

Faculdade de computação – Universidade Federal do Pará (UFPA)

**Sistema Web para Troca de Alocação dos Professores nas Salas de forma Dinâmica na
Faculdade de Computação da UFPA de Castanhal**

Valter Gabriel Brito da Silva

Prof. Dr. Thiago Antônio Sidônio Coqueiro

RESUMO

O trabalho apresenta o desenvolvimento e implantação de um sistema web para otimizar a alocação dinâmica de salas de aula na Faculdade de Computação da UFPA, Campus Castanhal. A solução visa automatizar o processo de distribuição de salas, facilitar a troca e reserva de horários por parte dos professores, reduzir conflitos e burocracia, e tornar a gestão acadêmica mais ágil e transparente. O sistema foi desenvolvido utilizando Angular e TypeScript no front-end, ASP.NET 8 e C# no back-end, e PostgreSQL como banco de dados. A proposta busca melhorar a comunicação entre docentes e administração, além de centralizar as informações em uma plataforma acessível e eficiente.

Palavras-chave: Alocação de salas; Sistema web; Automatização; Angular; ASP.NET; Usabilidade.

ABSTRACT

This work presents the development and implementation of a web system to optimize the dynamic allocation of classrooms at the Faculty of Computing of UFPA, Castanhal Campus. The solution aims to automate the process of room distribution, facilitate schedule changes and reservations by teachers, reduce conflicts and bureaucracy, and make academic management more agile and transparent. The system was developed using Angular and TypeScript for the front-end, ASP.NET 8 and C# for the back-end, and PostgreSQL as the database. The proposal seeks to improve communication between faculty and administration, as well as to centralize information on an accessible and efficient platform.

Keywords: Room allocation; Web system; Automation; Angular; ASP.NET; Usability.

¹E-mail do autor: valter.silva@ufpa.com.br

²E-mail do orientador: tcoqueiro@ufpa.br

1. INTRODUÇÃO

A gestão eficiente dos ambientes acadêmicos é fundamental para o funcionamento e a qualidade das instituições de ensino superior (Ferreira, Corrêa e Rissi, 2019). Com o crescimento do número de turmas, cursos e demandas específicas de professores e alunos, torna-se cada vez mais importante contar com métodos que facilitem a organização das atividades acadêmicas.

No contexto da Faculdade de Computação (FACOMP) da Universidade Federal do Pará (UFPA), Campus Castanhal, um dos principais desafios enfrentados é a troca de alocação das salas de aula ao longo do semestre, especialmente quando há necessidade de realocação de professores devido à mudanças de horários.

Atualmente, segundo relato obtido em reunião a coordenação da FACOMP, esse processo de troca é realizado manualmente, exigindo comunicação direta entre os docentes, o que, em alguns momentos, exige o uso de aplicativos de mensagem para comunicação, o que pode gerar um incômodos devido ao trabalho interferir na vida pessoal (Favatto, Both Belém, 2023). Além disso, é necessário um contato direto à direção da faculdade para que as mudanças sejam refletidas no calendário acadêmico, que atualmente é gerenciado utilizando planilhas e o Google Calendário, o que nem sempre ocorre. Esse método, além de ser demorado, pode provocar falhas, retrabalho e sobrecarga administrativa, prejudicando a eficiência e o planejamento das atividades acadêmicas.

Diante dessa situação, o objetivo deste trabalho é mostrar o desenvolvimento de um sistema web que facilite a troca de alocações entre professores ao longo do calendário acadêmico. Com ele, busca-se diminuir a necessidade de comunicação direta com a coordenação ou entre os docentes para realizar essas trocas. Além disso, o sistema também pretende automatizar a alocação de aulas, levando em conta a carga horária de cada disciplina.

2. TRABALHOS CORRELATOS

O trabalho de Lima (2020) apresenta o sistema web *SALAS*, desenvolvido para a automação da alocação de salas na UFC. O sistema realiza a distribuição automática das turmas conforme restrições de capacidade de sala, horários, acessibilidade dos alunos e recursos disponíveis em cada sala. A ferramenta agiliza o processo de organização dos espaços e evita conflitos na alocação, tornando a gestão mais eficiente.

Já a proposta de Oliveira (2018) consiste em um aplicativo online para centralizar as reservas de salas no CINTED/UFRGS. O sistema impede reservas duplicadas, valida agendamentos e oferece uma visão em tempo real da ocupação dos ambientes. Antes da implementação, o controle era realizado por meio de planilhas, o que dificultava a visualização dos horários ocupados e não evitava conflitos de agendamento.

O diferencial da proposta apresentada neste trabalho está na possibilidade de os próprios professores solicitarem e trocarem reservas diretamente pelo sistema, sem depender do setor administrativo ou contato direto entre os professores, promovendo uma gestão mais flexível.

Além disso, o sistema divide automaticamente as disciplinas ao longo da semana, levando em conta a carga horária, eliminando ajustes manuais repetitivos. Diferentemente de outras soluções que focam na montagem inicial da grade semestral, o sistema prioriza que mudanças e trocas durante o semestre sejam realizadas de forma simples e rápida, representando um avanço em relação aos trabalhos analisados.

3. FERRAMENTAS UTILIZADAS

3.1 Github

O GitHub é uma plataforma de versionamento de código baseada em Git, amplamente utilizada para colaboração em projetos de software (GITHUB, 2024). No contexto deste trabalho, foi escolhido para organizar as tarefas e acompanhar o progresso do sistema, aproveitando recursos como o controle de versões e o uso de issues para registro das demandas. Essa escolha trouxe maior segurança no armazenamento do código e facilitou a gestão das etapas de desenvolvimento.

3.2 Angular

O Angular é um framework de desenvolvimento web mantido pelo Google que adota uma arquitetura baseada em componentes e utiliza TypeScript como linguagem principal (GOOGLE, 2023). A escolha do Angular neste projeto se deve à sua solidez e ao suporte da comunidade, além de oferecer recursos que permitem a criação de interfaces modulares.

3.3 TypeScript

O TypeScript é uma linguagem que estende o JavaScript com recursos de tipagem estática e orientação a objetos (MICROSOFT, 2024). Sua utilização no desenvolvimento do frontend deste trabalho proporcionou maior confiabilidade ao código, permitindo identificar erros em tempo de compilação e facilitando a manutenção futura do sistema.

3.4 .NET

O .NET, plataforma de desenvolvimento da Microsoft (MICROSOFT, 2024), foi adotado para a implementação do backend do sistema, utilizando a linguagem CSharp. Essa combinação foi escolhida por oferecer integração nativa com bancos de dados e suporte a criação de APIs. No contexto do trabalho, o uso do .NET e do CSharp permitiu desenvolver funcionalidades críticas, como notificações automáticas e gerenciamento das trocas de alocação.

3.5 PostgreSQL

O PostgreSQL, banco de dados relacional open-source (POSTGRES GLOBAL DEVELOPMENT GROUP, 2024), foi selecionado por sua confiabilidade e compatibilidade com o .NET. Além de suportar transações, ele possibilitou lidar de forma eficiente com as relações

entre professores, salas, turmas e disciplinas. Essa escolha contribuiu para a consistência dos dados e para a escalabilidade do sistema.

4. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, abordaremos os conceitos fundamentais para o desenvolvimento desse sistema web.

4.1 API

No sistema desenvolvido, a API, que é um conjunto de protocolos e definições que permite a comunicação padronizada entre diferentes sistemas de software (FIELDING, 2000), e funciona como intermediária entre o frontend (Angular) e o backend (.NET). A API organiza a comunicação entre os módulos, permitindo operações como criação de alocações, solicitação de trocas e envio de notificações. Assim, a API foi um elemento central para integrar as diferentes partes da aplicação e garantir a transparência no fluxo de informações entre professores e administração.

4.2 Usabilidade

A usabilidade, no contexto deste trabalho, refere-se à facilidade com que professores e administradores conseguem interagir com o sistema sem necessidade de treinamento complexo, pois o conceito de usabilidade é o de que qualquer pessoa seja capaz de utilizar um produto com tranquilidade (SEBRAE, 2023). Mais do que um conceito teórico, a preocupação com usabilidade orientou decisões de interface, como a simplificação de formulários e a inclusão de notificações automáticas por e-mail. Dessa forma, buscou-se garantir que o sistema fosse acessível a todos os usuários, independentemente de seu nível de familiaridade com tecnologia.

4.3 Arquitetura Modular

A arquitetura modular é uma abordagem de design de software em que um sistema é dividido em módulos independentes, onde cada um é responsável por uma funcionalidade específica (APPMaster, 2023) e ela foi adotada para estruturar o sistema em partes independentes, como módulos de professores, turmas, disciplinas e salas. Essa abordagem facilitou tanto o desenvolvimento inicial quanto a manutenção e a evolução futura da aplicação. No caso deste trabalho, a modularidade permitiu implementar funcionalidades de forma incremental, garantindo que novos requisitos pudessem ser incorporados sem comprometer as partes já consolidadas do sistema.

5. METODOLOGIA

5.1 Levantamento de Requisitos

O processo de levantamento de requisitos foi realizado através de uma reunião com a coordenação da Faculdade de Computação da UFPA, no campus Castanhal. Para compreender melhor as necessidades relacionadas à alocação e gestão das salas, foi realizada uma entrevista com a direção para entender as demandas institucionais, como a troca dinâmica de professores. Além disso, um questionário foi enviado para os professores com o fito de validar as funcionalidades também do lado docente. O questionário continha as seguintes perguntas:

- I. O sistema precisa permitir que os professores visualizem suas alocações em salas existentes?
- II. O sistema precisa permitir que professores solicitem troca de alocações com outros professores, sem intermédio com a coordenação?
- III. O sistema precisa permitir que outros professores aceitem ou rejeitem solicitações de troca de alocações?
- IV. O sistema precisa permitir que o administrador realize trocas de alocações diretamente, sem necessidade de aceite?
- V. O sistema deve ser acessível por navegador web moderno, sem necessidade de instalação local?

Embora a amostra de respostas ao questionário tenha sido limitada, as contribuições obtidas corroboraram integralmente as funcionalidades e requisitos estabelecidos. O conjunto dessas informações foi essencial para a especificação dos requisitos funcionais e não funcionais do sistema.

5.2 Requisitos Funcionais

Os requisitos funcionais estabelecem o que o sistema deve realizar para atender às demandas dos usuários e garantir que os objetivos do projeto sejam cumpridos (TURINE; MASIERO 1996). No contexto da gestão de salas, destacam-se os seguintes pontos:

- I. RF01: Permitir que o administrador crie, edite e exclua alocações de salas de aula.
- II. RF02: Permitir que professores visualizem suas alocações de sala.
- III. RF04: Permitir que outros professores aceitem ou rejeitem solicitações de troca de alocações.
- IV. RF05: Permitir que o administrador realize trocas de alocações diretamente, sem necessidade de aceite.

5.3 Requisitos Não Funcionais

Os requisitos não funcionais dizem respeito às características de qualidade que o sistema deve apresentar, como desempenho, segurança e facilidade de uso e manutenção (TURINE; MASIERO 1996). Para este projeto, foram definidos os requisitos abaixo:

- I. RNF01: O sistema deve ser acessível por meio de navegador web, sem necessidade de instalação local.
- II. RNF02: A interface do usuário deve garantir boa usabilidade.
- III. RNF04: O sistema deve ser compatível com os principais navegadores (Chrome, Firefox, Edge).
- IV. RNF05: O sistema deve permitir fácil manutenção e evolução futura, adotando arquitetura modular e documentação adequada.

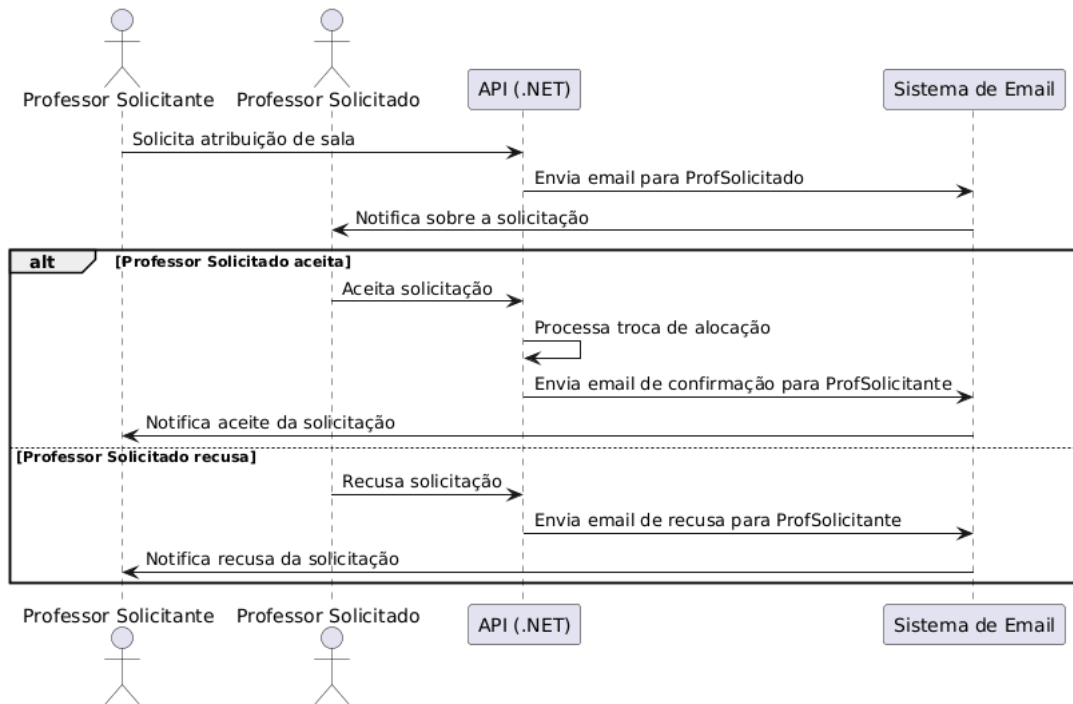
5.4 Modelagem

Na etapa de modelagem, foi buscado representar graficamente a estrutura e o funcionamento do sistema, utilizados diagramas que foram elaborados com a ferramenta PlantUML:

As Figuras 1 e 2 representam o **Diagrama de Casos de Uso**, que tem como objetivo mapear as funcionalidades do sistema e as interações entre os diferentes atores (professores e administradores).

A Figura 1 apresenta o processo de alocação de salas envolve a solicitação realizada por um professor, que é encaminhada para a API desenvolvida em .NET. Essa API é responsável por processar o pedido e enviar um e-mail ao professor solicitado, informando-o sobre a solicitação de alocação. Ao receber o e-mail, o professor solicitado pode optar por aceitar ou recusar a alocação diretamente pelo sistema. Caso o professor aceite, a API registra o aceite, realiza a troca de alocação das salas e notifica o professor solicitante sobre a aprovação por meio de um novo e-mail. Se o pedido for recusado, a API também registra a decisão e comunica ao professor solicitante sobre a recusa, garantindo assim transparência e agilidade em todo o fluxo de negociação e alocação dos ambientes.

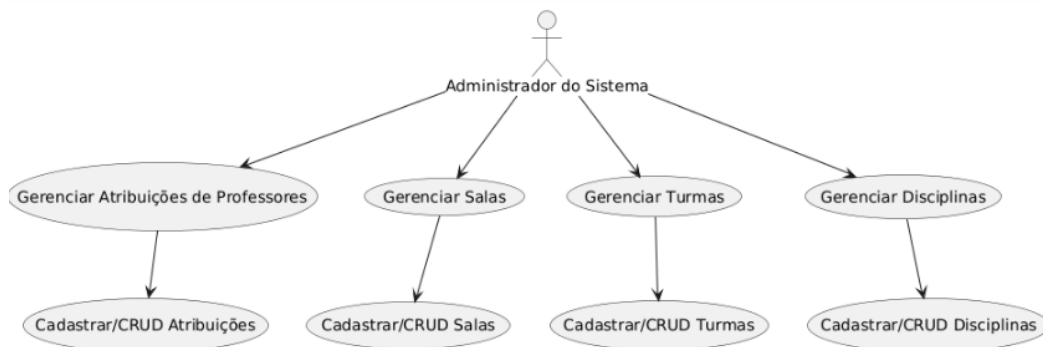
Figure 1: Diagrama de Sequência



Fonte: Elaborado pelo autor usando o PlantUML.

Já a Figura 2 traz o detalhamento dos casos de uso voltados para o administrador, evidenciando as alocações específicas relacionadas à gestão e controle das reservas e alocações de salas.

Figure 2: Diagrama de Casos de Uso – detalhamento

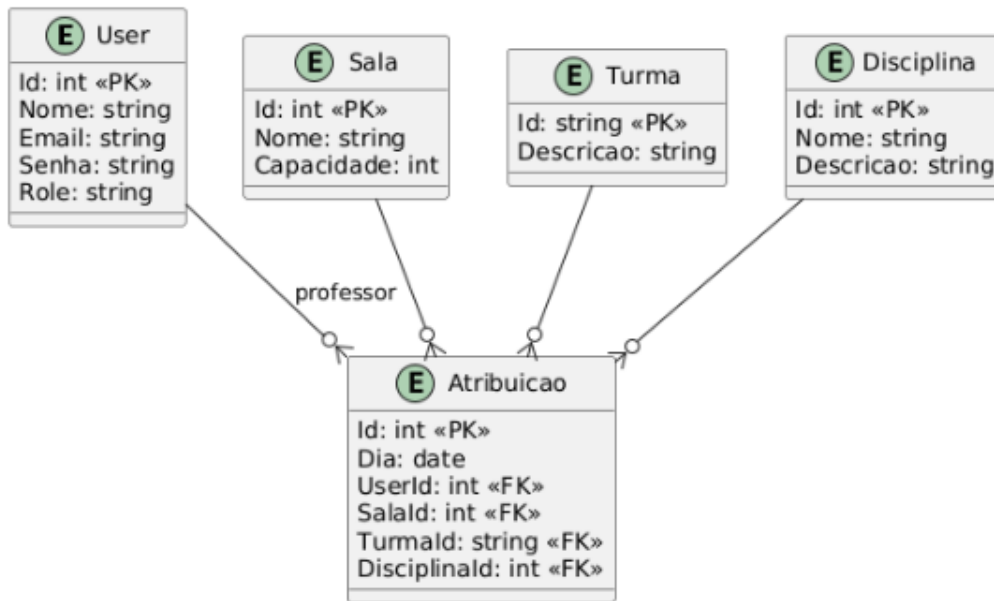


Fonte: Elaborado pelo autor usando o PlantUML.

A Figura 3 apresenta o **Diagrama de Banco de Dados (DER)**, que representa a estrutura de dados do sistema de alocação de salas. Ele é composto pelas entidades User (Professor), Sala, Turma, Disciplina e Atribuicao. A entidade User armazena informações dos usuários, como nome, e-mail, senha e perfil de acesso. Sala contempla dados sobre os ambientes físicos, incluindo nome e capacidade. Turma e Disciplina registram, respectivamente, os grupos de

alunos e as matérias ofertadas, com descrições específicas. A entidade central Atribuicao relaciona essas entidades, representando o processo de alocação de salas, turmas e disciplinas a um determinado usuário em uma data específica. Ela faz uso de chaves estrangeiras para vincular os registros de usuários, salas, turmas e disciplinas, permitindo o controle detalhado das alocações realizadas no sistema.

Figure 3: Diagrama de Banco de Dados (DER)



Fonte: Elaborado pelo autor usando o PlantUML.

5.5 Implementação

Durante o processo de seleção das tecnologias para este projeto, foi priorizada a escolha de ferramentas que combinassem modernidade e facilidade de implementação e manutenção. Para o gerenciamento dos dados, foi selecionado o PostgreSQL.

Para o desenvolvimento do backend, foi utilizado o .NET, que permite o desenvolvimento de APIs seguras. O frontend foi desenvolvido utilizando AngularJS, proporcionando uma interface intuitiva para os usuários do sistema. O versionamento do código foi realizado através do GitHub, uma plataforma que promove o backup seguro do código fonte.

6. INTERFACES DE NAVEGAÇÃO

O sistema oferece uma interfaces que permitem a administração completa dos principais recursos acadêmicos, como professores, salas, turmas, disciplinas e alocações de reservas. Para cada uma dessas entidades, existem telas específicas de listagem e de gerenciamento.

A Figura 4 exemplifica a tela de listagem de professores, apresentando todos os docentes cadastrados no sistema.

Figure 4: Tela de listagem de professores



Fonte: Captura de tela do sistema, realizada pelo autor.

Já a Figura 5 ilustra a tela de cadastro de professores, permitindo incluir novos registros e informações existentes.

Figure 5: Tela de cadastro de professores

The image shows a registration form titled "Cadastrar Professor". The form contains the following fields and elements:

- Nome:** A text input field.
- Telefone:** A text input field.
- Email:** A text input field.
- Cor:** A color selection button with a black horizontal bar.
- Usuário:** A text input field.
- Senha:** A text input field.
- Ativo:** A checkbox that is currently checked.
- Cadastrar:** A blue button with the text "Cadastrar".

Fonte: Captura de tela do sistema, realizada pelo autor.

Da mesma forma, a Figura 6 representa a listagem de disciplinas disponíveis. Essa tela facilita a visualização de todas as disciplinas cadastradas, permitindo ao usuário acessar rapidamente informações relevantes.

Figure 6: Tela de listagem de disciplinas

Nova Disciplina

Lista de Disciplinas

Buscar por nome, código ou sigla...

Nome:
Empreendedorismo ao Agro

Código:
EAOAG

Sigla:
EA2

Carga Horária:
90

Salvar **Cancelar**

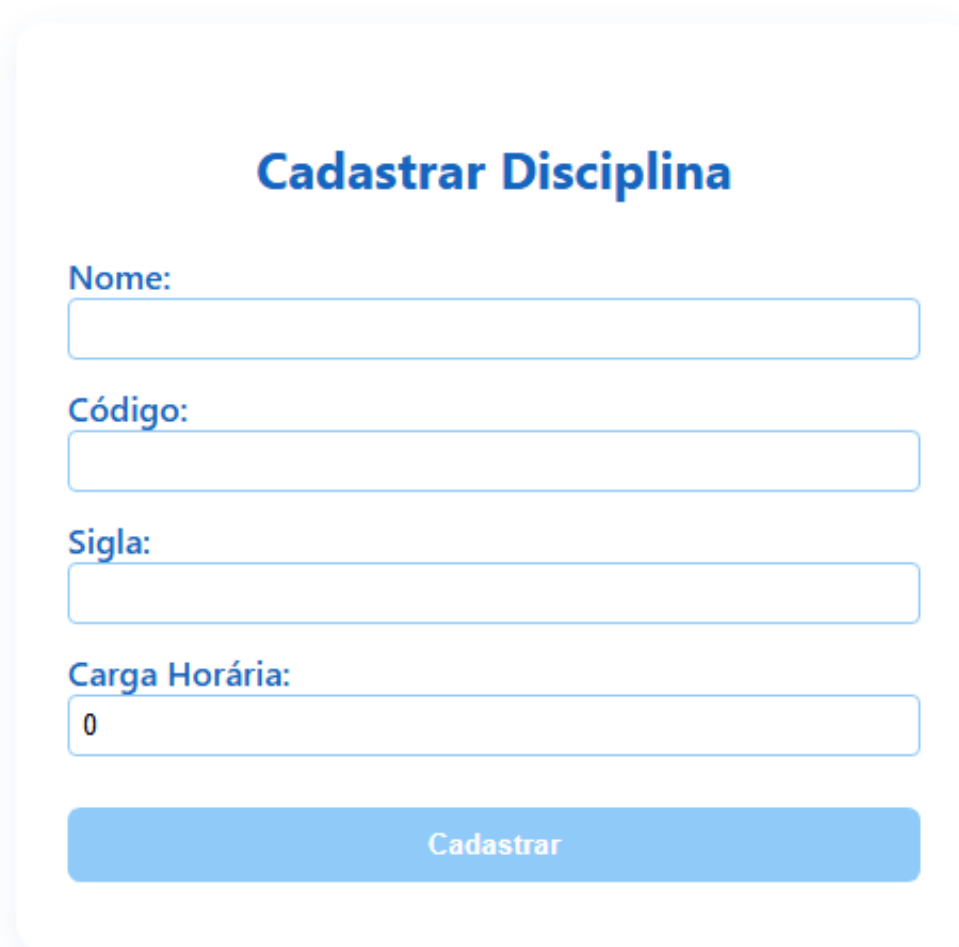
Algoritmos (ALG) Cód: ALG10 CH: 30 Aulas: 7 **Editar**

Calculo 1 (c1) Cód: 1252 CH: 60 Aulas: 15 **Editar**

Fonte: Captura de tela do sistema, realizada pelo autor.

A Figura 7 mostra o cadastro de disciplinas. Uma das principais funções do cadastro de disciplina é a definição da carga horária, que é usada para popular de forma automática os dias de aulas na hora de atribuir essa disciplina para um docente.

Figure 7: Tela de cadastro de disciplinas



The image shows a web form titled "Cadastrar Disciplina" (Register Discipline). The form is contained within a light blue rounded rectangle. It features four input fields, each with a label to its left: "Nome:" (Name), "Código:" (Code), "Sigla:" (Acronym), and "Carga Horária:" (Credit Hours). The "Carga Horária" field contains the number "0". Below the input fields is a blue button with the text "Cadastrar" (Register).

Fonte: Captura de tela do sistema, realizada pelo autor.

As funcionalidades relativas a turmas seguem o mesmo padrão. A Figura 8 exibe a tela de listagem de turmas.

Figure 8: Tela de listagem de turmas

Nova Turma

Turmas

Buscar por nome, bloco ou turno...

Nome:
Sistemas 2020.2

Bloco:
8

Turno:
Vespertino

Salvar Cancelar

Eng 2019.4 Bloco: 11 Turno: Matutino Editar Excluir

Fonte: Captura de tela do sistema, realizada pelo autor.

O cadastro de turmas permite ao usuário inserir novas, ilustrado na Figura 9.

Figure 9: Tela de cadastro de turmas

Criar Nova Turma

Nome da Turma

Bloco/Período/Semestre

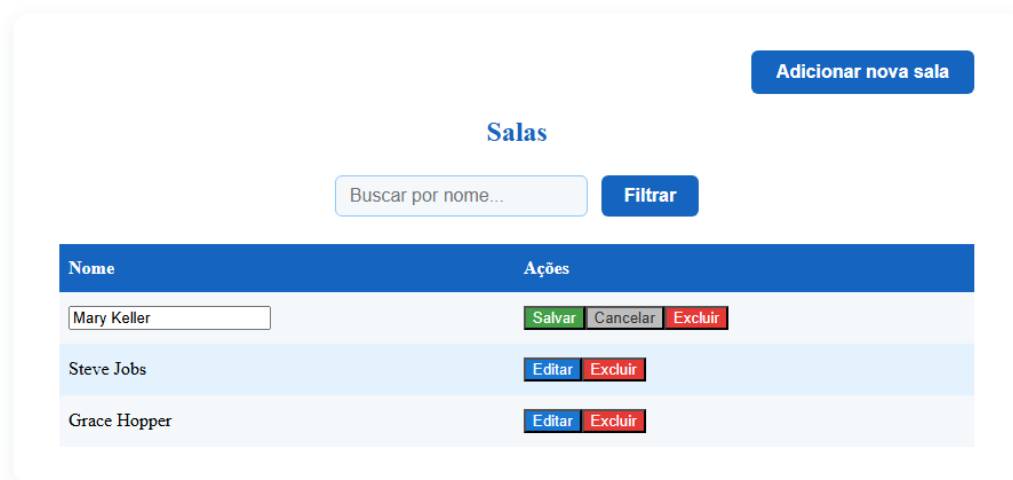
Turno
Matutino

Criar Turma

Fonte: Captura de tela do sistema, realizada pelo autor.

Já as funcionalidades relativas às salas também seguem a mesma lógica. A Figura 10 mostra a listagem de salas disponíveis no sistema.

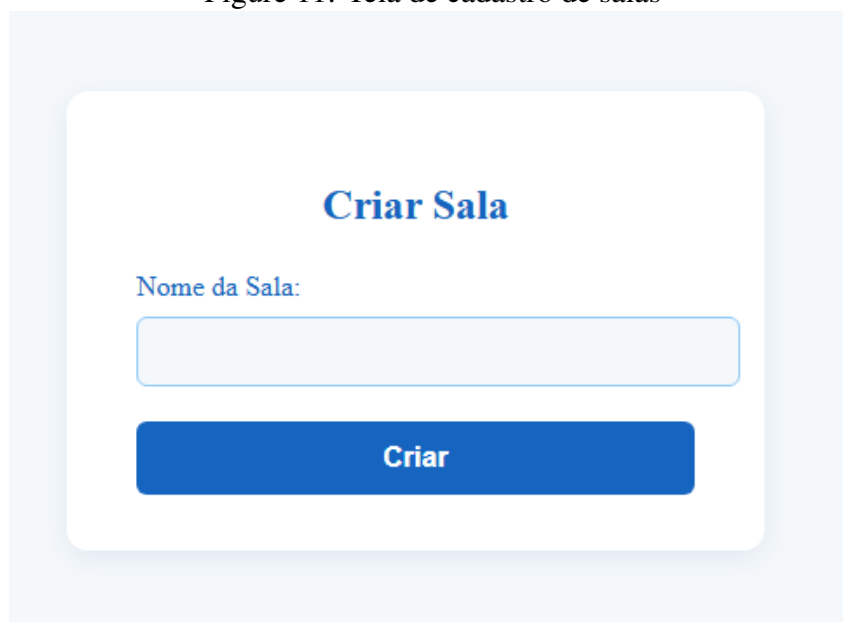
Figure 10: Tela de listagem de salas



Fonte: Captura de tela do sistema, realizada pelo autor.

O cadastro de salas é apresentado na Figura 11, permitindo a inclusão de novas salas.

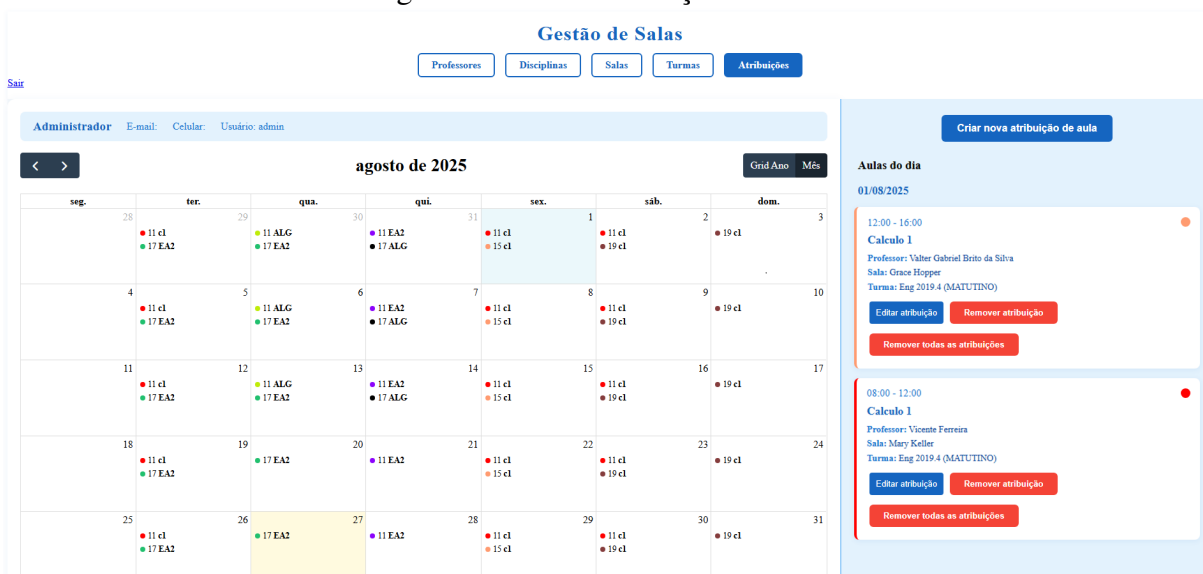
Figure 11: Tela de cadastro de salas



Fonte: Captura de tela do sistema, realizada pelo autor.

A Figura 12 mostra a visualização mensal das alocações, permitindo ao administrador acompanhar as reservas feitas em cada mês.

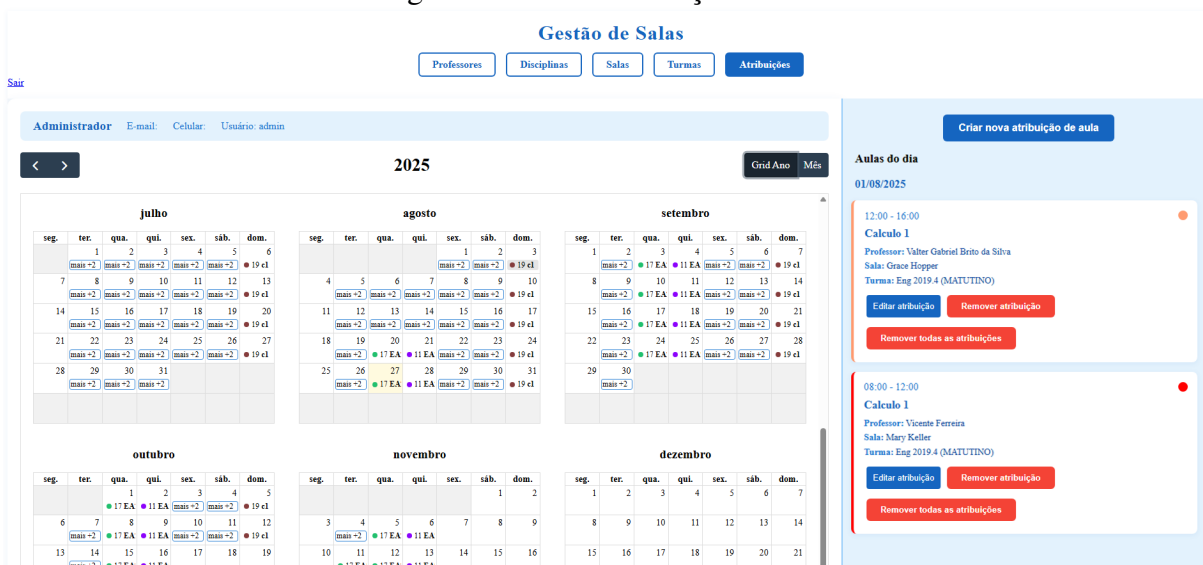
Figure 12: Tela de alocações mensais



Fonte: Captura de tela do sistema, realizada pelo autor.

A visão anual das alocações pode ser observada na Figura 13, facilitando o planejamento e o controle das reservas ao longo do ano.

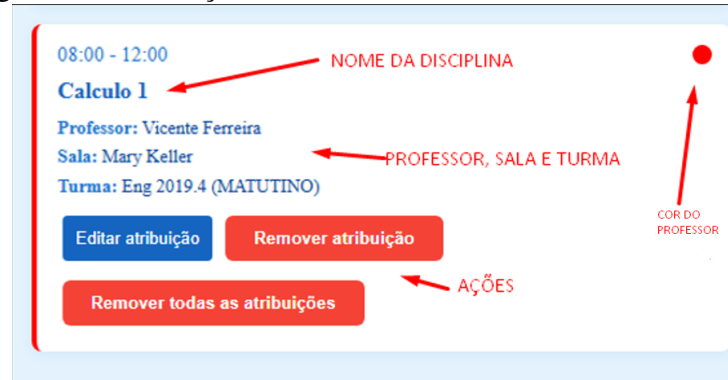
Figure 13: Tela de alocações anuais



Fonte: Captura de tela do sistema, realizada pelo autor.

A Figura 14 apresenta a interface do sistema após o usuário, no caso um professor, clicar em cima de um dia, numa aba ao lado é informado ao administrador as alocações já criadas para aquele dia, além de botões de ação para remover a alocação daquele dia, ou remover todas as alocações daquele professor.

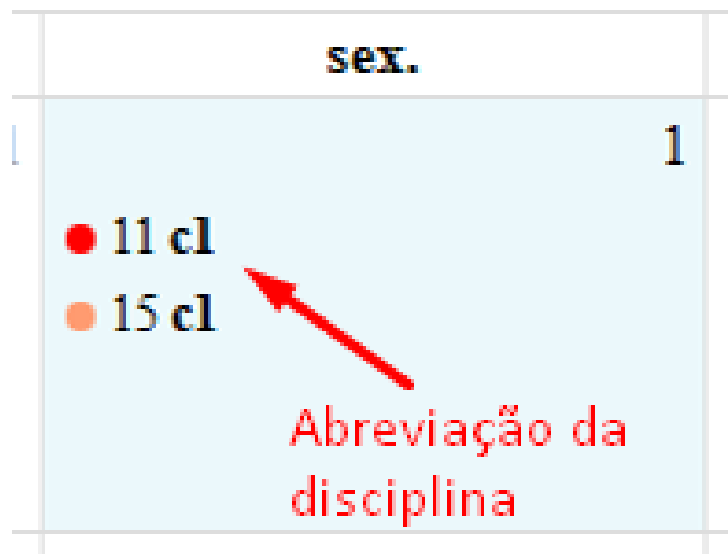
Figure 14: Nomeações dos itens ao clicar em um dia da semana



Fonte: Captura de tela do sistema, realizada pelo autor.

A Figura 15 mostra como as disciplinas são nomeadas de forma abreviada no calendário semanal. Cada célula exibe a sigla da disciplina associada ao horário, permitindo rápida identificação visual das alocações feitas para o dia selecionado.

Figure 15: Nomeação do item do dia da semana no calendário



Fonte: Captura de tela do sistema, realizada pelo autor.

A Figura 16 destaca que essa aba na direita mostra o botão de solicitar alocação quando o sistema é acessado por um professor.

Figure 16: Container de ações do professor

The image shows a screenshot of a web interface titled "Aulas do dia" (Classes of the day) for the date "01/08/2025". It displays two class entries for "Calculo 1". Each entry includes the time slot, professor name, room, and class name, along with a "Solicitar atribuição" (Request assignment) button. The first entry is for the time slot 12:00 - 16:00, taught by Professor Valter Gabriel Brito da Silva in room Grace Hopper. The second entry is for the time slot 08:00 - 12:00, taught by Professor Vicente Ferreira in room Mary Keller. Both entries are for the class "Eng 2019.4 (MATUTINO)".

Aulas do dia

01/08/2025

12:00 - 16:00

Calculo 1

Professor: Valter Gabriel Brito da Silva

Sala: Grace Hopper

Turma: Eng 2019.4 (MATUTINO)

Solicitar atribuição

08:00 - 12:00

Calculo 1

Professor: Vicente Ferreira

Sala: Mary Keller

Turma: Eng 2019.4 (MATUTINO)

Solicitar atribuição

Fonte: Captura de tela do sistema, realizada pelo autor.

A Figura 17 ilustra o processo de criação de uma nova alocação no sistema. Inicialmente, o administrador seleciona o professor, a sala, a turma e a disciplina correspondente. Em seguida, indica, por meio do calendário, os dias da semana em que a disciplina será ministrada, estabelecendo o dia inicial para o início das atividades. Para maior precisão e orientação durante o processo de inserção, foi disponibilizado um campo de texto abaixo do calendário.

Figure 17: Tela de cadastro de alocações

Atribuir Aulas

Atribuir Aulas

Professor: Turma:

Disciplina: Sala:

Horário de Início: Horário de Término:

Dias selecionados:

29/07/2025

< > **agosto de 2025**

seg.	ter.	qua.	qui.	sex.	sáb.	dom.
28	29	30	31	1	2	3
	● 08 c1 ● 14 EA2	● 08 ALG ● 14 EA2	● 08 EA2 ● 14 ALG	● 08 c1 ● 12 c1	● 08 c1 ● 16 c1	● 16 c1
4	5	6	7	8	9	10
	● 08 c1 ● 14 EA2	● 08 ALG ● 14 EA2	● 08 EA2 ● 14 ALG	● 08 c1 ● 12 c1	● 08 c1 ● 16 c1	● 16 c1
11	12	13	14	15	16	17
	● 08 c1 ● 14 EA2	● 08 ALG ● 14 EA2	● 08 EA2 ● 14 ALG	● 08 c1 ● 12 c1	● 08 c1 ● 16 c1	● 16 c1
18	19	20	21	22	23	24
	● 08 c1 ● 14 EA2	● 14 EA2	● 08 EA2	● 08 c1 ● 12 c1	● 08 c1 ● 16 c1	
25	26	27	28	29	30	31

Como selecionar os dias de aula:

1. **Selecione no calendário os dias da semana** em que esta disciplina ocorrerá (ex: todas as segundas e quartas)
2. **Inicie pela data de início da disciplina** - o primeiro dia selecionado deve ser a data oficial de início
3. **O sistema calculará automaticamente** os demais dias de aula com base na carga horária total da disciplina

Observação: Você pode selecionar múltiplos dias, desde que não exceda 7 dias semanais.

Fonte: Captura de tela do sistema, realizada pelo autor.

Considerando que cada disciplina possui uma carga horária previamente definida, o sistema, ao efetuar a alocação, calcula o número de encontros necessários dividindo a carga horária total pela duração de cada aula, parâmetro este também configurado durante a alocação.

Dessa forma, o sistema distribui automaticamente as aulas nos dias selecionados, de acordo com a periodicidade estabelecida. Por exemplo, caso uma disciplina de sessenta horas seja atribuída para ocorrer às segundas-feiras, com cada aula tendo duração de quatro horas, o sistema colocará essa disciplina para as próximas quinze segundas-feiras consecutivas. Caso a disciplina ocorra mais de uma vez por semana, o sistema entende isso e também divide as aulas de acordo.

Quando um docente solicita uma troca, o outro professor recebe automaticamente um e-mail detalhado sobre a solicitação, podendo aceitá-la ou recusá-la diretamente através dos botões disponibilizados na mensagem de email que o sistema envia (conforme ilustrado na Figura 18).

Figure 18: Email de solicitação de troca



Fonte: Captura de tela do email, realizada pelo autor.

Caso a solicitação seja aceita, o solicitante recebe uma confirmação por e-mail, informando a nova alocação da sala, bem como data e horário do evento (Figura 19).

Figure 19: Email de aceite de solicitação



Fonte: Captura de tela do email, realizada pelo autor.

Por outro lado, se a solicitação for recusada, o sistema envia uma notificação ao usuário. (Figura 20).

Figure 20: Email de recusa de solicitação



Fonte: Captura de tela do email, realizada pelo autor.

Essas funcionalidades promovem uma comunicação clara, rápida e automatizada entre os usuários, ampliando a eficiência e a organização na gestão de reservas e trocas de salas, e garantindo o controle dos recursos acadêmicos de forma segura e eficaz.

7. TESTES E RESULTADOS

Para a avaliação do sistema desenvolvido, foram realizados testes de usabilidade com usuários que não possuíam perfil de professor, tendo como objetivo validar as funcionalidades propostas e identificar possíveis dificuldades na interação com a interface. A escolha por usuários sem experiência docente foi intencional, visando obter uma perspectiva imparcial sobre o uso do sistema e garantir que a solução fosse intuitiva mesmo para usuários sem familiaridade prévia com sistemas de atribuição de salas.

A coleta de dados foi realizada através de questionários aplicados aos participantes ao término de cada sessão de teste. O instrumento de avaliação utilizou uma escala Likert de 5 pontos, onde o valor 1 representava o nível mais baixo de satisfação (indicando altos índices de erros e/ou dificuldades na execução das tarefas), enquanto o valor 5 correspondia ao nível mais alto de satisfação (indicando execução das tarefas com mínimos erros ou dificuldades). Esta

metodologia permitiu quantificar objetivamente a experiência dos usuários com cada funcionalidade do sistema.

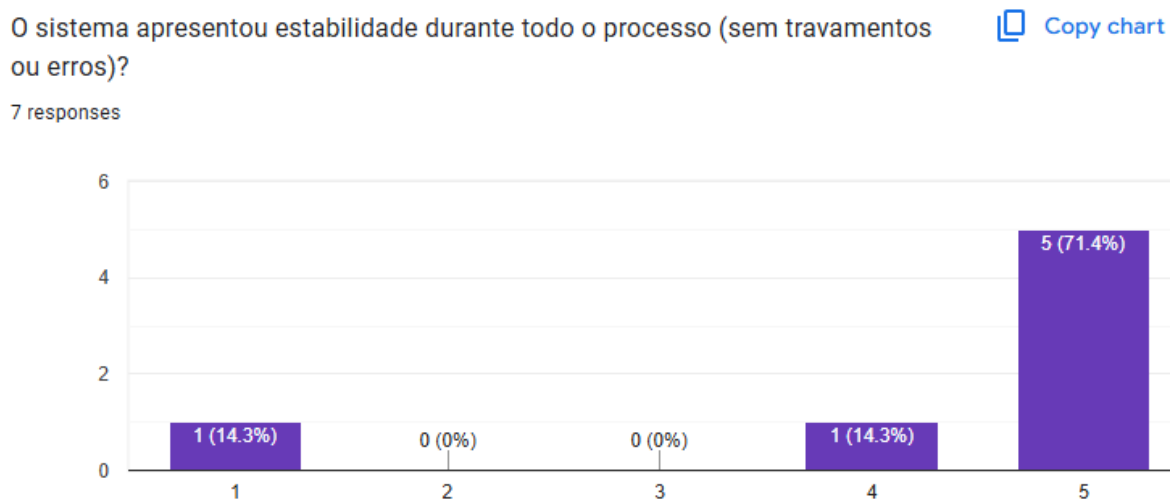
O gráfico apresentado a seguir demonstra o nível de avaliação obtido para cada uma das funcionalidades testadas, fornecendo uma visão consolidada do desempenho geral do sistema em termos de usabilidade e eficiência operacional.

A avaliação referente ao cadastro de professores no sistema demonstrou resultados plenamente satisfatórios. Todos os participantes dos testes atribuíram a nota máxima a esta funcionalidade, indicando que consideraram o processo de cadastro extremamente intuitivo, claro e fácil de utilizar. Não foram relatadas dúvidas, dificuldades ou sugestões de melhoria para essa etapa, o que evidencia que o sistema atendeu integralmente às expectativas dos usuários nesse aspecto.

No que diz respeito ao cadastro de turmas, os resultados dos testes evidenciaram plena satisfação dos usuários: 100% dos participantes consideraram essa funcionalidade fácil de utilizar e compreender, atribuindo sempre a nota máxima em sua avaliação. O processo foi visto como simples e direto, sem relatos de dificuldades ou apontamentos para melhorias, evidenciando a eficiência e a clareza da interface do sistema nesse aspecto.

A Figura 21 mostra o resultado da avaliação do cadastro de disciplinas. A funcionalidade foi aprovada pelos usuários, que destacaram a praticidade na definição da carga horária e vinculação às turmas

Figure 21: Avaliação da funcionalidade: cadastro de disciplinas



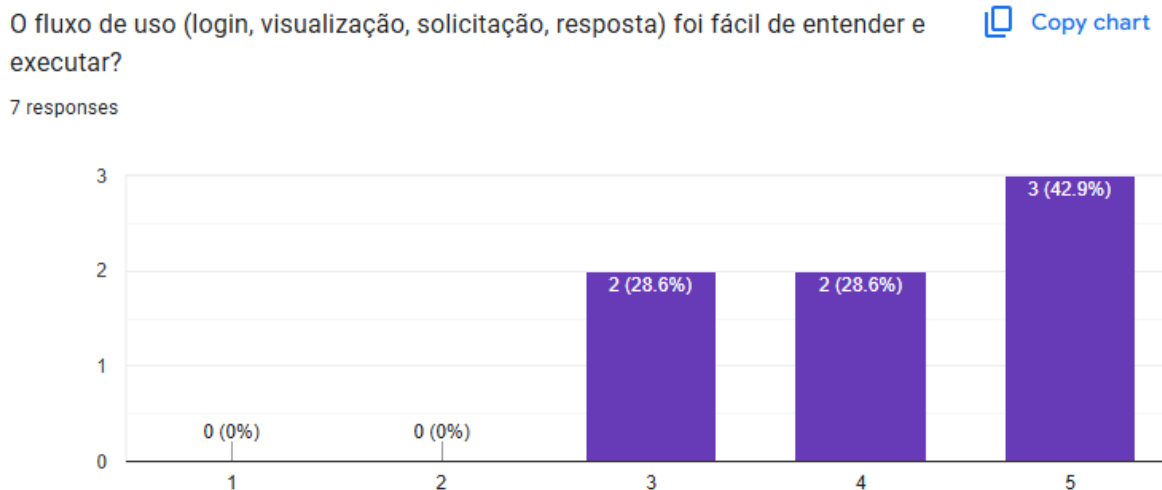
Fonte: Captura de tela da pergunta do Google Forms, realizada pelo autor.

A avaliação do cadastro de salas no sistema foi unânime entre os participantes dos testes. Todos os usuários relataram que a funcionalidade atendeu plenamente às expectativas, destacando a facilidade no processo de cadastro, clareza da interface e eficiência na execução das tarefas. Não foram identificadas dificuldades ou sugestões de melhoria para esta etapa, evidenciando

que o sistema proporcionou uma experiência positiva e satisfatória a todos os avaliadores.

Conforme mostrado na Figura 22, a avaliação das alocações de reservas indica bom entendimento dos usuários sobre como reservar e visualizar alocações de salas e horários.

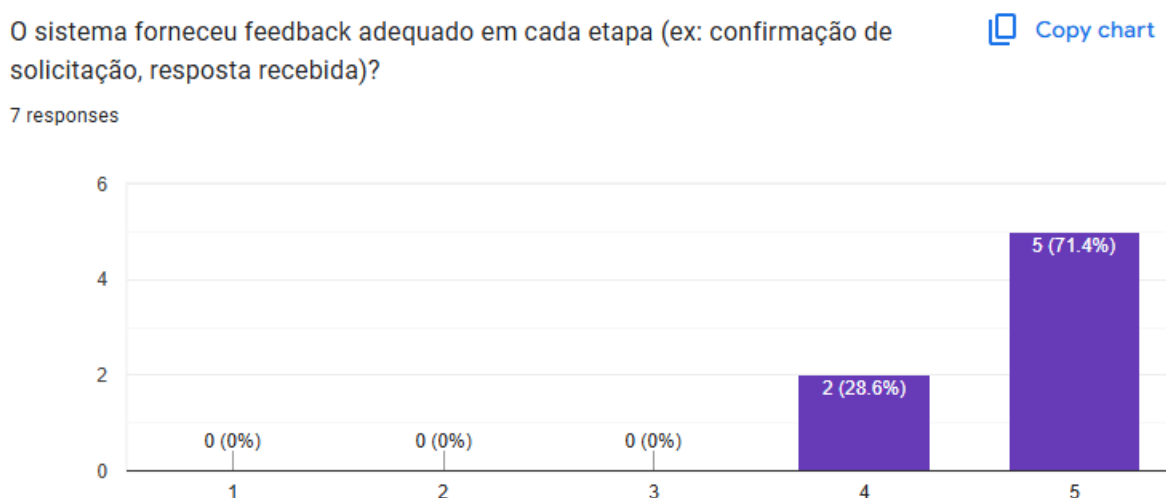
Figure 22: Avaliação da funcionalidade: alocações de reservas



Fonte: Captura de tela da pergunta do Google Forms, realizada pelo autor.

A Figura 23 trata da funcionalidade de troca de horários. Os testes demonstraram que os usuários tiveram facilidade em propor e aceitar trocas, aumentando a flexibilidade da alocação de salas.

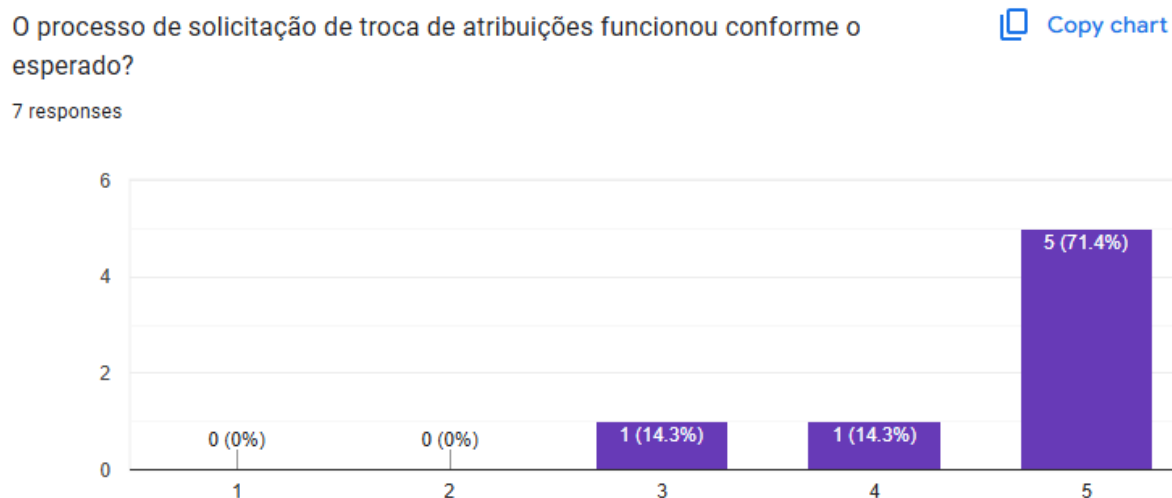
Figure 23: Avaliação da funcionalidade: troca de horários



Fonte: Captura de tela da pergunta do Google Forms, realizada pelo autor.

Por fim, a Figura 24 apresenta a avaliação geral do sistema, onde os usuários manifestaram uma impressão positiva sobre o funcionamento e a usabilidade da plataforma.

Figure 24: Avaliação geral do sistema



Fonte: Captura de tela da pergunta do Google Forms, realizada pelo autor.

7.1 Melhorias Futuras

Ficou claro com os testes que, apesar do objetivo final ter sido alcançado, há oportunidades significativas de aprimoramento do sistema. Com base nas respostas dos testadores, as principais sugestões para versões futuras incluem:

Orientação visual aprimorada: Alguns usuários tiveram dificuldade em perceber que a ação de solicitação de troca ocorre ao clicar diretamente sobre o evento. Sugere-se incluir indicações visuais claras ou instruções sobre como realizar essa ação, facilitando o entendimento para novos usuários.

Melhor usabilidade no login: Foi relatada dificuldade ao fazer login, pois o campo de e-mail é sensível a letras maiúsculas. Recomenda-se ajustar o sistema para aceitar o e-mail independentemente da capitalização, tornando o processo mais intuitivo.

Acompanhamento das solicitações: Os testadores destacaram a importância de visualizar as solicitações feitas e recebidas, com o status atualizado. A inclusão de uma aba ou seção dedicada para o acompanhamento dessas solicitações pode tornar o sistema mais transparente e organizado.

Notificações: A sugestão de adicionar uma aba de notificações no site, indicando de forma clara as solicitações recebidas e enviadas, pode trazer agilidade ao processo e garantir que o usuário esteja sempre informado sobre o andamento das trocas.

Criação de eventos em dias vagos: Outra funcionalidade sugerida é permitir que o usuário crie um evento em dias que estejam livres, sujeito à aprovação. Isso amplia a flexibilidade do sistema e pode atender a demandas específicas dos usuários.

8. CONCLUSÃO

A gestão eficiente dos ambientes acadêmicos representa um desafio recorrente nas instituições de ensino superior, especialmente diante da necessidade crescente por processos mais flexíveis, transparentes e ágeis. Este trabalho descreveu o desenvolvimento de um sistema web para otimizar a alocação dinâmica de salas de aula na Faculdade de Computação da UFPA, Campus Castanhal, buscando solucionar questões como burocracia excessiva, sobrecarga administrativa e falhas de comunicação comuns aos métodos tradicionais.

O sistema apresentado diferencia-se de abordagens já existentes justamente pelo foco não só no planejamento inicial, mas também na gestão das mudanças que ocorrem ao longo do semestre letivo. Por meio da plataforma, docentes podem negociar e efetuar trocas diretamente, com notificações automáticas e registro centralizado, o que promove maior autonomia e colaboração entre os envolvidos, além de reduzir retrabalho.

A escolha das tecnologias Angular, .NET e PostgreSQL proporcionou uma base escalável e alinhada com as melhores práticas de desenvolvimento, resultando em uma interface moderna e integrações seguras nas operações. O trabalho seguiu uma metodologia estruturada, envolvendo desde o levantamento de requisitos até a modelagem, implementação e validação do sistema.

Apesar dos testes terem sido realizados em ambiente local e controlado, sem a participação de usuários finais, os resultados indicam que a solução atende aos principais objetivos propostos. A validação interna permitiu ajustes e correções, aprimorando a usabilidade e a eficiência do sistema.

Como proposta para trabalhos futuros, recomenda-se a realização de testes de campo com professores e gestores administrativos, visando obter feedbacks reais que possam contribuir para o aprimoramento da ferramenta. Além disso, sugere-se considerar integrações com outros sistemas institucionais e a ampliação de funcionalidades, como relatórios de uso e análise preditiva de conflitos.

Por fim, o sistema desenvolvido representa uma modernização na gestão acadêmica da FA-COMP/UFPA, com uma proposta alinhada às demandas atuais do ensino superior. Ao automatizar processos burocráticos, centralizar informações e incentivar a colaboração, o projeto contribui para a melhoria contínua das condições de ensino e aprendizagem.

REFERÊNCIAS

- LAUDON, K. C.; LAUDON, J. P. *Sistemas de informação gerenciais*. 15. ed. São Paulo: Pearson, 2021.
- LEAL, R. S. Manutenção de código de sistema de alocação — UFC Quixadá. Quixadá: Universidade Federal do Ceará, 2023. Trabalho acadêmico. Disponível em: <https://repositorio.ufc.br/handle/riufc/73948/>. Acesso em: 12 jan. 2025.
- LIMA, A. S. Desenvolvimento de sistema para automação da alocação de salas.

Quixadá: Universidade Federal do Ceará, 2020. Trabalho acadêmico. Disponível em: <https://repositorio.ufc.br/handle/riufc/55042/>. Acesso em: 14 jan. 2025.

MICROSOFT. ASP.NET 8 documentation. Disponível em: <https://learn.microsoft.com/aspnet/core>. Acesso em: 7 jul. 2025.

OLIVEIRA, F. C. de. App online de reservas no CINTED/UFRGS. Porto Alegre: Universidade Federal do Rio Grande do Sul, 2018. Trabalho acadêmico. Disponível em: <https://lume.ufrgs.br/handle/10183/185080/>. Acesso em: 03 fev. 2025.

POSTGRESQL. The PostgreSQL Global Development Group. Disponível em: <https://www.postgresql.org/docs/>. Acesso em: 07 jul. 2025.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software*. 8. ed. Porto Alegre: AMGH, 2016.

TYPESCRIPT. TypeScript documentation. Disponível em: <https://www.typescriptlang.org/docs/>. Acesso em: 07 jul. 2025.

FAVATTO, N. C.; BOTH, J.; BELEM, I. C. Relação entre a vida pessoal e a vida profissional do professor de educação física. *Journal of Physical Education*, v. 34, e3445, 2023. DOI: 10.4025/jphyseduc.v34i1.3445.

FERREIRA, Gabriela de Souza; CORRÊA, Cláudia Prim; RISSI, Maurício. Gestão de processos em instituições de ensino superior: um estudo bibliométrico. Florianópolis: Universidade Federal de Santa Catarina, 2019. Disponível em: <https://repositorio.ufsc.br/handle/123456789/201887>. Acesso em: 18 jul. 2025.

TURINE, M. A. S.; MASIERO, P. C. Especificação de requisitos: uma introdução. São Carlos: Instituto de Ciências Matemáticas de São Carlos, Universidade de São Paulo, 1996. Disponível em: <https://repositorio.usp.br/item/000906321>. Acesso em: 28 ago. 2025.

GOOGLE. Angular – Introduction to Angular. Disponível em: <https://angular.io/guide/what-is-angular>. Acesso em: 04 ago. 2025.

BIERMAN, G.; ABECASSIS, D. TypeScript: Modern JavaScript with Static Typing. Microsoft, 2020. Disponível em: <https://www.typescriptlang.org>. Acesso em: 04 ago. 2025.

GITHUB. About Issues. 2024. Disponível em: <https://docs.github.com/pt>. Acesso em: 04 ago. 2025.

FIELDING, Roy Thomas. Architectural styles and the design of network-based software architectures. 2000. Tese (Doutorado) - University of California, Irvine, 2000.

SEBRAE. O que é usabilidade na web? A ideia é facilitar a vida do cliente. Sebrae – Mercado e Vendas, 07 maio 2023. Disponível em: (<https://sebrae.com.br/sites/PortalSebrae/o-que-e-usabilidade-na-web-a-ideia-e-facilitar-a-vida-do-cliente,5d94bd1a10597810VgnVCM1000001b00320aRCRD>). Acesso em: 7 set.

2025.

APPMASTER. Por que usar uma arquitetura modular no design de software? AppMaster – blog, 22 ago. 2023. Disponível em: <https://appmaster.io/pt/blog/por-que-usar-uma-arquitetura-modular-no-design-de-software?> . Acesso em: 7 set. 2025.