

# PROTÓTIPO DE SISTEMA INTELIGENTE PARA A ALOCAÇÃO EFICIENTE DE ASSENTOS EM SALAS DE AULA NA UNESC

Gustavo Ribeiro da Silva<sup>1</sup>, Sérgio Coral<sup>2</sup>

**Resumo:** O crescimento contínuo do número de estudantes nas universidades tem intensificado os desafios relacionados à alocação eficiente de assentos em salas de aula. Quando realizada manualmente, essa tarefa demanda tempo, é suscetível a erros e frequentemente resulta em uso inadequado dos espaços disponíveis. Para mitigar esse problema, este trabalho propõe o desenvolvimento de um sistema inteligente capaz de automatizar o processo de distribuição de assentos, utilizando algoritmos de decisão para considerar restrições de capacidade e horários, aliados a redes neurais artificiais que aprendem com dados históricos para aperfeiçoar as recomendações. A solução proposta apresenta uma interface intuitiva que facilita o trabalho dos gestores acadêmicos, tornando o processo de alocação mais rápido, preciso e adaptável às necessidades da instituição.

**Palavras-chave:** alocação de assentos; otimização; algoritmos de decisão; redes neurais; gestão acadêmica.

---

<sup>1</sup> Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), gutasvin@unesc.net

<sup>2</sup> Orientador. Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), sergio@unesc.net

**ABSTRACT:** The continuous growth in the number of university students has intensified the challenges associated with efficiently allocating seats in classrooms. When performed manually, this task is time-consuming, prone to errors, and often results in inadequate use of available spaces. To address this problem, this work proposes the development of an intelligent system capable of automating the seat distribution process, using decision algorithms to handle capacity and scheduling constraints, combined with artificial neural networks that learn from historical data to improve recommendations. The proposed solution includes an intuitive interface that supports academic administrators, making the allocation process faster, more accurate, and more adaptable to the institution's needs.

**Keywords:** seat allocation; optimization; decision trees; neural networks; academic management.

## 1 INTRODUÇÃO

Nos últimos anos, o crescimento contínuo do número de estudantes e de cursos nas instituições de ensino superior tem intensificado os desafios relacionados à gestão acadêmica. Entre as atividades administrativas mais complexas está a alocação eficiente de salas de aula, que envolve a análise de múltiplos fatores, como número de alunos, disponibilidade de horários, infraestrutura física e requisitos específicos de cada disciplina. Tradicionalmente, esse processo é realizado de forma manual, por meio de planilhas ou sistemas simples, o que torna a tarefa trabalhosa e suscetível a falhas humanas (Sales, 2015).

Com o aumento da diversidade de cursos e do volume de turmas, a administração acadêmica enfrenta dificuldades crescentes para garantir a melhor distribuição dos espaços disponíveis. A alocação inadequada de salas pode ocasionar ociosidade de ambientes, sobreposição de horários e desconforto para alunos e professores. Em muitos casos, o número de estudantes ultrapassa a capacidade das salas, resultando em situações de superlotação, nas quais nem todos conseguem um assento adequado, comprometendo o conforto e a qualidade do aprendizado. Além disso, o processo manual consome tempo dos gestores e frequentemente resulta em decisões baseadas em tentativas e erros, comprometendo a eficiência institucional (Couto; Paulino, 2019).

Com  $m$  professores e  $n$  disciplinas, o número de possibilidades torna-se imenso, e a probabilidade de erros aumenta conforme novas

variáveis são introduzidas (FILHO, 2016). Diante desse cenário, torna-se fundamental a busca por soluções tecnológicas que automatizem e otimizem esse processo, reduzindo a carga administrativa e melhorando a qualidade da alocação.

Diante desse cenário, e considerando que em 2024 a Universidade do Extremo Sul Catarinense (UNESC) registrou um aumento significativo no corpo discente, superando 16.000 alunos matriculados (SAVI, 2024), torna-se evidente a necessidade de sistemas automatizados e inteligentes para alocação de salas, utilizando algoritmos avançados e técnicas de inteligência artificial (IA).

A aplicação de sistemas inteligentes na gestão acadêmica tem se mostrado uma alternativa promissora, capaz de unir técnicas de otimização, algoritmos de decisão e inteligência artificial para apoiar gestores na tomada de decisões complexas. Ao empregar abordagens computacionais modernas, é possível considerar restrições múltiplas, preferências institucionais e dados históricos, obtendo resultados mais precisos e adaptáveis à realidade de cada instituição (Bayer; Eckhardt; Machado, 2011).

Diversas pesquisas exploram métodos para resolver o Problema de Alocação de Salas (PAS), como programação linear e modelos matemáticos. Por exemplo, Sales, Müller e Simonetto (2015) propuseram um modelo de programação linear que melhorou significativamente a taxa de ocupação das salas ao aplicá-lo ao Centro de Tecnologia da Universidade Federal de Santa Maria (UFSM), demonstrando a capacidade dessa abordagem em otimizar o uso do espaço físico.

Embora técnicas tradicionais ofereçam soluções consistentes para o Problema de Alocação de Salas (PAS), este trabalho propõe a utilização de métodos baseados em Inteligência Artificial, como árvores de decisão e redes neurais, que se destacam pela capacidade de adaptação e aprendizado. Essas abordagens permitem que o sistema identifique padrões, ajuste as alocações em tempo real e aperfeiçoe continuamente seus resultados a partir dos dados obtidos, tornando o processo mais preciso e eficiente.

Com base nessas técnicas, o trabalho apresenta uma solução que integra modelos de previsão, otimização automática e uma interface de gestão intuitiva, permitindo automatizar o processo de alocação de salas. A combinação desses elementos resulta em uma ferramenta prática e acessível, capaz de apoiar gestores acadêmicos na tomada de decisões,

reduzir falhas operacionais e modernizar a administração institucional.

Os objetivos específicos desta pesquisa consistem em: analisar o processo atual de alocação de salas em instituições de ensino e suas principais limitações; projetar uma interface de gestão intuitiva para o cadastro e controle das informações acadêmicas; implementar técnicas de Inteligência Artificial, como redes neurais e árvores de decisão, para otimizar o processo de alocação; integrar os módulos de interface e inteligência artificial em um sistema funcional; e, por fim, avaliar a eficiência da solução proposta por meio de testes e simulações que representem situações reais do ambiente universitário.

## **2 MATERIAIS E MÉTODOS**

Esta seção descreve os materiais e métodos utilizados no desenvolvimento do sistema inteligente para alocação de assentos em salas de aula, abordando as ferramentas, tecnologias e linguagens de programação empregadas, bem como as etapas de concepção, implementação e integração dos módulos que compõem a aplicação.

O desenvolvimento seguiu uma metodologia incremental, que permitiu a validação de cada componente de forma independente antes da integração final. Inicialmente, foram definidos os requisitos funcionais com base no problema de alocação de salas, seguidos pelo desenvolvimento da interface, implementação da camada de processamento e incorporação dos modelos de aprendizado de máquina. Por fim, cada etapa foi testada em cenários simulados para garantir o funcionamento adequado do sistema.

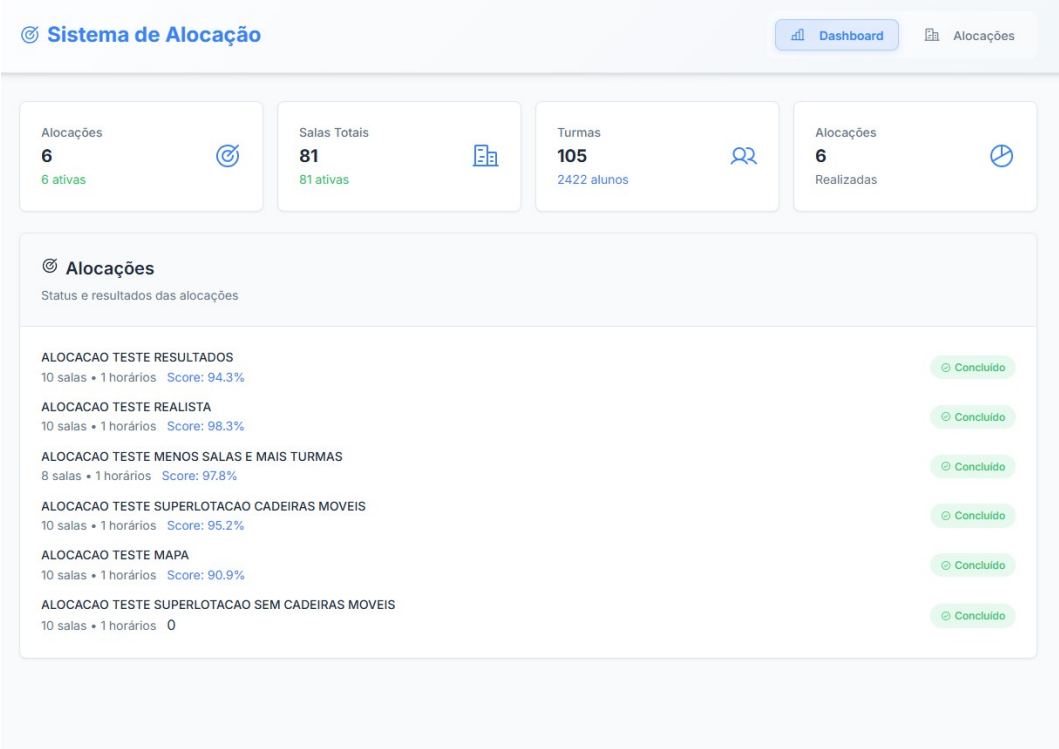
### **2.1 CAMADA DE INTERFACE**

A camada de interface do sistema foi desenvolvida utilizando o framework React com a linguagem TypeScript, escolhidos por oferecerem flexibilidade, modularidade e bom desempenho em aplicações web modernas. Essa camada tem como função principal permitir a interação do usuário com o sistema de forma simples e direta.

A interface foi desenvolvida de forma simples e objetiva, priorizando a usabilidade e a intuitividade durante o processo de cadastro e gerenciamento das alocações de turmas. O design minimalista busca reduzir a complexidade das interações, tornando o uso mais acessível ao profissional responsável. Como o protótipo foi inicialmente projetado para uma única instituição, não foi implementado um sistema de login, o que torna o acesso ainda mais direto e o início das alocações mais ágil.

A aplicação é composta por uma seção inicial de dashboard, que apresenta diversas informações relacionadas às alocações, turmas, alunos e indicadores quantitativos gerais. Esse painel da Figura 1 tem como objetivo fornecer uma visão ampla e consolidada do funcionamento do sistema, permitindo o acompanhamento em tempo real dos principais dados.

Figura 1 - Painel principal do sistema (dashboard).



Fonte: Do autor.

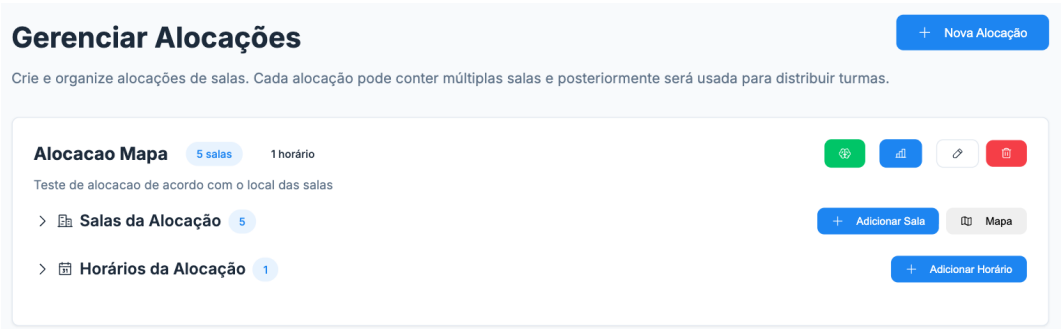
Além disso, a aplicação conta com uma página dedicada às alocações, na qual o usuário pode visualizar todas as distribuições já realizadas, bem como acessar recursos de cadastro e gerenciamento de turmas, salas e horários. Essa área foi desenvolvida com foco na clareza das informações e na praticidade das ações, garantindo que o processo de alocação ocorra de forma rápida e organizada.

Cada alocação é exibida em um bloco independente, destacando dados essenciais como quantidade de salas, horários e descrição do cenário e oferecendo atalhos diretos para funções como adicionar salas, definir horários ou visualizar a disposição no mapa. Essa estrutura favorece uma navegação mais fluida e intuitiva, tornando o gerenciamento das alocações mais eficiente e visualmente compreensível.

A Figura 2 apresenta a interface de gerenciamento das

alocações. Nela é possível visualizar os diferentes cenários de teste criados, como os de superlotação, diversidade e alocação realista, cada um composto por suas respectivas salas e horários. Essa tela permite ao usuário configurar, editar e acompanhar a execução das alocações de forma organizada e intuitiva.

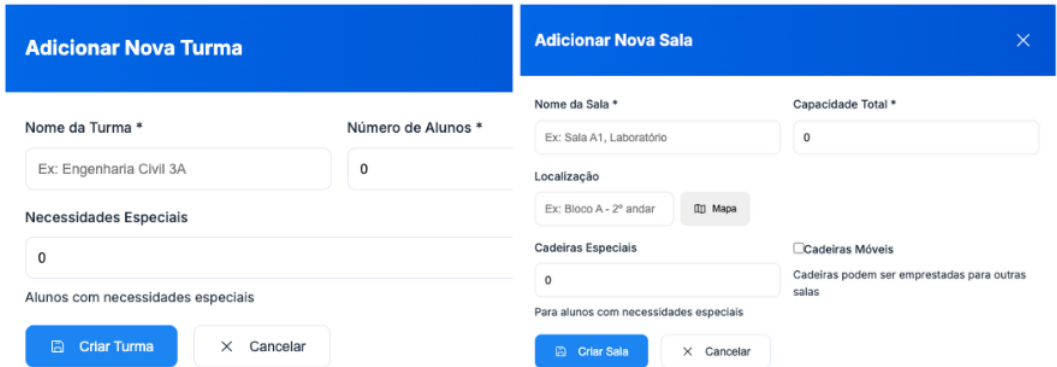
Figura 2 - Tela de listagem e gerenciamento de alocações.



Fonte: Do autor.

A Figura 3 ilustra as interfaces de cadastro de turmas e salas utilizadas pelo sistema. No módulo de turmas, o usuário informa o nome da disciplina, o número total de alunos e a quantidade de estudantes com necessidades especiais, permitindo que essas restrições sejam consideradas no processo de alocação. Já no módulo de salas, podem ser configuradas características estruturais como capacidade total, localização, disponibilidade de cadeiras especiais e a possibilidade de uso de cadeiras móveis, que podem ser remanejadas entre ambientes. Esses elementos compõem o conjunto de informações institucionais que alimentam o algoritmo, garantindo que as restrições reais das instalações acadêmicas sejam incorporadas ao processo de tomada de decisão.

Figura 3 - Interfaces de cadastro de turmas e salas.



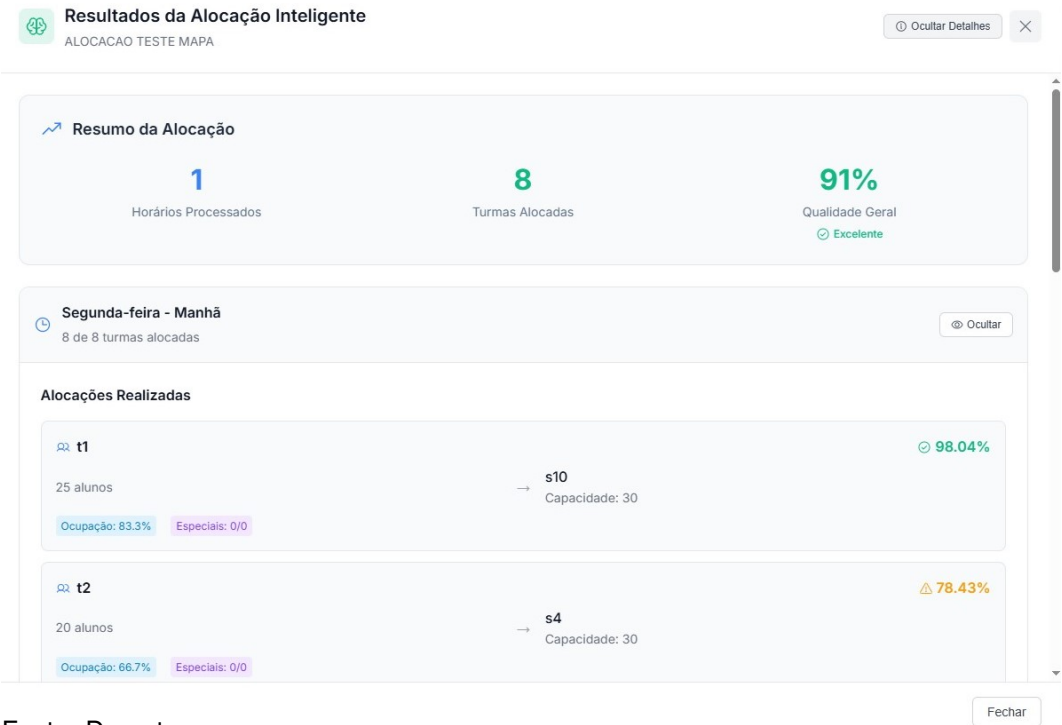
Fonte: Do autor.

Durante o uso, o usuário pode inserir os dados de forma

estruturada e, em seguida, acionar o processamento automático. O resultado é exibido diretamente na interface, indicando as combinações ideais entre turmas e salas conforme os critérios definidos.

A Figura 4 apresenta a janela de informações detalhadas gerada após a execução de cada cenário de alocação. Nela são exibidos indicadores como a porcentagem de qualidade geral, o número de turmas alocadas, a taxa de ocupação das salas e a correspondência entre capacidade e demanda. Essa visualização permite ao usuário analisar individualmente cada alocação realizada, facilitando a identificação de padrões, possíveis inconsistências e o desempenho geral do sistema em tempo real.

Figura 4 - Modal de informações detalhadas.



Fonte: Do autor.

## 2.2 CAMADA DE PROCESSAMENTO

A camada de processamento foi desenvolvida em Python, sendo responsável por realizar toda a análise e otimização das alocações de turmas nas salas. Ela se comunica com um servidor desenvolvido em Node.js, que atua como intermediário entre a interface e o módulo de inteligência artificial.

O Node.js é uma plataforma que permite executar código JavaScript no lado do servidor, sendo amplamente utilizada para aplicações web modernas devido à sua leveza e capacidade de lidar com múltiplas

requisições simultâneas. No projeto, ele foi utilizado para receber os dados do frontend, encaminhá-los ao script Python responsável pelo processamento e retornar ao usuário os resultados das alocações geradas pela IA.

O módulo de inteligência artificial foi desenvolvido em linguagem Python, utilizando técnicas de aprendizado de máquina e otimização para identificar as melhores combinações entre turmas e salas. Para isso, foram utilizadas bibliotecas amplamente consolidadas na área de ciência de dados:

- **scikit-learn**: responsável pela criação, treinamento e avaliação do modelo de aprendizado de máquina (Scikit-learn Developers, 2025);
- **NumPy**: usada para cálculos numéricos e manipulação de vetores e matrizes (NumPy Developers, 2025);
- **pandas**: facilita o tratamento e análise de dados em formato de tabelas (pandas Developers, 2025);
- **SciPy**: aplicada nas etapas de otimização e resolução de problemas matemáticos, como o método de alocação ótima (SciPy Developers, 2025);
- **NetworkX**: empregada para modelar e resolver problemas de fluxo e grafos, permitindo uma alocação mais eficiente (NetworkX Developers, 2025).

Essas bibliotecas, em conjunto, forneceram a base para o processamento de dados e a implementação dos modelos de aprendizado de máquina utilizados no projeto.

O algoritmo inicia com uma etapa de pré-processamento e validação dos dados, garantindo que todas as informações de turmas e salas estejam corretas e consistentes. Ele verifica, por exemplo, se as salas possuem capacidade válida, se as turmas têm número de alunos informado e se as cadeiras especiais estão devidamente cadastradas. Quando são informadas as coordenadas geográficas das salas, o sistema calcula a distância entre elas usando a fórmula de Haversine (um método que mede a distância entre dois pontos na superfície da Terra considerando a curvatura do planeta). Com essas distâncias, o algoritmo identifica blocos de salas próximas por meio do DBSCAN, uma técnica de agrupamento que detecta automaticamente regiões com alta concentração de pontos.

Na sequência, o sistema cria uma tabela de combinações possíveis entre turmas e salas, e para cada par são calculados diversos indicadores. Entre eles estão a taxa de ocupação (relação entre alunos e capacidade da sala), o déficit ou sobra de lugares, a compatibilidade de cadeiras especiais e o aproveitamento do espaço. Também é avaliado se a turma poderia ser acomodada com o auxílio de cadeiras móveis e, quando aplicável, a proximidade entre salas. A partir desses dados, o algoritmo gera um “rótulo heurístico” (um valor que indica se aquela combinação seria considerada adequada segundo as regras de alocação) servindo como base para o aprendizado do modelo.

Na etapa de aprendizado de máquina, o sistema utiliza uma Árvore de Decisão, um modelo que aprende regras a partir dos dados e faz previsões sobre quais combinações turma–sala são mais compatíveis. Esse modelo é treinado e ajustado automaticamente (com testes internos para melhorar o desempenho), e o resultado é uma pontuação de compatibilidade entre 0 e 1. Essa pontuação pode ser combinada com outros critérios, como o aproveitamento da sala e, opcionalmente, com o resultado de uma Rede Neural, que ajuda o sistema a reconhecer padrões mais complexos entre as variáveis.

Por fim, o algoritmo executa uma etapa de otimização, onde escolhe o melhor conjunto global de combinações entre turmas e salas. Nessa fase, são usados métodos matemáticos conhecidos como Algoritmo Húngaro (que encontra o melhor pareamento entre dois conjuntos) e Fluxo de Custo Mínimo (que busca a distribuição mais eficiente com o menor “custo” possível). Essa otimização também considera situações práticas, como empréstimo de cadeiras móveis entre salas, priorizando sempre o maior aproveitamento e o respeito às restrições de acessibilidade.

Após o processamento, o sistema gera um relatório com os resultados da alocação, incluindo métricas de desempenho e explicações sobre os critérios adotados. Quando uma turma não pode ser alocada, o algoritmo indica o motivo como falta de salas, excesso de alunos ou escassez de cadeiras especiais. Dessa forma, a camada de processamento não apenas realiza as alocações de forma inteligente, mas também justifica cada decisão tomada, garantindo transparência, auditabilidade e apoio à tomada de decisão pela instituição.

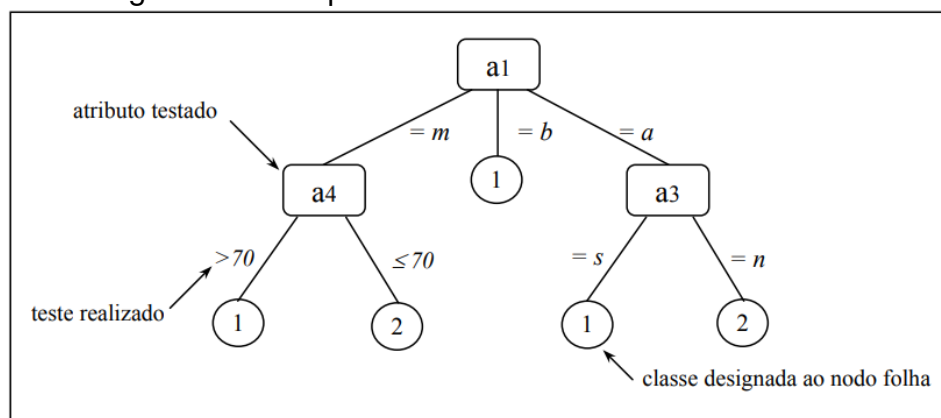
## 2.3 MODELOS DE APRENDIZADO E OTIMIZAÇÃO

O módulo de inteligência artificial do sistema combina modelos de aprendizado supervisionado com métodos de otimização combinatória para apoiar o processo de alocação de assentos em salas. Enquanto os modelos de aprendizado são responsáveis por avaliar o grau de compatibilidade entre turmas e salas, os métodos de otimização têm como função selecionar, dentre as possibilidades avaliadas, aquelas que formam a melhor distribuição global.

O primeiro modelo implementado foi a árvore de decisão, que realiza a análise das variáveis de entrada a partir de divisões hierárquicas nos dados. Esse tipo de modelo é amplamente utilizado em tarefas de classificação, pois permite representar de forma estruturada as condições que levam a determinados resultados (Garcia, 2003).

A Figura 5 ilustra de forma simplificada o funcionamento de uma árvore de decisão, na qual cada nó representa um atributo avaliado e cada ramificação indica uma condição ou teste. Esse modelo permite gerar regras de classificação claras, facilitando a interpretação das decisões geradas pelo sistema.

Figura 5 - Exemplo de estrutura de árvore de decisão.



Fonte: Garcia (2003).

No contexto do sistema desenvolvido, a árvore de decisão é empregada para gerar regras que relacionam variáveis como taxa de ocupação, disponibilidade de cadeiras especiais e capacidade das salas, classificando o nível de adequação entre cada turma e sala.

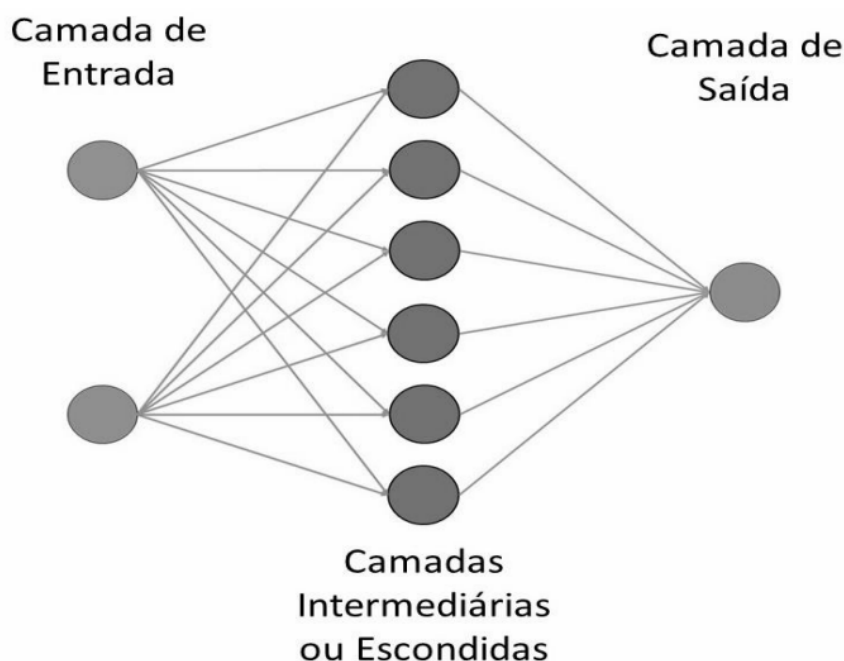
Após a etapa inicial de classificação realizada pela árvore de decisão, o sistema aplica redes neurais artificiais para tratar relações mais complexas entre as variáveis envolvidas.

Segundo Carvalho (2009), as redes neurais artificiais são

sistemas computacionais que simulam o modo como o cérebro humano processa informações, aprendendo a partir de exemplos e experiências. Cada rede é composta por diversas unidades de processamento interligadas, chamadas neurônios artificiais, que se comunicam de forma semelhante às conexões sinápticas do sistema nervoso biológico.

A Figura 6 mostra a estrutura básica de uma rede neural artificial, composta por uma camada de entrada, camadas intermediárias (ou escondidas) e uma camada de saída. Cada neurônio está conectado aos seguintes por meio de pesos ajustáveis, permitindo que a rede aprenda padrões nos dados e produza uma saída a partir das informações recebidas na entrada.

Figura 6 - Rede neural com camadas intermediárias.



Fonte: Lima et al. (2018)

Essas redes são compostas por camadas de unidades interconectadas que ajustam seus pesos conforme os exemplos analisados, sendo capazes de reconhecer padrões em dados que não seguem uma relação linear direta Haykin (2011). No projeto, elas são empregadas para refinar as pontuações geradas pela árvore de decisão, atribuindo um valor contínuo (probabilidade) de compatibilidade a cada par turma-sala.

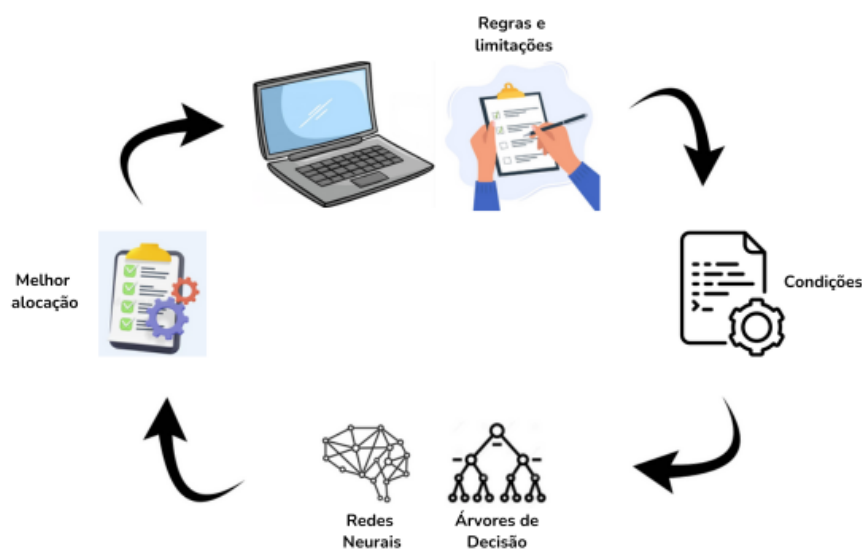
Com base nessas pontuações, o sistema executa a etapa de otimização, na qual são aplicados métodos matemáticos como o Algoritmo Húngaro e o Fluxo de Custo Mínimo para determinar a melhor combinação global entre turmas e salas. O Algoritmo Húngaro, conforme descrito

por Santos (2014), é um método de otimização baseado na manipulação de uma matriz-custo, no qual são realizadas operações sistemáticas de redução de linhas e colunas que preservam a solução ótima. A partir dessas transformações, o algoritmo identifica uma configuração de zeros que representa o pareamento de menor custo entre dois conjuntos, garantindo uma alocação eficiente das tarefas.

Dessa forma, enquanto os modelos de aprendizado supervisionado avaliam o grau de compatibilidade entre cada turma e sala, os algoritmos de otimização estruturam essas combinações de modo a produzir uma distribuição globalmente ótima. Esse processo assegura que a utilização dos espaços seja equilibrada, respeitando simultaneamente as restrições de capacidade, acessibilidade e recursos disponíveis, resultando em uma alocação coerente e alinhada às necessidades institucionais.

A Figura 7 apresenta o fluxo geral do sistema, no qual a interface recebe as regras e limitações definidas pelos gestores, que são convertidas em condições avaliadas pelos modelos de decisão e de redes neurais. Com base nessas informações, o sistema calcula a melhor combinação entre turmas e salas. Essa estrutura integrada entre interface, servidor intermediário e modelos de aprendizado de máquina automatiza o processo de alocação e serviu de base para os testes descritos na próxima seção.

Figura 7 - Fluxo geral do sistema de alocação inteligente de salas.



Fonte: Do autor.

### 3 RESULTADOS E DISCUSSÃO

Com o objetivo de verificar a efetividade da abordagem desenvolvida, o sistema foi testado em múltiplos cenários que simulam condições típicas e críticas de alocação acadêmica. A interpretação dos resultados permite avaliar não apenas o desempenho computacional da solução, mas também sua viabilidade prática enquanto ferramenta de apoio à tomada de decisão institucional. Os pontos analisados contemplam eficiência, adequação às restrições e qualidade das alocações geradas.

Para avaliar o desempenho do modelo proposto em diferentes condições operacionais, foram definidos seis cenários de teste que simulam situações reais e extremas de alocação de assentos em salas. Esses cenários consideram variações na quantidade de salas, capacidade dos ambientes, presença de cadeiras móveis, distâncias geográficas e limitações estruturais. O Quadro 1 apresenta a descrição de cada cenário, bem como os resultados obtidos em termos de eficiência, acurácia e comportamento do sistema diante das restrições impostas.

De forma geral, os testes indicaram que o protótipo se mostrou eficaz na distribuição de assentos ao atender simultaneamente critérios de capacidade das salas, acessibilidade, uso equilibrado dos ambientes e compatibilidade entre turmas e espaços. Considerando os seis cenários definidos para avaliação, variando capacidade, distâncias, presença de cadeiras móveis e limitações estruturais, o algoritmo manteve elevados índices de compatibilidade e eficiência, demonstrando estabilidade nos resultados obtidos. As evidências desses comportamentos podem ser observadas no Quadro 1, que reúne os valores de eficiência, acurácia e o desempenho do sistema diante das restrições impostas em cada cenário.

Quando submetido a cenários mais desafiadores como situações de superlotação, limitação de recursos ou necessidade de utilização de salas com cadeiras móveis, o sistema apresentou comportamento coerente com as regras definidas, priorizando combinações mais adequadas segundo os parâmetros de ocupação e proximidade. Esses resultados demonstram que a solução é capaz de lidar adequadamente com cenários reais de alocação, mantendo a qualidade das decisões mesmo sob condições adversas.

Um dos caminhos explorados na literatura é o uso de programação linear, como no estudo de Sales (2015), que empregou ferramentas como CPLEX e ZIMPL para gerar soluções ótimas. Embora

eficazes, essas técnicas demandam elevado processamento e não lidam bem com mudanças inesperadas, algo comum na organização de horários e salas. A solução desenvolvida neste trabalho, por outro lado, prioriza flexibilidade e adaptação em tempo real, mesmo que isso envolva abrir mão de um rigor matemático absoluto.

Outros estudos seguiram um caminho mais heurístico, buscando um equilíbrio entre qualidade da solução e tempo de execução. É o caso de Couto e Paulino (2019), que aplicaram Simulated Annealing combinado com algoritmos gulosos para melhorar a ocupação das salas. Apesar dos bons resultados, o método ainda exigia ajustes manuais frequentes. A abordagem aqui proposta se diferencia ao incorporar aprendizado de máquina, permitindo que o sistema aprenda padrões a partir de dados históricos e reduza a necessidade de intervenção humana.

Embora os modelos utilizados tenham apresentado desempenho satisfatório, trabalhos futuros podem investigar algoritmos mais avançados de aprendizado supervisionado, como Random Forest, Gradient Boosting ou modelos específicos para dados tabulares, que podem aumentar a precisão das estimativas de compatibilidade especialmente em cenários complexos ou com grande variabilidade entre as turmas.

Há também pesquisas voltadas ao uso de meta-heurísticas para reduzir o esforço computacional mantendo bons níveis de qualidade. Cirino (2016) exploraram técnicas como Busca Tabu, Algoritmos Genéticos e GRASP, alcançando soluções próximas do ótimo com execução mais rápida. No entanto, essas estratégias normalmente demandam calibrações específicas e não oferecem integração com interfaces ou rotinas administrativas.

Uma alternativa interessante para trabalhos futuros seria explorar resolutores de otimização mais sofisticados, como Programação Linear Inteira, OR-Tools, CPLEX ou Gurobi, que podem oferecer soluções mais rápidas ou mais próximas do ótimo em cenários de grande escala. A comparação sistemática entre esses métodos e os algoritmos empregados neste estudo poderia ampliar o entendimento sobre o custo computacional e a qualidade das soluções geradas.

De modo geral, percebe-se que os estudos anteriores tendem a privilegiar ou o rigor matemático ou o desempenho computacional, enquanto a proposta apresentada busca unir desempenho, adaptação e usabilidade, tornando-se mais adequada ao cotidiano de instituições que precisam reagir rapidamente a mudanças nas condições de alocação.

Embora exista uma variedade de pesquisas dedicadas à alocação acadêmica, grande parte delas aborda problemas em nível mais amplo, como a distribuição de salas ou a elaboração completa de horários. Já a alocação em nível de assentos envolve um detalhamento maior da capacidade e da organização interna dos espaços, o que exige considerações mais específicas. Esse contexto reforça a importância de métodos capazes de lidar com esse nível adicional de detalhe, justificando o desenvolvimento de abordagens voltadas diretamente para esse tipo de situação.

Esse cenário também contribui para a escolha de uma arquitetura híbrida neste trabalho. Enquanto métodos matemáticos tradicionais priorizam a busca por soluções ótimas, eles podem apresentar limitações quando há necessidade de ajustes frequentes ou reprocessamentos rápidos. A integração entre heurísticas, modelos preditivos e otimização automática oferece maior flexibilidade operacional, permitindo que o sistema responda de forma eficiente a alterações nas condições de alocação.

Além disso, muitos estudos concentram-se na formulação dos modelos ou na eficiência dos algoritmos, sem avançar para aspectos de suporte ao usuário ou integração com rotinas administrativas. Ao reunir visualização, tomada de decisão e recomendações em um único fluxo, o protótipo desenvolvido aproxima-se das necessidades reais de gestão acadêmica, oferecendo uma solução prática que vai além da análise puramente computacional.

Apesar das melhorias apresentadas, o sistema ainda depende da qualidade dos dados fornecidos e da atualização constante das informações institucionais. Mesmo assim, sua capacidade de adaptar-se a restrições como capacidade das salas, limitações estruturais, distâncias geográficas, presença de cadeiras móveis e critérios institucionais, aliada à obtenção de resultados consistentes, com índices de eficiência superiores a 94%, evidencia um avanço prático relevante em relação às abordagens tradicionais. Dessa forma, a solução proposta se destaca não apenas pela precisão computacional, mas também pela flexibilidade operacional, usabilidade e potencial de aplicação real na gestão de espaços acadêmicos.

Além disso, uma ampliação natural do sistema envolve integrar outros componentes da gestão acadêmica, como a geração automática de horários, a alocação de professores e a detecção antecipada de conflitos.

Essa evolução permitiria transformar o protótipo em um solucionador completo do problema de alocação acadêmica, incorporando múltiplas camadas de restrições e ampliando significativamente sua utilidade prática para instituições de médio e grande porte.

Para o treinamento da rede neural, foram utilizados 100 registros de dados contendo combinações previamente avaliadas entre turmas e salas. Cada registro incluía informações como capacidade da sala e número de alunos.

Esse conjunto de dados serviu como base para que o modelo aprendesse padrões fundamentais do processo de alocação, permitindo gerar previsões consistentes sobre a adequação entre ambientes e turmas mesmo em cenários não observados durante o treinamento.

O uso de um conjunto de dados mais amplo no treinamento das redes neurais é fundamental para que o modelo aprenda padrões mais complexos e representativos do ambiente acadêmico. Com um volume maior de exemplos reais, a rede tende a aprimorar sua capacidade preditiva, reduzindo erros de classificação e aumentando a precisão das estimativas de compatibilidade entre turmas e salas. Esse enriquecimento dos dados contribui diretamente para um desempenho mais robusto do sistema, especialmente em cenários caracterizados por maior variabilidade e complexidade estrutural.

Para permitir uma interpretação consistente dos resultados, três critérios principais foram utilizados na análise dos cenários:

**Eficiência da alocação:** representa a qualidade média das combinações escolhidas pelo sistema. Essa medida considera fatores como ocupação adequada da sala, compatibilidade de capacidades e respeito às restrições definidas. O valor final é apresentado em porcentagem, onde valores mais altos indicam alocações mais equilibradas e coerentes.

**Acurácia do modelo:** indica o desempenho do algoritmo de aprendizado de máquina ao prever se uma combinação turma-sala é apropriada. Essa métrica avalia o quanto o modelo acerta suas classificações, também expressa em porcentagem. Ela reflete a capacidade do sistema de distinguir corretamente boas e más combinações antes da otimização.

**Acurácia espacial** (quando aplicável): utilizada nos cenários que envolvem distância geográfica entre salas. Mede se as alocações foram realizadas privilegiando salas próximas e agrupamentos coerentes,

garantindo menor deslocamento e melhor organização física.

Essas métricas servem como base para a análise dos resultados apresentados a seguir, permitindo avaliar de forma consistente o desempenho do sistema em diferentes condições operacionais. Além de possibilitar comparações diretas entre os cenários, elas também ajudam a identificar quais componentes contribuíram mais para a qualidade final das alocações. O Quadro 1 apresenta os cenários testados, acompanhados de suas descrições e dos resultados observados, permitindo avaliar o desempenho do sistema em diferentes condições.

Quadro 1 - Cenários de teste do sistema de alocação inteligente

| Cenário                               | Descrição                                                                                                                                                                                                                                    | Resultados Observados                                                                                                                                                                                                                                   |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>C1 Alocação<br/>Teste Realista</b> | Cenário com 10 salas de capacidades variadas (20 a 60 lugares) e 10 turmas distribuídas em um único horário. O objetivo foi simular uma alocação próxima à realidade institucional, contemplando salas com recursos móveis e acessibilidade. | O sistema alocou 10/10 turmas (100%), com eficiência global de 98% e acurácia de 88,89%. Todas as alocações respeitaram critérios de capacidade e acessibilidade, sem conflitos ou inconsistências. O desempenho geral foi classificado como excelente. |
| <b>C2 Alocação<br/>Teste Mapa</b>     | Cenário com 10 salas distribuídas em diferentes locais do campus, simulando variação geográfica entre blocos e endereços. O objetivo foi avaliar o comportamento do sistema em alocações espacialmente dispersas.                            | O algoritmo alocou 10/10 turmas (100%), com eficiência média de 91% e acurácia espacial de 100%. As turmas foram corretamente associadas a salas próximas e compatíveis, mantendo excelente qualidade geral e coerência na distribuição geográfica.     |

|                                                                           |                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>C3 Alocação<br/>Teste<br/>Resultados</b>                               | Configuração com 10 salas de diferentes capacidades (15 a 60 lugares) e turmas com variação de tamanho e necessidades especiais. O cenário buscou testar a precisão do modelo em múltiplos parâmetros simultaneamente.  | O sistema alocou 10/10 turmas (100%), com eficiência média de 94% e acurácia geral de 88,89%. As alocações atenderam adequadamente às restrições de capacidade e acessibilidade, apresentando ótimo equilíbrio entre uso de recursos e correspondência de requisitos.                                                                         |
| <b>C4 Alocação<br/>Teste<br/>Superlotação<br/>Cadeiras<br/>Móveis</b>     | Cenário de alta demanda, com 10 salas pequenas e presença de cadeiras móveis para compensar limitações de espaço. O objetivo foi analisar a adaptação do sistema em situações de superlotação parcial.                  | O sistema conseguiu alocar 5/10 turmas (50%), alcançando eficiência média de 95% entre as alocações realizadas. As salas móveis ampliaram a capacidade efetiva em até 8 assentos, permitindo alocações ajustadas. As demais turmas não foram alocadas por restrições estruturais, demonstrando coerência na validação das limitações físicas. |
| <b>C5 Alocação<br/>Teste<br/>Superlotação<br/>Sem Cadeiras<br/>Móveis</b> | Cenário com 10 salas fixas e capacidades reduzidas, testando a resposta do sistema diante de turmas maiores do que o limite das salas. O objetivo foi verificar a robustez da validação estrutural sem recursos móveis. | Nenhuma turma pôde ser alocada (0/10), resultando em 0% de eficiência e acurácia nula. O sistema identificou corretamente a inviabilidade de alocação, encerrando o processo sem inconsistências, o que demonstra sua robustez frente a limitações físicas rígidas.                                                                           |
| <b>C6 Alocação<br/>Teste Menos<br/>Salas e Mais<br/>Turmas</b>            | Cenário com 8 salas disponíveis e 10 turmas, simulando restrição real de espaço e exigências de acessibilidade. O objetivo foi avaliar a priorização e otimização das alocações sob limitação estrutural.               | O sistema alocou 8/10 turmas (80%), com eficiência média de 98% e acurácia geral de 90,91%. As duas turmas restantes não foram alocadas por falta de espaço compatível.                                                                                                                                                                       |

Fonte: Do autor.

A análise dos resultados demonstra que o sistema manteve desempenho consistente em todos os cenários, com eficiência média

superior a 94% nos casos de alocação viável. Os menores índices observados ocorreram em situações de superlotação extrema, onde as restrições físicas impossibilitaram a distribuição completa das turmas. Ainda assim, o algoritmo apresentou comportamento estável, encerrando corretamente o processo sem gerar falsos positivos. Esses resultados reforçam a confiabilidade do modelo mesmo sob condições adversas.

Durante a análise de desempenho, verificou-se que o tempo de processamento dos cenários variou entre 3 e 9 segundos, dependendo da complexidade das restrições envolvidas. Esse intervalo demonstra que o sistema mantém um desempenho eficiente mesmo em condições de maior carga computacional. O cenário "Alocação Teste Superlotação Cadeiras Móveis" apresentou o maior tempo registrado, com aproximadamente 8,50 segundos, devido à combinação de múltiplas variáveis, como excesso de turmas, cadeiras móveis e requisitos de acessibilidade. Apesar disso, o algoritmo concluiu todas as execuções de forma estável e previsível, confirmando sua adequação para aplicações práticas em ambientes acadêmicos.

A Tabela 1 apresenta um resumo dos tempos de processamento registrados em cada cenário, bem como a relação entre salas e turmas alocadas. A sigla "A." indica Alocação, utilizada para abreviar os nomes dos cenários.

Tabela 1 - Tempo de processamento e relação salas–turmas por cenário

| Cenário                                   | Tempo de Processamento (segundos) | Salas/Turmas |
|-------------------------------------------|-----------------------------------|--------------|
| A. Teste Mapa                             | 7,75 s                            | 10/10        |
| A. Teste Menos Salas e Mais Turmas        | 7,88 s                            | 8/10         |
| A. Teste Realista                         | 7,70 s                            | 10/10        |
| A. Teste Superlotação Cadeiras Móveis     | 8,50 s                            | 10/10        |
| A. Teste Superlotação Sem Cadeiras Móveis | 3,72 s                            | 10/10        |
| A. Teste Resultados                       | 8,37 s                            | 10/10        |

Fonte: Do autor.

De modo geral, os resultados obtidos confirmam a eficiência e a estabilidade do sistema proposto, demonstrando que a integração entre

aprendizado de máquina e métodos de otimização pode resolver de forma eficaz o problema de alocação de salas em ambientes acadêmicos. A análise dos cenários revelou que o modelo mantém bom desempenho mesmo diante de restrições severas, validando sua capacidade de adaptação e confiabilidade. Assim, os experimentos realizados comprovam que o sistema não apenas automatiza o processo de alocação, mas também o aprimora, oferecendo suporte concreto à tomada de decisão e contribuindo para uma gestão mais inteligente dos espaços institucionais.

#### **4 CONCLUSÃO**

O desenvolvimento do sistema inteligente evidenciou que técnicas de inteligência artificial podem aprimorar a gestão acadêmica. A união entre interface, camada de processamento e modelos de decisão permitiu gerar alocações coerentes, estáveis e adequadas às demandas institucionais, mesmo diante de cenários com restrições complexas.

Durante o desenvolvimento, algumas dificuldades se destacaram. A padronização dos dados e a validação dos atributos exigiram cuidado constante, pois erros mínimos poderiam comprometer todo o processamento. Também foi desafiador definir pesos e critérios que representassem corretamente as regras institucionais, exigindo inúmeros testes até que o comportamento do modelo se tornasse estável.

Além desses aspectos conceituais, houve desafios técnicos na implementação do código. Foi necessário estudar exemplos, comparar abordagens e realizar experimentos sucessivos até chegar a uma estrutura confiável. A geração das features, o cálculo das distâncias e a integração dos métodos de otimização demandaram pesquisa e ajustes contínuos para que todas as etapas do sistema funcionassem de forma coerente.

Os resultados obtidos reforçam as vantagens da aplicação de IA nesse tipo de problema. A combinação entre árvore de decisão, rede neural e métodos de otimização permitiu identificar padrões, priorizar as melhores soluções e alcançar índices de eficiência superiores a 94% nos cenários viáveis. Isso representa um avanço claro em relação à alocação manual, reduzindo tempo, esforço e o risco de decisões inconsistentes.

Apesar dos bons resultados, o sistema permanece dependente da qualidade dos dados fornecidos. A atualização correta das informações institucionais continua sendo um fator essencial para garantir previsões confiáveis e alocações adequadas.

Por fim, recomenda-se ampliar o conjunto de dados utilizado

no treinamento das redes neurais e avaliar algoritmos mais avançados de aprendizado e otimização. A integração com módulos adicionais, como geração de horários e alocação de professores, pode transformar a solução em uma ferramenta ainda mais completa e eficaz para instituições de médio e grande porte.

## REFERÊNCIAS

BAYER, F. M.; ECKHARDT, M.; MACHADO, R. **Automação de Sistemas**. [S.l.]: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria; Escola Técnica Aberta do Brasil. 2011.

CARVALHO, A. P. d. L. F. d. **Introdução às Redes Neurais Artificiais**. São Carlos, SP: [s.n.], 2009. Disponível em: <https://sites.icmc.usp.br/andre/research/neural/>. Acesso em: 15 out. 2025.

CIRINO, R. B. Z. **Abordagens de Solução para o Problema de Alocação de Aulas a Salas**. [S.l.]: Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação. 2016.

COUTO, D. T. d.; PAULINO, F. C. **Estudo do problema de alocação de salas aplicado ao IFG – Câmpus Goiânia**. [S.l.]: Instituto Federal de Educação, Ciência e Tecnologia de Goiás. 2019.

GARCIA, S. C. **O Uso de Árvores de Decisão na Descoberta de Conhecimento na Área da Saúde**. [S.l.]: Universidade Federal do Rio Grande do Sul. 2003.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. [S.l.]: Bookman. 2011.

LIMA, L. A. de et al. **Um Estudo em Rede de Análise Paraconsistente**. [S.l.]: Logistics Challenges in the New Economy. 2018.

NetworkX Developers. **NetworkX Documentation**. 2025. Disponível em: [<https://networkx.org/>](https://networkx.org/). Acesso em: 2 dez. 2025.

NumPy Developers. **NumPy Documentation**. 2025. Disponível em: [<https://numpy.org/>](https://numpy.org/). Acesso em: 2 dez. 2025.

pandas Developers. **pandas Documentation**. 2025. Disponível em: [<https://pandas.pydata.org/>](https://pandas.pydata.org/). Acesso em: 2 dez. 2025.

SALES, E. d. S.; MÜLLER, F. M.; SIMONETTO, E. d. O. **Solução do problema de alocação de salas utilizando um modelo matemático multi-índice**. Dissertação (Mestrado) — Universidade Federal de Santa Maria, Santa Maria, 2015. Disponível em: <https://cdsid.org.br/sbpo2015/wp-content/uploads/2015/08/142757.pdf>. Acesso em: 15 set. 2024.

SALES, E. dos S. **Problema de alocação de salas e a otimização dos espaços no Centro de Tecnologia da UFSM**. [S.l.]: Universidade Federal de Santa Maria, Centro de Ciências Sociais e Humanas, Programa de Pós-Graduação em Administração. 2015.

SANTOS, C. E. S. dos. **Método Húngaro e Aplicações**. Dissertação (Mestrado) — Universidade Federal Rural de Pernambuco, Recife, ago. 2014.

Scikit-learn Developers. **Scikit-learn Documentation**. 2025. Disponível em: <<https://scikit-learn.org/>>. Acesso em: 2 dez. 2025.

SciPy Developers. **SciPy Documentation**. 2025. Disponível em: <<https://scipy.org/>>. Acesso em: 2 dez. 2025.