

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA
PARAÍBA CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS**

**SISULENS: UMA FERRAMENTA PARA A ANÁLISE DA EVOLUÇÃO
DA PROCURA DOS CURSOS SUPERIORES DISPONÍVEIS NO SISU**

VERÍSSIMO PEREIRA NÓBREGA TERCEIRO

Cajazeiras

2026



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

VERÍSSIMO PEREIRA NOBREGA TERCEIRO

**SISULENS: UMA FERRAMENTA PARA A ANÁLISE DA EVOLUÇÃO
DA PROCURA DOS CURSOS SUPERIORES DISPONÍVEIS NO SISU**

Trabalho de Conclusão de Curso apresentado junto ao
Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas do Instituto Federal de
Educação, Ciência e Tecnologia da Paraíba - Campus
Cajazeiras, como requisito à obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Dr. Fábio Gomes de Andrade

Aprovada em: **14 de agosto de 2026.**

Prof. Dr. Fábio Gomes de Andrade - Orientador

IFPB - Campus Cajazeiras

Prof. Me. Francisco Paulo de Freitas Neto - Avaliador

IFPB - Campus Cajazeiras

Prof. Me. Diogo Dantas Moreira - Avaliador

Documento assinado eletronicamente por:

- **Fabio Gomes de Andrade**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 15/08/2026 08:22:05.
- **Diogo Dantas Moreira**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 15/08/2026 09:39:12.
- **Francisco Paulo de Freitas Neto**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 15/08/2026 11:30:53.

Este documento foi emitido pelo SUAP em 15/08/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 923719
Verificador: 2c93558e29
Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000
<http://ifpb.edu.br> - (83) 3532-4100

IFSPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

N754s Nóbrega Terceiro, Veríssimo Pereira.
Sisulens : uma ferramenta para a análise da evolução da procura dos cursos superiores disponíveis no SISU / Veríssimo Pereira Nóbrega Terceiro.– 2026.

52f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2026.

Orientador(a): Prof. Dr. Fabio Gomes de Andrade.

1. Sistema de gestão. 2. Educação superior. 3. SISU. 4. Políticas educacionais. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

AGRADECIMENTOS

A meus pais por todo apoio e incentivo que possibilitaram minha formação.

Aos amigos que fiz durante a graduação, pela companhia em momentos importantes.

Ao professor Fabio, pela orientação e disposição durante o desenvolvimento do trabalho.

RESUMO

Este trabalho descreve o desenvolvimento de uma ferramenta de visualização dos dados das inscrições do Sistema de Seleção Unificado (SISU), abrangendo o período de 2017 a 2023, sendo este o intervalo com os dados mais recentes disponíveis no momento de escrita deste trabalho, com o objetivo de aprofundar a análise sobre o acesso ao ensino superior nas instituições públicas brasileiras. Com base nos microdados disponíveis no Portal de Dados Abertos do Ministério da Educação, em conformidade com a Lei de Acesso à Informação, a ferramenta transforma dados brutos em representações visuais intuitivas, por meio de gráficos e dashboards. Com essa abordagem, facilita-se a identificação de tendências, a evolução do número de candidatos e a distribuição geográfica e desempenho dos candidatos, contribuindo assim para a formulação de políticas públicas e decisões estratégicas na área educacional. A ferramenta foi desenvolvida utilizando tecnologias modernas, garantindo acessibilidade e eficiência na disseminação do conhecimento.

Palavras-chave: SISU, educação superior, dados abertos, visualização de dados.

ABSTRACT

This work describes the development of a data visualization tool for the enrollment data of the Unified Selection System (SISU), covering the period from 2017 to 2023, which represents the most recent data available at the time of writing this study, with the objective of deepening the analysis of access to higher education in Brazilian public institutions. Based on the microdata available from the Ministry of Education's Open Data Portal, in compliance with the Access to Information Law, the tool transforms raw data into intuitive visual representations, using graphs and dashboards. With this approach, it facilitates the identification of trends, the evolution of the number of applicants, and the geographic distribution and performance of candidates, thereby contributing to the formulation of public policies and strategic decisions in the educational field. The tool was developed using modern technologies, ensuring accessibility and efficiency in the dissemination of knowledge.

Keywords: SISU, higher education, open data, data visualization.

LISTA DE FIGURAS

Figura 1 - O Processo ETL.....	23
Figura 2 - Arquitetura dos módulos da aplicação.....	29
Figura 3 - Página de Visão Geral.....	40
Figura 4 - Página de Busca de Cursos com ofertas agregadas.....	41
Figura 5 - Página de Lista de Ofertas de uma denominação de curso.....	41
Figura 6 - Página de Detalhe do Curso (aba Histórico e Notas).....	43
Figura 7 - Página de Detalhe do Curso (aba Perfil dos Candidatos).....	44
Figura 8 - Página de Desempenho por UF.....	45
Figura 9 - Página de Comparação de Cursos.....	46

LISTA DE QUADROS

Quadro 1 - Esquema lógico da tabela sisu_inscricoes.....	25
Quadro 2 - Views materializadas do banco de dados.....	32
Quadro 3 - Registro de endpoints da API.....	36
Quadro 4 - Rotas e páginas da SPA.....	38

LISTA DE CÓDIGOS FONTE

Código-fonte 1 - Definição da view materializada mv_curso_municipio_candidato.....	34
Código-fonte 2 - Hook de Cache-Control aplicado a todas as respostas.....	35

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CLI	<i>Command line interface</i>
CSV	<i>Comma separated values</i>
ENEM	Exame Nacional do Ensino Médio
ETL	<i>Extract, Transform, Load</i>
IES	Instituição de Ensino Superior
LAI	Lei de Acesso à Informação
MEC	Ministério da Educação
ORM	<i>Object Relational Mapper</i>
SISU	Sistema de Seleção Unificada
TCC	Trabalho de Conclusão do Curso
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

1. INTRODUÇÃO.....	11
1.1 MOTIVAÇÃO.....	13
1.2 OBJETIVOS.....	14
1.2.1 OBJETIVO GERAL.....	14
1.2.2 OBJETIVOS ESPECÍFICOS.....	14
2. METODOLOGIA.....	16
3. SOLUÇÃO PROPOSTA.....	18
3.1 HISTÓRIAS DE USUÁRIO.....	18
3.2 REQUISITOS FUNCIONAIS.....	20
3.3 REQUISITOS NÃO FUNCIONAIS.....	21
3.4 O PROCESSO DE COLETA DE DADOS.....	22
3.5 O ESQUEMA LÓGICO DO BANCO DE DADOS.....	25
3.6 ARQUITETURA DO SOFTWARE.....	28
3.7 IMPLEMENTAÇÃO.....	30
3.7.1 IMPLEMENTAÇÃO DA CAMADA DE DADOS.....	30
3.7.2 IMPLEMENTAÇÃO DO BACK-END.....	34
3.7.3 IMPLEMENTAÇÃO DO FRONT-END.....	37
4. CONSIDERAÇÕES FINAIS.....	48
REFERÊNCIAS.....	50

1. INTRODUÇÃO

O acesso à educação superior na maior parte das instituições públicas brasileiras atualmente é realizada principalmente por meio do *Sistema de Seleção Unificada* (SISU). Trata-se de um sistema no qual as instituições públicas de ensino superior oferecem as vagas dos seus cursos para os candidatos participantes do *Exame Nacional do Ensino Médio* (ENEM). Os candidatos escolhem os cursos do seu interesse e o SISU os classifica a partir de suas notas. O SISU começou a ser implementado em 2010, após a reformulação do ENEM ocorrida no ano anterior, e tem como principal objetivo democratizar o acesso ao ensino superior e mitigar desigualdades sociais. Os participantes podem se candidatar de acordo com os seus interesses em até duas opções de curso.

Entre os anos de 2017 e 2023, o SISU gerou uma quantidade significativa de dados relativos às inscrições. Os microdados referentes a essas inscrições estão disponíveis para *download* no Portal de Dados Abertos do Governo Federal¹. Essa disponibilidade se dá pela aplicação da *Lei de Acesso à Informação* (LAI) (BRASIL, 2011), que assegura que qualquer cidadão tem o direito de acessar dados e informações públicas de forma transparente e acessível. A LAI alinha-se à definição de dados abertos adotada pelo *World Wide Web Consortium* (W3C) no Manual dos Dados Abertos produzido pela *Open Knowledge Foundation* (OKF) (OKF, 2011), que define que o dado aberto é um dado que pode ser livremente utilizado, reutilizado e redistribuído por qualquer pessoa.

Desde a promulgação da LAI, uma grande quantidade de dados abertos têm sido disponibilizada por órgãos públicos municipais, estaduais e federais. Por exemplo, o portal de dados abertos do governo federal oferece hoje acesso a mais de quatorze mil conjuntos de dados disponibilizados por mais de duzentos órgãos. Além disso, existem governos estaduais, como os dos estados de São Paulo² e Rio Grande do Sul³, e municipais, como as prefeituras de Recife⁴ e Belo Horizonte⁵, que disponibilizam os seus próprios portais de dados abertos. O desenvolvimento desses portais faz com que hoje exista uma grande quantidade de dados abertos

¹ Disponível em <<https://dadosabertos.mec.gov.br/sisu>>

² <http://catalogo.governoaberto.sp.gov.br/>

³ <https://dados.rs.gov.br/>

⁴ <http://dados.recife.pe.gov.br/>

⁵ <https://dados.pbh.gov.br/>

governamentais disponíveis para a população, e qualquer cidadão pode fazer o download, visualizar, analisar ou criar aplicações de software sobre esses dados livremente.

Uma característica importante dos dados abertos governamentais é que muitos deles são volumosos, ou seja, são compostos por uma grande quantidade de registros. A análise de grandes volumes de dados apresenta desafios significativos para a extração de informações compreensíveis. Ferramentas e métodos eficientes são essenciais para organizar, filtrar e apresentar esses dados de maneira a gerar insights relevantes. Sem plataformas adequadas de visualização, a identificação de padrões, tendências e oportunidades pode ser prejudicada, comprometendo a eficácia na tomada de decisões.

No Portal Único de Acesso ao Ensino Superior do MEC existe uma ferramenta de visualização⁶ de dados baseada em *Power BI* que oferece uma tabela interativa que permite aos usuário filtrar e visualizar informações relacionadas aos programas SISU, FIES e ProUni, incluindo detalhes sobre cursos, quantidade de vagas, notas de corte e localização das instituições de ensino, com base de dados limitada aos anos de 2020 até 2023, no momento do desenvolvimento deste trabalho.

Embora essa ferramenta forneça uma visão abrangente dos programas mencionados, focando em instituições públicas e privadas, há uma necessidade específica de aprofundar a análise dos dados de inscrições do SISU. Por exemplo, a ferramenta atual não permite a análise sobre a evolução do número de participantes ao longo do tempo, os cursos cuja procura tem aumentado ou diminuído, o perfil demográfico dos alunos que buscam cada tipo de curso, entre outros.

O desenvolvimento de uma ferramenta especializada para a visualização desses dados pode oferecer conhecimento mais detalhado e útil para pesquisadores, gestores e demais interessados. Tal ferramenta permitiria transformar dados brutos em informações estratégicas, contribuindo para uma compreensão mais profunda da dinâmica de procura por cursos superiores em instituições públicas no Brasil e, conseqüentemente, para a melhoria das políticas educacionais.

⁶ Disponível em <<https://acessounico.mec.gov.br/busca>>

1.1 MOTIVAÇÃO

A disponibilização de dados abertos no setor público, embora importante para garantir a transparência e o acesso à informação, não é suficiente para que conhecimento significativo seja extraído, sendo assim necessário melhorar sua análise e visualização (Silva, 2018). Podemos identificar que os dados do SISU, por si só, são apenas o meio para compreender uma problemática mais ampla: a dificuldade de monitorar e analisar a procura por cursos superiores no Brasil, embora haja algumas análises sobre a influência do SISU nas escolhas dos estudantes.

Por exemplo, o artigo publicado por Ariovaldo e Nogueira (2021) investiga como o SISU impactou a escolha de cursos superiores na Universidade Federal de Viçosa, com foco nas licenciaturas, que apresentaram aumento na relação candidato por vaga após a implementação desse sistema. Os resultados indicaram que o SISU influencia as escolhas dos estudantes, tornando-as mais estratégicas, e que as desigualdades sociais e escolares são fatores relevantes na tomada de decisão dos candidatos. O estudo mostra ainda como é desafiador obter uma compreensão completa sobre as tendências de procura pelos cursos de graduação oferecidos pelo SISU, quais cursos mostram tendência de aumento ou perda interesse dos inscritos ao longo do tempo e qual é o perfil detalhado dos alunos que se inscrevem em determinados cursos.

Os dados disponibilizados pelo Ministério da Educação e Cultura (MEC) referentes às inscrições do SISU compreendem os anos de 2017 até 2023, e contêm informações sobre o candidato, a Instituição de Ensino Superior (IES) na qual ele desejou ingressar e o curso escolhido. A partir dessas informações, é possível realizar análises sobre a evolução da procura por cursos ao longo dos anos, identificar tendências de preferência por áreas de conhecimento, mapear a distribuição geográfica e desempenho dos candidatos e avaliar a distribuição geográfica das inscrições.

Com o objetivo de atribuir sentido e exibir para a população as tendências, correlações e *insights* que seriam de difícil identificação sobre a busca pelo acesso ao ensino superior no Brasil, este trabalho propõe uma ferramenta que permite uma análise e visualização mais simples desses dados. Essa ferramenta traduz informações brutas em representações visuais intuitivas, como gráficos e

dashboards, permitindo que esses dados sejam compreendidos de forma clara e acessível, facilitando a tomada de decisões e a disseminação do conhecimento para um público amplo.

Para os estudantes, a ferramenta permite decisões informadas sobre a oferta de cursos e políticas de inclusão. Há potencial também para ser um recurso para pesquisadores que estudam padrões educacionais e socioeconômicos, facilitando a identificação de tendências e correlações significativas ao longo do tempo. Ademais, para gestores, a ferramenta desempenha um papel na formulação e avaliação de políticas públicas de educação. Ao mesmo tempo, o trabalho promove a transparência no processo seletivo do SISU, oferecendo uma interface acessível que permite aos cidadãos acompanhar e entender melhor sobre o sistema.

1.2 OBJETIVOS

Esta seção descreve os objetivos deste Trabalho de Conclusão de Curso (TCC).

1.2.1 OBJETIVO GERAL

O objetivo geral deste TCC consiste no desenvolvimento de uma ferramenta que permita a análise dos dados sobre o acesso ao ensino superior nas instituições públicas do país.

1.2.2 OBJETIVOS ESPECÍFICOS

O trabalho também tem os seguintes objetivos específicos:

- Tornar acessíveis a toda a população as informações obtidas pela análise dos dados sobre o acesso ao ensino superior;
- Facilitar a identificação de tendências de procura pelos cursos de graduação oferecidos pelo SISU ao longo dos anos;
- Ajudar os gestores de instituições de ensino no processo de tomada de decisões, por meio da disponibilização de dados importantes como o número de alunos que procuram a instituição e o perfil desses alunos;
- Disponibilizar dados sobre o interesse dos estudantes, a demanda por cursos superiores e o desempenho no ENEM, permitindo uma compreensão detalhada das tendências educacionais;

- Auxiliar os gestores da educação na formulação de políticas públicas voltadas para a ampliação do acesso ao ensino superior na rede pública, com base nas informações fornecidas.

2. METODOLOGIA

Para garantir o êxito de um projeto, é fundamental definir com clareza os processos e métodos de execução que conduziu ao alcance dos objetivos propostos. Nesse sentido, é imprescindível compreender não apenas o propósito central do trabalho, mas também as técnicas e abordagens que foram empregadas ao longo do seu desenvolvimento. Assim, para a implementação deste TCC, foram delineadas as seguintes atividades:

- **Revisão do estado da arte (A1):** nesta etapa, foi realizada uma revisão detalhada da literatura, estudos e projetos anteriores que tratam de temas correlatos ao objeto deste trabalho, como, por exemplo, a ferramenta de visualização de dados baseada em *Power BI* no portal único de acesso ao ensino superior do MEC. Esse referencial teórico e prático serviu para manter o foco na proposta central e assegurar a originalidade do projeto;
- **Elaboração do documento (A2):** esta atividade compreende a redação deste documento de conclusão de curso, que integra os fundamentos teóricos, metodológicos e as expectativas quanto aos resultados do trabalho;
- **Levantamento de requisitos (A3):** durante esta atividade, foram identificados e avaliados os requisitos essenciais para atingir os objetivos gerais e específicos do projeto, bem como foram definidas as funcionalidades e ferramentas que a ferramenta deveria oferecer aos seus usuários;
- **Definição da arquitetura do projeto (A4):** nesta atividade, estabeleceu-se a estrutura da ferramenta, determinando os seus módulos e a lógica de comunicação entre eles, de forma a garantir uma organização clara e coesa para o desenvolvimento da solução;
- **Desenvolver o mecanismo de coleta de dados (A5):** nesta etapa foi implementado um fluxo para a extração, limpeza e armazenamento dos dados, assegurando sua integridade e qualidade;
- **Prototipação do sistema (A6):** nesta etapa foram criados protótipos com base nos requisitos coletados, estabelecendo a referência para a implementação e documentando os elementos de design, experiência e interface do usuário;

- **Implementação (A7):** nesta etapa ocorreu o desenvolvimento e a integração dos módulos da ferramenta proposta, conforme a arquitetura definida, realizando a codificação efetiva da solução.

3. SOLUÇÃO PROPOSTA

Este capítulo descreve o desenvolvimento da ferramenta proposta por este TCC. Inicialmente, são apresentadas as histórias de usuário, e posteriormente, os requisitos funcionais e não funcionais. Em seguida, o capítulo trata do processo de coleta, tratamento e organização dos dados no banco de dados. Depois, o capítulo detalha a sua arquitetura e, por fim, descreve a sua implementação.

3.1 HISTÓRIAS DE USUÁRIO

As histórias de usuário descritas a seguir refletem as funcionalidades implementadas no sistema.

US01 - Painel de visão geral

Como usuário, **quero** ver um painel de visão geral ao acessar o *SisuLens*, **para que** eu tenha uma noção rápida do volume e distribuição das inscrições no SISU.

Critérios de aceitação:

- O painel exibe o total de inscrições de 1ª opção por ano em um gráfico de linha (2017–2023);
- O painel exibe um mapa coroplético do Brasil com a distribuição de inscrições por UF da IES;
- O painel exibe os 5 cursos com maior número de inscrições no último ano disponível;
- O painel exibe os 5 estados com maior número de candidatos (UF de origem);
- O mapa é clicável e redireciona para a tela de busca de cursos com o filtro de UF da IES aplicado.

US02 - Busca e seleção de curso

Como candidato ao SISU ou pesquisador, **quero** buscar um curso por nome e filtrar os resultados por grau, turno, UF da IES, região e sigla da IES, **para que** eu encontre rapidamente o recorte que me interessa.

Critérios de aceitação:

- O campo de busca aceita texto parcial e retorna sugestões em tempo real;

- É possível filtrar por: grau (bacharelado, licenciatura, tecnólogo), turno, UF da IES, região e sigla da IES;
- A lista de resultados exibe: nome do curso, IES, UF e número total de inscrições no último ano;
- A seleção de um curso abre a tela de análise de curso (US03).

US03 - Análise de um curso

Como candidato ao SISU ou pesquisador, **quero** visualizar o histórico de um curso específico, **para que** eu entenda sua evolução de demanda, concorrência e nota de corte ao longo dos anos.

Critérios de aceitação:

- A tela exibe o nome do curso, IES, campus, grau, turno e modalidades de concorrência;
- Gráficos de linha mostram a evolução do número de inscrições, nota de corte e nota média por ano e modalidade;
- A distribuição de candidatos por município de origem é apresentada, aplicando supressão de dados para grupos com menos de 5 candidatos para conformidade com a LGPD.

US04 - Comparação entre cursos

Como candidato ao SISU, **quero** comparar dois cursos lado a lado, **para que** eu possa tomar uma decisão mais informada sobre qual opção escolher.

Critérios de aceitação:

- O usuário pode selecionar até 2 cursos para comparação simultânea;
- A tela exibe, para cada curso: evolução de inscrições, nota de corte e nota média dos candidatos;
- Os gráficos de comparação utilizam eixos Y normalizados para facilitar a leitura;
- O usuário pode trocar um dos cursos comparados sem perder a seleção do outro, e o estado da comparação é compartilhável por meio da URL.

US05 - Análise do perfil dos candidatos

Como pesquisador, **quero** analisar o perfil dos candidatos de um curso, **para que** eu entenda quem são os inscritos e o seu desempenho.

CrITÉrios de aceitação:

- A tela exibe a distribuição por sexo dos candidatos inscritos, de forma agregada, em um gráfico de rosca;
- A tela exibe a distribuição por município de origem dos candidatos;
- A tela exibe o histograma de notas dos candidatos inscritos;
- Os dados podem ser filtrados por ano e por modalidade de concorrência;
- Todos os agrupamentos com menos de 5 candidatos são suprimidos.

US06 - Análise de desempenho por UF

Como formulador de políticas públicas, **quero** comparar a nota média dos candidatos por estado ao longo dos anos, **para que** eu possa acompanhar tendências de desempenho por UF.

CrITÉrios de aceitação:

- Um gráfico de linha exibe a nota média dos candidatos por UF ao longo dos anos;
- É possível selecionar múltiplas UFs para comparação simultânea no mesmo gráfico;
- O gráfico mostra a média nacional como linha de referência;
- É possível filtrar por região e por nome do curso para refinar a comparação.

3.2 REQUISITOS FUNCIONAIS

Para o desenvolvimento deste TCC foram definidos os seguintes requisitos funcionais:

- **RF01 - Painel de visão geral:** a ferramenta deve exibir, na página inicial, um painel com métricas e visualizações chave (Inscrições de 1ª opção por ano, Distribuição por UF da IES, Top 5 cursos, Top 5 UFs de candidatos), com filtro disponível por ano;
- **RF02 - Consulta e filtragem de cursos:** a ferramenta deve disponibilizar um endpoint de busca de cursos com filtros por texto parcial, grau, turno, UF da IES, região, sigla da IES e ano, retornando detalhes como nome do curso, IES, campus e total de inscrições;
- **RF03 - Análise histórica de curso:** dado um curso e um campus, a ferramenta deve retornar as séries temporais de total de inscrições, nota de

corte e nota média dos candidatos, bem como a distribuição de candidatos por origem geográfica (município - UF), agrupados por ano, edição e modalidade de concorrência. Grupos com menos de 5 candidatos são suprimidos para conformidade com a LGPD;

- **RF04 - Comparação entre dois cursos:** a ferramenta deve permitir a comparação simultânea de até dois cursos, exibindo as séries temporais de inscrições, nota de corte e nota média em gráficos com o mesmo eixo Y;
- **RF05 - Análise de desempenho por UF da IES:** a ferramenta deve retornar a nota média dos candidatos agrupada por UF da IES (campus) e ano, com filtros por região, estado e nome do curso, incluindo a média nacional como referência.
- **RF06 - Análise de perfil de candidatos:** a ferramenta deve retornar a distribuição por sexo, por município de origem e o histograma de notas dos candidatos inscritos, com opção de filtrar por ano e modalidade e com supressão de grupos com menos de 5 registros.

3.3 REQUISITOS NÃO FUNCIONAIS

Os requisitos não-funcionais especificam as qualidades do sistema:

- **RNF01 - Desempenho:** todas as respostas da API devem ser retornadas em tempo abaixo de 200 milissegundos. Este limite foi definido considerando a fluidez na interatividade com os dashboards. Para alcançar essa meta, foram usadas visões materializadas para a agregação de dados e índices estratégicos para consultas parametrizadas;
- **RNF02 - Privacidade e conformidade com a LGPD:** A API não deve expor dados individuais. Para isso, toda informação deve ser agregada (contagens, médias) e os agrupamentos com menos de 5 registros devem ser suprimidos;
- **RNF03 - Segurança da API:** deve ser feita a implementação de *rate limiting* (máximo 100 requisições/min/IP), CORS restritivo e validação de parâmetros de entrada. A prevenção de injeção de SQL decorre da parametrização das consultas pelo Drizzle ORM;
- **RNF04 - Responsividade:** a interface deve ser utilizável em telas com largura mínima de 360px, com layout adaptado para mobile, tablet e desktop;

- **RNF05 - Disponibilidade dos dados:** o banco de dados deve ser somente leitura pela API, e a base de dados deve ser estática após a execução do ETL.

3.4 O PROCESSO DE COLETA DE DADOS

Os dados brutos selecionados para este estudo referem-se às inscrições realizadas nos processos seletivos do SISU, disponíveis para download em formato CSV no portal de dados abertos do MEC⁷. Atualmente, estão acessíveis os dados dos anos de 2016 a 2023, período em que o SISU contou com duas edições anuais, cada uma composta por conjuntos de dados referentes à chamada regular e à lista de espera. Cada arquivo consiste em uma tabela onde uma linha representa uma inscrição e as colunas contêm informações detalhadas sobre o processo seletivo, a instituição de ensino, o curso e o candidato inscrito⁸.

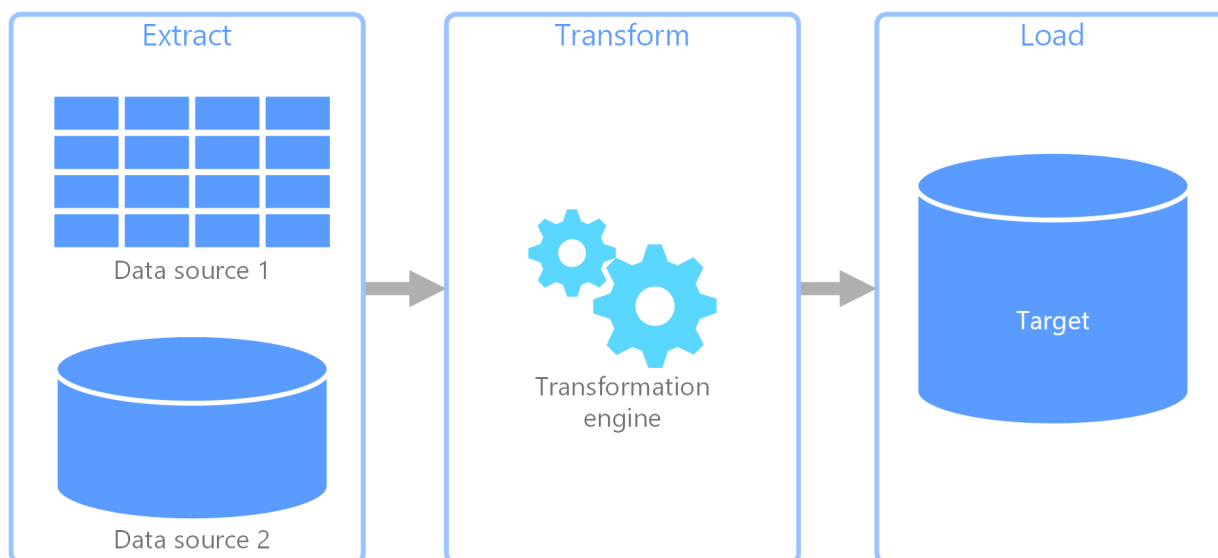
Para o desenvolvimento deste trabalho optou-se por utilizar os dados das chamadas regulares das edições de 2017 a 2023. A exclusão dos dados das listas de espera deve-se à sua menor relevância para os objetivos propostos, uma vez que as chamadas regulares fornecem uma visão mais direta e representativa das tendências de procura pelos cursos superiores. A análise sobre a lista de espera ficou registrada como possível trabalho futuro.

Neste trabalho, o processo de Extração, Transformação e Carga (ETL) dos dados foi utilizado para preparar os dados para visualização. Esse processo é descrito na Figura 1. A etapa de extração envolveu o *download* dos dados brutos em formato CSV do portal de dados abertos do MEC. Em seguida, ocorreu a transformação, com a limpeza, padronização e tratamento de inconsistências para garantir a qualidade dos dados. O processo de transformação foi implementado por meio do desenvolvimento de scripts escritos na linguagem de programação *Python*, com o uso da biblioteca *pandas* para limpeza e transformação dos dados. Por fim, na fase de carga, os dados processados foram inseridos em um banco de dados relacional implementado no Sistema de Gerenciamento de Bancos de Dados (SGBD) *Postgre SQL* 16. Cada uma dessas etapas é detalhada a seguir.

⁷ Disponível em <<https://dadosabertos.mec.gov.br/sisu>>

⁸ A documentação de quais são e o que significam cada coluna e seus tipos de dados está disponível em: <<https://dadosabertos.mec.gov.br/sisu/item/133-dicionario-de-dados>>

Figura 1 - O Processo ETL



Fonte: MICROSOFT (2025).

A etapa de extração consistiu na leitura dos arquivos CSV obtidos no portal. Embora todas as edições descrevam o mesmo fenômeno, os arquivos não foram publicados sob um padrão único: as edições de 2017, 2018, 2022 e 2023 empregam a barra vertical (|) como separador de campos, enquanto as de 2019 a 2021 utilizam o ponto-e-vírgula (;). A codificação de caracteres alterna entre *ISO-8859-1* e *UTF-8*, e o número de colunas varia de 52, em 2017, a 56, nas edições de 2022 e 2023. Para absorver essa variação, a leitura é feita por tentativa em cascata entre dois perfis: separador ponto-e-vírgula com codificação *UTF-8* e, caso a leitura de uma amostra inicial falhe, barra vertical com codificação *ISO-8859-1*. A leitura ocorre em blocos de 100.000 linhas, de modo que arquivos com milhões de registros sejam processados sem esgotar a memória disponível.

Na etapa de transformação, cada bloco de dados passa por uma sequência de operações de limpeza e padronização, devido a problemas presentes nos dados brutos. O primeiro problema é a inconsistência dos nomes de colunas entre as edições. Os arquivos de 2017 e os de 2019 a 2021 nomeiam os campos com marcadores de tipo herdados das planilhas oficiais: *NU_* para valores numéricos, *SG_* para siglas, *ST_* para situações e *NO_* para nomes. Enquanto isso, os arquivos de 2018, 2022 e 2023 já os trazem sem esses marcadores. Um mesmo dado aparece, portanto, ora como *NU_ANO*, ora como *ANO*; ora como *SG_UF_IES*, ora como *UF_IES*. A padronização remove essas sequências (convertendo *NO_* em

NOME_) e, em seguida, aplica um mapa explícito de renomeações para os casos que não seguem regra alguma, a exemplo de:

- *CO_IES_CURSO* passou a ser *CODIGO_CURSO*;
- *DS_GRAU* passou a ser *GRAU*;
- *TP_SEXO* passou a ser *SEXO* ;
- *DT_NASCIMENTO* passou a ser *DATA_NASCIMENTO*

O segundo problema é a presença de colunas sem utilidade analítica ou restritas a apenas algumas edições. As edições de 2022 e 2023 acrescentaram os campos *DS_PERIODICIDADE*, *TP_COTA* e *QT_VAGAS_CONCORRENCIA*, inexistentes nos anos anteriores; a de 2017, por sua vez, não possui o campo *MATRICULA*. Colunas dessa natureza são descartadas e, ao final, todos os blocos são reordenados segundo um formato canônico de 53 campos, o que assegura que qualquer edição, independentemente de seu leiaute original, seja carregada com a mesma estrutura.

O terceiro problema é a divergência de tipos e formatos. As notas do ENEM, os pesos atribuídos pelos cursos a cada área de conhecimento e as médias mínimas exigidas são publicados com a vírgula como separador decimal, conforme a convenção brasileira, e precisam ser convertidos para o ponto decimal esperado pelo banco de dados: sem essa conversão, um valor como *624,58* seria interpretado como texto, e não como número. De modo semelhante, a coluna que registra a classificação do candidato apresenta valores ausentes que impedem sua leitura direta como número inteiro, sendo necessário preencher tais lacunas antes da conversão de tipo.

A esse trabalho de padronização soma-se uma operação de enriquecimento. Os arquivos originais informam apenas a unidade federativa da instituição de ensino, sem qualquer indicação de região geográfica. Como parte dos objetivos deste trabalho envolve a comparação entre as regiões do país, a coluna *REGIAO_IES* é derivada da UF da instituição por meio de um dicionário de correspondência entre as 27 unidades federativas e as cinco regiões (Norte, Nordeste, Centro-Oeste, Sudeste e Sul).

Por fim, na etapa de carga, cada bloco já transformado é serializado em memória e inserido no *PostgreSQL* por meio do comando *COPY ... FROM STDIN*,

forma de inserção em massa substancialmente mais rápida do que a inserção linha a linha e adequada ao volume de milhões de registros da base.

3.5 O ESQUEMA LÓGICO DO BANCO DE DADOS

Esta seção descreve a estrutura dos dados no banco de dados relacional PostgreSQL. Para a implementação desse banco de dados, optou-se por uma abordagem de tabela única desnormalizada, contendo todas as inscrições, otimizada para consultas analíticas. O esquema lógico detalhado desta tabela, incluindo a classificação de campos sensíveis (nunca expostos), é apresentado no Quadro 1.

Alguns dados referentes à nota do inscrito no quadro seguinte possuem siglas que representam as diferentes provas do ENEM: L (Linguagens), CH (Ciências Humanas), CN (Ciências da Natureza), M (Matemática) e R (Redação).

Quadro 1 - Esquema lógico da tabela sisu_inscricoes

DADO	TIPO	DESCRIÇÃO
id	bigint	Chave substituta (surrogate) gerada na carga
ANO	integer	Ano do processo seletivo
EDICAO	integer	Número da edição no ano de referência
ETAPA	integer	Código numérico da etapa (chamada regular / lista de espera)
DS_ETAPA	text	Descrição textual da etapa
CODIGO_IES	integer	Código da IES (cadastro e-MEC)
NOME_IES	text	Nome da IES
SIGLA_IES	text	Sigla da IES
UF_IES	text	UF do campus-sede da IES
CODIGO_CAMPUS	integer	Código do campus

NOME_CAMPUS	text	Nome do campus
REGIAO_IES	varchar(15)	Região do Brasil do campus
UF_CAMPUS	text	UF do campus
MUNICIPIO_CAMPUS	text	Município do campus
CODIGO_CURSO	integer	Código do curso
NOME_CURSO	text	Nome do curso
GRAU	text	Grau do curso
TURNIO	text	Turno do curso
TIPO_MOD_CONCORRENCIA	text	Tipo da modalidade de concorrência
MOD_CONCORRENCIA	text	Descrição da modalidade de concorrência
PESO_L, PESO_CH, PESO_CN, PESO_M, PESO_R	double precision	Pesos por área definidos pelo curso
NOTA_MINIMA_L, NOTA_MINIMA_CH, NOTA_MINIMA_CN, NOTA_MINIMA_M, NOTA_MINIMA_R	double precision	Notas mínimas exigidas por área
MEDIA_MINIMA	double precision	Média mínima exigida pelo curso
CPF	text	(campo sensível) CPF parcial do estudante

INSCRICAO_ENEM	text	(campo sensível) Inscrição ENEM parcial
INSCRITO	text	(campo sensível) Nome do estudante
SEXO	text	(campo sensível) Sexo do estudante
DATA_NASCIMENTO	text	(campo sensível) Data de nascimento (formato variável)
MATRICULA	text	(campo sensível) Situação de matrícula
UF_CANDIDATO	text	UF de residência do estudante
MUNICIPIO_CANDIDATO	text	Município de residência do estudante
OPCAO	integer	1ª ou 2ª opção do estudante
NOTA_L, NOTA_CH, NOTA_CN, NOTA_M, NOTA_R	double precision	Notas brutas por área
NOTA_L_COM_PESO NOTA_CH_COM_PESO, NOTA_CN_COM_PESO, NOTA_M_COM_PESO, NOTA_R_COM_PESO	double precision	Notas com peso aplicado
NOTA_CANDIDATO	double precision	Nota com que o candidato concorreu
NOTA_CORTE	double precision	Nota de corte da modalidade/curso

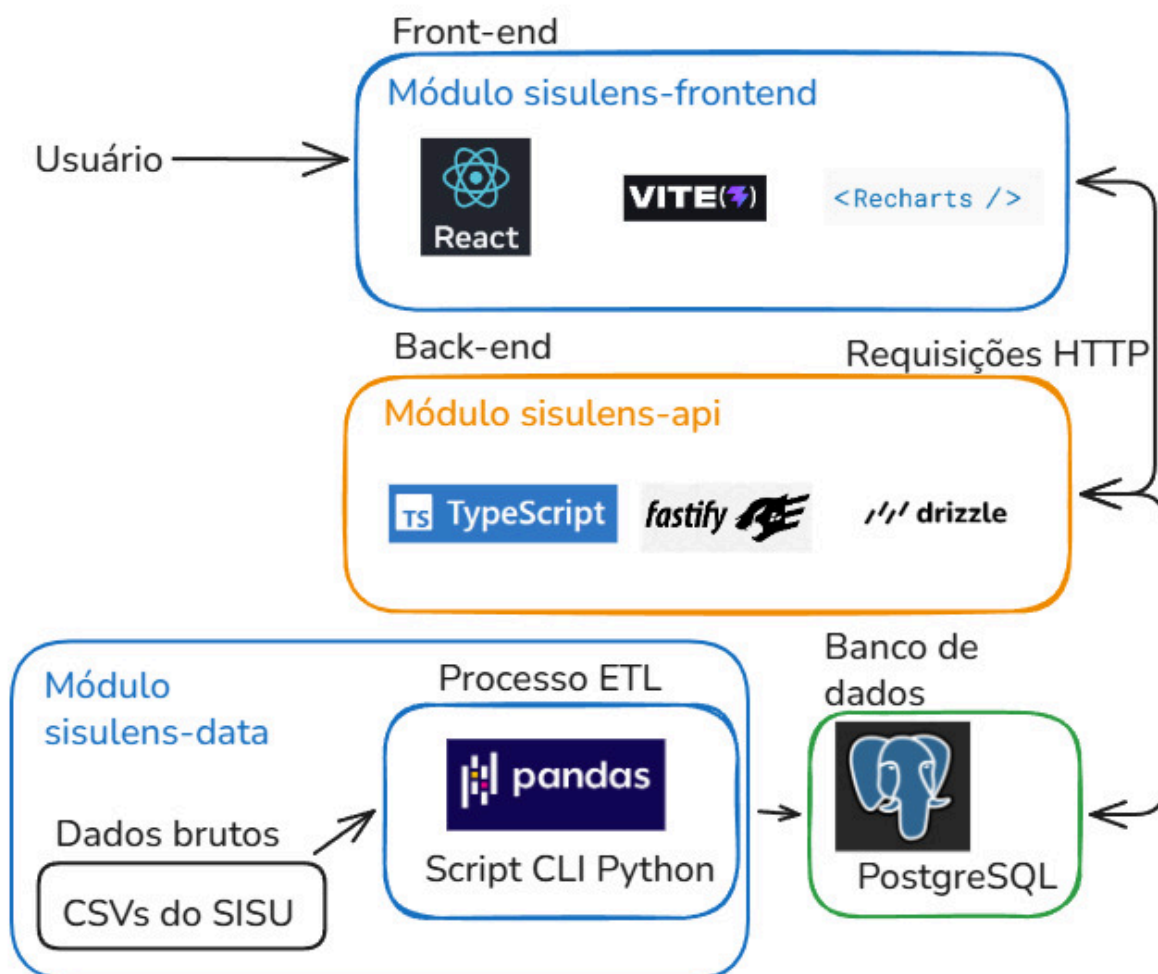
CLASSIFICACAO	integer	Colocação do estudante no curso
APROVADO	text	Se foi aprovado na chamada regular

Fonte: Elaborado pelo autor.

3.6 ARQUITETURA DO SOFTWARE

A arquitetura da ferramenta proposta é baseada em uma abordagem modular, dividida em três camadas principais: *front-end*, *back-end* e camada de dados. Essa arquitetura é mostrada na Figura 2. O *front-end* corresponde à interface gráfica com a qual o usuário irá interagir diretamente. O *back-end* é a camada responsável pelo processamento das informações, gerenciamento da lógica de negócios e comunicação com o banco de dados. O *front-end* e o *back-end* se comunicam por meio de APIs *RESTful*. Por fim, a camada de dados envolve o armazenamento, recuperação e organização dos dados necessários para a aplicação.

Figura 2 - Arquitetura dos módulos da aplicação



Fonte: Elaborado pelo autor

A interface do usuário foi desenvolvida utilizando a tecnologia *React*, devido à sua flexibilidade, desempenho e ampla comunidade de suporte. O gerenciamento de estado foi dividido em dois contextos: para o estado do servidor, foi utilizada a biblioteca *TanStack Query*, que oferece cache automático, controle de *re-fetching* e estados de carregamento; para o estado local da interface, como filtros ativos e preferências de visualização, foi utilizado o *Zustand*, uma biblioteca minimalista com API baseada em *hooks*. Para a visualização de gráficos, foi utilizada a biblioteca *Recharts*, que possibilita a criação de gráficos interativos e *dashboards* dinâmicos, otimizada para uso com *React*.

O *back-end* é o responsável por processar as requisições da interface, realizar consultas complexas e aplicar filtros e agregações dos dados. O seu desenvolvimento se deu com *Node.js* e *TypeScript* utilizando o *framework* *Fastify*. O

banco de dados é acessado por meio do *Drizzle ORM*, um *query builder* com tipagem estática completa em TypeScript e sem *overhead* de tempo de execução. A API conta ainda com os plugins *@fastify/rate-limit*, que limitam o número de requisições por cliente e protegem a ferramenta contra uso abusivo, e *@fastify/cors*, para o controle de origens permitidas.

A camada de dados é composta pelo banco de dados relacional *PostgreSQL*, que foi escolhido devido à sua robustez e ao seu desempenho na resolução de consultas analíticas. O processo de coleta de dados envolve o processo ETL descrito anteriormente. Esse processo se dá a partir de um *script Python* implementado como interface de linha de comando (CLI) que orquestra todo o processo.

Para otimizar o desempenho das consultas sobre o grande volume de registros, foram criadas visões materializadas que pré-computam as agregações utilizadas com maior frequência pela interface, como totais por ano, região, UF e curso. Essas visões são atualizadas uma única vez após a execução do ETL, eliminando a necessidade de realizar varreduras completas na tabela de inscrições a cada requisição da API. Os detalhes de implementação de cada camada são apresentados na seção seguinte.

3.7 IMPLEMENTAÇÃO

Esta seção detalha como a arquitetura descrita na seção anterior foi concretamente implementada. A ferramenta foi organizada em um único repositório composto por três módulos independentes, cada um com seu próprio conjunto de tecnologias e responsabilidades.

3.7.1 IMPLEMENTAÇÃO DA CAMADA DE DADOS

A camada de dados foi implementada no módulo *sisulens-data*, escrito integralmente em *Python 3* com o uso das bibliotecas *pandas* para o processamento tabular, *psycopg2* para a comunicação com o *PostgreSQL* e *SQLAlchemy* como camada complementar de acesso ao banco.

O ponto de entrada do processo é a interface de linha de comando (CLI) escrita em *Python*, que aceita os argumentos *--ano* (um ou mais anos a processar), *--etapa* (com valor padrão *chamada_regular*), *--semestre* e *--all* (para processar todos os anos suportados de uma só vez). A orquestração segue estritamente o

fluxo Extração, Transformação e Carga, processando os dados em blocos (*chunks*) para viabilizar o tratamento de arquivos volumosos sem esgotar a memória.

As três etapas do processo ETL, cujas operações de leitura, limpeza, padronização e enriquecimento, que foram descritas na seção 3.4, foram implementadas em *scripts* distintos: a extração foi implementada em um script *pipeline/extract.py*, que consome os arquivos por meio do parâmetro *chunksize* da função *pandas.read_csv* e devolve um gerador de blocos; a transformação foi implementada no script *pipeline/transform.py*, cuja função *transform_chunk* encadeia as rotinas de renomeação (apoiada no mapa *IND_COLUMNS*), descarte de colunas (*COLUMNS_TO_DROP*), conversão de tipos, atribuição de região geográfica (*REGION_MAPPING*) e reordenação para o formato canônico (*COLUMNS_REORDER*); e a carga em *pipeline/load.py*, que serializa cada bloco em um *buffer* de memória (*StringIO*) delimitado por barra vertical e o insere no banco com *copy_expert*, do *psycopg2*. Os parâmetros que governam o processo são: anos suportados, mapeamentos de colunas e regiões e o DDL da tabela, estão centralizados em um arquivo de configuração *pipeline/config.py*.

Ao final do processo de carga, todas as informações são armazenadas em uma única tabela desnormalizada chamada *sisu_inscricoes*. A decisão pela desnormalização, em vez de um modelo relacional normalizado com múltiplas tabelas e junções, deve-se à natureza analítica e não transacional do sistema: a base é estática após o ETL e é destinada apenas à leitura, o que torna as junções desnecessárias e privilegia a velocidade de consulta. Caso haja alterações nos dados oficiais do SISU, todo o processo de carga e processamento deverá ser refeito, dado que o trabalho já identificou a existência de variações na estrutura dos dados originais ao longo dos anos.

Para acelerar as consultas mais frequentes da interface, foram criadas oito visões materializadas (*materialized views*) do *PostgreSQL*. Uma visão materializada armazena fisicamente o resultado de uma consulta de agregação, permitindo que a API leia dados pré-computados sem varrer a tabela de fatos completa a cada requisição. Todas as visões compartilham uma característica fundamental: filtram exclusivamente as inscrições de primeira opção de curso (*OPCAO = 1*), evitando a dupla contagem de um mesmo candidato que pode aparecer em duas opções de curso.

As visões contam inscrições por meio de *COUNT(*)*, e não candidatos distintos. A contagem de pessoas distintas foi deliberadamente descartada porque nenhuma das colunas capazes de identificá-las se presta a esse fim nos microdados publicados: *INSCRICAO_ENEM* é divulgada mascarada, com quantidade variável de dígitos visíveis entre as edições. A edição de 2018, por exemplo, apresenta apenas 123 valores distintos em milhões de registros, e *DATA_NASCIMENTO* altera a granularidade ao longo do período, sendo publicada como DD/MM/AA em 2017 e 2018, DD/MM/AAAA entre 2019 e 2022 e apenas o ano em 2023. Aplicar *COUNT(DISTINCT)* sobre esses campos produziria subcontagem severa e não comparável entre anos.

Como o recorte *OPCAO = 1* já corresponde, na prática, a aproximadamente uma linha por candidato em cada curso e campus, a contagem de inscrições é a medida consistente ao longo dos sete anos analisados. É nesse sentido que o termo "inscrições" deve ser lido em todos os gráficos da ferramenta.

O Quadro 2 apresenta as visões e suas agregações.

Quadro 2 - Views materializadas do banco de dados

VIEW	AGREGAÇÃO	AGRUPAMENTO (GROUP BY)	REQUISITO
<i>mv_inscritos_por_ano</i>	<i>COUNT(*)</i>	ano	RF01
<i>mv_inscricoes_por_uf_campus</i>	<i>COUNT(*)</i>	UF do campus	RF01
<i>mv_inscricoes_por_uf_candidato</i>	<i>COUNT(*)</i>	UF de origem do candidato	RF01
<i>mv_inscricoes_por_curso_ano</i>	<i>COUNT(*)</i>	ano, código e nome do curso	RF01
<i>mv_cursos_por_campus_ano</i>	<i>COUNT(*)</i>	ano, curso, grau, turno, campus, IES, região, UF, município	RF02

<i>mv_curso_campus_historico</i>	<i>COUNT(*), MAX(NOTA_CORTE) , AVG(NOTA_CANDID ATO)</i>	curso, campus, IES, ano, edição, modalidade	RF03
<i>mv_curso_municipio_candidato</i>	<i>COUNT(*)</i>	curso, campus, município e UF de origem	RF03 / RF06
<i>mv_media_notas</i>	<i>AVG(NOTA_CANDID ATO)</i>	ano, região, UF do campus, nome do curso	RF05

Fonte: Elaborado pelo autor.

Duas particularidades das visões merecem destaque. A primeira é a conformidade com a LGPD: a *view mv_curso_municipio_candidato*, que descreve a distribuição de origem dos candidatos no nível mais granular (município), aplica a cláusula *HAVING COUNT(*) >= 5*, suprimindo grupos com menos de cinco candidatos para impedir a inferência de dados individuais. A segunda é o filtro de qualidade dos dados de desempenho: as *views mv_curso_campus_historico* e *mv_media_notas*, que calculam médias de notas, exigem adicionalmente *NOTA_CANDIDATO > 0*, descartando registros sem nota válida. O Código-fonte 1 exemplifica a definição da visão de distribuição por município, evidenciando ambos os aspectos.

Código fonte 1 - Definição da view materializada mv_curso_municipio_candidato

SQL

```
CREATE MATERIALIZED VIEW IF NOT EXISTS
mv_curso_municipio_candidato AS
SELECT
```

```

"CODIGO_CURSO" AS codigo_curso,
"CODIGO_CAMPUS" AS codigo_campus,
"MUNICIPIO_CANDIDATO" AS municipio_candidato,
"UF_CANDIDATO" AS uf_candidato,
COUNT(*) AS total
FROM sisu_inscricoes
WHERE "OPCAO" = 1
AND "MUNICIPIO_CANDIDATO" IS NOT NULL
AND "UF_CANDIDATO" IS NOT NULL
GROUP BY "CODIGO_CURSO", "CODIGO_CAMPUS",
"MUNICIPIO_CANDIDATO", "UF_CANDIDATO"
HAVING COUNT(*) >= 5;

```

Fonte: Elaborado pelo autor.

Além das visões, também foram criados 8 índices do tipo B-tree sobre as colunas mais utilizadas nos filtros das consultas (como ano, UF, grau e códigos de curso e campus), reduzindo o tempo de resposta das requisições parametrizadas.

Como alternativa de otimização, uma camada de cache distribuída poderia ter sido empregada para armazenar os resultados das agregações mais frequentes. No entanto, sua implementação foi descartada em favor das visões materializadas, uma vez que o desempenho se mostrou satisfatório com uma arquitetura mais simples e sem o custo operacional de gerenciar uma infraestrutura adicional.

3.7.2 IMPLEMENTAÇÃO DO BACK-END

A API foi implementada no módulo *sisulens-api* utilizando o framework *Fastify* 5.6.1 sobre *Node.js*, com *TypeScript* 5.9 e o *Drizzle* ORM 0.45.1 para o acesso tipado ao banco de dados. O projeto adota o padrão de módulos *ECMAScript* (ESM) e o servidor é exposto na porta 8082.

O ponto de entrada configura os aspectos transversais da aplicação. O controle de origens é feito pelo plugin *@fastify/cors*, restringindo as requisições à origem do *front-end* definida em variável de ambiente. A proteção contra uso abusivo

é garantida pelo plugin `@fastify/rate-limit`, configurado para aceitar no máximo 100 requisições por minuto por cliente. Um manipulador de erros global preserva as respostas de erro de validação (código 4xx) e mascara falhas internas (código 5xx) com uma mensagem genérica, evitando o vazamento de detalhes de implementação. Por fim, um hook `onSend` intercepta todas as respostas bem-sucedidas e injeta o cabeçalho de cache HTTP, conforme o Código-fonte 2.

Código fonte 2 - Hook de Cache-Control aplicado a todas as respostas

TypeScript

```
fastify.addHook('onSend', async (_request, reply) => {  
  if (reply.statusCode < 400) {  
    reply.header('Cache-Control', 'public,  
max-age=86400')  
  }  
})
```

Fonte: Elaborado pelo autor.

Como os dados são históricos e imutáveis após o ETL, o cabeçalho *Cache-Control: public, max-age=86400* autoriza o cache das respostas por 24 horas, reduzindo a carga sobre o banco de dados ao evitar requisições repetidas. A conexão com o PostgreSQL é gerenciada em *src/db/client.ts* por um pool limitado a 5 conexões simultâneas, o que previne o esgotamento de conexões do banco durante rajadas de requisições do painel, com um *statement_timeout* de 25 segundos que cancela consultas excessivamente longas.

O esquema do banco é declarado de forma tipada, que mapeia a tabela *sisu_inscricoes* e duas das oito visões materializadas (*mv_cursos_por_campus_ano* e *mv_media_notas*). As demais visões são acessadas por consultas parametrizadas. Esse arquivo também documenta explicitamente o bloco de colunas de Informação Pessoal Identificável: *CPF*, *INSCRICAO_ENEM*, *INSCRITO*, *DATA_NASCIMENTO* e *MATRICULA*, que jamais devem ser selecionadas nas rotas, em conformidade com o requisito não funcional de privacidade (RNF02).

A API expõe seis *endpoints* de produção e um *endpoint* de depuração disponível apenas fora do ambiente de produção. O Quadro 3 apresenta o registro completo das rotas, associando cada uma às visões ou tabelas que consulta e ao requisito funcional que atende.

Quadro 3 - Registro de endpoints da API

MÉTODO E ROTA	VIEW / TABELA CONSULTADA	REQUISITO
GET /overview	<i>mv_inscritos_por_ano,</i> <i>mv_inscricoes_por_uf_campus,</i> <i>mv_inscricoes_por_curso_ano,</i> <i>mv_inscricoes_por_uf_candidato</i>	RF01
GET /cursos/tipos	<i>mv_cursos_por_campus_ano</i>	RF02
GET /cursos	<i>mv_cursos_por_campus_ano</i>	RF02
GET /cursos/:codigo_curso/campus/:codigo_campus	<i>mv_curso_campus_historico,</i> <i>mv_curso_municipio_candidato</i>	RF03
GET /media-notas	<i>mv_media_notas</i>	RF05
GET /cursos/:codigo_curso/campus/:codigo_campus/perfil	<i>sisu_inscricoes (tabela base)</i>	RF06
GET /debug/schema	<i>metadados</i> <i>(information_schema)</i>	Sem requisito associado

Fonte: Elaborado pelo autor.

A implementação adota dois padrões de consulta complementares. As rotas que operam sobre objetos mapeados no *schema* do *Drizzle* utilizam o *query builder* tipado (*/cursos*, */cursos/tipos*, */media-notas* e */perfil*), ao passo que as rotas que agregam visões não mapeadas empregam *templates* SQL parametrizados por meio de *db.execute(sql...)* (*/overview* e o histórico de curso). Em ambos os casos, não foi feito uso de *sql.raw()* e de qualquer interpolação de *strings* na construção das consultas: todo valor de origem externa é vinculado como *placeholder*, o que elimina a superfície de injeção de SQL nas consultas implementadas (RNF03).

Adicionalmente, as rotas que recebem parâmetros de entrada os validam por meio de *schemas* JSON processados pelo validador do próprio *Fastify*, que conferem tipo e faixa de valores. Os termos de busca textual, por sua vez, são repassados a expressões *ILIKE*, de modo que os operadores “%” e “_” eventualmente digitados pelo usuário são interpretados como parte do padrão de busca, comportamento sem implicação de segurança, mas que altera a semântica da consulta.

O endpoint */perfil* merece destaque por ser o único que consulta diretamente a tabela base *sisu_inscricoes*, aplicando explicitamente o filtro *OPCAO = 1* no código da rota. Ele computa, para um curso e campus específicos, a distribuição dos candidatos por sexo, por município de origem e por faixas de nota (histograma em intervalos de 50 pontos). Para atender à LGPD, qualquer agrupamento com menos de cinco registros tem a sua contagem anulada e marcada com o campo *suppressed*, exibindo < 5 na interface. Este limiar é utilizado para evitar a identificação de indivíduos, conforme boas práticas de anonimização de dados. O número 5 foi escolhido arbitrariamente.

3.7.3 IMPLEMENTAÇÃO DO FRONT-END

A interface foi implementada no módulo *sisulens-frontend* como uma *Single Page Application* construída com *React* 19.1 e empacotada pela ferramenta *Vite* 7.1. A estilização utiliza o *Tailwind CSS* 4.1 em conjunto com componentes derivados do catálogo *shadcn/ui*, construídos sobre a biblioteca de primitivos acessíveis *Radix UI*, esta sim uma dependência de execução, e ícones da *Lucide*. O roteamento é feito

com o *React Router* 7.13, o estado de servidor com o *TanStack Query* 5.95, o estado de interface com o *Zustand* 5, e os gráficos com o *Recharts* 2.15.

A aplicação carrega cada tela sob demanda no momento em que o usuário navega até ela, o que reduz o volume transferido no primeiro acesso. Todas as telas são renderizadas dentro de uma casca comum, responsável pela barra de navegação e pelo rodapé com a atribuição obrigatória da fonte dos dados, presente em todas as páginas. Um limite de erro envolve a aplicação como um todo: falhas inesperadas de renderização são contidas e apresentadas como mensagem de erro, sem derrubar a interface. O Quadro 4 lista as rotas da aplicação e as páginas correspondentes.

Quadro 4 - Rotas e páginas da SPA

ROTA	PÁGINA	PROPÓSITO
/	Visão Geral	Panorama nacional das inscrições
/cursos	Busca de Cursos	Busca de curso agrupados por grau
/cursos/tipo/:grau/:nomeCurso	Lista de Ofertas	Ofertas concretas (IES/campus) de um curso
/cursos/:codigoCurso /campus/:codigoCampus	Detalhe do Curso	Histórico, notas e perfil dos candidatos de uma oferta
/media-notas	Desempenho por UF	Comparação da nota média entre estados
/comparar	Comparação de Cursos	Comparação lado a lado de dois cursos
*	Página 404	Rota de fallback

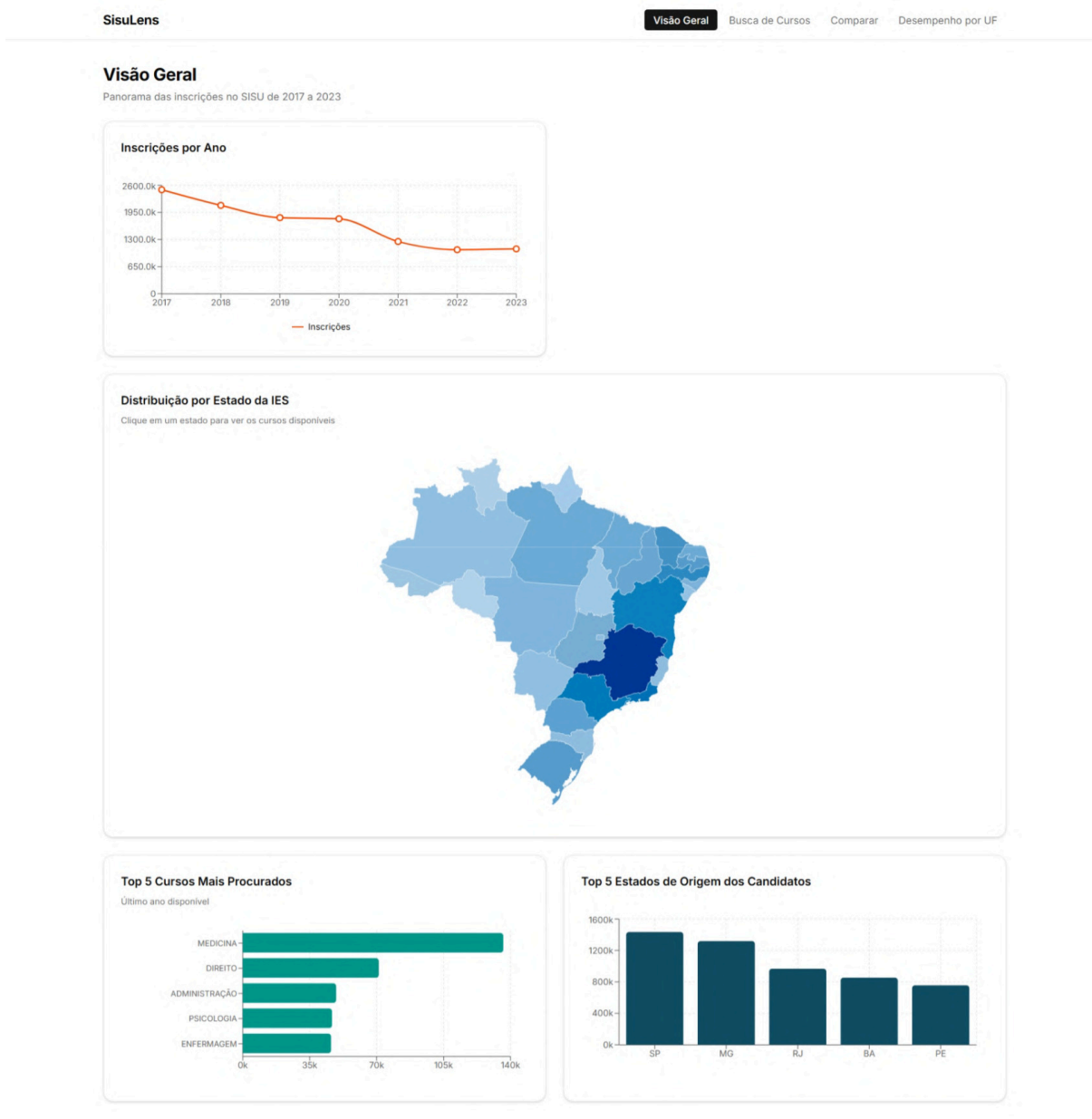
Fonte: Elaborado pelo autor.

A barra de navegação expõe apenas quatro links principais (Visão Geral, Busca de Cursos, Comparar e Desempenho por UF); as duas rotas intermediárias são acessadas por navegação em profundidade (*drill-down*), configurando um fluxo de busca em duas etapas: o usuário localiza um tipo de curso, refina para uma oferta específica e, então, acessa o detalhe. As telas implementadas são descritas a seguir.

A página de Visão Geral (Figura 3) apresenta um painel com quatro visualizações, cujos recortes temporais convém distinguir: um gráfico de linha com a evolução anual das inscrições ao longo de todo o período; um gráfico de barras com os cinco cursos mais procurados no último ano disponível; e outro com os cinco estados de origem que mais concentraram candidatos, também no acumulado do período; e um mapa coroplético do Brasil com a distribuição por UF da instituição, referente ao acumulado de 2017 a 2023.

O mapa é interativo: ao clicar em um estado, o usuário é direcionado à busca de cursos já filtrada por aquela UF. Além disso, ao posicionar o cursor sobre um estado, exibe-se um cartão flutuante com o nome do estado por extenso, o número total de inscrições, o número de instituições e o de cursos ofertados naquela UF, acompanhados da indicação de que o clique conduz à busca de cursos.

Figura 3 - Página de Visão Geral



Fonte: Elaborado pelo autor.

A busca de cursos foi dividida em duas etapas. A primeira etapa é a página de Busca de Cursos (Figura 4), que lista denominações de curso agregadas em âmbito nacional. Nela, cada resultado corresponde a um par denominação-grau (por exemplo, "Medicina", grau bacharelado) e apresenta dados consolidados sobre todas as instituições, o número de ofertas e o total de inscrições. A tela dispõe de um campo de busca textual com *debounce* de 300 milissegundos. A consulta é

disparada apenas após uma pausa na digitação, evitando uma requisição por tecla pressionada. Há filtros por grau, UF da instituição e ano de referência, e paginação com tamanho de página ajustável.

Figura 4 - Página de Busca de Cursos com ofertas agregadas

Fonte: Elaborado pelo autor.

A seleção de uma denominação conduz à segunda etapa, a página de Lista de Ofertas (Figura 5). Nessa página são listados todos os cursos da denominação escolhida anteriormente. Cada item identifica a IES, o campus (município e UF) e o turno, acompanhados do total de inscrições daquela oferta. Os filtros disponíveis são: sigla de IES, turno, região, UF da instituição e ano de referência. A escolha de uma oferta conduz à página de detalhe do curso.

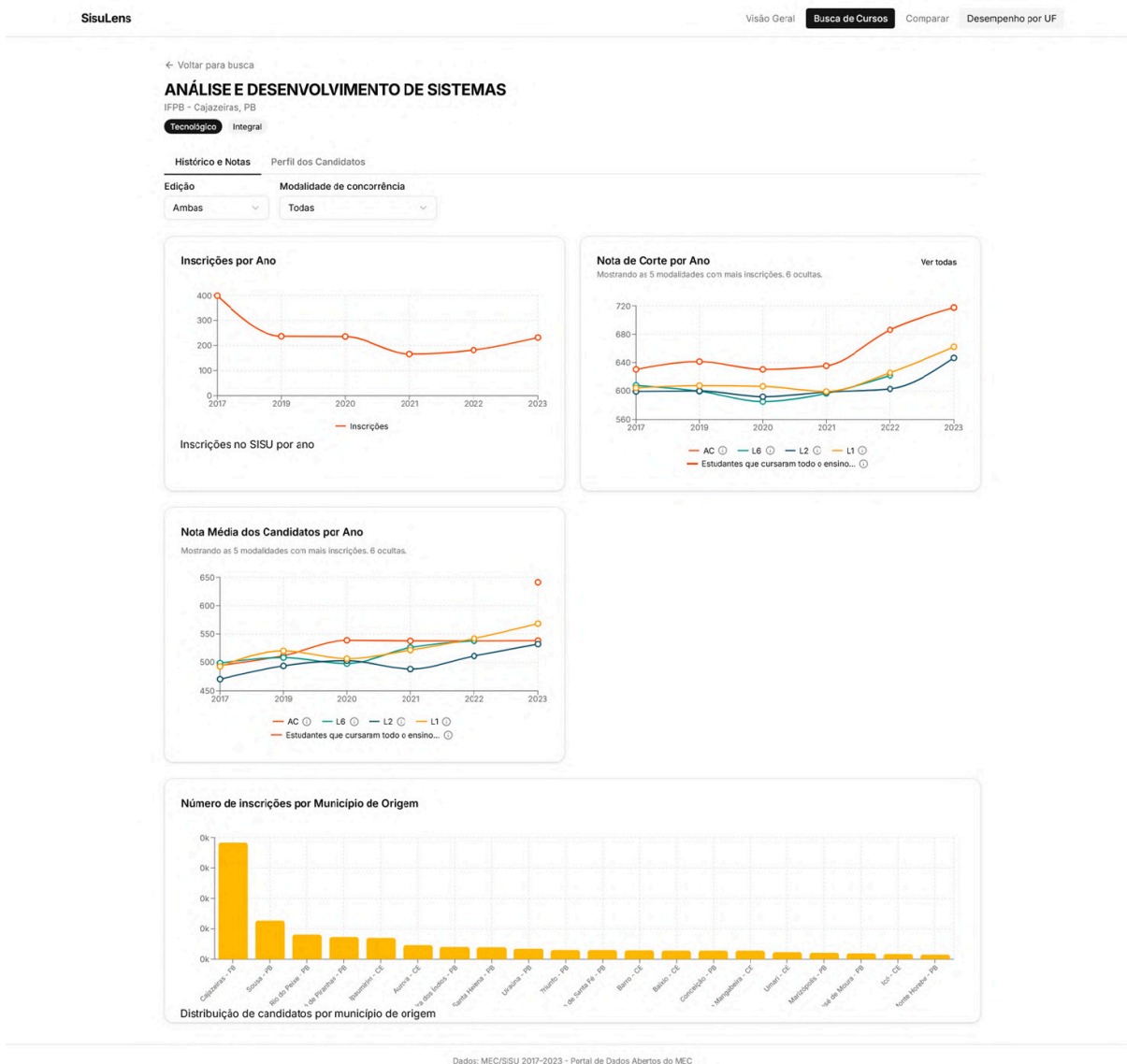
Figura 5 - Página de Lista de Ofertas de uma denominação de curso

Fonte: Elaborado pelo autor.

A página de Detalhe do Curso (Figuras 6 e 7) organiza a análise de uma oferta específica em duas abas. A aba *Histórico e Notas* exibe a evolução das inscrições por ano, a nota de corte e a nota média por modalidade de concorrência, e a distribuição de candidatos por município de origem. A aba *Perfil dos Candidatos* apresenta, para um ano selecionável pelo usuário, a distribuição por sexo (em gráfico de rosca), a distribuição por município de origem e o histograma de notas em faixas de 50 pontos. Todas as visualizações de perfil respeitam a supressão de grupos com menos de cinco candidatos, exibindo “< 5” nesses casos.

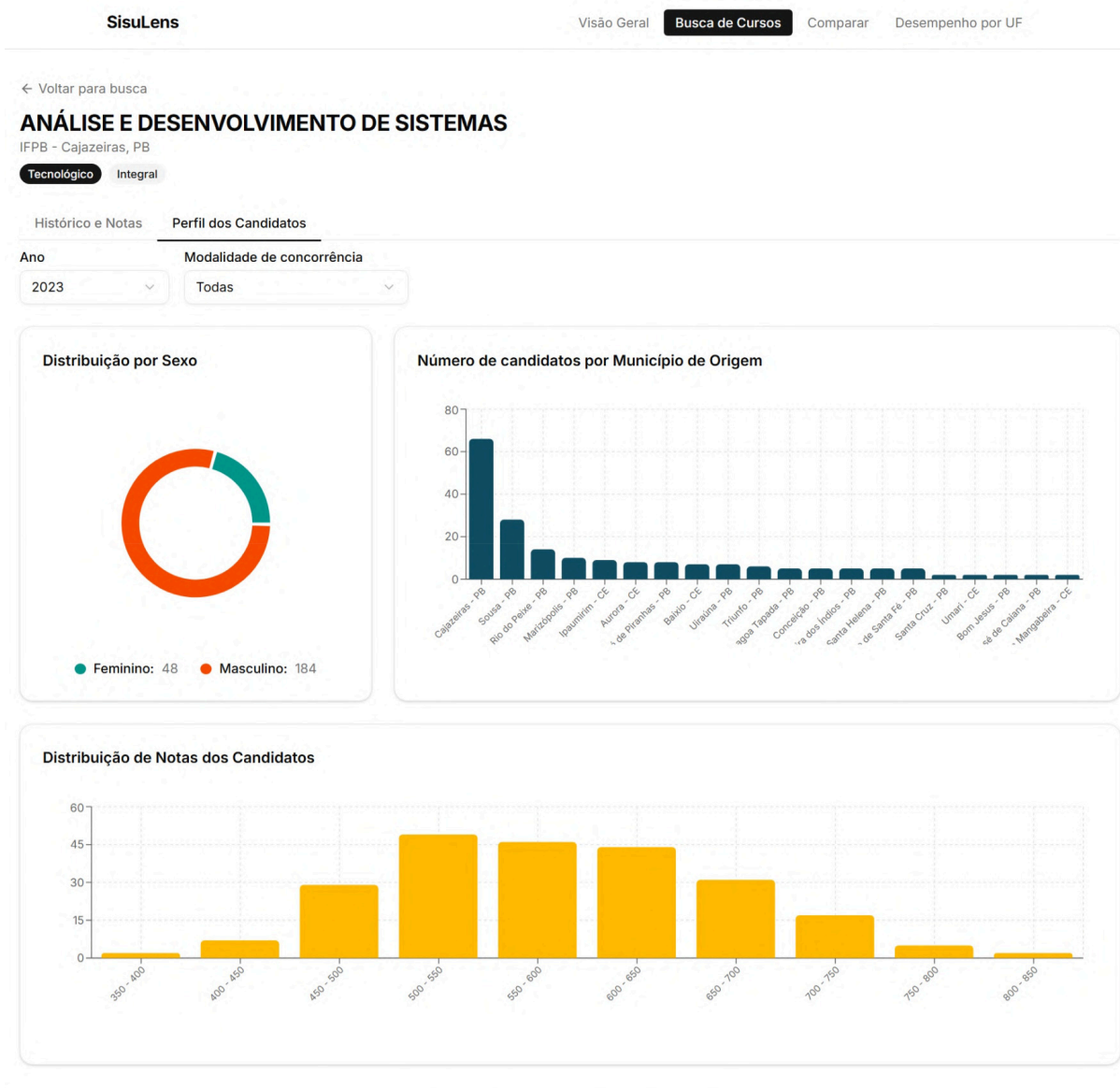
Cabe observar que tanto as séries da aba histórica quanto as agregações da aba de perfil consideram exclusivamente as inscrições de primeira opção com nota válida registrada, recorte que deve ser levado em conta na leitura da curva de inscrições, cujos totais são inferiores aos apresentados no painel de visão geral. Para categorias de alta cardinalidade, como as modalidades de concorrência, a interface mostra por padrão as cinco mais relevantes, com a opção de exibir todas.

Figura 6 - Página de Detalhe do Curso (aba Histórico e Notas)



Fonte: Elaborado pelo autor.

Figura 7 - Página de Detalhe do Curso (aba Perfil dos Candidatos)



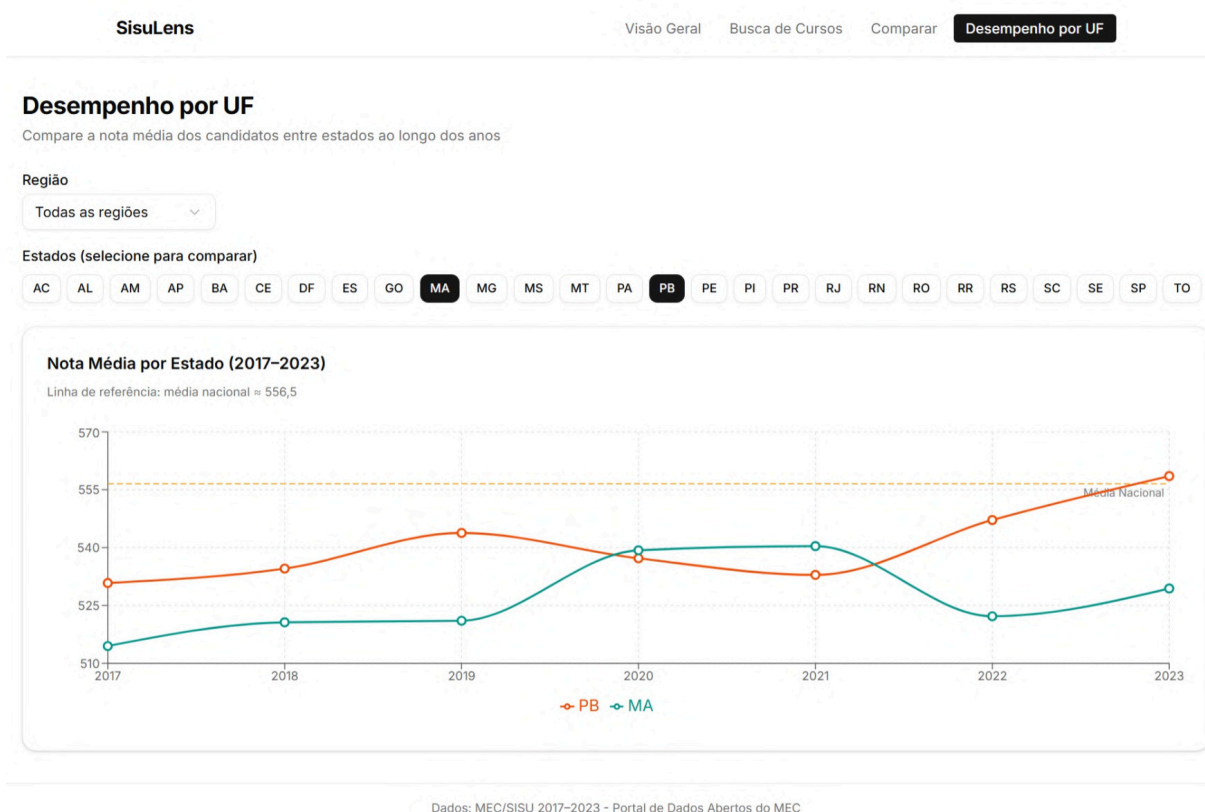
Fonte: Elaborado pelo autor.

A página de Desempenho por UF (Figura 8) permite comparar o desempenho entre estados ao longo do período, exibindo um gráfico multilinha com uma série por UF selecionada, acompanhada de uma linha de referência horizontal correspondente à média do recorte selecionado no período. O usuário pode selecionar múltiplos estados simultaneamente e restringir a análise a uma região.

O indicador apresentado é a média das médias por curso, não ponderada pelo número de candidatos: a visão materializada armazena uma média de nota por ano, região, UF do campus e nome do curso, e a API agrega essas linhas por média

aritmética simples, o que atribui a cada curso o mesmo peso, independentemente do volume de inscrições. O valor deve, portanto, ser interpretado como o desempenho médio da oferta de cursos do estado, e não como a média de desempenho dos seus candidatos, indicador cujo cálculo exigiria ponderação pelo número de inscritos. A linha de referência representa a média do período inteiro, e não uma série anual; quando um filtro de região está ativo, ela corresponde à média daquela região.

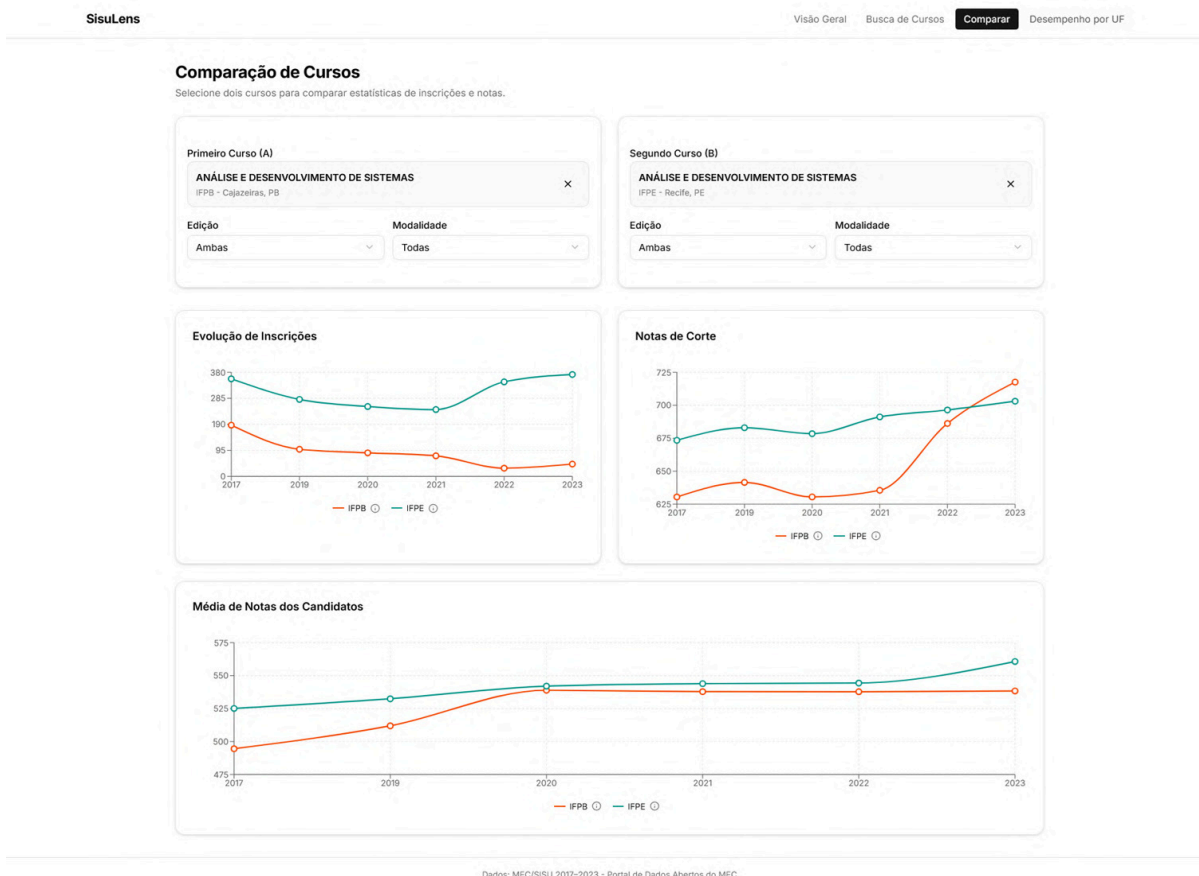
Figura 8 - Página de Desempenho por UF



Fonte: Elaborado pelo autor.

A página de Comparação de Cursos (Figura 9) posiciona dois cursos lado a lado, sobrepondo em um mesmo eixo as séries de inscrições, nota de corte e nota média. O estado completo da comparação é serializado na URL, o que torna qualquer comparação compartilhável por meio de endereço.

Figura 9 - Página de Comparação de Cursos



Fonte: Elaborado pelo autor.

O gerenciamento de estado segue uma segregação rígida entre estado de servidor e estado de interface. Todas as chamadas HTTP passam por um único *wrapper* central, a função *apiFetch()*, que define a URL base a partir de variável de ambiente e impõe um *timeout* de 10 segundos por meio de *AbortController*, falhando rapidamente em caso de indisponibilidade da API. Sobre esse *wrapper*, o *TanStack Query* administra o *cache* em memória das respostas (com *staleTime* de 5 minutos e até duas tentativas), enquanto o *Zustand* mantém apenas os filtros da interface. O estado de paginação e o estado da tela de comparação residem na *query string*, garantindo que essas visões sejam recarregáveis e compartilháveis.

A acessibilidade também foi levada em consideração. Assim, todos os gráficos e o mapa são pareados com um componente *ChartAccessibleTable*, que renderiza uma tabela de dados equivalente, visível apenas a leitores de tela, garantindo que nenhuma informação exista unicamente sob forma gráfica.

O sistema adota uma cromia monocromática para toda a interface, reservando a cor exclusivamente para os dados, por meio de uma paleta fixa de cinco cores. São vedados gradientes, emojis, efeitos de desfoque e imagens de destaque; os números são formatados no padrão brasileiro; as animações limitam-se, a transições de cor e opacidade de curta duração (cerca de 160 milissegundos); e os estados de carregamento utilizam *skeletons* em vez de indicadores giratórios.

4. CONSIDERAÇÕES FINAIS

O Portal Único de Acesso ao Ensino Superior do MEC disponibiliza uma ferramenta interativa baseada em *Power BI* que permite filtrar e visualizar informações relativas aos programas SISU, FIES e ProUni no período de 2020 a 2023. Conforme discutido no capítulo de introdução, essa solução cobre um recorte temporal mais restrito do que o adotado neste trabalho e não contempla a análise da evolução do número de inscrições ao longo da série histórica, a comparação entre cursos, a análise do desempenho dos candidatos por unidade federativa nem a caracterização do perfil dos inscritos em uma oferta específica. Esse contexto justifica uma abordagem complementar, voltada à exploração interativa e mais aprofundada dos microdados, capaz de apoiar a identificação de tendências e desafios do acesso ao ensino superior.

Para solucionar esse problema, este trabalho propôs o *Sisu Lens*, uma ferramenta de visualização de dados que permite a exploração interativa das informações das inscrições do SISU entre os anos de 2017 e 2023. Utilizando um processo ETL baseado em *Python* e *PostgreSQL*, os dados são extraídos, transformados e organizados em agregações pré-computadas, apresentadas por meio de gráficos interativos, e listas filtráveis em uma interface construída com *React*. A solução torna acessíveis recortes dos microdados que antes exigiam o processamento local de arquivos com milhões de registros.

Na primeira parte do desenvolvimento, foram realizadas as atividades fundamentais de planejamento e fundamentação do projeto: o levantamento de requisitos, a revisão da literatura pertinente e a definição da arquitetura de *software*, além da seleção das tecnologias. Adicionalmente, iniciou-se o desenvolvimento dos *scripts* em *Python*, possibilitando a exploração inicial dos dados coletados.

Na segunda parte, os três módulos da ferramenta foram implementados e integrados: a camada de dados (ETL em *Python/PostgreSQL* com uma tabela desnormalizada e oito *views* materializadas), a API (em *Fastify/Drizzle ORM*, somente leitura e com atenção à privacidade e ao desempenho) e a interface (SPA em *React*, com as telas de visão geral, busca de cursos, detalhe de curso, comparação e desempenho por UF). Com isso, considera-se atendido o objetivo geral do trabalho, bem como os objetivos específicos relativos à identificação de

tendências de procura por cursos, à disponibilização de dados sobre demanda e desempenho no ENEM e à caracterização da distribuição geográfica dos candidatos.

Cabe explicitar as limitações desta versão da ferramenta. No que se refere aos dados, foram processadas exclusivamente as chamadas regulares do SISU entre 2017 e 2023, de modo que as demais etapas do processo seletivo não estão representadas. As agregações contabilizam inscrições de primeira opção, e não candidatos distintos, uma vez que os identificadores publicados nos microdados não permitem a contagem confiável de pessoas, conforme discutido na seção 3.7.1; e as séries históricas por curso, bem como o perfil dos candidatos, consideram apenas os registros com nota válida, o que as torna não somáveis aos totais do painel de visão geral.

Como trabalhos futuros, identificam-se, em primeiro lugar, as funcionalidades cujo escopo não foi contemplado nesta versão: a análise histórica dedicada por instituição de ensino; o enriquecimento da análise geográfica com uma página dedicada, dotada de alternância entre as visões de IES e de candidato e de filtros por região; e a análise do fluxo geográfico origem x destino dos candidatos, que exigiria uma nova agregação cruzando a UF de residência com a UF da IES. Soma-se a elas a ampliação da análise de perfil com atributos socioeconômicos (renda, raça/cor e escolaridade), condicionada à disponibilização de dados não presentes na base atual.

Adicionalmente, aponta-se como trabalho futuro a consideração dos *datasets* referentes às listas de espera do SISU. Embora tais dados tenham sido excluídos do escopo deste trabalho, sua inclusão poderia ampliar o escopo da ferramenta ao contemplar a dinâmica integral do processo seletivo.

Decorrem das limitações apontadas os seguintes encaminhamentos técnicos: a ponderação da média de notas pelo número de inscritos, que tornaria o indicador de desempenho por UF diretamente interpretável como média dos candidatos; a separação, na visão histórica, entre a contagem de inscrições e o cálculo das médias; a adoção de índices de trigramas (*pg_trgm*) para acelerar a busca textual por nome de curso; a criação de índices únicos nas visões materializadas, requisito para a atualização sem bloqueio de leitura; a avaliação sistemática de desempenho e a auditoria de acessibilidade; a implementação de testes automatizados e de integração contínua; e a implantação pública da ferramenta, condição para que as informações efetivamente alcancem o público a que se destinam.

REFERÊNCIAS

ARIOVALDO, T. C. de C.; NOGUEIRA, C. M. M. **O Sisu e a escolha pelas licenciaturas da Universidade Federal de Viçosa**. Estudos em Avaliação Educacional, São Paulo, v. 32, p. e06763, 2021. DOI: 10.18222/ea.v32.6763. Disponível em: <<https://publicacoes.fcc.org.br/ea/article/view/6763>>. Acesso em: 10 fev. 2025.

BRASIL. **Lei nº 12.527, de 18 de novembro de 2011**. Regula o acesso a informações previsto no inciso XXXIII do art. 5º, no inciso II do § 3º do art. 37 e no § 2º do art. 216 da Constituição Federal; altera a Lei no 8.112, de 11 de dezembro de 1990; revoga a Lei no 11.111, de 5 de maio de 2005, e dispositivos da Lei no 8.159, de 8 de janeiro de 1991; e dá outras providências. Diário Oficial [da] República Federativa do Brasil, Brasília, DF, 18 dez. 2011c. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm>. Acesso em: 25 ago. 2024.

DRIZZLE TEAM. **Drizzle ORM Documentation**. Disponível em: <<https://orm.drizzle.team/docs/overview>>. Acesso em: 23 mar. 2026.

FASTIFY. **Fastify Documentation**. Disponível em: <<https://fastify.dev/docs/latest/>>. Acesso em: 19 out. 2025.

IBM. **O que é uma API REST?**. Disponível em: <<https://www.ibm.com/br-pt/topics/rest-apis>>. Acesso em: 4 mar. 2025.

MICROSOFT. **Extract, transform, and load (ETL). Microsoft Learn**. Disponível em: <<https://learn.microsoft.com/en-us/azure/architecture/data-guide/relational-data/etl>>. Acesso em: 10 fev. 2025.

OPENJS FOUNDATION. **Node.js Documentation**. Disponível em: <<https://nodejs.org/en/docs/>>. Acesso em: 4 mar. 2025.

OPEN KNOWLEDGE FOUNDATION. **Manual Dos Dados Abertos: Governo**. [S.l.], 2011. Disponível em: <<https://acervo.ceweb.br/acervos/conteudo/bd2fe4f3-3230-43f1-bd50-5b029f61ae16>>. Acesso em: 06 set. 2024.

PANDAS DEVELOPMENT TEAM. **pandas documentation**. Disponível em: <<https://pandas.pydata.org/docs/>>. Acesso em: 4 mar. 2025.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL Documentation**. Disponível em: <<https://www.postgresql.org/docs/>>. Acesso em: 15 jul. 2026.

REACT. **React Reference Overview**. Disponível em: <<https://react.dev/reference/react>>. Acesso em: 04 mar. 2025.

REACT ROUTER. **React Router Documentation**. Disponível em: <<https://reactrouter.com/en/main>>. Acesso em: 15 jul. 2026.

RECHARTS. **Recharts Documentation**. Disponível em: <<https://recharts.org/en-US/api>>. Acesso em: 4 mar. 2025.

SHADCN. **shadcn/ui Documentation**. Disponível em: <<https://ui.shadcn.com/>>. Acesso em: 15 jul. 2026.

SILVA, Fabiano Couto Corrêa da. **Visualização de dados abertos no setor público**. Revista Ibero-Americana de Ciência da Informação, [S. l.], v. 11, n. 2, p. 523–540, 2018. DOI: 10.26512/rici.v11.n2.2018.8341. Disponível em: <<https://periodicos.unb.br/index.php/RICI/article/view/8341>>. Acesso em: 25 ago. 2024.

TAILWIND CSS. **Tailwind CSS Documentation**. Disponível em: <<https://tailwindcss.com/docs>>. Acesso em: 15 jul. 2026.

TANSTACK. **TanStack Query Documentation.** Disponível em: <<https://tanstack.com/query/latest/docs/framework/react/overview>>. Acesso em: 23 mar. 2026.

VITE. **Vite Documentation.** Disponível em: <<https://vite.dev/>>. Acesso em: 15 jul. 2026.

ZUSTAND. **Zustand Documentation.** Disponível em: <<https://zustand.docs.pmnd.rs/>>. Acesso em: 23 mar. 2026.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Cajazeiras - Código INEP: 25008978
	Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
	CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

TCC com ficha catalográfica e folha de aprovação

Assunto:	TCC com ficha catalográfica e folha de aprovação
Assinado por:	Verissimo Pereira
Tipo do Documento:	Dissertação
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Verissimo Pereira Nobrega Terceiro, DISCENTE (202212010020) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 17/08/2026 12:41:21.

Este documento foi armazenado no SUAP em 17/08/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1953717
Código de Autenticação: 69719b1163

