

CENTRO UNIVERSITÁRIO DE SANTA FÉ DO SUL
DEPARTAMENTO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

Fabrício Gonçalves Ferreira

**SISTEMA PARA PREVISÃO DE EVASÃO ESCOLAR
UTILIZANDO INTELIGÊNCIA ARTIFICIAL
SPEEIA**

Santa Fé do Sul – São Paulo

2023

Fabrício Gonçalves Ferreira

**SISTEMA PARA PREVISÃO DE EVASÃO ESCOLAR
UTILIZANDO INTELIGÊNCIA ARTIFICIAL
SPEEIA**

Trabalho de Conclusão de Curso submetido ao Departamento de Tecnologia em Análise e Desenvolvimento de Sistemas - UNIFUNEC como requisito parcial para a obtenção do título de Bacharel em Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador(a): José Paulo Codinhoto
Coorientador(a): Marcos Antônio Estremote

Santa Fé do Sul – São Paulo

2023

Dedico este projeto aos meus irmãos, aos meus pais, aos meus amigos, ao meu coorientador e ao meu orientador.

AGRADECIMENTOS

Ao meu professor e amigo, Gustavo Henrique Massaro professor do curso de Análise e Desenvolvimento de Sistemas da Unifunec de Santa Fé do Sul, por todas as orientações para o software e principalmente para a diagramação do software

Ao meu irmão, Felipe de Oliveira Alves, que apesar de não fazer parte da área da Tecnologia da Informação, me orientou com a ortografia, documentação e os processos do meu Trabalho de Conclusão de Curso, permitindo aprimorar a qualidade do projeto, além de que, me deu apoio psicológico e financeiro. Sem o apoio dele, definitivamente este projeto não teria a qualidade que tem hoje.

Ao meu coorientador e amigo, Marcos Antônio Estremote professor do curso de Análise e Desenvolvimento de Sistemas da Unifunec de Santa Fé do Sul, que permitiu que está documentação tenha a qualidade atual, ajudando na diagramação, na correção e na estrutura da documentação, além de que foi um ótimo amigo ajudando em tarefas além do seu escopo.

Ao José Paulo Codinhoto, meu orientador e coordenador do curso de Análise e Desenvolvimento de Sistemas da Unifunec de Santa Fé do Sul, além de um amigo, sendo a principal pessoa responsável por esse projeto existir, me provendo conselhos, broncas e orientações quando necessário. Permitindo que o tema, o projeto e o software se desenvolvesse até o estado atual em que se encontra.

*Um ser humano deve transformar informação em inteligência ou conhecimento.
Tendemos a esquecer que nenhum computador jamais fará uma nova pergunta.*
(Grace Hopper)

RESUMO

A evasão escolar é um problema que está presente em todas as instituições de ensino do país e do mundo. A evasão pode ocorrer mediante a diversos fatores, sejam estes internos ou externos a instituição de ensino, o que pode prejudicar o bom funcionamento dessas instituições que dependem do número de alunos, sejam estas públicas ou privadas. A evasão escolar efetua-se em três estágios, antes da evasão, quando o discente não decidiu evadir-se, durante a evasão, quando o discente decide evadir e após a evasão, quando ele já evadiu. O trabalho foca em antes da evasão, onde possui menos impactos para todos os envolvidos, para isso, é primordial fazer o acompanhamento das características de centenas de alunos, no entanto, fazer isso manualmente, torna-se inviável e muito custoso. Então é vital o auxílio das tecnologias e das ferramentas atuais para que esse processo possa ser agilizado e abrangido. Vale ressaltar a tecnologia de Inteligência Artificial, ela busca realizar o levantamento e a exploração de informações por meio de algoritmos que recebem dados. Portanto, foi desenvolvido um sistema capaz de indicar possíveis evasões escolares, para isso, ele usa a inteligência artificial para fazer o acompanhamento de características do discente, especificamente das notas e da frequência. Assim foi estabelecido uma ferramenta de apoio e gerenciamento das informações, além de análises automatizadas para que os usuários tenham informações que estejam além da percepção primária dos indivíduos e que a inteligência artificial consiga encontrar padrões que vão além da percepção humana. Portanto, foi estabelecido critérios baseados em notas e frequência para identificar os possíveis candidatos a evasão escolar, mediante a esta análise e comparação com situações anteriores, o sistema busca encontrar padrões e alertar aos gestores escolares, demonstrando uma estimativa de possíveis chances de evasão, possibilitando assim que os gestores tomem ações.

Palavras-chave: Tecnologia da Informação. Gestão Escolar. Educação. Software.

ABSTRACT

School dropout is a problem that is present in all educational institutions in the country and the world. Evasion can occur due to several factors, whether internal or external to the educational institution, which can harm the proper functioning of these institutions that depend on the number of students, whether public or private. Dropping out of school takes place in three stages, before dropping out, when the student has not decided to drop out, during dropping out, when the student decides to drop out and after dropping out, when he has already dropped out. The work focuses on before dropout, where it has less impact on everyone involved. To achieve this, it is essential to monitor the characteristics of hundreds of students, however, doing this manually becomes unfeasible and very costly. Therefore, the assistance of current technologies and tools is vital so that this process can be streamlined and covered. It is worth highlighting Artificial Intelligence technology, which seeks to collect and explore information through algorithms that receive data. Therefore, a system was developed capable of indicating possible school dropouts. To achieve this, it uses artificial intelligence to monitor student characteristics, specifically grades and attendance. Thus, an information support and management tool was established, in addition to automated analysis so that users have information that is beyond the primary perception of individuals and that artificial intelligence can find patterns that go beyond human perception. Therefore, criteria based on grades and frequency were established to identify possible candidates for school dropout. Through this analysis and comparison with previous situations, the system seeks to find patterns and alert school managers, demonstrating an estimate of possible chances of dropout, enabling as soon as managers take action. **Keywords:** Information Technology. School Management. Education. Software.

LISTA DE ILUSTRAÇÕES

Figura 1 – Gráfico referente a pergunta “Qual o tipo de nota é atribuído aos alunos?”	31
Figura 2 – Gráfico referente a pergunta “Quantos professores podem lecionar em uma mesma matéria?”	31
Figura 3 – Gráfico referente a pergunta “O software acadêmico consegue analisar a evasão escolar na sua escola?”	32
Figura 4 – Gráfico referente a pergunta “Você usaria um software de previsão de evasão escolar?”	33
Figura 5 – Gráfico referente a pergunta “Conhece algum software de previsão da evasão escolar?”	33
Figura 6 – Gráfico referente a pergunta “Qual o nível de importância de 0 a 5 você dá para conhecer a taxa de evasão da instituição de ensino?”	34
Figura 7 – Gráfico referente a pergunta “Com que frequência você considera relevante atualizar as taxas de evasão?”	34
Figura 8 – Gráfico referente a pergunta “Você acha relevante que um software de evasão escolar notifique quando um aluno está com altas chances de evadir?”	35
Figura 9 – Gráfico referente a pergunta “Você acha interessante que seja possível criar grupos de alunos com mais chances de evadir dentre todos os alunos cadastrados?”	36
Figura 10 – Arquitetura MTV em forma de pilha	41
Figura 11 – Logo Oficial do HTML5	42
Figura 12 – Logo Não Oficial do CSS3	43
Figura 13 – Logo Não Oficial do JavaScript	43
Figura 14 – Logo Oficial do Python	44
Figura 15 – Logo Oficial do Bootstrap	45
Figura 16 – Logo Oficial do Django	46
Figura 17 – Logo Oficial do L ^A T _E X	46
Figura 18 – Logo Oficial do Git	47
Figura 19 – Logo Oficial do PostgreSQL	48
Figura 20 – Logo Não Oficial do SQL	49
Figura 21 – Logo Oficial do Overleaf	49
Figura 22 – Logo Não Oficial do GitBash	50
Figura 23 – Logo Oficial do Figma	51
Figura 24 – Logo Oficial do Visual Studio Code	51
Figura 25 – Logo Não Oficial do Illustrator	52
Figura 26 – Logo Oficial do GitHub	52
Figura 27 – Logo Não Oficial do PowerPoint	53

Figura 28 – Modelo Conceitual	55
Figura 29 – Modelo Lógico	56
Figura 30 – Modelo Físico	57
Figura 31 – Diagrama de Classe	73
Figura 32 – Diagrama de casos de uso	74
Figura 33 – Diagrama de sequência Entra	75
Figura 34 – Diagrama de sequência Sai	77
Figura 35 – Diagrama de sequência Cadastra Aluno	79
Figura 36 – Diagrama de sequência Consulta Aluno	81
Figura 37 – Diagrama de sequência Altera Aluno	83
Figura 38 – Diagrama de sequência Exclui Aluno	85
Figura 39 – Diagrama de sequência Altera Usuário	87
Figura 40 – Diagrama de sequência Cadastra Usuário	89
Figura 41 – Diagrama de sequência Consulta Usuário	91
Figura 42 – Diagrama de sequência Exclui Usuário	93
Figura 43 – Diagrama de sequência Cadastra Perfil	95
Figura 44 – Diagrama de sequência Altera Perfil	97
Figura 45 – Diagrama de sequência Exclui Perfil	99
Figura 46 – Diagrama de sequência Calcula Evasão Escolar	101
Figura 47 – Diagrama de sequência Gera Relatório EE	103
Figura 48 – Diagrama de sequência Lista Aluno	105
Figura 49 – Diagrama de implantação	107
Figura 50 – Explicação do nome do software	108
Figura 51 – Paleta de cores	109
Figura 52 – Exemplo da fonte ‘Josefin Sans’	110
Figura 53 – Exemplo da fonte ‘Montserrat’	110
Figura 54 – Logo azul	111
Figura 55 – Logo branca	111
Figura 56 – Matriz de correlação	119
Figura 57 – Gráfico de dispersão	120

LISTA DE TABELAS

Tabela 1 – Taxas de evasão nas redes publicas do Brasil	20
Tabela 2 – Taxas de evasão no ensino superior do Brasil	21

LISTA DE ABREVIATURAS E SIGLAS

INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
EAD	Ensino a Distância
ADS	Análise e Desenvolvimento de Sistemas
TCC	Trabalho de Conclusão de Curso
IA	Inteligência Artificial
POO	Programação Orientada a Objetos
UML	<i>Unified Modeling Language</i>
BD	Banco de dados
SGBD	Sistema Gerenciador de Banco de Dados
DBMS	<i>Data Base Management System</i>
SciELO	<i>Scientific Electronic Library Online</i>
CAPES	Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
MEC	Ministério da Educação
IDE	<i>Integrated Development Environment</i>
VSC	<i>Visual Studio Code</i>
EDB	<i>Enterprise Data Base</i>
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
SQL	<i>Structured Query Language</i>
MVC	<i>Model View Controller</i>
MTV	<i>Model Template View</i>
WWW	<i>World Wide Web</i>
DVCS	<i>Distributed Version Control Systems</i>
SDVC	Sistemas Distribuídos de Controle de Versão

SGBDRO Sistema Gerenciador de Banco de Dados Objeto Relacional
HTTPS *Hyper Text Transfer Protocol Secure*

LISTA DE QUADROS

Quadro 1 – Quadro referente a pergunta 'Qual a taxa de evasão escolar?'	32
Quadro 2 – Comparação de estilos arquiteturais	39
Quadro 3 – Dicionário de dados da tabela Pessoa.	58
Quadro 4 – Dicionário de dados da tabela Administrador.	59
Quadro 5 – Dicionário de dados da tabela Usuário.	59
Quadro 6 – Dicionário de dados da tabela Grupo.	60
Quadro 7 – Dicionário de dados da tabela Aluno.	61
Quadro 8 – Dicionário de dados da tabela Aluno_Grupo.	61
Quadro 9 – Dicionário de dados da tabela Disciplina.	62
Quadro 10 – Dicionário de dados da tabela Aluno_Disciplina.	63
Quadro 11 – Dicionário de dados da tabela Nota.	64
Quadro 12 – Dicionário de dados da tabela Aula.	65
Quadro 13 – Dicionário de dados da tabela Disciplina_Aula.	66
Quadro 14 – Dicionário de dados da tabela Semestre.	67
Quadro 15 – Dicionário de dados da tabela Presença.	67
Quadro 16 – Dicionário de dados da tabela Ano.	68
Quadro 17 – Dicionário de dados da tabela Semestre_Ano.	69
Quadro 18 – Dicionário de dados da tabela Histórico.	70
Quadro 19 – Documentação do caso de uso Entra.	76
Quadro 20 – Documentação do caso de uso Sai.	78
Quadro 21 – Documentação do caso de uso Cadastra Aluno.	80
Quadro 22 – Documentação do caso de uso Consulta Aluno.	82
Quadro 23 – Documentação do caso de uso Altera Aluno.	84
Quadro 24 – Documentação do caso de uso Exclui Aluno.	86
Quadro 25 – Documentação do caso de uso Altera Usuário.	88
Quadro 26 – Documentação do caso de uso Cadastra Usuário.	90
Quadro 27 – Documentação do caso de uso Consulta Usuário.	92
Quadro 28 – Documentação do caso de uso Exclui Usuário.	94
Quadro 29 – Documentação do caso de uso Cadastra Perfil.	96
Quadro 30 – Documentação do caso de uso Altera Perfil.	98
Quadro 31 – Documentação do caso de uso Exclui Perfil.	100
Quadro 32 – Documentação do caso de uso Calcula Evasão Escolar.	102
Quadro 33 – Documentação do caso de uso Gera Relatório EE.	104
Quadro 34 – Documentação do caso de uso Lista Aluno.	106
Quadro 35 – Abreviações das variáveis.	113
Quadro 36 – Bibliotecas utilizadas.	114

LISTA DE CÓDIGOS

Código 1 – Código SQL da tabela Pessoa	58
Código 2 – Código SQL da tabela Administrador	59
Código 3 – Código SQL da tabela Usuário	60
Código 4 – Código SQL da tabela Grupo	60
Código 5 – Código SQL da tabela Aluno	61
Código 6 – Código SQL da tabela Aluno_Grupo	62
Código 7 – Código SQL da tabela Disciplina	63
Código 8 – Código SQL da tabela Aluno_Disciplina	64
Código 9 – Código SQL da tabela nota	65
Código 10 – Código SQL da tabela Aula	65
Código 11 – Código SQL da tabela Disciplina_Aula	66
Código 12 – Código SQL da tabela Semestre	67
Código 13 – Código SQL da tabela Presença	68
Código 14 – Código SQL da tabela Ano	69
Código 15 – Código SQL da tabela Semestre_Ano	70
Código 16 – Código SQL da tabela Histórico	71
Código 17 – Código da IA: Parte 1	115
Código 18 – Código da IA: Parte 2	116
Código 19 – Código da IA: Parte 3	117

LISTA DE EQUAÇÕES

Equação 1 – Equação de regressão linear múltipla	112
Equação 2 – Equação de regressão linear múltipla do modelo	113

SUMÁRIO

I	APRESENTAÇÃO	19
1	INTRODUÇÃO	20
1.1	CONTEXTUALIZAÇÃO	20
1.2	DELIMITAÇÃO	21
1.3	PROBLEMATIZAÇÃO	22
1.4	HIPÓTESES	22
1.5	OBJETIVO GERAL	22
1.6	OBJETIVOS ESPECÍFICOS	22
1.7	JUSTIFICATIVA	23
1.8	METODOLOGIA	24
1.8.1	<i>Natureza</i>	24
1.8.2	<i>Abordagem</i>	24
1.8.3	<i>Objetivos</i>	24
1.8.4	<i>Procedimentos</i>	24
1.8.5	<i>Método</i>	24
1.8.6	<i>Procedimentos</i>	24
1.8.7	<i>Conceitos</i>	25
1.8.8	<i>Tecnologias</i>	26
1.8.9	<i>Ferramentas</i>	27
1.9	ESTRUTURA DA MONOGRAFIA	28
II	DESENVOLVIMENTO	29
2	ENGENHARIA DE REQUISITOS	30
2.1	LEVANTAMENTO DE REQUISITOS	30
2.2	ESPECIFICAÇÃO DOS REQUISITOS	36
2.2.1	<i>Descrição dos requisitos funcionais</i>	36
3	ARQUITETURA DE SOFTWARE	38
3.1	ESTILO ARQUITETURAL	38
3.2	PADRÃO ARQUITETURAL	40
3.3	TECNOLOGIAS	41
3.3.1	HTML5	41
3.3.2	CSS	42
3.3.3	<i>JavaScript</i>	43

3.3.4	<i>Python</i>	44
3.3.5	<i>Bootstrap</i>	45
3.3.6	<i>Django</i>	45
3.3.7	<i>LaTeX</i>	46
3.3.8	<i>Git</i>	46
3.3.9	<i>PostgreSQL</i>	47
3.3.9.1	<i>Licença</i>	48
3.3.9.2	<i>Conformidade</i>	48
3.3.9.3	<i>Linguagem</i>	48
3.3.10	<i>SQL</i>	48
3.4	FERRAMENTAS	49
3.4.1	<i>Overleaf</i>	49
3.4.2	<i>Gitbash</i>	50
3.4.3	<i>Figma</i>	50
3.4.4	<i>Visual Studio Code</i>	51
3.4.5	<i>Illustrator</i>	51
3.4.6	<i>GitHub</i>	52
3.4.7	<i>brModelo</i>	52
3.4.8	<i>PgAdmin</i>	52
3.4.9	<i>PowerPoint</i>	53
4	ADMINISTRADOR DE DADOS	54
4.1	MODELAGEM DE DADOS	54
4.1.1	<i>Modelo de dados conceitual</i>	54
4.1.2	<i>Modelo de dados lógico</i>	56
4.1.3	<i>Modelo de dados físico</i>	57
4.2	DICIONÁRIO DE DADOS E SCRIPTS	58
4.2.1	<i>Tabela Pessoa</i>	58
4.2.2	<i>Tabela Administrador</i>	58
4.2.3	<i>Tabela Usuário</i>	59
4.2.4	<i>Tabela Grupo</i>	60
4.2.5	<i>Tabela Aluno</i>	61
4.2.6	<i>Tabela Aluno_Grupo</i>	61
4.2.7	<i>Tabela Disciplina</i>	62
4.2.8	<i>Tabela Aluno_Disciplina</i>	63
4.2.9	<i>Tabela Nota</i>	64
4.2.10	<i>Tabela Aula</i>	65
4.2.11	<i>Tabela Disciplina_Aula</i>	66
4.2.12	<i>Tabela Semestre</i>	66
4.2.13	<i>Tabela Presença</i>	67

4.2.14	<i>Tabela Ano</i>	68
4.2.15	<i>Tabela Semestre_Ano</i>	69
4.2.16	<i>Tabela Histórico</i>	70
5	MODELAGEM DE SOFTWARE	72
5.1	DIAGRAMA DE CLASSES	72
5.2	DIAGRAMA DE CASOS DE USO	73
5.3	DIAGRAMAS DE SEQUÊNCIA	75
5.3.1	<i>Pessoa</i>	75
5.3.1.1	<i>Entra</i>	75
5.3.1.2	<i>Sai</i>	76
5.3.1.3	<i>Cadastra Aluno</i>	78
5.3.1.4	<i>Consulta Aluno</i>	81
5.3.1.5	<i>Altera Aluno</i>	83
5.3.1.6	<i>Exclui Aluno</i>	85
5.3.2	<i>Administrador</i>	86
5.3.2.1	<i>Altera Usuário</i>	87
5.3.2.2	<i>Cadastra Usuário</i>	88
5.3.2.3	<i>Consulta Usuário</i>	90
5.3.2.4	<i>Exclui Usuário</i>	92
5.3.3	<i>Usuário</i>	94
5.3.3.1	<i>Cadastra Perfil</i>	95
5.3.3.2	<i>Altera Perfil</i>	96
5.3.3.3	<i>Exclui Perfil</i>	98
5.3.4	<i>IA</i>	100
5.3.4.1	<i>Calcula Evasão Escolar</i>	100
5.3.5	<i>Relatório</i>	102
5.3.5.1	<i>Gera Relatório EE</i>	102
5.3.6	<i>Sistema</i>	104
5.3.6.1	<i>Lista Aluno</i>	104
5.4	DIAGRAMA DE IMPLANTAÇÃO	106
6	DESIGN	108
6.1	NOME	108
6.2	PALETA DE CORES	108
6.3	FONTES	109
6.4	LOGOS	110
7	INTELIGÊNCIA ARTIFICIAL	112
7.1	MODELO	112

7.2	EQUAÇÃO	112
7.3	ALGORITMO	113
8	RESULTADOS	118
9	CONCLUSÃO	121
	REFERÊNCIAS	122

Parte I

Apresentação

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Desde de 1988, a educação foi reconhecida como um direito constitucional para todos na constituição federal, sendo um dever do estado e da família em promover a educação. Em outras palavras, o estado deve garantir que os jovens estejam na escola e que a família deverá apoiar o aluno na escola. Embora esteja garantida como um direito, o cenário atual demonstra que há obstáculos que atrapalham a garantia do direito a educação, entre essas adversidades, está a evasão escolar.

A evasão escolar é um conceito muito amplo na sociedade, não estabelecido de forma unânime, todavia, segundo o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), ela realiza-se quando o aluno efetua uma saída antes da conclusão do ano, série ou ciclo, independentemente do motivo ([BRASIL. MINISTÉRIO DA EDUCAÇÃO, 2017](#)). Por consequência, representa uma status de insucesso do aluno de acordo com o objetivo de promover o aluno a uma condição superior a de ingresso. Decorrente disso, status de insucessos representam aspectos negativos a educação brasileira, de modo que nos distâncias da garantia do direito a educação. Mas, para prosseguir com o entendimento, é preciso definir a taxa de evasão.

De acordo com [Abandono... \(2022\)](#), a taxa de evasão é, “indicador de fluxo ou transição escolar que representa o percentual de alunos matriculados em um ano letivo que não efetua matrícula no ano letivo seguinte, exceto os concluintes da última série do ensino médio.”. Seguindo as definições, é possível filtrar quais alunos atendem a definição de evasão escolar para se ter um parâmetro da situação atual. Mediante o exposto, surge a dúvida, qual a situação atual da evasão escolar sobre a educação brasileira?

A situação atual da evasão escolar no Brasil podem ser vistas em dois níveis, a nível de educação básica ou a nível de ensino superior. A nível de educação básica, foram encontrado dados dos anos finais do ensino fundamental e do ensino médio voltados a rede pública desde 2014 a 2019. Estes dados podem ser visualizados na tabela 1.

Tabela 1 – Taxas de evasão nas redes publicas do Brasil.

Ano	Ensino Fundamental	Ensino Médio
2014-2015	5,5%	12,2%
2015-2016	4,8%	9,9%
2016-2017	4,6%	10%
2017-2018	4,4%	9,4%
2018-2019	4,5%	10,7%

Fonte: [Evolução... \(20??\)](#)

Por outro lado, a nível de ensino superior, foram encontrados dados dos cursos presenciais e Ensino a Distância (EAD), estes dados vão desde 2015 a 2019. Eles podem ser visualizados na tabela 2.

Tabela 2 – Taxas de evasão no ensino superior do Brasil.

Ano	Ensino a Distância (EAD)	Presencial
2015	28,7%	18,4%
2016	30,4%	18,5%
2017	27,9%	18,6%
2018	31,6%	18,5%
2019	31,6%	18,4%

Fonte: [Mapa... \(2021\)](#)

Observa-se que houve uma diminuição da taxa de evasão no ensino básico, por outro ângulo, no ensino superior ela se manteve constante com pequenas oscilações. Mesmo com a diminuição da taxa no ensino básico, a existência da taxa por si só, já é uma preocupação, ainda mais, quando as taxas do ensino superior se mantém ou aumentam, visto que, a evasão escolar traz prejuízos. Dentre esses malefícios, estão dano severos a saúde financeira da instituição, além dos danos causados a sociedade.

1.2 DELIMITAÇÃO

Para que não haja danos a instituição e a sociedade, podem ser tomados diversos caminhos a fim de diminuir a taxa de evasão. Estes caminhos, vão desde o entendimento do porquê a evasão acontece, passando pelo momento que ela está acontecendo e até medidas que podem ser tomadas para evitar que a evasão acontece, ou seja, quando o aluno já está decidido a evadir, também existe o cenário de após a evasão. Dito isso, aborda-se como chegamos a uma delimitação de tema.

Relembrando que este Trabalho de Conclusão de Curso (TCC) é situado dentro do curso de Análise e Desenvolvimento de Sistemas (ADS) que busca ensinar o processo de análise e de desenvolvimento de softwares afim de resolver um determinado problema. Assim, o presente projeto buscou analisar e desenvolver um software para se aproximar da solução para a evasão escolar, o que limita ele a duas situações, tentar diminuir as evasões escolares ou a ajudar a retomar alunos evadidos. A partir das duas situações, existem três cenários, eles são em ordem cronológica, antes do aluno decidir a evadir, quando o aluno decide evadir e após a evasão do aluno.

O cenário mais promissor para se desenvolver o software, é antes do aluno decidir a evadir, isso se dá porque o aluno ainda não evadiu, logo, ainda não foi causado nenhum impacto aos pontos, social, financeiro, político e legislativo. Já enquanto o estudante decidiu a evadir, tem os mesmos benefícios do cenário anterior, todavia, com a desvantagem de que o aluno está decidido a evadir, o que tentar convence-ló do contrário, torna-se um processo mais difícil. Por

fim, o último cenário, é o depois da evasão e este, afeta os 4 pontos citados anteriormente. Visto tudo isso, está claro que o melhor cenário para se desenvolver um software, é para antes do aluno evadir.

1.3 PROBLEMATIZAÇÃO

Posto isso, surge a seguinte dúvida: de que forma o ramo de desenvolvimento de software aplicado ao cenário anterior a evasão do discente pode tentar predizer quais alunos estão sujeitos a evadir?

1.4 HIPÓTESES

Os softwares são capazes de calcular e processar diversas tarefas humanas, com a vantagem de fazer isso muito mais rápido e poder fazer várias tarefas ao mesmo tempo a depender do *hardware* utilizado pelo *software*. O que significa que alguma tarefa que levaria horas para um ser humano fazer, os softwares são capazes de fazer isso em alguns segundos. Quando aplicado ao problema em questão, somente o software não é capaz de predizer quais alunos estão sujeitos a evadir, já que ele não possui a capacidade raciocinar igual aos humanos, que por sua vez são capazes de analisar dados e encontrar padrões e partir disso concluir tendências sobre novos dados.

Por não conseguir raciocinar, o ser humano desenvolveu as Inteligências Artificiais (IA) e vem as melhorando com passar do tempo. As IAs buscam processar dados afim de executar tarefas que os humanos conseguem fazer, como raciocinar, aprender e evoluir [ISO and IEC and IEEE \(2017, p. 29\)](#). Com elas, é possível desenvolver um software complementar que é capaz de analisar os dados dos alunos ao longo do tempo, encontrar padrões sobre esses dados, tal como evasões escolares, por fim, aplicar em novos alunos e tentar predizer possíveis evasões escolares.

Portanto, um sistema que possua uma IA que tem a capacidade de encontrar padrões e que é capaz de analisar os dados dos alunos ao longo do tempo, é competente para predizer possíveis evasões escolares.

1.5 OBJETIVO GERAL

Desenvolver um sistema que utiliza IA, onde ela tem a capacidade de encontrar padrões e faz o acompanhamento dos alunos da instituição de ensino para tentar predizer possíveis evasões escolares.

1.6 OBJETIVOS ESPECÍFICOS

1. Fazer o levantamento de requisitos;

2. Modelar o software;
3. Programar o software;
4. Coletar dados do alunos evadidos;
5. Treinar a IA com os dados coletados;

1.7 JUSTIFICATIVA

Este trabalho se justifica no meio social, visto que o dano causado a sociedade a longo prazo pode ser grande, devido a um aluno não estar estudando afim de que seu conhecimento modernize e resolva problemas da sociedade, tais como, ambiental, estrutural, financeiro e muitos outros.

Ademais, o trabalho também se justifica por meio da instituição, dado que a evasão do aluno da instituição diminui a arrecadação da mesma, o que pode impedir que a instituição melhore, dificultando a oferta de serviços melhores. Também, a evasão do aluno pode causar índices ruins para a reputação da instituição, criando visões distorcidas da mesma.

Também, há motivos políticos em função de que o Brasil não está nas melhores posições do ranking de educação mundial, atualmente situado na posição 53º PISA... (2019), o que do ponto de vista dos outros países, pode gerar a ideia de que o Brasil não é um país para se estudar ou para investir no meio acadêmico.

Inclusive, encontra-se justificativa no âmbito legislativo, pois, a depender dos resultados que o software entregar, ele poderá servir como um apoio para que os índices de evasão escolar diminuem e faça com que a lei do direito a educação, esteja a um passo mais próxima de ser alcançada.

Outro ponto, é o meio científico, já que este trabalho, pode servir para contribuir com o conhecimento científico, descartando ou aprimorando essa possibilidade como um possível apoio na resolução da evasão escolar nas instituições, permitindo que a comunidade científica chegue em uma resposta mais rápida para se resolver a evasão escolar.

Ainda por cima, existem razões pessoais para que esse trabalho seja desenvolvido, mais especificadamente, experiência pessoal. Uma vez que interrompi minha graduação por problemas que foram se acumulando e por decisões erradas. Isso poderia ser evitado, caso um trabalho semelhante a este existisse, onde ele poderia ter me ajudado a não evadir.

Por fim, também há uma motivação em ajudar pessoas que estão caminhando para a decisão de evadir, pois, interromper os estudos nunca é o caminho, pois a educação liberta você e permite que cresça em diversos aspectos da vida.

1.8 METODOLOGIA

1.8.1 *Natureza*

Como citado anteriormente, o curso de ADS busca ensinar a analisar e desenvolver sistemas afim de resolver um determinado problema ou fazer com que o problema seja menos danoso, mais rápido ou qualquer outra característica que possa ser melhorada. Logo, naturalmente este TCC tem a sua natureza como Aplicada, pois busca desenvolver conhecimento para solucionar um problema.

1.8.2 *Abordagem*

A abordagem da metologia deste trabalho tem caráter quantitativo e qualitativo, pois o problema de pesquisa indica que teremos que obter opiniões e dados para a conclusão do projeto, opiniões que tem característica qualitativa e os dados que tem característica quantitativa.

1.8.3 *Objetivos*

Para os objetivos da pesquisa, esta pesquisa tem aspecto descritiva, dado que baseamos parte do trabalho em diversos outros autores que produziram trabalhos semelhantes e também tem aspecto exploratório, em razão de levantarmos dados que não estão documentados em nenhum meio com o fim de utilizá-los para diversos fins.

1.8.4 *Procedimentos*

Quanto a parte dos procedimentos utilizados, basicamente foi um para cada objetivo específico do projeto, sendo o levantamento voltado para o primeiro objetivo específico, a pesquisa bibliográfica para o segundo objetivo específico, a pesquisa documental para o quarto objetivo específico e a pesquisa-ação para o quinto objetivo específico.

1.8.5 *Método*

Como já pode ter percebido, este trabalho tem como método o hipotético dedutivo, já que a partir da hipótese demonstrado na seção **HIPÓTESES** vamos colocar a hipótese à prova para ver se ela se sustenta.

1.8.6 *Procedimentos*

Devido a natureza do projeto em questão e para obter uma maior qualidade e precisão, o projeto necessita de dois tipos de procedimentos para a obtenção de dados. O primeiro é o procedimento bibliográfico e o segundo é o de levantamento de campo. Cada um deles tem propósitos diferentes para o *software* e irão atuar em etapas diferentes.

O primeiro procedimento utilizado, foi a pesquisa bibliográfica. Ela serviu para a escolha de quais dados dos alunos serão utilizados no *software*. Decidir os dados apenas utilizando a cabeça, tornaria o *software* mais impreciso e o projeto menos verídico.

Já o segundo procedimento, foi o de levantamento, esse se fez necessário, já que ele permite uma maior assertividade da taxa de evasão, visto que ele retira dados do mundo real. A finalidade deste procedimento é fazer o levantamento de dados dos alunos evadidos para que seja feito o treinamento da inteligência artificial do *software*. Novamente, pode-se alimentar aleatoriamente a IA com dados dos alunos, mas isso, poderia ocasionar em menor taxa de precisão da IA com relação a taxa de evasão.

1.8.7 Conceitos

Para entender o desenvolvimento do *software* e do projeto, é preciso entender alguns conceitos. Eles vão desde abstração do mundo real para *software* até o armazenamento dos dados utilizados.

Existem diversos paradigmas de programação, entre eles, está a Programação Orientada a Objetos (POO), este paradigma permite modelar o mundo real para o mundo digital, por isso, esse é um dos motivos que escolhemos utiliza-lo no nosso projeto. O [Noleto \(2021\)](#) diz sobre a POO, “O paradigma da POO (Programação Orientada a Objetos) é um modelo de análise, projeto e programação baseado na aproximação entre o mundo real e o mundo virtual, através da criação e interação entre objetos, atributos, códigos, métodos, entre outros.”.

Já para modelar o *software* para que seja legível para os profissionais da área, foi utilizada a *Unified Modeling Language* (UML), ela permite modelar o *software* através de diversos recursos. [Bock, Rumbaugh e Jacobson \(2012\)](#) define a UML da seguinte forma:

“A UML (*Unified Modeling Language*) é uma linguagem-padrão para a elaboração da estrutura de projetos de *software*. Ela poderá ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de *software*.”.

Antes de entender onde foi armazenado os dados do *software*, é preciso entender como armazenar dados, o Banco de Dados (BD) atua para resolver esse problema, assim, em conforme com [Date \(2003, p. 46\)](#), “O banco de dados, por si só, pode ser considerado como o equivalente eletrônico de um armário de arquivamento; ou seja, ele é um repositório ou recipiente para uma coleção de arquivos de dados computadorizados.”.

Por fim, o último conceito a ser exposto, são os Sistemas Gerenciadores de Bancos de Dados (SGBD), com ele é possível fazer a gestão de bancos de dados através de *software*, assim como diz [Juliano \(2014\)](#):

“O Sistema de Gestão de Bases de Dados (SGBD) ou *Data Base Management System* (DBMS) é *software* utilizado para gerir Bases de Dados, permitindo criar base de dados, modificar Base de dados, eliminar bases de dados, inserir dados na Base de Dados e Eliminar dados da base de dados.”.

1.8.8 *Tecnologias*

Para o desenvolvimento do *software*, foi utilizados diversas tecnologias cada uma com um propósito e uma área de atuação. Foi utilizado a linguagem de marcação, de estilização, de programação e de consulta, isso se dá pela essência do *software*, já que ele é da arquitetura Cliente-Servidor.

Para a criação do site, foram utilizadas três linguagens, o *HyperText Markup Language* (HTML), ou em português, Linguagem de Marcação de HiperTexto, o *Cascading Style Sheets* (CSS), em português, Folha de Estilo em Cascata e o JavaScript. O HTML serve para a estruturação do site, como afirma o [HTML... \(2016\)](#):

“HTML é a linguagem para descrever a estrutura das páginas da Web. [...]. Com HTML, os autores descrevem a estrutura das páginas usando marcação. Os elementos da linguagem rotulam partes do conteúdo como “parágrafo”, “lista”, “tabela” e assim por diante.”

Já o CSS serve para a estilização do site, isso em harmonia com a [HTML... \(2016\)](#), que diz, “CSS é a linguagem para descrever a apresentação de páginas da Web, incluindo cores, layout e fontes. Permite adaptar a apresentação a diferentes tipos de dispositivos, como telas grandes, telas pequenas ou impressoras.”.

Para completar a tríade da *web*, tem o JavaScript, ela é uma linguagem de programação que atua no lado do cliente (ou em inglês, *Client-Side*). Ela trabalha na parte de “animação” da página, dando vida a ela, com ela, é possível criar funções, estruturas dinâmicas e outros.

Para programar toda a lógica do software, e também aprender de forma fácil e desenvolver rápido, optou-se pela linguagem Python, veja o que diz [O... \(2023d\)](#):

“Python é uma linguagem fácil de aprender e poderosa. Ela tem estruturas de dados de alto nível eficientes e uma abordagem simples mas efetiva de programação orientada a objetos. A elegância de sintaxe e a tipagem dinâmica de Python aliadas com sua natureza interpretativa, o fazem a linguagem ideal para programas e desenvolvimento de aplicações rápidas em diversas áreas e na maioria das plataformas.”

A linguagem padrão para consulta em banco de dados relacionais é a *Structured Query Language* (SQL), ou em português, Linguagem de Consulta Estruturada. Observe a explicação do [O... \(2023c\)](#):

“A Linguagem de consulta estruturada (SQL) é uma linguagem de programação para armazenar e processar informações em um banco de dados relacional. Um banco de dados relacional armazena informações em formato tabular, com linhas e colunas representando diferentes atributos de dados e as várias relações entre os valores dos dados. Você pode usar instruções SQL para armazenar, atualizar, remover, pesquisar e recuperar informações do banco de dados. Também pode usar SQL para manter e otimizar a performance do banco de dados.”

O banco de dados usado no projeto, foi escolhido baseado nas necessidades do projeto, assim, chegou-se a escolha de que o BD PostgreSQL é o mais adequado para o projeto, o [Sobre](#)

(2023) explica, “PostgreSQL é um poderoso sistema de banco de dados relacional de objeto de código aberto que usa e estende a linguagem SQL combinada com muitos recursos que armazenam e dimensionam com segurança as cargas de trabalho de dados mais complicadas.”.

1.8.9 Ferramentas

Para fazer escolha de quais dados dos alunos através de pesquisa bibliográfica, optou-se por usar sites que disponibilizam periódicos para leitura. Os sites que utilizamos foram o Scientific Electronic Library Online (SciELO, ou em português, Biblioteca Eletrônica Científica *Online*). Já o outro, Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), é um site governamental do Ministério da Educação (MEC), ele disponibiliza diversos periódicos em português e outros assinados com editoras internacionais.

Já para o levantamento de dados, será utilizado o método de questionários *online*, eles são instrumentos de coleta de dados online que são preenchidos pelos informantes. Foi utilizado o questionário *online* do Google, o Google *Forms*, é uma ferramenta que permite a criação, personalização e disponibilização de questionários *online*.

A Integrated Development Environment (IDE) que foi utilizada para a programação do software foi o Visual Studio Code (VSC), há diversos motivos para essa escolha que serão abordados mais a frente. Veja a definição da Icons... (2023) sobre o VSC:

“Visual Studio Code é um editor de código-fonte leve, mas poderoso, executado em sua área de trabalho e disponível para Windows, macOS e Linux. Ele vem com suporte integrado para JavaScript, TypeScript e Node.js e possui um rico ecossistema de extensões para outras linguagens e tempos de execução (como C++, C#, Java, Python, PHP, Go, .NET).”

Já para a criação dos *wireframes*, *mockups* e protótipos, foi utilizado o programa Figma, um *software* que disponibiliza ferramentas para a construção artes. Seguindo a explicação da Villain (2023):

“O Figma é uma plataforma colaborativa para construção de *design de interfaces* e protótipos, pertencente a empresa Figma, Inc., lançada em 2016 por Dylan Field e Evan Wallace, com o objetivo de criar uma ferramenta gratuita que trouxesse colaboração entre pessoas e times, permitindo criar um produto para as mais diversas plataformas, mantendo a acessibilidade do sistema.”.

O SGBD que foi utilizado para o gerenciamento do nosso banco de dados é o PgAdmin, ele traz diversas ferramentas interessantes que auxilia na criação do *software*. Observe a definição do Perguntas... (2023a), “pgAdmin é uma ferramenta de gerenciamento para PostgreSQL e bancos de dados relacionais derivados, como o EDB Advanced Server do EnterpriseDB . Ele pode ser executado como um aplicativo web ou desktop.”.

1.9 ESTRUTURA DA MONOGRAFIA

Além do capítulo em questão, sendo ele a **INTRODUÇÃO**, que demonstra a contextualização, delimitação, justificativa, objetivos e metodologia. Há também os capítulos, **INTRODUÇÃO, INTRODUÇÃO, INTRODUÇÃO, INTRODUÇÃO, INTRODUÇÃO, INTRODUÇÃO, INTRODUÇÃO, INTRODUÇÃO**.

O capítulo **INTRODUÇÃO** manifesta-se os passos necessários para que o requisitos do software fossem decididos. Dentro deste é possível observar o levantamento de requisitos e a especificação dos requisitos.

No capítulo **INTRODUÇÃO** deste TCC demonstra a arquitetura do *software*, a arquitetura de um *software* é o segundo passo na construção de um *software*. Ele define a necessidades do *software* a um nível macro, sendo elas derivadas a partir do requisitos. Por fim, é analisado as necessidades e definido a arquitetura do sistema, as ferramentas, as tecnologias e outros.

No capítulo **INTRODUÇÃO**, é mostrado o passo a passo para mostrar a estrutura que guarda os dados. Nesse capítulo é demonstrado os modelos de dados, os *scripts* de cada tabela e outros. Entre os modelos, modelo conceitual, modelo lógico e o modelo físico.

Enquanto a engenharia de dados modela os dados, o capítulo **INTRODUÇÃO** modela o *software* em diversas visões. Com essas visões, é possível ter uma ideia mais de como o *software* irá se comportar, quais dados ele irá trabalhar, como ele se relaciona com outros dados e assim por diante. Essas visões são comumente chamadas de diagramas, as mais comuns são, o diagrama de classes, o diagrama de casos de uso, o diagrama de sequência e o diagrama de implantação.

O capítulo **INTRODUÇÃO** dispõe quais cores foram utilizadas no sistema, quais fontes foram usadas, como se chegou ao nome do sistema, a logo do projeto e as telas do sistema.

O capítulo **INTRODUÇÃO** apresenta qual modelo de IA foi utilizado para calcular a taxa de evasão, bem como pode ser visto o cálculo deste modelo, o código do modelo, as bibliotecas utilizadas no código do modelo e assim por diante.

O capítulo **INTRODUÇÃO** demonstra quais foram os resultados que obtive com o modelo de IA, lá você pode observar a correlação entre variáveis utilizadas e os resultados obtidos do modelo.

Por fim, mas não menos importante, há o capítulo **INTRODUÇÃO**, neste capítulo você pode observar que conclusão que obtive do projeto, juntamente com sugestões do que pode ser feito quando se tomar este trabalho como base.

Parte II

Desenvolvimento

2 ENGENHARIA DE REQUISITOS

A engenharia de requisitos é ponto de partida para todo software, a parte dele se entende o que é necessário para o software Vazquez e Simões (2016). Ela algo que é praticado dentro da disciplina da engenharia de software afim da obtenção, documentação e manutenir deles Vazquez e Simões (2016). Após estes requisitos serem definidos, eles são utilizados para atingir as regras de negócio Vazquez e Simões (2016).

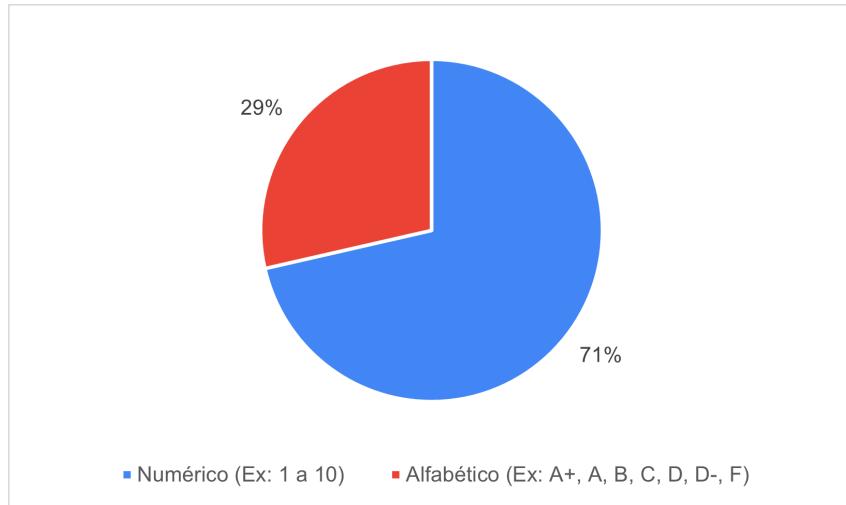
2.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é um sub-passo da engenharia de requisitos, que determina através de técnica entender o que é o negócio e os requisitos dele. Ela se trata de um processo de aquisição de informações através de técnicas e ferramentas afim de compreender de que forma o software irá funcionar e quais requisitos são necessário para que ele funcione Vazquez e Simões (2016). Esse levantamento é analisado, documentado, revisado afim de descobrir erros, omissões e redundâncias Vazquez e Simões (2016).

Uma das técnicas utilizadas para o levantamento de requisitos, foi através de um questionário. O questionário foi submetido para pessoas que fazem parte do corpo operacional das instituições, ele se concentra na cidade de Santa Fé do Sul e cidades próximas a ela. Realizado entre, 23/09/2023 a 31/10/2023 e ele obteve um total 7 respostas.

Outra técnica utilizada, foi por meio de reuniões, realizadas regularmente para definir quais requisitos são essenciais para a primeira implementação. Posto essas informações, abaixo pode ser visto o resultado das perguntas. Na figura 1, o gráfico questiona aos entrevistados sobre qual tipo de nota é atribuído ao aluno:

Figura 1 – Gráfico referente a pergunta “Qual o tipo de nota é atribuído aos alunos?”

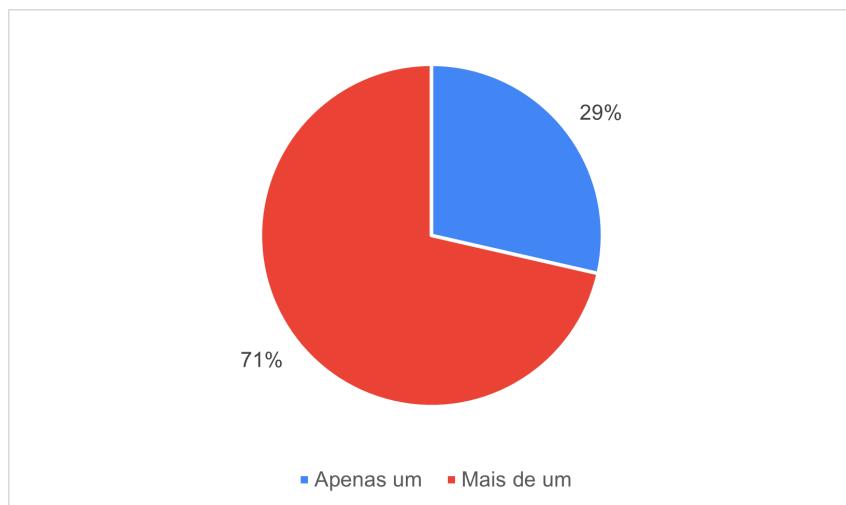


Fonte: elaborado pelo autor (2023).

A partir da figura 1, o gráfico demonstra que 71% dos entrevistados utilizam o tipo 'Numérico' e 29% utiliza o tipo 'Alfabético', o que significa que a maior parte dos entrevistados utilizam o tipo 'Numérico'.

Por sua vez, a figura 2, o gráfico interroga quantos professores podem lecionar em uma mesma matéria:

Figura 2 – Gráfico referente a pergunta “Quantos professores podem lecionar em uma mesma matéria?”



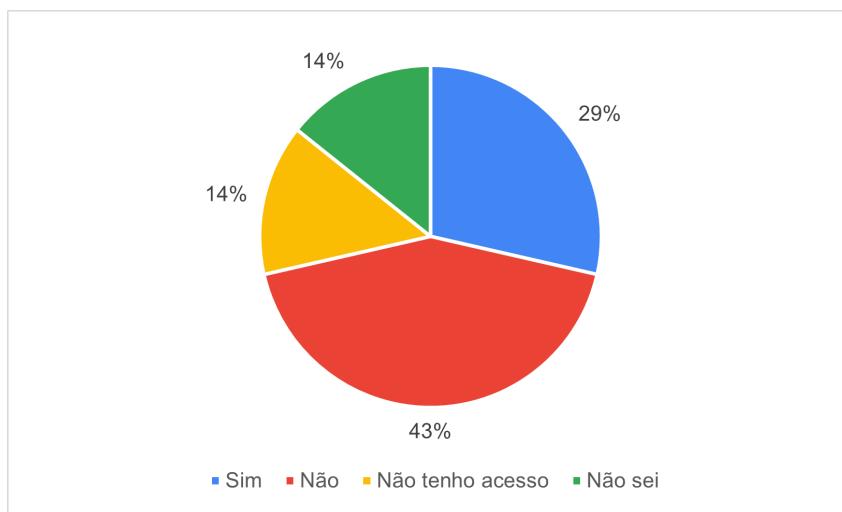
Fonte: elaborado pelo autor (2023).

Na figura 2, o gráfico apresenta que em 71% das instituições permitem que mais de um

professor possa lecionar em uma matéria e que 29% das instituições não permitem que mais de um professor lecione em uma matéria, ou seja, a maioria das instituições permitem que mais de um professor lecione em uma matéria.

Na figura 3, o gráfico indaga se o software é capaz de analisar a evasão escolar na sua instituição:

Figura 3 – Gráfico referente a pergunta “O software acadêmico consegue analisar a evasão escolar na sua escola?”



Fonte: elaborado pelo autor (2023).

Na figura 3, o gráfico revela que 43% das respostas dizem que os software das instituições não conseguem analisar a evasão escolar da instituição, 29% diz que conseguem analisar, 14% diz que não tem acesso a essa informação e que 14% não sabe, assim dizendo, a maioria dos software acadêmicos implementados não conseguem analisar a evasão escolar.

No quadro 1, revela as respostas referente ao questionamento de qual a taxa de evasão escolar:

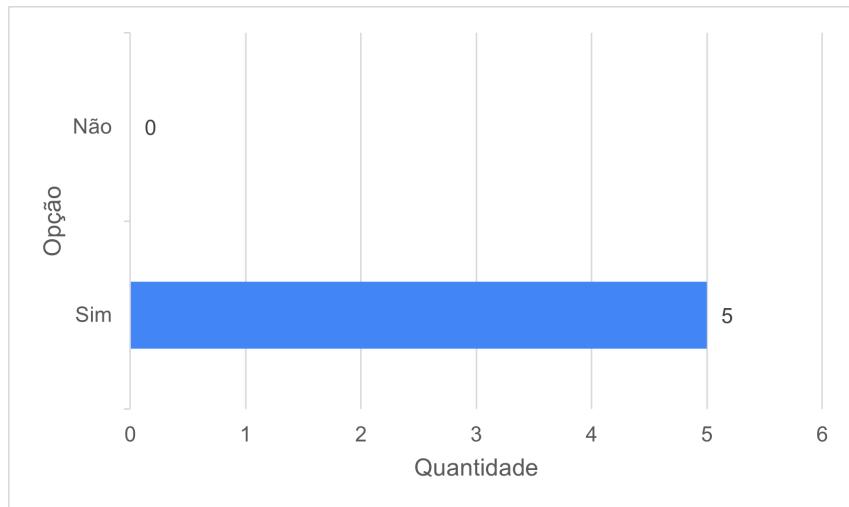
Quadro 1 – Quadro referente a pergunta ’Qual a taxa de evasão escolar?’

#	Resposta
1	"Depende do curso, mas infelizmente, no modular algo em torno do 20%"

Fonte: elaborado pelo autor (2023).

Na figura 4, o gráfico pergunta se você usaria um software de previsão de evasão escolar:

Figura 4 – Gráfico referente a pergunta “Você usaria um software de previsão de evasão escolar?”

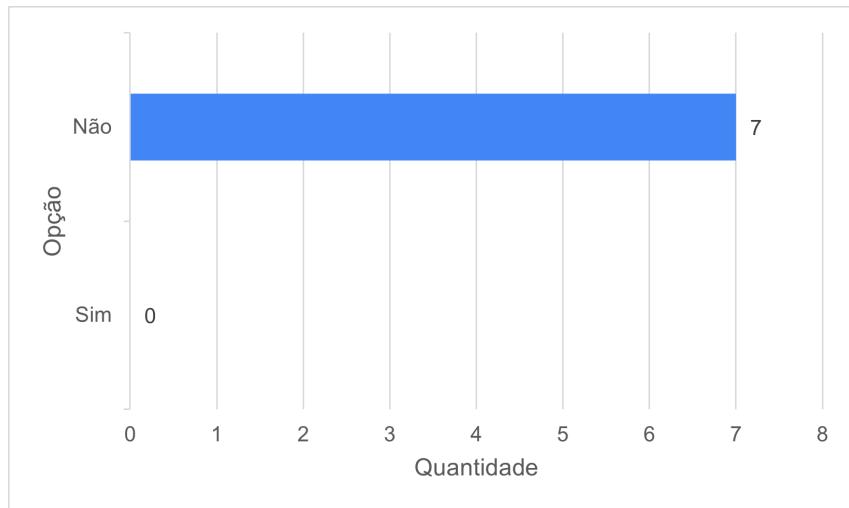


Fonte: elaborado pelo autor (2023).

Na figura 4 expressa que todas as respostas da questão, afirmaram que usariam um software de previsão de evasão escolar.

Na figura 5, o gráfico interpela se você conhece algum software de previsão de evasão escolar:

Figura 5 – Gráfico referente a pergunta “Conhece algum software de previsão da evasão escolar?”

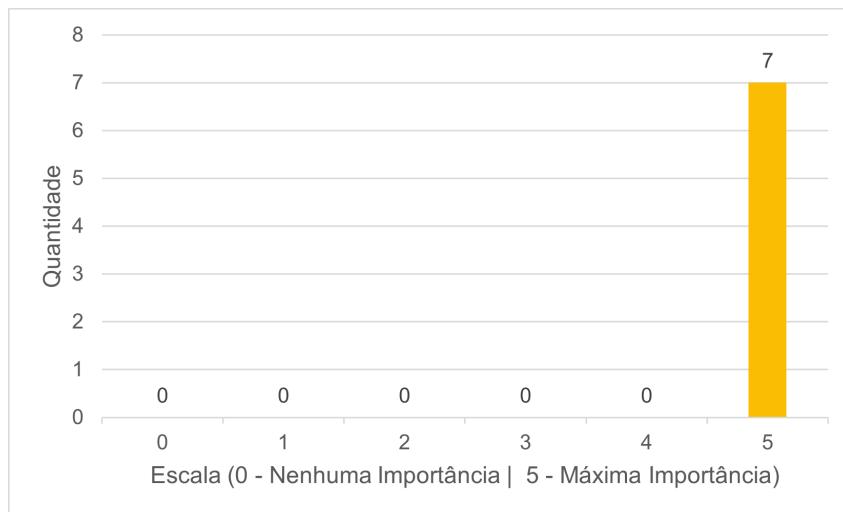


Fonte: elaborado pelo autor (2023).

Na figura 5 demonstra que todas as respostas da pergunta, afirmaram que não conhecem um software de previsão de evasão escolar.

Na figura 6, o gráfico inquiri qual nível de importância em uma escala de 0 a 5 você dá para conhecer a taxa de evasão da instituição::

Figura 6 – Gráfico referente a pergunta “Qual o nível de importância de 0 a 5 você dá para conhecer a taxa de evasão da instituição de ensino?”

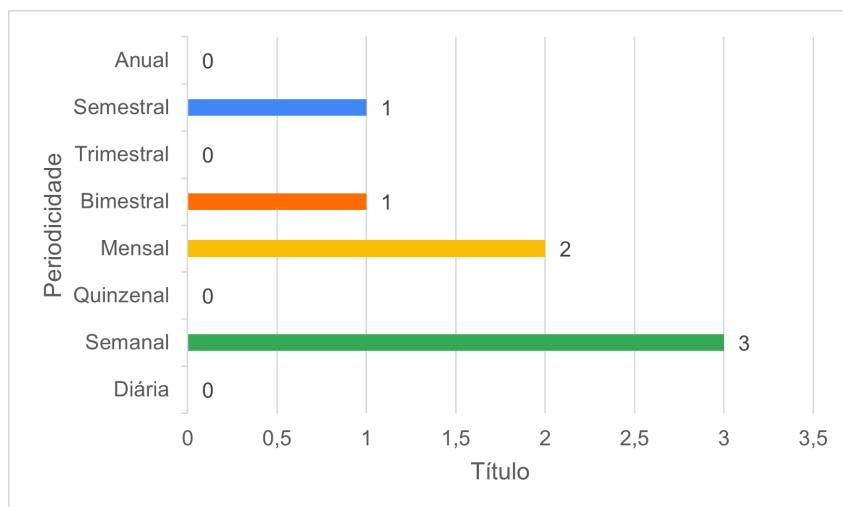


Fonte: elaborado pelo autor (2023).

A figura 6 transmite que todas as respostas do questionamento, proveram a máxima importância em saber a taxa de evasão escolar.

Na figura 7, o gráfico questiona com que frequência é relevante atualizar as taxas de evasão:

Figura 7 – Gráfico referente a pergunta “Com que frequência você considera relevante atualizar as taxas de evasão?”

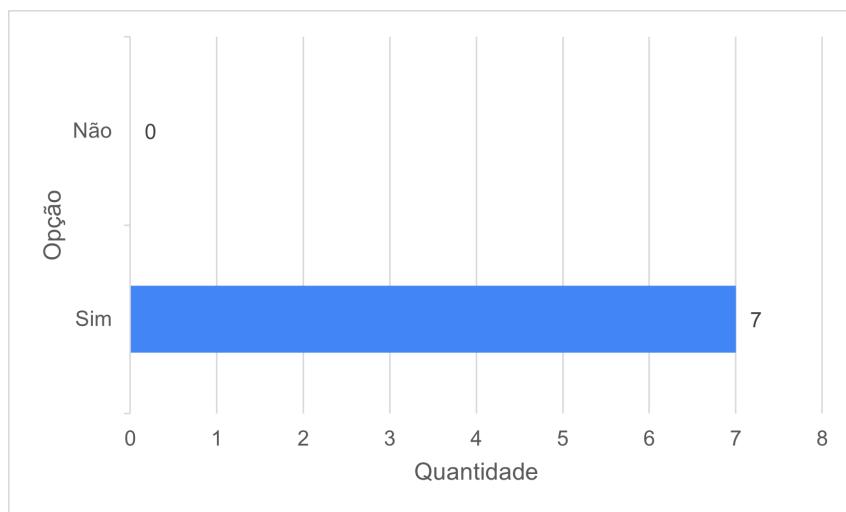


Fonte: elaborado pelo autor (2023).

A figura 7 exprimiu que 3 das 7 respostas afirmam que a frequência deve ser semanal, que 2 de 7 deve ser mensal, que 1 de 7 deve ser bimestral, que 1 de 7 deve ser semestral e o restante nenhum foi selecionado.

Na figura 8, o gráfico interroga se é relevante que um software de evasão escolar notifique quando aluno está com altas chances de evadir:

Figura 8 – Gráfico referente a pergunta “Você acha relevante que um software de evasão escolar notifique quando um aluno está com altas chances de evadir?”

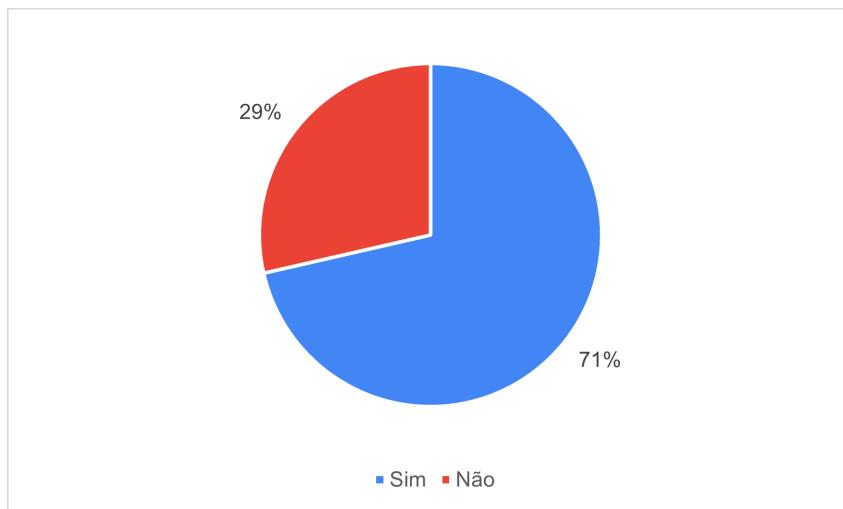


Fonte: elaborado pelo autor (2023).

A figura 8 denota que todas as respostas do questionamento, afirmaram que o software deve notificar quando um aluno está com altas chances de evadir.

Na figura 9, o gráfico indaga se você acha interessante que seja possível criar grupos de alunos com mais chances de evadir dentre todos os alunos cadastrados:

Figura 9 – Gráfico referente a pergunta “Você acha interessante que seja possível criar grupos de alunos com mais chances de evadir dentre todos os alunos cadastrados?”



Fonte: elaborado pelo autor (2023).

A figura 9 exibe que 71% das respostas, afirma que acha interessante ser possível criar grupos com alunos com mais chances de evadir, já para 29%, acham que não é interessante.

2.2 ESPECIFICAÇÃO DOS REQUISITOS

Observação: A maioria dos sistemas permite o conceito chamado *Create Read Update Delete* (CRUD) de dados, é uma das coisas mais básicas quando se trata de softwares. Ele se tratar de poder cadastrar, mostrar, atualizar e excluir dados de determinar categoria. Então a maioria ou todo requisitos vão estar resumidos utilizando este conceito para evitar repetição.

2.2.1 Descrição dos requisitos funcionais

- Pessoa
 - O sistema irá permitir o CRUD de pessoas além do ‘Entrar’ e ‘Sair’, sendo que as características utilizadas pelo software são o *e-mail* e a senha. Isso será utilizado para fazer a autenticação no sistema
- Grupo
 - O sistema irá permitir o CRUD de grupos, sendo que as características utilizadas pelo software é apenas o nome. Isso será utilizado para fazer a agrupar alunos que tem mais chances de evadir.
- Aluno

- O sistema irá permitir o CRUD de alunos, sendo que as características utilizadas pelo software são o registro da matrícula e a taxa de evasão.
- Disciplina
 - O sistema irá permitir o CRUD de disciplinas, sendo que as características utilizadas pelo software é o nome e a sigla. Isso será utilizado para saber quais disciplinas os alunos faltaram, além de outras coisas.
- Aula
 - O sistema irá permitir o CRUD de aulas, sendo que as características utilizadas pelo software é apenas a data. Isso será utilizado para saber quais aulas os alunos faltaram, além de outras coisas.
- Semestre
 - O sistema irá permitir o CRUD de semestres, sendo que as características utilizadas pelo software é apenas o valor. Isso será utilizado para saber diferenciar alunos que repetiram disciplinas.
- Ano
 - O sistema irá permitir o CRUD de anos, sendo que as características utilizadas pelo software é apenas o valor. Isso será utilizado para saber diferenciar alunos que repetiram disciplinas.
- Nota
 - O sistema irá permitir o CRUD de notas, sendo que as características utilizadas pelo software é apenas o valor. Isso será utilizado para saber diferenciar as notas dos alunos.

3 ARQUITETURA DE SOFTWARE

Todo *software* possui uma arquitetura, um projeto que demonstra como os componentes e conectores se comportam (PAUTASSO, 2021). A arquitetura de *software* é tomar as principais decisões de projeto para o sistema, tais como, qual estilo arquitetural será utilizado, qual padrão arquitetural será utilizado, quais ferramentas serão utilizadas, quais tecnologias vão ser utilizadas e etc (PAUTASSO, 2021). Isso torna a construção do *software* muito mais clara e porque ele vai atuar naquele formato (PAUTASSO, 2021).

Assim, você entende como vai fazer e por quê vai fazer daquela forma (PAUTASSO, 2021). Ela parte da engenharia de requisitos, tomando suas decisões a fim de fazer as escolhas mais adequadas para o sistema de acordo com seus requisitos.

3.1 ESTILO ARQUITETURAL

Um estilo arquitetural se trata de como estruturar seu *software* em um nível macro, onde você decide como será a comunicação entre um lado (normalmente o cliente) e o outro (normalmente a aplicação) (RICHARDS, 2015). A depender do estilo arquitetural que você utilizar, eles podem ser mais escaláveis, mais caros, mais adaptáveis, mais eficiente e outras coisas mais (RICHARDS, 2015). Além disso, necessidades externas ao *software* também contam, como experiência da equipe, tempo para ficar pronto, o quanto você pode investir e assim vai.

Cada projeto exige necessidades diferentes, deste jeito, você utilizar um estilo arquitetural que não atenda suas necessidades faz com que você elimine a possibilidade de tirar o máximo proveito do projeto. Existem diversos estilos arquiteturais, entre eles, *Microkernel*, *Event-driven*, *Space-based*, *Layered* e *Space-based* (RICHARDS, 2015, p. 66). Na tabela 2 você pode ver uma comparação entre 10 estilos arquiteturais.

Quadro 2 – Comparação de estilos arquiteturais

	Agilidade	Facilidade de implantação	Testabilidade	Performance	Escalabilidade	Facilidade de desenvolvimento
Layered	Baixa	Baixa	Alta	Baixa	Baixa	Alta
Event-Driven	Alta	Alta	Baixa	Alta	Alta	Baixa
Microkernel	Alta	Alta	Alta	Alta	Baixa	Baixa
Microservices	Alta	Alta	Alta	Baixa	Alta	Alta
Spaced-Based	Alta	Alta	Baixa	Alta	Alta	Baixa
Client-Server	Baixa	Alta	Alta	Alta	Alta	Alta
Master-Slave	Baixa	Alta	Alta	Alta	Alta	Alta
Pipe-Filter	Alta	Alta	Alta	Baixa	Alta	Alta
Todas as informações foram retirada do (DHADUK, 2020)						

Fonte: elaborado pelo autor (2023).

Baseado nesta tabela, o estilo arquitetural mais adequado ao projeto, é o *Client-Server*, em português, Cliente-Servidor. Esse estilo é basicamente dividido em duas partes, o Cliente, que é quem acessa o serviço que o servidor oferece, a maioria dos serviços disponibilizados pela web, utilizam a arquitetura Cliente-Servidor, devido a facilidade de implantação e desenvolvimento [Sommerville \(2011, p. 113\)](#). A outra parte, é o servidor, ele hospeda um serviço dentro dele, normalmente esse serviço se diz respeito a uma aplicação completa, ela provê uma determinada solução [Sommerville \(2011, p. 113\)](#).

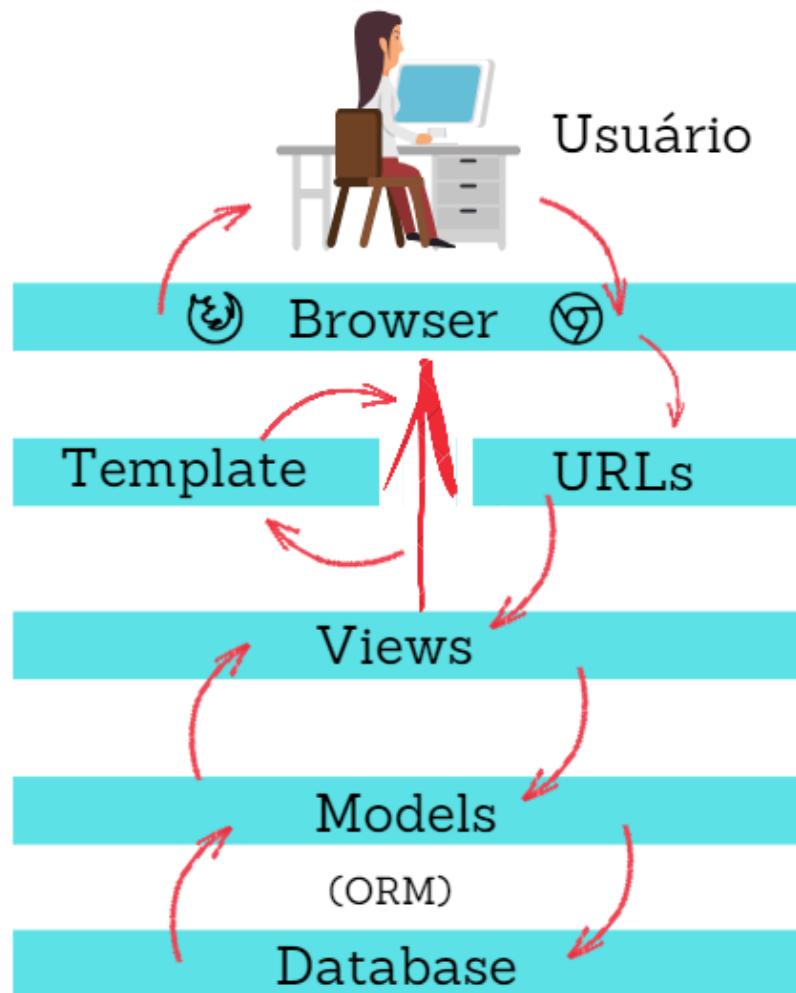
3.2 PADRÃO ARQUITETURAL

O *Django* usa uma padrão arquitetural muito parecido com o padrão *Model-View-Controller* (MVC), ele se chama *Model-Template-View* (MTV). No padrão MVC, a *Model* é entendido como um modelo para os dados e a lógica de processamento; A *View* é vista em como você apresenta os dados, além de poder possuir algumas lógicas de processamento; Já a *Controller* é quem controla o fluxo de dados entre a *View* e a *Model* ([PRESSMAN; MAXIM, 2021](#)).

O *Django* entende a sua arquitetura um pouco diferente, por isso, ele muda a sigla para MTV. Para o *Django*, a *Model* do MVC continua com a mesma função da *Model* do MTV, definir um modelo para um grupo de objetos. Já o *Template* do MTV assume a função da *View* do MVC, ela descreve como os dados serão mostrados ([PERGUNTAS..., 2023b](#)). Por fim, a *View* do MTV, é responsável por quais dados serão mostrados e não como serão mostrados ([PERGUNTAS..., 2023b](#)).

Pelo *Django* utilizar MTV em sua estrutura, logicamente temos que utilizar ele em nosso projeto, por estarmos utilizando o *Django*. Logo abaixo, na imagem 10 você pode ver a arquitetura MTV em forma de pilha.

Figura 10 – Arquitetura MTV em forma de pilha



Fonte: [Pinheiro \(2020\)](#).

3.3 TECNOLOGIAS

Nesta seção, aborda-se as tecnologias do sistema e de outros processos para o projeto. As tecnologias facilitam, ajuda e proporciona novas 'portas' para o projeto, além de trazer diversos outros benefícios. Observe que utiliza-se diversas tecnologias diferentes para diferentes áreas do projeto, como o desenvolvimento, a documentação, o levantamento de dados, a apresentação e assim por diante.

3.3.1 HTML5

Com a necessidade do compartilhamento automatizado de informações entre cientistas em universidades e institutos em todo o mundo, Tim Berners-Lee inventou a *World Wide Web* (WWW),

o “lugar” onde você publica seu site ([UMA... , 2023](#)). Mas para a publicação do site, era necessário uma linguagem universal para definir a estrutura desses sites ([HTML/TREINAMENTO/O... , 2011](#)). Então, em 1993, Tim Berners-Lee lançou a primeira versão da linguagem de publicação na WWW, o HTML. A logo oficial do HTML5 pode ser vista abaixo na figura 11.

Figura 11 – Logo Oficial do HTML5



Fonte: [ASSUMA... \(2023\)](#).

O HTML é a linguagem que define as estrutura do site, o HTML cuida da estrutura e do conteúdo de um site ([HTML... , 2016](#)). Com o HTML é possível criar tabelas, listas, menus, formulários e outros ([HTML... , 2016](#)). No momento atual, o HTML se encontra na versão 5.3, conhecido como HTML5. Ela é ‘primeira’ linguagem da tríade das linguagens da *web*.

3.3.2 CSS

O CSS normalmente é a segunda linguagem a se aprender da tríade das linguagens da *web*, enquanto que o HTML define a estrutura do site, o CSS define a estilização do site ([CASTRO; HYSLOP, 2013](#), p. 179). O CSS permite alterar cores tamanhos, estilos, fontes e outros ([DUCKETT, 2011](#)). A logo não oficial do CSS pode ser vista abaixo na figura 12.

Figura 12 – Logo Não Oficial do CSS3

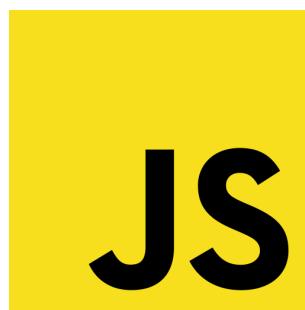
Fonte: [Rudloff \(2016\)](#).

No tempo atual o CSS, juntamente com o HTML, acumula por volta de 52,97% de usuários, essa unificação é feita pois quando se aprende HTML, normalmente e em seguida se aprende CSS, por isso eles são vistos como um na pesquisa do *StackOverflow*([DEVELOPER... , 2023](#)).

3.3.3 JavaScript

E pra completar a tríade, aí é que entra o JavaScript, diferente das outras duas linguagens da tríade, o JavaScript é uma linguagem de programação ([ESPECIFICAÇÃO... , 2023](#)). Inicialmente o JavaScript foi desenvolvido para ser uma linguagem da *web*, permitindo ‘animar’ páginas da *web* e executar comandos lado do cliente ([ESPECIFICAÇÃO... , 2023](#)). Contudo, ele evoluiu tanto ao ponto que ‘escapou’ da *web* e agora consegue atuar em diversos ambientes e em escalas diferentes ([ESPECIFICAÇÃO... , 2023](#)). A logo não oficial do JavaScript pode ser vista abaixo na figura 13.

Figura 13 – Logo Não Oficial do JavaScript

Fonte: [Willians \(2011\)](#).

O JavaScript evoluiu tanto que chegou ao ponto de conquistar o mercado, ele conseguiu alcançar a marca de 63.61% de desenvolvedores que o utilizam ([DEVELOPER... , 2023](#)). O JavaScript em uma página *web* de uma forma simples, dá a vida a página, controlando ações, permissões, restrições, animações e entre outros.

3.3.4 Python

O *Python* é uma linguagem de programação interpretada, interativa, fácil de aprender e orientada a objetos, por ser interpretada e fácil de aprender, ela é muito utilizada para a produção de projetos que tem pouco tempo ([O... , 2023d](#)). Ele consegue atuar em diversas áreas, desde em *softwares desktop* até para *web* e IA, e roda em diversas plataformas como *Windows*, *Linux* e *Mac*. A logo oficial do Python pode ser vista abaixo na figura 14.

Figura 14 – Logo Oficial do Python



Fonte: [O... \(2023\)](#).

Nos dias de hoje, o *Python* é a segunda linguagem mais utilizada, ela é usada por 49.28% dos programadores. Só perdendo o pódio para a linguagem *Javascript*, que é utilizada por 63.61% ([DEVELOPER... , 2023](#)).

A escolha de linguagens de programação para o desenvolvimento *backend* se deu por diversos fatores, tais como, dificuldade de aprendizagem, especialidade da linguagem, atuação da linguagem, velocidade da linguagem e outros. Mas um critério que não pode ser usado para a escolha da linguagem, é a taxa de uso dela, visto que é necessidades de um projeto que definem a escolha de uma linguagem, sejam eles internos e/ou externos ao projeto. Antes de tudo, é preciso entender as necessidades internas e externas do projeto.

O contexto do projeto está sendo desenvolvendo como um TCC e iniciado um ano antes do final do curso. O projeto foi dividido em dois semestres, no primeiro semestre está sendo utilizado para a elaboração da documentação e no segundo semestre ele vai ser utilizado para o desenvolvimento do projeto. O que nos deixa com um tempo curto, o que exige uma linguagem fácil de aprender e que seja rápida para desenvolver projetos. e ele foi desenvolvido em duas partes

Devido o projeto ter o objetivo de prever uma possível evasão escolar utilizando IA, isso cria a necessidade de que a linguagem escolhida tenha bibliotecas para o desenvolvimento da IA e a análise de dados.

Caso a familiaridade de linguagens do orientador e dos integrantes coincide com as linguagens filtradas, elas devem ser levadas em consideradas, já que, elimina o passo de ter que aprender uma linguagem nova. Por fim, a linguagem também precisa suportar o DB PostgreSQL, pois, iremos utilizar ele como o nosso banco de dados.

Em razão dessas necessidades, chega-se a conclusão de que a linguagem de programação Python, é a escolha adequada para o projeto.

3.3.5 *Bootstrap*

O Bootstrap é um *framework CSS*, ele permite que você criar sites responsivos e bonitos de forma muito simples (NOLETO, 2022). O uso do Bootstrap agiliza o desenvolvimento da estilização e animação do site, através de códigos prontos em CSS e JavaScript (NOLETO, 2022). A popularidade do *Framework* tomou conta do seu nicho, que conta com uma taxa de uso de 80,6% dos programadores (GREIF, 2022). A logo oficial do Bootstrap pode ser vista abaixo na figura 15.

Figura 15 – Logo Oficial do Bootstrap



Fonte: Diretrizes... (20??a).

O Bootstrap possibilita um desenvolvimento mais rápido do *design* do site, em razão de que códigos CSS e JavaScript já estão prontos, basta apenas 'referenciar' eles (NOLETO, 2022). Códigos de componentes como cartões, lista suspensas, barras de navegação, cartões, *navbar*, *forms* e assim por diante (NOLETO, 2022).

3.3.6 *Django*

No universo da linguagem Python, existem diversos *framework web* que podem ser utilizados para a *web* em que cada um atende necessidades diferentes. Para que o projeto SPEEIA, o Django foi o escolhido como *framework web*, o Django é um *framework web* que visa produzir *softwares* (INTRODUÇÃO..., 2023). Dentre os *frameworks* disponíveis na linguagem Python, o django é o segundo mais utilizado, com 39% de usuários (DEVELOPER..., 2022). A logo oficial do Django pode ser vista abaixo na figura 16.

Figura 16 – Logo Oficial do Django

Fonte: [Logotipos... \(2023\)](#).

O Django traz diversas vantagens, desde desenvolvimento rápido, ser seguro, versátil e até escalável ([INTRODUÇÃO..., 2023](#)). Ele é projetado para que você se preocupe mais com o conteúdo do site e deixe a maior parte do trabalho do desenvolvimento *web* na mão dele ([INTRODUÇÃO..., 2023](#)). Com isso, o Django consegue lançar aplicativos em intervalos de tempo muito pequenos, se tornando um *framework web* de desenvolvimento rápido ([INTRODUÇÃO..., 2023](#)).

3.3.7 *LATEX*

Em 1985, o matemático Leslie Lamport finalizou a primeira versão do projeto *LATEX*, que recebeu um nome curioso, chamado de *LATEX2.09* ([LAMPORT, 1994](#), p. 1). O *LATEX* é um sistema totalmente gratuito que visa produzir documentos tipográficos com altíssima qualidade, por produzir documentos de altíssima qualidade, ele se tornou referência na comunicação e publicação de documentos científicos ([LATEX..., 2023](#)). Ele faz com que você se preocupe com a parte do conteúdo do documento e deixe para o sistema de composição a parte de design do documento ([LAMPORT, 1994](#), p. 7). A logo oficial do *LATEX* pode ser vista abaixo na figura 17.

Figura 17 – Logo Oficial do *LATEX*Fonte: [Cereda \(2023\)](#).

3.3.8 *Git*

Como na maioria dos projetos em que o produto ou algo que é relacionado ao produto surgem novas versões do mesmo, existe diversos problemas que podem acontecer e desvantagens

em gerenciar esses arquivos manualmente (CHACON; STRAUB, 2014a). Dentre eles, apagar todos/alguns arquivos e querer recuperá-los, duplicar os arquivos do projeto para fazer uma nova versão, guardar todos os arquivos em único local, colaboração em tempo real e entre outros (CHACON; STRAUB, 2014a). Aí, é que entram os *Distributed Version Control Systems* (DVCSs), ou em português, Sistemas Distribuídos de Controle de Versão (SDCV), esses sistemas conseguem resolver todos e problemas e muito mais, existem diversos software desse tipo, como o Git, Mercurial, Bazaar ou Darcs (CHACON; STRAUB, 2014a). A logo oficial do Git pode ser vista abaixo na figura 18.

Figura 18 – Logo Oficial do Git



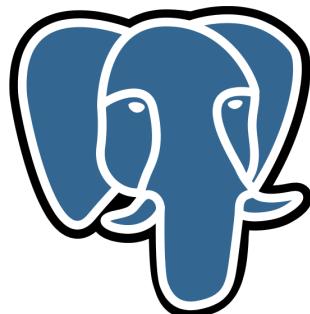
Fonte: [Long \(20??\)](#).

Em 2005, o criador do *Kernel Linux*, o Linus Torvald, junto a comunidade, lançava o mais conhecido e provavelmente o mais poderoso DCVS dentre os DVCSs existentes, o Git (CHACON; STRAUB, 2014b). No momento presente, o Git dispara na frente dos outros DCVS, a sua utilização chega na marca de 93.87% de usuários (2022..., 2023). Optamos por ele devido ele ser um DVCS muito completo, atender todas as necessidades do projeto e por quê os integrantes e orientadores do projetos já utilizam essa ferramenta, o que elimina a necessidade de aprender uma nova ferramenta e consequentemente gastar mais tempo do já não temos.

3.3.9 PostgreSQL

A escolha do Banco de Dados (BD) foi feita a partir de diversos pontos como licença, tipos de dados, preço e linguagens de programação aceitas. Essas características foram filtrando os bancos de dados até que chegamos a conclusão de que o PostgreSQL era a opção mais adequada para o projeto. A logo oficial do PostgreSQL pode ser vista abaixo na figura 19.

Figura 19 – Logo Oficial do PostgreSQL



Fonte: [Logo \(2022\)](#).

O PostgreSQL é um Sistema Gerenciador de Banco de Dados Objeto Relacional (SGBDRO), ele é um SGBDRO gratuito e *open-source* ([SOBRE, 2023](#)). Por ser um SGBDRO, ele utiliza e é baseado na linguagem SQL, além disso, ele é capaz de trabalhar com diversas outras linguagens, como Perl, Python e Tcl ([SOBRE, 2023](#)). Ele também é o SGBDRO mais utilizado, ele alcança 45,55% dos usuários ([DEVELOPER..., 2023](#)). Veja abaixo alguns pontos que fizeram ele ser escolhido.

3.3.9.1 Licença

O PostgreSQL permite o uso, a cópia, a modificação e a distribuição do software e da documentação, seja para qualquer fim. Ademais, o PostgreSQL bem como o seu SGBD (pgAdmin4) não aplica qualquer taxa sobre o usuário do software ([A..., 2009](#)).

3.3.9.2 Conformidade

O PostgreSQL é o banco que está com a maior conformidade do padrão SQL de 2016, possuindo 170 de 179 dos recursos obrigatórios do padrão ([SOBRE, 2023](#)).

3.3.9.3 Linguagem

O PostgreSQL suporta diversas linguagens procedurais, entre elas, está o Python, que como dito anteriormente, é a linguagem que vai ser utilizada no projeto ([SOBRE, 2023](#)).

3.3.10 SQL

Como dito anteriormente, optamos por utilizar o banco de dados PostgreSQL, ele é banco de dados relacional, o que significa que ele utiliza a linguagem SQL. A SQL, é a linguagem padrão para banco de dados relacionais ([ELMASRI; NAVATHE, 2011](#)). Ela permite que você faça consultas, inserções, alterações, exclusões e muito mais ([ELMASRI; NAVATHE, 2011](#)). A logo não oficial do SQL pode ser vista abaixo na figura 20.

Figura 20 – Logo Não Oficial do SQL



Fonte: [juicy_fish \(2023\)](#).

A linguagem está tão difundida por causa dos bancos de dados relacionais, que faz seu uso chegar a 48.66% ([DEVELOPER... , 2023](#)).

3.4 FERRAMENTAS

Nesta seção, aborda-se as ferramentas utilizadas no projeto, as ferramentas são criadas a afim de facilitar, agilizar e ajudar o usuário a executar tarefas. Assim como as tecnologias, utilizou-se diversas ferramentas para os mais variados processos do projeto. Alguns desses processos são, documentação, levantamento de dados, desenvolvimento, armazenamento de dados e outras coisas mais.

3.4.1 Overleaf

O *Overleaf* é um tecnologia *web*, que permite construir documentos textuais através de uma interface de edição de texto. Devido ele utilizar o *LAT_EX*em sua base no momento de criar os documentos. Ele funciona através comandos, ou seja, o que você digitar não é o resultado final do seu documento ([CRIANDO... , 2023](#)). A logo oficial do Overleaf pode ser vista abaixo na figura 21.

Figura 21 – Logo Oficial do Overleaf



Fonte: [Overleaf... \(2023\)](#).

O *Overleaf* é uma ferramente de edição de textos profissionais online que utiliza a tecnologia *LAT_EX*, com ele é possível criar desde documentos A4 até apresentações estáticas

(CRIANDO..., 2023). Com o *Overleaf*, é possível editar o documento de qualquer lugar do mundo, desde que as condições permitam (CRIANDO..., 2023).

3.4.2 *Gitbash*

O Git foi desenvolvido para rodar em sistemas Unix, como o Linux e Mac, ele é composto por diversos utilitários de linha de comando Matoso (2020). Devido o Windows não ser um sistema operacional baseado em Unix, o seu terminal se comporta de forma diferente, o que torna a compatibilidade com o git limitada Matoso (2020). Afim de resolver isso, foi desenvolvido o Git Bash, ele é uma aplicação Windows que emula um terminal Unix onde consegue oferecer o mesmo conjunto de comandos, de forma a tornar a integração mais completa Matoso (2020). A logo não oficial do GitBash pode ser vista abaixo na figura 22.

Figura 22 – Logo Não Oficial do GitBash

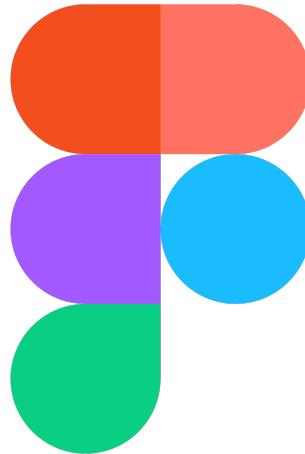


Fonte: Download... (20??).

3.4.3 *Figma*

O Figma é uma ferramenta de desenvolvimento de artes, com ela é possível que gerentes, designers, desenvolvedores e outros possam produzir suas artes (O..., 20??). Com ele é possível que você desenhe de qualquer lugar do mundo, já que, ele funciona na *web* e no *desktop* (O..., 20??). Atualmente o Figma é a ferramente de *design* principal mais utilizada no mundo, chegando a marca 78,36% de usuários (UI..., 2022). A logo oficial do Figma pode ser vista abaixo na figura 23.

Figura 23 – Logo Oficial do Figma

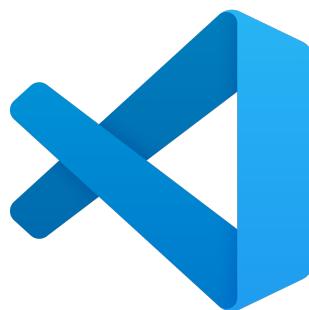


Fonte: [Diretrizes... \(20??b\)](#).

3.4.4 *Visual Studio Code*

O VSC é um editor de texto leve, o VSC é muito customizável, permitindo que você escreva seus códigos para qualquer linguagem/extensão, em geral, o editor é mais utilizado por programadores, já que dificilmente não roda em versão das plataformas como Windows, Linux ou Mac. Na época presente, o VSC é o editor de texto mais utilizado do mundo, o edito agrupa 73.71% dos programadores em volta do mundo [Developer... \(2023\)](#). A logo oficial do VSC pode ser vista abaixo na figura 24.

Figura 24 – Logo Oficial do Visual Studio Code



Fonte: [Icons... \(2023\)](#).

3.4.5 *Illustrator*

O Adobe Illustrator é uma ferramenta de *software* gráfico muito poderosa focada na edição de imagem vetoriais [Souza \(2019\)](#). Nela, é possível, editar, excluir, alterar, dimensionar, estilizar diversos tipos de objetos como quadrados, textos, estrelas, imagens e muitos outros [Souza \(2019\)](#). A logo não oficial do Illustrator pode ser vista abaixo na figura 25.

Figura 25 – Logo Não Oficial do Illustrator



Fonte: [Adobe... \(2021\)](#).

3.4.6 GitHub

O GitHub é uma plataforma para hospedar e gerenciar os códigos, ele atua na plataforma *web* permitindo que você compartilhe seus códigos e faça um “*backup*” das versões dos seus códigos. O GitHub, como o nome sugere, é compatível com o sistema de controle de versão Git, então é possível “conectar” e “subir” as versões do código gerenciado pelo Git. A logo oficial do GitHub pode ser vista abaixo na figura 26.

Figura 26 – Logo Oficial do GitHub



Fonte: [GitHub... \(2022\)](#).

3.4.7 brModelo

O brModelo é uma ferramenta de modelagem brasileira, permitindo “desenhar” modelos visuais das três etapas de modelagem de dados. A modelagem conceitual, a modelagem lógica e a modelagem física. Eles são muito úteis para entender a estrutura dos dados, cada um deles tem um propósito para cada objetivo.

3.4.8 PgAdmin

O PgAdmin é um SGBD, ele foi desenvolvido para o banco de dados PostgreSQL ([PERGUNTAS... , 2023a](#)). O PgAdmin pode rodar tanto na *Web* quanto no *desktop* seja no Linux, macOS e Windows e em navegadores como Chrome, Firefox, Edge e Safari ([PERGUNTAS... , 2023a](#)). A utilização do PgAdmin facilita diversos processos no gerenciamento do BD, por isso, o uso dele trás diversos benefícios.

3.4.9 *PowerPoint*

O PowerPoint é uma ferramenta muito utilizada no mundo, ela serve como um suporte visual para as suas apresentações [O... \(2023b\)](#). Com ela é possível criar slides e servir como um componente complementar ao conteúdo da apresentação, trazendo apenas pontos de “gatilhos” que lembrem você, o conteúdo da sua apresentação [O... \(2023b\)](#). Devido a isso, você se torna o foco da apresentação, ao invés das pessoas ficarem lendo slides [O... \(2023b\)](#). A logo não oficial do PowerPoint pode ser vista abaixo na figura 27.

Figura 27 – Logo Não Oficial do PowerPoint



Fonte: [perfect \(2023\)](#).

4 ADMINISTRADOR DE DADOS

O administrador de dados visa de forma constante e sistemática atualizar os modelos de dados que são utilizados para a criação e atualizam do banco de dados Roberta (2007). É claro que as funções do administrador de dados não se restringem apenas a essas Roberta (2007). Dentre os modelos estão o conceitual, o lógico e físico, cada um deles tem uma finalidade muito importante para o projeto Roberta (2007).

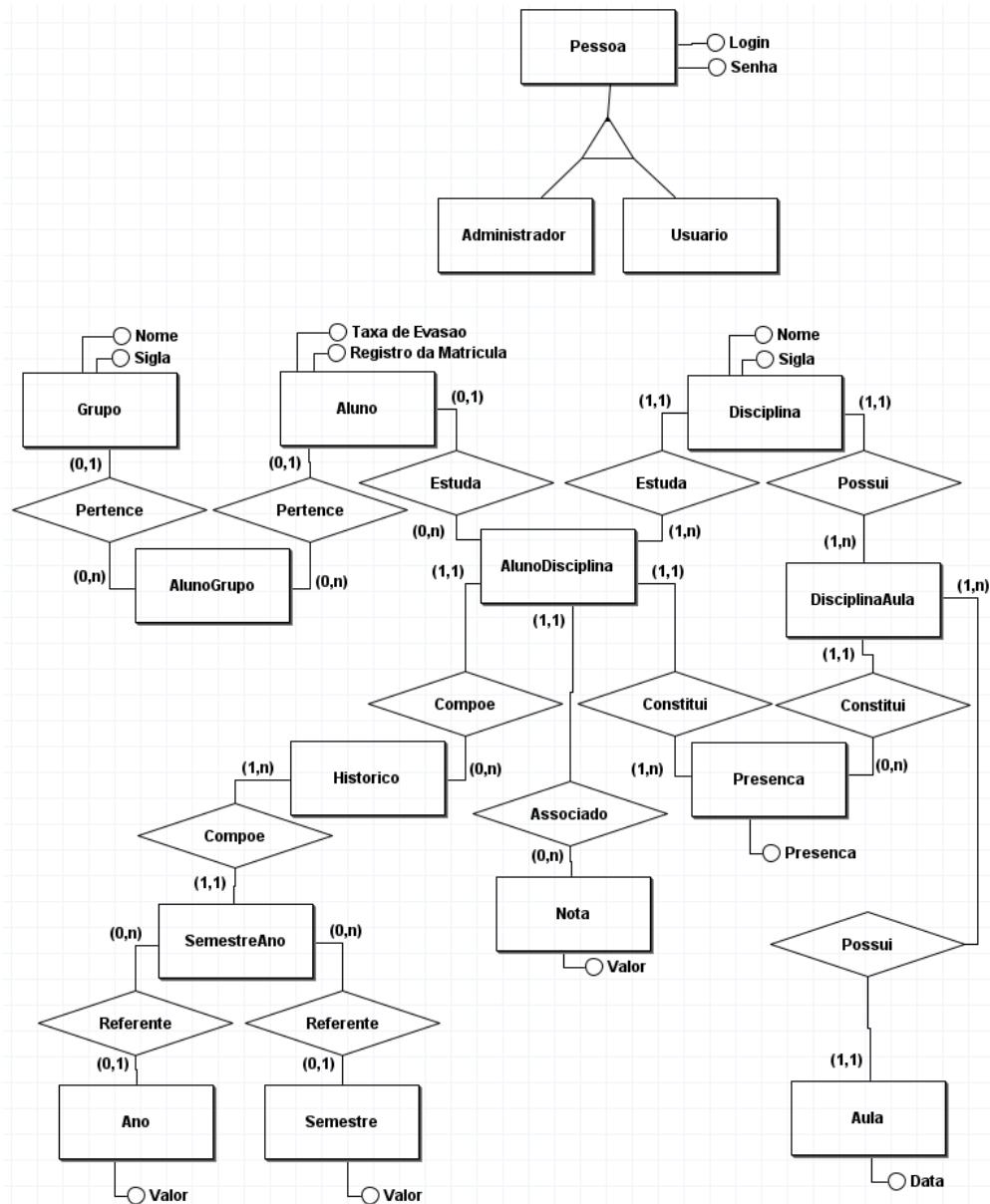
4.1 MODELAGEM DE DADOS

A modelagem de dados é um processo onde se cria modelos para dados, este modelo permite uma melhor visualização de como os dados estão relacionados, quais nome serão utilizados para eles, quais são os tipos desses dados (O..., 2023a). Basicamente, existem três tipos de modelos de dados, o modelo conceitual, o modelo lógico e modelo físico.

4.1.1 *Modelo de dados conceitual*

A modelagem de dados conceitual trata-se de um primeiro momento dentro do projeto lógico que é depois da Análise de Requisitos (TEOREY et al., 2014, p. 12-14). Ela é uma representação simples das regras de negócio, das entidades do software e dos dados dessas entidades (TEOREY et al., 2014, p.65-65). Possui a finalidade de ser inteligível por pessoas mais leigas (normalmente o cliente) e pelo projetista do banco de dados (TEOREY et al., 2014, 18-18). Com propósito de compreender se a análise de requisitos foi executado com precisão por ambas as partes (cliente e fornecedor do software), além disso, manter uma comunicação mais acessível com pessoas que não possuem conhecimento técnico. O modelo de dados conceitual do projeto pode ser visto na figura 28.

Figura 28 – Modelo Conceitual

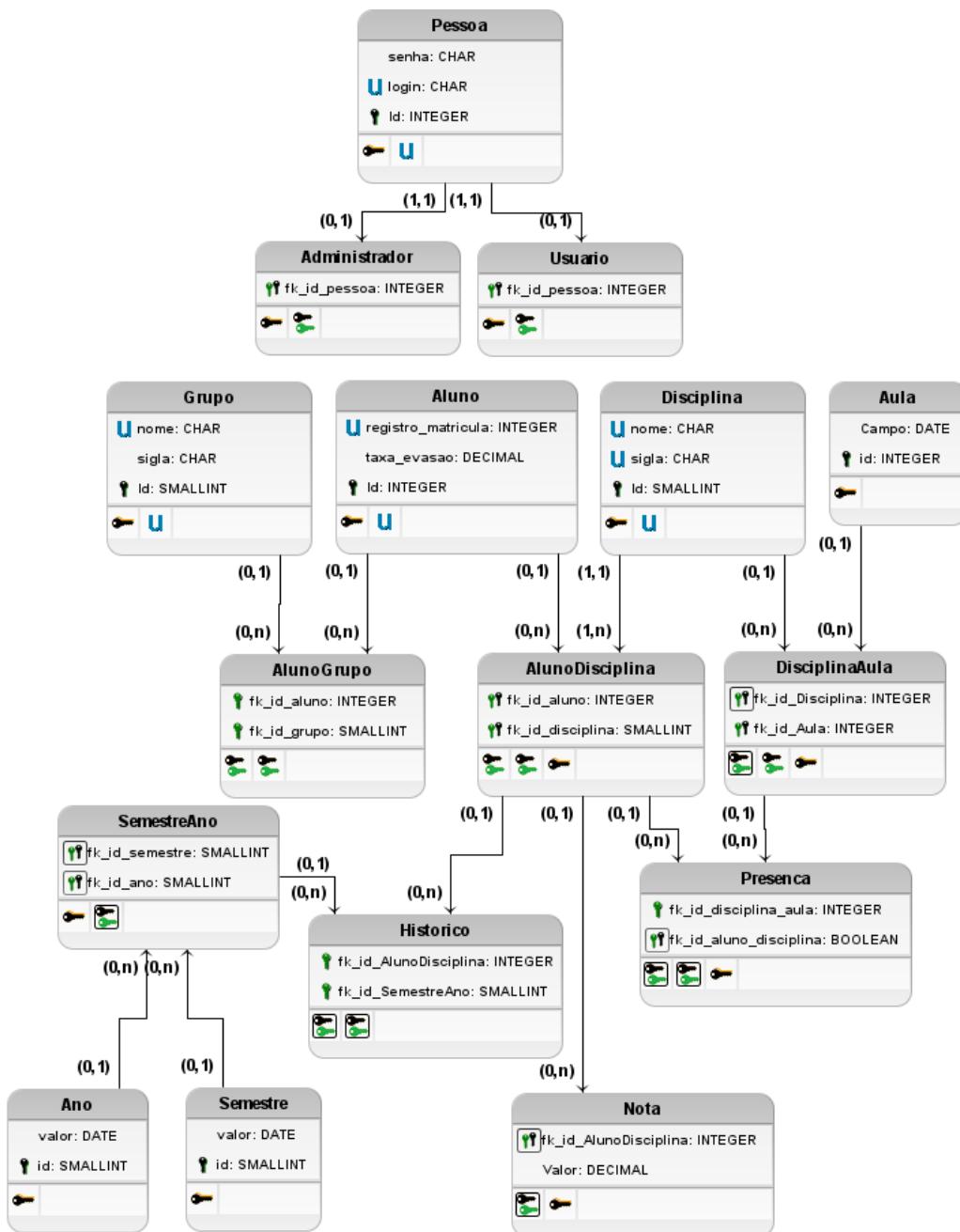


Fonte: elaborado pelo autor (2023).

4.1.2 Modelo de dados lógico

O modelo lógico é o meio termo entre pouco técnico e abrangente (modelo de dados conceitual) e o muito técnico e específico (modelo de dados físico), podendo ser visto como uma ponte entre o modelo conceitual e o modelo físico (O... , 2023a). Esse modelo vai mais fundo nas necessidades de negócio, paralelamente ele é modelado para algum tipo de banco de dados (relacional, chave-valor, orientado a objetos, etc) (DANILO, 2018). O modelo de dados lógico do projeto pode ser visto na figura 29.

Figura 29 – Modelo Lógico

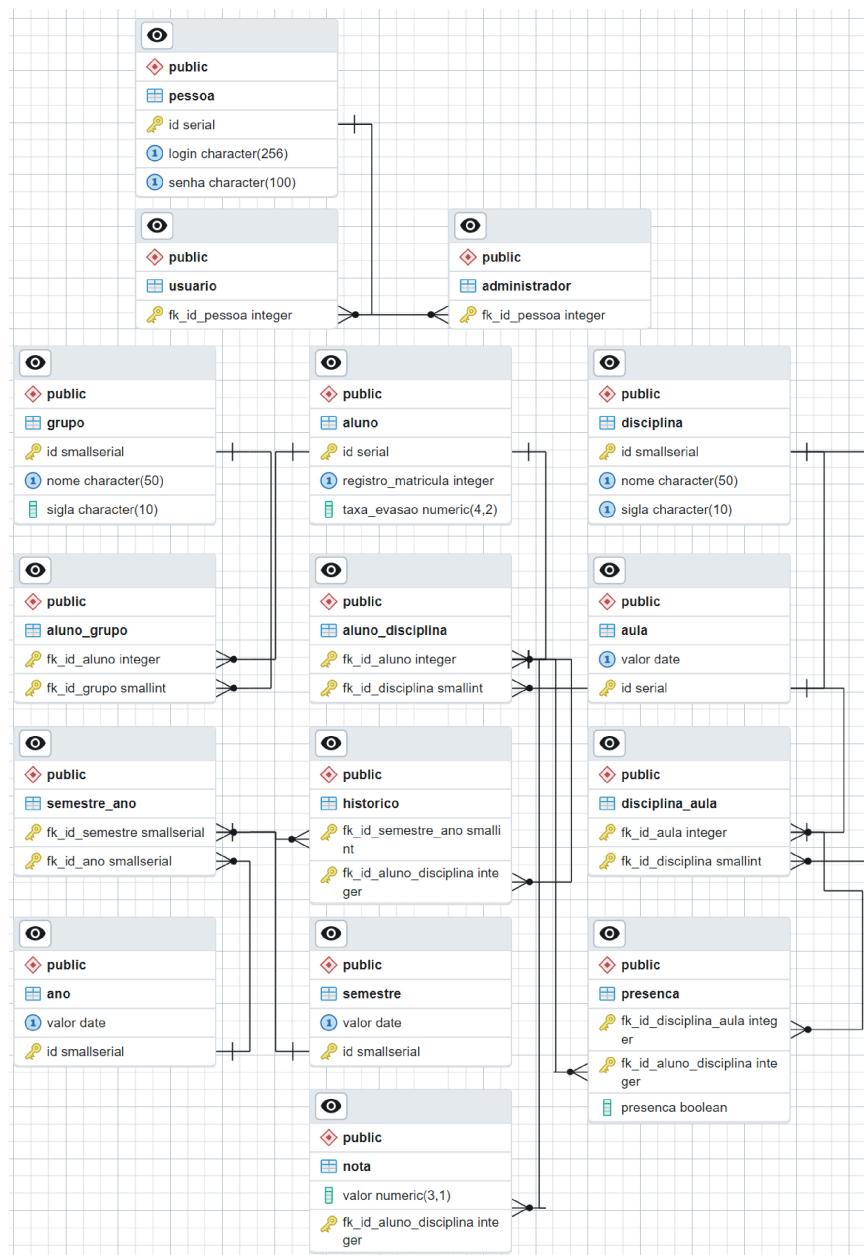


Fonte: elaborado pelo autor (2023).

4.1.3 Modelo de dados físico

O modelo de dados físico é o nível mais profundo antes da implementação do banco de dados, ele desenvolvido baseado nas regras do SGBD que será utilizado e também sob outros parâmetros (MODELAGEM..., 2023). Seu objetivo é ajudar a visualizar o banco de dados é através de uma representação, a representação específica como o banco de dados será criado para um determinado SGBD, assim é possível obter um melhor entendimento sobre os objetos e relações dos dados daquele software (MODELAGEM..., 2023). O modelo de dados físico do projeto pode ser visto na figura 30.

Figura 30 – Modelo Físico



Fonte: elaborado pelo autor (2023).

4.2 DICIONÁRIO DE DADOS E SCRIPTS

O dicionário de dados é um modelo importante para a documentação do banco de dados, com ele é possível ter uma visualização mais estruturada e organizada que os modelos de dados. O dicionário de dados normalmente é criado em tabelas, onde dispõe de diversos metadados como nome, tamanho, tipo, *Primary Key* e quantidade de campos, os metadados podem ser de diversas estruturas como tabelas, relacionamento, entidades, campos, etc ([DICIONÁRIO... 2023](#)). Com isso, os programadores não precisam consultar o profissional de banco e facilita o entendimento e a visualização da estrutura do banco de dados.

4.2.1 Tabela Pessoa

O dicionário de dados da *Tabela Pessoa* pode ser vista na tabela 3 e o script SQL da *Tabela Pessoa* pode ser visto no código 1.

Quadro 3 – Dicionário de dados da tabela Pessoa.

Nome:	Pessoa							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
id	SERIAL				Sim			
login	CHARACTER	256		Sim			Sim	
senha	CHARACTER	100		Sim				

Fonte: elaborado pelo autor (2023).

Código 1 – Código SQL da tabela Pessoa

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.pessoa(
3     id serial,
4     login character(256) NOT NULL,
5     senha character(100) NOT NULL,
6     CONSTRAINT pk_pessoa PRIMARY KEY (id),
7     CONSTRAINT unicos_grupo UNIQUE (login, senha)
8   );
9 END;

```

Fonte: elaborado pelo autor (2023).

4.2.2 Tabela Administrador

O dicionário de dados da *Tabela Administrador* pode ser vista na tabela 4 e o script SQL da *Tabela Administrador* pode ser visto no código 2.

Quadro 4 – Dicionário de dados da tabela Administrador.

Nome:	Administrador							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
fk_id_pessoa	INTEGER				Sim			Sim

Fonte: elaborado pelo autor (2023).

Código 2 – Código SQL da tabela Administrador

```

1 BEGIN;
2     CREATE TABLE IF NOT EXISTS public.administrador(
3         fk_id_pessoa integer,
4         PRIMARY KEY (fk_id_pessoa)
5     );
6     ALTER TABLE IF EXISTS public.administrador
7         ADD CONSTRAINT fk_administrador FOREIGN KEY (fk_id_pessoa)
8             REFERENCES public.pessoa (id) MATCH SIMPLE
9             ON UPDATE NO ACTION
10            ON DELETE NO ACTION
11            NOT VALID;
12 END;

```

Fonte: elaborado pelo autor (2023).

4.2.3 Tabela Usuário

O dicionário de dados da *Tabela Usuário* pode ser vista na tabela 5 e o *script* SQL da *Tabela Usuário* pode ser visto no código 3.

Quadro 5 – Dicionário de dados da tabela Usuário.

Nome:	Usuário							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
fk_id_pessoa	INTEGER				Sim			Sim

Fonte: elaborado pelo autor (2023).

Código 3 – Código SQL da tabela Usuário

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.usuario(
3     fk_id_pessoa integer,
4     PRIMARY KEY (fk_id_pessoa)
5   );
6   ALTER TABLE IF EXISTS public.usuario
7     ADD CONSTRAINT fk_usuario FOREIGN KEY (fk_id_pessoa)
8     REFERENCES public.pessoa (id) MATCH SIMPLE
9     ON UPDATE NO ACTION
10    ON DELETE NO ACTION
11    NOT VALID;
12 END;

```

Fonte: elaborado pelo autor (2023).

4.2.4 Tabela Grupo

O dicionário de dados da *Tabela Grupo* pode ser vista na tabela 6 e o *script* SQL da *Tabela Grupo* pode ser visto no código 4.

Quadro 6 – Dicionário de dados da tabela Grupo.

Nome:	Grupo							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
id	SMALLSERIAL				Sim			
nome	CHARACTER	50		Sim			Sim	
sigla	CHARACTER	10		Sim				

Fonte: elaborado pelo autor (2023).

Código 4 – Código SQL da tabela Grupo

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.grupo(
3     id smallserial,
4     nome character(50) NOT NULL,
5     sigla character(10) NOT NULL,
6     PRIMARY KEY (id),
7     CONSTRAINT unicos_grupo UNIQUE (nome)
8   );
9 END;

```

Fonte: elaborado pelo autor (2023).

4.2.5 Tabela Aluno

O dicionário de dados da *Tabela Aluno* pode ser vista na tabela 7 e o script SQL da *Tabela Aluno* pode ser visto no código 5.

Quadro 7 – Dicionário de dados da tabela Aluno.

Nome:	Aluno							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
id	SERIAL				Sim			
registro_matricula	INTEGER			Sim			Sim	
taxa_evasao	NUMERIC	4	2					

Fonte: elaborado pelo autor (2023).

Código 5 – Código SQL da tabela Aluno

```

1 BEGIN;
2     CREATE TABLE IF NOT EXISTS public.aluno(
3         id serial,
4         registro_matricula integer NOT NULL,
5         taxa_evasao numeric(4, 2),
6         CONSTRAINT pk_aluno PRIMARY KEY (id),
7         CONSTRAINT unicos UNIQUE (registro_matricula)
8     );
9 END;

```

Fonte: elaborado pelo autor (2023).

4.2.6 Tabela Aluno_Grupo

O dicionário de dados da *Tabela Aluno_Grupo* pode ser vista na tabela 8 e o script SQL da *Tabela Aluno_Grupo* pode ser visto no código 6.

Quadro 8 – Dicionário de dados da tabela Aluno_Grupo.

Nome:	Aluno_Grupo							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
fk_id_aluno	INTEGER				Sim			Sim
fk_id_grupo	SMALLINT				Sim			Sim

Fonte: elaborado pelo autor (2023).

Código 6 – Código SQL da tabela Aluno_Grupo

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.aluno_grupo(
3     fk_id_aluno integer,
4     fk_id_grupo smallint,
5     CONSTRAINT pk_aluno_grupo PRIMARY KEY (fk_id_aluno,
6       fk_id_grupo)
7   );
8
9   ALTER TABLE IF EXISTS public.aluno_grupo
10    ADD CONSTRAINT fk_id_grupo FOREIGN KEY (fk_id_grupo)
11      REFERENCES public.grupo (id) MATCH SIMPLE
12      ON UPDATE NO ACTION
13      ON DELETE NO ACTION
14      NOT VALID;
15
16   ALTER TABLE IF EXISTS public.aluno_grupo
17    ADD CONSTRAINT fk_id_aluno FOREIGN KEY (fk_id_aluno)
18      REFERENCES public.aluno (id) MATCH SIMPLE
19      ON UPDATE NO ACTION
20      ON DELETE NO ACTION
21      NOT VALID;
22
23 END;

```

Fonte: elaborado pelo autor (2023).

4.2.7 Tabela Disciplina

O dicionário de dados da *Tabela Disciplina* pode ser vista na tabela 9 e o script SQL da *Tabela Disciplina* pode ser visto no código 7.

Quadro 9 – Dicionário de dados da tabela Disciplina.

Nome:	Disciplina							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
id	SMALLSERIAL				Sim			
nome	CHARACTER	50		Sim			Sim	
sigla	CHARACTER	10		Sim			Sim	

Fonte: elaborado pelo autor (2023).

Código 7 – Código SQL da tabela Disciplina

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.disciplina(
3     id smallserial,
4     nome character(50) NOT NULL,
5     sigla character(10) NOT NULL,
6     PRIMARY KEY (id),
7     CONSTRAINT unico_disciplina UNIQUE (nome, sigla)
8   );
9 END;

```

Fonte: elaborado pelo autor (2023).

4.2.8 Tabela Aluno_Disciplina

O dicionário de dados da *Tabela Aluno_Disciplina* pode ser vista na tabela 10 e o *script* SQL da *Tabela Aluno_Disciplina* pode ser visto no código 8.

Quadro 10 – Dicionário de dados da tabela Aluno_Disciplina.

Nome:	Aluno_Disciplina							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
fk_id_aluno	INTEGER				Sim			Sim
fk_id_disciplina	SMALLINT				Sim			Sim

Fonte: elaborado pelo autor (2023).

Código 8 – Código SQL da tabela Aluno_Disciplina

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.aluno_disciplina(
3     fk_id_aluno integer,
4     fk_id_disciplina smallint,
5     PRIMARY KEY (fk_id_aluno, fk_id_disciplina)
6   );
7
8   ALTER TABLE IF EXISTS public.aluno_disciplina
9     ADD CONSTRAINT fk_id_aluno FOREIGN KEY (fk_id_aluno)
10    REFERENCES public.aluno (id) MATCH SIMPLE
11    ON UPDATE NO ACTION
12    ON DELETE NO ACTION
13    NOT VALID;
14
15   ALTER TABLE IF EXISTS public.aluno_disciplina
16     ADD CONSTRAINT fk_id_disciplina FOREIGN KEY (fk_id_disciplina
17       )
18     REFERENCES public.disciplina (id) MATCH SIMPLE
19     ON UPDATE NO ACTION
20     ON DELETE NO ACTION
21     NOT VALID;
21 END;

```

Fonte: elaborado pelo autor (2023).

4.2.9 Tabela Nota

O dicionário de dados da *Tabela Nota* pode ser vista na tabela 11 e o *script* SQL da *Tabela Nota* pode ser visto no código 9.

Quadro 11 – Dicionário de dados da tabela Nota.

Nome:	Nota							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
fk_id_aluno_ disciplina	INTEGER				Sim			Sim
valor	NUMERIC	3	1	Sim				

Fonte: elaborado pelo autor (2023).

Código 9 – Código SQL da tabela nota

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.nota(
3     valor numeric(3, 1) NOT NULL,
4     fk_id_aluno_disciplina integer,
5     PRIMARY KEY (fk_id_aluno_disciplina)
6   );
7
8   ALTER TABLE IF EXISTS public.nota ADD CONSTRAINT
9     fk_id_aluno_disciplina FOREIGN KEY (fk_id_aluno_disciplina,
10    fk_id_aluno_disciplina)
11    REFERENCES public.aluno_disciplina (fk_id_aluno,
12      fk_id_disciplina) MATCH SIMPLE
13    ON UPDATE NO ACTION
14    ON DELETE NO ACTION
15    NOT VALID;
16 END;

```

Fonte: elaborado pelo autor (2023).

4.2.10 Tabela Aula

O dicionário de dados da *Tabela Aula* pode ser vista na tabela 12 e o *script* SQL da *Tabela Aula* pode ser visto no código 10.

Quadro 12 – Dicionário de dados da tabela Aula.

Nome:	Aula							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
id	SERIAL				Sim			
valor	DATE			Sim			Sim	

Fonte: elaborado pelo autor (2023).

Código 10 – Código SQL da tabela Aula

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.aula(
3     valor date NOT NULL,
4     id serial,
5     PRIMARY KEY (id),
6     CONSTRAINT unico_aula UNIQUE (valor)
7   );
8 END;

```

Fonte: elaborado pelo autor (2023).

4.2.11 Tabela Disciplina_Aula

O dicionário de dados da *Tabela Disciplina_Aula* pode ser vista na tabela 13 e o *script SQL* da *Tabela Disciplina_Aula* pode ser visto no código 11.

Quadro 13 – Dicionário de dados da tabela Disciplina_Aula.

Nome:	Disciplina_Aula							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
fk_id_aula	INTEGER				Sim			Sim
fk_id_disciplina	SMALLINT				Sim			Sim

Fonte: elaborado pelo autor (2023).

Código 11 – Código SQL da tabela Disciplina_Aula

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.disciplina_aula(
3     fk_id_aula integer,
4     fk_id_disciplina smallint,
5     PRIMARY KEY (fk_id_aula, fk_id_disciplina)
6   );
7
8   ALTER TABLE IF EXISTS public.disciplina_aula
9     ADD CONSTRAINT fk_id_aula FOREIGN KEY (fk_id_aula)
10    REFERENCES public.aula (id) MATCH SIMPLE
11    ON UPDATE NO ACTION
12    ON DELETE NO ACTION
13    NOT VALID;
14
15   ALTER TABLE IF EXISTS public.disciplina_aula
16     ADD CONSTRAINT fk_id_disciplina FOREIGN KEY (fk_id_disciplina
17       )
18     REFERENCES public.disciplina (id) MATCH SIMPLE
19     ON UPDATE NO ACTION
20     ON DELETE NO ACTION
21     NOT VALID;
22 END;
```

Fonte: elaborado pelo autor (2023).

4.2.12 Tabela Semestre

O dicionário de dados da *Tabela Semestre* pode ser vista na tabela 14 e o *script SQL* da *Tabela Semestre* pode ser visto no código 12.

Quadro 14 – Dicionário de dados da tabela Semestre.

Nome:	Semestre							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
id	SMALLSERIAL				Sim			
valor	DATE			Sim			Sim	

Fonte: elaborado pelo autor (2023).

Código 12 – Código SQL da tabela Semestre

```

1 BEGIN;
2     CREATE TABLE IF NOT EXISTS public.semestre(
3         valor date NOT NULL,
4         id smallserial,
5         PRIMARY KEY (id),
6         CONSTRAINT unico_semestre UNIQUE (valor)
7     );
8 END;

```

Fonte: elaborado pelo autor (2023).

4.2.13 Tabela Presença

O dicionário de dados da *Tabela Presença* pode ser vista na tabela 15 e o script SQL da *Tabela Presença* pode ser visto no código 13.

Quadro 15 – Dicionário de dados da tabela Presença.

Nome:	Presença							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
fk_id_disciplina_aula	INTEGER				Sim			Sim
fk_id_aluno_disciplina	INTEGER				Sim			Sim
presenca	BOOLEAN			Sim				

Fonte: elaborado pelo autor (2023).

Código 13 – Código SQL da tabela Presença

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.presenca(
3     fk_id_disciplina_aula integer,
4     fk_id_aluno_disciplina integer,
5     presenca boolean NOT NULL,
6     CONSTRAINT pk_presenca PRIMARY KEY (fk_id_disciplina_aula,
7       fk_id_aluno_disciplina)
8   );
9
10  ALTER TABLE IF EXISTS public.presenca
11    ADD CONSTRAINT fk_id_disciplina_aula FOREIGN KEY (
12      fk_id_disciplina_aula, fk_id_disciplina_aula)
13    REFERENCES public.disciplina_aula (fk_id_aula,
14      fk_id_disciplina) MATCH SIMPLE
15    ON UPDATE NO ACTION
16    ON DELETE NO ACTION
17    NOT VALID;
18
19  ALTER TABLE IF EXISTS public.presenca
20    ADD CONSTRAINT fk_id_aluno_disciplina FOREIGN KEY (
21      fk_id_aluno_disciplina, fk_id_aluno_disciplina)
22    REFERENCES public.aluno_disciplina (fk_id_aluno,
23      fk_id_disciplina) MATCH SIMPLE
24    ON UPDATE NO ACTION
25    ON DELETE NO ACTION
26    NOT VALID;
27
28 END;

```

Fonte: elaborado pelo autor (2023).

4.2.14 Tabela Ano

O dicionário de dados da *Tabela Ano* pode ser vista na tabela 16 e o *script* SQL da *Tabela Ano* pode ser visto no código 14.

Quadro 16 – Dicionário de dados da tabela Ano.

Nome:	Ano							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
id	SMALLSERIAL				Sim			
valor	DATE			Sim			Sim	

Fonte: elaborado pelo autor (2023).

Código 14 – Código SQL da tabela Ano

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.ano(
3     valor date NOT NULL,
4     id smallserial,
5     PRIMARY KEY (id),
6     CONSTRAINT unico_semestre UNIQUE (valor)
7   );
8 END;

```

Fonte: elaborado pelo autor (2023).

4.2.15 Tabela Semestre_Ano

O dicionário de dados da *Tabela Semestre_Ano* pode ser vista na tabela 17 e o script SQL da *Tabela Semestre_Ano* pode ser visto no código 15.

Quadro 17 – Dicionário de dados da tabela Semestre_Ano.

Nome:	Semestre_Ano							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
fk_id_semestre	SMALLSERIAL				Sim			Sim
fk_id_ano	SMALLSERIAL				Sim			Sim

Fonte: elaborado pelo autor (2023).

Código 15 – Código SQL da tabela Semestre_Ano

```

1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.semestre_ano(
3     fk_id_semestre smallserial,
4     fk_id_ano smallserial,
5     PRIMARY KEY (fk_id_semestre, fk_id_ano)
6   );
7
8   ALTER TABLE IF EXISTS public.semestre_ano
9     ADD CONSTRAINT fk_id_ano FOREIGN KEY (fk_id_ano)
10    REFERENCES public.ano (id) MATCH SIMPLE
11    ON UPDATE NO ACTION
12    ON DELETE NO ACTION
13    NOT VALID;
14
15   ALTER TABLE IF EXISTS public.semestre_ano
16     ADD CONSTRAINT fk_id_semestre FOREIGN KEY (fk_id_semestre)
17    REFERENCES public.semestre (id) MATCH SIMPLE
18    ON UPDATE NO ACTION
19    ON DELETE NO ACTION
20    NOT VALID;
21 END;

```

Fonte: elaborado pelo autor (2023).

4.2.16 Tabela Histórico

O dicionário de dados da *Tabela Histórico* pode ser vista na tabela 18 e o script SQL da *Tabela Histórico* pode ser visto no código 16.

Quadro 18 – Dicionário de dados da tabela Histórico.

Nome:	Histórico							
Schema:	Public							
Descrição:	Descreve os metadados da tabela.							
Campos/Colunas:								
Nome	Tipo	Tamanho/ Precisão	Escala	Não Nulo	Chave Primária	Padrão	Único	Chave Estrangeira
fk_id_semestre_ano	SMALLINT				Sim			Sim
fk_id_aluno_disciplina	INTEGER				Sim			Sim

Fonte: elaborado pelo autor (2023).

Código 16 – Código SQL da tabela Histórico

```
1 BEGIN;
2   CREATE TABLE IF NOT EXISTS public.historico(
3     fk_id_semestre_ano smallint,
4     fk_id_aluno_disciplina integer,
5     CONSTRAINT id_historico PRIMARY KEY (fk_id_semestre_ano,
6       fk_id_aluno_disciplina)
7   );
8
9   ALTER TABLE IF EXISTS public.historico
10    ADD CONSTRAINT fk_id_aluno_disciplina FOREIGN KEY (
11      fk_id_aluno_disciplina, fk_id_aluno_disciplina)
12      REFERENCES public.aluno_disciplina (fk_id_aluno,
13        fk_id_disciplina) MATCH SIMPLE
14      ON UPDATE NO ACTION
15      ON DELETE NO ACTION
16      NOT VALID;
17
18   ALTER TABLE IF EXISTS public.historico
19    ADD CONSTRAINT fk_id_semestre_ano FOREIGN KEY (
20      fk_id_semestre_ano, fk_id_semestre_ano)
21      REFERENCES public.semestre_ano (fk_id_semestre, fk_id_ano)
22      MATCH SIMPLE
23      ON UPDATE NO ACTION
24      ON DELETE NO ACTION
25      NOT VALID;
26
27 END;
```

Fonte: elaborado pelo autor (2023).

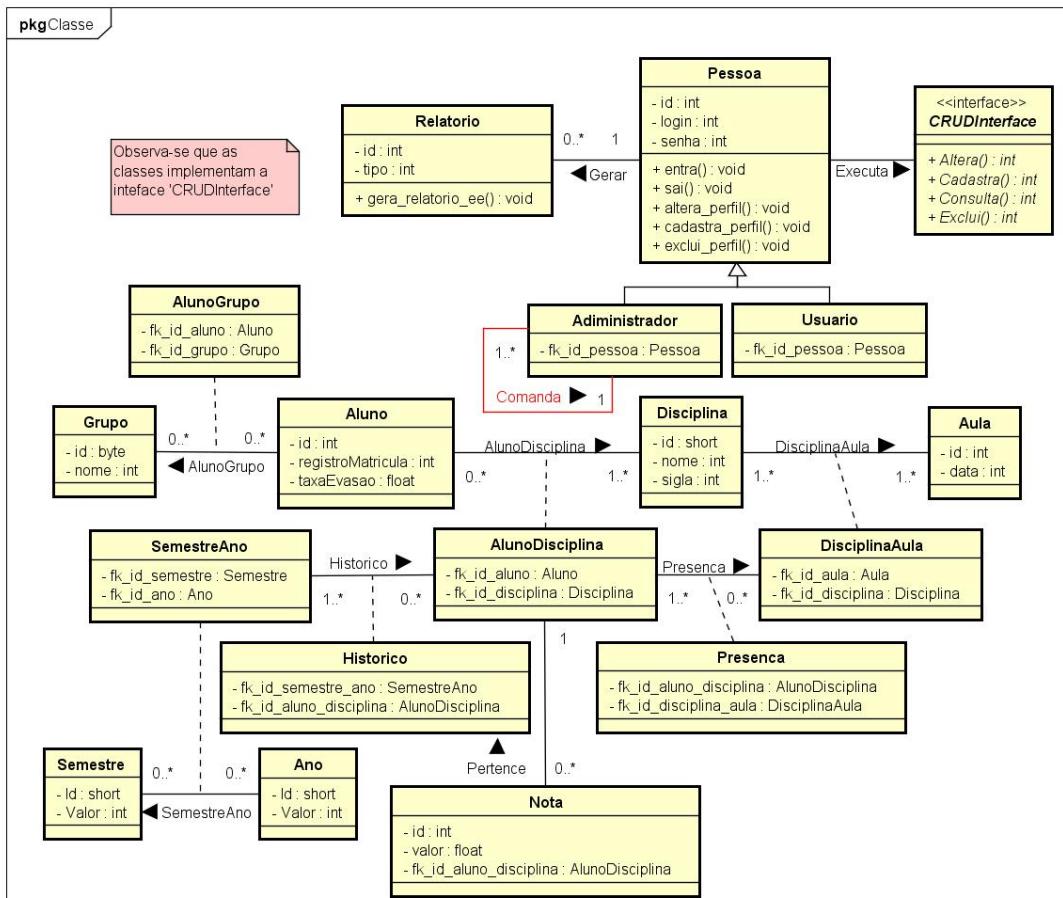
5 MODELAGEM DE SOFTWARE

A modelagem de sistemas é outro processo muito importante na engenharia de *software*, a partir da qual é possível entender o sistema de forma abstrata. A mesma é o processo de 'desenhar' modelos gráficos que apresentam uma determinada visão do sistema, para o processo de criação dessas visões, é utilizado alguma notação [Sommerville \(2011\)](#). No contemporâneo, a notação mais utilizada é a UML, a própria permite desenhar os programas por diversas perspectivas, comumente chamados de diagramas, assim como os diagramas de classes, diagrama de casos de uso, diagramas de sequência e outros [Sommerville \(2011\)](#).

5.1 DIAGRAMA DE CLASSES

Devido o diagrama de classes ser encontrados com uma maior frequência na modelagem de sistemas orientados a objetos por causa da visualização, especificação e documentação de modelos estruturais, ele é um importante diagrama da UML ([BOOCH; RUMBAUGH; JACOBSON, 2012](#), p. 175). O mesmo permite modelar uma visão estática das classes do sistema, nele, é possível visualizar relacionamentos, atributos e suas características, métodos e suas características, cardinalidades e outros ([BOOCH; RUMBAUGH; JACOBSON, 2012](#), p. 175). Além das vantagens já citadas, com o diagrama de classes é possível construir sistemas executáveis por meio da engenharia direta e reversa ([BOOCH; RUMBAUGH; JACOBSON, 2012](#), p. 175). O [DIAGRAMA DE CLASSES](#) pode ser visto na figura 31.

Figura 31 – Diagrama de Classe

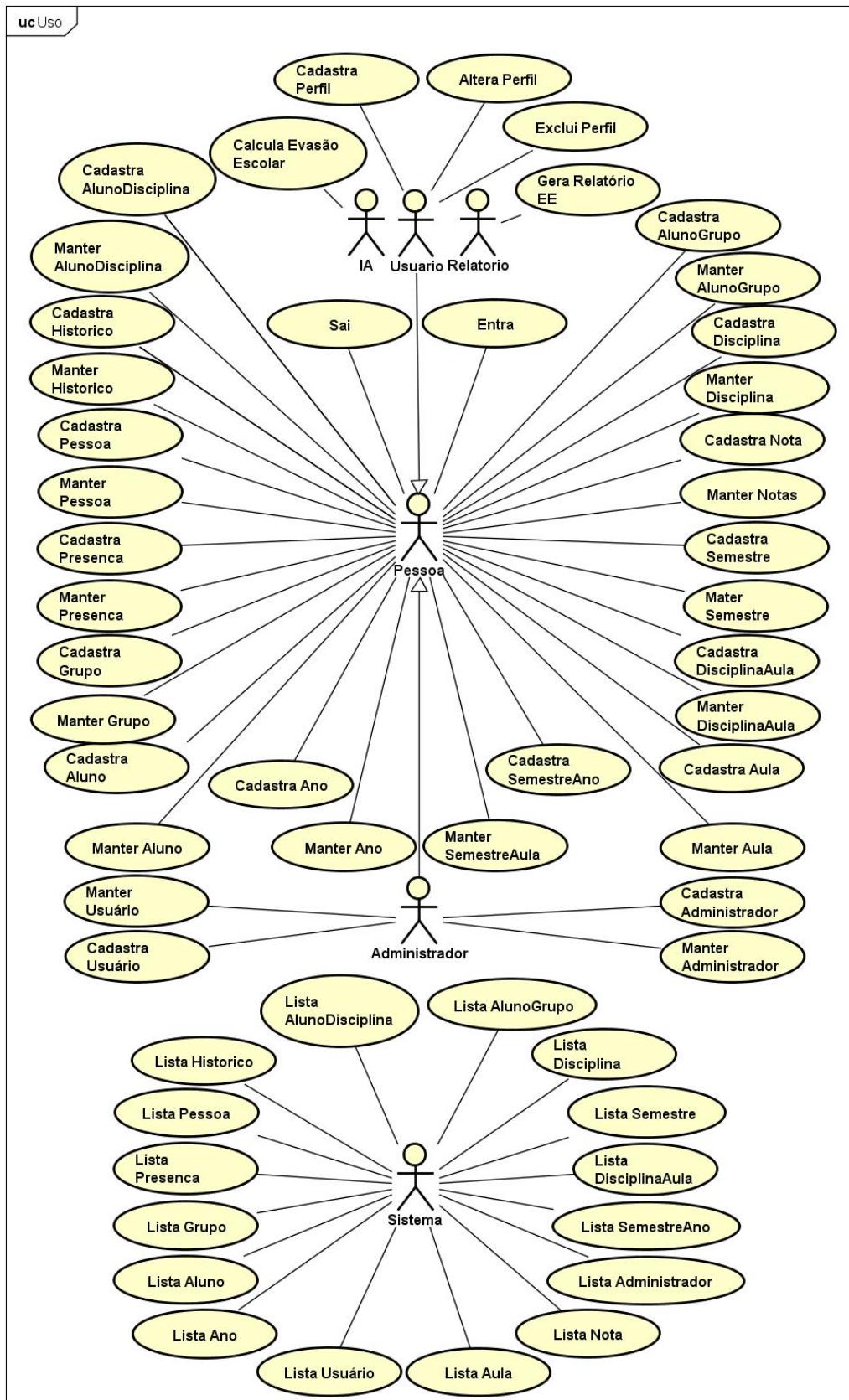


Fonte: elaborado pelo autor (2023).

5.2 DIAGRAMA DE CASOS DE USO

O diagrama de caso de uso é outro diagrama muito importante na modelagem do software, pois ele define quem vai fazer o que (BOOCH; RUMBAUGH; JACOBSON, 2012, p. 345). Através dele é possível especificar, visualizar e documentar os comportamentos dos atores do sistema, que vão desde os usuários até o próprio sistema (BOOCH; RUMBAUGH; JACOBSON, 2012, p. 343-345). O [DIAGRAMA DE CASOS DE USO](#) pode ser visto na figura 32.

Figura 32 – Diagrama de casos de uso



Fonte: elaborado pelo autor (2023).

5.3 DIAGRAMAS DE SEQUÊNCIA

Os diagramas de sequência, também chamados de diagrama de interação tem o objetivo de especificar, visualizar, construir e documentar a interação cronológica de objetos e seus relacionamentos (BOOCH; RUMBAUGH; JACOBSON, 2012, p. 378). Normalmente ele representa algum caso de uso, através dele é possível definir os atores, os objetos, as mensagens, os comportamentos e outros, onde você o constrói as interações cronologicamente (BOOCH; RUMBAUGH; JACOBSON, 2012, p. 377).

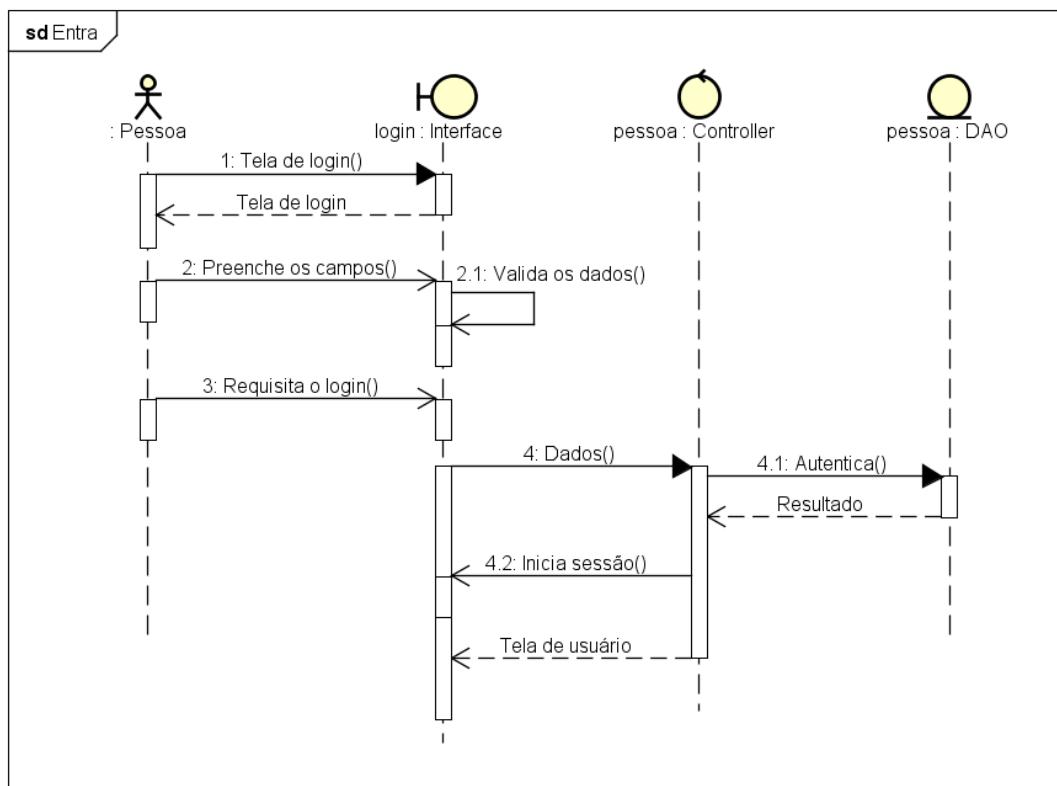
5.3.1 Pessoa

Esta seção engloba os diagramas de sequência dos casos de uso do ator *Pessoa* e a documentação dos diagramas de sequência.

5.3.1.1 Entra

O diagrama de sequência da função “Entra” pode ser vista na figura 33 e a documentação do mesmo pode ser visto no quadro 19.

Figura 33 – Diagrama de sequência Entra



Fonte: elaborado pelo autor (2023).

Quadro 19 – Documentação do caso de uso Entra.

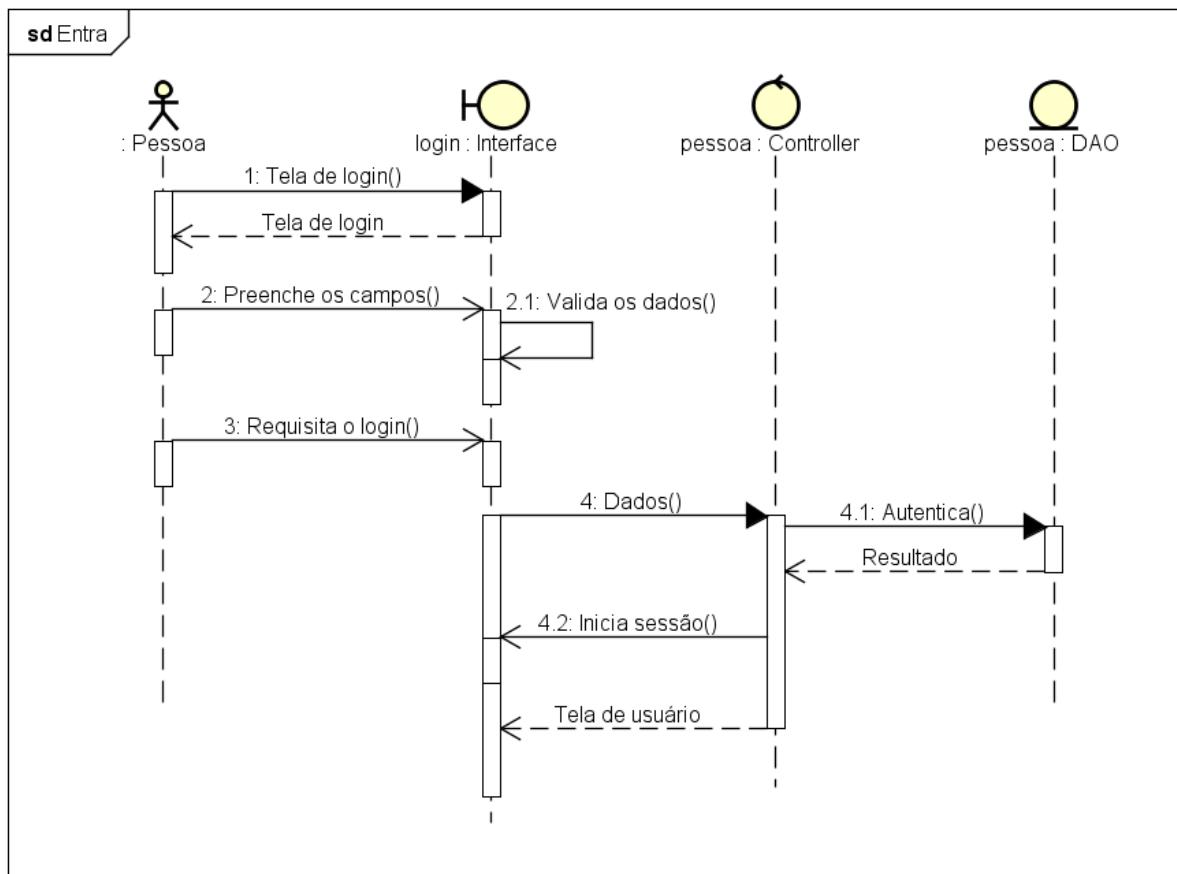
Documentação	
Nome do caso de uso	Entra
Caso de uso geral	
Autor principal	Pessoa
Atores secundários	
Resumo	Descreve o fluxo percorrido para entrar no sistema
Pré condições	Ter uma conta cadastrada
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de entrar	
	2 - Redireciona para a tela de entrar
3 - Preenche os campos	
	4 - Válida os dados
5 - Requisita a entrada no sistema	
	6 - Envia os dados para o servidor
	7 - Procura no banco de dados
	8 - Retorna mensagem de sucesso da entrada
	9 - Inicia a sessão
	10 - Redireciona para a tela de usuário
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	4.1 - Informa ao usuário que os valores estão em desacordo com as regras do campo
	8.1 - Retorna que não foi possível procurar no banco de dados
	8.2 - Retorna ao item 3
	8.3 - Informa ao usuário que não foi possível procurar no banco de dados
	10.1 - Redireciona para a tela de erro
Restrições/Validações	

Fonte: Elaborado pelo autor (2023).

5.3.1.2 Sai

O diagrama de sequência da função “Sai” pode ser vista na figura 34 e a documentação do mesmo pode ser visto no quadro 20.

Figura 34 – Diagrama de sequência Sai



Fonte: elaborado pelo autor (2023).

Quadro 20 – Documentação do caso de uso Sai.

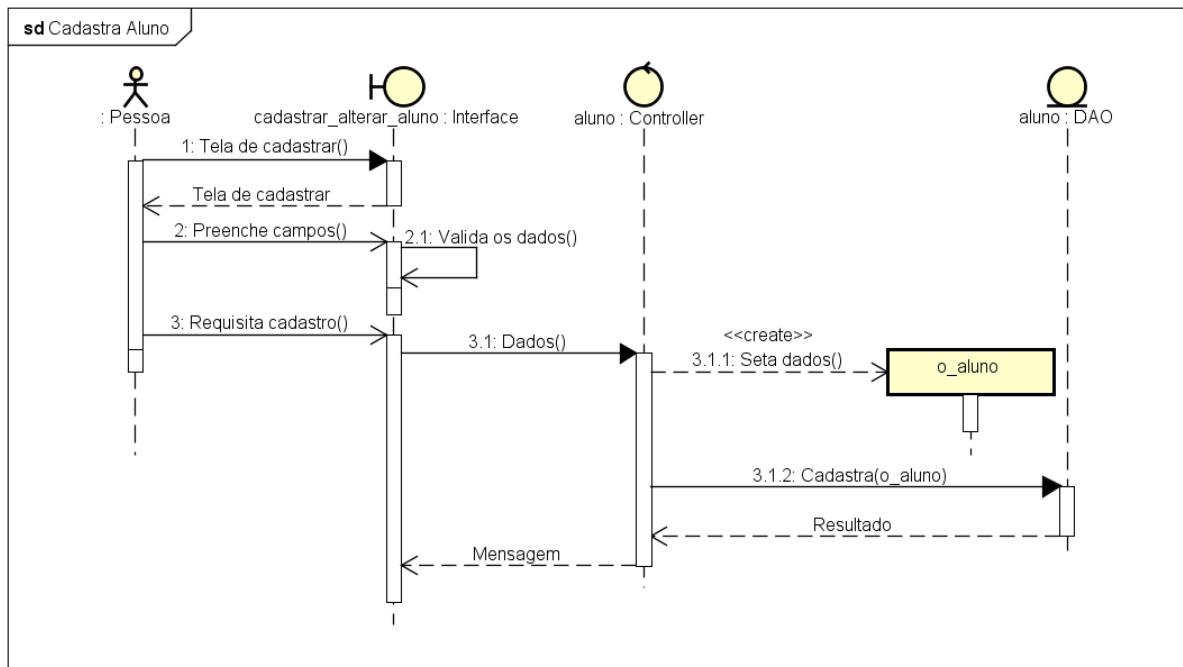
Documentação	
Nome do caso de uso	Sai
Caso de uso geral	
Autor principal	Pessoa
Atores secundários	
Resumo	Descreve o fluxo percorrido para sair do sistema
Pré condições	Estar logado
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a saída do sistema	
	2 - Envia a requisição para o servidor
	3 - Encerra sessão
	4 - Redireciona para a tela de entrar
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	4.1 - Redireciona para a tela de erro
Restrições/Validações	

Fonte: Elaborado pelo autor (2023).

5.3.1.3 Cadastra Aluno

O diagrama de sequência da função “Cadastra Aluno” pode ser vista na figura 35 e a documentação do mesmo pode ser visto no quadro 21.

Figura 35 – Diagrama de sequência Cadastra Aluno



Fonte: elaborado pelo autor (2023).

Quadro 21 – Documentação do caso de uso Cadastra Aluno.

Documentação	
Nome do caso de uso	Cadastra Aluno
Caso de uso geral	
Autor principal	Pessoa
Atores secundários	
Resumo	Descreve o fluxo percorrido para o cadastro de um aluno
Pré condições	Estar logado
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de alterar/cadastrar dos administradores	
	2 - Redireciona para a tela de alterar/cadastrar dos administradores
3 - Preenche campos	
	4 - Válida os dados
5 - Requisita o cadastro	
	6 - Envia os dados para o servidor
	7 - Cadastra no banco de dados
	8 - Retorna mensagem de sucesso do cadastro
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	4.1 - Informa ao usuário que os valores estão em desacordo com as regras dos campos
	6.1 - Redireciona para a tela de erro
	8.1 - Retorna que não foi possível cadastrar no banco de dados
	8.2 - Retorna ao item 3
	8.3 - Informa ao usuário que não foi possível cadastrar no banco de dados
Restrições/Validações	

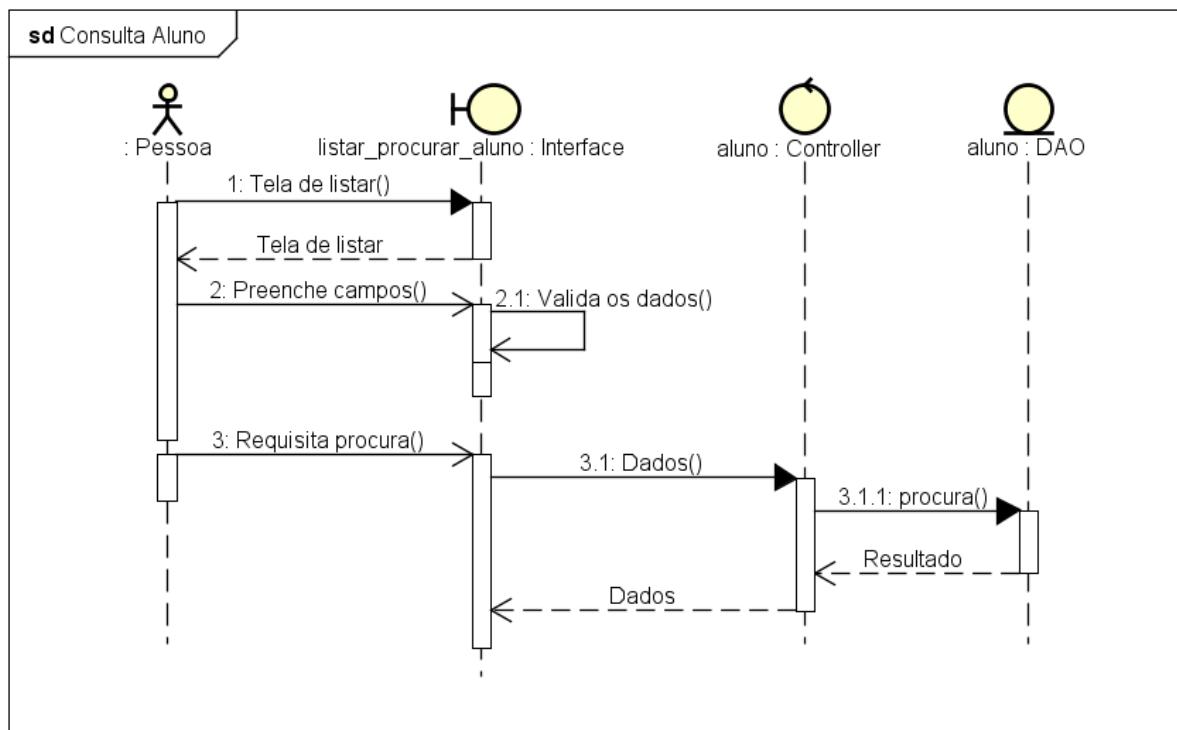
Fonte: Elaborado pelo autor (2023).

A especificação do caso de uso 21 faz referência aos seguintes casos de uso: Pessoa Cadastra AlunoDisciplina, Pessoa Cadastra Historico, Pessoa Cadastra Pessoa, Pessoa Cadastra Presenca, Pessoa Cadastra Grupo, Pessoa Cadastra Aluno, Pessoa Cadastra Ano, Pessoa Cadastra SemestreAno, Pessoa Cadastra Aula, Pessoa Cadastra DisciplinaAula, Pessoa Cadastra Semestre, Pessoa Cadastra Nota, Pessoa Cadastra Disciplina e Pessoa Cadastra AlunoGrupo.

5.3.1.4 Consulta Aluno

O diagrama de sequência da função “Consulta Aluno” pode ser vista na figura 36 e a documentação do mesmo pode ser visto no quadro 22.

Figura 36 – Diagrama de sequência Consulta Aluno



Fonte: elaborado pelo autor (2023).

Quadro 22 – Documentação do caso de uso Consulta Aluno.

Documentação	
Nome do caso de uso	Consulta Aluno
Caso de uso geral	
Autor principal	Pessoa
Atores secundários	
Resumo	Descreve o fluxo percorrido para a consulta do aluno
Pré condições	Estar logado e ter alunos cadastrados
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de procurar/listar dos alunos	
	2 - Redireciona para a tela de procurar/listar dos alunos
3 - Preenche campo de busca	
	4 - Válida os dados
5 - Requisita a procura do aluno	
	6 - Envia os dados para o servidor
	7 - Procura no banco de dados
	8 - Retorna mensagem de sucesso da consulta
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	4.1 - Informa ao usuário que os valores estão em desacordo com as regras do campo
	6.1 - Redireciona para a tela de erro
	8.1 - Retorna que não foi possível procurar no banco de dados
	8.2 - Retorna ao item 2
	8.3 - Informa ao usuário que não foi possível procurar no banco de dados
Restrições/Validações	

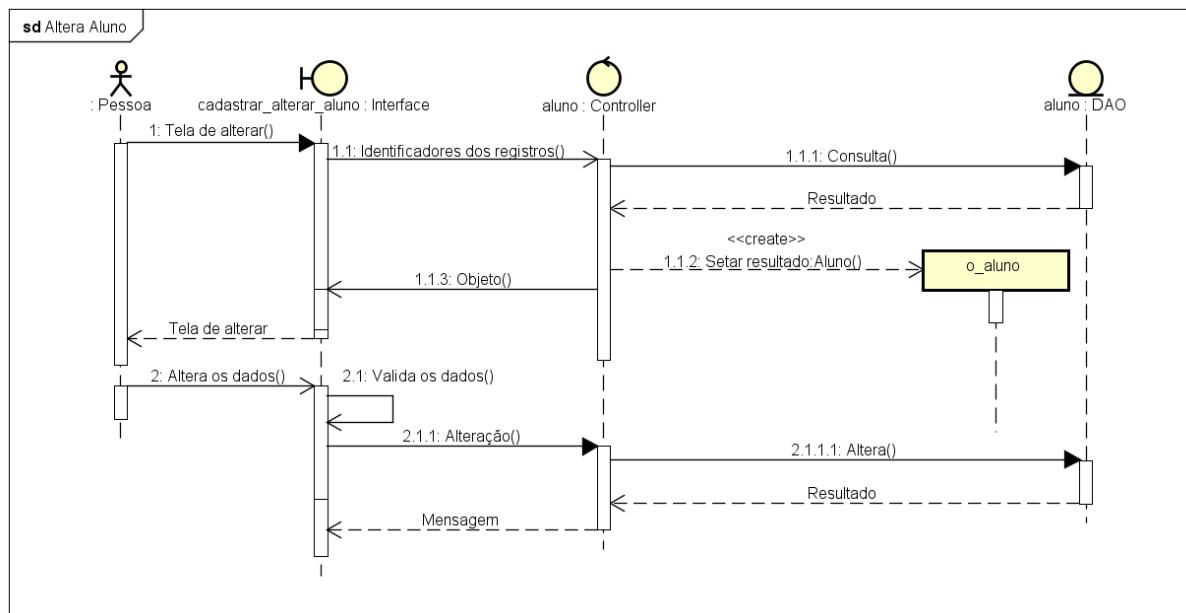
Fonte: Elaborado pelo autor (2023).

A especificação do caso de uso 22 faz referência aos seguintes casos de uso: Pessoa Consulta AlunoDisciplina, Pessoa Consulta Historico, Pessoa Consulta Pessoa, Pessoa Consulta Presenca, Pessoa Consulta Grupo, Pessoa Consulta Aluno, Pessoa Consulta Ano, Pessoa Consulta SemestreAno, Pessoa Consulta Aula, Pessoa Consulta DisciplinaAula, Pessoa Consulta Semestre, Pessoa Consulta Nota, Pessoa Consulta Disciplina e Pessoa Consulta AlunoGrupo.

5.3.1.5 Altera Aluno

O diagrama de sequência da função “Altera Aluno” pode ser vista na figura 37 e a documentação do mesmo pode ser visto no quadro 23.

Figura 37 – Diagrama de sequência Altera Aluno



Fonte: elaborado pelo autor (2023).

Quadro 23 – Documentação do caso de uso Altera Aluno.

Documentação	
Nome do caso de uso	Altera Aluno
Caso de uso geral	Pessoa
Autor principal	
Atores secundários	
Resumo	Descreve o fluxo percorrido para alterar os dados do aluno
Pré condições	Estar logado e ter alunos cadastrados
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de alterar/cadastrar do aluno	
	2 - Redireciona para a tela de alterar/cadastrar do aluno
3 - Altera os dados dos campos	
4 - Requisita a alteração	
	5 - Válida os dados
	6 - Envia os dados para o servidor
	7 - Atualiza os dados no banco de dados
	8 - Retorna mensagem de sucesso da alteração
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	6.1 - Informa ao usuário que os valores estão em desacordo com as regras do campo
	7.1 - Redireciona para a tela de erro
	9.1 - Retorna que não foi possível atualizar os dados no banco de dados
	9.2 - Retorna ao item 3
	9.3 - Informa ao usuário que não foi possível atualizar os dados no banco de dados
Restrições/Validações	

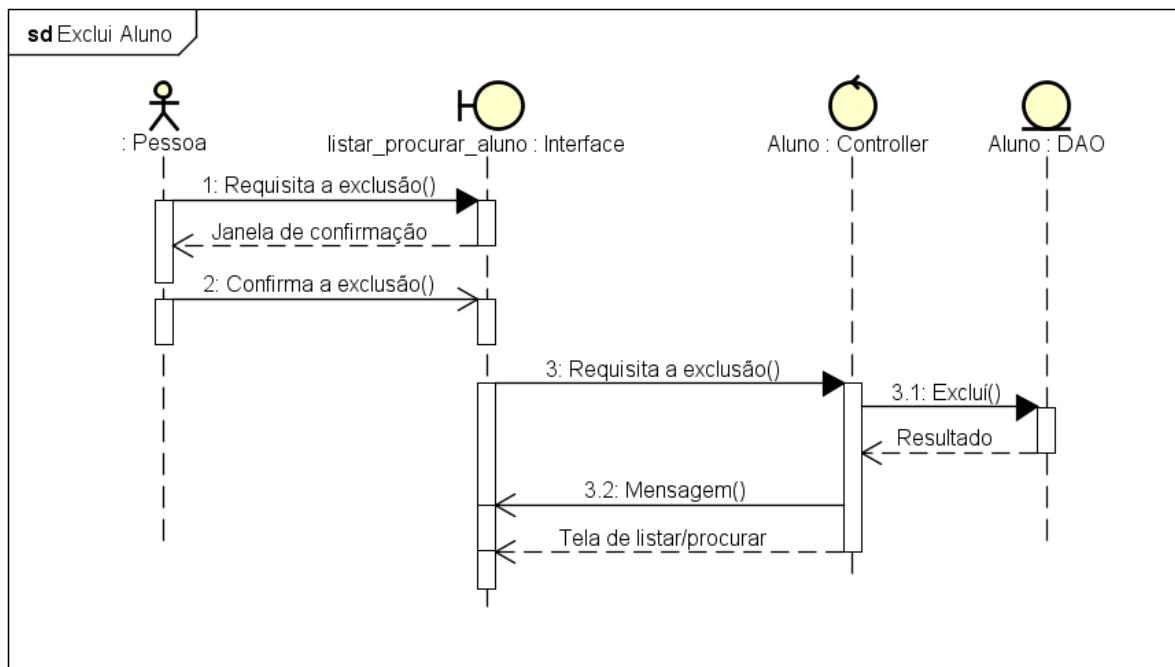
Fonte: Elaborado pelo autor (2023).

A especificação do caso de uso [23](#) faz referência aos seguintes casos de uso: Pessoa Altera AlunoDisciplina, Pessoa Altera Historico, Pessoa Altera Pessoa, Pessoa Altera Presenca, Pessoa Altera Grupo, Pessoa Altera Aluno, Pessoa Altera Ano, Pessoa Altera SemestreAno, Pessoa Altera Aula, Pessoa Altera DisciplinaAula, Pessoa Altera Semestre, Pessoa Altera Nota, Pessoa Altera Disciplina e Pessoa Altera AlunoGrupo.

5.3.1.6 Exclui Aluno

O diagrama de sequência da função “Exclui Aluno” pode ser vista na figura 38 e a documentação do mesmo pode ser visto no quadro 24.

Figura 38 – Diagrama de sequência Exclui Aluno



Fonte: elaborado pelo autor (2023).

Quadro 24 – Documentação do caso de uso Exclui Aluno.

Documentação	
Nome do caso de uso	Exclui Aluno
Caso de uso geral	
Autor principal	Pessoa
Atores secundários	
Resumo	Descreve o fluxo percorrido para excluir um aluno
Pré condições	Estar logado e ter alunos cadastrados
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
2 - Requisita a exclusão	
	3 - Mostra janela de confirmação de exclusão
4 - Confirma a exclusão	
	5 - Envia a requisição para o servidor
	6 - Exclui no banco de dados
	7 - Retorna mensagem de sucesso da exclusão
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	7.1 - Redireciona para a tela de erro
	9.1 - Retorna que não foi possível excluir no banco de dados
	9.2 - Retorna ao item 3
	9.3 - Informa ao usuário que não foi possível excluir no banco de dados
Restrições/Validações	

Fonte: Elaborado pelo autor (2023).

A especificação do caso de uso [24](#) faz referência aos seguintes casos de uso: Pessoa Exclui AlunoDisciplina, Pessoa Exclui Historico, Pessoa Exclui Pessoa, Pessoa Exclui Presenca, Pessoa Exclui Grupo, Pessoa Exclui Aluno, Pessoa Exclui Ano, Pessoa Exclui SemestreAno, Pessoa Exclui Aula, Pessoa Exclui DisciplinaAula, Pessoa Exclui Semestre, Pessoa Exclui Nota, Pessoa Exclui Disciplina e Pessoa Exclui AlunoGrupo.

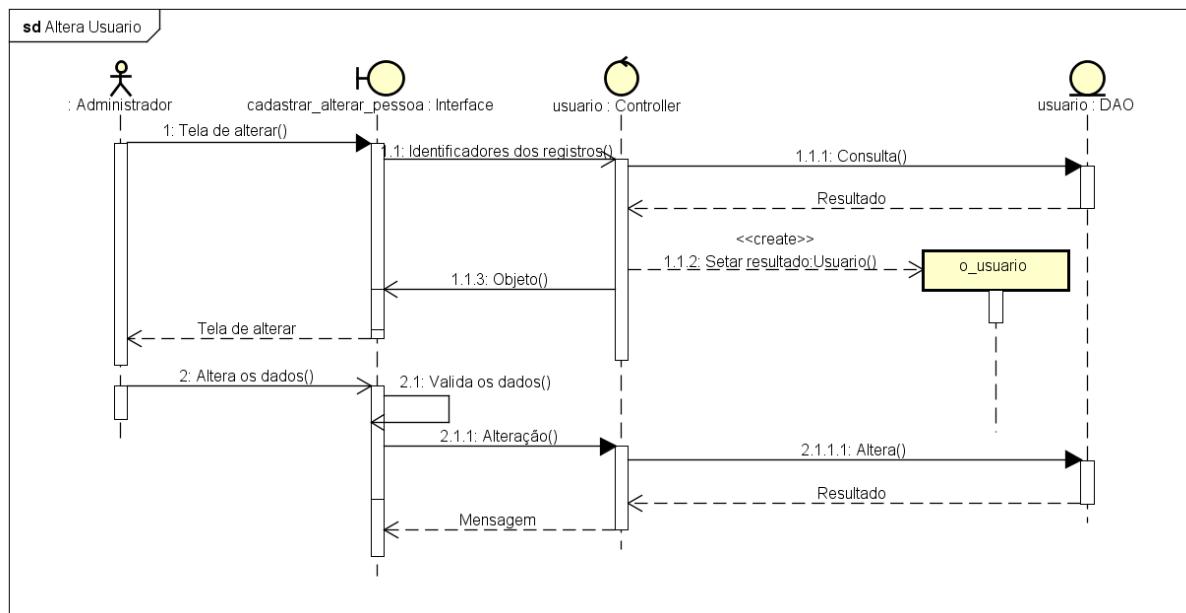
5.3.2 Administrador

Esta seção engloba os diagramas de sequência dos casos de uso do ator *Administrador* e a documentação dos diagramas de sequência.

5.3.2.1 Altera Usuário

O diagrama de sequência da função “Altera Usuário” pode ser vista na figura 39 e a documentação do mesmo pode ser visto no quadro 25.

Figura 39 – Diagrama de sequência Altera Usuário



Fonte: elaborado pelo autor (2023).

Quadro 25 – Documentação do caso de uso Altera Usuário.

Documentação	
Nome do caso de uso	Altera Usuário
Caso de uso geral	
Autor principal	Administrador
Atores secundários	
Resumo	Descreve o fluxo percorrido para alterar os dados do usuário
Pré condições	Estar logado, ser administrador e ter usuários cadastrados
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de alterar/cadastrar dos usuários	
	2 - Redireciona para a tela de alterar/cadastrar do usuário
3 - Altera os dados dos campos	
4 - Requisita a alteração	
	5 - Válida os dados
	6 - Envia os dados para o servidor
	7 - Atualiza os dados no banco de dados
	8 - Retorna mensagem de sucesso da alteração
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	6.1 - Informa ao usuário que os valores estão em desacordo com as regras do campo
	7.1 - Redireciona para a tela de erro
	9.1 - Retorna que não foi possível atualizar os dados no banco de dados
	9.2 - Retorna ao item 3
	9.3 - Informa ao usuário que não foi possível atualizar os dados no banco de dados
Restrições/Validações	

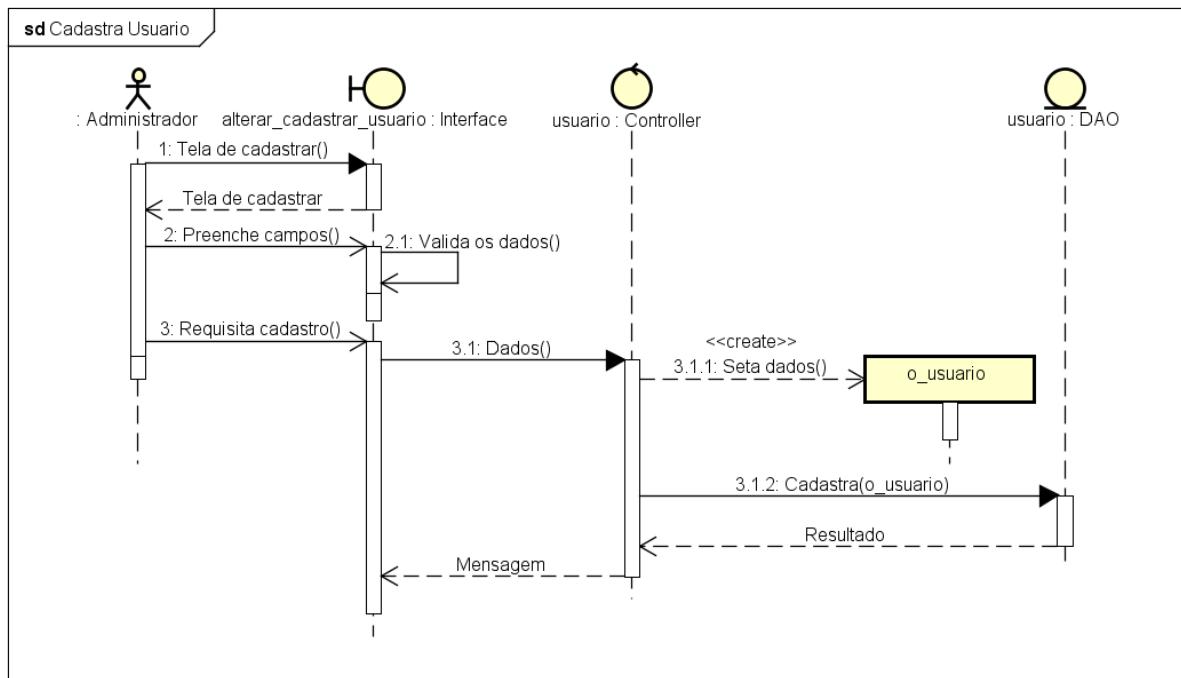
Fonte: Elaborado pelo autor (2023).

A especificação do caso de uso 25 faz referência aos seguintes casos de uso: Administrador Altera Administrador

5.3.2.2 Cadastra Usuário

O diagrama de sequência da função “Cadastra Usuário” pode ser vista na figura 40 e a documentação do mesmo pode ser visto no quadro 26.

Figura 40 – Diagrama de sequência Cadastra Usuário



Fonte: elaborado pelo autor (2023).

Quadro 26 – Documentação do caso de uso Cadastra Usuário.

Documentação	
Nome do caso de uso	Cadastra Usuário
Caso de uso geral	
Autor principal	Administrador
Atores secundários	
Resumo	Descreve o fluxo percorrido para o cadastro de usuário
Pré condições	Estar logado e ser um administrador
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de alterar/cadastrar do usuário	
	2 - Redireciona para a tela de alterar/cadastrar do usuário
3 - Preenche campos	
	4 - Válida os dados
5 - Requisita o cadastro	
	6 - Envia os dados para o servidor
	7 - Cadastra no banco de dados
	8 - Retorna mensagem de sucesso do cadastro
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	4.1 - Informa ao usuário que os valores estão em desacordo com as regras dos campos
	6.1 - Redireciona para a tela de erro
	8.1 - Retorna que não foi possível cadastrar no banco de dados
	8.2 - Retorna ao item 3
	8.3 - Informa ao usuário que não foi possível cadastrar no banco de dados
Restrições/Validações	

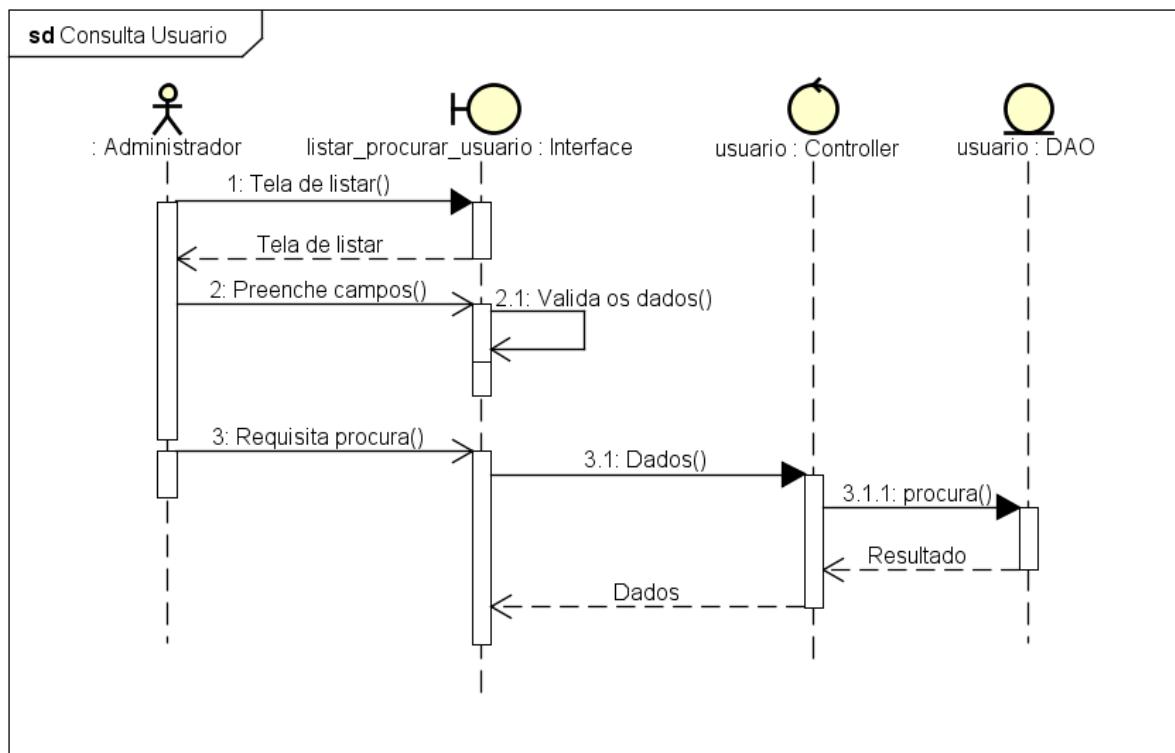
Fonte: Elaborado pelo autor (2023).

A especificação do caso de uso 26 faz referência aos seguintes casos de uso: Administrador Cadastra Administrador

5.3.2.3 Consulta Usuário

O diagrama de sequência da função “Consulta Usuário” pode ser vista na figura 41 e a documentação do mesmo pode ser visto no quadro 27.

Figura 41 – Diagrama de sequência Consulta Usuário



Fonte: elaborado pelo autor (2023).

Quadro 27 – Documentação do caso de uso Consulta Usuário.

Documentação	
Nome do caso de uso	Consulta Usuário
Caso de uso geral	
Autor principal	Administrador
Atores secundários	
Resumo	Descreve o fluxo percorrido para a consulta de usuários
Pré condições	Estar logado e ter usuários cadastrados
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de procurar/listar os usuários	
	2 - Redireciona para a tela de procurar/listar os usuários
3 - Preenche campo de busca	
	4 - Válida os dados
5 - Requisita a procura	
	6 - Envia os dados para o servidor
	7 - Procura no banco de dados
	8 - Retorna mensagem de sucesso da consulta
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	4.1 - Informa ao usuário que os valores estão em desacordo com as regras do campo
	6.1 - Redireciona para a tela de erro
	8.1 - Retorna que não foi possível procurar no banco de dados
	8.2 - Retorna ao item 2
	8.3 - Informa ao usuário que não foi possível procurar no banco de dados
Restrições/Validações	

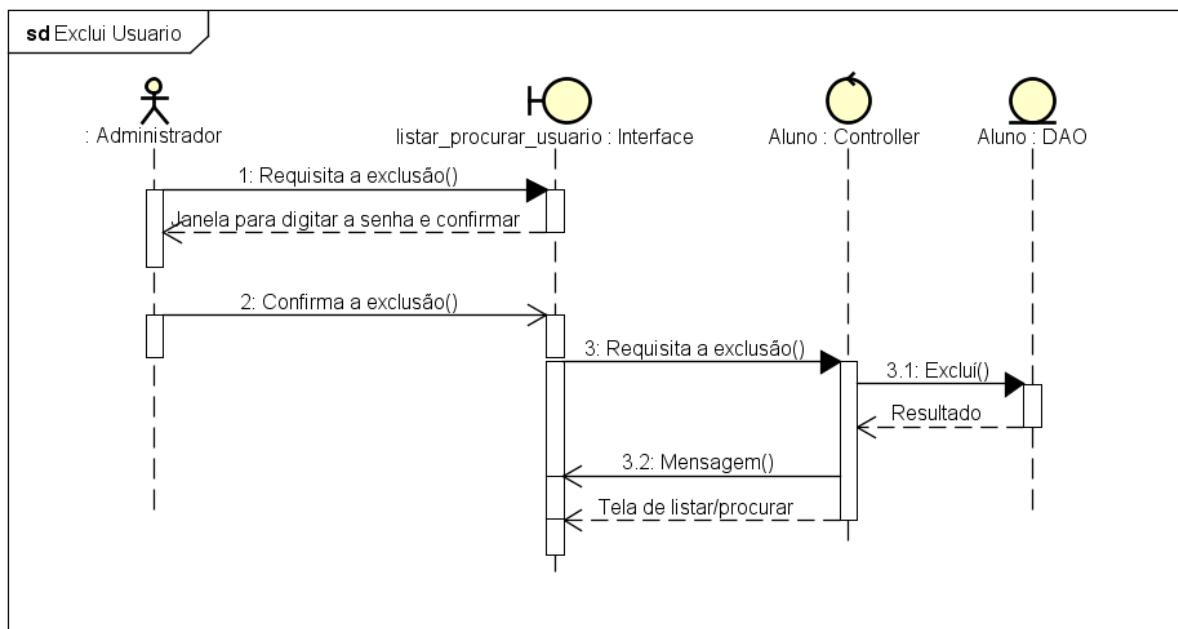
Fonte: Elaborado pelo autor (2023).

A especificação do caso de uso 27 faz referência aos seguintes casos de uso: Administrador Consulta Administrador

5.3.2.4 Exclui Usuário

O diagrama de sequência da função “Exclui Usuário” pode ser vista na figura 42 e a documentação do mesmo pode ser visto no quadro 28.

Figura 42 – Diagrama de sequência Exclui Usuário



Fonte: elaborado pelo autor (2023).

Quadro 28 – Documentação do caso de uso Exclui Usuário.

Documentação	
Nome do caso de uso	Exclui Usuário
Caso de uso geral	
Autor principal	Administrador
Atores secundários	
Resumo	Descreve o fluxo percorrido para excluir usuário
Pré condições	Estar logado e ter usuários cadastrados
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Caso de uso Lista Usuário	
	2 - Redireciona para a tela de procurar/listar os usuários
3 - Requisita a exclusão do usuário	
	4 - Mostra janela de confirmação de exclusão e digitar senha
5 - Confirma a exclusão	
	6 - Envia a requisição para o servidor
	7 - Exclui no banco de dados
	8 - Retorna mensagem de sucesso da exclusão
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	7.1 - Redireciona para a tela de erro
	9.1 - Retorna que não foi possível excluir no banco de dados
	9.2 - Retorna ao item 3
	9.3 - Informa ao usuário que não foi possível excluir no banco de dados
Restrições/Validações	

Fonte: Elaborado pelo autor (2023).

A especificação do caso de uso 28 faz referência aos seguintes casos de uso: Administrador Exclui Administrador

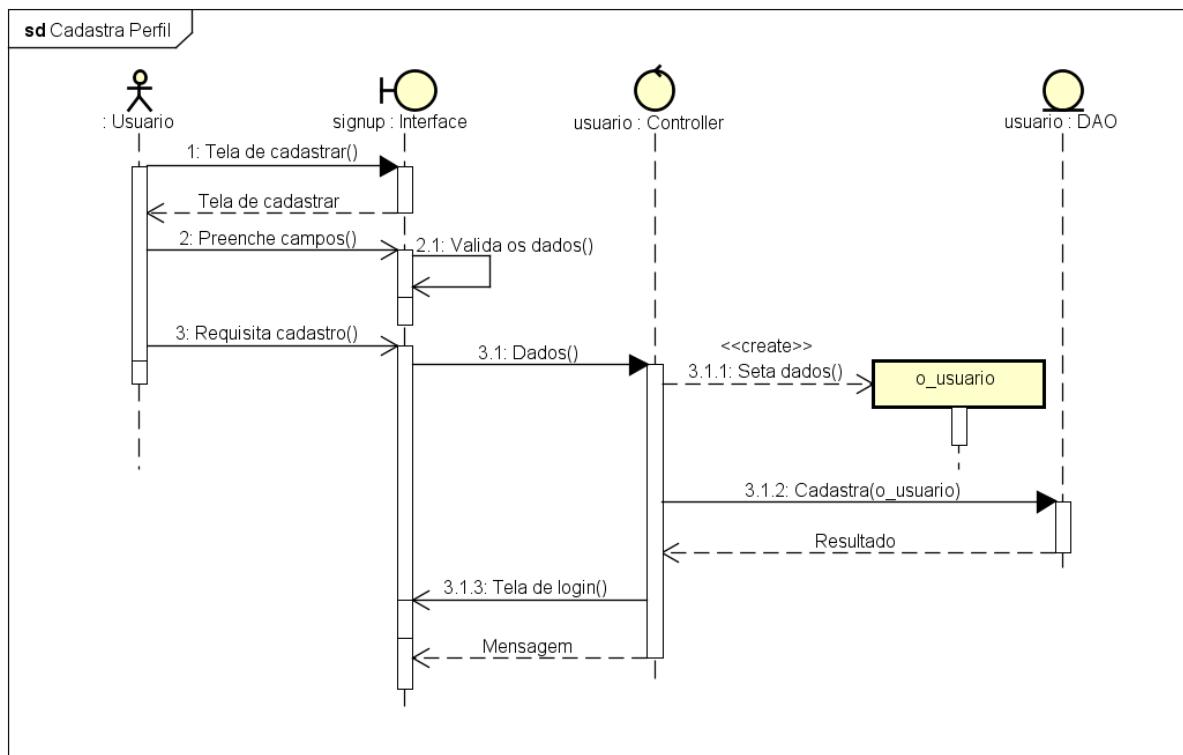
5.3.3 Usuário

Esta seção engloba os diagramas de sequência dos casos de uso do ator *Usuário* e a documentação dos diagramas de sequência.

5.3.3.1 Cadastra Perfil

O diagrama de sequência da função “Cadastra Perfil” pode ser vista na figura 43 e a documentação do mesmo pode ser visto no quadro 29.

Figura 43 – Diagrama de sequência Cadastra Perfil



Fonte: elaborado pelo autor (2023).

Quadro 29 – Documentação do caso de uso Cadastra Perfil.

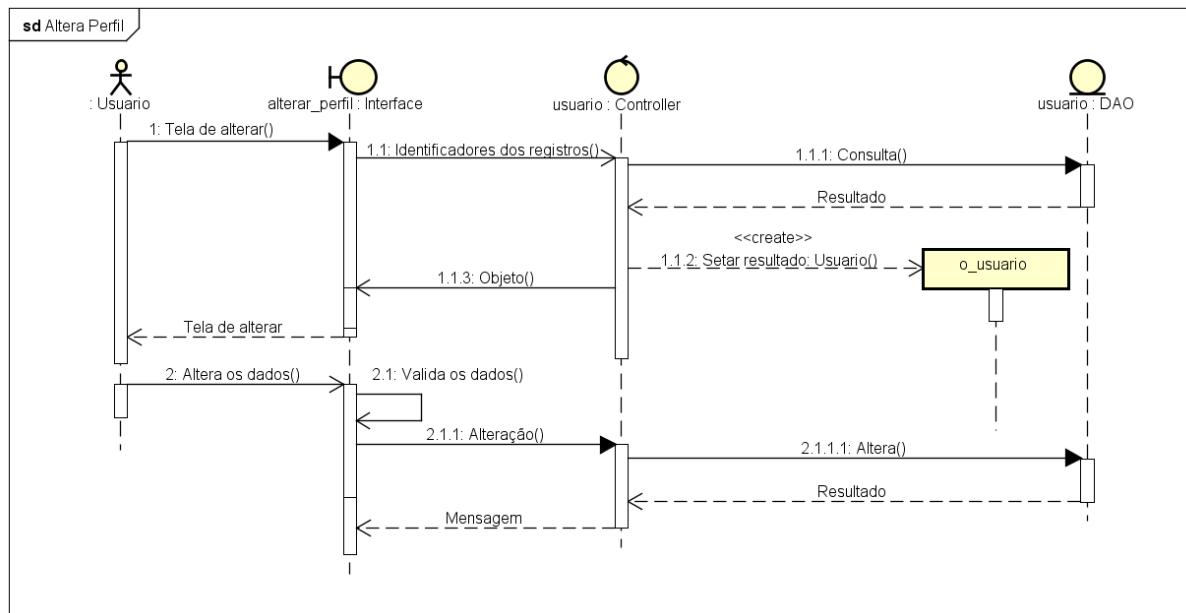
Documentação	
Nome do caso de uso	Cadastra Perfil
Caso de uso geral	
Autor principal	Usuário
Atores secundários	
Resumo	Descreve o fluxo percorrido para o cadastro de um perfil
Pré condições	Não ter uma conta naquele e-mail
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de cadastro de perfil	
	2 - Redireciona para a tela de cadastro de perfil
3 - Preenche campos	
	4 - Válida os dados
5 - Requisita o cadastro	
	6 - Envia os dados para o servidor
	7 - Cadastra no banco de dados
	8 - Retorna mensagem de sucesso do cadastro
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	4.1 - Informa ao usuário que os valores estão em desacordo com as regras dos campos
	6.1 - Redireciona para a tela de erro
	8.1 - Retorna que não foi possível cadastrar no banco de dados
	8.2 - Retorna ao item 3
	8.3 - Informa ao usuário que não foi possível cadastrar no banco de dados
Restrições/Validações	

Fonte: Elaborado pelo autor (2023).

5.3.3.2 Altera Perfil

O diagrama de sequência da função “Altera Perfil” pode ser vista na figura 44 e a documentação do mesmo pode ser visto no quadro 30.

Figura 44 – Diagrama de sequência Altera Perfil



Fonte: elaborado pelo autor (2023).

Quadro 30 – Documentação do caso de uso Altera Perfil.

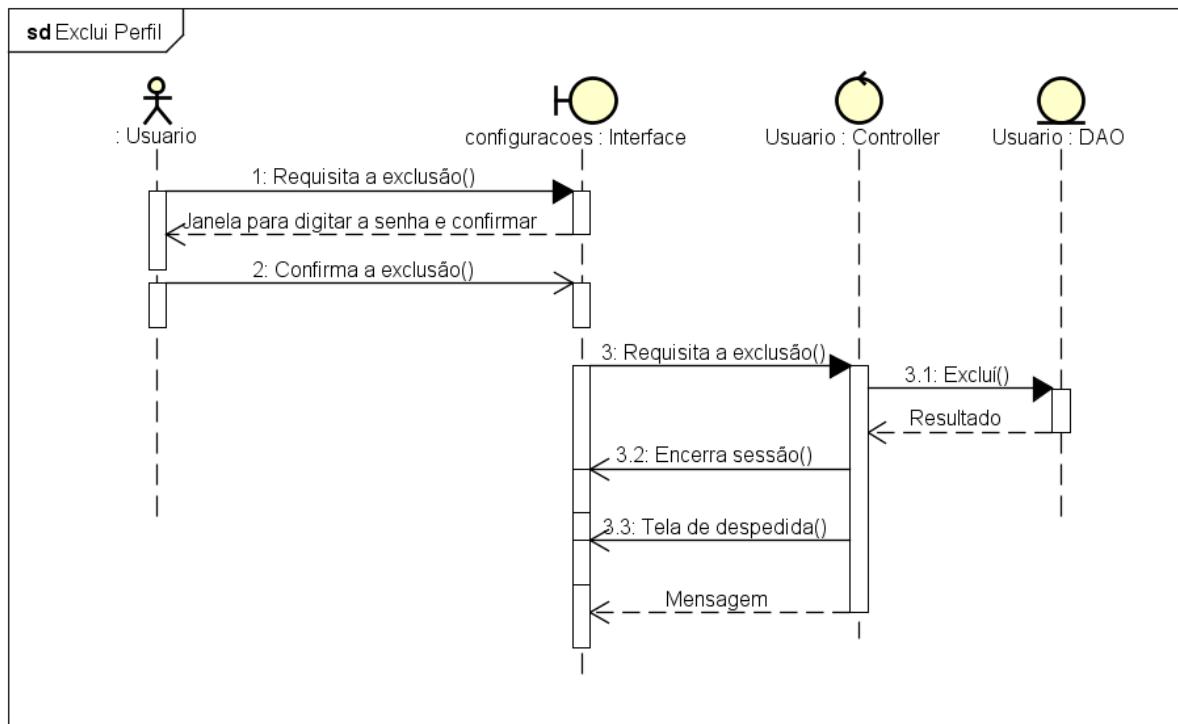
Documentação	
Nome do caso de uso	Altera Perfil
Caso de uso geral	
Autor principal	
Atores secundários	
Resumo	Descreve o fluxo percorrido para alterar os dados do perfil
Pré condições	Estar logado
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de alterar do perfil	
	2 - Redireciona para a tela de alterar do perfil
3 - Altera os dados dos campos	
4 - Requisita a alteração	
	5 - Válida os dados
	6 - Envia os dados para o servidor
	7 - Atualiza os dados no banco de dados
	8 - Retorna mensagem de sucesso da alteração
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	6.1 - Informa ao usuário que os valores estão em desacordo com as regras do campo
	7.1 - Redireciona para a tela de erro
	9.1 - Retorna que não foi possível atualizar os dados no banco de dados
	9.2 - Retorna ao item 3
	9.3 - Informa ao usuário que não foi possível atualizar os dados no banco de dados
Restrições/Validações	

Fonte: Elaborado pelo autor (2023).

5.3.3.3 Exclui Perfil

O diagrama de sequência da função “Exclui Perfil” pode ser vista na figura 45 e a documentação do mesmo pode ser visto no quadro 31.

Figura 45 – Diagrama de sequência Exclui Perfil



Fonte: elaborado pelo autor (2023).

Quadro 31 – Documentação do caso de uso Exclui Perfil.

Documentação	
Nome do caso de uso	Exclui Perfil
Caso de uso geral	
Autor principal	Usuário
Atores secundários	
Resumo	Descreve o fluxo percorrido para excluir o perfil
Pré condições	Estar logado
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a exclusão do perfil	
	2 - Mostra janela de confirmação
3 - Confirma a exclusão	
	4 - Envia a requisição para o servidor
	5 - Exclui no banco de dados
	6 - Retorna mensagem de sucesso da exclusão
	7 - Encerra a sessão
	8 - redireciona para a página de despedida
	9 - Constrói a página de despedida
Fluxo Alternativo	
Ações do autor	Ações do sistema
	4.1 - Redireciona para a tela de erro
	6.1 - Retorna que não foi possível excluir no banco de dados
	6.2 - Retorna ao item 3
	6.3 - Informa ao usuário que não foi possível excluir no banco de dados
	8.1 - Redireciona para a tela de erro
Restrições/Validações	

Fonte: Elaborado pelo autor (2023).

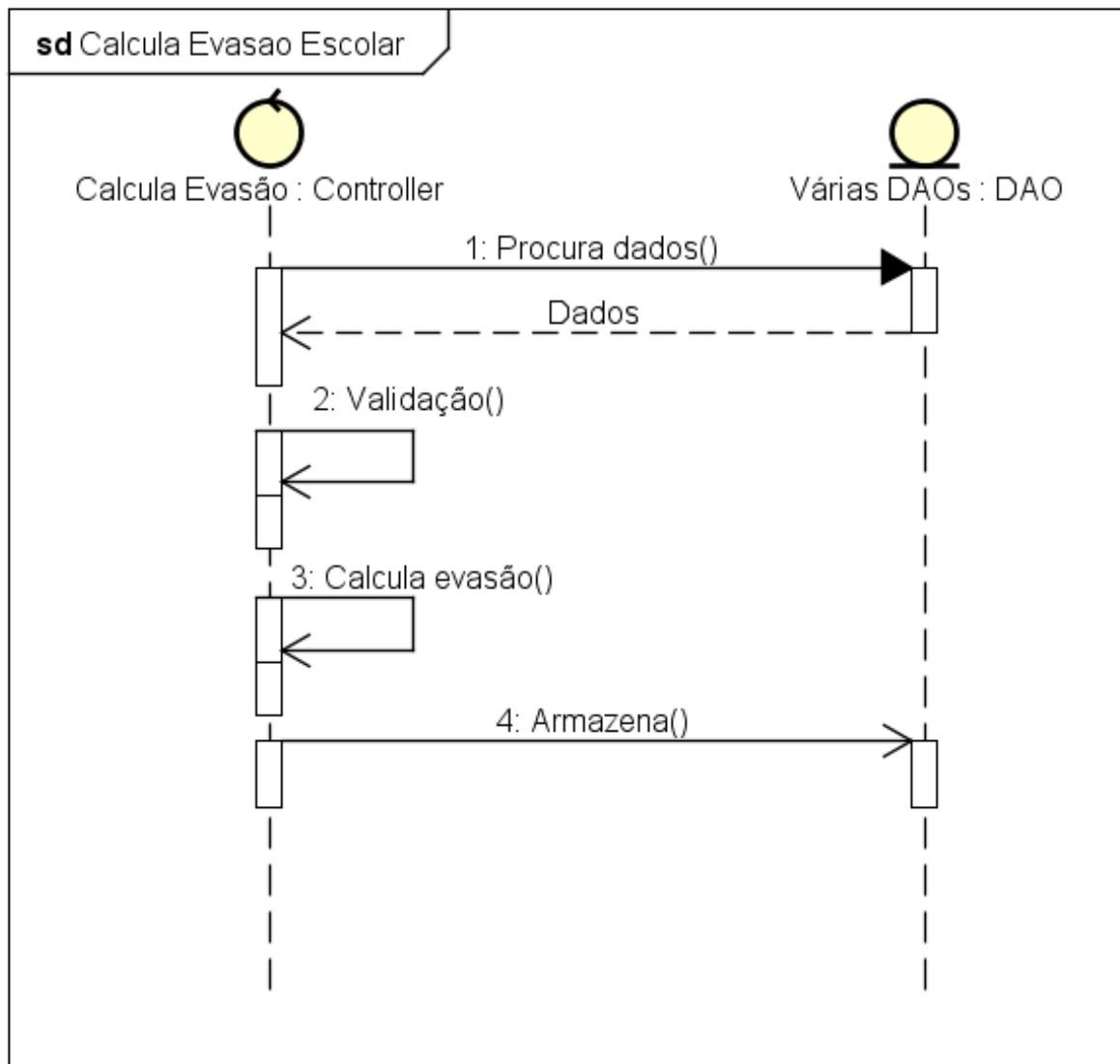
5.3.4 IA

Está seção engloba os diagramas de sequência dos casos de uso do ator *IA* e a documentação dos diagramas de sequência.

5.3.4.1 Calcula Evasão Escolar

O diagrama de sequência da função “Calcula Evasão Escolar” pode ser vista na figura 46 e a documentação do mesmo pode ser visto no quadro 32.

Figura 46 – Diagrama de sequência Calcula Evasão Escolar



Fonte: elaborado pelo autor (2023).

Quadro 32 – Documentação do caso de uso Calcula Evasão Escolar.

Documentação	
Nome do caso de uso	Calcula Evasão Escolar
Caso de uso geral	
Autor principal	IA
Atores secundários	
Resumo	Descreve o fluxo percorrido para o cálculo da evasão escolar
Pré condições	Ter dados relacionados aos alunos
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
	1 - Requisita os dados relacionados aos alunos
	2 - Retorna uma lista dos dados relacionados aos alunos
	3 - Verifica se existem dados relacionados aos alunos
	4 - Gera dados da evasão escolar
	5 - Armazena os dados
Fluxo Alternativo	
Ações do autor	Ações do sistema
	3.1 - Cancela a operação
Restrições/Validações	

Fonte: Elaborado pelo autor (2023).

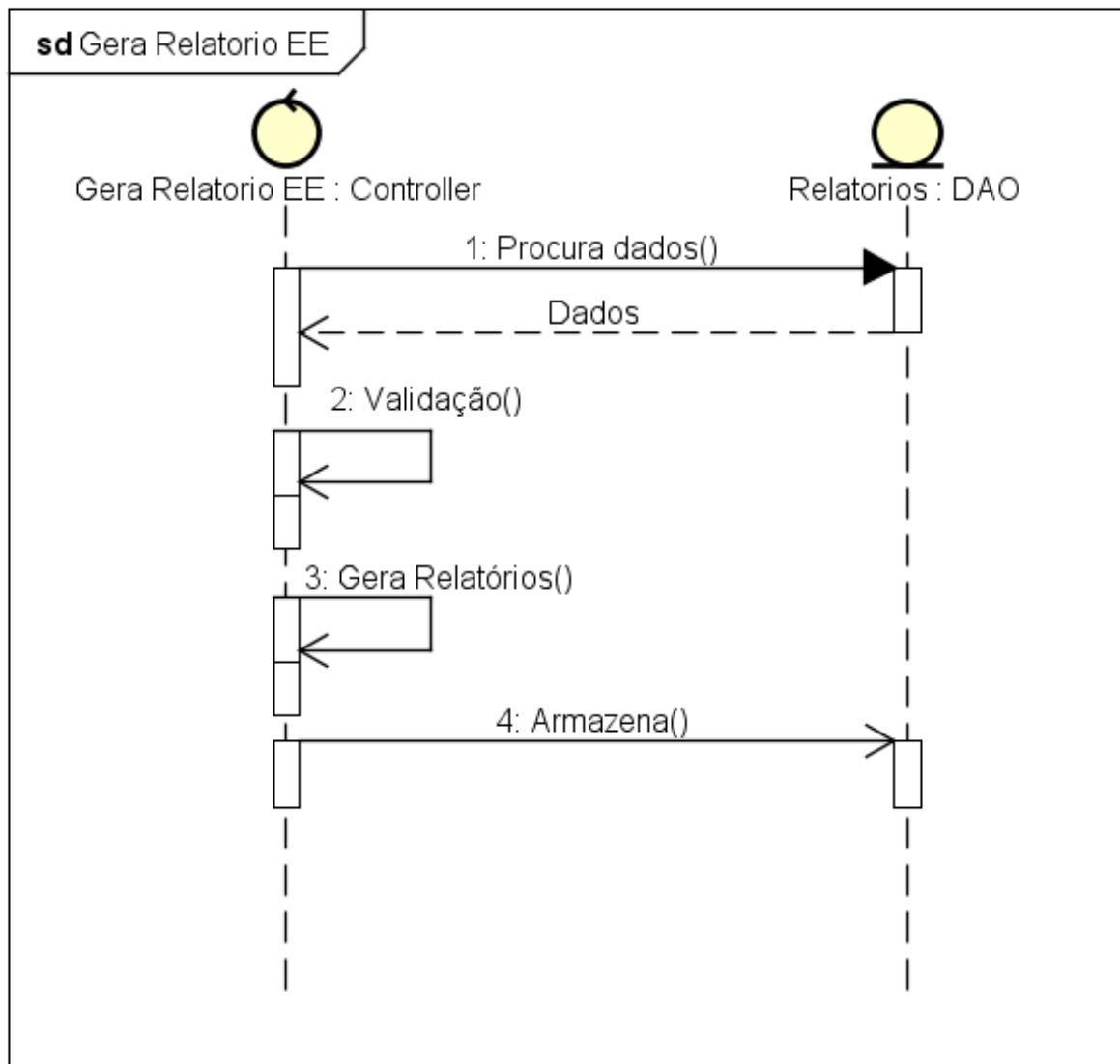
5.3.5 Relatório

Esta seção engloba os diagramas de sequência dos casos de uso do ator *Relatório* e a documentação dos diagramas de sequência.

5.3.5.1 Gera Relatório EE

O diagrama de sequência da função “Gera Relatório EE” pode ser vista na figura 47 e a documentação do mesmo pode ser visto no quadro 33.

Figura 47 – Diagrama de sequência Gera Relatório EE



Fonte: elaborado pelo autor (2023).

Quadro 33 – Documentação do caso de uso Gera Relatório EE.

Documentação	
Nome do caso de uso	
Caso de uso geral	
Autor principal	
Atores secundários	
Resumo	
Pré condições	
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
	1 - Requisita os dados relacionados aos alunos
	2 - Retorna uma lista dos dados relacionados aos alunos
	3 - Verifica se existem dados relacionados aos alunos
	4 - Gera dados da evasão escolar
	5 - Armazena os dados
Fluxo Alternativo	
Ações do autor	Ações do sistema
	3.1 - Cancela a operação
Restrições/Validações	

Fonte: Elaborado pelo autor (2023).

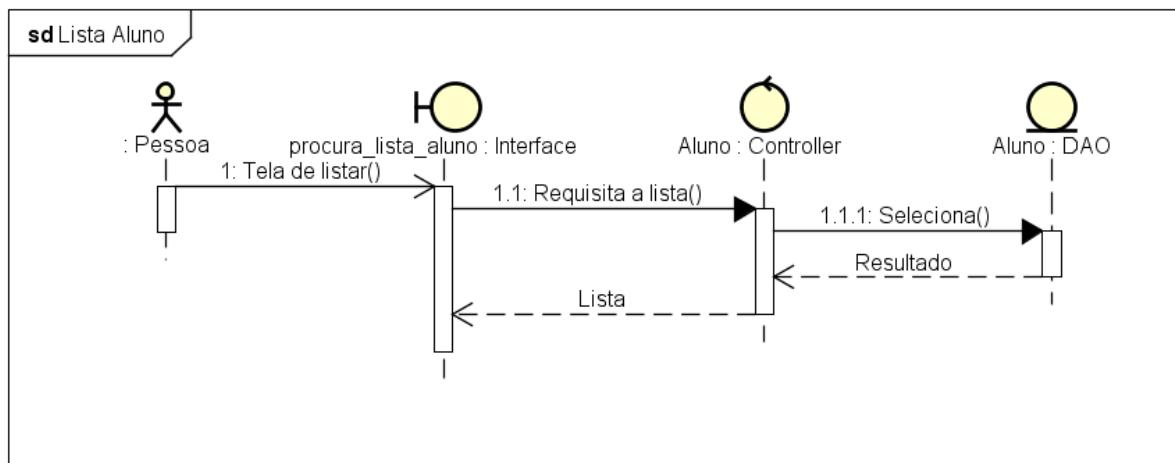
5.3.6 Sistema

Está seção engloba os diagramas de sequência dos casos de uso do ator *Sistema* e a documentação dos diagramas de sequência.

5.3.6.1 Lista Aluno

O diagrama de sequência da função “*Lista Aluno*” pode ser vista na figura 48 e a documentação do mesmo pode ser visto no quadro 34.

Figura 48 – Diagrama de sequência Lista Aluno



Fonte: elaborado pelo autor (2023).

Quadro 34 – Documentação do caso de uso Lista Aluno.

Documentação	
Nome do caso de uso	Lista Aluno
Caso de uso geral	
Autor principal	Pessoa
Atores secundários	
Resumo	Descreve o fluxo percorrido para listar os alunos
Pré condições	Estar logado e ter alunos cadastrados
Pós condições	
Fluxo Normal	
Ações do autor	Ações do sistema
1 - Requisita a tela de procurar/listar dos alunos	
	2 - Envia a requisição para o servidor
	3 - Procura no banco de dados
	4 - Retorna uma lista de alunos
	5 - Redireciona para a tela de procurar/listar dos alunos
Fluxo Alternativo	
Ações do autor	Ações do sistema
	2.1 - Redireciona para a tela de erro
	3.1 - Retorna que não foi possível procurar no banco de dados
	3.2 - Retorna ao item 3
	3.3 - Informa ao usuário que não foi possível procurar no banco de dados
	5.1 - Redireciona para a tela de erro
Restrições/Validações	

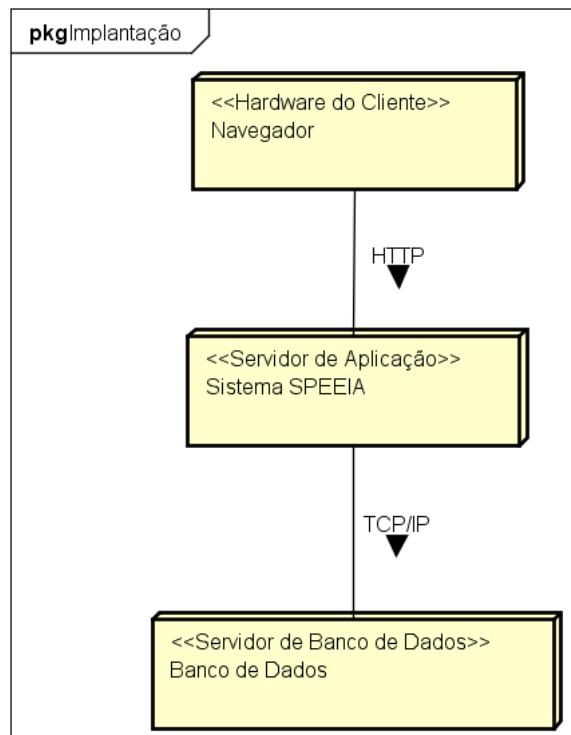
Fonte: Elaborado pelo autor (2023).

A especificação do caso de uso 34 faz referência aos seguintes casos de uso: Sistema Lista AlunoDisciplina, Sistema Lista Historico, Sistema Lista Pessoa, Sistema Lista Presenca, Sistema Lista Grupo, Sistema Lista Ano, Sistema Lista Usuario, Sistema Lista Aula, Sistema Lista Nota, Sistema Lista Administrador, Sistema Lista SemestreAno, Sistema Lista DisciplinaAula, Sistema Lista Semestre, Sistema Lista Disciplina e Sistema Lista AlunoGrupo.

5.4 DIAGRAMA DE IMPLANTAÇÃO

O diagrama de implantação organiza o *hardware* e os *softwares* utilizados em todo o projeto. Com ele é possível obter uma visualização geral de quais *hardwares* se comunicam entre-si e com quais *softwares* são utilizados dentro desses *hardwares*, além disso, também demonstrado em quais protocolos eles se comunicam (BOOCH; RUMBAUGH; JACOBSON, 2012, p. 538). O diagrama de implantação do software pode ser visto na figura 49.

Figura 49 – Diagrama de implantação



Fonte: elaborado pelo autor (2023).

6 DESIGN

Design é um processo muito importante para a empresa, pois com ela é possível que a empresa atraia mais clientes para conhecer seu produtos. O *design* consiste em criar produtos ou serviços com excelente técnica e estética, afim de agregar valor ao seu projeto Patel (2023). Quando aplicado de forma correta, ele impacta as diversas outras áreas da empresa, vendas, marketing, experiência do usuário e outros Patel (2023).

6.1 NOME

O nome do software se fez por uma lógica simples e que foi conveniente para a ideia do projeto. Utilizamos a primeira letra de cada palavra do título do trabalho na mesma ordem, dai surgiu o nome ‘SPEEIA’. Não intencionalmente, o nome do software dá a ideia de observar, o que podemos entender como observar os alunos. A explicação em forma ilustrativa pode ser vista na figura 50.

Figura 50 – Explicação do nome do software



Fonte: elaborado pelo autor (2023).

6.2 PALETA DE CORES

A escolha da paleta de cores foi baseado no artigo do marqueteiro americano Neil Patel, nesse artigo ele expõe diversos pontos para se escolher as cores de um projeto. O artigo também aborda o por que das cores serem um ponto importante a se analisar ao montar o seu negócio e

como ela influencia na compra de produtos e serviço. A paleta de cores do projeto pode ser vista na figura 51 e o artigo do marqueteiro Neil Pastel pode ser lido no link: [clique aqui](#).

Figura 51 – Paleta de cores

Primary color:	#68BCFD #68BCFD	#3CAAFF #3CAAFF	#1097FF #1097FF	#0392FF #0392FF	#0169BA #0169BA		
Complement color:	#FFBASF #FFBASF	#FFA62E #FFA62E	#FF9200 #FF9200		#FF9200 #FF9200		
Primary color:							
Complement color:							

Fonte: [paletton... \(2023\)](#).

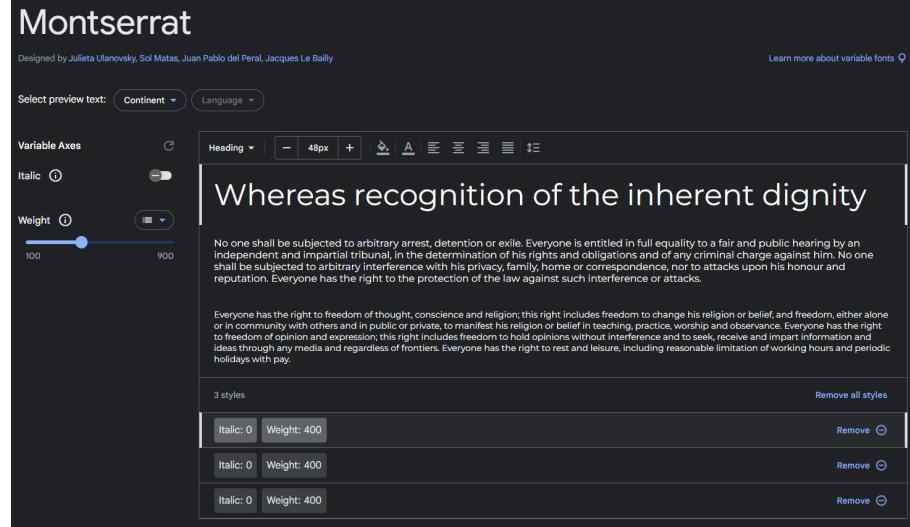
6.3 FONTES

A escolha das fontes utilizadas no projeto foi baseado no artigo da *designer* Midori Nediger, a partir do artigo, foi possível escolher duas fontes, uma para os títulos dos textos e outra para o corpo dos textos. A fonte para o corpo dos textos é a 'Montserrat' e a fonte para os títulos é a 'Josefin Sans'. O exemplo da fonte 'Josefin Sans' pode ser vista na figura 52 e o exemplo da fonte 'Montserrat' pode ser vista na figura 53.

Figura 52 – Exemplo da fonte ‘Josefin Sans’

Fonte: [Josefin... \(2023\)](#).

Figura 53 – Exemplo da fonte ‘Montserrat’

Fonte: [Montserrat \(2023\)](#).

6.4 LOGOS

Normalmente, todo projeto possui uma logo, as logos servem como um símbolo para que as pessoas possam referenciar aquele símbolo a empresa que a divulga. Assim, é possível que as pessoas rapidamente possam se lembrar qual tipo de serviço/produto aquela empresa oferece. A logo do projeto pode ser vista na figura 54 e as variações da mesma pode ser vista na figura 55.

Figura 54 – Logo azul



Fonte: elaborado pelo autor (2023).

Figura 55 – Logo branca



Fonte: elaborado pelo autor (2023).

7 INTELIGÊNCIA ARTIFICIAL

7.1 MODELO

Como mostrado anteriormente, o modelo utilizado para o cálculo da taxa de evasão escolar, é a regressão linear múltipla. Diferente do modelo de regressão simples, que assume que o valor da variável dependente depende apenas de uma variável independente ([LARSON; FARBER, 2015](#), p. 454-455). o modelo de regressão linear múltipla entende que a variável depende de vários fatores. Ou seja, enquanto o primeiro modelo assume um variável determinante, o segundo modelo assume diversas variáveis determinantes ([LARSON; FARBER, 2015](#), p. 475).

Aplicando os modelos ao nosso cenário, no primeiro modelo pode se observar que apenas uma variável poderia influenciar na taxa de evasão, tais como nota, frequência, período, idade e entre outros. Já o segundo modelo assume que diversas características poderia influenciar na taxa de evasão, essas podem ser, nota, frequência, período, idade e outras. Observando a realidade, entende-se que a decisão de evadir se dá por diversos motivos, não somente uma variável, mas um conjunto delas, por isso, o modelo mais adequado entre os dois apresentados é modelo de regressão linear múltipla. Equação de regressão linear múltipla, pode ser vista na equação 1:

Equação 1 – Equação de regressão linear múltipla

$$y = b + m_1x_1 + m_2x_2 + \dots + m_kx_k \quad (7.1)$$

Fonte: [Larson e Farber \(2015](#), pg. 475).

7.2 EQUAÇÃO

Posto a equação acima, iremos adaptar ao nosso modelo, ou seja, definir as variáveis independentes e a variável dependente. Dessa forma é possível visualizar como ficou a estrutura da equação de regressão linear múltipla do nosso modelo. Equação de regressão linear múltipla do nosso modelo, pode ser vista na equação 2:

Equação 2 – Equação de regressão linear múltipla do modelo

$$\begin{aligned}
 TAX_EVASAO = & b + FREQ_A1_M2_1X_1 + FREQ_A1_M3_2X_2 + \\
 & FREQ_A1_M4_3X_3 + FREQ_A1_M5_4X_4 + FREQ_A1_M6_5X_5 + \\
 & FREQ_A1_M7_6X_6 + FREQ_A1_M8_7X_7 + FREQ_A1_M9_8X_8 + \\
 & FREQ_A1_M10_9X_9 + FREQ_A1_M11_{10}X_{10} + FREQ_A1_M12_{11}X_{11} + \\
 & FREQ_A2_M2_{12}X_{12} + FREQ_A2_M3_{13}X_{13} + FREQ_A2_M4_{14}X_{14} + \\
 & FREQ_A2_M5_{15}X_{15} + FREQ_A2_M6_{16}X_{16} + FREQ_A2_M7_{17}X_{17} + \\
 & FREQ_A2_M8_{18}X_{18} + FREQ_A2_M9_{19}X_{19} + FREQ_A2_M10_{20}X_{20} + \\
 & FREQ_A2_M11_{21}X_{21} + FREQ_A2_M12_{22}X_{22} + FREQ_A3_M2_{23}X_{23} + \\
 & FREQ_A3_M3_{24}X_{24} + FREQ_A3_M4_{25}X_{25} + FREQ_A3_M5_{26}X_{26} + \\
 & FREQ_A3_M6_{27}X_{27} + FREQ_A3_M7_{28}X_{28} + FREQ_A3_M8_{29}X_{29} + \\
 & FREQ_A3_M9_{30}X_{30} + FREQ_A3_M10_{31}X_{31} + FREQ_A3_M11_{32}X_{32} + \\
 & FREQ_A3_M12_{33}X_{33} + NOTA_A1_B1_{34}X_{34} + NOTA_A1_B2_{35}X_{35} + \\
 & NOTA_A1_B3_{36}X_{36} + NOTA_A1_B4_{37}X_{37} + NOTA_A2_B1_{38}X_{38} + \\
 & NOTA_A2_B2_{39}X_{39} + NOTA_A2_B3_{40}X_{40} + NOTA_A2_B4_{41}X_{41} + \\
 & NOTA_A3_B1_{42}X_{42} + NOTA_A3_B2_{43}X_{43} + NOTA_A3_B3_{44}X_{44} + \\
 & NOTA_A3_B4_{45}X_{45}
 \end{aligned} \tag{7.2}$$

Fonte: Larson e Farber (2015, pg. 475).

Quadro 35 – Abreviações das variáveis.

Abreviação	Significado
<i>FREQ</i>	FREQUÊNCIA.
<i>A</i>	ANO.
<i>M</i>	MÊS.
<i>M</i>	MÊS.
<i>NOTA</i>	NOTA.
<i>B</i>	BIMESTRE.

Fonte: Elaborado pelo autor (2023).

7.3 ALGORITMO

Como demonstrado, a linguagem de programação escolhida para desenvolver este projeto foi o Python. Além dos pontos mostrados anteriormente para a escolha do mesmo, há outros pontos que se deu a escolha. Esta linguagem é muito poderosa quando se trata de criar algoritmos de inteligência artificial, devido a sua fácil compreensão e velocidade na codificação por ser uma linguagem interpretada.

Para o desenvolvimento do algoritmo de IA, utilizou-se diversas bibliotecas para agilizar o processo de desenvolvimento. Dentre elas, foi utilizado *pandas*, *numpy*, *math* e diversas outras

cada qual com seu propósito. As bibliotecas utilizadas para o desenvolvimento da IA pode ser vista no quadro 36.

Quadro 36 – Bibliotecas utilizadas.

Nome	Descrição
<i>Matplotlib</i>	A biblioteca <i>Matplotlib</i> é voltada para visualização de dados, o que significa que você visualizar gráficos de linhas, gráficos de barras e gráficos de dispersão no <i>Matplotlib</i> e muito outros.
<i>Pandas</i>	O <i>Pandas</i> é uma biblioteca de manipulação e análise de dados, isso providênciça facilidade no momento de mexer, limpando, tratando e fazendo análises sobre eles.
<i>NumPy</i>	<i>NumPy</i> é uma poderosa biblioteca para se trabalhar matrizes, ela é um pouco semelhante ao <i>Pandas</i> , mas o seu foco está nos <i>arrays</i> .
<i>Scikit Learn</i>	É com essa biblioteca as IAs supervisionadas e não supervisionadas brilham, ela disponibiliza diversos modelos para IA, diversas métricas e muitos outros utilitários.
<i>Math</i>	A biblioteca <i>Math</i> permite que você utilize diversas funções para cálculos de operações básicas da matemática. Tais como, potenciação, fatorial, permutação, logarítmica, trigonometria e assim por diante.
<i>Seaborn</i>	<i>Seaborn</i> é outra biblioteca para visualização de dados, normalmente para fazer gráficos mais robustos e com um design melhor.
<i>Scipy</i>	O <i>Scipy</i> é uma biblioteca voltada para matemáticos, cientistas e engenheiros, ajudando a fazer cálculos complexos de suas áreas.

Fonte: Elaborado pelo autor (2023).

O código do nosso modelo pode ser visto nos códigos 17, 18 e 19.

Código 17 – Código da IA: Parte 1

```
1 #INTITULA E DIVIDE O PROCESSO
2 def divisor(title):
3     string = "=====|{}|=====".format(title)
4     len_string = len(string)
5     print("=". * len_string)
6     print(string)
7     print("=". * len_string)
8
9 # IMPORTAÇÕES
10 ## PLOTAR GRÁFICOS
11 from matplotlib import pyplot as plt
12 from scipy.stats import pearsonr
13 import seaborn as sns
14 ## MANIPULAÇÃO DE DADOS
15 import pandas as pd
16 ## MANIPULAÇÃO DE MATRIZES E ARRAYS
17 import numpy as np
18 ## UTILIDADES
19 from sklearn.utils import shuffle
20 ## MODELOS
21 from sklearn import linear_model
22 ## DATASET
23 ### DIVIDIR DATASET
24 from sklearn.model_selection import train_test_split
25 ## MÉTRICAS
26 from sklearn.metrics import r2_score as r2, mean_absolute_error as
    mae, mean_squared_error as mse
27 ## OPERAÇÕES MATEMÁTICAS
28 from math import sqrt
29 ## FUNÇÕES PARA O SO
30 import os
31
32 # DATASET
33 ## CONSEGUIR DADOS
34 path_atual = os.path.dirname(os.path.realpath(__file__))
35 path_file = path_atual + "\\dadosformatados.xlsx"
36
37 divisor("INFORMAÇÕES DATASET")
38 ## CARREGAR O DATASET PARA O DATAFRAME E MOSTRAR INFORMAÇÕES SOBRE O
    DATAFRAME
39 dataframe = pd.read_excel(path_file, "plan2")
40 print(dataframe.head())
41 print(dataframe.describe())
```

Fonte: elaborado pelo autor (2023).

Código 18 – Código da IA: Parte 2

```
1 divisor("FORMATANDO DATASET")
2 ##FORMATANDO DATASET
3 dataframe = shuffle(dataframe, random_state=42)
4 print(dataframe.head())
5
6 #SEPARANDO OS DADOS NÚMERICOS
7 dataframe_numeric = dataframe.drop(['NOME', 'SITUACAO', 'SUM_FREQ', ,
8     'SUM_NOTA', 'PERC_FREQ', 'PREC_NOTA', 'SUM_NOTA_FREQ'], axis=1)
9
10 divisor("SEPARANDO COLUNAS")
11 #SEPARANDO AS COLUNAS INDEPENDENTES
12 x = dataframe.drop(['NOME', 'SITUACAO', 'SUM_FREQ', 'SUM_NOTA', ,
13     'PERC_FREQ', 'PREC_NOTA', 'SUM_NOTA_FREQ', 'TAX_EVASAO'], axis=1)
14 print(x.head())
15
16 #SEPARANDO A COLUNA DEPENDENTE
17 y = dataframe['TAX_EVASAO']
18 print(y.head())
19
20 divisor("DIVIDINDO DATASET")
21 ##DIVIDIR DATASET
22 x_treino, x_teste, y_treino, y_teste = train_test_split(x, y,
23     test_size=0.3, random_state=42)
24
25 #PLOTAR DADOS DE TREINO
26 corr = dataframe_numeric.corr()
27 mask = np.triu(np.ones_like(corr, dtype=np.bool_))
28 f, ax = plt.subplots(figsize=(11,9))
29 cmap = sns.diverging_palette(220, 10, as_cmap=True)
30 sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0, square=
31     True, linewidths=.5, cbar_kws={"shrink": .5})
32 plt.show()
33
34 # TREINAR MODELO
35 modelo = linear_model.LinearRegression()
36 modelo.fit(x_treino, y_treino)
37
38 divisor("INTERCEPTO E COEFICIENTES")
39 print('A Intercepto: ', modelo.intercept_)
40 print('B Inclinação: ', modelo.coef_)
```

Fonte: elaborado pelo autor (2023).

Código 19 – Código da IA: Parte 3

```
1 ## PLOTAR DADOS DE TESTE COM MODELO
2 plt.scatter(y_teste, predicoes, color="blue")
3 #plt.plot(engine_size_teste, modelo.coef_[0][0] * engine_size_teste +
4 #          modelo.intercept_[0], '-g')
4 plt.title("Gráfico dos dados de teste com modelo: {}".format(pearsonr
5             (y_teste, predicoes)[0]))
5 plt.ylabel("Y teste (Valores reais)")
6 plt.xlabel("Predicted vallues")
7 plt.show()
8
9 #MÉTRICAS
10 print("Soma dos Quadrados dos Erros (SSE): %.2f" % np.sum([(predicoes
11 - y_teste) ** 2]))
12 print("Erro QuadrÁtico MÉdio (MSE): %.2f" % mse(y_teste, predicoes))
13 print("Erro Absoluto MÉdio (MAE): %.2f" % mae(y_teste, predicoes))
14 print("Raiz do Erro QuadrÁtico MÉdio (RMSE): %.2f" % sqrt(mse(y_teste
15 , predicoes)))
16 print("R2-score: %.2f" % r2(predicoes, y_teste))
```

Fonte: elaborado pelo autor (2023).

8 RESULTADOS

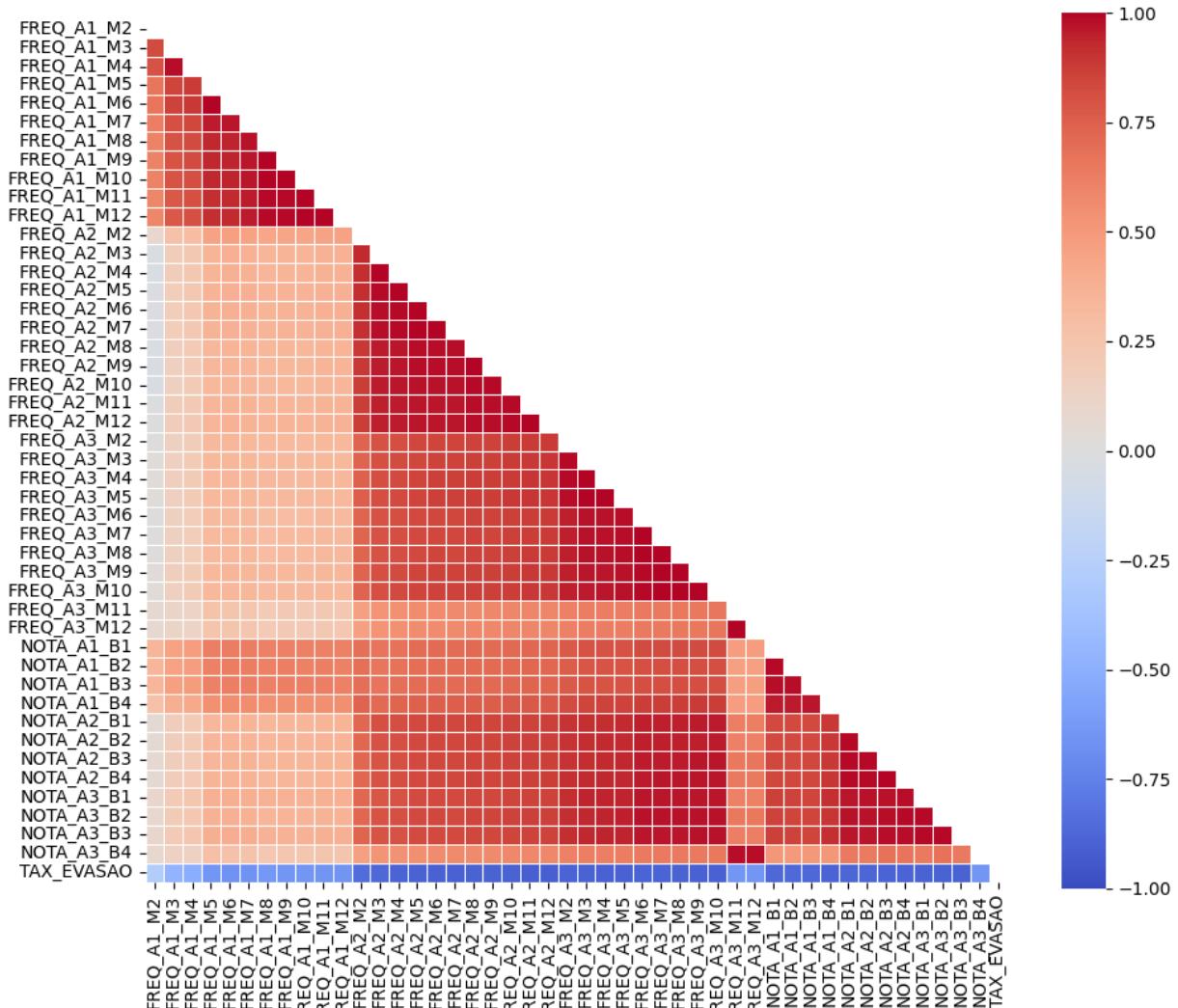
A matriz de correlação de Pearson demonstra a correlação entre as variáveis em si utilizando o coeficiente de Pearson. O coeficiente é classificado de -1 até 1 a correlação entre a variável independente e a dependente (LARSON; FARBER, 2015, p. 442).

Quando o valor do coeficiente do valor é ‘-1’, implica em uma correlação negativa perfeita, ou seja, enquanto a variável independente cresce, a variável dependente diminui (LARSON; FARBER, 2015, p. 443). Quando o coeficiente é ‘0’, significa que não há correlação entre as variáveis, a mudança na variável independente não altera o valor da variável dependente (LARSON; FARBER, 2015, p. 443). Por fim, quando o coeficiente é ‘1’, resulta em uma correlação positiva perfeita entre as variáveis, em outra palavras, enquanto a variável independente cresce, a variável dependente também cresce (LARSON; FARBER, 2015, p. 442).

Na matriz de correlação de Pearson das variáveis da taxa de evasão, observa-se que entre as variáveis independentes e a variável dependente há correlações positivas. Isso significa que quando uma variável independente diminui, a variável dependente diminui, já com relação as variáveis independentes para a taxa de evasão, percebe-se que há correlação positiva em todos os pares.

Isso nos diz que quando as variáveis independentes diminuem faz com que a variável dependente diminua, ou seja, conforme as notas e frequências diminuem, a taxa de evasão aumenta. A matriz de correlação das variáveis da inteligência artificial pode ser vista na figura 56.

Figura 56 – Matriz de correlação

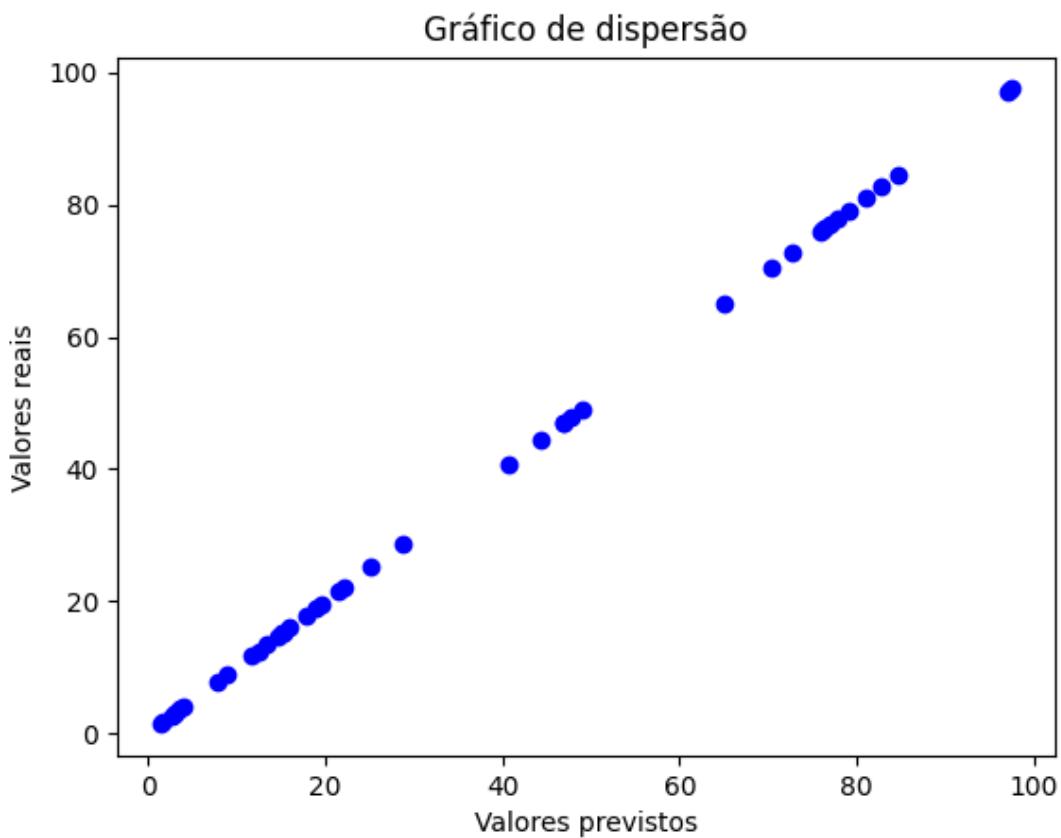


Fonte: elaborado pelo autor (2023).

O gráfico de dispersão permite visualizar o formato que os inúmeros pares apresentam, sendo eles utilizados como coordenadas no plano cartesiano, isso permite analisar qual comportamento os dados irão representar, assim é possível refletir qual função mais se aproxima do formato dos dados (LARSON; FARBER, 2015, p. 57).

No gráfico 57 é plotado os valores reais da taxa de evasão utilizando o eixo das ordenadas (y) e os valores previstos utilizando o eixo das abscissas (x). Devido a natureza da função do modelo utilizado, espera-se que a plotagem dos dados resulte em formato próximo a função de uma função linear. Qualquer ponto que fugir da linha imaginária, significa que aquele valor previsto não está sendo 100% explicado pelo modelo.

Figura 57 – Gráfico de dispersão



Fonte: elaborado pelo autor (2023).

Como pode ser observado no gráfico 57, ele aparenta que as previsões do modelo estão 100% de acordo com os dados reais.

9 CONCLUSÃO

Com base no que foi apresentado neste TCC, podemos validar a hipótese deste trabalho, relembrando que ela afirma que uma IA que tem a capacidade de encontrar padrões e que é capaz de analisar os dados dos alunos ao longo do tempo, é competente para predizer possíveis evasões escolares. Para validar este ponto, foi se necessário o desenvolvimento de um software que permitisse o cadastro de dados dos alunos e que em seu código, tenha implementado um modelo de predição com IA. Ao final deste trabalho, a parte de cadastro foi desenvolvida parcialmente, já a parte da IA foi desenvolvida com sucesso.

Com base no que foi apresentado no capítulo **CONCLUSÃO**, podemos concluir que o software desenvolvimento integrado com IA desenvolvida deste projeto, pode predizer possíveis evasões escolares. Devido o software conseguir predizer possíveis evasões escolares, os responsáveis por observar as estatísticas das evasões, podem tomar decisões de forma direta ou indireta, afim de evitar que a evasão aconteça. Assim o software pode ajudar em diversos âmbitos, não somente a instituição, bem como a sociedade, o Brasil, o meio científico e outros.

Os futuros trabalhos que irão se basear neste trabalho podem buscar responder questões que este não trabalho não respondeu ou também podem buscar melhorar o que foi apresentado aqui. Aprimorando a introdução ou o desenvolvimento deste trabalho, aprofundando em suas bases teóricas, como também aperfeiçoar o software ou a inteligência artificial desenvolvidos.

Esses futuros trabalhos podem responder se as variáveis utilizadas neste trabalhos são o suficiente para calcular a taxa de evasão. Será que características como endereço do individuo, mensalidade do curso, idade do aluno, situação financeira do aluno e outras características influenciam na decisão de evasão.

Será que o modelo de regressão linear múltipla é modelo que mais explica a taxa de evasão escolar. Sera que outros tipos de modelos não são mais adequados que o modelo de regressão linear, por exemplo, regressão não linear, *Nearest Neighbors*, *Naive Bayes*, *Decision Trees* e dentre outros

Talvez a escolha das ferramentas e tecnologias foi a mais adequada, a linguagem de programação Python conseguiu suprir as necessidades do projeto ou outra linguagem como o Java poderia ter feito isso de uma forma melhor. Estas são algumas de muitas questões que futuros trabalhos podem responder.

REFERÊNCIAS

- 2022 Developer Survey. *Survey*, 2023. Disponível em: <<https://survey.stackoverflow.co/2022/#version-control-version-control-system>>. Acesso em: 10 jun. 2023. Citado na página 47.
- A licença do PostgreSQL. *Open Source*, fev 2009. Disponível em: <<https://opensource.org/license/postgresql/>>. Acesso em: 16 mai. 2023. Citado na página 48.
- ABANDONO e evasão escolar de adolescentes e jovens: Cenário do território nacional. *imds*, 2022. Disponível em: <<https://imdsbrasil.org/doc/Imds005-2022%20%20Territ%C3%B3rioAbandono&Evas%C3%A3o.pdf>>. Acesso em: 27 fev. 2023. Citado na página 20.
- ADOBE Illustrator Logo. *Logo Download*, ago 2021. Disponível em: <<https://logodownload.org/adobe-illustrator-logo/>>. Acesso em: 29 jul. 2023. Citado na página 52.
- ASSUMA O CONTROLE — SUA WEB, SEU LOGOTIPO. *w3*, 2023. Disponível em: <<https://www.w3.org/html/logo/>>. Acesso em: 10 jun. 2023. Citado na página 42.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: Guia do usuário*. Fábio Freitas da Silva and Cristina de Amorim Machado. 2. ed. Rio de Janeiro: Elsevier, 2012. 757 p. ISBN 978-85-352-8565-9. Citado 5 vezes nas páginas 25, 72, 73, 75 e 106.
- BRASIL. MINISTÉRIO DA EDUCAÇÃO. *Metodologia de Cálculo dos Indicadores de Fluxo da Educação Superior*: 1984. Brasília, DF, 2017. 41 p. Disponível em: <https://download.inep.gov.br/informacoes_estatisticas/indicadores_educacionais/2017/metodologia_indicadores_trajetoria_curso.pdf>. Citado na página 20.
- CASTRO, E.; HYSLOP, B. *Guia Prático & Visual: HTML5 e CSS3*. 7. ed. Rio de Janeiro: Alta Books, 2013. 550 p. Citado na página 42.
- CEREDA, P. Latex, latex3 and expl3 logos and usage. *Latex Project*, 2023. Disponível em: <<https://www.latex-project.org/about/logos/>>. Acesso em: 02 ago. 2023. Citado na página 46.
- CHACON, S.; STRAUB, B. 1.1 primeiros passos - sobre o controle de versão. *Git*, 2014. Disponível em: <<https://git-scm.com/book/en/v2>>. Acesso em: 01 ago. 2023. Citado na página 47.
- CHACON, S.; STRAUB, B. 1.2 introdução - uma breve história do git. *Git*, 2014. Disponível em: <<https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>>. Acesso em: 01 ago. 2023. Citado na página 47.
- CRIANDO um documento no Overleaf. *Overleaf*, 2023. Disponível em: <https://pt.overleaf.com/learn/how-to/Creating_a_document_in_Overleaf>. Acesso em: 10 jun. 2023. Citado 2 vezes nas páginas 49 e 50.
- DANILO. Qual a diferença entre modelagem conceitual, lógica e física? *StackOverflow*, 2018. Disponível em: <<https://pt.stackoverflow.com/questions/294699/qual-a-diferen%C3%A7a-entre-modelagem-conceitual-l%C3%C3%83gica-e-f%C3%A9%C3%ADca>>. Citado na página 56.

DATE, C. J. *Introdução a Sistemas de Banco de Dados*. Daniel Vieira. 8. ed. Rio de Janeiro: Elsevier, 2003. 1623 p. ISBN 978-85-352-8445-4. Citado na página 25.

DEVELOPER Ecosystem: Python. *Jet Brains*, 2022. Disponível em: <<https://www.jetbrains.com/lp/devecosystem-2022/python/>>. Acesso em: 01 ago. 2023. Citado na página 45.

DEVELOPER Survey 2023. Survey, 2023. Disponível em: <<https://survey.stackoverflow.co/2023/>>. Acesso em: 28 jul. 2023. Citado 5 vezes nas páginas 43, 44, 48, 49 e 51.

DHADUK, H. 10 padrões de arquitetura de software que você deve conhecer: Os padrões de arquitetura de software são muito importantes, pois podem ser usados para resolver vários problemas. escolha o tipo de padrões de arquitetura de software neste blog. *Simform*, abr 2020. Disponível em: <<https://www.simform.com/blog/software-architecture-patterns/>>. Acesso em: 8 ago. 2023. Citado na página 39.

DICIONÁRIO de Dados. *Sankhya*, abr 2023. Disponível em: <<https://developer.sankhya.com.br/docs/dicion%C3%A1rio-de-dados>>. Acesso em: 16 mai. 2023. Citado na página 58.

DIRETRIZES da marca. *Bootstrap*, 20?? Disponível em: <<https://getbootstrap.com/docs/5.3/about/brand/>>. Acesso em: 29 jul. 2023. Citado na página 45.

DIRETRIZES da marca registrada Figma. *figma*, 20?? Disponível em: <<https://www.figma.com/using-the-figma-brand/>>. Acesso em: 27 jul. 2023. Citado na página 51.

DOWNLOAD Git Bash vector (SVG) logo. *World Vector Logo*, 20?? Disponível em: <<https://worldvectorlogo.com/pt/logo/git-bash>>. Acesso em: 29 jul. 2023. Citado na página 50.

DUCKETT, J. *HTML & CSS: design and build websites*. Indianapolis: John Wiley & Sons, 2011. 490 p. ISBN 978-1-118-00818-8. Citado na página 42.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco de Dados*. 4. ed. [S.l.]: Pearson Addison Wesley, 2011. 788 p. ISBN 978-85-4301-381-7. Citado na página 48.

ESPECIFICAÇÃO de linguagem ECMAScript® 2024: Rascunho ECMA-262 / 26 de julho de 2023. TC39, jul 2023. Disponível em: <<https://tc39.es/ecma262/>>. Acesso em: 31 jul. 2023. Citado na página 43.

EVOLUÇÃO e intensidade do abandono e da evasão escolar: Uma análise a partir dos dados do censo escolar. *imds*, 20?? Disponível em: <<https://imdsbrasil.org/abandono-e-evasao/evolucao-e-intensidade-do-abandono-e-da-evasao-escolar-uma-analise-a-partir-dos-dados-do-censo-escolar>>. Acesso em: 01 mar. 2023. Citado na página 20.

GITHUB Logos and Usage. *GitHub*, 2022. Disponível em: <<https://github.com/logos>>. Acesso em: 27 jul. 2023. Citado na página 52.

GREIF, S. Css frameowrks: Frameworks and libraries that give you pre-made components or styles. *State of CSS*, 2022. Disponível em: <<https://2022.stateofcss.com/en-US/>>. Acesso em: 02 ago. 2023. Citado na página 45.

HTML & CSS. W3, 2016. Disponível em: <<https://www.w3.org/standards/webdesign/htmlcss>>. Acesso em: 07 mar. 2023. Citado 2 vezes nas páginas 26 e 42.

HTML/TREINAMENTO/O que é HTML?: O que é html? W3, nov 2011. Disponível em: <https://www.w3.org/community/webed/wiki/HTML/Training/What_is_HTML%3F>. Acesso em: 10 mar. 2023. Citado na página 42.

ICONS and names usage guidelines. *Visual Studio*, 2023. Disponível em: <<https://code.visualstudio.com/brand>>. Acesso em: 28 jul. 2023. Citado 2 vezes nas páginas 27 e 51.

INTRODUÇÃO ao Django. *Mozilla*, 2023. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Django/Introduction>>. Acesso em: 03 ago. 2023. Citado 2 vezes nas páginas 45 e 46.

ISO AND IEC AND IEEE. *Systems and software engineering — Vocabulary*. Switzerland, 2017. 536 p. Disponível em: <<https://sci-hub.se/https://doi.org/10.1109/IEEEESTD.2017.8016712>>. Citado na página 22.

JOSEFIN Sans. *Google Fonts*, 2023. Disponível em: <<https://fonts.google.com/specimen/Josefin+Sans>>. Acesso em: 14 out. 2023. Citado na página 110.

JUICY_FISH. Servidor sql grátis ícone. *Flaticon*, 2023. Disponível em: <https://www.flaticon.com/br/icone-gratis/servidor-sql_5815478>. Acesso em: 29 jul. 2023. Citado na página 49.

JULIANO. Gerenciamento de banco de dados: Análise comparativa de sgbd's. *DevMedia*, 2014. Disponível em: <<https://www.devmedia.com.br/gerenciamento-de-banco-de-dados-analise-comparativa-de-sgbd-s/30788>>. Acesso em: 08, mar. 2023. Citado na página 25.

LAMPORT, L. *Latex: A document preparation system, user's guide and reference manual*. 2. ed. [S.l.]: Addison Wesley, 1994. 279 p. ISBN 0-201-52983-1. Citado na página 46.

LARSON, R.; FARBER, B. *Estatística Aplicada*. José Fernan Pereira. 6. ed. São Paulo: Pearson Education, 2015. 659 p. ISBN 978-85-4301-811-9. Citado 4 vezes nas páginas 112, 113, 118 e 119.

LATEX – Um sistema de preparação de documentos. *Latex Project*, 2023. Disponível em: <<https://www.latex-project.org/>>. Acesso em: 02 ago. 2023. Citado na página 46.

LOGO. *PostgreSQL*, abr 2022. Disponível em: <<https://wiki.postgresql.org/wiki/Logoa>>. Acesso em: 03 ago. 2023. Citado na página 48.

LOGOTIPOS oficiais do Django. *Django*, 2023. Disponível em: <<https://www.djangoproject.com/community/logos/>>. Acesso em: 29 jul. 2023. Citado na página 46.

LONG, J. Logos. *Git*, 20?? Disponível em: <<https://git-scm.com/downloads/logos>>. Acesso em: 10 jun. 2023. Citado na página 47.

MAPA do Ensino Superior: Evasão. *Semesp*, 2021. Disponível em: <<https://www.semesp.org.br/mapa/edicao-11/brasil/evasao/>>. Acesso em: 01 mar. 2023. Citado na página 21.

MATOSO, D. Git bash: Como instalar e usar o terminal do git no windows. *Web Dev Drops*, fev 2020. Disponível em: <<https://www.webdevdrops.com/git-bash-como-instalar-usar/>>. Acesso em: 27 fev. 2020. Citado na página 50.

MODELAGEM física de dados. *Erwin*, 2023. Disponível em: <<https://www.erwin.com/br-pt/solutions/data-modeling/physical.aspx>>. Citado na página 57.

MONTSERRAT. *Google Fonts*, 2023. Disponível em: <<https://fonts.google.com/specimen/Montserrat>>. Acesso em: 14 out. 2023. Citado na página 110.

NOLETO, C. Poo: tudo sobre programação orientada a objetos! *Betrybe*, mar 2021. Disponível em: <<https://blog.betrybe.com/tecnologia/poo-programacao-orientada-a-objetos/>>. Acesso em: 27 abr. 2023. Citado na página 25.

NOLETO, C. Bootstrap: o que é, como usar e para que serve esse framework. *Betrybe*, jul 2022. Disponível em: <<https://blog.betrybe.com/bootstrap/>>. Acesso em: 02 ago. 2023. Citado na página 45.

O logotipo do Python. *Python*, 2023. Disponível em: <<https://www.python.org/community/logos/>>. Acesso em: 28 jul. 2023. Citado na página 44.

O que é Figma? *Figma*, 20?? Disponível em: <<https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>>. Acesso em: 27 jul. 2023. Citado na página 50.

O que é modelagem de dados? *Amazon*, 2023. Disponível em: <<https://aws.amazon.com/pt/what-is/data-modeling/#:~:text=Modelagem%20de%20dados%20%C3%A9%20o,de%20informa%C3%A7%C3%B5es%20de%20qualquer%20organiza%C3%A7%C3%A3o.>> Citado 2 vezes nas páginas 54 e 56.

O que é PowerPoint e para que serve. *SlidesGo*, 2023. Disponível em: <<https://slidesgo.com/pt/slidesgo-school/tutoriais-do-powerpoint/o-que-e-powerpoint-e-para-que-serve/>>. Acesso em: 11 set. 2023. Citado na página 53.

O que é SQL? *Amazon*, 2023. Disponível em: <<https://aws.amazon.com/pt/what-is/sql/>>. Acesso em: 07 mar. 2023. Citado na página 26.

O tutorial de Python. *Python*, jul 2023. Disponível em: <<https://docs.python.org/pt-br/3/tutorial/>>. Acesso em: 28 jul. 2023. Citado 2 vezes nas páginas 26 e 44.

OVERLEAF Official Logos. *Overleaf*, 2023. Disponível em: <<https://www.overleaf.com/for/partners/logos>>. Acesso em: 27 jul. 2023. Citado na página 49.

PALETTON.COM. *Paletton*, 2023. Disponível em: <<https://paletton.com/#uid=a3x0I0kxb+ +k9+mqd+WUj+VZECE>>. Acesso em: 14 out. 2023. Citado na página 109.

PATEL, N. Design: o que é, tipos e importância para as empresas. *Neil Patel*, 2023. Disponível em: <<https://neilpatel.com/br/blog/design-o-que-e/>>. Acesso em: 14 out. 2023. Citado na página 108.

PAUTASSO, C. *Software Architecture*: visual lecture notes. Victoria: LeanPub, 2021. 688 p. Citado na página 38.

PERFECT, P. Power point grátis ícone. *Flaticon*, 2023. Disponível em: <https://www.flaticon.com/br/icone-gratis/power-point_888874?term=powerpoint&page=1&position=1&origin=search&related_id=888874>. Acesso em: 29 jul. 2023. Citado na página 53.

PERGUNTAS frequentes. *pgAdmin*, 2023. Disponível em: <<https://www.pgadmin.org/faq/>>. Acesso em: 7 ago. 2023. Citado 2 vezes nas páginas 27 e 52.

PERGUNTAS Frequentes: Geral. *Django*, 2023. Disponível em: <<https://docs.djangoproject.com/en/4.2/faq/general/>>. Acesso em: 9 ago. 2023. Citado na página 40.

PINHEIRO, F. Entendendo o mtv do django: Veja neste artigo como funciona o padrão de projeto utilizando em projetos django, o mtv. *TreinaWeb*, abr 2020. Disponível em: <<https://www.treinaweb.com.br/blog/entendendo-o-mtv-do-django>>. Acesso em: 18 out.2023. Citado na página 41.

PISA 2018: PARA VOLTAR A AVANÇAR, BRASIL PRECISA INVESTIR NA PROFISSIONALIZAÇÃO E NO PRESTÍGIO DA CARREIRA DOS PROFESSORES. *Todospelaeducacao*, dez 2019. Disponível em: <<https://todospelaeducacao.org.br/noticias/Pisa-2018/>>. Acesso em: 26 nov. 2023. Citado na página 23.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software*: Uma abordagem profissional. Francisco Araújo da Costa. 9. ed. Porto Alegre: AMGH Editora Ltda, 2021. 1305 p. ISBN 9781259872976. Citado na página 40.

RICHARDS, M. *Software Architecture Patterns*: Understanding common architecture patterns and when to use them. 2. ed. Sebastopol: O'Reilly Media, 2015. 46 p. ISBN 978-1-491-92424-2. Citado na página 38.

ROBERTA. Administração de dados x administração de base de dados. *DevMedia*, 2007. Disponível em: <<https://www.devmedia.com.br/administracao-de-dados-x-administracao-de-base-de-dados/4658>>. Citado na página 54.

RUDLOFF. File:css3 logo and wordmark.svg. *Wikimedia Commons*, Mai 2016. Disponível em: <https://commons.wikimedia.org/wiki/File:CSS3_logo_and_wordmark.svg>. Acesso em: 28 jul. 2023. Citado na página 43.

SOBRE. *PostgreSQL*, 2023. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 03 ago. 2023. Citado 2 vezes nas páginas 27 e 48.

SOMMERVILLE, I. *Engenharia de Software*. Kalinka Oliveria and Ivan Bosnic. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p. ISBN 978-85-7936-108-1. Citado 2 vezes nas páginas 40 e 72.

SOUZA, I. de. Adobe illustrator: dicas e atalhos para a ferramenta de edição. *Rockcontent*, fev 2019. Disponível em: <<https://rockcontent.com/br/blog/adobe-illustrator/>>. Acesso em: 20 fev. 2019. Citado na página 51.

TEOREY, T. et al. *PROJETO E MODELAGEM DE BANCO DE DADOS*. Daniel Vieira. 2. ed. Rio de Janeiro: Elsevier, 2014. 309 p. ISBN 978-85-352-6445-6. Citado na página 54.

UI Design. *UX tools*, 2022. Disponível em: <<https://uxtools.co/survey/2022/ui-design>>. Acesso em: 27 jul. 2022. Citado na página 50.

UMA breve história da web. *CERN*, 2023. Disponível em: <<https://www.home.cern/science/computing/birth-web/short-history-web>>. Acesso em: 10 jun. 2023. Citado na página 42.

VAZQUEZ, C. E.; SIMÕES, G. S. *Engenharia de Requisitos*: Software orientado ao negócio. Rio de Janeiro: BRASPORT Livros e Multimídia Ltda, 2016. 419 p. Citado na página 30.

VILLAIN, M. Figma: o que é a ferramenta, design e uso. *Alura*, 2023. Disponível em: <<https://www.alura.com.br/artigos/figma>>. Acesso em: 07 mar. 2023. Citado na página 27.

WILLIANS, C. File:unofficial javascript logo 2.svg. *Wikimedia Commons*, out 2011. Disponível em: <https://commons.wikimedia.org/wiki/File:Unofficial_JavaScript_logo_2.svg>. Acesso em: 27 jul. 2023. Citado na página 43.