou a importância da educação, utilizando como exemplo um tio que havia abandonado a escola ainda na quarta série e não trabalhava. Apesar de ser agricultor, de a escola localizar-se em sentido oposto ao seu local de trabalho e de não conseguir reverter a situação naquele dia, suas palavras permaneceram comigo ao longo da vida. Entre os poucos momentos em que o vi expressar orgulho de forma explícita, guardo como o mais marcante quando afirmou que um agricultor e uma professora, ambos de origem humilde, conseguiram colocar quatro filhos em uma universidade federal. Mesmo sendo técnico em agronomia e não podendo auxiliar diretamente nas atividades escolares, nunca deixou de acreditar no potencial dos filhos, crença que se tornou um dos maiores legados de sua vida.

Dedico também este trabalho à minha mãe, que esteve presente de forma constante, incansável e silenciosa, especialmente no auxílio às atividades de matemática, no cuidado diário, no incentivo e na confiança depositada em mim nos momentos em que nem eu mesmo acreditava. Sua paciência, dedicação e apoio incondicional foram fundamentais para que eu persistisse e chegasse até aqui.

Por fim, dedico ao meu tio, que exerceu o papel de um segundo pai em minha vida e na de meus irmãos, contribuindo tanto financeiramente quanto na educação e na formação de valores. Foi ele quem me auxiliou nos estudos até a oitava série, em uma fase em que eu demonstrava pouco interesse pela escola, e também quem despertou em mim o interesse por videogames, o qual, mais tarde, se transformaria no gosto pela área de Tecnologia da Informação. Sua presença, apoio e dedicação foram decisivos para minha trajetória acadêmica e pessoal.

## Resumo

**Resumo:** A organização eficiente de processos e a adequada coordenação de recursos são essenciais para o bom funcionamento das instituições e para a melhoria da qualidade dos serviços prestados à população, especialmente no domínio da saúde pública, área diretamente relacionada à oferta de serviços de saneamento básico. Nesse contexto, este trabalho apresenta o desenvolvimento de um sistema web voltado ao gerenciamento de atendimentos presenciais na área de saneamento, destinados ao público usuário desses serviços, com início em 2025. A solução tecnológica proposta tem como objetivo otimizar os processos de atendimento por meio da automação de agendamentos, controle de tempo e expediente, monitoramento dos serviços e disponibilização de informações dos atendentes, promovendo maior eficiência operacional e possibilitando o acompanhamento das atividades em tempo real. O projeto fundamenta-se nos princípios da engenharia de software, programação web e gerenciamento de dados dinâmicos, contribuindo para o desenvolvimento acadêmico e para a aplicação prática. Como resultados esperados, destacam-se a otimização do fluxo de informações, a democratização do acesso aos dados, a melhoria da qualidade do atendimento, o aumento da satisfação dos usuários e o fortalecimento da gestão organizacional, refletindo positivamente nos serviços públicos de saúde e saneamento.

**Palavras chaves:** Algoritmo, web socket, API REST, Sistema de Informação, Gestão

**Abstract:** The efficient organization of processes and the proper coordination of resources are essential for the effective functioning of institutions and for improving the quality of services provided to the population, especially within the domain of public health, which is directly related to the provision of basic sanitation services. In this context, this study presents the development of a web-based system aimed at managing in-person services in the sanitation sector, intended for users of these services, initiated in 2025. The proposed technological solution seeks to optimize service processes through the automation of scheduling, time and work shift control, service monitoring, and the availability of attendant information, promoting greater operational efficiency and enabling real-time activity tracking. The project is grounded in the principles of software engineering, web programming, and dynamic data management, contributing to both academic development and practical application. The expected results include the optimization of information flow, the democratization of access to data, improved service quality, increased user satisfaction, and the strengthening of organizational management, positively impacting public health and sanitation services.

**Keywords:** Algorithm, WebSocket, REST API, Information System, Management

# Sumário

<b>1</b>	<b>Introdução</b>	<b>6</b>
1.1	Contexto de desenvolvimento do projeto . . . . .	7
1.2	Justificativa . . . . .	7
1.3	Objetivos . . . . .	8
<b>2</b>	<b>A empresa e sua atuação</b>	<b>8</b>
<b>3</b>	<b>Contexto de desenvolvimento do projeto</b>	<b>10</b>
<b>4</b>	<b>Desenvolvimento realizado na empresa</b>	<b>11</b>
4.1	A problemática e a solução proposta . . . . .	11
4.2	Tecnologias utilizadas . . . . .	14
4.2.1	Arquitetura de microsserviços . . . . .	16
4.2.2	Padrão CQRS . . . . .	16
4.2.3	Padrão Unit of Work . . . . .	17
4.2.4	Keycloak . . . . .	17
4.2.5	WebSocket . . . . .	18
4.2.6	Jsreport . . . . .	20
4.2.7	Ferramenta de apoio à gestão . . . . .	21
4.3	Contribuição . . . . .	21
4.3.1	Descrição e análise dos desenvolvidos . . . . .	22
<b>5</b>	<b>Dificuldades encontradas</b>	<b>26</b>
<b>6</b>	<b>Impactos da sua formação no seu trabalho</b>	<b>27</b>
<b>7</b>	<b>Conclusão</b>	<b>27</b>
	<b>Referências Bibliográficas</b>	<b>28</b>

# 1 Introdução

A gestão eficiente, quando alinhada ao desenvolvimento de software, torna-se um fator estratégico para a otimização do atendimento ao público, especialmente em organizações que lidam com grande volume de demandas e informações. A integração entre práticas de organização administrativa e soluções tecnológicas permite estruturar processos, padronizar fluxos de trabalho e aprimorar a tomada de decisões, resultando em maior agilidade, controle e qualidade nos serviços prestados. Nesse cenário, o uso de sistemas de informação desenvolvidos com base em princípios da engenharia de software contribui para a melhoria da eficiência operacional, a redução de falhas nos atendimentos e o fortalecimento da relação entre a instituição e seus usuários.

Vale ressaltar que a gestão representa um elemento fundamental para a organização, eficiência e sucesso em qualquer área do conhecimento. Compreender seus conceitos e princípios permite uma aplicação mais estruturada e embasada, favorecendo o planejamento, a utilização adequada dos recursos e a definição de estratégias eficazes para alcançar os objetivos propostos. Assim, a gestão atua como um instrumento essencial para orientar processos, otimizar resultados e promover o desenvolvimento das atividades (Martins et al, 2023).

De acordo com Nonato (2023), a gestão da informação é composta por seis princípios interrelacionados: a identificação das necessidades informacionais, a aquisição, a organização e o armazenamento das informações, o desenvolvimento de produtos informacionais, a distribuição e o uso da informação. Esses elementos formam um ciclo contínuo que garante a eficiência na utilização dos dados dentro de qualquer processo organizacional. Nesse contexto, ao se tratar do desenvolvimento de uma aplicação voltada à gestão de atendimentos, observa-se que todos esses princípios são aplicados de maneira integrada, contribuindo para a estruturação, controle e otimização das etapas envolvidas no sistema.

No cotidiano profissional, é comum o surgimento de diversas problemáticas que impactam diretamente a rotina de trabalho, especialmente aquelas relacionadas à organização, comunicação e gestão do tempo. Diante desse cenário, surge a necessidade de otimizar processos e melhorar a eficiência operacional no ramo de serviços de saneamento básico, buscando soluções que promovam atendimentos mais ágeis. Por conseguinte, a adoção de ferramentas tecnológicas, representa uma estratégia eficaz para enfrentar esses desafios (Rogério Júnior *et al*, 2021).

Ao considerar o contexto de grandes empresas e instituições que realizam atendimentos presenciais, observa-se a crescente necessidade de automatizar processos a fim de otimizar o tempo e aprimorar a experiência do cliente. A automação dos serviços de atendimento contribui para reduzir o tempo de espera, minimizar a formação de filas e melhorar o fluxo operacional, além de diminuir a sobrecarga dos atendentes, que frequentemente precisam lidar simultaneamente com o público e com a organização manual de fichas ou registros.

A implementação de sistemas tecnológicos voltados à gestão de atendimentos possibilita maior eficiência, controle e agilidade nos processos, beneficiando tanto a equipe de trabalho quanto os usuários do serviço. Assim, investir em soluções digitais representa não apenas um avanço tecnológico, mas também uma estratégia de gestão que visa à qualidade, produtividade e satisfação dos clientes (Santos *et al*, 2019).

A necessidade de uma empresa por um sistema eficiente de gerenciamento do processo de atendimento tornou-se evidente diante das dificuldades enfrentadas no controle de agendamentos, no monitoramento do tempo de atendimento entre funcionários e clientes, na verificação de expedientes, no registro de senhas atendidas e no acompanhamento dos serviços realizados. Esses desafios tornaram-se recorrentes e impactavam diretamente a qualidade e a agilidade dos atendimentos.

Diante desse cenário, surge a proposta de desenvolver uma aplicação web voltada à gestão de atendimentos presenciais para usuários de uma empresa que fornece serviços de saneamento básico, com o objetivo de automatizar e integrar todas as etapas do processo. Essa iniciativa mostra-se relevante por atender às necessidades tanto dos usuários quanto dos gestores, promovendo maior organização, transparência e agilidade nas atividades diárias. Além disso, a solução proposta incorpora princípios da gestão da informação, o que contribui para otimizar a tomada de decisões (Gasparini; Pretto, 2020).

Ao considerar a análise do contexto e as necessidades da empresa em relação à gestão de atendimentos aos usuários de serviços de saneamento básico, surge o seguinte questionamento: como o desenvolvimento de uma aplicação web para gestão de atendimentos presenciais pode automatizar processos, otimizar a organização e a eficiência operacional, e contribuir para a tomada de decisões pelos gestores?

## **1.1 Contexto de desenvolvimento do projeto**

Em decorrência das demandas do mercado por maior eficiência e agilidade nos atendimentos, a empresa responsável pelo setor de saneamento básico identificou a necessidade de otimizar seus processos de atendimento ao cliente. Nesse sentido, buscou-se aprimorar a forma de interação entre atendentes e usuários, com o objetivo de reduzir a burocratização e tornar o atendimento mais ágil e eficaz.

Nesse contexto, a implementação de uma ferramenta tecnológica adequada tornou-se estratégica para a substituição do sistema anteriormente empregado, o qual apresentava limitações operacionais. A migração para uma plataforma tecnologicamente mais atual possibilitou a informatização e a integração dos processos de atendimento, promovendo maior agilidade na tomada de decisão gerencial. Ademais, essa modernização contribuiu significativamente para a mitigação da formação de filas, a redução do tempo de espera dos usuários e o aprimoramento da eficiência operacional do serviço prestado.

O desenvolvimento do projeto foi iniciado no ano de 2025, com o objetivo de aprimorar o gerenciamento dos atendimentos ao cliente por meio da modernização dos processos internos e da integração de recursos tecnológicos voltados à eficiência operacional. A iniciativa surgiu da necessidade de otimização do fluxo de informações, garantindo maior agilidade e precisão nas etapas de atendimento. Nesse contexto, a proposta teve como finalidade a democratização do acesso às informações, a facilitação das atividades desempenhadas pelos atendentes. Além disso, contempla a automatização de etapas burocráticas, promovendo a melhoria da qualidade dos serviços prestados, o aumento da satisfação dos usuários e o fortalecimento da gestão organizacional.

A adoção de um sistema tecnologicamente mais avançado, aliado à sua usabilidade aprimorada, proporcionou maior eficiência no atendimento ao cliente, reduzindo significativamente o tempo de espera. Além disso, a melhoria nos mecanismos de busca e recuperação de informações possibilitou o acesso rápido e preciso aos registros, contribuindo para a resolução mais eficaz das queixas e demandas apresentadas pelos usuários.

## **1.2 Justificativa**

O desenvolvimento de uma aplicação web para a gestão de atendimentos presenciais aparentava ser uma tarefa de implementação simples e direta. Porém, durante o processo de construção e aperfeiçoamento do sistema, constatou-se que o projeto exigia um nível de complexidade técnica mais elevado, principalmente nas etapas relacionadas à transmissão e à atualização de dados em tempo real.

A necessidade de garantir que as informações sobre atendimentos, senhas e tempos de espera fossem exibidas de forma imediata e em tempo real demandou a utilização de tecnologias específicas como .NET (backend) e Vue.js (frontend) com a biblioteca SignalR para intermediar a comunicação via WebSocket, além de um planejamento minucioso do sistema. Esse desafio técnico envolveu a integração entre módulos de controle, algoritmos de fila e comunicação contínua entre cliente e servidor, exigindo atenção à eficiência e à estabilidade da aplicação.

A elaboração do projeto contribui significativamente para o aperfeiçoamento das competências em engenharia de software, programação web e gerenciamento de dados dinâmicos. Além disso, destaca-se a sua relevância prática, ao propor uma solução tecnológica capaz de gerenciar e otimizar os processos de atendimento.

### **1.3 Objetivos**

O objetivo geral do trabalho consiste no desenvolvimento de um sistema web voltado ao gerenciamento dos atendimentos presenciais em uma empresa prestadora de serviços de esgotamento sanitário, destinado à resolução de demandas dos usuários da rede de esgoto, tais como esclarecimentos e regularização de pagamentos de boletos, solicitações de serviços. O sistema tem como propósito automatizar os processos de agendamento, controle de tempo e expediente, monitoramento dos serviços realizados e disponibilização de informações dos atendentes, visando otimizar a eficiência operacional, qualificar o atendimento ao usuário e permitir o acompanhamento das atividades em tempo real.

Dentre os objetivos específicos têm-se:

- Analisar as necessidades da empresa quanto ao gerenciamento de atendimentos presenciais, identificando os principais pontos que exigem automatização.
- Projetar e desenvolver uma aplicação web que permita o agendamento de atendimentos, o controle de filas e o registro de serviços prestados.
- Implementar funcionalidades de monitoramento em tempo real, utilizando tecnologias como WebSocket para visualização dinâmica dos atendimentos.
- Criar algoritmos de controle de fila e agendamento, considerando regras específicas para diferentes postos de atendimento.
- Desenvolver módulos de controle de expediente, tempo médio de atendimento e desempenho dos funcionários.
- Disponibilizar relatórios e informações gerenciais que auxiliem na tomada de decisão e na melhoria da eficiência operacional.
- Testar e validar o sistema, garantindo o correto funcionamento das funcionalidades e a usabilidade para administradores e atendentes.

## **2 A empresa e sua atuação**

A Blue Technology é uma empresa pernambucana de tecnologia voltada ao desenvolvimento de software e prestação de serviços de Tecnologia da Informação (TI), com foco em soluções de gestão empresarial, Enter-

prise Resource Planning (ERPs) e plataformas digitais. A empresa foi fundada em 2013 e, desde então, vem se consolidando como uma fábrica de software com experiência em projetos para clientes privados e órgãos públicos da região.<sup>1</sup>

Originalmente estruturada como uma prestadora de serviços de suporte e desenvolvimento, a Blue Technology expandiu seu portfólio para oferta de produtos Software as a Service (SaaS) e soluções de gestão, atuando também como fornecedora para órgãos públicos e empresas do setor terciário.

A empresa oferece serviços de desenvolvimento de software sob medida, produtos de gestão empresarial, implantação e suporte, além de projetos de transformação digital. O mercado-alvo inclui empresas privadas de médio/grande porte e órgãos públicos da administração estadual e municipal, com forte atuação regional no Nordeste e expansão planejada para outras regiões do país.

Em relação ao foco de clientes, são empresariais e institucionais incluindo contratos e projetos voltados ao setor público.

A estruturação da empresa apresenta quadro de sócios e administradores registrados em bases públicas, com lideranças executivas visíveis em perfis institucionais. A estrutura segue modelo típico de empresas de tecnologia de médio porte: diretoria/fundador, áreas comerciais, de produto, engenharia (fábrica de software), suporte e operações.

O ingresso na Blue Technology ocorreu em 22 de julho de 2024, inicialmente no cargo de desenvolvedor full stack pleno, com atuação nas tecnologias .NET e Vue.js. Durante a trajetória profissional na organização, houve participação em projetos vinculados a dois clientes do setor público, contribuindo para o desenvolvimento, manutenção e evolução de sistemas corporativos com escopos voltados à gestão administrativa, controle de atendimentos, organização de fluxos operacionais e gerenciamento de informações institucionais. Além disso, foram desenvolvidos e aprimorados aplicativos destinados ao público usuário, permitindo a verificação de débitos, a abertura de alertas para ocorrência de vazamentos de água e esgoto, bem como a consulta do consumo de água. As soluções implementadas tiveram como finalidade otimizar processos internos, ampliar a transparência das informações, aumentar a eficiência operacional e qualificar a prestação de serviços públicos, assegurando maior confiabilidade dos dados e melhor atendimento ao cidadão.

Nesse contexto, os projetos desenvolvidos contemplaram sistemas corporativos integrados voltados à gestão operacional, comercial e administrativa, abrangendo soluções destinadas à consulta e ao acompanhamento do consumo de volume de água pelos usuários, ao parcelamento e à negociação de débitos, bem como ao controle da quantidade de água captada, tratada e distribuída até o consumidor final. Ademais, o escopo incluiu sistemas responsáveis pelo gerenciamento de ordens de fornecimento de serviços e materiais, essenciais para o suporte às atividades operacionais, assim como ferramentas para avaliação do desempenho de colaboradores, contribuindo para o aprimoramento dos processos internos, o apoio à tomada de decisão gerencial e o aumento da eficiência organizacional.

No ambiente desta empresa de saneamento básico, houve participação na integração de oito projetos distintos, dos quais seis foram acompanhados desde a fase de concepção, abrangendo as etapas de levantamento de requisitos, desenvolvimento e testes. Cada projeto apresentou desafios técnicos, tais como a integração entre sistemas legados e novas aplicações, padronização de Application Programming Interfaces (APIs), garantia de desempenho e escalabilidade, correção de falhas, versionamento de código e adequação a diferentes arqui-

---

<sup>1</sup>Almeida Machado Serviços em Gestão de Negócios LTDA, “Blue technology — consulta empresa,” ‘Econodata, 2025, acesso em: 08 dez. 2025. [Online]. Available: <https://www.econodata.com.br/consulta-empresa/18191583000140-almeida-machado-servicos-em-gestao-de-negocios-ltda>

teturas de software. Paralelamente, emergiram desafios de negócio, relacionados à compreensão e tradução das necessidades dos clientes, cumprimento de prazos, adaptação às regras específicas de cada órgão público, priorização de demandas, garantia da continuidade dos serviços e alinhamento das soluções tecnológicas aos objetivos institucionais. Esses desafios contribuíram significativamente para a ampliação da experiência prática em ambientes reais de desenvolvimento de software e para o aprimoramento de competências técnicas e organizacionais.

No âmbito dos desafios técnicos enfrentados, destaca-se a necessidade de desenvolver e integrar múltiplos sistemas que dependiam do acesso ao banco de dados DB2 for i, da International Business Machines Corporation (IBM), o qual apresenta restrições quanto à integração direta com aplicações desenvolvidas na plataforma .NET, em virtude da inexistência de drivers oficiais disponibilizados pelo fabricante. Diante dessa limitação tecnológica, tornou-se necessária a concepção de uma camada intermediária de comunicação, materializada por meio do desenvolvimento de uma API utilizando a linguagem Python, responsável por realizar a comunicação com o banco de dados e disponibilizar os dados de forma segura e padronizada para os demais sistemas. Essa solução arquitetural possibilitou a interoperabilidade entre tecnologias distintas, garantindo a continuidade dos projetos, além de contribuir para o aprimoramento das competências técnicas relacionadas à integração de sistemas, arquitetura de software e resolução de problemas em ambientes corporativos complexos.

No que se refere aos desafios de natureza gerencial e de negócios, observou-se a adoção, por parte do cliente, de um modelo de desenvolvimento em cascata, o qual pressupõe a definição completa e estável dos requisitos nas fases iniciais do projeto. Contudo, identificaram-se recorrentes dificuldades decorrentes da especificação inadequada ou incompleta dos requisitos, resultando em solicitações frequentes de alterações durante a fase de desenvolvimento. Em determinados cenários, verificou-se o início da implementação de funcionalidades sem que os requisitos estivessem formalmente consolidados, bem como situações em que não havia qualquer documentação escrita referente às fórmulas de cálculo necessárias ao sistema. Nesses casos, tornou-se indispensável a realização de reuniões diretas com o cliente, com o objetivo de compreender detalhadamente o passo a passo para a correta geração dos cálculos, evidenciando fragilidades no processo de levantamento e documentação de requisitos. Adicionalmente, após os alinhamentos realizados, fez-se necessária a reescrita do requisito, incorporando de forma clara e precisa as informações corretas relativas às fórmulas e às regras de cálculo, de modo a garantir a correta implementação da funcionalidade, bem como a rastreabilidade, o controle de escopo e a redução de retrabalho ao longo do desenvolvimento.

O reconhecimento pelo desempenho e pela evolução técnica demonstrada, a promoção surgiu como consequência em julho de 2025, ao cargo de desenvolvedor sênior, passando a assumir responsabilidades ampliadas no processo de desenvolvimento, na tomada de decisões técnicas e na orientação de atividades dentro dos projetos.

### **3 Contexto de desenvolvimento do projeto**

O projeto foi iniciado no ano de 2025, com o objetivo de aprimorar o gerenciamento dos atendimentos ao cliente por meio da modernização dos processos internos e da integração de recursos tecnológicos voltados à eficiência operacional. A iniciativa surgiu da necessidade de otimização do fluxo de informações, garantindo maior agilidade e precisão nas etapas de atendimento. Nesse contexto, a proposta teve como finalidade a democratização do acesso às informações, a facilitação das atividades desempenhadas pelos atendentes. Além disso, contempla a automatização de etapas burocráticas, promovendo a melhoria da qualidade dos serviços

prestados, o aumento da satisfação dos usuários e o fortalecimento da gestão organizacional.

Pela equipe responsável pelo projeto, considerando as entradas e saídas de pessoas, passaram seis desenvolvedores, dois estagiários, dois testadores, uma coordenadora de projeto, três analistas de requisitos e um líder técnico. As atividades desempenhadas abrangeram o desenvolvimento, a validação e a integração das funcionalidades do sistema, assegurando a conformidade com os requisitos previamente definidos, a qualidade do produto e o cumprimento dos prazos estabelecidos durante o processo de desenvolvimento.

## **4 Desenvolvimento realizado na empresa**

O desenvolvimento do projeto na empresa proporcionou uma experiência prática significativa, caracterizada por desafios técnicos e pela necessidade de constante colaboração entre os membros da equipe. Ao longo do processo, foi possível compreender de forma mais aprofundada as etapas que compõem o ciclo de engenharia de software, desde o levantamento e a análise de requisitos até a implementação e a realização de testes das funcionalidades propostas.

A atuação em um ambiente corporativo real possibilitou o aprimoramento de competências técnicas e interpessoais, fundamentais para o desempenho profissional na área de tecnologia. Entre as habilidades desenvolvidas, destacam-se a capacidade de liderança técnica, a organização de tarefas, a comunicação eficaz e o trabalho colaborativo, consideradas essenciais para a execução de projetos de software em contextos empresariais. A experiência em uma equipe reduzida, composta por profissionais com diferentes níveis de senioridade, demandou elevado senso de responsabilidade e proatividade, especialmente na resolução de problemas e no apoio às atividades dos demais integrantes do grupo.

Em relação ao aspecto técnico, a execução do projeto envolveu a aplicação de tecnologias e conceitos adquiridos ao longo da formação acadêmica, tais como estruturas de dados, algoritmos e boas práticas de programação. Esses conhecimentos foram empregados na implementação de mecanismos para agendamento de atendimentos, na configuração do painel eletrônico e na integração de funcionalidades em tempo real, viabilizadas por meio do uso de WebSocket. As atividades realizadas contribuíram para o aprofundamento da compreensão acerca do funcionamento interno de sistemas complexos, consolidando a aplicação prática dos conteúdos teóricos assimilados durante o curso.

### **4.1 A problemática e a solução proposta**

A necessidade de um atendimento mais eficiente e ágil surgiu diante das dificuldades relacionadas à gestão do tempo e ao excesso de burocracias, que tornavam o processo lento e pouco produtivo. A ausência de um sistema integrado resultava no aumento das filas, na demora dos atendimentos e na insatisfação dos usuários. Por conseguinte, o efeito cumulativo dessas demoras ocasionava atrasos na resolução das demandas e gerava acúmulo de trabalho, comprometendo a agilidade e a eficácia na entrega das soluções aos casos atendidos.

A problemática relacionada ao tempo de espera e à excessiva burocratização na resolução dos serviços, a otimização e a agilidade no atendimento foram identificadas como estratégias fundamentais para o aprimoramento do relacionamento entre os usuários dos serviços de saneamento básico e a empresa responsável por sua prestação.

Diante dessa situação, foi desenvolvido um microsserviço integrado ao portal de serviços, com o propósito

de solucionar as demandas anteriormente mencionadas. A implementação desse recurso teve como objetivo otimizar o gerenciamento dos atendimentos, reduzir a morosidade dos processos e aprimorar a eficiência operacional do sistema.

O sistema contempla o processo de atendimento ao cliente, abrangendo as etapas de agendamento, geração da senha de atendimento, direcionamento para a mesa correspondente e finalização da solicitação, permitindo a identificação de cada fase do processo e o encaminhamento eficiente para os atendimentos subsequentes. Ao ingressar no ambiente de atendimento, o cliente recebe uma senha gerada por um funcionário, a qual é registrada na base de dados e inserida na fila de espera, sendo posteriormente chamada na mesa de atendimento. Após a realização do atendimento, o sistema efetua o encerramento da solicitação ou, quando necessário, encaminha a demanda ao setor competente, assegurando o rastreamento das informações, a integridade do fluxo operacional e a redução do tempo de espera.

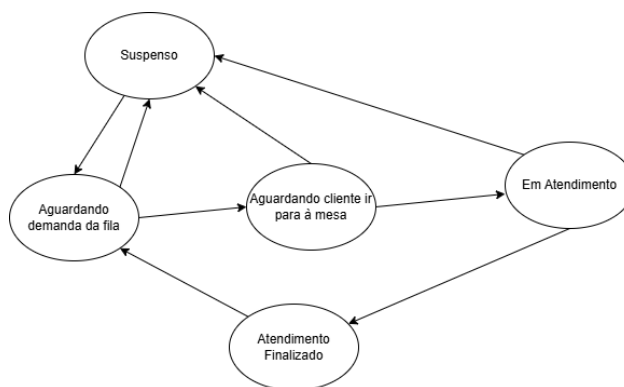
A problemática identificada no sistema anteriormente utilizado, atualmente em processo de substituição, está relacionada às dificuldades técnicas e ao elevado custo associados à implementação de novas funcionalidades, bem como à realização de manutenções evolutivas e corretivas. Tais limitações decorrem, principalmente, do fato de o sistema ter sido desenvolvido em linguagem Cobol, tecnologia considerada legada e que apresenta restrições quanto à flexibilidade, escalabilidade e integração com soluções mais modernas. Além disso, a escassez de profissionais especializados nessa linguagem contribui para o aumento dos custos operacionais e para a dependência de recursos específicos, impactando diretamente a agilidade na adaptação do sistema às novas demandas do negócio e às constantes evoluções tecnológicas.

O microsserviço integrado ao portal de serviços possibilita a centralização e a integração das principais demandas dos usuários, atuando como um mecanismo de apoio à resolução dos problemas anteriormente identificados no processo de atendimento. Desse modo, os tipos de serviços disponibilizados para atendimento são cadastrados e gerenciados pelos próprios funcionários, o que confere maior flexibilidade e autonomia na criação, alteração e manutenção desses serviços. Essa característica reduz a dependência de intervenções técnicas complexas, acelera a adaptação do sistema às novas necessidades institucionais e favorece a contínua evolução da solução proposta.

A solução proposta, com o objetivo de proporcionar um serviço de atendimento mais ágil e eficiente, evidenciou a definição de fluxos operacionais otimizados. A modelagem desses fluxos permite identificar etapas passíveis de melhoria, bem como direcionar os atendimentos de forma mais expedita, contribuindo para a redução do tempo de espera e para a racionalização dos processos.

Os estados do atendente, apresentados na Figura 1, representam a perspectiva operacional desse colaborador em relação às demandas dos clientes. Por meio dessa visualização, é possível identificar as diferentes situações do atendimento, tais como o atendimento em andamento, a conclusão da solicitação ou a necessidade de encaminhamento para outro setor, permitindo melhor controle do fluxo de trabalho e maior eficiência na gestão das demandas. Diante disso, a modelagem dos estados do atendente possibilita ao sistema realizar o acompanhamento em tempo real das atividades executadas, fornecendo subsídios para a supervisão e o balanceamento de acordo com as filas de prioridade conforme configurado pelo usuário.

Figura 1: Máquina de estados do atendente



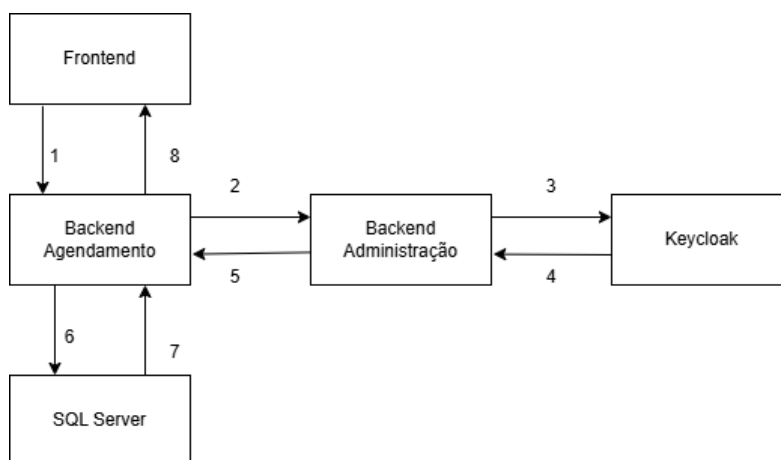
Fonte: Elaborado pelo autor (2026)

Embora essa funcionalidade já estivesse presente no sistema anteriormente utilizado, sua implementação na nova solução foi aprimorada por meio de uma abordagem mais atualizada, intuitiva e remodelada. As melhorias incorporadas visam otimizar a usabilidade, facilitar a interação dos usuários com o sistema e aumentar a eficiência na execução das atividades, contribuindo para um desempenho mais eficaz dos processos de atendimento.

Ademais, o redesenho dessa funcionalidade possibilitou a adequação do sistema às reais necessidades operacionais dos usuários, considerando aspectos como fluxo de navegação, organização das informações e redução de etapas desnecessárias. Essa reformulação contribuiu para a padronização dos procedimentos, diminuição de erros operacionais e maior agilidade na execução das tarefas, refletindo diretamente na qualidade do atendimento e na eficiência dos processos internos da organização.

A Figura 2 apresenta o fluxo de comunicação entre os principais componentes envolvidos no processo de autenticação e processamento das requisições do sistema. O diagrama ilustra a interação entre o frontend, os serviços de backend, o provedor de identidade Keycloak e o banco de dados SQL Server.

Figura 2: Fluxograma do Sistema



Fonte: Elaborado pelo autor (2026)

O processo inicia-se quando o frontend realiza uma requisição ao backend de agendamento (1), responsável

por receber as solicitações provenientes da interface do usuário e coordenar o processamento das funcionalidades relacionadas ao sistema de atendimento. Em todas as operações é exigida validação de identidade ou controle de acesso, o backend de agendamento encaminha a requisição ao backend de administração (2), que atua como intermediador no gerenciamento de autenticação e permissões.

Em seguida, o backend de administração realiza a comunicação com o Keycloak, enviando uma solicitação de validação das credenciais ou de verificação de autorização do usuário (3). O Keycloak, por sua vez, processa a autenticação e retorna as informações necessárias para confirmar a identidade e as permissões associadas ao usuário autenticado (4).

Após a validação, o backend de administração retorna a resposta ao backend de agendamento (5), permitindo que o processamento da requisição continue no sistema. Quando necessário, o backend de agendamento realiza operações de persistência ou consulta diretamente no banco de dados SQL Server (6), responsável pelo armazenamento das informações relacionadas aos atendimentos, agendamentos e demais dados operacionais da aplicação.

Após a execução da operação no banco de dados, os resultados retornam ao backend de agendamento (7), que realiza o tratamento final da resposta antes de encaminhá-la ao frontend (8), concluindo assim o ciclo completo da requisição.

## 4.2 Tecnologias utilizadas

Para a elaboração do projeto, foram utilizadas ferramentas que possibilitaram a construção integrada ao portal de serviços da empresa. Entre as tecnologias empregadas, destaca-se o .NET 6, que utiliza a linguagem C# como base para o desenvolvimento da aplicação. Também foi adotado o Vue.js 3.3.4, uma biblioteca JavaScript voltada à construção de interfaces de usuário e aplicações web interativas. Para o armazenamento, organização e gerenciamento dos dados, empregou-se o SQL Server, garantindo segurança e eficiência na manipulação das informações. O GitLab foi utilizado para o gerenciamento do código-fonte e controle de versionamento do projeto, enquanto o Docker foi aplicado com o intuito de simplificar a implantação do sistema, aumentar a portabilidade das aplicações e aprimorar a escalabilidade em ambientes corporativos e em nuvem.

O uso do .NET 6 foi fundamental para a construção da camada de backend, que exige processamento rápido, segurança e estabilidade. Essa tecnologia permitiu implementar APIs Representational State Transfer (REST) de forma padronizada e eficiente, facilitando a comunicação entre o portal e os demais sistemas corporativos. Vale resaltar a relevante compatibilidade nativa com contêineres Docker, que simplificou o processo de implantação e integração contínua. Dessa maneira, a maturidade da linguagem C# também contribuiu para maior produtividade e facilidade de manutenção do código.

Diante disso, a adoção do .NET 6 possibilitou a implementação de uma arquitetura modular e escalável, favorecendo a separação de responsabilidades entre as camadas da aplicação e o reaproveitamento de componentes. Essa abordagem contribui para a evolução contínua do sistema, permitindo a inclusão de novas funcionalidades com menor impacto sobre as estruturas já existentes. Do ponto de vista operacional, a combinação entre desempenho, segurança e suporte a práticas modernas de desenvolvimento reforça a confiabilidade da solução proposta e sua aderência às demandas do ambiente corporativo.

Em relação a utilização do Vue.js na construção da interface de usuário, permitiu a criação de telas leves, responsivas e de fácil interação. Sua estrutura baseada em componentes possibilitou o desenvolvimento de

funcionalidades reutilizáveis, facilitando a manutenção e evolução do sistema. Sendo assim, sua curva de aprendizado mais acessível e excelente integração com APIs REST tornou o processo de desenvolvimento mais ágil e consistente com os padrões modernos de aplicações web.

Além disso, o Vue.js contribuiu para a melhoria da experiência do usuário ao permitir a atualização dinâmica dos dados na interface sem a necessidade de recarregamento completo das páginas, proporcionando maior fluidez na navegação. A utilização de recursos como diretivas reativas e gerenciamento de estado favoreceu a consistência das informações exibidas, reduzindo erros de interação e aprimorando a comunicação entre a interface e a camada de backend. Dessa forma, a escolha dessa tecnologia mostrou-se adequada às necessidades do projeto, alinhando desempenho, usabilidade e escalabilidade.

O SQL Server foi escolhido para armazenar, organizar e gerenciar todas as informações necessárias à aplicação. Por sua robustez, ele garante integridade e segurança aos dados, enquanto recursos como índices contribuem para consultas mais rápidas e eficientes. Ademais, o uso do SQL Server está alinhado ao padrão tecnológico já adotado pela empresa, o que facilita tanto a integração quanto a administração do sistema.

Diante disso, o SQL Server oferece mecanismos avançados de controle transacional, como suporte a transações que garantem atomicidade, consistência, isolamento e durabilidade (ACID), procedimentos armazenados e triggers, os quais contribuem para a consistência e a confiabilidade das operações realizadas pela aplicação. Esses recursos permitem a implementação de regras de negócio diretamente na camada de dados, reduzindo redundâncias e fortalecendo a integridade das informações. Dessa forma, a escolha do SQL Server reforça a estabilidade do sistema e assegura a escalabilidade necessária para acompanhar o crescimento das demandas operacionais.

Por conseguinte o uso do GitLab desempenhou papel central no gerenciamento do código-fonte, garantindo histórico de alterações, rastreabilidade, segurança e trabalho colaborativo entre os desenvolvedores. Com isso, seus pipelines de integração contínua (CI) possibilitaram automatizar testes, validações e deploys, aumentando a qualidade do software entregue e reduzindo erros humanos no processo de implantação.

Vale ressaltar que, a utilização do GitLab favoreceu a padronização dos fluxos de desenvolvimento por meio do controle de versões e da adoção de boas práticas, como revisão de código e gerenciamento de branches. Esses recursos contribuíram para a detecção antecipada de inconsistências, facilitaram a manutenção evolutiva da aplicação e proporcionaram maior confiabilidade ao ciclo de vida do software, alinhando o processo de desenvolvimento às exigências de qualidade e segurança do ambiente corporativo.

Em relação ao uso do Docker, ele facilitou a criação de ambientes consistentes para desenvolvimento, homologação e produção. Isso reduziu problemas decorrentes de diferenças de configuração entre servidores e melhorou a escalabilidade da aplicação, permitindo que novos serviços fossem disponibilizados rapidamente. Dessa forma, a combinação do Docker com os pipelines do GitLab padronizou e agilizou o processo de entrega contínua.

Ainda sobre a adoção do Docker, possibilitou o isolamento adequado dos serviços e de suas dependências, garantindo maior previsibilidade no comportamento da aplicação em diferentes ambientes. Essa abordagem favoreceu a portabilidade do sistema, facilitando sua implantação em infraestruturas distintas e contribuindo para a otimização do uso de recursos computacionais. Como resultado, a solução alcançou maior confiabilidade operacional e flexibilidade para futuras expansões, alinhando-se às boas práticas de arquiteturas modernas baseadas em contêineres.

O Keycloak foi utilizado para garantir a segurança do acesso ao sistema, centralizando o processo de

autenticação dos usuários e simplificando o gerenciamento de permissões. Ele permitiu integrar o portal de forma segura ao ecossistema corporativo, além de facilitar a aplicação de políticas de segurança como renovação de tokens, autenticação multifator e controle de sessões. A utilização do Keycloak trouxe maior padronização e robustez ao controle de identidade dentro da aplicação.

Além disso, o Keycloak contribuiu para o fortalecimento da governança de segurança ao possibilitar a definição de perfis de acesso baseados em papéis (roles), garantindo que cada usuário visualize e execute apenas as funcionalidades compatíveis com suas atribuições. Essa abordagem reduziu riscos associados a acessos indevidos e facilitou a auditoria das ações realizadas no sistema, promovendo maior conformidade com as boas práticas de segurança da informação e com os requisitos institucionais de controle e rastreabilidade.

## **4.2.1 Arquitetura de microsserviços**

A arquitetura de microsserviços consiste em estruturar uma aplicação como um conjunto de pequenos serviços independentes, cada um responsável por uma funcionalidade específica e capaz de operar, ser implantado e evoluir de forma autônoma. Esse modelo se diferencia da abordagem monolítica por permitir maior modularidade, escalabilidade e resiliência, já que cada serviço comunica-se com os demais por meio de APIs padronizadas e pode utilizar tecnologias, bancos de dados e linguagens distintas conforme sua necessidade. No contexto deste projeto o sistema foi desenvolvido como um microsserviço por integrar um portal de serviços maior, exigindo que sua funcionalidade pudesse operar de forma isolada, sem impactar outros módulos do portal. Essa escolha arquitetural facilita a manutenção, permite atualizações individuais, reduz riscos de indisponibilidade global e garante que apenas os componentes mais demandados sejam escalados, contribuindo para um ambiente mais flexível, robusto e adequado à expansão contínua do portal.

Diante desse cenário a adoção da arquitetura de microsserviços favorece a manutenção e a evolução contínua da aplicação, uma vez que alterações ou atualizações em um serviço específico podem ser realizadas sem impactar o funcionamento dos demais componentes do sistema. Sendo assim, essa abordagem contribui para o balanceamento de carga e para a tolerância a falhas, permitindo que eventuais indisponibilidades sejam isoladas e tratadas de forma controlada, o que resulta em maior disponibilidade, desempenho e confiabilidade da solução como um todo.

## **4.2.2 Padrão CQRS**

No contexto da solução desenvolvida, adotou-se uma abordagem inspirada no padrão arquitetural Command Query Responsibility Segregation (CQRS), o qual propõe a separação conceitual entre as operações responsáveis pela modificação do estado do sistema (comandos) e aquelas destinadas exclusivamente à recuperação de informações (consultas). Essa estratégia mostrou-se adequada diante da complexidade do domínio do atendimento ao público, no qual coexistem operações críticas de escrita, como a geração, o encaminhamento e o encerramento de atendimentos, bem como múltiplos fluxos de leitura voltados ao acompanhamento das filas, dos estados dos atendentes e à exibição de informações em painéis eletrônicos.

A aplicação dos princípios do CQRS possibilitou uma organização mais clara das responsabilidades da aplicação, contribuindo para a redução do acoplamento entre as camadas de leitura e escrita. As operações de comando concentram-se nas regras de negócio relacionadas à gestão dos atendimentos, assegurando a consistência e a integridade dos dados, enquanto as consultas foram estruturadas e otimizadas para atender às

demandas de visualização e monitoramento, muitas vezes exigindo respostas rápidas e agregações específicas de informações.

Essa separação favoreceu a escalabilidade e a manutenibilidade do sistema, uma vez que as demandas de leitura puderam ser atendidas de forma eficiente, sem impactar negativamente o desempenho das operações de escrita. Desta forma, a adoção do CQRS facilitou a evolução do sistema, permitindo ajustes e melhorias em cada fluxo de forma independente, de acordo com as necessidades operacionais e de negócio.

### 4.2.3 Padrão Unit of Work

No desenvolvimento da solução proposta, foi adotado o padrão arquitetural *Unit of Work*, cujo objetivo é gerenciar de forma centralizada o ciclo de vida das operações realizadas sobre os dados, garantindo que múltiplas alterações sejam tratadas como uma única unidade transacional. Esse padrão mostrou-se especialmente relevante em um contexto no qual diversas operações relacionadas ao atendimento ao público precisam ser executadas de forma consistente, como a atualização do estado do atendimento, o registro de históricos e o encaminhamento de demandas entre setores.

A aplicação do Unit of Work possibilitou o controle das transações de maneira mais segura e organizada, assegurando que todas as operações realizadas durante um fluxo de negócio fossem confirmadas ou revertidas de forma conjunta. Dessa forma, caso ocorra alguma falha durante o processamento, o sistema é capaz de manter a integridade dos dados, evitando estados inconsistentes ou parcialmente persistidos na base de dados.

O uso desse padrão contribuiu para a separação de responsabilidades na camada de acesso a dados, reduzindo o acoplamento entre a lógica de negócio e os mecanismos de persistência. Ao centralizar o controle das transações, o *Unit of Work* facilita a manutenção do código, melhora a legibilidade das operações e favorece a reutilização de componentes, tornando a aplicação mais robusta e alinhada às boas práticas de engenharia de software. Essa abordagem também se mostrou compatível com a arquitetura de microsserviços adotada, reforçando a confiabilidade e a previsibilidade das operações realizadas pelo sistema.

### 4.2.4 Keycloak

O Keycloak é uma plataforma de código aberto voltada ao gerenciamento de identidade e controle de acesso, amplamente utilizada para implementar mecanismos de autenticação e autorização em aplicações modernas. Desenvolvido pela Red Hat, o Keycloak permite centralizar o gerenciamento de usuários, credenciais e permissões, oferecendo suporte a diversos protocolos de segurança amplamente adotados na web, como OAuth 2.0, OpenID Connect e SAML (RED HAT, 2023).

A utilização de ferramentas de gerenciamento de identidade torna-se fundamental em ambientes corporativos, especialmente em sistemas que exigem controle rigoroso de acesso e integração entre múltiplos serviços. Nesse contexto, o Keycloak atua como um provedor de identidade (Identity Provider – IdP), sendo responsável por autenticar os usuários e emitir tokens de acesso utilizados pelas aplicações para validar permissões e garantir a segurança das operações realizadas.

No projeto desenvolvido, o Keycloak foi utilizado como mecanismo central de autenticação do sistema, permitindo a integração segura entre o portal de serviços e os demais módulos da aplicação. Por meio dessa abordagem, o processo de autenticação foi externalizado da lógica da aplicação, possibilitando maior padronização

e simplificação do gerenciamento de usuários e sessões.

Além disso, o Keycloak possibilita a implementação de políticas de segurança avançadas, como autenticação multifator, gerenciamento de sessões, renovação automática de tokens e controle de acesso baseado em papéis (Role-Based Access Control – RBAC). Essas funcionalidades contribuem para reduzir riscos relacionados a acessos indevidos e fortalecer a governança de segurança da informação dentro do ambiente corporativo.

Outro benefício relevante está na possibilidade de integração com diferentes aplicações e serviços, permitindo que múltiplos sistemas utilizem um mesmo mecanismo de autenticação centralizado. Essa característica favorece a implementação de soluções de Single Sign-On (SSO), nas quais o usuário realiza login uma única vez e passa a acessar diferentes sistemas sem a necessidade de novas autenticações.

Dessa forma, a adoção do Keycloak contribui para o aumento da segurança, da escalabilidade e da padronização do processo de autenticação, tornando-se uma solução adequada para sistemas corporativos que demandam controle eficiente de identidades e acessos.

## 4.2.5 WebSocket

A utilização de comunicação em tempo real por meio do protocolo WebSocket desempenhou papel fundamental no funcionamento do sistema desenvolvido. Essa tecnologia possibilitou a atualização imediata das informações exibidas nos painéis eletrônicos, eliminando a necessidade de consultas periódicas ao servidor para verificação de mudanças de estado.

Por meio do WebSocket, eventos relevantes do sistema, como a geração de uma nova senha, a alteração do estado de um atendimento, o início ou encerramento de chamadas e o redirecionamento de clientes, são propagados instantaneamente aos usuários. Essa abordagem garante maior sincronização entre os diferentes pontos de visualização do sistema, proporcionando uma experiência mais fluida e coerente tanto para os atendentes quanto para os usuários finais.

A comunicação em tempo real contribuiu significativamente para a eficiência operacional da solução, reduzindo a latência na disseminação das informações e melhorando a percepção de agilidade do sistema. Além disso, o uso de WebSocket mostrou-se adequado para ambientes com múltiplos acessos simultâneos, assegurando desempenho, confiabilidade e atualização contínua das informações, características essenciais para aplicações corporativas voltadas ao atendimento ao público.

A implementação prática dessa técnica é detalhada na Listagem 1, que ilustra a lógica de consumo do Hub no lado do cliente. O trecho de código demonstra o uso da função *invoke* para chamar procedimentos remotos no servidor, como *obter-ultimas-senhas* e *obter-painel-eletronico*, tratando as respostas por meio de *Promises* para atualizar o estado da aplicação (ex: *this.senhaAtual*, *this.layout*). Também é apresentada a configuração de ouvintes de eventos através do método *on*, permitindo que o frontend reaja em tempo real a notificações enviadas pelo servidor, como *nova-senha* e *interromper-leitura*.

Listagem 1: Consumo de Métodos do Hub no Frontend

```
1 obterUltimasSenhas() {
2   this.painelEletronicoHub.connection.invoke('obter-ultimas-senhas')
3   .then((senhas) => {
4     if (senhas.length > 0) this.senhaAtual = senhas[0];
5     for (let i = 1; i <= this.quantidadeSenhasHistorico; i++) {
6       this.ultimasSenhas.push(i < senhas.length ? senhas[i] : { codigo: '', mesa:
```

```

        '' });
7     }
8     });
9 },
10 obterPainelEletronico() {
11     this.painelEletronicoHub.connection.invoke('obter-painel-eletronico').then((painel)
12         => {
13         this.layout = painel.layoutPainel;
14         this.toque = this.toques[painel.estiloToque];
15         this.quantidadeSenhasHistorico = painel.quantidadeSenhasHistorico;
16         this.nomePostoAtendimento = painel.nomePostoAtendimento;
17         this.quantidadeRechamadaSenha = painel.quantidadeRechamadaSenha;
18         this.tempoRetardoRechamadaPainelSegundos = painel.
19             tempoRetardoRechamadaPainelSegundos;
20     });
21 },
22 obterMensagensPostoAtendimento() {
23     this.painelEletronicoHub.connection.invoke('obter-mensagens').then((mensagensDtos) =>
24     {
25     const mensagens = mensagensDtos
26         .filter((x) => x.tipo == 0)
27         .map((mensagem) => ({
28             tipo: 'mensagem',
29             titulo: mensagem.mensagemTexto,
30             texto: mensagem.textoInformativo,
31         }));
32     this.mensagens.push(...mensagens);
33     });
34 },
35 configurarEventos() {
36     this.painelEletronicoHub.connection.on('nova-senha', (senha) => {
37         this.inserirEmEspera(senha);
38     });
39     this.painelEletronicoHub.connection.on('interromper-leitura', (codigo) => {
40         if (codigo == this.senhaAtual.codigo)
41             this.interromper = true;
42     }
43     );
44 },

```

A Listagem 2 apresenta a implementação de um Hub do SignalR no ambiente .NET. O código detalha os métodos de ciclo de vida da conexão, como *OnConnectedAsync* e *OnDisconnectedAsync*, que gerenciam a entrada e saída de usuários em grupos específicos baseados no identificador *postoAtendimentoId*. Ademais, observam-se métodos decorados com o atributo *[HubMethodName]*, como *ObterMensagens* e *ObterUltimas-Senhas*, que utilizam o padrão Mediator via biblioteca MediatR para realizar consultas assíncronas ao banco de dados e retornar objetos de transferência de dados (DTOs) ao cliente.

#### Listagem 2: Métodos do Hub de Painel Eletrônico no Backend

```

1 public override async Task OnConnectedAsync()
2 {
3     var postoAtendimentoId = Context.GetHttpContext().Request.Query[
4         PostoAtendimentoIdQueryKey];
5     await AssistirPainelEletronico(postoAtendimentoId);

```

```

5     await base.OnConnectedAsync();
6 }
7 public override async Task OnDisconnectedAsync(Exception? exception)
8 {
9     var postoAtendimentoId = Context.GetHttpContext()!.Request.Query[
10         PostoAtendimentoIdQueryKey];
11     await AssistirPainelEletronico(postoAtendimentoId);
12     await PararAssistirPainelEletronico(postoAtendimentoId);
13     await base.OnDisconnectedAsync(exception);
14 }
15 private async Task AssistirPainelEletronico(string postoAtendimentoId)
16 {
17     await Groups.AddToGroupAsync(Context.ConnectionId, postoAtendimentoId);
18 }
19 private async Task PararAssistirPainelEletronico(string postoAtendimentoId)
20 {
21     await Groups.RemoveFromGroupAsync(Context.ConnectionId, postoAtendimentoId);
22 }
23 [HubMethodName("obter-mensagens")]
24 public async Task<List<ObterMensagensPorPostoAtendimentoDto>> ObterMensagens()
25 {
26     return await _mediator.Send(new ObterMensagensPorPostoAtendimentoIdQuery(new Guid(
27         Context.GetHttpContext()!.Request.Query[PostoAtendimentoIdQueryKey])););
28 }
29 [HubMethodName("obter-painel-eletronico")]
30 public async Task<PainelEletronicoDto> ObterPainelEletronico()
31 {
32     var postoId = new Guid(Context.GetHttpContext()!.Request.Query[
33         PostoAtendimentoIdQueryKey]);
34     return await _mediator.Send(new ObterPainelEletronicoPorPostoAtendimentoIdQuery(
35         postoId));
36 }
37 [HubMethodName("obter-ultimas-senhas")]
38 public async Task<List<SenhaPainelEletronicoDto>> ObterUltimasSenhas()
39 {
40     var postoId = new Guid(Context.GetHttpContext()!.Request.Query[
41         PostoAtendimentoIdQueryKey]);
42     return await _mediator.Send(new ObterUltimasSenhasPostoAtendimentoQuery(postoId));
43 }

```

## 4.2.6 Jsreport

Para atender à necessidade de geração de relatórios dinâmicos e personalizáveis, a solução desenvolvida incorporou a ferramenta jsreport, uma plataforma especializada na criação e renderização de documentos a partir de dados estruturados. Sua utilização possibilitou a geração de relatórios em formatos como PDF, permitindo a apresentação organizada e padronizada das informações relacionadas aos atendimentos, filas, históricos e indicadores do sistema.

Do ponto de vista técnico, o jsreport foi integrado ao sistema por meio de requisições ao serviço responsável pela geração dos documentos, recebendo os dados provenientes da camada de consultas da aplicação. Essa integração está alinhada aos princípios de separação de responsabilidades, uma vez que a lógica de geração

e formatação dos relatórios permanece desacoplada das regras de negócio centrais do sistema. Dessa forma, alterações nos modelos de relatório não impactam diretamente o código responsável pelo processamento dos atendimentos, contribuindo para a manutenibilidade e a evolução contínua da solução.

Em relação à utilização do jsreport, ela favoreceu a padronização visual dos documentos gerados e garantiu maior confiabilidade na apresentação das informações, aspecto fundamental em um ambiente corporativo. A possibilidade de versionamento dos templates e de reutilização de componentes visuais contribuiu para a consistência dos relatórios, ao mesmo tempo em que facilitou a incorporação de novas demandas, reforçando a adequação da ferramenta às necessidades do projeto e às expectativas dos usuários finais.

#### **4.2.7 Ferramenta de apoio à gestão**

No contexto do projeto, diversas ferramentas de apoio à gestão foram adotadas com o objetivo de melhorar a organização dos processos, facilitar a comunicação entre os membros da equipe e garantir maior controle sobre as entregas. O Google Drive foi utilizado como repositório central para armazenamento e organização dos artefatos, permitindo que documentos, relatórios e materiais de suporte fossem compartilhados e atualizados de forma colaborativa. Essa estrutura contribuiu para a padronização das entregas e para o fácil acesso aos materiais por todos os envolvidos. Em relação ao acompanhamento das atividades de desenvolvimento, o GitLab desempenhou um papel fundamental. A plataforma foi utilizada tanto para o gerenciamento do repositório de código quanto para a visualização do board de tarefas, o que permitiu acompanhar o progresso de cada atividade de forma clara e estruturada. Dessa forma, o GitLab forneceu suporte para a implementação de pipelines de Continuous Integration/Continuous Delivery (CI/CD), possibilitando a automação de etapas importantes, como build, testes e deploy, aumentando a confiabilidade e a eficiência do processo de desenvolvimento. As reuniões diárias de acompanhamento (dailies), essenciais para o alinhamento contínuo da equipe, garantindo comunicação síncrona, organizada e acessível a todos os integrantes. Esse formato possibilitou que cada membro apresentasse seu progresso, dificuldades e próximos passos, promovendo transparência e colaboração.

### **4.3 Contribuição**

As principais contribuições do projeto concentraram-se no aprimoramento das funcionalidades relacionadas à gestão e ao acompanhamento dos atendimentos. Entre elas, destacam-se o gerenciamento das mesas de atendimento e o gerenciamento de agendamentos, que possibilitam uma melhor organização do fluxo de serviços. Por conseguinte, foram implementados recursos voltados à parametrização e ao gerenciamento das mensagens exibidas nos painéis eletrônicos, bem como à visualização desses painéis em tempo real. O projeto também contemplou o monitoramento das filas de atendimento e o acompanhamento das senhas por atendente, permitindo maior controle operacional e contribuindo para a eficiência e transparência do processo de atendimento ao público.

Adicionalmente, o projeto proporcionou melhorias significativas no suporte à tomada de decisão gerencial, por meio da disponibilização de informações consolidadas e atualizadas sobre o desempenho dos atendimentos. A possibilidade de acompanhar indicadores operacionais, como tempo médio de espera, volume de atendimentos por setor e status das filas, contribuiu para a identificação de gargalos e para o planejamento de ações corretivas. Dessa forma, as funcionalidades implementadas ampliam a capacidade de gestão dos serviços, promovendo maior eficiência operacional e qualidade no atendimento aos usuários.

Durante sua execução, houve também atuação como ponto de apoio técnico aos demais integrantes da equipe, auxiliando no esclarecimento de dúvidas e oferecendo orientação em atividades de maior complexidade, contribuindo assim para o alinhamento e a qualidade das entregas do time.

Além disso, essa atuação colaborativa favoreceu a disseminação de conhecimentos técnicos e a padronização das práticas de desenvolvimento adotadas ao longo do projeto. O compartilhamento de experiências e soluções contribuiu para a mitigação de erros recorrentes, o aumento da produtividade da equipe e o fortalecimento do trabalho integrado, refletindo positivamente na consistência do código, no cumprimento dos prazos estabelecidos e na qualidade final da solução desenvolvida.

### 4.3.1 Descrição e análise dos desenvolvidos

A Listagem 3 apresenta um trecho da classe `PostoAtendimento` que é responsável pelo cálculo dos horários disponíveis. O método `CalcularHorariosRestantes` tem como objetivo identificar os intervalos de tempo ainda disponíveis para agendamento em uma data específica, considerando as regras de funcionamento previamente configuradas.

Listagem 3: Método da classe `PostoAtendimento` que obtém os horários livres

```
1 public List<HorarioAtendimento> CalcularHorariosRestantes(IEnumerable<
2     AgendamentoPresencial> agendamentos, DateTime data)
3 {
4     var resultado = new List<HorarioAtendimento>();
5     var cronograma = CronogramasAtendimento
6         .Find(x => x.Status == CronogramaStatus.Ativo &&
7             x.DataInicio <= data.Date &&
8             x.DataFim >= data.Date);
9     var feriados = EventosCalendario
10        .SelectMany(x => x.DataInicio.ObterDatasPeriodo(x.DataFim));
11     var parametro = cronograma?.ParametrosHorario.FirstOrDefault();
12     var diaSemana = ((int)data.DayOfWeek + 1).ToString();
13     if (cronograma == null ||
14         parametro == null ||
15         feriados.Any(x => x.Date == data.Date) ||
16         !parametro.DiasDaSemana.Contains(diaSemana))
17     {
18         return resultado;
19     }
20     var horaAtual = parametro.HorarioInicial;
21     while (horaAtual <= parametro.HorarioFinal)
22     {
23         var marcacoes = agendamentos
24             .Count(x => x.DataAgendamento == data.Date + horaAtual);
25         var atendentes = parametro.Redutores
26             .Find(x => x.HorarioInicial <= horaAtual &&
27                 x.HorarioFinal >= horaAtual &&
28                 x.DiasDaSemana.Contains(diaSemana))?.QuantidadeAtendentes ?? parametro
29             .QuantidadeAtendentes;
30         if (marcacoes < atendentes)
31         {
32             resultado.Add(new HorarioAtendimento
33                 {
34                     DataInicio = data.Date + horaAtual,
35                     DataFim = data.Date + horaAtual + 1,
36                     QuantidadeAtendentes = atendentes - marcacoes,
37                     Status = CronogramaStatus.Livre
38                 });
39             horaAtual++;
40         }
41     }
42     return resultado;
43 }
```

```

32         Horario = horaAtual,
33         Vagas = atendentes - marcacoes
34     });
35 }
36     horaAtual += parametro.Intervalo;
37 }
38     return resultado;
39 }

```

Inicialmente, o método valida se existe um cronograma ativo para a data informada, verificando o período de vigência do cronograma, a ocorrência de feriados e a compatibilidade com o dia da semana. Caso alguma dessas condições não seja atendida, o método encerra sua execução retornando uma lista vazia de horários disponíveis.

Em seguida, são obtidos os parâmetros de horário do posto de atendimento, incluindo o horário inicial, o horário final e o intervalo entre atendimentos. A partir dessas informações, o método percorre os horários possíveis dentro do período de funcionamento, contabilizando os agendamentos já existentes para cada intervalo.

Para cada horário analisado, é realizada a verificação da quantidade máxima de atendentes disponíveis, considerando possíveis redutores aplicáveis ao dia da semana e ao intervalo de horário. Quando o número de agendamentos é inferior à capacidade de atendimento, o horário é considerado disponível e incluído na lista de resultados, juntamente com a quantidade de vagas restantes.

Ao final do processamento, o método retorna uma lista contendo apenas os horários válidos para novos agendamentos, garantindo que as regras operacionais do posto de atendimento sejam respeitadas.

A Listagem 4 apresenta o método responsável pelo monitoramento da fila de atendimento em um posto de atendimento presencial. O método *Handle* implementa a lógica de obtenção e filtragem das senhas emitidas, retornando informações consolidadas para acompanhamento em tempo real do fluxo de atendimento.

#### Listagem 4: Relatório de atendimentos realizados

```

1 public async Task<List<MonitoramentoFilaAtendimentoDto>> Handle(
2     ObterMonitoramentoFilaAtendimentoQuery request, CancellationToken cancellationToken)
3 {
4     var resultados = await _repositorioSenhas.ObterTodos(x =>
5         request.PostoAtendimentoId == x.PostoAtendimentoId &&
6         x.DataHoraEmissao.Date == DateTime.Now.Date &&
7         (!request.ServicoAtendimentoIds!.Any() || request.ServicoAtendimentoIds!.
8             Contains(x.ServicoAtendimentoId!.Value)) &&
9         (!request.AtendentesIds!.Any() || request.AtendentesIds!.Contains(x.
10             AtendenteId!.Value)) &&
11         (!request.MesasAtendimentosIds!.Any() || request.MesasAtendimentosIds!.
12             Contains(x.Atendimento.Expediente.MesaAtendimentoId)) &&
13         (string.IsNullOrWhiteSpace(request.SenhaGerada) || x.SenhaGerada.Contains
14             (request.SenhaGerada)) &&
15         (!request.Status.HasValue || x.Status == request.Status) &&
16         (!request.TipoSenha.HasValue || x.TipoSenha == request.TipoSenha) &&
17         (!request.HorarioInicial.HasValue || x.DataHoraEmissao >= request.
18             HorarioInicial) &&
19         (!request.HorarioFinal.HasValue || x.DataHoraEmissao <= request.
20             HorarioFinal),
21         null,

```

```

15         new List<string> { "ServicoAtendimento", "AtendenteTriagem.
16             Expedientes", "Atendimento.Expediente" });
17     var senhasIds = resultados.Select(x => x.FilaPostoSenhaId).ToList();
18     var suspensoes = await _repositorioSuspensoes.ObterTodos(x => senhasIds.Contains(x.
19         FilaPostoSenhaId!.Value));
20     var dtos = resultados.OrderByDescending(x => x.DataHoraEmissao)
21         .DistinctBy(x => x.FilaPostoSenhaId)
22         .Select(x => new MonitoramentoFilaAtendimentoDto
23     {
24         FilaPostoSenhaId = x.FilaPostoSenhaId,
25         SenhaGerada = x.SenhaGerada,
26         NomeServico = x.ServicoAtendimento?.NomeServico,
27         HoraChegada = x.DataHoraEmissao.ToString("HH:mm"),
28         TempoEspera = x.ObterTempoEspera(),
29         TempoEsperaTotal = x.ObterTempoEsperaTotal(),
30         HoraRedirecionada = FilaPostoSenha.ObterHorarioRedirecionamento
31         (suspensoes.FirstOrDefault(suspensao => x.FilaPostoSenhaId == suspensao.
32             FilaPostoSenhaId)?.DataHoraSuspensao),
33     }).ToList();
34     return dtos;
35 }

```

Inicialmente, o método realiza uma consulta assíncrona ao repositório de senhas, aplicando filtros baseados no identificador do posto de atendimento e na data atual. Além disso, são considerados critérios opcionais definidos na requisição, como serviço de atendimento, atendente responsável, mesa de atendimento, número da senha, status, tipo de senha e intervalo de horário de emissão. Esses filtros permitem uma consulta flexível e parametrizável, adequada a diferentes cenários de monitoramento.

Após a obtenção das senhas, o método identifica os registros relacionados a suspensões de atendimento, associando essas informações às respectivas senhas emitidas. Em seguida, os resultados são ordenados pela data e hora de emissão de forma decrescente, garantindo que as informações mais recentes sejam priorizadas.

Na etapa final, os dados são transformados em DTO, contendo informações relevantes para o monitoramento da fila, tais como identificação da senha, serviço associado, horário de chegada, tempo de espera individual, tempo total de espera e eventual redirecionamento decorrente de suspensão. O método então retorna a lista de dados consolidados, que pode ser utilizada por camadas superiores do sistema para exibição ou análise do atendimento.

Dessa forma, o método contribui para o acompanhamento eficiente da fila de atendimento, fornecendo subsídios para a gestão operacional do posto de atendimento e para a tomada de decisões em tempo real.

A Listagem 5 apresenta o módulo responsável pelo controle da chamada e anúncio das senhas no painel eletrônico. O conjunto de métodos implementa a lógica de atualização da senha em atendimento, gerenciamento da fila de espera e anúncio automático das senhas por meio de recursos sonoros e de síntese de voz.

#### Listagem 5: Métodos de controle do painel eletrônico

```

1     methods: {
2         atualizarSenha(senha) {
3             this.ultimasSenhas.unshift(this.senhaAtual);
4             this.ultimasSenhas.pop();
5             this.senhaAtual = senha;
6         },

```

```

7   inserirEmEspera(senha) {
8       this.senhasEmEspera.push(senha);
9       this.lerProximaSenha();
10  },
11  lerProximaSenha() {
12      if (this.senhasEmEspera.length < 1 || this.lendo) return;
13      this.lendo = true;
14      this.interromper = false;
15      this.atualizarSenha(this.senhasEmEspera.shift());
16      let repeticoes = 0;
17      this.anunciarSenhaAtual();
18      const intervalo = setInterval(() => {
19          if (this.interromper || repeticoes == this.quantidadeRechamadaSenha) {
20              this.interromper = false;
21              this.lendo = false;
22              clearInterval(intervalo);
23              this.lerProximaSenha();
24              return;
25          }
26          this.anunciarSenhaAtual();
27          repeticoes++;
28      }, this.tempoRetardoRechamadaPainelSegundos * 1000);
29  },
30  anunciarSenhaAtual() {
31      const audio = document.getElementById('campainha');
32      audio.src = this.toque;
33      audio.play();
34      const msg = new SpeechSynthesisUtterance(`senha: \${this.senhaAtual.codigo}
35          dirigir-se a mesa \${this.senhaAtual.mesa}`);
36      msg.rate = 0.9;
37      msg.lang = 'pt-BR';
38      setTimeout(() => {
39          window.speechSynthesis.cancel();
40          window.speechSynthesis.speak(msg);
41      }, 1000);
42  },

```

O método *atualizarSenha* tem a função de manter o histórico das últimas senhas chamadas, atualizando a senha atualmente em atendimento e garantindo a preservação das informações recentes para exibição nos painéis do posto. Em paralelo, o método *inserirEmEspera* é responsável por adicionar novas senhas à fila de espera e acionar o processo de leitura da próxima senha disponível.

O método *lerProximaSenha* gerencia o fluxo de chamadas de forma sequencial, assegurando que não ocorram sobreposições durante o anúncio das senhas. Para isso, são utilizados mecanismos de controle que impedem múltiplas execuções simultâneas e definem o número máximo de repetições de chamada, respeitando o intervalo de tempo configurado entre cada anúncio.

Por fim, o método *anunciarSenhaAtual* realiza a emissão sonora da senha em atendimento, utilizando um sinal audível e a funcionalidade de síntese de voz do navegador para informar o código da senha e a mesa de atendimento correspondente. Essa abordagem contribui para a acessibilidade e para a organização do fluxo de atendimento, facilitando a comunicação com os usuários presentes no posto de atendimento.

Dessa forma, o módulo apresentado possibilita a automação do processo de chamada de senhas, garantindo

maior eficiência operacional e melhor experiência aos usuários do sistema.

## 5 Dificuldades encontradas

Durante a execução do projeto, foram identificadas diversas dificuldades relacionadas à composição e à estabilidade da equipe envolvida. A rotatividade de profissionais configurou-se como um dos principais desafios, uma vez que alguns integrantes permaneceram por períodos reduzidos, comprometendo a continuidade das atividades, a transferência de conhecimento e a consolidação da curva de aprendizado coletiva. Além disso, o número reduzido de colaboradores, em sua maioria com baixo nível de senioridade técnica, demandou maior acompanhamento por parte da equipe mais experiente, bem como tempo adicional para a realização de determinadas tarefas.

Outro fator que contribuiu para a complexidade do processo foi a substituição do analista de requisitos em duas ocasiões distintas ao longo do desenvolvimento. Essa instabilidade impactou diretamente a definição e a manutenção do escopo funcional do sistema, ocasionando lacunas e ambiguidades em alguns requisitos previamente levantados. Como consequência, determinadas funcionalidades precisaram ser melhor especificadas apenas durante a fase de implementação, o que resultou em retrabalho e na necessidade de revisões pontuais no planejamento originalmente estabelecido.

Adicionalmente, observou-se que parte dos requisitos não estava suficientemente clara ou documentada de forma objetiva, especialmente no que se refere à geração de relatórios. Em alguns casos, o cliente não conseguiu explicitar de maneira precisa quais informações deveriam ser apresentadas, tampouco os critérios de formatação e agrupamento dos dados. Essa falta de clareza dificultou a validação das entregas intermediárias e exigiu múltiplas interações para alinhamento das expectativas, prolongando o tempo de desenvolvimento dessas funcionalidades.

Somado a isso, as constantes mudanças no responsável pela análise de requisitos contribuíram para a perda de histórico decisório e de entendimento do contexto do negócio. Tal cenário reforçou a necessidade de revisões frequentes e de reinterpretação das demandas, aumentando a complexidade do processo de desenvolvimento. Apesar dessas dificuldades, foram adotadas estratégias de comunicação contínua e validações incrementais junto ao cliente, de modo a mitigar os impactos negativos e assegurar a aderência da solução final às necessidades identificadas.

Além das dificuldades já mencionadas, também se observou a necessidade de ajustes contínuos no processo de comunicação entre a equipe técnica e o cliente. Em diversos momentos, as informações fornecidas eram incompletas ou excessivamente genéricas, o que dificultava a correta interpretação das necessidades reais do sistema. Essa limitação tornou imprescindível a realização de reuniões adicionais de alinhamento e validação, bem como a apresentação de protótipos e versões preliminares das funcionalidades, especialmente no caso dos relatórios, para que o cliente pudesse visualizar os resultados e, a partir disso, refinar suas solicitações. Embora essas ações tenham contribuído para o amadurecimento dos requisitos ao longo do projeto, elas também impactaram o tempo de desenvolvimento e evidenciaram a importância de um levantamento de requisitos mais estruturado e consistente desde as fases iniciais.

## **6 Impactos da sua formação no seu trabalho**

A formação acadêmica exerceu influência significativa nas atividades desempenhadas ao longo do projeto, não apenas do ponto de vista técnico, mas também no desenvolvimento de competências profissionais. Em razão do maior nível de senioridade em relação aos demais integrantes da equipe, tornou-se necessário assumir responsabilidades adicionais, como a condução de tarefas de maior complexidade, a tomada de decisões técnicas estratégicas e o apoio contínuo aos colegas. Essa atuação favoreceu o aprimoramento de habilidades relacionadas à liderança técnica, comunicação interpessoal, cooperação em equipe e gestão de problemas em um ambiente de desenvolvimento colaborativo.

Os conhecimentos adquiridos ao longo do curso mostraram-se fundamentais para a execução das diferentes etapas do projeto, especialmente no que se refere à análise e à resolução de problemas computacionais complexos. As disciplinas de Algoritmos e Estruturas de Dados e de Laboratório de Programação forneceram a base necessária para o desenvolvimento do algoritmo de verificação de disponibilidade de horários para cada dia da semana, considerando variáveis como a quantidade de atendentes disponíveis, os períodos de indisponibilidade, os múltiplos intervalos ao longo do dia e as regras específicas de funcionamento do sistema. Esses fatores influenciam diretamente o processo de marcação de agendamentos, a configuração do painel eletrônico e a definição de regras de negócio mais elaboradas, exigindo soluções eficientes e bem estruturadas.

Além disso, a aplicação prática dos conteúdos estudados ao longo da graduação possibilitou a utilização de tecnologias e conceitos mais avançados, como a implementação de comunicação em tempo real por meio de WebSockets, essencial para o monitoramento dinâmico do atendimento e para a atualização imediata das informações exibidas aos usuários. Da mesma forma, o emprego de estruturas de fila mostrou-se uma solução adequada para o gerenciamento e o processamento ordenado de eventos e mensagens no sistema, contribuindo para maior desempenho, confiabilidade e escalabilidade da aplicação.

Por fim, a vivência proporcionada pelo projeto permitiu consolidar a integração entre teoria e prática, ampliando a compreensão sobre os desafios enfrentados em ambientes reais de desenvolvimento de software. Essa experiência contribuiu de forma significativa para a formação profissional, ao estimular o pensamento crítico, a autonomia na busca por soluções e a capacidade de adaptação a cenários complexos e dinâmicos, reforçando a importância da formação acadêmica como base para o desenvolvimento de soluções tecnológicas robustas, eficazes e alinhadas às necessidades do contexto organizacional.

## **7 Conclusão**

De forma geral, a participação no projeto contribuiu de maneira significativa para o aprimoramento acadêmico e profissional, ao possibilitar a efetiva integração entre os conhecimentos teóricos adquiridos ao longo da formação universitária e sua aplicação prática em um ambiente real de desenvolvimento de software. A experiência proporcionou uma visão mais abrangente e aprofundada dos processos que compõem a engenharia de software, abrangendo desde as etapas iniciais de levantamento, análise e especificação de requisitos até as fases de implementação, testes, validação e entrega contínua de funcionalidades, evidenciando a complexidade e a interdependência entre essas atividades.

Além do fortalecimento técnico, o envolvimento no projeto permitiu o desenvolvimento de competências relacionadas à organização e à gestão do trabalho em equipe, especialmente em um contexto orientado por práticas ágeis. Foi possível vivenciar, de forma concreta, aspectos como liderança técnica, tomada de decisão

baseada em critérios técnicos e organizacionais, priorização de demandas e adaptação a mudanças de escopo, habilidades fundamentais para o bom desempenho em projetos de software de médio e grande porte. Essas experiências contribuíram para a compreensão do papel do profissional de tecnologia não apenas como executor de tarefas, mas também como agente ativo no planejamento e na evolução das soluções desenvolvidas.

A atuação em um ambiente corporativo colaborativo também favoreceu o aprimoramento de competências interpessoais, tais como comunicação eficaz, cooperação entre os membros da equipe e capacidade de oferecer suporte técnico aos colegas diante de desafios e imprevistos. O contato constante com diferentes perfis profissionais e níveis de experiência reforçou a importância do trabalho coletivo, do compartilhamento de conhecimento e da postura proativa na resolução de problemas, aspectos essenciais para a construção de soluções consistentes e alinhadas às necessidades do negócio.

Dessa forma, o projeto possibilitou não apenas a consolidação do domínio de ferramentas, tecnologias e técnicas próprias da área de desenvolvimento de software, mas também o fortalecimento de habilidades profissionais e comportamentais indispensáveis à atuação no mercado de trabalho. Conclui-se, portanto, que a experiência contribuiu de maneira relevante para a formação integral, preparando o discente para enfrentar desafios futuros com maior maturidade técnica, visão sistêmica e capacidade de adaptação às constantes transformações do setor de tecnologia da informação.

## Referências

- [1] Almeida Machado Serviços em Gestão de Negócios LTDA, “Blue technology — consulta empresa,” Econodata, 2025, acesso em: 08 dez. 2025. [Online]. Available: <https://www.econodata.com.br/consulta-empresa/18191583000140-almeida-machado-servicos-em-gestao-de-negocios-ltda>
- [2] L. R. Júnior *et al.*, “Desenvolvimento de um sistema de controle remoto para automação de saídas digitais e analógicas por meio de uma aplicação web e uma aplicação local utilizando linguagem python,” *Brazilian Journal of Development*, vol. 7, no. 8, pp. 75 820–75 833, 2020. [Online]. Available: <https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/33740/pdf>
- [3] L. A. Martins *et al.*, “Gestão do conhecimento: análise do nível de maturidade de uma instituição pública,” *Brazilian Journal of Information Science: Research Trends*, vol. 17, p. e023010, Feb. 2023. [Online]. Available: <https://doi.org/10.36311/1981-1640.2023.v17.e023010>
- [4] R. d. S. Nonato *et al.*, “Gestão da informação: uma revisão sistemática da literatura sobre teorias, modelos e metodologias,” *Brazilian Journal of Information Science: Research Trends*, vol. 17, p. e023015, Apr. 2023. [Online]. Available: <https://doi.org/10.36311/1981-1640.2023.v17.e023015>
- [5] J. M. N. d. Santos *et al.*, “Desenvolvimento de webapp para gestão de alfaiataria,” in *16th International Conference on Information Systems and Technology Management*, 2023.