UNIVERSIDADE FEDERAL DO PARANÁ

ARTHUR VIANNA LANDEO
CASSIANO RICARDO S C FILHO
TATIANE PORTELA MEDEIROS

SEAS - STUDENT EVASION ANALYSIS SYSTEM

CURITIBA
2019

ARTHUR VIANNA LANDEO
CASSIANO RICARDO S C FILHO
TATIANE PORTELA MEDEIROS

SEAS - STUDENT EVASION ANALYSIS SYSTEM

Trabalho apresentado como requisito para obtenção de grau no curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná.

Orientador: Prof. Dr. Alexander Robert Kutzke.

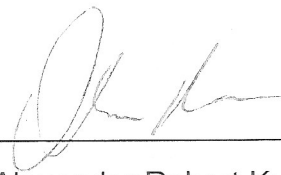
CURITIBA
2019

TERMO DE APROVAÇÃO

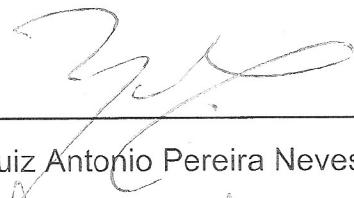
ARTHUR VIANNA LANDEO
CASSIANO RICARDO SANTANA DA CRUZ FILHO
TATIANE PORTELA MEDEIROS

SEAS – Student Evasion Analysis System

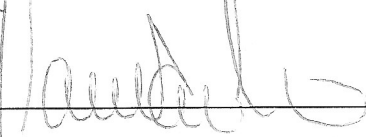
Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.



Prof. Alexander Robert Kutzke
Orientador – SEPT/UFPR



Prof. Luiz Antonio Pereira Neves
SEPT/UFPR



Profª. Mario de Paula Soares Filho
SEPT/UFPR

Curitiba, 06 de Dezembro de 2019.

AGRADECIMENTOS

Agradecemos aos nossos pais, pelo apoio constante na trajetória de nossas vidas. Aos professores, pela instrução e oportunidade de desenvolvimento e construção do conhecimento. Aos colegas, pelo tempo que passamos juntos, aprendendo a conviver e respeitar. E aos amigos, por representarem presenças que jamais esqueceremos.

RESUMO

Este trabalho tem o intuito de construir um sistema capaz de predizer a evasão de um aluno com base em seu desempenho acadêmico, utilizando técnicas de mineração de dados e aprendizado de máquina, para então prover uma ferramenta que auxilie o corpo docente a observar dados de discentes e facilitar uma tomada de decisão que previna o aluno de abandonar a graduação.

Palavras chave: Evasão. Aprendizado de máquina. Predição. Mineração de dados.

ABSTRACT

This project has the objective to develop a system that is capable of predicting the evasion of students using their academic performance data, utilizing data ming and machine learning techniques and provide a tool that supports the teachers observe and make it easier the decision making that prevents the student from dropping of college.

Keyword: Evasion. Machine learning. Prediction. Data mining.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
AVA	Ambiente Virtual de Aprendizagem
BD	Banco de Dados
CSV	Comma-separated values
DTL	Django Template Language
DW	Data Warehouse
EDM	Educational Data Mining
HTML	HyperText Markup Language
IA	Inteligência Artificial
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
IRA	Índice de Rendimento Acumulado
KDD	Knowledge-discovery in databases
KNN	K-nearest-neighbor
MEC	Ministério da Educação
MLP	Multi-layer Perceptron
MVC	Model, View, Controller
MVT	Model, View, Template
ORM	Object relational Mapping
RNA	Redes Neurais Artificiais
SEAS	Student Evasion Analysis System
SEF	Secretaria de Educação Fundamental
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
STI	Sistemas Tutores Inteligentes
TADS	Tecnologia em Análise e Desenvolvimento de Sistemas
UFPR	Universidade Federal do Paraná
WEB	World Wide Web
XML	Extensible Markup Language

SUMÁRIO

1. INTRODUÇÃO

O ensino superior é o nível de formação iniciado após a conclusão do ensino médio, e dá ao aluno a formação em uma área específica de conhecimento, que permite o desempenho de uma profissão que exija uma formação própria.

> A formação superior em nível de graduação tem uma boa diversidade de finalidades, sendo a mais geral e abrangente a formação de cidadãos informados e responsáveis, capazes de satisfazer, por meio de suas carreiras profissionais, as expectativas pessoais e da sociedade. Nesse nível de abrangência, o papel da educação superior é atender às demandas de (a) preparação para o exercício da cidadania ativa (em sociedades democráticas), (b) preparação para o mercado de trabalho, (c) desenvolvimento pessoal e (d) desenvolvimento e manutenção permanentemente atualizada de uma base de conhecimentos. (COSTA et al., 2017, p. 13)

A graduação no ensino superior mostra-se uma realização muito importante para a sociedade, pois oferece ao cidadão a possibilidade de uma formação muito mais avançada que os níveis de ensino anteriores. Visto que o mercado de trabalho exige, a cada dia que passa, níveis de formação cada vez mais altos, a graduação é um dos requisitos mínimos para acessar vagas mais qualificadas (MARTINS; OLIVEIRA, 2017, p. 33).

O censo sobre a educação superior produzido pelo INEP/MEC (2018) registrou que 8.286.663 de pessoas se matricularam em cursos de graduação, tanto em organizações públicas quanto privadas, número este, que sempre aumenta a cada estudo realizado. Mas, apesar do grande número de inscritos nas instituições de ensino superior, o número de pessoas que alcança a conclusão de seu curso, é de longe, bem menos expressiva.

Isso acontece devido a diversos fatores, um deles é a falta de acompanhamento do corpo discente por parte de seus docentes. Por isso este trabalho propõe o desenvolvimento de um sistema que facilita esse acompanhamento.

1.1. PROBLEMA

A evasão e a retenção de alunos nos cursos superiores são problemas que afetam não somente alunos e alunas, mas todo a sociedade. A evasão

caracteriza-se pelo fenômeno em que um aluno abandona o curso em que está matriculado e não obtém diploma (FILHO, 2007). Retenção, por sua vez, é um acontecimento em que o estudante precisa de mais tempo que o previsto no calendário acadêmico para concluir a graduação (MEC/SEF, 1997, p. 23). Fenômeno este que além de trazer malefícios a sociedade, como o retardamento do ingresso de profissionais qualificados no mercado de trabalho, pode também acarretar em uma possível evasão.

Diversos são os fatores que podem levar à evasão e retenção acadêmica, Marques e Silva (2017) citam alguns exemplos como falta de afinidade com o curso, indisponibilidade de horário, situações financeiras, entre outros, apesar de muitos serem fatores externos à faculdade existem também os internos como a dificuldade de integração acadêmica e adaptação do estudante ao curso, descompromisso dos docentes com o curso, bem como deficiências didático-pedagógicas dos mesmos.

Devido a esses fatores, o número de evasão no ensino superior tem aumentado cada vez mais ao longo dos anos. De acordo com o INEP/MEC (2019), o número de matrículas desvinculadas nas universidades foi de 2.187.411 somando as públicas e privadas, e tendo em vista que, no ano anterior, o mesmo censo registrou 8.286.663 de alunos matriculados, tem-se 26% de matrículas desvinculadas.

É proposto então, desenvolver um sistema que centraliza os dados dos discentes de um curso de ensino superior, auxiliando o corpo docente a analisar esses dados de forma clara e objetiva, e que sinaliza com antecedência possíveis evasões para que professores consigam centralizar esforços em alunos e alunas em situação de evasão, e, assim, aprimorar o processo de ensino dos estudantes.

1.2. OBJETIVO GERAL

Desenvolver e documentar um sistema que analisa e gera predições sobre o corpo discente, e disponibiliza os dados gerados para o corpo docente, para que estes tenham maior compreensão sobre os alunos e suas dificuldades, com o objetivo de tornar as tomada de decisões mais fáceis, aperfeiçoadas e específicas.

1.3. OBJETIVOS ESPECÍFICOS

- Modelar e desenvolver um sistema que auxilie o corpo docente a visualizar e interpretar dados referentes à evasão/retenção dos estudantes do TADS;
- Desenvolver um algoritmo que utiliza inteligência artificial para compor uma análise dos dados do corpo discente;
- Disponibilizar esta análise e a predição gerada para o corpo docente utilizando uma interface web acessível e de fácil entendimento;
- Documentar o sistema desenvolvido.

1.4. JUSTIFICATIVA

Muitas vezes os docentes não fazem um acompanhamento contínuo dos discentes durante o curso. Uma das razões para isso é pela não disponibilização dos dados dos alunos e alunas de uma maneira de fácil acesso a professores e funcionários. Em geral, o acesso a esses dados é realizado por vias informais ou não sistematizadas (relatórios individuais gerados pela secretaria do curso, conversa com alunos, entre outros).

No curso superior de Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná (UFPR), de 2005 a 2018, de 568 alunos que concluíram o curso, apenas 157 conseguiram se formar nos 6 semestres previstos, ou seja, o índice de retenção do TADS chega a 72,3%. Estes números são preocupantes e apontam para um problema maior por trás do processo pedagógico do curso. A evasão e a retenção levam a desperdício de recursos humanos e financeiros para a instituição, provoca danos à sociedade por retardar o ingresso de profissionais qualificados no mercado de trabalho e pode levar o estudante a ter problemas ao longo do desenvolvimento de sua carreira, dificultando sua ascensão financeira e intelectual em meio a sociedade em que vive.

Atualmente, devido a vasta popularização da rede de computadores, interligação de sistemas e utilização de sistemas WEB, uma vasta gama de dados é

produzida e coletada diariamente, em diversos setores. O âmbito educacional é uma destas áreas, mas somente um aglomerado de dados brutos não se faz eficiente para a extração de conhecimento. É preciso reverter estas informações em benefícios para as instituições e para tanto, pode-se fazer o uso da mineração de dados. A área de mineração de dados para a educação ganhou tanta relevância que ganhou sua própria sigla, a EDM - *Educational Data Mining*.

> A área emergente de Mineração de Dados Educacionais procura desenvolver ou adaptar métodos e algoritmos de mineração existentes, de tal modo que se prestem a compreender melhor os dados em contextos educacionais, produzidos principalmente por estudantes e professores, considerando os ambientes nos quais eles interagem, tais como AVAs, Sistemas Tutores Inteligentes (STIs), entre outros. (COSTA, BAKER, AMORIM, MAGALHÃES, & MARINHO, 2013)

Portanto, a proposta deste trabalho é a criação de uma ferramenta que realiza análise dos dados de alunos e ex-alunos do TADS e crie um modelo de predição que facilite com que os professores possam ter acesso a previsões sobre o comportamento de atuais alunos e possam então tomar decisões antecipadas, baseadas na análise de fatos e histórico do curso, para prevenir que os números de retenção e evasão continuem no nível atual.

2. FUNDAMENTAÇÃO TEÓRICA

Evasão e retenção escolar possuem diversas definições e interpretações diferentes entres os autores, não sendo algo de classificação exata, tampouco algo que possui consenso em sua nomenclatura, sem objetivo claro em suas causas e motivações (FILHO; ARAÚJO, 2017, pg 35). Quanto ao que se tem unanimidade, é que tanto a evasão quanto a retenção escolar representam um grande problema para a educação brasileira.

Neste capítulo, descrevem-se as definições de evasão e retenção escolar que foram adotadas para o desenvolvimento do projeto, além de adentrar em suas possíveis consequências e impactos na educação superior e mercado de trabalho. Descrevem-se também os algoritmos de aprendizado de máquinas estudados ao longo do trabalho, suas distinções e características.

2.1. A EVASÃO ESCOLAR

Evasão, segundo o dicionário Aurélio de língua portuguesa, é a ação de abandonar algo, desistência, abandono (AURÉLIO, 2019). Para Riffel e Malacarne (2010), de evasão escolar entende-se a “fuga ou abandono da escola em função da realização de outra atividade”. Para o INEP (1998), evasão e abandono possuem definições distintas. Abandono seria a situação em que o aluno desliga-se da instituição mas retorna em um período seguinte, já a evasão seria o ato de desistir o sistema escolar e não retornar mais.

Outros autores vão além e classificam a evasão e o abandono como “fracasso escolar” e afirmam que não existe uma única causa raiz para se entender as motivações da desistência, mas sim a soma de diversos fatores sociais, indo muito além da simples necessidade de trabalho, desestruturação familiar ou problemas na formação da educação de base (MACHADO, 2001).

Para o desenvolvimento deste estudo, o sentido adotado para o termo evasão necessitava ser uma classificação não abstrata, passível de ser quantificada, para que a aplicação de algoritmos de aprendizado de máquinas pudessem ser realizados. Para tanto, optou-se por definir o termo evasão como descrito por Silva

Filho (2007), que estabelece evasão como *a medida total do número de alunos que tendo entrado em um determinado curso de ensino superior, não obteve diploma no final do curso.*

2.2. RETENÇÃO ESCOLAR

A retenção escolar é caracterizada pela situação em que um estudante do ensino superior demanda um tempo superior ao previsto do calendário pedagógico para concluir o curso em que se matriculou (MEC/SEF, 1997, p.23). A retenção não é um fenômeno benéfico para o cenário da educação brasileira pois, além de poder levar a evasão, acarreta também em retardar o ingresso de um trabalhador qualificado no mercado de trabalho.

Além disso, retenção escolar leva a sobrecarga de recursos na instituição que abriga o aluno. Alguns autores, como Pereira et al. (2015), afirmam que a retenção escolar no ensino superior deve ser considerada um grave problema no processo de ensino por causar problemas não só para o estudantes, que podem ter prejuízos na vida pessoal, de natureza sociológica, por considerar a dificuldade para terminar a faculdade um fracasso pessoal (CAVESTRO; ROCHA, 2006), de ordem profissional, por causar atraso no crescimento de sua carreira e também financeiro, problema este que por si, acarreta em outra cadeia de problemas no decorrer da vida.

A retenção causa problemas também dentro da instituição por comprometer a produtividade do sistema de ensino, dificultar o planejamento da distribuição de recursos humanos e financeiros, criar vagas ociosas e alongamento de grades curriculares ultrapassadas (PEREIRA et al., 2015). O último a ser afetado seria a sociedade em si, pois a retenção retarda o retorno social e a disponibilização de cidadãos profissionais graduados no ensino superior para o mercado de trabalho.

Em um caso específico, no curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas da UFPR, foi observado que de 2005 a 2018, 72.3% dos alunos formados precisaram de mais de 6 períodos para se formar. Isso evidencia a existência de um grave problema a ser estudado em tal curso. Nesse sentido, com a ajuda das técnicas de aprendizado de máquinas, os dados históricos

de todos os alunos que frequentaram e frequentam o curso até então podem contribuir para o auxílio a tomadas de decisão do corpo docente. Um possível auxílio seria uma ferramenta que os possibilidade agir antes que um aluno ou uma aluna que caminha para uma retenção expressiva não se torne mais um caso de evasão.

Dessa forma, para a produção deste trabalho, considera-se retidos todos os alunos que precisaram de no mínimo 1 semestre a mais que os previsto para a conclusão ideal do curso.

Uma vez determinados os conceitos de evasão e retenção, é necessário elucidar conceitos relacionados ao desenvolvimento de sistemas capazes de realizar predição de informações com base na análise de dados educacionais. Assim, os conceitos de *Data Mining* e *Machine Learning* são descritos nas seções 2.3 e 2.4.

2.3. DATA MINING

De acordo com Sidhu et al. (2007) *data mining* é o trabalho, geralmente conduzido em áreas com grandes quantidades de dados, de analisar e identificar padrões no interior dos dados, o que pode eventualmente se tornar conhecimento.

Outros autores como Sahu et al. (2011) descrevem o *data mining* como o uso automático de técnicas de análise de dados para a descoberta de relações previamente desconhecidas entre dados. Esse mesmo autor também descreve a relação entre *Knowledge Discovery in Databases* (KDD) e o *data mining*. Mesmo frequentemente os dois sendo tratados como sinônimos o *data mining* na verdade é um dos processos do KDD.

Os processos do KDD são descritos por Sahu et al. (2011) da seguinte forma:

- Limpeza dos dados: quando acontece a limpeza dos dados irrelevantes do conjunto de dados;
- Integração dos dados: onde múltiplos dados de diferentes origens se juntam em uma única origem;
- Seleção de dados: etapa que os dados relevantes para se fazer uma análise são escolhidos e recuperados;

- Transformação de dados: quando os dados selecionados tomam uma forma adequada para os processos de *data mining*;
- *Data mining*: onde técnicas e procedimentos são utilizados para reconhecer padrões nos dados;
- Avaliação de padrões: baseando-se em certas medidas padrões que representam conhecimentos são identificados;
- Representação do conhecimento: etapa final que apresenta o conhecimento obtido para o usuário.

Muitas vezes essas etapas são realizadas em conjunto. Um exemplo comum disso é a junção das etapas de limpeza e integração para a formação de um *data warehouse*, estes são definidos por Chaudhuri e Dayal (1997) como uma coleção de dados integrado, orientado ao contexto, com variação de tempo e não volátil utilizada principalmente para tomadas de decisão em organizações.

A etapa de *data mining* pode ser realizada com quatro tipos de algoritmos definidos por Kaur e Singh (2017) da seguinte forma:

- Árvores de classificação: Técnica utilizada para classificação de uma variável categórica de um dado a partir de outras variáveis, o resultado disso é uma árvore de decisão;
- Regressão logística: Variante da técnica de regressão estatística com o conceito expandido para lidar com classificação, gerando uma fórmula matemática que prevê a probabilidade de ocorrência de dados;
- Redes Neurais: Baseado na arquitetura de cérebros animais, esse conceito foi definido por Ferneda (2006) como, modelos que buscam simular o processamento de um cérebro humano, utilizando unidades de processamento simples, chamadas de neurônios, que se unem através de conexões sinápticas;
- Agrupamento: Uma técnica que cria agrupamentos baseando-se na proximidade dos dados relacionados.

Existem diversas implementações desses tipos de algoritmos e todos eles fazem parte da área de inteligência artificial (IA) chamada *machine learning* ou aprendizado de máquina.

2.4. MACHINE LEARNING

Machine learning ou aprendizado de máquina é um dos campos que chamamos de inteligência artificial (IA). Seu objetivo consiste em compreender uma estrutura de dados, os utilizar como treinamento em um modelo útil que pessoas possam entender e utilizar, transformando dados antes abstratos em conhecimento, podendo então interpretar novos dados e classificá-los de maneira apropriada a partir de uma generalização do que lhe foi apresentado anteriormente (TAGLIAFERRI et al, 2019).

Existem três paradigmas que subdividem o *Machine Learning*: Aprendizado Supervisionado, Semi Supervisionado e Não Supervisionado. Estes diferem-se em como os dados de treinamento são utilizados nos modelos. Essas categorias são definidas por Simeone (2018):

Aprendizado supervisionado é utilizado quando os dados de treinamento são subdivididos em dados entrada (*input*) e resultado desejado (*output*), dessa forma o sistema pode ser comparado constantemente ao resultado esperado para aprimorar-se, e então construir um classificador que tenha uma boa chance de alcançar resultados semelhantes com outros dados.

O aprendizado não supervisionado utiliza dados que não possuem os *outputs* desejados para o treinamento, ou seja, que não estão previamente classificados. Esses modelos fazem um agrupamento dos dados que possui e ao classificar procura em qual agrupamento o novo conjunto de dado se encontra. Geralmente focam em descobrir propriedades do mecanismo que gera os dados.

Já a categoria semi supervisionado, *reinforcement learning* ou aprendizagem por reforço, se encontra entre as outras duas, pois possuem características de não supervisionados, porém possuem supervisão por meio de feedbacks que apontam estimativas de acertos ou erros baseados nas *inputs*/*outputs*.

2.4.1. Scripts Estimadores

Scripts estimadores ou *estimators* são os modelos que recebem o treinamento e podem classificar novos dados. O projeto utiliza apenas os *scripts*

supervisionados, já que os dados possuem o resultado desejado para serem utilizados no treinamento dos modelos.

De acordo com a documentação da *scikit-learn* (2010), biblioteca em python utilizada no projeto, os quatro modelos utilizados nos testes da aplicação são descritos da seguinte forma:

- KNN ou *K-nearest-neighbor* pode ser utilizado tanto como um *script* supervisionado como não supervisionado. Seu princípio é simples: para classificar um dado procura-se um número predefinido de "vizinhos", que são previamente inseridos durante o treinamento e, então, baseia-se neles para definir a classificação do novo dado;
- *Gaussian Processes Naive Bayes* utiliza um processo mais complexo para a encontrar a relação probabilística dos dados com sua classificação. Utiliza como base o teorema de Bayes;
- *Decision Trees* ou árvores de decisão tem um objetivo simples: criar um modelo que prediz o valor aprendendo regras de decisão simples dos dados de treinamento. Esse algoritmo também permite a plotagem das regras que são utilizadas;
- *Multi-layer Perceptron* (MLP) segue o conceito de redes neurais (RNA), e algoritmos baseados nisso possuem um funcionamento bem parecido. Kaur e Singh (2017) explicam seu funcionamento da seguinte forma: A rede é constituída de camadas de nódulos, a primeira camada é a de entrada, seguida de uma camada invisível e por último a camada de saída. Cada nódulo tem um peso atribuído e, por tentativa e erro, o algoritmo altera esse peso até que se atinja um critério de parada.

3. MATERIAIS E MÉTODOS

Neste Capítulo, é explicado as metodologias de desenvolvimento de projeto, bem como o fluxo de trabalho empregado no decorrer da implementação da pesquisa, as ferramentas, frameworks e bibliotecas utilizadas.

3.1. O MÉTODO DE DESENVOLVIMENTO E ACOMPANHAMENTO DE TAREFAS

A equipe foi formada por membros multifuncionais e desenvolvedores *FullStack*. Todos os membros contribuíram ativamente na tomada de decisão das tecnologias e algoritmos utilizados para a concepção do projeto. Entretanto, focos específicos foram designados aos membros, como *front-end*, *back-end*, infra-estrutura e documentação.

A equipe inicialmente era constituída de dois membros. Devido a conflitos de horário e divergências de agenda, optou-se no início por adaptar as reuniões diárias para semanais, feitas presencialmente junto com o orientador, ou via *software*, sem a presença do mesmo. Ao longo do desenvolvimento do projeto, a equipe integrou mais um membro.

A equipe utilizou estas reuniões digitais para manter-se ativamente integrada em relação ao andamento do projeto e das tarefas individuais de cada membro, além de em conjunto tomar decisões de priorização de atividades.

Todo e qualquer documento ligado direta (código-fonte, diagramas, documentação) ou indiretamente (referências, artigos úteis, etc) ao projeto foi disponibilizado na nuvem, com possibilidade de edição conjunta em tempo real.

Todos os diagramas referentes a modelagem do sistema SEAS encontram-se nos apêndices deste documento.

Não foram utilizadas ferramentas de auxílio a visualização do andamento de tarefas, como um *Kanban*, por exemplo. A equipe optou por manter-se um diálogo fluido e constante, deixando a critério de cada membro se organizar com suas tarefas.

3.2. A METODOLOGIA DE VERSIONAMENTO DE CÓDIGO FONTE

Desde o início do projeto optou-se por realizar versionamento de código-fonte distribuído baseado em GIT.

GIT é um sistema de versionamento de código distribuído, livre e *open source*, desenhado para lidar com projetos de pequena a larga escala com eficiência e velocidade (GIT, 2019).

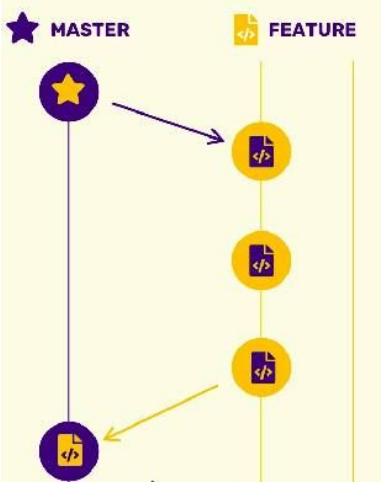
O objetivo de usar um sistema de versionamento de código é garantir integridade entre as mudanças, guardando informações relevantes a cada mudança, como quem, quando, porque e o que isto altera (LOELIGER; MCCULLOUGH, 2012).

Em versionamento GIT, dois conceitos foram muito importantes para o versionamento correto do desenvolvimento do projeto, são eles *Commits* e *Branches*.

Commits são metadados enviados junto a cada alteração de código fonte, que incluem uma mensagem, autor, data e uma hash-SHA1 imutável que salva um snapshot do estado do projeto (CHACON; STRAUB, 2014).

Branches são divisões lógicas do fluxo do projeto, que trabalham como ponteiros que apontam para *commits* específicos e possibilitam que alterações enviadas ao versionamento não alterem de imediato a versão estável do código, podendo serem fundidas quando estáveis a versão estável do projeto (CHACON; STRAUB, 2014). Em GIT, existe uma branch padrão chamada de master, onde o código estável é salvo e versionado.

FIGURA 1 - FLUXO ULTRA-SIMPLIFICADO DE FEATURE-BRANCH



Fonte: Gabriel Stocco, adaptado (2019).

O fluxo GIT utilizado no projeto consistiu na criação de *feature branches* que derivam da master. Tais *branches* eram desenvolvidas localmente e depois fundidas novamente a master quando se tornavam estáveis, como mostra a Figura 1.

3.3. TECNOLOGIAS UTILIZADAS

As tecnologias escolhidas para o desenvolvimento do projeto não se adéquam somente as necessidades deste trabalho mas também seguem uma tendência atual de linguagens mais utilizadas no mundo. Python, a linguagem de programação motriz desta pesquisa, foi eleita pelo terceiro ano seguido uma das linguagens de programação mais amadas e desejadas no mundo, em uma pesquisa com 87,354 respostas, realizada pelo *Stack Overflow*, famoso site de perguntas e respostas sobre tecnologia, em 2019 (STACK OVERFLOW SURVEY, 2019).

Python é uma linguagem de programação *open source*, interpretada, não tipada, multi-paradigma, idealizada por Guido van Rossum no final dos anos 80, com o objetivo de ser uma linguagem com curva de aprendizado curta, de leitura simples e fácil entendimento, com alta propensão a extensão por meio de sua comunidade e com performance razoável. Python é umas das linguagens mais amadas do mundo (STACK OVERFLOW SURVEY, 2019), não só por sua facilidade mas por sua comunidade de desenvolvedores apaixonados por boas práticas e agilidade. Nos últimos anos, a comunidade de data-mining adotou esta linguagem como uma das principais ferramentas de desenvolvimento de soluções nesta área, devido a facilidade em se usar bibliotecas e APIs de alto nível, que possibilitam a criação de gráficos, cálculos avançados e visualização de dados, como a biblioteca *Pandas*, *NumPy*, entre outras.

O SGBD (Sistema Gerenciador de Banco de Dados) escolhido foi o Mysql, em sua versão estável 5.7. Mysql é um dos mais populares sistemas de banco de dados, criado em 1995 (WIDENIUS, 2009), é mantido e atualizado até hoje. Foi escolhido para este projeto pelo seu alto nível de maturidade, performance e desempenho.

Como *framework* de desenvolvimento web, optou-se pela utilização do Django, um framework python de alto nível de código aberto e colaborativo, que

permite fácil gerenciamento do ambiente de desenvolvimento, possui módulos de segurança nativos que evitam erros comuns de desenvolvimento no controle da segurança da aplicação, é portátil e escalável (MOZZILA.ORG, 2019).

Como biblioteca de *Machine Learning*, utilizou-se o *scikit-learn*, uma biblioteca em python que implementa diversos algoritmos sobre aprendizado de máquina, como funções de preparação de dados, algoritmos estimadores, até algoritmos que medem a acurácia de uma estimação.

Para a criação dos gráficos apresentados na tela de dashboard, representado na Figura 6 presente no seção 4.2.1, utilizou-se a biblioteca *highcharts.js*. Para a construção de tabelas dinâmicas a biblioteca escolhida foi a *DataTable*.

Para a criação do *front-end*, além da utilização das bibliotecas JavaScript citadas acima, utilizou-se também a linguagem de templates Jinja2, que é uma linguagem criada para se utilizar com Python, provendo a utilização de blocos de lógica para a filtragem de dados dentro de arquivos estáticos como HTML ou XML.

A identidade visual do sistema foi adaptada a partir de um template *open source* de WebApp criado para Dashboards chamado de *AdminLTE 2*. Um template de HTML responsivo baseado no *framework Bootstrap 3* e *Jquery*.

Além disso, uma ferramenta que foi de grande importância e que foi utilizada desde o início do projeto para o configuração do ambiente de desenvolvimento, portabilidade e gerenciamento de dependências foi o Docker Community Edition, 19.03. Docker é uma ferramenta para a criação de orquestração de Containers Linux, que possibilita que o ambiente de desenvolvimento se torne imutável a partir da criação de imagens de sistema, facilitando muito o processo de portabilidade sem a preocupação de gerenciamento de dependências de ambiente. Neste trabalho, docker foi utilizado vastamente para provisionar, construir e destruir bancos de dados Mysql, além de servir para construção de uma imagem base com toda a configuração necessária para se rodar o ambiente Django.

3.4. MATERIAIS DISPONÍVEIS

No início do projeto, foi disponibilizado para a equipe um arquivo contendo dados anonimizados de todos os alunos que cursaram o TADS na UFPR, datando desde o início do curso, em 2005, até o primeiro semestre de 2018. Este arquivo continha detalhes de todas as matrículas feitas neste período. Destes detalhes os mais relevantes consistiram em:

- Hash-SHA1 anonimizada da identificadora do aluno;
- Curso do aluno;
- Forma de ingresso;
- Data de ingresso;
- Forma de evasão;
- Data de evasão;
- HASH-SHA1 anonimizada do nome da disciplina;
- HASH-SHA1 anonimizada do código da disciplina
- Carga horária da disciplina que o aluno se matriculou;
- Média final obtida;
- Faltas que o aluno teve na disciplina;
- Período em que a matrícula foi feita.
- Estado atual da matrícula.

Após o recebimento do arquivo que seria utilizado para a criação do sistema a equipe precisava compreender os dados e então decidir quais iriam ser utilizados. Assim, definiu-se o dicionário de dados apresentado no Quadro 1.

QUADRO 1 - DICIONÁRIO DE DADOS

Dado histórico	**Descrição**
ID_ALUNO	Identificador do aluno no sistema
ANO_INGRESSO	Ano do ingresso do aluno no curso
PERIODO_INGRE_ITEM	Período do ingresso do aluno no curso 201 - 1º semestre 202 - 2º semestre
ANO_EVASAO	Ano da evasão do aluno
PERIODO_EVA_ITEM	Período da evasão do aluno 201 - 1º semestre 202 - 2º semestre

COD_CURSO	Identificador do curso
COD_ATIV_CURRIC	Identificador da disciplina
NOME_ATIV_CURRIC	Nome da disciplina
PERIODO_ITEM	Período da matrícula de um aluno em uma disciplina 201 - 1º semestre 202 - 2º semestre
CH_TOTAL	Carga horário total de uma disciplina
FORMA_EVASAO_ITEM	Forma de evasão de um aluno
FORMA_INGRE_ITEM	Forma de ingresso de um aluno
MATR_ALUNO	Identificador da matrícula do aluno em curso (GRR)
MEDIA_FINAL	Média final de um aluno em uma matricula em uma disciplina
NUM_FALTAS	Pendências de um aluno nas aulas em uma matricula em uma disciplina
ANO	Ano da matrícula de um aluno em uma disciplina
SITUACAO_ITEM	Situação ou Resultado de uma matrícula de um aluno em uma disciplina

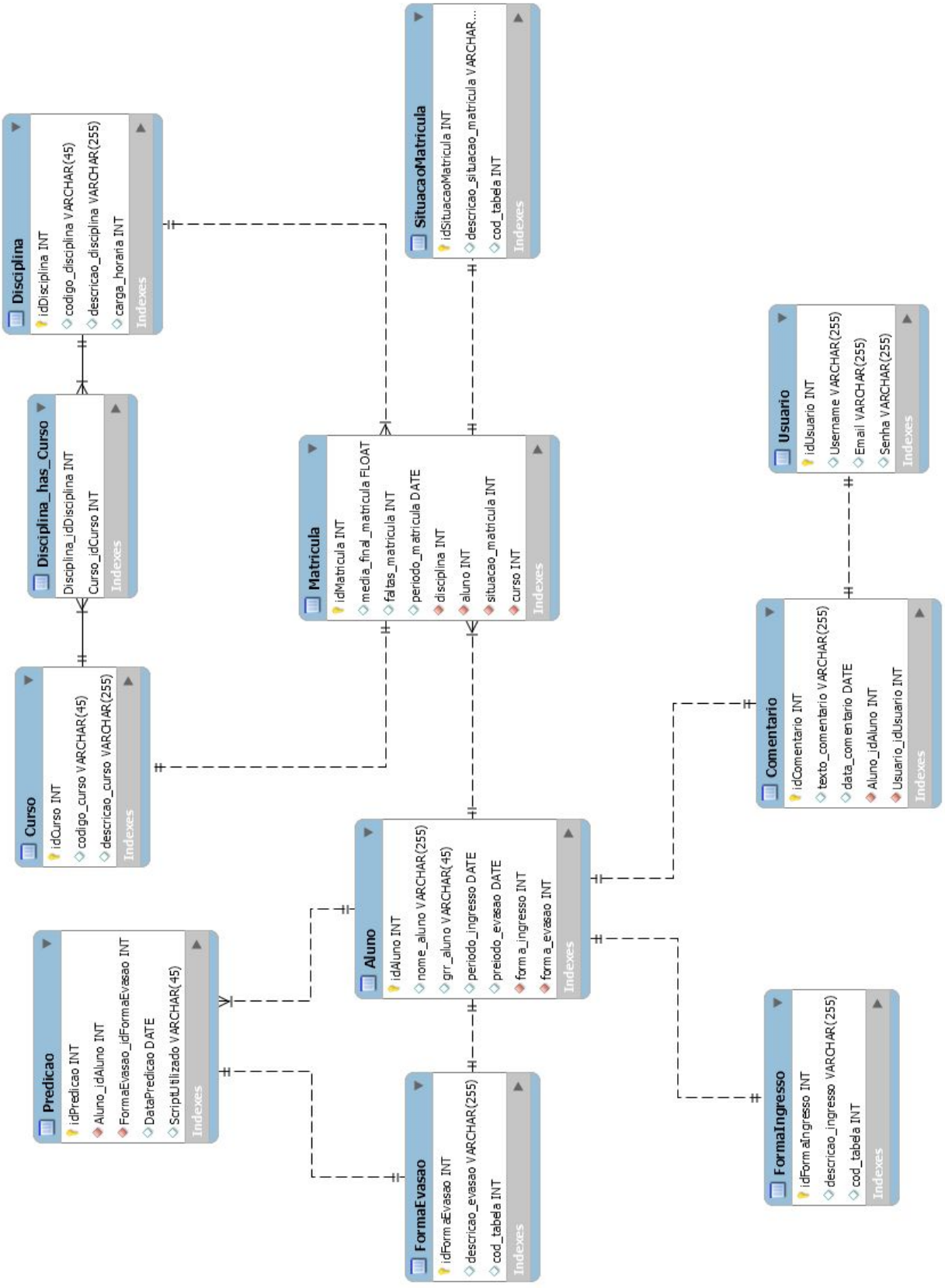
Fonte: Os autores (2019).

3.5. TRANSFORMAÇÃO E LIMPEZA DE DADOS

Para iniciar o desenvolvimento do *data mining* a equipe ainda precisava realizar os três primeiros passos do KDD, que são limpeza, integração e seleção dos dados que serão utilizados. A limpeza ocorreu no primeiro momento do desenvolvimento, junto com a importação dos dados disponibilizados para um banco de dados.

Neste banco de dados tem-se os mesmos dados dos arquivos, sem nenhuma transformação. Entretanto, a necessidade de algumas novas tabelas surgiu durante o desenvolvimento do sistema. Nesse sentido, foi adicionada a tabela que contém dados produzidos pelo sistema pelos scripts de predição, nomeada “Predição”, e a tabela nomeada de “Comentário” que é onde se salva os comentários criados pelos usuários a respeito de um aluno. Tal tabela contém os dados da estimativa realizada como resultado, data e em que *script* foi feito, como mostra a Figura 2.

FIGURA 2 - DIAGRAMA ENTIDADE RELACIONAMENTO - BANCO RELACIONAL



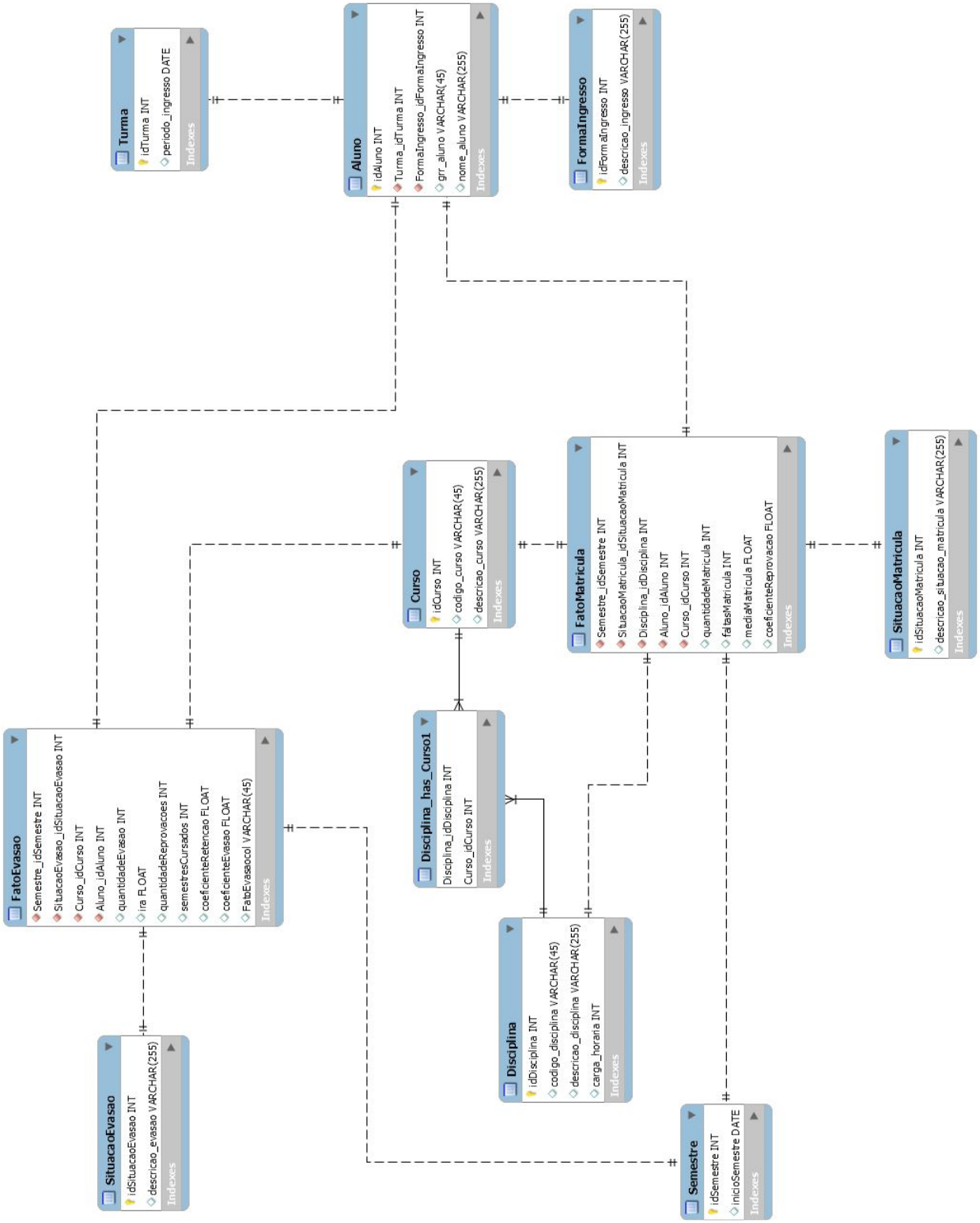
Fonte: Os autores (2019).

Para a inserção dos dados neste banco foi criado um *script* em python que lê o arquivo original dos dados dos discentes e os insere no banco. Na etapa de conclusão do desenvolvimento do trabalho, esse *script* foi refatorado em classes e funções para que pudesse ser reaproveitado para a inserção de novos dados.

Após a importação do arquivo em um banco de dados, iniciou-se o desenvolvimento do segundo passo do KDD, a integração dos dados. Este passo é a transformação dos dados existentes em novos que serão utilizados no *data mining* do sistema.

A equipe decidiu armazenar esses dados em um novo banco. Porém, este segundo banco não segue um modelo relacional como o primeiro. O segundo banco de dados segue o conceito chamado de *Data Warehouse* (DW), como é apresentado na Figura 3.

FIGURA 3 - DIAGRAMA ENTIDADE RELACIONAMENTO - DATA WAREHOUSE



Fonte: Os autores (2019).

Como demanda o conceito de *data warehouse*, de acordo com Chaudhuri e Dayal (1997) sua estrutura geralmente é multidimensional e tem dois elementos essenciais: as tabelas fatos e as tabelas dimensões.

Uma tabela fato é o acontecimento em si, é onde os dados calculados são salvos, as *foreign keys* desta tabela representam uma tabela dimensão, que por sua vez representam uma característica de um evento. A relação de uma tabela dimensão com a tabela fato é de “um para muitos”. De acordo com Kimball e Ross (2002) A tabela fato contém dois tipos de colunas que representam a dimensão, colunas preenchidas, em que seu valor indica uma característica (dimensão) do fato e colunas nulas, que indicam que a fato representa um aglomerado de acontecimentos.

O *data warehouse* construído possui duas *tabelas fato*: “fatoMatricula” e “fatoEvasao”, e algumas tabelas dimensões, que são utilizadas para navegação e pesquisa dos dados, sendo uma delas uma dimensão que representa a data em que as fatos ocorreram que é a dimensão “semestre”.

A tabela “fatoMatricula” é construída para representar as matrículas de um aluno que já foram completas, portanto suas dimensões são "Disciplina", "Aluno", "Curso", o resultado desta matrícula chamado de "situacaoMatricula" e a dimensão de tempo. Os atributos calculados são:

- "quantidadeMatricula": quantidade de matrículas que uma instância representa. Em instâncias que representam mais de uma fato acontece a soma de todas as instâncias representadas;
- "faltasMatricula": quantidade de faltas do aluno nas aulas da disciplina. Em instâncias que representam mais de uma fato esse atributo representa a média de faltas das instâncias representadas;
- "mediaMatricula": nota da matrícula representada. Em instâncias que representam mais de uma fato esse atributo representa a média das notas;
- "coeficienteReprovacao": porcentagem de reprovação da matrícula representada. Em instâncias que representam mais de uma fato esse atributo é a média das porcentagens.

A "fatoEvasao" é construída quando acontece a evasão de um aluno. Suas dimensões são “Aluno”, “Curso”, “situacaoEvasao” e a dimensão tempo. Os atributos calculados são:

- “quantidadeEvasao”: quantidade de matrículas no curso que uma instância representa. Em instâncias que representam mais de uma fato este atributo é a somatória desse mesmo atributo;
- “quantidadeReprovacoes”: soma da quantidade de reprovações da instância de evasão. Em instâncias que representam mais de uma fato este atributo é a média das quantidades.
- “ira”: média dos IRA’s (Índice de Rendimento Acumulado) de cada tupla de evasão. De acordo com o Conselho de Ensino, Pesquisa e Extensão (1997) o cálculo do IRA segue a seguinte fórmula “IRA = Somatório (nota x c.h. da disciplina cadastrada no Histórico Escolar do aluno) dividido pela carga horária total cadastrada no Histórico Escolar do aluno”. Em instâncias que representam mais de uma fato este atributo é a média dos IRA’s.
- “semestresCursados”: Quantidade de semestres cursados. Em instâncias que representam mais de uma fato este atributo é a média dos semestres.
- “coeficienteEvasao”: Porcentagem de evasão da tupla. Em instâncias que representam mais de uma fato este atributo é a média das porcentagens.
- “coeficienteRetencao”: Porcentagem de formandos que ficaram retidos. Em instâncias que representam mais de uma fato este atributo é a média das porcentagens.

Então, com o segundo banco de dados pronto, as etapas de limpeza e integração dos dados estão prontas.

Para a construção desse *data warehouse* criou-se um *script* em python que faz a criação das tuplas, realizando os cálculos necessários, das instâncias que representam uma ou várias fatos. Ao final do desenvolvimento esse script foi refatorado para que fosse possível inserção de novos dados.

3.6. PREDIÇÃO

Após a construção do segundo banco, elucidado na Seção anterior, é dado início ao desenvolvimento do terceiro passo do projeto: a criação de um *script* que realiza previsões de evasão para alunos matriculados a partir dos dados gerados. Esta etapa do desenvolvimento englobaria as etapas finais do KDD, seleção, preparação e *data mining*.

Para auxiliar em tal tarefa foram selecionadas duas bibliotecas para desenvolvimento com estimadores, a *scikit-learn* e *pandas*. Com essas duas bibliotecas juntas não seria necessário a codificação de algoritmos estimadores, pois estes já encontram-se implementados pelas bibliotecas. O desafio nesta etapa seria, então, a escolha dos melhores estimadores e seus atributos.

Para auxiliar na escolha dos atributos e do estimador, no início dessa etapa do desenvolvimento, foi criado um *script* que realiza o teste do estimador e apresenta os resultados obtidos. A lógica desse *script* consiste em :

- Recuperar dados do DW;
- Preparar lista de atributos utilizados no teste. Isso inclui retirar dados que não seriam utilizados no teste e a normalização dos dados. Alguns testes foram realizados sem a normalização e a acurácia do estimador foi entre 1% e 3% menor;
- Separação dos dados de treinamento e de teste;
- Treinamento do estimador selecionado com os dados de treinamento;
- Classificação dos dados de teste;
- Apresentação dos resultados.

A recuperação dos dados do DW teve de ser feita com cautela, pois os dados que devem representar apenas um acontecimento de uma fato evasão e, como DWs possuem tuplas que representam mais de um acontecimento, isso poderia afetar a predição. A normalização dos dados é feita com a mesma biblioteca que implementa os estimadores, pois de acordo com a sua própria documentação alguns dos modelos não teriam a mesma acurácia. Após a normalização, é realizado a separação dos dados em duas listas, 70% dos dados são dedicados a lista de

treinamento e os 30% restantes a de teste, em alguns testes essa proporção foi ajustada para 90% e 10%.

Após a preparação dos dados inicia-se o treinamento do estimador escolhido e logo após a classificação dos dados de teste. O resultado da estimativa gerada pelo estimador é comparada com o resultado desejado que os dados de teste possuem e, assim, tem-se a quantidade de acertos, de erros, de falsos-negativos e de falsos-positivos. É possível, assim, o cálculo da acurácia do estimador treinado.

Nos testes deste projeto os seguintes estimadores foram utilizados: *K nearest neighbor* (KNN), *Gaussian Process Naive Bayes*, *Decision Tree* e *Multi-layer perceptron* (MLP). Contudo, no início dos testes apenas o KNN foi utilizado. Isso se deu pois a equipe deu prioridade na definição dos atributos do que a quantidade de resultados.

Para a definição dos atributos foram utilizados diversos fatores. A princípio a metodologia inicial foi utilizar os dados gerais de uma evasão, como quantidade de retenções e IRA, porém nos primeiros testes esses atributos não passaram de 68% de acurácia. Portanto, como um dos objetivos era produzir a previsão mais correta possível, iniciou-se novos testes.

Logo, os próximos testes foram realizados com um novo atributo. Dessa vez, foram utilizadas matrículas do aluno e seus resultados. Porém, um fator adicional desse atributo que foi levado em conta: o “peso” das diferentes disciplinas. Então construiu-se um “peso” sobre as disciplinas de acordo com a quantidade de reprovações que ela continha. Quanto maior o número de reprovações maior o peso. Os resultados representaram um aumento de 14% de acurácia em relação ao testes iniciais.

Ainda não satisfeitos com esse resultado dos testes, foi decidido retirar o peso das reprovações dos alunos, e utilizar um atributo mais simples: a quantidade de matrículas feitas. Isso criaria um certo contraste com a quantidade de retenções, e ajudaria estimadores como o KNN, que baseia sua predição a partir da proximidade dos dados. Os resultados desse teste confirmaram a teoria pois a acurácia foi de 95%.

Agora que a equipe se encontrava satisfeita com a acurácia de sua retenção um novo fator foi adicionado ao problema. A utilização dos dados de matrículas significava que os alunos no início do curso teriam predições incompatíveis com alguém que possui mais matrículas para ser utilizado na previsão. Então decidiu-se utilizar um atributo que representava o desempenho do aluno durante o curso, a saber, a percentagem de reprovações em suas matrículas. Tal atributo também é a união de dois atributos utilizados anteriormente que são a quantidade de retenções e a quantidade de matrículas. Esses testes foram feitos com os quatro modelos de estimadores apresentados no Capítulo 2.

Os resultados foram de 87% de acurácia. Porém, como os atributos representam um balanceamento entre as previsões que seriam feitas de alunos que estão no início do curso, e alunos que já tiveram mais oportunidades de ter seus dados coletados, foi decidido utilizar esses atributos como finais.

Um dos problemas que surgiram com essa predição foi a grande quantidade de estimativas falso-positivas, que são dados que deveriam ser estimados negativamente mas são estimados positivamente. No caso deste trabalho, significa que alunos que o sistema deveria apontar como possíveis evasores são apontados como alunos em situação aceitável. Então, com o intuito de diminuir a taxa de falso positivos aprimorou-se mais ainda os parâmetros dos estimadores. Todavia, os resultados não obtiveram melhora significativa. A acurácia total do estimador aumentou, chegando a 89%, porém, dentro dos 11% restantes o número de falsos-positivos continua alto, representando até 75% deste percentual de erro.

De todas as formas, concluiu-se que a melhor decisão que podia ser tomada no momento do estudo era a de se utilizar de uma generalização que pudesse abranger grades curriculares diferentes, e que tenha precisão eficiente desde o começo da trajetória acadêmica do aluno.

4. APRESENTAÇÃO DO SISTEMA

Neste capítulo é descrita a arquitetura que foi utilizada para desenvolver o sistema, e também as funcionalidades presentes em cada tela.

4.1. ARQUITETURA DO SISTEMA

O *framework* de desenvolvimento web que optou-se por utilizar neste trabalho, utiliza uma arquitetura de componentes derivada do clássico padrão MVC (*Model*, *View*, *Controller*). Em Django, utiliza-se o padrão MVT (*Model*, *View*, *Template*).

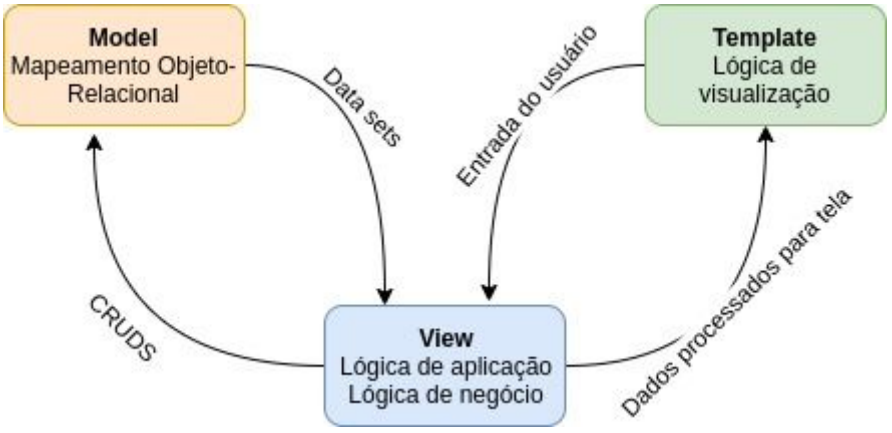
A camada *Model* do Django oferece uma API de alto nível para fazer a interação com o banco de dados, utilizando o conceito de ORM (Object relational Mapping). Este tipo de abstração do banco de dados possui a vantagem de permitir tratar os dados como se estivesse utilizando um sistema orientado a objetos puro, a camada de ORM isola os acessos e as consultas realizados ao banco de dados em uma camada de objetos consultados por API. Este modelo torna as aplicações mais confiáveis e mais seguras (BERNARDI, 2006) por realizar tratamento preventivo as consultas, não permitindo erros comuns que poderiam abrir brechas para SQL injection, por exemplo. Além de dar a possibilidade ao programador de trocar de banco de dados, através de um arquivo de configuração, sem precisar reescrever a lógica da aplicação.

A camada *View* atua como a *Controller* do MVC e é responsável por receber as requisições vindas do usuário, processar a entrada, formatar uma resposta e então solicitar a renderização de um template. Esta camada reside do lado do servidor e é nela que está toda a lógica da aplicação e do negócio.

A camada *Template* é responsável por manipular templates de arquivos estáticos, injetar e expandir o conteúdo de variáveis e assim criar arquivos HTML - ou XML, CSV, etc - dinâmicos. Para isto, o Django dispõe de uma DTL, ou Django Template Language, chamada de Jinja2. A camada de *template* permite o uso de variáveis, que são expandidas para valores no momento em que o *template* é renderizado, tags, que controlam a lógica do *template* e filtros, que adicionam

funcionalidades ao *templates*, como por exemplo a transformação de uma *string* em caixa alta no momento da renderização. A camada de *template* facilita a vida do programador eliminando ao máximo a duplicação de código e deixando os arquivos estáticos enxutos, permitindo uma melhor organização dos mesmos (RAMOS, 2018). A Figura 4 ilustra o funcionamento básico da arquitetura MVT.

FIGURA 4 - ARQUITETURA MVT



Fonte: Os autores (2019).

Em complemento à arquitetura, a lógica e os processos de mineração de dados representam grande parte do sistema, estes serão descritos nos próximos tópicos.

4.1.1. Banco de dados

Existem dois bancos de dados no sistema, o primeiro, chamado de *default* é responsável por guardar os dados dos alunos que ainda não evadiram do curso, além dos dados de autenticação no sistema e a predição realizada pelos scripts estimadores. O segundo, chamado *data warehouse*, é responsável por guardar os cálculos referentes aos alunos que evadiram o curso nos últimos 15 anos.

Ambos os bancos possuem scripts cronológicos que realizam uma limpeza de dados, no primeiro banco os alunos que evadiram o curso são retirados da base, e, no segundo banco os dados que possuem mais de 15 anos são excluídos.

4.1.2. Fluxo de importação de dados

A primeira etapa desse fluxo de importação de dados é receber um arquivo no formato csv, e garantir sua validade. Ao validar o csv o sistema verifica se o arquivo recebido possui o formato correto, e caso isso se confirme, o sistema confirma se os dados encontrados no cabeçalho do arquivo, onde estão o nome das colunas do csv, estão nomeados corretamente com o modelo mostrado da tela de Importação de CSV, representado na figura 15, na seção 4.2.6.

Então, o sistema inicia a importação desses dados para os bancos de dados. O primeiro banco, como possui os dados brutos do arquivo, é o primeiro a ser atualizado.

Com a finalização dessa primeira importação, o *script* inicia a limpeza dos dados do *data warehouse*. Caso existam dados com 15 anos ou mais de idade neste banco de dados eles são deletados em sua devida ordem. Isso evita *overtraining* da predição e também agiliza as consultas realizadas no BD.

Após a limpeza, o sistema inicia a inserção dos dados novos no segundo banco. A inserção neste banco é complicada pois deve-se atualizar também as fatos que representam mais de um fato, então primeiro se insere fatos que são únicas e depois se faz o *update* das outras.

A próxima etapa no fluxo é a limpeza do primeiro banco de dados, pois com a garantia que os dados estão salvos no *data warehouse* não existe razão para que os alunos que já evadiram continuem neste banco.

Então, é iniciada a etapa de predição dos dados. A predição acontece automaticamente com o processo de inserção de dados, logo após o *update* do *data warehouse* e a limpeza do primeiro banco. O processo segue os seguintes passos:

- Seleciona-se os dados dos alunos que ainda estão matriculados;
- Prepara-se os dados selecionados, ou seja, os dados são normalizados. A normalização desses dados deve ser feita pela mesma instância que os dados de treinamento são normalizados, portanto o *scaler* é salvo em um arquivo dentro da pasta no servidor. A checagem desse arquivo acontece todas as vezes antes de se instanciar um normalizador.

- Instância-se o *script* de predição, caso a instância não tenha nenhum treinamento anterior seleciona-se os dados de treinamento do *data warehouse* e faz-se o treinamento dessa nova instância. Assim como o normalizador o *script* estimador é salvo em um arquivo dentro da pasta no servidor. A checagem, assim como o normalizador também é feita antes de se iniciar um nova instância do *script* estimador.
- Inicia-se a predição dos dados selecionados e os resultados são salvos no primeiro banco.

Os atributos utilizados para a predição de evasão de um aluno são o seu IRA e a porcentagem de reprovação de suas matrículas. Com esses atributos a previsão de um aluno no começo ou no final do curso podem dar o mesmo resultado, porém a precisão da predição de evasão de um aluno de desempenho precário ao final do curso sempre será maior, pois, por encontrar-se mais próximo do término da graduação, uma mudança brusca nos atributos é menos provável devido a natureza desses atributos, que sempre levam em conta dados anteriores.

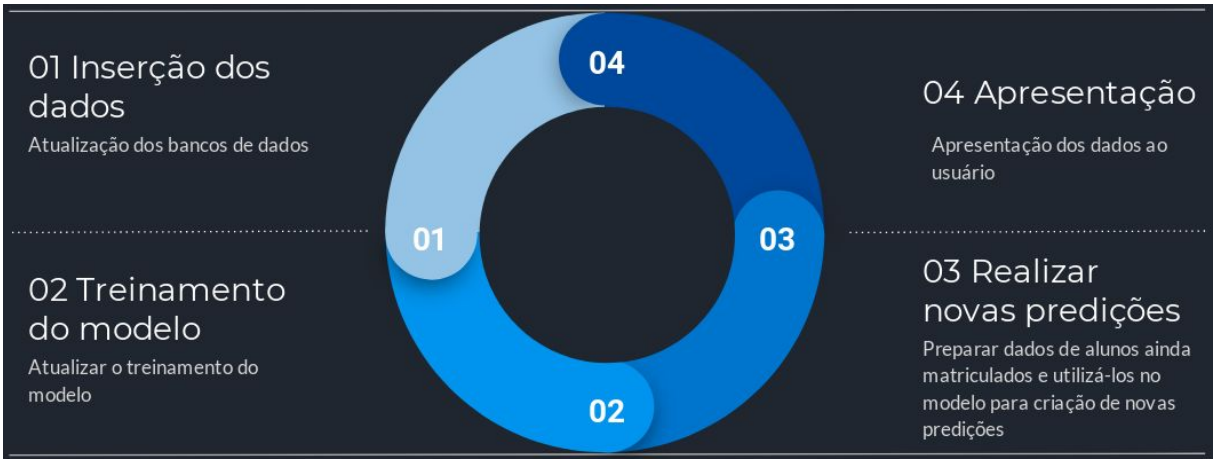
O normalizador utilizado segue a seguinte fórmula matemática:

$$z = (x - u) / s$$

Para encontrar *z*, que representa o resultado da normalização de *x*, diminui-se dele *u*, que é a média adquirida através do treinamento e então o resultado é dividido pelo *s*, que é o desvio padrão do atributo.

Resumidamente o fluxo de importação de dados segue o processo demonstrado na Figura 5.

FIGURA 5 - FLUXO DE IMPORTAÇÃO DE DADOS



Fonte: Os autores (2019).

O script que atingiu maior acurácia na predição é MLP; portanto, foi escolhido como script estimador padrão utilizado no sistema. No entanto, cerca de 10% das predições realizadas pelo algoritmo podem conter falso-positivos, ou seja, discentes que deveriam ser previstos como evasores podem vir a ser previstos como alunos em situação normal e um futuro formando. Cabe ressaltar que os dados devem ser sempre analisados com cautela e, mesmo que um aluno não seja classificado como possível evasor, o usuário do sistema deve sempre levar em conta um desempenho precário como alerta.

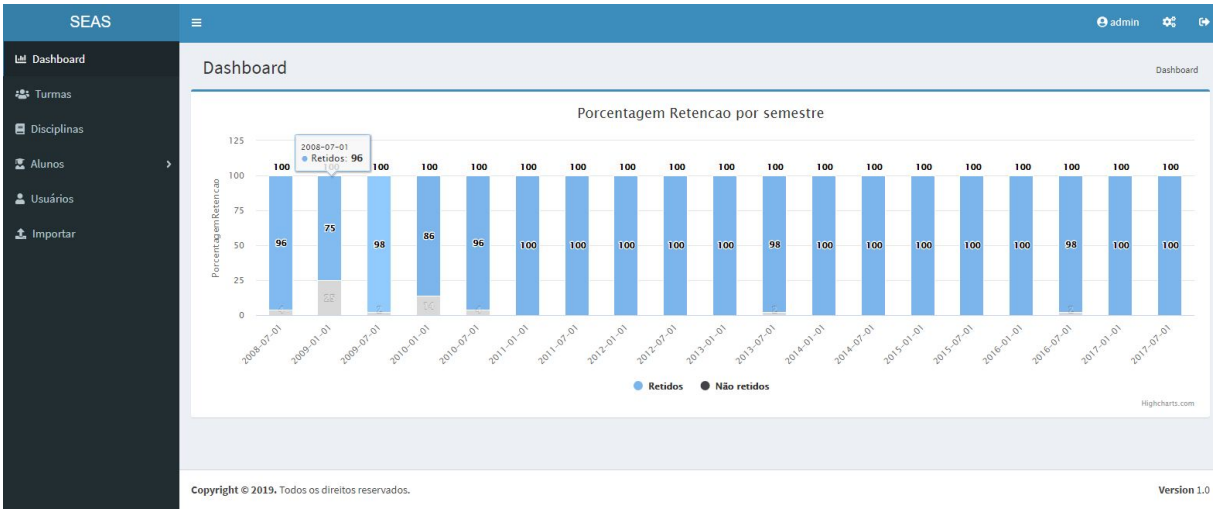
4.2. APRESENTAÇÃO DAS TELAS

Dentro do sistema é possível encontrar seis interfaces principais que representam todas as suas respectivas funcionalidades, estas são descritas e apresentadas abaixo.[1]

4.2.1. Dashboard

Nesta tela são apresentados gráficos com os dados calculados pelo sistema. Na figura seguinte é apresentado a porcentagem de retenção dos alunos formados por semestre.

FIGURA 6 - TELA DE DASHBOARD



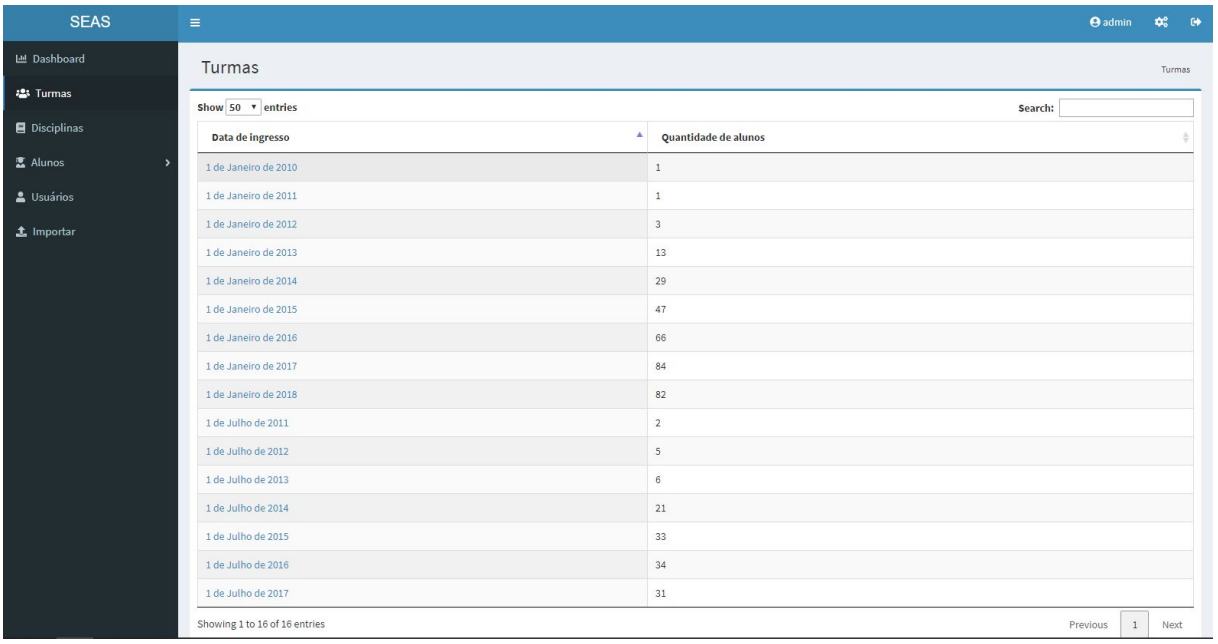
Fonte: Os autores (2019).

[1] Todos os prints apresentados neste capítulo foram feitos em servidor de teste e com os dados anonimizados

4.2.2. Turmas

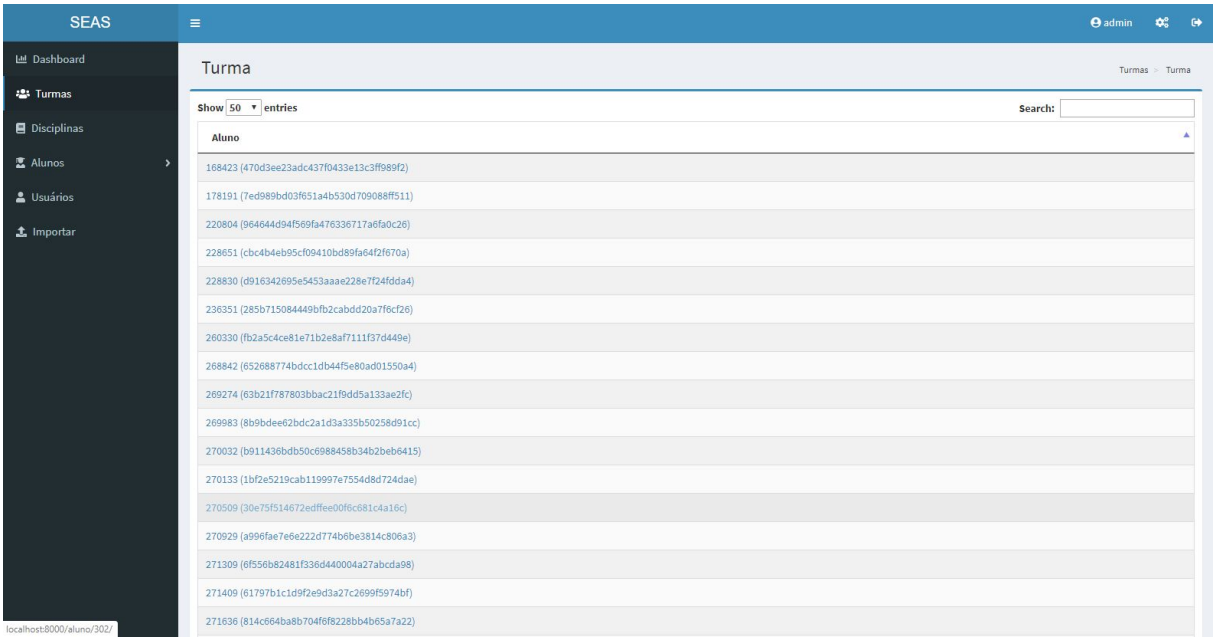
Nesta interface é apresentado ao usuário as turmas que ainda possuem matrículas. As turmas são separadas por ano de ingresso dos alunos. Ao selecionar uma delas, uma lista de alunos que ainda estão matriculados é apresentada ao usuário.

FIGURA 7 - TELA DE LISTAGEM DE TURMAS



Fonte: Os autores (2019).

FIGURA 8 - TELA DE DETALHE DE TURMAS

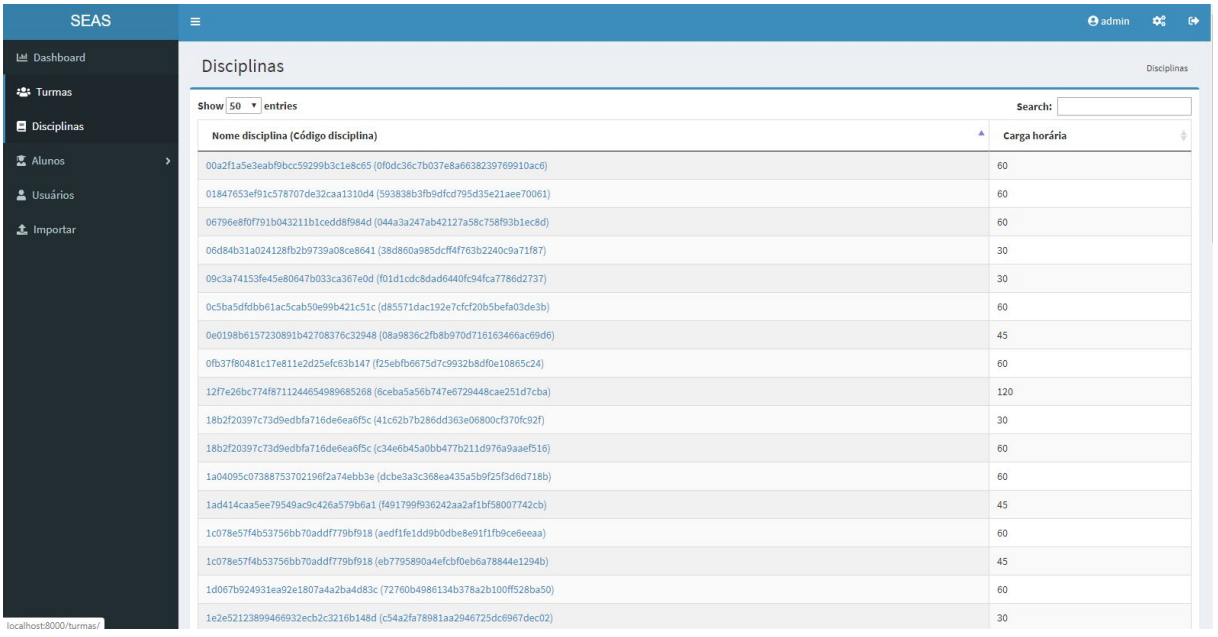


Fonte: Os autores (2019).

4.2.3. Disciplinas

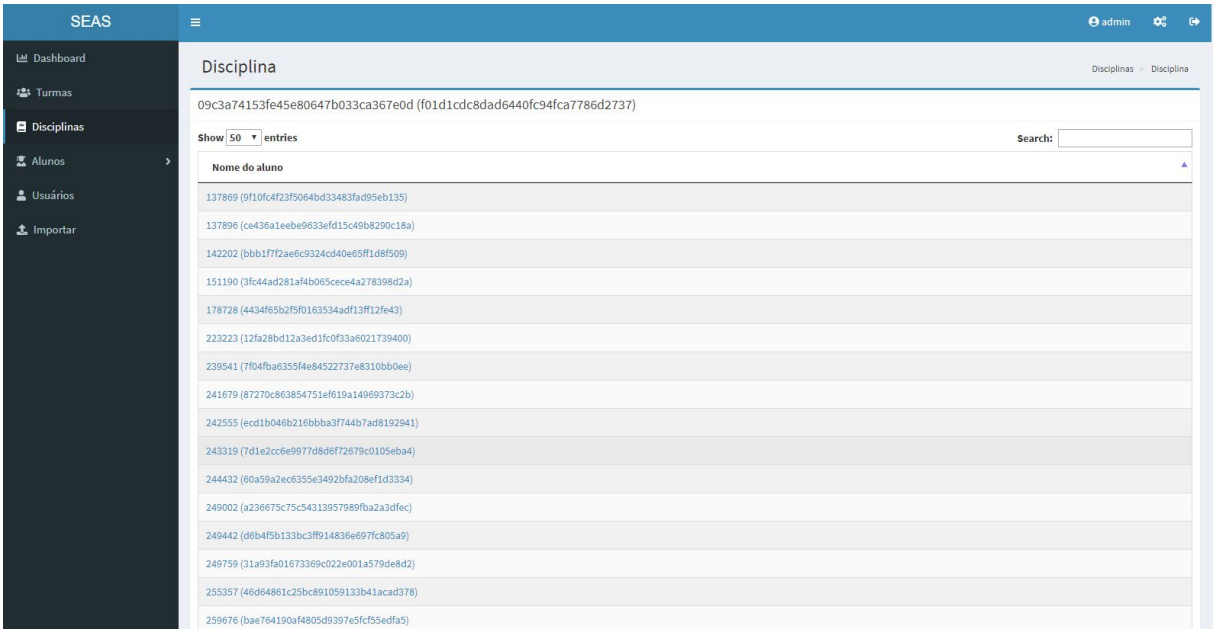
Esta interface apresenta ao usuário uma lista de todas as disciplinas cadastradas no sistema. Ao selecionar uma delas o sistema leva a uma página de detalhes, que mostra a porcentagem de reprovação na disciplina e uma lista de todos os alunos matriculados ordenada alfabeticamente.

FIGURA 9 - TELA DE LISTAGEM DE DISCIPLINAS



Fonte: Os autores (2019).

FIGURA 10 - TELA DE DETALHE DE DISCIPLINAS

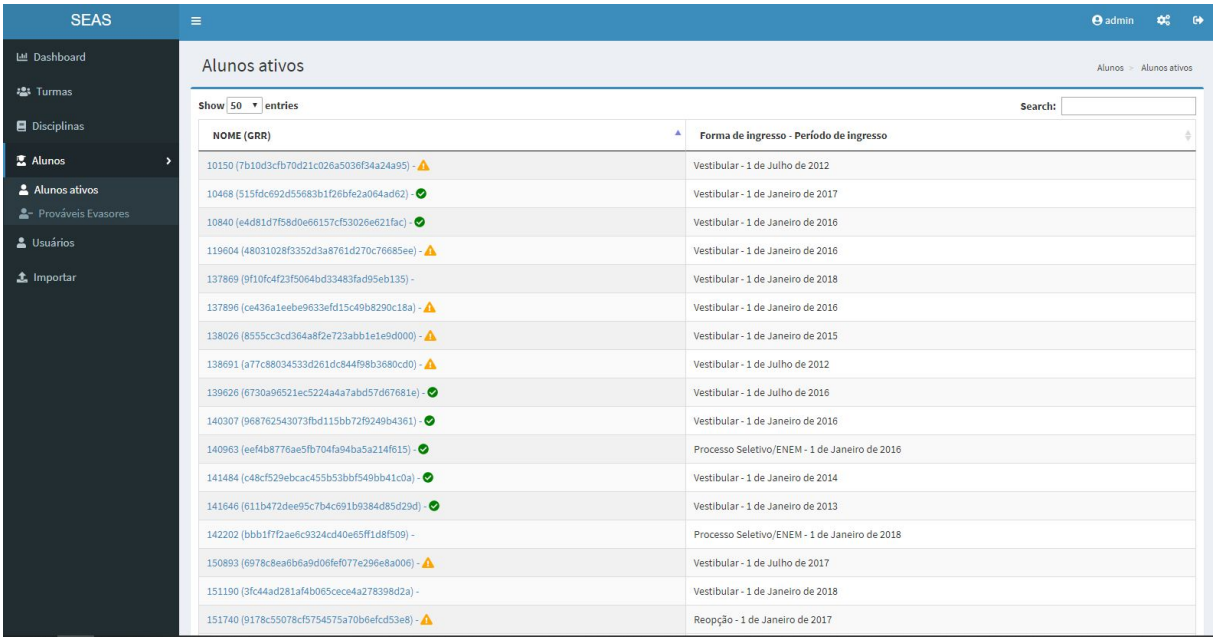


Fonte: Os autores (2019).

4.2.4. Alunos

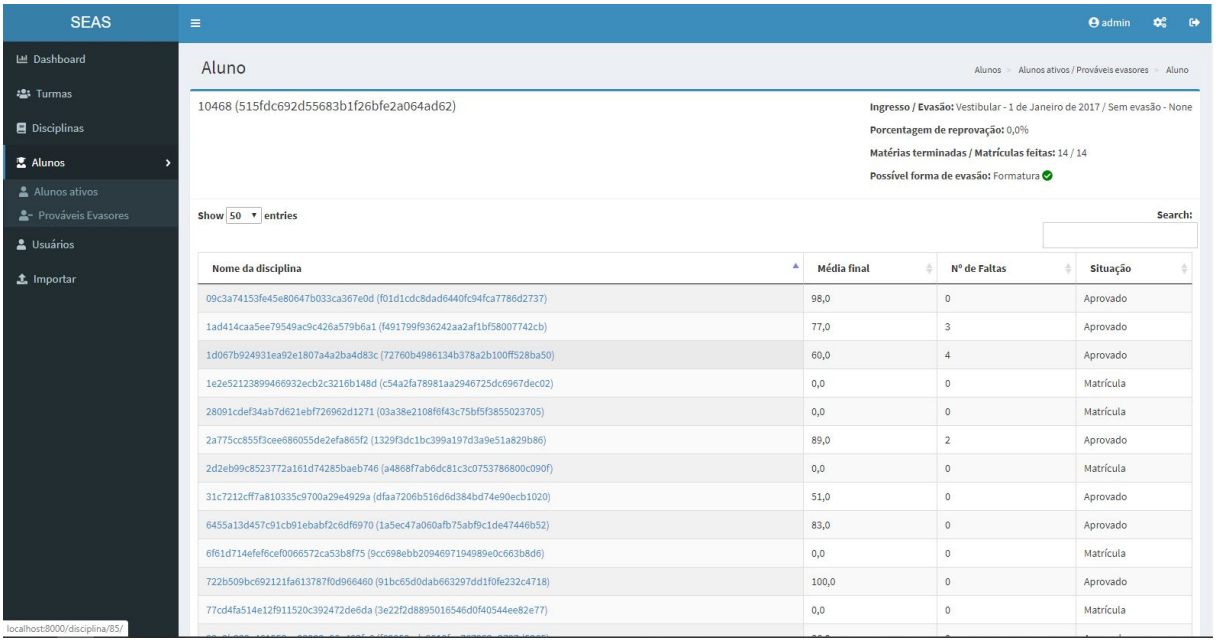
Apresenta uma lista com todos os alunos que estão matriculados e não evadiram. Ao lado dos campos “Nome” e “GRR”, é apresentado um ícone que representa o resultado da predição feita pelo sistema. Ao selecionar um dos alunos, uma lista com as disciplinas que esse aluno já se matriculou é apresentada e mostra média final da disciplina, o número de faltas, e situação (aprovado, reprovado, matriculado, etc.) de cada uma.

FIGURA 11 - TELA DE LISTAGEM DE ALUNOS



Fonte: Os autores (2019).

FIGURA 12 - TELA DE DETALHE DE ALUNO

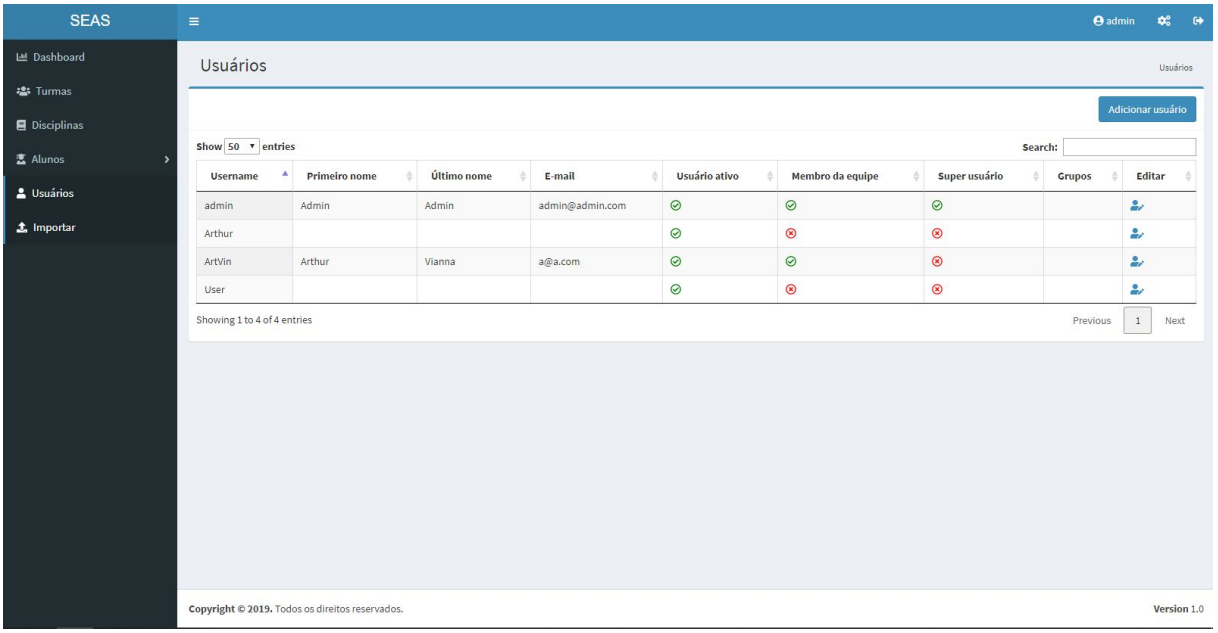


Fonte: Os autores (2019).

4.2.5. Usuários

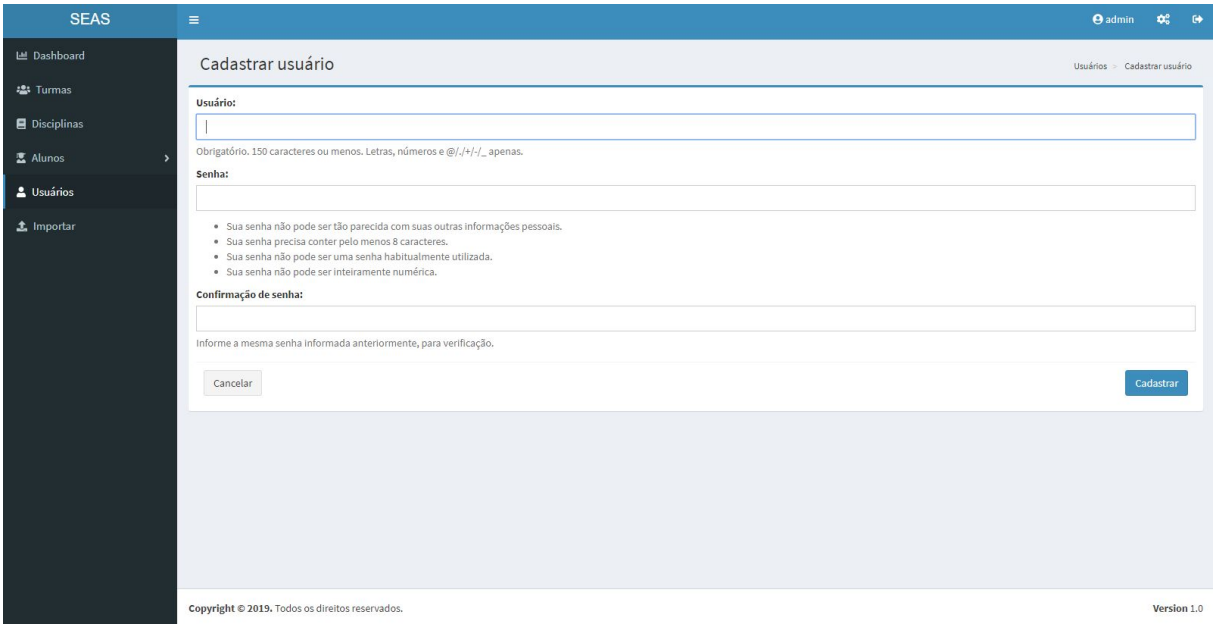
Nessa tela é apresentado uma lista com todos os usuários cadastrados no sistema. Também pode-se fazer cadastro de novos usuários e edição. Somente o administrador do sistema tem acesso à essa página.

FIGURA 13 - TELA DE LISTAGEM DE USUÁRIOS



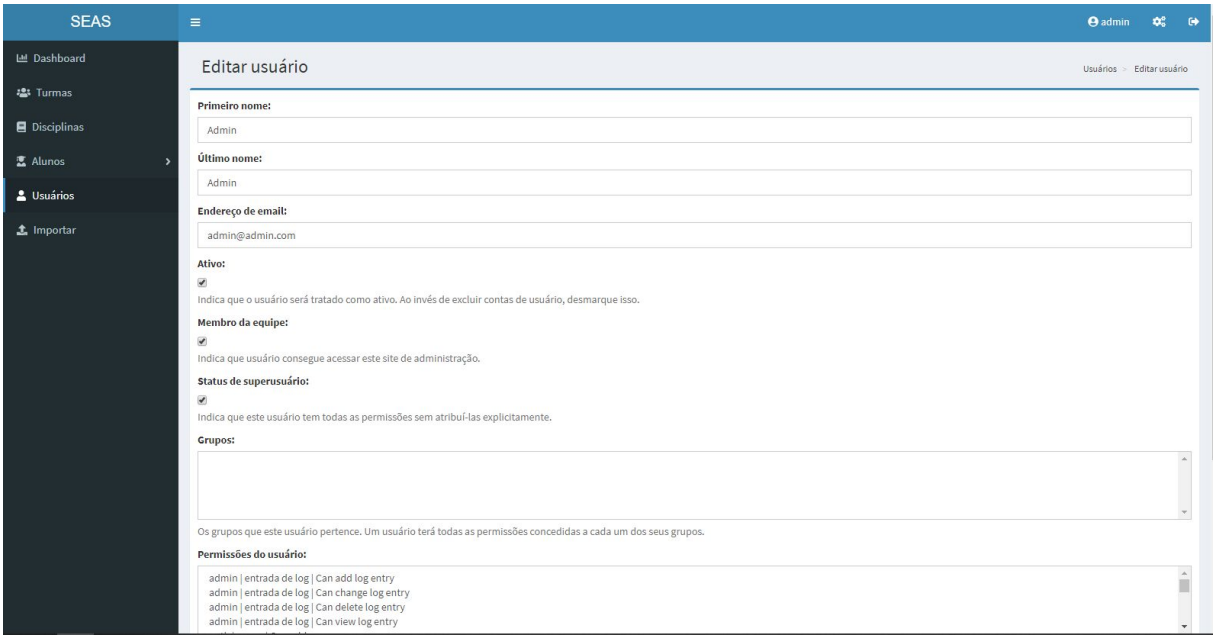
Fonte: Os usuários (2019).

FIGURA 14 - TELA DE CADASTRO DE USUÁRIO



Fonte: Os usuário (2019).

FIGURA 15 - TELA DE EDIÇÃO DE USUÁRIO

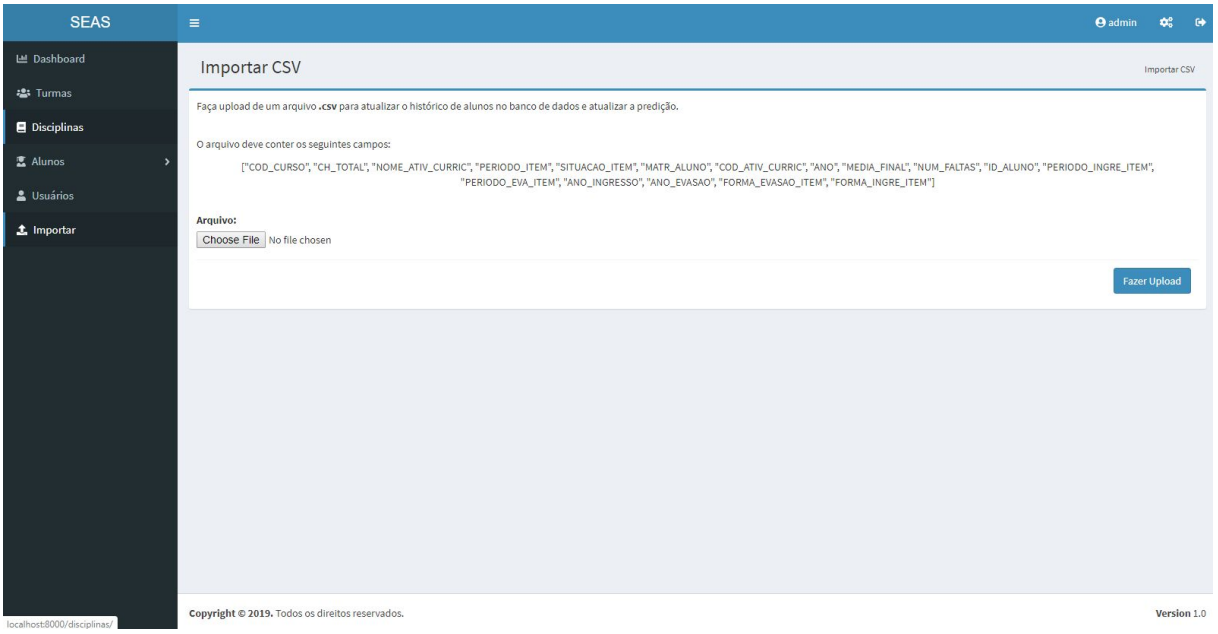


Fonte: Os autores (2019).

4.2.6. Upload CSV

Nesta interface, é possível iniciar o processo de inserção de dados do sistema. Somente o administrador do sistema tem acesso à essa página. Para que o processo seja iniciado o arquivo enviado pelo usuário deve atender certas características como ser um arquivo csv e conter os dados necessários para a importação, informação essa que está descrita na tela.

FIGURA 16 - TELA DE IMPORTAÇÃO DE CSV



Fonte: Os autores (2019).

5. CONSIDERAÇÕES FINAIS

O curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas ainda sofre com altas taxas de evasão e retenção, essa última, segundo os dados levantados por esta pesquisa, beira os 74% dos alunos formados. A formulação deste trabalho partiu como proposta do corpo docente, em forma de orientação, para que junto aos alunos fosse criada uma ferramenta que pudesse auxiliar na diminuição desses números.

O produto final criado a partir deste trabalho, batizado de SEAS - Student Evasion Analysis System, atingiu o objetivo proposto e conseguiu estimar, com acurácia de 89%, se um aluno do TADS tende ou não a evadir o curso dado o seu desempenho acadêmico até o momento. Alcançando a solução proposta para o problema descrito de forma concisa e eficiente, criou-se uma ferramenta em software que pode vir a ajudar o corpo docente a acompanhar o desempenho individual de seus alunos.

Ao longo do desenvolvimento deste projeto, fez-se necessário levar em consideração fatores que transcendem uma simples planilha de dados brutos, como as mudanças de grades e ementas que ocorreram desde a fundação do curso até o momento presente, incluindo a fase em que essa graduação se encontra hoje, em que duas grades coexistem no curso. Foi preciso estudar com cautela todos os atributos de cada aluno, observando somente os fatos, sem deixar ser levado por viés psicológico agregado em parte do material de referência lido, que apontava para diversos fatores sociológicos, culturais e pessoais que levam um indivíduo a decidir desistir de um curso superior.

Tinham-se somente os dados de desempenho acadêmico de cada aluno, a partir dos quais fez-se necessária a extração atributos que pudessem estimar com precisão, sem conhecer o passado do aluno, se esse deixaria ou não o curso ao longo de sua vida acadêmica. Foi preciso generalizar os atributos o suficiente para serem relevantes mesmo com todos esses fatores em conta.

O sistema funciona e consegue prever com acurácia relevante a probabilidade de um aluno deixar o curso, mas ainda não atingiu seu potencial máximo. Como os dados só representam o desempenho dos alunos no curso, e não

existe a possibilidade de usarmos dados da vida escolar do aluno antes do ingresso na universidade, é preciso sempre esperar o primeiro semestre ser concluído para que a primeira estimativa seja feita. Outra limitação do sistema é a forma de inserção de novos dados. Atualmente o sistema aceita apenas dados através de um arquivo que siga um modelo específico, nos mesmos padrões do arquivo disponibilizado pelo TADS no início da pesquisa. Uma solução inteligente para este problema, que abriria a possibilidade de extensão deste sistema, seria uma integração com o sistema de histórico escolar da Universidade Federal do Paraná.

Futuramente pode-se também criar uma funcionalidade que utiliza os dados minerados para realizar outras previsões, como por exemplo prever o resultado de um aluno em uma disciplina específica, levando em consideração dados de outros alunos com perfis similares.

Outra funcionalidade que poderia ser adicionada ao sistema, seria a integração com um sistema de visualização de dados, que possibilitaria ao usuário realizar as pesquisas que desejar nos dados, e então plotar gráficos dinâmicos relacionados a esta pesquisa, possibilitando que o usuário estenda o sistema ao limite de sua imaginação.

De todas as formas, o sistema se provou útil e eficiente, bastando que novas funcionalidades e adaptações sejam feitas para que este possa vir a tornar-se um ferramental de auxílio não somente ao corpo docente do curso de TADS, mas a Universidade Federal do Paraná inteira.

REFERÊNCIAS

BERNARDI, Diogo Alencar. **Técnicas de mapeamento objeto relacional**. SQL Magazine, [s. l.], ed. 40, 2006. Disponível em: https://www.devmedia.com.br/tecnicas-de-mapeamento-objeto-relacional-revista-sql-magazine-40/6980. Acesso em: 28 nov. 2019.

CAVESTRO, Julio de Melo; ROCHA, Fabio Lopes. **Prevalência de depressão entre estudantes universitários**. Jornal Brasileiro de Psiquiatria, [s. l.], v. 55, ed. 4, p. 265-267, 2006. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0047-20852006000400001. Acesso em: 23 nov. 2019.

CHACON, Scott; STRAUB, Ben. **Pro Git**. 2. ed. [S. l.: s. n.], 2014. Disponível em: https://git-scm.com/book/en/v2. Acesso em: 20 nov. 2019.

CHAUDHURI, Surajit; DAYAL, Umeshwar. **An Overview of Data Warehousing and OLAP Technology**. ACM SIGMOD Record, [s. l.], 1997. Disponível em: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/sigrecord.pdf . Acesso em: 26 nov. 2019.

CONSELHO DE ENSINO, PESQUISA E EXTENSÃO. CEPE. **RESOLUÇÃO Nº 37/97**. Resolução 37/97 CEPE, [S. l.], 1997. Disponível em: http://www.soc.ufpr.br/portal/wp-content/uploads/2016/07/cepe-37-97-alterada-pela-Res-70-18.pdf. Acesso em: 26 nov. 2019.

COSTA, Evandro; BAKER, Ryan S.J.d.; AMORIM, Lucas; MAGALHÃES, Jonathas; MARINHO, Tarsis. **Mineração de Dados Educacionais: Conceitos, Técnicas, Ferramentas e Aplicações**. Jornada de Atualização em Informática na Educação, [s. l.], 2013. Disponível em: https://br-ie.org/pub/index.php/pie/article/viewFile/2341. Acesso em: 26 nov. 2019.

COSTA, Francisco José da et al. **Diplomação, evasão e retenção: modelo longitudinal de análise para o ensino superior**. Editora UFPB, João Pessoa - PB, 2017. Disponível em: http://biblioteca.virtual.ufpb.br/files/diplomaaao_evasao_e_retenaao_modelo_longitudinal_de_analise_para_o_ensino_superior_1510325886.pdf. Acesso em: 22 nov. 2019.

EVASÃO. In: **AURÉLIO, Dicionário Online de Português**. Porto: 7Graus, 2019. Disponível em: https://www.dicio.com.br/evasao/. Acesso em: 23 nov. 2019.

FERNEDA, Edberto. **Redes neurais e sua aplicação em sistemas de recuperação de informação**. Ci. Inf. [online]. 2006, vol.35, n.1, pp.25-30. ISSN 0100-1965. Disponível em: http://dx.doi.org/10.1590/S0100-19652006000100003 Acesso em: 23 nov. 2019.

FILHO, Raimundo Barbosa Silva; ARAÚJO, Ronaldo Marcos de Lima. **Evasão e abandono escolar na educação básica no Brasil: fatores, causas e possíveis consequências**. Educação por Escrito, Porto Alegre, v. 8, ed. 1, p. 35-48, 2017. Disponível em: http://revistaseletronicas.pucrs.br/ojs/index.php/porescrito/article/view/24527. Acesso em: 23 nov. 2019.

FILHO, Roberto Leal Lobo e Silva. **A evasão no ensino superior brasileiro**. Cadernos de Pesquisa, [s. l.], 1 set. 2007. Disponível em: http://publicacoes.fcc.org.br/ojs/index.php/cp/article/view/346/350secaio.com/personal/TC/enegep2008_TN_STO_078_545_11614.pdf. Acesso em: 22 nov. 2019.

GIT. [S. l.], 2019. Disponível em: https://git-scm.com/. Acesso em: 20 nov. 2019.

INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA. **Sinopse Estatística da Educação Superior 2017**. Brasília: INEP, 2018. Disponivel em: http://portal.inep.gov.br/basica-censo-escolar-sinopse-sinopse. Acesso em: 25 nov. 2019.

INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA. **Sinopse Estatística da Educação Superior 2018**. Brasília: INEP, 2019. Disponivel em: http://portal.inep.gov.br/basica-censo-escolar-sinopse-sinopse. Acesso em: 25 nov. 2019.

KAUR, Nirmal; SINGH, Gurpinder. **A Review Paper On Data Mining And Big Data**. International Journal of Advanced Research in Computer Science, [s. l.], 2017. Disponível em: http://www.ijarcs.info/index.php/Ijarcs/article/view/3789/3270. Acesso em: 26 nov. 2019.

KIMBALL, Ralph; ROSS, Margy. **The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling**. [S. l.: s. n.], 2002. Disponível em: http://users.itk.ppke.hu/~szoer/DW/Kimball%20&%20Ross%20-%20The%20Data%20Warehouse%20Toolkit%202nd%20Ed%20%5BWiley%202002%5D.pdf. Acesso em: 26 nov. 2019.

LOELIGER, Jon; MCCULLOUGH, Matthew. **Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development**. [S. l.: s. n.], 2012. Disponível em: https://books.google.com.br/books?id=aM7-Oxo3qdQC&dq=git+versioning&lr=&hl=pt-BR&source=gbs_navlinks_s. Acesso em: 20 nov. 2019.

MACHADO, Nílson José. **A Universidade e a organização do conhecimento: a rede, o tácito, a dádiva**. Estudos Avançados, São Paulo, v. 15, n. 42, 2001. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-40142001000200018. Acesso em: 25 nov. 2019.

MARQUES, Waldemar; SILVA, Hércules Ferrari Domingues. **Evasão na Educação Superior no Brasil: Desafio à gestão acadêmica**. Quaestio, [S. l.], p. 198-208, 6 mar. 2017. Disponível em: http://periodicos.uniso.br/ojs/index.php/quaestio/article/view/2994. Acesso em: 10 out. 2019.

MARTINS, Bibiana Volkmer; OLIVEIRA, Sidinei Rocha de. **Qualificação Profissional, Mercado de Trabalho e Mobilidade Social: Cursos Superiores de Tecnologia**. Sociedade, Contabilidade e Gestão, Rio de Janeiro, v. 12, ed. 2, 2017. Disponível em: https://www.lume.ufrgs.br/bitstream/handle/10183/172610/001059630.pdf?sequence=1. Acesso em: 22 nov. 2019.

MEC/SEF. Secretaria de Educação Fundamental. **Parâmetros curriculares nacionais: introdução aos parâmetros curriculares nacionais**. Secretaria de Educação Fundamental, Brasília, 1997. Disponível em: http://portal.mec.gov.br/seb/arquivos/pdf/livro01.pdf. Acesso em: 25 nov. 2019.

MOZZILA.ORG, Comunidade. **Introdução ao Django**. [S. l.], 2019. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Django/Introdu%C3%A7%C3%A3o. Acesso em: 27 nov. 2019.

PEREIRA, Alexandre Severino et al. **Fatores relevantes no processo de permanência prolongada de discentes nos cursos de graduação presencial: um estudo na Universidade Federal do Espírito Santo**. Ensaio: Avaliação e Políticas Públicas em Educação, Rio de Janeiro, v. 13, n. 89, 2015. Disponível em: http://www.scielo.br/scielo.php?script=sci_abstract&pid=S0104-40362015000401015&lng=en&nrm=iso&tlng=pt. Acesso em: 25 nov. 2019.

RAMOS, Vinicius. **Desenvolvimento web com Python e Django: Template**. [S. l.], 2018. Disponível em: https://pythonacademy.com.br/blog/desenvolvimento-web-com-python-e-django-template. Acesso em: 28 nov. 2019.

RIFFEL, Sonia Marmol; MALACARNE, Vilmar. **Evasão Escolar No Ensino Médio: O Caso do Colégio Estadual Santo Agostinho No Município De Palotina - PR**. Dia a Dia Educação, [s. l.], 2010. Disponível em: http://www.diaadiaeducacao.pr.gov.br/portals/pde/arquivos/1996-8.pdf. Acesso em: 25 nov. 2019.

SAHU, Hemlata; SHRMA, Shalini; GONDHALAKAR, Seema. **A Brief Overview on Data Mining Survey.** International Journal of Computer Technology and Electronics Engineering (IJCTEE), [s. l.], v. 1, ed. 3, 2011. Disponível em: https://pdfs.semanticscholar.org/f44d/2c02e22ae27364e0bcfbfcb5bed74b0aa2e1.pdf. Acesso em: 26 nov. 2019.

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.Disponível em: https://scikit-learn.org/stable/index.html. Acesso em: 26 nov. 2019.

SIDHU, Amandeep S. et al. Knowledge Discovery in Biomedical Data Facilitated by Domain Ontologies. In: ZHU, Xingquan et al. **Knowledge Discovery and Data Mining: Challenges and Realities**. [S. l.: s. n.], 2007. cap. 10, p. 189-201. Disponível em: https://pdfs.semanticscholar.org/9cff/a70f39b654ebf77c655825b2204c4890fcb6.pdf. Acesso em: 26 nov. 2019.

SIMEONE, Osvaldo. **A Very Brief Introduction to Machine Learning With Applications to Communication Systems**. IEEE Transactions on Cognitive Communications and Networking, [s. l.], v. 4, n. 4, 21 nov. 2018. Disponível em: https://ieeexplore.ieee.org/abstract/document/8542764. Acesso em: 23 nov. 2019.

STACK OVERFLOW SURVEY. **Developer Survey Results: Most Popular Technologies**. [S. l.], 2019. Disponível em: https://insights.stackoverflow.com/survey/2019#most-popular-technologies. Acesso em: 27 nov. 2019.

TAGLIAFERRI, Lisa. An Introduction to Machine Learning. In: BOUCHERON, Brian; MORALES, Michelle; BIRBECK, Ellie; WAN, Alvin. **Python Machine Learning Projects**. [S. l.: s. n.], 2019. cap. 3. Disponível em: https://assets.digitalocean.com/books/python/machine-learning-projects-python.pdf. Acesso em: 26 nov. 2019.

WIDENIUS, Michael. **Five Questions With Michael Widenius – Founder And Original Developer of MySQL**. [S. l.], 2009. Disponível em: www.opensourcereleasefeed.com. Acesso em: 28 nov. 2019.

APÊNDICE A - LISTA DE REQUISITOS

REQUISITOS FUNCIONAIS

- RF01 - O sistema deve permitir a criação de novos usuários utilizadores do sistema, do usuário é guardado: username, nome, e-mail e senha.
- RF02 - O sistema deve permitir a edição de usuários utilizadores do sistema.
- RF03 - O sistema deve permitir a exclusão de usuários utilizadores do sistema.
- RF04 - O sistema deve permitir a inclusão de uma nova planilha de dados, a planilha necessita estar em formato .csv e se adequar aos campos: GRR_ALUNO, NOME_ALUNO, COD_DISCIPLINA, MEDIA, FALTAS, CH_TOTAL.
- RF05 - O sistema deve permitir a visualização de dados referente a disciplinas.
- RF06 - O sistema deve permitir a visualização de dados referente a turmas.
- RF07 - O sistema deve permitir a visualização de dados referente a alunos.
- RF08 - O sistema deve permitir login na plataforma solicitando login e senha.
- RF09 - O sistema deve classificar alunos como possíveis evasores/não evasores.

REQUISITOS NÃO FUNCIONAIS

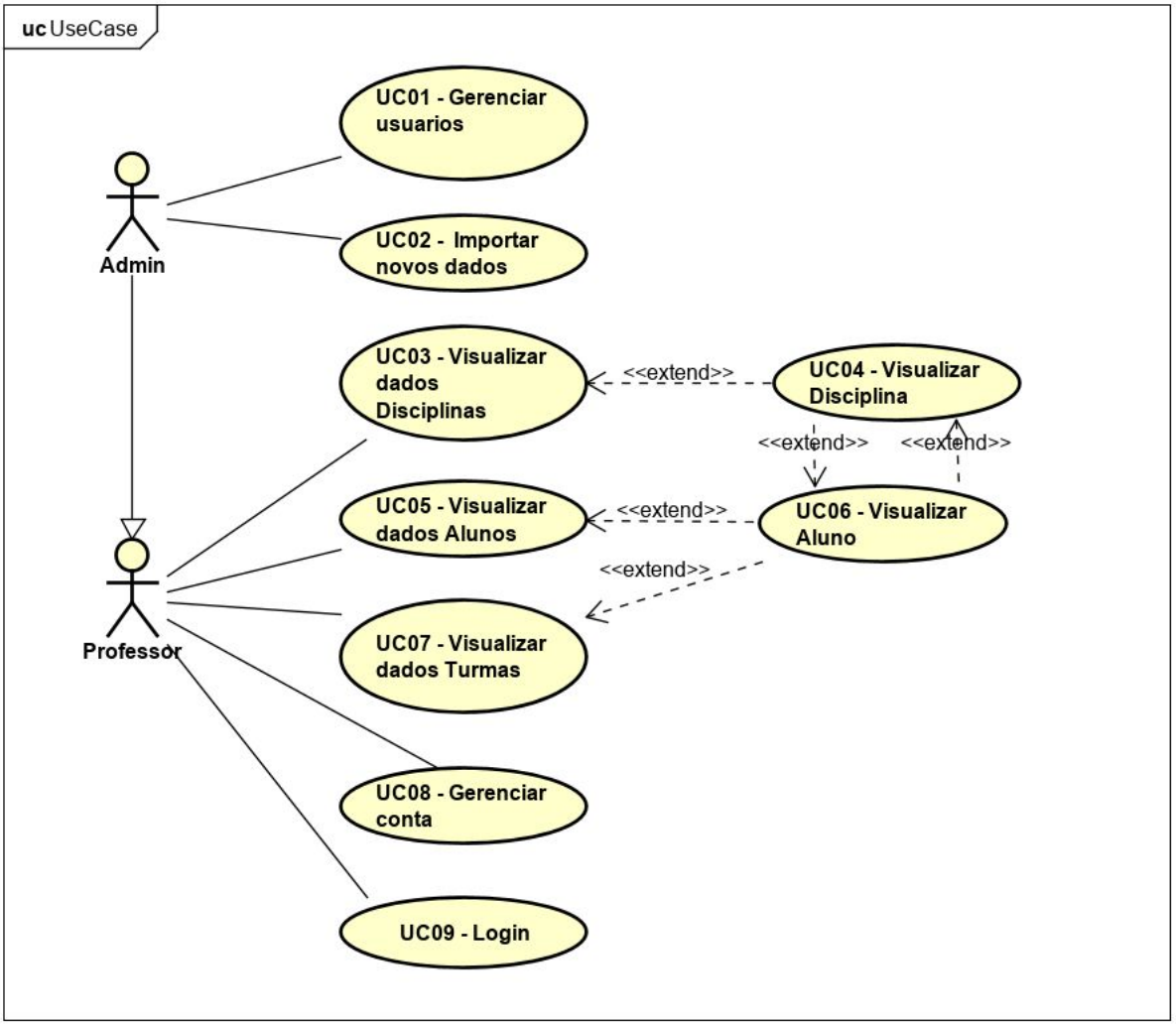
- RNF01 - O sistema deveria apresentar gráficos referentes aos estudos realizados.
- RNF02 - O sistema não deve permitir criação de usuários duplicados.

REQUISITOS ORGANIZACIONAIS

- RORG01 - Linguagem de Programação: O Sistema será feito em Django, framework muito utilizado no desenvolvimento ágil na plataforma WEB.
- RORG02 - Banco de dados: Será utilizado banco de dados relacional MySQL.
- RORG03 - O sistema deve rodar em um servidor de código aberto (GNU/Linux).

APÊNDICE B - DIAGRAMA E DESCRIÇÃO DE CASO DE USO

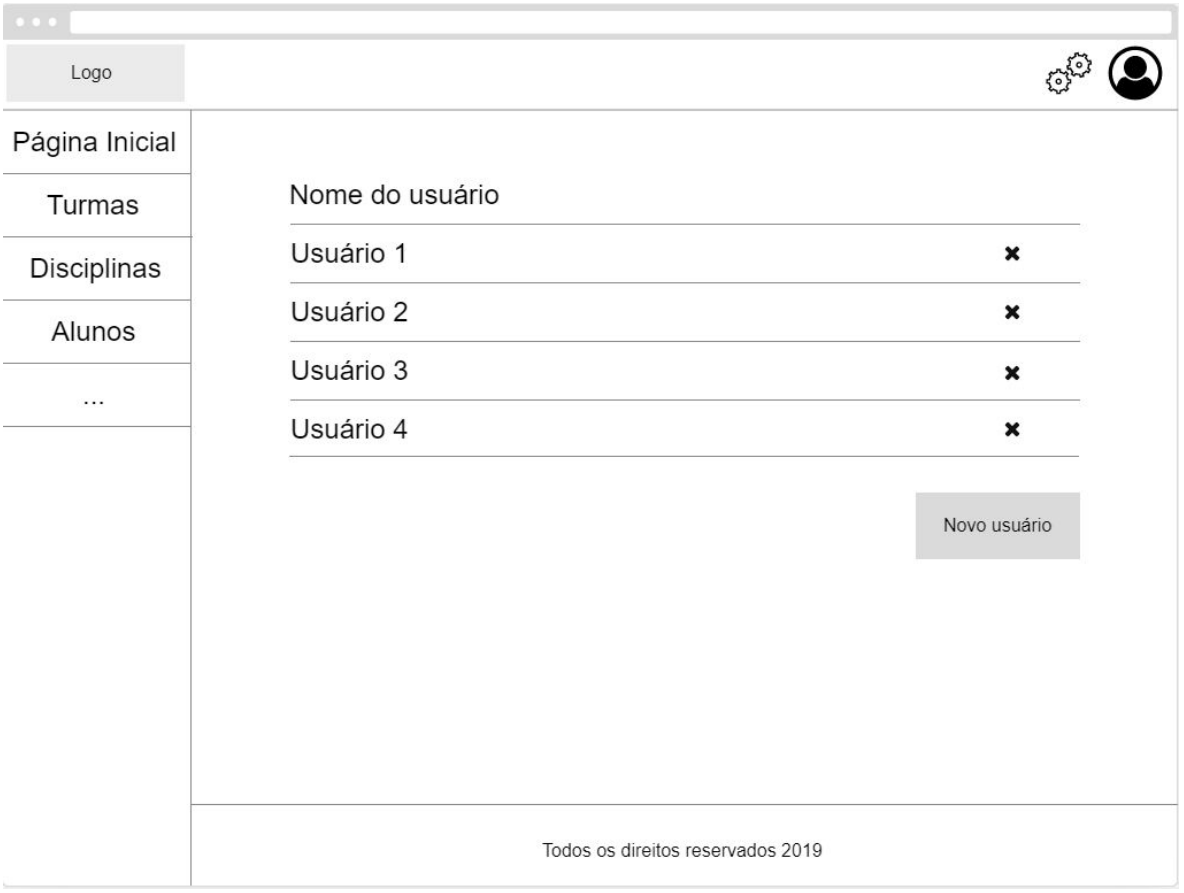
FIGURA 17 - DIAGRAMA DE CASO DE USO



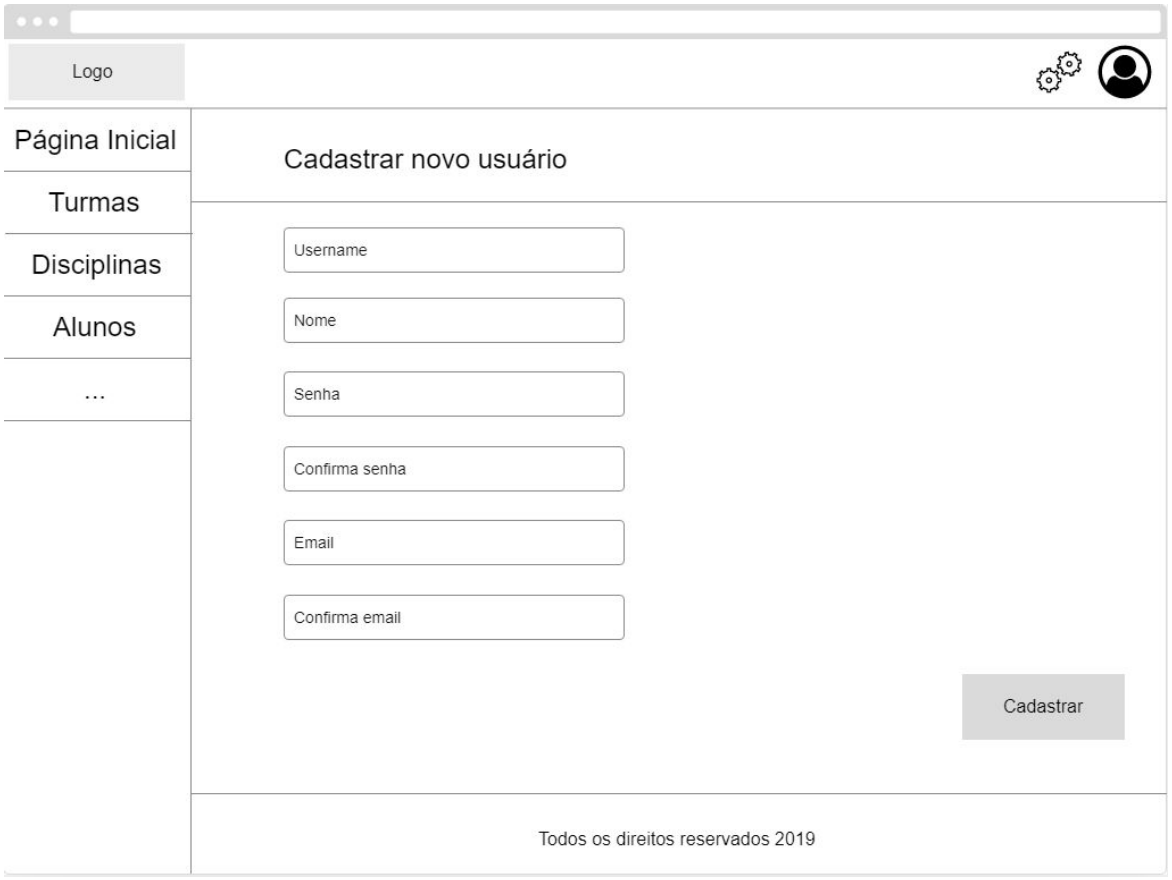
Fonte: Os autores (2019).

QUADRO 2 - DESCRIÇÃO UC01: GERENCIAR USUÁRIOS

Nome	UC01 - Gerenciar usuários
Ator principal	Administrador
Descrição	Usuário poderá excluir ou cadastrar novos usuários no sistema
Pré-condição	Estar logado com um usuário administrador
Pós-condição	Após o fim normal deste caso de uso: 1. O sistema deve salvar os dados do usuário cadastrado 2. O sistema deve remover o usuário escolhido caso o usuário administrador selecione o botão de remover.
Data views	



DV1



DV2

Fluxo Principal	1. O sistema apresenta uma lista contendo nome e email de todos os usuários cadastrados (DV1). 2. O usuário clica em “Cadastrar usuário” (A1). 3. O sistema apresenta uma tela com um formulário para preenchimento dos dados do usuário (DV2). 4. O usuário entra com os dados de cadastro e clica em “Cadastrar” (E1). 5. O usuário novo é cadastrado no sistema (E2). 6. O sistema exibe uma mensagem de sucesso. 7. Caso de uso é reiniciado.
Fluxo Alternativo	**A1**. Remover usuário: 1. O usuário clica em “Remover usuário” do usuário desejado. 2. O sistema exibe uma mensagem de confirmação de exclusão (DV3). 3. O usuário é removido do sistema (E3). 4. O sistema exibe uma mensagem de sucesso. 5. Caso de uso é reiniciado
Fluxo de exceção	**E1**. Campo preenchido incorretamente: 1. O sistema destaca os campos incorretos para a correção.

	2. Progressão no fluxo é bloqueada enquanto campo estiver incorreto. **E2**. Erro ao cadastrar um usuário no sistema: 1. O sistema exibe uma mensagem de erro. 2. O usuário é encaminhado para o formulário. **E3**. Erro ao remover usuário: 1. O sistema exibe uma mensagem de erro. 2. O usuário é encaminhado para a lista de usuários.

Fonte: Os autores (2019).

QUADRO 3 - DESCRIÇÃO UC02: IMPORTAR NOVOS DADOS

Nome	UC02 - Importar novos dados
Ator principal	Administrador
Descrição	Usuário inicia import de novos dados do corpo discente no banco de dados
Pré-condição	Estar logado com um usuário administrador
Pós-condição	Sistema recebe arquivo com dados do corpo discente e inicia leitura e escrita no banco de dados

Data views

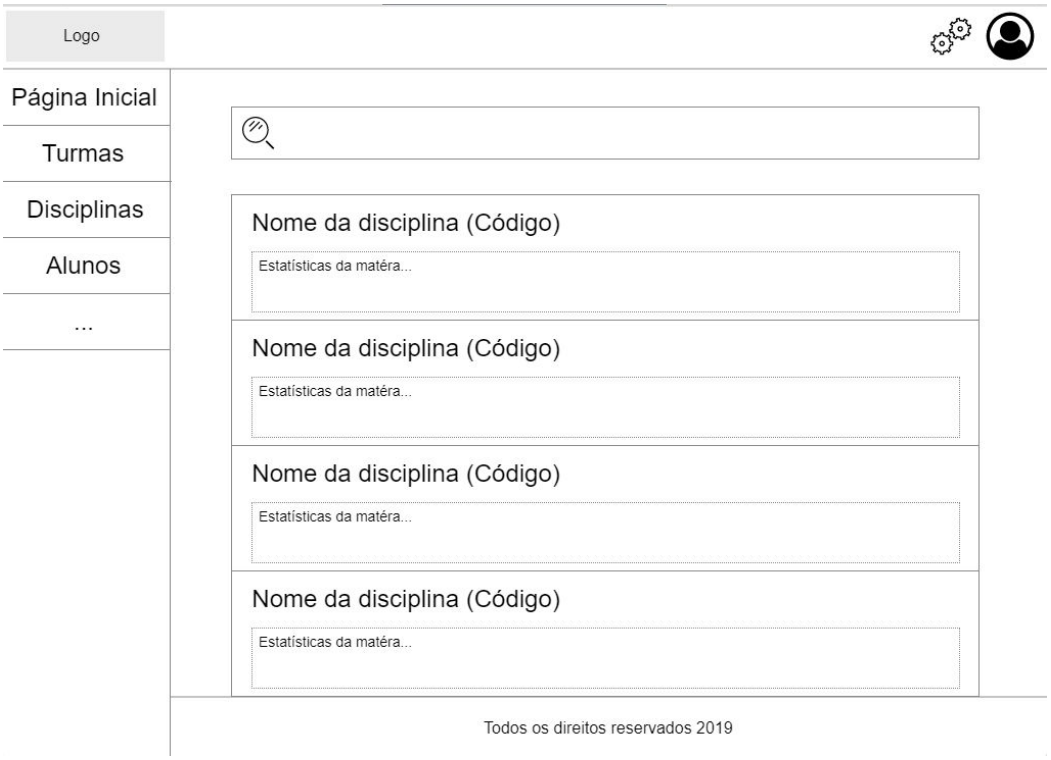
DV1

Fluxo Principal	1. Sistema exibe tela com as especificações e modelo do arquivo 2. Usuário seleciona arquivo com dados do corpo discente que seguem o modelo requisitado (DV1,E1). 3. Sistema exibe mensagem de sucesso.
Fluxo Alternativo	N/A
Fluxo de exceção	**E1**. Arquivo não segue o modelo requisitado

	1. Sistema exibe mensagem de erro. 2. Progressão no fluxo é bloqueada enquanto o arquivo enviado não seguir o modelo

Fonte: Os autores (2019).

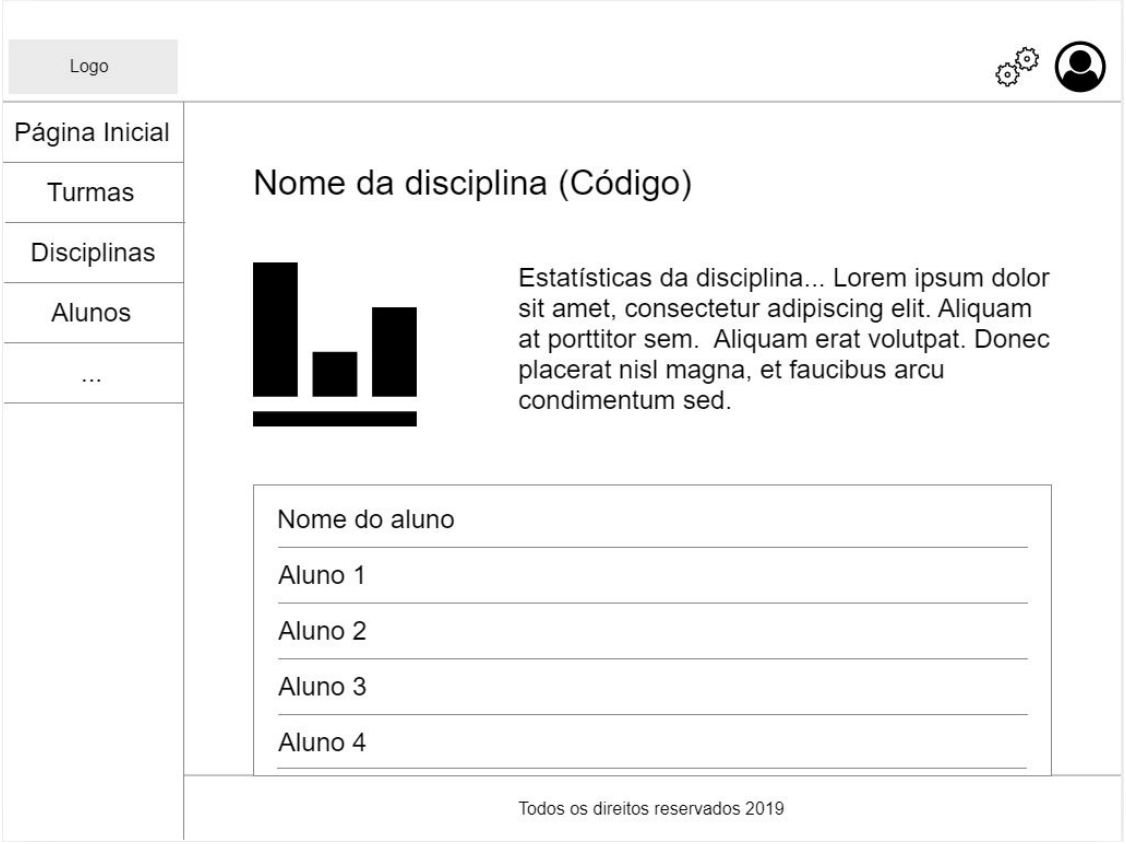
QUADRO 4 - DESCRIÇÃO UC03: VISUALIZAR DADOS DISCIPLINAS

Nome	UC03 - Visualizar dados Disciplinas
Ator principal	Docente
Descrição	Usuário pode navegar por uma lista de disciplinas cadastradas e visualizar dados relacionados
Pré-condição	Usuário deve estar logado no sistema
Pós-condição	
Data views	
 DV1	
Fluxo Principal	1. Sistema apresenta uma lista com todas as disciplinas cadastradas e um campo de pesquisa (DV1, A1, E1). 2. Usuário seleciona uma disciplina da lista. 3. Inicia o caso de uso [UC04 - Visualizar Disciplina].
Fluxo Alternativo	**A1**. Pesquisa : 1. Usuário digita algo no campo de pesquisa. 2. Lista é filtrada com o input do Usuário.(E1)
Fluxo de exceção	**E1**. Nenhuma disciplina encontrada : 1. Sistema exibe mensagem informando que nenhuma disciplina foi encontrada.

Fonte: Os autores (2019).

QUADRO 5 - DESCRIÇÃO UC04: VISUALIZAR DISCIPLINA

Nome	UC04 - Visualizar Disciplina
Ator principal	Docente
Descrição	Usuário visualiza dados da disciplina selecionada
Pré-condição	Usuário deve estar logado no sistema e uma disciplina deve ser referenciada
Pós-condição	
Data views	



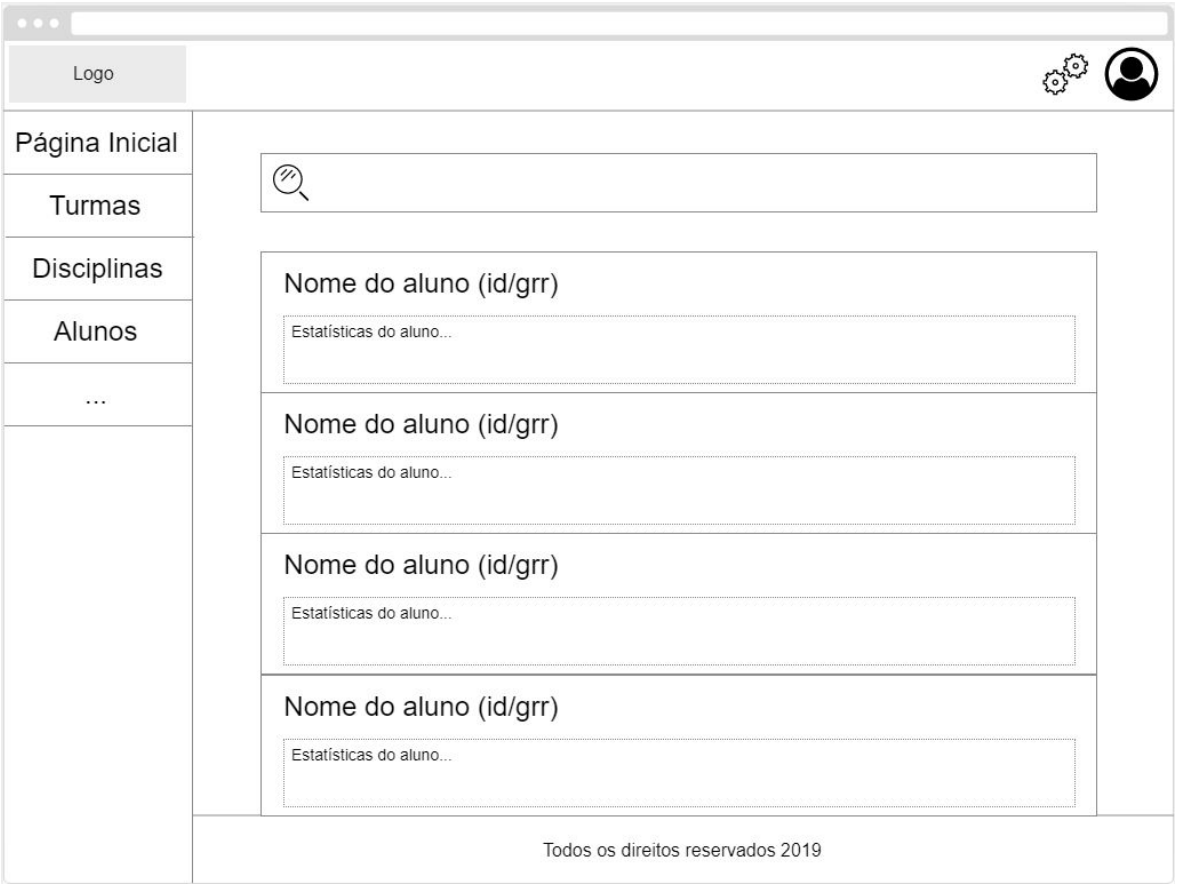
DV1

Fluxo Principal	1. Sistema apresenta detalhes da disciplina referenciada e uma lista de alunos matriculados nela (DV1, E1). 2. Usuário seleciona um aluno da lista. 3. Inicia o caso de uso [UC06 - Visualizar Aluno].
Fluxo Alternativo	N/A
Fluxo de exceção	**E1**. Nenhuma disciplina referenciada : 1. Sistema exibe mensagem informando que nenhuma disciplina foi referenciada.

Fonte: Os autores (2019).

QUADRO 6 - DESCRIÇÃO UC05: VISUALIZAR DADOS ALUNOS

Nome	UC05 - Visualizar dados Alunos
Ator principal	Docente
Descrição	Usuário pode navegar por uma lista de alunos cadastrados e visualizar dados relacionados
Pré-condição	Usuário deve estar logado no sistema
Pós-condição	
Data views	



DV1

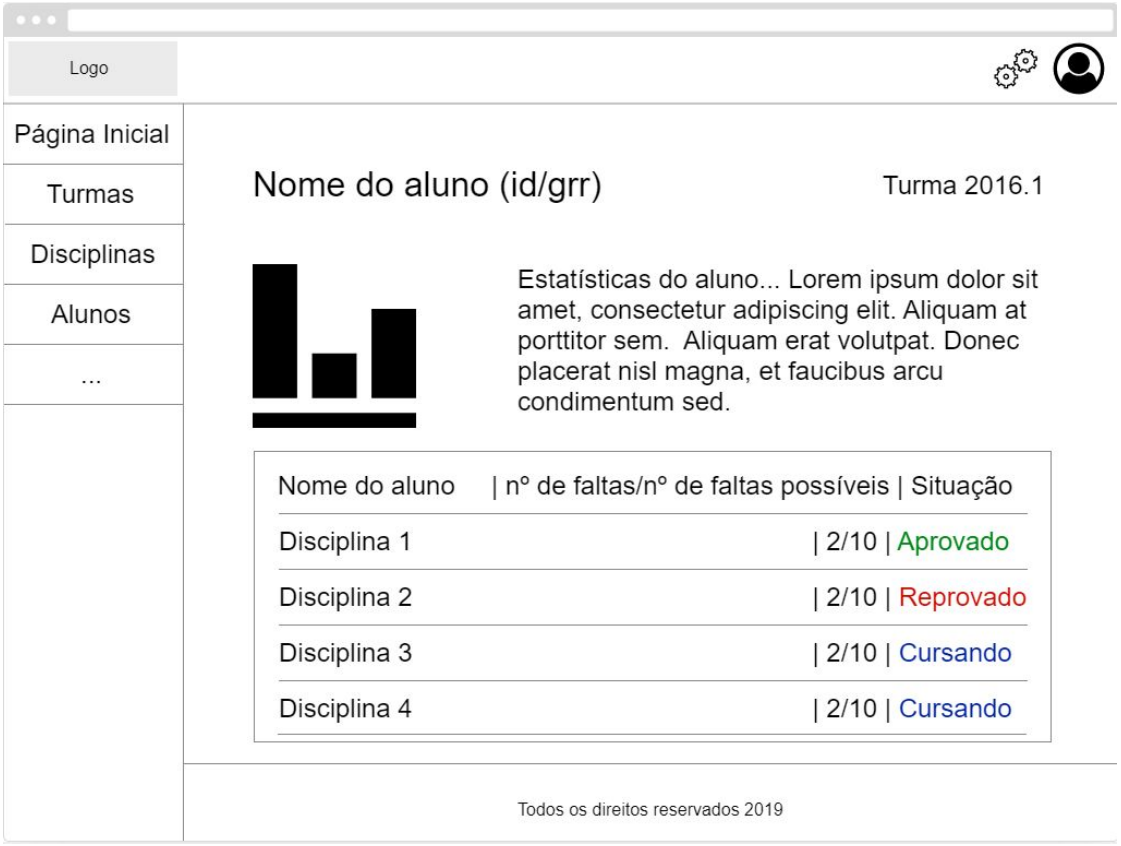
Fluxo Principal	1. Sistema apresenta uma lista com todos os alunos cadastrados e um campo de pesquisa (DV1, A1, E1). 2. Usuário seleciona um aluno da lista. 3. Inicia o caso de uso [UC06 - Visualizar Aluno].
Fluxo Alternativo	**A1**. Pesquisa : 1. Usuário digita algo no campo de pesquisa. 2. Lista é filtrada com o input do Usuário.(E1)
Fluxo de exceção	**E1**. Nenhum aluno encontrado :

	1. Sistema exibe mensagem informando que nenhum aluno foi encontrado.

Fonte: Os autores (2019).

QUADRO 7 - DESCRIÇÃO UC06: VISUALIZAR ALUNO

Nome	UC06 - Visualizar Aluno
Ator principal	Docente
Descrição	Usuário visualiza dados do aluno selecionado.
Pre-condição	Usuário deve estar logado no sistema e um aluno deve ser referenciado
Pós-condição	
Data views	



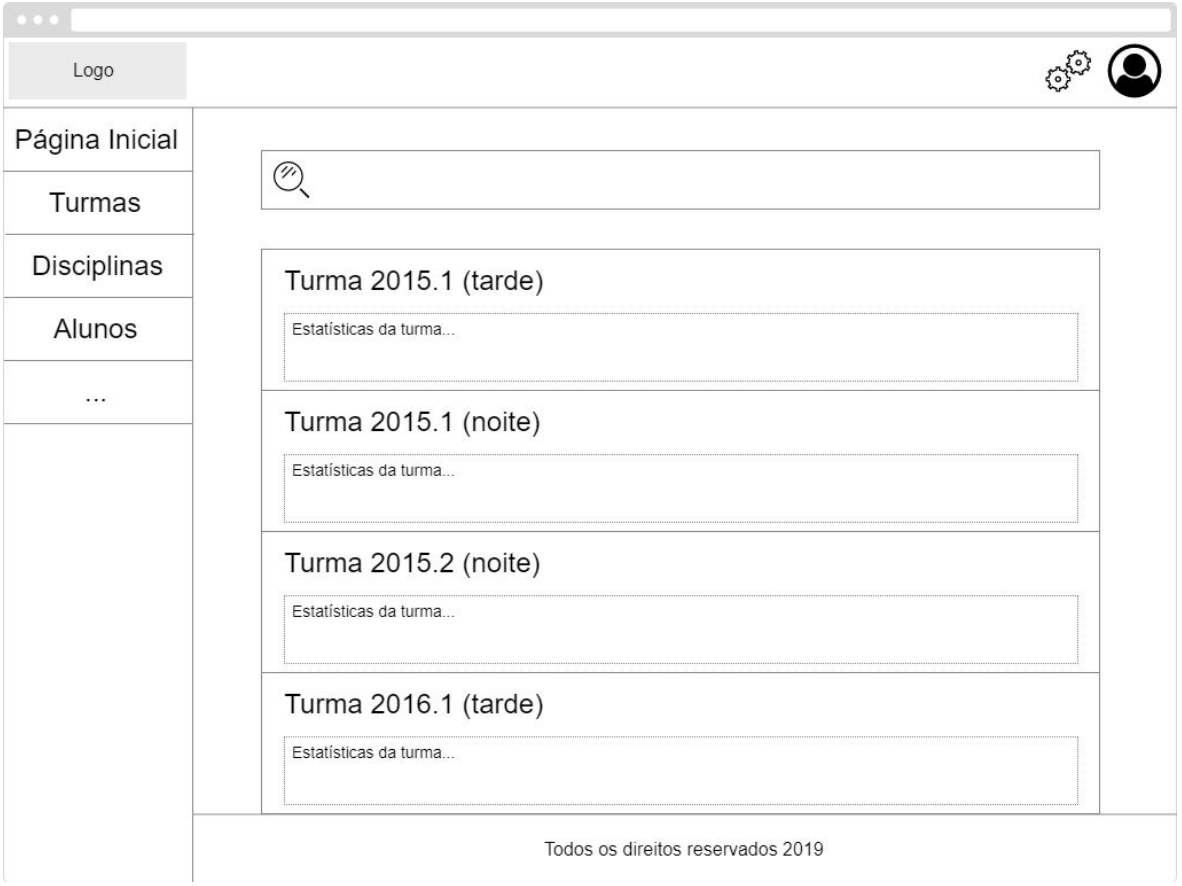
DV1

Fluxo Principal	1. Sistema apresenta detalhes do aluno referenciado e uma lista com suas matrículas (DV1, E1). 2. Usuário seleciona uma matrícula da lista. 3. Inicia o caso de uso [UC06 - Visualizar Disciplina].
Fluxo Alternativo	N/A
Fluxo de exceção	**E1**. Nenhum aluno referenciado : 1. Sistema exibe mensagem informando que nenhum aluno foi referenciado.

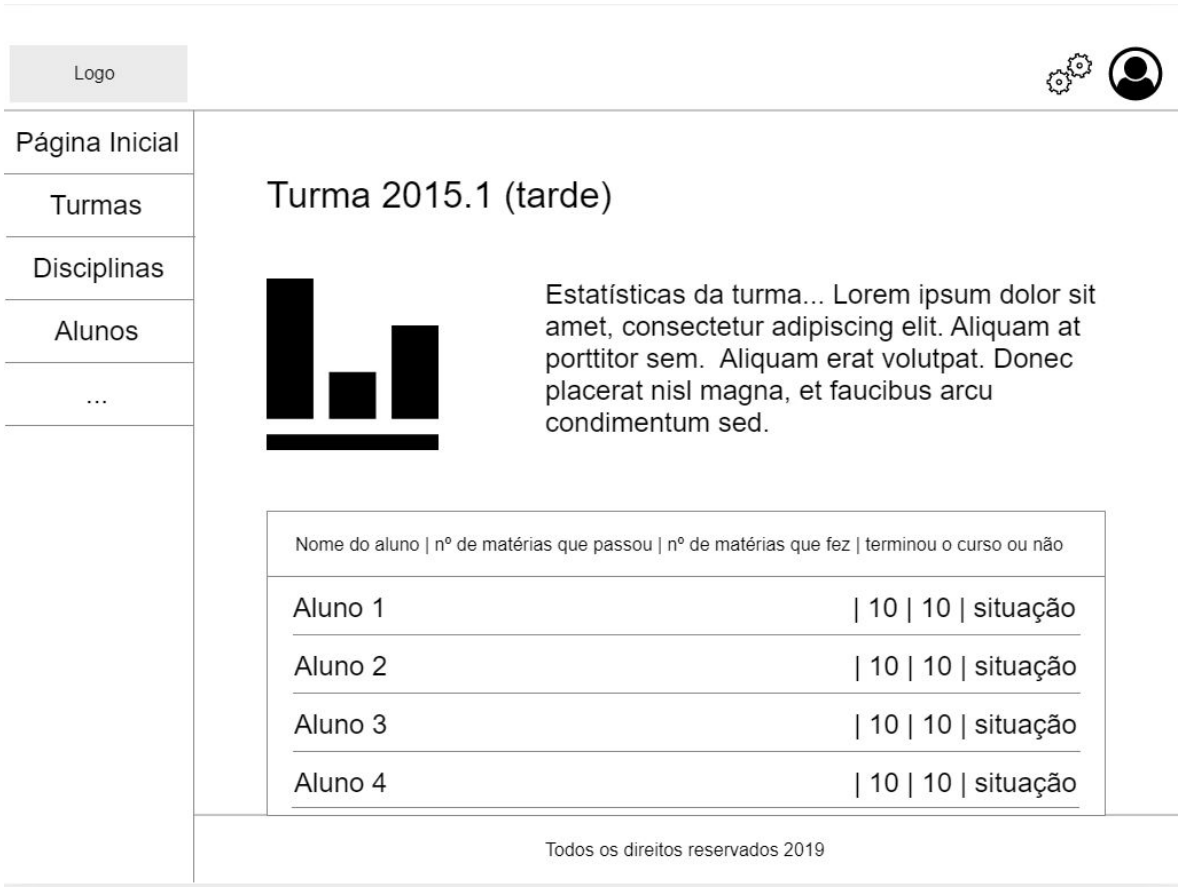
Fonte: Os autores (2019).

QUADRO 8 - DESCRIÇÃO UC07: VISUALIZAR DADOS TURMAS

Nome	UC07 - Visualizar dados Turmas
Ator principal	Docente
Descrição	Usuário pode navegar por uma lista de turmas e visualizar dados relacionados
Pre-condição	Usuário deve estar logado no sistema
Pós-condição	
Data views	



DV1

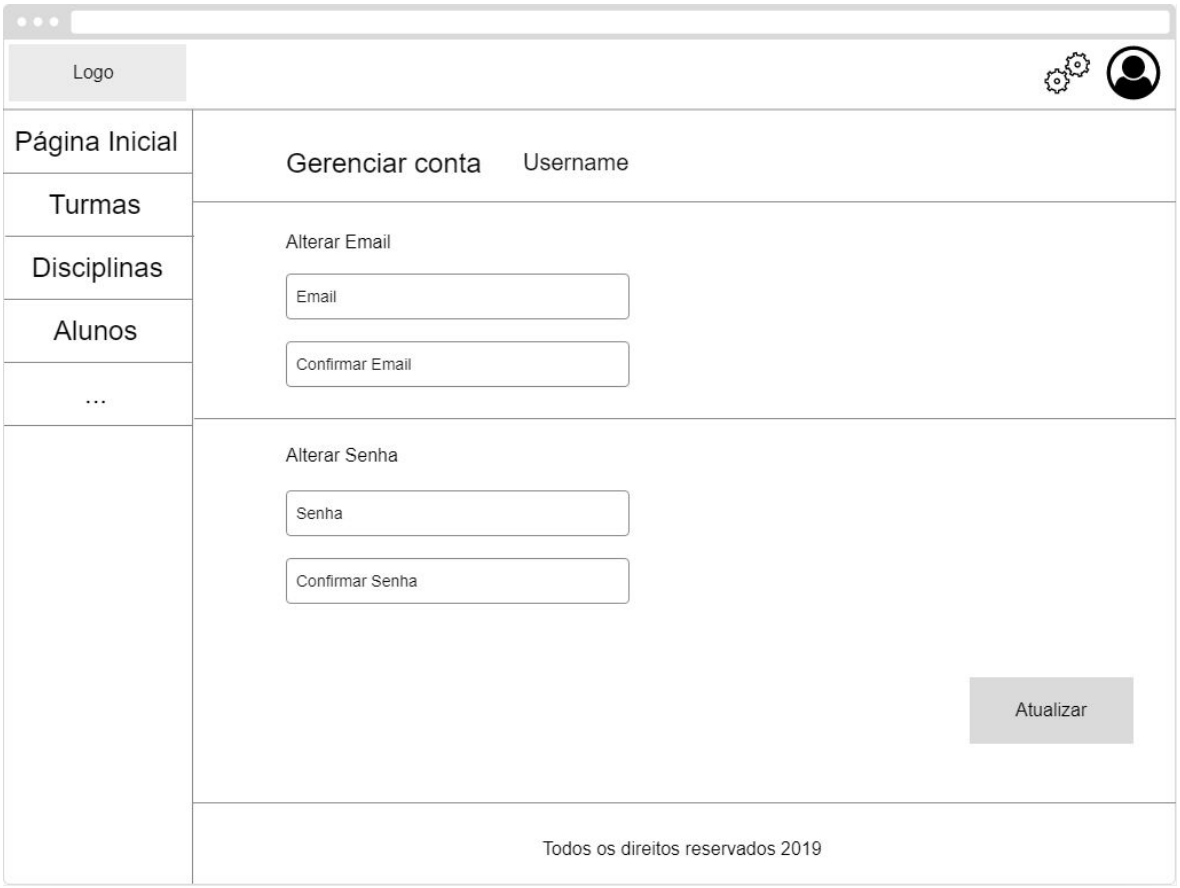


DV2

Fluxo Principal	1. Sistema apresenta uma lista com todas as turmas cadastradas e um campo de pesquisa (DV1, A1, E1). 2. Usuário seleciona uma turma da lista. 3. Sistema apresenta detalhes da turma selecionada e uma lista com os alunos da turma (DV2, E1). 4. Usuário seleciona um aluno da lista. 5. Inicia o caso de uso [UC06 - Visualizar Aluno].
Fluxo Alternativo	**A1**. Pesquisa : 1. Usuário digita algo no campo de pesquisa.(E2) 2. Lista é filtrada com o input do Usuário.(E1)
Fluxo de exceção	**E1**. Nenhuma turma encontrada : 1. Sistema exibe mensagem informando que nenhuma turma foi encontrada. **E2**. Input incorreto : 1. Sistema exibe mensagem informando que input no campo de pesquisa está incorreto.

Fonte: Os autores (2019).

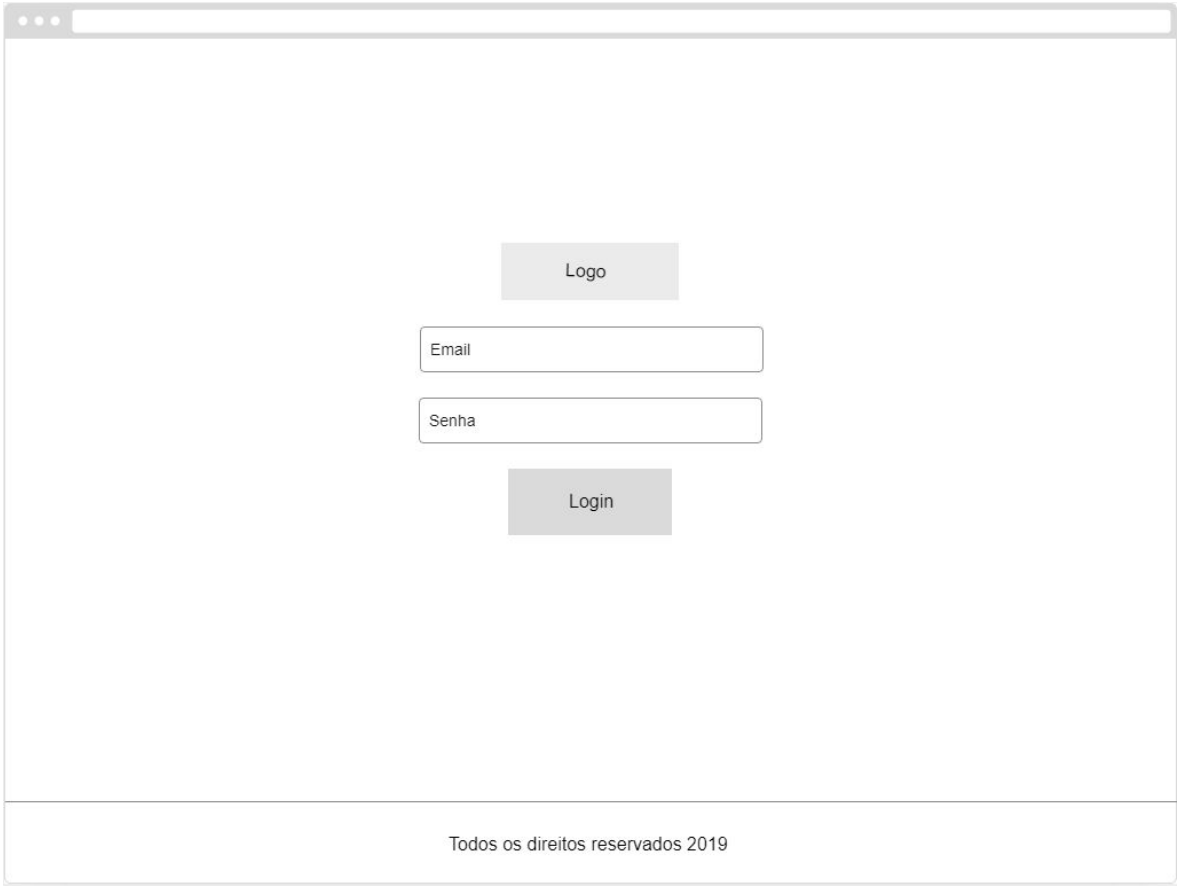
QUADRO 9 - DESCRIÇÃO UC08: GERENCIAR CONTA

Nome	UC08 - Gerenciar conta
Ator principal	Docente
Descrição	Usuário pode visualizar e editar dados da conta logada
Pré-condição	Usuário deve estar logado no sistema
Pós-condição	Salvar dados caso sejam editados
Data views	
 DV1	
Fluxo Principal	1. Sistema apresenta um formulário com os dados da conta logada (DV1). 2. Usuário modifica seus dados e clica em “atualizar” (E1). 3. Sistema atualiza dados do usuário (E2). 4. O sistema exibe uma mensagem de sucesso. 5. Caso de uso encerrado
Fluxo Alternativo	N/A
Fluxo de exceção	**E1**. Campo preenchido incorretamente:

	1. O sistema destaca os campos incorretos para a correção. 2. Progressão no fluxo é bloqueada enquanto campo estiver incorreto. **E2**. Erro ao cadastrar um usuário no sistema: 1. O sistema exibe uma mensagem de erro. 2. O usuário é encaminhado para o formulário.

Fonte: Os autores (2019).

QUADRO 10 - DESCRIÇÃO UC09: LOGIN

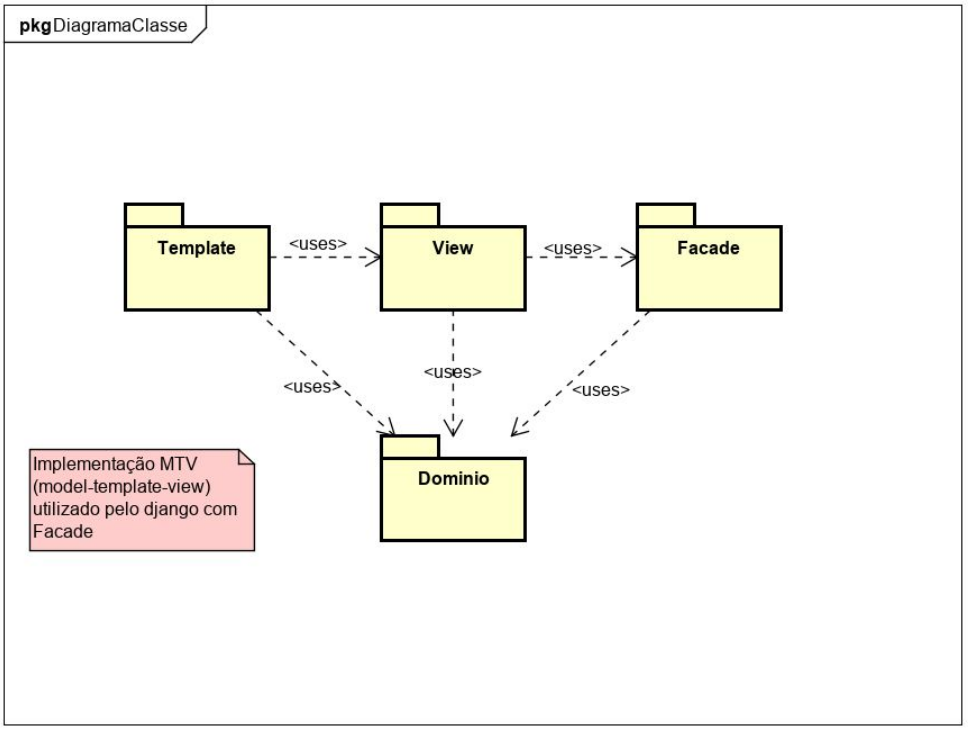
Nome	UC09 - Login
Ator principal	Visitante
Descrição	Um usuário acessa sua conta.
Pre-condição	Cadastro do usuário.
Pós-condição	Usuário logar no sistema.
Data views	
 DV1	
Fluxo Principal	1. O sistema exibe um formulário para inserção das credenciais de acesso (email e senha) (DV1). 2. O usuário clica em “Logar”. 3. O sistema verifica credenciais (E1). 4. O sistema exibe a tela inicial referente ao tipo de usuário logado. 5. Caso de uso é encerrado.
Fluxo Alternativo	N/A
Fluxo de exceção	**E1**. Usuário ou senha incorretos:

	1. O sistema exibe mensagem “usuário/senha incorretos”. 2. Caso de uso é reiniciado. **E2**. Usuário inexistente: 1. O sistema exibe mensagem “usuário inexistente”. 2. Caso de uso é reiniciado.

Fonte: Os autores (2019).

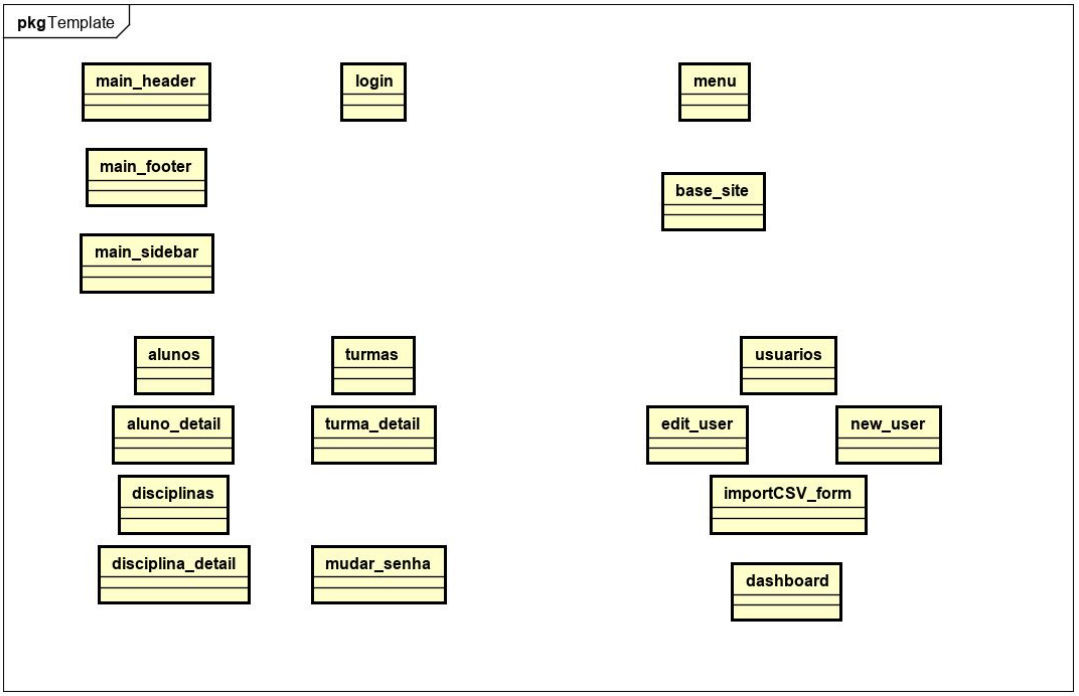
APÊNDICE C - DIAGRAMAS DE CLASSES

FIGURA 18 - DIAGRAMA DE CLASSES: DIAGRAMA GERAL



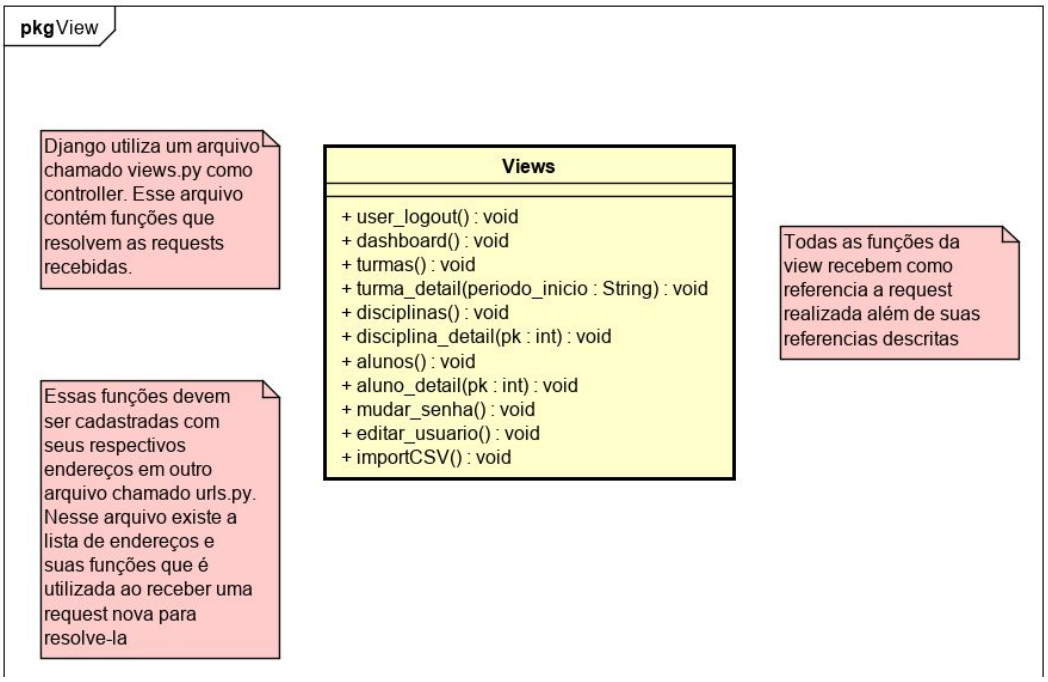
Fonte: Os autores (2019).

FIGURA 19 - DIAGRAMA DE CLASSES: DIAGRAMA DO PACOTE TEMPLATE



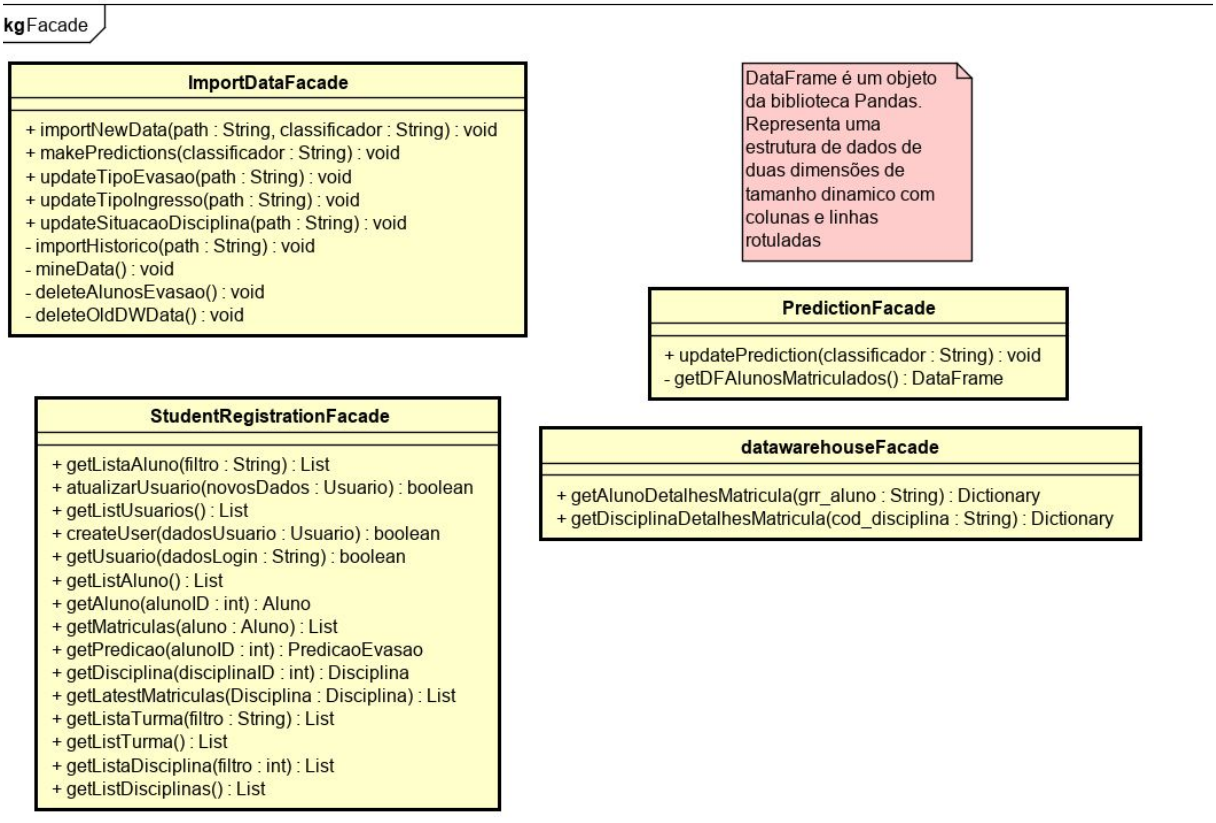
Fonte: Os autores (2019).

FIGURA 20 - DIAGRAMA DE CLASSES: DIAGRAMA DO PACOTE VIEW



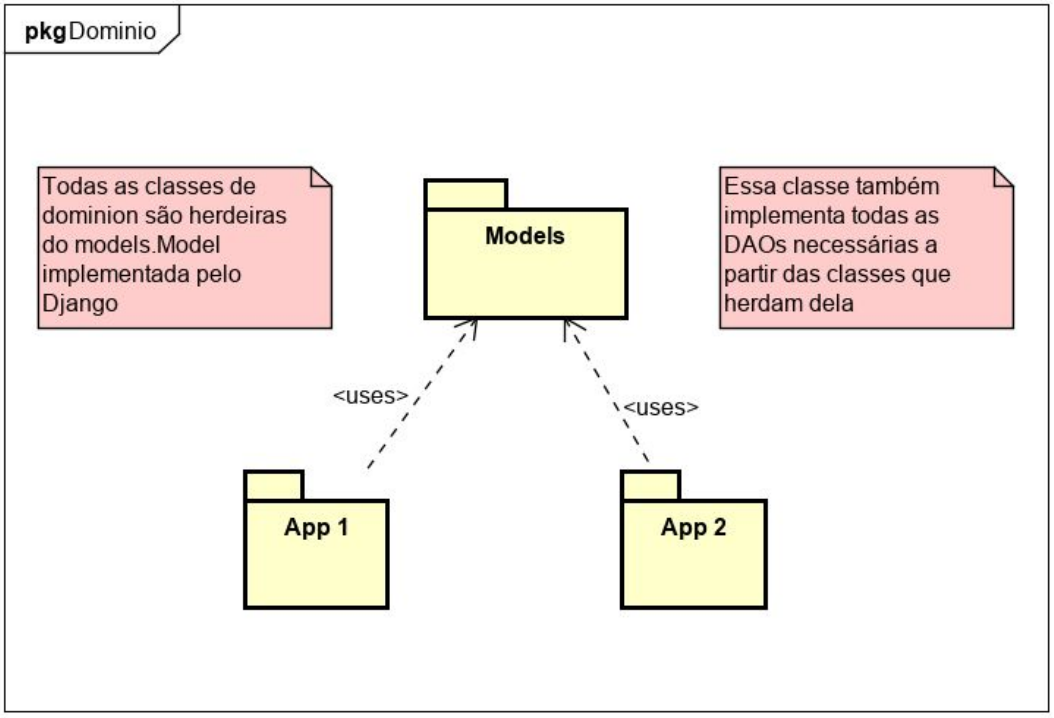
Fonte: Os autores (2019).

FIGURA 21 - DIAGRAMA DE CLASSES: DIAGRAMA DO PACOTE FACADE



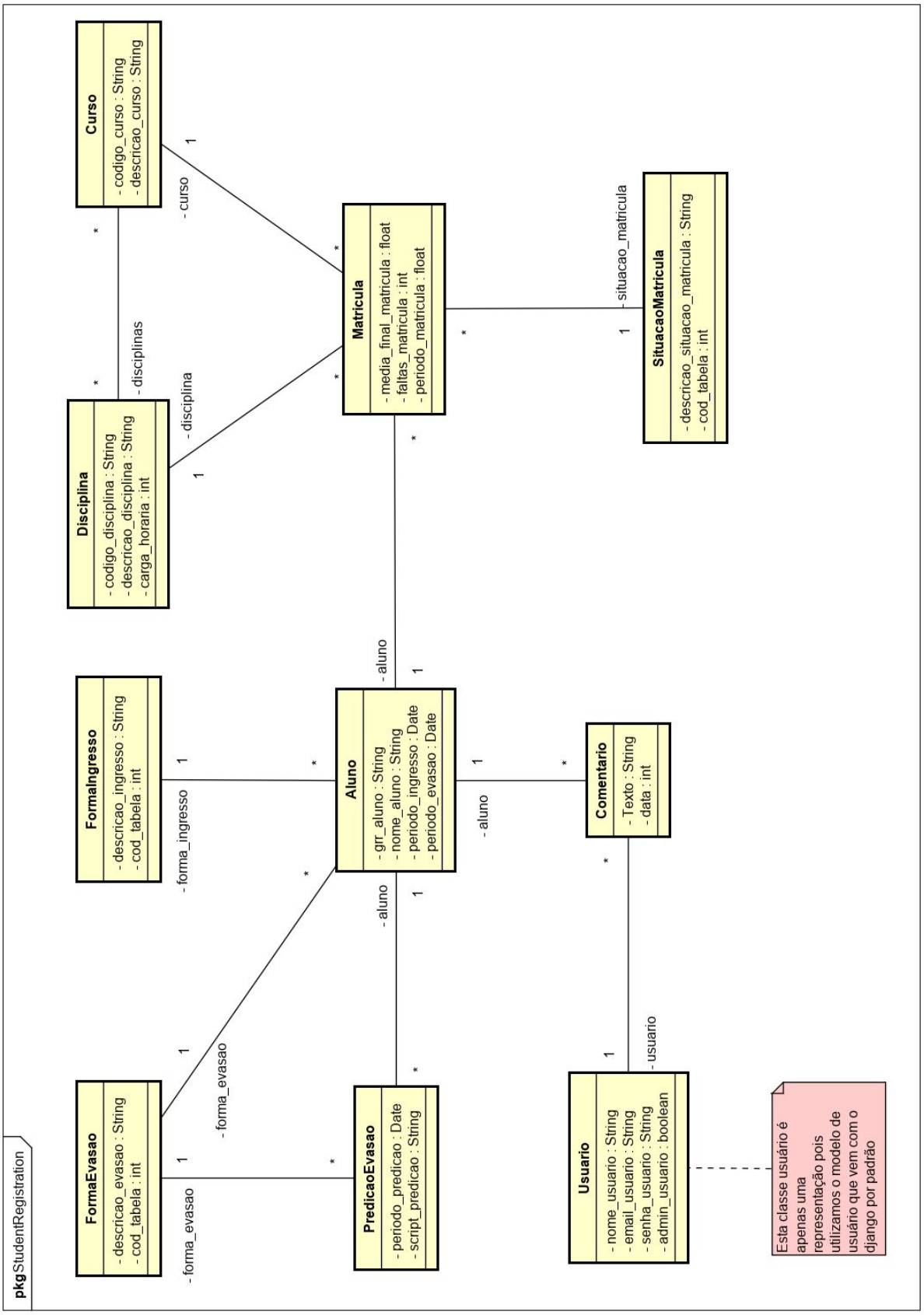
Fonte: Os autores (2019).

FIGURA 22 - DIAGRAMA DE CLASSES: DIAGRAMA DO PACOTE DOMÍNIO



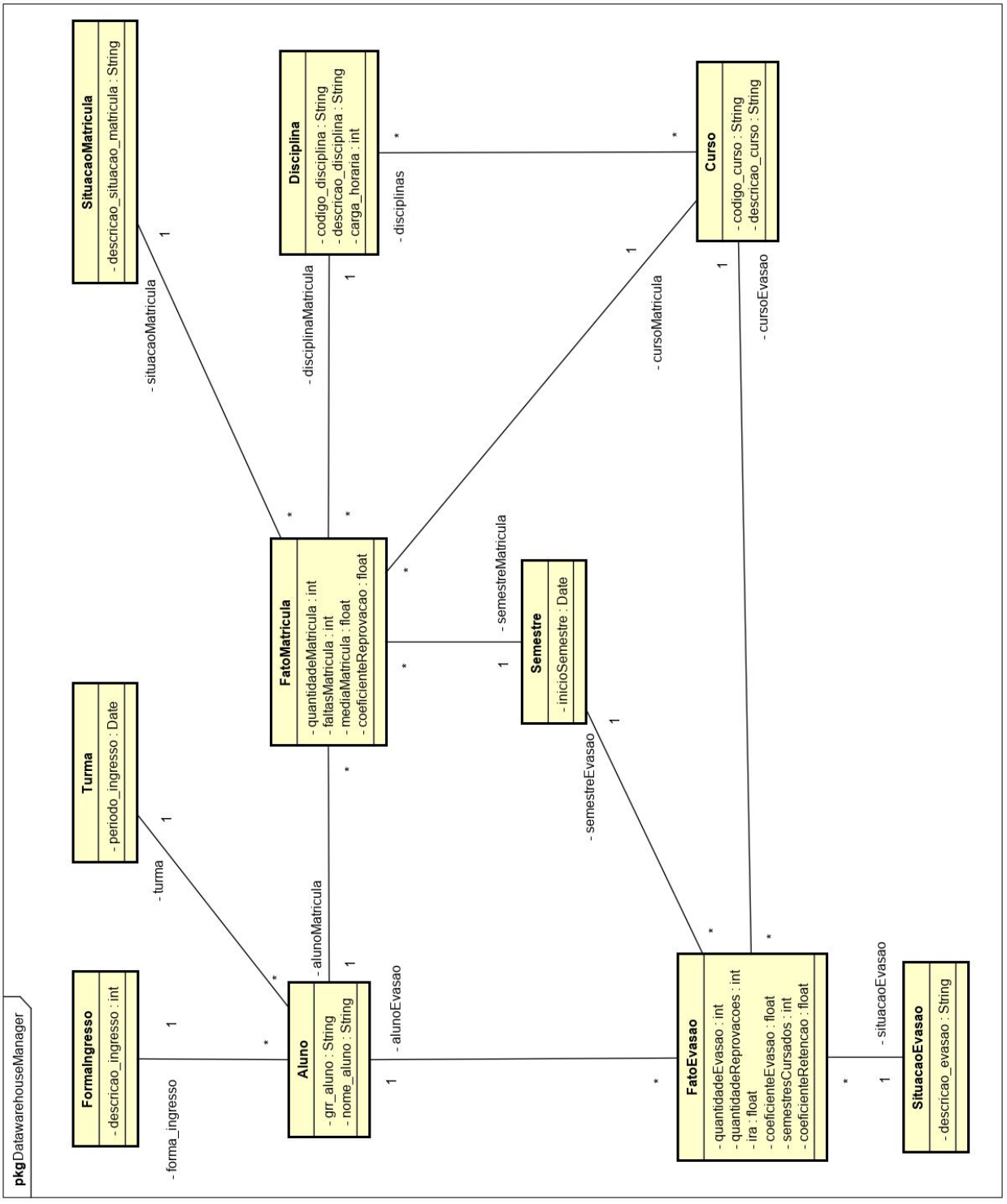
Fonte: Os autores (2019).

FIGURA 23 - DIAGRAMA DE CLASSES: DIAGRAMA DO PACOTE APP1



Fonte: Os autores (2019).

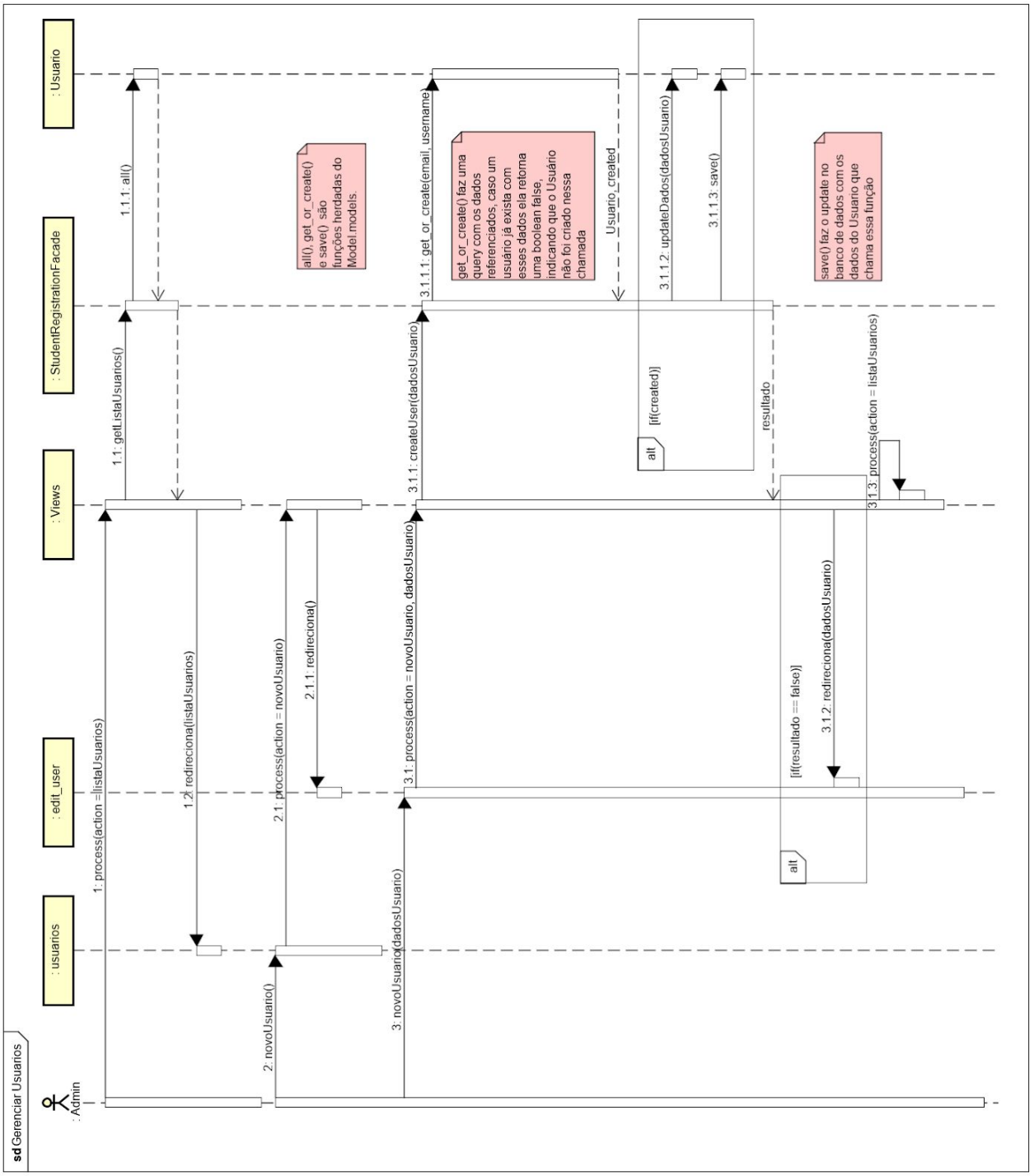
FIGURA 24 - DIAGRAMA DE CLASSES: DIAGRAMA DO PACOTE APP2



Fonte: Os autores (2019).

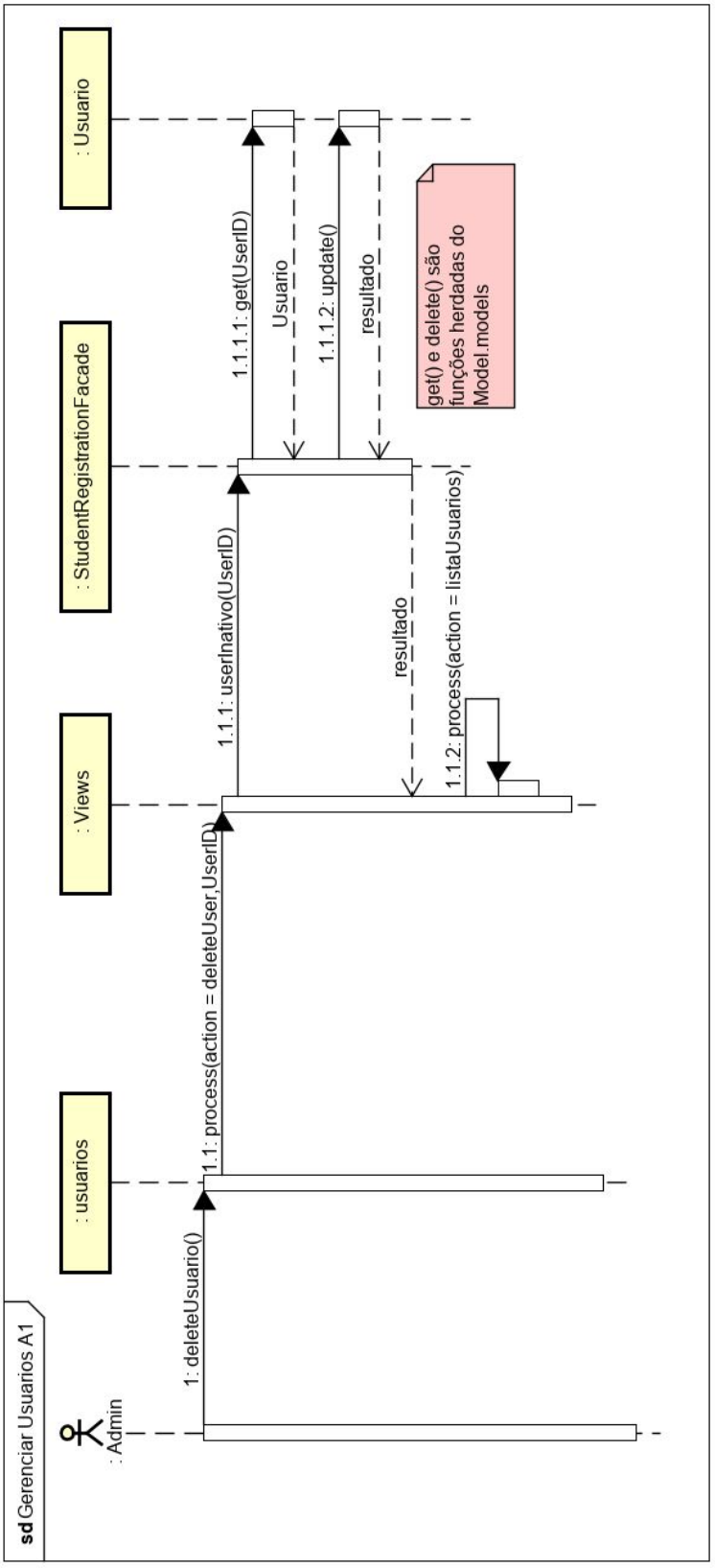
APÊNDICE D - DIAGRAMAS DE SEQUÊNCIA

FIGURA 25 - DIAGRAMA DE SEQUÊNCIA: GERENCIAR USUÁRIOS



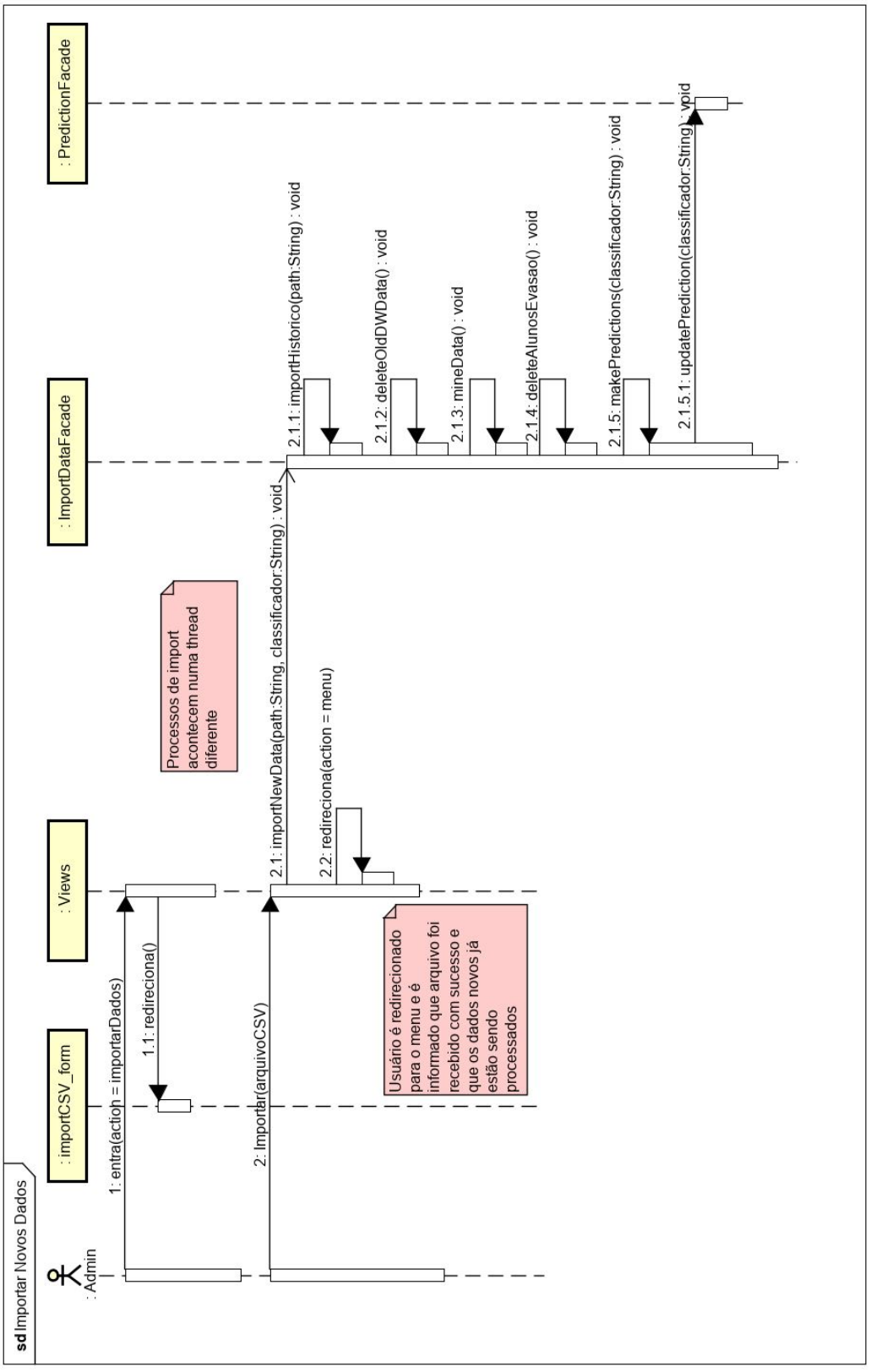
Fonte: Os autores (2019).

FIGURA 26 - DIAGRAMA DE SEQUÊNCIA: GERENCIAR USUÁRIOS - FLUXO ALTERNATIVO 1



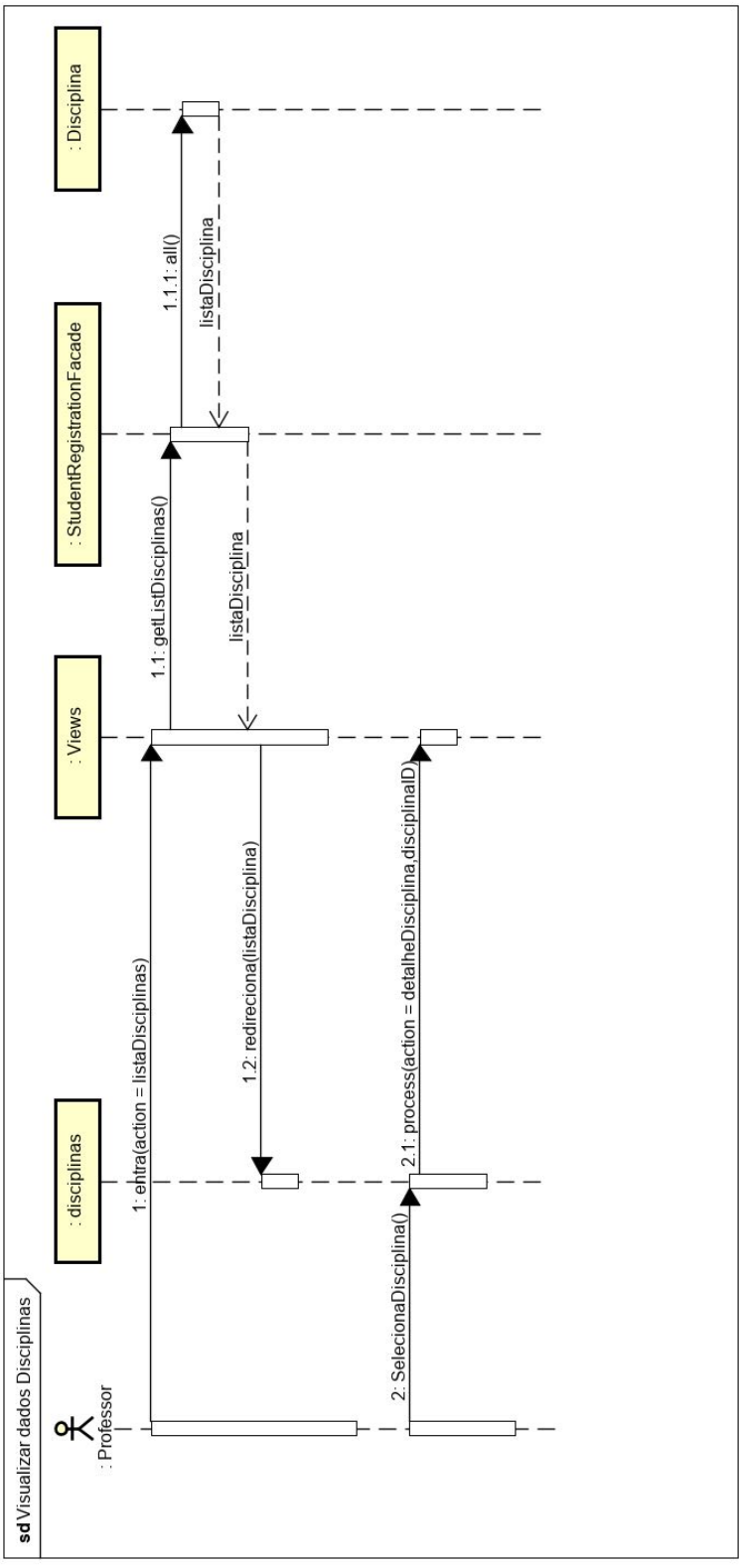
Fonte: Os autores (2019).

FIGURA 27 - DIAGRAMA DE SEQUÊNCIA: IMPORTAR NOVOS DADOS



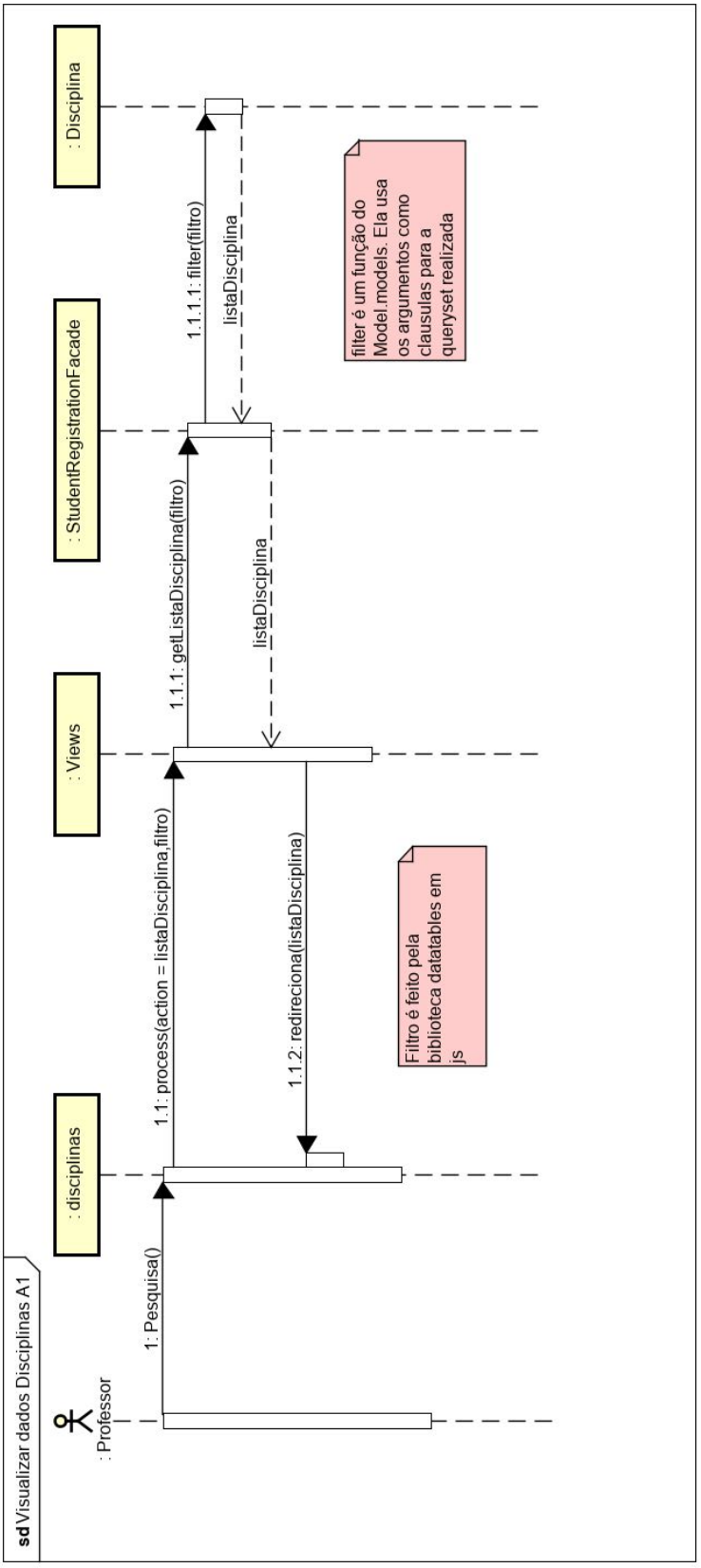
Fonte: Os autores (2019).

FIGURA 28 - DIAGRAMA DE SEQUÊNCIA: VISUALIZAR DADOS DISCIPLINAS



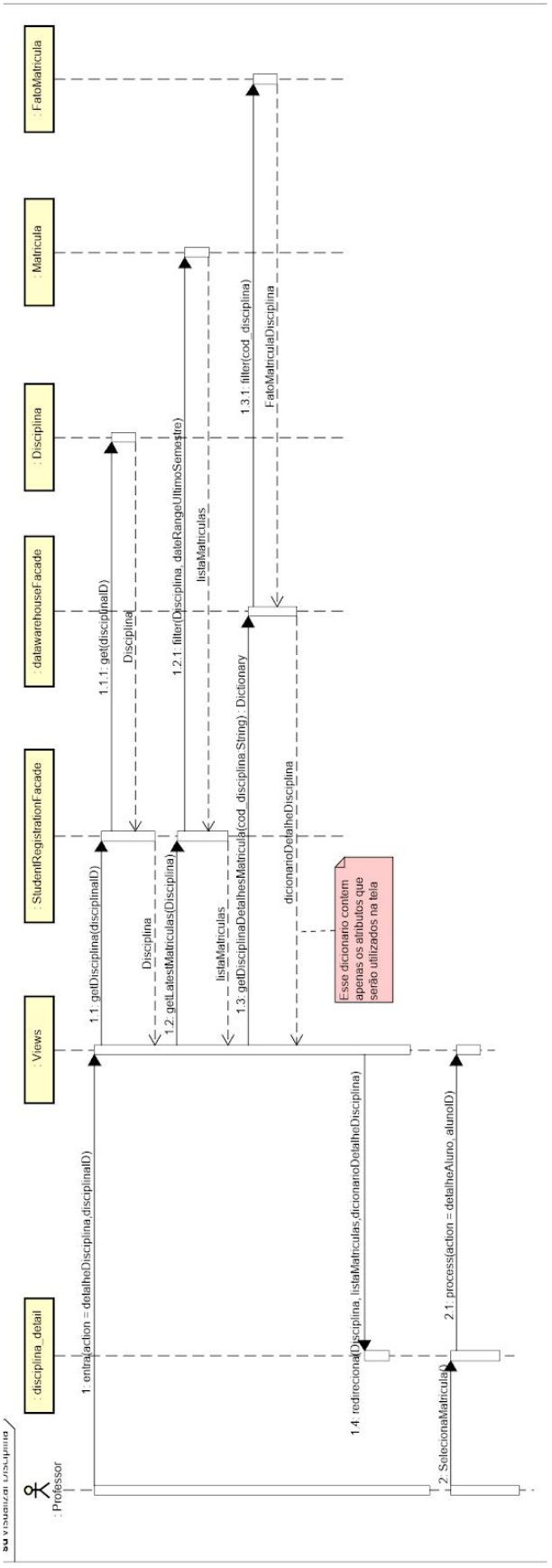
Fonte: Os autores (2019).

FIGURA 29 - DIAGRAMA DE SEQUÊNCIA: VISUALIZAR DADOS DISCIPLINAS - FLUXO ALTERNATIVO 1



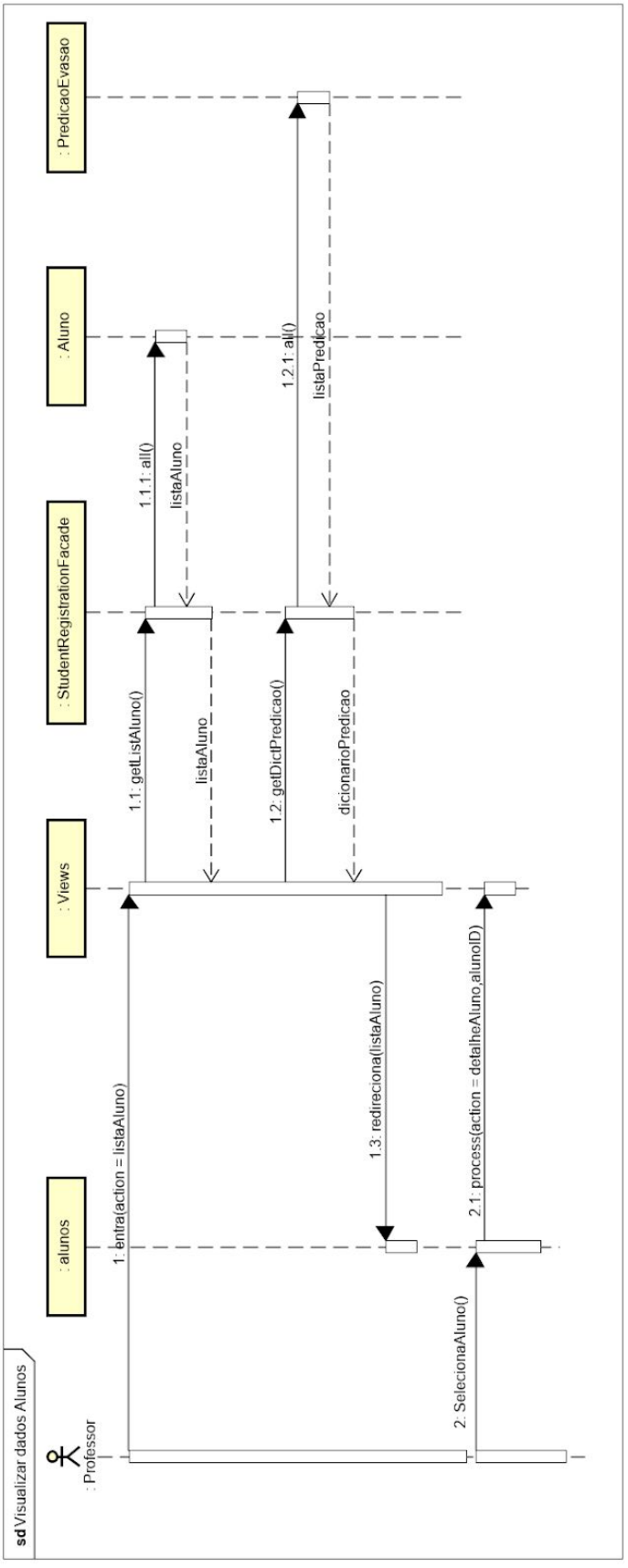
Fonte: Os autores (2019).

FIGURA 30 - DIAGRAMA DE SEQUÊNCIA: VISUALIZAR DISCIPLINA



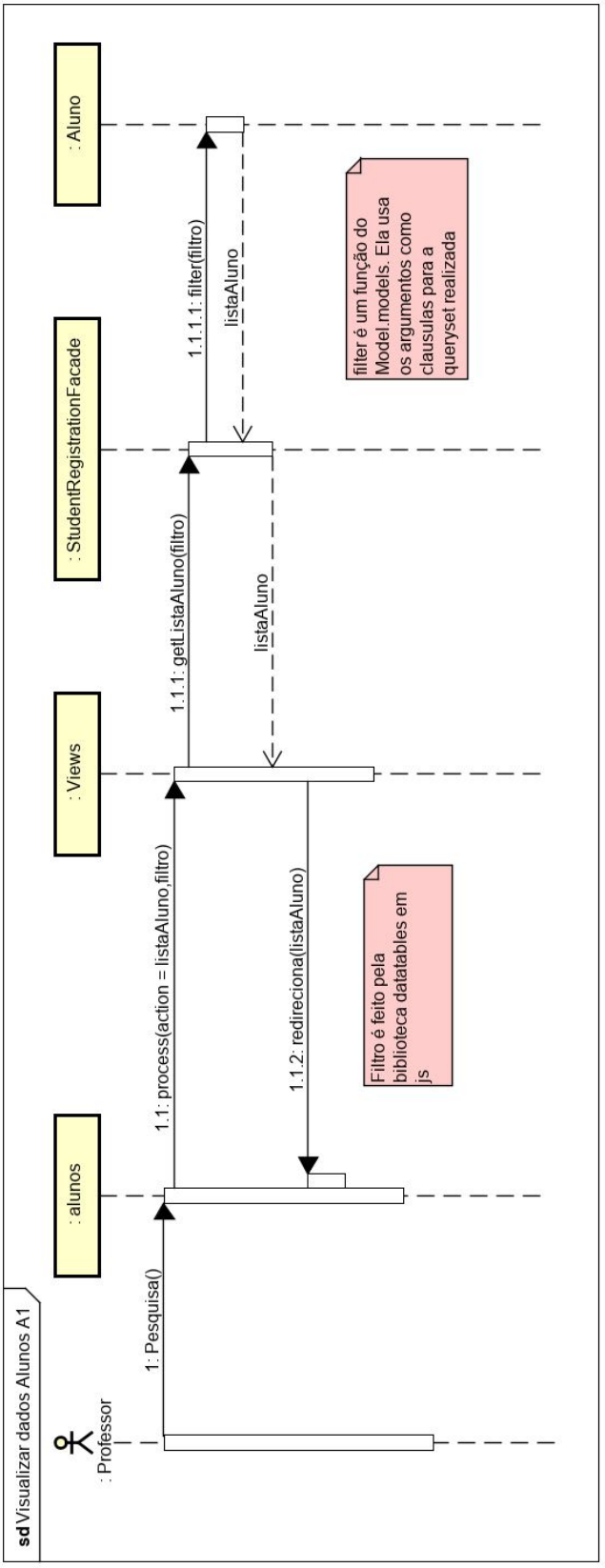
Fonte: Os autores (2019).

FIGURA 31 - DIAGRAMA DE SEQUÊNCIA: VISUALIZAR DADOS ALUNOS



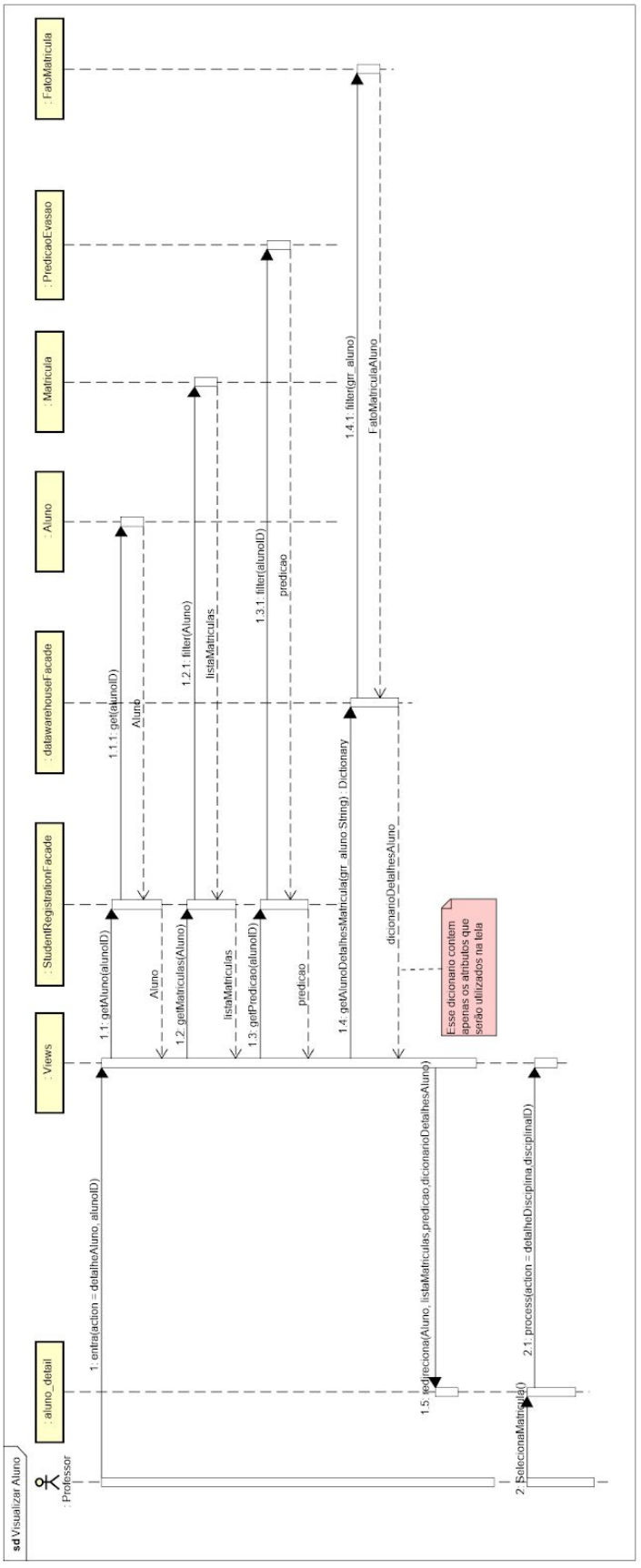
Fonte: Os autores (2019).

FIGURA 32 - DIAGRAMA DE SEQUÊNCIA: VISUALIZAR DADOS ALUNOS - FLUXO ALTERNATIVO 1



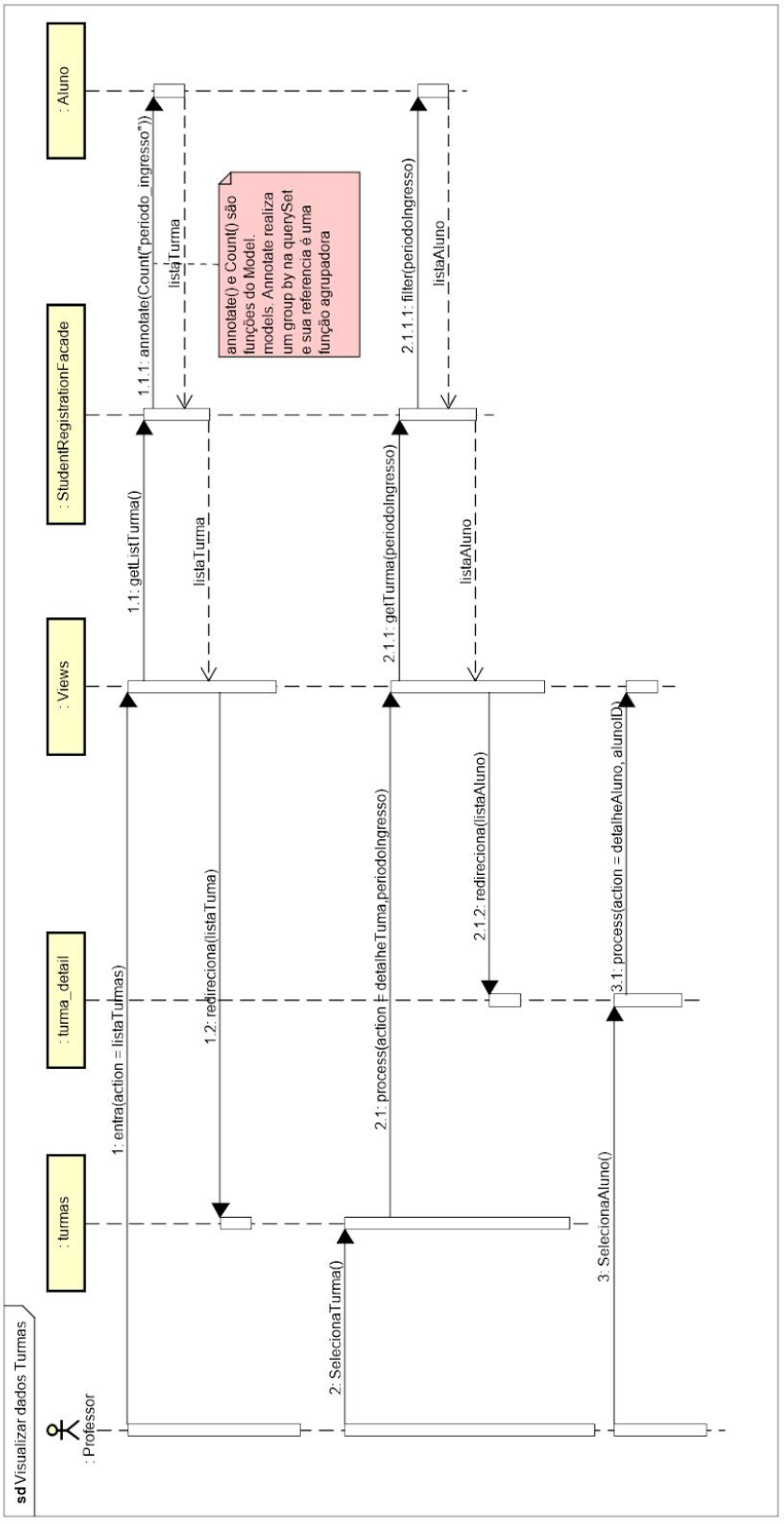
Fonte: Os autores (2019).

FIGURA 33 - DIAGRAMA DE SEQUÊNCIA: VISUALIZAR ALUNO



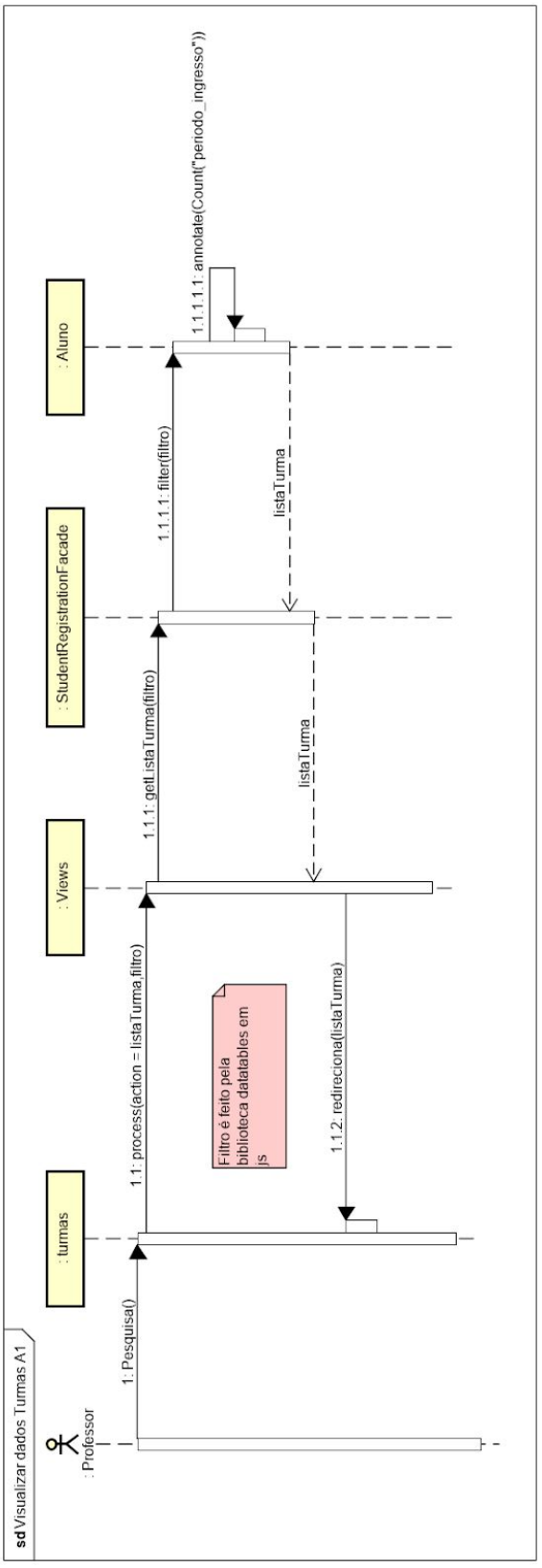
Fonte: Os autores (2019).

FIGURA 34 - DIAGRAMA DE SEQUÊNCIA: VISUALIZAR DADOS TURMAS



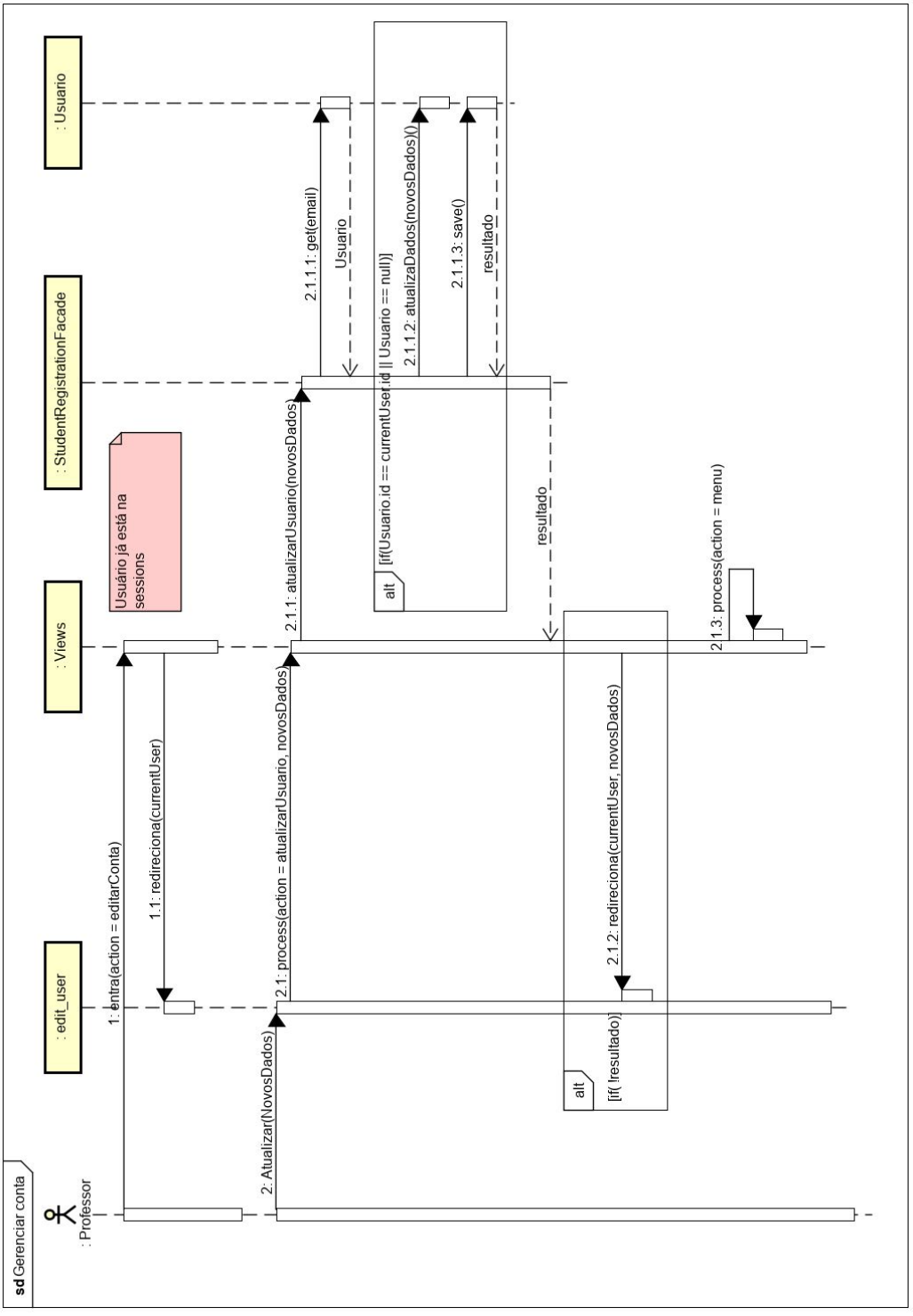
Fonte: Os autores (2019).

FIGURA 35 - DIAGRAMA DE SEQUÊNCIA: VISUALIZAR DADOS TURMAS - FLUXO ALTERNATIVO 1



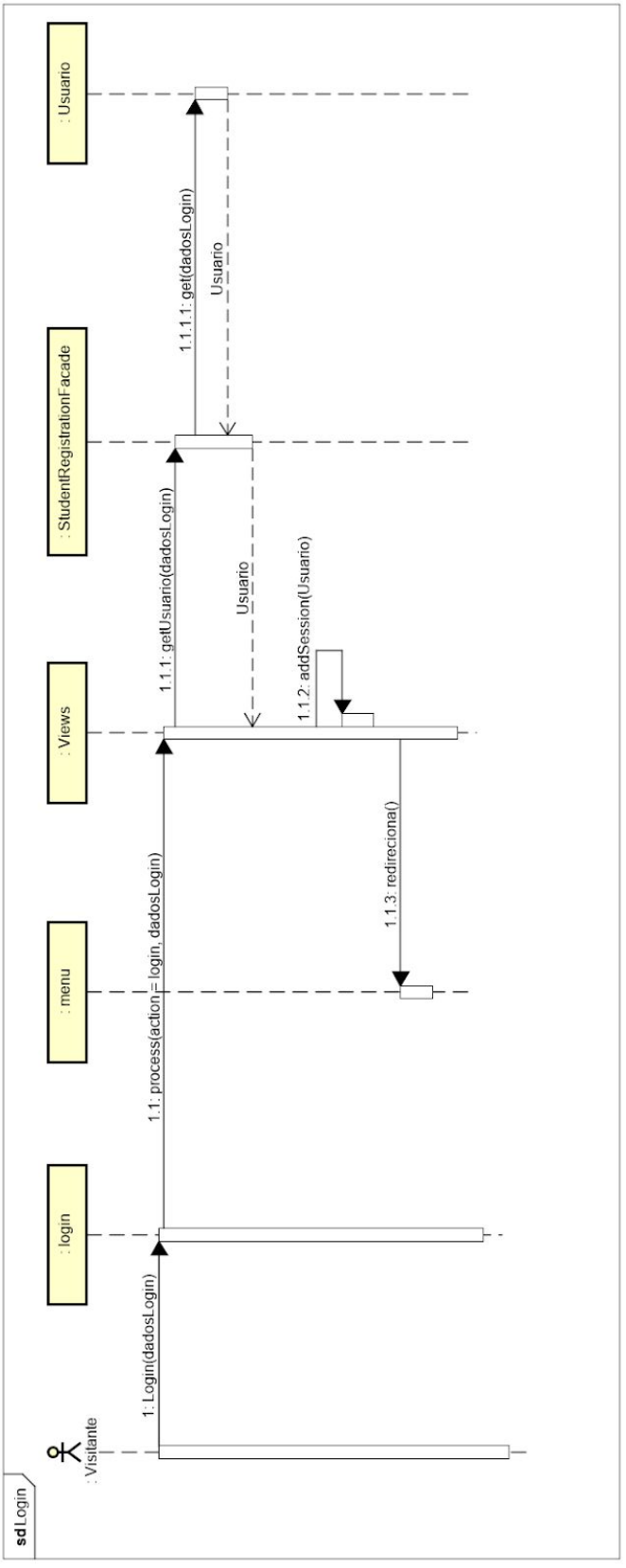
Fonte: Os autores (2019).

FIGURA 36 - DIAGRAMA DE SEQUÊNCIA: GERENCIAR CONTA



Fonte: Os autores (2019).

FIGURA 37 - DIAGRAMA DE SEQUÊNCIA: LOGIN



Fonte: Os autores (2019).