CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
Faculdade de Tecnologia de Mauá

Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma

Adriano das Chagas Barros
Brendon Gomes da Silva
Elias Sousa Barbosa
Rafael Ricardo Gonçalves

Volts – Sistema de Gestão de Trabalho Voluntário

Mauá
2025

Adriano das Chagas Barros
Brendon Gomes da Silva Elias
Sousa Barbosa
Rafael Ricardo Gonçalves

Volts – Sistema de Gestão de Trabalho Voluntário

Relatório técnico-científico apresentado à Fatec de Mauá, como exigência parcial para obtenção do título de Tecnólogo em Desenvolvimento de Software Multiplataforma, sob a orientação da profa. Me. Luana Lourenço.

Mauá
2025

RESUMO

O presente trabalho apresenta o desenvolvimento da plataforma Volts, um sistema web voltado à gestão e organização de atividades de trabalho voluntário. O projeto foi idealizado com o propósito de oferecer uma solução tecnológica capaz de centralizar informações, otimizar a comunicação entre voluntários, líderes e administradores e garantir maior eficiência na distribuição e acompanhamento das escalas de atuação. A concepção da plataforma partiu da necessidade de modernizar a gestão do voluntariado, que muitas vezes ocorre de forma descentralizada, por meio de planilhas e aplicativos de mensagens, dificultando o controle e a visibilidade das ações. Durante o processo de desenvolvimento, foram aplicadas metodologias ágeis, com destaque para o framework Scrum, que possibilitou a divisão do projeto em ciclos iterativos e entregas contínuas. A implementação seguiu princípios de engenharia de software, priorizando a qualidade do código, a modularidade e a escalabilidade do sistema. Foram utilizadas tecnologias modernas como C# com ASP.NET Core no back-end, React e TypeScript no front-end e os bancos de dados SQLite e PostgreSQL para os ambientes de desenvolvimento e produção, respectivamente. A plataforma também foi desenvolvida com foco em usabilidade e acessibilidade digital, utilizando o framework shadcn/UI e seguindo as diretrizes do W3C (World Wide Web Consortium). Testes funcionais e de integração foram realizados para garantir a estabilidade e o correto funcionamento dos módulos, enquanto avaliações de acessibilidade buscaram assegurar que o sistema fosse inclusivo a diferentes perfis de usuários. Os resultados obtidos demonstram que o Volts é uma ferramenta eficaz e de fácil utilização, capaz de simplificar o gerenciamento de grupos e voluntários, automatizar tarefas e promover maior engajamento social por meio da tecnologia. A aplicação se mostra uma alternativa viável e escalável para organizações que buscam aprimorar a administração de suas ações voluntárias de forma moderna, segura e colaborativa.

Palavras-chave: Gestão de voluntariado. Desenvolvimento web. Plataforma digital. Metodologias ágeis. Acessibilidade. Sistemas de informação. Engenharia de software. Inclusão digital. Transformação digital. API. React. ASP.NET Core.

SUMÁRIO

1 INTRODUÇÃO

O trabalho voluntário é uma das verdadeiras expressões de solidariedade. Possui a capacidade de conectar pessoas e possibilitar grandes transformações na sociedade.

Mas, mesmo sendo uma ferramenta poderosa de ação e transformação social, enfrenta desafios que dificultam o seu funcionamento, tais como: a má comunicação, gestão de pessoas e tarefas de forma ineficaz, entre outros.

Esses desafios prejudicam o voluntariado, desmotivando os voluntários, afetando a continuidade das ações e comprometendo o alcance dos objetivos.

Este relatório técnico apresenta a concepção, objetivos, fundamentos e a metodologia adotada para o seu desenvolvimento e aplicação.

Espera-se mostrar como a tecnologia pode facilitar a gestão do trabalho voluntário, aumentar a motivação dos participantes e ampliar o impacto positivo das ações sociais nas comunidades atendidas.

1.1 Contextualização

Ao longo da história, o trabalho voluntário tem se apresentado como uma reposta às diversas necessidades que o poder público não consegue suprir de maneira plena, especialmente em áreas como saúde, educação e segurança.

Dessa forma, a própria sociedade se organiza para preencher as lacunas deixadas pelo poder público e busca alternativas para apoiar uns aos outros, disponibilizando tempo, recursos e habilidades em prol do bem coletivo.

Essa mobilização social não é um fenômeno novo e sim uma prática que acompanha a humanidade em diferentes épocas.

O voluntariado é marcado pela solidariedade e pelo desejo de contribuir com a construção de uma realidade melhor.

A partir desse entendimento, o trabalho voluntário pode assumir várias formas, e todas carregam a sua nobreza, como, auxiliar em abrigos para pessoas em situação de

rua ou vulnerabilidade social, ajudar uma ong de cuidados a animais abandonados, dar aulas em um cursinho pré-vestibular comunitário entre tantas outras atividades.

Porém, muitos programas de voluntariado enfrentam obstáculos ao tentar suprir essa carência com agilidade, principalmente quanto à gestão de pessoas, organização do trabalho e acompanhamento de resultados e dessa forma, acabam por comprometer tanto o trabalho realizado quanto a motivação dos voluntários envolvidos.

Esse cenário se desenha frutífero ao desenvolvimento de tecnologias que auxiliem à gestão de um trabalho de tamanha nobreza como o voluntariado, influenciado diretamente no impacto social gerado.

1.2 Objetivo

Este relatório técnico tem como objetivo apresentar o Projeto Volts, com todas as suas características: nascimento, objetivos funcionalidades e metodologias.
Especificamente busca-se:

- Identificar os principais desafios enfrentados na gestão de voluntários;
- Propor soluções tecnológicas para otimizar a comunicação e coordenação de atividades;
- Demonstrar como o sistema Volts contribui para maior transparência e eficácia do gerenciamento de ações voluntárias.

1.3 Justificativa

A gestão eficiente do trabalho voluntário é um fator determinante para o sucesso de ações sociais, pois garante maior engajamento, motivação e resultados mensuráveis.

No entanto, a ausência de ferramentas adequadas e sistemas de acompanhamento prejudica a eficiência das atividades e acaba por desestimular os participantes.

A necessidade de simplificar a gestão do trabalho voluntário e a comunicação entre os colaboradores justifica o nascimento do Projeto Volts, uma solução robusta, na gestão do trabalho voluntário.

O projeto Volts oferece uma solução que centraliza informações, organiza grupos, agenda escalas de trabalho e tarefas e promove a transparência nas ações voluntárias.

Dessa forma, o projeto Volts contribui para a otimização da gestão interna, e para o fortalecimento do vínculo entre voluntários e instituições ampliando o alcance das iniciativas e colabora para o impacto positivo nas comunidades atendidas.

1.4 Problema de pesquisa

Muitas organizações que prestam trabalho voluntário enfrentam dificuldades para gerenciar o seu trabalho.

Há instituições que ainda trabalham de forma manual ou com sistemas pouco integrados. Isso gera retrabalho, perda de informações importantes e, muitas vezes, desmotivação entre os voluntários.

Essa falta de tecnologia e de organização torna difícil o planejamento estratégico, a coordenação de equipes e a análise de resultados de maneira eficiente.

Diante disso, surge a pergunta central desta pesquisa: como criar uma solução tecnológica que simplifique a gestão de voluntários, melhore a comunicação e garanta mais transparência, eficiência e facilidade para acompanhar os resultados das ações sociais?

O projeto Volts surge como uma resposta a essa questão, tornando-se um aliado estratégico das organizações, oferecendo um sistema poderoso que fortalece a gestão do trabalho voluntário.

1.5 Estrutura do relatório

No Capítulo 1, apresenta-se a introdução, que contextualiza o tema, expõe a motivação do estudo e descreve seus objetivos gerais e específicos.

O Capítulo 2 aborda o desenvolvimento do projeto, incluindo a fundamentação teórica, as tecnologias empregadas, a metodologia adotada e os testes realizados ao longo do processo de construção.

O Capítulo 3 descreve as especificações técnicas do sistema Volts, detalhando seus requisitos funcionais e não funcionais, os fluxos de processos e as interfaces do front-end.

O Capítulo 4 trata da implementação do sistema, contemplando a configuração do banco de dados, a estruturação da API como elo entre front-end e back-end, ainda é abordado nesse capítulo as diretrizes de qualidade e acessibilidade adotadas no sistema Volts, destacando as boas práticas aplicadas para garantir confiabilidade, usabilidade e inclusão digital.

O Capítulo 5 aborda e detalha os testes executados em cada etapa do desenvolvimento.

Por fim, o Capítulo 6 traz as considerações finais, principais conclusões, limitações do estudo, bem como indica possibilidades de aprimoramento e expansão futura.

2 DESENVOLVIMENTO

2.1 Fundamentação teórica

O voluntariado tem se destacado como uma potente forma de participação social, promovendo impacto positivo na sociedade. Sobre isso, Domeneguetti (2001), afirma que hoje em dia, voluntariado é um tema de suma importância para qualquer nação. Os governos definitivamente não conseguiram cumprir com suas obsessões de controle e abrangência, deixando "a peteca cair" em muitos setores, tais como saúde, educação, transportes, telecomunicações, energia e segurança.

Affonso (2018) acrescenta que o trabalho voluntário não é uma realidade contemporânea ou algo inventado pelas sociedades em sua nova configuração,

tratase de um processo que sempre esteve presente em toda a história da humanidade.

Nota-se que a boa gestão dos voluntários potencializa sua contribuição nas atividades da instituição, refletindo positivamente no desempenho organizacional.

Neste sentido, Oliveira (2018) salienta que o fator humano precisa ser gerido de forma eficaz e eficiente para conferir flexibilidade, agilidade e comprometimento organizacional, pois ainda segundo Hibels (2013) a maneira mais fácil de destruir um voluntário é desperdiçando o seu tempo. As pessoas responsáveis por coordenar voluntários, em particular, precisam ter esta lição em mente.

Desta forma se faz necessário que a gestão do trabalho voluntário seja feita de maneira eficaz, utilizando-se de recursos tecnológicos que possibilitem maior organização, comunicação e monitoramento das atividades.

Pressman (2011) afirma que todo projeto de software é motivado por alguma necessidade de negócios — a necessidade de corrigir um defeito em uma aplicação existente; a necessidade de adaptar um “sistema legado” a um ambiente de negócios em constante transformação; a necessidade de estender as funções e os recursos de uma aplicação existente, ou a necessidade de criar um novo produto, serviço ou sistema.

Dessa forma, observa-se que o desenvolvimento de software não apenas atende demandas específicas das organizações, mas também reflete a crescente dependência tecnológica da sociedade contemporânea.

Valente (2020) destaca que no mundo moderno, tudo é software. Hoje em dia, por exemplo, empresas de qualquer tamanho dependem dos mais diversos sistemas de informação para automatizar seus processos. Governos também interagem com os cidadãos por meio de sistemas computacionais... Empresas vendem, por meio de sistemas de comércio eletrônico. Software está também embarcado em diferentes dispositivos e produtos de engenharia, incluindo automóveis, aviões, satélites.

Essa presença do software na sociedade evidencia seu papel essencial e a maneira como pessoas e organizações interagem com o mundo moderno, pois ainda segundo Valente (2020) o software está contribuindo para renovar indústrias e

serviços tradicionais, como telecomunicações, transporte em grandes centros urbanos, hospedagem, lazer e publicidade.

Dessa forma, a adoção de ferramentas digitais representa não apenas a modernização dos processos, mas também um reflexo da transformação digital que vem redefinindo a forma como as organizações são estruturadas e interagem com seus colaboradores, segundo Baltzan (2016) as organizações que querer sobreviver no século XXI devem reconhecer o imenso poder da tecnologia, realizar mudanças organizacionais necessárias em relação a isso e aprender a operar de maneira totalmente diferente.

2.1.1 Revisão de literatura

O avanço da tecnologia e a crescente digitalização dos processos organizacionais têm impulsionado o uso de plataformas voltadas à gestão e automação de atividades, como é o caso da Volts.

O desenvolvimento de sistemas desse tipo exige o uso de metodologias e práticas consolidadas em Engenharia de Software, capazes de garantir eficiência, qualidade e confiabilidade no produto final.

Segundo Valente (2020) a Engenharia de Software trata da aplicação de abordagens sistemáticas, disciplinadas e quantificáveis para desenvolver, operar, manter e evoluir software. Ou seja, Engenharia de Software é a área da Computação que se preocupa em propor e aplicar princípios de engenharia na construção de software.

Assim, a seguir são apresentadas as principais técnicas e metodologias empregadas no desenvolvimento da plataforma Volts, destacando suas características, benefícios e importância para a construção de soluções tecnológicas robustas e escaláveis.

2.1.2 Técnicas de desenvolvimento de software

O processo de desenvolvimento da plataforma Volts envolve a aplicação de técnicas modernas que buscam otimizar o ciclo de vida do software, garantindo agilidade,

qualidade e capacidade de adaptação às necessidades dos usuários e das organizações, pois como afirma Santos et al. (2021) a tecnologia e a programação são os pilares dessas empresas e não apenas entregam entretenimento, mas também geram novas formas de comunicação, novas formas de trabalhar e novas formas de ver o mundo.

Entre as abordagens mais utilizadas no contexto atual destacam-se o desenvolvimento ágil, a integração contínua e a entrega contínua, que juntas proporcionam maior flexibilidade e controle sobre as etapas do projeto.

O desenvolvimento ágil tem como base a colaboração constante entre os membros da equipe e os usuários, priorizando entregas curtas e adaptabilidade às mudanças de requisitos. Nesse contexto, Martins (2007) defende que as metodologias ágeis surgiram com a finalidade de se desenvolver software de forma mais direta e menos burocrática. Em 2001 surgiu o Manifesto Ágil cujo objetivos são justamente atender as inovações contínuas, adaptabilidade de produtos, entregas com cronograma reduzido, adaptabilidade do processo e das pessoas e por fim resultados confiáveis

Mesmo sendo uma abordagem mais simples e flexível, a Metodologia Ágil requer disciplina e organização das equipes para que os resultados esperados sejam atingidos.

A metodologia ágil adotada no desenvolvimento da plataforma Volts foi o Scrum, amplamente utilizada em projetos de software por favorecer o trabalho colaborativo, a entrega contínua e a adaptação às mudanças. Brod (2013) traz a definição de que o Scrum é uma metodologia para construir software incrementalmente em ambientes complexos, em que os requisitos não são claros ou que mudam frequentemente.

De acordo com Badoco et al. (2018) a intenção deste método não é deixar de lado os processos e ferramentas, a documentação, a negociação de contratos ou o planejamento, mas pretende-se colocá-los em segundo plano, priorizando as iterações e pessoas, com o acompanhamento do stakeholder das mudanças que venha acontecer no processo de desenvolvimento do software.

Dessa forma, a equipe foi organizada de forma a dividir responsabilidades específicas, de acordo com as competências de cada integrante. Enquanto um membro ficou responsável pela interface do usuário (front-end), outro concentrou-se no desenvolvimento do back-end. Além disso, houve a participação de um integrante dedicado à criação de protótipos e design no Figma, e outro responsável pela documentação técnica do projeto. Essa estrutura permitiu uma execução integrada, com revisões constantes e comunicação eficiente entre os participantes, seguindo os princípios centrais da metodologia ágil.

Além da metodologia Scrum, a equipe da plataforma Volts adotou práticas de Integração Contínua (CI) e Entrega Contínua (CD), componentes essenciais do ciclo de DevOps.

A Integração Contínua permite que alterações no código sejam automaticamente testadas e integradas ao repositório principal, reduzindo a ocorrência de erros e facilitando a manutenção do sistema.

Para isso, foi utilizado o GitHub, que fornece recursos robustos de versionamento e controle de código, indo em direção ao entendimento de Sato (2013) ao afirmar que o uso de ferramentas de automação e integração contínua é essencial para manter a qualidade do software e reduzir o tempo entre desenvolvimento e entrega,

Por fim, a Entrega Contínua possibilitou disponibilizar novas versões do sistema de forma rápida e segura, promovendo agilidade no ciclo de desenvolvimento e maior satisfação dos usuários.

De acordo com Sommerville (2011), a integração entre práticas ágeis e automação de entregas é fundamental para alcançar qualidade e produtividade em ambientes de desenvolvimento modernos.

Assim, o uso de metodologias ágeis combinado com ferramentas de integração e entrega contínua consolidou-se como um fator essencial para o sucesso e a sustentabilidade da plataforma Volts.

2.1.3 Linguagens de programação e frameworks

O desenvolvimento de sistemas multiplataforma exige a utilização de linguagens e frameworks que ofereçam desempenho, segurança e integração eficiente entre as camadas da aplicação.

No caso do projeto Volts, o back-end foi desenvolvido em C#, uma linguagem de programação moderna, orientada a objetos e desenvolvida pela Microsoft, projetada para ser simples segura e robusta.

Carlucci et al. (2021) afirma que desde a sua criação o C# adiciona com frequência recursos para dar suporte a novas cargas de trabalho e práticas de design de software emergente... por se tratar de uma linguagem de programação com forte adoção e em constante desenvolvimento, as opções são infinitas.

Essa tecnologia é amplamente adotada no mercado por sua arquitetura modular e por facilitar a manutenção e evolução contínua das aplicações (ALBUQUERQUE, 2020).

Foi utilizado o framework ASP.NEP Core, desenvolvido pela Microsoft, utilizado para construir aplicações web modernas, APIs e serviços de forma rápida, segura e escalável.

Com o uso do ASP.NET Core, possibilita a criação de Web APIs REST full modernas, de alto desempenho e independentes de plataforma, realizando a integração eficiente entre o front-end e o back-end, garantindo que os dados circulem de maneira segura, organizada e padronizada.

Esse tipo de arquitetura favorece a escalabilidade, a reutilização de componentes e a facilidade de manutenção, já que cada serviço pode ser consumido de forma independente por diferentes clientes.

Já o front-end foi implementado utilizando React, em conjunto com HTML, CSS e TypeScript. O React é uma biblioteca JavaScript voltada para a construção de interfaces dinâmicas e reativas, promovendo uma melhor experiência do usuário e facilitando a reutilização de componentes, dando celeridade ao desenvolvimento.

Stefanov (2019) salienta que a biblioteca React ajuda você a definir a UI (interface do usuário) de uma vez por todas. Então, quando o estado da aplicação

mudar, a UI será reconstruída de modo a reagir à alteração, e você não precisará fazer nada adicional.

A utilização do TypeScript, por sua vez, agrega tipagem estática ao código JavaScript, reduzindo erros em tempo de execução e melhorando a produtividade da equipe de desenvolvimento. A utilização de tipagem ajuda no momento de depuração (debug) do nosso código, prevenindo alguns possíveis bugs ainda em tempo de desenvolvimento (Adriano 2021).

Para a estilização, foi utilizada a biblioteca shadcn/ui, que integra componentes otimizados para usabilidade e alinhados às boas práticas de design moderno. Essa abordagem favorece a construção de interfaces intuitivas e consistentes, promovendo uma experiência mais agradável aos usuários. O foco em usabilidade é um aspecto essencial no desenvolvimento moderno de sistemas, garantindo que todos possam interagir de maneira eficiente com a aplicação.

No que se refere ao banco de dados, o projeto utiliza SQLite em ambiente de desenvolvimento, pela sua leveza e facilidade de configuração, o que agiliza os testes locais e o versionamento do banco junto ao código-fonte.

Em ambiente de produção, foi adotado o PostgreSQL, um sistema gerenciador de banco de dados relacional robusto, seguro e de código aberto, amplamente reconhecido por sua conformidade com os padrões SQL e pela capacidade de lidar com alto volume de dados de maneira eficiente.

Segundo Carvalho (2022) o PostgreSQL tem tido aumento de popularidade por ser um sistema com melhores garantias de confiabilidade, melhores recursos de consulta, mais operação previsível, melhores recursos de consulta, suporta muitos tipos de dados sofisticados, possui mais operação previsível, fácil de usar, seguro e rápido.

2.1.4 Acessibilidade

A acessibilidade digital é um elemento essencial no desenvolvimento de plataformas web, pois garante que pessoas com diferentes tipos de limitações —

visuais, auditivas, motoras ou cognitivas — possam utilizar os sistemas de forma autônoma e eficiente.

Ignorar a acessibilidade é um erro num mundo em que cada vez mais países criam leis e normativos que protegem os direitos das pessoas com deficiência, garantindo seu direito à equidade de acesso (W3C – Capítulo São Paulo, [s.d.]).

No contexto da plataforma Volts, que visa conectar e gerenciar o trabalho voluntário, a acessibilidade não é apenas uma questão técnica, mas também um valor social, alinhado à própria missão de inclusão e colaboração que o projeto representa. Desenvolver com acessibilidade significa criar uma experiência que atenda a todos os usuários, independentemente de suas condições ou dispositivos utilizados.

Durante o desenvolvimento da interface da Volts, foram adotadas práticas e ferramentas que promovem acessibilidade e usabilidade, garantindo uma navegação mais intuitiva e inclusiva. O uso da biblioteca Shadcn/ui, em conjunto com o React, foi um fator determinante nesse processo, pois ela fornece componentes prontos que seguem os padrões de acessibilidade da W3C (WCAG). Esses componentes são desenvolvidos com foco em leitura por leitores de tela, contrastes adequados de cor e interações compatíveis com teclado, o que torna a interface mais adaptável a diversos perfis de usuários.

Além do aspecto técnico, a acessibilidade também contribui para a qualidade e sustentabilidade do software. Aplicações acessíveis são mais bem avaliadas em mecanismos de busca, alcançam um público maior e reforçam a imagem institucional de responsabilidade social.

A acessibilidade não é uma funcionalidade, algo opcional ou supérfluo a ser pensado ao final do desenvolvimento: ela é uma característica fundamental para o sucesso e a qualidade de qualquer produto (W3C – Capítulo São Paulo, [s.d.]).

Dessa forma, a plataforma Volts consolida seu propósito não apenas de conectar pessoas ao voluntariado, mas também de promover a inclusão digital e social por meio da tecnologia.

2.1.5 Metodologias de teste de software

A construção de um software não é uma tarefa simples, e a depender das características e dimensões do sistema a ser criado, pode se tornar algo bem complexo de ser feito. Por isso, estará sujeita a diversos tipos de problemas que podem acabar resultando em um produto com comportamento diferente do que se esperava.

Delamaro et al. (2007), afirma corretamente que muitos fatores podem ser identificados como causas de tais problemas, mas a maioria deles têm uma única origem: erro humano. Como a maioria das atividades de engenharia, a construção de um software depende principalmente da habilidade, da interpretação e da execução das pessoas que o constroem; por isso, erros acabam surgindo, mesmo com a utilização de métodos e ferramentas de engenharia de software.

Sendo assim, os testes de software são etapas essenciais no processo de desenvolvimento, pois garantem que o sistema funcione conforme os requisitos e mantenha sua qualidade ao longo das entregas.

O objetivo da atividade de teste não é, como podem pensar muitos, mostrar que um programa está correto. Ao invés disso, o objetivo é mostrar a presença de defeitos e corrigi-los caso existam. Quando a atividade de teste é realizada de maneira criteriosa e embasada tecnicamente, o que se tem é uma certa “confiança” de que se comporta corretamente para grande parte do seu domínio de entrada. (Delamaro et al. 2007).

A aplicação de metodologias de teste permite identificar falhas, inconsistências e pontos de melhoria, contribuindo para a confiabilidade do produto final. No contexto do projeto, foram utilizados diferentes tipos de testes, cada um com objetivos e abordagens complementares, de modo a assegurar tanto a funcionalidade quanto a robustez do sistema.

Entre as técnicas aplicadas, os testes de caixa preta foram utilizados para avaliar o comportamento externo da aplicação, sem levar em conta sua estrutura interna de código. Esse método foca nas entradas e saídas do sistema, verificando se as respostas produzidas estão de acordo com o esperado.

Essa abordagem foi especialmente útil para validar os fluxos principais do sistema e a integração entre suas diferentes camadas, garantindo que as funcionalidades entregassem os resultados esperados sob diversas condições.

Todavia, conforme Delamaro et al. (2007), o domínio de entrada pode ser infinito ou muito grande, de modo a tornar o tempo da atividade de teste inviável, fazendo com que essa alternativa se torne impraticável. Essa limitação da atividade de teste, que não permite afirmar, em geral, que o programa esteja correto, fez com que fossem definidas as técnicas de teste e os diversos critérios pertencentes a cada uma delas.

Tendo isso em mente, foi seguido à etapa de testes de caixa branca, que por sua vez, tiveram como objetivo examinar a lógica interna da aplicação, avaliando o código-fonte, os caminhos de execução e as estruturas de controle. Essa metodologia possibilitou uma análise mais detalhada de como o sistema processa os dados e executa suas operações, permitindo identificar erros lógicos, condições não tratadas e trechos de código redundantes. Essa verificação minuciosa contribuiu para o aumento da eficiência do sistema e a melhoria do desempenho geral da aplicação.

Por fim, foram realizados testes baseados em defeitos, cujo propósito é identificar falhas recorrentes ou potenciais vulnerabilidades a partir de erros comuns observados em ciclos anteriores de desenvolvimento. Essa metodologia auxilia na prevenção de problemas já conhecidos, promovendo um processo de melhoria contínua.

No geral, o que distingue essencialmente as três técnicas de teste – Funcional, Estrutural e Baseada em Erros – é a fonte utilizada para definir os requisitos de teste. Além disso, cada critério de teste procura explorar determinados tipos de defeitos, estabelecendo requisitos de teste para os quais valores específicos do domínio de entrada do programa devem ser definidos com o intuito de exercitá-los. (Delamaro et al. 2007).

A combinação dos três tipos de testes — caixa preta, caixa branca e baseados em defeitos — garantiu uma cobertura ampla de verificação, assegurando que o sistema fosse entregue com alta qualidade e confiabilidade, alinhado às boas práticas de engenharia de software.

2.2 Metodologia

O desenvolvimento da plataforma Volts foi conduzido com uma abordagem aplicada, voltada à solução de um problema comum em diversas instituições que contam com grupos de trabalho voluntário. Em muitos contextos, a gestão dessas equipes é realizada de forma descentralizada, por meio de planilhas, aplicativos de mensagens e outras ferramentas informais, o que dificulta o controle das atividades, a comunicação entre os participantes e o acompanhamento dos resultados.

A plataforma Volts foi idealizada para centralizar e otimizar a gestão do voluntariado, promovendo maior organização, integração e transparência nos processos. Por meio de uma interface intuitiva e recursos tecnológicos modernos, busca-se proporcionar uma experiência mais eficiente tanto para os coordenadores quanto para os voluntários, facilitando o planejamento, o monitoramento e a execução das ações.

A metodologia de desenvolvimento adotada garantiu qualidade, confiabilidade e reprodutibilidade em todas as etapas, desde a análise do problema até a validação do produto final, consolidando o Volts como uma ferramenta digital flexível e aplicável a diferentes tipos de organizações e projetos sociais.

2.2.1 Tipo de pesquisa

O presente trabalho é caraterizado como uma pesquisa ação, conforme definida por Thiollent (2011) a pesquisa-ação é um tipo de pesquisa social com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo e no qual os pesquisadores e os participantes representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo.

Este tipo de pesquisa é apropriado quando o pesquisador atua de forma participativa no processo de transformação da realidade estudada, propondo soluções e avaliando seus resultados de maneira contínua.

No contexto deste estudo, a pesquisa-ação foi aplicada ao desenvolvimento da plataforma Volts, que visa aprimorar a gestão de trabalhos voluntários por meio do uso

de tecnologias digitais. A abordagem permitiu que o processo de desenvolvimento fosse avaliado e ajustado em tempo real, de acordo com as necessidades identificadas, promovendo aprendizado constante e melhoria contínua.

Assim, além de gerar um produto funcional, a pesquisa possibilitou refletir sobre as metodologias e técnicas utilizadas, contribuindo tanto para a prática do desenvolvimento de software quanto para o avanço do conhecimento sobre gestão tecnológica de atividades voluntárias.

2.2.2 Método de coleta de dados

A coleta de dados neste trabalho foi conduzida de acordo com os princípios da pesquisa-ação, caracterizada pela participação ativa do pesquisador no processo de desenvolvimento e análise do objeto de estudo. Os dados foram obtidos de forma contínua e integrada às etapas de concepção, implementação e testes da plataforma Volts.

Durante o desenvolvimento, foram registrados observações, decisões técnicas, resultados de testes funcionais e de performance, bem como interações entre os membros da equipe. Esses registros permitiram analisar os impactos das metodologias aplicadas, como o uso do framework SCRUM, a integração contínua via GitHub, e as práticas de testes automatizados e manuais.

Além disso, a análise qualitativa das atividades desenvolvidas possibilitou identificar padrões, desafios e oportunidades de melhoria, tanto no processo de codificação quanto na gestão do projeto. Essa forma de coleta de dados favorece o aperfeiçoamento progressivo do software, em consonância com os ciclos iterativos e colaborativos que caracterizam a pesquisa-ação.

Assim, o método adotado garantiu uma visão prática e reflexiva sobre o desenvolvimento da plataforma, permitindo que as informações coletadas fossem utilizadas para aprimorar continuamente o produto e fundamentar as conclusões deste estudo.

2.2.3 Procedimentos de análise

Os procedimentos de análise adotados neste trabalho tiveram como base a observação sistemática e a avaliação prática das etapas de desenvolvimento da plataforma Volts. A análise foi conduzida de forma contínua, acompanhando cada ciclo iterativo da metodologia ágil SCRUM, de modo a assegurar que os resultados parciais fossem avaliados, validados e aprimorados ao longo do processo.

Foram executados testes unitários e testes de integração para validar o correto funcionamento dos módulos do sistema e a comunicação entre eles. Também foram aplicados testes de caixa preta e caixa branca, com o objetivo de verificar a conformidade com os requisitos funcionais e a estrutura interna do código.

Além disso, utilizaram-se ferramentas automatizadas de versionamento e integração contínua, como o GitHub, que auxiliaram na análise de desempenho, detecção de falhas e controle de versões.

Complementarmente, foram realizados testes práticos e observacionais com usuários, voltados à avaliação da facilidade de navegação, da experiência do usuário (UX) e de aspectos de acessibilidade da aplicação.

Essa etapa possibilitou identificar oportunidades de melhoria na interface e na interação, contribuindo para o aperfeiçoamento da usabilidade, da eficiência geral e da satisfação dos usuários. A análise dos resultados obtidos permitiu direcionar ajustes técnicos e funcionais, assegurando maior estabilidade e aderência aos objetivos definidos para o sistema.

2.2.4 Critérios de avaliação

A avaliação da plataforma Volts foi conduzida com base em critérios técnicos e qualitativos que permitiram mensurar sua eficácia, confiabilidade e aderência aos objetivos propostos.

Os principais aspectos considerados foram funcionalidade, performance, segurança, usabilidade, acessibilidade e manutenibilidade.

O critério de funcionalidade avaliou se as principais operações do sistema atendiam corretamente aos requisitos definidos e se os módulos interagiam de maneira integrada e estável.

A performance foi analisada a partir da capacidade do sistema em processar solicitações de forma ágil, sem lentidão ou falhas, garantindo fluidez no uso cotidiano.

Em relação à segurança, verificaram-se mecanismos de controle de acesso, armazenamento de dados e proteção de informações sensíveis, buscando assegurar a integridade e a confidencialidade dos dados dos usuários.

Já os aspectos de usabilidade e acessibilidade foram avaliados por meio de testes com usuários, levando em consideração a clareza da interface, a facilidade de

navegação e a conformidade com boas práticas de design inclusivo, conforme as diretrizes do W3C – Web Accessibility Initiative (WAI).

Por fim, a manutenibilidade foi analisada com base na organização do códigofonte, documentação e facilidade de atualização da aplicação. Esses critérios, em conjunto, garantiram uma visão ampla da qualidade do sistema, permitindo validar a eficiência técnica e a adequação do Volts às necessidades práticas de gestão de serviços e equipes.

3 DETALHAMENTO TÉCNICO

3.1 Especificação técnica do sistema Volts

A especificação do sistema Volts apresenta a descrição técnica e funcional do software, abrangendo os principais aspectos relacionados à sua arquitetura, tecnologias utilizadas, requisitos e interface. Essa etapa é essencial para documentar de forma estruturada as características do sistema, garantindo a clareza sobre seu funcionamento e seus objetivos.

A partir dessa especificação, é possível compreender as decisões de design e implementação adotadas, bem como os componentes que compõem a solução. Além disso, ela fornece subsídios para a análise de desempenho, manutenção e futuras evoluções da plataforma.

3.2 Ficha técnica

Nome do sistema/software:	Volts – Sistema de Gestão de Trabalho Voluntário
Versão:	1.0
Desenvolvedores:	Adriano Barros Brendon Gomes Elias Barbosa Rafael Goncalves
Data de criação:	19 de agosto de 2025
Plataforma(s):	Web
Linguagens utilizadas:	C#, JavaScript, TypeScript, HTML, CSS
Frameworks / Bibliotecas:	React, ASP.NET Core – Web API REST full, Shadcn ui
Banco de dados:	PostgreSQL, SQLite
Requisitos de hardware:	Navegador com suporte a HTML5, dispositivo com Android 8.0 ou superior, iOS 12 ou superior
Requisitos de software:	.NET SDK 8.0, Node.js 20+, PostgreSQL 16
Ferramentas de desenvolvimento:	Visual Studio Code
Objetivo:	Plataforma de gestão de atividades e serviços voluntários

3.3 Levantamento de requisitos

O levantamento de requisitos é uma das etapas mais importantes do processo de desenvolvimento de software, pois define as bases sobre as quais todo o sistema será construído. Essa fase tem como objetivo compreender de forma clara e detalhada as necessidades dos usuários, os objetivos do projeto e as restrições técnicas envolvidas, transformando essas informações em especificações que orientarão o desenvolvimento.

De acordo com Pressman (2011), a meta é identificar o problema, propor elementos da solução, negociar diferentes abordagens e especificar um conjunto preliminar de requisitos da solução em uma atmosfera que seja propícia para o cumprimento da meta.

Essa abordagem reforça a importância da comunicação e colaboração entre desenvolvedores, clientes e demais partes interessadas, a fim de garantir que o produto final atenda de maneira eficiente às expectativas e demandas do usuário.

No contexto do sistema VOLTS, o levantamento de requisitos foi conduzido com foco na usabilidade, segurança e desempenho, visando proporcionar uma experiência fluida e confiável aos usuários, além de garantir integridade e escalabilidade nas operações realizadas pela plataforma.

3.3.1 Requisitos funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, ou seja, suas funcionalidades e comportamentos esperados. Eles definem as ações que o usuário poderá realizar, as respostas do sistema a essas ações e os processos necessários para atender aos objetivos do projeto.

No caso da plataforma Volts, esses requisitos envolvem funcionalidades como cadastro e autenticação de usuários, gerenciamento de grupos e escalas de trabalho voluntário, controle de permissões de acesso e envio de notificações automáticas.

Tabela 1 – Requisitos funcionais

ID	Requisito Funcional	Descrição
RF01	Cadastro de voluntário	O sistema deve permitir que usuários não cadastrados realizem seu cadastro informando nome, e-mail, grupo de interesse e outros dados necessários.
RF02	Login de usuário	O sistema deve permitir que usuários cadastrados façam login utilizando e-mail e senha válidos.
RF03	Edição de perfil	O usuário autenticado deve poder atualizar suas informações pessoais, como nome, telefone e foto de perfil.

RF04	Visualização de escalas	O voluntário deve poder visualizar as escalas de trabalho do grupo ao qual pertence.
RF05	Candidatar-se a escalas	O voluntário deve poder se inscrever em escalas de dias e horários disponíveis.
RF06	Cancelar participação futura	O voluntário deve poder cancelar sua participação em escalas futuras, notificando o líder.
RF07	Gerenciar escalas do grupo	O líder deve poder criar, editar e excluir escalas do seu grupo.
RF08	Gerenciar voluntários	O líder deve poder adicionar ou excluir voluntários do grupo, conforme necessidade.
RF09	Criar e gerenciar grupos	O administrador deve poder criar, editar e excluir grupos de trabalho.
RF10	Gerenciar líderes	O administrador deve poder atribuir e remover líderes de grupos existentes.
RF11	Visualizar escalas globais	O administrador deve poder visualizar as escalas de todos os grupos.
RF12	Alterar escalas globais	O administrador deve poder alterar qualquer escala, independentemente do grupo.
RF13	Excluir usuários	O administrador deve poder excluir voluntários e líderes do sistema.
RF14	Controle de permissões	O sistema deve restringir funcionalidades de acordo com o perfil do usuário (voluntário, líder, administrador).
RF15	Painel de controle	O sistema deve oferecer um painel de controle (dashboard) com informações resumidas sobre escalas, grupos e voluntários.
RF16	Filtro de escalas	O sistema deve permitir filtrar escalas por data, grupo e status (aberta, preenchida, concluída).
RF17	Controle de acesso seguro	O sistema deve validar autenticação e sessões de usuários de forma segura, impedindo acessos indevidos.
RF18	Interface responsiva	O sistema deve se adaptar automaticamente a diferentes tamanhos de tela (desktop, tablet e celular).

Fonte: elaborado pelo autor (2025)

3.3.2 Requisitos Não Funcionais

Os requisitos não funcionais tratam de como o sistema deve se comportar, abrangendo aspectos de desempenho, segurança, confiabilidade, acessibilidade e usabilidade. Esses requisitos garantem que a aplicação ofereça uma experiência eficiente, estável e inclusiva aos usuários, além de atender a boas práticas de desenvolvimento e operação em ambientes web modernos.

Tabela 2 – Requisitos não funcionais

ID	Requisito Funcional	Descrição
RNF01	Desempenho	O sistema deve carregar a página inicial e as principais telas em até 3 segundos, em condições normais de rede.
RNF02	Escalabilidade	A aplicação deve suportar o aumento do número de usuários e grupos sem perda significativa de desempenho.
RNF03	Segurança	O sistema deve garantir autenticação segura, com senhas criptografadas e proteção contra-ataques comuns (SQL Injection, XSS).
RNF04	Disponibilidade	O sistema deve estar disponível para acesso pelo menos 99% do tempo mensal.
RNF05	Usabilidade	A interface deve ser intuitiva e de fácil navegação, permitindo que usuários sem experiência técnica possam utilizá-la.
RNF06	Acessibilidade	O sistema deve seguir as diretrizes do W3C (WCAG 2.1), garantindo compatibilidade com leitores de tela e navegação por teclado.
RNF07	Portabilidade	A plataforma deve ser compatível com os principais navegadores (Chrome, Firefox, Edge, Safari) e dispositivos móveis.
RNF08	Manutenibilidade	O código-fonte deve ser modular e documentado, permitindo fácil atualização e correção de falhas.
RNF09	Confiabilidade	O sistema deve manter integridade dos dados durante operações simultâneas e falhas inesperadas.
RNF10	Backup e recuperação	O banco de dados deve possuir mecanismos automáticos de backup e recuperação para evitar perda de informações.

Fonte: elaborado pelo autor (2025)

3.4 Modelagem e fluxos de processos

A etapa de modelagem do sistema tem como objetivo representar, de forma visual e estruturada, o funcionamento interno do Volts, destacando as interações entre os diferentes tipos de usuários e o sistema. Essa representação é fundamental para compreender o fluxo das operações, identificar possíveis gargalos e assegurar que os processos estejam alinhados aos objetivos funcionais do projeto.

Os diagramas e fluxos de processos permitem uma visão clara da jornada do usuário, desde o primeiro contato com a plataforma até as ações realizadas durante o uso cotidiano. Essa abordagem facilita tanto a fase de desenvolvimento quanto futuras manutenções ou expansões do sistema, pois fornece uma base visual de referência para a lógica de funcionamento.

3.4.1 Fluxo do visitante e processo de cadastro

Este fluxograma demonstra o percurso de um visitante ao acessar o sistema pela primeira vez, contemplando as etapas de registro, envio de informações e confirmação de cadastro. O processo foi desenhado para ser simples, intuitivo e seguro, garantindo que novos usuários consigam se cadastrar sem dificuldades.

Figura 1 – Fluxo do visitante e processo de cadastro

Fonte: Elaborado pelo autor (2025)

3.4.2 Fluxo do voluntário

O voluntário, após ter seu cadastro aprovado, pode acessar o dashboard do sistema, visualizar e atualizar seu perfil pessoal, além de consultar as escalas de atividades de seu grupo. Este fluxograma descreve as principais interações realizadas nesse processo, representando o uso cotidiano do sistema.

Figura 2 – Fluxo do voluntário (acesso ao dashboard e escolha de escalas)

Fonte: Elaborado pelo autor (2025)

3.5 Interface do sistema Volts

A interface do sistema Volts foi desenvolvida com foco na usabilidade e na experiência do usuário, oferecendo uma navegação simples, agradável e adaptada aos diferentes perfis — visitantes, voluntários, líderes e gestores. O design prioriza a clareza visual, a organização das informações e a facilidade de acesso às principais funcionalidades do sistema.

Foram considerados princípios de acessibilidade digital, como contraste adequado, legibilidade e navegação intuitiva, assegurando que o sistema seja inclusivo e acessível a todos os usuários.

A seguir, são exibidas algumas telas do sistema Volts, demonstrando sua estrutura visual e principais funcionalidades.

3.5.1 Tela Inicial

A tela inicial do sistema Volts apresenta-se como o ponto de entrada para os usuários, reforçando o propósito central da plataforma: conectar, organizar e engajar voluntários. Seu slogan, “Sua Energia Transforma”, traduz de forma clara a missão do sistema de promover o voluntariado de maneira acessível e colaborativa.

Figura 7 – Tela inicial da plataforma Volts

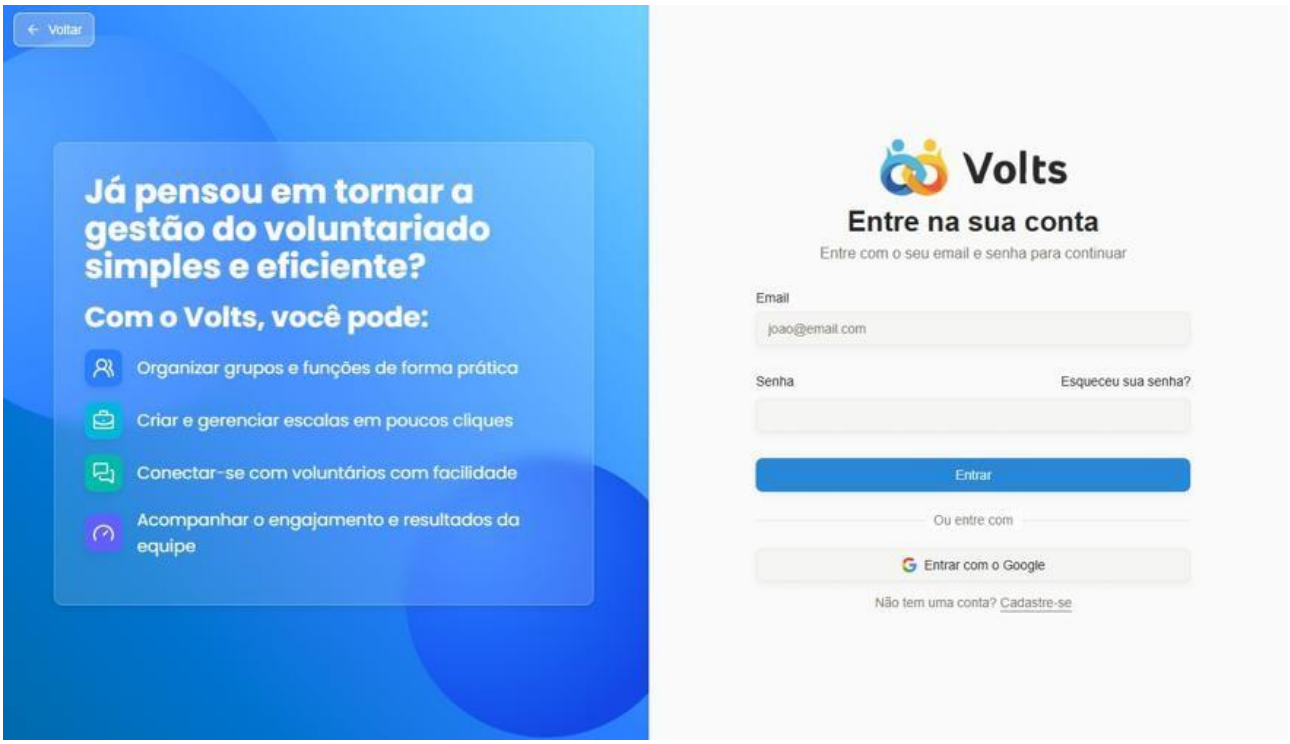
Fonte: Elaborado pelo autor (2025)

3.5.2 Tela de login

A tela de login é o ponto de acesso principal ao sistema, permitindo que usuários cadastrados autentiquem suas credenciais de forma segura. O processo utiliza autenticação baseada em JWT (JSON Web Token), garantindo que apenas perfis autorizados tenham acesso às funcionalidades internas da plataforma.

Visualmente, a tela foi projetada para transmitir clareza e confiança, com campos de entrada diretos, feedback em caso de erro e botões de acesso rápido, como “Entrar” e “Esqueci minha senha”. Essa abordagem reduz a fricção no acesso e reforça a segurança dos dados dos usuários.

Figura 8 – Tela de login da plataforma Volts



Fonte: Elaborado pelo autor (2025)

3.5.3 Tela de cadastro

A tela de cadastro é o ponto de entrada para novos voluntários interessados em participar das atividades. O formulário coleta informações básicas, como nome, email, senha e grupo de interesse, que são posteriormente avaliadas pelo líder responsável. Após o envio, o candidato recebe uma notificação automática sobre o status da solicitação.

O design da tela foi pensado para ser acolhedor e intuitivo, utilizando elementos visuais que reforçam o propósito social da plataforma e incentivam o engajamento dos usuários desde o primeiro contato.

Figura 9 – Tela de cadastro da plataforma Volts

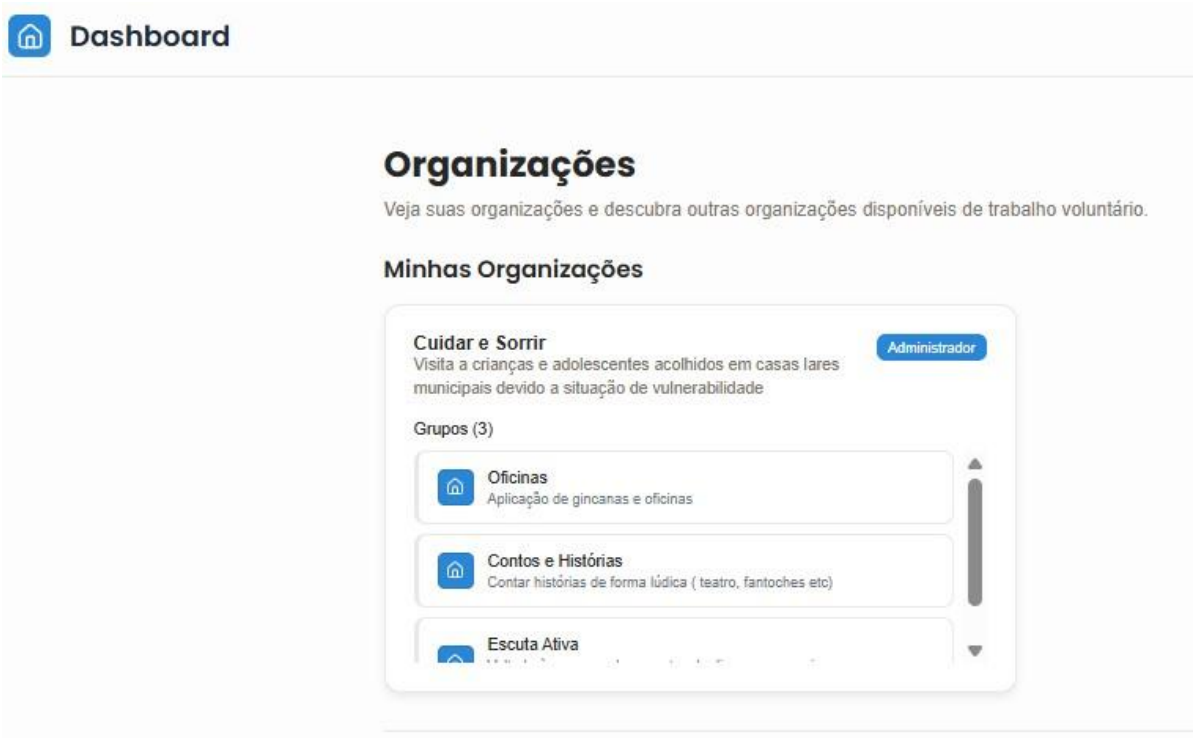
Fonte: Elaborado pelo autor (2025)

3.5.4 Tela dashboard do voluntário

Após o login, o voluntário é direcionado ao seu painel de controle (dashboard), onde pode visualizar suas escalas de trabalho, eventos futuros e histórico de participação. A interface apresenta informações organizadas em cartões e listas dinâmicas, facilitando a navegação e a compreensão das atividades em andamento.

Além disso, o dashboard permite ao usuário se candidatar a novas escalas, editar sua disponibilidade e atualizar seus dados de perfil. Essa área representa o núcleo funcional da experiência do voluntário, concentrando as principais interações e promovendo autonomia na gestão de suas atividades dentro do sistema.

Figura 9 – Tela de cadastro da plataforma Volts



Fonte: Elaborado pelo autor (2025)

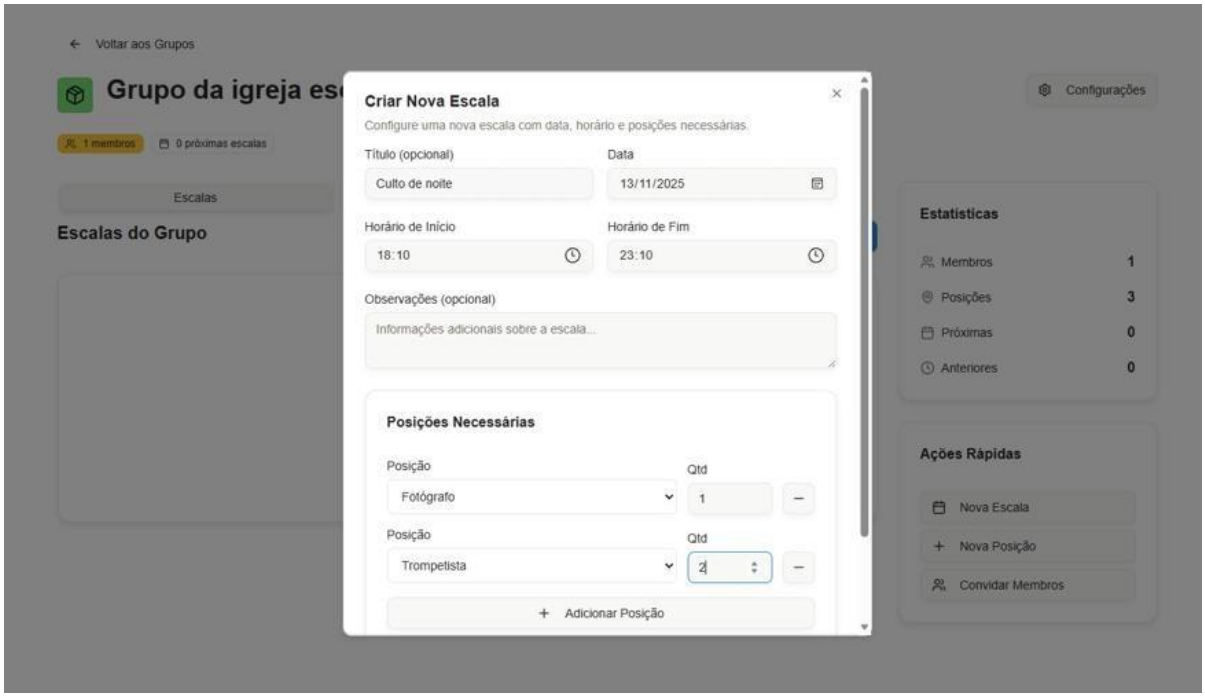
3.5.5 Tela de criação de escala

A tela de criação de escala foi desenvolvida para simplificar o processo de organização das atividades dos voluntários dentro dos grupos. Por meio dela, o líder

ou administrador pode definir novas escalas de trabalho, especificando data, horário, local, funções e quantidade de participantes necessários.

A interface prioriza a praticidade e a clareza na disposição dos campos, permitindo que o usuário configure todas as informações essenciais de maneira rápida e intuitiva.

Figura 9 – Tela de cadastro da plataforma Volts



Fonte: Elaborado pelo autor (2025)

3.5.6 Demais telas

Além das telas apresentadas, foram desenvolvidas outras interfaces que compõem o fluxo completo de uso da plataforma Volts, incluindo áreas de gestão de grupos, administração de usuários, aprovação de cadastros e visualização de relatórios. Essas telas seguem o mesmo padrão visual e estrutural adotado nas demais, garantindo consistência, usabilidade e coerência no design do sistema.

No entanto, para evitar que o trabalho fique excessivamente extenso e dado o objetivo de destacar apenas os principais elementos da interface, as demais telas não serão exibidas neste documento. Todas fazem parte do conjunto funcional da plataforma e podem ser acessadas no repositório oficial do projeto, disponível no GitHub.

4 IMPLEMENTAÇÃO

4.1 Implementação do sistema

A etapa de implementação representa a concretização das etapas de análise e modelagem, transformando o planejamento do projeto em um sistema funcional.

No desenvolvimento do Volts, foram implementadas as camadas de interface, lógica de negócio e persistência de dados, de forma integrada e segura.

A API (Application Programming Interface) atua como elo entre o front-end e o banco de dados, permitindo a troca estruturada de informações e garantindo a consistência das operações realizadas pelos diferentes perfis de usuários.

4.2 Configuração do Banco de Dados

O trecho a seguir realiza a configuração do banco de dados SQLite, utilizado para armazenar informações persistentes do sistema, como dados de usuários, grupos, escalas e registros de participação.

```
// Configuração do banco de dados SQLite
builder.Services.AddDbContext<VoltsDbContext>(options =>
    options.UseSqlite(builder.Configuration.GetConnectionString("DefaultConnection")));
```

A integração é feita por meio do Entity Framework Core (EF Core), que permite a manipulação de dados por meio de objetos (ORM — Object Relational Mapping), facilitando a manutenção e reduzindo erros de consulta direta ao banco.

4.3 Configuração da API e Segurança

A configuração adequada da API é essencial para garantir o funcionamento correto, eficiente e confiável de um sistema. Uma API bem estruturada define regras claras de comunicação entre serviços, reduz falhas e melhora a escalabilidade. Aliada a isso, a implementação de medidas de segurança — como autenticação, autorização, controle de acessos e proteção contra ameaças — é fundamental para preservar a integridade dos dados e evitar vulnerabilidades que possam comprometer o sistema.

Dessa forma, uma configuração sólida e segura não apenas assegura o bom desempenho da aplicação, mas também fortalece a confiança dos usuários e a proteção das informações envolvidas.

4.3.1 Configuração de Autenticação e Autorização

Este código implementa a autenticação baseada em JWT (JSON Web Token), garantindo que apenas usuários autenticados possam acessar recursos protegidos da API.

```
// Autenticação JWT
var jwtSettings = builder.Configuration.GetSection("Jwt"); var
secretKey = jwtSettings["SecretKey"];
builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
.AddJwtBearer(options =>
  {
     options.TokenValidationParameters = new TokenValidationParameters
     {
        ValidateIssuer = true,
        ValidateAudience = true,
        ValidateLifetime = true,
        ValidateIssuerSigningKey = true,
        ValidIssuer = jwtSettings["Issuer"],
        ValidAudience = jwtSettings["Audience"],
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(secretKey))
     };
  });
```

Esse mecanismo assegura a integridade e a confidencialidade dos dados trafegados, permitindo o controle de acesso individualizado para voluntários, líderes e administradores.

4.3.2 Configuração de CORS (Cross-Origin Resource Sharing)

A configuração de CORS (Cross-Origin Resource Sharing) define quais origens externas podem se comunicar com a API, prevenindo acessos indevidos.

```
// CORS para permitir acesso do front-end
builder.Services.AddCors(options =>
{
    options.AddPolicy("DevCors", policy =>
    {
        policy.WithOrigins("http://localhost:5173")
              .AllowAnyHeader()
              .AllowAnyMethod()
              .AllowCredentials();
    });
});
```

No ambiente de desenvolvimento, foi permitido o acesso do front-end hospedado localmente, garantindo a integração entre as interfaces e os serviços da aplicação durante os testes.

4.4 Prática de qualidade e acessibilidade digital

Esta seção apresenta as principais práticas adotadas para garantir que o sistema VOLTS seja desenvolvido com padrões elevados de qualidade e alinhado às diretrizes de acessibilidade digital. O objetivo é assegurar que o sistema seja confiável, eficiente e inclusivo, possibilitando que diferentes perfis de usuários — incluindo pessoas com limitações visuais, motoras ou cognitivas — possam utilizá-lo de forma plena.

As subseções a seguir descrevem as abordagens aplicadas durante o desenvolvimento, contemplando tanto processos de engenharia de software voltados à qualidade quanto as técnicas empregadas na avaliação e implementação de acessibilidade no ambiente digital.

4.4.1 Metodologia de desenvolvimento e qualidade

O desenvolvimento do sistema VOLTS foi conduzido com base em uma metodologia ágil, priorizando a entrega contínua de funcionalidades e a adaptação rápida às necessidades identificadas durante o processo de construção. Essa abordagem favoreceu uma comunicação constante entre os integrantes da equipe e permitiu a revisão e o aprimoramento frequente das etapas de desenvolvimento.

A aplicação de boas práticas de engenharia de software foi fundamental para garantir a qualidade do produto. Foram adotados critérios de padronização de código, versionamento contínuo e revisões periódicas das funcionalidades implementadas. Essas medidas contribuíram para manter o sistema estável, seguro e com boa performance, reduzindo a ocorrência de falhas e retrabalhos.

Além disso, a qualidade do projeto foi assegurada por meio da integração entre o planejamento detalhado, os testes sistemáticos e a verificação constante de conformidade com os requisitos funcionais e não funcionais definidos. Esse conjunto de práticas garantiu que o VOLTS atendesse aos objetivos propostos, oferecendo uma solução confiável e eficiente para o gerenciamento de atividades voluntárias.

Por fim, a equipe de desenvolvimento adotou uma postura de melhoria contínua, prevendo revisões periódicas do sistema e a implementação de novas funcionalidades conforme a evolução das necessidades dos usuários. Essa perspectiva de aprimoramento contínuo visa assegurar que o VOLTS se mantenha atualizado, escalável e alinhado às melhores práticas de tecnologia e gestão de software.

4.4.2 Avaliação de acessibilidade

A acessibilidade é um fator essencial no desenvolvimento do sistema Volts, considerando que a plataforma deve permitir o uso eficiente por diferentes perfis de usuários, incluindo pessoas com limitações visuais, motoras ou cognitivas. A avaliação

de acessibilidade foi conduzida com o objetivo de verificar a conformidade da aplicação com as diretrizes da WCAG 2.1 (Web Content Accessibility Guidelines), garantindo que todos os usuários possam interagir com o sistema de forma autônoma e intuitiva.

Para apoiar esse processo, o projeto fez uso do framework shadcn/UI, baseado no Radix UI, que oferece componentes React acessíveis e conformes aos padrões WAI-ARIA. Essa escolha contribuiu significativamente para a criação de uma interface consistente e inclusiva, reduzindo barreiras de navegação e assegurando que os elementos interativos fossem devidamente interpretados por leitores de tela e navegáveis por teclado. Além disso, o uso desse framework garantiu foco visível, estrutura semântica adequada e comportamento previsível dos componentes em diferentes dispositivos e resoluções.

A avaliação prática de acessibilidade envolveu o uso de ferramentas automatizadas, como o Lighthouse e o WAVE, para identificar possíveis problemas de contraste, hierarquia de elementos e ausência de descrições alternativas em imagens e botões. Também foram realizados testes manuais, nos quais se verificou a navegação completa por teclado e a leitura dos elementos por leitores de tela. Esses testes permitiram avaliar a experiência de uso sob diferentes condições, assegurando que o sistema fosse funcional mesmo sem o uso de mouse.

Os resultados obtidos demonstraram boa conformidade com os princípios de perceptibilidade, operabilidade, compreensibilidade e robustez, propostos pelas WCAG 2.1. Ainda assim, foram identificadas oportunidades de aprimoramento, como o ajuste de contraste em áreas secundárias e a padronização de descrições alternativas em ícones não textuais. Essas melhorias estão previstas para as próximas iterações do sistema.

A análise final indica que o Volts apresenta um bom nível de acessibilidade, compatível com os padrões modernos de desenvolvimento web. A aplicação das diretrizes e o uso de frameworks acessíveis contribuíram para tornar o sistema mais inclusivo, reforçando o compromisso do projeto com a experiência do usuário e com os princípios de design universal.

5 TESTES DO SISTEMA

5.1 Técnicas e testes de software

Esta seção apresenta as técnicas utilizadas para garantir a qualidade, confiabilidade e usabilidade do Volts, sistema desenvolvido para gerenciar atividades de trabalho voluntário. O processo de teste teve como objetivo assegurar o correto funcionamento das funcionalidades.

5.2 Testes unitários

Os testes unitários são uma técnica de verificação de software que tem como objetivo avaliar o funcionamento individual de pequenas partes do código, conhecidas como unidades — geralmente funções, métodos ou componentes isolados. Esse tipo de teste busca garantir que cada unidade do sistema opere conforme o esperado, independentemente das demais partes do programa

Delamaro et al.(2007) afirma que como cada unidade é testada separadamente, o teste de unidade pode ser aplicado à medida que ocorre a implementação das unidades e pelo próprio desenvolvedor, sem a necessidade de dispor-se do sistema totalmente finalizado.

No desenvolvimento da plataforma Volts, os testes unitários desempenharam um papel fundamental na garantia da qualidade e da confiabilidade do sistema. Eles foram aplicados principalmente nos módulos críticos da aplicação, assegurando que as regras de negócio e os comportamentos esperados fossem corretamente implementados e mantidos ao longo do processo de desenvolvimento.

A execução dos testes foi realizada em ambiente controlado, com o códigofonte versionado no GitHub, o que possibilitou rastrear alterações e manter um histórico preciso de cada modificação. Essa abordagem permitiu detectar erros logo nas etapas iniciais do ciclo de desenvolvimento, reduzindo o custo de correção e evitando impactos em outras funcionalidades.

Os módulos mais testados incluíram o cadastro e autenticação de voluntários, o gerenciamento de grupos e escalas de trabalho e o sistema de notificações. Para a automação e execução dos testes, foram utilizadas ferramentas amplamente adotadas no ecossistema web, como o Jest, que permite a criação de cenários simulados e a análise dos resultados de forma eficiente. Essa prática reforçou a integridade do código e contribuiu para a estabilidade e manutenção contínua da plataforma.

5.3 Testes de integração

Após a validação das partes individuais do sistema por meio dos testes unitários, realiza-se a etapa de testes de integração. Nessa fase, o objetivo é verificar se os diferentes módulos e componentes do VOLTS — como o cadastro de voluntários, a gestão de projetos, o controle de horas e o sistema de comunicação — funcionam corretamente quando interagem entre si.

Os testes de integração buscam identificar falhas que podem surgir na troca de informações entre os módulos. Por exemplo, é importante garantir que, ao cadastrar um novo voluntário, suas informações sejam corretamente transmitidas para o módulo de projetos e apareçam nas listas de disponibilidade. Da mesma forma, quando um

coordenador aprova uma participação, o sistema deve atualizar os registros no banco de dados e refletir essa alteração na área do voluntário.

Indo ao encontro do entendimento de Delamaro et al. (2007) à medida que as diversas partes do software são colocadas para trabalhar juntas, é preciso verificar se a interação entre elas funciona de maneira adequada e não leva a erros.

Portanto, esse tipo de teste é essencial para assegurar a consistência e a fluidez das operações dentro do VOLTS, prevenindo erros de comunicação entre partes do sistema que, isoladamente, funcionam bem. Assim, os testes de integração garantem que o conjunto do sistema opere de forma coesa, proporcionando uma experiência confiável e integrada aos usuários.

5.4 Testes de performance

Os testes de performance tiveram como objetivo avaliar a capacidade do sistema VOLTS em lidar com múltiplos acessos simultâneos e grandes volumes de dados, garantindo que a aplicação mantenha boa responsividade e estabilidade em diferentes condições de uso.

Durante o processo, foram simuladas situações de uso intensivo, como cadastros simultâneos de voluntários, geração de relatórios e atualizações de campanhas. Esses testes buscaram identificar gargalos no carregamento de páginas, lentidão nas consultas e possíveis falhas de resposta do servidor.

A partir dos resultados obtidos, foram realizados ajustes no banco de dados e otimizações em trechos de código que envolviam operações repetitivas ou consultas complexas, garantindo um desempenho mais fluido e confiável, mesmo em cenários de alta demanda.

5.5 Testes de segurança

Os testes de segurança foram aplicados com o intuito de assegurar a integridade, confidencialidade e proteção dos dados dos voluntários e das organizações cadastradas no sistema VOLTS.

Entre as principais validações realizadas, destacam-se: a verificação de autenticação segura dos usuários, a proteção contra acessos não autorizados, e o bloqueio de possíveis vulnerabilidades comuns, como injeção de código e manipulação de URLs.

Além disso, foram analisados os processos de armazenamento e transmissão de informações, garantindo que dados sensíveis, como credenciais e informações pessoais, fossem tratados de forma segura. Essas medidas visaram reforçar a

confiabilidade do sistema e a conformidade com boas práticas de segurança em aplicações web.

6. CONSIDERAÇÕES FINAIS

6.1 Principais conclusões

O desenvolvimento do Volts possibilitou compreender, na prática, as etapas envolvidas na criação de um sistema web voltado à gestão de trabalho voluntário. O projeto demonstrou que é possível unir tecnologia e organização social em uma solução eficiente, intuitiva e acessível.

Durante a implementação, foram aplicadas metodologias ágeis que permitiram ciclos iterativos de melhoria contínua, garantindo a entrega de um produto funcional e alinhado às necessidades dos diferentes perfis de usuários — voluntários, líderes e administradores.

Os testes realizados evidenciaram a eficiência do sistema em gerenciar escalas, aprovações e comunicações entre os participantes de forma centralizada e automatizada. Além disso, o uso de frameworks modernos, como o shadcn/UI, contribuiu para elevar o nível de acessibilidade e padronização da interface, tornando a navegação mais agradável e inclusiva.

6.2 Recomendações

Recomenda-se a continuidade do aprimoramento do Volts, com foco na ampliação das funcionalidades e na implementação de recursos que aumentem a integração entre usuários e grupos. Sugere-se, ainda, a aplicação de testes com um número maior de usuários em ambiente real, a fim de obter dados mais abrangentes sobre a usabilidade e o desempenho da aplicação.

Outra recomendação importante é o fortalecimento da camada de segurança, incluindo autenticação multifator e criptografia avançada de dados, de modo a garantir a proteção das informações pessoais dos voluntários. A criação de uma versão mobile responsiva ou um aplicativo nativo também se apresenta como um caminho promissor para tornar o sistema ainda mais acessível.

6.3 Limitações do estudo

Entre as limitações observadas, destaca-se o fato de que o sistema foi testado em um ambiente controlado e com um grupo reduzido de usuários, o que restringe a generalização dos resultados.

Além disso, nem todas as funcionalidades planejadas inicialmente foram implementadas, devido ao tempo disponível e à complexidade técnica envolvida. Um exemplo disso é a ausência do recurso de disparo automático de e-mails ou notificações, cuja implementação depende da contratação de serviços externos pagos, o que tornou inviável sua integração na versão atual do sistema.

Outra limitação diz respeito à ausência de integração com sistemas externos, como plataformas de comunicação e gestão de eventos, o que poderia ampliar as possibilidades de uso e tornar o sistema mais completo.

6.4 Perspectivas futuras

Para o futuro, espera-se evoluir o Volts incorporando novas tecnologias e recursos que fortaleçam sua proposta de valor. Entre as melhorias previstas, destacam-se:

- Implementação de um aplicativo mobile com notificações em tempo real;
- Implementação de novos requisitos funcionais
- Implementação de recursos de comunicação em tempo real entre voluntários
- Integração com serviços de agenda e calendário, como Google Calendar;
- Criação de relatórios analíticos para líderes e administradores;
- Aprimoramento da inteligência de dados, utilizando indicadores de participação e engajamento;
- Expansão do sistema para outros contextos, como gestão de eventos, voluntariado corporativo e projetos sociais comunitários.

Essas evoluções visam transformar o Volts em uma ferramenta cada vez mais completa, confiável e inclusiva, reafirmando seu papel como um facilitador na organização e valorização do trabalho voluntário.

REFERÊNCIAS

ADRIANO, Thiago da Silva. Guia prático de TypeScript: melhore suas aplicações JavaScript. São Paulo: Casa do Código, 2021.

AFFONSO, Lígia Maria Fonseca. Mobilização social. Florianópolis: SAGAH Educação SA, 2018.

BADOCO, Maurício; ALMEIDA, Renato Inácio de; LUCAS, Carlos Alberto de. Metodologia ágil na gestão de projetos de software. Revista EduFatec: Educação, Tecnologia e Gestão, v. 1, n. 1, p. 1–18, jan./jun. 2018.

BALTZAN, Paige. Tecnologia orientada para gestão [recurso eletrônico]. Tradução de Rodrigo Dubal. 6. ed. Porto Alegre: AMGH, 2016.

BROD, C. Scrum: Guia prático para projetos ágeis. São Paulo: Novatec, 2013.

CARLUCCI, André; SANTOS, Carlos dos; ANDRADE, Claudenir; ALMEIDA, Rafael; CARNEIRO, Ray; HADDAD, Renato. C# para iniciantes. São Paulo: Agência Hex, 2021.

CARVALHO, Vinícius. PostgreSQL: banco de dados para aplicações web modernas. São Paulo: Casa do Código, 2022. ISBN 978-85-5519-255-5.

DELAMARO, Márcio Eduardo; MALDONADO, José Carlos; JINO, Mario. Introdução ao teste de software. [S.l.]: [s.n.], [s.d.]. ISBN 978-85-352-2634-8.

DOMENEGHETTI, Ana Maria. Voluntariado: gestão do trabalho voluntário em organizações sem fins lucrativos. São Paulo: Esfera, 2001. ISBN 85-87293-25-7.

HYBELS, Bill. A Revolução do Voluntariado: liberando o poder de cada um de nós. 1. ed. São Paulo: Vida, 2013. ISBN 978-8538302650.

OLIVEIRA, Luana Yara Miolo de. Gestão de pessoas. Porto Alegre: SAGAH Educação SA, 2018.

PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. 7. ed. Porto Alegre: AMGH, 2011.

SATO, Danilo. DevOps na prática: entrega de software confiável e automatizada. São Paulo: Casa do Código, 2014.

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

STEFANOV, Stoyan. Primeiros passos com React: construindo aplicações web. Tradução de Lúcia A. Kinoshita. São Paulo: Novatec, 2019.

THIOLLENT, Michel. Metodologia da pesquisa-ação. 18. ed. São Paulo: Cortez, 2011.

VALENTE, Marco Túlio de Oliveira. Engenharia de Software Moderna: princípios e práticas para desenvolvimento de software com produtividade. Belo Horizonte: Universidade Federal de Minas Gerais, 2020.

W3C – Capítulo São Paulo. Cartilha de acessibilidade na web: fascículo VI – avaliando acessibilidade e resultados. Coordenação geral de Reinaldo Ferraz e Amanda Marques. Redação de Lêda Spelta e Fernanda Lobato. São Paulo: W3C; Comitê Gestor da Internet no Brasil (CGI.br); Núcleo de Informação e Coordenação do Ponto BR (NIC.br), [s.d.].

FORMULÁRIO DE IDENTIFICAÇÃO

<table>
<tr><th colspan="4">DADOS DO RELATÓRIO TÉCNICO E/OU CIENTÍFICO</th></tr>
<tr><td colspan="2">Título e subtítulo:
Volts: Sistema de Gestão de Trabalho Voluntário</td><td colspan="2">Tipo de relatório:
Relatório Técnico-Científico</td></tr>
<tr><td colspan="2">Instituição:
Faculdade de Tecnologia de Mauá - FATEC</td><td colspan="2">Data: 04/12/2025</td></tr>
<tr><td colspan="2">Curso:
Tecnologia em Desenvolvimento de Software Multiplataforma</td><td colspan="2">Nota:</td></tr>
<tr><td colspan="4">Autor(es):
Adriano Barros, Brendon Gomes, Elias Barbosa, Rafael Goncalves</td></tr>
<tr><td colspan="4">Orientador:
Profa. Me. Luana Lourenço</td></tr>
<tr><td colspan="4">Banca avaliadora: Ivan Carlos Pavão, Carlos Ronny de Souza</td></tr>
<tr><td colspan="4">Produto desenvolvido:
Plataforma Web Volts</td></tr>
<tr><td colspan="4">Especificações técnicas:
ReactJS, HTML, CSS, TypeScript, C#, ASP.NET Core, PostgreSQL.</td></tr>
<tr><td colspan="4">Palavras-chave/Descritores: Gestão de voluntariado. Desenvolvimento web. Plataforma digital. Metodologias ágeis. Acessibilidade. Sistemas de informação. Engenharia de software. Inclusão digital. Transformação digital. API. React. ASP.NET Core.</td></tr>
<tr><td>Edição
1</td><td>Nº de páginas
46</td><td>Nº do volume/parte
1</td><td>Área do conhecimento:
Tecnologia da Informação</td></tr>
<tr><td colspan="4">Observações/notas</td></tr>
</table>

Fonte: Estrutura adaptada pela autora com base na norma ABNT NBR 10719:2015.