

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**SISU ANALYZER: APLICAÇÃO VOLTADA PARA ANÁLISE DA EVOLUÇÃO DOS
CURSOS OFERTADOS PELO SISTEMA DE SELEÇÃO UNIFICADA**

LEONARDO MENDES MONTEIRO

Cajazeiras-PB

2025

LEONARDO MENDES MONTEIRO

**SISU ANALYZER: APLICAÇÃO VOLTADA PARA ANÁLISE DA EVOLUÇÃO DOS
CURSOS OFERTADOS PELO SISTEMA DE SELEÇÃO UNIFICADA**

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. Me. Afonso Serafim Jacinto

Cajazeiras-PB

2025

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

M775s Monteiro, Leonardo Mendes.

SISU analyzer : aplicação voltada para análise da evolução dos cursos ofertados pelo sistema de seleção unificada / Leonardo Mendes Monteiro. – Cajazeiras, 2025.

37f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.

Orientador: Prof. Me. Afonso Serafim Jacinto.

1. Desenvolvimento de sistemas. 2. Sistema de seleção unificada. 3. Análise de dados. 4. Cursos universitários. I. Instituto Federal da Paraíba. II. Título.

IFPB/CZ

CDU: 004.4(043.2)



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

LEONARDO MENDES MONTEIRO

**SISU ANALYZER: APLICAÇÃO VOLTADA PARA ANÁLISE DE TENDÊNCIAS E EVOLUÇÃO
DOS CURSOS OFERTADOS PELO SISTEMA DE SELEÇÃO UNIFICADA**

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. Me. Afonso Serafim Jacinto

Aprovada em: **24 de Março de 2025.**

Prof. Me. Afonso Serafim Jacinto - Orientador

Prof. Esp. João Igor Barros Rocha - Avaliador
IFPB - Campus Cajazeiras

Prof. Esp. Walyne Rachel Soares Mangueira - Avaliador
ECIT Professora Adilina de Sousa Diniz

Documento assinado digitalmente
 **WALYNE RACHEL SOARES MANGUEIRA**
Data: 25/03/2025 16:48:32-0300
Verifique em <https://validar.it.gov.br>

Documento assinado eletronicamente por:

- Afonso Serafim Jacinto, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 25/03/2025 07:45:46.
- Joao Igor Barros Rocha, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 25/03/2025 09:41:14.

Este documento foi emitido pelo SUAP em 25/03/2025. 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 687655
Verificador: 3e0a8e0c3c
Código de Autenticação:



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

RESUMO

O presente trabalho aborda o desenvolvimento do SISU ANALYZER, um sistema *web* projetado para analisar dados abertos do Sistema de Seleção Unificada (SISU). O objetivo principal é analisar a procura por cursos nas instituições públicas de ensino superior entre 2016 e 2023, utilizando informações disponibilizadas pelo Ministério da Educação (MEC). Para atingir essa finalidade, o projeto começa com a coleta e verificação dos dados abertos do SISU. O SISU ANALYZER será estruturado segundo a arquitetura Model-View-Controller (MVC) e empregará tecnologias como ReactJS, Python, Node.js, PostgreSQL, ORM Prisma, Jupyter Notebook, Char.js e Vite, assegurando uma interface intuitiva e eficiente análise dos dados. Este sistema visa contribuir para uma melhor compreensão do cenário educacional no Brasil, facilitando a tomada de decisões tanto por estudantes quanto por instituições de ensino.

Palavras-chave: Análise de Dados; SISU; Dados Abertos.

ABSTRACT

This paper discusses the development of SISU ANALYZER, a web-based system designed to analyze open data from the Unified Selection System (SISU). The main objective is to analyze the demand for courses in public higher education institutions between 2016 and 2023, using information provided by the Ministry of Education (MEC). To achieve this goal, the project begins with the collection and verification of open data from SISU. SISU ANALYZER will be structured according to the Model-View-Controller (MVC) architecture and will employ technologies such as ReactJS, Python, Node.js, PostgreSQL, ORM Prisma, Jupyter Notebook, Char.js, and Vite, ensuring an intuitive interface and efficient data analysis. This system aims to contribute to a better understanding of the educational scenario in Brazil, facilitating decision-making by both students and educational institutions.

Keywords: Data Analysis; SISU; Open Data.

LISTA DE FIGURAS

Figura 1: Padrão da Arquitetura MVC	13
Figura 2: Diagrama de Casos de Uso	19
Figura 3: Importação do Pandas e leitura de arquivos CSV	21
Figura 4: Seleção das colunas essenciais do banco	22
Figura 5: Processo de padronização de colunas	22
Figura 6: Processo de transformação de tipo de dado	23
Figura 7: Processo de exportação de banco de dados para CSV	23
Figura 8: Criação do banco de dados no PostgreSQL	24
Figura 9: Processo de importação dos arquivos para o banco de dados	24
Figura 10: Tela inicial	25
Figura 11: Tela de analisar a demanda por curso	26
Figura 12: Tela da demanda por curso em específico	27
Figura 13: Tela de Nota de corte	27
Figura 14: Tela do perfil dos candidatos	28
Figura 15: Gráfico de linha sobre a concorrência de ADS ao longo dos anos	29
Figura 16: Gráfico de pizza do SISU 2023	30
Figura 17: Gráfico de pizza do SISU 2016	30
Figura 18: Distribuição etária dos candidatos ao curso de ADS no SISU 2023	31

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CRUD	Create, Read, Update e Delete
ENEM	Exame Nacional do Ensino Médio
IA	Inteligência Artificial
IDC	International Data Corporation
LAI	Lei de acesso à informação
MEC	Ministério da Educação
MVC	Model-View-Controller
ORM	Object-Relational Mapping
RF	Requisitos Funcionais
RNF	Requisitos Não funcionais
SGBD	Sistema Gerenciador de Banco de Dados
SISU	Sistema de Seleção Unificada
SQL	Structured Query Language
TCC	Trabalho de Conclusão de Curso
UI	User Interface
UNIFESP	Universidade Federal de São Paulo

SUMÁRIO

1. INTRODUÇÃO	8
1.1. Objetivos	10
1.1.1. Objetivo Geral	10
1.1.2. Objetivos Específicos	10
1.2. Organização do Documento	10
2. METODOLOGIA	11
3. DESCRIÇÃO DO SISTEMA	12
3.1. Arquitetura do Sistema	12
3.2. TECNOLOGIAS UTILIZADAS	14
3.2.1 ReactJS	14
3.2.2 Python	15
3.2.3 Node.js	15
3.2.4 PostgreSQL	15
3.2.5 Prisma	16
3.2.6 Jupyter Notebook	16
3.2.7 Chart.js	16
3.2.8 Vite	17
3.3 MODELAGEM	17
3.3.1 Requisitos Funcionais	17
3.3.2 Requisitos não funcionais	18
3.3.3 Diagrama de Casos de Uso	19
4. IMPLEMENTAÇÃO	20
4.1 Banco De Dados	20
4.2 Front-End	25
5. RESULTADOS E DISCUSSÕES	30
6. CONSIDERAÇÕES FINAIS	33
REFERÊNCIAS	34

1. INTRODUÇÃO

Nos últimos anos, o volume de dados digitais tem crescido de forma exponencial, impulsionado por diversos fatores, como o aumento do uso da internet, o crescimento das redes sociais, desenvolvimento de tecnologias, armazenamento em nuvem e etc. De acordo com o relatório da International Data Corporation (IDC) citado por Coughlin (2018), estima-se que até 2025 serão gerados anualmente 175 zettabytes de dados. O crescimento destes dados será resultado da incorporação de agentes inteligentes que utilizam aprendizado de máquina e outras formas de inteligência artificial para analisar a crescente quantidade de dados gerados pelas coisas digitais em nossas vidas. (COUGHLIN, 2018).

Com a ascensão de tecnologias avançadas, como inteligência artificial (IA), Gomes (2017) define “a inteligência artificial sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana”. Dentro da IA existe uma área de estudo chamada Machine Learning ou Aprendizado de Máquina para Faceli (apud Cerri e Carvalho, 2017) é uma área de pesquisa que visa ao desenvolvimento de programas de computador com a capacidade de aprender a executar uma determinada tarefa com sua própria experiência. Com isso, tornou-se possível extrair insights valiosos desses grandes volumes de dados. Para este volume de dados, Alecrim (apud Rosa, 2022) abordam o termo Big Data, definindo “Big Data como conjuntos de dados tão amplos que possuem a necessidade da utilização de ferramentas capazes de lidar com grandes volumes de dados para assim as informações possam ser identificadas e analisadas”.

Mediante isso, a análise de Big Data permite que empresas e organizações identifiquem padrões, façam previsões precisas e tomem decisões informadas, aprimorando a eficiência operacional e a personalização de serviços. Segundo Weiner (2014) “Os dados realmente impulsionam tudo o que fazemos.” De acordo com Redator (2019) do portal It Forum, foi realizada uma pesquisa da MicroStrategy feita no segundo trimestre de 2019, contou com entrevistas com 500 profissionais de inteligência de negócios e análises em todo o Brasil, Alemanha, Japão, Reino Unido e Estados Unidos, todos os entrevistados têm conhecimentos sobre o uso de dados

e análises e envolvimento direto na tomada de decisões quando se trata da solução de análise de sua empresa. Os entrevistados representam 10 indústrias, (serviços financeiros, governo, saúde, hospitalidade, manufatura, varejo, tecnologia, serviços profissionais, telecomunicações e outros), e “94% dos entrevistados acreditam que dados e análises serão essenciais para o crescimento de sua empresa e sua transformação digital.” No Brasil, com a implementação da Lei nº 12.527/2011, Lei de acesso à informação (LAI), a cartilha da Lei de Acesso à Informação no Brasil (2013) ressalta:

A Lei nº 12.527 de 2011, conhecida como Lei de Acesso à Informação (LAI), trata de assuntos de interesse da União, dos Estados, do Distrito Federal e também dos Municípios. Como a própria Constituição Federal de 1988 prevê, todos têm direito a receber dos órgãos públicos tanto informações de seu interesse particular, quanto de interesse coletivo ou geral, lembrando-se sempre que algumas exceções existem para a própria segurança da sociedade e do Estado. Importante também lembrar que esta Lei inclui toda a Administração Direta e Indireta, considerando aqui também as entidades controladas direta ou indiretamente pelos Municípios. (CARTILHA LAI, 2013)

Diante do exposto, em território nacional, sob os preceitos do Ministério da Educação (MEC), o Sistema de Seleção Unificada (SISU) é a plataforma pela qual dá-se a seleção e distribuição das vagas destinadas às instituições públicas de ensino superior, que serão preenchidas pelos estudantes que estão concluindo o ensino médio e que participaram do Exame Nacional do Ensino Médio (ENEM). Na plataforma do MEC, há informações acerca das inscrições realizadas durante os processos seletivos já realizados, em forma de dados abertos.

Os dados abertos são os tipos de informações públicas ou privadas em forma de dados, que estão disponíveis para serem acessadas e utilizadas por qualquer pessoa, com o objetivo principal de causar transparência e promover melhorias na sociedade. A ideia de dados abertos principalmente os governamentais começou a se tornar mais popular a partir de 2009, quando diversos países, como Estados Unidos, Nova Zelândia, Canadá e Reino Unido, começaram a propor novas iniciativas de abertura de suas informações.

Com isso, este trabalho tem como objetivo fazer uma análise dos dados abertos do SISU nos anos de 2016 a 2022, e criar um sistema que tem a finalidade

de fazer uma transparência do processo seletivo, analisando os cursos mais procurados e menos procurados, qual vem crescendo ou diminuindo durante os anos no Brasil.

1.1. Objetivos

1.1.1. Objetivo Geral

Propor um sistema *web* para analisar dados do SISU (2016-2023), promovendo transparência ao identificar os cursos mais e menos procurados, as notas de corte e o perfil dos candidatos, como faixa etária e distribuição por sexo.

1.1.2. Objetivos Específicos

- Coletar e organizar os dados disponibilizados pelo SISU de 2016 a 2023 para uso no sistema de análise;
- Realizar a extração, transformação e carga dos dados.
- Criar um banco de dados estruturado com as informações coletadas
- Desenvolver uma interface de funcional para a visualização das análise do SISU, incluindo a evolução da demanda por cursos e áreas de estudo ao longo dos anos;

1.2. Organização do Documento

O restante do documento está estruturado da seguinte forma: o Tópico 2 apresenta a metodologia utilizada no projeto; o Tópico 3 descreve o sistema, detalhando suas tecnologias e arquitetura; o Tópico 4 aborda a implementação do sistema; o Tópico 5 exemplifica seu uso, considerando o curso de Análise e Desenvolvimento de Sistemas (ADS); e, por fim, o Tópico 6 traz as considerações finais do projeto.

2. METODOLOGIA

Neste tópico são abordadas as etapas realizadas para a criação do TCC, desde sua concepção até a implementação do sistema proposto. As etapas realizadas foram as seguintes:

1. Coleta e organização dos dados: Os dados do SISU (2016-2023) foram extraídos dos arquivos públicos do MEC, filtrando apenas a primeira chamada regular de cada ano e selecionando as colunas essenciais.

2. Processo de ETL: Utilizando Python¹ e a biblioteca Pandas², os dados foram extraídos, padronizados, convertidos para os formatos adequados e armazenados em arquivos processados.

3. Criação do banco de dados: Os dados tratados foram inseridos em um banco PostgreSQL³, estruturado para otimizar o armazenamento e a recuperação das informações.

4. Desenvolvimento da interface: Uma interface em React⁴ foi criada para visualização interativa dos dados, permitindo análises dinâmicas sobre demanda, notas de corte e perfil dos candidatos.

5. Análise da concorrência e perfil dos candidatos: Com o SISU ANALYZER, foram analisadas a evolução da concorrência e o perfil dos inscritos no curso de ADS do IFPB - Campus Cajazeiras, considerando gênero e faixa etária.

¹ <https://www.python.org/>

² <https://pandas.pydata.org/>

³ <https://www.postgresql.org/>

⁴ <https://react.dev/>

3. DESCRIÇÃO DO SISTEMA

O SISU ANALYZER é um sistema desenvolvido para *web* que tem como objetivo fazer a análise dos dados abertos do Sistema de Seleção Unificada (SISU), na qual essas informações são disponibilizada pelo *site* do Ministério da Educação (MEC), com a finalidade de demonstrar transparência no processo seletivo, fazendo uma análise dos cursos ofertados pelo Sisu em escolas públicas, como os cursos mais e menos procurados, as maiores e menores notas de corte e o perfil dos candidatos, com o intuito demonstrar de forma intuitiva as estatísticas no processo seletivo do SISU.

O sistema faz uma análise das inscrições de cada candidato no SISU, com objetivo de fazer um levantamento dos cursos mais e menos concorridos em cada ano do processo seletivo e também analisa as maiores e menores notas de corte por curso e por fim ele também tem a função para identificar o perfil dos candidatos de cada curso em específico, por meio da faixa etária e da distribuição de candidatos por sexo. Essa análise permite uma compreensão melhor do mercado educacional, ajudando candidatos e instituições de ensino a fazerem escolhas mais estratégicas, se baseando em dados concretos que demonstram a mudança de preferências.

Além disso, vale destacar que existe o portal dados abertos da Unifesp, onde tem semelhança com o sistema proposto, na qual promove a transparência dos dados abertos de qualquer sistema de seleção da Unifesp, principalmente do sistema de seleção do SISU, onde mostra as relações de maior e menor nota de alunos que entraram na Unifesp por este sistema, separando por campus e curso e disponibilizando os dados em forma de planilha e gráficos.

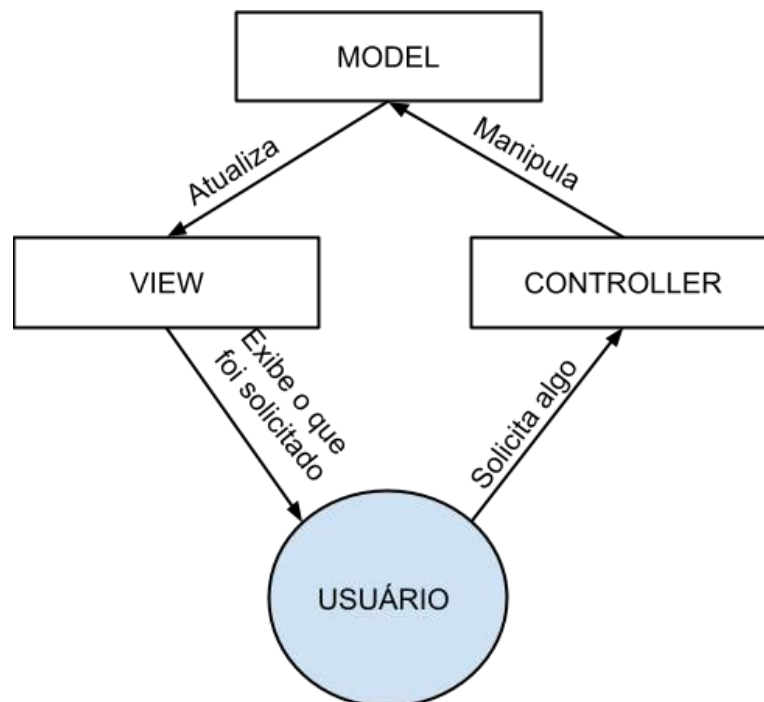
3.1. Arquitetura do Sistema

A arquitetura utilizada para o desenvolvimento do sistema é o padrão *Model-View-Controller* (MVC), foi criada por Trygve Reenskaug, iniciou em 1979 o padrão de projeto MVC, no artigo "Applications Programming in Smalltalk-80: How to use Model-View-Controller", foi descrito e mostrado a implementação do padrão.

Nessa arquitetura, o projeto é implementado em três camadas diferentes e independentes.

- A primeira delas é a camada Modelo, que é a parte da aplicação que é responsável por lidar com os dados, como eles são organizados e manipulados.
- Outra camada é a de Visualização, que é onde fica localizada a interface gráfica, ou seja, é a parte do *front-end*.
- A última camada é a de Controle, na qual é designada a receber requisições da camada de Visualização e processar e determinar o que deve ser feito, portanto ela é responsável por fazer a ligação entre as camadas de *View* e *Model*.

Figura 1: Padrão da Arquitetura MVC



Fonte: Tássio Auad (2014)

A camada *Model* no sistema, terá a ferramenta ORM Prisma, que faz a comunicação diretamente com o banco de dados PostgreSQL, na qual tem a função de fazer as operações de criar, ler, atualizar e remover no banco de dados, que quando feito as operações, serão atualizadas e mostradas na *View*.

A *View* é a camada que mostra a interface para o usuário, na qual exibe as informações e alterações salvas na camada *Model*, a tecnologia utilizada para desenvolver a interface foi o ReactJS, com isso tornou a criação de telas mais simplificada e dinâmica.

O *Controller* faz a interação entre a *View* e o *Model*, onde vai receber a solicitação da camada *view*, através do usuário que faz uma solicitação para o *controller*, na qual manipula o *Model* de acordo com o que foi pedido na *View* e depois retorna a atualização. Esta API foi desenvolvida através do Node.js⁵, com isso foi possível utilizar a linguagem de programação JavaScript para implementar o código.

3.2. TECNOLOGIAS UTILIZADAS

Este tópico tem como objetivo apresentar as tecnologias que foram utilizadas no desenvolvimento do SISU ANALYZER, para garantir uma base sólida na sua implementação, com a manutenção e a escalabilidade do sistema.

3.2.1 ReactJS

O React é uma biblioteca de *front-end* JavaScript de código aberto, que é utilizada para fazer interfaces de usuários em páginas *web*. De acordo com o *site* Hora de Codar (2024), o termo “*Front-end*” é usado no desenvolvimento *web* para descrever tudo o que é apresentado aos usuários quando eles visitam um *site* ou usam um aplicativo da *web*. Em termos simples, é a parte que você vê e com a qual interage”. O React foi criado pelo Facebook em 2011, com a finalidade de tornar mais fácil e intuitivo a criação de interfaces *web*. Ela é utilizada para criar componentes de interface que sejam reutilizáveis, com isso, o código fica mais eficiente e facilita a manutenção do código e o desenvolvimento, pois permite a reutilização de componentes em várias partes da aplicação. No sistema o ReactJS foi utilizado para criação de componentes de interfaces de usuário (UI), para visualização do usuário.

⁵ <https://nodejs.org/pt>

3.2.2 Python

Python é uma linguagem de programação que foi lançada em 1991 por Guido Van Rossum. Segundo a documentação oficial do Python (2024), “Python é uma linguagem fácil de aprender e poderosa. Ela tem estruturas de dados de alto nível eficientes e uma abordagem simples mas efetiva de programação orientada a objetos. A elegância de sintaxe e a tipagem dinâmica de Python aliadas com sua natureza interpretativa, o fazem a linguagem ideal para programas e desenvolvimento de aplicações rápidas em diversas áreas e na maioria das plataformas”. Esta linguagem não foi utilizado diretamente no sistema, ela teve a função de uma ferramenta complementar, na qual foi utilizada um *script*, ou seja, um código em python para filtrar e povoar as tabelas do banco, somente com os dados necessários que irão ser utilizados no sistema e que vão ser salvos no banco de dados postgres.

3.2.3 Node.js

O Node.js é um *software* de código aberto, que utiliza a linguagem de programação JavaScript ao lado do servidor, por utilizar o motor v8 do Google Chrome, se torna possível executar JavaScript fora do navegador. Com o Node.js, ficou possível que os desenvolvedores utilizem o JavaScript para criação de servidores e Apis, desenvolvendo aplicações *web* rápidas e escaláveis. O Node.js foi utilizado no sistema, para a implementação do *back-end*, com isso se tornou possível utilizar Javascript ao lado do servidor. Para a plataforma de aprendizagem de tecnologia Awari (2023), *back-end* é a parte do sistema que lida com a lógica e o processamento de dados, trabalhando nos bastidores para garantir o funcionamento correto de uma aplicação.

3.2.4 PostgreSQL

De acordo com a documentação oficial do PostgreSQL (2024), o PostgreSQL é um sistema de gerenciamento de banco de dados (SGBD), de objeto relacional e código aberto. Foi iniciado o desenvolvimento dele, no Departamento de Ciências da Computação da Universidade da Califórnia, Berkeley. Ele é um dos bancos de dados mais avançados e usados no mundo, por causa da sua robustez,

compatibilidade com padrões SQL e por suportar grandes volumes de dados. Foi utilizado para armazenar os dados abertos, ou seja, tinha a função de armazenar, gerenciar e consultar as informações.

3.2.5 Prisma

O Prisma é um ORM, ou seja, uma ferramenta de mapeamento objeto-relacional para aplicações em Node.js e TypeScript, na qual simplifica a interação com bancos de dados, pois não precisa a utilização de SQL diretamente. Ele tem uma API intuitiva e tipada para fazer as principais funções de manipulação de dados: Criar, ler, atualizar e deletar, também conhecidas como CRUD (*Create, read, update e Delete*). Foi escolhida esta ORM por causa da seguinte vantagem, de acordo com a documentação do Prisma (2014), o Prisma ORM torna mais fácil para os desenvolvedores raciocinar sobre suas consultas de banco de dados, fornecendo uma API limpa e segura para enviar consultas de banco de dados que retorna objetos JavaScript antigos e simples. No sistema, o prisma ⁶foi utilizado para fazer a comunicação entre o postgres e o *back-end*.

3.2.6 Jupyter Notebook

O Jupyter Notebook⁷ é um Ambiente de Desenvolvimento Integrado, do inglês, *Integrated Development Environment* (IDE) baseado na *web*, utilizado para a criação e o compartilhamento de documentos que possuem códigos, textos, equações matemáticas e visualizações de gráficos, tabelas e diagramas. Sua popularidade se destaca especialmente nas áreas de ciência de dados, aprendizado de máquina e análise de dados, devido à sua flexibilidade e capacidade de executar trechos de código de forma interativa. No sistema, o Jupyter Notebook teve um papel importante para implementar a Extração, Transformação e Carregamento (ETL) dos dados.

3.2.7 Chart.js

O Chart.js⁸ é uma biblioteca JavaScript de código aberto utilizada para a criação de gráficos interativos e responsivos em aplicações *web*. Sua aplicação

⁶ <https://www.prisma.io/>

⁷ <https://jupyter.org/>

⁸ <https://www.chartjs.org/>

neste sistema é utilizada pela necessidade de apresentar a visualização dos dados de forma clara e acessível, permitindo uma melhor interpretação das informações por meio de representações gráficas dinâmicas.

3.2.8 Vite⁹

O Vite é uma ferramenta de *build* para *front-end* que auxilia no desenvolvimento ao disponibilizar um ambiente rápido e eficiente, foi utilizada no projeto por causa de sua simplicidade e facilidade de uso e também por conta da capacidade que ele tem de agilizar o processo de desenvolvimento.

3.3 MODELAGEM

3.3.1 Requisitos Funcionais

- **RF01 - Fazer análise da demanda por curso:** O sistema permite que o usuário selecione a opção para fazer uma análise da demanda por curso, onde o software vai gerar gráficos entre os anos de 2016 a 2023.
- **RF 2 - Fazer análise por nota de corte por curso:** O sistema deve permitir ao usuário selecionar a opção para fazer a análise por nota de corte por curso, onde o *software* fará a análise das notas de corte dos cursos ofertados pelo SISU.
- **RF 3 - Fazer análise do perfil dos candidatos por curso:** O *software* disponibiliza a opção para o usuário fazer uma análise do perfil dos candidatos por curso.
- **RF 4 - Fazer análise dos cursos mais concorridos:** Uma vez que o usuário selecionou a opção demanda por curso, o sistema vai gerar uma análise dos cursos mais concorridos, de acordo com a filtragem selecionada pelo usuário.
- **RF 5 - Fazer análise dos cursos menos concorridos:** Uma vez que o usuário selecionou a opção demanda por curso, o sistema irá gerar um gráfico com as análises dos cursos menos concorridos, com base na filtragem escolhida pelo usuário.
- **RF 6 - Fazer análise das maiores notas de corte:** Uma vez que o usuário selecionou a opção nota de corte por curso no SISU, o *software* gera uma

⁹ <https://vite.dev/>

análise com as maiores notas de corte por curso, assim mostrando as instituições com as maiores notas de corte por curso selecionado pelo usuário.

- **RF 7 - Fazer análise das menores notas de corte:** Uma vez que o usuário optou por selecionar a opção nota de corte por curso, o sistema gera um gráfico com a análise das menores notas de corte por curso no SISU, assim mostrando as instituições com as menores notas de corte por curso, de acordo com o curso selecionado pelo usuário.
- **RF 8 - Fazer análise da faixa etária:** Uma vez que o usuário selecionou a opção perfil dos candidatos por curso no SISU, o *software* gera uma análise com a faixa etária dos candidatos do curso de uma instituição específica, de acordo com a filtragem do curso selecionado pelo usuário.
- **RF 9 - Fazer análise por sexo:** Uma vez que o usuário selecionou a opção perfil dos candidatos por curso no SISU, o sistema gera uma análise com a proporção dos candidatos masculinos e femininos do curso de um campus específico, de acordo com a filtragem do curso escolhida pelo usuário.

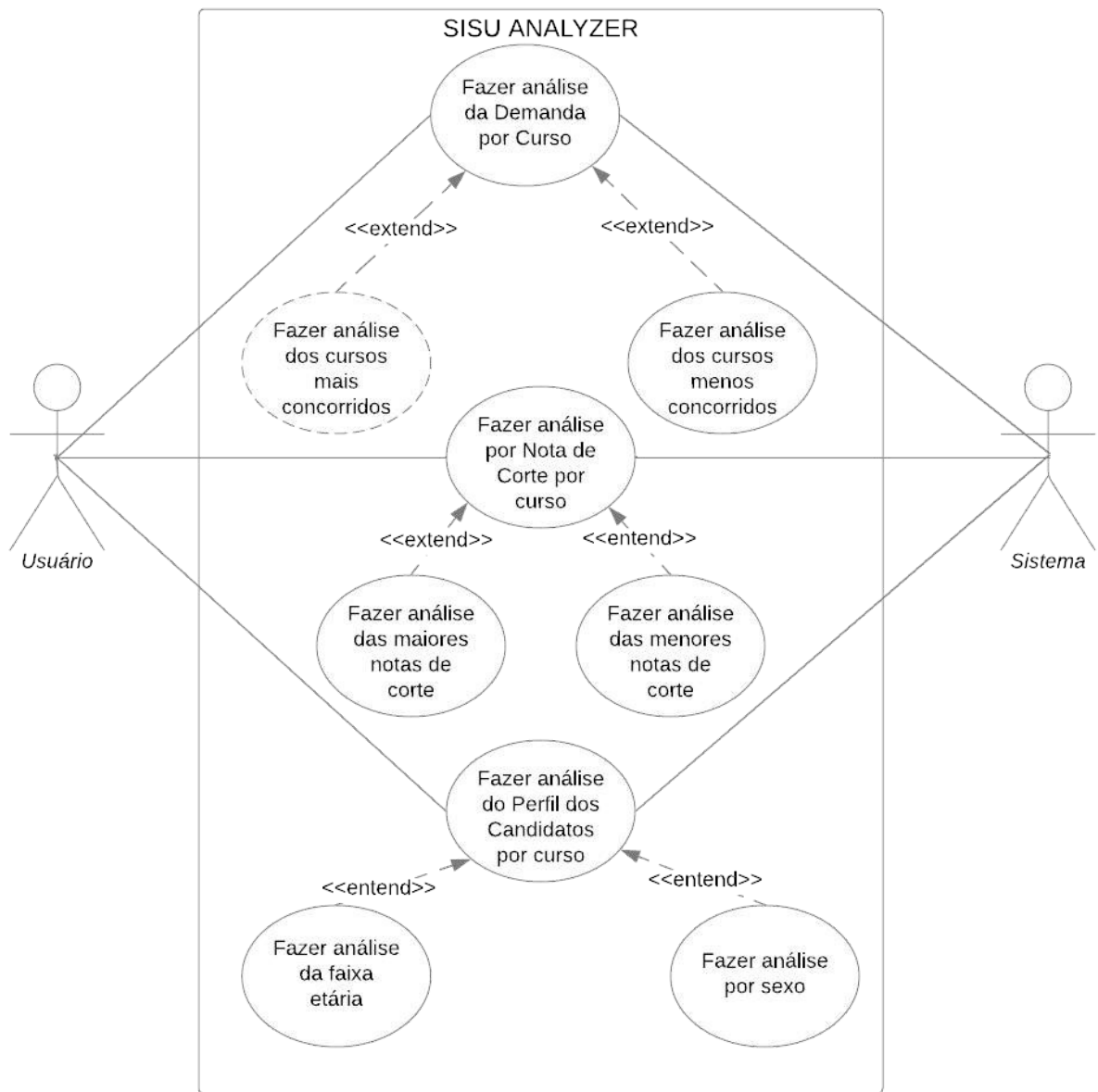
3.3.2 Requisitos não funcionais

- **RNF 1 - Disponibilidade:** O sistema deve ficar disponível 99% do tempo, exceto quando for necessário fazer uma manutenção, seja para modificar e atualizar algumas funcionalidades ou para corrigir algum *bug* no sistema.
- **RNF 2 - Usabilidade:** O sistema terá uma interface intuitiva e simples.
- **RNF 3 - Qualidade de código:** O código do sistema deve ser organizado, legível e de fácil manutenção.
- **RNF 4 - Portabilidade:** O sistema terá compatibilidade com vários navegadores modernos.

3.3.3 Diagrama de Casos de Uso

O diagrama de casos de uso apresentado na Figura 3 ilustra de forma detalhada, todas as ações possíveis que o usuário e o sistema podem fazer, com o intuito de apresentar cada etapa, que deve ser seguida para a utilização das funcionalidades que o sistema oferece.

Figura 2: Diagrama de Casos de Uso



Fonte: Autor

4. IMPLEMENTAÇÃO

Esta seção apresenta a implementação do SISU ANALYZER, detalhando o desenvolvimento desde a estruturação do banco de dados até a criação da camada de *front-end*.

4.1 Banco De Dados

Um banco de dados é um conjunto de dados armazenados de forma estruturada, geralmente gerenciado por um Sistema de Gerenciamento de Banco de Dados (SGBD). Neste projeto, foi utilizado o PostgreSQL, o qual tem a função de gerenciar, armazenar e buscar informações dos dados do SISU.

Os dados abertos utilizados foram extraídos no *site* do Ministério da Educação (MEC), que disponibiliza arquivos públicos no formato *Comma-Separated Values* (CSV), contendo informações sobre os processos seletivos do SISU entre os anos de 2016 a 2023. Foram utilizados apenas os arquivos referentes à primeira chamada regular de cada ano. Antes de serem armazenados, os dados passaram por um processo de Extração, Transformação e Carga (ETL).

O ETL, como o nome já diz, é o processo de extrair, transformar e carregar os dados, permitindo que eles recebam algum tratamento antes de serem armazenados no banco de dados. A partir disso, é possível fazer uma análise eficiente mesmo com dados de diferentes fontes. O processo de ETL foi realizado com a utilização da linguagem de programação Python, utilizando a IDE Jupyter Notebook e a biblioteca Pandas¹⁰ para fazer a manipulação dos dados em forma de tabelas.

O processo de ETL seguiu as seguintes etapas:

1. Leitura e armazenamento dos arquivos csv

Os dados foram extraídos e carregados por meio do comando `read_csv` da biblioteca Pandas, garantindo que cada arquivo fosse lido e armazenado corretamente. O Pandas é uma biblioteca amplamente utilizada para manipulação e análise de dados em Python, oferecendo suporte nativo para a integração com muitos formatos de arquivo ou fontes de dados, como CSV,

¹⁰ <https://pandas.pydata.org/>

Excel, SQL, JSON e Parquet (PANDAS, 2025). A Figura 3 ilustra o processo de importação do Pandas e a leitura dos arquivos CSV.

Figura 3: Importação do Pandas e leitura de arquivos CSV

```
[1]: import pandas as pd

Ler e armazenar cada arquivo csv em sua respectiva variável

[2]: sisu2023 = pd.read_csv('chamada_regular_sisu_2023_1.csv', encoding='iso-8859-1', sep='|', low_memory=False)
[3]: sisu2022 = pd.read_csv('chamada_regular_sisu_2022_1.csv', encoding='iso-8859-1', sep='|', low_memory=False)
[4]: sisu2021 = pd.read_csv('ListagemChamadaRegular_2021-1.csv', sep=';', low_memory=False)
[5]: sisu2020 = pd.read_csv('ListagemChamadaRegular_2020-1.csv', sep=';', low_memory=False)
[6]: sisu2019 = pd.read_csv('ListagemChamadaRegular_2019-1.csv', sep=';', low_memory=False)
[7]: sisu2018 = pd.read_csv('Relatório Chamada Regular SISU 1 2018.csv', encoding='iso-8859-1', sep='|', low_memory=False)
[8]: sisu2017 = pd.read_csv('ListagemChamadaRegular_2017-1.csv', encoding='iso-8859-1', sep='|', low_memory=False)
[9]: sisu2016 = pd.read_csv('relatorio_chamada_regular_SISU_1.2016.csv', sep='|', low_memory=False)
```

Fonte: Autor

2. Seleção das colunas mais relevantes

Para otimizar o banco de dados e evitar informações desnecessárias, foram selecionadas apenas as colunas essenciais para o funcionamento do projeto. A Figura 4 ilustra esse processo, destacando as colunas escolhidas: *ano*, *sigla_eis*, *nome_campus*, *uf_campus*, *municipio_campus*, *nome_curso*, *grau*, *mod_concorrencia*, *sexo*, *data_nascimento*, *nota_candidato*, *nota_corte* e *aprovado*.

3. Padronização dos nomes das colunas

Os nomes das colunas foram uniformizados para garantir consistência e facilitar a manipulação dos dados. Por exemplo, as colunas *'DS_MOD_CONCORRENCIA'* e *'ds_modalidade'* foram transformadas em *'MOD_CONCORRENCIA'*. A Figura 5 ilustra o processo de padronização das colunas.

Figura 4: Seleção das colunas essenciais do banco

```
Selecionando somente as colunas que serão utilizadas

sisu2023 = sisu2023.loc[:, ['ANO', 'SIGLA_IES', 'NOME_CAMPUS', 'UF_CAMPUS', 'MUNICIPIO_CAMPUS', 'NOME_CURSO', 'GRAU', 'MOD_CONCORRENCIA',
                          'SEXO', 'DT_NASCIMENTO', 'NOTA_CANDIDATO', 'NOTA_CORTE', 'APROVADO']]

sisu2022 = sisu2022.loc[:, ['ANO', 'SIGLA_IES', 'NOME_CAMPUS', 'UF_CAMPUS', 'MUNICIPIO_CAMPUS', 'NOME_CURSO', 'GRAU', 'MOD_CONCORRENCIA',
                          'SEXO', 'DATA_NASCIMENTO', 'NOTA_CANDIDATO', 'NOTA_CORTE', 'APROVADO']]

sisu2021 = sisu2021.loc[:, ['NU_ANO', 'SIGLA_IES', 'NO_CAMPUS', 'SG_UF_CAMPUS', 'NO_MUNICIPIO_CAMPUS', 'NO_CURSO', 'DS_GRAU', 'DS_MOD_CONCORRENCIA',
                          'TP_SEXO', 'DT_NASCIMENTO', 'NU_NOTA_CANDIDATO', 'NU_NOTACORTE', 'ST_APROVADO']]

sisu2020 = sisu2020.loc[:, ['NU_ANO', 'SIGLA_IES', 'NO_CAMPUS', 'SG_UF_CAMPUS', 'NO_MUNICIPIO_CAMPUS', 'NO_CURSO', 'DS_GRAU', 'DS_MOD_CONCORRENCIA',
                          'TP_SEXO', 'DT_NASCIMENTO', 'NU_NOTA_CANDIDATO', 'NU_NOTACORTE', 'ST_APROVADO']]

sisu2019 = sisu2019.loc[:, ['NU_ANO', 'SIGLA_IES', 'NO_CAMPUS', 'SG_UF_CAMPUS', 'NO_MUNICIPIO_CAMPUS', 'NO_CURSO', 'DS_GRAU', 'DS_MOD_CONCORRENCIA',
                          'TP_SEXO', 'DT_NASCIMENTO', 'NU_NOTA_CANDIDATO', 'NU_NOTACORTE', 'ST_APROVADO']]

sisu2018 = sisu2018.loc[:, ['ANO', 'SIGLA_IES', 'NOME_CAMPUS', 'UF_CAMPUS', 'MUNICIPIO_CAMPUS', 'NOME_CURSO', 'GRAU', 'MOD_CONCORRENCIA',
                          'SEXO', 'DATA_NASCIMENTO', 'NOTA_CANDIDATO', 'NOTA_CORTE', 'APROVADO']]

sisu2017 = sisu2017.loc[:, ['NU_ANO', 'SIGLA_IES', 'NO_CAMPUS', 'SG_UF_CAMPUS', 'NO_MUNICIPIO_CAMPUS', 'NO_CURSO', 'DS_GRAU', 'DS_MOD_CONCORRENCIA',
                          'TP_SEXO', 'DT_NASCIMENTO', 'NU_NOTA_CANDIDATO', 'NU_NOTACORTE', 'ST_APROVADO']]

sisu2016 = sisu2016.loc[:, ['nu_ano', 'sigla_ies', 'campus', 'uf_campus', 'municipio_campus', 'nome_curso', 'grau', 'ds_modalidade',
                          'tp_sexo', 'dt_nascimento', 'nota_concorrencia', 'nu_notacorte', 'st_aprovado']]
```

Fonte: Autor

Figura 5: Processo de padronização de colunas

```
Padronizando os nomes das colunas

sisu2023.rename(columns={'DT_NASCIMENTO': 'DATA_NASCIMENTO'}, inplace=True)

sisu2021.rename(columns={'NU_ANO': 'ANO', 'NO_CAMPUS': 'NOME_CAMPUS', 'SG_UF_CAMPUS': 'UF_CAMPUS', 'NO_MUNICIPIO_CAMPUS': 'MUNICIPIO_CAMPUS',
                        'NO_CURSO': 'NOME_CURSO', 'DS_GRAU': 'GRAU', 'DS_MOD_CONCORRENCIA': 'MOD_CONCORRENCIA', 'TP_SEXO': 'SEXO',
                        'DT_NASCIMENTO': 'DATA_NASCIMENTO', 'NU_NOTA_CANDIDATO': 'NOTA_CANDIDATO', 'NU_NOTACORTE': 'NOTA_CORTE',
                        'ST_APROVADO': 'APROVADO'}, inplace=True)

sisu2020.rename(columns={'NU_ANO': 'ANO', 'NO_CAMPUS': 'NOME_CAMPUS', 'SG_UF_CAMPUS': 'UF_CAMPUS', 'NO_MUNICIPIO_CAMPUS': 'MUNICIPIO_CAMPUS',
                        'NO_CURSO': 'NOME_CURSO', 'DS_GRAU': 'GRAU', 'DS_MOD_CONCORRENCIA': 'MOD_CONCORRENCIA', 'TP_SEXO': 'SEXO',
                        'DT_NASCIMENTO': 'DATA_NASCIMENTO', 'NU_NOTA_CANDIDATO': 'NOTA_CANDIDATO', 'NU_NOTACORTE': 'NOTA_CORTE',
                        'ST_APROVADO': 'APROVADO'}, inplace=True)

sisu2019.rename(columns={'NU_ANO': 'ANO', 'NO_CAMPUS': 'NOME_CAMPUS', 'SG_UF_CAMPUS': 'UF_CAMPUS', 'NO_MUNICIPIO_CAMPUS': 'MUNICIPIO_CAMPUS',
                        'NO_CURSO': 'NOME_CURSO', 'DS_GRAU': 'GRAU', 'DS_MOD_CONCORRENCIA': 'MOD_CONCORRENCIA', 'TP_SEXO': 'SEXO',
                        'DT_NASCIMENTO': 'DATA_NASCIMENTO', 'NU_NOTA_CANDIDATO': 'NOTA_CANDIDATO', 'NU_NOTACORTE': 'NOTA_CORTE',
                        'ST_APROVADO': 'APROVADO'}, inplace=True)

sisu2017.rename(columns={'NU_ANO': 'ANO', 'NO_CAMPUS': 'NOME_CAMPUS', 'SG_UF_CAMPUS': 'UF_CAMPUS', 'NO_MUNICIPIO_CAMPUS': 'MUNICIPIO_CAMPUS',
                        'NO_CURSO': 'NOME_CURSO', 'DS_GRAU': 'GRAU', 'DS_MOD_CONCORRENCIA': 'MOD_CONCORRENCIA', 'TP_SEXO': 'SEXO',
                        'DT_NASCIMENTO': 'DATA_NASCIMENTO', 'NU_NOTA_CANDIDATO': 'NOTA_CANDIDATO', 'NU_NOTACORTE': 'NOTA_CORTE',
                        'ST_APROVADO': 'APROVADO'}, inplace=True)

sisu2016.rename(columns={'nu_ano': 'ANO', 'sigla_ies': 'SIGLA_IES', 'campus': 'NOME_CAMPUS', 'uf_campus': 'UF_CAMPUS',
                        'municipio_campus': 'MUNICIPIO_CAMPUS', 'nome_curso': 'NOME_CURSO', 'grau': 'GRAU', 'ds_modalidade': 'MOD_CONCORRENCIA',
                        'tp_sexo': 'SEXO', 'dt_nascimento': 'DATA_NASCIMENTO', 'nota_concorrencia': 'NOTA_CANDIDATO', 'nu_notacorte': 'NOTA_CORTE',
                        'st_aprovado': 'APROVADO'}, inplace=True)
```

Fonte: Autor

4. Conversão dos tipos de dados

Algumas colunas precisaram ter seus dados convertidos para outro tipo de dado para garantir o funcionamento adequado do sistema. Esse foi o caso das colunas *nota_candidato* e *nota_corte*, originalmente armazenadas como string, que precisaram ser convertidas para o formato numérico *float*. Para que a conversão fosse realizada corretamente, foi necessário substituir as

vírgulas por pontos, pois o Python interpreta casas decimais utilizando pontos. A Figura 6 ilustra esse processo de transformação do tipo de dado.

Figura 6: Processo de transformação de tipo de dado

```
Transformando as colunas NOTA_CANDIDATO E NOTA_CORTE de string para valor numérico (float)

sisu2023['NOTA_CANDIDATO'] = sisu2023['NOTA_CANDIDATO'].str.replace(',', '.')
sisu2023['NOTA_CANDIDATO'] = sisu2023['NOTA_CANDIDATO'].astype(float)
sisu2023['NOTA_CORTE'] = sisu2023['NOTA_CORTE'].str.replace(',', '.')
sisu2023['NOTA_CORTE'] = sisu2023['NOTA_CORTE'].astype(float)
sisu2022['NOTA_CANDIDATO'] = sisu2022['NOTA_CANDIDATO'].str.replace(',', '.')
sisu2022['NOTA_CANDIDATO'] = sisu2022['NOTA_CANDIDATO'].astype(float)
sisu2022['NOTA_CORTE'] = sisu2022['NOTA_CORTE'].str.replace(',', '.')
sisu2022['NOTA_CORTE'] = sisu2022['NOTA_CORTE'].astype(float)
sisu2021['NOTA_CANDIDATO'] = sisu2021['NOTA_CANDIDATO'].str.replace(',', '.')
sisu2021['NOTA_CANDIDATO'] = sisu2021['NOTA_CANDIDATO'].astype(float)
sisu2021['NOTA_CORTE'] = sisu2021['NOTA_CORTE'].str.replace(',', '.')
sisu2021['NOTA_CORTE'] = sisu2021['NOTA_CORTE'].astype(float)
```

Fonte: Autor

5. Salvamento dos dados processados

Após a conversão e padronização, os dados foram exportados e armazenados em arquivos CSV, prontos para serem carregados no banco de dados PostgreSQL. A Figura 7 ilustra o processo de exportação dos dados para os arquivos CSV.

Figura 7: Processo de exportação de bancos de dados para CSV

```
Salvando os dataBase em arquivos csv

sisu2023.to_csv("sisu2023.csv", index=False)
sisu2022.to_csv("sisu2022.csv", index=False)
sisu2021.to_csv("sisu2021.csv", index=False)
sisu2020.to_csv("sisu2020.csv", index=False)
sisu2019.to_csv("sisu2019.csv", index=False)
sisu2018.to_csv("sisu2018.csv", index=False)
sisu2017.to_csv("sisu2017.csv", index=False)
sisu2016.to_csv("sisu2016.csv", index=False)
sisu2016['NOME_CURSO'].count()
```

Fonte: Autor

6. Criação do banco de dados no PostgreSQL

Depois dos dados processados serem salvos, foram feitas as criações das tabelas no banco de dados do PostgreSQL. A Figura 8 ilustra o processo de criação das tabelas no PostgreSQL.

Figura 8: Processo de criação das tabelas no banco de dados

```
create table sisu2023 (  
    id SERIAL PRIMARY KEY,  
    ano INTEGER,  
    sigla_ies TEXT,  
    nome_campus TEXT,  
    uf_campus VARCHAR(2),  
    municipio_campus TEXT,  
    nome_curso TEXT,  
    grau TEXT,  
    mod_concorrencia TEXT,  
    sexo VARCHAR(1),  
    data_nascimento INTEGER,  
    nota_candidato REAL,  
    nota_corte REAL,  
    aprovado VARCHAR(1)  
)
```

Fonte: Autor

7. Importação dos arquivos no banco de dados

Após a criação das tabelas no PostgreSQL, foi iniciado a importação dos arquivos processados, para o banco de dados. A figura 9 ilustra o processo da importação dos arquivos para o banco de dados.

Figura 9: Processo de importação dos arquivos para o banco de dados

```
COPY sisu2023(ano, sigla_ies, nome_campus, uf_campus, municipio_campus, nome_curso, grau, mod_concorrencia, sexo,  
    data_nascimento, nota_candidato, nota_corte, aprovado) FROM 'C:\Program Files\PostgreSQL\17\bin\Database\sisu2023.csv'  
with(  
    DELIMITER ',',  
    HEADER,  
    FORMAT CSV  
)
```

Fonte: Autor

4.2 Front-End

O *front-end* representa a camada vista pelo usuário, na qual ele pode interagir com a aplicação. No SISU ANALYSER, essa camada foi criada com React, escolhido por ser uma tecnologia fácil e intuitiva na implementação de interfaces, por ser capaz de utilizar a componentização, fazendo com que seja possível fazer a reutilização de códigos, fazendo a implementação ser mais rápida e intuitiva.

Nesta subseção, são apresentadas as telas do SISU ANALYSER e suas funcionalidades.

TELA INICIAL

A Figura 10 exibe a tela inicial do sistema, com o objetivo principal de introduzir para o usuário uma breve explicação sobre o objetivo e as principais funcionalidades, que são: demanda por cursos, notas de corte e o perfil dos candidatos ao longo dos anos de 2016 a 2023 no SISU.

Figura 10: Tela inicial



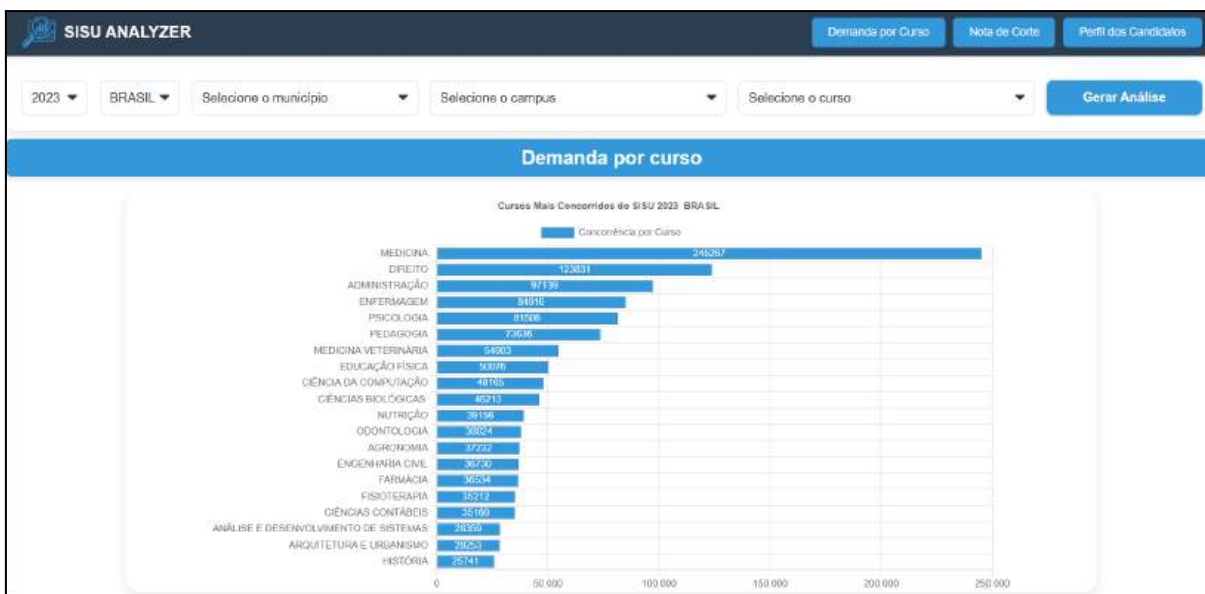
Fonte: Autor

ANÁLISE DA DEMANDA POR CURSO

A Figura 11 apresenta a tela da função de analisar a demanda por curso, na qual o usuário pode filtrar essa análise de acordo com as seguintes opções: Ano,

Nível Geográfico (Brasil ou estado), Município, Campus e também por Curso específico. Ao escolher a filtragem e clicar no botão “Gerar Análise”, o sistema retornará dois gráficos, o primeiro com os cursos mais concorridos de acordo com a filtragem e outro com os cursos menos concorridos.

Figura 11: Tela de analisar a demanda por curso

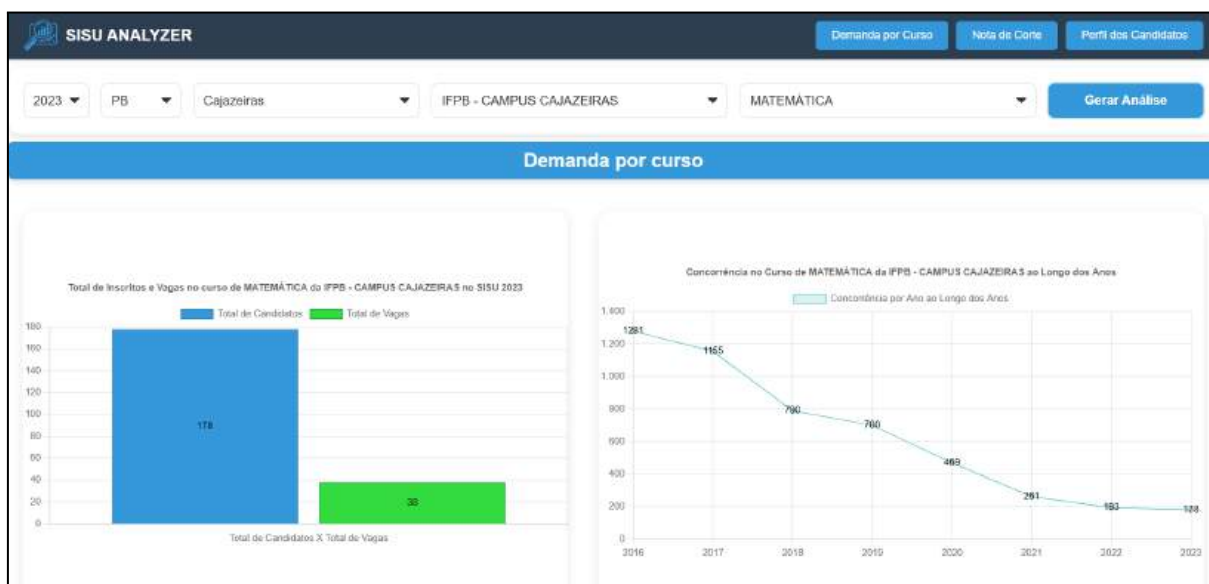


Fonte: Autor

DEMANDA POR CURSO ESPECÍFICO

A Figura 12 mostra a tela de análise da demanda por um curso específico. O usuário pode selecionar o campus e o curso que deseja analisar. Após escolher o curso e clicar em “Gerar Análise”, o sistema exibe dois gráficos: o primeiro é um gráfico de barras que compara o total de candidatos com o número de vagas disponíveis; o segundo é um gráfico de linha que mostra a evolução da concorrência desse curso no SISU ao longo dos anos.

Figura 12: Tela da demanda por curso em específico

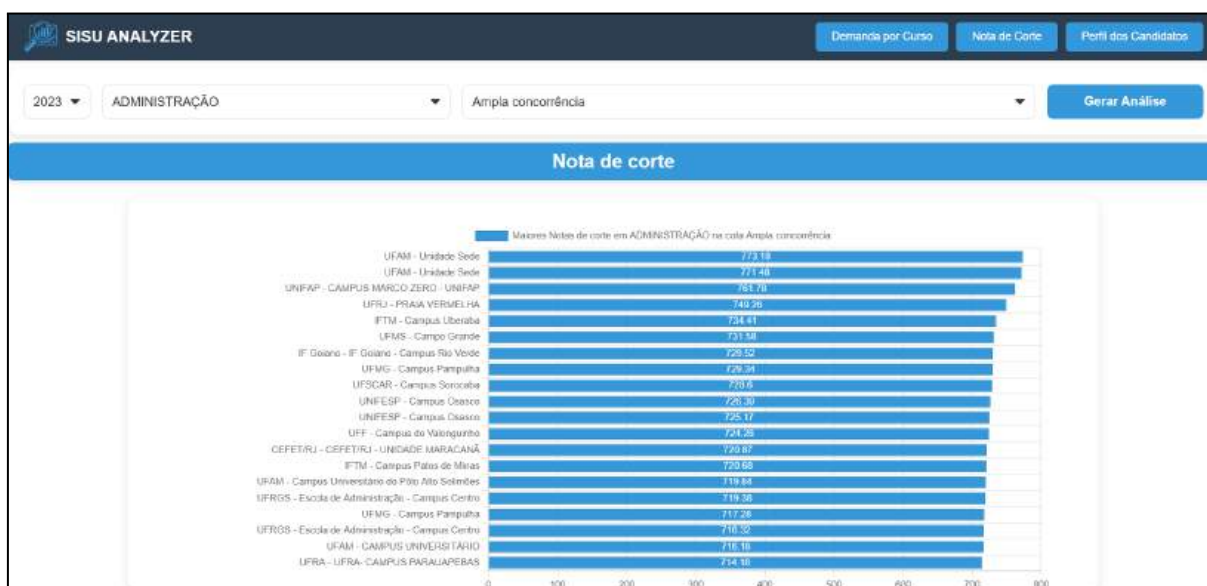


Fonte: Autor

NOTA DE CORTE

A Figura 13 apresenta a tela da nota de corte, na qual o usuário pode filtrar a análise das notas de corte do SISU, de acordo com o ano, curso e o modo de concorrência. Ao clicar em “Gerar Análise”, o sistema vai carregar dois gráficos de barras com as maiores e menores notas de corte das instituições que ofertam o curso selecionado em específico.

Figura 13: Tela de Nota de corte

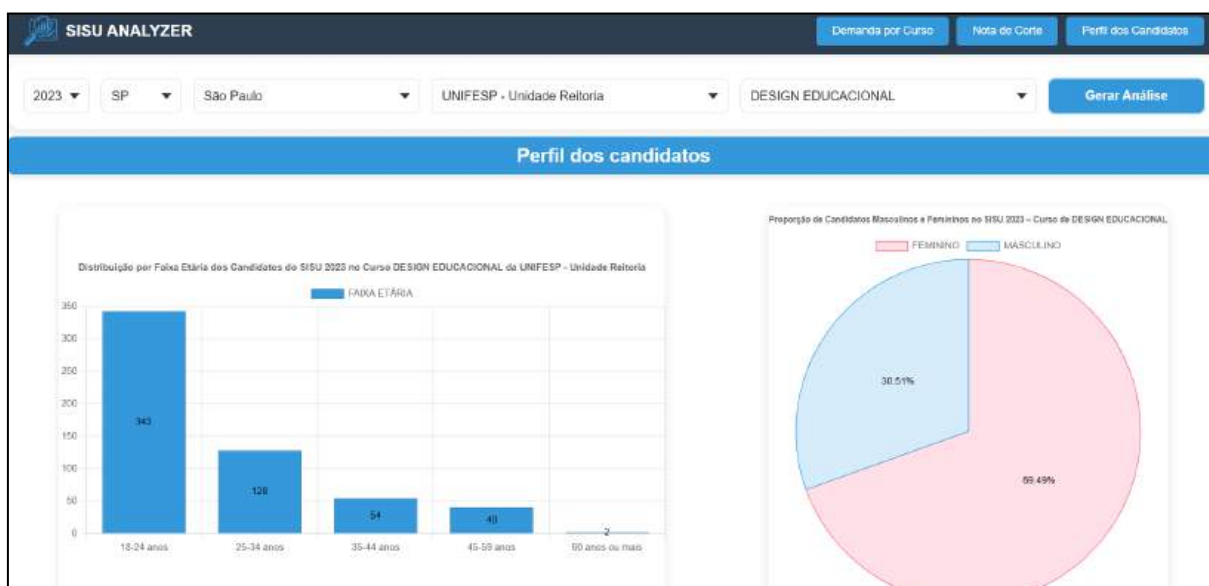


Fonte: Autor

PERFIL DOS CANDIDATOS

A Figura 14 apresenta a tela do perfil dos candidatos, onde o usuário pode ver o perfil das pessoas que concorreram em um curso pelo SISU. Nesse caso, o usuário é obrigado a selecionar todas as opções para fazer a análise, tendo que escolher de acordo com o ano, estado, município, campus e curso. Ao gerar a análise o sistema vai retornar dois gráficos um de barra com a distribuição da faixa etária dos candidatos que concorreram ao curso específico da instituição selecionada e outro gráfico de pizza, que tem a proporção de candidatos masculinos e femininos daquele curso.

Figura 14: Tela do perfil dos candidatos



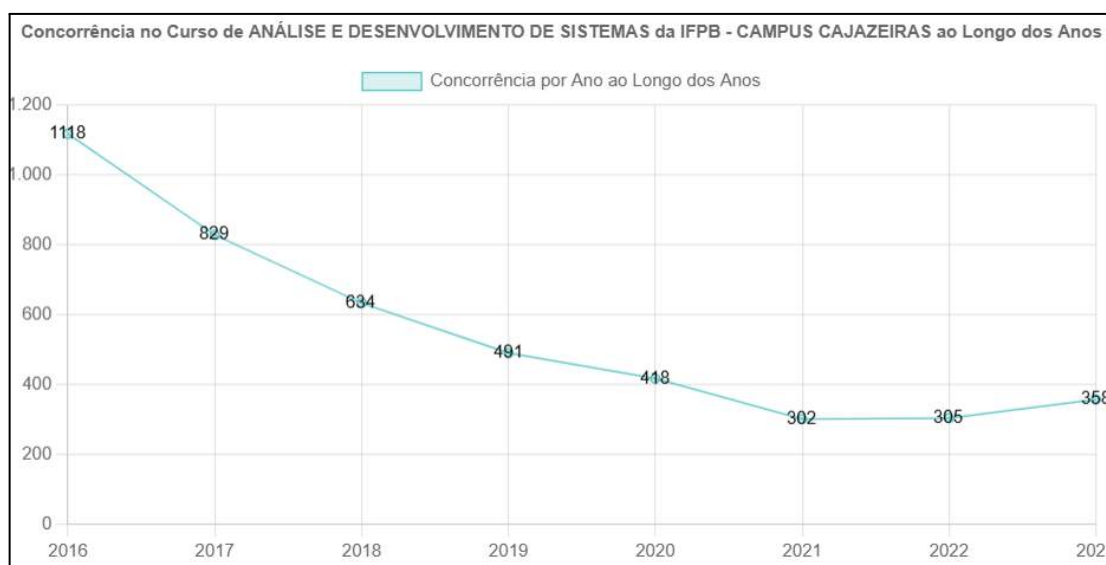
Fonte: Autor

5. RESULTADOS E DISCUSSÕES

Nesta seção é apresentada a análise da concorrência e o perfil dos candidatos ao curso de Análise e Desenvolvimento de Sistemas da instituição IFPB - Campus Cajazeiras ao longo dos anos no SISU, utilizando o SISU ANALYZER. São abordadas as variações na concorrência, a distribuição entre candidatos masculinos e femininos e o perfil etário dos inscritos, com base nos dados de 2016 e 2023.

A Figura 15 exibe o gráfico de linha que apresenta a concorrência do curso de ADS da instituição IFPB Campus Cajazeiras ao longo dos anos no SISU. É possível observar uma queda da concorrência desse curso, que em 2016 teve 1118 inscritos e em 2023 teve apenas 358 candidatos, o que representa uma diminuição de mais de 70% na concorrência entre esses anos.

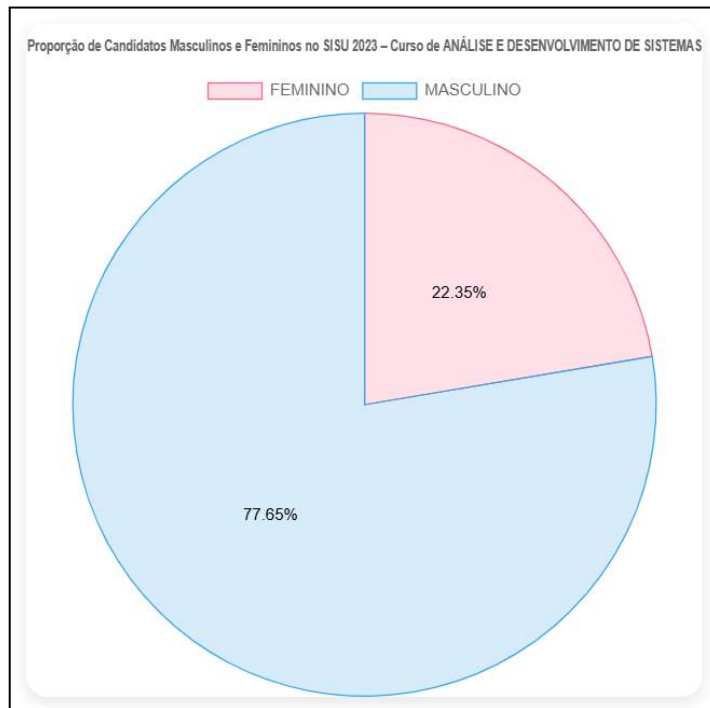
Figura 15: Gráfico de linha sobre a concorrência de ADS ao longo dos anos



Fonte: Autor

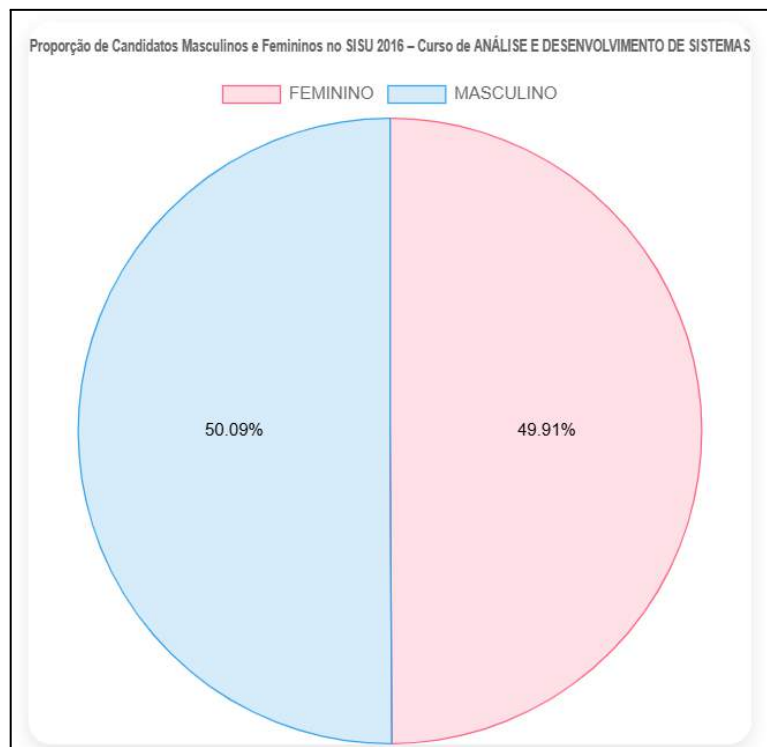
As Figuras 16 e 17 ilustram gráficos de pizza que comparam a proporção de candidatos masculinos e femininos ao curso em 2016 e 2023. Em 2023, conforme mostrado na Figura 10, a concorrência foi de 22% de mulheres e 77% de homens. Já em 2016, na Figura 11, a concorrência era mais equilibrada, com aproximadamente 51% de homens e 49% de mulheres, indicando uma diminuição na participação feminina ao longo dos anos.

Figura 16: Gráfico de pizza do SISU 2023



Fonte: Autor

Figura 17: Gráfico de pizza do SISU 2016

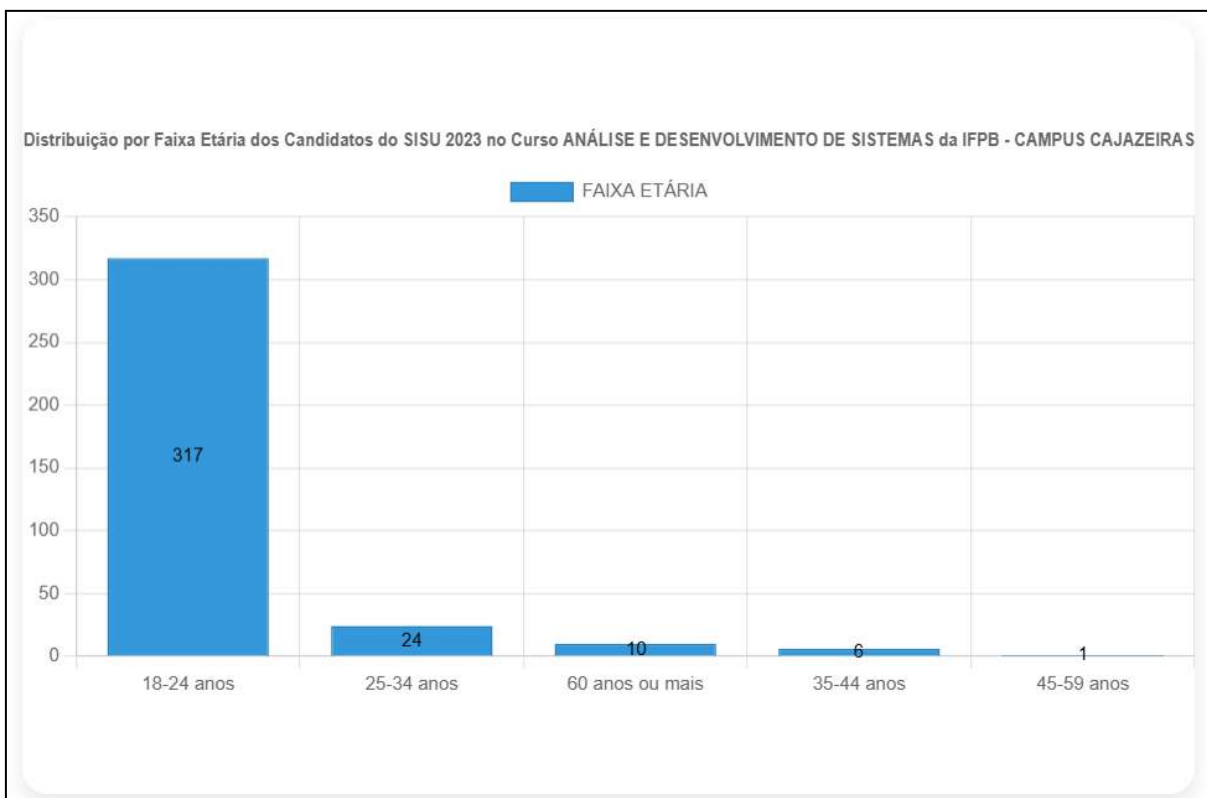


Fonte: Autor

A Figura 18 ilustra um gráfico de barra sobre a distribuição por faixa etária dos candidatos ao curso de Análise e Desenvolvimento de Sistemas do IFPB Campus Cajazeiras no Sisu em 2023, na qual demonstra a predominância de candidatos na faixa etária de 18 a 24 anos, que é de 317 inscritos. Onde a partir disso podemos avaliar que o perfil da maioria dos candidatos ao curso é composto por jovens, que vem do término do ensino médio.

Ademais, observa-se uma participação muito menor de candidatos nas faixa etárias com idades mais avançadas. A faixa de 25 a 34 anos apresentou 24 inscritos, já a faixa de 60 ou mais apresentou 10 e as faixas etárias de 35 e 44 anos e 45-59 anos apresentaram, 6 e 1 candidatos respectivamente. Mostrando que o curso não é tão procurado por pessoas com idades mais avançadas, o que pode indicar a dificuldade de pessoas dessas faixas etárias de se familiarizar e se identificar com as tecnologias e metodologias que tem no curso de Análise e Desenvolvimento de Sistemas.

Figura 18: Distribuição etária dos candidatos ao curso de ADS no SISU 2023



Fonte: Autor

6. CONSIDERAÇÕES FINAIS

O SISU ANALYZER foi planejado para ser uma ferramenta com o objetivo de analisar os comportamentos dos cursos, ofertados pelo Sistema de Seleção Unificada (SISU) em instituições públicas de ensino superior. Com isso, o sistema permite o acesso e a transparência sobre a evolução dos cursos ofertados pelo SISU, ao longo dos anos de 2016 a 2023.

Por meio da identificação dos cursos mais e menos procurados durante os anos, o SISU ANALYZER oferece informações valiosas para estudantes, gestores educacionais e pesquisadores interessados na dinâmica do ensino superior público no Brasil. Esses dados permitem uma análise mais crítica sobre as preferências dos candidatos ao longo do tempo e também podem promover debates que poderão incentivar futuras políticas educacionais.

Então, o sistema não só apenas deixa o processo mais transparente, mas também democratiza o acesso à informação, proporcionando uma compreensão mais aprofundada sobre as características do processo para ingressar no ensino superior público. Portanto, o SISU ANALYZER ajuda no planejamento estratégico dos estudantes como também das universidades, auxiliando a tomar decisões mais informadas e embasadas.

Como sugestão para trabalhos futuros para o aprimoramento da ferramenta proposta neste trabalho. São a automatização do processo de etl e a implementação de mecanismos para fazer atualizações de novos dados. Com essas melhorias o sistema vai se tornar mais otimizado e eficiente, por reduzir a necessidade de intervenção manual e garantindo atualizações de forma automática e precisa.

REFERÊNCIAS

- AUAD, Tássio. **O padrão MVC**. 2014. Disponível em: <https://tassioauad.com/o-padrao-mvc/>. Acesso em: set. 2024.
- AWARI. **O que é Backend e Como Funciona? Guia Completo para Iniciantes**. 2023. Disponível em: <https://awari.com.br/o-que-e-backend-e-como-funciona-guia-completo-para-iniciantes/>. Acesso em: set. 2024.
- BRASIL. **Cartilha LAI: Lei de Acesso à Informação no Brasil**. 2013. Disponível em: <https://www12.senado.leg.br/transparencia/indice-de-transparencia-dos-portais-legislativos/arquivos/sobre/cartilha-lai>. Acesso em: ago. 2024.
- BRASIL. Ministério da Educação. **SISU - Dados Abertos - MEC**. Disponível em: <https://dadosabertos.mec.gov.br/sisu>. Acesso em: out. 2024.
- CERRI, Rodrigo; CARVALHO, André Carlos P. L. F. **Aprendizado de máquina: breve introdução e aplicações**. 2017. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/184785/1/Aprendizado-de-maquina-breve-introducao.pdf>. Acesso em: set. 2024.
- CHART.JS. **Chart.js documentation**. Disponível em: <https://www.chartjs.org/docs/2.9.4/>. Acesso em: fev. 2025.
- COUGHLIN, Thomas. **175 Zettabytes by 2025**. 2018. Disponível em: <https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/?sh=64082fa25459>. Acesso em: jul. 2024.
- GOMES, Dennis dos Santos. **Inteligência artificial: conceitos e aplicações**. 2017. Disponível em: https://www.professores.uff.br/screspo/wp-content/uploads/sites/127/2017/09/ia_intro.pdf. Acesso em: set. 2024.

HORA DE CODAR. **Frontend: o que é, para que serve, cursos, carreira e como aprender.** 2024. Disponível em: <https://horadecodar.com.br/frontend/>. Acesso em: set. 2024.

MITTELBACH, F.; GOOSSENS, M.; BRAAMS, J.; CARLISLE, D.; ROWLEY, C. **The LaTeX companion.** [S.l.]: Addison-Wesley Professional, 2004.

ORACLE. **O que é um banco de dados? | Oracle Brasil.** Disponível em: <https://www.oracle.com/br/database/what-is-database/>. Acesso em: fev. 2025.

PANDAS. **Pandas documentation.** Disponível em: <https://pandas.pydata.org/>. Acesso em: 12 mar. 2025.

POSTGRESQL. **PostgreSQL: documentation 16: 1. What is PostgreSQL?.** 2024. Disponível em: <https://www.postgresql.org/docs/current/intro-what-is.html>. Acesso em: set. 2024.

PYTHON. **O tutorial de Python — Documentação Python Python 3.12.6.** 2024. Disponível em: <https://docs.python.org/pt-br/3/tutorial/index.html>. Acesso em: set. 2024.


PRISMA/DOCS. **Why Prisma ORM? Comparison with SQL query builders & ORMs | Prisma Documentation.** 2024. Disponível em: <https://www.prisma.io/docs/orm/overview/introduction/why-prisma>. Acesso em: set. 2024.

REDATOR. **60% das empresas no Brasil já usam Analytics para apoiar sua estratégia.** 2019. Disponível em: <https://itforum.com.br/60-das-empresas-no-brasil-ja-usam-analytics-para-apoiar-sua-estrategia/>. Acesso em: jul. 2024.

ROSA, Daniel Gervilla. **O uso da ciência e da análise de dados nos processos de tomada de decisão.** 2022. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/11095>. Acesso em: ago. 2024.

WEINER, Jeff. Frase Jeff Weiner: **Os dados realmente impulsionam tudo o que...** 2023. Disponível em:

<https://wdiscover.com.br/frase-jeff-weiner#:~:text=Frases%20Jeff%20Weiner%20%E2%80%93%20Confira%20agora!%20%E2%80%93%20COs%20dados%20realmente%20impulsionam>. Acesso em: jul. 2024.

	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 Restrito

Trabalho de conclusão

Assunto:	Trabalho de conclusão
Assinado por:	Leonardo Monteiro
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Restrito
Hipótese Legal:	Direito Autoral (Art. 24, III, da Lei no 9.610/1998)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Leonardo Mendes Monteiro, DISCENTE (202112010025) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 31/03/2025 18:08:14.

Este documento foi armazenado no SUAP em 31/03/2025. 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: 1441929

Código de Autenticação: 2b43264ac6

