



UNIVERSIDADE FEDERAL DA BAHIA
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA QUÍMICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO
DE PROCESSOS

JONATHAS MIRANDA LIMA

DASHBOARD DE GESTÃO ACADÊMICA PARA O CURSO DE ENGENHARIA DE
CONTROLE E AUTOMAÇÃO DE PROCESSOS DA UFBA

SALVADOR

2023

JONATHAS MIRANDA LIMA

**DASHBOARD DE GESTÃO ACADÊMICA PARA O CURSO DE ENGENHARIA DE
CONTROLE E AUTOMAÇÃO DE PROCESSOS DA UFBA**

Trabalho de conclusão de curso apresentado ao colegiado do curso de Engenharia de Controle e Automação de Processos da Universidade Federal da Bahia como requisito parcial para obtenção do título de bacharel em Engenharia de Controle e Automação de Processos.

Orientador: Prof. Dr. Daniel Diniz Santana

Coorientador: Prof. Dr. Raony Maia Fontes

SALVADOR

2023

FOLHA DE APROVAÇÃO

JONATHAS MIRANDA LIMA

DASHBOARD DE GESTÃO ACADÊMICA PARA O CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DE PROCESSOS DA UFBA

Trabalho de Conclusão de Curso apresentado ao colegiado do curso de graduação de Engenharia de Controle e Automação de Processos da Universidade Federal da Bahia como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação de Processos.

Comissão examinadora:  Documento assinado digitalmente
DANIEL DINIZ SANTANA
Data: 22/12/2023 17:10:25-0300
Verifique em <https://validar.iti.gov.br>

Orientador

Prof. Dr. Daniel Diniz Santana (UFBA)

 Documento assinado digitalmente
RAONY MAIA FONTES
Data: 22/12/2023 19:05:59-0300
Verifique em <https://validar.iti.gov.br>

1º Membro Avaliador

Prof. Dr. Raony Maia Fontes (UFBA)

 Documento assinado digitalmente
MARCIO ANDRE FERNANDES MARTINS
Data: 22/12/2023 17:24:33-0300
Verifique em <https://validar.iti.gov.br>

2º Membro Avaliador

Prof. Dr. Márcio André Fernandes Martins (UFBA)

 Documento assinado digitalmente
IZABEL CRISTINA SILVA XAVIER
Data: 22/12/2023 21:47:51-0300
Verifique em <https://validar.iti.gov.br>

3º Membro Avaliador

Izabel Cristina Silva Xavier (UFBA)

Salvador, BA, 13 de dezembro de 2023.

AGRADECIMENTOS

Primeiramente, agradeço a Deus por me permitir alcançar esta conquista. Aos meus familiares, em especial à minha mãe Áurea, ao meu pai Jailton e aos meus irmãos Jailton e Joanderson, agradeço por todo suporte ofertado, por acreditar em meu potencial, pelo amor, preocupação e fé. À minha grande parceira de vida, Isabelly, agradeço por estar sempre presente, pelos incentivos, paciência e, acima de tudo, por todo amor oferecido. Aos amigos, obrigado pela parceria. Aos meus orientadores, Professores Daniel e Raony, agradeço pelos ensinamentos, oportunidade e pelo apoio.

A todos que estiveram presentes e que contribuíram, de alguma maneira, para a realização deste sonho, meus sinceros agradecimentos.

RESUMO

Este trabalho se propõe a desenvolver um *dashboard* interativo voltado para gestão acadêmica com os principais indicadores do curso de Engenharia de Controle e Automação de Processos da UFBA a fim de apoiar as tomadas de decisões do colegiado. Os dados foram tratados com o auxílio da linguagem de programação Python integrada ao Power BI®, *software* utilizado para representação visual das informações. O *dashboard* permite relacionar as disciplinas que possuem maior dependência dos pré-requisitos sugeridos na matriz curricular, avaliar as taxas de aprovação e reprovação em relação à antecipação ou atraso nas inscrições das disciplinas, bem como observar essas relações por período letivo. Além disso, possibilita a análise do desempenho individual dos alunos e oferece apoio ao planejamento acadêmico com base nas sugestões de disciplinas para os alunos. O Dashboard Acadêmico CCECA é uma ferramenta que segue as melhores práticas de interface amigável e pode ser alimentada com novos dados semestralmente, agregando valor às análises e informações que podem ser extraídas dele.

Palavras-chave: Dashboard, Power BI®, Python, Indicadores de Gestão Acadêmica.

ABSTRACT

This work aims to develop an interactive dashboard for academic management with key indicators of the Process Control and Automation Engineering at UFBA to support the Collegiate's decision making. Data was processed using the Python programming language integrated with Power BI[®], a software used for visual representation of information. The dashboard enables the examination of curricular courses that have a strong dependence on suggested prerequisites in the curriculum, measuring the approval and failure rates concerning the anticipation or delay in course enrollments, and observe these relationships per semester. Furthermore, it allows the analysis of individual student performance and provides support for academic planning based on course recommendations. The Academic Dashboard CCECA is a tool that follows the best practices of a user-friendly interface and can be updated with new data semester by semester, adding value to the analyses and information that can be extracted from it.

Keywords: Dashboard, Power BI[®], Python, Academic Management Indicators.

LISTA DE FIGURAS

Figura 1: Exemplo de BoxPlot: comparativo do índice de qualidade de produtos em 3 plantas.	16
Figura 2: a) Gráfico de dispersão com Sxy positiva. b) Gráfico de dispersão com Sxy negativa.	17
Figura 3: Quadrante Gartner das plataformas de análise e business intelligence.	21
Figura 4: Exemplo de um esquema de estrela de vendas.	23
Figura 5: Exemplo de <i>dashboard</i> criado no Power BI®.....	24
Figura 6: Dashboard Acadêmico CCECA.	26
Figura 7: Estrutura da planilha de históricos.....	27
Figura 8: Estrutura da planilha de disciplinas.	28
Figura 9: Estrutura da aba Geral da planilha de pré-requisitos.	28
Figura 10: Estrutura da aba Pré-Requisitos da planilha Pré-Requisitos.xlsx (a).	30
Figura 11: Estrutura da aba Pré-Requisitos da planilha Pré-Requisitos.xlsx (b).	30
Figura 12: Correlação entre resultados X cumprimento de pré-requisitos de 1ª ordem.	31
Figura 13: Correlação entre resultados X cumprimento de pré-requisitos até 3ª ordem.....	31
Figura 14: Correlação entre resultados X cumprimento de pré-requisitos até 3ª ordem com peso de semestre de pré-requisito.	31
Figura 15: Relacionamento star schema.	33
Figura 16: Selecionando diretório de instalação do Python.	35
Figura 17: Estrutura da pasta 'Base PowerBI'.	36
Figura 18: Opção Transformar dados do Power BI®.....	36
Figura 19: Selecionar a tabela desejada.	37
Figura 20: Selecionar opção Editor Avançado.	37
Figura 21: Alterar o Caminho da pasta Base PowerBI.	38
Figura 22: Opção Fechar e Aplicar.....	38
Figura 23: Capa.	39
Figura 24: Overview.....	40
Figura 25: Pré-Requisitos.....	41
Figura 26: Pré-Requisitos - Tipo de Disciplina.	42
Figura 27: Pré-Requisitos e Resultados no Tempo.	43
Figura 28: Antecipação Atraso.....	44
Figura 29: Antecipação Atraso - Tipo de Disciplina.	44
Figura 30: Antecipação Atraso e Resultado no Tempo.....	45
Figura 31: Aluno.....	46
Figura 32: Planejamento.	47
Figura 33: Bibliotecas.....	53
Figura 34: Definindo diretórios e criando DFs.....	54

Figura 35: Concatenando dados de histórico.	54
Figura 36: Criando DFs Auxiliares e Tratando dados.	55
Figura 37: Tratamento dos dados.	56
Figura 38: Organizando DF aux_equivalencia.	57
Figura 39: Preenchendo colunas de disciplinas no DF aux_equivalencia.	58
Figura 40: Preenchendo colunas do DF aux_equivalencia.	59
Figura 41: Calculando semestre de equivalência no DF aux_equivalencia.	59
Figura 42: Calculando a porcentagem de cumprimento de pré-requisitos para cada inscrição em disciplinas do DF Historico.	60
Figura 43: Acrescentando algumas colunas ao DF Historico.	61
Figura 44: Segmentação em intervalos das colunas Cumprimento de pré-requisito 2 e Cumprimento de pré-requisito 3.	62
Figura 45: Criando um DF de sugestão de disciplinas para os alunos.	63
Figura 46: Criando o DF vagas_planejamento.	63
Figura 47: Criando o DF alunos.	65
Figura 48: Apagando os DFs que não serão mais utilizados.	65
Figura 49: Criando ambiente virtual com versão do python 3.7.7 chamado envpowerbi.	67
Figura 50: Confirmando a criação do ambiente virtual e acessando-o.	68
Figura 51: Instalando o pacote Jupyter.	69
Figura 52: Instalando o pacote pandas.	69
Figura 53: Selecionando o ambiente virtual no Power BI®.	70
Figura 54: Opção Script do Python em Obter Dados.	71
Figura 55: Colando o script do Python no Power BI®.	71
Figura 56: Selecionando as tabelas importadas do script.	72

LISTA DE TABELAS

Tabela 1: Ranking Tiobe de Linguagens de Programação 2023.....	18
Tabela 2: Nomes antigos e novos das colunas renomeadas	32
Tabela 3: Tipos de registros em que considerou ou não se o aluno estava afastado.	64
Tabela 4: Funções DAX utilizadas no trabalho.....	73
Tabela 5: Disciplinas por grupo.....	77

LISTA DE ABREVIATURAS

AA	Aprovação
AM	Aprovação
AP	Aprovação
API	Application Programming Interface
DAX	Data Analysis Expressions
DF	Data Frame
DI	Dispensa Interna
DU	Dispensa UFBA
E/S	Entrada/Saída
IES	Instituições de Ensino Superior
KPI	Key Performance Indicator
OB	Obrigatórias
OP	Optativa
RF	Reprovação por Falta
RM	Reprovação por Nota
RR	Reprovação por Nota
STI	Superintendência de Tecnologia da Informação
TR	Trancamento
UFBA	Universidade Federal da Bahia

SUMÁRIO

1	INTRODUÇÃO	12
1.1	CONTEXTUALIZAÇÃO E JUSTIFICATIVA.....	12
1.2	OBJETIVOS DO TRABALHO.....	14
1.3	ESTRUTURA DO TRABALHO.....	14
2	FUNDAMENTOS TEÓRICOS	15
2.1	ESTATÍSTICA E ANÁLISE DE DADOS.....	15
2.2	LINGUAGENS DE PROGRAMAÇÃO.....	17
2.3	FERRAMENTAS DE ANÁLISE DE DADOS.....	19
2.4	INTEGRAÇÃO PYTHON E POWER BI®.....	23
2.5	DASHBOARD.....	24
3	DASHBOARD ACADÊMICO CCECA	26
3.1	APRESENTAÇÃO E FUNCIONALIDADES.....	26
3.2	SELEÇÃO DO SOFTWARE.....	27
3.3	FONTE DE DADOS.....	27
3.4	TRATAMENTO DE DADOS POWER BI®.....	30
4	GUIA DE USO DO DASHBOARD ACADÊMICO CCECA	34
4.1	REQUISITOS PARA FUNCIONAMENTO.....	34
4.2	PRIMEIRA UTILIZAÇÃO.....	34
4.3	INTERFACES DO DASHBOARD ACADÊMICO CCECA.....	38
5	CONCLUSÃO	48
5.1	SUGESTÕES PARA TRABALHOS FUTUROS.....	48
6	REFERÊNCIAS	50
7	APÊNDICE A: TRATAMENTO DOS DADOS COM O PYTHON	53
8	APÊNDICE B: INTEGRAÇÃO ENTRE PYTHON E POWER BI® E IMPORTAÇÃO DE DADOS	66
8.1	INTEGRAÇÃO PYTHON E POWER BI®.....	66
8.2	IMPORTAÇÃO DOS DADOS.....	70
9	APÊNDICE C: CÁLCULO DAS MEDIDAS	73
10	APÊNDICE D: GRUPO DE DISCIPLINAS	77

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E JUSTIFICATIVA

Nos últimos anos, tem-se observado uma redução nas taxas de diplomação de diversos cursos de Instituições de Ensino Superior (IES), principalmente pelo abandono da graduação em definitivo causado pelo mau desempenho nos componentes curriculares (PINHEIRO, 2021). Fagundes, Luce e Espinar (2014) ressaltam a importância de avaliar a qualidade de ensino e o desempenho dos estudantes através de indicadores de êxito acadêmico, além da inclusão da abordagem qualitativa na avaliação, a fim de lograr maior conhecimento da complexa, plural e multifacetária realidade humana.

Através de um estudo em uma instituição australiana, Krause e Coates (2008) comprovaram a importância de o estudante apresentar um bom desempenho no primeiro ano do curso para manter o nível acadêmico nos anos posteriores. Além disso, identificaram a necessidade de as instituições estabelecerem um programa de orientação a fim de garantir os melhores resultados acadêmicos para os seus alunos.

Bungâu, Pop e Borza (2017) informam que a taxa de estudantes de engenharia da Romênia que deixam o ensino superior é de, aproximadamente, 77% e que essa taxa se assimila aos números globais. Para Pinheiro (2021), os fatores que levam à desistência do curso podem ser de cunho: econômico, social, cultural, individual ou institucional. No que se refere ao engajamento acadêmico, os principais problemas que influenciam o abandono da graduação são o baixo desempenho educacional e as políticas abordadas pelas instituições.

Pinheiro (2021) verificou altas taxas de evasão e baixas taxas de diplomação nos cursos de Engenharia da Escola Politécnica da Universidade Federal da Bahia (UFBA) e afirma haver uma estreita relação entre fatores acadêmicos e a persistência dos estudantes no curso. Para a autora, nenhum dos nove cursos da instituição apresentou percentual de diplomação acima de 70%, quando a meta global seria de 90%. Isso resalta a importância de implantar intervenções que proporcionem a melhoria do desempenho dos estudantes.

Têm-se observado uma crescente adoção de Dashboards por parte dos gestores, uma vez que esses painéis traduzem a estratégia em objetivos, índices, iniciativas e tarefas adaptadas às necessidades de diferentes grupos e pessoas dentro de uma organização. Além disso, esses recursos aprimoram a comunicação e a coordenação estratégica, resultando em melhorias no desempenho corporativo (PERRONI, 2021). Os benefícios relacionados aos *Dashboards* são melhoria na performance organizacional, melhor visibilidade e integração das informações, ampliar a percepção dos problemas, fornecimento de informações práticas,

facilitar a tomada de decisões com embasamento, dentre outros (BUTTIGIEG; PACE; RATHERT, 2017).

Nesse contexto, indica-se a utilização de ferramentas como *Dashboards* com os principais indicadores educacionais para auxiliar a gestão acadêmica a obter métricas acerca do desempenho dos estudantes. Segundo a Microsoft (2023e), *Dashboards* conectam todos os tipos de métricas e origens de dados para ajudar organizações a extrair informações relevantes e apresentá-las de maneiras fáceis, atuais e rápidas para que se entenda os detalhes dos mais valiosos dados e se descubra respostas para perguntas cruciais. Para a Microsoft (2023e), ao ligar *dashboards* à métricas específicas ou indicadores chave de desempenho (do inglês *Key Performance Indicator*, KPI), ganha-se inteligência de negócio e a capacidade de aprofundar informações específicas de modo a monitorar continuamente o êxito organizacional.

Conhecendo-se as métricas, a instituição poderá seguir com orientações acadêmicas para os alunos que não atingiram o desempenho esperado.

A orientação tem como objetivo contribuir para a integração dos estudantes à vida universitária, contemplando aspectos pedagógicos, itinerário curricular, informações sobre políticas e normas da Universidade, assistência estudantil, participação em projetos e eventos, realização de estágios e aconselhamento acadêmico-profissional (UFBA, 2022b, p. 42).

O curso de Engenharia de Controle e Automação de Processos da UFBA teve seu início em 2009 com um projeto pedagógico que não contemplava pré-requisitos, que tinha o objetivo de fornecer autonomia aos estudantes. Porém, em 2019, ocorreu uma reestruturação da matriz curricular, na qual foram incluídos pré-requisitos para os componentes do núcleo básico (matemática e física), (UFBA, 2022^a214). Nessas condições, surgiu a necessidade de acompanhar o desempenho dos alunos e entender se a falta de pré-requisitos e a liberdade para cursar disciplinas avançadas, a qualquer momento do curso, prejudica o desenvolvimento estudantil.

Com base no cenário apresentado e nas necessidades do curso descritas no Projeto Pedagógico, associado à reforma curricular motivada pelas Diretrizes Curriculares para Engenharia – Resolução CNE/CES nº 01/2019, percebe-se que é preciso preservar a autonomia do estudante e utilizar indicadores para identificar alunos com baixo desempenho e/ou com maior risco de evasão, possibilitando uma atuação mais assertiva do colegiado. Nesse contexto, observa-se a oportunidade de aplicar ferramentas de análises de dados e de fornecer um *Dashboard* de gestão capaz de apoiar no processo de tomada de decisão do colegiado, através da disponibilização de uma visão generalista do curso e por estudante, além de ofertar um planejamento acadêmico baseado nos resultados encontrados. Pode-se citar que

as principais ferramentas técnicas utilizadas na solução do problema adotadas neste estudo são: a linguagem de programação Python e o *software* Power BI®.

1.2 OBJETIVOS DO TRABALHO

1.2.1 Objetivo Geral

Desenvolver um *dashboard open source* e interativo voltado para gestão acadêmica com indicadores para o curso de Engenharia de Controle e Automação de Processos da UFBA a fim de apoiar as tomadas de decisões do colegiado.

1.2.2 Objetivos Específicos

- Desenvolver KPIs para acompanhamento das taxas de aprovação, reprovação e evasão do curso de forma generalista;
- Desenvolver KPIs capazes de identificar alunos que necessitam de orientação acadêmica;
- Desenvolver método para apoiar o planejamento acadêmico no que se refere a oferta de disciplinas;
- Utilizar, no desenvolvimento do *dashboard*, uma arquitetura que segue as melhores prática de interface amigável para o usuário.

1.3 ESTRUTURA DO TRABALHO

Este trabalho foi estruturado em 5 capítulos. O primeiro apresenta uma introdução do problema e a sua justificativa, bem como os objetivos do estudo. O segundo capítulo contempla os fundamentos teóricos, que forneceram subsídio para o desenvolvimento da ferramenta. O terceiro apresenta o objeto de estudo desse trabalho, seu desenvolvimento e seu guia de uso. O quarto capítulo expõe as conclusões e, por fim, o quinto e último capítulo apresenta as referências. Este trabalho conta, também, com quatro apêndices que podem auxiliar o leitor a entender o processo de desenvolvimento da ferramenta.

2 FUNDAMENTOS TEÓRICOS

2.1 ESTATÍSTICA E ANÁLISE DE DADOS

O campo da estatística lida com a coleta, apresentação, análise e uso de dados para tomar decisões, resolver problemas e planejar produtos e processos. Técnicas estatísticas são utilizadas para ajudar a entender variabilidade no sentido de identificar quais fontes potenciais são mais importantes ou quais têm maior impacto no problema estudado e, com isso, auxiliar no processo de tomada de decisão (MONTGOMERY; RUNGER, 2009).

Para Mitra (2016), há inúmeros benefícios no uso da estatística como, por exemplo, saber quando tomar ações corretivas, determinar a capacidade do processo de atender às necessidades do cliente, identificar o momento ideal para realizar melhorias no processo e quais tipos de ações a serem aplicadas ou quando não interferir, já que todo processo possui variações e isso não significa, necessariamente, que este está fora de controle.

Para estudar o comportamento de algumas variáveis é necessário resumir os dados de forma que se apresente um ou alguns valores que sejam representativos da série. Para isso, emprega-se uma das seguintes medidas de posição (BUSSAB; MORETTIN, 2014):

- a) Média Aritmética: é a soma das observações dividida pelo número delas.
- b) Moda: definida como a realização mais frequente do conjunto de valores observados. Em alguns casos pode haver mais de uma moda, ou seja, a distribuição dos valores pode ser bimodal, trimodal etc.
- c) Mediana: é a realização que ocupa a posição central da série de observações quando estão ordenadas em ordem crescente. Quando o número de observações for par, usa-se como a mediana a média aritmética das duas observações centrais.

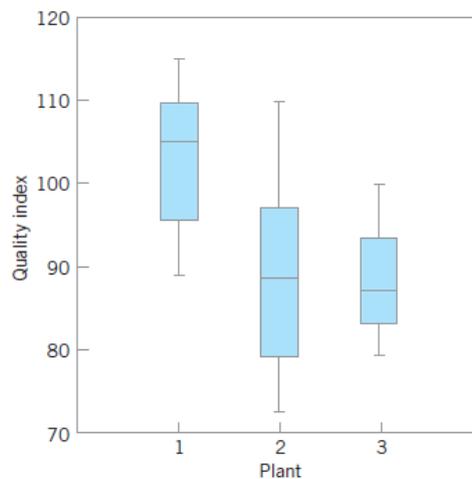
Em alguns casos, as medidas de posição citadas acima não são as mais adequadas para representar um conjunto de dados quando, por exemplo: a) são afetados, de forma exagerada, por fatores extremos e b) não é possível identificar a simetria da distribuição de dados. Dessa forma, utiliza-se o *quantil de ordem p* ou *p quantil*, indicada por $q(p)$, em que p é uma proporção qualquer, $0 < p < 1$, tal que $100p\%$ das observações sejam menores do que $q(p)$ (BUSSAB; MORETTIN, 2014):

- a) 1° *quantil* = 25° *percentil*: $q_{(0,25)} = q_1$;
- b) *Mediana* = 2° *quantil* = 50° *percentil*: $q_{(0,50)} = q_2$;
- c) 3° *quantil* = 75° *percentil*: $q_{(0,75)} = q_3$.

2.1.1 Box Plot

Segundo Montgomery (2012), o Box Plot, Figura 1, é um gráfico que exibe, simultaneamente, várias características importantes dos dados, como localização ou tendência central, dispersão ou variabilidade, afastamento da simetria e identificação de observações que estão extraordinariamente distantes da maior parte dos dados (“outliers”). O gráfico exibe os 3 quartis, o mínimo e o máximo dos dados em uma caixa retangular, alinhados horizontalmente ou verticalmente. A caixa inclui a faixa interquartil com a linha esquerda (ou inferior) no primeiro quartil q_1 , a linha direita (ou superior) no terceiro quartil q_3 , a linha central representante do segundo quartil $q_2 = mediana$ e as linhas das extremidades, que representam os valores extremos.

Figura 1: Exemplo de BoxPlot: comparativo do índice de qualidade de produtos em 3 plantas.



Fonte: Montgomery (2012).

2.1.2 Correlação

Para Wheelan (2016), a correlação mede o grau em que dois fenômenos estão relacionados entre si. Duas variáveis possuem correlação positiva se a variação numa delas é associada a uma variação da outra no mesmo sentido, tal como a relação entre peso e altura: pessoas mais altas pesam mais e pessoas mais baixas pesam menos (em média). Uma correlação é negativa se uma variação positiva numa das variáveis está associada a uma variação negativa na outra, tal como a relação entre exercício e peso.

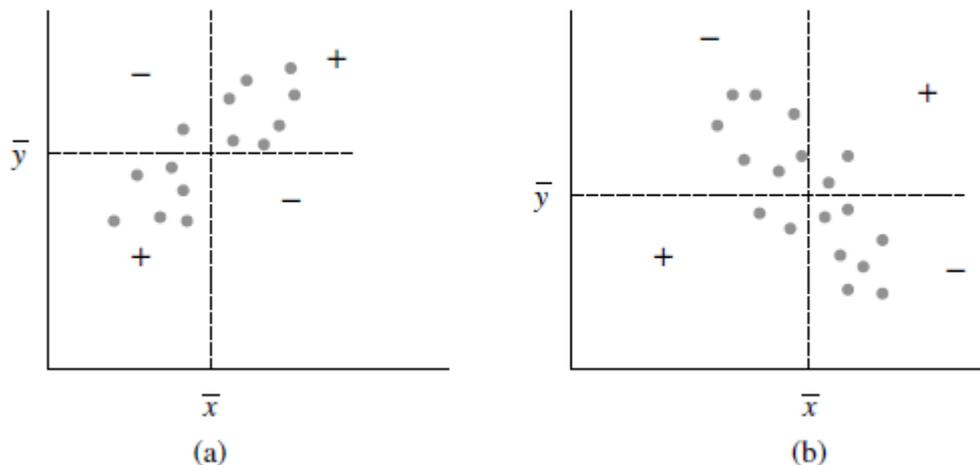
Segundo Devore (2006), dado n pares de observações $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, calcula-se a quantidade S_{xy} por:

$$S_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^n x_i y_i - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right) / n \quad (1)$$

\bar{x} e \bar{y} são os valores médios das observações de x e y , respectivamente e n o número de pares observados. Uma relação positiva significa que S_{xy} será positiva. Um argumento análogo mostra que, quando a relação for negativa, S_{xy} será negativa. Isso está ilustrado na Figura 2. Adicionalmente, o Coeficiente de Correlação Amostral (r) de n pares $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ é dado por:

$$r = \frac{S_{xy}}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}} = \frac{S_{xy}}{\sqrt{S_{xx}} \sqrt{S_{yy}}} \quad (2)$$

Figura 2: a) Gráfico de dispersão com S_{xy} positiva. b) Gráfico de dispersão com S_{xy} negativa.



Fonte: Devore (2006).

Segundo Wheelan (2016), uma correlação igual a 1, muitas vezes descrita como perfeita, significa que qualquer alteração em uma das variáveis está associada com uma alteração equivalente na outra variável no mesmo sentido. Quanto mais perto de 1 ou -1 estiver a correlação, mais forte é a associação. Uma correlação de 0 (ou próxima a 0) significa que as variáveis não têm associação significativa entre si.

2.2 LINGUAGENS DE PROGRAMAÇÃO

Para Melo e Silva (2003), uma linguagem de programação é um conjunto de recursos que podem ser compostos para constituir programas específicos, mais um conjunto de regras de composição que garantem que todos os programas podem ser implementados em computadores com qualidade apropriada. Após determinar a sintaxe, que determina a forma de manipular programas em uma linguagem, e a semântica, que determina a interpretação pretendida para cada elemento sintático, é preciso definir uma forma de implementar cada

elemento da linguagem e isso é feito por processadores: compiladores e/ou interpretadores.

Alura (2023) define linguagem de programação como idioma com regras de escrita, para garantir a comunicação com o computador e diz que as linguagens possuem muitas características diferentes entre si, desde a forma como as palavras e os símbolos especiais são definidos, até a forma como as "instruções" dadas ao computador são traduzidas até o momento de saírem das interfaces, onde são digitados códigos para serem encaminhadas aos processadores dos computadores.

Tiobe (2023) apresenta um *ranking* de popularidade das linguagens de programação entre engenheiros qualificados no mundo inteiro, que serve para apoiar na escolha da linguagem utilizada ao iniciar a construção de um novo *software*, Tabela 1.

Tabela 1: Ranking Tiobe de Linguagens de Programação 2023.

out/23	out/22	Linguagem de Programação	Classificação	Varição na Classificação
1	1	Python	14.82%	-2.25%
2	2	C	12.08%	-3.13%
3	4	C++	10.67%	+0.74%
4	3	Java	8.92%	-3.92%
5	5	C#	7.71%	+3.29%
6	7	JavaScript	2.91%	+0.17%
7	6	Visual Basic	2.13%	-1.82%
8	9	PHP	1.90%	-0.14%
9	10	SQL	1.78%	+0.00%
10	8	Assembly language	1.64%	-0.75%

Fonte: Adaptado de Tiobe (2023).

A seguir, estuda-se de forma mais aprofundada o Python, ferramenta auxiliar na análise deste trabalho.

2.2.1 Python

Mckinney (2023) diz que o Python surgiu em 1991 e tornou-se uma das mais populares linguagens de programação por volta de 2005. Sua popularização se deve à facilidade e rapidez de criação de pequenos programas, ou *scripts*, para automatizar tarefas e por ser utilizada para desenvolvimento de sites com o uso de numerosos *frameworks web*. Nos últimos 20 anos, o Python passou a ser considerada uma importante linguagem para a ciência de dados, *machine learning* e desenvolvimento de *softwares* em geral, principalmente no ambiente acadêmico e no setor empresarial.

Para Vasiliev (2023), a linguagem de programação Python é fácil de entender e é a escolha ideal para acessar, manipular e obter *insights* de dados de qualquer tipo. Para o autor,

além de possuir um conjunto sofisticado de estruturas *built-in* para operações básicas e um ecossistema robusto de bibliotecas *open source* para análise e manipulação de dados com qualquer nível de complexidade, na prática, é possível implementar inúmeras operações com uma única linha de código e pode-se integrar a ciência de dados com outras tarefas para criar aplicações completas e totalmente funcionais.

O Python possui um conjunto extenso de bibliotecas que oferece uma ampla gama de recursos. Essas ferramentas fornecem acesso à funcionalidade do sistema, como E/S (Entrada/Saída) de arquivos que de outra forma seriam inacessíveis no Python, assim como módulos que fornecem soluções padronizadas para problemas cotidianos. Alguns módulos são explicitamente projetados para incentivar e aprimorar a portabilidade de programas em Python, abstraindo os detalhes da plataforma em APIs (Interface de Programação de Aplicativos, do inglês *Application Programming Interface*) neutras. O Python (2023) diz que, Além da biblioteca padrão, há uma coleção ativa de centenas de milhares de componentes (de programas e módulos individuais a pacotes e frameworks de desenvolvimento de aplicativos inteiros), disponíveis no Python Package Index. Ainda de acordo com Python (2023), a maior parte das bibliotecas consiste em uma coleção de módulos e todas elas contêm exceções e funções embutidas, que são objetos que podem ser usados por todo o código Python sem a necessidade de uma instrução do tipo *import*.

De acordo com Data Science Academy (2023), as bibliotecas essenciais para análise de dados são Pandas e Numpy. A Pandas é um conjunto de ferramentas em Python que oferece estruturas de dados dinâmicas, ágeis e expressivas, criadas para simplificar a manipulação de dados rotulados ou relacionais. Seu propósito central é servir como uma base sólida e avançada para a análise eficiente de dados do mundo real dentro do ambiente Python (PANDAS, 2023). A NumPy representa o pilar essencial para a computação científica no ambiente Python. Esta biblioteca oferece um objeto *array* multidimensional, diversos derivados (como arrays mascarados e matrizes) e uma gama de funções para realizar operações rápidas em *arrays*. Inclui operações matemáticas, lógicas, manipulação de estrutura, ordenação, seleção, E/S de dados, transformadas discretas de Fourier, álgebra linear básica, operações estatísticas elementares, simulação aleatória e uma ampla gama de outras funcionalidades (NUMPY, 2023).

2.3 FERRAMENTAS DE ANÁLISE DE DADOS

Com a crescente disponibilidade de dados, é necessário utilizar ferramentas de análise capazes de fornecer informações valiosas para o negócio. Para isso, algumas ferramentas

estão disponíveis no mercado para auxiliar neste processo (ESTATMG, 2022): (i) o Power BI[®], que se destaca pela eficiência, qualidade e pela visualização de dados; (ii) o Excel[®], que é um dos softwares de análises focado em planilhas eletrônicas mais conhecidos do mercado, capaz de fazer qualquer tipo de cálculo e gráfico; (iii) o Data Studio[®], que é uma ferramenta de inteligência de negócio capaz de criar relatórios e painéis informativos para análises; e outras.

Para Bonel (2019), O Power BI[®] e o Excel[®] são grandes ferramentas, onde o Power BI[®] foi feito para gestão de dados e informação, já o Excel[®] é para operação do dia a dia. Para o autor, as ferramentas se complementam pois, em alguns casos, uma precisa da outra.

Para Oliveira (2022), as vantagens do Power BI[®] em relação ao Data Studio[®] são: a) maior quantidade de recursos e personalizações de seus visuais; b) conseguir se conectar e fazer relações com uma grande variedade de fonte de dados incluindo *scripts* Python, artifício que foi necessário nesse trabalho e c) uma gama de materiais em texto e vídeo e cursos *online*, além de uma comunidade virtual extremamente ativa.

Em 2023, O Power BI[®] é disponibilizado de três formas, as licenças Premium, Pro e Free. Já o Data Studio[®] é disponibilizado gratuitamente para quem tem uma conta do Google (OLIVEIRA, 2022). Em caso de uso doméstico, o Excel[®] se apresenta nas versões 365 Family e 365 Personal; para uso empresarial, a ferramenta é disponibilizada em quatro versões: 365 Business Basic, 365 Business Standard, 365 Business Premium e 365 Apps para Pequenos e Médios Negócios (MICROSOFT, 2023f).

A seguir, estuda-se de forma mais aprofundada a ferramenta Power BI[®], ferramenta de análise deste trabalho.

2.3.1 Power BI[®]

O Power BI[®] é uma ferramenta de *Business Intelligence*, onde é possível desenvolver análises, relatórios e dashboards, apresentando de forma gráfica e de fácil interpretação respostas para perguntas de negócios que auxiliam no processo de tomada de decisão. Por ser uma tecnologia da Microsoft, o Power BI[®] possui uma curva de aprendizado extremamente baixa, voltada para profissionais que não são de tecnologia da informação (BONEL, 2019).

Para Hyman (2022), o Power BI[®] é um produto que reúne muitos aplicativos menores baseados em nuvens com um objetivo específico: organizar, coletar, gerir e analisar muitos dados. Depois de extrair as fontes de dados e processá-las, o Power BI[®] pode ajudar a apresentar resultados de formas muito mais visuais.

Para O'Connor (2018), o Power BI[®] possui as seguintes capacidades:

- a) Processar dados de grandes bancos de dados, até mesmo aqueles que podem ler 1 milhão de linhas por hora por usuários;
- b) Gerar *dashboards* e relatórios de uma forma significativa para análise para o usuário final;
- c) Conectar facilmente falhas e erros para identificar a metodologia e abordagem adequada para o usuário.

O Power BI® também foi citado pelo Cx Today (2023) como plataforma líder de análise e *business Intelligence* e pela sua predominância no mercado quando se trata da adoção dos usuários se comparada a outras vinte plataformas. Esse comparativo levou em consideração várias características, dentre elas: informações automatizadas, preparação de dados, conectividade com fonte de dados, visualização de dados e *storytelling*. A Figura 3 mostra a classificação de cada uma das vinte plataformas analisadas.

Figura 3: Quadrante Gartner das plataformas de análise e business intelligence.



Fonte: Cx Today (2023).

2.3.1.1 Data Analysis Expressions (DAX)

Para lidar com dados críticos e para melhor análise e solução de problemas de negócio, utiliza-se Data Analysis Expressions (DAX), que é uma coleção de funções, operadores e constantes dentro do Power BI® (O'CONNOR, 2018). Para o autor, conceitos como Coluna Calculada e Medidas precisam ser compreendidos antes de seguir para o entendimento da DAX:

- a) Coluna Calculada: o resultado do cálculo de dados provenientes de duas colunas existentes é armazenado em uma terceira coluna. A nova coluna é chamada de Coluna Calculada;
- b) Medida: realiza cálculos dinamicamente em dados conforme há interação com relatórios do Power BI®. As medidas podem ser simples, como uma soma das vendas, ou podem ser complexas, como cálculos que contêm filtros adicionais ou combinações de duas ou mais tabelas de dados.

Expressões DAX utilizam uma sintaxe específica:

$$\text{Total Sales} = \text{SUM}(\text{Sales}[\text{Price}]) \quad (3)$$

calcula o total de vendas de uma base de dados, cujos elementos são: (i) *Total Sales*: nome da medida ou coluna calculada; (ii) =: operador que marca o começo da função; (iii) *SUM*: é uma função DAX que soma todos os números na coluna *Sales[Price]*; (iv) (): os parênteses contêm, no mínimo, um argumento; (v) *Sales*: nome da tabela em análise; (vi) *Price*: coluna da tabela onde a soma será aplicada.

2.3.1.2 Star Schema

Segundo Roy, Deb e Arora (2023), para relacionar dados, o Power BI® utiliza o *Star Schema*, ou esquema de estrela, que é uma abordagem de modelagem que requer a classificação das tabelas como fato ou dimensão. Em um esquema de estrela, tabelas de dimensões descrevem entidades de negócio como produtos, pessoas, tempo etc. e contém uma ou mais colunas chaves que agem como identificadores exclusivos. A partir disso, as tabelas de dimensões descrevem colunas que representam informações adicionais sobre a entidade.

Segundo Ferrari e Russo (2017), uma vez mentalizado que a tabela se divide em duas categorias, fica fácil entender que fatos estão relacionados a dimensões. No exemplo a seguir, para um único produto existem diversas vendas, ou seja, existe uma relação que envolve vendas e tabelas de produto. Ao desenvolver um esquema de estrela, colocando todas as dimensões ao redor da tabela fato única, obtém-se a Figura 4.

Figura 4: Exemplo de um esquema de estrela de vendas.



Fonte: Ferrari e Russo (2017).

2.4 INTEGRAÇÃO PYTHON E POWER BI®

É possível executar *scripts* do Python diretamente no Power BI® e importar os conjuntos de dados resultantes para um modelo de dados do Power BI Desktop®. Com base nesse modelo, pode-se criar e compartilhar relatórios. A integração exige a instalação dos dois pacotes do Python (Pandas e NumPy). Para isso, em um console ou *shell*, utiliza-se a ferramenta de linha de comando ‘pip’ para instalar os pacotes. A ferramenta ‘pip’ é empacotada com as versões mais recentes do Python.

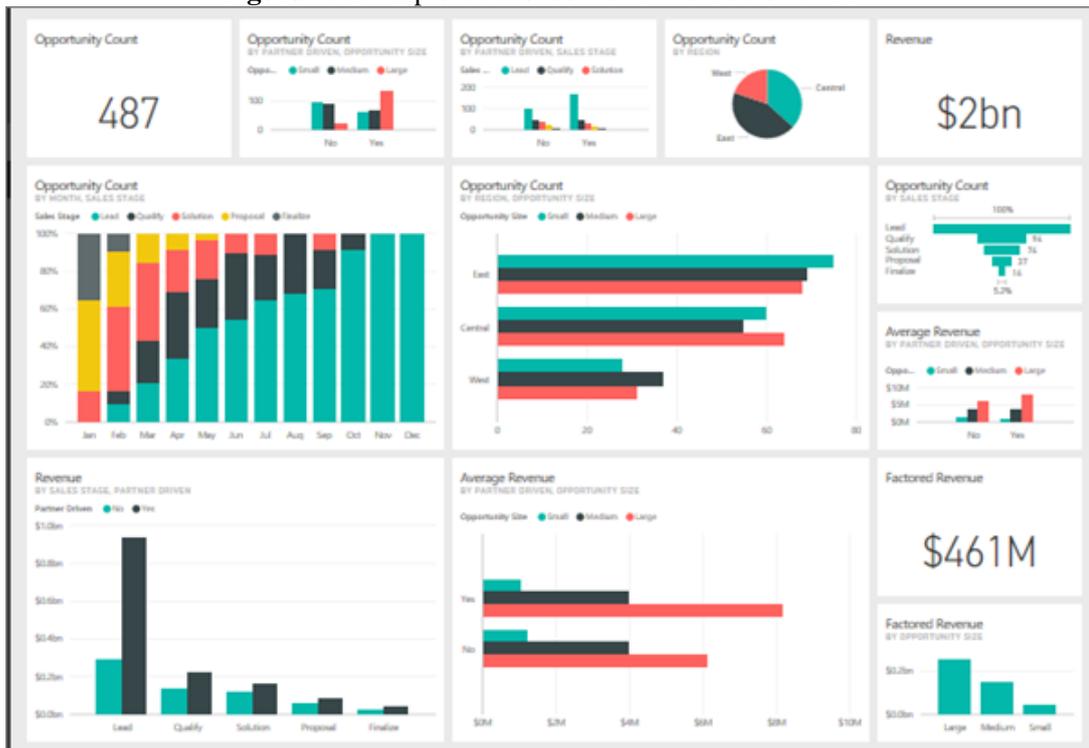
Para habilitar o *script* Python no Power BI®, deve-se seguir os seguintes passos (MICROSOFT, 2023c):

1. No Power BI®, selecione Arquivo > Opções e Configurações > Opções > *Script* do Python. A página Opções de *scripts* do Python é exibida;
2. Se necessário, deve-se fornecer ou editar o caminho de instalação local do Python em Diretórios base do Python detectados. Se há mais de uma instalação local do Python, deve-se selecionar a que deseja usar;
3. Selecione OK.

2.5 DASHBOARD

Dashboards fornecem visualizações poderosas de conjuntos de dados na forma de gráficos. Para O'Connor (2018), eles podem ser analisados em tempo real, tornando-os mais interativo para os usuários e permitindo que eles obtenham *insights* relevantes rapidamente. Ainda segundo o autor, essa ferramenta pode incluir *links* ou blocos para relatórios relacionados, o que ajuda os usuários a navegarem de um relatório para outro. Estas peças podem ser facilmente personalizadas de acordo com a preferência do usuário. Além de adicionar imagens, os usuários podem modificar a forma como eles estão dispostos no painel. A Figura 5 apresenta um exemplo de *dashboard* criado no Power BI®.

Figura 5: Exemplo de *dashboard* criado no Power BI®.



Fonte: Microsoft (2023a).

O *Dashboard* da Figura 5 traz os elementos mais importantes sobre as oportunidades de negócio de uma empresa em formato gráfico, como por exemplo: a quantidade de oportunidades divididas por critérios como sim e não, pelo tamanho do negócio ou pela região do negócio combinando esses critérios; a proporção do estágio das negociações em cada mês; e as receitas referentes a cada categoria.

Em geral, os *Dashboards* oferecem vários recursos analíticos importantes para o usuário, pois permitem que eles respondam à maioria das perguntas diretamente da interface do painel, sem precisar fazer login em outros pacotes ou módulos de *softwares* para fazer

análises mais profundas (RASMUSSEN; BANSAL; CHEN, 2009). Além disso, essa ferramenta pode contar com várias funções como análises estatísticas, mineração de dados, análise preditiva e indicadores chaves de desempenho que são capazes de capacitar gerentes e analistas para apoiar iniciativas de gestão de desempenho.

3 DASHBOARD ACADÊMICO CCECA

Nesta seção é apresentado o *dashboard* que foi desenvolvido, chamado de Dashboard Acadêmico CCECA (Colegiado do Curso de Engenharia de Controle e Automação). Em seguida são apresentados a ferramenta escolhida para o desenvolvimento do *dashboard*, a fonte de dados necessária, bem como o seu tratamento no Power BI®.

3.1 APRESENTAÇÃO E FUNCIONALIDADES

O Dashboard Acadêmico CCECA, Figura 6, foi desenvolvido para atender as necessidades do colegiado do curso de Engenharia de Controle e Automação de Processos no sentido de fornecer indicadores de desempenho estudantil através do histórico dos estudantes ativos, oferecer auxílio no planejamento de oferta de disciplinas e evidenciar quais alunos necessitam de orientação acadêmica.

Figura 6: Dashboard Acadêmico CCECA.



Fonte: Elaborado pelo autor.

Dessa forma, o dashboard apresentado possui as funções de identificar disciplinas que possuem maiores índices de aprovações, reprovações e trancamento; identificar disciplinas que possuem maior dependência dos pré-requisitos sugeridos na matriz curricular; mensurar o impacto da antecipação ou atraso nas inscrições das disciplinas sobre taxas de aprovação, reprovação e evasão; observar as relações descritas acima por período letivo; analisar o desempenho acadêmico por aluno e avaliar a sua trajetória e colaborar com o planejamento

acadêmico através da sugestão de até quatro disciplinas por aluno.

3.2 SELEÇÃO DO SOFTWARE

Levando em consideração os objetivos propostos neste trabalho, foram consideradas duas ferramentas de visualização de dados, o Power BI® e o Data Studio®. Com base nas informações contidas no referencial teórico e considerando que a licença *Free* do Power BI® já é suficiente para cumprir com os objetivos desse trabalho, optou-se pela utilização do Power BI®. Além disso, devido ao alto volume de dados para tratamento e cálculos complexos de variáveis, optou-se por utilizar um *script* Python como auxílio. Para a elaboração do código Python, foi utilizada a ferramenta Jupyter Notebook® para organizar o código em si e, então, vincular ao Power BI®.

Para mais informações sobre o *script* Python e integração entre Python e Power BI®, consultar os Apêndices A e B, respectivamente.

3.3 FONTE DE DADOS

A fonte de dados desse trabalho é dividida em três partes: históricos, disciplinas e pré-requisitos, e são encontradas em uma pasta específica chamada ‘Base PowerBI’.

3.3.1 Históricos

Os históricos são compostos por uma ou várias planilhas do Excel®, disponibilizada pela Superintendência de Tecnologia da Informação da UFBA (STI), que contêm os dados de todos os alunos ativos e estão no formato xlsx. A estrutura dessa planilha é composta por catorze colunas, Figura 7. A natureza das disciplinas varia entre OB (Obrigatória) e OP (Optativa). Os resultados atingidos pelos alunos podem variar entre aprovação (AA, AM e AP), dispensa interna (DI), dispensa UFBA (DU), reprovação por falta (RF), reprovação por nota (RR e RM), trancamento (TR) e não concluído (NC), esse resultado é referente a menos de 0,1% dos dados e não foi utilizado nas análises deste trabalho. As planilhas de históricos utilizadas no desenvolvimento deste dashboard são referentes aos alunos ativos nos semestres de 2023.1 e 2023.2 e correspondem a um quantitativo de 426 estudantes.

Figura 7: Estrutura da planilha de históricos.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Matrícula	Curso	Início_Curso	Ingresso	Per_Letivo	Época	Discip	Início_Disc.	Nome da Disciplina	Carga_Hor	Crédito	Natureza	Nota	Resultado
2	201510915	188140	20192	20151	20151	Normal	MATA01	20151	GEOMETRIAANALÍTICA	68	0	OB		RF
3	201510915	188140	20192	20151	20151	Normal	MATA02	20151	CÁLCULO A	102	0	OB		RF
4	201510915	188140	20192	20151	20152	Normal	ENGG11	20151	ESTUDOS ESPECIAIS EM ENGENHARI	68	0	OP		RF
5	201510915	188140	20192	20151	20152	Normal	MATA38	20151	PROJETO DE CIRCUITOS LÓGICOS	68	0	OB	5.6	AP

Fonte: Elaborado pelo autor.

3.3.2 Disciplinas

A planilha de disciplinas é uma tabela baseada na matriz curricular, que traz algumas informações de todas as disciplinas obrigatórias do curso, composta por quatro colunas, Figura 8. As disciplinas foram classificadas em grupos da sua respectiva área (Automação, Cálculo, Controle, Física, Profissionalizante, Programação e Química – Apêndice D).

Figura 8: Estrutura da planilha de disciplinas.

	A	B	C	D
1	Cód. Disciplina	Nome da Disciplina	Tipo Disciplina	Semestre Disciplina
2	MATA01	Geometria Analítica	Cálculo	1
3	MATA02	Cálculo A	Cálculo	1
4	ENGF57	Introdução à Engenharia de Controle e Automação	Profissionalizante	1
5	QUIB50	Fundamentos de Química	Química	1
6	MATA38	Projeto de Circuitos Lógicos	Programação	2

Fonte: Elaborado pelo autor.

3.3.3 Pré-Requisitos

A planilha ‘Pré-Requisitos.xlsx’ traz a informação de pré-requisitos sugeridos na matriz curricular do curso de todas as disciplinas obrigatórias. Vale ressaltar que se trata de uma sugestão, já que o curso tem como premissa não possuir pré-requisitos. Essa tabela é dividida em duas abas, a aba ‘Geral’, que é uma aba de auxílio que mostra as informações condensadas e é dividida em sete colunas, Figura 9. Os pré-requisitos sugeridos das disciplinas foram coletados da seção de Conteúdo Programático dos Componentes Curriculares do curso de Engenharia de Controle e Automação de Processos, no site da Escola Politécnica da UFBA.

Figura 9: Estrutura da aba Geral da planilha de pré-requisitos.

	A			B	C	D	E	F	G
1	Carga Horária			Código	Componente	Pré-requisitos	Semestre		
2	T	P	Total						
3	68	34	102	ENGF57	Introd. à Eng. de Controle e Automação	-	1		
4	102	0	102	MATA02	Cálculo A	-	1		
5	68	0	68	MATA01	Geometria Analítica	-	1		
6	34	0	34	QUIB50	Fundamentos de Química	-	1		
7	68	34	102	FIS121	Física 1	-	2		
8	68	0	68	MATA07	Algebra Linear A	MATA01	2		
9	102	0	102	MATA03	Cálculo B	MATA01, MATA02	2		

Fonte: Elaborado pelo autor.

A segunda aba dessa planilha é chamada de ‘Pré-Requisitos’. Cada pré-requisito é destrinchado nessa tabela, onde cada um deles deverá ocupar uma linha. Essa aba é composta por catorze colunas. A coluna chamada ‘Ordem’ informa a ordem de cada pré-requisito, 1 para pré-requisitos diretos, 2 para pré-requisitos diretos dos pré-requisitos de ordem 1 e 3 para

pré-requisitos diretos dos pré-requisitos de ordem 2. A coluna ‘Pesos Apenas de 1ª Ordem’ são os pesos de cada pré-requisito de ordem 1:

$$P = \mathbf{1}/Q_{p1} \quad (4)$$

em que, P é o peso calculado do pré-requisito para uma dada disciplina e Q_{p1} é a quantidade de pré-requisitos de ordem 1 para essa dada disciplina;

A coluna ‘Pesos até 3ª Ordem – [(60,30,10), (66,34)]’ calcula os pesos atribuídos para cada pré-requisito, considerando peso de 60% para todos os pré-requisitos de ordem 1, peso de 30% para todos os pré-requisitos de ordem 2 e peso de 10% para todos os pré-requisitos de ordem 3. Se a disciplina não possuir pré-requisitos de ordem 3, os pesos para os pré-requisitos de ordem 1 são de 66% e de ordem 2 são de 34%. Se a disciplina não tiver pré-requisitos de ordem 2 e 3, o peso é distribuído igualmente entre os pré-requisitos de ordem 1. O cálculo desses pesos pode ser visto a seguir:

$$\begin{aligned} P_1 &= \frac{OU(0, 60; 0, 66; 1, 00)}{Q_{p1}} \\ P_2 &= \frac{OU(0, 30; 0, 34)}{Q_{p2}} \\ P_3 &= \frac{0, 10}{Q_{p3}} \end{aligned} \quad (5)$$

em que P_1 , P_2 e P_3 são os pesos calculados dos pré-requisitos de ordem 1, 2 e 3, respectivamente, para uma dada disciplina, Q_{p1} , Q_{p2} e Q_{p3} são as quantidades de pré-requisitos de ordem 1, 2 e 3, respectivamente, dessa disciplina;

A coluna ‘Pesos até 3ª Ordem – [(60,30,10), (66,34)] e Pesos de Semestre’ calcula o peso de cada disciplina da mesma forma como foi calculado para a coluna anterior, porém, para cada grupo de pré-requisitos de mesma ordem, foi realizada uma média ponderada, onde o peso para cada pré-requisito é igual ao semestre em que ele está alocado na matriz curricular. O cálculo desse peso pode ser visto a seguir:

$$\begin{aligned} P_1 &= \frac{OU(0, 60; 0, 66; 1, 00)}{Q_{p1}} * \frac{S}{\sum S_1} \\ P_2 &= \frac{OU(0, 30; 0, 34)}{Q_{p2}} * \frac{S}{\sum S_2} \\ P_3 &= \frac{0, 10}{Q_{p3}} * \frac{S}{\sum S_3} \end{aligned} \quad (6)$$

em que S é o semestre do pré-requisito em que se está calculando o peso e $\sum S_1$, $\sum S_2$ e $\sum S_3$ são os somatórios dos semestres dos pré-requisitos de ordem 1, 2 e 3, respectivamente. A

primeira parte da estrutura da aba ‘Pré-Requisitos’ da planilha ‘Pré-Requisitos.xlsx’ pode ser conferida na Figura 10.

Figura 10: Estrutura da aba Pré-Requisitos da planilha Pré-Requisitos.xlsx (a).

	A	B	C	D	E	F	G
1	Cód. Disciplina	Nome da Disciplina	Tipo Disciplina	Semestre Disciplina	Tem Pré-Requisito?	Pré-Requisitos	Cód. Pré-Requisito
29	ENGG03	MÉTODOS NUMÉRICOS NA ENGENHARIA	Profissionalizante	5	Sim	MATA04, MATA07	MATA04
30	ENGG03				Sim		MATA07
31	ENGG03				Sim		MATA03
32	ENGG03				Sim		MATA01
33	ENGG03				Sim		MATA02

Fonte: Elaborado pelo autor.

A segunda parte da estrutura da planilha anterior pode ser conferida na Figura 11.

Figura 11: Estrutura da aba Pré-Requisitos da planilha Pré-Requisitos.xlsx (b).

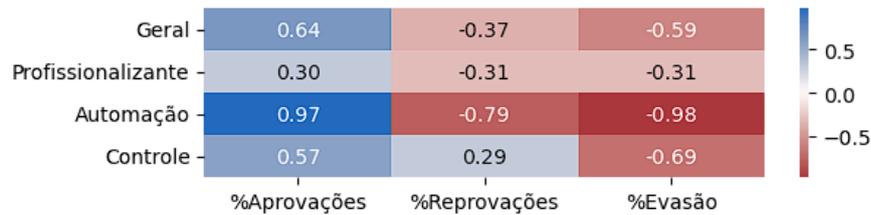
	A	G	H	I	J	K	L	M	N
1	Cód. Disciplina	Cód. Pré-Requisito	Nome do Pré-Requisito	Pré-Requisito do Pré-Requisito	Ordem	Semestre do Pré-Requisito	Pesos Apenas de 1ª Ordem	Pesos até 3ª Ordem - [(60,30,10), (66, 34)]	Pesos até 3ª Ordem - [(60,30,10), (66, 34)] e Pesos de Semestre
29	ENGG03	MATA04	CÁLCULO C	MATA03	1	3	0,500	0,300	0,360
30	ENGG03	MATA07	ÁLGEBRA LINEAR A	MATA01	1	2	0,500	0,300	0,240
31	ENGG03	MATA03	CÁLCULO B	MATA01, MATA02	2	2	0,000	0,150	0,200
32	ENGG03	MATA01	GEOMETRIA ANALÍTICA	-	2	1	0,000	0,150	0,100
33	ENGG03	MATA02	CÁLCULO A	-	3	1	0,000	0,100	0,100

Fonte: Elaborado pelo autor.

3.4 TRATAMENTO DE DADOS POWER BI®

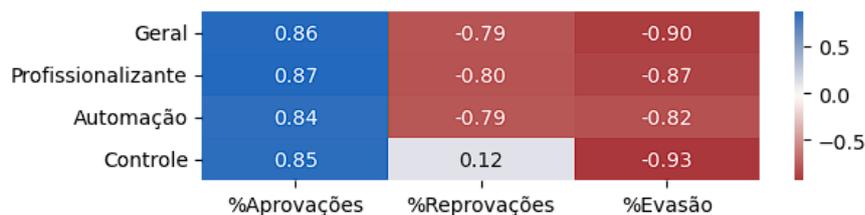
Após a importação dos dados, realizou-se o tratamento deles no editor do Power BI®. As tabelas importadas foram ‘aux_equivalencia’, ‘alunos’, ‘Historico’, ‘Historico_completo’, ‘sugestao’ e ‘vagas_planejamento’. Foi definido o tipo de dado de todas as colunas das tabelas e os tipos escolhidos foram: texto, número inteiro, percentual e número decimal.

As colunas ‘Ingresso’ e ‘Value’, da tabela ‘sugestao’, foram removidas pois não eram necessárias. Foi realizado um estudo de correlação entre o cumprimento de pré-requisito e os resultados aprovação, reprovação e evasão (reprovações por falta e trancamentos), onde a coluna ‘Cumprimento de pré-requisito 3’ teve a correlação mais forte. Sendo assim, foram removidas as colunas ‘Cumprimento de pré-requisito 1’, ‘Cumprimento de pré-requisito 2’ e ‘G_cump_pre2’ da tabela ‘Historico’. A Figura 12 apresenta a correlação entre os resultados e o ‘Cumprimento de pré-requisito 1’.

Figura 12: Correlação entre resultados X cumprimento de pré-requisitos de 1ª ordem.

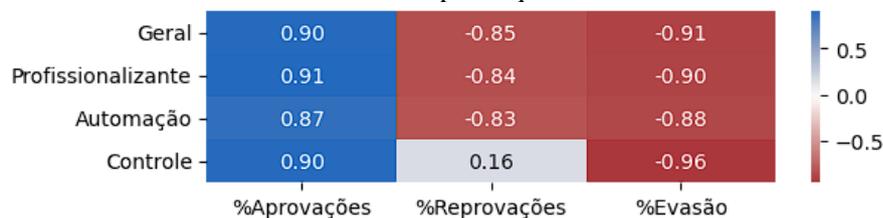
Fonte: Elaborado pelo autor.

A Figura 13 apresenta a correlação entre os resultados e o ‘Cumprimento de pré-requisito 2’.

Figura 13: Correlação entre resultados X cumprimento de pré-requisitos até 3ª ordem.

Fonte: Elaborado pelo autor.

A Figura 13 apresenta a correlação entre os resultados e o ‘Cumprimento de pré-requisito 3’.

Figura 14: Correlação entre resultados X cumprimento de pré-requisitos até 3ª ordem com peso de semestre de pré-requisito.

Fonte: Elaborado pelo autor.

Algumas tabelas tiveram suas colunas renomeadas para melhor entendimento. A tabela ‘alunos’ teve a coluna ‘Tx. Conclusão Disciplinas’ renomeada para ‘%Conclusão Disciplinas’; a tabela ‘Historico_completo’ teve as colunas ‘Discip.’ e ‘Nome da Disciplina’ renomeadas para ‘Cód. Disciplina’ e ‘Nome | Status’, respectivamente. As demais mudanças podem ser conferidas na Tabela 2 abaixo.

Tabela 2: Nomes antigos e novos das colunas renomeadas

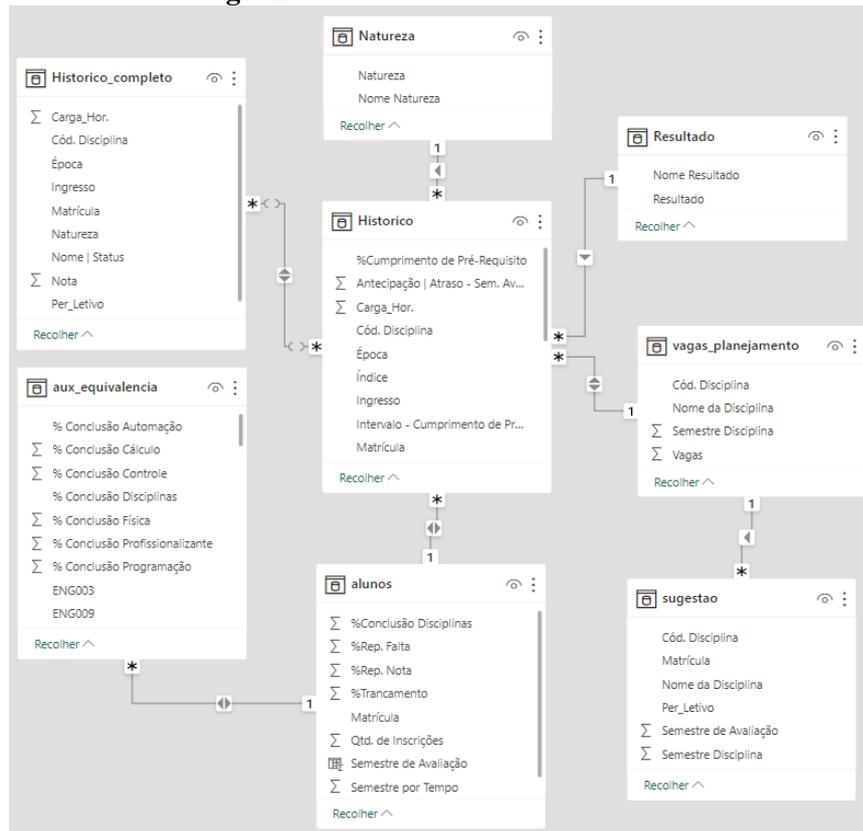
aux_equivalencia		Historico	
Nome antigo	Nome novo	Nome antigo	Nome novo
Tx. Conclusão Disciplinas	% Conclusão Disciplinas	Discip.	Cód. Disciplina
Tx. Conclusão Cálculo	% Conclusão Cálculo	Nome da Disciplina	Nome Status
Tx. Conclusão Física	% Conclusão Física	Cumprimento de pré-requisito 3	%Cumprimento de Pré-Requisito
Tx. Conclusão Programação	% Conclusão Programação	G_cump_pre3	Intervalo – Cumprimento de Pré-Requisito
Tx. Conclusão Profissionalizante	% Conclusão Profissionalizante		
Tx. Conclusão Automação	% Conclusão Automação		
Tx. Conclusão Controle	% Conclusão Controle		

Fonte: Elaborado pelo autor.

Foram acrescentadas duas colunas na tabela ‘Historico’, a primeira chamada ‘Índice’, incluída com a fórmula `{=Table.AddIndexColumn("#Colunas Renomeadas", "Índice", 1, 1, Int64.Type)}`, que enumera todas as linhas da tabela começando do valor 1 e variando uma unidade por linha, e a segunda coluna chama-se ‘Rank Per_Letivo’, acrescentada com a fórmula `{=Table.AddRankColumn("#Índice Adicionado", "Rank Per_Letivo", {"Per_Letivo", Order.Descending}, [RankKind = RankKind.Dense])}`, que ranqueia os períodos letivos do mais novo para o mais antigo, iniciando de 1.

Ao conjunto de tabelas foram acrescentadas duas tabelas auxiliares, a primeira chamada ‘Resultado’, que faz o link entre o código de cada resultado e a descrição dele, e a segunda chamada ‘Natureza’, que faz o link entre o código da natureza da disciplina e a descrição desse código. As tabelas desse trabalho foram relacionadas no modelo *star schema* e pode ser visto na Figura 15.

Figura 15: Relacionamento star schema.



Fonte: Elaborado pelo autor.

Para efeito de organização, foi criada uma tabela chamada 'Medidas', onde foram realizados os cálculos das medidas utilizadas nesse *dashboard*. Todas as funções DAX utilizadas para cálculo das medidas podem ser conferidas no Apêndice C deste trabalho.

4 GUIA DE USO DO DASHBOARD ACADÊMICO CCECA

Esta seção contém um guia para utilização do Dashboard Acadêmico CCECA, que pode ser baixado no link a seguir juntamente com a base de dados necessária para a sua utilização:

- <https://1drv.ms/f/s!Ahv2eDw0xo709A4TW0wx1eHYJGSY?e=HOIkzx>

4.1 REQUISITOS PARA FUNCIONAMENTO

Para o desenvolvimento deste *dashboard* foi utilizado o Power BI[®] em conjunto com o Python. Ao utilizar o *dashboard*, pressupõe-se que a máquina operante já possui o Python e o Power BI[®] instalados. Os dois podem ser baixados em seus respectivos sites oficiais, mas recomenda-se a instalação do Python através do pacote Anaconda.

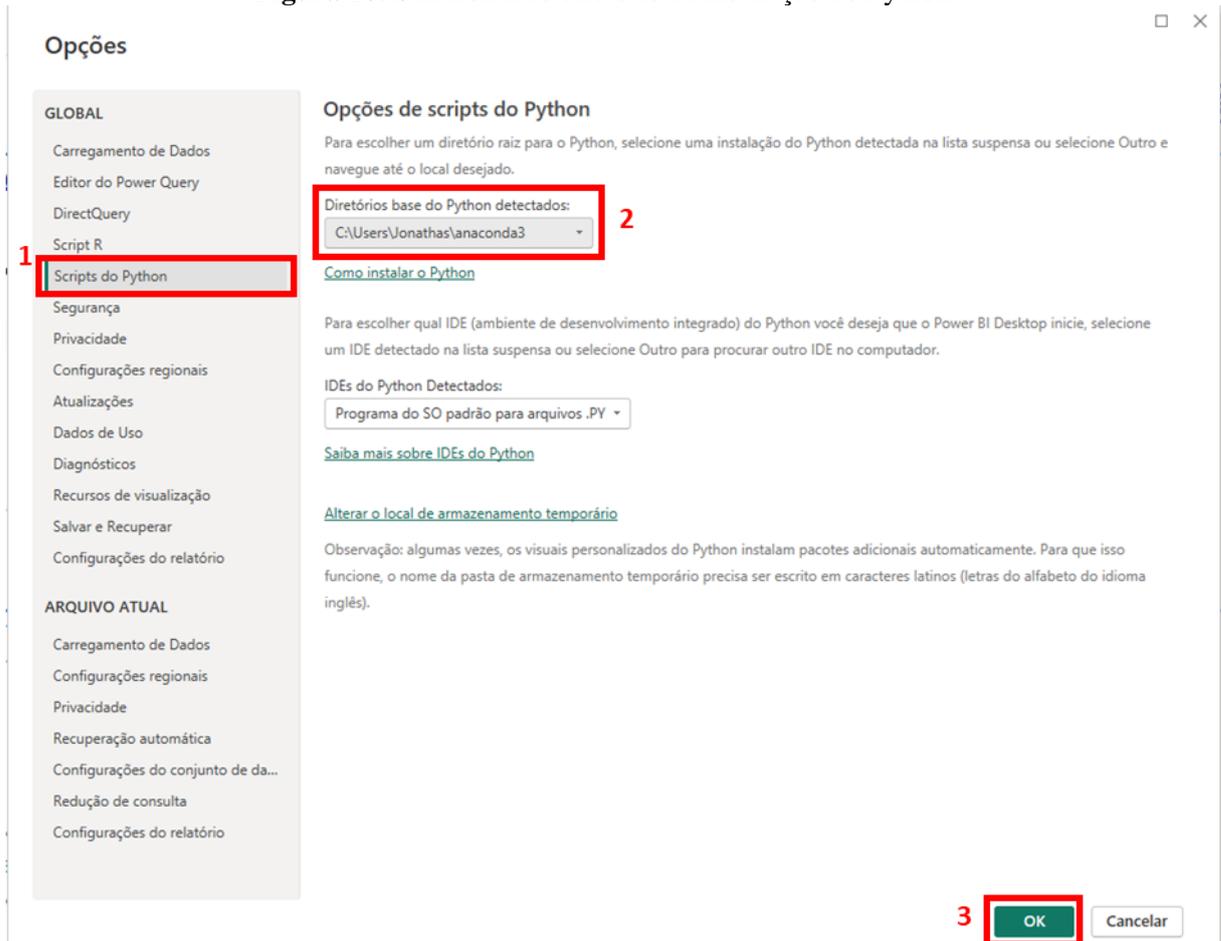
Segundo a Microsoft (2023d), os requisitos para funcionamento do Power BI[®] são:

- Windows 8.1 ou Windows Server 2012 R2 ou posterior;
- .NET 4.7.2 ou posterior;
- Navegador Microsoft Edge;
- Memória (RAM): ao menos 2 GB disponíveis, o recomendado são 4 GB ou mais;
- Exibição: Pelo menos 1440 x 900 ou 1600 x 900 (16:9) é exigido;
- CPU: é recomendado usar um processador de 1 GHz (gigahertz) e x64 (64 bits) ou superior.

4.2 PRIMEIRA UTILIZAÇÃO

Após se certificar de que todas as instalações foram realizadas, o usuário deve fazer o download do Dashboard Acadêmico CCECA e garantir que o Power BI[®] esteja integrado com o Python. De acordo com a Microsoft (2023b), a versão do Python que é suportada pelo Power BI[®] é a 3.7.7, porém a versão mais atual é a 3.12. Para evitar possíveis erros de compatibilidade, recomenda-se criar um ambiente virtual com a versão suportada pelo Power BI[®], seguindo o passo a passo apresentado na seção 7.1 do Apêndice B. Alternativamente, se o usuário optar por não criar o ambiente virtual, no Power BI[®] deve-se seguir o seguinte caminho: Arquivo > Opções e configurações > Opções > Scripts do Python. No campo ‘Diretórios base do Python detectados’, deve-se selecionar o local de instalação do Python e clicar em OK, Figura 16.

Figura 16: Selecionando diretório de instalação do Python.



Fonte: Elaborado pelo autor.

Em seguida, o usuário deve salvar a pasta com as fontes de dados do *dashboard* em seu computador respeitando algumas condições. A pasta deve ter o nome 'Base PowerBI' e deve ter a seguinte estrutura: 1) Uma pasta chamada 'Históricos' com todas as planilhas no formato 'xlsx' com o histórico de cada aluno; 2) Um arquivo chamado 'Disciplinas.xlsx' com as informações de todas as disciplinas obrigatórias e 3) Um arquivo chamado 'Pré-Requisitos.xlsx' com as informações de todos os pré-requisitos de cada disciplina obrigatória. As informações de cada planilha e sua estrutura podem ser vistas com mais detalhes na seção 3.3 deste trabalho. A Figura 17 mostra a estrutura da pasta 'Base PowerBI'.

Figura 17: Estrutura da pasta ‘Base PowerBI’.

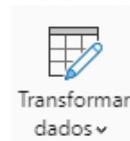
Nome	Status	Data de modificação	Tipo	Tamanho
Históricos	✓	30/10/2023 17:12	Pasta de arquivos	
Disciplinas	✓	29/10/2023 12:21	Planilha do Micro...	11 KB
Pré-Requisitos	✓	29/10/2023 18:37	Planilha do Micro...	44 KB

Fonte: Elaborado pelo autor.

As planilhas da pasta ‘Históricos’ são disponibilizadas pela STI da UFBA e devem iniciar seu nome com ‘Hist’ ou ‘hist’. Ao fim de cada semestre, quando os resultados dos alunos já estiverem no sistema UFBA, deve-se salvar a planilha com o histórico dos alunos na pasta ‘Históricos’, mantendo os arquivos anteriores. Caso seja necessário salvar uma nova planilha de histórico em um mesmo semestre, deve-se apagar a planilha mais antiga deste semestre e manter as planilhas dos semestres passados.

O usuário pode fazer alterações na planilha ‘Disciplinas.xlsx’, desde que mantenha a estrutura original do arquivo. Será possível também fazer alterações na planilha ‘Pré-Requisitos.xlsx’ adicionando pré-requisitos a uma disciplina obrigatória ou retirando-os, respeitando a estrutura original.

No Dashboard Acadêmico CCECA, o usuário deverá selecionar a opção ‘Transformar dados’ da seção Página Inicial, Figura 18.

Figura 18: Opção Transformar dados do Power BI®.

Fonte: Elaborado pelo autor.

Na janela que será aberta, o usuário deverá seguir os seguintes passos:

1. Selecionar a tabela ‘aux_equivalencia’ na seção de Consultas, Figura 19;

Figura 19: Selecionar a tabela desejada.

	A ^B _C Matrícula	A ^B _C Ingresso	A ^B _C Per_Letivo
1	201510915	20151	20151
2	201510915	20151	20152
3	201510915	20151	20161
4	201510915	20151	20162
5	201510915	20151	20171
6	201510915	20151	20172
7	201510915	20151	20181
8	201510915	20151	20182
9	201510915	20151	20191

Fonte: Elaborado pelo autor.

2. Clicar na opção Editor Avançado da seção Página Inicial, Figura 20;

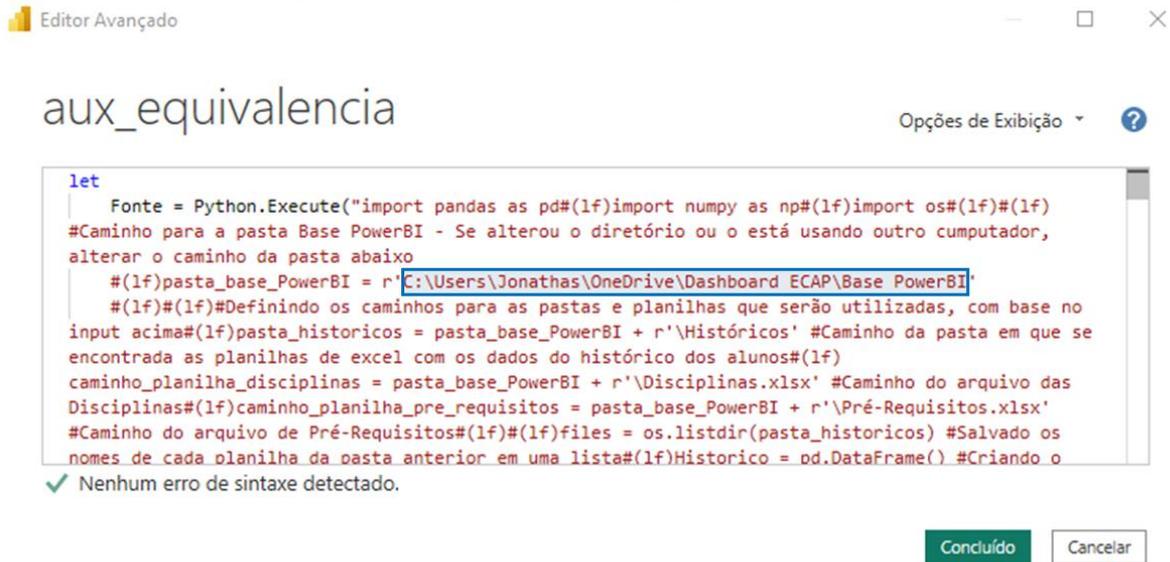
Figura 20: Selecionar opção Editor Avançado.

	A ^B _C Matrícula	A ^B _C Ingresso	A ^B _C Per_Letivo
1	201510915	20151	20151
2	201510915	20151	20152
3	201510915	20151	20161
4	201510915	20151	20162
5	201510915	20151	20171
6	201510915	20151	20172
7	201510915	20151	20181
8	201510915	20151	20182
9	201510915	20151	20191

Fonte: Elaborado pelo autor.

3. Na nova janela que se abriu, deve-se alterar o caminho selecionado na Figura 21, para o caminho da pasta 'Base PowerBI' do seu computador e clicar em Concluído;

Figura 21: Alterar o Caminho da pasta Base PowerBI.



Fonte: Elaborado pelo autor.

4. Repetir os passos de 1 a 3 para as tabelas 'alunos', 'Historico', 'Historico_completo', 'sugestao' e 'vagas_planejamento';
5. Quando finalizar o processo acima para as seis tabelas, clicar na opção Fechar e Aplicar da seção Página Inicial, Figura 22;

Figura 22: Opção Fechar e Aplicar.



Fonte: Elaborado pelo autor.

6. Após ter feito a alteração no código, deve-se permitir as alterações que serão solicitadas pelo Power BI®.
- Feito isso, será possível a utilização do dashboard.

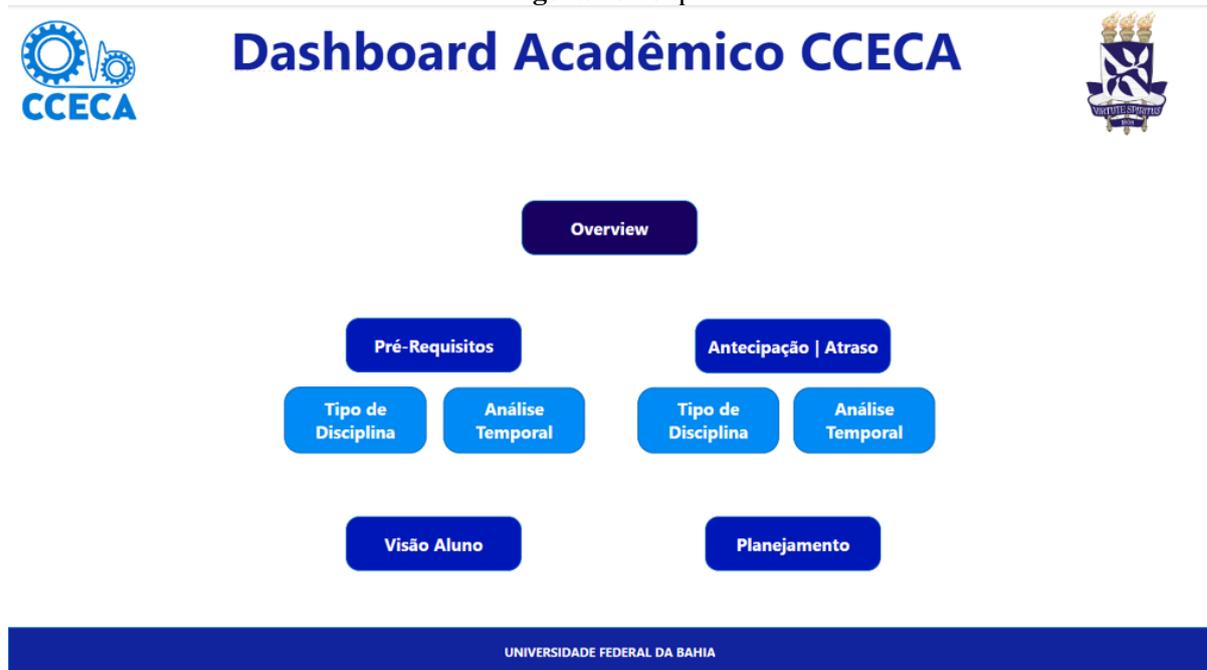
4.3 INTERFACES DO DASHBOARD ACADÊMICO CCECA

4.3.1 Capa

A primeira página do *dashboard* é a capa, Figura 23, nela encontra-se botões nomeados que levarão o usuário para outras páginas do *dashboard*. Os botões funcionarão quando o usuário segurar a tecla 'Ctrl' do teclado e clicar com o mouse. Alternativamente, o usuário pode acessar as outras páginas do *dashboard* selecionando-as na parte inferior da tela,

funcionando como abas.

Figura 23: Capa.



Fonte: Elaborado pelo autor.

4.3.2 Overview

A segunda página é chamada de 'Overview', Figura 24, nela o usuário pode ter algumas informações gerais do curso separadas em duas seções. Na parte superior é possível filtrar as informações conforme o semestre letivo, então é mostrada a quantidade de alunos ativos, a média da quantidade de disciplinas por aluno, a quantidade de alunos sem inscrição, a quantidade e proporção entre aprovações, reprovações por falta, reprovações por nota e trancamentos e a quantidade e proporção de disciplinas obrigatórias, optativas e eletivas. Na parte inferior, pode-se aplicar o filtro por tipo de disciplina e ver o *ranking* das disciplinas com maiores percentuais de aprovações, reprovações por falta, reprovações por nota e trancamentos.

Figura 24: Overview.



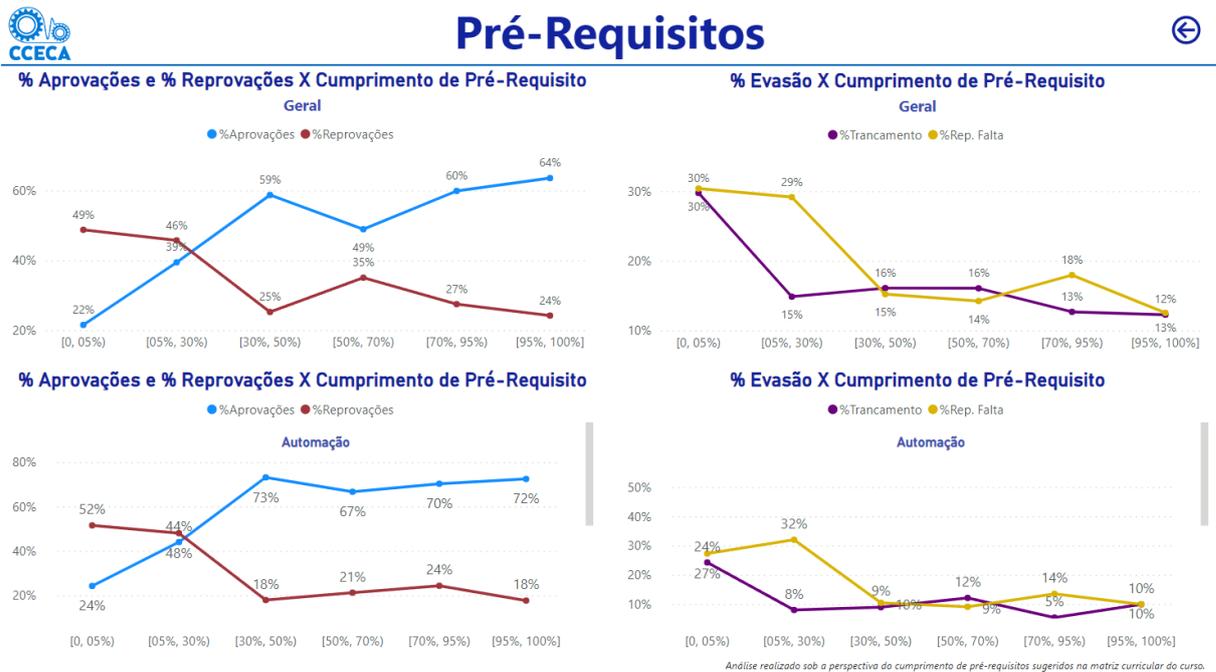
Fonte: Elaborado pelo autor.

4.3.3 Pré-Requisitos

Um conjunto de quatro gráficos compõem a terceira página chamada de 'Pré-Requisitos', Figura 25. Nesta tela será possível relacionar as taxas de aprovação, reprovação e evasão com o cumprimento de pré-requisitos. Neste exemplo o usuário poderá verificar que, quanto mais pré-requisitos sugeridos forem cumpridos, maiores serão os índices de aprovação e menores os índices de reprovação e evasão.

Os dois gráficos da esquerda mostram, para todo o conjunto de dados, as porcentagens de aprovações e reprovações pela quantidade de cumprimento de pré-requisitos sugeridos. O gráfico inferior contém uma barra de rolagem para mostrar três gráficos, um para cada tipo de disciplina a seguir: automação, controle e profissionalizante. Os gráficos da direita são análogos aos da esquerda, porém mostram as porcentagens de reprovações por falta e trancamento.

Figura 25: Pré-Requisitos.

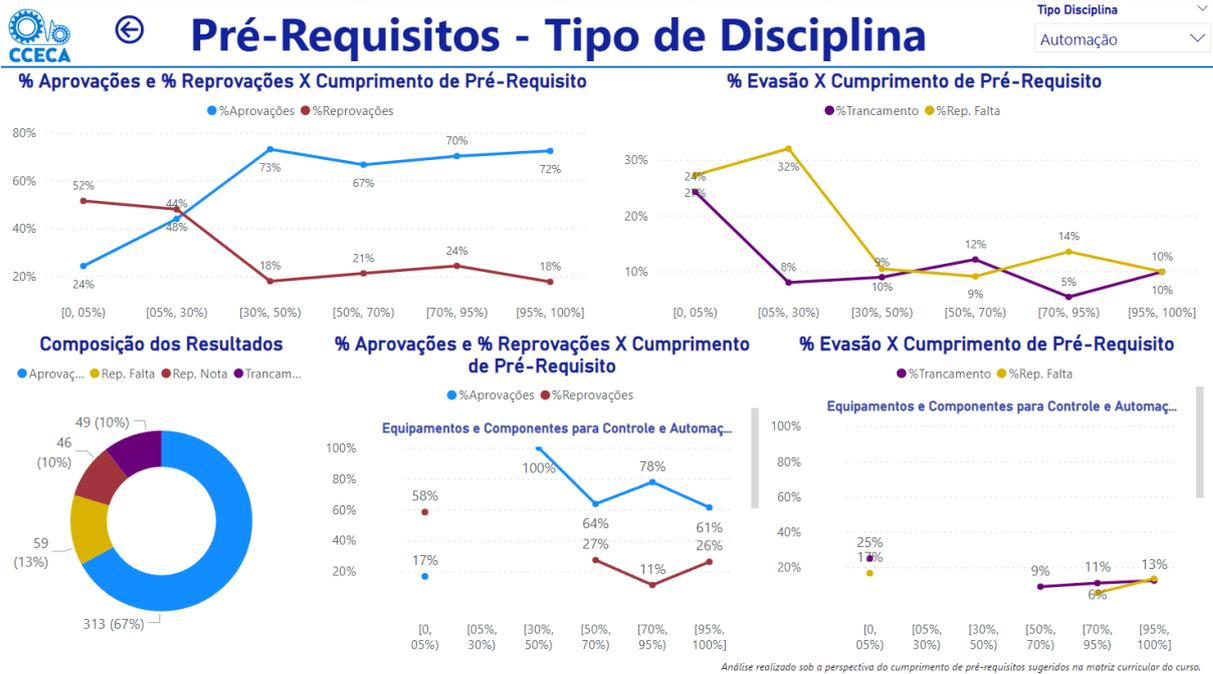


Fonte: Elaborado pelo autor.

4.3.4 Pré-Requisitos – Tipo de Disciplina

A quarta página, chamada de ‘Pré-Requisitos – Tipo de Disciplina’, Figura 26, permite verificar a relação entre o cumprimento de pré-requisitos sugeridos e os resultados, porém, com uma visão mais granular dos dados, possibilitando avaliar essa relação por disciplina. Assim como na página anterior, os gráficos superiores associam as porcentagens dos resultados com o cumprimento de pré-requisitos sugeridos. O gráfico inferior esquerdo mostra a proporção e a quantidade de aprovados, reprovados por falta, reprovados por nota e trancamentos. Os outros dois gráficos inferiores relacionam as porcentagens de resultado com o cumprimento de pré-requisitos sugeridos de cada disciplina, nesses gráficos existem uma barra de rolagem onde é possível ter acesso às disciplinas.

Figura 26: Pré-Requisitos - Tipo de Disciplina.

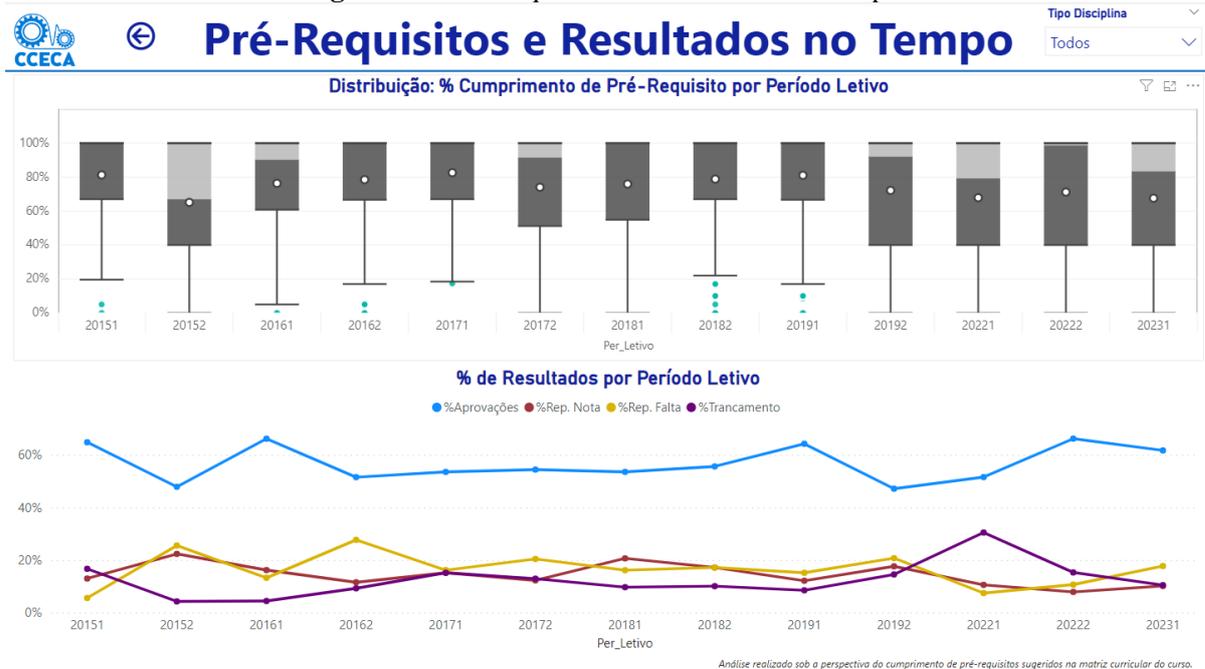


Fonte: Elaborado pelo autor.

4.3.5 Pré-Requisito e Resultados no Tempo

A quinta página se chama 'Pré-Requisito e Resultados no tempo', Figura 27. Nela é possível observar o histórico do curso através de análises estatísticas, que permitem identificar em quais semestres houve mais inscrições com baixa porcentagem de cumprimento de pré-requisitos sugeridos e a distribuição de resultados nesses semestres. Aqui também é possível filtrar ou não a página pelo tipo de disciplina. O gráfico superior mostra a distribuição do cumprimento de pré-requisitos sugeridos através do box plot para cada um dos treze últimos períodos letivos, com exceção dos semestres que ocorreram durante a pandemia de COVID19 (20201, 20202, 20211 e 20212, período em que os alunos estavam cursando as disciplinas remotamente e seus resultados não estavam sendo computados como aprovado e reprovado e sim como dispensa). Neste gráfico, espera-se que as caixas do box plot sejam as menores possíveis e mais próximas do 100%, pois representam uma menor variabilidade e maiores taxas de cumprimento de pré-requisitos. O gráfico inferior mostra as porcentagens de aprovação, reprovação por falta, reprovação por nota e trancamento para cada um dos períodos letivos do gráfico anterior.

Figura 27: Pré-Requisitos e Resultados no Tempo.



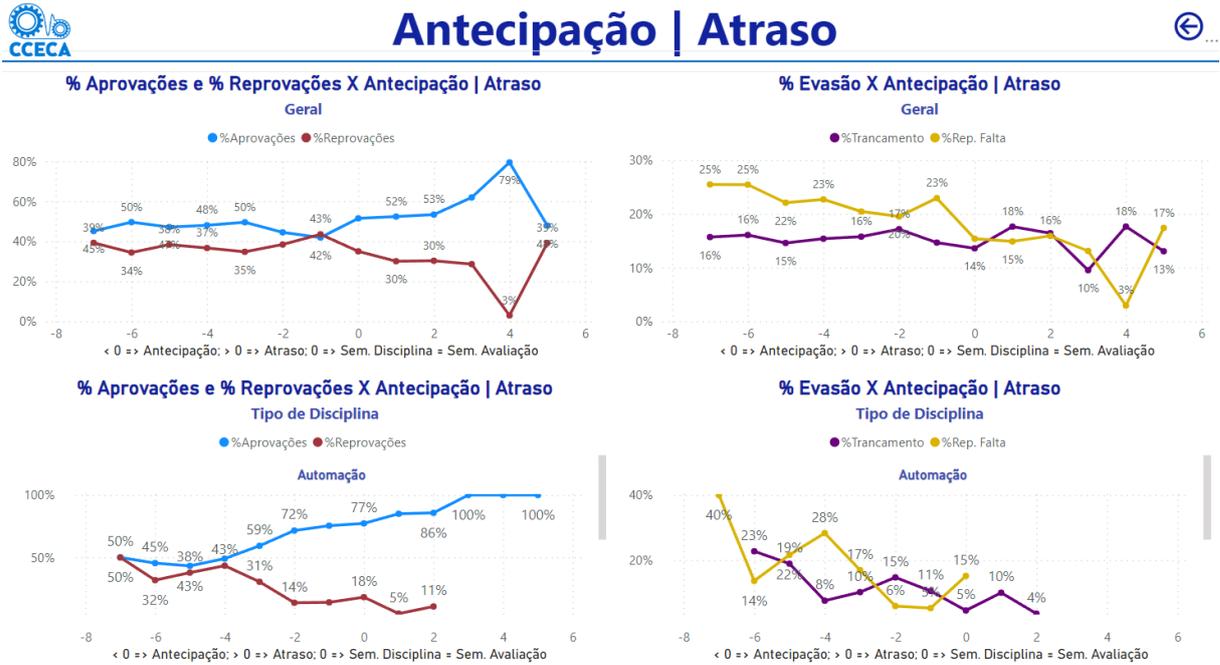
Fonte: Elaborado pelo autor.

Observa-se uma correlação positiva entre o cumprimento de pré-requisitos sugeridos e as aprovações, bem como uma correlação negativa entre esses pré-requisitos e as reprovações e evasão. Assim, pode-se inferir que alunos que cumprem as sugestões possuirão um melhor desempenho. Logo, é recomendado que o aluno siga a matriz curricular do curso. Vale ressaltar que o objetivo do colegiado é ser flexível e fornecer autonomia aos estudantes, cabendo aos estudantes analisar a viabilidade de cumprir ou não os pré-requisitos sugeridos.

4.3.6 Antecipação | Atraso

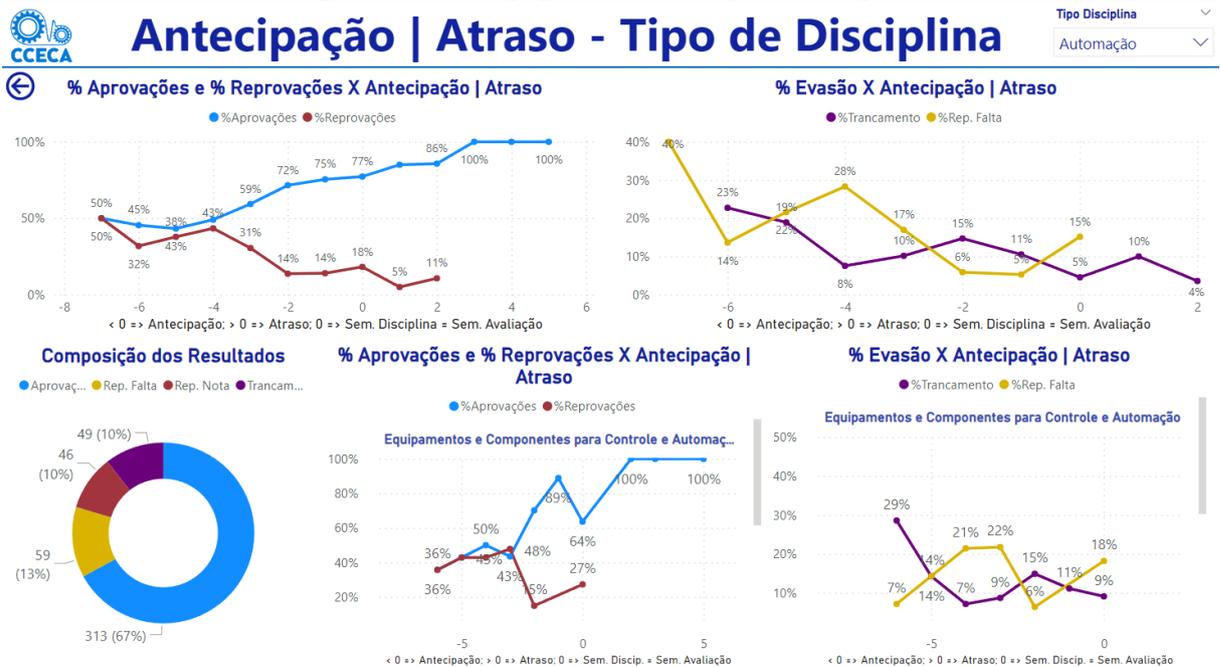
As páginas de 6 a 8 do *dashboard* são chamadas de ‘Antecipação | Atraso’, ‘Antecipação | Atraso – Tipo de Disciplina’ e ‘Antecipação | Atraso e Resultado no Tempo’, Figuras 28 a 30. Essas páginas são análogas as páginas 3 a 5 respectivamente, porém, analisa-se a variável antecipação ou atraso em relação ao semestre de avaliação do aluno (quantidade de semestres em sequência que o aluno cumpriu mais do que 50% da quantidade de disciplinas). Essa variável é calculada pela subtração entre o semestre de avaliação e o semestre da disciplina. Ou seja, se o resultado for negativo, quer dizer que o aluno antecipou a disciplina e o valor absoluto representa a quantidade de semestres antecipados. Se o valor for positivo, significa que o aluno atrasou a disciplina e o valor representa a quantidade de semestres atrasados. Se o valor for igual a zero, significa que o aluno se matriculou na disciplina em conformidade com o seu semestre de avaliação.

Figura 28: Antecipação | Atraso.



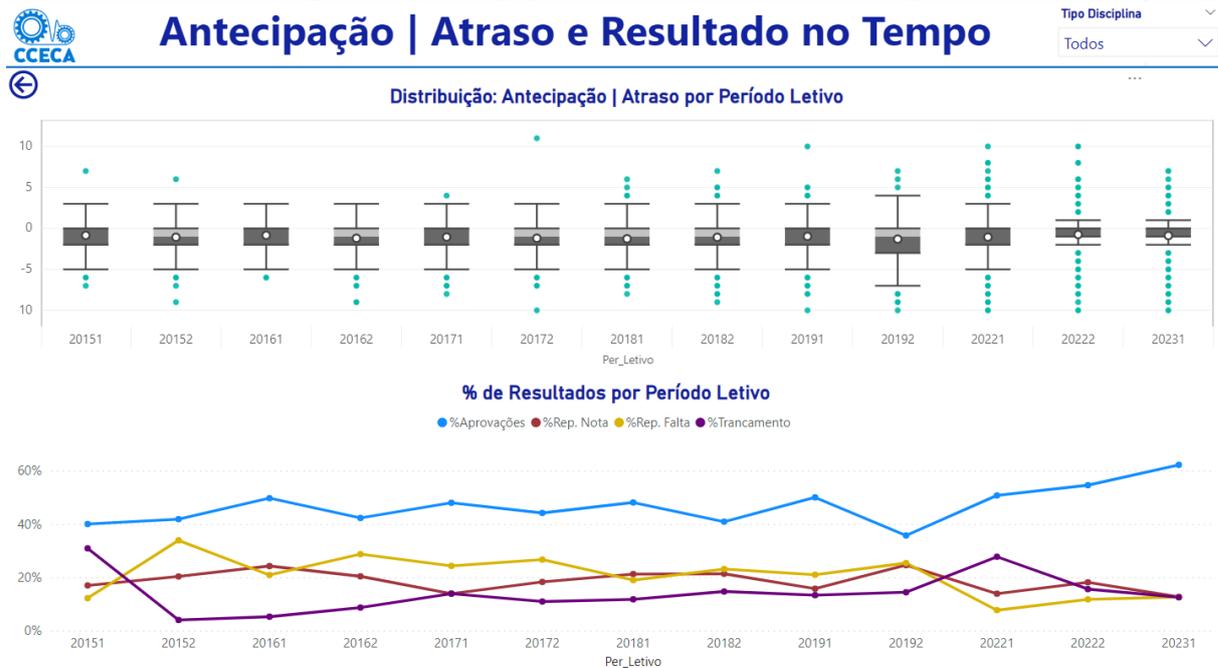
Fonte: Elaborado pelo autor.

Figura 29: Antecipação | Atraso - Tipo de Disciplina.



Fonte: Elaborado pelo autor.

Figura 30: Antecipação | Atraso e Resultado no Tempo.



Fonte: Elaborado pelo autor.

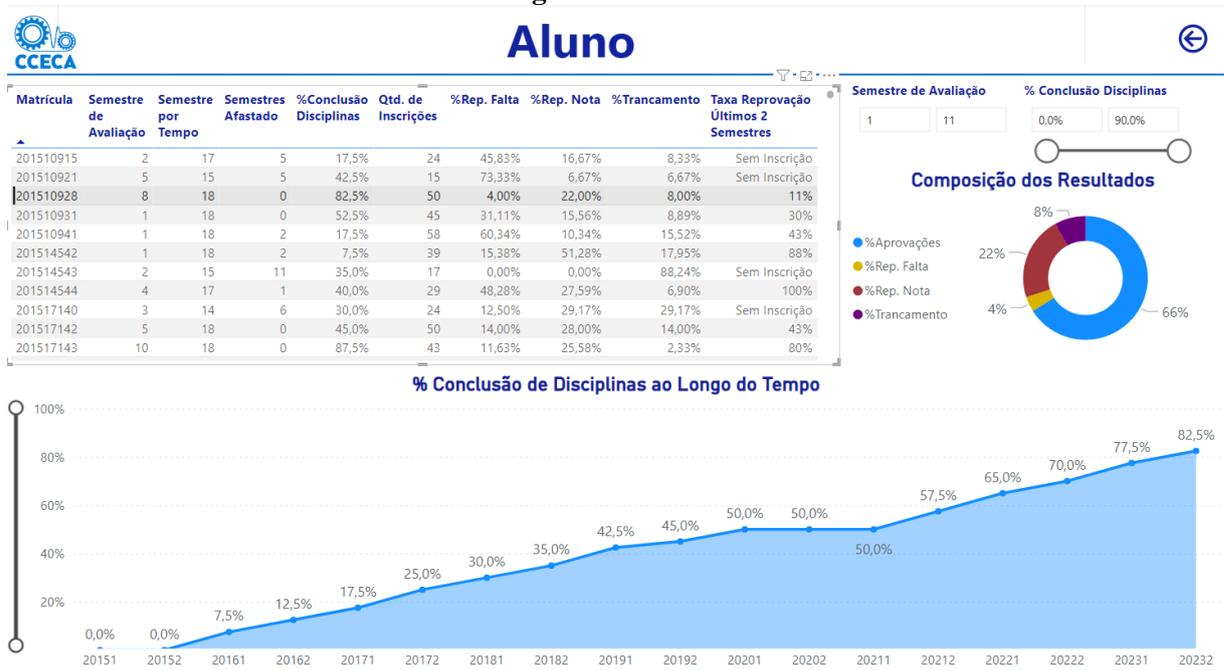
4.3.7 Aluno

A página nove, chamada 'Alunos', traz as métricas mais recentes sobre cada aluno do curso, Figura 31. O usuário pode filtrar a página pelo semestre de avaliação do aluno e pela porcentagem de conclusão de disciplinas. A tabela traz as colunas: 1) Matrícula: número de matrícula; 2) Semestre de Avaliação: semestre de avaliação mais recente; 3) Semestre por Tempo: quantidade de semestre corridos desde o início do aluno no curso; 4) Semestre afastados: quantidade de semestres que o aluno ficou afastado durante todo o seu período no curso (na Tabela 3 do Apêndice A pode ser visto os tipos de entrada no histórico em que foi considerado afastamento); 5) % Conclusão Disciplinas: mostra a porcentagem de conclusão de disciplinas; 6) Qtd. de Inscrições: quantidade de disciplinas que o aluno já se matriculou; 7) % Rep. Falta: porcentagem de reprovações por falta; 8) % Rep. Nota: porcentagem de reprovações por nota; 9) % Trancamento: porcentagem de trancamentos realizado pelo aluno e 10) Taxa de Reprovação Últimos 2 Semestres: mostra se nos últimos dois semestres o aluno se matriculou e, caso tenha se matriculado, apresenta a porcentagem de reprovações nesses dois semestres.

Nessa tabela, qualquer coluna poderá ser classificada, sendo possível estabelecer um ranking de alunos e identificar aqueles com desempenho inferior ao esperado. Além disso, quando selecionado um aluno na tabela, o gráfico de rosca mostrará a proporção entre as aprovações, reprovações por falta, reprovações por nota e trancamentos desse aluno. Já o

gráfico de área mostrará a evolução do cumprimento das disciplinas do aluno selecionado ao longo dos últimos períodos letivos. Nesse gráfico existe um controle deslizante onde é possível alterar os limites do eixo y para melhor visualização.

Figura 31: Aluno.



Fonte: Elaborado pelo autor.

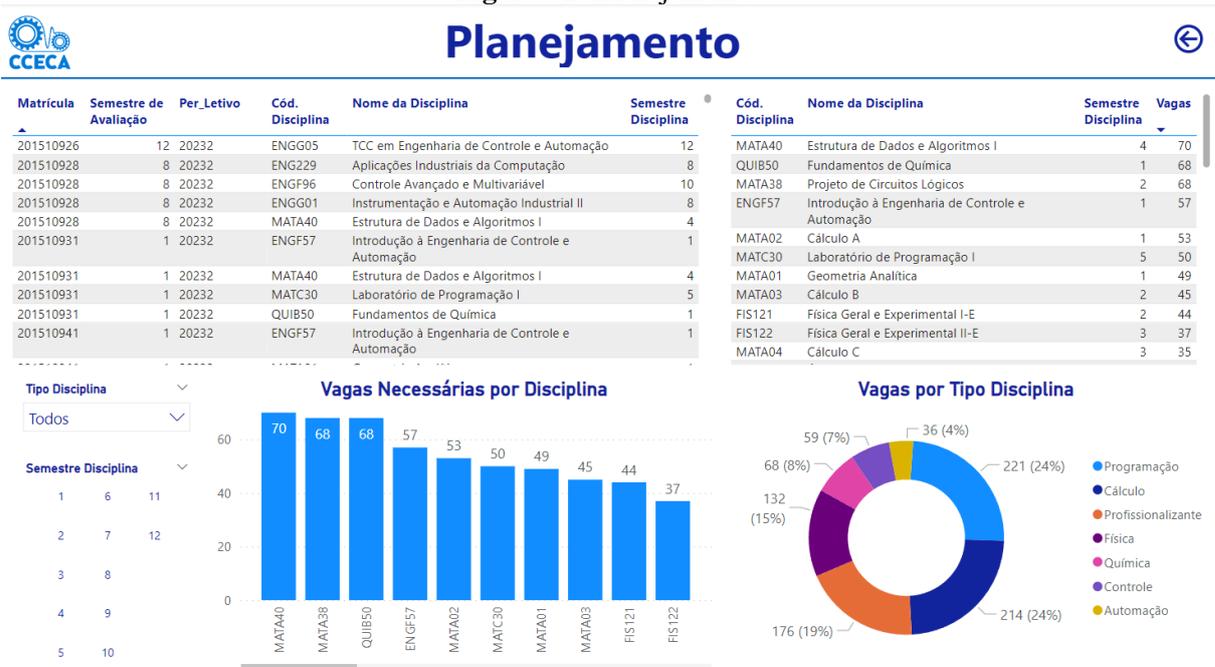
4.3.8 Planejamento

A décima e última página desse *dashboard* é chamada de ‘Planejamento’, Figura 32. A primeira tabela desta página é referente à sugestão de disciplinas e, para cada aluno, é sugerido um conjunto de, no máximo, quatro disciplinas obrigatórias com base naquelas que já foram concluídas e seguindo a ordem das disciplinas que compõem a matriz curricular. Essa tabela é composta por seis colunas, sendo elas: 1) Matrícula: código de matrícula; 2) Semestre de avaliação do aluno; 3) Per_Letivo: período letivo da sugestão (a sugestão para períodos futuros será realizada quando o período letivo atual já tiver resultados de conclusão de disciplinas); 4) Cód. Disciplina: código da disciplina que está sendo sugerida; 5) Nome da Disciplina: nome da disciplina que está sendo sugerida e 6) Semestre Disciplina: semestre da disciplina que está sendo sugerida.

A segunda tabela dessa página é referente a quantidade de vagas por disciplina que deve ser aberta para atender a todas as sugestões da tabela anterior. Essa tabela tem quatro colunas: 1) Cód. Disciplina: código da disciplina; 2) Nome da disciplina; 3) Semestre Disciplina e 4) Vagas: apresenta a quantidade de vagas que devem ser abertas para cada disciplina.

A parte inferior central exibe um gráfico de colunas ranqueando as disciplinas que mais necessitam de vagas conforme as informações da segunda tabela. O gráfico de rosca exibe a proporção de vagas que devem ser abertas por grupo de disciplinas. Na parte inferior esquerda estão dispostos dois filtros que atuam em toda a página, o primeiro por tipo de disciplinas e o segundo pelo semestre da disciplina. Ainda nessa página, o usuário poderá selecionar qualquer disciplina, seja no gráfico ou na segunda tabela, e verá todos os números de matrícula que receberam sugestão para essa disciplina. Em todas as páginas, exceto na capa, existe o botão ↶ que, ao clicar, faz retornar para a capa.

Figura 32: Planejamento.



Fonte: Elaborado pelo autor.

5 CONCLUSÃO

Neste trabalho foi apresentada uma ferramenta de gestão acadêmica em formato de *dashboard* que tem como objetivo reunir os principais indicadores de desempenho estudantil e dar suporte ao colegiado no processo de tomada de decisão, além de sugerir disciplinas aos estudantes e oferecer apoio ao planejamento acadêmico fundamentado nos resultados encontrados.

O *dashboard* fornece análises detalhadas sobre o desempenho dos estudantes em relação ao cumprimento de pré-requisitos sugeridos e outros fatores, como antecipação ou atraso em relação ao semestre de avaliação. Os gráficos mostram percentuais de aprovações, reprovações e outros resultados, e permitem filtrar os dados por grupos de disciplina. O *dashboard* oferece uma visão abrangente do desempenho dos alunos ao longo do tempo.

No que se refere a orientação acadêmica, são relacionados os principais indicadores que auxiliam na identificação de alunos com baixo desempenho, evidenciando quais estudantes necessitam de suporte personalizado. Além disso, pode-se observar a evolução do aluno ao longo dos semestres, sendo possível identificar prováveis abandonos.

Quanto ao planejamento acadêmico, é possível sugerir disciplinas para cada aluno com base naquelas que já foram cursadas e na sequência proposta pela matriz curricular, a fim de desenvolver a maturidade estudantil. Como consequência, gera-se uma lista das disciplinas para atender às necessidades dos alunos.

O Dashboard Acadêmico CCECA é uma ferramenta que segue as melhores práticas de interface amigável e pode ser alimentada com dados semestralmente, agregando valor às análises e informações que podem ser extraídas dele.

Assim, o *dashboard* é capaz de fornecer informações relevantes para a gestão acadêmica do colegiado. A limitação encontrada diz respeito às restrições de dados, notou-se a ausência de elementos sociais e pessoais, que poderiam enriquecer o resultado deste trabalho.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Para trabalhos futuros, recomenda-se atualizar a forma de calcular a sugestão de disciplinas para os alunos considerando também os pré-requisitos sugeridos já cursados, garantindo uma antecipação mais assertiva de disciplina. Para o planejamento, deve-se considerar disciplinas anuais e o histórico da quantidade de vagas que já vem sendo ofertadas, a fim de realizar um ajuste fino e definir a quantidade necessária.

Recomenda-se tornar esse dashboard uma plataforma online para que o acesso a essas

informações seja democratizado, fazendo também uma interface com plataformas de inteligência artificial.

6 REFERÊNCIAS

ALURA, 2023. **Linguagens de programação: uma breve introdução contextualizada**. Disponível em: <<https://www.alura.com.br/artigos/linguagem-programacao>>. Acesso em: 02 de novembro de 2023.

BONEL, C. **Power BI black belt: um treinamento faca na caveira através dos principais pilares de um projeto prático de business intelligence, usando o Microsoft Power BI**. 1ª ed., São Paulo: PerSe, 2019, 401p.

BUNGĂU, C.; POP, A. P.; BORZA, A. **Dropout of first year undergraduate students: A case study of engineering students**. Balkan Region Conference on Engineering and Business Education, Sibiu v. 2, n. 1, p. 349-356, 2017.

BUSSAB, D.O.; MORETTIN, P.A. **Estatística Básica**. 8 ed., São Paulo: Saraiva, 2014, 569p.

BUTTIGIEG, S.C; PACE, A.; RATHERT, C. **Hospital performance dashboards: a literature review**. Journal of Health Organization and Management, v. 31, n. 3, p. 385-406, 2017

CONDA, 2017. **Conda cheat sheet**. Disponível em: <https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf>. Acesso em: 31 de outubro de 2023.

CXTODAY, 2023. **Gartner magic quadrant for analytics and business intelligence platforms 2023**. Disponível em: <<https://www.cxtoday.com/data-analytics/>>. Acesso em: 26 de outubro de 2023.

DATA SCIENCE ACADEMY, 2023. **Top 25 Bibliotecas Python Para Data Science**. Disponível em: <https://blog.dsacademy.com.br/top-25-bibliotecas-python-para-data_science/>. Acesso em: 27 de novembro de 2023.

DEVORE, J. L. **Probabilidade e estatística: para engenharia e ciências**. 6ª ed., São Paulo: Cengage Learning, 2006, 692p.

ESTATMG, 2022. **Conheça As 5 Melhores Ferramentas Para Análise De Dados**. Disponível em: <<https://estatmg.com.br/2022/08/05/conheca-as-5-melhores-ferramentas-para-analise-de-dados/>>. Acesso em: 02 de novembro de 2023.

FAGUNDES, C. V.; LUCE, M. B.; ESPINAR, S. R. **O desempenho acadêmico como indicador de qualidade da transição Ensino Médio-Educação Superior**. Ensaio: avaliação e políticas públicas em educação, Rio de Janeiro, v. 22, n. 84, p. 635-670, 2014.

FERRARI, A; RUSSO, M. **Analyzing data with Microsoft Power BI and Power Pivot for Excel**. 1ª ed., Washington: Microsoft Press, 2017, 256p.

HYMAN, J.A. **Microsoft Power BI for dummies**. 1ª ed., Hoboken: John Wiley & Sons, 2022, 416p.

IDRIS, I. **NumPy: Beginner's Guide**. 3ª ed., Birmingham: Packt Publishing Ltd., 2015, 348p.

KRAUSE, K; COATES, H. **Students Engagement in first-year university**. Assessment and Evaluation in Higher Education, Mount Gravatt, v. 33, n. 5, p. 493-505, 2008.

MCKINNEY, W. **Python for Data Analysis**. 1ª ed., Sebastopol: O'Reilly Media, Inc., 2012, 463p.

MCKINNEY, W. **Python para análise de dados: tratamento de dados com pandas, numpy e jupyter**. 3ª ed., São Paulo: Novatec Editora, 2023, 624p.

MELO, A.C.V; SILVA, F.S.C. **Princípios de linguagens de programação**. 1ª ed., São Paulo: Blucher, 2003, 304p.

MICROSOFT, 2023a. **Introdução a dashboards para designers do Power BI**. Disponível em: <<https://learn.microsoft.com/pt-br/power-bi/create-reports/service-dashboards>>. Acesso em: 29 de outubro de 2023.

MICROSOFT, 2023b. **Saiba quais pacotes do Python têm suporte no Power BI**. Disponível em: <<https://learn.microsoft.com/pt-br/power-bi/connect-data/service-python-packages-support>>. Acesso em: 31 de outubro de 2023.

MICROSOFT, 2023c. **Executar scripts do Python no Power BI Desktop**. Disponível em: <<https://learn.microsoft.com/pt-br/power-bi/connect-data/desktop-python-scripts>>. Acesso em: 02 de novembro de 2023.

MICROSOFT, 2023d. **Obter o Power BI Desktop**. Disponível em: <<https://learn.microsoft.com/pt-br/power-bi/fundamentals/desktop-get-the-desktop>>. Acesso em: 04 de novembro de 2023.

MICROSOFT, 2023e. **O que é um dashboard de dados**. Disponível em: <<https://powerbi.microsoft.com/pt-pt/data-dashboards/>>. Acesso em: 05 de novembro de 2023.

MICROSOFT, 2023f. **Encontre o melhor plano do Microsoft 365**. Disponível em: <<https://www.microsoft.com/pt-br/microsoft-365>>. Acesso em: 07 de novembro de 2023.

MITRA, A. **Fundamentals of quality control and improvement**. 4ª ed., Nova Jersey: John Wiley & Sons, 2016, 791p.

MONTGOMERY, D.C. **Introduction to statistical quality control**. 7ª ed., Nova Jersey: John Wiley & Sons, 2012, 754p.

MONTGOMERY, D.C; RUNGER, G.C. **Estatística aplicada e probabilidade para engenheiros**. 4ª ed., Rio de Janeiro: LTC, 2009, 496p.

NUMPY, 2023. **NumPy documentation**. Disponível em: <<https://numpy.org/doc/stable/index.html>>. Acesso em: 27 de novembro de 2023.

O'CONNOR, E. **Microsoft Power Bi Dashboards: step by step**. 1ª ed., USA: Pearson Education, 2018, 304p.

OLIVEIRA, R.B.P.; LICHTNOW, D. **Uma Análise Comparativa das Ferramentas Microsoft Power BI e Google Data Studio**. Anais da XVII Escola Regional de Banco de Dados, Lages, p. 127-130, 2022.

PANDAS, 2023. **Package Overview**. Disponível em: <https://pandas.pydata.org/docs/getting_started/overview.html>. Acesso em: 27 de novembro de 2023

PERRONI, F. **Manual para a utilização do dashboard de gestão de pessoas da rede EBSEH**. Dissertação (Mestrado Profissional em Gestão e Inovação em Saúde) – Programa de Pós-graduação em Gestão e Inovação em Saúde, Universidade Federal do Rio Grande do Norte. Natal, p.128. 2021

PINHEIRO, S. M. C. **Uma abordagem dos modelos de longa duração para análise de sobrevivência da evasão de estudantes em cursos de engenharia: EPUFBA como um estudo de caso**. Tese (Doutorado em Engenharia Industrial) – Programa de Pós-graduação em Engenharia Industrial, Universidade Federal da Bahia. Salvador, p.138. 2021.

PYTHON, 2023. **Documentação Python 3.12.0**. Disponível em: <<https://docs.python.org/pt-br/3/>>. Acesso em: 27 de outubro de 2023.

RASMUSSEN, N.H; BANSAL, M; CHEN. C.Y. **Business Dashboards: A Visual Catalog for Design and Deployment**. 1ª ed., Nova Jersey: John Wiley & Sons, 2009, 411p.

ROY, A; DEB, R; ARORAA, G. **Self-Service Analytics with Power BI**. 1ª ed., Londres: BPB Online, 2023, 409p.

TIOBE, 2023. **TIOBE Index for October 2023**. Disponível em: <<https://www.tiobe.com/tiobe-index/>>. Acesso em: 02 de novembro de 2023.

UFBA. **Projeto político pedagógico do curso de graduação em engenharia de controle e automação de processos**. Salvador, novembro de 2022a.

UFBA. **Regulamento de ensino de graduação e pós-graduação**. Salvador, dezembro de 2022b.

VASILIEV, Y. **Python para ciência de dados: uma introdução prática**. 1ª ed., São Paulo: Novatec Edidora, 2023, 256p.

WHEELAN, C. **Estatística: o que é para que serve como funciona**. 1ª ed., Rio de Janeiro: Zahar, 2016, 328p.

7 APÊNDICE A: TRATAMENTO DOS DADOS COM O PYTHON

Foram escolhidas as bibliotecas *pandas*, *numpy* e *os*, Figura 33. A biblioteca *pandas* é voltada para lidar com *Data Frame* (DF), que é uma estrutura bidimensional tabular com rótulos de linhas e colunas, com uma gama de funções para realizar manipulação e tratamento de dados de uma forma rápida e fácil (MCKINNEY, 2012). A biblioteca *numpy*, que é uma abreviação de *numerical Python*, possui uma longa lista de funções matemáticas úteis, incluindo funções de álgebra linear, transformação de Fourier e rotinas para geração de números aleatórios (IDRIS, 2015). A biblioteca *os* fornece uma maneira fácil de escrever, ler e criar arquivos e manipular diretórios (PYTHON, 2023).

Figura 33: Bibliotecas.

```
import pandas as pd
import numpy as np
import os
```

Fonte: Elaborado pelo autor.

O algoritmo se inicia ao salvar em uma variável o diretório da pasta ‘Base PowerBI’ fornecida pelo usuário, nela contém todos os dados necessários para realização dos cálculos e tratamento. Em seguida, em outras três variáveis são salvos os diretórios da pasta Históricos e dos arquivos ‘Disciplinas.xlsx’ e ‘Pré-Requisitos.xlsx’.

Criou-se uma lista chamada *files* com o nome de cada planilha dentro da pasta ‘Históricos’ e três DFs, um vazio chamado Históricos e outros dois chamados *df_disciplinas*, que recebeu a planilha ‘Disciplinas.xlsx’, e *pre_requisitos*, que recebeu a planilha ‘Pré-Requisitos.xlsx’, Figura 34.

Figura 34: Definindo diretórios e criando DFs.

```
#Caminho para a pasta Base PowerBI - Se alterou o diretório ou o está usando outro computador, alterar o caminho da pasta abaixo
pasta_base_PowerBI = r'C:\Users\Jonathas\OneDrive\Dashboard ECAP\Base PowerBI'

#Definindo os caminhos para as pastas e planilhas que serão utilizadas, com base no input acima
pasta_historicos = pasta_base_PowerBI + r'\Históricos' #Caminho da pasta em que se encontrada as planilhas de excel com os dados do histórico dos alunos
caminho_planilha_disciplinas = pasta_base_PowerBI + r'\Disciplinas.xlsx' #Caminho do arquivo das Disciplinas
caminho_planilha_pre_requisitos = pasta_base_PowerBI + r'\Pré-Requisitos.xlsx' #Caminho do arquivo de Pré-Requisitos

files = os.listdir(pasta_historicos) #Salvado os nomes de cada planilha da pasta anterior em uma lista
Historico = pd.DataFrame() #Criando o data frame que receberá os dados das planilhas

#Salvando data frame das disciplinas e de pré-requisitos
df_disciplinas = pd.read_excel(caminho_planilha_disciplinas)
pre_requisitos = pd.read_excel(caminho_planilha_pre_requisitos, sheet_name = 'Pré-Requisitos')
```

Fonte: Elaborado pelo autor.

A Figura 35 mostra que na variável `files_xlsx` foi concatenado o diretório da pasta ‘Base PowerBI’ com cada valor dentro da lista `files`, a fim de salvar o caminho de cada planilha que está na pasta ‘Históricos’. Nessa etapa foi criado um verificador de segurança para que apenas as planilhas em que o nome comece com “Hist” ou “hist” tenha o seu caminho salvo. Em seguida, através de um laço de repetição, para cada planilha dentro da pasta ‘Históricos’, padronizou-se as colunas com o mesmo nome e os dados foram concatenados no DF Historico.

Figura 35: Concatenando dados de histórico.

```
#Concatenando o caminho da pasta e o nome de cada planilha em uma lista. Aqui inclui também
#um teste para selecionar apenas planilhas em que seu nome comece com "hist" ou "Hist"
files_xlsx = [pasta_historicos + '\\ ' + f for f in files if f.lower().startswith('hist')]

#Concatenando cada arquivo no data frame Historico
Historico = pd.read_excel(files_xlsx[0])
colunas = Historico.columns #Criando uma lista com o nome das colunas da primeira planilha
for f in range(len(files_xlsx)-1):
    data = pd.read_excel(files_xlsx[f+1]) #Salvando os dados da planilha seguinte
    data = data.set_axis(colunas, axis = 1) #Alterando o nome das colunas para ficar igual à primeira planilha
    Historico = pd.concat([Historico, data], ignore_index = True) #Concatenando as planilhas
```

Fonte: Elaborado pelo autor.

Na Figura 36 foram tratados os dados do DF Historico, onde foram excluídas colunas que são desnecessárias para este trabalho. Então, foram criados DFs auxiliares a partir do DF Historico, que ajudaram no cálculo de variáveis ao longo do algoritmo, são eles: 1) aux_equivalencia: no qual considerou-se apenas as colunas Matrícula, Ingresso e Período Letivo de cada aluno; 2) Historico_completo: cópia exata de Historico e 3) alunos_sem_inscricao: onde foram consideradas apenas as linhas em que não tem disciplina registrada para o aluno. Além disso, também foi criada uma lista com todos os períodos letivos que estão presentes no DF Historico.

Figura 36: Criando DFs Auxiliares e Tratando dados.

```
#Retirando colunas que não serão utilizadas na análise e renomeando as que sobraram
Historico.drop(['Curso', 'Inicio_Curso', 'Inicio_Disc.', 'Crédito'], axis = 1, inplace = True)
Historico = Historico.set_axis(['Matrícula', 'Ingresso', 'Per_Letivo', 'Época', 'Discip.',
                              'Nome da Disciplina', 'Carga_Hor.', 'Natureza', 'Nota',
                              'Resultado'], axis = 1)

#Criando um data frames auxiliares para fazer o tratamento e mineração
aux_equivalencia = Historico.copy()
Historico_completo = Historico.copy()
alunos_sem_inscricao = Historico.copy()
aux_equivalencia.drop(['Época', 'Discip.', 'Nome da Disciplina', 'Carga_Hor.', 'Natureza',
                      'Nota', 'Resultado'], axis = 1, inplace = True) #Retirando colunas do data frame auxiliar
alunos_sem_inscricao = alunos_sem_inscricao[-alunos_sem_inscricao['Discip.'].notna()] #Considerando apenas as linhas que não tem código de disciplina

#Criando uma lista com todos os períodos letivos, atual e passados
lista_periodos = Historico[Historico['Discip.'].notna()]['Per_Letivo'].sort_values().unique()
```

Fonte: Elaborado pelo autor.

Continuou-se realizando o tratamento de dados na Figura 37. Para padronização do DF Historico, os resultados do tipo AA e AM foram alterados para AP e os resultados do tipo RM foi alterado para RR. Além disso, também foram retirados do DF os resultados TR referentes ao semestre de 20201 que, por conta da pandemia do COVID19, todos os alunos tiveram suas matrículas trancadas forçadamente. Ainda no DF Historico, foram excluídas todas as linhas em que não continha valores na coluna de resultados. Após isso, foram excluídas todas as duplicatas dos quatro DFs.

Figura 37: Tratamento dos dados.

```
#Tratando dados do data frame historicos
Historico['Resultado'].replace(['AA','AM'],'AP', inplace = True) #Alterando resultado (AA e AM) para AP
Historico['Resultado'].replace('RM','RR', inplace = True) #Alterando resultado RM para RR
Historico = Historico[-((Historico['Per_Letivo'] == 20201) & (Historico['Resultado'] == 'TR'))] #Retirando resultados 'TR' no semestre 20201. Nesse semestre os...
#...trancamentos foram forçados por conta da pandemia.

Historico.dropna(subset = ['Resultado'], inplace = True) #Excluindo linhas vazias da coluna Resultado

#Excluindo linhas repetidas nos data frames
Historico = Historico.drop_duplicates(ignore_index = True)
aux_equivalencia = aux_equivalencia.drop_duplicates(ignore_index = True)
Historico_completo = Historico_completo.sort_values('Resultado').drop_duplicates(subset = ['Matrícula','Per_Letivo','Época','Discip.','Nota'], ignore_index = True)
#Na linha acima, classificando por resultado para manter as linhas com resultado e excluir as sem resultado
alunos_sem_inscricao = alunos_sem_inscricao.drop_duplicates(ignore_index = True)
```

Fonte: Elaborado pelo autor.

Na Figura 38 foram identificados o número de matrícula e semestre de ingresso dos alunos que estão matriculados no semestre corrente, esses dados foram concatenados ao DF aux_equivalencia com o período letivo equivalente ao semestre seguinte para, futuramente, sugerir disciplinas para cada aluno e auxiliar no planejamento de alocação de vagas das disciplinas obrigatórias do curso. Criou-se as colunas Semestre por Tempo (quantidade de semestres corridos desde a matrícula do aluno), Semestre de Avaliação (quantidade de semestres em sequência que o aluno cumpriu mais do que 50% da quantidade de disciplinas), Tx. Conclusão Disciplinas, Tx. Conclusão para cada tipo de disciplina e uma coluna para cada disciplina no DF aux_equivalencia, que receberam o valor inicial igual a 0. Nessa etapa também foi criado um dicionário para relacionar cada disciplina nova de física (teórica e prática) com as disciplinas antigas de física.

Figura 38: Organizando DF aux_equivalencia.

```
#Adicionando Linhas ao data frame de equivalência referente ao próximo semestre após o atual para fazer análise de planejamento
alunos_ativos = Historico[Historico['Per_Letivo'] == lista_periodos[-1]].loc[:,['Matrícula', 'Ingresso']].drop_duplicates()
#Na linha acima, salvando matrícula e semestre de ingresso dos alunos ativos do semestre mais atual.
alunos_ativos['Per_Letivo'] = lista_periodos[-2] + 10 #adicionando semestre seguinte para matrículas de alunos ativos no semestre atual para posteriormente...
#...calcular sugestões para alunos e planejamento para o colegiado de disciplinas do semestre seguinte.
aux_equivalencia = pd.concat([aux_equivalencia, alunos_ativos], ignore_index = True) #Concatenando a informação acima ao data frame de equivalencia

#Criando colunas para cálculo das informações no data frame auxiliar (cálculo de semestre de avaliação)
colunas = ['Semestre por Tempo', 'Semestre de Avaliação', 'Tx. Conclusão Disciplinas',
          'Tx. Conclusão Cálculo', 'Tx. Conclusão Física', 'Tx. Conclusão Programação',
          'Tx. Conclusão Profissionalizante', 'Tx. Conclusão Automação', 'Tx. Conclusão Controle']
disciplinas = df_disciplinas.sort_values('Semestre Disciplina')['Cód. Disciplina'].to_list() #Criando uma lista com as disciplinas obrigatórias a partir do...
#...df_disciplinas.
fisicas = {'FIS121': ['FISD36', 'FISD42'], 'FIS122': ['FISD34', 'FISD41'], #Criando dicionário com as disciplinas novas de física que foram separadas entre...
          'FIS123': ['FISD37', 'FISD40'], 'FIS124': ['FISD38', 'FISD39']} #...teórica e prática.

#Acrescentando colunas vazias (para cálculo posterior) ao data frame aux_equivalencia
aux_equivalencia[colunas] = 0 # Adicionando colunas com valores nulos
aux_equivalencia[disciplinas] = 0 # Adicionando colunas com valores nulos
```

Fonte: Elaborado pelo autor.

Para cada linha do DF Historico em que o aluno foi aprovado em disciplina obrigatória de qualquer período letivo ou dispensou uma disciplina obrigatória durante o período da pandemia do COVID19 (semestres 20201, 20202, 20211 e 20212, período em que os alunos estavam cursando as disciplinas remotamente e seus resultados não estavam sendo computados como aprovado e reprovado e sim como dispensa), foi registrado o valor 1 nos períodos seguintes à aprovação no DF aux_equivalencia. Para cada linha do DF Historico em que o aluno dispensou uma disciplina obrigatória fora do período da pandemia do COVID19, foi registrado o valor 1 no período da dispensa e nos períodos seguintes no DF aux_equivalencia. Quando se trata de física, o valor foi registrado nas novas disciplinas, independentemente de o aluno ter cursado no modelo antigo ou novo. Caso tenha cursado no modelo antigo, as duas disciplinas teórica e prática recebem o valor 1, Figura 39.

Figura 39: Preenchendo colunas de disciplinas no DF aux_equivalencia.

```
#Lógica para calcular em qual semestre os alunos foram aprovados e salvar no data frame de equivalência a aprovação nos semestres subsequentes.
AP_ou_DI_e_DU_pandemia = Historico[((Historico['Resultado'].isin(['DI', 'DU'])) & #Filtrando alunos que dispensaram disciplinas...;
                                   (Historico['Per_Letivo'].isin([20201,20202,20211,20212])) & #...no periodo da pandermia (Referente à aprovações nesse periodo)...
                                   (Historico['Discip.'].isin(disciplinas))) | #...considerando apenas disciplinas obrigatórias...
                                   ((Historico['Resultado'] == 'AP') & #...OU aprovações em qualquer semestre...
                                   (Historico['Discip.'].isin(disciplinas)))] #...considerando apenas disciplinas obrigatórias
for j in range(len(AP_ou_DI_e_DU_pandemia)):
#if abaixo, para casos das disciplinas antigas de física, que tem duas formas de medir: com disciplinas antigas (teoria e prática juntas) e com disciplinas novas...
#...(teoria e prática separadas)
    if AP_ou_DI_e_DU_pandemia.iloc[j,4] in ['FIS121', 'FIS122', 'FIS123', 'FIS124']:
        aux_equivalencia.loc[(aux_equivalencia['Matrícula'] == AP_ou_DI_e_DU_pandemia.iloc[j,0]) & #Salvando no data frame equivalência, para cada matrícula de...
                             (aux_equivalencia['Per_Letivo'] > AP_ou_DI_e_DU_pandemia.iloc[j,2]), #...aluno, considerando o filtro acima, nos semestres seguintes...
                             fisicas[AP_ou_DI_e_DU_pandemia.iloc[j,4]]] = 1 #...à aprovação, o número 1 na coluna da disciplina em questão.
    else: #Para disciplinas diferentes de física, seguindo a mesma lógica anterior.
        aux_equivalencia.loc[(aux_equivalencia['Matrícula'] == AP_ou_DI_e_DU_pandemia.iloc[j,0]) &
                             (aux_equivalencia['Per_Letivo'] > AP_ou_DI_e_DU_pandemia.iloc[j,2]),
                             AP_ou_DI_e_DU_pandemia.iloc[j,4]] = 1

#Lógica para calcular em qual semestre os alunos dispensaram alguma disciplina e salvar no data frame de equivalência
DI_DU_fora_pandemia = Historico[(Historico['Resultado'].isin(['DI', 'DU'])) & #Filtrando apenas alunos que dispensaram disciplinas...
                                -(Historico['Per_Letivo'].isin([20201, 20202, 20211, 20212])) & #...Excetuando as do periodo da pandemia que já foram consideradas...
                                (Historico['Discip.'].isin(disciplinas))] #...antes, considerando apenas disciplinas obrigatórias.
for j in range(len(DI_DU_fora_pandemia)):
    if DI_DU_fora_pandemia.iloc[j,4] in ['FIS121', 'FIS122', 'FIS123', 'FIS124']: #Para casos de disciplinas antigas de física
        aux_equivalencia.loc[(aux_equivalencia['Matrícula'] == DI_DU_fora_pandemia.iloc[j,0]) & #Salvando no data frame equivalência, para cada matrícula de aluno...
                             (aux_equivalencia['Per_Letivo'] >= DI_DU_fora_pandemia.iloc[j,2]), #..., considerando o filtro acima, no semestre da dispensa e nos...
                             fisicas[DI_DU_fora_pandemia.iloc[j,4]]] = 1 #...subsequentes, o número 1 na coluna da disciplina em questão.
    else: #Para disciplinas diferentes de física, seguindo a mesma lógica anterior.
        aux_equivalencia.loc[(aux_equivalencia['Matrícula'] == DI_DU_fora_pandemia.iloc[j,0]) &
                             (aux_equivalencia['Per_Letivo'] >= DI_DU_fora_pandemia.iloc[j,2]),
                             DI_DU_fora_pandemia.iloc[j,4]] = 1
```

Fonte: Elaborado pelo autor.

Seguindo o preenchimento das colunas de disciplinas do parágrafo anterior, cada disciplina antiga de física recebeu o valor 1 se as duas disciplinas novas de física (prática e teórica) possuírem valor 1. Em seguida, calculou-se outras colunas do DF aux_equivalencia: a coluna Semestre por Tempo, calculada pela subtração entre o Período Letivo e o Semestre de Ingresso e as colunas referentes à Tx. de Conclusão: calculadas através da média entre as colunas de disciplinas de cada grupo ou de todas as disciplinas no geral, isso porque cada disciplina recebe o valor de 1 quando concluída e 0 quando pendente, Figura 40.

Figura 40: Preenchendo colunas do DF aux_equivalencia.

```
#Atribuindo valores para as disciplinas de física, receberá valor 1 quando tanto a disciplina teórica quanto a prática tiverem valor 1
aux_equivalencia['FIS121'] = aux_equivalencia['FISD36']*aux_equivalencia['FISD42']
aux_equivalencia['FIS122'] = aux_equivalencia['FISD34']*aux_equivalencia['FISD41']
aux_equivalencia['FIS123'] = aux_equivalencia['FISD37']*aux_equivalencia['FISD40']
aux_equivalencia['FIS124'] = aux_equivalencia['FISD38']*aux_equivalencia['FISD39']

#Atribuindo valores para as colunas semestre por tempo e taxas de conclusão
aux_equivalencia['Semestre por Tempo'] = ((aux_equivalencia['Per_Letivo'] - aux_equivalencia['Ingresso']) // 10) * 2 + 1 + (
    (aux_equivalencia['Per_Letivo'] - aux_equivalencia['Ingresso']) % 10 > 0).apply(int)
aux_equivalencia['Tx. Conclusão Cálculo'] = aux_equivalencia[df_disciplinas[df_disciplinas['Tipo Disciplina'] == 'Cálculo']['Cód. Disciplina']].mean(axis = 1)
aux_equivalencia['Tx. Conclusão Física'] = aux_equivalencia[['FIS121', 'FIS122', 'FIS123', 'FIS124']].mean(axis = 1)
aux_equivalencia['Tx. Conclusão Programação'] = aux_equivalencia[df_disciplinas[df_disciplinas['Tipo Disciplina'] == 'Programação']['Cód. Disciplina']].mean(axis = 1)
aux_equivalencia['Tx. Conclusão Profissionalizante'] = aux_equivalencia[df_disciplinas[df_disciplinas['Tipo Disciplina'] == 'Profissionalizante']['Cód. Disciplina']].mean(
    axis=1)
aux_equivalencia['Tx. Conclusão Automação'] = aux_equivalencia[df_disciplinas[df_disciplinas['Tipo Disciplina'] == 'Automação']['Cód. Disciplina']].mean(axis = 1)
aux_equivalencia['Tx. Conclusão Controle'] = aux_equivalencia[df_disciplinas[df_disciplinas['Tipo Disciplina'] == 'Controle']['Cód. Disciplina']].mean(axis = 1)
aux_equivalencia['Tx. Conclusão Disciplinas'] = aux_equivalencia[df_disciplinas[-df_disciplinas['Cód. Disciplina'].isin(['FISD36', 'FISD42', 'FISD34',
    'FISD41', 'FISD37', 'FISD40', 'FISD38', 'FISD39'])]['Cód. Disciplina']].mean(axis = 1)
```

Fonte: Elaborado pelo autor.

Ainda no DF aux_equivalencia, o semestre de avaliação de cada aluno foi calculado considerando o maior semestre em sequência em que mais do que 50% das disciplinas foram cumpridas. Para o cálculo foram utilizados dois laços de repetição aninhados e, para cada linha do DF, testou-se os semestres de um a doze, sendo que, quando encontrado um semestre em que o aluno não cumpriu mais do que 50%, o laço de repetição interno era interrompido, Figura 41.

Figura 41: Calculando semestre de equivalência no DF aux_equivalencia.

```
#Atribuindo valores para a coluna de semestre de avaliação, para avançar o aluno deve ter concluído acima de 50% das disciplinas em cada semestre anterior.
for j in range(len(aux_equivalencia)):
    media = True
    semestre = 0
    while media == True and semestre < 12:
        semestre = semestre + 1
        media = aux_equivalencia.loc[j, df_disciplinas[(df_disciplinas['Semestre Disciplina'] == semestre) &
            -(df_disciplinas['Cód. Disciplina'].isin(['FISD36', 'FISD42', 'FISD34',
                'FISD41', 'FISD37', 'FISD40', 'FISD38',
                'FISD39']))]]['Cód. Disciplina']].mean() > 0.5
    aux_equivalencia.loc[j, 'Semestre de Avaliação'] = semestre #Atribuindo o semestre de avaliação do aluno
```

Fonte: Elaborado pelo autor.

Na Figura 42, no DF Historico acrescentou-se três colunas, Cumprimento de pré-requisito 1, 2 e 3, informando qual a porcentagem de pré-requisitos cumpridos por cada aluno quando se matriculou em uma dada disciplina. A primeira coluna se refere ao cumprimento de pré-requisitos diretos, a segunda se refere ao cumprimento dos pré-requisitos diretos e indiretos até a 3ª ordem, e a terceira coluna se refere ao cumprimento dos pré-requisitos diretos e indiretos até a 3ª ordem, considerando também o peso do semestre do pré-requisito. O cálculo foi realizado com auxílio de um laço de repetição *for* e as estruturas *if* e *else*. As disciplinas não obrigatórias e as disciplinas que não têm pré-requisitos receberam valor 1 nessas três colunas. Nessa etapa também foi adicionada uma coluna chamada “Tem Pré-Requisito?” ao DF Historico, essa coluna recebe Sim caso a disciplina tenha pré-requisito e Não caso contrário.

Figura 42: Calculando a porcentagem de cumprimento de pré-requisitos para cada inscrição em disciplinas do DF Historico.

```
#Calculando o cumprimento de pré-requisitos
Historico[['Cumprimento de pré-requisito 1', 'Cumprimento de pré-requisito 2', 'Cumprimento de pré-requisito 3']] = 0
#Na linha acima, acrescentando as colunas de cumprimento de pré-requisitos ao data frame histórico: o primeiro considerando apenas pré-requisitos de ordem 1,...
#segundo considerando pré-requisitos de até ordem 3 com os pesos [60, 30, 10] ou [66, 34] e terceiro, idem ao segundo, considerando também pesos para o semestre...
#do pré-requisito.
for j in range(len(Historico)):
    matricula = Historico.iloc[j,0]
    periodo = Historico.iloc[j,2]
    discip = Historico.iloc[j,4]
    if (discip not in disciplinas) or (pre_requisitos[pre_requisitos['Cód. Disciplina'] == discip]['Tem Pré-Requisito?'].iloc[0] == 'Não'):
        Historico.loc[j, ['Cumprimento de pré-requisito 1', 'Cumprimento de pré-requisito 2', 'Cumprimento de pré-requisito 3']] = 1
        #Na linha acima, atribuindo 1 para disciplinas não obrigatórias ou que não tem pré-requisito.
        Historico.loc[j, 'Tem Pré-Requisito?'] = 'Não'
        #Na linha acima, acrescentando uma coluna ao data frame que diz se a disciplina tem pré-requisito ou não, aqui atribuindo com "não".
    else:
        Historico.loc[j, ['Cumprimento de pré-requisito 1', 'Cumprimento de pré-requisito 2', 'Cumprimento de pré-requisito 3']] = (
            pre_requisitos[pre_requisitos['Cód. Disciplina'] == discip].iloc[:,[11,12,13]]*aux_equivalencia.loc[(aux_equivalencia['Matricula'] == matricula) &
            (aux_equivalencia['Per_Letivo'] == periodo),pre_requisitos[pre_requisitos['Cód. Disciplina'] == discip].iloc[:,6]].T.values).sum().values
        #Nas linhas acima, atribuindo a porcentagem de pré-requisitos cumpridos, multiplicando os pesos de cada pré-requisito por 1 (se concluiu) ou 0 (se não...
        #...concluiu), os pesos estão no data frame 'pre_requisitos' e a conclusão ou não está no data frame 'aux_equivalencia'
        Historico.loc[j, 'Tem Pré-Requisito?'] = 'Sim' #Atribuindo "sim" a coluna "Tem pré-requisito?".
```

Fonte: Elaborado pelo autor.

Foram acrescentadas ao DF Historico as colunas Tipo Disciplina e Semestre Disciplina provenientes do DF df_disciplinas e as colunas

Semestre por Tempo e Semestre de Avaliação provenientes do DF aux_equivalencia, Figura 43. Nos DFs Historico e aux_equivalencia, acrescentou-se a coluna Matricula_Per_Letivo, que é a concatenação das colunas de Matrícula e Per_Letivo, que será usada no futuro como relacionamento entre os dois DFs. Nessa etapa também foi acrescentada uma nova coluna ao DF Historico chamada “Antecipação | Atraso – Sem. Avaliação”, essa coluna é calculada pela subtração entre o semestre de avaliação do aluno e o semestre da disciplina que ele se matriculou, ou seja, resultados menores que 0 significam antecipação, maiores que 0 significam atraso e valores iguais a 0 indicam que o semestre de avaliação é igual ao semestre da disciplina.

Figura 43: Acrescentando algumas colunas ao DF Historico.

```
#Acrescentando algumas colunas ao data frame Historico
Historico = pd.merge(Historico, df_disciplinas.loc[:,['Cód. Disciplina', 'Tipo Disciplina', 'Semestre Disciplina']], left_on = 'Discip.',
                    right_on = 'Cód. Disciplina', how = 'left') #Acrescentando as colunas 'Tipo Disciplina' e 'Semestre Disciplina' do data frame df_disciplinas
Historico.drop('Cód. Disciplina', axis = 1, inplace = True) #Excluindo a coluna 'Cód. Disciplina' que foi acrescentada pelo processo acima
Historico['Matricula_Per_Letivo'] = Historico['Matricula'].astype(str) + Historico['Per_Letivo'].astype(str)
#Na linha acima, criando uma coluna auxiliar, concatenação de 'Matricula' e 'Per_Letivo', ao data frame Historico
aux_equivalencia['Matricula_Per_Letivo'] = aux_equivalencia['Matricula'].astype(str) + aux_equivalencia['Per_Letivo'].astype(str)
#Na linha acima, criando uma coluna auxiliar, concatenação de 'Matricula' e 'Per_Letivo', ao data frame aux_equivalencia
Historico = pd.merge(Historico, aux_equivalencia.loc[:, ['Matricula_Per_Letivo', 'Semestre por Tempo', 'Semestre de Avaliação']], how = 'left')
#Na linha acima, acrescentando as colunas 'Semestre por tempo' e 'Semestre de Avaliação' ao data frame historico, provenientes do data frame aux_equivalencia
Historico['Antecipação | Atraso - Sem. Avaliação'] = Historico['Semestre de Avaliação'] - Historico['Semestre Disciplina']
#Na linha acima, criando coluna 'Antecipação | Atraso - Sem. Avaliação' = 'Semestre de Avaliação - Semestre Disciplina'
```

Fonte: Elaborado pelo autor.

Criou-se no DF Historico duas colunas G_cump_pre2 e G_cump_pre3 que são uma segmentação em intervalos das colunas Cumprimento de pré-requisito 2 e Cumprimento de pré-requisito 3, respectivamente, Figura 44. Os intervalos criados para essas duas colunas são [0%, 05%), [05%, 30%), [30%, 50%), [50%, 70%), [70%, 95%) e [95%, 100%].

Figura 44: Segmentação em intervalos das colunas Cumprimento de pré-requisito 2 e Cumprimento de pré-requisito 3.

```
#Criando agrupamentos para o cumprimento de pré-requisitos
faixa = [0,0.05,0.30,0.5,0.7,0.95,1.01] #Lista com os limites de cada intervalo
Historico['G_cump_pre2'] = pd.cut(Historico['Cumprimento de pré-requisito 2'], bins = faixa, right = False, labels = ['[0, 05%)', '[05%, 30%)', '[30%, 50%)',
                                                            '[50%, 70%)', '[70%, 95%)', '[95%, 100%]'])

#Na linha acima, criando coluna com agrupamento para o pré-requisito, considerando pré-requisitos até 3ª Ordem
Historico['G_cump_pre3'] = pd.cut(Historico['Cumprimento de pré-requisito 3'], bins = faixa, right = False, labels = ['[0, 05%)', '[05%, 30%)', '[30%, 50%)',
                                                            '[50%, 70%)', '[70%, 95%)', '[95%, 100%]'])

#Na linha acima, criando coluna com agrupamento para o pré-requisito, considerando pré-requisitos até 3ª Ordem com pesos de semestre do pré-requisito
```

Fonte: Elaborado pelo autor.

Em seguida foi criado um DF chamado sugestao, Figura 45, onde cada aluno ativo do período letivo vigente terá quatro sugestões de disciplinas. Esse novo DF tem a mesma estrutura do DF aux_equivalencia mantendo apenas as colunas Matrícula, Ingresso, Per_Letivo e todas as disciplinas obrigatórias, com exceção das novas disciplinas de física. Além disso, para a coluna Per_Letivo, só foram consideradas as linhas do último período que se tinha registro no DF aux_equivalencia. Todas as colunas de disciplinas do DF sugestao foram transformadas em linhas e acrescentou-se também as colunas Semestre Disciplina e Nome da Disciplina do DF df_disciplinas. Após isso, classificou-se o DF sugestao do menor para o maior pela coluna Semestre Disciplina. Por fim, foram filtradas apenas as colunas em que os alunos ainda não haviam sido aprovados e, para cada número de matrícula, foi mantida as quatro primeiras disciplinas. Logo, as sugestões priorizam as disciplinas que são de um menor semestre.

Figura 45: Criando um DF de sugestão de disciplinas para os alunos.

```
#Criando um data frame de sugestões de quatro disciplinas para alunos ativos. A sugestão para semestres futuros só ocorrerá quando pelo menos um aluno concluir...
#...uma disciplina do último semestre, seja com aprovação, reprovação, trancamento, etc.
sugestao = aux_equivalencia[aux_equivalencia['Per_Letivo'] == aux_equivalencia['Per_Letivo'].sort_values(
    ).unique()[-1]].drop(['Semestre por Tempo', 'Tx. Conclusão Disciplinas', 'Tx. Conclusão Cálculo',
    'Tx. Conclusão Física', 'Tx. Conclusão Programação',
    'Tx. Conclusão Profissionalizante', 'Tx. Conclusão Automação',
    'Tx. Conclusão Controle', 'FISD36', 'FISD42', 'FISD34', 'FISD41', 'FISD37',
    'FISD40', 'FISD38', 'FISD39'], axis = 1)

#Acima, criando um data frame chamado sugestão com a mesma estrutura de aux_equivalencia com exceção de algumas colunas e filtrando apenas o período letivo de...
#...maior valor.
sugestao = pd.melt(sugestao, id_vars = ['Matrícula', 'Ingresso', 'Per_Letivo', 'Semestre de Avaliação'], var_name = 'Cód. Disciplina')
#Na linha acima, transformando as colunas referentes à disciplinas em linhas
sugestao = pd.merge(sugestao, df_disciplinas.loc[:,['Cód. Disciplina', 'Semestre Disciplina', 'Nome da Disciplina']],
    how = 'left').sort_values(by = ['Matrícula', 'Semestre Disciplina'])
#Na linha acima, acrescentando as colunas 'Semestre Disciplina' e 'Nome da Disciplinas' ao data frame sugestão e classificando as linhas pelo número de matrícula...
#...e semestre da disciplina.
sugestao = sugestao[sugestao['value'] == 0].groupby('Matrícula').head(4) #Filtrando apenas as 4 disciplinas com semestre mais baixo para cada aluno ativo.
```

Fonte: Elaborado pelo autor.

Foi criado um DF chamado vagas_planejamento com a mesma estrutura do DF df_disciplinas, excetuando a coluna Tipo Disciplina, classificado do menor para o maior pela coluna Semestre Disciplina, Figura 46. Em seguida, foi acrescentada uma coluna chamada Vagas, que é a contagem de vezes que cada disciplina aparece no DF sugestao. Caso uma disciplina não apareça, ela é preenchida com o valor 0, ou seja, a coluna Vagas representa a quantidade de vagas necessárias em cada disciplina para atender todas as necessidades dos alunos do DF sugestao.

Figura 46: Criando o DF vagas_planejamento.

```
#Criando um data frame que mostra a quantidade de vagas necessárias para cada disciplina para que atendida as necessidades dos alunos de acordo com o data frame...
#de sugestao.
vagas_planejamento = df_disciplinas.sort_values(by = 'Semestre Disciplina')[['Cód. Disciplina', 'Nome da Disciplina', 'Semestre Disciplina']]
#Na linha acima, criando o data frame vagas_planejamento com a mesma estrutura do data frame df_disciplinas excluindo a coluna 'Tipo Disciplina' e classificando...
#por 'Semestre da disciplina'.
vagas_planejamento['Vagas'] = vagas_planejamento['Cód. Disciplina'].map(sugestao['Cód. Disciplina'].value_counts()).fillna(0)
#Na linha acima, acrescentando a coluna 'Vagas' que é uma contagem da quantidade de vezes que cada disciplina aparece no data frame sugestao e preenchendo...
#possíveis valores null com 0 para disciplinas que não apareceram como sugestão.
```

Fonte: Elaborado pelo autor.

Na Figura 47, foi criado um DF chamado alunos com intuito de trazer algumas estatísticas referente aos alunos ao longo de sua vida

acadêmica. Esse DF foi criado a partir do DF aux_equivalencia, mantendo uma linha para cada número de matrícula e, para as colunas Semestre por Tempo e Tx. Conclusão Disciplinas, foi utilizado o maior valor encontrado no DF aux_equivalencia, ou seja, o mais atual. Em seguida, um DF auxiliar chamado taxa foi criado através de uma tabela dinâmica, onde o índice foi o número de matrícula dos alunos e as colunas são as contagens de resultados AP, RF, RR, TR e o total de cada linha. A partir daí, criou-se as colunas %Rep. Falta, %Rep. Nota e %Trancamento calculadas pela divisão entre a contagem de RF, RR e TR dividido cada um pelo total de cada linha. Depois, as colunas referentes a contagem de cada resultado e a linha referente ao total de cada coluna foram excluídas. A coluna referente ao total de cada linha foi renomeada para Qtd. de Inscrições, que representa a quantidade de inscrições em disciplinas que cada aluno realizou ao longo da vida acadêmica. Em seguida, criou-se uma coluna igual ao índice com os números de matrícula para auxiliar, futuramente, na junção dos dois DFs e os números de índice foram resetados para o padrão. Mesclou-se ao DF alunos todas as colunas do DF taxa e preencheu-se possíveis células vazias com o valor 0. Por fim, foi criada uma coluna ao DF alunos chamada Semestres Afastado, que mostra quantas vezes cada aluno se afastou completamente do curso, considerando os dados contidos no DF alunos_sem_inscricao. A Tabela 3 a seguir mostra em quais registros da coluna Nome da Disciplina do DF alunos_sem_inscrição foi considerado, ou não, se o aluno estava afastado.

Tabela 3: Tipos de registros em que considerou ou não se o aluno estava afastado.

Considerou que estava afastado	Não considerou que estava afastado
Sem Inscrição em Disciplina	Trancamento Especial SLS - Res. 04/2020 CONSUNI
Trancamento de Matricula	Trancamento Especial - Res. 04/2020 CONSUNI
Trancamento de Matrícula Especial	MATRÍCULA CONDICIONAL - § 6 - Art. 74 do REG
Trancamento Esp.-§3-Art.47-Lei 9394/96	Matrícula Condicional §6 - Art.77 REGPG
Trancamento por Tempo Determinado	Aluno em Intercâmbio Ciência Sem Fronteiras
Trancamento Total - SMU	Concedido Permanencia
Trancamento Total Inscricao Disciplinas	Matrícula Condicional - ART 75, I e II, REGPG
Trancamento Total Problema Saúde	Dilatacao de Prazo

Fonte: Elaborado pelo autor.

Figura 47: Criando o DF alunos.

```

#Criando data frame que mostra algumas informações e estatísticas de cada aluno
alunos = aux_equivalencia.groupby('Matrícula')[['Semestre por Tempo', 'Tx. Conclusão Disciplinas']].max().reset_index()
#Na linha acima, criando data frame aluno com índice 'Matrícula' e colunas 'semestre por Tempo' e 'Tx. Conclusão Disciplinas' (valores para o último semestre em...
#...que o aluno apareceu no histórico)
taxa = Historico[-(Historico['Resultado'].isin(['DI', 'DU'])) &
                -(Historico['Resultado'].isin(['NC', 'RP']))].pivot_table(values = 'Discip.', index = 'Matrícula', columns = 'Resultado', aggfunc = 'count',
                                fill_value = 0, margins = True)
#Nas linhas acima, criando data frame auxiliar, que mostra quantidade de cada resultado para cada aluno, excluindo os resultados 'NC', 'RP', 'DI' e 'DU'
taxa['%Rep. Falta'] = taxa['RF']/taxa['All'] #Calculando taxa de Reprovação por falta
taxa['%Rep. Nota'] = taxa['RR']/taxa['All'] #Calculando taxa de Reprovação por nota
taxa['%Trancamento'] = taxa['TR']/taxa['All'] #Calculando taxa de Trancamento
taxa.drop(['AP', 'RF', 'RR', 'TR'], axis = 1, inplace = True) #Excluindo colunas desnecessárias
taxa.drop(['All'], axis = 0, inplace = True) #Excluindo linha referente ao total
taxa = taxa.rename(columns={'All': 'Qtd. de Inscrições'}) #Renomeando coluna 'All' para 'Qtd. de Inscrições'
taxa['Matrícula'] = taxa.index #Criando uma nova coluna igual ao índice, pois o número de matrícula era o índice desse data frame
taxa.reset_index(drop=True, inplace=True) #Fazendo um reset no índice
alunos = pd.merge(alunos, taxa, left_on = 'Matrícula', right_on = 'Matrícula', how = 'left') #Juntando as informações do data frame taxa no data frame alunos
alunos[['Qtd. de Inscrições', '%Rep. Falta', '%Rep. Nota', '%Trancamento']] = alunos[['Qtd. de Inscrições', '%Rep. Falta', '%Rep. Nota', '%Trancamento']].fillna(0)
#Na linha acima, preenchendo os valores null com 0.
#Na linha abaixo, criando uma lista para desconsiderar no próximo data frame.
desconsiderar = ['Trancamento Especial SLS - Res. 04/2020 CONSUNI ', 'Trancamento Especial - Res. 04/2020 CONSUNI ',
                'MATRÍCULA CONDICIONAL - § 6 - Art. 74 do REG', 'Matrícula Condicional §6 - Art.77 REGPG ', 'Aluno em Intercâmbio Ciência Sem Fronteiras ',
                'Concedido Permanencia ', 'Matrícula Condicional - ART 75, I e II, REGPG ', 'Dilatacao de Prazo']
alunos_sem_inscricao = alunos_sem_inscricao[-alunos_sem_inscricao['Nome da Disciplina'].isin(desconsiderar)]
#A proposta do data frame acima é ter as entradas em que os alunos não se matricularam em nenhuma disciplina com intuito de saber em quantos semestres ele não...
#...se matriculou. Foi aplicado um filtro para tratar as linhas que não correspondem a evasão
alunos['Semestres Afastado'] = alunos['Matrícula'].map(alunos_sem_inscricao['Matrícula'].value_counts()).fillna(0).astype(int)
#Na linha acima, acrescentando ao data frame alunos, quantas vezes cada aluno deixou de se matricular no data frame alunos_sem_inscricao e preenchendo possíveis...
#...valores null com 0 para alunos que sempre se matricularam

```

Fonte: Elaborado pelo autor.

Os DFs alunos_ativos, alunos_sem_inscrição, AP_ou_DI_e_DU_pandemia, data, DI_DU_fora_pandemia, taxa, pre_requisitos e df_disciplinas foram apagados, pois não serão mais utilizados, Figura 48.

Figura 48: Apagando os DFs que não serão mais utilizados.

```

#Na linha abaixo, apagando data frames que não serão utilizados.
del alunos_ativos, alunos_sem_inscricao, AP_ou_DI_e_DU_pandemia, data, DI_DU_fora_pandemia, taxa, pre_requisitos, df_disciplinas

```

Fonte: Elaborado pelo autor.

8 APÊNDICE B: INTEGRAÇÃO ENTRE PYTHON E POWER BI® E IMPORTAÇÃO DE DADOS

8.1 INTEGRAÇÃO PYTHON E POWER BI®

Segundo a Microsoft (2023b), a versão do Python suportada pelo Power BI® é a 3.7.7, porém a versão mais atual do Python é a 3.12, superior à versão suportada pelo Power BI®. Então, foi necessário criar um ambiente virtual com o Python na versão 3.7.7 para evitar problemas durante a integração entre Python e Power BI®.

Para criar esse ambiente virtual foi utilizado o Anaconda Prompt, terminal proveniente do Anaconda, que já é configurado para execução dos comandos do Python. Seguindo os comandos da documentação do Conda (2017), foi utilizado o comando ‘conda create --name envpowerbi python=3.7.7’ no terminal para a criação do ambiente virtual chamado envpowerbi com a versão do python 3.7.7. O terminal informará em qual pasta será criado esse ambiente e quais pacotes serão instalados, Figura 49.

Figura 49: Criando ambiente virtual com versão do python 3.7.7 chamado envpowerbi.

```

Anaconda Prompt - conda create --name envpowerbi python=3.7.7

(base) C:\Users\Jonathas>conda create --name envpowerbi python=3.7.7
Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful attempt using repodata from current_repodata.json, retrying with next repodata source. Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 23.9.0

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=23.9.0

## Package Plan ##

  environment location: C:\Users\Jonathas\anaconda3\envs\envpowerbi

  added / updated specs:
    - python=3.7.7

The following packages will be downloaded:

  package                                     build                                     size
  -----
  certifi-2022.12.7                          py37haa95532_0                          149 KB
  openssl-1.1.1w                              h2bbff1b_0                              5.5 MB
  pip-22.3.1                                  py37haa95532_0                          2.7 MB
  python-3.7.7                                h81c818b_4                              14.3 MB
  setuptools-65.6.3                          py37haa95532_0                          1.1 MB
  wheel-0.38.4                                py37haa95532_0                          82 KB
  wincertstore-0.2                            py37haa95532_2                          15 KB
  -----
  Total:                                     23.9 MB

The following NEW packages will be INSTALLED:

  ca-certificates pkgs/main/win-64::ca-certificates-2023.08.22-haa95532_0
  certifi         pkgs/main/win-64::certifi-2022.12.7-py37haa95532_0
  openssl        pkgs/main/win-64::openssl-1.1.1w-h2bbff1b_0
  pip            pkgs/main/win-64::pip-22.3.1-py37haa95532_0
  python         pkgs/main/win-64::python-3.7.7-h81c818b_4
  setuptools     pkgs/main/win-64::setuptools-65.6.3-py37haa95532_0
  sqlite         pkgs/main/win-64::sqlite-3.41.2-h2bbff1b_0
  vc             pkgs/main/win-64::vc-14.2-h21ff451_1
  vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
  wheel          pkgs/main/win-64::wheel-0.38.4-py37haa95532_0
  wincertstore   pkgs/main/win-64::wincertstore-0.2-py37haa95532_2

Proceed ([y]/n)?

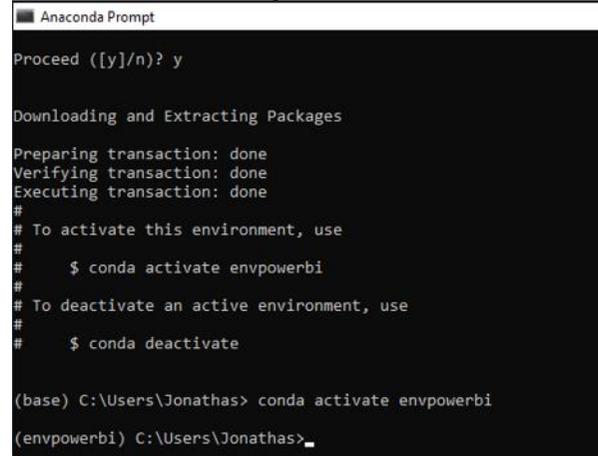
```

Fonte: Elaborado pelo autor.

Em seguida, será solicitada a confirmação com ‘y’ para sim e ‘n’ para não. Após confirmar, será realizado o download e extração dos

pacotes. Posteriormente, utilizou-se o comando ‘conda activate envpowerbi’ para acessar e usar o novo ambiente, Figura 50.

Figura 50: Confirmando a criação do ambiente virtual e acessando-o.



```
Anaconda Prompt
Proceed ([y]/n)? y
Downloading and Extracting Packages
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate envpowerbi
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\Jonathas> conda activate envpowerbi
(envpowerbi) C:\Users\Jonathas>
```

Fonte: Elaborado pelo autor.

Após criar e acessar o ambiente virtual, é necessário fazer a instalação de bibliotecas e pacotes que serão utilizadas. O primeiro a ser instalado foi o Jupyter, utilizando o comando ‘pip install jupyter’, Figura 51.

Figura 51: Instalando o pacote Jupyter.

```

Anaconda Prompt

(base) C:\Users\Jonathas> conda activate envpowerbi

(envpowerbi) C:\Users\Jonathas> pip install jupyter
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting ipykernel
  Downloading ipykernel-6.16.2-py3-none-any.whl (138 kB)
----- 138.5/138.5 kB 1.2 MB/s eta 0:00:00
Collecting notebook
  Downloading notebook-6.5.6-py3-none-any.whl (529 kB)
----- 529.8/529.8 kB 5.6 MB/s eta 0:00:00
Collecting jupyter-console
  Downloading jupyter_console-6.6.3-py3-none-any.whl (24 kB)
Collecting qtconsole
  Downloading qtconsole-5.4.4-py3-none-any.whl (121 kB)
----- 121.9/121.9 kB 7.4 MB/s eta 0:00:00

```

Fonte: Elaborado pelo autor.

Após aguardar o download e instalação de todos os pacotes, foi utilizado o comando ‘pip install pandas’ para instalação da biblioteca pandas, Figura 52.

Figura 52: Instalando o pacote pandas.

```

Anaconda Prompt

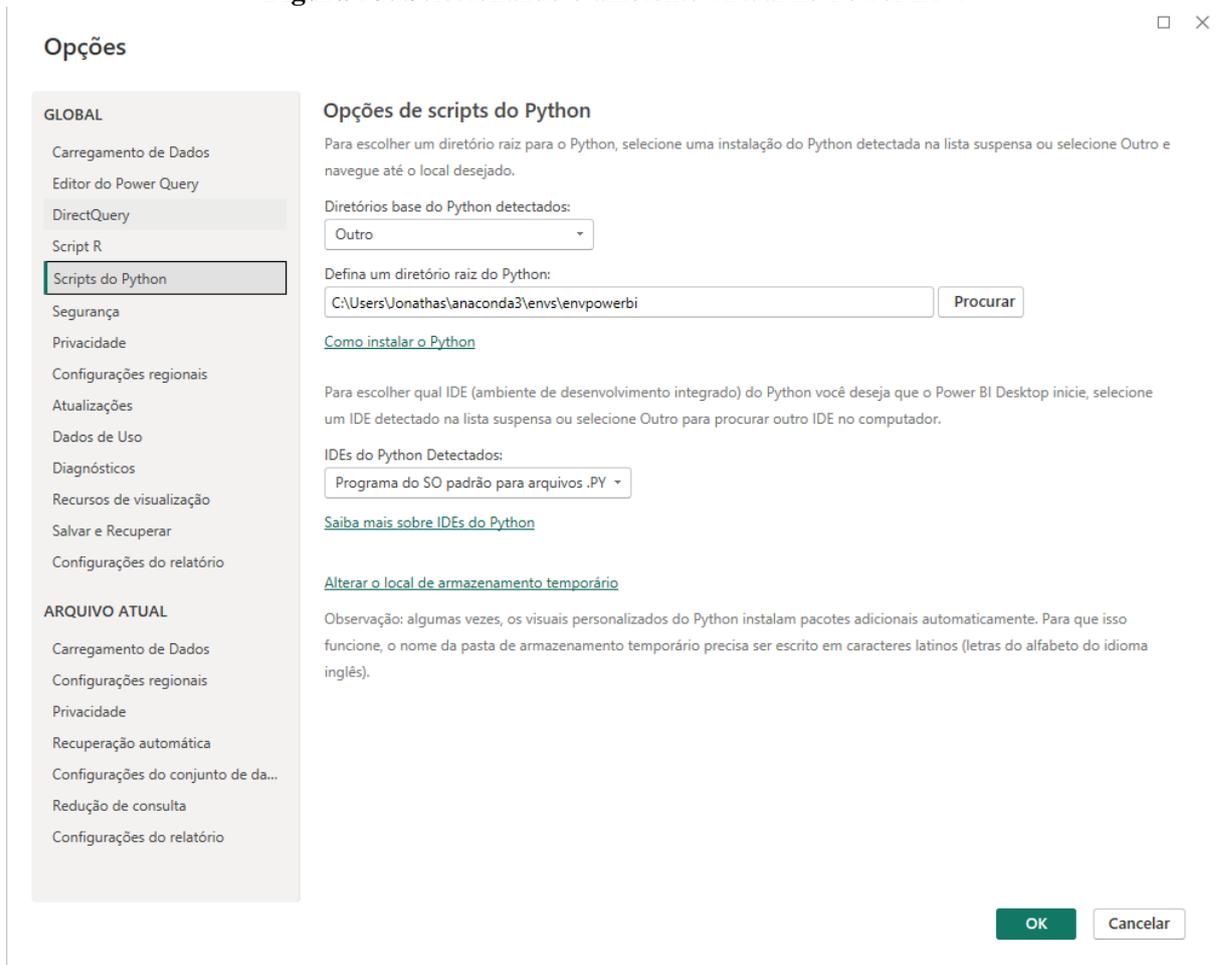
(envpowerbi) C:\Users\Jonathas> pip install pandas
Collecting pandas
  Downloading pandas-1.3.5-cp37-cp37m-win_amd64.whl (10.0 MB)
----- 10.0/10.0 MB 13.9 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\jonathas\anaconda3\envs\envpowerbi\lib\site-packages (from pandas) (2.8.2)
Collecting numpy>=1.17.3
  Downloading numpy-1.21.6-cp37-cp37m-win_amd64.whl (14.0 MB)
----- 14.0/14.0 MB 18.2 MB/s eta 0:00:00
Collecting pytz>=2017.3
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
----- 502.5/502.5 kB 15.4 MB/s eta 0:00:00
Requirement already satisfied: six>=1.5 in c:\users\jonathas\anaconda3\envs\envpowerbi\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Installing collected packages: pytz, numpy, pandas
Successfully installed numpy-1.21.6 pandas-1.3.5 pytz-2023.3.post1

```

Fonte: Elaborado pelo autor.

Após realizar as instalações, foi necessário configurar o Power BI® para detectar o ambiente virtual criado no passo anterior. Para isso, no Power BI®, deve-se seguir o caminho: Arquivo > Opções e configurações > Opções > Scripts do Python. No campo Diretórios base do Python detectados foi escolhida a opção Outro e, em seguida, no campo Defina um diretório raiz do Python, foi escolhido o diretório onde foi realizada a instalação do ambiente virtual, Figura 53.

Figura 53: Selecionando o ambiente virtual no Power BI®.

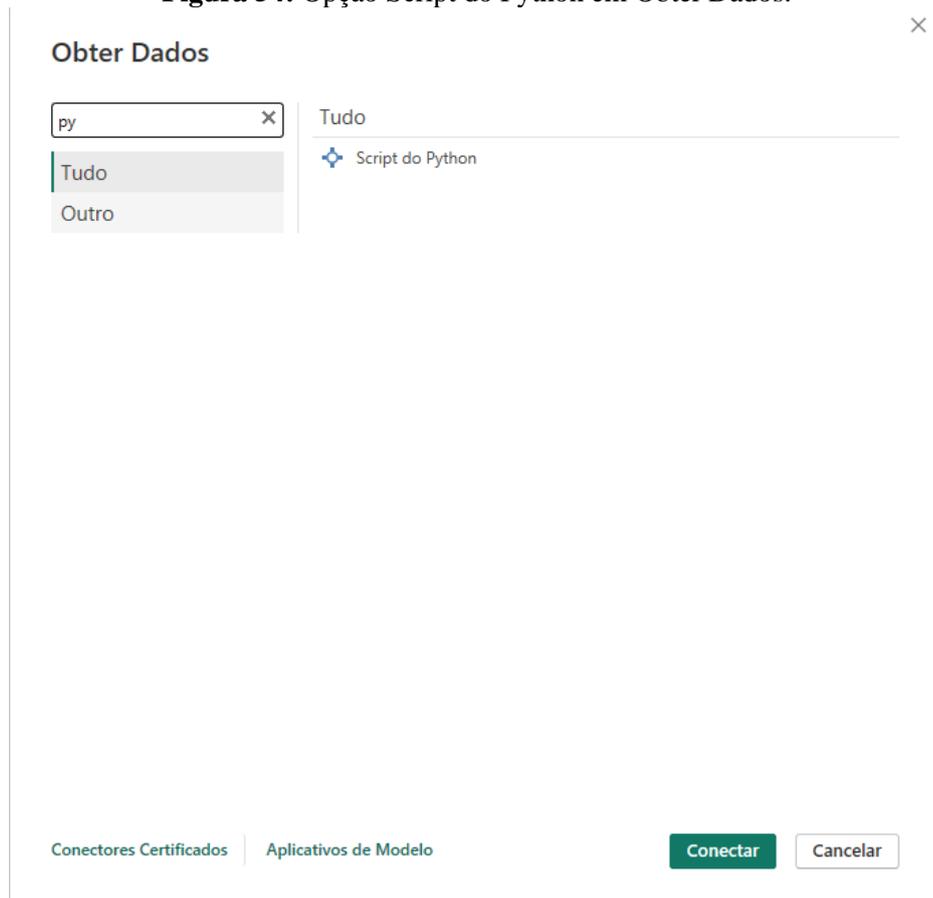


Fonte: Elaborado pelo autor.

8.2 IMPORTAÇÃO DOS DADOS

Para importar os dados é preciso seguir o caminho Página Inicial > Obter dados > Mais... > e, após isso, selecionar a opção Script do Python, Figura 54.

Figura 54: Opção Script do Python em Obter Dados.



Fonte: Elaborado pelo autor.

Em seguida, na tela que apareceu, foi colado todo o *script* desenvolvido no Apêndice A desse trabalho, Figura 55.

Figura 55: Colando o script do Python no Power BI®.



Fonte: Elaborado pelo autor.

Após carregar o *script*, foram marcadas todas as tabelas importadas e selecionada a opção Transformar Dados, Figura 56.

Figura 56: Selecionando as tabelas importadas do script.

The screenshot shows a window titled "Navegador" with a search bar and a list of tables under the "Python [6]" category. The table "vagas_planejamento" is selected. The interface includes a search bar, a list of tables with checkboxes, and buttons for "Carregar", "Transformar Dados", and "Cancelar".

Cód. Disciplina	Nome da Disciplina	Semes
MATA01	Geometria Analítica	
MATA02	Cálculo A	
ENGF57	Introdução à Engenharia de Controle e Automação	
QUIB50	Fundamentos de Química	
MATA38	Projeto de Circuitos Lógicos	
FIS121	Física Geral e Experimental I-E	
MATA03	Cálculo B	
MATA07	Álgebra Linear A	
FISD36	Física Geral Teórica I	
FISD42	Física Geral Experimental I	
FISD41	Física Geral Experimental II	
FISD34	Física Geral Teórica II	
MATA04	Cálculo C	
FIS122	Física Geral e Experimental II-E	
MATA37	Introdução à Lógica de Programação	
MATA48	Arquitetura de Computadores	
MATA40	Estrutura de Dados e Algoritmos I	
ENGD02	Estatística na Engenharia	
FIS123	Física Geral e Experimental III-E	
FISD37	Física Geral Teórica III	
FISD40	Física Geral Experimental III	
FISD38	Física Geral Teórica IV	
FISD39	Física Geral Experimental IV	

Fonte: Elaborado pelo autor.

9 APÊNDICE C: CÁLCULO DAS MEDIDAS

A Tabela 4 a seguir mostra as medidas calculadas no Power BI[®] através das funções DAX.

Tabela 4: Funções DAX utilizadas no trabalho.

Nome	Descrição	Função
%Aprovação	Calcula a porcentagem de aprovações. Na tabela Historico, é feita a divisão entre a contagem de linhas com resultado AP e a contagem de linhas com resultados AP, RF, RR e TR.	%Aprovações = <code>DIVIDE(CALCULATE(COUNTROWS(Historico); Historico[Resultado] = "AP"); CALCULATE(COUNTROWS(Historico); Historico[Resultado] IN {"AP"; "RF"; "RR"; "TR"})); 0)</code>
%Rep. Falta	Calcula a porcentagem de reprovações por falta. Na tabela Historico, é feita a divisão entre a contagem de linhas com resultado RF e a contagem de linhas com resultados AP, RF, RR e TR.	%Rep. Falta = <code>DIVIDE(CALCULATE(COUNTROWS(Historico); Historico[Resultado] = "RF"); CALCULATE(COUNTROWS(Historico); Historico[Resultado] IN {"AP"; "RF"; "RR"; "TR"})); 0)</code>
%Rep. Nota	Calcula a porcentagem de reprovações por nota. Na tabela Historico, é feita a divisão entre a contagem de linhas com resultado RR e a contagem de linhas com resultados AP, RF, RR e TR.	%Rep. Nota = <code>DIVIDE(CALCULATE(COUNTROWS(Historico); Historico[Resultado] = "RR"); CALCULATE(COUNTROWS(Historico); Historico[Resultado] IN {"AP"; "RF"; "RR"; "TR"})); 0)</code>
%Reprovações	Calcula a porcentagem de reprovações por falta e por nota. Na tabela Historico, é feita a divisão entre a contagem de linhas com resultados RF e RR e a contagem de linhas com resultados AP, RF, RR e TR.	%Reprovações = <code>DIVIDE(CALCULATE(COUNTROWS(Historico); Historico[Resultado] in {"RF"; "RR"}); CALCULATE(COUNTROWS(Historico); Historico[Resultado] IN {"AP"; "RF"; "RR"; "TR"})); 0)</code>
%Trancamento	Calcula a porcentagem de trancamento. Na tabela Historico, é feita a divisão entre a contagem de linhas com resultado TR e a contagem de linhas com resultados AP, RF, RR e TR.	%Trancamento = <code>DIVIDE(CALCULATE(COUNTROWS(Historico); Historico[Resultado] = "TR"); CALCULATE(COUNTROWS(Historico); Historico[Resultado] IN {"AP"; "RF"; "RR"; "TR"})); 0)</code>
Alunos ativos	Calcula a quantidade de alunos ativos. Na tabela Historico, é feita a contagem de valores distintos de matrícula.	Alunos Ativos = <code>DISTINCTCOUNT(Historico[Matrícula])</code>

Continua

Nome	Descrição	Função
Alunos sem inscrição	Calcula a quantidade de alunos que não se matricularam. Na tabela Historico_completo, é feita uma contagem de valores distintos de matrícula, considerando apenas linhas cujo valor da coluna Nome Status era uma das opções a seguir: Sem Inscrição em Disciplina, Trancamento de Matrícula, Trancamento Total Inscricao Disciplinas, Trancamento Total Problema Saúde, Trancamento por Tempo Determinado, Trancamento Total – SMU, Trancamento Esp.-§3- Art.47-Lei 9394/96, Trancamento de Matrícula Especial.	Alunos sem Inscrição = <code>CALCULATE(DISTINCTCOUNT(Historico_completo[Matrícula]); Historico_completo[Nome Status] in {"Sem Inscrição em Disciplina "; "Trancamento de Matrícula"; "Trancamento Total Inscricao Disciplinas "; "Trancamento Total Problema Saúde"; "Trancamento por Tempo Determinado "; "Trancamento Total - SMU "; "Trancamento Esp.-§3-Art.47- Lei 9394/96"; "Trancamento de Matrícula Especial "})</code>
Aprovações	Calcula a quantidade de aprovações. Na tabela Historico, é feita uma contagem das linhas cujo resultado é igual a AP.	Aprovações = <code>CALCULATE(COUNT(Historico[Resultado]); Historico[Resultado] = "AP")</code>
Eletivas	Calcula a quantidade de inscrições em disciplinas eletivas. Na tabela Historico, é feita uma subtração entre a contagem das linhas em que a coluna Natureza não tem valor e o resultado é diferente de DI e DU e a contagem das linhas referentes as disciplinas novas de física quando o resultado é diferente de DI e DU (subtração se faz necessária pois as disciplinas novas de física não têm valor na coluna Natureza).	Eletivas = <code>CALCULATE(COUNTROWS(Historico); Historico[Natureza] = BLANK(); Historico[Resultado] <> "DI"; Historico[Resultado] <> "DU") - CALCULATE(COUNTROWS(Historico); Historico[Cód. Disciplina] in {"FISD36"; "FISD42"; "FISD34"; "FISD41"; "FISD37"; "FISD40"; "FISD38"; "FISD39"}; Historico[Resultado] <> "DI"; Historico[Resultado] <> "DU")</code>

Continua

Nome	Descrição	Função
Média de disciplinas por aluno	Calcula a quantidade média de disciplinas por aluno, fazendo uma divisão das medidas N° de inscrições em Disciplinas por Alunos Ativos	Média de disciplinas por aluno = $[N^{\circ} \text{ de inscrições em Disciplinas}] / [\text{Alunos Ativos}]$
N° de inscrições em Disciplinas	Calcula a quantidade de inscrições em disciplinas. Na tabela Historico, é feita a contagem de linhas não vazias da coluna Cód. Disciplina e que o resultado é diferente de DI e DU.	$N^{\circ} \text{ de inscrições em Disciplinas} = \text{CALCULATE}(\text{COUNT}(\text{Historico}[\text{Cód. Disciplina}]); \text{Historico}[\text{Cód. Disciplina}] \neq \text{BLANK}()); \text{Historico}[\text{Resultado}] \neq \text{"DI"}; \text{Historico}[\text{Resultado}] \neq \text{"DU"})$
Obrigatórias	Calcula a quantidade de inscrições em disciplinas obrigatórias. Na tabela Historico, é feita a contagem de linhas cujo valor da coluna Natureza são iguais a OB e o resultado é diferente de DI e DU, somada à contagem das linhas que contêm as disciplinas novas de física e o resultado é diferente de DI e DU.	$\text{Obrigatórias} = \text{CALCULATE}(\text{COUNTROWS}(\text{Historico}); \text{Historico}[\text{Natureza}] = \text{"OB"}; \text{Historico}[\text{Resultado}] \neq \text{"DI"}; \text{Historico}[\text{Resultado}] \neq \text{"DU"}) + \text{CALCULATE}(\text{COUNTROWS}(\text{Historico}); \text{Historico}[\text{Cód. Disciplina}] \text{ in } \{\text{"FISD36"}; \text{"FISD42"}; \text{"FISD34"}; \text{"FISD41"}; \text{"FISD37"}; \text{"FISD40"}; \text{"FISD38"}; \text{"FISD39"}\}); \text{Historico}[\text{Resultado}] \neq \text{"DI"}; \text{Historico}[\text{Resultado}] \neq \text{"DU"})$
Optativas	Calcula a quantidade de inscrições em disciplinas optativas. Na tabela Historico, é feita a contagem de linhas cujo valor da coluna Natureza são iguais a OP e o resultado é diferente de DI e DU	$\text{Optativas} = \text{CALCULATE}(\text{COUNTROWS}(\text{Historico}); \text{Historico}[\text{Natureza}] = \text{"OP"}; \text{Historico}[\text{Resultado}] \neq \text{"DI"}; \text{Historico}[\text{Resultado}] \neq \text{"DU"})$
Rep. Falta	Calcula a quantidade de reprovações por falta. Na tabela Historico, é feita uma contagem das linhas cujo resultado é igual a RF.	$\text{Rep. Falta} = \text{CALCULATE}(\text{COUNT}(\text{Historico}[\text{Resultado}]); \text{Historico}[\text{Resultado}] = \text{"RF"})$
Rep. Nota	Calcula a quantidade de reprovações por nota. Na tabela Historico, é feita uma contagem das linhas cujo resultado é igual a RF.	$\text{Rep. Nota} = \text{CALCULATE}(\text{COUNT}(\text{Historico}[\text{Resultado}]); \text{Historico}[\text{Resultado}] = \text{"RR"})$
Reprovações	Calcula a quantidade de reprovações. Na tabela Historico, é feita uma contagem das linhas cujo resultado é igual a RF ou RR.	$\text{Reprovações} = \text{CALCULATE}(\text{COUNT}(\text{Historico}[\text{Resultado}]); \text{Historico}[\text{Resultado}] \text{ in } \{\text{"RF"}; \text{"RR"}\})$

Continua

Nome	Descrição	Função
Taxa Reprovação Últimos 2 Semestres	Calcula a taxa de reprovações nos dois semestres mais recentes. Na tabela Historico, é feita uma contagem das linhas cujo valor da coluna Rank Per_Letivo é igual a 1 ou 2 e Resultado igual a AP, RF, RR ou TR. Se o valor da contagem for igual a vazio, então o resultado é [Sem Inscrição]. Se o valor da contagem for diferente de vazio, o resultado será a divisão da contagem das linhas, cujo valor da coluna Rank Per_Letivo é igual a 1 ou 2 e resultado igual a RF ou RR, pela primeira contagem.	<pre>Taxa Reprovação Últimos 2 Semestres = IF(ISBLANK(CALCULATE(COUNTROWS(Historico); Historico[Rank Per_Letivo] in {1; 2}; Historico[Resultado] in {"AP"; "RF"; "RR"; "TR"})); "Sem Inscrição"; IF(ISBLANK(CALCULATE(COUNTROWS(Historico);Historico[Rank Per_Letivo] in {1; 2}; Historico[Resultado] in {"RF"; "RR"})); 0; DIVIDE(CALCULATE(COUNTROWS(Historico);Historico[Rank Per_Letivo] in {1; 2}; Historico[Resultado] in {"RF"; "RR"}); CALCULATE(COUNTROWS(Historico); Historico[Rank Per_Letivo] in {1; 2}; Historico[Resultado] in {"AP"; "RF"; "RR"; "TR"}); 0)))</pre>
Trancamentos	Calcula a quantidade de trancamentos. Na tabela Historico, é feita uma contagem das linhas cujo resultado é igual a TR.	<pre>Trancamentos = CALCULATE(COUNT(Historico[Resultado]); Historico[Resultado] = "TR")</pre>

Fonte: Elaborado pelo autor.

10 APÊNDICE D: GRUPO DE DISCIPLINAS

Tabela 5: Disciplinas por grupo.

Automação	Cálculo	Controle	Física	Profissionalizante	Programação	Química
Equipamentos e Componentes para Controle e Automação	Geometria Analítica	Princípios dos Processos Contínuos	Física Geral e Experimental I-E	Introdução à Engenharia de Controle e Automação	Projeto de Circuitos Lógicos	Fundamentos de Química
Instrumentação Aplic. À Indústria do Petróleo e Gás Natural	Cálculo A	Análise de Processos e Sistemas I	Física Geral Teórica I	Estatística na Engenharia	Arquitetura de Computadores	
Instrumentação e Automação Industrial I	Cálculo B	Controle de Processos Químicos	Física Geral Experimental I	Métodos Numéricos na Engenharia	Introdução à Lógica de Programação	
Instrumentação e Automação Industrial II	Álgebra Linear A	Análise de Processos e Sistemas II	Física Geral e Experimental II-E	Processos de Fabricação I	Estrutura de Dados e Algoritmos I	
Instrumentação e Automação Industrial III	Cálculo C	Controle Avançado e Multivariável	Física Geral Teórica II	Eletricidade	Laboratório de Programação I	
		Controle e Sistemas Não Lineares	Física Geral Experimental II	Engenharia Econômica e Gestão		
			Física Geral e Experimental III-E	Mecânica dos Sólidos e Materiais		
			Física Geral Teórica III	Ciências do Ambiente		
			Física Geral Experimental III	Aplicações Industriais da Computação		
			Física Geral e Experimental IV-E	Projeto e Planejamento Industrial		
			Física Geral Teórica IV	Planejamento e Controle da Produção		
			Física Geral Experimental IV	Operação e Segurança de Plantas Industriais		
				Estágio em Engenharia de Controle e Automação		
				TCC em Engenharia de Controle e Automação		

Fonte: Elaborado pelo autor.