

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
MINAS GERAIS - *CAMPUS* IBIRITÉ  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

João Victor Ferreira de Castro Pires da Cunha

**DESENVOLVIMENTO DE UM SISTEMA EDUCACIONAL  
INTELIGENTE INTEGRADO COM LANGCHAIN**

Ibirité -MG  
2026

JOÃO VICTOR FERREIRA DE CASTRO PIRES DA CUNHA

**DESENVOLVIMENTO DE UM SISTEMA EDUCACIONAL  
INTELIGENTE INTEGRADO COM LANGCHAIN**

Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus* Ibirité para a obtenção do título de Engenheiro de Controle e Automação.

**Orientador:** Prof. Dr. Carlos Dias da Silva Junior

Ibirité - MG  
2026

**João Victor Ferreira de Castro Pires da Cunha**

**DESENVOLVIMENTO DE UM SISTEMA EDUCACIONAL INTELIGENTE  
INTEGRADO COM LANGCHAIN**

Trabalho de conclusão de curso apresentado ao Cursode Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Ibirité para a obtenção do título de Engenheiro de Controle e Automação.

Aprovado em 23 de janeiro de 2026 pela banca examinadora



Documento assinado eletronicamente por **Livia de Maria Calado Machado Soares, Professora EBTT**, em 12/03/2026, às 09:31, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Amanda do Carmo Silva, Professora**, em 12/03/2026, às 11:40, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Carlos Dias da Silva Junior, Professor EBTT**, em 12/03/2026, às 11:49, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **2647596** e o código CRC **F2707932**.

---

C972d Cunha, João Victor Ferreira de Castro Pires da

Desenvolvimento de um sistema educacional inteligente integrado com LangChain / João Victor Ferreira de Castro Pires da Cunha. – 2026.

46f.;il.

Orientador: Carlos Dias da Silva Júnior.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Instituto Federal de Minas Gerais. *Campus Ibirité*, 2026.

1. Inteligência artificial. 2. Sistemas educacionais 3. Aplicativo Móvel. 4. LangChain. 5. OpenAI. I. Silva Júnior, Carlos Dias da. II. Instituto Federal de Minas Gerais. *Campus Ibirité*. III. Título.

CDD 006.3

---

Catálogo: Rejane Valéria Santos - CRB-6/2907

Dedico este trabalho ao meu avô, cujo exemplo de caráter, esforço e dignidade sempre guiou meus passos.

## **AGRADECIMENTOS**

Chegar até aqui não foi um processo linear. Foram noites longas, dúvidas técnicas, refatorações, testes que falharam e aprendizados que só vieram depois de muitos erros.

Agradeço à minha família, que acreditou em mim mesmo quando eu estava exausto. Ao meu orientador, que não apenas orientou este trabalho, mas me ensinou a pensar de forma crítica e estruturada.

Aos professores do IFMG, que ajudaram a construir a base que tornou este projeto possível.

E aos amigos que estiveram presentes nos momentos de pausa, porque equilíbrio também faz parte da engenharia.

“Education is not preparation for life; education is life it self.”

John Dewey

## RESUMO

O avanço da inteligência artificial (IA) tem transformado o campo educacional, promovendo novas formas de análise e personalização da aprendizagem. Este trabalho propõe o desenvolvimento de um sistema educacional inteligente baseado no framework LangChain, integrado a modelos de linguagem da OpenAI, para gerar uma visão analítica detalhada do desempenho e perfil dos alunos. O sistema é construído sobre o framework Django, utilizando PostgreSQL como banco de dados e uma infraestrutura composta por Gunicorn, Nginx, RabbitMQ e Redis, garantindo escalabilidade, desempenho e confiabilidade. A solução visa automatizar a análise de dados acadêmicos, produzindo relatórios interpretáveis que apoiam decisões pedagógicas e administrativas. Além do impacto técnico, o projeto busca contribuir socialmente para a democratização do acesso a tecnologias inteligentes na educação, apresentando uma alternativa moderna, eficiente e de baixo custo para instituições públicas e privadas.

**Palavras-chave:** Inteligência artificial. LangChain. Django. OpenAI. Sistemas educacionais inteligentes.

## ABSTRACT

The advancement of artificial intelligence (AI) has transformed the educational field, fostering new approaches to analysis and personalized learning. This work proposes the development of an intelligent educational system based on the LangChain framework, integrated with OpenAI language models, to generate a detailed analytical overview of student performance and profiles. The system is built using the Django framework, with PostgreSQL as the database and an infrastructure that includes Gunicorn, Nginx, RabbitMQ, and Redis, ensuring scalability, performance, and reliability. The solution aims to automate academic data analysis, producing interpretable reports that support pedagogical and administrative decision-making. Beyond its technical impact, the project also seeks to contribute socially to the democratization of access to intelligent technologies in education, presenting a modern, efficient, and low-cost alternative for public and private institutions.

**Keywords:** Artificial intelligence. LangChain. Django. OpenAI. Intelligent educational systems.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura geral do sistema proposto . . . . .	25
Figura 2 – Tela de <i>login</i> do sistema. . . . .	35
Figura 3 – Painel de gerenciamento de matérias e matrícula de alunos. . . . .	36
Figura 4 – Tela de cadastro e listagem de aulas associadas a uma matéria. . . . .	36
Figura 5 – Tela de cadastro e gerenciamento de tarefas associadas às aulas. . . . .	37
Figura 6 – Tela de cadastro de provas com vinculação a aulas e tarefas. . . . .	37
Figura 7 – Tela de lançamento e gerenciamento de notas por prova e aluno. . . . .	38
Figura 8 – Painel do aluno com acesso às funcionalidades e ao assistente inteligente. . .	39
Figura 9 – Consulta de tarefas pendentes realizada por meio do <i>chatbot</i> . . . . .	39
Figura 10 – Consulta da próxima prova realizada por meio do <i>chatbot</i> . . . . .	40
Figura 11 – Recomendação de matéria para estudo gerada pelo assistente inteligente. . .	40

## **LISTA DE TABELAS**

Tabela 1 – Comparação entre LMS tradicional (Moodle) e o sistema proposto . . . . .	24
---	----

## **LISTA DE ABREVIATURAS E SIGLAS**

AI-LMS	Artificial Intelligence Learning Management System (Sistema de Gestão de Aprendizagem com Inteligência Artificial)
API	Application Programming Interface (Interface de Programação de Aplicações)
CRUD	Create, Read, Update, Delete (Criar, Ler, Atualizar e Remover/Apagar)
DRF	Django REST Framework
HTTP	HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)
IA	Inteligência Artificial
IFMG	Instituto Federal de Minas Gerais
JWT	JSON Web Token (Token Web JSON)
JSON	JavaScript Object Notation (Notação de Objetos JavaScript)
LLM	Large Language Model (Modelo de Linguagem de Grande Escala)
LMS	Learning Management System (Sistema de Gestão de Aprendizagem)
PLN	Processamento de Linguagem Natural
REST	Representational State Transfer (Transferência de Estado Representacional)
SQL	Structured Query Language (Linguagem de Consulta Estruturada)
SQTL	Software Quality and Teaching-Learning (Qualidade de Software e Ensino-Aprendizagem)

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>14</b>
<b>1.1</b>	<b>Objetivos . . . . .</b>	<b>15</b>
<i>1.1.1</i>	<i>Objetivo geral . . . . .</i>	<i>15</i>
<i>1.1.2</i>	<i>Objetivos específicos . . . . .</i>	<i>16</i>
<b>2</b>	<b>REFERENCIAL TEÓRICO . . . . .</b>	<b>17</b>
<b>2.1</b>	<b>Sistemas Educacionais Inteligentes e Ambientes Virtuais de Aprendizagem (LMS) . . . . .</b>	<b>17</b>
<i>2.1.1</i>	<i>Comparação entre o Moodle e o sistema proposto . . . . .</i>	<i>18</i>
<b>2.2</b>	<b>Inteligência Artificial e Modelos de Linguagem na Educação . . . . .</b>	<b>19</b>
<b>2.3</b>	<b>O <i>framework</i> LangChain e o Papel da OpenAI . . . . .</b>	<b>20</b>
<b>2.4</b>	<b>Django, Django REST <i>framework</i> e Arquitetura de <i>back-end</i> . . . . .</b>	<b>21</b>
<b>2.5</b>	<b>RabbitMQ e Arquiteturas Orientadas a Eventos . . . . .</b>	<b>22</b>
<b>2.6</b>	<b>Trabalhos Relacionados . . . . .</b>	<b>22</b>
<b>3</b>	<b>METODOLOGIA . . . . .</b>	<b>25</b>
<b>3.1</b>	<b>Visão geral da arquitetura do sistema . . . . .</b>	<b>25</b>
<b>3.2</b>	<b>Desenvolvimento do <i>back-end</i> acadêmico . . . . .</b>	<b>26</b>
<b>3.3</b>	<b>Sistema de autenticação e controle de acesso . . . . .</b>	<b>26</b>
<b>3.4</b>	<b>Desenvolvimento do <i>front-end</i> web . . . . .</b>	<b>26</b>
<b>3.5</b>	<b>Implementação do assistente inteligente . . . . .</b>	<b>27</b>
<b>3.6</b>	<b>Integração entre os módulos . . . . .</b>	<b>27</b>
<b>3.7</b>	<b>Validação e testes da aplicação . . . . .</b>	<b>28</b>
<b>3.8</b>	<b>Funcionalidades do sistema . . . . .</b>	<b>28</b>
<i>3.8.1</i>	<i>Funcionalidades disponíveis para o aluno . . . . .</i>	<i>28</i>
<i>3.8.2</i>	<i>Funcionalidades disponíveis para o professor . . . . .</i>	<i>28</i>
<i>3.8.3</i>	<i>Modelo de permissões e controle de acesso . . . . .</i>	<i>29</i>
<i>3.8.4</i>	<i>Organização e relacionamento entre as entidades . . . . .</i>	<i>29</i>
<i>3.8.5</i>	<i>Gestão de datas e prazos . . . . .</i>	<i>29</i>
<i>3.8.6</i>	<i>Papel do assistente inteligente nas funcionalidades . . . . .</i>	<i>29</i>
<b>3.9</b>	<b>Funcionamento do chatbot e mecanismo de recomendação . . . . .</b>	<b>30</b>

3.9.1	<i>Fluxo geral de funcionamento do chatbot</i>	30
3.9.2	<i>Recursos principais do chatbot</i>	30
3.9.3	<i>Mecanismo de cálculo da “melhor matéria”</i>	30
4	<b>RESULTADOS</b>	<b>32</b>
4.1	<b>Cenários de validação e conjunto de dados</b>	32
4.2	<b>Validação da gestão acadêmica</b>	32
4.3	<b>Validação da interface e da experiência de uso</b>	33
4.4	<b>Validação do assistente inteligente</b>	33
4.5	<b>Análise do mecanismo de recomendação</b>	34
4.6	<b>Limitações observadas</b>	34
4.7	<b>Demonstração do uso do sistema</b>	34
4.7.1	<i>Acesso ao sistema</i>	35
4.7.2	<i>Fluxo de uso pelo professor</i>	35
4.7.3	<i>Fluxo de uso pelo aluno</i>	38
4.7.4	<i>Uso do chatbot como camada complementar</i>	38
4.7.5	<i>Demonstração do mecanismo de recomendação</i>	39
4.8	<b>Síntese dos resultados</b>	41
5	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	<b>42</b>
5.1	<b>Publicações Realizadas</b>	42
5.2	<b>Trabalhos Futuros</b>	43
	<b>REFERÊNCIAS</b>	<b>44</b>

# 1 INTRODUÇÃO

A crescente presença da inteligência artificial (IA) nas mais diversas áreas da sociedade tem promovido transformações profundas na forma como informações são produzidas, interpretadas e utilizadas. Setores como saúde, indústria, comércio e educação vêm incorporando, de maneira progressiva, sistemas inteligentes capazes de apoiar processos decisórios, automatizar tarefas complexas e ampliar a capacidade humana de análise. Nesse contexto, observa-se uma transição gradual de sistemas meramente informatizados para sistemas capazes de interpretar dados, reconhecer padrões e gerar conhecimento a partir de grandes volumes de informação, configurando um novo paradigma para os ambientes educacionais (Garzón, 2025)

É de conhecimento geral que no campo educacional, essa transformação ocorre em um cenário marcado, em grande medida, pela ampla adoção de sistemas e plataformas digitais intensificada a partir do período da pandemia de COVID-19, que acelerou a digitalização de processos acadêmicos e a incorporação de ambientes virtuais de aprendizagem (UNESCO, 2020). Somase a esse contexto o aumento significativo da quantidade de dados gerados sobre o percurso dos estudantes. Registros de acesso a plataformas, desempenho em avaliações, histórico acadêmico e interações em ambientes virtuais compõem um conjunto heterogêneo de informações que, quando analisadas de forma adequada, podem oferecer subsídios valiosos para a melhoria da qualidade do ensino e para o acompanhamento individualizado do aluno (Li; Liu, 2025).

Entretanto, embora exista uma grande disponibilidade de dados, sua exploração ainda é, em muitos casos, limitada por abordagens tradicionais de análise, frequentemente baseadas em relatórios estáticos, indicadores isolados e processos manuais de consolidação de informações. Esse modelo dificulta a obtenção de uma visão integrada e contextualizada sobre o processo de aprendizagem, reduzindo o potencial estratégico dos dados educacionais e limitando sua utilização como instrumento efetivo de apoio à tomada de decisão pedagógica e administrativa.

Paralelamente, os avanços recentes nos modelos de linguagem de larga escala (*Large Language Models – LLMs*) abriram novas possibilidades para a interpretação e a síntese de informações em linguagem natural. Modelos como os da família GPT demonstram capacidade de compreender contextos complexos, correlacionar informações provenientes de diferentes fontes e gerar textos analíticos e explicativos de forma automatizada. Essas características têm impulsionado o surgimento de aplicações em áreas como atendimento automatizado, análise documental e suporte à decisão, incluindo o domínio educacional (Brynjolfsson; Li; Raymond, 2023; Quisi-Peralta *et al.*, 2024).

No contexto da educação, o uso de LLMs possibilita não apenas a automação de tarefas, mas também a construção de sistemas capazes de produzir interpretações mais ricas sobre o desempenho estudantil, identificar padrões de comportamento e apoiar intervenções pedagógicas mais precisas. No entanto, a integração dessas tecnologias a sistemas educacionais reais ainda apresenta desafios relevantes, especialmente no que diz respeito à organização das informações, à confiabilidade das análises e à escalabilidade das soluções propostas.

Diante desse cenário, emerge o seguinte problema de pesquisa: como integrar modelos de linguagem de larga escala a sistemas educacionais de forma a oferecer uma visão contextualizada, confiável e útil sobre o desempenho e o perfil dos estudantes, mantendo requisitos de escalabilidade, viabilidade técnica e sustentabilidade da solução?

A investigação desse problema justifica-se sob múltiplas perspectivas. Do ponto de vista educacional, sistemas mais inteligentes e integrados podem contribuir para a personalização da aprendizagem, para a identificação precoce de dificuldades e para o aprimoramento das estratégias de ensino (Antunes *et al.*, 2018; Yan *et al.*, 2023). Sob a ótica social, soluções desse tipo podem auxiliar na democratização do acesso a ferramentas avançadas de análise educacional, inclusive em contextos institucionais com recursos limitados. Já do ponto de vista técnico, o trabalho aborda o desafio de integrar diferentes paradigmas e tecnologias de software em uma arquitetura coesa, robusta e escalável.

Nesse contexto, este trabalho propõe o desenvolvimento de um sistema educacional inteligente que utiliza o *framework* LangChain em conjunto com modelos de linguagem da OpenAI para gerar uma visão analítica e contextualizada (*overview*) sobre o desempenho e o perfil do aluno. A proposta fundamenta-se na ideia de que a combinação entre modelos de linguagem, técnicas modernas de engenharia de software e uma infraestrutura adequada pode ampliar significativamente o valor extraído dos dados educacionais.

A solução proposta busca não apenas automatizar processos já existentes, mas também oferecer uma nova forma de interação com as informações acadêmicas, na qual gestores, docentes e estudantes possam acessar interpretações mais ricas, integradas e orientadas ao contexto. Dessa forma, espera-se contribuir para uma gestão educacional mais eficiente, transparente e orientada por dados (Hsueh *et al.*, 2025; Li; Liu, 2025).

Este trabalho está organizado da seguinte forma: no Capítulo 2 é apresentada a fundamentação teórica, abordando conceitos relacionados à inteligência artificial, modelos de linguagem, sistemas educacionais inteligentes e plataformas de gestão de aprendizagem. O Capítulo 3 descreve a metodologia adotada e a arquitetura geral da solução proposta. O Capítulo 4 apresenta o desenvolvimento e a implementação do sistema. O Capítulo 5 discute os resultados obtidos e as análises realizadas. Por fim, o Capítulo 6 apresenta as conclusões e as possibilidades de trabalhos futuros.

## 1.1 Objetivos

### 1.1.1 *Objetivo geral*

Desenvolver um sistema educacional inteligente que utilize o *framework* LangChain integrado a modelos de linguagem da OpenAI para gerar uma visão analítica (*overview*) do desempenho e perfil do aluno, apoiando a tomada de decisão pedagógica e administrativa.

O sistema será implementado em Django, com base de dados PostgreSQL e infraestrutura composta por ferramentas como Gunicorn, Nginx, RabbitMQ e Redis, de modo a garantir alta disponibilidade, escalabilidade e desempenho.

Além de aplicar conceitos de inteligência artificial e processamento de linguagem natural, o projeto busca demonstrar a viabilidade técnica e prática de uma arquitetura moderna para gestão educacional, promovendo automação, eficiência e inovação na análise de dados acadêmicos.

### 1.1.2 *Objetivos específicos*

1. Projetar e desenvolver uma aplicação *web* em Django que centralize informações sobre alunos, disciplinas e desempenho acadêmico, servindo como base para as análises inteligentes.
2. Integrar o *framework* LangChain à aplicação, possibilitando consultas semânticas, geração de relatórios e resumos automáticos baseados em dados provenientes do banco PostgreSQL.
3. Empregar modelos da OpenAI para interpretar contextos, gerar análises textuais e produzir um *overview* sobre o perfil do estudante, contemplando aspectos quantitativos e qualitativos.
4. Implementar tarefas assíncronas por meio do RabbitMQ e de um broker como Redis, garantindo o processamento em paralelo de relatórios e análises sem prejudicar o desempenho geral do sistema.
5. Validar a precisão e relevância das análises geradas pelo sistema, comparando resultados obtidos via LangChain e OpenAI com dados reais de desempenho estudantil e relatórios acadêmicos tradicionais.
6. Documentar o processo de desenvolvimento, abordando desde a modelagem de dados em PostgreSQL até a integração das ferramentas de IA e os aspectos de segurança, desempenho e escalabilidade da aplicação.

## 2 REFERENCIAL TEÓRICO

### 2.1 Sistemas Educacionais Inteligentes e Ambientes Virtuais de Aprendizagem (LMS)

Os *Learning Management Systems* (LMS) constituem a base tecnológica para o ensino a distância e o apoio a ambientes híbridos de aprendizagem. Essas plataformas centralizam conteúdos, avaliações e interações entre alunos e professores, permitindo acompanhar o progresso discente de forma estruturada.

No entanto, estudos recentes indicam que os LMS tradicionais enfrentam limitações significativas em termos de personalização da experiência do aluno e integração de tecnologias inteligentes (Li; Liu, 2025).

Em uma análise comparativa recente, Jain, Nandan e Bhosale (2024) examinaram diversas plataformas LMS amplamente utilizadas, avaliando-as quanto à eficiência técnica, usabilidade, segurança e recursos pedagógicos. Os resultados evidenciam o elevado grau de maturidade dessas soluções no suporte à gestão acadêmica e à organização de conteúdos educacionais. Entretanto, tais plataformas permanecem predominantemente orientadas a funções administrativas e de distribuição de conteúdo, não tendo como foco a exploração avançada dos dados educacionais para fins de análise semântica, interpretação automatizada ou suporte inteligente à tomada de decisão pedagógica.

De forma complementar, Sanchez, Penarreta e Poma (2024) realizaram um estudo comparativo de 45 plataformas LMS voltadas ao ensino superior por meio da métrica SCTL (*Software Quality and Teaching-Learning*), um modelo avaliativo que combina critérios de qualidade de software — como usabilidade, confiabilidade e eficiência — com aspectos pedagógicos relacionados às ferramentas de ensino-aprendizagem oferecidas pelas plataformas. O estudo apresenta um panorama técnico do estado da arte dessas soluções e confirma sua eficácia na gestão de cursos e usuários, reforçando seu papel como infraestruturas consolidadas de apoio à administração acadêmica.

Em conjunto, esses trabalhos indicam que, embora os LMS tradicionais apresentem elevada maturidade técnica e ampla adoção institucional, ainda existe uma lacuna significativa no uso sistemático de técnicas de inteligência artificial e aprendizado de máquina como suporte à análise de desempenho discente, à interpretação de dados educacionais e à personalização do processo de aprendizagem, o que evidencia a relevância e a atualidade da proposta deste trabalho.

De acordo com Garzón (2025), a aplicação de técnicas modernas de inteligência artificial, incluindo aprendizado de máquina, aprendizado profundo e processamento de linguagem natural (PLN), permite que sistemas educacionais avancem para além da simples análise quantitativa de desempenho, tornando-se capazes de contemplar também dimensões qualitativas do processo de aprendizagem, como níveis de engajamento, padrões de comportamento e indí-

cios de dificuldades cognitivas, ampliando significativamente o potencial de personalização e de apoio às decisões pedagógicas.

Dessa forma, o sistema proposto neste trabalho busca preencher essa lacuna ao combinar funcionalidades de um LMS tradicional com agentes inteligentes e chatbots integrados via LangChain e OpenAI, oferecendo relatórios automatizados e personalização contínua. Essa integração posiciona o projeto na categoria dos *AI-LMS*, uma nova geração de plataformas que unem robustez administrativa e inteligência contextual.

### 2.1.1 Comparação entre o Moodle e o sistema proposto

O Moodle representa um exemplo consolidado de LMS tradicional, cujo foco principal está na gestão de cursos, usuários, conteúdos e avaliações. Sua arquitetura e seu modelo de uso são orientados à organização e à disponibilização de informações educacionais, cabendo ao professor e ao gestor a tarefa de interpretar os dados apresentados e tomar decisões pedagógicas a partir deles.

Nesse contexto, o Moodle atua principalmente como um repositório estruturado de informações acadêmicas e como uma plataforma de mediação entre alunos e docentes.

O sistema proposto neste trabalho, por sua vez, adota uma abordagem híbrida.

Assim como um LMS convencional, ele possui funcionalidades centrais de gestão acadêmica, incluindo o cadastro e a manutenção de disciplinas, aulas, tarefas, provas e notas. Entretanto, sobre essa base administrativa, é incorporada uma camada adicional de inteligência artificial, materializada por meio de um assistente conversacional integrado ao sistema.

A principal diferença conceitual entre as duas abordagens reside no papel atribuído ao sistema no processo educacional. Enquanto no Moodle a plataforma é essencialmente passiva, limitando-se a armazenar e apresentar dados, no sistema proposto o ambiente passa a assumir um papel ativo na mediação das informações, respondendo perguntas dos alunos, consultando dados acadêmicos internos e fornecendo orientações sobre o que estudar, o estado de tarefas e informações institucionais relevantes.

Do ponto de vista técnico, o Moodle não possui, em sua arquitetura central, mecanismos nativos de interpretação semântica ou de raciocínio automatizado sobre os dados educacionais.

Já o sistema desenvolvido neste trabalho integra modelos de linguagem por meio do *framework* LangChain, utilizando-os como camada de geração de respostas, e emprega o LangGraph para o roteamento de intenções e controle de estado do agente conversacional.

Embora a utilização desses *frameworks* ainda esteja em estágio inicial e focada principalmente na interação sob demanda, essa arquitetura já caracteriza uma mudança de paradigma em relação aos LMS tradicionais.

Outra distinção relevante está no modelo de uso da informação. No Moodle, dados como

notas, tarefas e disciplinas são consumidos de forma predominantemente descritiva, exigindo que o usuário humano realize a síntese e a interpretação.

No sistema proposto, esses mesmos dados passam a ser utilizados diretamente pelo assistente para compor respostas contextualizadas, permitindo que o aluno interaja com o ambiente educacional de forma mais natural e orientada a objetivos.

É importante ressaltar que o escopo do sistema desenvolvido é deliberadamente mais restrito do que o de plataformas maduras como o Moodle.

Funcionalidades como fóruns, upload de conteúdos, sistemas completos de avaliação online e submissão de trabalhos não fazem parte do foco do projeto.

Em contrapartida, a proposta central não é competir em completude funcional, mas demonstrar a viabilidade e o potencial de uma nova classe de sistemas, os chamados *AI-LMS*, nos quais as funcionalidades tradicionais de gestão educacional são combinadas com agentes inteligentes capazes de interpretar e mediar os dados acadêmicos.

Dessa forma, enquanto o Moodle representa o estado da arte dos LMS tradicionais orientados à gestão e à organização de conteúdo, o sistema proposto neste trabalho posiciona-se como um primeiro passo em direção a uma nova geração de plataformas educacionais, nas quais o ambiente não apenas armazena informações, mas também participa ativamente do processo de orientação e suporte ao estudante por meio de inteligência artificial.

## 2.2 Inteligência Artificial e Modelos de Linguagem na Educação

A aplicação de técnicas de inteligência artificial (IA) na educação tem se intensificado nos últimos anos, especialmente em sistemas voltados à análise de desempenho, apoio à aprendizagem e automação de tarefas administrativas. Entre essas técnicas, destacam-se os modelos de linguagem de larga escala (LLMs), como os modelos da família GPT, que permitem o processamento e a geração de textos em linguagem natural a partir de grandes volumes de dados.

Brynjolfsson, Li e Raymond analisam o impacto desse tipo de tecnologia sobre atividades intelectuais e mostram que sistemas baseados em LLMs podem ser utilizados como ferramentas de apoio à síntese de informações, à elaboração de diagnósticos e à produção de relatórios interpretativos (Brynjolfsson; Li; Raymond, 2023). Embora o estudo não seja restrito ao domínio educacional, seus resultados indicam que esse tipo de modelo é adequado para cenários em que é necessário transformar grandes quantidades de dados em informação compreensível para tomada de decisão, o que inclui o contexto acadêmico.

No campo específico da educação, Antunes et al. investigam o uso de chatbots educacionais como interfaces de apoio ao aluno (Antunes *et al.*, 2018). Os autores descrevem sistemas voltados principalmente à resolução de dúvidas e à revisão de conteúdos, destacando que essas ferramentas podem ampliar o acesso dos estudantes a suporte fora do horário de aula. Ao mesmo

tempo, o estudo aponta que a maioria dessas soluções ainda opera com fluxos de interação relativamente simples, baseados em regras ou em respostas pré-definidas.

O surgimento dos LLMs permite superar parte dessas limitações, possibilitando a construção de agentes capazes de interpretar perguntas em linguagem natural e de gerar respostas mais flexíveis e contextualizadas. Além disso, quando integrados a *frameworks* de orquestração, como o LangChain, esses modelos podem ser conectados a bases de dados e sistemas internos, passando a atuar não apenas como interfaces conversacionais, mas também como camadas de acesso e interpretação de informação estruturada.

No entanto, a utilização de IA nesse tipo de contexto também levanta questões importantes. Yan et al. discutem problemas relacionados à confiabilidade das respostas, à privacidade dos dados e à necessidade de supervisão humana em sistemas educacionais baseados em modelos de linguagem (Yan *et al.*, 2023). Os autores argumentam que, embora essas tecnologias possam auxiliar o processo educacional, elas não devem operar de forma opaca ou substituir completamente o julgamento humano em decisões pedagógicas.

Diante dessas considerações, o sistema proposto neste trabalho utiliza modelos de linguagem como ferramenta de apoio à organização e à interpretação de dados acadêmicos, e não como substituto da avaliação docente. A IA é empregada para sintetizar informações, responder consultas e gerar recomendações, mantendo o usuário humano como responsável final pela tomada de decisão.

Assim, o uso de modelos de linguagem no contexto deste projeto não se restringe à automação de tarefas isoladas, mas se insere em uma proposta mais ampla de suporte inteligente à gestão e ao acompanhamento acadêmico, na qual o sistema passa a atuar como uma camada intermediária entre os dados armazenados e os usuários que deles necessitam.

### 2.3 O *framework* LangChain e o Papel da OpenAI

O LangChain é um *framework* voltado à construção de aplicações baseadas em modelos de linguagem, cujo principal objetivo é organizar fluxos de processamento que combinam chamadas a LLMs com acesso a fontes externas de dados, como bancos de dados, APIs e documentos. Diferentemente de um uso direto de modelos de linguagem, o *framework* fornece uma estrutura para encadear etapas de consulta, recuperação de informação e geração de respostas de forma controlada.

Hsueh et al. analisam o uso do LangChain como infraestrutura para sistemas baseados em recuperação aumentada de informação (*retrieval-augmented generation*) e destacam sua modularidade e sua capacidade de estruturar pipelines compostos por múltiplas etapas de consulta, processamento e síntese (Hsueh *et al.*, 2025). No trabalho dos autores, o *framework* é utilizado como camada intermediária entre modelos de linguagem e bases de dados, permitindo que as respostas geradas sejam fundamentadas em informações externas e não apenas no conhecimento

implícito do modelo.

Em uma aplicação voltada especificamente ao contexto educacional, Quisi-Peralta et al. utilizam o LangChain em conjunto com modelos da OpenAI para a construção de agentes capazes de analisar portfólios de aprendizagem e gerar relatórios descritivos sobre o progresso dos estudantes (Quisi-Peralta *et al.*, 2024). Nesse estudo, o papel do *framework* é coordenar o acesso aos dados educacionais e organizar o fluxo de geração das respostas, evidenciando seu uso como mecanismo de orquestração, e não apenas como uma biblioteca de chamadas a modelos de linguagem.

Esses trabalhos indicam que a principal contribuição do LangChain não está no modelo de linguagem em si, mas na sua função como camada de integração e controle do fluxo de informações. Ao estruturar a interação entre dados, ferramentas externas e modelos de linguagem, o *framework* permite que o sistema deixe de atuar apenas como repositório de dados e passe a oferecer respostas construídas a partir de múltiplas fontes de informação.

No contexto deste trabalho, o LangChain é utilizado com essa mesma finalidade de orquestração. Ele atua como componente responsável por coordenar as consultas ao banco de dados, a organização do contexto e o acionamento dos modelos da OpenAI, integrando-se à *API REST* desenvolvida em Django. Dessa forma, o modelo de linguagem não opera de forma isolada, mas sempre mediado por uma camada que controla quais dados são utilizados e como a resposta é construída.

O uso combinado do LangChain com modelos da OpenAI permite que informações quantitativas, como notas, frequência e prazos de atividades, sejam reunidas e apresentadas em forma de respostas textuais interpretativas. Essa abordagem diferencia o sistema proposto dos LMS convencionais, que se limitam, em geral, à exibição direta de métricas, sem uma camada intermediária responsável por sintetizar e contextualizar essas informações para o usuário.

## 2.4 Django, Django REST *framework* e Arquitetura de *back-end*

O Django é um *framework* Python de alto desempenho voltado ao desenvolvimento de aplicações *web* seguras e escaláveis. Sua estrutura baseada no padrão *Model-View-Template* (MVT) garante modularidade e facilita a manutenção de sistemas complexos (Carmo, 2023).

O Django REST *framework* (DRF) estende o Django, permitindo a criação de *APIs RESTful* que servem de ponte entre o *back-end* e os módulos externos, como o LangChain e os modelos de IA da OpenAI.

Essa arquitetura possibilita que a aplicação seja consumida tanto por interfaces *web* quanto por agentes inteligentes, mantendo segurança e padronização no tráfego de dados.

DigitalOcean (2025) destacam que a combinação entre Django, PostgreSQL, Nginx e Gunicorn forma uma infraestrutura estável e eficiente para implantação em produção, garantindo

alta disponibilidade.

Essa estrutura forma a espinha dorsal do sistema proposto, em que a camada REST fornece os dados educacionais, o LangChain processa e interpreta as informações, e a IA da OpenAI gera insights personalizados.

Essa composição garante escalabilidade, segurança e interoperabilidade requisitos fundamentais para um LMS inteligente.

## 2.5 RabbitMQ e Arquiteturas Orientadas a Eventos

O RabbitMQ é um *message broker* amplamente usado em sistemas distribuídos para tratar comunicações assíncronas entre serviços.

No estudo de Arafat *et al.* (2025), que compara diferentes sistemas de mensageria em arquiteturas orientadas a eventos, verifica-se que tais arquiteturas mediadas por mensageria propiciam escalabilidade, tolerância a falhas e bom desempenho sob cargas elevadas, sendo apropriadas para aplicações que requerem alta disponibilidade e processamento contínuo de eventos.

Voreakou *et al.* (2025) complementam que o RabbitMQ se destaca pela eficiência energética e pela estabilidade operacional, o que o torna adequado para ambientes de uso contínuo, como plataformas educacionais online.

No sistema proposto, o RabbitMQ é responsável por gerenciar tarefas em segundo plano, como geração de relatórios e atualizações periódicas de métricas, garantindo que o desempenho do LMS não seja afetado durante o uso intenso.

A integração entre RabbitMQ, Django REST e LangChain cria um fluxo contínuo de processamento e resposta, no qual o *back-end* envia dados, o *framework* de IA realiza inferências e o resultado retorna ao ambiente de aprendizagem de forma transparente e escalável.

Isso representa um avanço considerável sobre os LMS tradicionais, que não possuem mecanismos internos de mensageria nem automação inteligente.

## 2.6 Trabalhos Relacionados

Os ambientes virtuais de aprendizagem (LMS) evoluíram ao longo das últimas décadas principalmente como plataformas voltadas à gestão de cursos, usuários e conteúdos. Sistemas amplamente adotados nesse contexto consolidaram-se como infraestruturas administrativas estáveis para apoio ao ensino presencial e a distância, mas, em sua maioria, permanecem centrados na organização e na disponibilização de informações.

Entre os exemplos mais difundidos, o Moodle destaca-se como uma das plataformas mais utilizadas mundialmente, sendo desenvolvido em PHP conforme descrito em sua documentação oficial (Moodle, 2025). O Canvas LMS, mantido pela Instructure Inc., é construído

sobre o *framework Ruby on Rails* (Instructure, 2021), enquanto o Blackboard Learn é desenvolvido em Java, utilizando versões recentes da JDK (Blackboard, 2020). Esses sistemas ilustram a diversidade de linguagens e arquiteturas empregadas em LMS tradicionais, mas também evidenciam um traço comum: o foco principal permanece na gestão administrativa e na apresentação de dados de forma predominantemente descritiva.

Embora robustas do ponto de vista funcional, essas plataformas não possuem, em sua arquitetura central, integração nativa com modelos de linguagem de larga escala (LLMs), chat-bots semânticos ou mecanismos de análise contextual baseados em inteligência artificial (Li; Liu, 2025; Hsueh *et al.*, 2025). Como consequência, a interpretação dos dados acadêmicos continua sendo uma tarefa essencialmente delegada ao usuário humano, que precisa navegar pelas interfaces e sintetizar manualmente as informações apresentadas.

Além disso, muitas dessas plataformas adotam arquiteturas predominantemente monolíticas, como é o caso do Moodle, o que pode dificultar tanto a escalabilidade quanto a integração com serviços externos de processamento distribuído. Esse modelo arquitetural contrasta com abordagens mais recentes, baseadas em serviços desacoplados e processamento assíncrono.

Em trabalhos mais recentes voltados ao uso de IA na educação, observa-se uma tentativa de introduzir camadas de inteligência sobre os dados educacionais. Li e Liu propõem sistemas que utilizam modelos de linguagem para interpretar informações acadêmicas e gerar relatórios analíticos (Li; Liu, 2025), enquanto Hsueh *et al.* exploram arquiteturas baseadas em recuperação aumentada de informação para fundamentar respostas geradas por LLMs em bases de dados externas (Hsueh *et al.*, 2025). Esses estudos, no entanto, não têm como foco principal a integração direta desse tipo de abordagem a um LMS completo de gestão acadêmica.

O sistema proposto neste trabalho posiciona-se nesse espaço intermediário. Do ponto de vista arquitetural, adota uma estrutura modular baseada em Django (Python), complementada por RabbitMQ e Redis para o processamento de tarefas assíncronas, o que permite uma separação mais clara entre as camadas de gestão de dados, de inteligência artificial e de interface. Essa organização facilita a comunicação entre o *back-end* acadêmico, o LangChain e os modelos da OpenAI, estabelecendo um fluxo contínuo de consulta, interpretação e geração de respostas.

Do ponto de vista funcional, enquanto os LMS tradicionais priorizam a organização de cursos e usuários, o presente projeto propõe uma plataforma do tipo *AI-LMS* orientada à análise semântica e à mediação inteligente das informações. A integração entre Django REST *framework*, PostgreSQL e LangChain permite que os dados acadêmicos deixem de ser apenas exibidos e passem a ser interpretados pelo próprio sistema, resultando em respostas em linguagem natural, resumos e recomendações contextualizadas.

Em síntese, o trabalho diferencia-se das soluções convencionais não por competir em termos de completude funcional, mas por demonstrar a viabilidade de integrar modelos de linguagem de larga escala, processamento assíncrono de eventos e uma arquitetura *web* modular

em um único ambiente. Essa combinação caracteriza o sistema como uma evolução conceitual dos LMS tradicionais, no qual a plataforma deixa de ser apenas um repositório de informações e passa a atuar como um agente ativo de mediação e interpretação dos dados educacionais.

Tabela 1 – Comparação entre LMS tradicional (Moodle) e o sistema proposto

<b>Critério</b>	<b>Moodle (LMS tradicional)</b>	<b>Sistema Proposto (AI-LMS)</b>
Papel do sistema	Plataforma de organização e disponibilização de conteúdo	Sistema ativo que acompanha, analisa e orienta o aluno
Tipo de interação	Interface baseada em navegação por telas e menus	Interface tradicional integrada a uma interface conversacional (chatbot)
Uso de Inteligência Artificial	Não possui IA nativa integrada ao fluxo principal do sistema	Possui camada de IA baseada em LangChain e LangGraph integrada ao sistema
Personalização da experiência	Limitada a filtros e visualizações estáticas	Dinâmica e baseada no desempenho, pendências e proximidade de avaliações
Forma de acesso às informações	O usuário precisa navegar manualmente pelas telas	O usuário pode obter informações por meio de perguntas em linguagem natural
Capacidade de recomendação	Inexistente ou dependente de plugins externos	Nativa, baseada em cálculo de score por disciplina
Integração entre módulos	Módulos relativamente independentes	Todos os módulos são integrados por meio de uma API única
Atualização de dados	Exclusivamente por meio das interfaces administrativas	Pode ser realizada tanto pelas interfaces quanto pelo chatbot
Modelo de uso	Reativo: o usuário precisa buscar a informação	Proativo: o sistema sugere ações e prioridades automaticamente
Complexidade para o usuário	Exige que o usuário compreenda a estrutura interna do sistema	Permite o uso sem a necessidade de conhecer a estrutura interna
Automação do acompanhamento acadêmico	Inexistente de forma nativa	Nativa, baseada em regras e métricas de desempenho

### 3 METODOLOGIA

Este capítulo descreve a metodologia adotada para o desenvolvimento do sistema educacional inteligente proposto neste trabalho. A abordagem utilizada é de natureza aplicada e experimental, com foco na implementação prática de uma plataforma web que integra funcionalidades típicas de um *LMS* com um assistente inteligente baseado em modelos de linguagem. O desenvolvimento foi dividido em módulos, contemplando o *back-end*, o *front-end*, o módulo de inteligência artificial e os mecanismos de integração entre esses componentes.

#### 3.1 Visão geral da arquitetura do sistema

O sistema foi projetado segundo uma arquitetura modular, composta por quatro blocos principais: (i) um *back-end* acadêmico responsável pela gestão dos dados e regras de negócio, (ii) um *front-end* web para interação com alunos e professores, (iii) um módulo de inteligência artificial baseado em *LangChain* e *LangGraph*, e (iv) um sistema de autenticação e controle de acesso baseado em tokens *JWT*.

A comunicação entre os componentes ocorre por meio de uma *API REST* desenvolvida com *Django REST Framework*. O *front-end*, implementado em *React* com *Vite*, consome essa *API* tanto para as operações tradicionais do sistema acadêmico quanto para a interação com o assistente inteligente.

O banco de dados utilizado no ambiente de desenvolvimento é o *PostgreSQL*, responsável pelo armazenamento das informações acadêmicas. De forma simplificada, o fluxo principal do sistema ocorre da seguinte maneira: o usuário interage com a interface web, o *front-end* envia requisições para a *API Django*, que por sua vez consulta o banco de dados ou aciona o módulo de inteligência artificial, retornando a resposta processada para o cliente.

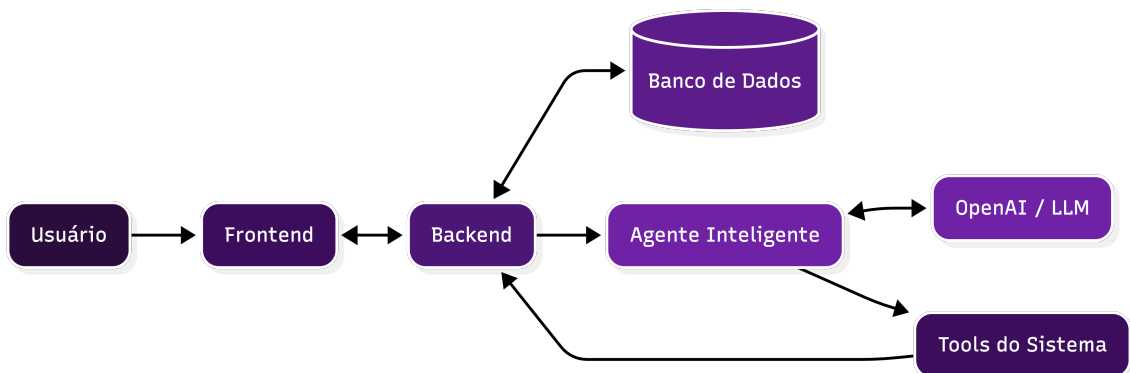


Figura 1 – Arquitetura geral do sistema proposto

Fonte: Elaboração própria.

A Figura 1 apresenta a arquitetura geral do sistema proposto, evidenciando a separação em camadas de apresentação, serviços, domínio e dados, bem como a integração do módulo de

inteligência artificial baseado em *LLM* e memória persistente com os demais componentes da plataforma.

### 3.2 Desenvolvimento do *back-end* acadêmico

O *back-end* do sistema foi desenvolvido utilizando o framework *Django* em conjunto com o *Django REST Framework (DRF)*, adotando uma arquitetura orientada a serviços baseada em *API REST*. Essa camada é responsável por centralizar as regras de negócio e o acesso aos dados persistidos no banco de dados *PostgreSQL*.

Foram implementadas entidades fundamentais para o domínio educacional, incluindo usuário, disciplina, aula, tarefa, prova e nota. Para cada uma dessas entidades, foram criados *endpoints* de criação, leitura, atualização e remoção (*CRUD*), permitindo a gestão completa das informações acadêmicas pelo sistema.

Além das operações básicas de *CRUD*, o *back-end* também disponibiliza ações específicas, como a consulta de tarefas pendentes, a verificação de provas futuras e a atualização do status de tarefas.

Essas operações são utilizadas tanto pelo *front-end* quanto pelo módulo de inteligência artificial, garantindo que ambos acessem a mesma fonte de dados e mantenham consistência nas informações apresentadas.

### 3.3 Sistema de autenticação e controle de acesso

O sistema implementa um mecanismo de autenticação baseado em login e senha, utilizando *JSON Web Tokens (JWT)* por meio da biblioteca *SimpleJWT*. Cada usuário possui um perfil associado, podendo ser classificado como aluno ou professor, o que permite restringir o acesso a determinadas funcionalidades da plataforma.

Esse mecanismo garante que apenas usuários autenticados possam acessar o sistema e que as operações realizadas pelo assistente inteligente sejam contextualizadas de acordo com o usuário logado, possibilitando respostas personalizadas e coerentes com sua situação acadêmica.

### 3.4 Desenvolvimento do *front-end* web

O *front-end* da aplicação foi desenvolvido utilizando a biblioteca *React* em conjunto com a ferramenta *Vite*, adotando uma abordagem de interface baseada em componentes.

A aplicação web possui painéis distintos para alunos e professores, organizados em abas e seções que permitem a gestão e a consulta das informações acadêmicas. Além das telas de administração e visualização de dados, o sistema também incorpora uma interface de *chat* inte-

grada aos *dashboards*, por meio da qual o usuário pode interagir diretamente com o assistente inteligente.

Todas as interações do *front-end* com o sistema ocorrem por meio de requisições *HTTP* à *API REST* desenvolvida no *back-end*.

### 3.5 Implementação do assistente inteligente

O núcleo inteligente do sistema foi desenvolvido utilizando os frameworks *LangChain* e *LangGraph*, integrados a modelos de linguagem compatíveis com a *API* da *OpenAI*, com suporte adicional a provedores alternativos como *OpenRouter* e *DeepSeek*.

O funcionamento do assistente segue o seguinte fluxo: o usuário envia uma mensagem pela interface de *chat*, o *front-end* encaminha essa mensagem para um *endpoint* específico da *API*, que aciona o *LangGraph*. Esse componente é responsável por identificar a intenção da mensagem e definir o fluxo de processamento adequado.

Em seguida, são utilizadas ferramentas que realizam consultas ao banco de dados por meio dos modelos do *Django*, reunindo as informações necessárias para compor o contexto da resposta.

Por fim, o *LangChain* organiza esse contexto e o envia ao modelo de linguagem, que gera a resposta final a ser retornada ao usuário.

Atualmente, o agente é capaz de reconhecer intenções como consulta de tarefas pendentes, atualização de status de tarefas, verificação de próximas provas, recomendação do que estudar, solicitação de informações institucionais e pedidos gerais de ajuda. Caso nenhuma intenção específica seja identificada, o sistema utiliza um mecanismo de *fallback* para responder de forma genérica.

### 3.6 Integração entre os módulos

A integração entre os diferentes módulos do sistema ocorre de forma centralizada pela *API REST*. Tanto o *front-end* quanto o módulo de inteligência artificial utilizam os mesmos *endpoints* e modelos de dados, garantindo consistência e reutilização de código.

Do ponto de vista arquitetural, o sistema opera segundo dois fluxos principais: (i) *React* → *API Django* → Banco de dados, para operações tradicionais de gestão acadêmica, e (ii) *React* → *API Django* → *LangGraph* → Modelo de linguagem → *API Django* → *React*, para as interações mediadas pelo assistente inteligente.

Essa organização modular facilita a manutenção do sistema e permite a evolução independente de cada componente, especialmente no que diz respeito ao aprimoramento futuro do módulo de inteligência artificial.

### 3.7 Validação e testes da aplicação

A validação do sistema foi realizada por meio de testes manuais de uso, nos quais diferentes cenários de interação foram simulados, incluindo a navegação pelas telas administrativas, o cadastro e a consulta de dados acadêmicos, bem como a interação com o assistente inteligente por meio do *chat*.

Foram testados casos de uso como a consulta de tarefas pendentes, a verificação de provas futuras e a solicitação de recomendações de estudo, com o objetivo de verificar a coerência das respostas e a correta integração entre os módulos do sistema.

Embora não tenham sido implementados testes automatizados até o momento, os testes manuais permitiram validar o funcionamento geral da plataforma e identificar pontos de melhoria para trabalhos futuros.

### 3.8 Funcionalidades do sistema

O sistema foi projetado para atender dois perfis principais de usuários: alunos e professores, cada um com permissões e funcionalidades específicas. De forma geral, a plataforma oferece recursos de gestão acadêmica básica integrados a um assistente inteligente capaz de consultar e manipular esses dados de forma contextualizada.

#### 3.8.1 Funcionalidades disponíveis para o aluno

O aluno pode acessar as aulas associadas às disciplinas nas quais está matriculado, bem como consultar tarefas, provas e notas de provas. Além disso, o sistema permite que o aluno acompanhe os prazos de entrega de tarefas e as datas das provas por meio das telas de listagem.

Uma funcionalidade central para o aluno é o gerenciamento do status de suas tarefas, que podem assumir os estados *pendente*, *em andamento* e *concluída*. O próprio aluno pode atualizar esses estados conforme seu progresso.

Além das funcionalidades tradicionais de consulta, o aluno pode interagir com o assistente inteligente por meio de uma interface de *chat*, realizando perguntas sobre sua situação acadêmica, como próximas provas, tarefas pendentes e recomendações de estudo, bem como solicitar a execução de ações simples, como a atualização do status de tarefas.

#### 3.8.2 Funcionalidades disponíveis para o professor

O professor possui acesso às funcionalidades de gestão acadêmica do sistema. Ele pode criar, editar e remover disciplinas, aulas, tarefas e provas, além de lançar, editar e remover notas de provas dos alunos matriculados em suas disciplinas. Também é responsabilidade do professor

gerenciar as matrículas dos alunos em suas disciplinas, garantindo que apenas os estudantes associados tenham acesso aos dados correspondentes.

As permissões do professor são restritas às disciplinas sob sua responsabilidade, não sendo possível modificar informações de outras áreas.

### **3.8.3 Modelo de permissões e controle de acesso**

O sistema adota um modelo de permissões baseado em perfis de usuário. Alunos possuem acesso apenas para leitura às entidades acadêmicas, com exceção da possibilidade de atualizar o status de suas próprias tarefas.

Professores, por sua vez, possuem acesso completo (*CRUD*) às entidades relacionadas às disciplinas que ministram. A criação de usuários ocorre por meio de registro no sistema, não havendo um módulo administrativo completo para o gerenciamento arbitrário de contas, o que reforça o foco da aplicação na gestão acadêmica e na interação inteligente com os dados.

### **3.8.4 Organização e relacionamento entre as entidades**

Do ponto de vista do modelo de dados, uma disciplina pode possuir várias aulas e vários alunos associados. Cada aula pertence a uma disciplina e pode conter diversas tarefas.

As provas estão associadas às disciplinas e podem se relacionar conceitualmente com aulas e tarefas. As notas pertencem simultaneamente a uma prova e a um aluno.

O status de tarefa representa a relação entre um aluno e uma tarefa específica.

### **3.8.5 Gestão de datas e prazos**

O sistema apresenta, nas telas de listagem, as datas das provas e os prazos de entrega das tarefas, permitindo que o aluno acompanhe seus compromissos acadêmicos.

Embora não exista uma visualização dedicada de calendário ou agenda, essas informações podem ser consultadas tanto pelas telas do sistema quanto por meio do assistente inteligente, que é capaz de responder, por exemplo, qual é a próxima prova agendada.

### **3.8.6 Papel do assistente inteligente nas funcionalidades**

O assistente inteligente atua como uma camada adicional de interação com o sistema. Ele não apenas consulta dados acadêmicos para responder perguntas em linguagem natural, mas também é capaz de executar ações simples, como a atualização do status de tarefas.

Dessa forma, o assistente funciona como uma interface alternativa e complementar às telas tradicionais do sistema, auxiliando alunos e professores a interpretar e utilizar as informações disponíveis na plataforma.

Em síntese, a principal funcionalidade do sistema consiste na integração entre a gestão acadêmica básica (aulas, tarefas, provas e notas) e um assistente inteligente, que permite aos usuários não apenas acessar os dados, mas também interagir com eles de forma mais natural e orientada a objetivos.

### 3.9 Funcionamento do chatbot e mecanismo de recomendação

Esta seção descreve de forma mais detalhada o funcionamento interno do *chatbot* do sistema, com ênfase no fluxo de decisão baseado em *LangGraph* e no mecanismo de cálculo utilizado para a recomendação da melhor matéria para estudo.

#### 3.9.1 Fluxo geral de funcionamento do chatbot

O funcionamento do *chatbot* segue um fluxo estruturado composto pelas seguintes etapas: o usuário envia uma mensagem pela interface de *chat*, o *front-end* encaminha essa mensagem para a *API*, que aciona o *LangGraph*.

O *LangGraph* é responsável por identificar o papel do usuário e a intenção da mensagem, selecionando o nó de execução apropriado. Em seguida, o nó executa as operações necessárias, que podem incluir consultas ao banco de dados ou a geração de dados intermediários.

Após isso, a função de geração de resposta (*generate\_response*) organiza o contexto e o encaminha ao modelo de linguagem, que produz a resposta final retornada ao usuário.

Atualmente, o sistema reconhece as seguintes intenções principais: listar tarefas pendentes, atualizar o status de tarefas, listar a próxima prova, recomendar matéria para estudo, informar dados institucionais da escola e fornecer ajuda geral ou respostas de *fallback*.

#### 3.9.2 Recursos principais do chatbot

O *chatbot* oferece como principais recursos: a consulta de tarefas pendentes do aluno, a consulta da próxima prova associada ao aluno ou professor, a atualização do status de tarefas (*pendente*, *em andamento* e *concluída*), a recomendação de estudo por meio da indicação da melhor matéria, o fornecimento de informações institucionais da escola e a resposta a solicitações gerais de ajuda.

Todas essas funcionalidades utilizam como fonte de dados exclusivamente o banco de dados relacional do sistema, por meio dos modelos *Materia*, *NotaProva*, *Prova* e *TarefaStatus*.

#### 3.9.3 Mecanismo de cálculo da “melhor matéria”

A recomendação de estudo é realizada por meio de um mecanismo heurístico que calcula um *score* para cada matéria associada ao aluno, levando em consideração três fatores principais:

desempenho acadêmico, quantidade de tarefas pendentes e proximidade de provas.

Para cada matéria, é calculada a média das notas de provas do aluno. Caso não existam notas registradas, assume-se o valor padrão 10,0. Esse valor é normalizado dividindo-se a média por 10, resultando em um número entre 0 e 1.

Como o objetivo é priorizar matérias com pior desempenho, utiliza-se o complemento desse valor (1 - média normalizada) no cálculo final. Esse fator possui peso 0,4 na composição do *score*.

O segundo fator considerado é a proporção de tarefas pendentes em relação ao total de tarefas da matéria. Esse valor é calculado pela razão entre o número de tarefas pendentes e o número total de tarefas, resultando em um valor normalizado entre 0 e 1.

Esse fator possui peso 0,3 no cálculo do *score*.

O terceiro fator considera a proximidade da próxima prova da matéria. Caso exista uma prova futura, é calculada a quantidade de dias restantes até a data da prova.

Esse valor é normalizado por meio da função  $\max(0, 1 - \frac{\text{dias}}{30})$ , de forma que provas mais próximas resultem em valores mais altos, enquanto provas distantes ou inexistentes resultem em valores próximos de zero.

Esse fator possui peso 0,3 na composição do *score*.

O *score* final de cada matéria é calculado pela seguinte fórmula:

$$\begin{aligned} \text{score} = & (1 - \text{media\_normalizada}) \times 0,4 \\ & + \text{tarefas\_normalizadas} \times 0,3 \\ & + \text{proximidade\_normalizada} \times 0,3 \end{aligned}$$

Após o cálculo, as matérias são ordenadas em ordem decrescente de *score*. A matéria com maior valor é recomendada como a melhor matéria para estudo no momento.

Além disso, o sistema também gera um *ranking* contendo, para cada matéria, a média de notas, a quantidade de pendências e os dias restantes para a próxima prova, permitindo transparência na recomendação apresentada ao usuário.

## 4 RESULTADOS

Este capítulo apresenta os resultados obtidos a partir da implementação e validação do sistema proposto. O objetivo principal desta etapa foi verificar se o protótipo funcional desenvolvido atende aos objetivos estabelecidos no trabalho, especialmente no que diz respeito à integração entre a gestão acadêmica tradicional e o uso de um assistente inteligente baseado em modelos de linguagem.

O sistema encontra-se no estágio de protótipo funcional, sendo utilizado neste trabalho como uma prova de conceito arquitetural. Dessa forma, os resultados aqui apresentados não têm como foco a avaliação de desempenho ou escalabilidade, mas sim a verificação do funcionamento integrado dos módulos e da coerência das funcionalidades implementadas em cenários de uso representativos.

### 4.1 Cenários de validação e conjunto de dados

A validação do sistema foi realizada por meio da execução de cenários completos de uso, contemplando tanto as funcionalidades administrativas quanto as funcionalidades voltadas ao aluno e ao assistente inteligente. Entre os principais cenários testados, destacam-se: a criação de disciplinas, aulas, tarefas e provas por parte do professor, a matrícula de alunos, o lançamento de notas, a consulta dessas informações pelo aluno e a interação com o *chatbot* para obtenção de informações e execução de ações.

Para a realização desses testes, foram utilizados dados fictícios gerados com o auxílio de modelos de linguagem, de forma a simular um ambiente educacional realista, contendo múltiplas disciplinas, tarefas, provas e registros de notas. Essa abordagem permitiu avaliar o comportamento do sistema em situações mais próximas de um uso real, sem a necessidade de utilizar dados sensíveis.

Esses cenários formam a base para as análises apresentadas nas seções seguintes, que discutem tanto o funcionamento do sistema acadêmico quanto o comportamento do assistente inteligente de forma integrada.

### 4.2 Validação da gestão acadêmica

A primeira etapa da validação concentrou-se na verificação do funcionamento do núcleo acadêmico do sistema. Foram testadas as principais operações de criação, leitura, atualização e remoção (*CRUD*) das entidades centrais, incluindo disciplinas, aulas, tarefas, provas e notas.

Os testes indicaram que as relações entre essas entidades — como a associação de aulas e tarefas a uma disciplina, e de notas a provas e alunos — comportaram-se de maneira consistente. Além disso, o controle de permissões entre os perfis de aluno e professor mostrou-se adequado,

garantindo que cada tipo de usuário pudesse acessar e modificar apenas as informações correspondentes ao seu papel no sistema.

Essa validação é fundamental, pois o funcionamento correto dessa camada é um pré-requisito para a atuação do assistente inteligente, que depende diretamente desses dados para realizar consultas, inferências e recomendações.

### 4.3 Validação da interface e da experiência de uso

Com a camada de dados validada, foi possível avaliar o comportamento da interface web. O sistema permitiu a navegação entre as diferentes seções e a visualização consistente das informações cadastradas. Alterações realizadas pelo professor, como o cadastro de novas tarefas ou o lançamento de notas, foram refletidas corretamente nas telas acessadas pelo aluno.

Durante os testes realizados, não foram observados problemas relevantes de usabilidade ou inconsistências visuais que comprometessem o uso do sistema. Embora a interface não tenha como objetivo competir com plataformas educacionais maduras, ela mostrou-se suficiente para suportar os fluxos de uso necessários para a validação do protótipo.

Essa estabilidade da interface reforça a ideia de que o sistema pode ser utilizado como base para a integração com o módulo de inteligência artificial, sem que falhas na camada visual comprometam a análise dos resultados.

### 4.4 Validação do assistente inteligente

Uma vez estabelecida a consistência do sistema acadêmico, a validação concentrou-se no comportamento do assistente inteligente. Os testes realizados demonstraram que o *chatbot* foi capaz de interpretar corretamente diferentes intenções de usuário, consultar os dados armazenados no banco e retornar respostas coerentes com o estado atual do sistema.

Além de consultas, como a listagem de tarefas pendentes e a verificação da próxima prova, o assistente também foi capaz de executar ações simples, como a atualização do status de tarefas. Essas ações foram refletidas corretamente tanto no banco de dados quanto na interface do usuário, evidenciando que o assistente não atua apenas como um componente informativo, mas como uma camada funcional integrada ao sistema.

Esse comportamento confirma a viabilidade do uso de modelos de linguagem como uma interface operacional para sistemas educacionais estruturados, e não apenas como um mecanismo de conversação genérica.

## 4.5 Análise do mecanismo de recomendação

Um dos pontos centrais do sistema é o mecanismo de recomendação da melhor matéria para estudo. Durante os testes, foi possível observar que a recomendação apresentada pelo sistema variou de acordo com modificações no estado acadêmico do aluno, como alterações nas notas, na quantidade de tarefas pendentes e na proximidade de provas.

Essas variações indicam que o modelo heurístico implementado responde dinamicamente às mudanças nos dados, e não produz recomendações estáticas ou arbitrárias. Embora não se trate de um sistema de recomendação baseado em aprendizado de máquina, os resultados mostram que mesmo uma abordagem baseada em regras e métricas simples pode produzir sugestões coerentes e úteis no contexto educacional.

Essa observação reforça o papel do sistema como uma prova de conceito, demonstrando que a integração entre dados acadêmicos estruturados e um agente inteligente pode gerar comportamentos adaptativos relevantes.

## 4.6 Limitações observadas

Apesar dos resultados positivos, algumas limitações ficaram evidentes durante o processo de validação. A principal delas refere-se à ausência de testes em larga escala, o que impede qualquer conclusão sobre desempenho, escalabilidade ou robustez do sistema em ambientes com muitos usuários simultâneos.

Além disso, o protótipo não contempla funcionalidades avançadas comuns em *LMS* consolidados, como envio de arquivos, fóruns ou geração automática de relatórios. Essas limitações são coerentes com o escopo experimental do trabalho e reforçam o caráter de prova de conceito da solução proposta.

## 4.7 Demonstração do uso do sistema

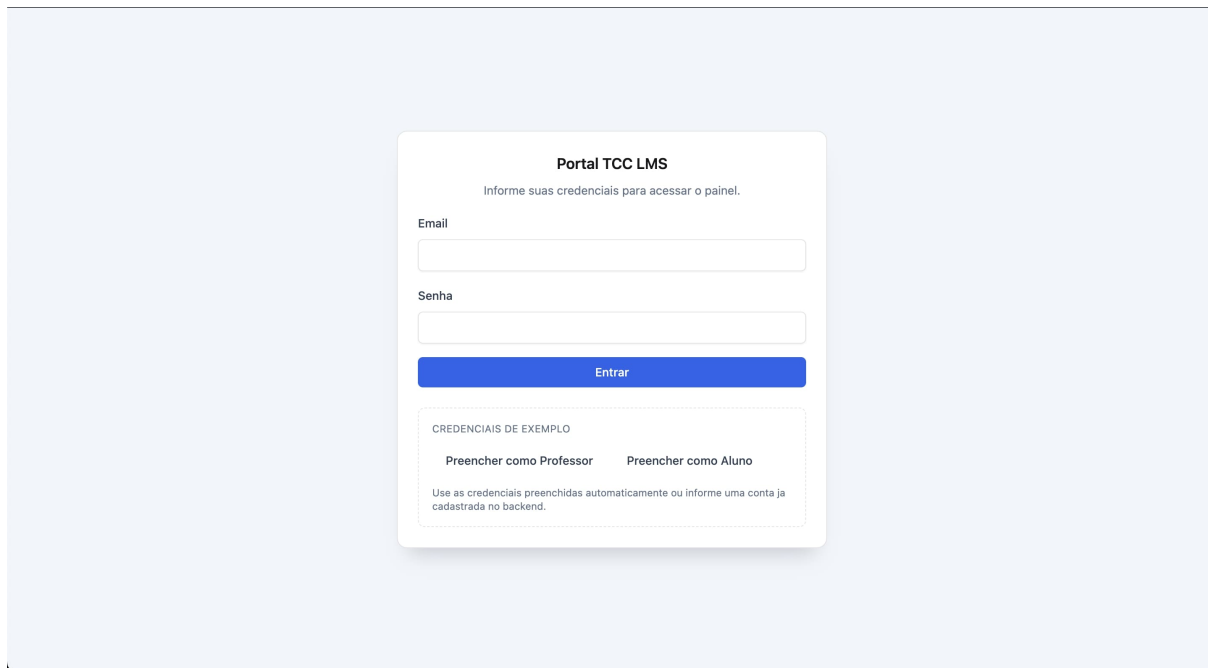
Esta seção apresenta o uso prático do sistema por meio de um fluxo completo de interação, partindo da preparação do ambiente pelo professor, passando pelo consumo das informações pelo aluno e, por fim, pela utilização do assistente inteligente como camada complementar de interação. O objetivo não é apenas ilustrar as telas do sistema, mas demonstrar que os diferentes módulos funcionam de forma integrada e coerente, refletindo corretamente as alterações realizadas nos dados.

Para os testes apresentados, parte-se do pressuposto de que os usuários já se encontram cadastrados no sistema, de modo que o foco da demonstração está nos fluxos principais de uso após a autenticação.

### 4.7.1 Acesso ao sistema

O acesso ao sistema é realizado por meio de uma tela de *login*, que solicita as credenciais do usuário e também disponibiliza atalhos para o preenchimento automático de contas de exemplo, conforme apresentado na Figura 2. Após a autenticação, o usuário é direcionado para o painel principal, cuja interface se adapta de acordo com o perfil (professor ou aluno).

Figura 2 – Tela de *login* do sistema.



Fonte: Elaboração própria.

### 4.7.2 Fluxo de uso pelo professor

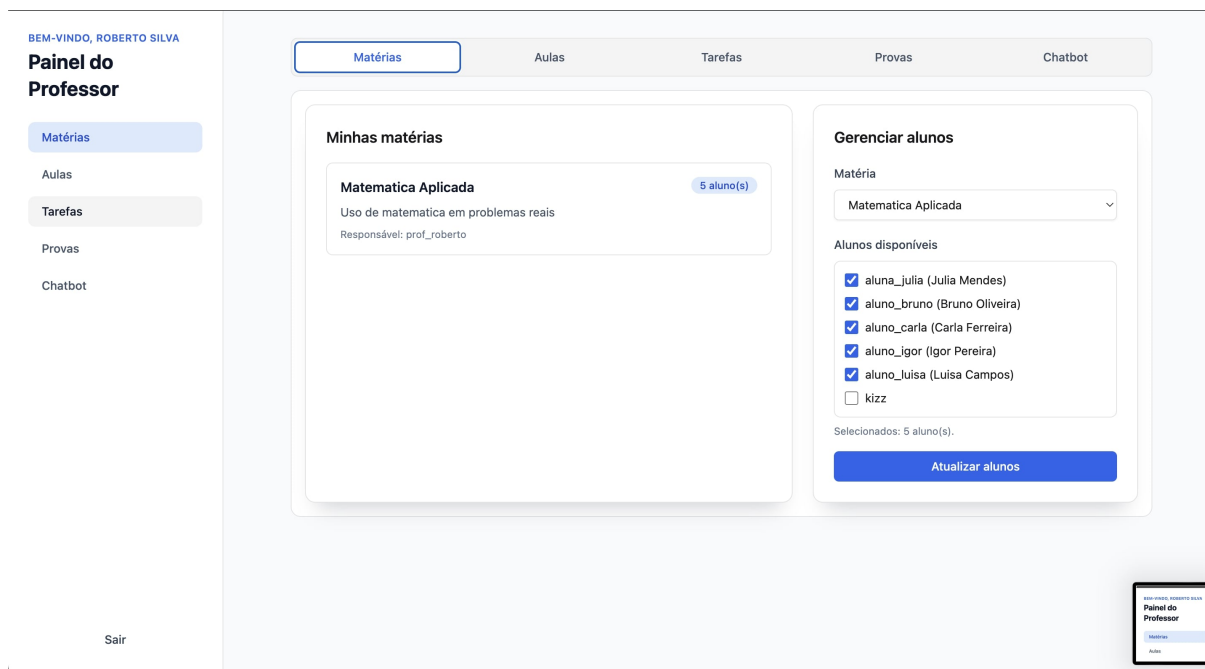
O primeiro conjunto de interações demonstradas corresponde ao uso do sistema pelo professor, responsável por estruturar os dados acadêmicos que posteriormente serão consumidos pelos alunos e pelo assistente inteligente.

Inicialmente, o professor acessa o painel de matérias, onde pode visualizar as disciplinas sob sua responsabilidade e gerenciar a matrícula dos alunos em cada uma delas, conforme ilustrado na Figura 3. Essa etapa estabelece o vínculo entre alunos e matérias, que é essencial para o funcionamento correto das demais funcionalidades.

Em seguida, o professor passa a cadastrar as aulas associadas a uma matéria específica. A Figura 4 apresenta a tela de gerenciamento de aulas, na qual é possível tanto visualizar as aulas já existentes quanto criar novas, informando título, descrição, data e a matéria correspondente.

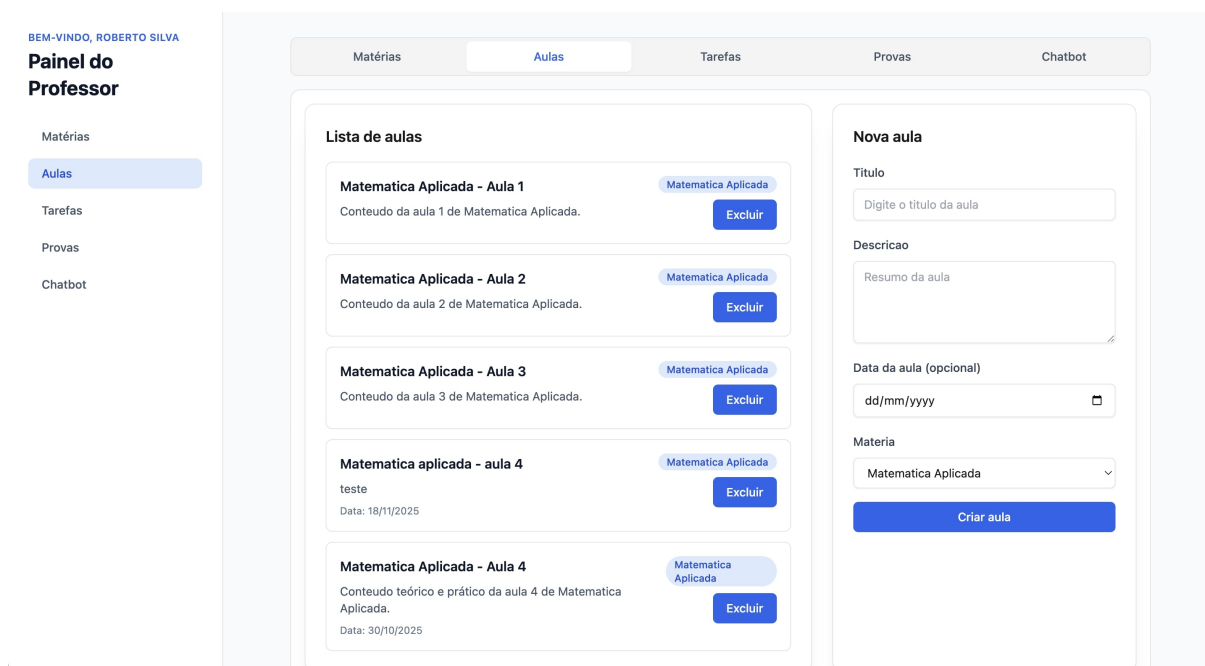
Após a definição das aulas, o professor pode cadastrar as tarefas vinculadas a cada uma delas. A Figura 5 mostra a interface de criação e listagem de tarefas, onde cada tarefa fica explicitamente associada a uma aula.

Figura 3 – Painel de gerenciamento de matérias e matrícula de alunos.



Fonte: Elaboração própria.

Figura 4 – Tela de cadastro e listagem de aulas associadas a uma matéria.

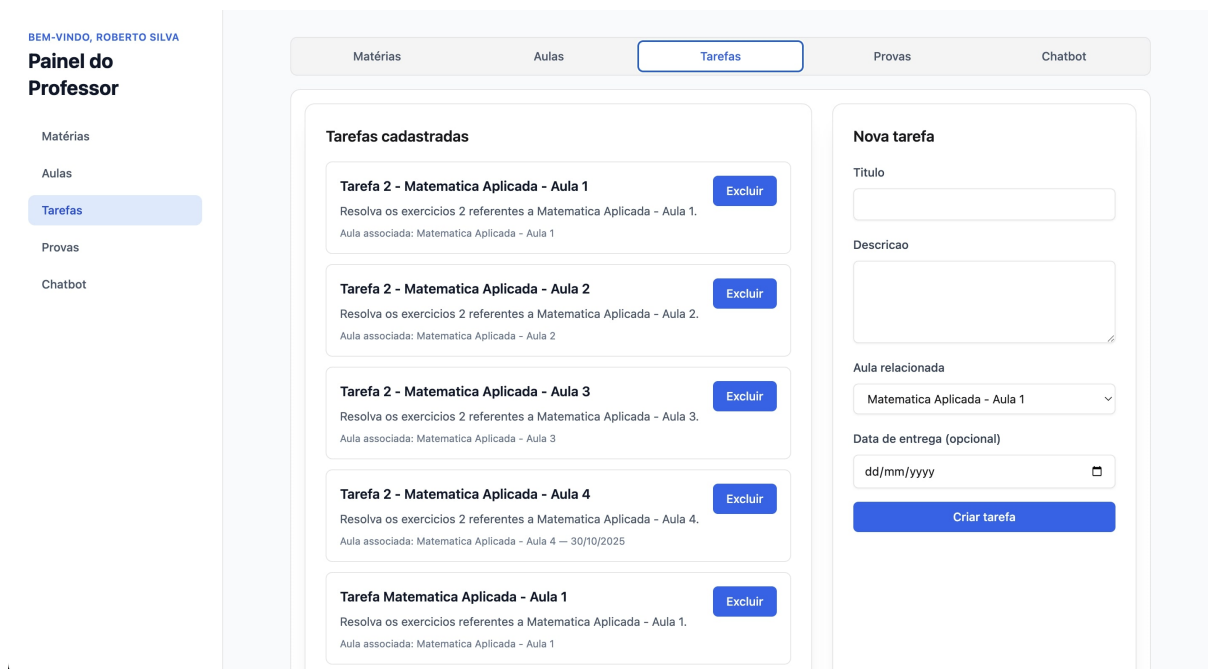


Fonte: Elaboração própria.

O próximo passo consiste no cadastro das provas. Conforme apresentado na Figura 6, o professor pode criar uma nova prova associada a uma matéria, além de vincular a ela aulas e tarefas específicas.

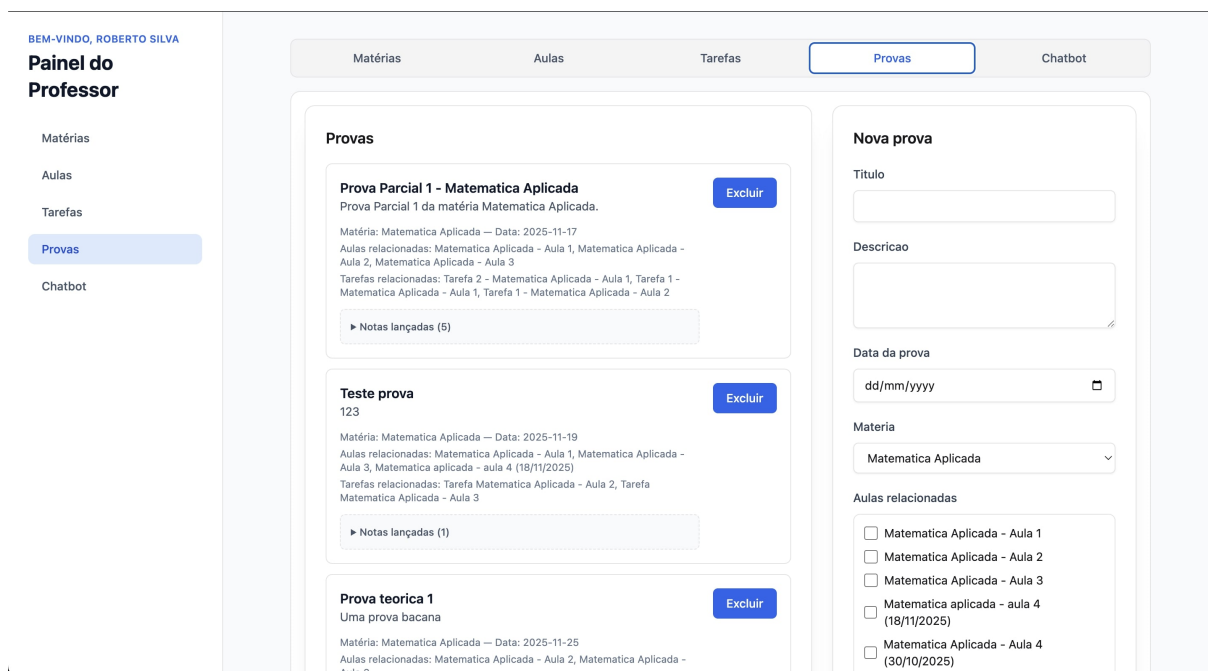
Por fim, o professor pode lançar as notas dos alunos para cada prova cadastrada. A Fi-

Figura 5 – Tela de cadastro e gerenciamento de tarefas associadas às aulas.



Fonte: Elaboração própria.

Figura 6 – Tela de cadastro de provas com vinculação a aulas e tarefas.



Fonte: Elaboração própria.

gura 7 ilustra a interface de lançamento de notas, na qual é possível selecionar a prova, o aluno e registrar a nota obtida, além de um comentário opcional.

Figura 7 – Tela de lançamento e gerenciamento de notas por prova e aluno.

Tarefas relacionadas: Tarefa 2 - Matematica Aplicada - Aula 1, Tarefa 1 - Matematica Aplicada - Aula 1, Tarefa 1 - Matematica Aplicada - Aula 2

▶ Notas lançadas (5)

**Prova Final - Matematica Aplicada** Excluir

Prova Final da matéria Matematica Aplicada.

Matéria: Matematica Aplicada — Data: 2026-01-11

Aulas relacionadas: Matematica Aplicada - Aula 1, Matematica Aplicada - Aula 2, Matematica Aplicada - Aula 3

Tarefas relacionadas: Tarefa 2 - Matematica Aplicada - Aula 1, Tarefa 1 - Matematica Aplicada - Aula 1, Tarefa 1 - Matematica Aplicada - Aula 2

▶ Notas lançadas (5)

Sair

Tarefa 1 - Matematica Aplicada - Aula 1  
Tarefa 1 - Matematica Aplicada - Aula 2  
Tarefa 1 - Matematica Aplicada - Aula 3  
Tarefa 1 - Matematica Aplicada - Aula 4

Criar prova

**Lançar nota**

Prova  
Prova Parcial 1 - Matematica Aplicada

Aluno  
aluna\_julia (Julia Mendes)

Nota (0 a 10)  
6,15

Comentário (opcional)  
Reforçar alguns pontos para melhorar.

Salvar nota Remover nota

Fonte: Elaboração própria.

### 4.7.3 Fluxo de uso pelo aluno

Uma vez que a estrutura acadêmica foi definida pelo professor, o aluno passa a consumir essas informações por meio de seu próprio painel. A Figura 8 apresenta o painel do aluno, que disponibiliza acesso às aulas, tarefas, provas e ao assistente inteligente.

Por meio dessa interface, o aluno pode consultar as tarefas pendentes, acompanhar as provas agendadas e verificar as notas já lançadas. Além disso, o sistema permite que o próprio aluno atualize o status de suas tarefas.

### 4.7.4 Uso do *chatbot* como camada complementar

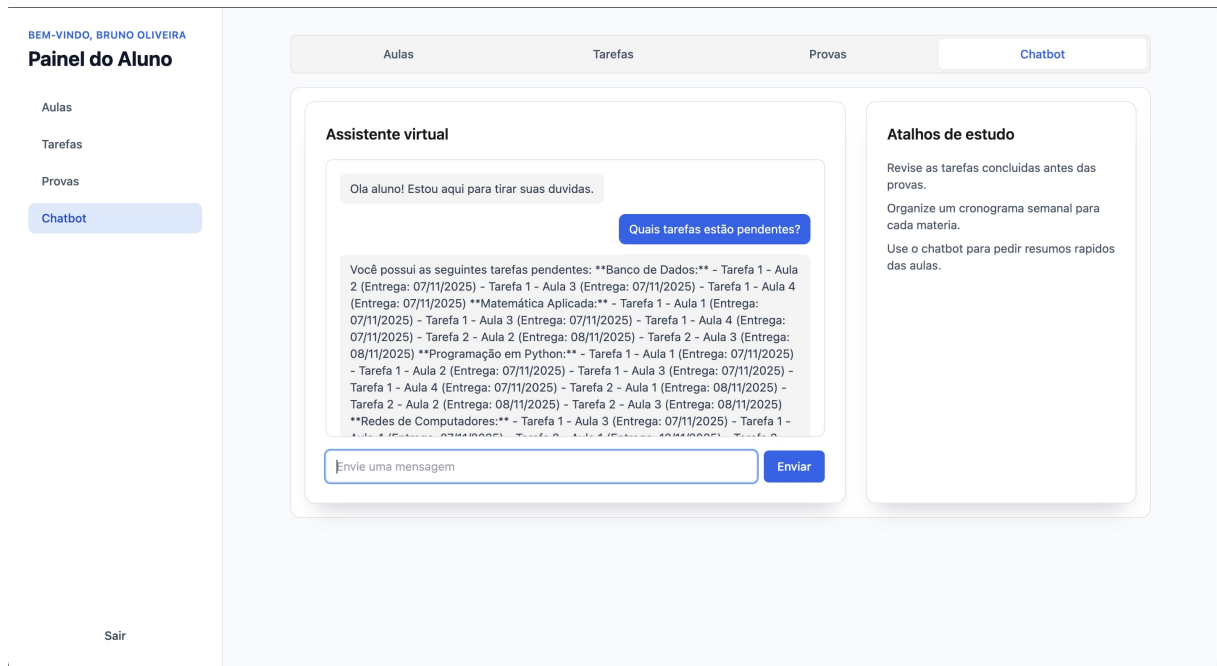
Paralelamente ao uso tradicional da interface web, o sistema disponibiliza um assistente inteligente acessível por meio de uma interface de *chat* integrada ao painel do aluno.

Por meio do *chatbot*, o usuário pode realizar consultas em linguagem natural, como, por exemplo, solicitar a listagem de tarefas pendentes, conforme ilustrado na Figura 9.

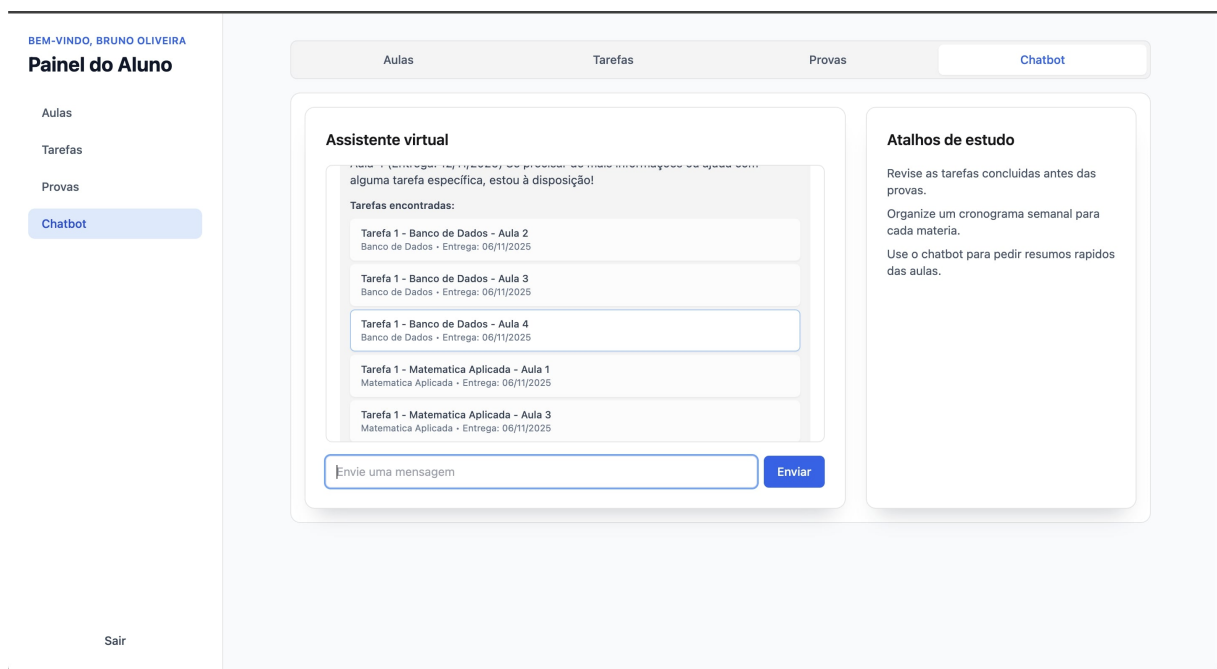
Também é possível consultar qual é a próxima prova agendada, conforme apresentado na Figura 10.

Além de consultas, o assistente também foi projetado para executar ações simples, como a atualização do status de tarefas.

Figura 8 – Painel do aluno com acesso às funcionalidades e ao assistente inteligente.



Fonte: Elaboração própria.

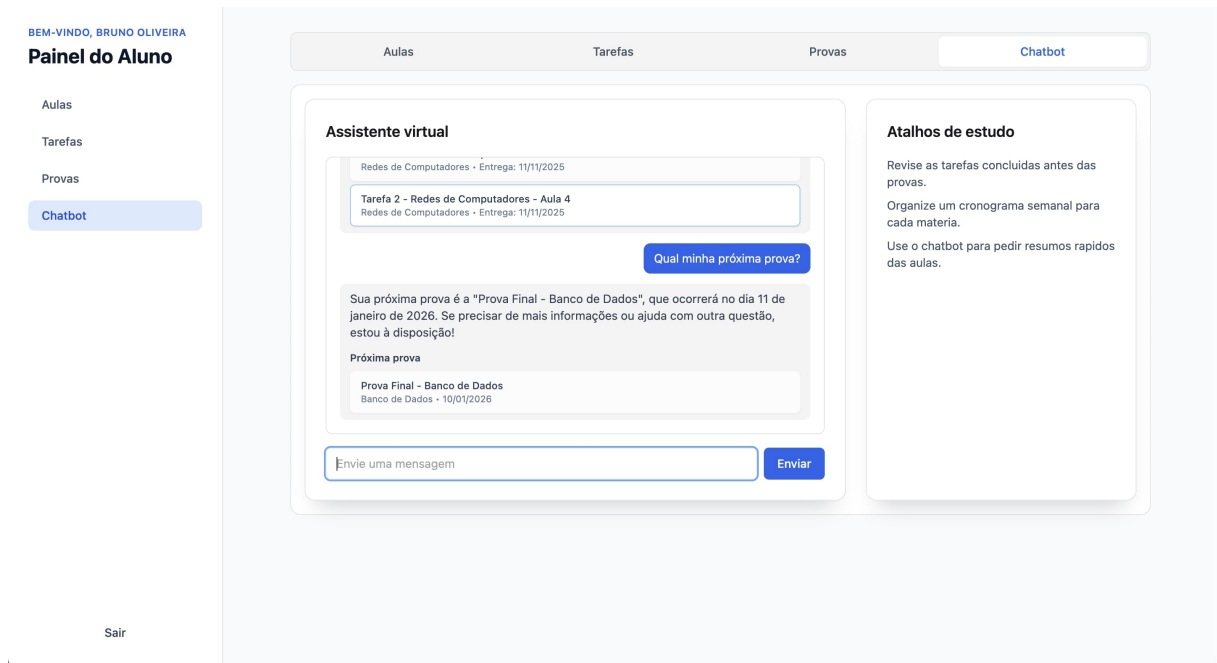
Figura 9 – Consulta de tarefas pendentes realizada por meio do *chatbot*.

Fonte: Elaboração própria.

#### 4.7.5 Demonstração do mecanismo de recomendação

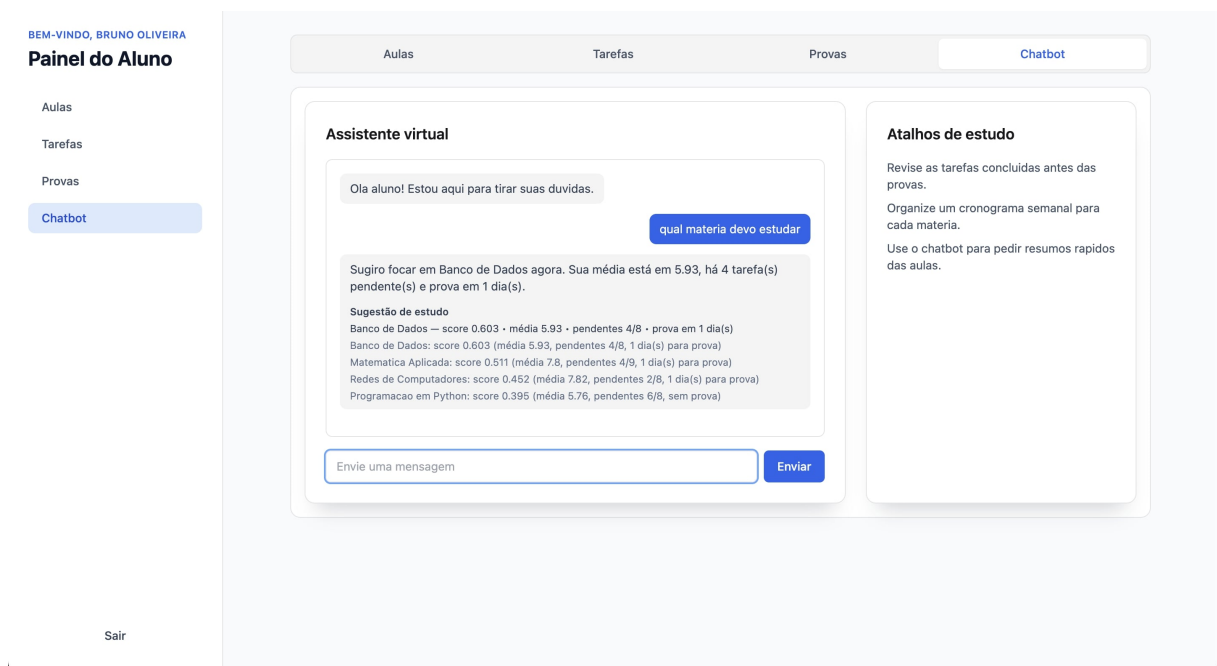
Uma das funcionalidades centrais do assistente inteligente é a recomendação de qual matéria deve ser priorizada para estudo, conforme exemplificado na Figura 11.

Figura 10 – Consulta da próxima prova realizada por meio do *chatbot*.



Fonte: Elaboração própria.

Figura 11 – Recomendação de matéria para estudo gerada pelo assistente inteligente.



Fonte: Elaboração própria.

Esse comportamento é dinâmico e responde às mudanças reais no estado acadêmico do aluno.

## 4.8 Síntese dos resultados

De forma geral, os resultados obtidos indicam que o sistema atinge seu objetivo principal, que é demonstrar a viabilidade arquitetural da integração entre um *LMS* e um assistente inteligente baseado em modelos de linguagem. A validação em cenários práticos mostra que essa integração é tecnicamente possível e funcional, ainda que o sistema não esteja, neste momento, pronto para uso em ambiente produtivo.

Assim, o trabalho cumpre seu papel como uma demonstração conceitual e técnica, servindo como base para futuras evoluções e aprofundamentos na área de sistemas educacionais inteligentes.

## 5 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou o desenvolvimento de um sistema educacional inteligente que integra funcionalidades tradicionais de um *LMS* a uma camada de inteligência artificial baseada em modelos de linguagem, com o objetivo de apoiar o acompanhamento acadêmico e a tomada de decisão dos estudantes.

Ao longo do projeto, foi possível demonstrar que a integração entre os módulos administrativos, o banco de dados acadêmico e o assistente conversacional permite não apenas a consulta de informações, mas também a execução de ações diretamente por meio de linguagem natural, como a atualização de status de tarefas e a obtenção de recomendações personalizadas de estudo. Esse modelo supera a abordagem tradicional de plataformas educacionais passivas, nas quais o usuário precisa navegar manualmente pelas interfaces para interpretar seus próprios dados.

A principal contribuição do sistema está na introdução de um mecanismo de priorização baseado em múltiplos fatores, como desempenho acadêmico, quantidade de pendências e proximidade de avaliações, consolidado em um *score* de recomendação que orienta o aluno sobre quais disciplinas demandam maior atenção. Essa abordagem transforma dados brutos em informações interpretáveis, reduzindo a carga cognitiva do estudante e oferecendo suporte contínuo ao planejamento de estudos.

Do ponto de vista técnico, a arquitetura proposta demonstrou ser consistente, modular e extensível, integrando *backend*, *frontend* e camada de *IA* por meio de uma *API* única, garantindo coerência entre as informações apresentadas nas interfaces gráficas e nas respostas fornecidas pelo assistente inteligente. Os testes realizados em cenários completos de uso validaram a viabilidade da solução e a consistência dos fluxos implementados.

Dessa forma, conclui-se que o sistema atinge os objetivos propostos, demonstrando que a aplicação de técnicas de inteligência artificial em ambientes educacionais pode ampliar significativamente o papel dessas plataformas, que deixam de ser apenas repositórios de conteúdo e passam a atuar como agentes ativos de apoio ao processo de aprendizagem.

### 5.1 Publicações Realizadas

Até o momento da conclusão deste trabalho, não houve a submissão ou publicação de artigos científicos diretamente derivados desta pesquisa. No entanto, os resultados obtidos e a arquitetura proposta apresentam potencial para geração de publicações futuras em eventos e periódicos da área de sistemas educacionais, inteligência artificial aplicada à educação e tecnologias educacionais.

## 5.2 Trabalhos Futuros

Como continuidade deste trabalho, diversas extensões podem ser exploradas para ampliar o impacto e a robustez da solução desenvolvida. Entre as principais possibilidades, destacam-se:

- A incorporação de modelos de aprendizado de máquina para análise preditiva de desempenho, permitindo a identificação antecipada de riscos de reprovação ou evasão.
- A ampliação do sistema de recomendação para considerar fatores adicionais, como histórico de engajamento, tempo de estudo e padrões de comportamento do aluno.
- A implementação de mecanismos de explicabilidade das recomendações, de modo que o sistema possa justificar, em linguagem natural, os motivos de cada sugestão apresentada.
- A realização de estudos de caso com turmas reais, visando avaliar o impacto pedagógico do sistema em cenários de uso prolongado.
- A integração com outras plataformas institucionais e ambientes virtuais já existentes, permitindo maior interoperabilidade e adoção em contextos reais.

Essas extensões podem contribuir para transformar a solução apresentada em uma plataforma ainda mais completa, robusta e alinhada às demandas contemporâneas da educação mediada por tecnologia.

## REFERÊNCIAS

- ANTUNES, C. A. *et al.* Chatbots na educação: uma revisão sistemática da literatura. **RENOTE**, v. 16, n. 1, p. 123–132, 2018. Disponível em: <<https://seer.ufrgs.br/index.php/renote/article/view/86019>>. Citado 2 vezes nas páginas 15 e 19.
- ARAFAT, M. *et al.* Next-generation event-driven architectures: Performance, scalability, and intelligent orchestration across messaging frameworks. **arXiv preprint arXiv:2510.04404**, 2025. Disponível em: <<https://arxiv.org/abs/2510.04404>>. Citado na página 22.
- Blackboard. **Release 3800.0.0 (Build: 3800.0.0-rel.20+111b512)**. [S.l.], 2020. Disponível em: <https://help.blackboard.com/node/31981>. Acesso em: 3 nov. 2025. Citado na página 23.
- BRYNJOLFSSON, E.; LI, D.; RAYMOND, L. R. Generative ai at work. **MIT Sloan Review**, 2023. Disponível em: <<https://www.nber.org/papers/w31161>>. Citado 2 vezes nas páginas 14 e 19.
- CARMO, K. X. **Um estudo comparativo entre tecnologias de back-end: Node.js, Django Rest Framework e ASP.NET Core**. 2023. Disponível em: <<https://repositorio.ufc.br/ri/handle/riufc/77574>>. Citado na página 21.
- DigitalOcean. **Set Up Django with Postgres, Nginx and Gunicorn on Ubuntu**. [S.l.], 2025. Disponível em: <<https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunicorn-on-ubuntu>>. Citado na página 21.
- GARZÓN, J. Systematic review of artificial intelligence in education. **Multimodal Technologies and Interaction**, v. 9, n. 8, p. 84, 2025. Disponível em: <<https://www.mdpi.com/2414-4088/9/8/84>>. Citado 2 vezes nas páginas 14 e 17.
- HSUEH, H. *et al.* Development and evaluation of learning portfolio query interface using langchain. **MDPI**, 2025. Disponível em: <<https://www.mdpi.com/2673-4591/92/1/40>>. Citado 3 vezes nas páginas 15, 20 e 23.
- Instructure. **Canvas Architecture Overview**. Salt Lake City, 2021. Disponível em: <https://www.instructure.com/sites/default/files/file/2021-01/Canvas-Architecture-Jan-2021.pdf>. Acesso em: 3 nov. 2025. Citado na página 23.
- JAIN, A.; NANDAN, M.; BHOSALE, R. Comparative analysis of frequently used e-learning platforms. **Frontiers in Education**, v. 9, 2024. Disponível em: <<https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2024.1431531/full>>. Citado na página 17.
- LI, Q.; LIU, Y. An intelligent educational system: Analyzing student behavior and academic performance using multi-source data. **Electronics (MDPI)**, v. 14, 2025. Disponível em: <<https://www.mdpi.com/2079-9292/14/16/3328>>. Citado 4 vezes nas páginas 14, 15, 17 e 23.
- Moodle. **PHP for Moodle**. [S.l.], 2025. Disponível em: <https://docs.moodle.org/en/PHP>. Acesso em: 3 nov. 2025. Citado na página 22.
- QUISI-PERALTA, F. *et al.* Intelligent educational agent for education support using large language models through langchain. **ResearchGate**, 2024. Disponível em: <[https://www.researchgate.net/publication/378512861\\_Intelligent\\_Educational\\_Agent\\_for\\_Education\\_Support\\_Using\\_Long\\_Language\\_Models\\_Through\\_Langchain](https://www.researchgate.net/publication/378512861_Intelligent_Educational_Agent_for_Education_Support_Using_Long_Language_Models_Through_Langchain)>. Citado 2 vezes nas páginas 14 e 21.

SANCHEZ, L.; PENARRETA, J.; POMA, X. S. Learning management systems for higher education: a brief comparison. **Discover Education**, v. 3, p. 58, 2024. Disponível em: <<https://link.springer.com/article/10.1007/s44217-024-00143-5>>. Citado na página 17.

UNESCO. **Education: From Disruption to Recovery**. 2020. Acesso em: 3 nov. 2025. Disponível em: <<https://www.unesco.org/en/covid-19/education-response>>. Citado na página 14.

VOREAKOU, C. *et al.* Exploring energy consumption efficiency of open source messaging systems. **arXiv preprint arXiv:2506.10693**, 2025. Disponível em: <<https://arxiv.org/abs/2506.10693>>. Citado na página 22.

YAN, S. *et al.* Practical and ethical challenges of large language models in education: A systematic scoping review. **arXiv preprint arXiv:2303.13379**, 2023. Disponível em: <<https://arxiv.org/abs/2303.13379>>. Citado 2 vezes nas páginas 15 e 20.