

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
FACULDADE DE TECNOLOGIA DE CARAPICUÍBA
CURSO SUPERIOR EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**CARLOS ALEXANDRE ANDRADE DE SOUSA
LEANDRO NUNES DA SILVA
THALÍA DOS ANJOS SALDANHA
VITORIA REGIA DOS SANTOS MORAES**

**SUPPORT NEXUS: GERENCIAMENTO DE ORDENS DE SERVIÇO PARA TI EM
UMA FACULDADE DE TECNOLOGIA DE CARAPICUÍBA**

**CARAPICUÍBA
2025**

**CARLOS ALEXANDRE ANDRADE DE SOUSA
LEANDRO NUNES DA SILVA
THALÍA DOS ANJOS SALDANHA
VITORIA REGIA DOS SANTOS MORAES**

**SUPPORT NEXUS: GERENCIAMENTO DE ORDENS DE SERVIÇO PARA TI EM
UMA FACULDADE DE TECNOLOGIA DE CARAPICUÍBA**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Carapicuíba, do Centro Estadual de Educação Tecnológica Paula Souza, como exigência parcial para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Mário Marques.

**CARAPICUÍBA
2025**

**CARLOS ALEXANDRE ANDRADE DE SOUSA
LEANDRO NUNES DA SILVA
THALÍA DOS ANJOS SALDANHA
VITORIA REGIA DOS SANTOS MORAES**

**SUPPORT NEXUS: GERENCIAMENTO DE ORDENS DE SERVIÇO PARA TI EM
UMA FACULDADE DE TECNOLOGIA DE CARAPICUÍBA**

Aprovado em (dia) de (mês) de (ano)

Trabalho de Graduação apresentado á Faculdade de Tecnologia de Carapicuíba do centro estadual de Educação Tecnológica Paula Souza, como exigência parcial para a obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

BANCA EXAMINADORA

Prof. Dr. Mário Marques - Orientador
Fatec Carapicuíba

Prof. Dr. Luiz Sérgio De Souza
Fatec Carapicuíba

Prof. Msc. José Luís Barboza Lobianco
Fatec Carapicuíba

RESUMO

Esta monografia aborda os desafios relacionados à manutenção e suporte de computadores e periféricos em ambientes acadêmicos, com foco específico em uma Faculdade Pública do Estado de São Paulo. O sistema proposto visa facilitar a comunicação entre o corpo docente e a equipe de suporte de Tecnologia da Informação (TI), diminuindo o tempo de inatividade e aprimorando a experiência de ensino-aprendizagem. Entre os benefícios esperados, destacam-se a redução do tempo de inatividade, melhoria na qualidade das aulas, economia de recursos e aumento da satisfação dos usuários. O plano de ação delineado incluiu o desenvolvimento de um sistema com a aplicação de métodos ágeis, com medidas de monitoramento e melhorias contínuas para garantir sua eficácia a longo prazo. Como resultado, obteve-se o sistema *Support Nexus* com potencial para aprimorar o suporte de TI, possibilitando a redução do tempo de resolução dos chamados e contribuindo para uma experiência mais eficiente para seus usuários.

Palavras-chave: Suporte. Sistema de Informação. Tecnologia da Informação.

ABSTRACT

This monograph addresses the challenges related to the maintenance and support of computers and peripherals in academic environments, with a specific focus on a Public College in the State of São Paulo. The proposed system aims to facilitate communication between the teaching staff and the Information Technology (IT) support team, reducing downtime and improving the teaching–learning experience. Among the expected benefits are the reduction of downtime, improvement in class quality, resource savings, and increased user satisfaction. The outlined action plan included the development of a system using agile methods, with monitoring measures and continuous improvements to ensure its long-term effectiveness. As a result, the Support Nexus system was obtained, with the potential to enhance IT support, reduce the time required to resolve service requests, and contribute to a more efficient experience for its users.

Keywords: *Support. Information System. Information Technology.*

LISTA DE FIGURAS

<i>Figura 1 - Processo de abertura e tratamento das ordens de serviço.....</i>	18
<i>Figura 2 - Diagrama de Atividade do Support Nexus.....</i>	20
<i>Figura 3 - Cadastro de usuário</i>	34
<i>Figura 4 - Página de Login.....</i>	34
<i>Figura 5 - Recuperar senha</i>	35
<i>Figura 6 - Painel Professor.....</i>	37
<i>Figura 7 - Gerenciar usuários.....</i>	39

LISTA DE QUADROS

<i>Quadro 1: Qualidade da Informação</i>	17
<i>Quadro 2: Atribuições dos principais atores do sistema</i>	19
<i>Quadro 3: Entradas, Processos, Saídas e Objetivos</i>	19
<i>Quadro 4: Requisitos Funcionais.....</i>	21
<i>Quadro 5: Requisitos Não Funcionais</i>	23
<i>Quadro 6: Regras de Negócio</i>	25
<i>Quadro 7 - Rastreabilidade de Requisitos</i>	27

LISTA DE SIGLAS

API - *Application Programming Interface*

CSS - *Cascading Style Sheets*

CSV - *Comma-Separated Values*

CRUD - *Create Read Update Delete*

HTML - *HyperText Markup Language*

HTTPS - *Hypertext Transfer Protocol Secure*

JWT - *JSON Web Token*

PDF - *Portable Document Format*

RA - Registro Acadêmico

RF - Requisito Funcional

RN - Regra de Negócio

RNF - Requisito Não Funcional

SQL - *Structured Query Language*

TI - Tecnologia da Informação

UUID - *Universally Unique Identifier*

SUMÁRIO

1. INTRODUÇÃO	10
2. FUNDAMENTAÇÃO TEÓRICA	12
2.1 TECNOLOGIA: Tecnologia da Informação em Ambientes Acadêmicos	12
2.2 DO FRONTEND AO BACKEND: arquitetura e tecnologias fundamentais para o desenvolvimento	12
3. DESENVOLVIMENTO DA TEMÁTICA.....	15
3.1 Abordagem Metodológica	15
3.2 Integração com a Tecnologia da Informação.....	16
3.3 Sobre Critérios de Qualidade e o Desempenho do Sistema.....	16
3.4 Estrutura Operacional do Sistema e seus Subsistemas	18
3.5 Entradas, Processos, Saídas e Objetivos	19
3.6 Diagrama de Atividades.....	20
3.7 Requisitos.....	21
3.7.1 Requisitos funcionais	21
3.7.2 Requisitos não funcionais	23
3.8 Regras de Negócios.....	25
3.9 Histórias de Usuário.....	26
3.10 Tabela de Rastreabilidade	27
4 DESENVOLVIMENTO DO SISTEMA	32
4.1 Processo de desenvolvimento	32
4.1.1 – Sprint Semana 1 – Cadastro	33
4.1.2 – Sprint Semana 2 – <i>Login</i>	34
4.1.3 – Sprint Semana 3 – Reset de Senha.....	35
4.1.4 – Sprint Semana 4 – Ordens.....	35
4.1.5 – Sprint Semana 5 – Correção de Erros e Melhorias	36
4.1.6 – Sprint Semana 6 – Correções de Erros e Melhorias II.....	37
4.1.7 – Sprint Semana 7 – Meu Perfil	38
4.1.8 – Sprint Semana 8 – Admin	38
4.1.9 – Sprint Semana 9 – Toques Finais.....	39
5. CONCLUSÃO	40
REFERÊNCIAS.....	42

1. INTRODUÇÃO

O avanço acelerado da tecnologia tem transformado diversos setores, incluindo a educação. Nesse contexto, os sistemas de informação desempenham um papel essencial na otimização de processos acadêmicos, especialmente no suporte à Tecnologia da Informação. Segundo Laudon e Laudon (2022), um sistema de informação é um conjunto de componentes interconectados que coleta, processa, armazena e distribui informações para apoiar a tomada de decisões, o controle e a coordenação dentro de uma organização, além de auxiliar na análise de problemas, visualização de questões complexas e criação de novos produtos. Com base nessa definição, é possível compreender a sua importância como ferramenta estratégica que não apenas facilita a gestão das operações, mas também contribui significativamente para a melhoria dos processos dentro das instituições. Nesse sentido, esta pesquisa tem como objetivo o desenvolvimento do *Support Nexus*, um Sistema de Gerenciamento de Ordens de Serviço para a Fatec Carapicuíba, com a proposta de melhorar a eficiência operacional e a comunicação entre professores e a equipe de TI.

No desenvolvimento desta pesquisa, surgiram algumas questões: Quais funcionalidades o sistema deve oferecer para atender de maneira eficaz às necessidades tanto de professores quanto da equipe de TI? Quais desafios técnicos e de usabilidade podem surgir durante a criação do *Support Nexus*?

Parte-se do pressuposto teórico de que a ausência de um sistema estruturado para a criação e encaminhamento de chamados dificulta o desenvolvimento das atividades relacionadas ao uso dos computadores e periféricos nas salas, comprometendo a eficiência da equipe de suporte e gerando aumento no descontentamento de alunos e professores, devido à falta de agilidade na resolução de problemas causados por equipamentos defeituosos.

A hipótese central deste estudo é que a implementação de um Sistema de Gerenciamento de Ordens de Serviço para Suporte de TI na Fatec Carapicuíba tem potencial para melhorar a eficiência operacional. A ausência de um sistema estruturado para a criação e encaminhamento de chamados têm comprometido a agilidade nas manutenções de computadores e periféricos nas salas, o que impacta negativamente o trabalho da equipe de suporte e gera descontentamento entre alunos e professores. Acredita-se que, com a implementação desse sistema, será possível reduzir o tempo de resposta para as demandas de manutenção, melhorar a comunicação entre professores e a equipe de TI, minimizar o tempo

de inatividade das atividades educacionais e, assim, aprimorar a experiência de ensino e aprendizagem na instituição.

Em relação aos objetivos específicos, foram realizadas atividades como a análise de sistemas de gerenciamento de ordens de serviço, a fim de identificar funcionalidades e boas práticas relevantes, e, posteriormente, o desenvolvimento do sistema *Support Nexus*.

O desenvolvimento do sistema *Support Nexus* para a gestão de ordens de serviço na Fatec Carapicuíba envolveu a implementação de duas áreas principais: o *front-end* e o *back-end*.

O *front-end* do sistema corresponde à interface visual que permite ao usuário interagir com o *Support Nexus* de forma prática e intuitiva. Ele foi desenvolvido para oferecer navegação fluida e acesso facilitado às funcionalidades principais, como criação e acompanhamento de ordens de serviço, garantindo uma boa experiência em diferentes dispositivos. O *back-end* do sistema é responsável pelo processamento das informações e pelo funcionamento interno do *Support Nexus*. Ele gerencia os dados e garante que as ações realizadas pelos usuários sejam registradas e executadas com segurança e eficiência, sustentando todas as funcionalidades oferecidas pelo sistema. A integração entre o *front-end* e o *back-end* foi fundamental para garantir a funcionalidade completa do sistema, permitindo que o fluxo de informações entre as partes fosse realizado de maneira ágil e sem falhas. Essas duas áreas, embora distintas, trabalharam juntas para criar uma solução completa para o gerenciamento de ordens de serviço na Fatec Carapicuíba, com o objetivo de melhorar a comunicação entre professores e a equipe de TI, aumentar a eficiência operacional e aprimorar a experiência dos usuários.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, serão expostos os referenciais teóricos que sustentaram o desenvolvimento do sistema proposto, discutindo conceitos essenciais que nortearam a criação do *Support Nexus*.

2.1 TECNOLOGIA: Tecnologia da Informação em Ambientes Acadêmicos

Para embasar a proposta deste estudo, é essencial definir conceitos fundamentais que são pilares para a compreensão do papel da TI em ambientes organizacionais e acadêmicos.

A Tecnologia da Informação comprehende *hardware*, *software*, redes e bancos de dados, desempenhando um papel fundamental na otimização de processos, na gestão de recursos e na promoção da inovação. Em ambientes educacionais, a TI é especialmente relevante, pois facilita a gestão de informações acadêmicas, o suporte técnico e a integração de ferramentas digitais no processo de ensino-aprendizagem. Segundo McGee e Prusak (1994), a informação é um ativo organizacional estratégico, cujo valor aumenta com a precisão, a relevância e a combinação de diferentes fontes.

Por fim, a Gestão da Informação envolve processos como a identificação de necessidades, organização, distribuição e uso estratégico da informação. Conforme Choo (2006), essa gestão é essencial para promover a adaptação a mudanças e fornecer suporte à tomada de decisões, alinhando os objetivos organizacionais ao uso eficaz da informação e da tecnologia. Em um contexto acadêmico, a gestão eficiente da informação pode melhorar a qualidade dos serviços de suporte de TI, garantindo que as demandas dos usuários sejam atendidas de forma ágil e eficaz.

2.2 DO FRONTEND AO BACKEND: arquitetura e tecnologias fundamentais para o desenvolvimento

As tecnologias para o desenvolvimento web são inúmeras, porém, para a elaboração do *Support Nexus* o escopo foi limitado em: *Replit*, *HTML*, *CSS*, *JavaScript*, *Node.js* e *PostgreSQL*, tais tecnologias serão apresentadas a seguir.

Engkamat Yii e Sinatra Gran (2023) entendem o *Replit* como uma plataforma de codificação baseada em navegador que permite a colaboração em tempo real entre pares no

desenvolvimento, revisão e depuração de projetos de programação. Além de eliminar a necessidade de configuração local, a ferramenta oferece suporte a múltiplas linguagens de programação e funcionalidades como edição simultânea e compartilhamento de código. O *Replit* é altamente valorizado por sua acessibilidade, interatividade e por facilitar a colaboração na construção de projetos em equipe.

O *HyperText Markup Language* é a linguagem fundamental para a construção de qualquer página web. Ele é responsável por estruturar o conteúdo, fornecendo a base sobre a qual o restante da aplicação será desenvolvido. Como descreve Silva (2008), o *HTML* é utilizado para organizar elementos como textos, imagens, links e formulários, mas não lida com a parte visual ou com a interatividade da página. Para que essa estrutura ganhe forma e funcionalidade, é necessário utilizar outras tecnologias.

É nesse ponto que o *CSS* entra em cena, complementando o *HTML* ao aplicar estilos ao conteúdo. O *CSS* define como os elementos estruturados no *HTML* serão exibidos, controlando aspectos como cores, fontes, margens, alinhamentos e o *layout* da página. Além disso, o *CSS* permite criar designs responsivos, adaptando o site para diferentes dispositivos, como *desktops*, *tablets* e *smartphones* (Mcfarland, 2012). Dessa forma, o *HTML* e o *CSS* trabalham juntos para garantir tanto a estrutura quanto a aparência da página.

No entanto, a interatividade do site exige o uso do *JavaScript*, que adiciona dinamismo às páginas web. Enquanto o *HTML* e o *CSS* cuidam da estrutura e da apresentação, o *JavaScript* permite que o usuário interaja com a página. "A ampla maioria dos sites modernos usam *JavaScript* e todos [...] os navegadores modernos incluem *JavaScript*" (Flanagan, 2012). Além disso, o *JavaScript* é utilizado tanto no *front-end*, para melhorar a experiência do usuário, quanto no *back-end*, por meio do *Node.js*, tornando-se a linguagem que conecta e integra as diversas partes da aplicação.

O *Node.js* é a plataforma que possibilita a execução do *JavaScript* no lado do servidor. Segundo Pereira (2014) o *Node.js* é uma plataforma altamente eficiente para construir aplicações web escaláveis e rápidas. Ele adota um modelo de I/O não bloqueador e orientado a eventos, o que torna possível gerenciar várias requisições simultâneas com ótimo desempenho.

Para completar a estrutura da aplicação, o *PostgreSQL* será responsável pelo armazenamento e gerenciamento de dados essenciais. O sistema *open source* gerencia o banco de dados de maneira relacional e orientado a objeto, o *PostgreSQL* organiza dados em tabelas e utiliza a linguagem *SQL* para realizar operações de consulta, inserção, atualização e exclusão

(Worsley; Drake, 2002). Ele é ideal para armazenar informações de usuários, produtos, ou qualquer outro dado essencial para o funcionamento do *site*.

3. DESENVOLVIMENTO DA TEMÁTICA

O presente estudo foi motivado pela identificação da necessidade de um sistema que otimizasse a gestão de ordens de serviço no suporte de TI. A proposta é caracterizada como uma pesquisa aplicada, de natureza exploratória e descritiva, voltada para a solução de problemas concretos vivenciados no ambiente institucional. Como resultado desse processo, foi concebido o sistema *Support Nexus*, com o objetivo de atender às especificidades da instituição por meio de uma solução prática, eficaz e tecnicamente embasada.

3.1 Abordagem Metodológica

A principal base para o desenvolvimento do projeto foi a experiência empírica dos orientandos, que atuam ou atuaram na instituição e, por isso, puderam avaliar de forma prática o funcionamento do ambiente estudado. Complementarmente, recorreu-se à revisão bibliográfica para contextualizar o tema e apoiar a construção da solução proposta. Segundo Prodanov e Freitas (2013), a revisão bibliográfica é importante para identificar as bases teóricas que orientam a pesquisa e posicioná-la no contexto acadêmico. Nesse processo, foram consultados conceitos e boas práticas sobre o desenvolvimento de sistemas de informação voltados ao gerenciamento de ordens de serviço, evidenciando a relevância de soluções organizadas e acessíveis para melhorar o suporte técnico em ambientes educacionais.

Complementando essa fundamentação teórica, a vivência de colaboradores da equipe de TI da Fatec Carapicuíba permitiu compreender, com maior profundidade, as necessidades específicas da faculdade, como a implementação de processos de manutenção mais eficientes, a disponibilização de um sistema acessível para docentes e funcionários, a oferta de *feedback* eficaz tanto para os solicitantes quanto para a equipe técnica, além do controle detalhado das tarefas realizadas, do gerenciamento dos materiais utilizados e da identificação de indicadores que apontem pontos críticos no suporte.

Com base nesses dados e nas referências teóricas levantadas, foi desenvolvido um sistema utilizando princípios das metodologias ágeis, que segundo Chin (2004), configuraram-se como uma alternativa eficaz por permitir que as equipes se adaptem rapidamente às mudanças, sem comprometer a estrutura do projeto. A gestão ágil, nesse contexto, busca equilibrar flexibilidade e controle, oferecendo um processo prático, repetível e alinhado à realidade dinâmica das organizações contemporâneas.

Entre essas abordagens, adotou-se o *framework Scrum*, que organiza o desenvolvimento em ciclos curtos chamados *sprints*, nos quais são planejadas, executadas e revisadas pequenas partes do sistema (Schwaber e Sutherland, 2020). Esse modelo favorece a colaboração contínua, o feedback rápido e ajustes constantes, o que permitiu construir o *Support Nexus* de forma iterativa e alinhada às demandas reais da instituição.

O ciclo de trabalho para o desenvolvimento do projeto foi organizado em *sprints* semanais, nos quais as tarefas eram divididas entre os membros no início da semana e executadas ao longo dos dias seguintes. Ao final de cada ciclo, realizava-se uma revisão das entregas realizadas, permitindo identificar pontos de melhoria, validar o progresso e promover ajustes necessários. Com base nessa avaliação, um novo conjunto de atividades era definido para a sprint seguinte, garantindo um fluxo contínuo de desenvolvimento iterativo e colaborativo.

3.2 Integração com a Tecnologia da Informação

A Tecnologia da Informação foi um recurso essencial na metodologia aplicada à elaboração do sistema, viabilizando sua criação e execução. A construção de um ambiente digital facilitou a simulação do fluxo de comunicação entre docentes e a equipe técnica. Para Laudon e Laudon (2020), sistemas bem projetados promovem integração e acessibilidade, enquanto O'Brien (2004) ressalta sua importância para respostas ágeis e eficazes.

Nesse contexto, a TI não apenas sustentou a parte técnica do projeto, mas também funcionou como um instrumento metodológico, contribuindo diretamente para a avaliação da efetividade do sistema diante das necessidades práticas.

3.3 Sobre Critérios de Qualidade e o Desempenho do Sistema

Os critérios de qualidade da informação foram fundamentais para assegurar que o sistema atenda às expectativas da comunidade acadêmica. De acordo com Laudon e Laudon(2020), aspectos como precisão, completude, economia, flexibilidade, confiabilidade, relevância, simplicidade, pontualidade, verificabilidade, acessibilidade e segurança são indispensáveis para o desenvolvimento de sistemas eficazes. Esses elementos foram priorizados no desenvolvimento do sistema, garantindo que ele ofereça suporte à tomada de

decisão e às operações diárias da instituição.

Quadro 1: Qualidade da Informação

Atributo	Descrição
Precisão	Ausência de erros, essencial para a confiabilidade dos dados.
Completude	Inclusão de todas as informações necessárias para a tomada de decisões.
Economia	Equilíbrio entre o custo e o valor da informação.
Flexibilidade	Capacidade de adaptação às necessidades de diferentes usuários.
Confiabilidade	Relação com os métodos de coleta e as fontes utilizadas.
Relevância	Pertinência da informação para o processo decisório.
Simplicidade	Evitar complexidades desnecessárias, facilitando a compreensão.
Pontualidade	Entrega da informação no momento adequado.
Verificabilidade	Possibilidade de validação da informação.
Acessibilidade e Segurança	Garantia de que a informação estará disponível apenas para usuários autorizados.

Fonte: Elaborado pelos autores, 2025

No Quadro 1, estão sintetizados os principais critérios de qualidade da informação descritos por Laudon e Laudon (2020), os quais foram cuidadosamente considerados e aplicados ao longo do desenvolvimento do sistema.

Conforme O'Brien (2004), o desempenho do sistema foi monitorado e ajustado continuamente com base no *feedback* obtido nas fases de entrada, processamento e saída de dados. Essa abordagem garante não apenas a detecção e correção de falhas, mas também a adaptação do sistema a novas demandas.

Portanto, o controle de desempenho do sistema é essencial para garantir sua eficácia. O *feedback* gerado nas etapas de entrada, processamento e saída foi monitorado e avaliado, permitindo ajustes contínuos para garantir que o sistema atenda aos padrões estabelecidos. Esse controle não apenas corrige eventuais falhas, mas também capacita os usuários finais a compreender os componentes básicos dos sistemas de informação, promovendo melhores práticas de gestão e utilização.

3.4 Estrutura Operacional do Sistema e seus Subsistemas

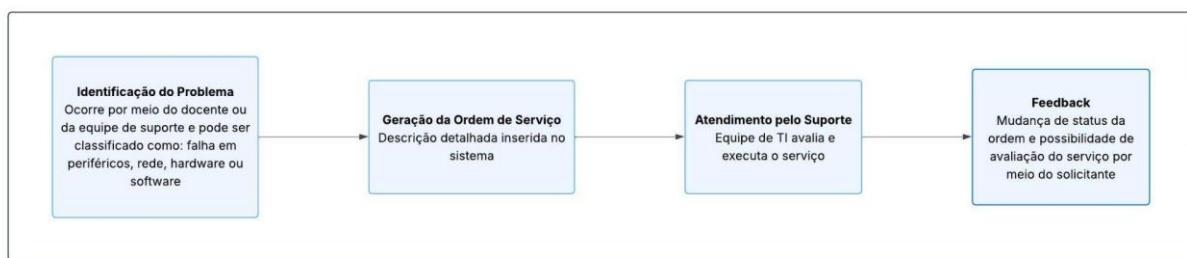
Para compreender o funcionamento do *Support Nexus*, é fundamental analisar a organização e a interação de seus elementos estruturais. Sua operação baseia-se em um fluxo lógico de ações articuladas com os atores responsáveis por cada etapa, proporcionando uma visão integrada e eficiente do processo de gerenciamento de ordens de serviço.

Entre as principais funcionalidades do sistema, destacam-se:

- Abertura de solicitações para substituição de periféricos com defeito;
- Instalação de *softwares* utilizados em atividades acadêmicas;
- Manutenção preventiva e corretiva de computadores;
- Troca de cabos de rede e conectores entre equipamentos, como televisores e computadores.

A aplicação segue uma sequência lógica de procedimentos, iniciando-se com o cadastro do usuário e finalizando com a conclusão do atendimento. A seguir, apresenta-se a Figura 1 que é uma representação de um fluxograma que ilustra o percurso das informações no sistema referente a abertura e resolução de ordens mostrando as ações correspondentes em cada etapa do processo.

Figura 1 - Processo de abertura e tratamento das ordens de serviço



Fonte: Elaborado pelos autores, 2025

O sistema é composto por três atores principais: o docente, a equipe de suporte e o administrador. No entanto, para compreender adequadamente o fluxo apresentado no diagrama anterior, é essencial destacar apenas os dois atores diretamente envolvidos nessa atividade: o corpo docente e a equipe de TI.

Ambos desempenham papéis complementares dentro do processo ilustrado no

fluxograma, seja na identificação do problema, na geração da ordem de serviço ou no atendimento realizado pelo suporte. Cada grupo contribui de maneira distinta para a eficiência operacional do sistema, assegurando que as solicitações sejam registradas, avaliadas e solucionadas com agilidade. O Quadro 2, apresentado a seguir, sintetiza de forma clara e objetiva as responsabilidades atribuídas a cada um desses atores no contexto do fluxo de atendimento.

Quadro 2: Atribuições dos principais atores do sistema

Atores	Responsabilidades
Corpo Docente	<ul style="list-style-type: none"> - Identificar falhas em equipamentos, rede ou softwares nas salas de aula. - Registrar a ordem de serviço no sistema. - Descrever o problema detalhadamente. - Avaliar ordens concluídas.
Equipe de TI	<ul style="list-style-type: none"> - Identificar falhas em equipamentos, rede ou softwares nas salas de aula. - Registrar a ordem de serviço no sistema. - Monitorar as ordens de serviço recebidas. - Encerrar as ordens descrevendo como solucionou o problema.

Fonte: Elaborado pelos autores, 2025

3.5 Entradas, Processos, Saídas e Objetivos

Para compreender o funcionamento do sistema de forma clara e objetiva, foi realizada a identificação dos principais elementos que compõem sua lógica operacional. Conforme destacam Stair e Reynolds (2011), a eficiência de um sistema está diretamente relacionada à integração coerente entre seus componentes estruturais, como as entradas, os processos internos e os resultados obtidos. A seguir, o Quadro 3 apresenta uma visão consolidada desses componentes em adição aos objetivos de cada ação:

Quadro 3: Entradas, Processos, Saídas e Objetivos

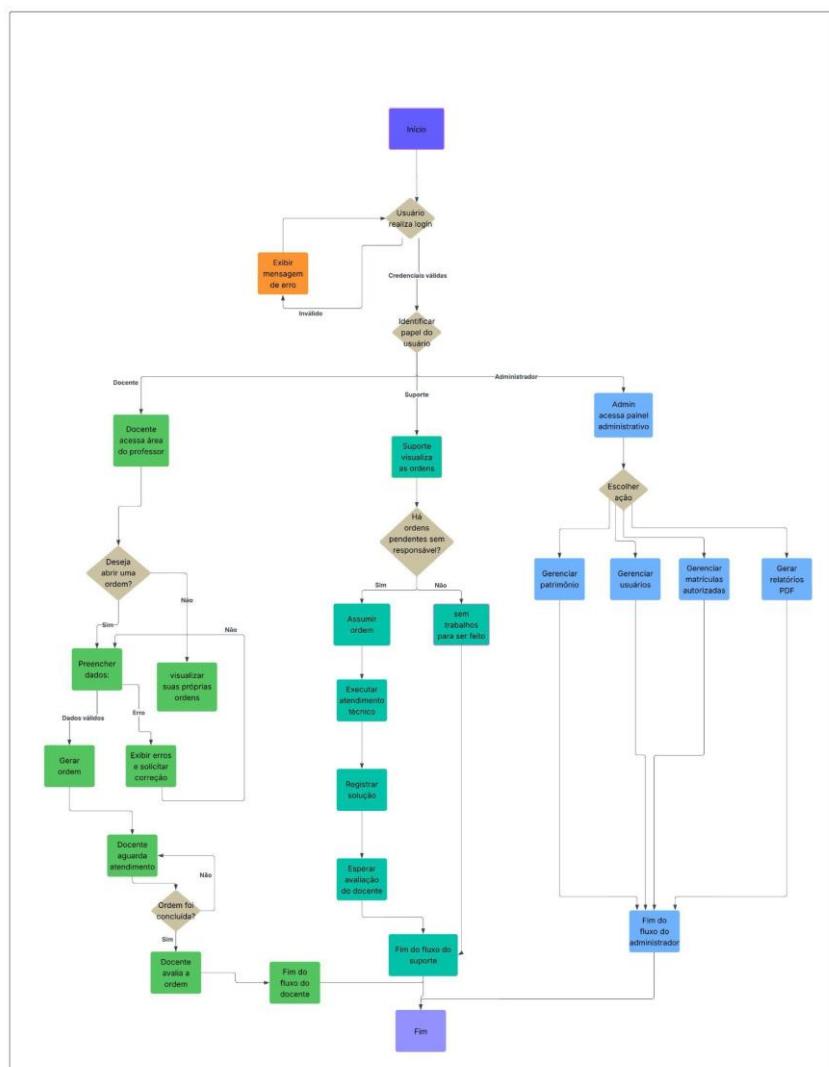
Entradas	Processos	Saídas	Objetivos
<ul style="list-style-type: none"> - Dados dos atores (nome, matrícula, e-mail institucional, senha) 	<ul style="list-style-type: none"> - Registrar dados dos atores no sistema 	<ul style="list-style-type: none"> - Credenciais cadastradas e usuários autorizados a acessar o sistema. 	<ul style="list-style-type: none"> - Dinamizar a comunicação de problemas entre os docentes e a equipe de suporte de TI
<ul style="list-style-type: none"> - Ordem de serviço (relatar problema, solicitar instalação) 	<ul style="list-style-type: none"> - criar ordem - assumir ordem - concluir ordem - avaliar ordem 	<ul style="list-style-type: none"> - ordem gerada - ordem assumida - ordem concluída - ordem avaliada 	<ul style="list-style-type: none"> - Gerar o ciclo de atendimento a solicitações de serviço

Fonte: Elaborado pelos autores, 202

3.6 Diagrama de Atividades

Para representar o fluxo operacional do sistema de gerenciamento de ordens de serviço, foi elaborado o Diagrama de Atividades que ilustra, de forma sequencial, as principais etapas envolvidas no atendimento das solicitações. De acordo com Pressman (2011), os diagramas de atividades são ferramentas úteis para representar o fluxo de ações em um sistema, facilitando a visualização dos processos envolvidos de forma sequencial e lógica. Observe a Figura 2 a seguir:

Figura 2 - Diagrama de Atividade do Support Nexus



Fonte: Elaborado pelos autores, 2025

A representação dessas etapas no Diagrama de Atividades ajuda a visualizar como as ações fluem entre os atores e qual o funcionamento do sistema.

3.7 Requisitos

Segundo Sommerville (2011), os requisitos são elementos que representam as características e restrições definidas por clientes e usuários para um sistema. Eles desempenham um papel fundamental no processo de desenvolvimento, pois orientam o projeto com base nas necessidades identificadas. Esses requisitos costumam ser organizados em duas categorias principais: funcionais e não funcionais, que serão detalhadas a seguir em conjunto com as regras de negócio.

3.7.1 Requisitos funcionais

Requisitos funcionais descrevem as funcionalidades específicas que o sistema deve oferecer para atender às necessidades dos usuários (Vazquez e Simões, 2016). Eles indicam o comportamento esperado do sistema diante de determinadas ações ou eventos, como o cadastro de usuários, abertura de chamados, envio de notificações, entre outros. Em outras palavras, são as tarefas que o sistema deve ser capaz de executar para cumprir seus objetivos.

Quadro 4: Requisitos Funcionais

Identificação	Descrição
RF001	O sistema deve permitir o cadastro de professores e membros da equipe de TI.
RF002	O sistema deve permitir <i>login</i> com autenticação de usuário e senha.
RF003	O sistema deve permitir ao professor poder criar uma nova ordem de serviço informando o problema e sala.
RF004	O sistema deve permitir ao professor poder criar uma nova ordem de serviço solicitando instalações de aplicativos.
RF005	O sistema deve listar todas as ordens de serviço criadas pelo usuário autenticado
RF006	O sistema deve encaminhar automaticamente a ordem de serviço para a equipe de TI após a criação.
RF007	O sistema deve permitir a equipe de TI deve conseguir visualizar todas as

	ordens pendentes.
RF008	O sistema deve permitir a equipe de TI deve conseguir atualizar o status da ordem de serviço (Em andamento, resolvido, etc.).
RF009	O sistema deve permitir o registro de comentários dentro da ordem de serviço.
RF010	O sistema deve permitir que o professor receba notificações sobre atualizações na sua ordem.
RF011	O sistema deve permitir a filtragem de ordens de serviço por status ou data.
RF012	O sistema deve possuir um painel de administrador.
RF013	O sistema deve permitir que o professor avalie o serviço após seu encerramento.
RF014	O sistema deve permitir o upload de imagens para ilustrar o problema relatado.
RF015	O sistema deve exibir um painel administrativo com estatísticas de uso.
RF016	O sistema deve permitir a redefinição de senha mediante um link válido recebido por e-mail.
RF017	O sistema deve permitir que administradores editem e excluam ordens de serviço.
RF018	O sistema deve permitir que administradores criem novos usuários com validação de matrícula, e-mail e cargo.
RF019	O sistema deve permitir que administradores atualizem ou excluam dados de usuários.
RF020	O sistema deve permitir que usuários alterem sua própria senha mediante validação da senha atual.
RF021	O sistema deve permitir a criação, atualização e exclusão de itens de patrimônio
RF022	O sistema deve gerar relatórios em PDF
RF023	O sistema deve permitir a importação em lote de matrículas autorizadas através de arquivos CSV
RF024	O sistema deve permitir upload de foto de perfil.
RF025	O sistema deve registrar automaticamente a data de criação de cada ordem de serviço.

RF026	A visualização dos patrimônios cadastrados é restrita a usuários com o papel de administrador.
RF027	Ordens encerradas não podem ser reabertas.
RF028	O sistema deve aceitar apenas cadastros com endereços de <i>e-mail</i> nos domínios @fatec.sp.gov.br ou @proton.me.
RF029	O sistema deve atualizar automaticamente o status de ordens vencidas para Não Concluída
RF030	O sistema deve criar alertas automaticamente baseados em dias úteis restantes e ausência de responsável
RF031	Cada e-mail cadastrado deve ser único no sistema.
RF032	Ordens com status Não Concluída não podem ser editadas, assumidas ou ter seu status alterado
RF033	Apenas usuários com papel administrador podem editar ordens de serviço, exceto aquelas com status Não Concluída.

Fonte: Elaborado pelos autores, 2025

3.7.2 Requisitos não funcionais

Requisitos não funcionais definem as restrições do sistema, ou seja, descrevem como ele deve se comportar ao executar suas funcionalidades. Esses requisitos não estão ligados diretamente às funções do sistema, mas sim a características como desempenho, segurança, usabilidade, compatibilidade, disponibilidade e manutenibilidade. Eles garantem que o sistema funcione de maneira eficiente e confiável, mesmo em diferentes contextos de uso (Vazquez e Simões, 2016).

Quadro 5: Requisitos Não Funcionais

Identificação	Descrição
RNF001	O sistema deve ser acessível por navegador.
RNF002	O tempo de carregamento das páginas do sistema não deve exceder 3 segundos em uma conexão de banda larga estável.

RNF003	A comunicação entre <i>front-end</i> e <i>back-end</i> deve ocorrer por meio de <i>API</i> utilizando protocolo <i>HTTPS</i> quando hospedado em ambiente de produção.
RNF004	O sistema deve manter o tempo de resposta estabelecido no RNF002 mesmo com um volume de armazenamento superior a 10.000 ordens de serviço registradas no banco de dados.
RNF005	O sistema deve integrar-se com uma API de e-mail para envio de notificações.
RNF006	A autenticação deve ser realizada por meio de <i>token JWT</i> , garantindo acesso apenas a usuários com credenciais válidas.
RNF007	O sistema deve ter um método básico de autenticação para garantir que apenas usuários autorizados acessem suas funcionalidades.
RNF008	A interface gráfica deve seguir as diretrizes de usabilidade e componentes visuais do Material Design, garantindo consistência na tipografia, cores e feedback visual ao usuário.
RNF009	O sistema deve funcionar nos principais navegadores: <i>Chrome</i> , <i>Firefox</i> e <i>Edge</i> .
RNF010	O sistema deve permitir a realização de <i>backup</i> automático diário.
RNF011	As atualizações de versão do sistema não devem comprometer mais do que 0,5% do tempo operacional mensal e devem ser realizadas exclusivamente em janelas de manutenção pré-agendadas (finais de semana ou horário noturno).
RNF012	A arquitetura do sistema deve permitir expansão futura de funcionalidades sem necessidade de reescrever toda a aplicação.
RNF013	O <i>front-end</i> deve ser responsivo, adaptando-se a diferentes tamanhos de tela.
RNF014	Mensagens de erro e sucesso devem ser exibidas de forma comprehensível ao usuário.
RNF015	As senhas devem ser criptografadas utilizando o algoritmo <i>bcrypt</i> .

Fonte: Elaborado pelos autores, 2025

3.8 Regras de Negócios

Diferente dos requisitos funcionais, que descrevem o que o sistema deve fazer, e dos requisitos não funcionais, que definem como essas funcionalidades devem ser executadas, as regras de negócio determinam as condições, restrições e critérios específicos da organização sob as quais essas funcionalidades devem operar.

Quadro 6: Regras de Negócio

Identificação	Descrição
RN001	Para salvar uma ordem de serviço, é obrigatório informar os campos: “tipo de ambiente”, “local” e “tipo de problema”.
RN002	Somente usuários com o papel de suporte podem concluir ordens de serviço.
RN003	No momento da abertura, o usuário deve selecionar o tipo da ordem de serviço, definindo-a como Problema ou Instalação, para que o atendimento seja direcionado adequadamente.
RN004	A exclusão de usuários é permitida apenas para contas com o papel de administrador.
RN005	A matrícula e nome informado no cadastro deve existir na tabela de matrículas autorizadas com status ativo
RN006	A matrícula de professores deve conter exatamente 5 dígitos numéricos. A matrícula de suporte deve conter exatamente 13 dígitos numéricos.
RN007	A data limite de uma ordem é calculada como 9 dias corridos após a data de criação.
RN008	Ordens com status diferente de Concluída ou Não Concluída que ultrapassam a data limite recebem status Não Concluída.
RN009	A prioridade da ordem é definida com base nos dias úteis restantes até o prazo final
RN010	Ao concluir uma ordem, é obrigatório informar a descrição da solução aplicada.

RN011	Relatórios de desempenho devem calcular métricas de ordens concluídas, não concluídas, taxa de sucesso e avaliação média por técnico.
-------	---

Fonte: Elaborado pelos autores, 2025

3.9 Histórias de Usuário

Histórias de usuário: são descrições simples e informais de funcionalidades que devem ser implementadas em um sistema, redigidas sob a perspectiva do usuário final. Segundo Cohn (2004), seu objetivo é capturar o que o usuário deseja realizar com o sistema e qual valor essa ação proporciona. Utilizadas amplamente em metodologias ágeis, como o *Scrum*, essas histórias seguem geralmente um formato padrão:

Como [tipo de usuário], **quero** [ação], **para** [benefício].

1. **Como** professor, **quero** cadastrar minha conta no sistema **para** poder acessar e criar ordens de serviço.
2. **Como** membro da equipe de TI, **quero** ter acesso ao login com autenticação de usuário e senha **para** garantir que apenas pessoas autorizadas utilizem o sistema.
3. **Como** professor, **quero** criar uma ordem de serviço detalhando o problema, a sala e a descrição **para** que a equipe de TI saiba o que precisa ser resolvido.
4. **Como** professor, **quero** poder visualizar todas as ordens de serviço que criei **para** acompanhar o andamento de cada uma delas.
5. **Como** membro da equipe de TI, **quero** que a ordem de serviço seja encaminhada automaticamente após sua criação **para** que eu possa começar a trabalhar nela sem atrasos.
6. **Como** membro da equipe de TI, **quero** visualizar todas as ordens de serviço pendentes **para** priorizar as tarefas mais urgentes.
7. **Como** membro da equipe de TI, **quero** atualizar o status da ordem de serviço (Em andamento, resolvido, etc.) **para** manter o professor informado sobre o progresso.
8. **Como** professor, **quero** registrar comentários na ordem de serviço **para** fornecer detalhes adicionais ou fazer perguntas à equipe de TI.
9. **Como** professor, **quero** receber poder ver sempre que houver atualizações em minha ordem de serviço **para** estar sempre informado sobre o progresso.

10. **Como** professor, **quero** filtrar as ordens de serviço **para** encontrar rapidamente as ordens relevantes para minha atividade.
11. **Como** administrador da equipe de TI, **quero** gerar relatórios com dados das ordens de serviço **para** analisar a performance do sistema e a eficiência da equipe de TI.
12. **Como** professor, **quero** avaliar o serviço após o encerramento da ordem **para** fornecer feedback sobre a qualidade do atendimento recebido.
13. **Como** professor, **quero** enviar imagens junto à ordem de serviço **para** ilustrar melhor o problema e facilitar a resolução.
14. **Como** administrador membro da equipe de TI, **quero** poder gerenciar os usuários do sistema (incluir, editar, excluir) **para** manter o controle sobre quem tem acesso às funcionalidades do sistema.
15. **Como** administrador, **quero** poder cadastrar os itens de patrimônio no sistema **para** manter o controle sobre os patrimônios da instituição.

Tendo em vista as histórias de usuário apresentadas, que descrevem as funcionalidades do sistema sob a perspectiva dos diferentes atores, torna-se necessário estabelecer um mecanismo que garanta a correspondência entre essas histórias e os requisitos especificados anteriormente. Nesse contexto, a tabela de rastreabilidade surge como ferramenta essencial para mapear essas relações.

3.10 Tabela de Rastreabilidade

A tabela de rastreabilidade constitui-se numa técnica que estabelece o cruzamento entre os requisitos funcionais e não funcionais especificados no projeto e sua relação com as histórias de usuário. Essa abordagem possibilita o registro explícito dos relacionamentos entre requisitos e as histórias, facilitando a identificação de lacunas e garantindo que todas as funcionalidades planejadas sejam implementadas no sistema.

A rastreabilidade de requisitos é uma característica de sistemas nos quais requisitos são claramente ligados às suas fontes e aos artefatos criados durante o ciclo de vida de desenvolvimento do sistema baseado nesses requisitos. Ela estabelece um elo entre mudanças das necessidades dos usuários e a evolução dos sistemas de computação, sendo uma base para o gerenciamento do conhecimento organizacional (RAMESH e JARKE, 2001).

Quadro 7 - Rastreabilidade de Requisitos

ID	História de Usuário (HU)	Requisitos Funcionais (Critérios de Aceitação)	Requisitos Não Funcionais (Qualidade/Restrições)
----	-----------------------------	---	---

HU01	Como professor, quero cadastrar minha conta no sistema para poder acessar e criar ordens de serviço.	RF001 (Cadastro de prof/TI) RF024 (Upload de foto) RF028 (Validar domínio e-mail) RF031 (E-mail único)	RF015 (Criptografia senha/bcrypt) RNF014 (Mensagens de erro/sucesso)
HU02	Como membro da equipe de TI, quero ter acesso ao login com autenticação de usuário e senha para garantir que apenas pessoas autorizadas utilizem o sistema.	RF002 (Login usuário/senha) RF016 (Redefinição de senha) RF020 (Alterar própria senha)	RNF006 (Token JWT) RNF007 (Autenticação básica) RNF015 (Criptografia)
HU03	Como professor, quero criar uma ordem de serviço detalhando o problema, a sala e a descrição para que a equipe de TI saiba o que precisa ser resolvido.	RF003 (Criar OS com sala/descrição) RF004 (Solicitar instalação apps) RF025 (Data criação automática)	RNF004 (Desempenho com volume dados) RNF008 (Material Design) RNF014 (Feedback visual)
HU04	Como professor, quero poder visualizar todas as ordens de serviço que criei para acompanhar o andamento de cada uma delas.	RF005 (Listar OS do usuário) RF011 (Filtro status/data)	RNF002 (Carregamento < 3s) RNF013 (Responsividade)
HU05	Como membro da equipe de TI, quero que a ordem de serviço seja encaminhada	RF006 (Encaminhamento auto) RF030 (Alertas de prazo/resp.)	RNF005 (API de E-mail) RNF003 (Comunicação via API)

	automaticamente após sua criação para que eu possa começar a trabalhar nela sem atrasos.		
HU06	Como membro da equipe de TI, quero visualizar todas as ordens de serviço pendentes para priorizar as tarefas mais urgentes.	RF007 (Visualizar pendentes) RF029 (Atualizar vencidas auto)	RNF002 (Carregamento < 3s) RNF004 (Volume > 10k registros)
HU07	Como membro da equipe de TI, quero atualizar o status da ordem de serviço (Em andamento, resolvido, etc.) para manter o professor informado sobre o progresso.	RF008 (Atualizar status) RF027 (Não reabrir encerradas) RF032 (Bloqueio ed. "Não Concluída") RF033 (Permissão edição admin)	RNF003 (HTTPS) RNF014 (Feedback de sucesso)
HU08	Como professor, quero registrar comentários na ordem de serviço para fornecer detalhes adicionais ou fazer perguntas à equipe de TI.	RF009 (Registrar comentários)	RNF013 (Layout responsivo no chat) RNF014 (Feedback envio)
HU09	Como professor, quero receber poder ver sempre que houver atualizações em minha ordem de serviço para estar sempre informado sobre o progresso.	RF010 (Receber notificações)	RNF005 (Envio e-mail) RNF014 (Notificação visual no sistema)
HU10	Como professor, quero	RF011 (Filtragem status/data)	RNF002 (Tempo resposta)

	filtrar as ordens de serviço para encontrar rapidamente as ordens relevantes para minha atividade.		filtro) RNF008 (Usabilidade dos filtros)
HU11	Como administrador da equipe de TI, quero gerar relatórios com dados das ordens de serviço para analisar a performance do sistema e a eficiência da equipe de TI.	RF015 (Painel estatísticas) RF022 (Relatório PDF)	RNF002 (Geração ágil) RNF009 (Compatibilidade browsers)
HU12	Como professor, quero avaliar o serviço após o encerramento da ordem para fornecer feedback sobre a qualidade do atendimento recebido.	RF013 (Avaliação pós-encerramento)	RNF008 (Componentes visuais/estrelas) RNF014 (Confirmação)
HU13	Como professor, quero enviar imagens junto à ordem de serviço para ilustrar melhor o problema e facilitar a resolução.	RF014 (Upload imagens)	RNF004 (Armazenamento) RNF012 (Expansão futura/storage)
HU14	Como administrador membro da equipe de TI, quero poder gerenciar os usuários do sistema (incluir, editar, excluir) para manter o controle sobre quem tem acesso às funcionalidades do sistema.	RF012 (Painel admin) RF018 (Criar usuários/cargos) RF019 (Atualizar/excluir dados) RF023 (Importação CSV lote)	RNF006 (Segurança JWT) RNF015 (Segurança dados) RNF010 (Backup dados)
HU15	Como administrador,	RF021 (Gerenciar Patrimônio)	RNF007 (quem pode acessar

	<p>quero poder cadastrar os itens de patrimônio no sistema para manter o controle sobre os patrimônios da instituição.</p>	<p>RF026 (Visualização restrita de patrimônio)</p>	<p>(dados específicos)</p>
--	--	---	----------------------------

Fonte: Elaborado pelos autores, 2025

4 DESENVOLVIMENTO DO SISTEMA

O desenvolvimento do sistema *Support Nexus* foi conduzido por meio de um planejamento estruturado e iterativo, garantindo a integração eficiente entre *front-end*, *back-end* e banco de dados. O objetivo central desta etapa concentrou-se em traduzir os requisitos levantados em uma solução funcional e escalável para a gestão de ordens de serviço da instituição, aplicando na prática os conceitos estudados ao longo do projeto.

A divisão de responsabilidades e a organização metodológica do processo de desenvolvimento foram fundamentais para que cada fase evoluísse de forma colaborativa, mantendo-se alinhada aos objetivos do sistema e aos padrões de qualidade estabelecidos. O *front-end* concentrou-se na construção de interfaces responsivas e na experiência do usuário, priorizando navegação intuitiva, acessibilidade e design consistente em diferentes dispositivos. O *back-end* ficou responsável pelo processamento das informações, implementação das regras de negócio e comunicação segura com o banco de dados.

4.1 Processo de desenvolvimento

O desenvolvimento do *front-end* foi estruturado em quatro etapas principais, o que permitiu uma evolução gradual e organizada das funcionalidades. O primeiro etapa concentrou-se na criação da estrutura base do sistema e das páginas fundamentais para o funcionamento inicial, como a página inicial com a apresentação do sistema, a página sobre com a explicação do propósito da plataforma, as páginas de *login* e cadastro para gerenciamento de acesso dos usuários e a página de termos de uso com as regras e políticas da aplicação.

A segunda etapa voltou-se para a área do professor, destinada aos usuários docentes da plataforma. Nessa fase, foram desenvolvidas as páginas de painel do professor, que oferece uma visão geral das atividades e ordens relacionadas ao docente, de perfil do professor para edição de dados pessoais e profissionais, de criação de ordens de serviço e de acompanhamento das solicitações já realizadas por meio da página minhas ordens.

Em seguida, a terceira etapa abordou o desenvolvimento da área do suporte, direcionada à equipe responsável pelo atendimento das solicitações. Foram criadas as páginas de painel do suporte, que reúne informações sobre as ordens em andamento, perfil do suporte, criação de

novas ordens quando necessário pela equipe, listagem de todas as ordens cadastradas no sistema com filtros e recursos de acompanhamento e, por fim, a página minhas ordens, destinada ao atendente que deseja visualizar as solicitações abertas por ele próprio.

A quarta e última etapa do front-end foi a elaboração da área do administrador, que possui páginas de visualização das ordens do sistema, gerenciamento de patrimônio, geração de relatórios e gerenciamento de usuários. Essas páginas foram projetadas para oferecer ao administrador controle completo sobre ordens, usuários, relatórios e patrimônio.

Após a conclusão do desenvolvimento do *front-end*, iniciou-se a etapa de *back-end*, que foi organizada em um total de nove sprints. Cada sprint teve duração de uma semana e foi dedicada à integração e implementação das funcionalidades correspondentes a uma parte específica do *front-end*. A seguir, encontram-se a descrição detalhada do que foi desenvolvido no período dessas oito semanas.

4.1.1 – Sprint Semana 1 – Cadastro

Na primeira semana foi desenvolvido o *back-end* responsável pelo processo de cadastro de usuários, implementando mecanismos de segurança e validação. Para isso, foram criadas duas tabelas no banco de dados: a tabela *matriculas_autorizadas*, destinada a armazenar os dados das pessoas previamente autorizadas a se registrar na plataforma (incluindo nome, e-mail institucional, RA ou matrícula e papel no sistema); e a tabela *users*, responsável por guardar as credenciais criadas pelos usuários, com senha criptografada utilizando *hash bcrypt* para garantir segurança.

A lógica definida para essa etapa consiste na verificação, no momento do cadastro, das informações fornecidas pelo usuário (nome completo, e-mail institucional, RA, número de matrícula e senha). Quando o formulário é submetido, o sistema consulta a tabela *matriculas_autorizadas* para confirmar se o RA (no caso de alunos que atuam no suporte) ou o número de matrícula (no caso de professores) existe no banco e qual é o papel (*role*) associado àquele identificador. Caso exista e os dados correspondam, o cadastro é autorizado e as credenciais são salvas na tabela *users* com a senha devidamente criptografada. Caso contrário, o sistema bloqueia a criação da conta e retorna uma mensagem de erro específica. É possível verificar como a página de cadastro foi estruturada a partir da figura 3, que será apresentada a seguir.

Figura 3 - Cadastro de usuário

A interface de usuário para cadastro de usuário é exibida em um formulário com fundo escuro. No topo, uma barra contém o ícone de usuário e o link "Cadastrar". O formulário principal, intitulado "Cadastro de Usuário", instrui o usuário a "Preencha todos os campos para criar sua conta". Ele contém quatro campos: "Nº Matrícula" com o valor "1234567890123", "Senha" com placeholder "Mínimo 6 caracteres", "Nome Completo" com placeholder "Nome completo" e "Confirmar Senha" com placeholder "Digite novamente". Abaixo, há um campo para "Email Institucional" com placeholder "seuemail@fatec.sp.gov.br". No final do formulário, há dois botões: "Cadastrar Usuário" (em destaque) e "Cancelar".

Fonte: Elaborado pelos autores, 2025

4.1.2 – Sprint Semana 2 – Login

Na segunda semana, foi desenvolvido o funcionamento do sistema de *login*. A lógica consiste em verificar, sempre que o usuário insere suas credenciais, se há correspondência na tabela *users*. Em caso afirmativo, o acesso é liberado e o usuário é redirecionado para a interface correspondente ao seu papel no sistema. O controle de acesso é baseado em papéis (*roles*), definidos na tabela *matriculas_autorizadas*, que podem ser suporte, professor ou administrador. Na figura 4 encontra-se a captura da tela de *login*.

Figura 4 - Página de Login

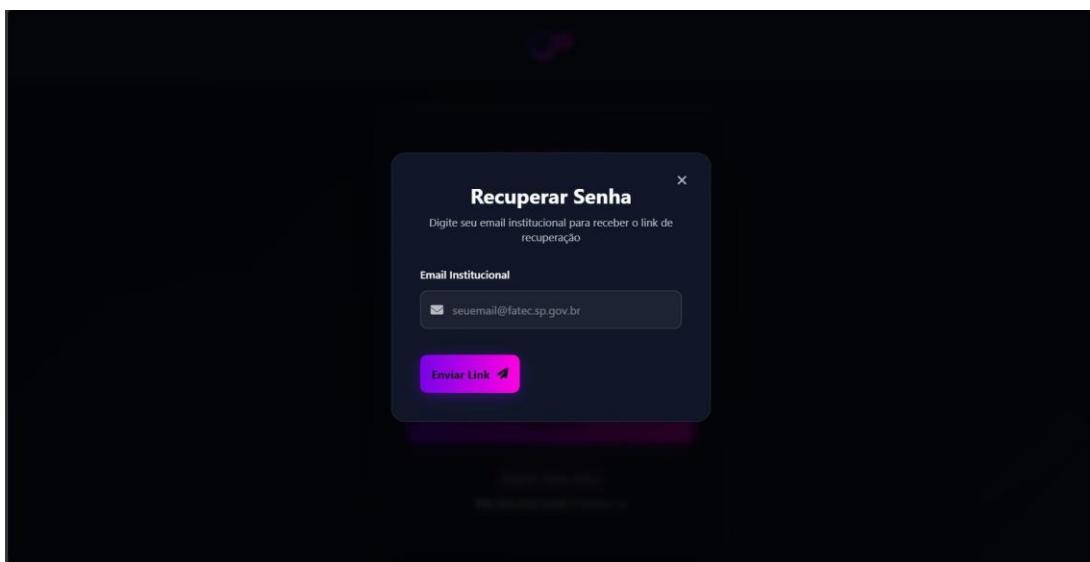
A interface de usuário para login é exibida em um formulário com fundo escuro. No topo, uma barra contém o ícone de usuário e o link "Entrar". O formulário principal, intitulado "LOGIN", instrui o usuário a "Acesse sua conta para gerenciar ordens de serviço". Ele contém dois campos: "Email Institucional" com placeholder "seuemail@fatec.sp.gov.br" e "Senha" com placeholder "Digite sua senha". Abaixo, há um botão "Entrar" com ícone de seta. À base do formulário, há links para "Esqueci minha senha" e "Não tem uma conta? Cadastre-se".

Fonte: Elaborado pelos autores, 2025

4.1.3 – Sprint Semana 3 – Reset de Senha

Na terceira semana foi criada a tabela `ResetSenha`, responsável por armazenar temporariamente a nova senha digitada pelo usuário, permitindo sua substituição na tabela `users`. O processo inicia-se no modal de redefinição de senha, no qual o usuário informa o *e-mail* cadastrado. Como é demonstrado na figura 5:

Figura 5 - Recuperar senha



Fonte: Elaborado pelos autores, 2025

A partir desse dado, o sistema valida se o e-mail existe na base de usuários e, em caso positivo, gera um *token* único e temporário (válido por 1 hora). Por meio da plataforma *Resend*, é enviado automaticamente um e-mail contendo um *link* seguro de recuperação. Ao acessar o *link*, o usuário é direcionado para uma página específica do *site* onde pode definir sua nova senha, que será criptografada e atualizada na tabela `users`, substituindo a anterior.

4.1.4 – Sprint Semana 4 – Ordens

A quarta semana foi dedicada ao desenvolvimento do *back-end* do sistema de ordens de serviço, que representa o núcleo funcional da aplicação. Para isso, foram criadas cinco tabelas relacionais: `ordens`, que funciona como tabela principal contendo dados comuns a todas as solicitações (*id*, usuário solicitante, data de criação, *status*, prioridade, técnico responsável, local, avaliação e etc); `ordens_problemas`, destinada a armazenar informações específicas relacionadas a chamados de problemas (descrição detalhada, componente afetado);

ordens_instalacoes, voltada para solicitações de instalação de programas (nome do programa, versão desejada, laboratório de destino); ordens_anexos, responsável por armazenar os caminhos das imagens enviadas pelo usuário como evidências; e ordens_alertas, responsável por registrar e controlar os alertas automáticos das ordens que serão enviados para a equipe de suporte via notificações internas.

As solicitações podem envolver problemas em componentes das salas (como computadores, projetores, cabos de rede ou periféricos) ou a instalação de programas em determinados ambientes. As ordens da categoria problema permitem o envio de até três imagens para auxiliar na descrição visual do que está acontecendo, facilitando o diagnóstico pela equipe técnica. Cada solicitação criada recebe um identificador único e *timestamp* de criação.

Usuários com papel de professor podem visualizar apenas as ordens que eles próprios criaram, garantindo privacidade. Usuários com papel de suporte ou administrador têm acesso à visualização de todas as ordens registradas no sistema, com diferentes níveis de permissão para edição. Na interface do suporte, as ordens precisam ser explicitamente aceitas por um atendente; ao serem aceitas, mudam automaticamente de status de "em aberto" para "em andamento", e o campo técnico_responsavel é preenchido. Apenas o técnico que aceitou a ordem pode finalizá-la ou solicitar mais informações ao solicitante. Caso a ordem não seja resolvida em até 9 dias corridos após ser aberta, o sistema automaticamente altera seu status para "não concluída".

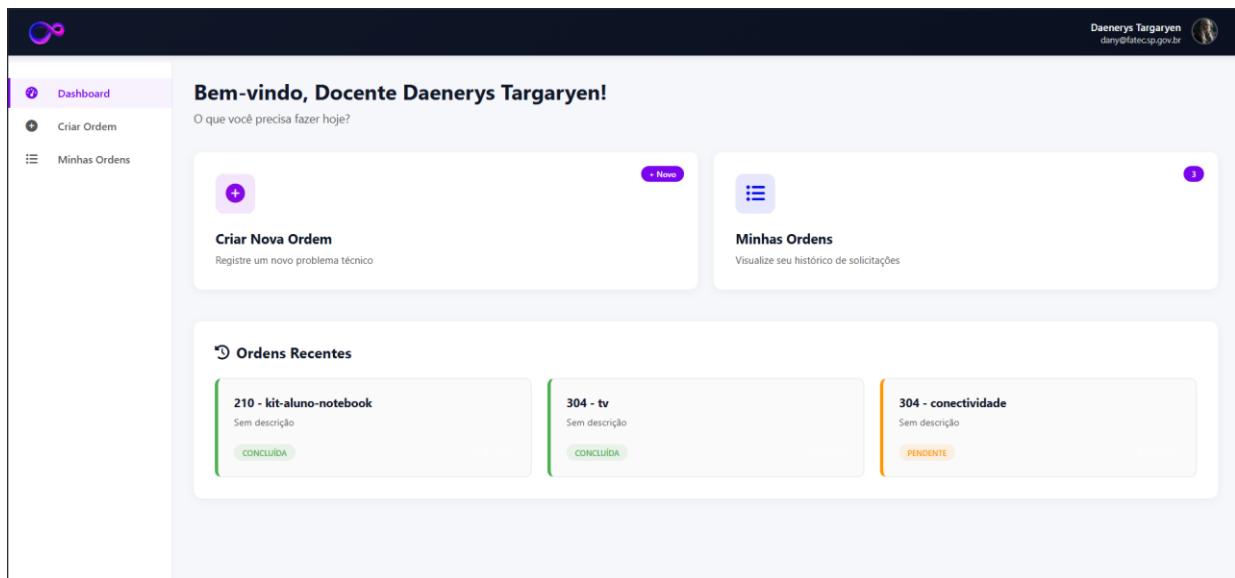
4.1.5 – Sprint Semana 5 – Correção de Erros e Melhorias

Após quatro semanas de desenvolvimento do *back-end*, tornou-se necessária uma revisão completa do sistema para garantir qualidade, identificar inconsistências e propor melhorias. Durante essa semana, foram anotados todos os pontos que precisavam de correção. O código do *back-end* foi refatorado e modularizado, e diversos ajustes foram realizados no *front-end*, como a remoção de botões desnecessários nas páginas de *login* e cadastro. Informações não essenciais ao processo de cadastro, como data de nascimento e número de telefone, foram retiradas dos formulários e suas respectivas colunas removidas do banco de dados. Também foram adicionados recursos de autocomplete nos campos de *e-mail* presentes nas páginas de cadastro, *login* e redefinição de senha, ativados automaticamente quando o usuário digita o caractere “@”.

4.1.6 – Sprint Semana 6 – Correções de Erros e Melhorias II

Na sexta semana, as correções iniciadas anteriormente foram continuadas, com foco especial na parte visual do sistema. Foram adicionados elementos gráficos, como o logotipo na página “Sobre” e uma imagem padrão na tela de *login*. Também foram incorporados limitadores de *cards* na seção “ordens recentes” presente nos painéis do professor e do suporte. A saudação “Bem-vindo professor” foi revisada para “Bem-vindo, docente”, buscando maior precisão terminológica. A seguir na figura 6, encontra-se uma captura que demonstra como ficou o painel inicial da sessão do professor:

Figura 6 - Painel Professor



Fonte: Elaborado pelos autores, 2025

Paralelamente às melhorias de estilo, tentou-se corrigir uma falha na lógica do *back-end*: o sistema estava recebendo apenas ordens do tipo “relatar problemas”, enquanto ordens do tipo “solicitar instalação” não eram enviadas pelo *front-end*. Após diversas tentativas de solucionar o problema sem modificar a estrutura existente, constatou-se que a causa estava na construção do formulário no *front-end*. Assim, concluiu-se que seria necessário desmembrar o formulário em dois componentes independentes para que ambos os tipos de ordens fossem corretamente enviados ao *back-end* e salvos no banco de dados.

4.1.7 – Sprint Semana 7 – Meu Perfil

Durante a sétima semana, resolvemos o problema do front-end refatorando a estrutura dos formulários. Abandonamos o uso de um formulário padrão centralizado e criamos instâncias de formulário específicas para cada modal. Essa mudança na camada de apresentação trouxe maior clareza e modularidade, otimizando a manutenção. Além disso, foi desenvolvida integralmente a lógica do *back-end* referente à área "Meu Perfil", permitindo que os usuários alterem suas senhas de forma segura (com validação de senha atual e criptografia da nova senha) e façam upload de uma foto de perfil. A foto é armazenada no servidor com nome único gerado por UUID e o caminho é salvo no banco de dados, sendo exibida em todas as áreas do sistema onde o usuário está logado.

4.1.8 – Sprint Semana 8 – Admin

No Sprint 8, concentrou-se o desenvolvimento da lógica do *back-end* e da estrutura do banco de dados correspondente às páginas da sessão do administrador, assegurando a integração completa entre as camadas do sistema e implementando funcionalidades avançadas de gestão. A página relacionada ao perfil segue a mesma lógica e estrutura da versão destinada ao suporte, mantendo consistência na experiência do usuário. A página inicial da seção, denominada painel-admin, funciona como um *dashboard* estratégico de navegação, exibindo métricas consolidadas do sistema.

A funcionalidade de Gerenciamento de Patrimônio é sustentada pela tabela patrimonio, que atua como a fonte única e central de dados para o inventário de ativos. Os atributos de tal tabela são: o *id_do_patrimônio*, a descrição detalhada, o local de alocação e o *status* operacional, que define o estado do ativo no momento.

A página de gerenciamento de usuários, por sua vez, permite que administradores adicionem manualmente usuários individuais, editem informações de usuários existentes ou removam acessos quando necessário, realizando operações *CRUD* diretamente sobre as tabelas matriculas_autorizadas e users, como pode ser visto pela figura 7:

Figura 7 - Gerenciar usuários

Fonte: Elaborado pelos autores, 2025.

Por fim, foi desenvolvida a página de importação de matrículas em lote, responsável por receber o arquivo *CSV* enviado pelo administrador (com formato padronizado definido), processar as informações linha a linha, e salvar as matrículas autorizadas na tabela *matriculas_autorizadas*. O sistema realiza verificação e fornece *feedback* detalhado sobre o processo de importação, indicando quantos registros foram importados com sucesso.

4.1.9 – Sprint Semana 9 – Toques Finais

Na última semana, foram realizados os ajustes finais no sistema, preparando-o para implantação. Uma nova revisão abrangente permitiu identificar pontos que ainda necessitavam de aprimoramento, especialmente na lógica de ordens e na experiência do usuário final. A finalização de uma ordem passou a exigir obrigatoriamente que o técnico registre uma breve descrição do procedimento realizado antes de encerrá-la, atendendo aos requisitos funcionais.

Além disso, foi desenvolvida integralmente a página de relatórios do administrador, responsável por coletar e agrregar os dados armazenados no banco de ordens, aplicar filtros customizáveis (por período, status, técnico, tipo de solicitação) e gerar relatórios detalhados em formato PDF. Os relatórios incluem tabelas e indicadores de performance, podendo ser impressos ou salvos digitalmente pelo administrador para análise gerencial e tomada de decisões estratégicas sobre a operação do suporte técnico.

5. CONCLUSÃO

O desenvolvimento do *Support Nexus* para a Fatec Carapicuíba trouxe à tona importantes desafios relacionados à gestão de ordens de serviço no contexto acadêmico, e o sistema proposto se mostrou uma solução eficaz para a otimização do processo de suporte de TI. A implementação do sistema possibilitará uma comunicação mais eficiente entre os docentes e a equipe de TI, contribuindo significativamente para a redução do tempo de inatividade dos equipamentos e para a melhoria na qualidade das aulas.

O projeto foi estruturado com o uso de metodologias ágeis, o que permitiu uma evolução contínua do sistema durante o desenvolvimento, com *feedback* constante e ajustes conforme as necessidades da instituição. O uso de *front-end* e *back-end* bem integrados assegurou uma interface intuitiva e interativa para os usuários, ao mesmo tempo em que garantiu o processamento seguro e eficaz das ordens de serviço no servidor. As tecnologias escolhidas, como Node.js para o *back-end* e PostgreSQL para o banco de dados, se mostraram adequadas para garantir a escalabilidade e a confiabilidade do sistema.

A conclusão deste trabalho indica que um sistema como o *Support Nexus* tem potencial para melhorar a gestão de suporte de TI em instituições acadêmicas, além de reforçar a relevância da integração entre front-end e back-end no desenvolvimento de soluções tecnológicas mais completas. Embora este projeto apresente apenas um protótipo, os resultados obtidos permitem sugerir que a solução pode contribuir para a eficiência operacional, reduzindo o tempo de resposta às demandas de manutenção e tendendo a aumentar a satisfação dos usuários, incluindo professores e técnicos.

Com base no que foi desenvolvido até aqui, é possível inferir que o *Support Nexus* possui grande potencial de expansão e adaptação para outras instituições de ensino, oferecendo um modelo promissor para o gerenciamento de ordens de serviço em ambientes acadêmicos.

Os testes conceituais e a arquitetura implementada indicam que a solução pode vir a otimizar processos de suporte de TI, especialmente no que diz respeito à comunicação entre o corpo docente e a equipe técnica. A integração entre as camadas de front-end e back-end resultou em uma plataforma funcional dentro do escopo do protótipo, demonstrando que o sistema pode evoluir para atender necessidades reais e permitir futuras melhorias e inovações.

A partir desta implementação inicial, diversas oportunidades de expansão surgem, como a adição de notificações em tempo real, autenticação segura, automação de processos e até mesmo inteligência artificial para análise preditiva de problemas. Tecnologias como Socket.IO, Elasticsearch, OAuth e machine learning se apresentam como caminhos viáveis para aprimorar ainda mais a experiência do usuário e o desempenho do sistema.

Este projeto representa apenas o início de uma possível evolução rumo a uma plataforma mais robusta, escalável e inovadora. Futuros desenvolvedores são convidados a dar continuidade ao trabalho, explorar novas tecnologias e contribuir para o fortalecimento de uma solução que pode vir a beneficiar diversas instituições. Com o avanço dessas melhorias, o Support Nexus tem o potencial de se tornar uma referência na gestão de TI acadêmica, promovendo eficiência e inovação no suporte aos usuários.

REFERÊNCIAS

CHIN, Gary. **Agile project management**. New York: AMACOM, 2004. p. 4-5. Disponível em: <http://www.maxwideman.com/papers/agile/agile.pdf>. Acesso em: 10 abr. 2025.

CHOO, Chun Wei. A Teoria Organizacional da Informação. In: WILSON, T. D. (Ed.). **Annual Review of Information Science and Technology (ARIST)**. [S.l.]: Information Today, 2006. v. 40, n. 1, p. 219–257.

COHN, Mike. **User stories applied**: for agile software development. Boston: Addison-Wesley Professional, 2004.

ENGKAMAT, Adeline; YII, Ming Leong; SINATRA GRAN, Shirley. **Replit**: a simple approach to real-time collaborative coding. [S.l.: s.n.], 2023.

FLANAGAN, David. **JavaScript**: o guia definitivo. 6. ed. Porto Alegre: Bookman, 2012.

LAUDON, Kenneth C.; LAUDON, Jane P. **Management Information Systems**: managing the digital firm. 16. ed. New York: Pearson, 2020.

LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de informação gerenciais**: administrando a empresa digital. 11. ed. Porto Alegre: Bookman, 2022.

MCFARLAND, David Sawyer. **CSS3**: the missing manual. Sebastopol: O'Reilly Media, 2012.

MCGEE, James V.; PRUSAK, Laurence. **Gerenciamento estratégico da informação**: aumente a competitividade e a eficiência de sua empresa utilizando a informação como uma ferramenta estratégica. Rio de Janeiro: Elsevier, 1994.

MILANI, André. **MySQL**: guia do programador. São Paulo: Novatec, 2007.

O'BRIEN, James A. **Introduction to Information Systems**. 16. ed. New York: McGraw-Hill Education, 2011.

PEREIRA, Caio Ribeiro. **Aplicações web real-time com Node.js**. São Paulo: Casa do Código, 2014.

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

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do trabalho científico**: métodos e técnicas da pesquisa e do trabalho acadêmico. 2. ed. Novo Hamburgo: Feevale, 2013.

PROTÓTIPO. In: **MICHAELIS on-line**. São Paulo: Editora Melhoramentos, [202-?]. Disponível em: <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/prot%C3%B3tipo/>. Acesso em: 11 abr. 2025.

RAMESH, B.; JARKE, M. Towards reference models for requirements traceability. **IEEE Transactions on Software Engineering**, New York, v. 27, n. 1, p. 58-93, jan. 2001.

SCHWABER, Ken; SUTHERLAND, Jeff. **O Guia do Scrum**: o guia definitivo para o Scrum: as regras do jogo. [S.l.: s.n.], 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf>. Acesso em: 10 abr. 2025.

SILVA, Maurício Samy. **Criando sites com HTML**: sites de alta qualidade com HTML e CSS. São Paulo: Novatec, 2008.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

STAIR, Ralph M.; REYNOLDS, George W. **Principles of information systems**. 10. ed. Boston: Cengage Learning, 2011.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira. **Engenharia de Requisitos**: software orientado ao negócio. Rio de Janeiro: Brasport, 2016.

WEILL, Peter; ROSS, Jeanne W. **IT governance**: how top performers manage IT decision rights for superior results. Boston: Harvard Business Press, 2004.

WORSLEY, John; DRAKE, Joshua D. **Practical PostgreSQL**. Sebastopol: O'Reilly Media, 2002.