UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MARCOS EDUARDO DE SOUZA
MAURO VICTOR DE ARAUJO NASCIMENTO

FERRAMENTA PARA ANÁLISE DE DESEMPENHO ACADÊMICO

RIO DE JANEIRO
2025

MARCOS EDUARDO DE SOUZA
MAURO VICTOR DE ARAUJO NASCIMENTO

FERRAMENTA PARA ANÁLISE DE DESEMPENHO ACADÊMICO

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. João Carlos Pereira da Silva, D.Sc.

RIO DE JANEIRO
2025

Souza, Marcos Eduardo de
S729f Ferramenta para análise de desempenho acadêmico / Marcos Eduardo de Souza e Mauro Victor de Araujo Nascimento. – Rio de Janeiro, 2025.

71 f.

Orientador: João Carlos Pereira da Silva.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação)- Universidade Federal do Rio de Janeiro, Instituto de Computação, Bacharel em Ciência da Computação, 2025.

1. Plataforma de consulta. 2. Painel. 3. Análise de dados. 4. Ciência da Computação. I. Nascimento, Mauro Victor de Araujo. II. Silva, João Carlos Pereira da (Orient.). III. Universidade Federal do Rio de Janeiro, Instituto de Computação. IV. Título.

MARCOS EDUARDO DE SOUZA

MAURO VICTOR DE ARAUJO NASCIMENTO

FERRAMENTA PARA ANÁLISE DE DESEMPENHO ACADÊMICO

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 17 de Janeiro de 2025.

BANCA EXAMINADORA:

gov.br Documento assinado digitalmente
JOAO CARLOS PEREIRA DA SILVA
Data: 29/01/2025 16:45:13-0300
Verifique em https://validar.iti.gov.br

João Carlos Pereira da Silva
D.Sc. (Instituto de Computação - UFRJ)

gov.br Documento assinado digitalmente
JULIANA VIANNA VALERIO
Data: 07/02/2025 17:22:31-0300
Verifique em https://validar.iti.gov.br

Juliana Vianna Valerio
D.Sc. (Instituto de Computação - UFRJ)

gov.br Documento assinado digitalmente
JULIANA BAPTISTA DOS SANTOS FRANCA
Data: 29/01/2025 16:47:52-0300
Verifique em https://validar.iti.gov.br

Juliana Baptista dos Santos França
D.Sc. (Instituto de Computação - UFRJ)

Dedicamos este trabalho aos nossos pais, cujo esforço e apoio incondicional tornaram possível a realização deste sonho.

Eu, Marcos, dedico também ao meu orientador, Prof. João Carlos Pereira da Silva, D.Sc., por sua orientação incansável e por tornar possível a conclusão desta tarefa tão desafiadora. À minha antiga orientadora do primeiro curso de Matemática Aplicada, profª. Monique Robalo Moura Carmona, D.Sc., pela valiosa orientação e apoio durante minha jornada de permanência e a transferência de curso.

AGRADECIMENTOS

Agradecemos ao Prof. João Carlos Pereira da Silva, D.Sc., pela paciência, orientação e incentivo inestimáveis ao longo deste processo.

Ao SIGA, por disponibilizar os dados anonimizados que permitiram o desenvolvimento mais realista do sistema.

Eu, Marcos, gostaria de agradecer aos meus pais e irmãos, por todo o apoio, incentivo nos momentos difíceis e compreensão diante das minhas ausências enquanto me dedicava à realização deste trabalho.

Eu, Mauro, agradeço à minha mãe e meu pai, bem como à minha noiva, Carla, meus irmãos e sobrinhos, além de meu melhor amigo, Jefferson, por estarem comigo durante todo o meu desenvolvimento na universidade e por zelarem por mim, alegrando minha vida.

Aos nossos colegas de curso, com quem compartilhamos momentos intensos ao longo dos últimos anos. O companheirismo e a troca de experiências foram fundamentais para o nosso crescimento, tanto pessoal quanto profissional.

Por fim, estendemos nossa gratidão a todos que, de alguma forma, contribuíram para a realização deste trabalho, seja com palavras de apoio, ensinamentos ou gestos de solidariedade.

"I choose a lazy person to do a hard job.
Because a lazy person will find an easy way to do it."

Bill Gates

RESUMO

O estudo em questão busca a construção de uma plataforma voltada à análise de dados acadêmicos na Universidade Federal do Rio de Janeiro (UFRJ). A iniciativa surge da necessidade de facilitar o acompanhamento de rendimento dos alunos, e substituir métodos manuais para a consulta de médias e notas, os quais demandam um esforço significativo por parte dos professores, que se veem obrigados a examinar minuciosamente os boletins individuais de cada discente. O escopo do projeto consiste em tratar dados e disponibilizar uma plataforma de consulta ágil, capacitando os docentes a acessar o histórico de registros das disciplinas e dos alunos, auxiliando, assim, a tomada de decisões embasadas em informações precisas e acessíveis.

Palavras-chave: plataforma; painel; análise de dados; Ciência da Computação; UFRJ; plataforma de consulta.

ABSTRACT

This study aims to develop a platform for analyzing academic data for the Federal University of Rio de Janeiro (UFRJ). The initiative arises from the need to replace manual methods for consulting students' grades and averages, which require significant effort from professors who must meticulously examine each student's individual report cards. Such a procedure hinders comprehensive student monitoring and limits the thorough analysis of course data, aimed at its continuous improvement. The project's scope involves processing data and providing an agile consultation platform, enabling faculty members to access historical records of courses and students, thereby facilitating decision-making based on precise and readily accessible information.

Keywords: platform; data analysis; dashboard; Computer Science; UFRJ; consultation platform.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CR	Coeficiente de Rendimento
CRA	Coeficiente de Rendimento Acumulado
SIGA	Sistema Integrado de Gestão Acadêmica
UFRJ	Universidade Federal do Rio de Janeiro
DRE	Divisão De Registro De Estudantes
SiSU	Sistema de Seleção Unificada
ENEM	Exame Nacional do Ensino Médio
COAA	Comissão de Orientação e Acompanhamento Acadêmico
CPO	Corpo de Professores Orientadores
BCC	Bacharelado em Ciências da Computação
IC	Instituto de Computação
BOA	Boletins de Orientação Acadêmica
SQL	Structured Query Language
DW	Data Warehouse
BI	Business Intelligence

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO DO PROBLEMA

O acompanhamento do rendimento acadêmico de estudantes universitários é um desafio recorrente em diversos cursos. Professores e coordenações frequentemente necessitam de informações detalhadas sobre o desempenho dos alunos para embasar estratégias pedagógicas, identificar estudantes em situação de risco acadêmico e implementar ações de suporte mais efetivas. Entretanto, o acesso a esses dados, na maioria das vezes, envolve processos manuais, como a consulta de boletins individuais e a compilação de informações em planilhas, o que demanda esforço significativo e está sujeito a erros. Além disso, as universidades enfrentam outro problema significativo: a evasão dos alunos. Muitas vezes associada a diversos fatores como dificuldades no aprendizado e falta de suporte institucional. Um estudo realizado por PRESTES; FIALHO, 2018, estimou que, em 2018, as evasões geraram prejuízo de aproximadamente R$ 611.302,03 em apenas um campus da Universidade Federal da Paraíba, evidenciando os danos decorridos, que podem ser financeiros e educacionais. Portanto, a implementação de uma solução que consolide e permita a visualização e interpretação de dados acadêmicos de forma automatizada é uma necessidade latente.

No contexto específico do curso de Bacharelado em Ciência da Computação da UFRJ, de acordo com GONÇALVES; FILHO; GOMES, 2023, na coordenação do curso, 80% dos professores relataram problemas com a visualização de informações de desempenho acadêmico em 2023. A partir dessa informação, a equipe citada projetou uma ferramenta onde pudesse ser realizado um acompanhamento de alunos, para uso de orientadores, onde pudessem ser visualizados dados de rendimentos acadêmicos individuais. No entanto, para obter uma visão geral de todos os alunos do curso ou grupos de alunos, seria necessária uma visão que consolide todas as informações dos alunos, e, também, dados de disciplinas. Atualmente, para acessar informações detalhadas do rendimento acadêmico de alunos, os professores precisam acessar o site da instituição, baixar os boletins de cada estudante individualmente e consolidar as informações manualmente, um processo demorado e suscetível a falhas. Esse problema não se limita a um único departamento, pois as coordenações da UFRJ utilizam o Sistema Integrado de Gestão Acadêmica (SIGA), que ainda não possui visualizações para análise dos dados acadêmicos. Tal lacuna compromete a eficiência do acompanhamento acadêmico e pode impactar negativamente a experiência dos estudantes.

Sem uma plataforma de suporte para análise de dados, é comprometida a capacidade dos professores de identificar alunos em situação de risco, seja por jubilamento ou por necessidade de maior apoio pedagógico. Como resultado, intervenções que deveriam ocorrer

de forma precoce podem ser adiadas, contribuindo para o aumento dos índices de reprovação e evasão. Somado a isso, tal ferramenta poderia auxiliar a coordenação do curso no embasamento para tomadas de decisões, uma vez que a clareza sobre o histórico dos estudantes, com a identificação de padrões de desempenho, poderiam orientar mudanças na grade curricular, no planejamento de disciplinas e nas atividades de reforço.

Esse projeto busca suprir essa demanda ao desenvolver uma ferramenta que facilite o acesso às informações de desempenho, possibilitando que os professores identifiquem, com agilidade, quais alunos necessitam de maior atenção. Ao fornecer uma visão integrada do progresso acadêmico nas disciplinas, espera-se reduzir os índices de reprovação e abandono, promovendo uma melhor experiência acadêmica e resultados mais satisfatórios tanto para os alunos quanto para a instituição.

Em MILLECAMP et al., 2018, projeto de uma equipe belga, foi desenvolvido um painel para que tutores da universidade pudessem avaliar os resultados acadêmicos anuais de seus alunos, com o objetivo de auxiliar casos de risco de reprovação e idealizar um plano de estudos para o ano seguinte. Nessa ferramenta, são mostrados dados específicos de cada aluno durante seu primeiro ano na universidade, e há indicadores comparando o rendimento do mesmo com outros colegas, além de modelos preditivos que definem um perfil para o aluno e buscam compreender quais são as matérias em que pode haver dificuldade no futuro. Essa também é uma plataforma para uso de orientadores, e que cria análises comparativas simples de notas, porém, sem trazer uma visão que permitiria a compreensão de todo o contexto de um curso de graduação.

Outra iniciativa que utilizou painéis de análise de dados acadêmicos foi SILVA; NETTO; SOUZA, 2016, com o objetivo de compreender como o uso dessas ferramentas pode contribuir para o rendimento dos alunos. No software são registrados os resultados das avaliações feitas em sala pelos alunos, para que o professor possa acompanhá-los, visualizando as taxas de acerto, índices de aprovação e informações sobre as questões usadas nas avaliações, agrupadas por dificuldade. Desse modo, após o uso da página com gráficos relevantes para observação, foi concluído que a ferramenta contribuiu positivamente para que o professor compreenda de modo objetivo quais são os aspectos de maior dificuldade dos alunos, e veja casos em que são necessárias intervenções de suporte educacional.

Diante disso, nosso projeto tem como objetivo desenvolver um painel de dados para facilitar o acesso e a visualização de informações acadêmicas de forma rápida, intuitiva e integrada. Essa solução proporciona uma interface interativa e visualmente informativa, que exibirá gráficos e indicadores relevantes sobre o rendimento acadêmico dos alunos. Contudo, para que esses dados estejam disponíveis para uso, parte do projeto envolve a aplicação de Engenharia de Dados, com foco no processamento e tratamento das informações, garantindo que os dados estejam consistentes e prontos para análise. Esse processo é fundamental para assegurar a integridade e a confiabilidade dos dados exibidos na plataforma. A ausência de dados bem tratados comprometeria a qualidade das visualizações

e poderia levar a interpretações incorretas.

Com esse foco, a finalidade é criar uma ferramenta de visualização e estruturar um fluxo eficiente para a atualização e manutenção dos dados, permitindo que a plataforma evolua ao longo do tempo, absorvendo novos dados e permanecendo como um recurso confiável para a equipe acadêmica. Dessa forma, o sistema poderá se tornar um ponto central no acompanhamento do desempenho dos alunos, fornecendo informações precisas e atualizadas.

A monografia está organizada da seguinte forma: no capítulo 2, apresentamos a base teórica da implementação realizada, com explicações das tecnologias e processos utilizados. No capítulo 3, detalhamos o trabalho de engenharia de dados realizado para transformar o material recebido em informações consistentes para análise. Somado a isso, no capítulo 4, apresentamos a implementação do sistema, mostrando o resultado obtido. Concluímos o artigo no capítulo 5, onde discutimos a situação posterior à finalização do software e propomos melhorias para o futuro do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ENGENHARIA DE DADOS

A Engenharia de Dados é o campo responsável pelo planejamento, desenvolvimento e manutenção de arquiteturas de dados, possibilitando que informações sejam extraídas, transformadas e carregadas para análises e suporte à tomada de decisão KIMBALL; ROSS, 2013. Esse processo permite transformar dados brutos em informações úteis, especialmente em sistemas que visam a tomada de decisão, como o desenvolvido neste trabalho.

Com a aplicação correta, é possível estruturar, organizar e otimizar informações ao longo de todas as etapas do ciclo de vida dos dados, desde a coleta e processamento até a análise. Essa disciplina é importante para garantir que os dados estejam acessíveis, consistentes e confiáveis.

2.2 CICLO DE VIDA DA ENGENHARIA DE DADOS

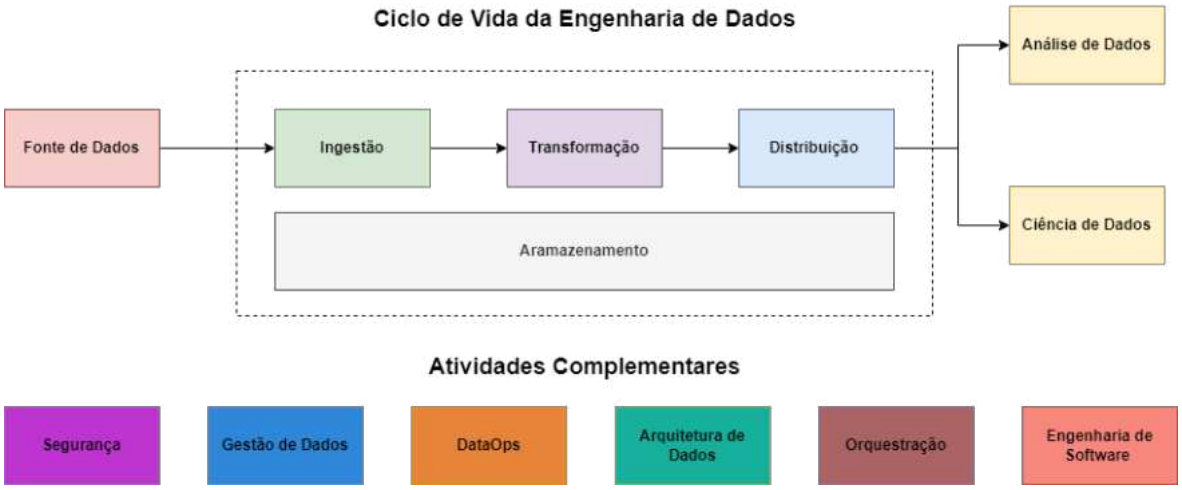
O ciclo de vida da Engenharia de Dados é um processo abrangente que engloba várias etapas, desde a ingestão de dados até sua distribuição para análises e aplicações em ciência de dados. Cada uma dessas fases desempenha um papel importante no preparo dos dados para diferentes tipos de aplicações, como análises de dados e aprendizado de máquina REIS; HOUSLEY, 2022.

Segundo INMON, 2005, a organização e o gerenciamento do ciclo de vida dos dados são indispensáveis, pois asseguram a consistência e a acessibilidade das informações ao longo do tempo. Esse gerenciamento vai além do simples armazenamento de dados, abrangendo práticas de governança e orquestração que garantem a eficiência do sistema de dados.

A Figura 1 ilustra o ciclo de vida da Engenharia de Dados, destacando as fases principais e o armazenamento que serve como base para esse ciclo. Além disso, estão representadas as **atividades complementares** que sustentam cada fase, tais como segurança, gestão de dados, arquitetura de dados, orquestração e engenharia de software. Essas atividades complementares são componentes que asseguram a integridade, governança e qualidade dos dados durante todo o processo. Cada fase do ciclo é descrita a seguir:

- **Fonte de Dados**: As origens dos dados podem incluir sistemas transacionais, APIs, sensores, bancos de dados ou outras fontes que fornecem informações brutas para o ciclo de engenharia de dados.
- **Ingestão**: Fase inicial em que os dados são coletados e carregados para um ambiente de armazenamento, seja em seu estado bruto ou pré-processado, garantindo a integridade das informações ao longo do transporte.

Figura 1 – Ciclo de Vida da Engenharia de Dados



Fonte: Adaptado de REIS; HOUSLEY, 2022

- **Transformação**: Nesta etapa, os dados são processados, limpos, integrados e convertidos em formatos úteis e padronizados, visando atender às necessidades específicas de análise e consumo.
- **Distribuição**: Após a transformação, os dados são disponibilizados para os consumidores finais ou sistemas, como dashboards, relatórios, ou ferramentas analíticas, assegurando acessibilidade e usabilidade.
- **Armazenamento**: Serve como base de sustentação do ciclo, oferecendo infraestrutura para guardar os dados em diferentes formatos ou camadas, como bruta, processada e analítica.
- **Análise de Dados**: Utiliza os dados transformados para explorar padrões, gerar insights e apoiar a tomada de decisões informadas.
- **Ciência de Dados**: Fase avançada que aplica modelos estatísticos e algoritmos de aprendizado de máquina para realizar previsões e otimizações baseadas nos dados disponíveis.

Além dessas etapas principais, o ciclo de vida é apoiado por atividades complementares que desempenham papéis importantes:

- **Segurança**: Garante a proteção e confidencialidade dos dados, protegendo contra acessos não autorizados e garantindo conformidade com regulamentos.
- **Gestão de Dados**: Envolve práticas de governança, definição de políticas e controle de qualidade dos dados.

- **DataOps**: Promove automação, integração e entrega contínua em pipelines de dados, aumentando a eficiência operacional.
- **Arquitetura de Dados**: Define e organiza a infraestrutura necessária para suportar e otimizar o fluxo de dados.
- **Orquestração**: Coordena e monitora os processos e fluxos de dados entre diferentes etapas do ciclo.
- **Engenharia de Software**: Apoia o ciclo com ferramentas e aplicações que viabilizam a execução e a manutenção do sistema de dados.

2.3 ETL (EXTRAÇÃO, TRANSFORMAÇÃO E CARREGAMENTO)

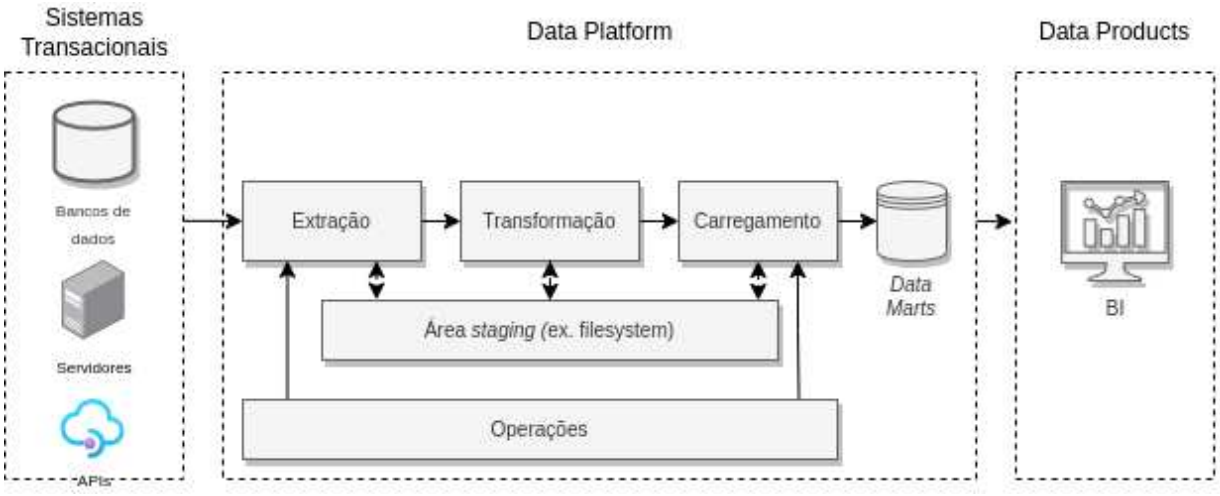
ETL é a junção das siglas em inglês *Extract*, *Transform* e *Load*, que em português significam Extração, Transformação e Carregamento. O processo de ETL é amplamente reconhecido como a espinha dorsal das operações de armazenamento e análise de dados, pois permite a conversão de dados brutos em informações prontas para consulta e análise VASSILIADIS et al., 2009. Cada uma das etapas do ETL desempenha um papel fundamental no fluxo de dados:

- **Extração (*Extract*)**: Consiste em recuperar dados brutos de uma ou mais fontes, como bancos de dados, arquivos, APIs ou fluxos de dados em tempo real, consolidando-os em um repositório de dados centralizado.
- **Transformação (*Transform*)**: Envolve a limpeza, estruturação, enriquecimento e conversão dos dados para adequá-los a um modelo de dados final. Nessa etapa, são realizados processos de normalização, agregação e padronização, visando garantir a consistência e a qualidade dos dados.
- **Carregamento (*Load*)**: Refere-se ao carregamento dos dados transformados em um *data warehouse* ou repositório de dados final, onde estarão disponíveis para uso em ferramentas de *Business Intelligence* (BI), análise de dados ou outras aplicações.

Além de organizar as etapas de manipulação de dados, o processo de ETL define a sequência lógica em que essas etapas são executadas para garantir a qualidade das informações.

A Figura 2 ilustra o processo de ETL em uma plataforma de dados completa. Os dados são extraídos de sistemas transacionais, como bancos de dados, servidores e APIs, e movidos para uma área de *staging*, que atua como um local de armazenamento temporário para dados brutos ou minimamente processados, permitindo o controle e auditoria dos dados antes de serem transformados e carregados.

Figura 2 – Exemplo de processo de ETL.



Fonte: https://www.engdeanalytics.com.br/chapters/13/01/etl.html

2.4 ARQUITETURA DE DADOS

A arquitetura de dados é responsável por definir a estrutura, organização e o fluxo de informações dentro de uma plataforma de dados, garantindo que os dados sejam armazenados, acessados e processados de maneira eficiente. Com o crescimento exponencial do volume, variedade e velocidade dos dados, o uso de um *Data Lake* tem se tornado uma solução essencial dentro dessa arquitetura. Um *Data Lake* é um repositório centralizado que permite armazenar dados estruturados e não estruturados em seu formato bruto, bem como em diferentes estados de processamento. Essa abordagem oferece flexibilidade para consolidar dados de diversas fontes em um único local escalável, possibilitando análises avançadas e transformações subsequentes de maneira mais eficiente REIS; HOUSLEY, 2022.

No contexto deste projeto, o termo *Data Lake* é utilizado para descrever a organização dos dados em camadas (*Bronze*, *Silver* e *Gold*) dentro de uma estrutura de pastas local. Embora soluções corporativas de *Data Lake* geralmente utilizem sistemas escaláveis e distribuídos, como *Amazon S3* ou *Azure Data Lake Storage*, a implementação adotada atende aos objetivos deste trabalho, que é armazenar e processar os dados de forma centralizada e organizada.

2.4.1 Arquitetura em Camadas

A arquitetura em camadas organiza o *Data Lake* em diferentes estágios de transformação dos dados, refletindo níveis crescentes de refinamento. Essa divisão permite um fluxo de dados mais eficiente e facilita a governança, qualidade e acessibilidade dos dados ao longo de todo o processo analítico KIMBALL; ROSS, 2013; REIS; HOUSLEY, 2022.

Abaixo, detalhamos cada uma dessas camadas:

- **Camada *Landing***: É o ponto de entrada dos dados no *Data Lake*, onde os dados brutos são armazenados no seu formato original. Nesta camada, os dados são recebidos diretamente das fontes (bancos de dados transacionais, APIs, etc.) e não sofrem qualquer transformação. A camada *Landing* garante a integridade e preservação dos dados originais, funcionando como uma referência para futuras auditorias ou reprocessamentos.

- **Camada *Bronze***: Após a ingestão, os dados são movidos para a camada Bronze, onde passam por uma organização inicial para facilitar a consulta. Embora os dados ainda estejam em sua forma bruta, eles podem ser levemente estruturados ou particionados para simplificar o acesso. A camada Bronze serve como uma cópia dos dados originais, preservando-os para auditoria e mantendo a rastreabilidade.

- **Camada *Silver***: Nesta etapa, os dados são transformados, limpos e enriquecidos, eliminando duplicatas e corrigindo inconsistências. A camada *Silver* representa dados mais refinados, prontos para análises intermediárias ou aplicações operacionais. Nessa fase, são realizadas transformações mais profundas, como padronização dos tipos das colunas e tratamento de valores nulos, melhorando a qualidade dos dados.

- **Camada *Gold***: A camada *Gold* é destinada aos dados prontos para consumo, otimizados para análise final e geração de relatórios. Aqui, os dados estão totalmente transformados e organizados para atender às necessidades de negócios, sendo usados em aplicações de Inteligência de Negócios (*Business Intelligence - BI*), visualizações de dados e aprendizado de máquina. A camada *Gold* fornece acesso rápido e direto aos dados, maximizando a eficiência das consultas e permitindo uma tomada de decisão informada e ágil.

2.5 DATA WAREHOUSING E MODELAGEM DIMENSIONAL

O conceito de *Data Warehousing* surgiu como uma solução para a necessidade de consolidar e organizar grandes volumes de dados provenientes de diversas fontes, visando proporcionar uma estrutura eficiente para consulta e análise de informações INMON, 2005. Um *Data Warehouse* (DW) é um repositório centralizado que armazena dados de forma integrada, não volátil e orientada por assuntos, possibilitando que analistas e tomadores de decisão acessem informações consistentes e históricas. A estruturação dos dados em um DW permite consultas e relatórios mais rápidos, além de facilitar a geração de *insights* por meio de ferramentas de visualização.

Na construção de um *Data Warehouse*, a Modelagem Dimensional é uma abordagem popular que visa simplificar o acesso e análise de dados. Essa técnica é focada na criação

de estruturas que facilitem o entendimento e a navegação pelos dados, organizando-os de maneira a otimizar o desempenho das consultas. A Modelagem Dimensional é particularmente útil para cenários de análise de dados em grandes volumes, onde a velocidade de resposta das consultas é crítica KIMBALL; ROSS, 2013.

2.5.1 Modelagem Dimensional

A modelagem dimensional é baseada na criação de um esquema estrela (ou *Star Schema*) ou esquema floco de neve (ou *Snowflake Schema*), onde os dados são organizados em torno de duas principais estruturas: tabelas de fatos e tabelas de dimensão.

- **Tabelas de Fatos**: Essas tabelas armazenam dados quantitativos ou eventos específicos que ocorrem em um processo de negócio, como transações de vendas, pontuações de testes ou registros de matrículas. Cada linha em uma tabela de fatos representa uma ocorrência mensurável, associada a um conjunto de métricas numéricas (fatos) e chaves que conectam a tabela de fatos às tabelas de dimensão.
- **Tabelas de Dimensão**: As tabelas de dimensão contêm dados descritivos e contextuais que fornecem detalhes sobre os fatos. Elas incluem detalhes como nomes, datas, categorias e outras características que ajudam a segmentar os dados para análise.

No Modelo Estrela (*Star Schema*), as tabelas de dimensão estão diretamente conectadas à tabela de fatos, formando uma estrutura visual semelhante a uma estrela, como pode ser observado na Figura 3. Esse modelo é simples e de fácil compreensão, uma vez que minimiza o número de junções (*joins*) necessários para realizar consultas, o que o torna ideal para cenários onde a velocidade de consulta é uma prioridade.

Por exemplo, no cenário da Figura 3, a tabela de fatos *Vendas* é conectada diretamente a dimensões como *Data*, *Região*, *Revendedor*, *Funcionário*, e *Produto*. Essa organização permite consultas rápidas para analisar vendas por categoria de produto, cliente ou região.

Em contraste, o Modelo Floco de Neve (*Snowflake Schema*) expande a estrutura do Esquema Estrela ao normalizar as tabelas de dimensão, dividindo-as em tabelas menores para evitar redundância. Essa abordagem forma uma estrutura mais complexa e ramificada, como mostrado na Figura 4.

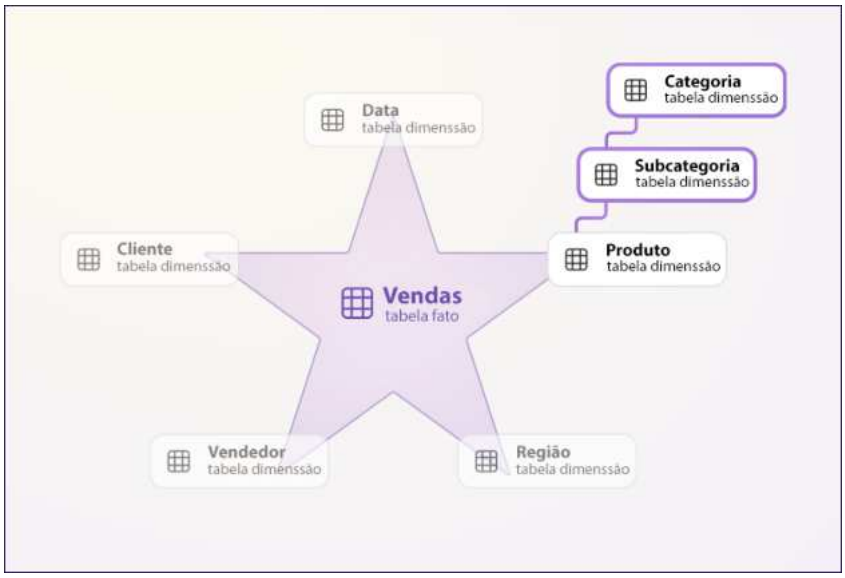
Na Figura 4, a dimensão *Produto* é dividida em subdimensões como *Subcategoria* e *Categoria*. Embora esse modelo reduza o espaço de armazenamento, ele exige mais junções (*joins*) para consultas, o que pode impactar o desempenho em grandes volumes de dados.

Neste projeto, optamos por utilizar o modelo estrela devido à sua simplicidade e fácil manutenção. Além disso, buscamos oferecer um sistema acessível e prático que funcione como uma ferramenta de apoio para a coordenação do curso, possibilitando análises rápidas e intuitivas sobre o desempenho acadêmico.

Figura 3 – Exemplo de Esquema Estrela.

Fonte: Adaptado de https://learn.microsoft.com/pt-br/power-bi/guidance/star-schema

Figura 4 – Exemplo de Esquema Floco de Neve.



Fonte: Adaptado de https://learn.microsoft.com/pt-br/power-bi/guidance/star-schema

3 DESENVOLVIMENTO DA ENGENHARIA DE DADOS

Este capítulo apresenta o desenvolvimento do *backend* do projeto, detalhando o ciclo de vida dos dados e a modelagem dimensional de um *Data Warehouse* (DW). O objetivo principal é demonstrar a viabilidade da execução dos processos de ETL utilizando ferramentas de software livre.

3.1 ARQUITETURA DE DADOS

A arquitetura de dados do projeto foi projetada com o objetivo de garantir o fluxo eficiente das informações, desde a ingestão até a análise final. A Figura 5 apresenta a estrutura geral, composta por diferentes camadas de armazenamento e ferramentas para manipulação dos dados, além de soluções para orquestração e governança de dados.

A arquitetura é dividida em quatro partes principais:

- **Fonte de Dados:** Os dados são obtidos por meio do Sistema Integrado de Gestão Acadêmica (SIGA) da UFRJ, em formato XLSX, fornecidos pela equipe da UFRJ de forma anonimizada.

- **Ingestão de Dados:** A ingestão dos dados ocorre manualmente. A planilha é carregada pelo usuário na camada *Landing*, onde os dados brutos são disponibilizados para o início do processamento.

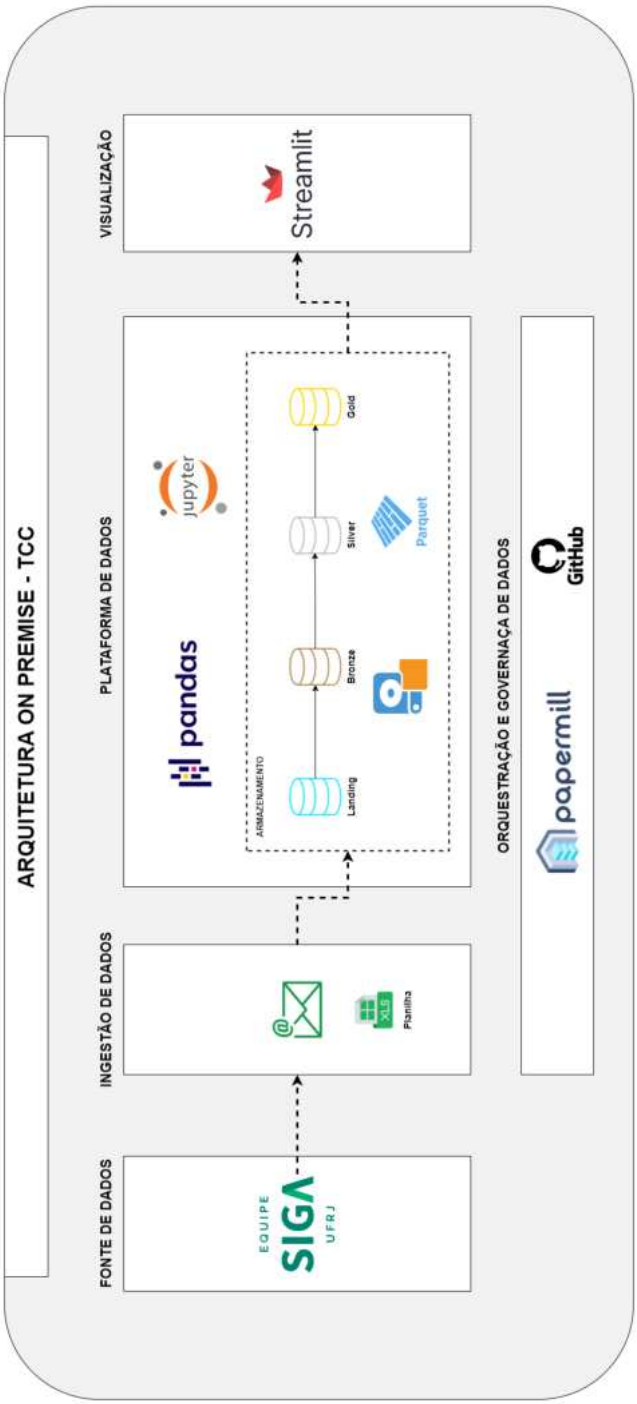
- **Plataforma de Dados:** O processamento dos dados ocorre em quatro camadas: *Landing*, Bronze, *Silver* e *Gold*, cada uma com um papel específico no ciclo de vida dos dados. Utilizamos as bibliotecas Pandas[1] e Jupyter[2] para o processamento, enquanto o armazenamento dos dados é realizado no formato *Parquet*, um formato de arquivo baseado em colunas, muito eficiente na compressão, permitindo armazenar mais dados em menos espaço. O armazenamento é realizado localmente no computador.

- **Inteligência de Negócios:** Após o processamento completo, os dados da camada *Gold* são utilizados na interface de visualização desenvolvida com *Streamlit*[3], uma biblioteca em Python que permite criar aplicações web interativas e personalizáveis, com foco em dados e visualizações.

- **Orquestração e Governança de Dados:** As ferramentas *Papermill* e *GitHub* são utilizadas para automação e governança dos processos de ETL. O *Papermill* per-

[1] https://pandas.pydata.org
[2] https://jupyter.org
[3] https://streamlit.io/

Figura 5 – Arquitetura de Dados do Projeto.



Fonte: Elaborado pelo autor.

mite a execução parametrizada de notebooks *Jupyter*, facilitando a orquestração dos fluxos de trabalho e a automação de pipelines de dados. Já o *GitHub* é empregado para o versionamento do código e a rastreabilidade das alterações, garantindo controle colaborativo sobre o desenvolvimento. É importante ressaltar que, no uso do *GitHub*, nenhum dado sensível foi exposto no repositório, sendo armazenado apenas

o código de tratamento e manipulação dos dados.

3.2 MODELAGEM DE DADOS

Conforme discutido na Seção 2.5.1, este projeto adota o modelo estrela para a estruturação dos dados. Para garantir consistência e clareza no manuseio das informações,foram implementadas convenções padronizadas na nomenclatura de colunas e tabelas.

Os nomes das colunas seguem o padrão `<Prefixo_NomeObjeto>`, no qual o prefixo identifica o tipo de dado armazenado, como exemplificado na Tabela 1, onde o prefixo indica o tipo de dado armazenado. Por sua vez, os nomes das tabelas também seguem uma convenção similar, com o prefixo `d_` para tabelas dimensão e `f_` para tabelas fato, conforme ilustrado na Figura 6. Um exemplo completo da aplicação dessas nomenclaturas na modelagem de dados pode ser encontrado no Apêndice D.

Tabela 1 – Convenções de Nomeação das Colunas

Abreviatura	Categoria	Descrição	Exemplo de Uso
NU	Número	Representa quantidades, identificadores ou categorias numéricas.	`NU_Idade`, `NU_Quantidade`
TP	Tipo	Armazena valores categóricos pré-definidos, como estados ou status.	`TP_Status`, `TP_Categoria`
DT	Data	Armazena valores do tipo data.	`DT_Nascimento`, `DT_Cadastro`
DS	Descrição	Armazena textos descritivos.	`DS_Produto`, `DS_Observacao`
VL	Valor	Armazena valores numéricos, possivelmente com casas decimais.	`VL_Preco`, `VL_Desconto`
CD	Código	Armazena valores inteiros, geralmente usados para chaves ou atributos numéricos.	`CD_Produto`, `CD_Cliente`
NM	Nome	Representa nomes de objetos, pessoas, eventos, etc.	`NM_Cliente`, `NM_Cidade`
SK	Chave Substituta (*Surrogate Key*)	É um identificador único em uma tabela.	`SK_Cliente`, `SK_Cidade`

Fonte: Adaptado de https://blog.fabianobento.com.br/2011/09/17/padroes-para-nomenclatura-em-um-banco-de-dados

Conforme ilustrado na Figura 6, foram desenvolvidas três tabelas fato e cinco dimen-

sões para atender às necessidades analíticas do projeto. A tabela fato *F_SITUACAO_MATRICULA* tem como objetivo registrar e analisar os eventos relacionados à situação da matrícula dos alunos. Já a tabela fato *F_DESEMPENHO_SEMESTRE* foi projetada para monitorar o desempenho acadêmico ao longo dos semestres, considerando métricas como CR e CRA. Por fim, a tabela fato *F_DESEMPENHO_ACADEMICO* foca na análise detalhada do desempenho dos alunos em cada disciplina cursada, proporcionando uma visão granular dos resultados acadêmicos.

Já para as dimensões, temos a dimensão *D_ALUNO*, que armazena registros únicos de cada aluno com suas informações pessoais, como nome e data de nascimento. A dimensão *D_SITUACAO* contém todas as situações possíveis relacionadas à matrícula e ao status das disciplinas, como "matriculado", "aprovado" ou "reprovado". A dimensão *D_PERIODO* registra todos os períodos já cursados pelos alunos, permitindo análises temporais do desempenho acadêmico. Por fim, a dimensão *D_CURSO* centraliza os dados dos cursos de graduação, incluindo aqueles que os alunos estão matriculados ou já frequentaram.

Um aspecto interessante desta modelagem é a utilização da tabela fato *F_SITUACAO_MATRICULA*, que se enquadra na categoria de tabelas fato sem fato (*Factless Fact Tables*). Esse tipo de tabela não contém métricas ou valores numéricos, apenas chaves estrangeiras que fazem referência a dimensões. Elas são projetadas para rastrear eventos ou ocorrências que não exigem agregações numéricas, mas que são essenciais para registrar mudanças no estado de uma entidade ao longo do tempo.

No caso deste projeto, complementando o que já foi mencionado, a tabela *F_SITUACAO_MATRICULA* é utilizada para registrar a situação acadêmica dos alunos em diferentes períodos, sem a necessidade de métricas numéricas associadas. Um exemplo típico de tabela fato sem fato em outros contextos seria o registro de atendimentos a estudantes em uma faculdade, onde não há métricas evidentes a serem armazenadas, apenas o rastreamento de ocorrências. Optamos por essa modelagem para facilitar a legibilidade e a simplicidade das consultas SQL, garantindo uma estrutura mais clara e eficiente para análise dos dados.

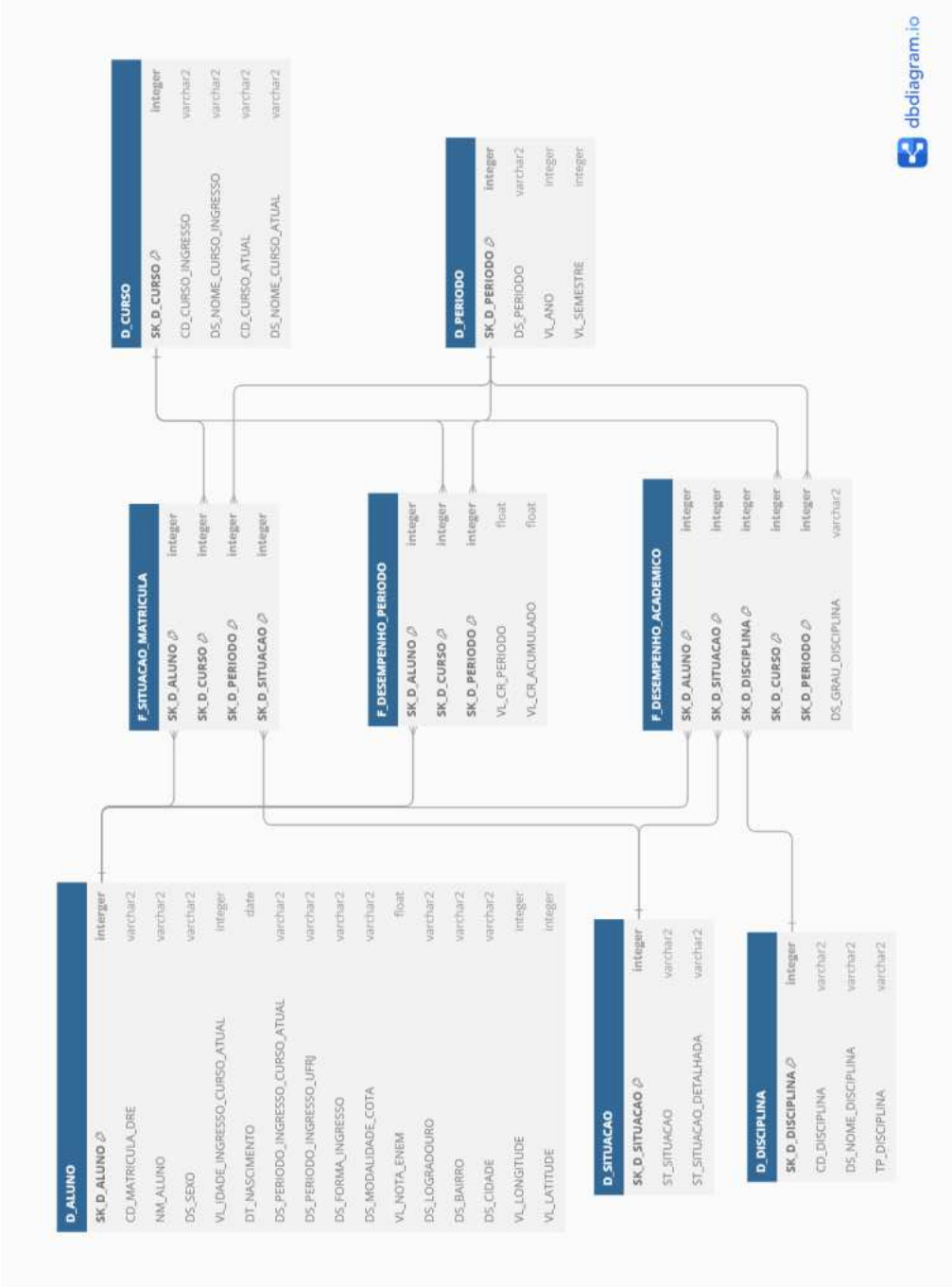
3.3 CICLO DE VIDA DOS DADOS E PROCESSO ETL

A seguir, apresentamos como cada uma das etapas do ciclo de vida e do ETL é implementada neste projeto.

3.3.1 Ingestão e Extração de Dados

Na fase de ingestão, os dados foram fornecidos em formato de planilha XLSX pela equipe do SIGA, conforme apresentado na Figura 5 e detalhado no Apêndice A. O arquivo, contendo 2.705 registros, contempla o período de 2000/1 a 2021/2 e foi entregue já

Figura 6 – Modelagem Star Schema



Fonte: Elaborado pelo autor.

anonimizado, contendo informações sobre os alunos e as disciplinas cursadas. Não houve uma extração ativa via *API* ou banco de dados por parte da equipe, pois os dados foram diretamente disponibilizados pela equipe do SIGA.

As colunas presentes no arquivo incluem dados como semestre de ingresso, curso atual e de ingresso, situação da matrícula, nota no ENEM, informações de cotas, além de históricos de desempenho acadêmico, como coeficiente de rendimento e lista de disciplinas

cursadas. Essas informações são representadas como uma fotografia da situação acadêmica dos alunos no momento em que a planilha foi cedida, refletindo o estado atual dos vínculos e desempenhos registrados.

É importante destacar que, em uma ingestão típica, as informações são inicialmente armazenadas de forma desagregada e tratadas individualmente, sendo agregadas apenas na etapa final, de acordo com a modelagem dimensional. No entanto, neste caso, os dados já foram entregues em formato agregado desde a ingestão. Como resultado, grande parte do tratamento de dados será realizada na camada *Gold*, onde os dados são refinados e organizados para análise final, conforme será discutido na próxima seção 3.3.2.

3.3.2 Transformação dos Dados

Os dados ingeridos, em sua forma bruta, foram inicialmente carregados na camada de *Landing*. Nesta etapa, os dados foram armazenados como ponto de chegada e convertidos para o formato *Parquet*. Após essa conversão, os dados foram transferidos para a camada Bronze, e os arquivos originais foram deletados da camada *Landing* para liberar espaço.

Na camada Bronze, os dados foram mantidos inalterados, funcionando como uma cópia fiel do arquivo original para fins de auditoria e preservação de integridade. Essa camada serve como um ponto de referência confiável para futuras verificações, garantindo que qualquer transformação possa ser rastreada até o dado bruto original.

Após essa etapa, os dados passaram para a camada *Silver*, onde foram realizados tratamentos gerais para melhorar sua qualidade, assegurando que estivessem prontos para análises mais avançadas. Como os dados originais eram anonimizados, foi necessário enriquecer o conjunto de dados com a adição de novas colunas, utilizando diferentes abordagens. As colunas adicionadas e suas respectivas descrições são apresentadas a seguir:

- **Nome do Aluno:** Utilizamos a biblioteca Python *Faker*[4] para gerar nomes fictícios. Essa biblioteca cria objetos aleatórios, como nomes, que foram utilizados neste projeto.
- **DRE:** O número de DRE foi gerado levando em consideração o formato padrão "1YYSSXXXXX", onde:
  - "1" é o prefixo fixo;
  - "YY" representa os dois últimos dígitos do ano de ingresso;
  - "SS" indica o semestre (ajustado para 0 ou 1);
  - "XXXXX" são cinco dígitos únicos gerados aleatoriamente.

[4] https://faker.readthedocs.io/

- **Idade no Curso Atual:** Calculada com base na diferença entre a data de nascimento do aluno e o ano de ingresso no curso, refletindo o tempo decorrido desde o início dos estudos.

- **Endereço:** Foi desenvolvida uma API que realiza requisições ao site *4Devs*[5], afim de realização solicitações de endereços reais do Rio de Janeiro para preenchimento dessa coluna.

- **Latitude e Longitude:** A partir dos endereços gerados pelo *4Devs*, utilizamos a biblioteca *GeoPy*[6] para obter as coordenadas geográficas (latitude e longitude), permitindo análises demográficas e visuais no contexto do projeto.

Um ponto importante a ser mencionado é que a geração de latitude e longitude foi baseada no bairro do aluno. Como muitos alunos podem residir no mesmo bairro, seria ineficiente realizar requisições repetidas para o mesmo local. Para otimizar o processo, foi implementado um mapeamento dos bairros já solicitados, armazenado na própria camada *Silver*, funcionando como um banco de dados de localização.

O fluxo seguiu a lógica a seguir:

- **Verificação prévia:** Antes de realizar uma nova requisição, verificávamos se o bairro já possuía as coordenadas geográficas armazenadas no banco.

- **Requisição condicional:** Caso as coordenadas estivessem disponíveis, elas eram reutilizadas; caso contrário, uma nova requisição era feita à API.

- **Atualização incremental:** Para bairros novos, as coordenadas obtidas eram adicionadas ao banco de *coord_enderecos*, evitando requisições futuras redundantes.

Por fim, na camada *Gold*, realizamos transformações detalhadas e específicas para consolidar os dados. Essa etapa incluiu a desagregação dos dados recebidos do SIGA, o tratamento necessário e a reestruturação conforme a modelagem descrita na Figura 6.

Embora as dimensões *D_PERIODO* e *D_ALUNO* estivessem prontas para uso, já que suas colunas não apresentavam dados aninhados, outras dimensões e tabelas fato exigiram transformações adicionais. Em especial, as dimensões *D_DISCIPLINA* e *D_SITUACAO*, assim como as tabelas fato *F_DESEMPENHO_ACADEMICO*, *F_SITUACAO_PERIODO* e *F_SITUACAO_MATRICULA*, necessitaram de desagregação devido à complexidade das colunas aninhadas nos dados originais.

Para facilitar a compreensão, imagine que temos apenas duas tabelas: Alunos e Disciplinas, e aplicamos os tratamentos nas mesmas. Nas disciplinas, a primeira etapa para obter os dados correspondentes é a desagregação da coluna *disciplinasCursadas*. Essa

[5] https://www.4devs.com.br/
[6] https://geopy.readthedocs.io/en/stable/

coluna contém uma lista de disciplinas em formato de *string*, com informações agrupadas sobre cada disciplina cursada por um aluno, conforme ilustrado na Figura 7

Figura 7 – Coluna disciplinasCursadas

Fonte: Elaborado pelo autor.

Para normalizar[7] essa estrutura, foi necessário transformar essa *string* em uma lista, separando-a por quebras de linha, e, em seguida, realizar um *explode* que é uma técnica de manipulação de dados que transforma listas aninhadas em várias linhas, ou seja, explode, de modo que cada elemento da lista se torne uma nova linha individual, de modo a obter cada registro de disciplina individualmente associado a um aluno. Após a normalização, analisamos cada registro, extraímos o código, o nome da disciplina, o grau, a situação e o período em que a disciplina foi cursada, transformando essas informações em novas colunas: *codDisciplina*, *nomeDisciplina*, *grauDisciplina*, *situacaoDisciplina* e *periodoDisciplina*.

Com os dados das disciplinas normalizados, realizamos um processo para corrigir erros de escrita nos nomes das disciplinas. Por exemplo, "Top Esp em Eng de Softaware" foi corrigido para "Top Esp em Eng de Software". Ajustamos também casos em que o mesmo código de disciplina apresentava nomes diferentes devido à mudança de currículo, como "Computacao para Informatica", que passou a ser "Computacao I (CC)". Adicionamos uma nova coluna de situação *situacaoDisciplinaDetalhada* onde mantemos o valor original,

[7] A normalização tratada neste projeto refere-se às formas normais (FN1, FN2, FN3) da modelagem de banco de dados, que visam organizar os dados de forma estruturada e eliminar redundâncias.

e na coluna *situacaoDisciplina* realizamos uma abreviação das situações conforme a tabela. 2

Tabela 2 – Mapeamento de Situações das Disciplinas

Situação Detalhada	**Situação**
Aprovado	Aprovado
Reprovado media	Reprovado
Repr falta/media	Reprovado
Reprovado faltas	Reprovado
Grau incompleto	Grau incompleto

Fonte: Próprio Autor.

Por fim, para enriquecer ainda mais os dados, realizamos um *web scraping* no portal SIGA para analisar os currículos do curso de Ciência da Computação e determinar se uma disciplina é Obrigatória ou Eletiva. Para os alunos ingressantes antes do currículo de 2004, realizamos uma inferência para criar um currículo estimado, denominado "período 0000-2004", uma vez que o portal SIGA não disponibiliza o currículo anterior ao ano de 2004.

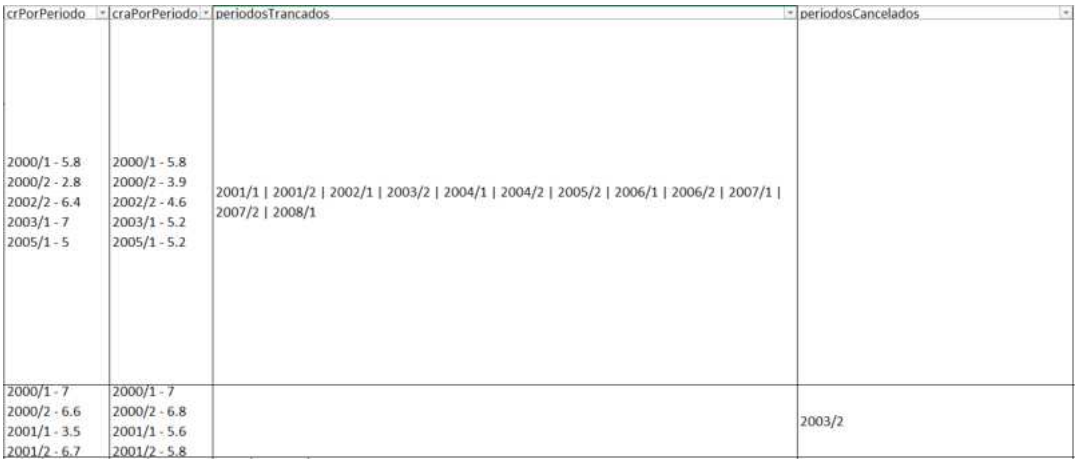
Esse processo envolveu a análise dos alunos formados que ingressaram antes de 2004. Agrupamos os dados por período e disciplinas cursadas, identificando a frequência com que certas disciplinas eram realizadas em determinados períodos. Por exemplo, se 80% dos alunos cursavam Cálculo 3 no terceiro período três anos após o ingresso, a probabilidade de essa disciplina ser obrigatória nesse período era considerada alta. Para validar essas inferências, utilizamos o currículo de 2004-2010, verificando se algumas disciplinas permaneceram no mesmo período, o que nos permitiu consolidar o currículo anterior a 2004.

Apesar de termos obtido mais informações sobre as disciplinas, como carga horária, créditos e requisitos, optamos por não utilizá-las devido à complexidade elevada que isso geraria. O tratamento dessas informações exigiria lidar com diversas equivalências entre disciplinas, o que implicaria a criação de regras de negócio complexas. Consideramos que esse tipo de dado é mais facilmente obtido diretamente com a equipe do SIGA.

Agora, vamos tratar as informações de aluno. As colunas com os dados dos alunos necessárias para a construção das fatos *F_SITUACAO_MATRICULA* e *F_DESEMPENHO_ACADEMICO* incluem *periodosCancelados*, *periodosTrancados*, *craPorPeriodo* e *crPorPeriodo*.

Como ilustrado na Figura 8, é necessário realizar a normalização dessas colunas, conforme foi feito anteriormente em disciplinas. Para construir a fato *F_SITUACAO_MATRICULA*, que representa um dado temporal, é preciso determinar a situação do aluno em cada período específico. Com os dados já normalizados, definimos inicialmente que o período para a situação "Ativa" corresponde ao maior período registrado na co-

Figura 8 – Colunas aninhas de alunos



crPorPeriodo	craPorPeriodo	periodosTrancados	periodosCancelados
2000/1 - 5.8 2000/2 - 2.8 2002/2 - 6.4 2003/1 - 7 2005/1 - 5	2000/1 - 5.8 2000/2 - 3.9 2002/2 - 4.6 2003/1 - 5.2 2005/1 - 5.2	2001/1 | 2001/2 | 2002/1 | 2003/2 | 2004/1 | 2004/2 | 2005/2 | 2006/1 | 2006/2 | 2007/1 | 2007/2 | 2008/1	
2000/1 - 7 2000/2 - 6.6 2001/1 - 3.5 2001/2 - 6.7	2000/1 - 7 2000/2 - 6.8 2001/1 - 5.6 2001/2 - 5.8		2003/2

Fonte: Elaborado pelo autor.

luna *periodoIngressoUFRJ*. Para identificar os períodos em que o aluno esteve trancado, utilizamos a coluna *periodosTrancados*.

A definição das situações "Concluído" ou "Cancelado" foram mais desafiadoras, pois, em alguns casos, a coluna *periodosCancelados* não continha registros para alunos que cancelaram ou concluíram o curso; em outros, havia alunos com múltiplos cancelamentos em diferentes períodos.

Para contornar essa limitação, ao identificar se o aluno concluiu ou cancelou, analisamos o maior período registrado na coluna *periodosCancelados* e comparamos com o último período registrado no CRA do aluno. Se o período na coluna *periodosCancelados* for maior, consideramos este como o período em que o aluno cancelou ou concluiu. Caso contrário, utilizamos o período imediatamente posterior ao último registro de CRA para definir a conclusão ou cancelamento.

Para construir o fato *F_SITUACAO_PERIODO*, foi necessário apenas obter o período do CRA e CR juntamente com os seus respectivos valores para aquele semestre. Para essa tabela tivemos a necessidade de remover CRs duplicados, pois em alguns casos um aluno possuía dois valores distintos de CR para um mesmo período, como ilustrado na Figura 9. Para resolver isso, optamos por manter o CR com o maior valor.

Assim como em disciplinas, criamos uma nova coluna para registrar uma versão detalhada da situação de matrícula, conforme pode ser visto na Tabelas 3.

Figura 9 – Aluno com dois CR no mesmo período.

Fonte: Elaborado pelo autor.

Tabela 3 – Mapeamento de Situações da Matrícula

Situação Detalhada	**Situação**
Ativa	Ativa
Rematrícula por destrancamento automático	Ativa
Rematrícula por ativação do segmento referente via AGF	Ativa
Rematrícula por destrancamento ou descancelamento	Ativa
Cancelada por abandono	Cancelada
Cancelada por abandono definitivo	Cancelada
Cancelada por morte	Cancelada
Cancelada por outros motivos	Cancelada
Cancelada por rendimento escolar insuficiente	Cancelada
Cancelada por transferência	Cancelada
Cancelada por ultrapassar prazo de integralização	Cancelada
Cancelamento a pedido	Cancelada
Cancelamento por conclusão de Mobilidade Acadêmica	Cancelada
Cancelamento por decisão judicial	Cancelada
Cancelamento por opção de curso	Cancelada
Cancelamento por opção de instituição	Cancelada
Cancelamento por ultrapassagem do prazo máximo de trancamento	Cancelada
Trancada	Trancada
Trancamento Solicitado	Trancada
Trancamento automático: Perigo de cancelamento imediato	Trancada
Cancelada por conclusão de curso	Concluído

Fonte: Próprio Autor.

A situação de Exclusão Lógica não consta na tabela, pois foi descartada, conforme explicado por FERREIRA; CANAANE, 2021. Essa situação se refere a alunos que participaram de todo o processo de ingresso e tiveram suas matrículas registradas no sistema, mas desistiram da vaga antes de iniciar qualquer atividade acadêmica. Por não agregarem informações relevantes para a análise, os registros associados a essa situação foram excluídos do conjunto de dados.

3.3.3 Carga dos Dados

Após as transformações realizadas nos dados e a criação das tabelas de dimensões e fatos, os resultados finais são armazenados na camada *Gold*, que representa a última etapa do processo. A figura 10 ilustra a organização do *Data Lake*, destacando as camadas *Landing*, *Bronze*, *Silver* e *Gold*.

Para fins ilustrativos, os dados da camada *Landing*, que normalmente são apagados após a carga, foram mantidos. Nessa camada, encontram-se três principais objetos:

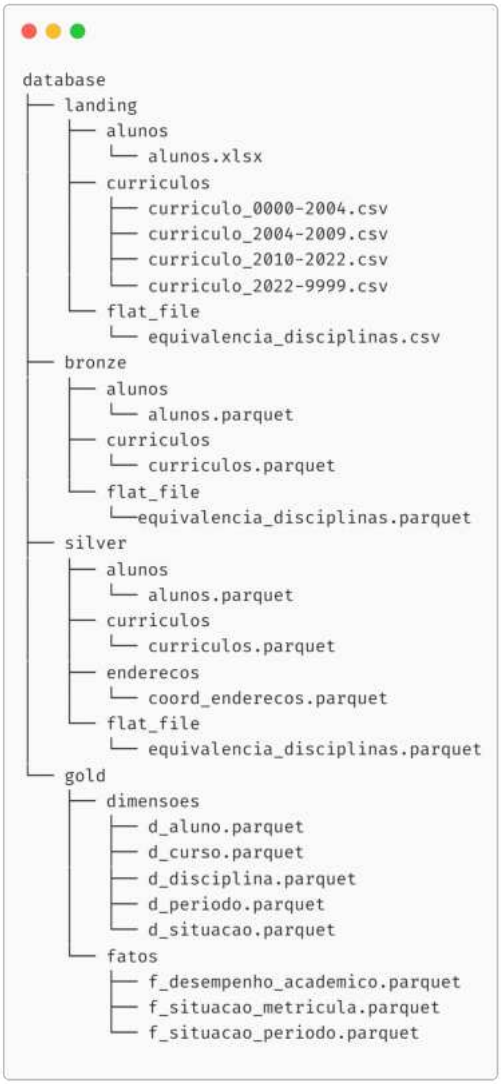
- **Alunos**: Dados recebidos do SIGA no formato `xlsx`.
- **Currículos**: Dados extraídos por meio de *web scraping* no portal do SIGA.
- **Flat File**: Arquivos gerados manualmente pelo desenvolvedor para realizar mapeamentos de equivalências de disciplinas.

Na camada *Bronze*, os dados já estão formatados, agrupados e padronizados no formato *parquet*. Já na camada *Silver* contém dados mais refinados e estruturados, com um novo objeto, *enderecos*, que funciona como um banco de dados de latitude e longitude de bairros. Esse objeto foi criado para otimizar o enriquecimento dos dados com informações geográficas, como descrito anteriormente. Por fim, na camada *Gold*, encontram-se as tabelas de dimensões e fatos, modeladas de acordo com os objetivos do projeto.

Em um ambiente ideal, após serem salvos na camada *Gold*, esses dados seriam transferidos para uma estrutura de banco de dados dedicada à análise de dados, como um *Data Warehouse*. Essa abordagem permitiria que os dados fossem consumidos diretamente por diversos usuários, eliminando a necessidade de que cada um executasse o processo ETL de forma independente.

No entanto, no contexto deste projeto, onde o sistema é projetado para uso individual, a implementação de um *Data Warehouse* local não se justifica. A criação e manutenção de um banco de dados DW para apenas um usuário aumentariam a complexidade sem trazer benefícios proporcionais. Por questões de simplicidade e eficiência, os dados serão armazenados localmente e consumidos diretamente da camada *Gold* para as visualizações.

Figura 10 – Estrutura hierárquica do *Data Lake*.



Fonte: Elaborado pelo autor.

3.4 ORQUESTRAÇÃO DO PROCESSO ETL

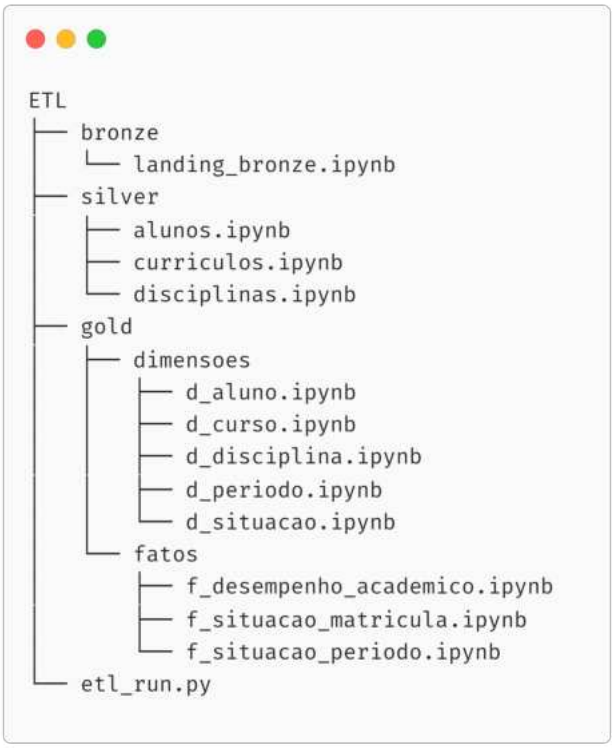
Para controlar a ordem e o momento de execução dos *notebooks*[8] na etapa de ETL, optamos por utilizar a biblioteca *Papermill*, que atende bem às nossas necessidades, permitindo a definição de uma sequência de execução e a parametrização dos *notebooks*, que torna o processo mais eficiente e organizado.

O projeto conta com doze *notebooks* distribuídos entre diferentes camadas, cada um com uma finalidade específica, conforme detalhado na Seção 3.3.2. A organização dessas camadas, juntamente com o *script* principal `etl_run.py`, pode ser visualizada na Figura

[8] Notebook são documentos interativos que combinam código, visualizações e explicações textuais.

11.

Figura 11 – Organização notebook ETL.



Fonte: Elaborado pelo autor.

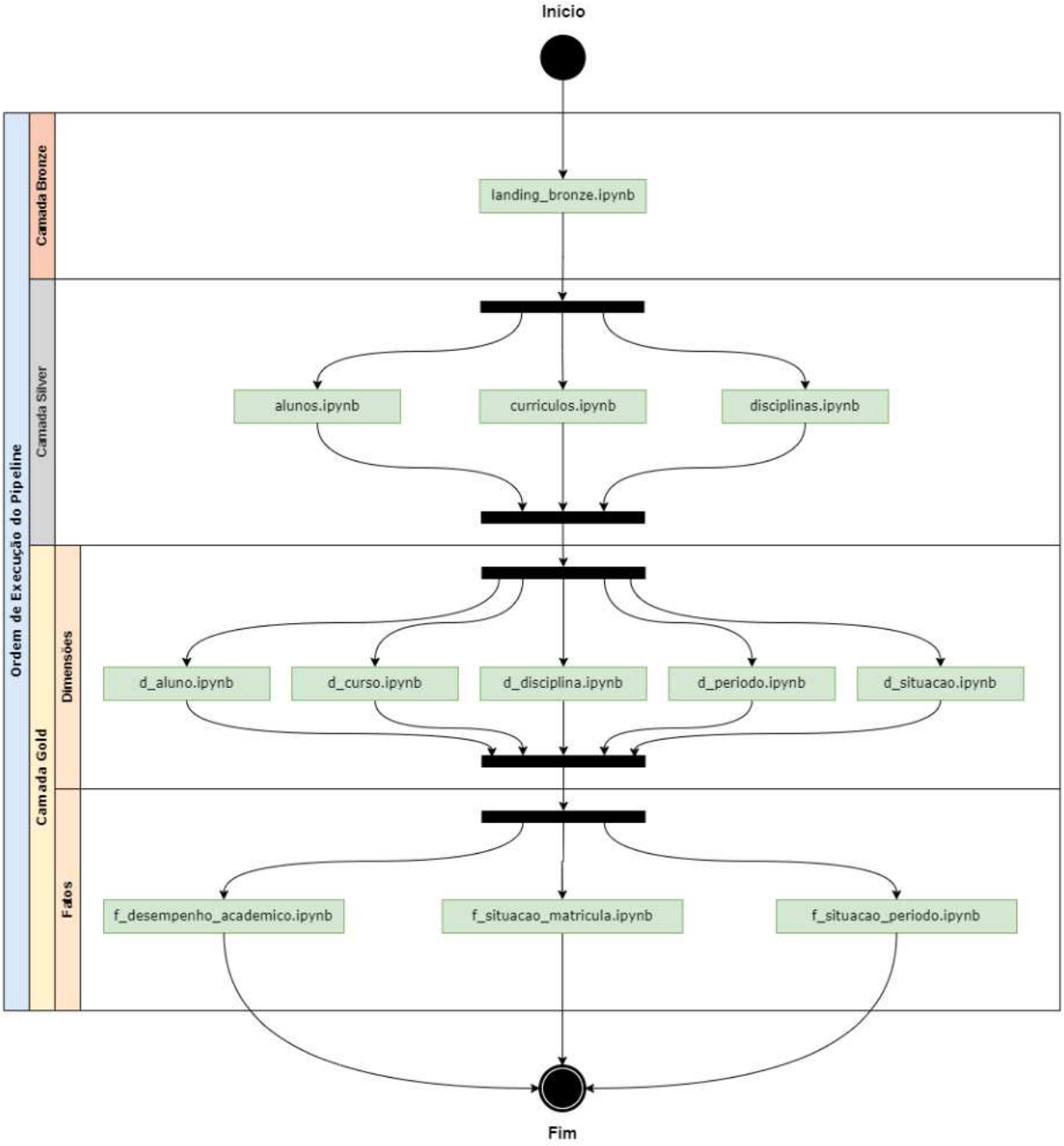
O *script* `etl_run.py` é responsável por orquestrar a execução dos *notebooks*. Ele garante a sequência correta de execução e a configuração adequada dos parâmetros em cada etapa do pipeline ETL. O fluxo de execução pode ser visualizado no diagrama de estados da Figura 12

Apesar de a camada *Bronze* conter apenas um notebook, `landing_bronze.ipynb`, ele é executado três vezes para processar diferentes tipos de dados (alunos, flat_file e currículos), destacando a importância da parametrização proporcionada pelo *Papermill*.

Quanto ao agendamento da ingestão de dados, esta ocorre de forma manual, sem um dia ou horário fixo. A equipe do SIGA fornece o arquivo necessário mediante solicitação, que, após recebido, deve ser enviado para a camada *Landing* e inicia-se o pipeline de execução por meio do *Papermill*.

Em um cenário ideal, o processo seria automatizado, permitindo a captura dos dados do dia anterior ao processamento (D-1) ou ao final de cada semestre, garantindo maior regularidade e atualizações frequentes. No entanto, devido à ausência de acesso direto ao banco de dados do SIGA, essa etapa depende exclusivamente da disponibilidade do

Figura 12 – Ordem de Execução dos notebook.



Fonte: Elaborado pelo autor.

arquivos de dados. Isso representa uma limitação no fluxo, destacando a importância de futuras melhorias para integrar e automatizar a ingestão de dados.No futuro, espera-se que o sistema seja integrado ao SIGA e seja capaz de realizar a coleta diária dos dados.

4 DESENVOLVIMENTO DA APLICAÇÃO

Para iniciar o desenvolvimento do software, primeiramente foi discutida sua modelagem, visto que, de acordo com PáDUA; CAZARINI, 2003, muitas falhas ocorrem em projetos devido à falta de conhecimento das atividades envolvidas no sistema de informação. Para obter insumos relevantes, foi consultado o professor João Carlos Pereira da Silva (Instituto de Computação - UFRJ), membro da COAA (Comissão de Orientação e Acompanhamento Acadêmico), e que também já atuou como Coordenador do Curso de Bacharelado de Ciência da Computação da UFRJ. Desse modo, com o auxílio para determinar as análises demandadas, utilizamos diagramas como ferramentas de nossa modelagem, para que fosse possível mapear as interações entre os componentes do projeto, identificando seus requisitos e antecipando possíveis desafios do desenvolvimento. Essas atividades proporcionam maior clareza e facilitam a comunicação entre os membros da equipe para manter o alinhamento sobre a solução que deve ser construída.

4.1 MODELAGEM DO SISTEMA

No primeiro momento foi levantada uma lista de funcionalidades e perguntas que gostaríamos que o sistema respondesse (apêndice C). Dessa maneira, a modelagem foi construída dividindo módulos com suas respectivas funcionalidades e dados, apresentados na figura 13.

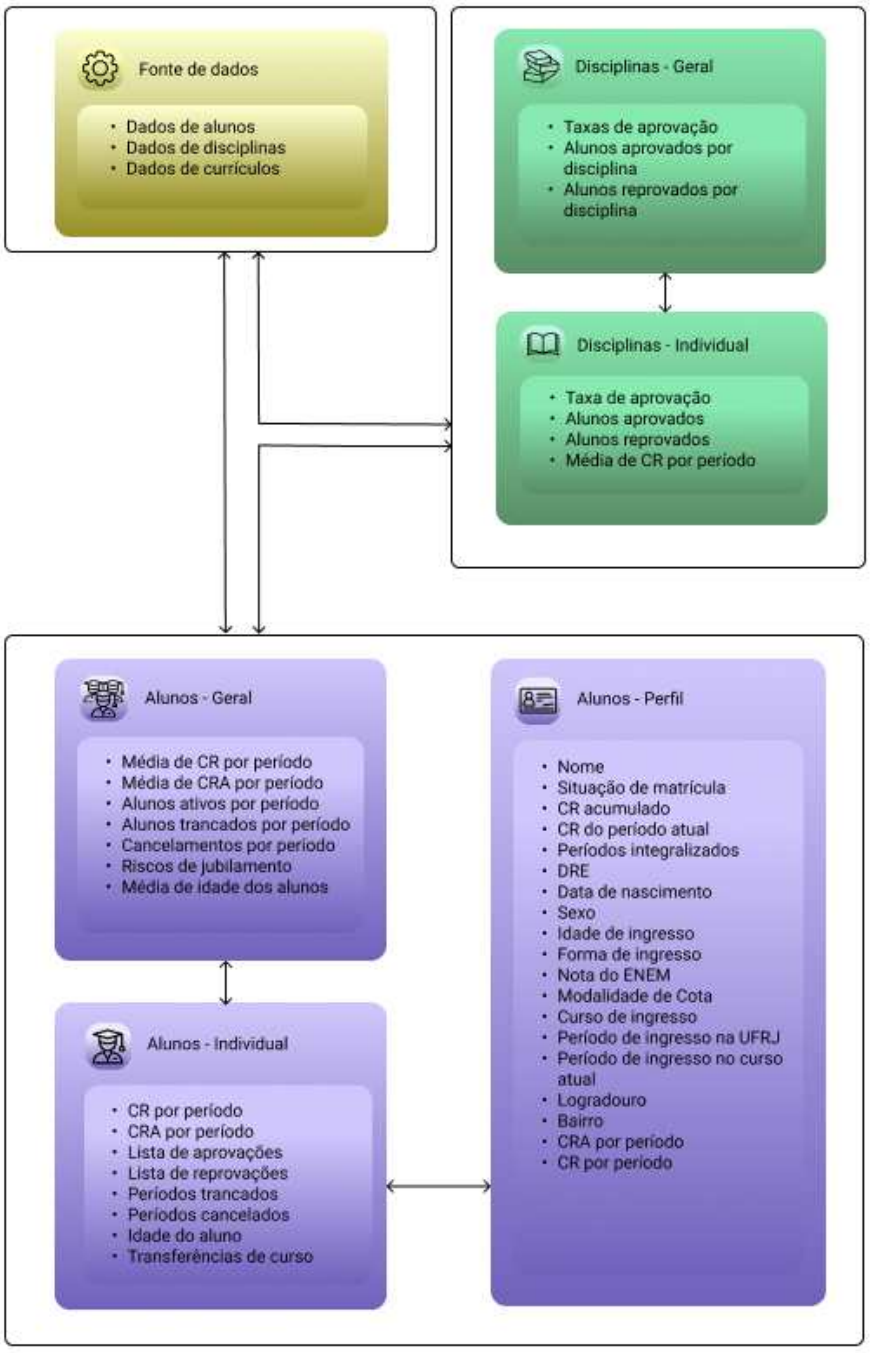
No módulo *Fonte de Dados* os usuários carregam os arquivos necessários para alimentar o sistema: dados de alunos, disciplinas e currículos. Após o carregamento, o processo de ETL pode ser iniciado, permitindo que os dados sejam processados e preparados. Após isso, o usuário é capaz de visualizar as análises das demais páginas. Em um momento posterior, caso haja necessidade de atualizar a base de dados, basta retornar a essa página e carregar novos arquivos.

Em *Disciplinas - Geral* é apresentada uma visão panorâmica sobre as disciplinas. Indicadores como taxas de aprovação, número de alunos aprovados e reprovados por disciplina são exibidos para permitir uma análise ampla do desempenho acadêmico. Esses dados ajudam a identificar padrões gerais, como disciplinas com alta taxa de reprovação, que podem indicar necessidade de ajustes pedagógicos.

Já a página de *Disciplinas - Individual* foca em uma disciplina específica, detalhando sua taxa de aprovação, número de alunos aprovados e reprovados, além da média de coeficientes de rendimento por período. Essa abordagem é essencial para entender os resultados de períodos letivos distintos, oferecendo informações úteis para os professores e coordenadores do curso.

Em *Alunos - Geral* há uma visão agregada sobre o corpo discente. Indicadores como

Figura 13 – Páginas do sistema



Fonte: Elaborado pelo autor.

a média de CR e CRA por período, número de alunos ativos, trancamentos e cancelamentos por período, riscos de jubilamento e a média de idade dos alunos ajudam a compreender o perfil dos estudantes, desempenhos e retenção, identificando tendências a nível institucional.

O módulo *Alunos - Individual* oferece dados sobre cada aluno, incluindo o desempenho

em termos de CR e CRA por período, listas de disciplinas em que houve aprovações e reprovações, períodos trancados e cancelados, idade e transferências de curso. Essas informações são úteis para entender a trajetória acadêmica do aluno, auxiliando na tomada de decisões administrativas ou pedagógicas específicas.

Somado a isso, a página *Perfil do Aluno* apresenta informações demográficas e acadêmicas detalhadas, como nome, situação de matrícula, coeficientes de rendimento (atual e acumulado), períodos integralizados, DRE, forma de ingresso, nota do ENEM, modalidade de cota, histórico de desempenho e informações de endereço. Esse módulo sintetiza dados essenciais que ajudam a traçar o perfil do aluno de forma completa. Além disso, o usuário é capaz de navegar entre todos os módulos (fonte de dados, disciplinas e alunos), conforme fluxo representado pelas setas que ligam as páginas.

Para complementar o planejamento do projeto, também foram criados diagramas de sequência, que representam interações entre os usuários e funcionalidades. Esses diagramas ilustram a ordem das atividades realizadas no sistema e a troca de informações entre seus componentes, permitindo uma visão temporal dos processos.

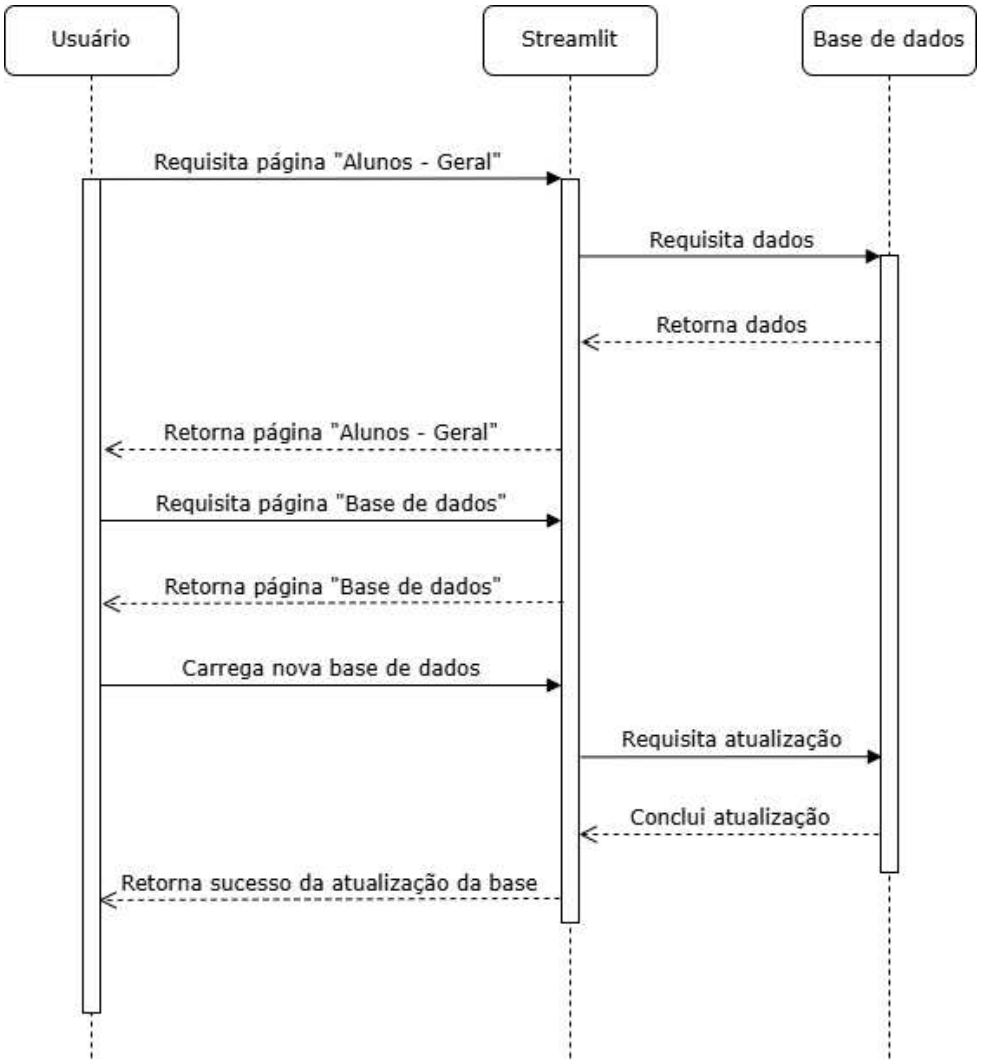
O primeiro diagrama de sequência é o de atualização da base de dados do sistema, representado na Figura 14. Esse processo foi selecionado devido à importância de manter a base com dados dos períodos letivos atualizados, o que fará com que essa atividade ocorra frequentemente. Inicialmente, o fluxo começa quando o usuário solicita a visualização da página *Alunos - Geral* na interface da aplicação. O Streamlit, ao receber essa requisição, realiza uma consulta à base de dados para obter as informações necessárias.

A base de dados, por sua vez, processa a requisição e retorna os dados para o Streamlit, que compõe a página requisitada e a disponibiliza ao usuário. Caso não haja nenhuma base de dados disponível, o usuário receberá, na página em que estiver, um aviso sobre a necessidade de atualizar a base. Em um segundo momento, o usuário acessa a página de *Base de Dados* através de uma nova requisição ao Streamlit. Essa página é retornada diretamente ao usuário, fornecendo a interface necessária para o carregamento de novos arquivos.

Após selecionar e carregar os arquivos correspondentes, o Streamlit realiza uma comunicação com a base de dados, solicitando a atualização. Neste estágio, o processo de ETL (Extração, Transformação e Carga) é executado, integrando as novas informações. Uma vez concluída a atualização, a base de dados informa ao Streamlit que a operação foi finalizada, e este, por sua vez, notifica o usuário sobre o sucesso da atualização.

O segundo diagrama de sequência, na figura 15, modela o processo de aplicação de filtros, muito comum na usabilidade do sistema, uma vez que por diversas ocasiões o usuário deve filtrar as informações para realizar análises específicas. O fluxo do diagrama inicia quando o usuário solicita a página *Alunos - Geral* ao Streamlit. Ao receber essa requisição, o Streamlit consulta a base de dados para obter as informações completas necessárias para compor a página. A base de dados processa a solicitação e retorna os

Figura 14 – Atualização de base de dados

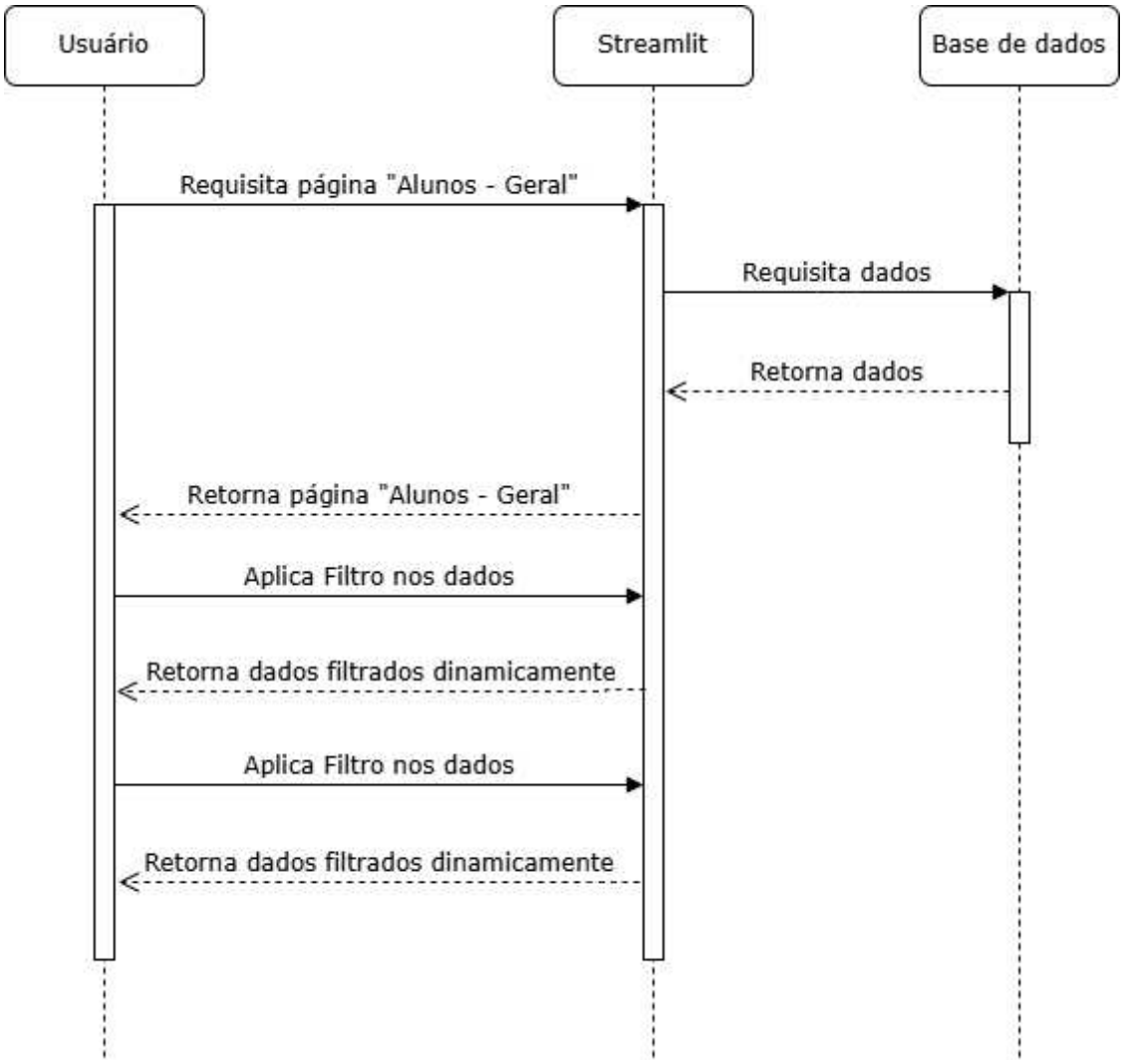


Fonte: Elaborado pelo autor.

dados solicitados, permitindo que o Streamlit prepare e entregue a página requisitada ao usuário.

Após visualizar a página, o usuário pode aplicar um filtro, tendo disponíveis os seguintes: *sexo* (feminino ou masculino), *período letivo* (2000/1 a 2021/1) e *modalidade de cota* (baixa renda, rede pública, ampla concorrência, etc.). Como os dados necessários já estão disponíveis na memória cache, a aplicação processa o filtro dinamicamente, sem realizar uma nova consulta à base de dados. Os dados filtrados são então apresentados ao usuário de forma imediata. Em seguida, o usuário aplica um segundo filtro, e, novamente, o Streamlit processa o filtro de forma dinâmica, utilizando os dados já carregados. Após o processamento, os dados filtrados são apresentados ao usuário, refletindo as condições impostas. Esse diagrama (15) ilustra a eficiência do sistema ao realizar operações de filtragem sem a necessidade de acessar a base de dados repetidamente. Assim, obtendo a visão planejada para o sistema através dessas ferramentas, foi possível iniciar o desenvolvimento

Figura 15 – Aplicação de filtros



Fonte: Elaborado pelo autor.

do software.

4.2 IMPLEMENTAÇÃO

Para o desenvolvimento do software de visualização de dados, utilizamos o Streamlit[1], uma ferramenta que possui destaque por sua simplicidade e eficiência na criação de aplicações focadas em análises de dados, fatores vistos como prioritários, tendo em vista a pesquisa realizada em KIMBALL; ROSS, 2013, que demonstra que 70% das pessoas costumam demandar análises que envolvem mais de uma variável, trazendo solicitações complexas. Por isso, é necessário que o sistema criado possua velocidade para carregamento de gráficos e execução de cálculos.

[1] Página da web disponível em https://streamlit.io/.

No software desenvolvido, a base de dados é consultada e as informações são guardadas em cache no navegador para alimentar as páginas. Dessa maneira, quando um gráfico depende de dados que já foram processados anteriormente, o Streamlit reutiliza os resultados armazenados, evitando a reexecução dessa lógica, desde que os parâmetros das funções e os dados de entrada não tenham sido alterados. Assim, cada página pode apresentar diversos gráficos, e, internamente a fonte de dados principal é particionada em grupos menores para cada visualização. Por exemplo, para obter a média de notas da disciplina de "Computação I"no período de 2017/1, são agrupados apenas os dados de rendimento relevantes para essa análise, filtrando a disciplina. Depois, o filtro do intervalo de período letivo é aplicado de maneira dinâmica, uma vez que pode ser alterado rapidamente pelo usuário no menu de filtros. Dessa maneira, após visualizar as informações, o usuário poderia mudar o período letivo para 2018/1, e o sistema recarregaria os gráficos da página rapidamente, respondendo à alteração.

É possível, também, combinar conjuntos distintos para obter análises personalizadas. Como exemplo, para obter um gráfico de quantidade de alunos aprovados e reprovados na disciplina de "Computação I"por período letivo, teríamos um conjunto que agrupa os alunos com situação de conclusão "aprovado", enquanto outro conjunto agrupa os dados de alunos reprovados, e, depois, ambos os conjuntos seriam unidos para apresentar as informações em um único gráfico. A partir das características descritas para a aplicação, serão apresentadas, nessa seção, as interfaces construídas para cada uma das páginas, bem como as explicações dos elementos presentes nas mesmas.

4.2.1 Página Geral de Alunos

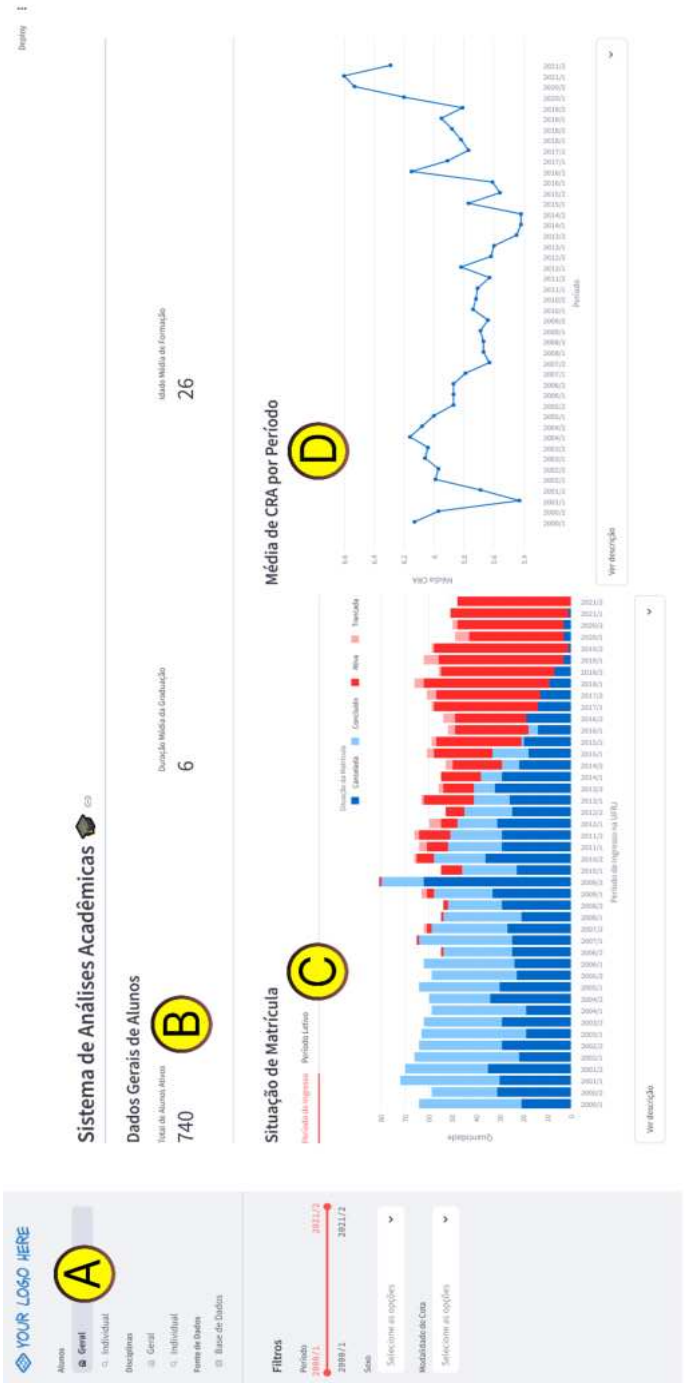
Ao iniciar o sistema, a primeira página acessada é a seção geral de alunos, como é possível visualizar na Figura 16.

Para navegar pelas páginas, pode-se usar a barra encontrada no canto esquerdo da tela (figura 16.A). É possível observar, na parte superior da tela, um destaque para informações do número total de alunos ativos (figura 16.B), duração média da graduação em anos e idade média com que os alunos concluíram o curso. Somado a isso, há também gráficos que abordam a distribuição de situações de matrículas por período de ingresso e período letivo (figura 16.C), e a média de coeficiente de rendimento acumulado dos alunos por período (figura 16.D).

Mais abaixo, na mesma página, existem outras visualizações que podem ser vistas na na Figura 17, como a tabela de alunos em situação de risco (figura 17.A), para facilitar o acompanhamento de casos sensíveis, além de um mapa informativo de concentração de alunos por bairro na cidade do Rio de Janeiro (figura 17.B).

Ao fim da página, como pode ser visto na Figura 18, também é abordada, em um gráfico de colunas (figura 18.A), a relação entre ingressantes e concluintes do curso, além de um gráfico violino (figura 18.B), que representa a distribuição do tempo que os alunos

Figura 16 – Página Geral de Alunos



Fonte: Elaborado pelo autor.

levam para concluir a formação, onde a linha de dentro de cada gráfico é a mediana do tempo, e a caixa sinaliza o intervalo interquartil.

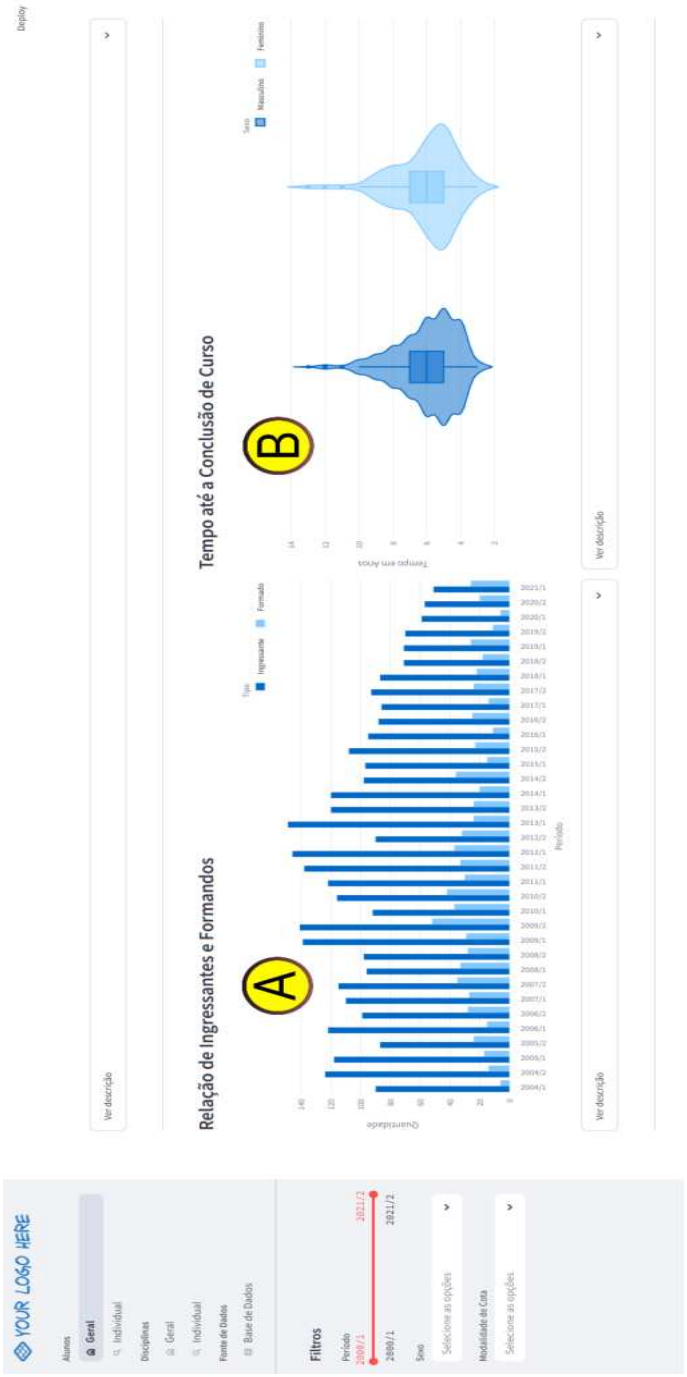
Figura 17 – Página Geral de Alunos

Fonte: Elaborado pelo autor.

4.2.2 Página Individual de Alunos

Para que seja possível encontrar informações de um único aluno, na página apresentada na Figura 19 é possível pesquisar (figura 19.A) digitando nome ou o código de DRE do aluno, para que seja feito um acompanhamento detalhado. Somado a isso, pode-se filtrar

Figura 18 – Página Geral de Alunos



Fonte: Elaborado pelo autor.

os alunos por situação de matrícula, modalidade de cota, período de ingresso na UFRJ, período de ingresso no curso, coeficiente de rendimento do período atual e o coeficiente de rendimento acumulado atual. O resultado da busca é mostrado na tabela da parte inferior da tela (figura 19.B).

Figura 19 – Página Individual de Alunos

Fonte: Elaborado pelo autor.

4.2.3 Página de Perfil do Aluno

A partir da página individual de alunos, ao clicar em uma Matrícula DRE, é aberta a página de perfil do aluno, com informações detalhadas a respeito do indivíduo selecionado, como mostra a Figura 20.

Entre os dados encontrados na região superior da página (figura 20.A), estão:

- Nome do aluno.
- Situação da matrícula.
- CR Acumulado.
- CR do período atual.
- Quantidade de períodos integralizados.
- DRE.
- Data de Nascimento.
- Sexo.
- Idade de ingresso na UFRJ.
- Forma de ingresso.
- Nota do ENEM.
- Modalidade de Cota.
- Curso de ingresso na UFRJ.
- Período de ingresso na UFRJ.
- Período de ingresso no curso atual.
- Logradouro.
- Bairro.

Somado a isso, há também dois gráficos, um com o histórico do coeficiente de rendimento acumulado ao longo dos períodos letivos (figura 20.B), e outro com o coeficiente de rendimento de cada período (figura 20.C).

Na seção inferior da mesma página, há uma tabela (figura 21.A) a respeito da situação do aluno em disciplinas, com dados sobre suas aprovações, reprovações, transferências (se aplicável), e detalhes sobre a quantidade de tentativas de conclusão de disciplinas, com a descrição do número de reprovações, caso tenham ocorrido.

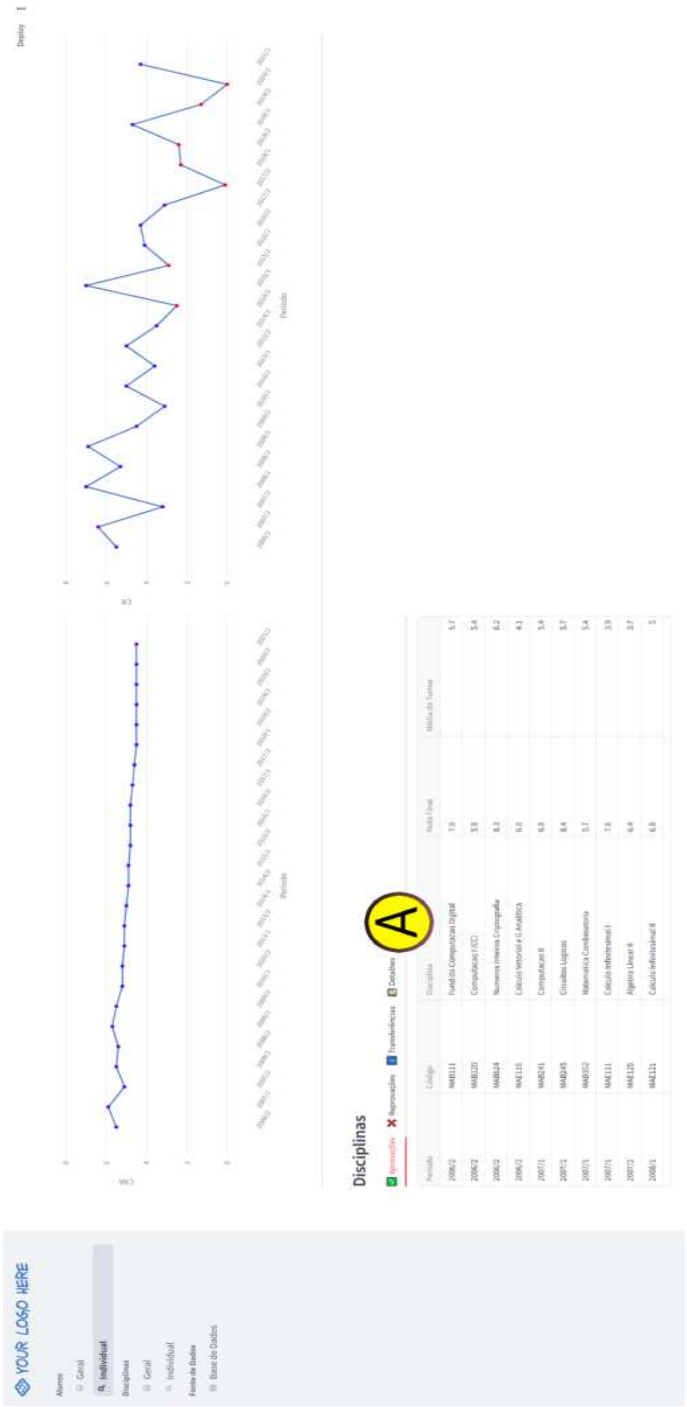
Figura 20 – Página de Perfil do Aluno

Fonte: Elaborado pelo autor.

4.2.4 Página Geral de Disciplinas

Nessa seção, vista na Figura 22, com foco em visualizar e comparar as taxas de aprovações das disciplinas, é apresentada uma busca por disciplinas (figura 22.A), que deve ser realizada por nome ou código da matéria procurada. A pesquisa pode ser feita com

Figura 21 – Página de Perfil do Aluno

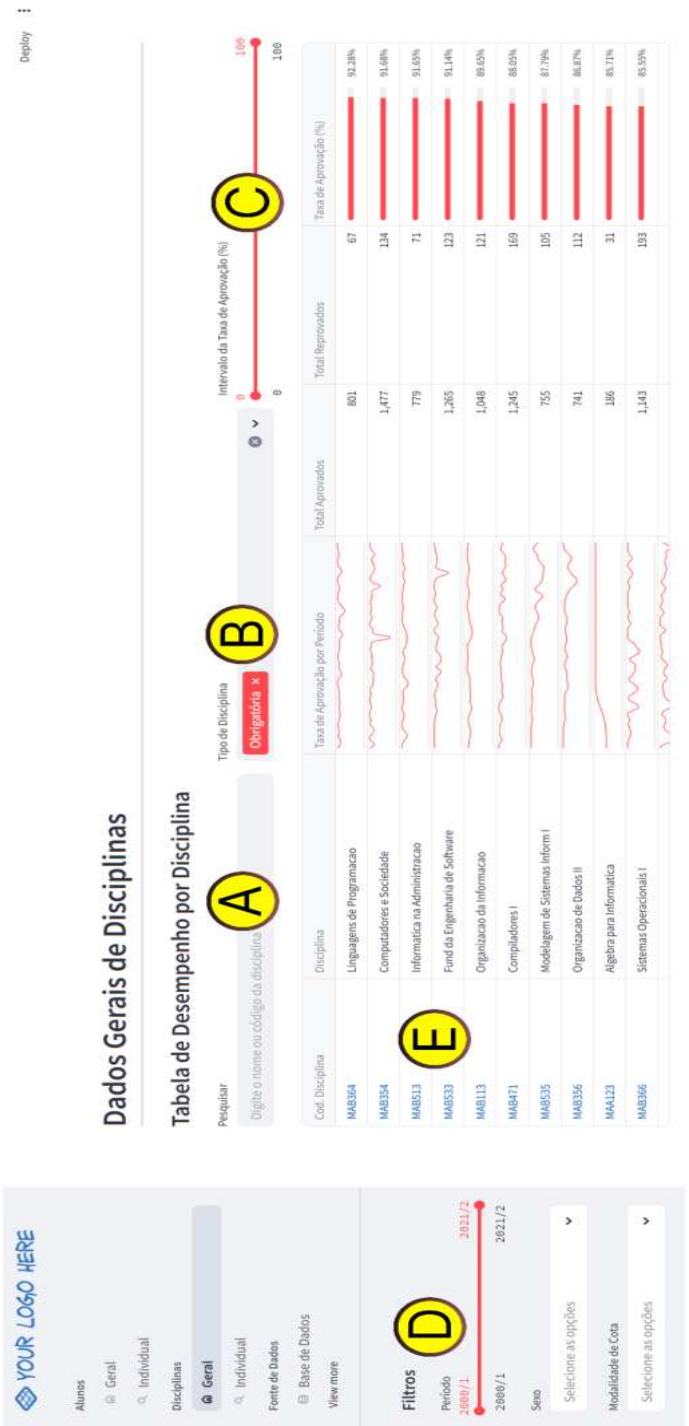


Fonte: Elaborado pelo autor.

filtros de "Tipo de disciplina"(figura 22.B), que descreve se é uma disciplina obrigatória ou eletiva, e a partir do percentual da taxa de aprovação geral (figura 22.C) da disciplina. Além disso, podemos reduzir os dados com os filtros da aba à esquerda da tela (figura 22.D), para visualizar, por exemplo, dados somente referentes a um sexo ou modalidade de cota específicos. Por fim, o resultado é mostrado em uma lista de disciplinas com seus

dados gerais (figura 22.E), podendo ser ordenando para mostrar primeiro as maiores ou as menores taxas de aprovação.

Figura 22 – Página Geral de Disciplinas



Fonte: Elaborado pelo autor.

4.2.5 Página Individual de Disciplinas

Na seção em questão, mostrada na Figura 23, há também uma busca (figura 23.A) por código e nome da disciplina, porém, nesse caso, o objetivo é unicamente encontrar uma disciplina para abrir uma visão específica sobre a mesma ao clicar no código da disciplina (figura 23.B).

Após o clique, são apresentadas, como na Figura 24, informações gerais sobre a disciplina, como índice de aprovação (figura 24.A) e total de alunos que cursaram em dado intervalo de períodos letivos (figura 24.B). Além disso, há um filtro de sexo e modalidade de cotas (figura 24.C), para especificar todas as informações apresentadas nessa tela.

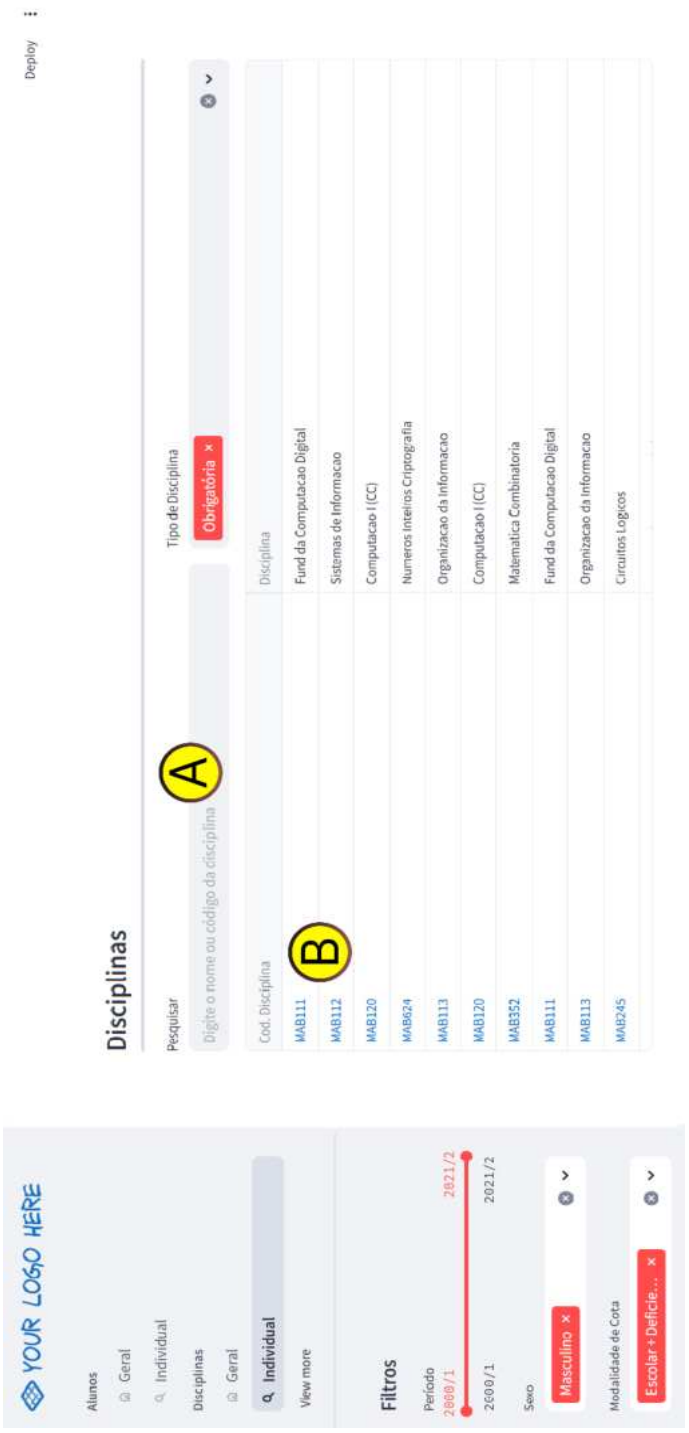
Outro aspecto importante são os gráficos apresentados na página, que demonstram:

- A média de grau dos alunos por período em um gráfico de linhas (figura 24.D).
- A quantidade de alunos aprovados e reprovados por período na disciplina, em um gráfico de colunas categorizadas (figura 24.E).

Dessa maneira, é possível obter uma visão precisa das informações sobre a disciplina em questão, facilitando o acompanhamento do desempenho dos alunos com diversos filtros.

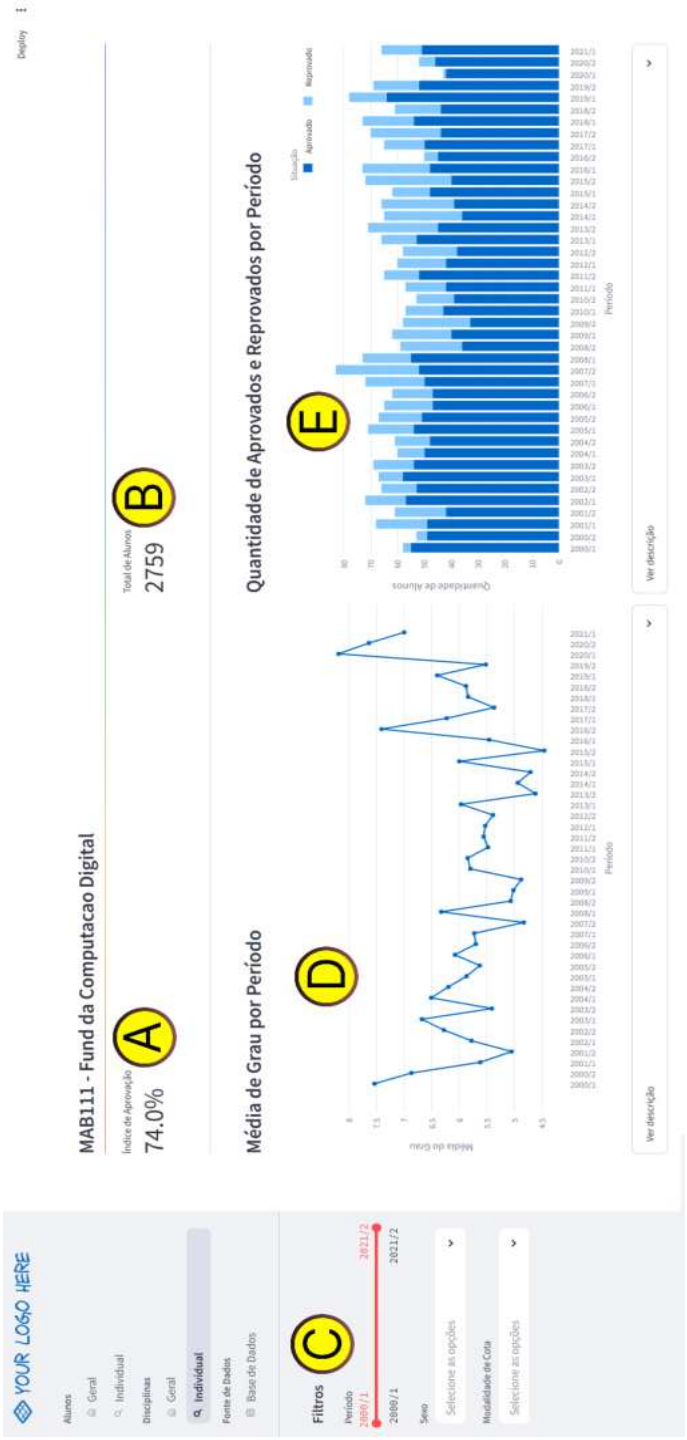
A partir dessas funcionalidades, o projeto tem como objetivo permitir o rápido acesso a gráficos que suportem análises de rendimentos acadêmicos, com uma interface simples, de maneira a auxiliar na orientação dos alunos.

Figura 23 – Página Individual de Disciplinas



Fonte: Elaborado pelo autor.

Figura 24 – Página Individual de Disciplinas



Fonte: Elaborado pelo autor.

5 CONCLUSÃO

O desenvolvimento do software apresentado neste estudo visa facilitar o acompanhamento do desempenho acadêmico dos alunos da UFRJ, proporcionando uma alternativa prática e visualmente amigável ao método tradicional, que exige muitos processos manuais. O sistema foi projetado para permitir que professores e orientadores acessem informações essenciais, como aprovações, reprovações e identificação de alunos em situações críticas, de forma rápida e consolidada. Além disso, a interface oferece a consulta de taxas de aprovação por disciplina, proporcionando uma visão abrangente do rendimento acadêmico e facilitando a tomada de decisões baseadas em dados.

Para viabilizar essas funcionalidades, desenvolvemos uma estrutura adaptável de leitura de arquivos, permitindo o carregamento de dados extraídos do SIGA em planilhas com o mesmo esquema de colunas. Essa solução assegura a consistência e a usabilidade das análises fornecidas, mesmo que o processo de importação dos dados ainda exija intervenção manual. Além disso, também foi construída uma categorização de disciplinas entre obrigatórias e eletivas, e a correção ortográfica de erros nos nomes, o que contribuiu para a precisão do sistema e minimizou inconsistências nos relatórios gerados. Adicionalmente, a escolha pela linguagem Python, aliada ao uso do framework Streamlit, foi determinante para alcançar uma interface ágil e intuitiva, ampliando a capacidade de visualização de dados.

Como evolução futura, propõe-se o desenvolvimento de uma integração direta ao banco de dados do SIGA, automatizando o processo de extração diária e garantindo informações sempre atualizadas dos registros. Com essa mudança, o processo de ETL precisará passar por uma refatoração. Com o acesso direto ao banco, será possível lidar com as tabelas individualmente, em vez de trabalharmos com dados já agregados, como ocorre atualmente. Seguindo as boas práticas da Engenharia de Dados, o agrupamento será realizado apenas na etapa final, correspondente à camada *Gold*.

Além disso, muitos dados adicionais, como endereços, nomes de alunos, DRE, currículos de disciplinas e filtros por disciplinas, não precisarão ser criados, pois espera-se que essas informações já estejam disponíveis no banco do SIGA. Acreditamos também que a conexão com o banco do SIGA proporcionará acesso aos dados dos professores, permitindo listar e filtrar disciplinas por professor, além de calcular o índice de reprovação das disciplinas ministradas por cada docente. Isso possibilitará uma análise mais detalhada para entender se as reprovações estão relacionadas às disciplinas em si ou aos professores que as lecionam.

Por fim, será possível criar uma nova camada, para o uso de métodos de aprendizado de máquina. Essa camada tornará o sistema mais robusto, permitindo prever, por exemplo, quais alunos têm maior probabilidade de evadir do curso após determinada quantidade

de reprovações.

Assim, espera-se que o sistema possa efetivamente apoiar a formação de alunos, permitindo que a equipe docente adote uma abordagem mais estratégica e objetiva. Dessa forma, o projeto representa uma ferramenta útil e expansível para a Universidade, demonstrando a capacidade da tecnologia e da análise de dados em enriquecer o ensino superior.

REFERÊNCIAS

FERREIRA, P. P. S. K.; CANAANE, I. d. O. **Desempenho estudantil: uma análise da situação atual do bacharelado em ciência da computação**. Trabalho de Conclusão de Graduação — Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2021. Orientador: Silva, João Carlos.

GONÇALVES, A. C.; FILHO, F. d. S. C.; GOMES, W. M. d. O. Sistema de apoio ao acompanhamento e orientação acadêmica: um projeto de desenvolvimento de software. Universidade Federal do Rio de Janeiro, 2023.

INMON, W. H. **Building the Data Warehouse**. 4th. ed. Hoboken, New Jersey: Wiley, 2005.

KIMBALL, R.; ROSS, M. **The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling**. 3rd. ed. Hoboken, New Jersey: Wiley, 2013.

MILLECAMP, M. et al. A qualitative evaluation of a learning dashboard to support advisor-student dialogues. In: **Proceedings of the 8th international conference on learning analytics and knowledge**. [S.l.: s.n.], 2018. p. 56–60.

PRESTES, E. M. d. T.; FIALHO, M. G. D. Evasão na educação superior e gestão institucional: o caso da universidade federal da paraíba. **Ensaio: Avaliação e Políticas Públicas em Educação**, SciELO Brasil, v. 26, p. 869–889, 2018.

PáDUA, F. S. M. d.; CAZARINI, E. W. **A importância da técnica de modelagem organizacional EKD no desenvolvimento de diagramas use case**. Dissertação (Mestrado) — Universidade de São Paulo, 2003.

REIS, J.; HOUSLEY, M. **Fundamentals of Data Engineering: Plan and Build Robust Data Systems**. Sebastopol, California: O'Reilly Media, 2022.

SILVA, E. V. da; NETTO, J. F. d. M.; SOUZA, R. A. L. de. O uso de dashboard na identificaçao do desempenho de alunos de matemática básica. 2016.

VASSILIADIS, P. et al. **Fundamentals of Data Warehouses**. Berlin, Heidelberg: Springer Science & Business Media, 2009.

APÊNDICES

APÊNDICE A – ESTRUTURA E EXEMPLOS DOS DADOS RECEBIDOS PELO SIGA

Tabela 4 – Exemplo de Dados Recebidos do SIGA

CAMPO	**DESCRIÇÃO**	**EXEMPLO**
periodoIngressoUFRJ	Período em que o aluno ingressou na UFRJ	2015/1
cursoIngressoUFRJ	Curso que o aluno ingressou na UFRJ	Bacharelado em Ciência da Computação
codCursoIngresso	Identificador único do curso de ingresso	3101070000
cursoAtual	Curso atual do aluno	Ciência da Computação
codCursoAtual	Identificador único do curso atual	3109000100
periodoIngressoCursoAtual	Período em que o aluno entrou no curso atual	2015/1
formaIngresso	Forma de ingresso do aluno na UFRJ	SiSU - Sistema de Seleção Unificada
situacaoMatriculaAtual	Situação atual da matrícula do aluno	Cancelada por conclusão de curso
notaEnem	Nota obtida no Enem	900.72
modalidadeCota	Modalidade utilizada no SiSU	Escolar + Renda + Racial
dataNascimento	Data de nascimento	29/02/1996
sexo	Gênero do aluno	M
disciplinasCursadas	Disciplinas cursadas em formato de string	2013/2 - MAB111 Fund da Computação Digital - 005 - Reprovado media 2013/2 - MAB112 Sistemas de Informação - 033 - Repr falta/media
crPorPeriodo	CR por período	2013/1 - 5.4 2013/2 - 0.2 2014/1 - 1
craPorPeriodo	CRA por período	2013/1 - 5.5 2013/2 - 3.3 2014/1 - 2.6

CAMPO	**DESCRIÇÃO**	**EXEMPLO**
periodosTrancados	Períodos em que a situação foi trancada	2020/1 | 2020/2
periodosCancelados	Períodos em que a situação foi cancelada	2019/2
periodosCRMenor3	Períodos em que o CR foi menor que 3	2015/1 | 2017/2
reprovacoes	Disciplinas em que o aluno foi reprovado	2015/2 - FIW125 Mecânica, Oscilação e Ondas - 038 - Reprovado media

Fonte: Dados fornecidos pelo SIGA/UFRJ.

APÊNDICE B – DEPENDÊNCIAS DO PROJETO

O projeto foi desenvolvido em Python versão `3.12.6`. Abaixo estão listadas as dependências divididas por componentes:

B.1 ENGENHARIA DE DADOS E PROCESSO ETL

```
numpy==2.1.0
pandas==2.2.2
faker==28.4.1
openpyxl==3.1.5
papermill==2.6.0
unidecode==1.3.8
geopy==2.4.1
```

B.2 DESENVOLVIMENTO DA APLICAÇÃO

```
numpy==2.1.0
pandas==2.2.2
plotly==5.22.0
scipy==1.14.1
```

APÊNDICE C – FUNCIONALIDADES DO SISTEMA

- **Acessar Página Alunos - Geral:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Abrir da página principal com a visão geral de informações dos alunos.
  - Finalidade: Permitir acesso a dados agregados dos alunos.
- **Abrir a barra lateral de navegação:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Clicar na barra lateral de navegação.
  - Finalidade: Acessar diferentes seções e funcionalidades da plataforma.
- **Esconder a barra lateral de navegação:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Ocultar a barra de navegação lateral para melhorar o espaço visual da tela.
  - Finalidade: Proporcionar melhor visualização dos dados principais.
- **Visualizar a quantidade de alunos ativos:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar o número total de alunos com matrícula ativa.
  - Finalidade: Fornecer um indicador rápido da quantidade de alunos ativos.
- **Visualizar a quantidade de alunos com matrícula trancada no momento:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar o total de alunos com matrícula trancada atualmente.
  - Finalidade: Identificar o número de alunos em trancamento.
- **Visualizar a média de idade dos alunos do curso:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar a média de idade dos alunos matriculados.
  - Finalidade: Obter informações demográficas do curso.
- **Visualizar o gráfico de situação de matrícula por período:**

  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar as situações de matrícula por períodos específicos.
  - Finalidade: Acompanhar a evolução da situação acadêmica dos alunos.

- **Visualizar o gráfico de média de CRA dos alunos por período:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar a média do Coeficiente de Rendimento Acumulado (CRA) dos alunos por período.
  - Finalidade: Monitorar o desempenho acadêmico médio dos alunos ao longo dos períodos.

- **Visualizar o gráfico de situação de matrícula por período de ingresso:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar um gráfico que relaciona situação de matrícula ao período de ingresso dos alunos.
  - Finalidade: Analisar a situação acadêmica dos alunos com base no ingresso.

- **Visualizar a relação dos alunos em situação de risco de jubilamento:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar lista dos alunos que se encontram em situação de risco acadêmico.
  - Finalidade: Identificar alunos com rendimento crítico e risco de jubilamento.

- **Acessar Página de Alunos - Individual:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Abrir página com informações detalhadas sobre cada aluno.
  - Finalidade: Obter dados detalhados do desempenho e histórico de um aluno específico.

- **Visualizar a lista de alunos do curso:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar a lista completa dos alunos matriculados no curso.
  - Finalidade: Acessar dados gerais de todos os alunos.

- **Filtrar a lista de alunos por nome:**
  - Ator: Usuário de apoio pedagógico.

  - Ação: Filtrar a lista para mostrar apenas alunos com determinado nome.
  - Finalidade: Facilitar a localização de um aluno específico por nome.

- **Filtrar a lista de alunos por DRE:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Filtrar a lista de alunos utilizando o número de DRE.
  - Finalidade: Encontrar um aluno com base no identificador DRE.

- **Filtrar a lista de alunos por período de ingresso na UFRJ:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Filtrar alunos de acordo com o período em que ingressaram na UFRJ.
  - Finalidade: Identificar alunos de diferentes períodos de entrada.

- **Filtrar a lista de alunos por período de ingresso no curso:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Filtrar alunos conforme o período de ingresso no curso.
  - Finalidade: Analisar dados dos alunos ingressantes por períodos.

- **Filtrar a lista de alunos por situação de matrícula:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Filtrar a lista para exibir alunos conforme a situação de matrícula atual.
  - Finalidade: Identificar alunos com diferentes situações acadêmicas.

- **Filtrar a lista de alunos por modalidade de cota:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Filtrar alunos conforme a modalidade de cota pela qual ingressaram.
  - Finalidade: Permitir análise dos alunos conforme seu tipo de ingresso.

- **Filtrar a lista de alunos por CR do período atual:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Filtrar a lista de alunos por coeficiente de rendimento do período atual.
  - Finalidade: Avaliar o desempenho acadêmico atual dos alunos.

- **Filtrar a lista de alunos por CRA do período atual:**
  - Ator: Usuário de apoio pedagógico.

  - Ação: Filtrar a lista de alunos conforme o CRA acumulado até o período atual.
  - Finalidade: Analisar a situação acadêmica acumulada dos alunos.

- **Pesquisar um nome na lista de alunos:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Buscar nomes específicos na lista de alunos.
  - Finalidade: Acessar dados de alunos específicos.

- **Acessar a página individual do aluno ao clicar na matrícula DRE:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Ser direcionado para a página de informações detalhadas de um aluno ao selecionar seu DRE.
  - Finalidade: Obter dados específicos do aluno.

- **Visualizar informações gerais do aluno em sua página individual:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar detalhes gerais sobre o aluno, como DRE, nome, curso e situação acadêmica.
  - Finalidade: Acessar informações completas do aluno.

- **Visualizar lista de períodos trancados pelo aluno:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar os períodos em que o aluno trancou a matrícula.
  - Finalidade: Obter histórico de trancamentos acadêmicos do aluno.

- **Visualizar gráfico de CR acumulado do aluno por período:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar gráfico que mostra o CR acumulado do aluno ao longo dos períodos.
  - Finalidade: Acompanhar a progressão do desempenho acadêmico do aluno.

- **Visualizar gráfico de CR do aluno por período:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar gráfico que mostra o CR do aluno em cada período.
  - Finalidade: Analisar o desempenho do aluno ao longo dos períodos.

- **Visualizar lista de disciplinas em que o aluno obteve aprovação:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar disciplinas nas quais o aluno foi aprovado.
  - Finalidade: Acompanhar as aprovações e desempenho do aluno.
- **Visualizar lista de disciplinas em que o aluno obteve reprovação:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar disciplinas nas quais o aluno foi reprovado.
  - Finalidade: Identificar pontos de dificuldade no histórico do aluno.
- **Visualizar lista de transferências do aluno:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar histórico de transferências de curso ou instituição do aluno.
  - Finalidade: Obter registro de transferências acadêmicas.
- **Visualizar lista de quantidade de reprovações do aluno em cada disciplina:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar a quantidade de reprovações acumuladas do aluno por disciplina.
  - Finalidade: Avaliar dificuldades persistentes do aluno em determinadas disciplinas.
- **Acessar a página de Disciplinas - Geral:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Acessar a visão geral das disciplinas ofertadas.
  - Finalidade: Visualizar informações agregadas sobre as disciplinas.
- **Visualizar lista de disciplinas com maior taxa de aprovação:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar disciplinas com altos índices de aprovação dos alunos.
  - Finalidade: Identificar disciplinas com desempenho acadêmico positivo.
- **Pesquisar na lista de disciplinas:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Buscar termos específicos na lista de disciplinas.

  - Finalidade: Facilitar a localização de informações de disciplinas.

- **Acessar a página de Disciplinas - Individual:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Acessar informações de uma disciplina específica.
  - Finalidade: Obter detalhes sobre uma disciplina.

- **Visualizar a quantidade de aprovações e reprovações de uma disciplina por período:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar dados sobre a quantidade de aprovações e reprovações dos alunos em uma disciplina específica, segmentados por período.
  - Finalidade: Analisar o desempenho dos alunos em determinada disciplina ao longo dos períodos.

- **Visualizar a média de desempenho dos alunos em uma disciplina por período:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar a média de desempenho dos alunos em uma disciplina ao longo dos períodos.
  - Finalidade: Avaliar o desempenho médio dos alunos em cada período para uma disciplina.

- **Visualizar a moda de desempenho dos alunos em uma disciplina por período:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Visualizar a moda das notas dos alunos em uma disciplina ao longo dos períodos.
  - Finalidade: Identificar padrões de desempenho recorrentes dos alunos em cada período.

- **Filtrar os gráficos da página pelo nome da disciplina:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Filtrar visualizações da disciplina com base no nome da disciplina selecionada.
  - Finalidade: Facilitar a análise específica de disciplinas pelo nome.

- **Filtrar os gráficos da página pelo código da disciplina:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Filtrar as visualizações com base no código da disciplina.
  - Finalidade: Permitir a consulta precisa de uma disciplina pelo seu código específico.
- **Filtrar os gráficos da página pelo intervalo de períodos:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Filtrar dados da disciplina apenas para os períodos selecionados.
  - Finalidade: Analisar o desempenho de uma disciplina dentro de um intervalo de períodos.
- **Acessar a página de Base de dados:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Acessar à página onde o usuário pode gerenciar a base de dados utilizada na plataforma.
  - Finalidade: Possibilitar a atualização e manutenção da base de dados.
- **Escolher um arquivo para carregar base de dados:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Selecionar um arquivo para ser carregado como nova base de dados na plataforma.
  - Finalidade: Substituir a fonte de dados anterior.
- **Executar o ETL para atualizar a base de dados:**
  - Ator: Usuário de apoio pedagógico.
  - Ação: Executar as transformações de dados.
  - Finalidade: Atualizar a base de dados com novas informações acadêmicas.

APÊNDICE D – MODELOS DIMENSIONAL E EXEMPLOS DE REGISTROS

Tabela 5 – Exemplo de Registro para Tabela `D_ALUNO`

CAMPO	**EXEMPLO**
SK_D_ALUNO	f5cbe09ad0475390ce96
CD_MATRICULA_DRE	112512347
NM_ALUNO	João da Silva
DS_SEXO	Masculino
VL_IDADE_INGRESSO_CURSO_ATUAL	18
DT_NASCIMENTO	01/01/2000
DS_PERIODO_INGRESSO_CURSO_ATUAL	2020/1
DS_PERIODO_INGRESSO_UFRJ	2020/1
DS_FORMA_INGRESSO	SiSU
DS_MODALIDADE_COTA	Racial + Renda
VL_NOTA_ENEM	850.5
DS_LOGRADOURO	Rua das Flores
DS_BAIRRO	Centro
DS_CIDADE	Rio de Janeiro
VL_LONGITUDE	-43.357034
VL_LATITUDE	-22.849544

Tabela 6 – Exemplo de Registro para Tabela `D_CURSO`

CAMPO	**EXEMPLO**
SK_D_CURSO	cb88287d73dd29006
CD_CURSO_INGRESSO	3101070000
DS_NOME_CURSO_INGRESSO	Engenharia Életrica
CD_CURSO_ATUAL	3109000100
DS_NOME_CURSO_ATUAL	Ciência da Computação

Tabela 7 – Exemplo de Registro para Tabela `D_PERIODO`

CAMPO	**EXEMPLO**
SK_D_PERIODO	a7746158bdb1b508b
DS_PERIODO	2021/1

VL_ANO	2021
VL_SEMESTRE	1

Tabela 8 – Exemplo de Registro para Tabela `D_DISCIPLINA`

CAMPO	**EXEMPLO**
SK_D_DISCIPLINA	26ae25e81ccc
CD_DISCIPLINA	MAB120
DS_NOME_DISCIPLINA	Computacao I (CC)
TP_DISCIPLINA	Obrigatória

Tabela 9 – Exemplo de Registro para Tabela `D_SITUACAO`

CAMPO	**EXEMPLO**
SK_D_SITUACAO	be3ad86618e4fabd
ST_SITUACAO	Concluido
ST_SITUACAO_DETALHADA	Cancelada por conclusão de curso

Tabela 10 – Exemplo de Registro para Tabela `F_DESEMPENHO_PERIODO`

CAMPO	**EXEMPLO**
SK_D_ALUNO	f59e475394b4d1afd
SK_D_CURSO	26ae25e81ccc85aec
SK_D_PERIODO	cb88287d73dd2962
VL_CR_PERIODO	7.5
VL_CR_ACUMULADO	8.0

Tabela 11 – Exemplo de Registro para Tabela `F_SITUACAO_MATRICULA`

CAMPO	**EXEMPLO**
SK_D_ALUNO	6808a36207567e953
SK_D_CURSO	fbfecc3ab3d37c2110
SK_D_PERIODO	aa36d9d738180d3e95
SK_D_SITUACAO	e6ec8667d6f4052cd7f

Tabela 12 – Exemplo de Registro para Tabela `F_DESEMPENHO_ACADEMICO`

CAMPO	**EXEMPLO**
SK_D_ALUNO	f59e475394b4d1afd3
SK_D_DISCIPLINA	26ae25e81ccc85aecdad
SK_D_CURSO	cb88287d73dd290062
SK_D_PERIODO	9b9d29eed04d79ce77
SK_D_SITUACAO	0d31bc09bac94cf51ac4
DS_GRAU_DISCIPLINA	8.5