



UNIVERSIDADE FEDERAL DO AMAZONAS  
INSTITUTO DE COMPUTAÇÃO  
CURSO DE ENGENHARIA DE SOFTWARE



---

NELSON PEREIRA DE CARVALHO NETO

### **Sistema de Apoio Acadêmico: IcompCare**

Uma Abordagem Segura e Integrada para Gestão de Atendimentos no ICOMP/UFAM

Manaus-AM

2025

NELSON PEREIRA DE CARVALHO NETO

**Sistema de Apoio Acadêmico: IcompCare**

Uma Abordagem Segura e Integrada para Gestão de Atendimentos no ICOMP/UFAM

**Versão Original**

Trabalho de Conclusão de Curso apresentado ao Instituto de Computação da Universidade Federal do Amazonas como parte dos requisitos necessários para obtenção do grau de Bacharel em Engenharia de Software.

Orientador: Profa. Dra. Ana Carolina Oran

Manaus-AM

2025

## **Sistema de Apoio Acadêmico: IcompCare**

NELSON PEREIRA DE CARVALHO NETO

Trabalho de Conclusão de Curso (TCC) apresentado ao Instituto de Computação da Universidade Federal do Amazonas como parte dos requisitos necessários para obtenção do grau de Bacharel em Engenharia de Software.

Manaus-AM, 15 de janeiro de 2025.

## **RESUMO**

CARVALHO NETO, Nelson Pereira de. **Sistema de Apoio Acadêmico: IcompCare: Uma Abordagem Segura e Integrada para Gestão de Atendimentos no ICOMP/UFAM. 2025.** 50f. Trabalho de Conclusão de Curso (Engenharia de Software) - Universidade Federal do Amazonas, Manaus-AM, 2025.

A crescente demanda por suporte à saúde mental no ensino superior impulsiona a busca por ferramentas tecnológicas que otimizem a gestão dos atendimentos. Visando aprimorar os serviços psicopedagógicos oferecidos na Universidade Federal do Amazonas (UFAM), este trabalho apresenta o desenvolvimento do IcompCare, um Sistema de Apoio Acadêmico. A plataforma web foi projetada para agregar eficiência, segurança e organização ao processo, servindo como uma evolução aos fluxos de trabalho tradicionais. A solução introduz um portal de agendamento online, onde alunos podem visualizar a disponibilidade da equipe e solicitar horários. Para os profissionais, o sistema oferece um painel de gestão de consultas e um prontuário eletrônico com dados criptografados. O sistema permite ainda o compartilhamento seguro de observações não-clínicas com professores, promovendo um acompanhamento integrado. O projeto utilizou .NET 8, Vue.js e PostgreSQL.

**Palavras-chave:** apoio psicológico, sistema web, gestão acadêmica, privacidade de dados, engenharia de software.

## **ABSTRACT**

CARVALHO NETO, Nelson Pereira de. **Academic Support System: IcompCare: A Secure and Integrated Approach for Service Management at ICOMP/UFAM. 2025. 50f.** Undergraduate thesis (Computer Science) - Federal University of Amazonas, Manaus-AM, 2025.

The growing demand for mental health support in higher education drives the search for technological tools that optimize service management. Aiming to enhance the psychopedagogical services offered at the Federal University of Amazonas (UFAM), this work presents the development of IcompCare, an Academic Support System. The web platform is designed to add efficiency, security, and organization to the process. The solution introduces an online scheduling portal where students can view staff availability. For professionals, the system offers a management dashboard and an electronic record module with encrypted data. The system also allows for the secure sharing of specific observations with professors, promoting integrated student follow-up. The project was developed using .NET 8, Vue.js, and PostgreSQL.

**Keywords:** psychological support, web system, academic management, data privacy, software engineering.

## **LISTA DE SIGLAS**

**API** Application Programming Interface

**CSS** Cascading Style Sheets

**DTO** Data Transfer Object

**HTML** HyperText Markup Language

**ICOMP** Instituto de Computação

**JWT** JSON Web Token

**LGPD** Lei Geral de Proteção de Dados

**MER** Modelo Entidade-Relacionamento

**ORM** Object-Relational Mapping

**REST** Representational State Transfer

**RF** Requisitos Funcionais

**RNF** Requisitos Não Funcionais

**SGBD** Sistema Gerenciador de Banco de Dados

**SPA** Single Page Application

**SQL** Structured Query Language

**TCC** Trabalho de Conclusão de Curso

**UFAM** Universidade Federal do Amazonas

## LISTA DE ILUSTRAÇÕES

Figura 1	Esquema do Banco de Dados . . . . .	27
Figura 2	Tela de Login do Sistema . . . . .	35
Figura 3	Armazenamento de Senhas com Hash Criptográfico . . . . .	35
Figura 4	Formulário de Agendamento Público . . . . .	36
Figura 5	Resumo do Agendamento . . . . .	36
Figura 6	Confirmação da Solicitação de Agendamento . . . . .	37
Figura 7	Email de Confirmação de Agendamento . . . . .	37
Figura 8	Painel de Gestão de Agendamentos . . . . .	38
Figura 9	Detalhes de um Agendamento . . . . .	39
Figura 10	Vincular Estudante ao Agendamento . . . . .	39
Figura 11	Confirmação de Vinculação do Estudante ao Agendamento . . . . .	40
Figura 12	Seleção do Tipo de Atendimento . . . . .	40
Figura 13	Email de Confirmação de Agendamento . . . . .	41
Figura 14	Email de Cancelamento de Agendamento . . . . .	41
Figura 15	Tela de Sessões do Profissional . . . . .	42
Figura 16	Prontuário Eletrônico com Editor de Texto Rico . . . . .	43
Figura 17	Encerramento da Sessão . . . . .	43
Figura 18	Histórico de Sessões do Estudante . . . . .	44
Figura 19	Conclusão do Agendamento . . . . .	44
Figura 20	Dados do Prontuário Eletrônico Criptografados . . . . .	44
Figura 21	Painel do Professor - Lista de Alunos por Disciplina Ofertada . . . . .	45
Figura 22	Detalhes do Aluno e Observações . . . . .	46

## **LISTA DE TABELAS**

Tabela 1	Requisitos Funcionais do Sistema . . . . .	19
Tabela 2	Requisitos Não Funcionais do Sistema . . . . .	21

# SUMÁRIO

<b>Lista de Siglas</b>	<b>v</b>
<b>1 Introdução</b>	<b>11</b>
1.1 Objetivos . . . . .	11
1.1.1 Objetivo Geral . . . . .	11
1.1.2 Objetivos Específicos . . . . .	11
1.2 Metodologia . . . . .	11
1.3 Organização do Trabalho . . . . .	12
<b>2 Referencial Teórico</b>	<b>13</b>
2.1 Saúde Mental no Ensino Superior . . . . .	13
2.2 Arquitetura de Software . . . . .	13
2.2.1 Clean Architecture . . . . .	14
2.2.2 Padrão API REST . . . . .	14
2.3 Desenvolvimento Web . . . . .	14
2.3.1 Single Page Applications (SPA) . . . . .	14
2.4 Segurança e Proteção de Dados . . . . .	15
2.4.1 Criptografia e Hash . . . . .	15
<b>3 Trabalhos Relacionados</b>	<b>16</b>
3.1 Sistema para Gerenciamento e Agendamento de Consultas para Psicólogos e Clientes . . . . .	16
3.2 SystemPsi: Sistema Gerenciador para Psicólogos em Atuação Remota . . . . .	17
3.3 Sistema para Agendamento de Serviços . . . . .	17

<b>4 Materiais e Métodos</b>	<b>18</b>
4.1 Levantamento de Requisitos . . . . .	18
4.1.1 Requisitos Funcionais (RF) . . . . .	18
4.1.2 Requisitos Não Funcionais (RNF) . . . . .	21
4.2 Tecnologias utilizadas . . . . .	22
4.2.1 Backend . . . . .	22
4.2.2 Frontend . . . . .	23
4.2.3 Banco de Dados . . . . .	24
4.2.4 Ferramentas de Desenvolvimento . . . . .	24
4.3 Banco de Dados . . . . .	25
4.3.1 Modelo Entidade-Relacionamento . . . . .	25
4.3.2 Estratégias de Segurança e Integridade . . . . .	28
4.4 User Stories . . . . .	29
4.4.1 User Stories Principais . . . . .	29
<b>5 Resultados e Discussões</b>	<b>34</b>
5.1 Apresentação do Sistema . . . . .	34
5.1.1 Autenticação e Controle de Acesso . . . . .	34
5.1.2 Fluxo de Agendamento Público . . . . .	35
5.1.3 Gestão de Agendamentos pelo Profissional . . . . .	38
5.1.4 Realização de Sessões e Prontuário Eletrônico . . . . .	42
5.1.5 Área do Professor . . . . .	45
5.2 Discussão dos Resultados . . . . .	46
5.2.1 Eficiência Operacional . . . . .	46
5.2.2 Segurança da Informação . . . . .	47
5.2.3 Integração e Acompanhamento . . . . .	47

<b>6 Conclusões</b>	<b>48</b>
6.1 Contribuições . . . . .	48
6.2 Trabalhos Futuros . . . . .	49
<b>Referências</b>	<b>50</b>

# 1 INTRODUÇÃO

A gestão de serviços de apoio psicopedagógico em universidades como a UFAM demanda ferramentas eficientes. Processos manuais para agendamento, registro e comunicação são frequentemente lentos e apresentam riscos à privacidade dos dados dos estudantes. A tecnologia web oferece uma solução direta para modernizar e proteger esses fluxos de trabalho.

Este trabalho apresenta o desenvolvimento do IcompCare, uma plataforma web criada para centralizar e otimizar a gestão dos atendimentos no Instituto de Computação (ICOMP).

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

Desenvolver um sistema web seguro e eficiente para automatizar o gerenciamento do serviço de apoio psicopedagógico do ICOMP/UFAM.

### 1.1.2 Objetivos Específicos

- Levantar os requisitos funcionais e não funcionais do sistema.
- Projetar o banco de dados para o armazenamento seguro das informações.
- Implementar módulos de agendamento, prontuário eletrônico e perfis de acesso.
- Assegurar segurança através de criptografia.

## 1.2 Metodologia

A metodologia adotada foi baseada em um modelo de desenvolvimento iterativo e incremental, permitindo a construção do software em ciclos e validações contínuas com a stakeholder.

der (psicóloga do ICOMP). Também foram adotadas práticas de engenharia de software como análise, projeto, desenvolvimento e teste.

### **1.3 Organização do Trabalho**

O trabalho está organizado em seis capítulos. O Capítulo 2 apresenta o Referencial Teórico. O Capítulo 3 discute os Trabalhos Relacionados. O Capítulo 4 detalha Materiais e Métodos. O Capítulo 5 apresenta os Resultados e o Capítulo 6 as Considerações Finais.

## 2 REFERENCIAL TEÓRICO

Este capítulo apresenta a fundamentação teórica necessária para o desenvolvimento do Sistema de Apoio Acadêmico. São abordados os conceitos relacionados à gestão da saúde mental no ambiente universitário, seguidos pelos princípios de Engenharia de Software utilizados, incluindo arquitetura de sistemas, desenvolvimento web moderno e segurança da informação com foco na legislação vigente.

### 2.1 Saúde Mental no Ensino Superior

A saúde mental no contexto universitário tem se tornado um tema de crescente relevância. O ambiente acadêmico, caracterizado por prazos rigorosos, alta carga cognitiva e pressão por desempenho, pode atuar como um estressor significativo. Conforme apontado por Leão *et al.* (2018), a incidência de ansiedade e depressão entre universitários tende a ser superior à da população geral, o que demanda das instituições de ensino a criação de mecanismos de suporte eficazes.

A informatização desses serviços de apoio não é apenas uma questão de modernização, mas de acessibilidade e eficácia. Sistemas de gestão clínica permitem um acompanhamento mais próximo, reduzem o estigma associado à busca presencial por informações e garantem a continuidade do tratamento através de um histórico organizado e seguro.

### 2.2 Arquitetura de Software

A arquitetura de software refere-se à estrutura fundamental de um sistema, compreendendo seus componentes, as relações entre eles e os princípios que guiam seu design e evolução. Uma arquitetura bem definida é crucial para a manutenibilidade e escalabilidade do software.

## 2.2.1 Clean Architecture

A Clean Architecture (Arquitetura Limpa), proposta por Martin (2017), é um padrão de design de software que visa a separação de preocupações. Seu principal objetivo é tornar o sistema independente de frameworks, banco de dados e interfaces de usuário.

O modelo organiza o software em camadas concêntricas, onde a regra de dependência estabelece que as camadas internas (Domínio e Aplicação) não devem conhecer nada sobre as camadas externas (Interface e Infraestrutura). Isso permite que as regras de negócio permaneçam puras e testáveis, enquanto detalhes técnicos podem ser substituídos com mínimo impacto no núcleo do sistema. Essa abordagem facilita a manutenção a longo prazo e a adaptação a novas tecnologias.

## 2.2.2 Padrão API REST

O estilo arquitetural REST (Representational State Transfer) define um conjunto de restrições para a criação de web services. Em uma API REST, a comunicação entre cliente e servidor ocorre através de requisições HTTP padronizadas, tratando os dados como recursos que podem ser criados, lidos, atualizados ou excluídos (operações CRUD).

A adoção de APIs REST permite o desacoplamento total entre o frontend e o backend. O servidor processa as regras de negócio e retorna dados (geralmente em formato JSON), enquanto o cliente (seja web ou mobile) consome esses dados e os apresenta ao usuário.

## 2.3 Desenvolvimento Web

O desenvolvimento de aplicações web evoluiu do modelo tradicional de renderização no servidor para abordagens focadas na experiência do usuário e na interatividade.

### 2.3.1 Single Page Applications (SPA)

Uma Single Page Application (SPA) é uma aplicação web que carrega uma única página HTML e atualiza dinamicamente o conteúdo conforme o usuário interage com o sistema, sem a necessidade de recarregar a página inteira a cada ação.

Essa abordagem, viabilizada por frameworks JavaScript modernos como Vue.js, React e Angular, proporciona uma experiência de usuário mais fluida e responsiva, semelhante à de

aplicativos desktop nativos. O processamento da interface é transferido para o navegador do cliente, reduzindo a carga no servidor e o tráfego de rede.

## 2.4 Segurança e Proteção de Dados

Com a digitalização de registros médicos e acadêmicos, a segurança da informação torna-se um requisito não funcional crítico, envolvendo a confidencialidade, integridade e disponibilidade dos dados.

### 2.4.1 Criptografia e Hash

A criptografia é a técnica fundamental para garantir a confidencialidade dos dados. Ela consiste em codificar a informação de modo que apenas partes autorizadas, detentoras de uma chave específica, possam decifrá-la.

Para o armazenamento de senhas, utilizam-se funções de hash criptográfico. Diferente da criptografia simétrica ou assimétrica, o hash é uma via de mão única: ele transforma a senha em uma cadeia de caracteres de tamanho fixo impossível de ser revertida para o texto original. Isso garante que, mesmo em caso de vazamento do banco de dados, as senhas dos usuários não sejam expostas.

## 3 TRABALHOS RELACIONADOS

Neste capítulo, são apresentados trabalhos acadêmicos e projetos de software que abordam temas correlatos ao deste TCC, como sistemas de agendamento online, gerenciamento de consultas pedagógicas e plataformas de apoio ao estudante. A análise destes trabalhos permite identificar funcionalidades consolidadas, tecnologias empregadas e, fundamentalmente, as lacunas que justificam o desenvolvimento da presente proposta.

### 3.1 Sistema para Gerenciamento e Agendamento de Consultas para Psicólogos e Clientes

Souza (2021) propôs uma plataforma web para a gestão de agendamentos de consultas pedagógicas. O sistema inclui funcionalidades essenciais como agendamento, registro de anamneses e geração de relatórios, operando com perfis de usuário distintos para o profissional e para o cliente. As tecnologias utilizadas foram PHP, JavaScript, HTML, CSS e MySQL.

A principal relevância deste trabalho reside na sua abordagem direta ao agendamento online no contexto da pedagogia, validando a arquitetura de perfis de usuário (profissional e paciente) que é similar à base do nosso sistema (profissional e aluno). No entanto, sua proposta é focada na relação dual cliente-profissional. O diferencial do nosso projeto se manifesta na introdução de um terceiro perfil, o de professor, e na criação de um ecossistema integrado ao ambiente acadêmico, com um canal de comunicação específico e controlado, funcionalidade não abordada por este trabalho.

## 3.2 SystemPsi: Sistema Gerenciador para Psicólogos em Atuação Remota

O SystemPsi é uma ferramenta tecnológica apresentada por Stefen *et al.* (2022), desenvolvida no Instituto Federal Catarinense (IFC). O sistema foi projetado para auxiliar profissionais em atendimentos remotos, e seu desenvolvimento contou com a validação de um profissional da área para garantir a pertinência das funcionalidades. A pilha de tecnologias inclui PHP e MySQL.

Este trabalho se destaca pela sua metodologia, que enfatiza a importância da validação das funcionalidades junto ao usuário final — o profissional de pedagogia. Essa abordagem reforça a metodologia adotada em nosso projeto, que também se baseia na colaboração direta com a profissional do ICOMP. O SystemPsi, contudo, concentra-se nas necessidades do profissional em um contexto de atuação remota geral. Nossa proposta avança ao especializar a ferramenta para o nicho universitário, atendendo não apenas às necessidades do profissional, mas também às dinâmicas de interação com alunos e professores dentro de uma instituição de ensino (STEFEN *et al.*, 2022).

## 3.3 Sistema para Agendamento de Serviços

Kieras (2019) desenvolveu um sistema genérico para o agendamento de serviços, composto por uma aplicação web e um aplicativo móvel.

A principal contribuição deste trabalho para o nosso projeto não está no domínio da aplicação, mas sim na sua sólida documentação de engenharia de software. O detalhamento da análise de requisitos, dos casos de uso e do planejamento das funcionalidades serve como uma excelente referência metodológica para a estruturação do presente TCC. Enquanto o sistema da UTFPR foi projetado para ser genérico e aplicável a diversos contextos de agendamento, nosso sistema se aprofunda em um domínio específico — o apoio psicopedagógico universitário. Essa especialização permite a criação de funcionalidades sob medida, como o prontuário eletrônico e a visualização restrita para professores, que não fariam parte de uma solução genérica.

## 4 MATERIAIS E MÉTODOS

Este capítulo detalha os materiais, as ferramentas e os procedimentos metodológicos empregados no desenvolvimento do Sistema de Apoio Acadêmico. Serão abordados o processo de levantamento de requisitos, as tecnologias selecionadas para a implementação e a modelagem do banco de dados que estrutura a aplicação.

### 4.1 Levantamento de Requisitos

A etapa de levantamento de requisitos foi fundamental para definir o escopo e as funcionalidades do sistema. O processo foi conduzido através de reuniões com a profissional do ICOMP, a principal interessada (stakeholder) no projeto, e com a orientação da professora Dra. Ana Carolina Oran. Nessas reuniões, foi possível mapear o fluxo de trabalho existente, identificar as limitações do método atual e especificar as necessidades dos futuros usuários (alunos, professores e a própria profissional).

As funcionalidades desejadas foram então estruturadas e classificadas em duas categorias principais, conforme a prática da Engenharia de Software: Requisitos Funcionais e Requisitos Não Funcionais.

#### 4.1.1 Requisitos Funcionais (RF)

Os Requisitos Funcionais descrevem as ações e funcionalidades que o sistema deve ser capaz de executar. Eles definem o comportamento do software sob a perspectiva do usuário. A Tabela 1 apresenta a lista de requisitos funcionais identificados para o projeto.

Tabela 1: Requisitos Funcionais do Sistema

<b>ID</b>	<b>Categoria</b>	<b>Descrição</b>
RF001	Autenticação	O sistema deve permitir login de usuários com email institucional e senha.
RF002	Agendamento Público	Qualquer usuário deve poder acessar a página pública de agendamento de atendimentos.
RF003	Agendamento Público	O sistema deve exibir a disponibilidade dos profissionais em formato de calendário.
RF004	Agendamento Público	O sistema deve permitir a solicitação de agendamento informando dados do estudante e motivo da visita.
RF005	Notificações	O sistema deve enviar email de confirmação quando uma solicitação for recebida.
RF006	Notificações	O sistema deve enviar email quando um agendamento for confirmado pelo profissional.
RF007	Notificações	O sistema deve enviar email quando um agendamento for cancelado.
RF008	Notificações	O sistema deve enviar email quando um agendamento for remarcado.
RF009	Gerenciamento (Profissional)	O profissional deve poder visualizar e gerenciar solicitações de agendamento.
RF010	Gerenciamento (Profissional)	O profissional deve poder confirmar, cancelar ou remarcar agendamentos.
RF011	Gerenciamento (Profissional)	O profissional deve poder configurar sua disponibilidade de horários.
RF012	Gerenciamento (Profissional)	O profissional deve poder iniciar uma sessão a partir de um agendamento confirmado.
RF013	Sessões	O profissional deve poder registrar anotações da sessão em prontuário eletrônico criptografado.
RF014	Sessões	O profissional deve poder finalizar sessões e marcar o agendamento como concluído.
RF015	Sessões	O sistema deve registrar automaticamente data/hora de início e término das sessões.
RF016	Tipos de Sessão	O sistema deve permitir cadastro de tipos de sessão com nome, duração e descrição.

Tabela 1: Requisitos Funcionais do Sistema (continuação)

<b>ID</b>	<b>Categoria</b>	<b>Descrição</b>
RF017	Observações	O profissional deve poder cadastrar observações sobre estudantes.
RF018	Observações	O profissional deve poder categorizar observações por tipo.
RF019	Área do Professor	Professores devem poder visualizar disciplinas que lecionam organizadas por semestre.
RF020	Área do Professor	Professores devem poder visualizar alunos matriculados em suas disciplinas.
RF021	Área do Professor	Professores devem poder visualizar observações dos profissionais sobre seus alunos.
RF022	Administração	O administrador deve poder gerenciar usuários do sistema (criar, editar, ativar/desativar).
RF023	Administração	O administrador /ou profissional deve poder gerenciar semestres letivos.
RF024	Administração	O administrador /ou profissional deve poder gerenciar disciplinas.
RF025	Administração	O administrador /ou profissional deve poder vincular professores a disciplinas em semestres específicos.
RF026	Administração	O administrador /ou profissional deve poder matricular alunos em turmas.
RF027	Administração	O administrador /ou profissional deve poder gerenciar tipos de observações disponíveis no sistema.
RF028	Controle de Acesso	O sistema deve implementar perfis de usuário: Admin, Profissional, Estudante, Assistente e Professor.
RF029	Controle de Acesso	O sistema deve restringir acesso a funcionalidades baseado no perfil do usuário.

*Fonte: O Autor (2025)*

#### 4.1.2 Requisitos Não Funcionais (RNF)

Os Requisitos Não Funcionais especificam os critérios de qualidade e as restrições operacionais do sistema. Eles não se referem a uma funcionalidade específica, mas sim a como o sistema deve operar em termos de desempenho, segurança, usabilidade, entre outros. A Tabela 2 detalha os requisitos não funcionais do projeto.

Tabela 2: Requisitos Não Funcionais do Sistema

ID	Categoria	Descrição
RNF001	Tecnologia	O sistema deve ser uma aplicação web acessível através de navegadores modernos (Chrome, Firefox, Safari, Edge).
RNF002	Arquitetura	O backend deve seguir arquitetura limpa (Clean Architecture) com separação em camadas (Domain, Application, Infrastructure).
RNF003	Tecnologia	O sistema deve utilizar .NET 8 para o backend e Vue.js 3 para o frontend.
RNF004	Banco de Dados	O sistema deve utilizar PostgreSQL como sistema gerenciador de banco de dados.
RNF005	Usabilidade	A interface da tela de solicitação deve ser responsiva e adaptável a dispositivos móveis, tablets e desktops.
RNF006	Segurança	O acesso ao sistema deve ser controlado por autenticação JWT (JSON Web Token).
RNF007	Segurança	As senhas dos usuários devem ser armazenadas utilizando hash.
RNF008	Segurança	Os prontuários eletrônicos devem ser armazenados de forma criptografada no banco de dados.
RNF009	Segurança	O acesso aos dados de pacientes deve ser restrito apenas aos profissionais autorizados.
RNF010	Segurança	Professores devem visualizar apenas observações de estudantes matriculados em suas disciplinas.
RNF011	Internacionalização	O sistema deve oferecer suporte a múltiplos idiomas (Português, Inglês, Espanhol).
RNF012	Confiabilidade	O sistema de envio de emails deve utilizar fila assíncrona para garantir entrega.

Tabela 2: Requisitos Não Funcionais do Sistema (continuação)

ID	Categoría	Descrição
RNF013	Confiabilidade	O sistema de emails deve implementar retry automático em caso de falhas temporárias.
RNF014	Usabilidade	Todas as notificações por email devem ser enviadas em português brasileiro.
RNF015	Usabilidade	As datas e horários devem ser exibidas no fuso horário de Manaus (UTC-4).
RNF016	Desempenho	O sistema deve utilizar índices de banco de dados para otimizar consultas frequentes.
RNF017	Manutenibilidade	O código deve seguir padrões de nomenclatura e boas práticas de desenvolvimento.
RNF018	Manutenibilidade	Tipos de sessão e observações devem ser gerenciáveis sem alteração de código.
RNF019	Acessibilidade	A interface deve seguir princípios básicos de acessibilidade (contraste, navegação por teclado).
RNF020	Portabilidade	O sistema deve ser containerizável usando Docker para facilitar implantação.

Fonte: O Autor (2025)

## 4.2 Tecnologias utilizadas

A escolha das tecnologias para o desenvolvimento do Sistema de Apoio Acadêmico foi guiada por critérios de modernidade, robustez, escalabilidade e conformidade com as melhores práticas de engenharia de software. O sistema foi construído seguindo uma arquitetura cliente-servidor, com clara separação entre o frontend (interface do usuário) e o backend (lógica de negócio e acesso a dados).

### 4.2.1 Backend

O backend do sistema foi desenvolvido utilizando a plataforma .NET 8 (MICROSOFT, 2025), a versão mais recente (Long-Term Support) do framework da Microsoft para desenvolvimento de aplicações multiplataforma. A escolha do .NET se justifica por sua performance, segurança, ampla documentação e suporte da comunidade.

A estrutura do backend foi organizada seguindo os princípios da **Clean Architecture** (Arquitetura Limpa), proposta por Martin (2017). Esta abordagem promove a separação de responsabilidades através de camadas bem definidas:

- **IcompCare.Domain**: Camada de domínio que contém as entidades do negócio, enums e interfaces de repositórios. Esta camada é independente de frameworks externos e representa o núcleo da aplicação.
- **IcompCare.Application**: Camada de aplicação que implementa os casos de uso do sistema através de serviços. Contém as DTOs (Data Transfer Objects) e as interfaces dos serviços.
- **IcompCare.Infrastructure**: Camada de infraestrutura que implementa os detalhes técnicos, como acesso ao banco de dados através do Entity Framework Core, serviços de email e criptografia.
- **IcompCare.Api**: Camada de apresentação que expõe os endpoints REST através de controllers, implementa middlewares de autenticação e tratamento de erros.

As principais bibliotecas e pacotes utilizados no backend incluem:

- **Entity Framework Core 8.0.11**: ORM (Object-Relational Mapper) para mapeamento objeto-relacional e acesso ao banco de dados.
- **Npgsql.EntityFrameworkCore.PostgreSQL 8.0.11**: Provider do Entity Framework para PostgreSQL.
- **Microsoft.AspNetCore.Authentication.JwtBearer 8.0.2**: Implementação de autenticação via tokens JWT.
- **EFCore.NamingConventions 8.0.3**: Conversão automática de nomenclatura para snake\_case no banco de dados.
- **System.Net.Mail**: Biblioteca nativa do .NET para envio de emails via SMTP.

#### 4.2.2 Frontend

O frontend foi desenvolvido como uma **Single Page Application (SPA)** utilizando **Vue.js 3.5.22** (VUE.JS, 2025), um framework JavaScript progressivo e reativo. O Vue.js foi escolhido

por sua curva de aprendizado suave, excelente performance e ecossistema maduro de ferramentas.

A aplicação frontend utiliza **TypeScript 5.9.3** como linguagem principal, trazendo tipagem estática ao JavaScript e melhorando a manutenibilidade e robustez do código. O gerenciamento de rotas é feito através do **Vue Router 4.6.3**, permitindo navegação client-side e controle de acesso baseado em permissões.

Para a estilização da interface, foi adotado o **Tailwind CSS 4.1.17**, um framework CSS utility-first que permite a construção rápida de interfaces responsivas e consistentes. O projeto utiliza o **Vite 7.1.11** como build tool e dev server, proporcionando hot module replacement (HMR) extremamente rápido durante o desenvolvimento.

As principais bibliotecas complementares do frontend incluem:

- **vue-i18n 11.2.1**: Sistema de internacionalização para suporte a múltiplos idiomas (português, inglês e espanhol).
- **vue-toastification 2.0.0-rc.5**: Biblioteca para exibição de notificações toast ao usuário.
- **lucide-vue-next 0.554.0**: Conjunto de ícones SVG otimizados e personalizáveis.
- **@vueup/vue-quill 1.0.0-beta.11**: Editor de texto rico (WYSIWYG) para criação de prontuários e observações.

### 4.2.3 Banco de Dados

O **PostgreSQL 17** (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2025) foi escolhido como sistema gerenciador de banco de dados relacional. O PostgreSQL é um SGBD open-source robusto, conhecido por sua confiabilidade, conformidade com padrões SQL e recursos avançados como suporte nativo a tipos JSON, índices complexos e criptografia.

A escolha do PostgreSQL também se alinha com o requisito de segurança e conformidade, uma vez que oferece recursos nativos de criptografia de dados em repouso e em trânsito, além de controle granular de permissões de acesso.

### 4.2.4 Ferramentas de Desenvolvimento

O desenvolvimento do projeto contou com o apoio de diversas ferramentas modernas:

- **Git:** Sistema de controle de versão distribuído para rastreamento de mudanças no código.
- **Docker:** Plataforma de containerização utilizada para facilitar a implantação e garantir consistência entre ambientes de desenvolvimento e produção.
- **ESLint e Prettier:** Ferramentas de linting e formatação de código para manter a qualidade e consistência do código TypeScript/JavaScript.
- **Swagger/OpenAPI:** Documentação automática da API REST, facilitando o consumo dos endpoints pelo frontend.

Esta pilha tecnológica moderna e bem estabelecida no mercado garante que o sistema seja mantável, escalável e alinhado com as melhores práticas da indústria de desenvolvimento de software.

## 4.3 Banco de Dados

A modelagem do banco de dados é um dos pilares fundamentais do sistema, pois define como as informações são estruturadas, armazenadas e relacionadas. O banco de dados foi projetado seguindo os princípios de normalização, garantindo a integridade referencial e minimizando redundâncias. O esquema completo do banco de dados está apresentado na Figura 1.

### 4.3.1 Modelo Entidade-Relacionamento

O sistema foi modelado utilizando o PostgreSQL como SGBD, aproveitando seus recursos avançados como tipos de dados customizados (ENUMs), triggers, funções e índices otimizados. A estrutura do banco de dados foi organizada em tabelas que representam as principais entidades do domínio do problema.

#### Entidades Principais:

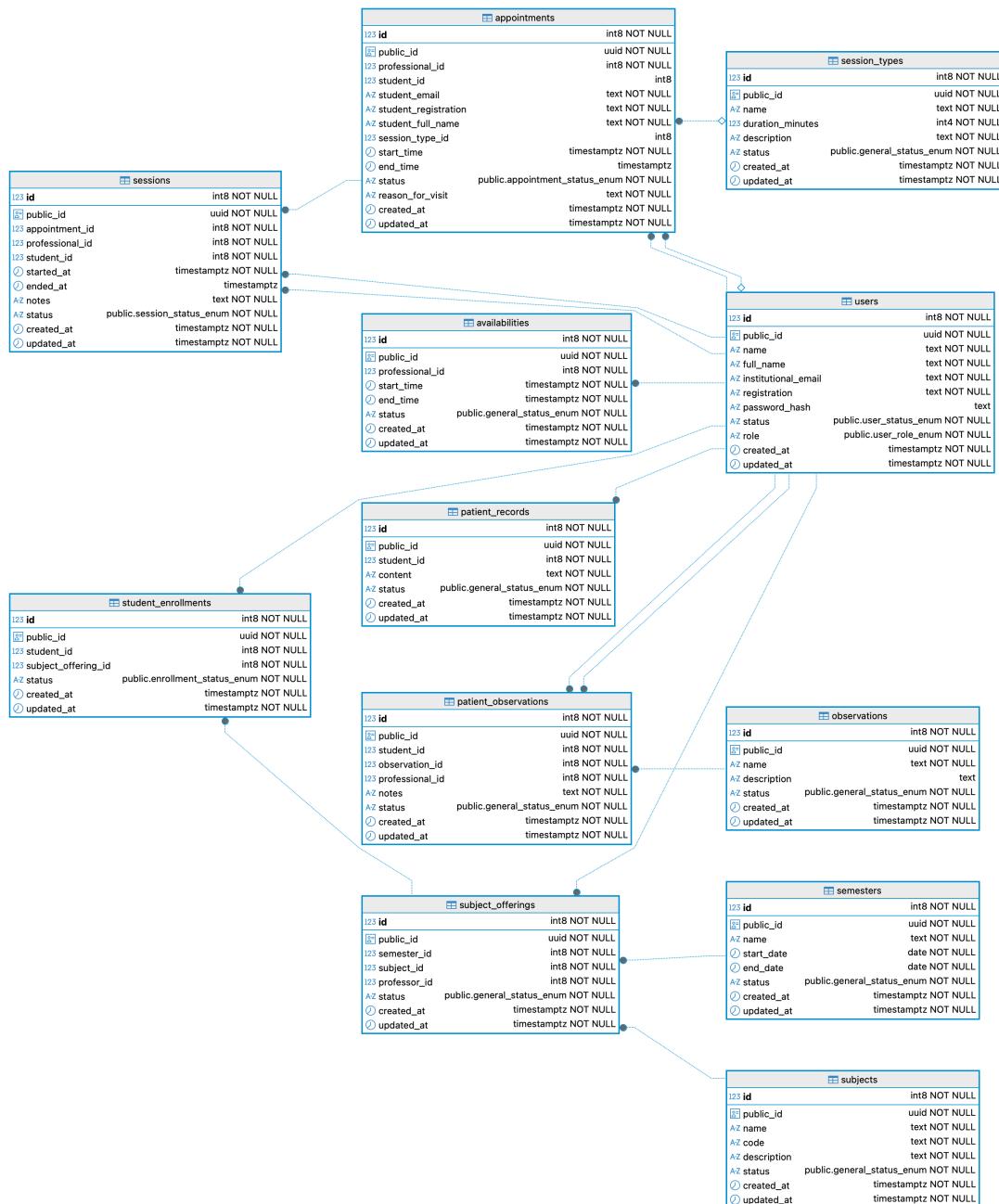
1. **users (Usuários):** Armazena todos os usuários do sistema, independente do perfil (Admin, Profissional, Estudante, Assistente, Professor). Contém dados de identificação, credenciais de acesso e status da conta. A coluna `role` define o tipo de perfil através de um ENUM, permitindo controle de acesso granular.

2. **availabilities (Disponibilidades):** Registra os períodos de disponibilidade configurados pelos profissionais. Cada registro possui horário de início e fim, permitindo que o sistema exiba ao público apenas os horários em que o profissional está disponível para atendimento.
3. **appointments (Agendamentos):** Representa as solicitações de atendimento. Armazena informações do estudante (mesmo que não cadastrado no sistema), do profissional, horários e status do agendamento. O status é controlado por um ENUM com os valores: **pending** (pendente), **confirmed** (confirmado), **in\_session** (em sessão), **cancelled** (cancelado) e **completed** (concluído).
4. **sessions (Sessões):** Registra as sessões de atendimento efetivamente realizadas. Está vinculada a um agendamento confirmado e armazena horário de início, término e anotações do profissional. As anotações são armazenadas de forma criptografada para garantir sigilo.
5. **session\_types (Tipos de Sessão):** Tabela de configuração que define os tipos de atendimento disponíveis (ex: "Primeira Consulta", "Retorno", "Orientação"). Cada tipo possui nome, descrição e duração padrão em minutos.
6. **patient\_records (Prontuários):** Armazena os prontuários eletrônicos dos estudantes. O conteúdo é criptografado, garantindo que apenas profissionais autorizados possam acessar informações sensíveis.
7. **observations (Tipos de Observação):** Tabela de configuração que categoriza os tipos de observações que podem ser feitas sobre estudantes (ex: "Dificuldade de Concentração", "Necessita Acompanhamento").
8. **patient\_observations (Observações de Pacientes):** Registra observações específicas feitas por profissionais sobre estudantes. Estas observações podem ser compartilhadas de forma controlada com professores que lecionam para o estudante em questão.
9. **semesters (Semestres):** Representa os períodos letivos, com datas de início e fim. Permite organizar ofertas de disciplinas por período acadêmico.
10. **subjects (Disciplinas):** Cadastro das disciplinas oferecidas pela instituição, contendo código, nome e descrição.
11. **subject\_offerings (Ofertas de Disciplinas):** Vincula disciplinas a semestres e professores, representando uma turma específica sendo oferecida em um período letivo.

12. **student\_enrollments** (**Matrículas**): Registra as matrículas dos estudantes nas turmas.

Esta tabela é crucial para o controle de acesso dos professores às observações, pois um professor só pode visualizar informações de estudantes matriculados em suas turmas.

Figura 1: Esquema do Banco de Dados



Fonte: O Autor (2025)

### 4.3.2 Estratégias de Segurança e Integridade

O banco de dados implementa diversas estratégias para garantir segurança e integridade:

**Integridade Referencial:** Todas as relações entre tabelas são garantidas através de chaves estrangeiras (FOREIGN KEY), impedindo inconsistências como referências a registros inexistentes.

**Identificadores Públicos:** Cada tabela possui dois tipos de identificadores:

- `id`: Chave primária sequencial interna (BIGINT), utilizada para otimização de índices e joins.
- `public_id`: UUID público exposto pela API, impedindo enumeração de registros e aumentando a segurança.

**Tipos Enumerados (ENUMs):** Status e perfis de usuário são implementados como ENUMs nativos do PostgreSQL (ex: `user_role_enum`, `appointment_status_enum`), garantindo que apenas valores válidos sejam armazenados e facilitando queries.

**Triggers de Atualização:** Um trigger `update_modified_column()` é aplicado a todas as tabelas, atualizando automaticamente o campo `updated_at` sempre que um registro é modificado, permitindo auditoria de alterações.

**Índices Otimizados:** O banco possui índices estratégicos em campos frequentemente utilizados em consultas:

- Índices em chaves estrangeiras para otimizar joins
- Índices compostos em (`start_time`, `end_time`) para consultas de disponibilidade e agendamentos
- Índices em campos de busca como `student_id`, `professional_id`, `semester_id`

**Constraints de Validação:** O banco implementa validações a nível de dados:

- CHECK constraints para garantir que datas de término sejam posteriores às datas de início
- CHECK constraints para validar formato de emails (presença de @)
- UNIQUE constraints para prevenir duplicação de dados críticos

**Criptografia:** Embora a criptografia dos prontuários seja implementada na camada de aplicação (antes de persistir no banco), o PostgreSQL oferece suporte adicional para criptografia de dados em repouso através de extensões como pgcrypto, caso necessário no futuro.

Esta modelagem robusta e bem estruturada garante que o sistema seja escalável, mantível e seguro, atendendo aos requisitos funcionais e não funcionais especificados, especialmente no que tange à proteção de dados sensíveis.

## 4.4 User Stories

As user stories foram desenvolvidas para detalhar os requisitos funcionais do sistema de forma compreensível para os stakeholders e para orientar o desenvolvimento. Cada user story segue o formato proposto por Cohn (2004), descrevendo uma funcionalidade do ponto de vista do usuário final, com critérios de aceitação bem definidos.

### 4.4.1 User Stories Principais

#### US001 - Autenticação no Sistema

Como um usuário do sistema (profissional ou professor), eu gostaria de fazer login com meu email institucional e senha para acessar as funcionalidades disponíveis para meu perfil.

##### Critérios de Aceitação:

- O sistema deve exibir um formulário de login com campos de email e senha
- Credenciais inválidas devem retornar uma mensagem de erro clara
- Após login bem-sucedido, um token JWT deve ser gerado e armazenado
- O token deve expirar após um período determinado de inatividade
- O usuário deve poder fazer logout, invalidando seu token

#### US002 - Agendar Atendimento (Visitante)

Como um visitante não autenticado, eu gostaria de visualizar a disponibilidade dos profissionais e solicitar um agendamento fornecendo meus dados básicos (nome, matrícula, email).

##### Critérios de Aceitação:

- A página pública de agendamento deve estar acessível sem login
- O calendário deve exibir apenas horários com disponibilidade confirmada
- O formulário deve aceitar dados do solicitante e do estudante
- Após submeter, o sistema deve gerar um agendamento com status "pendente"
- Um email de confirmação deve ser enviado imediatamente

### **US003 - Gerenciar Agendamentos (Profissional)**

Como um profissional, eu gostaria de visualizar todas as solicitações de agendamento pendentes e poder confirmá-las ou cancelá-las.

#### **Critérios de Aceitação:**

- O painel deve listar agendamentos organizados por status
- Cada agendamento deve exibir: nome do estudante, data/hora solicitada, motivo da visita
- Ao confirmar, um email de confirmação deve ser enviado ao estudante
- Ao cancelar, o profissional deve poder remarcar para outra data
- O sistema deve validar conflitos de horários antes de confirmar

### **US004 - Iniciar Sessão de Atendimento**

Como um profissional, eu gostaria de iniciar uma sessão a partir de um agendamento confirmado para registrar as anotações do atendimento.

#### **Critérios de Aceitação:**

- O botão de iniciar sessão deve estar disponível apenas para agendamentos confirmados
- Ao iniciar, a data/hora de início deve ser registrada automaticamente
- O sistema deve impedir que o profissional inicie múltiplas sessões simultaneamente
- Um editor de texto rico deve estar disponível para registrar anotações
- Ao finalizar, a data/hora de término deve ser registrada automaticamente

### **US005 - Configurar Disponibilidade**

Como um profissional, eu gostaria de definir meus horários de disponibilidade para que os estudantes possam visualizar quando posso atender.

#### **Critérios de Aceitação:**

- O sistema deve permitir cadastrar períodos de disponibilidade com hora de início e fim
- Deve ser possível configurar múltiplos períodos (ex: segunda a sexta de 8h às 12h)
- Disponibilidades que já possuem agendamentos não devem poder ser removidas
- O calendário público deve refletir as alterações em tempo real
- As disponibilidades devem ser exibidas no fuso horário de Manaus (UTC-4)

### **US006 - Visualizar Disciplinas e Alunos (Professor)**

Como um professor, eu gostaria de visualizar as disciplinas que leciono neste semestre e os alunos matriculados em cada uma delas.

#### **Critérios de Aceitação:**

- O painel do professor deve listar disciplinas organizadas por semestre
- Ao clicar em uma disciplina, deve exibir a lista completa de alunos
- O sistema deve permitir visualizar dados básicos de cada aluno (nome, matrícula)
- Os dados devem estar sincronizados com as matrículas registradas no sistema

### **US007 - Visualizar Observações de Alunos**

Como um professor, eu gostaria de visualizar as observações registradas pelos profissionais sobre meus alunos para acompanhar melhor seu desenvolvimento.

#### **Critérios de Aceitação:**

- As observações devem estar disponíveis na visualização de detalhes do aluno
- Apenas observações de alunos matriculados nas disciplinas do professor devem ser visíveis

- Cada observação deve exibir: data, tipo, descrição e profissional que a registrou
- O professor não deve ter permissão para editar ou deletar observações
- As observações devem estar categorizadas por tipo

#### **US008 - Gerenciar Usuários (Administrador/Profissional)**

Como um administrador do sistema, eu gostaria de criar, editar e desativar contas de usuários, além de atribuir perfis apropriados.

##### **Critérios de Aceitação:**

- O painel de administração deve permitir buscar usuários por nome ou email
- Ao criar um novo usuário, uma senha temporária deve ser gerada
- O sistema deve permitir alterar o perfil de um usuário existente
- Usuários desativados não devem conseguir fazer login
- Todas as ações de alteração devem ser registradas com data/hora

#### **US009 - Gerenciar Semestres e Disciplinas (Administrador/Profissional)**

Como um administrador, eu gostaria de cadastrar semestres letivos, disciplinas e vincular professores a disciplinas para estruturar a oferta acadêmica.

##### **Critérios de Aceitação:**

- O sistema deve permitir criar semestres com datas de início e fim
- Disciplinas devem ter código único, nome e descrição
- A vinculação professor-disciplina-semestre deve estar clara e organizável
- O sistema deve impedir datas inválidas (fim anterior ao início)
- Deve ser possível visualizar histórico de ofertas anteriores

#### **US010 - Notificação de Agendamento Confirmado**

Como um estudante, eu gostaria de receber um email confirmando que meu agendamento foi aceito pelo profissional.

##### **Critérios de Aceitação:**

- O email deve ser enviado em português brasileiro
- Deve conter: nome do profissional, data/hora confirmada, local/modo de atendimento
- O email deve incluir instruções sobre como preparar-se para a sessão
- O email deve ser enviado dentro de 5 minutos após confirmação
- Deve haver mecanismo de retry caso o envio falhe inicialmente

## 5 RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados obtidos com o desenvolvimento do Sistema de Apoio Acadêmico, demonstrando as principais funcionalidades implementadas e como elas atendem aos requisitos definidos. Serão exibidas as interfaces do sistema e discutidos os benefícios trazidos pela automação dos processos de agendamento e acompanhamento de estudantes.

### 5.1 Apresentação do Sistema

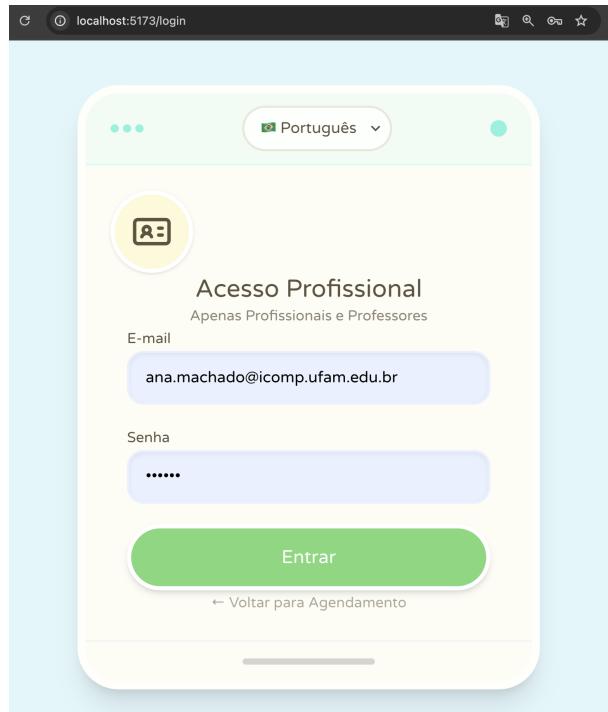
O sistema foi implantado e validado em ambiente de homologação, permitindo testar todos os fluxos de uso previstos. A interface foi projetada para ser intuitiva e responsiva, garantindo acessibilidade em diferentes dispositivos.

#### 5.1.1 Autenticação e Controle de Acesso

A tela de login (Figura 2) é o ponto de entrada para os usuários autenticados (profissionais, professores e administradores). O sistema utiliza autenticação via token JWT, garantindo segurança na comunicação.

A segurança dos dados foi um resultado crítico alcançado. Todas as senhas são armazenadas com hash criptográfico (Figura 3), garantindo que mesmo em caso de acesso indevido ao banco de dados, as credenciais dos usuários permaneçam protegidas.

Figura 2: Tela de Login do Sistema



Fonte: O Autor (2025)

Figura 3: Armazenamento de Senhas com Hash Criptográfico

	A2 full_name	A2 password_hash
1	Ana Machado	\$2a\$11\$g8gbTNYZSlitWn5uuQXnu.1sUtpcifeNpjYGyaF33Z7y6Erk7g3.

Fonte: O Autor (2025)

### 5.1.2 Fluxo de Agendamento Público

Uma das principais funcionalidades do sistema é permitir que qualquer usuário visualize a disponibilidade dos profissionais e solicite um agendamento. A Figura 4 mostra a visualização do calendário público, onde é possível preencher e solicitar um atendimento.

Figura 4: Formulário de Agendamento Público

The screenshot shows a web-based appointment scheduling form titled "Agendar Atendimento". The form is in Portuguese and includes the following fields:

- Informações Pessoais:**
  - Nome Completo: Esther Vieira e Silva
  - E-mail Institucional: nelson16.code@gmail.com
  - Matrícula: 22152282
  - Motivo da Consulta: Consulta de rotina.
- Selecionar Profissional:** Ana Machado
- Horários Disponíveis:** dom., 7 de dez. 20:30

Fonte: O Autor (2025)

Após preencher os dados e selecionar um horário, é mostrado o resumo do agendamento (Figura 5).

Figura 5: Resumo do Agendamento



Fonte: O Autor (2025)

Após a revisão do formulário, o sistema valida os dados e envia a solicitação (Figura 6).

Figura 6: Confirmação da Solicitação de Agendamento



Fonte: O Autor (2025)

O sistema dispara automaticamente um email de confirmação de recebimento para o solicitante (Figura 7), garantindo feedback imediato sobre a ação realizada.

Figura 7: Email de Confirmação de Agendamento



Fonte: O Autor (2025)

### 5.1.3 Gestão de Agendamentos pelo Profissional

O profissional possui um painel administrativo onde pode visualizar todas as solicitações recebidas (Figura 8). O sistema organiza os agendamentos por status (pendente, confirmado, cancelado), facilitando o gerenciamento diário.

Figura 8: Painel de Gestão de Agendamentos

The screenshot shows a web-based appointment management system. At the top, there's a header bar with a back arrow, forward arrow, refresh icon, and a search bar containing 'localhost:5173/dashboard/appointments'. Below the header is a sidebar on the left with the user's name 'Ana Machado Profissional' and several navigation options: Disponibilidade, **Agendamentos** (highlighted in green), Sessões, Tipos de Sessão, Observações, Semestres, Disciplinas, Professores, Ofertas, and Alunos. The main content area is titled 'Agendamentos' and subtitle 'Próximas sessões com seus pacientes.' It features a search bar with fields for 'Buscar' (Search), 'Status' (Todos os Status), and 'Tipo de Sessão' (Todos os Tipos). Below the search bar are five cards representing upcoming sessions:

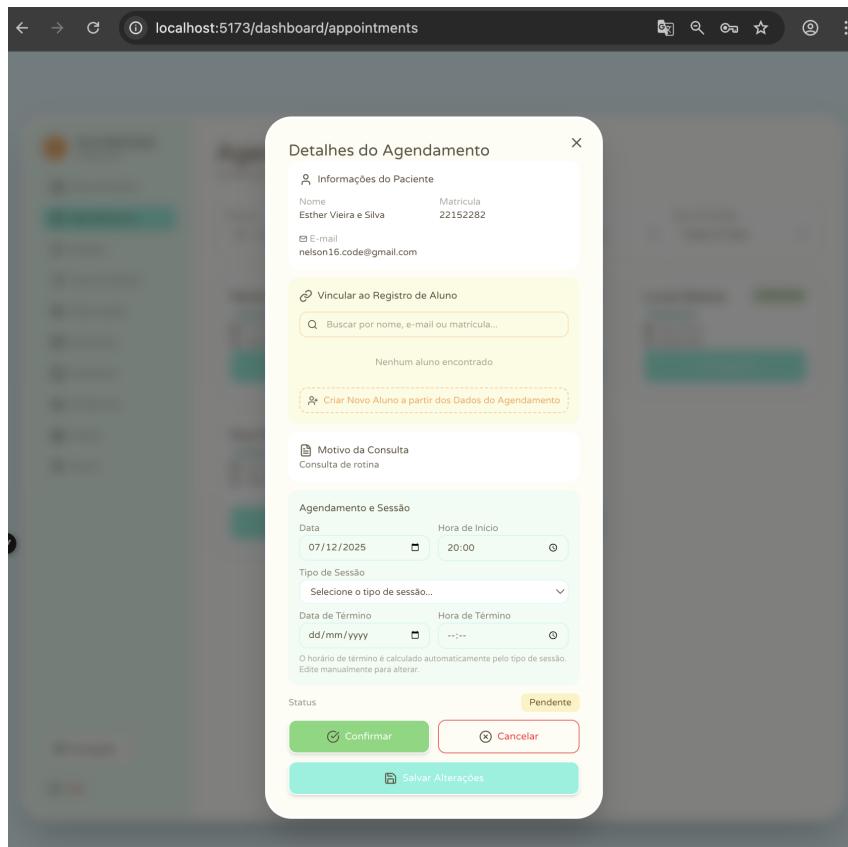
- Nelson Neto** (Concluído): Jubilamento, Fri, Dec 5, 08:00 PM. [Ver Detalhes](#)
- Emmus Souza** (Cancelado): Consulta, Fri, Dec 5, 08:30 PM. [Ver Detalhes](#)
- Lucas Bastos** (Confirmado): Jubilamento, Sat, Dec 6, 08:00 PM. [Ver Detalhes](#)
- Raul Mirabal** (Concluído): Problemas com os Pais, Sat, Dec 6, 09:00 PM. [Ver Detalhes](#)
- Esther Vieira e Silva** (Pendente): Sem tipo de sessão, Sun, Dec 7, 08:00 PM. [Ver Detalhes](#)

At the bottom of the dashboard are language selection ('Português') and exit ('Sair') buttons.

Fonte: O Autor (2025)

A partir deste painel, o profissional pode visualizar detalhes de cada agendamento (Figura 9) e tomar ações como confirmar, cancelar ou remarcar.

Figura 9: Detalhes de um Agendamento



Fonte: O Autor (2025)

O sistema permite cadastrar um novo aluno com base nas informações do agendamento, ou mostra que o aluno já existe

Figura 10: Vincular Estudante ao Agendamento



Fonte: O Autor (2025)

Figura 11: Confirmação de Vinculação do Estudante ao Agendamento



Fonte: O Autor (2025)

Em seguida, é necessário escolher o tipo de atendimento (visível apenas para o profissional) que implica na duração da sessão.

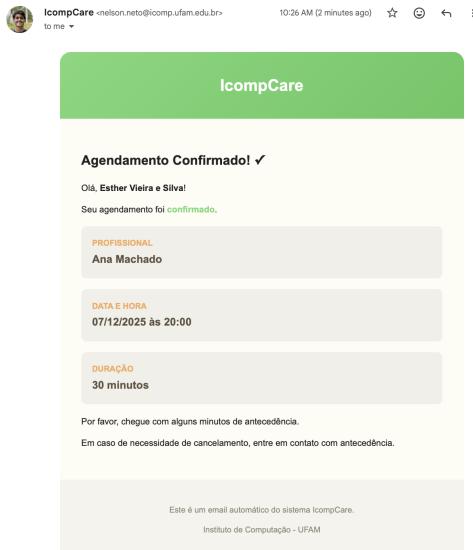
Figura 12: Seleção do Tipo de Atendimento



Fonte: O Autor (2025)

Após a confirmação pelo profissional, um email é enviado ao estudante contendo todos os detalhes do agendamento (Figura 13), incluindo data, horário e informações sobre o atendimento.

Figura 13: Email de Confirmação de Agendamento



Fonte: O Autor (2025)

Caso seja necessário cancelar, o sistema também envia notificação apropriada ao estudante (Figura 14).

Figura 14: Email de Cancelamento de Agendamento

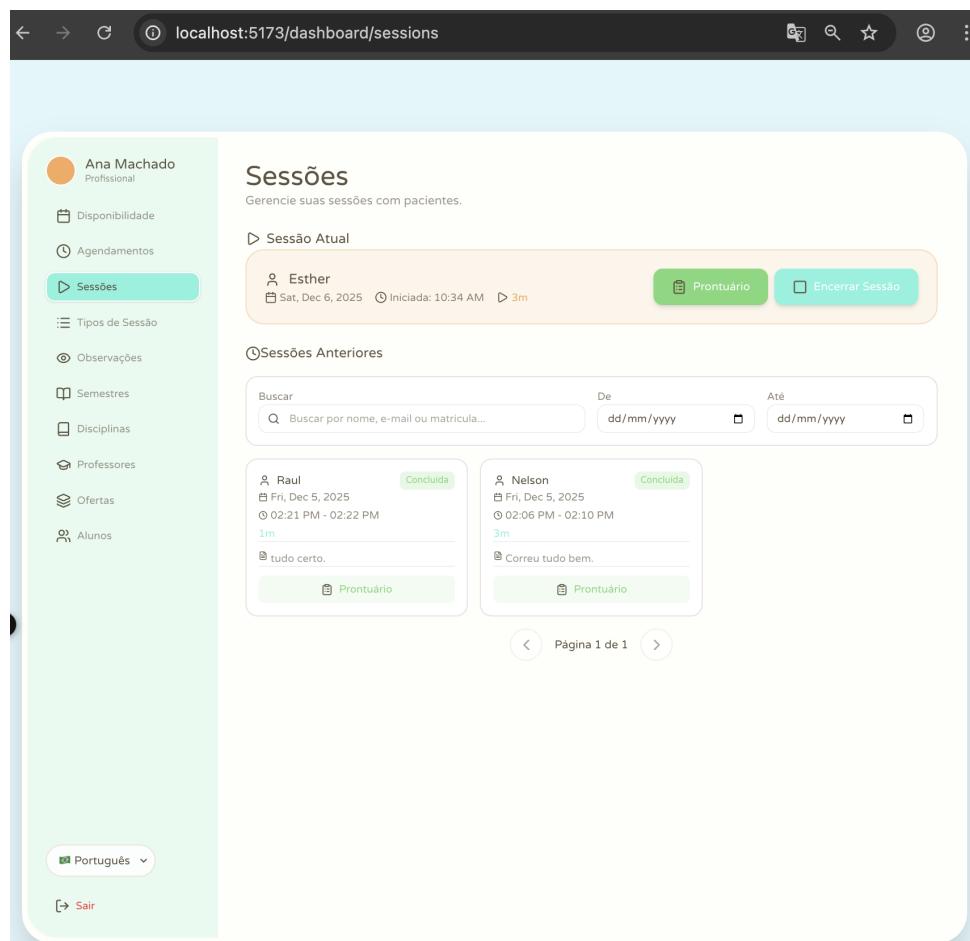


Fonte: O Autor (2025)

### 5.1.4 Realização de Sessões e Prontuário Eletrônico

Para os atendimentos confirmados, o sistema oferece uma interface dedicada para a realização da sessão, onde é mostrado a atual sessão ativa e sessões anteriores (Figura 15).

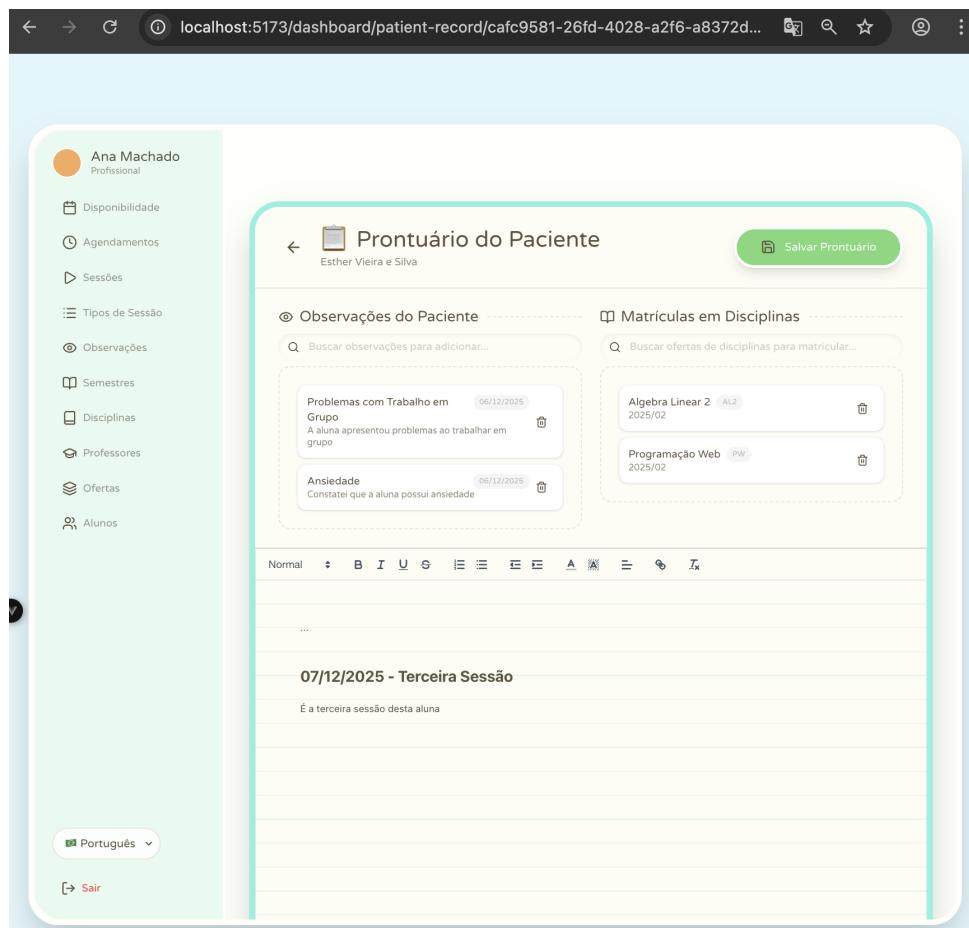
Figura 15: Tela de Sessões do Profissional



Fonte: O Autor (2025)

Durante a sessão, o profissional pode escrever o Prontuário Eletrônico utilizando um editor de texto rico (Figura 16) para organizar melhor suas anotações.

Figura 16: Prontuário Eletrônico com Editor de Texto Rico



Fonte: O Autor (2025)

Ao finalizar a sessão (Figura 17), é possível adicionar uma nota específica para validação futura.

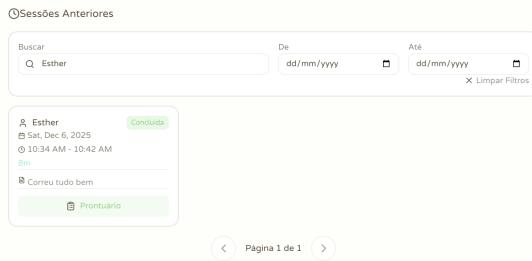
Figura 17: Encerramento da Sessão



Fonte: O Autor (2025)

O histórico de sessões anteriores pode ser consultado (Figura 18), permitindo ao profissional acompanhar a evolução do estudante ao longo do tempo. O acesso aos detalhes de sessões passadas permite uma continuidade efetiva no acompanhamento, com todas as anotações anteriores disponíveis de forma organizada e segura.

Figura 18: Histórico de Sessões do Estudante



Fonte: O Autor (2025)

Figura 19: Conclusão do Agendamento



Fonte: O Autor (2025)

Um ponto crucial é a segurança dos dados: todas as anotações salvas são criptografadas antes de serem persistidas no banco de dados (Figura 20), garantindo o sigilo das informações sensíveis dos estudantes.

Figura 20: Dados do Prontuário Eletrônico Criptografados

	<code>id</code>	<code>public_id</code>	<code>student_id</code>	<code>content</code>	<code>status</code>
1	3	cc81982-a8a1-4fb1-9e0b-2504f5e7c85	5	0AKys3Q0hlRwEQ9S9mBDOSnVSsc2PlVz0ze1ZzfpXT4eOc4y	active
2	36	caf9581-26fd-4028-a2f6-a8372d44bf9	9	k5WhJCa5blgNW/OhmQCR94uyMnnrHcf/PytkVK3YbAhgas2Ui	active

Fonte: O Autor (2025)

### 5.1.5 Área do Professor

Os professores têm acesso a uma área específica onde podem visualizar as disciplinas que lecionam e a lista de alunos matriculados (Figura 21).

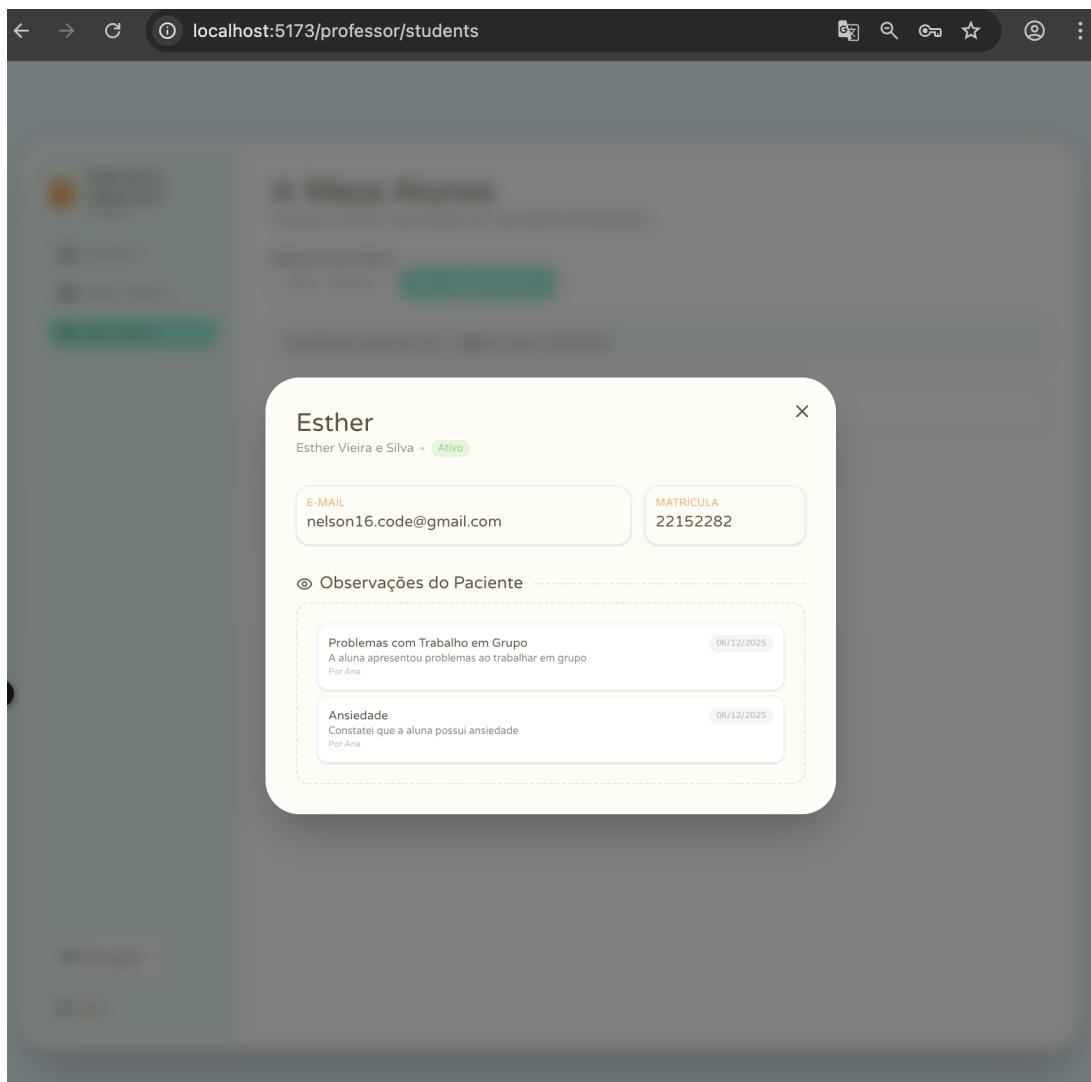
Figura 21: Painel do Professor - Lista de Alunos por Disciplina Ofertada

The screenshot shows a web browser window with the URL `localhost:5173/Professor/students`. On the left, there is a sidebar with the user profile "Nakamura Nakamura Professor" and navigation links for "Disciplinas" and "Minhas Ofertas". The main content area is titled "Meus Alunos" and displays a message: "Visualize os alunos matriculados em suas ofertas de disciplinas." Below this, a button "Selecionar uma Oferta" is shown with two options: "CAL2 - Cálculo 2" and "AL2 - Algebra Linear 2", with "AL2 - Algebra Linear 2" being highlighted. A message below says "Visualizando alunos de: AL2 - Algebra Linear 2 (2025/02)". There is a search bar with the placeholder "Buscar por nome, e-mail ou matrícula...". Two student cards are displayed: Esther (Matrícula: 22152282, E-mail: nelson16.code@gmail.com) and Raul (Matrícula: BEBE, E-mail: raul@icomp.com). At the bottom, there are navigation arrows and the text "Página 1 de 1". The footer includes language selection ("Português") and a "Sair" (Logout) link.

Fonte: O Autor (2025)

Ao acessar os detalhes de um aluno, o professor pode visualizar observações compartilhadas pelos profissionais de apoio (Figura 22). Esta funcionalidade promove a integração entre o apoio psicopedagógico e o corpo docente, permitindo um acompanhamento mais holístico do estudante.

Figura 22: Detalhes do Aluno e Observações



Fonte: O Autor (2025)

## 5.2 Discussão dos Resultados

A implementação do sistema traz melhorias significativas em relação ao processo manual anterior. A seguir, discutem-se os principais ganhos observados.

### 5.2.1 Eficiência Operacional

A automação do agendamento reduz drasticamente o tempo gasto pela profissional com a gestão de horários. Anteriormente, a confirmação de um único atendimento poderia exigir

a troca de múltiplos emails e encontros presenciais. Com o sistema, o estudante visualiza a disponibilidade real e solicita o horário desejado, restando ao profissional apenas a aprovação com um clique.

### **5.2.2 Segurança da Informação**

A substituição de anotações em papel ou arquivos de texto simples por um prontuário eletrônico criptografado elevou o nível de segurança. O controle de acesso baseado em perfis garante que apenas pessoas autorizadas visualizem dados sensíveis.

### **5.2.3 Integração e Acompanhamento**

A funcionalidade de compartilhamento de observações com professores criou um canal formal e seguro de comunicação. Isso facilita a identificação precoce de dificuldades acadêmicas e permite que os professores adaptem suas estratégias pedagógicas baseados em informações concretas fornecidas pelo apoio psicopedagógico.

Em suma, o sistema atingiu os objetivos propostos, entregando uma solução robusta, segura e eficiente para o gerenciamento do apoio acadêmico no ICOMP.

## 6 CONCLUSÕES

O desenvolvimento do Sistema de Apoio Acadêmico para o Instituto de Computação (ICOMP) da UFAM representou um passo importante na modernização e eficiência dos processos de atendimento aos estudantes. Este trabalho alcançou seu objetivo principal de projetar e implementar uma solução web capaz de gerenciar agendamentos, prontuários e o acompanhamento acadêmico de forma segura e integrada.

A adoção de tecnologias modernas como .NET 8 (MICROSOFT, 2025) e Vue.js 3 (VUE.JS, 2025) permitiu a construção de uma aplicação robusta, escalável e com excelente experiência de usuário. A arquitetura limpa (Clean Architecture) utilizada no backend garante que o sistema seja de fácil manutenção e evolução, permitindo a incorporação de novas funcionalidades sem comprometer a estabilidade do código existente.

### 6.1 Contribuições

As principais contribuições deste trabalho podem ser resumidas em:

- **Otimização do Fluxo de Trabalho:** A automação do agendamento eliminou gargalos operacionais, permitindo que a profissional de apoio dedique mais tempo ao atendimento dos alunos e menos a tarefas burocráticas.
- **Segurança e Privacidade:** A implementação de criptografia nos prontuários e o controle rigoroso de acesso asseguram a confidencialidade das informações, atendendo a requisitos legais e éticos.
- **Integração Pedagógica:** A ferramenta de comunicação entre o apoio psicopedagógico e os professores fortalece a rede de suporte ao estudante, permitindo ações mais coordenadas e eficazes.
- **Registro Histórico:** A centralização dos dados permite a geração futura de relatórios

e estatísticas que podem embasar decisões institucionais sobre políticas de assistência estudantil.

## 6.2 Trabalhos Futuros

Apesar de o sistema atender aos requisitos iniciais, o desenvolvimento de software é um processo contínuo. Como sugestões para trabalhos futuros e evolução da plataforma, destacam-se:

- **Integração com Sistemas da UFAM:** Implementar integração com o sistema acadêmico oficial da universidade para importação automática de dados de alunos, disciplinas e matrículas, eliminando a necessidade de cadastro manual.
- **Módulo de Relatórios Avançados:** Desenvolver um dashboard com indicadores de desempenho (KPIs), como número de atendimentos por período, principais motivos de procura e taxas de evasão, auxiliando na gestão estratégica.
- **Notificações via WhatsApp:** Expandir o sistema de notificações para incluir envio de mensagens via WhatsApp, aumentando a taxa de visualização e confirmação por parte dos estudantes.
- **App Mobile:** Desenvolver um aplicativo móvel nativo para estudantes, facilitando ainda mais o acesso ao agendamento e acompanhamento de suas solicitações.

Conclui-se que o sistema desenvolvido possui grande potencial para impactar positivamente a comunidade acadêmica do ICOMP, servindo como modelo que pode ser expandido para outras unidades da universidade.

## REFERÊNCIAS

COHN, M. **User Stories Applied: For Agile Software Development.** [S.l.]: Addison-Wesley Professional, 2004.

KIERAS, R. W. **Sistema para Agendamento de Serviços.** Monografia (TCC) — Universidade Tecnológica Federal do Paraná, Campo Mourão, 2019.

LEÃO, A. M. *et al.* Prevalência e fatores associados à depressão e ansiedade entre estudantes universitários da Área da saúde de um grande centro urbano do nordeste do brasil. **Revista Brasileira de Educação Médica**, v. 42, n. 4, p. 55–65, 2018.

MARTIN, R. C. **Clean Architecture: A Craftsman’s Guide to Software Structure and Design.** [S.l.]: Prentice Hall, 2017.

MICROSOFT. **ASP.NET Core Documentation.** [S.l.], 2025. Disponível em: <<https://learn.microsoft.com/aspnet/core>>. Acesso em: 05 dez. 2025.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL 17 Documentation.** [S.l.], 2025. Disponível em: <<https://www.postgresql.org/docs/17/>>. Acesso em: 05 dez. 2025.

SOUZA, W. Q. d. **Sistema para gerenciamento e agendamento de consultas para psicólogos e clientes.** Monografia (TCC) — Instituto Federal de Goiás, Jataí, 2021.

STEFEN, L. E. *et al.* Systempsi: Sistema gerenciador para psicólogos em atuação remota. In: **Anais do Latinoware.** [S.l.: s.n.], 2022.

VUE.JS. **Vue.js Documentation.** [S.l.], 2025. Disponível em: <<https://vuejs.org/guide/introduction.html>>. Acesso em: 05 dez. 2025.