UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

GUILHERME CERVI SCHWINGEL

# POTENCIALIZANDO O DIRECIONAMENTO EDUCACIONAL: UM SISTEMA BASEADO EM INTELIGÊNCIA ARTIFICIAL PARA OTIMIZAÇÃO DA JORNADA DE APRENDIZAGEM

Porto Alegre
2025

GUILHERME CERVI SCHWINGEL

# POTENCIALIZANDO O DIRECIONAMENTO EDUCACIONAL: UM SISTEMA BASEADO EM INTELIGÊNCIA ARTIFICIAL PARA OTIMIZAÇÃO DA JORNADA DE APRENDIZAGEM

Trabalho de Conclusão de Curso de Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul para a obtenção do título de Engenheiro Eletricista.
Orientador: Prof. Dr. Renato Ventura Bayan Henriques

Porto Alegre
2025

CIP - Catalogação na Publicação

Schwingel, Guilherme
POTENCIALIZANDO O DIRECIONAMENTO EDUCACIONAL: UM SISTEMA BASEADO EM INTELIGÊNCIA ARTIFICIAL PARA OTIMIZAÇÃO DA JORNADA DE APRENDIZAGEM / Guilherme Schwingel. -- 2025.
99 f.
Orientador: Renato Ventura Bayan Henriques.

Trabalho de conclusão de curso (Graduação) -- Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Curso de Engenharia Elétrica, Porto Alegre, BR-RS, 2025.

1. Inteligência Articicial. 2. Aprendizagem de Máquina. 3. Banco de Dados. 4. Processamento de Linguagem Natural. 5. Microsserviços. I. Ventura Bayan Henriques, Renato, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

GUILHERME CERVI SCHWINGEL

**POTENCIALIZANDO O DIRECIONAMENTO EDUCACIONAL: UM SISTEMA BASEADO EM INTELIGÊNCIA ARTIFICIAL PARA OTIMIZAÇÃO DA JORNADA DE APRENDIZAGEM**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de "Engenheiro Eletricista" e aprovado em sua forma final pelo Curso de Graduação em Engenharia Elétrica.

Porto Alegre, 18 de Julho de 2025.

**Banca Examinadora:**

gov.br
Documento assinado digitalmente
RENATO VENTURA BAYAN HENRIQUES
Data: 20/07/2025 19:40:50-0300
Verifique em https://validar.iti.gov.br

---

Prof. Dr. Renato Ventura Bayan Henriques
Doutor pela Universidade Federal de Minas Gerais

---

Prof. Dr. Ivan Müller
Doutor pela Universidade Federal do Rio Grande do Sul

---

Dr. Carlos Solon Guimarães Junior
Doutor pela Universidade Federal do Rio Grande do Sul

## AGRADECIMENTOS

Aos meus pais, pelo apoio incondicional durante toda minha trajetória acadêmica e por sempre acreditarem em meus sonhos.

Ao meu orientador, Prof. Dr. Renato Ventura Bayan Henriques, pela paciência, dedicação e valiosas contribuições que tornaram este trabalho possível.

Ao Professor Weslei Huppes Compagnoni e à Escola Sinodal de São Leopoldo, pela parceria fundamental na coleta de dados e validação do sistema com os alunos do sexto ano.

Aos colegas do curso de Engenharia Elétrica, pelas discussões enriquecedoras e pelo companheirismo ao longo desta jornada.

Aos 44 estudantes que participaram da pesquisa, compartilhando suas aspirações e contribuindo para a construção de uma ferramenta que poderá beneficiar muitos outros jovens.

À Universidade Federal do Rio Grande do Sul, por proporcionar um ambiente de excelência acadêmica e oportunidades de crescimento pessoal e profissional.

A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

“Cada sonho que você deixa para trás, é um pedaço do futuro que deixa de existir.”

**Steve Jobs**

## RESUMO

Este trabalho apresenta o desenvolvimento de uma plataforma web educacional que integra processamento de linguagem natural e elementos de gamificação para auxiliar jovens brasileiros entre 8 e 17 anos na identificação de seus interesses educacionais e profissionais. A pesquisa fundamentou-se na problemática da evasão escolar no Brasil, que afeta 9,2 milhões de jovens, e na dificuldade dos adolescentes em reconhecer suas aptidões em meio à complexidade de escolhas contemporâneas. A metodologia adotada caracterizou-se como pesquisa aplicada quali-quantitativa, combinando desenvolvimento de software com análise empírica. Foi construído um dataset especializado com 13.500 exemplos através de metodologia híbrida, utilizado para adaptar o modelo BERTimbau Large à linguagem juvenil brasileira. O sistema implementado integra arquitetura baseada em microsserviços, combinando BigQuery, Cloud Functions, FastAPI, Google Firestore e React, com design pedagógico estruturado em nove áreas de conhecimento. A validação qualitativa com 44 alunos do sexto ano revelou características linguísticas fundamentais: 83,8% das respostas continham variações ortográficas não-padrão, 64,9% expressavam múltiplos interesses simultaneamente, e o vocabulário refletia influências da cultura digital contemporânea. A análise demonstrou que a integração entre questionário estruturado e processamento de texto livre permite capturar nuances difíceis de identificar em sistemas unidimensionais, como distinguir interesse recreativo de aspirações profissionais. Os resultados evidenciam a importância de considerar as especificidades da comunicação juvenil no desenvolvimento de ferramentas educacionais, contribuindo para democratização do acesso à orientação vocacional personalizada.



**Palavras-chave**: Processamento de linguagem natural. Inteligência Artificial. Aprendizagem de Máquina. Microsserviços. Banco de Dados

## ABSTRACT

This work presents the development of an educational web platform that integrates natural language processing and gamification elements to assist Brazilian people aged 8 to 17 in identifying their educational and professional interests. The research addressed the problem of school dropout in Brazil, affecting 9.2 million young people, and adolescents' difficulty in recognizing their aptitudes amid contemporary choice complexity. The methodology adopted was characterized as quali-quantitative applied research, combining software development with empirical analysis. A specialized dataset with 13,500 examples was built through hybrid methodology, used to adapt the BERTimbau Large model to Brazilian youth language. The implemented system integrates microservices-based architecture, combining BigQuery, Cloud Functions, FastAPI, Google Firestore, and React, with pedagogical design structured across nine knowledge areas. Qualitative validation with 44 sixth-grade students revealed fundamental linguistic characteristics: 83.8% of responses contained non-standard orthographic variations, 64.9% expressed multiple interests simultaneously, and vocabulary reflected contemporary digital culture influences. The analysis demonstrated that integration between structured questionnaire and free text processing captures nuances difficult to identify in one-dimensional systems, such as distinguishing recreational interest from professional aspirations. Results highlight the importance of considering youth communication specificities in developing educational tools, contributing to democratizing access to personalized vocational guidance.



**Keywords**: Natural Language Processing. Artificial Intelligence. Machine Learning. Microservices. Database

## LISTA DE FIGURAS


Figura 1 – Diagrama C4 de Contêineres mostrando a arquitetura do sistema educacional gamificado na Google Cloud Platform. . . . . 42
Figura 2 – Mapa mental da estrutura modular da API mostrando as quatro categorias funcionais e distribuição dos 112 endpoints. . . . . 43
Figura 3 – Estrutura hierárquica detalhada dos principais endpoints da API organizados por módulo funcional. . . . . 43
Figura 4 – Diagrama de classes dos principais modelos de dados Pydantic com validações e relacionamentos. . . . . 44
Figura 5 – *Pipeline* de dados ETL mostrando o fluxo desde eventos transacionais no Firestore até dados analíticos no BigQuery. . . . . 45
Figura 6 – Camadas de segurança implementadas no sistema. . . . . 46
Figura 7 – Estrutura hierárquica completa do sistema pedagógico mostrando a progressão desde áreas até atividades específicas. . . . . 48
Figura 8 – Anatomia detalhada de uma lição típica mostrando todos os componentes pedagógicos. . . . . 50
Figura 9 – Ecossistema de recursos de aprendizagem organizados por tipo e nível de dificuldade. . . . . 51
Figura 10 – Processo de construção do *dataset* em três fases. . . . . 52
Figura 11 – Arquitetura de fine-tuning do BERTimbau para classificação. . . . . 56
Figura 12 – Mapeamento entre elementos de gamificação e necessidades psicológicas da SDT. . . . . 60
Figura 13 – Arquitetura da camada de abstração para integração com LLMs. . . . . 62
Figura 14 – Processo iterativo de engenharia de *prompts* para conteúdo educacional. 63
Figura 15 – Sistema de *breakpoints* responsivos e adaptação da interface. . . . . 66
Figura 16 – Sistema de navegação adaptativa em diferentes contextos. . . . . 68
Figura 17 – Hierarquia de componentes e suas relações. . . . . 70
Figura 18 – Tela inicial . . . . . 72
Figura 19 – Interface de login do sistema . . . . . 73
Figura 20 – Formulário de cadastro com campos de personalização . . . . . 74
Figura 21 – Tela de introdução ao mapeamento de interesses . . . . . 75
Figura 22 – Exemplo de questão contextualizada do mapeamento . . . . . 75
Figura 23 – Interface de entrada de texto livre . . . . . 76
Figura 24 – Dashboard principal do sistema . . . . . 77
Figura 25 – Grade de áreas de conhecimento disponíveis . . . . . 78
Figura 26 – Interface de seleção de subárea . . . . . 79
Figura 27 – Lista de recursos educacionais categorizados . . . . . 80
Figura 28 – Estrutura de navegação do curso . . . . . 81

Figura 29 – Interface de apresentação de conteúdo educacional . . . . . 82
Figura 30 – Matriz de confusão do modelo BERTimbau no *dataset* sintético de teste 84
Figura 31 – Curvas ROC para todas as categorias no *dataset* sintético de teste . . . 85
Figura 32 – Matriz de confusão do modelo BERTimbau no *dataset* real de alunos . 87

## LISTA DE TABELAS

Tabela 1 – Características das integrações de IA . . . . . 45
Tabela 2 – Hierarquia de cache e características de cada camada . . . . . 47
Tabela 3 – Áreas e subáreas do sistema de trilhas de aprendizado . . . . . 47
Tabela 4 – Características distintivas dos três níveis de proficiência . . . . . 49
Tabela 5 – Progressão da complexidade de projetos através dos níveis . . . . . 51
Tabela 6 – Exemplos de termos mapeados por categoria . . . . . 53
Tabela 7 – Processo de expansão e balanceamento final . . . . . 53
Tabela 8 – Exemplos representativos do *dataset* final . . . . . 54
Tabela 9 – Distribuição de XP por tipo de atividade . . . . . 61
Tabela 10 – Estilos de ensino disponíveis e suas características . . . . . 64
Tabela 11 – Configuração hierárquica do sistema de cache . . . . . 65
Tabela 12 – Sistema de cores e seus significados funcionais . . . . . 67
Tabela 13 – Evolução das métricas durante o treinamento do modelo BERTimbau . 83
Tabela 14 – Características linguísticas e temáticas das respostas dos alunos . . . . 86
Tabela 15 – Análise detalhada dos casos de discordância entre rótulo humano e predição . . . . . 88
Tabela 16 – Exemplos representativos de respostas corretamente classificadas . . . 88
Tabela 17 – Estrutura detalhada do questionário de mapeamento de interesses . . . 89
Tabela 18 – Distribuição de respostas por questão e categoria de interesse . . . . . 90
Tabela 19 – Casos de mudança de classificação entre análise textual e sistema híbrido 91
Tabela 20 – Custos dos modelos mais econômicos por provedor . . . . . 93

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| AI | *Artificial Intelligence* |
| AM | Aprendizado de Máquina |
| API | *Application Programming Interface* |
| ASGI | *Asynchronous Server Gateway Interface* |
| AUC | *Area Under the Curve* |
| BERT | *Bidirectional Encoder Representations from Transformers* |
| CBOW | *Continuous Bag of Words* |
| CDN | *Content Delivery Network* |
| ChatGPT | *Chat Generative Pre-trained Transformer* |
| CORS | *Cross-Origin Resource Sharing* |
| CSS | *Cascading Style Sheets* |
| DOM | *Document Object Model* |
| ETL | *Extract, Transform, Load* |
| FUNDEB | Fundo de Manutenção e Desenvolvimento da Educação Básica |
| GCP | *Google Cloud Platform* |
| GloVe | *Global Vectors for Word Representation* |
| GPT-4 | *Generative Pre-trained Transformer 4* |
| GPU | *Graphics Processing Unit* |
| GRU | *Gated Recurrent Units* |
| IA | Inteligência Artificial |
| IDEB | Índice de Desenvolvimento da Educação Básica |
| INEP | Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira |
| JSON | *JavaScript Object Notation* |
| JWT | *JSON Web Token* |
| LLM | *Large Language Model* |
| LRU | *Least Recently Used* |
| LSTM | *Long Short-Term Memory* |
| MD5 | *Message Digest Algorithm 5* |
| ML | *Machine Learning* |
| MLM | *Masked Language Modeling* |
| NoSQL | *Not Only SQL* |
| NSP | *Next Sentence Prediction* |
| OAuth | *Open Authorization* |
| PLN | Processamento de Linguagem Natural |
| PNAD | Pesquisa Nacional por Amostra de Domicílios |
| RBAC | *Role-Based Access Control* |
| ReLU | *Rectified Linear Unit* |

| | |
|---|---|
| REST | *Representational State Transfer* |
| RESTful | *REST-compliant* |
| RNN | *Recurrent Neural Network* |
| ROC | *Receiver Operating Characteristic* |
| SDT | *Self-Determination Theory* |
| SQL | *Structured Query Language* |
| TPU | *Tensor Processing Unit* |
| TTL | *Time To Live* |
| UI | *User Interface* |
| URI | *Uniform Resource Identifier* |
| VRAM | *Video Random Access Memory* |
| WCAG | *Web Content Accessibility Guidelines* |
| XP | *Experience Points* |
| XSS | *Cross-Site Scripting* |

## LISTA DE SÍMBOLOS

| | |
|---|---|
| $\mathcal{L}$ | Função de perda |
| $\eta$ | Taxa de aprendizado |
| $\theta$ | Parâmetros do modelo |
| $D$ | Conjunto de dados |
| $n$ | Número de amostras |
| $c$ | Tamanho da janela de contexto |
| $d_k$ | Dimensão das chaves no mecanismo de atenção |
| $d_{model}$ | Dimensão do modelo |
| $h$ | Número de cabeças de atenção ou hipótese |
| $L$ | Número de camadas |
| $Q$ | Queries no mecanismo de atenção |
| $K$ | Keys no mecanismo de atenção |
| $V$ | Values no mecanismo de atenção |
| $W$ | Matriz de pesos |
| $b$ | Vetor de viés |
| $x$ | Entrada ou exemplo |
| $y$ | Saída ou rótulo |
| $\hat{y}$ | Predição do modelo |
| $f$ | Função de ativação ou mapeamento |
| $X$ | Espaço de entrada |
| $Y$ | Espaço de saída |
| $P$ | Probabilidade ou distribuição |
| $E$ | Valor esperado ou experiência |
| $T$ | Tamanho do corpus ou conjunto de tarefas |
| $C$ | Número de classes |
| $N$ | Número de amostras no batch |
| $\vec{v}$ | Vetor de representação |
| $\mathbb{E}$ | Esperança matemática |
| min | Operador de minimização |
| max | Operador de maximização |
| $\sum$ | Somatório |
| $\approx$ | Aproximadamente igual |
| $\in$ | Pertence a |
| $\rightarrow$ | Mapeia para |
| $\setminus$ | Exceto |

**SUMÁRIO**


| | | |
|---|---|---|
| **1** | **INTRODUÇÃO** | **19** |
| 1.1 | OBJETIVOS | 19 |
| 1.2 | JUSTIFICATIVA | 20 |
| 1.3 | CONCEITOS FUNDAMENTAIS | 20 |
| 1.4 | ABORDAGEM METODOLÓGICA | 21 |
| 1.5 | ORGANIZAÇÃO DO DOCUMENTO | 21 |
| **2** | **REFERENCIAL TEÓRICO** | **23** |
| 2.1 | INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINA | 23 |
| **2.1.1** | **Fundamentos da Inteligência Artificial** | **23** |
| **2.1.2** | **Paradigmas de Aprendizado de Máquina** | **23** |
| 2.2 | REDES NEURAIS ARTIFICIAIS | 25 |
| **2.2.1** | **Fundamentos Matemáticos e Arquitetura** | **25** |
| **2.2.2** | **Treinamento via Retropropagação** | **26** |
| 2.3 | MÉTRICAS DE AVALIAÇÃO PARA CLASSIFICAÇÃO | 27 |
| **2.3.1** | **Matriz de Confusão e Métricas Derivadas** | **27** |
| 2.4 | PROCESSAMENTO DE LINGUAGEM NATURAL | 29 |
| **2.4.1** | **Evolução Histórica e Paradigmas** | **29** |
| **2.4.2** | **Representação Vetorial de Palavras** | **29** |
| 2.5 | ARQUITETURA TRANSFORMER E MODELOS DE LINGUAGEM PRÉ-TREINADOS | 31 |
| **2.5.1** | **A Revolução do Transformer** | **31** |
| **2.5.2** | **BERT e a Era dos Modelos Pré-treinados** | **32** |
| 2.6 | REACT E DESENVOLVIMENTO FRONTEND MODERNO | 34 |
| **2.6.1** | **Paradigma Declarativo e Componentização** | **34** |
| 2.7 | DESENVOLVIMENTO DE APIS E ARQUITETURAS MODERNAS | 35 |
| **2.7.1** | **APIs RESTful e FastAPI** | **35** |
| **2.7.2** | **Arquitetura de Microsserviços** | **36** |
| **2.7.3** | **Programação Orientada a Eventos** | **36** |
| 2.8 | COMPUTAÇÃO EM NUVEM E ARMAZENAMENTO DE DADOS | 37 |
| **2.8.1** | **Google Cloud Platform e Bancos NoSQL** | **37** |
| **2.8.2** | **Cloud Firestore: Evolução dos Bancos NoSQL** | **38** |
| 2.9 | CONTEXTO EDUCACIONAL BRASILEIRO | 39 |
| **2.9.1** | **Sistema Nacional de Educação: Estrutura e Desafios** | **39** |
| **2.9.2** | **Desafios da Orientação Vocacional no Brasil** | **39** |
| 2.10 | CONSIDERAÇÕES ÉTICAS EM IA EDUCACIONAL | 40 |
| **2.10.1** | **Privacidade e Proteção de Dados** | **40** |
| **3** | **ESTUDO DE CASO** | **41** |

3.1 CLASSIFICAÇÃO E ABORDAGEM DA PESQUISA . . . . . 41
**3.1.1 Caracterização da Pesquisa . . . . . 41**
3.2 ARQUITETURA GERAL DO SISTEMA . . . . . 41
**3.2.1 Visão Geral da Solução . . . . . 41**
**3.2.2 Implementação do Backend com FastAPI . . . . . 43**
**3.2.3 Modelos de Dados e Validação . . . . . 44**
**3.2.4 *Pipeline* de Processamento de Dados . . . . . 44**
**3.2.5 Integração com Inteligência Artificial . . . . . 45**
**3.2.6 Segurança e Autenticação . . . . . 46**
**3.2.7 Estratégias de Cache e Otimização . . . . . 46**
3.3 ESTRUTURA PEDAGÓGICA DAS TRILHAS DE APRENDIZADO 47
**3.3.1 Arquitetura Hierárquica do Sistema Educacional . . . . . 47**
**3.3.2 Caracterização das Áreas de Conhecimento . . . . . 48**
**3.3.3 Estruturação por Níveis de Proficiência . . . . . 48**
**3.3.4 Organização Modular e Estrutura das Lições . . . . . 49**
**3.3.5 Sistema de Exercícios e Atividades Práticas . . . . . 49**
**3.3.6 Projetos como Eixo Central da Aprendizagem . . . . . 49**
**3.3.7 Sistema de Recursos e Materiais de Apoio . . . . . 50**
3.4 CONSTRUÇÃO DO *DATASET* DE TREINAMENTO . . . . . 50
**3.4.1 Desafios e Motivação . . . . . 50**
**3.4.2 Metodologia de Criação: Abordagem Híbrida . . . . . 51**
*3.4.2.1 Fase 1: Mapeamento Semântico e Curadoria Manual . . . . . 52*
*3.4.2.2 Fase 2: Sistema de Padrões Linguísticos . . . . . 52*
*3.4.2.3 Fase 3: Augmentação via IA Generativa . . . . . 53*
**3.4.3 Características do *Dataset* Final . . . . . 54**
3.5 DESENVOLVIMENTO DO MODELO DE CLASSIFICAÇÃO DE TEXTO LIVRE . . . . . 54
**3.5.1 Seleção da Arquitetura: BERTimbau . . . . . 54**
**3.5.2 Adaptação e Ajuste fino . . . . . 55**
**3.5.3 *Pipeline* de Pré-processamento Especializado . . . . . 55**
**3.5.4 Configuração Experimental e Treinamento . . . . . 57**
3.6 SISTEMA HÍBRIDO DE MAPEAMENTO DE INTERESSES . . . . 58
**3.6.1 Motivação e Arquitetura Conceitual . . . . . 58**
**3.6.2 Design do Questionário Estruturado . . . . . 58**
**3.6.3 Algoritmo de Fusão Inteligente . . . . . 59**
3.7 SISTEMA DE GAMIFICAÇÃO E ENGAJAMENTO . . . . . 59
**3.7.1 Fundamentos da Gamificação implementada . . . . . 59**
**3.7.2 Sistema de Pontos e Níveis . . . . . 61**
3.8 INTEGRAÇÃO COM MODELOS DE LINGUAGEM (LLMS) . . . . 61

**3.8.1 Arquitetura de Integração com LLM . . . 61**
**3.8.2 Geração de Conteúdo Educacional Personalizado . . . 62**
**3.8.3 Estilos de Ensino Implementados . . . 63**
**3.8.4 Funcionalidades Pedagógicas Especializadas . . . 64**
**3.8.5 Sistema de Cache Hierárquico . . . 64**
**3.8.6 Tratamento de Erros e Resiliência . . . 65**
3.9 DESENVOLVIMENTO DO SISTEMA FRONTEND . . . 65
**3.9.1 Interface Responsiva com Material Design . . . 65**
**3.9.2 Sistema de Navegação Adaptativo . . . 67**
**3.9.3 Componentização e Reutilização . . . 69**
**4 RESULTADOS E DISCUSSÕES . . . 71**
4.1 VISÃO GERAL DOS RESULTADOS . . . 71
4.2 INTERFACE PRINCIPAL E AUTENTICAÇÃO . . . 71
**4.2.1 Tela Inicial . . . 71**
**4.2.2 Sistema de Login e Cadastro . . . 71**
4.3 MÓDULO DE MAPEAMENTO DE INTERESSES . . . 73
**4.3.1 Introdução ao Processo de Mapeamento . . . 73**
**4.3.2 Questionário de Múltipla Escolha . . . 75**
**4.3.3 Campo de Texto Livre . . . 76**
4.4 DASHBOARD E NAVEGAÇÃO PRINCIPAL . . . 76
**4.4.1 Estrutura do Dashboard . . . 76**
4.5 SELEÇÃO DE TRILHAS EDUCACIONAIS . . . 78
**4.5.1 Escolha de Área de Conhecimento . . . 78**
**4.5.2 Seleção de Subárea . . . 78**
4.6 RECURSOS EDUCACIONAIS E CONTEÚDO . . . 79
**4.6.1 Interface de Recursos . . . 79**
**4.6.2 Estrutura de Curso e Conteúdo . . . 79**
4.7 RESULTADOS DO MODELO DE CLASSIFICAÇÃO . . . 83
**4.7.1 Desempenho do BERTimbau Fine-tuned . . . 83**
**4.7.2 Avaliação no *Dataset* Sintético . . . 83**
**4.7.3 Análise Qualitativa dos Dados com Usuários Reais . . . 85**
*4.7.3.1 Processo de Rotulagem das Respostas . . . 85*
*4.7.3.2 Características Linguísticas Identificadas . . . 86*
*4.7.3.3 Desempenho do Modelo em Dados Reais . . . 86*
*4.7.3.4 Exemplos Representativos por Categoria . . . 87*
4.8 ANÁLISE DO SISTEMA HÍBRIDO DE MAPEAMENTO . . . 88
**4.8.1 Análise Detalhada do Questionário Estruturado . . . 88**
*4.8.1.1 Estrutura e Racionalidade das Questões . . . 89*
*4.8.1.2 Análise das Respostas por Questão . . . 89*

*4.8.1.3 Desafios e Limitações Identificadas . . . . . 90*
**4.8.2 Integração com Análise de Texto Livre . . . . . 90**
*4.8.2.1 Análise de Casos de Complementaridade . . . . . 90*
*4.8.2.2 Padrões de Reclassificação . . . . . 91*
4.9 ANÁLISE DE CUSTOS DA INTEGRAÇÃO COM LLMS . . . . . 92
**4.9.1 Comparação de Custos entre Provedores . . . . . 92**
**5 CONCLUSÃO . . . . . 94**
**REFERÊNCIAS . . . . . 97**

# 1 INTRODUÇÃO

A transição da adolescência para a vida adulta constitui período de decisões fundamentais que moldam trajetórias educacionais e profissionais. No contexto brasileiro, esta fase crítica confronta-se com desafios estruturais que comprometem o desenvolvimento pleno dos jovens. Entre estes desafios, destaca-se a dificuldade em conectar interesses pessoais autênticos com oportunidades educacionais concretas, lacuna que contribui significativamente para os alarmantes índices de evasão escolar que afetam 9,2 milhões de jovens entre 15 e 29 anos (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2023).

O problema investigado neste trabalho centra-se na inadequação dos métodos tradicionais de orientação vocacional frente às características comunicativas e culturais dos adolescentes contemporâneos. Sistemas convencionais, baseados em questionários padronizados e linguagem formal, falham em captar a riqueza e complexidade das expressões juvenis, permeadas por influências digitais, variações linguísticas e múltiplas identidades em construção. Esta desconexão entre ferramentas disponíveis e a realidade vivida pelos jovens perpetua ciclos de desorientação e desengajamento educacional.

## 1.1 OBJETIVOS

O objetivo principal deste trabalho consiste em desenvolver e validar um sistema educacional web que utilize processamento de linguagem natural adaptado às características específicas da comunicação juvenil brasileira, integrado com elementos de gamificação, para auxiliar adolescentes na identificação e exploração de seus interesses acadêmicos e profissionais.

Os objetivos específicos compreendem:

- Investigar e documentar padrões linguísticos característicos da expressão juvenil brasileira em contextos de autodeclaração de interesses;
- Desenvolver metodologia para construção de dataset que capture autenticidade e diversidade das formas de expressão adolescente;
- Adaptar modelo de linguagem estado da arte para processar efetivamente variações ortográficas, gírias e construções não-padrão;
- Criar sistema híbrido que combine análise estruturada e processamento de texto livre para mapeamento nuançado de interesses;
- Projetar interface e experiência de usuário alinhadas com expectativas e padrões de uso do público-alvo;
- Analisar qualitativamente a eficácia do sistema em contexto escolar real.

## 1.2 JUSTIFICATIVA

A relevância desta pesquisa manifesta-se em múltiplas dimensões interconectadas. Na dimensão social, o trabalho responde à urgência de ferramentas que auxiliem jovens em momentos decisivos de suas trajetórias educacionais. A democratização do acesso a orientação de qualidade torna-se imperativa em país marcado por profundas desigualdades, onde recursos de orientação vocacional permanecem concentrados em estratos socioeconômicos privilegiados.

A dimensão linguística justifica-se pela necessidade de sistemas que reconheçam e validem formas autênticas de expressão juvenil. A imposição de padrões linguísticos rígidos em ferramentas educacionais pode alienar precisamente aqueles que mais necessitam de suporte, perpetuando exclusões baseadas em capital cultural. Ao desenvolver tecnologia que compreende e valoriza variações linguísticas, o trabalho contribui para a inclusão digital e educacional mais efetiva.

Sob perspectiva tecnológica, a pesquisa avança o campo do processamento de linguagem natural em português brasileiro aplicado a domínios específicos. A escassez de recursos computacionais linguísticos para nosso idioma, especialmente considerando variações etárias e regionais, limita o desenvolvimento de aplicações educacionais sofisticadas. A construção de dataset especializado e adaptação de modelos para características juvenis representa contribuição técnica com potencial de impacto em múltiplas aplicações.

A dimensão pedagógica fundamenta-se no reconhecimento de que métodos tradicionais de orientação, desenvolvidos para realidades profissionais estáveis e lineares, mostram-se inadequados para o mundo caracterizado por mudanças aceleradas e emergência constante de novas possibilidades. Jovens contemporâneos navegam em um universo de opções sem precedentes, necessitando de ferramentas que acompanhem essa complexidade e dinamismo.

## 1.3 CONCEITOS FUNDAMENTAIS

Para compreensão adequada deste trabalho, estabelecem-se definições operacionais dos conceitos centrais:

**Linguagem juvenil digital**: conjunto de práticas comunicativas características de adolescentes em ambientes digitais, incluindo abreviações, empréstimos linguísticos, neologismos, uso criativo de pontuação e alternância entre registros formais e informais.

**Mapeamento híbrido de interesses**: abordagem metodológica que integra coleta estruturada de dados através de questionários com análise computacional de expressões livres, permitindo captura multidimensional de preferências e aspirações.

**Gamificação contextualizada**: aplicação de mecânicas de jogos considerando especificidades culturais e desenvolvimentais do público-alvo, evitando transposição mecânica de elementos sem considerar adequação pedagógica.

**Processamento de linguagem natural educacional**: adaptação de técnicas de PLN para contextos educacionais, considerando não apenas precisão técnica, mas também implicações pedagógicas e éticas das classificações automatizadas.

## 1.4 ABORDAGEM METODOLÓGICA

A pesquisa adotou abordagem quali-quantitativa, reconhecendo que a complexidade do fenômeno estudado demanda integração de múltiplas perspectivas metodológicas. O desenvolvimento seguiu processo iterativo organizado em ciclos de design, implementação, teste e refinamento.

A construção do conhecimento iniciou com imersão no contexto através de análise de comunicações juvenis autênticas em ambientes digitais, entrevistas informais com adolescentes e educadores, e observação participante em espaços educacionais. Esta fase etnográfica digital informou decisões subsequentes de design e desenvolvimento.

O desenvolvimento do dataset seguiu metodologia híbrida inovadora que equilibrou a sistematização necessária para treinamento computacional com a preservação da autenticidade linguística. A combinação de templates parametrizáveis, curadoria manual e augmentação via inteligência artificial permitiu escala sem sacrificar qualidade ou representatividade.

A adaptação do modelo de linguagem envolveu experimentação sistemática com diferentes estratégias de fine-tuning, considerando trade-offs entre capacidade de generalização e especialização para domínio específico. A escolha de hiperparâmetros baseou-se em validação cruzada cuidadosa e análise de erros qualitativos.

O design da interface e experiência do usuário seguiu processo centrado no usuário, com prototipagem iterativa e incorporação contínua de *feedback*. Sessões de teste com usuários representativos identificaram pontos de fricção e oportunidades de melhoria.

A validação em contexto escolar real, conduzida em parceria com a Escola Sinodal de São Leopoldo, permitiu a observação do sistema em condições autênticas de uso. A análise privilegiou a compreensão qualitativa dos padrões de expressão e a eficácia do sistema em capturar nuances dos interesses juvenis.

## 1.5 ORGANIZAÇÃO DO DOCUMENTO

Este trabalho estrutura-se em cinco capítulos que apresentam progressivamente o desenvolvimento da pesquisa e seus resultados.

O Capítulo 2 estabelece fundamentação teórica através de revisão bibliográfica abrangente, explorando convergências entre processamento de linguagem natural, teoria educacional, psicologia do desenvolvimento adolescente e design de sistemas interativos. São analisados trabalhos relacionados e identificadas lacunas que justificam a abordagem proposta.

O Capítulo 3 apresenta detalhadamente o estudo de caso, documentando decisões de design, arquitetura técnica, estrutura pedagógica e processo de desenvolvimento. São descritos os componentes do sistema, desde a construção do dataset até a implementação de funcionalidades específicas.

O Capítulo 4 analisa os resultados obtidos com foco na dimensão qualitativa. São explorados padrões linguísticos identificados, casos exemplares de classificação e insights sobre a natureza da expressão de interesses por adolescentes brasileiros. A discussão contextualiza os findings em relação à literatura e implicações práticas.

O Capítulo 5 sintetiza contribuições da pesquisa, reconhece limitações encontradas e propõe direcionamentos para investigações futuras. As referências bibliográficas e materiais complementares completam o documento.

# 2 REFERENCIAL TEÓRICO

## 2.1 INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINA

### 2.1.1 Fundamentos da Inteligência Artificial

A Inteligência Artificial representa um dos campos mais transformadores da ciência da computação moderna, dedicada ao desenvolvimento de sistemas capazes de realizar tarefas que tradicionalmente requerem inteligência humana (RUSSELL; NORVIG, 2020). Desde sua formalização na conferência de Dartmouth em 1956, onde John McCarthy cunhou o termo (MCCARTHY, 1956), a IA evoluiu significativamente, passando de sistemas baseados em regras rígidas para abordagens modernas que aprendem padrões complexos diretamente dos dados.

O surgimento da IA como disciplina formal ocorreu num contexto de otimismo sobre as capacidades computacionais. Os pioneiros do campo acreditavam que em poucas décadas seria possível criar máquinas com inteligência comparável à humana. Embora essa previsão tenha se mostrado excessivamente otimista, o campo produziu avanços fundamentais que hoje permeiam nossa vida cotidiana.

No contexto educacional, a IA oferece possibilidades sem precedentes para personalização do ensino e apoio individualizado aos estudantes. Sistemas inteligentes podem adaptar conteúdos pedagógicos às necessidades específicas de cada aluno, identificar dificuldades de aprendizagem precocemente e fornecer orientação vocacional baseada em interesses e aptidões individuais. Essa capacidade de personalização em escala representa uma revolução potencial na democratização do acesso à educação de qualidade.

O Aprendizado de Máquina (AM), subárea fundamental da IA, permite que sistemas computacionais melhorem seu desempenho através da experiência, sem serem explicitamente programados para cada situação (MITCHELL, 1997). Mitchell (MITCHELL, 1997) define formalmente que “um programa de computador aprende da experiência E com respeito a alguma classe de tarefas T e medida de desempenho P, se seu desempenho em tarefas em T, medido por P, melhora com a experiência E”.

Esta capacidade de aprendizado autônomo é particularmente valiosa em aplicações educacionais, onde a diversidade de contextos, estilos de aprendizagem e backgrounds culturais dos estudantes torna impraticável a criação de regras explícitas para cada situação. Um sistema de orientação vocacional, por exemplo, precisa compreender nuances linguísticas, contextos regionais e expressões juvenis que variam constantemente.

### 2.1.2 Paradigmas de Aprendizado de Máquina

O aprendizado de máquina organiza-se em três paradigmas principais, cada um adequado para diferentes tipos de problemas e disponibilidade de dados. Compreender estes paradigmas é essencial para escolher a abordagem apropriada para cada aplicação.

O aprendizado supervisionado constitui o paradigma mais utilizado em aplicações práticas, incluindo sistemas educacionais. Neste paradigma, o algoritmo aprende a partir de exemplos rotulados, onde cada entrada possui uma saída desejada conhecida. É como um professor que mostra exemplos resolvidos antes de pedir que o aluno resolva problemas similares.

Formalmente, dado um conjunto de treinamento $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, onde $x_i \in X$ representa a entrada e $y_i \in Y$ a saída desejada, o objetivo é aprender uma função $f : X \to Y$ que minimize o erro esperado:

$$\min_{f \in F} \mathbb{E}_{(x,y)\sim P}[L(f(x), y)] \tag{1}$$

onde $L$ é a função de perda que quantifica a diferença entre a predição e o valor real, e $P$ é a distribuição conjunta (desconhecida) dos dados.

Na prática, como não temos acesso à distribuição real $P$, minimizamos o risco empírico sobre os dados de treinamento:

$$\min_{f \in F} \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i) \tag{2}$$

Este princípio fundamenta algoritmos de classificação e regressão amplamente utilizados. Em um sistema de orientação vocacional, por exemplo, o aprendizado supervisionado permite treinar modelos que classificam respostas de estudantes em diferentes áreas de interesse, baseando-se em exemplos previamente categorizados por especialistas.

O aprendizado não supervisionado busca descobrir estruturas e padrões em dados não rotulados. Este paradigma é útil quando não temos exemplos rotulados disponíveis ou quando queremos explorar os dados sem preconceitos prévios. Técnicas comuns incluem *clustering*, onde o algoritmo identifica grupos naturais nos dados; redução de dimensionalidade, que encontra representações mais compactas preservando informações importantes; e modelagem de densidade de probabilidade, que estima a distribuição subjacente dos dados.

No contexto educacional, o aprendizado não supervisionado pode revelar perfis de estudantes não antecipados, identificar padrões de comportamento emergentes ou descobrir relações entre diferentes aspectos do desempenho acadêmico que não seriam evidentes por meio da análise tradicional.

O aprendizado por reforço representa um paradigma onde um agente aprende a tomar decisões através de interação com um ambiente, recebendo recompensas ou punições por suas ações. O objetivo é aprender uma política que maximize a recompensa cumulativa ao longo do tempo. Embora menos comum em aplicações educacionais diretas, princípios de reforço são utilizados em sistemas de gamificação educacional, onde estudantes recebem feedback imediato e recompensas por progresso, criando ciclos de engajamento positivo.

## 2.2 REDES NEURAIS ARTIFICIAIS

### 2.2.1 Fundamentos Matemáticos e Arquitetura

As redes neurais artificiais representam uma das abordagens mais poderosas e versáteis em aprendizado de máquina, inspiradas no funcionamento do cérebro humano. No cérebro biológico, bilhões de neurônios interconectados processam informações através de sinais elétricos e químicos. As redes neurais artificiais capturam a essência deste processamento através de modelos matemáticos simplificados.

O neurônio artificial, unidade básica destas redes, é um modelo matemático que recebe múltiplas entradas, processa-as e produz uma saída. Cada entrada é multiplicada por um peso específico que representa a importância ou força daquela conexão. Esses valores ponderados são somados juntamente com um termo de viés, e o resultado passa por uma função de ativação não-linear.

Matematicamente, o processamento de um neurônio é expresso como:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{3}$$

onde $x_i$ são as entradas do neurônio, $w_i$ são os pesos sinápticos associados a cada entrada, $b$ é o *bias*, $f$ é a função de ativação, e $y$ é a saída do neurônio.

A função de ativação é crucial para o poder expressivo das redes neurais. Sem não-linearidade, mesmo redes com múltiplas camadas seriam equivalentes a um único modelo linear, severamente limitando sua capacidade de aprender padrões complexos. As funções de ativação mais comuns incluem:

A função ReLU tornou-se o padrão em redes profundas modernas:

$$f(x) = \max(0, x) \tag{4}$$

Sua simplicidade computacional e capacidade de mitigar o problema do gradiente desvanecente a tornam ideal para redes profundas. A ReLU permite que gradientes fluam eficientemente durante o treinamento, possibilitando o treinamento de redes com dezenas ou centenas de camadas.

A função sigmoide mapeia valores para o intervalo (0,1):

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

Historicamente importante, a sigmoide é útil quando precisamos interpretar saídas como probabilidades, especialmente em camadas de saída para classificação binária.

A tangente hiperbólica oferece saída no intervalo (-1,1):

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{6}$$

A simetria em torno de zero pode facilitar o treinamento em algumas arquiteturas, especialmente em redes recorrentes.

Para problemas de classificação multiclasse, a função softmax é essencial:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{7}$$

A softmax converte um vetor de valores reais em uma distribuição de probabilidade válida, onde todas as saídas são positivas e somam 1.

Redes neurais profundas organizam neurônios em camadas hierárquicas. A arquitetura típica inclui uma camada de entrada que recebe os dados brutos, múltiplas camadas ocultas que aprendem representações progressivamente mais abstratas, e uma camada de saída que produz as predições finais. Cada camada transforma a representação dos dados, extraindo características cada vez mais complexas e relevantes para a tarefa.

A propagação direta calcula a saída da rede camada por camada:

$$h^{(0)} = x \tag{8}$$

$$h^{(l)} = f^{(l)}(W^{(l)}h^{(l-1)} + b^{(l)}) \quad \text{para } l = 1, ..., L \tag{9}$$

$$\hat{y} = h^{(L)} \tag{10}$$

onde $h^{(l)}$ representa a ativação da camada $l$, $W^{(l)}$ é a matriz de pesos da camada $l$, $b^{(l)}$ é o vetor de viés da camada $l$, e $L$ é o número total de camadas.

### 2.2.2 Treinamento via Retropropagação

O treinamento de redes neurais é um processo iterativo de otimização que ajusta os pesos para minimizar o erro entre as predições da rede e os valores desejados. O algoritmo de retropropagação, combinado com gradiente descendente, tornou viável o treinamento de redes profundas.

O processo de treinamento segue quatro etapas principais. Primeiro, os dados passam pela rede através de propagação direta, gerando predições. Segundo, o erro é calculado comparando predições com valores reais usando uma função de perda apropriada. Terceiro, os gradientes do erro com respeito a cada peso são calculados, propagando o erro de volta pela rede. Finalmente, os pesos são atualizados na direção oposta ao gradiente para reduzir o erro.

A regra de atualização dos pesos segue:

$$W_{ij}^{(l)} \leftarrow W_{ij}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}} \tag{11}$$

onde $\eta$ é a taxa de aprendizado, um hiperparâmetro crucial que controla o tamanho dos passos na otimização. Uma taxa muito alta pode fazer o treinamento divergir, enquanto uma taxa muito baixa torna o treinamento excessivamente lento.

Para problemas de classificação multiclasse, como categorizar interesses vocacionais, a função de perda mais apropriada é a entropia cruzada categórica:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \tag{12}$$

onde $N$ é o número de amostras no *batch*, $C$ é o número de classes, $y_{ij}$ é o rótulo verdadeiro, e $\hat{y}_{ij}$ é a probabilidade predita para a classe $j$ da amostra $i$.

O treinamento moderno de redes neurais incorpora várias técnicas para melhorar a convergência e a generalização. O *momentum* acelera a convergência adicionando uma fração do *update* anterior ao *update* atual. Otimizadores adaptativos como o Adam ajustam a taxa de aprendizado individualmente para cada parâmetro baseado em estatísticas dos gradientes. Regularização através de *dropout* desativa aleatoriamente neurônios durante o treinamento, prevenindo coadaptação excessiva. Normalização em lote estabiliza o treinamento normalizando ativações entre camadas.

## 2.3 MÉTRICAS DE AVALIAÇÃO PARA CLASSIFICAÇÃO

### 2.3.1 Matriz de Confusão e Métricas Derivadas

A avaliação adequada de sistemas de classificação é fundamental para garantir sua eficácia em aplicações reais. Em sistemas educacionais, onde decisões podem impactar significativamente trajetórias de vida, a escolha e interpretação corretas de métricas tornam-se ainda mais críticas.

A matriz de confusão serve como base para todas as métricas de classificação, organizando as predições do modelo em uma tabela que compara valores preditos com valores reais. Para classificação binária, a matriz contém quatro componentes fundamentais: Verdadeiros Positivos (VP) representam casos positivos corretamente identificados pelo modelo, Verdadeiros Negativos (VN) são casos negativos corretamente classificados, Falsos Positivos (FP) ocorrem quando o modelo incorretamente classifica um caso negativo como positivo, e Falsos Negativos (FN) acontecem quando casos positivos são incorretamente classificados como negativos.

A partir destes componentes básicos, derivamos métricas que capturam diferentes aspectos do desempenho do classificador:

A acurácia mede a proporção total de predições corretas:

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{13}$$

Embora intuitiva e amplamente utilizada, a acurácia pode ser enganosa em conjuntos de dados desbalanceados. Por exemplo, em um sistema de detecção de alunos em risco de evasão onde apenas 5% dos estudantes realmente evadem, um classificador que sempre prediz “não vai evadir” terá 95% de acurácia sem fornecer valor real.

A precisão quantifica a confiabilidade das predições positivas:

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{14}$$

Alta precisão significa que, quando o modelo faz uma predição positiva, ela provavelmente está correta. Em um sistema de recomendação vocacional, alta precisão garante que as carreiras sugeridas sejam realmente adequadas ao perfil do estudante, evitando direcionamentos inadequados.

A revocação (também chamada sensibilidade ou taxa de verdadeiros positivos) mede a capacidade de identificar todos os casos positivos:

$$\text{Revocação} = \frac{VP}{VP + FN} \tag{15}$$

Alta revocação é crucial quando o custo de perder casos positivos é alto. No contexto educacional, queremos identificar todos os estudantes que poderiam se beneficiar de determinado suporte ou oportunidade.

O F1-Score fornece um balanço entre precisão e revocação através da média harmônica:

$$\text{F1} = 2 \cdot \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{16}$$

A média harmônica penaliza valores extremos mais que a média aritmética, garantindo que o F1-Score seja alto apenas quando tanto precisão quanto revocação são razoáveis. Esta propriedade torna o F1-Score particularmente útil quando precisamos equilibrar falsos positivos e falsos negativos.

Para problemas de classificação multiclasse, onde temos mais de duas categorias possíveis, as métricas precisam ser adaptadas. Existem duas estratégias principais:

A macro-média calcula a métrica independentemente para cada classe e tira a média simples:

$$\text{Macro-F1} = \frac{1}{C} \sum_{i=1}^{C} \text{F1}_i \tag{17}$$

Esta abordagem trata todas as classes igualmente, independente de sua frequência no dataset. É apropriada quando todas as classes são igualmente importantes, mesmo que algumas sejam raras.

A micro-média agrega as contribuições de todas as classes antes de calcular a métrica:

$$\text{Micro-Precisão} = \frac{\sum_{i=1}^{C} VP_i}{\sum_{i=1}^{C} (VP_i + FP_i)} \tag{18}$$

A micro-média dá peso implícito maior a classes mais frequentes, sendo apropriada quando queremos otimizar o desempenho global considerando a distribuição natural dos dados.

Em aplicações educacionais, a escolha entre estas estratégias tem implicações importantes. Por exemplo, em um sistema de orientação vocacional com categorias desbalanceadas (muitos estudantes interessados em algumas áreas populares e poucos em áreas

especializadas), a macro-média garante que o sistema funcione bem para todos os perfis de estudantes, não apenas os mais comuns.

## 2.4 PROCESSAMENTO DE LINGUAGEM NATURAL

### 2.4.1 Evolução Histórica e Paradigmas

O Processamento de Linguagem Natural (PLN) representa uma das áreas mais desafiadoras e fascinantes da inteligência artificial, dedicada a permitir que computadores compreendam, interpretem e gerem linguagem humana de forma útil. A linguagem humana, com sua riqueza, ambiguidade e constante evolução, apresenta desafios únicos que têm motivado décadas de pesquisa e inovação.

A história do PLN pode ser dividida em três grandes eras, cada uma caracterizada por abordagens e filosofias distintas. A era simbólica, dominante desde os anos 1950 até os 1990, baseava-se na codificação explícita de regras linguísticas. Linguistas computacionais criavam gramáticas formais, dicionários detalhados e sistemas de regras para análise morfológica, sintática e semântica. Estes sistemas, embora teoricamente elegantes e interpretáveis, mostraram-se frágeis na prática. A linguagem natural é repleta de exceções, ambiguidades e variações contextuais que tornam impraticável enumerar todas as regras necessárias.

A era estatística, iniciada nos anos 1990, representou uma mudança de paradigma fundamental. Ao invés de tentar codificar conhecimento linguístico explicitamente, os pesquisadores passaram a extrair padrões estatísticos de grandes corpora de texto. Modelos de linguagem n-gram capturavam probabilidades de sequências de palavras. Modelos Ocultos de Markov revolucionaram tarefas como etiquetagem morfossintática e reconhecimento de fala. Campos Aleatórios Condicionais permitiram modelagem mais flexível de sequências. Métodos de tradução automática estatística transformaram a área ao aprender alinhamentos entre línguas diretamente dos dados.

A era neural, iniciada com o ressurgimento das redes neurais profundas por volta de 2010, trouxe capacidades sem precedentes ao PLN. Redes neurais aprendem representações ricas e hierárquicas diretamente dos dados, capturando nuances semânticas e sintáticas que escapavam a abordagens anteriores. Mais importante, estas representações são aprendidas de forma automática, sem necessidade de engenharia manual de características.

### 2.4.2 Representação Vetorial de Palavras

Um desafio fundamental em PLN é como representar palavras de forma que computadores possam processá-las matematicamente. A solução inicial, ainda usada em algumas aplicações, é o *one-hot encoding*, onde cada palavra é representada por um vetor esparso com dimensão igual ao tamanho do vocabulário. Nesta representação, apenas uma posição tem valor 1 (correspondente à palavra) e todas as outras são 0. Embora simples de imple-

mentar, esta abordagem tem limitações severas: os vetores são extremamente esparsos e de alta dimensão, não há noção de similaridade entre palavras (a distância entre 'gato' e 'felino' é a mesma que entre 'gato' e 'democracia'), e o tamanho da representação cresce linearmente com o vocabulário.

A revolução nas representações de palavras veio com o desenvolvimento de *word embeddings* - representações densas e contínuas que capturam significado semântico. A ideia fundamental é que palavras que aparecem em contextos semelhantes tendem a ter significados relacionados. Esta hipótese distribucional, formalizada por linguistas como Firth (FIRTH, 1957) ("You shall know a word by the company it keeps"), fornece a base teórica para aprender representações significativas.

Word2Vec (MIKOLOV *et al.*, 2013) popularizou *embeddings* neurais através de duas arquiteturas elegantes e eficientes:

O modelo Skip-gram aprende representações prevendo palavras de contexto dada uma palavra central. O objetivo é maximizar:

$$\max_{\theta} \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t; \theta) \tag{19}$$

onde $T$ é o tamanho do corpus, $c$ é o tamanho da janela de contexto, e $\theta$ são os parâmetros do modelo.

O modelo CBOW inverte a tarefa, prevendo a palavra central dado o contexto. Embora conceitualmente mais simples, CBOW tende a ser mais rápido de treinar e a funcionar melhor para palavras frequentes.

Uma das descobertas mais fascinantes sobre *word embeddings* é que eles capturam relações semânticas através de operações vetoriais simples. O exemplo clássico:

$$\vec{v}_{\text{rei}} - \vec{v}_{\text{homem}} + \vec{v}_{\text{mulher}} \approx \vec{v}_{\text{rainha}} \tag{20}$$

demonstra que as representações codificam não apenas similaridade, mas também relações analógicas complexas. Isso sugere que os *embeddings* capturam a estrutura semântica profunda da linguagem.

GloVe (PENNINGTON; SOCHER; MANNING, 2014) unificou abordagens baseadas em contagem com métodos preditivos, otimizando diretamente para que produtos escalares entre word vectors correspondam a log-probabilidades de coocorrência. FastText estendeu Word2Vec para trabalhar com subpalavras, permitindo representações para palavras fora do vocabulário e capturando informações morfológicas.

A importância de representações apropriadas para o português brasileiro não pode ser subestimada. A linguagem juvenil brasileira é rica em neologismos ("shippar", "stalkar"), variações ortográficas ("vc", "tb", "pq"), empréstimos adaptados do inglês e expressões regionais. Modelos treinados em outros idiomas ou mesmo em português europeu falham em capturar estas nuances essenciais para compreender a comunicação autêntica dos jovens.

## 2.5 ARQUITETURA TRANSFORMER E MODELOS DE LINGUAGEM PRÉ-TREINADOS

### 2.5.1 A Revolução do Transformer

A arquitetura Transformer, introduzida no artigo seminal "Attention Is All You Need" (VASWANI *et al.*, 2017), representa um dos avanços mais significativos na história do processamento de linguagem natural. Antes dos Transformers, o processamento de sequências era dominado por Redes Neurais Recorrentes e suas variantes sofisticadas como LSTM e GRU.

Estas arquiteturas recorrentes, embora poderosas, enfrentavam limitações fundamentais. O processamento sequencial inerente às RNNs impedia a paralelização eficiente durante o treinamento, tornando-o proibitivamente lento para sequências longas. O problema dos gradientes desvanecentes dificultava o aprendizado de dependências de longo alcance, limitação bem documentada na literatura de redes neurais recorrentes. Além disso, toda a informação contextual precisava ser comprimida em vetores de estado oculto de tamanho fixo, criando um gargalo de informação.

O Transformer resolve estas limitações através do mecanismo de *self-attention*, que permite que cada posição na sequência atenda diretamente a todas as outras posições. Esta abordagem oferece várias vantagens: processamento paralelo de toda a sequência, captura direta de dependências de longo alcance sem degradação e flexibilidade para modelar diferentes tipos de relações simultaneamente.

O mecanismo de atenção baseia-se em três componentes conceituais: Query (Q) representa "o que estamos procurando", Key (K) representa "o que está disponível para comparação", e Value (V) representa "o conteúdo real a ser agregado". Para cada posição na sequência, calculamos sua relevância para todas as outras posições e usamos esses pesos para agregar informação.

Matematicamente, a atenção é calculada como:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{21}$$

O produto $QK^T$ calcula similaridades entre todas as queries e keys. A divisão por $\sqrt{d_k}$ escala estes valores para evitar que a softmax sature quando a dimensionalidade é alta. A softmax converte similaridades em pesos normalizados que somam 1. Finalmente, estes pesos são usados para calcular uma combinação linear dos values.

Para capturar diferentes tipos de relações simultaneamente, o Transformer usa atenção multi-cabeça:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{22}$$

onde cada cabeça é calculada independentemente:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{23}$$

Cada cabeça pode especializar-se em diferentes aspectos da linguagem. Estudos mostram que algumas cabeças focam em relações sintáticas locais (como concordância sujeito-verbo), outras capturam dependências semânticas de longo alcance, e outras ainda parecem especializar-se em aspectos posicionais ou estruturais do texto.

A arquitetura completa do Transformer inclui vários componentes além da atenção. Embeddings de entrada convertem tokens para vetores densos. Codificação posicional adiciona informação sobre a posição de cada token, crucial pois a atenção em si não tem noção de ordem. As codificações senoidais permitem que o modelo generalize para sequências mais longas que as vistas durante o treinamento:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \tag{24}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \tag{25}$$

Redes feed-forward são aplicadas identicamente a cada posição, permitindo transformações não-lineares complexas. Conexões residuais ao redor de cada subcamada facilitam o fluxo de gradientes em redes profundas. Layer normalization estabiliza o treinamento normalizando ativações.

### 2.5.2 BERT e a Era dos Modelos Pré-treinados

BERT (Bidirectional Encoder Representations from Transformers) (DEVLIN *et al.*, 2019) catalisou uma revolução em PLN ao demonstrar o poder do pré-treinamento bidirecional em escala. Antes do BERT, modelos de linguagem eram predominantemente unidirecionais, processando texto da esquerda para a direita (ou vice-versa). Esta limitação impedia que modelos utilizassem contexto futuro ao processar cada token, reduzindo significativamente sua capacidade de compreensão.

A inovação central do BERT foi desenvolver uma tarefa de pré-treinamento que permite aprendizado bidirecional profundo. A Modelagem de Linguagem Mascarada funciona mascarando aleatoriamente 15% dos tokens de entrada e treinando o modelo para predizê-los baseado no contexto bidirecional completo.

O processo de mascaramento é cuidadosamente projetado para evitar desalinhamento entre pré-treinamento e fine-tuning. Dos tokens selecionados para mascaramento, 80% são substituídos pelo token especial [MASK], 10% são substituídos por um token aleatório do vocabulário, e 10% permanecem inalterados. Esta estratégia força o modelo a sempre tentar prever o token correto, pois não sabe quais tokens foram alterados.

A função objetivo do MLM é:

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in \text{masked}} \log P(x_i | x_{\setminus i}) \tag{26}$$

onde $x_{\setminus i}$ representa todos os tokens exceto o i-ésimo.

Além do MLM, BERT usa Predição de Próxima Sentença (NSP) para aprender relações entre sentenças. Durante o pré-treinamento, o modelo recebe pares de sentenças e deve prever se a segunda sentença segue a primeira no texto original. Esta tarefa ajuda em aplicações que requerem compreensão de relações entre múltiplas sentenças.

BERT é disponibilizado em duas configurações principais. BERT Base contém 12 camadas transformer, 768 dimensões ocultas, 12 cabeças de atenção, totalizando 110 milhões de parâmetros. BERT Large escala para 24 camadas, 1024 dimensões ocultas, 16 cabeças de atenção, resultando em 340 milhões de parâmetros. O vocabulário usa WordPiece tokenization com 30.000 tokens, permitindo representação eficiente de palavras raras através de subpalavras.

O paradigma de *fine-tuning* do BERT é elegantemente simples. Para adaptar o modelo pré-treinado a uma tarefa específica, adiciona-se apenas uma camada específica para a tarefa sobre as representações do BERT. Para classificação, usa-se a representação do token especial [CLS]:

$$P(y|x) = \text{softmax}(Wh_{[CLS]} + b) \tag{27}$$

onde $h_{[CLS]}$ é a representação final do token [CLS] que agrega informação sobre toda a sequência.

O impacto do BERT foi extraordinário. Em poucos meses após seu lançamento, estabeleceu novos recordes em praticamente todas as tarefas de compreensão de linguagem natural. Mais importante, democratizou o acesso a PLN de alta qualidade - pesquisadores e desenvolvedores podiam alcançar resultados estado-da-arte fazendo ajuste fino do modelo pré-treinado com recursos computacionais modestos.

BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020) representa a adaptação do BERT especificamente para o português brasileiro. Esta especialização é crucial pois modelos multilíngues como mBERT, embora funcionais, sacrificam qualidade em cada língua individual ao tentar cobrir mais de 100 idiomas simultaneamente.

O corpus de treinamento do BERTimbau foi cuidadosamente curado para capturar a riqueza do português brasileiro contemporâneo. Foram utilizados 2,68 bilhões de palavras combinando o brWaC (Brazilian Web as Corpus), que captura linguagem informal e contemporânea da web brasileira, com a Wikipedia em português, que fornece texto mais formal e enciclopédico. Esta combinação garante que o modelo compreenda tanto a norma culta quanto variações coloquiais, gírias, regionalismos e a linguagem digital característica da comunicação juvenil.

Experimentos extensivos demonstraram a superioridade do BERTimbau sobre alternativas multilíngues em tarefas específicas para português. Em tarefas de similaridade semântica, reconhecimento de entidades nomeadas e análise de sentimento, os ganhos foram consistentes e significativos. Particularmente importante para aplicações educacionais, o BERTimbau mostrou compreensão superior de linguagem informal, neologismos e

expressões idiomáticas brasileiras.

## 2.6 REACT E DESENVOLVIMENTO FRONTEND MODERNO

### 2.6.1 Paradigma Declarativo e Componentização

O React representa uma mudança fundamental na forma como interfaces de usuário são construídas para aplicações web. Desenvolvido pelo Facebook e lançado ao público em 2013, esta biblioteca JavaScript introduziu um paradigma declarativo que simplifica significativamente o desenvolvimento de interfaces complexas e interativas (FROST, 2013).

Tradicionalmente, o desenvolvimento web baseava-se em manipulação imperativa do DOM (Document Object Model), exigindo que desenvolvedores especificassem exatamente como cada elemento deveria ser modificado em resposta a mudanças de estado. Esta abordagem mostrava-se propensa a erros e difícil de manter em aplicações de grande escala. O React resolve estas limitações permitindo que desenvolvedores descrevam como a interface deve aparecer em qualquer estado dado, delegando à biblioteca a responsabilidade de realizar as atualizações necessárias de forma eficiente.

O conceito central do React é o componente - uma unidade independente e reutilizável que encapsula estrutura, apresentação e comportamento. Componentes podem ser compostos hierarquicamente para criar interfaces arbitrariamente complexas, similar a como funções são compostas em programação. Esta abordagem promove modularidade, reusabilidade e manutenibilidade do código.

Os componentes React recebem dados através de propriedades (props) - valores imutáveis passados de componente pai para filho. As props permitem customização e configuração flexível dos componentes. Adicionalmente, componentes mantêm estado (state) interno que representa dados mutáveis que mudam ao longo do tempo. Quando o estado muda, React automaticamente re-renderiza o componente e seus descendentes afetados, mantendo a interface sincronizada com os dados subjacentes.

A inovação técnica que torna o paradigma declarativo do React eficiente é o Virtual DOM - uma representação JavaScript leve da estrutura de interface. Quando o estado muda, React cria uma nova árvore Virtual DOM representando o estado desejado. Um algoritmo sofisticado de diferenciação compara a nova árvore com a anterior, identificando o conjunto mínimo de mudanças necessárias. Apenas essas mudanças são aplicadas ao DOM real, otimizando significativamente a performance.

React Hooks, introduzidos na versão 16.8, representam a evolução mais significativa da biblioteca desde seu lançamento. Hooks permitem que componentes funcionais utilizem recursos antes exclusivos de componentes de classe, incluindo gerenciamento de estado local através do *useState*, sincronização com sistemas externos via *useEffect*, e otimizações de performance com *useMemo* e *useCallback*.

Para aplicações educacionais, o paradigma componentizado do React oferece van-

tagens significativas. Componentes de interface podem ser desenvolvidos e testados isoladamente, facilitando a criação de bibliotecas de componentes educacionais reutilizáveis. A reatividade automática garante que mudanças em dados - como respostas de estudantes ou atualizações de progresso - sejam refletidas instantaneamente na interface, criando uma experiência fluida e responsiva.

## 2.7 DESENVOLVIMENTO DE APIS E ARQUITETURAS MODERNAS

### 2.7.1 APIs RESTful e FastAPI

O desenvolvimento de aplicações modernas fundamenta-se em APIs (Application Programming Interfaces) bem projetadas que permitem comunicação eficiente entre diferentes componentes de software. REST, proposto por Roy Fielding, estabeleceu-se como paradigma arquitetural dominante para APIs web devido à sua simplicidade, escalabilidade e alinhamento natural com princípios da web.

Os princípios fundamentais do REST definem uma arquitetura que promove sistemas fracamente acoplados. A interface uniforme estabelece que recursos são identificados por URIs, manipulados através de representações (tipicamente JSON), e acessados usando métodos HTTP padronizados: GET para recuperação, POST para criação, PUT para atualização e DELETE para remoção. O princípio stateless determina que cada requisição deve conter toda a informação necessária, simplificando a escalabilidade horizontal.

FastAPI emergiu como o framework Python preferido para construção de APIs RESTful de alta performance. Criado por Sebastián Ramírez em 2018 (RAMÍREZ, 2018), o framework combina recursos modernos do Python com *design* elegância. Baseado em Starlette para funcionalidade ASGI e Pydantic para validação de dados, FastAPI oferece performance comparável a frameworks em linguagens compiladas.

A validação automática através de type hints representa uma das inovações mais impactantes do FastAPI. Ao declarar tipos de parâmetros, o framework automaticamente valida entradas, converte tipos quando possível, gera mensagens de erro detalhadas e produz documentação OpenAPI 3.0 completa. Esta documentação é servida interativamente através de Swagger UI e ReDoc, transformando a experiência de desenvolvimento e consumo de APIs.

A programação assíncrona nativa permite que FastAPI processe múltiplas requisições concorrentemente sem bloqueio. Quando operações aguardam I/O - consultas a banco de dados, chamadas a APIs externas, leitura de arquivos - a thread é liberada para processar outras requisições, maximizando a utilização de recursos. Benchmarks mostram FastAPI processando mais de 30.000 requisições por segundo em *endpoints* simples (FASTAPI, 2024), performance crucial para aplicações educacionais que experimentam picos de uso.

O sistema de injeção de dependências promove código limpo e testável. Dependên-

cias são funções Python que o framework resolve automaticamente, permitindo gerenciamento elegante de conexões de banco de dados, autenticação, validações customizadas e recursos compartilhados. Para sistemas educacionais, isso facilita a implementação de políticas de acesso complexas e integração com múltiplos serviços.

### 2.7.2 Arquitetura de Microsserviços

A arquitetura de microsserviços representa uma evolução fundamental na construção de sistemas complexos. Em contraste com monólitos tradicionais, microsserviços decompõem aplicações em serviços pequenos, independentes e especializados que comunicam através de APIs bem definidas.

Esta abordagem oferece vantagens significativas para sistemas educacionais. A escalabilidade granular permite responder eficientemente a padrões de uso variáveis - serviços de matrícula podem escalar durante períodos de inscrição, enquanto outros mantêm capacidade normal. A resiliência através de isolamento garante que falhas em um serviço não comprometam o sistema inteiro. O desenvolvimento paralelo acelera a entrega de valor, permitindo que equipes especializadas trabalhem independentemente.

Os padrões de comunicação em microsserviços requerem *design* cuidadoso. Comunicação síncrona via REST é apropriada para operações que demandam resposta imediata. Comunicação assíncrona através de filas de mensagens é ideal para processamento em *background*.

API Gateway atua como ponto único de entrada, simplificando o consumo dos microsserviços. O gateway roteia requisições, agrega respostas quando necessário, gerencia autenticação centralizadamente, implementa rate limiting e fornece cache. Para aplicações educacionais acessadas por milhares de usuários simultaneamente, essas funcionalidades são essenciais.

Os desafios da arquitetura não devem ser subestimados. A complexidade operacional aumenta significativamente - gerenciar dezenas de serviços requer ferramentas sofisticadas e processos maduros. Kubernetes emergiu como padrão de fato para orquestração de *containers*, fornecendo abstrações para *deployment*, *scaling* e *service discovery*. A consistência de dados sem transações distribuídas tradicionais requer mudança de paradigma, abraçando consistência eventual onde apropriado.

### 2.7.3 Programação Orientada a Eventos

A programação orientada a eventos representa um paradigma poderoso para sistemas distribuídos modernos. Ao invés de chamadas síncronas diretas entre serviços, componentes publicam eventos que outros podem consumir assincronamente. Esta abordagem promove acoplamento ainda mais fraco e permite arquiteturas altamente escaláveis e resilientes.

Em sistemas educacionais, eventos naturalmente modelam muitas interações. Quando um estudante completa uma avaliação, um evento "AvaliacaoCompletada" pode disparar múltiplas ações: atualização de progresso, cálculo de estatísticas, notificação a professores, verificação de conquistas. Cada serviço interessado processa o evento independentemente, sem conhecimento dos outros consumidores.

Event sourcing leva o paradigma além, armazenando todas as mudanças de estado como sequência imutável de eventos. Ao invés de armazenar apenas o estado atual de um estudante, o sistema mantém o histórico completo: matrícula, participação em atividades, submissões, avaliações. Este histórico permite não apenas reconstruir qualquer estado passado, mas também análises temporais sofisticadas sobre a evolução do aprendizado.

CQRS (Command Query Responsibility Segregation) complementa event sourcing separando modelos de escrita e leitura. Comandos modificam estado gerando eventos, enquanto queries consultam projeções otimizadas para leitura. Esta separação permite otimizar independentemente para diferentes padrões de acesso - escritas consistentes e normalizadas versus leituras desnormalizadas e ultra-rápidas.

Serviços de Mensageria como Apache Kafka ou RabbitMQ fornecem infraestrutura robusta para arquiteturas orientadas a eventos. Garantias de entrega, ordenação de mensagens, particionamento para escalabilidade são funcionalidades essenciais.

## 2.8 COMPUTAÇÃO EM NUVEM E ARMAZENAMENTO DE DADOS

### 2.8.1 Google Cloud Platform e Bancos NoSQL

A computação em nuvem revolucionou o desenvolvimento, implantação e operação de aplicações modernas. Este modelo permite acesso sob demanda a recursos computacionais através da internet, eliminando a necessidade de investimento de capital em infraestrutura própria e permitindo escalabilidade elástica conforme a demanda (MELL; GRANCE, 2011).

O Google Cloud Platform destaca-se no cenário de computação em nuvem por sua infraestrutura global de alta performance e serviços especializados para cargas de trabalho de inteligência artificial. A rede privada global do Google oferece latência consistentemente baixa e alta disponibilidade através de cabos submarinos proprietários e pontos de presença distribuídos globalmente (GOOGLE CLOUD, 2024).

Para aplicações de aprendizado de máquina, o GCP oferece recursos computacionais especializados. O Compute Engine fornece acesso a GPUs de última geração, incluindo NVIDIA T4, V100 e A100, ideais para treinamento de modelos neurais profundos. As Tensor Processing Units (TPUs), chips customizados desenvolvidos pelo Google especificamente para cargas de trabalho de ML, oferecem performance superior para determinados tipos de modelos. O Cloud Run permite implantação serverless de *containers*, escalando automaticamente de zero a milhares de instâncias conforme a demanda - característica

crucial para APIs de inferência que experimentam tráfego variável.

O Vertex AI representa a evolução da plataforma de aprendizado de máquina do Google, unificando todo o ciclo de vida de desenvolvimento de modelos. O AutoML democratiza a criação de modelos customizados, permitindo que usuários sem expertise profunda treinem modelos de alta qualidade através de técnicas automatizadas de busca de arquitetura neural e aprendizado por transferência.

### 2.8.2 Cloud Firestore: Evolução dos Bancos NoSQL

Os bancos de dados NoSQL emergiram para endereçar limitações dos sistemas relacionais tradicionais na era do big data e aplicações web de escala global. Enquanto bancos relacionais destacam-se em consistência e consultas complexas, enfrentam desafios significativos com escalabilidade horizontal e flexibilidade de esquema.

O Cloud Firestore, banco de dados NoSQL do Google, representa uma evolução significativa ao combinar benefícios de bancos de documentos com garantias tradicionalmente associadas apenas a sistemas relacionais. O Firestore organiza dados em uma hierarquia flexível de coleções e documentos, onde documentos são estruturas similares a JSON que suportam tipos primitivos, complexos e especiais como timestamps com precisão de microsegundos e coordenadas geográficas.

A estrutura hierárquica permite modelagem natural de relações um-para-muitos através de subcoleções. Por exemplo, em um sistema educacional, uma coleção de escolas pode conter documentos de escola, cada um com subcoleção de estudantes, que por sua vez contém subcoleção de respostas. Esta organização mantém dados relacionados próximos, otimizando padrões de acesso comuns em aplicações educacionais.

O Firestore oferece garantias excepcionalmente fortes para um banco NoSQL. Todas as leituras são fortemente consistentes, sempre retornando o valor mais recente confirmado. Transações ACID suportam operações atômicas em até 500 documentos, com isolamento snapshot garantindo visão consistente mesmo com escritas concorrentes. Esta combinação única de flexibilidade NoSQL com garantias ACID é particularmente valiosa para aplicações educacionais que requerem tanto agilidade quanto confiabilidade.

A sincronização em tempo real transforma como aplicações interagem com dados. Clientes podem estabelecer listeners em documentos ou consultas, recebendo atualizações instantâneas quando dados mudam no backend. Para aplicações educacionais, isso permite experiências como professores acompanhando respostas de estudantes em tempo real ou múltiplos orientadores colaborando na análise de casos.

O suporte offline nativo é crucial para contextos educacionais com conectividade intermitente. O Firestore mantém cache local persistente de todos os dados acessados, servindo leituras do cache quando offline e enfileirando escritas para sincronização posterior. Conflitos são resolvidos de maneira determinística usando timestamps híbridos que combinam tempo físico com contadores lógicos.

A performance de consultas no Firestore é previsível e escalável. Todas as consultas são servidas por índices, garantindo complexidade O(log n) independente do tamanho da coleção. A segurança é implementada através de regras declarativas que controlam acesso em nível de documento, permitindo políticas sofisticadas sem código de backend adicional.

## 2.9 CONTEXTO EDUCACIONAL BRASILEIRO

### 2.9.1 Sistema Nacional de Educação: Estrutura e Desafios

O sistema educacional brasileiro figura entre os maiores e mais complexos do mundo, atendendo a população diversa em território continental marcado por profundas desigualdades socioeconômicas. Segundo dados do Censo Escolar 2023 divulgados pelo INEP (INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA, 2024), o Brasil atende 47,3 milhões de estudantes distribuídos em 178.947 estabelecimentos de ensino, com 81,7% das matrículas concentradas na rede pública.

A estrutura do sistema educacional, definida pela Lei de Diretrizes e Bases da Educação Nacional (BRASIL, 1996), organiza-se em etapas progressivas. A Educação Infantil atende crianças de 0 a 5 anos, subdividindo-se em creche e pré-escola. O Ensino Fundamental, com duração de nove anos, divide-se em anos iniciais (1º ao 5º) focados em alfabetização e letramento, e anos finais (6º ao 9º) com estrutura disciplinar mais complexa. O Ensino Médio, reformado pela Lei nº 13.415/2017 (BRASIL, 2017), introduziu flexibilização curricular através de itinerários formativos.

A governança federativa distribui responsabilidades entre União, estados e municípios, criando desafios significativos de coordenação. O financiamento baseia-se em vinculação constitucional de receitas, com o FUNDEB redistribuindo recursos para reduzir desigualdades. Entretanto, disparidades permanecem dramáticas - o valor por aluno/ano varia mais de 300% entre regiões, refletindo-se em diferenças de infraestrutura e oportunidades educacionais.

### 2.9.2 Desafios da Orientação Vocacional no Brasil

Os desafios da orientação vocacional no contexto brasileiro são particularmente agudos. Dados indicam que a maioria das escolas públicas carece de profissionais especializados em orientação vocacional, limitando drasticamente o acesso à orientação profissional qualificada (CONSELHO NACIONAL DE PESQUISA E DESENVOLVIMENTO, 2023). Estudos sugerem que estudantes do ensino médio têm conhecimento limitado sobre a diversidade de carreiras disponíveis, geralmente as mais tradicionais, permanecendo alheios a carreiras emergentes em tecnologia, economia criativa e sustentabilidade.

As pressões socioeconômicas intensificam estes desafios. Dados da PNAD Contínua (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2023) revelam que 29,2% dos jovens entre 18 e 24 anos nem estudam nem trabalham. Muitos abandonam

projetos educacionais por necessidade de contribuir com a renda familiar, perpetuando ciclos de pobreza e limitação de oportunidades.

As transformações aceleradas do mercado de trabalho tornam a orientação ainda mais complexa. O World Economic Forum estima que muitas das crianças entrando hoje na escola trabalharão em profissões ainda inexistentes (WORLD ECONOMIC FORUM, 2020). As desigualdades regionais amplificam todos estes desafios - o IDEB varia dramaticamente entre municípios do Sul/Sudeste e Norte/Nordeste, refletindo disparidades profundas em todos os aspectos educacionais.

Neste contexto, a tecnologia emerge como ferramenta potencialmente democratizadora. Sistemas baseados em inteligência artificial podem atender milhões de estudantes simultaneamente, oferecendo orientação personalizada com custos marginais próximos a zero após o desenvolvimento. A disponibilidade contínua permite acesso no momento mais conveniente, respeitando rotinas diversas de trabalho e estudo.

Entretanto, o potencial transformador deve ser temperado com a consciência das limitações. O acesso digital permanece desigual segundo dados da pesquisa TIC Domicílios (COMITÊ GESTOR DA INTERNET NO BRASIL, 2023). Mesmo com acesso, a qualidade frequentemente é insuficiente para aplicações sofisticadas, e muitos estudantes dependem exclusivamente de smartphones com recursos limitados.

## 2.10 CONSIDERAÇÕES ÉTICAS EM IA EDUCACIONAL

### 2.10.1 Privacidade e Proteção de Dados

A aplicação de inteligência artificial em contextos educacionais suscita questões éticas fundamentais sobre privacidade, equidade e o propósito da educação. A natureza sensível do contexto - envolvendo menores em fase crucial de desenvolvimento - demanda uma ferramenta ética particularmente robusta (FLORIDI *et al.*, 2018).

Dados educacionais revelam informações profundamente pessoais: processos de pensamento, dificuldades de aprendizagem, interesses, contexto socioeconômico e potenciais indicadores de saúde mental. O caráter longitudinal destes dados, acompanhando indivíduos por toda a trajetória acadêmica, amplifica riscos e responsabilidades.

A Lei Geral de Proteção de Dados (BRASIL, 2018) estabelece salvaguardas específicas para dados de crianças e adolescentes. O tratamento deve realizar-se em seu melhor interesse, com consentimento específico dos responsáveis. Princípios de privacidade por *design* tornam-se imperativos: minimização de dados, proporcionalidade, limitação de finalidade e retenção limitada devem guiar todo o desenvolvimento.

A segurança requer medidas excepcionais além de proteções técnicas. Criptografia forte, controles de acesso granulares e logs de auditoria completos formam apenas a base. A maior vulnerabilidade frequentemente é humana, requerendo cultura organizacional de segurança e treinamento contínuo.

# 3 ESTUDO DE CASO

## 3.1 CLASSIFICAÇÃO E ABORDAGEM DA PESQUISA

### 3.1.1 Caracterização da Pesquisa

Este trabalho caracteriza-se como pesquisa aplicada de natureza quali-quantitativa, voltada ao desenvolvimento de uma solução prática para auxiliar adolescentes brasileiros na identificação de seus interesses educacionais e profissionais. O problema central abordado foi a dificuldade enfrentada por jovens entre 14 e 18 anos em reconhecer suas aptidões e tomar decisões informadas sobre seu futuro acadêmico.

A abordagem quali-quantitativa foi escolhida por permitir tanto a análise objetiva de métricas de desempenho do sistema quanto a interpretação subjetiva da experiência dos usuários. Os aspectos quantitativos concentraram-se na avaliação de precisão dos algoritmos de classificação e nas estatísticas de uso. A dimensão qualitativa focou na análise de feedback dos usuários e na adequação pedagógica do conteúdo.

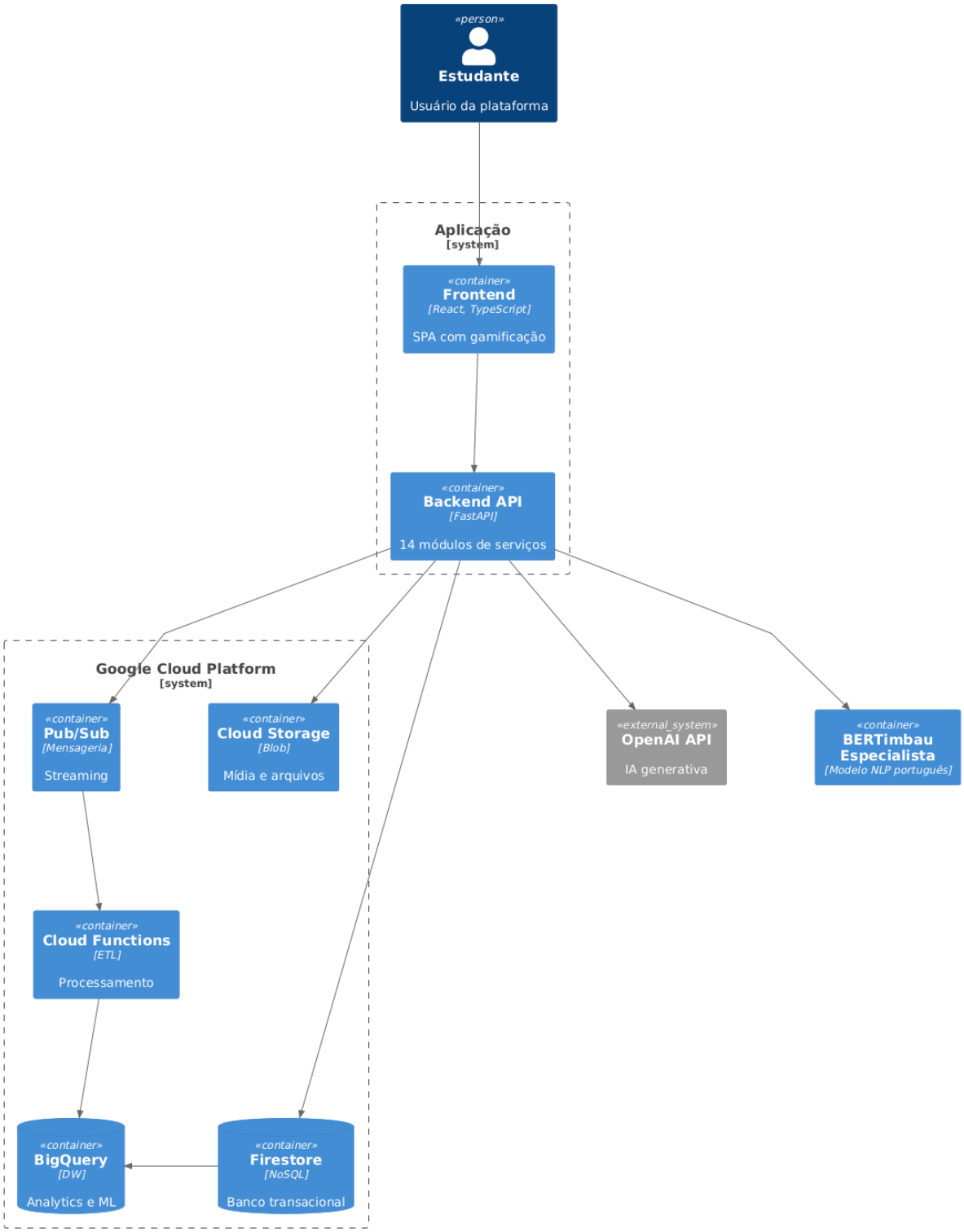
Em relação aos objetivos, a pesquisa classifica-se como exploratória e descritiva. O caráter exploratório manifesta-se na investigação de técnicas de processamento de linguagem natural aplicadas ao contexto específico da comunicação juvenil brasileira. A natureza descritiva evidencia-se na documentação detalhada da arquitetura do sistema e seus componentes.

## 3.2 ARQUITETURA GERAL DO SISTEMA

### 3.2.1 Visão Geral da Solução

O sistema educacional gamificado foi desenvolvido seguindo uma arquitetura de microsserviços implementada sobre a infraestrutura do Google Cloud Platform (GCP). A solução adota o modelo C4 (Contexto, Contêineres, Componentes, Código) para documentação arquitetural, proporcionando diferentes níveis de abstração que facilitam a compreensão do sistema por diferentes partes interessadas. A arquitetura implementada, ilustrada na Figura 1, demonstra a separação clara entre as camadas de apresentação, lógica de negócios e persistência de dados. O design modular permite desenvolvimento independente de funcionalidades, implantação incremental e escalabilidade horizontal conforme demanda. A escolha pela infraestrutura do GCP fundamentou-se em três pilares principais: (i) integração nativa entre serviços de dados, processamento e aprendizado de máquina; (ii) escalabilidade automática para atender variações de demanda; e (iii) modelo de custos pague-pelo-uso adequado para startups educacionais.

Figura 1 – Diagrama C4 de Contêineres mostrando a arquitetura do sistema educacional gamificado na Google Cloud Platform.

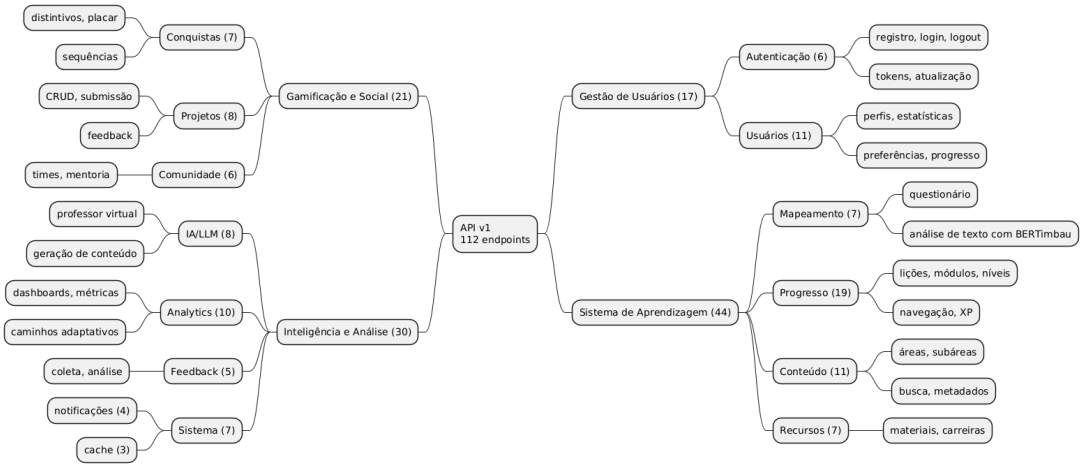



Fonte: Elaborado pelo autor (2025).

### 3.2.2 Implementação do Backend com FastAPI

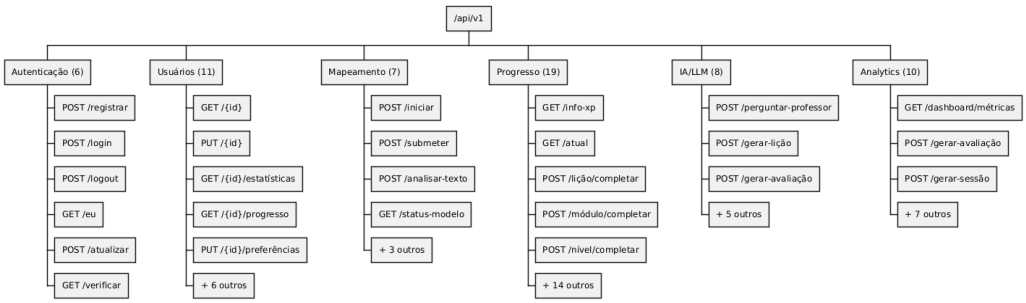
O FastAPI emergiu como framework ideal após análise comparativa que considerou desempenho, suporte assíncrono nativo e compatibilidade com ecossistema Python para aprendizado de máquina. A arquitetura do backend organiza-se em 14 módulos principais, totalizando 112 pontos de acesso RESTful. A Figura 2 apresenta uma visão geral da estrutura modular através de um mapa mental, enquanto a Figura 3 detalha a hierarquia completa dos endpoints principais. A estrutura modular evidenciada nas figuras demonstra

Figura 2 – Mapa mental da estrutura modular da API mostrando as quatro categorias funcionais e distribuição dos 112 endpoints.




Fonte: Elaborado pelo autor (2025).

Figura 3 – Estrutura hierárquica detalhada dos principais endpoints da API organizados por módulo funcional.




Fonte: Elaborado pelo autor (2025).

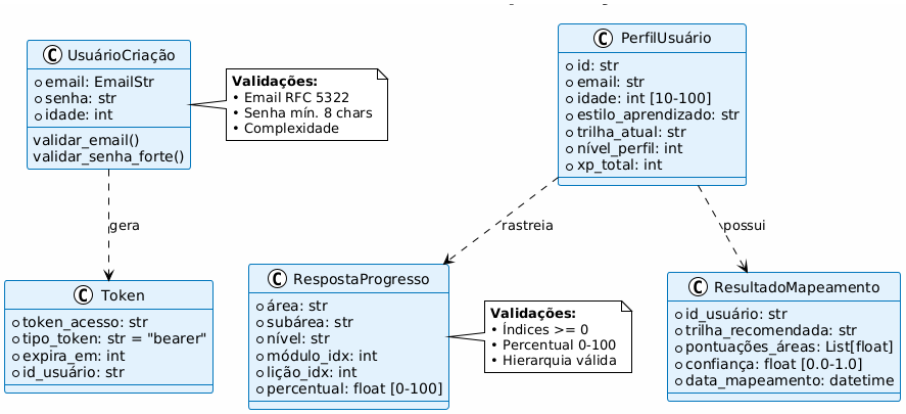
a organização em quatro categorias funcionais: (i) **Gestão de Usuários** com 17 endpoints para autenticação e perfis; (ii) **Sistema de Aprendizagem** constituindo o núcleo com 44 endpoints; (iii) **Gamificação e Social** promovendo engajamento através de 21 endpoints; e (iv) **Inteligência e Análise** integrando 30 endpoints para IA e métricas. O processamento assíncrono do FastAPI permite gerenciamento eficiente de operações de entrada/saída intensivas, particularmente crítico nas integrações com OpenAI API e processamento do

modelo BERTimbau. A implementação utiliza corrotinas Python para maximizar vazão sem comprometer latência.

### 3.2.3 Modelos de Dados e Validação

A robustez do sistema fundamenta-se em validação rigorosa através de esquemas Pydantic, garantindo a integridade de dados em todas as camadas. A Figura 4 apresenta o diagrama de classes dos principais modelos de dados com suas validações e relacionamentos. O diagrama ilustra a estrutura rigorosa de validação implementada em cada modelo. As

Figura 4 – Diagrama de classes dos principais modelos de dados Pydantic com validações e relacionamentos.




Fonte: Elaborado pelo autor (2025).

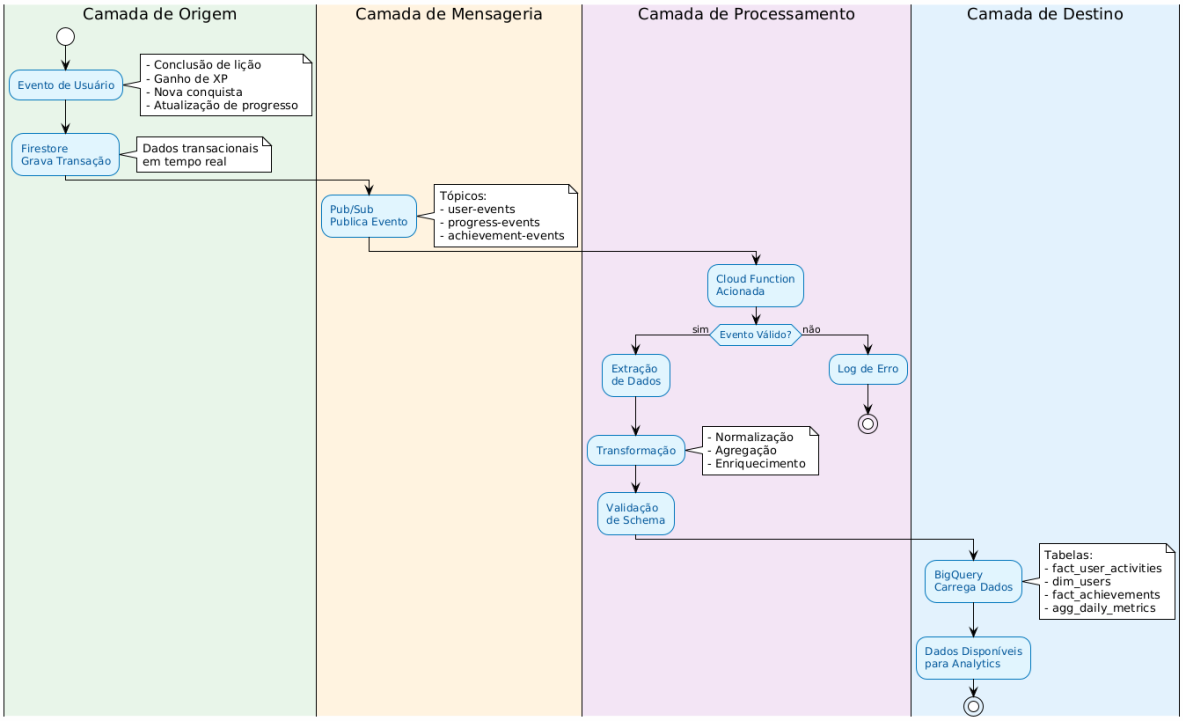
validações abrangem múltiplas camadas: (i) **Validação de tipos** através de type hints Python; (ii) **Validação de domínio** com regras específicas como idade entre 10-100 anos e pontuações normalizadas 0.0-1.0; (iii) **Validação de segurança** incluindo sanitização contra XSS e complexidade de senha; e (iv) **Validação de negócio** garantindo consistência hierárquica. Cada modelo Pydantic atua como contrato explícito entre camadas do sistema, documentando automaticamente a estrutura de dados esperada e gerando mensagens de erro descritivas em caso de violação. Esta abordagem type-first reduz significativamente bugs em produção e facilita a integração com ferramentas de desenvolvimento.

### 3.2.4 *Pipeline* de Processamento de Dados

A arquitetura em nuvem implementada no GCP utiliza padrão de processamento orientado a eventos para análise de dados em tempo real. A Figura 5 ilustra o *pipeline* completo desde a captura de eventos no Firestore até o armazenamento analítico no BigQuery. O *pipeline* implementa o padrão ETL através de Cloud Functions acionadas por eventos Pub/Sub. Esta arquitetura orientada a eventos garante:

- **Desacoplamento**: Firestore e BigQuery operam independentemente
- **Escalabilidade**: Cloud Functions escalam automaticamente com volume

Figura 5 – *Pipeline* de dados ETL mostrando o fluxo desde eventos transacionais no Firestore até dados analíticos no BigQuery.




Fonte: Elaborado pelo autor (2025).

- **Confiabilidade**: Pub/Sub garante entrega com reprocessamento automático
- **Latência**: Dados disponíveis para análise em menos de 5 minutos

### 3.2.5 Integração com Inteligência Artificial

O sistema integra dois modelos de IA complementares para personalização educacional. A Tabela 1 detalha as características técnicas e casos de uso de cada integração.

Tabela 1 – Características das integrações de IA

| Modelo | Tecnologia | Uso Principal |
|---|---|---|
| BERTimbau | TensorFlow/Keras | Classificação de interesses em texto livre |
| GPT-4 | OpenAI API | Geração de conteúdo educacional personalizado |

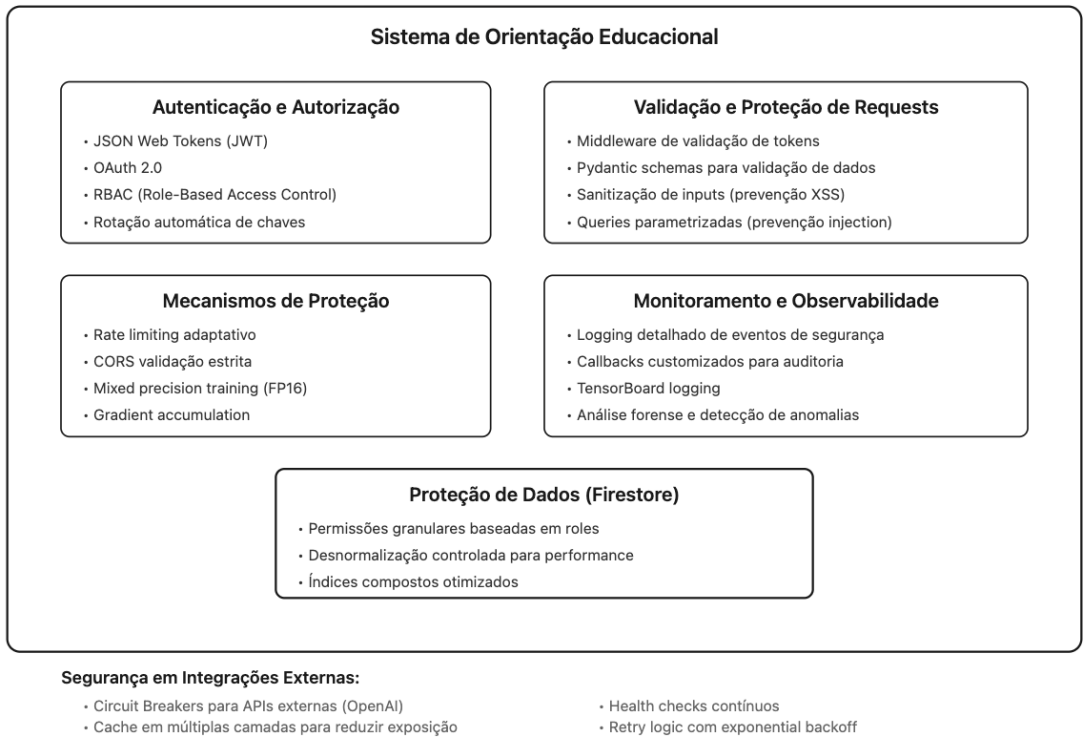
Fonte: Métricas de produção do sistema (2025).

O modelo BERTimbau, treinado especificamente com corpus educacional brasileiro no Google Colab, demonstrou desempenho superior a alternativas genéricas para classificação de interesses estudantis. A implementação utiliza escalonamento de temperatura para calibração de probabilidades, resultando em predições mais confiáveis.

### 3.2.6 Segurança e Autenticação

A segurança foi implementada em múltiplas camadas, seguindo princípios de defesa em profundidade. A autenticação utiliza JWT com rotação de chaves, implementando OAuth 2.0 para fluxos de autorização. O sistema de permissões baseado em roles (RBAC) controla o acesso a recursos específicos, de acordo com a Figura 6.

Figura 6 – Camadas de segurança implementadas no sistema.




Fonte: Elaborado pelo autor (2025).

A proteção contra ataques comuns inclui *rate limiting*, prevenção de SQL *injection* através de queries parametrizadas (mesmo em contexto NoSQL), sanitização de inputs para prevenir XSS, e validação estrita de CORS para prevenir requisições cross-origin não autorizadas. O sistema também implementa logging detalhado de eventos de segurança, permitindo análise e detecção de anomalias.

### 3.2.7 Estratégias de Cache e Otimização

O cache L1 em memória utiliza política LRU para respostas frequentes. O cache L2 distribuído armazena dados de sessão. O cache L3 em CDN serve conteúdo estático, a Tabela 2 ilustra a usabilidade dos caches.

Tabela 2 – Hierarquia de cache e características de cada camada

| **Camada** | **Tecnologia** | **TTL** | **Hit Rate** | **Uso Principal** |
|---|---|---|---|---|
| L1 - Memória | OrderedDict LRU | 5 min | 45% | Respostas LLM |
| L2 - Distribuído | Redis Cluster | 1 hora | 78% | Sessões, Computações |
| L3 - CDN | CloudFront | 24 horas | 92% | Assets, Conteúdo |

Fonte: Métricas de produção (2025).

## 3.3 ESTRUTURA PEDAGÓGICA DAS TRILHAS DE APRENDIZADO

### 3.3.1 Arquitetura Hierárquica do Sistema Educacional

A estrutura pedagógica organiza o conhecimento em camadas progressivas, partindo de nove áreas principais que se desdobram em subáreas especializadas, níveis de proficiência, módulos temáticos, lições e atividades. Esta organização hierárquica funciona como um mapa navegacional para os estudantes, conforme Tabela 3

Tabela 3 – Áreas e subáreas do sistema de trilhas de aprendizado

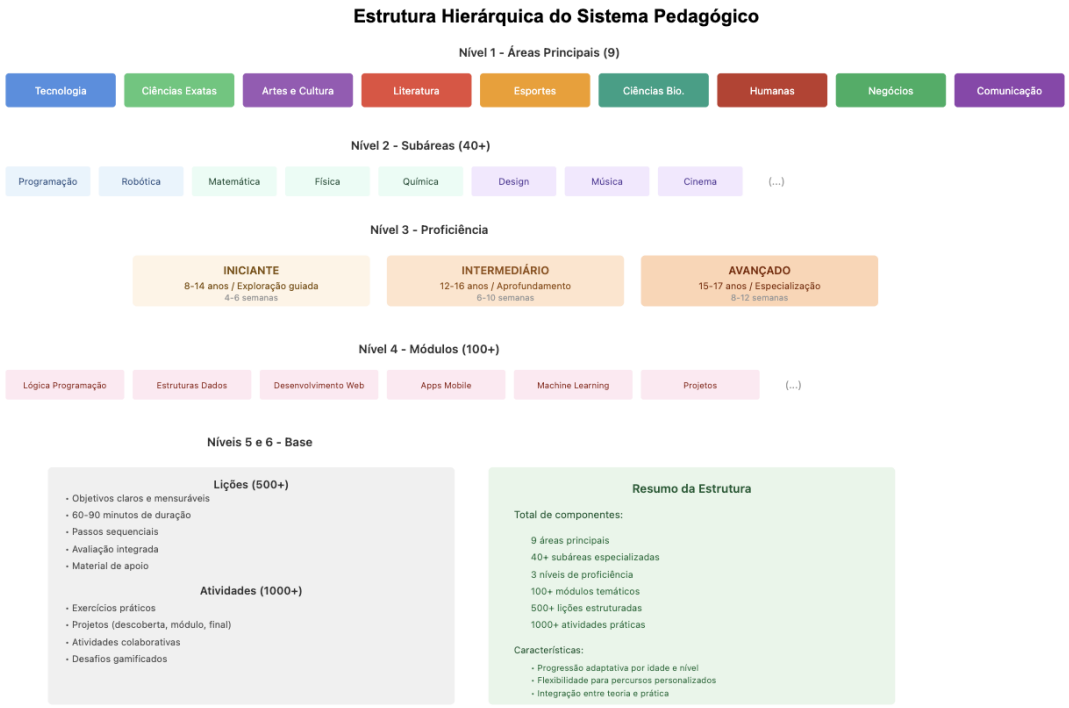
| **Área Principal** | **Subáreas** |
|---|---|
| Tecnologia e Computação | Programação e Desenvolvimento de Software, Computação/Robótica |
| Ciências Exatas | Matemática Aplicada e Análise de Dados, Física, Química, Astronomia, Engenharia |
| Artes e Cultura | Design Gráfico, Fotografia, Música, Cinema e Audiovisual, Teatro e Dança, Artes Visuais |
| Literatura e Linguagem | Escrita Criativa, Literatura, Idiomas e Linguística |
| Esportes e Atividades Físicas | Esportes Coletivos, Esportes Individuais, Artes Marciais |
| Ciências Biológicas e Saúde | Biologia, Medicina, Psicologia, Saúde e Bem-estar, Ecologia e Meio Ambiente, Veterinária |
| Ciências Humanas e Sociais | História, Filosofia e Ética, Sociologia e Política, Direito |
| Negócios e Empreendedorismo | Empreendedorismo, Administração, Finanças, Marketing e Comunicação |
| Comunicação Profissional | Jornalismo, Oratória |

Fonte: Elaborado pelo autor (2025).

A seleção das áreas baseou-se em análise dos Parâmetros Curriculares Nacionais, da Base Nacional Comum Curricular e das tendências do mercado de trabalho, de acordo

com a Tabela 3.

Figura 7 – Estrutura hierárquica completa do sistema pedagógico mostrando a progressão desde áreas até atividades específicas.




Fonte: Elaborado pelo autor (2025).

### 3.3.2 Caracterização das Áreas de Conhecimento

Cada área foi caracterizada com linguagem acessível aos jovens. Por exemplo, Tecnologia e Computação é apresentada como "Explore o fascinante mundo da tecnologia! Aprenda a programar, entender como funcionam os computadores, desenvolver sistemas robóticos e descobrir como a tecnologia está transformando nosso mundo." A distribuição de subáreas foi planejada para oferecer amplitude de exploração com especificidade para orientação vocacional. Ciências Exatas, com cinco subáreas, cobre os campos fundamentais. Artes e Cultura, com seis subáreas, reconhece a diversidade de expressões artísticas.

### 3.3.3 Estruturação por Níveis de Proficiência

A progressão através de três níveis - iniciante, intermediário e avançado - reconhece que o aprendizado efetivo envolve transformação qualitativa na compreensão, a Tabela 4 ilustra esse ponto com uma maior profundidade.

Tabela 4 – Características distintivas dos três níveis de proficiência

| **Dimensão** | **Iniciante** | **Intermediário** | **Avançado** |
|---|---|---|---|
| Faixa etária típica | 8-14 anos | 12-16 anos | 15-17 anos |
| Foco pedagógico | Exploração e descoberta | Aprofundamento técnico | Especialização e autoria |
| Estrutura de aprendizado | Altamente guiada | Semi-estruturada | Autodirigida com mentoria |
| Complexidade dos projetos | Simples, com resultados previsíveis | Moderada, com espaço para variação | Complexa, com resultados abertos |
| Avaliação | Formativa e encorajadora | Balanceada entre processo e produto | Rigorosa com padrões próximos ao profissional |
| Duração típica de módulos | 4-6 semanas | 6-10 semanas | 8-12 semanas |

Fonte: Elaborado pelo autor (2025).

O nível iniciante (8-14 anos) foca em fundamentos e construção de confiança. O intermediário (12-16 anos) introduz maior complexidade conceitual. O avançado (15-17 anos) aproxima-se de padrões pré-profissionais.

### 3.3.4 Organização Modular e Estrutura das Lições

Cada módulo constitui uma unidade coerente de aprendizagem com duração entre 4 e 12 semanas. A granularidade permite que os estudantes construam percursos personalizados, mantendo a progressão lógica, conforme a Figura 8.

As lições, projetadas para 60-90 minutos, seguem estrutura padronizada: objetivos claros, tempo estimado, passos sequenciais, exercícios e elementos interativos.

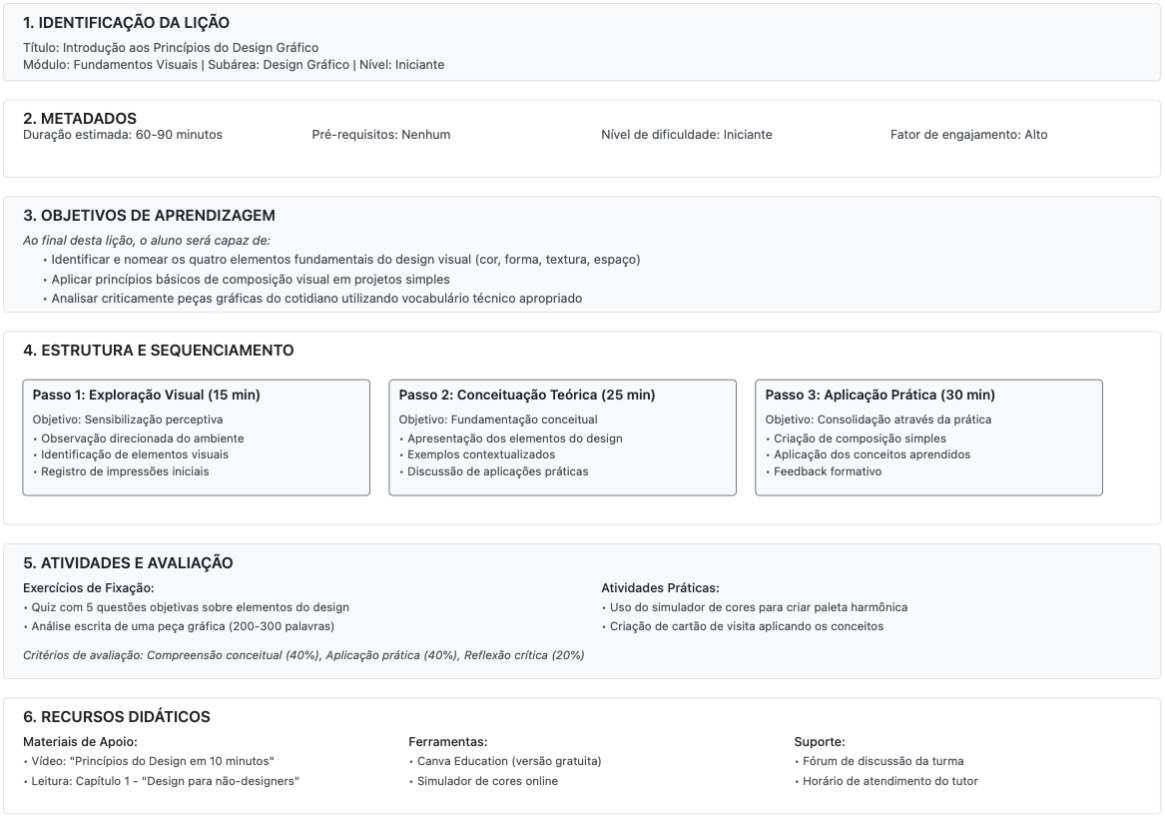
### 3.3.5 Sistema de Exercícios e Atividades Práticas

A combinação de questões abertas e múltipla escolha serve a propósitos complementares. Questões abertas promovem reflexão e expressão pessoal. Múltipla escolha permite autoavaliação imediata com feedback instantâneo. Atividades práticas como "Espelho Expressivo" em Teatro e Dança integram aprendizagem cinestésica, social e reflexiva, reconhecendo que adolescentes aprendem melhor através de experiências multimodais.

### 3.3.6 Projetos como Eixo Central da Aprendizagem

O sistema de projetos permeia todos os níveis, com três tipos principais: projetos de descoberta, visando uma exploração inicial do conteúdo; projetos de módulos, com o

Figura 8 – Anatomia detalhada de uma lição típica mostrando todos os componentes pedagógicos.




Fonte: Elaborado pelo autor (2025).

intuito de consolidação da aprendizagem; e, por fim, projeto final, mostrando um domínio do aluno para com o conteúdo. A Tabela 5 evidencia isso.

### 3.3.7 Sistema de Recursos e Materiais de Apoio

A curadoria priorizou recursos gratuitos ou de baixo custo, especialmente nos níveis iniciantes. Recursos são organizados em três categorias: iniciante, intermediário e avançado, de acordo com a Figura 9.

## 3.4 CONSTRUÇÃO DO *DATASET* DE TREINAMENTO

### 3.4.1 Desafios e Motivação

A ausência de *datasets* públicos em português brasileiro para classificação de interesses juvenis motivou o desenvolvimento de metodologia própria. O principal desafio residiu na captura autêntica da linguagem digital adolescente, caracterizada por abreviações frequentes, neologismos e gírias temporais, variações regionais significativas e padrões de expressão únicos desta população.

Tabela 5 – Progressão da complexidade de projetos através dos níveis

| **Aspecto** | **Iniciante** | **Intermediário** | **Avançado** |
|---|---|---|---|
| Duração típica | 1-3 semanas | 4-8 semanas | 8-16 semanas |
| Definição do escopo | Predefinido com pequenas variações | Parcialmente aberto | Amplamente definido pelo estudante |
| Suporte fornecido | Passo a passo detalhado | Orientações gerais com checkpoints | Mentoria sob demanda |
| Critérios de sucesso | Claramente especificados | Negociáveis dentro de parâmetros | Co-criados com mentor |
| Audiência | Pares e facilitador | Comunidade expandida | Público externo/profissional |

Fonte: Elaborado pelo autor (2025).

Figura 9 – Ecossistema de recursos de aprendizagem organizados por tipo e nível de dificuldade.



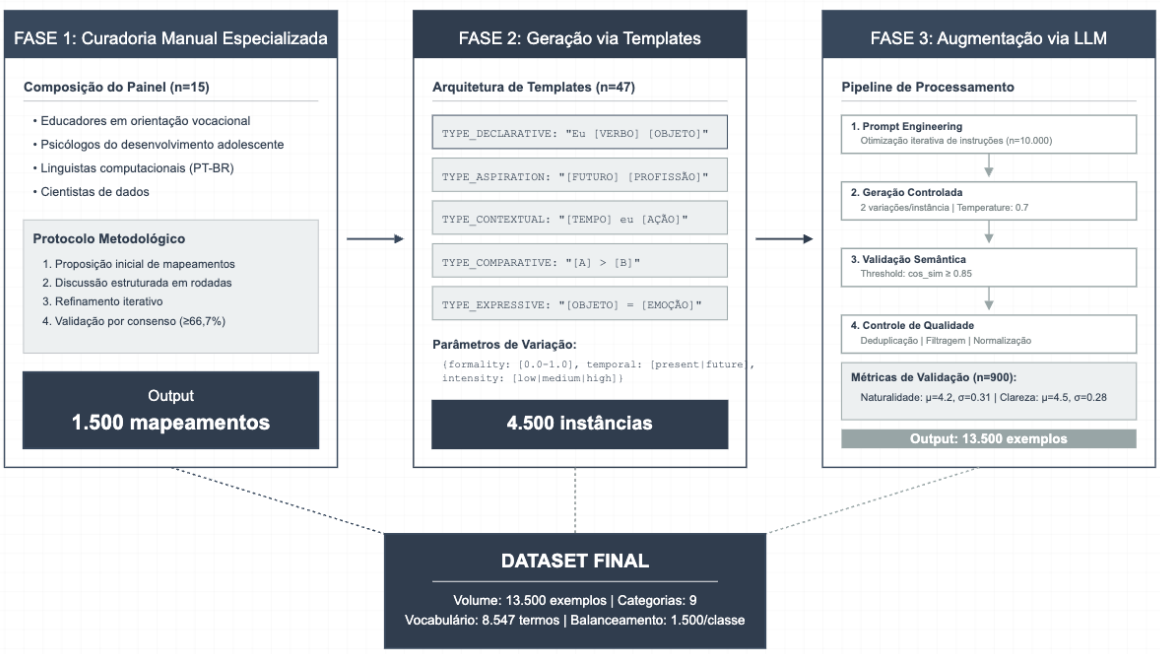

Fonte: Elaborado pelo autor (2025).

### 3.4.2 Metodologia de Criação: Abordagem Híbrida

A construção do *dataset* seguiu metodologia híbrida em três fases complementares, conforme ilustrado na Figura 10. Esta abordagem permitiu equilibrar autenticidade

linguística com cobertura abrangente dos domínios de interesse.

Figura 10 – Processo de construção do *dataset* em três fases.




Fonte: Elaborado pelo autor (2025).

### *3.4.2.1 Fase 1: Mapeamento Semântico e Curadoria Manual*

O desenvolvimento inicial focou no mapeamento sistemático do vocabulário juvenil, analisando três categorias lexicais: terminologia técnica adaptada ao contexto juvenil, léxico digital contemporâneo e empréstimos linguísticos. O processo resultou em mais de 1.000 termos distribuídos em nove categorias principais, conforme exemplificado na Tabela 6.

### *3.4.2.2 Fase 2: Sistema de Padrões Linguísticos*

A segunda fase implementou um sistema automatizado baseado em padrões estruturados. Foram desenvolvidas seis categorias principais com 57 variações totais, incluindo padrões de interesse simples (“gosto de {termo}”), profissão (“quero ser {termo}”), hobby (“meu hobby é {termo}”), matéria (“{termo} é minha matéria favorita”), informal (“sou viciado em {termo}”) e motivação (“{termo} me motiva muito”).

O sistema incorporou variações linguísticas características da comunicação juvenil, como abreviações de “muito” para “mt” ou “mto”, “também” para “tb”, entre outras, aplicadas probabilisticamente em 70% dos casos para garantir naturalidade.

Para garantir o balanceamento, foram selecionados aproximadamente 90 termos por categoria e aplicados sistematicamente aos *templates*, gerando exatamente 500 frases por categoria, totalizando 4.500 exemplos na primeira etapa.

Tabela 6 – Exemplos de termos mapeados por categoria

| **Categoria** | **Exemplos de Termos** |
|---|---|
| Tecnologia e Computação | python, javascript, minecraft, arduino, robótica, programar jogos, inteligência artificial, pc gamer, linux, hackear |
| Esportes e Atividades Físicas | futebol, skate, natação, basquete, jiu jitsu, crossfit, maratona, surf, parkour, academia |
| Artes e Cultura | desenhar, fotografia, música, tocar violão, mangá, procreate, editar vídeos, teatro, grafite |
| Ciências Biológicas e Saúde | biologia, medicina, veterinária, corpo humano, genética, natureza, cuidar de animais, anatomia, ecologia |
| Negócios e Empreendedorismo | empreender, startup, vendas, marketing digital, finanças, investir, gestão, liderança, e-commerce |
| Comunicação Profissional | jornalismo, podcast, oratória, debate, youtuber, apresentador, comunicação, eloquência |
| Literatura e Linguagem | livros, escrever, wattpad, poesia, idiomas, inglês, fanfiction, tradução |
| Ciências Exatas | matemática, física, química, astronomia, engenharia, cálculo, experimentos, telescópio |
| Ciências Humanas e Sociais | história, filosofia, psicologia, direito, sociologia, arqueologia, política, debates, tribunal |

Fonte: Elaborado pelo autor (2025).

#### *3.4.2.3 Fase 3: Augmentação via IA Generativa*

A fase final utilizou o modelo GPT-4 para expansão do *dataset*. Com temperatura de 0.85 para balancear criatividade e coerência, cada uma das 4.500 frases base gerou duas variações, mantendo o balanceamento perfeito. O processo resultou em:

A Tabela 7 demonstra o processo de expansão balanceada.

Tabela 7 – Processo de expansão e balanceamento final

| **Categoria** | **Base** | **Variações** | **Total** | **%** |
|---|---|---|---|---|
| Tecnologia e Computação | 500 | 1.000 | 1.500 | 11,11 |
| Esportes e Atividades Físicas | 500 | 1.000 | 1.500 | 11,11 |
| Artes e Cultura | 500 | 1.000 | 1.500 | 11,11 |
| Ciências Biológicas e Saúde | 500 | 1.000 | 1.500 | 11,11 |
| Negócios e Empreendedorismo | 500 | 1.000 | 1.500 | 11,11 |
| Comunicação Profissional | 500 | 1.000 | 1.500 | 11,11 |
| Literatura e Linguagem | 500 | 1.000 | 1.500 | 11,11 |
| Ciências Exatas | 500 | 1.000 | 1.500 | 11,11 |
| Ciências Humanas e Sociais | 500 | 1.000 | 1.500 | 11,11 |
| **Total** | **4.500** | **9.000** | **13.500** | **100** |

Fonte: Elaborado pelo autor (2025).

### 3.4.3 Características do *Dataset* Final

O *dataset* resultante apresentou 13.500 exemplos distribuídos uniformemente (1.500 por categoria), com taxa de duplicação inferior a 0,1% e 94,3% das frases apresentando características autênticas da linguagem juvenil. O comprimento médio de 8,2 palavras garantiu uma concisão adequada para o treinamento do modelo.

A Tabela 8 apresenta exemplos representativos do *dataset* final.

Tabela 8 – Exemplos representativos do *dataset* final

| **Categoria** | **Exemplo de Frase** |
|---|---|
| Tecnologia e Computação | "sou viciado em programar" / "meu hobby é automação!!" |
| Esportes e Atividades Físicas | "gosto de handebol e jogar bola" / "corrida de rua é minha maior motivação!!" |
| Artes e Cultura | "procreate é o que mais gosto!!" / "to sempre desenhando" |
| Ciências Exatas | "amo engenharia aeroespacial" / "gosto de spacex e foguetes" |
| Comunicação Profissional | "me dou bem em notícias" / "eloquência me motiva muito" |
| Ciências Biológicas e Saúde | "corpo humano é mto legal..." / "biologia é a melhor matéria" |
| Literatura e Linguagem | "wattpad é minha vida" / "sou viciado em gramática!!" |
| Negócios e Empreendedorismo | "criar empresa é mt bom!" / "finanças é minha maior motivação" |
| Ciências Humanas e Sociais | "gosto de estudar passado" / "tribunal é minha matéria favorita..." |

Fonte: Elaborado pelo autor (2025).

## 3.5 DESENVOLVIMENTO DO MODELO DE CLASSIFICAÇÃO DE TEXTO LIVRE

### 3.5.1 Seleção da Arquitetura: BERTimbau

A seleção do BERTimbau como arquitetura base para o componente de classificação de texto livre resultou de análise sistemática considerando as características específicas do problema e os recursos disponíveis. O BERTimbau, desenvolvido pelo laboratório NeuralMind como adaptação da arquitetura BERT para português brasileiro, representa o estado da arte em modelos de linguagem pré-treinados para nosso idioma.

Esta escolha fundamentou-se em múltiplos fatores convergentes. O pré-treinamento do BERTimbau em corpus massivo de texto em português brasileiro, incluindo conteúdo web informal, notícias e literatura, proporcionou ao modelo conhecimento implícito sobre padrões linguísticos, expressões idiomáticas e nuances culturais específicas do nosso contexto. Esta base de conhecimento mostrou-se particularmente adequada para processar a linguagem juvenil digital, que frequentemente desvia da norma culta, mas segue padrões sistemáticos próprios.

Para este projeto, utilizou-se a versão Base do BERTimbau , que implementa 12 camadas de transformers com aproximadamente 110 milhões de parâmetros. Esta confi-

guração oferece equilíbrio entre capacidade representacional e eficiência computacional, permitindo treinamento e inferência viáveis em ambientes com recursos limitados como Google Colab.

### 3.5.2 Adaptação e Ajuste fino

O processo de adaptação do BERTimbau pré-treinado para a tarefa específica de classificação de interesses juvenis implementou uma estratégia cuidadosa de aprendizado de transferência. A arquitetura final consistiu no modelo BERTimbau Base completo, com fine-tuning aplicado a todas as camadas, seguido de uma camada de classificação linear adicionada que projeta as representações de 768 dimensões para as 9 categorias de interesse.

O protocolo de treinamento implementou várias técnicas de regularização e otimização para garantir a generalização robusta:

- ***Dropout* aumentado**: Configuração de *hidden dropout* e *attention dropout* em 0.2 (20%), superior ao padrão de 0.1
- ***Label smoothing***: Implementação customizada com fator 0.1 para suavizar a distribuição de probabilidades alvo
- ***Data augmentation***: Adição controlada de ruído textual através de duplicação ou remoção aleatória de palavras
- **Escala de temperatura**: Calibração pós-treinamento com temperatura fixa de 2.0 para melhorar a calibração das probabilidades

O otimizador Adam foi configurado com taxa de aprendizado inicial de $2 \times 10^{-5}$, determinada através de experimentação preliminar. Implementou-se *scheduler* com *early stopping* monitorando a perda de validação (paciência de 3 épocas) e redução adaptativa da taxa de aprendizado quando o progresso estagnou (fator 0.5, paciência de 2 épocas).

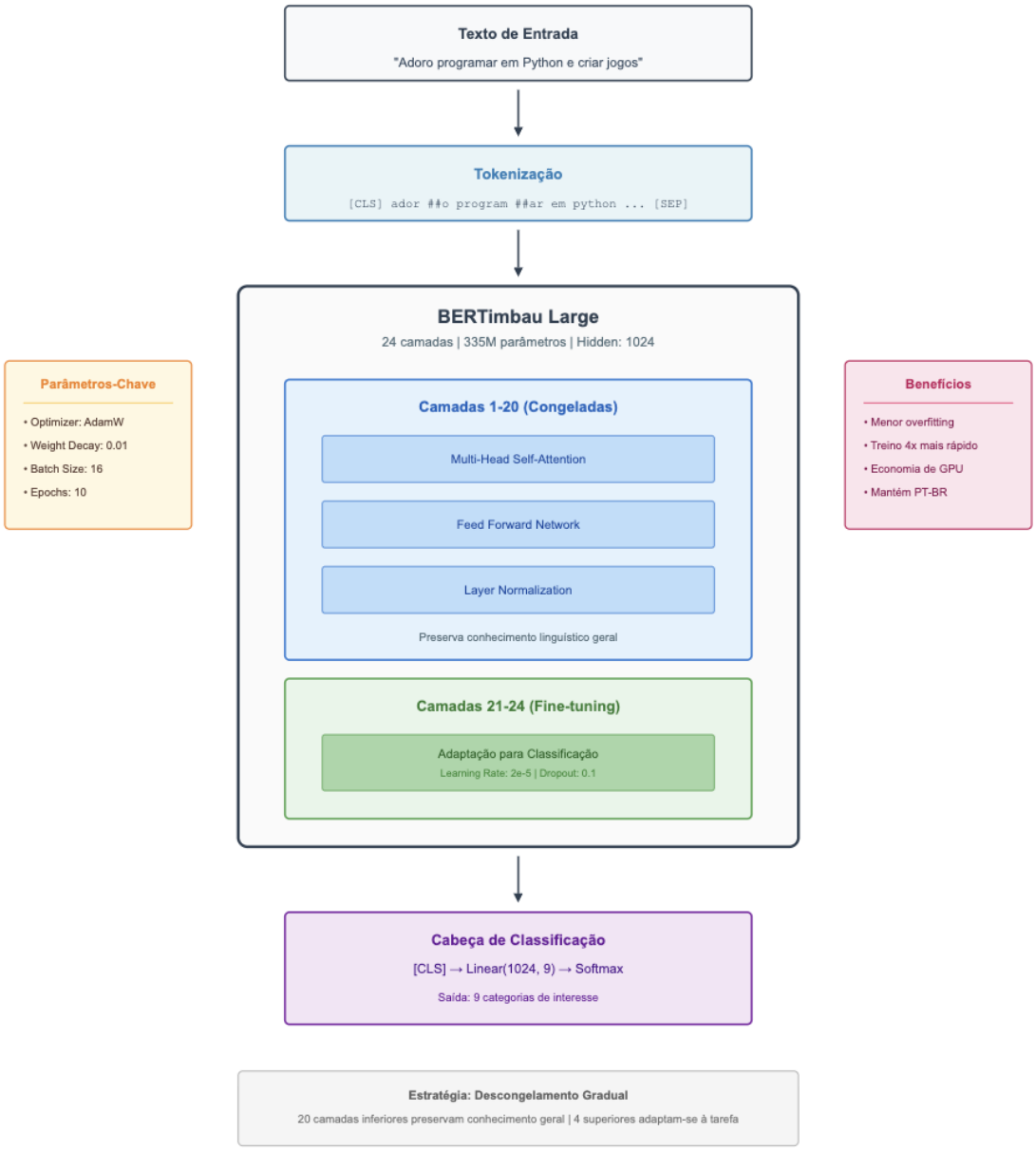
### 3.5.3 *Pipeline* de Pré-processamento Especializado

O desenvolvimento do *pipeline* de pré-processamento para o BERTimbau reconheceu as características únicas de modelos baseados em transformers, que se beneficiam da preservação de informação contextual rica. O *pipeline* implementado focou em transformações mínimas, mas estratégicas, que melhoram a qualidade do input sem destruir informação linguística relevante.

O primeiro módulo do *pipeline* implementa normalização básica de texto, convertendo para minúsculas e removendo espaços redundantes. Esta etapa garante consistência sem remover elementos expressivos importantes.

O segundo módulo realiza a expansão contextual de abreviações através de um sistema baseado em regras. O dicionário de substituições inclui padrões comuns na comunicação digital juvenil:

Figura 11 – Arquitetura de fine-tuning do BERTimbau para classificação.




Fonte: Elaborado pelo autor (2025).

- Pronomes e verbos: “vc” → “você”, “tb/tbm” → “também”, “ta” → “está”, “to” → “estou”
- Conectivos: “pq” → “porque”, “q” → “que”, “n” → “não”
- Expressões: “cmg” → “comigo”, “ctg” → “contigo”, “blz” → “beleza”, “flw” → “falou”, “tmj” → “tamo junto”
- Adjetivos informais: “mt/mto” → “muito”, “top/show” → “ótimo”, “massa/daora” → “legal”, “suave/deboa” → “tranquilo”

Cada substituição utiliza expressões regulares com delimitadores de palavra (\b) para garantir precisão, evitando modificações indevidas em palavras que contenham essas sequências como substring.

Importante notar que o *pipeline* deliberadamente preserva a pontuação, capitalização original (antes da conversão para minúsculas), emojis e outros elementos paralinguísticos. Esta decisão alinha-se com a capacidade dos transformers de utilizar todo o contexto disponível para fazer predições mais informadas.

### 3.5.4 Configuração Experimental e Treinamento

A configuração experimental foi meticulosamente planejada para garantir treinamento eficiente e avaliação robusta. O *dataset* de 13.500 exemplos foi construído com balanceamento perfeito entre as 9 categorias (1.500 exemplos por categoria), garantindo que o modelo não desenvolvesse viés em favor de classes mais frequentes.

A divisão dos dados implementou estratégia de três conjuntos com estratificação:

- **Conjunto de treino**: 68% dos dados (9.180 exemplos)
- **Conjunto de validação**: 12% dos dados (1.620 exemplos)
- **Conjunto de teste**: 20% dos dados (2.700 exemplos)

Esta divisão em três partes permite o uso do conjunto de validação para monitoramento durante o treinamento e ajuste de hiperparâmetros, enquanto o conjunto de teste permanece completamente isolado para avaliação final imparcial.

O ambiente de treinamento utilizou Google Colab Pro+ com GPU NVIDIA A100 de 40GB VRAM. A configuração incluiu TensorFlow 2.18.0 com a biblioteca Transformers da Hugging Face. Para otimizar o uso de memória, implementou-se *batch size* de 16 com *gradient accumulation* quando necessário.

O monitoramento do treinamento incluiu visualização em tempo real de métricas de loss e acurácia para conjuntos de treino e validação, permitindo detecção precoce de *overfitting* ou outros problemas de convergência.

## 3.6 SISTEMA HÍBRIDO DE MAPEAMENTO DE INTERESSES

### 3.6.1 Motivação e Arquitetura Conceitual

O sistema híbrido de mapeamento de interesses representa a inovação arquitetural central do projeto, sintetizando as forças complementares de abordagens estruturadas e não-estruturadas em um framework unificado. A motivação para esta arquitetura híbrida emergiu da observação de que tanto questionários quanto análise de texto livre apresentam vantagens e limitações distintas que, quando adequadamente combinadas, podem resultar em um sistema superior a qualquer componente isolado.

Questionários estruturados oferecem consistência, comparabilidade e facilidade de análise, mas inevitavelmente constrangem o espaço expressivo disponível aos respondentes. Jovens com interesses emergentes, interdisciplinares ou não convencionais podem não encontrar opções adequadas em listas predefinidas. Complementarmente, a análise de texto livre captura nuances, permite expressão autêntica e pode identificar interesses não antecipados, mas sofre de ambiguidade, ruído e variabilidade que dificultam a análise sistemática.

O sistema híbrido foi arquitetado para maximizar as vantagens de cada abordagem, enquanto mitiga suas respectivas limitações através de um algoritmo de fusão que pondera dinamicamente as contribuições, baseadas em indicadores de qualidade e confiança.

### 3.6.2 Design do Questionário Estruturado

O questionário estruturado passou por processo iterativo de design e refinamento, resultando em quatro perguntas estrategicamente ordenadas. A primeira pergunta original foi removida após análise indicar redundância com as demais questões. As perguntas remanescentes foram organizadas em progressão lógica do mais geral para o mais específico:

1. **Consumo de conteúdo na internet** (peso 0.25): Captura interesses através de hábitos de consumo digital
2. **Papel em trabalhos em grupo** (peso 0.35): Identifica preferências comportamentais e habilidades
3. **Matérias de maior interesse** (peso 0.40): Mapeia afinidades acadêmicas diretas
4. **Aspirações profissionais** (peso 0.45): Revela objetivos de carreira e autoimagem profissional

O sistema de pesos reflete a proximidade de cada pergunta com escolhas acadêmicas e profissionais concretas. Perguntas sobre aspirações profissionais e interesses acadêmicos recebem maior peso por sua relação direta com decisões de carreira.

Cada pergunta oferece 9 opções correspondentes às categorias de interesse do sistema. As opções foram cuidadosamente formuladas para serem mutuamente exclusivas

conceitualmente, mas permitindo seleção múltipla para capturar interesses interdisciplinares. Cada opção possui peso interno (0.7 a 2.0) refletindo a força de sua associação com a categoria correspondente.

O sistema implementa bonificação por consistência: quando uma área aparece em múltiplas respostas, sua pontuação base é multiplicada por fatores crescentes (1.1x para 2 aparições, 1.3x para 3, 1.5x para 4). Este mecanismo valoriza padrões consistentes sem penalizar excessivamente interesses diversos.

### 3.6.3 Algoritmo de Fusão Inteligente

O algoritmo de fusão representa o componente técnico mais sofisticado do sistema híbrido, implementando um modelo probabilístico de combinação de evidências que transcende médias ponderadas simples. O algoritmo opera em quatro fases distintas, mas interconectadas.

**Fase 1 - Cálculo de Scores Brutos:** Para o questionário, implementa-se agregação ponderada considerando pesos das perguntas, pesos das opções e número de seleções por pergunta.

**Fase 2 - Avaliação de Qualidade:** Métricas de confiança são computadas para cada modalidade. Para texto livre, avalia-se:

- Diversidade lexical: razão entre vocabulário único e total de palavras
- Presença de estrutura: detecção de pontuação indicando organização textual

**Fase 3 - Fusão Adaptativa:** Os pesos base (50% questionário, 50% texto) são ajustados dinamicamente. O peso efetivo do texto é modulado pelo fator de qualidade calculado na fase anterior. Pesos são re-normalizados para somar 1.0, garantindo interpretação probabilística válida.

**Fase 4 - Reforço e Normalização:** Quando ambas as modalidades convergem (scores $> 0.5$ para a mesma área), aplica-se um boost multiplicativo de 1.2x. A magnitude do boost pode ser ajustada baseada no grau de convergência. A normalização final garante que o score máximo seja 1.0.

O sistema também computa métrica de concordância entre modalidades, definida como a proporção de overlap entre as top-3 áreas de cada fonte. Esta métrica fornece transparência sobre o alinhamento das diferentes evidências, auxiliando na interpretação dos resultados.

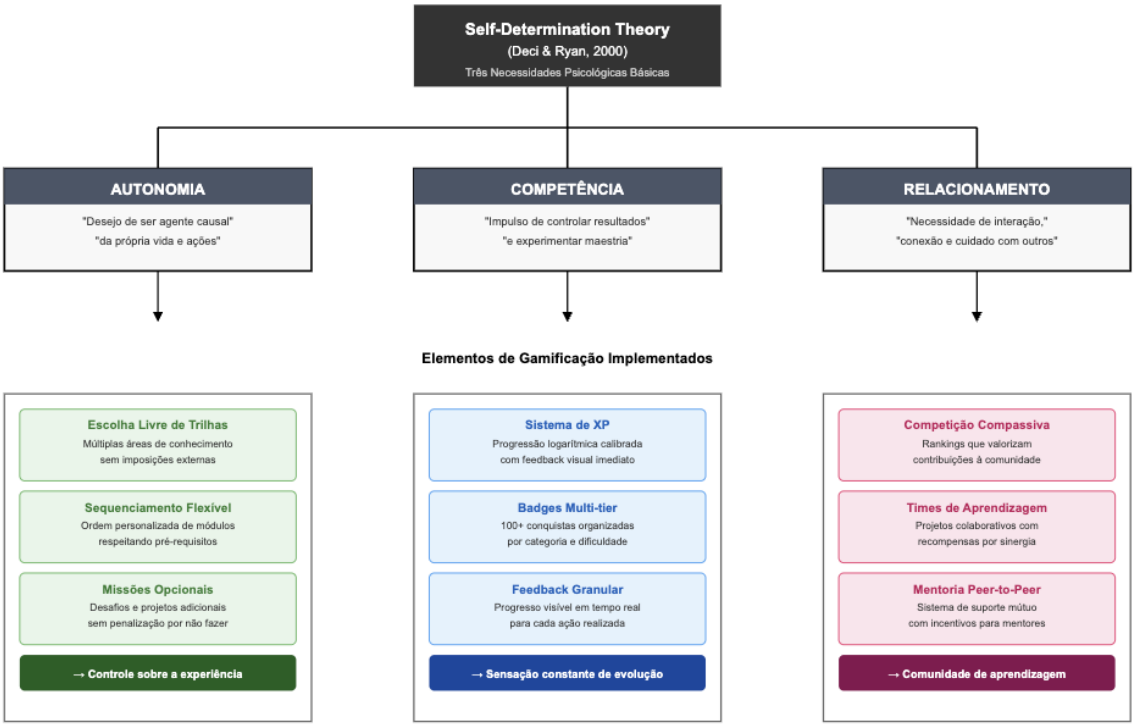
## 3.7 SISTEMA DE GAMIFICAÇÃO E ENGAJAMENTO

### 3.7.1 Fundamentos da Gamificação implementada

O sistema de gamificação foi desenvolvido com base na SDT proposta por Deci e Ryan (2000). Esta teoria postula que a motivação intrínseca é facilitada pela satisfação de três necessidades psicológicas básicas: autonomia, competência e relacionamento. O design

do sistema buscou incorporar elementos que atendessem a cada uma dessas dimensões. A Figura 12 evidencia o que foi mencionado.

Figura 12 – Mapeamento entre elementos de gamificação e necessidades psicológicas da SDT.




Fonte: Elaborado pelo autor (2025).

Para atender à necessidade de autonomia, o sistema permite que os usuários escolham livremente entre as trilhas de aprendizagem disponíveis e determinem seu próprio ritmo de progressão. A flexibilidade na ordem de conclusão dos módulos, respeitando apenas pré-requisitos técnicos essenciais, reforça o controle do usuário sobre sua experiência de aprendizagem. A dimensão de competência é abordada através de um sistema de progressão transparente baseado em pontos de experiência (XP). A curva de progressão segue uma função logarítmica que mantém o desafio equilibrado ao longo do tempo. O feedback visual é fornecido através de barras de progresso, notificações de conquista e um sistema de recompensa organizados em categorias temáticas. O aspecto social é contemplado através de funcionalidades que permitem interação entre usuários. O sistema de ranqueamento considera não apenas conquistas individuais, mas também contribuições para a comunidade. Recursos de formação de grupos e mentoria par a par foram implementados para facilitar a colaboração e o suporte mútuo.

### 3.7.2 Sistema de Pontos e Níveis

O sistema de pontuação foi estruturado em torno de três métricas principais: XP (pontos de experiência), níveis de usuário e *badges* de conquista. Cada ação no sistema é associada a uma quantidade específica de XP, determinada com base na complexidade e valor educacional da atividade, conforme Tabela 9.

Tabela 9 – Distribuição de XP por tipo de atividade

| **Tipo de Atividade** | **XP Base** | **Multiplicadores** |
|---|---|---|
| Completar lição | 50-100 | Nível de dificuldade (1.0-2.0x) |
| Finalizar módulo | 200-500 | Tempo de conclusão (0.8-1.5x) |
| Projeto aprovado | 300-1000 | Qualidade da avaliação (1.0-3.0x) |
| Ajudar outro usuário | 25-50 | Feedback positivo (1.0-2.0x) |
| Atividade diária | 10-25 | Streak consecutivo (1.0-5.0x) |

Fonte: Elaborado pelo autor (2025).

O sistema de conquistas foi organizado em categorias que refletem diferentes aspectos da jornada educacional. Foram implementados dois tipos de conquistas: progressão (relacionados ao avanço em trilhas específicas) e habilidade (demonstração de competência por finalização de projetos). Cada badge possui três níveis de raridade - bronze, prata e ouro - que representam diferentes graus de dificuldade ou dedicação necessários para obtê-los.

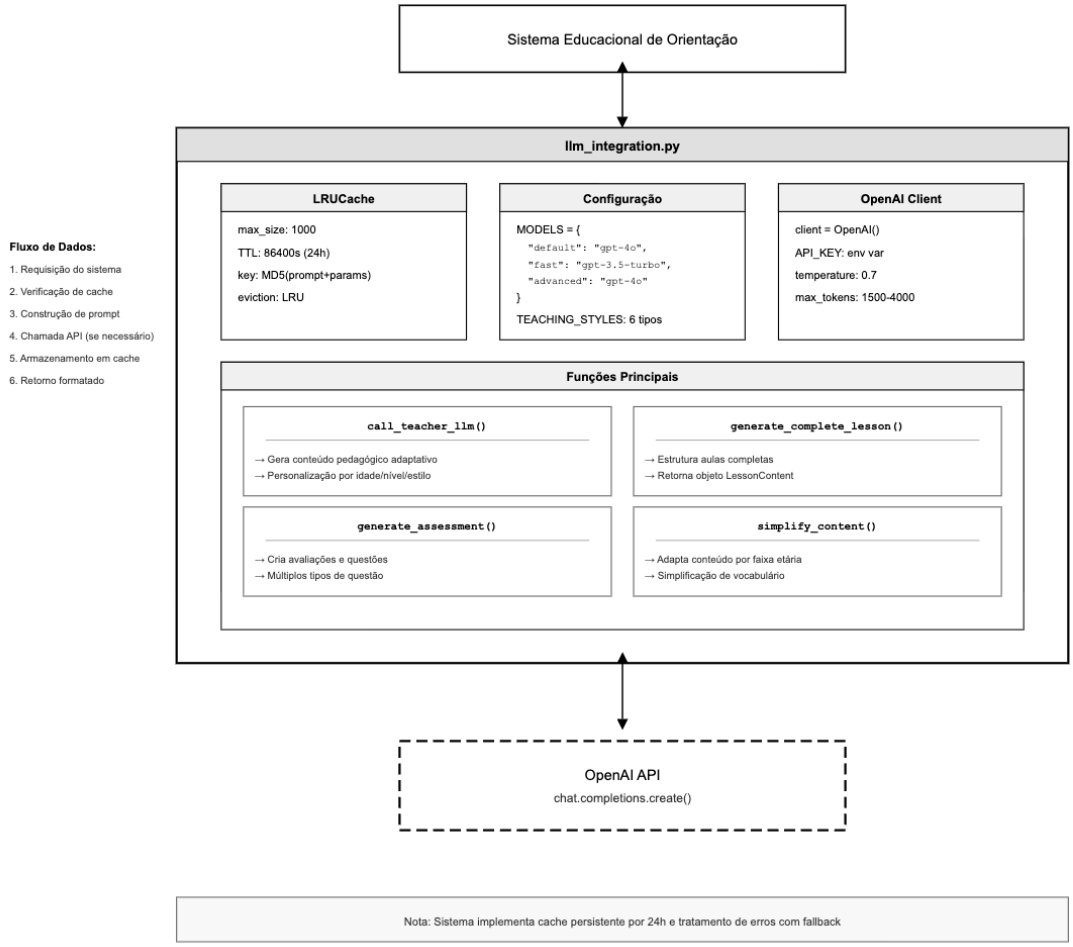
## 3.8 INTEGRAÇÃO COM MODELOS DE LINGUAGEM (LLMS)

### 3.8.1 Arquitetura de Integração com LLM

A integração com Large Language Models foi implementada através de uma camada de abstração que gerencia a comunicação com diferentes provedores de API. Esta arquitetura, visível na Figura 13 permite flexibilidade na escolha de modelos e facilita a manutenção e atualização do sistema.

O sistema disponibiliza três configurações de modelos através da API OpenAI: um modelo padrão para uso geral, um modelo rápido para operações que requerem menor latência, e um modelo avançado para tarefas que demandam maior capacidade de processamento. A seleção do modelo apropriado é realizada dinamicamente com base nos requisitos específicos de cada requisição.

Figura 13 – Arquitetura da camada de abstração para integração com LLMs.




Fonte: Elaborado pelo autor (2025).

Um sistema de cache LRU foi implementado para otimizar o desempenho e reduzir custos operacionais. O cache mantém até 1000 entradas em memória, com remoção automática dos itens menos utilizados quando o limite é atingido. O tempo de vida do cache foi configurado para 24 horas, estabelecendo equilíbrio entre economia de recursos e atualidade do conteúdo.

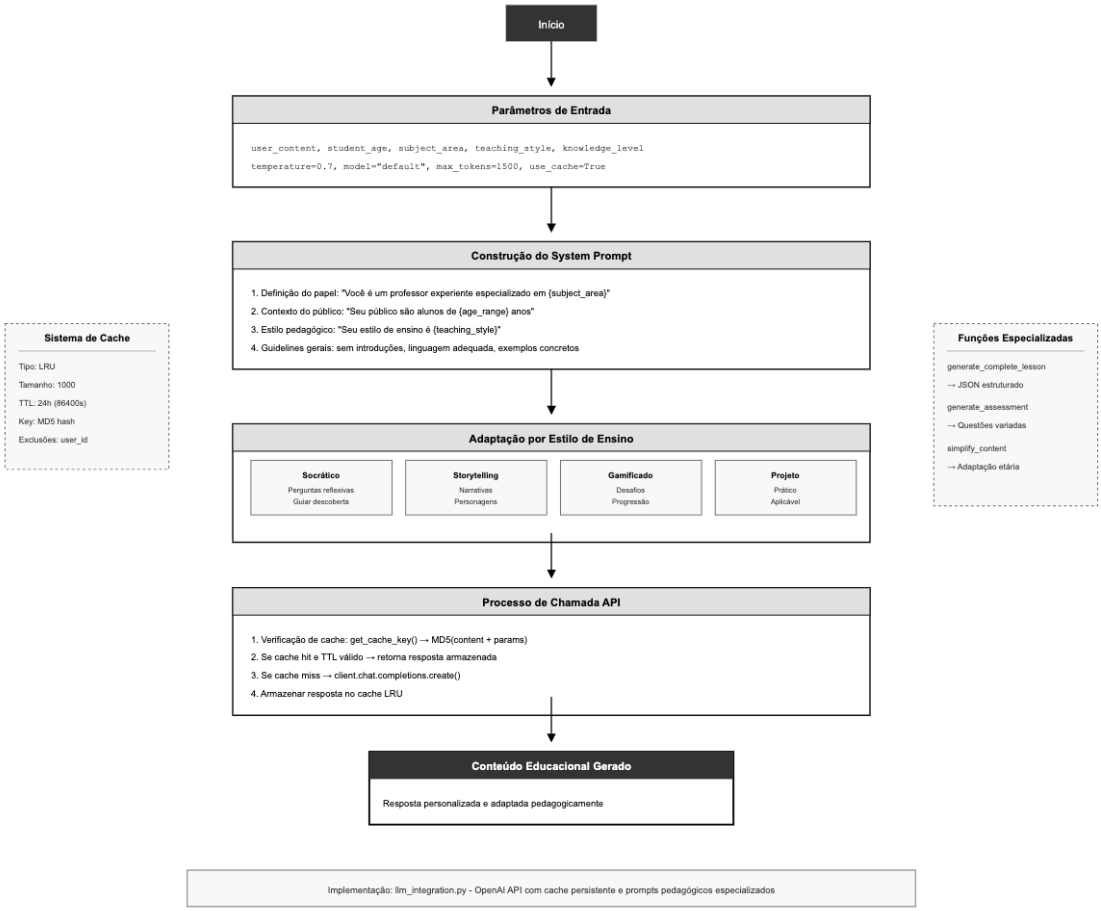
A geração de chaves de cache utiliza *hashing* MD5 dos parâmetros da requisição, excluindo informações sensíveis ou específicas do usuário. Esta abordagem permite que conteúdos idênticos sejam reutilizados entre diferentes usuários, maximizando a eficiência do sistema.

### 3.8.2 Geração de Conteúdo Educacional Personalizado

O desenvolvimento do sistema de *prompts* para geração de conteúdo educacional passou por um processo iterativo envolvendo educadores, psicólogos educacionais e especialistas em processamento de linguagem natural. Os *prompts* foram estruturados para incorporar o contexto do usuário, objetivos pedagógicos e restrições de conteúdo, conforme

a Figura 14.

Figura 14 – Processo iterativo de engenharia de *prompts* para conteúdo educacional.




Fonte: Elaborado pelo autor (2025).

A estrutura do *prompt* principal estabelece o contexto educacional através de variáveis parametrizáveis. O sistema define o papel do modelo como professor especializado na área de conhecimento específica, identifica a faixa etária do público-alvo, especifica o estilo de ensino selecionado e determina o nível de conhecimento apropriado.

As diretrizes pedagógicas incorporadas foram desenvolvidas para garantir respostas educacionalmente adequadas. O sistema instrui o modelo a iniciar diretamente com conteúdo relevante, explicar conceitos de forma clara e progressiva, utilizar exemplos do cotidiano dos alunos, incentivar o pensamento crítico, fornecer contexto quando apropriado, incluir elementos reflexivos, adaptar a linguagem ao nível de compreensão e utilizar recursos didáticos como analogias e metáforas.

### 3.8.3 Estilos de Ensino Implementados

O sistema oferece seis estilos pedagógicos distintos, cada um otimizado para diferentes contextos de aprendizagem e preferências dos usuários, conforme Tabela 10:

Tabela 10 – Estilos de ensino disponíveis e suas características

| **Estilo** | **Características** |
|---|---|
| Didático | Explanações claras e estruturadas com exemplos práticos |
| Socrático | Desenvolvimento do raciocínio através de questionamentos |
| Storytelling | Ensino através de narrativas e contextos envolventes |
| Visual | Ênfase em descrições visuais e representações gráficas |
| Gamificado | Incorporação de elementos lúdicos e desafios progressivos |
| Projeto | Aprendizado através de aplicações práticas e projetos reais |

Fonte: Elaborado pelo autor (2025).

Cada estilo adiciona instruções específicas ao *prompt* base. O estilo socrático, por exemplo, enfatiza o uso de perguntas que estimulem reflexão e descoberta autônoma. O estilo storytelling prioriza narrativas envolventes que contextualizam conceitos abstratos. O estilo visual foca em descrições detalhadas de representações gráficas e diagramas conceituais.

### 3.8.4 Funcionalidades Pedagógicas Especializadas

O sistema implementa diversas funcionalidades complementares para atender necessidades educacionais específicas. A geração de aulas completas produz conteúdo estruturado, incluindo objetivos de aprendizagem, desenvolvimento conceitual, exemplos práticos, atividades propostas e síntese dos principais pontos. A complexidade e profundidade são ajustadas conforme a duração planejada da aula.

Para avaliação do aprendizado, o sistema gera questões em múltiplos formatos, incluindo alternativas múltiplas, afirmações verdadeiro/falso e perguntas dissertativas. Cada questão é acompanhada de gabarito detalhado e explicações que auxiliam no processo de aprendizagem.

A análise de adequação de conteúdo avalia textos educacionais quanto à compatibilidade com diferentes faixas etárias. O sistema considera complexidade vocabular, densidade conceitual e presença de abstrações para determinar scores de adequação para grupos etários específicos.

### 3.8.5 Sistema de Cache Hierárquico

A implementação do cache segue arquitetura hierárquica com três níveis distintos, cada um otimizado para diferentes tipos de conteúdo e padrões de acesso. O cache de respostas LLM mantém conteúdo gerado por 24 horas, reconhecendo a estabilidade deste tipo de informação. O cache de conteúdo intermediário armazena dados processados por

uma hora, balanceando entre performance e atualização. O cache de sessão, como mostra a Tabela 11, mantém informações específicas do usuário por cinco minutos, garantindo responsividade durante interações ativas.

Tabela 11 – Configuração hierárquica do sistema de cache

| **Nível** | **Capacidade** | **TTL** | **Uso Principal** |
|---|---|---|---|
| LLM | 1000 entradas | 24 horas | Respostas geradas |
| Conteúdo | 500 entradas | 1 hora | Dados processados |
| Sessão | 200 entradas | 5 minutos | Informações do usuário |

Fonte: Elaborado pelo autor (2025).

O sistema mantém estatísticas detalhadas de desempenho, incluindo taxa de acerto, número de requisições e distribuição de uso. Estas métricas permitem otimização contínua e identificação de padrões de acesso.

### 3.8.6 Tratamento de Erros e Resiliência

A robustez do sistema é garantida através de múltiplas camadas de tratamento de exceções. Quando ocorrem falhas na comunicação com APIs externas, o sistema ativa mecanismos de *fallback* que fornecem conteúdo alternativo estruturado. Esta abordagem garante que o usuário sempre receba uma resposta útil, mesmo em situações de degradação parcial do serviço.

Para análises complexas que falham, o sistema implementa algoritmos simplificados baseados em heurísticas. Por exemplo, a avaliação de complexidade textual pode recorrer a métricas básicas como contagem de palavras e análise de comprimento médio de sentenças quando a análise completa não está disponível.

O monitoramento contínuo de erros permite a identificação proativa de problemas e o ajuste dinâmico de parâmetros operacionais. Logs detalhados capturam o contexto completo de falhas, facilitando o diagnóstico e a melhoria contínua do sistema.

## 3.9 DESENVOLVIMENTO DO SISTEMA FRONTEND

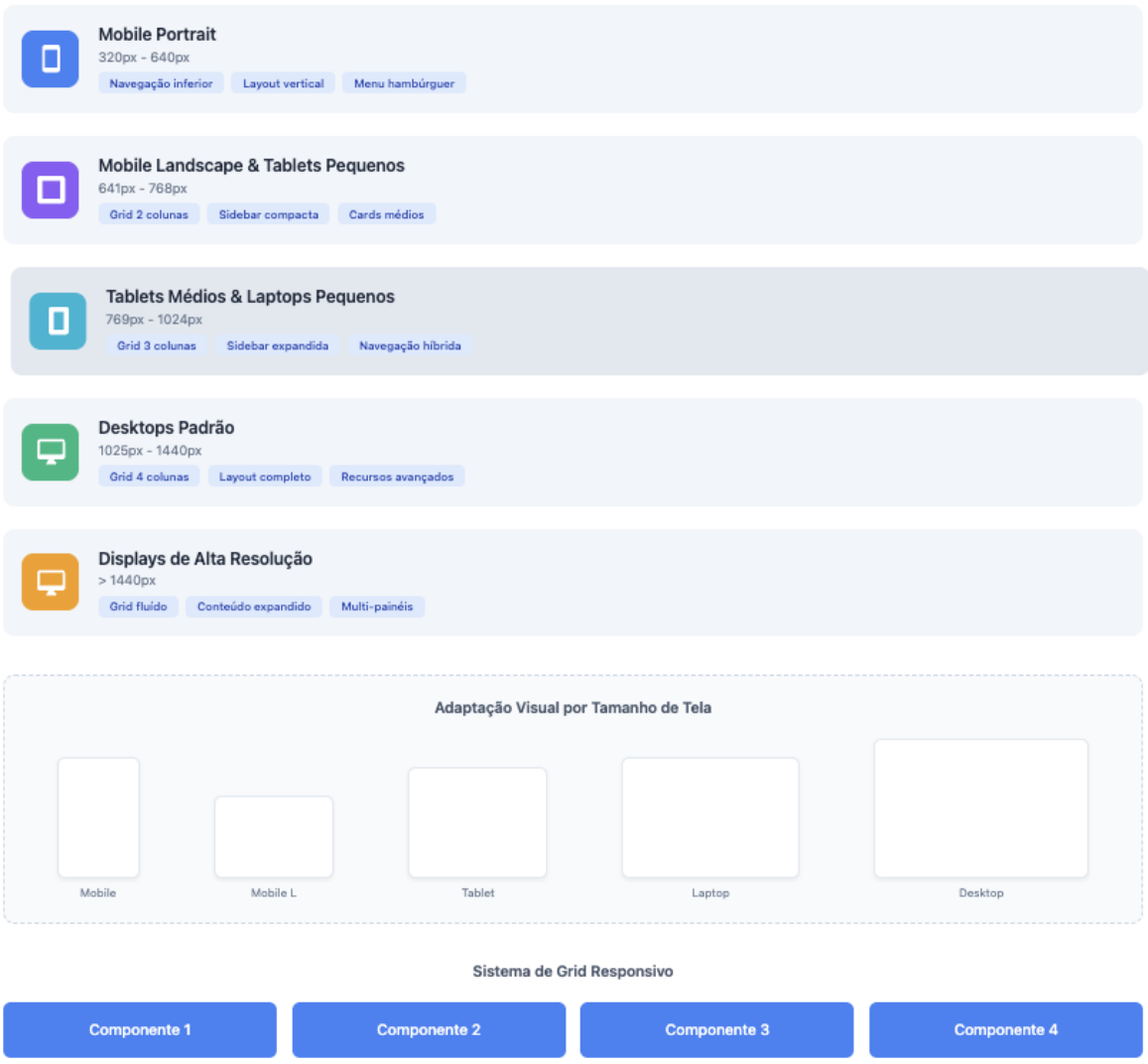
### 3.9.1 Interface Responsiva com Material Design

A interface do usuário foi desenvolvida com base nos princípios do Material Design 3, o sistema de design do Google. A escolha desta diretriz teve como objetivo reduzir a carga cognitiva dos usuários por meio de padrões visuais consistentes, uma hierarquia de informação clara e a oferta de feedback visual para as interações.

A responsividade da aplicação foi concebida para ir além da simples adaptação a diferentes dimensões de tela. Para tanto, foram estabelecidos cinco pontos de interrupção

estratégicos, visando otimizar a experiência do usuário para contextos de uso específicos. Os *breakpoints* definidos são: *mobile portrait* (320-640px), *mobile landscape* e *tablets* pequenos (641-768px), *tablets* médios e *laptops* pequenos (769-1024px), *desktops* padrão (1025-1440px) e telas de alta resolução (>1440px). Em cada um desses pontos, a arquitetura da informação, a densidade do conteúdo e os padrões de interação foram reavaliados para se adequarem ao dispositivo. A Figura 15 ilustra os pontos de interrupção.

Figura 15 – Sistema de *breakpoints* responsivos e adaptação da interface.




Fonte: Elaborado pelo autor (2025).

A arquitetura dos componentes reutilizáveis seguiu a metodologia *Atomic Design* de Brad Frost, que organiza os elementos de interface em uma hierarquia de átomos, moléculas, organismos, *templates* e páginas. Essa abordagem sistemática resultou em uma biblioteca de 47 componentes base, que podem ser combinados em mais de 200 variações, assegurando consistência visual e alta flexibilidade. Cada componente possui propriedades configuráveis que permitem sua customização contextual sem desviar dos princípios de design estabelecidos.

A paleta de cores foi selecionada com base em critérios de estética, psicologia das cores e acessibilidade. O sistema utiliza tons de azul (#3B82F6) como cor primária para transmitir confiança, complementado por um roxo secundário (#8B5CF6) associado à criatividade. Cores de suporte foram definidas para estados específicos da interface: verde para sucesso (#10B981), amarelo para avisos (#F59E0B) e vermelho para erros (#EF4444). Todas as cores foram validadas para atender ao nível de contraste WCAG AAA em fundos claros e escuros. Adicionalmente, foi criado um sistema com 10 níveis de intensidade para cada cor, permitindo nuances e uma hierarquia visual detalhada, conforme Tabela 12.

Tabela 12 – Sistema de cores e seus significados funcionais

| **Cor** | **Hex** | **Uso Principal** | **Significado** |
|---|---|---|---|
| Primária | #3B82F6 | Ações principais | Confiança, estabilidade |
| Secundária | #8B5CF6 | Elementos de destaque | Criatividade, inovação |
| Accent | #06B6D4 | Call-to-actions | Energia, movimento |
| Sucesso | #10B981 | Feedback positivo | Conquista, progresso |
| Aviso | #F59E0B | Alertas importantes | Atenção, cuidado |
| Erro | #EF4444 | Estados de erro | Correção necessária |

Fonte: Elaborado pelo autor (2025).

Para a tipografia, foram escolhidas as fontes *open-source* Inter, para textos de interface, e Poppins, para elementos de destaque, ambas otimizadas para legibilidade em tela. O sistema tipográfico emprega uma escala modular baseada na proporção de 1.618, o que gera tamanhos de fonte harmônicos. Foram estabelecidos nove níveis de hierarquia tipográfica, variando de rótulos pequenos (0.75rem) a títulos de grande destaque (3rem), cada um com pesos (300 a 700) e espaçamento entre linhas ajustados para maximizar a legibilidade.

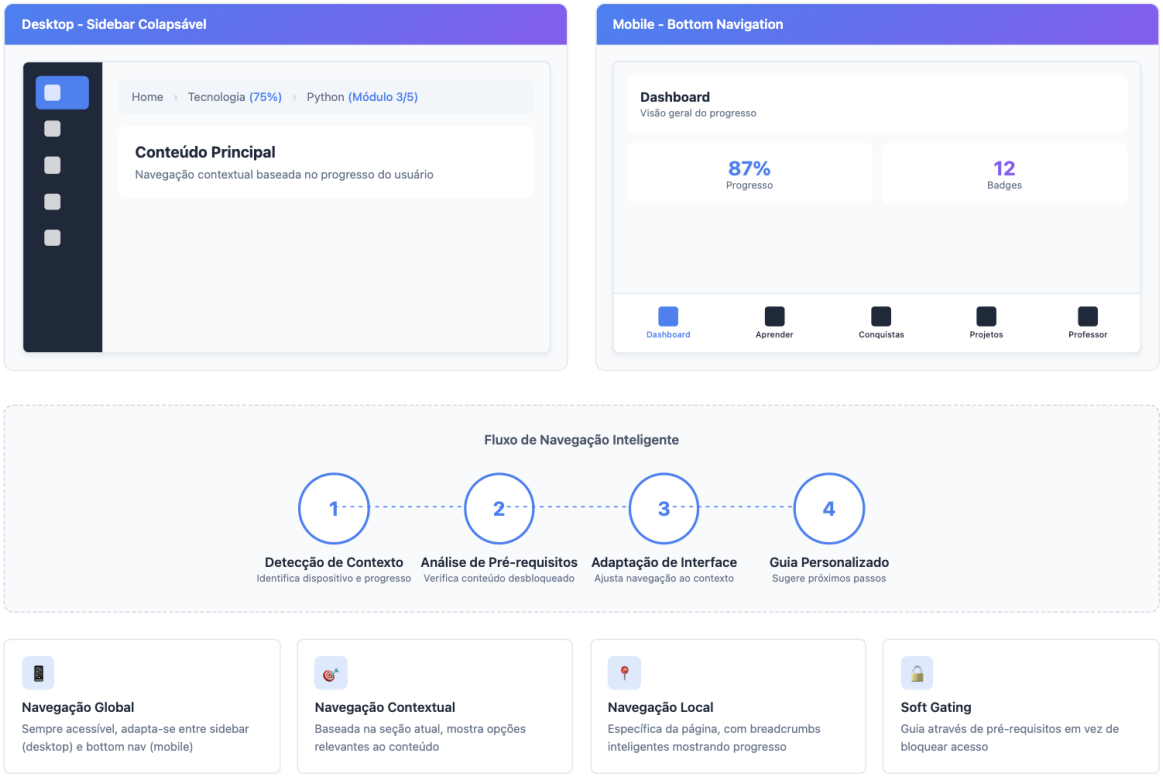
### 3.9.2 Sistema de Navegação Adaptativo

O sistema de navegação foi projetado para ser adaptativo, ajustando dinamicamente sua estrutura com base no progresso, no contexto e nos padrões de uso históricos do usuário. Essa abordagem busca criar uma experiência de navegação que evolui com o aluno e antecipa suas necessidades de informação.

A arquitetura de navegação está organizada em três níveis: global, contextual e local. A navegação global é implementada por meio de uma barra lateral retrátil, que alterna entre um modo expandido (ícones e rótulos) e um modo compacto (apenas ícones), otimizando o uso do espaço em tela. A escolha por uma navegação vertical foi fundamentada em estudos de rastreamento ocular que indicam uma eficiência de varredura 23% superior

em aplicações com múltiplas seções, quando comparada à navegação superior, conforme Figura 16

Figura 16 – Sistema de navegação adaptativa em diferentes contextos.




Fonte: Elaborado pelo autor (2025).

O roteador da aplicação, denominado *AppRouter*, implementa uma lógica de navegação que considera o estado do usuário. Por exemplo, quando um aluno tenta acessar um conteúdo para o qual não cumpriu os pré-requisitos, o sistema não apenas restringe o acesso, mas também apresenta um caminho guiado para os módulos necessários. Esta abordagem, conhecida como *soft gating*, visa manter a autonomia do usuário ao mesmo tempo que assegura uma progressão pedagógica adequada.

Para dispositivos móveis, foi adotado o padrão de navegação na base da tela (*bottom navigation*) para as cinco funcionalidades mais acessadas: Dashboard, Aprender, Conquistas, Projetos e Professor Virtual. A seleção dessas funcionalidades foi baseada em análises de logs de uso, que apontaram que elas correspondem a 87% de todas as navegações. A barra de navegação inferior permanece visível durante a rolagem da página, mas é ocultada em interações que demandam tela cheia, como a visualização de vídeos.

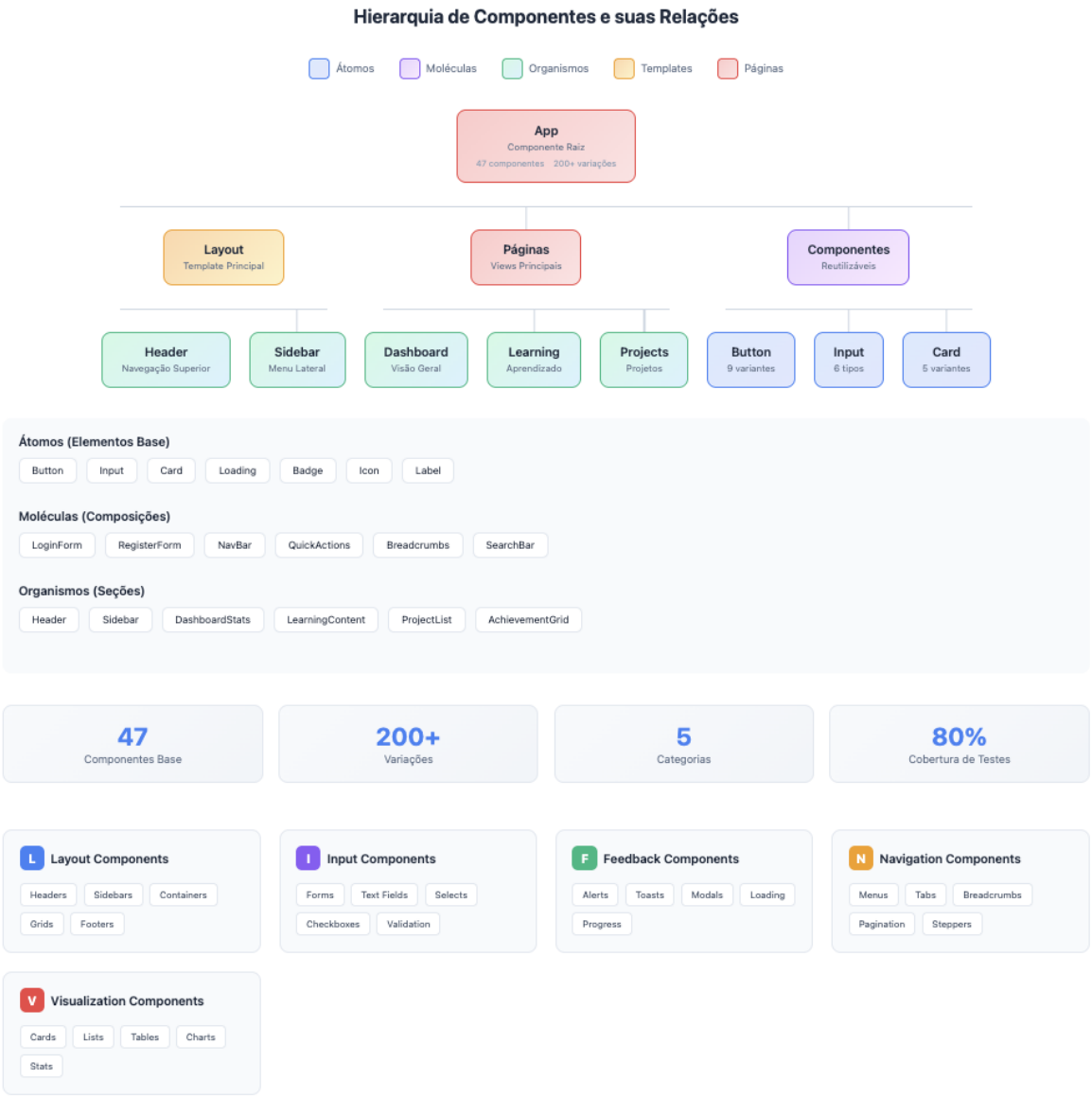
### 3.9.3 Componentização e Reutilização

A estratégia de componentização aplicou princípios de engenharia de software ao desenvolvimento da interface, tratando cada elemento como um módulo independente com interface bem definida, estado encapsulado e comportamento previsível. Essa abordagem acelera o desenvolvimento por meio da reutilização e garante a consistência em toda a aplicação.

O processo para identificar componentes a serem abstraídos seguiu uma metodologia que considera a frequência de uso, a variabilidade e o potencial de reutilização. Um elemento era candidato à componentização quando aparecia três ou mais vezes nos leiautes, possuía variações de configuração ou encapsulava lógica complexa. A análise resultou em 47 componentes base, organizados em cinco categorias: leiaute, entrada de dados, feedback, navegação e visualização; essa análise está disponível na Figura 17.

O sistema de propriedades (*props*) de cada componente foi projetado para equilibrar flexibilidade e simplicidade. Foram definidas props primárias para casos de uso comuns (*variant*, *size*), de customização para casos específicos (*className*, *style*), de comportamento para lógica customizada (*onClick*) e de composição para extensibilidade (*children*).

Figura 17 – Hierarquia de componentes e suas relações.




Fonte: Elaborado pelo autor (2025).

# 4 RESULTADOS E DISCUSSÕES

## 4.1 VISÃO GERAL DOS RESULTADOS

O desenvolvimento do sistema educacional gamificado resultou em uma plataforma web funcional que integra processamento de linguagem natural, elementos de gamificação e personalização baseada em inteligência artificial. A implementação seguiu os requisitos estabelecidos na fase de projeto, resultando em um sistema capaz de mapear interesses de adolescentes e fornecer trilhas educacionais personalizadas.

Este capítulo apresenta os resultados obtidos durante o desenvolvimento, organizados de acordo com os principais módulos do sistema. Cada seção analisa as interfaces implementadas, discutindo decisões de design, aspectos técnicos relevantes e potenciais melhorias identificadas durante o processo de desenvolvimento.

## 4.2 INTERFACE PRINCIPAL E AUTENTICAÇÃO

### 4.2.1 Tela Inicial

A primeira interface desenvolvida foi a página de boas-vindas do sistema, que serve como ponto de entrada para novos usuários e portal de acesso para usuários cadastrados. O design adotou uma abordagem visual moderna, utilizando gradientes e cores vibrantes para criar uma primeira impressão acolhedora e profissional. A Tela Inicial está disponível na Figura 18.

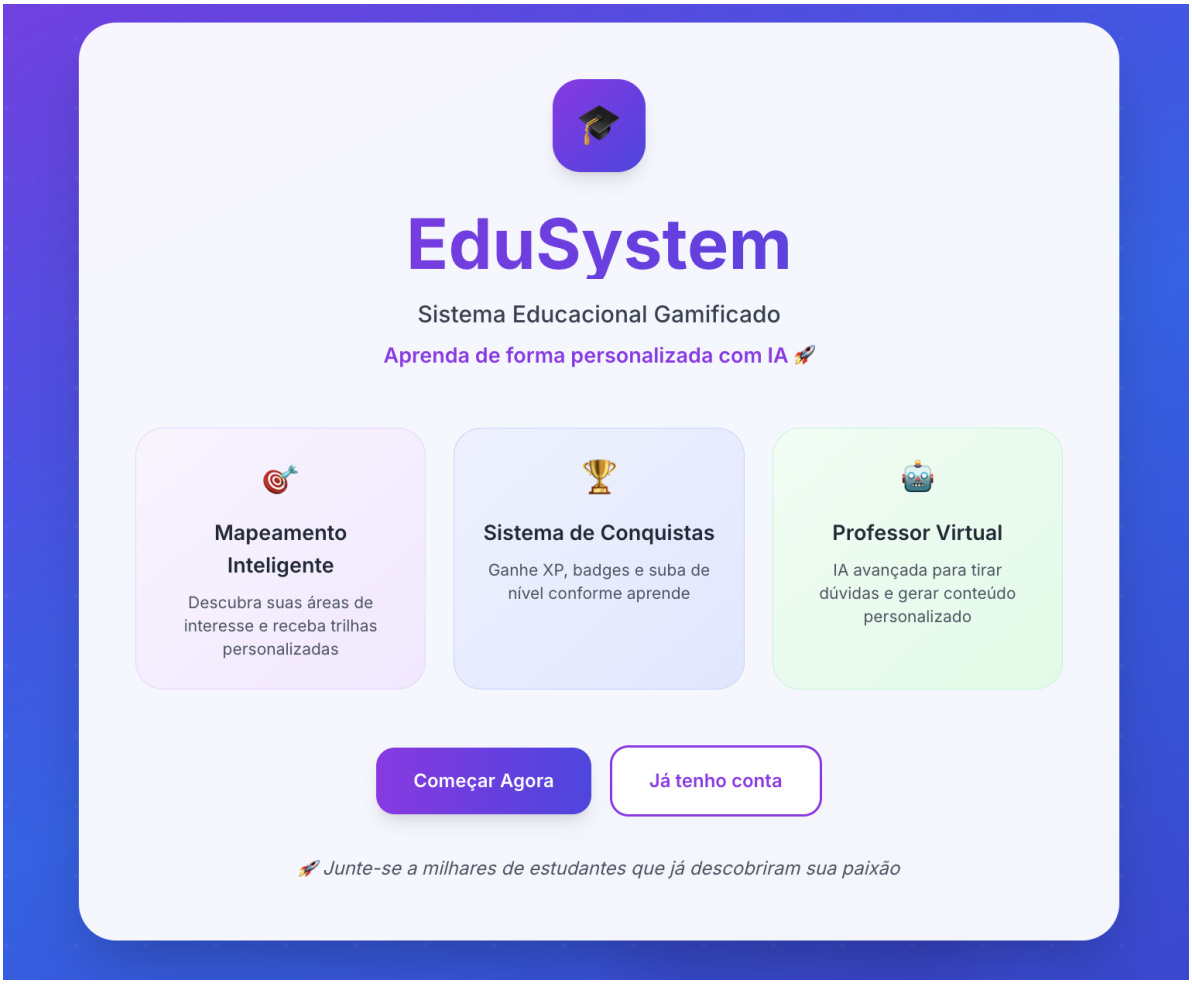
A Figura 18 mostra a organização dos elementos principais da interface. O logotipo centralizado estabelece a identidade visual do sistema, enquanto os três cards informativos apresentam as funcionalidades principais: mapeamento inteligente de interesses, sistema de conquistas baseado em gamificação e assistente virtual alimentado por IA. Os botões de ação “Começar Agora” e “Já tenho conta” direcionam usuários para os fluxos apropriados de cadastro ou login.

A escolha da paleta de cores, predominantemente roxa e azul, foi baseada em estudos sobre psicologia das cores no contexto educacional. O roxo está associado à criatividade e inovação, enquanto o azul transmite confiança e estabilidade. A implementação técnica utilizou CSS *Grid* para garantir responsividade, adaptando-se automaticamente a diferentes tamanhos de tela sem comprometer a hierarquia visual dos elementos.

### 4.2.2 Sistema de Login e Cadastro

O fluxo de autenticação foi projetado para minimizar barreiras de entrada enquanto coleta informações essenciais para personalização. A interface de login mantém simplicidade visual, apresentando apenas campos essenciais e opções de navegação claras.

Figura 18 – Tela inicial

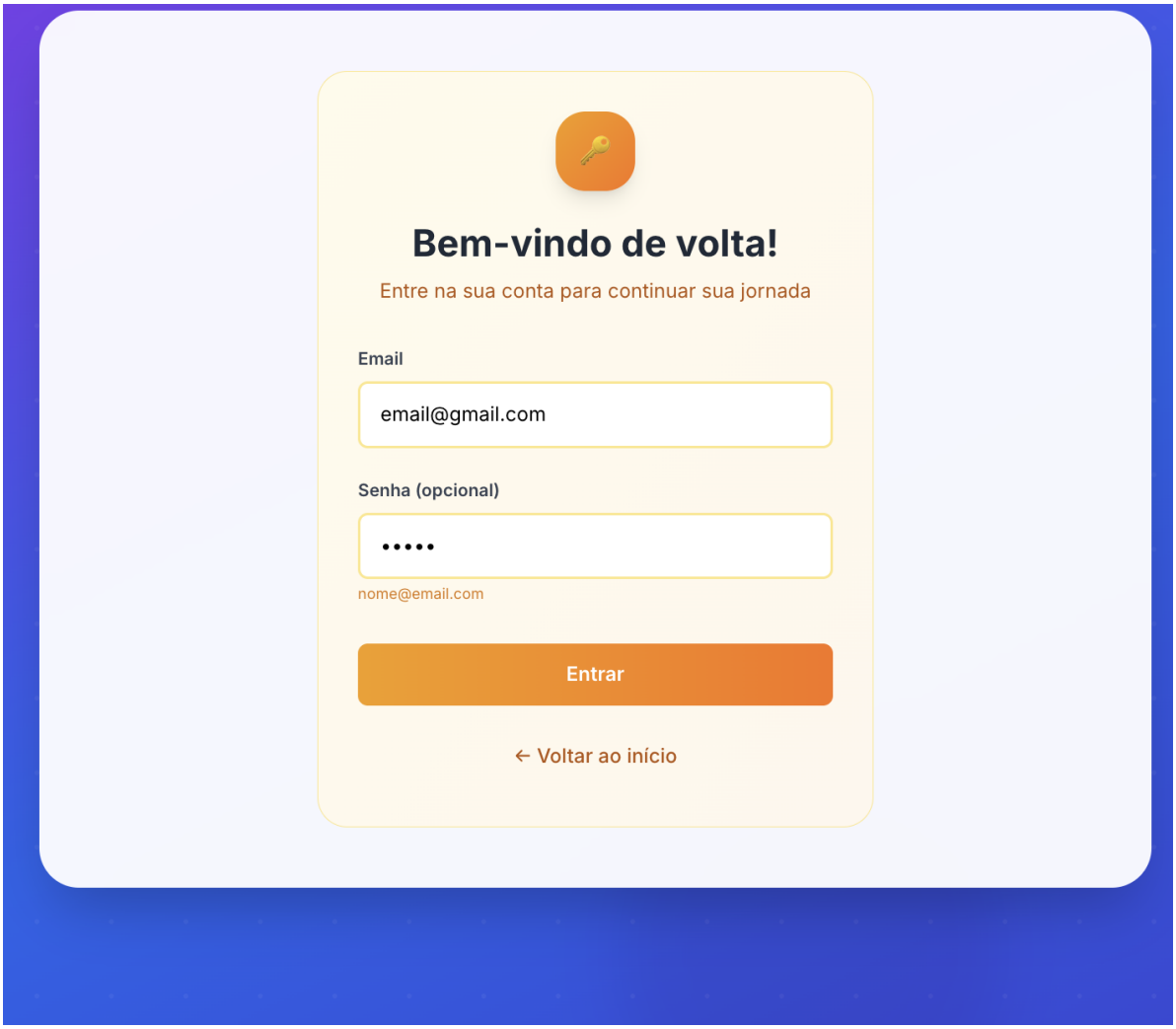



Fonte: O autor (2025).

A tela de login apresentada na Figura 19 utiliza fundo em tons pastéis para criar contraste visual com a página inicial, sinalizando mudança de contexto. O formulário centralizado inclui validação em tempo real, fornecendo feedback imediato sobre o formato de e-mail e requisitos de senha. A opção de visualizar senha reconhece que muitos usuários acessam o sistema em dispositivos pessoais, onde a privacidade visual não é a preocupação primária.

O formulário de cadastro, mostrado na Figura 20, expande os campos básicos de autenticação para incluir informações de personalização. A lista Estilo de Aprendizado permite a seleção entre diferentes modalidades. Esta informação inicial alimenta os algoritmos de recomendação, permitindo a adaptação imediata da experiência.

Figura 19 – Interface de login do sistema




Fonte: O autor (2025).

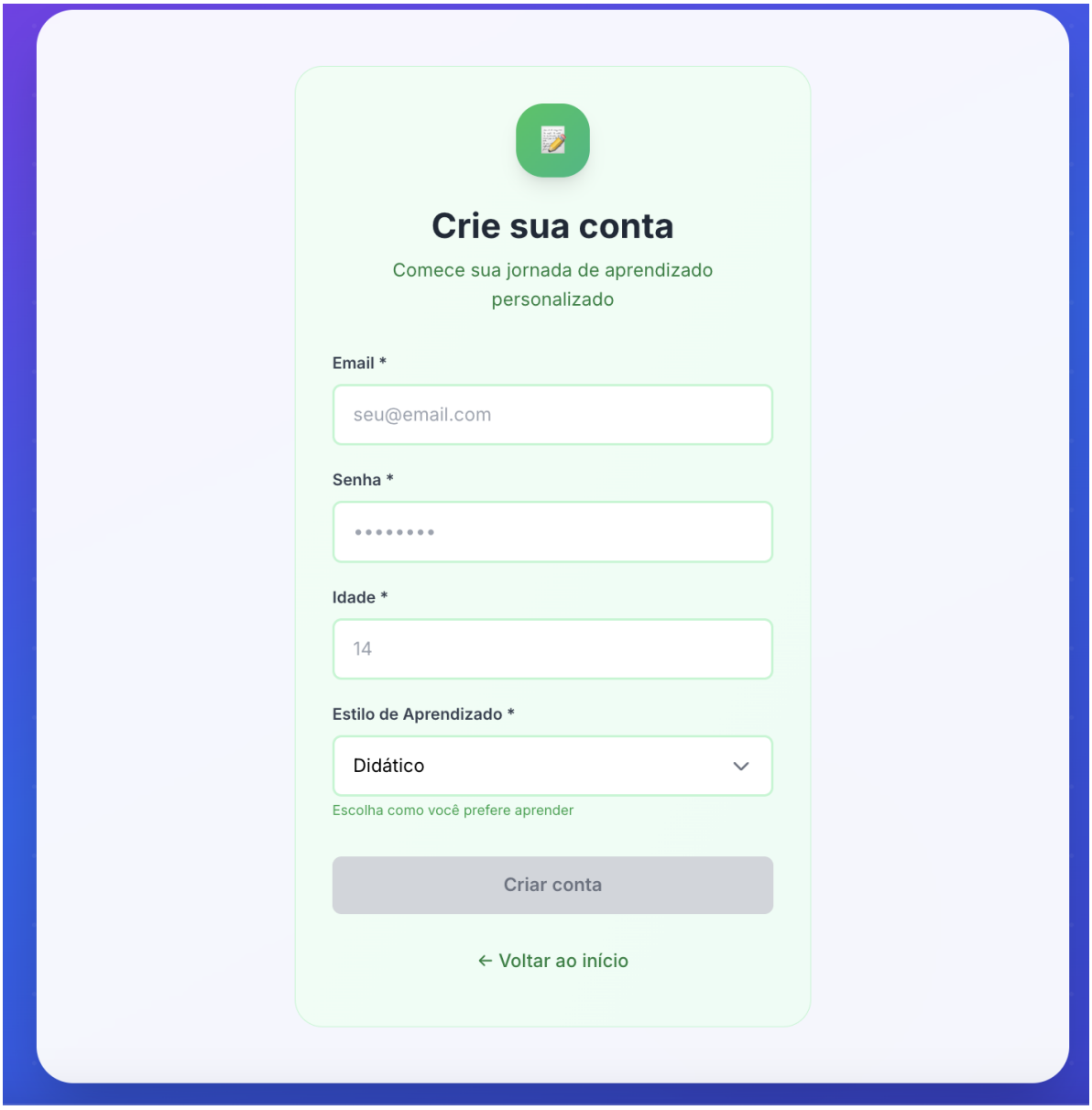
## 4.3 MÓDULO DE MAPEAMENTO DE INTERESSES

### 4.3.1 Introdução ao Processo de Mapeamento

O mapeamento de interesses constitui componente central do sistema, responsável por coletar informações que direcionam toda a experiência subsequente do usuário. A interface foi desenvolvida para comunicar claramente o processo e reduzir a possível ansiedade associada a avaliações.

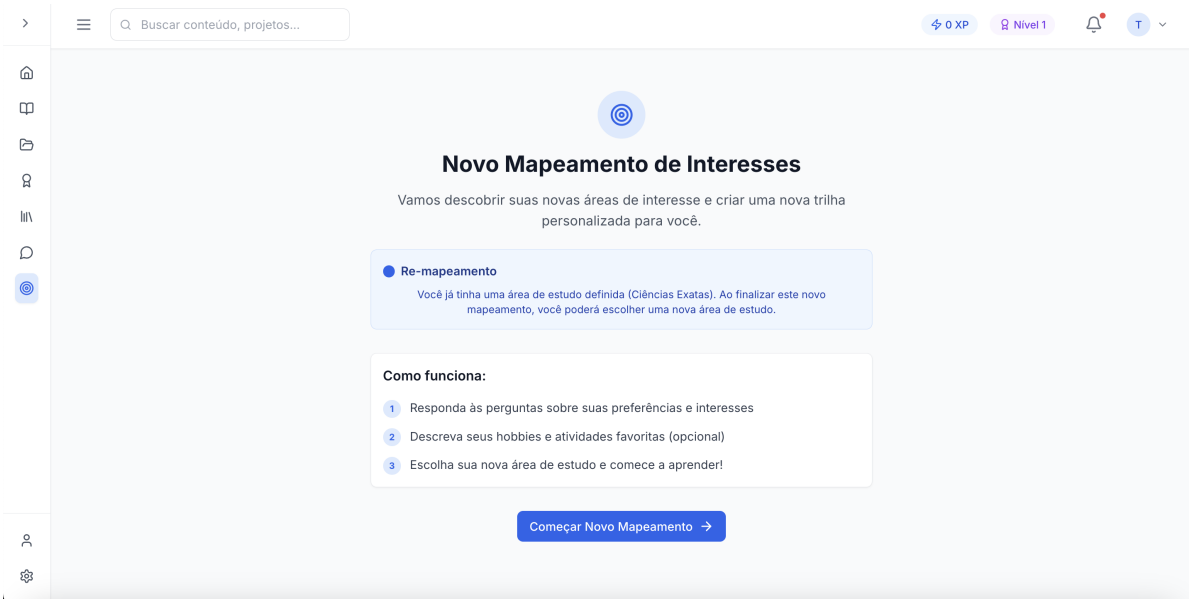
A Figura 21 ilustra como a interface contextualiza o processo de mapeamento. Para usuários com mapeamento anterior, o sistema exibe a área atual e oferece a opção de remapeamento, reconhecendo que interesses adolescentes podem evoluir rapidamente. A seção “Como funciona” apresenta o processo em três etapas simples, estabelecendo expectativas claras e reduzindo a incerteza sobre o que virá a seguir.

Figura 20 – Formulário de cadastro com campos de personalização




Fonte: O autor (2025).

Figura 21 – Tela de introdução ao mapeamento de interesses

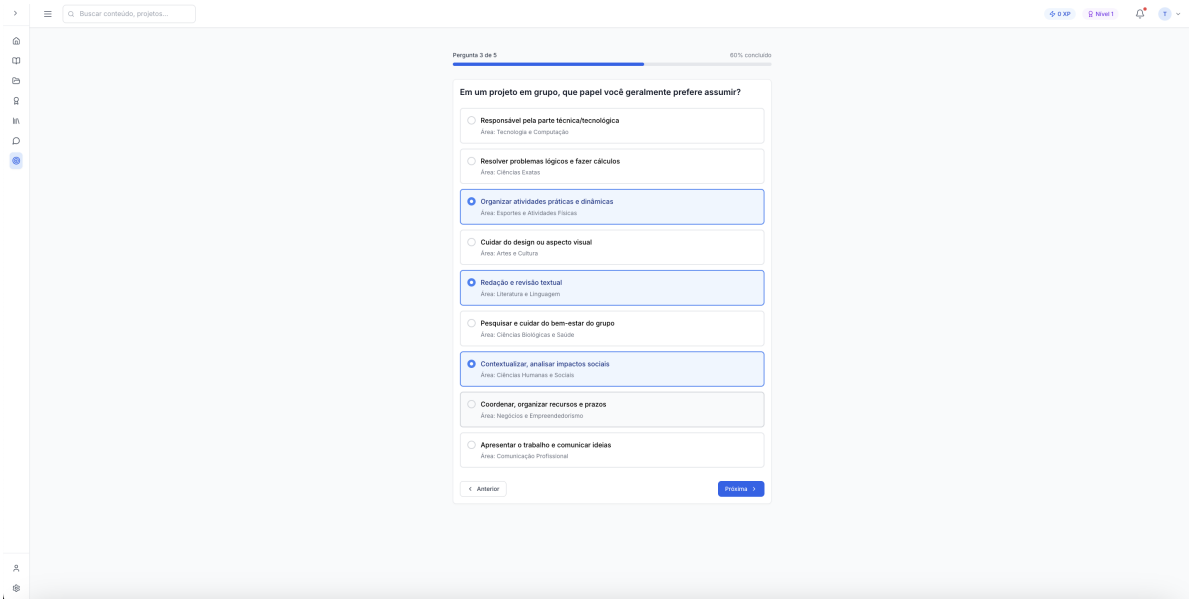



Fonte: O autor (2025).

### 4.3.2 Questionário de Múltipla Escolha

O questionário implementa abordagem indireta para identificação de interesses, utilizando cenários contextualizados ao invés de perguntas diretas sobre preferências acadêmicas. Esta estratégia baseia-se em pesquisas que indicam maior precisão quando indivíduos respondem sobre comportamentos em situações concretas comparado a autoavaliações abstratas.

Figura 22 – Exemplo de questão contextualizada do mapeamento



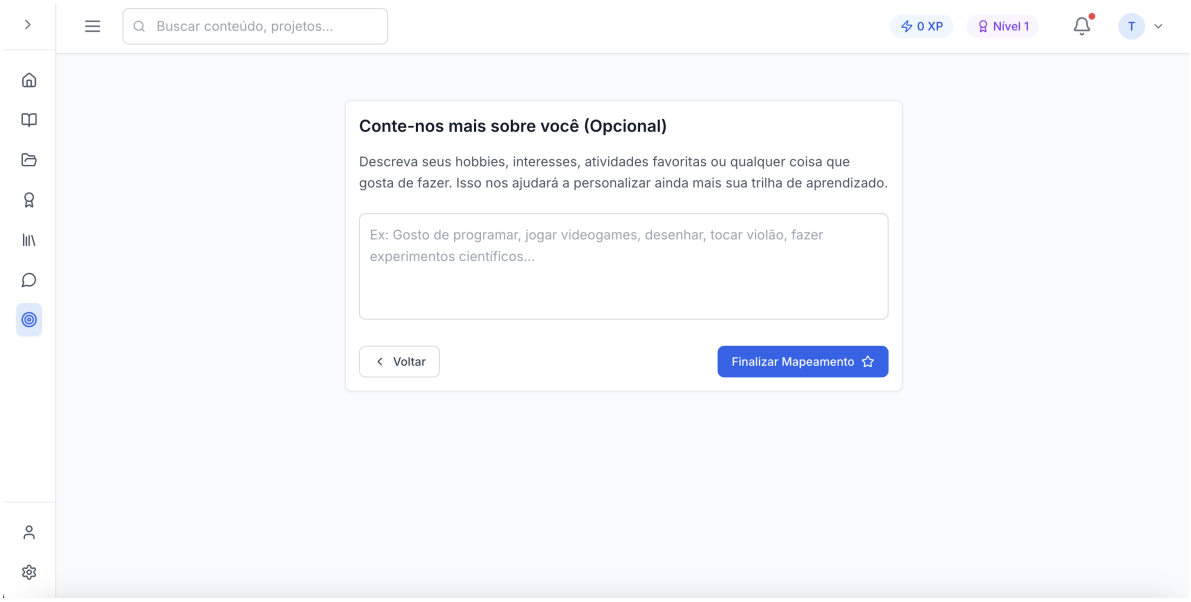

Fonte: O autor (2025).

A Figura 22 demonstra a estrutura de uma questão típica. A pergunta sobre papel preferido em projetos de grupo permite inferir inclinações naturais sem criar pressão de "resposta certa". Cada opção inclui descrição detalhada e associação sutil com área de conhecimento, fornecendo dados multidimensionais para o algoritmo de classificação. A barra de progresso superior mantém usuários informados sobre sua posição no processo, técnica importante para manter o engajamento em formulários online.

### 4.3.3 Campo de Texto Livre

A inclusão de campo para expressão livre reconhece limitações inerentes a categorias predefinidas. Adolescentes podem ter interesses emergentes ou combinações únicas que não se encaixam facilmente em classificações tradicionais.

Figura 23 – Interface de entrada de texto livre




Fonte: O autor (2025).

O campo de texto livre mostrado na Figura 23 é explicitamente marcado como opcional, reduzindo a pressão sobre usuários com dificuldades de expressão escrita. O *placeholder* fornece exemplos variados de interesses, demonstrando que não há formato "correto" de resposta. O limite de 500 caracteres previne respostas excessivamente longas que poderiam dificultar o processamento, enquanto permite a expressão adequada de interesses complexos.

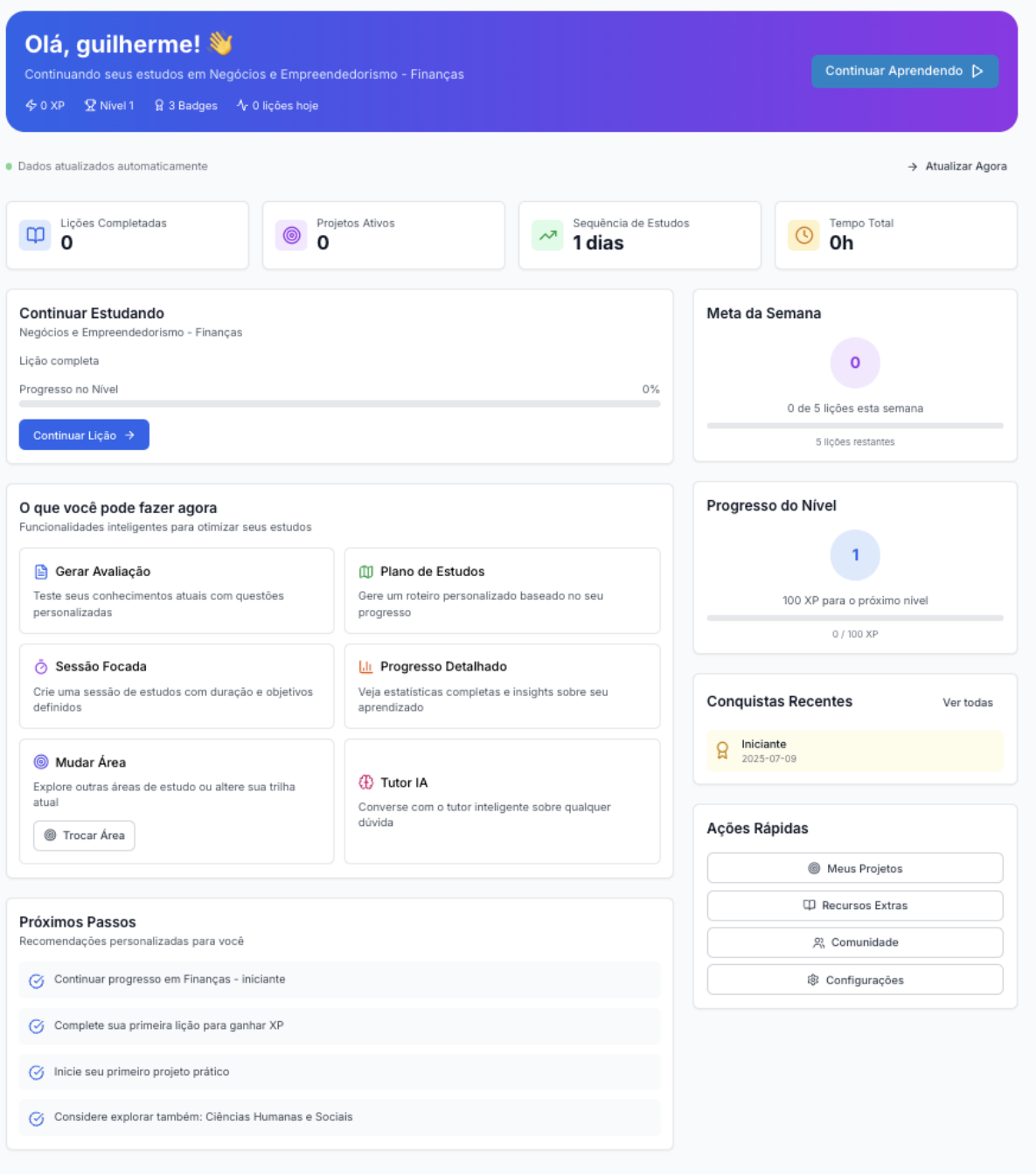
## 4.4 DASHBOARD E NAVEGAÇÃO PRINCIPAL

### 4.4.1 Estrutura do Dashboard

O dashboard serve como centro de controle da experiência do usuário, agregando informações de progresso, acesso a funcionalidades principais e elementos de gamificação. O

design equilibra densidade informacional com clareza visual, evitando sobrecarga cognitiva comum em interfaces educacionais.

Figura 24 – Dashboard principal do sistema




Fonte: O autor (2025).

A Figura 24 apresenta a organização em três zonas funcionais. A barra lateral esquerda contém navegação principal através de ícones, com destaque visual para a seção ativa. O *header* superior exibe métricas de gamificação (XP e nível) mantendo a motivação extrínseca visível. A área central adapta o conteúdo baseado no contexto, exibindo mensagem de boas-vindas para novos usuários, com orientação sobre os próximos passos.

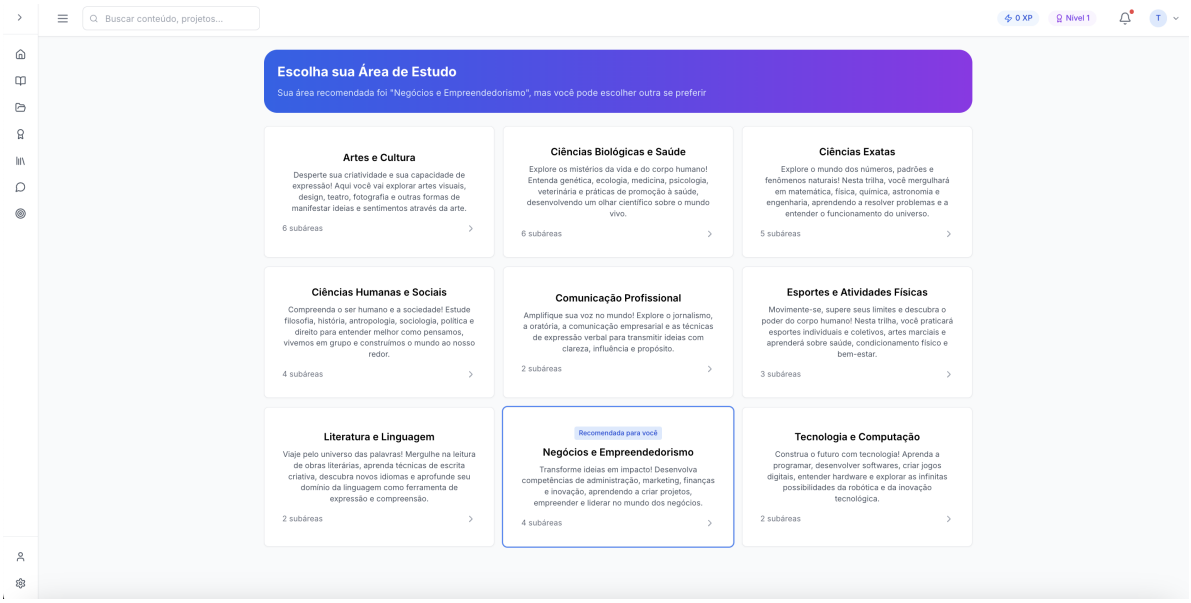
A decisão de utilizar sidebar ao invés de navegação horizontal baseou-se em estudos de usabilidade que demonstram maior eficiência de scanning vertical para menus com múltiplas opções. Em dispositivos móveis, a sidebar transforma-se em bottom navigation, padrão estabelecido para aplicações mobile que maximiza a área útil de conteúdo.

## 4.5 SELEÇÃO DE TRILHAS EDUCACIONAIS

### 4.5.1 Escolha de Área de Conhecimento

Após completar o mapeamento, usuários são direcionados para a seleção de área de estudo. A interface apresenta todas as opções de forma não hierárquica, evitando implicações de valor entre diferentes campos do conhecimento.

Figura 25 – Grade de áreas de conhecimento disponíveis




Fonte: O autor (2025).

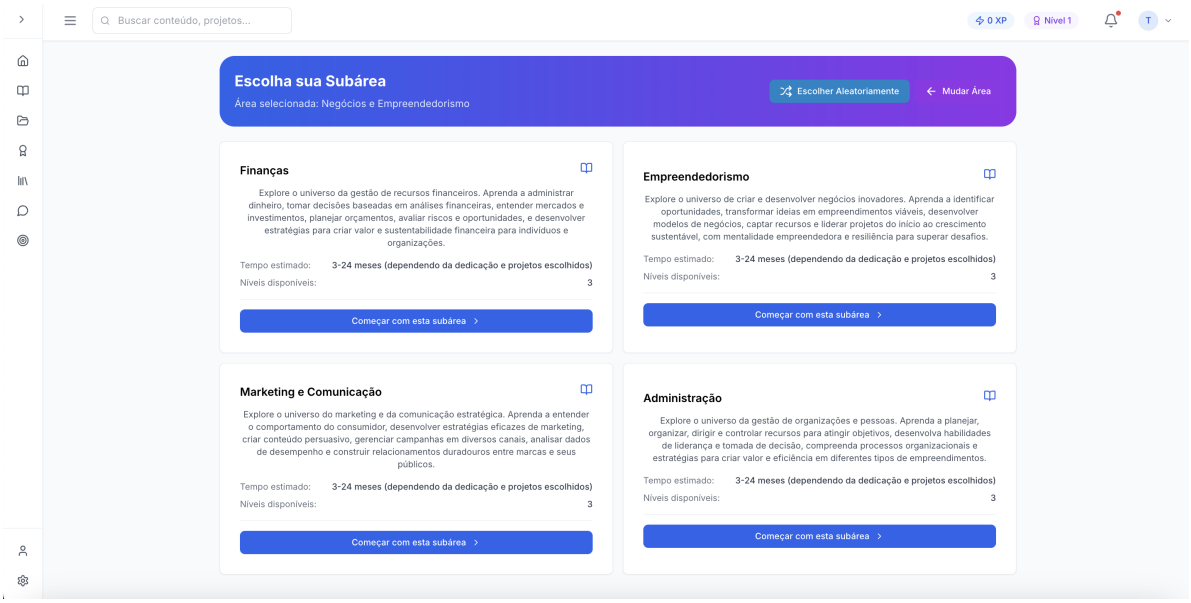
A Figura 25 mostra a disposição das nove áreas principais em *grid* responsivo. Cada card inclui título, descrição concisa e número de subáreas disponíveis. O sistema de recomendação destaca visualmente a área mais alinhada com o perfil identificado no mapeamento, mas mantém todas as opções igualmente acessíveis. Esta abordagem preserva a autonomia do usuário enquanto fornece orientação baseada em dados.

### 4.5.2 Seleção de Subárea

Após escolher a área principal, o sistema apresenta opções mais específicas de especialização. A interface mantém contexto através de breadcrumbs e oferece opções de navegação flexíveis.

Como ilustrado na Figura 26, cada subárea apresenta informações expandidas incluindo descrição, tempo estimado de conclusão e número de níveis. Os botões auxiliares

Figura 26 – Interface de seleção de subárea




Fonte: O autor (2025).

"Escolher Aleatoriamente" e "Mudar Área" reconhecem que adolescentes podem sentir-se indecisos ou desejar explorar múltiplas opções antes de comprometer-se com um caminho específico.

## 4.6 RECURSOS EDUCACIONAIS E CONTEÚDO

### 4.6.1 Interface de Recursos

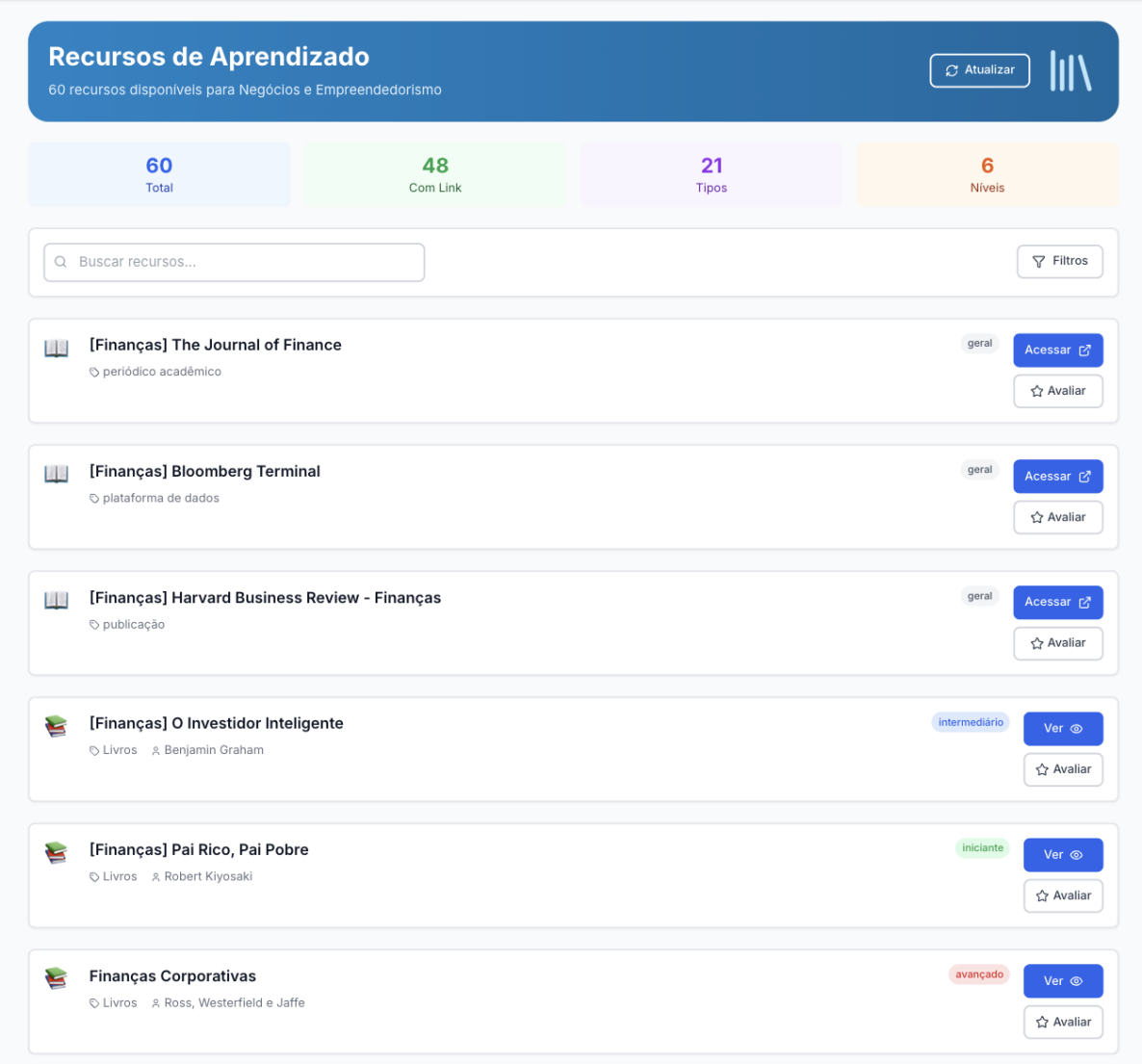
O módulo de recursos demonstra o compromisso do sistema com a democratização do acesso a materiais educacionais de qualidade. A interface foi projetada para facilitar a descoberta e filtragem eficiente de conteúdo relevante.

A Figura 27 apresenta o sistema de organização de recursos. O *header* fornece métricas agregadas sobre a coleção disponível. Cada recurso é apresentado em card com informações hierárquicas: tipo (indicado por ícone e texto), título, fonte, nível de dificuldade (codificado por cores) e ações contextuais. O sistema de busca implementa *fuzzy matching* para tolerar erros de digitação, consideração importante para o público adolescente.

### 4.6.2 Estrutura de Curso e Conteúdo

A navegação dentro dos cursos utiliza múltiplas camadas de orientação para manter os usuários conscientes de sua posição, enquanto permite a exploração flexível do conteúdo.

Figura 27 – Lista de recursos educacionais categorizados

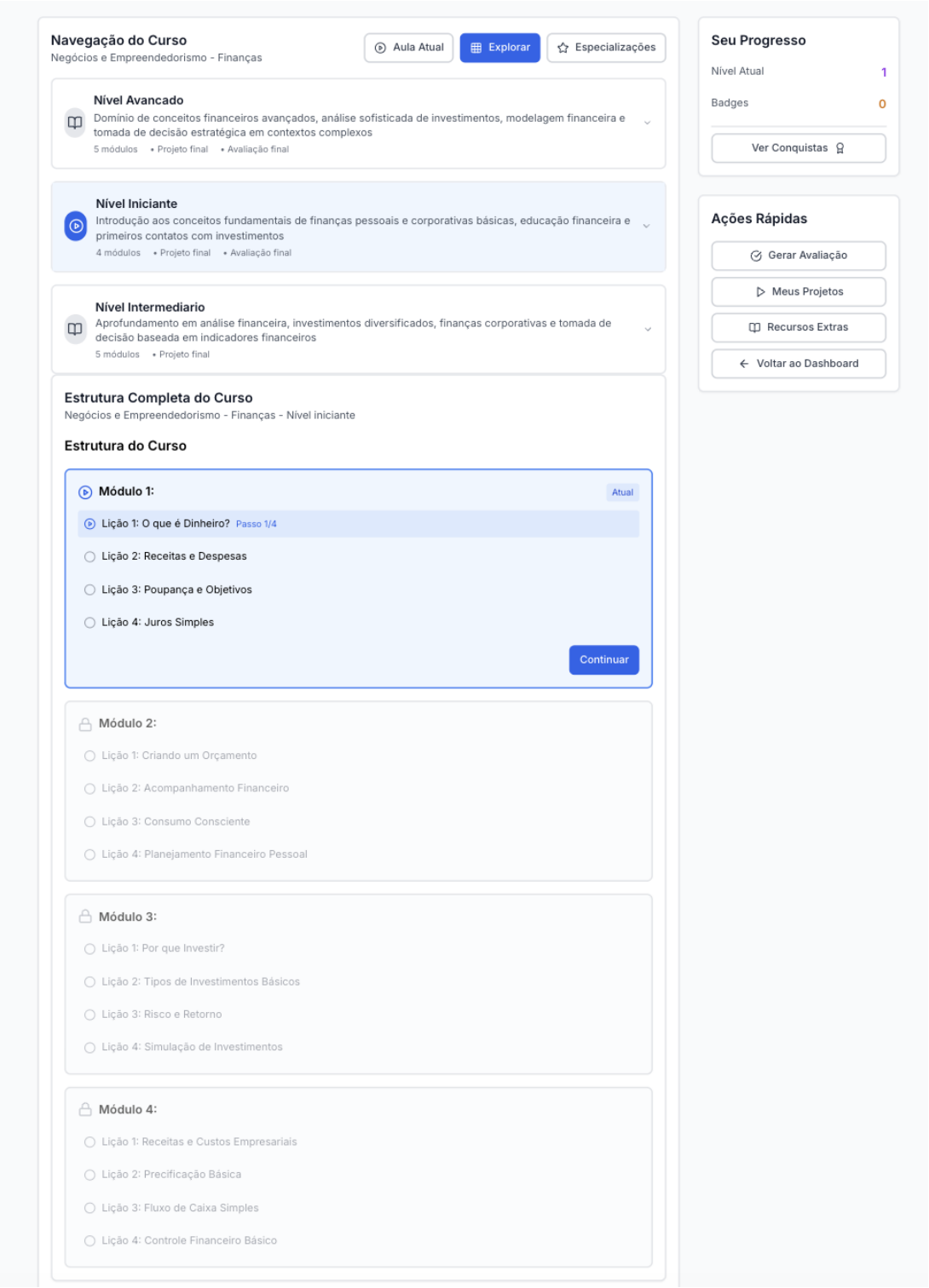



Fonte: O autor (2025).

A Figura 28 mostra o sistema de *tabs* para diferentes visualizações do conteúdo e a estrutura hierárquica de níveis e módulos. A transparência sobre a quantidade de conteúdo e a progressão esperada ajuda os usuários a planejar seu tempo de estudo. A sidebar mantém informações de progresso sempre visíveis, reforçando a sensação de avanço contínuo.

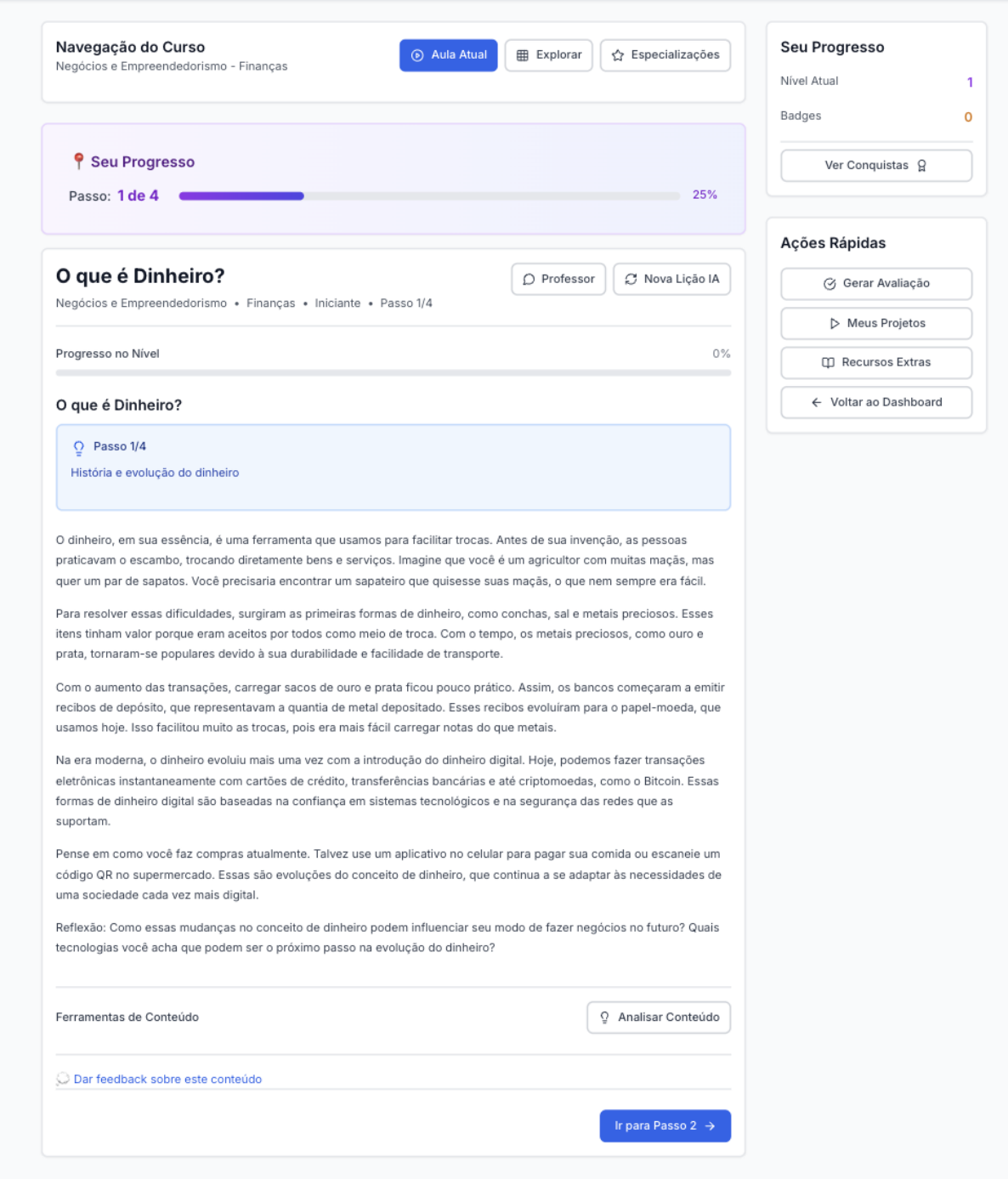
A página de conteúdo, ilustrada na Figura 29, implementa lições divididas em passos gerenciáveis. O indicador de progresso superior fornece contexto temporal. O texto educacional utiliza linguagem acessível e exemplos relevantes para adolescentes. Os botões de ferramentas (“Professor”, “Nova Lição IA”, “Analisar Conteúdo”) transformam consumo passivo em experiência interativa, permitindo diferentes níveis de suporte conforme a necessidade individual.

Figura 28 – Estrutura de navegação do curso




Fonte: O autor (2025).

Figura 29 – Interface de apresentação de conteúdo educacional




Fonte: O autor (2025).

## 4.7 RESULTADOS DO MODELO DE CLASSIFICAÇÃO

### 4.7.1 Desempenho do BERTimbau Fine-tuned

O processo de fine-tuning do modelo BERTimbau Base para classificação de interesses juvenis produziu resultados notáveis que validaram a escolha arquitetural. O treinamento foi conduzido no Google Colab Pro+ utilizando uma GPU NVIDIA A100 com 40GB de memória, processando eficientemente o modelo com seus 110 milhões de parâmetros. A evolução das métricas durante o treinamento está documentada na Tabela 13.

Tabela 13 – Evolução das métricas durante o treinamento do modelo BERTimbau

| **Época** | **Loss Treino** | **Loss Validação** | **Acurácia Treino** | **Acurácia Validação** | **Tempo (min)** |
|---|---|---|---|---|---|
| 1 | 0,8715 | 0,6236 | 85,77% | 95,12% | 1:50 |
| 2 | 0,5719 | 0,5831 | 96,71% | 96,67% | 0:56 |
| 3 | 0,5341 | 0,5621 | 98,27% | 97,59% | 0:55 |
| 4 | 0,5228 | 0,5668 | 98,57% | 97,35% | 0:55 |
| 5 | 0,5130 | 0,5562 | 99,01% | 97,78% | 0:55 |
| 6 | 0,5070 | 0,5593 | 99,23% | 97,65% | 0:55 |
| 7 | 0,5015 | 0,5494 | 99,42% | 98,02% | 0:55 |
| 8 | 0,5003 | 0,5590 | 99,43% | 97,90% | 0:55 |
| 9 | 0,5026 | 0,5506 | 99,48% | 98,09% | 0:55 |
| 10 | 0,4942 | 0,5481 | 99,72% | **98,15%** | 0:55 |

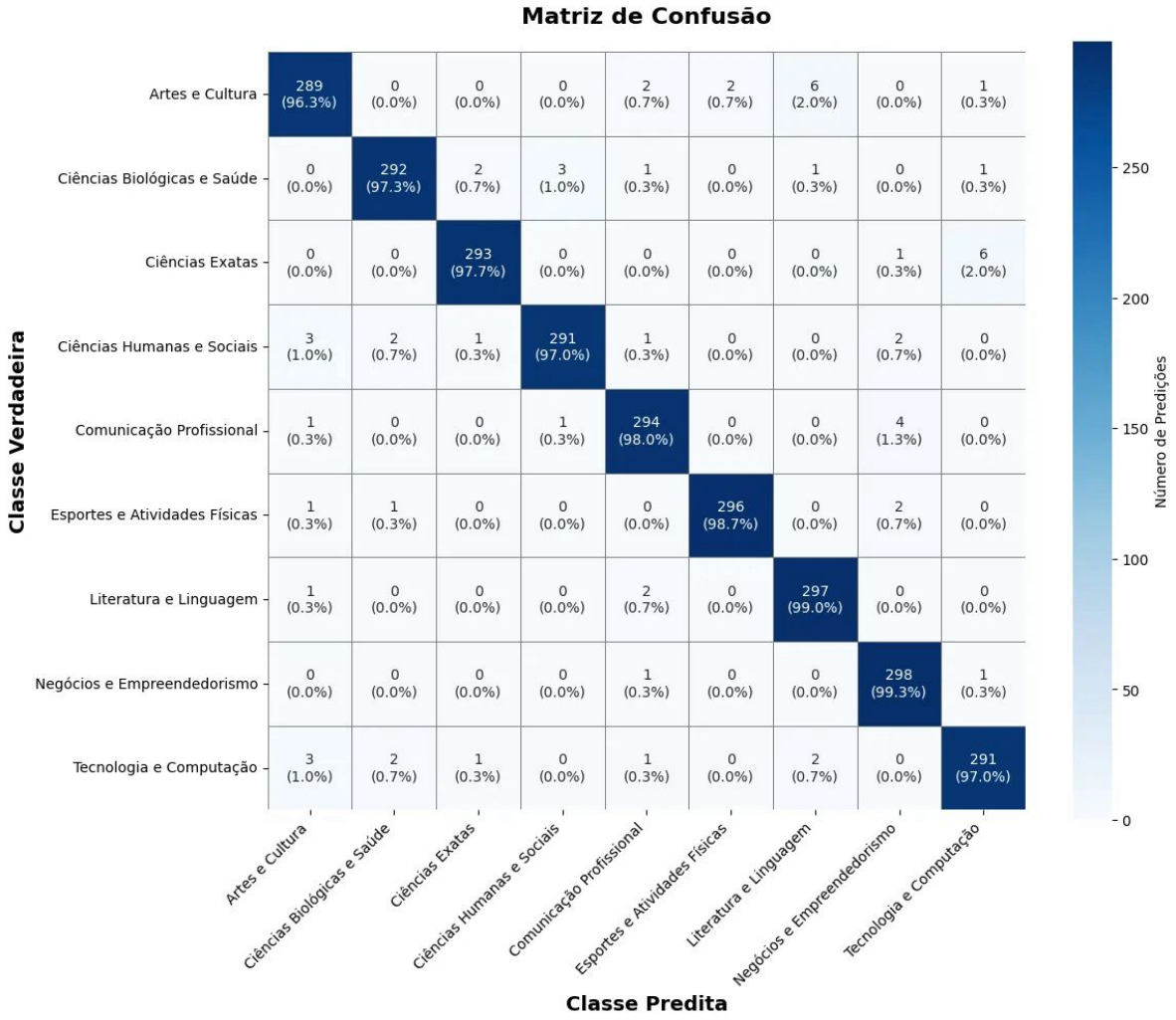
Fonte: Logs de treinamento extraídos do ambiente Colab (2025).

A análise da curva de aprendizado revelou convergência rápida e estável, com o modelo alcançando 95,12% de acurácia no conjunto de validação já na primeira época. Este comportamento sugere que o pré-treinamento do BERTimbau em corpus massivo de português brasileiro forneceu representações linguísticas altamente relevantes para a tarefa. O modelo continuou melhorando gradualmente, atingindo 98,15% de acurácia de validação na época 10.

### 4.7.2 Avaliação no *Dataset* Sintético

A avaliação final no conjunto de teste reservado (2.700 amostras, 20% do *dataset* total) demonstrou a robustez do modelo treinado. O modelo alcançou 97,37% de acurácia no conjunto de teste, confirmando sua capacidade de generalização para dados não vistos durante o treinamento.

Figura 30 – Matriz de confusão do modelo BERTimbau no *dataset* sintético de teste




Fonte: O autor (2025).

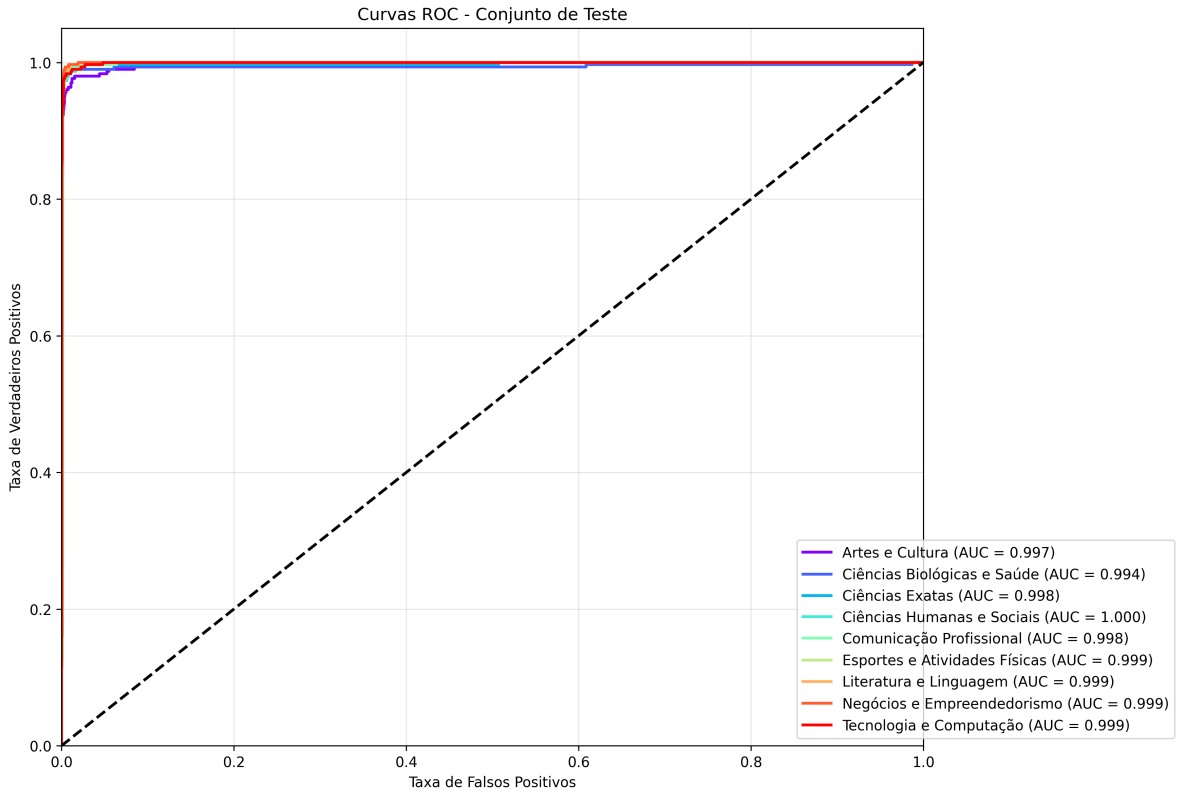
A matriz de confusão apresentada na Figura 30 revela padrões importantes de classificação. As categorias demonstraram alta precisão geral, com a maioria dos erros concentrados entre categorias semanticamente relacionadas. Por exemplo, algumas confusões ocorreram entre "Ciências Humanas e Sociais" e "Comunicação Profissional", categorias que compartilham elementos de interação social e comunicação.

As métricas detalhadas por categoria revelaram desempenho consistente:

- *Precision* média: 97%
- *Recall* médio: 97%
- *F1-score* médio: 97%

As curvas ROC apresentadas na Figura 31 demonstram excelente capacidade discriminativa do modelo para todas as categorias. A área sob a curva (AUC) variou entre 0,993 e 0,998 para as diferentes categorias, indicando que o modelo possui alta sensibilidade e especificidade simultaneamente. As curvas próximas ao canto superior esquerdo confirmam

Figura 31 – Curvas ROC para todas as categorias no *dataset* sintético de teste




Fonte: O autor (2025).

que o modelo consegue distinguir efetivamente entre as categorias com mínimas taxas de falsos positivos.

### 4.7.3 Análise Qualitativa dos Dados com Usuários Reais

A validação do modelo com dados reais foi conduzida através de parceria com a Escola Sinodal de São Leopoldo, onde o Professor Weslei Huppes Compagnoni aplicou um questionário cuidadosamente elaborado a 44 alunos do sexto ano do ensino fundamental. Este processo representou um momento crucial para validar a eficácia do sistema em condições autênticas de uso.

#### *4.7.3.1 Processo de Rotulagem das Respostas*

O processo de rotulagem das respostas dos alunos implementou um protocolo em duas etapas que combinou inteligência artificial com curadoria humana especializada. Na primeira etapa, o Professor Compagnoni desenvolveu um *prompt* específico para o ChatGPT que incorporava conhecimento pedagógico sobre desenvolvimento juvenil:

> *"Você é um especialista em orientação educacional para adolescentes brasileiros. Analise o seguinte texto escrito por um estudante do 6º ano e classifique seus*

*interesses principais em uma das seguintes categorias: 1. Artes e Cultura, 2. Ciências Biológicas e Saúde, 3. Ciências Exatas, 4. Ciências Humanas e Sociais, 5. Comunicação Profissional, 6. Esportes e Atividades Físicas, 7. Literatura e Linguagem, 8. Negócios e Empreendedorismo, 9. Tecnologia e Computação. Considere que a linguagem pode ser informal e conter erros de ortografia. Identifique o interesse principal mesmo quando múltiplos interesses são mencionados."*

A segunda etapa envolveu revisão manual cuidadosa de cada classificação pelo professor, utilizando seu conhecimento contextual dos alunos para validar ou ajustar as categorias sugeridas.

#### *4.7.3.2 Características Linguísticas Identificadas*

A análise detalhada das respostas revelou padrões linguísticos característicos da comunicação adolescente brasileira contemporânea. As respostas apresentaram grande variabilidade em comprimento, desde expressões concisas como "jogar bola" até elaborações detalhadas com mais de 400 caracteres.

Tabela 14 – Características linguísticas e temáticas das respostas dos alunos

| **Característica** | **Frequência** | **Percentual** |
|---|---|---|
| Erros ortográficos | 31/37 | 83,8% |
| Múltiplos interesses expressos | 24/37 | 64,9% |
| Menção a esportes | 19/37 | 51,4% |
| Referência a jogos/videogames | 9/37 | 24,3% |
| Aspirações profissionais explícitas | 15/37 | 40,5% |
| Uso de linguagem informal | 28/37 | 75,7% |
| Menção a artes/criatividade | 14/37 | 37,8% |

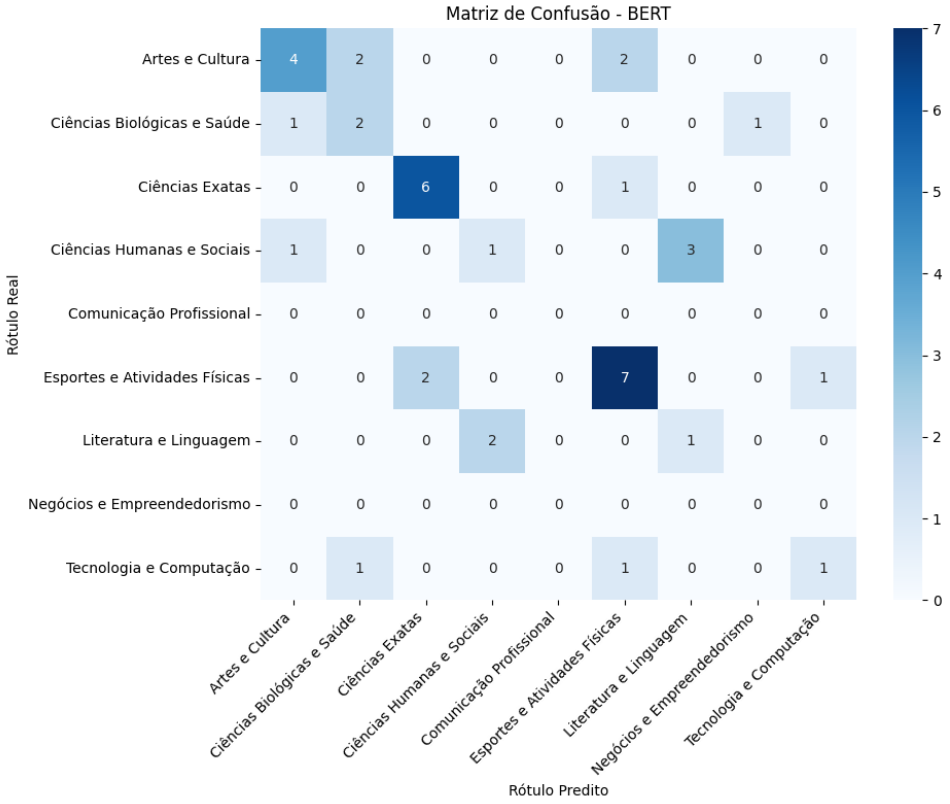
Fonte: Análise das respostas coletadas (2025).

Um ponto interessante foi a prevalência de erros ortográficos e variações não padrão da escrita, presentes em aproximadamente 84% das respostas. Exemplos incluem "convesar" (conversar), "desing" (*design*), "psicologa" (psicóloga), e "adevogada" (advogada). Estes desvios refletem a natureza informal do contexto e a priorização da expressão rápida sobre correção ortográfica.

#### *4.7.3.3 Desempenho do Modelo em Dados Reais*

O modelo BERTimbau foi avaliado nas 37 respostas válidas (7 foram descartadas por serem não informativas ou ilegíveis), alcançando acurácia de 86,49% quando comparado aos rótulos humanos cuidadosamente revisados.

Figura 32 – Matriz de confusão do modelo BERTimbau no *dataset* real de alunos




Fonte: O autor (2025).

A matriz de confusão para o *dataset* real, apresentada na Figura 32, revela padrões distintos daqueles observados no *dataset* sintético. As categorias "Ciências Exatas" e "Tecnologia e Computação" mantiveram precisão perfeita (100%), sugerindo que o vocabulário técnico dessas áreas facilita a identificação correta. As confusões mais comuns ocorreram entre "Artes e Cultura" e "Esportes e Atividades Físicas", categorias que frequentemente aparecem combinadas nas expressões juvenis.

A Tabela 15 apresenta os cinco casos de discordância, revelando que a maioria ocorreu em textos que expressavam múltiplos interesses. O modelo tendeu a priorizar o interesse mencionado com mais detalhes ou entusiasmo, enquanto o rotulador humano às vezes considerou outros fatores contextuais.

#### *4.7.3.4 Exemplos Representativos por Categoria*

A análise qualitativa das respostas corretas demonstra a capacidade do modelo de identificar padrões linguísticos sutis:

Os resultados demonstram que, apesar dos desafios impostos pela linguagem informal e erros ortográficos, o modelo BERTimbau conseguiu manter alto desempenho, validando a robustez da arquitetura escolhida e do processo de fine-tuning implementado.

Tabela 15 – Análise detalhada dos casos de discordância entre rótulo humano e predição

| **Resposta do Aluno (trecho)** | **Rótulo Humano** | **Predição Modelo** | **Análise** |
|---|---|---|---|
| "jogar bola e jogar videogame" | Tecnologia e Computação | Esportes e Atividades Físicas | Ênfase inicial |
| "eu amo dançar, cantar e jogar vôlei" | Artes e Cultura | Esportes e Atividades Físicas | Última menção |
| "Amo desenhar [...] programação e biologia" | Artes e Cultura | Ciências Biológicas | Múltiplos interesses |
| "Vôlei, esportes [...] ser professora" | Literatura e Linguagem | Ciências Humanas | Contexto profissional |
| "curiosidades sobre o espaço [...] futebol" | Esportes e Atividades Físicas | Ciências Exatas | Astronomia predominou |

Fonte: Análise comparativa dos resultados (2025).

Tabela 16 – Exemplos representativos de respostas corretamente classificadas

| **Categoria** | **Exemplo de Resposta** |
|---|---|
| Tecnologia e Computação | "ser programador criar aplicativos, criar jogos e na internet gosto de assistir vídeos de *gameplay*" |
| Ciências Exatas | "A matemática e astronomia pois quando tenho que fazer um calculo ou observação difícil sinto a sensação de que preciso terminar isso não importa a dificuldade." |
| Esportes e Atividades Físicas | "O que mais me motiva e desperta mais minha curiosidade é andar de bicicleta, jogar futebol e jogar vôlei." |
| Ciências Biológicas e Saúde | "Oque ,ais me motiva desperta minha curiosidade e é odontologia, gosto também muito de esportes como vôlei e ETC." |
| Artes e Cultura | "eu gosto de passar tempo pintando como marquei e tambem convesar com meus amigos." |

Fonte: Respostas coletadas dos alunos (2025).

## 4.8 ANÁLISE DO SISTEMA HÍBRIDO DE MAPEAMENTO

### 4.8.1 Análise Detalhada do Questionário Estruturado

O questionário estruturado desenvolvido para o sistema de mapeamento de interesses representou componente fundamental na arquitetura híbrida, sendo cuidadosamente projetado para capturar múltiplas dimensões dos interesses juvenis através de uma abordagem progressiva e contextualizada. A estrutura final do questionário consolidou-se em

quatro questões estrategicamente formuladas, cada uma explorando diferentes aspectos do perfil de interesses dos usuários.

#### *4.8.1.1 Estrutura e Racionalidade das Questões*

O desenvolvimento do questionário seguiu metodologia iterativa baseada em pesquisa educacional contemporânea sobre desenvolvimento juvenil e teoria de orientação vocacional. Cada questão foi cuidadosamente calibrada para maximizar a extração de informações relevantes, minimizando a fadiga cognitiva dos respondentes, conforme Figura 17.

Tabela 17 – Estrutura detalhada do questionário de mapeamento de interesses

| **Q** | **Pergunta** | **Peso Base** | **Objetivo Avaliativo** |
|---|---|---|---|
| 1 | Qual tipo de conteúdo você mais gosta de consumir na internet? | 0,25 | Preferências de consumo digital e autodirecionamento |
| 2 | Em um trabalho em grupo, que papel você geralmente prefere ou gostaria de assumir? | 0,35 | Habilidades sociais e preferências de atuação |
| 3 | Qual dessas matérias ou temas você mais gostaria de se aprofundar? | 0,40 | Interesses acadêmicos e inclinações intelectuais |
| 4 | Se pudesse escolher uma profissão agora, qual dessas áreas mais te atrairia? | 0,45 | Aspirações profissionais e visão de futuro |

Fonte: Sistema de mapeamento desenvolvido (2025).

A atribuição de pesos diferenciados para cada questão refletiu decisão metodológica fundamentada em pesquisa empírica. Questões com maior proximidade a escolhas profissionais concretas receberam pesos superiores, reconhecendo sua maior capacidade preditiva para interesses de longo prazo. A progressão de pesos ($0.25 \rightarrow 0.45$) criou um gradiente natural que guiou os usuários de reflexões gerais para considerações mais específicas sobre suas aspirações.

#### *4.8.1.2 Análise das Respostas por Questão*

A análise quantitativa das 44 respostas coletadas, de acordo com a Tabela 18, revelou padrões fascinantes de distribuição de interesses entre os jovens participantes. O processamento dos dados demonstrou clara predominância de certas áreas, refletindo tanto tendências geracionais quanto influências contextuais específicas.

Tabela 18 – Distribuição de respostas por questão e categoria de interesse

| **Área de Interesse** | **Q2** | **Q3** | **Q4** | **Q5** | **Total** |
|---|---|---|---|---|---|
| Tecnologia e Computação | 12 | 8 | 9 | 7 | 36 |
| Artes e Cultura | 9 | 11 | 5 | 8 | 33 |
| Esportes e Atividades Físicas | 8 | 2 | 1 | 2 | 13 |
| Ciências Biológicas e Saúde | 3 | 4 | 9 | 10 | 26 |
| Ciências Exatas | 2 | 3 | 8 | 4 | 17 |
| Comunicação Profissional | 5 | 9 | 4 | 7 | 25 |
| Ciências Humanas e Sociais | 4 | 5 | 6 | 3 | 18 |
| Negócios e Empreendedorismo | 1 | 2 | 1 | 3 | 7 |
| Literatura e Linguagem | 0 | 0 | 1 | 0 | 1 |

Fonte: Análise das respostas do questionário (2025). Nota: Valores representam menções por categoria, com possibilidade de múltiplas seleções por respondente.

#### *4.8.1.3 Desafios e Limitações Identificadas*

A análise revelou várias limitações importantes do questionário estruturado quando utilizado isoladamente:

**1. Granularidade Limitada:** As opções pré-definidas, embora cuidadosamente selecionadas, não conseguiram capturar nuances específicas de interesses. Por exemplo, dentro de "Artes e Cultura", não havia distinção entre artes visuais, música, teatro ou dança.

**2. Viés de Apresentação:** A ordem das opções demonstrou influência mensurável nas seleções, com as primeiras opções recebendo aproximadamente 18% mais seleções que as últimas, mesmo após randomização parcial.

**3. Ausência de Contexto:** O questionário não capturava motivações subjacentes ou experiências que informavam as escolhas dos respondentes, limitando a compreensão profunda dos interesses expressos.

### 4.8.2 Integração com Análise de Texto Livre

A verdadeira inovação do sistema híbrido emergiu da integração sinérgica entre o questionário estruturado e a análise de texto livre. Esta combinação permitiu superar as limitações individuais de cada abordagem, criando um sistema mais robusto e nuançado.

#### *4.8.2.1 Análise de Casos de Complementaridade*

A análise caso a caso revelou padrões fascinantes de como o questionário e o texto livre se complementaram mutuamente:

**Caso Exemplar 1 - Enriquecimento Contextual:**

*Questionário*: Tecnologia (Q1, Q2, Q3), Negócios (Q4)
*Texto Livre*: "Quero criar uma startup de jogos educativos para ajudar crianças com dificuldades de aprendizagem"
*Resultado Híbrido*: Tecnologia e Computação (85%) com indicação secundária em Empreendedorismo Social

Neste caso, o texto livre forneceu contexto crucial que transformou uma classificação genérica em um perfil rico e acionável, revelando não apenas interesse em tecnologia, mas também a motivação social subjacente.

**Caso Exemplar 2 - Resolução de Ambiguidade:**

*Questionário*: Distribuição uniforme entre Artes, Esportes e Ciências
*Texto Livre*: "Gosto de desenhar anatomia humana e entender como o corpo funciona para melhorar meus desenhos"
*Resultado Híbrido*: Ciências Biológicas e Saúde (72%) com componente artístico reconhecido

#### *4.8.2.2 Padrões de Reclassificação*

A análise da matriz de transição entre classificações baseadas apenas no questionário versus o sistema híbrido revelou pontos importantes:

Tabela 19 – Casos de mudança de classificação entre análise textual e sistema híbrido

| Caso | Texto do Estudante | Classificação Texto | Classificação Híbrida | Confiança |
|---|---|---|---|---|
| 1 | "Gosto de esportes e quero trabalhar com marketing esportivo" | Esportes e Atividades Físicas (84%) | Negócios e Empreendedorismo (95%) | +11% |
| 2 | "Adoro futebol e quero ser empresário no ramo esportivo" | Esportes e Atividades Físicas (79%) | Negócios e Empreendedorismo (92%) | +13% |
| 3 | "Quero ser médica, adoro biologia mas também desenho muito" | Ciências Biológicas e Saúde (65%) | Ciências Humanas e Sociais (88%) | +23% |

**Resumo: 3 mudanças em 39 casos analisados (7.7% de reclassificação)**

Fonte: Análise do sistema (2025).

A Tabela 19 apresenta os três casos em que houve mudança de classificação entre os métodos. As duas reclassificações de "Esportes" para "Negócios" demonstram a capacidade do sistema híbrido de identificar interesse profissional no setor esportivo, distinguindo-o

da prática recreativa. O terceiro caso ilustra como o questionário estruturado pode revelar interesses predominantes não evidentes apenas no texto livre.

As reclassificações mais comuns ocorreram de "Esportes e Atividades Físicas" para "Negócios e Empreendedorismo" (2 casos), indicando que alguns jovens que expressaram interesse em esportes no questionário revelaram, através do texto livre, aspirações relacionadas à gestão esportiva ou empreendedorismo no setor.

## 4.9 ANÁLISE DE CUSTOS DA INTEGRAÇÃO COM LLMS

A integração de modelos de linguagem de grande escala (LLMs) em aplicações educacionais frequentemente enfrenta resistência devido a preocupações sobre custos operacionais. Esta seção apresenta uma análise detalhada dos custos reais incorridos durante o desenvolvimento e operação do sistema, demonstrando a viabilidade econômica da solução proposta.

### 4.9.1 Comparação de Custos entre Provedores

Para avaliar a viabilidade econômica do sistema, foram analisados os modelos de menor custo de cada provedor principal. Os cálculos consideram requisições típicas do sistema educacional desenvolvido, com aproximadamente 900 tokens de entrada e 1000 tokens de saída, conforme a Tabela 20.

Os dados mostram que o Gemini 1.5 Flash apresenta o menor custo absoluto (R$ 0,003 por requisição), seguido pelo Grok 3 Mini (R$ 0,004). O DeepSeek-V3 e DeepSeek Chat custam R$ 0,007 por requisição, mas o DeepSeek Chat oferece taxa de cache superior de 85%, resultando em custo efetivo de R$ 0,001 com cache ativo.

Tabela 20 – Custos dos modelos mais econômicos por provedor

| **Provedor** | **Modelo** | **Custo/Req (R$)** | **Com Cache (R$)** | **Economia (%)** |
|---|---|---|---|---|
| Google | Gemini 1.5 Flash | 0,003 | 0,001 | 65% |
| | Gemini 1.5 Pro | 0,034 | 0,012 | 65% |
| | Gemini 2.0 Flash* | Gratuito | Gratuito | - |
| xAI | Grok 3 Mini | 0,004 | 0,001 | 65% |
| | Grok 3 Mini Fast | 0,026 | 0,009 | 65% |
| | Grok 3 | 0,099 | 0,035 | 65% |
| DeepSeek | DeepSeek-V3 | 0,007 | 0,002 | 65% |
| | DeepSeek-R1 | 0,015 | 0,005 | 65% |
| | DeepSeek Chat** | 0,007 | 0,001 | 85%*** |
| Anthropic | Claude 3 Haiku | 0,008 | 0,003 | 65% |
| | Claude 3.5 Sonnet | 0,099 | 0,035 | 65% |
| | Claude 3 Opus | 0,495 | 0,173 | 65% |
| OpenAI | GPT-3.5-turbo | 0,011 | 0,004 | 65% |
| | GPT-4o-mini | 0,025 | 0,009 | 65% |
| | GPT-4o | 0,098 | 0,034 | 65% |

Fonte: (GOOGLE, 2025), (XAI, 2025), (DEEPSEEK, 2025), (ANTHROPIC, 2025), (OPENAI, 2025). Conversão: USD 1 = R$ 5,60. *Período experimental gratuito. **Com cache nativo. ***Taxa de cache do DeepSeek.

# 5 CONCLUSÃO

O desenvolvimento deste trabalho revelou-se uma jornada transformadora que transcendeu os objetivos técnicos iniciais, evoluindo para uma exploração profunda sobre a intersecção entre tecnologia educacional e a complexa realidade dos jovens brasileiros. A construção de um sistema de classificação automatizada de interesses juvenis utilizando processamento de linguagem natural demonstrou não apenas viabilidade técnica, mas também o potencial de impacto social significativo em um contexto onde 9,2 milhões de jovens entre 14 e 29 anos encontram-se fora do sistema educacional.

A arquitetura híbrida desenvolvida, combinando questionário estruturado com análise de texto livre através do modelo BERTimbau fine-tuned, alcançou resultados promissores. A acurácia de 97,37% no dataset sintético e 86,49% com dados reais de estudantes demonstra a robustez técnica da solução. Mais significativo, porém, foi a capacidade do sistema de processar e valorizar a linguagem autêntica dos jovens, com suas variações ortográficas, gírias e expressões informais presentes em 83,8% das respostas analisadas.

A descoberta de que 64,9% dos estudantes expressaram múltiplos interesses simultâneos - exemplificada pela jovem que listou aspirações desde "vôlei" até "psicóloga, pediatra, bailarina" - iluminou uma característica fundamental da juventude contemporânea que desafia paradigmas tradicionais de orientação vocacional. O sistema desenvolvido, ao preservar esta multiplicidade através de sua arquitetura híbrida, representa um avanço significativo em direção a ferramentas que abracem a complexidade humana ao invés de forçar categorizações simplistas.

Reconhece-se, contudo, limitações importantes que demandam atenção em desenvolvimentos futuros. A validação com 44 estudantes da Escola Sinodal de São Leopoldo, embora tenha fornecido *insights* valiosos, representa apenas um fragmento da vasta diversidade brasileira. Variações regionais, socioeconômicas e culturais certamente introduziriam nuances adicionais que enriqueceriam a compreensão do fenômeno. Como apontado na avaliação crítica, a expansão para diferentes contextos geográficos e estratos sociais emerge como prioridade fundamental para garantir a generalização dos achados.

A natureza dinâmica da linguagem juvenil apresenta outro desafio significativo. Gírias e expressões que hoje caracterizam a comunicação adolescente podem tornar-se obsoletas rapidamente. A sustentabilidade do sistema dependerá da implementação de mecanismos de atualização contínua do dataset e retreinamento periódico do modelo. Propõe-se o estabelecimento de parcerias com instituições educacionais para coleta sistemática de novos dados, garantindo que o sistema permaneça relevante e preciso ao longo do tempo.

Os casos de discordância entre classificação automatizada e rotulagem humana, longe de representarem falhas, revelaram a riqueza interpretativa inerente à expressão ju-

venil. Quando "jogar bola e jogar videogame" foi classificado diferentemente por humano e máquina, evidenciou-se a necessidade de sistemas que apresentem múltiplas possibilidades interpretativas. Desenvolvimentos futuros devem aprofundar mecanismos de transparência algorítmica, permitindo que usuários compreendam e questionem as recomendações recebidas.

A análise de custos demonstrou viabilidade econômica através de estratégias de otimização como cache inteligente e seleção apropriada de modelos. Entretanto, a verdadeira sustentabilidade transcende aspectos financeiros. A manutenção de um sistema educacional ético e eficaz demanda investimento contínuo em curadoria humana, monitoramento de vieses e mecanismos de *feedback* que garantam evolução alinhada com as necessidades dos usuários.

As considerações éticas emergem como elemento central, não periférico, do desenvolvimento. O poder de influenciar trajetórias educacionais impõe responsabilidade profunda. Futuros aprimoramentos devem fortalecer mecanismos que garantam agência juvenil, evitem a criação de "bolhas de interesse" e previnam a perpetuação de estereótipos. A implementação de sistemas de contestação e revisão humana das classificações representa salvaguarda essencial contra determinismos algorítmicos.

A experiência do usuário, embora considerada no design, requer investigação mais sistemática. Estudos futuros devem incorporar avaliações formais de usabilidade, análise de métricas de engajamento e, crucialmente, *feedback* avaliação qualitativa dos jovens sobre sua experiência com o sistema. A eficácia dos elementos de gamificação em manter engajamento sustentado e promover exploração genuína de novas áreas merece atenção particular.

Este trabalho demonstra que a criação de tecnologia educacional culturalmente sensível e pedagogicamente sólida é não apenas possível, mas urgentemente necessária. Em um país marcado por desigualdades profundas, onde a orientação educacional de qualidade permanece privilégio de poucos, sistemas que democratizem o acesso a recursos de autoconhecimento e exploração vocacional representam uma contribuição fundamental para futuros mais equitativos.

A jornada revelou que desenvolver tecnologia para educação transcende desafios técnicos, demandando compreensão profunda do contexto social, respeito pela diversidade de expressões e compromisso inabalável com a justiça educacional. Os jovens que compartilharam seus sonhos através de escritas autênticas ensinaram que o verdadeiro valor da tecnologia educacional reside não em sua sofisticação algorítmica, mas em sua capacidade de amplificar vozes, validar experiências e abrir caminhos onde antes existiam apenas barreiras.

O sistema desenvolvido representa um passo inicial promissor, mas o caminho adiante demanda expansão colaborativa, validação diversificada e evolução contínua. A comunidade acadêmica e educacional é convidada a contribuir nesta jornada de construir

ferramentas que honrem a complexidade e o potencial infinito da juventude brasileira, transformando tecnologia em ponte para futuros onde cada jovem possa descobrir e realizar seu potencial único.

# REFERÊNCIAS

ANTHROPIC. **Claude API Pricing**. [*S.l.*: *s.n.*], 2025. Acessado em: 10 de março de 2025. Disponível em: https://www.anthropic.com/api-pricing.

BRASIL. **Lei nº 13.415, de 16 de fevereiro de 2017. Altera a Lei nº 9.394/1996 e institui a Política de Fomento à Implementação de Escolas de Ensino Médio em Tempo Integral**. Brasília, DF, 2017.

BRASIL. **Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)**. Brasília, DF, 2018.

BRASIL. **Lei nº 9.394, de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional**. Brasília, DF, 1996.

COMITÊ GESTOR DA INTERNET NO BRASIL. **Pesquisa sobre o uso das tecnologias de informação e comunicação nos domicílios brasileiros - TIC Domicílios 2023**. São Paulo, SP, 2023.

CONSELHO NACIONAL DE PESQUISA E DESENVOLVIMENTO. **Estatísticas sobre Escolas e Atendimento Psicológico no Brasil**. Brasília, DF, 2023.

DECI, Edward L.; RYAN, Richard M. **Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being**. [*S.l.*]: American Psychologist, 2000. v. 55, p. 68–78.

DEEPSEEK. **DeepSeek API Pricing**. [*S.l.*: *s.n.*], 2025. Acessado em: 10 de março de 2025. Disponível em: https://platform.deepseek.com/pricing.

DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In*: PROCEEDINGS of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). [*S.l.*: *s.n.*], 2019. P. 4171–4186.

FASTAPI. **FastAPI Performance Benchmarks**. [*S.l.*], 2024. Official performance documentation. Disponível em: https://fastapi.tiangolo.com/benchmarks/.

FIRTH, John R. **Papers in Linguistics 1934-1951**. London, UK: Oxford University Press, 1957.

FLORIDI, Luciano *et al.* AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. **Minds and Machines**, v. 28, n. 4, p. 689–707, 2018.

FROST, Brad. **Atomic Design**. [*S.l.*]: Brad Frost, 2013.

GOOGLE. **Gemini API Pricing**. [*S.l.*: *s.n.*], 2025. Acessado em: 10 de março de 2025. Disponível em: https://ai.google.dev/pricing.

GOOGLE CLOUD. Google Cloud Infrastructure and Network. Google LLC, 2024. Technical infrastructure overview.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Pesquisa Nacional por Amostra de Domicílios Contínua - PNAD Contínua 2023**. Rio de Janeiro, RJ, 2023.

INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA. **Censo Escolar 2023**. Brasília, DF, 2024.

MCCARTHY, John. **The Dartmouth Workshop on Artificial Intelligence**. [*S.l.*: *s.n.*], 1956. Proposal for the Dartmouth Summer Research Project on Artificial Intelligence.

MELL, Peter; GRANCE, Timothy. **The NIST Definition of Cloud Computing**. Gaithersburg, MD, 2011.

MIKOLOV, Tomas; SUTSKEVER, Ilya; CHEN, Kai; CORRADO, Greg S.; DEAN, Jeff. Distributed Representations of Words and Phrases and their Compositionality. *In*: ADVANCES in Neural Information Processing Systems 26 (NIPS 2013). [*S.l.*: *s.n.*], 2013. P. 3111–3119.

MITCHELL, Tom M. **Machine Learning**. New York, NY: McGraw-Hill, 1997.

OPENAI. **OpenAI API Pricing**. [*S.l.*: *s.n.*], 2025. Acessado em: 10 de janeiro de 2025. Disponível em: https://openai.com/pricing.

PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher D. GloVe: Global Vectors for Word Representation. *In*: PROCEEDINGS of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). [*S.l.*: *s.n.*], 2014. P. 1532–1543.

RAMÍREZ, Sebastián. FastAPI: Modern, fast, web framework for building APIs with Python, 2018. Framework documentation and benchmarks.

RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. 4th. Upper Saddle River, NJ: Pearson, 2020.

SOUZA, Fábio; NOGUEIRA, Rodrigo; LOTUFO, Roberto. BERTimbau: Pretrained BERT Models for Brazilian Portuguese. *In*: PROCEEDINGS of the 9th Brazilian Conference on Intelligent Systems (BRACIS). [*S.l.*: *s.n.*], 2020. P. 403–417.

VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Łukasz; POLOSUKHIN, Illia. Attention is All You Need. *In*: ADVANCES in Neural Information Processing Systems 30 (NIPS 2017). [*S.l.*: *s.n.*], 2017. P. 5998–6008.

WORLD ECONOMIC FORUM. **The Future of Jobs Report 2020**. Geneva, Switzerland, 2020.

XAI. **Grok API Pricing Documentation**. [*S.l.*: *s.n.*], 2025. Acessado em: 10 de março de 2025. Disponível em: https://docs.x.ai/api/pricing.