



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA ZONA SUL
CURSO DE LICENCIATURA EM COMPUTAÇÃO

FRANCISCO BORGES DA PAZ JUNIOR

SISTEMA INTELIGENTE DE PREDIÇÃO DE EVASÃO ACADÊMICA:
DESENVOLVIMENTO E INTEGRAÇÃO DE MODELO DE APRENDIZADO DE
MÁQUINA NO CURSO DE LICENCIATURA EM COMPUTAÇÃO

TERESINA
2026

FRANCISCO BORGES DA PAZ JUNIOR

SISTEMA INTELIGENTE DE PREDIÇÃO DE EVASÃO ACADÊMICA:
DESENVOLVIMENTO E INTEGRAÇÃO DE MODELO DE APRENDIZADO DE
MÁQUINA NO CURSO DE LICENCIATURA EM COMPUTAÇÃO

Trabalho de Conclusão de Curso (artigo científico) apresentado como exigência parcial para obtenção do diploma do Curso de Licenciatura em Computação do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Zona Sul.

Orientador: Prof. Msc. Kelson Carvalho Santos.

TERESINA

2026

SISTEMA INTELIGENTE DE PREDIÇÃO DE EVASÃO ACADÊMICA: DESENVOLVIMENTO E INTEGRAÇÃO DE MODELO DE APRENDIZADO DE MÁQUINA NO CURSO DE LICENCIATURA EM COMPUTAÇÃO

Francisco Borges da Paz Junior¹

Prof. Msc. Kelson Carvalho Santos²

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema inteligente voltado à predição da evasão acadêmica no curso de Licenciatura em Computação do Instituto Federal do Piauí, Campus Teresina Zona Sul. O abandono no ensino superior é um problema complexo, motivado por fatores socioeconômicos, institucionais e pedagógicos. O objetivo geral consistiu em construir uma plataforma *web* integrada a um modelo de inteligência artificial capaz de identificar padrões de risco e apresentar alertas aos gestores educacionais. Adotou-se abordagem qualitativa apoiada na engenharia de *software*, com o sistema organizado em módulos: um *back-end* em *Django*, responsável pelo processamento e pelo armazenamento dos dados, e um *front-end* em *JavaScript*, com *Axios* e *Chart.js*, responsável pela interface. O desenvolvimento do motor preditivo evoluiu de uma primeira versão heurística, baseada em pesos definidos manualmente, para a implementação de um modelo de inteligência artificial. Os experimentos avaliaram o armazenamento dos dados, a renderização do painel e o mecanismo de contingência, culminando em um teste controlado que comprovou a superioridade da inteligência artificial ao corrigir os resultados matemáticos extremos gerados pelo modelo de pesos. Conclui-se que, apesar do retreinamento do modelo com dados reais da instituição estar em fase final, a arquitetura preditiva e de *software* foi validada com sucesso por meio de uma base de dados sintética, resultando em um protótipo robusto, transparente e estruturalmente pronto para atuar na prevenção da evasão acadêmica.

Palavras-chave: aprendizado de máquina; evasão acadêmica; gestão educacional; painel interativo.

ABSTRACT

This paper presents the development of an intelligent system designed to predict academic dropout in the Licentiate Degree in Computing at the Federal Institute of Piauí, Teresina Zona Sul Campus. Higher education dropout is a complex problem driven by socioeconomic, institutional, and pedagogical factors. The general objective was to build a web platform integrated with an artificial intelligence model capable of identifying risk patterns and presenting alerts to educational managers. A qualitative approach supported by software engineering was adopted, with the system organized into modules: a Django back-end, responsible for processing and storing the data, and

¹ Graduando em Licenciatura em Computação. Instituto Federal do Piauí. catzs.20221121inf0251@aluno.ifpi.edu.br

² Professor Orientador. Doutorando em Ciência da Computação na Faculdade de Computação - FACOM da Universidade Federal de Uberlândia - UFU. Instituto Federal do Piauí. kelson@ifpi.edu.br.

a JavaScript front-end, with Axios and Chart.js, responsible for the interface. The development of the predictive engine evolved from an initial heuristic version, based on manually defined weights, to the implementation of an artificial intelligence model. The experiments assessed data storage, dashboard rendering, and the contingency mechanism, culminating in a controlled that confirmed the superiority of the artificial intelligence by correcting the extreme mathematical results generated by the weight-based model. It is concluded that, although the retraining of the model with real institutional data is in its final stage, the predictive and software architecture was successfully validated through a synthetic database, resulting in a robust, transparent prototype structurally ready to act in the prevention of academic dropout.

Keywords: machine learning; academic dropout; educational management; interactive dashboard.

Data de aprovação: 25/07/2025

LISTA DE FIGURAS

Figura 1 – Fluxograma dos métodos e tecnologias utilizadas

Figura 2 – Fluxo geral do sistema

Figura 3 – Entradas do modelo de inteligência artificial (aluno ingressante)

Figura 4 – Tela de configurações do sistema

Figura 5 – Trecho de código do mecanismo baseado em pesos

Figura 6 – Exemplo de teste do sistema de predição na fase inicial

Figura 7 – Resultado da predição com o modelo final

Figura 8 – Painel de indicadores e alertas de risco (*front-end*)

Figura 9 – Exibição da predição no modelo de pesos

Figura 10 – Exibição da predição no modelo de inteligência artificial

LISTA DE TABELAS

Tabela 1 – Dicionário de atributos e variáveis de entrada do sistema

Tabela 2 – Fatores e pesos utilizados na primeira versão do sistema de predição

Tabela 3 – Variáveis de entrada do modelo preliminar (aluno já matriculado)

Tabela 4 – Variáveis de entrada do modelo atual (aluno ingressante)

LISTA DE SIGLAS

API Interface de Programação de Aplicações (*Application Programming Interface*)

CATZS Campus Teresina Zona Sul

CSV Valores Separados por Vírgula (*Comma-Separated Values*)

DRF Django REST *Framework*

IA Inteligência artificial

IFPI Instituto Federal do Piauí

ML Aprendizado de máquina (*Machine Learning*)

MVT Model-View-Template

REST Transferência de estado representacional (*Representational State Transfer*)

SM Salário mínimo

SPA Aplicação de página única (*Single Page Application*)

SUMÁRIO

1 INTRODUÇÃO	10
2 REFERENCIAL TEÓRICO	11
2.1 A EVASÃO NO ENSINO SUPERIOR E OS DESAFIOS NA ÁREA DE COMPUTAÇÃO	12
2.2 ARQUITETURAS DE SOFTWARE DESACOPLADAS E APIS RESTFUL	12
2.3 VISUALIZAÇÃO DE DADOS PREDITIVOS E HEURÍSTICAS DE USABILIDADE	13
2.4 TRABALHOS RELACIONADOS	13
2.5 DIFERENCIAL DA SOLUÇÃO PROPOSTA	15
3 METODOLOGIA	15
3.1 LEVANTAMENTO DE TECNOLOGIAS E FERRAMENTAS	17
3.2 ESTRUTURA ORGANIZADA EM CAMADAS	18
3.3 ENTRADA DOS DADOS	19
3.4 O SISTEMA DE PREDIÇÃO DE RISCO	20
3.4.1 PRIMEIRA VERSÃO: PESOS DEFINIDOS MANUALMENTE	20
3.4.2 SEGUNDA VERSÃO: MODELO PRELIMINAR DE INTELIGÊNCIA ARTIFICIAL	22
3.4.3 VERSÃO ATUAL: MODELO VOLTADO AO ALUNO INGRESSANTE	22
3.5 ARMAZENAMENTO DOS DADOS	23
3.6 CONFIGURAÇÕES DO SISTEMA	24
4 EXPERIMENTOS E RESULTADOS	24
4.1 PASSO 1: TESTE DA VERSÃO INICIAL BASEADA EM PESOS	25
4.2 PASSO 2: EXECUÇÃO DO MODELO DE INTELIGÊNCIA ARTIFICIAL GENÉRICO	26
4.3 PASSO 3: PREDIÇÃO COM O MODELO CONCLUÍDO E INTEGRADO AO SISTEMA	26
4.4 RENDERIZAÇÃO ASSÍNCRONA DO PAINEL	27
4.5 PASSO 4: PREDIÇÃO COM O MODELO ATUAL EM COMPARAÇÃO COM O MODELO DE PESOS	28

4.6 DISCUSSÃO DOS RESULTADOS	30
4.7 LIMITAÇÕES E IMPLICAÇÕES	30
5 CONCLUSÕES	32
REFERÊNCIAS	33

1 INTRODUÇÃO

O aumento na disponibilidade de dados e o amadurecimento das técnicas de aprendizado de máquina (*Machine Learning* – ML) ampliaram o uso de sistemas inteligentes em áreas como saúde, negócios e educação. Esses sistemas transformam grandes volumes de informação em conhecimento aplicável à decisão (Domingos, 2017), o que pode apoiar o trabalho de diferentes profissionais, entre eles os gestores educacionais do Instituto Federal do Piauí (IFPI), Campus Teresina Zona Sul (CATZS).

Um sistema inteligente é um conjunto de algoritmos capazes de aprender com dados, tomar decisões e se adaptar a novas situações sem intervenção humana constante. Utilizando técnicas como aprendizado de máquina, estes sistemas são aplicados em diversas áreas, desde assistentes virtuais até análises preditivas (Russell; Norvig, 2021). No contexto educacional, ferramentas dessa natureza oferecem uma alternativa às abordagens institucionais, que atuam somente depois de o abandono já ter ocorrido e, por isso, frequentemente não conseguem reverter a situação antes que o aluno evada.

A partir deste problema, este trabalho propõe uma plataforma *web* que integra a predição de evasão a uma interface de visualização. O sistema apresenta as estimativas de risco, visualiza os estudantes em situação de vulnerabilidade e permite inserir novos dados a cada período letivo, de modo a dar suporte contínuo à gestão e ir melhorando o modelo fazendo com que ele se retroalimente ano após ano.

O interesse pelo tema decorre das investigações sobre o abandono no ensino superior, fenômeno que ultrapassa as dificuldades estritamente acadêmicas. A evasão gera prejuízos institucionais e sociais, como o desperdício de investimentos públicos e a redução do número de profissionais qualificados disponíveis ao mercado (Silva Filho *et al.*, 2007). Suas causas costumam combinar vulnerabilidades socioeconômicas, dificuldades de manter a rotina acadêmica, fragilidades organizacionais e desmotivação pessoal, e relacionam-se ao grau de integração acadêmica e social do estudante com a instituição (Tinto, 1975).

Diante desse quadro, formula-se o seguinte problema de pesquisa: a implementação de um sistema inteligente e preditivo pode fornecer informações

antecipadas que auxiliem os gestores do IFPI CATZS na tomada de decisões a fim de mitigar a evasão no curso de Licenciatura em Computação?

Como hipótese, admite-se que integrar um modelo de *ML* a uma plataforma *web* reduz a barreira de conhecimento técnico dos gestores, ao traduzir resultados estatísticos em representações visuais compreensíveis e ao gerar alertas capazes de orientar intervenções pedagógicas no momento oportuno (Few, 2006).

O objetivo geral é desenvolver uma plataforma *web* junto com um modelo de aprendizado de máquina para a predição da evasão acadêmica no curso de Licenciatura em Computação do IFPI CATZS, combinando um mecanismo preditivo a um sistema. Para alcançá-lo, definem-se os seguintes objetivos específicos:

- Mapear as variáveis de entrada (*features*) socioeconômicas e acadêmicas dos discentes, estabelecendo os requisitos de entradas no *front-end*;
- Realizar o levantamento de requisitos do sistema e definir a arquitetura de software, as ferramentas e as tecnologias;
- Projetar e implementar a integração *web* entre o *front-end* e a camada de processamento do modelo no *back-end*;
- Construir painéis interativos que centralizem indicadores sobre os alunos com maior e menor risco de evadir
- Desenvolver o sistema de predição em duas versões: uma inicial baseada em pesos, para os testes, e a definitiva baseada em aprendizado de máquina, mantendo a primeira como contingência quando o modelo treinado estiver indisponível.

2 REFERENCIAL TEÓRICO

Este capítulo reúne os fundamentos que sustentam o trabalho. A seção 2.1 discute o fenômeno da evasão no ensino superior e os desafios específicos da área de Computação. A seção 2.2 apresenta as arquiteturas de software em camadas e as *APIs REST*, que orientaram a construção da plataforma. A seção 2.3 trata da visualização de dados preditivos e das heurísticas de usabilidade que guiaram o

painel. Por fim, a seção 2.4 analisa os trabalhos relacionados e a seção 2.5 destaca o diferencial da solução proposta.

2.1 A EVASÃO NO ENSINO SUPERIOR E OS DESAFIOS NA ÁREA DE COMPUTAÇÃO

A evasão acadêmica nas instituições brasileiras de ensino superior é um problema social, econômico e pedagógico persistente. O abandono do curso não interrompe apenas um projeto individual: também representa desperdício de recursos públicos e reduz a entrada de profissionais qualificados em setores estratégicos (Silva Filho *et al.*, 2007). Em cursos de ciências exatas e tecnológicas, como a Licenciatura em Computação, o quadro tende a se agravar por fatores próprios da área, entre eles o grau de abstração exigido nas disciplinas de programação e a atração do mercado de trabalho, que com frequência absorve o estudante antes da conclusão do curso.

As causas da evasão são múltiplas e podem ser agrupadas em duas dimensões. A dimensão socioeconômica reúne fatores como vulnerabilidade financeira, necessidade de conciliar trabalho e estudo e longos deslocamentos; a dimensão acadêmica envolve baixo rendimento, reprovações acumuladas e desengajamento. Tinto (1975) associa a permanência ao grau de integração acadêmica e social do estudante na instituição. Quando as ações institucionais se restringem a medidas reativas, adotadas somente após a oficialização do abandono, os índices de retenção tendem a permanecer baixos. Esse cenário justifica um monitoramento preditivo e preventivo apoiado por ferramentas computacionais.

2.2 ARQUITETURAS DE SOFTWARE DESACOPLADAS E *APIS RESTFUL*

O desenvolvimento de sistemas *web* modernos apoia-se na separação de responsabilidades, que favorece a manutenção e a extensibilidade (Sommerville, 2018). Sistemas acadêmicos monolíticos costumavam reunir interface e regras de negócio em um único bloco de código, o que dificultava tanto a manutenção quanto a integração de módulos de ciência de dados. A arquitetura desacoplada atenua esse problema ao dividir a aplicação em duas partes independentes: o *back-end* e o *front-end*.

O *back-end*, operando como uma *API REST (Representational State Transfer)*, atua como provedor de serviços e dados sem estado (*Stateless*), usando formatos de

intercâmbio como o *JSON*. Esse modelo permite que o *framework* gerencie a persistência em banco de dados e execute as rotinas de inferência de forma isolada. O *front-end*, por sua vez, é uma aplicação de página única (*Single Page Application*–SPA) responsável pela experiência do usuário e pela renderização dos dados por meio de requisições assíncronas, o que favorece o desempenho e a modularidade (Sommerville, 2018).

2.3 VISUALIZAÇÃO DE DADOS PREDITIVOS E HEURÍSTICAS DE USABILIDADE

A utilidade de um modelo de *Machine Learning* em contextos institucionais depende de como suas previsões chegam a quem decide. Para Few (2006), painéis de indicadores não devem ser meras coleções de gráficos, e sim instrumentos de comunicação visual que reduzem a carga cognitiva do usuário e convertem dados em informação de uso imediato. Na gestão educacional, a ausência de interfaces interpretáveis costuma atuar como barreira à adoção de sistemas inteligentes por profissionais de áreas não técnicas.

Para diminuir essa barreira, o projeto de interfaces preditivas pode apoiar-se nas heurísticas de usabilidade de Nielsen (1993). Princípios como a visibilidade do estado do sistema (deixar claro o nível de risco calculado) e a correspondência entre o sistema e o mundo real (recorrer a metáforas visuais, como as cores de semáforo e os selos que indicam o nível de criticidade) elevam a usabilidade da plataforma. Combinados a relatórios visuais dinâmicos, esses princípios ajudam a transformar resultados estatísticos em alertas operacionais e, assim, a direcionar melhor as intervenções pedagógicas.

2.4 TRABALHOS RELACIONADOS

A aplicação da engenharia de *software* e da inteligência artificial ao contexto acadêmico tem sido objeto de diversas pesquisas recentes. Esses trabalhos abrangem desde a automação de processos administrativos e a digitalização de registros escolares até o projeto de interfaces de gestão e a construção de modelos preditivos de evasão. Analisar essas abordagens permite situar a contribuição deste trabalho, identificando o que já foi consolidado e quais lacunas permanecem abertas.

No campo da gestão de processos acadêmicos, Lima Junior et al. (2020) desenvolveram o sistema “Fatec Estágios”, voltado ao gerenciamento e à divulgação

de vagas de estágio. Os autores adotaram a metodologia ágil *Scrum* e integraram a solução ao sistema acadêmico da instituição, o que evidencia a viabilidade de sistemas *web* acoplados aos processos internos de uma instituição de ensino. A proximidade com o presente trabalho está justamente nessa integração de uma aplicação *web* à rotina institucional. O foco dos autores, contudo, recai sobre a automação de um fluxo administrativo, de caráter reativo: o sistema organiza informações já existentes, mas não se propõe a antecipar problemas acadêmicos, como o risco de abandono.

Na mesma direção, Franklin e Samuel Filho (2020) propuseram um *software* de gestão escolar construído em *Dart* com o *framework Flutter*, focado na automação de rotinas docentes e na centralização de notas, faltas e diários de classe em aplicativos móveis. A principal contribuição está em digitalizar o registro acadêmico, que é precisamente o insumo de que os modelos preditivos necessitam para funcionar. Ainda assim, o sistema limita-se ao armazenamento e à consulta das informações: os dados são organizados e disponibilizados, mas não há uma etapa de análise que os transforme em alertas ou em apoio à decisão. Percebe-se, assim, um estágio anterior ao que este trabalho pretende alcançar, em que a informação é registrada, mas ainda não é interpretada.

No campo do design de interfaces aplicado à gestão, Lavareda et al. (2024) propuseram uma interface baseada no método *Kanban* para setores de teste de *software*, empregando questionários de levantamento de requisitos, *personas* e *wireframes* para mapear a alocação de horas e reduzir a sobrecarga das equipes. Embora o domínio de aplicação seja distinto do educacional, o percurso de projeto centrado no usuário adotado pelas autoras é diretamente relevante para esta pesquisa. O trabalho reforça que a clareza e a organização visual da informação são determinantes para a adoção de uma ferramenta por profissionais sem formação técnica em computação, preocupação que orientou a construção da visualização por painéis proposta neste estudo.

No campo da modelagem preditiva, o trabalho mais próximo desta proposta é o de Soares e Carvalho (2023), que implementaram um sistema baseado no algoritmo *Random Forest* para prever a evasão estudantil em um ambiente universitário, na Universidade Estadual do Tocantins. Os autores estruturaram classificadores por período letivo e relataram acurácias elevadas na identificação precoce de estudantes

propensos ao abandono, o que demonstra a eficácia de técnicas de aprendizado de máquina aplicadas ao problema da evasão. A diferença central em relação ao presente trabalho está no destino do resultado: naquele estudo, o foco recai sobre o desempenho do modelo em si, e a saída é dirigida a quem domina a leitura estatística, sem uma camada de apresentação pensada para o gestor educacional que precisará agir com base naquela informação.

2.5 DIFERENCIAL DA SOLUÇÃO PROPOSTA

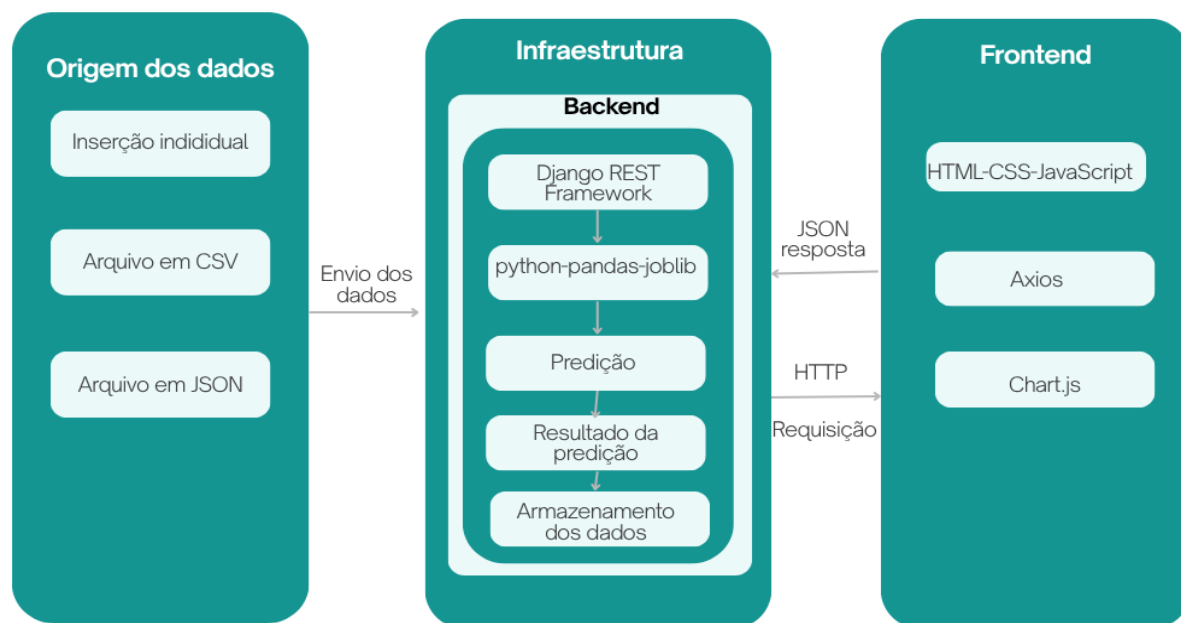
Embora esses trabalhos tratem da automação de diários escolares, da predição de evasão, da gestão de estágios e de ferramentas de estudo baseadas em modelos de linguagem, observa-se a carência de uma solução que conecte a predição de risco, obtida por algoritmos de aprendizado de máquina, à camada operacional de tomada de decisão pedagógica.

O diferencial deste trabalho está em reunir um mecanismo de predição e uma plataforma de visualização voltada aos gestores do IFPI CATZS. Diferentemente de soluções puramente computacionais, que exigem manipulação estatística e produzem apenas relatórios de probabilidades ou alguns projetos que tratam só da coleta, tratamento dos dados e criação do modelo de aprendizado de máquina a proposta traduz o resultado do modelo em indicadores de risco (alto, médio e baixo) apoiados nas heurísticas de Nielsen (1993). Além disso, quando o modelo de inteligência artificial não está disponível, o sistema mantém-se operante por meio da primeira versão, baseada nos pesos.

3 METODOLOGIA

Este estudo caracteriza-se como pesquisa qualitativa de natureza aplicada (Minayo, 2014), apoiada na engenharia de *software*, com o sistema dividido em duas camadas, *back-end* e o *front-end*, conectadas pelo fluxo de ingestão e pelas trocas *HTTP/JSON* entre cliente e servidor. A Figura 1 sintetiza os métodos e as tecnologias empregados no desenvolvimento do sistema, bem como a arquitetura que é de forma desacoplada, a fim de facilitar a manutenção.

Figura 1 – Fluxograma dos métodos e tecnologias utilizadas



Fonte: elaborada pelo autor (2026).

A figura organiza-se em três zonas. À esquerda, a origem dos dados reúne as três formas de entrada do perfil do aluno: a inserção individual, o arquivo em *CSV* e o arquivo em *JSON*. Ao centro, a infraestrutura abriga o *back-end*, no qual o *Django REST Framework* recebe os dados, as bibliotecas *Python* (*pandas* e *joblib*) preparam as informações, o componente de predição calcula o risco de evasão, o resultado da predição é gerado e, em seguida, ocorre o armazenamento dos dados. À direita, o *front-end* é responsável pela exibição, construído com *HTML*, *CSS* e *JavaScript*, com o *Axios* para a comunicação e o *Chart.js* para os gráficos do painel. As setas indicam o envio dos dados da origem para o *back-end* e a troca entre *back-end* e *front-end*, por meio de requisições *HTTP* e respostas em *JSON*.

O desenvolvimento foi organizado em etapas sequenciais: revisão teórica e levantamento de requisitos; mapeamento das variáveis de entrada; definição da estrutura do sistema; modelagem dos dados; implementação do *back-end* e do *front-end*; desenvolvimento das versões do sistema de predição; e, por fim, integração, experimentos e análise dos resultados. Para facilitar a manutenção e as futuras atualizações, o sistema foi dividido em módulos, separando a interface, o processamento dos dados e o armazenamento.

A Figura 2 ilustra o fluxo geral do sistema, da entrada dos dados à exibição dos resultados.

Figura 2 – Fluxo geral do sistema



Fonte: elaborada pelo autor (2026).

O fluxo apresenta seis etapas. A primeira é a entrada dos dados, feita pelo formulário ou pelos arquivos de importação. A segunda é a validação, que confere os campos recebidos. A terceira é o processamento, que organiza as informações no formato esperado. A quarta é o sistema de predição, que calcula o risco de evasão. A quinta é o armazenamento do aluno e do resultado no banco de dados. A sexta é a exibição dos resultados no painel.

3.1 LEVANTAMENTO DE TECNOLOGIAS E FERRAMENTAS

Para a construção da plataforma, realizou-se um levantamento criterioso das tecnologias, priorizando ferramentas de código aberto (*open-source*) que oferecessem suporte robusto tanto para o desenvolvimento *web* quanto para a integração nativa com algoritmos de inteligência artificial. O ecossistema tecnológico do projeto foi definido nas seguintes etapas:

- **Linguagem de programação e *back-end*:** Optou-se pela linguagem *Python*, que é o padrão da indústria e da academia para o desenvolvimento de soluções em ciência de dados e *Machine Learning*. Para o servidor (*back-end*), utilizou-se o *framework Django* em conjunto com o *Django REST Framework (DRF)*. Essa escolha justifica-se pela facilidade de criação de APIs seguras e pela capacidade do *Django* de operar nativamente com *scripts* de ML em *Python*.
- **Inteligência Artificial:** Para a construção, treinamento e extração de métricas do modelo *Random Forest*, utilizou-se a biblioteca *Scikit-Learn*, uma das ferramentas mais consolidadas para aprendizado de máquina em *Python*, que

permite estruturar as árvores de decisão e extrair as regras de explicabilidade do modelo.

- *Front-end e Visualização*: A interface com o usuário foi construída utilizando *HTML5*, *CSS3* e *JavaScript* puro (Vanilla JS), sem a adoção de *frameworks* pesados, garantindo leveza ao sistema. Para a comunicação assíncrona com o servidor (enviar e receber dados sem recarregar a página), adotou-se a biblioteca *Axios*. Já para a construção do painel interativo e dos alertas visuais, fundamentais para a usabilidade dos gestores, utilizou-se a biblioteca gráfica *Chart.js*.
- *Banco de Dados e Hospedagem*: Durante o desenvolvimento local, utilizou-se o banco *SQLite* padrão do *Django*. Na fase de produção e testes em nuvem, o sistema migrou para o *Neon*, um serviço de banco de dados *PostgreSQL* em nuvem e disponibilizado de forma gratuita na *Vercel* facilitando também a integração com o *front-end*, que, por sua vez, foi hospedado na plataforma *Vercel*, garantindo que a aplicação estivesse disponível na *web* para a simulação.

3.2 ESTRUTURA ORGANIZADA EM CAMADAS

A plataforma divide-se em dois módulos independentes, conectados por requisições *HTTP*: o *back-end*, responsável pelo processamento e pelo armazenamento dos dados, e o *front-end*, responsável pela interface com o usuário e visualização dos dados. Essa divisão em módulos facilita a manutenção e as atualizações futuras do sistema. O *back-end* recebe os dados enviados pelo usuário, realiza o processamento necessário, acessa o banco de dados e retorna os resultados ao sistema. Para essa comunicação entre o *back-end* e a interface, foi utilizado o *Django REST Framework*. O *front-end* foi construído com *HTML*, *CSS* e *JavaScript*, sem uso de *frameworks*, o que mantém a interface fácil de hospedar na plataforma *Vercel*.

Duas bibliotecas de *front-end* merecem destaque. O *Axios* é uma biblioteca em que o cliente envia as requisições do *front-end* ao *back-end* e recebe as respostas, permitindo atualizar a tela sem recarregar a página. O *Chart.js* é uma biblioteca de

geração de gráficos, usada para desenhar as visualizações do painel: como o gráfico de rosca da distribuição de risco e o de barras por faixa de probabilidade. Caso o *Chart.js* não seja carregado, os gráficos são desativados sem comprometer o restante da página.

A organização do sistema segue o padrão MVT (*Model-View-Template*), no qual o *Model* representa os dados e a forma como são armazenados, a *View* contém a lógica que responde às solicitações do usuário e o *Template* corresponde à apresentação das páginas. Essa separação ajuda a manter cada parte do sistema com uma responsabilidade bem definida.

3.3 ENTRADA DOS DADOS

A entrada dos dados dos alunos pode ser feita de três formas. A primeira é o cadastro individual, por meio de um formulário no qual o gestor informa os dados de um estudante. A segunda é a importação por arquivo CSV (*Comma-Separated Values*), que permite cadastrar vários estudantes de uma só vez, facilitando o registro em lote. A terceira é a importação por arquivo *JSON*, que também aceita vários registros ao mesmo tempo.

Em qualquer uma das formas, os dados informados são organizados em um formato padronizado e os campos ausentes recebem valores predefinidos. O perfil completo do aluno é guardado em um único campo (*dados_json*), o que permite armazenar tanto informações socioeconômicas quanto acadêmicas sem alterar a estrutura do banco de dados. A Tabela 1 apresenta os atributos considerados e seus valores padrão, exceto os campos marcados como "não informado", pois não é possível definir uma padronização exata para essas variáveis antes que os alunos efetivamente entrem no curso e os dados fiquem mais claros. Como o modelo é voltado para o momento do ingresso do aluno, o termo "não informado" atua como um valor provisório de segurança.

Tabela 1 – Dicionário de atributos e variáveis de entrada do sistema

Chave	Tipo	Descrição	Padrão
matricula	String (máx. 50)	Identificador único do discente (obrigatório)	Matricula[ID]
nome	String (máx. 255)	Nome completo ou identificador anonimizado	Aluno
turma	String	Turma ou curso do aluno (opcional)	não informado
sexo	Catégorico	Sexo do aluno	não informado

Chave	Tipo	Descrição	Padrão
idade	Numérico	Idade do estudante	18
estado_civil	Categórico	Estado civil do aluno	solteiro
tem_filhos	Numérico (0/1)	Indica se o aluno tem filhos	0
renda_familiar	Categórico	Faixa de renda em salários mínimos (SM)	1_3sm
situacao_trabalho	Categórico	Relação de trabalho e regime de dedicação	só estuda
escola_tipo	Categórico	Natureza da escola de ensino médio de origem	Pública
cotas	Categórico	Categoria de ingresso por cota (ex.: escola pública)	não informado
distancia	Categórico	Distância estimada até o campus	5_15km
transporte	Categórico	Modalidade principal de deslocamento	Público

Fonte: elaborada pelo autor (2026).

3.4 O SISTEMA DE PREDIÇÃO DE RISCO

O cálculo do risco de evasão é feito no arquivo *servicos.py*. Ao longo do projeto, essa parte do sistema passou por três experimentos, correspondentes a três versões do sistema de predição: a primeira baseada em pesos definidos manualmente; a segunda com um modelo preliminar de inteligência artificial, treinado apenas para validar a integração do sistema; e a terceira, o modelo definitivo, treinado com dados reais da instituição coletados e tratados.

3.4.1 PRIMEIRA VERSÃO: PESOS DEFINIDOS MANUALMENTE

A primeira versão foi elaborada no início do projeto, antes de o modelo de inteligência artificial estar disponível. Como o modelo estava em construção, optou-se por essa abordagem a fim de não atrasar o desenvolvimento do sistema, priorizando a eficiência do projeto e o andamento dos passos estipulados de construção do sistema inteligente. Em vez de aprender a partir de dados, o sistema inicial atribui pesos a fatores socioeconômicos ao risco de evasão como cotas, distância entre a residência e a instituição, renda familiar, como o estudante se locomove até o campus, também foi inserido fatores pessoais como filhos e estado civil, somando esses pesos para estimar o risco de cada estudante. A Tabela 2 apresenta os fatores considerados e o peso atribuído a cada um. Essa versão foi mantida como alternativa de segurança, acionada quando o modelo de inteligência artificial não está disponível.

Tabela 2 – Fatores e pesos utilizados na primeira versão do sistema de predição

Critério	Condição detectada	Pesos	Fundamento
Renda familiar	até 1 SM	+0,20	Pressão de subsistência econômica sobre a permanência.
	1 a 3 SM	+0,10	
	acima de 5 SM	-0,08	
Regime de trabalho	estuda e trabalha (integral)	+0,22	Conflito de tempo e sobrecarga da dupla jornada.
	estuda e trabalha	+0,12	
	só estuda	-0,05	
Tipo de escola	pública	+0,10	Aproxima assimetrias na formação de base.
	privada	-0,08	
Distância do campus	acima de 30 km	+0,18	Custos e cansaço ligados ao deslocamento.
	15 a 30 km	+0,10	
	5 a 15 km	+0,04	
	até 5 km	-0,05	
Transporte	público	+0,08	Exposição a atrasos do transporte coletivo.
	próprio	-0,05	
Fatores pessoais	tem filhos	+0,10	Responsabilidades familiares e fatores etários.
	estado civil: casado	+0,06	
	idade	+0,07	

Fonte: elaborada pelo autor (2026).

Os valores da coluna “Pesos” representam o quanto cada fator altera o escore de risco do aluno. Valores positivos aumentam o risco estimado de evasão e valores negativos o reduzem; quanto maior o valor, em módulo, maior a influência do fator no resultado final. O escore corresponde à soma desses pesos, e a versão baseada em pesos utilizava esse total para estimar o risco de cada estudante.

3.4.2 SEGUNDA VERSÃO: MODELO PRELIMINAR DE INTELIGÊNCIA ARTIFICIAL

A segunda versão substituiu as regras manuais por um modelo de *ML*. Nesta etapa, o objetivo não era obter um resultado definitivo, e sim validar a integração entre o sistema e um modelo de aprendizado de máquina, ou seja, verificar se os dados dos alunos eram corretamente enviados ao modelo, processados e devolvidos ao sistema na forma de um risco de evasão. Por esse motivo, o modelo preliminar foi treinado apenas com dados de teste, servindo unicamente para comprovar o funcionamento desse fluxo, e não representa o modelo final que será apresentado posteriormente com os dados validados.

A Tabela 3 apresenta as variáveis de entrada utilizadas por esse modelo preliminar. Como se pode observar, várias delas só existem depois que o aluno já está cursando, caso da frequência, da nota média e das reprovações, o que restringia o uso do modelo aos estudantes já matriculados.

Tabela 3 – Variáveis de entrada do modelo preliminar (aluno já matriculado)

Campo	Tipo	Descrição
-------	------	-----------

renda_familiar_sm	Numérico	Renda familiar (salários mínimos)
frequencia_pct	Numérico	Frequência média às aulas (%)
distancia_km	Numérico	Distância até o campus (km)
reprovacoes	Numérico	Número de disciplinas reprovadas
acesso_internet	Binário	Possui internet em casa (1 = sim)
idade	Numérico	Idade do aluno
transporte	Categórico	Transporte público ou próprio
Filho	Binário	Se tem filho(s)

Fonte: elaborada pelo autor (2026).

3.4.3 VERSÃO ATUAL: MODELO VOLTADO AO ALUNO INGRESSANTE

A versão atual do sistema de predição utiliza um classificador do tipo *Random Forest*, composto por 150 árvores de decisão, voltado ao aluno ingressante. Diferentemente do modelo preliminar, ele usa apenas informações disponíveis no ato da matrícula, o que permite estimar o risco de evasão desde o ingresso do aluno, antes de existirem notas ou registros de frequência. São 13 variáveis de entrada, que abrangem características pessoais, socioeconômicas, de deslocamento e de ingresso, apresentadas na Tabela 4. O modelo está em operação no sistema e, à medida que os dados reais dos estudantes da instituição forem inseridos, será retreinado para que suas predições melhorem, refletindo o comportamento da evasão no campus.

Tabela 4 – Variáveis de entrada do modelo atual (aluno ingressante)

Campo	Tipo	Valores ou descrição
idade	Numérico	Idade do candidato
estado_civil	Categórico	solteiro / casado / outro
renda_mensal_sm	Numérico	Renda mensal (salários mínimos)
acesso_internet	Binário	Possui internet em casa (1 = sim)
distancia_km	Numérico	Distância até o campus (km)
zona_residencia	Categórico	zona sul / zona sudeste / fora de Teresina
transporte	Categórico	público / próprio / carona / a pé / bicicleta
situacao_ocupacional	Categórico	somente estuda / trabalha e estuda / somente trabalha
cotista	Binário	Ingressou por cota (1 = sim)
tipo_escola	Categórico	pública / privada / mista
pcd	Binário	Portador de necessidades especiais (1 = sim)
cursou_superior	Binário	Já cursou outro curso superior (1 = sim)
tem_filhos	Binário	Tem filho(s) (1 = sim)

Fonte: elaborada pelo autor (2026).

A Figura 3 apresenta o formulário de cadastro com as variáveis de entrada do modelo, nela o gestor irá inserir os dados que irão compor os dados de entrada para

enviar ao modelo e após será processada pelo modelo de *ML* e apresentará os dados no painel.

Figura 3 – Entradas do modelo de inteligência artificial (aluno ingressante)

Fonte: elaborada pelo autor (2026).

No formulário, o gestor informa as 13 variáveis do aluno ingressante, como idade, estado civil, renda mensal, distância até o campus, meio de transporte, situação ocupacional e ingresso por cota. Ao salvar, o sistema processa a predição e exibe o resultado da predição na tela.

3.5 ARMAZENAMENTO DOS DADOS

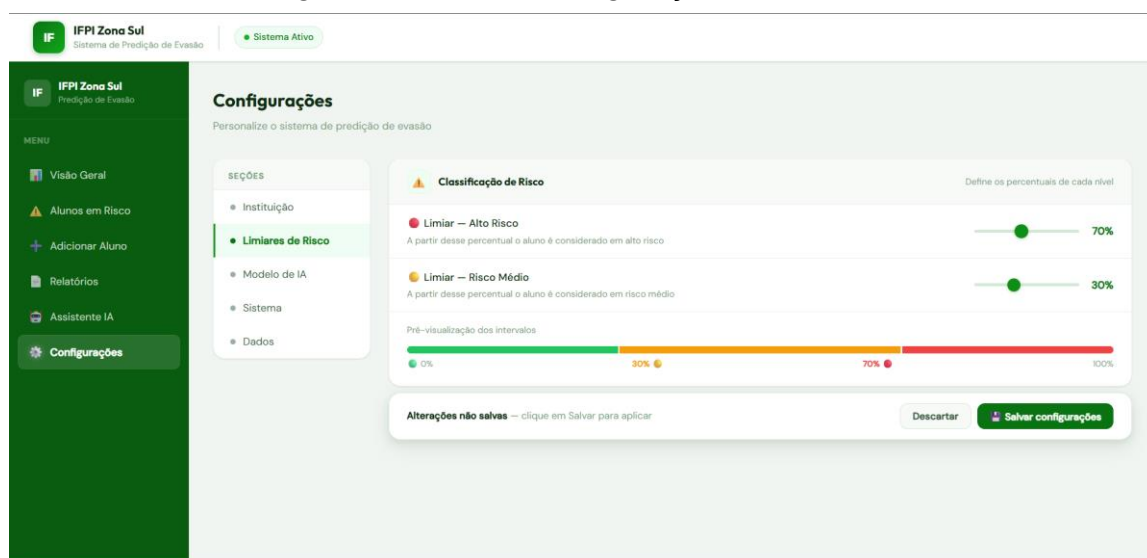
O armazenamento é feito por meio do banco de dados, gerenciado pelo *Django*. O sistema registra os dados de cada aluno e o respectivo risco de evasão e permite atualizar as informações de um estudante já cadastrado sem a necessidade de realizar um novo cadastro. Cada aluno é identificado pela matrícula e possui um único resultado de predição associado.

Durante o desenvolvimento, o sistema utilizou um banco de dados local, armazenado na própria máquina. Na fase de produção, o sistema passou a utilizar um banco de dados na nuvem, integrado à plataforma em que a aplicação está hospedada.

3.6 CONFIGURAÇÕES DO SISTEMA

A Figura 4 apresenta a tela de configurações do sistema, na qual o gestor pode ajustar os parâmetros da plataforma diretamente pela interface.

Figura 4 – Tela de configurações do sistema



Fonte: elaborada pelo autor (2026).

O sistema conta com uma área de configurações que permite aos gestores ajustar o funcionamento da plataforma diretamente pela interface, sem a necessidade de manutenção no código. Nela estão reunidas seções de instituição, limiares de risco, modelo de IA, sistema e dados. É possível, por exemplo, definir os percentuais que classificam cada nível de risco. Na configuração usada nos testes, alunos com probabilidade a partir de 70% entram em alto risco e, a partir de 30%, em risco médio, com uma pré-visualização dos intervalos na própria tela. As alterações são salvas no banco de dados e passam a valer em todo o painel.

4 EXPERIMENTOS E RESULTADOS

A validação do sistema foi conduzida por meio dos passos a serem seguidos, com o objetivo de testar as diferentes abordagens preditivas implementadas ao longo do desenvolvimento. A execução dos experimentos obedeceu a uma ordem cronológica e evolutiva, dividida em quatro cenários: o teste da versão heurística (pesos), o teste do modelo genérico de integração, a predição com o modelo treinado definitivo e, por fim, a análise comparativa da eficácia dessas abordagens.

4.1 PASSO 1: TESTE DA VERSÃO INICIAL BASEADA EM PESOS

O primeiro experimento avaliou o motor de inferência heurístico do protótipo, construído exclusivamente a partir de pesos pré definidos manualmente. Neste cenário, o sistema não utilizava aprendizado de máquina, mas sim regras fixas baseadas em fatores socioeconômicos e acadêmicos, conforme detalhado na Figura 5. A Figura 5 apresenta o trecho de código do mecanismo baseado em pesos, em que cada fator do perfil do aluno soma ou subtrai um valor fixo do escore final.

Figura 5 – Trecho de código do mecanismo baseado em pesos

```
def _predicao_simulada_legada(dados_aluno: dict) -> dict:
    """Predição heurística – fallback quando o modelo não está disponível."""
    score = 0.0

    frequencia = dados_aluno.get('frequencia_media')
    rendimento = dados_aluno.get('rendimento_medio')
    reprovacoes = dados_aluno.get('disciplinas_reprovadas')

    if frequencia is not None:
        score += max(0.0, (75.0 - float(frequencia)) / 75.0) * 0.35
    if rendimento is not None:
        score += max(0.0, (5.0 - float(rendimento)) / 5.0) * 0.35
    if reprovacoes is not None:
        score += min(float(reprovacoes) * 0.08, 0.20)

    renda = dados_aluno.get('renda_familiar', '')
    if renda == 'ate_1sm': score += 0.20
    elif renda == '1_3sm': score += 0.10
    elif renda == 'acima_5sm': score -= 0.08

    trabalho = dados_aluno.get('situacao_trabalho', '')
    if trabalho == 'estuda_trabalha_integral': score += 0.22
    elif trabalho == 'estuda_trabalha': score += 0.12
    elif trabalho == 'so_estuda': score -= 0.05

    escola = dados_aluno.get('escola_tipo', '')
    if escola == 'publica': score += 0.10
    elif escola == 'privada': score -= 0.08

    distancia = dados_aluno.get('distancia', '')
    if distancia == 'acima_30km': score += 0.18
    elif distancia == '15_30km': score += 0.10
    elif distancia == '5_15km': score += 0.04
    elif distancia == 'ate_5km': score -= 0.05
```

Fonte: elaborada pelo autor (2026).

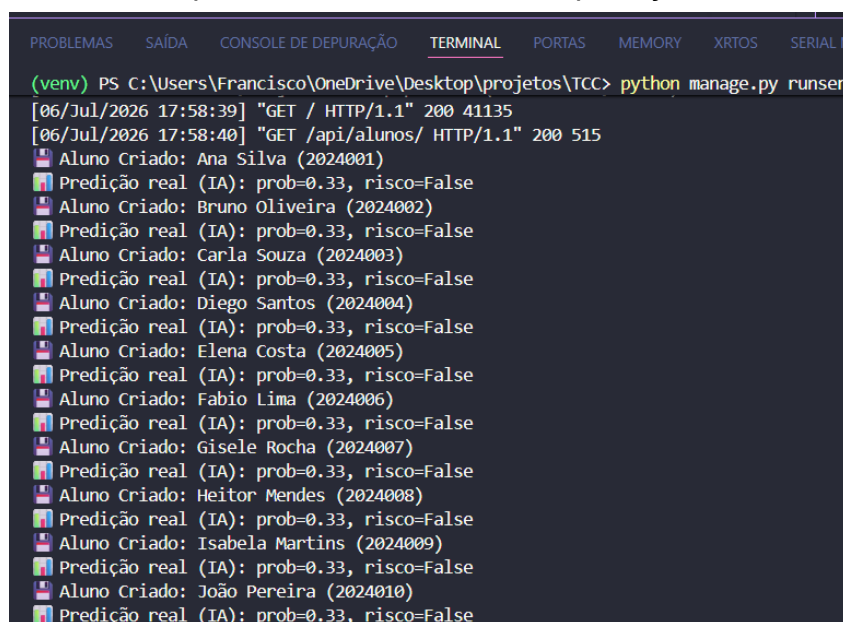
O teste consistiu em inserir dados de alunos para observar o retorno do cálculo realizado pelo mecanismo apresentado na Figura 5. O experimento confirmou que o fluxo de cálculo funcionava, porém evidenciou uma grave falha metodológica nesta abordagem: o sistema atribuía pesos de forma linear e desproporcional para alunos com vulnerabilidades múltiplas, gerando anomalias fora de série onde o risco de evasão alcançava exatos 100% caso o discente atendesse a muitos critérios de risco simultaneamente. Essa ineficiência comprovou a necessidade de transição para uma abordagem não linear de inteligência artificial que foi construída posteriormente.

4.2 PASSO 2: EXECUÇÃO DO MODELO DE INTELIGÊNCIA ARTIFICIAL GENÉRICO

Para superar as limitações dos pesos, o segundo cenário implementou um modelo de inteligência artificial preliminar. Nesta etapa, o objetivo central não era obter um classificador perfeito, mas sim validar a arquitetura de integração, garantindo que os dados inseridos no *front-end* chegassem ao *back-end*, fossem processados por um algoritmo de *Machine Learning* e devolvidos corretamente à interface.

O experimento utilizou uma base de dados sintética, contendo variáveis voltadas a alunos já matriculados, para treinar o modelo de forma genérica. O teste de execução obteve sucesso em acionar a inteligência artificial para produzir e exibir as predições, comprovando o funcionamento do fluxo de ponta a ponta sem falhas de comunicação no servidor. Contudo, por ter sido treinado com dados de teste e o modelo ser muito genérico, este modelo serviu apenas como validação de *software*, sem compromisso com a acurácia no mundo real. A Figura 6 registra esse teste inicial, com as predições retornadas pelo servidor.

Figura 6 – Exemplo de teste do sistema de predição na fase inicial



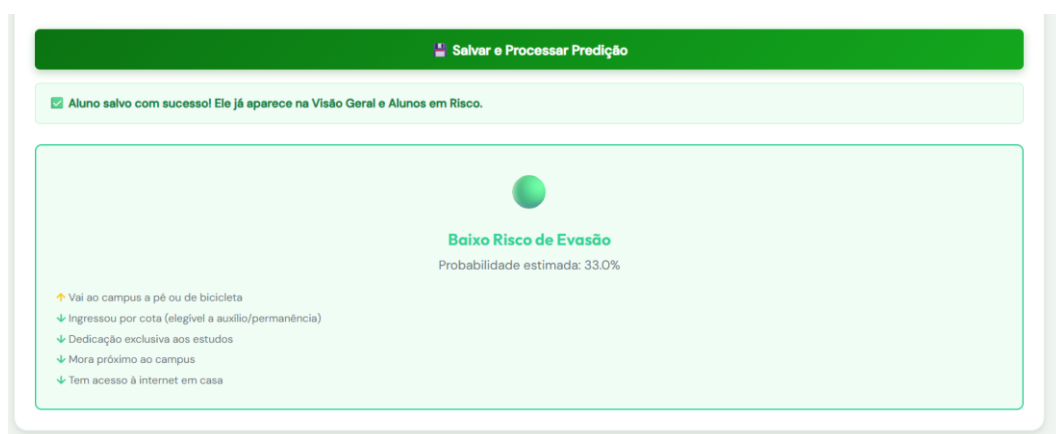
```
(venv) PS C:\Users\Francisco\OneDrive\Desktop\projetos\TCC> python manage.py runserver
[06/Jul/2026 17:58:39] "GET / HTTP/1.1" 200 41135
[06/Jul/2026 17:58:40] "GET /api/alunos/ HTTP/1.1" 200 515
Aluno Criado: Ana Silva (2024001)
Predição real (IA): prob=0.33, risco=False
Aluno Criado: Bruno Oliveira (2024002)
Predição real (IA): prob=0.33, risco=False
Aluno Criado: Carla Souza (2024003)
Predição real (IA): prob=0.33, risco=False
Aluno Criado: Diego Santos (2024004)
Predição real (IA): prob=0.33, risco=False
Aluno Criado: Elena Costa (2024005)
Predição real (IA): prob=0.33, risco=False
Aluno Criado: Fabio Lima (2024006)
Predição real (IA): prob=0.33, risco=False
Aluno Criado: Gisele Rocha (2024007)
Predição real (IA): prob=0.33, risco=False
Aluno Criado: Heitor Mendes (2024008)
Predição real (IA): prob=0.33, risco=False
Aluno Criado: Isabela Martins (2024009)
Predição real (IA): prob=0.33, risco=False
Aluno Criado: João Pereira (2024010)
Predição real (IA): prob=0.33, risco=False
```

Fonte: elaborada pelo autor (2026).

4.3 PASSO 3: PREDIÇÃO COM O MODELO CONCLUÍDO E INTEGRADO AO SISTEMA

A Figura 7 exibe a saída do teste com o modelo concluído. O terceiro experimento avaliou a versão definitiva do sistema, impulsionada pelo classificador *Random Forest* (com 150 árvores de decisão) e focada no perfil do aluno ingressante. Cadastrou-se um estudante por meio do formulário do sistema para observar o processamento em tempo real. Ao salvar os registros, o servidor processou a predição com base nos atributos do discente (como deslocamento, cotas e idade) e retornou imediatamente o resultado ao painel. Além de calcular um risco de evasão (no exemplo, classificado como "baixo risco"), esta versão do modelo demonstrou alta capacidade de explicabilidade, listando os parâmetros exatos que pesaram a favor ou contra a probabilidade final de evasão do estudante no exemplo abaixo ele demonstra que o modo como vai ao campus é um fator que pesa mais.

Figura 7 – Resultado da predição com o modelo final



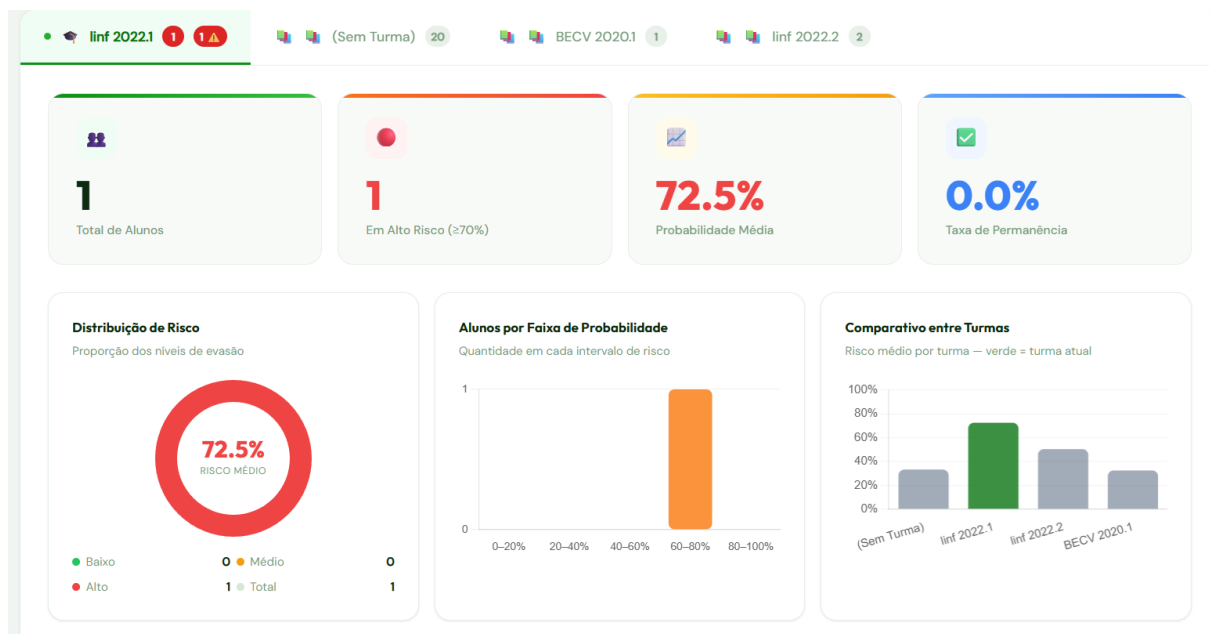
Fonte: elaborada pelo autor (2026).

4.4 RENDERIZAÇÃO ASSÍNCRONA DO PAINEL

A Figura 8 ilustra o painel de indicadores exibido. O último cenário avaliou a resposta do *front-end* após o processamento no servidor, observando como os dados são inseridos e atualizados, a fim de verificar se os valores exibidos correspondiam fielmente aos calculados pelo servidor. Os arquivos *JavaScript* modulares (*adicionar_aluno.js*, *alunos_risco.js* e *relatorios.js*) dispararam requisições assíncronas via *Axios*, que retornaram as estruturas *JSON* calculadas pelo *back-end*.

Os dados alimentaram os cartões de indicadores, as tabelas ordenadas por nível de risco e os gráficos gerenciados pelo *Chart.js*, resultando em uma interface responsiva.

Figura 8 – Painel de indicadores e alertas de risco (*front-end*)



Fonte: elaborada pelo autor (2026).

4.5 PASSO 4: PREDIÇÃO COM O MODELO ATUAL EM COMPARAÇÃO COM O MODELO DE PESOS

Além de garantir o funcionamento da arquitetura do *software*, também foi necessário avaliar a qualidade das predições. Para mostrar a diferença prática entre o modelo inicial (baseado em pesos) e o modelo de inteligência artificial, foi realizado um teste comparativo usando o mesmo perfil de aluno nas duas versões.

Nesse teste, inseriu-se no sistema um estudante fictício chamado "Kelson", preenchendo exatamente os mesmos dados socioeconômicos, acadêmicos e de deslocamento em ambos os motores de cálculo.

Na primeira versão, baseada em pesos manuais (ilustrada na Figura 9), o fato de o aluno apresentar várias vulnerabilidades fez com que o sistema somasse todas as penalidades de uma só vez. Como o cálculo é apenas uma soma direta, o resultado atingiu exatos 100% de probabilidade de evasão. Esse comportamento mostrou uma limitação importante dessa abordagem: ela apenas acumula os riscos e gera valores extremos, sem considerar que algumas características podem equilibrar ou anular as outras no mundo real. A Figura 9 mostra a exibição da predição na versão baseada em pesos.

Figura 9 – Exibição da predição no modelo de pesos

Lista de Alunos e Risco de Evasão
Alunos ordenados do maior para o menor risco. ● Alto ≥70% | ● Médio 40–70% | ● Baixo <40%

ALUNO (MATRÍCULA)	PROBABILIDADE DE EVASÃO	NÍVEL DE RISCO	DETALHES
Kelson (2022112bcv01251)	100.0%	Alto	Renda: >5SM Trabalha integral Pública Dist: 5–15km

Fonte: elaborada pelo autor (2026).

Por outro lado, quando o sistema usou o modelo de inteligência artificial com os mesmos dados de entrada (ilustrado na Figura 10), o resultado foi bem diferente. O algoritmo conseguiu analisar as informações de forma combinada e retornou um risco de 54,8%, enquadrando o aluno no nível de risco médio, o que é muito mais realista. Além de mostrar esse valor, a tela também lista quais fatores aumentaram ou diminuíram o risco, ajudando o gestor a entender o motivo daquele resultado. A Figura 10 mostra a exibição da predição na versão com o modelo de inteligência artificial, para o mesmo perfil de aluno.

Figura 10 – Exibição da predição no modelo de inteligência artificial

IFPI Painel de Evasão
Curso de Computação Carregar Dados

Visão Geral
Alunos em Risco
Relatórios
Configurações
Adicionar Aluno

Lista de Alunos e Risco de Evasão
Alunos ordenados do maior para o menor risco. ● Alto ≥70% | ● Médio 40–70% | ● Baixo <40%

ALUNO (MATRÍCULA)	PROBABILIDADE DE EVASÃO	NÍVEL DE RISCO	DETALHES
Kelson carvalho (2022112bcv01251)	54.8%	Médio	Renda: 5.5 SM Tem filhos Dist: 15km

Fonte: elaborada pelo autor (2026).

Essa discrepância comprova a eficácia da solução proposta por este trabalho. Enquanto o modelo de pesos comporta-se como uma calculadora rígida propensa a anomalias extremas, a inteligência artificial analisa os atributos de forma não linear, entregando ao IFPI uma ferramenta preditiva altamente transparente, calibrada e alinhada ao contexto humano do ambiente escolar.

4.6 DISCUSSÃO DOS RESULTADOS

Em conjunto, os cenários executados demonstram o sucesso tanto da arquitetura de *software* quanto da modelagem de inteligência artificial proposta neste trabalho. Os testes confirmaram que o sistema é capaz de receber os dados dos discentes, processar o cálculo de risco de forma rápida e exibir os resultados em um painel interativo. A evolução e a necessidade do motor de predição ficaram claras ao longo dos experimentos. A primeira versão, baseada em pesos manuais, revelou-se engessada e propensa a gerar falsos resultados extremos — como os 100% de risco, por apenas acumular penalidades de forma isolada. A transição para o modelo treinado resolveu definitivamente essas discrepâncias: a inteligência artificial provou ser capaz de analisar o perfil do aluno de maneira combinada e não linear, entregando predições muito mais realistas e, principalmente, apontando os fatores que explicam aquele resultado. Adicionalmente, a infraestrutura mostrou-se resiliente. A troca automática entre as versões assegura que, em caso de indisponibilidade do servidor principal de inteligência artificial, o sistema recorra ao antigo modelo de pesos apenas como um mecanismo de contingência. Isso garante que a plataforma não trave e que os gestores sempre tenham uma estimativa de risco no painel.

Dessa forma, os experimentos indicam que a plataforma funciona de ponta a ponta com eficácia. Embora a avaliação e o treinamento da inteligência artificial tenham utilizado uma base de dados não muito robusta nesta etapa do projeto, a arquitetura comprova que o sistema está estruturalmente pronto para ser alimentado e retreinado com os dados reais do IFPI. Ele se consolida, portanto, como uma ferramenta tecnológica viável para auxiliar na mitigação da evasão acadêmica.

4.7 LIMITAÇÕES E IMPLICAÇÕES

O trabalho encontrou algumas limitações importantes. A principal foi a indisponibilidade do modelo definitivo, treinado com dados reais da instituição, durante o período de desenvolvimento. Isso limitou a realização de testes mais próximos da realidade institucional. Por esse motivo, adotou-se inicialmente uma abordagem baseada em pesos definidos manualmente e, em seguida, um modelo preliminar treinado apenas com dados de teste, cujo objetivo foi unicamente validar o funcionamento do sistema.

Ligada a essa questão está a quantidade de dados disponíveis para o treinamento. O modelo atual foi treinado com uma base pequena de dados, já que se trata de um único curso e de poucas turmas por ano. Bases pequenas dificultam o aprendizado de padrões consistentes: o modelo pode se ajustar demais aos poucos casos que conhece e errar em perfis diferentes dos vistos no treinamento causando um *overfitting*. Situações menos frequentes, como alunos com deficiência ou com filhos, também podem aparecer em número insuficiente para que seu efeito sobre a evasão seja aprendido de forma confiável. Por isso, os primeiros resultados obtidos com dados reais deverão ser lidos com cautela e o modelo precisará ser reavaliado à medida que novas turmas forem incorporadas à base.

Outra limitação relevante foi o tempo necessário ao desenvolvimento do *software*. A implementação das funcionalidades, os ajustes na interface e a integração entre os diferentes módulos exigiram diversas etapas de testes e refinamentos até que o protótipo atingisse um funcionamento satisfatório.

A migração do banco de dados para a *online* foi realizada, mas a plataforma utilizada, por se tratar de um plano gratuito, apresenta um limite de armazenamento reduzido. Isso poderá dificultar o registro de novos dados no futuro, uma vez que o volume de informações tende a crescer a cada ano letivo.

Outra limitação observada na modelagem preditiva refere-se à dificuldade do sistema em correlacionar adequadamente as variáveis de distância residencial e modalidade de transporte. Atualmente, o modelo não pondera com precisão a interação não linear entre esses fatores, especialmente nos casos de alunos que residem próximo do campus e se deslocam a pé ou de bicicleta um exemplo está na figura 7 que ele não correlacionou a distância com o modo como o aluno se locomove até o campus. Como os atributos tendem a ser avaliados de forma isolada, o cálculo de risco pode falhar em reconhecer que a extrema proximidade geográfica neutraliza as vulnerabilidades habituais de transporte (como custos e atrasos), evidenciando a necessidade de refinar o peso dessas variáveis combinadas nas futuras atualizações e retreinamentos do algoritmo.

Há ainda uma limitação na primeira versão do sistema de predição, baseada em pesos. Embora essa abordagem seja válida para boa parte dos casos, um

estudante que atenda a todos os critérios de risco pode receber uma probabilidade de evasão de 100% , como visto na figura 9, o que representa um resultado extremo e pouco realista.

5 CONCLUSÕES

Este trabalho partiu do problema de saber se um sistema inteligente e preditivo poderia fornecer informações antecipadas capazes de auxiliar os gestores do IFPI CATZS a mitigar a evasão no curso de Licenciatura em Computação. O desenvolvimento e os experimentos realizados mostram que a proposta resultou em um protótipo funcional: a plataforma recebe dados de diferentes fontes, calcula o risco de evasão e o apresenta em um painel de interpretação, que reúne indicadores acadêmicos e socioeconômicos com o risco organizado em três faixas estruturadas.

O motor de predição do sistema evoluiu de forma clara ao longo do projeto. A primeira versão, baseada em pesos definidos manualmente, mostrou-se matematicamente limitada ao acumular penalidades e gerar falsos resultados extremos de risco. Para solucionar essa deficiência, implementou-se a arquitetura baseada no modelo de inteligência artificial. A eficácia e a superioridade dessa abordagem foram comprovadas mediante um teste comparativo utilizando um perfil de estudante fictício, o qual demonstrou a capacidade do algoritmo de processar correlações não lineares e de entregar um risco realista e com alta explicabilidade.

Conclui-se que, apesar da baixa quantidade de dados, a arquitetura preditiva e de *software* foi validada com sucesso por meio da base de dados inserida para os testes. O protótipo cumpriu plenamente o objetivo de demonstrar o funcionamento do fluxo completo, de ponta a ponta. Como trabalhos futuros, prevê-se: finalizar o retreinamento do modelo com mais dados; realizar uma avaliação prática do impacto do sistema sobre a atuação preventiva dos gestores; integrar a plataforma nativamente aos sistemas de controle acadêmico do instituto; e, por fim, ampliar o uso para os demais cursos ofertados pelo campus.

REFERÊNCIAS

DOMINGOS, Pedro. *O algoritmo mestre: como a busca pelo algoritmo de aprendizado de máquina definitivo recriará nosso mundo*. São Paulo: Novatec, 2017.

FEW, Stephen. *Information dashboard design: the effective visual communication of data*. Sebastopol: O'Reilly Media, 2006.

FRANKLIN, Matheus Maião; SAMUEL FILHO, Ronaldo Aparecido. Desenvolvimento de um Sistema de Gestão Escolar com o uso da Linguagem Dart com *Framework* Flutter. *Revista Eletrônica e-Fatec*, Garça, v. 10, n. 1, p. 1-10, 2020.

LAVAREDA, Adriely de Souza *et al.* Proposta de interface do sistema Kanban para gestão de projetos em um setor de teste de software. *Revista Produção Online*, v. 23, n. 4, p. 5155, 2024.

LIMA JUNIOR, Antonio Carlos *et al.* *Desenvolvimento de um sistema para gestão de estágios*. 2020. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) – Faculdade de Tecnologia de Americana, Centro Paula Souza, Americana, 2020.

MINAYO, Maria Cecília de Souza. *O desafio do conhecimento: pesquisa qualitativa em saúde*. 14. ed. São Paulo: Hucitec, 2014.

NIELSEN, Jakob. *Usability engineering*. Boston: Academic Press, 1993.

RUSSELL, Stuart; NORVIG, Peter. *Inteligência artificial*. 4. ed. Rio de Janeiro: LTC, 2021.

SILVA FILHO, Roberto Leal Lobo *et al.* A evasão no ensino superior brasileiro. *Cadernos de Pesquisa*, São Paulo, v. 37, n. 132, p. 641-659, 2007.

SOARES, Leandra Cristina Cavina Piovesan; CARVALHO, Rafael Lima de. Sistema inteligente para previsão de evasão escolar em um ambiente universitário. *Revista Humanidades & Inovação*, Palmas, v. 10, n. 9, 2023.

SOMMERVILLE, Ian. *Engenharia de software*. 10. ed. São Paulo: Pearson Education do Brasil, 2018.

TINTO, Vincent. Dropout from higher education: a theoretical synthesis of recent research. *Review of Educational Research*, Washington, v. 45, n. 1, p. 89-125, 1975.